2.5D and 3D I/O designs for energy-efficient memory-logic integration towards thousand-core on-chip

Sai Manoj Pudukotai Dinakarrao

2015

Sai Manoj Pudukotai Dinakarrao. (2015). 2.5D and 3D I/O designs for energy-efficient memory-logic integration towards thousand-core on-chip. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65486>

<https://doi.org/10.32657/10356/65486>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**2.5D AND 3D I/O DESIGNS FOR
ENERGY-EFFICIENT MEMORY-LOGIC
INTEGRATION TOWARDS THOUSAND-
CORE ON-CHIP**

SAI MANOJ PUDUKOTAI DINAKARRAO
**SCHOOL OF ELECTRICAL & ELECTRONIC
ENGINEERING**

2015

**2.5D AND 3D I/O DESIGNS FOR
ENERGY-EFFICIENT MEMORY-LOGIC
INTEGRATION TOWARDS THOUSAND-
CORE ON-CHIP**

SAI MANOJ PUDUKOTAI DINAKARRAO

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technology University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

Acknowledgement

First and foremost, I would like to express the deepest gratitude to my advisor Prof. Yu Hao, for his invaluable support both in my research and life during all years of my work in NTU. I have been extremely fortunate to work under his supervision and I deeply appreciate his time, patience and support throughout my Ph.D. career. His assistance, constructive comments, and technical insights have always been most helpful in addressing my research issues. He taught me to set high standards for myself and made me realize the importance of global competence. For me, he sets a good example of what a researcher should be like: have enthusiasm towards research, always think out of box, and work hard to solve problems.

I would like to express my special thanks to Dr. Yu Mingbin, Dr. Roshan Weerasekera, Prof. Siek Liter, Prof. Tan Chuan Seng, and Prof. Chang Chip-hong, for giving me generous amount of time whenever I needed some help. I also want to thank the Virtus and EEE GPO staff Gek Eng, Dorothy, David Robert, Karen, Chow Kam Wah, Christina, and many others who have been really friendly and patient to help me many times on administrative matters. I would like to express my thanks to my wonderful colleagues, Wang Kanwen, Xu Dongjun, Huang Hantao, Wang Fei, Song Yang, Li Nan, Shang Yang, Huang Xiwei, Wu Sih-Sian, Wang Yuhao, Li Peng, Ma shunli, Guo Jing, Shen Shanlan, Jin Xin, Wang Xiaolong, Chen Shuai, Yang Chang, Fei Wei, Sunil Gavaskar, Abhik Das, Sachin Kumar, Syed Ali, and many others who accompanied me to walk through my most memorable years.

Last but not the least, I owe my deepest gratitude to all my family members, especially my parents, my sister and my brother-in-law, who have done everything for me, including sacrificing the joys of their own lives so that I could be happy and successful in my life. Thanks for being with me on each and every step of my life. It is their unconditional love that motivates me to set higher targets.

Abstract

In the past few decades, the design of computers has been primarily driven by improving performance with faster clock frequency of single-core processor using transistor scaling. The transistor scaling towards high performance of fast clock frequency is, however, stuck recently due to the constraint of power density (or thermal reliability). By exploiting parallelism, multi-core processor based design of computers has emerged to scale up performance of throughput under power budget. As such, the scaling paradigm has changed to integrate as many processor cores as possible on one single chip. In the traditional 2D based memory-logic integration, the scalability of many-core integration is limited by the communication between the cores and memory to access via I/O interconnections, which pose stringent requirements for high utilization efficiency of both bandwidth and power.

Stacking of one layer (with core or memory blocks) above the other with short-distance RC interconnects of through-silicon vias (TSVs), termed as *3D integration* has become one promising alternative many-core integration technique with high bandwidth. It is however constrained by severe thermal reliability concern due to poor heat dissipation in 3D. The other alternative for many-core integration is to place core and memory blocks on one common substrate, which are further interconnected with the aid of middle-distance transmission lines (underneath the substrate) of through-silicon interposers (TSIs), termed as *2.5D integration*.

In order to explore the fundamental challenges of I/O interconnections by 3D TSVs and 2.5D TSIs for many-core integration, in this thesis, we have studied various design aspects starting from device level modeling to system level management with the following works done.

- Firstly, we have developed physical models of 3D TSV and 2.5D TSI I/Os.
 1. We first study 3D TSV I/Os, which are short-distance RC interconnects with inverter buffers. The traditional physical model treats TSV as a resistor with linear electrical-thermal dependence, which is not accurate. With the stacking of layers the heat starts to accumulate in the top layer resulting in

temperature gradient. Further, due to difference in operating temperatures and coefficient-of-thermal-expansion (CTE) of TSV and silicon substrate, TSVs exert mechanical stress on the substrate, resulting in stress gradient. A detailed study of TSV structure reveals that the existence of liner material around TSV metal-fill, used to avoid ion diffusion, can significantly affect the electrical property of TSVs. The liner material results in the formation of MOS capacitance (MOSCAP), which varies nonlinearly with temperature when applied with biasing voltage. As such, with the consideration of liner material, a nonlinear electrical-thermal-mechanical coupled TSV delay model is proposed in this thesis that can provide accurate electrical delay and also delay variations under temperature and stress gradient.

What is more, to alleviate the TSV delay variation, dummy TSV based insertion is applied to balance thermal and stress gradients such that the resulting delay variation can be minimized. Taking the design of a 3D clock-tree as a case study, a nonlinear optimization problem is formulated for the insertion of dummy TSVs to reduce the clock-skew. Various 3D clock-tree benchmarks are utilized in experiments. It has been observed that with the insertion of dummy TSVs, clock-skew can be reduced by 61.3% on average with the proposed nonlinear electrical-thermal-mechanical delay model.

2. We also study 2.5D TSI I/Os, which are middle-distance transmission lines (T-line) with designed buffers. A thermal-resilient nature can be observed in the 2.5D integration using TSI I/Os, when compared to the 3D integration using TSV I/Os. Furthermore, to overcome channel loss, the design of low-voltage differential signalling (LVDS) and current-mode-logic (CML) buffers with pre-emphasis stage is proposed. The post-layout simulation results in UMC 65nm CMOS process have shown an energy-efficiency of $0.48pJ/b$ and $0.075pJ/b$ for LVDS and CML buffers respectively.
- Secondly, we have developed power I/O management for 2.5D/3D memory-logic integration towards thousand-core on-chip. The power I/O management is to explore reconfigurability by studying the data-pattern of power-signatures. To avoid the dark-silicon dilemma, a dynamic-voltage scaling (DVS) technique is proposed. We employ on-chip single-inductor multiple-output (SIMO) power converters for better power delivery efficiency and to alleviate the area overhead. A reconfigurable switch network is utilized to connect the power converters and cores based on the data-pattern of power-signature characteristics. Cores are initially clustered based on the magnitude of power-signature and the power converters are

allocated accordingly, called space multiplexing. Furthermore, inside one cluster, based on the phase of the power-signature, a power converter is assigned in a time multiplexed manner. In addition, a demand-response based workload scheduling is performed to reduce the peak-power and achieve workload balance. The proposed power management is verified by system models with physical design parameters and benched power traces of workloads. For a 64-core, experiment results have shown 40.53% peak-power reduction and $2.5\times$ balanced workload along with 42.86% reduction for the required number of power converters.

- Lastly, we have developed signal I/O management for 2.5D/3D memory-logic integration towards thousand-core on-chip.
 1. The first signal I/O management is to explore reconfigurability by studying memory-access data-patterns for communication efficiency and bandwidth improvement. We propose a design of memory controller to effectively manage the memory-logic communication by utilizing the available 2.5D TSI I/Os. To match the huge demand to access the memory by the cores, the demands are classified in space based on their magnitude and are connected to the corresponding port of the memory controller. At each port, cores are assigned with priority, based on which I/O channels are allocated in a time multiplexed manner. This space-time multiplexing scheme is deployed inside the memory controller. The proposed reconfigurable data-pattern aware memory controller with space-time multiplexed 2.5D TSI I/O is verified by a system-level simulator with benchmarked workloads, shows up to 58.85% bandwidth balancing and 11.90% QoS improvement.
 2. The second signal I/O management is to explore the trade-off between quality-of-service (QoS) and I/O voltage-swing for improvement of power efficiency. Communication between memory and logic blocks contributes significant amount to overall power consumption of the system. Using a uniform voltage-swing may incur large power and is not needed when a certain amount of bit-error-rate (BER) can be tolerated. A self-adaptive output-voltage swing adjustment is introduced for the energy-efficient I/O communication. Instead of transmitting a signal with a large voltage swing, a Q-learning based I/O management is deployed to adaptively tune the I/O output-voltage swing under constraints of both power and BER. Furthermore, to overcome slower convergence of conventional Q-learning, an accelerated Q-learning is applied to adaptively tune the output-voltage swing.

Experimental results show that the adaptive 2.5D I/Os designed in 65nm C-MOS can achieve an average of 12.5mW I/O power, 4GHz bandwidth and 3.125pJ/bit energy-efficiency for one channel under 10^{-6} BER. With the use of conventional Q-learning and further accelerated Q-learning, we can further achieve 12.95% and 18.89% power reduction and 14% and 15.11% energy-efficiency improvement compared to the use of the uniform output-voltage swing based I/O communication.

In summary, the main contribution of this thesis can be summarized as follows. Firstly, the physical models for TSV and TSI are developed considering the nonlinear MOSCAP due to the liner material surrounding the I/O interconnect. Secondly, the data-pattern aware I/O management is developed with the reconfigurable switch network under the control of a space-time multiplexing algorithms. Lastly, reinforcement-based Q-learning algorithms based I/O management is developed to balance the trade-off between the power budget and BER constraints. As a conclusion, the proposed 2.5D and 3D I/O designs have shown a great potential for many-core integration towards energy-efficient thousand-core on chip in the near future.

Contents

Acknowledgement	i
Abstract	ii
List of Figures	x
List of Tables	xvi
1 Introduction	1
1.1 Thousand-core On-chip	1
1.2 State-of-Art Many-core Microprocessors	4
1.3 Memory-logic Integration	6
1.3.1 2D Integration Challenges	6
1.3.2 3D Integration	10
1.3.3 2.5D Integration	12
1.4 Contributions of This Work	12
1.5 Organization of the Thesis	14
2 Fundamentals and Literature Review	15
2.1 3D Integration	15
2.1.1 Overview	15
2.1.2 Challenges	17
2.2 2.5D Integration	22
2.2.1 Overview	23
2.2.2 Challenges	25
2.3 Summary	28
3 Device Model	29
3.1 Introduction	29
3.2 Nonlinear MOSCAP Model	30
3.3 TSV Device Model	33
3.3.1 Electrical Model	33
3.3.2 Mechanical Model	35

3.3.3	Delay Model	39
3.3.4	Power Model	43
3.4	TSI Device Model	43
3.4.1	Delay Model	44
3.4.2	Power Model	47
3.5	Summary	49
4	Physical Designs	51
4.1	Introduction	51
4.2	3D Clock-tree Design	52
4.2.1	Problem Formulation	52
4.2.2	Clock-skew Reduction	57
4.3	2.5D I/O Design	66
4.3.1	LVDS I/O Buffer	67
4.3.2	CML I/O Buffer	69
4.3.3	Simulation Results	69
4.4	Summary	73
5	Power I/O Management	75
5.1	Introduction	75
5.2	System Power Model	77
5.2.1	Core and DRAM Power Model	77
5.2.2	System Power Breakdown	78
5.3	3D Power I/O Management	82
5.3.1	System Architecture	83
5.3.2	Problem Formulation	85
5.4	ILP based Management	87
5.5	Space-Time Multiplexing based Management	88
5.5.1	Adaptive Clustering	89
5.5.2	Scheduling of Workloads	101
5.6	Summary	106
6	Signal I/O Bandwidth Management	109
6.1	Introduction	109
6.2	2.5D Bandwidth Management	110
6.2.1	System Architecture	111
6.2.2	Problem Formulation	112
6.3	Signal I/O Bandwidth Management	113

6.3.1	Memory-access Data-pattern	113
6.3.2	Reconfigurable Memory Controller	116
6.4	Summary	121
7	Signal I/O Voltage-swing Management	123
7.1	Introduction	123
7.2	2.5D Voltage-swing Management	124
7.2.1	System Architecture	124
7.2.2	Problem Formulation	126
7.3	Signal I/O Voltage-swing Management	127
7.3.1	Reinforcement Q-learning	127
7.3.2	Adaptive I/O Circuit Design	138
7.4	Summary	143
8	Conclusion and Future Work	145
8.1	Conclusion	145
8.2	Recommendations for Future Work	146
A	Publication list	149
A.1	Journal	149
A.2	Conference	150
A.3	Workshop/Poster	151
A.4	Miscellaneous	152

List of Figures

1.1	How to build one Exa-scale data center	2
1.2	Scaling of single-core electronic computer by VLSI integration	3
1.3	Can we integrate an Exa-scale data center on chip? [1]	3
1.4	ITRS Roadmap of thousand-core on chip	3
1.5	ITRS projection for power index and RC-delay for 2D interconnects [2]	4
1.6	(a) MIT 16-core RAW microprocessor [3]; (b) Intel 80-tile microprocessor [1]; (c) Intel 48-tile microprocessor [4]	6
1.7	Typical multi-core system architectural view	7
1.8	(a) PCB based interconnection of two dies with backplane trace; (b) corresponding channel loss	8
1.9	(a) Positive loop between leakage current and temperature; (b) trend showing leakage current versus temperature	9
1.10	Basic architecture for (a) 3D integration by vertical TSV interconnect; (b) 2.5D integration by TSI interconnect	10
2.1	Simplified TSV fabrication procedure	16
2.2	Simplified flow to form (a) 3D IC; (b) 2.5D IC from traditional 2D ICs .	16
2.3	(a) 3D integration of 64-cores and 64-memory blocks; (b) 64-core tier layout; (b) 256KB SRAM tier layout [5]	17
2.4	Power management schemes: (a) per-chip DVFS; (b) per-core DVFS; and (c) clustered DVFS	19
2.5	System architecture for application aware DVFS [6]	20
2.6	3D multi-core architecture with per-core power converters [7]	21
2.7	Tiles connected by mesh network with each color representing the voltage/frequency island [8]	22
2.8	Simplified process of TSI Fabrication [9]	23
2.9	Composition of components in a typical TSI [10]	24
2.10	(a) Two ARM Cortex-A9 dual core processors connected by 65nm silicon interposer; (b) Cross section of 2.5D TSI [11]	25

2.11	Low swing data link with SRLR embedded in each router [12]	26
2.12	Target 3D hardware architecture [13]	26
2.13	Memory-logic integrated architecture with 3D memory controller [13] .	27
2.14	3D DRAM sub system architecture with single vertical channel [14] . .	28
3.1	(a) Signal TSV and dummy TSV in 3D-IC; (b) 3D view of a circular TSV; (c) equivalent simplified RC-circuit of a TSV	31
3.2	(a) Typical <i>CV</i> curve of nonlinear MOSCAP model for signal TSV; (b) exertion of stress on from TSVs to transistors/drivers in a unit square-grid	32
3.3	(a) Variation of TSV capacitance with temperature; (b) variation of TSV delay with temperature for different TSV bundles based on (3.3)	35
3.4	FEA simulation curves of the thermo-mechanical stress exerted by a single Cu-TSV on Si substrate as a function of distance from the edge of TSV	36
3.5	(a) Variation of TSV stress with distance; (b) variation of TSV stress with temperature and TSV density based on (3.4)	37
3.6	(a) Variation of NMOS/PMOS carrier mobility with distance under TSV stress based on (3.6); (b) variation of driver delay with distance under TSV stress	39
3.7	(a) TSV with buffer at its ends; (b) delay model of 3D clock-tree with nonlinear electrical-thermal-mechanical coupled model	39
3.8	Signal delay of one single TSV under different technologies based on (3.12)	42
3.9	2.5D integration of cores on one common substrate using TSI	44
3.10	Temperature dependent delay model of one TSI STL or DTL I/O channel	45
3.11	Delay and power comparison under different length for (a) TSV I/Os based on (3.8) and (3.15); (b) TSI I/Os based on (3.21) and (3.22) . . .	47
3.12	Delay and power comparison under different temperature for TSV I/Os (based on (3.8) and (3.15)) and TSI I/Os (based on (3.21) and (3.22)) . .	48
3.13	Total power and energy efficiency with different bandwidths: (a) at 25°C; (b) at 100°C based on (3.15) and (3.22)	48
4.1	3D clock-tree distribution network at different tiers: (a) Clock-tree with 14 TSV bundle locations in a H-tree; (b) clock-tree with 28 TSV bundle locations in a H-tree; (c) layer configuration under non-uniform temperature distribution	53
4.2	(a) 3D heat-removal path by dummy TSVs; (b) 3D-view of dummy TSV insertion	54

4.3	Temperature gradient reduction in a 4-layer 3D clock-tree with dummy TSVs under different power densities based on (4.4)	56
4.4	(a) Setup of TSV distributions for stress gradient calculation; (b) variation of stress gradient reduction with TSV density and temperature gradient	57
4.5	Clock-skew reduction before and after insertion of dummy TSVs	63
4.6	3D clock-tree after insertion of dummy TSV (black dots) with balanced clock-skews for: (a) Tier 0; (b) Tier 1; (c) Tier 2; (d) Tier 3	64
4.7	2.5D TSI I/O interconnection between core and memories its cross section view	66
4.8	Typical LVDS I/O architecture [15]	67
4.9	A LVDS transmitter with pre-emphasis technique	68
4.10	A schematic of the proposed CML I/O interface	69
4.11	TSI realized as T-line	70
4.12	Layout of two I/O Interface	70
4.13	(a) Simulation waveform of LVDS output eyediagram; (b) Waveform of LVDS passed through TSI connection	70
4.14	(a) Simulation waveform of CML output eyediagram; (b) Waveform of CML passed through TSI connection	71
4.15	EM simulation setup for TSV I/O	71
4.16	Insertion loss for $10\mu m$ diameter TSV I/O based on EM simulations	72
4.17	Layout of TSI I/O	72
4.18	Insertion loss for $3mm$ TSI I/O based on EM simulations	73
5.1	Power breakdown of: (a) 2.5D integration at $25^{\circ}C$; (b) 2.5D integration at $120^{\circ}C$; (c) 3D integration at $25^{\circ}C$; (d) 3D integration at $120^{\circ}C$ based on simulations in gem5	79
5.2	(a) Thermal power and electrical power cycles; (b) thermal runaway and threshold temperature	80
5.3	Thermal runaway analysis of data server by 3D and 2.5D integrations from simulations	81
5.4	3D reconfigurable power switch network for demand-supply matching between on-chip multi-output power converters and many-core microprocessors	84
5.5	Functional units of space-time multiplexing based power management for DVS with SIMO power converter	85
5.6	Space and time multiplexing with on-chip SIMO power converters	86

5.7	Power-signature pattern extracted in each time-slot of control-cycle . . .	89
5.8	Extraction of power-signature for workloads based on (5.10)	90
5.9	Runtime power tracking and prediction for benchmark <i>gcc</i> : (a) power prediction using (5.11); (b) voltage-level transition	91
5.10	Resource allocation by adaptive clustering: grouping by power-signature magnitude and subgrouping by power-signature phase	95
5.11	Results of adaptive clustering for a 32-core microprocessor in two consecutive control-cycles	100
5.12	(a) Workload before demand-response scheduling (b) workload after demand-response scheduling with peak reduction and balancing with slack calculated based on (5.20)	103
5.13	Peak-power reduction for 4 subgroups of 64-core case	105
6.1	Side-view of 2.5D many-core microprocessor and main memory integration by TSI I/O channels	111
6.2	Memory-access data-pattern aware reconfigurable memory controller with space-time multiplexing	111
6.3	Analysis of memory-access data pattern	114
6.4	SPEC 2006 memory-access data-patterns: (a) memory access for execution time of 10ms; (b) memory access for one control-cycle of 1ms . .	115
6.5	Illustration of LLC MPKI memory-access data-pattern: (a) bandwidth; (b) priority	118
6.6	Adaptive clustering of cores at two consecutive control-cycles	119
6.7	Variation in number cores allocated to a cluster with control-cycles . . .	120
6.8	Bandwidth balance across TSI I/O channels with: (a) 16-cores; (b) 64-core under randomly distributed benchmarks of SPEC 2006, Parsec and Phoenix	121
6.9	Communication QoS efficiency improvement by STM I/O management for 16-core based on (6.4)	122
7.1	(a) Core-memory integration by 2.5D TSI I/O interconnect and its cross sectional view; (b) adaptive tuning I/O based on error checking and correction	125
7.2	Trade-off between BER and power under various benchmarks: (a) <i>FFT</i> benchmark; (b) <i>Filetransfer</i> benchmark; (c) <i>bzip2</i> benchmark; (d) <i>gzip</i> benchmark.	126
7.3	State transition based on conventional Q-learning	130

7.4	Flowchart showing conventional Q-learning based self-adaptive voltage swing tuning	131
7.5	Example of one adaptive I/O control by basic Q-learning	132
7.6	State transition based on accelerated Q-learning	133
7.7	Flowchart of accelerated Q-learning based self-adaptive voltage swing tuning	135
7.8	Example of one adaptive I/O control by accelerated Q-learning.	136
7.9	(a) Transmitter with 8:1 serializer; (b) adaptive tuning of driver tail current; (c) TSI realized by a T-line	139
7.10	(a) The architecture of DLL based CDR; (b) voltage controlled delay cell in the DLL	140
7.11	Encoding and decoding at transmitter and receiver	141
7.12	(a) Data encoding and decoding; (b) transmitted data format	141
7.13	Eye diagram of output data with different driver current (or output-voltage swing) levels: (a) $2mA$; (b) $3mA$; (c) $4mA$ based on simulations .	142
7.14	The eye-diagrams under adaptive current (or power) adjustment by output-voltage swing tuning	143
8.1	Overall system architecture for multi-core memory 2.5D integrated microprocessor.	147
8.2	Fabricated multi-core processor with memory and accelerator blocks. .	147

List of Tables

2.1	Off-chip [16] vs on-chip [17] power converter	18
3.1	TSV Electrical Parameters	34
4.1	3D clock-skew reduction by linear and nonlinear delay models	65
5.1	System setup for data server system components	79
5.2	System settings of 3D many-core microprocessors, on-chip power converters, TSVs and power switches	97
5.3	Comparison of average power consumption and controller runtime for STM by ILP and STM by adaptive clustering	98
5.4	Adaptive clustering result for 64-core microprocessor	100
5.5	Comparison of number of allocated power converters under different power management schemes	101
5.6	Demand-response based workload scheduling result for 32-core microprocessor	104
5.7	Comparison of number of allocated power converters with and without workload balancing	105
5.8	Peak reduction and workload balancing by demand-response scheduling	106
6.1	System parameters	119
6.2	Number of requests served and achieved QoS	122
7.1	System settings for memory-logic integration with TSI I/O	125
7.2	Power and run-time comparisons for the adaptive tuning with Q-learning under various benchmarks	137

Chapter 1

Introduction

1.1 Thousand-core On-chip

With the advancement in cloud computing with big-data analytic capability, the amount of data to be handled by the data centers are increasing tremendously and reaching Exa-scale (10^{18} Flops) [18]. To process such large amount of data, numerous multi-core microprocessor cores can be integrated on a single chip for the Exa-scale data processing as accelerators. As one example shown in Figure 1.1, let's calculate the resource requirement for a data center to have Exa-scale processing capability. We assume that each core can have four floating point units (FPUs), each having 1.5GHz processing speed. As such, each core can reach a performance of 6GFlops. To perform 1Exa-Flop operations, a large number of such cores and memory blocks are needed to be integrated. The integration can be arranged in a hierarchical manner, namely by chips, nodes and racks. One needs to integrate 742 cores on a single chip. Moreover, a set of 16 DRAMs, each DRAM of 1GB capacity and one chip form a node. Furthermore, 12 such nodes connected with routers form a group. As such, one needs to combine 32 groups of processing capability 1.7Peta-Flop as a rack. Lastly, in order to achieve 1Exa-Flop performance, 583 such racks are needed to be integrated together. This setup demands large amount of bandwidth (few tens of Gbps) and 68MW power, with 20,000sft area. The resource to build many of Exa-scale data centers will obviously require too much consumption of bandwidth, power and space that may be beyond the capability of the current human society.

The similar situation actually happened in history when humans built their first electronic computer. The first electronic computer i.e., electronic numeric integrator and calculator (ENIAC) [19] was built in 1946, made of vacuum tubes, shown in Figure 1.2. It had a processing capability of only 5KHz, but consumed power of 150KW within an

Figure 1.1: How to build one Exa-scale data center

area of 167m^2 . Due to the advance of CMOS transistor scaling and the advancement in VLSI integration, one of the recent (2008) microprocessors from Intel, i7 quad-core processor as shown in Figure 1.2, only occupies an area of 263mm^2 and consumes a power of 130W, but with a processing capability of 1GHz. As such, it motivates us with a similar question: can we integrate an Exa-scale data center on chip, shown in Figure 1.3? Figure 1.4 shows the prediction of the number of cores that will be integrated on a chip. One can observe that it is predicted that thousands of cores will be integrated on a single chip, along with large amount of memory in the next 10 years.

With the integration of thousand-cores on a single-chip, a large amount of power is consumed for communication between cores compared to the computation power. The use of traditional 2D interconnects such as on-chip wire, off-chip PCB trace and etc., will result in a large power dissipation and latency with degraded scaling performance for integrating thousand-core on-chip [20]. The ITRS road map [2] for the power index and RC-delay of a 2D on-chip interconnects for 1mm length of Copper (Cu) is presented in Figure 1.5, which indicates that in the next few years, the power index and delay of global interconnects both become non-scalable. Moreover, 2D off-chip interconnects such as back-plane PCB traces are obviously with lossy channels that require overdesign of I/O equalization with huge power overhead. On the other hand, optical interconnects can provide high-speed communication but always come

Figure 1.2: Scaling of single-core electronic computer by VLSI integration

Figure 1.3: Can we integrate an Exa-scale data center on chip? [1]

Figure 1.4: ITRS Roadmap of thousand-core on chip

Figure 1.5: ITRS projection for power index and RC-delay for 2D interconnects [2]

with an additional cost of optical-to-electronic conversion with no CMOS based light source, detector and modulator. In this thesis, to meet the high-speed, low power and high bandwidth demands, the future thousand-core memory-logic integration by 3D integration is explored. 3D integration by short-distance through-silicon vias (TSV) interconnects [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 5, 32, 33, 34, 35, 36, 37, 38] and 2.5D integration by middle-distance through-silicon interposer (TSI) interconnects [10, 20, 39, 40, 41] can be cost efficient in meeting large bandwidth, low power and latency requirements.

In this chapter, we first present state-of-the-art many-core microprocessor designs, followed by the analysis and comparison for memory-logic integrations of 2D, 3D and 2.5D, respectively.

1.2 State-of-Art Many-core Microprocessors

Due to simplicity of individual cores in a multi-core microprocessor, the multi-core microprocessor outperforms a single large core microprocessor in terms of area, power, speed, number of functional units and energy efficiency [42]. A multi-core processor is power efficient because of the flexibility to utilize multiple clock frequencies and voltage levels proportional to the demands. However, use of higher voltage levels may increase the power consumption nearly cubic to the increase in frequency [43]. In the past few years, a large number of many-core microprocessors have been developed. In this section, we provide a glimpse of few many-core microprocessor examples.

In a many-core microprocessor, the communication between core-core/memory is

often the bottleneck. The interconnection for communication can be by centralized structures like bus interconnects, ring structures or by a fixed interconnection. This kind of communication structures may be feasible for a small number of cores, but may not be scalable for thousand-core chips, which is expected to happen in the next decade or so. A distributed communication can be performed between cores, with the help of routers. Tiled architectures are introduced, which are composed of small computational cores instead of one large core. The small cores are simple, area- and energy-efficient compared to one large core. Few of the multi-core processors include 16-core Raw processor [3] which is commercialized in Tiler, TRIPS [44], Tiler's TILE 64 [45, 46] and Intel's Tera-scale research processor [47], SPARC SoC processor from Sun Microsystems [48], 48/80-core microprocessors by Intel [1, 4], and 64-core microprocessor by IBM [49] are few of the famous multi-core processors. It is also good to mention that Nvidia Tesla K80 servers are designed with nearly 2GPUs with each consisting of 2496 CUDA cores [50]. Few of the tiled multi-core processors from industry can be seen in Figure 1.6. Each of the cores is connected with a router, for the purpose of effective communication [43]. Tiled cores with distributed computation and communication structures are efficient and scalable [45, 46, 47]. Unlike centralized on-chip communication where a large global interconnect is used, distributed communication is only routed through the on-chip network and uses only components between the source and destination. In tiled microprocessors, the efficiency can be further improved by deciding the mapping of tasks to cores and how best to utilize them. In a tiled architecture, each core can be of different size. Addition of new cores in tiled design is easy compared to a centralized interconnect scheme [43].

A 16-core Raw microprocessor [3], which is a prototype tiled multi-core microprocessor is shown in Figure 1.6(a), fabricated by MIT in 2003. This Raw microprocessor is commercialized in Tiler microprocessors. The Raw microprocessor in [3] is designed at 180nm technology node outperforms P3 processor [51] with an average throughput advantage of $10.8\times$ (by cycles) and $7.6\times$ (by time). This high performance is achieved due to the high pin bandwidth available to the off-chip memory [3].

An NoC architecture of 80-tile architecture implemented at 65nm CMOS process and arranged in the form of a 2D mesh (10×8) is shown in Figure 1.6(b). The chip is expected to run at a maximum frequency of 3.13GHz and peak performance of 1.0TFLOPS at 1V and 4GHz maximum frequency with 1.28TFLOPS peak performance at 1.2V, respectively. The whole chip consumes an area of $275mm^2$, with each tile consuming $3mm^2$ of area. A 48-core processor is fabricated by Intel [4] in 2010. The processor is implemented in a 45nm CMOS process with a total die area $567mm^2$, with each tile having 2 cores and an area of $18.7mm^2$, as shown in Figure 1.6(c). An operating fre-

Figure 1.6: (a) MIT 16-core RAW microprocessor [3]; (b) Intel 80-tile microprocessor [1]; (c) Intel 48-tile microprocessor [4]

quency of 1GHz for core and 2GHz for router at voltage of 1.14V is observed.

1.3 Memory-logic Integration

For many-core microprocessor design, one needs to integrate the cores with memory through interconnect. The cores load data from the memory, perform computation and then write back data to the memory. Cores need both on-chip and off-chip interconnects to access the on-chip memory (cache) and off-chip memory, resulting in a long-distance communication and larger power consumption in 2D. More I/O pins and more channels are required to access memory for high bandwidth requirement. In the following, we show the challenges faced in the traditional memory-logic integration, and further present the solutions by the proposed 2.5D and 3D memory-logic integrations.

1.3.1 2D Integration Challenges

Undoubtedly the introduction of many-core microprocessors has revolutionized the design of computers. For example, workloads on processors can be distributed and designers can start thinking of parallelizing workloads to increase the processing throughput. Unfortunately, the available 2D integration technique is inefficient for many-core microprocessor at a large scale. Few of the challenges in traditional 2D integration are briefly discussed below.

1.3.1.1 Scalability

Reduction in transistor size has fueled the processing speeds of logic blocks. Memory-logic integration in traditional 2D manner will have limited on-chip memory with a large off-chip memory due to the large area occupied by cores and other logic blocks

Figure 1.7: Typical multi-core system architectural view

(Figure 1.7). With the advancement in technology nodes, interconnect delay started to dominate the delay, because 2D RC interconnects cannot be scaled at the same rate as gates. It is expected that the latency due to interconnects can be cut down by nearly 50% by moving off-chip memory blocks to on-chip [52]. Furthermore, to cope with increasing application demands, a large amount of on-chip memory is projected by ITRS [2]. However the amount of available space is limited in 2D integration with cores integrated.

1.3.1.2 Channel Loss

The off-chip interconnection between cores and memory blocks in 2D integration is typically performed by printed circuit board (PCB) with backplane [53] containing sockets into which other boards and corresponding components can be plugged in. As discussed, this methodology is not scalable for large number of cores due to routing and wire-length limitations. More importantly, long traces ($\geq 25cm$) and non-ideal vias on the PCB can cause severe loss on the backplane. To compensate the losses and achieve high data rate, current starved circuits are needed with area overhead and also power overhead.

Figure 1.8(a) shows the interconnection of multiple chips using the wire-line backplane PCB interconnects. The channel loss for the corresponding integration is depicted in Figure 1.8(b). One can observe nearly $24dB$ channel loss at a frequency of $5GHz$ [53]. Further, due to the available limited area, it is not possible to accommodate a large number of I/O pins, hence the number of I/Os are as well limited and puts upper bound on the bandwidth available in 2D interconnect.

Figure 1.8: (a) PCB based interconnection of two dies with backplane trace; (b) corresponding channel loss

1.3.1.3 I/O Circuit Design

A proper design of the interface channel must adhere to a complex set of specifications and constraints. These include voltage levels and noise, bit-error rate (BER), signal jitter and slew rate. Proper design and optimization of these circuits are needed to meet the design specifications at an acceptable cost in terms of area and power [54, 55, 56, 57, 58].

To meet the performance and signal integrity constraints for the I/O drivers, good models of the package and board are needed to account for the capacitive loading and coupling as well as the inductive coupling in the system [28, 36, 37]. Early in the design cycle where the package is not routed yet or where no board models exist, good estimations are needed. In high-end designs, it is no longer sufficient to use ad-hoc metrics like rules of thumb to make decisions on the type and size of the I/Os as well as on the capacitive and inductive values of the load that the driver sees. Good virtual models that can capture the package and board effects are needed. Moreover, complicated compensation of data and clock signals at receivers is required.

1.3.1.4 Thermal Management

The operating temperature of the chips depends on the ambient temperature, power density of cores and the heat-dissipation capacity of the chip [32, 59, 39]. In a multi-core microprocessor, a large number of hot-spots can be observed due to larger power density. When the on-chip temperature increases beyond the specified operating temperature, the functionality of the chip is questionable. Moving workloads from a core with hot-spots to another may not always be an efficient solution and may involve overhead.

Moreover, if one observes the state-of-the-art microprocessors, memory occupies more area than processing units and consumes more power as well due to static power dominated by leakage current. With the scaling down of transistor size, the transistor cannot be completely turned off. As such, even in the low/off state, leakage current is present. This leakage current can form a positive feedback loop with temperature to

Figure 1.9: (a) Positive loop between leakage current and temperature; (b) trend showing leakage current versus temperature

further increase temperature, resulting in a thermal runaway failure [60, 61, 39].

Figure 1.9 shows the occurrence of a thermal runaway failure with the increase of temperature and leakage current in a memory-logic integrated system. When the on-chip temperature increases, it forms a positive feedback loop with the leakage current and further increases the temperature. Beyond a certain temperature, the chip will cross its operating temperature range, indicating a thermal runaway failure and malfunction.

1.3.1.5 Power Management

A thousand-core integrated chip with memory blocks consumes a large amount of power, resulting in an energy-inefficient data center on chip. The demands from cores are time varying, providing a uniform voltage-level to all cores results in high power density. As such a dynamic voltage and frequency scaling (DVFS) [62, 63, 64, 8, 65, 66] based power management has to be adopted. DVFS is efficient in terms of power management, but it may not be efficient at large scale due to involved off-chip power converters in traditional 2D integration. Additionally, it is complicated when delivering multiple voltage supplies to core and memory blocks in 2D integration.

1.3.1.6 I/O Management

In a many-core memory-logic integrated system, a large number of I/Os are required. Cores communicate with the external environment such as memory through local and global I/O interconnects. The achieved bandwidth depends on the number of I/Os utilized and the extent to which I/Os are utilized. To improve the effective bandwidth and utilize the available I/Os effectively, proper scheduling techniques with the aid of the memory controller needs to be carried out, which becomes complicated due to limited

Figure 1.10: Basic architecture for (a) 3D integration by vertical TSV interconnect; (b) 2.5D integration by TSI interconnect

space and routing concerns in 2D integration. Improper scheduling results in stalling of requests and affects the quality-of-service (QoS) [67]. Furthermore, inter processor/memory communication is often the bottleneck limiting the overall performance in a multi-core microprocessor [68]. Use of point-to-point or shared bus communication may not be scalable and efficient at large scales.

To overcome the above mentioned challenges, designers started exploring new integration techniques for many-core memory-logic integration. The possible integration techniques that is scalable towards thousand-core on-chip can be 3D integration by through-silicon via (TSV) interconnects and 2.5D integration by through-silicon interposer (TSI) interconnects. The 3D integration is carried out by vertical stacking of one layer over the other. Short distance through-silicon vias (TSVs) are utilized for the vertical interconnections. TSVs can help in signal or power delivery as well as heat-dissipation from the top layers to the heat-sink. Another possible many-core integration technique is 2.5D integration using the middle-distance TSIs. In 2.5D integration, the dies are placed on one common substrate and connected through TSIs. We introduce the 3D and 2.5D integration techniques respectively below.

1.3.2 3D Integration

Stacking of dies, termed as *three dimensional* (3D) integration can be seen in Figure 1.10(a), which shows the basic idea of 3D integration to vertically connect chip A and Chip B using the cylindrical TSVs.

Thanks to the recent advancement in 3D integration technology [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 5, 32, 33, 34, 35, 36, 37, 38], which can reduce the physical distance between the memory and the logic blocks, it has shown great potential to integrate thousands of cores with scaled performance superior to that of the 2D integration. 3D in-

tegration is identified as key and promising path, not only to facilitate the continuation of the conventional scaling, but also to enable the “more-than-Moore” heterogeneous integration of vast different functionalities into a system in a single chip form. Conventional wire-bond interconnections for 3D stacked packages could not satisfy the requirements for low power consumption and signal integrity because of its larger inductance and conductance loss. Hence, it is clear that shorter interconnection technologies are required. Flip-chip technology presents one of the possible solutions, but it is difficult to meet the high chip density requirement due to the large ball size and pitch. In 3D integration, traditional RC-interconnects cannot be utilized, because the scaling of global on-chip wires is not proportionate with the scaling of other features. As discussed previously, at smaller technology nodes, interconnect delays dominate the gate delays. Hence, vias are formed for inter layer communication in vertical direction. Thus, through-silicon via (TSV) emerged as the next potential technology which could provide shorter interconnection; this translates to a lower inductance and conductance loss. In general, the height of TSVs is few tens of μm and diameter of few μm . TSV can also assist in heat removal, which is a critical challenge faced in 3D integration.

As such, multiple device layers can be vertically connected by TSVs, resulting in not only a shorter length but also a higher integration density. Obviously, the system size can be dramatically reduced with diverse components compactly integrated. Potentially superior performance can be obtained from 3D stacking in terms of smaller footprint size, higher clock speed and heterogeneous integration, which in turn can reduce the chip cost. Also, numerous new design freedoms are given by the 3D stacking. As a result, the heterogeneous components fabricated by analog/RF, MEMS, or digital process can be integrated together into one system with a low fabrication cost and a high yield-rate. More importantly, as there is a significant boost in communication bandwidth, the 3D integration is well suited for I/O-centric system with concurrent data processing, ideally towards the thousand-core on-chip system integration. The use of TSV reduces the interconnect lengths, resulting in reduced latency, increase in bandwidth due to small RC values, and also consumes less power. It has been stated in [52], that for every doubling of stack height, performance per Watt increases by 20-30%.

It is inherent that due to the vertical stacking, heat gets accumulated in the top layers because of long heat-dissipation path to heat-sink. This accumulated heat can endanger the operational reliability of 3D IC. Another challenge imposed by TSV is its inherent parasitic capacitance which could add additional delay to signal transmission.

1.3.3 2.5D Integration

Though the benefits of 3D integration are too lucrative to ignore, thermal reliability concern in 3D integration [26, 69, 37, 39] required designers to look for an alternative for many-core memory-logic integration. Recent introduction of through-silicon interposer (TSI) opened doors for 2.5D integration. In 2.5D integration different dies i.e., multi-core microprocessor and memory blocks are integrated along the horizontal direction with the aid of through-silicon interposer (TSI) interconnects on one common substrate [10, 20, 39, 40, 41]. In contrast to 3D TSVs, which are usually designed as short RC-interconnect for inter-layer communication, TSIs are designed as a transmission line (T-line) targeted for high-speed, middle-distance communication. Compared to 2D RC-interconnects with repeaters, T-lines demonstrate better latency, speed and bandwidth but with large area overhead [70]. As TSIs are deployed underneath the common substrate, the area overhead is actually mitigated in the 2.5D integration.

Figure 1.10(b) shows a basic idea of 2.5D integration. Chips A and B are placed on one common substrate and TSI realized by T-line is deployed underneath the substrate to connect them. As all the dies are at uniform distance from the heat-sink, the thermal and mechanical reliability concerns are dramatically relaxed in 2.5D integration. Compared to 3D integration, the integration density of 2.5D integration is less with interconnect length of few *mm*. What is more, the TSI can be realized as a single-ended T-line (STL) or a differential ended T-line (DTL) [71, 72]. STL consumes less power compared to DTL, but DTL achieves better bit-error-rate (BER) compared to STL.

1.4 Contributions of This Work

In this thesis, firstly, we have developed physical models of 3D TSV and 2.5D TSI I/Os. With the stacking of layers, the heat starts to accumulate in the top layer resulting in a temperature gradient. Furthermore, due to difference in operating temperatures and coefficient-of-thermal-expansion (CTE) of TSV and silicon substrate, TSVs exert mechanical stress on the substrate, resulting in a stress gradient. A detailed study of the TSV structure reveals that the existence of liner material around TSV metal-fill, used to avoid ion diffusion, significantly affects the electrical property of TSVs. The liner material results in the formation of metal-oxide-semiconductor capacitance (MOSCAP), which varies nonlinearly with temperature. As such, with the consideration of liner material, a nonlinear electrical-thermal-mechanical coupled TSV delay model is proposed in this thesis that can provide accurate electrical delay. This nonlinear TSV delay model considering the electrical-thermal-mechanical coupling is developed by the respondent

author and is published in TCAD'13. In contrast to 3D integration, a thermal-resilient nature is observed in the 2.5D integration using TSI I/Os. TSI modeled as transmission line and its comparison with TSV appeared in D&T'15 and DATE'14 with the modeling as a major contribution from the respondent author. From the design perspective, to alleviate the TSV delay variation, a dummy TSV insertion is applied to balance thermal and stress gradients such that the resulting delay variation is minimized. As a case study, various 3D clock-tree benchmarks are experimented and the corresponding results are published in TCAD'13. It has been observed that with the insertion of dummy TSVs, clock-skew can be reduced by 61.3% on average with the proposed nonlinear electrical-thermal-mechanical delay model. Furthermore, to overcome the channel loss, design of LVDS and CML buffers with pre-emphasis stage is proposed. The post-layout simulation results in UMC 65nm CMOS process have shown an energy-efficiency of $0.48pJ/b$ and $0.075pJ/b$ for LVDS and CML buffers respectively.

To address the power I/O management in 2.5D/3D memory-logic integration towards thousand-core on-chip, a dynamic-voltage scaling (DVS) technique by employing on-chip single-inductor multiple-output (SIMO) power converters is proposed. A reconfigurable switch network is utilized to connect the power converters and cores based on the power-signature characteristics. Cores are initially clustered based on the magnitude of power-signature and the power converters are allocated accordingly. Furthermore, inside one cluster, based on the phase of power-signature, power converter is assigned in a time multiplexed manner. In addition, a demand-response based workload scheduling is performed to reduce the peak-power and achieve workload balance. The proposed power management is verified by system models with physical design parameters and benched power traces of workloads. For a 64-core, experiment results have shown 40.53% peak-power reduction and $2.5\times$ balance in workload along with 42.86% reduction for the required number of power converters. The above mentioned power management work for many-core microprocessor is mainly contributed by the respondent author and published in TC'15 and DAC'13

To address the signal I/O communication power and bandwidth, we studied memory-access data-patterns. We propose a design of memory controller to effectively manage the memory-logic communication by utilizing the available 2.5D TSI I/Os. To match the huge memory-access demands from cores, the demands are classified in space based on their magnitude and are connected to a port of the memory controller. At each port, cores are assigned with priority, based on which I/O channels are allocated in a time multiplexed manner. This space-time multiplexing scheme is deployed inside the memory controller. The proposed reconfigurable data-pattern aware reconfigurable memory controller with space-time multiplexed 2.5D TSI I/O is verified by a system-level simulator

with benchmarked workloads, shows up to 58.85% bandwidth balancing and 11.90% QoS improvement. This data-pattern aware memory controller's working principle and data pattern classifications is the major contribution from the respondent author and is published in D&T'15. To reduce the signal I/O communication power a self-adaptive output-voltage swing adjustment is introduced. Instead of a transmitting signal with large voltage swing, a Q-learning based I/O management is deployed to adaptively tune the I/O output-voltage swing under constraints of both power and BER. Furthermore, to overcome slower convergence of conventional Q-learning, an accelerated Q-learning is applied to adaptively tune the output-voltage swing, which is the major contribution of the respondent author and is published in TC'15. Experimental results show that the adaptive 2.5D I/Os designed in 65nm CMOS can achieve an average of 12.5mW I/O power, 4GHz bandwidth and 3.125pJ/bit energy-efficiency for one channel under 10^{-6} BER. With the use of conventional Q-learning and further accelerated Q-learning, we can further achieve 12.95% and 18.89% power reduction and 14% and 15.11% energy-efficiency improvement compared to the use of the uniform output-voltage swing based I/O communication.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents an overview of 3D and 2.5D integrations and literature review of the works addressing power and bandwidth management in 3D/2.5D many-core microprocessor. A detailed study of 3D TSVs and 2.5D TSIs and corresponding device modeling is presented in Chapter 3. A 3D clock-tree design based on the TSV modeling and buffer design for 2.5D TSI T-lines are presented in Chapter 4.

With the increase in number of cores, the number of I/Os increases as well, but still are limited, hence a need for I/O management rises, which is described in the last two chapters of this thesis. Power supply to the cores are supplied with the aid of power I/Os, a power management technique with the minimal number of power I/Os are discussed in Chapter 5. Due to limited number of signal I/Os, the bandwidth of the memory-access is limited. To improve the bandwidth utilization efficiency, a design of data-pattern aware memory controller is illustrated in Chapter 6. To address the communication power in many-core memory-logic integrated system, a Q-learning based adaptive voltage-swing tuning at the transmitter is proposed and presented in Chapter 7. Conclusions are drawn in Chapter 8 with suggestions for future work.

Chapter 2

Fundamentals and Literature Review

This chapter presents an overview of 3D and 2.5D integrations and few previous works on many-core integration addressing power and bandwidth management. First, we will present the overview of 3D integration, followed by 2.5D integration.

2.1 3D Integration

To achieve many-core memory-logic integration, designers explored stacking of dies in vertical direction. Stacking of one layer (with core or memory blocks) above the other with short- distance RC interconnects of through-silicon vias (TSVs), termed as 3D integration has become one promising alternative many-core integration technique. With the development in through-silicon via (TSV) fabrication, wafer alignment, thinning and bonding techniques, the 3D integration further gained importance in industry and academia. 3D integration can provide a very good integration density with low latency, high speed and low power. 3D integration supports heterogeneous integration. In this section, we first present the fabrication process of 3D IC followed by challenges.

2.1.1 Overview

For a better understanding of 3D ICs, we study the procedure of its fabrication. Fabrication of TSVs are different from traditional RC interconnects, due to its short interconnect distance and lower parasitics compared to traditional RC interconnects. As aforementioned, the delay caused by TSVs should be smaller than that from the logic gates. To model the TSV interconnect, a detailed study of TSV fabrication is needed. TSVs provide electrical interconnection between stacked dies as well as a heat-dissipation path. TSVs are generally considered cylindrical in shape with metal-fill in the middle surrounded by liner material to avoid ion diffusion. Fabrication of TSVs can be described

Figure 2.1: Simplified TSV fabrication procedure

Figure 2.2: Simplified flow to form (a) 3D IC; (b) 2.5D IC from traditional 2D ICs

in brief as follows. The filling material of TSV should possess good electrical conductivity as well as thermal conductivity. Tungsten (W), poly-silicon and copper (Cu) can be considered as TSV fill materials. Copper (Cu) is widely selected as TSV fill material due to its low cost and low resistance [73, 74]. To fabricate TSVs, deep reactive ion etching (DRIE), laser or chemical etching is performed first. After etching, liner material is deposited to prevent the metal ion diffusion from TSV to the substrate. Later, TSV material such as copper (Cu) is filled in the etched region at high temperature. After annealing to low temperature, the substrate is thinned and the current layer can be aligned and integrated with other layers. The whole process of TSV fabrication is shown in Figure 2.1. Due to the existence of liner and annealing process, the physical TSV becomes electrical, thermal and mechanical coupled device.

A simplified process of building 3D IC from traditional 2D dies is shown in Figure 2.2(a). Firstly, the dies that needs to be vertically stacked are thinned so that vias can be formed easily and the thickness of the integrated die is not so large. Once the thinning

Figure 2.3: (a) 3D integration of 64-cores and 64-memory blocks; (b) 64-core tier layout; (c) 256KB SRAM tier layout [5]

is performed, vias are formed and the connections to other layers are made. Finally, the whole integrated die is sent for packaging and other processes.

Few 3D integrated memory-logic microprocessors are designed in academia. From the academia, first 3D memory-logic integrated system is fabricated using 130nm GF process, called 3D massively parallel processor with stacked memory (3D MAPS) [5, 75] by Georgia Institute of Technology consists of 47,490 I/O TSVs and 6,540 dummy or thermal TSVs, used for heat-dissipation, shown in Figure 2.3. The die occupies an area of $25mm^2$. The supply voltage for 3D-MAPS is $1.5V$, with IR-drop inside a single core and memory tile is about $13mV$ and $10mV$, respectively. A total maximum IR drop of $78mV$ is observed. A maximum operating frequency of $277MHz$ is obtained. A maximum of $63.8GB/s$ was achieved.

2.1.2 Challenges

3D integration though is an efficient solution for many-core memory-logic integration, few of the key challenges needs to be addressed. It is inherent that due to the vertical stacking, heat gets accumulated in the top layers and the heat-dissipation becomes a serious concern due to long heat-dissipation path. This accumulated heat can endanger the operational reliability of 3D IC. Further, when a large number of cores are integrated on a single-chip, it becomes power hungry. It needs to be noted that the temperature of the chip depends on the power consumption as well.

TSV metal fill is surrounded by the liner material to prevent the migration of charge carriers from substrate to TSV and vice-versa. This existence of liner impacts the capacitance and the mechanical property of the TSV at higher temperature. Thermal management is one of the major concerns in 3D integration. Numerous works are carried out for thermal management by allocation of thermal TSVs, such as in [27, 76, 23, 77, 25, 28]. Thermal TSVs are proven to be effective in dissipation of heat because of its higher

Table 2.1: Off-chip [16] vs on-chip [17] power converter

Type	Physical area	Transition delay	Power efficiency
Off-chip	$13.7mm^2$	$1\mu s$	93%
On-chip	core: $0.14mm^2$ Decap+circuitry: $1.13mm^2$	$20ns$	87%

heat conductivity [23, 77, 25, 28, 5]. In addition to TSV based thermal management, microfluid based thermal management as well is carried out [78, 79, 80, 81]. Still there are few challenges to be addressed such as floor planning, routing, testing etc. As this thesis is more focused towards energy-efficient I/O design, we focus more on power management. However, as thermal management cannot be ignored in 3D integration we consider a 3D clock-tree design as a case study and show that, with the insertion of dummy or thermal TSVs (I/Os), heat-dissipation can be performed and critical parameters such as clock-skew can be reduced by proper insertion of dummy TSVs. We present various power management techniques presented in the past few years below.

A vast amount of research is carried out in 3D integration to address the power management issues. Power converter or voltage regulator is one of the key elements in the power management. Power management techniques are often hindered by the off-chip power converters, which adjusts voltage/frequency levels at a scale of few micro-second time-scale, which is inefficient. On-chip power converters [82, 7, 63, 4, 83, 64, 84, 49, 85] can provide fast switching time to different voltage levels compared to off-chip power converters. On-chip power converters can provide granularity of nano-second time scale for power management, compared to micro-second granularity by off-chip power converter. A simple comparison between on-chip and off-chip power converters presented in [64] is given in Table 2.1.

One can observe from Table 2.1 that transition delay for on-chip power converters are much smaller, which helps to perform power management at very small temporal granularity. Even though the area of on-chip power converters are large, 3D integration provides room to allocate power converters on-chip. Multi-core power management can be classified based on the prediction of power traces. Static and adaptive predictive techniques are two kinds of prediction based power management policies. Static techniques include fixed timeout [86], predictive shut down [87, 88] and predictive wakeup [89] techniques. For fixed timeout, a timer is set. If the system remains in idle state after time-out, the system is set to remain in off state, until a request is received. Unfortunately, errors in prediction and waiting for timeout causes wastage of power. To overcome the wastage of power, a predictive shutdown technique is proposed in [87, 88]. A non-

Figure 2.4: Power management schemes: (a) per-chip DVFS; (b) per-core DVFS; and (c) clustered DVFS

linear regression method is implemented for prediction based on the past history and issues a shutdown command. However, the type of regression to be adapted needs to be decided manually, and large amounts of data needs to be trained. To overcome the power wastage waiting for timeout, a predictive wakeup strategy is proposed in [89]. A wakeup command is issued after a certain amount of time, even if no request is received. This process will help in serving the request without further delay, but increases power consumption. Coming to the adaptive techniques, they are more efficient in power management compared to that of static methods, due to its dependence on workload characteristics. In [90], one single time out value is assumed and is increased when causing too many shutdowns and decreased when more timeouts can be tolerated. Dynamic-voltage-and-frequency-scaling (DVFS) is one of the widely used power management techniques in multi-core microprocessors [86, 62, 63, 64, 8]. The frequency/voltage of the processor core is adaptively adjusted based on the workload characteristics.

DVFS based power management can be classified based on the levels of granularity at which it is performed, namely per-chip DVFS, per-core DVFS and clustered DVFS. Figure 2.4 shows a basic overview of the above mentioned DVFS granularity levels.

- **Per-chip DVFS:** As shown in Figure 2.4(a), a single voltage/frequency level is used for the whole chip. Normally, the best voltage or frequency levels are chosen and that level is assigned to whole chip.
- **Per-core DVFS:** Each core is assigned with a power converter to perform the power management, as depicted in Figure 2.4(b). Each power converter executes a independent DVFS schedule. With the increase in number of cores, the number of required power converters also increase.
- **Clustered DVFS:** A group of cores (cluster) are driven by power converter(s). It needs to be noted that within a cluster, the DVFS schedule is same. However, the

Figure 2.5: System architecture for application aware DVFS [6]

DVFS schedule between two different clusters can be different and independent.

As we can see from the above definitions, that per-chip DVFS is not efficient due to the fact that all cores will be supplied with uniform voltage-level, resulting in an increased power density. However, we present power management techniques in literature at each granularity level below.

In [6], an application aware DVFS technique is proposed. The framework for the DVFS based power management is shown in Figure 2.5. Traditionally, when a new application provided as input to the system, the profiling table (consisting of process ID and DVFS level) are cleared and calculated again. This introduces a large overhead and latency. To avoid this, a profiling table is implemented at OS level, which stores the optimal level for each application and whenever similar kind of application is input, the level is fetched from the profiling table at OS level. In [6], energy-delay product (EDP) for each application is calculated and based on the EDP value, microprocessor determines the voltage and frequency for each core dynamically. To obtain the EDP value for an application, it is executed on program behavior analysers such as SimPoint [91]. Lower EDP value is better than higher value. Based on the EDP value, the corresponding DVFS level (from n levels) are chosen and assigned. Even though the computation cost of DVFS levels are slightly lower (for an re-executing application), this method is limited only for mobile environment with limited kind of applications.

In [7], effectiveness of DVFS power management on different temporal granularity scales are discussed. However, as per-core DVFS is employed, area overhead is large. In [64], a per-core kind of DVFS power management scheme is proposed for 3D integration. The power management scheme is applied to the cache as well to benefit from fine

Figure 2.6: 3D multi-core architecture with per-core power converters [7]

grained DVFS. The architecture utilized for power management in [64] is shown in Figure 2.6. In Figure 2.6, two layers are considered, where the logic layer consists of cores and power converters or voltage regulators and the top layer consists of memory blocks. The bottom layer consists of 8 cores with each core having a power converter and 8 power converters allocated in the memory layer for SRAM based L2 cache. Another 9 power converters are stacked for MRAM and MRAM based L2 caches. Before applying DVFS to the system, the power and performance characteristic profiles of workloads are obtained to determine the highest voltage and frequency levels. These levels are used as performance constraint for online DVFS mechanism. Based on the workload characteristics, the voltage and frequency levels are dynamically adjusted to achieve better energy efficiency. However, due to use of per-core DVFS mechanism, area overhead arises in this case. This kind of per-core DVFS techniques are not efficient when thousands of cores need to be integrated on a chip.

As the on-chip power converters occupy large area, per-core power management is unrealistic, hence a clustered power management technique is implemented. Few components of the on-chip power converters are scalable with technology, however the number of on-chip power converters that can be accommodated on a chip is limited. Due to smaller distance to the cores, the transmission loss and other effects are smaller compared to that of off-chip power converters and thus reliable. Under clustered power management scheme, a group of cores will be managed by power converter(s). As such, clustered DVFS is an intermediate point between extremes of per-chip and per-core DVFS schemes. In [63, 4, 8] voltage-frequency islands are utilized for power management of many-core microprocessors. In [8], the dynamic control is performed by monitoring the queue occupancy of the single-chip cloud platform. The adapted system architecture is shown in Figure 2.7. Each color represents a voltage/frequency island. The power management is performed as follows. Based on the queue occupancy, aver-

Figure 2.7: Tiles connected by mesh network with each color representing the voltage/frequency island [8]

age arrival rate and service rate are initially updated. Based on the previous states and target occupancy, the state feedback is computed and the best frequency divider is selected. A state-based feedback control to network (based on the occupancy of queue) is implemented to save energy at run time as well to develop robustness against workload variation. However, as each core is statistically assigned with one fixed island, such a voltage/frequency assignment cannot be optimal with response to the time-varying characteristics of workloads.

In [92], a clustered DVFS power management technique is presented. Clustering of cores is performed based on the temporal correlation between voltage levels. Initially, an off-line approach is used to determine the best possible per-core DVFS scheme. To improve the efficiency and avoid area overhead, clustering is carried out. Clustering is performed by K-Means algorithm. Finally, based on the formed clusters, an optimal schedule for cluster is obtained, similar to that of per-core DVFS finding. As clustered based power management requires less number of power converters than per-core DVFS, and can provide better control over power management than per-chip DVFS, we explore power management in this direction. In this thesis, we perform DVS by clustering. Further to reduce the number of power converters, we implement workload scheduling. More details about the proposed method is presented in Chapter 5.

2.2 2.5D Integration

Despite of the advantages such as high integration density, heterogeneous integration, short interconnects, low power and high bandwidth, 3D integration suffers from severe thermal and mechanical reliability concerns [26, 69, 37, 39]. In 2.5D integration,

multiple dies are placed on one common substrate and connected by through-silicon interposer (TSI) deployed underneath the substrate. TSIs are realized as high-speed and low power transmission lines with few *mm* in length. Compared to 3D integration, 2.5D integration is thermally resilient. Here, first we present the fabrication process of 2.5D IC followed by the challenges.

2.2.1 Overview

Figure 2.8: Simplified process of TSI Fabrication [9]

To have a better understanding of 2.5D integration, we study the fabrication process of TSI interconnect first. TSI Fabrication process is shown in Figure 2.8 [9]. Initially, a lithography is performed on the substrate, after which a deep reactive ion etching (DRIE) is done using a plasma etcher to form the via holes (Figure 2.8(a)). The formed via is then coated with the liner material (Figure 2.8(b)), followed by TSV-fill metal material deposition, copper (Cu) in our case. Even though the copper can form electrical interconnect between layers, it is removed because wiring structures have to be structured and Cu may not be compatible with many of demands. Moreover, most applications need polymer spacer layer between silicon and wiring layer [9]. Hence a layer of silicon oxide or liner material is deposited. A lid of thin electroplated copper (Cu) is created on

the top of each filled TSV (Figure 2.8(c)), because TSVs may be too small to be directly accessed by polymer vias. If multi-metal layers are needed, they are created using these Cu pads (Figure 2.8(d)). After the completion of processing on the front side, high density wiring on the back side needs to be formed, which may become tough in case of thin substrate layers. To overcome this difficulty, a glue material is applied on the top and a support wafer is attached using wafer to wafer bonding. Similar to the front-end, substrate is etched on the back side (Figure 2.8(e), (f)) and liner material is deposited using chemical vapor deposition (CVD) technique for having a polymer space (Figure 2.8(g)). A short cleaning of the revealed TSV on the backside is performed and a silicon dry etching is performed to create a stand-off between copper plug and surrounding silicon (Figure 2.8(h)). Finally, after having a small opening to TSV, metal layers are deposited for having electrical connection path (Figure 2.8(i)).

Figure 2.9: Composition of components in a typical TSI [10]

Figure 2.9 shows the components to integrate multiple ICs using the TSI interconnect. One can see the microbumps attached to the IC to connect to the TSI substrate. Further, with the aid of microbumps, the connection is provided to the metal layers. The TSV is connected to the bottom metal layer and further connected to the package with the help of ball grid arrays. In Figure 2.9, four metal layers are utilized. The number of metal layers vary based on the fabrication process and the requirements.

Figure 2.2(b) presents a simplified overview of the 2.5D integrated IC from tradition-

Figure 2.10: (a) Two ARM Cortex-A9 dual core processors connected by 65nm silicon interposer; (b) Cross section of 2.5D TSI [11]

al 2D IC dies. A substrate on which the 2.5D integration has to be performed is chosen first and the vias are drilled to form the T-line based TSI underneath the substrate. The dies are placed on the common substrate and the connection between the vias and the pads of the die are performed by the bonding.

A fabricated multiple processor chip using 2.5D integration from [11] is shown in Figure 2.10. The 2.5D chip integrated two 28nm SoC ARM dual core CortexTM-A9 processors. The interposer provides a 16GB/sec full duplex data rate between these two chips. The interposer was designed in 65nm technology. The design supports up to 4mm maximum signal length between the two dies and preserves minimum distance between two dies as 0.5mm. The TSVs are at 40 μ m pitch in the TSI design. I/O power is estimated as 0.5-0.6pJ/bit.

2.2.2 Challenges

In a many-core microprocessor, communication between memory and logic blocks plays an important role. The communication power and bandwidth are few of the key factors determining the efficiency of the communication. Compared to the 3D TSVs, 2.5D TSIs are longer in length and consume more power. To reduce the communication power, low-voltage swing signals can be utilized. Few of the works in 2D integration make use of low-voltage swing communication [93, 94, 95, 12]. In [12], a novel low-swing signalling circuit with self-resetting logic repeaters (SRLRs) are embedded within each router of a mesh NoC. SRLR achieves low-swing signalling due to the inherent channel attenuation. As SRLRs are available at each router, the data loss is reduced and is energy efficient. The design is made such that it is robust to global variations. Figure 2.11 shows a 10mm SRLR based link for the mesh NoC, with each router at 1mm distance and pulse

Figure 2.11: Low swing data link with SRLR embedded in each router [12]

Figure 2.12: Target 3D hardware architecture [13]

modulator (PM) and demodulator (DM) at both ends. As the proposed method uses single-ended signalling, it is more prone to channel noise. In addition, as each router has SRLR embedded, it causes area overhead, though the design is optimized. In most of the works cited above, uniform voltage swing is adapted. However, as the BER need not to be low all the time for workloads, adaptive voltage swing based communication can be utilized for communication power reduction. In this thesis we use machine learning algorithm like reinforcement Q-learning to adaptively tune the voltage swing under both BER and power budget constraints.

In addition to communication power, communication bandwidth also plays a key role in determining the performance. A memory controller is commonly used as the bridge

Figure 2.13: Memory-logic integrated architecture with 3D memory controller [13]

between memory and core blocks. Many of the works on memory controllers address 3D integration, such as [30, 14, 13, 96]. However, compared to 3D TSVs, utilisation of bandwidth is less in 2.5D TSIs. In this thesis, we present an I/O management scheme to effectively utilize the bandwidth of 2.5D TSI I/Os. We present few works on memory controller here. Memory controllers handle one of the most critical and important problems in multi-core design i.e., manage bandwidth requirements of a high performance communication infrastructure. State-of-the-art memory controllers [97] are designed for narrow off-chip interfaces, which are no more needed in 3D/2.5D integration. In addition, they are quite complex with many complex features to maximize the exploitable interface bandwidth. The major difference between state-of-the-art memory controllers and 3D controllers is mainly in the wide I/O data bus with less complex and energy efficient memory interface [14]. An early approach for 3D DRAM stacking on memory layer with memory controller is presented in [30]. The memory interface is modified such that use of narrow interfaces (pin-out) and address multiplexing is removed. However, due to the use of a shared bus, the scalability of the system is limited. In [13] a specialized 3D DRAM controller with faster path to local memory neighborhood and a standard NoC communication facility for accesses to remote memory is developed. The targeted system architecture in [13] is shown in Figure 2.12. The memory access scheme [13] in 2D manner is shown in Figure 2.13. In Figure 2.13, the local accesses are routed directly to the memory, whereas remote access are routed through the NoC. This takes the advantage of low latency and high bandwidth. However, use of one memory controller per core may cause an area overhead.

In [14], a 3D DRAM channel controller is presented, shown in Figure 2.14. In the first stage, data width adaptation between the front-end buses (processor side) and the synchronization queues, and to cache the portion of the row that has been accessed in the previous command are provided. As such, when a “hit” command is issued, the

Figure 2.14: 3D DRAM sub system architecture with single vertical channel [14]

data from the cache is forwarded. In case of a “miss”, the command is transferred to the buffering stage for further elaboration. The synchronization stage is composed of dual clock FIFOs to provide buffering and an asynchronous communication approach between the DRAM and the frontend clock domain. Lastly, the frontend command translation and data preparation, sending to the DRAM the right command sequences are performed. As previously mentioned, the memory controllers in 3D helps in utilizing the bandwidth effectively compared to that of the 2.5D TSIs, because of its smaller length and less power consumption. In this thesis, we explore the memory controller design for 2.5D design. One needs to note that our design explores the interface between processor and memory, similar to [98]. The internal structure of memory is left untouched. More details about proposed 2.5D memory controller is presented in Chapter 6.

2.3 Summary

In this chapter, we introduced the simplified fabrication methods for 3D TSV and 2.5D TSI I/Os in a brief manner. Further, we discussed the formation of 3D and 2.5D integrated systems from the traditional 2D ICs. Power management and bandwidth management are few of the concerns in a many-core microprocessor. Power management techniques for many-core microprocessors stemming from a granularity of per-core DVFS to per-chip DVFS with multiple works available in literature are discussed. Further, low-voltage signalling techniques in the literature is discussed to reduce signal I/O communication power. Lastly, to effectively utilize the I/O bandwidth, memory controller designs available in the literature are presented.

Chapter 3

Device Model

3.1 Introduction

Many-core integration is not possible by the traditional 2D integration due to the severe loss caused by long PCB wire trace and poor energy-efficiency with long latency [53, 99, 100]. 3D integrated circuits (3D ICs) emerged as one of the promising solutions for many-core memory-logic integration utilizing the through-silicon via (TSV) interconnects. Utilization of TSVs as interconnects in 3D ICs significantly reduces the wire length, latency and power dissipation in global interconnects such as memory buses and also clock-tress [101, 24, 29, 30, 26, 31, 102, 5, 73, 28, 103]. However, one robust 3D IC design requires a careful examination of TSV I/O interconnects in which electrical states such as delays are coupled from multiple physical domains. Firstly, thermal reliability is one of the critical concerns, since the heat-sink is far from the top layer [104, 28]. The metal-fill of TSV is surrounded by a liner (Silicon dioxide or low dielectric) to avoid ion diffusion as well as to provide isolation. As such, there exists a large temperature gradient between substrate and TSV, resulting in an electrical-thermal coupling that can result in delay variation of TSV. A nonlinear MOS-capacitance (MOSCAP) is formed between TSV and substrate due to existence of liner material. Owing to the differences in the coefficient of thermal expansion (CTE) of TSV material and substrate material, a large mechanical stress is induced onto the substrate by TSV at high operating temperatures, leading to the variation in driver delays by the electrical-mechanical coupling. Hence, a physics-based electrical-thermal-mechanical coupled model for TSV needs to be developed.

To overcome the issues in 3D integration, an alternate approach for many-core memory-logic integration is the recently introduced through-silicon interposer (TSI) [10] I/O interconnect based 2.5D integration. In contrast to a TSV which is a short RC-interconnect

for inter-layer communication, a TSI is usually designed as a transmission line (T-line), targeted for high-speed mid-distance communication between main memory and cores on one common substrate. Compared to the RC-interconnect with repeaters, 2D single-ended T-line (STL) or differential T-line (DTL) [71, 72] with current-mode-logic (CML) buffers [20] have demonstrated better latency, power and bandwidth performance but with large area overhead. However, in 2.5D integration, the TSI based T-line can be designed through and under the common substrate, thus mitigating the area overhead. Ideally TSIs can be deployed for memory-logic integration with high performance yet low area overhead. Similar to TSVs, the liner material is in contact with the metal-fill in TSIs, hence the TSI capacitance is modeled as a nonlinear MOSCAP as well. As memory and logic components are spread on the common substrate that is close to a heat-sink, the thermal reliability concern of the integrated high-performance server is relaxed in 2.5D integration. In this chapter, based on recent measurement results in [102, 73, 74, 105, 106], a nonlinear electrical-thermal-mechanical coupled 3D TSV and 2.5D TSI delay and power models are developed. However, design of TSI I/O links is one of the challenges to be addressed. I/Os contribute a significant amount to the overall power consumption [68]. Hence, a low power I/O design has to be implemented.

In this chapter, first we will present the nonlinear MOSCAP model, followed by delay and power models of 3D TSVs and a clock-tree design. Lastly, delay and power models of 2.5D TSI interconnects with buffer designs are discussed.

3.2 Nonlinear MOSCAP Model

Existing Models and Shortcomings With the development of 3D integration, a large amount of research work is carried out on modeling of 3D TSV interconnects. Among them, we discuss a few most important works carried out on modeling of TSV interconnects. In [107, 108, 109] TSVs are modeled as short RC interconnects. The capacitance is considered to be constant or non-varying with temperature, whereas resistance of TSV is considered as linearly linearly dependent on the temperature T and capacitance as a constant, as given in (3.1).

$$\begin{aligned} R_T &= R_0(1 + \alpha \cdot \delta T); \quad \delta T = T - T_0; \\ D_{RC} &= R_0 C_0(1 + \alpha \cdot \delta T) \end{aligned} \tag{3.1}$$

where R_T is the temperature-dependent resistance; R_0 represents the resistance at room temperature; C_0 represents the capacitance at room temperature; δT is the difference between the operating temperature T and room temperature T_0 ; and α is the temperature-

Figure 3.1: (a) Signal TSV and dummy TSV in 3D-IC; (b) 3D view of a circular TSV; (c) equivalent simplified RC-circuit of a TSV

dependent coefficient for resistor. Thus, the traditional RC interconnect can be modeled as linearly dependent on the temperature.

However, in the previously proposed model, the constant or non-varying constant is not true, because the liner material surrounding the TSV-metal fill Figure 3.1(a) is not considered. It needs to be noted that capacitance of TSVs plays major role compared to the inductance due to its shorter length. Additionally, the impedance matching is not required for TSVs because the loss in the short interconnect is negligible.

Proposed Model TSVs that are deployed for inter-layer signal communication are called signal TSVs. A signal TSV and its cross sectional view is shown in Figure 3.1(a) and 3.1(b) respectively. The equivalent and simplified circuit representation of signal TSV is depicted in Figure 3.1(c). When voltage is applied across TSV, existence of liner material around TSV forms a MOSCAP, and is nonlinearly dependent on the temperature due to different CTEs of TSV and substrate [102, 103]. This nonlinear capacitance depends on biasing voltage (V_{BIAS}) as well as temperature [73, 36, 37]. The difference in work-function between metal-material of the TSV and the substrate results in the existence of a depletion region. The radius of the depletion region varies with the biasing voltage and temperature, resulting in a nonlinear capacitance. Based on the CV curves presented in Figure 3.2(a), one can observe that it is divided into accumulation, depletion and inversion regions separated by flat-band voltage (V_{FB}) and threshold-voltage (V_T). At higher frequencies ($>1\text{MHz}$), the inversion region can be further divided into deep-depletion and inversion regions. One needs to note that TSI also consists of TSVs.

Based on this MOSCAP model and the measurement results presented in [102, 73,

Figure 3.2: (a) Typical CV curve of nonlinear MOSCAP model for signal TSV; (b) exertion of stress on from TSVs to transistors/drivers in a unit square-grid

105, 106], the RC-delay parameters of temperature dependent TSVs can be given as

$$R_T = \frac{\rho h}{\pi r_{metal}^2}; \quad \frac{1}{C_T} = \frac{1}{C_{ox}} + \frac{1}{C_{dep}}; \quad (3.2)$$

$$C_{ox} = \frac{2\pi\epsilon_{ox}h}{\ln(\frac{r_{ox}}{r_{metal}})}; \quad C_{dep} = \frac{2\pi\epsilon_{si}h}{\ln(\frac{r_{dep}}{r_{ox}})}$$

where R_T , C_T represents the temperature-dependent resistance and capacitance of TSV respectively; C_{ox} and C_{dep} are the liner and depletion-region capacitances of TSV respectively; TSV height is represented by h ; ρ is the resistivity of metal-material of TSV; ϵ_{ox} and ϵ_{si} are the dielectric constants of silicon dioxide and silicon respectively; and r_{metal} , r_{ox} and r_{dep} are the outer radii of TSV metal, silicon and depletion-region respectively, as shown in Figure 3.1(b). The radius of the TSV metal, oxide and depletion region varies with the temperature.

As the thermal conductivity of liner material (SiO_2) is nearly one hundred times lower than that of the silicon substrate, the liner prevents the dissipation of heat from the devices on the substrate. This accumulated heat results in hot-spots at and around signal TSVs. As shown by measurement results in [73], the TSV capacitance approaches liner capacitance at high temperature due to the existence of hot-spots and nonlinear temperature dependence.

3.3 TSV Device Model

As discussed previously, the temperature impacts the electrical delay of the TSV. In addition, the temperature also affects the mechanical reliability of the system. The exerted mechanical stress from the TSV has an impact on the mobility of charge carriers at deep sub-micron levels and the delay of the drivers get affected. As TSVs are connected to the drivers, the combined impact is preferred to that of only TSV characteristics. In this section, we discuss thermal coupling to electrical and mechanical properties, followed by coupled delay and power models.

3.3.1 Electrical Model

We derive the electrical model for the TSV by considering the MOSCAP formed between Cu fill and the silicon substrate of TSV. Even though at higher frequencies, the CV curve of signal TSVs tend to be flat, with the change in V_{BIAS} , the TSV capacitance C_T in the deep-depletion region tends to vary nonlinearly with temperature due to r_{dep} . The resistance of the TSV, R_T , can be modeled as linearly dependent with the temperature, as given in (3.3). The nonlinear temperature-dependent TSV capacitance C_T based on the measurement results from fabricated TSVs in [73] can be given as

$$R_T = R_0(1 + \alpha\delta T); \quad C_T = C_0 + \beta_1 T + \beta_2 T^2 \quad (3.3)$$

where C_T , C_0 are the TSV capacitances at temperature T and room temperature respectively; R_0 is the TSV resistance at room temperature T_0 ; α is the temperature dependent coefficient for resistance; β_1 and β_2 are the first and second order temperature dependent coefficients of C_T respectively; and δT represents the difference between T and T_0 . The values of β_1 and β_2 are determined experimentally and reported in [73].

The nonlinear variation of the capacitance is different from the traditional via characterization, modeled as linearly variant with the temperature. As such, the nonlinear electrical-thermal coupling of signal TSVs is strong and brings significant impact of the temperature on the timing in 3D integration.

Accuracy Analysis Based on the above developed models, we present a comparison of the nonlinear TSV model and the linearly modeled RC-interconnect delay model below. Signal TSV is modeled by considering the nonlinear temperature dependent capacitance as described in (3.3) and RC interconnect modeled as in (3.1). A few measurement results for electrical parameters for a TSV is obtained from the collaboration from Prof. Tan's group of NTU. The values of capacitance with different dielectric materials are

Table 3.1: TSV Electrical Parameters

	PETEOS (/cm)	Low-k (/cm)
Liner capacitance (C_{ox})	22.4pF 22.1pF (CV)	17.5pF 15.9pF (CV)
Si Depletion radius (r_2)	$r_2 - r_1 =$ 265nm	$r_2 - r_1 =$ 265nm
Si Depletion capacitance (C_{dep})	62.4pF	62.4pF
C_{min}	17.5pF 15.6pF (CV)	13.7pF 12.1pF (CV)

reported in Table 3.1.

It can be observed from Table 3.1, the TSV delay is nearly 16ps, whereas the delay reported from the model in Figure 3.3(b) is nearly 18ps, thus the accuracy of the proposed method is nearly 90%, which is good enough for performing the calculations. However, to have a better and accurate analysis, we consider the previous work from [73] and calibrate the scaling parameters required to estimate the TSV delay.

Signal TSV with the following parameters are chosen for the purpose of comparison: height $40\mu m$, diameter of $15\mu m$ with a corresponding resistance of $44m\Omega$ at room temperature. Based on the measured results reported in [73], the values for coefficients of temperature dependent parameters α , C_0 , β_1 and β_2 in (3.1) and (3.3) are used as: $0.00125/K$, $88.8fF$, $0.0667fF/K$ and $0.0014fF/K^2$, respectively. In addition, for the reliability consideration, a bundle of TSVs are used for signal distribution instead of one single TSV. TSV bundle is formed by grouping a few number of TSVs, named as T2, T4, T8 and T10 to represent 2, 4, 8 and 10 TSVs in each bundle, respectively.

Validation The nonlinear variation of the signal TSV MOSCAP based on (3.3) is presented in Figure 3.3(a). It needs to be noted from Figure 3.3(a), at high temperatures the TSV capacitance varies nonlinearly due to the existence of liner material and difference in CTE. The same is explained mathematically in (3.3). For example, one signal TSV capacitance at room temperature $25^\circ C$ is $87fF$, at $75^\circ C$ is nearly $93fF$, and at $150^\circ C$ is $113fF$, which shows a nonlinear growth. Simulations with TSV process variations are carried out by varying its capacitance value by maximum of 10%. The according variation in delay of signal TSVs is however, less than 3%, and hence is negligible when compared to thermal or mechanical impact.

To study the delay characteristics, an inverter buffer is connected at both the ends of TSV RC interconnect. The length of input and output 2D wires to signal TSV are assumed to be as small as possible. The nonlinear effects of temperature on RC-delay at different temperatures for different TSV bundles T2, T4, T8 and T10 are shown in

Figure 3.3: (a) Variation of TSV capacitance with temperature; (b) variation of TSV delay with temperature for different TSV bundles based on (3.3)

Figure 3.3(b). Though the nonlinear temperature-dependent MOSCAP contributes to a significant amount of delay, the use of signal TSV bundles can help in reduction of the temperature, thereby reducing the overall skew.

It needs to be observed from Figure 3.3(b), that the delay for a T8-bundle at 120°C reaches nearly 100ps , which is 67% of the half-clock cycle for a 3.3GHz multi-processor. For a normal temperature of 75°C , the delay for a TSV T8-bundle is nearly 60ps ; and at the maximum temperature of 200°C , the delay for a T8-bundle reaches nearly 140ps , which is nearly 46% of a clock-cycle of 3.3GHz multi-processor. This delay is of serious concern if no proper cooling technique is applied.

Additionally, it needs to be noted that if the TSV is modeled as the traditional linear coupled model, then the estimated delay will be inaccurate. For example, for a T10-bundle, at a temperature of nearly 125°C , the delay with the nonlinear coupled model is nearly 130ps whereas with the traditional linear coupled model, the delay is 120ps which is inaccurate. This difference can effect the design considerations as well as timing constraints during design of 3D IC.

3.3.2 Mechanical Model

In addition to the electrical coupling, we develop a mechanical model of the TSV. Differences in the operating temperature and mismatch of CTE between TSV and the substrate results in exertion of mechanical stress. Exerted mechanical stress by multiple TSVs on the substrate found by principle of superposition [74] is given by

$$\sigma_i = -\frac{B\Delta T\Delta\alpha}{2}\left(\frac{R_i}{r_i}\right)^2$$

$$\sigma = -\sum_{i=1}^n \sigma_i = -\frac{Bn\Delta\alpha\Delta T}{2}\left(\frac{R}{r}\right)^2; \quad n = \eta A \quad (3.4)$$

where σ_i is the stress from i -th TSV; B is the biaxial modulus; $\Delta\alpha$ is the CTE difference between TSV material and substrate, dependent on TSV and substrate materials; ΔT is the temperature difference; R_i is the radius of the i -th TSV; r_i represents the distance of a transistor or a driver from center of i -th TSV; and n represents the number of TSVs with a TSV density of η in area A .

For the simplicity of illustration, all the TSVs are considered to be of same radius R , and drivers are approximated to be at same distance r from the center of TSVs. Based on these assumptions, we observe a thermal-mechanical dependence to characterize the mechanical stress. In thermal-mechanical coupling, the main focus is on exertion of mechanical stress with respect to temperature and TSV density.

Accuracy Analysis and Validation A quick revisit of the finite element analysis (FEA) for the stress exerted by Cu TSV on the substrate is presented in Figure 3.4. It can be observed that the stress decreases with the increase in distance. For example, the FEA results presented in Figure 3.4, which fits the measurement results (as per our personal discussion with Prof. Tan of NTU), has a stress of early 43MPa for 10 μm TSV at 2 μm distance, whereas with the presented model in (3.4), the exerted stress is nearly 40MPa, which is slightly lower, but can be utilized for the purpose of carrying out simulations. Thus, the result of FEA analysis fits the stress model presented in (3.4).

Figure 3.4: FEA simulation curves of the thermo-mechanical stress exerted by a single Cu-TSV on Si substrate as a function of distance from the edge of TSV

Figure 3.5: (a) Variation of TSV stress with distance; (b) variation of TSV stress with temperature and TSV density based on (3.4)

Further, to validate the thermal-mechanical impact, we follow a similar setup as in Figure 3.2(b). Mechanical stress from TSV is caused by difference of CTEs between TSV and substrate. Different layers of 3D-IC have different temperatures and hence TSVs and the substrate will be at different temperature, resulting in a stress gradient. For the purpose of simulation, all TSVs are considered to have a diameter of $15\mu\text{m}$ and a density of $400/\text{mm}^2$. The exerted mechanical stress from TSV on device is given by equation (3.4).

By considering all the TSVs, exerted mechanical stress on a driver placed inside a square surrounded by TSVs is shown in Figure 3.5(a). It needs to be observed that, stress and mobility remains nearly uniform out of a particular distance, which defines the keep-out-zone (KoZ); but for the area inside KoZ, there is a significant variation in stress and mobility observed. In our experiment, a keep-out-zone of $3\mu\text{m}$ is considered.

The coupled impact of TSV density and temperature gradient on stress is shown in Figure 3.5(b). When the temperature gradient increases and at high TSV density, the amount of exerted stress on the substrate will be high. Hence, in order to reduce the TSV stress on substrate, temperature gradient has to be reduced as well. What is more, the amount of stress can be varied when the TSV density is different. Let us consider a temperature gradient of 150°C , the stress at a TSV density of $200/\text{mm}^2$ is 28.93MPa , and the stress at a TSV density of $400/\text{mm}^2$ is 57.87MPa , indicating that stress depends on the temperature gradient as well as TSV density. The mechanical stress from TSVs also affects the carrier mobilities of drivers on the substrate. Based on the alignment of the transistors and the amount of exerted stress, the carrier mobility varies [74, 105, 106]. This variation in charge carrier mobility adversely affects the delay of drivers. Variation in carrier mobility due to exerted stress [74, 105, 106] is given by

$$\frac{\delta\mu}{\mu} = -\Pi \times \sigma; \quad m = -\Pi_x \quad (3.5)$$

where $\delta\mu/\mu$ is the ratio of mobility variation; Π is the tensor of piezo-resistive coefficients; σ represents the mechanical stress from TSV; and m represents the maximum value of Π among all the directions (x, y, z), in order to capture its most significant impact on the transistor.

For example, Π_x is used to represent the value of the maximum value in tensor Π . m indicates the enhancement factor along the direction that results in maximum stress. The direction of maximum stress is different for NMOS and PMOS devices, resulting in different amount of mobility variations [74, 105, 106] for the same amount of stress, due to the inherent property of materials. The hole mobility increases in vertical direction, whereas electron mobility increases in horizontal direction. The ratio of mobilities with and without stress can be given as

$$\frac{\mu_s}{\mu} = 1 + \frac{\delta\mu}{\mu} = 1 + m\sigma \quad (3.6)$$

where μ_s and μ represents the mobility of charge carriers with and without impact of stress respectively; and $\delta\mu/\mu$ is the mobility variation ratio.

With the increase in the amount of exerted stress, the ratio of mobilities with and without stress μ_s/μ also increases. This variation in mobility results in a change of source resistance of the driver, given by

$$R_D^M = \frac{R_D}{1 + \frac{\delta\mu}{\mu}} = \frac{R_D}{1 + m\sigma} \quad (3.7)$$

where R_D^M is the driver resistance with impact of stress; and R_D is the driver resistance without impact of stress.

The simulation results for electrical-mechanical impacts on the driver is presented. As the amount of exerted mechanical stress varies, the deformation in lattice structure also varies, resulting in variation of carrier mobility.

For the purpose of illustration, a single TSV having a diameter of $15\mu m$ is considered as the source of stress, the variation in mobility and delay due to the exerted stress with different distance is discussed here. Considering a keep-out-zone of $3\mu m$, variation of mobility and delay with a distance for a $22nm$ metal gate PMOS and a NMOS placed on substrate is shown in Figure 3.6(a). The delay of one according driver is shown in Figure 3.6(b). Note that the simulations are carried out in a SPICE simulator [110].

Moreover, one can have the following observations from Figure 3.6(a). The amount

Figure 3.6: (a) Variation of NMOS/PMOS carrier mobility with distance under TSV stress based on (3.6); (b) variation of driver delay with distance under TSV stress

Figure 3.7: (a) TSV with buffer at its ends; (b) delay model of 3D clock-tree with nonlinear electrical-thermal-mechanical coupled model

of stress exerted varies with the distance, which is in agreement with (3.4). In addition, with the same amount of stress, the variation in hole mobility is higher than electron mobility, due to the material properties. For example, for a distance of $2\mu m$, there is a variation of nearly 4% and 1.8% in mobilities of holes and electrons, respectively.

What is more, variation of the driver delay in the presence of mechanical stress is shown in Figure 3.6(b). It can be seen that as the distance increases, variation in the delay decreases due to reduction in the exerted stress. There is a decrease of just 1.6% delay at a distance of $4\mu m$ whereas nearly 3.7% at distance of $1\mu m$. So, if the keep-out-zone is increased, there will be less stress with small variation of mobility and delay.

3.3.3 Delay Model

Based on the previously presented and validated electrical, thermal and mechanical impacts at device level in the previous section, a coupled model for electrical delay of a TSV I/O with inverter buffers at its both ends, as shown in Figure 3.7(a) is developed.

3.3.3.1 Electrical-Thermal Coupled Delay Model

As aforementioned, temperature has a significant impact on the signal TSV forming a nonlinear MOSCAP as given in (3.2). By considering nonlinear electrical-thermal coupling of signal TSV in (3.3), the signal delay D_{TSV1} for clock-tree shown in Figure 3.7(b) can be calculated as

$$D_{TSV1} = R_{in}\alpha\beta_2T^3 + R_{in}[(1 - \alpha T_0)\beta_2 + \alpha\beta_1]T^2 + [\alpha(D_0 + R_{in}C_0) + (1 - \alpha T_0)R_{in}\beta_1]T + (1 - \alpha T_0)(R_{in}C_0 + D_0); \quad (3.8)$$

where

$$R_{in} = \frac{R_D}{S_D} + S_{w1}R_{w1} + \frac{R_T}{2S_T};$$

$$D_0 = \frac{1}{2}(S_{w1}^2R_{w1}C_{w1} + S_{w2}R_{w2}S_L C_L) + \left(\frac{R_T}{S_T} + S_{w1}R_{w1} + \frac{S_{w2}R_{w2}}{2}\right) (S_{w2}C_{w2} + S_L C_L) + \frac{R_D}{S_D}(S_{w1}C_{w1} + S_{w2}C_{w2}S_L C_L + S_D C_P) \quad (3.9)$$

where R_{in} is the total resistance counted from the TSV capacitor; C_L is the load capacitance; α is the temperature dependent coefficient of TSV resistance R_T ; β_1, β_2 are the first order and second order temperature dependent coefficients of TSV capacitance C_T respectively; T represents the temperature; D_0 is the delay of the circuit shown in Figure 3.7(b) without TSV; and R_{w1}, R_{w2} are the unit wire resistances and their according sizing parameters are S_{w1} and S_{w2} .

For TSV I/O delay calculation with electrical-thermal coupling in (3.8), the delay contribution from horizontal metal wires and buffers are also considered. The nonlinearity in delay is mainly introduced due to nonlinear MOSCAP of TSV, and horizontal metal wire is modeled as linear temperature dependent resistor.

3.3.3.2 Electrical-Mechanical Coupled Delay Model

The impact of mechanical stress on the transistor or driver is studied here. The mechanical stress from TSVs has non-negligible impact on the driver resistance as given in (3.7). Hence the impact of mechanical stress on the delay of driver connected to TSV has to be considered during modeling for better accuracy. Delay D_{TSV2} with electrical-mechanical coupling is calculated as

$$D_{TSV2} = \frac{D_\sigma}{1 + m\sigma} + D_w; \quad (3.10)$$

where

$$D_\sigma = \frac{R_D}{S_D} [C_P S_D + S_{w1} C_{w1} + S_T C_0 + S_{w2} C_{w2} + S_L C_L];$$

$$D_w = S_{w1} R_{w1} \left[\frac{S_{w1} C_{w1}}{2} + S_T C_0 + S_{w2} C_{w2} + S_L C_L \right] + S_{w2} C_{w2} \left[\frac{S_{w2} C_{w2}}{2} + \frac{R_0}{S_T} \right] + \frac{R_0 C_0}{2}. \quad (3.11)$$

Here D_σ and D_w are the stress dependent and independent TSV delays respectively; R_D is the driver resistance without impact of stress σ ; R_0 , C_0 are the temperature independent TSV resistance and capacitance respectively; m is the mobility enhancement factor; C_L is the load capacitance; α is the temperature dependent coefficient of TSV resistance R_T ; β_1, β_2 are the first order and second order temperature dependent coefficients of TSV capacitance C_T ; T represents the temperature; and S_{w1} , S_{w2} are the scaling parameters of unit wire resistances.

From (3.10), one can observe that D_σ is composed of stress dependent and independent components. Mechanical stress mainly affects the driver. A large amount of stress can deform the substrate, resulting in broken links as well.

3.3.3.3 Electrical-Thermal-Mechanical Coupled Delay Model

In the previous sections, the delay model of TSV with coupling of electrical or mechanical properties to temperature are discussed. In this section, a coupled impact to the delay of TSV is presented. The electrical delay D_{TSV} considering electrical-thermal-mechanical coupling is given by

$$D_{TSV} = D_c + D(T) + D(\sigma) + D(T, \sigma); \quad (3.12)$$

where

$$D_c = D_0 - D_\sigma + R_0 C_0 (1 + \alpha T_0) - \frac{R_0 (S_{w2} C_{w2} + S_L C_L) \alpha T_0}{S_T};$$

$$D(\sigma) = \frac{R_D}{S_D (1 + m\sigma)} [S_D C_P + S_{w1} C_{w1} + S_T C_0 + S_L C_L + S_{w2} C_{w2}]; \quad (3.13)$$

$$D(T) = R_0 \alpha \beta_2 T^3 + R_0 [(1 - \alpha T_0) \beta_2 + \alpha \beta_1] T^2 + [R_0 \beta_1 (1 - \alpha T_0) + \alpha R_0 C_0 + S_{w1} R_{w1} S_T \beta_1 + \frac{R_0 \alpha (S_{w2} C_{w2} + S_L C_L)}{S_T}] T;$$

Figure 3.8: Signal delay of one single TSV under different technologies based on (3.12)

$$D(T, \sigma) = \frac{R_D}{S_D(1 + m\sigma)} [S_T \beta_1 T + S_T \beta_2 T^2].$$

Here D_c is a constant delay composed of D_0 and D_σ given in (3.8) and (3.10) respectively; $D(T)$ is the delay as function of temperature only; $D(\sigma)$ represents the delay as function of stress only; $D(T, \sigma)$ is the delay as function of both temperature and stress; C_L is the load capacitance; α is the temperature dependent coefficient of TSV resistance R_T ; β_1, β_2 are the first order and second order temperature dependent coefficients of TSV capacitance C_T ; T represents the temperature; D_0 is the delay of the circuit shown in Figure 3.7(b) without TSV; D_σ and D_w are stress dependent and independent delays respectively; R_D is the driver resistance without impact of stress σ ; R_0, C_0 are the temperature independent TSV resistance and capacitance respectively; m is mobility enhancement factor; and S_{w1}, S_{w2} are the scaling parameters of unit wire resistances.

As a summary, delay in the 3D case is quite in contrast with the 2D case that has only a linear dependence on temperature. The nonlinearity dependence on temperature in 3D case arises from TSV MOSCAP.

Considering a 3D TSV clock-tree similar to in Figure 3.7(b), the variation of delay on a single signal TSV with drivers in different technologies are presented in Figure 3.8.

Figure 3.8 shows the delay values for a single TSV having buffers at both ends. A 22nm CMOS inverter is used as buffer with the following settings: sizing parameters and driver settings are $\frac{R_D}{S_D} = 100\Omega$ and $S_D C_P = S_L C_L = 2fF$. It needs to be noted that one signal TSV adds up the delay by nearly $4\times$, indicating that the TSV delay is comparable to the minimum sized driver delay. As signal TSV is modeled as a nonlinear MOSCAP with nonlinear temperature-dependence, the delay increases with temperature signifi-

cantly when electrical-thermal coupling is considered. It can be observed that the delay can increase by nearly 15% for all technologies when temperature is increased at 200°C . Then, with the electrical-mechanical coupling considered, the delay with stress gradient at a reference temperature of 75°C is calculated, which is found to be of 9% lower than the delay without insertion of TSV as stress enhances mobility. By considering all these effects, there is approximately 10% delay variation introduced at 200°C .

3.3.4 Power Model

In addition to delay modeling and clock-tree design, we explore the dependence of TSV power on other physical parameters. The dynamic power of an N -channel 3D TSV with supply voltage of V_{DD} can be given as

$$\begin{aligned} P_{TSV} &= N \cdot C_T \cdot V_{DD}^2 \cdot f \\ &= N \cdot (C_0 + \beta_1 T + \beta_2 T^2) \cdot V_{DD}^2 \cdot f \end{aligned} \quad (3.14)$$

where P_{TSV} is the 3D TSV interconnect power; f is the clock frequency; and C_T is the temperature dependent 3D TSV capacitance, given in (3.3).

From (3.14), it needs to be noted that the dynamic power of TSV I/O increases with temperature due to nonlinear MOSCAP. Additionally, the drivers attached to one TSV I/O channel also dissipate power due to its load capacitance C_L . Thus, the total power dissipation P_{TSV_total} considering TSV channel power P_{TSV} can be given as

$$P_{TSV_total} = P_{TSV} + N \cdot C_L \cdot V_{DD}^2 \cdot f; \quad (3.15)$$

where P_{TSV} is N -channel TSV power, as given in (3.14).

Simulations to obtain the dynamic power of 3D TSV interconnects are carried under different temperature and length constraints are carried out and will be compared with its counterpart TSI in the next section.

3.4 TSI Device Model

2.5D integration by TSI realized as T-line is another alternative for many-core memory-logic integration, in which the thermal reliability concern is alleviated. The heat spreading in 2.5D integration is much stronger than in 3D integration, because the heat-sink is closer to the substrate layer, and there would be less hotspots and temperature gradient in 2.5D IC, as one can observe in Figure 3.9. Moreover, as the dimension of TSVs in TSIs are smaller and due to less operating temperatures, mechanical impact can be neglected.

Figure 3.9: 2.5D integration of cores on one common substrate using TSI

However, as liner material is still in presence as a part of TSI, a temperature dependent delay model for TSI has to be developed similar to that of TSVs. Due to existence of liner material, TSI capacitance needs to be modeled as a MOSCAP varying nonlinearly with temperature, as given in (3.2) and (3.3).

3.4.1 Delay Model

Previous Models A few works have been carried out on modeling of delay model for TSI interconnects. In [111, 112, 113, 114], TSI lines are modeled as traditional RC interconnects similar to (3.1). As TSIs are designed for middle-distance communication, RC interconnects incurs larger loss and thus fabricated as transmission lines, as shown in 2.9. In [10], they are modeled as transmission lines (T-lines) to achieve higher bandwidth with lower power. If one can observe the physical structure of TSIs, the liner is still in contact with metal fill, and the above works does not consider the impact of liner.

Proposed Model When compared to the 2D RC-interconnects, TSI realized as transmission line (T-line) has smaller delay, low power and higher bandwidth but larger area cost. However, as TSIs are deployed underneath the substrate with no area overhead (Figure 3.9). As such, it is ideal to realize TSI by T-line for a high speed and low power 2.5D I/Os, as shown in Figure 3.10(a). T-line can be designed as single-ended T-line (STL) or differential-ended T-line (DTL) depending on the requirements. Note that the delay of T-line depends on its characteristic impedance and operating mode. Hence, the characteristic impedance is first determined to model the temperature dependent delay.

Figure 3.10: Temperature dependent delay model of one TSI STL or DTL I/O channel

3.4.1.1 T-line Model

The characteristic impedance Z_s of a single-ended T-line (STL) in Figure 3.10(b)(i) is

$$Z_s = \sqrt{\frac{(R + j\omega L)}{(G + j\omega C_{TI})}}; \quad \gamma = \sqrt{(R + j\omega L)(G + j\omega C_{TI})} \quad (3.16)$$

where γ is the propagation constant; ω is the angular frequency; C_{TI} is the temperature dependent TSI capacitance ($C_{TI} = C + \beta_1 T + \beta_2 T^2$); and based on [70], resistance R , inductance L , capacitance C and shunt conductance G per unit length can be determined as

$$R = \frac{R_s}{\pi a} \left(\frac{\frac{d}{2a}}{\sqrt{(\frac{d}{2a})^2 - 1}} \right); \quad L = \frac{\mu_0}{\pi} \ln\left(\left[\frac{d}{2a}\right] + \sqrt{\left(\frac{d}{2a}\right)^2 - 1}\right);$$

$$G = \frac{\pi}{\ln\left(\left[\frac{d}{2a}\right] + \sqrt{\left(\frac{d}{2a}\right)^2 - 1}\right)}; \quad C = \frac{\pi \epsilon}{\ln\left(\left[\frac{d}{2a}\right] + \sqrt{\left(\frac{d}{2a}\right)^2 - 1}\right)}. \quad (3.17)$$

Here, $R_s (= \sqrt{\pi \mu_0 f / \sigma})$ is the surface resistance of T-line for skin effect; a and d are the radius and pitch of the T-line respectively; μ_0 and ϵ are the permittivity of vacuum and permeability of material respectively; σ is the conductivity of the conductor; ω is the angular frequency; and f is the operating frequency.

For a differential T-line (DTL), there exists a mutual inductive and capacitive coupling, because of which characteristic impedance Z_{diff} becomes

$$Z_{diff} = 2 \sqrt{\frac{(L - L_m)}{(C_{TI} + C_m)}} \quad (3.18)$$

where C_m , L_m are mutual inductance and capacitance of DTL respectively; and C_{TI} is

temperature dependent TSI capacitance.

3.4.1.2 Delay of T-line

For a T-line operating above takeover frequency (lossless LC region), the temperature dependent delay of a T-line D_{TSI} can be given as

$$D_{TSI} = \frac{1}{\omega} \sqrt{\text{Im}(\gamma Z_0) \times \text{Im}\left(\frac{\gamma}{Z_0}\right)} = \sqrt{LC_{TI}}$$

$$\gamma = \begin{cases} j\omega\sqrt{LC_{TI}} & STL; \\ 2\omega\sqrt{LC_{TI}} & DTL; \end{cases} \quad Z_0 = \begin{cases} \sqrt{\frac{L}{C_{TI}}} & STL; \\ 2\sqrt{\frac{L}{C_{TI}}} & DTL; \end{cases} \quad (3.19)$$

where $\text{Im}(x)$ indicates the imaginary part of x ; Z_0 is the characteristic impedance of T-line which can be either DTL or STL as in (3.16) and (3.18) respectively; and C_{TI} is the temperature dependent TSI capacitance.

Thus, the delay of TSI can be written as a function of inductance L and capacitance C_{TI} , nonlinearly dependent on temperature T . The simplified TSI delay based on (3.19) can be given as

$$D_{TSI} = \sqrt{L(C + \beta_1 T + \beta_2 T^2)}. \quad (3.20)$$

The delay of TSI connected with a driver with load capacitance C_L and a resistance of R_D can be given as

$$D_{25d-io} = R_D C_L + D_{TSI}. \quad (3.21)$$

When compared to the temperature dependent delay model of 3D TSV I/O in (3.8), one can observe that the temperature dependence of 2.5D TSI given in (3.21) is much weaker with square-root dependence. This makes 2.5D TSI delay less prone to temperature variation compared to 3D TSV. The temperature gradient δT is much smaller for 2.5D TSI than 3D TSV interconnects. The comparison of 3D TSV and 2.5D TSI I/Os delay with temperature and length will be presented later.

Accuracy and Validation As the developed model is based on the physical composition of the TSI, it is assumed to be more precise than the previous models. There are no measurement results reported for TSIs. The value of delay for a TSI of length 1.5mm is reported to be nearly 10ps, whereas the value of TSI delay for same length for a TSI to interconnect microprocessors reported in [115] is nearly 8ps. The reported model has an error of 2ps, which can be tolerated while simulating large integrated systems.

Figure 3.11: Delay and power comparison under different length for (a) TSV I/Os based on (3.8) and (3.15); (b) TSI I/Os based on (3.21) and (3.22)

3.4.2 Power Model

Based on the driver resistance R_D and characteristic impedance Z_0 in (3.19), dynamic power P_{TSI} of a N -channel TSI I/O can be calculated as

$$P_{TSI} = \frac{N \cdot V_{DD}^2 \cdot s}{(R_D + Z_0)} f; \quad R_D = \frac{\rho d}{A_c} \quad (3.22)$$

where s is the duration of signal pulse; R_D is the driver resistance; d and A_c are the length and cross sectional area of the T-line respectively; and ρ is the resistivity of the material.

The rising trend of Z_0 is linear to the temperature variation which causes power of TSI interconnect less sensitive to the temperature as TSV. By observing (3.22) and (3.15), it can be concluded that the power of TSI I/O channel is less dependent on temperature (with square-root dependence) than TSV I/O channel (quadratic dependence). Additionally, the temperature of a 2.5D IC will be smaller compared to a 3D IC.

3.4.2.1 TSV and TSI Comparison

Delay and power trends under different TSV and TSI interconnect lengths are discussed here. TSV and TSIs are modeled as discussed previously. Figure 3.11(a) shows result for TSV I/Os. The delay of one channel of TSV is around $22.50ps$ of $50\mu m$ length and $40.50ps$ of $90\mu m$ length under $25^\circ C$ respectively with linear relation. The power of TSV shows similar relation with $0.11mW$ of $50\mu m$ length and $0.20mW$ of $90\mu m$ length. Figure 3.11(b) shows result for TSI I/Os. The delay of one channel of TSI is around $34.52ps$ of $1.5mm$ length and $69.04ps$ of $3.0mm$ length under $25^\circ C$ respectively, also with linear relation. The power of TSI is $4.11mW$ for both lengths at $25^\circ C$.

The delay and power comparison under different temperature for TSV and TSI I/Os is presented below. Figure 3.12 shows that the nonlinear temperature-dependent capacitance has much stronger coupling for TSV I/Os than TSI I/Os. For example, the delay

Figure 3.12: Delay and power comparison under different temperature for TSV I/Os (based on (3.8) and (3.15)) and TSI I/Os (based on (3.21) and (3.22))

Figure 3.13: Total power and energy efficiency with different bandwidths: (a) at 25°C; (b) at 100°C based on (3.15) and (3.22)

of TSV and TSI are $22.50ps$ and $34.52ps$ at $25^{\circ}C$ respectively. When the temperature rises to $130^{\circ}C$, the delay of TSV and TSI becomes $31.95ps$ and $35.53ps$, indicating an increase of 42% and 3%, respectively. This shows that the TSV I/O delay is more sensitive to the temperature. Though the power curves look linear or constant, for variation of temperature from $25^{\circ}C$ to $130^{\circ}C$ the power of TSV and TSI shows incremental change of 20% and 2% respectively.

3.4.2.2 Energy-efficiency Analysis

The power and energy-efficiency relation with bandwidth is presented in Figure 3.13. The bandwidth can be adjusted by varying the number of I/O channels. On average, 2.5D integration consumes 62% more power than 3D integration at $25^{\circ}C$. If heat dissipation is not well designed, temperature will form a positive feedback loop with leakage current and result in thermal runaway failure. Note that the delay of the TSVs will be greatly affected when the temperature is high, thus decreases the bandwidth. Here the energy-efficiency is defined as energy consumption per bit. It can be seen that energy efficiency will decrease as the number of channel increases. On average, 2.5D integra-

tion consumes slightly more energy per bit than 3D integration at 25°C. But when the temperature rises to 100°C, 2.5D integration achieves almost the same energy efficiency as 3D integration. However, with the increase in number of channels, 2.5D TSIs are more energy efficient than 3D TSVs.

3.5 Summary

3D integration by TSVs and 2.5D integration by TSIs are the possible efficient many-core memory-logic integration techniques at large-scale. In 3D IC, signal TSVs utilized for inter-layer signal connections, are modeled as MOSCAP, varying nonlinearly with temperature. Further, difference in CTEs and operating temperatures between TSV and substrate results in exertion of stress from TSV on drivers which modifies mobility and delay. Hence, an electrical-thermal-mechanical coupled delay and power models are developed for TSV. Similar power and delay models are developed for TSIs. Performance comparison for 3D TSV and 2.5D TSI are performed. It can be concluded that TSIs are thermal resilient than TSVs. The device models of TSV and TSIs are developed by the respondent author and is published in TCAD'13 and DATE'14 respectively.

Chapter 4

Physical Designs

4.1 Introduction

In this chapter, we present the 3D clock-tree design with the aim to optimize the clock-skew by the insertion of dummy TSVs, followed by low-power I/O buffer design for 2.5D TSIs based on the delay and power models developed in the previous chapter. To dissipate the accumulated heat in top-layers, thermal TSVs i.e., dummy TSVs, used for heat dissipation can be inserted. As a case study, a clock-tree design is considered. The design of the clock-tree is primarily involved with reduction of delay difference at different sinks, known as *skew* [116, 117, 118, 109, 119, 37]. Compared to a clock-tree in 2D ICs, the one in a 3D ICs will experience much larger temperature and stress gradient both vertically and horizontally. As signal TSVs are deployed to route a 3D clock-tree over the entire 3D chip, such a non-uniform temperature difference can lead to a significant clock-skew by electrical-thermal coupling of signal TSVs [120, 119]. Such an electrical-thermal coupling becomes nonlinear when liner material is considered due to the difference in CTE of liner and TSV metal fills and operating temperature. Moreover, the TSV induced stress also affects the mobility and delay of drivers, which further worsens the clock-skew over the entire 3D chip by electrical-mechanical coupling of drivers [74]. As such, the traditional clock-tree design methods [116, 117, 118] without considering temperature and stress gradients will become inaccurate and unreliable. The thermal-aware 3D clock-tree synthesis has been discussed in [108] considering thermal profile. A 3D embedding method was developed in [121] to reduce the wire-length. Further optimization in [122] is developed to reduce power and slew-rate. However, previous methods only conduct the clock network optimization based on linear or none electrical-thermal-mechanical coupling, ignoring the TSV physical model and hence is not accurate. As such, there is no specific problem formulated for clock-tree design

based on the reduction of both thermal and stress gradients in 3D IC.

In 2.5D integration, TSIs are modeled as high-speed, low power transmission lines (T-lines). These T-lines require special design of buffers to achieve better reliability, with low power. Traditional LVDS or CML buffers consume more power. Hence, LVD-S buffer with pre-emphasis stage is designed for achieving better reliability with low power. Similarly, a CML buffer with strong inductive coupling is designed to alleviate the channel loss.

In this chapter, we will first present the 3D clock-tree design with dummy TSV insertion to improve the heat-dissipation with balanced stress and thermal gradients. Further, the design of I/O buffers for TSIs with low power is presented.

4.2 3D Clock-tree Design

Delay in 3D circuits is another parameter that varies with the temperature, hence delay/skew optimization by insertion of TSVs are studied in this section. A 3D clock-tree shown in Figure 4.1 with TSVs for vertical interconnects is considered as a case study for determining the impact of different physical parameters on the electrical delay. The design of a 3D clock-tree is primarily involved with reduction of delay difference at different clock-sinks, known as clock-skew [116, 117, 118, 109, 123, 124, 36, 119]. Compared to clock-tree in a 2D IC, a clock-tree in 3D IC experiences much larger temperature and stress gradient both vertically and horizontally. As such, the traditional clock-tree design methods [116, 117, 118] without considering both temperature and stress gradients simultaneously will become inaccurate and unreliable. Additionally, such methods only conduct the clock network optimization based on a linear or a partial or no electrical-thermal-mechanical coupling, ignoring the TSV physical model and hence is not accurate. As such, there is no specific problem formulated for clock-tree design considering the reduction of both thermal and stress gradients in 3D IC. In a 3D clock-tree design, unlike 2D design, all the impacts discussed in previous chapter needs to be considered. A nonlinear problem is formulated to reduce the clock-skew.

4.2.1 Problem Formulation

By considering all the physical and electrical impacts, delay at a clock-sink i of a 3D clock-tree needs to be modeled as a nonlinear function Γ of temperature and stress gradients. Thus, the delay at the i^{th} clock-sink can be modeled as $\delta D_i = f(\Gamma)$. *Clock-skew S* is defined as the maximal delay difference between two clock-sinks. Note that by adding dummy TSVs, one can balance the temperature and stress gradients and can further re-

Figure 4.1: 3D clock-tree distribution network at different tiers: (a) Clock-tree with 14 TSV bundle locations in a H-tree; (b) clock-tree with 28 TSV bundle locations in a H-tree; (c) layer configuration under non-uniform temperature distribution

duce delay and skew. As such, we have the following problem formulation towards reliable 3D clock-tree design under temperature and stress gradient.

Problem 1: For a pre-synthesized zero-skew 3D clock-tree with N_S sinks, using signal TSVs for inter-tier connections, the clock-skew S needs to be estimated by considering the position and number of TSVs, i.e., temperature and stress gradients Γ , and by considering nonlinear electrical-thermal-mechanical coupling. A large number of dummy TSVs that can be inserted to minimize S under Γ .

$$S = \max : |\delta D_i - \delta D_j|, 0 \leq i, j \leq N_S \quad (4.1)$$

where δD_i and δD_j are delays from sinks i, j respectively.

A nonlinear optimization is deployed to further minimize the clock-skew under non-uniform temperature and stress gradients. Note the pre-synthesized zero-skew 3D clock-tree is based on the work in [107] considering the wire length and driver but without electrical-thermal-mechanical coupling. To solve the above mentioned clock-skew reduction problem by insertion of dummy TSVs, reduction in temperature and stress gradients, their corresponding sensitivities, and skew sensitivity with dummy TSV density needs to be studied first.

Figure 4.2: (a) 3D heat-removal path by dummy TSVs; (b) 3D-view of dummy TSV insertion

4.2.1.1 Thermal and Stress Gradient Sensitivity Analysis

As aforementioned, dummy TSVs are used for vertical interconnects without signal connection through it. As dummy TSVs are filled with metal-material copper (Cu) having a good thermal conductivity of $400W/m.K$, it can provide vertical heat-dissipation path to reduce the temperature gradient. One can observe from (3.4), the exerted mechanical stress decreases with temperature. Altogether, insertion of dummy TSVs can balance density of TSV distribution and reduce temperature and stress gradients. However, insertion of dummy TSVs at improper locations can further worsen temperature and stress gradients. To solve the above mentioned clock-skew problem by insertion of dummy TSVs, reduction in temperature and stress gradients, their corresponding sensitivities, and skew sensitivity with dummy TSV density needs to be studied.

Reduction of Thermal Gradient For a chip level analysis, impact of single dummy TSV is ineffective. Dummy TSVs are modeled in terms of local density η as shown in Figure 4.2, where dummy TSVs occupy an area of ηA on a regular chip of area A . Considering the vertical heat-dissipation, the thermal conductivity λ , given by

$$\lambda = (\eta + \delta\eta)\lambda_{TSV} + (1 - (\eta + \delta\eta))\lambda_0; \quad \delta n = \delta\eta A \quad (4.2)$$

where λ_0 is the initial thermal conductivity; η is the initial TSV density; $\delta\eta$ is the change in TSV density; and δn represents change in number of TSVs with respect to initial number of TSVs n .

Temperature gradient reduction with the change in TSV density $\delta\eta$ is given by

$$\delta T = T - T_0 = \frac{P \cdot l}{A\lambda_0} \cdot \frac{\delta\eta}{\frac{\lambda}{\lambda_{TSV} - \lambda_0} + \eta + \delta\eta} \quad (4.3)$$

where P is the heat power flowing from chip to heat-sink; and l is the length of heat-transfer path distance with a chip area of A . From (4.3), one needs to note that as $\delta\eta$ approaches or becomes larger than $\lambda/(\lambda_{TSV} - \lambda_0) + \eta$, the reduction in temperature due to dummy TSVs becomes saturated.

Further, sensitivity of temperature gradient with TSV density is studied. In (4.3), it needs to be noted that after a certain dummy TSV density the reduction in temperature gradient tends to saturate. Sensitivity of temperature gradient with respect to the dummy TSV density can be given by

$$\frac{\partial T}{\partial \eta} = \frac{P \cdot l}{A\lambda_0} \cdot \frac{\eta_0}{(\eta_0 + \eta)^2}; \quad \eta_0 = \frac{\lambda_0}{\lambda_{TSV} + \lambda_0}. \quad (4.4)$$

From (4.4), it can be clearly concluded that, when the dummy TSV density η is smaller than its saturation value η_0 , i.e., $\eta \ll \eta_0$, the sensitivity of temperature gradient with dummy TSV density remains almost constant; and as $\eta \gg \eta_0$ the sensitivity approaches zero, implying that reduction of temperature tends to saturate. Thus, temperature sensitivity function with dependence on dummy TSV density can be used during the optimization of dummy TSV insertion.

To study the temperature gradient experimentally, a 4-layer stacked 3D IC is considered with each layer provided with same power density of P , which serves as the heat source and simulated in COMSOL. Study is performed for different values of thermal power densities with $P = \{6, 80, 115\} W/m^2$. In experiment, temperature at each layer is collected without dummy TSVs initially. Note that the liner material of dummy TSV is Si_3N_4 , of which the thickness is $200nm$. Based on the formed temperature distribution, dummy TSVs are inserted with density upper and lower bounds.

Temperature reduction in each layer with insertion of dummy TSVs is presented in Figure 4.3. It is experimentally observed that the reduction in temperature initially increases but tends to saturate after a particular limit. It can also be observed that reduction in temperature for bottom layer, i.e., layer-3 is less than other layers. As it is close to heat-sink, insertion of dummy TSV does not make big difference compared with other layers. In addition, the maximum inserted dummy density observed is $400/mm^2$ with the maximum temperature reduction of nearly $50^\circ C$ in layer-0 i.e. the top layer.

Figure 4.3: Temperature gradient reduction in a 4-layer 3D clock-tree with dummy TSVs under different power densities based on (4.4)

Reduction of Stress Gradient For calculating the stress gradient variation, we consider a set up of a unit square having four TSVs at its four corners. All the transistors or devices inside the square will experience the stress from all the four TSVs. The stress contour from TSVs and its impact on each of transistors is shown in Figure 3.2(b). The stress on each of the transistors can be calculated using (3.4). What is more, it needs to be observed that there will be reduction in stress gradient with insertion of dummy TSVs at proper locations, and balances stress due to reduction of temperature. The stress gradient reduction $\delta\sigma$ caused by TSV density difference $\delta\eta$ can be given as

$$\delta\sigma = -\frac{B\Delta\alpha\Delta T}{2}\left(\frac{R}{r}\right)^2\delta n = -\frac{B\Delta\alpha\Delta T}{2}\left(\frac{R}{r}\right)^2\delta\eta A \quad (4.5)$$

where $\delta\eta$ represents change in TSV density due to insertion of δn additional TSVs. When the density becomes more uniform, the stress gradient also becomes smaller.

Note that stress and stress gradient depends on TSV density and temperature gradient. Stress gradient at different temperature gradients and TSV densities is shown in Figure 4.4. For the purpose of simulation, two blocks A and B in which TSV density can be varied and constant respectively are considered. As shown in Figure 4.4(a), block B has a constant TSV density of $400/mm^2$. Block A has a temperature gradient of $180^\circ C$ and its TSV density can be varied. The stress gradient between the blocks A and B is shown in Figure 4.4(b). As TSV density is increased, the stress gradient tends

Figure 4.4: (a) Setup of TSV distributions for stress gradient calculation; (b) variation of stress gradient reduction with TSV density and temperature gradient

to decrease. The stress gradient reduction for same setup but with temperature gradient of 250°C is also plotted. The reduction in stress gradient with TSV density is given as

$$\frac{\partial \sigma}{\partial \eta} = -\frac{B\Delta\alpha\Delta T A}{2}\left(\frac{R}{r}\right)^2. \quad (4.6)$$

The stress gradient sensitivity in (4.6) may look as independent from TSV density, but depends on radius of TSV and area, which has impact on TSV density. When the radius of TSV becomes very small compared to the distance, the stress gradient tends to saturate early compared to TSVs with larger radius. All these sensitivity analysis will be deployed in the optimization of clock-skew reduction.

4.2.2 Clock-skew Reduction

The nonlinear electrical-thermal-mechanical coupling transforms the 3D clock-skew reduction problem into a nonlinear optimization problem. Here, a nonlinear programming based algorithm is developed for the insertion of dummy TSVs to reduce the clock-skew. The nonlinear optimization of 3D clock-tree is performed at micro-architecture level by dividing each layer into $M \times N$ grids. If one TSV passes through a grid g_i , the delay contributed by that grid needs to be calculated by the developed coupled electrical-thermal-mechanical model in (3.12). Generally, temperature has much higher impact than stress. Moreover, the linear delay from horizontal metal wires and buffers are also considered in (4.7). The skew from i^{th} individual grid can be given as

$$S_i = \begin{cases} S_c + [w_1 \delta T_i + w_2 (\delta T_i)^2 + w_3 (\delta T_i)^3] \\ + [s_0 + s_1 \delta T_i + s_2 (\delta T_i)^2] \delta \sigma_i : & 3D \text{ signal TSVs;} \\ z_0 + z_1 \delta T_i : & 2D \text{ wires;} \end{cases} \quad (4.7)$$

where

$$\begin{aligned}
S_c &= D_c + [R_0\beta_1 + \alpha R_0C_0 + S_{w1}R_{w1}S_T\beta_1 + \frac{R_0\alpha(S_{w2}C_{w2} + S_L C_L)}{S_T}]T_0 + R_0\beta_2T_0^2; \\
w_1 &= 2R_0(\beta_2 + \alpha\beta_1)T_0 + [R_0\beta_1 + \alpha R_0C_0 + S_{w1}R_{w1}S_T\beta_1 + \frac{R_0\alpha(S_{w2}C_{w2} + S_L C_L)}{S_T}]; \\
w_2 &= 2R_0\alpha\beta_2T_0 + R_0(\beta_2 + \alpha\beta_1); \quad w_3 = R_0\beta_2;
\end{aligned} \tag{4.8}$$

and

$$\begin{aligned}
s_0 &= -\frac{mR_D}{S_D(1+m\sigma_0)^2}[(S_DC_P + S_{w1}C_{w1} + S_TC_0 + S_L C_L + S_{w2}C_{w2}) + S_T\beta_1T_0 + S_T\beta_2T_0^2]; \\
s_1 &= -\frac{mR_D}{S_D(1+m\sigma_0)^2}[S_T\beta_1 + 2S_T\beta_2T_0]; \quad s_2 = -\frac{mR_D}{S_D(1+m\sigma_0)^2}S_T\beta_2;
\end{aligned}$$

$$z_0 = R_0C_0; \quad z_1 = R_0C_0\alpha. \tag{4.9}$$

Here w_i , s_i are the skew coefficients in the presence of TSV; z_0 , z_1 are the skew coefficients in the absence of TSV; δT_i and $\delta \sigma_i$ represent temperature and stress gradients in i^{th} grid respectively; and S_c is the temperature and stress independent coefficient.

To optimize the clock-skew, dummy TSVs can be inserted to reduce the temperature and stress gradients. Here the impact of dummy TSV insertion on temperature and stress gradients and corresponding sensitivities are presented, followed by clock-skew sensitivity to dummy TSV density and nonlinear optimization for clock-skew reduction.

4.2.2.1 Clock-skew Sensitivity

The sensitivity of the clock-skew with respect to dummy TSV density plays an important role during the optimization. From (4.7), the sensitivity of clock-skew in i^{th} grid can be derived as follows

$$\frac{\partial S_i}{\partial \eta_i} = (S_{T,T}^i + S_{T,\sigma}^i) \cdot \frac{\partial T}{\partial \eta} + (S_{\sigma,\sigma}^i + S_{\sigma,T}^i) \cdot \frac{\partial \sigma}{\partial \eta}; \tag{4.10}$$

where

$$S_{T,T}^i = w_1 + 2w_2\delta T_i + 3w_3(\delta T_i)^2;$$

$$S_{T,\sigma}^i = (s_1 + 2s_2\delta T_i)\delta\sigma_i; \quad (4.11)$$

$$S_{\sigma,\sigma}^i = s_0; \quad S_{\sigma,T}^i = s_0 + s_1\delta T_i + s_2(\delta T_i)^2.$$

Here, $S_{T,T}^i$ and $S_{T,\sigma}^i$ represent the temperature and temperature-stress gradient coefficients for temperature sensitivity in the i^{th} grid; $S_{\sigma,\sigma}^i$ and $S_{\sigma,T}^i$ represent the stress and stress-temperature gradient coefficients for stress sensitivity in the i^{th} grid; and s_i, w_i are the skew coefficients in the presence of TSV and is given in (4.7).

The clock-skew sensitivity depends on the temperature and stress gradient sensitivities, which are eventually related to the dummy TSV density, and can be determined from (4.4) and (4.6). Based on the clock-skew sensitivity, the updated temperature and stress based on the new dummy TSV density η_i , in the i^{th} grid can be given by

$$T_i^{new} = T_i + \gamma_T^i P_i \eta_i; \quad \sigma_i^{new} = \sigma_i + \gamma_\sigma^i \eta_i \quad (4.12)$$

where γ_T^i and γ_σ^i are temperature and stress gradients sensitivity in the i^{th} grid with respect to the dummy TSV density, and are given as $\partial T_i / \partial \eta_i$ and $\partial \sigma_i / \partial \eta_i$ determined from (4.4) and (4.6), respectively.

Moreover, T_i, σ_i represent the temperature and stress in the i^{th} grid; and P_i, η_i represents the heat power density and TSV density respectively in the i^{th} grid. Based on the updated values of temperature and stress and by updating the TSV density, the skew and its sensitivity values in (4.7) and (4.10) can be updated.

4.2.2.2 Nonlinear Optimization

Clock-tree branch B_k is a set of grids that branch k , passes through, $B_k = \{g_i | \text{branch } k \text{ passes } g_i\}$, with g_i representing i^{th} grid. Therefore, the clock-skew of a clock-tree branch is the sum of skews from all the grids it passes through

$$\mathbf{S} = \sum_{i \in B_k} S_i \quad (4.13)$$

where S_i represents the skew from i^{th} grid among set B_k . Based on the derived sensitivity and skew function, a nonlinear optimization can be performed.

Substituting (4.7), (4.10) and (4.12) into (4.13), the clock-tree branch skew $\mathbf{S}$ converts to a quadratic function of inserted dummy TSV density η_i . By considering clock-

skew from each grid, one can represent clock-skew into a matrix form as

$$\mathbf{S} = c + f^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T H \mathbf{x} \quad (4.14)$$

where

$$c = S_c; \quad f = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{(M \times N)} \end{pmatrix}; \quad \mathbf{x} = \begin{pmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{(M \times N)} \end{pmatrix}; \quad H = \begin{pmatrix} H_{0,0} & \cdots & H_{0,N_s} \\ H_{1,0} & \cdots & H_{1,N_s} \\ \vdots & \ddots & \vdots \\ H_{(M \times N),0} & \cdots & H_{(M \times N),N_s} \end{pmatrix}$$

$$f_i = \begin{cases} (S_{T,T}^i + S_{T,\sigma}^i) \gamma_T^i + (S_{\sigma,\sigma}^i + S_{\sigma,T}^i) \gamma_\sigma^i : & \text{if } i \in B_k; \\ 0 : & \text{else}; \end{cases}$$

$$H_{i,j} = \begin{cases} (6w_3 \delta T_j \gamma_T^j + 2w_2 \gamma_T^j + s_1 \gamma_\sigma^j + 2s_2 \delta T_j + 2s_2 \delta_\sigma^i \gamma_T^j + 2s_2 \delta T_i \gamma_\sigma^j) \gamma_T^i \\ + (s_1 \gamma_T^j + 2s_2 \delta T_i \gamma_T^j) \gamma_\sigma^i : & \text{if } i, j \in B_k; \\ 0 : & \text{else}. \end{cases}$$

Here, c represents the zero-order coefficient of clock-skew; f , H represent the linear and nonlinear coefficients of clock-skew respectively; $\mathbf{x}$ represents the dummy TSV density vector; N_s represents total number of sinks; $M \times N$ is the total number of grids; $S_{T,T}^i$, $S_{T,\sigma}^i$, $S_{\sigma,\sigma}^i$, $S_{\sigma,T}^i$ are the coefficients of clock-skew sensitivity in i^{th} grid given in (4.11); γ_T^i and γ_σ^i are temperature and stress gradient sensitivities in i^{th} grid given by (4.12); P_i represents the thermal power in the i^{th} branch; B_k represents the clock-tree branch; S_c is the coefficient of clock-skew defined in (4.8); δT_i , $\delta \sigma_i$ are the temperature, stress gradients in i^{th} grid; and w_i, s_i are the coefficients of clock-skew and is given in (4.7). Note that each column in f vector and H matrix represents the clock-skew sensitivity.

Since clock-skew is the difference in the delay between two clock-sinks, the problem thus becomes to minimize the skew variance over all clock-tree branches B_k , i.e., to minimize variance of $\mathbf{S}$ in (4.14)

$$\min : f(S) = \frac{1}{N_s - 1} \sum_{k=1}^{N_s} (\mathbf{S} - \bar{\mathbf{S}})^2 \quad (4.15)$$

where $\mathbf{S}$ represents the clock-skew for B_k clock-tree branches; and $\bar{\mathbf{S}}$ represents the average skew of $\mathbf{S}$ for N_s sinks.

As such, $f(S)$ becomes a quadratic function of $\mathbf{x}$, given by

$$\bar{S} = \frac{1}{N_s} \sum_{k=1}^C \mathbf{S} = \bar{c} + \bar{f}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \bar{H} \mathbf{x}. \quad (4.16)$$

Where c and $\bar{c}$ represents the zero-order coefficient of clock-skew and its mean; f and $\bar{f}$ represent the linear coefficient vector of clock-skew and its mean; H and $\bar{H}$ represent the nonlinear coefficient matrix of clock-skew and its mean. Substituting (4.16) in (4.15), the original problem can be rewritten as one polynomial function by

Problem 2:

$$\begin{aligned} \min : f(\mathbf{x}) = & \frac{1}{N_s - 1} \sum_{k=1}^{N_s} (\hat{c}^2 + 2\hat{c}\hat{f}^T \mathbf{x} + \mathbf{x}^T (\hat{f}\hat{f}^T + \hat{c}\hat{H}_k) \mathbf{x} + \hat{f}^T \mathbf{x}^T \mathbf{x} \hat{H}_k \mathbf{x} \\ & + \frac{1}{4} \mathbf{x}^T \hat{H}_k \mathbf{x} \mathbf{x}^T \hat{H}_k \mathbf{x}) \end{aligned} \quad (4.17)$$

where $\hat{c} = c - \bar{c}$, $\hat{f} = f - \bar{f}$ and $\hat{H} = H - \bar{H}$. All these values represent the deviations from their respective means.

As the clock-skew depends on the TSV density matrix $\mathbf{x}$, neither too many nor too few TSVs can be inserted, because of the design constraint below

$$lb \leq \mathbf{x} \leq ub \quad (4.18)$$

where lower bound lb is determined by foundry process such as minimum metal density, and upper bound ub is determined by the maximum allowed overhead with respect to area and signal routing.

4.2.2.3 Conjugate Gradient Solving

Now the objective to minimize the clock-skew becomes minimizing (4.17). This can be done by finding the optimum value of the dummy TSV density vector $\mathbf{x}$, that also satisfies constraints in (4.18). Conjugate gradient method with line-search in [125] can be an efficient method to solve this nonlinear equation with given constraints.

To remove inequalities in Problem 2, Karush-Kuhn-Tucker (KKT) optimization along with Lagrange penalty factor ξ is used to reformulate original problem as

Problem 3:

$$\min : f^*(\mathbf{x}) = f(\mathbf{x}) + \xi h^2(\mathbf{x}) \quad (4.19)$$

with

$$h(\mathbf{x}) = \begin{cases} 0, & lb \leq \mathbf{x} \leq ub \\ \varphi >> 0, & otherwise \end{cases} \quad (4.20)$$

where $f^*(\mathbf{x})$ is the objective function by considering boundary conditions i.e., removing inequalities of $f(\mathbf{x})$; ξ is the Lagrange penalty factor, determined as a stationary point of $f(\mathbf{x})$; and φ is a weighting parameter.

Conjugate gradient method iteratively searches for the value of $\mathbf{x}$ that can minimize $f^*(\mathbf{x})$ along the search gradient vector g_k , which points to the direction where the greatest rate of variation of objective function $f^*(\mathbf{x})$ lies. To achieve a converged solution, at iteration k , the search direction vector d_k , which indicates the direction where the objective variable has to be varied, moves in a negative gradient direction to minimize the variation. Thus, a new search direction vector d_{k+1} can be obtained by linear addition of the previous search direction vector d_k with the current negative search gradient vector g_k . Therefore, the next search direction vector becomes

$$d_{k+1} = -\nabla f^*(\mathbf{x}_k)^T + \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} d_k; \quad g_k = -\nabla f^*(\mathbf{x}_k). \quad (4.21)$$

Where d_{k+1} is the search direction vector; and g_k, g_k^T are search gradient vector of $\mathbf{x}_k$ and its transpose, determined from the slope of search vector.

Based on the search direction vector d_k , and gradient g_k , the optimal value for step-size α_k is decided to minimize the function $f^*(\mathbf{x}_k + \alpha_k d_k)$. The new vector $\mathbf{x}$ can be updated based on the previous search direction vector d_k as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k d_k; \quad \alpha_k = \frac{g_k^T g_k}{g_k^T \nabla f^*(\mathbf{x}) g_k}. \quad (4.22)$$

This iterative search for the optimum TSV density vector $\mathbf{x}$ stops when the difference in successive approximations of $\mathbf{x}_k$ reaches the threshold. To perform with the faster convergence and avoid the local minimum, the problem is solved by starting with some randomly generated approximation of $\mathbf{x}_0$. Once the final value for $\mathbf{x}$ is reached, then it can satisfy (4.19) and density constraints based on the stress and temperature gradients, leading to the reduction of the clock-skew. The nonlinear optimization of clock-skew reduction for 3D clock-tree design by the insertion of dummy TSVs is presented below.

With the help of 3D-ACME [126], a 3D-IC thermal simulator based on Hotspot [127], temperature distribution at each layer can be obtained. Moreover, average temperature based on SPEC 2000 [128] benchmarks are taken to avoid application specific temperature distribution [108].

Figure 4.5: Clock-skew reduction before and after insertion of dummy TSVs

As the clock-skew is the difference in delay between two sinks, the variation of clock-skew with TSVs of one 3D H-tree is shown in Figure 4.5. It can be noted from Figure 4.5, that the initial clock-skew without insertion of dummy TSVs is higher than the clock-skew after insertion of dummy TSVs. For example, for a row-grid 16 and column-grid 24, having dummy TSV inserted results in a clock-skew reduction of nearly 17ps. To reduce the area overhead caused by dummy TSVs, the dummy TSV insertion density is limited to 7% of the total area.

Additionally, a 3D clock-tree from the IBM benchmark *r5* [129] is studied. The 3D clock-tree after insertion of dummy TSVs is shown in Figure 4.6 with TSVs in each layer indicated by solid dots. Dummy TSV insertion is performed under nonlinear optimization as discussed in this section. One can observe that a large number of TSVs are inserted in the top layer, i.e., tier 0 compared to other layers since the top layer is farthest one from the heat-sink.

Finally, the comparison of clock-skew for different benchmarks before and after insertion of dummy TSVs is given in Table 4.1 with a detailed summary. It presents the impact of 3D electrical-thermal-mechanical coupled delay model and the insertion of dummy TSVs to reduce gradient. Clock-skew values are reported in pico-seconds and runtime in seconds. The delay models with consideration of nonlinear electrical-thermal-mechanical-thermal impacts result in clock-skew reduction by 61.3% on an average, listed under '*nonlin*' column, compared to clock-skew without insertion of dummy TSVs, listed under '*Orig*' column. Note that the reduced clock-skew by linear modeling is listed under '*Lin*' column with 49.1% clock-skew reduction compared to '*Orig*'. A reduction of 12.2% in in clock-skew observed by the nonlinear than the linear model.

Figure 4.6: 3D clock-tree after insertion of dummy TSV (black dots) with balanced clock-skews for: (a) Tier 0; (b) Tier 1; (c) Tier 2; (d) Tier 3

Table 4.1: 3D clock-skew reduction by linear and nonlinear delay models

Type	HTree1 (14 signal TSVs & 63 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	15.39	9.54	38.01%	16.98	2.45	84.08%	76.15
T4	26.62	8.07	69.68%	16.85	3.92	85.27%	77.11
T8	47.41	11.38	76.00%	17.05	7.03	85.17%	77.19
T10	58.64	14.8	74.76%	17.22	10.02	82.91%	77.36
Mean	-	-	64.61%	17.02	-	84.36%	76.95
Type	HTree2 (28 Signal TSVs & 64 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	23.55	8.15	65.39%	17.41	3.31	85.94%	79.02
T4	44.13	11.61	73.69%	17.51	4.81	89.10%	79.59
T8	82.00	14.60	82.20%	17.40	9.00	89.02%	79.98
T10	103.70	15.50	82.80%	17.49	9.51	90.83%	79.32
Mean	-	-	76.58%	17.45	-	88.72%	79.48
Type	r1 (45 Signal TSVs & 202 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	30.50	17.48	42.69%	49.80	14.46	52.59%	127.29
T4	61.87	32.66	47.21%	35.16	25.55	58.71%	164.01
T8	121.10	66.14	45.39%	38.28	53.18	56.08%	182.71
T10	152.7	83.56	45.28%	44.64	69.17	54.70%	197.49
Mean	-	-	45.14%	42.00	-	55.52%	167.88
Type	r2 (60 Signal TSVs & 365 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	35.13	23.48	33.16%	160.80	19.33	44.97%	475.36
T4	69.75	46.28	33.65%	123.36	34.17	51.00%	485.19
T8	134.70	86.89	35.49%	127.80	71.01	47.28%	490.58
T10	169.00	118.89	29.65%	167.16	88.08	47.88%	513.43
Mean	-	-	32.99%	144.84	-	55.52%	491.14
Type	r3 (75 Signal TSVs & 515 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	32.36	20.19	37.60%	264.84	19.21	40.63%	751.53
T4	64.80	39.97	38.32%	204.96	31.17	51.89%	682.90
T8	125.60	79.86	36.41%	213.36	60.71	51.67%	658.80
T10	157.70	95.47	39.46%	277.20	82.80	47.50%	708.39
Mean	-	-	37.95%	240.12	-	47.92%	700.40
Type	r4 (90 Signal TSVs & 604 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	31.68	17.97	43.28%	254.16	16.53	47.83%	912.39
T4	64.57	36.21	43.92%	278.64	26.13	59.53%	844.46
T8	126.80	70.61	44.31%	280.80	64.50	49.13%	898.21
T10	159.80	88.81	44.42%	393.12	76.83	51.92%	909.25
Mean	-	-	43.95%	301.44	-	52.10%	891.08
Type	r5 (90 Signal TSVs & 1479 buffers)						
	Orig	Lin	Impr%	Time (s)	Nonlin	Impr%	Time(s)
T2	35.00	21.00	40.00%	798.72	18.90	46.00%	1992.60
T4	68.40	38.40	43.86%	834.00	28.50	58.33%	1843.77
T8	131.00	74.60	43.05%	870.24	63.80	51.30%	1705.82
T10	164.10	93.40	43.08%	1126.20	79.43	51.60%	1934.662
Mean	-	-	42.08%	907.32	-	51.81%	1869.21
Overall	-	-	49.10%	-	-	61.30%	-

Figure 4.7: 2.5D TSI I/O interconnection between core and memories its cross section view

4.3 2.5D I/O Design

2.5D integration can significantly improve the bandwidth and reduce the power consumption compared to the conventional 2D integration, in which wire-bonding and PCB-level interconnects introduces a significant latency and power inefficiency [130]. However, different from the RC-interconnect based I/O scheme by 3D TSV, the challenge of 2.5D TSI is to design a high-speed and low-power I/O i.e., a serial data link with high data-rate per Joule. The recent microprocessor I/O in [131] has achieved high-speed performance with less than $1pJ/bit$ in 2D integration. But there are no works to address the 2.5D I/O. Hence, we discuss the low power I/O buffer designs for 2.5D integration. A 2.5D integrated memory-logic system is shown in Figure 4.7 with TSIs deployed underneath the common substrate. One can observe that signals from digital logic of the core are transferred to I/O circuits, and then transmitted or received as differential signals at the two chips. All the I/Os in the core and memory are attached to microbumps, which make TSI connection in silicon substrate.

At very high frequency, there is a frequency-loss from TSI T-line. For example, the resistance of the T-line increases with frequency as,

$$G_{skin}(f) = e^{-(1+j)l\sqrt{\pi\mu\sigma f}} \quad (4.23)$$

where l is T-line length, μ is the permeability, and σ is the conductivity.

Similarly, the dielectric loss of a T-line is given by

$$G_{di}(f) = e^{-l\sqrt{\epsilon_r}\tan\delta f/c} \quad (4.24)$$

where ϵ_r is the dielectric constant; $\tan\delta$ is the loss tangent of material, and c is the speed of light [132]. Thus, one needs to develop methods to compensate the frequency-dependent loss for 2.5D TSI T-line based I/O designs when operating in lossy mode.

The drivers that will be mostly used for T-line can be low-voltage differential signalling (LVDS) buffer and current-mode logic (CML) buffers. Design of low power I/O with LVDS and CML buffers are discussed in the next sections.

4.3.1 LVDS I/O Buffer

Typical LVDS interface shown in Figure 4.8 consists of a current source with nominal current of $3.5mA$ to drive the differential TSI T-line terminated with 100Ω load. LVDS receivers have high input impedance with driving current flowing through the termination resistor, thereby creating a valid logic “one” or “zero” [15]. The resistor at the receiver end can not only provide the optimum T-line impedance matching, but also acts as the load resistor for current source to generate a low-voltage swing of $250mV$ - $400mV$.

Considering the loss of TSI T-line discussed previously, to improve margins of the data eye-diagram, an amplitude pre-emphasis is used in the LVDS design. The schematic of LVDS transmitter with pre-emphasis is shown in Fig 4.9. In the left part, transistors M_5 - M_{12} contribute to a pre-emphasized signal generation. The signal IP and IN are differential data signals coming from digital logic input, and IPD and IND are the reverse input data delayed by a time constant τ .

Assume that the amplitude of pre-emphasis gain is G , i.e., the ratio of current in the pre-emphasis driver I_{em} and the normal driver current I . The transfer function for the pre-emphasized stage can be expressed as

$$H_{pre-em} = \frac{\tau A(s)}{A(s)} = I \cdot (1 - Ge^{-s\tau}) \quad (4.25)$$

where $A(s)$ is the transfer function of normal LVDS transmitter. When a first-order Pade

Figure 4.8: Typical LVDS I/O architecture [15]

Figure 4.9: A LVDS transmitter with pre-emphasis technique

approximation is used in this function, one can have

$$\begin{aligned}
 H_{pre-em} &= I \cdot \left(1 - G \frac{1 - s(\tau/2)}{1 + s(\tau/2)} \right) \\
 &= I \cdot (1 - G) \frac{1 + s(\tau/2)(1 + G)/(1 - G)}{1 + s(\tau/2)}.
 \end{aligned} \tag{4.26}$$

Equation (4.26) reveals that the amplitude pre-emphasis can reduce the DC gain, but can enhance the high frequency response. The transfer function has an additional pole at $1/\tau$. The high frequency compensation, which is $\pm(1 + G)$ in (4.26), is performed when there is a data transition. In this design, use of the digital inverter chain delays the input digital signals with a time constant $\tau = 45ps$. In order to achieve the low-power consumption, the current source of LVDS buffer can be controlled to drive about only $350mV$ peak-to-peak voltage swing at the termination resistor. The ratio of pre-emphasis current gain is set as $G = 0.28$ for the purpose of evaluation.

The timing graph in Figure 4.9 illustrates the operation of the overall LVDS buffer. When the input IP is high at the time instant t_1 , transistors $M_{2,3,6,11}$ are turned on. The main current I flows into a termination resistor to generate the output voltage. At the same time, the delayed and reverse signal IPD is high to turn on transistors M_8 and M_9 at the time instant t_2 . So, during the period between t_1 and t_2 , the amplitude of the output is driven by the sum of two current sources, I and I_{em} . As such, one can achieve a higher voltage swing at the output rising-edge to compensate the high frequency loss. Similarly, in the period between t_3 and t_4 , the output falling-edge is emphasized.

Figure 4.10: A schematic of the proposed CML I/O interface

4.3.2 CML I/O Buffer

In addition to LVDS buffer, design of current-mode-logic (CML) buffer for 2.5D TSI T-line is explored as well. CML is a differential digital logic circuit that can transmit data at high frequency. A two-stage CML I/O buffer as transmitter with a pre-amp stage is shown in Figure 4.10. Two CML buffers are cascaded such that the output of pre-amp stage is connected to the input of driving stage. When the digital logic input is IN, the differential branches of CML buffers are ON and OFF that are controlled by two input transistors M_1 and M_2 . Note that the CML buffer requires a full current switching and the current flows through the ON branch only. In this design, there is no tail current used to achieve a full current swing. The currents through the resistor and inductor loads are controlled by the input voltage swing of two transistors (V_{GS}) and their size. Thus, the size of two pairs of transistors are designed carefully in order to provide enough driving ability with low current consumption at the same time.

Moreover, CML buffer is required to drive load through the microbump pad with 3mm TSI T-line, which has smaller load than the bonding wire with package trace. This means that NMOS transistors of the second stage CML buffer in Figure 4.10 can have a smaller size and smaller gate-to-channel capacitance. Thus, voltage level of power supply can be lowered to 0.6V to decrease the power consumption.

Additionally, to compensate high frequency loss of a 2.5D TSI T-line, inductor loads are designed as cross coupled. Hence, there is a fast data-pass through inductors when the CML buffer transmits high speed signal. The first-stage of CML transmitter works as a pre-amplifier and provides a common mode voltage input to the next stage to ensure the input NMOS pairs in the second stage work in saturation region all the time.

4.3.3 Simulation Results

For the evaluation of 2.5D TSI based integration by CML and LVDS buffers, all chips are assembled face down and are attached to microbumps. They are connected by $20\mu m$

Figure 4.11: TSI realized as T-line

Figure 4.12: Layout of two I/O Interface

width TSI T-line through the common substrate. The I/O circuits are the communication units to deliver data signals between cores and memories. The TSI T-line is made of aluminum, and the dielectric is silicon dioxide. The T-line length, width and characteristic impedance are $3mm$, $20\mu m$ and 50Ω , respectively. TSI modeled as transmission line (T-line) is shown in Figure 4.11.

The two I/O Interface circuits are designed in UMC $65nm$ CMOS process using 1P6M layers. Figure 4.12 illustrates the layout of LVDS and CML transceiver circuits. The layout consists of six microbumps distributed in the surrounding (the yellow octagons), which are used for microbumps fabricated and TSI connection between

Figure 4.13: (a) Simulation waveform of LVDS output eyediagram; (b) Waveform of LVDS passed through TSI connection

Figure 4.14: (a) Simulation waveform of CML output eyediagram; (b) Waveform of CML passed through TSI connection

Figure 4.15: EM simulation setup for TSV I/O

chips. The whole chip area is $550\mu m \times 330\mu m$, with about $60\mu m \times 120\mu m$ area for LVDS transceiver. In CML circuits, the inductors cost most of the chip area, which is $90\mu m \times 90\mu m$ for each.

Figure 4.13 and 4.14 shows the post layout simulation results of the LVDS and CML I/O buffers respectively. From Figure 4.13, it can be observed that the LVDS I/O interface can achieve 360mV peak-to-peak output swing and 563fs cycle-to-cycle jitter at 10GHz bandwidth. This waveform consists about 2dB amplitude pre-emphasis with the 0.28 ratio of current between emphasis and normal circuit. The power consumption is only 4.8mW under 1.2V supply, with about 0.48pJ/b energy efficiency. And in Figure 4.13(b), when the signals are transmitted through the TSI transmission lines between chips, the signal has 30mV loss in simulation. But the 2dB amplitude pre-emphasis part, which compensate the loss exactly, can protect the performance of high frequency edges. For the CML circuits, it can work under only 0.6V power supply, with 1.6mA total current consumption due to amplitude boosted inductors. As shown in Figure 4.14, it can provide 240mV peak-to-peak differential signal swing and 453fs jitter with 12.8Gb/s bandwidth, and 196mV peak-to-peak voltage after TSI T-line loss.

Figure 4.16: Insertion loss for $10\mu m$ diameter TSV I/O based on EM simulations

Figure 4.17: Layout of TSI I/O

4.3.3.1 EM Simulation of TSV and TSI I/O

EM simulations are carried out for 3D TSV and 2.5D TSI I/Os in Ansoft HFSS. Frequencies up to 20GHz are swept to measure the insertion loss for TSV and TSI I/Os. The substrate is lightly doped silicon with bulk conductivity of $5Siemens/m$, the bulk conductivity of copper TSV is set as $5.7 \times 10^7 Siemens/m$.

Simulation setup for TSVs is shown in Figure 4.15. Each TSV is of $100\mu m$ in height and $10\mu m$ diameter. The distance between two TSVs is set as $100\mu m$. From Figure 4.16, one can observe that with the increase in frequency the insertion loss increase. This is due to the increase in Ohmic loss with frequency. An insertion loss of 0.5dB is found for TSV at a frequency of 8GHz.

EM simulations are carried out for 3mm TSI I/Os, shown in Figure 4.17. The dielectric layer is polymer with effective permittivity of 3.5F/m and 0.02 loss tangents, and the bulk conductivity of transmission line is set as $5.7 \times 10^7 Siemens/m$. Similar to TSVs, with the increase in frequency, the insertion loss increases, due to the increase in Ohmic loss of the metal as well as the radiation loss into substrate. For example, at 8GHz, the insertion loss of TSI is found to be nearly 8dB. TSI has larger loss compared to the TSV due to its longer length and the conductive substrate.

Figure 4.18: Insertion loss for 3mm TSI I/O based on EM simulations

4.4 Summary

3D integration by TSVs and 2.5D integration by TSIs are the possible efficient many-core memory-logic integration techniques at large-scale. In 3D IC, signal TSVs utilized for inter-layer signal connections, varying nonlinearly with temperature and dummy TSVs are helpful in reducing temperature and stress gradients. A 3D clock-tree design with insertion of dummy TSVs using nonlinear optimization is presented in this chapter. A clock-skew reduction of nearly 61% is observed with the developed electrical-thermal-mechanical coupled delay model. The above clock-tree design is published in TCAD'13 with nonlinear optimization based dummy TSV insertion as the main contribution from the respondent author. To alleviate the transmission loss in TSI T-lines, special CML and LVDS I/O buffer designs are proposed. A large channel loss is observed in middle length TSI T-line compared to short distance TSV. The above mentioned I/O buffer designs are published in 3D IC conf.'13, with modeling of TSI T-line and TSVs as the major contribution from the respondent author.

Chapter 5

Power I/O Management

5.1 Introduction

In a 3D integrated many-core microprocessor with large amounts of memory [133, 101, 24, 29, 27, 28, 134, 135, 103, 5, 74, 136, 107, 36, 65] power management at system level is one of the challenges to be handled along with thermal dissipation. Thermal management in a 3D IC can be performed by insertion of dummy TSVs, as discussed in previous chapter. This thesis focuses more on the power management of many-core microprocessor. As previously mentioned, 3D integration can accommodate large number of cores in a single IC with large integration density. However, such a high density integration in 3D can introduce severe power and thermal issues, which may significantly affect the system performance and reliability. Improper power management will have two consequences: increase in temperature and power density. For a many-core microprocessor with DRAM, at high temperatures, leakage current will dominate the total power, which when coupled with temperature forms a positive-feedback and results in thermal runaway failure [60, 61]. Hence, from the device perspective, power models of core and DRAM with 3D TSV interconnects needs to be studied first. Additionally, to avoid such failures lower temperatures by cooling techniques or proper power management has to be performed, which will be discussed here.

In a many-core microprocessor the power demands of cores will be different and vary with time, providing uniform voltage-level can result in high power density. To avoid a dark-silicon dilemma for many-core microprocessors, effective dynamic-voltage-scaling (DVS) [62, 63, 64, 8] based power management has to be developed to provide cores with multi-level voltages at scale of hundreds or thousands of cores. As such, supplying multi-level supply voltages with maintenance of low power density has become an emerging issue to address [7, 4, 83, 49].

From physical hardware perspective, off-chip power converters may not be scalable for the surge of current demand by 3D many-core microprocessors due to long delivery latency, large delivery loss and severe delivery integrity [137]. On-chip power converters [82, 7, 63, 4, 83, 64, 84, 49, 85] are explored to provide prompt DVS power management with efficient power delivery. Since the chip area is quite limited for many-core microprocessors and the on-chip power converters may occupy considerable amount of area due to non-scalable inductor, one power converter per-core based design cannot be deployed for the power management of many-core microprocessors. As such, there is a dire need to develop a reusing scenario that can fully utilize the on-chip power converters. What is more, by integrating cores on one chip, the remaining area is quite limited for on-chip power converters with buck inductor. A single-inductor-multiple-output (SIMO) power converter [138, 139, 83] can be utilized to save area. One common single buck inductor is deployed to provide multi-level voltages in a time-multiplexed manner. The capability of SIMO converters, however, still has limited scalability for many-core microprocessors. The 3D integration introduces additional room for integration of on-chip power converters. The work in [49] has demonstrated the possibility to design on-chip power converters integrated with 64-tile network-on-chip in 3D/2.5D manner. As such, it is meaningful to explore 3D designs that can provide effective demand-supply matching for DVS power management of large-scale cores and converters. Optimization techniques to overcome the thermal reliability issues as discussed in previous chapters can be implemented in 3D integration. Additional advantage of 3D integration is its short interconnect lengths, which is the key for power delivery.

From the cyber management perspective [34, 140, 141, 142], power management for a many-core microprocessor will not be the same as that for single-core microprocessor, because there may exist multi-time-scale demands of supply voltages from different cores. Different power management schemes for many-core microprocessor are explored in [86, 82, 7, 143, 63, 4, 83, 64, 8, 49, 65, 144, 145], but the main challenge for a scalable DVS power management is still not resolved. In [63, 4, 8] voltage-frequency islands are utilized for power management of many-core microprocessors. However, as each core is statistically assigned with one fixed island, such a voltage/frequency assignment is not optimal for dynamic workloads. On the other hand, [7, 64] introduces the concept of a time-grained power management. Such a power converter per-core based power management may not be scalable for large number of microprocessors. The recent work in [145] utilizes a controlled switch network to connect a set of cores to a set of power converters with space multiplexing but ignores the possibility of time multiplexing. A space-time multiplexing based power management in [65] addresses a dynamic power management with reconfigurable switch network to provide a demand-

supply matching between many-core microprocessors and power converters. It has full flexibility in terms of I/O connections as well as the number of power converters.

There exists similarity between smart power management of many-core microprocessor and smart-grid though at different time-scale with different workload behaviors. Thereby, the study of workload behavior with classification and also demand-response method can be leveraged from the smart-grid management [146, 147] to deal with the large-scale on-chip demand-supply matching problem. In addition, workload balancing and peak-power reduction is also addressed in the proposed approach.

The proposed power management system is verified by the system-level behavior model implemented in SystemC-AMS for up to 64-core microprocessor. The physical design parameters are based on 130nm CMOS process with TSV models. In this chapter, core and DRAM power models with 3D TSV and 2.5D TSI I/Os are presented first. Further the possibility of thermal runaway failure in 3D and 2.5D integrated systems are investigated. On-chip power I/O management of 2-tier 3D many-core microprocessor by space-time multiplexing based techniques is discussed in the later part.

5.2 System Power Model

In a many-core memory-logic integrated system, the major contributors for the power are the core, memory and I/Os. We discussed about modeling of I/O power in the previous chapter. Here, we discuss the power models of core and DRAM memory.

5.2.1 Core and DRAM Power Model

Power dissipation of a microprocessor core P_{core_total} is the sum of its dynamic and leakage powers [148] and can be given as

$$P_{core_total} = \underbrace{\eta_a * C_{core} * V_{DD} * \Delta V * f}_{P_{core_dynamic}} + \underbrace{V_{DD} * I_{leakage}}_{P_{core_leakage}}; \quad (5.1)$$

$$I_{leakage} = \underbrace{A_s \cdot \frac{W_d}{L_d} \cdot v_T^2 \left(1 - e^{\frac{-V_{DS}}{v_T}}\right) \cdot e^{\frac{V_{GS} - V_{th}}{ns \cdot v_T}}}_{I_{subthreshold}} + \underbrace{W_d \cdot L_d \cdot A_J \left(\frac{T_{oxr}}{T_{ox}}\right)^{nt} \frac{V_g \cdot V_{aux}}{T_{ox}^2} e^{-B_J T_{ox}(a-b|V_{ox}|)(1+c|V_{ox}|)}}_{I_{gate_leakage}}; \quad (5.2)$$

where $v_T = k \cdot T / q$ is the thermal voltage; $P_{core_dynamic}$ is the core dynamic power; $P_{core_leakage}$ is the core leakage power; η_a represents the activity factor; C_{core} is the

core load capacitance; V_{DD} is the supply voltage with ΔV swing; f is the clock frequency; L_d and W_d are the effective device channel length and width respectively; ns is the subthreshold swing coefficient; A_s , A_J , B_J , a , b and c are the technology-dependent constants; nt is a fitting parameter; T_{ox} and T_{oxr} are the gate dielectric and reference oxide thickness respectively; and V_{aux} is the auxiliary temperature dependent function for density of tunneling carriers.

In addition, considering the DRAM as an array of n identical banks with each bank having a size M and B I/O channels per bank, the dynamic power of DRAM P_{DRAM} is

$$P_{DRAM} = n \cdot B \cdot C_{channel} \cdot V_{DD}^2 \cdot f \quad (5.3)$$

where $C_{channel}$ is the channel capacitance; $n \cdot B = N$ is the total number of channels; V_{DD} is the supply voltage; and f is the clock frequency.

Assuming that one DRAM memory bank with the size of M has the leakage current $I_{leakage}$, as given in (5.2), the total leakage power of DRAM P_{Dleak} can be given as

$$P_{Dleak} = n * M * V_{DD} * I_{leakage}. \quad (5.4)$$

It needs to be observed that the leakage current $I_{leakage}$ varies exponentially with temperature. This may have two consequences. Firstly, the leakage power can dominate the total power. Secondly, the leakage power may form a positive feedback loop with temperature, which can lead to *thermal runaway failure*. For example, in high-performance computing, with the increase in bandwidth $N = n \cdot B$ i.e., the number of banks n , the leakage power of DRAM also goes up. As such, the increase of bandwidth by adding more I/O channels may worsen the electrical-thermal coupling to power and increase the probability of thermal runaway. Therefore, one may need to design a high speed and low power I/O channel with each I/O channel supporting a large bandwidth.

5.2.2 System Power Breakdown

To evaluate the above mentioned power models and the thermal runaway failure, the gem5 [149] simulator is used. The multi-core system is set up as 16-core x86 microprocessors (1GHz) to operate in gem5, integrated with McPAT [148] to analyze the power of core, CACTI [150] for DRAM power and a modified thermal simulator [126] is employed to provide the thermal profiles for both 2.5D TSI and 3D TSV integrations. Each core has 32KB L1 instruction and data cache, respectively. From the McPAT and CACTI simulation results, the area estimations are obtained and presented in Table 5.1. The total area is about $288mm^2$ for 2.5D integration and $64mm^2$ for 3D integration.

Figure 5.1: Power breakdown of: (a) 2.5D integration at 25°C; (b) 2.5D integration at 120°C; (c) 3D integration at 25°C; (d) 3D integration at 120°C based on simulations in gem5

What is more, the parameters of TSV and TSV I/Os are also presented in Table 5.1. The number of channels for both 2.5D and 3D integrations is 64. The length of TSV I/O is 50 μm , which is the distance of adjacent layers; and the length of TSV I/O is 1.5mm, which is the distance between two ICs.

Table 5.1: System setup for data server system components

Components	Description	Value	Area Estimation
Core	Frequency	1.0 GHz	2.469 mm ²
	L1 cache size	32 KByte	
	Cache block size	64 Byte	
DRAM	Number of banks	4	32.025 mm ²
	Bank size	64 MByte	
	Page size	8192 bits	
	Bus width	128 bits	
TSV I/O	Number of channels	64	19 μm^2
	Length of interconnect	50 μm	
	Delay per channel	22.50ps	
TSI I/O	Number of channels	64	10 μm^2
	Length of interconnect	1.5mm	
	Delay per channel	34.52ps	

The system power breakdown (core, memory, I/O) pie chart is shown in Figure 5.1. Power consumption for 2.5D integration executing SPEC 2006 benchmark 401.bzip2 and Phoenix benchmark KMeans at 25°C and 120°C are shown in Figure 5.1(a) and 5.1(b) respectively. Power consumption for 3D integration executing SPEC2006 benchmark 401.bzip2 and Phoenix benchmark KMeans at 25°C and 120°C are shown in Figures 5.1(c) and 5.1(d) respectively.

From Figure 5.1, one can have the following observations. The power of DRAM is the dominant factor, especially at high temperatures. For example, in Figure 5.1(b), 2.5D integration at 120°C, the DRAM power accounts for 65.33%, while the core power

Figure 5.2: (a) Thermal power and electrical power cycles; (b) thermal runaway and threshold temperature

accounts for just 34.24%. It is because the DRAM leakage power will increase dramatically with the rise of temperature. Moreover, both power of TSI and TSV I/Os occupy a small portion of the pie chart. The TSI I/O power is less than 6% of the whole system power consumption in Figure 5.1(a) and (b), and is insensitive to the temperature. What is more, as shown in Figure 5.1(c) and (d), the TSV I/O power contributes less than 4% of the whole system power consumption. With the temperature rising, TSV I/O power percentage will also increase, due to the increase in capacitance.

5.2.2.1 Thermal Runaway Failure

The second kind of electrical-thermal coupling is between power and temperature. Under large bandwidth conditions, leakage power varying exponentially with the temperature dominates DRAM power. What is even worse, leakage power and temperature forms a positive feedback loop resulting in thermal run away failure.

From Figure 5.2(a) it needs to be observed that the thermal power $P_{thermal}$ is sampled at *ms*-scale based on electrical power at *ns*-scale acting as thermal source. Thermal power $P_{thermal}$ is given by

$$P_{thermal} = P_{core_dynamic} + P_{leakage} = P_{core_dynamic} + a_2 * e^{a_1 \cdot T}; \quad (5.5)$$

where $P_{core_dynamic}$ is the dynamic power of core, as in (5.1); and a_1 and a_2 are the coefficients to describe the exponential dependence between leakage power and temperature.

It can be observed from (5.5), thermal leakage power $P_{leakage}$ increases with temperature T . This increase in thermal power will form a positive feedback loop and increase temperature, resulting in thermal reliability concern and can be termed as thermal runaway, as the leakage power is sensitive to the temperature changing.

With each core serving as a thermal power source, interaction between g adjacent

Figure 5.3: Thermal runaway analysis of data server by 3D and 2.5D integrations from simulations

cores in neighboring grids result in a flow of heat flux through them, resulting in temperature increase with time. The system thermal dynamics with heat-sink is given by

$$C_{TR} \frac{dT}{dt} = P_{thermal} - \sum_{j=1}^g \frac{T - T_j}{R_j}; \quad (5.6)$$

where C_{TR} is the thermal capacitance of a core, and R_j is the thermal resistance path from g chips to the heat-sink.

If the thermal source grows much faster than the heat removal ability of heat-sink, temperature increases exponentially resulting in a thermal runaway failure. As shown in Figure 5.2(b), $T_{threshold}$ represents the maximum temperature beyond which thermal runaway failure happens. Placing heat-sink closer to processing cores can avoid thermal runaway failure. The 2.5D integration has a much close heat-removal path to heat-sink, the thermal-removal ability (R_j) has much better performance than 3D integration.

We evaluated thermal runaway failure of high performance data server by both 2.5D and 3D integration. The simulation is performed using both general purpose benchmarks and cloud-based benchmarks. Each cluster of cores execute different benchmarks. The number of memory layers for 3D integration is set flexible for the purpose of evaluation. The initial temperature is at the room temperature (25°C). The heat-sink size is set as $4.0\text{cm} \times 4.0\text{cm}$ in 2.5D integration and $2.0\text{cm} \times 2.0\text{cm}$ in 3D integration both with a heat-removal resistance as 4.6K/W .

The dynamic system temperature trend under 2.5D integration and 3D integration with different number of layers is shown in Figure 5.3. For 3D integration, we vary the number of layers to see its heat dissipation performance from 2 to 5, i.e., 3 layers means it has one logic layer and two memory layers. When the temperature goes beyond the

threshold temperature (100°C), the heat-sink cannot dissipate heat produced by the system, resulting in formation of a positive feedback with the leakage current and dramatic increase in system temperature leading to thermal runaway failure. Whereas, the temperature of 2.5D integration is maintained between 50°C and 70°C . In 3D integration, the temperature stays stable for less than 4 layers. With 5 layers in 3D integration, the temperature rising trends can be observed after 6 millions execution cycles, which rises quickly beyond 100°C when thermal runaway happens, for the current setup. As such, high performance data server formed by the 3D integration is more liable to the risk of thermal runaway issue, if proper cooling technique or power management is not applied. In this case, 4 layers setting is the best from thermal perspective.

5.3 3D Power I/O Management

To avoid the thermal runaway failure and unnecessary hotspots in the 3D IC power management can be carried out. As aforementioned, power management of many-core microprocessor using on-chip power converters is more effective. The DVS power management with the use of SIMO power converter is presented here. Based on a 3D re-configurable power switch network, space-time multiplexing (STM) based DVS power management is discussed for demand-supply matching between many-core microprocessors and multi-level on-chip power converters. The power switch network can be configured to perform space-time multiplexing between power converters and cores with connections by vertical through-silicon-vias (TSVs) in 3D, which can be formulated as two subproblems: resource allocation of power converters and workload scheduling. The objective of resource allocation is to achieve an optimal solution with the minimum number of power converters, while satisfying the constraints of both demands from different cores and hardware limitations from power converters. In order to solve the demand-supply matching problem for many-core microprocessors at a large-scale, integer-linear programming (ILP) and adaptive clustering of cores is deployed by learning and classifying power-signature pattern of workloads. In general, similar workloads will be distributed to a number of cores for parallel computation. Thus, those cores with similar workloads will show similar power-signature patterns and hence can be clustered together with a similar voltage-level. This is different from the single-core DVS that depends on the load current. What is more, power-signature of workloads with different patterns are initially classified by their magnitude levels as groups such that power converters are allocated to be shared in space between different groups, called *space-multiplexing*. In each group, power converters are further reused among different subgroups, formed based on their phases at different time instants, called *time-multiplexing*. Afterwards, the

workload scheduling can be performed in a demand-response fashion, where workload is the amount of task performed on a core in one time-slot. The workloads on each of the allocated power converters are measured with available slacks determined. Based on the available slacks, the workload scheduling is performed without violating the workload priorities resulting in workload balancing and peak-power reduction.

5.3.1 System Architecture

The 3D many-core microprocessor system architecture shown in Figure 5.4 is utilised for power management and basically composed of two tiers. The bottom tier is for power management, includes arrays of power converters and power switches. Each power converter is SIMO type, capable of supplying multi-level voltages by one buck inductor (See Figure 5.5). The top tier includes array of many-core microprocessors. In between these two array-structured tiers, there are through-silicon-vias (TSVs), controlled by power switches, to connect power converters and cores. Note that with the use of through-silicon interposers (TSIs), a 2.5D integration of multi-core microprocessor and on-chip power converters is demonstrated with silicon prototype in [49]. Moreover, there is one local super-capacitor for each core, working as local power storage to supply voltage when power converter is not available during multiplexing. The design and dimensions of TSVs are optimized for both speed of reconfigurability, the maximum driving current, and the thermal conductivity. One needs to note that 3D integration does not have to mean the traditional memory-logic integration. Similar to the recent work by IBM in [49], the 2.5D/3D integration can be utilized for on-chip power management efficiently as well as scalability for large number of many-core microprocessors. What is more, the 3D architecture with space-time multiplexing (STM) proposed here has further provided the flexibility and also improved the efficiency when utilizing power converters for many-core microprocessors. Similar to [64, 4, 8], we evaluate the performance by using data-domain specified benchmark set like SPEC2000, which contains both memory-bound and CPU-bound applications with predictable data-patterns as well as power traces. Additionally, benchmarks from embedded applications such as MPEG4, JPEG decoder etc., can also be used.

To perform DVS power management for large-scale many-core microprocessors, one can model it by a demand-supply system composed of the following three components:

- *Power Demand*: a set of cores C with demanded voltage-levels with set-size N_c . Each core c_i has a voltage-level demand of $v_d(c_i)$ to meet the deadline of its running workload and frequency. In addition, $v_a(c_i)$ is the allocated voltage-level to core c_i after power management and corresponding frequency.

Figure 5.4: 3D reconfigurable power switch network for demand-supply matching between on-chip multi-output power converters and many-core microprocessors

- *Power Supply*: a set of power converters R with set-size N_r . Each power converter outputs the voltage-level $v(r_i), v(r_i) \in V$ to supply the cores, where V is the set of available voltage-levels before power management.
- *Power Switch Network*: a set of reconfigurable switch-boxes SW with set-size N_s to connect between R and C for demand-supply matching.

The power management circuit for DVS is shown in Figure 5.5. Initially, the voltage and current sensors sample voltage and current values from the cores as power profile. By tracking power profiles of cores, the demanded voltage-levels of cores for the next period of control can be tracked, which is 400ns, in our simulations. The value for next period of control is predicted based on the pre-stored training look-up-table. The data analytic of workloads can be performed to configure STM by learning and classifying power-signature patterns of workloads. Next, the DVS power management unit decides the optimal STM configuration that can match the demand of cores with supply from the minimum number of power converters. Figure 5.6 shows how to perform STM by on-chip SIMO power converters [138] utilizing single inductor to provide multiple voltage-levels. The objective is to design SIMO power converters configured to satisfy the demand from cores. Based on the configuration of switches $(S_1, \dots, S_{N_g})$, corresponding voltage-level will be generated at the outputs of power converters to each group, divided in space, i.e., space-multiplexing. Power converters allocated to one group can be further reused by cores among subgroups divided in time, i.e., time-multiplexing. As

Figure 5.5: Functional units of space-time multiplexing based power management for DVS with SIMO power converter

such, one power converter is reused maximally to connect with one core at one allocated space-slot and time-slot.

Note that in the traditional island-based [151] and SIMO-based [138, 139, 83] power management approaches, the connections between power converters and cores are assumed to be fixed, which is not feasible to provide the matched supply to many cores at large-scale. The power management technique discussed in next few sections is scalable for a large number of cores but with the assumption that power management block including controller, switches, power converters and power delivery network are all in one layer different from the layer of cores. As such, it may require a different physical design when routing power/ground networks to deliver to the layer of cores. Moreover, packaging large number of power converters as well as cores may increase power density and hence a thorough cooling design is required for thermal reliability concern.

5.3.2 Problem Formulation

As aforementioned in the beginning of this chapter, the primary challenge here is to solve a large-scale DVS power management with matched demand-supply. Though there exists various workloads with different power-signature patterns, most of them can be classified by magnitude and phase when similar workloads are distributed to different cores. As such, if one can perform clustering of cores by learning and classifying power-signature patterns of distributed workloads, the complexity for demand-supply matched DVS power management can be accordingly reduced.

Figure 5.6: Space and time multiplexing with on-chip SIMO power converters

With the further consideration for the minimum number of power converters, one can formulate a resource (power converter) allocation subproblem as follows.

Subproblem 1: Resource allocation problem is to decide the minimum number of power converters such that demands from cores can be satisfied.

What is more, there may exist power slacks that can be utilized without violating the workload execution priority or deadline. One can vary/shift workloads on over-loaded power converter at one time-slot to other time-slot with under-loaded power converters in a demand-response fashion. As such, the peak-power can be reduced as well as workload can be balanced at power converters, which can be formulated as the second subproblem below after the first subproblem is done.

Subproblem 2: Workload scheduling problem is to delay over-loaded workloads to under-loaded time-slots based on availability of slack and without violation of priority.

To solve the above mentioned demand-supply matching by space-time multiplexing, an integer linear programming (ILP) based optimization for small-scale and adaptive clustering for large-scale problems are utilized. Before discussing more details of the

algorithm, we redefine resource allocation problem as follows:

Resource allocation for STM: there are N_r power converters shared spatially among N_c cores, connected by N_s reconfigurable power switches with each power converter capable of switching among N_v different voltage-levels at a fixed time-slot H to supply multiple voltage-levels simultaneously.

5.4 ILP based Management

The space-time multiplexing (STM) problem was solved by an integer linear programming (ILP) in [65] with reformulated problem as below.

ILP Optimization of STM: there are N_r power converters shared spatially among N_c cores, connected by N_s reconfigurable power switches with each power converter capable of switching among N_v different voltage-levels at a fixed time-slot H to supply multiple voltage-levels simultaneously.

To perform ILP, first the constraints to be satisfied has to be defined. Due to physical hardware limitations, a power converter can be connected to a core only if the following constraints are met: (i) the maximal power converter inductance current does not exceed its maximal value, I_L ; and (ii) the maximal core voltage-drop is within a specified value ΔV during multiplexing. Thus, one can formulate the following ILP optimization to determine the STM configuration.

One can have the following reformulation in the form of linear equations as

$$\begin{aligned}
\min: & \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \sum_{v=1}^{N_v} v_v \cdot x_{ij}^v \\
\text{s.t.}: & \text{(i)} \sum_{j=1}^{N_r} \sum_{v=1}^{N_v} x_{ij}^v = 1, \forall 1 \leq i \leq N_c \\
& \text{(ii)} \sum_{j=1}^{N_r} \sum_{v=1}^{N_v} v_v \cdot x_{ij}^v \geq v_d(c_i), \forall 1 \leq i \leq N_c \\
& \text{(iii)} \sum_{i=1}^{N_c} i_v \cdot x_{ij}^v \leq I_L, \forall 1 \leq j \leq N_r, 1 \leq v \leq N_v \\
& \text{(iv)} \sum_{i=1}^{N_c} \sum_{v=1}^{N_v} x_{ij}^v \leq 1 + \frac{\Delta V \cdot C_L}{I_{max} H}, \forall 1 \leq j \leq N_r \\
& \text{(v)} N_{min} \leq \sum_{i=1}^{N_c} \sum_{v=1}^{N_v} x_{ij}^v \leq N_{max}, \forall 1 \leq j \leq N_r.
\end{aligned} \tag{5.7}$$

Therefore, the STM problem is now simplified to minimize (5.7) with the corresponding constraints being satisfied. This implies that the total voltage-levels allocated

to cores are minimized. The constraints defined in (5.7) implies: (i) each core is connected to at most one power converter at a particular time-slot; (ii) the allocated voltage-level must satisfy the demand of the core; (iii) the maximal inductance current does not exceed its maximal value I_L ; (iv) the maximal core voltage-drop at any time instant H for a core of capacitance C does not exceed ΔV ; and (v) each power converter has minimum and maximum limits on the number of cores it could connect. The first two constraints ensure that adequate voltage is supplied to each core, the next two constraints ensure that inductance current is not exceeded its limits and core voltage drop is minimal and the last constraint ensures that no power converter will be under or over utilized.

In (5.7), the boolean variable x_{ij}^v equals 1 if and only if the core $c_i \in C$ is supplied by power converter $r_j \in R$ with the voltage-level $v_v \in V$, as explained in (5.8).

$$x_{ij}^v = \begin{cases} 1 & c_i \text{ supplied by } r_j \text{ at voltage-level } v_v \\ 0 & \text{otherwise} \end{cases}. \quad (5.8)$$

Solution to Subproblem 1 The reformulated STM problem can be solved by linear program *lp_solve* [152] deployed on one of the microprocessor cores with typical solving time ranging from microseconds [65], which is faster when compared to off-chip converters based DVS management in the scale of seconds. Resource allocation can be performed using the above mentioned ILP optimization, however the runtime increases exponentially with number of cores. In the following, a power-signature learning based adaptive clustering is proposed to address the scalability problem for DVS power management of many-core microprocessors.

5.5 Space-Time Multiplexing based Management

As the number of cores increases, the complexity of solving ILP as well as the runtime increases. As such, scalable solution is required for the resource allocation when dealing with a many-core system. To perform resource allocation for large-scale system in less time, learning and classifying of workloads power-signature pattern are employed to cluster cores by adaptive clustering developed as discussed below. Solution for demand-supply matching with less number of power converters i.e., subproblem 1 is discussed first followed by peak-power reduction and workload balancing by demand-response based workload scheduling. The main assumption here is based on the observation that similar workloads will be distributed to a number of cores for thread-level parallelism. As a result, they will have similar power-signature patterns and can be clustered together with the same voltage-level.

Figure 5.7: Power-signature pattern extracted in each time-slot of control-cycle

5.5.1 Adaptive Clustering

The adaptive clustering of cores is done by learning similarity of power-signature patterns of workloads. The high-precision power profiles may not be necessary for clustering and hence the first step of learning power-signature is to have a power-signature extract by envelope.

5.5.1.1 Power-signature Extraction

Before proposing solutions, a few definitions are presented below.

Control-cycle: amount of time required to finish all the workloads in a group.

Time-slot: amount of time required to finish all the workloads in a subgroup.

Since it is impractical to perform power management with the use of continuous data of power profiles, one needs to extract power-signature from power profiles. In the following, procedure to obtain power-signature pattern of one workload in one control-cycle by extracting the *envelope* from the power profile is presented. Based on the extracted peak-power envelope, or *power-signature*, one can build workload behavior model to be utilized in the following resource allocation as well as workload scheduling.

Assume that in a control-cycle T^i for i^{th} group having N_k subgroups, each core is assigned with one workload. Relation between control-cycle T^i and time-slot T_j^i is

$$T^i = \sum_{j=1}^{N_k} T_j^i. \quad (5.9)$$

As such, in one time-slot T_j^i , power-signature (peak-power envelope) P_{s_z} is extracted for workload $p_z(t)$ from core c_z , $c_z \in C$ of one subgroup by

$$P_{s_z}(T_j^i) = \max(p_z(t)). \quad (5.10)$$

Figure 5.8: Extraction of power-signature for workloads based on (5.10)

This is repeated for whole control-cycle T^i . Thus, the peaks are extracted and a peak envelope is formed. Power-signature extraction by forming peak-power envelope is shown in Figure 5.7. Power-signature is indicated by the blue line. In the example shown in Figure 5.7, control-cycle is comprised of 3 time-slots, indicated by the red dotted line. Power-signature for core c_i with power profile p_i is denoted by P_{s_i} ($P_{s_i} = \max(p_i)$). Based on the learning of power signatures of workloads, classification of cores can be performed in two steps namely grouping in space and subgrouping in time.

In order to allocate the voltage-level, load power has to be tracked and predicted. Prediction of power level is performed using the auto-regression (AR) algorithm [153]. At sampling interval t , based on the previous recorded load power values $p_i(t), p_i(t-1), p_i(t-2), \dots, p_i(t-M)$ the transient power $p_i(t+1)$ needed for the next time instant can be predicted by

$$p_i(t+1) = \sum_{j=0}^M a_j p_i(t-j) + \varepsilon \quad (5.11)$$

where a_j is the auto-regression coefficient, ε is the prediction error and M is the order of the prediction model. AR coefficients can be calculated based on the least-squares method. Prediction is performed in every control-cycle and accordingly the predicted power is calculated and corresponding voltage-level is allocated. This makes the power converters allocation in a runtime fashion. It is ideally assumed that the driving capability of a power converter is independent of its voltage-level.

Figure 5.8 shows an example of the extracted power-signatures. The extracted power-signature is shown by red color line with control-cycle of 400ns and time-slot of 100ns. In two control cycles, there are total 8 power envelopes extracted for 8 time-slots.

To decide the demanded voltage-level under space-time multiplexing power management, the power-signature needs to be tracked and predicted. Here, power-signature tracking and prediction is performed based on (5.11). The order of prediction M is set as

Figure 5.9: Runtime power tracking and prediction for benchmark *gcc*: (a) power prediction using (5.11); (b) voltage-level transition

8 to guarantee the precision, and has a prediction error of 0.3% on average for SPEC2000 benchmarks. Based on the predicted power, the required voltage-level is looked up in the LUT. Figure 5.9 shows the power tracking and prediction of the core under benchmark *gcc*. As seen from Figure 5.9(a), predicted power (denoted by red line) using AR closely matches the actual power demand (denoted by blue line). Since the power consumption is in proportion to the supplied voltage-level, larger power consumption demands for higher voltage-level. Based on the predicted power-signature values and power-voltage pairs in the look-up-table, the supply voltage-levels to be allocated to cores, shown in Figure 5.9(b). For example, when the predicted power-signature magnitude is between 0.17W and 0.19W, a voltage-level of 0.8V needs to be supplied.

5.5.1.2 Space Multiplexing: Grouping

The cores with similar power-signature patterns in terms of magnitudes are grouped into one. For example, z -th group g_z , $g_z \in G$, of cores can be formed by the following criteria

$$g_z = \{c_i; v_d(c_i) = v_d(c_j) = v_z, \forall i, j = 1, \dots, N_c, z \leq N_v\}. \quad (5.12)$$

Here, v_z , $v_z \in V$ represents the voltage-level; and $v_d(c_i)$, $v_d(c_i) \in V$ represents voltage-level demand of core c_i , $c_i \in C$.

Based on the power-signature magnitude levels, different groups are formed. Each

group may contain different number of cores which have similar power-signature magnitudes, but may differ in power-signature phase. The number of cores in a group can change at different control-cycles. Note that grouping process is based on the comparison of levels and hence has less complexity of computation.

5.5.1.3 Time Multiplexing: Subgrouping

For cores in the same group with similar magnitude levels, they can be further classified based on the power-signature phase, i.e., execution behavior with time. The study of power-signature phases cannot be simply classified based on power-signature magnitudes. Considering a set of power-signature patterns, subgroup k_s , $k_s \in K$, can be formed by the following criteria

$$k_s = \{c_i; (v_d(c_i) = v_d(c_j) = v_z) \& (P_{s_i} \sim P_{s_j}), \forall i, j = 1, \dots, N_c\}. \quad (5.13)$$

Here, P_{s_i} represents power-signature of core c_i , $c_i \in C$ in one time-slot; $v_d(c_i)$, $v_d(c_i) \in V$ represents the demanded voltage-level of core c_i ; and v_z , $v_z \in V$ represents the voltage-level allocated to group g_z .

To form subgroups based on the power-signature phases, similarity between power-signature phases can be exploited. To find similarity between phases of power-signatures P_{s_i} and P_{s_j} in a group having between N power-signatures in one control-cycle, correlation in term of covariance matrix has to be evaluated as

$$X = \frac{1}{N} \sum_{i,j=1}^N (P_{s_i} - \overline{P_s})(P_{s_j} - \overline{P_s})^T \quad (5.14)$$

where $\overline{P_s}$ is the mean of all power-signatures ($\frac{1}{N} \sum_{i=1}^N (P_{s_i})$).

Based on the order of covariance matrix X , the number of subgroups N_k can be analyzed by the singular-value-decomposition (SVD) of X as

$$X = \mathbf{U} \times \mathbf{S} \times \mathbf{V}^{-1} \quad (5.15)$$

where

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,N} \end{pmatrix}; \quad \mathbf{U} = \begin{pmatrix} \mathbf{u}_{1,1} & \mathbf{u}_{1,2} & \cdots & \mathbf{u}_{1,N} \\ \mathbf{u}_{2,1} & \mathbf{u}_{2,2} & \cdots & \mathbf{u}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{N,1} & \mathbf{u}_{N,2} & \cdots & \mathbf{u}_{N,N} \end{pmatrix};$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_{1,1} & 0 & \cdots & 0 \\ 0 & \mathbf{s}_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{s}_{N,N} \end{pmatrix}; \quad \mathbf{V} = \begin{pmatrix} \mathbf{v}_{1,1} & \mathbf{v}_{1,2} & \cdots & \mathbf{v}_{1,N} \\ \mathbf{v}_{2,1} & \mathbf{v}_{2,2} & \cdots & \mathbf{v}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_{N,1} & \mathbf{v}_{N,2} & \cdots & \mathbf{v}_{N,N} \end{pmatrix}.$$

Matrices $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices with $\mathbf{S}$ as the diagonal matrix. One needs to note that SVD-based workload characterization is deployed in off-line learning of workload data and look-up-table (LUT) built online. As such, the runtime can be efficient for large-scale problem and is reported in Table 5.3.

Based on the rank analysis of $\mathbf{S}$, the number of subgroups N_k is decided. A new matrix can be formed with N_k independent vectors, extracted from either of the orthogonal matrices. Let the newly formed matrix be $\mathbf{V}_k$, assuming it is extracted from $\mathbf{V}$. The product of $\mathbf{V}_k$ with the covariance matrix X will result in a reduced matrix X_k , which forms basis of the clustering for subgrouping.

$$X_k = X \times \mathbf{V}_k \quad (5.16)$$

where

$$X_k = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N_k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,N_k} \end{pmatrix}; \quad \mathbf{V}_k = \begin{pmatrix} \mathbf{v}_{1,1} & \mathbf{v}_{1,2} & \cdots & \mathbf{v}_{1,N_k} \\ \mathbf{v}_{2,1} & \mathbf{v}_{2,2} & \cdots & \mathbf{v}_{2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_{N,1} & \mathbf{v}_{N,2} & \cdots & \mathbf{v}_{N,N_k} \end{pmatrix}. \quad (5.17)$$

Based on the reduced matrix, the subgrouping can be performed by selecting the maximum value in each column and assigning it to corresponding subgroup.

The control time reported in Table 5.3 is for the total power management, which includes not only the converter switching time but also the time for performing workload characterization. Please note that there are two parts in workload characterization. The first part is the off-line SVD-based learning of workload data, which may consume a long time. The second part is the on-line look-up-table based clustering and prediction of workload data, which can be accomplished within a few nanoseconds. As such, the runtime can be efficient for a large-scale problem.

5.5.1.4 Allocation of Power Converters

Once the groups and subgroups based on power-signature pattern learning are formed, the maximum workloads of one subgroup can be determined. As such, the minimum

Algorithm 1 Adaptive clustering based space-time multiplexing

INPUT: Power profile matrix P with power profile vectors p_i

1. In one control-cycle, extract power-signatures P_{s_i} from power profile vectors $P_{s_i} = \max(p_i)$
2. Perform grouping of power-signatures by magnitude $g_z = \{c_i; |P_{s_i}| = v_z\}$
3. For each group g_z , compute the covariance matrix $X \in X^{N \times N}$
4. Perform SVD: $X = \mathbf{U} \times \mathbf{S} \times \mathbf{V}^{-1}$
5. Determine number of subgroups: $N_k = \text{rank}(\mathbf{S})$
6. Compute the first N_k singular-value vectors $v_1, \dots, v_{N_k}$ of $\mathbf{V}$
7. Let $\mathbf{V}_k = [v_1, \dots, v_{N_k}] \in R^{N \times N_k}$ and $X_k = X \times \mathbf{V}_k$
8. Add i -th core to j -th subgroup if $X_k(i, j)$ is maximum in the i -th row
9. Form P_k matrices within the group by finding corresponding indices in power profile matrix P
10. Perform the same subgrouping process for all N_g groups

OUTPUT: New clustered matrices $P_{z,k}$ ($z = 1, \dots, N_g; k = 1, \dots, N_k$)

number of power converters are determined to supply to that subgroup of cores. This results in a feasible solution to solve subproblem 1, which can be rephrased as below.

$$\begin{aligned} \min: & \sum_{j=1}^{N_g} r_j \\ \text{s.t.:} & \quad (\text{i}) \ v_a(c_i) \geq v_d(c_i), \forall c_i \in C \\ & \quad (\text{ii}) \ d(r_j) \leq N_{\max}, \forall r_j \in R. \end{aligned} \tag{5.18}$$

If one can determine the minimum number of power converters r_j for each group, the total number of power converters for N_g groups can be correspondingly minimized. Note that constraint (i) guarantees that the supplied voltage-level $v_a(c_i)$, $v_a(c_i) \in V$ from power converter will satisfy the demanded voltage-level $v_d(c_i)$, $v_d(c_i) \in V$ from core c_i , $c_i \in C$. Moreover, constraint (ii) imposes the driving ability $d(r_j)$ of each power converter is $N_{\max}$, i.e., the maximum number of cores to drive. The driving ability varies with the supplied voltage-level: the higher the voltage-level is, the lower the number of cores that one power converter could drive. With the increase in supplied voltage-level by a power converter, the load current increases thereby reducing its driving capability.

Next, the minimization in required total number of power converters which can be solved by grouping and subgrouping is discussed. By grouping, power converters can be shared in space among N_g number of groups and subgrouping makes sharing of power converters inside one group in time. Based on the driving ability d^j of power converters in group g_j , $g_j \in G$, and maximum number of cores among different subgroups $\max(c_j)$, the maximum number of power converters allocated can be determined as

$$r_{g_j} = \max(c_j) / d^j. \tag{5.19}$$

Figure 5.10: Resource allocation by adaptive clustering: grouping by power-signature magnitude and subgrouping by power-signature phase

As such for the whole system, the total number of power converters needed will be $\sum_{j=1}^{N_g}(r_{g_j})$, which is the minimum number to satisfy the demand-supply matching. Therefore, the proposed learning of large number of power-signatures of workloads can classify cores into group and subgroup to allocate the power converters, and hence reduce the complexity for large-scale problem in contrast to the previously developed method by ILP. In summary, the large-scale demand-supply matching can be efficiently solved by the above-mentioned two-step clustering in every control-cycle. The obtained r_{g_j} represents the minimum number of power converters to satisfy demand-supply matching i.e., solution to subproblem 1.

The procedure for the space-time multiplexing is further summarized in Algorithm 1. As one example, the formulation of groups and subgroups of cores by learning of power-signature pattern with the reuse of on-chip SIMO power converters in space and time is illustrated in Figure 5.10 and described below.

1. In the first control-cycle, cores with power-signatures: $P_{s1}, P_{s2}, P_{s3}, P_{s4}$ and P_{s5} are at one power magnitude level and other cores are at a different power magnitude level. As such, one can assign them to two groups with voltage-levels v_1 for group 1 and v_2 for group 2 ($v_1 > v_2$). Thus, power converters can be shared in space. We assume the driving ability of power converter supplying voltage-levels v_1, v_2 to be 1 and 2, respectively.
2. In group 2, based on power-signature phase similarity, cores with power-signatures

P_{s1}, P_{s2}, P_{s3} are clustered to form subgroup 1 and cores with power-signatures P_{s4}, P_{s5} are clustered to form subgroup 2. To satisfy the demand of a group, subgroup with more number of cores is selected i.e., subgroup 1 having 3 cores. The corresponding number of power converters required is calculated based on (5.19) and 2 power converters (r_1, r_2) allocated to group 2 under voltage-level of v_1 and driving ability of 2.

3. In the first time-slot, power converters are connected to cores in subgroup 1 and in the next time-slot, power converters are connected to cores in subgroup 2. Thus, power converters are shared in time by time-multiplexing.
4. Similarly, for group 1, cores with power-signatures P_{s6} and P_{s7} are clustered to form subgroup 1; and core with power-signature P_{s8} is allocated to subgroup 2 due to different power-signature phase. Considering subgroup 1 with 2 cores, the number of power converters is calculated and 2 power converters (r_3, r_4) are allocated under voltage-level of v_1 and driving ability of 1.
5. In the next control-cycle, due to change in power-signature magnitude and phase, core with power-signature P_{s5} is allocated to group 1 and other cores remain in same group. Hence in group 2, subgroups 1 and 2 both will have 2 cores each. As such, 1 power converter supplying voltage-level v_2 can satisfy the demand.
6. Whereas, in group 1, cores with power-signatures P_{s6}, P_{s7} and P_{s8} form subgroup 1 due to similarity in power-signature phase; and core with power-signature P_{s5} is allocated to subgroup 2 due to difference in phase.
7. As such, in group 1, subgroup 1 will have 3 cores and subgroup 2 will have 1 core. To satisfy the demands, subgroup 1 having 3 cores is considered and number of power converters needed will be 3. Instead of adding a new power converter, power converter r_2 (previously allocated to group 2) can be used group 1 to provide voltage-level v_1 .

ILP vs Adaptive Clustering ILP is a P-hard problem to solve and the time required to solve ILP increases exponentially with the number of cores. On the other hand, for adaptive clustering based power management, the space multiplexing has a complexity of $O(M)$, where M represents the number of power demands and $O(N^2)$ for SVD of a $N \times N$ matrix. Hence, the total complexity is $O(M + N^2)$, which is much smaller than the ILP based solution.

The proposed system is validated by Matlab and system-level models built from SystemC-AMS. Table 5.2 summarizes the system design specifications. All units are

Table 5.2: System settings of 3D many-core microprocessors, on-chip power converters, TSVs and power switches

Item	Description	Symbol	Value	Size
Microprocessor	Performance	N.A.	410 DMIPS	$1.5mm^2$
	Frequency	f_c	250MHz	
	Power Consumption	P_c	0.4W	
Power Converter	Input Voltage	V_{in}	2.4V	$1.6mm^2$
	Output Voltage	V_{out}	0.6V, 0.8V, 1.0V, 1.2V	
	Load Current	I_L	120mA, 150mA, 220mA, 350mA	
	Number of Phases	N.A.	2	
	Inductor per Phase	L	1nH	
	Switching Frequency	f_s	50-200MHz	
	Peak Efficiency	N.A.	77%	
TSV	Length	l	$25\mu m$	$455\mu m^2$
	Diameter	W	$5\mu m$	
	Isolation Film	r	120nm	
	Resistance	R_{TSV}	20m Ω	
	Capacitance	C_{TSV}	37 fF	
Power Switch	Width	w_s	4mm	$520\mu m^2$
	Length	len	130nm	
	Switching Time	N.A.	300ns	

modeled at CMOS 130nm process. The specification of low-power MIPS microprocessor [154] is taken as the core model. Each core has a nominal frequency of 250MHz with the maximal power consumption of 0.4W. Benchmarks from SPEC2000 [128] are simulated by Wattch [155] simulator to generate power profiles. The extracted power-signatures from power profiles are used as workload models. The typical control-cycle for power management is set to 400ns.

A 2-phase multi-output power converter [84] is designed to generate four different voltage-levels. As driving ability of a power converter depends on supply voltage-level, driving abilities are set as 4, 3, 2, 1 for voltage-levels 0.6V, 0.8V, 1.0V and 1.2V respectively. Look-up-table for power and voltage-levels is set as (<0.17W, 630MHz, 0.6V), (0.17W-0.19W, 680MHz, 0.8V), (0.19-0.21W, 730MHz, 1.0V) and (>0.21W, 780MHz, 1.2V). Moreover, the inductance value in power converter is 1nH per phase to support the maximum current on the buck inductor. Such an inductor requires an area of $0.25mm^2$, occupying 30% area of the power converter. The local super-capacitor for each core is set as $1\mu F$ (max value) to support time-multiplexing scheme between subgroups. The value of local capacitor needs to be selected based on targeted application and available on-chip space. The design of on-chip power converter thereby needs to consider the limitation of inductor and capacitor area, which are placed both in 3D integration and hence the minimum area overhead to core area.

In addition, the vertical TSV [135] with the size of $455\mu m^2$ works as connections

Table 5.3: Comparison of average power consumption and controller runtime for STM by ILP and STM by adaptive clustering

Number of Cores	Benchmarks	Power per Core (mW)			Power Saving (%)		Controller Runtime (ms)	
		ILP	Adaptive Clustering	Non-DVS	ILP	Adaptive Clustering	ILP	Adaptive Clustering
4	Set 1: art, eon, lucas, wupwise	279.50	248.00	393.71	29.01%	37.00%	7.30	7.73
	Set 2: apsi, gcc, gzip, mcf	168.32	195.69	349.34	51.82%	43.98%	9.50	10.73
	Set 3: facerec, galgel, twolf, crafty	224.95	233.32	366.14	38.56%	36.27%	7.20	6.42
	Set 4: vortex, parser, mgrid, sixtrack	240.06	237.84	385.85	37.78%	38.36%	10.70	10.30
8	Set 1 + Set 2	223.17	221.85	371.53	39.93%	40.29%	25.00	20.68
	Set 1 + Set 3	252.24	240.66	379.93	33.61%	36.65%	27.10	20.17
	Set 1 + Set 4	260.04	242.92	389.78	33.29%	37.68%	37.00	24.74
	Set 2 + Set 3	195.34	196.64	357.74	45.40%	45.03%	21.70	20.31
	Set 2 + Set 4	202.65	216.77	367.60	44.87%	41.03%	30.40	25.70
	Set 3 + Set 4	231.71	235.78	376.00	38.38%	37.29%	29.80	25.23
16	All Sets	309.38	277.25	373.36	17.22%	25.74%	50.80	47.32
32	All Sets	319.93	290.63	374.14	14.49%	22.32%	336.30	97.35
64	All Sets	284.56	220.44	387.04	26.48%	43.04%	187500.00	120.29
Average		245.53	235.22	374.81	34.68%	40.38%	N.A.	N.A.

between cores and power converters. According to the model in [103], it has a DC-resistance of $20m\Omega$. Considering the maximum current of $330mA$, the IR-drop of TSV is around $7mV$, which is quite small. Note that the capacitor of TSVs is in fF scale and hence does not influence the load capacitance. What is more, for each TSV channel, one switch box is assigned with N_r power switches to support the core-converter connection. The switch box offers a compact reconfigurable unit driven by the controller. The power switch inside each switch box occupies $520\mu m^2$ and is able to deliver the maximum core current with switching time of $300ns$. As such, the TSV coupling is also quite small to be considered under such a low-activity switching.

Runtime and Power Saving Comparison Experimental results using different sets of benchmarks from SPEC2000 [128] on different number of cores are presented in Table 5.3. Performance comparison is made for the non-STM method, ILP-based STM method and the adaptive clustering based STM method. Firstly, we compare the power saving of ILP and adaptive clustering when compared to the non-STM power management. Considering benchmarks in set 1 and set 2, there is a power saving of 29.01% and 51.82% by the ILP and 37.00% and 43.98% by the adaptive clustering, respectively. The power saving depends on the workload deployed on the core. On average, the respective power savings are 34.68% and 40.38% with the ILP and the adaptive clustering based power management compared to the non-STM power management. What is more, the runtime comparison between the ILP method and the adaptive clustering method is observed as well. When applying benchmarks in set 1 and set 2 on an 8-core system, the runtime of ILP and adaptive clustering is $25ms$ and $20.68ms$ respectively. When on a 32-core system, the runtime of ILP becomes $336.30ms$ whereas adaptive clustering takes $97.35ms$, which is nearly linearly increased with cores. Furthermore, when the

number of cores is increased to 64, the runtime of ILP is nearly 1000 times slower than the proposed adaptive clustering based power management. Please note that the control time report in Table 5.3 is for the total power management, which includes off-line SVD-based learning of workload data and the converter switching time along with prediction and allocation of voltage-levels using LUT. The power converter switching time and LUT based prediction consumes few *ns*.

Adaptive Clustering Adaptive clustering of cores by learning similarity of power-signature patterns of workloads is described. In the previous ILP based resource allocation, complexity lies in searching the large solution space to satisfy the constraints involved, whereas in adaptive clustering groups and subgroups are formed resulting in smaller search space and reduced complexity and less runtime.

Adaptive clustering performed for 32-core and 64-core microprocessors and corresponding results are presented here. Input power traces are first divided into 4 groups based on their power magnitudes, then in each group, subgroups are formed based on their power phases. Based on the extracted power-signatures, cores with similar power-signature magnitudes are grouped and provided with the same voltage-levels. Power converters are allocated at different space (voltage-levels) for space-multiplexing. Inside each group, cores with similar power-signature phases are further subgrouped. Power converters are further allocated in different time for time-multiplexing. Based on the maximal number of cores among subgroups of a group, the number of power converters for each group are determined and allocated.

Figure 5.11 illustrates the adaptive clustering result for 32-cores at two consecutive control-cycles. Different filling patterns represent different groups or voltage-levels. Numbers on the downright-corner of cores represent subgroups. For example, in the first control-cycle, 12-th core is assigned to subgroup 4 with voltage-level 1 (group 1). But in the next control-cycle, it is assigned to subgroup 1 of same voltage-level. Similarly, in the first control-cycle, 7-th core is assigned to subgroup 3 with voltage-level 1 (group 1) supplied by a voltage-level of 0.6V. Whereas in the next control-cycle, it is assigned to subgroup 2 with voltage-level 3 (group 3), supplied by a voltage-level of 1.0V. Thus, the voltage transition takes place with the aid of STM by adaptive clustering.

For 64-core case, Table 5.4 summarizes the clustering results. The numbers in the table represent the core IDs. The runtime of whole process is small and nearly 120ms. One can observe that different groups and subgroups have different number of cores allocated and even some of the subgroups are left without any core indicating different power-signatures may have similar phase. For example, group 1 has been allocated with 14 cores, whereas group 2 has 11 cores; inside which subgroup 4 is empty but

Figure 5.11: Results of adaptive clustering for a 32-core microprocessor in two consecutive control-cycles

other subgroups of group are allocated with cores. This unoccupied subgroups indicates idleness of the power converter i.e., availability of slack. Considering group 1 in Table 5.4, the maximum number of cores in a subgroup among different subgroups is 6 and power converter driving ability 4, hence 2 power converters supplying a voltage-level of 0.6V is sufficient to drive cores in this group.

Table 5.4: Adaptive clustering result for 64-core microprocessor

	Subgroup 1	Subgroup 2	Subgroup 3	Subgroup 4
Group 1	31, 37, 52 58, 59, 63	33,43	7, 8, 14	12, 49, 54
Group 2	17, 40, 41, 50 51, 56, 62	22, 42	27, 29	N/A
Group 3	6, 21, 32 36, 39, 46 47, 64	9, 15, 16 20, 26 ,28 35, 53, 55	1, 5 11, 18 19, 38	N/A
Group 4	2, 3, 23, 25 34, 44, 45, 48 57, 60, 61	10, 13	N/A	4, 24, 30

Next, the STM by adaptive clustering can lead to the reduction in number of power converters needed for demand-supply matching. When comparing to two schemes,

namely space-multiplexing (SM) and time-multiplexing (TM), STM takes advantage of both space-multiplexing (SM) and time-multiplexing (TM) with consideration of its driving ability. Table 5.5 shows the comparison of number of power converters needed for a 32-core and 64-core cases with the SM and TM schemes. One can observe a reduction of 55.00% (SM) and 35.71% (TM) in number of power converters for a 32-core microprocessor, while 41.67% (SM) and 36.36% (TM) number of power converters can be reduced for the case of 64-core. Therefore, STM by adaptive clustering can perform resource allocation with minimum number of power converters to reduce the area overhead and also on-chip implementation cost.

Table 5.5: Comparison of number of allocated power converters under different power management schemes

		STM	SM	TM	STM/SM	STM/TM
32-core	Group 1	1	2	3	-50.00%	-66.67%
	Group 2	1	2	2	-50.00%	-50.00%
	Group 3	3	7	5	-57.14%	-40.00%
	Group 4	4	9	4	-55.56%	0.00%
	Total	9	20	14	-55.00%	-35.71%
64-core	Group 1	2	4	6	-50.00%	-66.67%
	Group 2	3	4	7	-25.00%	-57.14%
	Group 3	5	12	9	-58.33%	-44.44%
	Group 4	11	16	11	-31.25%	0.00%
	Total	21	36	33	-41.67%	-36.36%

5.5.2 Scheduling of Workloads

Once resource allocation is performed based on the learning of power-signature patterns, workload scheduling needs to be performed to reduce peak-power and achieve uniform workload balance. Recall that workload is the amount of work a core performs in one time-slot. A demand-response based workload scheduling will be developed towards uniform distribution with reduction in peaks on one power converter. Demand-response based workload scheduling is performed in two steps, namely peak-power envelope extraction and peak reduction.

5.5.2.1 Slack Calculation

Once the peak envelope for a subgroup k_j is formed, it is compared with threshold power $P_{th}(z)$ of group g_z to determine *slack*, which is the amount of extra workloads a power

converter can handle without getting over-loaded at a time-slot, and is calculated as

$$s(z, j) = P_{th}(z) - P_s(T_j^z). \quad (5.20)$$

If the value of slack is negative then, the allocated power converter r_j is over-loaded and not capable of handling extra workload at that time-slot. After calculating the amount of slack, workload on power converter r_j can be rescheduled without violating priority. We call such a scheduling as *demand-response based workload scheduling*.

The procedure for scheduling by considering priorities $b_a, b_a \in B$ of workload $l_a, l_a \in L$ is described in Algorithm 2. It is deployed after resource allocation. Workload $l_a, l_a \in L$ assigned to subgroup $k_j, k_j \in K$ of group $g_z, g_z \in G$ is denoted as $l_a(z, j)$.

Algorithm 2 Demand-response based workload scheduling

Require: Initial set: voltage-levels V , workloads L , priorities B and slack S

```

1: if  $s(z, i) > 0$  then
2:   while  $j > i$  do
3:     if  $s(z, j) < 0, s(z, i) > 0 \ \&\& \ \exists l_a(z, j)$  with  $b_a == 1$  then
4:        $l_a(z, j) \rightarrow l_a(z, i)$ 
5:        $s(z, j) ++$ ;
6:        $s(z, i) --$ ;
7:     end if
8:   end while
9:   while  $v_y < v_z \ \&\& \ j > i$  do
10:    if  $s(y, j) < 0, s(z, i) > 0 \ \&\& \ \exists l_a(y, j)$  with  $b_a == 1$  then
11:       $l_a(y, j) \rightarrow l_a(z, i)$ 
12:       $s(y, j) ++$ ;
13:       $s(z, i) --$ ;
14:    end if
15:  end while
16:  while  $j < i$  do
17:    if  $s(z, j) < 0, s(z, i) > 0 \ \&\& \ \exists l_a(z, j)$  with  $b_a == 0$  then
18:       $l_a(z, j) \rightarrow l_a(z, i)$ 
19:       $s(z, j) ++$ ;
20:       $s(z, i) --$ ;
21:    end if
22:  end while
23: end if

```

In the formulated algorithm, set of voltage-levels V is given as input along with workload priorities B and calculated slacks S . It is aforementioned that a power converter can handle a workload $l_a(z, i), l_a(z, i) \in L$ only if it has a positive slack $s(z, i), s(z, i) \in S$ in subgroup $k_i, k_i \in K$ of group $g_z, g_z \in G$. When a power converter is over-loaded due to larger workload or errors in predicting power traces, the workload with lower

Figure 5.12: (a) Workload before demand-response scheduling (b) workload after demand-response scheduling with peak reduction and balancing with slack calculated based on (5.20)

priority will be shifted to another under-loaded power converter, as explained in Lines 2-8 of Algorithm 2. After workloads are scheduled (if needed) within the same group, workload scheduling between different groups is performed, and it is important that the allocated voltage-level after scheduling must satisfy its demanded voltage-level. The same is shown in Lines 9-15 of Algorithm 2. After high priority workloads are scheduled for over-loaded power converters, workloads with low priority on an over-loaded power converter can be scheduled to an under-loaded power converter within the group. Lines 17-22 of Algorithm 2 explains the scheduling of low priority workloads within a group. Low priority workloads can be delayed, ideally till the availability of slack. Though Algorithm 2 is defined for two levels of priority, it can be extended to multiple-levels.

Peak-power reduction can be shown by the shift in workloads on a power converter from one time-slot with negative slack to another time-slot having a positive slack. Example in Figure 5.12 shows the normalized peaks of four subgroups. Before performing demand-response based workload scheduling, subgroup 2 and subgroup 3 are overloaded and subgroups 1 and 4 have slacks for scheduling. The peak value in subgroup 2 and 3 is 5, which means there are 5 peaks in those two subgroups. Peak-power reduction is then achieved with the comparison of the highest value in the subgroups before and after the demand-response scheduling. After the demand-response scheduling, peak value will be reduced to 4. So, a 20% peak-power reduction will be achieved. Workload balancing can be determined by calculating the standard deviation among cores between subgroups in a group. A larger standard deviation implies a less balanced workload.

Table 5.6: Demand-response based workload scheduling result for 32-core microprocessor

Workload distribution before workload scheduling (Algorithm 1)				
	Subgroup 1	Subgroup 2	Subgroup 3	Subgroup 4
Group 1	31	N/A	7, 8, 14	12
Group 2	17	22	27, 29	N/A
Group 3	6, 21, 32	9, 15, 20 26, 28	1, 5, 11 18, 19	16
Group 4	2, 3, 23, 25	10, 13	N/A	4, 24, 30
Workload distribution after workload scheduling (Algorithm 2)				
	Subgroup 1	Subgroup 2	Subgroup 3	Subgroup 4
Group 1	31, 7	N/A	8, 14	12
Group 2	17	22	29	N/A
Group 3	6, 21, 32, 9	15, 20, 26, 28	1, 5, 11, 18	16, 19
Group 4	3, 25	10, 13, 27	2, 23	4, 24, 30

5.5.2.2 Workload Balancing

The aforementioned demand-response based workload scheduling can be deployed to solve subproblem 2, which can be reformulated as

$$\begin{aligned}
\min: & \sum_{j=1} | \sum_{z=1} s(z, j) | \\
\text{s.t.}: & P_s(T_j^z) < P_{th}(z).
\end{aligned} \tag{5.21}$$

Solution to this problem is to minimize the overall sum of slacks. This can be achieved by rescheduling workloads that overloads power converters. Based on the value of slack for a subgroup k_j , $k_j \in K$, if the slack is negative, then the workload on that subgroup needs to be delayed or advanced to other time-slot. As such, the workloads are allocated to subgroups with highly negative slack, and the difference in slack is reduced. As a result, peak-power reduction and workload balancing can be achieved eventually.

Let us discuss the result of demand-response based workload scheduling of allocated power converters. The peak reduction is calculated as the difference in peak-power value before and after scheduling. The workload balancing is achieved by having uniform number of workloads on a power converter. The workload balance is calculated by averaging the standard-deviations of workloads on each power converter.

Firstly, based on the availability of slack and the workload priority, demand-response based workload scheduling is performed. Table 5.6 presents the results of demand-response based workload scheduling (Algorithm 2) performed in addition to adaptive clustering based resource allocation for a 32-core case. For example, considering core

Table 5.7: Comparison of number of allocated power converters with and without workload balancing

		Adaptive clustering	Workload balancing	Reduction
32-core	Group 1	1	1	0.00%
	Group 2	1	1	0.00%
	Group 3	3	2	33.33%
	Group 4	4	3	25.00%
	Total	9	7	22.22%
64-core	Group 1	2	1	50.00%
	Group 2	3	2	33.33%
	Group 3	5	3	40.00%
	Group 4	11	6	45.45%
	Total	21	12	42.86%

Figure 5.13: Peak-power reduction for 4 subgroups of 64-core case

27, it is initially assigned to subgroup 3 of group 2; but after performing demand-response based workload scheduling, it is shifted to subgroup 2 of group 4. Note that the voltage supplied by group 4 is higher than group 2, which implies that the allocated voltage is higher than demand voltage. As such, this shifting of workload reduces the peak-power on power converters, thereby avoiding over-loading of power converters. As this implementation involves comparisons, not much runtime overhead is incurred.

What is more, based on equation (5.21), the number of power converters required to meet demand from cores can be calculated. After performing demand-response based

Table 5.8: Peak reduction and workload balancing by demand-response scheduling

	Peak Reduction	Balance before	Balance after
Group 1	33.33%	0.91	0.58 (1.57X)
Group 2	50.00%	1.09	0.75 (1.45X)
Group 3	33.33%	0.93	0.17 (5.59X)
Group 4	45.45%	0.51	0.36 (1.41X)
Average	40.53%	0.86	0.46 (2.50X)

workload scheduling, the maximum number of cores in a subgroup is eventually decreased. This leads to the reduction in the number of power converters required to drive the cores. For example, if STM by adaptive clustering is performed on a 32-core microprocessor, group 3 has a maximum of 5 cores among its subgroups driven by power converters that have a driving capacity of 2. To meet the demand, 3 power converters are allocated. However, when the workload balancing is performed, the maximum number of cores among subgroups is reduced to 4, demanding only 2 converters. Thus, there is a reduction of 33.33% in power converters needed for group 2. The reduction in power converters after workload scheduling by DVS is summarized in Table 5.7.

For a 64-core microprocessor, results of peak-power reduction is shown in Figure 5.13, in group 4 the normalized peak-power value has been reduced from 11 to 6 with 45.45% peak-power reduction. The average standard deviation of workload on each power converter before and after scheduling are 0.86 and 0.46 respectively. Table 5.8 shows the summarized results of peak reduction and workload balancing by demand-response based workload scheduling for a 64-core microprocessor. One can observe an average of 40.53% peak-power reduction and $2.50\times$ workload balancing.

5.6 Summary

Power management of many-core microprocessors with a small number of power converters is discussed in this chapter. A DVS based on-chip power management is performed with the help of on-chip SIMO power converters. A space-time multiplexing based on-chip power management is developed for demand-supply matching between on-chip power converters and many-core microprocessors. The demand-supply matching problem is subdivided into resource allocation and workload scheduling subproblems. Two solutions for the resource allocation subproblem: integer linear programming for small-scale and adaptive clustering for large-scale are presented. Reduction in the number of power converters is also observed while using STM based on-chip

power management. What is more, demand-response based workload scheduling is deployed by utilizing the available slacks to achieve a reduced peak-power as well as balanced workload. Experimental results for a 64-core microprocessor have shown that the space-time multiplexing can reduce peak-power by 40.53% and improve load balancing by $2.50\times$ on average along with a reduction of 45.45% in required power converters. Power management with adaptive clustering and workload scheduling are the major contributions of the respondent author and is published in TC'15.

Chapter 6

Signal I/O Bandwidth Management

6.1 Introduction

Existing many-core microprocessors with shared memory-logic integration [156] has limited bandwidth that is non-scalable for exa-scale computing. With the increase in the number of integrated multi-core and memory blocks for a cloud server, the number of signal I/Os grows dramatically, hence there is an emerging need to develop high data-rate and low power I/O circuits [157, 158]. Though 3D integration by stacking several layers of dies vertically using through-silicon via (TSV) I/O [133, 29, 31, 159, 37, 75, 160, 5, 96] provides a scalable integration, accumulated heat in top layers and complexity of heat removal techniques can limit the flexibility of the design [161, 27, 28]. As discussed in previous chapters, 2.5D integration by through-silicon interposers (TSIs) in common substrate has a strong heat dissipation capability [10, 39]. With the design of TSI based transmission line (T-line), one can further achieve high bandwidth and low power [71, 20] I/O communication without the area overhead, because the TSI is fabricated underneath the common substrate. Compared to 3D TSV for memory-logic integration, As such, the TSI T-line based I/O circuit designs have become the recent interest towards energy-efficient integration of multi-core microprocessors and main memory.

The use of many-core microprocessors with shared memory for data-oriented computation with large number of signal I/Os impose two main challenges namely bandwidth and I/O communication power management. As there is a limited number of TSI I/O channels to access the shared memory, this calls for a channel reutilization under quality-of-service (QoS) [67] constraint. Memory controller [162, 163, 14, 164, 165, 96] is commonly employed as the bridge between cores and shared memory. State-of-the-art memory controller [97] is mainly designed as off-chip interfaces with a large number

of I/O pins for memory-logic integration to be managed. Memory controllers can be virtually partitioned into two parts, front end and back end [14]. The front end performs port arbitration and I/O queueing and the back end deals with the memory part. Similar to [166], here we discuss a protocol scheme to manage the I/O ports with the aid of the memory controller. and assume state-of-the-art back end which interfaces with memory.

Note that the management of memory controllers can be classified into static and dynamic upon whether the requests are determined at run time or design time. Static memory controllers such as [167] execute the controlling scheme based on the commands provided at the design time. A major drawback of the static memory controllers is their limited capability for one already specified application at design time. Dynamic memory controllers [168] can perform scheduling at run time based on the real-time requests. It is however non-scalable for many-core microprocessors to access one shared memory. For example, firm real-time controllers (FRT) are designed to meet the timing constraints [169], which are only efficient when all the requests are of the same kind due to less locality exploitation. Soft or no real-time memory controllers [163] maximize bandwidth, irrespective of satisfying timing requirements, and lacks memory efficiency with a better quality-of-service (QoS). In this chapter, we focus on the reconfigurable dynamic memory controller for the reuse of I/O channels based on the memory access characteristics. As such, one can improve the utilization of the I/O channels with a good QoS.

In this chapter, firstly, based on the memory-access characteristics, we introduce a reconfigurable dynamic memory controller for the reuse of 2.5D I/O channels such that one can improve the utilization rate with good QoS. To satisfy the demands from cores to access the shared memory with matched supply of 2.5D I/O channels, a QoS-constrained communication is achieved with reconfigurability of I/O channel connection by space-time multiplexing using crossbar-switches.

6.2 2.5D Bandwidth Management

Signal I/Os are utilized for the communication between cores and memory blocks. However, a 2.5D integration by TSI I/Os has less bandwidth utilization compared to 3D integration by TSV I/Os. Additionally, the communication power is as well significant due to frequent communication between memory and microprocessor blocks. In this section, we present the system architecture with signal I/Os and formulate bandwidth and voltage-swing management problems.

Figure 6.1: Side-view of 2.5D many-core microprocessor and main memory integration by TSI I/O channels

Figure 6.2: Memory-access data-pattern aware reconfigurable memory controller with space-time multiplexing

6.2.1 System Architecture

Figure 6.1 shows the 2.5D architecture utilized for many-core memory-logic integration with the smart reconfigurable memory controller for TSI I/O channel management. It consists of 2 different dies on one common substrate with TSI I/Os. The die on the left composes of many-core microprocessors and the die on the right is the main memory. Cores can access the main memory through a reconfigurable crossbar switch-network [170] to perform the space-time multiplexing (STM) inside the memory controller by configuring TSI I/O channel connections between many-core microprocessors and the shared main memory. The crossbar switch-network is suitable for such a many-core microprocessor with shared memory by the following advantages: simple one-hop routing

and ease of implementing QoS policy. Besides, it can be easily reconfigured based on the data-pattern analysis. The switch-network can be reconfigured to connect with TSI I/O channels. A more detailed view of the proposed architecture is presented in Figure 6.2. DRAMs are assumed to be having a complex architecture to satisfy large bandwidth demands from multiple cores with reasonable speed. As explained in [171] the DRAM is organised in the form of cell arrays, with rows read or written through a series of data movements. A number of banks can be operated concurrently and share same data, address and command bus. This DRAM configuration is the default configuration in DRSim and is not modified. The system employs two clocks with cores running at high-speed and memory at lower clock, the memory controller is helpful in synchronising the memory and processor clocks. In fact, one can model the 2.5D system architecture by a demand-supply system with the following three components:

- *I/O channel demander*: a set of microprocessor cores C of set-size N_c having different bandwidth demands. Each core c_i has a bandwidth demand of $B_d(c_i)$.
- *I/O channel supplier*: N_{ch} TSI I/O channels which together can supply bandwidth B_T . The allocated bandwidth $B_a(c_i)$ for each core c_i is supplied with TSI I/O channels by the reconfigurable memory controller.
- *I/O channel controller*: a set of M memory ports with scheduler inside the reconfigurable memory-controller to map requests from N_c cores to memory through N_{ch} TSI I/O channels under demand-supply matching by flexible crossbar switch-network. Note that the TSI I/O channel management in memory controller can be implemented by simple logics of address decoder, data queue, request queue and scheduler.

6.2.2 Problem Formulation

As there exists time-varying heterogeneous bandwidth demands from cores to access the shared main memory, the TSI I/O channels may not be fully utilized under a fixed connection. To manage I/O channels with time-varying bandwidth demanded from many cores, the main idea here is to learn the memory-access data-patterns so that one can perform a reconfigurable space-time multiplexing inside the memory controller to fully utilize the TSI I/O channels. For example, two cores with workloads that have similar data-patterns can be classified into the same cluster. Overloading of I/Os due to large bandwidth demands (by clustering the cores) are mitigated by assigning priority inside a cluster. Based on the classified clusters, each with a demanded signature bandwidth, one can further schedule the cores to occupy the I/O channels inside one cluster based on the

priority. This is how the complexity for a large-scale demand-supply matched signal I/O management can be reduced to implement inside the smart memory controller. One can formulate a space-time multiplexing design problem as follows:

Problem: A reconfigurable memory controller can perform space-time multiplexing that adaptively allocates N_{ch} TSI I/Os to N_c cores such that:

$$\begin{aligned}
\min: & \sum_{i=1}^{N_c} |B_a(c_i) - B_d(c_i)| \\
s.t.: & (i) B_a(c_i) \geq B_d(c_i); \\
& (ii) Q_i \leq E_R \quad \forall i = 1, 2, \dots, N_{ch}
\end{aligned} \tag{6.1}$$

where $B_d(c_i)$ and $B_a(c_i)$ are the demanded and allocated bandwidths for core c_i respectively; Q_i and E_R is the number of request at i -th channel and maximum requests capacity of one channel respectively. The main objective is to reuse I/O channels such that the demand and supply can be matched. In the following, we present a solution by studying the memory-access data patterns.

6.3 Signal I/O Bandwidth Management

In this section, we study the memory-access data-pattern characteristics first, followed by classification of memory-access data-pattern to allocate the signal I/Os.

6.3.1 Memory-access Data-pattern

The analysis of memory-access data-pattern with the according QoS analysis which will be utilized for space-time multiplexing is presented here. In this chapter, we define control-cycle with period of T_c , and a control-cycle is divided into time-slots with period of T_s . Within each control-cycle T_c , I/O channels are allocated; while at each time-slot T_s , cores are allocated with I/O channels.

6.3.1.1 LLC MPKI Pattern

There exist many approaches to describe the memory-access data-pattern of workload-s. Last-level cache (LLC) misses-per-kilo-instructions (MPKI) is an important metric to indicate the communication intensiveness between cores and memory. Higher LLC MPKI indicates larger bandwidth requirement. DRAM row buffer hit-rate is another metric to show the spatial locality of the workload. Since, the focus is mainly on

Figure 6.3: Analysis of memory-access data pattern

the logic-memory communication, LLC MPKI pattern is considered for analyzing the memory-access data-pattern. Note that the memory accesses number can be inferred by LLC MPKI [172].

Memory-access number $\mathbf{M}(c_i)$ for a core c_i within a control-cycle T_c is the sum of access number at each time-slots of T_s , given as

$$\mathbf{M}(c_i)|_{T_c} = \sum_{t=1}^{T_c/T_s} \mathbf{M}(c_i)|_t. \quad (6.2)$$

Bandwidth demand $B_d(c_i)$ of core c_i is related to the memory-access number by

$$B_d(c_i) = \frac{\mathbf{M}(c_i) * L_c}{T_c} \quad (6.3)$$

where L_c is the last-level cache line size; and $\mathbf{M}(c_i)$ is memory-access number.

Based on the memory access number, one can classify the communication traffic pattern of cores as clusters indicated by the similar demanded bandwidth. What is more, inside one cluster, the arrival time of requests and the number of memory access requests can be different among cores. As such, one can further define priority R_i , indicating core c_i 's ability to occupy the allocated I/O channels at time-slot T_s . The priority R_i will be raised when core c_i has an early arrival request or more number of memory access requests to be served if multiple requests arrive at same time. Core assigned with the highest priority will be allocated with the I/O channel for accessing memory.

Classification of memory-access data-pattern based on the bandwidth demands (LLC and number of memory requests) is shown in Figure 6.3. We classify benchmarks as high, medium and low memory-access benchmarks. For example, a core running

Figure 6.4: SPEC 2006 memory-access data-patterns: (a) memory access for execution time of 10ms; (b) memory access for one control-cycle of 1ms

lbm benchmark having LLC MPKI of 59 (represented by vertical bar) and 233k memory requests can be categorized under high memory-access, whereas a core running *libquantum* benchmark will be classified under low memory-access benchmarks due to low LLC MPKI of 0.24 and 1776 memory requests. One needs to observe that LLC and number of memory access requests are proportional.

Further, the process of classifying data-patterns with similar memory-access numbers is illustrated here. From Figure 6.4(a), one can see that *lbm* and *gobmk* have similar bandwidth signatures and high memory-access demands compared to *libquantum* and *mcf*. Thus *lbm* and *gobmk* can form a cluster (space multiplexing). In Figure 6.4(b) we present the memory-access number within one control-cycle. Variation in number of memory-access demands of cores in one cluster (*lbm* and *gobmk*) can be observed, based on which priorities can be assigned to occupy the I/O channels.

6.3.1.2 Quality of Service

The system performance is sensitive to long-latency memory requests because instructions dependent on the long latency load cannot proceed until the load completes. As main memory access latency is much higher than cache access, the last-level cache miss will inevitably stall the core executions. More number of memory access served indicates a better performance. Hence, the memory controller must balance the accesses from different cores with good quality-of-service (QoS) maintenance mechanisms [173].

We define QoS as the ratio of total number of requests processed to the total number of requests. Higher *QoS* and more requests processed indicates a better performance. Hence, QoS is considered as the metric to determine the matching between demand and

the supply, given by

$$QoS = \frac{\sum_{i=1}^{N_c} r_i \mathbf{M}(c_i)}{\sum_{i=1}^{N_c} \mathbf{M}(c_i)} \quad (6.4)$$

where r_i is the ratio of processed requests for core c_i .

6.3.2 Reconfigurable Memory Controller

To solve the demand-supply matching problem by space-time multiplexing inside the memory controller, the time-varying memory-access data-pattern (LLC MPKI) of cores is first extracted and then utilized in space multiplexing for channel allocation as well as in time multiplexing for time-slot allocation.

6.3.2.1 Space Multiplexing: Channel Allocation

Here, the adaptive allocation of I/Os to the cores by space multiplexing is discussed. The demand here is the bandwidth requirement from the cores running workloads and the supply is I/O to support the access to memory. To deal with large number of cores, the cores can be classified based on memory-access data-pattern (LLC MPKI).

Clustering here can be defined as the process of grouping cores with similar demanded signature bandwidth. Each cluster is allocated with cores and connected to one of the ports of memory controller through on-chip routers. Inside the memory controller, cores are virtually partitioned and allocated to each port of the reconfigurable memory controller which can meet the demanded bandwidth. The number of cores for different clusters may be different. Note that the configuration of virtual clusters can change adaptively. If the demand does not vary, the core is expected to remain in same cluster.

For example, z -th cluster G_z at port p_z is formed by

$$G_z = \{c_i | B_d(c_i) \leq B(p_z); c_i \in C, z = 1, 2, \dots, M\} \quad (6.5)$$

where $B_d(c_i)$ is the bandwidth demanded from core c_i ; and $B(p_z)$ is the bandwidth allocated to the port p_z of the memory controller. Such a clustering by the magnitude of the memory-access is called space multiplexing. Once the clusters are formed, cores in one cluster can be further assigned with priorities to access TSI I/O channels in a time-multiplexed manner to avoid overloading of TSI I/O channels.

6.3.2.2 Time Multiplexing: Time-slot Allocation

The time-slot allocation for time multiplexing can be described as follows. Memory-access requests from different cores can arrive at different time-instants and can have

different memory-access number. The core with earliest request will access the I/O channel first, while other cores stalled.

Core with the early request arrival time will be assigned the high priority defined as

$$R_i = H \text{ if } t_a(c_i) < t_a(c_j). \quad (6.6)$$

where H indicates high priority and $t_a(c_i)$ indicates the arrival time of the memory-access request from core c_i ; and R_i indicates the priority for core c_i . In case, when there are multiple requests with same arrival time, the priority is assigned to the core with more number of memory access requests. This is how the time multiplexing is performed inside the cluster based on the priority assigned to the core. It needs to be noted that the priority can change due to the change in the memory-access data-pattern (LLC MPKI), and so does the multiplexing of I/Os.

6.3.2.3 Space-Time Multiplexing based I/O Bandwidth Management

The space-time multiplexing based I/O management inside the memory controller is given in Algorithm 3.

Algorithm 3 Proposed signal I/O bandwidth management

Require: Set of cores C , ports P , bandwidth demands $B_d(c_i)$

```

1: for  $i = 1 : N_c$  do
2:    $G_z = \{c_i | B(p_{z-1}) < B_d(c_i) \leq B(p_z); z = 1, 2, \dots, M\}$ 
3: end for
4: channel allocation for each cluster
5: for  $z = 1 : M$  do
6:   With all cores inside  $G_z$ 
7:   for  $t = 1 : T_c/T_s$  do
8:     if  $t_a(c_i) = \min(t_a | G_z)$  then
9:        $R_i = H$ 
10:    else if  $t_a(c_i) = t_a(c_j)$  then
11:       $R_i = H$  if  $M(c_i) > M(c_j)$ 
12:    end if
13:   end for
14: end for

```

Ensure: cluster G , allocated I/O channels

Firstly, set of ports P are labeled in ascending order of bandwidth it can provide i.e., $B(p_z) > B(p_{z-1})$. Based on the demanded bandwidth, cores with similar bandwidth demands are grouped into one cluster and connected to one port with required TSI I/Os. As such, $B(p_z) = B_d(c_i) > B_d(c_j)$. This is discussed in Line 1-4 of Algorithm 3. Once

Figure 6.5: Illustration of LLC MPKI memory-access data-pattern: (a) bandwidth; (b) priority

the cores are clustered, priority based scheduling will be performed at each time-slot. The requests from cores will be processed on a first-come-first-serve principle. At each time-slot, if there is only one request, the corresponding core will be served first (Line 8-9). If there are more than one requests, then the core with more number of requests is assigned higher priority and served (Line 11). Here ‘served’ implies that the TSI I/O channel is occupied by the core with memory access demand. This process is repeated for every control-cycle T_c . Thus, I/O channel management can be performed in a space-time multiplexed manner to improve I/O utilization and QoS.

With the above mentioned space-time multiplexing based I/O management, demand-supply matching is achieved with a better utilization rate of TSI I/Os. This improvement of QoS and I/O utilization rate helps to achieve a better energy-efficiency. This implementation may pose some challenges due to extra control logic required.

Figure 6.5 shows the memory-access data-pattern by LLC MPKI from different cores. Figure 6.5(a) illustrates core bandwidth demand $B_d(c_i)$, where $i = 1, 2, 3, 4$. At control-cycle 1, cores c_1, c_3 and c_4 are allocated with bandwidth $B(p_1)$, while core c_2 with bandwidth $B(p_2)$. As such, cores with different workloads can be clustered based on the magnitude in space, for space multiplexing.

Further within the cluster consisting of cores c_1, c_3 and c_4 , all cores will compete for channel at each time-slot. To avoid overloading of I/O channels, a priority based time multiplexing is implemented. Figure 6.5(b) shows the allocation of channels based on the priority at a time-slot. Memory access request from one core is shown by a rectangular box. At time-slot 1, since there is only one request from c_4 , I/O channel will be allocated to it. For the next time-slot, there are simultaneous requests from both c_1 and c_3 , so priority needs to be assigned to access memory through I/O channel. Here, when multiple requests arrive at the same time, priority is decided based on the total number of memory access requests, which is 2 for c_1 and 4 for c_3 . Hence, core c_3 is

Figure 6.6: Adaptive clustering of cores at two consecutive control-cycles

assigned with a higher priority and occupies the I/O channel. Core assigned with the high priority is shown with red outline. Thus, the cores are further classified based on the phase or priority in time, for time multiplexing. We set the control-cycle T_c and time-slot T_s as $1ms$ and $0.1ms$ respectively.

To evaluate the I/O bandwidth management, gem5 simulator [149] is utilized for many-core microprocessors and DrSim simulator [174] is employed for shared main memory. The proposed space-time multiplexing I/O management using smart reconfigurable memory controller is implemented inside the DrSim simulator. The TSI I/O channel model is based on [10] and the length is $1.5mm$. The crossbar switch-network inside memory controller is estimated with $1mm^2$ area and $1GHz$ frequency for a 64-core system under $32nm$ design [170]. The benchmarks are selected from SPEC 2006 [175] which have high memory-access demand, PARSEC [176] with medium memory-access demand and Phoenix [177] with less memory-access demand. Furthermore, we assume a baseline system which has fixed core-to-memory connections, while the proposed system employs the reconfigurable connections. A reconfigurable crossbar switch with STM logic may incur small power and latency overhead similar to [156]. Additional details about system architecture is presented in Table 6.1.

Table 6.1: System parameters

Processor core	1GHz x86
L1 I-cache	32kB private, 64B cache line
L1 D-cache	32kB private, 64B cache line
L2 cache	256kB private, 64B cache line
Main memory	4GB capacity, 800MHz DDR3-1066 channel, x8 DRAM chips, 8 banks per channel

Figure 6.7: Variation in number cores allocated to a cluster with control-cycles

Adaptive Clustering Analysis Here, the adaptive clustering analysis of cores based on the signature of demanded bandwidth is discussed. For a 16-core case, the assumption is the total number of I/O channels are 8 and the number of ports is 4 and each core is assigned randomly with a benchmark. For the baseline system with fixed connections, each port is assigned with four cores and each port is connected to 2 I/O channels. While for the proposed system, each port connects to 4, 2, 1 and 1 I/O channels respectively to meet the memory access demands from high traffic to low traffic benchmarks. A similar setup is assumed for 64-core case as well.

Figure 6.6 illustrates the adaptive clustering result of 16-core case at two consecutive control-cycles (5ms to 6ms). Different filling patterns represent different clusters. Cluster 1 handles high-traffic workloads, cluster 2 are used for middle-traffic workloads, and cluster 3 and cluster 4 are allocated with low-traffic workloads. For example at 5ms, core 12 running *gcc* benchmark is assigned to low traffic cluster 4, but is assigned to middle traffic cluster 2 in the next control-cycle due to time varying memory-access characteristics. Similarly, Figure 6.7 shows the variation in number of cores allocated to a cluster with time. For example, at 3ms, cluster 4 is allocated with 18 cores, but is allocated with 20 cores at 4ms.

Bandwidth Balancing Bandwidth balancing across all ports can improve the I/O channel utilization efficiency. We use requests per channel to measure the traffic flow at each control-cycle and calculate the standard deviation for all four ports. The standard deviation shows how much variation from the average value. A low standard deviation indicates more bandwidth balancing. Bandwidth balancing for a 16-core and 64-core

Figure 6.8: Bandwidth balance across TSI I/O channels with: (a) 16-cores; (b) 64-core under randomly distributed benchmarks of SPEC 2006, Parsec and Phoenix

microprocessor, with randomly distributed SPEC 2006, Parsec and Phoenix benchmarks to cores is shown in Figure 6.8. For example in 64-core microprocessor, at 5ms the baseline system show standard deviation of 26k memory requests, while the proposed system just requires 11.5k with 55.90% bandwidth balancing improvement. The average of standard deviation for baseline and proposed system is 27k and 13k memory requests, indicating a 14k reduction in deviation. On average the proposed system can improve the bandwidth balancing by 58.25% under 16-core case and 58.85% under 64-core case.

QoS Analysis For a 16-core microprocessor with SPEC 2006, Parsec and Phoenix benchmarks randomly distributed on cores, comparison of QoS for proposed system and baseline architecture is presented first. Improvement in QoS by proposed system is shown in Figure 6.9. The average QoS for the baseline system is 0.472, whereas for the proposed STM achieves a QoS of 0.561. Further considering a 10ms span (100 time-slots) as an example, the average QoS achieved by proposed, baseline and time multiplexing are 0.589, 0.470 and 0.529 respectively, as presented in Table 6.2. This indicates nearly 11.90% more requests are served and improvement in QoS by the proposed method compared to baseline. Additionally, the number of requests served the memory controller with the method is also given in Table 6.2. Whereas for 64-core case the proposed system achieves a QoS of 0.577 compared to QoS of 0.461 and 0.514 achieved by baseline and time multiplexing techniques respectively. Use of space multiplexing alone may result in large congestion at memory controller, hence not compared.

6.4 Summary

In this chapter, a data-pattern aware reconfigurable memory controller is demonstrated for 2.5D memory-logic integrated microprocessor. With the reconfigurable crossbar

Figure 6.9: Communication QoS efficiency improvement by STM I/O management for 16-core based on (6.4)

Table 6.2: Number of requests served and achieved QoS

For 16-core microprocessor		
Method	Requests served	QoS
Baseline	448578	0.473
Time multiplexing	523840	0.528
Proposed	580399	0.589
For 64-core microprocessor		
Method	Requests served	QoS
Baseline	2092083	0.461
Time multiplexing	2340843	0.514
Proposed	2622005	0.577

switch-network inside the memory controller, bandwidth demands from many-core to access the shared memory can be managed by reuse of available limited number of T-SI I/O channels. A space-time multiplexing based communication between cores and memory is realized by reusing the I/O channels with improved communication efficiency. By adaptive clustering cores upon the magnitude of memory-access patterns, I/O channels are allocated to core clusters by space multiplexing. With further considering priority upon the phase of memory-access patterns, time-slots are allocated to access I/O channels in one cluster by time multiplexing. The adapted architecture is verified by system-level simulator with benchmarked workloads, which shows 58.85% bandwidth balancing with 11.90% QoS improvement. The data-pattern classification and scheduling of memory-access requests are the contributions from the respondent author and is published in D&T'15.

Chapter 7

Signal I/O Voltage-swing Management

7.1 Introduction

With the increase in the number of integrated multi-core and memory blocks for a cloud server, the number of signal I/Os grows dramatically, hence there is an emerging need to develop high data-rate and low power I/O circuits [157, 158]. It is reported in [68] that I/Os consumes nearly 50% of system dynamic power. Previous 2D wire-line communication by PCB trace of backplane [53, 99, 100] has a large latency and poor signal-to-noise ratio (SNR) and large channel loss. Compared to 2D PCB trace for memory-logic integration, 2.5D TSI I/O has much higher energy efficiency in communication [71, 178].

On-line machine learning has been recently practiced [179, 180, 181, 182, 183, 40]. For example, Q-learning [184, 185] can be utilized to find an optimal action-selection policy from the set of states. One conventional Q-learning based management is applied to adjust the level of output-voltage swing at transmitter (associated with cores) such that one can achieve a reduced power under specified BER requirement. However, conventional Q-learning is limited by the slow convergence rate [186]. In [187], a large range of learning rates are dynamically applied to increase the convergence rate. However, the complexity increases with the range of learning rates. An accelerated Q-learning can improve the policy decision by using the prior knowledge of the system with a Markov decision process (MDP) [180]. In [180], accelerated Q-learning is applied to a multimedia system to achieve trade-off between distortion rate and complexity.

Previous I/O circuit designs such as [93, 94, 95, 12] assume a constant and large output-voltage swing. For 2D wire-line communication, a large output-voltage swing is required to compensate the channel loss and noise of the PCB trace. But a large output-voltage swing with high data-rate results in large power consumption. In [12],

a single-ended low-swing voltage transmission scheme has been proposed with self-resetting logic repeaters (SRLRs) embedded inside the routers. Due to the use of single-ended signalling, the communication is much prone to the noise. Meanwhile, bit error rate (BER) requirements are not stringent for all applications. Therefore, use of constant and large output-voltage swing for I/O communication may be an over design with low utilization. Thereby, in order to have an energy-efficient I/O communication, one can develop a dynamic output-voltage swing scaling to adjust the output-voltage swing level under the dynamic BER constraint [188, 189].

Firstly, one basic Q-learning based management is applied to adjust the level of output-voltage swing at transmitter of 2.5D TSI I/Os such that one can achieve a reduced power under specified BER requirement. To overcome the slow convergence issue in the basic Q-learning algorithm, an online reinforcement Q-learning based management is further developed to adaptively adjust the output-voltage swing levels of 2.5D TSI I/Os. Instead of transmitting data under a fixed large voltage-swing, an on-line reinforcement Q-learning based management is applied to select the output-voltage swing at the transmitter. Based on the historical data, the voltage-swing adjustment is formulated as a MDP problem solved by model-free reinforcement learning under constraints of both power budget and BER. To accelerate the adjustment convergence, a prediction of BER and power as virtual experience is applied to reinforcement Q-learning algorithm. A corresponding 2.5D TSI I/O is designed in 65nm CMOS process for multi-level output-voltage swing with balanced power and BER.

7.2 2.5D Voltage-swing Management

7.2.1 System Architecture

Figure 7.1(a) shows the block diagram of memory-logic integration by the 2.5D TSI I/O. Each of the core and memory blocks comprises of transmitter as well as receiver to enable a full duplex communication. To further reduce the I/O communication power between logic and memory blocks, a self-adaptive voltage-swing adjustment is required. The current-mode logic (CML) buffer is shown in Figure 7.1(b) with tunable tail-current (based on the output of I/O controller) to adaptively tune the output-voltage swing. By adjusting the I/O output-voltage swing, I/O communication power can be reduced with improved energy efficiency compared to the previous designs [93, 94, 95] that utilize the fixed full output-voltage swing. However, BER increases when the output-voltage swing decreases. Hence, a trade-off needs to be maintained between the I/O communication power and BER, which requires an optimized on-line management. The geometry and

Table 7.1: System settings for memory-logic integration with TSI I/O

Item	Description	Value	Size
Microprocessor	Technology node	65nm	0.3mm ²
	Frequency	500MHz	
	Dissipation power	15mW	
I/O controller	Output-voltage swing	0.1V, 0.15V, 0.2V, 0.3V	0.03mm ²
	Driving current	2mA, 3mA, 4mA, 5mA	
	Number of levels	4	
	Switching time	0.4ns	
TSI	Length	3mm	3mm ²
	Inductance	300pH	
	Resistance	5Ω	
	Capacitance	60fF	
Memory	SRAM	16 KB	0.2mm ²
	Power dissipation	6mW

Figure 7.1: (a) Core-memory integration by 2.5D TSI I/O interconnect and its cross sectional view; (b) adaptive tuning I/O based on error checking and correction

technology details are presented in Table 7.2.1. To solve the problem 2 of voltage-swing tuning, we make use of the online Q-learning.

7.2.1.1 BER and Communication Power Trend

Before performing the voltage-swing management, we present the trend of BER, communication power versus voltage-swing levels for different benchmarks. Figure 7.2

Figure 7.2: Trade-off between BER and power under various benchmarks: (a) *FFT* benchmark; (b) *Filetransfer* benchmark; (c) *bzip2* benchmark; (d) *gzip* benchmark.

shows the trade-off between BER (blue circle) and the I/O communication power (black rectangle). With the increase in output-voltage swing, BER decreases at the cost of the I/O communication power, which is validated through 4 benchmarks. For example, for *filetransfer* benchmark, at an output voltage-swing of $350mV$, the I/O communication power is $0.693mW$ with a BER of $4.85E - 7$; whereas at an output-voltage swing of $400mV$, BER goes down to $4.47E - 9$ at the cost of increased communication power by $0.953mW$. This observation shows that there can be a balance point where we can use less power with the BER guaranteed. Furthermore, as the sensitivity of power and BER with voltage-swing level are different for each benchmark, adaptive tuning of the I/O voltage swing based on the conventional Q-learning and further accelerated Q-learning can help to achieve an optimal trade-off between the communication power and BER.

7.2.2 Problem Formulation

As previously discussed, the BER at the receiver increases with the decrease of I/O communication power due to channel loss and noise. Thus, one needs to find an optimal output-voltage swing at the transmitter such that balanced power reduction is obtained

with the maintained BER, which can be defined as the following problem.

Problem: Tune the output-voltage swing at the transmitter to achieve low power at the cost of BER based on the I/O communication channel characteristics.

$$\begin{aligned}
& \text{Opt. } P_{w_i}, BER_i \\
& \text{S.T. (i) } P_{w_i} \leq P_{w_T} \\
& \quad \quad \quad (ii) BER_i \leq BER_T
\end{aligned} \tag{7.1}$$

where P_{w_i} and BER_i denotes the I/O communication power and BER under the i -th output-voltage swing level V_i . Note that the BER and power are both functions of the output-voltage swing. P_{w_T} and BER_T represents the targeted I/O communication power and BER of one TSI I/O channel under the normal operation. With the increase of output-voltage swing, the I/O communication power increases and BER decreases and vice-versa. As such, the output-voltage swing level V_i needs to be adaptively tuned for optimizing the I/O communication power and BER simultaneously. Here, a self-adaptive tuning of the output-voltage swing at transmitter is performed based on the conventional Q-learning and further accelerated Q-learning discussed later in this chapter.

7.3 Signal I/O Voltage-swing Management

7.3.1 Reinforcement Q-learning

Here, the basics of Q-learning theory is first presented followed by the conventional Q-learning, modeling of Markov decision process (MDP) and the according accelerated Q-learning algorithm for adaptive voltage-swing tuning.

7.3.1.1 Q-learning Theory

Machine learning algorithms such as Q-learning theory [190] and reinforcement-based Q-learning [180] are generally practised to find an optimal action-selection policy from the set of states S . Both these algorithms evaluate *state* and *action* pairs from the previous inputs. To solve (7.1) using conventional Q-learning and further by accelerated Q-learning algorithms, the I/O communication power P_w and bit-error-rate BER are considered as the state vector; and the tuning of output-voltage swing level V_i as the action. State vector S can be given as

$$S = \langle P_w, BER \rangle .$$

Before describing the basic Q-learning and reinforcement-based Q-learning algorithms, few terminologies are presented below:

- State S : set of states indicating the value of system variable(s). We consider the communication power and BER as the state vectors.
- Action A : set of actions indicating the change of state. We consider the change of output-voltage swing level as the action.
- State transition probability $P(s_i, a_k, s_j)$: probability indicating whether to take an optimal action a_k based on Q-learning or perform random action.
- Reward $R(s_i, a_k, s_j)$: evaluation value of action a_k to change the state from s_i to s_j , which is dependent on historical BER and power P_w .
- Q-value $Q(s_i, a_k)$: set of accumulated Q-values to measure the benefits of taking action a_k at state s_i .
- Expected Q-value $\hat{Q}(s_i, a_k)$: set of values to measure the expected benefits of taking action a_k at state s_i .
- Policy: process of state change under sequence of action.

In order to obtain the state and action pairs and form a look-up-table (LUT), the input samples are trained. A sample LUT can be formed as follows:

Action (Voltageswing)	State	
	Power	BER
$a_1(V_1)$	P_{w1}	BER_1
$\vdots$	$\vdots$	$\vdots$

The input samples are collected at regular time intervals, called control cycle, at scale of ns . Control cycle can be defined as the minimum time required for the state transition. Duration of control cycle is based on the speed of I/O controller circuit. The next state variable needs to be predicted with an action for the next input sample. This can be done by calculating a reward function to achieve an optimally estimated value based on the state vectors, given by

$$R = f(P_w, BER). \quad (7.2)$$

Here, reward R is a function of communication power P_w and BER value BER as the state vectors. The relation between state variables and reward value is presented later.

The next state and the current state can be the same depending on the workload characteristic. The reward R forms a part of the expected Q-value, which decides the direction of state transition. The optimal estimation is chosen among the set of states to satisfy the required criteria by taking the corresponding action selected from the formed LUT. The expected Q-value is calculated as

$$\hat{Q}(s_i, a_k) = Q(s_i, a_k)(1 - \alpha) + \alpha(R + \gamma E). \quad (7.3)$$

Here α , γ denotes the learning rate and discount factor respectively. The optimal estimation E of state s_i can be calculated as follows

$$E = \min\{\hat{Q}(s_i, a_k)\}, k = 0, \dots, M. \quad (7.4)$$

Here, $\hat{Q}(s_i, a_k)$ represents the expected Q-value after taking action a_k ; M denotes the number of possible actions available at state s_i . The optimal estimate can be *min* or *max* depending on the reward function.

The 2.5D adaptive TSI I/O verification is performed in Cadence Virtuoso (Ultrasim-Verilog) and Matlab. An 8-core MIPS microprocessor with 8-bank of SRAM memory is designed with GF 65nm CMOS. The 2.5D TSI T-line is of length 3mm and 10μm width, driven by the CML buffer. The power traces are measured from Cadence Virtuoso and control cycle is set as 1ns, larger than switching time of I/O controller. The I/O management controller is based on the basic Q-learning and reinforcement Q-learning output to balance the I/O communication power and BER at receiver. The look-up-table (LUT) is designed with I/O communication power and BER, stored in LUT for adaptively tuning the output-voltage swing. A LUT is set up as follows: (100mV, 6.27E – 2mW, 7.03E – 2BER), (150mV, 1.41E – 1mW, 1.35E – 2BER), (200mV, 2.51E – 1mW, 1.61E – 3BER), and (300mV, 5.64mE – 1mW, 4.93E – 6BER). This LUT is almost robust, since this depends on characteristics of the circuit rather than the application. The learning rate α and discount factor γ are set as 0.5 and 0.9 respectively. Auto-regression (AR) of order 8 is used for load current (or I/O communication power) prediction. The error between the predicted and actual values is less than 0.3%. The circuit is designed in Cadence with the according technology PDK. The overall I/O performance can provide a minimum of 76mV peak-to-peak signal swing with 4Gb/s bandwidth, and the power consumption is only 12.5mW when implemented in 65nm CMOS process. The adaptive self-tuning of output-voltage swing may come with a little area overhead of 0.03mm² for additional control circuits and a latency of 100-200ps.

Figure 7.3: State transition based on conventional Q-learning

7.3.1.2 Adaptive I/O Voltage-swing Tuning

The self-adaptive tuning of the output-voltage swing at the CML buffer is performed based on the conventional Q-learning. The I/O communication power P_{w_i} and BER BER_i corresponding to output-voltage swing level V_i are considered as the components of state vectors S and output-voltage swing level as the action variable.

Figure 7.3 illustrates an example depicting the change of states by the conventional Q-learning algorithm. In this example, four states and five actions are considered. When the system is in state s_1 (s_2) and action a_1 (a_2) is selected, then the system changes to state s_2 (s_1); when the action a_3 is chosen, the state changes to s_3 (s_4) and remains in same state when action a_5 is chosen. Similarly, other state transitions also happens. One needs to note that the state transition depends on the action chosen at the current state.

Algorithm 4 Conventional Q-learning based adaptive tuning of output-voltage swing

Require: Communication power trace P_i , BER feedback from receiver and look-up-table (LUT)

Ensure: Adaptive tuning of output-voltage swing V_i

- 1: Predict tail current: $I_t(k+1) = \sum_{i=0}^{N-1} w_i I_t(k-i) + \xi$
 - 2: Calculate corresponding communication power and BER
 - 3: Reward: $R_w(s_i, a_k, s_{i+1}) = b_1 \Delta(P_{w_i}) + b_2 \Delta(BER_i)$
 - 4: $\hat{Q}(s_i, a_k) \leftarrow Q(s_i, a_k)(1 - \alpha) + \alpha(R_w + \gamma E)$
 - 5: Optimal value estimate: $E = \min\{\hat{Q}(s_i, a_k)\}, j = 0, \dots, M$
 - 6: By adjusting tail current using control bits, tune corresponding V_i
-

A self-adaptive output-voltage swing tuning by conventional Q-learning is presented in Algorithm 4. LUT is formed with output-voltage swing as action and the I/O communication power and BER as the state vectors. The tail current of the CML buffer is firstly predicted by auto-regression, as given in Line 1 of Algorithm 4.

Figure 7.4: Flowchart showing conventional Q-learning based self-adaptive voltage swing tuning

Based on the predicted tail current, corresponding I/O communication power and the BER are calculated. Further, using the present I/O communication power and BER values, reward R_w is calculated. Since we have two factors, we consider the weighted sum of I/O communication power and BER. The reward function is given as follows

$$R_w(s_i, a_k, s_{i+1}) = b_1 \Delta(Pw_i) + b_2 \Delta(BER_i) \quad (7.5)$$

here b_1 and b_2 denote the weighted coefficients for normalized rewards of the communication power $\Delta(Pw_i)$ and BER $\Delta(BER_i)$. This calculation of reward is given in Line 3 of Algorithm 4. After calculating the reward, the expected Q-value is calculated from Line 4 of Algorithm 4, and the optimal action is selected based on Q-values, as in Line 5 of Algorithm 4. This is how the adaptive tuning is performed by implementing the conventional Q-learn algorithm. As a summary, the whole flow of adaptive tuning by the conventional Q-learning algorithm is shown in Fig. 7.4.

An example of adaptive tuning of output-voltage swing by conventional Q-learning is shown in Figure 7.5. Initially, the voltage-swing for the next control-cycle is predicted as $294mV$. As it is impractical to have an LUT with continuous values, we map the incoming value to the closest value in LUT. Thus, the voltage-swing $294mV$ is fit to state s_4 because of the nearest voltage level. Based on the predicted value and the current value, reward is calculated, as in (7.5). Further the corresponding voltage-swing level is obtained based on the calculated reward and optimal estimate function. This results in setting the voltage-swing level (action) to $200mV$ with $0.25mW$ communication power i.e., state s_3 . It needs to be noted that the reduction in voltage swing level is due to the

Figure 7.5: Example of one adaptive I/O control by basic Q-learning

tolerance to BER. Further, the BER and I/O communication power are updated.

7.3.1.3 Accelerated Q-learning

The conventional Q-learning algorithm [185] converges to the optimal after unlimited or large number of iterations that may be too slow for convergence [186]. To overcome this convergence issue, an accelerated Q-learning for adaptive tuning of output-voltage swing can be performed to achieve low power and faster convergence. Here, an example to illustrate the difference between conventional Q-learning and accelerated Q-learning is presented, followed by the modeling of a MDP and the according reinforcement Q-learning algorithm for the adaptive tuning.

One example with 4-state is shown in Figure 7.6. For state s_1 , action a_1 can change its state to state s_2 with probability $P(s_1, a_1, s_2)$; For state s_2 , action a_2 can change its state to state s_1 with probability $P(s_2, a_2, s_1)$. Whereas action a_5 causes no change in state, whose probability is given as $P(s_2, a_5, s_2)$. The state transition probability P is given by a decaying function. The probability under the decaying function is given by $P = 1 - 1/(\log(N_{s_i} + 2))$ with N_{s_i} denoting the number of visits to state s_i . The transition probability based action will ensure the visit to all states at starting period. This will calculate Q-value to every available state accordingly. After this, Q-value based action will dominate and the optimal action with smallest Q-value will be selected.

To find the optimal value of MDP, probability based action selection, accelerated Q-learning algorithm can be utilized to evaluate the pair of state and action as the Q-value.

Figure 7.6: State transition based on accelerated Q-learning

conventional Q-learning algorithm converges to the optimal value after unlimited number of iterations [186]. Accelerated Q-learning [179] can be utilized to find an optimal solution with a faster convergence based on the predicted next state and the according transition probability with an initialized random action at first few states. Initial random actions make the system explore environment faster and more easily find optimal states. To achieve a faster convergence, the transition probability is utilized to select the action instead of directly selecting the next state. Further, the reinforcement Q-learning can be utilized to find the optimal point for the modeled MDP to solve the discussed adaptive tuning problem. Accelerated Q-learning is presented in Algorithm 5.

The first phase is initialization to form a LUT with states and actions. In addition, the transition probability P for all the states is set as 1 and the reward is set to a maximum value L . This process of initialization is presented as $Init()$ of Algorithm 5.

Prediction of the next state is performed to obtain the corresponding action. In the action selection phase, given by $Selection()$, the Q-value for the state and action pair is found iteratively, where the expected Q-value is given by

$$\hat{Q}(s_i, a_k) = (1 - \alpha) * Q(s_i, a_k) + \alpha * \text{delta} \quad (7.6a)$$

$$\text{delta} = R(s_i, a_k, s_{i+1}) + \gamma * \min_{a \in A} (Q(s_i, a_k)). \quad (7.6b)$$

where $Q(s_i, a_k)$ represents the accumulated Q-value and $\hat{Q}(s_i, a_k)$ represents the expected Q-value after taking action. after taking the action a_k to the next state.

In each iteration, the action is selected either based on the transition probability or based on the minimum Q-value (or policy). If the transition probability is larger than the threshold, a random action is selected; otherwise, the action policy with the minimum Q-value is selected. The random action will happen at the first few rounds to explore the design space. As the learning process continues, the policy action with the calculated

Algorithm 5: Accelerated Q learning algorithm**Input:** Communication power P_w , BER feedback**Output:** Output-voltage**function** Init() $1 \rightarrow P(s_i, a_k, s_{i+1})$ Reward $R(s_i, a_k, s_{i+1}) = L$ $v_{predict} \rightarrow V_{s_i}$

Selection()

end function**function** Selection() **for** $k = 1 : n$ $V_{s_i}, BER_i \rightarrow s_i \in S$ $\hat{Q}(s_i, a_k) \leftarrow (1 - \alpha) * Q(s_i, a_k) +$ $\alpha * (R(s_i, a_k, s_{i+1}) + \gamma * \min(Q(s_i, a_k)))$ **If** $P(s_i, a_k, s_{i+1}) > rand(0, 1)$ $a_k \leftarrow rand(A)$ **else** $a_k \leftarrow \min(\hat{Q}(s_i, a_k))$ **end if**

Update()

end for**end function****function** Update() Reward: $R(s_i, a_k, s_{i+1}) = b_1 \Delta(P_i) + b_2 \Delta(BER_i)$ Update Policy (s_i, a_i), based on new Q $\forall s_i \in S \{$ $a_k \leftarrow rand(A)$ $Q(s_i, a_k) = \hat{Q}(s_i, a_k)$ $P(s_i, a_k, s_{i+1}) = 1 - \frac{1}{\log(N_{s_i} + 2)}$ $\}$ **end function**

Q-value will dominate and become more accurate to use. As such, a higher probability exists that the action a_k with the minimum Q-value will be selected. The policy action with the minimum Q-value can be described as below

$$a_k \leftarrow \min(\hat{Q}(s_i, a_k)). \quad (7.7)$$

Lastly, the *Update()* phase is activated at the end of each iteration of *Selection()* function. The reward is defined as the weighted value of BER and P_w and updated as given in (7.5).

At the end of *Update()*, each state will be randomly visited and Q-value will be

Figure 7.7: Flowchart of accelerated Q-learning based self-adaptive voltage swing tuning

updated accordingly as given in (7.6). The transition probability $P(s_i, a_k, s_{i+1})$ is also updated as N_{s_i} (increases with the number of visits to state s_i) after each iteration.

Note that with the prediction of states s_i as given in function *Init()* and *Update()* and the transition probability, the convergence to the optimal solution is accelerated [180]. This is done at the end of each round with the random action a_k to visit the state s_i . In brief, the whole flow of adaptive tuning by the accelerated Q-learning is presented as a flowchart in Figure 7.7.

The LUT can be implemented online with the corresponding control bits calculated and feedback to CML buffer to tune the DAC current of the CML buffer. Note that LUT can be implemented in the hardware with multiple AND/OR partial matching logic circuit instead of read only memory (ROM). This LUT based implementation has higher speed and low power consumption compared to the ROM.

An example of one adaptive tuning by the accelerated Q-learning is shown in Fig. 7.8. Initially, the voltage-swing for the next control-cycle is predicted as $294mV$. The voltage-swing $294mV$ with power $0.51mW$ fits to the state s_4 because of the nearest voltage level. The action will be selected based on the probability check as in *Selection()* of Algorithm 2. As shown in Fig. 7.8, the solid red line and red dotted line indicate the policy-based action selection and probability-based action selection, respectively. For the probability-based action selection (red dotted line), a random action is selected to visit any of the available states. This takes place with high probability at the starting period to ensure the visit to all the available states, where its Q-value will be updated accordingly. Afterwards, the action with minimum Q-value will dominate and be considered as the optimal action. As Fig. 7.8 shows, action a_4 is eventually selected leading to the voltage level associated with state s_2 . Afterwards, the new voltage-swing

Figure 7.8: Example of one adaptive I/O control by accelerated Q-learning.

is assigned based on the state s_2 as $0.141mW$. As such, the driver tail-current is tuned to have the output voltage-swing as $150mV$. Lastly, the reward $R(s_4, a_4, s_2)$ will be updated based on the feedback of BER and power obtained at Rx. One needs to note that as shown in Fig. 7.5, conventional Q-learning selects state s_3 due to action based state change, however it may converge to state s_2 after few iterations.

7.3.1.4 Comparison of Conventional and Accelerated Q-learning

The I/O communication power saving is verified for various SPEC benchmarks [128] in Table 7.2 by the self-adaptive tuning using: no Q-learning (normal); the conventional Q-learning (conv); and the accelerated Q-learning (acc). It shows that the accelerated Q-learning algorithm is more power efficient with faster convergence. For example, for *bzip2* benchmark, the I/O communication power without the adaptive tuning is $0.267mW$, which is reduced to $0.224mW$ when adaptively tuned by the conventional Q-learning; and is further reduced to $0.210mW$ with the accelerated Q-learning. On average, the power consumption of the system (transmitter, receiver and the I/O) is $19mW$ with an energy efficiency of $4.75pJ/bit$ for the I/O without the adaptive tuning, which is further reduced to $12.5mW$ by the adaptive tuning. On average, the power saving of 18.89% and energy efficiency improvement of 15.11% is achieved by the adaptive tuning based on the reinforcement Q-learning; and 12.95% of power saving and 14% of energy-efficiency improvement are achieved when using the conventional Q-learning. What is more, on average, accelerated Q-learning takes $0.091s$ for convergence, whereas

Table 7.2: Power and run-time comparisons for the adaptive tuning with Q-learning under various benchmarks

Benchmark	Communication power (<i>mW</i>)			Power saving		Run time(s)		
	Acc	Conv	Normal	Acc	Conv	Acc	Conv	<i>Improvement</i>
ammp	0.231	0.250	0.296	22.11%	15.54%	0.0804	0.1049	23.28%
applus	0.474	0.494	0.555	14.64%	10.99%	0.0817	0.1051	22.25%
apsi	0.652	0.674	0.744	12.34%	9.41%	0.0817	0.1051	22.25%
art	0.199	0.216	0.255	22.01%	15.29%	0.0829	0.1057	21.55%
bzip2	0.210	0.224	0.267	21.19%	16.10%	0.0829	0.1056	21.50%
crafty	0.381	0.419	0.461	17.39%	9.11%	0.0822	0.1061	22.53%
eon	0.234	0.253	0.298	21.32%	15.10%	0.0807	0.1051	23.24%
equake	0.232	0.249	0.295	21.07%	15.59%	0.0821	0.1047	21.60%
facerec	0.333	0.364	0.409	18.53%	11.00%	0.0833	0.1060	21.42%
fft	0.153	0.179	0.199	23.12%	10.05%	0.2002	0.2309	13.30%
file transfer	0.512	0.554	0.598	14.58%	7.36%	0.2057	0.2485	17.22%
fma3d	0.539	0.561	0.623	13.35%	9.95%	0.0850	0.1083	21.51%
galgel	0.241	0.261	0.307	21.56%	14.98%	0.0820	0.1069	23.29%
gap	0.284	0.310	0.355	19.86%	12.68%	0.0808	0.1036	22.01%
gcc	0.507	0.536	0.587	13.60%	8.69%	0.0838	0.1064	21.24%
gzip	0.238	0.257	0.302	21.08%	14.90%	0.0817	0.1051	22.26%
lucas	0.518	0.548	0.602	13.90%	8.97%	0.0818	0.1053	22.32%
mcf	0.248	0.264	0.307	19.41%	14.01%	0.0817	0.1056	22.63%
mesa	0.228	0.243	0.287	20.45%	15.33%	0.0846	0.1065	20.56%
mgrid	0.274	0.297	0.346	20.83%	14.16%	0.0818	0.1051	22.17%
parser	0.429	0.458	0.508	15.43%	9.84%	0.0825	0.1084	23.89%
perlbmk	0.207	0.226	0.266	22.21%	15.04%	0.0822	0.1052	21.86%
sixtrack	0.341	0.357	0.412	17.39%	13.35%	0.0819	0.1065	23.10%
swim	0.242	0.262	0.310	21.82%	15.48%	0.0812	0.1052	22.81%
twolf	0.216	0.235	0.279	22.38%	15.77%	0.0839	0.1037	19.09%
vortex	0.225	0.239	0.288	21.94%	17.01%	0.0832	0.1056	21.21%
vprs	0.240	0.255	0.300	19.96%	15.00%	0.0826	0.1059	22.00 %
wupwise	0.424	0.442	0.502	15.39%	11.95%	0.0836	0.1058	20.98%
Average	-	-	-	18.89 %	12.95%	0.0912	0.1152	20.83%

conventional Q-learning takes 0.115s. The reported time includes the training of samples. The run time has improved with accelerated Q-learning by an average of 20.83% compared to conventional Q-learning based I/O management as shown in Table 7.2

7.3.2 Adaptive I/O Circuit Design

In this section, we present the system model for power and BER. This is followed by the design of transmitter, receiver and the error check block.

7.3.2.1 System Model

The prediction of power and BER of I/O communication channel and their models are discussed here. The first component of the state vector is the I/O communication power. The system power model includes the I/O communication power of driver and the TSI T-line. Both are the functions of the output-voltage swing V_i . For the CML based driver with TSI T-line [70], I/O communication power is given by

$$P_{wi} = V_i \cdot (I_t + \frac{\eta \cdot V_{dd} \cdot s}{R_D + Z_{diff}} \cdot f). \quad (7.8)$$

Here I_t is driver tail current; s is duration of signal pulse; η is activity factor; R_D is the resistance of driver; and Z_{diff} is the characteristic impedance of the TSI T-line.

The tail current I_t at the current control cycle is set by analog design and can be obtained from measurement. Tail current for the next control cycle can be predicted by auto-regression (AR) as

$$I_t(k+1) = \sum_{i=0}^{M-1} w_i I_t(k-i) + \xi. \quad (7.9)$$

Here $I_t(k+1)$ denotes the predicted tail current at $(k+1)$ -th control cycle; w_i represents the auto-regression coefficient; ξ is the prediction error and M represents the order of the AR prediction. Based on the predicted tail current, the I/O communication power for next control cycle is calculated as $P_{wi}(k+1)$.

The second component of the state vector is BER of I/O communication, which is feedback from the receiver. BER depends on the output-voltage swing, external noise, channel noise and some other parameters [191]. In a wire-line communication system [192], the BER can be estimated with the dependence on the output-voltage swing as

$$BER_i = \frac{1}{2} \operatorname{erfc}\left(\frac{V_i}{\sqrt{2}\sigma_v}\right). \quad (7.10)$$

Figure 7.9: (a) Transmitter with 8:1 serializer; (b) adaptive tuning of driver tail current; (c) TSI realized by a T-line

Here $erfc$ is the complementary error function; V_i refers to the i -th output-voltage swing level and σ_v is the standard deviation of the noise.

As such, BER can be obtained from the ECC at the receiver by counting errors during a period. During the learning process, based on the obtained BER at ECC under one output-voltage swing, the standard deviation of noise, σ_v is estimated from (7.10).

7.3.2.2 Transmitter and Receiver

In details, the Tx employs a 8:1 serializer to convert 8-bit parallel data into serial data as shown in Figure 7.9(a). Four digital D flip-flops are implemented as a shift-register chain for each of the odd (D_1, D_3, D_5, D_7) and even (D_0, D_2, D_4, D_6) bits of data. This is followed by a 2:1 MUX to combine them altogether. A current-mode logic (CML) output driver is used to drive the TSI T-line from the Tx to the Rx on the common substrate. The CML output stage is powered by the fixed supply (1.2V). The I/O communication power P_w depends on the output-voltage swing and the tail current of the driver. one can generate control bits to tune the tail current of the CML driver and alter the output-voltage swing, as shown in Figure 7.9(b).

What is more, compared to the traditional serial I/Os based on the backplane PCB trace [99, 100], the 2.5D TSI I/Os do not need the complex equalizer circuits at the receiver due to the small signal loss in the TSI T-line channel. A sampler at the front-

Figure 7.11: Encoding and decoding at transmitter and receiver

Figure 7.12: (a) Data encoding and decoding; (b) transmitted data format

C to generate parity bit vector P as shown in Figure 7.12(a), where the parity generator consists of a set of *AND* and *XOR* gates. Thus, the total encoded data to be transmitted will be 40-bit for every 32-bit of data. One MUX is implemented for serial transmission. The data format of the transmitted data is presented in Figure 7.12(b).

At the receiver end, the 32-bit data is stored in the input FIFO and the last 7-bits of 8-bit (parity) is utilized for error checking and correction. The checking result vector (R) is generated from the parity code P . By summing the result vector, one can detect if any bit is wrong and a left-shifter is used to correct bit error. The current implementation of ECC has capability to correct 1-bit error but detect multiple bit errors. It can be used to obtain the historical BER at the receiver and is further feedback to the transmitter. BER is the ratio of total number of error bits found to the total number of bits transmitted.

As shown in Figure 7.9(b), the CML driver with variable current source is set by the DAC current and load resistor. The DAC tail current source is composed of a group of current sources connected in parallel with switches controlled by the control bits

Figure 7.13: Eye diagram of output data with different driver current (or output-voltage swing) levels: (a) $2mA$; (b) $3mA$; (c) $4mA$ based on simulations

generated from the I/O controller. When the driver tail current is varied, the output-voltage swing will change. Generally, the load resistor is set 50Ω for the TSI T-line impedance matching. Here the tail current source is varied from $2mA$ to $5mA$.

7.3.2.4 Voltage-swing Tuning Results

The characteristics of eye-diagrams with the driver currents are presented in Figure 7.13 by introducing 10% clock cycle-to-cycle jitter (noise) at the TSI I/O channel. Note that different driving currents can make different eye openings under the noise in channel. A larger eye opening is associated with a higher current driving ability (or a larger output-voltage swing), which has a minimum opening of $76mV$ amplitude and $77ps$ timing margin with $2mA$ driver current and further increases with the driving current. One needs to observe that with the increase in driver current BER decreases, but at the cost of power. As BER does not need to be low at all times, one can leverage the trade-off between the power reduction and the necessary BER.

We further study the eye-diagram under the control of adaptive tuning is performed to verify the functionality of the adaptive I/O circuit tuning. Figure 7.14 shows the current consumption under different levels of the output-voltage swing. The sources of error are introduced in three stages: stage 1 is to introduce 20% of clock jitter; stage 2 is to add additional 10% receiver offset; and stage 3 is to further add additional 10% power supply noise. As discussed previously, with the increase in noise, the tail current at the CML buffer is varied to tune the output-voltage swing. For example, for stage 1, which has only clock jitter, the current is increased to $5mA$ to increase the eye-opening as $(103mV, 62ps)$. With the increase in noise i.e., for stage 3, the current is increased adaptively. The difference in eye-diagrams with tuning the output-voltage swing and without tuning the output-voltage swing is shown in Figure 7.14. One can observe that for stage 3, without tuning the tail current, the eye opening is $96mV$, but the eye opening increases to $112mV$ by adaptively changing the current. Similar improvement in eye openings is shown for other stages as well.

Figure 7.14: The eye-diagrams under adaptive current (or power) adjustment by output-voltage swing tuning

7.4 Summary

In this chapter, an adaptive voltage swing tuning at I/O is demonstrated for 2.5D memory-logic integrated microprocessor. It is observed that the communication power increases with voltage-swing level and BER reduces with voltage-swing level. Hence, a trade-off between these parameters is required to save communication power with better BER. To reduce the I/O communication power, an adaptive tuning of output-voltage swing is adapted at the I/O. Conventional Q-learning based adaptive voltage swing tuning is employed first and a reinforcement Q-learning based voltage swing tuning is further employed. When compared to the uniform output-voltage swing based I/O management, the I/O managements by basic Q-learning and the reinforcement-based Q-learning can achieve 12.95% and 18.89% power reduction and 14% and 15.11% energy efficiency improvement, respectively. An accelerated Q-learning based voltage-swing tuning to enhance the convergence and save the communication power is the major contribution of the respondent author and the work is published in TC'15.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

To conclude, this thesis has shown a thorough study of 3D and 2.5D I/O designs towards thousand-core on-chip microprocessor stemming from device level, to circuit level, and all the way to the system level. At the device level, we performed modeling for 3D through-silicon via (TSV) and 2.5D through-silicon interposer (TSI) interconnects based on a detail study of physical characteristics. This in-depth study of physical composition of TSV/TSIs revealed the coupling of electrical and mechanical characteristics on temperature due to the existence of liner material around metal-fill. As a case study, a 3D clock-tree design is considered and a nonlinear optimization based TSV insertion is performed to minimize the clock-skew. For various 3D clock-tree benchmarks, insertion of dummy TSVs in an optimized manner can reduce clock-skew by 61.3% on average using the proposed nonlinear electrical-thermal-mechanical delay model. From the circuit level, design of low-power CML and LVDS buffers with pre-emphasis stage is proposed for TSI I/Os and post-layout simulation results have shown an energy efficiency of 0.075pJ/bit and 0.48pJ/bit , respectively.

From the system level, we mainly dealt with the power I/O and signal I/O managements. Firstly, power I/O management for 3D/2.5D many-core microprocessor with on-chip power converters is explored. A dynamic-voltage scaling (DVS) power I/O management by space-time multiplexing is proposed. Cores are clustered first based on the magnitude of power-signatures and corresponding power converters are allocated. Further, inside a cluster, power converters are shared in time, based on the power-signature phase. Thus, the total number of power converters required are reduced along with demand-supply matching. In addition to the resource (power converter) allocation, workload balancing is performed to avoid overloading of power converters. On average,

40.53% power saving is achieved with the proposed space-time multiplexing based power I/O management with 42.86% reduction for the required number of power converters for a 64-core microprocessor.

From the signal I/O perspective, bandwidth management and voltage-swing tuning for core-memory communication are addressed. A dynamic memory controller design is proposed to enhance the bandwidth utilization of available I/Os. To perform bandwidth management with less complexity, cores are classified based on their memory-access demand magnitudes. Further, based on the memory-access priority, I/O allocation is carried out. The configuration of I/O allocation changes dynamically based on the memory-access data-pattern characteristics. This space-time multiplexing scheme is deployed inside the memory controller. The proposed reconfigurable data-pattern aware memory controller achieves nearly 58.85% bandwidth balancing with 11.90% QoS improvement. Lastly, to reduce the I/O communication power between core and memory blocks in a many-core integrated system, the output-voltage swing tuning is proposed. Conventional Q-learning algorithm is firstly deployed to tune the output-voltage swing with communication power and bit-error rate (BER) constraints. To address the convergence issue in conventional Q-learning, an accelerated Q-learning for output-voltage swing tuning is deployed. With the use of conventional Q-learning and further accelerated Q-learning, 12.95% and 18.89% power reduction and 14% and 15.11% energy-efficiency improvement is achieved compared to the use of the uniform output-voltage swing based I/O communication.

8.2 Recommendations for Future Work

Based on above works, there are a few recommended future works for this thesis.

The first recommended work is to build a 2.5D integrated system with multi-core chip, external memories and accelerators. The first task will be to design a pipelined microprocessor to perform faster computations on the data. Secondly, hardware accelerators needs to be designed to carry out some special operations such as floating point operations, video processing etc., to improve the performance of the chip. Traditional memory can be used or faster flash memories can be designed for higher memory access. Based on the designed blocks, a 2.5D TSI I/O link design has to be carried out have an integrated system as shown in Figure 8.1.

As a first step, we have completed the design of microprocessor, memory and accelerator blocks using Global Foundry (GF) 65nm process. This work has been published in [115], the chip without TSI I/Os is shown in Figure 8.2. Design of TSI I/O links have to be carried out. The total number of microbumps from the microprocessor block is

Figure 8.1: Overall system architecture for multi-core memory 2.5D integrated micro-processor.

Figure 8.2: Fabricated multi-core processor with memory and accelerator blocks.

nearly 400 and 4 accelerators and 8 memory blocks needs to be integrated. Thus, the future work is to design TSI I/Os to form a whole system.

Secondly, to achieve thousand core on-chip integration, multiple 3D ICs (each with memory and logic layers) can be integrated by 2.5D TSI I/Os. This hybrid integration can successfully integrate huge number of cores and memory blocks with high-speed, low power interconnects with relaxed thermal reliability concern.

Appendix A

Publication list

A.1 Journal

1. Sai Manoj P D, Hao Yu, and Kanwen Wang, 3D Many-core Microprocessor Power Management by Space-Time Multiplexing based Demand-supply Matching, *IEEE Transactions on Computers (TC)*, (DOI: 10.1109/TC.2015.2389827) (In press).
2. Sai Manoj P D, Hao Yu, Hantao Huang, and Dongjun Xu, A Q-Learning based Self-adaptive I/O Communication for 2.5D Integrated Many-core Microprocessor and Memory, *IEEE Transactions on Computers (TC)*, (DOI: 10.1109/TC.2015.2439255) (In press).
3. Sai Manoj P D, Dongjun Xu, Hao Yu, Kanwen Wang, Ningmei Yu, and Mingbin Yu, A 2.5D Memory-logic Integration with Data-pattern Aware Memory Controller, *IEEE Design and Test*, (DOI: 10.1109/MDAT.2015.2440413) (In press).
4. Leibin Ni, Sai Manoj P D, Yang Song, Chenjie Gu, and Hao Yu, A Zonotoped Macromodeling for Eye-diagram Verification in High-speed I/O Links with Jitter and Parameter Variations, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, (Accepted).
5. Yang Song, Hao Yu, and Sai Manoj P D, Reachability-based Robustness Verification and Optimization of SRAM Dynamic Stability under Process Variations, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol.33, no.4, pp585-598, April 2014.
6. Sai Manoj P D, Hao Yu, Yang Shang, Chuan Seng Tan, and Sung Kyu Lim, Reliable 3D Clock-tree Synthesis Considering Nonlinear Capacitive TSV Model with Electrical-thermal-mechanical Coupling, *IEEE Transactions on Computer-Aided*

A.2 Conference

1. Jie Lin, Shikai Zhu, Zhiyi Yu, Dongjun Xu, *Sai Manoj P D*, and Hao Yu, A Scalable and Reconfigurable 2.5D Integrated Multicore Processor on Silicon Interposer, *IEEE Custom Integrated Circuits Conference (CICC)*, September 2015.
2. *Sai Manoj P D*, Kanwen Wang, Hantao Huang and Hao Yu, Smart I/Os: A Data-pattern Aware 2.5D Interconnect with Space-Time Multiplexing, *IEEE/ACM System Level Interconnect Prediction (SLIP)*, June 2015.
3. Shunli Ma, Janyuan Ren, *Sai Manoj P D*, Hao Yu and Roshan Weeraskera, A 9.8Gbps, 6.5mW Forwarded-clock Receiver with Phase Interpolator and Equalized Current Sampler in 65nm CMOS, *IEEE International Microwave Symposium (IMS)* (Student Paper Competition Finalist), June 2015.
4. *Sai Manoj P D*, Hao Yu, Chenjie Gu, and Cheng Zhuo, A Zonotoped Macromodeling for Reachability Verification of Eye-diagram in High-speed I/O Links with Jitter, *IEEE/ACM International Conference on Computer-Aided-Design (ICCAD)*, November 2014.
5. Hantao Huang, *Sai Manoj P D*, Dongjun Xu, Hao Yu, and Zhigang Hao, Reinforcement Learning based Self-adaptive Voltage-swing Adjustment of Through-silicon Interposer I/Os for Many-core Microprocessor and Memory Communication, *IEEE/ACM International Conference on Computer-Aided-Design (ICCAD)*, November 2014.
6. Dongjun Xu, *Sai Manoj P D*, Hantao Huang, Ningmei Yu, and Hao Yu, An Energy-efficient 2.5D Through-silicon Interposer I/O with Self-adaptive Adjustment of Output-voltage Swing, *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, August 2014.
7. Sih-Sian Wu, Kanwen Wang, *Sai Manoj P D*, Tsung-Yi Ho, and Hao Yu, A Thermal Resilient Integration of Many-core Microprocessor and Main Memory by 2.5D TSI I/Os, *ACM/IEEE Design Automation and Test Conference in Europe (DATE)*, March 2014.

8. Yang Song, *Sai Manoj P D*, and Hao Yu, Zonotope-based Nonlinear Model Order Reduction for Fast Performance Bound Analysis of Analog Circuits with Multiple-interval-valued Parameter Variations, *ACM/IEEE Design Automation and Test Conference in Europe (DATE)*, March 2014.
9. Yang Song, *Sai Manoj P D*, and Hao Yu, A Robustness Optimization of SRAM Dynamic Stability by Sensitivity-based Reachability Analysis, *IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 2014.
10. Jiacheng Wang, Shunli Ma, *Sai Manoj P D*, Mingbin Yu, Roshan Weerasekera, Hao Yu, High-Speed and Low-Power 2.5D I/O Circuits for Memory-logic-integration by Through-Silicon Interposer, *IEEE International 3D System Integration Conference (3DIC)*, October 2013.
11. *Sai Manoj P D*, Kanwen Wang, and Hao Yu, Peak Power Reduction and Workload Balancing by Space-time Multiplexing based Demand-supply Matching for 3D Thousand-core Microprocessors, *ACM/IEEE Design Automation Conference (DAC)*, June 2013.
12. *Sai Manoj P D*, and Hao Yu, Cyber-Physical Management for Heterogeneously Integrated 3D Thousand-core On-chip Microprocessor, *IEEE International Symposium on Circuits and Systems (ISCAS)*, (Invited special session) May 2013.
13. Yang Song, Hao Yu, *Sai Manoj P D* and Guoyong Shi, SRAM Dynamic Stability Verification by Reachability Analysis with Consideration of Threshold Voltage Variation, *ACM International Symposium on Physical Design (ISPD)*, March 2013.

A.3 Workshop/Poster

1. Ni Leibin, *Sai Manoj P D*, Yang Song, and Hao Yu, A Zonotoped Reachability Verification for High-speed I/O Links, *International Workshop on Design Automation for Analog and Mixed-signal Circuits*, November 2014.
2. *Sai Manoj P D*, Leibin Ni, Hao Yu, Chenjie Gu, and Cheng Zhuo, A Zonotoped Macromodeling for Reachability Verification of Eye-diagram in High-speed I/O Links with Jitter, *ACM/IEEE Design Automation Conference (DAC)* (Work-in-progress session, poster), June 2014.

3. Kanwen Wang, *Sai Manoj P D*, and Hao Yu, A Data-pattern Aware Memory Controller with Space-Time Multiplexing for 2.5D Shared-Memory Many-core Microprocessors, *ACM/IEEE Design Automation Conference (DAC)* (Work-in-progress session, poster), June 2014.

A.4 Miscellaneous

1. Participant of Ph.D. Forum in *ACM/IEEE Design Automation Conference (DAC)*, June 2014.
2. A. Richard Newton Young Research Fellow Award in *ACM/IEEE Design Automation Conference (DAC)*, June 2013.

Bibliography

- [1] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar, “An 80-Tile 1.28TFLOPS network-on-chip in 65nm CMOS,” in *Int. Solid-State Circuits Conf. (ISSCC)*. IEEE, 2007.
- [2] “ITRS Roadmap 2013,” <http://www.itrs.net/Links/2013ITRS/2013Chapters/2013ExecutiveSummary.pdf>.
- [3] M. Taylor, J. Psota, A. Saraf, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, A. Agarwal, W. Lee, J. Miller, D. Wentzlaff, I. Bratt, B. Greenwald, H. Hoffmann, P. Johnson, and J. Kim, “Evaluation of the raw microprocessor: an exposed-wire-delay architecture for ILP and streams,” in *Int. Symp. on Computer Architecture (ISCA)*, 2004.
- [4] J. Howard, S. Dighe, S. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, and R. Van Der Wijngaart, “A 48-core IA-32 processor in 45nm CMOS using on-die message-passing and DVFS for performance and power scaling,” *IEEE J. of Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, Jan 2011.
- [5] M. Healy, K. Athikulwongse, R. Goel, M. Hossain, D. H. Kim, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, M. Jung, B. Ouellette, M. Pathak, H. Sane, G. Shen, D. H. Woo, X. Zhao, G. Loh, H.-H. Lee, and S. K. Lim, “Design and analysis of 3D-MAPS: a many-core 3D processor with stacked memory,” in *IEEE Custom Integrated Circuits Conf. (CICC)*, 2010.
- [6] J. Kong, J. Choi, L. Choi, and S. W. Chung, “Low-cost application-aware DVFS for multi-core architecture,” in *Int. Conf. on Convergence and Hybrid Information Technology*, 2008.

- [7] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, 2008.
- [8] R. David, P. Bogdan, R. Marculescu, and U. Ogras, "Dynamic power management of voltage-frequency island partitioned network-on-chip using intel's single-chip cloud computer," in *ACM/IEEE Int. Symp. on Network on Chip (NoCs)*, 2011.
- [9] K. Zoschke, J. Wolf, C. Lopper, I. Kuna, N. Jurgensen, V. Glaw, K. Samulewicz, J. Roder, M. Wilke, O. Wunsch, M. Klein, M. Suchodoletz, H. Oppermann, T. Braun, R. Wieland, and O. Ehrmann, "TSV based silicon interposer technology for wafer level fabrication of 3D SiP modules," in *IEEE Electronic Components and Technology*, 2011.
- [10] J. Cubillo, R. Weerasekera, Z. Z. Oo, E.-X. Liu, B. Conn, S. Bhattacharya, and R. Patti, "Interconnect design and analysis for through silicon interposers (TSIs)," in *IEEE Int. 3D Systems Integration Conf. (3DIC)*, 2012.
- [11] D. Patterson, M. Kelly, R. Reed, S. Eplett, Z. Kutlu, and R. Alapati, "2.5D TSV enablement of a multiple processor SiP," *Chipscale review*, vol. 18, no. 3, pp. 26–34, May 2014.
- [12] S. Park, M. Qazi, L.-S. Peh, and A. P. Chandrakasan, "40.4fJ/bit/mm low-swing on-chip signaling with self-resetting logic repeaters embedded within a mesh NoC in 45nm SOI CMOS," in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2013.
- [13] I. Loi and L. Benini, "An efficient distributed memory interface for many-core platform with 3D stacked DRAM," in *ACM/IEEE Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, 2010.
- [14] C. Weis, N. Wehn, L. Igor, and L. Benini, "Design space exploration for 3D-stacked DRAMs," in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2011.
- [15] M. Chen, J. Silva-Martinez, M. Nix, and M. Robinson, "Low-voltage low-power LVDS drivers," *IEEE J. of Solid-State Circuits (JSSC)*, vol. 40, no. 2, pp. 472–479, 2005.

- [16] W. Namgoong, M. Yu, and T. Meng, "A high-efficiency variable-voltage CMOS dynamic DC-DC switching regulator," in *IEEE Int. Solid-State Circuits Conference*, 1997.
- [17] P. Hazucha, G. Schrom, J. Hahn, B. Bloechel, P. Hack, G. Dermer, S. Narendra, D. Gardner, T. Karnik, V. De, and S. Borkar, "A 233-MHz 80%-87% efficient four-phase DC-DC converter utilizing air-core inductors on package," *IEEE J. of Solid-State Circuits*, vol. 40, no. 4, pp. 838–845, April 2005.
- [18] K. Uchiyama, "Power-efficient heterogeneous parallelism for digital convergence," in *IEEE Symp. on VLSI Circuits*, 2008.
- [19] H. H. Goldstine and A. Goldstine, "The electronic numerical integrator and computer (eniac)," *IEEE Annals of the History of Computing*, vol. 18, no. 1, pp. 10–16, March 1996.
- [20] J. Wang, S. Ma, P. Manoj, M. Yu, R. Weerasekera, and H. Yu, "High-speed and low-power 2.5D I/O circuits for memory-logic-integration by through-silicon interposer," in *IEEE Int. 3D Systems Integration Conf. (3DIC)*, 2013.
- [21] K. Banerjee, S. Souri, P. Kapur, and K. Saraswat, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 602–633, May 2001.
- [22] C. S. Tan, R. J. Gutmann, and L. R. Reif, *Overview of Wafer-Level 3D ICs*. Springer US, 2008.
- [23] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, 2004.
- [24] B. Goplen and S. Sapatnekar, "Thermal via placement in 3D ICs," in *IEEE/ACM Int. Symp. on Physical Design (ISPD)*, 2005.
- [25] H. Yu, Y. Shi, L. He, and T. Karnik, "Thermal via allocation for 3D ICs considering temporally and spatially variant thermal power," in *Int. Symp. on Low Power Electronics and Design (ISLPED)*, 2006.
- [26] H. Yu, J. Ho, and L. He, "Simultaneous power and thermal integrity driven via stapling in 3D ICs," in *ACM/IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, 2006.

- [27] H. Yu, Y. Shi, L. He, and T. Karnik, "Thermal via allocation for 3-D ICs considering temporally and spatially variant thermal power," *IEEE Trans. on Very Large Scale Integration (TVLSI) Systems*, vol. 16, no. 12, pp. 1609–1619, Dec 2008.
- [28] H. Yu, J. Ho, and L. He, "Allocating power ground vias in 3D ICs for simultaneous power and thermal integrity," *ACM Trans. on Design Automation of Electronic Systems (TODAES)*, vol. 14, no. 3, pp. 41:1–41:31, Jun 2009.
- [29] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, "Design space exploration for 3D architectures," *ACM J. on Emerging Technologies in Computing Systems (JET-CAS)*, vol. 2, no. 2, pp. 65–103, Apr 2006.
- [30] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner, "Picoserver: Using 3D stacking technology to enable a compact energy efficient chip multiprocessor," in *Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM, 2006.
- [31] M. Motoyoshi, "Through-silicon via (TSV)," *IEEE Proceedings*, vol. 97, no. 1, pp. 43–48, 2009.
- [32] H. Qian, X. Huang, H. Yu, and C. H. Chang, "Real-time thermal management of 3D multi-core system with fine-grained cooling control," in *IEEE Int. 3D Systems Integration Conf. (3DIC)*, 2010.
- [33] X. Huang, H. Yu, and W. Zhang, "NEMS based thermal management for 3D many-core system," in *ACM/IEEE Int. Symp. on Nanoscale Architectures*, June 2011, pp. 218–223.
- [34] H. Qian, X. Huang, H. Yu, and C. Chang, "Cyber-physical thermal management of 3D multi-core cache-processor system with microfluidic cooling," *ASP J. of Low Power Electronics*, vol. 7, no. 1, pp. 110–121, Feb 2011.
- [35] H. Qian, C.-H. Chang, and H. Yu, "An efficient channel clustering and flow rate allocation algorithm for non-uniform microfluidic cooling of 3D integrated circuits," *Integration VLSI Journal*, vol. 46, no. 1, pp. 57–68, Jan 2013.
- [36] Y. Shang, C. Zhang, H. Yu, C. S. Tan, X. Zhao, and S. K. Lim, "Thermal-reliable 3D clock-tree synthesis considering nonlinear electrical-thermal-coupled TSV," in *Asia and South Pacific Design Automation Conf. (ASP-DAC)*. ACM/IEEE, 2013.

- [37] P. Manoj, H. Yu, Y. Shang, C. S. Tan, and S. K. Lim, “Reliable 3-D clock-tree synthesis considering nonlinear capacitive TSV model with electrical-thermal-mechanical coupling,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 11, pp. 1734–1747, Nov 2013.
- [38] P. Manoj, H. Yu, and K. Wang, “3D many-core microprocessor power management by space-time multiplexing based demand-supply matching,” *IEEE Trans. on Computers*, vol. PP, no. 99, pp. 1–1, 2015.
- [39] S.-S. Wu, K. Wang, P. Manoj, T.-Y. Ho, M. Yu, and H. Yu, “A thermal resilient integration of many-core microprocessors and main memory by 2.5D TSI I/Os,” in *Design, Automation and Test in Europe Conf. (DATE)*. ACM/IEEE, 2014.
- [40] D. Xu, P. Manoj, H. Huang, N. Yu, and H. Yu, “An energy-efficient 2.5D through-silicon interposer I/O with self-adaptive adjustment of output-voltage swing,” in *ACM/IEEE Int. Symp. on Low Power Electronics and Design (ISLPED)*, 2014.
- [41] H. Hantao, P. Manoj, D. Xu, H. Yu, and Z. Hao, “Reinforcement learning based self-adaptive voltage-swing adjustment of 2.5D I/Os for many-core microprocessor and memory communication,” in *ACM/IEEE Int. Conf. on Computer-Aided Design*, 2014.
- [42] A. Agarwal and M. Levy, “Going multicore presents challenges and opportunities,” *Embedded Systems Design*, vol. 20, no. 4, Apr 2007.
- [43] M. B. Taylor and et.al., *Tiled Multicore Processors*, ser. Integrated Circuits and Systems. Springer, 2009, ch. 1, pp. 1–34.
- [44] K. Sankaralingam, R. Nagarajan, R. McDonald, R. Desikan, S. Drolia, M. S. Govindan, P. Gratzf, D. Gulati, H. Hanson, C. Kim, H. Liu, N. Ranganathan, S. Sethumadhavan, S. Shariff, P. Shivakumar, S. Keckler, and D. Burger, “Distributed microarchitectural protocols in the TRIPS prototype processor,” in *IEEE/ACM Int. Symp. on Microarchitecture*, 2006.
- [45] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown III, and A. Agarwal, “On-chip interconnection architecture of the tile processor,” *IEEE Micro*, vol. 27, no. 5, pp. 15–31, Sep 2007.
- [46] M. Baron, “Tilera’s core communicate better,” in *Microprocessor Report*, Nov 2007.

- [47] —, “Low-key Intel 80-core intro: The tip of the iceberg,” in *Microprocessor Report*, Apr 2007.
- [48] J. Shin, K. Tam, D. Huang, B. Petrick, H. Pham, C. Hwang, H. Li, A. Smith, T. Johnson, F. Schumacher, D. Greenhill, A. Leon, and A. Strong, “A 40nm 16-core 128-thread CMT SPARC SoC processor,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2010.
- [49] N. Sturcken, E. O’Sullivan, N. Wang, P. Herget, B. Webb, L. Romankiw, M. Petracca, R. Davies, R. Fontana, G. Decad, I. Kymissis, A. Peterchev, L. Carloni, W. Gallagher, and K. Shepard, “A 2.5D integrated voltage regulator using coupled-magnetic-core inductors on silicon interposer delivering 10.8A/mm²,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2012.
- [50] “Nvidia Tesla GPU,” <http://www.nvidia.com/object/tesla-servers.html>.
- [51] R. C. Whaley, A. Petitet, and J. J. Dongarra, “Automated empirical optimization of software and the ATLAS project,” *Parallel Computing*, vol. 27, pp. 3–35, 2000.
- [52] A. Jantsch, M. Grange, and D. Pamunuwa, *The Promises and Limitations of 3-D Integration*, ser. Integrated Circuits and Systems. Springer, 2011, pp. 27–44.
- [53] J. Bulzacchelli, M. Meghelli, S. Rylov, W. Rhee, A. Rylyakov, H. Ainspan, B. Parker, M. Beakes, A. Chung, T. Beukema, P. Pepeljugoski, L. Shan, Y. Kark, S. Gowda, and D. Friedman, “A 10-Gb/s 5-tap DFE/4-tap FFE transceiver in 90-nm CMOS technology,” *IEEE J. of Solid-State Circuits (JSSC)*, vol. 41, no. 12, pp. 2885–2900, 2006.
- [54] R. Palmer, J. Poulton, W. Dally, J. Eyles, A. Fuller, T. Greer, M. Horowitz, M. Kellam, F. Quan, and F. Zarkeshvari, “A 14mW 6.25Gb/s transceiver in 90nm CMOS for serial chip-to-chip communications,” in *IEEE Int. Solid-State Circuits Conf.*, 2007.
- [55] W. J. Dally and J. Poulton, *Digital Systems Engineering*. Cambridge Univ. Press, 1998.
- [56] M.-J. Lee, W. Dally, and P. Chiang, “Low-power area-efficient high-speed I/O circuit techniques,” *IEEE J. of Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, Nov 2000.
- [57] H. Hatamkhani, F. Lambrecht, V. Stojanovic, and C.-K. K. Yang, “Power-centric design of high-speed I/Os,” in *ACM/IEEE Design Automation Conf. (DAC)*, 2006.

- [58] G. Balamurugan, J. Kennedy, G. Banerjee, J. Jaussi, M. Mansuri, F. O'Mahony, B. Casper, and R. Mooney, "A scalable 5-15 Gbps, 14-75 mW low-power I/O transceiver in 65 nm CMOS," *IEEE J. of Solid-State Circuits*, vol. 43, no. 4, pp. 1010–1019, April 2008.
- [59] A. Papanikolaou, D. Soudris, and R. Radojcic, *Introduction to Three-Dimensional Integration*, ser. IC Stacking Process and Design. Springer, 2011, pp. 1–13.
- [60] J. H. Choi, A. Bansal, M. Meterelliyoz, J. Murthy, and K. Roy, "Leakage power dependent temperature estimation to predict thermal runaway in finfet circuits," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, 2006.
- [61] A. Vassighi and M. Sachdev, "Thermal runaway in integrated circuits," *IEEE Trans. on Device and Materials Reliability*, vol. 6, no. 2, pp. 300–305, 2006.
- [62] T. Burd and et.al., "A dynamic voltage scaled microprocessor system," *IEEE J. of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov 2000.
- [63] P. Choudhary and D. Marculescu, "Power management of voltage/frequency island-based systems using hardware-based methods," *IEEE Trans. on VLSI Systems*, vol. 17, no. 3, pp. 427–438, Feb 2009.
- [64] J. Zhao, X. Dong, and Y. Xie, "An energy-efficient 3D CMP design with fine-grained voltage scaling," in *ACM/IEEE DATE Conf.*, 2011.
- [65] K. Wang, H. Yu, B. Wang, and C. Zhang, "3D reconfigurable power switch network for demand-supply matching between multi-output power converters and many-core microprocessors," in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2013.
- [66] P. Manoj, K. Wang, and H. Yu, "Peak power reduction and workload balancing by space-time multiplexing based demand-supply matching for 3D thousand-core microprocessor," in *ACM/IEEE Design Automation Conf.*, 2013.
- [67] A. Gulati, A. Merchant, M. Uysal, P. Padala, and P. Varman, "Workload dependent IO scheduling for fairness and efficiency in shared storage systems," in *IEEE Int. Conf. on High Performance Computing (HiPC)*, 2012.
- [68] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," in *Int. Workshop on System Level Interconnect Prediction*, 2004.

- [69] M. Mondal, A. Ricketts, S. Kirollos, T. Ragheb, G. Link, N. Vijaykrishnan, and Y. Massoud, “Thermally robust clocking schemes for 3D integrated circuits,” in *Design, Automation Test in Europe Conf. (DATE)*. ACM/IEEE, 2007.
- [70] I. Ndip, B. Curran, K. Lobbicke, S. Guttowski, H. Reichl, K. Lang, and H. Henke, “High-frequency modeling of TSVs for 3-D chip integration and silicon interposers considering skin-effect, dielectric quasi-TEM and slow-wave modes,” *IEEE Trans. on Components, Packaging and Manufacturing Technology*, vol. 1, no. 10, pp. 1627–1641, 2011.
- [71] T. Ishii, H. Ito, M. Kimura, K. Okada, and K. Masu, “A 6.5-mW 5-Gbps on-chip differential transmission line interconnect with a low-latency asymmetric tx in a 180nm CMOS technology,” in *IEEE Asian Solid-State Circuits Conf. (ASSCC)*, 2006.
- [72] A. Carpenter, J. Hu, O. Kocabas, M. Huang, and H. Wu, “Enhancing effective throughput for transmission line-based bus,” in *Int. Symp. on Computer Architecture (ISCA)*. ACM/IEEE, 2012.
- [73] G. Katti, A. Mercha, M. Stucchi, Z. Tokei, D. Velenis, J. Van Olmen, C. Huyghebaert, A. Jourdain, M. Rakowski, I. Debusschere, P. Soussan, H. Oprins, W. Dehaene, K. De Meyer, Y. Travaly, E. Beyne, S. Biesemans, and B. Swinnen, “Temperature dependent electrical characteristics of through-silicon-via (TSV) interconnections,” in *IEEE Int. Interconnect Technology Conf. (IITC)*, 2010.
- [74] J. seok Yang, K. Athikulwongse, Y.-J. Lee, S. K. Lim, and D. Pan, “TSV stress aware timing analysis with applications to 3D-IC layout optimization,” in *ACM/IEEE Design Automation Conf. (DAC)*, 2010.
- [75] D. Kim, K. Athikulwongse, M. Healy, M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. Woo, X. Zhao, J. Kim, H. Choi, G. Loh, H.-H. Lee, and S. Lim, “Design and analysis of 3D-MAPS (3D massively parallel processor with stacked memory),” *IEEE Trans. on Computers*, vol. 64, no. 1, pp. 112–125, Jan 2015.
- [76] J. Cong and Y. Zhang, “Thermal via planning for 3-D ICs,” in *IEEE/ACM Int. Conf on Computer-Aided Design (ICCAD)*, 2005.
- [77] D. H. Kim, K. Athikulwongse, and S. K. Lim, “A study of through-silicon-via impact on the 3D stacked IC layout,” in *IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, 2009.

- [78] D. B. Tuckerman and R. F. W. Pease, “High-performance heat sinking for VLSI,” *IEEE Electronic Device Letters*, vol. EDL-2, no. 5, pp. 126–129, May 1981.
- [79] J.-M. Koo, S. J. Im, L. Jiang, and K. E. Goodson, “Integrated microchannel cooling for three-dimensional electronic circuit architectures,” *J. of Heat Transfer*, vol. 127, no. 1, pp. 49–58, Feb 2005.
- [80] B. Dang, P. Joseph, M. Bakir, T. Spencer, P. Kohl, and J. Meindl, “Wafer-level microfluidic cooling interconnects for GSI,” in *IEEE Int. Interconnect Technology Conf. (IITC)*, 2005.
- [81] A. Coskun, D. Atienza, T. Rosing, T. Brunschweiler, and B. Michel, “Energy-efficient variable-flow liquid cooling in 3D stacked architectures,” in *IEEE/ACM Design, Automation Test in Europe Conf. (DATE)*, 2010.
- [82] Y. Cho and N. Chang, “Energy-aware clock frequency assignment in microprocessors and memory devices for dynamic voltage scaling,” *IEEE Trans. on CAD of Integrated Circuits and Systems (TCAD)*, vol. 26, no. 6, pp. 1030–1040, Jun 2007.
- [83] R. Bondade and D. Ma, “Hardware-software codesign of an embedded multiple-supply power management unit for multicore SoCs using an adaptive global/local power allocation and processing scheme,” *ACM Trans. Des. Autom. Electron. Syst. (TODAES)*, vol. 16, no. 3, pp. 31:1–31:27, Jun 2011.
- [84] W. Kim, D. Brooks, and G.-Y. Wei, “A fully-integrated 3-level DC/DC converter for nanosecond-scale DVS with fast shunt regulation,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2011.
- [85] S. Kose and E. Friedman, “Distributed on-chip power delivery,” *IEEE J. on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 4, pp. 704–713, Dec 2012.
- [86] L. Benini, A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” *IEEE Trans. on VLSI Systems*, vol. 8, no. 3, pp. 299–316, Jun 2000.
- [87] R. Golding, P. Bosh, and J. Wilkes, “Idleness is not sloth,” in *HP Laboratories Tech. Report HPL-96-140*, 1996.

- [88] M. Srivastava, A. Chandrakasan, and R. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 4, no. 1, pp. 42–55, March 1996.
- [89] C.-H. Hwang and A.-H. Wu, "A predictive system shutdown method for energy saving of event-driven computation," in *ACM/IEEE Int. Conf. on Computer-Aided Design*, 1997.
- [90] F. Douglass, P. Krishnan, and B. N. Bershad, "Adaptive disk spin-down policies for mobile computers," in *USENIX Symp. on Mobile and Location-Independent Computing*, 1995.
- [91] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically characterizing large scale program behavior," *ACM SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 45–57, Oct. 2002.
- [92] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered DVFS," in *ACM/IEEE Design, Automation Test in Europe Conference Exhibition (DATE)*, 2011.
- [93] J. Tschanz and N. Shanbhag, "A low-power, reconfigurable adaptive equalizer architecture," in *Asilomar Conf. on Signals, Systems, and Computers*, 1999.
- [94] N. Tzartzanis and W. Walker, "Differential current-mode sensing for efficient on-chip global signaling," *IEEE J. of Solid-State Circuits*, vol. 40, no. 11, pp. 2141–2147, Nov 2005.
- [95] J. sun Seo, R. Ho, J. Lexau, M. Dayringer, D. Sylvester, and D. Blaauw, "High-bandwidth and low-energy on-chip signaling with adaptive pre-emphasis in 90nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2010.
- [96] "Hybrid Memory Cube Consortium," <http://hybridmemorycube.org/tool-resources.html>.
- [97] Denali Software Inc., "Databahn DRAM memory controller IP," 2009, <https://www.denali.com/en/products/databahndram.jsp>.
- [98] M. Facchini, T. Carlson, A. Vignon, M. Palkovic, F. Catthoor, W. Dehaene, L. Benini, and P. Marchal, "System-level power/performance evaluation of 3D stacked DRAMs for mobile applications," in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2009.

- [99] S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE J. of Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, 2007.
- [100] M. Pozzoni and et.al., "A multi-standard 1.5 to 10Gb/s latch-based 3-tap DFE receiver with a SSC tolerant CDR for serial backplane communication," *IEEE J. of Solid-State Circuits*, vol. 44, no. 4, pp. 1306–1315, 2009.
- [101] J. Cong and Y. Zhang, "Thermal-driven multilevel routing for 3D-ICs," in *ACM/IEEE ASP-DAC*, 2005.
- [102] T. Bandyopadhyay, R. Chatterjee, D. Chung, M. Swaminathan, and R. Tummala, "Electrical modeling of through silicon and package vias," in *IEEE Int. Conf. on 3D System Integration (3DIC)*, 2009.
- [103] G. Katti, M. Stucchi, K. de Meyer, and W. Dehaene, "Electrical modeling and characterization of through silicon via for three-dimensional ICs," *IEEE Trans. on Electron Devices*, vol. 57, no. 1, pp. 256–262, Jan 2010.
- [104] A. Mercha, G. Van der Plas, V. Moroz, I. de Wolf, P. Asimakopoulos, N. Minas, S. Domae, D. Perry, M. Choi, A. Redolfi, C. Okoro, Y. Yang, J. Van Olmen, S. Thangaraju, D. Sabuncuoglu Tezcan, P. Soussan, J. Cho, A. Yakovlev, P. Marchal, Y. Travalay, E. Beyne, S. Biesemans, and B. Swinnen, "Comprehensive analysis of the impact of single and arrays of through silicon vias induced stress on high-k/metal gate CMOS performance," in *IEEE Int. Electron Devices Meeting (IEDM)*, 2010.
- [105] D. F. Lim, K. C. Leong, and C. S. Tan, "Selection of underfill material in Cu hybrid bonding and its effect on the transistor keep-out-zone," in *3D-IC*. IEEE, 2011.
- [106] K. Ghosh, J. Zhang, L. Zhang, Y. Dong, H. Li, C. Tan, G. Xia, and C. Tan, "Strategy for TSV scaling with consideration on thermo-mechanical stress and acceptable delay," in *Microsystems, Packaging, Assembly and Circuits Tech. Conf. (IMPACT)*. IEEE, 2012.
- [107] X. Zhao, J. Minz, and S. K. Lim, "Low-power and reliable clock network design for through-silicon via (TSV) based 3D ICs," *IEEE Trans. on Components, Packaging, and Manufacturing Technology*, vol. 1, no. 2, pp. 247–259, Feb 2011.

- [108] J. Minz, X. Zhao, and S. K. Lim, “Buffered clock tree synthesis for 3D ICs under thermal variations,” in *IEEE/ACM Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2008.
- [109] M. Cho, S. Ahmed, and D. Pan, “TACO: Temperature aware clock-tree optimization,” in *ACM/IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, 2005.
- [110] “NVM SPICE,” <http://www.nvmspice.org>.
- [111] R. Chang, N. Talwalkar, C. Yue, and S. Wong, “Near speed-of-light signaling over on-chip electrical interconnects,” *IEEE J. of Solid-State Circuits*, vol. 38, no. 5, pp. 834–838, May 2003.
- [112] C. Kim, D. Burger, and S. W. Keckler, “An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches,” *SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 211–222, Oct 2002.
- [113] R. Nagarajan, K. Sankaralingam, D. Burger, and S. Keckler, “A design space evaluation of grid processor architectures,” in *ACM/IEEE Int. Symp. on Microarchitecture*, 2001.
- [114] S. Palacharla, N. P. Jouppi, and J. E. Smith, “Complexity-effective superscalar processors,” *SIGARCH Comput. Archit. News*, vol. 25, no. 2, pp. 206–218, May 1997.
- [115] J. Lin, S. Zhu, Z. Yu, D. Xu, S. M. P. D., and H. Yu, “A scalable and reconfigurable 2.5d integrated multicore processor on silicon interposer,” in *IEEE Custom Integrated Circuits Conf.*, 2015.
- [116] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, and A. Kahng, “Zero skew clock routing with minimum wirelength,” *IEEE Trans. on Circuits and Systems II*, vol. 39, no. 11, pp. 799–814, Nov 1992.
- [117] R. S. Tsay, “An exact zero-skew clock routing algorithm,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 2, pp. 242–249, Feb 1993.
- [118] J. Cong, A. B. Kahng, C.-K. Koh, and C.-W. A. Tsao, “Bounded-skew clock and steiner routing,” *ACM Trans. on Design Automation of Electronic Systems*, vol. 3, no. 3, pp. 341–388, Jul 1998.

- [119] S. Basir-Kazeruni, H. Yu, F. Gong, Y. Hu, C. Liu, and L. He, "SPECOC: Stochastic perturbation based clock tree optimization considering temperature uncertainty," *Elsevier Integration, the VLSI Journal*, vol. 46, no. 1, pp. 57–68, Jan 2013.
- [120] Y. J. Kim, Y. K. Joshi, A. G. Fedorov, Y.-J. Lee, and S.-K. Lim, "Thermal characterization of interlayer microfluidic cooling of three-dimensional integrated circuits with nonuniform heat flux," *J. of Heat Transfer*, vol. 132, no. 4, pp. 214–219, Feb 2010.
- [121] T.-Y. Kim and T. Kim, "Clock tree embedding for 3D ICs," in *IEEE ASP-DAC*, 2010.
- [122] X. Zhao, J. Minz, and S. K. Lim, "Low-power and reliable clock network design for through-silicon-via (TSV) based 3D ICs," *IEEE Trans. on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 247–259, Feb 2011.
- [123] J. Long, J. C. Ku, S. Memik, and Y. Ismail, "SACTA: A self-adjusting clock tree architecture for adapting to thermal-induced delay variation," *IEEE Trans. on VLSI Systems*, vol. 18, no. 9, pp. 1323–1336, Sep 2010.
- [124] X. Zhao and S. K. Lim, "Power and slew-aware clock network design for through-silicon-via (TSV) based 3D ICs," in *IEEE/ACM Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2010.
- [125] D. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer Verlag, 2008, vol. 116.
- [126] "3D-ACME, 3D-IC steady state temperature simulator," <http://www.3dacme.allalla.com/>.
- [127] "Hotspot," <http://lava.cs.virginia.edu/hotspot/>.
- [128] "SPEC 2000 CPU benchmark suits," <http://www.spec.org/cpu/>.
- [129] "IBM clock tree benchmarks," <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/>.
- [130] J. A. Plaza, A. Collado, E. Cabruja, and J. Esteve, "Piezoresistive Accelerometers for MCM Package," *IEEE J. of Microelectromechanical Systems*, vol. 11, no. 6, pp. 794–801, 2002.
- [131] F. O'Mahony and et.al., "The future of electrical I/O for microprocessors," in *Int. Symp. on VLSI Design, Automation and Test (VLSI-DAT)*, 2009.

- [132] A. Deutsch, “Electrical Characteristics of Interconnections for High-Performance Systems,” *Proceedings of the IEEE*, vol. 86, no. 2, pp. 315–357, 1998.
- [133] W. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. Sule, M. Steer, and P. Franzon, “Demystifying 3D ICs: the pros and cons of going vertical,” *IEEE Design and Test of Computers (DTC)*, vol. 22, no. 6, pp. 498–510, Nov 2005.
- [134] R. Ramanujam and B. Lin, “A layer-multiplexed 3D on-chip network architecture,” *IEEE Embedded Systems Letters*, vol. 1, no. 2, pp. 50–55, Aug 2009.
- [135] G. Van der Plas, P. Limaye, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Lintén, M. Stucchi, K. Guruprasad, D. Velenis, D. Shinichi, V. Cherman, B. Vanderveelde, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de Potter de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, W. Dehaene, Y. Travaly, P. Marchal, and E. Beyne, “Design issues and considerations for low-cost 3D TSV IC technology,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2010.
- [136] D. H. Woo, N. H. Seong, D. Lewis, and H.-H. Lee, “An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth,” in *IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, 2010.
- [137] Y. Panov and M. Jovanovic, “Design considerations for 12-V/1.5-V, 50-A voltage regulator modules,” *IEEE Trans. on Power Electronics*, vol. 16, no. 6, pp. 776–783, Nov 2001.
- [138] D. Ma, W.-H. Ki, C. ying Tsui, and P. Mok, “Single-inductor multiple-output switching converters with time-multiplexing control in discontinuous conduction mode,” *IEEE J. of Solid-State Circuits*, vol. 38, no. 1, pp. 89–100, Jan 2003.
- [139] M.-H. Huang and K.-H. Chen, “Single-inductor multi-output (SIMO) DC-DC converters with high light-load efficiency and minimized cross-regulation for portable devices,” *IEEE J. of Solid-State Circuits*, vol. 44, no. 4, pp. 1099–1111, Apr 2009.
- [140] M. Sabry, A. Coskun, D. Atienza, T. Rosing, and T. Brunschweiler, “Energy-efficient multi-objective thermal control for liquid-cooled 3D stacked architectures,” *IEEE Trans. on CAD of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 11, pp. 2015–2029, Nov 2011.

- [141] X. Huang, C. Zhang, H. Yu, and W. Zhang, "A nano-electro-mechanical-switch based thermal management for 3D integrated many-core memory-processor system," *IEEE Trans. on Nanotechnology (TNANO)*, vol. 11, no. 3, pp. 588–600, May 2012.
- [142] P. Manoj and H. Yu, "Cyber-physical management for heterogeneously integrated 3D thousand-core on-chip microprocessor," in *IEEE Int. Symp. on Circuits And Systems (ISCAS)*, 2013.
- [143] Y. Tan, W. Liu, and Q. Qiu, "Adaptive power management using reinforcement learning," in *ACM/IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, 2009.
- [144] M. Shafique, B. Vogel, and J. Henkel, "Self-adaptive hybrid dynamic power management for many-core systems," in *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, 2013.
- [145] W. Lee, Y. Wang, and M. Pedram, "VRCon: Dynamic reconfiguration of voltage regulators in a multicore platform," in *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, 2014.
- [146] S. Samii, M. Selkala, E. Larsson, K. Chakrabarty, and Z. Peng, "Cycle-accurate test power modeling and its application to soc test architecture design and scheduling," *IEEE Trans. on CAD of Integrated Circuits and Systems (TCAD)*, vol. 27, no. 5, pp. 973–977, May 2008.
- [147] Y. H. Katz, D. A. Culler, S. S. S. Alspaugh, Y. Chen, S. Dawson-haggerty, P. Dutta, M. He, X. Jiang, L. Keys, A. Krioukov, K. Lutz, J. Ortiz, P. Mohan, E. Reutzel, J. Taneja, J. Hsu, and S. Shankar, "An information-centric energy infrastructure: The berkley view," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 7–22, Mar 2011.
- [148] S. Li, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *IEEE MICRO*, 2009.
- [149] N. Binkert, B. Beckmann, G. Black, S. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. Hower, T. Krishna, S. Sardashti, and et.al., "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, Aug 2011.
- [150] "CACTI model," <http://www.hpl.hp.com/research/cacti/>.

- [151] S. Garg, D. Marculescu, R. Marculescu, and U. Ogras, “Technology-driven limits on DVFS controllability of multiple voltage-frequency island designs: a system-level perspective,” in *ACM/IEEE Design Automation Conf. (DAC)*, 2009.
- [152] “ILP solver 5.5,” <http://lpsolve.sourceforge.net/5.5/>.
- [153] “Autoregression analysis,” <http://paulbourke.net/miscellaneous/ar/>.
- [154] “MIPS processor cores,” <http://www.mips.com/products/processor-cores/>.
- [155] “Wattch version 1.02,” <http://www.eecs.harvard.edu/~dbrooks/wattch-form.html>.
- [156] A. Vahidsafa and S. Bhutani, “SPARC M6: Oracle’s Next Generation Processor for Enterprise Systems,” in *HOT CHIPS*, 2013.
- [157] R. Kumar, V. Zyuban, and D. M. Tullsen, “Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling,” in *IEEE Int. Symp. on Computer Arch.*, 2005.
- [158] S. Rusu, H. Muljono, D. Ayers, S. Tam, W. Chen, A. Martin, S. Li, S. Vora, R. Varada, and E. Wang, “Ivytown: A 22nm 15-core enterprise xeon® processor family,” in *IEEE Solid-State Circuits Conf. (ISSCC)*, 2014.
- [159] I. Loi, P. Marchal, A. Pullini, and L. Benini, “3D NoCs - unifying inter & intra chip communication,” in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, 2010.
- [160] Y.-J. Lee and S. K. Lim, “Ultra high density logic designs using monolithic 3D integration,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits*, vol. 32, no. 12, pp. 1892–1905, Dec 2013.
- [161] J. Zhang, M. O. Bloomfield, jian Qiang Lu, R. D. Gutmann, and T. S. Cale, “Thermal stresses in 3D IC inter-wafer interconnects,” *Elsevier J. of Microelectronic engineering*, vol. 82, no. 3, pp. 534–547, Dec 2005.
- [162] T. Zhang, K. Wang, Y. Feng, X. Song, L. Duan, Y. Xie, X. Cheng, and Y.-L. Lin, “A customized design of DRAM controller for on-chip 3D DRAM stacking,” in *IEEE Custom Integrated Circuits Conf. (CICC)*, 2010.
- [163] B. Akesson, P.-C. Huang, F. Clermidy, D. Dutoit, K. Goossens, Y.-H. Chang, T.-W. Kuo, P. Vivet, and D. Wingard, “Memory controllers for high-performance and real-time mpsoes requirements, architectures, and future trends,” in *ACM/IEEE Int. Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2011.

- [164] C. Weis, I. Loi, L. Benini, and N. Wehn, “An energy efficient DRAM subsystem for 3D integrated SoCs,” in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2012.
- [165] K. Chandrasekar, C. Weis, B. Akesson, N. Wehn, and K. Goossens, “System and circuit level power modeling of energy-efficient 3D-stacked wide I/O DRAMs,” in *ACM/IEEE Design, Automation Test in Europe Conf. (DATE)*, 2013.
- [166] B. Jacob, “The memory system: You can’t avoid it, you can’t ignore it, you can’t fake it,” *Synthesis Lectures on Computer Architecture*, vol. 4, no. 1, pp. 1–77, 2009.
- [167] S. Bayliss and G. A. Constantinides, “Methodology for designing statically scheduled application-specific SDRAM controllers using constrained local search,” in *IEEE Int. Conf. on Field-Programmable Technology (FPT)*, 2009.
- [168] M. Paolieri, E. Quiones, F. Cazorla, and M. Valero, “An analyzable memory controller for hard real-time CMPs,” *IEEE Embedded Systems Letters*, vol. 1, no. 4, pp. 86–90, Dec 2009.
- [169] L. Steffens, M. Agarwal, and P. Wolf, “Real-time analysis for memory access in media processing SoCs: A practical approach,” in *IEEE Euromicro Conf. on Real-Time Systems*, 2008.
- [170] K. Sewell, R. Dreslinski, T. Manville, S. Satpathy, N. Pinckney, G. Blake, M. Cieslak, R. Das, T. Wenisch, D. Sylvester, D. Blaauw, and T. Mudge, “Swizzle-switch networks for many-core systems,” *IEEE J. on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 278–294, 2012.
- [171] A. Hansson, N. Agarwal, A. Kolli, T. Wenisch, and A. N. Udipi, “Simulating dram controllers for future system architecture exploration,” in *Int. Symp. on Performance and Analysis of Systems and Software*, 2014.
- [172] A. Hilton, S. Nagarakatte, and A. Roth, “iCFP: Tolerating all-level cache misses in in-order processors,” in *IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, 2009.
- [173] M. K. Jeong, M. Erez, C. Sudanthi, and N. Paver, “A QoS-aware memory controller for dynamically balancing GPU and CPU bandwidth use in an MPSoC,” in *ACM/IEEE Design Automation Conf. (DAC)*, 2012.

- [174] M. K. Jeong, D. H. Yoon, and M. Erez, “DrSim: A platform for flexible DRAM system research,” <http://lph.ece.utexas.edu/public/DrSim>.
- [175] “SPEC CPU2006 Benchmark,” <http://www.spec.org/cpu2006/>.
- [176] “PARSEC Benchmark,” <http://parsec.cs.princeton.edu/>.
- [177] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis, “Evaluating mapreduce for multi-core and multiprocessor systems,” in *IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, 2007.
- [178] C. Erdmann, D. Lowney, A. Lynam, A. Keady, J. McGrath, E. Cullen, D. Breathnach, D. Keane, P. Lynch, M. De La Torre, R. De La Torre, P. Lim, A. Collins, B. Farley, and L. Madden, “A heterogeneous 3D-IC consisting of two 28nm FPGA die and 32 reconfigurable high-performance data converters,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2014.
- [179] A. Gosavi, “A reinforcement learning algorithm based on policy iteration for average reward: Empirical results with yield management and convergence analysis,” *Machine Learning Journal*, vol. 55, no. 1, pp. 5–29, Apr 2004.
- [180] N. Mastronarde and M. van der Schaar, “Online reinforcement learning for dynamic multimedia systems,” *IEEE Trans. on Image Processing*, vol. 19, no. 2, pp. 290–305, Feb 2010.
- [181] M. Triki, A. Ammari, Y. Wang, and M. Pedram, “Reinforcement learning-based dynamic power management of a battery-powered system supplying multiple active modes,” in *IEEE European Modelling Symp. (EMS)*, 2013.
- [182] H. Shen, Y. Tan, J. Lu, Q. Wu, and Q. Qiu, “Achieving autonomous power management using reinforcement learning,” *ACM Trans. on Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 2, pp. 24:1–24:32, Apr 2013.
- [183] N. Mastronarde, K. Kanoun, D. Atienza, P. Frossard, and M. van der Schaar, “Markov decision process based energy-efficient on-line scheduling for slice-parallel video decoders on multicore systems,” *IEEE Trans. on Multimedia*, vol. 15, no. 2, pp. 268–278, Feb 2013.
- [184] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *ACM J. of Artificial Intelligence Research*, vol. 4, no. 1, pp. 237–285, May 1996.

- [185] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning Journal*, vol. 8, no. 3-4, pp. 279–292, May 1992.
- [186] M. L. Littman, “Value-function reinforcement learning in markov games,” *ACM Cognitive Systems Research*, vol. 2, no. 1, pp. 55–66, Apr 2001.
- [187] M. Ghavamzadeh, H. J. Kappen, M. G. Azar, and R. Munos, “Speedy Q-learning,” in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2011.
- [188] I. Foster, A. Roy, and V. Sander, “A quality of service architecture that combines resource reservation and application adaptation,” in *IEEE Int. Workshop on Quality of Service*, 2000.
- [189] V. Raghunathan, M. B. Srivastava, and R. K. Gupta, “A survey of techniques for energy efficient on-chip communication,” in *ACM/IEEE Design Automation Conf. (DAC)*, 2003.
- [190] E. E-Dar and Y. Mansour, “Learning rates for Q-learning,” *J. of Machine Learning*, vol. 5, pp. 1–25, 2003.
- [191] S. K. Das, S. K. Sen, and R. Jayaram, “Call admission and control for quality-of-service provisioning in cellular networks,” in *IEEE Int. Conf. on Universal Personal Communications Record*, 1997.
- [192] R. A. Shafik and et.al., “On the extended relationships among EVM, BER and SNR as performance metrics,” in *IEEE Int. Conf. on Electrical and Computer Engineering*, 2006.
- [193] D. K. Bhattacharryya and S. Nandi, “An efficient class of SEC-DED-AUED codes,” in *Int. Symp. on Parallel Architectures, Algorithms, and Networks*, 1997.