UNIVERSITY OF CALIFORNIA

Santa Barbara

Assessing VeSFET Monolithic 3D Technology in Physical Design, Dynamic Reconfigurable Computing, and Hardware Security

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical and Computer Engineering

by

Ping-Lin Yang

Committee in charge:

Professor Malgorzata Marek-Sadowska, Chair

Professor Kwang-Ting Tim Cheng

Professor Wojciech P. Maly

Professor Li-C. Wang

January 2018

The dissertation of Ping-Lin Yang is approved.

_____

Kwang-Ting Tim Cheng

_____

Wojciech P. Maly

_____

Li-C. Wang

_____

Malgorzata Marek-Sadowska, Committee Chair

December 2017

Assessing VeSFET Monolithic 3D Technology in Physical Design,

Dynamic Reconfigurable Computing, and Hardware Security

by

Ping-Lin Yang

ACKNOWLEDGEMENTS

This dissertation and all the research works I have finished during my Ph.D. journey would not have been possible without my advisor and mentor Professor Malgorzata Marek-Sadowska. I would like to express my deepest gratitude to her. Over hundreds of discussions, she guided me, enlightened me, inspired me, was always open-minded, and gave me precise and constructive suggestions at all times. Her wisdom, knowledge, philosophy, diligence, patience, and attitudes toward work and life have greatly influenced me. I feel very lucky and thankful to have such the great advisor and mentor in one of the most important chapter of my life.

I deeply appreciate Professor Wojciech P. Maly, the inventor of VeSFET, and as my co-advisor and another mentor. He always provided me valuable comments, critiques, and out-of-the-box ideas with his deep insights of VeSFET. I admire his passion of researches and his determination of driving new technology for the human society, which inspired me every time we talked or met.

I want to express my gratitude to Professor Kwang-Ting Tim Cheng and Professor Li-C. Wang. Their valuable and constructive comments and suggestions helped me to shape my researches and this dissertation better.

This work is a fruit of collective efforts. I am grateful to Dr. Xiang Qiu and Dr. Vivek S. Nandakumar, for sharing their knowledge and past works on VeSFET research. I really appreciate my mentors in IBM Research, Dr. Terence B. Hook, Dr. Philip J. Oldiges, and Dr. Bruce B. Doris, for their guidance and knowledge on device level researches, which gave me deeper understanding on VeSFET behavior in advanced technology nodes. I thank Professor

Dominik Kasprowicz and Professor Andrzej Pfitzner, for improving the VeSFET transistor models, which made the experiments possible. I want to express my appreciation to all former or current colleagues and friends in the lab: Dr. Miguel Lastras, Ali Abbasinasab, Dr. Zhong Guan, Shabnam Larimian, Dr. Chun-Kai Hsu, Dr. Fan Lin, Dr. Amirali Ghofrani, Dr. Rui Wu, Dr. Nicole Fern, Yuyang Wang, and Leilai Shao. The conversations with you were always helpful and joyful, which enriched my research and life.

I would like to express my deepest gratitude to my dearest family: my parents, my wife Chin-Mao, and my younger sister Hui-Chen, and my dear friends living nearby on the same Ph.D. journey. They are my strongest supports and always there for sharing my happiness, pressure, or even depression. Especially for my wife Chin-Mao, she is always understanding and supportive, taken care of our daily life and our little daughter, which allowed me to stride ahead.

At the end, I deeply appreciate the support of National Science Foundation (NSF) under Grant CCF-1320401. In addition, I want to express my gratitude to the computing resource support from the Center for Scientific Computing from the California NanoSystems Institute (CNSI) and Materials Research Laboratory (MRL) at the University of California, Santa Barbara, supported by NSF MRSEC (DMR-1121053) and NSF (CNS-0960316). Furthermore, I would like to thank the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, for all the help of administrative works and IT infrastructure.

VITA OF PING-LIN YANG

December 2017

EDUCATION

Doctor of Philosophy in *Electrical and Computer Engineering*
University of California, Santa Barbara, December 2017 (expected)
Advisor: Professor Malgorzata Marek-Sadowska

Master of Science in *Electrical Engineering*
National Tsing Hua University, Hsinchu, Taiwan, July 2007
Advisors: Professor Yarsun Hsu, Professor Ching-Te Chiu, and Professor Shuo-Hung Hsu

Bachelor of Science in *Electrical Engineering*
National Tsing Hua University, Hsinchu, Taiwan, June 2005

PROFESSIONAL EMPLOYMENT

09/2013 – 12/2017: **Graduate Student Researcher**, VLSI CAD Lab, Department of Electrical and Computer Engineering, University of California, Santa Barbara

06/2015 – 09/2015: **Research Intern**, IBM, Albany, NY

10/2007 – 09/2013: **R&D Principal Engineer**, TSMC, Hsinchu, Taiwan

07/2005 – 07/2007: **Research Assistant**, High Performance Computer & Communication Lab, Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan

09/2006 – 01/2007: **Teaching Assistant**, Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan

09/2002 – 10/2006: **Web and Database Programmer**, Computer and Communication Center, National Tsing Hua University, Hsinchu, Taiwan

PUBLICATIONS

[1] **P. -L. Yang**, X Qiu, and M. Marek-Sadowska, "VeSFET monolithic 3D technology physical design assessment," *IEEE Transactions on Very Large Scale Integration Systems (T-VLSI)*, 2017 (Submitted and under review process)

[2] **P. -L. Yang** and M. Marek-Sadowska, "High performance architecture using fast dynamic reconfigurable accelerators," *IEEE Transactions on Very Large Scale Integration Systems (T-VLSI)*, 2017 (Reviewed, minor revision, and under further review process)

[3] **P. -L. Yang** and M. Marek-Sadowska, "Making split-fabrication more secure," in *Proc. of the 35th IEEE/ACM International Conference on Computer-Aided Design (ICCAD'16)*, Nov. 2016, pp. 1-8

[4] **P. -L. Yang** and M. Marek-Sadowska, "A fast, fully verifiable, and hardware predictable ASIC design methodology," in *Proc. of the 34th IEEE International Conference on Computer Design (ICCD'16)*, Oct. 2016, pp. 364-367

[5] **P. -L. Yang**, T. B. Hook, P. J. Oldiges, and B. B. Doris, "Vertical slit FET at 7-nm node and beyond," *IEEE Transactions on Electron Devices (T-ED)*, vol. 63, no. 8, Aug. 2016, pp. 3327-3334

[6] H. Hsieh, S. H. Dhong, C. -C. Lin, M. -Z. Kuo, K. -F. Tseng, **P. -L. Yang**, K. Huang, M. -J. Wang, and W. Hwang, "Custom 6-R, 2- or 4-W multi-port register files in an ASIC SOC with a DVFS window of 0.5 V, 130 MHz to 0.96 V, 3.2 GHz in a 28-nm HKMG CMOS technology," in *Proc. of the IEEE Custom Integrated Circuits Conference (CICC'15)*, Sep. 2015, pp. 1-3

[7] **P. -L. Yang**, M. Marek-Sadowska, and W. Maly, "Performance assessment of VeSFET-based SRAM," in *Proc. of the* 11th *IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC'15)*, Jun. 2015, pp. 79-82

[8] M. -Z. Kuo, H. Hsieh, S. Dhong, **P. -L. Yang**, C. -C. Lin, R. Tseng, K. Huang, M. -J. Wang, and W. Hwang, "A 16 kB tile-able SRAM macro prototype of an operating window of 4.8GHz at 1.12V VDD to 10 MHz at 0.5V in a 28-nm HKMG CMOS," in *Proc. of the IEEE Custom Integrated Circuits Conference (CICC'14)*, Sep. 2014, pp. 1-4

[9] S. Dhong, R. Guo, M. -Z. Kuo, **P. -L. Yang**, C. -C. Lin, K. Huang, M. -J. Wang, and W. Hwang, "A 0.42V Vccmin ASIC-compatible pulse-latch solution as a replacement for a traditional master-slave flip-flop in a digital SOC," in *Proc. of the IEEE Custom Integrated Circuits Conference (CICC'14)*, Sep. 2014, pp. 1-4

[10] **P. -L. Yang**, C. -C. Lin, M. -Z. Kuo, S. -H. Dhong, C. -M. Lin, K. Huang, C. -N. Peng, and M. -J. Wang, "A 4-GHz universal high-frequency on-chip testing platform for IP validation," in *Proc. of the 32nd IEEE VLSI Test Symposium (VTS'14)*, Apr. 2014, pp. 1-6

[11] M. -Z. Kuo, O. Takahashi, **P. -L. Yang**, C. -C. Lin, M. -J. Wang, P. -W. Wang, and S. -H. Dhong, "A HKMG 28nm 1GHz fully-pipelined tile-able 1MB embedded SRAM IP with 1.39mm^2 per MB," in *Proc. of the IEEE Custom Integrated Circuits Conference (CICC'13)*, Sep. 2013, pp. 1-4

[12] C. -T Chiu, Y. -H. Hsu, J. -M Wu, S. -H. Hsu, M. -S. Kao, H. -C. Tzeng, M. -C. Du, **P. -L. Yang**, M. -H. Lu, F. Chen, H. -Y. Lin, and Y. -S. Hsu, "An 8×8 20 Gbps reconfigurable load balanced TDM switch IC for high-speed networking," *Journal of Signal Processing Systems*, vol. 66, no. 1, Jan. 2012 pp. 57-73

[13] Y. -H. Hsu, M. -H. Lu, **P. -L. Yang**, F. -T. Chen, Y. -H. Li, M. -S. Kao, C. -H. Lin, J. -M. Wu, S. -H. Hsu, Y. S. Hsu, and C. -T. Chiu, "A quarter-rate 2.56/3.2Gbps 16/20:1 SERDES interface in 0.18μm CMOS technology," *International Journal of Electrical Engineering (IJEE)*, vol. 16, no. 1, Feb. 2009, pp. 39-49

[14] Y. -H. Hsu, M. -H. Lu, **P. -L. Yang**, F. -T. Chen, Y. -H. Li, M. -S. Kao, C. -H. Lin, C. -T. Chiu, J. -M. Wu, S. -H. Hsu, and Y. -S. Hsu, "A 28Gbps 4×4 switch with low jitter SerDes using area-saving RF model in 0.13μm CMOS technology," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'08)*, May. 2008, pp. 3086-3089

[15] C. -T. Chiu, J. -M. Wu, S. -H. Hsu, Y. Hsu, M. -H. Lu, **P. -L. Yang**, F. -T. Chen, Y. -H. Li, Y. -H. Hsu, and M. -S. Kao, "A quarter-rate 2.56/3.2Gbps 16/20:1 SERDES interface in 0.18μm CMOS technology," in *Proc. of the 18th VLSI Design/CAD Symposium*, Aug. 2007, pp. 634-637

[16] C. -T. Chiu, Y. -H. Hsu, M. -S. Kao, H. -C. Tzeng, M. -C. Tu, **P. -L. Yang**, M. -H. Lu, F. Chen, H. -Y. Lin, J. -M. Wu, S. -H. Hsu, and Y. -S. Hsu, "A scalable load balanced Birkhoff-von Neumann symmetric TDM switch IC for high-speed networking applications," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'07)*, May. 2007, pp. 2754-2757

[17] C. -H. Hsiao, M. -S. Kao, C. -H. Jen, Y. -H. Hsu, **P. -L. Yang**, C. -T. Chiu, J. -M. Wu, S. -H. Hsu, and Y. -S. Hsu, "A 3.2 Gbit/s CML transmitter with 20:1 multiplexer in 0.18 CMOS technology," in *Proc. of the 13th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES'06)*, Jun. 2006, pp. 179-183

PATENTS

[1] "Contactless communications using ferromagnetic material," U.S. Patent 9,412,721, Aug. 9, 2016

[2] "Electron beam lithography methods including time division multiplex loading," U.S. Patent 9,378,926, Jun. 28, 2016

[3] "Clock regenerator," U.S. Patent 9,362,899, Jun. 7, 2016

[4] "Charged particle lithography system with a long shape illumination beam," U.S. Patent 9,202,662, Dec. 1, 2015

[5] “Electron beam lithography systems and methods including time division multiplex loading,” Korea Patent 10-1563055-0000, Oct. 19, 2015

[6] “Electron beam lithography systems and methods including time division multiplex loading,” U.S. Patent 8,941,085, Jan. 27, 2015

[7] “3D IC configuration with contactless communication,” U.S. Patent 8,921,160, Dec. 30, 2014

[8] “Contactless communications using ferromagnetic material,” U.S. Patent 8,760,255, Jun. 24, 2014

[9] “Conditional cell placement,” U.S. Patent 8,560,997, Oct. 15, 2013

[10] “3D IC configuration with contactless communication,” U.S. Patent 8,513,795, Aug. 20, 2013

[11] “Power-down circuit with self-biased compensation circuit,” U.S. Patent 7,760,009, Jul. 20, 2010

[12] “Power-down circuit with self-biased compensation circuit,” China Patent CN101753126A, Jun. 23, 2010

ABSTRACT

Assessing VeSFET Monolithic 3D Technology in Physical Design,

Dynamic Reconfigurable Computing, and Hardware Security

by

Ping-Lin Yang

With the continuous demands on integrating more functions and devices on a single chip, the technology has been evolving along the scaling path for decades. The transistor feature size has been scaled down from μm order toward 7nm, 5nm, and even below. Conventional MOSFET / FinFET devices are approaching physical limitations. It is extremely difficult to integrate more devices solely by further transistor scaling. Besides scaling, 3D integration technologies offer attractive features. By stacking devices, it increases device density and reduces wire length, which implies better PPA (performance, power consumption, and area). However, the increased power density and the extra overhead of inter-tier connections are significant concerns for deploying 3D integration technologies. For sustaining future technology growth, it is expected that fundamental changes of device structure are required.

Vertical Slit Field Effect Transistor (VeSFET) is a novel transistor with unique structure and characteristics. It is two-side accessible and low power consuming, which is 3D integration friendly. This dissertation investigates VeSFET technology and proposes unique and powerful applications, which are not feasible using MOSFET technologies. The scope of

our studies includes SRAM, monolithic 3D physical design assessment, 3D FPGA, fast and hardware predictable ASIC design methodology, high performance reconfigurable architecture with dynamic reconfigurable accelerators, and hardware security.

In this dissertation, SRAM performance assessment shows that VeSFET SRAM is speed competitive to CMOS SRAM and consumes much less power. The monolithic 3D physical design assessment compares MOSFET and VeSFET monolithic 3D integrated circuits. In particular, the IR-drop on power delivery network (PDN) and clock distribution network (CDN) characteristics are assessed. Due to VeSFET's lower power consumption, 3D VeSFET ICs have lower IR-Drop and CDN power consumption. A fast, fully verifiable, and hardware predictable ASIC design methodology using VeSFET 3D FPGA is presented. The performance comparison of VeSFET 2D and 3D FPGA shows that the 3D FPGA is faster, smaller, and consumes less power. A high performance reconfigurable architecture is proposed using VeSFET fast reconfigurable 3D FPGA-based accelerators with novel bitstream replacement method. The system level performance evaluation shows significant improvement over the system with no accelerator or with conventional FPGA-based accelerator. Toward trustworthy hardware designs, a secure split-fabrication method using VeSFET is proposed for addressing hardware security concerns, such as piracy prevention, hardware Trojan prevention and detection. Any Trojan insertion or design tampering can be easily detected.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Challenges of Future ICs

With the continuous demands on integrating more functions and devices on a single chip, the technology has been evolving along the scaling path for decades. However, recent and coming technologies such as 10nm, 7nm, 5nm, and below are approaching physical limitations. It is extremely difficult to integrate more devices solely by further transistor scaling. Although a chip can have more devices and functions by growing its size on 2D *X*- and *Y*-directions, there are limitations. Larger chips are more likely of having defects, which reduces yield. The longer signal wire lengths degrade the chip performance. Furthermore, drastic increase of power consumption and heat generation per unit chip area are unaccompanied by improvements in heat dissipation efficiency. These phenomena result in dark silicon effect when only a fraction of devices present on a chip can be operated at the same time. Dark silicon effect gradually limits the growth of future ICs. According to the results in [1-2], over 50% of the chip will not be utilized at 8nm.

However, the need of higher computing power is endless. Abundant applications in our daily life require faster computer chips. Such performance demand can be satisfied by integrating more devices together, better physical implementation method, or using more efficient computing architectures. In physical design's aspect, instead of growing chip size

along 2D directions, devices can be stacked vertically in 3D *Z*-direction allowing for integrating more devices within the same footprint. 3D integrated circuits (3D ICs) have been a focus of attention for years. They offer higher device density, smaller chip footprint, and shorter interconnect lengths. 3D ICs with better PPA (performance, power consumption, and area) over conventional 2D ICs have been reported in [3-16]; however, there are tradeoffs [9].

In computing architecture's aspect, power and utilization walls hinder further increase of general processors' computing capability achievable through scaling, which had been the main driver of the electronic industry for decades. To address this bottleneck, the ideas of moving specific tasks from general-purpose cores to specially designed computing units have been proposed. Conservation cores architectures [17] delegate jobs to specialized processors to save energy. Accelerator-rich architecture [18-20] includes a sea of heterogeneous dedicated hardware accelerators implementing different functions that may be invoked by applications running on the system. The performance, either speed or power, of specific applications can benefit from these specialized computing units. However, a system may need to handle wide spectrum of applications. Data centers or cloud computing systems need to respond to all kinds of requests; their workloads demand flexible and efficient computing platforms. In such cases, it is impractical to provide all kinds of specialized accelerators in accelerator-rich architectures. To address the demands of flexibility and performance, reconfigurable computing architectures provide attractive characteristics for this kind of systems [21-22]. The key idea is to offload system tasks to reconfigurable processing units, such as accelerators implemented by FPGAs. Thus, the functions of processing units can be reconfigured according to the workloads demand. However, the reconfiguration time is one of the concerns toward fast dynamic reconfigurable systems.

Besides the chip performance considerations, making trustworthy hardware has attracted designers' attentions especially for those sensitive applications. Today many design houses must outsource their design fabrication to a third party which is often an overseas foundry. Split-fabrication is proposed for combining the FEOL capabilities of an advanced but untrusted foundry with the BEOL capabilities of a trusted foundry [23-25]. Hardware security in this business model relates directly to the front-end foundry's ability to interpret the partial circuit design it receives in order to reverse engineer or insert malicious circuits. The published experimental results indicate that a relatively large percentage of the split nets can be correctly guessed [26] and there is no easy way of detecting the possibly inserted Trojans.

1.2 3D Integration Technologies

Although 3D ICs offer attractive features, there are tradeoffs [9] for integrating circuits vertically, such as higher power density and extra vertical inter-tier connections passing through device layers. Those inter-tier connection channels may occupy parts of the chip resulting in reduced area and placement freedom for transistors and other devices. The increased power density causes more problems with IR-drop and heat dissipation. In the era of aggressively scaled transistors and low VDD power supply, the acceptable maximum IR-drop is tightly constrained. Also, the increased heat dissipation may cause various temperature dependent problems, such as reliability, shorter product life, as well as unmodeled device and performance behaviors. Even worse, it may lead to dark silicon effect [8] [9] when devices on a chip cannot be fully operated due to the limited power and thermal budgets. The increasing power consumption is one of the key issues that hinder the growth of 3D integration; it can even be a showstopper. There are several 3D integration technologies; the best known uses

Through-Silicon VIAs (TSVs). TSVs connect two or more independent dies vertically with μ-bumps and costs extra area overhead, which is not ignorable. Newer technology called monolithic 3D (3Dm) integration integrates transistors and interconnects tier-by-tier on the same wafer. The inter-tier interconnects are implemented as small monolithic inter-tier VIAs (MIVs), whose sizes are like the usual metal VIAs [8]. Thus, abundant vertical inter-tier channels are feasible with these small MIVs. Although MIVs' area overhead is much less than TSVs', a portion of the die size needs to be reserved for MIVs. The area overhead is not zero.

1.3 VeSFET (Vertical Slit FET)

MOSFET technology has dominated the IC industry for decades and is approaching the physical limitation. Engineers and scientists have been working hard for a good substitution, which can continue the future growth. To keep technology rolling, it is expected that fundamental changes will be required.

Vertical Slit Field Effect Transistor (VeSFET) is a novel transistor with attractive characteristics, such as two-side accessibility, lower power consumption, high regularity, circle-based patterning, and good thermal properties [27-40]. The device's two-side accessible pillars are two-side routable, which mitigates routing congestion issues [35] in 2D designs and make the device 3D-integration friendly without extra area overhead for inter-tier connections. Reference [36] describes a three-tier 3D VeSFET SRAM cell. Furthermore, VeSFET has good thermal properties [37-38], which is another key advantage for 3D ICs. It has a special structure that is very different than those of the existing MOSFETs or FinFETs. The prototypes in 65nm-equivalent technology have been successfully fabricated [28] [29]. A device level study for 7nm equivalent technology shows that VeSFET's characteristics are

competitive to FinFET with comparable dimensions [31]. These characteristics make it a potentially good technology for future 2D/3D ICs.

1.4 Dissertation Organization

This dissertation investigates VeSFET monolithic 3D integration technology in system and physical levels. The domains include SRAM, physical design, design methodology, FPGA, dynamic reconfigurable architecture, and hardware security. Unique application and architectures are proposed, which are not feasible using MOSFET technologies. Chapter 2 introduces the fundaments of VeSFET technology including device, circuit, layout, SRAM, and monolithic 3D integration. SRAM performance is assessed and presented in Section 2.3. A physical design assessment of VeSFET monolithic 3D integration is provided in Chapter 3; which includes power delivery network (PDN)'s IR-drop analysis and clock distribution network (CDN)'s characteristics. Chapter 4 proposes a fast, fully verifiable, and hardware predictable ASIC design methodology using 3D FPGAs. The performance of VeSFET 3D FPGA is assessed. In Chapter 5, a high performance dynamic reconfigurable architecture using fast dynamic reconfigurable accelerators is proposed and assessed. In Chapter 6, a secure split-fabrication method is proposed for trustworthy hardware. Chapter 7 concludes this dissertation and discusses future opportunities.

Chapter 2

VeSFET Technology

Emerging technologies provide opportunities for more efficient implementations of many applications. Vertical Slit Field Effect Transistor (VeSFET) is one of such emerging transistors with unique characteristics, which offer advantages over MOSFET technology.

2.1 Device Structure

VeSFET is a highly regular twin-gate device with four vertical cylinder metal pillars and a horizontal channel. VeSFET's highly regular structure is intended for mitigating the fabrication difficulties while the device size is aggressively scaled. Figure 2.1 shows VeSFET structure and illustrates how transistors are integrated on a chip. The four metal pillars implement four vertical transistor terminals: two gates (G1 & G2), a drain (D), and a source (S). Such pillar structure naturally acts as vertical routing and heat dissipation channel, which is very friendly for 3D integration. The two gates offer better channel and leakage current control. Pillar radius (r) defines the technology feature size. VeSFET can be patterned with $2r$ diameter circles as shown in Figure 2.1 (c). This circle-based patterning offers the potential of optical proximity correction (OPC) free fabrication [28]. This characteristic simplifies OPC process, which is one of the major fabrication concerns while the transistor feature size is aggressively shrunk. The footprint of a single transistor bounded by the centers of four pillars is $4r \times 4r = 16r^2$. VeSFET pillar height (h) determines the driving strength of a single

transistor. *h* is conceptually equivalent to CMOS channel width. Since VeSFET is a twin-gate device, the equivalent transistor width is 2*h*. A VeSFET-based chip is composed of a regular array of metal pillars as shown in Figure 2.1 (b) and (d), each square bounded by four adjacent pillars is a feasible VeSFET location.

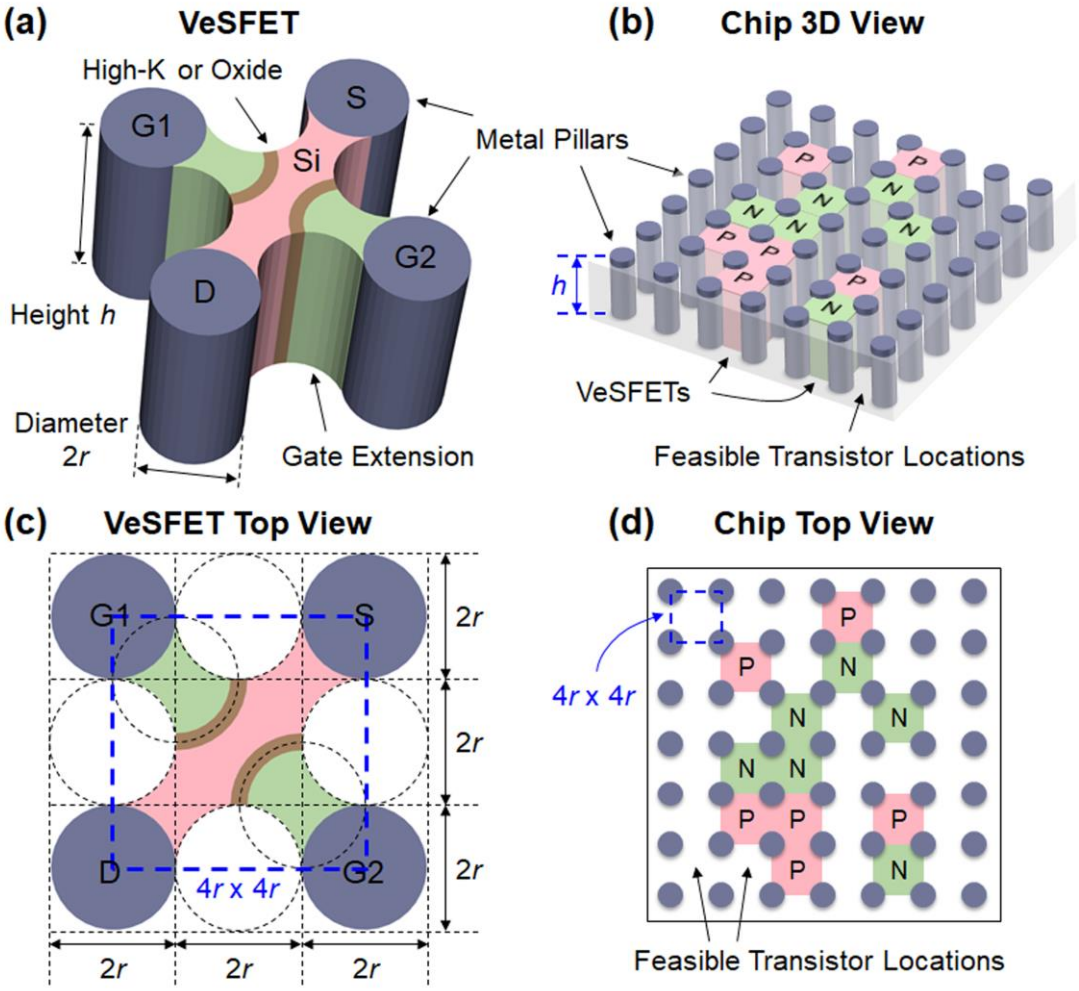


Figure 2.1. VeSFET; (a) 3D view of a single VeSFET; (b) 3D view of a VeSFET-based chip; (c) top view of a single VeSFET; (d) top view of a VeSFET-based chip. VeSFETs are placed as an array formed by metal pillars.

Another unique and useful feature of VeSFET is the possibility of controlling its behavior by properly adjusting signals applied to each of its gates. VeSFET can be implemented as tied gate configuration (TGC) or independent gate configuration (IGC). In a TGC VeSFET, two

gates are hard-wired together and the transistor works as a normal MOSFET with a single gate. IGC VeSFET offers area and circuit design advantages; each of the two gates is controlled by an independent signal. An IGC VeSFET can be realized as an AND- or OR-type transistor by adjusting fabrication parameters [32] [33]. For an AND-type IGC VeSFET, the channel is turned on only when both gates are asserted; for an OR-type IGC VeSFET, it requires only one asserted gate to turn the channel on. Thus, an AND or an OR function can be realized by a single transistor. Table 2.1 summarizes the transistor operation modes of different VeSFET types. Figure 2.2 shows the IGC transistor representation and the circuit implementations of NAND2 and NOR2 functions. Only two IGC-VeSFETs are required for such functions, in contrast to the conventional CMOS implementations that require four transistors. IGC VeSFET also offers an attractive capability for transistor level circuit designs. The designer can treat one of the two gate terminals as a transistor gate and the other as a transistor profile adjuster. Each transistor's profile, such as I-V curve, threshold voltage, etc., can be adjusted by properly controlling one of the two gate terminals.

Table 2.1. AND- and OR-type VeSFETs with TGC and IGC Configurations

Gate Configuration			**AND-type**		**OR-type**	
Configuration	**Gate 1**	**Gate 2**	**P-FET**	**N-FET**	**P-FET**	**N-FET**
TGC / IGC	0	0	On	Off	On	Off
	1	1	Off	On	Off	On
IGC	0	1	Off	Off	On	On
	1	0	Off	Off	On	On

VeSFET in 65nm-equivalent technology has been fabricated [28] [29]. Its chip-level performance is studied in [34] using standard cell design methodology in 65nm-equivalent technology with pillar radius r = 50nm and heights h = 200nm and 400nm. The results show the power delay product is 35% of comparable 65nm CMOS. Reference [31] reports the

device-level comparison of VeSFETs with pillar radius r = 10nm and FinFET in 7nm technology using TCAD simulations. The results indicate high I_{eff} to I_{off} ratio, low gate capacitance, and competitive drivability with respect to a comparable FinFET.

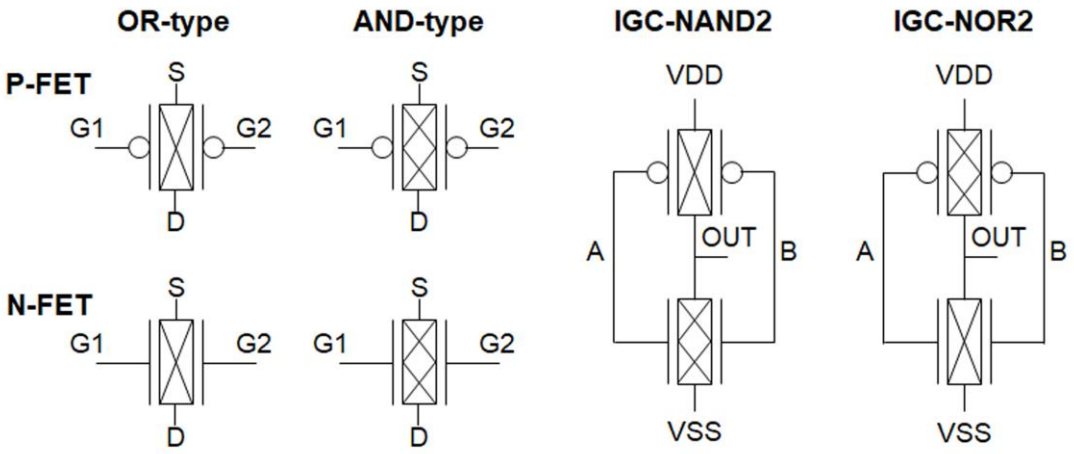


Figure 2.2. Independent gate configuration (IGC) VeSFETs and the circuits.

2.2 Circuits and Layouts

In a VeSFET 2D chip, the devices are placed as a highly regular transistor array called *canvas*. Horizontal, vertical, and diagonal metal directions are feasible [28] [34]. To achieve better design manufacturability, design rules of metal layers are strictly defined using pillar radius r [28]. All wires within a metal layer are parallel, and their locations are aligned with pillars. VeSFET-based 2D design methodology and the performance comparisons to CMOS designs are developed and reported in [34]. It reports that VeSFET designs consume only 35% dynamic power and 2.6% leakage power of the comparable CMOS-based designs.

In the literature, circuits implemented in *basic canvas* and *chain canvas* were studied [34]. Besides them, there is a third implementation called *full canvas*. Figure 2.3 shows 1*P*1*N* inverter layouts implemented using each of them. In the *basic canvas*, the transistor array

consists of devices that do not share pillars. The maximum transistor utilization (i.e. the ratio of the number of used transistors to the maximum number of possible transistors) of *basic canvas* is 25%. In the *full canvas*, two transistors may share two pillars and every minimum size square bounded by four pillars may contain a transistor. Layouts of certain circuit structures may ideally achieve 100% transistor utilization. In the *chain canvas*, two transistors may share one pillar and the maximum transistor utilization is 50%. Reference [34] compared the 2D circuits implemented using *basic* and *chain canvases*. *Chain canvas* yields better results than *basic canvas* because of its denser footprint. Denser layouts using *full canvas* and diagonal metal are possible. The *chain canvas* layout shown in Figure 2.3 has the pillar array 45-degree rotated, which fits well into standard cell layout style using vertical and horizontal metal wires only.

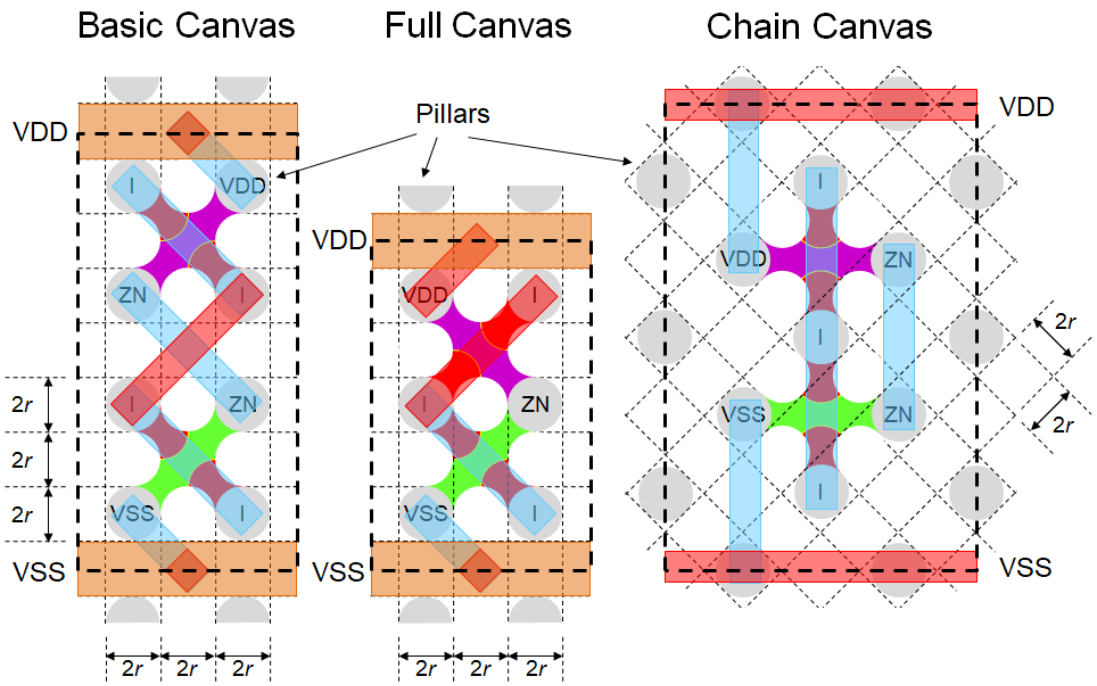


Figure 2.3. Three VeSFET circuit layout styles and the corresponding 1*P*1*N* inverter standard cell implementations.

Since VeSFET's driving current is determined by pillar height, larger and different driving strengths can be achieved with the same footprint. However, the pillar heights of VeSFETs on the same 2D design plane should be the same for process considerations. A quantization effect as in the case of FinFETs is expected; shorter pillars provide better granularity but require larger area for the desired driving strength; higher pillars contribute larger driving current per unit area, but the granularity is poorer.

2.3 SRAMs

SRAMs are fundamental blocks of VLSI systems, which are usually used for technology evaluation. This section provides a VeSFET SRAM performance assessment modeled by *CACTI 6.5* [41], a cache modeling tool. The results show that VeSFET SRAM design is speed competitive to CMOS SRAM with about 40% of dynamic read energy consumption and 35% of total power consumption for read access rate 100MHz.

2.3.1 6T Bit Cell

In planar CMOS SRAM cell designs, the width of pull-up (*PU*), pull-down (*PD*) and pass-gate (*PG*) transistors are well tuned to achieve the best performance for the target specific application domain. These transistors' lengths are also specially designed to achieve high yield and density. In VeSFET SRAM cell design, all transistor widths are equal to the pre-defined VeSFET height, the characteristics of SRAM bit cell can be determined by the type of transistor used. Table 2.2 shows the VeSFET 6T SRAM bit cell pre-layout *Hspice* simulation results with VDD = 0.8V. All transistors are in TGC configuration with radius r = 50nm and height h = 200nm. The VeSFET transistor models are based on TCAD simulation

results [39-40]. The static noise margin (*SNM*) is measured in the closed word-line (*WL*) scenario.

Table 2.2. VeSFET 6T SRAM Bit Cell Performance, VDD = 0.8V

VeSFET Type, TGC			*r* = 50nm, *h* = 200nm, by *Hspice*			
PU	***PD***	***PG***	***SNM***	**Write Voltage**	**Read Current**	**Leakage Current**
OR	OR	OR	299mV	217mV	10.44uA	93.81pA
AND	OR	OR	238mV	346mV	10.44uA	10.85pA
AND	AND	OR	346mV	356mV	3.58uA	0.036pA

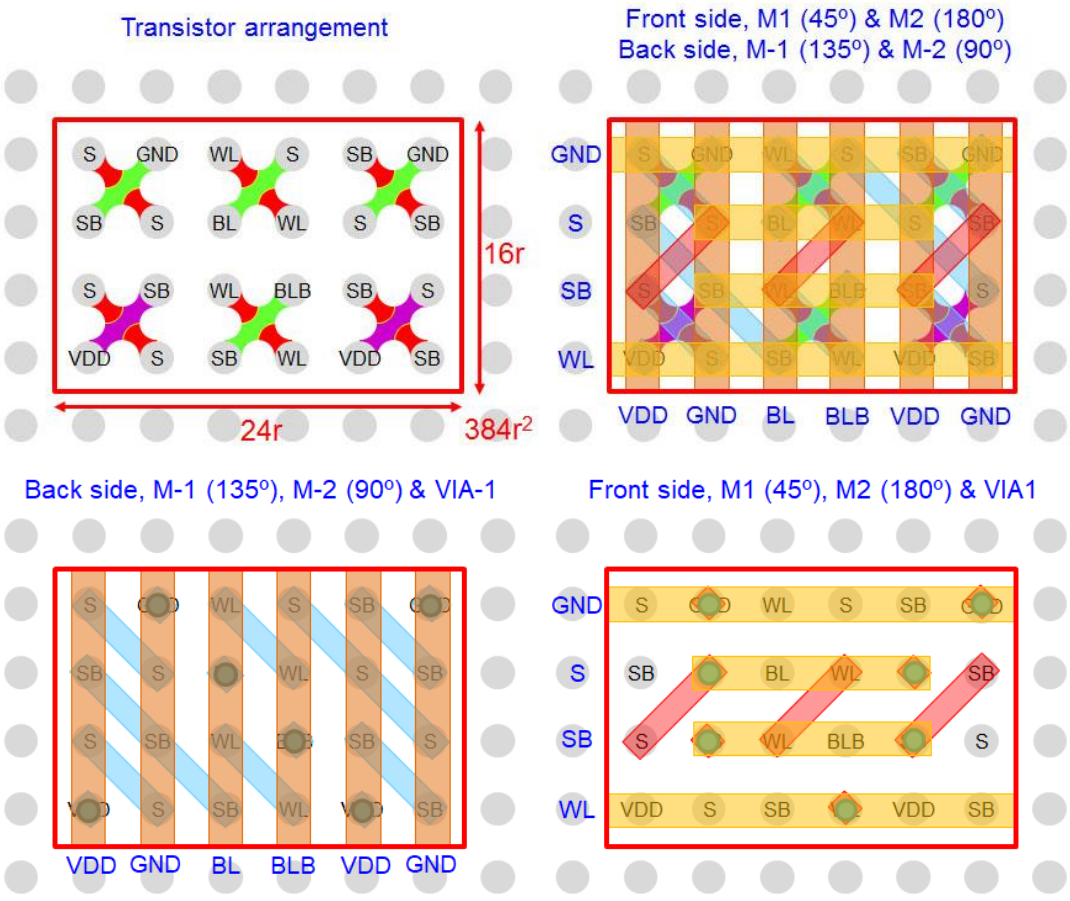


Figure 2.4. VeSFET cell layout *v0*, cell size = 24*r* x 16*r* = 384r^2

In VeSFET designs, metal pillars can be shared, which reduces the area and parasitic capacitance. For the worst-case scenario without any pillar sharing, the area of a 6T SRAM cell is 24*r* x 16*r* = 384r^2. Figure 2.4 shows the transistor arrangement and the layout of such a worst area SRAM bit cell *v0* with wire connections on both front and backside. A 3D diagram of cell *v0* is shown in Figure 2.5. *M-1* and *M-2* are two metal layers on the backside; *M1* and *M2* are on the front side. The metal wires in the same layer are all aligned in a unified direction and all metal segments start and end at pillars. These properties increase the regularity of layout patterns and reduce the manufacturing process complexity.

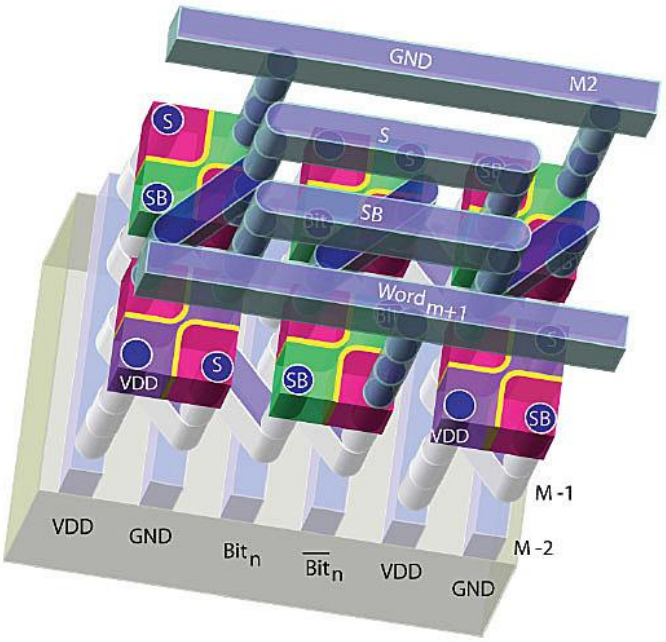


Figure 2.5. VeSFET cell *v0* 3D diagram (Courtesy of Professor Wojciech P. Maly, Department of Electrical and Computer Engineering, Carnegie Mellon University)

Two other one-side routed VeSFET layouts *v1* and *v2* are shown in Figure 2.6 and 2.7, respectively. Cell *v1* is optimized for power and speed, where *BL* and *WL* lengths within the cell are 12*r* and 28*r*; the cell area is 336r^2. Cell *v2* is optimized for area, where *BL* and *WL*

lengths within the cell are 16*r* and 20*r* and the cell area is 320r^2. If *r* = 50nm, the area of cell *v1* is 0.6µm x 1.4µm = 0.84µm^2, and of cell *v2* is 1µm x 0.8µm = 0.8µm^2. In these two layouts, all the transistors are TGC configured. Metal 1 (*M1*) is 135° oriented; metal 2, 3 and 4 (*M2*, *M3*, and *M4*) are 45°, 180° and 90° oriented respectively. The transistors and wires are well

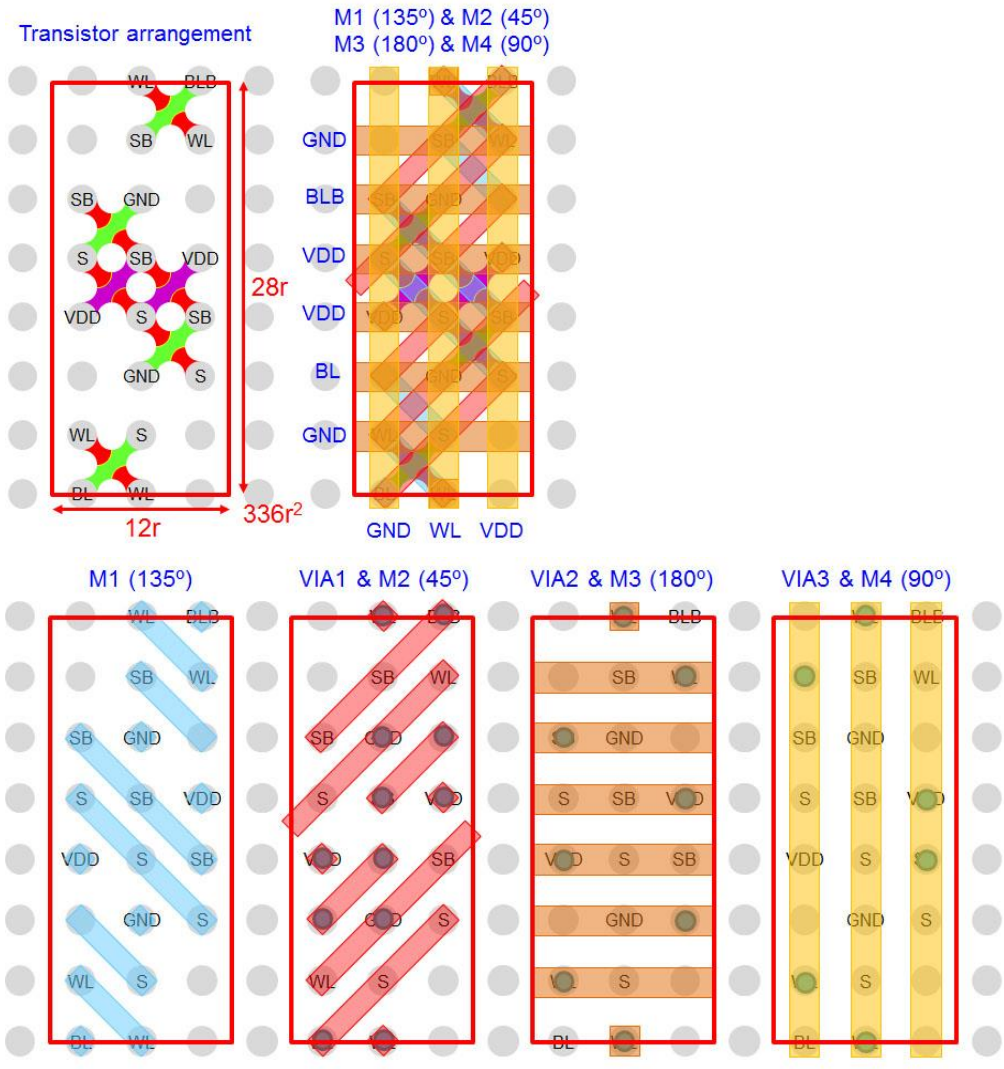


Figure 2.6. VeSFET cell layout *v1*, cell size = 12*r* x 28*r* = 336r^2

arranged, which allows for the cell abutment to form a sub-array, the VDD and GND power straps are all aligned with *BL* and *BLB* shielded to enhance power network robustness and to reduce *BL* coupling noise. In a traditional read scheme, *BL*s are pre-charged to VDD and 1)

keep floating if the read data is 1; 2) pulled down by a cell if the read data is 0. The voltage level on a floating *BL* could degrade due to the coupling effect from adjacent *BL*s pulling down. This degraded voltage level could cause read errors.

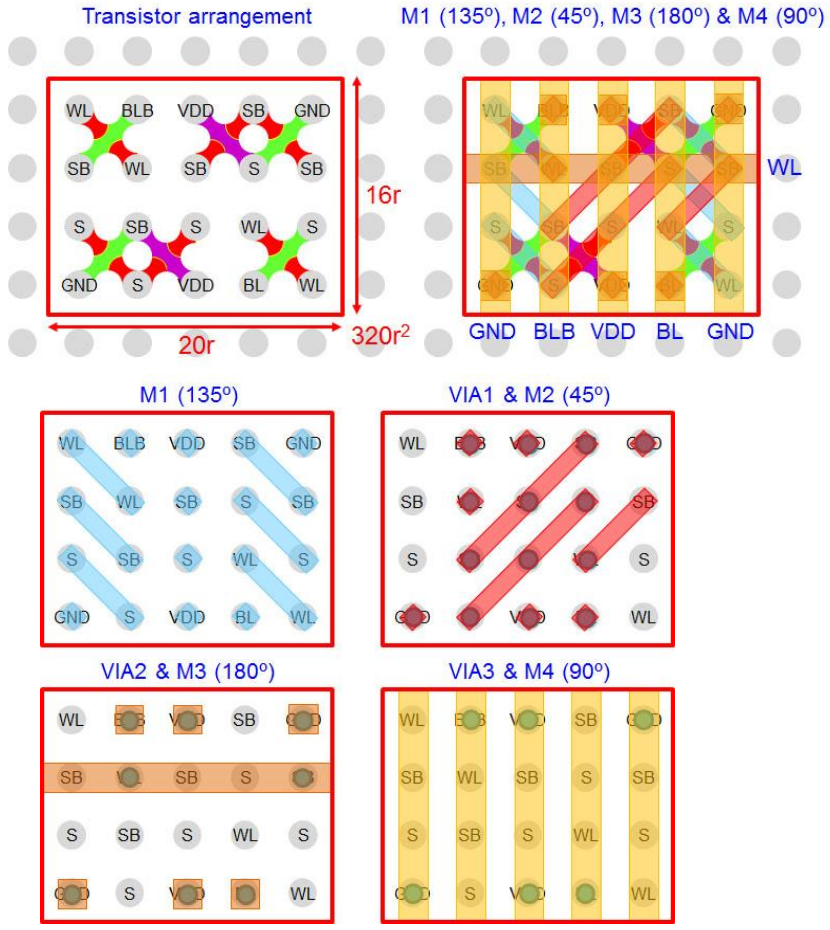


Figure 2.7. VeSFET cell layout *v2*, cell size = 20*r* x 16*r* = 320r^2

2.3.2 SRAM Array Performance

To evaluate VeSFET SRAM performance and to compare with CMOS designs, a CMOS cache modeling tool *CACTI 6.5* [41] is used and modified to support VeSFET technology. *CACTI*'s modeling engine is not touched to guarantee that: 1) CMOS SRAM results are not changed by the tool modifications, and 2) the modeling methods are the same. VeSFET

technology parameters required for *CACTI*'s computation are extracted by *Hspice* simulation based on the VeSFET transistor model with radius r = 50nm. There are three ITRS-based CMOS models originally embedded in *CACTI*: High Performance (*HP*), Low Operating Power (*LOP*), and Low Standby Power (*LSTP*). For our comparisons we assumed the default *CACTI* CMOS SRAM cell of area $146F^2$ (F is the feature size) and aspect ratio 1.46 [42]. Figure 2.8 shows the cell footprint from [43] which is referenced in [42] as the default cell model in *CACTI*. For 65nm technology, the cell size is 0.65µm x 0.949µm = 0.61685µm^2. Tables 2.3 and 2.4 summarize four simulated SRAM cells and six array configurations.

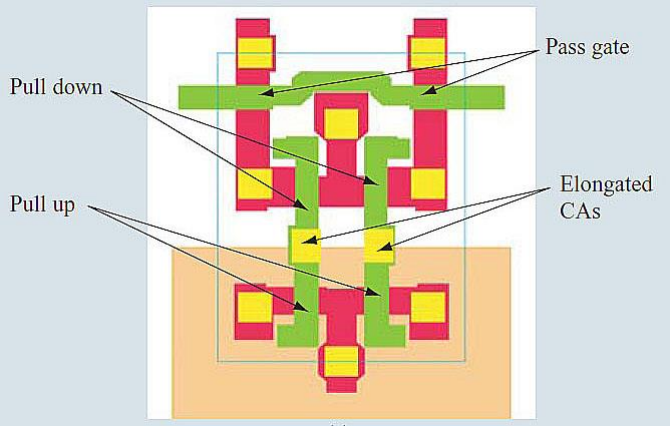


Figure 2.8. Default *CACTI* CMOS cell footprint [43] with area = $1.46F^2$ (F is transistor feature size) and aspect ratio 1.46

Table 2.3. SRAM Cells in the Simulation

Cells	WL Length (µm)	BL Length (µm)	Area (µm^2)
CACTI Default 65nm CMOS	0.65	0.949	0.61685
VeSFET cell *v0*, r = 50nm	1.2	0.8	0.96
VeSFET cell *v1*, r = 50nm	1.4	0.6	0.84
VeSFET cell *v2*, r = 50nm	1.0	0.8	0.80

Table 2.4. SRAM Array Configurations in the Simulation

Size (kByte)	Cells / WL	Cells / BL
0.5	32	32
2	64	64
4	128	64
8	128	128
16	256	128
32	256	256

In this simulation, OR-type Tied-Gate-Controlled (TGC) VeSFET SRAM with radius r = 50nm is compared to the *CACTI*'s default ITRS-based 65nm CMOS technology. We note here that it is not straightforward to decide which CMOS technology node is equivalent to VeSFET technology with r = 50nm. Here we assume equivalence based on metal layout design rules. If device layer rules were adopted, CMOS 65nm node would be equivalent to VeSFET with r much less than 50nm. The VeSFET height h is set to 200nm. The VDD of VeSFET is 0.8V, and the default 65nm CMOS VDDs in *CACTI* are 1.1V (*HP*), 0.8V (*LOP*) and 1.2V (*LSTP*).

We validate the speed and energy consumption of *CACTI*'s VeSFET simulation results. Circuits composed of a single word-line driver (*WLD*) and a word-line (*WL*) with different bit-cell loadings were simulated by *Hspice* and compared to the results extracted from *CACTI* simulation. Figure 2.9 shows the results. In *Hspice* simulation, the *WL* is modeled by RC π-models including SRAM's pass-gate (*PG*) gate loadings. The measured delay includes *WLD* delay and the *WL* RC delay from the output of *WLD* to the end of *WL*. It can be observed that the delay number reported by *CACTI* is greater than the delay measured by *Hspice*; this provides additional delay margin of performance modeling. The energy measured is the energy consumed per *WL* activation of a *WLD* driving a *WL* with SRAM gate loadings.

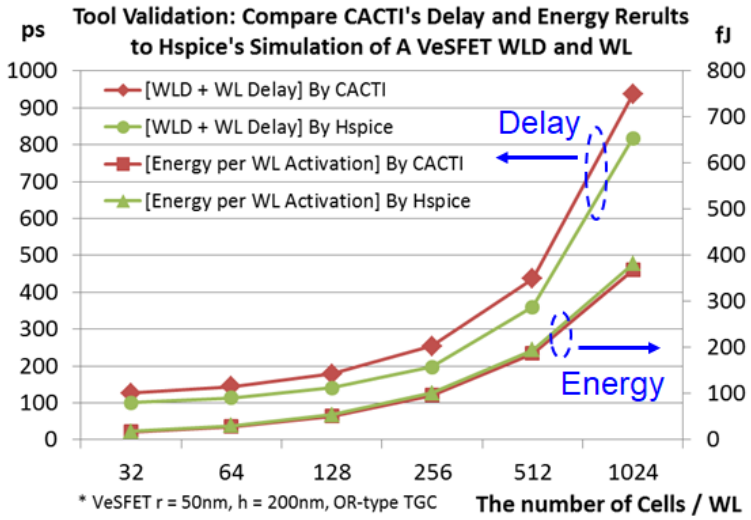


Figure 2.9. Tool validation: *CACTI* and *Hspice* delay and energy simulation results comparison of a *WLD* + *WL*. Based on OR-type TGC VeSFETs.

Tables 2.5 to 2.8 show the speed and power comparison for 65nm CMOS technologies and OR-type TGC VeSFET with height h = 200nm and radius r = 50nm. The speed of VeSFET SRAM is comparable to 65nm CMOS High Performance (*HP*) with about 40% of dynamic read energy consumption. Compared with 65nm CMOS Low Operating Power (*LOP*), the dynamic read energy is about 60%. VeSFETs consume about 2% of 65nm CMOS *LOP* leakage power. Although VeSFET's leakage power is greater than 65nm CMOS Low Standby Power (*LSTP*), VeSFET is much faster with about 35% total power including active and standby parts. VeSFET's twin-gate junctionless structure provides a better channel controllability and smaller parasitic capacitance, which are critical characteristics for low power designs.

Table 2.5. SRAM Array Access Time Comparison (ns)

Size kB	65nm CMOS			VeSFET Cell *v0*			VeSFET Cell *v1*			VeSFET Cell *v2*		
	HP	*LOP*	*LS TP*	ns	% to *HP*	% to *LOP*	ns	% to *HP*	% to *LOP*	ns	% to *HP*	% to *LOP*
0.5	0.81	1.31	1.95	0.63	77%	48%	0.59	73%	45%	0.62	76%	47%
2	0.86	1.40	2.05	0.70	82%	50%	0.66	77%	47%	0.70	81%	50%
4	0.87	1.42	2.10	0.74	85%	52%	0.70	81%	49%	0.73	84%	51%
8	1.00	1.68	2.43	0.94	95%	56%	0.87	87%	52%	0.93	94%	56%
16	1.02	1.72	2.49	1.03	101%	60%	0.97	95%	57%	1.01	99%	59%
32	1.21	2.09	2.97	1.38	114%	66%	1.24	102%	59%	1.36	112%	65%

Table 2.6. SRAM Array Dynamic Read Energy per Access (pJ)

Size kB	65nm CMOS			VeSFET Cell *v0*			VeSFET Cell *v1*			VeSFET Cell *v2*		
	HP	*LOP*	*LS TP*	pJ	% to *HP*	% to *LOP*	pJ	% to *HP*	% to *LOP*	pJ	% to *HP*	% to *LOP*
0.5	2.6	1.7	3.2	1.04	40%	61%	0.99	39%	58%	1.02	40%	60%
2	5.9	3.9	7.1	2.42	41%	61%	2.25	38%	57%	2.39	41%	61%
4	11.2	7.6	13.6	4.72	42%	62%	4.39	39%	58%	4.67	42%	62%
8	14.8	10.2	17.7	6.26	42%	62%	5.58	38%	55%	6.22	42%	61%
16	29.0	19.9	34.6	12.37	43%	62%	11.02	38%	55%	12.28	42%	62%
32	41.3	28.8	49.4	17.81	43%	62%	15.56	38%	54%	17.72	43%	61%

Table 2.7. SRAM Array Leakage Power

Size (kB)	Leakage Power (mW & µW)					
	65nm CMOS			VeSFET		
	HP (mW)	*LOP* (mW)	*LSTP* (µW)	Cell *v0* (µW)	Cell *v1* (µW)	Cell *v2* (µW)
0.5	0.4	0.01	0.02	0.18	0.18	0.18
2	1.6	0.03	0.08	0.59	0.58	0.59
4	2.9	0.05	0.14	1.05	1.04	1.05
8	5.7	0.10	0.28	2.06	2.05	2.06
16	10.7	0.19	0.54	3.87	3.84	3.87
32	21.2	0.38	1.06	7.62	7.61	7.62

Table 2.8. SRAM Array Total Power @ 100MHz

Size (kB)	Total Power (mW) @ Read Access Rate 100MHz					
	65nm CMOS			VeSFET		
	HP **(mW)**	***LOP*** **(mW)**	***LSTP*** **(mW)**	**Cell *v0* (mW)**	**Cell *v1* (mW)**	**Cell *v2* (mW)**
0.5	0.6	0.18	0.32	0.10	0.10	0.10
2	2.1	0.42	0.71	0.24	0.23	0.24
4	4.0	0.81	1.36	0.47	0.44	0.47
8	7.2	1.12	1.77	0.63	0.56	0.62
16	13.6	2.19	3.46	1.24	1.11	1.23
32	25.3	3.26	4.94	1.79	1.56	1.78

These preliminary results are very promising. Although VeSFETs are novel devices and were not well tuned for SRAM performance, VeSFET SRAM is speed competitive with much less power consumption than CMOS bulk. We believe that better VeSFET cell layouts and better device profiles are also possible. VeSFET technology could offer an attractive solution for future VLSI technology.

2.4 Two-side Accessibility and Monolithic 3D Integration

3D stacked transistor layers increase transistor density, support more functions, and the reduced chip size provides higher speed and consumes less power. A 3D IC stack can be implemented using different methods as shown in Figure 2.10. Through-silicon VIAs (TSVs) were developed to vertically connect two or more independent dies with μ-bumps. TSVs pass through the entire die thickness, their cross-sectional area cannot be too small for reliability and process reasons and the available channel count of vertical interconnects is limited. TSVs induce area overhead; other devices cannot use the regions occupied by TSVs.

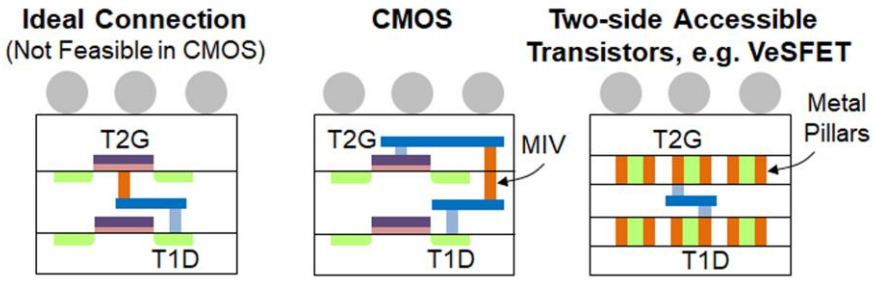


Figure 2.10. Different 3D IC technologies. Two-side accessible transistor monolithic 3D offers shortest cross-tier interconnects.

Monolithic 3D (3Dm) IC technology [3-5] [7-8] [10-16] was proposed to increase vertical channel counts, to reduce the length of vertical interconnect, and to eliminate μ-bump bonding. In MOSFET 3Dm technology, transistor tiers are fabricated on the same wafer and are vertically connected by monolithic inter-tier VIAs (MIVs). Although those MIVs are much smaller than TSVs, they have similar structure.

In MOSFET 3D designs, either with planar FETs, FinFETs, or tri-gate FETs, TSVs and MIVs are implemented on empty space and introduce area overhead. TSV is big and tall, it goes through the whole thickness of a die and connects the adjacent lower die via μ-bumps [5] [8-9]. It has large footprint and keep-out zone for process considerations. Dummy TSVs are often inserted, which costs more area, to guarantee the successful inter-tier connections and for thermal dissipation purposes. TSVs' locations are usually reserved prior to placing active elements. MIVs are very small, and they are usually inserted in empty space after other elements have been placed. The available locations for MIVs are constrained, which implies that their numbers may be limited, especially for designs reaching higher utilization, the location of a MIV may not be optimized, and extra interconnects are required to connect MIVs to their corresponding devices.

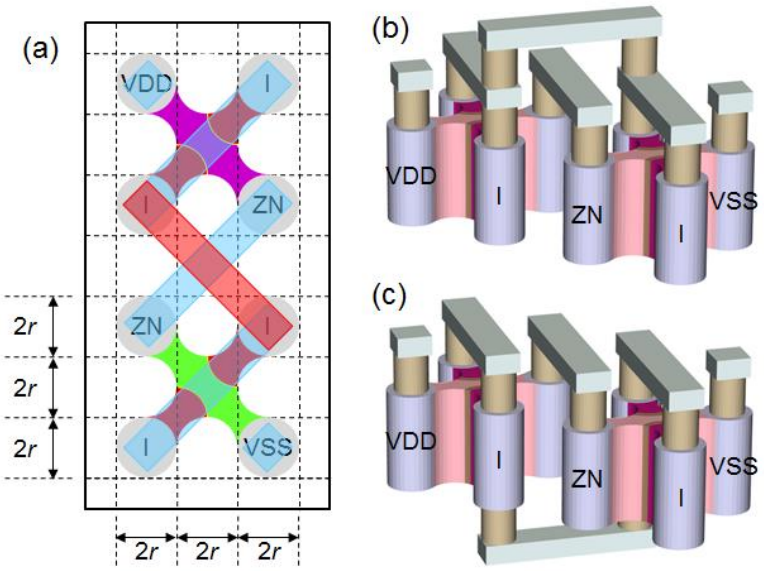


Figure 2.11. A Tied-Gate-Controlled (TGC) 1*P*1*N* inverter. (a) Layout; (b) Conventional structure one-side routed; (c) Two-side routed implementation.

Two-side accessible transistors such as VeSFET are very well suited to implement monolithic 3D integrated circuits. This is a unique characteristic that is not feasible in conventional transistors such as MOSFET, FinFET, SOI, etc. Vertical inter-tier connections can be implemented using front and back sides of a transistor layer. As illustrated in Figure 2.10, the best way to connect transistor 1's drain (T1D) and transistor 2's gate (T2G) is through T2G's bottom; however, this is not feasible in the current MOSFET technology. But in VeSFET technology, every transistor terminal (pillars) on a tier is freely accessible by its adjacent tiers. It offers the capability of creating ideally positioned, short, and less complicated inter-tier interconnects for 3D ICs. Beside 3D ICs, the two-side accessible pillars also benefit 2D ICs. This feature doubles routing resources since the chip can be routed on both the front and back sides [35]. Figure 2.11 (a-b) shows a VeSFET TGC 1*P*1*N* inverter layout routed on one side and Figure 2.11 (c) shows a VeSFET-unique two-sided implementation.

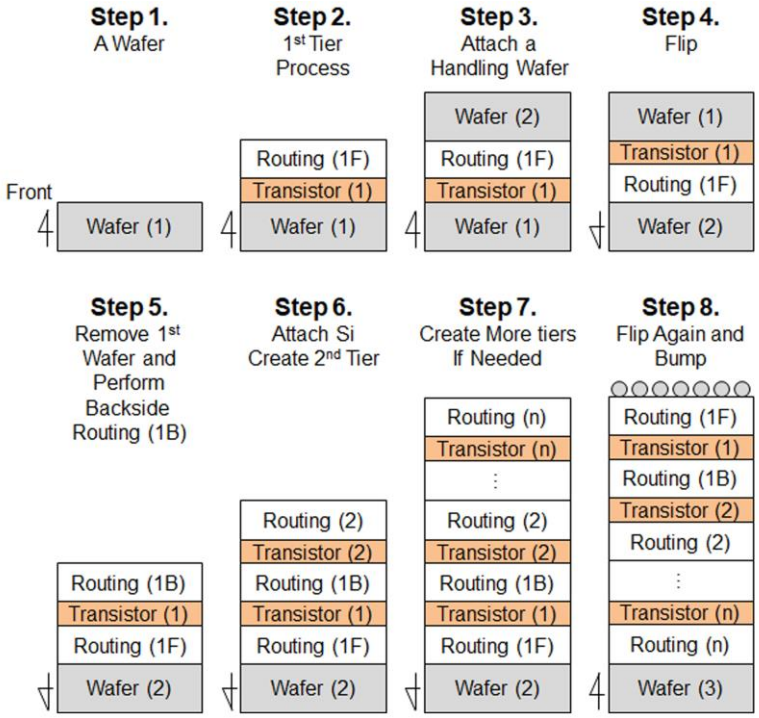


Figure 2.12. VeSFET-based monolithic 3D integration flow. In Step 8, the bumps can be made on the *n-th* tier directly or on the first tier with a flip.

Figure 2.12 illustrates the integration flow of a VeSFET monolithic 3D circuit. The first two steps are the same as in 2D circuits. Then, in the steps 3 to 5, the backside routing is created. The backside routing shares the same BEOL process flow as the front-side routing. This unique capability is achieved by VeSFET's two-side accessible metal pillars. After step 5, a two-side routed 2D chip is formed and is ready to be bumped as a regular 2D chip. Steps 6 and 7 create more tiers and form a monolithic 3D stack. At the end, the designer can decide to do bumping upon the routing on the *n-th* tier directly, or flip again to bump on the first tier. Figure 2.13 shows a cross-sectional view and the 3D structure of a two-tier VeSFET monolithic 3D IC stack.

Reference [8] illustrates the diameter comparison of MIVs (50nm), aggressively sized TSVs (5μm), and mini-TSVs (2μm) in 14nm technology, where the keep-out zones are not

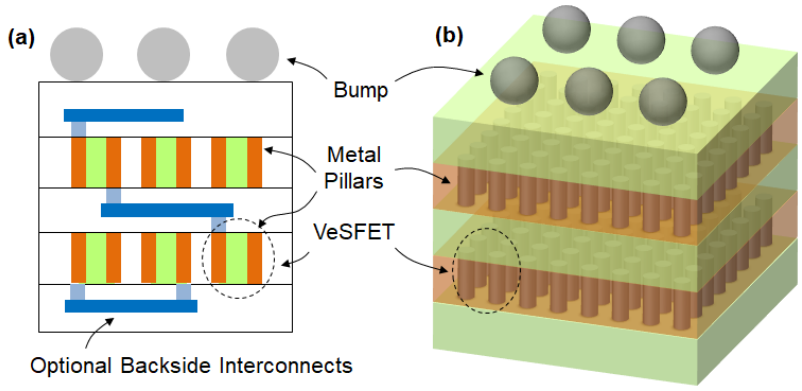


Figure 2.13. VeSFET-based monolithic 3D IC (a) cross-sectional view; (b) monolithic 3D IC structure.

included. The area of a TSV (mini-TSV) is 10000 (1600) greater than the area of an MIV, thus the number of available channel count can be greatly increased by using monolithic 3D integration technology. Fabricated VeSFETs in 65nm-equivalent technology have pillar diameter of 100nm [28-29] [33], and the 7nm-equivalent VeSFETs in [31] have pillar diameter of 20nm. Thus, we can predict the VeSFET pillar diameter in 14nm-equivalent technology to be 40nm. Comparing with the MIV and TSV diameters in 14nm technology reported in [8], the VeSFET pillar diameter is even smaller than an MIV. The comparisons are illustrated in Figure 2.14.

Besides the area overhead for inter-tier connections, the increased power density and thermal dissipation are crucial concerns for 3D IC. They can limit the number of devices integrated together and potentially become a showstopper. Higher power density also requires denser PDN to guarantee every device in the design receives proper power (VDD) and ground (VSS) levels. However, denser PDN reduces available signal routing resources, and lead to poorer PPA [44-45]. Reducing power consumption can mitigate these effects. VeSFET's low-power characteristic [30] [34] is very attractive for 3D integrating more devices, because the

power consumption is maintained on acceptable levels without causing thermal or IR-drop problems. In addition, VeSFET's native metal pillars are evenly and densely distributed across the whole chip area. These pillars behave as a mass count of thermal conduction channels, and help to dissipate heat across the whole stack to reach the heat sink. References [37] and [38] study the thermal performance of a VeSFET chip composed of 10 dies, the temperature increase is only 30% of a comparable CMOS-based chip. These characteristics make VeSFET a promising candidate for 3D integration.

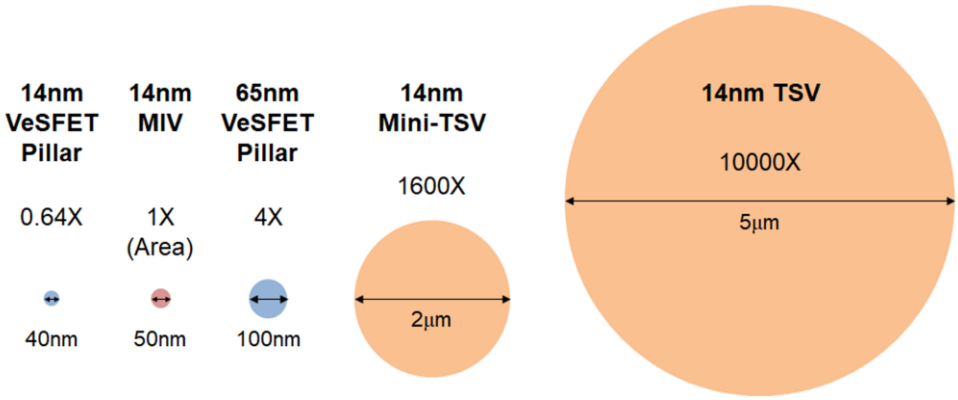


Figure 2.14. The size comparison among different 3D inter-tier interconnects technologies. The area of 14nm MIV [8] is the baseline set to 1X.

Chapter 3

PHYSICAL DESIGN ASSESSMENT

3.1 Introduction

3D circuit integration enables technology evolution when further transistor scaling becomes less feasible. The 3D stacked circuits (3D IC) have higher device density, smaller design footprint, shorter interconnect length, and significantly enhanced interconnect resources. However, higher power density is an unavoidable problem for 3D integrated ICs. Thermal and IR-drop budgets are difficult to satisfy by devices with increased number of transistors integrated together and supplied by low VDD. In this chapter, we investigate a new monolithic 3D integration technology based on Vertical Slit FET (VeSFET), which features low power consumption and self-embedded vertical inter-tier interconnect channels as described in Chapter 2. Thermal has been studied and reported in references [37-38]. In this chapter, we focus on power delivery network (PDN) and clock distribution network (CDN). VeSFET characteristics are assessed and compared with an equivalent CMOS technology. The goal is to assess the properties of VeSFET monolithic 3D integration technology as a path finding research for future 3D ICs.

Ten 2D placed-and-routed benchmark circuits implemented with VeSFET and CMOS technologies are partitioned into 2, 4, and 8 tiers using four different partition methods (Uniform, Thermal-balanced, IR-optimized, and Thermal-optimized). The results show

VeSFET-based monolithic 3D integration is promising even when the device, circuit designs, and layout styles are not tuned and are assumed under pessimistic conditions. On average over all test cases, VeSFET 3D IC's maximum static PDN IR-drop is 38.5% - 52.3% of CMOS designs. VeSFET designs' CDNs consume 70.6% - 73.7% power but have 120.3% - 194.9% clock skews compared to CMOS designs.

This chapter is organized as follows. Section 3.2 presents the methodology of the assessments performed, which include technology assumptions, 2D implementation and 3D partition methods. Detailed results and discussions of PDN and CDN assessments are presented in Sections 3.3 and 3.4, respectively. Section 3.5 summarizes this chapter.

3.2 Assessment Methodology

The purpose of this work is to provide an idea of how VeSFET and CMOS designs compare for monolithic 3D ICs, as a path finding research for future 3D ICs. We focus on physical design aspects, in terms of PDN IR-drop and CDN characteristics. These are crucial for chip performance, where IR-drop margin is small for advanced chips operated under low VDD; CDN affects timing and usually contributes to a great amount of active power consumption.

It is essential that both CMOS and VeSFET designs are implemented and assessed under the same conditions rather than trying to find their optimum design points. We selected 10 benchmark circuits of different sizes from OpenCores [46] and IWLS 2005 benchmark package [47]. They are listed in the order from small to large in Table 3.1 with the descriptions and the number of Flip-Flops (FFs). These circuits are implemented by both 65-nm commercial low power CMOS and 65-nm equivalent VeSFET devices as used in [34].

Table 3.1. The Benchmark Circuits from [46][47]

Name[†1]	Description	# of FFs
s38584	Logic	1159
s38417	Logic	1463
ac97_ctrl	Wishbone AC 97 controller	2530
aes_core	128-bit AES cipher	530
b17	Three copies of 80386 processor	1396
b18	Two copies of Viper processor and b17	3088
sha3	512-bit SHA-3 cryptographic hash function	2221
des3_perf	Triple DES optimized for performance	8808
viterbi_dec	Viterbi decoder compliant with the AXI4-Stream interface	26340
fft_256	Pipelined FFT/IFFT 256 points processor	42990

[†1]: Listed in the order of small to large

Pessimistic conditions are assumed for VeSFET technology: the device, circuits, and layouts are not tuned. *Chain canvas* layout style and TGC VeSFETs are used instead of *full canvas* style and IGC VeSFETs although they provide higher design density as discussed in Section 2.2. The VeSFET model used for standard cell library generation in [34] is based on compact current [39] and capacitance models [40]. First, 2D implementations are obtained using the same standard cell set, the same constraints, the same conditions, and the same design and optimization flows. Then, the 2D designs are partitioned into 2, 4, and 8 tiers using four different partition methods (Uniform, Thermal-balanced, IR-optimized, and Thermal-optimized). The same partition conditions are applied to CMOS and VeSFET designs. The final 3D partitioned designs are then assessed. The detailed methodologies and conditions are described in this section.

3.2.1 Technology Assumptions

The benchmark circuits are implemented with the same flow using comparable CMOS and VeSFET devices. The summary is listed in Table 3.2. VeSFET's pillar radius *r* is 50nm,

and its height *h* is 200nm for all transistors. VeSFET's nominal VDD is 0.8V [28] [34]; to conduct a fair comparison, the CMOS designs also operate with VDD = 0.8V. Although circuits using IGC VeSFETs reduce the number of transistors as discussed in Section 2.1, we use TGC VeSFETs only for making VeSFET circuits comparable to CMOS designs. VeSFET standard cell library contains 61 cells, the cell layouts are designed in *chain canvas* style with cell height 1.4 μm. It is a 10 routing track height library. The same cell set is selected from the CMOS standard cell library for a fair comparison; the cells are of 9-track height.

Table 3.2. Summary of CMOS and VeSFET Technologies

Item	**CMOS**	**VeSFET**
Technology	65-nm	65-nm equivalent OR-type TGC[†2] *r* = 50nm, *h* = 200nm
VDD	0.8V	0.8V
Canvas	-	*Chain canvas*
Standard Cell Height	1.8 μm (9-track height)	1.4 μm (10-track height)
Standard Cells and the Driving Strengthes[†1] (61 cells in total)	D-FF (*D1*), MUX2 (*D1, 2*) INV, BUF (*D1, 2, 4, 8, 16*) AND2, AND3, AND4 (*D1, 2, 4*) OR2, OR3, OR4 (*D1, 2, 4*) NAND2, NAND3, NAND4 (*D1, 2, 4*) NOR2, NOR3, NOR4 (*D1, 2, 4*) AOI211, AOI21, OAI211, OAI21 (*D1, 2, 4*)	

[†1]: (*Dn*) indicates the driving strength of a cell that is n-times of a minimum *D1* cell.

[†2]: OR-type TGC VeSFETs are used to make the circuit structures comparable to CMOS designs. Denser circuit layouts are feasible by using IGC VeSFETs.

The back-end of line (BEOL) metal and VIA characteristics assumed in this work are listed in Table 3.3 and Table 3.4. The characteristics of CMOS and VeSFET are properly aligned with specific width and pitch rules reflecting VeSFET design rules defined in [28]. The CMOS metal thickness, VIA height, metal sheet resistance (R_{sq}), and per-VIA resistance (R_{via}) are all extracted from the commercial technology files. VeSFET's corresponding

characteristics are set to be the same as CMOS, and its R_{via} is calculated by dimension scaling because VeSFET VIA is cylindrical not cuboidal.

Table 3.3. The BEOL Metal and VIA Characteristics

	CMOS			VeSFET		
Metal[†1]	***Mf***	***Mx***	***Mz***	***Mf***	***Mx***	***Mz***
Thickness (μm)	0.18	0.22	0.90	0.18	0.22	0.90
Width (μm)	0.09	0.10	0.40	0.07	0.07	0.28
Pitch (μm)	0.20	0.20	0.80	0.14	0.14	0.56
Rsq (Ω/□)	0.16	0.1399	0.0218	0.16	0.1399	0.0218
VIA		***Vx***	***Vz***		***Vx***	***Vz***
Width (μm) [†2]		0.1	0.36		0.07	0.28
Spacing (μm)		0.1	[†3] 0.34/0.54		0.07	0.28
Height (μm)		0.175	0.595		0.175	0.595
Rvia (Ω/VIA)		1.5	0.22		3.8977	0.463

[†1]: *Mf* is the layer connecting transistors, *Mx* is 1X metal, *Mz* is 4X metal.

[†2]: CMOS's VIAs are cuboid. VeSFET's VIAs are cylinder, the width here is the diameter of a VIA.

[†3]: Different spacing rules applied for different VIA array size.

Table 3.4. Metal Characteristics of Each Layer

CMOS		***M1***[†2]	***M2***	***M3***	***M4***	***M5***	***M6***	***M7***
Direction[†1]		H	V	H	V	H	V	H
Type		*Mf*	*Mx*	*Mx*	*Mx*	*Mx*	*Mz*	*Mz*
VeSFET	***M0***[†3]	***M1***[†2]	***M2***	***M3***	***M4***	***M5***	***M6***	***M7***
Direction[†1]	V	H	V	H	V	H	V	H
Type	*Mf*	*Mx*	*Mx*	*Mx*	*Mx*	*Mx*	*Mz*	*Mz*

[†1]: H: horizontal, V: vertical.

[†2]: Standard cell's power rails are on *M1*.

[†3]: VeSFET's first metal layer *M0* is for connecting transistor terminals, which function is similar to the poly gate layer of CMOS.

We use three types of metals called *Mf*, *Mx*, and *Mz*. *Mf* is the first metal layer connecting transistor terminals. *Mx* is the 1X metal, whose minimum metal width and spacing correspond to the minimum routing track of the technology. *Mz* is the 4X metal suitable for global long wires. Two types of VIAs, *Vx* (1X VIA) and *Vz* (4X VIA) are used to connect different metal layers.

To address the different wiring requirements of short local signals, long global signals, CDNs, and PDNs, using metal layers with different thicknesses and widths is a common solution. In this work, we use 4 *Mx* and 2 *Mz* metal layers above standard cells for routing as listed in Table 3.4. To align the naming of metal layers, the first metal layer in VeSFET technology is called *M0*, whose function is similar to the poly gate layer in CMOS for connecting transistor terminals. There is a VIA type difference between CMOS and VeSFET on *VIA5*, which connects *M5* (*Mx*) and *M6* (*Mz*). CMOS uses *Vz* as defined in the technology file but VeSFET uses *Vx*. This causes that VeSFET's *VIA5* conditions are pessimistic. The reason is that current VeSFET design rules assume that the metal width within a metal layer is fixed for better regularity. Therefore, the *VIA5* size needs to satisfy *M5*'s 1X *Mx* width.

PDN density (i.e., total power strap area in that metal layer / total area) is critical; denser PDN reduces IR-drop, but it hurts routability. The 2D designs are implemented with the PDN mesh density as listed in Table 3.5. The density and pitch shown in Table 3.5 include both VDD and VSS straps. Thus, for the VDD or the VSS straps, the pitches should be 2X and the densities should be half of the numbers shown in Table 3.5. *M1* straps are the standard cell power rails, which are pre-determined by standard cell layouts and cannot be changed in physical design stages. CMOS has wider and denser *M1* power rails than VeSFET. On the routing layers, power straps are created on *M4 – M7*, and VIA stacks are created to connect *M1* and *M4* straps directly. This is a common design approach for reserving more routing resources on lower metal layers, which is critical for local routing. For better power strap alignments, the pitches of *M4 – M7* are determined in terms of standard cell row height (*CRH*) as illustrated in Figure 3.1. For *M4* and *M5*, the pitches are 1 *CRH*; for *M6* and *M7*, they are

3 *CRH*. The strap densities are 20% for *M4* and *M5* and 40% for *M6* and *M7*; the desired widths are calculated based on the strap pitches and density.

Table 3.5. PDN Mesh Density for 2D Implementations[†1]

Metal Layer[†2]		***M1***	***M4***	***M5***	***M6***	***M7***
CMOS	*Width (μm)*	0.33	0.36	0.36	2.16	2.16
	Pitch (μm)	1.8	1.8	1.8	5.4	5.4
	Density[†3]	18.3%	20%	20%	40%	40%
VeSFET	*Width (μm)*	0.07	1.68	1.68	1.68	1.68
	Pitch (μm)	1.4	1.4	1.4	4.2	4.2
	Density[†3]	5%	20%	20%	40%	40%

[†1]: The PDN includes VDD and VSS meshes. The VDD or VSS PDN density is half of the number shown in the table.

[†2]: Standard cell's power rails are on *M1*. Power straps are created on *M4* to *M7*. VIA stacks are created between *M1* and *M4*.

[†3]: Density = Width / Pitch

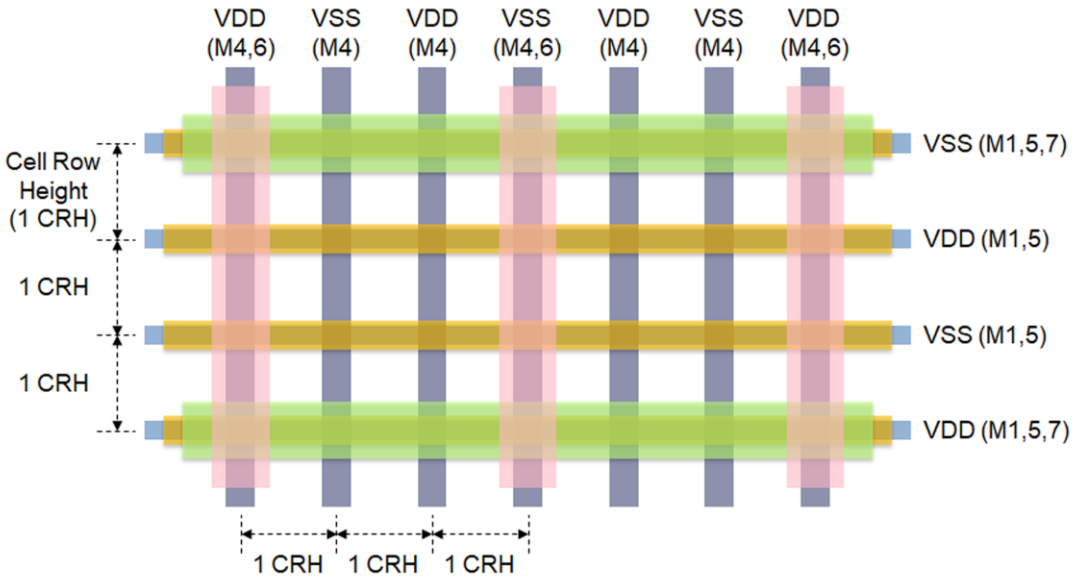


Figure 3.1. Power delivery network (PDN) mesh structure.

3.2.2 2D Implementations

The circuits are first synthesized using *Synopsys Design Compiler* with clock frequency 1GHz, to stress the performance optimization. Design constraints, synthesis commands, and optimization processes are identical in CMOS and VeSFET designs. The results are shown in Table 3.6. Then, *Cadence SoC Encounter* determines the physical implementations. Although VeSFET implementations have better power-delay product (PDP) and lower power consumption than CMOS designs [34], they are slower than CMOS. To perform fair assessments, the clock period T_{clk} is reduced to a reasonable speed in physical design stage. It is set to let the worst negative slack be within a reasonable range; both CMOS and VeSFET designs of a circuit operate under the same clock frequency. The floorplan aspect ratio is set to be 1. The area is set to achieve 70% of cell placement utilization based on the initial synthesized netlist. Both VeSFET and CMOS designs are implemented using the same flow and constraints using *SoC Encounter*'s engine.

Table 3.6. The Synthesis Results[†1]

Name	CMOS			VeSFET		
	# of Cells	**Area (μm^2)**	**WNS [†2] (ns)**	**# of Cells**	**Area (μm^2)**	**WNS [†2] (ns)**
s38584	6943	17987	0	8056	13207	0
s38417	8132	22347	0	8987	16528	-0.47
ac97_ctrl	10218	27971	0.10	10535	17821	0
aes_core	14919	30948	0	19016	28393	-0.58
b17	24149	49002	0	25269	36868	-1.06
b18	63641	121673	-0.73	63002	92000	-3.33
sha3	69077	147028	-0.28	68755	93523	-1.88
des3_perf	68986	158835	0.02	86291	126197	-0.24
viterbi_dec	138011	373050	0	141895	274125	-0.76
fft_256	225833	613602	-0.38	231857	417319	-1.42

[†1]: Synthesized with clock frequency 1GHz.
[†2]: Worst negative slack (WNS)

Detailed 2D implementation results are listed in Table 3.7. In general, VeSFET designs are slower as reported in [34]; however, VeSFET has lower dynamic power (P_{dyn}), leakage power (P_{leak}), and total power (P_{total}), which are critical for 3D ICs. Reference [34] indicates that VeSFET designs consume only 35% dynamic power and 2.6% leakage power of the CMOS design running at the same speed. The smaller power consumption is due to VeSFET's smaller chip area, and transistor's native characteristics of smaller parasitic capacitances and lower leakage current [28-31].

Table 3.7. 2D Implementation Results

CMOS	T_{clk} **(ns)**	**Area**[†1] **(μm²)**	**WNS (ns)**	P_{dyn} **(mW)**	P_{leak} **(μW)**	P_{total} **(mW)**
s38584	1.5	25695	0.271	9.66	42.2	9.70
s38417	2	31924	0.096	9.05	52.9	9.10
ac97_ctrl	1	39958	0.086	16.96	66.2	17.04
aes_core	3	44211	0.882	4.13	72.1	4.20
b17	2.5	70003	0.162	8.02	103.0	8.12
b18	5	173819	1.977	11.70	268.4	11.96
sha3	11	210042	4.912	22.88	273.3	23.15
des3_perf	3	226908	1.684	55.08	376.1	55.46
viterbi_dec	2.5	532928	0.015	97.72	858.3	98.58
fft_256	3	876573	0.239	139.18	1427.0	140.60
VeSFET						
s38584	1.5	18851	0.031	2.48	0.8	2.482
s38417	2	23593	0.064	2.37	1.1	2.371
ac97_ctrl	1	25448	-0.104	3.20	1.3	3.206
aes_core	3	40530	-0.055	2.12	1.5	2.120
b17	2.5	52643	0.004	3.01	2.8	3.008
b18	5	131318	0.141	4.85	5.8	4.857
sha3	11	133585	-0.017	13.07	6.2	13.080
des3_perf	3	184388	0.263	18.58	7.2	18.590
viterbi_dec	2.5	391602	-0.153	22.52	20.1	22.540
fft_256	3	596111	-0.042	27.83	23.5	27.860

[†1]: The floorplan is with aspect ratio = 1 and 70% cell placement utilization

3.2.3 3D Partition Methods

Past works proposed several 3D partition methods addressing various optimization goals. These include thermal-aware local stacking transformation [48], analytical 3D placement [49-50], placement-driven coupled with a modified 2D placement method [51-52], and path- and net-based partition targeting BEOL cost reduction [53].

In this work, since the goal is to compare CMOS and VeSFET 3D design characteristics instead of seeking the optimum design points, we developed a straightforward partition method that is similar to local stacking transformation. A 2D placed and detailed routed design is first divided into *main grids* as shown in Figure 3.2, then each *main grid* is divided into *sub-grids*. Based on different partition methods, each *sub-grid* within the *main grid* is assigned to a different tier, followed by stacking *sub-grids* into a 3D design.

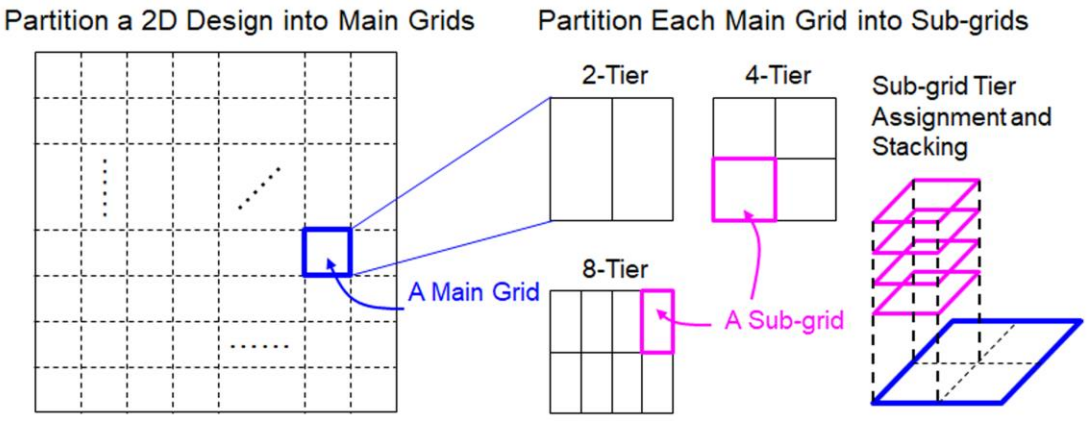


Figure 3.2. 3D partition method, a 2D design is divided into *main grids* and *sub-grids*. Each *sub-grid* is assigned to a tier according to different partition methods. Stack *sub-grids* to construct the partitioned 3D design.

The *main grids* are intended to be of the same size (width = w_m and height = h_m), however, due to the given 2D design area changes, the edge *main grids* may have different size (w_m' and h_m') as illustrated in Figure 3.3. Here, we set a *main grid* resizing condition to prevent

partitioning a grid that is too small. If the edge *main grids* height (h_m') or width (w_m') issmaller than 60% of the regular *main grid*'s height (h_m) or width (w_m), they are merged to the adjacent row or column, respectively. For a more robust 3D PDN, the upper and lower edges of each *sub-grid* row are decided to align standard cell's VDD rail or VSS rail. Thus, the power straps are vertically aligned after 3D stacking.

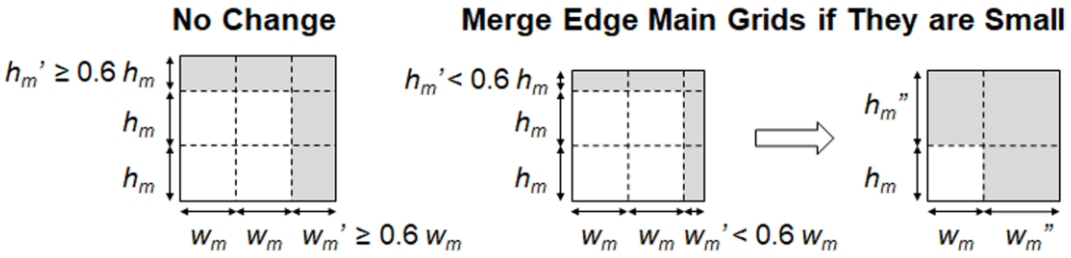


Figure 3.3. Edge *main grid* resizing. Each *main grid* is with the same size except the edge grids. Merge edge grids if they are small.

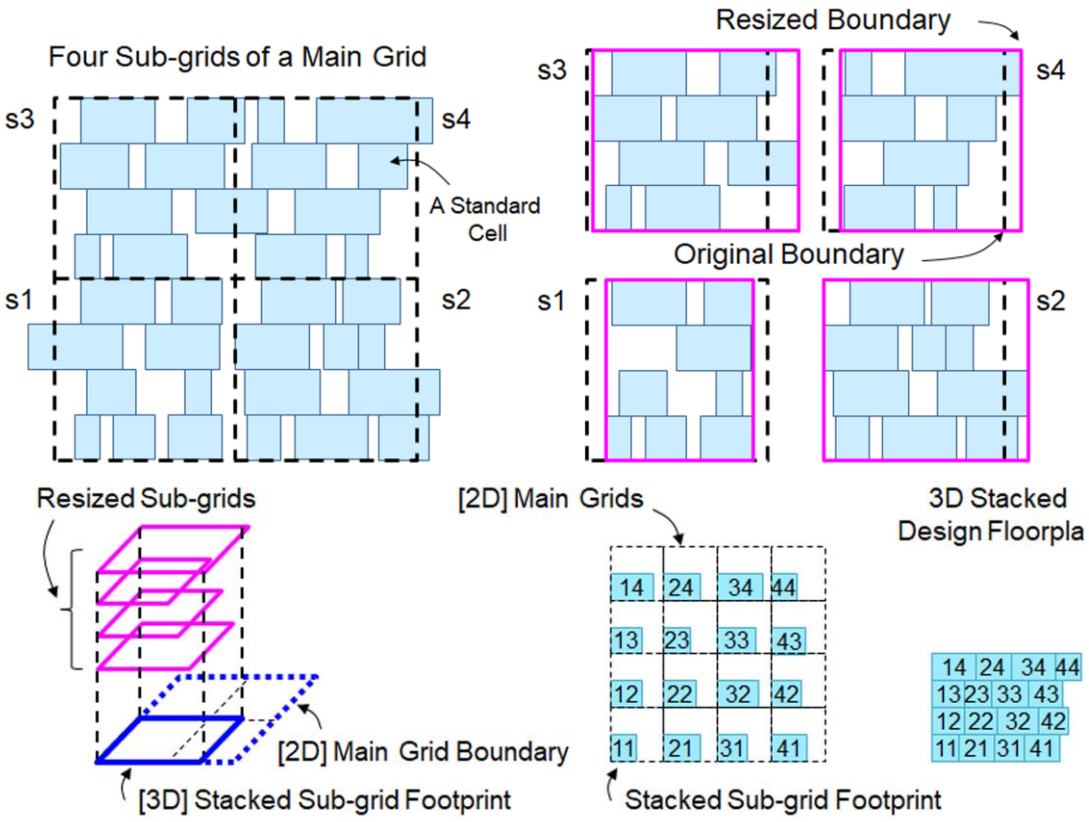


Figure 3.4. *Sub-grid* resizing and 3D stacking. The *sub-grid* width is resized according to the range of the standard cells in this *sub-grid*.

We adjust each *sub-grid's* size to save the footprint of the 3D IC, the method is shown in Figure 3.4 and described below. A standard cell with lower-left corner's coordinate (cX_{LL}, cY_{LL}) belongs to a *sub-grid* bounded by (sX_{LL}, sY_{LL}) (sX_{UR}, sY_{UR}) if: 1) $cX_{LL} \geq sX_{LL}$ and $cX_{LL} < sX_{UR}$, and 2) $cY_{LL} \geq sY_{LL}$ and $cY_{LL} < sY_{UR}$. Each *sub-grid* is resized in *x*-axis direction according to the range of the standard cells. The 3D stacked *sub-grids* footprint is determined by the largest *sub-grid* within this *main grid*. Then, these stacked *sub-grids* are shifted and tightly abutted and form the 3D partitioned design.

The power consumption of each cell after 3D partition is remodeled for better assessment. The total power consumption of each cell consists of three portions: 1) internal power (P_{int}), which is internally consumed by the cell itself; 2) switching power (P_{swt}), which is for charging

and discharging the cell's loadings of the wire and fan-out capacitance; and 3) cell's leakage power (P_{leak}). The wire length changed in the 3D design changes the wire capacitance portion of P_{swt}. The following equations (3.1) and (3.2) are used to estimate the power consumption of each cell:

$$P_{2D} = P_{int} + P_{leak} + P_{swt} \tag{3.1}$$

$$P_{3D} = P_{int} + P_{leak} + P_{swt} \times \left(\frac{C_{pins}}{C_{pins} + C_{wire}} + \frac{D_{3D}}{D_{2D}} \times \frac{C_{wires}}{C_{pins} + C_{wire}} \right) \tag{3.2}$$

Cadence *SoC Encounter* reports each individual cell's power consumption in a 2D design (P_{2D}) and the corresponding three power portions. After 3D partition, P_{int} and P_{leak} portions remain the same in the 3D design power consumption (P_{3D}). Pin capacitance of this cell's fan-outs (C_{pins}) and the wire capacitance (C_{wire}) contribute to P_{swt}. C_{pins} remains the same, but C_{wire} changes. We use the Manhattan distances in 2D and 3D designs (D_{2D} and D_{3D}) calculated by the locations of this cell's output pin and its fan-out cells' input pins to estimate the change of P_{swt} as shown in equation (3.2) and Figure 3.5. The power consumption of each tier is calculated for tier assignment in different partition methods.

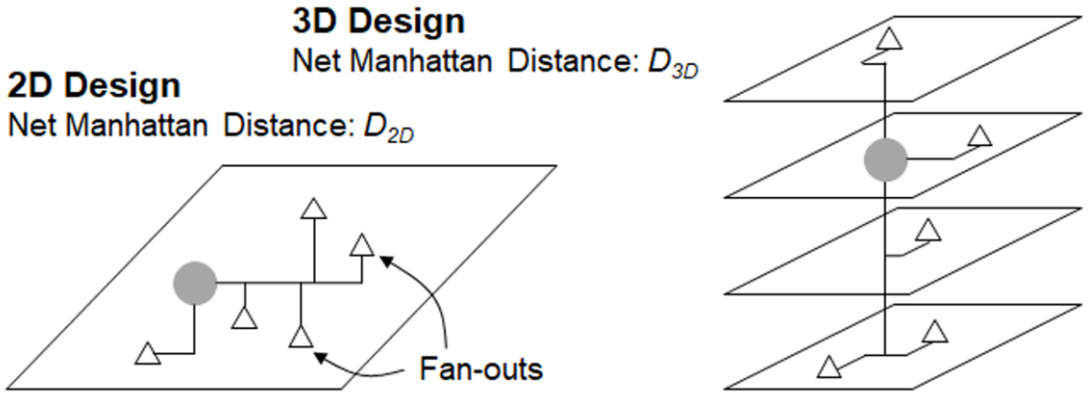


Figure 3.5. Net Manhattan distance in 2D and 3D designs

Power consumption distribution among tiers affect thermal and IR-drop results. Lower IR-drop is expected if the tiers consuming more power are placed closer to bumps; similarly, better thermal property is expected if such tiers are placed closer to heat sinks. We developed four different 3D partition methods to assess such scenarios as shown in Figure 3.6.

1. Uniform (*Uni*): regularly assign tiers by *sub-grid* locations in a *main grid*.
2. Thermal-balanced (*T-bal*): In each *main grid*, assign *sub-grids* to tiers using different orders of power consumption. The assignment order is rotated by a tier in each *main grid* for balancing the thermal generation (i.e., power consumption) of each tier. Figure 3.6 shows a 4-tier assignment example. In *main grid 1* (*m1*), the *sub-grids* are assigned to tiers 1, 2, 3, and 4 in the order of consuming least (green box) to most (red box) power. Then, in the *main grid 2* (*m2*), the assignment order is rotated by a tier; the second least (blue box) to the most power hungry (red box) *sub-grids* are assigned to tiers 1, 2, and 3. The least power hungry *sub-grid* (green box) is now assigned to tier 4. Then, for the next *main grids 3* and *4* (*m3* and *m4*), the orders are rotated by 1 and 2 tiers, respectively.
3. IR-optimized (*IR-opt*): Within each *main grid*, place the *sub-grids* consuming more power closer to the bump.
4. Thermal-optimized (*T-opt*): Like *IR-opt*, but place the *sub-grids* consuming more power closer to the heat sink.

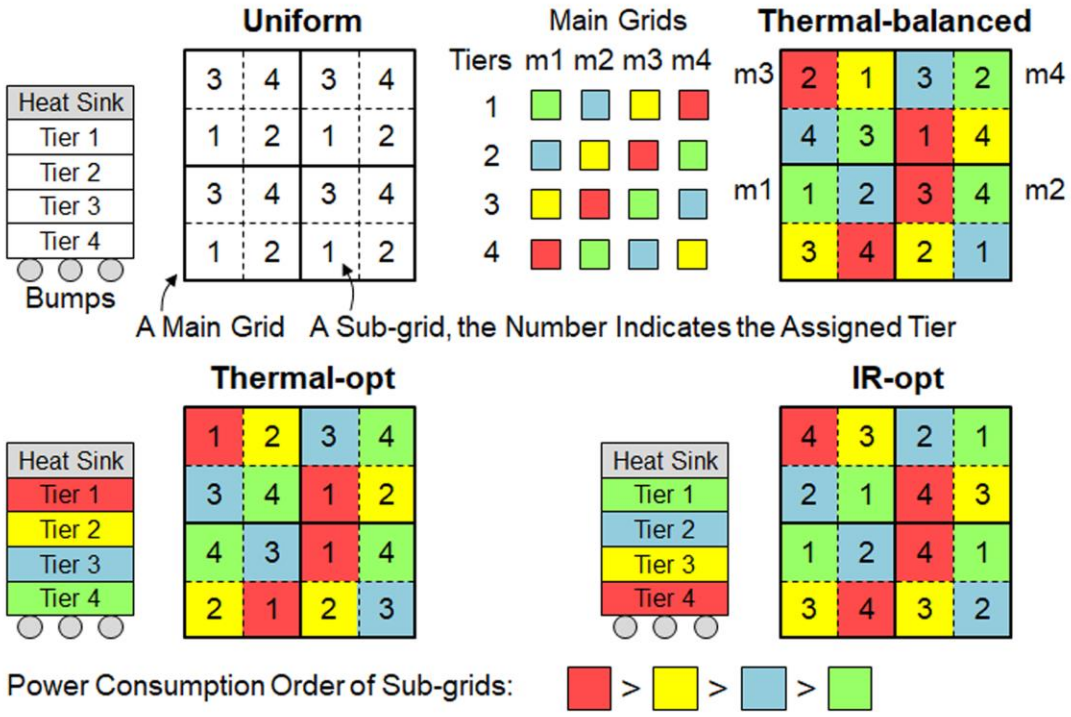


Figure 3.6. Four different 3D partition methods.

Table 3.8 shows circuit *fft_256*'s power consumption of each tier, 4-tier partitioned by different methods. The size of a *main grid* is set to 20 *CRH* × 20 *CRH*.

3.3 Power Delivery Network Assessments

We developed a set of programs for IR-drop simulation since there is no tool dedicated to 3D IC IR-drop analysis and capable of supporting VeSFET technology. The tool takes the PDN structure settings, the 3D partitioned cells placement locations, each cell's average power consumption, and the BEOL metal and VIA properties as the input data. Then, complete 3D IC PDN and current sink models are generated for *Hspice* simulation.

Table 3.8. *fft_256* 4-tier Partition, Power Consumption of Each Tier

CMOS	***Uni***	***T-bal***	***IR-opt***	***T-opt***
Tier 1 (Heat sink)	34092.9	34641.5	27439.3	43726.9
2	34540.5	34455.9	31226.1	36104.6
3	34768.9	34642.4	36104.6	31226.1
4 (Bump)	35094.7	34757.2	43726.9	27439.3
VeSFET				
Tier 1 (Heat sink)	6754.9	6592.8	3383.3	10670.7
2	6367.7	6509.2	5139.0	7201.2
3	6633.6	6628.0	7201.2	5139.0
4 (Bump)	6638.0	6664.3	10670.7	3383.3

Considering the simulation capability, we assessed static IR-drop on the VDD PDN only, i.e., 1) the values of current sinks are based on average power consumption, instead of dynamic current sources changing with time; and 2) the VSS PDN are set to ideal. In reality, the dynamic IR-drop values are larger than static IR-drop at certain times and the non-ideal VSS PDN contributes to IR-drop as well. But for a fair comparison of CMOS and VeSFET designs, the VDD PDN static IR-drop results are representative enough. We use a resistively modeled PDN, which is sufficient for static IR-drop analysis and mitigates the requirement of simulator's capability. We assume each tier has the same PDN structure as used in the 2D implementations described in Section 3.2.2.

3.3.1 Inter-tier Connection Technical Assumptions

A major difference of CMOS and VeSFET 3D integration is the inter-tier connection scheme. CMOS uses MIVs with sizes like regular metal-to-metal VIAs [8]. VeSFET uses the self-embedded pillars with diameter equal to 2*r*. Figure 3.7 and Table 3.9 show the assumed inter-tier connection schemes. In CMOS technology, a MIV connects the bottom metal layer (*Mbot*) on the upper tier and the top metal layer (*Mtop*) on the lower tier. For VeSFET, a pillar

and two contacts (CT) form the connection. The width of CMOS MIV is assumed to be the same as the regular 1X VIA *Vx*, i.e. the width is 100nm. The VeSFET pillar and CT diameter and pillar height *h* (transistor active region) are based on the technology definition, i.e. *h* = 200nm, pillar diameter = 2*r* = 100nm, and CT diameter = 0.7 × 2*r* = 70nm. The heights of inter-layer dielectric (ILD) and buried oxide (BOX) are estimated with referencing the dimensions illustrated in [28] [54-55]. CT heights are assumed to be 100nm. The resistance of MIVs, pillars, and CTs are dimensionally projected from the BEOL parameters listed in Table 3.3. It can be noticed that VeSFET's inter-tier connection is taller and with larger resistance than CMOS. In addition, *Mbot* is *M1* for CMOS and is *M0* for VeSFET. These assumptions are pessimistic for VeSFET.

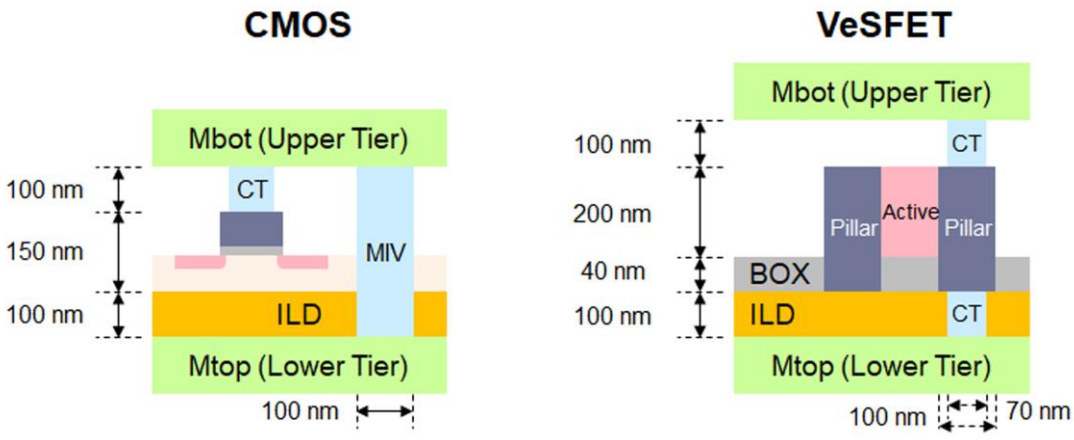


Figure 3.7. Inter-tier connection schemes assumed for IR-drop assessments.

Table 3.9. The Inter-tier Connection Parameters

	CMOS	VeSFET		
VIA Type	**MIV**	**Pillar**	**CT**	**Pillar + 2CT**
Width (nm) [†1]	100	100	70	-
Height (nm)	350	200 + 40	100	440
Rvia (Ω/each) [†2]	3.0	2.6192	2.2272	7.0737

[†1]: CMOS MIV is cuboidal; VeSFET pillar and CT are cylindrical.
[†2]: The resistance is dimensionally projected from regular VIAs

3.3.2 Current Sink Modeling

It is impractical to model each standard cell as an individual current sink at a specific location. This requires a huge number of current sinks and fine node granularity PDN model. We model current sinks using the approach illustrated in Figure 3.8, which greatly reduces the current sink amount yet maintains proper accuracy. The 2D design plane is gridded using the standard cell's power rails (on *M1*) and the vertical power straps on the first layer above cell's power rail (*M4* as shown in Table 3.5). Using the definition shown in Table V, the grid sizes are 1.8μm×1.8μm for CMOS and 1.4μm×1.4μm for VeSFET.

A grid is presented as the blue rectangle shown in Figure 3.8. All cells placed within a grid (determined by the center of the cell) are modeled as two current sinks attached on nodes A and B. The horizontal distances of cell's center to these two nodes determine the ratio modeled by the two current sinks. For example, in Figure 3.8, *Cell 1* and *Cell 2* with the average current I_1 and I_2 are modeled using I_A and I_B which connect to nodes A and B using the equation specified on Figure 3.8.

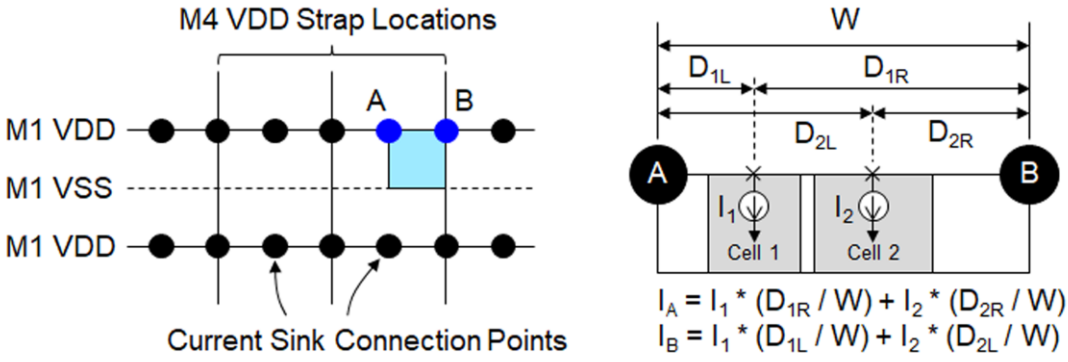


Figure 3.8. Current sink modeling. The current consumed by each cell is modeled and placed on specific connection points for reducing the data size of current sink models.

3.3.3 Inter-tier Connection Modeling

The advantage of VeSFET 3D technology is the evenly and densely distributed pillars, which work naturally as inter-tier signal connections and thermal distribution channels. In VeSFET designs, any pillar can be used for inter-tier connection, no matter if it belongs to a transistor or not. But in CMOS designs, MIVs can be inserted on empty space only. Figure 3.9 and Figure 3.10 show how the inter-tier connections for VDD are modeled. More aggressively, since all transistors have pillars, we can use all transistor terminals that connect to VDD as inter-tier VDD connections. But we don't use such aggressive method in this work because pessimistic cases are assumed for VeSFET designs.

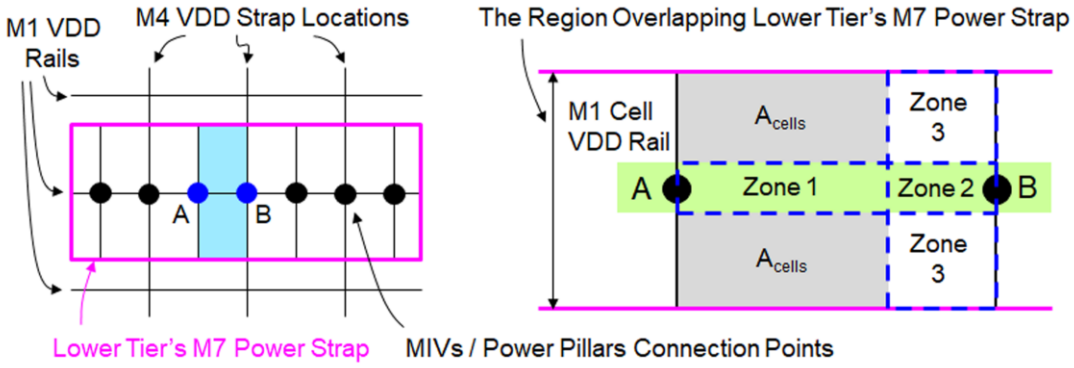


Figure 3.9. Inter-tier VDD connection modeling. CMOS's VDD MIVs can be placed on Zones 2 and 3; VeSFET's VDD pillars can be placed on Zones 1, 2, and 3.

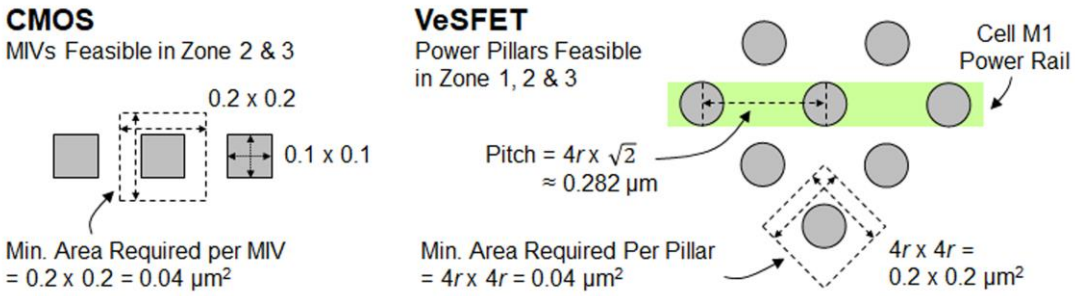


Figure 3.10. The pitches of CMOS MIVs and VeSFET pillars

Similar to the gridding approach for current sink modeling described in Section 3.3.2, the number of VDD MIVs or pillars are computed grid by grid in the regions covered by the lower tier's *M7* power strap as shown in Figure 3.9. The grids are defined in *x*-direction by the locations of vertical *M4* VDD and VSS power straps shown as black vertical bars. In each grid, the VDD MIVs or pillars are connected to the locations of the left and right ends of the upper tier's *M1* VDD rail. For example, the VDD MIVs or pillars in the grid colored in blue shown in Figure 3.9 are connected to the nodes A and B. We compute the total number of VDD MIVs or pillars allowed in this grid, then a half of them are connected to node A and the rests are connected to node B.

With the PDN structure defined in Section 3.2.1 and the 3D partition gridding methods described in Section 3.2.3, the *Mtop* on the lower tier (horizontal *M7*) is aligned to the standard cell power rail on the upper tier. In each grid, the zones feasible to have inter-tier PDN connection are categorized as Zones 1 – 3 as marked in Figure 3.9. Zones 1 and 2 are on the standard cell's power rail, where Zone 1 is the portion covering standard cells and Zone 2 is the portion in the empty region. In CMOS designs, MIVs can be dropped in Zone 2 only, but for VeSFET designs, the pillars in Zones 1 and 2 can be used for inter-tier connections. Those pillars in Zone 1 under standard cell power rail perfectly align to the power rail and provide direct vertical inter-tier PDN connections. Zone 3 is the empty region that is not covered by standard cell's power rail, which is feasible to create inter-tier connections in both CMOS and VeSFET designs.

We calculate the maximum amount of inter-tier VDD MIVs or pillars allowed in the Zones 1 – 3 of each grid. Then, the upper tier's VDD standard cell power rail is directly connected to the vertically aligned location on lower tier's *Mtop* via the inter-tier connection model shown in Figure 3.7 and Table 3.9. Special for VeSFET design, since its *Mbot* is on *M0* and the standard cell power rail is on *M1*, an extra vertical connection from *M0* to *M1* is also considered in the inter-tier connection model. The number of inter-tier PDN connections is modeled by the design rules as shown in Figure 3.10. For CMOS, the minimum area required by an MIV is 0.2 μm × 0.2μm. They can be placed on the empty place only (Zones 2 and 3). For VeSFET, there are two scenarios: 1) power connections can directly use those pillars placed under standard cell's power rails, which are marked as Zone 1 and 2. The pitch of such pillars is $\sqrt{2} \times 4r \approx 0.282$ μm; 2) use the remaining empty space (Zone 3). The minimum area required per pillar is $4r \times 4r = 0.2$ μm × 0.2μm.

3.3.4 Experimental Results

To investigate the effects of reducing PDN density, three PDN density settings are used for IR-drop assessments: 1) 20% for *M4* and *M5*, 40% for *M6* and *M7* (referred to as *PDN-20-40*), as shown in Table 3.5 for 2D implementations; 2) 10% and 20%, respectively (*PDN-10-20*); and 3) 5% and 10%, respectively (*PDN-5-10*). In the three configurations, the power straps widths are kept, and the different densities are obtained by changing strap pitches. Four ideal VDD sources are connected to the two ends of the first and the last *M7* power straps on the top tier that is attached to bumps. The *Hspice* simulation results of ten circuits implemented by CMOS and VeSFET technologies, three different 3D stacked tiers, and four partition methods are listed in Table 3.10 (a) – (c). The results are the maximum static IR-drop measured in the whole 3D stack. For better visualization, Figure 3.11 illustrates the results of all circuits with PDN density setting *PDN-10-20* partitioned into 2 and 8 tiers by four different methods; Figure 3.12 and 3.13 illustrate the results of *s38584* and *fft_256* with PDN density setting *PDN-5-10* partitioned into 2 and 8 tiers by four different methods

Among the four partition methods, Uniform (*Uni*) and Thermal-balanced (*T-bal*) methods have similar IR-drop behavior, because the total power consumption within 3D stack is more evenly distributed among tiers as the *fft_256* circuit results show in Table 3.8. IR-optimized (*IR-opt*), which places the tier consuming more power closer to bumps, has the best IR-drop results as expected. Thermal-optimized (*T-opt*) in contrast, has the worst IR-drop among four partition methods. The PDN resistive network is three-dimensional, and contributes IR-drop to both horizontal and vertical directions.

The IR-drop on horizontal direction is more significant for designs having larger footprint. By partitioning a design into more tiers, the horizontal IR-drop is mitigated, but the

contribution of vertical resistances, i.e., VIAs and inter-tier connections become more significant. Comparing smaller designs (e.g. *s38584*) to larger designs (e.g. *fft_256*), the horizontal IR-drop is less significant, thus their IR-drop increases while they are partitioned into more tiers. But for larger designs which have significant horizontal IR-drop, partitioning them into more tiers reduces the IR-drop if the PDN density on each tier is sufficient. We notice that the IR-drop of *fft_256* increases in 8-Tier partition under PDN density setting *PDN-5-10*, this is because the smaller PDN density increases the significance of horizontal IR-drop.

For the inter-tier connection assumptions stated in Table 3.9, VeSFET has taller inter-tier connection with higher resistance. This makes VeSFET designs more sensitive to the number of tiers. Due to this effect, the VeSFET to CMOS IR-drop ratio increases when the design is partitioned into more tiers. However, this effect reduces by the possibility of using more pillars as shown in Figure 3.9. Among all the cases, VeSFET designs' maximum static IR-drop measured in the 3D stack is 38.5% - 52.3% of CMOS designs.

Table 3.10 (a). CMOS and VeSFET Maximum Static IR-Drop Results, 2-Tier Partition (CMOS / VeSFET, mV) [†1]

	2-Tier			
PDN-20-40	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	0.66 / 0.24	0.65 / 0.22	0.61 / 0.24	0.66 / 0.23
s38417	0.51 / 0.20	0.51 / 0.18	0.50 / 0.18	0.52 / 0.20
ac97_ctrl	0.92 / 0.26	0.91 / 0.27	0.89 / 0.26	0.93 / 0.28
aes_core	0.29 / 0.16	0.30 / 0.16	0.29 / 0.16	0.30 / 0.16
b17	0.51 / 0.22	0.52 / 0.22	0.51 / 0.22	0.52 / 0.22
b18	0.79 / 0.36	0.79 / 0.36	0.78 / 0.35	0.79 / 0.36
sha3	1.23 / 0.77	1.23 / 0.76	1.22 / 0.76	1.23 / 0.77
des3_perf	3.21 / 1.38	3.21 / 1.38	3.17 / 1.38	3.22 / 1.39
viterbi_dec	6.37 / 1.47	6.37 / 1.47	6.36 / 1.47	6.37 / 1.48
fft_256	9.78 / 2.01	9.77 / 2.02	9.76 / 2.02	9.82 / 2.03
Avg. Ratio[†2]	**39.4%**	**39.0%**	**39.4%**	**39.3%**
PDN-10-20				
s38584	1.3 / 0.4	1.3 / 0.4	1.2 / 0.4	1.4 / 0.4
s38417	0.9 / 0.4	0.9 / 0.3	0.9 / 0.3	0.9 / 0.4
ac97_ctrl	1.8 / 0.6	1.8 / 0.6	1.7 / 0.5	1.8 / 0.6
aes_core	0.5 / 0.3	0.5 / 0.3	0.5 / 0.3	0.5 / 0.3
b17	0.8 / 0.4	0.8 / 0.4	0.8 / 0.4	0.8 / 0.4
b18	1.2 / 0.6	1.2 / 0.6	1.2 / 0.5	1.2 / 0.6
sha3	2.2 / 1.2	2.2 / 1.2	2.2 / 1.2	2.2 / 1.2
des3_perf	5.1 / 2.0	5.1 / 2.0	5.0 / 2.0	5.1 / 2.0
viterbi_dec	9.7 / 2.7	9.7 / 2.6	9.6 / 2.6	9.7 / 2.7
fft_256	16.5 / 3.2	16.5 / 3.2	16.5 / 3.2	16.5 / 3.2
Avg. Ratio[†2]	**39.4%**	**39.1%**	**39.5%**	**39.7%**
PDN-5-10				
s38584	2.7 / 0.9	2.5 / 0.9	2.4 / 0.9	2.8 / 0.9
s38417	2.1 / 0.9	2.1 / 0.8	2.0 / 0.8	2.2 / 0.9
ac97_ctrl	4.1 / 1.2	4.0 / 1.2	3.9 / 1.0	4.4 / 1.3
aes_core	1.1 / 0.5	1.1 / 0.5	1.0 / 0.5	1.2 / 0.6
b17	1.6 / 0.8	1.7 / 0.8	1.6 / 0.8	1.8 / 0.8
b18	2.0 / 0.9	2.0 / 0.9	2.0 / 0.8	2.0 / 0.9
sha3	3.3 / 1.9	3.3 / 1.9	3.2 / 1.8	3.3 / 1.9
des3_perf	9.9 / 3.1	9.9 / 3.1	9.6 / 3.0	10.0 / 3.2
viterbi_dec	14.4 / 4.5	14.4 / 4.5	14.2 / 4.5	14.4 / 4.6
fft_256	26.9 / 5.0	26.7 / 5.1	26.7 / 5.1	26.9 / 5.2
Avg. Ratio[†2]	**39.1%**	**38.5%**	**38.5%**	**39.1%**

[†1]: The IR-drop values shown as pairs in this table are the maximum static IR-drop of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS maximum static IR-drop ratio among 10 circuits.

Table 3.10 (b). CMOS and VeSFET Maximum Static IR-Drop Results, 4-Tier Partition (CMOS / VeSFET, mV) [†1]

	4-Tier			
PDN-20-40	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	0.86 / 0.34	0.79 / 0.35	0.72 / 0.28	0.87 / 0.38
s38417	0.65 / 0.27	0.65 / 0.28	0.60 / 0.25	0.70 / 0.31
ac97_ctrl	1.13 / 0.37	1.10 / 0.38	1.04 / 0.31	1.16 / 0.41
aes_core	0.30 / 0.18	0.30 / 0.19	0.27 / 0.16	0.36 / 0.20
b17	0.51 / 0.26	0.51 / 0.25	0.48 / 0.23	0.53 / 0.26
b18	0.73 / 0.34	0.73 / 0.33	0.71 / 0.31	0.74 / 0.34
sha3	0.95 / 0.66	0.95 / 0.66	0.94 / 0.64	0.96 / 0.67
des3_perf	3.39 / 1.24	3.39 / 1.22	3.35 / 1.20	3.43 / 1.26
viterbi_dec	5.57 / 1.20	5.54 / 1.19	5.52 / 1.18	5.58 / 1.22
fft_256	8.17 / 1.65	8.19 / 1.68	8.15 / 1.64	8.20 / 1.70
Avg. Ratio[†2]	41.7%	42.5%	40.6%	42.3%
PDN-10-20				
s38584	2.3 / 0.8	2.2 / 0.9	2.0 / 0.7	2.5 / 1.0
s38417	1.5 / 0.7	1.5 / 0.7	1.4 / 0.6	1.7 / 0.8
ac97_ctrl	2.9 / 1.0	2.8 / 1.0	2.6 / 0.7	3.1 / 1.2
aes_core	0.7 / 0.4	0.7 / 0.4	0.6 / 0.4	0.9 / 0.5
b17	1.0 / 0.6	1.0 / 0.6	0.9 / 0.5	1.1 / 0.6
b18	1.2 / 0.6	1.2 / 0.6	1.1 / 0.5	1.2 / 0.6
sha3	1.9 / 1.2	1.9 / 1.2	1.8 / 1.2	1.9 / 1.3
des3_perf	6.0 / 2.1	6.0 / 2.1	5.9 / 2.0	6.2 / 2.1
viterbi_dec	8.9 / 2.3	8.8 / 2.3	8.7 / 2.2	9.0 / 2.4
fft_256	13.8 / 2.7	13.9 / 2.7	13.8 / 2.6	14.0 / 2.8
Avg. Ratio[†2]	42.3%	43.0%	40.7%	43.7%
PDN-5-10				
s38584	7.6 / 2.3	7.5 / 2.4	6.5 / 1.9	8.3 / 2.7
s38417	4.8 / 2.1	4.7 / 2.1	4.0 / 1.7	5.3 / 2.6
ac97_ctrl	9.0 / 2.8	8.5 / 2.9	7.7 / 2.1	9.6 / 3.5
aes_core	2.0 / 1.0	2.0 / 1.1	1.5 / 0.9	2.6 / 1.3
b17	2.7 / 1.6	2.7 / 1.6	2.4 / 1.3	3.0 / 1.8
b18	2.7 / 1.3	2.6 / 1.3	2.4 / 1.1	2.8 / 1.5
sha3	3.5 / 2.6	3.5 / 2.6	3.4 / 2.4	3.7 / 2.8
des3_perf	10.5 / 4.2	10.4 / 4.2	9.9 / 3.8	11.0 / 4.5
viterbi_dec	13.8 / 4.4	13.6 / 4.4	13.2 / 4.0	14.1 / 4.7
fft_256	23.9 / 4.6	23.9 / 4.7	23.4 / 4.2	24.2 / 5.0
Avg. Ratio[†2]	42.9%	43.7%	41.6%	45.0%

[†1]: The IR-drop values shown as pairs in this table are the maximum static IR-drop of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS maximum static IR-drop ratio among 10 circuits.

Table 3.10 (c). CMOS and VeSFET Maximum Static IR-Drop Results, 8-Tier Partition (CMOS / VeSFET, mV) [†1]

	8-Tier			
PDN-20-40	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	1.59 / 0.68	1.57 / 0.72	1.40 / 0.54	1.83 / 0.86
s38417	1.24 / 0.60	1.22 / 0.60	1.04 / 0.48	1.38 / 0.75
ac97_ctrl	2.15 / 0.76	2.09 / 0.74	1.85 / 0.46	2.39 / 0.99
aes_core	0.52 / 0.33	0.55 / 0.34	0.41 / 0.26	0.65 / 0.39
b17	0.75 / 0.43	0.74 / 0.44	0.65 / 0.34	0.83 / 0.52
b18	0.79 / 0.40	0.79 / 0.39	0.73 / 0.34	0.84 / 0.44
sha3	1.24 / 0.89	1.24 / 0.89	1.20 / 0.83	1.27 / 0.94
des3_perf	3.36 / 1.36	3.34 / 1.36	3.21 / 1.25	3.49 / 1.46
viterbi_dec	6.12 / 1.47	6.10 / 1.48	5.97 / 1.34	6.21 / 1.57
fft_256	7.57 / 1.79	7.59 / 1.83	7.46 / 1.69	7.67 / 1.91
Avg. Ratio[†2]	45.9%	46.2%	42.6%	48.4%
PDN-10-20				
s38584	5.8 / 2.2	5.8 / 2.3	5.0 / 1.8	6.7 / 2.8
s38417	3.8 / 2.1	3.7 / 2.2	3.0 / 1.6	4.3 / 2.7
ac97_ctrl	6.9 / 2.4	6.7 / 2.5	5.7 / 1.4	7.9 / 3.4
aes_core	1.6 / 1.0	1.6 / 1.0	1.1 / 0.8	2.0 / 1.2
b17	2.0 / 1.3	2.0 / 1.3	1.7 / 0.9	2.4 / 1.6
b18	1.9 / 1.0	1.9 / 1.0	1.6 / 0.8	2.1 / 1.2
sha3	2.5 / 2.3	2.5 / 2.3	2.4 / 2.1	2.7 / 2.5
des3_perf	7.7 / 3.2	7.6 / 3.1	7.1 / 2.8	8.2 / 3.5
viterbi_dec	11.8 / 2.7	11.7 / 2.7	11.3 / 2.3	12.2 / 3.1
fft_256	12.6 / 3.5	12.6 / 3.5	12.2 / 3.1	13.0 / 3.9
Avg. Ratio[†2]	49.2%	49.8%	46.0%	52.3%
PDN-5-10				
s38584	20.5 / 6.5	20.2 / 6.6	17.1 / 5.1	23.6 / 8.3
s38417	13.6 / 6.1	13.1 / 6.2	10.8 / 4.6	15.7 / 7.9
ac97_ctrl	24.5 / 8.4	23.8 / 8.4	19.9 / 4.9	28.3 / 11.9
aes_core	5.3 / 3.6	5.4 / 3.6	3.6 / 2.9	6.9 / 4.5
b17	6.2 / 4.1	6.1 / 4.2	5.0 / 3.0	7.4 / 5.2
b18	5.2 / 3.2	5.2 / 3.3	4.3 / 2.5	6.2 / 4.0
sha3	7.4 / 6.4	7.4 / 6.3	6.8 / 5.5	8.0 / 7.2
des3_perf	22.3 / 8.9	22.1 / 8.8	19.5 / 7.6	24.7 / 10.2
viterbi_dec	25.0 / 6.4	24.9 / 6.4	22.9 / 4.8	26.9 / 7.9
fft_256	28.0 / 7.1	28.0 / 7.2	26.3 / 5.7	29.8 / 8.5
Avg. Ratio[†2]	48.7%	49.0%	45.6%	51.6%

[†1]: The IR-drop values shown as pairs in this table are the maximum static IR-drop of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS maximum static IR-drop ratio among 10 circuits.

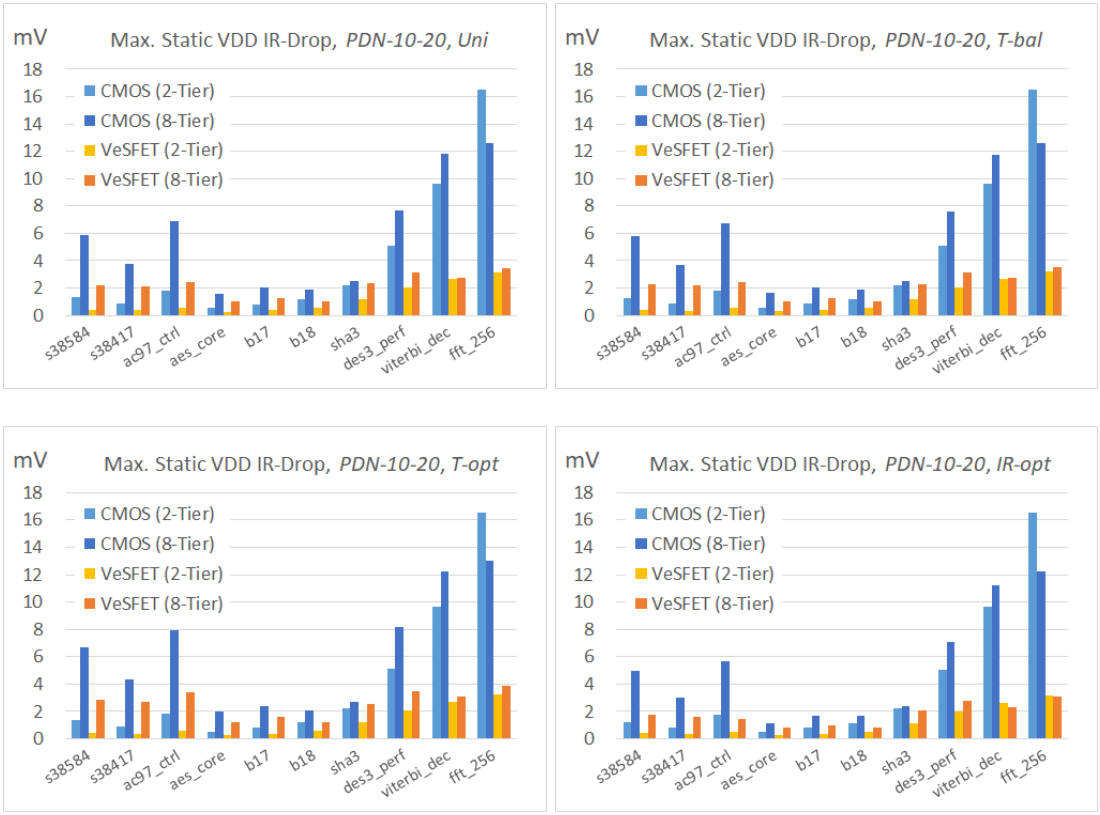


Figure 3.11. Max. static VDD IR-Drop, *PDN-10-20*, partitioned into 2 and 8 Tiers with four different partition methods. On average among all circuits in 2-Tier and 8-Tier, VeSFET to CMOS ratios are 44.3%, 44.5%, 46.0%, and 42.7% (*Uni*, *T-bal*, *T-opt*, and *IR-opt*)

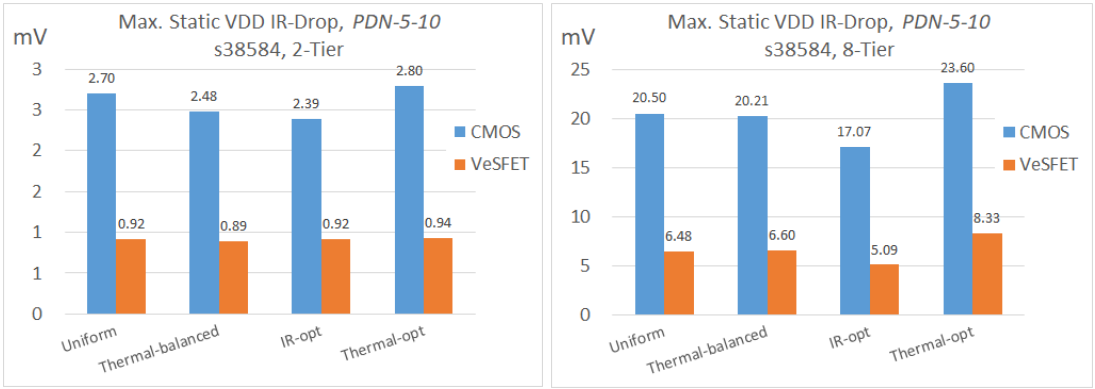


Figure 3.12. Max. static VDD IR-Drop, *PDN-5-10*, *s38584*, partitioned into 2 and 8 Tiers with four different partition methods.

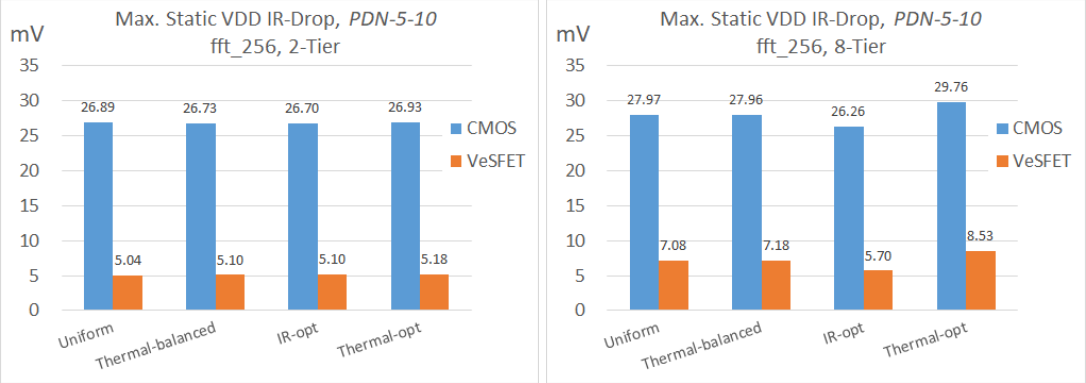


Figure 3.13. Max. static VDD IR-Drop, *PDN-5-10*, *fft_256*, partitioned into 2 and 8 Tiers with four different partition methods.

3.4 Clock Distribution Network Assessments

CDN contributes to a significant portion of overall dynamic power consumption of a chip. Clock skew is critical for timing. Since there is no true 3D clock distribution tool available, we use a 2D place and route tool to build the 3D CDN.

3.4.1 Clock Distribution Network Structure

The CDN network is assessed using the structure shown in Figure 3.14. Each tier has a clock tree; the locations of the sources of all trees are vertically aligned. Each tree has the maximum and minimum delay from the clock source to the corresponding Flip-Flops (Max. and Min. $T_{CS\text{-}FF}$) and the intra-tier skew $T_{skew\text{-}tier}$ = (Max. $T_{CS\text{-}FF}$ - Min. $T_{CS\text{-}FF}$). If the clock sources of all trees are timing aligned, the overall clock skew in this stack ($T_{skew\text{-}stack}$) is the maximum of all Max. $T_{CS\text{-}FF}$ among tiers minus the minimum of all Min. $T_{CS\text{-}FF}$ among tiers. However, $T_{skew\text{-}stack}$ can be improved by adjusting the clock source delay of tiers. The overall clock skew in this stack ($T_{skew\text{-}stack}$) equals to the maximum of the intra-tier skew ($T_{skew\text{-}tier}$) among tiers. Here, we assume the 3D CDN is implemented using this approach.

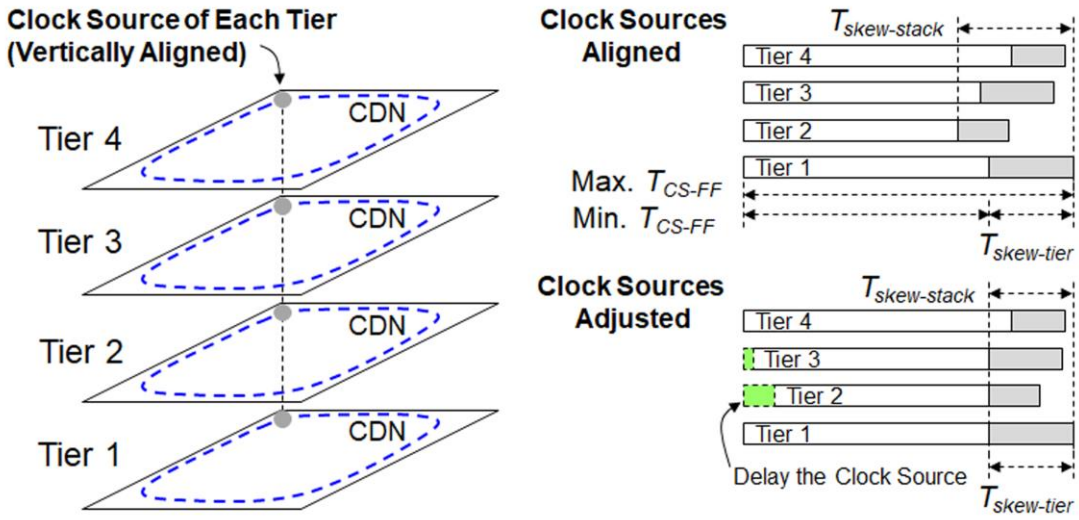


Figure 3.14. Clock distribution scheme assumption.

3.4.2 Experimental Results

The assessment starts from 3D partitioning of the 2D placed designs. The 2D designs are fully placed and ready to perform clock tree synthesis (CTS). *Cadence SoC Encounter* processes the partitioned designs tier by tier, with the specified 3D partitioned floorplan, the same PDN mesh as already explained in Section 3.2.1, and the 3D partitioned cell location. Then, *SoC Encounter* performs CTS and reports the CDN performance in each tier. The clock frequency is set to be the same as in the original 2D implementations; the maximum tolerable clock transition time is 100ps. All inverters and buffers in the standard cell libraries are allowed for CTS creation and optimization processes. The number of Flip-Flops and the clock frequency of each design are listed in Table 3.1 and Table 3.7. The same design flows, constraints, commands, and optimizations are applied to both CMOS and VeSFET designs in *SoC Encounter*.

The detailed results of total power consumption of all tiers (P_{tot}, in mW), overall clock skew ($T_{skew\text{-}stack}$, in ps) of the whole 3D stack, and the total number of buffers used in all clock trees are shown in Table 3.11 (a) – (c). For better visualization, Figure 3.15 illustrates the results of all circuits partitioned into 2 and 8 tiers by *Thermal-balanced* (*T-bal*) method.

The overall clock skew ($T_{skew\text{-}stack}$) is the maximum of intra-tier clock skew ($T_{skew\text{-}tier}$) measured among all tiers. On average among ten circuits and twelve 3D stacking schemes, the ratios of VeSFET to CMOS are: 1) P_{tot}: 70.6% - 73.7%, 72.4% on average; 2) $T_{skew\text{-}stack}$: 120.3% - 194.9%, 151.6% on average; and 3) number of buffers: 191.5% - 235.8%, 208.8% on average.

Table 3.11 (a). CMOS and VeSFET Clock Distribution Network (CDN) Characteristics, 2-Tier Partition (CMOS / VeSFET) [†1]

	2-Tier			
P_{tot} (mW)	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	0.8 / 0.7	0.9 / 0.7	0.9 / 0.7	0.9 / 0.7
s38417	1.2 / 0.6	0.8 / 0.6	0.8 / 0.6	0.8 / 0.6
ac97_ctrl	2.2 / 1.4	2.8 / 1.4	2.3 / 1.4	2.3 / 1.4
aes_core	0.2 / 0.2	0.2 / 0.2	0.2 / 0.2	0.2 / 0.2
b17	0.9 / 0.5	0.9 / 0.6	0.9 / 0.5	0.9 / 0.5
b18	0.8 / 0.6	0.7 / 0.6	0.8 / 0.6	0.8 / 0.6
sha3	0.3 / 0.3	0.3 / 0.3	0.3 / 0.3	0.3 / 0.3
des3_perf	3.3 / 2.4	3.3 / 2.5	3.3 / 2.4	3.3 / 2.4
viterbi_dec	11.8 / 8.9	11.7 / 8.7	11.8 / 8.5	11.8 / 8.5
fft_256	15.6 / 11.9	18.3 / 11.7	15.9 / 11.0	15.9 / 11.0
Avg. Ratio[†2]	73.6%	73.2%	71.6%	71.6%
$T_{skew\text{-}stack}$ (ps)				
s38584	7.9 / 4.8	6.2 / 5.2	7.0 / 4.2	7.0 / 4.2
s38417	6.0 / 6.8	9.5 / 9.4	9.3 / 6.7	9.3 / 6.7
ac97_ctrl	5.7 / 6.4	5.0 / 6.5	5.5 / 6.9	5.5 / 6.9
aes_core	4.7 / 3.8	6.1 / 5.1	6.6 / 6.5	6.6 / 6.5
b17	5.9 / 11.7	5.6 / 7.1	5.2 / 8.7	5.2 / 8.7
b18	9.7 / 17.4	9.3 / 23.8	10.3 / 22.5	10.3 / 22.5
sha3	13.0 / 15.9	11.0 / 19.2	13.6 / 24.3	13.6 / 24.3
des3_perf	15.3 / 24.1	17.8 / 23.5	13.6 / 28.7	13.6 / 28.7
viterbi_dec	11.2 / 37.3	13.7 / 62.7	11.2 / 40.3	11.2 / 40.3
fft_256	17.4 / 65.5	15.1 / 61.2	18.3 / 41.8	18.3 / 41.8
Avg. Ratio[†2]	173.4%	194.9%	172.0%	172.0%
# of Buffers				
s38584	32 / 99	47 / 91	48 / 82	48 / 82
s38417	60 / 99	45 / 102	32 / 97	32 / 97
ac97_ctrl	67 / 102	62 / 102	67 / 102	67 / 102
aes_core	29 / 51	14 / 55	20 / 47	20 / 47
b17	45 / 97	48 / 104	50 / 103	50 / 103
b18	92 / 196	91 / 202	90 / 208	90 / 208
sha3	79 / 197	73 / 203	76 / 184	76 / 184
des3_perf	214 / 478	216 / 517	219 / 484	219 / 484
viterbi_dec	660 / 1468	648 / 1353	662 / 1354	662 / 1354
fft_256	1018 / 2302	1013 / 2180	1066 / 1947	1066 / 1947
Avg. Ratio[†2]	215.2%	235.8%	214.9%	214.9%

[†1]: The values of total power (P_{tot}), overall clock skew ($T_{skew\text{-}stack}$ = Max. $T_{skew\text{-}tier}$), and the number of buffers shown as pairs are of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS ratio among 10 circuits.

Table 3.11 (b). CMOS and VeSFET Clock Distribution Network (CDN) Characteristics, 4-Tier Partition (CMOS / VeSFET) [†1]

	4-Tier			
P_{tot} (mW)	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	0.9 / 0.7	0.9 / 0.7	0.9 / 0.6	0.9 / 0.6
s38417	0.9 / 0.6	0.8 / 0.6	0.8 / 0.6	0.8 / 0.6
ac97_ctrl	2.2 / 1.6	2.1 / 1.6	2.2 / 1.6	2.2 / 1.6
aes_core	0.2 / 0.2	0.3 / 0.2	0.3 / 0.2	0.3 / 0.2
b17	0.7 / 0.6	0.7 / 0.6	0.7 / 0.6	0.7 / 0.6
b18	0.9 / 0.6	0.9 / 0.6	0.9 / 0.6	0.9 / 0.6
sha3	0.5 / 0.3	0.4 / 0.3	0.4 / 0.3	0.4 / 0.3
des3_perf	3.3 / 2.5	3.4 / 2.5	3.4 / 2.5	3.4 / 2.5
viterbi_dec	11.8 / 9.2	14.0 / 9.0	11.8 / 9.3	11.8 / 9.3
fft_256	17.2 / 11.7	16.0 / 11.5	16.8 / 11.7	16.8 / 11.7
Avg. Ratio[†2]	73.7%	73.0%	73.2%	73.2%
$T_{skew\text{-}stack}$ (ps)				
s38584	2.7 / 4.7	4.0 / 4.4	4.1 / 3.8	4.1 / 3.8
s38417	7.0 / 6.7	5.3 / 5.3	6.0 / 6.2	6.0 / 6.2
ac97_ctrl	8.1 / 5.0	4.8 / 4.7	10.3 / 4.1	10.3 / 4.1
aes_core	4.0 / 3.3	3.6 / 2.7	3.7 / 3.0	3.7 / 3.0
b17	7.8 / 10.3	6.6 / 7.6	6.6 / 10.6	6.6 / 10.6
b18	8.3 / 9.9	5.7 / 10.8	8.8 / 12.8	8.8 / 12.8
sha3	6.8 / 10.8	10.6 / 13.9	8.5 / 11.3	8.5 / 11.3
des3_perf	8.8 / 20.9	8.7 / 28.7	11.7 / 19.6	11.7 / 19.6
viterbi_dec	16.8 / 37.8	13.9 / 29.1	17.3 / 28.3	17.3 / 28.3
fft_256	15.1 / 64.0	17.2 / 36.0	13.3 / 37.7	13.3 / 37.7
Avg. Ratio[†2]	171.1%	156.7%	137.0%	137.0%
# of Buffers				
s38584	62 / 99	67 / 97	48 / 86	48 / 86
s38417	65 / 115	71 / 124	58 / 118	58 / 118
ac97_ctrl	53 / 145	83 / 148	73 / 153	73 / 153
aes_core	26 / 70	21 / 50	25 / 63	25 / 63
b17	62 / 142	51 / 128	67 / 127	67 / 127
b18	104 / 218	108 / 213	113 / 219	113 / 219
sha3	124 / 257	96 / 209	115 / 221	115 / 221
des3_perf	239 / 492	246 / 523	241 / 493	241 / 493
viterbi_dec	638 / 1605	659 / 1555	638 / 1627	638 / 1627
fft_256	1026 / 2238	1056 / 2214	1042 / 2316	1042 / 2316
Avg. Ratio[†2]	220.1%	206.0%	210.2%	210.2%

[†1]: The values of total power (P_{tot}), overall clock skew ($T_{skew\text{-}stack}$ = Max. $T_{skew\text{-}tier}$), and the number of buffers shown as pairs are of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS ratio among 10 circuits.

Table 3.11 (c). CMOS and VeSFET Clock Distribution Network (CDN) Characteristics, 8-Tier Partition (CMOS / VeSFET) [†1]

	8-Tier			
P_{tot} (mW)	***Uni***	***T-bal***	***IR-opt***	***T-opt***
s38584	1.1 / 0.7	1.0 / 0.7	1.0 / 0.7	1.0 / 0.7
s38417	0.9 / 0.6	1.0 / 0.6	1.0 / 0.6	1.0 / 0.6
ac97_ctrl	2.2 / 1.6	2.2 / 1.6	2.2 / 1.6	2.2 / 1.6
aes_core	0.3 / 0.2	0.3 / 0.2	0.3 / 0.2	0.3 / 0.2
b17	0.8 / 0.6	0.8 / 0.6	0.8 / 0.6	0.8 / 0.6
b18	1.0 / 0.7	0.9 / 0.7	1.1 / 0.7	1.1 / 0.7
sha3	0.4 / 0.3	0.5 / 0.3	0.3 / 0.3	0.3 / 0.3
des3_perf	3.8 / 2.6	3.7 / 2.6	3.6 / 2.6	3.6 / 2.6
viterbi_dec	12.1 / 9.6	12.3 / 9.6	12.2 / 9.3	12.2 / 9.3
fft_256	16.4 / 12.7	17.2 / 12.6	16.4 / 12.6	16.4 / 12.6
Avg. Ratio[†2]	71.4%	70.6%	72.1%	72.1%
$T_{skew\text{-}stack}$ (ps)				
s38584	4.0 / 2.5	5.1 / 2.7	6.2 / 3.4	6.2 / 3.4
s38417	5.1 / 3.0	3.0 / 2.9	4.5 / 3.5	4.5 / 3.5
ac97_ctrl	5.1 / 5.3	5.2 / 4.4	4.8 / 5.6	4.8 / 5.6
aes_core	2.6 / 2.6	2.2 / 2.1	4.0 / 1.8	4.0 / 1.8
b17	4.5 / 4.1	7.5 / 4.6	4.9 / 6.1	4.9 / 6.1
b18	8.2 / 6.0	9.9 / 5.7	8.5 / 4.8	8.5 / 4.8
sha3	12.4 / 10.0	13.0 / 10.8	10.4 / 10.9	10.4 / 10.9
des3_perf	8.2 / 12.1	8.9 / 13.9	7.9 / 19.3	7.9 / 19.3
viterbi_dec	10.7 / 31.9	10.0 / 35.5	12.2 / 31.5	12.2 / 31.5
fft_256	15.6 / 36.4	22.2 / 35.6	15.3 / 33.5	15.3 / 33.5
Avg. Ratio[†2]	124.9%	120.3%	130.2%	130.2%
# of Buffers				
s38584	62 / 105	68 / 106	61 / 125	61 / 125
s38417	80 / 131	74 / 132	78 / 123	78 / 123
ac97_ctrl	89 / 156	86 / 161	106 / 158	106 / 158
aes_core	28 / 55	53 / 63	57 / 67	57 / 67
b17	77 / 145	69 / 137	84 / 155	84 / 155
b18	150 / 260	137 / 287	147 / 306	147 / 306
sha3	106 / 269	138 / 254	110 / 252	110 / 252
des3_perf	270 / 551	304 / 567	303 / 551	303 / 551
viterbi_dec	684 / 1684	687 / 1666	682 / 1584	682 / 1584
fft_256	1023 / 2677	1043 / 2643	1031 / 2607	1031 / 2607
Avg. Ratio[†2]	203.2%	191.5%	191.8%	191.8%

[†1]: The values of total power (P_{tot}), overall clock skew ($T_{skew\text{-}stack}$ = Max. $T_{skew\text{-}tier}$), and the number of buffers shown as pairs are of the CMOS / VeSFET designs.

[†2]: It is the average VeSFET to CMOS ratio among 10 circuits.

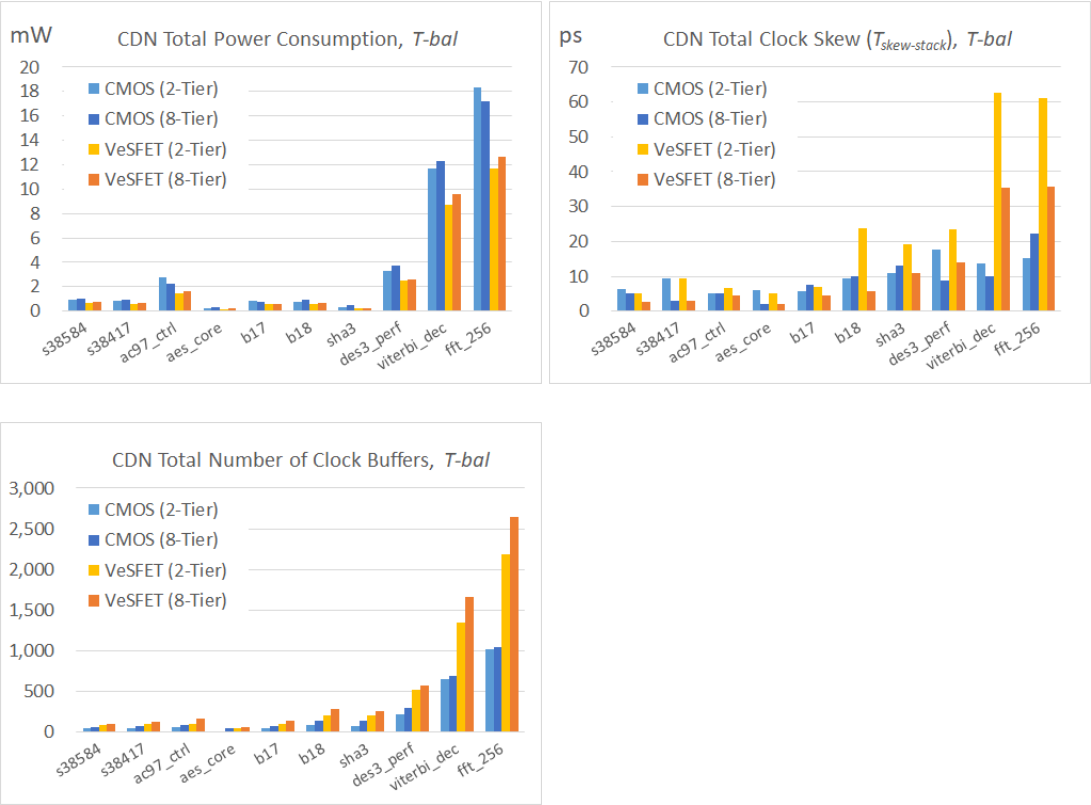


Figure 3.15. CDN characteristics of all circuits partitioned into 2 and 8 Tiers with *Thermal-balanced* (*T-bal*) method. On average of all circuits in 2-Tier and 8-Tier with *T-bal* method, VeSFET to CMOS ratios are 71.9%, 157.6%, and 213.6% (Total power consumption, clock skew, and number of clock buffers)

Similar to the 2D implementation results, VeSFET's CDN power consumption is less than CMOS designs. But due to VeSFET's weaker driving current in this untuned device model, it requires more buffers to distribute clock to all Flip-Flops and results in worse clock skew. Although the clock skew is still within an acceptable range considering the clock period, further optimizations of VeSFET device are suggested for better CDN performance.

3.5 Summary

The increased power density and the inter-tier connection area overhead are key concerns for 3D integration, they hinder the future growth and are potential showstoppers for 3D ICs.

A novel transistor VeSFET offers attractive characteristics for addressing these concerns. It features lower power consumption and zero inter-tier connection area overheads. This chapter presents the differences of VeSFET and CMOS monolithic 3D integration methods, as a path finding research for future 3D ICs. We developed four different 3D partition approaches and physical modeling methodologies, which cover different 3D IC behaviors for the thorough assessments. Two key design concerns are assessed in this chapter:

1. PDN IR-drop, which strongly determines device timing and functional behaviors. It must be well controlled to guarantee the IC can operate correctly. The margin is very limited in advanced technology nodes using low VDD.
2. CDN characteristics, which contribute a significant portion of overall dynamic power consumption and strongly affect timing performance.

Ten circuits, three 3D stacking schemes (2-, 4-, and 8-Tiers), four different partition methods (Uniform, Thermal-balanced, IR-optimized, and Thermal-optimized), and different PDN density are assessed. On average over all cases, the ratios of VeSFET designs to CMOS designs are: 1) the maximum static PDN IR-drop is 38.5% - 52.3% and 2) the CDN's power consumption is 70.6% - 73.7%, but the clock skew is 120.3% - 194.9% due to the weaker driving current in this untuned device model.

VeSFET technology is assessed under pessimistic conditions with the untuned device, circuits, and layout styles. Even under such conditions, VeSFET still shows promising potential. Its unique characteristics fit well for addressing the key concerns of power consumption, thermal, and inter-tier interconnects, which exist in current 3D integration technologies. Better results could be expected with further optimizations. This novel transistor offers new opportunities for future 3D IC technology development and applications.

Chapter 4

A FAST, FULLY VERIFIABLE, AND HARDWARE PREDICTABLE ASIC DESIGN METHODOLOGY USING 3D FPGAS

In this chapter, a fast, fully verifiable, and hardware predictable ASIC design methodology is proposed and demonstrated for the Vertical Slit Field Effect Transistor (VeSFET) based integrated circuits. The key enablers of this methodology are the unique and powerful capabilities of pillar-based transistor arrays with two-side accessible terminals and monolithic 3D integration. VeSFET is a successfully fabricated transistor of this kind. In the proposed methodology, the circuit is first designed as a 3D FPGA using a conventional FPGA design flow. With a little extra Back End of Line (BEOL) masking cost, the design implemented on the 3D FPGA is migrated to the final 2D ASIC, which has exactly the same performance as the 3D FPGA and the verification tasks performed on the 3D FPGA remain valid for the final 2D ASIC.

4.1 Introduction

Long design cycle, high design costs, difficulties of predicting performance and verifying the final hardware at the design stage, and the costs of hardware revisions are all critical concerns of application-specific integrated circuits (ASICs) designs. Process variation,

fabrication defects, model accuracy, and EDA tools' precision all increase unpredictability of the final hardware. The final ASIC chip cannot be fully verified and debugged at the design stage due to un-modeled factors caused by the impractically long computing time and limited resources. Software-based verification by EDA tools is not feasible either. Design revisions and re-spins are usually necessary after the first hardware sample is fabricated and tested, which causes extra turn-around costs in terms of time and money. As the technology advances, design and verification efforts, design and fabrication cycle, non-recurring engineering (NRE) and turn-around costs, design-for-testability (DFT) features, fabrication unpredictability, the complexity of modeling and simulation, and the difficulties of mitigating the gaps of pre- and post-silicon results are all expected to grow. These costs have become the major bottlenecks for nanometer-scale ASIC designs.

To break the limitations of software-based verification at the pre-silicon stage, hardware-based verification techniques using emulators and field programmable gate arrays (FPGAs) have been developed. Emulators such as Mentor Graphics Veloce Emulation Platform [56] compiles the design and emulates it on a specially designed machine. Emulation is a much faster process than running simulations and all the signals are visible. FPGAs have been widely used for functional verification and design prototyping. However, although emulators or FPGAs could perform hardware-based verifications, the layout of the verified design is very different from the final ASIC circuit layout and in advanced technologies many electrical behaviors are layout dependent.

Compared to conventional ASIC design flows, FPGAs' programmability offers many advantages, such as: 1) simpler design flow and shorter design cycle, 2) shorter turn-around time, 3) hardware-based logic design optimization and functional verification, 4) faster system

and software development, 5) shorter time-to-market, and 6) easier product upgrade and revision by providing a new bit-stream. In [57], the system development time of FPGA design flow is reported to be 9 to 12 months shorter than ASIC flow. However, these benefits come through a huge programming overhead needed for controlling logic and routing. In a previous study [16], it was found that 78% of the FPGA total area is occupied by routing resources and 86% is occupied by all routing resources and programming memory. An intuitive idea to achieve smaller chip area and better performance is using 3D integration. Monolithic 3D integrates transistor tiers on the same substrate, which provides abundant vertical connections.

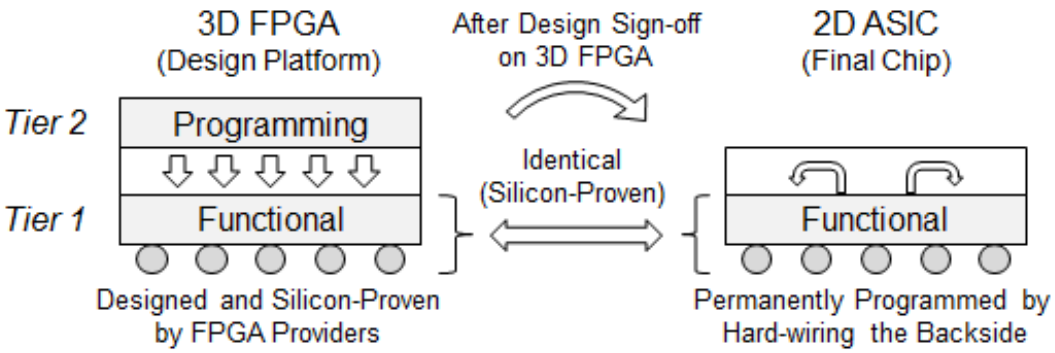


Figure 4.1. The key idea of the proposed ASIC design methodology

In this chapter we propose an ASIC design methodology based on the FPGA design flow to produce a fully verified hardware, to mitigate the end-hardware unpredictability, and to minimize the design gaps between the pre- and post-silicon stages. Figure 4.1 illustrates the key idea. A 3D monolithic FPGA with a smaller footprint and better performance than a 2D FPGA is used as an intermediate implementation platform. Its *Tier 1* hosts all the configurable functional elements and interconnects. *Tier 2* contains all the programming units connected to *Tier 1* devices by the inter-tier interconnects. This 3D FPGA is designed and fully verified by its provider. ASIC designers use this silicon-proven 3D FPGA platform to implement the

circuit by conventional FPGA design flow; all the advantages of using FPGAs are now available. All the design tasks including logic optimization, verification, performance prediction, etc. are based on real hardware.

For transistors fabricated as arrays with regularly positioned vertical terminals accessible from both the top and bottom sides, interconnects can be also made at the backside of the chip. After design sign-off on 3D FPGA, the programming values can be permanently provided to *Tier 1* by creating backside hard-wired interconnects, thus the final 2D ASIC can be created with exactly the same performance as 3D FPGA. It takes only a little Back End of Line (BEOL) masking cost of two metal layers to replace the programming logic of a verified and debugged design. The body of the 2D ASIC is identical as *Tier 1* of the 3D FPGA working as the design platform, thus designer's efforts of debugging, verification, and performance prediction performed on 3D FPGA remain valid for the ASIC implementation and FPGA providers' efforts make it silicon-proven. This proposed ASIC design methodology is fast, fully verifiable and hardware predictable.

In this work, successfully fabricated Vertical Slit Field Effect Transistor (VeSFET) technology [27-29] is selected to demonstrate the methodology. VeSFETs are fabricated as absolutely regular arrays of identical geometry devices with two-side accessibility. The transistor gates, source and drain terminals are implemented as two-side accessible metal pillars as described in Section 2.1. The performance of a 4-LUT based VeSFET FPGA is evaluated with real post-layout parameters by *VPR* 7.0 [58] [59]. Eleven MCNC benchmark circuits were implemented on 2D and 3D VeSFET FPGAs. Comparing the final 2D ASIC to the 2D FPGA, the performance of the final 2D ASIC as well as the performance of the 3D FPGA are on average 15% faster, consume 17% less power, and are 44% smaller.

Section 4.2 presents the proposed ASIC design methodology in detail. Section 4.3 describes VeSFET-based monolithic 3D FPGA. The migration cost is discussed un Section 4.4. The performance evaluation for MCNC circuits is given in Section 4.5. Section 4.6 summarizes this chapter.

4.2 The Design Methodology

FPGA design flow offers many advantages as compared to ASIC flow. Figure 4.2 shows the flows for FPGA and ASIC designs.

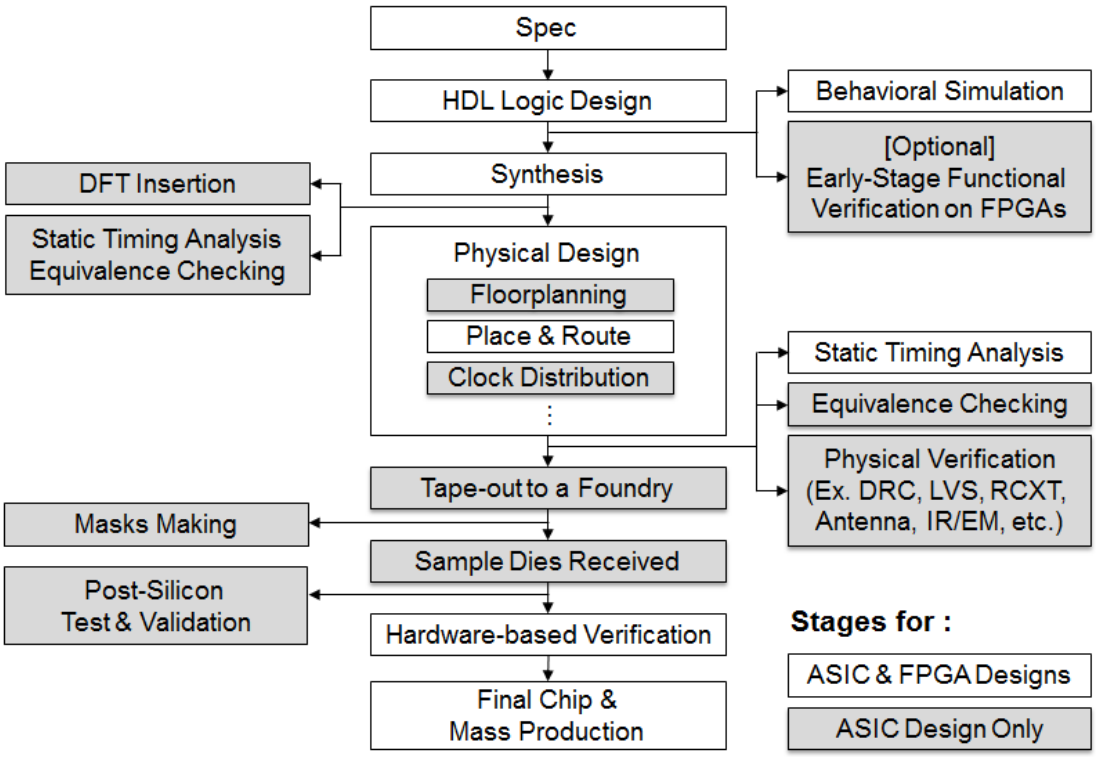


Figure 4.2. ASIC and FPGA design flows

The time consuming and complicated design stages such as physical design and its corresponding verifications, tape-out, fabrication, mask making, post-silicon tests and

validations are all required for an ASIC design but not necessary for FPGA-based implementations. The proposed design methodology is based on FPGA design flow, which has all the advantages FPGAs offer.

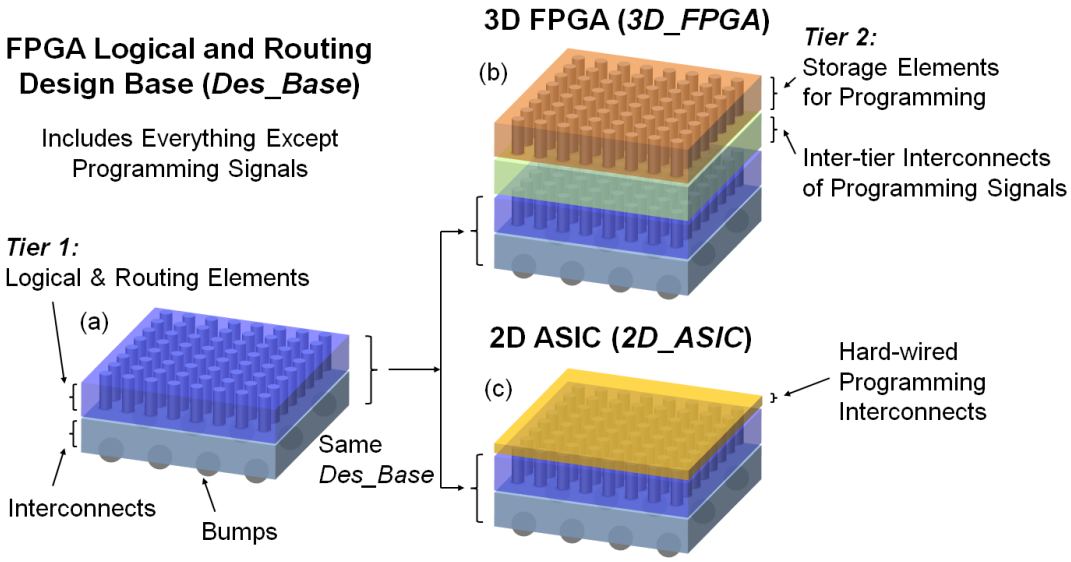


Figure 4.3. The chip structures of the proposed methodology. (a) The design base, which includes all the logical and routing elements of an FPGA (*Des_Base*); (b) 3D FPGA for design implementation (*3D_FPGA*); (c) The final 2D ASIC (*2D_ASIC*)

Figure 4.3 illustrates the chip structures of the design base (*Des_Base*), the 3D FPGA design implementation platform (*3D_FPGA*), and the final 2D ASIC chip (*2D_ASIC*). By leveraging monolithic 3D technology, a 2D chip can be partitioned into multiple tiers to reduce chip footprint and interconnect lengths. A 3D FPGA (*3D_FPGA*) is smaller, faster, and consumes less power than the equivalent 2D FPGA (*2D_FPGA*). Figure 4.4 shows the design flow and the design costs of this methodology. The design base (*Des_Base*) includes all the logical and routing elements (*Tier 1*) and interconnects of an FPGA. The logical and routing elements include configurable logic blocks (CLBs), switch boxes (SBs), connection boxes (CBs), hard macros (such as memories and DSPs), etc. CLBs provide the logic functions as

the building blocks of the design. SBs and CBs provide the flexible routings to connect CLBs and hard macros. These elements are all programmable to achieve different design functionalities, and are the source of FPGA's design flexibility.

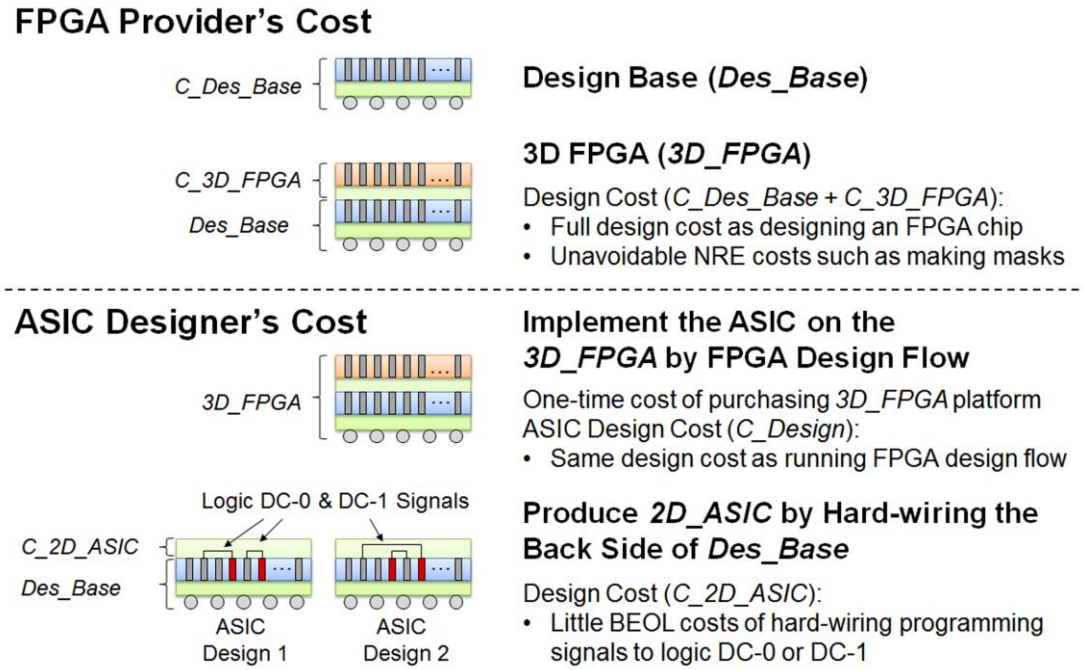


Figure 4.4. The design flow and the design costs of this methodology, for FPGA provider and ASIC designers

Tier 2 contains all the storage elements, such as SRAMs, to program the *Des_Base*. Between *Des_Base* and *Tier 2*, inter-tier interconnects provide the programming signals from the storage elements to the configurable elements in *Des_Base*. The *Des_Base*, *Tier 2*, and the inter-tier interconnects form the *3D_FPGA*. *Des_Base* and *3D_FPGA* are designed, verified, and fabricated by FPGA providers. The provider takes the full design cost as designing an FPGA chip, which includes the long design cycle, design and mask making NRE costs, and all the effort for hardware verification, validation, and debugging. In this proposed methodology, the ASIC designers do not need to take these costs.

In this ASIC design methodology, the design flow starts from the *3D_FPGA* purchased from the FPGA provider. The designers run the FPGA design flow as shown in Figure. 4.2 to design the ASIC. The design cycle is much shorter than conventional ASIC design flow. After the design is signed-off, the ASIC designers know the correct values of programming the configurable elements in *Des_Base*. The final 2D ASIC chip (*2D_ASIC*) can be fabricated with little BEOL costs (*C_2D_ASIC*) to hard-wire all the programming signals to logic DC-0 or DC-1 at the back side of *Des_Base*. The different ASIC designs can be easily obtained by providing different backside interconnects.

The costs of making an ASIC in this methodology is small. It takes a one-time cost of purchasing the *3D_FPGA* as the design platform, the design cost (*C_Design*) which includes all the design efforts and design time, and the BEOL cost of hard-wiring programming signals (*C_2D_ASIC*). The design cost (*C_Design*) is drastically reduced compared to a conventional ASIC design flow. All of the following efforts and costs of *Des_Base* are subsumed by the FPGA provider: 1) complicated and time consuming physical design and verification, 2) pre- and post-silicon verifications and calibration, 3) possible re-spins to make the hardware working, 4) mitigating all the unpredictable factors causing pre- and post-silicon performance gap, such as process variation, fabrication defects, model accuracy, and EDA tools precision, 5) mask making cost, etc. At the design stage working on *3D_FPGA*, the final *2D_ASIC* can be fully optimized, verified, and debugged since the design runs on exactly the same physical structure of *Des_Base*. In addition, the performance is fully predictable, which is the same as the performance of *3D_FPGA*.

4.3 VeSFET Monolithic 3D FPGA

An FPGA chip consists of three parts: 1) configurable logic elements and hard macros, 2) configurable routing elements, and 3) configuration memories, which are typically SRAMs. For different design purposes, one can decide how to partition FPGA into different number of tiers. The configuration SRAMs are the most intuitive parts to be placed on a separate tier since they don't affect timing and dynamic power consumption during normal operations. More aggressively, routing resources including switch boxes (SBs) and connection boxes (CBs) can be placed on another tier to further reduce the area and the interconnect wire length.

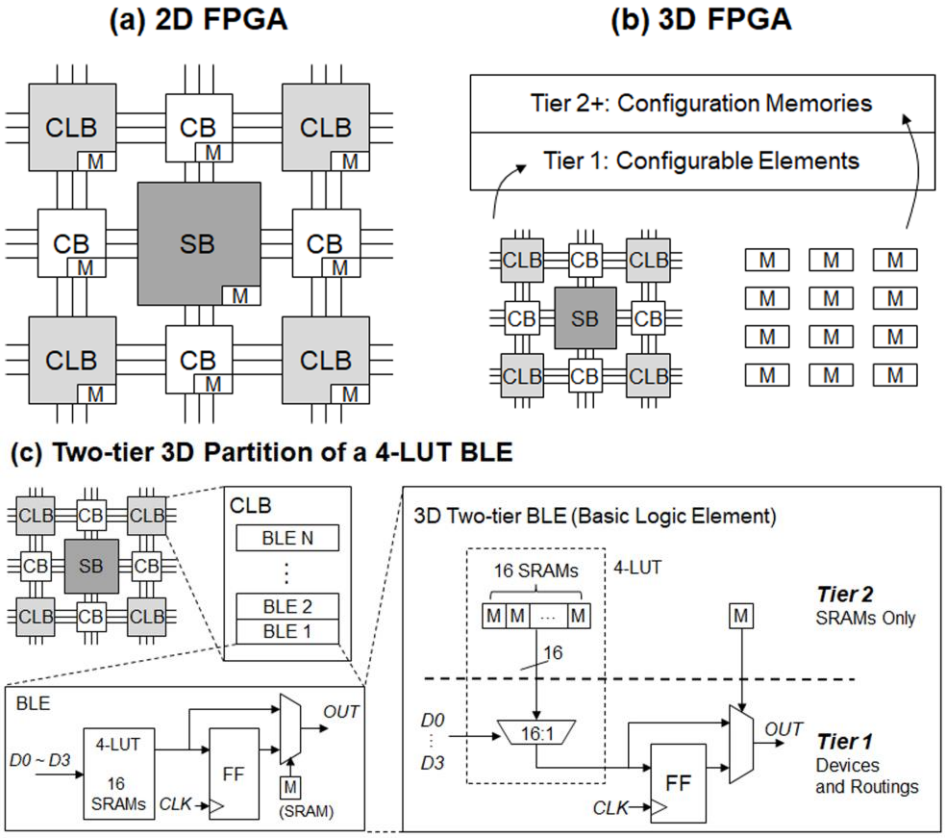


Figure 4.5. 2D to 3D FPGA migration. (a) a 2D conventional island style FPGA; i is configuration memory; (b) a 3D FPGA, all the configuration memories are placed on independent tier(s) (*Tier 2+*), the configurable elements are placed on *Tier 1*; (c) a two-tier implemented 3D 4-LUT basic logic element (BLE).

Figure 4.5 illustrates migration of a 2D FPGA to 3D. In a 2D conventional island style FPGA shown in Figure 4.5 (a), configuration memories (*M*) are placed together with configurable elements, such as configurable logic blocks (CLB), connection boxes (CB), and switch boxes (SB). In a monolithic 3D FPGA shown in Figure 4.5 (b), configuration memories are placed on a separate tier or several tiers (*Tier 2+*). The configurable elements and routing resources are placed on *Tier 1*.

A circuit-level example of migrating a basic logic element (BLE) implemented with 4-LUT (four-input look up table) into two tiers is shown in Figure 4.5 (c). The devices (multiplexers, flip-flops, etc.) and routing are placed on *Tier 1*. SRAMs placed on *Tier 2* provide configuration signals to control the function performed on *Tier 1*. Each CLB contains N identical Basic Logic Elements (BLEs) with direct connection to CLB's input and output ports. Each BLE is constructed by a 4-LUT, a Flip-Flop, and a MUX. In a *2D_FPGA*, each BLE requires 16 SRAM cells for a 4-LUT and an additional SRAM cell for final MUX selection. In the *3D_FPGA*, all the SRAM cells are moved to *Tier 2*. All the transistors besides those included in SRAMs are on *Tier 1*; with interconnects formed at the front side (package bump side) of *Tier 1*. Then, all the configuration signals from SRAMs and *Tier 2*'s own interconnects are formed at the back side (between *Tier 1* and *2*) of the *Tier 1*. They connect the SRAM cells, which will be formed on *Tier 2*, to the corresponding transistors in *Tier 1*. At the end, SRAM cells are formed on *Tier 2*.

The component layouts are designed, extracted, and then simulated to characterize the performance difference between the 2D and 3D circuits. Figure 4.6 shows the 2D and 3D layouts of a 4-LUT containing 16 SRAM cells, the circles are the metal pillars of VeSFETs. To achieve high physical regularity, VeSFET technology design rules restrict wires to be

strictly parallel on each metal layer, and the wire's start and end points have to be aligned on pillars. The diagonal wires are with width and spacing = 1.4*r*, the horizontal and vertical wires are with width and spacing = 2*r*. In the 4-LUT, the 16:1 MUX is realized by 4-stage transmission-gate type MUX2s. The four inputs *D0* ~ *D3* are internally buffered by a minimum Inverter (INV_1X) and an INV_2X of doubled drivability for positive phase, and an INV_2X for negative phase. The area of the SRAM cell used here is 20*r* × 16*r*, which is the VeSFET cell layout *v2* shown in Figure 2.7 in Section 2.3.1. For the purpose of further ASIC migration, Tie-High (Tie-H) and Tie-Low (Tie-L) circuits are inserted in *3D_FPGA Tier 1*, to provide logic "0" and "1" DC signals avoiding a direct connection to VDD / GND grids, which may result in reliability problems. Figure 4.7 shows conventional Tie-High, Tie-Low, and Tie-High & Low circuits.

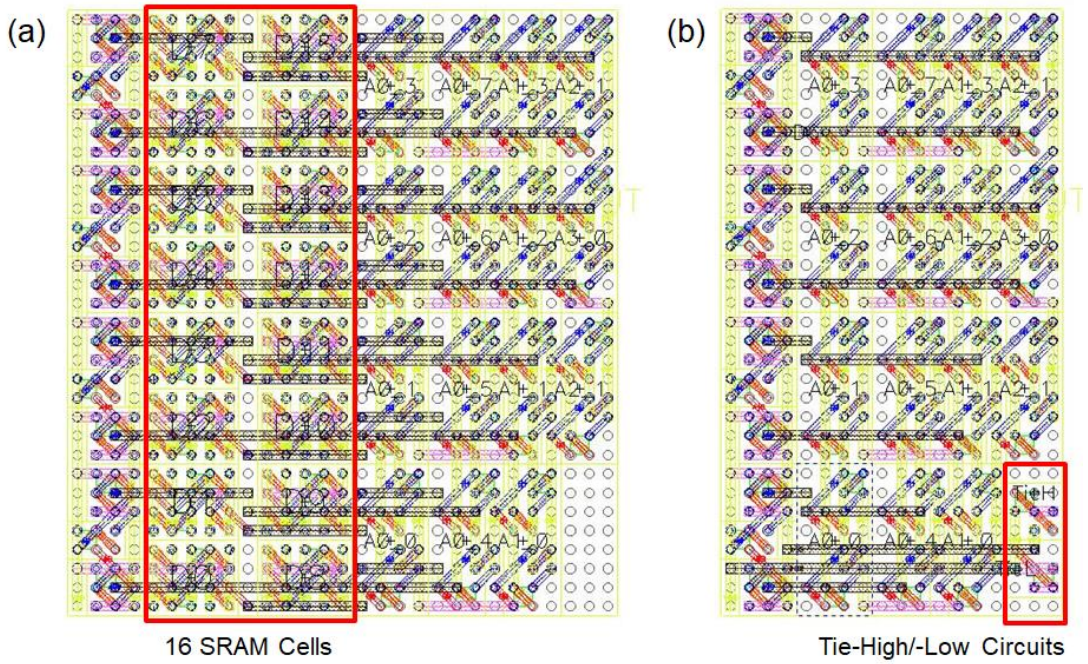


Figure 4.6. 4-LUT layouts. (a) 2D (116*r* x 128*r*, 100%); (b) 3D *Tier 1* (72*r* x 128*r*, 62%)

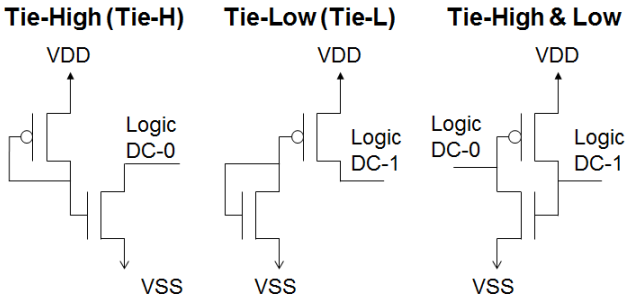


Figure 4.7. Conventional Tie-High, Tie-Low, and Tie-High & Low circuits

Table 4.1 summarizes the area, delay, and pin toggling energy of 2D and 3D 4-LUTs. The device model used here [39-40] is for VeSFET pillar radius r = 50nm, pillar height h = 200nm, tox = 4nm, *N*-type channel doping = 4e17 / cm^3, *P*-type channel doping = 5e17 / cm^3, and supply voltage VDD = 0.8V. From 2D to 3D, the area is reduced by 38%; the average *D0* to *OUT* delay is reduced by 1.4%, where *D0* is the selecting signal of the first MUX2 stage. The delay improvement is not significant since the delay is mainly determined by the SRAMs to *OUT* path, whose length is not significantly reduced. The pin toggling energy is reduced by 57.4% to 79.3%; 64.7% on average.

Table 4.1. 4-LUT performance comparison in *2D_FPGA* and *3D_FPGA*, (Pillar radius *r* = 50nm, *h* = 200nm, VDD = 0.8V)

4-LUT	Area (µm^2)	Delay (ps)	Pin Toggling Energy (fJ)			
			D0	***D1***	***D2***	***D3***
2D	37.12	328.06	9.82	5.54	2.57	1.40
3D	23.04	323.32	6.00	3.38	1.48	1.11
Ratio	62.1%	98.6%	61.1%	61.0%	57.4%	79.3%

4.4 3D FPGA to 2D ASIC Migration and the Costs

A *3D_FPGA* can be easily migrated to a *2D_ASIC* using the backside routing feature of two-side accessible transistors such as VeSFETs. As shown in Figure 4.3 and 4.4, the design base (*Des_Base*) contains all the necessary logic and routing elements such as CLBs, SBs, CBs, hard macros, clock networks, power delivery network, etc. Only the configuration signals are required to finish the design. In *3D_FPGA*, the inter-tier interconnects carry the configuration signals from SRAM cells, which are all DC signals after the FPGA is programmed.

All the design works including optimization and verification tasks, as well as the system and software development can be done on the *3D_FPGA*. At this stage, the designers can leverage all the advantages of FPGAs, such as: simpler design flow, fast design implementation, shorter design cycle and turn-around time, hardware based logic optimization and verification, etc. Once the design is signed-off on the *3D_FPGA*, the configuration SRAM cell values are fully determined. Based on the same *Des_Base* with VeSFET's unique structure, the backside routing can be performed to hard-configure the CLBs, SBs, CBs, or other blocks requiring configuration signals originally from the SRAM cells in *Tier 2*. In the *2D_ASIC* chip, all the transistors requiring configuration signals are now connected to the Tie-High (Tie-H) or Tie-Low (Tie-L) circuits at the backside of *Des_Base*. Since the same *Des_Base* is used in both *3D_FPGA* and *2D_ASIC*, both implementations have the same performance and all the verification done at the FPGA design stage remains valid for the final *2D_ASIC*.

The migration cost from *3D_FPGA* to *2D_ASIC* is very small; it requires only two layers of routing at the backside of *Des_Base*. The only new costs are the extra BEOL masks for

backside routing. Figure 4.8 shows the backside routing of the 4-LUT described in Section 4.3 and Figure 4.6. A 4-LUT requires 16 configuration signals from SRAMs. The dark gray circles show pillars of the corresponding transistors that need to be connected at the backside. A Tie-High logic DC-1 and a Tie-Low logic DC-0 pillar are shown in Figure 4.8 as well, which are the output of Tie-H and Tie-L circuits. In this layout, there are 18 vertical routing tracks and 32 horizontal tracks available with width = 2*r* and spacing = 2*r*. Since there are only at most two nets, logic DC-1 and DC-0, to be routed, the available routing resource is sufficient to route with only two metal layers. Figure 4.8 also shows an implementation with two metal layers, which can connect any configuration combinations.

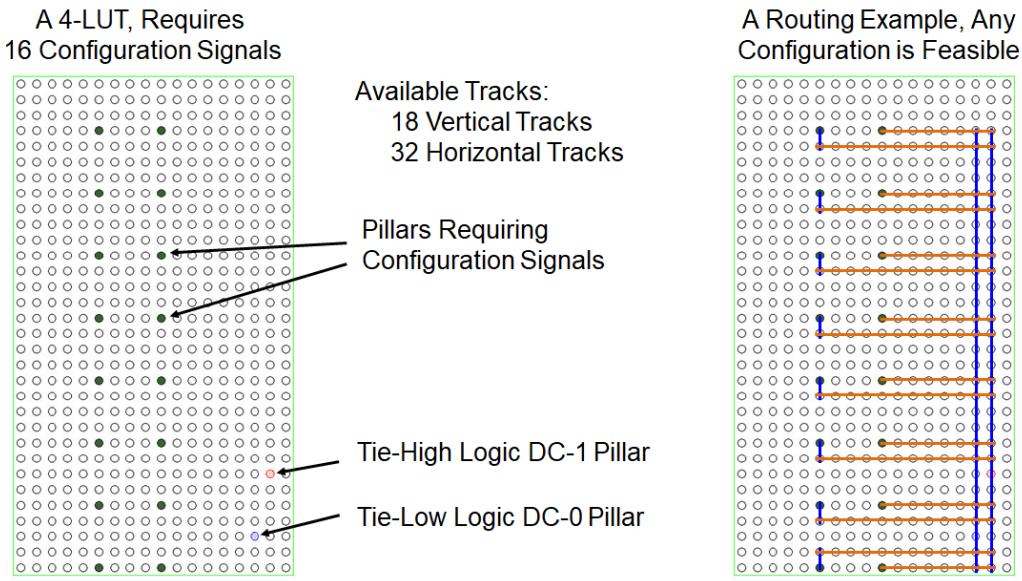


Figure 4.8. Two-layer backside routing of a 4-LUT in *2D_ASIC*

Another consideration related to the backside routing is the current drivability of the Tie-H and Tie-L circuits. There are two possible connections between a Tie-H / L circuit and the corresponding transistor. One is to a gate and the other one to a drain or source, as shown in Figure 4.9. For the case such as LUTs implemented with pass-gate type MUX structure, the

Tie-H / L circuit is connected to the transistor's drain or source. The charge and discharge currents are determined by the Tie-H / L circuit, which dominates the speed. To maintain the same performance on *3D_FPGA*, the Tie-H / L circuit needs to be designed with the same drivability, i.e. to have the same current drivability as an SRAM cell or a buffer. For the case such as the end MUX in a BLE, the Tie-H / L circuit is connected to transistors' gates. As long as the current provided by the Tie-H / L circuit is greater than the gate leakage current, the DC voltage can be maintained.

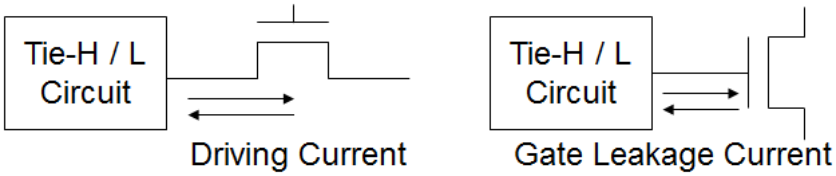


Figure 4.9. Tie-H / L circuit connection scenarios. Left: to drain or source; Right: to gate

4.5 Performance Evaluation

The performance is evaluated by *VPR* 7.0 for 11 MCNC LGSynth'91 benchmark circuits. The benchmark circuits are synthesized by *ABC* [60], and then iteratively synthesized until the gate count stabilizes. Then, the circuits are mapped to LUTs. In order to run power estimation in *VPR*, another tool *ACE 2.0* [61] is used to calculate the activity of each net, which is a required input for *VPR*'s power estimation flow. The activity calculation is based on assuming each primary input (PI) has: 1) 50% long term probability that it is logic high and 2) 50% toggling probability per clock cycle. Several representative circuits such as BLEs, MUXs, Flip-Flops, one-channel Switch Box, etc. are designed, their parasitics extracted from layouts, and *Hspice* simulated to get a more accurate performance estimation. The VeSFET

model and the FPGA architecture have been described in Section 4.3, the FPGA is conventional island-style implemented by 4-LUTs. The switch box flexibility *Fs* = 3, connection box's flexibility *Fc* = 0.5. There are four types of wire segments used: *Single*, *Double*, *Seg-3*, and *Seg-6* with one, two, three, and six CLB tile widths. The ratio of wire segment numbers is 24:40:36:96 for *Single*: *Double*: *Seg-3*: *Seg-6*. A portion of *VPR* code has been modified to reflect VeSFET's width quantization effect and the area calculation. The area is decided by *VPR* with the aspect radio set to 1. At first, we run *VPR* for a circuit on both *3D_FPGA* and *2D_FPGA* configurations, and obtain the minimum routing channel width required under the densest CLB packing calculated by *VPR*. Then we use the same routing channel width condition for both *3D_FPGA* and *2D_FPGA* and get the performance by *VPR*. Table 4.2 summarizes the circuit information after *ABC* synthesis and *VPR* place and route. PIs and POs are the primary inputs and primary outputs of the circuit. Figure 4.10 to 4.12 show the critical path delay, power consumption, and area comparisons between the *3D_FPGA* (same performance as the final *2D_ASIC*) and *2D_FPGA*. The FPGA area reported in Figure 4.12 includes all the fully or partially used CLBs and routing. Figure. 4.13 summarizes the *3D_FPGA* (*2D_ASIC*) to *2D_FPGA* ratios of delay, power consumption, power-delay product (PDP), and area.

Table 4.2. Benchmerk Circuits *ABC* Synthesis and *VPR* Place & Route Information

Circuits	*ABC* Synthesis				*VPR* P&R	
	# of PIs	# of POs	# of FFs	# of 4-LUTs	# of CLBs Used	% of CLB Usage
bigkey	262	197	224	1101	121	94.2%
C6288	32	32	0	532	81	82.7%
C7552	207	108	0	498	64	89.1%
clma	382	82	33	2729	361	94.7%
des	256	245	0	1310	169	97.0%
dsip	228	197	224	1108	121	95.0%
i10	257	224	0	758	100	95.0%
pdc	16	40	0	822	121	85.1%
s38417	28	106	1636	3448	441	97.7%
s38584.1	38	304	1426	3506	441	93.4%
spla	16	46	0	468	64	92.2%

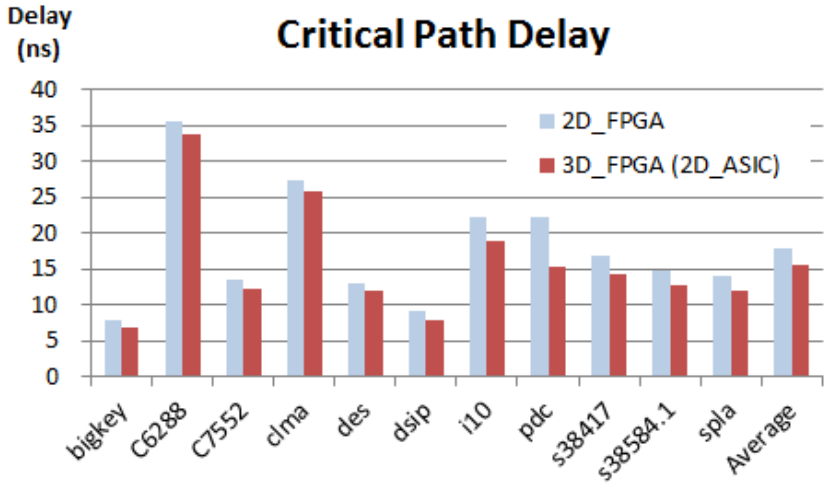


Figure 4.10. Critical path delay of *2D_FPGA* and *3D_FPGA* (*2D_ASIC*). On average, *2D_FPGA* is 17.9 ns, *3D_FPGA* (*2D_ASIC*) is 15.7 ns

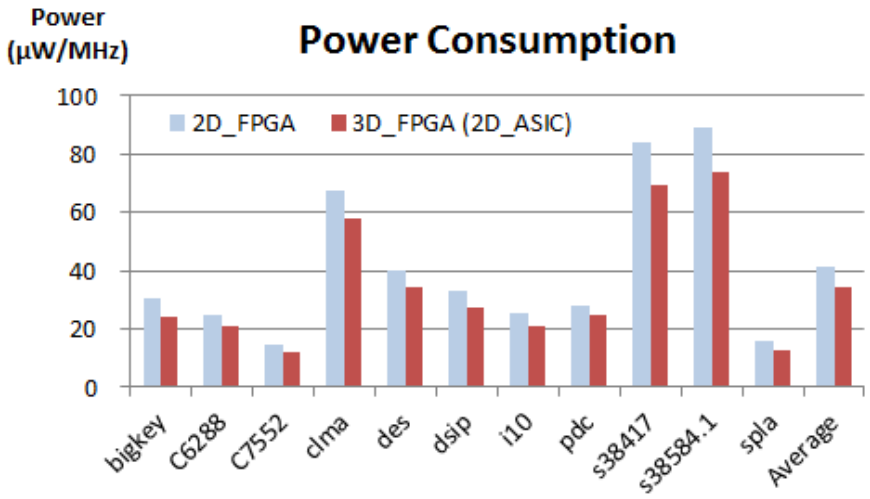


Figure 4.11. Power consumption comparison of *2D_FPGA* and *3D_FPGA* (*2D_ASIC*). On average, *2D_FPGA* is 41.1 µW/MHz, *3D_FPGA* (*2D_ASIC*) is 34.3 µW/MHz

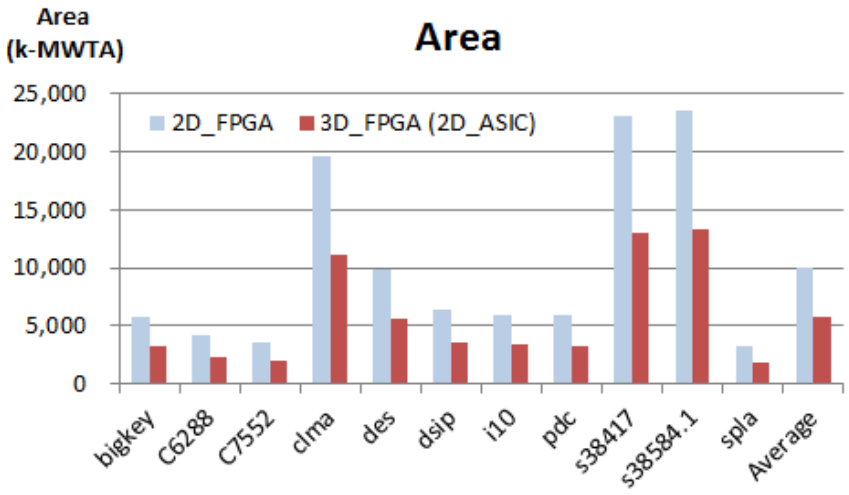


Figure 4.12. Area comparison of *2D_FPGA* and *3D_FPGA* (*2D_ASIC*), in Kilo Minimum Width Transistor Area (k-MWTA), for VeSFET with *r* = 50nm, MWTA = 0.04µm^2. On average, *2D_FPGA* is 10067.2 k-MWTA, *3D_FPGA* (*2D_ASIC*) is 5693.3 k-MWTA

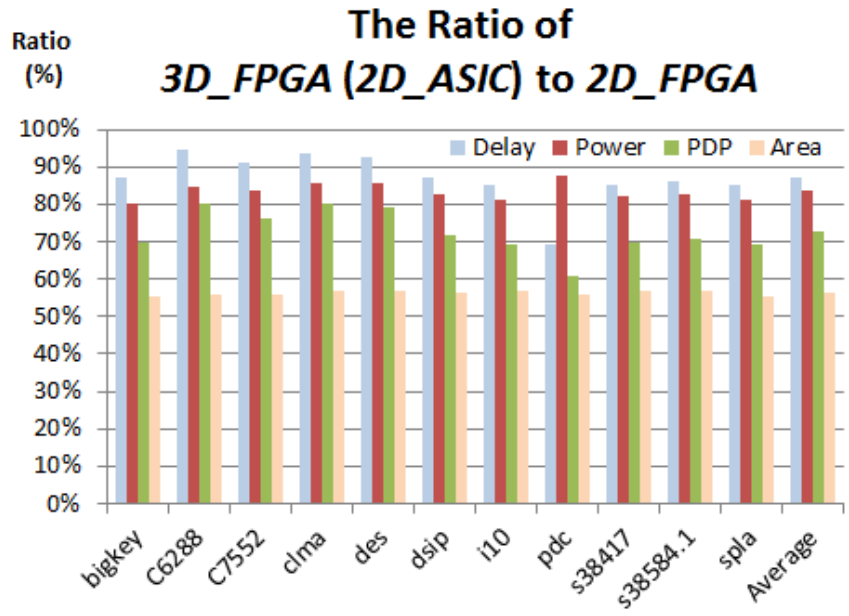


Figure 4.13. The delay, power consumption, power-delay product (PDP), and area ratios of *3D_FPGA* (*2D_ASIC*) to *2D_FPGA*. On average, *3D_FPGA* (*2D_ASIC*) is 87%, 83%, 72.6%, and 56% to *2D_FPGA* in delay, power consumption, PDP, and area, respectively

Comparing *3D_FPGA* to *2D_FPGA*, the critical path delay is 13% reduced on average, in terms of the maximum frequency achievable, *3D_FPGA* is 15% faster. The total power consumption is 17% reduced on average. The area is 44% reduced on average. The area reduction is similar to the 43% 2D to 2-tier 3D reduction reported in [16]. The power-delay product (PDP) of the *3D_FPGA* (*2D_ASIC*) is 72.6% of the *2D_FPGA*. This FPGA is based on the fundamental 4-LUT structure with no hard blocks. A better performance could be expected with a well-designed FPGA that includes hard blocks and more complex CLBs.

4.6 Summary

This chapter proposes a fast, fully verifiable, and hardware predictable ASIC design methodology for circuits implemented with two-side accessible pillar-array-based transistors.

This flow is possible because of such transistors' unique 3D monolithic integration and two-sided routing capabilities of such devices. In the proposed methodology, the ASIC design is first implemented on a two-tier 3D FPGA platform, which has all the logical and routing elements in *Tier 1* and the programming elements in *Tier 2*. The 3D FPGA which is fully hardware verified and validated by FPGA vendors is used as an intermediate platform by the ASIC designers to implement the design by a conventional FPGA design flow. After the design is signed-off and fully verified on the 3D FPGA, the final 2D ASIC is configured on a chip which is exactly the same as *Tier 1* of the 3D FPGA with two additional layers of interconnects at its backside. Those extra interconnects hard-program the configurable elements in *Tier 1* to logic DC-0 or DC-1 as the values provided by *Tier 2*'s programing elements in the 3D FPGA. Performance of the final 2D ASIC is exactly the same as that of the design implemented on the 3D FPGA and all the verifications are done based on the real and exactly the same hardware.

In this proposed methodology, all the time-consuming and costly efforts such as 1) physical design and verification, 2) pre- and post-silicon verification, validation, and calibration, 3) possible hardware re-spins, 4) mitigating all the unpredictable factors causing pre- and post-silicon performance gap, such as process variation, fabrication defects, model accuracy, and EDA tools precision, 5) mask making, etc. are performed by the 3D FPGA provider instead of the ASIC designers. The ASIC designers can implement the 2D design with all the benefits an FPGA design flow can provide and achieve better performance than a 2D FPGA chip.

The performance evaluation with eleven MCNC LGSynth'91 benchmark circuits shows that the 3D FPGA, as well as the final 2D ASIC, are 44% smaller in terms of area, 15% faster,

and consume 17% less power than the 2D FPGA on average. The power-delay product (PDP) ratio is 72.6%.

Chapter 5

HIGH PERFORMANCE DYNAMIC RECONFIGURABLE COMPUTING WITH ACCELERATORS

System accelerators improve performance, break power and utilization walls. They can be implemented by fixed-function hard macros or reconfigurable logic, such as Field Programmable Gate Arrays (FPGAs). For systems running various applications, dynamic reconfigurable accelerators offer a very attractive feature; however, the reconfiguration time is an unavoidable overhead. This chapter proposes high performance architecture with fast dynamic reconfigurable FPGA accelerators (F-RACCs) based on a novel bitstream re-programming method, which is feasible by using emerging technologies.

5.1 Introduction

Growing computational demands from various application domains urge development of new technologies and architectures to provide higher computing power. However, power and utilization walls hinder further increase of general processors' computing capability achievable through scaling, which had been the main driver of the electronic industry for decades. To address this bottleneck, the ideas of moving specific tasks from general-purpose cores to specially designed computing units have been proposed. Conservation cores architectures [17] delegate jobs to specialized processors to save energy. Accelerator-rich

architecture [18-20] includes a sea of heterogeneous dedicated hardware accelerators implementing different functions that may be invoked by applications running on the system. The performance, either speed or power, of specific applications can benefit from these specialized computing units.

However, a system may need to handle wide spectrum of applications. Data centers or cloud computing systems need to respond to all kinds of requests; their workloads demand flexible and efficient computing platforms. In such cases, it is impractical to provide all kinds of specialized accelerators in accelerator-rich architectures. To address the demands of flexibility and performance, reconfigurable computing architectures provide attractive characteristics for this kind of systems [21-22]. The key idea is to offload system tasks to reconfigurable processing units, such as accelerators implemented by FPGAs. Critical concern of deploying FPGA accelerators is the configuration time. For example, to partially reconfigure a region spanning 100 Configurable Logic Blocks (CLBs) in a Virtex-5 device, it takes 73.8μs [62], which for a 2GHz core clock rate translates to 147,600 cycles.

Limited data width of loading bitstream is the main bottleneck of FPGA configuration time. This is a physical limitation; the number of chip I/Os is limited, and all configuration bits must be distributed to their destinations across whole chip. To mitigate this bottleneck, new physical domain architectures are needed. Such architectures could be based on novel technologies, such as monolithic 3D integration, and two-side accessible transistors. Monolithic 3D integration technology can provide great amount of short vertical interconnection channels between devices on different tiers [3-8] [10-16]; however, extra monolithic inter-tier VIAs (MIVs) are required to connect adjacent tiers. Two-side accessible transistors such as Vertical Slit FET (VeSFET) [27-40], naturally provide vertical

interconnection channels within transistors for monolithic 3D integration. Every transistor terminal is directly accessible from its top and bottom sides so there is no need for extra area overhead for MIVs. In addition, the two-side accessibility provides opportunity of creating interconnects on both front and back sides of the chip [35-36].

This chapter proposes a high performance reconfigurable architecture using fast dynamic reconfigurable FPGA-based accelerators (F-RACCs) implemented with VeSFETs integrated into 3D monolithic structures. The proposed architecture is similar to the accelerator-rich architecture proposed in [18-20] with fixed function dedicated accelerators and includes CPU cores, caches, memories, accelerators, and Network-on-Chips (NoCs). But in our architecture, the accelerators are dynamically fast reconfigurable. The accelerators' functions change on the fly in a short time to satisfy the demand of the applications executing on the system. This provides flexibility for the system and solves the long reconfiguration time concern of FPGA-based accelerators.

The results are evaluated on eleven benchmarks from different domains by *PARADE* [63], a *gem5* [64] based cycle-accurate full-system simulation platform for accelerator-rich architectures. Comparing with the system using conventional FPGA accelerators (C-RACCs) partially configured with fastest configuration speed, we observe that our architecture using F-RACCs improves system performance on all eleven benchmarks and achieves maximum speedup 1.31x and 2.82x using 1 and 12 accelerator instances, respectively. The speedup over CPU software path without any accelerator is 94.93x with one and 565.12x with 12 F-RACCs.

The main contributions of this work are:

1. We propose a high performance flexible architecture using fast dynamic reconfigurable accelerators (F-RACCs). This architecture targets systems processing

wide spectrum of applications, such as cloud computing systems or data centers. It breaks the main performance bottleneck of deploying FPGA accelerators in a reconfigurable system – the configuration time.

2. We explore the benefits of emerging technologies: monolithic 3D integration and two-side accessible transistors such as VeSFET. To the best of our knowledge, this is the first work demonstrating 3D monolithic VeSFET-based architecture for rapid reconfigurable computing.
3. We combine monolithic 3D integration technology and VeSFETs to drastically reduce FPGA's reconfiguration time by physical structure evolution.
4. We propose an FPGA-based accelerator, whose function can be changed in several cycles. We describe a VeSFET-based implementation of this dynamic reconfigurable computing unit.
5. We create rapid reconfigurable computing capability to an accelerator-rich architecture by integrating our fast dynamic reconfigurable accelerators. System-level performance is assessed. To the best of our knowledge, this is the first accelerator-rich architecture supporting rapid reconfiguration computing.

The rest of the chapter is organized as follows. Section 5.2 provides an overview of prior works on reconfigurable systems. Section 5.3 describes details of the proposed architecture. Section 5.4 presents the simulation environment and experimental results. Section 5.5 concludes this chapter.

5.2 Prior Works on Reconfigurable Systems

Hardware reconfigurable accelerators improve system performance and provide system flexibility. Reference [21] presents a broad overview of reconfigurable computing systems,

including architectural aspects, technologies, computing models, tools, applications, and history. In [22], the authors give a survey of reconfigurable accelerators for cloud computing.

Many recently published works have demonstrated significant speedup over running on CPU software path [65-71]. Microsoft proposed Catapult fabric for data center services [65]. One FPGA is placed into each server of the medium scale reconfigurable fabric consisting of 1632 servers. Each server achieves 1.95x speedup for a fixed latency distribution in accelerating the Bing web search ranking engine. IBM offloads database-analytics queries to FPGAs for acceleration, which achieves 14.6x speedup [66]. A system proposed in [67] uses FPGAs to accelerate in-memory database operations of selection, sorting, and joining. It achieves 5.7x speedup for sorting operations.

References [68-69] describe implementation of Google's MapReduce [72] framework for large dataset processing using different reconfigurable accelerators. Reference [68] uses FPGAs to accelerate RankBoost [73], a ranking algorithm, and achieves 31.8x speedup. In [69], FPGA-implemented accelerators are used for k-means clustering algorithm achieving 15.5x to 20.6x speedup. Besides FPGAs, Graphics Processing Unit (GPU) may implement reconfigurable computing. GPUs have massively parallel architecture and can handle multiple tasks simultaneously. Axel cluster [70] deploys both FPGA and GPU accelerators in the system. It achieves 4.4x to 22.7x speedup for N-body simulation that models interaction of N particles influenced by gravity forces in the space. Reference [71] uses Coarse-Grained Reconfigurable Architecture (CGRA) to accelerate MapReduce framework without FPGAs or GPUs. CGRA includes an array of coarse-grained processing elements; they are connected and programmed to perform certain functions. The speedup of 30x to 60x is demonstrated in [71] on matrix multiplication, k-means clustering algorithm, and 2-D convolution.

Besides the system-level designs, prior works [74-76] proposed and assessed different reconfiguration techniques. Reference [74] proposed a time-multiplexed FPGA based on the same concept as the FPGA reconfiguration method used in this work. It has one active and eight inactive configurations in the memory distributed throughout the die. Its function can be changed by activating different configurations. Since all the memories and configurable elements are located on the same 2D plane, the area and the routing resources could be the bottlenecks. Reference [75] uses novel optically reconfigurable gate arrays (ORGAs), which are configured by laser array light, holographic memory, and photo diode array. Optical reconfiguration could provide shorter reconfiguration time and consume less power than electrically reconfiguration. However, the overhead comes from the integration with optical devices on system level. Reference [76] compares two very different FPGA implementation philosophies. The first one uses individually specialized kernels (accelerators). Each kernel requires a specialized FPGA configuration. The second type of FPGA is based on an overlay approach that saves the reconfiguration overheads. For all kernels, it uses the same instruction programmable overlay vector co-processor. The results show that specialized kernels are 2.5x faster than the overlay approach. Depending on the optimization goal, systems can be implemented by mixing the advantages of the individual specializations (higher performance, more reconfiguration overhead) and overlay approaches (slower, but less reconfiguration overhead).

5.3 The Proposed Architecture

Emerging technologies provide opportunities for more efficient implementations of many applications. The key elements of the proposed architecture are the fast dynamic reconfigurable FPGA-based accelerators (F-RACCs). The function of an FPGA is determined

by the bitstream. If we can switch the bitstream rapidly, the function can be fast dynamic reconfigurable. For conventional FPGA implementations, physical domain causes the reconfigurable bottleneck. Fundamental changes of physical implementation are required to break it.

The proposed high performance fast dynamic reconfigurable architecture targets systems running wide spectrum of applications. It is an accelerator-rich architecture implemented with fast dynamic reconfigurable FPGA accelerators (F-RACCs) instead of fixed-function dedicated heterogeneous accelerators.

5.3.1 Overview

The accelerator-rich architecture described in [18-20] [63] is shown in Figure 5.1. We select this architecture as the foundation for implementing our fast dynamic reconfigurable architecture because of availability of simulation tools and the corresponding benchmarks. It has many CPU cores, DRAM memory controllers, shared last-level caches (LLC), a sea of fixed-function specialized hardware accelerators (ACC), and a global accelerator manager (GAM). These units are connected by a Network-on-Chip (NoC). All CPU cores and accelerators share the coherent LLC and memory controllers.

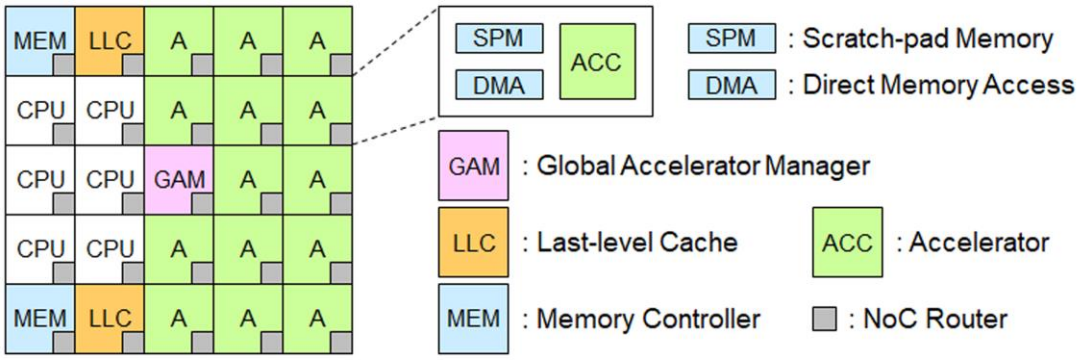


Figure 5.1. Accelerator-rich architecture in [18-20] and [63].

A CPU core has private L1 caches; it reads data from the shared coherent LLCs, which are typically L2 caches. If cache misses happen, data requests to DRAM memory controller are required. An accelerator has a local scratch-pad memory (SPM) for local storage and a Direct Memory Access (DMA) unit for data access. GAM owns a local SPM and DMA, and maintains a centralized Translation Lookaside Buffer (TLB). The TLB caches the virtual-to-physical address translations shared by all accelerators.

When a running application calls a function that can be offloaded to an accelerator, the CPU queries GAM for availability, waiting time, and expected computation time required by the accelerators implementing this function. Then, the CPU sends request to GAM to reserve and use the accelerator. At first, all the input data required by the accelerator are loaded into the accelerator's local SPM. When the computation is completed, the output data is written from SPM back to the shared LLC and memory. A CPU core receives accelerator's "job finish" notification through lightweight interrupt (LWI). The detailed descriptions of how the GAM interacts with CPU and accelerators are provided in [18-20] [63]. GAM maintains a resource table to track the accelerators, including their availability and waiting time.

It is intuitive that the system performance can be improved by providing more accelerating functions. The architecture in [18-20] and [63] uses a sea of fixed-function dedicated accelerators. Thus, there exists an upper bound on performance improvement due to the limited resources of chip area. To break this limitation, reconfigurable accelerators can provide the functions on demand while the number of accelerator instances remains the same. Figure 5.2 shows the proposed architecture, it maintains the major portion of the conventional accelerator-rich architecture as described before and uses F-RACCs to implement accelerators. Although the fixed-function accelerators may be well tuned and specialized for performance

and power optimization, F-RACCs provide the flexibility of rapidly changing functions as demanded by applications. The architecture of an F-RACC is described in Section 5.3.2.

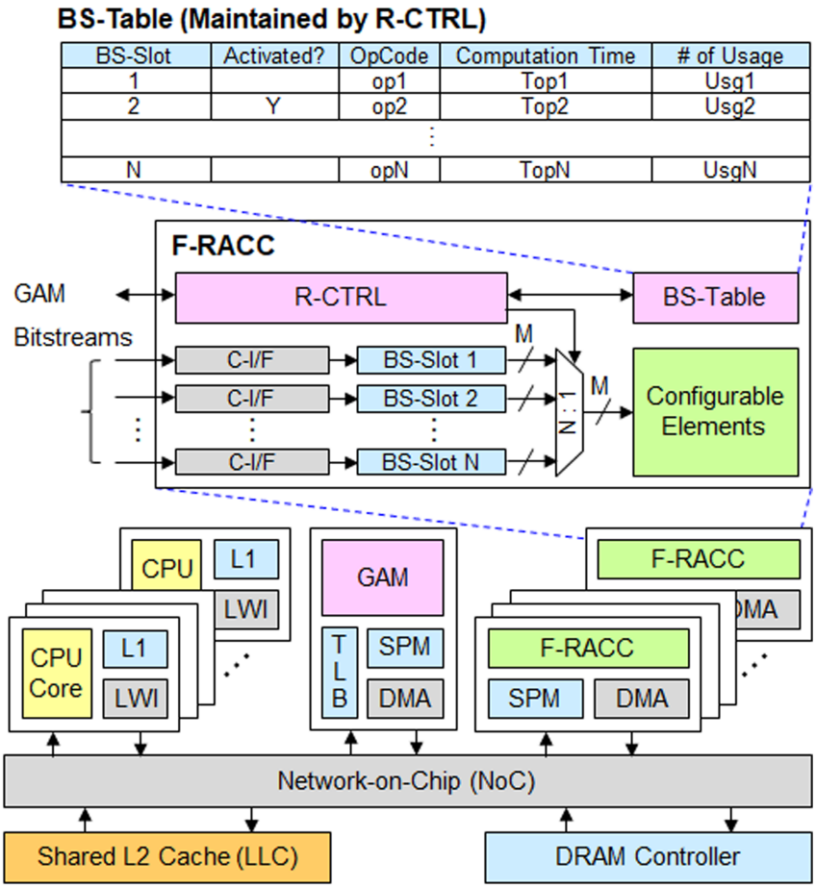


Figure 5.2. The proposed high performance reconfigurable architecture using fast dynamic reconfigurable accelerators (F-RACCs).

5.3.2 Fast Dynamic Reconfigurable Accelerators

A fast dynamic reconfigurable accelerator (F-RACC) is a monolithic 3D FPGA built of VeSFETs. Its physical architecture is described in Section 4.3.

An F-RACC, as shown in Figure 5.2, includes 1) FPGA's configurable elements (logic and routing), 2) several sets of bitstream slots (BS-Slots) and bitstream configuration interface(C-I/F), 3) bitstream switching logic, 4) a reconfiguration controller (R-CTRL), and

5) a bitstream configuration table (BS-Table). In FPGA-based F-RACCs, configurable logic elements and routings are placed on *Tier 1* of the monolithic 3D stacked chip. The storage elements of configuring bits are placed on *Tier 2+*. Multiple, independently programmable bitstream slots (BS-Slots) are provided on the storage element tiers. The BS-Slots store bitstreams of different functions; one of them is activated at a time to program the configurable elements on *Tier 1* and determines the function of the F-RACC.

A BS-Slot is configured in the same way as a bitstream is read in a conventional FPGAs, thus all the techniques applied in modern FPGAs to enhance bitstream loading speed are feasible here. Between *Tier 1* and *Tier 2+*, all the routing resources are used for distributing configuration signals, and all two-side accessible device terminals on *Tier 1* are directly accessible by *Tier 2+*. Thus, all configuration bits can program the FPGA in parallel. Activating different BS-Slots can rapidly reconfigure FPGA. When a BS-Slot is activated by the controlling bitstream switching logic, all its bits are distributed in parallel from upper 3D tiers (*Tier 2+*) to the target configurable elements on the lower 3D tier (*Tier 1*), as illustrated in Figure 5.3. In SRAM- or latch-based configuration memories, we can easily connect one of the cross-coupled inverter's nodes to bitstream switching logic.

Comparing to a 2D FPGA, with configuration memories (*M*) placed together with configurable elements, this approach provides several advantages: 1) there is enough room for including many BS-Slots; 2) distributing all configuration bits in parallel is feasible; and 3) chip area is smaller, with better timing and power characteristics, as reported in [15-16] and Chapter 4. These characteristics lead to a rapid reconfigurable architecture using smaller, faster, and less power consuming F-RACCs. Future implementations may consider emerging

Nonvolatile (NVM) cells [77] [78], which are smaller and consume less power than SRAM. More and lower-power BS-Slots may be provided to support more functions in an F-RACC.

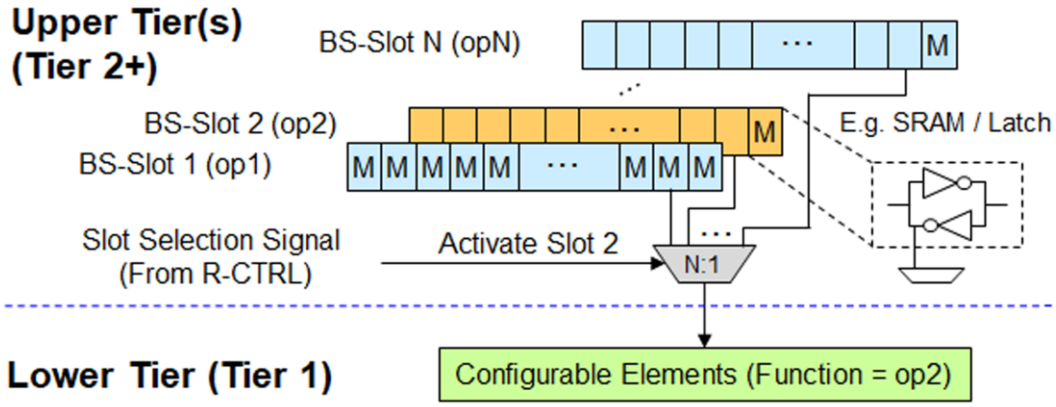


Figure 5.3. The configuration structure. R-CTRL selects bitstream slot 2 (BS-Slot 2) for activation. The configurable elements are programmed to function op2.

An FPGA accelerator is idle for reconfiguration at run-time, which leads to performance loss. For example, the whole bitstream size of the smallest FPGA of Xilinx Kintex-7 family, XC7K70T, is 24,090,592 bits [79] [80]. If this entire FPGA is programmed in parallel mode with maximum data width of 32 bits and configuration clock 100MHz, we can roughly calculate the required time is ≈ 7.53ms. If the main processor's core clock rate is 2GHz, it takes about 15 million core clock cycles to reconfigure the entire FPGA. To address this issue, modern FPGAs support partial reconfiguration at run-time [62], which greatly reduces the bitstream size.

The C-I/F is a bitstream configuration interface as used in conventional FPGAs; all the existing FPGA configuration methods are feasible for this C-I/F. It can take the advantages of modern bitstream configuration speedup techniques, such as partial reconfiguration, to improve performance. Each BS-Slot is independently, externally configured by a C-I/F, which means the system can arbitrarily configure any BS-Slot at any time as long as the F-RACC is

not executing the function configured by that BS-Slot. The size of a BS-Slot depends on the number of configuration elements in the F-RACC. If the goal is accelerating certain functions, the number of configured elements does not need to be as large as an entire commercial FPGA chip. It can be a small FPGA macro, as long as it contains enough resources for the accelerating functions.

The reconfiguration of an F-RACC is controlled by R-CTRL. It communicates with GAM, maintains the local BS-Table, and handles dynamic reconfiguration by activating different BS-Slots. BS-Table stores the information and usage status of each function configured by each bitstream stored in a BS-Slot. When a bitstream is written into a BS-Slot [n] through C-I/F [n], BS-Table stores its function (OpCode) and the computation time in the corresponding row [n], and resets its number of usage to 0. R-CTRL also marks the currently activated BS-Slot in BS-Table. When GAM queries the information of the F-RACC, the R-CTRL reads BS-Table and sends the following information to GAM: 1) the functions currently supported by this F-RACC, 2) time taken to configure this F-RACC to a specific function, and 3) the computation time required of the specific function. GAM uses the information for F-RACC resource management. It maintains a hardware accelerator resource table to track each accelerator's status [63].

However, it is impossible to provide unlimited BS-Slots with limited area resources. When a new function is queried to be configured but all the BS-Slots are occupied, it will replace an existing function according to BS-Slot replacement policy. This process is similar to CPU cache line replacement; we want to keep the frequently, recently, or going to be used data in the cache to improve the performance. Although it is possible that the application knows when and how to call and configure the accelerators in the program trace for hiding the programming

overheads, such optimizations work only if there is one application running on the system. The power of the proposed dynamic fast reconfigurable architecture is the capability of fast switching accelerator functions to meet the requirements of different applications running on the system. We need to have BS-Slot replacement policy to address the different needs from different applications executed together on the system.

In this architecture, the system performance depends on how fast the F-RACCs are configured into the functions requested by the application. If the BS-Slot replacement policy matches the application's accelerator usage behavior, the system can achieve higher performance. For example, if a program tends to call new accelerated functions, a Least Recently Used (LRU) replacement rule, which discards the bitstream stored in the least recently used BS-Slot, could lead to better performance. If a program uses a few specific accelerated functions often and sometimes calls a new function, a Least Frequently Used (LFU) replacement rule could be a better option. In this work, we use a "LFU then LRU" replacement for accommodating different behaviors of diverse programs. The bitstream from the least frequently used BS-Slot is discarded. When multiple candidates exist, the least recently used BS-Slot is discarded. Different replacement rules could improve/deteriorate programs having different behaviors. BS-Table tracks the usage of each BS-Slot. R-CTRL relies on the usage information to determine which function, i.e. bitstream, will be replaced. However, there is no best replacement policy for every application. A newly configured bitstream could be discarded by this "LFU then LRU" policy even if it will be used frequently in the future. Such effect can be mitigated by providing more BS-Slots or more F-RACCs. Since GAM has the hardware accelerator resource table for global resource tracking, other F-RACCs could be called instead of continuously replacing bitstreams in the same BS-Slot of an F-RACC.

5.3.3 Fast Dynamic Reconfiguration

Depending on the configuration status, an F-RACC requires different initialization time prior to executing the function. Figure 5.4 shows how an F-RACC performs dynamic reconfiguration and Table 5.1 lists the symbols used here. When GAM queries an F-RACC for availability of a function (OpCode) in demand, its R-CTRL checks the BS-Table for the status of the BS-Slots and responds to GAM with the initialization time *Tinit* required for this function, as shown in Figure 5.4(a).

There are three possible situations of the queried function:

1. The function is active, a bitstream implementing it has been written to a BS-Slot already, and this BS-Slot is selected to program the configurable elements. The F-RACC is currently configured and ready to execute the function. In this case, *Tinit* equal to *Treset* may be required to reset this accelerator for next usage.
2. The function is currently not supported, which means the F-RACC needs to take *Tinit* equal to the time *Tconfig* required for loading a new bitstream to a BS-Slot. *Tconfig* depends on the bitstream size and programming techniques implemented in C-I/F.
3. The function has been programmed in one of the BS-Slots, but this BS-Slot is not currently activated. Then, it takes *Tinit* equal to *Tswitch* for activating the BS-Slot by bitstream switching logic, which distributes all bits of the to-be-active bitstream to the configurable elements. As described in Section 5.3.2, all bits can be distributed in parallel, thus *Tswitch* can be just a few cycles.

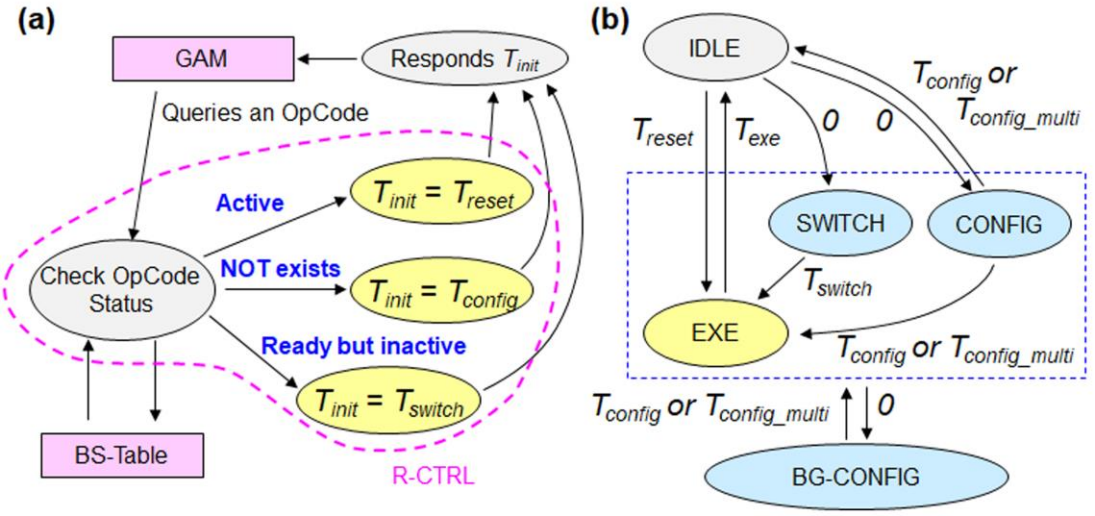


Figure 5.4. F-RACC dynamic reconfiguration states. (a) When GAM queries a function (OpCode), R-CTRL responds with the initialization time required according to the status of the BS-Slots. (b) The state diagram of an F-RACC, arcs are annotated with times taken to change the state.

Table 5.1. Symbols for F-RACC's Reconfiguration Status

Symbol	Meaning
Tinit	The time required to initialize an accelerator
Tconfig	The time required to read a new bitstream into a BS-Slot. It is determined by the bitstream size and the programming techniques implemented in C-I/F.
Tconfig_multi	The total *Tconfig* while reading multiple bitstream into BS-Slots in parallel. The reading of each bitstream may start at different time (*Tconfig_multi* ≥ *Tconfig*)
Tswitch	The time required to activate another BS-Slot
Treset	The time which may be required to reset the currently activated accelerator function for next usage
Texe	The execution time taken to finish execution of a function. i.e., the latency of an F-RACC

F-RACC can be in one of five reconfiguration states as shown in Figure 5.4 (b). 1) An idle state (IDLE), when it is neither programming a new bitstream to a BS-Slot nor executing a function. 2) Configuration state (CONFIG), when F-RACC is reading at least one new

bitstream to a BS-Slot, but it is not executing a function. 3) Switch state (SWITCH), when R-CTRL is activating another BS-Slot without reading any new bitstream. 4) Execution state (EXE), when F-RACC is running a function, but it is not reading any new bitstream. 5) Background-configuration state (BG-CONFIG), when at least one new bitstream is being configured to a deactivated BS-Slot in the background while this F-RACC is simultaneously executing (EXE), switching (SWITCH), or configuring (CONFIG). Since each BS-Slot is independently configurable by a C-I/F, BG-CONFIG state can be entered from CONFIG, SWITCH, or EXE states any time as long as the F-RACC is not executing the function configured by the target BS-Slot. The progress of background tasks performed in BG-CONFIG state does not affect the progress of any concurrent foreground tasks because the background tasks independently operate only on those deactivated BS-Slots.

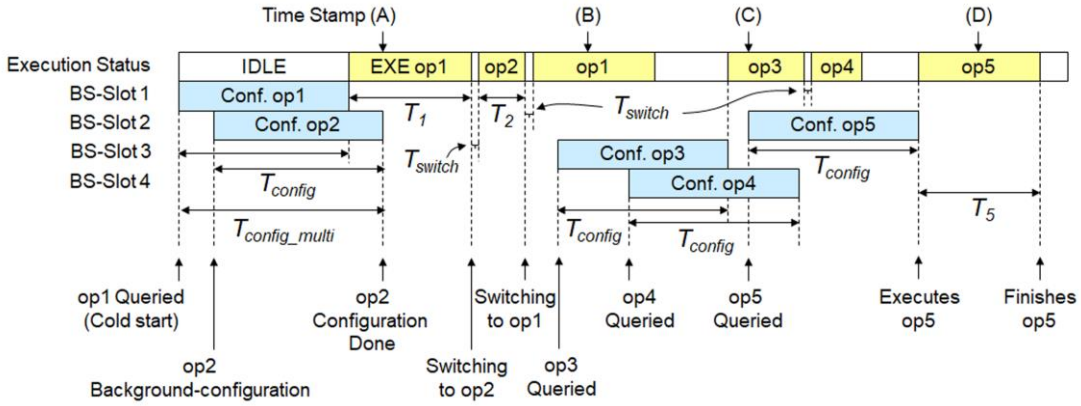
When system starts up, each F-RACC is in IDLE state. It moves to CONFIG state to read new bitstreams. Because an F-RACC can read multiple bitstreams in parallel with different starting times, it takes the configuration time *Tconfig_multi* or *Tconfig* to finish and leave CONFIG state. *Tconfig_multi* is the total overlapped *Tconfig* of programming all the BS-Slots reading new bitstreams. *Tconfig_multi* is equal or larger than *Tconfig*. After the configuration is done, the F-RACC may return to IDLE state or move to EXE state for starting an execution. In EXE state, it takes *Texe* to finish the function and return to IDLE state.

When there is a valid bitstream of the required function stored and ready in a BS-Slot, the F-RACC can leave IDLE state and enter EXE state 1) directly when this BS-Slot is active, or 2) through SWITCH state when the BS-Slot is inactive. For case 1), a time of *Treset* may be needed to reset the F-RACC for next usage. For case 2), a BS-Slot switching time *Tswitch* is required to change this F-RACC's function. Then, it goes back to IDLE state after the

computation is done in *Texe*. If background-configuration happens while in CONFIG, SWITCH, or EXE stage, the F-RACC jumps to BG-CONFIG stage and keeps running the original configuring, switching, or executing simultaneously. Once the background-configuration is done in *Tconfig_multi* or *Tconfig*, it returns to the original state (if the original tasks are still running) or IDLE state (if the background-configuration takes long enough time to finalize all the original tasks).

Figure 5.5 shows the timeline for an example of operating an F-RACC with four BS-Slots, starting from IDLE state with empty BS-Slots and BS-Table. At first, the bitstream of op1 is programmed into BS-Slot_1 for responding the query of op1. The bitstream of op2 is background-configured into BS-Slot_2 after a short time of starting programming op1. BS-Slot_1 is activated for execution. The F-RACC executes op1 after the configuration is done in *Tconfig*. At time stamp (A), the usage of op1 is updated to 1 and stored in the row 1 (Slot_1) of BS-Table. The F-RACC switches to execute op2 after the op1 computation time T_1 and function switch time *Tswitch*. After op2 is finished, it switches back to op1 again after *Tswitch* and executes it. Then, op3 and op4 are queried while F-RACC is executing op1. There are two empty BS-Slots, thus the bitstreams of op3 and op4 are now being programmed to BS-Slot_3 and BS-Slot_4. At time stamp (B), since op3 has been queried, the usage count of op3 is set to 1. After the second op1 finishes execution, the bitstream of op3 is still being programmed. It starts to execute op3 when the programming is done. Later, op5 is queried; however, there is no empty BS-Slot for this bitstream. Now R-CTRL checks the usage information stored in BS-Table at time stamp (C) to determine which bitstream to replace. At first, it searches for the least frequently used bitstream. If there are multiple bitstreams to choose, then it selects the one that has stayed longest in the BS-Slots. Thus, op2 in BS-Slot_2

is selected and is replaced by op5. Since the execution of op4 finishes earlier than the configuration of op5 is completed, the execution of op5 starts later after op5 configuration is done.



BS-Table Content in Each Time Stamp

(A)

Slot	Act.	Op	T_{exe}	# of Usage
1	Y	op1	T_1	1
2		op2	T_2	0
3				
4				

(B)

Slot	Act.	Op	T_{exe}	# of Usage
1	Y	op1	T_1	2
2		op2	T_2	1
3		op3	T_3	0
4				

(C)

Slot	Act.	Op	T_{exe}	# of Usage
1		op1	T_1	2
2		op2	T_2	1
3	Y	op3	T_3	1
4		op4	T_4	0

(D)

Slot	Act.	Op	T_{exe}	# of Usage
1		op1	T_1	2
2	Y	op5	T_5	1
3		op3	T_3	1
4		op4	T_4	1

Figure 5.5. The timeline and the BS-Table contents of a 4-BS-Slot F-RACC execution example. Tables (A) – (D) show the BS-Table contents in each time stamp marked on the timeline.

5.3.4 Performance Improvement Assessment

Performance of this architecture is mainly determined by the following factors: 1) how many computation tasks of an application can be offloaded to F-RACCs. We can expect better

performance with higher accelerator usage rate, if the accelerators are fast enough and called efficiently. 2) The cost of initializing a BS-Slot, i.e. the configuration time *Tconfig* of loading a new bitstream. This is an unavoidable cold start cost. 3) The probability that a required accelerated function is supported on time by at least one F-RACC in the system. This probability is similar to CPU cache hit rate. It is determined by three factors: a) the number of BS-Slots supported in an F-RACC, b) the number of F-RACC instances available in the system, and c) the accelerator's usage behavior of the application running on the system. In case c), if an application tends to call the same accelerating functions it has higher probability that the function queried is available on time. The match of BS-Slot replacement policy and application's behavior also affects this probability. Higher probability that a queried function can be accelerated shortly implies better system performance. 4) The speed of an accelerator, i.e., the latency to finish an accelerated function. 5) The time taken to activate another BS-Slot in an F-RACC, i.e. *Tswitch*.

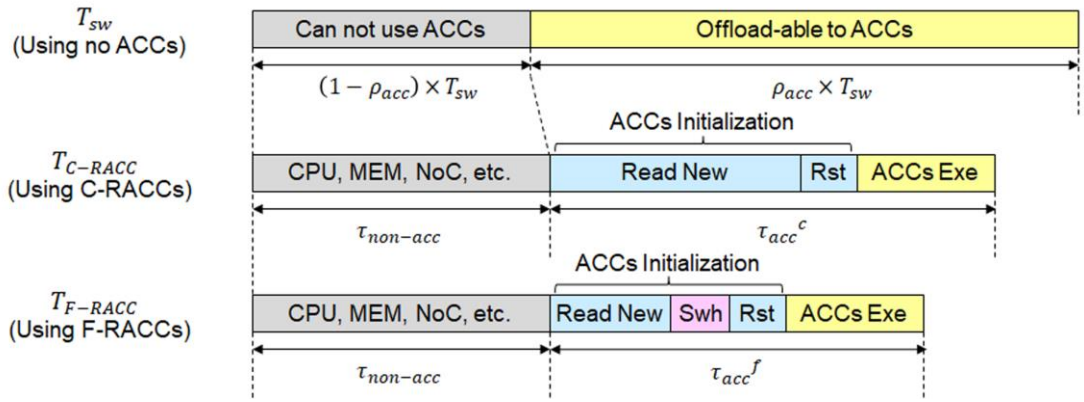


Figure 5.6. System performance breakdown for 1) no accelerators (ACCs), pure CPU software path, 2) using C-RACCs, and 3) using F-RACCs. The accelerator initialization portion can be in one of three states: reading a new bitstream; switching to another BS-Slot; and just resetting the accelerator for next coming usage, as described in Section 5.3.3, Figure 5.4 and Figure 5.5.

Table 5.2. Symbols for Performance Assessment

Symbol	Meaning
T_{sw}	The total time to execute an application using CPU software path only. It does not use any accelerator.
$T_{C\text{-}RACC}$	The total time to execute an application using conventional FPGA reconfigurable accelerators (C-RACCs)
$T_{F\text{-}RACC}$	The total time to execute an application using the proposed fast dynamic reconfigurable accelerators (F-RACCs)
ρ_{acc}	The percentage of an application that the computation tasks can be offloaded to accelerators ($1 > \rho_{acc} \geq 0$)
${\tau_{acc}}^{c}$, ${\tau_{acc}}^{f}$	The total time taken by accelerators (C- and F-RACCs)
$\tau_{non\text{-}acc}$	The total time taken by all the non-accelerated parts, such as CPUs, memory access, NoCs, etc
$N_{acc\text{-}qry}$	Number of accelerator queries ($N_{acc\text{-}qry} \geq 0$)
τ_{exe}	The average execution time to execute a function in an accelerator, i.e. accelerator's latency on average. It is the average among all accelerator calls in an application.
${\rho_{conf}}^{c}$, ${\rho_{conf}}^{f}$	The percentage of all accelerator queries which require a new bitstream configuration (C-RACCs: $1 > {\rho_{conf}}^{c} \geq 0$, F-RACCs: $1 > {\rho_{conf}}^{f} \geq 0$)
${\rho_{switch}}^{f}$	The percentage of all accelerator queries that the function called is ready in a BS-Slot to be activated ($1 > {\rho_{switch}}^{f} \geq 0$)
ρ_{reset}	The percentage of all accelerator queries that the function called is currently activated in an accelerator ($1 > \rho_{reset} \geq 0$)
τ_{conf}	The time required to read a new bitstream
${\tau_{switch}}^{f}$	The time required to activate another BS-Slot
τ_{reset}	The time which may be required to reset the currently activated accelerator function for next usage

Before we assess the performance improvement we illustrate the system performance breakdown in Figure 5.6. Table 5.2 provides the list of used symbols. T_{sw} is the time taken by an application running pure CPU software path with no accelerators. It has a portion (ρ_{acc}) of the computation tasks, which can be offloaded to accelerators, thus $\rho_{acc} \times T_{sw}$ is the portion offload-able to accelerators:

$$T_{sw} = (1 - \rho_{acc}) \times T_{sw} + \rho_{acc} \times T_{sw} \quad (5.1)$$

When accelerators are used, due to the extra communications required to control them, we have a larger total time taken by all the non-accelerated parts ($\tau_{non\text{-}acc}$) as shown in equation (5.2), such as CPUs, memory access, data transferring, NoCs, etc.

$$\tau_{non-acc} > (1 - \rho_{acc}) \times T_{sw} \tag{5.2}$$

The time taken by reconfigurable accelerators consists of initialization and execution. The initialization scenarios have been described in detail in Section 5.3.3. Here, we use the percentage ($\rho_{conf}{}^{c}$, $\rho_{conf}{}^{f}$, $\rho_{switch}{}^{f}$, ρ_{reset}) of total accelerator queries ($N_{acc\text{-}qry}$) to model the distribution of the scenarios. For the different implementations using C-RACCs and F-RACCs:

$$\rho_{conf}{}^{c} + \rho_{reset} = 1 \qquad (\mathrm{C-RACCs}) \tag{5.3}$$

$$\rho_{conf}{}^{f} + \rho_{switch}{}^{f} + \rho_{reset} = 1 \quad (\mathrm{F-RACCs}) \tag{5.4}$$

Because the calling sequences of the accelerated functions are the same in the application running on both C-RACC and F-RACC implementations, we can assume ρ_{reset} remains the same in both implementations, thus:

$$\rho_{conf}{}^{c} = \rho_{conf}{}^{f} + \rho_{switch}{}^{f} = 1 - \rho_{reset} \tag{5.5}$$

The average accelerator latency (τ_{exe}) is the average execution time taken to finish an accelerated function by an accelerator among all accelerator calls while executing an application. The total accelerators' execution time can be expressed as $\tau_{exe} \times N_{acc\text{-}qry}$. Assuming the two implementations take the same fixed time for each initialization scenarios (τ_{conf}, $\tau_{switch}{}^{f}$, τ_{reset}), we can have the total time taken by accelerators implemented by C-RACCs to be:

$$\tau_{acc}{}^{c} = \left(\rho_{conf}{}^{c} \times \tau_{conf} + \rho_{reset} \times \tau_{reset} + \tau_{exe}\right) \times N_{acc-qry} \quad (5.6)$$

$$T_{C-RACC} = \tau_{non-acc} + \tau_{acc}{}^{c} \quad (5.7)$$

And for using F-RACCs:

$$\tau_{acc}{}^{f} = \left(\rho_{conf}{}^{f} \times \tau_{conf} + \rho_{switch}{}^{f} \times \tau_{switch}{}^{f} + \rho_{reset} \times \tau_{reset} + \tau_{exe}\right) \times N_{acc-qry} \quad (5.8)$$

$$T_{F-RACC} = \tau_{non-acc} + \tau_{acc}{}^{f} \quad (5.9)$$

The time taken to activate another BS-Slot ($\tau_{switch}{}^{f}$) is generally much smaller than the time taken to read a complete new bitstream (τ_{conf}) as described in detail in Section 5.3.2.

$$\tau_{switch}{}^{f} \ll \tau_{conf} \quad (5.10)$$

Combining equations (5.5), (5.6), and (5.8), we can get:

$$\tau_{acc}{}^{c} - \tau_{acc}{}^{f} = \rho_{switch}{}^{f} \times \left(\tau_{conf} - \tau_{switch}{}^{f}\right) \times N_{acc-qry} \geq 0 \quad (5.11)$$

Thus, assuming the total time taken by all the non-accelerated parts ($\tau_{non-acc}$), is the same in both implementations, the speedup of using F-RACCs over using C-RACCs is shown in (5.12). The numerator in the equation is equation (5.11), which is equal or greater than zero. The two parameters in the denominator are both positive, thus the speedup (*Speedup*) is equal or greater than 1.

$$
\begin{aligned}
Speedup = \frac{T_{C-RACC}}{T_{F-RACC}} &= \frac{\tau_{non-acc} + {\tau_{acc}}^{c}}{\tau_{non-acc} + {\tau_{acc}}^{f}} \\
&= \frac{(\tau_{non-acc} + {\tau_{acc}}^{f}) + {\tau_{acc}}^{c} - {\tau_{acc}}^{f}}{\tau_{non-acc} + {\tau_{acc}}^{f}} \\
&= 1 + \frac{{\tau_{acc}}^{c} - {\tau_{acc}}^{f}}{\tau_{non-acc} + {\tau_{acc}}^{f}} \\
&= 1 + \frac{{\rho_{switch}}^{f} \times \left(\tau_{conf} - {\tau_{switch}}^{f}\right) \times N_{acc-qry}}{\tau_{non-acc} + {\tau_{acc}}^{f}} \geq 1
\end{aligned}
\tag{5.12}
$$

According to equations (5.11) and (5.12), the performance improvement of using F-RACCs over C-RACCs is determined by a) the percentage of initializing an F-RACC by BS-Slot switch, and b) the time differences of reading a new bitstream and switching BS-Slots. The larger a) and b) and more accelerator queries (larger $N_{acc\text{-}qry}$) the better. As for the improvement over pure CPU software path with no accelerators, the percentage of offload-able computing tasks (ρ_{acc}) mainly determines the scale of potential benefits accelerators can offer.

Figure 5.7 shows the speedup (*Speedup*) of using F-RACCs over using C-RACCs in different scenarios. Here, the switch time (${\tau_{switch}}^{f}$) is set to 1 as the base time unit. The number of accelerator queries ($N_{acc\text{-}qry}$) is fixed at 1000. The total time taken by all non-accelerated parts ($\tau_{non\text{-}acc}$) has two conditions, 1M and 100M time units. The time taken by reading a new bitstream (τ_{conf}) is 10,000 and 100,000 time units. The potential time taken by resetting an accelerator (τ_{reset}) is neglected and set to 0. Three conditions of accelerator's initialization cases are assessed, for ρ_{reset} = 20%: 1) high percentage of accelerator queries requires reading new bitstreams (${\rho_{conf}}^{f}$ = 70%, ${\rho_{switch}}^{f}$ = 10%), 2) half-half chance (${\rho_{conf}}^{f}$ = 40%, ${\rho_{switch}}^{f}$ = 40%), and 3) high percentage of accelerator queries that the required function is in one of the inactivate BS-Slots (${\rho_{conf}}^{f}$ = 10%, ${\rho_{switch}}^{f}$ = 70%). The proposed architecture provides higher

speedup when 1) reading a new bitstream costs more time, 2) higher chance a used function is called again (larger $\rho_{switch}{}^{f}$). It is intuitive that the speedup improvement diminishes when more time is taken by non-accelerated part (larger $\tau_{non\text{-}acc}$), since the performance benefit is contributed by accelerators.

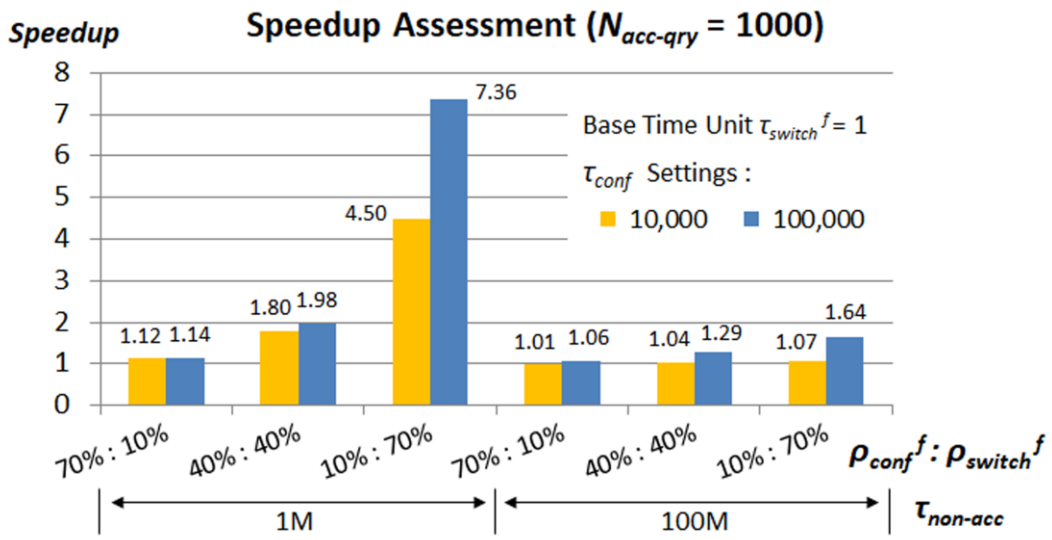


Figure 5.7. Speedup assessment of using F-RACCs over using C-RACC, based on 1000 accelerator queries ($N_{acc\text{-}qry}$). Setting the switch time $\tau_{switch}{}^{f}$ = 1 as the base time unit, for: a) τ_{conf} = 10,000 and 100,000, b) $\tau_{non\text{-}acc}$ = 1M and 100M, c) $\rho_{conf}{}^{f}$: $\rho_{switch}{}^{f}$ = 70%:10%, 40%:40%, 10%:70%, while ρ_{reset} is kept 20% and τ_{reset} = 0.

5.4 Experimental Results

Performance of the proposed architecture is evaluated using *PARADE* [63], a *gem5* [64] based cycle accurate full system simulator. *PARADE* simulator is designed for accelerator-rich architecture using fixed-function accelerators. In order to support the changing functions of F-RACCs, a portion of the codes was modified. The main modifications include: 1) instantiating F-RACC instances in the system instead of fixed-function dedicated accelerators, 2) providing a BS-Table and BS-Slots in each F-RACC, 3) implementing control logic in each

F-RACC, for monitoring the configuration status, changing accelerated functions, responding GAM the required F-RACC initialization time under different configuration cases, replacing the functions stored in BS-Slots, etc., and 4) BS-Slot replacement policy, LFU then LRU.

The basic specifications of the simulated x86 system are listed in Table 5.3. The system includes an 8-issue x86 out-of-order (OoO) CPU running at 2GHz. It has private 32kB two-way associate L1 I / D (Instruction / Data) cache. A 2MB eight-way associate L2 cache and four memory controllers of 512MB (total 2GB) 1600MHz DDR3 are shared by the CPU and all accelerators. Coherence is maintained by two-level MESI protocol. An operating system (OS) image of Linux kernel 2.6.22.9 is loaded; eleven benchmarks from four different applications are executed on this system.

Table 5.3. The Simulated x86 System

Item	Specification
CPU	One 8-issue x86 OoO core @ 2GHz
Coherence Protocol	2-level MESI
L1 I / D Cache	32 kB, 2-way associate, private
L2 Cache	2 MB, 8-way associate, shared
Memory	2GB (512MB x 4) 1600MHz DDR3
Operating System	Linux kernel 2.6.22.9

The benchmark applications are listed in Table 5.4 along with the number of different accelerating functions needed by each of them. These benchmarks are the original ones embedded in the source *PARADE* simulator [63] with no modifications. Detailed descriptions of these applications can be found in [19] and [63]. To get timing performance of each accelerator function implemented by conventional FPGAs, we run *Xilinx Vivado High Level Synthesis* (*HLS*) [81] to estimate their latency. A modern commercial Xilinx 7 series FPGA,

28nm Kintex-7 XC7K70T optimized for best price-performance, is the selected target device in *Vivado HLS*; it is the smallest device in Kintex-7 family.

Table 5.4. The Simulated Benchmarks

Domain	Application	Required # of Different Accelerating Functions
Medical Imaging	*Deblur*	4
	Denoise	1
	Registration	1
	Segmentation	1
Commercial	*BlackScholes*	1
	StreamCluster	5
	Swaptions	4
Computer Vision	*LPCIP_Desc*	1
Computer Navigation	*Robot_Localization*	1
	Disparity_Map	3
	EKF_SLAM	2

For a fairer comparison, we establish the baseline architecture, which uses conventional FPGA accelerators (C-RACC) with the fastest configuration speed. To estimate FPGA's configuration time *Tconfig*, we assume the fastest speed achievable in Xilinx FPGA. The configuration clock speed is 100MHz and the data width is 32 bits, thus the data rate is 3.2Gbps [62] [80]. Partial reconfiguration technique is applied; thus, an FPGA does not need to be completely reconfigured for a small accelerated function. We used a 100-CLB region in a Xilinx Virtex-5 FPGA to estimate the bitstream length using partial reconfiguration technique, which requires 236,160 bits [62]. This is the only number we could find from published documents; the information on 7 series in not available. Such region offers 800 6-LUTs, 800 Flip-Flops, 200 arithmetic and carry chains, 25,600 bits of distributed RAM, and 12,800 bits of shift registers [82]. It is representative for implementing accelerated functions in a reasonable scale. It takes 73.8μs to program a bitstream of length 236,160 bits with 3.2Gbps data rate [62], which for 2GHz core clock translates to 147,600 cycles.

Since F-RACC is a monolithic 3D design using VeSFETs, there is a speed improvement over 2D designs. Chapter 4 reports an 87% delay scaling factor of 3D VeSFET FPGA over 2D VeSFET FPGA. Because VeSFET is a new device in the early development stage, all the published performance results are based on untuned devices [28-29] [31] [34-36] [39-40]. It is hard to predict the performance of a well-tuned VeSFET. To conduct a fairer comparison in this work, we assumed that VeSFET has the same speed performance as CMOS. Thus, we calculated the latency of F-RACCs to be 87% (2D to 3D delay scaling factor) of the conventional FPGA results estimated by *Vivado HLS*. The BS-Slots switching time *Tswitch* is set to 5ns, i.e. 10 CPU cycles running at 2GHz, which is sufficient for a signal propagating through switching logic and moderate inter-tier wire length. The actual *Tswitch* delay estimation will be discussed in Section 5.4.1. The accelerator's potential reset time *Treset* is neglected and set to 0. The detailed conditions are listed in Table 5.5.

Table 5.5. Conditions Corresponding to Accelerators

Conditions	**C-RACC**	**F-RACC**
Tconfig	147,600 CPU cycles[†1]	147,600 CPU cycles[†1]
Tswitch	-	10 CPU cycles
Treset	0 CPU cycle	0 CPU cycle
Texe	Estimated by *Vivado HLS*	87% of C-RACC[†2]

[†1]: Using partial reconfiguration techniques with fastest configuration speed. Assuming same time required in C-RACC and F-RACC for reading a new bitstream externally into a BS-Slot.

[†2]: The 87% is the delay scaling factor of 3D VeSFET FPGA over 2D VeSFET FPGA, reported in Chapter 4.

5.4.1 Power and Area Overhead, and Switching Time Assessment

We performed circuit level simulation to estimate the power consumption and delay of switching BS-Slots. The VeSFET transistor models are based on compact current model [39]

and capacitance model [40]. The pillar radius *r* is 50nm and height *h* is 200nm, which is 65nm CMOS equivalent and is the same to the models used for standard cell characterization in [34] and for circuit simulation in Chapter 4. The metal wire structure is with width 0.1μm and spacing 0.1μm, which follows the design rules determined in [28] with pillar radius *r* = 50nm. We referenced a comparable 65nm CMOS BEOL technology file for estimating metal wire's thickness and the distance to the upper and lower metal layers. The metal thickness is set to 0.22μm and the vertical distances to adjacent metal layers are both 0.175μm. The metal wire resistance is calculated using copper resistivity of 1.678×10^{-8} Ω-m and the wire dimension. The metal wire capacitance is extracted by *Synopsys Raphael* 3D field solver under the worst scenario, i.e. the metal wire is fully covered by plates on the adjacent metal layers and has parallel long-running metal straps on the same layer with minimum spacing. The extracted metal resistance and capacitance for simulation are 0.763 Ω/μm and the 0.185 fF/μm, respectively.

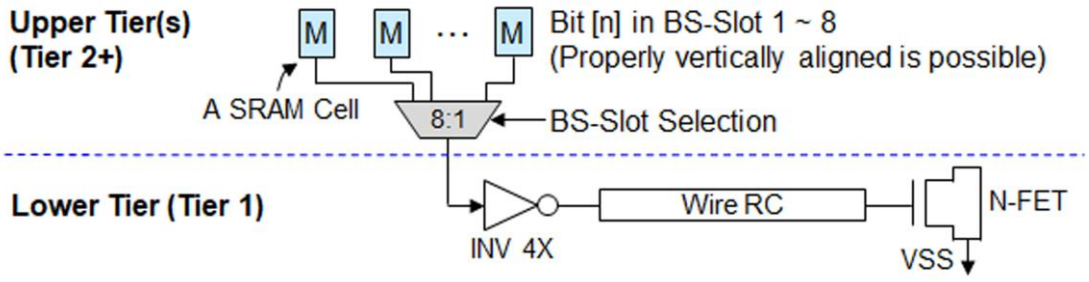


Figure 5.8. The circuit for delay and energy consumption simulations.

The circuit shown in Figure 5.8 was implemented for modeling the delay and power consumption per bit of switching 8 BS-Slots. It includes eight SRAM cells (8 independent bits from 8 BS-Slots) connected to a transmission gate type MUX8, the MUX8 is followed by a 4*P*4*N* inverter, which drives a long wire then connects to the gate terminals of an N-FET.

The N-FET models the transistor in the configurable element receiving the configuration signals from BS-Slots. The wire distance is determined by referencing the 4-LUT layout size in Section 4.3. For pillar radius r = 50nm, the size of a 4-LUT is $116r \times 128r$ = 5.8μm × 6.4μm for a 2D design with SRAM cells and is $72r \times 128r$ = 3.6μm × 6.4μm for a 3D design with SRAM cells placed in another tier. Since the BS-Slots, which are placed on other tiers, can be vertically aligned to the targeting configurable elements, the horizontal wire distance is small. We perform simulations with wire lengths 10 and 20μm, which is about 1.5x and 3x to the 4-LUT height. The *Hspice* simulation results are listed in Table 5.6.

Table 5.6. Delay and Energy Consumption of Switching 1 bit in 8 BS-Slots

Signal	**Wire Length = 10μm**		**Wire Length = 20μm**	
Toggling	**Delay (ps)**	**Energy (fJ)**	**Delay (ps)**	**Energy (fJ)**
0 → 1	92.09	1.370	104.21	2.580
1 → 0	78.37	0.652	88.06	0.831

The delay is from MUX selection signal toggling (50%) to the N-FET's gate rise to 80% or fall to 20%. The energy is per signal toggle.

The delay of distributing a bit from a BS-Slot to its destination is less than 105ps. It is much smaller than the 5ns *Tswitch* we assumed in this work. For the energy consumption of distributing 236,160 bits (100 CLBs), the total energy consumption is determined by how many signals toggle. If we assume the probability of each scenarios are all 25%, i.e. 0 → 1, 1 → 0, 0 → 0, and 1 → 1, there are 59,040 bits for each case. For those non-toggled signals, we neglected the energy consumption. Using the energy consumption per bit listed in Table 5.6 and the bit counts, the total energy consumption of distribution all 236,160 bits of switching 8 BS-Slots are 119pJ and 201pJ for wire length 10 and 20μm, respectively. If such energy is consumed in 5ns, the power consumptions are 23.9 and 40.3mW, respectively. This is still in

an acceptable range. To reduce the power consumption, it is feasible that the designers use a longer *Tswitch* and switch only a portion of the BS-Slot in a time.

The area trade-off is mainly from the extra storage elements required by BS-Slots. SRAM cells or latches can implement such storage elements. Section 2.3 assessed VeSFET SRAM designs and compared them to 65nm CMOS SRAMs. A 65nm-equivalent VeSFET 6T SRAM cell shown in Figure 2.7 (VeSFET cell layout *v2*) is $20r \times 16r = 320r^2$, it is $1\mu m \times 0.8\mu m = 0.8\mu m^2$ when the pillar radius $r = 50nm$. Thus, the total area for 236,160 bits (the region of 100 CLBs) is $188,928\mu m^2 \approx 434.66 \times 434.66\mu m^2$. The leakage currents of VeSFET SRAM cells are listed in Table 2.2, for the cell with decent read current, the leakage current is 10.85pA per cell. Under the conditions of nominal VDD = 0.8V and 236,160 cells, the total leakage power consumption is 2.05μW.

5.4.2 Performance Assessment

As discussed in Section 5.3.4, the system performance improvement of using F-RACC comes from the efficiency of switching accelerator functions. Providing more BS-Slots means more functions can be supported efficiently by an F-RACC. Intuitively, performance improvement saturates when the number of BS-Slots is sufficient to support all different accelerated functions called by an application. This saturation phenomenon is shown in Figure 5.9 with six applications using different number of accelerated functions (marked in the parentheses) and 1 F-RACC instance only. The value on *y*-axis is the normalized total cycle count taken to finalize the applications; the total cycle includes CPU processing, memory accesses, accelerator operations, data transferring, etc. They are normalized to the cycle counts taken with using 1 BS-Slot of each application, which is conceptually equivalent to C-RACC. With the number of BS-Slots increasing on *x*-axis, the cycle count reduces and saturates after

the number of BS-Slots equals to the number of accelerated functions required. The steepest cycle count drops happen just at the time before saturation, for example, *Swaptions*'s number of BS-Slots grows from 3 to 4. This is because the number of BS-Slots exactly matches the required function count, and eliminates all the costs of reading new bitstreams except the cold start unavoidable cost.

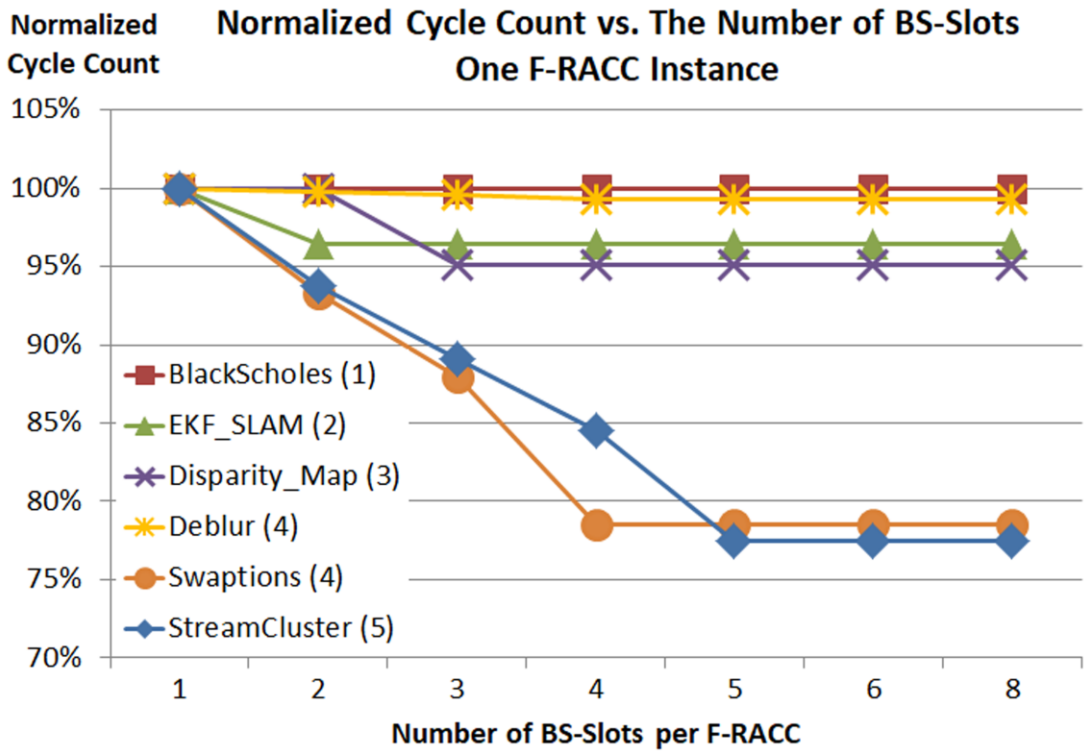


Figure 5.9. Cycle count reduction by providing more BS-Slots in an F-RACC. In "( )" we show the number of different accelerated functions required for that benchmark application. The reduction saturates when the number of BS-Slots is sufficient for supporting the number of different accelerated functions required.

To observe the system-level performance improvement, the simulations of the following conditions were performed:

1. Pure CPU software path (CPU-SW), no accelerators.
2. Using C-RACCs

3. Using F-RACCs, with the number of BS-Slots 2, 3, 4, 5, 6, and 8.
4. For 2) and 3), change the number of accelerator instances to be 1, 2, 3, 4, 6, 8, and 12.

Table 5.7. F-RACC Over C-RACC Speedup

Application[†1]	# of ACCs	# of BS-Slots[†2]			
		2	3	4	5
Deblur (4)	1	1.004	1.01	1.01	-
	4	1.01	1.01	1.03	-
	12	1.02	1.03	1.08	-
StreamCluster (5)	1	1.08	1.14	1.20	1.31
	4	1.14	1.24	1.64	1.92
	12	1.11	1.24	1.79	2.31
Swaptions (4)	1	1.08	1.15	1.29	-
	4	1.11	1.14	1.87	-
	12	1.12	1.28	2.82	-
Disparity_Map (3)	1	1.003	1.05	-	-
	4	1.05	1.23	-	-
	12	1.13	1.64	-	-
EKF_SLAM (2)	1	1.03	-	-	-
	4	1.22	-	-	-
	12	1.33	-	-	-

[†1]: The number of different accelerated functions called is noted in "( )".
[†2]: For conciseness, we show only the results up to the number of BS-Slots equal to the number of different accelerated functions required.

Table 5.7 shows the speedup of systems using F-RACCs over using C-RACCs (1, 4, and 12 accelerator instances), swept by the number of BS-Slots up to the functions required by the application. To make the figure concise, we show only the benchmark applications using more than 1 accelerated function, which is the case that F-RACCs contribute performance improvement by function switching. The result of *Swaptions* with 4 BS-Slots and 4 accelerator instances is exponentially interpolated by the adjacent results of using different number of accelerator instances with 4 BS-Slots, which is sufficient for predicting the trend of performance behaviors. This is because *PARADE* failed to simulate this configuration point of *Swaptions*. The improvements are discrete, which is mainly due to application's accelerator

usage behaviors and the percentage of accelerate-able computing tasks as discussed in Section 5.3.4. For 1 and 12 accelerator instances with sufficient number of BS-Slots, the speeds of all benchmark applications are improved and reach maximum speedup 1.31x and 2.82x respectively. The benchmark applications are all the original ones embedded in the source *PARADE* simulator [63], which are not optimized for this reconfigurable architecture. Higher speedup is possible by optimizing applications' accelerator usage behaviors.

Table 5.8. Speedup over CPU Software Path

Application[†1]	**C-RACC**			**F-RACC**[†3]		
	[†2] **1**	**4**	**12**	**1**	**4**	**12**
Deblur (4)	3.66	13.58	36.10	3.69	14.01	39.16
Denoise (1)	4.04	14.24	27.20	4.13	15.95	34.55
Registration (1)	2.21	8.29	22.85	2.22	8.34	23.20
Segmentation (1)	25.15	96.95	270.50	25.25	97.36	271.82
BlackScholes (1)	29.87	108.82	257.24	32.53	116.76	268.14
StreamCluster (5)	5.23	11.05	13.30	6.84	21.18	30.75
Swaptions (4)	73.77	163.17	200.48	94.93	305.29	565.12
LPCIP_Desc (1)	12.83	49.48	125.66	14.29	54.98	135.67
Robot_Loc. (1)	11.92	38.71	50.67	11.98	38.63	50.62
Disparity_Map (3)	3.86	10.94	21.25	4.06	13.50	34.82
EKF_SLAM (2)	9.18	27.70	36.52	9.46	33.81	48.45

[†1]: The number of different accelerated functions called is noted in "( )".
[†2]: The number of accelerator instances provided in the system.
[†3]: There are 8 BS-Slots per F-RACC instance, which guarantees sufficient support for the number of required accelerated functions.

The speedup over pure CPU software path is shown in Table 5.8. All eleven benchmark applications are included with accelerator instances 1, 4, and 12. For F-RACC, the number of BS-Slots is 8, which guarantees sufficient support for the number of accelerated functions required. By offloading computing tasks to accelerators, the speed can be greatly improved on all applications. There are several benchmarks requiring only one accelerated function. In these cases, the speed improvement difference between using F-RACCs and C-RACCs is mainly contributed by accelerators' latency (*Texe*) reduction from 2D to 3D FPGA

implementation. The maximum speedup 94.93x and 565.12x are observed on *Swaptions* when the system uses 1 and 12 accelerator instances respectively. A reference point of using dedicated specialized accelerator is reported in [63], where *Swaptions* reaches 130x speedup over CPU software path. This application computes swaption prices using Monte Carlo simulation of 8K datasets. The 130x speedup achieved by using dedicated fixed-function accelerators implies a great portion of the application is offloaded to accelerators. The 94.93x speedup achieved by using only 1 F-RACC instance is reasonable since the FPGA-based accelerator has larger latency than the fixed-function specialized accelerators and there exists unavoidable configuration overhead.

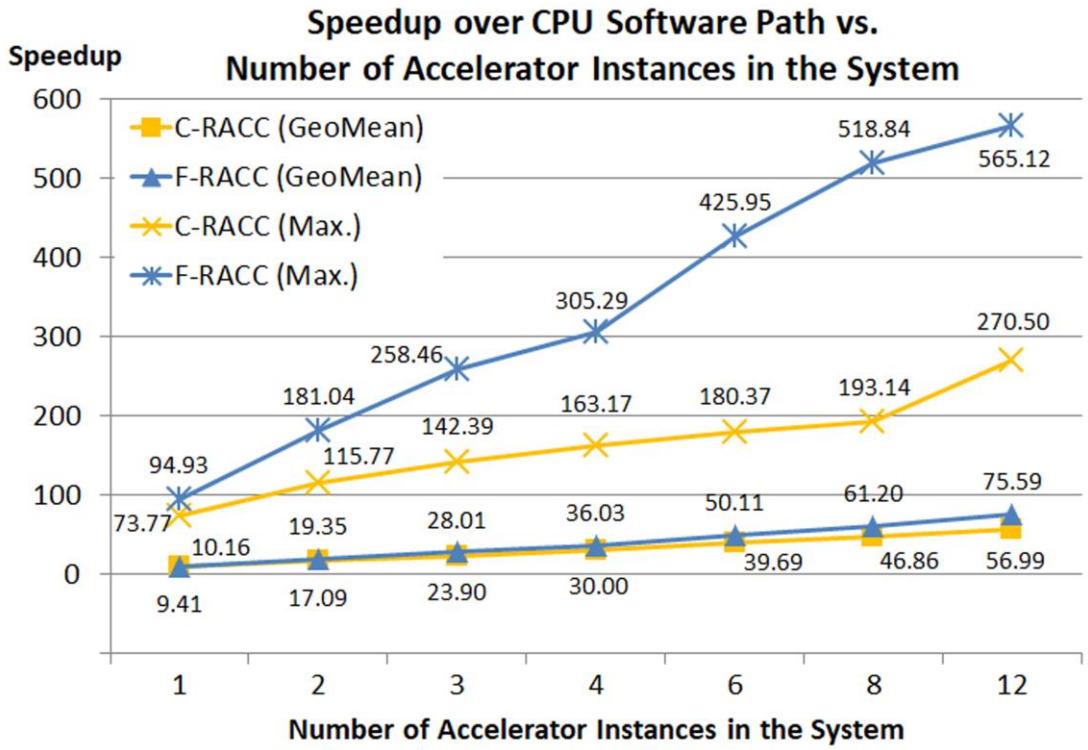


Figure 5.10. The geometric mean and maximum of system speedup over CPU software path among 11 benchmark applications with different number of accelerator instances. There are 8 BS-Slots per F-RACC.

The geometric mean and maximum system speedup over CPU software path among 11 benchmark applications with different number of accelerator instances are shown in Figure 5.10. Each F-RACC has 8 BS-Slots, thus the results of F-RACC is with the saturated fastest speed. Among all applications, the geometric mean of F-RACC speedup achieved 10.16x and 75.59x with 1 and 12 accelerator instances, respectively. It is intuitive that the speedup increases by providing more accelerator instances, which increases the probability that there exists an available accelerator instance to fulfill the request from the application. This effect applies to both C-RACC and F-RACC implementations. In addition, in F-RACC case, the increased number of accelerator instances increases the probability that a requested accelerated function can be implemented shortly by BS-Slot switching. This increases the $\rho_{switch}{}^{f}$ shown in equation (5.12) and increases the speedup of F-RACC over C-RACC implementations.

5.5 Summary

We proposed a high performance accelerator-rich architecture using fast dynamic reconfigurable accelerators (F-RACCs), which provide flexibility to the system for handling different workload demands in wide spectrum, such as data center or cloud computing systems. The traditional accelerator-rich architecture with a sea of fixed-function specialized accelerators may be impractical when abundant accelerators are required to satisfy wide spectrum applications.

The F-RACCs are implemented by monolithic 3D FPGAs using VeSFETs, which are two-side accessible transistors. With the support of emerging technologies, storing multiple bitstreams and fast distributing them in parallel are both feasible. Thus, the bitstreams can be dynamically switched efficiently. Comparing with the systems using conventional FPGA

accelerators (C-RACCs), which are supported by partial configuration techniques with fastest programming speed, the performance evaluation of 11 benchmark applications shows this architecture improved speed on all applications and achieved 1.31x and 2.82x (using 1 and 12 accelerator instances) maximum speedups. Comparing with the time taken by running on pure CPU software path without any accelerators, this architecture achieves geometric means 10.16x and 75.59x, maximum 94.93x and 565.12x using 1 and 12 accelerator instances, respectively.

The results of this work are promising; they suggest a possible direction with potential for future high performance dynamic reconfigurable architectures. The proposed implementation uses emerging technologies of monolithic 3D integration and a novel device VeSFET. These technologies are still in initial development stages. Further research and technology development are required for realizing this architecture and for better assessing the performance benefits and design tradeoffs.

Chapter 6

MAKING SPLIT-FABRICATION MORE SECURE

Today many design houses must outsource their design fabrication to a third party which is often an overseas foundry. Split-fabrication is proposed for combining the Front End of Line (FEOL) capabilities of an advanced but untrusted foundry with the Back End of Line (BEOL) capabilities of a trusted foundry. Hardware security in this business model relates directly to the front-end foundry's ability to interpret the partial circuit design it receives in order to reverse engineer or insert malicious circuits. The published experimental results indicate that a relatively large percentage of the split nets can be correctly guessed [26] and there is no easy way of detecting the possibly inserted Trojans.

In this chapter, we propose a secure split-fabrication design methodology for the Vertical Slit Field Effect Transistor (VeSFET) based integrated circuits. We take advantage of the VeSFET's unique and powerful two-side accessibility and monolithic 3D integration capability. In our approach the design is manufactured by two independent foundries, both of which can be untrusted. We propose the design partition and piracy prevention, hardware Trojan insertion prevention, and Trojan detection methods. In the 3D designs, some transistors are physically hidden from the front-end *foundry_1*'s view, which causes that it is impossible for this foundry to reconstruct the circuit.

6.1 Introduction

Due to the continuously increasing cost of chip manufacturing, maintaining a self-owned foundry is no longer an option for many design companies. The vast majority of integrated circuit (IC) companies follow the fabless manufacturing business model to save substantial capital and operating costs. In the globalized semiconductor industry each step of the chip building process including design, verification, manufacturing, testing, and packaging can be outsourced to an individual third party, which is often overseas. This trend leads to hardware security and trust vulnerabilities since the third party has access to some of the design secrets related to the services it offers. The main threats are: design piracy, IC overbuilding, hardware-based Trojan insertion, side channel attack, counterfeit ICs, and reverse engineering [83]. To regain a trustworthy design, many countermeasures have been proposed for different threats, such as split-fabrication, logic encryption, physical unclonable functions (PUFs), design obfuscation, IC camouflaging, etc. [83] [84].

During the IC design and production stages, the foundry has complete access to the final GDSII files including all the physical information of the chip, has the full control of the fabrication, and has the testing vectors and test plans. An untrusted foundry is a critical source of security vulnerabilities. It can reconstruct the design by physically inspecting the transistors and connections, which may lead to design piracy and counterfeit concerns. It can overbuild the ICs and sell them to black markets. It can even modify the design and insert malicious circuits and Trojans to hurt reliability and performance, steal information, create additional operating modes, take over the system's control, etc. To prevent the threats from an untrusted foundry, split-fabrication has been proposed [23-25]. It combines the Front End of Line (FEOL) capabilities of an advanced but untrusted semiconductor foundry with the Back End

of Line (BEOL) capabilities of a trusted semiconductor foundry. The transistors and the lower metal layers are manufactured as the FEOL process by an untrusted *foundry_1*, and then BEOL is processed by a trusted *foundry_2* to finalize the chip. In this scheme, *foundry_1* does not have the complete information about the chip, which makes it difficult for this foundry to reconstruct the design. However, *foundry_1* may exploit the heuristics used in floorplanning, placement, and routing CAD tools to make good guesses of net connections. Reference [26] demonstrates that a substantial portion of the missing BEOL connections could be reconstructed by the proximity attacks.

Besides the threats of design reconstruction, design modifications which include inserting malicious circuits and hardware-based Trojans are possible by the untrusted *foundry_1*. Trojans and malicious circuits are critical because they could change the chip's functions, behaviors, performance, durability, etc. Many works to prevent and detect Trojans have been proposed [85-97]. The detection methods can be generally categorized into three types: 1) logical test [88] [89]; 2) side-channel analysis, such as delay measurement [90-93], current sensing [94], thermal and power map [95], temperature tracking [96]; and 3) reverse engineering [97]. However, these techniques need additional efforts of test pattern generation, introduce circuit overhead, require precise measurements, or perform destructive and costly reverse engineering.

In this work, we propose a secure split-fabrication for Vertical Slit Field Effect Transistor (VeSFET)-based ICs. We propose the design flow that addresses vulnerabilities that exist in conventional approaches. The scheme outlined here does not require a trusted foundry. The proposed methodology prevents the untrusted foundries from reconstructing the design, prevents piracy and Trojan insertion. It also provides methods for Trojan detection. This

methodology leverages the VeSFET's two-side accessibility property [28] [35-36], which provides great benefits in wire connection and monolithic 3D integration. In a monolithic VeSFET 3D chip designed with the proposed methodology, some transistors are physically hidden from the FEOL *foundry_1*'s view, which causes that it is impossible for this foundry to reconstruct the circuit. In contrast to the conventional split-fabrication approach, in which most of the wires are connected by the trusted *foundry_2*, in the methodology we propose most of the wires are made by *foundry_1*. Only a very small fraction of the wires is made by *foundry_2*. The existence of a Trojan can be detected by static current: if any of the existing transistors is shifted or any extra transistors are added, a huge VDD-VSS crowbar current flows and can be easily detected. There is no extra circuit required to detect Trojans.

We propose algorithms for partitioning the wires to be made by two foundries, hiding the transistors in a 3D chip case, and camouflaging the space created by the hidden transistors. 10 MCNC LGSynth'91 benchmark circuits were designed by applying the proposed flow and then an attack was executed by the in-house developed proximity attacker with the objective to reconstruct the missing wire connections by *foundry_1*. With 5% nets manufactured by *foundry_2*, the average percentage of the correctly reconstructed partitioned nets is less than 1%.

This chapter is organized as the following. Section 6.2 describes the possible security threats of the split-fabrication method. Section 6.3 describes the complete methodology and Section 6.4 provides implementation details. The experiment settings and security assessment results are shown in Section 6.5. Section 6.6 summarizes this chapter.

6.2 Security Threats

For the chips made by conventional spilt-fabrication methods, the design is divided into two parts to be manufactured by independent foundries. The IC design house provides two GDSII files: 1) GDSII-1 for the untrusted *foundry_1* which includes the information of all transistors, floorplan, placement, a fraction of interconnects, all the pins' interconnect shapes on lower metal layers, and the location of empty space; 2) GDSII-2 for the trusted *foundry_2* which contains only the remaining interconnects and the elements required for packaging. *Foundry_1* could use the provided information to make a reasonable guess about the missing interconnects and then reconstruct a portion or even the complete design. Figure 6.1 shows the flow and the possible attacks by the untrusted *foundry_1*. The design reconstruction may lead to the threats of design secret exposure, piracy concern, chip counterfeit, and IC overbuilding.

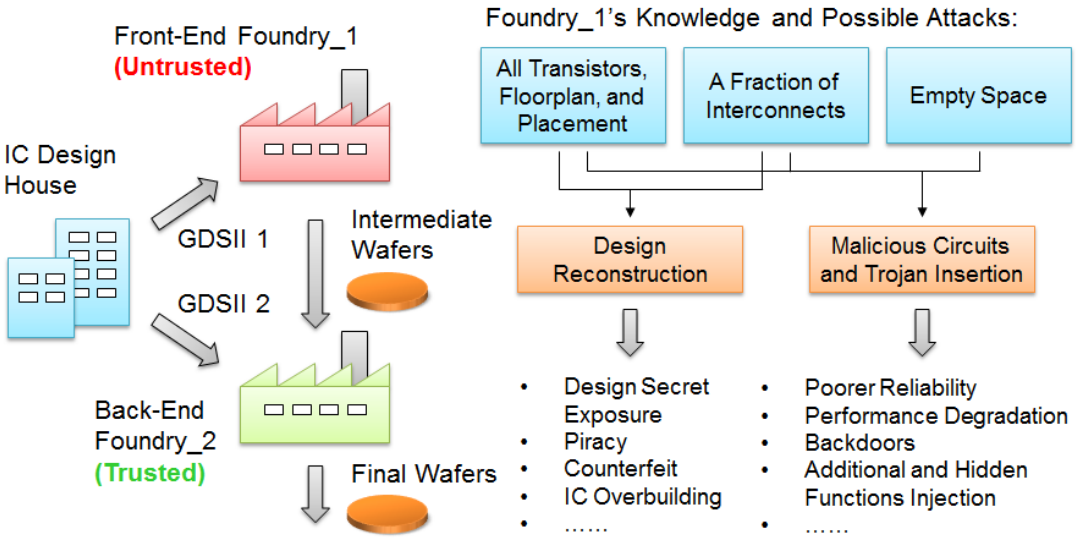


Figure 6.1. Security threats in a conventional split-fabrication method

Usually, a portion of a chip is occupied by decoupling capacitors (DeCaps) or is intentionally left as empty space due to design considerations such as IR drop, signal integrity, design rule requirements, etc. *Foundry_1* can use this knowledge to insert additional circuits and modify the chip's functionality without the design house's awareness. This kind of attack injects malicious circuits or Trojans to the chip for the purpose of degrading reliability or performance, creating a backdoor for remotely taking over the control, adding hidden functions, etc.

To mitigate security threats of the split-fabrication method and to create a trustworthy design, it is essential 1) to make the design reconstruction very difficult, and 2) to have easy and effective methods of Trojan prevention and detection. The conventional split-fabrication method still requires a trusted *foundry_2* to connect most of the wires to increase design reconstruction difficulty for the untrusted *foundry_1*. The requirement of using one trusted foundry reduces the IC design houses' freedom of selecting the foundry. It costs extra efforts to prove a foundry trustable and maintain the relationship. However, for very sensitive designs such as those for military applications, none of the foundries should be assumed trusted.

6.3 The Proposed Methodology

In contrast to conventional transistors such as MOSFET, FinFET, SOI, etc., VeSFET's terminals can be accessed from both top and bottom of the pillars. This characteristic leads to the possibility of two-side routing and offers a friendly monolithic 3D integration [28] [35-36]. In a VeSFET-based design, pillars naturally create numerous vertical connection channels and thermal dissipation paths. We utilize this unique characteristic and propose this secure split-fabrication design methodology.

6.3.1 Methodology for 2D Designs

With the powerful and unique two-side accessibility of VeSFET's transistor terminals, *foundry_2* can access every transistor directly at the backside of a VeSFET array, which highly increases the freedom of net partitioning for split-fabrication. In a conventional split-fabrication flow, either *foundry_1* or *foundry_2* sees the complete transistor connections on Metal 1. In a VeSFET-based design, the nets (interconnects) can be partitioned such that either foundry only sees a portion of the transistor connections on Metal 1 layers (front and back sides). In a 2D chip design as shown in Figure 6.2, all the devices and the majority of interconnects are manufactured by *foundry_1*, only a small portion of interconnects is selected to be made by *foundry_2* on the backside.

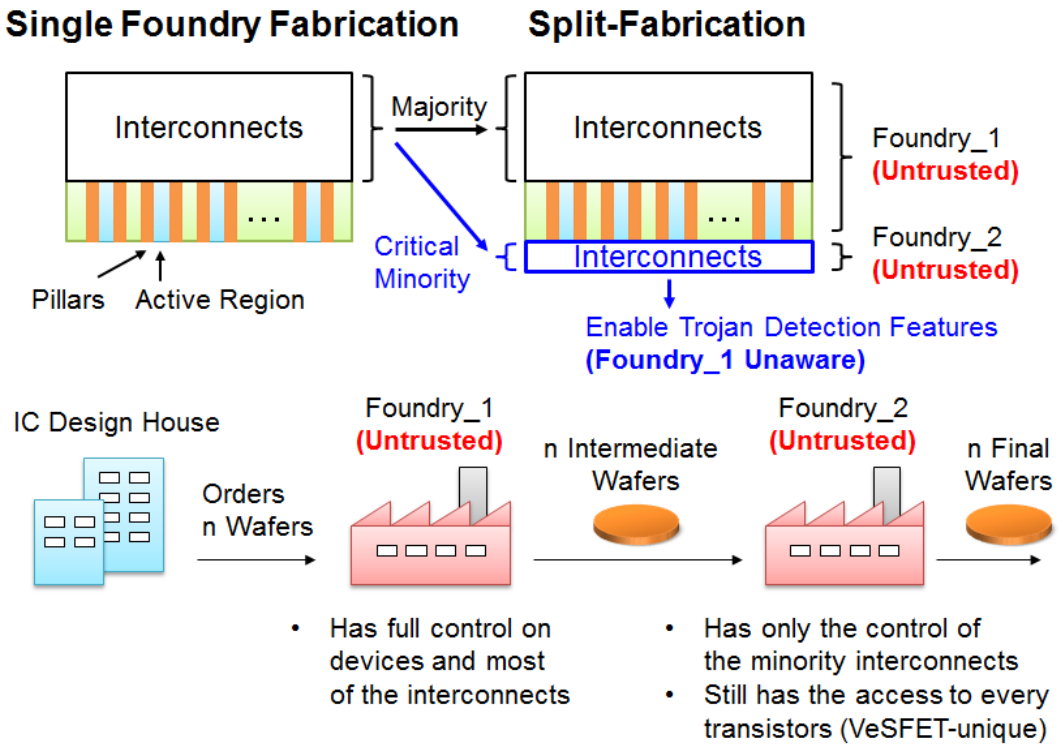


Figure 6.2. The overview of VeSFET-based split-fabrication method of a 2D design

The net partition guidelines are: 1) it should be difficult for *foundry_1* to reconstruct the design, and 2) as few as possible nets should be manufactured by *foundry_2*. The majority of interconnects will be made by *foundry_1*, because 1) it makes the chip performance more predictable, and 2) to limit *foundry_2*'s information since it finishes the chip. Besides the functional nets, *foundry_2* creates interconnects which enable Trojan detection features and are hidden from *foundry_1*.

Figure 6.3 outlines the Trojan detection method. Elements of the transistor array can be categorized into three groups: 1) functional transistors, 2) DeCap transistors, and 3) empty space which may consist of unformed transistors or formed unused transistors. The functional transistors are connected to implement the chip's function. The DeCap transistor gate, source and drain terminals connect to VDD / VSS power and ground rails. The empty space in a VeSFET design is special because the transistors are fully aligned as arrays. The pillar locations predefine the possible transistor locations; the pillars of a transistor exist even if a transistor is not formed. *Foundry_2* creates 1) properly selected small fraction of functional interconnects, 2) extra VDD and VSS interconnects at the backside of the DeCap transistors, and 3) the connections to VSS for all the remaining pillars not associated with any useful transistor; all three types of connections are unknown to *foundry_1*. If the design is attacked by *foundry_1*, any modification results in either functional interconnect short (front and back sides via pillars) or short to VDD or VSS. Large crowbar current can be measured and indicate *foundry_1*'s attack. There is no area or power overhead for this Trojan detection scheme because it does not require any circuit. An extra benefit of those VDD / VSS interconnects fabricated by *foundry_2* is the increased power network capacitance, which mitigates power bouncing when chip operates.

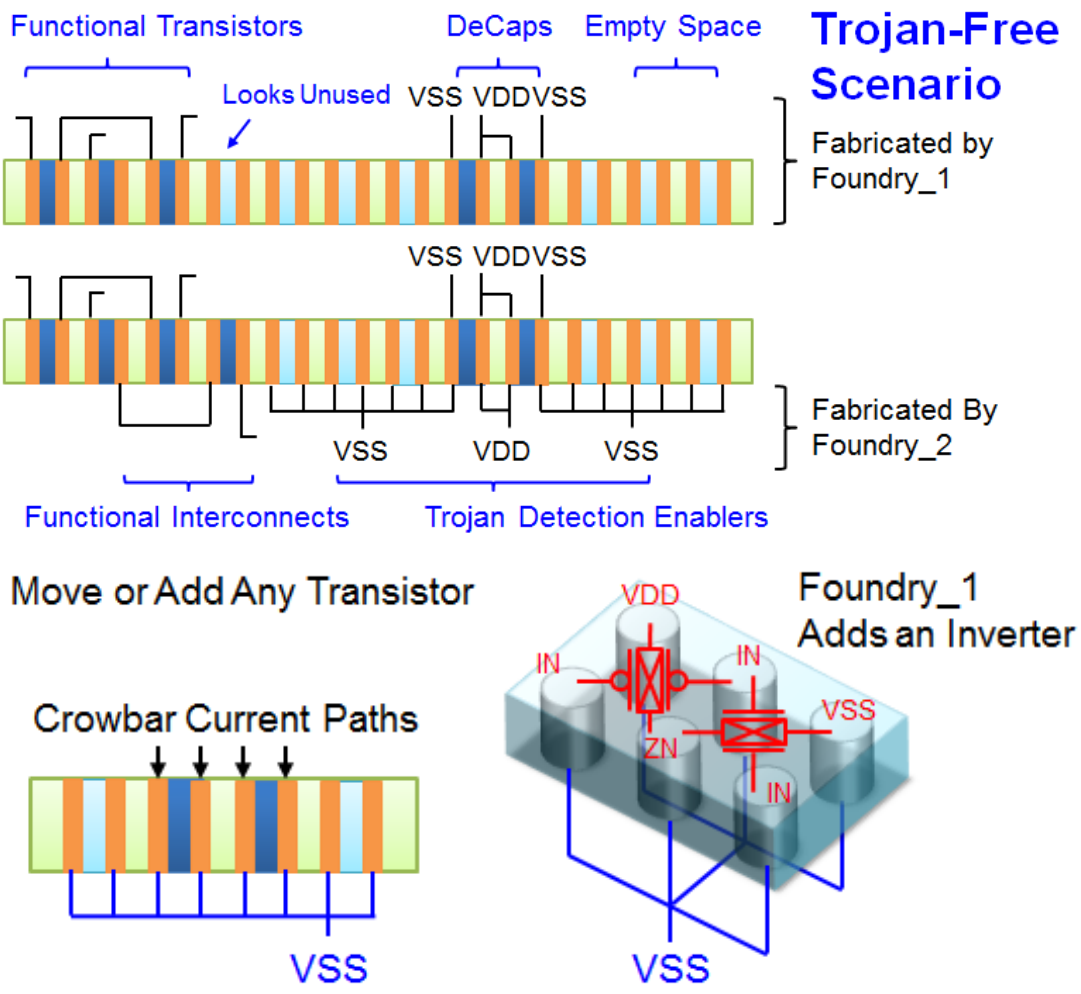


Figure 6.3. Trojan detection in a 2D chip

6.3.2 Methodology for 3D Designs

To leverage the existing 2D EDA tools for 3D designs, standard cell structures can be designed as shown in Figure 6.4. The VeSFETs are fabricated as arrays, in which all the transistors locations are predefined by the pillar positions. Thus, the pillars and the possible transistor locations are vertically aligned, and the locations of used and unused transistors are easily identifiable. A cell is designed with the same footprint on multiple tiers which are directly vertically connected. By controlling the existence of inter-tier connections, the

equivalent transistor width can be adjusted by the total pillar height *h* connected. The pillar height *h* of each tier could be different. A cell can be designed using different transistor pillar height *h* combinations to mitigate channel width quantization effect.

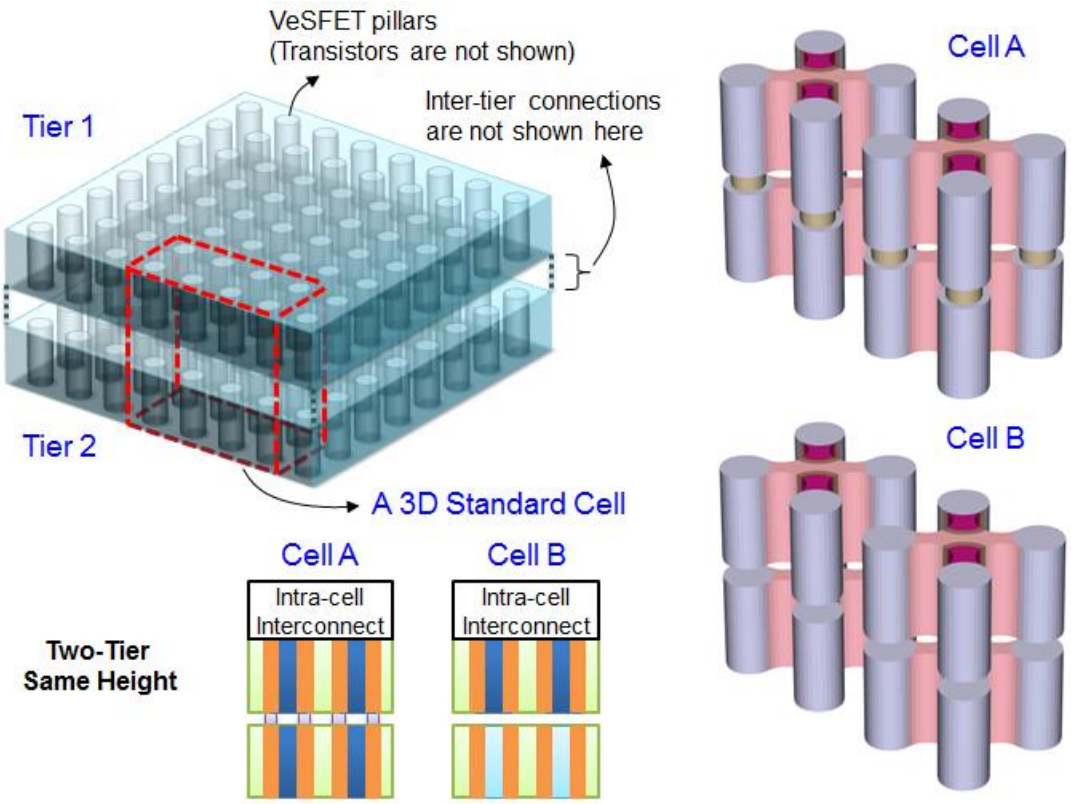


Figure 6.4. VeSFET-based 3D standard cells, design feasible with conventional 2D EDA tools

In a 3D design, besides the security features that exist in the 2D design, a portion of transistors in the lower tiers could be hidden and invisible to *foundry_1*. Figure 6.5 shows four inverters with the same footprint but different characteristics. As seen by *foundry_1*, only INV2X fully uses all the transistors; in the other implementations some transistors are unused. *Foundry_2* can use these leftover transistors to build functional circuits by creating backside interconnects. The original circuit can be partitioned such that a portion of it is constructed from these hidden unused transistors positioned under the existing circuits. When the hidden

transistors are used, *foundry_1* cannot reconstruct the design because a portion of the design is invisible to it. Although *foundry_2* may understand the functionality of these extra circuits, these standalone circuits do not provide *foundry_2* with enough information to reconstruct the design. Figure 6.6 shows the split-fabrication method for a 3D design; the transistors A, B, and C are unused by *foundry_1* and hidden from its view. They will be used by *foundry_2* to build some functional circuits.

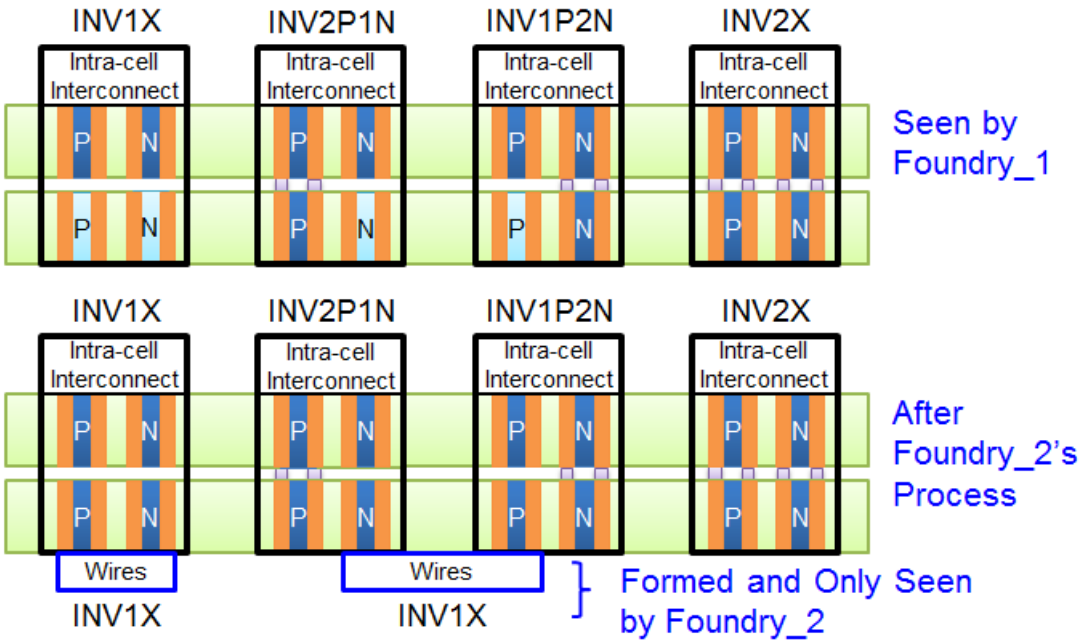


Figure 6.5. Extra available circuits constructed by *foundry_2*, invisible to *foundry_1*

The Trojan detection method in a 3D chip is similar to the method used in 2D chips but with an extra Trojan Detection Scan Path (*TD_Scan*). Since there are multiple tiers in the 3D chip, *foundry_1* may only change the upper transistor tiers. This change breaks the crowbar current path (front to back sides). Figure 6.7 shows the Trojan detection methods for a 3D chip. The *TD_Scan* path chains the pillars of non-functional transistors and prevents any modification on the design. Two attack scenarios are possible. In scenario 1, *foundry_1* moves

or adds transistors but is unaware of *TD_Scan*; these changes break the *TD_Scan* path or form crowbar current paths that could be easily detected. In scenario 2, *foundry_1* only changes the upper tiers of the design knowing that *TD_Scan* exists, and it tries to reconnect the scan path. This attack could be detected by measuring the active current change on *TD_Scan*. Details of the *TD_Scan* are discussed in Section 6.4.3.

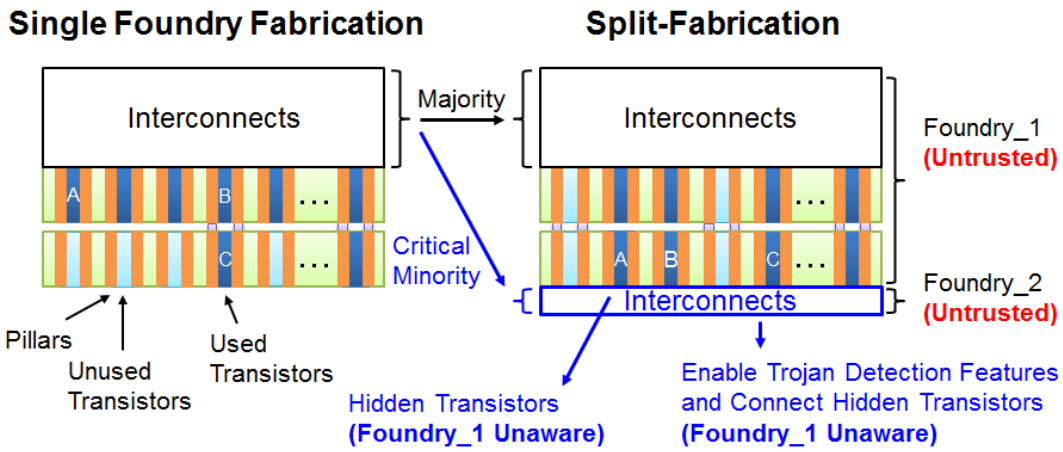


Figure 6.6. VeSFET-based 3D chip split-fabrication

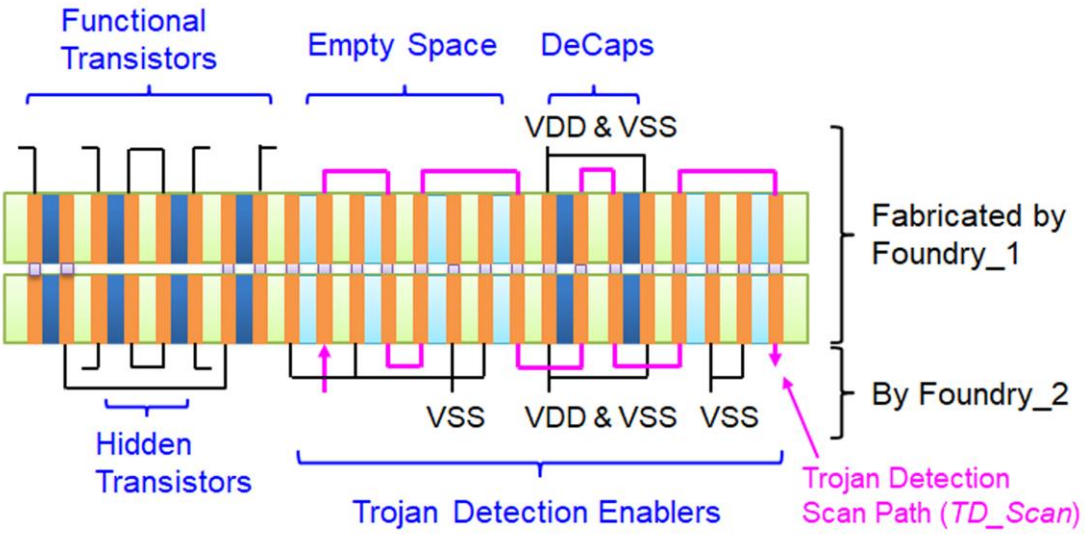


Figure 6.7. Trojan detection in a 3D chip

To enhance the overall security, logical encryption was proposed [98-100]. The idea is that a valid key must be provided to correctly activate the chip. A simple implementation with Flip-Flops (FFs) and XORs is shown in Figure 6.8, the function *F'* is determined by the *KEY* provided by the user and stored in the *KEY-FF*. In this work, we consider logical encryption as one of security features, but it is not necessary for the proposed methodology. The *TD_Scan* path can share the *KEY-FFs* to save area or it can use dedicated FFs if the design is not logically encrypted. After the scan detection, the *KEYs* are scanned through the same path to decrypt the chip. If *TD_Scan* shares the *KEY-FFs*, the only area and power overhead come from the scanning circuit pushing data into *TD_Scan* and checking the results from it. This scanning circuit can share parts of the circuits scanning the decrypting *KEYs* since their functions are very similar.

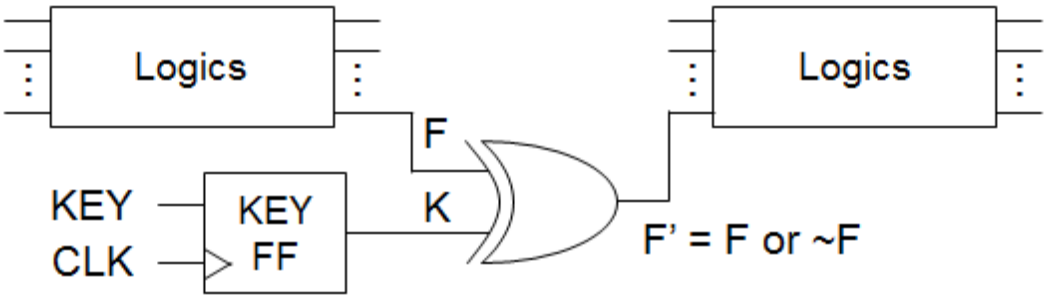


Figure 6.8. A simple logical encryption implementation

Table 6.1 summarizes the possible attacks and the scenarios seen by the two untrusted foundries, for both 2D and 3D designs. For design reconstruction threats, *foundry_1* may succeed for 2D designs, but it is very unlikely if the nets are well partitioned. It has no way to reconstruct a 3D chip since a portion of the transistors are hidden. For Trojan threats, the detection method is proposed to detect any change made by *foundry_1*. *Foundry_2* has no control of devices and has very limited knowledge of the design to insert functional Trojans.

As for reverse engineering and IC overbuilding attacks, they are meaningless for *foundry_1* since it needs *foundry_2* to finalize the chip or needs to guess the missing connections and circuits which is very difficult. *Foundry_2* cannot perform these two attacks because the source wafer count is limited by *foundry_1*.

Table 6.1. Possible Attacks and the Corresponding Situations Seen by the Two Untrusted Foundries

Threats	***Foundry_1***	***Foundry_2***
Design Reconstruction	2D IC: Very Difficult 3D IC: Impossible	Impossible due to a very limited information
Trojan Insertion	Possible, but will be detected.	No control of devices
Reverse Engineering	Meaningless	Impossible, the number of wafers is controlled by *foundry_1*
ICs Overbuild	Meaningless	

6.4 Implementation Details

This section provides the implementation details and algorithms, including net partition, transistor hiding, pin shaking, and Trojan detection scan path (*TD_scan)*.

6.4.1 Net Partition

The nets are partitioned to make the design reconstruction difficult for *foundry_1*. This kind of reconstruction is usually performed by proximity attacks, which rely on the pin locations and the known circuit structures [26]. Since *foundry_2* has a full access to every transistor, there are no constraints for selecting nets to be partitioned and fabricated by *foundry_2*. It is intuitive that it is harder to reconstruct a missing net if 1) there are many

missing nets clustered in a small region and 2) the net has many fan-outs (FO), as shown in Figure 6.9. The algorithm to select candidate nets to be partitioned applies the rules illustrated in Figure 6.10.

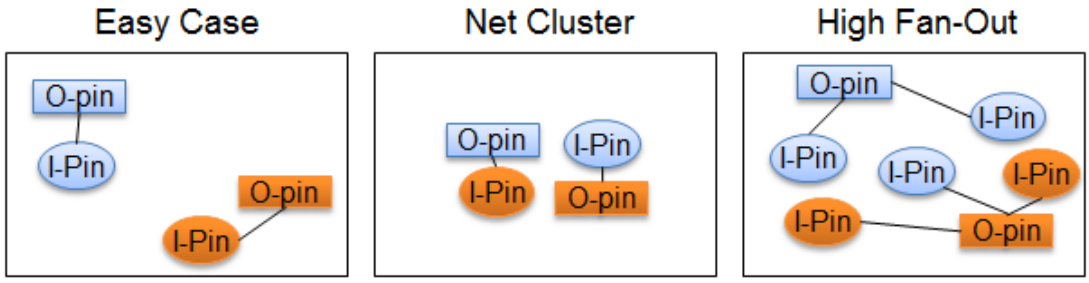


Figure 6.9. The difficulties of guessing missing nets

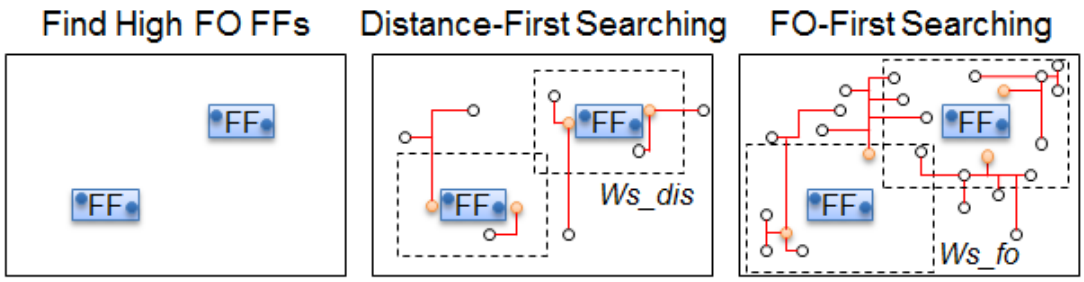


Figure 6.10. Selecting candidate nets to be partitioned

First, the high FO net driven by an FF's output Q-pin is selected followed by the net connecting to the same FF's input D-pin. This selection process searches FFs until the number of selected nets reaches the specified upper bound or all the FFs have been selected. All the selected Q and D pins are stored in the list *List_DQ*. The reasons for starting from FFs are 1) misconnections to FFs result in pipeline errors and 2) in general, FFs tend to have greater FOs than combinational gates. Next, two approaches with a given weighting are performed to select remaining nets: 1) distance-first search and 2) FO-first search. These two approaches are based on the FF pin locations on the nets selected in the first step. In both approaches, for

a Q-D pin pair from the *List_DQ* we select a pin or pins on several nets and partition the nets connecting to those pins. Then, the next pin pair in *List_DQ* is processed. This process repeats until the number of partitioned nets reaches the specified upper-bound. On average, the number of partitioned nets related to each pin pair in the *List_DQ* is similar in this searching process.

In the distance-first search method, a pin in a predefined searching window *Ws_dis* connecting to an un-partitioned net is selected when it has the minimum distance to the currently processed pin pair in the *List_DQ*. If there are multiple pins having the same distance, then the pin of a net with the highest FO is selected. The FO-first search method selects the pin connecting to a net having the highest FO in the searching window *Ws_fo*. If there are multiple-pins having the same FO, the one having the shortest distance is selected. All the selected candidate nets will be manufactured by *foundry_2*. *Foundry_1* would have to guess these net connections to reconstruct the design.

6.4.2 Transistor Hiding and Pin Shaking

In a 3D design, transistors could be hidden in the lower transistor tiers. Figure 6.11 illustrates how it works. In the Figure 6.11, small rectangles indicate footprints of standard cells, circles are pins related to partitioned nets, and the numbers mean the count of available transistors in the lower tiers of cells. If all the nets connecting to the standard cell are partitioned to be fabricated by *foundry_2*, this cell is a candidate for hiding. First, we check availability of unused transistors accessible for *foundry_2* in the lower tiers of the nearby cells in a searching window *Ws_hide*. If the available transistor count is enough, then this candidate cell is removed to be reconstructed by *foundry_2* using the distributed lower tier transistors and interconnects. Since this cell is physically hidden from *foundry_1*, it is impossible for

foundry_1 to correctly guess the nets connecting to it. An empty space is created by the hidden cell. It could provide clues for *foundry_1*. We move some nearby cells to this area to obfuscate layout for any distance-based proximity attackers designed to guess the missing nets. The area utilization factor is defined to control the upper bound of the empty area recovery. A candidate cell to be moved to the empty area satisfies the following two conditions: 1) at least one net corresponding to its pins is partitioned and this partitioned net is different from any of the partitioned nets connecting to the hidden cell, and 2) the cell's area is not greater than the available area of the empty space. The searching window is divided into grids, and at most one cell is selected from a grid. The selected candidate cells are then moved and evenly distributed in the empty space.

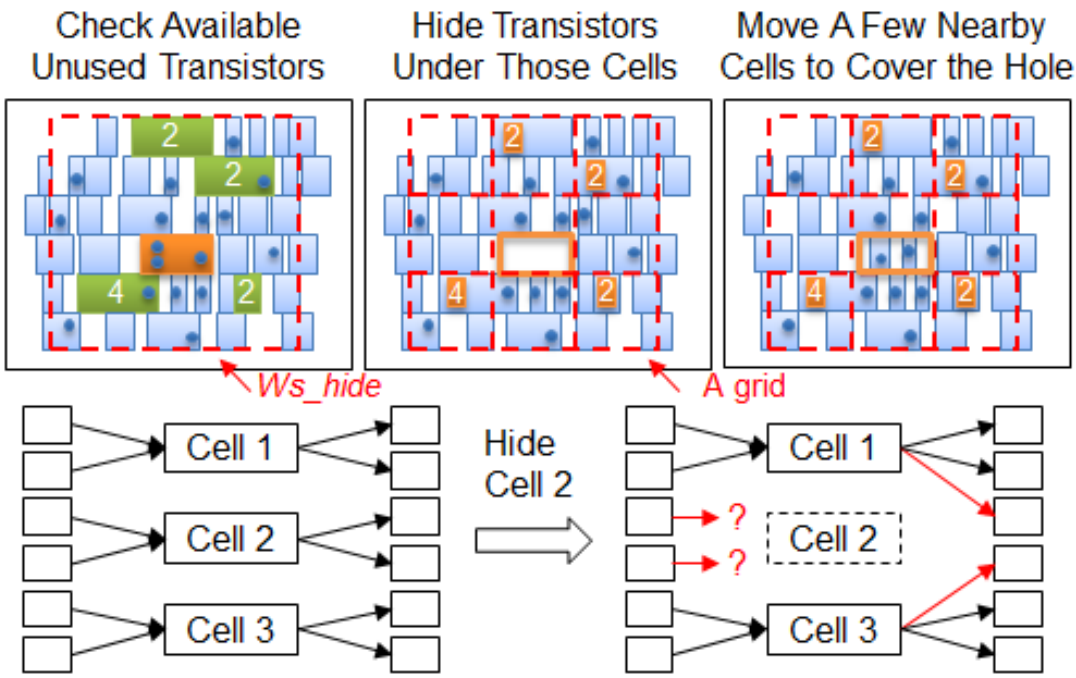


Figure 6.11. Transistor hiding and pin shaking

6.4.3 Trojan Detection Scan Path

Trojan detection scan path (*TD_Scan*) is specific to the 3D designs, because the crowbar current path may be broken if *foundry_1* changes only the upper tiers. Figure 6.12 shows the 3D structure of the *TD_Scan* and two attack scenarios as described in Section 6.3.2. If any of the four transistors adjacent to a pillar is used, the pillar must be used as a gate, a source, or a drain terminal. Thus, there is no need to chain all the pillars. Besides the chained non-functional pillars (i.e. no transistor has this pillar as a terminal or it is a terminal of an unused transistor), all other non-functional pillars are connected to VSS by *foundry_2* as shown in Figure 6.12 (c). For DeCap cells as shown in Figure 6.12 (d), one of the gates is used by *TD_Scan*, marked as C in the Figure, VDD and VSS are connected accordingly by *foundry_2*. If a Trojan is inserted by *foundry_1* unaware of the *TD_Scan*'s existence (attack scenario 1), two things may happen: The *TD_Scan* path may be broken or a crowbar current path to VSS may be formed. In attack scenario 2, *foundry_1* inserts Trojans using the upper tier transistors only and reconstructs the *TD_Scan* path, thus avoids breaking it and forming a crowbar current path.

Scenario 2 can be detected by measuring the active current of *TD_Scan* as shown in Figure 6.13. The original *TD_Scan* is designed to be phase aligned on all chained pillars, thus all pillars in the same pipeline stages are either all logic-0 or 1. First, we scan logic-zero to every stage in the chain; each pillar is now at logic-0 state. Then, we scan a logic-1 into the chain followed by an all logic-0 sequence. This logic-1 passes through the chain at a cycle *n* and is scanned into the pipeline stage *n*. It consumes a charge current $I_{charging\_pillars\_interconnects}\ (n)$ from VDD to charge the loadings (interconnects and pillars) within that pipeline stage changing state from logic-0 to 1. For the previous stage *n-1* changing

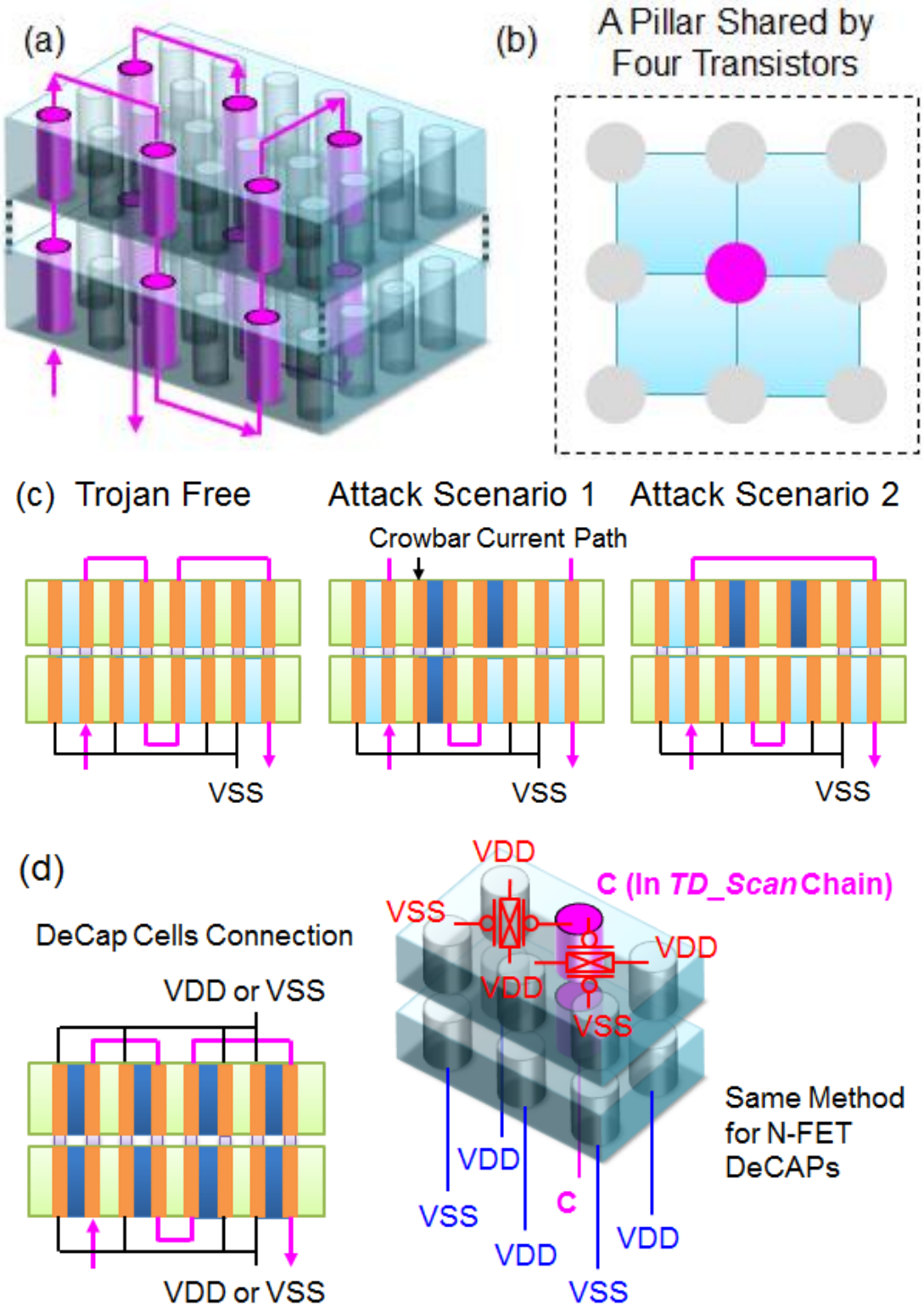


Figure 6.12. 3D Trojan detection scan chain (*TD_Scan*). (a) 3D structure; (b) four transistors share a pillar; (c) Trojan-free and attack scenarios; (d) DeCap cells

from logic-1 to 0 does not consume a large charge current from VDD for charging the pillars and interconnects, but it may consume a small charge current $I_{charging\_buf\_ff}(n-1)$ to the intermediate nodes of the buffers and FFs within that stage. For all the other stages maintaining logic-0, no current is consumed besides the leakage current $I_{leakage}$, which exists in all pipeline stages. The *Ivdd* measured in cycle *n* can be written as equation (6.1). Any modification of the chain changes the RC within the stage *n* and then changes the measured *Ivdd (n)*, thus the attack can be detected.

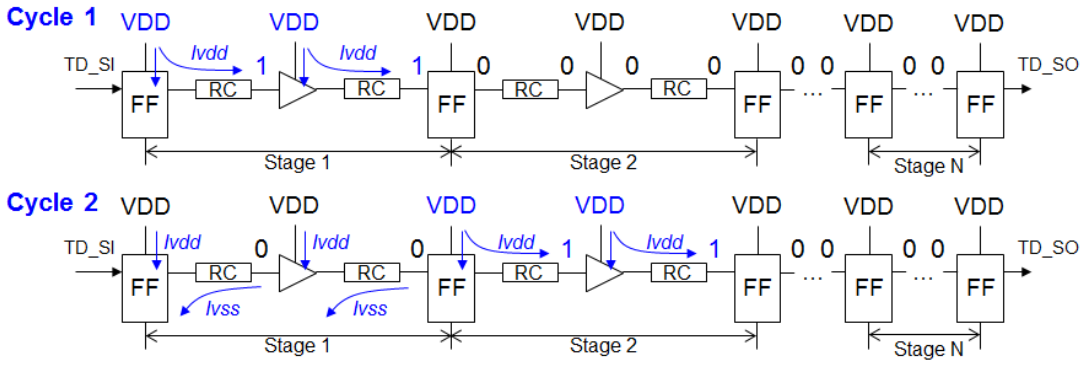


Figure 6.13. Trojan detection by *TD_Scan* path

$$I_{Vdd}(n) = I_{charging\_pillars\_interconnects}(n) + I_{charging\_buf\_ff}(n-1) + I_{leakage} \quad (6.1)$$

6.5 Security Evaluation

To evaluate effectiveness of this methodology, 10 MCNC LGSynth'91 benchmark circuits were designed using the proposed methodology in a three-tier 3D design fashion. The circuits are synthesized by *Synopsys Design Compiler* with a standard cell library containing 61 ten-track height 3D standard cells in *chain canvas* design style. The pillar heights *h* of the three

tiers are all the same, thus the dimension of every transistor is identical. The cells include Inverter, Buffer, NAND2, NOR2, NAND3, NOR3, XOR, and Flip-Flop gates; they can be categorized in three groups of 1) using tier 1 transistors only, 2) using tiers 1 and 2, and 3) using all three tiers. Thus, three different driving strengths are provided for the same cell footprint. Next, a FF & XOR based logical encryption is performed; the *KEY-FFs* and XOR gates are randomly inserted with an upper-bound of 5% area overhead. As discussed in Section 6.3.2, logical encryption is not required for this methodology, but it is included here for completing an entire security methodology. Then, the designs are floorplanned and placed by *Cadence Encounter Digital Implementation System* (*EDI*). The design aspect ratio is set to 1 and the placement utilization is 80%. On-the-fly design optimization is performed during placement; thus, the circuit structure may be changed.

The placement records of cell and pin locations, and the optimized netlists are reported by *EDI*. Then, they are processed by the tools we developed to perform net partitioning, transistor hiding, and pin shaking, which construct the design that *foundry_1* sees. 5% of the total nets are selected to be partitioned, 25% of them are high fan-out FFs related, and the remaining 75% are selected based on distance-first and FO-first search in 1:1 weighting. The searching window of distance-first *Ws_dis* is set to 6 cell-row-heights (*RH*) x 6 *RH*. The searching window of FO-first *Ws_fo* is 10 *RH* x 10 *RH*. After net partitioning, transistor hiding, and pin shaking are performed. All the feasible cells are selected and hidden. The searching window *Ws_hide* is set to 10 *RH* x 10 *RH*, and the filling utilization of the empty space created by transistor hiding in pin shaking stage is set to 70%.

We developed a proximity attacker which attempts to reconstruct the missing interconnects. The net reconstruction is based on the distance of open pins due to net

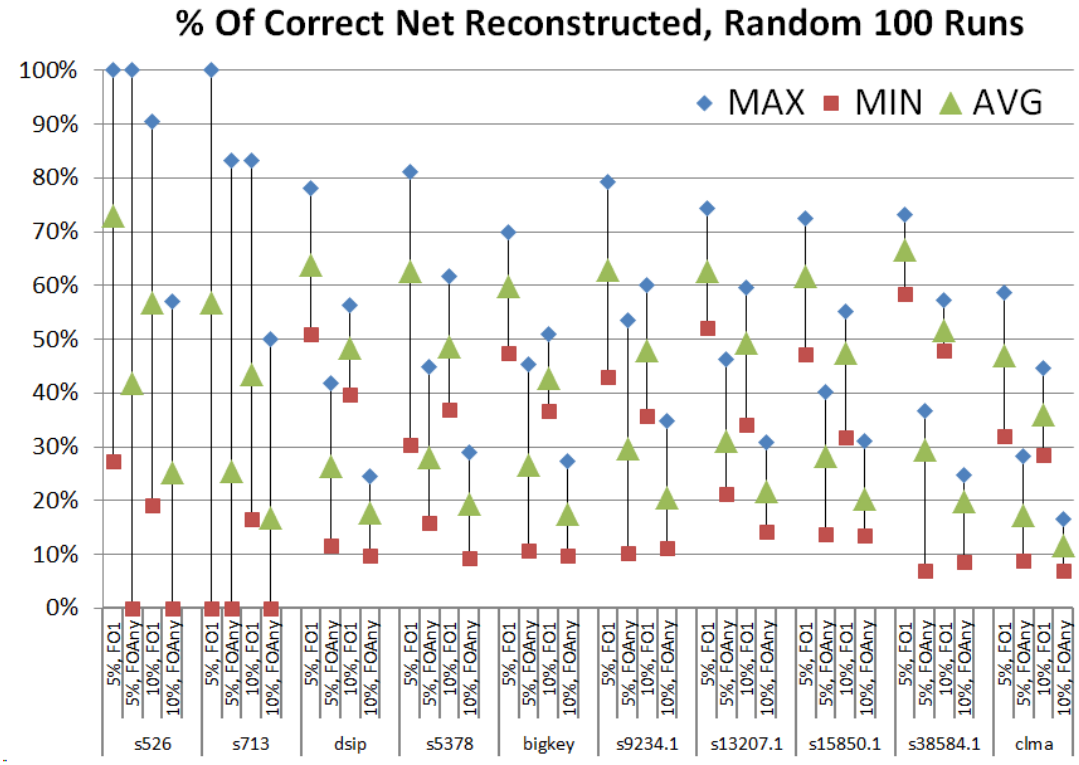


Figure 6.14. Proximity attack results, randomly select nets to be partitioned in four cases: (1) 5% *FO1*, (2) 5% *FOAny*, (3) 10% *FO1*, and (4) 10% *FOAny*

partitioning and the circuit structure. First, it connects each output pin (o-pin) and the nearest input pin (i-pin) if this connection does not create a combinational loop in the circuit. Then, it connects the remaining i-pins to their nearest o-pin if 1) this connection does not create a combinational loop and 2) every i-pin of a cell is driven by a different o-pin. To check the strength of this attacker, the benchmark circuits were randomly partitioned and attacked. Four partition cases were simulated: 1) 5% of the total nets were included in the partition, the nets having 1 fan-out only (*FO1*) were selected; 2) 5% of nets with no fan-out constraint (*FOAny*); 3) 10% of nets with *FO1* constraint, and 4) 10% of nets with no fan-out constraint (*FOAny*). Each circuit and each case were simulated 100 times. The results indicating the percentage of correct nets reconstructed are shown in Figure 6.14; some cases are 100% correctly

reconstructed. The results show that 1) the more nets partitioned, the harder it is to reconstruct them; and 2) high fan-out nets make reconstruction much more difficult since all fan-out pins have to be correctly connected.

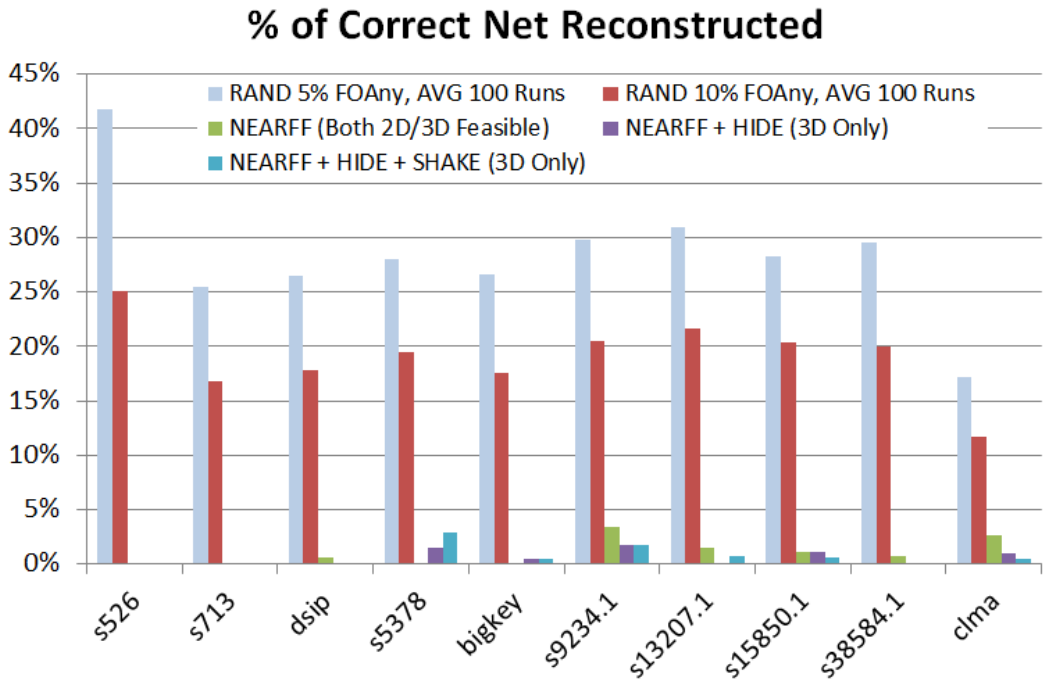


Figure 6.15. Proximity attack results for the designs implemented with the proposed methodology

Figure 6.15 shows the results for the designs implemented by our methodology. Three implementations (1) net partition only (*NEARFF*), which is feasible for both 2D and 3D designs, (2) *NEARFF* with cells hidden (*NEARFF + HIDE*), which is only feasible in 3D designs, and (3) *NEARFF + HIDE* with pins shaken (*NEARFF + HIDE + SHAKE*), which is only feasible in 3D designs as well. The other two data sets are the averages of 100 random runs of 5% and 10% nets with no fan-out constraints (*FOAny*) as shown in Figure 6.14. With our proposed methodology, the percentage of correct net reconstruction is dramatically reduced. In some circuits, not even a single net is correctly guessed. The average percentages

of the correct nets reconstructed for the ten circuits in the three cases are: 1% in *NEARFF*, 0.58% in *NEARFF + HIDE*, and 0.69% in *NEARFF + HIDE + SHAKE*.

6.6 Summary

This work proposes a secure split-fabrication design methodology for the VeSFET based 2D and 3D integrated circuits. The design partition and piracy prevention, hardware Trojan insertion prevention, and Trojan detection methods are described. In our approach, the design is fabricated by two independent untrusted foundries. By taking advantage of the VeSFET's unique and powerful two-side accessibility and monolithic 3D integration capability, several unique split-fabrication features are feasible, such as 1) the complete freedom of net partitioning, 2) making some transistors invisible to *foundry_1*, and 3) Trojan prevention and detection techniques enabled by *foundry_2*, which are invisible to *foundry_1*. These invisible features make it difficult or even impossible for *foundry_1* to reconstruct the design since some transistors are hidden. If *foundry_1* moves any of the existing transistors or adds any extra transistors, this is easily detected by the crowbar current flow in 2D and 3D designs, or by active current measurement in 3D designs. Ten MCNC LGSynth'91 benchmark circuits were designed with this methodology. With 5% nets manufactured by *foundry_2*, the average percentage of the correctly reconstructed partitioned nets is less than 1%.

Chapter 7

CONCLUSION AND FUTURE OPPORTUNITIES

7.1 Conclusions

In the era of transistor technology in 10nm, 7nm, 5nm, and even below, conventional MOSFET's physical limitation is approaching. Future growth becomes very challenging. It is not feasible to integrate more devices and functions barely by technology scaling, which is the path in the past decades. 3D IC and monolithic 3D IC offer attractive features, which allow designers to integrate more devices on the same chip footprint by stacking them. However, the higher power density need to be well addressed and those extra vertical inter-tier connections cost unavoidable area overheads in MOSFET designs. It is expected that fundamental changes of device structure are required for sustaining future technology developments. A novel transistor Vertical Slit Field Effect Transistor (VeSFET) offers attractive characteristics for future IC and monolithic 3D integration, such as two-side accessibility, lower power consumption, high regularity, circle-based patterning, and good thermal properties.

This dissertation investigates VeSFET monolithic 3D integration technology in system and physical levels. The domains include SRAM, physical design, design methodology, FPGA, dynamic reconfigurable architecture, and hardware security. Unique and power applications and architectures are proposed, which are not feasible using MOSFET

technologies. In section 2.3, VeSFET SRAM performance is assessed. The results show that VeSFET SRAM design is speed competitive to CMOS SRAM with about 40% of dynamic read energy consumption and 35% of total power consumption for read access rate 100MHz.

Chapter 3 provides a detailed physical assessment on VeSFET monolithic 3D integration and compared with an equivalent CMOS technology. In particular, power delivery network (PDN) IR-drop and clock distribution network (CDN) characteristics are investigated, which are crucial for chip function and performance. Different number of tiers and 3D partition methods are evaluated. The results show VeSFET-based monolithic 3D integration is promising even when the device, circuit designs, and layout styles are not tuned and are assumed under pessimistic conditions. On average over all test cases, VeSFET 3D IC's maximum static PDN IR-drop is 38.5% - 52.3% of CMOS designs. VeSFET designs' CDNs consume 70.6% - 73.7% power but have 120.3% - 194.9% clock skews compared to CMOS designs. The larger clock skew is due to the weaker driving current of this untuned transistor model, it requires more buffers to distribute clock to all Flip-Flops. Although the clock skew is still in an acceptable range considering the clock period, further optimizations on VeSFET device are suggested for better CDN performance.

A fast, fully verifiable, and hardware predictable ASIC design methodology using VeSFET 3D FPGA is proposed in Chapter 4. In the proposed methodology, the circuit is first designed as a 3D FPGA using a conventional FPGA design flow. With a little extra Back End of Line (BEOL) masking cost, the design implemented on the 3D FPGA is migrated to the final 2D ASIC, which has exactly the same performance as the 3D FPGA and the verification tasks performed on the 3D FPGA remain valid for the final 2D ASIC. The 2D ASIC has the same body as the silicon-proven 3D FPGA, which greatly mitigates the unpredictable factors

of fabrication. The proposed methodology retains all the benefits of FPGA design flow, such as short design cycle and flexibility, as well as hardware-based verification, debugging and performance prediction. VeSFET 3D FPGA is assessed, comparing to the VeSFET 2D FPGA, the performances are on average 15% faster, consume 17% less power and 44% less area.

To address the demands on computing power, a high performance architecture using fast dynamic reconfigurable accelerators (F-RACCs) is proposed and assessed in Chapter 5. This dynamic reconfigurable architecture offers very attractive features for systems running various applications in wide spectrums such as data center or cloud computing systems. The F-RACCs are implemented by monolithic 3D FPGAs using VeSFETs. Comparing with the systems using conventional FPGA accelerators (C-RACCs), which are supported by partial configuration techniques with fastest programming speed, this architecture improved speed on all applications and achieved 1.31x and 2.82x (using 1 and 12 accelerator instances) maximum speedups. Comparing with the time taken by running on pure CPU software path without any accelerators, this architecture achieves geometric means 10.16x and 75.59x, maximum 94.93x and 565.12x using 1 and 12 accelerator instances, respectively.

As the security threats are of ever greater concern, a very secure split-fabrication design methodology is proposed in Chapter 6. We propose the design partition and piracy prevention, hardware Trojan insertion prevention, and Trojan detection methods. In the 2D/3D designs using this method, any Trojan insertion or design tampering can be easily detected by crowbar current measurement or the Trojan detection scan path (T*D_scan*). In the 3D designs, some transistors are physically hidden from the front-end *foundry_1*'s view, which causes that it is impossible for this foundry to reconstruct the circuit. With 5% nets manufactured by the back-

end *foundry_2* and attacked by a proximity attacker, the average percentage of the correctly reconstructed partitioned nets is less than 1%.

In summary, VeSFET technology offers unique advantages which are not feasible in MOSFETs. Its two-side accessibility, abundant self-embedded inter-tier connections, regularity, and lower power consumption are very attractive features for future 2D / 3D ICs. It is a promising technology for sustaining the growth of future technologies.

7.2 Future Work and Opportunities

In this dissertation, VeSFET's unique characteristics are assessed and their powerful applications are demonstrated. Although VeSFET development is still in the early stages and it is not well-tuned yet, it has revealed new opportunities for future 2D / 3D ICs. Such opportunities are hard to be achieved with MOSFETs, because VeSFET has a revolutionary structure, which provides special capabilities. Besides the scopes presented and assessed in this dissertation, there are more opportunities and future works which are worthy for further explorations.

7.2.1 Device and Process Tuning

The successfully fabricated samples are in 65nm-equivalent technology [28-29], which are hard to directly compete CMOS / FinFET in the advanced technology nodes, such as 10nm and 7nm feature sizes. Although reference [31] conducts a device level TCAD study on 7nm node and shows promising results, it would be great to invest researches and developments on VeSFET device-level engineering, circuits, and layout designs. Such tasks can reveal and optimize the actual VeSFET performance, thus makes this technology more feasible and accessible.

To fully utilize VeSFET's monolithic 3D friendly characteristics, which allow designers to stack more devices. It would be beneficial to target on the process flows of stacking more tiers instead of only few tiers. The advantage of using VeSFET monolithic 3D integration over MOSFET monolithic 3D is expected to be more significant when the designs have more tiers.

7.2.2 Utilizing Independent Gate Configuration (IGC)

A unique advantage of VeSFET is the independent gate configuration (IGC) as presented in Section 2.1. Fewer device usage, denser layouts, and fine-grain transistor behavior adjustments are feasible with IGC VeSFET. However, past literatures and researches did not fully utilize this advantage.

There are many opportunities which are feasible with this feature. For analog or mixed-signal designs, since the behaviors and characteristics of each transistor can be adjusted by controlling the second gate, the analog designers will have the freedom of utilizing different transistor behaviors as they desire for circuit implementations. Better and more efficient circuit designs are possible. For digital designs, using AND- and OR-type IGC VeSFET intuitively reduces the transistor count; smaller standard cell footprints and denser designs can be achieved. Furthermore, fine-grain clock gating or circuit block enabling / disabling techniques can be implemented on transistor level by using the second date as the control signal. Circuit behavior adjustments, such as the tunings of delay, pulse width, and transition time are feasible using IGC VeSFETs. Post-fabrication design adjustment to overcome process variation is also possible. This research direction has the potential to boost the performance of VeSFET ICs and create new possibilities of better circuit designs.

7.2.3 Heterogeneous Monolithic 3D Integration

Since the terminals of each VeSFET can be directly accessed from the backside of the device layer, heterogeneous monolithic 3D integration could be possible. Each transistor terminal can be treated as a primary I/O by accessing them from the backside. A tier or several tiers of VeSFETs can be placed on top of existing device and metal layers, then directly connect them to the backside of the VeSFET layer. Different types of devices, such as CMOS logics, memory cells, non-volatile memory, etc. could be monolithic 3D integrated together with VeSFETs. Different high performance architectures could be achieved using such heterogeneous integration. For example, VeSFET may provide better implementations for in-memory computing, which has computing elements placed very close to memory for easier data access.

7.2.4 Low Power Applications

Prior published literatures have presented VeSFET's characteristics of lower power consumption. This feature has becoming more important with the continuous demands on higher computing power and more functions integrated together under limited power and thermal budgets. In the past decade, mobile devices dominated the growth of whole market and technology developments. Devices running on batteries, such as smartphones, wearable devices, internet of things (IoT) applications, etc., demand longer battery life, less heat generation, and larger computing power per Watt. For such applications, VeSFET designs offer attractive advantages. More devices can be integrated together and keep same power consumption and thermal dissipation.

7.2.5 Neural Networks and Machine Learning

Neural networks and machine learning have drawn attentions recently. Although the idea of artificial neural networks are not new, such applications now become more feasible with the maturity of technologies and the big amount of accessible data for system training. Neural networks require dense interconnects for neuron cells connections and weight adjustment schemes for network behavior tuning. To support dense interconnects, either 2D or 3D implementations can be benefited with VeSFET's two-side accessibility. For 2D designs, the backside routing offers doubled routing resources. For 3D designs, the abundant freely accessible inter-tier interconnects (VeSFET pillars) without extra area overhead are beneficial for cross-tier interconnects. The weight adjustment may be implemented using the second gates of IGC VeSFETs instead of using more transistors.

7.2.6 More Security Features

Hardware security features could be enhanced using IGC VeSFETs. The mixture of AND- and OR-type VeSFETs with IGC configuration could be utilized for logic encryption. A more complicated key can be created by using the second gate. The key most be correctly provided to the chip, thus the transistor behaviors can be well set to be properly functional.

We may also apply this concept for physical unclonable functions (PUFs). PUFs relies on physical characteristics, which are determined in fabrication process. Many small variations accumulate together and create an unclonable characteristic of this chip. Such process variations change the relationship of transistor behavior and the voltages applied on the second gate of IGC VeSFETs. We may implement PUFs using the mixture of different types of IGC

and TGC VeSFETs, a complicated key could be designed and the challenge–response authentication is harder to be predicted.

7.2.7 Testing, Repairing, and ECO

With the growth of chip complexity, design efforts, fabrication cost, and performance unpredictability, the importance of testing, repairing, and ECO (engineering change orders) all grow. The backside accessibility of each VeSFET may offer new approaches. New self-test features, repairing, redundancy features are possible by reconfiguring the well-arranged backside interconnects. Also, ECO tasks may create extra interconnects and enable spare cells at the backside of VeSFETs instead of changing regular routing layers at the front side. This mitigates the probability of performance changes due to the changed parasitic capacitance of the dense interconnects at the front side.

7.2.8 Image Sensing

Image sensing applications usually use pixel arrays. The image it processed, stored, and presented using regular pixels. For such application, VeSFET's regular structure could be a good fit. VeSFET pillars and transistor locations naturally form a highly regular array, each grid enclosed by four pillars is a square. We may use this characteristic to design image sensing chips, where each grid processes a pixel of the image. There is no study on image sensing techniques using VeSFETs, but such regular chip structure may bring advantages for image sensing applications.

Combining with heterogeneous monolithic 3D integration, computing elements can be placed on separated tiers below the image sensing tier for on-the-fly image processing or analysis. The computing elements can be implemented in many ways for different applications.

They can be image processing units, for handling the sensed image, such applications can be camera, video recorder, image detector, etc. They can be neural networks with machine learning capabilities for analyzing the input images immediately as surveillance systems.

7.2.9 Microchip Implants

The ideas of microchip implants have been proposed for medical or tracking applications. Biomedical chips can be implanted to human or animal bodies as biomedical monitors for sensing physiological signals. Even more aggressively, those chips may carry medicines and release them internally for better and more accurate treatments. Such devices can stay in the bodies longer with lower power consumption characteristics offered by VeSFET technology. Moreover, VeSFET's two-side accessibility may reveal new opportunities.

Conceptually, a layer of VeSFET pillar array can be designed and used as an array of electrical probes. One side of the pillars can be attached to the body for sensing, the other side of the pillars are reserved for signal interconnects. It may be possible to use a single VeSFET layer for both sensing and processing purposes or use multiple VeSFET and pure pillar layers together (e.g. a functional VeSFET layer for computing integrated with a pure pillar layer for probing, or several VeSFET layers integrated for different purposes). Such imaginations could be possible by properly leveraging VeSFET's unique structure and characteristics, which are not feasible in conventional devices.

References

[1] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. of the 38th ACM/IEEE International Symposium on Computer Architecture (ISCA'11),* Jun. 2011, pp. 365-376

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro,* vol. 32, no. 3, Apr. 2012, pp. 122-134

[3] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "High-density integration of functional modules using monolithic 3D-IC technology," in *Proc. of the 18th Asia and South Pacific Design Automation Conference (ASP-DAC'13),* Jan. 2013, pp. 681-686

[4] K. Acharya *et al.*, "Monolithic 3D IC design: power, performance, and area impact at 7nm," in *Proc. of the 17th International Symposium on Quality Electronic Design (ISQED'16),* Mar. 2016, pp. 41-48

[5] D. K. Nayak, S. Banna, S. K. Samal, and S. K. Lim, "Power, performance, and cost comparisons of monolithic 3D ICs and TSV-based 3D ICs," in *Proc. of the IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S'15),* Oct. 2015, pp. 1-2

[6] W. -T. J. Chan, A. B. Kahng, and J. Li, "Revisiting 3DIC benefit with multiple tiers," in *Proc. of the 18th ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP'16),* Jun. 2016, pp. 1-8

[7] K. Changl, K. Acharyal, S. Sinha, B. Cline, G. Yeric, and S. K. Lim, "Power benefit study of monolithic 3D IC at the 7nm technology node," in *Proc. of the IEEE/ACM*

International Symposium on Low Power Electronics and Design (ISLPED'15), Jul. 2015, pp. 201-206

[8] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim, "Monolithic 3D IC vs. TSV-based 3D IC in 14nm FinFET technology," in *Proc. of the IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S'16)*, Oct. 2016, pp. 1-2

[9] W. R. Davis *et al.*, "Demystifying 3D ICs: the pros and cons of going vertical," *IEEE Design & Test of Computers.*, vol. 22, no. 6, Dec. 2005, pp. 498-510

[10] S. Wong, A. El-Gamal, P. Griffin, Y. Nishi, F. Pease, and J. Plummer, "Monolithic 3D integrated circuits," in *Proc. of the International Symposium on VLSI Technology, Systems and Application (VLSI-TSA'07),* Apr. 2007, pp. 1-4

[11] T. -T. Wu *et al.*, "Low-cost and TSV-free monolithic 3D-IC with heterogeneous integration of logic, memory and sensor analogy circuitry for internet of things," in *Proc. of the 61st IEEE International Electron Devices Meeting (IEDM'15),* Dec. 2015, pp. 25.4.1-25.4.4

[12] O. Billoint *et al.*, "A comprehensive study of monolithic 3D cell on cell design using commercial 2D tool," in *Proc. of the Design, Automation and Test in Europe (DATE'15),* Mar. 2015, pp. 1192-1196

[13] Z. Zhang *et al.*, "Low-temperature monolithic three-layer 3-D process for FPGA," *IEEE Electron Device Letters (EDL)*, vol. 34, no. 8, Aug. 2013 pp. 1044-1046

[14] Z. Zhang, Y. Y. Liauw, C. Chen, and S. S. Wong, "Monolithic 3-D FPGAs," *Proceedings of the IEEE*, vol. 103, no. 7, Jul. 2015, pp. 1197-1210

[15] O. Turkyilmaz, G. Cibrario, O. Rozeau, P. Batude, and F. Clermidy, "3D FPGA using high-density interconnect monolithic integration," in *Proc. of the Design, Automation and Test in Europe (DATE'14)*, Mar. 2014, pp. 1-4

[16] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3-D FPGA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 26, no. 2, Feb. 2007, pp. 216-229

[17] G. Venkatesh *et al.*, "Conservation cores: reducing the energy of mature computations," in *Proc. of the 15th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'10)*, Mar. 2010, pp. 205-218

[18] J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, and G. Reinman, "Architecture support for accelerator-rich CMPs," in *Proc. of the 49th ACM/EDAC/IEEE Design Automation Conference (DAC'12)*, Jun. 2012, pp. 843-849

[19] J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, and G. Reinman, "Architecture support for domain-specific accelerator-rich CMPs," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 4s, article 131, Mar. 2014, pp. 131:1-131:26

[20] Y. -T. Chen *et al.*, "Accelerator-rich CMPs: from concept to real hardware," in *Proc. of the 31st IEEE International Conference on Computer Design (ICCD'13)*, Oct. 2013, pp. 169-176

[21] R. Tessier, K. Pocek, and A. DeHon, "Reconfigurable computing architectures," *Proceedings of the IEEE*, vol. 103, no. 3, Mar. 2015, pp. 332-354

[22] C. Kachris and D. Soudris, "A survey on reconfigurable accelerators for cloud computing," in *Proc. of the 26th International Conference on Field Programmable Logic and Applications (FPL'16)*, Sep. 2016, pp. 1-10

[23] K. Vaidyanathan, B. P. Das, E. Sumbul, R. Liu, and L. Pileggi, “Building trusted ICs using split fabrication,” in *Proc. of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST’14)*, May. 2014, pp. 1-6

[24] M. Jagasivamani, P. Gadfort, M. Sika, M. Bajura, and M. Fritze, “Split-fabrication obfuscation: metrics and techniques,” in *Proc. of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST’14),* May. 2014, pp. 7-12

[25] J. Valamehr *et al.*, “A 3-D split manufacturing approach to trustworthy system development,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 4, Apr. 2013, pp. 611-615

[26] J. Rajendran, O. Sinanoglu, and R. Karri, “Is split manufacturing secure?” in *Proc. of the Design, Automation and Test in Europe (DATE’13*), Mar. 2013, pp. 1259-1264

[27] Integrated circuits device system, and methodology of fabrication, by Wojciech P. Maly. (2015, Oct. 6). Patent US 9,153,689 B2

[28] W. Maly *et al.*, "Twin gate, vertical slit FET (VeSFET) for highly periodic layout and 3D integration," in *Proc. of the 18th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES’11)*, Jun. 2011, pp. 145-150

[29] Z. Chen *et al.*, "N-channel junction-less vertical slit field-effect transistor (VeSFET): fabrication-based feasibility assessment," in *Proc. of the International Conference on Solid-State and Integrated Circuit (ICSIC’12)*, Mar. 2012, pp. 1-5 / *International Proceedings of Computer Science and Information Technology (IPCSIT)*, vol. 32, 2012, pp 1-5

[30] X. Qiu, M. Marek-Sadowska, and W. Maly, “Vertical slit field effect transistor in ultra-low power applications,” in *Proc. of the 13th International Symposium on Quality Electronic Design (ISQED’12)*, Mar. 2012, pp. 384-390

[31] P. -L. Yang, T. B. Hook, P. J. Oldiges, and B. B. Doris, “Vertical slit FET at 7-nm node and beyond,” *IEEE Transactions on Electron Devices (T-ED)*, vol. 63, no. 8, Aug. 2016, pp. 3327-3334

[32] M. Weis, R. Emling, and D. Schmitt-Landsiedel, “Circuit design with independent double gate transistors,” *Advances in Radio Science (Adv. Radio Sci.)*, 7, May. 2009, pp. 231-236

[33] A. Kamath *et al.*, “Realizing AND and OR functions with single vertical-slit field-effect transistor,” *IEEE Electron Device Letters (EDL)*, vol. 33, no. 2, Feb. 2012, pp. 152-154

[34] X. Qiu, M. Marek-Sadowska, and W. P. Maly, "Characterizing VeSFET-based ICs with CMOS-oriented EDA infrastructure," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 33, no. 4, Apr. 2014, pp. 495-506

[35] X. Qiu and M. Marek-Sadowska, “Can pin access limit the footprint scaling?” in *Proc. of the 49th ACM/EDAC/IEEE Design Automation Conference (DAC’12)*, Jun. 2012, pp. 1100-1106

[36] M. Weis *et al.*, "Stacked 3-dimensional 6T SRAM cell with independent double gate transistors," in *Proc. of the IEEE International Conference on IC Design and Technology (ICICDT’09)*, May. 2009, pp. 169-172

[37] X. Qiu, M. Marek-Sadowska, and W. Maly, “3D chips can be cool: thermal study of VeSFET-based ICs,” in *Proc. of the 63rd IEEE Electronic Components and Technology Conference (ECTC’13)*, May. 2013, pp. 2349-2355

[38] X. Qiu, M. Marek-Sadowska, and W. P. Maly, "Three-dimensional chips can be cool: thermal study of VeSFET-based 3-D chips," *IEEE Transactions on Very Large Scale Integration Systems (T-VLSI),* vol. 23, no. 5, May. 2015, pp. 869-878

[39] A. Pfitzner, "Vertical-slit field-effect transistor (VeSFET) - design space exploration and DC model," in *Proc. of the 18th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES'11),* Jun. 2011, pp. 151-156

[40] D. Kasprowicz, "A compact model of VeSFET capacitances," in *Proc. of the 18th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES'11),* Jun. 2011, pp. 121-126

[41] N. Muralimanohar, R. Balasubramonian and N. P. Jouppi, "CACTI 6.0: a tool to model large caches," HP Laboratories, HPL-2009-85

[42] S. Thoziyoor, N. Muralimanohar, J. H. Ahn and N. P. Jouppi, "CACTI 5.1," HP Laboratories, HPL-2008-20

[43] S. S. Iyer, J. E. Barth Jr., P. C. Parries, J. P. Norum, J. P. Rice, L. R. Logan and D. Hoyniak, "Embedded DRAM: technology platform for the Blue Gene/L chip," *IBM Journal of Research and Development,* vol. 49, no. 2/3, March/May 2005, pp. 333-350

[44] S. K. Samal, K. Samadi, P. Kamal, Y. Du, and S. K. Lim, "Full chip impact study of power delivery network designs in monolithic 3D ICs," in *Proc. of the 33rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD'14),* Nov. 2014, pp. 565-572

[45] S. K. Samal, K. Samadi, P. Kamal, Y. Du, and S. K. Lim, "Full chip impact study of power delivery network designs in gate-level monolithic 3-D ICs," *IEEE Transactions*

on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 36, no. 6, Jun. 2017, pp. 992-1003

[46] *OpenCores*. [Online]. Available: http://opencores.org

[47] International Workshop on Logic & Synthesis (2005, Jun. 8), *IWLS 2005 Benchmarks*. [Online]. Available: http://iwls.org/iwls2005/benchmarks.html

[48] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Proc. of the 12th Asia and South Pacific Design Automation Conference (ASP-DAC'07)*, Jan. 2007, pp. 780-785

[49] J. Cong, G. Luo, J. Wei, and Y. Zhang, "A multilevel analytical placement for 3D ICs," in *Proc. of the 14th Asia and South Pacific Design Automation Conference (ASP-DAC'09)*, Jan. 2009, pp. 361-366

[50] G. Luo, Y. Shi, and J. Cong, "An analytical placement framework for 3-D ICs and its extension on thermal awareness," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 4, Apr. 2013, pp. 510-523

[51] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Placement-driven partitioning for congestion mitigation in monolithic 3D IC designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 4, Apr. 2015, pp. 540-553

[52] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Shunk-2-D: a physical design methodology to build commercial-quality monolithic 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 36, no. 10, Oct. 2017, pp. 1716-1724

[53] S. K. Samal, D. Nayak, M. Ichihashi, S. Banna, and S. K. Lim, "Tier partitioning strategy to mitigate BEOL degradation and cost issues in monolithic 3D ICs," in *Proc. of the 35th IEEE/ACM International Conference on Computer-Aided Design (ICCAD'16)*, Nov. 2016, pp. 1-7

[54] P. Batude *et al.*, "Advances, challenges and opportunities in 3D CMOS sequential integration," in *Proc. of the 57th IEEE International Electron Devices Meeting (IEDM'11)*, Dec. 2011, pp. 7.3.1-7.3.4

[55] S. K. H. Fung *et al.*, "65nm CMOS high speed, general purpose and low power transistor technology for high volume foundry application," in *Proc. of the Symposium on VLSI Technology (VLSIT'04)*, Jun. 2004, pp. 91-92

[56] Mentor Graphics Corporation, "Veloce2: high performance high capacity emulation systems," 2013. [Online] Available: http://s3.mentor.com/public_documents/datasheet/products/fv/emulation-systems/veloce2-brochure.pdf

[57] Altera Corporation, "HardCopy ASICs," Aug. 2010. [Online] Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/po/ss-hcasics.pdf

[58] V. Betz and J. Rose, "VPR: a new packing, placement and routing tool for FPGA research," in *Proc. of the 7th International Workshop on Field-Programmable Logic and Applications (FPL'97)*, Sep. 1997, pp. 213-222

[59] J. Luu *et al.* "VTR 7.0: next generation architecture and CAD system for FPGAs," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 2, Jun. 2014, pp. 6:1-6:30.

[60] Berkeley Logic Synthesis and Verification Group, "ABC: a system for sequential synthesis and verification,"
[Online] Available: http://www.eecs.berkeley.edu/~alanmi/abc/

[61] J. Lamoureux and S. J. E. Wilton, "Activity estimation for field-programmable gate arrays," in *Proc. of the 16th International Conference on Field Programmable Logic and Applications (FPL'06)*, Sep. 2006, pp. 1-8

[62] *Partial reconfiguration user guide, UG702 (v14.5)*, Xilinx Inc., San Jose, CA, USA, Apr. 2013

[63] J. Cong, Z. Fang, M. Gill, and G. Reinman, "PARADE: a cycle-accurate full-system simulation platform for accelerator-rich architectural design and exploration," in *Proc. of the 34th IEEE/ACM International Conference on Computer-Aided Design (ICCAD'15)*, Nov. 2015, pp. 380-387

[64] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, May 2011, pp. 1-7

[65] A. Putnam *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services," in *Proc. of the 41st ACM/IEEE International Symposium on Computer Architecture (ISCA'14)*, Jun. 2014, pp. 13-24

[66] B. Sukhwani *et al.*, "Database Analytics: a reconfigurable-computing approach," *IEEE Micro*, vol. 34, no. 1, Jan 2014, pp. 19-29

[67] J. Casper and K. Olukotun, "Hardware acceleration of database operations," in *Proc. of the 22nd ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'14)*, Feb. 2014, pp. 151-160

[68] Y. Shan *et al.*, "FPMR: MapReduce framework on FPGA a case study of RankBoost acceleration," in *Proc. of the 18th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'10)*, Feb. 2010, pp. 93-102

[69] Y. -M. Choi and H. K. -H. So, "Map-Reduce processing of K-means algorithm with FPGA-accelerated computer cluster," in *Proc. of the 25th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP'14)*, Jul. 2014, pp. 9-16

[70] K. H. Tsoi and W. Luk, "Axel: a heterogeneous cluster with FPGAs and GPUs," in *Proc. of the 18th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'10)*, Feb. 2010, pp. 115-124

[71] S. Liang *et al.*, "A coarse-grained reconfigurable architecture for compute-intensive MapReduce acceleration," *IEEE Computer Architecture Letters (CAL)*, vol. 15, no. 2, Jul. - Dec. 2016, pp. 69-72

[72] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM - 50th anniversary issue: 1958 - 2008*, vol. 51, no. 1, Jan. 2008, pp 107-113

[73] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *The Journal of Machine Learning Research*, vol. 4, Nov. 2003, pp. 933-969

[74] S. Trimberger, D. Carberry, A. Johnson, and J. Wong, "A time-multiplexed FPGA," in *Proc. of the 5th IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM'97)*, Apr. 1997, pp. 22-28

[75] M. Nakajima and M. Watanabe, "A sixteen-context dynamic optically reconfigurable

gate array," in *Proc. of the 3rd NASA/ESA Conference on Adaptive Hardware and Systems (AHS'09)*, Jul. 2009, pp. 120-125

[76] T. Kenter, H. Schmitz, and C. Plessl, "Exploring trade-offs between specialized dataflow kernels and a reusable overlay in a stereo matching case study," *International Journal of Reconfigurable Computing*, vol. 2015, no. 12, Jan. 2015, pp. 1-24

[77] A. Chen, "Emerging nonvolatile memory (NVM) technologies," in *Proc. of the 45th European Solid State Device Research Conference (ESSDERC'15)*, Sep. 2015, pp. 109-113

[78] W. -P. Lin *et al.*, "A nonvolatile look-up table using ReRAM for reconfigurable logic," in *Proc. of the IEEE Asian Solid-State Circuits Conference (A-SSCC'14)*, Nov. 2014, pp. 133-136

[79] *7 Series FPGAs overview, DS180 (v2.0)*, Xilinx Inc., San Jose, CA, USA, Sep. 2016

[80] *7 Series FPGAs configuration user guide, UG470 (v1.11)*, Xilinx Inc., San Jose, CA, USA, Sep. 2016

[81] *Vivado design suite user guide, high-level synthesis, UG902 (v2016.1)*, Xilinx Inc., San Jose, CA, USA, Apr. 2016

[82] *Virtex-5 FPGA user guide, UG190 (v5.4)*, Xilinx Inc., San Jose, CA, USA, Mar. 2012

[83] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, Aug. 2014, pp. 1283-1295

[84] J. Rajendran, O. Sinanoglu, and R. Karri, "Regaining trust in VLSI design: design-for-trust techniques," *Proceedings of the IEEE*, vol. 102, no. 8, Aug. 2014, pp. 1266-1282

[85] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, “Hardware Trojan attacks: threat analysis and countermeasures,” *Proceedings of the IEEE,* vol. 102, no. 8, Aug. 2014, pp. 1229-1247

[86] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, “Trustworthy hardware: identifying and classifying hardware Trojans,” *IEEE Computer,* vol. 43, no. 10, Oct. 2010, pp. 39-46

[87] M. Tehranipoor *et al.*, “Trustworthy hardware: Trojan detection and design-for-trust challenges,” *IEEE Computer*, vol. 44, no. 7, Jul. 2011 pp. 66-74

[88] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, “Towards Trojan-free trusted ICs: problem analysis and detection scheme,” in *Proc. of the Design, Automation and Test in Europe (DATE’08),* Apr. 2008, pp. 1362-1365

[89] T. F. Wu, K. Ganesan, Y. A. Hu, H. -S. P. Wong, S. Wong, and S. Mitra, "TPAD: hardware Trojan prevention and detection for trusted integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD),* vol. 35, no. 4, Apr. 2016, pp. 521-534

[90] Y. Jin and Y. Makris, “Hardware Trojan detection using path delay fingerprint,” in *Proc. of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST’08),* Jun. 2008, pp. 51-57

[91] I. Exurville, L. Zussa, J. -B. Rigaud, and B. Robisson, “Resilient hardware Trojans detection based on path delay measurements,” in *Proc. of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST’15),* May. 2015, pp. 151-156

[92] B. Cha and S. K. Gupta, “Trojan detection via delay measurements: a new approach to select paths and vectors to maximize effectiveness and minimize cost,” in *Proc. of the Design, Automation and Test in Europe (DATE’13),* Mar. 2013, pp. 1265-1270

[93] K. Xiao, X. Zhang, and M. Tehranipoor, “A clock sweeping technique for detecting hardware Trojans impacting circuits delay,” *IEEE Design & Test,* vol. 30, no. 2, Apr. 2013, pp. 26-34

[94] Y. Cao, C. -H. Chang, and S. Chen, “A cluster-based distributed active current sensing circuit for hardware Trojan detection,” *IEEE Transactions on Information Forensics and Security (T-IFS),* vol. 9, no. 12, Dec. 2014, pp. 2220-2231

[95] A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD),* vol. 33, no. 12, Dec. 2014, pp. 1792-1805

[96] D. Forte, C. Bao, and A. Srivastava, “Temperature tracking: an innovative run-time approach for hardware Trojan detection,” in *Proc. of the 32nd IEEE/ACM International Conference on Computer-Aided Design (ICCAD’13),* Nov. 2013, pp. 532-539

[97] C. Bao, D. Forte, and A. Srivastava, "On reverse engineering-based hardware Trojan detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD),* vol. 35, no. 1, Jan. 2016, pp. 49-57

[98] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri, “Fault analysis-based logic encryption,” *IEEE Transactions on Computers (T-C),* vol. 64, no. 2, Feb. 2015, pp. 410-424

[99] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. of the 49th ACM/EDAC/IEEE Design Automation Conference (DAC'12),* Jun. 2012, pp. 83-89

[100] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'15),* May. 2015, pp. 137-143