2014

Effect of Clock and Power Gating on Power Distribution Network Noise in 2D and 3D Integrated Circuits

Vinay C. Patil
University of Massachusetts - Amherst, patil.c.vinay@gmail.com

EFFECT OF CLOCK AND POWER GATING ON POWER DISTRIBUTION NETWORK NOISE IN 2D AND 3D INTEGRATED CIRCUITS

A Thesis Presented

by

VINAY C PATIL

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2014

Department of Electrical and Computer Engineering

EFFECT OF CLOCK AND POWER GATING ON POWER DISTRIBUTION NETWORK NOISE IN 2D AND 3D INTEGRATED CIRCUITS

A Thesis Presented

by

VINAY C PATIL

Approved as to style and content by:

_____
Wayne P. Burleson, Chair

_____
Sandip Kundu, Member

_____
Joseph Bardin, Member

_____
C.V. Hollot, Department Head
Department of Electrical and Computer Engineering

DEDICATION

To my parents and friends

ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Wayne Burleson and Prof. Sandip Kundu for all their support and guidance throughout this work. Their valuable feedback helped me to focus on the topics of this research. I also would like to thank Prof. Joseph Bardin for serving on my committee and providing valuable suggestions during the course of this work. Special thanks to Sudarshan and Arunkumar for several valuable discussions we had regarding this work and other topics of common interest. I would like to thank my lab-mates Krishna, Raghavan and Vikram for their suggestions on several related topics on VLSI and for their support and encouragement. I would also like to thank the University of Massachusetts Amherst for providing a wonderful environment to conduct research. Finally I would like to thank my parents for their constant love and support.

ABSTRACT

EFFECT OF CLOCK AND POWER GATING ON POWER DISTRIBUTION NETWORK NOISE IN 2D AND 3D INTEGRATED CIRCUITS

SEPTEMBER 2014

VINAY C PATIL
B.E., VISVESVARAYA TECHNOLOGICAL UNIVERSITY, INDIA
M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST
Directed by: Prof. Wayne P. Burleson

Increased budgetary constraints on power consumption in modern microprocessors has led to wide-scale adoption of both clock gating and power gating to aid in the reduction of power usage. But, gating introduces its own noise into the Power Distribution Network (PDN). In order to ensure a stable power supply, the worst-case noise from gating must be characterized to verify the integrity of the on-chip power grid.

In this work, power supply noise contribution at a particular Point of Interest (POI) from clock/power gated blocks is maximized at particular time and the *synthetic* gating pattern that results in the maximum noise is obtained for the interval 0 to target time. To aid in the efficient estimation of the noise we utilize wavelet based analysis as wavelets are a natural way of characterizing the time-frequency behavior of the power grid. The fundamental/base wavelets are constructed using the impedance profile of the power grid constituting the frequency-domain behavior of the grid. These wavelets are used as model current sources within the gated blocks and the voltage responses of the grid at the target location from these sources is tabulated accounting for the time-domain behavior of the power grid. The final *synthetic* waveforms of the current sources are composed of wavelets of multiple resolutions and the waveforms are obtained via a Linear Programming (LP) formulation (for clock gating) and Genetic Algorithm based

problem formulation (for Power Gating) which also output the gating patterns of the gated blocks and the maximum supply noise at the Point of Interest at the specified target time using the voltage responses previously tabulated.

We first analyze the effect of Clock Gating on PDN noise for a 2D Integrated Circuit (IC) power grid considering a set of clock gated blocks and a single POI. Then, the problem formulation is extended to a 3D IC by spreading the same set of gated blocks as in the 2D case across 3 tiers of the 3D Power grid. Experimental results will show that the wavelet based approach delivers the worst-case voltage noise and corresponding clock gating pattern for both 2D and 3D IC cases.

For the case of Clock Gating, we notice that the power grid impedance profile does not change during gating while for Power Gating, the power grid impedance profile changes based on which blocks in the grid are gated at a particular time requiring a more complex analysis as the voltage responses from the wavelets are different each time. So, we consider a small set of Power Gated blocks and study the effect of gating them on the supply noise for a single POI in both 2D and 3D ICs. Experimental Results and their analysis show that out approach delivers the worst-case noise and corresponding power gating pattern for both 2D and 3D cases.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Need for Clock and Power Gating

The continuous scaling of transistors and the increase in their frequency of operation has led to an increase in the overall power consumption of the chip. This increase in the complexity of the chip, in accordance with the Moore's Law, has also led to increased power densities within the die as shown in Figure 1.1. Power densities over 100W/cm^2 become unsustainable due to packaging limitations, forcing changes in micro-architecture and circuit design, and throttling of operation frequency to keep the densities within dissipation limits [2].

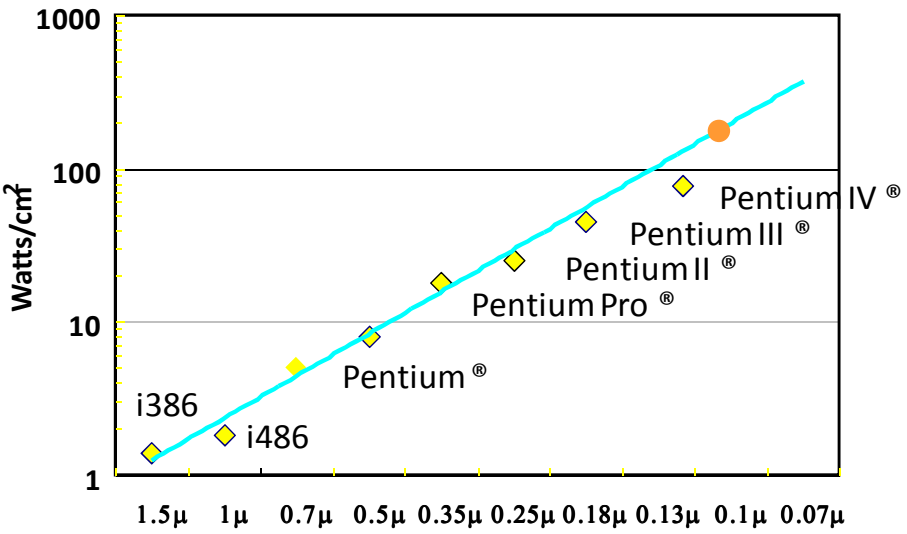


Figure 1.1 Power Density vs Technology Node [1]

A primary consumer of power on-chip is the Clock Distribution Network (CDN) or the Clock Tree. Traditionally, the CDN was said to consume anywhere between 30% to 50 % of the total dynamic power with ~ 80% of the CDN power dissipated in the leaf stages [3]. Certain architectural changes reduced the switching activity of the circuits which reduced the dynamic power consumption of the clock tree to a certain degree. Further analyzing the behavior of logic transitions during real time operations of the circuits allowed the introduction of Clock Gating (CG) which switches off certain Flip-Flops (FFs) when it is found that the input to a Flip-Flop from a previous combinational cloud has not changed. This allows for dynamic power savings down the logic chain as subsequent FFs also can be gated. An example is shown in Figure 1.2 where gating of the two FFs due to switching inactivity of Q1 saves dynamic power in both FFs and the combinational circuits between the FFs [3]. Proper analysis of the logic during Synthesis or RTL implementation can incur larger power savings by gating the clock further up the Clock Tree.

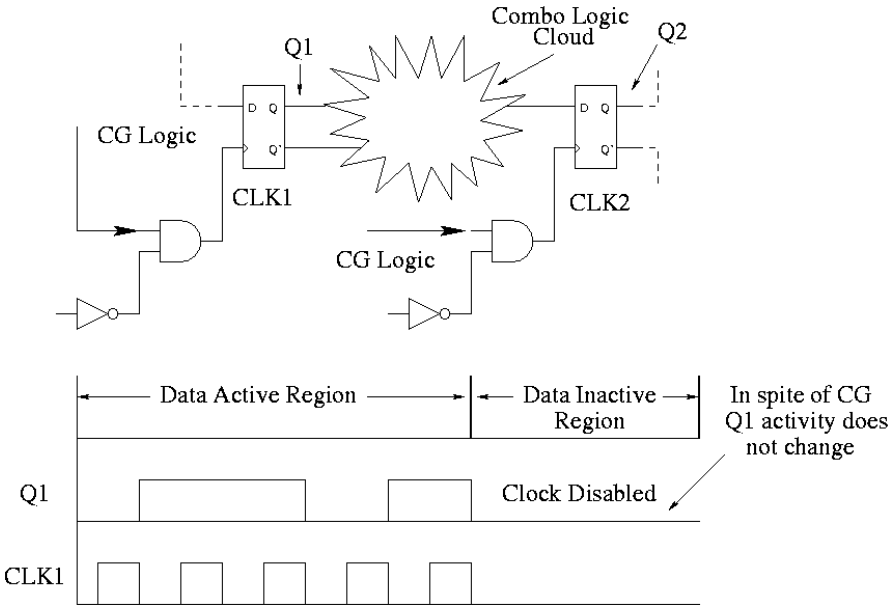


Figure 1.2 Example of Clock Gating [3]

Clock Gating can only reduce dynamic power consumption as leakage power consumption remains unchanged whether a certain circuit block is gated or not gated as the individual transistors in the block are still connected to the power grid. As technology scales down we see an increase in the leakage current [1] which has made leakage power increasingly dominant in advanced technology nodes. There are two types of leakages: active leakage occurs when a circuit is switching and standby leakage when it is idle. Although literature has shown that active leakage is an increasing proportion of the total power consumption, reaching 30% for 65nm technology [4][5], standby leakage is also important to consider as it increases with frequency [6] and must be dealt with, especially in mobile systems where large portions tend to operate in standby or sleep modes for most of the time. Thus, the need to reduce standby leakage power led to the introduction of Power Gating. The conceptual definition of Power Gating is as follows: a circuit is cut off from its power supply in sleep mode by means of a current switch. Figure 1.3 shows a popular implementation of Power Gating using Multi-threshold CMOS logic [7] where SL (and its complement) represent the sleep signal, V_{dd} and V_{ss} (or ground) are the real power lines, V_{ddv} and V_{ssv} are the virtual power lines and Q_1 (header) and Q_2 (footer) are the sleep transistors which act as the current switches mentioned before . In active mode, SL is kept low, Q_1 and Q_2 are turned on, and V_{ddv} and V_{ssv} are maintained close to V_{dd} and V_{ss} respectively. In sleep mode, SL is kept high, Q_1 and Q_2 are turned off, and V_{ddv} and V_{ssv} float; leakage from the low-V_t circuit is thus limited by high-V_t switches. For the sake of simplicity, only one switch may be employed in practice. Use of Power Gating does require additional considerations like data retention during sleep mode, sizing of the current switches, etc. and most importantly the effect on the power grid.

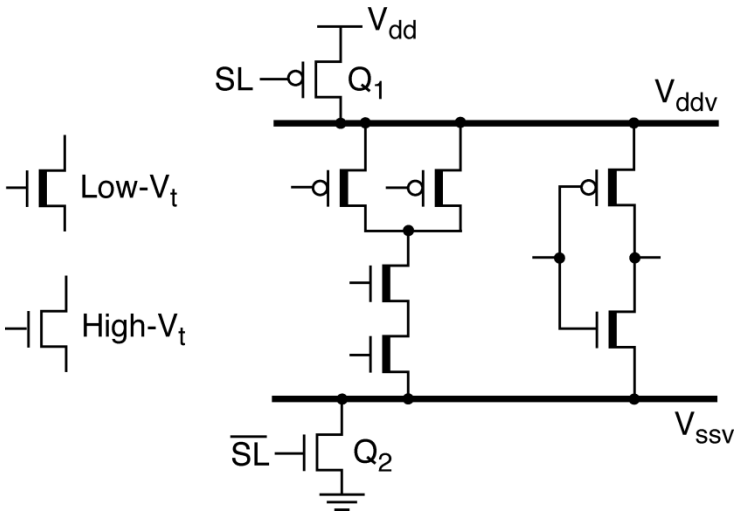


Figure 1.3 MTCMOS Power Gating implementation [7]

The authors of [8] demonstrate illustrate the implementation of both Clock and Power Gating for ISCAS '89 circuits and show how gating leads to significant dynamic and leakage power reductions.

1.2 Noise due to Clock and Power Gating

There are two forms of noise on the power lines, namely, IR drop from the RC elements of the power grid and the simultaneous switching noise (L*di/dt* drop) due to the parasitic inductances of the chip and packaging. The voltage drop across the grid may worsen if the noise input excites the grid's natural resonance frequency. This reduces the noise margins and in the worst case the circuit experiencing the noise may erroneously latch the wrong value or switch at the wrong time. Also, since the noise frequency is at or near the grid resonance frequency its effect will be more widespread and will last for a long time.

Although Clock Gating produces significant power savings, there are some penalties associated with the switching of large capacitances and currents during

transition in and out of gating which can excite the resonance frequencies of the power grid by interaction with the inductive parasitics of the grid leading to a large voltage drop near the location where the switching occurs. Early work in [9] showed the effect of such transient switching behavior due to clock gating on the power grid where gating caused large transients on the power line. Hence, it is imperative to study the effect of clock gating on the power grid noise.

Penalties associated with Power Gating occur during the wake-up process when a circuit transitions from standby to active mode and the current switches experience a large rush current. Due to the inductance from the power grid and the package, this rush current can cause *Ldi/dt* noise, which manifests itself as either ground bounce in the case where a footer is used or as V_{dd} fluctuation when a header is used. Another complication from power gating is the change in the impedance profile of the power grid due to change in a particular power gated block's status. This is because of the MTCMOS implementation shown in Figure 1.3. When the sleep transistors Q_1 and Q_2 are conducting (active mode) the real power lines (V_{dd} and V_{ss}) are connected to the parasitics of the circuits via the ON-resistances of the sleep transistors. When the sleep transistors are OFF (sleep mode) the real power lines only link to the OFF-resistances of Q_1 and Q_2 and the parasitics of the gated blocks are hidden from the view of the global grid. This is illustrated in Figure 1.4. Hence, we see that the global power grid impedance profile can change dynamically based on which circuit blocks that can be power gated are in what mode (standby or active?). This in turn can make characterization of noise in the grid difficult. In this work, we study and analyze of the aforementioned effect and find the voltage noise in the grid for a set of power-gated blocks separately.

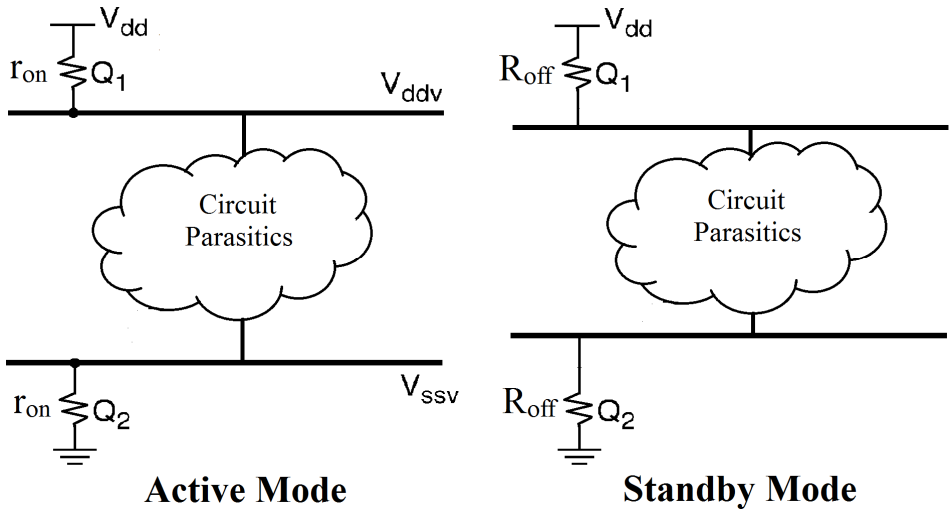


Figure 1.4 Impedance change of power grid due to Power Gating

1.3 Wavelet Analysis

As the power grid is represented in terms of RLC parasitic components, the grid represents a Linear Time Invariant (LTI) system. Wavelets can be used to accurately analyze the power grid as they provide a unifying framework for time-frequency decomposition of signals. They also have various other important applications in compression, adaptive filtering, signal detection, etc. In our work, wavelets are used to characterize the current sources attached to the power grid which will represent the loads. The wavelets that will compose the current source waveforms are constructed based on the impedance information of the power grid (frequency-domain) and hence, these wavelets can be used to synthesize current loads that will effectively target the resonant frequencies of the power grid in the time-domain enabling the analysis of the integrity of the power grid in the presence of worst-case noise. Modeling of the wavelets and construction of the *synthetic* current loads will be further discussed in Chapter 3.

1.4 Problem Statement

In this thesis, we analyze the effects of clock and power gating on the noise in a power grid. A set of gating-enabled blocks are considered to be attached to the power grid with each block consisting of a number of current sources/loads. We can model a LTI system consisting of the current loads as the input and their corresponding voltage response as the output. In this work, we construct the current load waveform using a set of wavelets by formulating a technique that uses a Linear Programming model to output the *synthetic* current loads and the worst case voltage noise due to the presence of these loads.

- We analyze the impedance profile of the power grid for either the 2D IC or the 3D IC case. From this profile, we construct a set of base wavelets that will be used in the construction of the current loads.
- For this work, we study the effect of Clock Gating on the noise in 2D and 3D power grids by specifying a target location on the grid where the noise is to be maximized at a target time. A Linear Program is formulated with the voltage responses of the wavelets, from each load location at the target location at the specified time, as inputs. A set of clock gating patterns for the gating-enabled blocks and final maximized voltage noise at the target location at the target time are produced as the outputs of the Program.
- We also the study of effect of Power Gating on voltage noise in 2D and 3D IC power grids using a similar wavelet technique. The problem cannot be solved with just a mathematical model. We make use of a Linear Programming model to generate an approximate solution and use Genetic

Algorithm to find a more optimal solution which yields us the power gating patterns and the maximum noise at a target location at the target time.

Thus, we will develop methodologies for effectively analyzing the integrity of a power grid in both 2D and 3D ICs in the presence of Clock and Power Gating.

1.5 Document Organization

The rest of this document is organized as follows:

- Chapter 2 deals with the necessary background information and prior work related to this work.
- Chapter 3 is dedicated to establishing the mathematical framework necessary to analyze the Clock Gating effect on noise in both 2D and 3D ICs.
- Chapter 4 discusses the experimental setup in detail and the results of the clock gating effect analysis for 2D and 3D ICs.
- Chapter 5 explains the difficulties of a pure mathematical model and details the heuristic search based model necessary to study the effect of Power Gating on noise in both 2D and 3D ICs.
- Chapter 6 describes the experimental setup and the results for the analysis of Power Gating noise effect on 2D and 3D IC power grids.
- Chapter 7 provides the conclusion for this work.

CHAPTER 2

BACKGROUND

We briefly discussed the concepts of clock gating and power gating in Chapter 1. In this chapter, we detail some background regarding 2D and 3D power distribution networks. Later we discuss the sources of noise in these networks and some of their effects on operation of an integrated circuit. We also introduce the concept of wavelet analysis and discuss the various types of wavelets that can be used. Later we describe, briefly, some of the prior work done using wavelet analysis and clock gating on power distribution networks.

2.1 Power Distribution Networks

Power Distribution Networks (PDNs) for high performance digital ICs are commonly structured as a multilayer grid. In the grid, straight power/ground (P/G) lines in each metal layer span the entire die (or a large functional block) and lines in adjacent metal layers are orthogonal to each other. Vias connect the adjacent orthogonal lines at the sites where they overlap. Figure 2.1 [10] shows a power grid concept where 3 layers of interconnects are depicted with power lines in dark grey and ground lines in light grey. Signal lines are placed between each adjacent P/G lines In each layer, the power and ground lines alternate to provide a decoupling effect to reduce coupling capacitance on the signal lines.

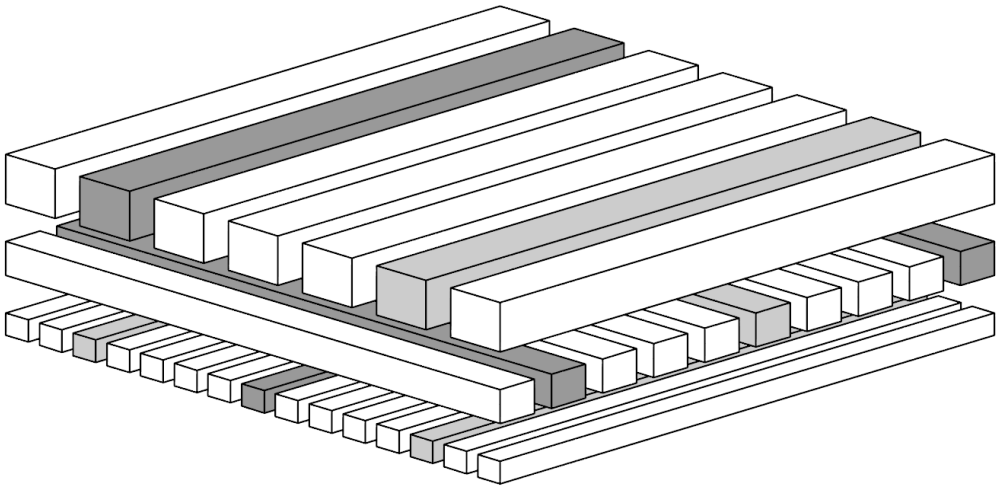


Figure 2.1 Multilayer grid structured Power Distribution Network [10]

To ensure the integrity of the power supply, a large fraction of on-chip metal resources are committed to creating the power grid. The PDN is usually determined early in the design process when there is little information about the specific power demands from each location of the chip. Allocating additional resources in later stages of design to ensure power integrity can create conflicts with other global signal lines and necessitate a prohibitively expensive redesign. Hence, PDNs tend to be conservatively designed [11].

Performance goals such as low impedance (to satisfy noise margins under high current loads), small area footprint and low current densities (greater reliability) are typically in conflict with each other. For example, widening the lines improves reliability and decreases impedance but, increases the area of the grid. But, replacing the lines with narrow interdigitated P/G lines can increase line resistance if area is maintained constant or increases area if the net cross section of the lines is kept constant.

2.2 Power Distribution Network for 3D ICs

3D integrated circuits have been proposed as one of the answers to maintain the continuation of Moore's Law and also to facilitate More-Than-Moore technology integration (heterogeneous technology integration) [1]. Various 3D Integration Technologies have been proposed [12]. Figure 2.2 shows some of them. Although some of these technologies have already been put into production, one of the most promising implementations on which extensive research is being done is using Through Silicon Vias (TSVs).

There are multiple ways to utilize TSVs for 3D integration which can be classified according to the nature of TSV fabrication (via-first or via-last), order of wafer stacking (Wafer to wafer, Die to Wafer and Die to die) and other classifications like the nature of bonding used to tie the different layers (Face-to-Face or Face-to-Back approach). All of these are discussed in detail in [1].

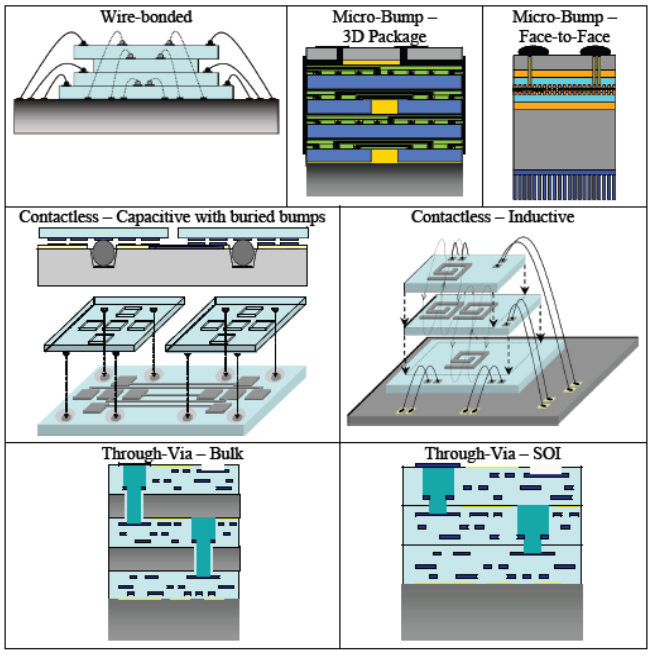


Figure 2.2 3D integration technologies [12]

A 3D power distribution network can be constructed using power/ground TSVs to connect the layers with each layer having their own local power grids. Figure 2.3 [13] shows a sample 4 tier (plane) structure with I/O pads.

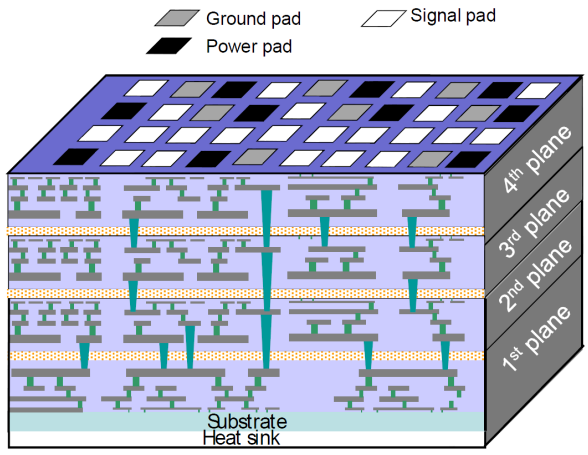


Figure 2.3 4 plane 3D IC with I/O pads in the topmost lane

TSVs provide a low impedance path between layers and hence, enhance current distribution. Some specifications for TSV dimensions are listed in Table 2.1.

Table 2.1 ITRS roadmap for TSV dimensions for global interconnects [1]

Global Level, W2W, D2W or D2D 3D-stacking	*2009-2012*	*2012-2015*
Minimum TSV diameter	4-8 µm	2-4µm
Minimum TSV pitch	8-16 µm	4-8 µm
Minimum TSV depth	20-50 µm	20-50 µm
Maximum TSV aspect ratio	5:1 – 10:1	10:1 – 20:1
Number of tiers	2-3	2-4

2.3 Noise in Power Distribution Networks

Increased switching speeds in VLSI circuits has led to increased probability that a large number of cells may switch at the same time causing considerable loading on the power grid and hence, inducing noise in the grid. The power grid may be modeled using

RLC parameters. In case only R is used to model the grid (resistive grid), we see that there is resistive noise which is also the IR (static) voltage drop in the grid. In the case of RL modeling, inductive components react to the change in current loads and introduce a Ldi/dt (dynamic) noise. If we include on-chip decoupling capacitances into the R or RL models then, they too affect the nature of the noise. Besides the on-chip contributors to noise, we also have the off-chip power delivery network (Voltage regulator, Motherboard and Package) that can contribute to noise in a significant way as there is a stark variation in the values when progressing from one component to the next. This leads reflections in the grid affecting the power supply noise. A model for the off-chip network is shown in Figure 4.1 in Chapter 4. Other sources of noise like thermal profile irregularities in the die, process variations, etc., also play an important role.

In the case of a 3D PDN, power/ground supply pad resources reduce to the order of 1/N compared to equivalent 2D implementation, where N is the number of tiers in the 3D PDN [14]. Also, since the TSVs are effectively large inductive elements their effect on noise becomes more prominent. There is also an increase in thermal noise due to difficulty in cooling which has become a major source of noise. Authors of [14] also discuss the frequency-dependent nature of the noise in 3D Ics in detail.

Excessive voltage drop due to the above sources cause glitches in the power supply lines and can lead to:

- Uncertainty in signal delays, which can affect clocking, and an increase in the delay along data paths limits the maximum frequency of operation for an integrated circuit.

- On-chip clock jitter also increases with noise and affect clocking. This also reduces the frequency of operation for an integrated circuit.
- Noise margins for the on-chip cells decreases which can affect their performance and produce an erroneous output. Also, there is an increase in crosstalk noise among the various signals.

2.4 Wavelet Analysis

Wavelet analysis is a powerful technique to decompose signals simultaneously in the time-frequency domain. The most common type of transform is the Fourier Transform (FT) which gives the frequency components of any arbitrary *stationary* signal. By *stationary* we mean that the frequency components do not change over the course of observation of the signal. But, Fourier Transform fails in case the signal frequency components change with time due the *non-stationary* nature of the signal. To analyze such signals and to decompose them, it is necessary to perform a transform that can track the time variations of the frequencies in a signal. This is where the Wavelet Transform comes in. To perform a wavelet transform we need a wavelet, which is also known as a localized waveform. Wavelets can be classified as either Continuous (Gaussian, Morlet) or Discrete (Haar, Daubechies). Some examples are shown in Figure 2.4 [15].

A continuous wavelet can be represented as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

where a is the wavelet scale and b is the translation factor and $\psi_{a,b}(t)$ is the wavelet.

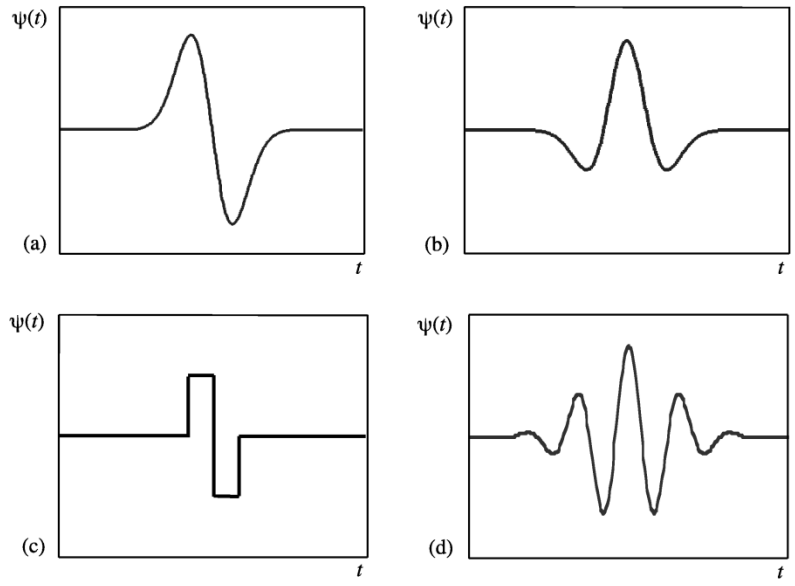


Figure 2.4 Different Types of wavelets are shown: (a) Gaussian, (b) Mexican Hat, (c) Haar and (d) Morlet

A Haar wavelet is one of the simplest wavelets and is defined for Figure 2.5(a) as:

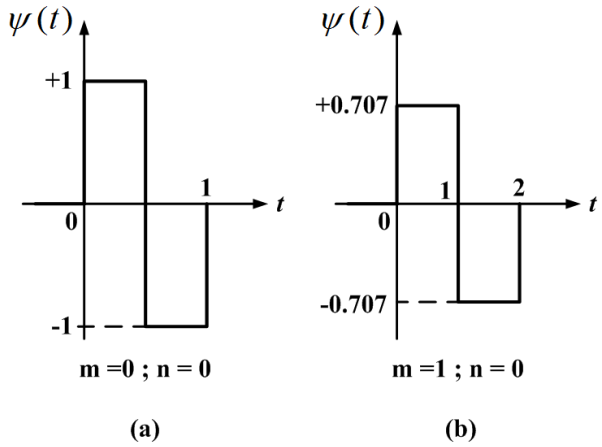


Figure 2.5 Haar wavelets

The discrete wavelet transform (DWT) can be written as:

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t - 2^m n}{2^m}\right) = A_m\psi(2^{-m}t - n) \quad (1)$$

where $a = 2^m$ and $b = 2^m n$ and $A_m = 2^{-m/2}$ is called the amplitude of the wavelet at scale *m*. In a DWT *m* and *n* are referred to as the scale and translation of a wavelet, respectively, even though the true scale and translation are *a* and *b*. The wavelets defined by (1) form a dyadic grid and these wavelets form an orthonormal basis that can be used to reconstruct any arbitrary signal *x(t)* as follows :

$$x(t) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} T_{m.n}\psi_{m,n}(t)$$

Unfortunately, such decomposition requires infinite number of wavelets to provide an exact signal representation. To reduce the size of the problem we can make use of Multi-Resolution Analysis (MRA) and decompose the signal using finite number of scales, *m*. An example of MRA using Haar wavelets is given in Figure 2.6 [16]. MRA relies on companion functions to wavelets called *Scaling Functions*, denoted by *φ(t)*, which can be scaled and translated in the same way as wavelets. From Figure 2.6, we can set frequency bounds (f_{min} and f_{max}) in which the signal needs to be analyzed and then, find the number of scales needed. Let there be m_0 scales. Scale m = 1 is associated with f_{max} and m=m_0 is associated with f_{min}.

Using such a formulation we can decompose the signal as:

$$x(t) = \mathrm{S}_{\mathrm{m}_0}\varphi_{\mathrm{m}_0}(\mathrm{t}) + \sum_{m=1}^{\mathrm{m}_0}\sum_{n=0}^{\mathrm{n}_\mathrm{m}} T_{m.n}\psi_{m,n}(t)$$

where $\mathrm{S}_{\mathrm{m}_0}$is the approximation coefficient and $T_{m,n}$ are detail coefficients. From Figure 2.6 m_0 = 3 and hence, we have 3 wavelets (m=1,2,3) and a scaling function, $\varphi_{\mathrm{m}_0}(\mathrm{t})$ is used to approximate the signals below f_{min}.

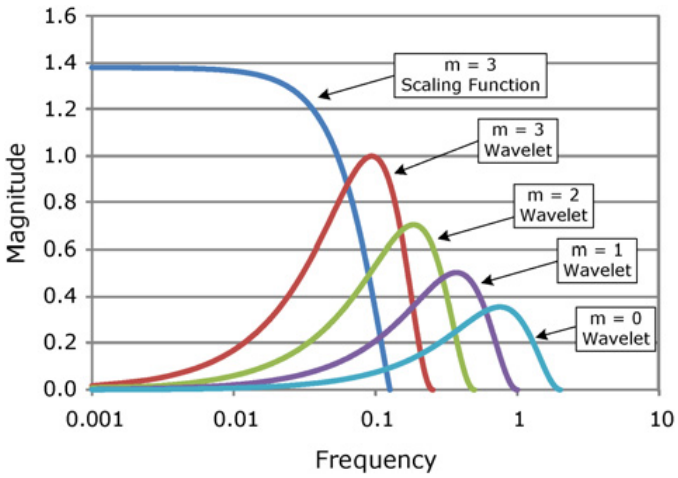


Figure 2.6 MRA using Haar wavelet [16]

2.5 Wavelet based technique applied to power delivery network

Imad A.Ferzli, et al. introduce the concept of time-frequency description of die current using wavelets [16]. Estimating worst case voltage drop on the PDN has both time and frequency dimensions and since wavelet allows us to capture time localized frequency information, a wavelet based framework has been built in their work to estimate the worst case current drawn for a given PDN. The authors use discrete wavelet transform to construct a current stimulus generated by switching of gates and then determine the worst case stimulus by formulating it as a Linear Programming problem. But the traces generated by them are not realistic as they do not consider the underlying logical dependencies in order to obtain the worst case.

Russ Joseph, et al. apply wavelet analysis technique to the problem of di/dt estimation and control in a modern microprocessor [17]. The authors use a wavelet based technique since the di/dt problem has natural frequency dependence and wavelets help us to capture the region of frequency bands in which there are large swings in current ripples. Thus, the authors propose an offline wavelet based estimation technique that can

accurately predict a benchmark's likely hood of causing voltage emergencies, and an on-line wavelet based control technique that uses key wavelet coefficients to predict and avert impending voltage emergencies.

2.6 Analysis of effect of Clock Gating on power supply noise

The effect of clock gating on the PDN and extraction of worst-case clock-gating patterns has been studied by W. Zhang, et al. [18]. The authors convert the current waveform into the frequency-domain using a Laplace Transform and then use a vector fitting method to obtain the voltage response in the time domain. Next, an algorithm is proposed which utilizes the time-domain voltage responses (defined over one cycle) corresponding to the unique clock domains. Using the principle of superposition on these responses, the algorithm outputs the worst-case clock patterns and the corresponding maximum voltage variation.

The work by W. Zhang, W. Yu, et al. [19] focuses on using Linear Programming for predicting a voltage violation and is built on the work presented in [18] but, extended to a 3D domain. The main focus is finding the amount of violation, the time duration for which the violation occurs and the gating patterns that resulted in the violation. The voltage responses are modeled using the same technique as in [18]. The authors propose a formulation of two Integer Linear Programming (ILP) models; the first model includes arbitrary leakage current sources and the second model represents the leakage current as a DC value.

2.7 Analysis of effect of Power Gating on power supply noise

H. Jiang, et al. [27] proposed a Genetic Algorithm-based approach to schedule power-gating to minimize supply noise. The authors develop a more accurate method to

estimate the worst-case noise by modeling the drop as a set of triangular waveforms. They then formulate the noise-aware scheduling problem and apply Genetic Algorithm-based optimization to obtain the solution. They extend the problem and develop an incremental scheduling procedure to find the optimal wake-up order of currently gated blocks considering the dynamic changes in the decapacitance configuration as active blocks provide additional decap to the power grid. Theirs is the first paper that addresses the scheduling problem for power-gating with respect to noise minimization.

CHAPTER 3

CLOCK GATING PROBLEM FORMULATION

In this chapter, we cover the key mathematical constructs needed and their use in this work with regards to Clock Gating. Wavelet analysis is useful for capturing the temporal variation of voltage drop by utilizing the frequency-domain information of the current loads and the power grid itself (impedance). First, we explain the determination of various wavelet parameters necessary for the final problem formulation. Next, the preliminaries for the LP formulation are explained and the process of including the clock gating information is shown. Later, we discuss the general design flow that will be employed for noise analysis of a 2D power grid. Then, we formulate the problem in terms of a linear programming model. Finally, the changes required for the extension of the analysis to a 3D power grid are explained.

The Problem Statement for the analysis of effect of clock gating on the noise at a particular point on the power grid is stated as:

- Maximize the voltage drop at a particular node of interest ('z') on the Power Distribution Network at time 't_0' in the presence of 'q' gating enabled current loads.
- Find the clock gating patterns for each of the 'q' current loads.

3.1 Construction of a wavelet

In the previous chapter, we discussed some basic properties of wavelets and the concept of discrete Haar wavelets. Each Haar wavelet is centered around a central frequency, f_c, which is defined using (2) [15].

$$f_c = \frac{2.33}{(\pi 2^m)} \tag{2}$$

where m is the wavelet scale as defined in (1).

The number of wavelets to use can be found from the impedance profile of the power grid where we define a frequency region of interest. Using the lower and upper frequency bounds (f_{min} and f_{max}, respectively) the value of m_0 (the largest scale which also is the total number of scales) is obtained. The value of m_0 is ceiled [16] and the wavelets are constructed using m = 1, 2,..., m_0. m = 1 is used to represent the shortest (or fastest) wavelet while m = m_0 represents the longest wavelet (or slowest).

$$m_0 = \left\lceil log_2 \left[\frac{2f_{max}}{f_{min}} \right] \right\rceil \tag{3}$$

Let u be the time over which a wavelet value is constant, which represents the time unit in our analysis. 'u' can be written as:

$$u = \frac{2.33}{2\pi f_{max}} = \frac{a_{min}}{2} \tag{4}$$

where a_{min} represents the scale of the smallest wavelet (m = 1).

The set of Haar wavelets obtained can be considered as bandpass filters (seen in frequency domain) [15] and provide a finer resolution for the analysis of the power grid. This form of Multi-Resolution Analysis (MRA) allows us to characterize the frequency response of a power grid using multiple wavelets. Since we assume our current loads have Piecewise Linear waveforms, we can construct these loads using a set of basis wavelets obtained by using (3) and (4). A current load consisting of a set of wavelets is given by (5) [16].

$$i(t) = i_{dc} + \sum_{m=1}^{m_0} \sum_{n=1}^{n_m} T_{m,n} \psi_{m,n}(t) \quad (5)$$

where i_{dc} is used as an approximation that accounts for the current stimuli from all the frequencies below f_{min} and $T_{m,n}$ represents the weights or detail coefficients [16] that are used to construct the required waveform $i(t)$ around an offset of i_{dc}, $\psi_{m,n}(t)$ represents the wavelet of m^{th} scale and n_m is the number of the wavelets at the m^{th} scale.

3.2 Incorporation of Clock Gating

Since the effect of clock-gating on the PDN has to be analyzed, we need to include a gating variable when constructing the synthetic current load. We can modify (5) by introducing a gating term, α_p, and the resulting current equation is given by (6).

$$i_j(t) = i_{min,j} + \sum_{p=1}^{t_0/u} (1 - \alpha_{p,j}) \sum_{m=1}^{m_0} \sum_{n=1}^{n_m} T_{m,n,j} \psi_{m.n}(t) \quad (6)$$

where $i_j(t)$ represents the j^{th} current source, t_0 is the time at which the voltage drop needs to be maximized, p is an index used to represent the time windows which are of unit size u and the number of such windows in which the current source can either be gated or non-gated is given by t_0/u (t_0 is assumed to be a multiple of u for simplicity), $\alpha_{p,j}$ represents a binary variable attached to the j^{th} current source and is explained in (7).

$$\alpha_{p,j} = \begin{cases} 1 & unit\ is\ gated \\ 0 & unit\ is\ not\ gated \end{cases} \quad (7)$$

Lastly, $i_{min,j}$ represents the leakage current when the j^{th} module that is assigned the current load $i_j(t)$ is clock-gated.

3.3 Design Flow

The general steps in the design flow used in this work are listed as follows:

3.3.1 Obtain the Impedance Response

Once the power grid is constructed, we find its impedance profile. This profile will allow us to set the frequency range of interest (f_{min} and f_{max}) and thus, construct the set of base wavelets for m = 1,..., m_0.

3.3.2 Calculate the wavelet parameters

We can specify the frequency region of interest in the impedance profile of the grid, based on our problem requirements, and calculate m_0 and u from (3) and (4). We can also set the target time, t_0, which should be equal to or greater than the time-span of the slowest wavelet (m= m_0). From these parameters we generate the set of basis wavelets that will be used for further analysis.

3.3.3 Generate the voltage response at point of interest

The power grid is loaded with ideal current loads, the number of which can be specified by the designer. For each of the j current loads, we input each one of the basis wavelets, $\psi_{m,n}(t)$, (given by m) and shift them backwards from time t_0 (n represents the number of backward shifts). Next, we find the voltage response from the j^{th} current load at our point of interest, z, at the target time, t_0, and this is represented as $h_{m,n,j,z}(t_0)$. A set of these responses is used to represent the complete voltage response of the j^{th} current load at z for all the wavelets. Figure 3.1 shows an example of a wavelet shifted backwards from time t_0 and its corresponding response at z.

3.3.4 Solve the Linear Programming (LP) model

The set of responses from the previous step for each of the current sources can be optimized as some of the time shifted wavelets produce a zero voltage drop at the point of

interest (eg. response from input $\psi_{1,4}$ at node z, $h_{1,4,j,z}(t_0)$, in Figure 3.1 is zero) and these inputs can be discarded to speed up the LP solution. Also, at this stage, we input the relevant data in the form of constraints and parameters like the leakage current values for each of the loads, the power constraints which decide the maximum current that is drawn by a load and assign each current load to their respective clock gating-enabled regions. Additional information like specific timing requirements of certain current loads can be provided in the form of more detailed constraints to make the overall formulation more realistic.

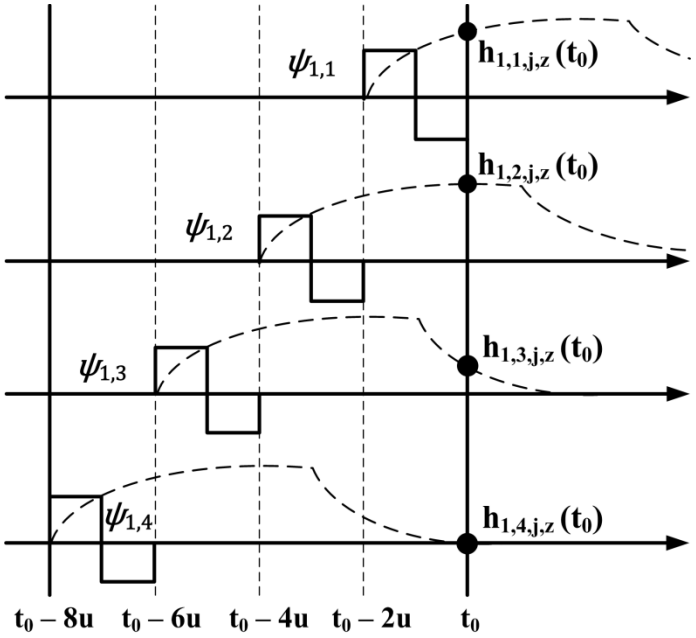


Figure 3.1 Shifted wavelets and their response at z

3.4 Linear Programming model formulation

In this section, the Objective Function and the constraints for the LP formulation are described. Given this formulation, we can maximize the voltage drop at a particular point, z.

3.4.1 Objective Function

Given the voltage drop responses, the objective function calculates the maximum voltage drop at the target location at a specific time, t_0. Let $v_{z,j}(t_0)$ be the voltage drop at point z due to the current load at j at time t_0. The drop is given by [16]:

$$v_{j,z}(t_0) = d_j i_{min,j} + \sum_{m=1}^{m_0} \sum_{n=1}^{n_{m,j}} h_{m,n,j,z}(t_0) T_{m,n,j} \tag{8}$$

where, d_j is the voltage drop at z when a current load of 1A is present at node j and $h_{m,n,j,z}(t_0)$ is the voltage response from a current load with stimulus $\psi_{m,n}(t)$.

The objective function is given by the linear superposition of all the voltage drop responses from all current stimuli, considering there are q stimuli in total. This is represented in (9).

Maximize:

$$v_z(t_0) = \sum_{j=1}^{q} v_{j,z}(t_0) \tag{9}$$

3.4.2 Constraint Generation

The effect of clock-gating can be mathematically encoded in the form of constraints as explained earlier. But, from (6) we see that the term $\alpha_{p,j} T_{m,n,j}$ is non-linear, as $\alpha_{p,j}$ and $T_{m,n,j}$ are both variables for the LP formulation. Considering the fact that $\alpha_{p,j}$ is a binary variable, we can linearize the non-linear term by defining (6) in the form of two separate constraints. These set of constraints are listed in (10).

$$0 \leq \sum_{m=1}^{m_0} T_{m,n,j} \psi_{m,n}(t)$$

$$\sum_{m=1}^{m_0} T_{m,n,j}\psi_{m,n}(t) \leq \left(1-\alpha_{p,j}\right)\left(i_{max,j}-i_{min,j}\right) \quad (10)$$

where, $\psi_{m,n,j}(t)$ can be either A_m or $-A_m$ or 0 (in case the wavelet is not present in the time slot, $A_m = 2^{-m/2}$), $i_{max,j}$ is the maximum current that the particular load consumes. Equation (10) provides the bounds for the wavelets that are part of the current load and specifies whether the current load is gated in a particular time slot, p.

The above constraints consider all the current loads to be under different clock-gated regions. But, we can group the constraints for current loads and assign the same α_p to them and hence, create the unique gated regions.

3.5 Extension to 3D PDN

In the case of analyzing the effect of clock gating on a 3D power grid, most of the formulation and design flow explained previously remains unchanged. We introduce another parameter, l, which will be used to represent the tier information on the power grid. So, modifying (6), we get (11) as stated below.

$$i_{j,l}(t) = i_{min,j,l} + \sum_{p=1}^{t_0/u}(1-\alpha_{p,j,l})\sum_{m=1}^{m_0}\sum_{n=1}^{n_m} T_{m,n,j,l}\psi_{m.n}(t) \quad (11)$$

We redefine q (total number of current sources) as q_l, total number of current loads in a tier, l. Each current load produces a set of voltage responses, $h_{m,n,j,l,z}(t_0)$, and the final voltage drop from (8) is redefined as (12). The various parameters represent the j^{th} current source in tier l. z is the Point of Interest and can be present in any of the tiers.

$$v_{j,l,z}(t_0) = d_{j,l}i_{min,j,l} + \sum_{m=1}^{m_0}\sum_{n=1}^{n_{m,j}} h_{m,n,j,l,z}(t_0)T_{m,n,j,l} \quad (12)$$

The Objective Function now becomes:

Maximize:

$$v_z(t_0) = \sum_{l=1}^{L} \sum_{j=1}^{q_l} v_{j,l,z}(t_0) \tag{13}$$

where *L* is the total number of tiers in the 3D PDN.

The current constraints are also modified to represent the 3D nature and are listed in (14).

$$0 \leq \sum_{m=1}^{m_0} T_{m,n,j,l} \psi_{m,n}(t)$$

$$\sum_{m=1}^{m_0} T_{m,n,j,l} \psi_{m,n}(t) \leq \left(1 - \alpha_{p,j,l}\right)\left(i_{max,j,l} - i_{min,j,l}\right) \tag{14}$$

CHAPTER 4

CLOCK GATING EXPERIMENTAL SETUP AND RESULTS

This chapter explains the experimental setup and results obtained for the analysis of the effect of clock gating on noise at a particular point on the power grid, for both a 2D and a 3D power grid.

4.1 Specifications for the 2D power grid

For the external power supply parasitics (from off-chip VRM to chip via motherboard and package), we use the Nehalem impedance matched model as shown in Figure 4.1 [20].

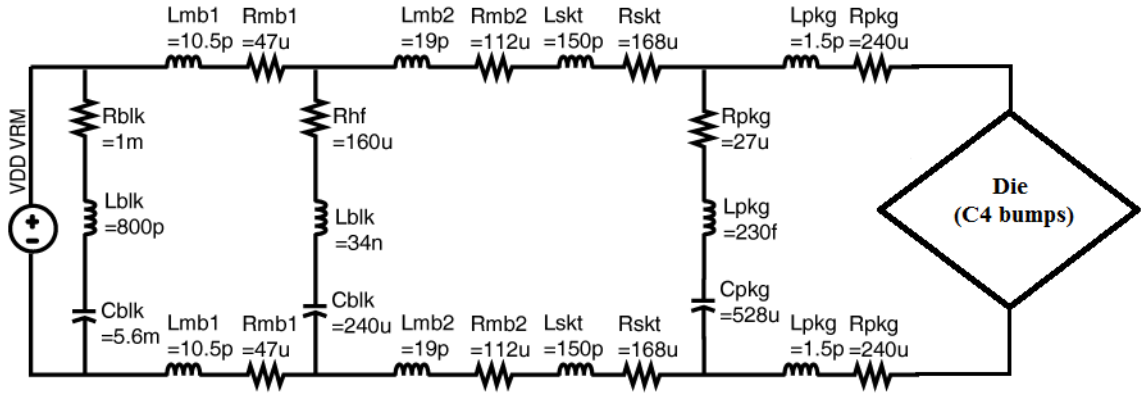


Figure 4.1 Power Delivery Model for Nehalem impedance profile

The die is connected to the external power delivery network via micro-C4 bumps each of which are represented by a resistance in series with an inductance and whose parasitics are listed as follows [21] :

- Resistance =40 mΩ
- Inductance = 70 pH

The on-chip power grid is constructed using M2 and M3 metal layers whose specifications are obtained from Interconnect Technology file in the NCSU PDK for 45nm [22]. We assume an inter-digitated power supply line model (alternating V_{dd} and V_{ss} lines in each metal layer) and infinite vias between layers forming ideal shorts between the lines. A unit cell of the power grid model, as constructed in Ansys Q3D extractor tool [23], along with the metal widths and pitches is shown in Figure 4.2.

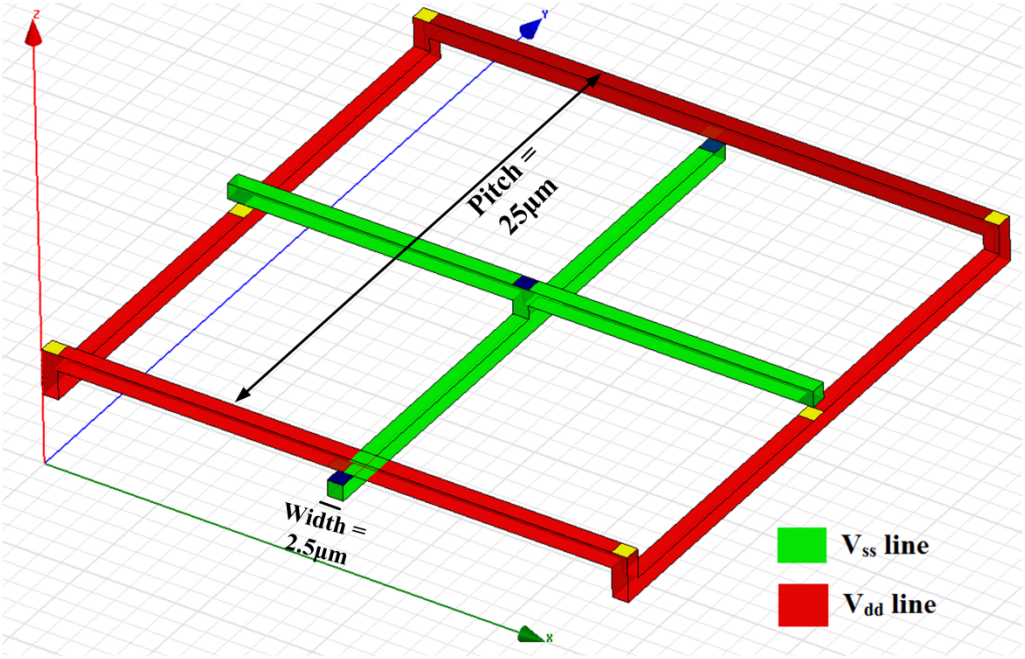


Figure 4.2 Unit cell of power grid

Using Ansys Q3D extractor tool [23], we find the equivalent RLC parasitics which are shown in Figure 4.3. This unit cell is used to construct the V_{dd} part of the power grid by instantiating the cell multiple times. For our work, we incorporate the parasitics of the ground grid into the power grid and assume ideal ground.

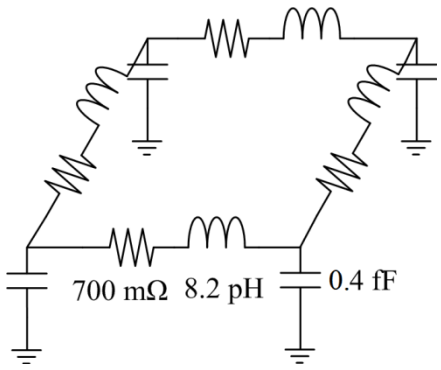


Figure 4.3 Unit cell parasitics

A die size of 1mm X 1mm is considered and the power grid is constructed for this die with the above specifications. Hence, we get a 40 X 40 nodes power grid where a node is one of the corners of unit cell from Figure 4.3. The C4 bumps are placed at a pitch of 100µm. Considering one of the corners of the grid, we run AC simulations using HSPICE to find the impedance profile of the power grid. The normalized impedance profile of the 2D power grid is shown in Figure 4.4. Due to the small die size we only get a single peak.

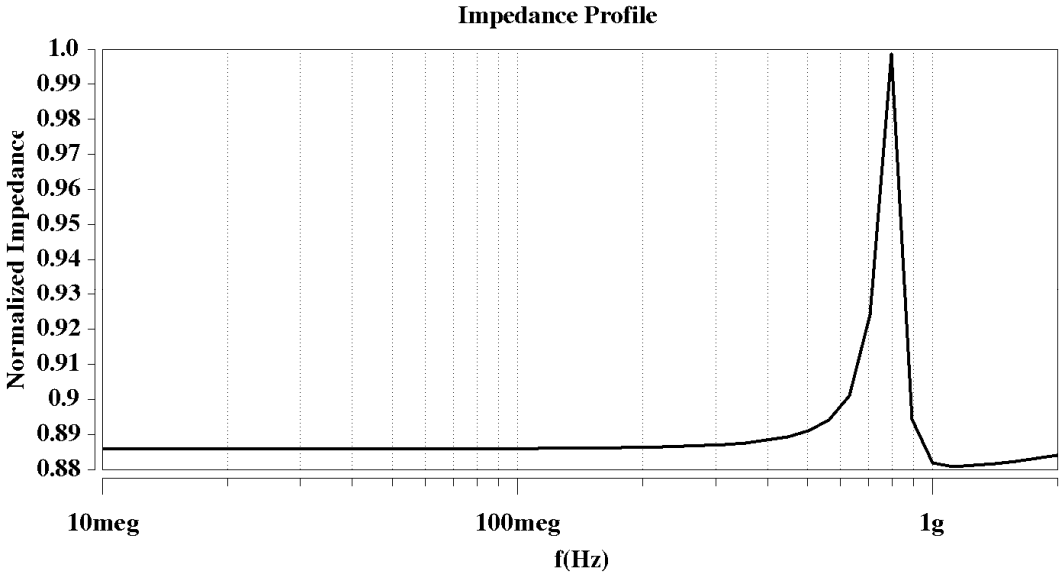


Figure 4.4 Normalized Impedance profile of 2D power grid

From Figure 4.4 we set the frequency range of interest as:

- f_{min} = 200 MHz
- f_{max} = 1 GHz

from which we get m_0 using (3) and u using (4) as :

- $m_0 = 4$
- u = 0.37 ns

We set the target time at which the voltage noise is to be maximized as:

- t_0 = 5.92 ns
- This gives us a total of t_0/u = 16 time slots.

For creating a clock gating-enabled block, we first consider a unit circuit - AES fast encryption block obtained from [24]. We synthesize the AES block in 45nm using Synopsys Design Compiler and Nangate Open Cell Library. Power analysis is done using VCS and Primetime-PX (both from Synopsys). Some specifications of this block are listed below.

- AES block has 14090 gates.
- Total Area = 16419 μm^2.
- Area in terms of nodes ~ 5 X 5 nodes on the power grid.
- Peak power = 80 mW
- Leakage power = 0.3 mW

Considering a nominal supply voltage of 1.1V, we get the current bounds for the AES block as :

- i_{min} = 0.27 mA
- i_{max} = 72.727 mA

For the purposes of this analysis, the AES block is represented by a current source at the center of the block with the given area. The current source is connected to the power grid at a single node. To create the unique clock-gated regions, the unit AES block is instantiated multiple times and spread arbitrarily into 10 arbitrary clock-gated regions. These regions are illustrated in Figure 4.5. Each region is assigned an α and all the current loads within that region are controlled by their respective α. The i_{min} and i_{max} of each region is the number of instances of unit block multiplied by the respective values of the minimum and maximum currents of the unit block. A total of 39 such instances are spread between the 10 regions which gives a total power budget of 3.2W for the grid (this is below the 4-8W/mm^2 power budget generally considered to be the maximum [25]). One of the grid corners which is farthest from all the loads is taken as the point of interest, z.

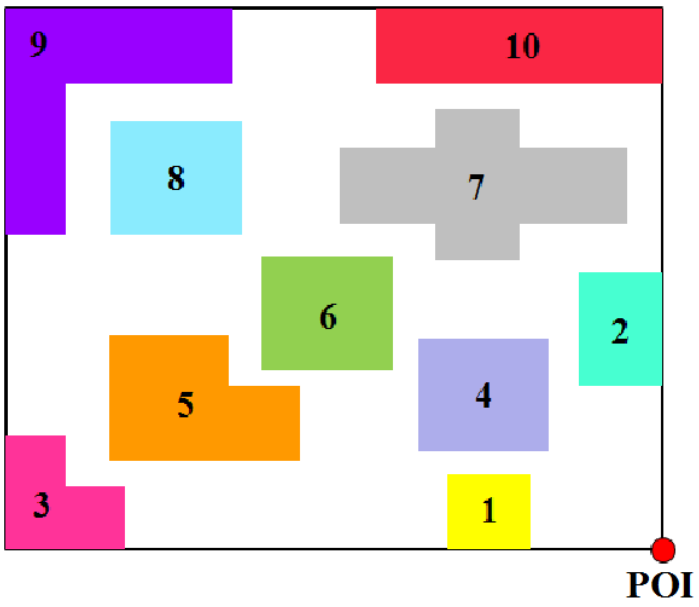


Figure 4.5 Clock Gated regions on the power grid

4.2 Results for the 2D power grid

Considering the above specifications, the LP formulation is solved using a solver like GLPK [26] which yields the gating pattern (vector α) for the 10 uniquely clock-gated domains shown in Table 4.1. Table 4.2 lists the voltage drop obtained at the point of interest (POI) at time t_0 using the LP. Using the values of currents in each time-slot obtained from the LP we run SPICE simulations and find the maximum voltage drop at the same POI. This is also listed in Table 4.2. The discrepancy between the two values of voltage drop obtained can be explained due to certain inherent inaccuracies of SPICE model and due to the Least Significant Bit approximations made in the specifications of the various components. Figure 4.6 shows the voltage drop waveform at the point of interest, z, from all the current sources at time t_0=5.92ns as obtained from SPICE. From Figure 4.6 we see that the frequency of the noise waveform is equal to ≈ 770 MHz which is very close to the peak frequency from the impedance profile shown in Figure 4.4.

Table 4.1 Worst-Case Clock Gating Patterns

Region	Pattern
1	0011001101100110
2	0011001100110110
3	0000000100010000
4	0011001101100110
5	0000000100010000
6	0011001101100110
7	0000000101000000
8	0011001101100110
9	0000000101000000
10	0011001100110110

Table 4.2 Voltage drop at Point of Interest (z)

	Voltage Drop (mV)
With LP	176
From SPICE	164

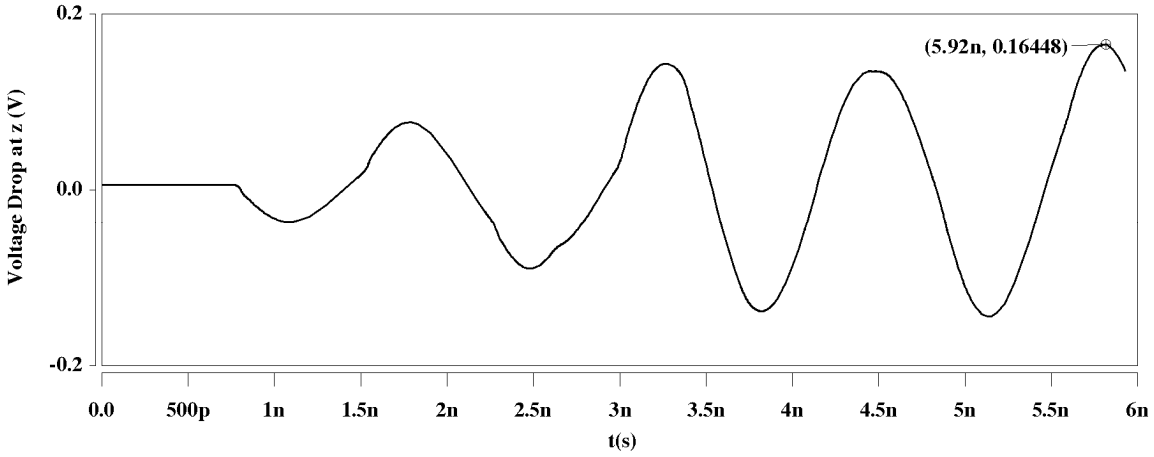


Figure 4.6 Voltage drop waveform at point of interest, z

To further verify that the obtained gating patterns are the worst, we consider 10,000 sets of random patterns. Each set consists of 10 16-bit random patterns which are assigned to the current sources in the uniquely gated domains and this used as the basis to run SPICE simulations to obtain the maximum voltage drop at z any time during the time period 0 to t_0. Fig. 4.7 shows the plot of the voltage drops along with the drop obtained from our methodology for comparison. We see that the LP formulation produced voltage drop is much larger than the ones obtained from the random sets and this drop is produced at the specified time t_0. A larger number of random patterns may yield a better solution but, would require significantly more time and resources.

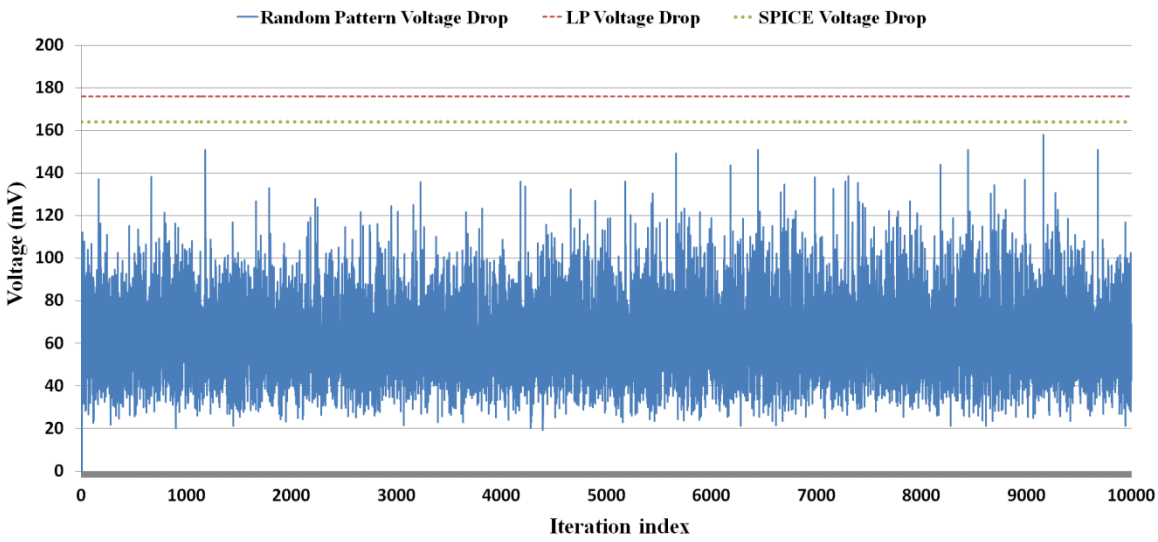


Figure 4.7 Voltage Drop Comparison

4.3 Specifications for the 3D power grid

In case of the 3D power grid, we retain the same external power supply parasitics as shown in Figure 4.1 and construct a power grid with 3 tiers. Also, we consider the same unit V_{dd} cell shown in Figure 4.3 to construct the power grids in each of the tiers. A representation of the 3 tiers used is shown in Figure 4.8. Tier 1 is the topmost tier (i.e., connected to the package via the C4 bumps), and Tiers 2 and 3 are the bottom tiers connected to the previous tiers using Through Silicon Vias (TSVs).

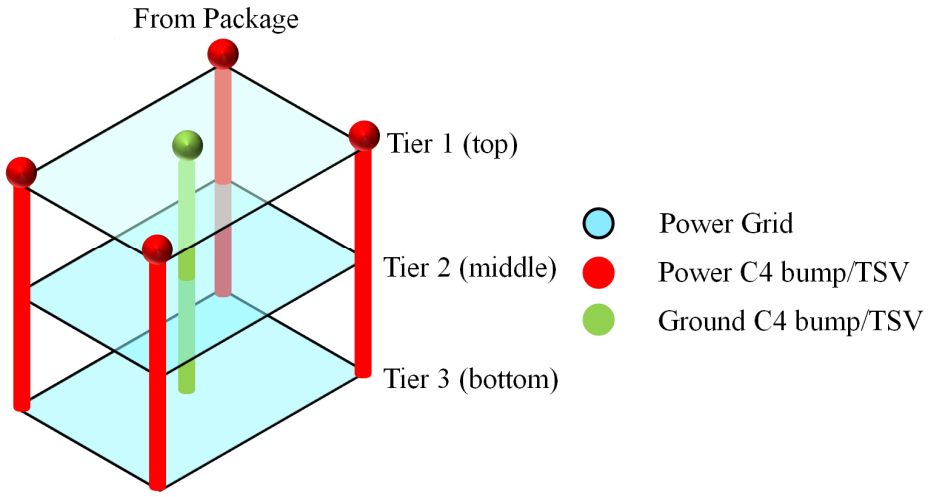


Figure 4.8 Representation of the 3D PDN

The TSV is constructed using Copper fill with a Silicon Dioxide dielectric coating surrounded by a low-κ Silicon substrate. The various parameters are listed below.

- TSV diameter = 10 µm
- TSV height = 50 µm (thickness of each tier)
- TSV pitch = 100 µm (aligned with C4 bumps)
- Dielectric coating thickness = 0.2 µm
- Silicon substrate conductivity = 100 kS/m (high conductivity)

The TSV parasitics are extracted using Ansys Q3D extractor assuming a center frequency of 1GHz. The RLC model for the TSV consists of series RL with two ground capacitances connected to either tier [31], as each tier is simulated as a ground plane, since we only consider a power TSV. Figure 4.9 shows the TSV as created in the tool and the parasitics are listed below:

- R_{TSV} = 20 mΩ
- L_{TSV} = 20 pH
- C_{TSV} = 25 fF

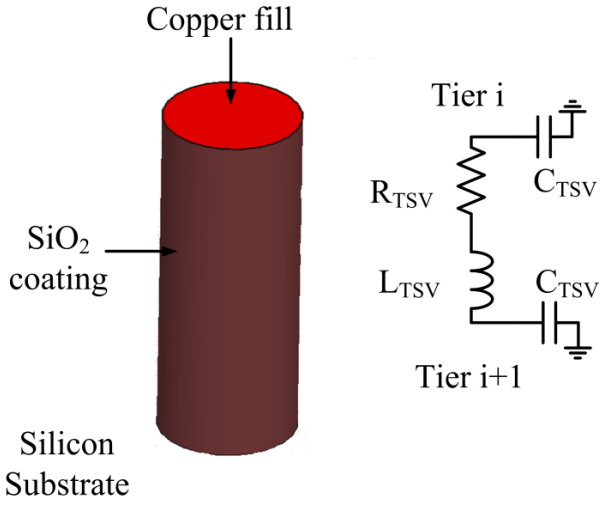


Figure 4.9 TSV parasitic representation

The power grid of each tier in the 3D PDN is of the size 23 X 23 nodes. Each of the 3D PDN's 3 tiers is assumed to have 4 arbitrary (but, symmetric across tiers) clock gating-enabled regions as shown in Figure 4.10. These 4 regions in turn are created using 13 instances of unit AES blocks described in the previous sections, leading to a total of 39 instances. Such a construction is done to facilitate comparison with the previous 2D PDN design.

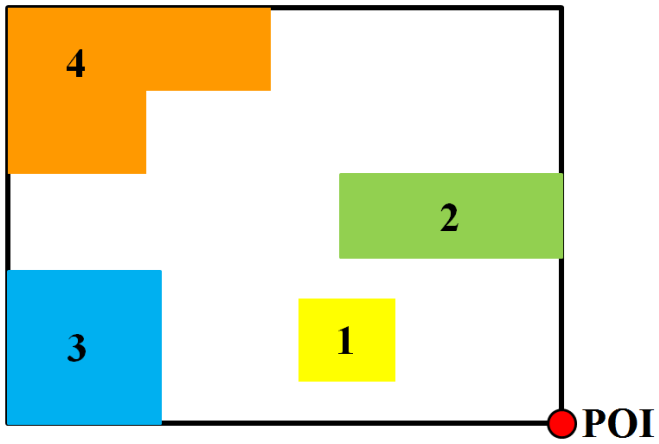


Figure 4.10 Top view of clock gated regions in a tier

For extraction of the impedance profile we choose a corner of each tier and run AC simulations using HSPICE. The impedance profile of the 3D PDN is shown in Figure 4.11. We notice there is an additional peak for Tier 3 at 2.7GHz in the 3D case and hence, our frequency bounds must be changed with respect to the 2D case. Also, we see that the first peak value increases and that the impedance rises after the first peak more prominently, with advancing tiers (further away from the package). We do not consider the profile beyond 3 GHz as we assume that the dominant excitation sources in our problem do not exceed this limit.

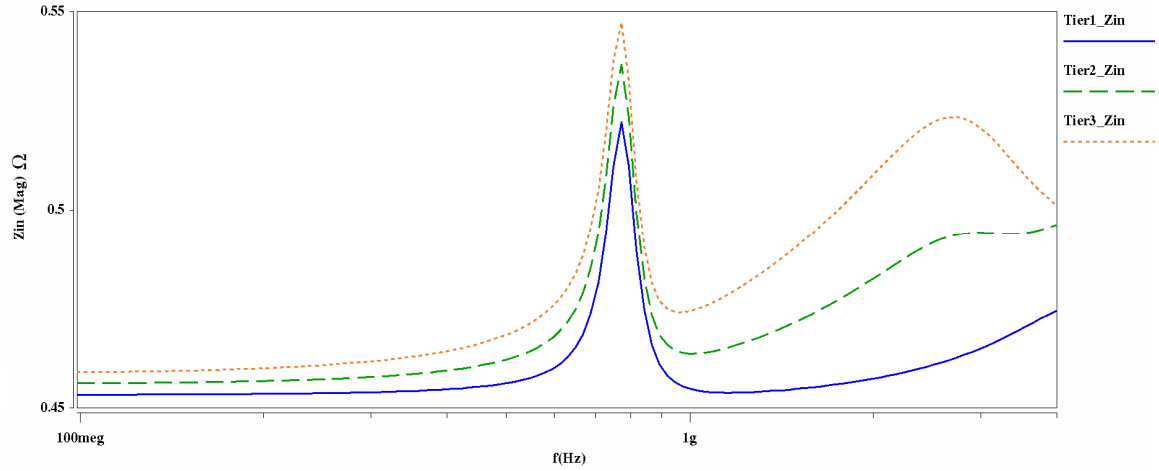


Figure 4.11 3D PDN Impedance Profile

From the impedance profile obtained, we set the frequency bounds and obtain the wavelet parameters as listed below :

- f_{min} = 270 MHz
- f_{max} = 2.7 GHz

from which we get m_0 using (3) and u using (4) as :

- $m_0 = 4$
- u = 0.13 ns

We set the target time at which the voltage noise is to be maximized as :

- t_0 = 2.08 ns
- This gives us a total of t_0/u = 16 time slots.

4.4 Results for the 3D PDN

To study the nature of the voltage drop across the tiers, we deploy the 3D LP formulation by considering a point of interest (POI) as shown in Figure 4.10 for each of the tiers with a target time t_0. The values of the voltage drop are tabulated in Table 4.3. Consistent with the nature of a 3D PDN we see that the voltage drop increases as the tiers

become located further away from the package power supply. The larger difference between Tiers 1 and 2 compared to the difference between Tiers 2 and 3 is evident from the impedance profile where Tier 2 and Tier 3 impedances increase more rapidly after the first peak than in the case of Tier 1. All the voltage drop values are higher than the drop obtained for 2D PDN confirming that the limited power supply resources (C4 pins) increases the overall voltage drop.

Table 4.3 Voltage drop values across the tiers

Tier #	Voltage Drop (mV)
1	204
2	221
3	234

Next, we extract the worst-case gating patterns from each tier when the POI is set on Tier 3. The set of gating patterns are tabulated in Table 4.4. The same process can be used to extract the gating pattern across the tiers for any point of interest.

Table 4.4 Clock Gating Patterns for POI in Tier 3

Region #	Tier 1	Tier 2	Tier 3
1	0010100000111100	0010100100011000	0000101010111100
2	0000101110011000	0010101000111100	0010101000111100
3	0000101000011100	0010100000111100	0010100000010100
4	0001010000000000	0001010000010000	0101000000010000

CHAPTER 5

POWER GATING PROBLEM FORMULATION

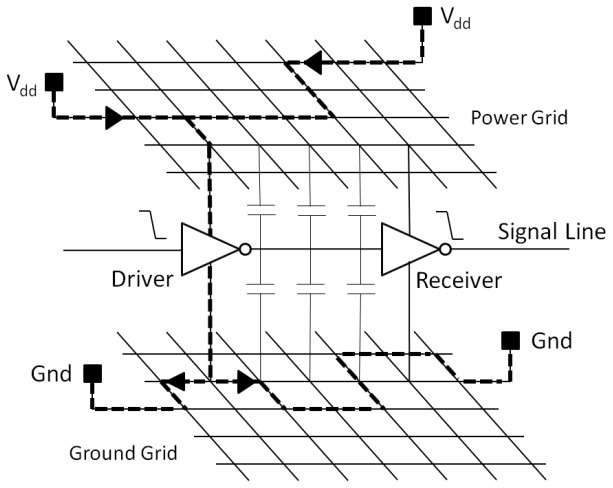
In this chapter, we explore the key issues with power gating that differentiate it from clock gating and describe the mathematical and simulation constructs needed to analyze the effect of power gating on supply noise. First we describe why Linear Programming model used in the earlier sections is no longer feasible for power gating. But, wavelet analysis with a Linear Programming model is still useful towards finding the final solution by providing an initial solution to drive the next phase of analysis. The next phase utilizes Genetic Algorithm and SPICE simulation based programming to find the final solution. The mathematical basics of Wavelet Analysis have been covered in Section 3.1. The rest of the problem formulation in terms of genetic programming is discussed next for both 2D and 3D power grids. Lastly, we discuss the general design flow that will be employed for noise analysis of either 2D or 3D power grids.

5.1 Issues with modeling the effect of power gating

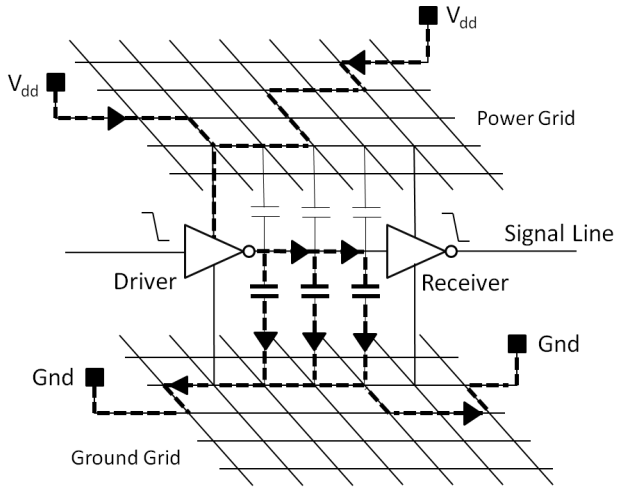
As discussed in Section 1.2, there is a high degree of non-linearity associated with power-gating with respect to supply noise as the changes in the grid impedance depend on the state of the power-gated blocks. Construction of a detailed mathematical model would involve knowing and capturing all the parameters of the power grid for the non-linear optimization problem.

One of the important parameters is the loop inductance in a power grid which varies with the varying current paths in the circuit and can affect the measurement

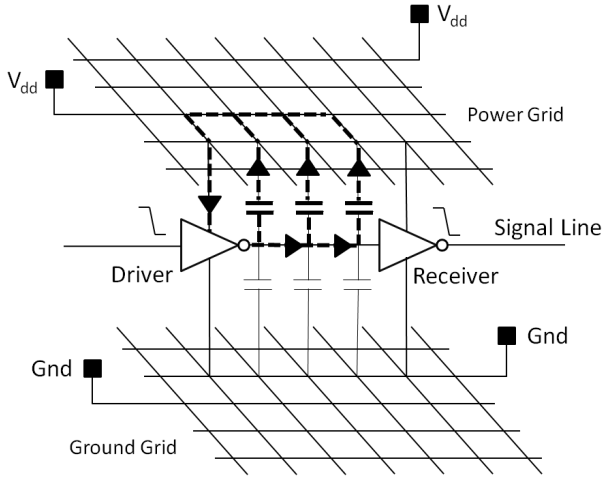
voltage drop across the power grid. Figure 5.1 [35] illustrates the various current paths between the power lines through the devices, interconnects and the ground lines that can be seen for a set of buffers on a signal line during various modes of operation of the driver. In the case of clock-gating where the grid impedance profile remains static, it is possible to obtain the relevant data from SPICE simulations for the Linear Programming model within a reasonable amount of error for an RLC model based power grid even though the loop inductance may not have been captured. Capturing loop inductance completely requires running detailed Electromagnetic (EM) simulations for all possible paths between voltage/ ground pins and loads on the power grid. This can be a resource intensive process for large designs. Dedicated commercial power analysis tool and more complex power grid models can allow us to capture the effects of loop inductance for real world applications with a reasonable investment of resources. In this work, we only model a unit power grid cell and capture its parasitics. Loop inductance for the whole power grid is not modeled. In conclusion, loop inductance provides one source of non-linearity in a power grid which affects the accuracy of any dynamic power analysis.



(a)



(b)



(c)

Figure 5.1 Current paths in Driver-Receiver-Grip topology [35]
(a) Short circuit current while gate is switching
(b) Charging current from V_{dd} to ground
(c) Discharging current via interconnect and gate capacitances

For the case of power-gating, the sleep transistors act as gates that connect/ disconnect the local power grids for the circuits to the global power grid which connects to the power supply pins. The current paths are now determined not only by the gates that are switching within the various gated and non-gated blocks but, also on the state of the

power-gated blocks in real-time. Hence, characterizing the grid for wavelet analysis by attaching various current loads, constructed using the wavelet parameters, and measuring the voltage drops at the point of interest, z, at target time t_0 becomes difficult. The purpose of the mathematical model is to find the power gating pattern to maximize noise at z at t_0 but, the model itself needs data to characterize all possible states of the power-gated blocks. This interdependency can introduce a large source of error. The mathematical model fails if we use a lower order objective function for optimization and solving a higher order function is hard. The sleep transistors constitute the second source of non-linearity in a power grid for the power-gating analysis.

So, the problem statement for analyzing the effect of power gating on power supply distribution noise can be summarized as:

Maximize the voltage drop at a point of interest, z, on the power grid at target time t_0 given

- Presence of power-gated and non-gated blocks (acting as loads) on the power grid
- Non-linearity of sleep transistors modeled using SPICE
- Loop inductance effect not modeled for a power grid. Only static RLC parasitics considered for SPICE simulations

In this work, we propose generating an approximate solution using a simplified Linear Programming model and using the solution to drive a heuristic search algorithm where the relevant data is generated with real-time SPICE simulations of the power grid. We choose Genetic Algorithm based heuristic search due to its flexibility. Section 5.3 discusses the relevant background for Genetic Algorithm.

For the initial solution using Linear Programming (LP) model, we assume that the power-gated blocks and their local grids are always connected to the global grid (Sleep transistors are always ON). This makes the problem comparable to the clock-gating case and we can use a similar Linear Programming model to generate a set of gating patterns for all the gated and non-gated blocks present on the grid. The main advantage of doing this is that it allows the Genetic Algorithm (GA) to focus its search on a region where the final optimal solution may be present and allows for faster convergence compared to a search with a random starting point. The results in Chapter 6 tabulate the non-optimal and optimal solutions from LP model and GA search and also illustrate the runtime improvement due to use of approximate solution to seed the GA search.

5.2 ILP formulation

Constraints Generation: In our work, we make use of Header switches (PMOS) as Sleep Transistors to control the Power Gating block. A schematic implementation is shown in Figure 5.1 where $SL = \{0,1\}$ is the sleep signal corresponding to {ON,OFF} which is input to the Sleep Transistor Q_1. V_{dd} is the global grid power supply and V_{ddv} is the virtual power supply for the gated block.

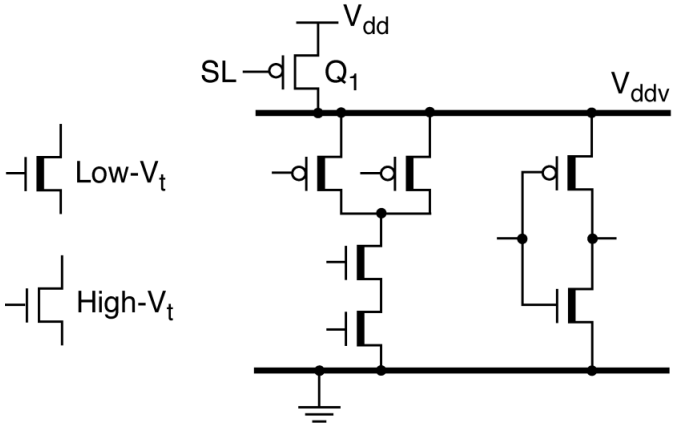


Figure 5.2 MTCMOS Power Gating implementation with Header Switch [7]

Given the implementation in Figure 5.2 we can find the Energy dependence of the entire block when the Sleep Transistor is turned ON and when it is turned OFF. This has been studied in detail by Z. Hu et al. in [28] and the key intervals in a power gating cycle are shown in Figure 5.3 [28]. Here, at T_1 the control circuit decides to power-gate the unit. Between T_1 and T_2 the signal is buffered and transferred to the Sleep Transistor and at T_2 the virtual V_{dd} begins to fall. Though the drop would be linear in time till it reaches the minimum at T_4 (virtual V_{dd} line fully discharged), in reality the leakage current also reduces and this increases the rate of drop in voltage. The interval between T_2 and T_4 is considered the minimum Idle Time for that particular power-gated unit and also referred to as the 'Sleep Cycle'. At T_5, the control logic decides to activate the unit again and the signal reaches the Sleep Transistor at T_6. The virtual V_{dd} line is charged back up to V_{dd} level between T_6 to T_7 which constitutes the 'Wakeup' cycle.

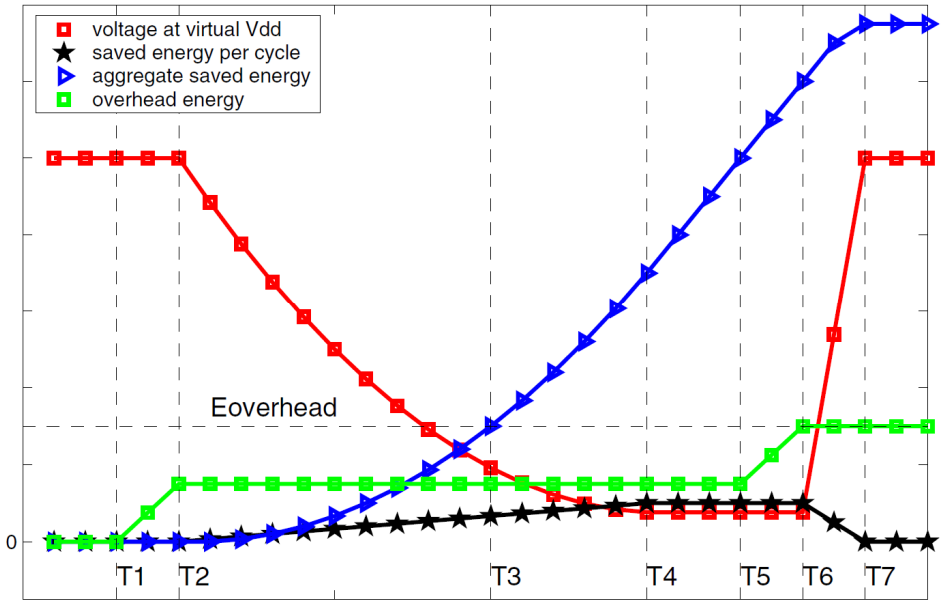


Figure 5.3 Key intervals in the power gating cycle [28]

For the purposes of the analysis methodology used for studying effect of Power Gating on voltage noise in this work, we make certain assumptions.

- We assume that there is no delay between the control logic to Sleep Transistor input. So, $T_2 = T_1$ and $T_5 = T_6$ from the previous case.

- Since we are analyzing a small design, we make the assumption that the 'Sleep' and 'Wakeup' cycles are of equal length and that the voltage transition on the virtual Vdd occurs linearly. So, $T_4 - T_2 = T_7 - T_6$. $Slope_{sleep} = - Slope_{wakeup}$.

- The standby leakage current, during power-gated state, is assumed to be zero.

The above assumptions contribute to the generation of constraints and can be easily adapted to a real design. This is explained later in this section. Also, the assumptions allow us to model the Sleep Transistor as a Voltage-Controlled Resistor whose resistance simulates the ON and OFF characteristics of the Sleep Transistor.

The basics of wavelet construction have already been covered in Section 3.1. To incorporate the Power Gating effect we make use of similar synthetic load current construction as described in Section 3.2. For a power-gated block the current load equation is described in (15).

$$i_j(t) = \sum_{p=1}^{t_0/u} (1 - \alpha_{p,j}) \sum_{m=1}^{m_0} \sum_{n=1}^{n_m} T_{m,n,j} \psi_{m.n}(t) \qquad (15)$$

where $i_j(t)$ represents all the current sources in the j^{th} power-gated block, t_0 is the time at which the voltage drop needs to be maximized, p is an index used to represent the time windows which are of unit size u and the number of such windows in which the current source can either be gated or non-gated is given by t_0/u (t_0 is assumed to be a multiple of

u for simplicity), $\alpha_{p,j}$ represents a binary variable attached to the j^{th} block and is explained in (16). There is no i_{min} term as we assume leakage current is zero for the gated block.

$$\alpha_{p,j} = \begin{cases} 1 & unit\ is\ gated \\ 0 & unit\ is\ not\ gated \end{cases} \tag{16}$$

The current equation for a non-gated block, if present, is same as (6) with $\alpha_{p,j} = 0$, always. This block can switch between maximum current and leakage current due to not being gated.

Now that the synthetic current load constraints have been set, we need to ensure that 'Sleep' and 'Wakeup' cycles which we will refer to as *Fall* and *Rise* transitions, respectively, do not occur concurrently. This can be set based on constraints on the values of α binary variable in various timeslots. Both these transitions are of unit time width, u, using the assumptions made earlier. So, the constraints must be satisfied for just over one timeslot.

The constraints consist of IF-ELSE conditions formulated as mathematical equations for ILP. We first deal with the *Rise* transition. For the power-gated block, based on the binary variable $\alpha_{p,j}$ for block j we conclude that the block is in the Rise transition if following pattern is observed.

- Given p and $p+1$ timeslots, if $\alpha_{p,j} = 1$ (gated) and $\alpha_{p+1,j} = 0$ (not-gated) then to ensure a proper *Rise* transition, we set $\alpha_{p+2,j} = 0$. The transition itself occurs in the $\alpha_{p,j}$ timeslot. Since the maximum number of timeslots is t_0/u, this is the limit for the value of $p+2$. The corresponding constraint is generated using an IF-ELSE mathematical function with the inputs being $\alpha_{p,j}$, $(1-\alpha_{p+1,j})$ and the output is $(1-\alpha_{p+2,j})$. Here, M is a large integer value greater than the sum of upper bounds of the rest of the variables in the equation.

$$M * (1 - \alpha_{p+2,j}) - 1 \geq \alpha_{p,j} - \alpha_{p+1,j} \quad (17)$$

For the case of the *Fall* transition, in *p* and *p+1* timeslots, if $\alpha_{p,j} = 0$ and $\alpha_{p+1,j} = 1$ then for a *Fall* transition, $\alpha_{p+2,j} = 1$. The corresponding set of constraints are generated with an IF-ELSE construct with inputs being $(1-\alpha_{p,j})$, $\alpha_{p+1,j}$ and the output is $\alpha_{p+2,j}$.

Equation (18) describes the mathematical constraints to achieve the above requirement.

$$M * \alpha_{p+2,j} - 1 \geq \alpha_{p+1,j} - \alpha_{p,j} \quad (18)$$

Now that we have described the process of setting the *Rise* and *Fall* transitions for an unit time *u*, we can extend this for more complex designs where the transitions occur over multiple timeslots and *Fall* transition takes more time than *Rise* transition. This is easily done by instantiating multiple IF-ELSE constructs to confirm with the transition requirements. Equations (17) and (18) can also be instantiated multiple times for each timeslot of concern to ensure that the *Fall* and *Rise* variables are not set in the same timeslot.

Objective Formulation: Due to the assumptions explained in Section 5.1 the objective function is formulated similar to the clock gating programming model in Section 3.4.1 with the added constraints for *Rise* and *Fall* transitions.

The complete problem formulation is described below where *G* is the total number of power-gated blocks and *N* is the number of non-gated block.

Maximize:

$$V_{drop}(t_0) = \sum_{j=1}^{G} \sum_{m=1}^{m_0} \sum_{n=1}^{n_{m,j}} T_{m,n,j} * h_{m,n,j,z}(t_0)$$

$$+ \sum_{j=1}^{N} \left[V_{dc,j} + \sum_{m=1}^{m_0} \sum_{n=1}^{n_{m,j}} T_{m,n,j} * h_{m,n,j,z}(t_0) \right]$$

Constraints:

For Gated Blocks:

$$0 \leq \sum_{m=1}^{m_0} T_{m,n,j} \psi_{m,n}(t) \leq \left(1 - \alpha_{p,j}\right)\left(i_{max,j} - i_{min,j}\right) for\ j \in (1,..,G)$$

Rise Transition:

$$1 - \alpha_{p+2,j} \geq \alpha_{p,j} - \alpha_{p+1,j}\ for\ p \in \left(1,..,\frac{t_0}{u} - 2\right), j \in (1,..,G)$$

Fall Transition:

$$\alpha_{p+2,j} \geq \alpha_{p+1,j} - \alpha_{p,j}\ for\ p \in \left(1,..,\frac{t_0}{u} - 2\right), j \in (1,..,G)$$

For Non-Gated Blocks:

$$i_{min,j} \leq \sum_{m=1}^{m_0} T_{m,n,j} \psi_{m,n}(t) \leq i_{max,j}\ for\ j \in (1,..,N)$$

In the above formulation m_0 is the maximum number of wavelets used, $n_{m,j}$ is the maximum number of shifts that each wavelet at scale m for the j^{th} block undergoes, $\psi_{m,n}(t)$ can be either A_m or $-A_m$ or 0 (in case the wavelet is not present in the time slot, $A_m = 2^{-m/2}$), $i_{max,j}$ is the maximum current that the particular load consumes and $i_{min,j}$ is the leakage current for the non-gated blocks.

The final output of the Programming model solution will be the approximate voltage drop estimate and the gating pattern for the power-gated blocks. The gating

patterns generated with this model will become the initial inputs to the genetic algorithm phase to provide a good starting point.

This particular model can be easily applied to 3D power grid similar to the clock gating model. The POI z is selected on a particular tier and the voltage responses for all the blocks distributed across the tiers is found at z.

5.3 Genetic Algorithm background [32]

A Genetic Algorithm is a search heuristic that mimics the process of Natural Selection and is a part of a much larger group of Evolutionary Algorithms. Techniques that mimic various processes of natural selection like Mutation, Selection, Crossover, etc can be codified in any convenient computer language. Genetic Algorithms have become quite popular in solving a large array of optimization problems.

There are two primary parts to a Genetic Algorithm:

- *Genetic Representation* of the solution domain. The most common representation is an array of bits. Other representations, like lists of numbers, can also be created.
- *Fitness* or *Evaluation Function* is used to evaluate the solution domain.

The evolution begins with the creation of a population of randomly generated individuals or candidate solutions. Each individual has a set of properties known as *Chromosomes* or *Genomes* which can be manipulated via the genetic operators like *Crossovers* or *Mutations*.

Evolution is an iterative process where a particular population is also denoted as a *Generation* and all members of the population are evaluated for their *fitness* using the *Fitness Function*. *Fitness* is represented as a value of the objective function for the

optimization problem being solved. The individuals with higher fitness are selected stochastically from the current population and their genomes are modified using the genetic operators to create a set of new individuals for the next generation.

Evolution continues till certain conditions are met by the solutions or after a fixed number of generations or if it is determined that further iterations can no longer produce a better result.

We will now describe in greater detail certain terms used in Genetic Algorithm based programming.

Initialization: A set of individuals are randomly generated to form an initial population whose size can vary depending on the size of the problem. The random generation allows for greater coverage of the search space. It is possible to "seed" certain solutions into the initial population to help the algorithm zone in on a particular part of the search space that is likely to contain the optimal solutions.

Selection: This is the process of choosing individuals with greater fitness than the rest of the population for breeding the next generation. The fitness of each individual in a population is the primary driver for the selection process. There are various methods used to select the best individuals. For example, Tournament Selection used in this work involves running several "tournaments" between individuals chosen at random within the population and the winner of each tournament is selected for Crossover. A larger tournament size reduces the chances of a weaker member being chosen. A tournament

size of 1 represents a random selection. Other Selection methodologies like roulette-wheel selection or truncation of best individuals can also be used.

Genetic Operators: The main operators on the chromosomes of individuals of a population are *Crossover* and *Mutation*.

Crossover is the operation of taking more than one parent solution and creating a child solution from them. There are several methods to perform a *Crossover*.

One-point crossover is where a single point on the parent strings is selected and the data beyond that point is swapped between parents. An example is shown in Figure 5.4.

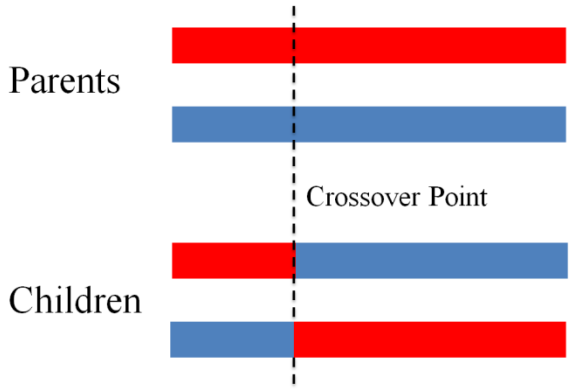


Figure 5.4 One-point crossover

Two-point crossover selects two locations on the parents to create the swapping regions. Figure 5.5 shows an example with two parents and two children.

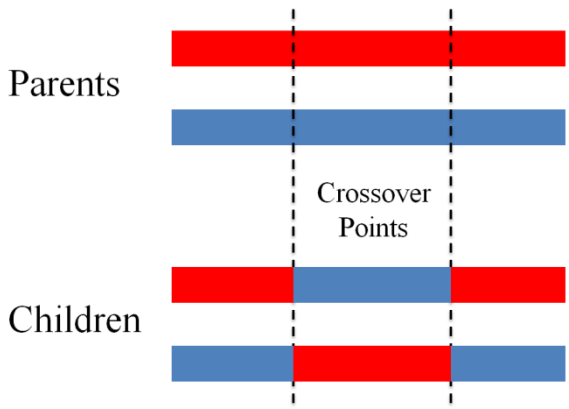


Figure 5.5 Two-point crossover

Uniform Crossover has a constant mixing ratio between the two parents. In difference to the previous two methods, Uniform Crossover allows the parents to contribute at a gene level rather than in segments. For example, if the mixing ration is 0.5 then the children will get half their genes from one parent and half from the other. Each parent gene has an exchange probability of 0.5. Figure 5.6 shows an example with mixing ratio at 0.5.

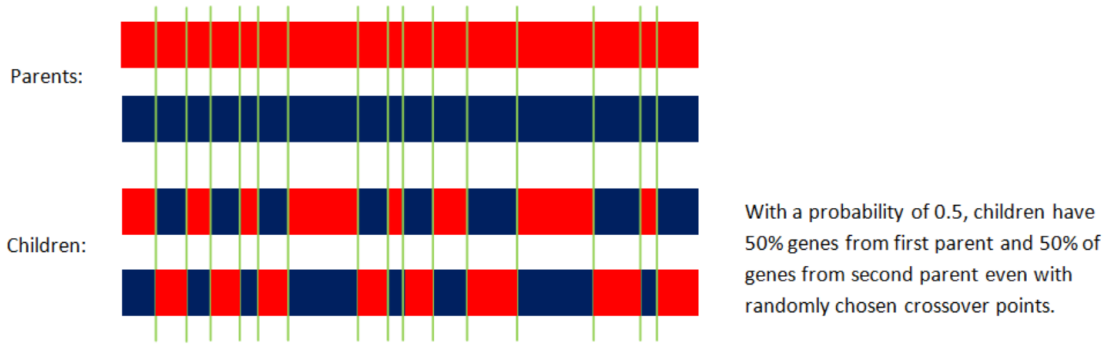


Figure 5.6 Uniform Crossover with 0.5 mixing ratio

There are various other methods but, the ones listed above are most used. It is shown in [34] that *One-point Crossover* performs better followed by *Uniform Crossover*

when performance was analyzed for a range of problems. Hence, in this work we utilize *One-point Crossover* during the coding of our genetic algorithm.

Mutation is the process of altering one or more gene values of a chromosome from its initial state. This is useful in maintaining the genetic diversity of population from one generation to the next. Probability of mutation should be kept low as a very high rate of mutation will make the searching process analogous to a random search. The main advantage of *Mutation* is to allow the genetic algorithm to avoid local minima which can cause the individuals of a population to be too similar and cause the search to stall.

Most *Mutation* methods involve a probability of an arbitrary bit in a sequence changing state. There are several methods of mutation like single-point mutation, inversion, floating point mutation, etc. An example for a *binary string mutation flip* is shown below. We make use of this method in our work.

Original string: 1 0 1 0 0 1 0 1

↓

Mutated string: 1 0 1 0 1 1 0 1

In the above example, the probability of mutation of a bit is 1/8 where binary string is 8-bit long.

Coding genetic algorithm from scratch can be a daunting task. Hence, we make use of an open-source Python package called Pyevolve [33] which allows us to quickly create a genetic program to solve our optimization problem. The package has an exhaustive set of options and methods to choose from and allows for quick selection of various methods for various parameters like *Crossover* method, *Selection* process, number of generations, etc. It allows support for a variety of *genome* types other than 1D

Binary string and the genetic algorithm engine is very efficient. Also, since Python has a good math library any kind of optimization problems can be solved by creating complex evaluation functions.

5.4 Design Flow

The general steps for the design flow used in this work are listed in this section.

5.4.1 Obtain the Impedance Response

Depending on the number of Power Gated blocks considered, their size and the sleep transistor parameters we construct the power grid as a RLC network. The impedance profiles for the constructed power grid is found for all the states of the power gated blocks. For 'G' power gated blocks there will be '2^G' states as each block can be either gated or not-gated. Using this we can set the frequency bounds (f_{min} and f_{max}) and we proceed with the construction of the set of base wavelets for m = 1,..., m_0.

5.4.2 Wavelet Parameter Calculations

Once we specify the frequency region of interest in the impedance profile of the grid we can calculate m_0 and u from (3) and (4). We can also set the target time, t_0, which should be equal to or greater than the time-span of the slowest wavelet (m= m_0). In this work, the target time is a multiple of u. The set of basis wavelets generated from the given parameters are used for further analysis.

5.4.3 Tabulate the voltage responses

The power grid is loaded with the necessary current loads. After assuming all the power gated blocks are always ON, for each set of current loads in the power gated and non-gated blocks we input each one of the basis wavelets, $\psi_{m,n}(t)$, (given by m) and shift them backwards from time t_0 (n represents the number of backward shifts). Figure 3.1

showed an example of a wavelet shifted backwards from time t_0 and its corresponding response at z. For the non-gated blocks we also obtain the DC responses similar to the clock gating case.

5.4.4 Solve the Integer Linear Programming (ILP) model

Now we generate and solve the ILP model with the relevant data obtained by assuming the power gated blocks are always ON. This will result in a pattern for the gated blocks which will serve as an initial input to the genetic algorithm.

5.4.5 Genetic Algorithm instantiation and solution

Utilizing the Python toolkit Pyevolve [33], we create the genetic algorithm. We instantiate a *genome* object which represents a 1D Binary String whose length is equal to the Number of Blocks X Number of timeslots. We should note that the ILP generates a binary pattern for all the blocks present in the grid and hence, this decides the length of the binary string used. Various parameters, like the type of Crossover function to use, and the evaluation function for the *genome* object are set.

Once the genome object is passed to the genetic algorithm engine, *ga*, an initial population is created (denoted as the 0^{th} generation) and evaluated. We need to access this population and replace one of its members with the initial binary string obtained from the ILP using a dedicated function. The population is then re-evaluated and sorted. This provides a good starting point for the genetic algorithm to search for a solution. The algorithm engine also accepts the total number of population sets, called *Generations*, to be generated and evaluated.

The evaluation function accesses each member of the population, denoted as a *chromosome*, and splits the 1D binary string to obtain the individual binary strings for all

the blocks and contains constraints to check for *Rise* and *Fall* transition concurrency for the power gated blocks' binary patterns which are designated as violations. A negative score is assigned to the *chromosomes* with these patterns. For a *chromosome* without these violations we then pass it to a SPICE script that will use the binary patterns for the various blocks as inputs and generate the voltage drop at the point of interest, z, at a particular time, t_0. This voltage drop is assigned as the score for that *chromosome*. The algorithm automatically does the evaluation for all members of the population of the i^{th} generation and assigns scores to them. The next generation is created using the chose Selection process and genetic operators, namely, Crossover and Mutation.

Once all the Generations have been evaluated the program outputs the binary string which is considered as the best individual. This string can then be split into its parts to obtain the gating pattern for the power gated blocks.

Extension to a 3D power grid is simple and the analysis can be performed for a point of interest on any tier of our choosing.

CHAPTER 6

POWER GATING EXPERIMENTAL SETUP AND RESULTS

In this section, we explore the experimental setup used for the analysis of the effect of power gating on supply noise in the case of both 2D and 3D IC power grids. The results are also analyzed and certain key observations are made.

6.1 Specification for the 2D Power Grid

The external power supply construction is the same as the one used for Clock Gating analysis as shown in Figure 4.1. The external supply is connected to the die via micro-C4 bumps with a standard pitch of 100 µm used throughout this work. The parasitics for each bump is listed below (From Section 4.1).

- Resistance = 40 mΩ
- Inductance = 70 pH

The on-chip power grid is slightly different from the construction for the Clock Gating analysis as the Power Gated blocks have their local power grids. Hence, we increase the number of metal layers to include M4 and M5, which form the global grid, with M2 and M3 now being used for the local gated block grids. Specifications for the construction of the interconnects is obtained from the Technology File of the NCSU PDK for 45nm [22]. The pitch is kept the same (at 25 µm) for both the local and global grids while the widths are adjusted to get the same unit cell parasitics in both grids. Figure 6.1 shows the unit cell of the global and local power grids. We assume the ground network is ideal in this work. So, the unit cell parasitics obtained are the same as shown in Figure 4.3.

The RLC parameters are:

- R = 700 mΩ - L = 8.2 pH - C = 0.4 fF

The die size is maintained at 1mm X 1mm which given the pitch gives us 40 X 40 nodes on the global power grid. This grid is constructed by replicating the unit cell.

For the construction of the local power grids we need to know the size of each power gated block. In this work, we assumed there are 2 power gated blocks, designated as Block A and Block B, which are 10 X 10 nodes in width. For loading these blocks the same unit of load of an AES circuit is taken as in the Clock Gating analysis. The specifications for the load as found earlier in Section 4.1 are listed again for convenience. The supply voltage is set at 1V.

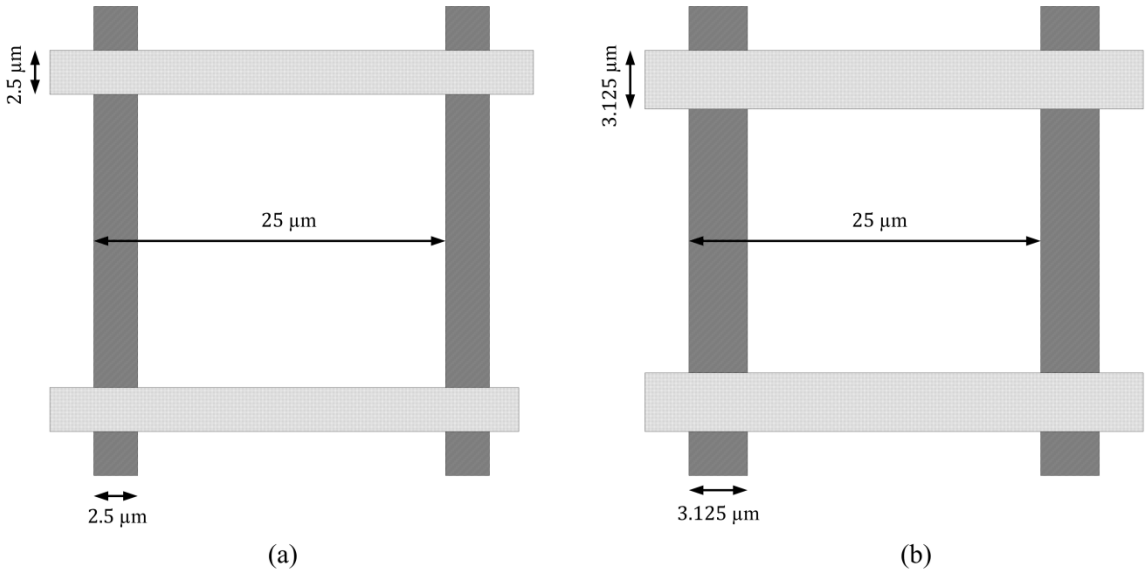


Figure 6.1 (a) Global Grid unit cell, (b) Local Grid unit cell

Table 6.1 AES load specifications

# of Gates	14090
Total Area	16419 μm^2
Area (Nodes)	5 X 5
Peak Power	80 mW
Leakage Power	0.3 mW
$\mathbf{i_{max}}$	80 mA
$\mathbf{i_{min}}$	0.3 mA

Given the 10 X 10 nodes wide power gated blocks we get 4 unit AES loads in each of the gated blocks. Also, we assume that the block consumes no power when gated. The maximum current consumption for the block is 320 mA (4 * unit AES load).

A non-gated block, designated as Block C, is also assumed to be present on the global power grid with the same width of 10 X 10 nodes. This block still consumes leakage power and hence, its consumption varies between 320 mW and 1.2 mW.

The local grids of the gated blocks are connected to the global grid via PMOS sleep transistors (Header cells). Sleep transistor parasitics depend on the size of the transistor. We refer to [29] to find the size of a sleep transistor. Equations (19) through (23) will describe the necessary steps needed to find the width of the sleep transistor for the given power gated blocks.

The delay of a single gate, τ_d, in the absence of a sleep transistor is given by (19).

$$\tau_d = \frac{C_L V_{dd}}{(V_{dd} - |V_{tL}|)^{\alpha}} \tag{19}$$

where C_L is the output capacitance of the gate, V_{dd} is the supply voltage, V_{tL} is the threshold voltage for the gate and α is the velocity saturation index, $1 \leq \alpha \leq 2$. For 45nm, α = 1.8 [30]. In the presence of the sleep transistor the new delay for the gate, τ_d^{sleep}, can be expressed as given in (20).

$$\tau_d^{sleep} = \frac{C_L V_{dd}}{(V_{dd} - V_X - |V_{tL}|)^\alpha} \tag{20}$$

where V_X is the voltage drop across the sleep transistor. Assuming a 5% degradation in performance is acceptable limit for proper circuit operation due to the presence of the sleep transistor we get,

$$\frac{\tau_d}{\tau_d^{sleep}} = 95\% \tag{21}$$

Solving (20) for V_X with α = 1.8 we get,

$$V_X = 0.0281(V_{dd} - |V_{tL}|) \tag{22}$$

Now, the current flowing through the sleep transistor in 'Triode' region, I_{sleep}, can be expressed as,

$$I_{sleep} = \mu_p C_{ox} \left(\frac{W}{L}\right)_{sleep} \left[(V_{dd} - |V_{tH}|)V_X - \frac{V_X^2}{2}\right] \tag{23}$$

where μ_p = 0.021 m^2/Vs is the hole mobility, C_{ox} = 19.7 X 10-7 F/m for 45nm is the gate oxide capacitance, V_{tL} = -0.3021 V is the low-threshold voltage for gated block's PMOS gate and V_{tH} = -0.5044 V is the high-threshold voltage for the sleep transistor, V_{dd} = 1V, Isleep = 320 mA, which is the maximum current that the transistor must handle. and L = 50 nm. These values are obtained from the NCSU PDK technology files [22]. Solving (28) and (29) for W_{sleep} we get,

$$W_{sleep} = 3640\ \mu m$$

Given the width of a sleep transistor that can handle 320 mA peak current, we split the transistors into 8 equal transistors, each of width 455 µm, and spread them around the periphery of the gated blocks and the locations are shown in Figure 6.2. These transistors are modeled as Voltage-Controlled Resistors (VCRs) in SPICE which connect the local and global power grids for the purpose of this work. We simulate the unit sleep transistors in HSPICE and find the on- and off- resistances which will set the bounds for the VCR description. We also find the drain and source capacitance values for the transistor. The parasitics are listed below.

- R_{on} = 10 Ω
- R_{off} = 10 MΩ
- $C_D = C_S$ = 1.3 pF

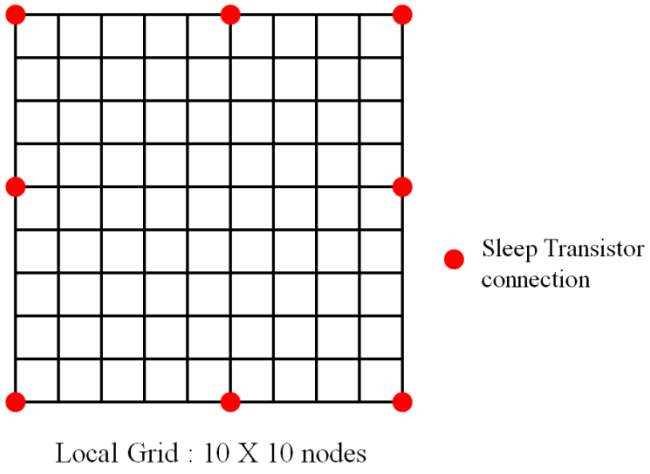


Figure 6.2 Sleep transistor locations on the local power grid

Using the above setup we construct the final power grid consisting of 2 power gated blocks (Blocks A and B) and a non-gated block (Block C). Their distribution on the grid is shown in Figure 6.3 along with the point of interest (POI).

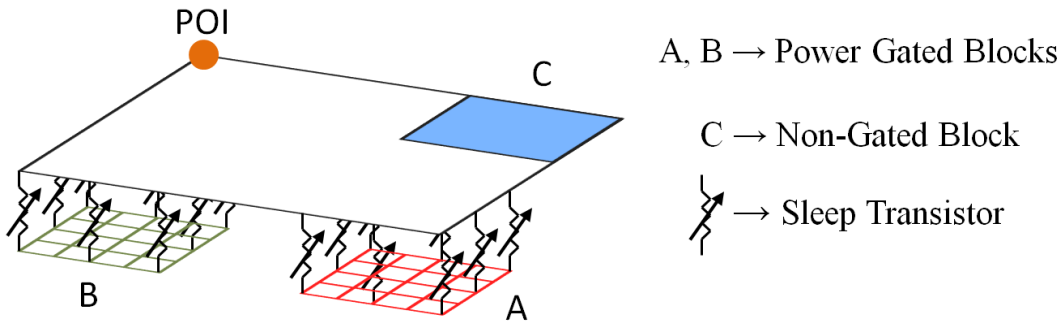


Figure 6.3 Block Locations on the Global Power Grid

This power network is simulated in HSPICE to find the normalized impedance profiles of the grid at the Point Of Interest (POI). We get 4 profiles depending on whether Blocks A and B are on ('A' or 'B') or off ('Ab' or 'Bb'). This profile is shown in Figure 6.4.

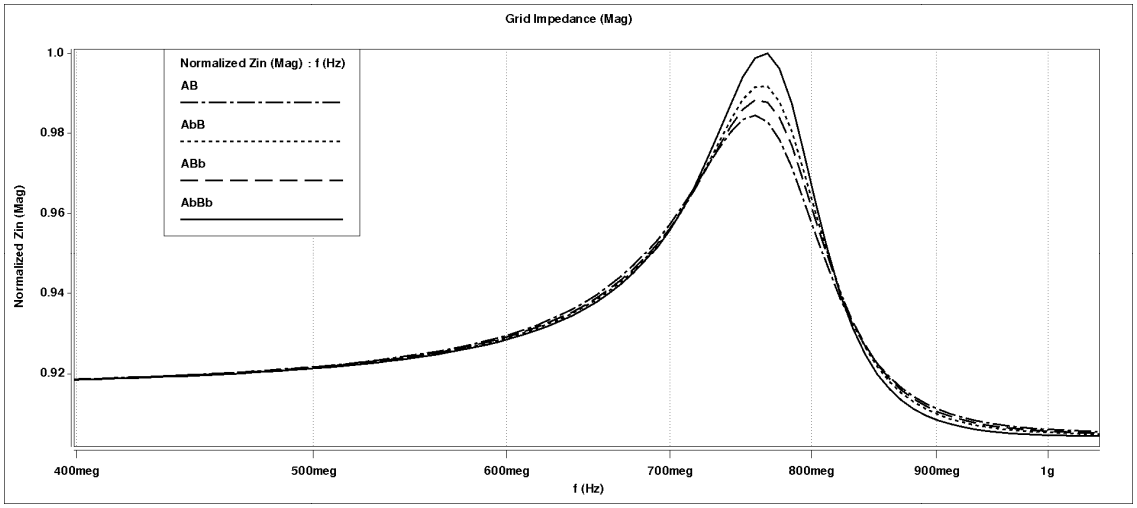


Figure 6.4 Normalized Impedance Profiles for the 2D power grid

From Figure 6.4 we set the frequency bounds as,

- f_{min} = 450 MHz

- f_{max} = 900 MHz

from which we get m_0 using (3) and u using (4) as:

- m_0 = 3

- u = 0.412 ns

Now, we set the target time at which to maximize the voltage noise as:

- t_0 = 3.296 ns

- This gives us a total of t_0/u = 8 time slots.

6.2 Results for the 2D power grid

Using the above construction we first formulate the Linear Programming (LP) model by assuming that the power gated blocks are always connected to the global grid. This enables us to generate an approximate solution that will provide a good target search area for the genetic algorithm to search for a better solution. The LP model is generated and solved using GLPK [26]. For the Genetic Algorithm we make use of the open-source package Pyevolve [33] and write the program in Python. Some of the parameters set in the Pyevolve package are listed below.

- *Genome* type = 1D Binary String
- *Genome* legth = 24 bits (3 blocks X 8 timeslots)
- # of Generations = 50
- *Selection* method = Tournament Selector
- *Crossover* method = Single-point Crossover
- *Mutation* method = String Mutator Flip

The gating patterns and the corresponding voltage drops obtained in SPICE for the LP model and the genetic algorithm (GA) are listed in Table 6.2. We also illustrate the advantage of LP solution being seeded into the initial population of the Genetic Algorithm (GA) in Table 6.3 where we see a large runtime improvement with respect to running Genetic Algorithm with randomly generated initial population.

Table 6.2 2D Power Grid Results

		Block	**Timeslot**								**Voltage Drop (mV)**
			1	2	3	4	5	6	7	8	
LP	α	A	0	1	1	0	1	1	0	0	12
		B	0	1	1	0	1	1	0	0	
	Voltage Control (V)	A	0	f	r	0	f	r	0	0	
		B	0	f	r	0	f	r	0	0	
GA	α	A	1	0	0	0	0	1	1	0	34
		B	0	0	1	1	1	1	1	0	
	Voltage Control (V)	A	r	0	0	0	0	f	r	0	
		B	0	0	f	1	1	1	r	0	

where $f \rightarrow$ Sleep transition slope (0V to 1V), $r \rightarrow$ Wake transition slope (1V to 0V) and α = {0,1} implies {ON,OFF}. The voltage control pattern is the required gating pattern for each of the power gated blocks. This pattern is given to the Sleep transistors, PMOS Header cells.

Table 6.3 Runtime Statistics

	Runtime	Improvement
Random Initial Population	11 hrs	-
LP seeded Population	3 hrs	72%

The Linear Programming (LP) model also outputs a voltage drop which is the best solution it found given the set of constraints and an objective function. We input the gating patterns obtained using the Genetic Algorithm (GA) back into the LP model and find a new voltage drop solution. Table 6.4 lists the voltage drop outputs of the LP model for both the gating patterns from the initial approximate LP model solution and from the GA solution and compares those to the values obtained from SPICE. We see that the LP model voltage drop is significantly higher compared to the corresponding drop obtained from SPICE. This is because the LP model assumed that the power grid was operating under linear constraints. Looking at the results from the GA model we see that the mathematical and SPICE solutions are closer showcasing the GA models ability to handle the non-linearity due to power-gating.

Table 6.4 Voltage drop accuracy for LP and GA models

Model	**Voltage Drop (mV)**	
	From LP model	**From SPICE**
LP	41.5	12
GA	31	34

Since the non-linear nature of the power gated grid cannot be fully captured using just a mathematical model, the LP model result does not produce the worst-case voltage drop. The Genetic Algorithm uses the voltage drops obtained from SPICE runs to score the *fitness* of each possible solution and hence, is better able to target the gating patterns to get the worst-case voltage drop. The voltage drop waveforms generated by both LP and GA models at the point of interest, *z*, are shown in Figure 6.5 proving the Genetic Algorithm's solution produces the worst-case drop at the target time.

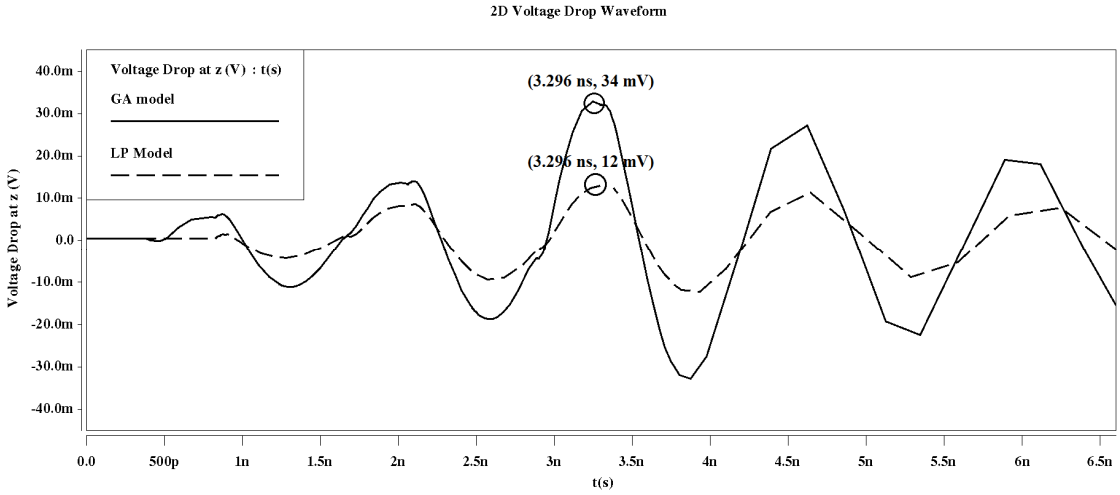


Figure 6.5 Worst-case voltage drop waveform at POI, *z*

To verify that the pattern obtained is indeed the worst-case pattern we need to run random pattern simulations. For this work we run 25,000 sets of random pattern simulations with each block having an 8-bit random vector assigned to it. We use a script that parses the random vector and generates the voltage control signals for the sleep transistors of the 2 power gated blocks and the current signals for all the 3 blocks. SPICE simulations are run and the maximum voltage drop obtained between the time periods 0 to t_0 is tabulated. Figure 6.6 shows the plot of the maximum drop values for the 25,000

sets of simulations and also, the drop values obtained from LP and corresponding SPICE simulation are plotted for comparison. We see that our methodology produces a better result than the random patterns.

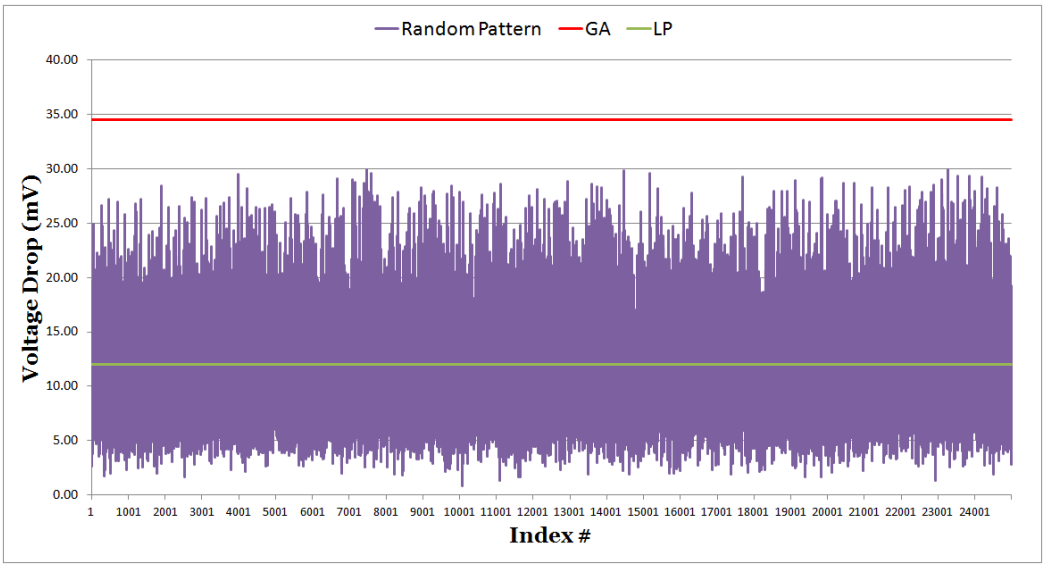


Figure 6.6 Voltage drop comparison

6.3 Specifications for the 3D power grid

In case of the 3D power grid, we retain the same external power supply parasitics as shown in Figure 4.1 and construct a power grid with 3 tiers with a grid size of 23 X 23 nodes. Also, we consider the same unit grid cell discussed in Section 6.1 to construct the power grids in each of the tiers. A representation of the 3 tiers used is shown in Figure 6.7. Tier 1 is the topmost tier (i.e., connected to the package via the C4 bumps), and Tiers 2 and 3 are the bottom tiers connected to the previous tiers using Through Silicon Vias (TSVs). The positions of the various blocks are also shown. The size of each block is retained at 10 X 10 nodes. The POIs for each tier are also marked.

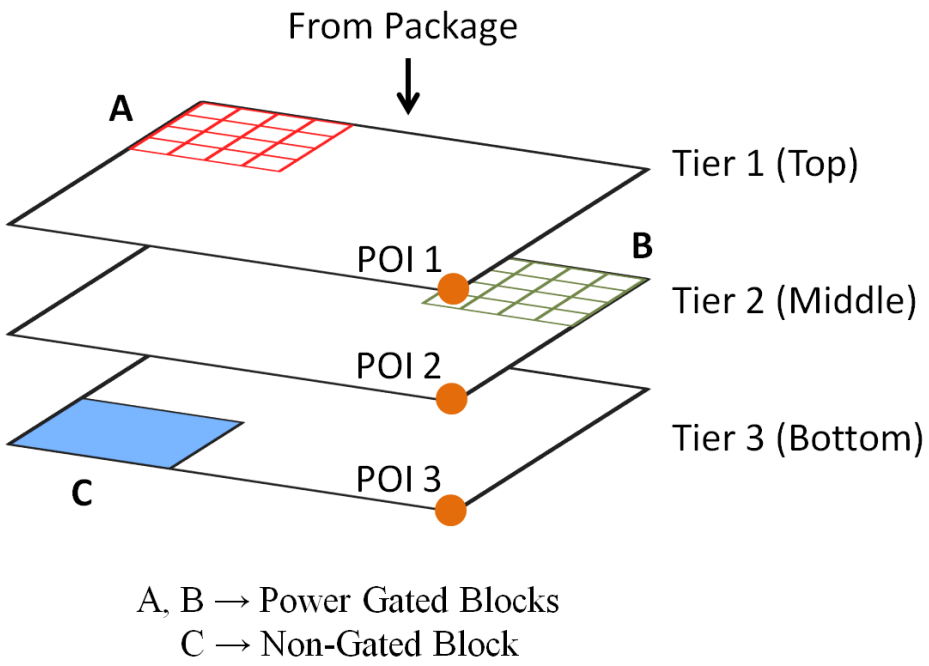


Figure 6.7 3D Power Grid with location of the blocks

The TSV parameters used are the same as in Section 4.3. The parasitics are relisted here for convenience.

- R_{TSV} = 20 mΩ
- L_{TSV} = 20 pH
- C_{TSV} = 25 fF

The power gated blocks are connected using 8 sleep transistors to the global grids in each layer. The sleep transistors are modeled as Voltage-Controlled Resistors (VCRs) for the SPICE simulations.

The impedance profile of the grid is obtained at the POIs for each tier in SPICE. We follow the same procedure as for the 2D grid by switching the Voltage-Controlled Resistors of the power gated blocks and observe the impedance profiles at each tier. Figure 6.7(a) shows the impedance profiles across the tiers for a particular case where

blocks A and B are gated. Figure 6.7(b) shows the impedance profiles for various gating combinations of A and B for a particular tier (Tier 2) which highlight the behavior of the grid in that tier.

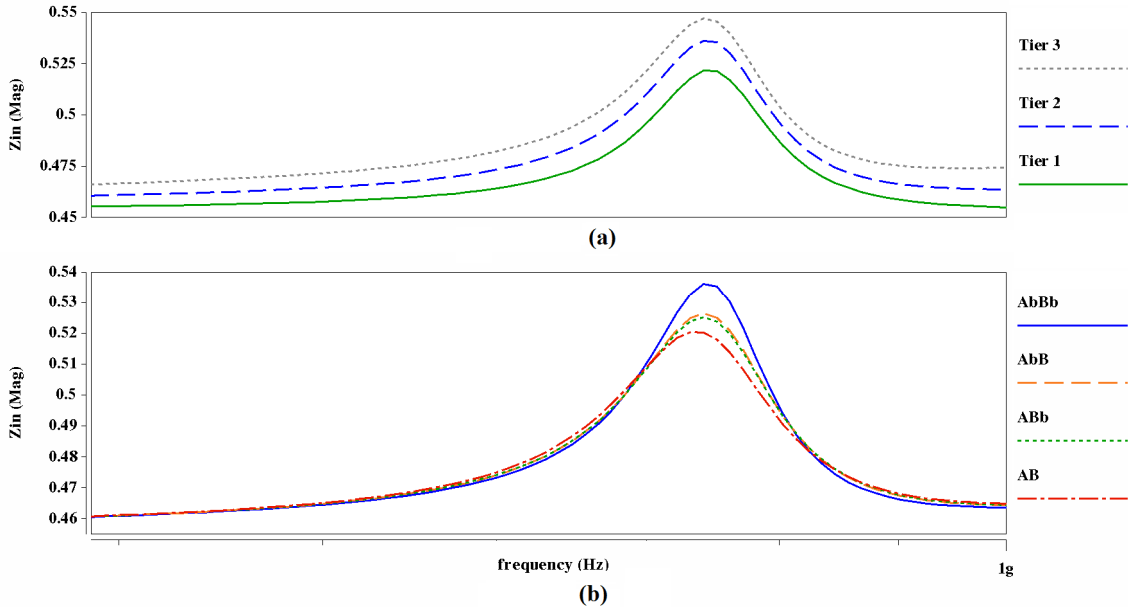


Figure 6.8 3D PDN Impedance Profile : (a) Across tiers for 'AbBb', (b) Tier 2 for all gating combinations

In this case, the peaks occur at similar frequencies compared to the 2D power grid. From the impedance profiles obtained, we set the frequency bounds and obtain the wavelet parameters as listed below:

- f_{min} = 450 MHz
- f_{max} = 900 MHz

from which we get m_0 using (3) and u using (4) as :

- $m_0 = 3$
- u = 0.412 ns

We set the target time at which the voltage noise is to be maximized as:

- t_0 = 3.296 ns
- This gives us a total of t_0/u = 8 time slots.

6.4 Results for the 3D PDN

We utilize the same parameters as the 2D PDN problem for the Pyevole [33] package. To study the power gating effect on the 3D Power Distribution Network (PDN), we construct the LP model for each tier targeting the POI on each tier and maximize the voltage drop at time t_0. Then, we feed the gating patterns generated from the LP to Genetic Algorithm for finding optimal solution for each tier.

Table 6.5 shows the gating patterns for the two power-gated blocks targeting the point of interest on each tier and the corresponding voltage drops obtained from the genetic algorithm based search. We see that as in the case of Clock Gating analysis the drop increases with the tiers. The voltage drop values for the higher tiers are also higher than the results from the power gating analysis for the 2D grid due to the limited power supply connections (C4 pins) to the external grid and the added parasitics of the TSVs. The tier closest to the C4 pins observes a slightly lower voltage drop as the absolute value of the impedance peak was lower than the 2D case. The larger drop between Tiers 1 and 2 compared to Tiers 2 and 3 is evident from the differences in the impedance values in Figure 6.8(a) across the three tiers.

Table 6.5 Gating Patterns and voltage drops for all Tiers

α	Block	Timeslot								Voltage Drop (mV)
		1	2	3	4	5	6	7	8	
Tier 1	A	0	1	1	0	0	0	0	0	30.86
	B	0	0	1	1	1	1	1	0	
Tier 2	A	0	0	1	1	1	1	1	0	38.42
	B	1	1	1	0	0	1	1	0	
Tier 3	A	0	0	1	1	0	0	1	1	42.89
	B	0	0	0	0	0	1	1	0	

where *Sleep* transition slope (0V to 1 V), *Wake* transition slope (1V to 0V) and α = {0,1} implies {ON,OFF}. The Voltage control pattern is the input to the PMOS Sleep transistors.

CHAPTER 7

CONCLUSION

We presented a technique to study the effect of clock gating on the voltage drop at a particular point on the power grid. Our solution utilized a wavelet based modeling of current loads on the grid. Wavelets allow us to characterize the frequency-domain information of the power grid and thus, allow our formulation to target the resonance frequencies of the grid. This gives us the worst-case voltage drop at any point of our choosing. Also, we described methods to extract the clock gating patterns that resulted in the worst-case drop using our formulation. We studied the gating effects for both a 2D power grid and 3D power grid and tabulated the gating patterns. We compared our solution for the worst-case voltage drop at the target location to the results obtained by running simulations with a large set of random gating patterns and recording the voltage drops.

We also perform a similar analysis to study the effect of Power Gating on the voltage drop at a particular point on the grid and highlight the differences between the analyses for clock gating and power gating. Also, we touched upon some of the difficulties that may arise while formulating the problem.

Information obtained from the Clock- and Power- Gating analyses is valuable to designers for analyzing the robustness of particular blocks in the case of noise and also the gating patterns may be used as a basis for including architectural or control circuit changes to compensate for the worst-case noise.

BIBLIOGRAPHY

[1] The International technology roadmap for semiconductors (ITRS), 2009

[2] M. R. Stan ,K. Skandro, K. Sankaranarayanan, S. Ghosh and S. Velusamy "Compact Thermal Modeling for Temperature aware Design" In *Proceedings DAC*, 2004

[3] Jairam S, et. al., "Clock gating for power optimization in ASIC design cycle theory & practice", In *Proceedings of the 13th international symposium on Low power electronics and design*(ISLPED '08), ACM, New York, NY, USA, 307-308, 2008

[4] H. Mair, et. al., "A 65-nm mobile multimedia applications processor with an adaptive power management scheme to compensate for variations", In *Proceedings of the Symposium on VLSI Circuits*, 2007

[5] S. Rusu, "A 65-nm dual-core multithreaded Xeon processor with 16-MB L3 cache", *IEEE J. Solid-State Circuits 42*, 1, 17–25, 2007

[6] Y. Ye, S. Borkar and V. De, "A new technique for standby leakage reduction in high performance circuits", In *Proceedings of the Symposium on VLSI Circuits*, 40–41, 1998

[7] Y. Shin, J. Seomun, K.-M. Choi and T. Sakurai, "Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs", *ACM Trans. Des. Autom. Electron. Syst. 15*, 4, Article 28, September 2010

[8] L. Li, K. Choi and H. Nan, "Effective algorithm for integrating clock gating and power gating to reduce dynamic and active leakage power simultaneously", *12th International Symposium on Quality Electronic Design* (ISQED '11), pp. 14-16, 2011

[9] M.D. Pant, P. Pant, D. S. Wills and V. Tiwari, "An architectural solution for the inductive noise problem due to clock-gating", In *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 255-257, 1999

[10] M. Popovich, A. Mezhiba, E.B. Friedman, Power Distribution Networks with On-Chip Decoupling Capacitors, Springer, 2007

[11] M. Benoit, S. Taylor, D. Overhauser, and S. Rochel, "Power Distribution in High-Performance Design," *Proceedings of the IEEE International Symposium on Low Power Electronics and Design*, pp. 274–278, August 1998

[12] W. Rhett Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498-510, Nov./Dec. 2005

[13] V. F. Pavlidis and G. D. Micheli, "Power distribution paths in 3-D ICS", *In Proceedings of the 19th ACM Great Lakes symposium on VLSI (GLSVLSI '09)*, ACM, New York, NY, USA, 263-268, 2009

[14] P. Jain, Tae-Hyoung Kim, J. Keane, C. H. Kim, "A multi-story power delivery technique for 3D integrated circuits," *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, pp.57-62, 2008

[15] P. S. Addison, The Illustrated Wavelet Transform Handbook, New York: Taylor & Francis, 2002

[16] I. A. Ferzli, E. Chiprout, and F. N. Najm, "Verification and co-design of the package and die power delivery system using wavelets", IEEE *Trans. Comp.-Aided Des. Integ. Cir. Sys, vol. 29, 1, pp.* 92-102, January 2010

[17] Russ Joesph , Zhigang Hu , Margaret Martonosi, "Wavelet Analysis for Microprocessor Design: Experiences with Wavelet-Based dI/dt Characterization", In *Proceedings* HPCA, 2004

[18] W. Zhang, et al., "Efficient power network analysis considering multidomain clock gating", IEEE *Trans. Comp.-Aided Des. Integ. Cir. Sys., vol.* 28, 9, pp. 1348-1358, September 2009

[19] W. Zhang, W. Yu, X. Hu, A. Shayan, A. E. Engin, and C. Cheng, "Predicting the worst-case voltage violation in a 3D power network", In *Proceedings of the 11th international workshop on System level interconnect prediction* (SLIP '09). ACM, New York, NY, USA, pp. 93-98, 2009

[20] J. Jang and W. P. Burleson, "An arbiter based on-chip droop detector system", In *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI* (GLSVLSI '11). ACM, New York, NY, USA, 1-6, 2011

[21] M. S. Gupta, J. L. Oatley, R. Joseph, Gu-Yeon Wei, D. M. Brooks, "Understanding Voltage Variations in Chip Multiprocessors using a Distributed Power-Delivery Network," *Design, Automation & Test in Europe Conference & Exhibition, DATE '07* , pp.1-6, 16-20, April 2007

[22] NCSU FreePDK45 kit, http://www.eda.ncsu.edu/wiki/FreePDK.

[23] Ansys Q3D extractor, Ansys Inc., http://www.ansoft.com/q3d/

[24] www.opencores.org

[25] Y. Lee, S. K. Lim, "Routing optimization of multi-modal interconnects in 3D ICs," Electronic Components and Technology Conference ECTC 2009. vol. 59, pp. 32-39, 26-29 May 2009

[26] http://www.gnu.org/software/glpk/

[27] H. Jiang, M. Marek-Sadowska, "Power gating scheduling for power/ground noise reduction," *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE* , pp.980,985, June 2008

[28] Z. Hu, et al., "Microarchitectural Techniques for Power Gating of Execution Units," *Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on* , pp.32,37, Aug. 2004

[29] M. Anis, A. Shawki, M. Mahmoud , M. Elmasry, " Dynamic And Leakage Power Reduction In MTCMOS Circuits Using An Automated Efficient Gate Clustering Technique", *Proc. of the 39th conference on Design Automation*, pp. 480-485, June 2002

[30] www.eng.auburn.edu/~agrawvd/.../project%20report_manish.doc

[31] I. Savidis, S. Kose, E.G. Friedman, "Power Noise in TSV-Based 3-D Integrated Circuits," *Solid-State Circuits, IEEE Journal of* , vol.48, no.2, pp.587,597, Feb. 2013

[32] http://en.wikipedia.org/wiki/Genetic_algorithm

[33] http://pyevolve.sourceforge.net/index.html

[34] J. Magalhães-Mendes, "A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop scheduling Problem", *WSEAS transactions on computers*, Vol. 12, No. 4, pp. 164-173, 2013

[35] K. Gala, V. Zolotov, R. Panda, B. Young, J. Wang, and D. Blaauw, "On-chip inductance modeling and analysis", *Proc. Design Automation Conference*, pp.63-68, 2000