

A Study on Power Saving and Latency Reduction Methods for Wireless 3D Network on Chips

Hao Zhang

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

DOCTOR OF PHILOSOPHY

School of Science for Open and Environmental Systems
Graduate School of Science and Technology
Keio University

March 2014

Preface

By the development of semiconductor technology, we can combine various IP cores such as processors, caches, and I/O modules on a chip. To connect a large number of cores, Network-on-Chip (NoC) that introduces a packet switched network has been widely studied. NoCs have been utilized in cost-effective embedded devices. Such applications often demand very tight design constraints in terms of cost and performance; thus the silicon budget available for their on-chip network should be modest.

Due to the increasing design cost of custom System-on-Chips (SoCs) in recent process technologies, System-in- Packages (SiPs) or 3D ICs that can select and stack necessary known-good-dies in response to given application requirements have become one of hopeful design choices. The three-dimensional Network-on-Chip (3D NoC) also becomes an emerging research topic. The 3D NoC architecture has been extensively studied in terms of its network topology, router architecture, and routing strategy.

We pay more attention to inductive coupling among variety 3D interconnection technology, because we believe it provides the flexibility to build the target 3D ICs by adding, removing, and swapping chips in a package after the chips have been fabricated, like building blocks.

However, two issues emerge: considerable huge power consumption and routing scheme for topology-agnostic characteristic. Only a few inductor can be embedded on the chip, because the total power consumption will be unacceptable. This issue effects the scale of wireless 3D NoC. For the routing scheme, the previous works never involve this issue for inductive coupling based wireless 3D NoC. The above problems must be solved, otherwise the inductive coupling based wireless 3D NoC will not be competitive.

For the above aim, this thesis explores an effective low power technology for ring and irregular based networks, respectively. The vertical links can be dynamically shut down according to the run-time vertical link utilization. Two suitable routing schemes for topology-agnostic wireless 3D NoC are also proposed that enable the chip could be replaced or removed after fabrication. The first one collects topology information at the chip boot stage. The topology will be considered as irregular topology. Up*/Down* routing is used for transferring packets; Then second one identifies mesh topology from the whole network which is considered as irregular topology by the first routing scheme, automatically. It uses dimension ordered routing in the extracted mesh structure as far as possible. Other parts of the network is treated as an irregular network and Up*/Down* routing is

applied. These above technologies can enhance the competence of inductive coupling based wireless 3D NoC.

Finally, we summarize our proposed methods, utilizing the on/off vertical link technique for saving power and employing improved Up*/Down* routing and mixed routing as routing scheme. Based on the above, we discuss how the current work could be explored further in order to develop power efficient wireless 3D NoC.

Acknowledgments

Firstly I would like to thank Keio Univeristy for giving me a chance to study abroad and providing me the very well acadenmic enviroment. I have obtained a lot during these years. With everyone's help, I have not felt any difficulty living in Japan and studying in a Japanese laboratory as a foreigner.

I would like to express my deepest appreciation to my supervisor, Professor Hideharu Amano, who is so kind, for his continuous support and guidance throughout my study. I have been fortunate to have such a great advisor who has outstanding skills and immense knowledge. I really like the enviroment here.

Also I would like to thank the reviewers of my thesis. I am grateful to Professor Kuroda, Professor Yamanaka, and Professor Matsutani for their careful reviews and valuable comments. Their constructive criticism and excellent advice have significantly improved this thesis.

I'm glad that I have chosen to join blackbus group in this lab. I have two great mates in the group. Matsutani-san has given me a lot advances and also support my study a lot. Koibuchi-san is a professor from National Institue of Informatics. He also gave me many helps about my study. I also feel lucky that I have many other group mates: Kagami-san, Kawano-san. Although Shin-sama has graduated, his paper also help me to finish this thesis. Many grateful thanks to them.

I would also like to thank all Amano lab members, past and present, for providing me daily inspirations and so much fun while studying abroad. I would like to extend my deep thanks to my fellow doctoral candidates: Amila Akagic, Mohamad Sofian bin Abu Talip and Takaaki Miyajima for sharing pleasures and sorrows as research assistant of GCOE program. Thank you for exchange ideas and provided me with many insightful technical discussions.

I would also like to thank Keio Leading-Edge Laboratory (KLL) of Science & Technology for providing research grant during three years of my study. This work was also supported in part by a Grant-in-Aid for the Global Center of Excellence for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science and Technology in Japan.

In addition, I am also very grateful to my Chinese friends who have shared enjoyable time at Keio University. So-san, Ikan-san, thank you for your great help and sharing our precious time in Japan. I will always remember and greatly value our friendship forever.

Last, I wish to express my heartfelt thanks to my lovely wife Qi and daughter Qianqian and my parents whose patient love enabled me to complete this work. Their support has been a great source of encouragement and strength.

Hao Zhang
Yokohama, Japan
March 2014

Contents

Preface	i
Acknowledgments	iii
1 Introduction	1
1.1 Background	1
1.2 Objective	3
1.3 Contributions	4
1.4 Thesis Organization	4
2 Network on Chip Technology	6
2.1 Switching Technique	6
2.2 Router Architecture	8
2.2.1 Router Components	8
2.2.2 Router Pipeline	10
2.3 Network Topology	11
2.3.1 Regular Topology	11
2.4 Packet Routing	12
2.4.1 Dimension-Order Routing	13
2.4.2 Turn Model	14
2.4.3 Up*/Down* Routing	15
2.5 Low Power Technologies	15
2.5.1 Microarchitecture Level Technologies	15
2.5.2 System-Level Technologies	16
2.6 Examples of NoC Architecture	18
2.6.1 Raw Processor	18
2.6.2 Intel Single-chip Cloud Computer (SCC)	20
2.7 Summary	21

3	Wireless Three-Dimension NoCs	22
3.1	3D IC	22
3.1.1	3D Interconnection Technologies	22
3.1.2	Topologies for 3D NoC	25
3.2	Wireless 3D NoCs	25
3.2.1	Baseline 2D and 3D NoC Architectures	25
3.2.2	Wireless 3D NoC Architecture	27
3.2.3	Network Requirements for Wireless 3D NoC	27
3.3	Plug-and-Play Routing Scheme	28
3.3.1	Control Protocol	29
3.3.2	Routing Protocol	29
3.3.3	Spanning Trees Optimization	31
3.4	A Prototype Wireless 3D NoC Chip	32
3.5	Summary	35
4	Power Saving Methods for Ring Wireless 3D NoC	37
4.1	Power Consumption Analysis for Vertical Links	37
4.2	Vertical On/Off Link Control Methods	39
4.3	Vertical On/Off Link Methods for Ring Wireless 3D NoC	40
4.3.1	Sleep Control Method for URSL	41
4.3.2	Sleep Control method for BRHL	42
4.3.3	Sleep Control method for BRFL	45
4.3.4	Implementation of On/Off Link Control Logic	46
4.4	Evaluations	47
4.5	Summary	53
5	Power Saving Methods for Irregular Wireless 3D NoC	55
5.1	Assumptions	55
5.2	Vertical On/Off Link Methods for Irregular Wireless 3D NoC	56
5.3	Evaluations	59
5.3.1	Target NoC Architecture and Evaluation Environments	59
5.3.2	Baseline Comparison between Irregular and Ring Approach	60
5.3.3	Power Saving	62
5.3.4	Performance Overhead	63
5.4	Summary	64
6	Latency Reduction Methods for Wireless 3D NoC	66
6.1	Vertical Bubble Flow	66
6.2	A Plug-and-Play Routing Algorithm	67

6.3	Mixed Routing Scheme	68
6.3.1	Assumptions	68
6.3.2	Recognition of the Mesh	70
6.3.3	Mixed Routing	71
6.3.4	Evaluations	73
6.4	Summary	80
7	Conclusions	81
7.1	Summary	81
7.2	Discussion	82
7.3	Future Work	83
	Bibliography	84
	Publications	92

List of Tables

3.1	Virtual channel (VC) assignment for a cache coherence protocol.	32
4.1	Power consumption of on chip network components [uW]	37
4.2	Idle time ratio in vertical links [%]	38
4.3	Processor parameters ($N = 4$)	47
4.4	NoC parameters ($N = 4$)	47
4.5	Implementation Costs	52
4.6	Simulation Result for LU	52
4.7	Simulation Result for IS	53
5.1	Irregular topologies to be tested (parameters).	59
6.1	Target architecture for network simulator	74

List of Figures

1.1	Network-on-Chip	2
1.2	Thesis organization	5
2.1	Tile-based NoC architecture	7
2.2	Protocol stack for NoC	8
2.3	Baseline virtual channel router	9
2.4	Example pipeline of flits through the baseline router.	10
2.5	Typical direct networks and their two-dimensional layout	11
2.6	Typical indirect networks and their two-dimensional layout	13
2.7	X-Y Routing in an 4x4 2D Mesh	13
2.8	Prohibited turn sets of three routing algorithms in Turn-Model	14
2.9	SCC top-level and tile architecture	20
3.1	2D IC (left) and 3D IC (right)	23
3.2	Three-dimensional (3D) IC technologies.	24
3.3	Router for mesh-based 3D NoCs.	25
3.4	Ring-based 3D NoC.	26
3.5	2D and 3D NoC architectures.	26
3.6	Up*/Down* routing paths with different spanning tree roots.	30
3.7	Design flow of spanning trees optimization.	31
3.8	Optimization result.	32
3.9	Prototype implementation of inductive-coupling 3D NoC [34].	33
3.10	Four chips stacked in a package	33
3.11	Structure of a vertical link	35
3.12	Circuit of inductive coupling link	36
4.1	Idle time ratio in vertical links	38
4.2	Three network configurations: (a) URSL, (b) BRFL, and (c) BRHL	41
4.3	PSM on URSL	43
4.4	FSM on URSL	43
4.5	PSM on BRHL	45

4.6	FSM on BRHL	45
4.7	Sleep control logic	46
4.8	Network power consumption before using the proposed methods.	49
4.9	Network power consumption after using the proposed methods	49
4.10	Application performance before using the proposed methods.	50
4.11	Application performance after using the proposed methods.	50
4.12	Power saving.	51
4.13	Performance overhead.	51
4.14	Latency and Power Saving under Different Thresholds	53
4.15	Latency vs. Power Saving	54
5.1	Dynamic Vertical Link On/Off Algorithm	56
5.2	Wake up signals transfer in ring topology	57
5.3	Wake up signals transfer in irregular topology	57
5.4	Red vertical link is set to ever-on link	58
5.5	Wake up paths in irregular network with 16-tile configuration	59
5.6	Irregular and ring topologies used in the evaluation part.	60
5.7	Comparisons between ring and irregular approaches	61
5.8	Power saving after using the proposed power optimization method.	63
5.9	Performance overhead after using the proposed power optimization method.	64
6.1	Vertical bubble flow network.	67
6.2	Layout of coils in 3D NoC	69
6.3	Example of Topology detection flow	71
6.4	CMPs consist of four layers.	73
6.5	Average hop count of Mixed routing and irregular routing at uniform traffic (T444)	74
6.6	Average hop count of Mixed routing and irregular routing at uniform traffic (T448)	75
6.7	Average hop count of Mixed routing and irregular routing at neighbor traffic (T444)	75
6.8	Average hop count of Mixed routing and irregular routing at neighbor traffic (T448)	76
6.9	Target 3D NoC T444 for CMP simulator.	77
6.10	Comparison of execution time of mixed routing and irregular routing (1/2)	78
6.11	Comparison of execution time of mixed routing and irregular routing (2/2)	78
6.12	Power saving improvements with mixed routing compare to up*/down* routing (T444)	79
6.13	Power saving improvements with mixed routing compare to up*/down* routing (T448)	79

Chapter 1

Introduction

1.1 Background

As the CMOS technology improves, the number of intellectual property (IP) cores integrated on a System-on-Chip (SoC) has been steadily increased. As reported in [1], for example, as many as 1,024 functional blocks, each of which consists of a MIPS-like processor and a cache memory, will be implementable on a single chip with a 35nm technology. Therefore, future on-chip interconnection networks will deal with many IP cores on a single chip, and they would be limiting factors for performance, area, and power consumption of the chip.

To avoid both bandwidth and wire-delay bottlenecks which on-chip bus structures create, on-chip switch-media networks called Networks-on-Chips (NoCs) have been studied as on-chip interconnects for the next generation [2–5]. Figure 1.1 shows an example NoC that consists of sixteen tiles each of which has a processing core and a router. NoC architectures are similar to those used in parallel computers and System Area Networks (SANs) [6, 7]. In these networks, source nodes (i.e., cores) generate packets that consist of a header and a payload data. Then on-chip routers transfer them through connected links, and destination nodes decompose them.

Regardless of the network on chip being used, there are four common aspects that must be considered in the design of any interconnection network. The first three aspects deal with high-level network issues of an interconnection network.

Router - The router architecture defines how the components are organized,

Topology - Topology is the roadmap of the network and defines how the channels and routers are connected.

Routing - Given the topology of an interconnection network, routing determines which path a packet takes from the source to its destination. An efficient routing algorithm is critical for approaching the performance bounds of the topology and load-balance the topology on adversarial traffic

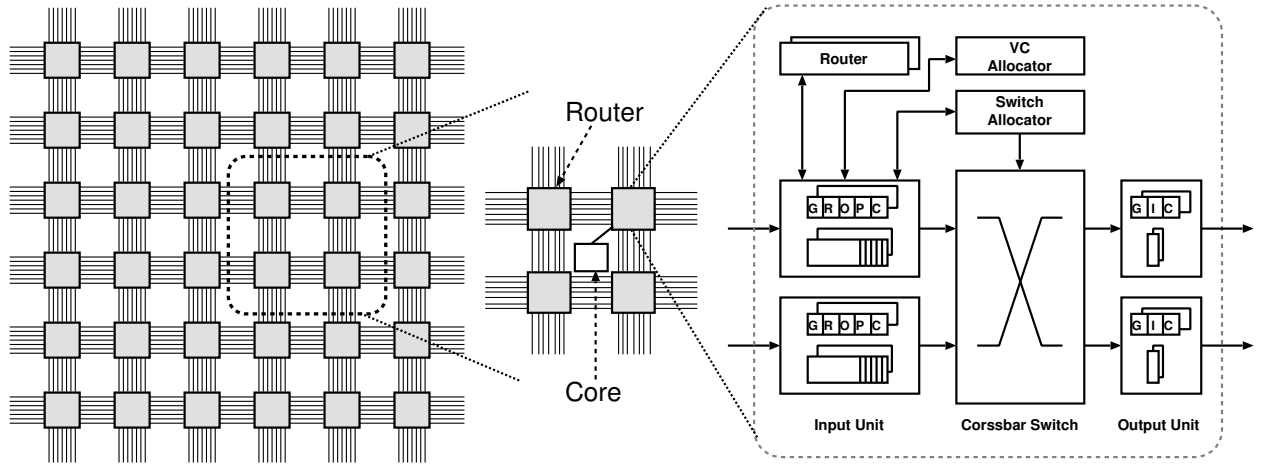


Figure 1.1: Network-on-Chip

patterns. The simplest and most popular deterministic routing widely used in off-chip and on-chip interconnection networks is dimension-order routing (DOR) [8–13]. Up*/down* routing [14] has been widely used to avoid deadlocks in SANs with arbitrary topologies using neither virtual channels nor buffers.

Low power technology - In order to reduce the increasing power consumed in interconnection networks, various system level power management techniques have been proposed. Soteriou and Peh [15] explored the design space of on/off interconnection network based on a dynamic power management technique where network links can be turn on/off in response to bursts and dips in traffic with a distribute style. [16] presented a variable frequency link for a power-aware interconnection network using the clock boosting mechanism, and applied a dynamic frequency scaling (DFS) to a 2D NoC in order to judiciously adjust link frequency based on link utilization. Power gating techniques have been applied to on chip routers [17, 18] so that standby power consumption can be reduced.

Nowadays, due to the increase of design cost of custom System-on-Chips (SoCs) in recent process technologies, System-in-Packages (SiPs) or 3D ICs that can select and stack necessary known-good-dies in response to given application requirements have become one of hopeful design choices. Various interconnection techniques have been developed to connect multiple chips in a 3D IC package: wire-bonding, micro-bump [19, 20], wireless (e.g., capacitive- and inductive-coupling) [21–24] between stacked dies, and through-silicon via (TSV) [21, 25] between stacked wafers. Many recent studies on 3D IC architectures focus on the TSV and micro-bump that offers the largest interconnect density. On the other hand, as another 3D integration technique, the inductive coupling can connect more than two known-good-dies without wire connections. We believe the inductive coupling is very hopeful, and can be popularized in the future. This is because inductive coupling provides a large

degree of flexibility to build the target 3D ICs, such as adding, removing, and swapping the chips in a package after the chip fabrication, like building blocks.

The three-dimensional Network-on-Chip (3D NoC) [26] is another emerging research topic. 3D ICs stack several smaller wafers or dies in order to reduce the wire length and wire delay, and the 3D NoC architecture has been extensively studied in terms of its network topology [27–29], router architecture [30–32], and routing strategy [33]. In this thesis, the inductive coupling is considered for 3D NoC where the number of processor chips and cache chips in the same package can be customized for a given application set. That is, if the application set requires more cache capacity or bandwidth, we can add more cache chips to satisfy such a demand. If the target application set has more thread-level parallelism, we can add more processor chips to improve the performance. Traditional wired 2D NoCs are used as intra-chip networks, while the inter-chip communication is based on wireless inductive-coupling in wireless 3D NoC. Since the wireless 3D CMP is a collection of various chips coming from different vendors (e.g., memory, processor, and GPU), we cannot expect any pre-determined network topology for each intra-chip communication; this makes it difficult to establish deadlock-free routing paths in such ad-hoc 3D CMPs.

1.2 Objective

The objective of this thesis is to study power saving and latency reduction routing for wireless 3D NoCs. The study is based on the viewpoints of low power and low latency for inductive coupling based wireless 3D NoC. To achieve this goal, the issue of reducing power consumption and routing schemes of the wireless 3D NoC are studied.

In the issue of power consumption, the goal is to reduce the power consumption of the wireless 3D NoC. Although the inductive coupling itself is energy-efficient (e.g., 0.14pJ per bit [23]), inductors continuously consume a certain amount of power, regardless of packet transfers. That is, inductors waste significant power, especially when the utilization of vertical links (i.e., inductors) is low, which is a typical use case of 3D ICs, in which most communications are within a chip while the communications between chips are not frequent. Although our wireless communication part of Cube-0 prototype can be applied to various applications, our first targets are embedded systems. Unlike the systems with powerful processors and a large mount of cache, the power consumption of wireless links will occupy a significant portion of a total power and should be minimized for embedded applications. Our objective is to find a suitable and effective low power technology. Dynamic on/off link control method is a hopeful candidate.

For the ring-based wireless 3D NoC, routing scheme has been proposed by [34]. However, the previous works have not proposed any routing scheme for inductive coupling based wireless 3D NoC with irregular topology. The wireless 3D NoC architecture is completely different from conventional 2D and 3D NoCs since its hardware components can be customized for given requirements after the chip fabrication. The packet routing protocol for the wireless 3D CMPs is completely different from

conventional 3D NoCs, since the number of nodes, the network topology, and network resources in a package cannot be expected to be known at the design time of the chips. So topology-agnostic routing algorithm (e.g. Up*/Down* routing) can be introduced here. However, only topology-agnostic routing algorithm is not enough. Since it ignores topological regularity of the vertical dimension of such chips when attempting path search, we need to enhance the simple topology-agnostic by taking topological regularity. In addition, we need to check if the power consumption can be further optimized by using the routing algorithm.

1.3 Contributions

In this thesis, we provide seven design options in order to reduce power consumption of inductive coupling based wireless 3D NoC and develop suitable routing scheme for wireless 3D NoC. Our main contributions can be highlighted as follows.

As NoC low power technology exploration, we introduce the following three topics:

1) We simulate and analyze the power consumption of inductive link (vertical link), on-chip router, and on-chip test core. It's necessary to reduce the power consumption of the wireless 3D on chip NoC.

2) We propose a novel dynamic on/off vertical link method including two low power modes. The corresponding wake up routing scheme and control logic are also proposed.

3) We apply the proposed method to three kinds ring-based network and irregular-based network. Then we evaluate the proposed method in terms of power saving and variety overheads.

As wireless 3D NoC routing scheme exploration, we introduce the following three topics:

4) We design an improved deadlock-free routing scheme for wireless 3D NoC by utilizing the regularity in irregular network. To optimize routing paths to topological structure of each chip, we classify two-surface NoC of each chip into is 2D mesh and irregular structure. In mesh topologies, X-Y-Z routing is used for well traffic distribution, whereas Up*/Down* routing is applied for reduced path hops in irregular topologies.

5) A mesh recognition algorithm is proposed to find mesh structure as much as possible.

6) A routing switching algorithm is proposed to switch from Up*/Down* routing to X-Y-Z routing, and vice versa.

The proposed schemes are evaluated in terms of hop counts and application performance.

1.4 Thesis Organization

The thesis is organized as follows, and the thesis organization is illustrated in Figure 1.2. Chapter 1 explains brief overview of NoC, 3D IC, 3D NoC as well as wireless 3D NoC. Thesis objectives and contributions are also highlighted. Chapter 2 gives an overview about NoC. Conventional 2D NoC and relative issues are discussed and examples are given. Chapter 3 surveys 3D IC and 3D

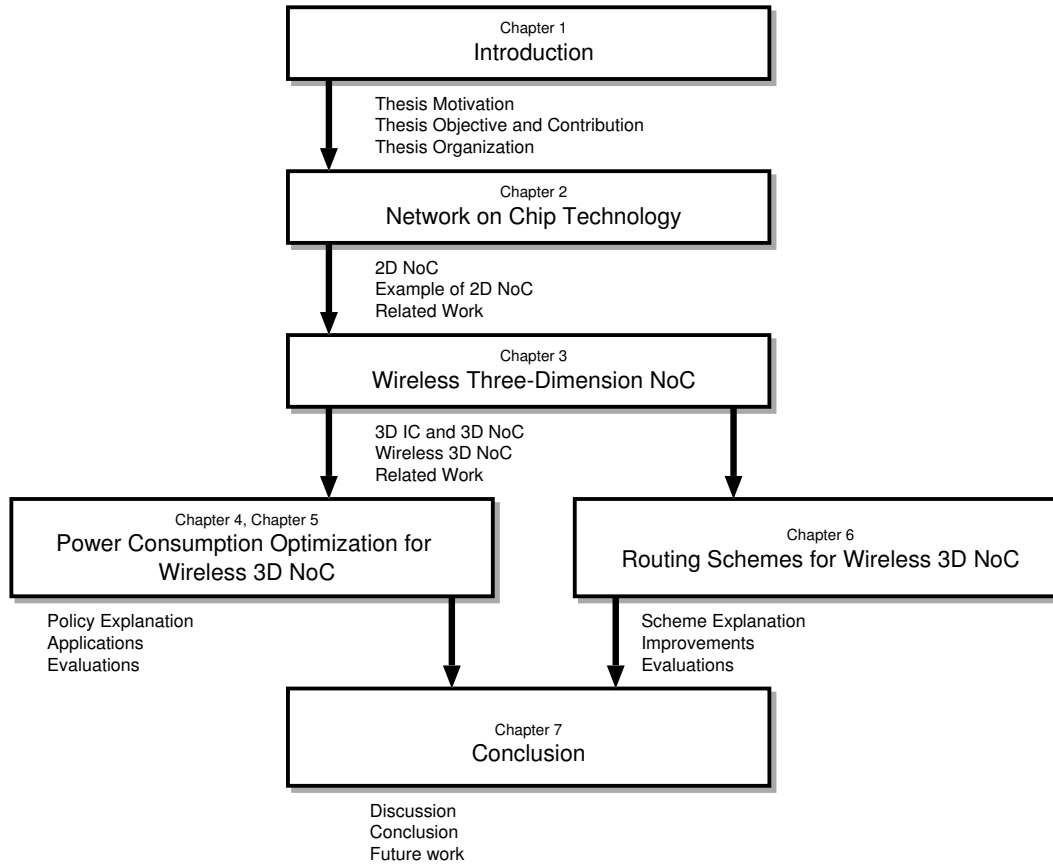


Figure 1.2: Thesis organization

NoC, especially inductive coupling based wireless 3D NoC. Then, a real wireless 3D NoC chip is introduced including chip architecture and structure of inductive link. Chapter 4 and 5 studies a novel low power technology for wireless 3D NoC by on/off vertical link dynamically. This technology is applied to ring and irregular networks. Evaluations are given and discussed in terms of power saving and overheads. Chapter 6 studies routing schemes for ring-based and irregular based wireless 3D NoC as well as an improved routing scheme. Performance and Power saving evaluations also are presented and analyzed. Finally, Chapter 7 summarizes and discusses the above studies, future directions are given and then concludes this thesis.

Chapter 2

Network on Chip Technology

In this chapter, we will survey the network on chip technology in the aspects of router architecture, network topology and packet routing schemes and low power technologies for on chip networks. Finally, we will introduce some real tile architectures.

Network-On-Chip (NoC) has been proposed as a promising replacement for buses and dedicated interconnections. Dally *et.al.* in [2] advocated the routing of packets rather than the routing of wires for on-chip networks. As shown in Figure 2.1, a chip employing an on-chip network is composed of a number of network clients: processors, DSPs, memories, peripheral controllers, gateways to networks on other chips, and custom logic. These top-level modules are connected to a network that routes packets between them instead of connecting them by routing dedicated wires.

Benini *et.al.* in [35] proposed NoC to be a new SoC paradigm. SoCs has been viewed as a micronetwork of components. The network is an abstraction of the communication among components. Figure 2.2 shows the protocol stack from which the micronetwork stack paradigm can be adopted, to abstract the electrical, logic, and functional properties.

2.1 Switching Technique

In the cases of switch-media networks, packets are transferred to their destination through multiple routers along the routing path in hop-by-hop manner. Each router forwards an incoming packet to the next router until the packet gets the final destination. The packet switching techniques decide when the router forwards the incoming packet to the neighboring router. They can be classified into three techniques: store-and-forward switching, wormhole switching, and virtual cut-through switching. They affect the network performance and buffer size needed for each router.

2) Wormhole (WH) Switching The most common switching technique used in NoCs is WH switching, in which each router stores only fractions of a packet (i.e., flit(s)) in its small buffer, and then it forwards the fractions to the next node. Since each router can forward flits of a packet before receiving the entire packet, these flits are often stored across multiple routers along the routing path.

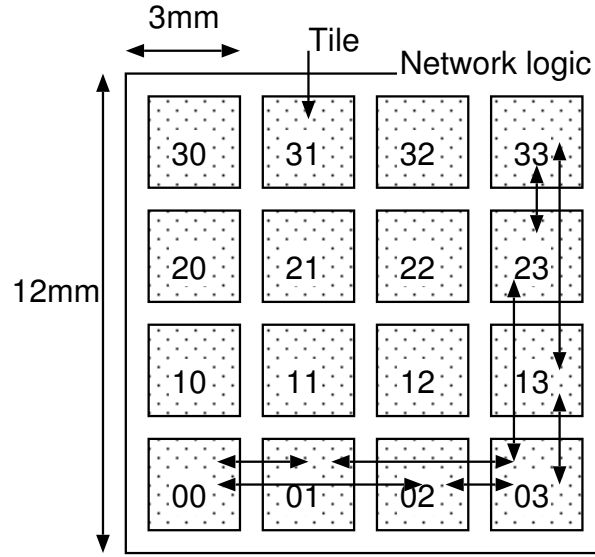


Figure 2.1: Tile-based NoC architecture

Their movement much looks like a worm. WH switching requires small buffers for each router so as to at least store a single flit in each hop, resulting in a light-weight router implementation. In addition, WH switching has advantages in the communication latency compared to SAF switching, since the maximum number of cycles required to transfer a packet is $Fh \times D + Fb$, which is much smaller than that of SAF switching when D and/or Fb become large. Therefore, WH switching is widely used in NoCs [9, 35–37]

Again, fractions of a packet can be stored across different nodes along the routing path; so the single packet often keeps occupying buffers in multiple routers along the path when the header of the packet cannot progress due to the conflicts. Such situation is referred to as “Head-of-Line (HoL) blocking”. The occupied buffers due to HoL blockings prevents other packet transfers that go through these lines. The problem of WH switching is the performance degradation due to the frequent HoL blockings.

3) WH Switching with Virtual Channels To equip multiple buffers in a single physical channel is the most common way to reduce the HoL blockings of WH switching. In this case, each of multiplexed buffers in a physical channel acts as a “virtual” channel [38], and incoming packets would progress if more than one virtual channels in a physical channel are not occupied. Therefore, the occurrence of HoL blockings will be reduced as the number of virtual channels multiplexed in a physical channel increases.

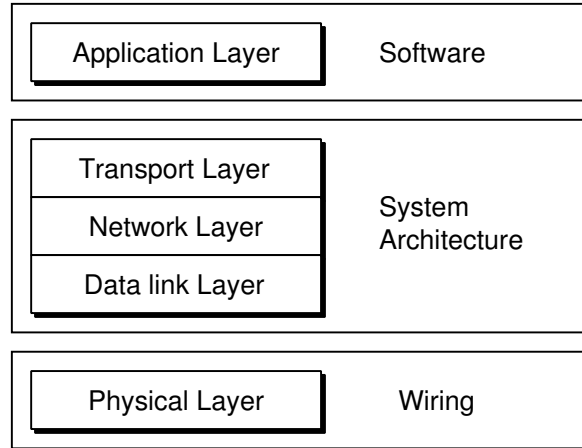


Figure 2.2: Protocol stack for NoC

2.2 Router Architecture

The router architecture has been studied for several decades for off-chip interconnects, and various architectural innovations of routers such as wormhole switching, pipelined packet transfer, and virtual channel mechanism were developed in 1980s and 90s. Although early NoCs simply have imported these architectures for on-chip purposes, various unique techniques intended to on-chip purposes are recently proposed [39–41].

General discussion about router architectures is provided in [6, 7, 42].

2.2.1 Router Components

We introduce a cost and speed model for wormhole routers here. This router consists of a routing computation unit, a virtual channel allocator, a switch allocator, a crossbar switch, and five input physical channels. Each physical channel has two virtual channels. A block diagram of the baseline virtual channel router architecture is shown in Figure 2.3.

- **Physical Channel** Each physical channel consists of up to four virtual channels, each of which has a FIFO buffer for storing four 64-bit flits. These FIFO buffers can be implemented with either SRAMs or registers, depending on the depth of the buffers, not the width. We assume that buffers should be implemented with SRAM macros if their depths are more than 32.
- **Routing Computation Unit** Routing computation unit is used to perform routing logic that determines the output port for the packet. The unit is usually very simple, because routing decisions are stored in the header flit prior to packet injection; thus routing tables that require register files for storing routing paths were not needed.

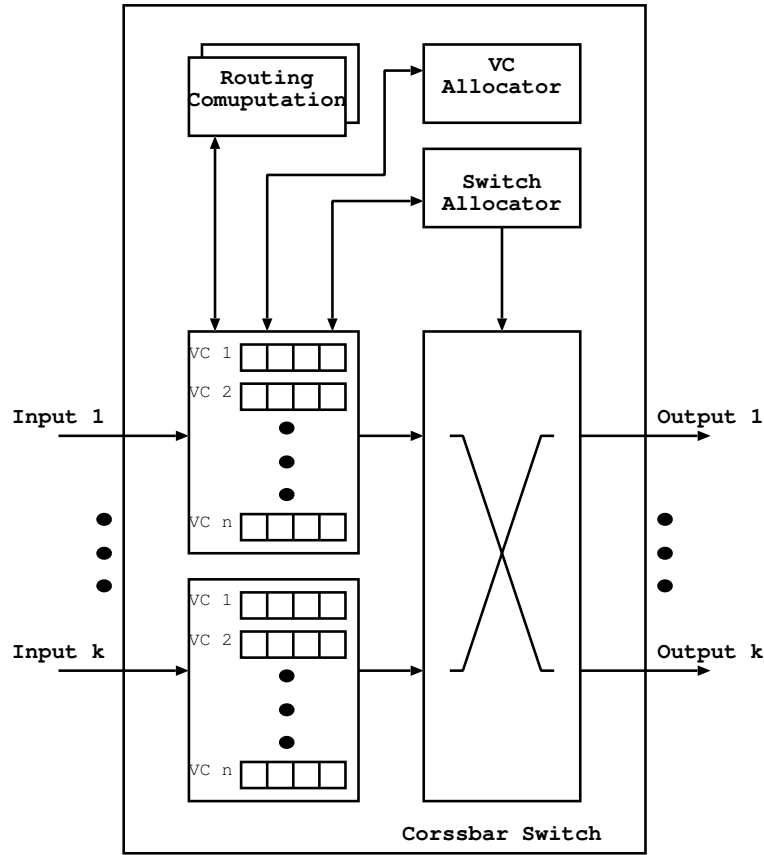


Figure 2.3: Baseline virtual channel router

- **Virtual Channel Allocator** Virtual channel allocator is used to allocate an unused target virtual channel when a packet arrives. Once more than one packet request the same virtual channel, the virtual channel allocator must arbitrate which packet wins the virtual channel.
- **Switch Allocator** The arbitration module takes the requested output and the switch status as inputs and generates switch connections. Simple arbitration algorithms implemented with log depth circuits are assumed.
- **Crossbar Switch** Most routers use a tree of gates, thus crossbar size grows as the product of the number of inputs and outputs. A virtual-channel design sometimes employs a $pv \times pv$ full crossbar, where p is the number of physical channels and v is the number of virtual channels. However, the crossbar complexity is significantly increased with the $pv \times pv$ crossbar, and its performance improvement will be limited because the data rate out of each input port is limited by its bandwidth [38].

2.2.2 Router Pipeline

Like the logic pipeline stages of a typical processor: instruction fetch, decode, execute, memory and writeback, these are logical stages that will fit into a physical pipeline depending on the actual clock frequency. Once a header flit arriving at an input port, input buffer store the arriving data. Then the pipeline starts to transfer packets. Body and tail flit follows a similar pipeline except tht thay do not go through RC and VA stages, instead inheriting the route and the virtual channel allocated by the header flit. The tail flit, on leaving the router, deallocates the virtual channel reversed by the header flit. The progression of a packet through the router can be defined as 4 steps:

1. **Router Computation (RC)**

Based on information stored in the header flit, the output port of the packet is selected.

2. **Virtual-channel Allocation (VA)**

The header flit then arbitrates for a virtual channel corresponding to its output port in this stage. A packet must gain exclusive access to downstream virtual channel associated with the output port from route computation. Once these per-packet steps are completed, per-flit scheduling of the packet can begin.

3. **Switch allocation (SA)** Upon successfully allocation of a virtual channel, the header flit proceeds to switch allocation stage where it arbitrates for the switch input and output ports. If there is a free buffer in its output virtual channel, a flit can vie for access to the crossbar.

4. **Switch Traversal (ST)** On winning the output port, the flit is then read from the buffer to its output. Then the flit is passed to the downstream router.

This process is done in pipeline way, as shown in Figure 2.4.

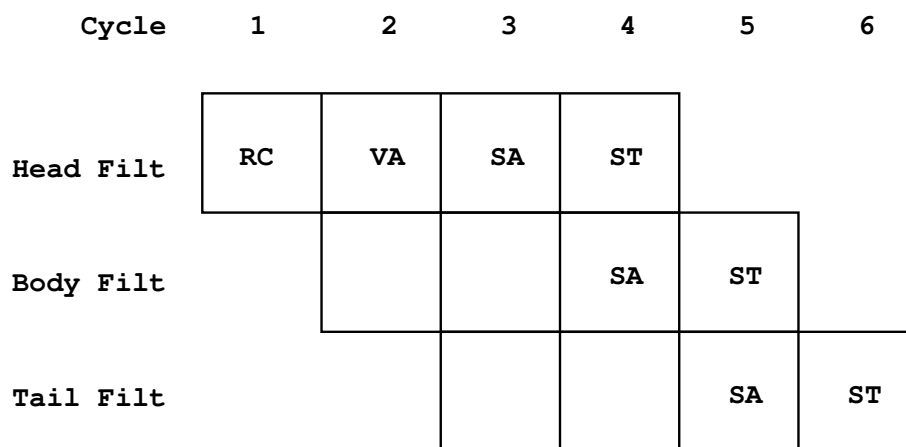


Figure 2.4: Example pipeline of flits through the baseline router.

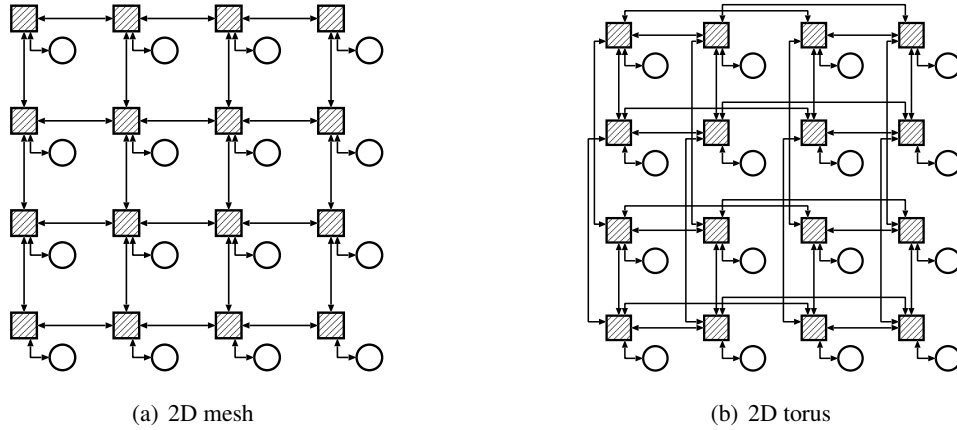


Figure 2.5: Typical direct networks and their two-dimensional layout

2.3 Network Topology

Network topology refers to the static arrangement of channels and nodes in an interconnection network, which means the path along which packets travel.

The most visible classification of network topology is based on the regularity of the connection pattern. In a regular topology, routers and cores are connected based on a certain pattern that can be represented mathematically by a set of functions. Otherwise it is referred to as irregular topology.

Regular topologies can be further divided into direct networks and indirect networks. A direct network consists of directly interconnected nodes, each of which has a processing core, a router, and a network interface. On the other hand, the router and core are separated in the case of an indirect network.

2.3.1 Regular Topology

In this section, we introduce some typical regular network topologies including direct networks and indirect networks.

They have different numbers of routers, different link lengths, and different numbers of ports per router, all of which affect throughput, amount of hardware for network resources, and energy consumption; thus a network topology should be carefully selected so as to meet the requirements of the target application.

2.3.1.1 Direct Networks

1. **2D Mesh** Two-dimensional mesh has been employed as a typical on-chip interconnect because their grid-based regular arrangement is intuitively considered to be matched to the two-dimensional VLSI layout. It can be generalized as k -ary n -mesh, where k is the number of nodes in a dimension (i.e., array size) and n is the number of dimensions. Although its node degree is five if $n = 2$, those of edge nodes and corner nodes are four and three, resulting in a

non-uniform structure. The best merit of 2D-mesh is scalability, and it provides uniform length of wires. Mesh topology is shown in Figure 2.5(a).

2. **2D Torus** Figure 2.5(b) shows a two-dimensional mesh that is the most popularly used topology in NoCs. Two-dimensional torus topology has wrap-around channels that connect nodes in top edge and bottom edge or nodes in left edge and right edge. Since all nodes in a torus have five ports, torus is a symmetric topology. Unlike 2D-mesh that does not have wrap-around channels, a 2D-torus has twice bisection bandwidth of a same-sized 2D-mesh due to its wrap-around channels. Torus topologies can also be generalized as k -ary n -cube.
3. **2D Ring** Mesh and torus networks can be described as k -ary n -cube. For instance, a 4×4 mesh or torus is a 4-ary 2-cube with 16 nodes. In each dimension, k nodes are connected with channels to their nearest neighbors. Thus, ring networks fall into the torus family of networks topologies as k -ary 1-cubes.

2.3.1.2 Indirect Networks

One of the most popular indirect network tree-based network. Tree-based topologies are also known as a hierarchical topologies. The type of network topology in which a central root node (the top level of the hierarchy) is connected to one or more other nodes that are one level lower in the hierarchy (i.e., the second level) with a point-to-point link between each of the second level nodes and the top level central root node, while each of the second level nodes that are connected to the top level central root node will also have one or more other nodes that are one level lower in the hierarchy (i.e., the third level) connected to it, also with a point-to-point link, the top level central root node being the only node that has no other node above it in the hierarchy.

And the most popular tree-based network is Fat-Tree. To mitigate the congestion around the root of the tree, Fat-Tree enhances the number of connections toward the root [43]. As stylized in [43,44], various forms of Fat-Tree can be created, and they can be expressed with a triple (p, q, c) , where p is the number of upward connections, q is the number of downward connections, and c is the number of upward connections that each core has.

Figure 2.6(a) shows a typical Fat Tree $(1,4,1)$, in which each router (except for top-rank routers) has one upward and four downward connections, and each core has one upward connections. This architecture is employed in CM-5 [45]. On the other hand, Figure 2.6(b) shows a more reasonable one labeled with $(2,4,1)$, which means every core has only one upward connection and it is used in [46].

2.4 Packet Routing

In this section, we briefly discuss various classes of routing algorithms.

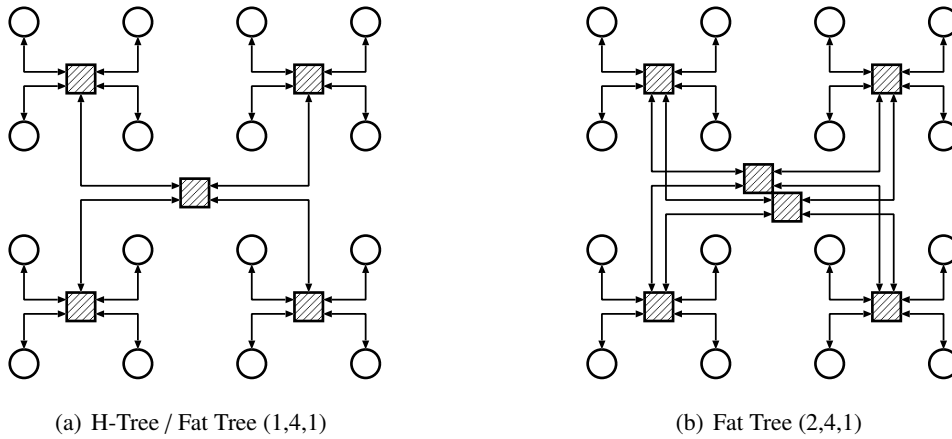


Figure 2.6: Typical indirect networks and their two-dimensional layout

Routing is the process during which data packets are forwarded from one node to another on a network until they reach their destinations. In this section some deadlock-free routing algorithms are introduced. Routing algorithms are generally divided into deterministic and adaptive.

2.4.1 Dimension-Order Routing

While numerous routing algorithms have been proposed, the most commonly used routing algorithm in on chip networks is dimension -ordering routing (DOR) due to the simplicity. DOR is an example of a deterministic routing algorithm.

With DOR, a packet traverses the network dimension by dimension, reaching the ordinate matching its destination before switching to the next dimension. In a 2D network such as mesh, X-Y dimension-ordered routing sends packets along the X-dimension first, followed by the Y-dimension. For example, as shown in Figure 2.7 packet travelling from (0, 2) to (2, 1) will first traverse two hops along the X-dimension, arriving at (2, 2), before traversing three hops along the Y-dimension to its destination.

2.4.2 Turn Model

A more sophisticated routing algorithm can be adaptive. For example, the path a message takes from (0, 2) to (2, 1) depends on the network traffic situation. A typical adaptive routing algorithm is turn model routing algorithm.

Assume that a packet is moving to its destination in a 2D mesh topology and routing decisions are implemented as a distributed routing. The packet has two choices in each hop when it is moving to its destination in a 2D mesh topology and routing decisions are implemented as a distributed routing. That is, the packet decides to go straight or turn to another dimension at every hop along the routing path until it reaches its final destination. However, several combinations of turns sometimes introduce cyclic dependencies that cause deadlocks. Glass and Ni analyzed special combinations of turns that

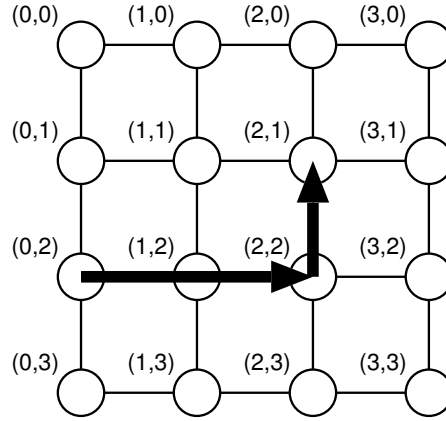


Figure 2.7: X-Y Routing in an 4x4 2D Mesh

never introduce deadlocks [47]. Their model is referred to as Turn-Model [47]. For dimension-order routing, turn-model can demonstrate the deadlock-freedom.

Figure 2.8(d) shows a set of prohibited turns in dimension-order routing. As we can see, the prohibited turn-set in dimension-order routing includes that in West-first routing. That is, the restriction of dimension-order routing is a superset of West-first routing's one. Dimension-order routing additionally prohibits two turns, which are not prohibited in West-first routing; thus the routing diversity of West-first routing is better than that of dimension-order routing. However, a routing algorithm that has high routing diversity does not always outperform that with lower diversity. Actually, West-first routing algorithm can select various routing paths, and it may select a routing path set with very poor traffic distribution that introduces hot spots and degrades the performance. On the other hand, dimension-order routing can always uniformly distribute the traffic and achieve good performance in the cases of uniform traffic, though its path diversity is poor.

2.4.3 Up*/Down* Routing

The discussion of routing algorithms in this chapter has assumed a regular topology such as a mesh or a torus. Irregular topologies can require special considerations in the development of a routing algorithm. Common routing implementations for irregular networks rely on routing table such as Up*/down* routing.

Up*/down* routing [14] has been widely used to avoid deadlocks in SANs with arbitrary topologies using neither virtual channels nor buffers. Up*/down* routing is based on the assignment of direction to network channels [14]. As the basis of the assignment, a spanning tree whose nodes correspond to switches in the network is built. The “up” end of each channel is then defined as follows: (1) the end whose node is closer to the root in the spanning tree; (2) the end whose node has the lower unique identifier (UID), if both ends are on nodes at the same tree level. A legal path must traverse zero or more channels in the up direction followed by zero or more channels in the down direction,

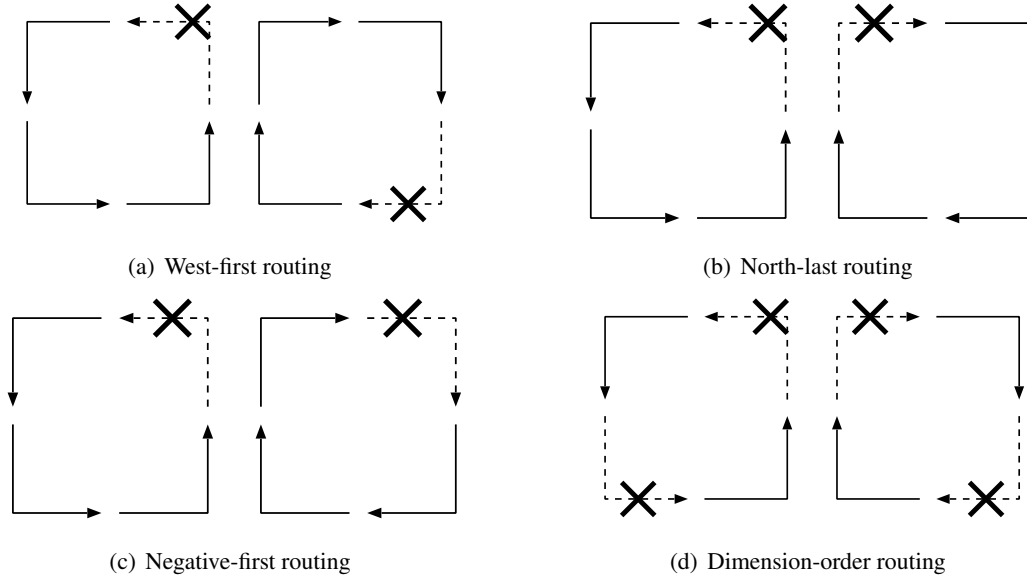


Figure 2.8: Prohibited turn sets of three routing algorithms in Turn-Model

and this rule guarantees deadlock-freedom while still allowing all hosts to be reached. However, an up*/down* routing algorithm tends to make unbalanced paths because it employs a one-dimensional directed graph.

2.5 Low Power Technologies

Various low-power techniques have been proposed for microprocessors, and they have also been applied to on-chip routers. In this section, we survey low-power techniques for microprocessors, off-chip routers, and on-chip routers.

We introduce some popular technologies from two different levels: microarchitecture level and system level.

2.5.1 Microarchitecture Level Technologies

2.5.1.1 RTL Power Optimization

Clock gating, operand isolation, and resource hibernation are well-known techniques for RTL power optimization. These techniques can be adopted for the design of network components such as switches, network interfaces, and FIFO buffers. Clock gating stops the clock to registers, which are not in use during specific clock cycles. Power saving is achieved by reducing switching activity on the clock signal to synchronous registers and the capacitive load on the clock tree. Operand isolation identifies redundant computations of datapath components and isolates such components using specific circuitry, preventing unnecessary switching activity. Determination of how operations can be identified and instantiation of isolating logic are key issues in operand isolation. Resource hi-

bernation is a coarse-grained technique which powers down modules with sleep modes, where each mode represents a trade-off between wake-up latency and power savings.

2.5.1.2 Multithreshold Circuits

Increasing the threshold voltage reduces the sub-threshold leakage exponentially, which also reduces the circuit's performance. Modern CMOS technology, referred to as MTCMOS (multithreshold CMOS), produces devices with different threshold voltage, allowing for an even better trade-off between static power and performance. For a network component logic circuit, such as switches and FIFO buffers, a higher threshold voltage can be assigned to those transistors in the noncritical paths, so as to reduce leakage current, although the performance is maintained due to the low-threshold transistors in critical paths. An MTCMOS circuit structure was analyzed by [48]. Algorithms for selecting and assigning an optimal high-threshold voltage transistor were investigated to reduce leakage power under performance constraints [49].

2.5.1.3 Buffer Allocation

Design of buffers in NoC influences power consumption, area overhead, and performance of the entire network. Buffers are a key component of a majority of network switches. Buffers have been estimated to be the single largest power consumer for a typical switch for an NoC. Application-specific buffer management schemes that allocate buffer depth for each input channel depending on the traffic pattern have been studied [50]. [51] achieved power and chip area savings by reducing the number of buffers with static and dynamic buffer allocation, enabling the repeaters to adaptively function as buffers during congestion. Banerjee [11] investigated energy-power characteristics of FIFOs to reduce buffer energy consumption in the context of an on-chip network. Power-aware buffers, which place idle buffers in an inactive mode, based on actual utilization, were proposed as an architectural technique for leakage power optimization in interconnection networks [52].

2.5.2 System-Level Technologies

2.5.2.1 Voltage and Frequency Scaling

The voltage and frequency scaling is a power saving technique that reduces the operating frequency and the supply voltage according to the applied load. The dynamic power consumption is related to the square of the supply voltage; thus, because a peak performance is not always required during the whole execution time in most cases, adjusting the frequency and supply voltage so as to at least achieve the required performance can reduce the dynamic power.

The voltage and frequency scaling techniques have been applied to microprocessors [53, 54], accelerators [55], and network links [56, 57]. In [56], the frequency and the voltage of network links

are dynamically adjusted based on the past utilization. In [57], the network link voltage is scaled down by distributing the traffic load using an adaptive routing.

The voltage and frequency scaling techniques can be classified into two schemes: *dynamic* and *static*. The frequency can be controlled by a PLL frequency divider, and the supply voltage can be adjusted by controlling an off-chip dc-dc converter [53]. They are adaptively adjusted in the dynamic scheme, while they are statically configured at the beginning of each application in the static scheme. The transition time of the clock rate and the supply voltage cannot be negligible in the dynamic scheme (e.g., 10,000 cycles [57] or $50\mu\text{s}$ [55]); so the frequent transitions sometimes overwhelm the benefits of dynamic scheme as reported in [57].

2.5.2.2 Runtime Power Gating

Power gating is a representative leakage-power reduction technique, which shuts off the power supply of idle blocks by turning off (or on) the power switches inserted between the VDD line and the blocks or between the GND line and the blocks. This concept has been applied to circuit blocks with various granularities, such as processor cores [58], execution units in a processor [59], and primitive gates [60].

We need to understand both the negative and positive impacts of power gating when we use it. Actually, a state transition between the sleep and active modes incurs a performance penalty, and turning the power switches on or off consumes an overhead energy, which means a short-term sleep rather increases the power consumption. In [59], an analytical model of the run-time power gating of the execution units in a microprocessor is provided. The following three parameters quoted from [59] affect the performance and energy.

- T_{wakeup} : Number of cycles required to charge up the local voltage of a sleeping block. A delay for turning on its power switch is also lumped into the T_{wakeup} value.
- $T_{\text{idledetect}}$: Number of cycles required to detect an idle duration in an active block and decide to shut off the block. A delay for turning off its power switch is also lumped into the $T_{\text{idledetect}}$ value.
- $T_{\text{breakeven}}$: Number of sleep cycles at least required to compensate for the overhead energy to turn the power switch on and off.

The T_{wakeup} value affects the performance (e.g., packet throughput of routers), since a pipeline stall will occur if a new request suddenly comes to a sleeping block. Also, $T_{\text{idledetect}}$ shortens the sleep duration of blocks, since an idle block must stay in the active state for $T_{\text{idledetect}}$ cycles before it decides to go to the sleep mode.

A short-term sleep of less than $T_{\text{breakeven}}$ cycles cannot compensate for the energy overhead of driving a power switch, and the power consumption will be increased; thus the $T_{\text{breakeven}}$ value

determines the benefits of power gating. The $T_{breakeven}$ value depends on various parameters, such as the sizes of a power switch and a decoupling capacitance. Hu *et. al.* report that $T_{breakeven} \approx 10$ based on the typical parameters of a recent microprocessor [59].

2.5.2.3 Dynamic On/Off Links

An alternative to save link power is to add hardware such that a link can be powered down when it is not used heavily. Control unit powers down links intelligently when their utilizations are below a certain threshold level in order to reduce the energy and power consumption for the network. A subset of highly used links can provide connectivity in the network. An adaptive routing strategy that intelligently uses a subset of links for communication was proposed, thereby facilitating DLS for minimizing energy consumption.

Chen *et. al.* propose power-aware router buffers based on Drowsy and Gated V_{dd} SRAMs to regulate their leakage power [52]. Soteriou *et. al.* provide a thorough discussion about power-aware networks whose links can be turned on and off, in terms of connectivity, routing, wake-up and sleep decisions, and router pipeline architecture [15, 61]. They explored the design space for communication channel turn-on/off depending on hardware counter measurement obtained from the network during run-time.

Compiler-directed approaches have benefits as compared to hardware-based approaches. Based on high-level communication analysis, these techniques determine the point at which a given communication link is idle and can be turned off to save power, without waiting for a certain period of time to be certain that the link has truly become idle. Similarly, the reactivation point which was identified automatically eliminates the turn on performance penalty. Chen *et al.* introduced a compiler-directed approach [52], which increases the idle periods of communication channels by reusing the same set of channels for as many communication messages as possible. [15] proposed a compiler-directed technique to turn off the communication channels to reduce NoC energy consumption.

2.6 Examples of NoC Architecture

This section will show how the NoC technologies introduced in this chapter are used in representative tile architectures.

2.6.1 Raw Processor

First, we introduce a famous multi-core processor, called Raw processor.

Raw microprocessor has been developed in MIT since the late 1990s [37, 62]. As reported in [37], a Raw microprocessor is built with 122 million transistors and it can execute sixteen different instructions such as load instruction, store instruction, ALU instructions, and FPU instructions in a single cycle. In addition, a total 2-MByte L1 cache memory is distributed over the chip.

The silicon area of the Raw microprocessor chip is divided into sixteen areas, each of which is referred to as a programmable tile. Each tile has the following modules:

- A single on-chip router for static communication
- Two on-chip routers for dynamic communication
- A single MIPS-like processor
- 32-KByte data cache
- 96-KByte instruction cache

The distance and wire delay between neighboring two tiles are determined by the dimensions of a tile. In Raw microprocessor design, the dimensions of a tile are adjusted so that the neighboring two tiles can communicate each other within a single cycle. Thus, the number of tiles integrated on a chip is increased as the process is scaled down. Authors in [37] suggest the future possibilities of more than hundreds tiles.

A Raw microprocessor consists of sixteen tiles. Each tile has its own computational resources and it is connected to four neighboring tiles (i.e., north, south, west, and east) via on-chip routers. Four 32-bit full-duplex links are used for router-router links and router-core links, respectively. Up to 12,500 wires are required for the network links; thus the interconnection area consumes the largest area in the Raw chip.

The Raw architecture has four different networks. Two of four networks are used as static networks, in which all packet transfers are statically scheduled at compile time of the applications. On the other hand, the remaining two networks are called dynamic networks, in which packets are dynamically generated according to the input data or computational results. Each tile (except for tiles in chip edges) is connected to four neighboring tiles via point-to-point links. These links are the longest wires that introduce the largest wire delay in the Raw microprocessor; thus the largest wire delay is predictable and it would not be affected even though the number of tiles integrated on a chip is further increased.

Static routers control two static networks. They are used to transfer single-word operands and data streams in an in-order manner. These data transfers are completely pre-scheduled at the compile time of the applications. Static routers are fully pipelined with five pipeline stages. Since there are two static networks, each static router in a tile has two crossbar switches. And the crossbar switch is connected to another crossbar in the same static router, a MIPS-like processor in the same tile, and four neighboring tiles.

Dynamic networks deal with message transfers that cannot be predicted at the compile time, such as packet transfers induced by cache misses and interrupts. The dynamic networks employ dimension-order routing and wormhole switching, as a routing algorithm and a switching technique, respectively. A packet consists of up to 31 flits including a header flit in which its destination address

is stored. A router takes one cycle to forward a single flit when the packet goes straight via the router, while it takes two cycles to forward it when it turns to another dimension at the router. Two cycles are also required to forward a flit from (to) computational resource in the same tile. Therefore, communication latency required to transfer a single flit from a source tile to a destination tile is $(2 + x + 1 + y + 2)$ cycles, where x and y are the hop counts in x - and y -dimension, respectively. That is, this flit transfer takes two cycles for sending the flit from the source processor to the first router, x cycles for traveling in x -dimension, one cycle for turn, y cycles for traveling in y -dimension, and two cycles for sending it from the last router to the destination processor.

2.6.2 Intel Single-chip Cloud Computer (SCC)

Intel's Tera-scale Computing Research Program created a research chip called the Single-chip Cloud Computer (the SCC) to study many-core CPUs, their architectures, and the techniques used to program them. Figure 2.9 shows a stylized view of the SCC chip. You can see the 24 tiles arranged in a 6×4 array. There is a router associated with each tile. Four memory controllers go to off-die (but on-board) DDR3 memory.

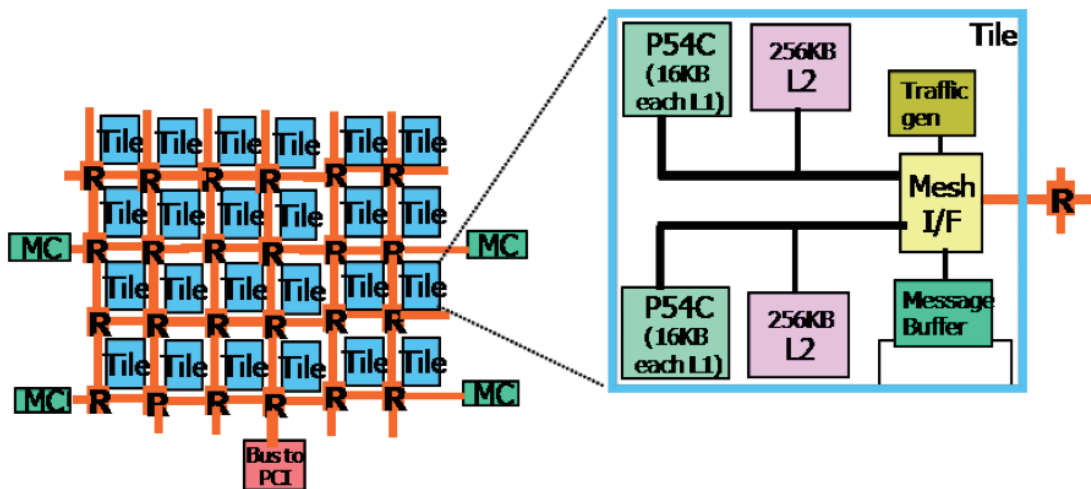


Figure 2.9: SCC top-level and tile architecture

The Management Console (MC) is a 64-bit PC running some version of Linux. Intel Labs provides software that runs on the Management Console to manage the SCC chip. Key features of this software are the ability to load a Linux image on each core or a subset of cores, to read and modify SCC configuration registers, and to load programs on the SCC cores. You can also write programs that run on the Management Console and configure the SCC platform. Running Linux on the cores is the most common configuration, but certainly not the only one. Some users may be interested in one of the research operating systems being developed for many-core systems.

The SCC has 24 tiles, 48 cores, two cores to a tile. Each core has L1 and L2 caches. The L1

caches are on the core; the L2 caches are on the tile next to the core. Each core has a 16KB L1 instruction cache and a 16KB L1 data cache. Each core's L2 cache is 256KB. Figure 2.9 also shows an overview of an individual tile.

Each tile also has a message passing buffer (MPB). This message passing buffer is 16KB. Its memory is shared among all the cores on the chip. With 24 tiles, the SCC provides 384KB of message passing buffer. When a message-passing program sends a message from one core to another, internally it is passing the data through the message passing buffers on the chip. The Traffic Generator is a unit used to test the performance capabilities of the mesh by injecting and checking traffic patterns and is not used in normal operation. The Mesh Interface Unit (MIU) connects the tile to the mesh. It packetizes data out to the mesh and de-packetizes data in from the mesh. It has buffers that queue data for flit organization. It controls the flow of data on the mesh with a credit-based protocol. It uses a round-robin scheme to arbitrate between the two cores on the tile.

Intel Labs believes the SCC is an ideal research platform to help accelerate many-core software research.

2.7 Summary

In this chapter, we surveyed on chip networks from the aspects of on chip router architecture, network topologies, routing algorithms and low power technologies for network on chip.

Firstly, a typical wormhole router architecture was introduced including five router components, pipeline of the on chip router.

Next, some popular network topologies was introduced. The most visible classification of network topology is regular topology and irregular topology. Regular topology can be further classified into direct network and indirect network. In a regular topology, routers and cores are connected based on a certain pattern that can be represented mathematically by a set of functions. Otherwise, it is referred to as irregular topology.

Then, existing routing algorithms and several low power technologies for on chip network were surveyed.

Finally, a real NoC-based processor called RAW was introduced to check how the above network on chip technologies are applied.

We surveyed only two-dimension network on chip, but three-dimension IC and Network on chip will be introduced in next chapter.

Chapter 3

Wireless Three-Dimension NoCs

This chapter gives an introduction of three-dimension IC (3D IC) and three-dimension network on chip (3D NoC). Main stream 3D interconnection technologies will be introduced. Inductive-coupling technology will be introduced as a keynote. At last, a real wireless 3D NoC prototype chip will be shown.

3.1 3D IC

Due to the increase of design cost of custom System-on-Chips (SoCs) in recent process technologies, System-in-Packages (SiPs) or 3D ICs that can select and stack necessary known-good-dies in response to given application requirements have become one of hopeful design choices. The wire delay is one of crucial problems in modern VLSI designs [63]. An attractive solution to the wire delay problem is three-dimensional IC technology that stacks multiple wafers or dice using vertical interconnects [21,64,65]. The current concept of NoCs is being extended to 3D NoCs [30,31,66,67]. Figure 3.1 shows examples of a 2D IC and a 3D IC that has equivalent resources to the planar one. Obviously, size of each tier (a tier refers to a wafer or a die in a 3D IC) can be downsized according to the number of tiers stacked together. Assuming that four tiers are stacked in a 3D IC as shown in the Figure 3.1, each wafer area is reduced to one quarter of the size and the length of the longest wire in a wafer becomes one half of the size compared with the planar one. Since the wire delay is related to the square of the wire length, the maximum delay is reduced to one quarter of the time. Consequently, the number of repeater buffers inserted in the wire can be reduced as well.

3.1.1 3D Interconnection Technologies

Various interconnection techniques have been developed to connect multiple chips in a 3D IC package: wire-bonding, micro-bump [19,20], wireless (e.g., capacitive-coupling and inductive-coupling) [21–24] between stacked dies, and through-silicon via (TSV) [21,25] between stacked wafers.

Interconnection technologies for 3D ICs are classified into wired and wireless approaches. In this section, both approaches are surveyed and the advantages of inductive-coupling are discussed.

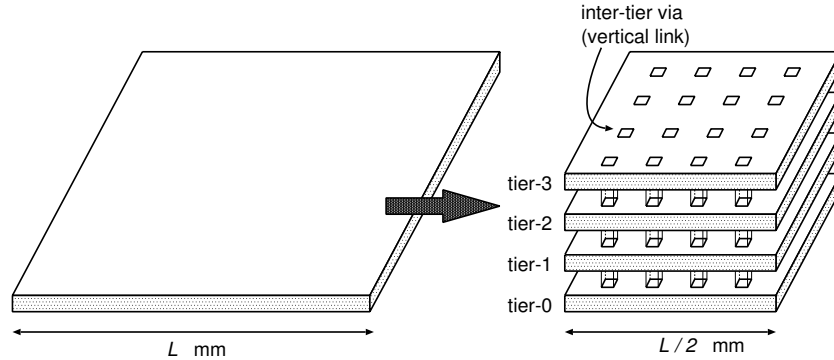


Figure 3.1: 2D IC (left) and 3D IC (right)

3.1.1.1 Wired 3D IC Technologies

Wire-bonding is a die-level interconnection that uses bonding wires. It is currently utilized for SiPs since no special process technology is required. However, because only the edges of a chip can be used for the bonding, the number of wires and their density are limited. Also, the long bonding wires can cause considerable wire delays for communication.

Micro-bump [19, 20] is a die-level interconnection that uses solder balls formed on the surface of each chip (Figure 3.2(a)). Thus, this approach is typically used for the face-to-face connection of only two dies. Although the face-to-back connection using long wires between solder balls of two chips is also possible, it introduces considerable wire delays since the wires are routed between solder balls on one chip and those on another chip via an outer edge (i.e., periphery) of the chip.

Through-silicon via (TSV) [21, 25] is a wafer-level interconnection that uses via-holes formed through multiple wafers (Figure 3.2(b)). More than two wafers can be connected by using the face-to-back connection. The footprint size of each TSV is small, so a high-density implementation is possible. However, the fabrication cost may increase due to the extra process steps for forming the TSVs. In addition, since wafers are connected via TSVs, the number and order of the wafers cannot be changed after the chip fabrication.

In summary, although these wired approaches have been utilized in real products (and especially TSV offers a high density implementation), they are not flexible enough for customizing the chips in a package in response to application requirements after the fabrication.

3.1.1.2 Wireless 3D IC Technologies

Capacitive-coupling [22] is a die-level wireless interconnection between two metal plates each of which being formed on the topmost metal layer of a chip (Figure 3.2(c)). It can connect two known-good-dies without wire connections. However, only the face-to-face connection is allowed

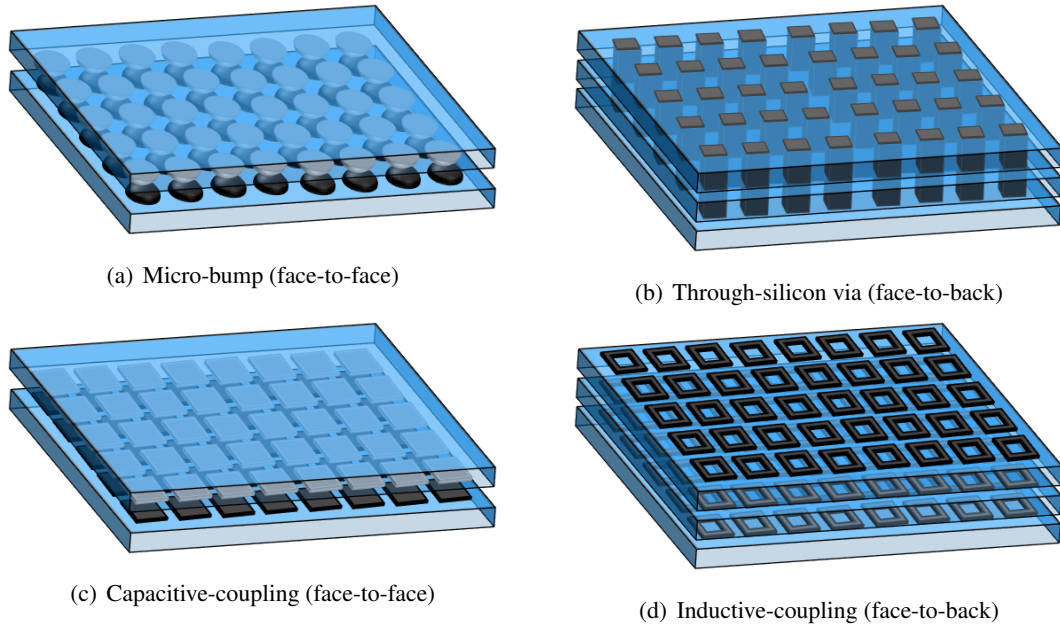


Figure 3.2: Three-dimensional (3D) IC technologies.

since its communication distance is typically shorter than chip thickness. Thus, the number of chips connected in a package is limited to only two.

Inductive-coupling [21, 23, 24, 68] is a die-level wireless interconnection that uses square coils as data transmitters (Figure 3.2(d)). The coils or inductors can be implemented with common metal layers of the chip; thus no special process technology is required for them. In addition, by increasing the number of turns of the coils using metal layers, the transmission gain can be enhanced and communication distance longer than the chip thickness is achieved. And the communication distance is mostly related to the coil size. Thus, more than two known-good-dies can be connected by using a face-to-back connection.

Although many recent studies on 3D IC architectures focus on the TSV that offers the largest interconnect density, we consider inductive-coupling that can connect more than three known-good-dies without wired connections; this is because inductive-coupling provides a large degree of flexibility to build the target 3D ICs, such as adding, removing, and swapping the chips in a package after the chip fabrication, like building blocks.

Compared to the other approaches, inductive-coupling is more feasible for 3D systems whose components should be customized for given application sets, because it has both the flexibility by contact-less connection and extensibility for stacking any number of known-good-dies. That is, addition, removal, and swapping of processor or cache chips are possible after the chip fabrication in response to the application set. Recently, the inductive-coupling techniques have been improved; the contact-less interface without an electrostatic-discharge (ESD) protection device achieves a high speed of more than 1GHz with a low energy dissipation (0.14pJ per bit) and a low bit-error rate

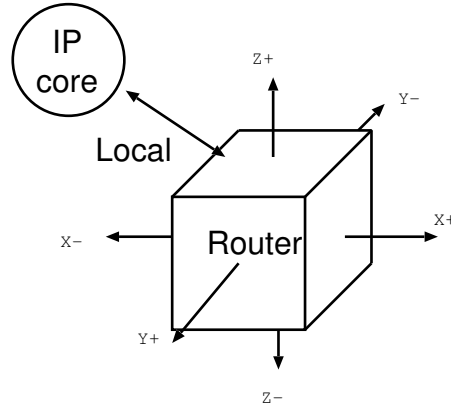


Figure 3.3: Router for mesh-based 3D NoCs.

($BER < 10^{-12}$) [23].

3.1.2 Topologies for 3D NoC

One of the well-known 2D NoC architectures is the 2D Mesh. This architecture consists of an $m \times n$ mesh of switches interconnecting IP blocks placed along with them. It is known for its regular structure and short interswitch wires. From this structure, a variety of 3D topologies can be derived. The straightforward extension of this popular planar structure is the 3D Mesh. Figure 3.1 shows an example of 3D Mesh-based NoC. It employs seven-port switches: one port to the IP core, one each to switches above and below, and one in each cardinal direction (North, South, East, and West), as shown in Figure 3.3.

As mentioned on Chapter 2, ring-based 2D NoC is special case of mesh-based 2D NoC.

Two on chip routers on same chip are connected with each other only when they are on the top or bottom chip. The routers belong to chips other than top or bottom ones are connected with upper or lower routers. Thus, these routers form a ring topology, as shown in Figure 3.4. Note that, it employs a five-port router, because there are only two ports in horizontal direction.

For the irregular-based topology, we will discuss it in Section 3.2.

3.2 Wireless 3D NoCs

First, conventional 2D NoC and 3D architectures are illustrated. Then, the wireless 3D NoC architecture and its requirements for the network part are introduced.

3.2.1 Baseline 2D and 3D NoC Architectures

NoCs are now a widespread architecture for high performance computing [69–75]. Figure 3.5(a) illustrates an example of NoC inspired by [70]. The chip is divided into sixteen tiles, and each tile has a single processor or four shared L2 cache banks, in addition to a single on-chip router. Red boxes

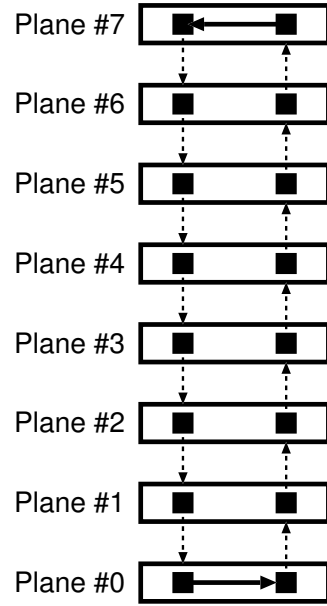


Figure 3.4: Ring-based 3D NoC.

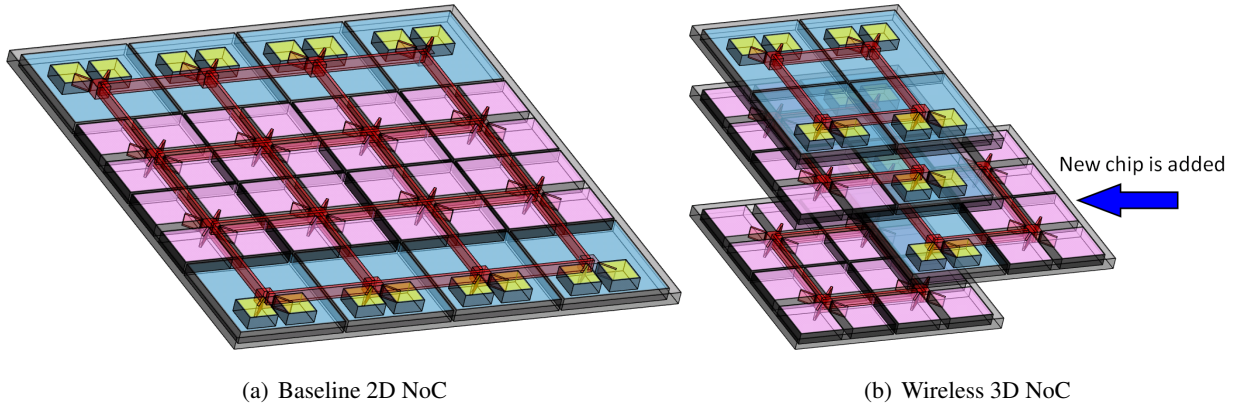


Figure 3.5: 2D and 3D NoC architectures.

and lines show on-chip routers and their links. Light blue boxes show processors. Yellow boxes show L1 instruction and data caches. Pink boxes show unified L2 cache banks. Wireless inductive-coupling is used for vertical links in wireless 3D NoC. Main memory modules and their controllers are connected to four edges of the chip (not shown in this figure). These tiles are interconnected via their routers. All L2 cache banks are shared by all processors; so the Static Non-Uniform Cache Architecture (SNUCA) [76] with a cache coherence protocol [77, 78] is assumed in this paper.

A natural extension of the baseline 2D NoC is the 3D NoC, in which processor, SRAM, and DRAM chips are stacked in a single package, and various organizations have been investigated [19, 31, 79–81]. 3D integration enables us to integrate multiple chips fabricated with different process technologies (e.g., logic and DRAM) into a single package in order to reduce the wire length, mitigate the pin count problem, and improve the performance. In addition, 3D NoC architecture [26] is a promising solution to reduce the communication latency. Regarding the 3D network architecture, the

network topology [28, 29], router architecture [30–32], and routing strategy [33] have already been extensively studied.

Most studies on 3D NoCs to date, however, assume the micro-bumps or TSVs (i.e., wired solution) for their inter-chip communication. None of them except [34] focuses on the wireless approach, which offers a higher degree of flexibility to add, remove, and swap the chips after the chip fabrication.

3.2.2 Wireless 3D NoC Architecture

The flexibility of wireless inter-chip communication is attractive for a wide range of 3D NoCs, because an NoC is collection of heterogeneous hardware components, such as processor, cache, and memory chips. In addition, their resource utilization is highly dependent on the target application set.

Figure 3.5(b) illustrates the proposed wireless 3D NoC architecture. In this example, the baseline 2D NoC is divided into four chips, each of which has four tiles: four processor tiles or four cache tiles. Wired links are used for a horizontal network that connects the four tiles on each chip, while wireless inductive-coupling links are used for the inter-chip network that connects four chips vertically. Memories and their controllers are connected to the bottom chip (not shown in this figure). We assume this simple 3D SNUCA as a baseline for simplicity, although various optimization techniques for 3D NoCs have been discussed [79, 80].

Three chip types are illustrated in this figure. The bottom (first) chip that consists of sixteen L2 cache banks is denoted as a cache chip. The second and third ones that consist of two processors and eight L2 cache banks are denoted as a hybrid chip. The top (forth) one that consists of four processors is denoted as a processor chip. Typically, application performance is limited by either computational power or memory bandwidth; thus the applications can be classified into computation-bound and memory-bound applications [82, 83].

Depending on the target application set, the wireless 3D NoC using the inductive-coupling enables us to customize the number and types of chips stacked in a package after the chip fabrication. For example, more processor chips can be added for computation-bound applications, while more cache chips will be useful for memory-bound applications. This flexibility is attractive, because designing a new mask pattern for each set of applications is too costly in recent process technologies. In other words, the conventional wired approaches have no other choice but to equip abundant hardware components to cope with future applications at chip design time. This is the firm advantage of the wireless 3D NoCs over the conventional wired approaches.

3.2.3 Network Requirements for Wireless 3D NoC

As discussed earlier, a wireless 3D NoC is a collection of various hardware components, such as processors, SRAMs, and DRAMs. They are based on different process technologies and fabricated

from different vendors. Therefore, irregular topology will be generated. As long as the chip satisfies minimum requirements (e.g., performance, chip size, and vertical link pitches), its detailed horizontal architecture (e.g., processor, memory, and horizontal NoC parameters) should be left to each vendor. A key challenge of the wireless 3D NoCs is to stack such post-fabricated hardware components, such as processor and memory chips, coming from different vendors so that these components can cooperate with each other, with minimum design rules.

We define the minimum design rules as follows:

- 1) All chips have inductors at pre-determined locations in order to form vertical links.
- 2) All the inductors are connected to input and output ports of on-chip routers in order to form a 3D NoC.
- 3) All the on-chip routers support the plug-and-play routing protocol will be proposed in Chapter 5.
- 4) All the hardware components are topologically reachable to at least an on-chip router that has the vertical link.

3.3 Plug-and-Play Routing Scheme

In this section, we introduce a plug-and-play routing scheme for wireless 3D NoCs with irregular topology.

A wireless 3D NoC is a collection of various hardware components, such as processors, SRAMs, and DRAMs. They are based on different process technologies and fabricated from different vendors. Hence, one cannot expect any pre-determined network topology for each chip. For example, one chip may have a mesh-based NoC while another one may have a ring-based NoC. In an extreme case, it is possible to have vertical links only (i.e., no intra-chip links), if the hardware components on the same chip communicate with each other rarely. In this case, their communication must go through another upper or lower chip via vertical links. Such routing paths cannot be expected to be known at the chip design time since a chip vendor cannot know its topological structures of upper and lower chips. Thus, conventional static routing strategies for NoCs are not enough for the wireless 3D CMPs.

To address the problems mentioned above, a plug-and-play routing protocol that exchanges the network information at boot time in order to establish efficient global routing paths is needed for the wireless 3D CMPs. The routing paths are established, based on spanning trees optimized for a given traffic pattern by using the following steps.

- **Step 0:** Traffic pattern of a target application set is pre-analyzed in order to generate a traffic trace that records the injection time, source node, destination node, and packet size of every packet transfer.

- **Step 1:** At boot time, the control protocol collects the network information including the planar topology and number of virtual channels from all chips. In addition, the traffic trace of the target application set (pre-analyzed at Step 0) is fed to the network as a hint for the routing optimization.
- **Step 2:** According to the network topology and traffic trace collected, the routing protocol generates one or more spanning trees for each message class¹ so as to minimize the average hop count and improve the application execution time.
- **Step 3:** The control protocol configures the routing tables of all routers in the NoC.

First, the control protocol that collects the network information from all chips is proposed. Second, the routing algorithm for 3D CMPs without the route optimization technique is illustrated. Finally, the optimization technique is proposed.

3.3.1 Control Protocol

Routing path calculation is performed by a single processor inside or outside a 3D CMP, based on the topological information collected from all chips. We assume that each vertical link controller knows the topological information of the chip. This assumption is reasonable since the horizontal information of each chip is known priori and can be embedded to the vertical link controller at the design time.

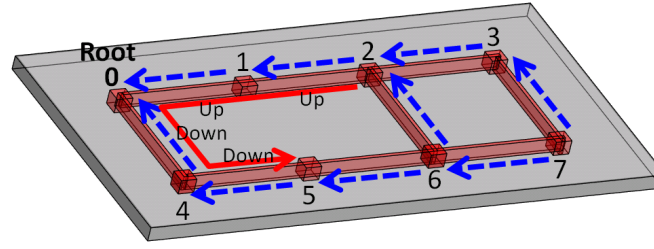
The global routing paths are established by using the following procedure at the boot time or configuration time. 1) A processor inside or outside the 3D CMP is selected as “routing computation node” manually. 2) The routing computation node broadcasts a “join” message to every vertical link controller in the 3D CMP to request their topological information. 3) The vertical link controllers send their topological information to the routing computation node. 4) The routing computation node calculates the global paths by a deadlock-free routing algorithm discussed in Section 3.3.2. 5) The routing computation node distributes the routing table information to every router in the 3D CMP.

In the test chip [34], each node has control registers that store various information, such as chip ID, number of chips, and routing modes. The control registers now are extended to indicate whether it is the routing computation node.

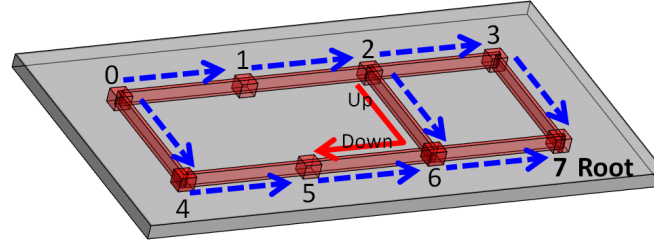
3.3.2 Routing Protocol

The wireless 3D CMP consists of a collection of processor and memory chips from different vendors, and a chip vendor cannot know the network structures of its upper and lower chips. Thus, there is no other choice but to assume that a given wireless 3D NoC represents an irregular 3D network topology.

¹A message class is a group of messages that can share the same virtual channel without introducing protocol deadlocks.



(a) Root node 0



(b) Root node 7

Figure 3.6: Up*/Down* routing paths with different spanning tree roots.

The network requirements for the wireless 3D CMP are close to those of System-area-Networks (SANs) that employ irregular network topologies. Up*/Down* routing is the most popular deadlock-free routing algorithm for irregular networks, and has been used in Autonet [14] and Myrinet [84]. To guarantee connectivity and deadlock-freedom for irregular networks, it exploits a spanning tree based directed graph in which up or down direction is assigned to each network channel. The proposed plug-and-play routing protocol with spanning trees optimization is built upon the Up*/Down* rule [14] for ensuring the deadlock-freedom.

Up*/Down* routing avoids deadlocks in irregular topologies without virtual channels, based on an assignment of direction to network channels [14]. As the basis of the assignment, a spanning tree whose nodes correspond to routers in the network is built. The “up” end of each channel is defined as follows: 1) the end whose node is closer to the root in the spanning tree; 2) the end whose node has the lower node ID number, if both ends are on nodes at the same tree level. Figure 3.6 shows two examples of spanning trees, each of which has a different spanning tree root (node 0 or 7). A legal path must traverse zero or more channels in the up direction followed by zero or more channels in the down direction, in order to guarantee deadlock-freedom while keeping the network reachability. For example, a red line in Figure 3.6(a) shows a routing path from node 2 to node 5. This routing path uses two up direction channels followed by two down direction channels.

Needless to say, the location of a spanning tree root affects the hop count and link utilization significantly. In Figure 3.6(a), for example, the routing path from node 2 to node 5 employs a non-minimal path, because the minimal path is prohibited by the spanning tree whose root is located at node 0; thus four hops are required to send a packet. In Figure 3.6(b), on the other hand, the minimal path is allowed by the spanning tree whose root is located at node 7; thus only two hops are required.

The next section proposes a route optimization technique that employs multiple spanning tree

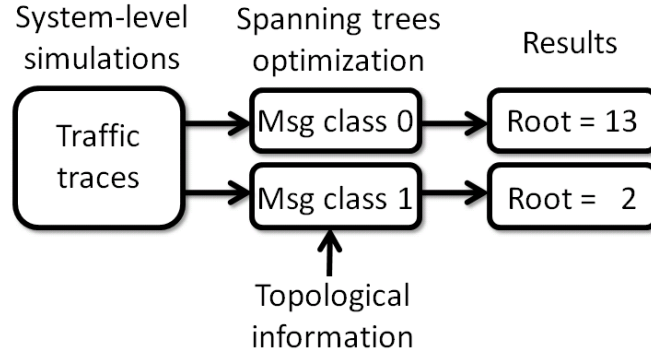


Figure 3.7: Design flow of spanning trees optimization.

roots, each of which is optimized for a class of messages.

3.3.3 Spanning Trees Optimization

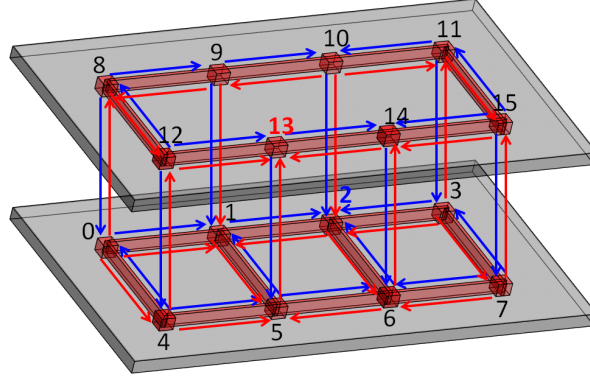
In general, Up*/Down* routing does not require any virtual channels to avoid structural deadlocks. However, we assume a shared cache CMP architecture, in which a cache coherence protocol is running on the NoC. Since such cache coherence protocols typically introduces end-to-end or request-and-reply protocol deadlocks, they are divided into multiple message classes, each of which uses one or more dedicated virtual channels.

Table 3.1 below shows an example of virtual channel assignment for a directory-based cache coherence protocol. Three virtual channels are used for the protocol messages in order not to introduce any end-to-end protocol deadlocks. The network resources (i.e., buffers) of one virtual channel and those of another virtual channel are logically separated and do not introduce any cyclic dependencies between them; thus we can assign different spanning trees for them, depending on their traffic pattern. For example, request messages from L1 caches (i.e., processors) to L2 cache banks move in the opposite direction of the reply messages from the L1 caches to the L2 caches. Since Up*/Down* routing introduces non-minimal paths and imbalanced link utilizations depending on the spanning tree root and traffic pattern, their optimal spanning tree roots that can minimize the hop count will be different. Thus, an optimal spanning tree root should be selected for each message class in order to improve the performance. An optimal combination of multiple spanning trees can be assigned to each message class in order to further shorten the hop count. However, its performance improvement is marginal in typical CMP sizes, such as 8 to 64 tiles, compared to the additional hardware cost.

Figure 3.7 shows a design flow of the proposed spanning trees optimization technique. First, the application traffic pattern on a target 3D CMP is pre-analyzed by system-level simulations in order to generate a traffic trace. Then, an optimal spanning tree root is selected for each message class in order to minimize the average hop count of the trace and improve the application execution time;

Table 3.1: Virtual channel (VC) assignment for a cache coherence protocol.

Message class 0 (VC0)	Request <i>from</i> {L1\$} <i>to</i> {L2\$ bank} Request <i>from</i> {L2\$ bank} <i>to</i> {L1\$}
Message class 1 (VC1)	Request <i>from</i> {L2\$ bank} <i>to</i> {directory controller} Request <i>from</i> {directory controller} <i>to</i> {L2\$ bank}
Message class 2 (VC2)	Response <i>from</i> {L1\$ or directory} <i>to</i> {L2\$ bank} Response <i>from</i> {L2\$ bank} <i>to</i> {L1\$ or directory}

**Figure 3.8:** Optimization result.

towards this end, we use the following cost function:

$$Cost = \sum_s \sum_d W_{s,d} H_{s,d}, \quad (3.1)$$

where $W_{s,d}$ denotes number of packets from source router s to the destination router d in the trace and $H_{s,d}$ denotes hop count from s to d . Figure 3.8 shows an example of spanning trees optimization for a 16-tile 3D CMP that defines just two message classes for simplicity. Packets that belong to the message class 0 are routed based on the red spanning tree whose root is node 13, while those belonging to the message class 1 are routed based on the blue spanning tree whose root is node 2. Note that calculation time for the best spanning tree root for each message class is small: 0.008sec for 8-tile, 0.020sec for 16-tile, 0.315sec for 32-tile, and 9.376sec for 64-tile with AMD Opteron processor 2.7GHz.

3.4 A Prototype Wireless 3D NoC Chip

The inductive-coupling is versatile and has been applied to various 3D NoC systems, such as multi-core processors [85] and dynamically reconfigurable processors [68]. In this section, some feasibility issues of inductive-coupling are illustrated through an example.

A prototype 3D NoC reported in [34] employs inductive-coupling links to form a vertical uni-directional ring network to connect more than three chips. It was fabricated by using Fujitsu 65nm

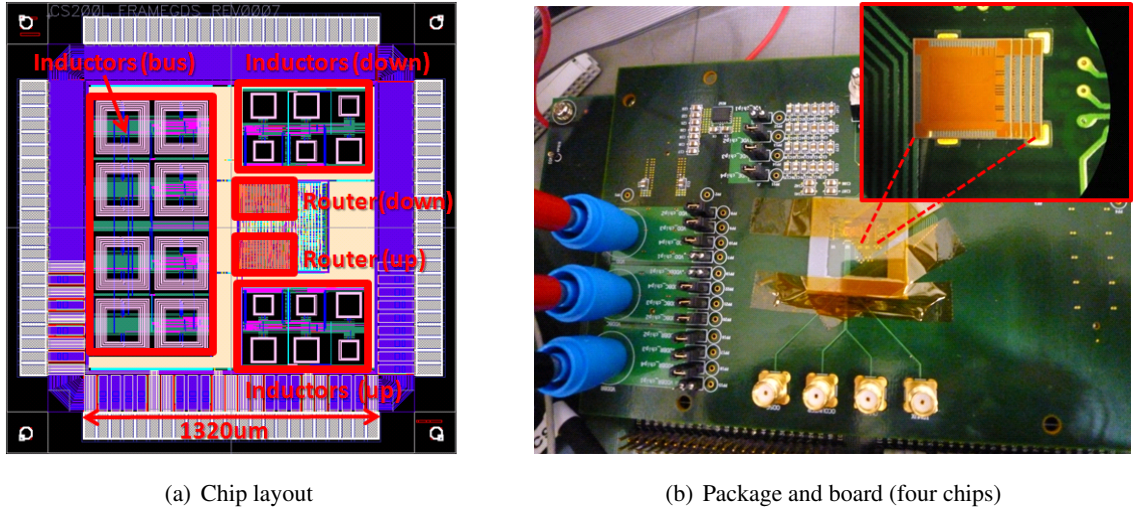


Figure 3.9: Prototype implementation of inductive-coupling 3D NoC [34].

CMOS process.

Figure 3.10 shows the side view of four chips in a package. To place the transceiver and receiver of the neighboring two chips at the same location, each chip is placed with a certain gap. This space is also used for bonding of power and ground wires. On the right side of Figure 3.10, Cube-0 chip architecture (top view) to show the details of chip organization. Figure 3.9 shows the chip layout and prototype system that stacks four chips.

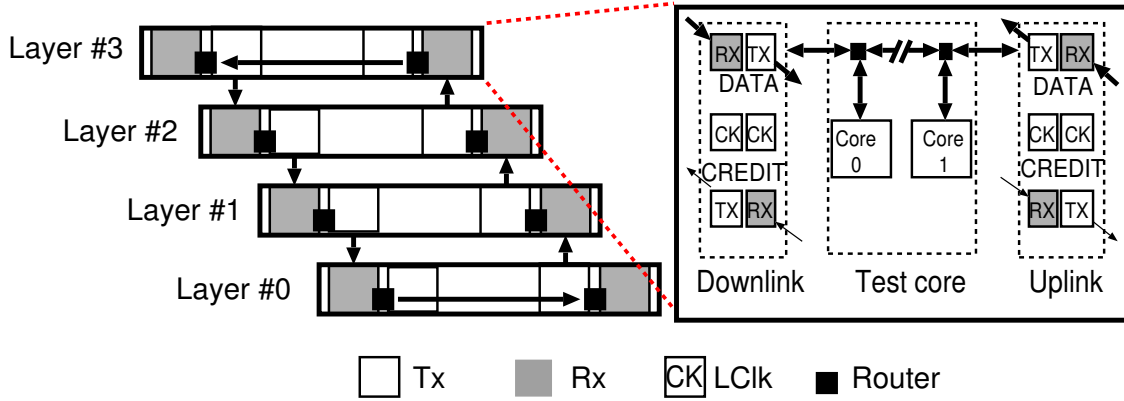


Figure 3.10: Four chips stacked in a package

As shown in Figure 3.9(a), each chip has two pairs of on-chip router and inductor, and one pair is used for uplink and another is used for downlink. To form a single uni-directional ring, two routers only on the bottom and top chips are connected via horizontal wired links. In addition to the ring network, a shared vertical bus is also implemented using inductive-coupling for the comparison purpose. The chip architecture is described in detail as follows:

- 1) **The Test core** consists of two cores and two routers. The core has a packet generator and

a packet receiver with a 45-bit packet counter. One router is connected to Downlink data transceiver and another for the Uplink. Although every chip has an on-chip wired link that connects two routers on the chip, only the top and bottom chips actually use one uni-directional link to form an inter-plane vertical ring.

2) The Downlink and 3) Uplink consist of two pairs of TX and RX channels, respectively. One TX and RX pair is used for 35-bit flit transfer, while the other pair is for 2-bit credit back used for the flow control. The wireless data are serially transferred at the double communication rate of the 4 GHz local clock shared by two neighboring chips. Thus, each TX channel is capable of transmitting a 35-bit flit during each 200 MHz system clock. The implementation of the inductors is shown in the following paragraph.

4) The Vertical bus consists of four TX/RX channels and four clock channels. The TX/RX channel works in TX mode when a time-slot is assigned to the chip, and in RX mode otherwise. As mentioned, an 8-cycle time-slot is periodically assigned to each chip for vertical broadcasting.

Inductors are implemented as a square coil with metal in common CMOS layout (Figure 3.9(a)). The data modulated by a driver are transferred between two coils placed at exactly the same position of two stacked dies, and it is received at the other chip by the receiver. Here, a set of a driver and an inductor for sending data is called TX channel, while the one with a receiver and an inductor is called RX channel.

In Figure 3.11, we can see the block diagram of inductors. One end of vertical link is made up with metal inductor, Tx/Rx configuration module, and clock counter. The equivalent circuit of inductive-coupling vertical links is also shown as Figure 3.12.

If a TX channel is placed at the same location of multiple RX channels in different chips, data multicast can be used. On the other hand, stacked multiple TX channels at the same location cannot send the data simultaneously in order to avoid the interference.

The footprint of an inductor ranges from $30\mu\text{m} \times 30\mu\text{m}$ [23, 24] to $150\mu\text{m} \times 150\mu\text{m}$ [21], depending on the process technology and communication distance (i.e., chip thickness).

Clock and power supplies are provided by using bonding wires in this prototype for simplicity, although wireless power transmission is already possible. A 200MHz clock is used for the on-chip routers, while a 4GHz clock is used for the inductors. The wireless data are transferred serially using the doubled communication rate of the 4GHz clock shared by neighboring two chips. Thus, communication bandwidth is up to 8Gbps per channel.

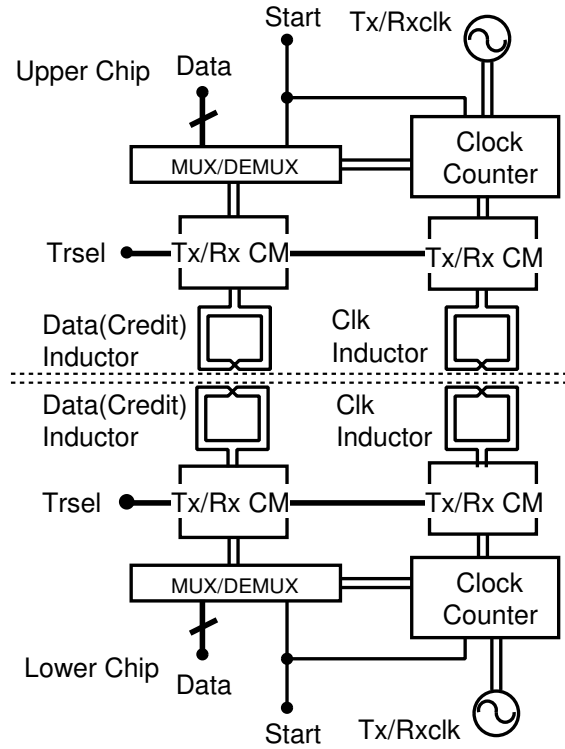


Figure 3.11: Structure of a vertical link

3.5 Summary

The three-dimensional Network on Chip (3D NoC) is an emerging research topic. 3D ICs stack several smaller wafers or dies in order to reduce the wire length and wire delay.

In this chapter, we surveyed several 3D IC interconnection technologies, especially inductive coupling technology. Because we believe it provides the flexibility to build the target 3D ICs by adding, removing, and swapping chips in a package after the chips have been fabricated, like building blocks.

We also extended our discussion from 2D NoC to 3D NoC. Router architecture and topology for 3D NoC was introduced. Then we paid close attention on inductive coupling based wireless 3D NoC, described network design rules.

At last, a real prototype wireless 3D NoC (Cube-0) using inductive coupling technology was introduced, including the chip architecture and inductor (vertical link) structure.

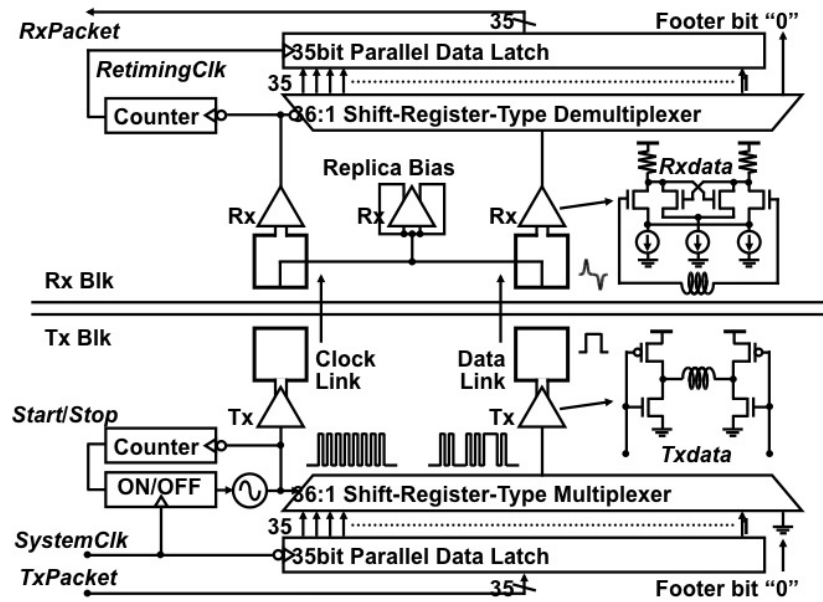


Figure 3.12: Circuit of inductive coupling link

Chapter 4

Power Saving Methods for Ring Wireless 3D NoC

This chapter presents about power consumption optimization for inductive coupling based wireless 3D Network on Chip. Power consumption optimization is essential issue for inductive-coupling based wireless 3D NoCs. A novel optimization scheme is proposed, also applied to two kinds of network on chip, followed by power consumption and performance evaluation for both networks.

4.1 Power Consumption Analysis for Vertical Links

In this thesis, the inductive coupling link designed for a test chip Cube-0 [86] is used as a target vertical link.

First, the power consumption of the target system (the vertical links of Cube-0) is analyzed. SPICE simulation is used for evaluation, since it is difficult to analyze power in real chip. By utilizing a 65nm BSIM model in SPICE simulation, data rate and power dissipation for inter-chip communications (i.e. inductive coupling links) of four stacked chips were calculated. Cube-0 was implemented with e-shuttle 65nm CMOS process, and mounted on a test print circuit board that was also developed to measure this real chip. The power consumption by the chip is calculated from the Ampere-meter and Voltmeter attached to the circuit board. By comparison, the total power of the simulation was almost the same value from a real chip. The power consumption of vertical link, on-chip router and on-chip core are shown in Table 4.1.

Table 4.1: Power consumption of on chip network components [uW]

Vertical link	Active	Idle	Sleep
Data Transmitter	4600	2500	0
Data Receiver	3500	9.2	0
On-chip Router	1110	-	-
On-chip Core	70	-	-

In Table 4.1, “Active” represents the case when data is transferred through the wireless channel, while “Idle” shows the case there is no data on the channel. The on-chip router and on-chip core

are shown in Figure 3.10. Since the on-chip core only includes a packets generator and a packet receiver, the power consumption is much lower than on-chip router. As shown in Table 4.1, the power consumption of vertical link is much higher than the on-chip router and on-chip core. It continuously consumes a certain amount of power, regardless of packet transfers.

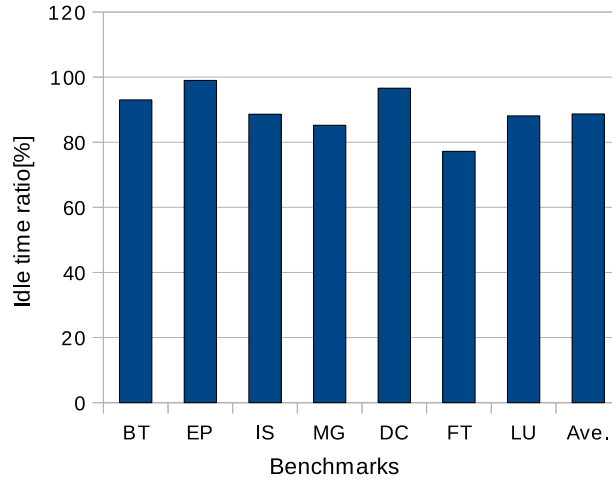


Figure 4.1: Idle time ratio in vertical links

Compared with traditional works [34, 87, 88] using power gating, we can clearly find that the power consumption of vertical link of wireless interconnection is large. For example, in our second prototype Cube-1 [89], a MIPS compatible CPU and an energy efficient accelerator CMA [90] are connected with a inductive coupling ring network. Since the power consumption of the CPU and CMA is about 15mW and 20mW, respectively, about 32% of total system power is occupied by the wireless NoCs. This is due to the big gap between operating frequencies for routers and vertical links (200MHz and 4GHz respectively). Moreover, vertical links continuously consume a certain amount of power, regardless of packet transfers. There is big waste to energy when the utilization of vertical links is low, which is a typical use case of 3D ICs that the most communications are within a chip while the communication between chips are not frequent. For example, Figure 4.1 shows ratio of idle time to application time in vertical links used in a full-system 3D CMP simulator which will be introduced in Section 4.4.0.1. For the vertical links, 86.7% of the application execution time are idle in average. The detailed values are also shown in Table 4.2.

Table 4.2: Idle time ratio in vertical links [%]

Benchmark programs	SP	MG	CG	BT	LU	Ave.
Idle time ratio	88.1	83.9	95.6	76.7	86.5	86.7

In “Sleep” mode, the transceiver is disabled by cutting the bias voltage supply. As shown in Figure 3.12, a simple on/off switch is connected to current source (bias voltage is mostly used as voltage controlled current source) at both ends of DxData (i.e. transmitter) and RxData (i.e. receiver).

We can shut down or wake up the transmitter and receiver by switching the current source. Unlike the power gating for digital circuits, it does not require energy overhead except the power used in the controller. Although it requires a certain time for wake-up, it is within a clock cycle for system clock (200MHz) of the digital part. Also, compared with the power gating in digital circuits, there is little overhead for sleep down and wake-up of the vertical link.

According to the above analysis, it is essential to use sleep mode aggressively for reducing the power consumption of wireless link. However, the problem is that once the link is in the sleep mode completely, no data can be transferred since it is difficult to provide power control network between different chips. In next section of this chapter, we propose methods for power control of wireless inductive coupling links.

4.2 Vertical On/Off Link Control Methods

In this section, we propose sleep control methods that shut down the vertical links depending on their utilizations.

Since it just stops the transceiver or receiver for the inductive coupling link, the state transition between sleep mode and active mode does not require energy overhead except the power consumption in the controller. It is like the clock gating used in digital circuits, and not related to power gating which requires energy overhead both for shutting down and activation. Moreover, the latency overhead of state transition between sleep mode and active mode is one clock cycle, same as the case between idle mode and active mode. As a result, we use sleep mode instead of idle mode which is employed in Cube-0. Note that, the analog part of inductive coupling link works at 4GHz while the digital part works at 200MHz clock. Although it takes a few clocks for stable working of the analog part, it can be available within one clock cycle of the digital part as the clock speed is much lower than that of analog part. All evaluations in Section 4.4 and Section 5.3 assume one clock cycle to activate inductive coupling links.

Two sleep modes are proposed here. Table 4.1 shows that the large part of power is consumed by the transmitter. So, if the transmitter for data (DTx) is shut down, a large part of power consumption can be saved. This mode is called partial sleep method (PSM). It only shuts down the data transmitter (DTx). Since other parts are running, the vertical link can be woken-up when the sender activates DTx in one clock cycle and sends the packets.

Another mode is the full sleep method (FSM). In the mode, the whole vertical link in the FSM including DTx, DRx, CTx, CRx, and LClk transceiver. Although the total analog part of wireless inductive link is shut down, the communication cannot be done until transceivers in the both chips are woken-up.

The sleep decision is based on the utilization of target vertical link and whether there are enough reachable paths to send wake up signals. We set a reasonable threshold utilization value first, then compare the link utilization within sample window (time slot) with the threshold when flit transmis-

sion completes. If the utilization is higher than the threshold, the state of the target vertical link is set from active to PSM. Otherwise, if there are paths to wake up the whole vertical, we set the state to FSM. If not, PSM is used again to only shut down the DTx. The decision to turn on a vertical link is basically the converse of the sleep mechanism and can be done as on-demand because of the property of fast on/off link transition. These two methods control the vertical links to switch between on and off dynamically when the system is run-time. Note that, here, “dynamic” does not mean dynamic power consumption.

In the following sections, we apply the proposed methods to two kinds of on chip networks: ring topology based network and irregular topology based network.

4.3 Vertical On/Off Link Methods for Ring Wireless 3D NoC

First, the following three network configurations are introduced for wireless 3D NoCs. Then, vertical link sleep control methods are proposed for these networks.

- **Uni-directional ring network with simplex vertical links (URSL)** is employed in Cube-0 which is the simplest case of our discussions in this paper. It is a single ring network where all data signals can flow in only one direction (counterclockwise or clockwise) through vertical links, as shown in Figure 4.2(a). That is, the configuration of inductors are fixed as a transmitter or receiver. This network has the lowest power consumption because only one pair of transceiver is used. However, the performance is worse than following two networks. The main reason is that data packets are transferred in one direction only, resulting in non-minimum packet transfers.
- **Bi-directional ring network with full-duplex vertical links (BRFL)** is an extension of URSL. It allows communication between neighboring chips in both direction simultaneously. The metal inductors are also pre-configured as a transmitter or receiver (i.e., they are not reconfigured at run-time). Thus, twice numbers of inductors used in the URSL are needed to form two uni-directional ring networks, as shown in Figure 4.2(b). However, twice the number of inductors undoubtedly will double the power consumption. Luckily, non-minimum routing is not introduced in this network. It has the highest performance among these three networks.
- **Bi-directional ring network with half-duplex vertical links (BRHL)** communicates for both directions, but only one direction at a time. That is, once one end begins receiving a signal from a transmitter, it must stay at the receiver mode while receiving the signal. Then, it can move to the transmitter mode. In Cube-0, the metal inductors can be reconfigured dynamically. It indicates that the vertical links can change their transmission direction like the red links, as shown in Figure 4.2(c). In this case, although non-minimum routing is not introduced because of its duplex property, transceivers need some clocks to be configured to opposite direction. That is,

BRHL has worse performance than BRFL, but better performance than URSL. In addition, the BRHL consumes more power than URSL due to dynamic reconfiguration power consumption. However, it still consumes less power than BRFL, because dynamic reconfiguration power consumption is quite small compared with transceiver itself.

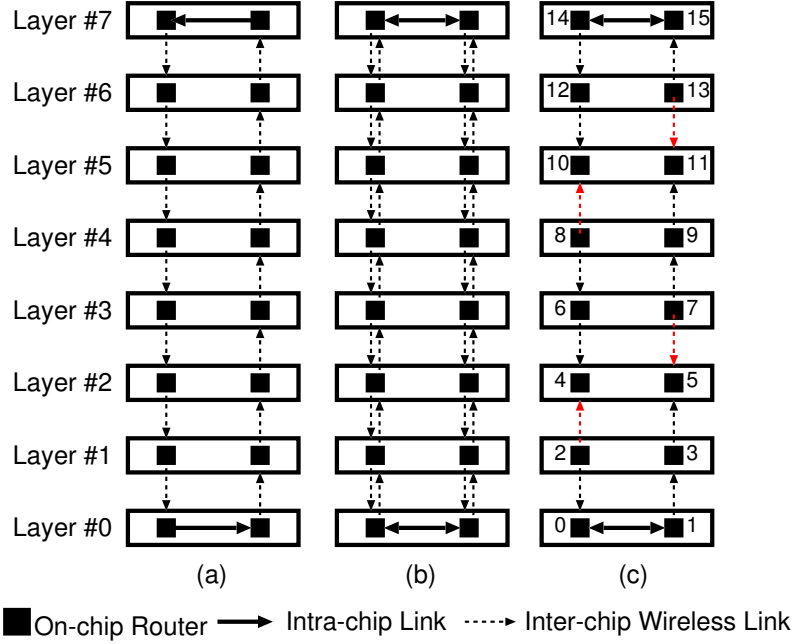


Figure 4.2: Three network configurations: (a) URSL, (b) BRFL, and (c) BRHL

For these three networks, we propose sleep control methods that shut down the vertical links depending on their utilizations. Such on/off control flows are summarized in Algorithm 1. The on/off link methods can be applied to above three ring networks. The details will be described in Sections 4.3.1, 4.3.2, and 4.3.3.

4.3.1 Sleep Control Method for URSL

The PSM is advantageous when the vertical link utilization is high. For instance, in Figure 4.3, when the red link utilization is higher than threshold, the sleep mechanism only shuts down the DTx immediately as long as no flit comes. The latency of sleep/wakeup is small, e.g., only one SClk cycle for Cube-0. As well as the fast vertical link shutdown, the vertical links also can be turned on within a very short time. For example, when a flit demands on to traverse a dormant vertical link, the DTx can be powered on within one SClk cycle.

In the case of lower vertical link utilization, the FSM is introduced for reducing the power consumption of URSL. In this method, the overhead of long latency to sleep/wakeup whole vertical link can be compensated by the low utilization. Figure 4.4 shows how the sleep and wakeup mechanisms work for the situation of lower vertical link utilization. We decompose Figure 3.10 into two figures.

Algorithm 1 Dynamic Vertical Link Shutdown Algorithm

```

1: if current_state == active && idata arrives then
2:     continue to sample;
3: else if current_state == active && !idata arrives then
4:     compute the utilization  $U_l$ ; compare with threshold  $U_t$ ;
5:     if  $U_l \geq U_t$  then
6:         shut down  $DTx$ ; current_state  $\leftarrow$  partial sleep;
7:     else
8:         shut down the whole vertical link; current_state  $\leftarrow$  full sleep;
9:     end if
10: else if current_state == partial sleep && idata arrives then
11:     current_state  $\leftarrow$  active; wake up the  $DTx$ ;
12: else if current_state == full sleep && idata arrives then
13:     if  $DRx$  current_state == sleep then
14:         send wakeup signal;
15:     else
16:         current_state  $\leftarrow$  active;
17:     end if
18: end if

```

That is, Figures 4.4(a) and (b) represent the side views of data links and credit links, respectively. The data and credit links have opposite directions. When the utilization of the red link is lower than a threshold, the sleep mechanism shuts down the whole red link. A sleep signal from layer 3 is sent to DRx and CTx which are located on the layer 2 via the red link one $SClk$ cycle in advance. Thus, the whole link can be shut down after one $SClk$ cycle. When we want to wake up the dormant vertical link, the green credit link (Figure 4.4(b)) is employed for sending wakeup signal to the other parts except DTx . However, before shutting down the whole red link, the state of the credit transceiver, which belongs to the blue vertical link, must be checked. The blue vertical link is between layer 2 and layer 3 as well as the red vertical link. As mentioned previously, the rule which can avoid the deadlock is introduced: if the credit transceiver is power on, we can shut down the whole red link. Otherwise, we have to shut down the DTx only. If both the whole links (red and blue) are shut down, they cannot be woken up. The reason is that they use each other's credit transceiver to wake up DRx , CTx , and $LClk$. That is, we cannot shut down the two vertical links which are between the same two layers simultaneously.

4.3.2 Sleep Control method for BRHL

The situation of BRHL is more complicated than that of URSL. We also analyze two cases: (1) high link utilization and (2) low link utilization. In this paper, the direction of vertical data link represents

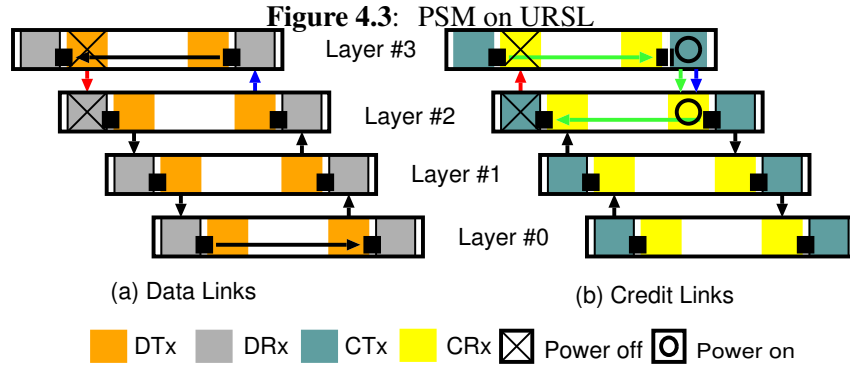
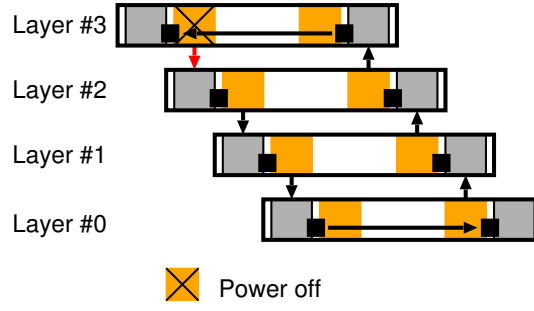


Figure 4.4: FSM on URSL

the one of whole vertical link. The direction of BRHL is configured to counterclockwise by default. In other words, all the vertical links are in the counterclockwise ring network if the clockwise link is not requested.

When the link utilization is high, PSM is also used here. The DTx is shut down by the sleep mechanism when no flit comes currently. As shown in Figure 4.5, the red link is the target vertical link to be shut down. When the lower utilization of the red link is found, the DTx on layer 2 is powered off. If coming flit demands the red link, we can power on the DTx within one SClk cycle. However, unlike the case of URSL, the following three issues should be addressed.

- 1) **How to configure the direction of the vertical links?** As Figure 4.5 shows, the direction of the green vertical link can be configured to clockwise (red link). Firstly, the green credit link is used to send a configuration signal to update the state of the DTx (CRx) to DRx (CTx). Then, DRx (CTx) can be configured as DTx (CRx). Thus, the green link can be changed to the red one. The configuration must wait until all flits on the target vertical link are transferred even though the latency is increased. Since the default direction of the vertical link network is counterclockwise, the configuration of the vertical link becomes the default state automatically as long as the transmission on clockwise vertical link finishes.
- 2) **Which links can be configured?** For Cube-0 prototype system which has four chips, we assume that a flit is transferred from the source node R4 to the destination node R6 (Figure 4.5). Thus seven hops are needed to reach the destination in the uni-direction ring network. In

bi-direction ring, on the other hand, only a single hop is needed. Although the hop number is reduced, there is a trade-off between distance and latency, because several cycles are required to configure the target vertical link. Hence, for Cube-0, if there are less than three hops in the routing path in the clockwise ring network, we use clockwise vertical links to transfer flits.

In general, zero-load latency of BRHL T is calculated as

$$T = \begin{cases} T_{ccw}, & \text{if the link direction is in default} \\ T_{cw}, & \text{if the link direction is clockwise} \end{cases} \quad (4.1)$$

where T_{ccw} is the latency for transferring a single flit from source node to destination node in the counterclockwise ring network, and T_{cw} is the latency for the clockwise ring network from the same source node to the same destination node. These values are calculated respectively as

$$T_{ccw} = (H_{ccw} + 1)T_{router} + H_{ccw}T_{link}, \quad (4.2)$$

$$T_{cw} = (H_{cw} + 1)T_{router} + H_{cw}T_{link} + H_{cw}T_{config}, \quad (4.3)$$

where H_{ccw} and H_{cw} are hop counts in counterclockwise ring network and clockwise one, respectively. T_{router} and T_{link} are latencies for transferring a header flit on a router and a vertical link, respectively.

The routing logic uses the above formulas to decide which ring network to be used. If $T_{ccw} > T_{cw}$, the clockwise ring is selected for transferring the flits. Otherwise, the default ring network is still used to transfer flits.

- 3) About the competition:** When two routers require the vertical link at the same SClk cycle, the router which uses clockwise ring network has higher priority than the one that uses counterclockwise ring. The rule is used in order to ensure smaller network latency as far as possible. That is, for instance, if router 4 and router 6 require the vertical link between layer 2 and layer 3, DRx on layer 2 can be configured to DTx as long as router 4 wants to use this link regardless of the link request from router 6.

When the vertical link utilization is lower than a threshold, the FSM shuts down the whole link. However, before shutting down the whole red link, we must check the state of credit link which belongs to the blue link as shown in Figure 4.6.

For the wakeup mechanism, there are also two situations:

- 1) If the flit that comes from R6 (counterclockwise ring network) wants to use the red link, we can use the green credit link to wake up the dormant DRx, CRx, and LClk (Figure 4.6).

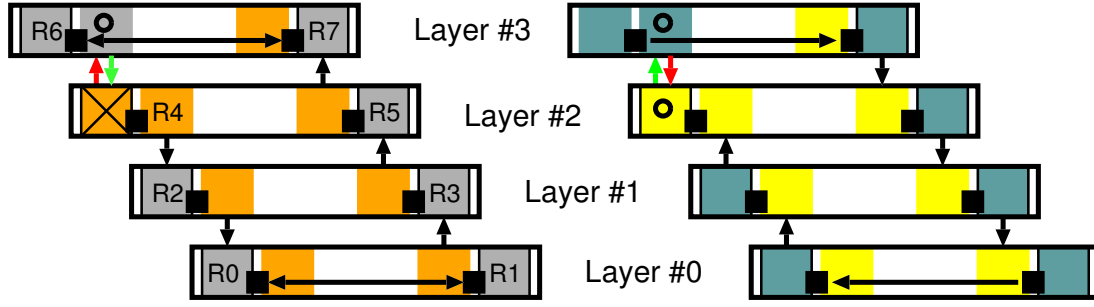


Figure 4.5: PSM on BRHL

- 2) The flit that comes from R4 cannot wake up the red vertical link due to a latency overhead, because in this case the blue vertical link has to be configured. Hence, the flit should go through the counterclockwise.

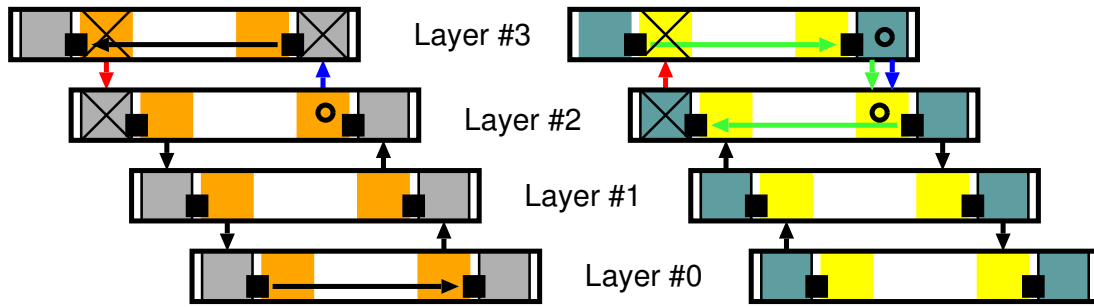


Figure 4.6: FSM on BRHL

4.3.3 Sleep Control method for BRFL

In the case of the BRFL, we analyze the PSM and FSM for two different utilization cases. The BRFL also has pre-configured transceivers. That is, after fabrication, the chip cannot configure the vertical link direction dynamically.

The PSM is used when vertical link utilization is low as well as the other two vertical link ring networks (i.e., URSL and BRHL), so we omit detail of this method here. The difference is that, compared to URSL, BRFL provides more resources which can be used for controlling the sleep and wakeup. This can ease the sleep and wakeup control especially when the vertical network utilization is high.

There are two vertical links with reciprocal directions connected to the same two routers. We call them brother vertical links. The FSM can use one data link to control sleep/wakeup on the brother vertical data link. However, the difference between the BRFL and URSL is that the BRFL has one more path to wake up the whole vertical link. Both brother link and neighboring link can be used to transfer wakeup signal. Other behaviors are the same as those of URSL.

4.3.4 Implementation of On/Off Link Control Logic

Three hardware modules are added for the proposed scheme: vertical link monitor, arbitration logic, and PSM/FSM manager. Figure 4.7 shows the diagram of the sleep control logic. Note that wakeup control logic is not required, because the on-demand method is used in this paper. As long as flit comes, the dormant DTx or the whole vertical link will be active without any judgment.

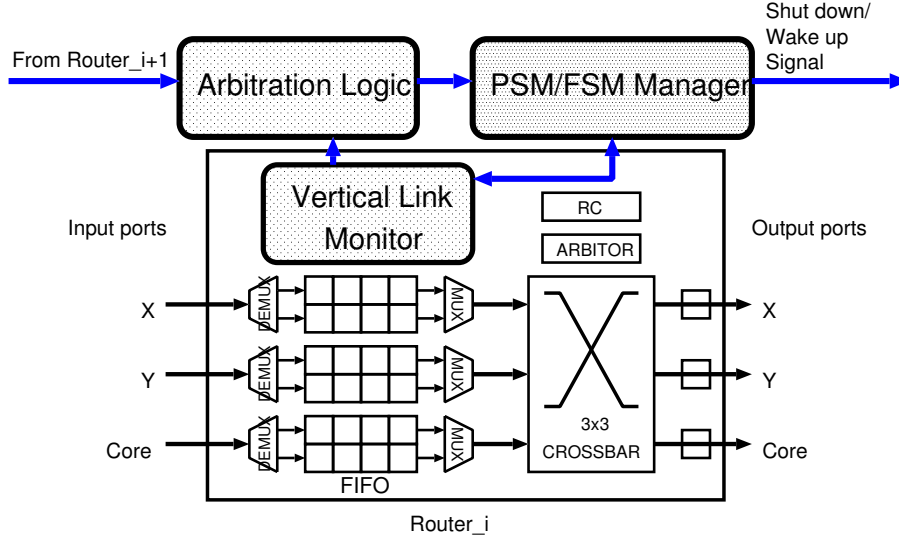


Figure 4.7: Sleep control logic

The vertical link monitor checks the state of the vertical link every cycle and computes the utilization of the vertical links between the neighboring chips depending on the size of sample window. Arbitration logic decides which vertical link (vertical links between the same neighboring chips are called neighbor links) can be shut down. If two request signals (from $router_i$ and $router_{i+1}$ which are on layer i as shown in Figure 4.7) are received, the arbitration logic decides which vertical link should be shut down depending on their utilizations. The vertical link which has lower utilization is prior to the one has higher utilization. If only one router sends the request, the arbitration logic gives decision signal to PSM/FSM manager directly. The arbitration logic is designed to avoid deadlock which leads neighbor vertical links sleep forever. The reason will be explained in Section 4.3.1. PSM/FSM manager decides which policy to use, and sends the sleep signal. PSM or FSM will be used on the target links.

Formula (1) shows the vertical link utilization $U_{verticallink}$.

$$U_{verticallink} = \frac{\sum_{t=i}^{t=i+w} f(t)}{w} \quad (4.4)$$

where

$$f(x) = \begin{cases} 0, & \text{if no coming flit in cycle } t \\ 1, & \text{if flit comes in cycle } t \end{cases} \quad (4.5)$$

Table 4.3: Processor parameters ($N = 4$)

Processor	UltraSPARC-III
L1 I/D cache size	64 KB (line:64B)
# of processors	4
L2 cache bank size	256 KB (assoc:4)
# of L2 cache banks	16
Memory size	4 GB
# of memory controllers	2

Table 4.4: NoC parameters ($N = 4$)

Topology	URSL, BRFL, BRHL ring
# of routers	8
Router pipeline	[RC/VSA][ST][LT]
Flit size	128 bit
# of message classes	3
Control packet size	1 flit
Data packet size	5 flit

where i is the start cycle of sampling, and w is the size of sample window in terms of system clock (SClk).

4.4 Evaluations

Firstly, experiment platform and method are introduced. Then, the BRHL and BRFL are compared with the URSL used in Cube-0 in term of application performance and power consumption in the situations with and without power optimization. Next, the power consumption and application performance of three ring network types are compared with those without power optimization. We also discuss the way how to decide the vertical link utilization threshold for each benchmark programs.

4.4.0.1 Experiment Setup

A full-system CMP simulator, Simics [91] with GEMS [92] is used to verify our proposed methods. It is performed to measure the real application performance. Tables 4.3 and 4.4 list the processor, cache, memory, and network parameters, where N is number of chips stacked. Proposed methods are evaluated in terms of application performance and the overall power consumption of 3D wireless NoC.

Full-system simulations of the wireless 3D CMPs that stack four chips which form a wireless ring network are performed to measure the real application performance. Each chip has one processor, four shared L2 cache banks, and two on-chip routers. Two memory controllers are connected to

the bottom chip. The cache architecture is SNUCA [76]. We modified a detailed network model of GEMS, called Garnet [93]. Garnet is an interconnection network model inside a full-system simulation framework GEMS. The power parameters based on power analysis of real router and vertical link design (Table 4.1) are fed to the simulator. The result (real power consumption and benchmark program execution time) of each evaluation can be automatically output after simulation. Then we can get the power consumption results of vertical links before and after power optimization. We use five parallel programs from the OpenMP implementation of NAS Parallel Benchmarks [94]. Sun Solaris 9 operating system is running on the simulator. These benchmark programs were compiled by Sun Studio 12 and are executed on Solaris 9.

The most important contribution of this paper is to improve the energy efficiency of the wireless 3D NoC over Cube-0 by means of introducing two sleep modes. So we quantitatively compare the proposed approaches with the conventional Cube-0 implementation, explained at the beginning of Section 4.1.

As shown in Table 4.6 and 4.7, for example, we can see the changes after using the proposed methods in the case of LU and IS, respectively. However, we can easily find the big gap between these two benchmark programs (e.g., the execution time running LU is as 7.5 times as IS's on average). Hence, in the following sections, valuation results of power consumption and execution time are shown as relative values (i.e., percentage) instead of absolute values in order to show the effect of the proposed methods clearly. Please note that, moreover, "before using the proposed methods" means using conventional method.

4.4.0.2 Baseline comparison among URSL, BRFL, and BRHL

Here, we compare the application execution time and network power consumption of the URSL, BRFL, and BRHL. The power consumption comparison among the URSL, BRFL, and BRHL is shown in Figures 4.8 and 4.9. The power consumption (Y-axis) is normalized so that the power consumption using the URSL (before using the proposed methods) indicates 100% in Figures 4.8 and 4.9. Meanwhile, execution time (Y-axis) of benchmark programs also using the URSL (before the proposed methods) is used as baseline in Figures 4.10 and 4.11.

Figures 12 and 4.9 respectively show the network power consumption before and after optimization, when five benchmark programs (IS, LU, DC, MG, and BT) running on the full-system simulator. Meanwhile, Figures 4.10 and 4.11 show the application execution time of the same benchmark programs before and after the optimization.

As shown in Figures 12 and 4.9, the BRFL consumes the largest power, and it exceeds the URSL by 11.7%-33% and the BRHL by 8.5%-28.5%, respectively. Regardless of the proposed on/off vertical link methods, the power consumption of the BRFL is the largest, while the URSL consumes the least power, because twice numbers of inductors used in the URSL are needed in the BRFL as mentioned in Section 4.2. The power consumption of the BRHL is slightly larger than

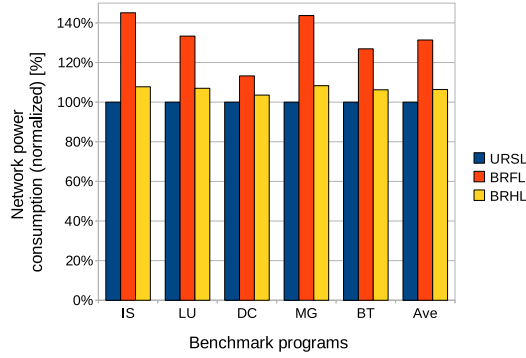


Figure 4.8: Network power consumption before using the proposed methods.

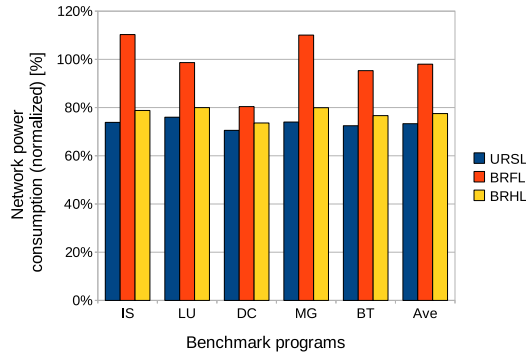


Figure 4.9: Network power consumption after using the proposed methods

that of URSL but much smaller than that of BRFL. The dynamic inductor configurations cause a certain amount of power consumption in the case the BRHL. That is, the total power consumption of BRHL is a little larger than that of URSL when a certain amount of inductors need to be configured dynamically. The application execution time of the URSL is the longest as shown in Figures 4.10 and 4.11 . It is much longer than the BRFL and BRHL. The BRFL and BRHL outperform the URSL by 10.6%-29.9% and 7%-25.8% respectively. To some extent, the BRHL has some advantages over the other two networks in the 3D NoC.

4.4.0.3 Power Saving

The main purpose of the proposed vertical on/off link control methods is to reduce power consumption of the wireless 3-D NoC. We also estimated the power saving when the same benchmark programs run on the simulator as shown in Figure 4.12. 23.4%-29.3% power saving is achieved. On average, 26.6%, 25.4%, and 26.8% power consumption can be saved in the URSL, BRFL, and BRHL, respectively.

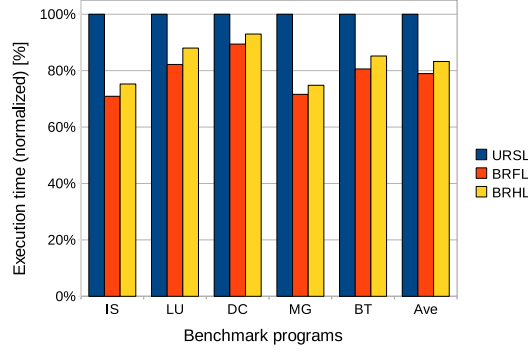


Figure 4.10: Application performance before using the proposed methods.

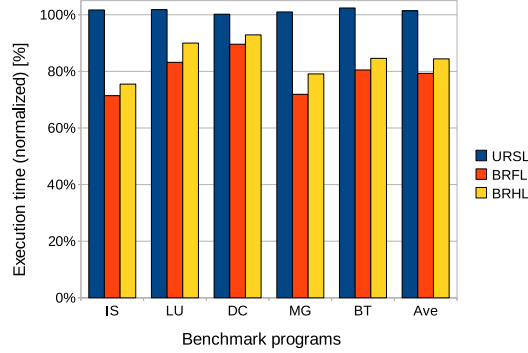


Figure 4.11: Application performance after using the proposed methods.

4.4.0.4 Performance Overhead

Commonly, power saving techniques cause a certain performance degradation. The following equation shows the latency overhead to wake up a sleeping vertical link.

$$T_{\text{latency}} = T_{\text{router}}N_{\text{router}} + T_{\text{link}}N_{\text{link}} + T_{\text{activate}}, \quad (4.6)$$

where, N_{router} , N_{link} represent the number of routers, links traversals on average. T_{router} , T_{link} correspond to the latency by transmitting wake up signal via one single router and one link. T_{activate} is the activation time for circuit elements after receiving wake up signal. As shown in Equation 4.6, the performance overhead consists of three parts. T_{router} depends on run-time traffic situation. Meanwhile, we assume T_{link} is one clock cycle. The first two parts depend on the number of routers and links by which wake up signals transfer, respectively. As mentioned in Section 4.1, we assume T_{activate} is one clock cycle. It follows that performance overhead is not only the time to activate sleeping circuits, but also the time to transfer wake up signals. Furthermore, wake up signal's transfer time is the main cause of the performance overhead. For the latency of shutting down, we assume one cycle clock to transfer shut down signal and one cycle clock to deactivate the vertical link.

Figure 4.13 shows how much performance is degraded by the proposed methods. Application

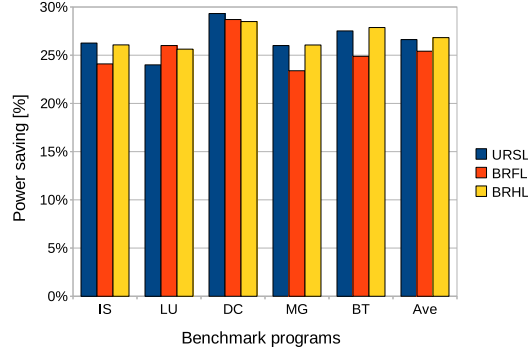


Figure 4.12: Power saving.

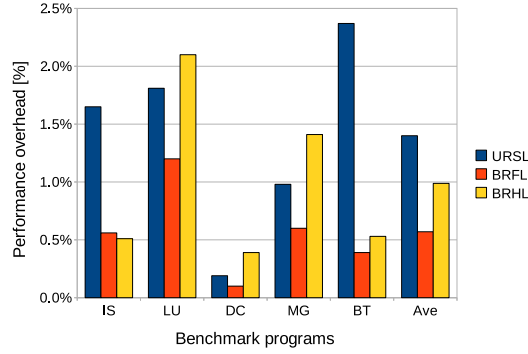


Figure 4.13: Performance overhead.

performance degrades by 0.1%-2.4%. On average, the application execution time can be 1.4%, 0.57%, and 1.0% longer than the time before using our proposed method in the URSL, BRFL, and BRHL, respectively. The results demonstrate that performance overhead of the proposed methods is quite small.

4.4.0.5 Synthesis Results

We synthesized the Cube-0 chip (not including wireless transceivers) before and after using the proposed vertical link on/off optimization with Synopsys Design Compiler. Fujitsu CS202SZ 65nm process technology is used for the design. The implementation costs of optimization hardware modules (Vertical link monitor, Arbitration logic and PSM/FSM manager) mainly embody in three aspects: area, frequency, and power consumption. Table 4.5 shows that the implementation area increases by 2.7% after the power optimization while the power consumption of one single chip, not including vertical links, increases by 3.1%. This power overhead is quite small compare with the energy reduction after using the proposed methods. Because the power consumption (8.1mW) of inductor as mentioned in Table 1, is much larger than the other parts, not to mention the power overhead (0.02mW shown in Table 4.5). In comparison to power saving gained from the proposed approach,

Table 4.5: Implementation Costs

Optimization	Area[mm ²]	Frequency[MHz]	Power[mW]
Before	2.6181	200	1.24
After	2.6250	200	1.28

Table 4.6: Simulation Result for LU

Optimization	Network	Execution time[clock]	Energy[J]
Before	BRFL	26731945	3.00E+8
	URSL	32523239	2.25E+8
	BRHL	28620450	2.41E+8
After	BRFL	27054033	2.22E+8
	URSL	33113266	1.71E+8
	BRHL	29258882	1.79E+8

this power overhead is negligible. Moreover, the operating frequency of Cube-0 is not affected by using our proposed low power regulations for vertical links. The whole overheads are quite small.

4.4.0.6 Effect of the Threshold on Power-Performance

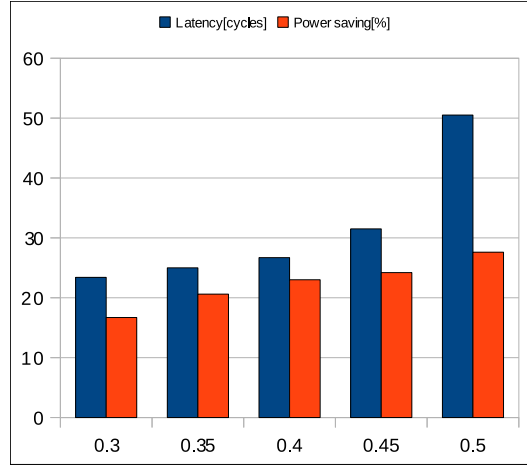
Power savings and latency-throughput are conflicting optimization goals. Higher power savings require shutting down more vertical links. However, fewer active vertical links increase the inter-chip traversal latency of packets, which was shown to degrade network throughput. Different applications have different power and performance demands. We should find an appropriate balance between these two goals.

In the history-based policy, a set of threshold parameters can be dynamically changed to adjust the on/off vertical link policy. Since latency/throughput performance is very important, we change the thresholds to explore the trade-off between power savings and latency/throughput performance under those network traffic conditions. In Figure 4.14, we show the values of the thresholds we set during each simulation. Under different threshold settings, latency and dynamic power savings are also different. With more aggressive threshold, we can obtain more power savings but at the cost of a higher latency.

In Figure 4.15, we show latency vs. power savings for LU application under URSL network. From the simulation results, we can see that, more power savings can be obtained but at the cost of a higher latency and throughput. That is, an improvement in one goal can only be obtained by degrading the other. Hence, we must select a good balance point for this benchmark program.

Table 4.7: Simulation Result for IS

Optimization	Network	Execution time[clock]	Energy[J]
Before	BRFL	3393218	4.31E+8
	URSL	4785367	2.97E+8
	BRHL	3601946	3.20E+8
After	BRFL	3412271	3.27E+8
	URSL	4864288	2.19E+8
	BRHL	3620275	2.34E+8

**Figure 4.14:** Latency and Power Saving under Different Thresholds

4.5 Summary

In this chapter, we have discussed about power optimization methods for inductive-coupling based wireless 3D network on chip. The power consumption of vertical link and existing problem are analyzed to show the necessity of this research.

We have presented a novel on/off link method to control the vertical links. One is called PSM, the other one is FSM. The proposed low-power techniques stop the transistors by cutting off the bias voltage in the transmitter of the vertical links only when their utilization are higher than the threshold. Meanwhile, the whole wireless vertical link will be shut down when the utilization is lower than the threshold in order to reduce the power consumption of wireless 3D NoCs.

Then this method is applied to ring based.

Finally, we evaluated the proposed methods. The results showed that the proposed methods reduced the network power consumption while the performance overhead is only Full-system many-core simulations using power parameters derived from a real chip implementation show that the proposed low-power techniques reduce the power consumption by 23.4%-29.3% in ring based network, while the average performance overhead is only 0.1%-2.4% in ring based network.

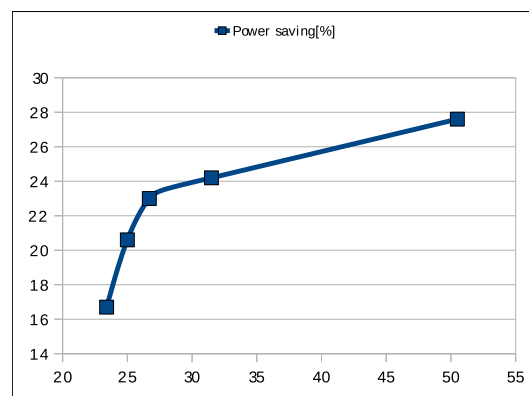


Figure 4.15: Latency vs. Power Saving

Chapter 5

Power Saving Methods for Irregular Wireless 3D NoC

In this chapter, we apply the vertical on/off link methods mentioned in Section 4.2 to the wireless 3D NoCs with irregular topology. Up*/Down* routing algorithm is used for transfer packages among nodes in wireless 3D NoCs. Meanwhile, an X-Y-Z like routing is proposed to transfer wake up signals.

5.1 Assumptions

In the general NoC, chips are vertically connected with the following rules: 1) all chips must have inductors at pre-determined locations. 2) The inductors are connected to input and output ports of on-chip routers to form vertical links, and 3) all the hardware components are topologically reachable to at least an on-chip router that has a vertical link. Any pre-determined network topology is not expected for each chip if these rules are satisfied. The total network in the system is treated as an irregular network, and Up*/Down* routing, a topology-agnostic routing algorithm without virtual channels is applied. Up*/Down* routing for irregular wireless 3D NoCs was introduced in Section 3.3.

It avoids deadlocks in irregular topologies without virtual channels, based on the assignment of direction (up or down) to network channels. Figure 3.6 has two examples of spanning trees, each of which has a different spanning tree root (node 0 or 7). A legal path must traverse zero or more channels upward followed by zero or more channels downward to guarantee freedom from deadlocks while maintaining network reachability. For example, the red solid line in Figure 3.6(a) indicates a routing path from nodes 2 to 5. This routing path uses two upward channels followed by two downward channels.

In order to select appropriate root node, we propose a root node selection algorithm in Section 3.3. Needless to say, the location of a spanning tree root significantly affects the hop count and utilization of links. For example, the routing path from nodes 2 to 5 in Figure 3.6(a) employs a non-

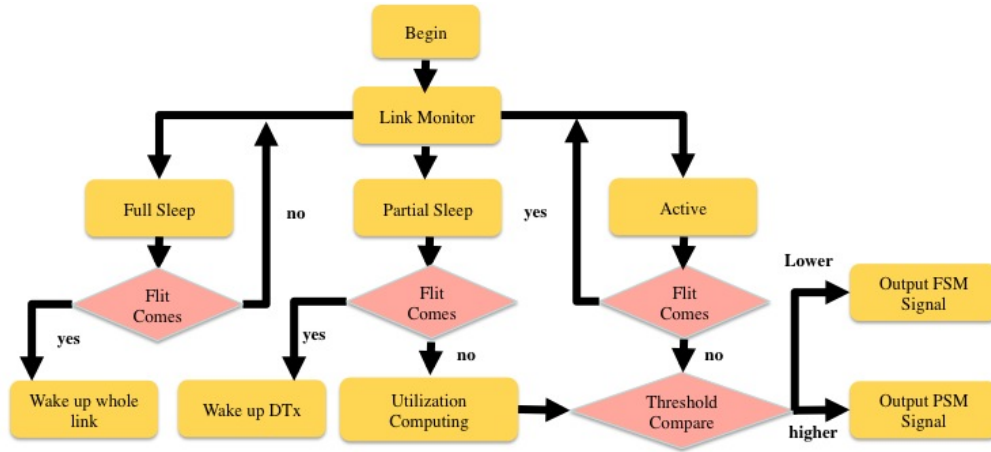


Figure 5.1: Dynamic Vertical Link On/Off Algorithm

minimal path, because the minimal path is prohibited by the spanning tree whose root is located at node 0; thus, four hops are required to send a packet. However, the minimal path in Figure 3.6(a) is allowed by the spanning tree whose root is located at node 7; thus, only two hops are required. We assume this routing algorithm is used for transferring data packets.

In addition, we assume that full-duplex vertical links are employed as communication model. Although spanning tree is a uni-directional kind of technique, here we assume that deadlock-free routings that use multi-spanning trees exist in our optimization. The deadlock-free routing uses spanning trees only for computing paths. Once path is computed, communication is done in full-duplex manner.

For the power consumption model, we use the same model as Chapter 4. The power consumption parameters for vertical link, on-chip router and communication controller are shown in Table 4.1.

5.2 Vertical On/Off Link Methods for Irregular Wireless 3D NoC

Two sleep modes (PSM and FSM) are also employed here. In PSM, only the data transmitter (DTx) is shut down. Since other parts are running, the vertical link can be woken-up when the sender activates DTx in one clock cycle and sends the packets. In FSM, the whole vertical link will be shut down.

We set a reasonable threshold utilization value first, then compare the link utilization within a sample window (time slot) with the threshold when flit transmission completes. If the utilization is higher than the threshold, the state of the target vertical link is set from active to PSM. Otherwise, if there are paths to wake up the whole vertical, we set the state to FSM. If not, PSM is used again to only shut down the DTx. The decision to turn on a vertical link is basically the converse of the sleep mechanism and can be done as on-demand because of the property of fast on/off link transition. This is basically same as that of ring based wireless 3D NoCs. Such on/off control flows are summarized in Figure 5.1.

The sleep decision is based on the utilization of target vertical link and whether there are enough

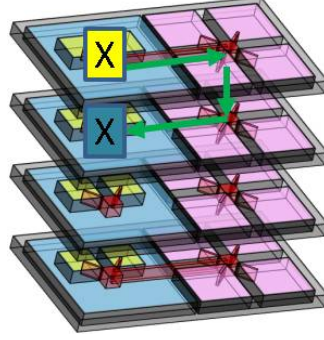


Figure 5.2: Wake up signals transfer in ring topology

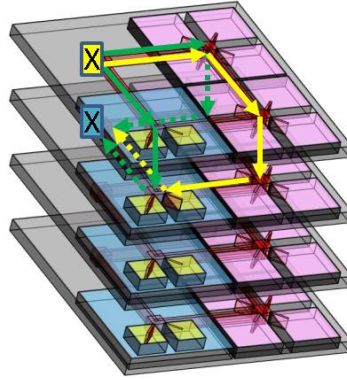


Figure 5.3: Wake up signals transfer in irregular topology

reachable paths to send wake up signals. Theoretically, every link can be used to wake up sleeping vertical links. For example, wake up signal can be sent through brother vertical link (routers are connected by a couple of links which are called brother links), neighboring vertical links, or remote vertical links.

As shown in Figure 5.3, several paths (green arrows and yellow arrows) are available to wake up the whole sleeping vertical link (yellow box with capital x is Tx, dark green box with capital x is Rx). We can see there is only one path in Figure 5.2. Meanwhile, there are more than one wake up paths in Figure 5.3. However, we have to think about the balance between power saving and performance overhead instead of selecting an arbitrary path.

In addition, a simple rule which can avoid the deadlock is introduced:

Rule 1: if there are one or more paths to send wake up signals, the whole vertical link can be shut down. Otherwise, we have to shut down the DTx only.

Obviously, our proposed on/off vertical links methods satisfy this rule. We prove as follows.

Proof:

1. In ring approach, the only two neighbor vertical links between two layers would use each other to wake up DRx, CTx, and LClk. Then, we cannot shut down these two vertical links simultaneously. One of them would check the other one's state before shutting down the

whole vertical link. If the other one's state is FSM, then shut down DTx only (PSM). That is, the vertical link in PSM state makes sure that there is one path to send wake up signals.

2. In irregular approach, we set at least one ever-on vertical link between two layers. This guarantees the connectivity of the inter-chip network. In other words, the ever-on vertical link avoids the network deadlock.

As shown in Figure 5.4, for example, the red vertical link is set to ever-on link. The wake up signals will be transferred through the red vertical link if nearer vertical link cannot be found. The vertical link which has the highest utilization will be selected as ever-on link by the system after several clock cycles sampling. The ever-on link will be always in power-on state or PSM state no matter how the environment changes. Thus, at least one path is provided to transfer wake up signals.

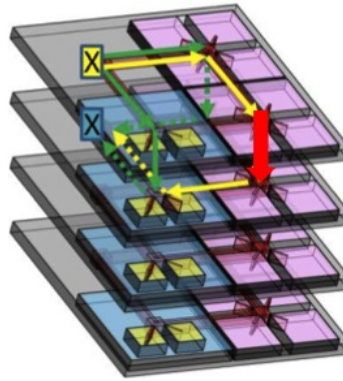


Figure 5.4: Red vertical link is set to ever-on link

Note that, data packets comply with Up/Down routing while wake up packets (i.e., signals) do not only comply with Up/Down routing. Since Up/Down routing forces packets not to transfer through some paths that may cause deadlocks, the system performance will be worse, if the wake up packets simply transfer by rule of Up/Down routing. Therefore, we consider two cases. (1) For neighbor vertical links, X-Z-X or Y-Z-Y routing is used. Figure 5.5 shows an example of path selection for waking up sleeping vertical link. Red lines and blue lines represent two spanning trees. If wake up packets transfer from node 14 to node 6, yellow path is prohibited by Up/Down routing, but allowed by X-Z-X routing. Thus, wake up packets have higher possibility to be transferred to destinations. (2) For remote vertical links, wake up packets are transferred according to Up/Down routing, and driven to transfer through ever-on vertical link. However, when nearer power-on or PSM state vertical link can be found before arriving ever-on vertical link, that power-on or PSM state vertical link will be selected to transfer wake up packets. Since the scale of target 3D NoC is small, we set only one ever-on vertical link between each two layers.

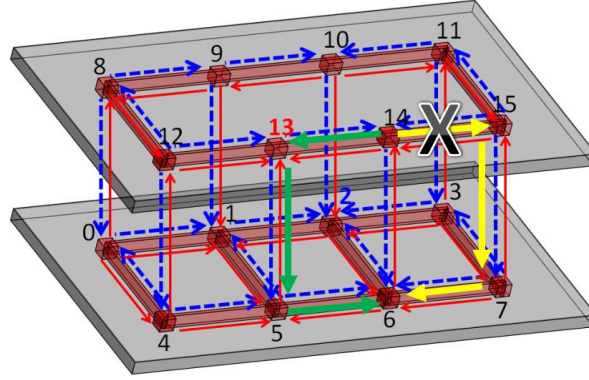


Figure 5.5: Wake up paths in irregular network with 16-tile configuration

Table 5.1: Irregular topologies to be tested (parameters).

	Topology	#routers	#CPUs	#L2\$ banks	#Memories
16-tile(ring)	(2,1,8)	16	4	32	4
16-tile(irr)	(2,2,4)	16	4	32	4
64-tile(irr)	(4,4,4)	64	8	128	16

5.3 Evaluations

Here, our main target is a generalized wireless network proposed in [95]. It is referred as “irregular approach” since it can cope with any network topologies. For comparison, we also show results of a single ring-based topology as “ring approach”.

First, the target CMP architecture that has 16 and 64 tiles and its evaluation environments are described. Second, the power consumption by irregular approach and ring approach is evaluated. Finally, the proposed on/off vertical link methods will be applied to both approaches. The impact of the proposed on/off vertical link in irregular approach is evaluated in terms of the performance overhead and saving of power consumption. It is also compared with ring approach for reference. Note that, because of the limitation of layer numbers in ring approach, we only compare the case of 16 tiles (i.e., 8 layers in ring approach while 4 layers in irregular approach).

5.3.1 Target NoC Architecture and Evaluation Environments

Here, we assumed shared-memory NoCs, in which each chip has one processor, each processor has one private L1 data and instruction caches, while the unified L2 cache banks are shared by all the processors. These processors (CPUs), L2 cache banks, and memory controllers (MCs) are interconnected via on-chip routers.

Table 5.1 shows the configurations for 16-tile and 64-tile wireless 3D NoCs. Topology (x, y, z) denotes a wireless 3D NoC that consists of z chips where each chip consists of $x \times y$ tiles. Figure 5.6 illustrates the irregular and ring topologies (we evaluate ring topology with 16-tile configuration) to

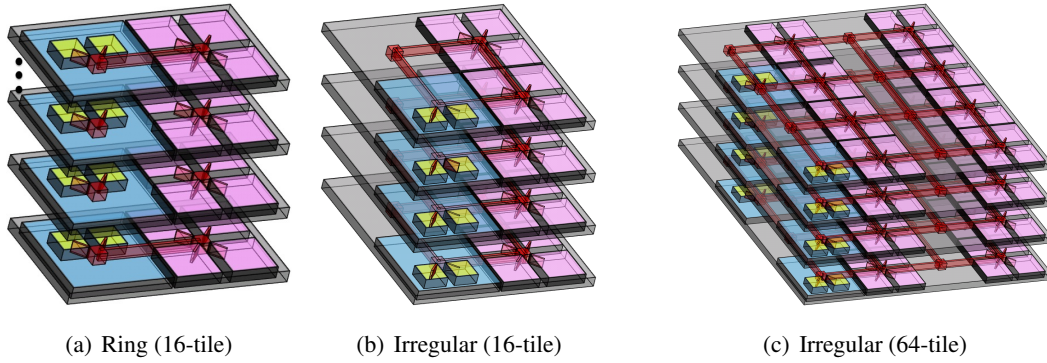


Figure 5.6: Irregular and ring topologies used in the evaluation part.

be evaluated in this paper. The number of chips z is fixed at four in the irregular approach, while their sizes correspond to (2,2) and (4,4) for 16-tile and 64-tile NoCs. In the ring topology, every chip has a horizontal link while only the top and bottom chip actually use it [34]. In the irregular topologies, horizontal links are randomly removed with 50% probability while vertical links are not removed.

We used a full-system CMP simulator that combines GEMS [92] and Wind River Simics [91] to simulate the wireless 3D CMPs. We modified a detailed network model inside GEMS, called Garnet [93]. The power parameters based on SPICE simulation are fed to the simulator. The results (network power consumption and application program execution time) of each evaluation can be automatically output after full-system simulation. The power consumption model will be shown in Subsection 4.2.2. That is, we can compare the results of network power consumption before and after power optimization for ring and irregular approaches, respectively. A directory-based MOESI coherence protocol that defines three message classes was used. Table 4.3 and 4.4 lists the processor and network parameters. We used parallel programs from the OpenMP implementation of NAS Parallel Benchmarks (NPB) [94]. Sun Solaris 9 operating system (OS) was running on the CMPs. These benchmark programs were compiled with Sun Studio 12 and were executed on Solaris 9 OS. The number of threads was set to four or eight, depending on the number of processors (e.g., four threads for 8-tile CMP).

5.3.2 Baseline Comparison between Irregular and Ring Approach

This section compares the irregular and ring approaches in terms of the application performance and power consumption.

One thousand irregular topologies were randomly generated for each CMP configuration to evaluate the irregular approach so that each horizontal link appeared with 50% probability while each vertical one appeared with 100%. Out of 1,000 trials for each network size, a network that had the closest hop count value to the average was selected for full-system CMP simulations. In addition, the irregular approach uses three VCs, so we call it Irr3 in this paper.

One or more spanning tree roots are selected with the proposed technique of optimizing spanning

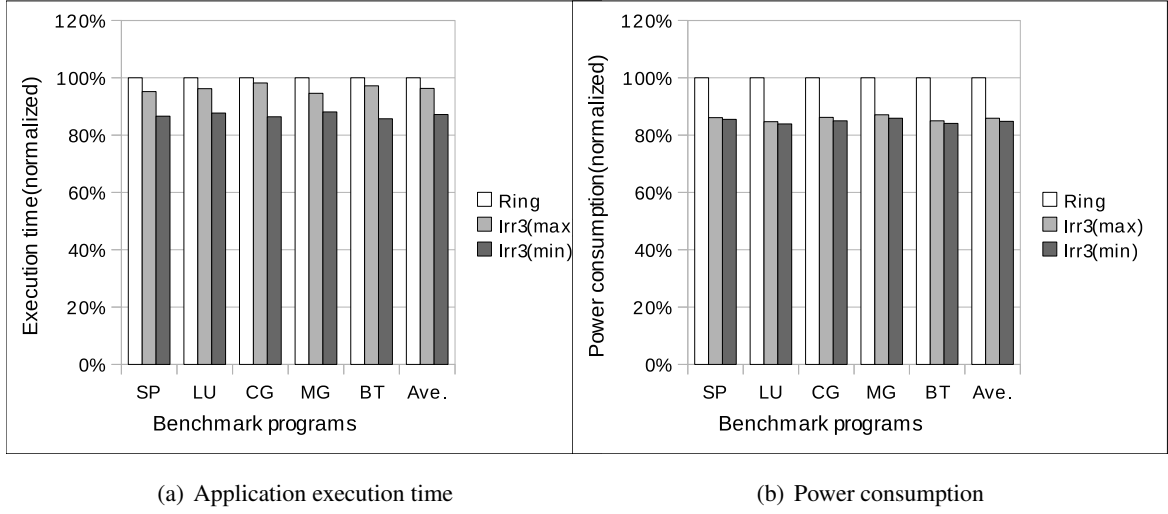


Figure 5.7: Comparisons between ring and irregular approaches

trees for each message class of applications using the cost function [95] to minimize the average hop count. Two configurations are used to compare with ring approach.

- **Irr3(max):** The worst spanning tree root that maximizes the average hop count is selected for each message class of each application.
- **Irr3(min):** The best spanning tree root that minimizes the average hop count is selected for each message class of each application.

In addition, [34] proposes to use Bubble flow control [96,97] for the vertical ring so as not to use any VCs. It uses a single 15-flit FIFO buffer for each physical channel in order to store three packets in each buffer. Thus, its buffer capacity is equivalent to three VCs each of which has a 5-flit FIFO buffer. In summary, the following configuration of ring approach is compared with Irr3(max) and Irr3(min).

- **Ring:** Ring approach with Bubble flow control (15-flit FIFO for each physical channel)

5.3.2.1 Application Performance

Figure 5.7(a) shows the application execution time of NPB applications with the irregular and ring approaches on the 16-tile and 64-tile configurations. The execution time (Y-axis) is normalized so that the execution time in ring approach indicates 100% in Figures 5.7(a). As expected, the irregular approach is better than the ring one that suffers due to poor scalability. The irregular configuration outperforms the ring one by up to 12.8% in terms of the application performance.

Up to here, we showed that the irregular approach outperforms the ring one in terms of the performance per cost.

5.3.2.2 Power Consumption

The most important contribution of this paper is improving the energy efficiency of the wireless 3D NoC by means of introducing two sleep modes. So we quantitatively compare ring and irregular approaches before and after using our proposed on/off vertical link methods. The following power model is used to calculate the power consumption of networks for these two approaches.

$$P_{NoC} = P_{router}N_{router} + P_{hlink}N_{hlink} + P_{vlink}N_{vlink}, \quad (5.1)$$

where N_{router} , N_{hlink} and N_{vlink} represent the number of routers, horizontal links, and vertical links traversals on average. P_{router} , P_{hlink} and P_{vlink} correspond to the power consumption of a router, a horizontal link, and a vertical link. The P_{router} and P_{vlink} are set according to Table 1, Section 4.3.5. For the P_{hlink} , horizontal links take up 65% power consumption of the whole 2D NoC [98] while routers take up 35%. Therefore, the P_{hlink} is set to 150 uW.

The power consumption comparison among the Ring, Irr3(max) and Irr3(min) is shown in Figure 5.7(b). The power consumption (Y-axis) is also normalized so that the power consumption of Ring approach indicates 100%. As shown in Figure 5.7(b), the Ring network with 16-tile configuration consumes the largest power, and it exceeds Irr3(max) by 12.9%-15.3% and Irr3(min) by 14.1%-16.0%, respectively. In the case 16-tile configuration, 28 full-duplex vertical links are employed in Ring network while 24 full-duplex vertical links are used in both Irregular networks. Thus, 14.3% more vertical links are employed in Ring network than in both irregular networks. Needless to say, more power is consumed by ring network. Meanwhile, the power consumption of the Irr3(max) is slightly larger than that of Irr3(min) but much smaller than that of Ring network due to the increased hop count.

5.3.3 Power Saving

In this section, we quantitatively compare the proposed approach with the conventional method (i.e., vertical link is going to be idle state when no flit comes). The conventional method is employed in Cube-0 [34] implementation. This method is applied to ring approach and irregular approach in 16-tile configuration.

For the flexibility of our proposed method, big power saving can be obtained. As shown in Figure 5.8(a), 23.4%-28.7% power saving in ring approach while 43.8%-50.0% in Irr3(max) and 46.7%-50.9% in Irr3(min) are achieved. On average, 25.4%, 46.2% and 48.5% power consumption can be saved in Ring, Irr3(max) and Irr3(min), respectively.

As mentioned in Section 5.2, enough available paths are needed for sending wake up signals. Obviously, only one vertical link can be shut down between two layers in ring approach. In irregular approach, by contrast, much more vertical links can be shut down because only one vertical link around the sleeping vertical link is needed to be awake. Since this vertical link is used to wake up the sleeping neighbor vertical link, other three vertical links around the sleeping vertical link can be shut

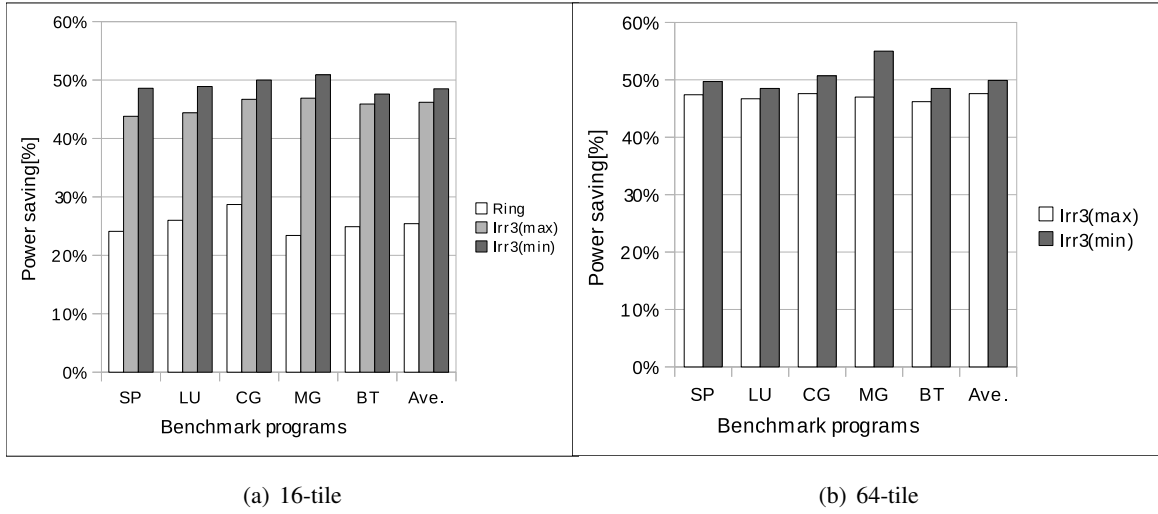


Figure 5.8: Power saving after using the proposed power optimization method.

down. Therefore, we can see in Figure 5.8(a), our proposed on/off link methods applied to irregular approach can obtain 20.8% more power saving than applied to ring approach on average. As seen in Figure 5.8(b), irregular approach with 64-tile can save more power consumption, but not so much.

In general, our proposed method applied to Irr3(min) can save more power than Irr3(max). According to our paper [95], 25.2% and 31.4% more hops are generated by Irr3(max) than Irr3(min) based on 16-tile and 64-tile configurations, respectively. Undoubtedly, the utilization of some vertical links will increase because of more hop counts. In terms of our proposed method, PSM is used when vertical link utilization is higher than a certain threshold while FSM is used in the case of lower utilization. And FSM can save much more power than PSM. From the above, Irr3(min) has more opportunity than Irr3(max) to use FSM, but not PSM. As shown in Figures 5.8(a) and 5.8(b), Irr3(min) can save 2.3% more power consumption than Irr3(max) on average.

5.3.4 Performance Overhead

Commonly, power saving techniques cause a certain performance degradation. Figures 5.9(a) and 5.9(b) shows how much performance is degraded by the proposed methods.

As shown in Figure 5.9(a) application performance degrades by 0.1%-2.4% in ring approach while 0.8%-2.2% in Irr3(max) and 0.4%-3.3% in Irr3(min). On average, the application execution time can be 0.6%, 1.3% and 1.5% longer than the time before using our proposed method in the Ring, Irr3(max) and Irr3(min), respectively. The results demonstrate that performance overhead of the proposed methods is quite small. Note that, the performance overheads are different among application programs. For example, in “CG” and “BT” which require a lot of data exchanges, the performance of Irr3(max) will become much worse than that of Irr3(min). By contrast, in other application programs, there are a small amount of data exchanges.

Again, it's clear that we could find many paths to wake up a sleeping vertical link according to

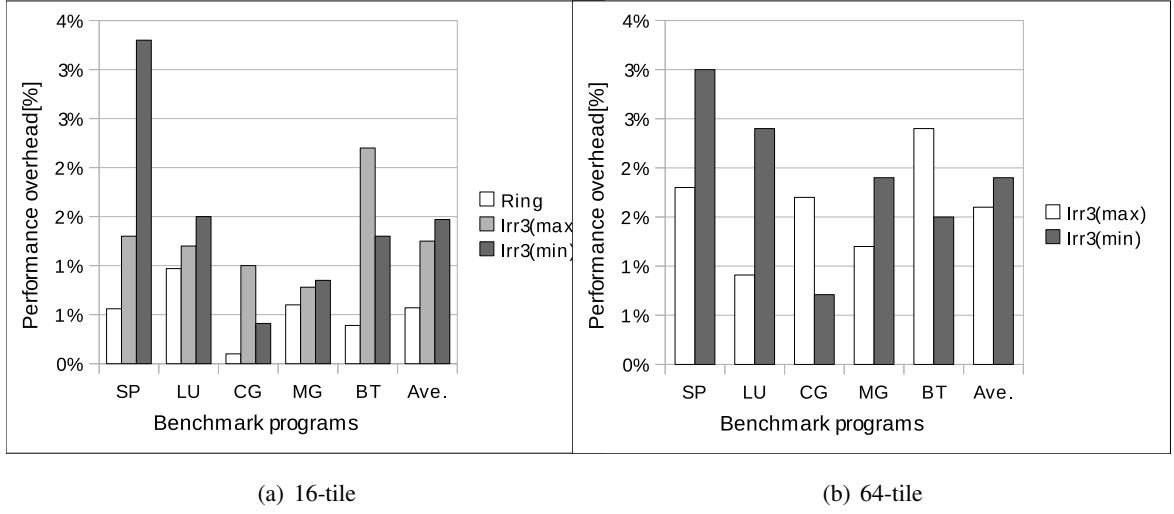


Figure 5.9: Performance overhead after using the proposed power optimization method.

the proposed method. There is only one path to wake up sleep vertical link in ring approach (i.e., neighbor vertical link). However, remote vertical links can be used to wake up the sleeping vertical link. The following equation shows the latency to wake up a sleeping vertical link.

$$T_{latency} = T_{router}N_{router} + T_{hlink}N_{hlink} + T_{vlink}, \quad (5.2)$$

where, N_{router} , N_{hlink} , and N_{vlink} represent the number of routers, horizontal links, and vertical links traversals on average. T_{router} , T_{hlink} , and T_{vlink} correspond to the latency by transmitting wake up signal via a router, a horizontal link, and a vertical link. Apparently, the latency to wake up a sleeping vertical link in ring approach is smaller than in irregular approach. Because the N_{hlink} is sometimes more than one in irregular case while equals to one in ring case. On average, in the case irr3(min) with 16-tile configuration, 0.9% more performance overhead is caused than in the ring case.

5.4 Summary

In this chapter, we have discussed about power optimization methods for inductive-coupling based wireless 3D network on chip. with irregular topology. The power consumption of vertical link and routing algorithm for transferring data packets are assumed. And then we proposed X-Y-Z like routing to transfer wake up signals.

We have presented a novel on/off link method to control the vertical links in the previous chapter. One is called PSM, the other one is FSM. The proposed low-power techniques stop the transistors by cutting off the bias voltage in the transmitter of the vertical links only when their utilization are higher than the threshold. Meanwhile, the whole wireless vertical link will be shut down when the utilization is lower than the threshold in order to reduce the power consumption of wireless 3D NoCs.

Finally, we evaluated the proposed methods. The results showed that the proposed methods

reduced the network power consumption while the performance overhead is only Full-system many-core simulations using power parameters derived from a real chip implementation show that the proposed low-power techniques reduce the power consumption by 43.8%-55.0% in irregular based network , while the average performance overhead is only 1.4% in wireless irregular 3D NoC.

Chapter 6

Latency Reduction Methods for Wireless 3D NoC

In Chapter 4, we have studied the power consumption optimization method using vertical on/off links to reduce the on chip network power consumption. We had successfully applied the method to ring based and irregular based wireless 3D network on chips, respectively.

In this Chapter, we will study the routing schemes for wireless 3D NoCs. A bubble flow routing algorithm will be simply introduced for ring based wireless 3D NoC; Up*/Down* routing algorithm will be explained. The Up*/Down* routing algorithm is used for irregular based wireless 3D NoC with multiple spanning tree structure. Moreover, a hybrid routing algorithm called Mixed Routing will be proposed for wireless 3D NoC to improve network performance. Finally, we will evaluate the last two routing algorithm according to performance and hops.

6.1 Vertical Bubble Flow

A prototype wireless 3D chip, called Cube-0, which is mentioned in Chap 3, was just implemented to test feasibility and explore flexibility of the inductive-coupling based wireless stacking technique. So it employed only a vertical ring network for the sake of simplicity.

Therefore, vertical bubble flow control [34] is proposed for ring topology with virtual cut-through (VCT) switching to avoid deadlock. Each chip has a pair of data transceivers for uplink and downlink to form a ring. The data transceiver for uplink receives data from a neighboring lower chip and transmits the data to the neighboring upper chip. The downlink is used for the opposite direction of the uplink.

The bubble flow control [96,97] is applied to the inter-plane vertical ring network of the building-block SiPs.

Figure 6.1 illustrates its behavior. In this figure, each input port of the routers has a buffer with a capacity of three packets. Each white box indicates an empty buffer with the capacity of a single packet while the gray box is an occupied one. The routers must follow the flow control rules listed below.

- **Flow control rule 1:** A packet on a ring can move to the next router along the ring when the

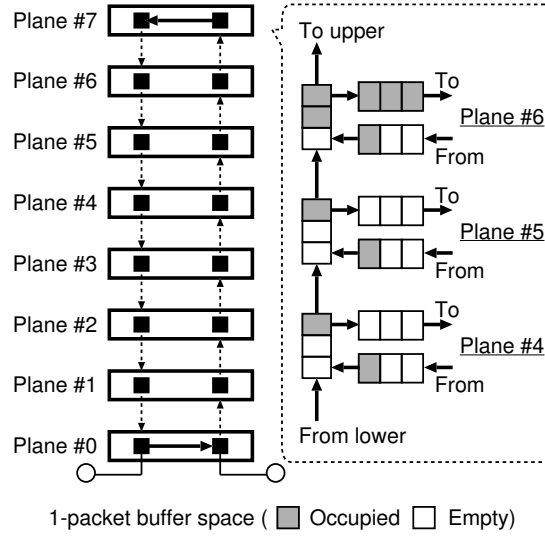


Figure 6.1: Vertical bubble flow network.

input buffer of the next router has an empty space of at least one packet.

- **Flow control rule 2:** An intra-plane network can inject a packet into a ring when the vertical input buffer of the ingress router has an empty space of at least two packets. For example, Planes 4 and 5 can inject a packet to the ring, while Plane 6 cannot.
- **Flow control rule 3:** A packet can exit from a ring only when the horizontal output buffer of the egress router has an empty space of at least one packet. Otherwise, the packet must go around the ring again (i.e., miss-routing). For example, packets destined to Planes 4 and 5 can exit from the ring, while packets destined to Plane 6 cannot exit and must go around the ring until an empty space appears in the horizontal output buffer of Plane 6.

Based on these rules, the packet injection is limited so as not to consume all the buffer resources in a ring. This guarantees that the packets can continuously move along the ring. As long as the packets are continuously moving on the ring, no deadlocks can occur even though multiple message classes coexist in the same virtual network. The reader is referred to [96] and [97] for details about the bubble flow control.

6.2 A Plug-and-Play Routing Algorithm

In section 3.3, we introduced a routing scheme for wireless 3D NoCs with irregular topology. Here, we describe it simply again.

The wireless 3D CMP consists of a collection of processor and memory chips from different vendors, and a chip vendor cannot know the network structures of its upper and lower chips. Thus, there is no other choice but to assume that a given wireless 3D NoC represents an irregular 3D

network topology. The network requirements for the wireless 3D CMP are close to those of System-area-Networks (SANs) that employ irregular network topologies. Up*/Down* routing is the most popular deadlock-free routing algorithm for irregular networks, and has been used in Autonet and Myrinet. To guarantee connectivity and deadlock-freedom for irregular networks, it exploits a spanning tree based directed graph in which up or down direction is assigned to each network channel. The proposed routing protocol with spanning trees optimization is built upon the Up*/Down* rule for ensuring the deadlock-freedom.

Up*/Down* routing avoids deadlocks in irregular topologies without virtual channels, based on an assignment of direction to network channels. As the basis of the assignment, a spanning tree whose nodes correspond to routers in the network is built. The “up” end of each channel is defined as follows: 1) the end whose node is closer to the root in the spanning tree; 2) the end whose node has the lower node ID number, if both ends are on nodes at the same tree level.

6.3 Mixed Routing Scheme

In this section, we will propose an improvement version of plug-and-play routing scheme, called mixed routing.

As shown in the last section, we cannot expect any pre-determined network topology for each intra-chip communication in wireless 3D NoCs. This makes it difficult to establish deadlock-free routing paths in such ad-hoc 3D CMPs. One solution is to use a simple ring as 3D networks and bubble flow control which is a topology independent deadlock free routing [34]. Although this approach is simple and suitable for small systems, performance is limited. Another method is introducing deadlock-free routing for irregular networks based on conservative acyclic graph using spanning trees [99]. We proposed a sophisticated method to select a root of spanning tree formed by 3D stacked irregular networks [95]. However, it ignores topological regularity of the vertical dimension of such chips when attempting path search.

From the practical viewpoint, most of networks in the chip would be two dimensional mesh, and regular mesh is also formed in the vertical direction. A regularity of topologies is usually a key factor to make a high-performance deadlock-free routing. The main contribution is that we design a novel deadlock-free routing protocol for such 3D chips makes use of regularity in irregular network on chips. To optimize routing paths to topological structure of each chip, we classify two-surface NoC of each chip into is 2D mesh and irregular structure. In mesh topologies, X-Y routing is used for well traffic distribution, whereas Up*/Down* routing is applied for reduced path hops in irregular topologies.

6.3.1 Assumptions

In order to stack chips for building a 3D NoC, the position of coils that makes a vertical link must has the same two-dimensional coordinate in all chips. We assume that coils are provided in each

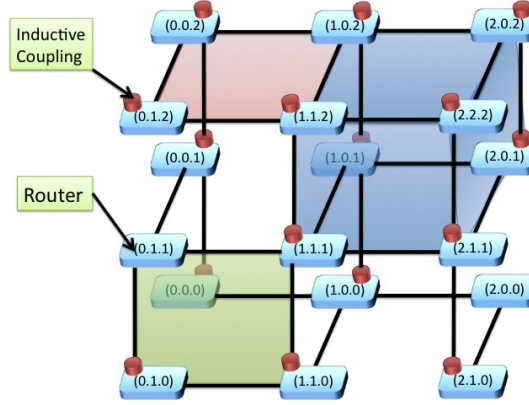


Figure 6.2: Layout of coils in 3D NoC

chip with a certain intervals as shown in Figure 6.2. A router is directly connected with each coil to transfer data from/to the intra-chip network. Three dimensional coordinates (x,y,z) are given to the router which is connected to the coil (x,y) of the z 'th chip. Here, we only care about the routing between such routers connected to the coil, although there may be other routers used only for intra-chip network. The routing between the numbered routers and routers for only intra-chip network will be done depending on the topology and the local rule of the intra-chip network.

From the viewpoint of real chip implementation, each chip is stacked with a certain small gap in order to avoid the electronic interference and save the space for bonding power supply wires [89]. However, logically, the 3D links can be treated just a linear network. Since the switching the receiver and transceiver can be done quickly in wireless interconnection network, here, a half duplex bi-directional communication is assumed on the linear network.

Note that, since each chip has different size of cores, it is difficult to provide an inductive coupling link (vertical link) at all (x,y) positions in a chip. Moreover, some vertical links have to be shut down for reducing the power consumption of Networks as mentioned in Chapter 4. Besides, some vertical links may encounter faults. Thus, these vertical links cannot work. All the above reasons make the vertical direction network an irregular one. However, of course, each chip must have a connectivity between upper and lower chips with some vertical links.

Since various type chips can be stacked, various intra-chip networks are connected. Here, we make the following assumption for each intra-chip network:

- all nodes on a chip are reachable only with its intra-chip network,
- an intra-chip network of each chip has a deadlock-free routing method.

If the intra-chip network of the stacked chip can work by itself, these assumptions are naturally satisfied.

According to the detected topology, the following standard routing algorithms are applied in the intra-chip network. Note that both algorithms can be deadlock free without virtual channels.

In the 2D mesh, the Dimension-Order Routing [8] (DOR) which is simple and common routing is used. Since the mesh does not have wrap around links, no virtual channel is needed.

On the other hand, Up*/Down* routing [14], which can prevent a deadlock, without using a virtual channel is applied in the irregular topology. To guarantee connectivity and deadlock-freedom for irregular networks, it exploits a spanning tree based directed graph in which up or down direction is assigned to each network channel. In this method, a certain node is selected as a root node, and the “up” end of each channel is defined as follows: 1) the end whose node is closer to the root in the spanning tree; 2) the end whose node has the lower node number, if both ends are on nodes at the same tree level.

Figure 3.6(a) and 3.6(b) show two examples of spanning trees, each of which has a different spanning tree root (node 0 or 7). A legal path must traverse zero or more channels in the up direction followed by zero or more channels in the down direction, in order to guarantee deadlock-freedom while keeping the network reachability. For example, a red line in Figure 3.6(a) shows a routing path from node 2 to node 5.

Needless to say, the location of a root node affects the hop count and link utilization significantly. In Figure 3.6(a), for example, the routing path employs a non-minimal path, because the minimal path is prohibited by the spanning tree whose root is located at node 0; thus 4 hops are required. In Figure 3.6(b), on the other hand, the minimal path is allowed by the spanning tree whose root is located at node 7; thus only two hops are required.

Matsutani et al. [95] proposed a sophisticated selection algorithms of the root of spanning trees used for Up*Down* routing in such 3D stacked chips. Here, the spanning tree root is selected so as to minimize the average hop count for given traffic pattern based on an exhaustive search, since the network size is not very large.

6.3.2 Recognition of the Mesh

If vertical links are fault-free and the all nodes in the system know of the topology information of all chips, the routing information can be stored in each node beforehand. However, with an NoC by using wireless inductive coupling links, various chips with their own intra-networks can be stacked, and it is difficult to fix the routing information beforehand. Also, considering faults, automatic topology recognition algorithm is advantageous.

Here, it is necessary to detect which is the topology of each chip, mesh or irregular. When the system boots, each node checks the neighboring nodes and layer information to detect the topology and the three-dimensional node number. Here, the topology is detected according to the following simple iterative algorithm.

— Iterative Algorithm of Mesh Recognition —

1. Find a minimum mesh consisting of four nodes connected with a ring structure.
2. A minimal mesh is treated as a 2nd-level node whose links are edges which are shared by other minimum meshes.
3. When such four nodes form a ring structure, it is treated as a 3rd-level node, and it is repeated until no ring structures is found or every node in the plane is included.

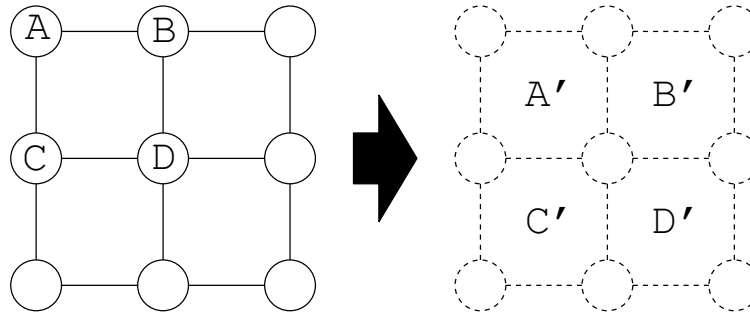


Figure 6.3: Example of Topology detection flow

It's able to get subnets with mesh topology from the irregular networks as much as possible according to the above iterative algorithm.

The left side of Figure 6.3 shows an example of 3×3 topology. First, it is confirmed whether node A connects with the node of the east side. If it is connected, the node on the east will be named as node B. Node B, node C, and node D are checked in the same way. If it checks being connected, the node A, B, C, and D form the minimum mesh A'. It is carried out repeatedly and forms small meshes A', B', C', and D' as shown in the right side of Figure 6.3.

Next, the meshes A', B', C', and D' are treated as a 2nd-level nodes and examined whether they are connected in the same way. In this case, the 3rd-level node becomes a 3×3 square mesh.

This process is applied for XY plane, YZ plane, and ZX plane. If the mesh structure in all planes can be connected, they will be treated as a 3D mesh.

In Figure 6.2, for example, light blue cube is the 3D mesh, light red square is the mesh on the XY plane, and light green square is the mesh on the ZX plane.

6.3.3 Mixed Routing

The general way for routing in a network consisting of various types of sub-networks is to treat the whole network as an irregular topology, and apply Up*/Down* routing.

However, UP*/Down* routing is commonly less effective than topology dependent routing like DOR routing in the mesh. By using the mesh recognition algorithm shown before, we can identify whether the source node or destination node belong to the same mesh, independent meshes or not.

Here, the following simple Mixed routing is proposed to make the use of regularity as possible:

Definition 1: Mixed routing:

Assume that the source router $S (s_x, s_y, s_z)$ sends a packet to destination router $D (d_x, d_y, d_z)$.

1. If source node and destination node are in the same chip ($s_z = d_z$), use the intra-chip routing depending on the topology of the chip.
2. If source node and destination node are in the same mesh topology, use the DOR in the mesh.
3. If source node is not in any mesh topology, use the Up*/Down* routing. Then, DOR is used in terms of a rule. This rule is called “topology switching algorithm (TSA)”.
4. If source node is in a mesh topology, use DOR in this mesh and then Up*/Down* routing is used in the rest of parts also in terms of TSA.

TSA requires to select a optimal boundary node on the boundary of the topology to which the source node belongs.

Definition 2: Topology switching routing (TSA):

Measure the distance among boundary nodes of the mesh to which the source node belongs and destination node when Up*/Down* routing is applied. The node with smallest distance is selected as a optimal boundary node. The DOR routing is applied from the source node to the optimal boundary node with virtual channel 0, then Up*/Down* routing is applied from the optimal boundary node to the destination node with virtual channel 1.

Theory 1: Mixed routing is deadlock free:

Proof:

1. In each network, Up*/Down* routing or the DOR is deadlock-free without virtual channel.
2. When the routing algorithm is switched, the virtual channel number is changed. Thus, topology switching routing is deadlock free.

Here, the routing algorithm is switched only once, since we assume two virtual channels. Thus, Up*/Down* routing is applied from the relay-node to the destination nodes even if the destination node belongs to the another different meshes. Adding more virtual channels will increase the flexibility if the system becomes large.

Figure 6.4 shows an example of 3D CMPs of the mixture topology. This CMPs consists of four layers and each chip has four nodes. Layer 0 is the bottom chip and layer 3 is the top chip of this 3D CMPs. layer 0 has node 0,1,2 and 3. Layer 1 has node 4,5,6 and 7. Layer 2 has node 8,9,10 and 11. Layer 3 has node 12,13,14 and 15. Layer 2 (disconnected node 8 and 9) are irregular topology. Using this figure, some examples of the routing are shown.

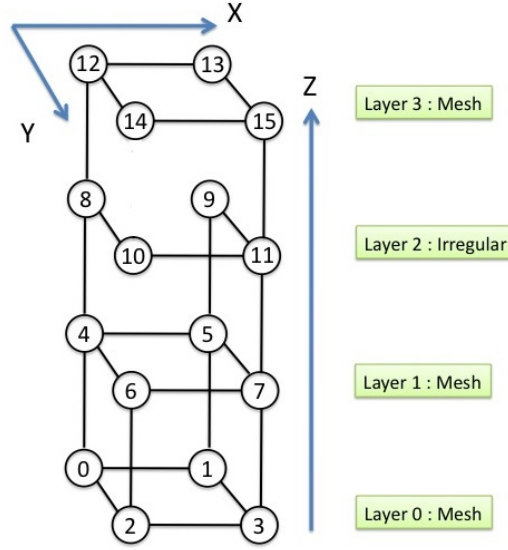


Figure 6.4: CMPs consist of four layers.

- Routing of node 0 to node 3: node 0 and node 3 are belongs to same layer that has mesh topology. So, the DOR is carried out.
- Routing of node 2 to node 7: node 2 belongs to layer 0, and node 7 belongs to layer 1, but dimension order routing is carried out because layer 0 and 1 compose 3D mesh structure.
- Routing of node 1 to node 10: node 1 belongs to 3D mesh structure (layer 0 and 1) and mesh of YZ plane (node 1, 3, 5, 7, 9 and 11). Node 10 belongs to irregular topology. Thus, Mixed routing is carried out. And vice versa.

6.3.4 Evaluations

In this section, the proposed Mixed routing is compared with a baseline irregular routing, in which the whole 3D network is treated as an irregular topology and Up*/Down* routing is used to route packets. The spanning tree roots for Up*/Down* routing are optimized so as to minimize the average hop count of each application based on an exhaustive search [95].

First, the evaluation environment and the target NoC architecture are described. Second, Mixed routing and irregular routing are compared in terms of the average hop counts using a network simulator. finally, full-system simulation results will be shown.

6.3.4.1 Network Simulations

To evaluate hop counts of Mix routing and irregular routing, we developed a network simulator called Irre_sim that has been employed by other papers, such as [100], etc. Irre_sim is a cycle-accurate

Table 6.1: Target architecture for network simulator

	T444	T448
Topology	(4,4,4)	(4,4,8)
Number of Nodes	64	128
Number of Z links	48 42 36	112 98 84

simulator of interconnection networks written in C++. Although the traffic is artificially generated, the operation of routers are based on the real hardware design. Here, a standard three-stage wormhole router and 5-flit (1-flit header + 4-flit body) packets are used.

We evaluated network configurations in Table 6.1 with the network simulator. Topology (x, y, z) denotes a wireless 3D NoC that consists of z chips where each chip consists of $x \times y$ nodes. For example, T444 (4,4,4) stacks four chips, each has 4×4 nodes.

Here, we use a number of chips with irregular topology as a parameter. Irregular topologies are automatically generated from 2D mesh network by randomly removing horizontal links with 50% possibility. We also remove some vertical links due to the irregularity of vertical direction networks as mentioned in Subsection 3.1. In addition, the number of vertical links (Z links) has been changed randomly as shown in Table 6.1. In order to keep connectivity of the network, removal which makes isolated nodes is prohibited.

The most commonly used Uniform traffic in which source and destination node are randomly selected is adopted. Neighbor traffic in which the destination node is selected adjacent node of the source node is also tried, since the Mixed routing is not effective in such a traffic with strong communication locality.

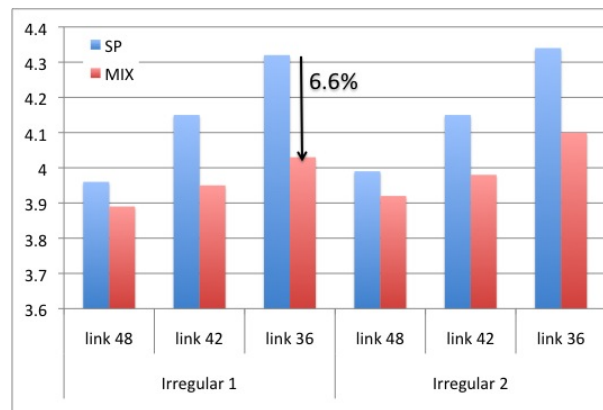
**Figure 6.5:** Average hop count of Mixed routing and irregular routing at uniform traffic (T444)

Figure 6.5 and Figure 6.6 show average hop count of Mixed routing and irregular routing with uniform traffic.

Each topology in the graph is based on the configuration shown in Table 6.1 with the number of

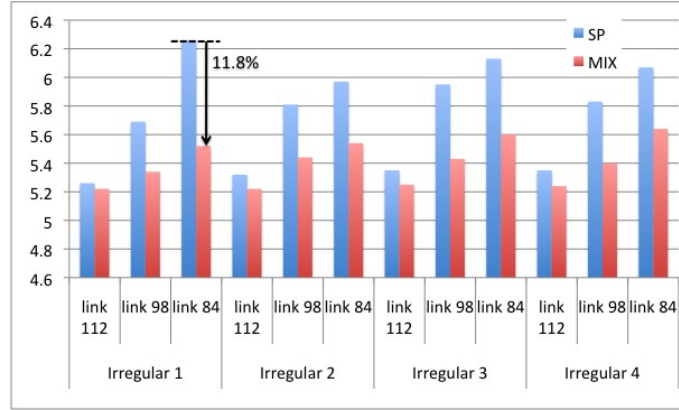


Figure 6.6: Average hop count of Mixed routing and irregular routing at uniform traffic (T448)

irregular chips as a parameter. Random irregular topologies are generated for Irregular 1-4 configurations. The meaning of link 48 is to have 48 vertical links. I also checked the situation of vertical link remove. For example, link 42 means 12.5% vertical links are removed randomly, link 36 means 25% vertical link are removed in same way.

In Figure 6.5, the hop count of Mixed routing is improved by 6.6% compared to that of irregular routing when irregular 1 with link 36 in uniform traffic. Similarly, in Figure 6.6, it is improved by 11.8% when irregular 1 with link 36. Note that even if all chips are irregular, the mesh can be formed in the vertical dimension, the performance of the Mixed routing thus overcomes the Up*Down* routing.

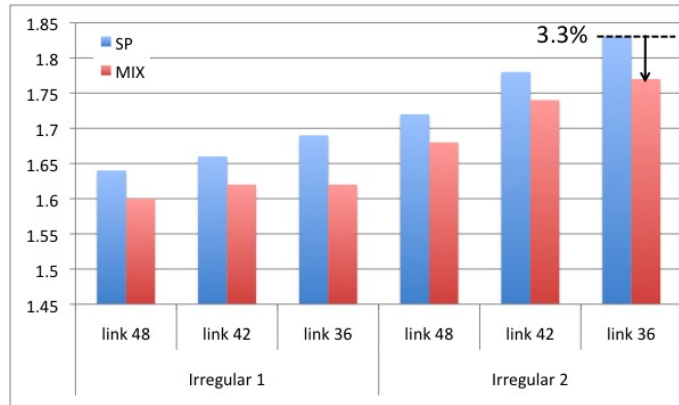


Figure 6.7: Average hop count of Mixed routing and irregular routing at neighbor traffic (T444)

Figure 6.7 and Figure 6.8 show the average hop count at neighbor traffic. In Figure 6.7, the hop count of Mixed routing is improved by 3.3% compared to that of irregular routing when irregular 2 with link 36 in uniform traffic. Similarly, in Figure 6.8, it is improved by 6.1% when irregular 3 with link 98. The results show that Mixed routing is effective even with the traffic with the strong com-

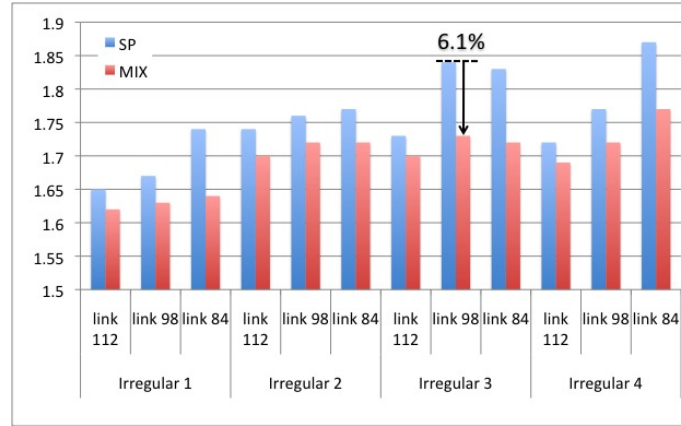


Figure 6.8: Average hop count of Mixed routing and irregular routing at neighbor traffic (T448)

munication locality. Since the average distance from source node to destination node with neighbor traffic pattern is much shorter than the case of uniform traffic pattern, the hop count improvement is not obvious that much.

From the above four figures, it is obvious that the hop counts increase as vertical link remove number increase in both Up*/Down* routing and Mixed routing. However, the hop count increment ratio with Mixed routing is smaller than that of Up*/Down* routing. Thus, when link is 36 and 84, the most benefit can be gained in these four figures.

We also compared the average hop counts of Mixed routing to the minimum hop counts computed with Dijkstra's algorithm. The difference between hop counts in Mixed routing and minimum hop counts is only 2.5% on average. Considering that the minimum routes computed with Dijkstra's algorithm may cause the deadlock, Mixed routing achieved near-optimal results with deadlock free paths.

Increasing the number of chips with irregular network stretches, the hop counts are increased as the number of chips with irregular network increases. However, when Mixed routing is used, the increasing ratio in uniform and neighbor traffic is much suppressed by using a small number of 2D mesh is used efficiently. The performance improvement becomes more obvious as the scale of networks become bigger. Therefore, we believe that the proposed routing algorithm is worth being applied to small and large scale of 3D NoCs.

6.3.4.2 Full-System Simulations

To evaluate the impact on the performance of wireless 3D NoCs, we also used a full-system CMP simulator: GEMS and Wind River Simics. Table 4.3 and 4.4 list the processor and network parameters. Eight parallel programs from the OpenMP implementation of NAS Parallel Benchmarks (NPB) are used here for running on the wireless 3D NoCs. Sun Solaris 9 operating system is assumed to run on the 4-chip NoCs. These benchmark programs were compiled by Sun Studio 12, and executed on

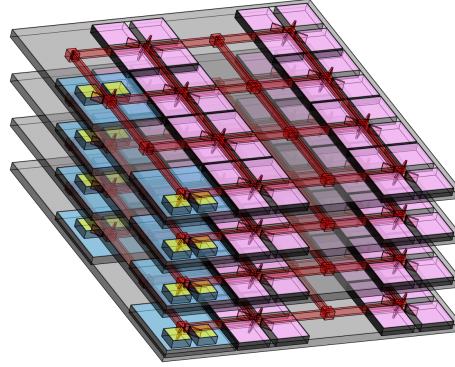


Figure 6.9: Target 3D NoC T444 for CMP simulator.

Solaris 9. The number of threads was set to four or eight, depending on the number of processors.

Figure 6.9 shows target 3D NoC for a full-system CMP simulator. Here, the blue tile shows CPU, the yellow one shows the L1 cache, and the red one is the L2 cache. Similar to the case of network simulation, the number of chips with irregular topologies is used as a parameter.

Here, we only check T444 case. Figure 6.10 and 6.11 show application execution time of Mixed routing normalized to that of Up*/Down* routing. Irr 0 is the case without irregular topology (i.e. mesh topology). Note that even in this case, meshes can be formed in the vertical direction. Irr 1-3 show cases with different random irregular topologies. The performance of Mixed routing is a little better than that of irregular routing except a few applications. Especially, the performance of Mixed routing in CG is improved up to 6.1% when irregular 2 with link 36 is used. Since the communications between chips are infrequent, the application execution time is not obviously improved by using Mixed routing.

In "CG", "MG" and "UA" which require a lot of data exchange, the performance improvement is large. On the other hand, in "DC" and "EP" with a small amount of data exchange, the performance is almost the same independent from the number of irregular network.

Although we didn't check the T448 case, it's reasonable that more performance improvement can be earned when the network scale becomes larger, such as T448 case, according to the network simulation results.

6.3.4.3 Power Saving Improvement

The most important contribution of this paper is improving the energy efficiency of the wireless 3D NoC by means of introducing two sleep modes. So here we compare the Up*/Down* routing and Mixed routing using our proposed on/off vertical link methods. The following power model is used to calculate the power consumption of networks for these two approaches.

$$P_{NoC} = P_{router}N_{router} + P_{hlink}N_{hlink} + P_{vlink}N_{vlink}, \quad (6.1)$$

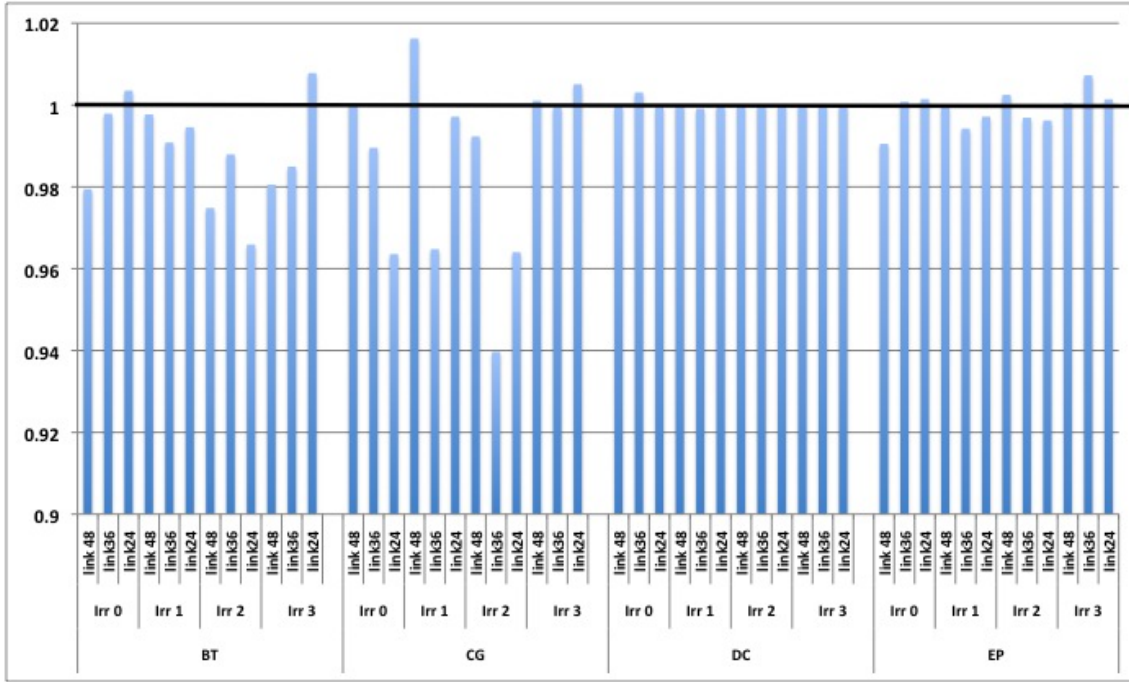


Figure 6.10: Comparison of execution time of mixed routing and irregular routing (1/2)

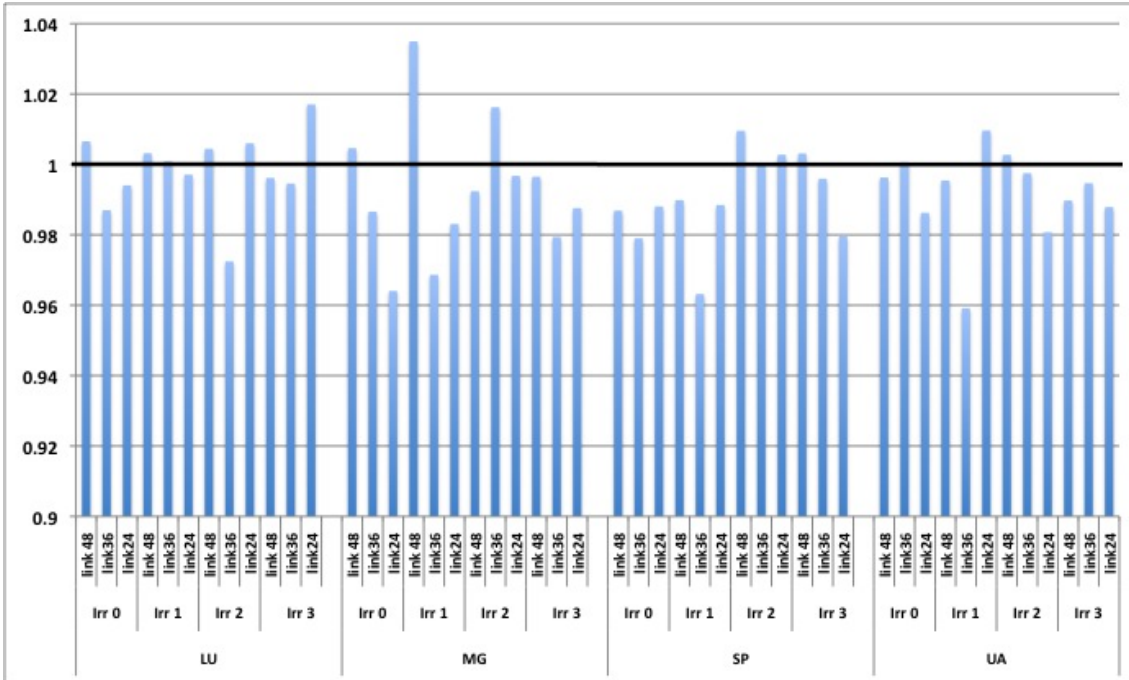


Figure 6.11: Comparison of execution time of mixed routing and irregular routing (2/2)

where N_{router} , N_{hlink} and N_{vlink} represent the number of routers, horizontal links, and vertical links traversals on average. P_{router} , P_{hlink} and P_{vlink} correspond to the power consumption of a router, a

horizontal link, and a vertical link. The P_{router} and P_{vlink} are set according to Table 1, Section 3. For the P_{hlink} , horizontal links take up 65% power consumption of the whole 2D NoC [98] while routers take up 35%. Therefore, the P_{hlink} is set to 150 uW.

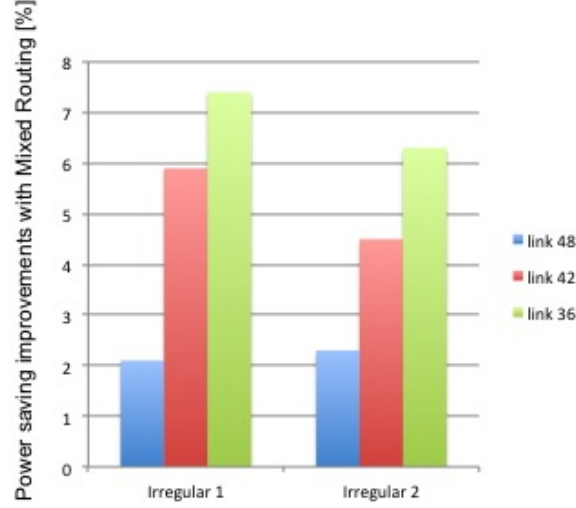


Figure 6.12: Power saving improvements with mixed routing compare to up*/down* routing (T444)

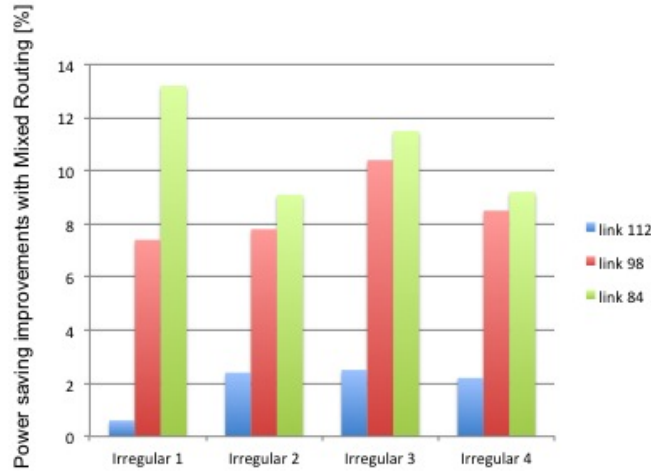


Figure 6.13: Power saving improvements with mixed routing compare to up*/down* routing (T448)

The most commonly used Uniform traffic in which source and destination node are randomly selected is adopted. Each topology in Figure 6.12 and Figure 6.13 is based on the configuration shown in Table 6.1 with the number of irregular chips as a parameter. Random irregular topologies are generated for Irregular 1-4 configurations. The meaning of link 48 is to have 48 vertical links. Figure 6.12 and Figure 6.13 show power saving improvements with Mixed routing compare to Up*/Down* routing. The vertical axis represents power saving improvements in style of percentage.

The horizontal axis represents the network configurations.

In Figure 6.12, the power saving of Mixed routing is improved by 7.4% compared to that of irregular routing when irregular 1 with link 36 in uniform traffic. Similarly, in Figure 6.13, it is improved by 13.2% when irregular 1 with link 36. Since the hop counts can be improved by using Mixed routing, the the switch time of vertical link can be reduced correspondingly. For example, when the number of vertical link is 36, the hop count improvement is less than that of 42 and 48 vertical links by using Mixed routing. The communication frequency between chips become lower. Thus, more vertical links can be shut down using FSM which is introduced in Chapter 4.

6.4 Summary

In this chapter, we introduced three routing schemes for inductive-coupling based wireless 3D NoC.

These chips can be formed into a single uni-directional ring network so as to fully exploit the flexibility of the wireless connections that enables us to add, remove, and swap chips in a package without updating any routing information. In this case, Vertical bubble flow was the best choice for ring based wireless 3D NoC.

Then, a plug-and-play routing scheme able to cope with the irregularity of the wireless 3D CMPs was introduced. The routing protocol exchanges the network information between all chips in order to establish the deadlock-free routing paths based on spanning trees.

Finally, an improvement version of plug-and-play routing scheme was proposed, called Mixed Routing. It classifies the target topology into 2D mesh, 3D mesh and irregular topology, and selects the routing algorithm based on its topology. Although the effect is depending on the number of 2D mesh in the 3D network, network simulation results revealed that Mixed Routing improved performance of irregular routing up to 11.8%. By using the full-system simulator, the execution time of parallel application programs is compared, and except a few cases, the performance with Mixed Routing overcomes that of irregular routing.

Since the hop counts can be improved by using Mixed routing, the the switch time of vertical link can be reduced correspondingly. The power saving is also improved by using Mixed routing instead of Up*/Down* routing. The results show that up to 13.2% power saving can be improved.

Chapter 7

Conclusions

7.1 Summary

Inductive-coupling is yet another 3D integration technique that can be used to stack more than three known-good-dies in a SiP without wire connections. It provides a large degree of flexibility in building target 3D ICs, such as enabling chips in the package to be added, removed, and swapped after the chips have been fabricated, similar to what is done with building blocks. However, problems also arise with these advantages, especially the issue of power consumption and routing algorithms. We proposed solutions to figure out the above problems.

In this thesis, we first introduced network on chip according to router architecture, network topologies, routing algorithm and low power technologies for on chip networks. After that, a commercial NoC-based chip is introduced. Then we move our discussion from 2D network on chip to 3D network on chip. 3D IC interconnection techniques are discussed before 3D network on chip. We believe the inductive coupling technology has big advantages compare to other 3D interconnection techniques. So we chose inductive coupling technique as a keynote to discuss. We discussed inductive coupling based wireless 3D NoC and a prototype wireless 3D NoC chip (cube-0) using inductive coupling.

In Chapter 4 and 5, we proposed on/off vertical link method in order to reduce the power consumption of inductive coupling links (vertical links). A big shortcoming of inductive coupling technique is high power consumption because of its high frequency. The proposed method is very simple but very effective to fix the above problem. We applied this method to cube-0 with ring topology. Afterwards, this method was applied to general wireless 3D NoC with irregular topology. The wake up routing algorithm must be modified at the same time.

We firstly proposed a Up*/Down* based plug-and-play protocol for general wireless 3D NoC in Chapter 3. The plug-and-play routing protocol is able to cope with the irregularity of the wireless 3D NoCs. It exchanges the network information between all chips in order to establish the deadlock-free routing paths based on spanning trees. In addition, a spanning trees optimization technique that selects the best spanning tree root for each message class in order to minimize the average hop count

and improve the application execution time. In Chapter 6, another routing scheme called mixed routing is also proposed considering the regularity of the wireless 3D NoC. It classifies the target topology into 2D mesh, 3D mesh and irregular topology, and selects the routing algorithm based on its topology. The evaluation results show that performance and power saving both are improved by using mixed routing.

We make a discussion on important issues for both design in Section 7.2. Finally, future directions are derived in Section 7.3.

7.2 Discussion

In Chapter 4, we proposed a low power technology to reduce the power consumption of the wireless 3D on chip network. The proposed low power strategy successfully reduced power consumption no matter for ring network or irregular network.

However, we did not take traffic load into consideration. Full system simulation results is showing a low traffic load. Then the wake up latency can be offset by low vertical link utilization. If block happened as traffic load becomes very high, the wake up signals are also possible to be blocked. Thus, the system performance would become worse. On the other hand, the proposed on/off vertical link method is now only effective for reducing power consumption. Since the static power consumption of wireless 3D NoC is difficult to measure, the tradition power gating and the proposed method cannot be verified effectively. Moreover, the power consumption might become lower as the process technology improves.

In Chapter 6, we proposed routing scheme and its improved version for general wireless 3D NoC. Full-system CMP simulations using parallel applications show that the proposed irregular approach outperforms the previous ring approach by up to 23.4%. The results also show that the proposed spanning trees optimization technique improves the application performance by up to 28.1%.

However, some problems are ignored. Calculation time mentioned in chapter 6 for the best spanning tree root for each message class is small when is network scale is small. When the network scale is 8-tile, the calculation time is 0.008 second, then 0.020 second for 16-tile, 0.315 second for 32-tile. Although 9.376 second for 64-tile is still small, the calculate time is 30 times as that of 32-tile. As the network scale increases, the calculate time grows exponentially.

For the its improved version, there are also some problems exist. We only consider the case of irregular topology and mesh topology, but neglect the other topologies. In addition, the routing algorithm can be switched only once, since we assume two virtual channels. Thus, Up*/Down* routing is applied from the relay-node to the destination nodes even if the destination node belongs to the another different meshes. If the system scale becomes large, add more virtual channel can make the routing algorithm switch more than once. However, obviously, the area and power consumption overhead will increase. Besides, the performance improvement is not big enough. The routing algorithm needs to be modified again for getting more performance benefits.

7.3 Future Work

Although the proposed low power policy works well for ring-based and irregular-based wireless 3D NoC in terms of evaluation results, we did not check the case of mesh-based wireless 3D NoC which is the most popular 3D NoC architecture. In the future, it is meaningful that apply the proposed low power method to mesh-based network.

The wake up latency is another problem which could effect the system performance. As the matter stands, look-ahead routing is a feasible plan. Router detects the arrival of packets two hops ahead, so as to mitigate the negative impacts of wakeup latency.

Moreover, the proposed power control mechanism works only dynamically. Considering the application property, the static on/off mechanism is advantageous since the target network allows any type of topologies. Shutting down of some links statically for the target applications will save much power without performance overhead. Combination of such a static control and dynamic control proposed here is a future direction.

It is also possible to modify the proposed mixed routing algorithm. Now, Mixed Routing simply uses layers which includes the source node or destination node. This strategy may cause the traffic congestion when a specific chip (e.g. processor) generates a lot of packets. Selecting the light traffic chips and switching the route is our next improvement policy. Furthermore, we must take other topology into consideration, because the current scheme is too simple to handle most cases. Checking the cases for many-core wireless 3D NoCs to verify if our proposed routing method is more efficient could be another considerable study.

Bibliography

- [1] Timothy Mark Pinkston and Jeonghee Shin. Trends Toward On-Chip Networked Microsystems. *International Journal of High Performance Computing and Networking*, 3(1):3–18, September 2005.
- [2] William J. Dally and Brian Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proceedings of the Design Automation Conference (DAC'01)*, pages 684–689, June 2001.
- [3] Luca Benini and Giovanni De Micheli. Networks on Chips: A New SoC Paradigm. *IEEE Computer*, 35(1):70–78, January 2002.
- [4] Luca Benini and Giovanni De Micheli. *Networks on Chips: Technology And Tools*. Morgan Kaufmann, 2006.
- [5] Sriram Vangal, Jason Howard, Gregory Ruhl, Saurabh Dighe, Howard Wilson, James Tschanz, David Finan, Priya Iyer, Arvind Singh, Tiju Jacob, Shailendra Jain, Sriram Venkataraman, Yatin Hoskote, and Nitin Borkar. An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'07)*, pages 184–185, February 2007.
- [6] Jose Duato, Sudhakar Yalamanchili, and Lionel M. Ni. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann, 2002.
- [7] William James Dally and Brian Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [8] William James Dally and Charles L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, 36(5):547–553, May 1987.
- [9] Thdore Marescaux, Andrei Bartic, Diederik Verkest, Serge Vernalde, and Rudy Lauwereins. Interconnection Networks Enable Fine-Grain Dynamic Multi-Tasking on FPGAs. In *Proceedings of the Field-Programmable Logic and Applications (FPL'02)*, pages 795–805, September 2002.
- [10] Jingcao Hu and Radu Marculescu. Energy-Aware Mapping for Tile-based NoC Architectures Under Performance Constraints. In *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 233–239, January 2003.
- [11] Nilanjan Banerjee, Praveen Vellanki, and Karam S. Chatha. A Power and Performance Model for Network-on-Chip Architectures. In *Proceedings of Design Automation and Test in Europe Conference (DATE'04)*, pages 1250–1255, February 2004.
- [12] Kenichiro Anjo, Yutaka Yamada, Michihiro Koibuchi, Akiya Jouraku, and Hideharu Amano. BLACK-BUS: A New Data-Transfer Technique using Local Address on Networks-on-Chips. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'04)*, April 2004.

- [13] Michihiro Koibuchi, Kenichiro Anjo, Yutaka Yamada, Akiya Jouraku, and Hideharu Amano. A Simple Data Transfer Technique using Local Address for Networks-on-Chips. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1425–1437, 2006.
- [14] Michael D. Schroeder, Andrew D. Birrell, Michael Burrows, Hal Murray, Roger M. Needham, and Thomas L. Rodeheffer. Autonet: A High-speed, Self-configuring Local Area Network Using Point-to-point Links. *IEEE Journal on Selected Areas in Communications*, 9:1318–1335, October 1991.
- [15] Vassos Soteriou and Li-Shiuan Peh. Exploring the Design Space of Self-Regulating Power-Aware On/Off Interconnection Networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(3):393–408, March 2007.
- [16] Seung Eun Lee and Nader Bagherzadeh. A variable frequency link for a power-aware network-on-chip (NoC). *Integr. VLSI J.*, 42:479–485, September 2009.
- [17] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS. *IEEE Journal of Solid-State Circuits*, 43(1):29–41, January 2008.
- [18] Hiroki Matsutani, Michihiro Koibuchi, Daihan Wang, and Hideharu Amano. Run-Time Power Gating of On-Chip Routers Using Look-Ahead Routing. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'08)*, pages 55–60, January 2008.
- [19] Bryan Black, Murali Annavaram, Ned Brekelbaum, John DeVale, Lei Jiang, Gabriel H. Loh, Don McCaule, Pat Morrow, Donald W. Nelson, Daniel Pantuso, Paul Reed, Jeff Rupley, Sadasivan Shankar, John Paul Shen, and Clair Webb. Die Stacking (3D) Microarchitecture. In *Proceedings of the International Symposium on Microarchitecture (MICRO'06)*, pages 469–479, December 2006.
- [20] Kuoichichi Kumagai, Changqi Yang, Satoshi Goto, Takeshi Ikenaga, Yoshihiro Mabuchi, and Kenji Yoshida. System-in-Silicon Architecture and its application to an H.264/AVC motion estimation fort 1080HDTV. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'06)*, pages 430–431, February 2006.
- [21] William Rhett Davis, John Wilson, Stephen Mick, Jian Xu, Hao Hua, Christopher Mineo, Ambarish M. Sule, Michael Steer, and Paul D. Franzon. Demystifying 3D ICs: The Pros and Cons of Going Vertical. *IEEE Design and Test of Computers*, 22(6):498–510, November 2005.
- [22] Kouichi Kanda, Danardonno Dwi Antono, Koichi Ishida, Hiroshi Kawaguchi, Tadahiro Kuroda, and Takayasu Sakurai. 1.27-Gbps/pin, 3mW/pin Wireless Superconnect (WSC) Interface Scheme. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'03)*, pages 186–187, February 2003.
- [23] Noriyuki Miura, Hiroki Ishikuro, Takayasu Sakurai, and Tadahiro Kuroda. A 0.14pJ/b Inductive-Coupling Inter-Chip Data Transceiver with Digitally-Controlled Precise Pulse Shaping. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'07)*, pages 358–359, February 2007.
- [24] Noriyuki Miura, Daisuke Mizoguchi, Mari Inoue, Kiichi Niitsu, Yoshihiro Nakagawa, Masamoto Tago, Muneo Fukaishi, Takayasu Sakurai, and Tadahiro Kuroda. A 1Tb/s 3W Inductive-Coupling Transceiver for Inter-Chip Clock and Data Link. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'06)*, pages 424–425, February 2006.

-
- [25] J. Burns, L. McIlrath, C. Keast, C. Lewis, A. Loomis, K. Warner, and P. Wyatt. Three-Dimensional Integrated Circuits for Low-Power High-Bandwidth Systems on a Chip. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'01)*, pages 268–269, February 2001.
 - [26] Abbas Sheibanyrad, Frederic Petrot, and Axel Janstch. *3D Integration for NoC-Based SoC Architectures*. Springer, 2010.
 - [27] Hiroki Matsutani, Michihiro Koibuchi, Yutaka Yamada, D. Frank Hsu, and Hideharu Amano. Fat H-Tree: A Cost-Efficient Tree-Based On-Chip Network. *IEEE Transactions on Parallel and Distributed Systems*, 20(8):1126–1141, August 2009.
 - [28] Brett Feero and Partha Pratim Pande. Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation. *IEEE Transactions on Computers*, 58(1):32–45, January 2009.
 - [29] Vasilis F. Pavlidis and Eby G. Friedman. 3-D Topologies for Networks-on-Chip. *IEEE Transactions on Very Large Scale Integration Systems*, 15(10):1081–1090, October 2007.
 - [30] Jongman Kim, Chrysostomos Nicopoulos, Dongkook Park, Reetuparna Das, Yuan Xie, Narayanan Vijaykrishnan, Mazin Yousif, and Chita Das. A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architectures. In *Proceedings of the International Symposium on Computer Architecture (ISCA'07)*, pages 138–149, June 2007.
 - [31] Feihui Li, Chrysostomos Nicopoulos, Thomas Richardson, Yuan Xie, Vijaykrishnan Narayanan, and Mahmut Kandemir. Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. In *Proceedings of the International Symposium on Computer Architecture (ISCA'06)*, pages 130–141, June 2006.
 - [32] Dongkook Park, Soumya Eachempati, Reetuparna Das, Asit K Mishra, Vijaykrishnan Narayanan, Yuan Xie, and Chita R Das. MIRA: A Multi-layered On-Chip Interconnect Router Architecture. In *Proceedings of the International Symposium on Computer Architecture (ISCA'08)*, pages 251–261, June 2008.
 - [33] Rohit Sunkam Ramanujam and Bill Lin. Randomized Partially-Minimal Routing on Three-Dimensional Mesh Networks. *IEEE Computer Architecture Letters*, 7(2):37–40, July 2008.
 - [34] Hiroki Matsutani, Yasuhiro Take, Daisuke Sasaki, Masayuki Kimura, Yuki Ono, Yukinori Nishiyama, Michihiro Koibuchi, Tadahiro Kuroda, and Hideharu Amano. A Vertical Bubble Flow Network using Inductive-Coupling for 3-D CMPs. In *Proceedings of the International Symposium on Networks-on-Chip (NOCS'11)*, pages 49–56, May 2011.
 - [35] Davide Bertozzi and Luca Benini. Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip. *IEEE Circuits and Systems Magazine*, 4(2):18–31, 2004.
 - [36] Jingcao Hu and Radu Marculescu. Energy- and Performance-Aware Mapping for Regular NoC Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(4):551–562, April 2005.
 - [37] Michael Bedford Taylor, Jason Sungtae Kim, Jason E. Miller, David Wentzlaff, Fae Ghodrati, Ben Greenwald, Henry Hoffmann, Paul Johnson, Jae-Wook Lee, Walter Lee, Albert Ma, Arvind Saraf, Mark Seneski, Nathan Shnidman, Volker Strumpfen, Matthew Frank, Saman P. Amarasinghe, and Anant Agarwal. The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs. *IEEE Micro*, 22(2):25–35, April 2002.

- [38] William James Dally. Virtual-Channel Flow Control. *IEEE Transactions on Parallel and Distributed Systems*, 3(2):194–205, 1992.
- [39] Robert Mullins, Andrew West, and Simon Moore. Low-Latency Virtual-Channel Routers for On-Chip Networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA'04)*, pages 188–197, June 2004.
- [40] Robert Mullins, Andrew West, and Simon Moore. The Design and Implementation of a Low-Latency On-Chip Network. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'06)*, pages 164–169, January 2006.
- [41] Jongman Kim, Chrysostomos Nicopoulos, Dongkook Park, Vijaykrishnan Narayanan, Mazin S. Yousif, and Chita R. Das. A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA'06)*, pages 14–15, June 2006.
- [42] Andrew A. Chien. A Cost and Speed Model for k-ary n-Cube Wormhole Routers. *IEEE Transactions on Parallel and Distributed Systems*, 9(2):150–162, February 1998.
- [43] Charles E. Leiserson. Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, October 1985.
- [44] Andre DeHon. Unifying Mesh- and Tree-Based Programmable Interconnect. *IEEE Transactions on Very Large Scale Integration Systems*, 12(10):1051–1065, October 2004.
- [45] Charles E. Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong-Chan, Shaw-Wen Yang, and Robert Zak. The Network Architecture of the Connection Machine CM-5. *Journal of Parallel and Distributed Computing*, 33(2):145–158, March 1996.
- [46] Cristian Grecu, Partha Pratim Pande, Andre Ivanov, and Res Saleh. Structured Interconnect Architecture: A Solution for the Non-Scalability of Bus-Based SoCs. In *Proceedings of the Great Lakes Symposium on VLSI (GVLSI'04)*, pages 192–195, April 2004.
- [47] Christopher J. Glass and Lionel M. Ni. The Turn Model for Adaptive Routing. In *Proceedings of the International Symposium on Computer Architecture (ISCA'92)*, pages 278–287, May 1992.
- [48] J. Kao, A. Chandrakasan, and D. Antoniadis. Transistor sizing issues and tool for multi-threshold cmos technology. In *Design Automation Conference, 1997. Proceedings of the 34th*, pages 409–414, 1997.
- [49] M. Liu, Wei-Shen Wang, and M. Orshansky. Leakage power reduction by dual-vth designs under probabilistic analysis of vth variation. In *Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on*, pages 2–7, 2004.
- [50] C.A. Nicopoulos, Dongkook Park, Jongman Kim, N. Vijaykrishnan, M.S. Yousif, and C.R. Das. Vichar: A dynamic virtual channel regulator for network-on-chip routers. In *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, pages 333–346, 2006.
- [51] Avinash Kodi, Ashwini Sarathy, and Ahmed Louri. Design of adaptive communication channel buffers for low-power area-efficient network-on-chip architecture. In *Proceedings of the 3rd ACM/IEEE Symposium on Architecture for Networking and Communications Systems, ANCS '07*, pages 47–56, New York, NY, USA, 2007. ACM.

-
- [52] Xuning Chen and Li-Shiuan Peh. Leakage Power Modeling and Optimization in Interconnection Networks. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'03)*, pages 90–95, August 2003.
 - [53] Masakatsu Nakai, Satoshi Akui, Katsunori Seno, Tetsumasa Meguro, Takahiro Seki, Tetsuo Kondo, Akihiko Hashiguchi, Hirokazu Kawahara, Kazuo Kumano, and Masayuki Shimura. Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor. *IEEE Journal of Solid-State Circuits*, 40(1):28–35, January 2005.
 - [54] Kevin Nowka, Gary Carpenter, Eric Mac Donald, Hung Ngo, Bishop Brockand Koji Ishii, Tuyet Nguyen, and Jeffrey Burn. A 0.9V to 1.95V Dynamic Voltage-Scalable and Frequency-Scalable 32b PowerPC Processor. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'02)*, pages 340–341, February 2002.
 - [55] Kentaro Kawakami, Jun Takemura, Mitsuhiro Kuroda, Hiroshi Kawaguchi, and Masahiko Yoshimoto. A 50% Power Reduction in H.264/AVC HDTV Video Decoder LSI by Dynamic Voltage Scaling in Elastic Pipeline. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(12):3642–3651, December 2006.
 - [56] Li Shang, Li-Shiuan Peh, and Niraj K. Jha. Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA'03)*, pages 79–90, January 2003.
 - [57] Jeffrey M. Stine and Nicholas P. Carter. Comparing Adaptive Routing and Dynamic Voltage Scaling for Link Power Reduction. *IEEE Computer Architecture Letters*, 3(1):14–17, January 2004.
 - [58] Makoto Ishikawa, Tetsuya Kamei, Yuki Kondo, Masanao Yamaoka, Yasuhisa Shimazaki, Motokazu Ozawa, Saneaki Tamaki, Mikio Furuyama, Tadashi Hoshi, Fumio Arakawa, Osamu Nishii, Kenji Hirose, Shinichi Yoshioka, and Toshihiro Hattori. A 4500 MIPS/W, 86 μ A Resume-Standby, 11 μ A Ultra-Standby Application Processor for 3G Cellular Phones. *IEICE Transactions on Electronics*, E88-C(4):528–535, April 2005.
 - [59] Zhigang Hu, Alper Buyuktosunoglu, Viji Srinivasan, Victor Zyuban, Hans Jacobson, and Pradip Bose. Microarchitectural Techniques for Power Gating of Execution Units. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'04)*, pages 32–37, August 2004.
 - [60] Kimiyoshi Usami and Naoaki Ohkubo. A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals. In *Proceedings of the International Conference on Computer Design (ICCD'06)*, October 2006.
 - [61] Vassos Soteriou and Li-Shiuan Peh. Design-Space Exploration of Power-Aware On/Off Interconnection Networks. In *Proceedings of the International Conference on Computer Design (ICCD'04)*, pages 510–517, October 2004.
 - [62] Michael Bedford Taylor, Walter Lee, Jason E. Miller, David Wentzlaff, Ian Bratt, Ben Greenwald, Henry Hoffmann, Paul Johnson, Jason Sungtae Kim, James Psota, Arvind Saraf, Nathan Shnidman, Volker Strumpfen, Matthew Frank, Saman P. Amarasinghe, and Anant Agarwal. Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams. In *Proceedings of the International Symposium on Computer Architecture (ISCA'04)*, pages 2–13, June 2004.
 - [63] Ron Ho, Kenneth W. Mai, and Mark A. Horowitz. The Future of Wires. *Proceedings of the IEEE*, 89(4):490–504, April 2001.

- [64] Bryan Black, Donald W. Nelson, Clair Webb, and Nick Samra. 3D Processing Technology and Its Impact on iA32 Microprocessors. In *Proceedings of the International Conference on Computer Design (ICCD'04)*, pages 316–318, October 2004.
- [65] Shamik Das, Andy Fan, Kuan-Neng Chen, Chuan Seng Tan, Nisha Checka, and Rafael Reif. Technology, Performance, and Computer-Aided Design of Three-Dimensional Integrated Circuits. In *Proceedings of the International Symposium on Physical Design (ISPD'04)*, pages 108–115, April 2004.
- [66] Vasilis F. Pavlidis and Eby G. Friedman. 3-D Topologies for Networks-on-Chip. In *Proceedings of the International System-on-Chip Conference (SOCC'06)*, pages 285–288, September 2006.
- [67] Hiroki Matsutani, Michihiro Koibuchi, and Hideharu Amano. Tightly-Coupled Multi-Layer Topologies for 3-D NoCs. In *Proceedings of the International Conference on Parallel Processing (ICPP'07)*, September 2007.
- [68] Shotaro Saito, Yoshinori Kohama, Yasufumi Sugimori, Yohei Hasegawa, Hiroki Matsutani, Toru Sano, Kazutaka Kasuga, Yoichi Yoshida, Kiichi Niitsu, Noriyuki Miura, Tadahiro Kuroda, and Hideharu Amano. MuCCRA-Cube: a 3D Dynamically Reconfigurable Processor with Inductive-Coupling Link. In *Proceedings of the Field-Programmable Logic and Applications (FPL'09)*, pages 6–11, September 2009.
- [69] Thomas William Ainsworth and Timothy Mark Pinkston. Characterizing the Cell EIB On-Chip Network. *IEEE Micro*, 27(5):6–14, September 2007.
- [70] Bradford M. Beckmann and David A. Wood. Managing Wire Delay in Large Chip-Multiprocessor Caches. In *Proceedings of the International Symposium on Microarchitecture (MICRO'04)*, pages 319–330, December 2004.
- [71] Paul Gratz, Changkyu Kim, Karthikeyan Sankaralingam, Heather Hanson, Premkishore Shivakumar, Stephen W. Keckler, and Doug Burger. On-Chip Interconnection Networks of the TRIPS Chip. *IEEE Micro*, 27(5):41–50, September 2007.
- [72] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. R-NUCA: Data Placement in Distributed Shared Caches. In *Proceedings of the International Symposium on Computer Architecture (ISCA'09)*, pages 184–195, June 2009.
- [73] Jason Howard et al. A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'10)*, pages 108–109, February 2010.
- [74] Ana Sonia Leon, Kenway W. Tam, Jinuk Luke Shin, David Weisner, and Francis Schumacher. A Power-Efficient High-Throughput 32-Thread SPARC Processor. *IEEE Journal of Solid-State Circuits*, 42(1):7–16, January 2007.
- [75] David Wentzlaff, Patrick Griffin, Henry Hoffmann, Liewei Bao, Bruce Edwards, Carl Ramey, Matthew Mattina, Chyi-Chang Miao, John F. Brown III, and Anant Agarwal. On-Chip Interconnection Architecture of the Tile Processor. *IEEE Micro*, 27(5):15–31, September 2007.
- [76] Changkyu Kim, Doug Burger, and Stephen W. Keckler. An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'02)*, pages 211–222, October 2002.

-
- [77] Milo M. K. Martin, Mark D. Hill, and David A. Wood. Token Coherence: Decoupling Performance and Correctness. In *Proceedings of the International Symposium on Computer Architecture (ISCA'03)*, pages 182–193, June 2003.
 - [78] Yan Solihin. *Fundamentals of Parallel Computer Architecture*. Solihin Publishing & Consulting LLC, 2009.
 - [79] Gabriel H. Loh. 3D-Stacked Memory Architectures for Multi-core Processors. In *Proceedings of the International Symposium on Computer Architecture (ISCA'08)*, pages 453–464, June 2008.
 - [80] Niti Madan, Li Zhao, Naveen Muralimanohar, Aniruddha N. Udiipi, Rajeev Balasubramonian, Ravishankar Iyer, Srihari Makineni, and Donald Newell. Optimizing Communication and Capacity in a 3D Stacked Reconfigurable Cache Hierarchy. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA'09)*, pages 262–274, February 2009.
 - [81] Yuan Xie, Gabriel H. Loh, Bryan Black, and Kerry Bernstein. Design Space Exploration for 3D Architectures. *ACM Journal on Emerging Technologies in Computing Systems*, 2:65–103, April 2006.
 - [82] Andreas Merkel, Jan Stoess, and Frank Bellosa. Resource-Conscious Scheduling for Energy Efficiency on Multicore Processors. In *Proceedings of the European Conference on Computer Systems (EuroSys'10)*, pages 153–166, April 2010.
 - [83] Sergey Zhuravlev, Sergey Blagodurov, and Alexandra Fedorova. Addressing Shared Resource Contention in Multicore Processors via Scheduling. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'10)*, pages 129–142, March 2010.
 - [84] N. J. Boden, Daniel I. A. Cohen, Robert Felderman, Alan E. Kulawik, Charles L. Seitz, Jakov N. Seizovic, and Wen king Su. Myrinet: A Gigabit-per-Second Local Area Network. *IEEE Micro*, 15(1):29–35, February 1995.
 - [85] N. Miura, K. Kasuga, M. Saito, and T. Kuroda. An 8Tb/s 1pJ/b 0.8mm²/Tb/s QDR Inductive-Coupling Interface Between 65nm CMOS and 0.1um DRAM. In *Proceedings of the International Solid-State Circuits Conference (ISSCC'10)*, pages 436–437, February 2010.
 - [86] Yasuhiro Take, Hiroki Matsutani, Daisuke Sasaki, Michihiro Koibuchi, Tadahiro Kuroda, and Hideharu Amano. 3-d noc with inductive-coupling links for building-block sips. *IEEE Transactions on Computers*, 99:1, 2012.
 - [87] D. Gebhardt and K. Stevens. Power reduction through physical placement of asynchronous routers. In *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, page 92, may 2009.
 - [88] M. Yamada, T. Yazaki, S. Nishimura, and N. Ikeda. Technologies to save power for carrier class routers and switches. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 385–388, 28 2008-aug. 1 2008.
 - [89] Yusuke Koizumi, Hideharu Amano, Hiroki Matsutani, Noriyuki Miura, Tadahiro Kuroda, Ryuichi Sakamoto, Mitaro Namiki, Kimiyoshi Usami, Masaaki Kondo, and Hiroshi Nakamura. CMA-Cube: A Scalable Reconfigurable Accelerator with 3-D Wireless Inductive Coupling Interconnect. In *the Demo Session at the 11th IEEE International Conference on Field Programmable Technology (ICFPT'12)*, December 2012.

- [90] N.Ozaki, Y.Yasuda, Y.Saito, D.Ikebuchi, M.Kimura, H.Amano, H.Nakamura, K.Usami, M.Namiki, M.Kondo . Cool Mega-Arrays: Ultralow-Power Reconfigurable Accelerator Chips. *IEEE Micro*, Vol.31, pages 6–18, 2011.
- [91] Peter S. Magnusson et al. Simics: A Full System Simulation Platform. *IEEE Computer*, 35(2):50–58, February 2002.
- [92] Milo M. K. Martin, Daniel J. Sorin, Bradford M. Beckmann, Michael R. Marty, Min Xu, Alaa R. Alameldeen, Kevin E. Moore, Mark D. Hill, and David A. Wood. Multifacet General Execution-driven Multiprocessor Simulator (GEMS) Toolset. *ACM SIGARCH Computer Architecture News (CAN'05)*, 33(4):92–99, November 2005.
- [93] Niket Agarwal, Li-Shiuan Peh, and Niraj Jha. Garnet: A Detailed Interconnection Network Model inside a Full-system Simulation Framework. Technical Report CE-P08-001, Princeton University, 2008.
- [94] H. Jin, M. Frumkin, and J. Yan. The OpenMP Implementation of NAS Parallel Benchmarks and Its Performane. In *NAS Technical Report NAS-99*, October 1999.
- [95] Hiroki Matsutani, Paul Bogdan, Radu Marculescu, Yasuhiro Take, Daisuke Sasaki, Hao Zhang, Michihiro Koibuchi, Tadahiro Kuroda, and Hideharu Amano. A Case for Wireless 3D NoCs for CMPs. In *Proceedings of the 18th Asia and South Pacific Design Automation Conference (ASP-DAC'13)*, January 2013.
- [96] Pablo Abad, Valentin Puente, Pablo Prieto, and Jose Angel Gregorio. Rotary Router: An Efficient Architecture for CMP Interconnection Networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA'07)*, pages 116–125, May 2007.
- [97] V. Puente, R. Beivide, J. A. Gregorio, J. M. Prellezo, J. Duato, and C. Izu. Adaptive Bubble Router: A Design to Improve Performance in Torus Networks. In *Proceedings of the International Conference on Parallel Processing (ICPP'99)*, pages 58–67, September 1999.
- [98] Li Shang, Li-Shiuan Peh, and N.K. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of the Ninth International Symposium on High-Performance Computer Architecture (HPCA2003)*, pages 91–102, February 2003.
- [99] J.Flich and et.al. A Survey and Evaluation of Topology Agnostic Deterministic Routing Algorithms. *IEEE Trans. on Parallel and Distributed Systems*, 23(3):405–425, 2012.
- [100] Akiya Jouraku, Michihiro Koibuchi, and Hideharu Amano. An Effective Design of Deadlock-Free Routing Algorithms Based on 2-D Turn Model for Irregular Networks. *IEEE Transactions on Parallel and Distributed Systems(TPDS)*, 18(3):320–333, March 2007.

Publications

Related Papers

Journal Papers

- [1] Hao Zhang, Hiroki Matsutani, Yasuhiro Take, Tadahiro Kuroda, Hideharu Amano, “Vertical Link On/Off Regulations for Inductive-Coupling Based 3-D NoCs”, *IEICE Transactions on Information and Systems*, Vol.E96-D, No. 12, pp. 2753–2764, Dec 2013.
- [2] Hao Zhang, Hiroki Matsutani, Michihiro Koibuchi, Hideharu Amano, “Dynamic Power Consumption Optimization for Inductive-Coupling based Wireless 3D NoCs”, *IPSSJ Transaction on System LSI Design Methodology*, Vol. xx, No. xx, pp. xxx–xxx, xx 2014.

International Conference Papers

- [3] Hao Zhang, Hideharu Amano, “Research on a Low Power Hybrid-Mesh Networks on Chip” *Proc. of IEEE 1st International Symposium on Access Spaces (IEEE-ISAS’11)*, poster session, Jun 2011.
- [4] Hao Zhang, Hiroki Matsutani, Yasuhiro Take, Tadahiro Kuroda, Hideharu Amano, “Vertical Link On/Off Control Methods for Wireless 3-D NoCs”, *Proc. of the 25th International Conference on Architecture of Computing Systems (ARCS’12)*, pp. 212-224, Feb 2012.
- [5] Hiroki Matsutani, Paul Bogdan, Radu Marculescu, Yasuhiro Take, Daisuke Sasaki, Hao Zhang, Michihiro Koibuchi, Tadahiro Kuroda and Hideharu Amano, “A Case for Wireless 3D NoCs for CMPs”, *Proc. of the 18th Asia and South Pacific Design Automation Conference (ASP-DAC’13)*, pp. 23-28, Jan 2013.
- [6] Hao Zhang, Hiroki Matsutani, Michihiro Koibuchi, Hideharu Amano, “Dynamic Power On/Off Method for 3D NoCs with Wireless Inductive-Coupling Links”, *Proc. of the 16th IEEE Symposium on Low-Power and High-Speed Chips (COOLCHIPS XVI)*, pp. 1-4, Apr 2013.
- [7] Daisuke Sasaki, Hao Zhang, Hiroki Matsutani, Michihiro Koibuchi, Hideharu Amano, “A Routing Strategy for Inductive-Coupling Based Wireless 3-D NoCs by Maximizing Topological Regularity”, *Proc. of the 13th International Conference on Algorithm and Architecture for Parallel Processing (ICA3PP’13)*, pp. 77-85, Dec 2013.