



Title	Proactive Supply Noise Mitigation and Design Methodology for Robust VLSI Power Distribution
Author(s)	陳, 俊
Citation	大阪大学, 2020, 博士論文
Version Type	VoR
URL	<a href="https://doi.org/10.18910/76645">https://doi.org/10.18910/76645</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Proactive Supply Noise Mitigation and Design  
Methodology for Robust VLSI Power Distribution

Submitted to  
Graduate School of Information Science and Technology  
Osaka University

January 2020

Jun CHEN



# Publications

## Journal Article (Refereed)

- [J1] Jun Chen, Hajime Kando, Toshiki Kanamoto, Cheng Zhuo, and Masanori Hashimoto, “A Multicore Chip Load Model for PDN Analysis Considering Voltage-Current-Timing Interdependency and Operation Mode Transitions,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 9, pp. 1669–1679, Sept. 2019.

## International Conference Papers (Refereed)

- [I1] Jun Chen, Toshiki Kanamoto, Hajime Kando, and Masanori Hashimoto, “An on-chip load model for off-chip PDN analysis considering interdependency between supply voltage, current profile and clock latency,” in *2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, May 2018.
- [I2] Jun Chen and Masanori Hashimoto, “A Frequency-Dependent Target Impedance Method Fulfilling Both Average and Dynamic Voltage Drop Constraints,” in *2019 IEEE 23rd Workshop on Signal and Power Integrity (SPI)*, Jun. 2019.

## Domestic Conference Papers (Unrefereed)

- [D1] Toshiki Kanamoto, Koki Kasai, Masashi Imai, Atsushi Kurokawa, Masanori Hashimoto, Jun Chen, and Hajime Kando, “Optimization of Re distribution Layer and On chip Capacitors for LSI with FOWLP,” in *DA Symposium*, 2018 (in Japanese).
- [D2] Toshiki Kanamoto, Koki Kasai, Masashi Imai, Atsushi Kurokawa, Masanori Hashimoto, Jun Chen, and Hajime Kando, “LSI Package Board Power Delivery Network Modeling for Capacitor Placement Optimization at 15nm Node,” in *DA Symposium*, 2017 (in Japanese).



# Summary

With the scaling down of the technology node, both power consumption, and supply noise are continuously increasing in modern VLSI designs. The emergent power supply noise through the power delivery network (PDN) can eventually degrade the chip timing performance or even cause malfunction. Therefore, an effective supply noise mitigation system and PDN design methodology are critically important to ensure robust VLSI power distribution.

Designing a high-quality low noise PDN system is a complex and challenging task, which requires many efforts from PDN design stage to operation stage and extensive consideration throughout PDN components. For example, using switched capacitor voltage regulator (SCVR) as the power supply solution involves supply voltage ripple. Parasitic resistance and inductance of PDN can induce dynamic voltage drop by load current variation. At the chip load stage, supply noise degrades chip operation performance. Meanwhile, chip operation variation brings load current variation, which in turn, causes supply noise. Jointly considering these interdependent and heterogeneous aspects is the major difficulty in PDN design and noise mitigation.

Traditionally, PDN designers rely on a simple voltage guard bound as design guidance. Following such guidance, designers assume a max allowed voltage drop, and then determine the parameters of PDN components to meet the voltage drop constraints. To explore PDN parameters and verify the performance, considerable design time and run-time efforts are necessary. Next, a reactive noise mitigation system is introduced to dynamically regulate the load voltage such that the voltage guard bound is maintained during the operation stage. Nevertheless, designers just blindly believe the chip performance is ensured if voltage guard bound guidance is followed.

However, for large VLSI designs such as many-core systems, activity variation among multiple cores can result in considerable emergent large power requirements within tens of clock cycles. Therefore, the traditional voltage guard bound guidance is very difficult to meet at the PDN design stage because the allowed PDN impedance can be as small as micro Ohms across the wide frequency range. Moreover, during the system operation stage, the traditional reactive noise mitigation system fails to compensate such emergent supply noise due to systematic issues such as voltage sensing latency, voltage boosting latency through PDN, and limited voltage scaling capability. Finally, even with a dedicated noise mitigation control system and PDN design, the actual chip

performance impact is still invisible to PDN designers due to using the over-simplified load model. Such an issue can, in turn, mislead the PDN and noise mitigation system design, resulting in under- or over-designed PDN system and unexpected supply noise impact. Besides, with the rising popularity of machine learning technology, the proactive noise mitigation system based on chip load power/current prediction instead of reactive one is discussed to conceal the PDN latency. However, the mitigation solution either suffers from high hardware overhead or low prediction accuracy. Hence, the practical proactive supply noise mitigation and design methodology for off-chip PDN remains an open problem.

To put proactive noise mitigation into practical and improve the PDN design methodology, there are two major challenges need to be addressed. The first is negative loop challenge of proactive noise mitigation. A proactive noise mitigation controller requires a long-term accurate power/current prediction to conceal the PDN voltage setup latency. However, existing long-term prediction requires high computation cost and consequently long computation latency. Besides, traditional switched capacitor voltage regulator (SCVR) is a common off-chip power supply solution, but off-chip SCVR has limited voltage scaling flexibility and long response time. These two bottlenecks demand further longer-term prediction. Such a negative loop makes proactive noise mitigation suffer from either high hardware overhead or low prediction accuracy. Various works are proposed to address this challenge. For example, low dropout (LDO) voltage regulator is proposed to achieve fast noise mitigation response, but at the cost of heat generation and low energy efficiency. Multi-ratio SCVRs are also studied but the output ripple and limited voltage scaling level remain open problems. Till now, practical methodology to design a proactive noise mitigation system has not been established.

The second challenge is the design gaps in PDN design methodology. The first gap exists between PDN design constraints and target impedance design. Target impedance methodology is a common practice to bridge the PDN impedance with voltage drop constraints. However, actual PDN impedance is defined in the frequency domain while the voltage drop constraints are given in the time domain. Although the current spectrum tells us that dynamic power noise distributes within a certain frequency range, how to determine detailed frequency-dependent target impedance remains an open problem. The second gap exists between on-chip timing information, and off-chip PDN verification and exploration. Conventionally, very simple load models are provided to off-chip PDN designers, and hence on-chip behavior cannot be analyzed by them for PDN verification. Besides, supply voltage and clock frequency may be controlled for each core or a group of cores. Such a system behavior affects the power supply noise significantly, but due to its complexity, it is difficult for on-chip designers to construct even a simple chip load model for PDN configuration exploration purposes. Without the critical on-chip timing information, existing over-simplified PDN design methodology can mislead the PDN design, resulting in under- or over-designed PDN, and under- or over-estimated supply noise impact.

---

This dissertation proposes the proactive supply noise mitigation and PDN design methodology by addressing the above two challenges. For the first negative loop challenge, this dissertation manages to break the negative loop from two aspects. The first is to lighten the prediction cost by developing a lightweight short-term average current predictor. The second is to relieve the prediction length requirement by introducing a scalable major-minor voltage regulator (MMVR) structure. For the second design gap challenge, this dissertation proposes a frequency-dependent target impedance design methodology that considers the constraints of both average and dynamic voltage drops. A concept of magnitude equivalent frequency (MEF) is proposed to simplify the frequency-dependent target impedance design. To validate and explore the noise impact, this dissertation proposes a chip load model that can provide the on-chip timing information, replay detailed voltage-dependent current profile, and extensively explore the inter-core operation mode variation within a short run-time.

With the proposed methods, firstly a lightweight current predictor is derived, which consists of six-layer decision tree regressor and achieves over 0.99 correlation for 50-cycle-ahead prediction. Secondly, the proposed MMVR power supply solution achieved over 3X voltage scaling range compared with traditional SCVR while the ripple is within 16mV, which is 1.6% of load voltage. The proactive noise mitigation system is constructed using MMVR and predictor. Experimental results with a multi-core RISC-V design show that the proposed proactive mitigation system can mitigate the supply noise within 30mV while the noise exceeds 70mV with the conventional reactive mitigation. Also, the average supply voltage is compensated during the full operation period. Thirdly, the frequency-dependent target impedance is obtained which fulfills the voltage drop constraints. Experiments confirm that the synthesized target impedance satisfied the constraints with less than 0.1% error in the actual processor load case. Fourthly, a compact chip load model is derived, which is mostly described by Verilog-A. Experimental results show that the proposed model reproduces the current profile, current peak, and timing data well even while it achieves over 300X run-time reduction compared to a transistor-level model. It is also experimentally demonstrated that land side capacitor is helpful to improve processor timing performance in test cases.

The proactive noise mitigation methodology discussed in this dissertation helps to relieve the emergent supply noise so that the robustness for VLSI power distribution can be ensured. The methods proposed in this dissertation are also helpful for PDN designers to mitigate the over- or under-designed PDN impacts, and reduce the design cost and iteration time by facilitating the PDN verification and exploration process.



# Acknowledgments

First of all, I would like to express my deepest gratitude to Professor Masanori Hashimoto in Osaka University for providing me a precious opportunity and an excellent environment to study as a doctoral student in his laboratory. All of my productive researches are credited to none other than him. His advanced perspective and thoughtful advice led me to the achievements.

I would like to appreciate Professor Tatsuhiro Tsuchiya, Professor Tetsuya Hirose, and Associate Professor Hiromitsu Awano in Osaka University for detailed reviews and insightful suggestions.

I would like to appreciate Associate Professor Jaehoon Yu in Tokyo Institute of Technology, Professor Yoshinori Takeuchi in Kindai University, Professor Toshiki Kanamoto in Hirosaki University, Professor Cheng Zhuo in Zhejiang University for the precious suggestions and enormous help throughout my doctoral research.

My appreciation also goes to Mr. Hajime Kando in Murata Manufacturing Co., Ltd for technical discussions and suggestions on power delivery network modeling and chip load modeling.

I would like to express my sincere appreciation to Assistant Professor Yutaka Masuda in Nagoya University, and Dr. Wang Liao in Kochi University of Technology, for precious discussions and support.

I would like to thank other colleagues who belong or belonged to the Integrated System Design Laboratory in Osaka University for daily discussions and their support: Dr. Tomoki Sugiura, Dr. Koichi Mitsunari, Mr. Ryutaro Doi, Mr. Pei-Hao Chen, Mr. Tai-Yu Cheng, Mr. Ryo Shirai. I express my heartfelt thanks to all members of the Integrated System Design Laboratory in Osaka University for having an interesting and comfortable time in the laboratory. I would like to thank the laboratory secretary Mrs. Asako Murakami for her various support.

Finally, I would like to extend my gratitude to my parents (Yimin Chen and Guiying Liu), Miss Ziyan Li, other relatives and friends. They always support me and encourage me with their best wishes, inspiration, and suggestion.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	Power Delivery Network and Supply Noise . . . . .	4
1.1.2	Low Noise PDN Design Methodology . . . . .	7
1.1.3	Supply Noise Mitigation System . . . . .	9
1.2	Major Challenges for Robust Power Distribution . . . . .	10
1.2.1	Negative Loop Challenge of Supply Noise Mitigation . . . . .	10
1.2.2	Gap Challenge in PDN Design Methodology . . . . .	13
1.3	Objective and Organization . . . . .	18
<b>2</b>	<b>Lightweight Short-Term Current Prediction</b>	<b>21</b>
2.1	Introduction . . . . .	21
2.2	Overall Structure of Proposed Proactive Noise Mitigation . . . . .	22
2.3	Lightweight Short-Term Current Predictor . . . . .	23
2.3.1	Prediction Label Construction . . . . .	23
2.3.2	Prediction Feature Construction . . . . .	25
2.3.3	Predictor Engine and Implementation Cost . . . . .	27
2.4	Noise Mitigation Controller . . . . .	27
2.5	Experimental Results . . . . .	28
2.6	Extended Discussion on Out-of-Order Processor . . . . .	32
2.7	Conclusion . . . . .	33
<b>3</b>	<b>Low-Latency Voltage Scaling Using Major-Minor Voltage Regulator</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Scalable Major-Minor Voltage Regulator . . . . .	36
3.3	Experimental Results . . . . .	38
3.3.1	MMVR Performance Experiment . . . . .	39
3.3.2	Proactive Versus Reactive Noise Mitigation . . . . .	40
3.4	Conclusion . . . . .	42

<b>4 Frequency-Dependent Target Impedance Methodology</b>	<b>43</b>
4.1 Introduction . . . . .	43
4.2 Overall Flow and Basic Impedance Shapes . . . . .	44
4.3 Magnitude Equivalent Frequency (MEF) . . . . .	46
4.3.1 MEF for Capacitance Dominant Impedance . . . . .	46
4.3.2 MEF for Inductance Dominant Impedance . . . . .	47
4.4 Derive Target Inductance and Target Capacitance . . . . .	49
4.5 Experimental Results . . . . .	50
4.5.1 Target Impedance Synthesis for Experiment . . . . .	50
4.5.2 Experimental Results Compared with Design Constraints . . . . .	52
4.6 Conclusion . . . . .	53
<b>5 Chip Load Model for PDN Verification and Exploration</b>	<b>55</b>
5.1 Introduction . . . . .	55
5.2 Multi-Core Chip Load Modeling . . . . .	56
5.2.1 Overview of Chip Load Modeling Flow . . . . .	56
5.2.2 Target Multi-Core PDN System and Usage Model . . . . .	58
5.2.3 Individual Core Load Model . . . . .	59
5.2.4 Core Load Model Characterization . . . . .	66
5.2.5 Resistance Profile Simulation Procedure . . . . .	67
5.3 Experimental Results . . . . .	68
5.3.1 Individual Core Experiment . . . . .	68
5.3.2 Multi-Core PDN System Experiment . . . . .	71
5.4 Conclusion . . . . .	76
<b>6 Conclusion</b>	<b>79</b>
<b>Appendix</b>	<b>83</b>
<b>Bibliography</b>	<b>91</b>

# List of Figures

1.1	Processor frequency and transistor count in the past 50 years. . . . .	2
1.2	IRDS prediction of board power, device supply voltage, and device threshold voltage. . . . .	3
1.3	Overall diagram of a power delivery network (PDN) system. . . . .	5
1.4	PDN circuit model using lumped RLC components. . . . .	5
1.5	Refining PDN impedance to meet target impedance constraint. . . . .	8
1.6	Decoupling capacitors allocation at different stages of PDN to reduce the PDN impedance. . . . .	8
1.7	Power delivery network (PDN) with reactive noise mitigation system. .	9
1.8	Negative loop challenge of supply noise mitigation. . . . .	11
1.9	PDN design gap challenge. . . . .	13
1.10	Two voltage profiles with same maximum voltage drop. . . . .	14
1.11	Voltage-current-timing interdependency. . . . .	16
1.12	Existing challenges for supply noise mitigation and PDN design. . . .	17
1.13	Proposed solutions for proactive noise mitigation and PDN design methodology. . . . .	17
2.1	Proposed structure for proactive supply noise mitigation. . . . .	22
2.2	Training and prediction flows with current predictor. . . . .	24
2.3	Determination of averaging period $P$ using voltage-current correlation. .	25
2.4	Digital voltage sensor. . . . .	28
2.5	RMSE versus prediction length. . . . .	30
2.6	Correlation versus prediction length. . . . .	30
2.7	Current prediction results with DT and SVM. . . . .	32
3.1	Proposed voltage regulator in proactive supply noise mitigation system. .	36
3.2	MMVR connection diagram. . . . .	37
3.3	Major VR with 2:1 conversion ratio. . . . .	37
3.4	Minor VR in normal mode with 2:1 conversion ratio. . . . .	37
3.5	Minor VR in scaling mode with 3:2 conversion ratio. . . . .	37
3.6	MMVR performance test circuit. . . . .	39
3.7	Comparison in voltage scaling range. . . . .	40

3.8	Comparison in ripple voltage. . . . .	40
3.9	MMVR efficiency versus load voltage. . . . .	41
3.10	Noise mitigation result for multicore RISC-V PDN. . . . .	42
4.1	RLC target impedance. . . . .	44
4.2	RL target impedance. . . . .	44
4.3	Overall flow of frequency-dependent target impedance methodology. . . . .	45
4.4	RC test circuit. . . . .	49
4.5	RL test circuit. . . . .	49
4.6	RLC-type target impedance synthesis. . . . .	51
4.7	RL-type target impedance synthesis. . . . .	51
4.8	Load current profiles at 1 GHz for experiments. . . . .	52
5.1	Flow of multi-core chip load modeling. . . . .	57
5.2	An example of block diagram of power delivery network for multi-core system. . . . .	58
5.3	Overall structure of individual core load model. . . . .	59
5.4	Time voltage variant resistors model structure. . . . .	60
5.5	Comparison of equivalent resistance during clock switching. . . . .	62
5.6	Clock latency estimation comparison. . . . .	62
5.7	Parasitic impedance model. . . . .	65
5.8	Parasitic impedance extracted by small signal analysis. . . . .	65
5.9	Critical path replica model. . . . .	66
5.10	Current waveform comparison within one clock cycle. . . . .	69
5.11	Load voltage waveform comparison within one clock cycle. . . . .	70
5.12	Clock latency estimation with 100 MHz supply noise. . . . .	71
5.13	Clock latency estimation with 1 GHz supply noise. . . . .	72
5.14	Peak current estimation with 100 MHz supply noise. . . . .	72
5.15	Peak current estimation with 1 GHz supply noise. . . . .	73
5.16	16-core cluster with power-ground mesh. . . . .	74
5.17	16-core cluster with clock tree and control signal. . . . .	74
5.18	Cycle-by-cycle critical path slack comparison during transient process. .	75
5.19	Cycle-by-cycle critical path worst slack of core #6. . . . .	75
5.20	Worst timing slack under different LSC configurations. . . . .	76

# List of Tables

2.1	Instruction categorization for RISC-V. . . . .	26
2.2	Overriding rule table with sensor output. . . . .	28
2.3	Prediction performance and hardware cost. . . . .	31
4.1	Derived target impedance parameters, and average and minimal voltages. . . . .	53
5.1	Average peak load current and average clock latency comparison at various supply voltages. . . . .	70



# Abbreviations

<b>ALU</b>	Arithmetic Logic Unit
<b>CMOS</b>	Complementary Metal-Oxide Semiconductor
<b>CPU</b>	Central Processing Unit
<b>CRC</b>	Cyclic Redundancy Check
<b>CSR</b>	Control and Status Register
<b>decap</b>	Decoupling Capacitor
<b>DT</b>	Decision Tree
<b>DVFS</b>	Dynamic Voltage and Frequency Scaling
<b>EDA</b>	Electronic Design Automation
<b>EMA</b>	Exponential Moving Average
<b>FPGA</b>	Field-Programmable Gate Array
<b>FPU</b>	Floating-Point Unit
<b>FU</b>	Functional Unit
<b>GPU</b>	Graphics Processing Unit
<b>IC</b>	Integrated Circuit
<b>IoT</b>	Internet of Things
<b>IPC</b>	Instructions per Cycle
<b>IRDS</b>	International Roadmap for Devices and Systems
<b>ITRS</b>	International Technology Roadmap for Semiconductors

<b>LDO</b>	Low Dropout
<b>LSC</b>	Land Side Capacitor
<b>LUT</b>	Lookup Table
<b>MEC</b>	Magnitude Equivalent Current
<b>MEF</b>	Magnitude Equivalent Frequency
<b>MMVR</b>	Major-Minor Voltage Regulator
<b>OoO</b>	Out of Order
<b>PC</b>	Program Counter
<b>PCB</b>	Printed Circuit Board
<b>PDN</b>	Power Delivery Network
<b>PG</b>	Power Ground
<b>PSN</b>	Power Supply Noises
<b>RMSE</b>	Root-Mean-Square-Error
<b>RP</b>	Resistance Profile
<b>SCVR</b>	Switched Capacitor Voltage Regulator
<b>SMA</b>	Simple Moving Average
<b>SOP</b>	System-on-Package
<b>SPICE</b>	Simulation Program with Integrated Circuit Emphasis
<b>SSN</b>	Simultaneously Switching Noise
<b>SV</b>	Support Vector
<b>SVM</b>	Support Vector Machine
<b>VCCS</b>	Voltage-Controlled Current Source
<b>VLSI</b>	Very Large Scale Integration
<b>VR</b>	Voltage Regulator
<b>VRG</b>	Voltage Regulator Group

# Chapter 1

## Introduction

This dissertation focuses on proactive supply noise mitigation and power delivery network design methodology. This chapter describes the research background and objectives of this dissertation. Section 1.1 will explain the background of the power delivery network (PDN) and supply noise impact, followed by the introduction of traditional noise mitigation system and PDN design methodology. Then, Section 1.2 discusses the main problems in existing methodology. Finally, the objectives and overall organization of this dissertation are presented in Section 1.3.

### 1.1 Background

Moore's law has been driving the semiconductor industry for over 50 years. As is diagrammed in Fig. 1.1, the transistor count per die kept doubling in bi-annual pace [1–4]. Meanwhile, though clock frequency for single thread reached operation has reached a plateau around the year of 2006, it still keeps 15% to 20% increment per technology node generation [1,5]. Following the ITRS2.0 and IRDS prediction [6–9], to the 2030s, even for a low power mobile device, the number of GPU and CPU cores can increase 10X within a decade. Such a technology scaling trend pushes the future chip design to the power wall [3, 10, 11], because ever increased frequency and transistor count will eventually hit the physical limitation such as thermal dissipation limit and battery capacity limit. Hence, reduced power consumption becomes the major technology drivers for the coming decade [9].

To continue the performance improvement under the power wall, aggressive supply voltage reduction is a necessity, especially for low power devices such as mobile phones and IoT devices [12–14]. According to the IRDS prediction of board power (for mobile device), device supply voltage, and threshold voltage shown as Fig. 1.2, the supply voltage of chip core logic can be as low as 0.55 V level to the year of 2034. Considering the threshold voltage keeps above 0.2 V, and ever increased power consumption, the noise margin will be continuously decreasing for the coming technology node.

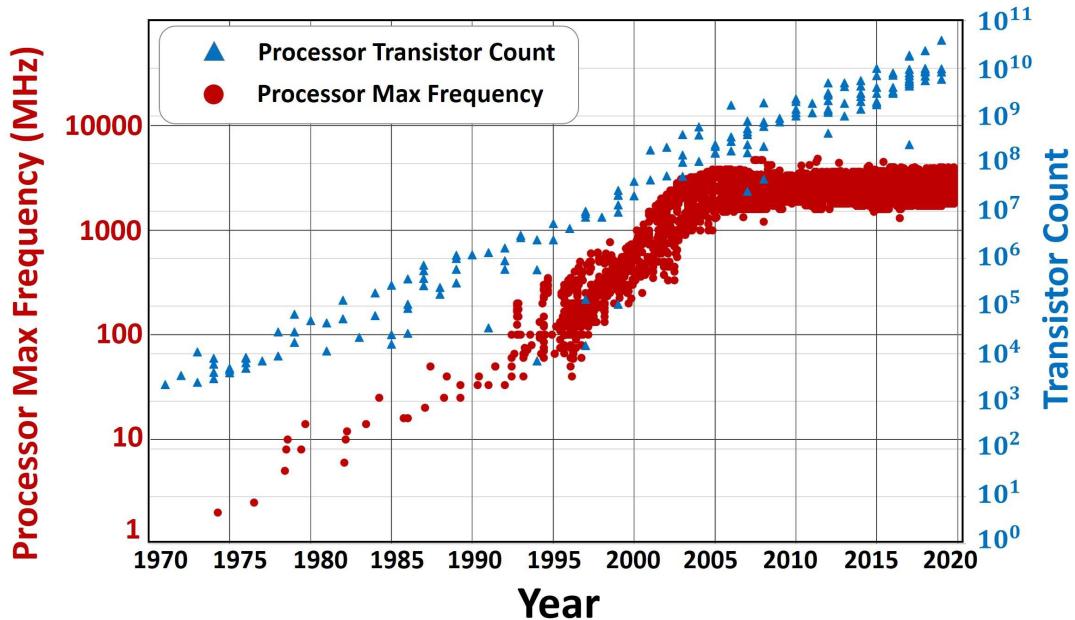


Figure 1.1: Processor frequency and transistor count in the past 50 years. Blue triangles represent the transistor count on a processor, and red dots represent the maximum frequency<sup>1</sup>.

A high-quality low-noise power distribution system is critically important to ensure the performance of next-generation very large scale integration (VLSI) system. It is because the supply noise magnitude is continuously increasing with the transistor numbers on die, and timing sensitivity to noise becomes more and more severe with the scaling down of the technology node. For example, Ahmed *et al.* [15] reported over 15% voltage drop in at-speed delay test on 180 nm SoC. At 55-nm node, the peak supply noise can reach 20%-30% of nominal voltage [16]. The multi-core system makes voltage droop even larger. Taking the worst voltage droop as an example, a dual-core system may experience 50% larger droop than a single-core system [17]. As for the noise-timing sensitivity, Saint-Laurent [18] reported over 7% timing impact under 1.3% VDD supply voltage noise after the 90-nm technology node. Reddi *et al.* [19] reported over 33% frequency loss due to 20% extra noise margin. Bhowmik [20] reported 2 MHz chip frequency degradation for every millivolt drop in a four-core processor. Gnad [21] reported over 3% timing delay increment caused by voltage drop, which is caused by toggling 8% of the flip-flops in the field-programmable gate array (FPGA). The supply noise challenge is thought to become severer at the even smaller node [22].

<sup>1</sup>Here, the processor data before 1995 is collected from [1], and data starts from 1995 is collected from [2]. The maximum frequency data is measured with integer benchmarks.

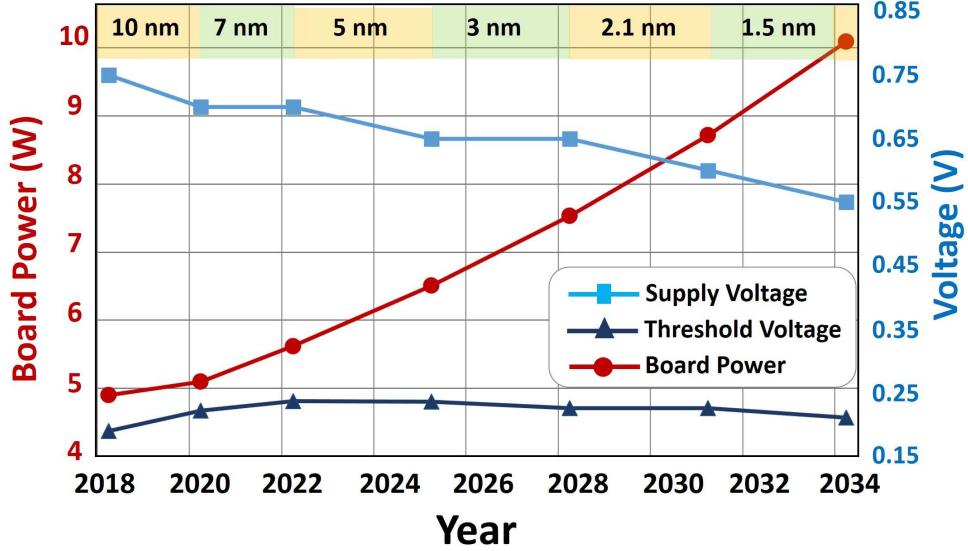


Figure 1.2: IRDS prediction of board power, device supply voltage, and device threshold voltage. The blue square line is predicted logic supply voltage, the blue triangle line is predicted device threshold voltage, and the red line is predicted board power for the mobile device. Each prediction step corresponds to technology node generation.

The discussion on supply noise mitigation and low-noise PDN design can be traced back to the 1970's. Till earlier 1990's, supply noise is mainly focused on the package level I/O noise, when Rainal [23] and Katopis [24] discussed the typical delta-I noise caused by inductive bonding wires. With the advancement of transistor integration, the impact of simultaneous switching noise (SSN) on CMOS I/O circuits is analyzed from the 1990's [25–27]. Though in this period, Davidson [28] reported the package level noise impacts the system-level performance by affecting propagation delay and clock skews, the supply noise interaction with chip internal behavior is still assumed to be a minor factor, and on-chip PDN is mainly modeled by simple lumped RLC components.

From the 1990's to 2000's, researchers began to analyze the detailed on-chip timing impact by supply voltage noise. Various on-chip PDN model is proposed, but exposing the detailed on-chip timing information considering supply noise is still a very difficult task [29]. During this period, Chen *et al.* [30] and Zhao [31] use RLC grid to model the detailed on-chip PDN noise, Eo *et al.* [32] and Tang [33] reported the SSN becomes an important issue for VLSI PDN design, Garben [34], Zhou *et al.* [35], and Ahmad *et al.* [36] discussed the interaction between on-chip PDN noise and off-chip packages at resonant frequencies, and the measurement of noise impact on multi-core and 3D-IC chips are also conducted. During this period, the on-chip noise impact is usually modeled by RLC grid and distributed current sources. Meanwhile, PDN design mainly

relies on voltage guard bound methodology for worst-case voltage drop. Decoupling capacitors are widely used to mitigate the noise impact at PDN design stage [37–40].

In 2000’s, Rahal-Arabi *et al.* [41, 42] firstly reported that the actual noise impact on the chip performance is much more complex than the previous assumption since conventional worst-case voltage guard bound methodology can lead to significantly over-designed PDN. As presented in Rahal-Arabi’s experiment, in some scenarios, removing the on-chip decoupling capacitor (decap) cells can even improve the chip performance. Several explanations are proposed for this issue. Chen *et al.* [43] points out the over-inserted decap cells significantly increase the tunneling current and leakage power, Hashimoto *et al.* [44], Ogasahara *et al.* [45], and Azais *et al.* [46] found the averaged noise has higher impact on logic and timing path, and actual timing impact is related to many aspects such as library cell sensitivity, temporal and spatial characteristics of supply noise.

Meanwhile, the supply noise becomes a critical concern with the popularity of low power designs. Though off-chip voltage regulators can be modulated for dynamic voltage scaling purposes [47–52], they are not suitable for mitigating emergent supply noise because of the slow response time. Instead, on-chip low-dropout (LDO) voltage regulator [53–57] is commonly used to mitigate on-chip supply noise. Other researchers try to dynamically schedule core activation [58], schedule workloads [59], modulate clock toggling phase [20], or exploit clock-data compensation effect [60, 61] to mitigate the supply noise impact. These traditional noise mitigation systems are basically designed in reactive flavor, which relies on the sensor to detect the voltage drop and then trigger noise mitigation.

Recently, with the rising of machine learning technology, various new methods are proposed for resolving traditional PDN design and noise mitigation problems. For example, in [62–66], machine learning is applied to optimize library cell selection, allocate decoupling cells, and localize the worst supply noise region on the power ground mesh. Meanwhile, with the improvement of chip package integration technology, the on-chip voltage regulator becomes a feasible power supply solution. For example, Gu [67] and Wang *et al.* [68] use on-chip SCVRs for VLSI fine-grained voltage regulation purpose.

This dissertation proposes proactive noise mitigation system which is inspired by these emerging technologies. Before the detailed discussion, the following subsections will present the basics of the power delivery network, supply noise, PDN design methodology, and typical noise mitigation system.

### 1.1.1 Power Delivery Network and Supply Noise

This subsection presents the basics of traditional power delivery network (PDN) system, and power supply noise source throughout the PDN system.

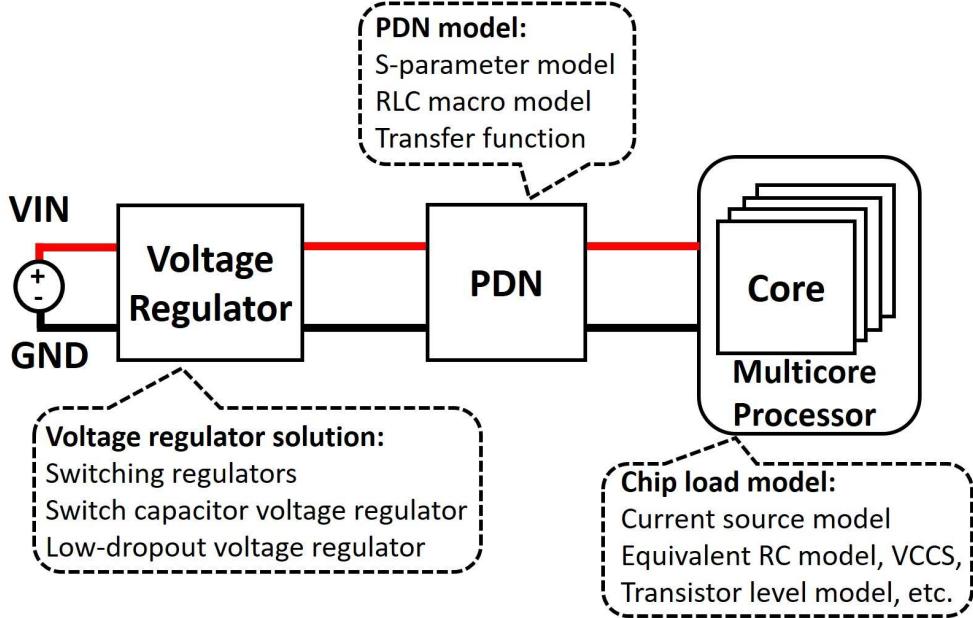


Figure 1.3: Overall diagram of a power delivery network (PDN) system. The red wire is power line, and the black wire is ground line.

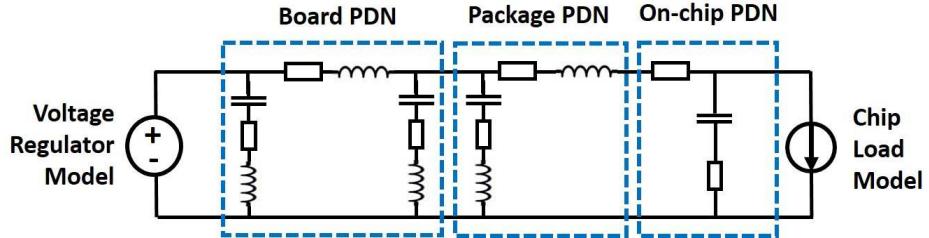


Figure 1.4: PDN circuit model using lumped RLC components.

### Power Delivery Network Structure

The overall PDN system structure used in this dissertation is well discussed in [22, 69–74], and can be diagrammed as Fig. 1.3, where the system is roughly divided into three components. Voltage regulator (VR) component serves as the power supplier, which converts higher DC voltage, usually from DC source or battery, to the lower DC voltage for VLSI use. The VR solution includes switching regulator (buck) [75, 76], low-dropout (LDO) voltage regulator [53–57], switched capacitor voltage regulator (SCVR) [48–52, 67, 77, 78], or combination with these solutions [56, 57, 79–81].

The role of the PDN component is to distribute supply voltage from the voltage regulator component to chip circuit elements such as logic gates and flip-flops(FFs) [82]. In a VLSI design, PDN circuit mainly corresponds to PCB and package. It usually consists

of passive components, and they are typically modeled by S-parameter model [83–86], RLC macro model [87, 88], or transfer function based methods [89–91]. An example PDN circuit model is shown as Fig. 1.4, where the conductors of the board, package, and on-chip circuit are modeled by lumped resistors and inductors. The decoupling capacitors between the power line and ground line are represented by series RLC components.

The chip load component can be modeled by current source model [92, 93], equivalent RC model [30, 94, 95], voltage-controlled current source model (VCCS) [96], or transistor-level SPICE model [97]. A typical PDN design task is to use the above models to assess the supply voltage noise for the chip load, then following the target impedance methodology [38–40] to refine the PDN impedance.

## Supply Noise Source

Power supply noise can arise from various parts of PDN. These noise sources include voltage regulator component, PDN component, and chip load component. Jointly considering the noise sources and mitigating the noise impact are usually a complex and challenging task.

Firstly, the voltage regulator component can induce supply noise. For the switching regulator, load current fluctuation can cause a large voltage drop across the inductor component. Hence, a large decoupling capacitor is necessary to suppress the output supply noise [74, 98]. For the switched capacitor voltage regulator (SCVR), during the SCVR operation, the flying capacitor is charged and discharged periodically, which results in voltage ripple at the VR output port [99–101]. Though the output ripple or VR output supply noise can be reduced by increasing the switching frequency or using a larger flying capacitor [78], the high switching frequency can degrade the power conversion efficiency, and the large flying capacitor can cause longer response time for voltage scaling.

Secondly, the PDN component causes voltage fluctuation. The PDN component noise mainly consists of IR drop and  $L(di/dt)$  drop, where R and L are equivalent resistance and inductance from the load side of PDN. The IR drop noise is proportional to the current drawn by the chip load. For high-performance chip design, the large number of simultaneously switching cells can cause considerable  $L(di/dt)$  drop and even dominate the supply noise [102, 103]. To reduce the PDN component noise, designers need to reduce PDN impedance by optimizing PDN circuits such as power ground mesh, power pads, and device package. The maximum allowed target impedance is derived to guide this process [40]. However, target impedance methodology, which will be explained in the next subsection, is increasingly difficult to fit with modern VLSI design [69], and can result in over- or under-designed PDN.

Thirdly, intra-core activity variation and inter-core interference can induce supply noise [17]. For example, simultaneous activity variation such as power-on or wake-up can happen on the individual cores and then, induce a significant local voltage droop

that can propagate to adjacent cores. Such a droop may reach 150 mV and may easily exceed the voltage guard bound [17, 104]. If a signal is propagating on a critical path in the victim core, the voltage droop causes extra path delay [105], or result in malfunction [106]. The inter-core noise-timing impact is even severer if the core is located far from power supply ports, such as at the center of shared power and ground mesh [107]. In another scenario, if adjacent cores stay in retention mode or idle mode, the parasitic capacitance in those cores can be used to mitigate the noise and consequent timing impact [108].

Finally, the above factors need to be considered jointly since each component is interdependent with each other. Hence, noise mitigation for a PDN system becomes a very difficult and complex problem. This dissertation will divide and conquer the problem in two aspects. The first aspect is the low noise PDN design methodology. The second aspect is the noise mitigation system. The background of these two aspects are presented in Section 1.1.2 and Section 1.1.3, respectively.

### 1.1.2 Low Noise PDN Design Methodology

Low noise PDN design is highly demanded for robust high-performance chip design. As is reported in [109], chip operation frequency can be improved by reducing the PDN impedance. Meanwhile, power consumption is a key concern of high-performance chip design. A low noise PDN design can reduce the overall power consumption, which in turn, decreases the hardware resource cost such as the number of power and ground pads [75, 110]. Therefore, for large scale and heterogeneous systems such as system-on-package (SOP) architectures, the requirement for sophisticated low-noise PDN will increase [22].

As a common practice, PDN design is based on target impedance methodology, which was first proposed by Smith *et al.* [40] in the 1990's. The basic idea of this methodology is to define an upper bound of PDN impedance, which is the target impedance. Target impedance  $Z_{target}$  can be defined as:

$$Z_{target} = \frac{V_{max\_drop}}{I}, \quad (1.1)$$

where  $V_{max\_drop}$  is the maximum allowable voltage drop, and  $I$  is the current requirement. In some works, the maximum allowed voltage drop is also noted as voltage guard bound, power supply tolerance, or noise margin. Depending on the design feature,  $V_{max\_drop}$  ranges from 5% to 10% of nominal voltage [40, 111]. The noise margin requirement is increasing with the advancement of technology and design complexity. For example, the necessary noise margin can reach 20%-30% of nominal voltage for 45nm and below IC chip design [16]. The current requirement  $I$  is selected as 50% of peak switching current [40], or maximum averaged current [69]. According to (1.1), if refined PDN impedance is smaller than  $Z_{target}$ , then the maximum voltage drop will be smaller than design constraints  $V_{max\_drop}$ .

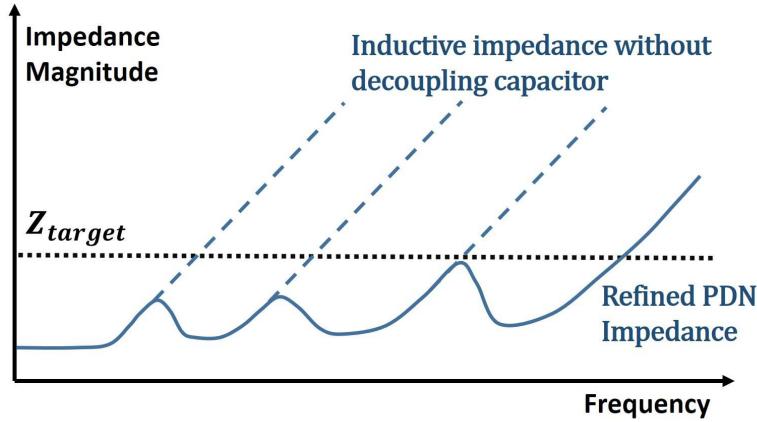


Figure 1.5: Refining PDN impedance to meet target impedance constraint.

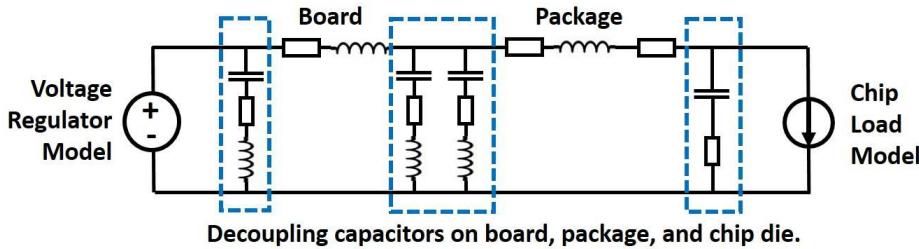


Figure 1.6: Decoupling capacitors allocation at different stages of PDN to reduce the PDN impedance.

The design flow of PDN impedance can be demonstrated as Fig. 1.5, where the target impedance magnitude is shown as the black dot line. PDN impedance magnitude is shown as the blue line. Since the impedance of board and package will be inevitably dominated by inductive impedance in high-frequency range, the original PDN impedance magnitude can exceed the target impedance, which is shown as dotted lines in Fig. 1.5.

The PDN impedance can be effectively reduced by allocating decoupling capacitors. This method was introduced in the early 1990's for mitigating the delta-I noise on the VLSI package I/O [37]. Later, various methods are proposed for allocating on-chip and off-chip decoupling capacitors considering different design constraints such as area, hardware cost, and maximum allowed voltage drop [38–40, 112, 113]. An example of decoupling capacitors allocation is depicted in Fig. 1.6, where the decoupling capacitor models are surrounded in the blue boxes. PDN designers need to consider design constraints and select various decoupling capacitors so that PDN impedance is lower than the target impedance from DC to at least the first harmonic of the clock frequency.

The target impedance methodology builds a simple, but over-simplified relation-

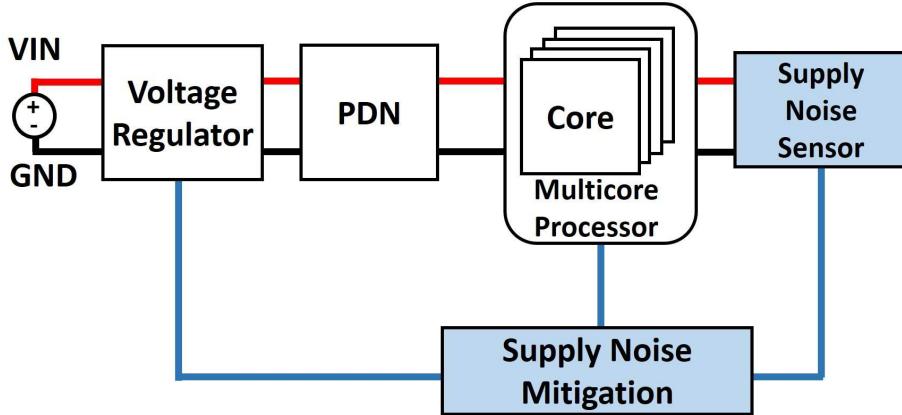


Figure 1.7: Power delivery network (PDN) with reactive noise mitigation system. Red wire is power line, black wire is ground line, and blue wire is control signal line.

ship between time-domain voltage drop constraints and frequency-domain PDN design guidance, and conveys a blind belief that, if maximum voltage drop is above the voltage guard bound, the chip load performance is ensured. However, such a belief usually causes over- or under-designed PDN. The details will be described in Section 1.2.2.

### 1.1.3 Supply Noise Mitigation System

With the trend of ever-increasing power consumption and decreasing supply noise margin shown as Fig. 1.2, the supply noise mitigation system becomes an important component for modern low power designs. The typical noise mitigation system is commonly designed in reactive flavor shown as Fig. 1.7, where the blue boxes represent the components of noise mitigation system including noise sensor component, and noise mitigation component. The noise mitigation system firstly measures the supply noise, and then, the measured result is sent to the noise mitigation component, which modulates various PDN components to mitigate supply noise. For the noise sensing part, compared with off-chip noise sensor structure [70, 74], on-chip sensor structures such as in-situ monitor [114–118] and on-chip path monitor [71, 72, 119, 120] have shorter response time.

For the noise mitigation component, off-chip voltage regulators can be modulated for dynamic voltage scaling purposes [47–52], but their response time is too slow compared with emergent supply noise in VLSI which can occur within tens of clock cycles. Low-dropout (LDO) voltage regulator [53–57] is commonly applied to suppress the supply noise, because LDO has simple structure and fast response time. Besides, Paul *et al.* [58] try to stagger the multi-core activation and achieved 10% less voltage droop. However, this method typically has over 100 ns transient process, which is not suitable for emergent noise droop. Meanwhile, staggering activation can cause even

larger voltage droop if the core activation is scheduled at the undershoot period of load voltage resonance [20]. Lam *et al.* [121] and Kaplan [122] intentionally schedule the clock skew to reduce the simultaneous switching cell count and hence, reduce peak supply voltage noise. Kim [60] exploits the clock-data compensation effect to relieve the voltage scaling requirement. Fan [61] applies the two-phase clock tree to reduce the simultaneous switching noise (SSN). Bhowmik [20] combines the staggering activation and frequency scaling to mitigate supply the noise. However, these methods require detailed on-chip clock and timing path design, which introduces additional design complexity and cost for the overall PDN system. Other researchers [123–125] use an analog amplifier to compensate supply noise. However, their methods still suffer from large area overhead or slow response time.

Though various methods are proposed for the reactive noise mitigation system, these methods often fail to compensate for emergent supply noise due to the long latency of PDN voltage boosting. To conceal such latency, proactive noise mitigation system using voltage scaling is studied in this dissertation. The details are described in Section 1.2.1.

## 1.2 Major Challenges for Robust Power Distribution

This section summarizes the major challenges for the existing PDN design methodology and supply noise mitigation system. This dissertation will focus on two major challenges for low-noise PDN design and supply noise mitigation system. The first challenge is related to supply noise mitigation. Existing long-term prediction requires high computation cost and consequently longer computation latency, which makes further longer-term prediction requirement. The challenge of negative noise mitigation loop makes proactive noise mitigation less effective. The second challenge is related to PDN design methodology. The traditional target impedance requirement is increasingly difficult to meet, and the timing impact is still invisible for off-chip PDN designers. These design gaps challenge can lead to over- or under- design PDN. The details and related works for these two major challenges are described in the followings.

### 1.2.1 Negative Loop Challenge of Supply Noise Mitigation

The negative loop challenge of noise mitigation is diagrammed in Fig. 1.8, where the original PDN system is shown in white boxes and bounded by dotted lines. Red arrows show the control flow of the proactive noise mitigation system, which consists of two steps marked in orange boxes. Firstly, noise prediction is performed, and then the noise mitigation decision is deduced and sent to the voltage scaling step to compensate for the incoming supply noise. In proactive mitigation system, the noise mitigation is performed before actual noise happens. The leading time of mitigation is denoted as prediction length, namely how far future is predicted. Meanwhile, the noise mitigation is usually delayed due to systematic issues such as voltage sensing latency or

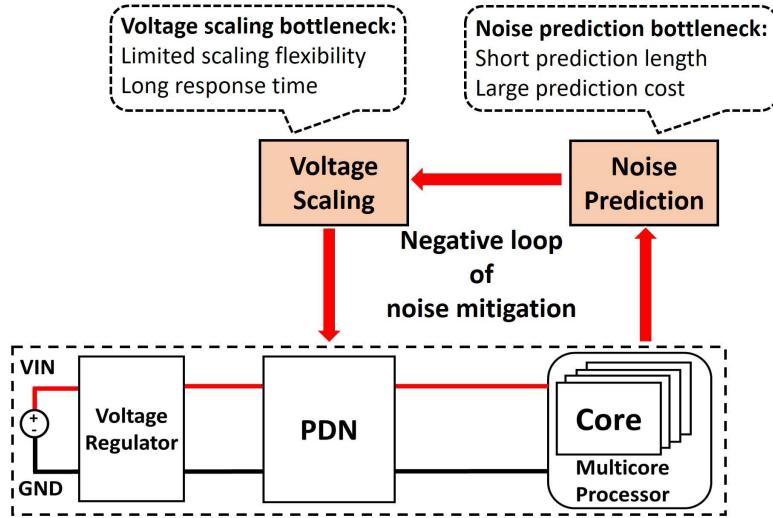


Figure 1.8: Negative loop challenge of supply noise mitigation.

voltage boosting latency through PDN. The delayed time is denoted as PDN latency. To proactively mitigate the noise, the prediction length should be longer than PDN latency. However, there are bottlenecks in both noise prediction step and voltage scaling step, which result in long PDN latency and short prediction length, and hence form a negative design loop challenge.

### Noise Prediction Bottleneck

The first bottleneck exists in the noise prediction step. Proactive noise mitigation relies on accurate prediction with low hardware and computational cost, and the prediction is supposed to be deduced from hardware signal switching events [126]. In [127–129], power, voltage drop, and timing delay prediction are studied. These studies commonly use internal hardware signals such as control signal and pipeline status registers as input features, and use linear kernel support vector machine (SVM) as the prediction engine.

However, directly monitoring a large amount of chip internal signals can result in overwhelming hardware overhead. Even though hardware signal features are carefully selected using correlation analysis, the prediction length reaches only 16 cycles [129]. Meanwhile, accurate SVM prediction is often achieved with kernel functions and a large number of support vectors, which means high computational cost for floating-point multiplication and addition over many support vectors. This expensive computation causes large hardware overhead and longer computation time of over 40 cycles [127], which requires even longer prediction length. Thus, a negative design loop arises for noise mitigation system.

## Voltage Scaling Bottleneck

The second bottleneck exists in the voltage scaling step. Proactive noise mitigation requires quick and continuous voltage scaling with a wide scaling range and small voltage ripple. However, this requirement is difficult to meet by using traditional voltage regulator solutions such as switching regulator, LDO, and SCVR.

The switching regulator solution can achieve 80% to 90% efficiency with simple structure [130]. However, the inductor component and decoupling capacitor component cause very long transient time, which makes low-latency voltage scaling infeasible. Lin *et al.* [131] try to integrate switching regulator on SoC to scale the load voltage. However, the inductor and capacitor components in this work introduce over  $10\ \mu\text{s}$  voltage scaling latency, which is too long to mitigate emergent VLSI supply noise. The LDO solution is widely adopted for high-performance VLSI design because it has a simple structure, small area overhead, and quick response time [53–55]. However, LDO can only scale down the voltage at the cost of heat generation and energy loss. Besides, LDO drops out voltage using a resistor, and hence LDO efficiency degrades when the voltage scaling range is large.

The SCVR solution is promising for modern VLSI design, since the major advantage of SCVR is high efficiency and high integration capability with chip package. SCVR for VLSI solution is firstly proposed by Dickson [77]. Gu [67] introduced on-chip digital SCVR to stabilize supply voltage. Wang *et al.* use on-chip SCVR in multi-core SoC implementations to achieve fine-grained DVFS [132].

However, SCVR has limited voltage scaling flexibility due to the fixed conversion ratio, which is not desirable for wide-range continuous voltage-scaling purpose [133–135]. Besides, SCVR can introduce supply voltage ripples during the flying capacitance charge and discharging process. The ripple issue can be suppressed by using a large flying capacitor or increasing the switching frequency. However, the high switching frequency can degrade the power conversion efficiency, and a large flying capacitor pushes the SCVR away from chip load, resulting in slow voltage scaling response time. To mitigate the ripple, Jevtic [136] uses on-chip frequency scaling to cancel the ripple impact. Breussegem and Steyaert [49], and Lu [52] use multi-phase SCVR to reduce ripple. However, these efforts cannot improve the voltage scaling capability of SCVR.

To improve the voltage scaling capability of SCVR, researchers try to introduce the low-dropout (LDO) voltage regulator as a secondary linear regulator [56, 57, 79–81, 137, 138]. However, the low energy efficiency problem for LDO is not desirable for low power devices. Other researchers try to improve the voltage scaling capability by proposing various SCVR structures. For example, Eireiner [116] and Pillonnet [139] use multi-VDD SCVR to achieve voltage scaling. Souvignet [50], Andersen [51], and Nguyen [78] use multiple SCVRs with different conversion ratio to switch between different output voltage levels. However, these solutions only provide discrete voltage levels, and multi-SCVR solution can introduce over 100 mV ripple during voltage level during transient switching process [48]. Andersen [51] and Jiang [140] try

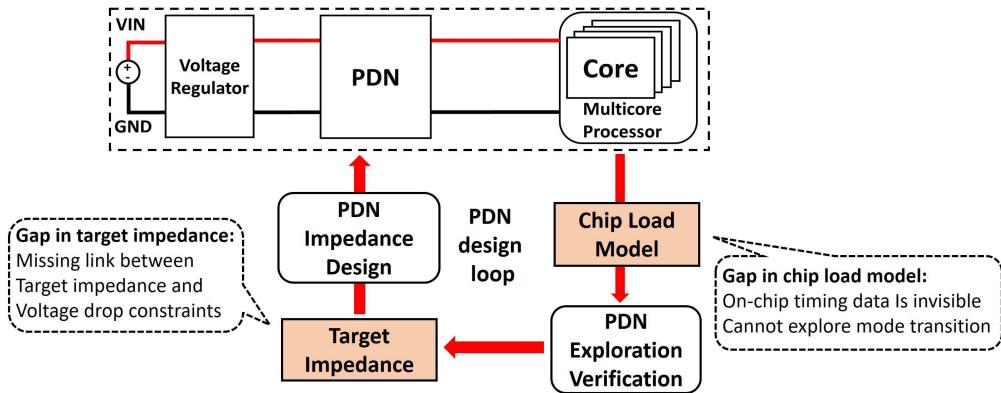


Figure 1.9: PDN design gap challenge.

to scale the voltage using multiple-conversion-ratio SCVRs or re-configurable SCVR. However, when dynamically switching the conversion ratio, the output ripple can be beyond 70 mV, which is 8.2% of load voltage [100]. Researchers also use multiple SCVRs to implement recursive SCVR structure [47, 141–144]. By dynamically reconfigure the connection between SCVRs, different output voltages can be obtained. However, recursive SCVR requires complex switching control circuits to avoid short current and to ensure the correct functionality of switches, resulting in extra hardware cost and efficiency loss, yet the output voltage ripple during voltage scaling is not well handled. Bang [101] tries to dynamically adjust flying capacitors according to load current, but this solution causes extra hardware cost to allocate many tiny capacitor groups.

In summary, SCVR has the advantage of high efficiency and high power density, though the voltage scaling and ripple issue are still open problems for traditional SCVR structure. This dissertation will use SCVR as a base structure for voltage scaling solution and overcome the limitations.

### 1.2.2 Gap Challenge in PDN Design Methodology

Traditional PDN design methodology usually results in under- or over-designed PDN, because there exists a gap challenge in the traditional PDN design loop. The design gap challenge can be diagrammed by Fig. 1.9, where the original PDN system is shown in white boxes and bounded by dotted lines. Red arrows show the PDN design loop and the design step with gap issues marked in red boxes. In this diagram, PDN designers firstly derive target impedance as design guidance. Then, PDN impedance is refined so that the target impedance constraint is satisfied. To verify and explore the PDN design under various chip load operation scenarios, a chip load model is needed to expose the impact of supply noise. PDN designers will further refine the PDN circuit according to the noise impact. Such a design loop can iterate several rounds to achieve the balance between PDN performance and cost.

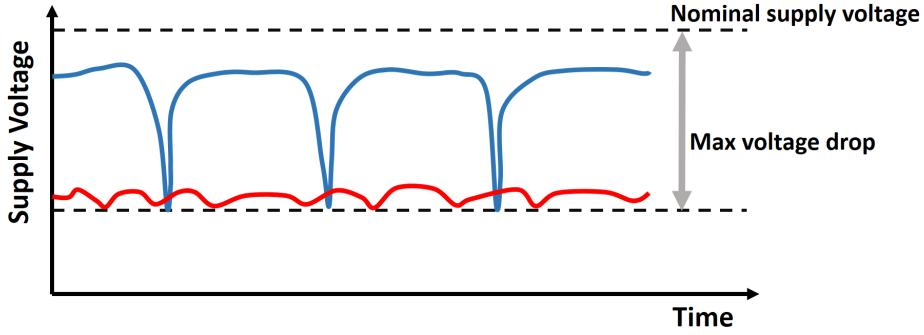


Figure 1.10: Two voltage profiles with same maximum voltage drop.

However, in the target impedance design stage, there lacks a proper method to bridge the time-domain voltage drop constraints and frequency-domain target impedance guidance. Meanwhile, the existing chip load modeling method cannot provide on-chip timing information within a reasonable simulation time and hence, timing impact is invisible to PDN designers. Such gaps can result in under- or over-designed PDN, and unexpected supply noise impact.

### Gap between Target Impedance and PDN Design Constraints

The first gap exists between target impedance and PDN design constraints, and ignoring the gap can result in under- or over-designed PDN. Such a gap problem arises because actual PDN impedance is defined in the frequency domain while the current profile and voltage drop constraint are given in time domain. Although the current spectrum tells us that dynamic power noise distributes within a certain frequency range, how to determine detailed frequency-dependent target impedance remains an open difficult problem. Besides, the average voltage drop constraint is not well handled in traditional target impedance, while the average drop can have a greater impact on chip performance than dynamic noise [18, 44, 94].

Let us take the voltage profiles in Fig. 1.10 to illustrate the gap impact. Here, given a load current profile, suppose two PDNs having different target impedance that satisfy the same maximum voltage drop constraint. Two voltage profiles corresponding to the different PDNs are depicted in red and blue. The red profile has lower average voltage and smaller ripple, which means chip performance is lower and the PDN for the red profile is over-designed in high-frequency range but under-designed in the low-frequency range.

To address this gap issue, researchers [145–150] try to approximate the time domain current profile as triangle or ramp so that the delta- $I$  noise, or  $L(di/dt)$  noise becomes a constant value and the PDN design flow is simplified. However, such approximation methods suffer from the fact that real current waveform may not be easily simplified to

the simple ramp or triangle shape. Oh *et al.* [151] use the current spectrum for deriving frequency-dependent target impedance. However, the constraint of the worst voltage drop, which is defined in the time domain, is difficult to convert into the frequency domain. Without a clear interpretation between the time domain and frequency domain, PDN designers have to rely on empirical methods such as iteration over the various resistor and capacitor configurations [152, 153].

### Gap between On-Chip Timing and Off-Chip PDN Verification Exploration

On-chip timing information is the primary metric in digital chip design. However, such timing information is usually invisible to off-chip PDN designers, which results in a design gap between on-chip timing and off-chip PDN design processes. As is discussed in [41, 42, 44–46], the worst voltage drop does not necessarily reflect the actual worst timing delay. Meanwhile, timing sensitivity to noise becomes severer with the scaling down of the technology [18–21], and therefore, ignoring such gap can mislead the PDN design.

The design gap problem is difficult to address because there lacks a simple yet accurate chip load model, which can expose the on-chip timing information while considering the voltage-current-timing interdependency and operation mode transition. The interdependency conception can be demonstrated in Fig. 1.11. In actual circuits, the supply noise interacts with chip timing performance such as clock latency and path delay [154, 155]. When supply voltage drops, signal propagation is delayed, clock latency gets longer, and the transistor switching current becomes smoother and smaller. When the load current becomes smaller due to the supply voltage drop, the dynamic noise becomes smaller, and its impact is naturally mitigated. However, simplified models such as the current source model in the piecewise linear format are irrelevant to voltage variance, and hence the supply noise is likely to be overestimated.

As for the operation mode transition requirement, in multi-core designs, there are many combinations of mode transitions. Also, their transition timings could affect noise magnitude and timing performance. To efficiently explore the impacts of modes and their transitions, the chip load model should have an interface that can easily and flexibly manipulate the operation modes of individual cores, which contributes to finding unexpected noise and consequent timing behaviors.

To fill the PDN-chip gap, researchers [16, 17, 120, 156–159] proposed various on-chip measurement modules that are used in the post-silicon validation stage. Modules such as critical path replica [157, 159, 160] and critical path monitor [120, 132] are developed to measure chip internal timing information. However, the inherent limitation of the post-silicon methodology is the silicon resource cost and the difficulty in design modification due to the late feedback. On the other hand, the pre-silicon simulation requires no silicon resource and provides feedback in design time. To perform the simulation, meanwhile, a chip load model that represents the chip behavior from the point of view of load current is necessary. The chip load model that consists of the on-chip

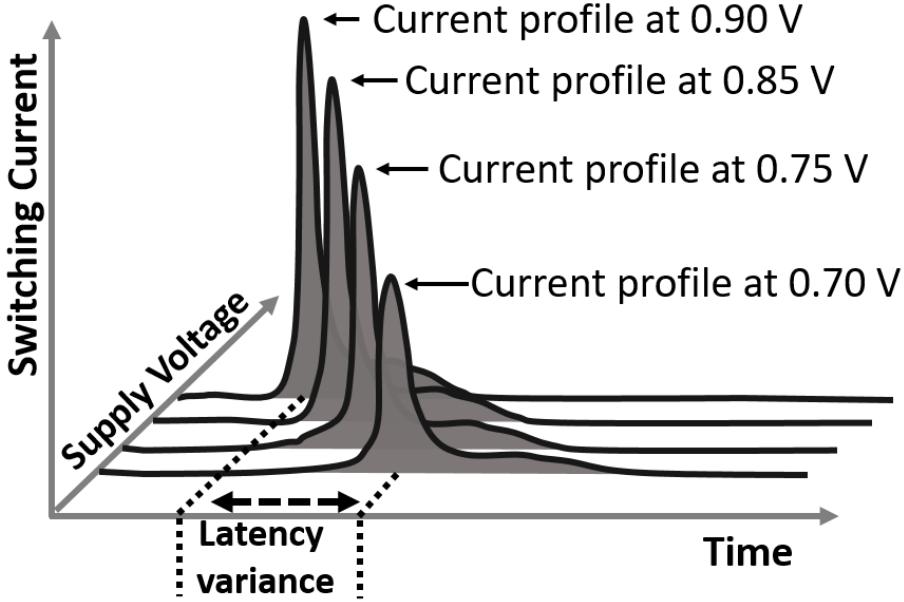


Figure 1.11: Voltage-current-timing interdependency.

PDN model and full transistor-level switching circuit model can replay the on-chip behavior with high accuracy. However, even a very short period run takes days or even months to finish. Extensive PDN design exploration is infeasible. To reduce the computational cost for the chip load model, the switching circuit is often modeled by a current source [92, 93] or equivalent RC circuit models [30, 94, 95]. Cui prepares multiple current profiles and manually switches the profile for different operation modes [92]. However, the current source model is usually described with a current profile in a piecewise linear format. Once a current profile is obtained under a given supply voltage, these piecewise linear current values are irrelevant to supply voltage variation. Hence, a large simulation error is introduced when the actual supply voltage has a significant dynamic supply noise. The current source can be also modeled by voltage-controlled current source (VCCS) [96, 97] to take into account the dependence of current on voltage. However, VCCS relies on instant voltage-current scaling, which is not suitable for replaying temporal behavior. On the other hand, RC circuit model can roughly model the voltage-current interdependency. This modeling method uses variant resistors, typically implemented by VCCS, to mimic the equivalent resistance of on- and off-state transistor. Then parasitic capacitors are characterized to mimic cell transition delay. However, even with careful characterizing effort on RC parameters, the over-simplified RC model is difficult to replay a detailed current profile for large-scale circuit operation.

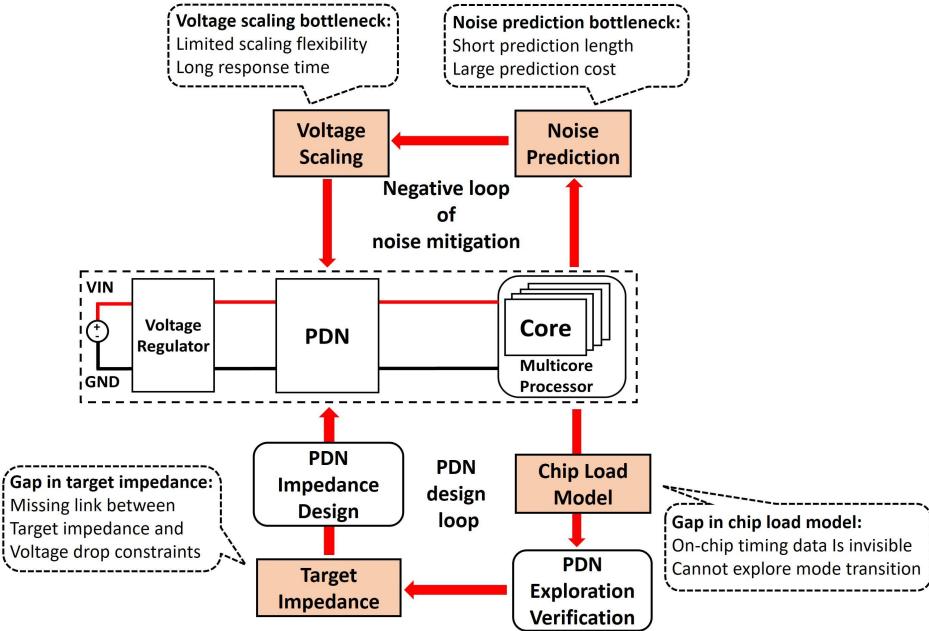


Figure 1.12: Existing challenges for supply noise mitigation and PDN design.

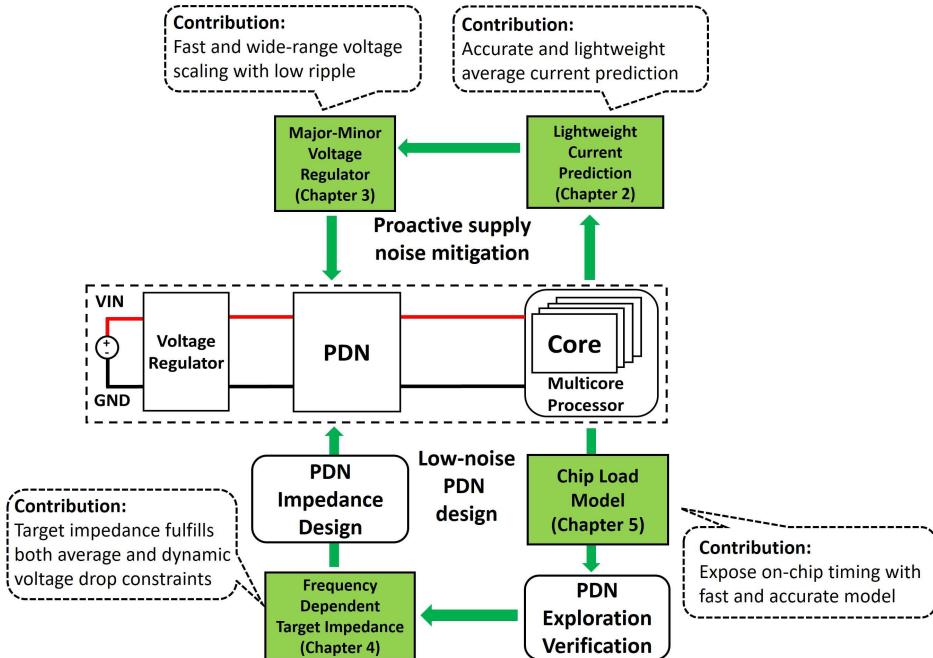


Figure 1.13: Proposed solutions for proactive noise mitigation and PDN design methodology.

## 1.3 Objective and Organization

The overall objective of this dissertation is to robustly provide the low-noise supply voltage through a VLSI power distribution system. To achieve the objective, this dissertation proposes a proactive noise mitigation system, and improves the PDN design methodology.

There are two main challenges and four problems in existing noise mitigation system and PDN design methodology. They are discussed in the previous Section 1.2 and summarized as Fig. 1.12, where the negative loop challenge is shown in the upper part, the design gap challenge is shown in the lower part. The main problems for each challenge are marked in the orange box. The contributions, proposed methods, and overall organization of this dissertation are diagrammed in Fig. 1.13, and each solution is marked in the green box. The proposed methods in Chapter 2 and Chapter 3 are used to construct the proactive noise mitigation system. The proposed methods in Chapter 4 and Chapter 5 aim to improve PDN design methodology.

The first contribution in Chapter 2 is to provide a lightweight current prediction solution, which is to solve the noise prediction bottleneck. The main difficulty here is to predict the near future noise with high accuracy and reasonable hardware overhead. The key idea to achieve this objective is to construct a lightweight short-term average current predictor using decision tree regressor. The decision tree regressor uses the instruction history of the processor as the input feature and averaged current as the prediction label. Based on the training result in the experiment, a six-layer decision tree predictor is implemented, and it achieves 50-cycle prediction length and over 0.99 correlation.

The second contribution in Chapter 3 is to provide a major-minor voltage regulator (MMVR) structure, which is to provide the fast and wide-range voltage scaling capability to solve the voltage scaling bottleneck. The main difficulty here is that traditional SCVR is not suitable to continuously scale the voltage because of the fixed conversion ratio. The key idea to overcome the difficulty is to propose a new major-minor voltage regulator (MMVR) structure, which consists of two SCVRs whose flying capacitance is much different. Major voltage regulator uses large flying capacitance to provide stable low ripple supply voltage. On the other hand, the minor voltage regulator is designed as a re-configurable SCVR structure, which can provide two different load voltage levels with small flying capacitance. This special structure enables minor voltage regulator to continuously scale supply voltage using simple switching frequency modulation. Meanwhile, a small flying capacitance means the capacitors in the minor voltage regulator can be integrated into a chip package to speed up voltage scaling. According to the experiment, MMVR achieved over 3X voltage scaling range compared with traditional SCVR while the ripple is within 16 mV, which is 1.6% of load voltage.

The third contribution in Chapter 4 is to provide a frequency-dependent target impedance methodology, which is to fill the gap between PDN voltage drop constraints and frequency domain impedance guidance. The main difficulty here is to bridge the

time domain voltage drop constraints and current profile with the frequency domain impedance curve. The key idea is to design the target impedance by introducing a new conception of magnitude equivalent frequency (MEF). That is, instead of analyzing the detailed time-domain current waveform, a sine waveform current can be used to reproduce the same magnitude of the voltage noise. The frequency of this sine waveform is defined as MEF. The conception of MEF can bridge the design gap and hence simplify the frequency-dependent target impedance design. The experiment confirmed that the synthesized target impedance satisfied the constraints with less than 0.1% error in the actual processor load case.

The fourth objective in Chapter 5 is to propose a chip load model that can provide the on-chip timing information, which is to fill the gap between off-chip PDN design and on-chip timing information. The main difficulty here is to construct a fast and accurate load model that can consider the voltage-current-timing interdependency and operation mode transitions. The key idea is to use the time-voltage-variant resistor to reproduce voltage-dependent load current taking into account voltage-dependent switching delay for a given operation mode. Then, multiple time-voltage-variant resistors are enabled or disabled by control logic interface so that mode transition is triggered. Critical paths are represented by the critical path replica module to replay critical path timing delay. Also, parasitic and intrinsic decoupling capacitances are modeled using small-signal analysis. Hence, the global and local clock latency, skew, and path delays can be computed with simulation. The experiment confirmed that the proposed chip load model achieves better correlation compared with the traditional current source based model and RC based model, while over 300X runtime reduction is achieved compared with full SPICE netlist simulation. The off-chip PDN modification experiments show the proposed model can guide off-chip PDN designers with on-chip timing information.

The rest of this dissertation is organized as follows. Chapter 2 presents the proactive noise mitigation system consisting of lightweight near-future current prediction and noise mitigation controller. The prediction is achieved by applying machine learning technology. Chapter 3 proposes a major-minor SCVR structure to achieve fast and wide-range voltage scaling. Chapter 4 proposes a frequency-dependent target impedance methodology to guide PDN design. The methodology considers both dynamic voltage drop and average voltage drop constraints. Chapter 5 presents chip load model which can expose the on-chip information for PDN verification. This capability is achieved by considering voltage-current-timing interdependency and operation mode transitions of chip load. Lastly, concluding remarks are given in Chapter 6.



# Chapter 2

## Lightweight Short-Term Current Prediction

To break the negative noise mitigation loop challenge discussed in Section 1.2.1, this chapter proposes a near-future current prediction method, which can accurately and quickly predict the near-future averaged current of chip load. The proposed method satisfies the prediction length requirement for the proposed proactive noise mitigation system.

### 2.1 Introduction

With the scaling down of the technology node, both power consumption and supply noise are continuously increasing, which causes timing degradation or even malfunctions in modern VLSI chips. Traditional reactive noise mitigation often fails to compensate for emergent supply noise due to the long latency of voltage boosting through the power delivery network (PDN). To conceal such latency, power/current prediction is studied toward proactive noise mitigation [127–129]. Proactive noise mitigation relies on accurate predictions with low hardware and computational cost. However, existing long-term prediction requires high computation cost and consequently longer computation latency, which makes further longer-term prediction requirements. This negative loop makes proactive noise mitigation less effective. To address this negative loop challenge, this chapter proposes a lightweight short-term average current predictor that achieves 50-cycle prediction length and over 0.99 correlation with a six-layer decision tree (DT) regressor.

The rest of this chapter is organized as follows. Section 2.2 explains the overall structure of the proposed proactive noise mitigation. Section 2.3 discusses the construction of lightweight near-future current predictor. Section 2.4 presents the controller and sensor structure of proactive mitigation system. Section 2.5 shows experimental results using RISC-V design. The conclusion of this chapter is given in Section 2.7.

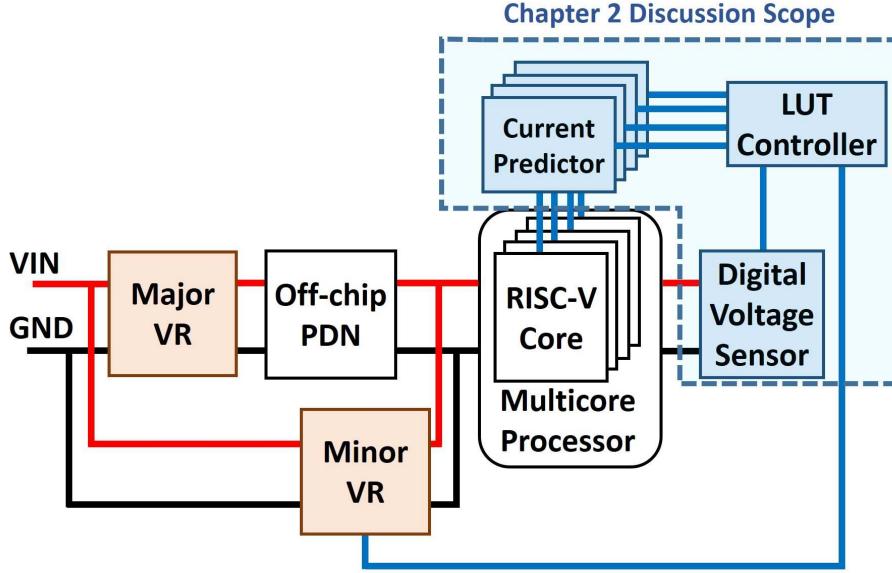


Figure 2.1: Proposed structure for proactive supply noise mitigation. Red lines are power wires, black lines are ground wires, and blue lines are control signal wires.

## 2.2 Overall Structure of Proposed Proactive Noise Mitigation

This section describes the overall structure of the proposed proactive noise mitigation. Fig. 2.1 shows the overall PDN structure with the proactive supply noise mitigation, where off-chip PDN and multicore processor are included in the original design. In this chapter, RISC-V Rocket core [161], which is an in-order single-issue core, is used in the processor module as an example. Here, it is noteworthy that the proposed method is basically independent of the processor core while minor ISA-dependent adaptation is necessary.

The first key component is the major-minor voltage regulator (MMVR), which is shown as orange boxes in Fig. 2.1. The major VR is placed outside the chip and serves as the main power supplier. The minor VR is placed close to the cores, possibly on the chip, and serves as a voltage regulator to mitigate noise. The second key component is the prediction and control units, which are shown as blue boxes in Fig. 2.1. For each RISC-V core, the dedicated current predictor obtains instruction information from I/O ports and then predicts future average current. The controller sums up the prediction results and decides noise mitigation action using a lookup table (LUT). A digital voltage sensor is equipped to override the mitigation action if the voltage is too high or too low for fail-safe purpose. Finally, the action signal is sent to minor VR for noise mitigation.

The remaining of this chapter details the current predictor and controller compo-

nents, which are surrounded by dotted blue lines in Fig. 2.1. Chapter 3 will present the details of MMVR.

## 2.3 Lightweight Short-Term Current Predictor

This section details the short-term current predictor. Fig. 2.2 shows the training and prediction flows of the predictor, where the left side illustrates the off-line training stage and the right side shows the on-line current prediction flow. The key training and prediction procedures are represented in blue blocks.

In the off-line training stage, firstly the training data is prepared from benchmark programs. Simulation is performed to generate current profiles and obtain the instruction at I/O ports for every cycle with logic/circuit simulator or power estimation tools. Then, a set of training label and features are constructed from the instructions and raw current profiles. After that, a decision tree-based predictor is trained. The predictor hardware is implemented accordingly using the training result. In the on-line current prediction stage, firstly the instructions are obtained from I/O ports. Next, the features are constructed and given to the predictor. The prediction results are collected to the controller for MMVR. The label and feature construction, and hardware implementation are discussed in the following.

### 2.3.1 Prediction Label Construction

This work uses a load current value averaged over a certain duration as the training label because of two reasons. Firstly, the load current is independent of PDN, and therefore designers can decouple the on-chip current prediction from the design of the noise controller and voltage regulator. Secondly, the averaged current value can be used as the load current at the PDN port since high-frequency cell switching current is naturally smoothed out by the parasitic impedance, especially by on-chip capacitance.

To generate the training label, the simple moving average (SMA) algorithm is used as a low pass filter to generate the average current value. The averaged current at  $k$ -th clock cycle is defined by:

$$I_{SMA}(k) = \frac{\sum_{j=(k-P+1)}^k I(j)}{P}, \quad (2.1)$$

where  $I(j)$  is the average current within  $j$ -th clock cycle, and  $P$  is the average period represented by clock cycle count. Here,  $P$  is determined by maximizing the summation of the correlation coefficients between voltage droop profile  $V^i$  and averaged current profile  $I_{SMA}^i$  multiplied by  $-1$  across  $N$  voltage drop events:

$$\underset{P}{\text{maximize}} \quad \sum_{i=1}^N \text{correlation}(V^i, -1 \cdot I_{SMA}^i). \quad (2.2)$$

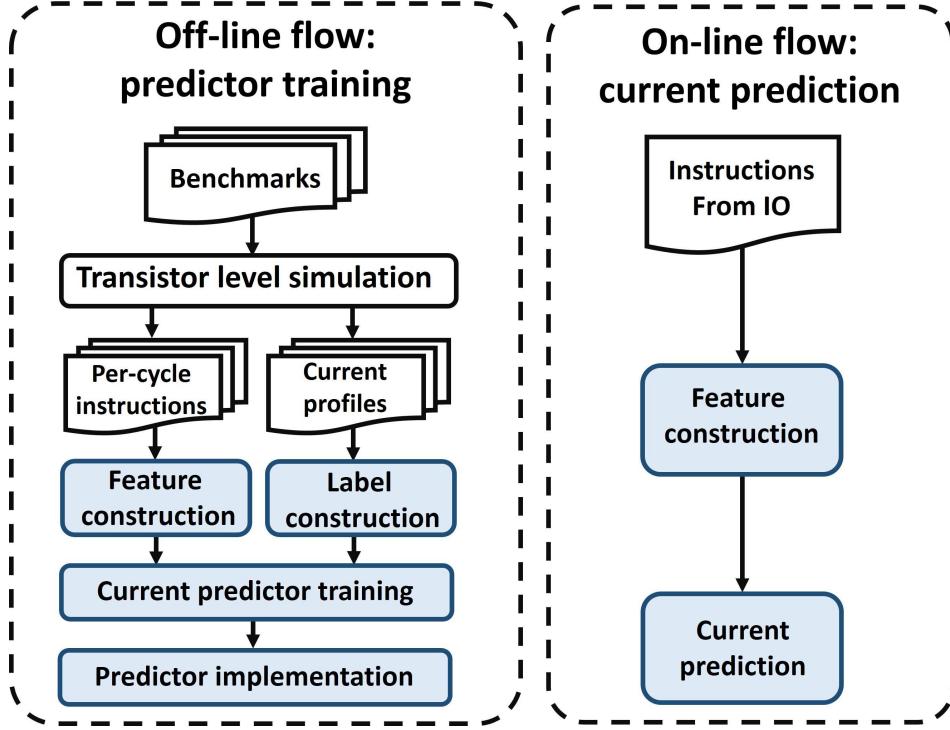


Figure 2.2: Training and prediction flows with current predictor.

Fig. 2.3 exemplifies the  $P$  selection process. First, designers need to run a transient simulation and get the voltage profile at the PDN load port with an actual current profile and PDN model. Next, designers need to collect the profiles of voltage droop events like Fig. 2.3(a) using, for example, a voltage drop threshold.

For those events, average current  $I_{SMA}(k)$  is derived by varying  $P$ , and calculate the correlation with Eq. (2.2). Then,  $P$  is selected by maximizing the average of the correlations. In the RISC-V design that will be explained in Section 2.5, the correlation reaches the maximum of 0.924 with  $P=90$ . In this case, the correspondence between the voltage in Fig. 2.3(a) and the current in blue in Fig. 2.3(b) is well preserved while the high-frequency components are eliminated. If  $P$  is not appropriately selected, for example 500 cycles, the correlation drops to 0.644, and the current pulse becomes much wider than the voltage droop as shown by the red line in Fig. 2.3(b). Such label misleads the noise mitigation action.

Next,  $I_{SMA}(k)$  is shifted by  $L(> 0)$  clock cycles, where  $L$  corresponds to future prediction length. Then, the training label, i.e. the future averaged current at  $k$ -th clock cycle is:

$$I'_{SMA}(k) = I_{SMA}(k + L). \quad (2.3)$$

Longer prediction length  $L$  is expected by proactive mitigation system. However, long-term prediction causes low accuracy and high implementation cost. Besides, processor

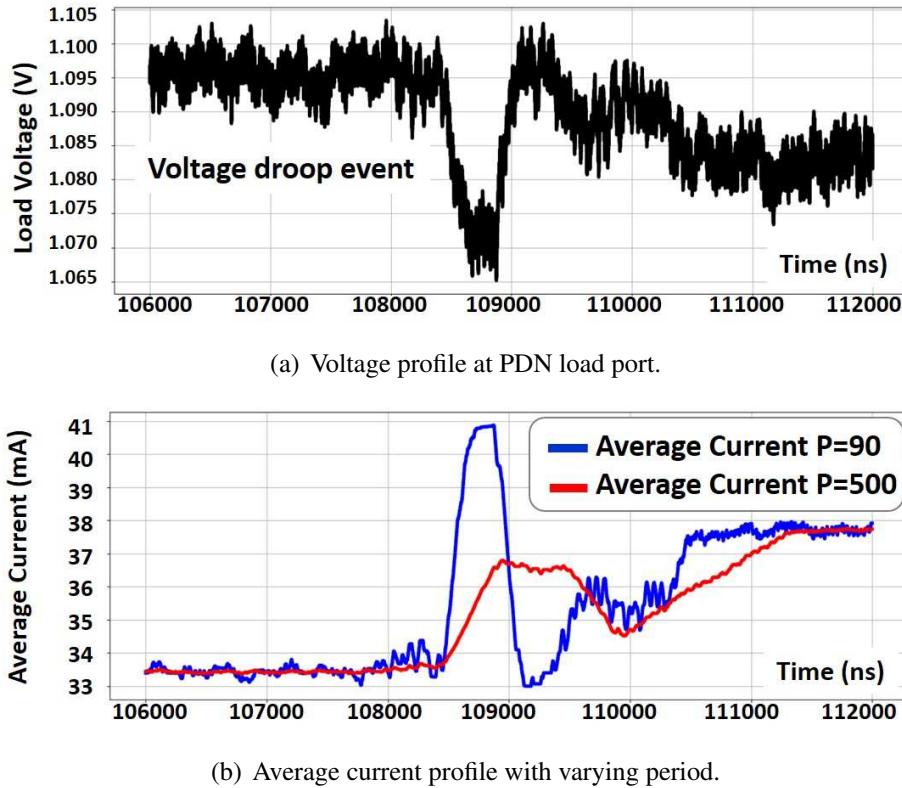


Figure 2.3: Determination of averaging period  $P$  using voltage-current correlation.

structure and configuration also affect  $L$  selection. In this dissertation, the prediction length  $L$  will be determined according to the prediction accuracy, correlation, and implementation cost with experimental evaluations in Section 2.5.

### 2.3.2 Prediction Feature Construction

This subsection describes the features suitable for future prediction supposing RISC-V instruction set as a representative one. A fundamental idea for future prediction is to exploit the temporal locality of processor operation and then suppose the average current in the near future has a strong correlation with the present and previous instructions. For example, when the recently fetched instructions include a lot of floating-point calculation, floating-point unit (FPU) is more likely to dominate the power consumption in several cycles. Furthermore, the instructions which will be fetched immediately after now tend to include floating-point instructions. Compared with conventional approaches that use only the internal hardware signals in the pipelines, longer-term prediction is expected to be feasible since these instructions have not been put into the pipelines yet. On the other hand, the number of available instructions is huge, and then,

Table 2.1: Instruction categorization for RISC-V.

Type No.	Categorization	Example instruction
1	Memory load instructions	lw, ld, lh, lb
2	Memory write instructions	sw, sd, sh, sb
3	Branch instructions	bne, blt, bge, blt
4	ALU instructions	add, sub, or, and
5	Integer multiply division	mul, div, rem
6	CSR access instructions	csrrw, csrrc, csrrwi
7	PC jump instructions	j, auipc, c.j
8	Floating point instructions	fsub, fadd, fmul
9	Routine switch instructions	ret, addi sp a0 1

for facilitating the feature construction, instructions are categorized into a small number of groups each of which has the similar hardware usage, such as FPU, cache, register files, etc., resulting in the similar power dissipation.

To put the above idea into use, firstly the instructions from the RISC-V I/O port are decoded and then categorized into nine types according to Table 2.1.

Let us define the instruction type  $T_i(k)$  of  $k$ -th clock cycle as:

$$T_i(k) = \begin{cases} 1 & \text{if } k\text{-th instruction belongs to type } i, \\ 0 & \text{otherwise.} \end{cases} \quad (2.4)$$

To eliminate the on-chip memory for saving the history of instruction type, the exponential moving average (EMA) algorithm is used to derive features  $F_i(k)$  in  $k$ -th cycle that represents how frequently  $i$ -th instruction type is fetched recently.

$$F_i(k) = \alpha T_i(k) + (1 - \alpha) F_i(k-1), \quad (0 < \alpha < 1). \quad (2.5)$$

When  $F_i(k)$  is close to 1, most of recently fetched instructions belong to  $i$ -th instruction type.  $\alpha$  is a coefficient that adjusts the weight on the current and historical instruction type. When  $\alpha$  is close to 1,  $F_i(k)$  is more sensitive to current instruction type. Conversely, when  $\alpha$  is close to 0, longer instruction type history is included.  $\alpha$  is determined by maximizing the summation of correlation between feature  $F_i$  and averaged

current profile  $I_{SMA}$ :

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^M |\text{correlation}(F_i, I_{SMA})|, \quad (2.6)$$

where  $M$  is the feature dimension. The result  $(1/\alpha)$  can be round to the nearest power-of-two integer to further reduce hardware implementation cost.

### 2.3.3 Predictor Engine and Implementation Cost

This work uses DT as the prediction engine because firstly, the algorithmic complexity and memory requirements for DT are much lower compared with SVM. This advantage is critical for quick prediction with low hardware cost. Secondly, DT has non-linear regression capability even with simple computation. On the other hand, SVM regressor, which is used in conventional works [127, 128], uses linear kernel which has the limited capability to regress training data. If SVM uses non-linear function as the kernel function, the regression to the non-linear functions becomes possible. However, the computational cost for such kernel functions is usually very high. Therefore, non-linear kernel SVM is not considered in this work. With the DT prediction engine, the training label and feature are the future averaged current in Eq. (2.3) and EMA of instruction type value in Eq. (2.5).

The hardware cost of DT predictor, denoted by  $H$ , consists of two factors:

$$H = H_{\text{feature}}(M) + H_{\text{node}}(2^D - 1), \quad (2.7)$$

where  $H_{\text{feature}}$  is the hardware cost for instruction decoding, categorizing, and feature construction. This cost is roughly proportional to the feature dimension  $M$ , which is nine in this work.  $H_{\text{node}}$  is the cost for decision nodes, and it increases exponentially with decision tree depth  $D$ . Therefore, small tree depth is highly desirable. The advantage and the necessary depth of DT will be experimentally discussed in Section 2.5.

## 2.4 Noise Mitigation Controller

The noise mitigation controller sums up the predicted values from the predictors and then uses lookup table (LUT) to decide noise mitigation action, that is, to set the conversion ratio and the switching frequency of the minor VR. As an example, if an average current jump is predicted, the controller will set the minor VR to voltage scaling mode and increase the switching frequency according to LUT. The LUT entry is experimentally determined.

To prevent wrong mitigation action at very high or very low voltage level, an on-chip digital voltage sensor is introduced to override the wrong LUT based prediction action. A simple digital voltage sensor structure is exemplified in Fig. 2.4, which is

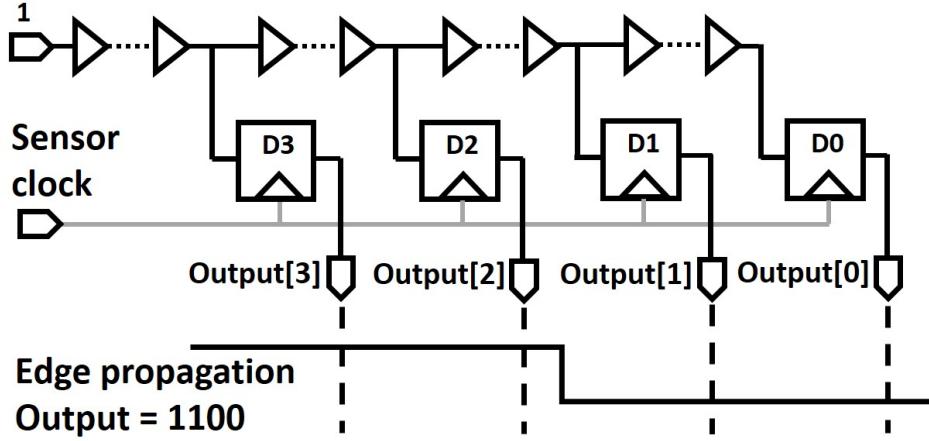


Figure 2.4: Digital voltage sensor.

Table 2.2: Overriding rule table with sensor output.

Output	Voltage range	Overriding rule
0000	Ultra low voltage	Perform voltage scaling up
1000	Low voltage range	Voltage scaling down prohibited
1100	Normal voltage range	Accept all LUT based action
1110	High voltage range	Voltage scaling up prohibited
1111	Ultra high voltage	Perform voltage scaling down

found in [162]. Here, the four-bit output varies from 0000 to 1111, depending on the supply voltage level. For example, if the voltage is too low, the sensor output is 1000, voltage scaling down action is prohibited, and only scaling up action is allowed. The entire overriding rule is shown in Table 2.2. It should be noted that, the voltage sensor is introduced for the fail-safe purpose. If the predictor can accurately work in all scenarios, the voltage sensor and override part can be removed from the noise mitigation system.

## 2.5 Experimental Results

This work uses 64-bit RISC-V Rocket core [161] as chip load. The core is synthesized with NanGate 45 nm Open Cell Library [163]. Nominal voltage is 1.1 V. Test benches are C programs including integer and floating-point calculation, matrix calculation, logic calculation, recursive functions, multi-threading, branch control flow, and

the combination of them. These benchmarks are derived from RISC-V regression test cases to cover most of the available functionality and representative instruction operations. Then, the I/O values regarding instructions and current profiles are generated via transistor-level simulation in this work. Next, feature and label construction are performed, and the total number of data samples is 2.58 million, where 50% of the data for training and the rest for testing.  $P$  in Eq. (2.1) is set to 90 according to Eq. (2.2), and  $\alpha$  in Eq. (2.5) is set to 1/32 according to Eq. (2.6). DT predictor is trained off-line with Sklearn package [164].

The performance of short-term current predictor is evaluated using root-mean-square-error (RMSE) and correlation coefficient. RMSE measures the overall prediction error, and correlation coefficient measures the prediction quality on rare events, for example, whether the predictor can track the emergent average current jump. The RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (I'_{SMA}(j) - \hat{I}'_{SMA}(j))^2}{N}}, \quad (2.8)$$

where  $N$  is the data set size,  $I'_{SMA}(j)$  is the training label, which is future averaged current at  $j$ -th clock cycle, and  $\hat{I}'_{SMA}(j)$  is the prediction result. The correlation coefficient is measured between  $I'_{SMA}(j)$  and  $\hat{I}'_{SMA}(j)$ . SVM prediction engine is chosen as a comparison, where the tolerance margin  $\epsilon$  is selected as 1 mA and 0.5 mA.

The RSME and correlation coefficient are evaluated by varying the prediction length. Figs. 2.5 and 2.6 show their results, where the blue and red lines are the results of the six-layer and ten-layer DTs, green and purple lines are the results of SVM predictions, respectively.

The deeper DT provides longer prediction length with the same accuracy, but the correlation still drops below 0.99 beyond 100 clock cycles. On the other hand, the SVM predictor shows worse accuracy and correlation at every prediction length. 50 clock cycles is selected as the prediction length because both the DTs achieve the correlation higher than 0.99 and the RMSE is almost constant. Besides, it can be observed in Fig. 2.5 that the RMSE slope of DT is sharper than that of SVM after 50 cycle prediction length. Because for the SVM regressor, the accuracy loss can be retrieved by introducing additional support vectors. In the experiment, the support vector count increased from 708 to 939 with the increase in prediction length. On the other hand, the decision tree regressor has a fixed number of decision nodes, and hence the decision tree regressor cannot add additional decision nodes to recover the overall accuracy. Therefore, RMSE of DT has a sharper slope than that of the SVM.

Next, hardware cost and prediction quality are compared between DT and SVM predictors, where SVM uses a linear kernel. Both the predictors are designed with 8-bit floating-point representation having four fraction bits to save hardware cost and improve the prediction robustness. To minimize the prediction latency, the predictors are designed for one-cycle completion. The hardware overhead is defined as the predictor area over RISC-V core area. The comparison results in Table 2.3 show that the deep DT

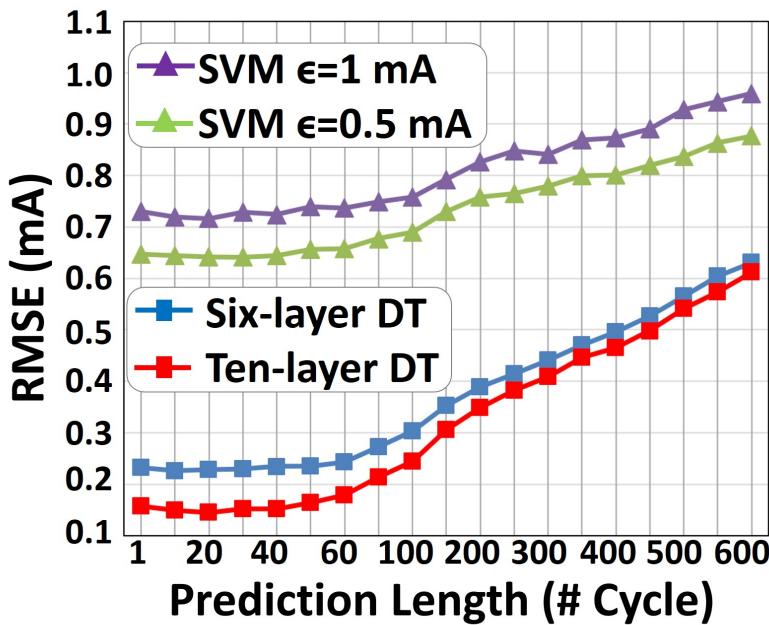


Figure 2.5: RMSE versus prediction length.

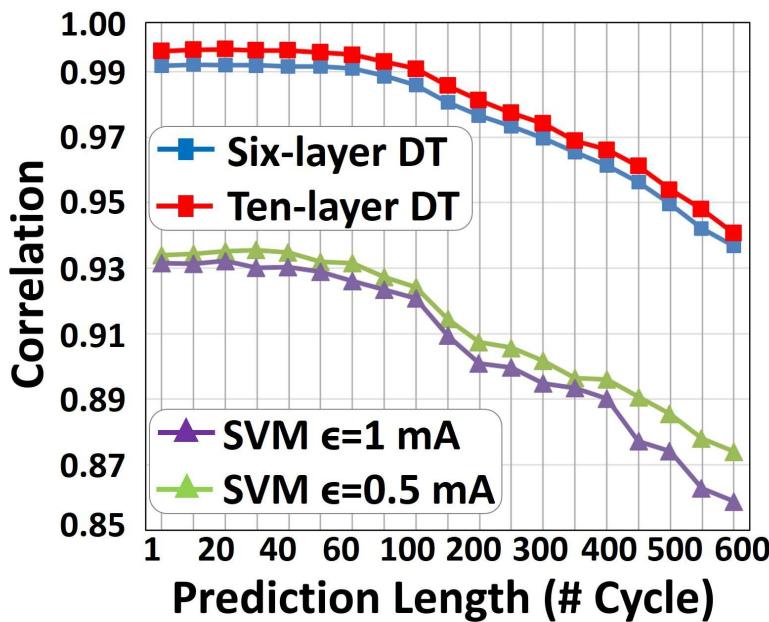


Figure 2.6: Correlation versus prediction length.

Table 2.3: Prediction performance and hardware cost.

DT depth	#Nodes	Overhead (%)	RMSE (mA)	Correlation
5	31	1.48	0.275	0.988
6	63	2.51	0.236	0.992
7	127	4.57	0.209	0.993
8	255	8.68	0.189	0.994
9	511	16.91	0.172	0.995
10	1023	33.37	0.157	0.996

SVM $\epsilon$ (mA)	#SVs	Overhead (%)	RMSE (mA)	Correlation
1	701	115.41	0.739	0.929
0.5	1442	236.88	0.656	0.932

predictor can achieve 0.996 correlation at the cost of 1023 decision nodes and 33.37% hardware overhead. When pursuing a practical lightweight predictor, the six-layer DT with 63 decision nodes is sufficient to achieve over 0.99 correlation with 2.51% overhead. Even though the SVM predictor provides worse accuracy, the number of support vectors (SVs) reaches 701, which requires 115.41% overhead for one-cycle computation. Note that even when a pipeline structure is adopted, the hardware cost does not decrease. If the prediction throughput is reduced, the hardware cost decreases. However, the substantial prediction length for voltage boosting becomes shorter due to both the latency increase and throughput decrease. Consequently, SVM prediction engine cannot be adopted.

Fig. 2.7 demonstrates the prediction accuracy of the DT and SVM predictors in time domain. Here, a recursive floating-point calculation benchmark is used as an example. The result shows that the six-layer DT has a better correlation with the average current profile, and the emergent current jump and drop are closely tracked. However, the SVM prediction induces a large variation from the actual future average current.

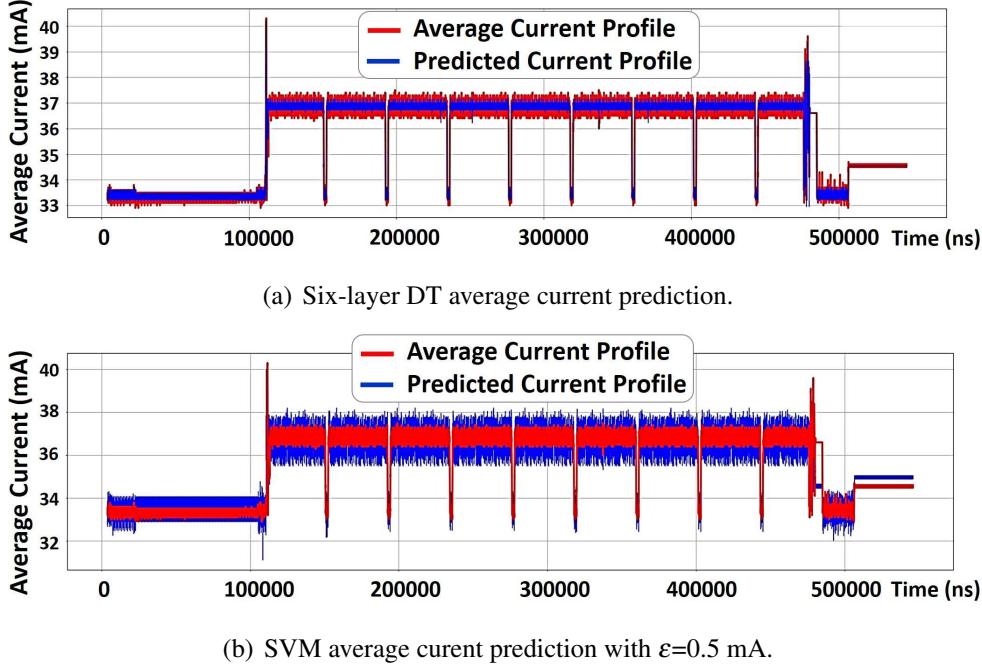


Figure 2.7: Current prediction results with DT and SVM.

## 2.6 Extended Discussion on Out-of-Order Processor

The current prediction for the out-of-order (OoO) processor is included as an extended discussion. Though the instruction execution order of OoO processor is dynamically scheduled depending on the resource and inter-instruction dependency, such OoO execution impact is supposed to be limited within the benchmarks used in the experiment. Meanwhile, the proposed method can be applied to OoO processor though the inter-instruction dependency might be included in the training feature to cover certain programs that have a high potential of instruction reordering.

Firstly, the prediction accuracy impact from out-of-order execution is limited within the used benchmarks. OoO processor usually has the instruction window which can hold around 100 instructions for issuing execution. However, not all the instructions can be reordered because the memory/register committing, branch, fence, return, and status register update instructions have a strong dependency on the previous execution result, and therefore they are strictly controlled from re-ordering and executed in sequence. In the common program, sequential instructions can occupy over 20% of total instructions [165]. In the benchmarks used in this chapter, the ratio of sequential instructions over total instructions ranges from 15.3% to 31.1%. That means, suppose an OoO processor can hold 100 instructions for OoO execution, then averagely three to six instructions can be re-ordered between sequential instructions. Considering a typi-

cal OoO operation, instructions per cycle (IPC) varies from one to two, and such local reordering may impact current consumption in merely several clock cycle level. Therefore, the reordering impact is limited compared with prediction length which is 50 clock cycle in this work.

Secondly, the prediction method may need to consider instruction dependency. For example, a derivative program may contain highly inter-dependent floating-point instructions [166]. In such a case, fewer instructions are reordered. On the other hand, a large-size array access program can sustain a longer period of out-of-order execution. In such a case, very few dependencies exist among local instructions. Therefore, instruction dependency may need to be included in the input feature. On the other hand, in this case, similar instructions are repeatedly executed, and hence the prediction might be easy.

Validating the above qualitative discussion with experiments is important. Performing an experimental validation is a future work.

## 2.7 Conclusion

This chapter has proposed a proactive noise mitigation structure using lightweight near-future current predictor, which is implemented with a simple six-layer decision tree, and achieves over 0.99 correlation for 50-cycle prediction length with the hardware overhead of 2.51%. The prediction length can be exploited for fast dynamic voltage scaling to achieve practical proactive noise mitigation. The dynamic voltage scaling part will be discussed in the next chapter.



# Chapter 3

## Low-Latency Voltage Scaling Using Major-Minor Voltage Regulator

This chapter presents the major-minor voltage regulator, which can achieve fast and continuous supply voltage scaling. The proposed solution, together with the noise mitigation controller, and current predictor proposed in Chapter 2, achieves the closed-loop solution of the proactive noise mitigation system.

### 3.1 Introduction

Proactive noise mitigation requires quick and continuous voltage scaling with a wide scaling range and small voltage ripple. Switched capacitor voltage regulator (SCVR) is a popular off-chip power supply solution for modern VLSI designs. However, off-chip SCVR has limited voltage scaling flexibility and long response time. Various methods are proposed to improve the voltage scaling capability, such as using low-dropout (LDO) voltage regulator as a secondary linear regulator [56, 57, 79–81, 137, 138], multi-VDD SCVR [116, 139], multiple SCVRs with different conversion ratio [50, 51, 78], multiple-conversion-ratio SCVRs or re-configurable SCVR [51, 140], and recursive SCVR structure [47, 141–144]. However, these solutions either suffer from discrete voltage level, large voltage ripple, long response time, or heavy design cost.

To address this challenge, this chapter proposes a major-minor voltage regulator (MMVR) structure, which consists of two SCVRs whose flying capacitances are much different. The proposed MMVR can provide continuous wide-range voltage scaling capability with simple switching frequency modulation.

Fig. 3.1 shows the overall PDN structure and the proactive supply noise mitigation, where the major-minor voltage regulator (MMVR) is shown as orange boxes. The major VR is placed outside the chip and serves as the main power supplier. The minor VR is placed close to the cores, possibly on the chip, and serves as a voltage regulator to mitigate noise. MMVR is modulated by the prediction and control units, which have

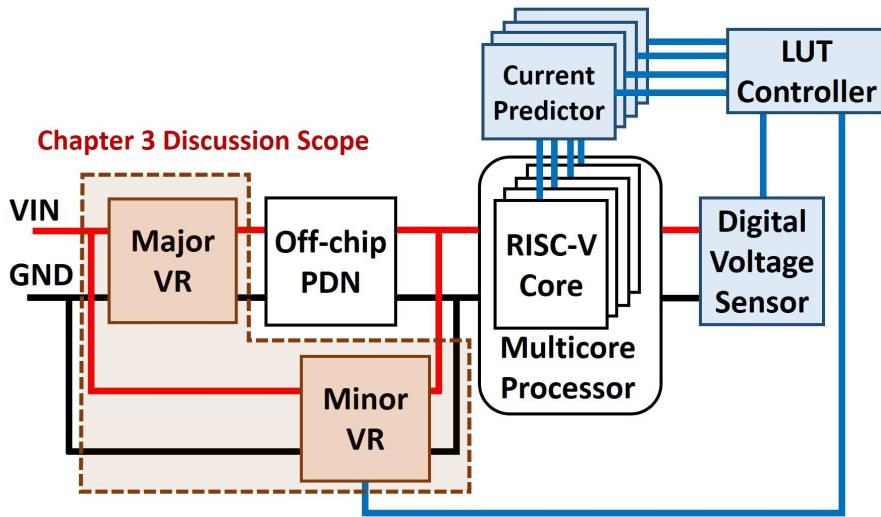


Figure 3.1: Proposed voltage regulator in proactive supply noise mitigation system. Red lines are power wires, black lines are ground wires, and blue lines are control signal wires.

been described in Chapter 2.

In the rest of this chapter, Section 3.2 explains the details of MMVR structure. Section 3.3 evaluates the performance of MMVR and the performance of entire proactive noise mitigation system.

## 3.2 Scalable Major-Minor Voltage Regulator

This section proposes a scalable switched capacitor voltage regulator called major-minor voltage regulator (MMVR). MMVR consists of major VR and minor VR, and its simplified connection is depicted in Fig. 3.2. Major VR serves as a major power supplier with a fixed conversion ratio and large flying capacitance. A typical 2:1 major VR structure is shown in Fig. 3.3, where the switches toggle with two-phase pulses  $\phi_1$  and  $\phi_2$ .  $C_{major}$  denotes the flying capacitance of major VR.

The minor VR with smaller flying capacitance is designed for voltage scaling, and it has conversion-ratio reconfigurability. By changing the switches status, the minor VR can operate in 2:1 normal mode (Fig. 3.4), and 3:2 scaling mode (Fig. 3.5). When an emergent power requirement arises, the minor VR is switched to the scaling mode. Also, the output voltage is scaled by modulating the switching frequency of minor VR. In this way, the output voltage of MMVR,  $V_{out}$ , can be scaled between 1/2 and 2/3 of the input voltage  $V_{in}$ .

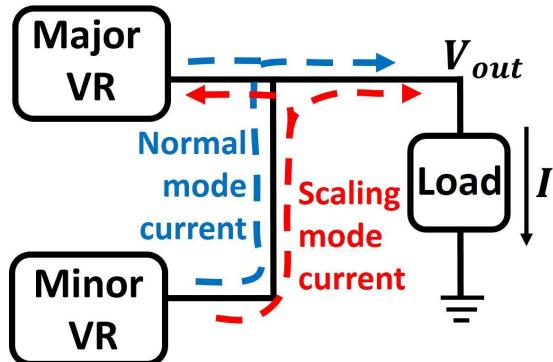


Figure 3.2: MMVR connection diagram.

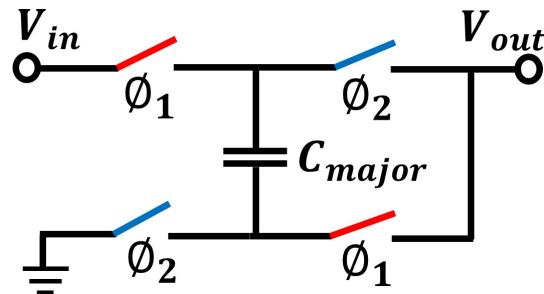


Figure 3.3: Major VR with 2:1 conversion ratio.

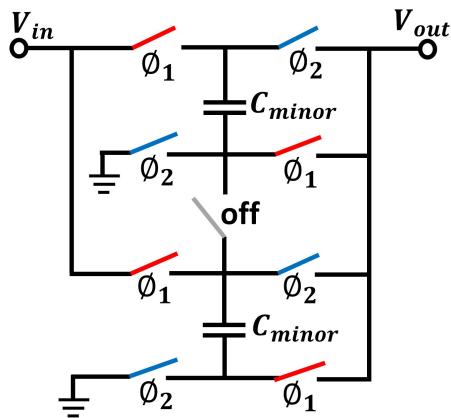


Figure 3.4: Minor VR in normal mode with 2:1 conversion ratio.

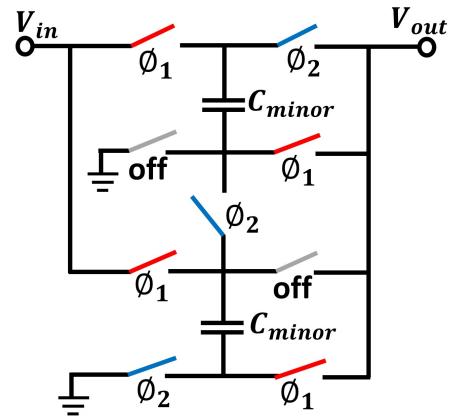


Figure 3.5: Minor VR in scaling mode with 3:2 conversion ratio.

SCVR causes voltage ripple every time the switches are turned on and off due to its operation principle. In MMVR, the ripple depends on the operation mode. When both the major and minor VRs work in normal mode with the same switching frequency, the

dynamic current flows like the blue dot line in Fig. 3.2, and MMVR is equivalent to a traditional SCVR. As is well studied in [100], the output ripple in normal mode can be approximated as:

$$V_{r\_norm} = \frac{\alpha I}{f_{sw}(C_{major} + C_{minor})}, \quad (3.1)$$

where  $\alpha$  is a structural coefficient for both major VR and minor VR,  $I$  is dynamic load current,  $f_{sw}$  is MMVR switching frequency, and  $C_{major}$  and  $C_{minor}$  are VR flying capacitance of major VR and minor VR, respectively.

On the other hand, when the minor VR works in voltage scaling mode with 3:2 conversion ratio, the dynamic current goes from the minor VR to the major VR in addition to the load since the output voltage of the minor VR is higher than that of the major VR, which is illustrated as the red dot line in Fig. 3.2. Considering that the minor VR has a different conversion ratio, structural coefficient, and switching frequency, the ripple discussion in [100] is extended accordingly. The dynamic load current  $I$  of MMVR can be approximated as:

$$I = I_{minor} - I_{major} = \frac{V_{r\_scale} f_{minor} C_{minor}}{\alpha_{minor}} - \frac{V_{r\_scale} f_{major} C_{major}}{\alpha_{major}}, \quad (3.2)$$

where  $I_{major}$  and  $I_{minor}$  are the dynamic current that go through major VR and minor VR, and  $V_{r\_scale}$  is the dynamic load voltage, which is the output voltage ripple. Then, the MMVR output ripple can be derived as:

$$V_{r\_scale} = \frac{I}{\frac{f_{major} C_{major}}{\alpha_{major}} - \frac{f_{minor} C_{minor}}{\alpha_{minor}}}, \quad (3.3)$$

where  $f_{major}$  and  $f_{minor}$  are switching frequencies of major VR and minor VR.  $C_{major}/C_{minor}$  and  $\alpha_{major}/\alpha_{minor}$  are VR flying capacitances and VR structural coefficients of major VR and minor VR, respectively. Eq. (3.3) suggests increasing the capacitance difference between major VR and minor VR to reduce the ripple. Therefore, this work intentionally uses a small flying capacitance for minor VR. In the experiment in Section 3.3, the capacitance ratio reaches ten. Such a small capacitor can be integrated into the chip package or even on the chip, and hence minor VR can be placed close to cores, and fast voltage response becomes feasible.

### 3.3 Experimental Results

The experimental condition is identical with that in Chapter 2. In addition, the predictor and controller of noise mitigation system are identical with those in Chapter 2. the total flying capacitance of major VR is 50 nF, and that of minor VR is 5 nF. Both major VR and minor VR are implemented using NanGate 45-nm CMOS model and capacitor components.

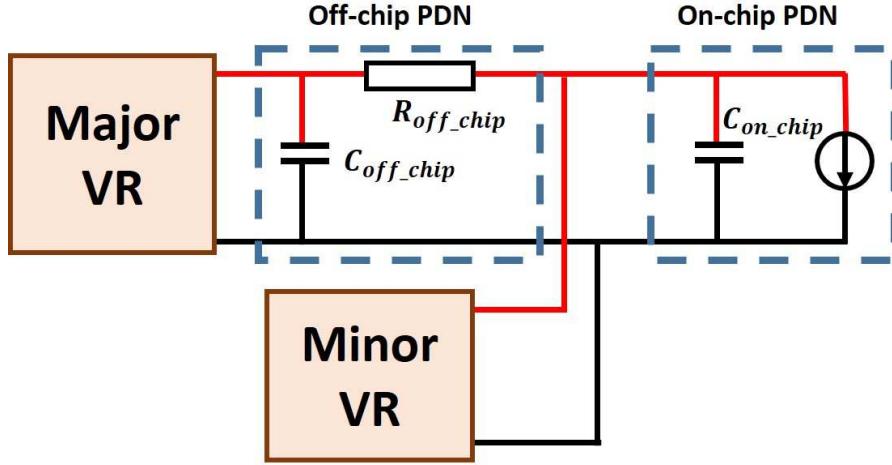


Figure 3.6: MMVR performance test circuit.

### 3.3.1 MMVR Performance Experiment

The first experiment compares the performance between the proposed MMVR and traditional SCVR in terms of the voltage scaling range. The test circuit is shown as Fig. 3.6, where the off-chip PDN and the on-chip PDN are intentionally simplified for demonstration purposes. In the test circuit,  $C_{off\_chip}$  is  $0.4 \mu\text{F}$ ,  $R_{off\_chip}$  is  $100 \text{ m}\Omega$ , and  $C_{on\_chip}$  is  $10 \text{ nF}$ . The nominal load voltage is  $1100 \text{ mV}$ . An  $800 \text{ mA}$  current source is attached as a load to mimic the power-hungry processor operations.

Fig. 3.7 shows the output voltage when the VR switching frequency is swept. The traditional SCVR output voltage is bounded at near  $970 \text{ mV}$  even with a high switching frequency, and the voltage scaling range is limited within  $40 \text{ mV}$ . On the other hand, MMVR can boost the output voltage to  $1048 \text{ mV}$ . The scaling range is  $3\times$  larger compared with the traditional SCVR.

Fig. 3.8 shows the output voltage ripple at different output voltage levels. The maximum ripple of MMVR is  $15.9 \text{ mV}$ , which means the MMVR and SCVR have a comparable ripple magnitude even while the major and minor voltage regulators are operating with different voltage conversion ratios and different switching frequencies.

Fig. 3.9 shows the MMVR conversion efficiency versus load voltage scaling range. When MMVR works in normal mode, the efficiency is identical to that of the traditional SCVR. When MMVR works in voltage scaling mode, the efficiency slightly drops, yet it is still above  $63.5\%$ . Note the scaling mode is only triggered in a short emergent period, and hence this small efficiency drop has the least impact on the overall efficiency.

Finally, the voltage scaling response time is compared. Conventional SCVR takes  $226.9 \text{ ns}$  to boost  $10 \text{ mV}$  load voltage, while MMVR takes  $15.6 \text{ ns}$ . Such a short response time relieves the prediction length requirements and makes the proactive noise mitigation possible with 50-cycle current prediction, which is achieved in Chapter 2.

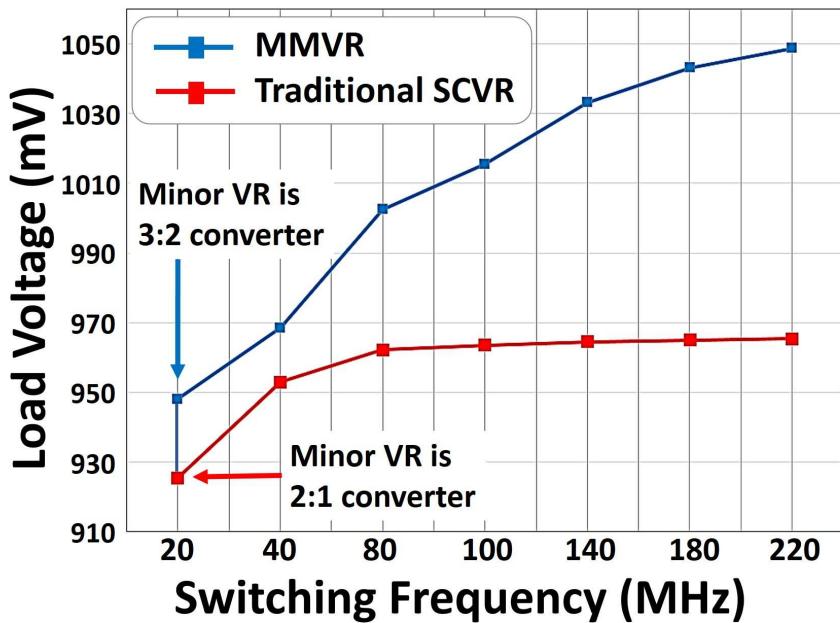


Figure 3.7: Comparison in voltage scaling range.

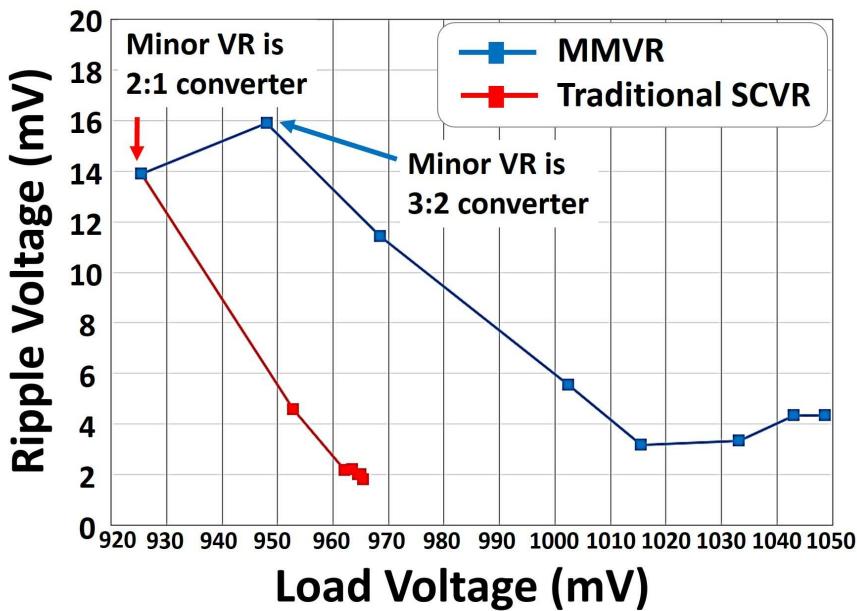


Figure 3.8: Comparison in ripple voltage.

### 3.3.2 Proactive Versus Reactive Noise Mitigation

This experiment demonstrates the effectiveness of proactive noise mitigation using the proposed MMVR solution and proactive noise mitigation system proposed in Chapter 2.

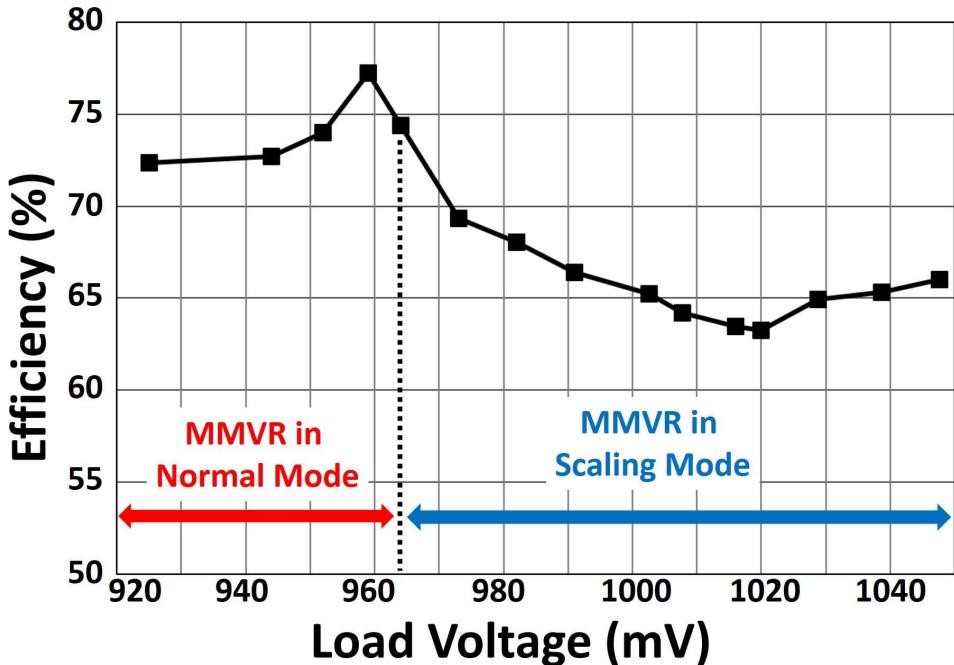


Figure 3.9: MMVR efficiency versus load voltage.

For this experiment, the system diagram with proactive mitigation is shown in Fig. 3.1. The nominal load voltage is set as 1100 mV. A four-core RISC-V chip load model is prepared with the proposed proactive noise mitigation method. As a comparison, this experiment includes a reactive noise mitigation method that modulates minor VR to boost the voltage if the load voltage drops below the low bound. For both the mitigation methods, the low voltage bound is set as 1010 mV. To compare the worst-case voltage drop, both setups run the same benchmark in each core, resulting in a large voltage drop during the initialization stage.

The voltage waveforms at the load are shown in Fig. 3.10, where the blue waveform corresponds to the proactive noise mitigation method and the red waveform is the reactive mitigation method. In the worst voltage drop case of proactive noise mitigation, the voltage is above 1040 mV, and the voltage recovers in 40 ns. The droop transient process is caused by PDN latency. Furthermore, the proactive noise mitigation method can stabilize the average load voltage around 1060 mV with ripple of less than 30 mV. As for the reactive mitigation, the voltage drop exceeds 70 mV, and the voltage goes below the 1010 mV bound because of the PDN latency. Also, the average voltage drop exceeds 20 mV during full operation period after 115  $\mu$ s.

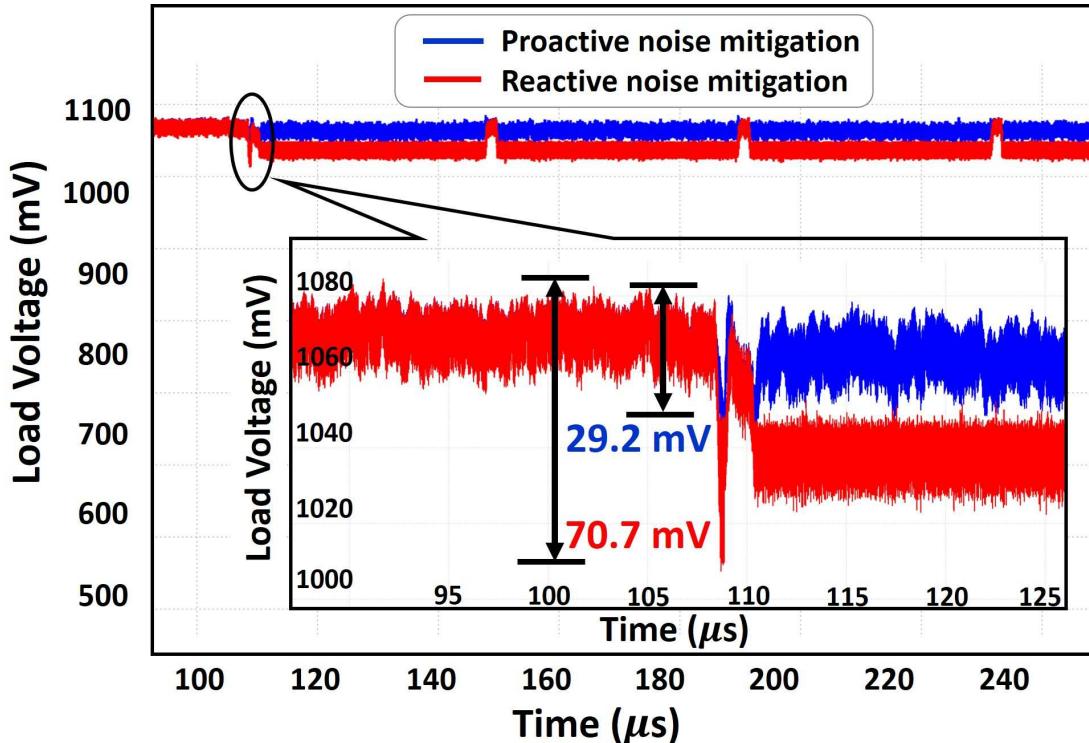


Figure 3.10: Noise mitigation result for multicore RISC-V PDN.

### 3.4 Conclusion

This chapter proposed major-minor voltage regulator, which provides over 3X scaling range compared with traditional SCVR even while the ripple is suppressed within 16 mV. This MMVR structure enables fast and wide-range voltage scaling. The system-level simulation validates the effectiveness of the MMVR, controller, and predictor. The voltage drop is mitigated within 30 mV by the proposed proactive mitigation, while it is 70 mV for traditional reactive mitigation. These results clarify the effectiveness of the proposed proactive noise mitigation system.

# Chapter 4

## Frequency-Dependent Target Impedance Methodology

This chapter proposes a frequency-dependent target impedance methodology, which considers the constraints of both average and dynamic voltage drops. To bridge the design gap between frequency and time domain, a concept of magnitude equivalent frequency (MEF) is proposed to simplify the frequency-dependent target impedance design. The proposed methodology is experimentally validated with various current loads.

### 4.1 Introduction

High-quality low-noise power delivery network (PDN) is demanded by every design to ensure its performance. Target impedance methodology is a common practice to guide PDN design. However, as is introduced in Chapter 1, traditional methodology has a design gap between the time-domain design constraints and the frequency-domain target impedance. Meanwhile, the average voltage drop constraint is not well handled. Ignoring such a design gap can result in under- or over-designed PDN as discussed in Section 1.2.2.

Target impedance should consider the constraints of both average and dynamic voltage drops, which means the target impedance value could vary depending on the frequency. Here, it should be noted that a number of possible frequency-dependent target impedances exist since the degree of freedom is much larger than the number of the given constraints. Among them, it is necessary to provide a simple frequency-dependent target impedance that has fewer parameters yet satisfies the constraints and has compatibility with PDN design.

This chapter proposes a frequency-dependent target impedance with four parameters of  $Z_{ac\_target}$ ,  $Z_{dc\_target}$ ,  $C_{target}$ , and  $L_{target}$ .  $Z_{ac\_target}$  and  $Z_{dc\_target}$  denote the target impedance magnitudes at middle frequency range and DC, respectively. Fig 4.1 shows

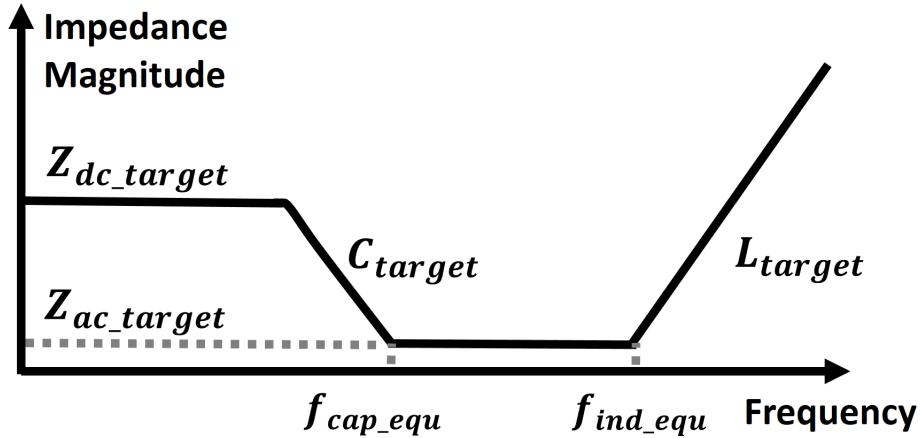


Figure 4.1: RLC target impedance.

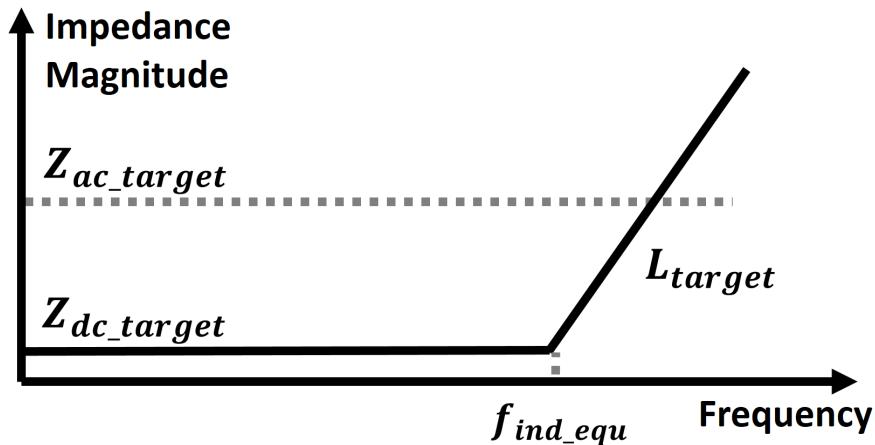


Figure 4.2: RL target impedance.

an example of frequency-dependent target impedance, in which  $Z_{dc\_target} > Z_{ac\_target}$ . To minimize PDN design cost, this work will find the minimum required capacitance, which is specified by target capacitance  $C_{target}$ , and the maximum allowable inductance, which is target inductance  $L_{target}$ .

## 4.2 Overall Flow and Basic Impedance Shapes

As the first step, PDN designers shall determine, or be given, the maximum allowable average and dynamic voltage drops,  $V_{avg\_allow}$  and  $V_{dyn\_allow}$ , as PDN design constraints. These constraints determine the basic target impedance shape in the frequency domain.

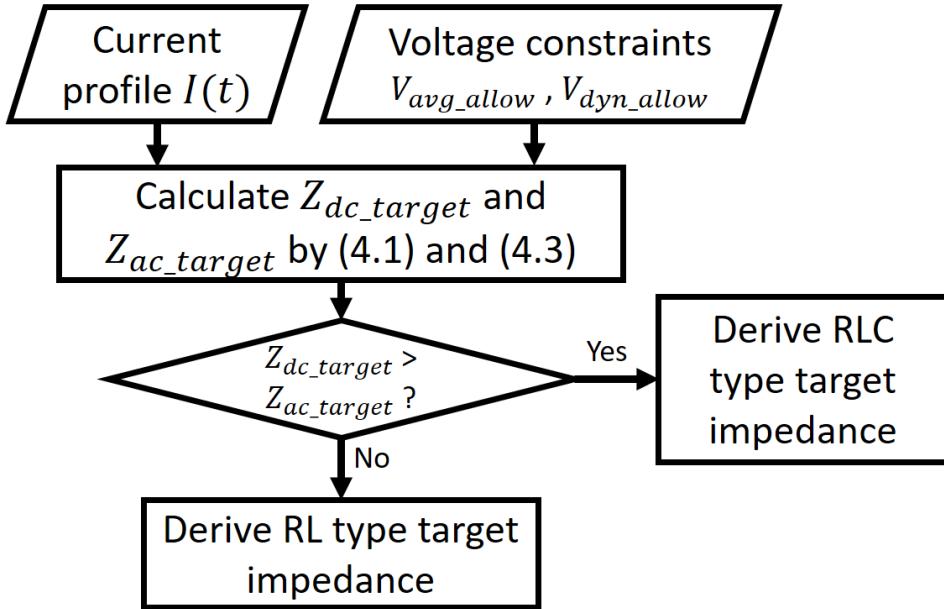


Figure 4.3: Overall flow of frequency-dependent target impedance methodology.

Supposing the average load current is  $I_{avg}$ , the target impedance in low frequency range including DC is:

$$Z_{dc\_target} = \frac{V_{avg\_allow}}{I_{avg}}. \quad (4.1)$$

As for the dynamic voltage drop constraint, let us first define the magnitude of load current  $I(t)$  and voltage  $V(t)$  as:

$$\begin{aligned} \text{Mag}(I(t)) &= I_{max} - I_{avg}, \\ \text{Mag}(V(t)) &= V_{avg} - V_{min}, \end{aligned} \quad (4.2)$$

where  $I_{max}$  is the maximum value of  $I(t)$ ,  $I_{avg}$  is the average value of  $I(t)$ ,  $V_{avg}$  is the average load voltage, and  $V_{min}$  is the minimum load voltage. The target impedance in the middle frequency range is:

$$Z_{ac\_target} = \frac{V_{dyn\_allow}}{\text{Mag}(I(t))}. \quad (4.3)$$

$Z_{ac\_target}$  can be either larger or smaller than  $Z_{dc\_target}$ , and then two types of target impedance shape exist.

Fig. 4.1 shows the target impedance shape in case of  $Z_{dc\_target} > Z_{ac\_target}$ , which is called RLC type. In this case, mitigating dynamic voltage drop is the main PDN design challenge. The PDN design goal is to find the minimum of required target capacitance  $C_{target}$  and the maximum of allowable target inductance  $L_{target}$  so that  $Z_{ac\_target}$  can be met with the minimal design resource.

Fig. 4.2 corresponds to the case of  $Z_{dc\_target} < Z_{ac\_target}$ , where the average voltage drop is the severer constraint than the dynamic voltage drop. This shape is called RL type. The goal is to find  $L_{target}$ , such that  $Z_{ac\_target}$  can be met with the minimal design resource. Another special case of  $Z_{dc\_target} = Z_{ac\_target}$  is treated as a corner case of RL-type target impedance.

Fig. 4.3 shows the overall flow of target impedance derivation, where current profile  $I(t)$  and voltage constraints of  $V_{avg\_allow}$  and  $V_{dyn\_allow}$  are given to the flow. The following explains how to derive  $C_{target}$  and  $L_{target}$  using a concept of MEF.

## 4.3 Magnitude Equivalent Frequency (MEF)

The key idea of MEF is, instead of analyzing the detailed current waveform, a sine waveform current can be used to reproduce the same magnitude of the voltage noise. The frequency of this sine waveform is defined as MEF. Once MEF is obtained for capacitance dominant impedance, such MEF can be regarded as the corner frequency  $f_{cap\_equ}$  to  $C_{target}$  in Fig. 4.1. Similarly, MEF for inductance dominant impedance is denoted as  $f_{ind\_equ}$ , which is used as the corner frequency to derive  $L_{target}$ . The derivation of  $C_{target}$  and  $L_{target}$  will be discussed in the next subsection. The remaining of this subsection proves the existence of such MEFs and discusses the property of MEF.

### 4.3.1 MEF for Capacitance Dominant Impedance

For capacitance dominant impedance, supposing the magnitudes of original load current  $I(t)$  and voltage  $V(t)$  are bounded, which is always hold in actual PDNs, it is necessarily to have a sine waveform current  $I_s(t)$  that has the same magnitude, that is,

$$\text{Mag}(I_s(t)) = \text{Mag}(I(t)). \quad (4.4)$$

Then,  $\text{Mag}(V_s(t))$  becomes a function of frequency for capacitance  $C$  dominant impedance:

$$\text{Mag}(V_s(t)) = \frac{\text{Mag}(I_s(t))}{2\pi C f_{cap\_equ}}. \quad (4.5)$$

Therefore, there exists a frequency of sine waveform  $f_{cap\_equ}$  that achieves

$$\text{Mag}(V_s(t)) = \text{Mag}(V(t)). \quad (4.6)$$

Hereafter,  $f_{cap\_equ}$  is denoted as capacitance MEF of load current. The existence of this capacitance MEF can be summarized by:

**Theorem 1** *Let  $I(t)$  be load current profile and  $V(t)$  be corresponding PDN voltage profile. If  $V(t)$  and  $I(t)$  are bounded,  $\text{Mag}(V(t))$  across capacitance dominant impedance can be reproduced by current  $I_s(t) = \text{Mag}(I(t)) \cdot \sin(2\pi f_{cap\_equ} \cdot t)$ .*

Such  $I_s(t)$  is called magnitude equivalent current (MEC) of capacitance dominant impedance. Furthermore, MEF value is independent of capacitance value. That is:

**Theorem 2** *Let  $V(t)$  be the voltage profile for the original current profile, and  $V_s(t)$  be the voltage profile for the MEC to the original current profile. Then, for all the capacitance dominant impedances,  $\text{Mag}(V_s(t)) = \text{Mag}(V(t))$  hold.*

With the definition of (4.2), the magnitudes of current and voltage satisfy the properties below, where  $N_A$  and  $N_B$  are arbitrary positive real numbers:

$$\begin{aligned}\text{Mag}(N_A \cdot I(t)) &= N_A \cdot \text{Mag}(I(t)), \\ \text{Mag}(N_B \cdot V(t)) &= N_B \cdot \text{Mag}(V(t)).\end{aligned}\tag{4.7}$$

Supposing a sine MEC current  $I_s(t)$  at MEF, then  $\text{Mag}(I_s(t)) = \text{Mag}(I(t))$  and  $\text{Mag}(V_s(t)) = \text{Mag}(V(t))$  are satisfied for capacitance  $C$  dominant impedance. Then for another capacitance  $C'$  dominant impedance

$$C' = N_C \cdot C \quad (N_C > 0),\tag{4.8}$$

the corresponding voltage magnitude for  $I_s(t)$  is:

$$\text{Mag}(V'_s(t)) = \frac{\text{Mag}(I_s(t))}{C' \cdot 2\pi f_{cap\_equ}} = \frac{\text{Mag}(V_s(t))}{N_C}.\tag{4.9}$$

Also,  $\text{Mag}(V(t))$  is inversely proportional to  $C$ , which can be explained using Fourier series of  $V(t)$  and  $V'(t)$ , where  $V'(t)$  is the voltage profile for  $C'$ . The coefficient for the same trigonometric function is  $N_C$  times different. Combining this relation with (4.7),  $\text{Mag}(V'(t))$  becomes

$$\text{Mag}(V'(t)) = \text{Mag}\left(\frac{V(t)}{N_C}\right) = \frac{\text{Mag}(V(t))}{N_C} = \frac{\text{Mag}(V_s(t))}{N_C}.\tag{4.10}$$

Since the rightmost terms of (4.9) and (4.10) are identical,  $\text{Mag}(V'_s(t)) = \text{Mag}(V'(t))$  still holds for different capacitances with the same MEC. Therefore, Theorem 2 is proved.

### 4.3.2 MEF for Inductance Dominant Impedance

For inductance dominant impedance, supposing the magnitudes of original load current  $I(t)$  and voltage  $V(t)$  are bounded, which is always hold in actual PDNs, there must be a sine waveform current  $I_s(t)$  that has the same magnitude, that is,

$$\text{Mag}(I_s(t)) = \text{Mag}(I(t)).\tag{4.11}$$

Then,  $\text{Mag}(V_s(t))$  becomes a function of frequency for inductance  $L$  dominant impedance:

$$\text{Mag}(V_s(t)) = 2\pi L f_{ind\_equ} \text{Mag}(I_s(t)). \quad (4.12)$$

Therefore, there exists a frequency of sine waveform  $f_{ind\_equ}$  that achieves

$$\text{Mag}(V_s(t)) = \text{Mag}(V(t)). \quad (4.13)$$

Let us denote  $f_{ind\_equ}$  as inductance MEF of load current. The existence of this inductance MEF can be summarized by:

**Theorem 3** *Let  $I(t)$  be load current profile,  $V(t)$  be corresponding PDN voltage profile. If  $I(t)$  and  $V(t)$  are bounded,  $\text{Mag}(V(t))$  across inductance dominant impedance can be reproduced by current  $I_s(t) = \text{Mag}(I(t)) \cdot \sin(2\pi f_{ind\_equ} \cdot t)$ .*

Such  $I_s(t)$  is called magnitude equivalent current (MEC) of inductance dominant impedance. Furthermore, MEF value is independent of inductance value. That is:

**Theorem 4** *Let  $V(t)$  be the voltage profile for the original current profile, and  $V_s(t)$  be the voltage profile for the MEC to the original current profile. Then, for all the inductance dominant impedances,  $\text{Mag}(V_s(t)) = \text{Mag}(V(t))$  hold.*

With the definition of (4.2), the magnitudes of current and voltage satisfy the properties below, where  $N_A$  and  $N_B$  are arbitrary positive real numbers:

$$\begin{aligned} \text{Mag}(N_A \cdot I(t)) &= N_A \cdot \text{Mag}(I(t)), \\ \text{Mag}(N_B \cdot V(t)) &= N_B \cdot \text{Mag}(V(t)). \end{aligned} \quad (4.14)$$

Supposing a sine MEC current  $I_s(t)$  at MEF, then  $\text{Mag}(I_s(t)) = \text{Mag}(I(t))$  and  $\text{Mag}(V_s(t)) = \text{Mag}(V(t))$  are satisfied for inductance  $L$  dominant impedance. Then for another inductance  $L'$  dominant impedance

$$L' = N_L \cdot L \quad (N_L > 0), \quad (4.15)$$

the corresponding voltage magnitude for  $I_s(t)$  is:

$$\text{Mag}(V'_s(t)) = L' \cdot 2\pi f_{ind\_equ} \text{Mag}(I_s(t)) = N_L \cdot \text{Mag}(V_s(t)). \quad (4.16)$$

Also,  $\text{Mag}(V(t))$  is proportional to  $L$ , which can be explained using Fourier series of  $V(t)$  and  $V'(t)$ , where  $V'(t)$  is the voltage profile for  $L'$  dominate impedance. The coefficient for the same trigonometric function is  $N_L$  times different. Combining this relation with (4.14),  $\text{Mag}(V'(t))$  becomes

$$\text{Mag}(V'(t)) = \text{Mag}(N_L V(t)) = N_L \text{Mag}(V(t)) = N_L \cdot \text{Mag}(V_s(t)). \quad (4.17)$$

Since the rightmost terms of (4.16) and (4.17) are identical,  $\text{Mag}(V'_s(t)) = \text{Mag}(V'(t))$  still holds for different inductances with the same MEC. Therefore, Theorem 4 is proved.

So far, the existence of MEFs for capacitance and inductance dominant impedances have been proved, and MEFs are independent of capacitance and inductance value.

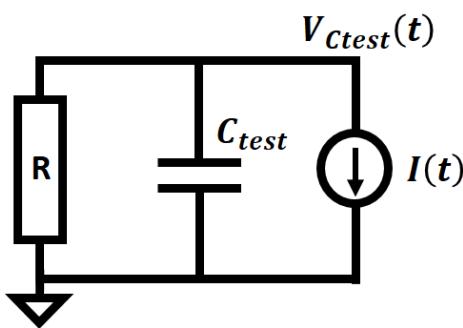


Figure 4.4: RC test circuit.

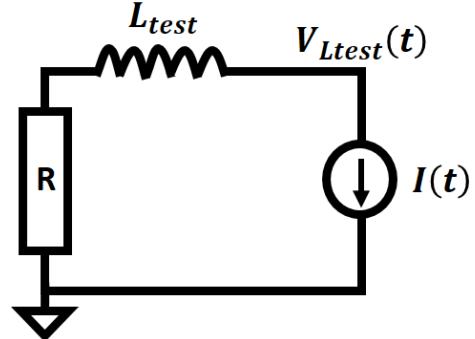


Figure 4.5: RL test circuit.

## 4.4 Derive Target Inductance and Target Capacitance

This section explains how to derive MEF and obtain target inductance and target capacitance.

MEF can be derived for any capacitance and inductance, as suggested in Theorem 2 and Theorem 4. The characterization circuit for capacitance MEF  $f_{cap\_equ}$  is shown in Fig. 4.4 and the circuit for inductance MEF  $f_{ind\_equ}$  is shown in Fig. 4.5, where the values of  $R$ ,  $C_{test}$ , and  $L_{test}$  can be arbitrarily set by designers. Given the load current profile  $I(t)$ , the output voltage  $V_{C_{test}}(t)$ ,  $V_{L_{test}}(t)$ , and their magnitudes  $\text{Mag}(I(t))$ ,  $\text{Mag}(V_{C_{test}}(t))$  and  $\text{Mag}(V_{L_{test}}(t))$  are obtained by simulation.

Note that although the values of  $C_{test}$  and  $L_{test}$  do not impact MEF thanks to Theorem 2 and Theorem 4, designers still need to select sufficiently large capacitance and inductance to ensure the circuit impedance is dominated by capacitance or inductance.

When the impedance of RC characterization circuit is capacitance  $C_{test}$  dominant,  $f_{cap\_equ}$  is derived as:

$$f_{cap\_equ} = \frac{\text{Mag}(I(t))}{\text{Mag}(V_{C_{test}}(t))} \frac{1}{2\pi C_{test}}. \quad (4.18)$$

Similarly, when the RL characterization circuit is dominated by inductance  $L_{test}$ ,  $f_{ind\_equ}$  is derived as:

$$f_{ind\_equ} = \frac{\text{Mag}(V_{L_{test}}(t))}{\text{Mag}(I(t))} \frac{1}{2\pi L_{test}}. \quad (4.19)$$

For RLC-type target impedance in Fig. 4.1, target impedance at middle frequency  $Z_{ac\_target}$  should be met between capacitance MEF  $f_{cap\_equ}$  and inductance MEF  $f_{ind\_equ}$ . For RL type-target impedance in Fig. 4.2, target impedance at middle frequency  $Z_{ac\_target}$  should be satisfied between DC and inductance MEF  $f_{ind\_equ}$ . The corresponding target capacitance and target inductance are

$$C_{target} = \frac{1}{2\pi f_{cap\_equ} Z_{ac\_target}}, \quad (4.20)$$

**Algorithm 1:** Derive target inductance and target capacitance

**Input:**  $I(t)$

*Main Routine :*

- 1: **if**  $\text{Mag}(V_{C_{test}}(t)) < \alpha \cdot \text{Mag}(V(t))_{\text{ref}}$  **then**
- 2:     Derive capacitance MEF  $f_{cap\_equ}$  by (4.18)
- 3:     Derive target capacitance  $C_{target}$  by (4.20)
- 4: **else**
- 5:     Abort with a message “Select larger  $C_{test}$ ”.
- 6: **end if**
- 7: **if**  $\text{Mag}(V_{L_{test}}(t)) > (1/\alpha) \cdot \text{Mag}(V(t))_{\text{ref}}$  **then**
- 8:     Derive inductance MEF  $f_{ind\_equ}$  by (4.19)
- 9:     Derive target inductance  $L_{target}$  by (4.21)
- 10: **else**
- 11:     Abort with a message “Select larger  $L_{test}$ ”.
- 12: **end if**

$$L_{target} = \frac{Z_{ac\_target}}{2\pi f_{ind\_equ}}. \quad (4.21)$$

The derivation of target inductance  $L_{target}$  and target capacitance  $C_{target}$  can be summarized as Algorithm 1. Now, all the parameters to define the proposed frequency-dependent target impedance have been derived, which are  $C_{target}$  in (4.20),  $L_{target}$  in (4.21),  $Z_{ac\_target}$  in (4.3), and  $Z_{dc\_target}$  in (4.1).

## 4.5 Experimental Results

This section verifies whether the proposed target impedance can satisfy the constraints of average and dynamic voltage drops.

### 4.5.1 Target Impedance Synthesis for Experiment

For this evaluation, a simulatable PDN that traces the frequency-dependent target impedance is necessary. On the other hand, the derived target impedance is a piecewise curve in frequency domain, and consequently the exact PDN realization is difficult.

Instead, T-shape RLC circuit in Fig. 4.6 is synthesized that tightly tracks the piecewise target impedance. The RL type impedance is synthesized in Fig. 4.7. When  $L_{target}$

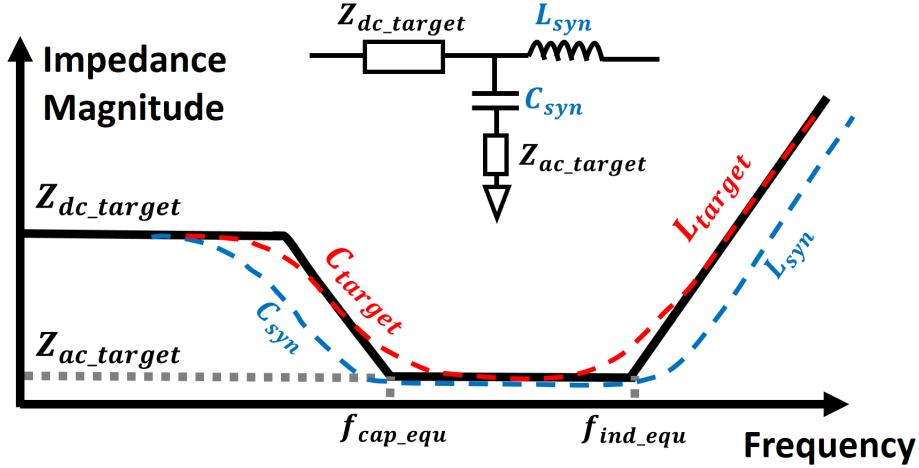


Figure 4.6: RLC-type target impedance synthesis.

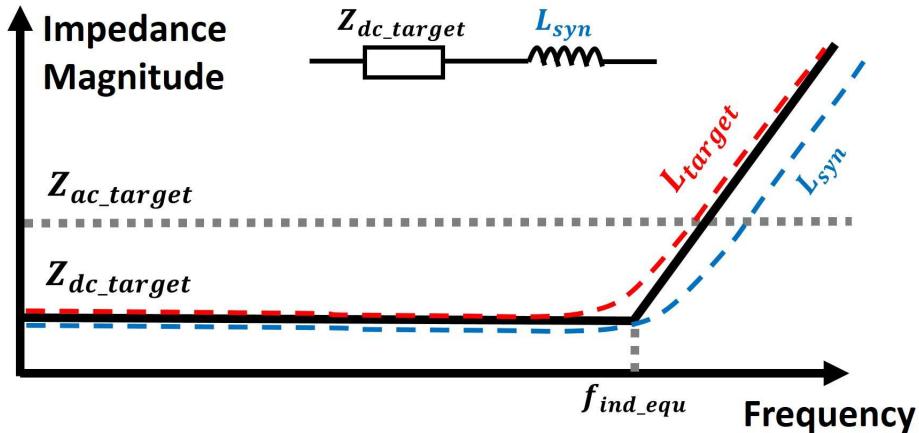


Figure 4.7: RL-type target impedance synthesis.

and  $C_{target}$  are used for the circuits, the voltage drop constraints can be violated because the impedance of the circuits is larger at the corner frequencies than the piecewise target impedance, which is depicted as the red dashed line in Fig. 4.6 and Fig. 4.7. To avoid this violation, this experiment uses larger capacitance  $C_{syn} = 10 \cdot C_{target}$ , and smaller inductance  $L_{syn} = 0.1 \cdot L_{target}$ . This minor modification can ensure the actual impedance is close to  $Z_{ac\_target}$  at the corner frequencies, which is plotted as the blue dashed line. It should be noted that this circuit synthesis is just one method and various approaches could be adopted in actual PDN design.

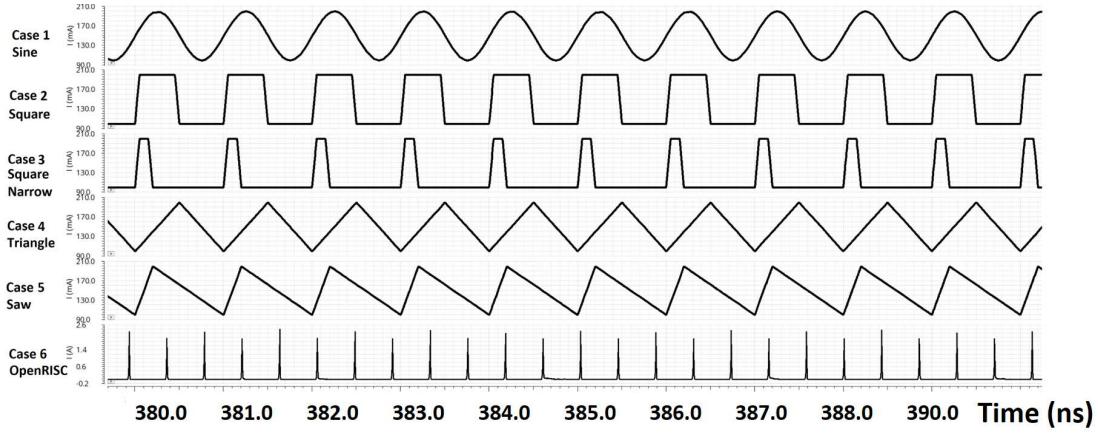


Figure 4.8: Load current profiles at 1 GHz for experiments. From top to bottom: Sine, Square, Narrow square, Triangle, Sawtooth, and OpenRISC.

#### 4.5.2 Experimental Results Compared with Design Constraints

To evaluate the applicability of the proposed methodology to various waveforms, this experiment prepared six load current profiles in Fig. 4.8. Cases 1-5 are artificial load waveforms. Cases 1-5 suppose 1 GHz operation, and their fluctuations range 100 mA to 200 mA. Case 6 is obtained from 32-bit OpenRISC [167] core logic operation.

Case 1 of sine waveform confirms that the inductance MEF and capacitance MEF are 1.0 GHz as expected. In cases 2 and 3, square waveforms with different widths of 400 ps and 100 ps are used to mimic sudden and short-duration module activations. In cases 4 and 5, triangle waveforms with different rising times of 500 ps and 200 ps aim to mimic typical digital circuit load. In the experiments, the constraints of maximum allowable voltage drop is set as  $V_{avg\_allow}=70$  mV and  $V_{dyn\_allow}=10$  mV. Given the nominal voltage as 800 mV, the minimum allowable voltage is 720 mV. Table 4.1 lists the derived values of  $Z_{dc\_target}$ ,  $Z_{ac\_target}$ ,  $C_{target}$  and  $L_{target}$ , where these four parameters define the proposed frequency-dependent target impedance. In the last two columns, the load minimal voltage  $V_{min}$  is obtained from the simulation with the synthesized T-shape RLC circuit. The average error of  $V_{avg}$  and  $V_{min}$  are 0.0003% and 0.3%, which indicates the PDNs that satisfy the frequency-dependent target impedance meet the given constraint of average and maximum voltage drops.

For OpenRISC case of 6, the load design is synthesized with NanGate 15 nm Open Cell Library at 1.2 GHz. The nominal voltage is 800 mV, and the constraints of  $V_{avg\_allow}=10$  mV and  $V_{dyn\_allow}=30$  mV are given. Then, the minimum allowable voltage is 760 mV. RLC-type target impedance is derived based on  $Z_{dc\_target}$ ,  $Z_{ac\_target}$ ,  $C_{target}$ , and  $L_{target}$ , which is listed in Table 4.1. This target impedance circuit can be synthesized as a T-shape RLC circuit in Fig. 4.6, and the simulation is run. The measured  $V_{min}=760.6$  mV, and  $V_{avg}=790.2$  mV. These results indicate that the proposed frequency-

Table 4.1: Derived target impedance parameters, and average and minimal voltages.

	$Z_{dc\_target}$ (mΩ)	$Z_{ac\_target}$ (mΩ)	$C_{target}$ (nF)	$L_{target}$ (pH)	$V_{avg}$ (mV)	$V_{min}$ (mV)
Case 1	466.6	200.0	0.8	31.8	730.0	722.5
Case 2	482.7	181.8	1.2	5.0	730.0	722.2
Case 3	608.7	117.6	0.7	5.0	729.9	720.9
Case 4	466.6	200.0	0.6	24.7	730.0	722.5
Case 5	466.6	200.0	0.5	19.8	730.0	722.5
Avg. Err.	-	-	-	-	0.0003%	0.3%
Case 6	251.9	12.5	0.35	0.01	790.2	760.6
Err.	-	-	-	-	0.02%	0.07%

dependent target impedance works well for actual processor workload including various frequency components.

## 4.6 Conclusion

This chapter has proposed a new frequency-dependent target impedance methodology that satisfies the constraints of both average and dynamic voltage drops. Given the voltage drop constraints and load current profile, frequency-dependent target impedance is derived. It is experimentally confirmed that, in the actual processor load case, the synthesized target impedance satisfies the average voltage drop constraint with 0.02% error, and overall voltage drop constraint is satisfied with 0.07% error. In the artificial load case, the synthesized target impedance satisfies the average voltage drop constraint with 0.0003% error, and the overall voltage drop constraint is satisfied with 0.07% error. These results show the effectiveness of the proposed PDN design methodology. Finally, the target impedance methodology proposed in this chapter is based on the single-port load scenario. The target impedance methodology of multi-port multi-stage PDN can be the next research step.



# Chapter 5

## Chip Load Model for PDN Verification and Exploration

This chapter proposes a chip load model for PDN verification and exploration purpose. To expose the on-chip noise-timing impact to PDN designers, the proposed chip load model provides the on-chip timing information, replays detailed voltage-dependent current profile and the inter-core operation mode variation with a short run-time.

### 5.1 Introduction

With the scaling down of the technology node, VLSI timing sensitivity to supply noise becomes more and more severe. Traditional voltage guard bound based methodology is inefficient for PDN verification and exploration since the worst voltage drop does not necessarily reflect the actual worst timing delay [44–46]. To help PDN designers to find potential design issues and avoid over/under-design, on-chip timing impact and detailed current-voltage profile need to be evaluated over various operation scenarios with a compact yet accurate chip load model.

The chip load model should meet two major requirements. Firstly, the chip load model needs to consider voltage-current-timing interdependency. Because in actual circuits, the supply noise affects chip timing performance such as clock latency and path delay as discussed in Section 1.2.2. Secondly, the chip load model should have an interface that can easily and flexibly manipulate the operation modes of individual cores, which contributes to find unexpected noise and consequent timing behaviors. In multi-core designs, for example, there are many combinations of mode transitions. Also, their transition timings could affect noise magnitude and timing performance.

This chapter proposes a chip load model, which can replay the on-chip timing information such as critical path delay, timing slack, and global clock skew, meanwhile, considering voltage-current-timing interdependency and operation mode transitions. The proposed modeling method can be scaled to large chip designs such as a multi-core sys-

tem. Furthermore, a control logic interface and critical path replica are introduced in the load model so that PDN designers can assess the on-chip timing information and explore the noise impact in different multi-core operation modes and their transitions. In terms of the simulation performance, compared with the transistor-level model, the model achieved over 300X run-time reduction in a test case. Compared with the current source model, the correlation of the current profile, current peak, and timing data is significantly improved. Furthermore, in the experiment, the proposed model illustrates the critical path slack variation caused by the mode transition process and land side capacitor (LSC) configurations. This usage example demonstrates LSC boosts processor clock frequency.

## 5.2 Multi-Core Chip Load Modeling

This section describes the details of the proposed multi-core chip load model. The overview of modeling flow is explained in subsection 5.2.1. Target multi-core system and usage model are explained in 5.2.2. Individual core load model is constructed in subsection 5.2.3. Detailed model characterization and simulation procedure are covered in subsections 5.2.4 and 5.2.5, respectively.

### 5.2.1 Overview of Chip Load Modeling Flow

Fig. 5.1 shows the overall modeling flow for a multi-core chip load. A chip load model consists of on-chip PDN model and switching circuit model. The on-chip PDN model consisting of various RLC components is used to deliver power to switching circuits, where this on-chip PDN model is supposed to be given to the flow. The procedure of switching circuit modeling is shown as gray blocks in Fig. 5.1. To complete the switching circuit modeling, three kinds of input materials are necessary, and they are explained in the following.

The first input material is the current profiles which are necessary to construct sub-models for clock path and data path, respectively. The current profiles for clock path are prepared over different voltage levels, and the current profiles for data path are prepared over different supply voltage levels and operation modes (for example, shut-down, clock-gated, full function, reset, etc.), where the mode selection is design dependent and the designers need to choose the modes that consume large and small power. Here, the current profiles are generated by transistor-level SPICE simulation in this work, but there are speed-up solutions provided by commercial EDA tools, which claims a reasonable time for current profile preparation. The second input material is transistor-level SPICE netlist to extract parasitic impedance. The third one is a set of critical path sub-circuits to generate sub-models replaying the worst cycle-by-cycle slack. With these three inputs, the voltage-current-timing dependent load model for individual core circuit is constructed.

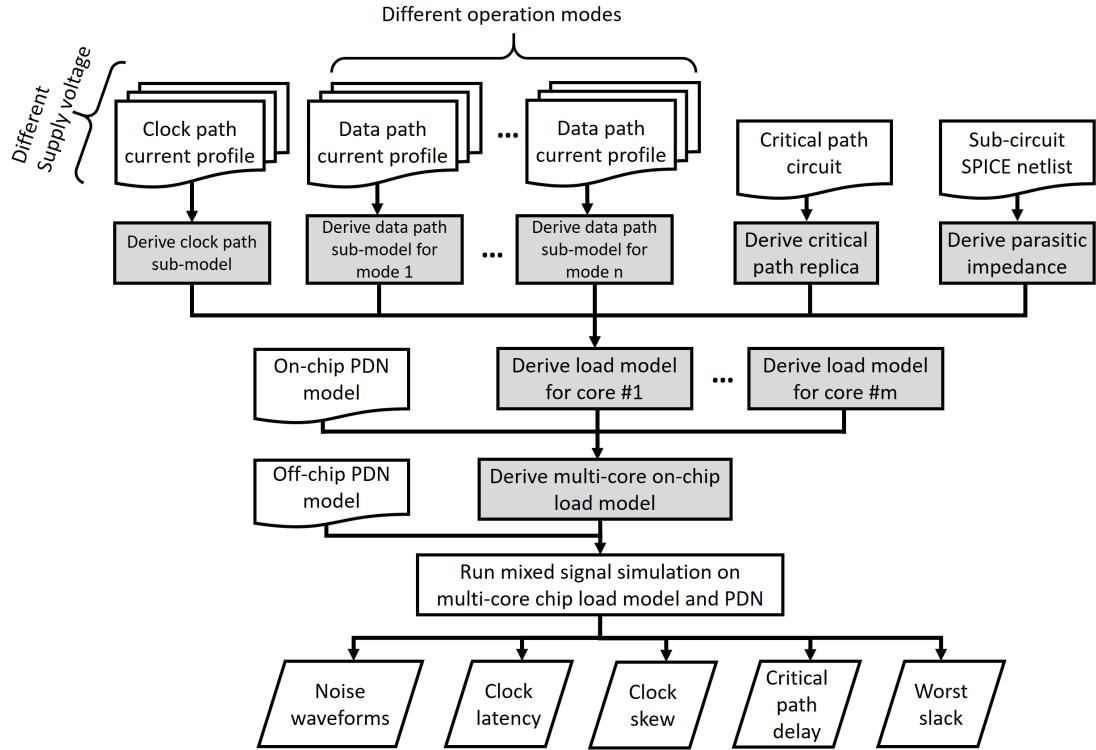


Figure 5.1: Flow of multi-core chip load modeling.

The same process is performed on other cores. By combining multi-core circuit model, which consists of multiple individual core models in parallel, with other on-chip PDN components such as bumps and PG meshes, the multi-core chip load model is built up. Finally, the on-chip load model is connected with off-chip PDN to form PDN system. Mixed signal simulation is executed for PDN system to generate on-chip and off-chip noise waveforms and on-chip timing information of clock latency, clock skew, critical path delay, and worst slack.

In this flow, the switching circuit model and on-chip PDN components can be constructed from sub-circuit level to individual core circuit level depending on the granularity of the provided circuit current profile and on-chip PDN model. Note that the granularity of the current profiles and on-chip PDN model affects simulation run-time and model construction time. Appropriate granularity should be selected such that large power operation and mode transitions inducing large current variation can be reproduced. Without losing the generality, in the remaining of this chapter, the model is built up from the individual core circuit level.

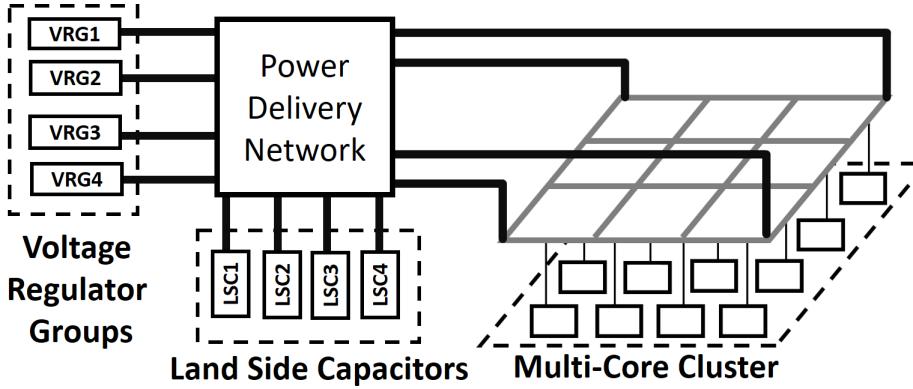


Figure 5.2: An example of block diagram of power delivery network for multi-core system.

### 5.2.2 Target Multi-Core PDN System and Usage Model

Let us illustrate a usage model with an example of multi-core system. The system block diagram is exemplified in Fig. 5.2. Suppose this example system is powered by multiple-phase voltage regulators separated into several voltage regulator groups (VRG). The supply voltage is delivered across the board and package, which are represented by the multi-port PDN in the diagram. The output of PDN is connected to on-chip power-ground mesh that supplies power to each core. Decoupling capacitors are attached to PDN at various locations. In Fig. 5.2, land side capacitor (LSC), which is gaining its importance in modern high-performance chips, is depicted. Tasks for PDN designers may include determining LSCs.

The multi-core cluster has many operation modes and their transitions. Individual cores may be activated or deactivated by clock and power gating according to environment and application requirements, and their workloads are scheduled and distributed by, for example, an operating system. Also, supply voltage and clock frequency may be controlled for each core or a group of cores. Such variations on PDN configuration and operation mode transitions can affect power supply noise and consequently impact chip timing. The proposed chip load model aims to provide timing information, such as clock skew, clock latency, path delay, and worst slack, to off-chip PDN designers so that, for example, various configurations of LSCs can be explored from a chip performance point of view.

The proposed load model is composed of multiple individual core load models. A high-level structure of individual core load model is depicted in Fig. 5.3, where the detail will be explained in the next subsection. The proposed model uses a time-voltage-variant resistor to reproduce voltage-dependent load current taking into account voltage-dependent switching delay for a given operation mode. There are multiple time-voltage-variant resistors, and they are enabled or disabled by control logic interface so that mode

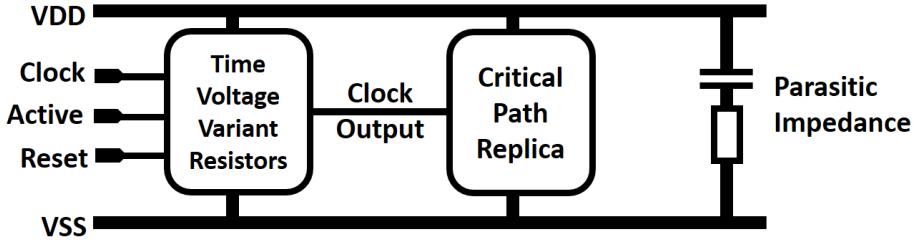


Figure 5.3: Overall structure of individual core load model.

transition is triggered. Critical paths are represented by the critical path replica module to replay critical path delay. Also, parasitic and intrinsic decoupling capacitances are modeled in Fig. 5.3.

Instantiating multiple individual core models, a multiple core load model is organized as a core cluster with a global clock distribution network, which is also modeled as time-voltage-variant resistors. Hence, the global and local clock latency, skew, and path delays can be computed with simulation. The next subsection will describe the details of the individual core load model.

### 5.2.3 Individual Core Load Model

This section discusses the details of the individual core load model. As discussed in the previous section, the main challenge for a single core load model is to replay the interdependency between voltage, current, and timing. Here, the interdependency modeling challenge is divided into sub-tasks. Firstly, for the current profile and supply voltage interdependency, it is necessary to model the voltage-dependent equivalent resistance of switching transistors. With this voltage-dependent resistance, the interdependency between the current profile and the supply voltage is naturally considered in the circuit simulation. Secondly, for the voltage-timing interdependency, it is necessary to develop clock path model and critical data path model that take into account supply voltage. Combining the clock latency and data path delay, the on-chip timing information can be provided. Finally, the current profile and timing should be aligned. Especially the switching peak current, which dominates the current profile, should be aligned with the clock latency. This task is achieved by the resistance profile method. The individual core load model is composed of three sub-models as explained with Fig. 5.3. The time-voltage-variant resistor model is responsible for reproducing the switching current in time domain. Changing the active and inactive time-voltage-variant resistor models corresponds to operation mode transition, which is triggered by control signals such as set or reset. The critical path replica model takes the output clock with latency, and reproduces the propagation delay in a set of the representative critical paths. The parasitic impedance is responsible for reproducing the voltage-current response in high

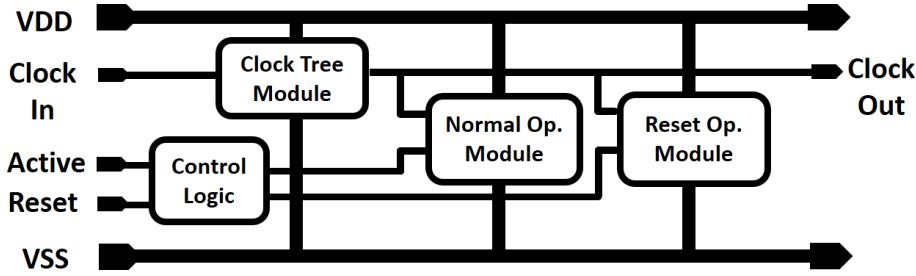


Figure 5.4: Time voltage variant resistors model structure.

frequency-domain.

Among the three components, developing the time-voltage-variant resistors is the key challenge to replay the interdependency between clock latency, current profile and supply voltage. This challenge is addressed by proposing a scaled resistance profile (RP) method, which will be explained in the subsections 5.2.3.1 to 5.2.3.4. Parasitic impedance is described in subsection 5.2.3.5 followed by critical path replica in subsection 5.2.3.6.

### 5.2.3.1 Time-Voltage-Variant Resistor Modeling

This section proposes a scaled profile method to model the time-voltage-variant resistor. The inside structure is shown in Fig. 5.4. The sub-modules of chip clock tree and data path are modeled separately.

In this diagram, only two modes of normal and reset operation are offered for simplifying the explanation. A reset signal is inputted to enable or disable the sub-model for different operation modes. The active signal is used to turn-on or shut-down the model. This structure is expandable for additional modes and sub-modules.

First, we define the RP element by a pair of  $(t_n(V_{DD}) \ r_n(V_{DD}))$ , where  $t_n$  is time in simulation and  $r_n$  is the equivalent load resistance.  $t_n$  and  $r_n$  are functions of supply voltage  $V_{DD}$ . The simulator updates the resistance  $r_n$  at  $t_n$  according to  $V_{DD}$ , and naturally deduces current by Ohm's law. Supposing a core load operation is composed of  $N$  RP elements, we define **RP** as a vector pair.

$$\mathbf{RP} = (\mathbf{T}_N \ \mathbf{R}_N), \quad (5.1)$$

where  $\mathbf{T}_N$  and  $\mathbf{R}_N$  are time and resistance vectors, respectively. Each RP element pair consists of  $t_n \in \mathbf{T}_N$  and  $r_n \in \mathbf{R}_N$ . The following two subsections explain the resistance vector modeling and time vector modeling, respectively.

### 5.2.3.2 Resistance Vector Modeling

Given a sub-module switching circuit,  $N_{tr}$  transistors are conductive. Suppose  $V_{DS}$  over a conductive transistor is small, and supply voltage  $V_{DD} \approx V_{GS}$ . Then, the equivalent resistance  $r(V_{DD})$  can be expressed by

$$r(V_{DD}) = \frac{V_{DD}}{\sum_{i=1}^{N_{tr}} I_i} \approx \left( \sum_{i=1}^{N_{tr}} \frac{(V_{DD} - V_T)}{k_i} \cdot \left( \frac{W_i}{L_i} \right) \right)^{-1}, \quad (5.2)$$

where  $I_i$ ,  $k_i$ ,  $L_i$  and  $W_i$  are drain current, conductivity factor, channel length, and channel width of individual transistors, respectively, and  $V_T$  is threshold voltage. From (5.2), the equivalent resistance of a switching circuit can be approximated to a function of  $V_{DD}$ . Meanwhile, since the equivalent resistance can be also derived from the supply voltage level and current profile via Ohm's law, the resistance can be expressed with a scaling factor by

$$r(V_{DD}) = r(V_0) \cdot SR(V_{DD}), \quad (5.3)$$

where  $V_{DD}$  is supply voltage,  $r(V_0)$  is the equivalent resistance derived from current profile at nominal supply voltage  $V_0$ , and  $SR(V_{DD})$  is the piecewise resistance scaling function fit from voltage and current profiles at different  $V_{DD}$  levels.

Fig. 5.5 exemplifies the advantage of this scaling method over conventional methods. A four-stage clock tree is selected for demonstration, in which, four different modeling methods are compared. The result labeled transistor-level SPICE model is obtained by simulating the transistor-level clock tree netlist, and it is the reference. The current source model is based on the current profile that is obtained from the transistor-level SPICE simulation result at nominal voltage. The RC model is constructed according to [94], and parameters are tuned manually so that clock latency and peak switching current are equal with the transistor-level simulation result at nominal voltage. The proposed model uses Verilog-A to implement the time-voltage-variant resistor. The resistance vector is scaled according to (5.3). With these four models, we varied the supply voltage level and measured the peak switching current for 100 clock cycles. Then, we divide the supply voltage by the averaged peak switching current to obtain the equivalent resistance. From the result, we can see that the RC model and current source model underestimate the resistance at the low supply voltage and overestimate it at the high supply voltage, and consequently the current is misestimated. On the other hand, the proposed model based on scaled resistance correlates closely with the transistor-level SPICE model simulation result as is expected.

### 5.2.3.3 Time Vector Modeling

Suppose a given path delay  $D$  is divided into  $N$  intervals and  $\Delta t_n$  denotes the  $n$ -th interval. Assuming intervals are sufficiently short, the interval duration is determined by average voltage  $V_{An}$  during the interval since the interval is impacted by transistor

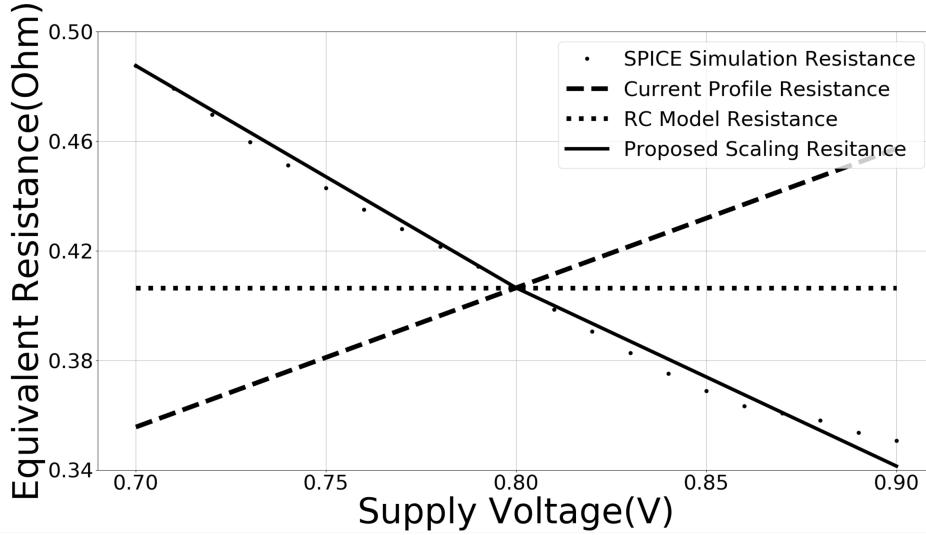


Figure 5.5: Comparison of equivalent resistance during clock switching. Constant supply voltage varies from 0.70 V to 0.90 V.

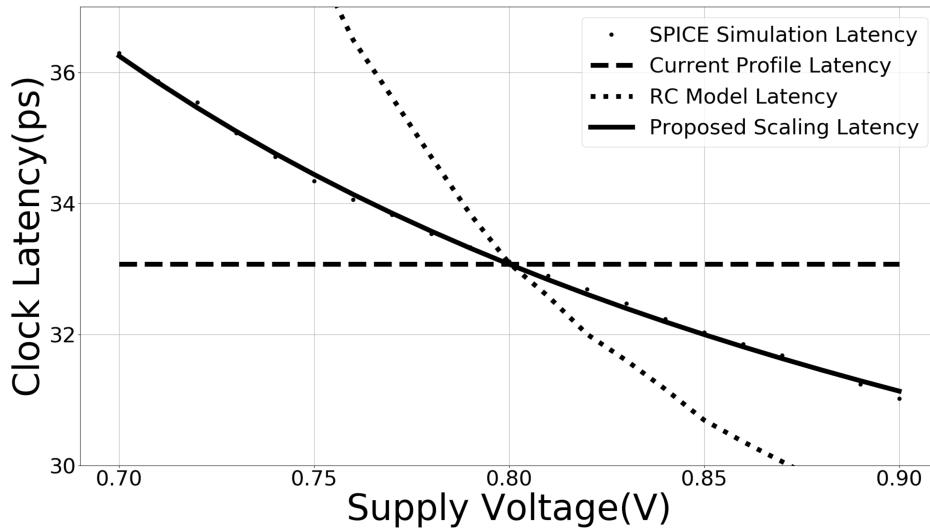


Figure 5.6: Clock latency estimation comparison. Constant supply voltage varies from 0.70 V to 0.90 V.

switching speed. This transistor switching includes RC charging and discharging processes with RC time constant, and hence the interval can also be scaled by time scaling function similar to resistance vector elements.

$$\Delta t_n(V_{An}) = \Delta t_n(V_0) \cdot ST_n(V_{An}), \quad (5.4)$$

where  $ST_n(V_{An})$  is the time scaling function for  $n$ -th interval. When the intervals are evenly distributed along the path, a single time scaling function  $ST(V_{An})$  can be used as the representative. In this case, the path delay is expressed as

$$D = \sum_{n=1}^N (\Delta t_n(V_0) \cdot ST(V_{An})). \quad (5.5)$$

Then, the time vector element  $t_n$  becomes

$$t_{n+1} = t_n + \Delta t_n(V_0) \cdot ST(V_{An}). \quad (5.6)$$

At a constant supply voltage  $V_{DD}$ , path delay (5.5) can be simplified as

$$D(V_{DD}) = D(V_0) \cdot ST(V_{DD}) = \sum_{n=1}^N \Delta t_n(V_0) \cdot ST(V_{DD}). \quad (5.7)$$

Time scaling function  $ST(V_{DD})$  can be extracted from the circuit simulation or static timing analysis with libraries at different voltages. With (5.3) and (5.6), designers can scale the resistance profile of (5.1), and deduce the clock latency under both constant supply voltage and dynamic supply noise by (5.7) and (5.5).

Fig. 5.6 shows the estimated latency of the four-stage clock tree. The transistor-level SPICE model, current source model, and RC model are constructed with the same configurations as Fig. 5.5. The proposed model uses Verilog-A to implement the time-voltage-variant resistor. The time vector is scaled according to (5.7). We can see RC model and current source model either over- or under-estimate the path delay under different supply voltage. The proposed model based on scaled latency, on the other hand, correlates closely with transistor-level SPICE simulation result.

#### 5.2.3.4 Operation Mode Transition

In the multi-core cluster, an individual core may transit across various operation modes. These modes have different current consumptions and then generate different dynamic supply noises. To replay the voltage-current-timing behavior around the mode transition, the resistance profile is prepared for each operation mode. When a core transits from an original mode to a new mode at simulation time  $t$ , the RP module of the original mode is disabled, which means the current through this RP module is set to zero. Meanwhile, the RP module of the new mode is activated, and the equivalent resistance of this RP module will be hereafter updated by the simulation engine. Such a transition process can be described with Verilog-A logic interface along with traditional Verilog test bench.

An example of mode transition is described in Algorithm 2. Suppose a data path RP module has three operation modes, which are shut-down mode, reset mode, and normal mode. The mode transition can be controlled by two signal pins named as “Reset” and “Active”. Depending on the logic level of control signal pins, the intended RP module is scheduled for simulation.

**Algorithm 2:** Operation Mode Transition Algorithm

**Input:** *Reset, Active*

*Main Routine :*

```

1: if Active signal is not set then
2:   Enable shut-down mode RP module
3:   Disable other modes' RP module
4: else
5:   if Reset is enabled then
6:     Enable reset mode RP module
7:     Disable other modes' RP module
8:   else
9:     Enable normal operation mode RP module
10:    Disable other modes' RP module
11:   end if
12: end if

```

### 5.2.3.5 Parasitic Impedance Modeling

For the parasitic impedance part, the equivalent circuit model shown in Fig. 5.7 is characterized with small signal analysis, where  $C_1$  and  $R_1$  represent the parasitic impedance and  $R_2$  is chip leak resistance.

Let us show an example of the extracted parasitic impedance of the processor core used for the experiments in the next section. By sweeping frequency of the small AC signal from 1 kHz to up to 1000 GHz, the equivalent impedance is obtained as Fig. 5.8. Then, the parameters  $R_1$ ,  $C_1$ , and  $R_2$  are derived by least squares fitting. Since the leakage current is included in RP,  $R_2$  is removed and only  $C_1$  and  $R_1$  are kept as the parasitic impedance part.

### 5.2.3.6 Critical Path Replica Modeling

The critical path replica structure is demonstrated in Fig. 5.9. The replica interface will duplicate the clock signal, supply voltage (VDD) and ground voltage (VSS) to the critical path circuit. Therefore, the critical path circuit is isolated from the main power supply. The replica interface is implemented in Verilog-A. The critical path circuit may accommodate a set of critical paths, and they can be, for example, a transistor-level netlist or a mathematical model. This work simply uses transistor-level netlist to model

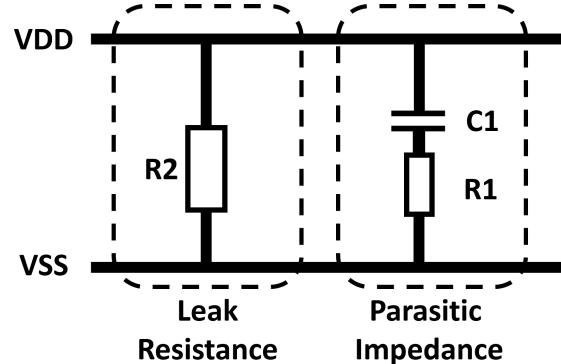


Figure 5.7: Parasitic impedance model.

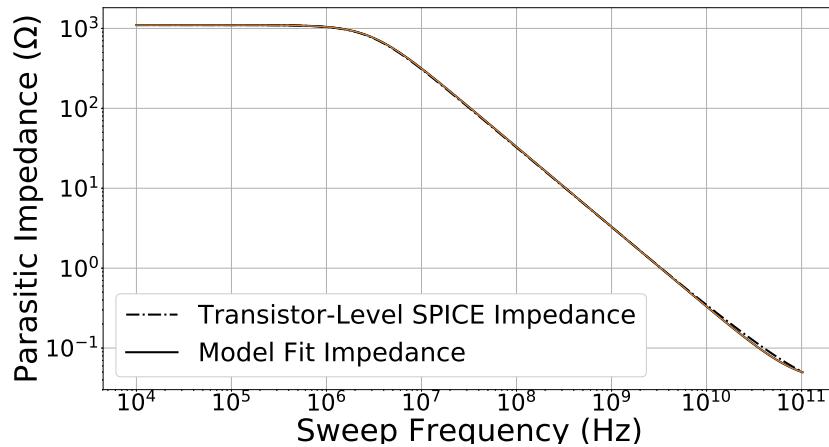


Figure 5.8: Parasitic impedance extracted by small signal analysis.

the set of critical paths. These critical paths are selected based on static timing analysis at various supply voltages. More sophisticated critical path selection and synthesis methods are well discussed in [16, 17, 120, 156–159]. During the model simulation, the worst critical path slack is measured for each cycle.

The multi-core chip load is composed of individual core load models. Also, a global clock distribution network is modeled as a time-voltage-variant resistor model and attached to the multi-core chip load. Then, the clock skew of  $n$ -core chip load is derived by:

$$Skew = \max |D_i - D_j| \quad (\forall i, j \in n), \quad (5.8)$$

where  $D_i$  and  $D_j$  are the clock latency to the clock terminals of sequential elements in cores  $i$  and  $j$ , respectively. The clock latency is derived by (5.5).

Since the critical path delay is reproduced by critical path replica model, the worst

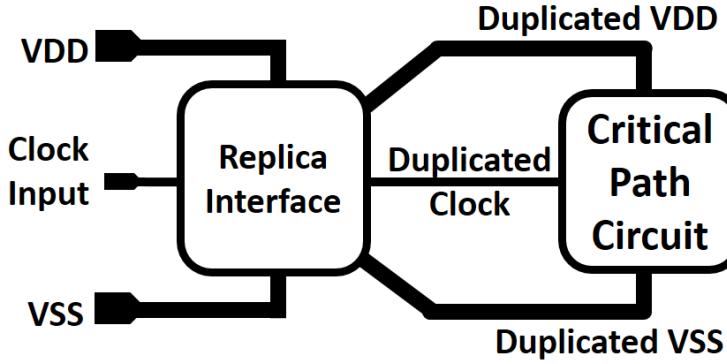


Figure 5.9: Critical path replica model.

timing slack at each clock cycle is derived by:

$$\text{Slack}(i) = T_{clk}(i+1) - T_{clk}(i) - T_{setup} - T_{path}(i), \quad (5.9)$$

where  $T_{clk}(i)$  is the time of the clock rising edge for  $i$ th clock cycle,  $T_{setup}$  is the setup time of sequential element, and  $T_{path}$  is the critical path delay.

#### 5.2.4 Core Load Model Characterization

This subsection summarizes the characterization procedure of the individual core load model. The individual core load model is composed of three sub-models demonstrated in Fig. 5.3. The parasitic impedance model is characterized by small signal analysis. The critical path replica model can be characterized from static timing analysis. As for the time-voltage-variant resistor model, both resistance vector and time vector need to be characterized to form the resistance profile. The items and scaling functions of resistance vector and time vector are characterized through the process below.

- Step 1:** Generate current profile at nominal voltage  $V_0$ , and measure path delay or clock latency  $D(V_0)$ .
- Step 2:** Convert current profile into resistance profile pair  $(r(V_0) t_n(V_0))$ .
- Step 3:** Obtain current profile for tens of clock cycles at different supply voltages, measure clock latency  $D(V_{DD})$ , and derive resistance profile pair  $(r(V_{DD}) t_n(V_{DD}))$ .
- Step 4:** Run fitting process and generate scaling functions for resistance vector and timing vector, according to (5.3) and (5.4).
- Step 5:** Compose resistance profile.

In Step 1, the current profile at a constant voltage can be generated by either traditional transistor-level simulation or more sophisticated power estimation tools. In Step 2, the resistance profile pair  $(t_n(V_0) \ r_n(V_0))$  is constructed with temporal discretization and Ohm's law. In Step 3, tens of clock cycle simulation is needed to derive latency and resistance profile as sample data, which will be used to build the scaling functions in Step 4. In Step 5, the final resistance profile is composed of time and resistance vectors defined by (5.1).

### 5.2.5 Resistance Profile Simulation Procedure

Suppose a resistance profile during a clock cycle is composed of  $N$  RP elements. Once a clock rising edge is detected, the first RP element will be selected to deduce equivalent resistance as  $r_1(V_{DD})$ . Then, the time to update the next RP element is also deduced with (5.4). Once resistance is determined at a given simulation time, the current value is computed by Ohm's law in a circuit simulator. Such procedure is performed until all the RP elements  $(t_n(V_{DD}) \ r_n(V_{DD}))$  are simulated. As a special case for clock tree RP module, once the simulation time after the clock signal is given is larger than the clock path delay, which is derived by (5.5), the input clock signal will be copied to the output clock signal port. Hence, the clock propagates with the computed clock path latency. Finally, the output clock signal is duplicated to critical path replica model, and critical path slack is measured in each cycle by (5.9). The RP simulation procedure can be described in Algorithm 3.

This algorithm can be implemented with Verilog-A, and hence our model can be co-simulated with Verilog and SPICE modules. By applying a similar approach to other subcircuit modules or modes, one can model larger-scale complex processors.

**Algorithm 3:** RP Module Simulation Procedure

**Input:**  $V_{DD}$ ,  $V_{in\_signal}$

**Output:**  $I$ ,  $V_{out\_signal}$

*Initialization :*

1: Set leak resistance

*Main Routine :*

2: **if**  $V_{in\_signal}$  is changed **then**

3:   **for**  $n = 1$  to  $N$  **do**

4:     Obtain  $r_n$  and  $t_n$  from  $RP$ .

5:     Calculate the resistance value with (5.3), and time interval with (5.4).

6:     Schedule the next resistance update time, which is derived by (5.6).

7:     Copy the  $V_{in\_signal}$  value to  $V_{out\_signal}$  once the time after the input signal is given becomes larger than the path delay in (5.5).

8:   **end for**

9: **end if**

## 5.3 Experimental Results

This section shows experimental results to validate the proposed model. The first part demonstrates the simulation quality for individual core load model. The second part conducts system-level experiments by building up a multi-core PDN system with chip load model and off-chip PDN for demonstrating the timing impact under different off-chip PDN configurations.

### 5.3.1 Individual Core Experiment

For the individual core experiment, a 32-bit OpenRISC [167] processor is prepared and synthesized with NanGate 15-nm open cell library [168]. The number of cells is over 17k, the clock frequency for the core processor logic is 1.2 GHz, and the average clock latency is 114.9 ps at 0.8 V supply voltage. A CRC checksum program is given to OpenRISC as workload. The characterization for 500-cycle operation finished within two hours in this test case. The implemented Verilog-A module is found in Appendix.

The first experiment illustrates the reproducibility of current and voltage waveforms comparing current source model, full SPICE netlist simulation, and the proposed on-chip load model. Then, the voltage source of 0.7V is connected to a two-port PDN described by S-parameter. The chip load model is connected to the output load port

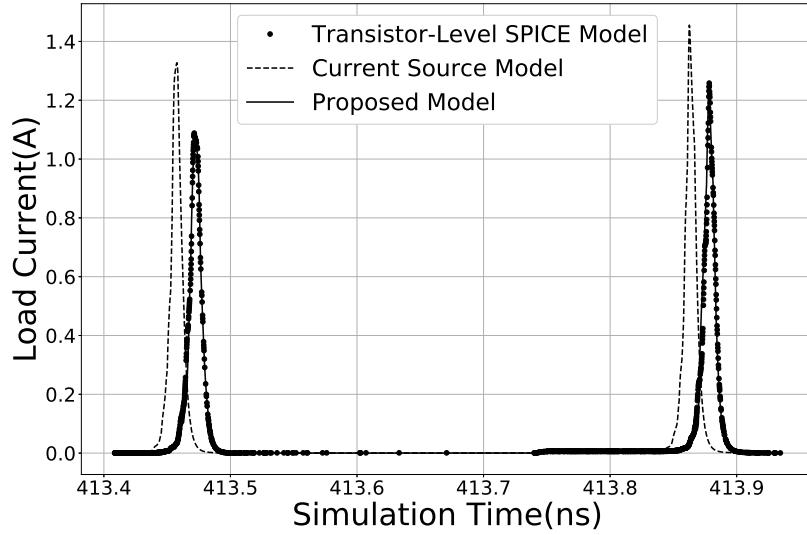


Figure 5.10: Current waveform comparison within one clock cycle.

of PDN. The current waveform is shown in Fig. 5.10, and the load voltage waveform is shown in Fig. 5.11. The waveform of the proposed model is depicted with solid line, from which we can find both the current and voltage waveforms correlate closely with the transistor-level SPICE simulation result (dot line). The interdependency among voltage, current, and switching time is also replayed. On the other hand, the current source (dash line) overestimates voltage noise and underestimates timing delay.

The second experiment evaluates the accuracy of the individual core load model quantitatively at different supply voltages from 0.7 to 0.9 V. The results are listed in Table 5.1. This evaluation simulated 200 clock cycles. For the peak current evaluation, the errors for 400 current peaks are calculated and averaged, where 400 peaks are 200 clock cycles multiplied by two peaks per clock cycle. The average error for individual peak currents is 2.4%. On the other hand, the conventional current source and RC model cannot attain such accuracy, and the average peak current errors are 17.6% and 10.5%, respectively. For the clock latency evaluation, the average latency error of the proposed model is 0.3%, whereas the average errors for the current source and RC models are 6.3% and 11.4% respectively. Especially, the current source model suffered up to 38.5% error in peak current estimation, and RC model had up to 39.2% error in latency estimation.

Thirdly, to validate the individual core load model under dynamic supply noise, a sinusoidal noise is injected with 100 mV amplitude whose frequency ranged from 100 MHz to 1 GHz, where 100 MHz is roughly 10x lower and 1 GHz is almost similar to the clock frequency. 100 clock cycles simulation is performed for both full-SPICE

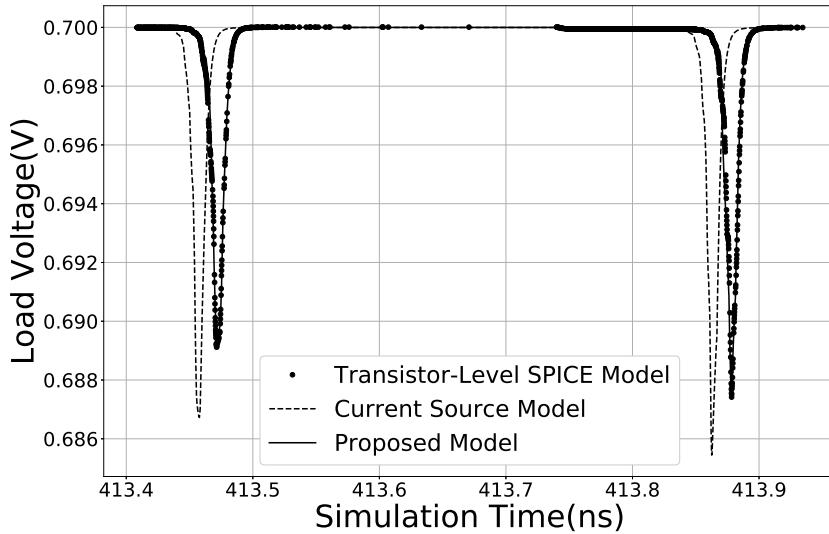


Figure 5.11: Load voltage waveform comparison within one clock cycle.

Table 5.1: Average peak load current and average clock latency comparison at various supply voltages.

Supply Volt.(V)	Peak Curr.(A)		Err(%)	Latency (ps)		Err(%)
	SPICE	Model		SPICE	Model	
0.70	1.38	1.36	1.4%	131.7	132.2	0.4%
0.73	1.54	1.50	2.7%	125.8	126.1	0.3%
0.77	1.74	1.70	2.7%	119.0	119.4	0.4%
0.80	1.91	1.87	2.0%	114.9	115.3	0.4%
0.83	2.10	2.03	3.2%	111.4	111.7	0.3%
0.87	2.35	2.27	3.3%	107.5	107.7	0.3%
0.90	2.50	2.47	1.2%	104.9	105.1	0.2%
<b>Avg.</b>	-	-	<b>2.4%</b>	-	-	<b>0.3%</b>

netlist and the proposed on-chip load model. Figs. 5.12 and 5.13 show the clock latency comparison. We can see both the clock latencies are well correlated. The average

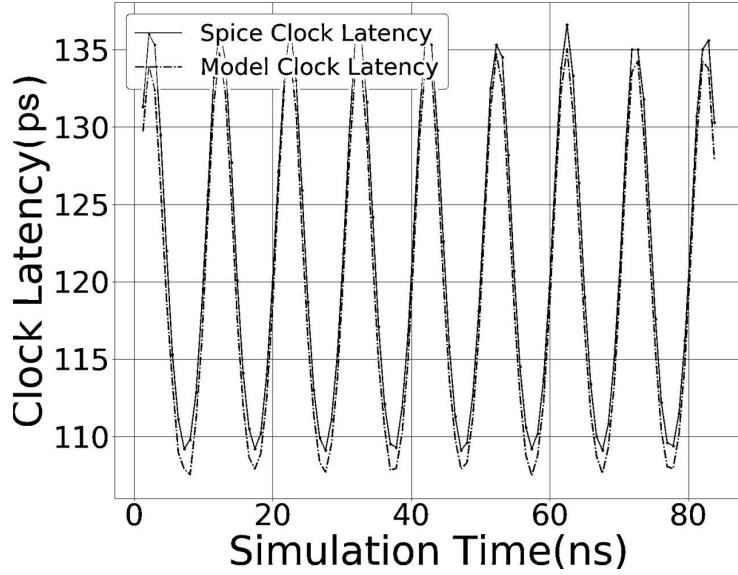


Figure 5.12: Clock latency estimation with 100 MHz supply noise.

latency errors are 1.5% for 100 MHz noise, and 2.6% for 1 GHz noise. The peak current under dynamic noise is also compared in Figs. 5.14 and 5.15. The average peak current errors are 2.3% for 100 MHz noise and 2.2% for 1 GHz noise.

### 5.3.2 Multi-Core PDN System Experiment

For larger system level experiments, a multi-core PDN system is prepared. The high-level schematic is demonstrated in Fig. 5.2, where four voltage regulator groups provide 16-phase 0.8 V DC supply voltage. The individual voltage regulator is implemented by 2:1 switched capacitor voltage regulator using CMOS model from NanGate 45-nm open cell library and capacitor components. The multi-port PDN represents PCB and package circuits and it is described by S-parameter file. The LSCs are modeled by RLC components. At the chip load side, the connection between 16-core cluster and power-ground mesh is depicted in Fig. 5.16. For each mesh grid, the segment resistance and inductance are 50.4 m $\Omega$ , and 5.6 fH, respectively. The clock signal propagates through a global clock tree shown in Fig. 5.17. The main process to construct the multi-core load model is done by python scripts, which takes around 15 minutes to convert a set of given current profiles and netlist to the chip load model. Extra manual work is also needed for writing glue logic and testbench scenarios. Assuming a template is given for the glue logic and testbench, this manual work takes minutes to hours depending on the size and complexity of the core.

Using this PDN system, the first experiment verifies the timing information accuracy

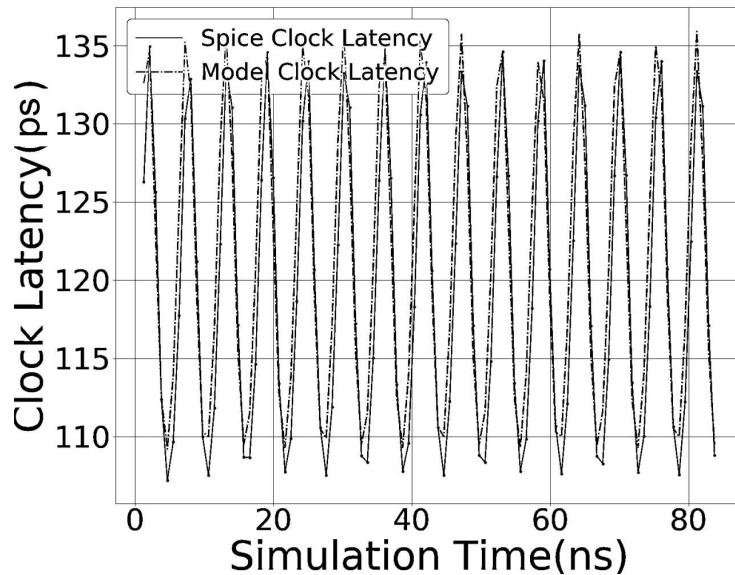


Figure 5.13: Clock latency estimation with 1 GHz supply noise.

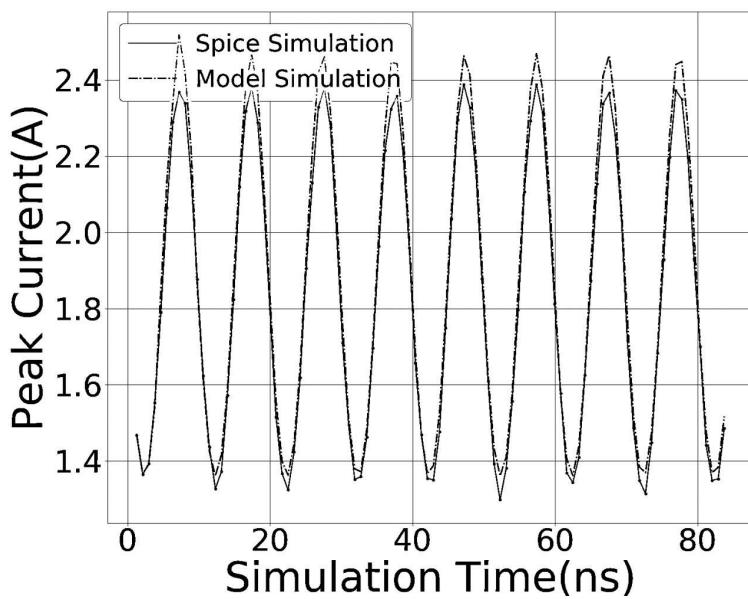


Figure 5.14: Peak current estimation with 100 MHz supply noise.

of individual core load model. The core load model is connected to the center of power-ground mesh, which is the position of core #6 in Fig. 5.16. Four current sources were

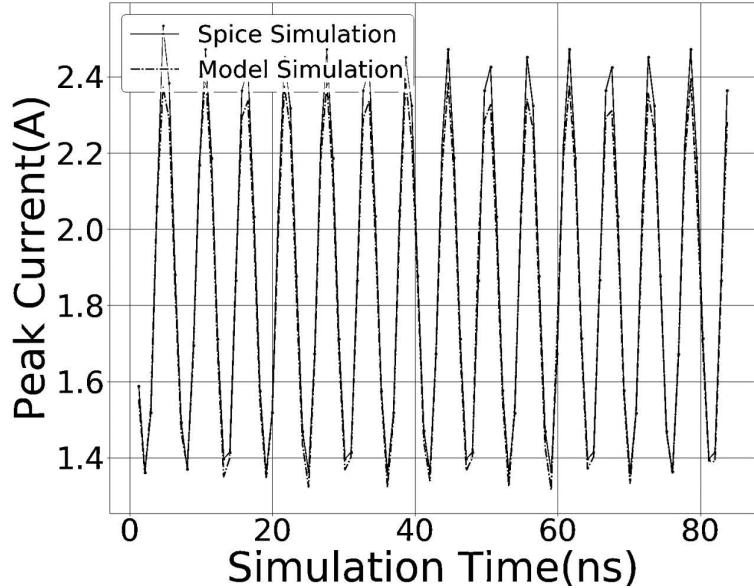


Figure 5.15: Peak current estimation with 1 GHz supply noise.

connected to the adjacent grids to mimic the transient process of neighboring cores. These current sources increase their current consumption from 40 mA to 400 mA at 470 ns, and then drop back to 200 mA in 2 ns. The cycle-by-cycle critical path slack is compared between transistor-level SPICE netlist and the proposed chip load model. The slack comparison result is shown in Fig. 5.18, where the average estimation error of the path slack is 0.1%, and the maximum error is 2.6%. In this simulation, the simulation with full transistor-level SPICE netlist takes 68,537 s, while that with the proposed model takes 172 s which means over 300X runtime reduction. Note that this runtime reduction is more significant when the system under evaluation is larger.

The next experiment evaluates the on-chip timing information for different PDN configurations and operation mode transition scenarios. In scenario 1, four cores are activated at the beginning, which are core #1, #2, #5, #6 in Fig. 5.16. Then, other twelve cores are activated simultaneously after 462 ns, followed by 15 ns reset operation mode, then switch to the normal operation mode. The CRC checksum program is used as the workload in the normal operation mode. In scenario 2, the same four cores are activated at the beginning as scenario 1, but the remaining twelve cores are activated in a gradual process, that is, every four cores are activated after 5 ns. In both scenarios, the LSC capacitance is varied from 0.08 nF to 20 nF, and then measure the critical path slack of core #6, which locates near the center of the power-ground mesh. The cycle-by-cycle slack is shown in Fig. 5.19.

From the simulation result, off-chip PDN designers can assess the LSC effectiveness

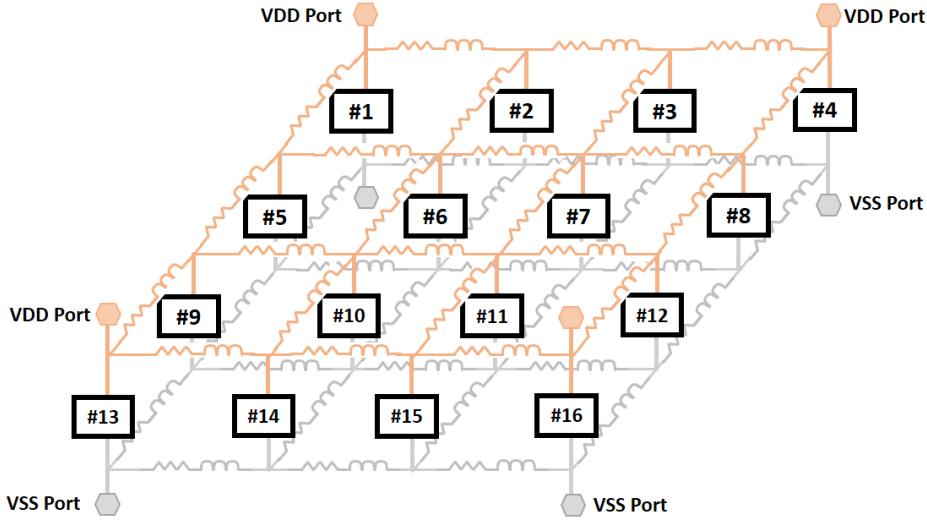


Figure 5.16: 16-core cluster with power-ground mesh.

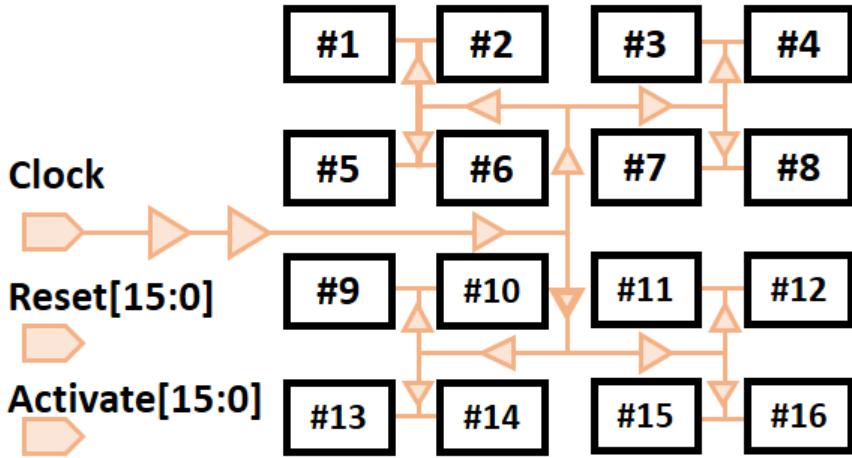


Figure 5.17: 16-core cluster with clock tree and control signal.

under different mode transition procedures. For example, when 12 cores are enabled simultaneously, at least 20 nF LSC capacitance is required to fix setup timing violation for core #6, which is shown as dot lines in Fig. 5.19. On the other hand, when the mode transition is scheduled in a gradual way, 4 nF LSC is sufficient to ensure 50 ps critical path slack. In this experiment setup, the average simulation run-time is 1087.5 s. This run-time range enables off-chip designers to explore PDN configurations over various mode transition scenarios.

The third experiment performs an experiment that tunes multi-core system performance with different PDN configurations. In this experiment, 16 core load models are constructed to form a multi-core cluster. As a core load configuration, eight cores are

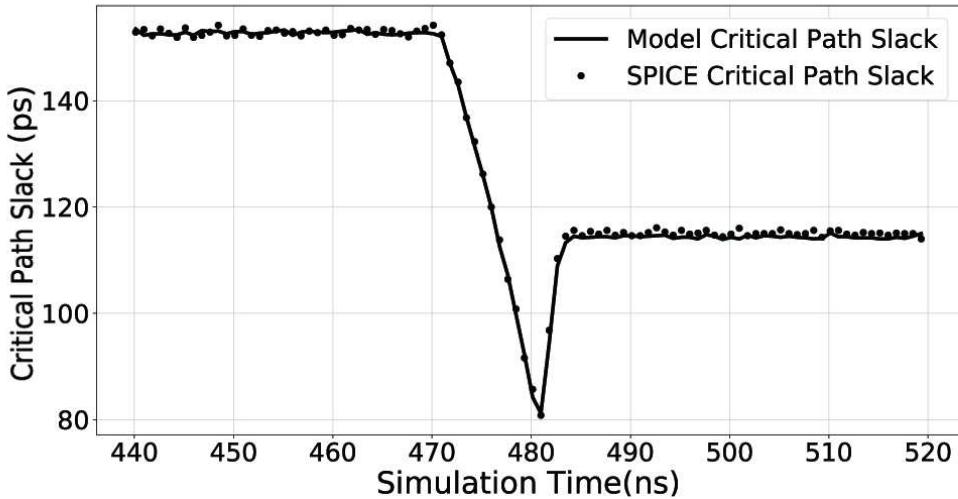


Figure 5.18: Cycle-by-cycle critical path slack comparison during transient process.

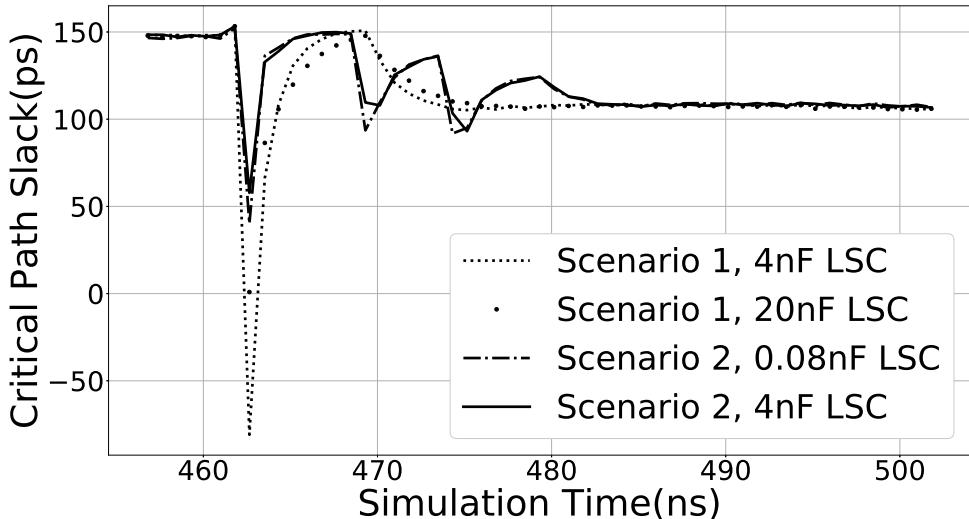


Figure 5.19: Cycle-by-cycle critical path worst slack of core #6.

turned off at the beginning, and then turn-on the remaining eight cores at 470ns. Each core switches to reset mode for 15ns before entering into normal operation mode. The CRC checksum program is used as the workload in the normal operation mode. As for off-chip PDN configuration, the input clock frequency is varied from 1.1 GHz to 1.3 GHz, and vary the LSC capacitance from 4 nF to 20 nF. The worst timing slack among the 16 cores is evaluated.

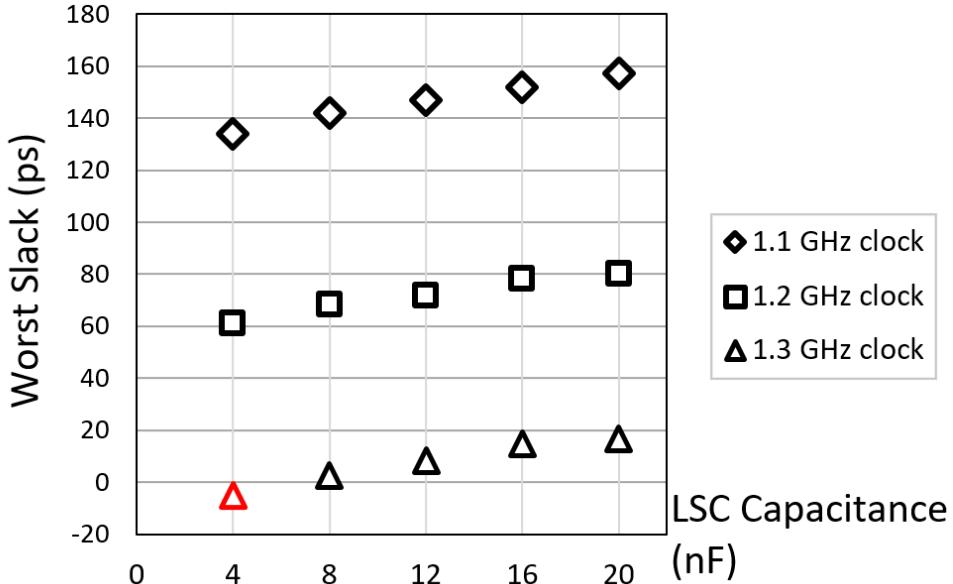


Figure 5.20: Worst timing slack under different LSC configurations. Three clock frequencies are inputted to the multi-core system.

Fig. 5.20 shows the result of the worst slack. From the simulation result, off-chip PDN designers can find the effectiveness of LSC capacitance on retrieving timing slack. For example, when LSC is increased from 4 nF to 20 nF, an extra timing slack of 20 ps is attained. When 1.3 GHz clock is driving the system, a negative timing slack of -5.1 ps, is presented by the load model, which is shown as a red triangle in Fig. 5.20. This timing data is helpful for off-chip PDN designers to assess the noise impact on chip performance. On the other hand, by increasing the LSC to 20 nF, the worst timing slack is improved to 16.7 ps, which means the chip timing constraint under 1.3 GHz frequency is satisfied with 20 nF LSC configuration. Such an off-chip PDN optimization becomes feasible with the proposed on-chip load model.

## 5.4 Conclusion

This chapter proposed a multi-core chip load model that could replay the load current and timing information under supply voltage noise. The model also supports extensive design exploration with operation mode variation and different PDN parameters.

In the single-core experiment, the clock latency is accurately replayed by the proposed chip load model. Compared with transistor-level model, the average latency errors are 1.5% under 100MHz supply noise scenario, and 2.6% for 1 GHz noise scenario. The average peak current errors are 2.3% for 100MHz noise scenario and 2.2% for 1 GHz noise scenario. In the multi-core system experiment, the proposed chip load model

can replay the timing information and the load current profile with high accuracy. Compared with transistor level model, the data path average slack error is 0.1%, and the maximum error is 2.6%. Meanwhile, over 300X runtime reduction is achieved compared with full SPICE netlist simulation. The off-chip PDN experiments also show the proposed model can guide off-chip PDN designers to tune the LSC parameters with on-chip timing information.



# Chapter 6

## Conclusion

Low-noise power distribution system is highly demanded in modern VLSI designs since emergent supply noise through PDN degrades the chip timing performance or even causes malfunction. On the other hand, both power consumption and supply noise are continuously increasing with the scaling down of the technology node. Therefore, effective supply noise mitigation system and low-noise PDN design methodology are critically important to ensure robust VLSI power distribution.

There are two major challenges for designing a robust VLSI power distribution system. The first is negative loop challenge of supply noise mitigation. This challenge is caused by the large hardware and computation cost of prediction engine, and the limited voltage scaling capability of voltage regulator. The second is design gap challenge for PDN design methodology. In target impedance design stage, there lacks a proper method to bridge the time-domain voltage drop constraints and frequency-domain target impedance guidance. In the PDN exploration and verification stage, the on-chip timing information is invisible to off-chip PDN designers. The first challenge causes high hardware overhead or low prediction accuracy in a proactive noise mitigation system. The second challenge causes under- or over-designed PDN, and hence unexpected noise-timing impact arises. This dissertation addressed these two challenges by proposing the proactive supply noise mitigation system, which is presented in Chapter 2 and Chapter 3, and improving the PDN design methodology, which is discussed in Chapter 4 and Chapter 5.

Chapter 2 provided a lightweight current prediction solution, which relieves the prediction cost with high prediction accuracy. The key idea is to construct a lightweight short-term average current predictor using the decision tree. The decision tree predictor reads instructions from processor I/O, and then construct the features by deriving the instruction type history. The label is constructed from averaged current profile. Both features and label are calculated by moving average algorithm to save memory cost. Based on experimental results, a lightweight short-term current predictor is derived, which consists of six-layer decision tree regressor and achieves over 0.99 correlation for 50-cycle-ahead prediction.

Chapter 3 proposed a major-minor voltage regulator structure, which provides the fast, continuous, and wide-range voltage scaling capability using switched capacitor voltage regulators. The main contribution is to propose a major-minor voltage regulator (MMVR) structure, which consists of two SCVRs whose flying capacitance is much different. The major voltage regulator uses large flying capacitance to provide stable low-ripple supply voltage. On the other hand, the minor voltage regulator is designed as a re-configurable SCVR structure, which can provide two different load voltage levels with small flying capacitance. This special structure enables minor voltage regulator to continuously scale supply voltage using simple switching frequency modulation. The small flying capacitance makes the minor voltage regulator be integrated into chip package to speed up voltage scaling. Also, the small flying capacitance introduces less ripple during voltage scaling operation mode. According to the experiment, MMVR achieved over 3X voltage scaling range compared with traditional SCVR while the ripple is within 16 mV, which is 1.6% of load voltage.

Proactive noise mitigation is constructed by combining the result from Chapter 2 and Chapter 3. Experimental results showed that the proposed proactive noise mitigation can mitigate the supply noise within 30mV while the noise exceeds 70mV with the conventional reactive mitigation system. Also, the average supply voltage is compensated during full operation period.

Chapter 4 proposed a frequency-dependent target impedance methodology, which fills the gap between PDN voltage drop constraints and frequency domain impedance guidance. The key idea is to design the target impedance by introducing a conception of magnitude equivalent frequency (MEF). That is, instead of analyzing the detailed time-domain current waveform, a sine waveform current can be used to reproduce the same magnitude of the voltage noise. The frequency of this sine waveform is defined as MEF. The adoption of MEF can bridge the design gap between target impedance and voltage drop constraints and hence, simplify the frequency-dependent target impedance design. It is experimentally confirmed that, in the actual processor load case, the synthesized target impedance satisfies the average voltage drop constraint with 0.02% error, and overall voltage drop constraint is satisfied with 0.07% error. In the artificial load case, the synthesized target impedance satisfies the average voltage drop constraint with 0.0003% error, and the overall voltage drop constraint is satisfied with 0.07% error. These results showed the effectiveness of the proposed target impedance methodology.

Chapter 5 proposed a chip load model that can provide the on-chip timing information for off-chip PDN verification and exploration purpose. The main idea is to use time-voltage-variant resistor to reproduce voltage-dependent load current taking into account voltage-dependent switching delay for a given operation mode. Then, multiple time-voltage-variant resistors are enabled or disabled by control logic interface so that mode transition is triggered. Critical paths are represented by the critical path replica module to replay critical path timing delay. Also, parasitic and intrinsic decoupling capacitances are modeled using small-signal analysis. Hence, the global and local clock

---

latency, skew, and path delays can be computed with simulation. In the single-core experiment, the clock latency is accurately replayed by the proposed chip load model. Compared with transistor-level model, the average latency errors are 1.5% under 100 MHz supply noise scenario, and 2.6% for 1 GHz noise scenario. The average peak current errors are 2.3% for 100 MHz noise scenario and 2.2% for 1 GHz noise scenario. In the multi-core system experiment, the proposed chip load model can replay the timing information and the load current profile with high accuracy. Compared with transistor-level model, the data path average slack error is 0.1%, and the maximum error is 2.6%. Meanwhile, over 300X runtime reduction is achieved compared with full SPICE netlist simulation. The off-chip PDN experiments also showed the proposed model can guide off-chip PDN designers to tune the LSC parameters with on-chip timing information.

The works in this dissertation contribute to the robust power distribution for high-performance VLSI design. The proposed proactive noise mitigation system mitigates the emergent voltage droop in VLSIs. The PDN design and chip load modeling methods proposed in this dissertation are helpful for PDN designers to avoid the over- or under-designed PDN, and reduce the design cost and iteration time by facilitating the PDN verification and exploration process.

On the other hand, there are still several future works. One of the future work relates to the prediction accuracy and prediction length. To further improve the prediction accuracy, the larger training set is desirable from the machine learning perspective. Meanwhile, the quantitative demonstration of the actual relationship between prediction length and processor structures requires extensive experiments and feature exploration, which also needs time-consuming simulation work. These simulation and exploration works are left for future research. Another future work relates to the voltage regulator part. The actual implementation of MMVR requires detailed hardware design and system-level integration efforts. Besides, the LDO regulator has high efficiency during small-range voltage scaling, and such merit can be exploited to work with MMVR solution in future work.



# Appendix

## Critical path replica source code

### SPICE source code of critical path replica module

```
.SUBCKT REPLICA CLK_IN PATH_OUT VDD_PIN VSS_PIN
XVOLT_DUP VDD_PIN VSS_PIN VDD_DUP VSS_DUP replica
XCLK_DUP CLK_IN VSS_PIN CLK_DUP VDD_DUP VSS_DUP clkdup
XPATH CLK_DUP PATH_OUT VDD_DUP VSS_DUP CriticalPath
.ENDS
```

### Verilog-A source code of replica sub-module

```
'include "constants.vams"
#include "disciplines.vams"

module replica(vdd_in, vss_in, vdd_out, vss_out) ;
  input vdd_in, vss_in;
  inout vdd_out, vss_out;
  electrical vdd_in, vss_in, vdd_out, vss_out;

  analog begin
    V(vdd_out) <+ V(vdd_in);
    V(vss_out) <+ V(vss_in);
  end
endmodule
```

### Verilog-A source code of clkdup sub-module

```
'include "constants.vams"
```

```

'include "disciplines.vams"
module clkdup(vsignin, vssin, vout, vout_h, vout_l) ;
input vsignin, vssin, vout_h, vout_l;
inout vout;
electrical vsignin, vssin, vout_h, vout_l, vout;

analog begin
  if (V(vsignin, vssin) >= 0.5*(V(vout_h, vout_l))) begin
    V(vout) <+ V(vout_h);
  end
  else begin
    V(vout) <+ V(vout_l);
  end
end
endmodule

```

## Verilog-A source code of clock tree module

```

'include "constants.vams"
'include "disciplines.vams"
module ct_mod(clk,clk_out,vdd_pin, vss_pin) ;
input clk;
inout vdd_pin, vss_pin;
output clk_out;
electrical clk, vdd_pin, vss_pin;
electrical clk_out;
real resist = 0;
real scalep = 0;
real tscale = 1;
real tbin = 0;
integer logic_one = -1;
real switch_time = 0;
analog begin
  // initial status
  @(initial_step) begin
    resist = 4000;
    tscale = 1;
    tbin = 0;
    switch_time = 0;
    logic_one = -100;

```

```

end

// resistance scaling factor
if (V(vdd_pin, vss_pin) >= 0.8) begin
  scalep = (V(vdd_pin, vss_pin)-0.8)*(-0.63);
  end
else begin
  scalep = (V(vdd_pin, vss_pin)-0.8)*(-0.87);
  end

// latency scaling factor
if (V(clk, vss_pin) >= 0.1) begin
  // rising edge
  tscale = ((64.4 * V(vdd_pin, vss_pin)- 5.711) / (V(vdd_pin,
    vss_pin) - 0.4016) ) / ((64.4 * 0.8 - 5.711) / (0.8 - 0.4016)
    );
  tbin = ( ((64.4 * V(vdd_pin, vss_pin)- 5.711) / (V(vdd_pin,
    vss_pin) - 0.4016) ) * 1p );
end
else begin
  // falling edge
  tscale = ((63.94 * V(vdd_pin, vss_pin)- 4.814) / (V(vdd_pin,
    vss_pin) - 0.3988) ) / ((63.94 * 0.8 - 4.814) / (0.8 -
    0.3988) );
  tbin = ( ((63.94 * V(vdd_pin, vss_pin)- 4.814) / (V(vdd_pin,
    vss_pin) - 0.3988) ) * 1p );
end

if (logic_one ==1) begin
  @(timer(switch_time + (1.6p) * tscale)) begin resist =
    1907.83859005 + scalep ; end //1062520
  @(timer(switch_time + (3.4p) * tscale)) begin resist =
    1607.94889981 + scalep ; end //1062520
  @(timer(switch_time + (4.4p) * tscale)) begin resist =
    1328.7622278 + scalep ; end //1062520
  @(timer(switch_time + (5.7p) * tscale)) begin resist =
    1362.93551735 + scalep ; end //1062520
  @(timer(switch_time + (6.7p) * tscale)) begin resist =
    1411.49870211 + scalep ; end //1062520
  @(timer(switch_time + (8.1p) * tscale)) begin resist =
    1174.97437724 + scalep ; end //1062520

```

```

@(timer(switch_time + (9.4p) * tscale)) begin resist =
  1377.75465935 + scalep ; end //1062520
@(timer(switch_time + (10.5p) * tscale)) begin resist =
  1242.23158153 + scalep ; end //1062520
// ... remaining resistance profile within a cycle is omitted
end

if (logic_one ==0) begin
  @(timer(switch_time + (2.0p) * tscale)) begin resist =
    1620.64851725 + scalep ; end //1105020
  @(timer(switch_time + (3.7p) * tscale)) begin resist =
    1214.66604403 + scalep ; end //1105020
  @(timer(switch_time + (4.7p) * tscale)) begin resist =
    1478.81193891 + scalep ; end //1105020
  @(timer(switch_time + (6.2p) * tscale)) begin resist =
    1256.07435201 + scalep ; end //1105020
  @(timer(switch_time + (7.7p) * tscale)) begin resist =
    1426.61408261 + scalep ; end //1105020
  @(timer(switch_time + (8.8p) * tscale)) begin resist =
    1302.51137336 + scalep ; end //1105020
// ... remaining resistance profile within a cycle is omitted
end

@(cross(V(clk) - 0.2, 1)) begin
  switch_time = $abstime;
  logic_one = 1;
end
@(cross( 0.6-V(clk), 1)) begin
  switch_time = $abstime;
  logic_one = 0;
end

// replay current profile
I(vdd_pin, vss_pin) <+ V(vdd_pin, vss_pin) / resist;

// enable signal transition if supply voltage is larger than 0.4 (
// threshold)
V(clk_out) <+ transition(V(vdd_pin, vss_pin)>0.4?V(clk):0, tbin);
end

```

## Verilog-A source code of operation mode sub-module

```

`include "constants.vams"
`include "disciplines.vams"
module op_mod(clk,rst,vdd_pin, vss_pin) ;
  input clk, rst;
  inout vdd_pin, vss_pin;
  electrical clk, rst, vdd_pin, vss_pin;
  real resist = 0;
  real scalep = 0;
  real tscale = 1;
  integer cycle = 0;
  real switch_time = 0;

analog begin
  // initial status
  @(initial_step) begin
    resist = 4000;
    cycle = 0;
    tscale = 1;
    switch_time = 0;
    logic_one = -100;
  end

  // resistance scaling factor
  if (V(vdd_pin, vss_pin) >= 0.8) begin
    scalep = (V(vdd_pin, vss_pin)-0.8)*(-0.63);
  end
  else begin
    scalep = (V(vdd_pin, vss_pin)-0.8)*(-0.87);
  end

  // latency scaling factor
  if (V(clk, vss_pin) >= 0.1) begin
    tscale = ((64.4 * V(vdd_pin, vss_pin)- 5.711) / (V(vdd_pin,
      vss_pin) - 0.4016) ) / ((64.4 * 0.8 - 5.711) / (0.8 - 0.4016)
      );
  end
  else begin
    tscale = ((63.94 * V(vdd_pin, vss_pin)- 4.814) / (V(vdd_pin,
      vss_pin) - 0.3988) ) / ((63.94 * 0.8 - 4.814) / (0.8 -

```

```
 0.3988) );
end

// reset rise
@(cross(V(rst) - 0.2, 1)) begin
  cycle = -100;
  switch_time = $abstime;
end
// reset fall
@(cross(V(rst) - 0.6, -1)) begin
  cycle = 0;
  switch_time = $abstime;
end

// resistance profile of each clock cycle (rising edge),
translated from current profile
if (cycle == 0) begin
  @(timer(switch_time + (0.1p) * tscale)) begin resist =
    59.6702057775 + scalep ; end //0
  @(timer(switch_time + (0.7p) * tscale)) begin resist =
    11.9846946557 + scalep ; end //0
  @(timer(switch_time + (2.0p) * tscale)) begin resist =
    6.41430806626 + scalep ; end //0
  @(timer(switch_time + (3.3p) * tscale)) begin resist =
    10.3012023861 + scalep ; end //0
  @(timer(switch_time + (4.4p) * tscale)) begin resist =
    17.1452744435 + scalep ; end //0
  @(timer(switch_time + (9.0p) * tscale)) begin resist =
    87.7546118816 + scalep ; end //0
  @(timer(switch_time + (10.2p) * tscale)) begin resist =
    116.631365235 + scalep ; end //0
// ... remaining resistance profile within a cycle is omitted
end

// resistance profile of each clock cycle (falling edge),
translated from current profile
if (cycle == 1) begin
  @(timer(switch_time + (1.7p) * tscale)) begin resist =
    1425.87573463 + scalep ; end //42520
  @(timer(switch_time + (3.1p) * tscale)) begin resist =
    1469.2492612 + scalep ; end //42520
```

```
@(timer(switch_time + (4.4p) * tscale)) begin resist =
  1127.04255275 + scalep ; end //42520
@(timer(switch_time + (5.7p) * tscale)) begin resist =
  1176.85624995 + scalep ; end //42520
// ... remaining resistance profile within a cycle is omitted
end

// ... remaining resistance profile of other cycles is omitted

// disable status
if (cycle <0) begin
  @(timer(switch_time + 0.0p)) begin resist = 4000; end //reset
  resistance
end

// update clock index
@(cross(V(clk) - 0.2, 1)) begin
  if (cycle>=0) begin
    cycle = cycle+1;
    switch_time = $abstime;
  end
end

@(cross(0.6-V(clk), 1)) begin
  if (cycle>=0) begin
    cycle = cycle+1;
    switch_time = $abstime;
  end
end

// replay current profile
I(vdd_pin, vss_pin) <+ V(vdd_pin, vss_pin) / resist;
end
endmodule
```



# Bibliography

- [1] CPU db. [Online]. Available: [cpudb.stanford.edu](http://cpudb.stanford.edu)
- [2] Standard performance evaluation corporation. [Online]. Available: [www.spec.org](http://www.spec.org)
- [3] P. A. Gargini, “How to successfully overcome inflection points, or long live Moore’s law,” *Computing in Science & Engineering*, vol. 19, no. 2, pp. 51–62, Mar.-Apr. 2017.
- [4] L. Xiu, “Time Moore: Exploiting Moore’s Law From The Perspective of Time,” *IEEE Solid-State Circuits Magazine*, vol. 11, no. 1, pp. 39–55, winter 2019.
- [5] M. Horowitz, “Computing’s energy problem (and what we can do about it),” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb. 2014, pp. 10–14.
- [6] W. M. Holt, “Moore’s law: A path going forward,” in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 8–13.
- [7] P. Gargini, “Roadmap evolution: From NTRS to ITRS, from ITRS 2.0 to IRDS,” in *2017 Fifth Berkeley Symposium on Energy Efficient Electronic Systems & Steep Transistors Workshop (E3S)*, Oct. 2017, pp. 1–62.
- [8] International technology roadmap for semiconductors 2.0 2015 edition executive report. [Online]. Available: [www.itrs2.net/itrs-reports.html](http://www.itrs2.net/itrs-reports.html)
- [9] International roadmap for devices and systems 2018 edition executive summary. [Online]. Available: <https://irds.ieee.org/editions/2018/executive-summary>
- [10] L. Kish, “Moore’s law and the energy requirement of computing versus performance,” *IEE Proceedings - Circuits, Devices and Systems*, vol. 151, no. 2, pp. 190–194, 12 April 2004.
- [11] T. M. Conte, “Rebooting Computing: The Road Ahead,” *Computer*, vol. 50, no. 1, pp. 20–29, Jan. 2017.

- [12] S. Kosonocky, T. Burd, K. Kasprak, R. Schultz, and R. Stephany, “Designing in scaled technologies: 32 nm and beyond,” in *2012 Symposium on VLSI Technology (VLSIT)*, June 2012, pp. 147–148.
- [13] K. Arabi, “Low power design techniques in mobile processes,” in *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Aug. 2014, pp. 1–1.
- [14] M. Badaroglu, K. Ng, M. Salmani, S. Kim, G. Klimeck, C.-P. Chang, C. Cheung, and Y. Fukuzaki, “More Moore landscape for system readiness - ITRS2.0 requirements,” in *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, Oct. 2014, pp. 147–152.
- [15] N. Ahmed, M. Tehranipoor, and V. Jayaram, “Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design,” in *2007 44th ACM/IEEE Design Automation Conference*, June 2007, pp. 533–538.
- [16] X. Wang, “A Novel Peak Power Supply Noise Measurement and Adaptation System for Integrated Circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 5, pp. 1715–1727, May 2016.
- [17] P. N. Whatmough, “Power Integrity Analysis of a 28 nm Dual-Core ARM Cortex-A57 Cluster Using an All-Digital Power Delivery Monitor,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 6, pp. 1643–1654, June 2017.
- [18] M. Saint-Laurent, “Impact of power-supply noise on timing in high-frequency microprocessors,” *IEEE Transactions on Advanced Packaging*, vol. 27, no. 1, pp. 135–144, Feb. 2004.
- [19] V. J. Reddi, S. Kanev, W. Kim, S. Campanoni, M. D. Smith, G.-Y. Wei, and D. Brooks, “Voltage Smoothing: Characterizing and Mitigating Voltage Noise in Production Processors via Software-Guided Thread Scheduling,” in *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, Dec. 2010, pp. 77–88.
- [20] S. Bhowmik, “Power Supply Noise Reduction of Multicore CPU by Staggering Current and Variable Clock Frequency,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 5, pp. 875–882, May 2018.
- [21] D. R. E. Gnad, “An Experimental Evaluation and Analysis of Transient Voltage Fluctuations in FPGAs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 10, pp. 1817–1830, Oct. 2018.
- [22] M. Swaminathan, “Power distribution networks for system-on-package: status and challenges,” *IEEE Transactions on Advanced Packaging*, vol. 27, no. 2, pp. 286–300, May 2004.

- [23] A. J. Rainal, "Computing inductive noise of chip packages," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 1, pp. 177–195, Jan. 1984.
- [24] G. Katopis, "Delta-I noise specification for a high-performance computing machine," *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1405–1415, Sept. 1985.
- [25] R. Senthinathan and J. Prince, "Simultaneous switching ground noise calculation for packaged CMOS devices," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 11, pp. 1724–1728, Nov. 1991.
- [26] A. Vaidyanath, B. Thoroddson, and J. Prince, "Effect of CMOS driver loading conditions on simultaneous switching noise," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, vol. 17, no. 4, pp. 480–485, Nov. 1994.
- [27] A. Kabbani and A. Al-Khalili, "Estimation of ground bounce effects on CMOS circuits," *IEEE Transactions on Components and Packaging Technologies*, vol. 22, no. 2, pp. 316–325, June 1999.
- [28] E. Davidson, "Delay factors for mainframe computers," in *Proceedings of the 1991 Bipolar Circuits and Technology Meeting*, 1991, pp. 116–123.
- [29] L.-R. Zheng, "Fast modeling of core switching noise on distributed LRC power grid in ULSI circuits," *IEEE Transactions on Advanced Packaging*, vol. 24, no. 3, pp. 245–254, Aug. 2001.
- [30] H. Chen and J. Neely, "Interconnect and circuit modeling techniques for full-chip power supply noise analysis," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, vol. 21, no. 3, pp. 209–215, Aug. 1998.
- [31] M. Zhao, "Hierarchical analysis of power distribution networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 2, pp. 159–168, Feb. 2002.
- [32] Y. Eo, W. Eisenstadt, J. Y. Jeong, and O.-K. Kwon, "New simultaneous switching noise analysis and modeling for high-speed and high-density CMOS IC package design," *IEEE Transactions on Advanced Packaging*, vol. 23, no. 2, pp. 303–312, May 2000.
- [33] K. Tang, "Simultaneous switching noise in on-chip CMOS power distribution networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 4, pp. 487–493, Aug. 2002.
- [34] B. Garben, "Frequency dependencies of power noise," *IEEE Transactions on Advanced Packaging*, vol. 25, no. 2, pp. 166–173, May 2002.

- [35] Y. Zhou, P. M. Harvey, B. Flachs, J. Liberty, G. Gervais, R. Mandrekar, H. H. Chen, and T. Tamura, “Distributed On-chip Power Supply Noise Characterization of the Cell Broadband Engine,” in *2007 IEEE Electrical Performance of Electronic Packaging*, Oct. 2007, pp. 99–102.
- [36] W. Ahmad, L.-R. Zheng, R. Weerasekera, Q. Chen, A. Y. Weldezion, and H. Tenhunen, “Power integrity optimization of 3D chips stacked through TSVs,” in *2009 IEEE 18th Conference on Electrical Performance of Electronic Packaging and Systems*, Oct. 2009, pp. 105–108.
- [37] R. Downing, P. Gebler, and G. Katopis, “Decoupling capacitor effects on switching noise,” *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 16, no. 5, pp. 484–489, Aug. 1993.
- [38] L. Smith, “Decoupling capacitor calculations for CMOS circuits,” in *Proceedings of 1994 IEEE Electrical Performance of Electronic Packaging*, 1994, pp. 101–105.
- [39] L. Smith, “Packaging and power distribution design considerations for a Sun Microsystems desktop workstation,” in *Electrical Performance of Electronic Packaging*, 1997, pp. 19–22.
- [40] L. Smith, R. Anderson, D. Forehand, T. Pelc, and T. Roy, “Power distribution system design methodology and capacitor selection for modern CMOS technology,” *IEEE Transactions on Advanced Packaging*, vol. 22, no. 3, pp. 284–291, Aug. 1999.
- [41] T. Rahal-Arabi, G. Taylor, M. Ma, and C. Webb, “Design and validation of the Pentium III and Pentium 4 processors power delivery,” in *2002 Symposium on VLSI Circuits*, 2002, pp. 220–223.
- [42] T. Rahal-Arabi, G. Taylor, M. Ma, J. Jones, and C. Webb, “Design and validation of the core and IOs decoupling of the Pentium III and Pentium 4 processors,” in *2002 IEEE 11th Topical Meeting on Electrical Performance of Electronic Packaging*, 2002, pp. 249–252.
- [43] H. Chen, J. Neely, M. Wang, and G. Co, “On-chip decoupling capacitor optimization for noise and leakage reduction,” in *16th Symposium on Integrated Circuits and Systems Design*, 2003, pp. 251–255.
- [44] M. Hashimoto, J. Yamaguchi, T. Sato, and H. Onodera, “Timing analysis considering temporal supply voltage fluctuation,” in *Proceedings of the ASP-DAC 2005*, 2005, pp. 1098–1101 Vol. 2.

- [45] Y. Ogasahara, T. Enami, M. Hashimoto, T. Sato, and T. Onoye, “Measurement results of delay degradation due to power supply noise well correlated with full-chip simulation,” in *IEEE Custom Integrated Circuits Conference 2006*, Sept. 2006, pp. 861–864.
- [46] F. Azais, Y. Bertrand, and M. Renovell, “An analysis of the timing behavior of CMOS digital blocks under Simultaneous Switching Noise conditions,” in *2009 12th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, April 2009, pp. 158–163.
- [47] L. G. Salem, “A Recursive Switched-Capacitor DC-DC Converter Achieving  $2N - 1$  Ratios With High Efficiency Over a Wide Output Voltage Range,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2773–2787, Dec. 2014.
- [48] B. Keller, “A RISC-V Processor SoC With Integrated Power Management at Sub-microsecond Timescales in 28 nm FD-SOI,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 7, pp. 1863–1875, July 2017.
- [49] T. V. Breussegem and M. Steyaert, “A 82% efficiency 0.5% ripple 16-phase fully integrated capacitive voltage doubler,” in *2009 Symposium on VLSI Circuits*, June 2009, pp. 198–199.
- [50] T. Souvignet, “A Fully Integrated Switched-Capacitor Regulator With Frequency Modulation Control in 28-nm FDSOI,” *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 4984–4994, July 2016.
- [51] T. M. Andersen, “A 10 W On-Chip Switched Capacitor Voltage Regulator With Feedforward Regulation Capability for Granular Microprocessor Power Delivery,” *IEEE Transactions on Power Electronics*, vol. 32, no. 1, pp. 378–393, Jan. 2017.
- [52] Y. Lu, “A Multiphase Switched-Capacitor DC-DC Converter Ring With Fast Transient Response and Small Ripple,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 579–591, Feb. 2017.
- [53] W.-C. Hsieh and W. Hwang, “In-situ self-aware adaptive power control system with multi-mode power gating network,” in *2008 IEEE International SOC Conference*, Sept. 2008, pp. 215–218.
- [54] M. Saint-Laurent, P. Bassett, K. Lin, Y. Wang, S. Le, X. Chen, M. Alradaideh, T. Wernimont, K. Ayyar, D. Bui, D. Galbi, A. Lester, and W. Anderson, “A 28nm DSP powered by an on-chip LDO for high-performance and energy-efficient mobile applications,” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb. 2014, pp. 176–177.

- [55] G. Rincon-Mora and P. Allen, “A low-voltage, low quiescent current, low drop-out regulator,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, pp. 36–44, Jan. 1998.
- [56] W. Chen, W.-H. Ki, P. Mok, and M. Chan, “Switched-capacitor power converters with integrated low dropout regulators,” *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 293–296 vol. 2, 2001.
- [57] S. Mondal and R. Paily, “An Efficient On-Chip Switched-Capacitor-Based Power Converter for a Microscale Energy Transducer,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 3, pp. 254–258, March 2016.
- [58] A. Paul, M. Amrein, S. Gupta, A. Vinod, A. Arun, S. Sapatnekar, and C. H. Kim, “Staggered Core Activation: A circuit/architectural approach for mitigating resonant supply noise issues in multi-core multi-power domain processors,” in *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*, Sept. 2012, pp. 1–4.
- [59] Y. Cheng, A. Todri-Sanial, A. Bosio, L. Dilillo, P. Girard, and A. Virazel, “Power supply noise-aware workload assignments for homogeneous 3D MPSoCs with thermal consideration,” in *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2014, pp. 544–549.
- [60] B. Kim, “A Supply-Noise Sensitivity Tracking PLL in 32 nm SOI Featuring a Deep Trench Capacitor Based Loop Filter,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 1017–1026, April 2014.
- [61] X. Fan, “Frequency-Domain Optimization of Digital Switching Noise Based on Clock Scheduling,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 7, pp. 982–993, July 2016.
- [62] Y.-C. Liu, C.-Y. Han, S.-Y. Lin, and J. C.-M. Li, “PSN-aware circuit test timing prediction using machine learning,” *IET Computers & Digital Techniques*, vol. 11, no. 2, pp. 60–67, 3 2017.
- [63] X. Liu, “Machine Learning for Noise Sensor Placement and Full-Chip Voltage Emergency Detection,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 3, pp. 421–434, March 2017.
- [64] S.-Y. Lin, Y.-C. Fang, Y.-C. Li, Y.-C. Liu, T.-S. Yang, S.-C. Lin, C.-M. Li, and E. J.-W. Fang, “IR drop prediction of ECO-revised circuits using machine learning,” in *2018 IEEE 36th VLSI Test Symposium (VTS)*, April 2018, pp. 1–6.
- [65] C. Ababei, “A Survey of Prediction and Classification Techniques in Multicore Processor Systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 5, pp. 1184–1200, 1 May 2019.

- [66] S. Moon, S. Nam, J. Son, and S. L. S. Lee, “An Approach for PDN Simplification of a Mobile Processor,” in *2018 IEEE 68th Electronic Components and Technology Conference (ECTC)*, May 2018, pp. 1706–1711.
- [67] J. Gu, “On-Chip Supply Noise Regulation Using a Low-Power Digital Switched Decoupling Capacitor Circuit,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 6, pp. 1765–1775, June 2009.
- [68] L. Wang, L. Wang, D. Shang, C. Zhuo, and P. Zhou, “Optimizing the Energy Efficiency of Power Supply in Heterogeneous Multicore Chips with Integrated Switched-Capacitor Converters,” in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2019, pp. 836–841.
- [69] T.-L. Wu, “Overview of Power Integrity Solutions on Package and PCB: Decoupling and EBG Isolation,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 346–356, May 2010.
- [70] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, “A dynamic voltage scaled microprocessor system,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov. 2000.
- [71] V. V. Kaenel, P. Macken, and M. Degrauwe, “A voltage reduction technique for battery-operated systems,” *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1136–1140, Oct. 1990.
- [72] P. Macken, M. Degrauwe, M. V. Paemel, and H. Oguey, “A voltage reduction technique for digital systems,” in *1990 37th IEEE International Conference on Solid-State Circuits*, 1990, pp. 238–239.
- [73] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, “System level analysis of fast, per-core DVFS using on-chip switching regulators,” in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, Feb. 2008, pp. 123–134.
- [74] R. Redl, “Ripple-Based Control of Switching Regulators-An Overview,” *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 2669–2680, Dec. 2009.
- [75] V. Kursun, S. Narendra, V. De, and E. Friedman, “Analysis of buck converters for on-chip integration with a dual supply voltage microprocessor,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 3, pp. 514–522, June 2003.
- [76] V. Kursun, S. Narendra, V. De, and E. Friedman, “Low-voltage-swing monolithic dc-dc conversion,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 5, pp. 241–248, May 2004.

- [77] J. Dickson, “On-chip high-voltage generation in MNOS integrated circuits using an improved voltage multiplier technique,” *IEEE Journal of Solid-State Circuits*, vol. 11, no. 3, pp. 374–378, June 1976.
- [78] B. Nguyen, “High-Efficiency Fully Integrated Switched-Capacitor Voltage Regulator for Battery-Connected Applications in Low-Breakdown Process Technologies,” *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 6858–6868, Aug. 2018.
- [79] S. T. Kim, Y.-C. Shih, K. Mazumdar, R. Jain, J. F. Ryan, C. Tokunaga, C. Augustine, J. P. Kulkarni, K. Ravichandran, J. W. Tschanz, M. M. Khellah, and V. De, “Enabling wide autonomous DVFS in a 22nm graphics execution core using a digitally controlled hybrid LDO/switched-capacitor VR with fast droop mitigation,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb. 2015, pp. 1–3.
- [80] F. U. Ahmed and M. H. Chowdhury, “An Asynchronous Reconfigurable Switched Capacitor Voltage Regulator,” in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2018, pp. 1110–1113.
- [81] X. Zhan, “Power Management for Multicore Processors via Heterogeneous Voltage Regulation and Machine Learning Enabled Adaptation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 11, pp. 2641–2654, Nov. 2019.
- [82] M. Hashimoto and R. Nair, *Power Integrity for Nanoscale Integrated Systems*. McGraw-Hill Education, 2014.
- [83] E. McGibney, “An overview of electrical characterization techniques and theory for IC packages and interconnects,” *IEEE Transactions on Advanced Packaging*, vol. 29, no. 1, pp. 131–139, Feb. 2006.
- [84] J. Andrews and S. Kabir, “Package model extraction from multi-port S-parameters,” in *IEEE 10th Topical Meeting on Electrical Performance of Electronic Packaging*, 2001, pp. 309–312.
- [85] Z. Chen, “A general co-design approach to multi-level package modeling based on individual single-level package full-wave S-parameter modeling including signal and power/ground ports,” in *2012 IEEE 62nd Electronic Components and Technology Conference*, May 2012, pp. 1687–1694.
- [86] J. Choi, “Modeling and analysis of power distribution networks for Gigabit applications,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 299–313, Oct.-Dec. 2003.

- [87] A. Bouchaala, "W-element RLGC matrices calculation for power distribution planes modeling using MCTL matrix method," *IEEE Electromagnetic Compatibility Magazine*, vol. 5, no. 3, pp. 61–69, Third Quarter 2016.
- [88] R. Neumayer, F. Haslinger, A. Stelzer, and R. Weigel, "Synthesis of SPICE-compatible broadband electrical models from n-port scattering parameter data," *2002 IEEE International Symposium on Electromagnetic Compatibility*, vol. 1, pp. 469–474 vol.1, 2002.
- [89] W.-C. Lee and T.-H. Chu, "Modeling of a Planar Metamaterial Power Divider/Combiner Using Transmission Matrix Method," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 4, pp. 205–207, April 2015.
- [90] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, July 1999.
- [91] K. Kang, W.-Y. Yin, and L.-W. Li, "Transfer functions of on-chip global interconnects based on distributed RLCG interconnects model," *2005 IEEE Antennas and Propagation Society International Symposium*, vol. 1A, pp. 524–527 Vol. 1A, 2005.
- [92] W. Cui, P. Parmar, J. Morgan, and U. Sheth, "Modeling the network processor and package for power delivery analysis," *2005 International Symposium on Electromagnetic Compatibility, 2005. EMC 2005.*, vol. 3, pp. 690–694 Vol. 3, 2005.
- [93] L. Zheng, "Full-Chip Power Supply Noise Time-Domain Numerical Modeling and Analysis for Single and Stacked ICs," *IEEE Transactions on Electron Devices*, vol. 63, no. 3, pp. 1225–1231, March 2016.
- [94] Y. Ogasahara, T. Enami, M. Hashimoto, T. Sato, and T. Onoye, "Validation of a Full-Chip Simulation Model for Supply Noise and Delay Dependence on Average Voltage Drop With On-Chip Delay Measurement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 10, pp. 868–872, Oct. 2007.
- [95] S. Lin and N. Chang, "Challenges in power-ground integrity," in *IEEE/ACM International Conference on Computer Aided Design 2001*, 2001, pp. 651–654.
- [96] Cadence, *Virtuoso Analog Design Environment User Guide*, 2016.
- [97] Synopsys, *HSPICE User Guide*, 2012.
- [98] B. P. Schweitzer and A. B. Rosenstein, "Free Running-Switching Mode Power Regulator: Analysis and Design," *IEEE Transactions on Aerospace*, vol. 2, no. 4, pp. 1171–1180, Oct. 1964.

- [99] S. Michael, “A design methodology for switched-capacitor dc-dc converters,” Ph.D. dissertation, UC Berkeley, May 2009.
- [100] B. Zimmer, “A RISC-V Vector Processor With Simultaneous-Switching Switched-Capacitor DC-DC Converters in 28 nm FDSOI,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 930–942, April 2016.
- [101] S. Bang, “A Low Ripple Switched-Capacitor Voltage Regulator Using Flying Capacitance Dithering,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 919–929, April 2016.
- [102] K. Arabi, “Power Supply Noise in SoCs: Metrics, Management, and Measurement,” *IEEE Design & Test of Computers*, vol. 24, no. 3, pp. 236–244, May-June 2007.
- [103] M. Popovich, “On-Chip Power Distribution Grids With Multiple Supply Voltages for High-Performance Integrated Circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 7, pp. 908–921, July 2008.
- [104] R. Thomas, K. Barber, N. Sedaghati, L. Zhou, and R. Teodorescu, “Core tunneling: Variation-aware voltage noise mitigation in GPUs,” in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 151–162.
- [105] P. I.-J. Chuang, C. Vezrytzis, D. Pathak, R. Rizzolo, T. Webel, T. Strach, O. Torreiter, P. Lobo, A. Buyuktosunoglu, R. Bertran, M. Floyd, M. Ware, G. Salem, S. Carey, and P. Restle, “Power supply noise in a 22nm z13 microprocessor,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2017, pp. 438–439.
- [106] S. Das, P. Whatmough, and D. Bull, “Modeling and characterization of the system-level Power Delivery Network for a dual-core ARM Cortex-A57 cluster in 28nm CMOS,” in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, July 2015, pp. 146–151.
- [107] A. Todri, M. Marek-Sadowska, and J. Kozhaya, “Power supply noise aware workload assignment for multi-core systems,” in *2008 IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2008, pp. 330–337.
- [108] N. James, P. Restle, J. Friedrich, B. Huott, and B. McCredie, “Comparison of Split-Versus Connected-Core Supplies in the POWER6 Microprocessor,” in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 298–604.

- [109] A. Waizman and C.-Y. Chung, “Package capacitors impact on microprocessor maximum operating frequency,” in *2001 Proceedings. 51st Electronic Components and Technology Conference*, 2001, pp. 118–122.
- [110] S. Borkar, “Low power design challenges for the decade,” in *Proceedings of the ASP-DAC 2001*, 2001, pp. 293–296.
- [111] P. Muthana, “Design, Modeling, and Characterization of Embedded Capacitor Networks for Core Decoupling in the Package,” *IEEE Transactions on Advanced Packaging*, vol. 30, no. 4, pp. 809–822, Nov. 2007.
- [112] Z. Qi, H. Li, S.-D. Tan, L. Wu, Y. Cai, and X. Hong, “Fast decap allocation algorithm for robust on-chip power delivery,” in *Sixth international symposium on quality electronic design (isqed'05)*, 2005, pp. 542–547.
- [113] M. Popovich, “Effective Radii of On-Chip Decoupling Capacitors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 7, pp. 894–907, July 2008.
- [114] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “Razor: a low-power pipeline based on circuit-level timing speculation,” in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture*, 2003, pp. 7–18.
- [115] S. Das, “A self-tuning DVS processor using delay-error detection and correction,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, April 2006.
- [116] M. Eireiner, “In-Situ Delay Characterization and Local Supply Voltage Adjustment for Compensation of Local Parametric Variations,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 7, pp. 1583–1592, July 2007.
- [117] P. N. Whatmough, “Circuit-Level Timing Error Tolerance for Low-Power DSP Filters and Transforms,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 6, pp. 989–999, June 2013.
- [118] S. Das, “RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.
- [119] K. Nowka, “A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1441–1447, Nov. 2002.
- [120] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, “A Distributed Critical-Path Timing Monitor for a 65nm

High-Performance Microprocessor,” in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 398–399.

[121] W.-C. Lam, C.-K. Koh, and C.-W. Tsao, “Power supply noise suppression via clock skew scheduling,” in *Proceedings International Symposium on Quality Electronic Design*, 2002, pp. 355–360.

[122] Y. Kaplan, “Mixing Drivers in Clock-Tree for Power Supply Noise Reduction,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 5, pp. 1382–1391, May 2015.

[123] M. Ang, R. Salem, and A. Taylor, “An on-chip voltage regulator using switched decoupling capacitors,” in *2000 IEEE International Solid-State Circuits Conference*, 2000, pp. 438–439.

[124] T. Tsukada, “An on-chip active decoupling circuit to suppress crosstalk in deep-submicron CMOS mixed-signal SoCs,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 67–79, Jan. 2005.

[125] J. Gu, “Design and Implementation of Active Decoupling Capacitor Circuits for Power Supply Regulation in Digital ICs,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 2, pp. 292–301, Feb. 2009.

[126] L. Ren, “Prediction of Power Supply Noise From Switching Activity in an FPGA,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, no. 3, pp. 699–706, June 2014.

[127] F. Ye, “On-Chip Droop-Induced Circuit Delay Prediction Based on Support-Vector Machines,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 4, pp. 665–678, April 2016.

[128] M. Kaliorakis, A. Chatzidimitriou, G. Papadimitriou, and D. Gizopoulos, “Statistical Analysis of Multicore CPUs Operation in Scaled Voltage Conditions,” *IEEE Computer Architecture Letters*, vol. 17, no. 2, pp. 109–112, 1 July-Dec. 2018.

[129] V. J. Reddi, M. S. Gupta, G. Holloway, G.-Y. Wei, M. D. Smith, and D. Brooks, “Voltage emergency prediction: Using signatures to reduce operating margins,” in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, Feb. 2009, pp. 18–29.

[130] M. D. Mulligan, B. Broach, and T. H. Lee, “A 3MHz Low-Voltage Buck Converter with Improved Light Load Efficiency,” in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 528–620.

- [131] J.-H. Lin, Y.-S. Ma, C.-M. Huang, L.-C. Lin, C.-H. Cheng, K.-H. Chen, Y.-H. Lin, S.-R. Lin, and T.-Y. Tsai, “A high-efficiency and fast-transient digital-low-dropout regulator with the burst mode corresponding to the power-saving modes of DC-DC switching converters,” in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 314–316.
- [132] S. Wang, “Light-Weight On-Chip Structure for Measuring Timing Uncertainty Induced by Noise in Integrated Circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1030–1041, May 2014.
- [133] M. Makowski and D. Maksimovic, “Performance limits of switched-capacitor DC-DC converters,” *Proceedings of PESC '95 - Power Electronics Specialist Conference*, vol. 2, pp. 1215–1221 vol.2, 1995.
- [134] S. R. Sanders, E. Alon, H.-P. Le, M. D. Seeman, M. John, and V. W. Ng, “The Road to Fully Integrated DC-DC Conversion via the Switched-Capacitor Approach,” *IEEE Transactions on Power Electronics*, vol. 28, no. 9, pp. 4146–4155, Sept. 2013.
- [135] H. Li, “Energy-Efficient Power Delivery System Paradigms for Many-Core Processors,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 3, pp. 449–462, March 2017.
- [136] R. Jevtic, “Per-Core DVFS With Switched-Capacitor Converters for Energy Efficiency in Manycore Processors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 4, pp. 723–730, April 2015.
- [137] G. Patounakis, “A fully integrated on-chip DC-DC conversion and power management system,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 443–451, March 2004.
- [138] Y. Lu, “A Reconfigurable Switched-Capacitor DC-DC Converter and Cascode LDO for Dynamic Voltage Scaling and High PSR,” in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Oct. 2018, pp. 509–511.
- [139] G. Pillonnet, “Dual-Input Switched Capacitor Converter Suitable for Wide Voltage Gain Range,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 5, no. 3, pp. 413–420, Sept. 2015.
- [140] J. Jiang, “Digital 2-/3-Phase Switched-Capacitor Converter With Ripple Reduction and Efficiency Improvement,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 7, pp. 1836–1848, July 2017.
- [141] D. El-Damak, S. Bandyopadhyay, and A. P. Chandrakasan, “A 93% efficiency reconfigurable switched-capacitor DC-DC converter using on-chip ferroelectric

capacitors,” in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 2013, pp. 374–375.

[142] R. Madeira, J. P. Oliveira, and N. Paulino, “A 130 nm CMOS Power Management Unit With a Multi-Ratio Core SC DC-DC Converter for a Super capacitor Power Supply,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 10, pp. 1445–1449, Oct. 2018.

[143] Y.-T. Lin, “A Fully Integrated Asymmetrical Shunt Switched-Capacitor DC-DC Converter With Fast Optimum Ratio Searching Scheme for Load Transient Enhancement,” *IEEE Transactions on Power Electronics*, vol. 34, no. 9, pp. 9146–9157, Sept. 2019.

[144] A. Urso and W. A. Serdijn, “A Switched Capacitor DC-DC Buck Converter for a Wide Input Voltage Range,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.

[145] J. Kim, S. Wu, H. Wang, Y. Takita, H. Takeuchi, K. Araki, G. Feng, and J. Fan, “Improved target impedance and IC transient current measurement for power distribution network design,” in *2010 IEEE International Symposium on Electromagnetic Compatibility*, July 2010, pp. 445–450.

[146] M.-S. Zhang, “New Power Distribution Network Design Method for Digital Systems Using Time-Domain Transient Impedance,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, no. 8, pp. 1399–1408, Aug. 2013.

[147] J. Kim, “Improved Target Impedance for Power Distribution Network Design With Power Traces Based on Rigorous Transient Analysis in a Handheld Device,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, no. 9, pp. 1554–1563, Sept. 2013.

[148] G. Chen and D. Oh, “Improving the target impedance method for PCB decoupling of core power,” in *2014 IEEE 64th Electronic Components and Technology Conference (ECTC)*, May 2014, pp. 566–571.

[149] J. Kim, “Closed-Form Expressions for the Noise Voltage Caused by a Burst Train of IC Switching Currents on a Power Distribution Network,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, no. 6, pp. 1585–1597, Dec. 2014.

[150] D. Oh and G. Chen, “Challenges and solutions for core power distribution network designs,” *IEEE Electromagnetic Compatibility Magazine*, vol. 5, no. 4, pp. 104–111, Fourth Quarter 2016.

- [151] D. Oh and Y. Shim, "Power integrity analysis for core timing models," in *2014 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, Aug. 2014, pp. 833–838.
- [152] Y. Kim, K. Kim, J. Cho, J. Kim, K. Kang, T. Yang, Y. Ra, and W. Paik, "Power distribution network design and optimization based on frequency dependent target impedance," in *2015 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, Dec. 2015, pp. 89–92.
- [153] K. Koo, "Fast Algorithm for Minimizing the Number of decap in Power Distribution Networks," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 3, pp. 725–732, June 2018.
- [154] Y. Shim, "System Level Modeling of Timing Margin Loss Due to Dynamic Supply Noise for High-Speed Clock Forwarding Interface," *IEEE Transactions on Electromagnetic Compatibility*, vol. 58, no. 4, pp. 1349–1358, Aug. 2016.
- [155] G. Bai, S. Bobba, and I. Hjj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *Proceedings of the 38th Design Automation Conference*, 2001, pp. 295–300.
- [156] R. Bertran, A. Buyuktosunoglu, P. Bose, T. J. Slegel, G. Salem, S. Carey, R. F. Rizzolo, and T. Strach, "Voltage Noise in Multi-Core Processors: Empirical Characterization and Optimization Opportunities," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec. 2014, pp. 368–380.
- [157] J. Kim, "Delay Monitoring System With Multiple Generic Monitors for Wide Voltage Range Operation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 37–49, Jan. 2018.
- [158] Q. Liu, "Capturing Post-Silicon Variations Using a Representative Critical Path," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 2, pp. 211–222, Feb. 2010.
- [159] K. A. Bowman, "A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, Jan. 2011.
- [160] D. Fick, N. Liu, Z. Foo, M. Fojtik, J. sun Seo, D. Sylvester, and D. Blaauw, "In situ delay-slack monitor for high-performance processors using an all-digital self-calibrating 5ps resolution time-to-digital converter," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb. 2010, pp. 188–189.

- [161] K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. A. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, and A. Waterman, “The rocket chip generator,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, Apr 2016. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html>
- [162] C. R. Lefurgy, “Active Guardband Management in Power7+ to Save Energy and Maintain Reliability,” *IEEE Micro*, vol. 33, no. 4, pp. 35–45, July-Aug. 2013.
- [163] NanGate FreePDK 45nm Cell Library. [Online]. Available: [https://www.silvaco.com/products/nangate/FreePDK45\\_Open\\_Cell\\_Library/index.html](https://www.silvaco.com/products/nangate/FreePDK45_Open_Cell_Library/index.html)
- [164] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [165] F. M. Sleiman and T. F. Wenisch, “Efficiently Scaling Out-of-Order Cores for Simultaneous Multithreading,” in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 431–443.
- [166] O. Mutlu, H. Kim, D. Armstrong, and Y. Patt, “An analysis of the performance impact of wrong-path memory references on out-of-order and runahead execution processors,” *IEEE Transactions on Computers*, vol. 54, no. 12, pp. 1556–1571, Dec. 2005.
- [167] OpenRISC. [Online]. Available: <https://openrisc.io>
- [168] NanGate FreePDK 15nm Cell Library. [Online]. Available: <https://www.eda.ncsu.edu/wiki/FreePDK15:Contents>