

Tightly-Coupled Multi-Layer Topologies for 3-D NoCs *

Hiroki Matsutani¹, Michihiro Koibuchi², and Hideharu Amano¹

¹Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama,
JAPAN 223-8522
{matutani,hunga}@am.ics.keio.ac.jp

²National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo,
JAPAN 101-8430
koibuchi@nii.ac.jp

Abstract

Three-dimensional Network-on-Chip (3-D NoC) is an emerging research topic exploring the network architecture of 3-D ICs that stack several smaller wafers for reducing wire length and wire delay. Although the network topology of 3-D NoC has been explored for a couple of years, there is still only a narrow range of choices.

In this paper, we propose a class of 3-D topologies called Xbar-connected Network-on-Tiers (XNoTs), which consist of multiple network layers tightly connected via crossbar switches. To make the best use of the short delay and high density of inter-wafer links, XNoTs topologies have crossbar switches that connect different layers and their cores. The planar topology on every layer can be independently customized so as to meet the cost-performance requirements, as far as network connectivity is at least guaranteed with the bottom layer. We also propose their routing algorithm, which guarantees deadlock-freedom by restricting the inter-layer packet transfer from a lower-numbered layer to a higher-numbered layer. Path sets at the bottom layer close to the heat sink of the chip can be selectively employed in order to mitigate the heat-dissipation problem of 3-D ICs. Several forms of XNoTs topologies including meshes, tori, and/or trees are created, and they are evaluated in terms of performance, cost, and energy consumption. As a result, we show that even with the flexibilities mentioned above, XNoTs achieve at least as high throughput as existing 3-D topologies for equivalent chip sizes.

1 Introduction

Network-on-Chips (NoCs) have been studied to connect a number of processing cores on a single chip by in-

*This work was supported by Joint Research Fund, "Network-on-Chip Architecture," National Institute of Informatics.

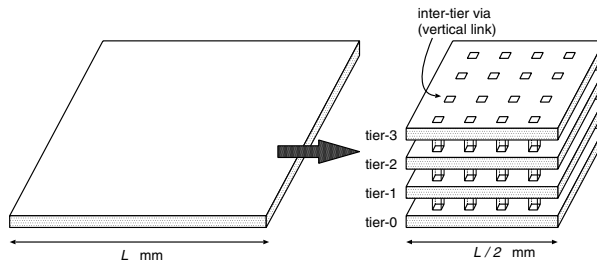


Figure 1. 2-D IC (left) and 3-D IC (right)

roducing a network structure similar to that of parallel computers[4]. They have been utilized not only for high-performance microarchitectures but also for cost-effective embedded devices used in consumer equipment such as set-top boxes or mobile wireless devices. Such embedded applications often demand very tight design constraints in terms of cost and performance; thus the silicon budget available for their on-chip network infrastructure should be modest as long as the required performance is met. NoC architecture provides a wide design space including network topology, routing algorithm, and router architecture, all of which affect the system performance at the expense of different amounts of network resources; therefore the network architecture for such embedded applications should be carefully selected so as to meet the requirements.

The current concept of NoCs is being extended to ICs that have three-dimensional structures, and it is expected to mitigate the wire delay, which is increasingly posing severe problems to modern VLSI design. One of the attractive solutions to the wire delay problem is three-dimensional IC technology that stacks multiple wafers or dice using vertical interconnects[2, 6, 7, 12]. Figure 1 shows examples of a 2-D IC and a 3-D IC that has equivalent resources to the planar one. Obviously, size of each tier (a tier refers a wafer or a die in a 3-D IC) can be downsized according to the number of tiers stacked together. Assuming that four tiers

are stacked in a 3-D IC as shown in the figure, each wafer area is reduced into 1/4 and the length of the longest wire in a wafer becomes 1/2 compared with the planar one. Since the wire delay is related to the square of the wire length, the maximum delay becomes 1/4 and the number of repeaters can be reduced as well.

Various 3-D interconnect approaches have been proposed: wire-bonding between stacked chips, microbump technology[2], contactless (i.e., wireless), and through-via between stacked wafers[6, 7]. In this paper, we assume through-wafer via technology, which is expected to offer both very high density of vertical interconnects and very short distance between wafers. The distance between wafers can range from $5\mu\text{m}$ to $50\mu\text{m}$ [12], which is much shorter than the wire length between cores on a tier, and the pitches of a through-wafer via can range from $1\mu\text{m}$ to $10\mu\text{m}$ square[6, 7, 12], depending on the manufacturing process such as wafer-to-wafer alignment. Although the size of a through-wafer via is smaller than those of other approaches (e.g., wire-bonding), it is still large compared with horizontal wire pitches. Thus, we need to take account of the area overhead required for the vertical interconnects when we design a topology for 3-D NoCs.

Three-dimensional NoC is an emerging research topic that explores the network architecture of 3-D ICs. Although their network topology has been explored for a couple of years and 3D-meshes[1] and vertical buses[12] have been used, there is still only a narrow range of choices. In this paper, we propose a class of 3-D topologies called Xbar-connected Network-on-Tiers (XNoTs), which consist of multiple network layers tightly connected via crossbar switches. To make the best use of the short delay and high density of inter-tier links, XNoTs topologies have crossbar switches that connect different tiers and their cores, and each tier can be independently customized so as to meet the cost-performance requirements. We also propose a deadlock-free routing algorithm for XNoTs. Several forms of XNoTs topologies including meshes, tori, and/or trees are created, and they are evaluated in terms of performance, cost, and energy consumption. Their pros and cons are discussed based on the evaluation results.

The rest of this paper is organized as follows. Section 2 surveys on-chip and 3-D network topologies. In Section 3, we provide a definition of XNoTs, typical examples of XNoTs, and a deadlock-free routing algorithm on them. Several forms of XNoTs are evaluated in Section 4, and conclusions are derived in Section 5.

2 Network Topologies

Figures 2(a)-2(e) show typical on-chip network topologies with 16 cores, where a white circle represents a network interface of the core and a shaded square represents a

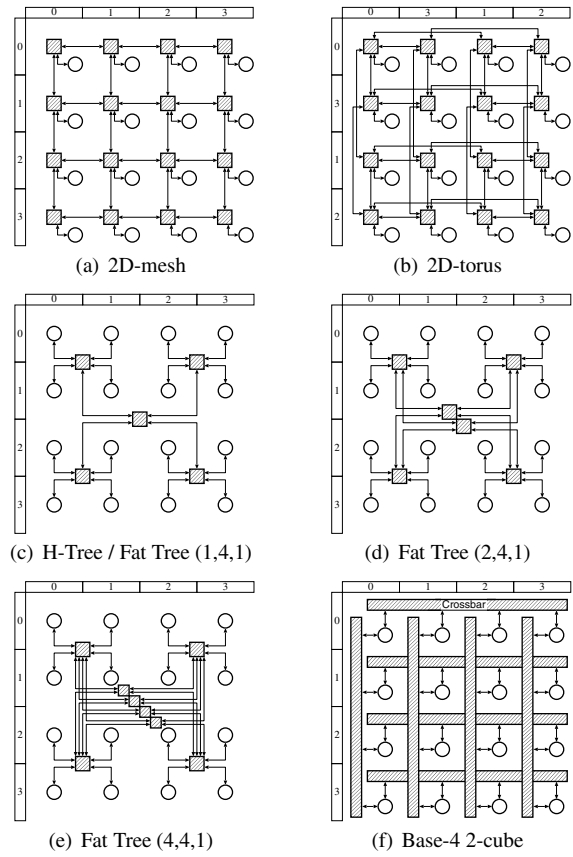


Figure 2. Typical network topologies

router connecting other routers or network interfaces. 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.

Network topologies can be classified into direct networks or indirect networks. A direct network consists of directly interconnected nodes (or tiles), each of which has a processing core, a router, and a network interface (NI). On the other hand, the router and core are separated in the case of an indirect network. That is, nodes consisting of a processing core and an NI are indirectly connected via routers. Typical direct and indirect networks are introduced here.

2.1 Direct Networks

Mesh and torus are popular in the microarchitecture domain, because their grid-based regular arrangement intuitively matches the two-dimensional VLSI layout. 2D-mesh based interconnects have been widely employed in 2-D microarchitectures (e.g., MIT RAW[15] and TRIPS Edge[3]).

As for the 3-D microarchitecture, 3D-mesh is a straightforward topology simply extended from 2D-mesh and it is expected to achieve good performance. However, as authors in [12] observed, 3D-mesh does not fully exploit the benefits of a 3-D microarchitecture because of the following reasons. 1) Compared with a router in 2D-mesh, one in 3D-mesh requires two additional channels for vertical connections (i.e., up and down), which require a larger crossbar switch and more channel buffers. 2) Tiers can be stacked very closely and latency for moving between tiers would be much smaller than that for core-to-core communications within a tier; therefore it is not necessary to buffer the flits at a router in every tier when they move vertically.

We propose a class of 3-D topologies that use crossbar switches for connecting different tiers and cores, in order to make the best use of the high-bandwidth of inter-wafer links. Since different tiers and several cores are connected via a crossbar switch, average hop count would be reduced, resulting lower energy consumption.

2.2 Indirect Networks

Tree-based topologies are typical examples of indirect networks. As well as mesh and torus, constant attention has been focused on tree-based ones, because of their relatively short hop-count that enables lower latency communication compared with mesh or torus.

As shown in Figure 2(c), every router in H-Tree has one upward and four downward connections, except for the root. Although H-Tree would be placed on a square die of a VLSI chip, the topology is equivalent to a simple tree, so it still has a common weak point of a tree. That is, links or routers around the root of the tree are frequently congested.

To mitigate the congestion around the root of the tree, Fat Tree increases the number of connections toward the root[11]. As illustrated in [11], various forms of Fat Tree can be created, and they can be expressed with a tuple (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(d) shows a Fat Tree (2,4,1), in which each router (except for top-rank routers) has two upward and four downward connections, and each core has one upward connection. SPIN architecture employs a very rich one labeled with (4,4,1), which means every router has four upward connections (Figure 2(e)). Note that a Fat Tree (1,4,1) is identical to the H-Tree mentioned above.

To the best of our knowledge, tree-based topologies have not been studied for 3-D NoCs, but the 3-D topologies proposed in this paper can include such trees. Another class of indirect networks is generated by replacing the directly connected links of n -dimensional mesh into crossbars. Figure 2(f) shows such networks, called base- m n -cube[14] or

Hypermesh[13]. In these networks, each node has a router for the n -dimensional connection, and for each dimension, a crossbar is provided to connect all links from the routers. Since the interconnecting capability of a crossbar is much higher than that of the directly connected links, this class of topology can achieve wider bandwidth and lower latency compared with corresponding direct networks. 3-D hyper-crossbar (base- m 3-cube) fits the 3-D implementation.

The defect of this class of topology is the large amount of hardware required for a large number of crossbars with a lot of ports, when the size becomes large. Only super-computers (e.g., CP-PACS and Hitachi's SR8000) that allow an expensive interconnection can use it. The only exception is the interconnection used in CPLD (Complex Programmable Logic Device), since extremely low-cost static crossbars without any buffers and arbiters can be used in such a programmable device.

3 Xbar-connected Network-on-Tiers

In this section, we propose a class of 3-D topologies called Xbar-connected Network-on-Tiers (XNoTs), which consist of multiple network layers tightly connected via crossbar switches. Each network layer can adopt an arbitrary topology (direct or indirect), whereas the network layers and cores are indirectly connected via crossbar switches.

3.1 Definition

Before defining of XNoTs, we define general networks on 3-D ICs as Network-on-Tiers.

Definition 1 A **Network-on-a-Tier (NoT)** is a planar on-chip network built on a tier, and it has inter-tier links for connecting to the other tiers.

Figure 3 shows a mesh-based NoT consisting of four routers and four partitioned cores. Each core has two-dimensional coordinates (x, y) , and each router has an inter-tier link. We call such a router a "tier router".

Three-dimensional coordinates (x, y, z) are assigned to each core on a 3-D NoC, as shown in Figure 4.

Definition 2 On the n -stacked NoTs, the **pillar** (x, y) is a set of inter-tier links placed on all cores which have the coordinates (x, y, z) , where $0 \leq z < n$.

Figure 4 shows an example of a three-stacked NoTs with four pillars.

Definition 3 An n -stacked **Xbar-connected NoTs topology (XNoTs)** is an n -stacked NoTs, where all inter-tier links on the pillar (x, y) and all local links to cores (x, y, z) are connected via a crossbar switch, where $0 \leq z < n$. At least the bottom NoT must guarantee reachability between any pair of cores.

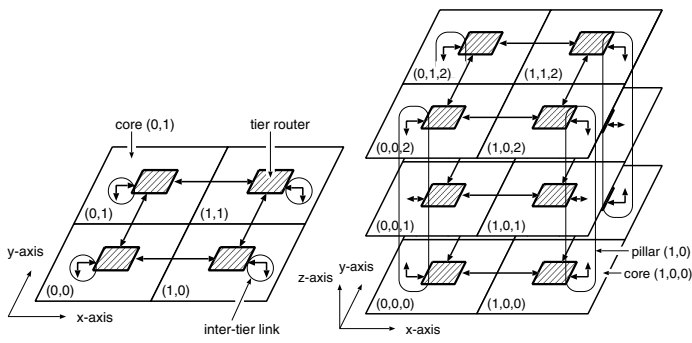


Figure 3. NoT

Figure 4. Stacked NoTs with four pillars

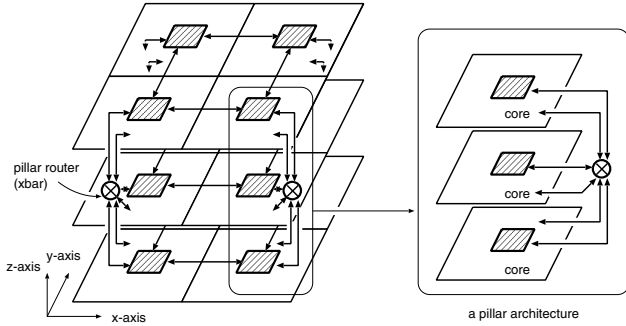


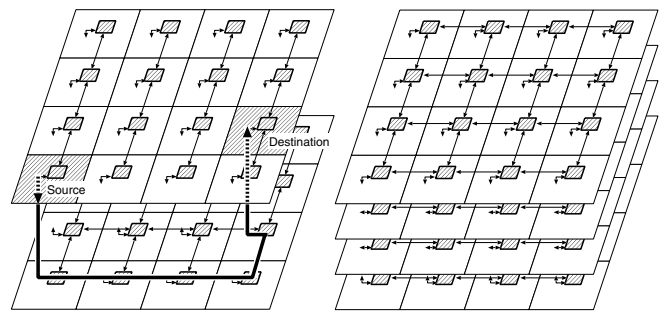
Figure 5. XNoTs

Figure 5 shows an example of an XNoTs topology, in which all inter-tier links and cores in every pillar are connected via a crossbar switch. We call such a crossbar switch a “pillar router” in order to distinguish it from tier routers.

Notice that since at least the bottom NoT (i.e., NoT on tier-0) always guarantees reachability between any pair of cores via pillar routers by itself, the same is not mandatory for the other NoTs. For instance, Figure 6(a) shows a two-stacked XNoTs topology, where the bottom NoT guarantees reachability between all cores but the upper one does not. Therefore, reachability between any pair of cores on tier-1 is still guaranteed by the bottom NoT, even if it is not guaranteed within the NoT on tier-1. In addition, it is possible to selectively employ a path set at the specific tiers, such as the bottom tier, which is close to the heat sink of the chip, in order to mitigate the heat-dissipation problem of 3-D ICs.

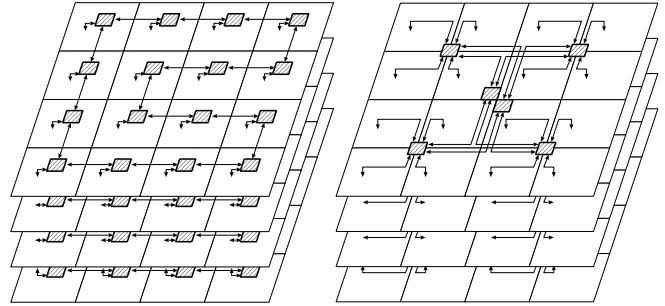
Various forms of XNoTs can be created, since XNoTs provide a flexibility to employ a proper network topology on each tier so as to meet the cost-performance requirements of the target application.

As an example of XNoTs, we created a mesh-based XNoTs topology called “X-mesh” (Figure 6(b)). Similarly, we created the following XNoTs topologies: a torus-based XNoTs topology (“X-torus”), a ring-based XNoTs topology (“X-ring”, Figure 6(c)), a tree-based XNoTs topology (“X-



(a) Different NoTs in an XNoTs

(b) X-mesh



(c) X-ring

(d) X-ft241

Figure 6. Examples of XNoTs topologies (pillar routers are not shown for simplicity)

ft241”, Figure 6(d))¹, and an asymmetric XNoTs topology that consists of different planar topologies (Figure 6(a)).

3.2 Deadlock-Free Routing Algorithm

Various routing algorithms can be utilized on an XNoTs topology as well as on the topology used in each tier. Routing algorithms are categorized as adaptive routing or deterministic routing. Adaptive routing that dynamically changes paths of packets can offer high channel utilization, but, unlike deterministic routing in which paths are statically fixed, it cannot guarantee in-order packet delivery that a communication protocol often requires. The designer thus selects one of them, depending on the system requirements.

Both adaptive and deterministic routings for XNoTs are designed using the following procedure. a) Impose the conditions for all intra-NoT communications to avoid deadlocks. b) Impose the conditions for all inter-NoT communications to avoid deadlocks. c) Search for the shortest paths that satisfy the above conditions.

a) Restrictions on Intra-NoT Communications

The routing restrictions on intra-tier communications

¹We named this tree-based XNoTs topology “X-ft241”, because a Fat Tree (2,4,1) is used for each NoT in it, as shown in Figure 6(d).

must satisfy the following two conditions: 1) deadlock-freedom is guaranteed as long as every packet is routed inside a single NoT; 2) the reachability between any pair of cores is guaranteed using only the bottom NoT.

Various deadlock-free routing algorithms or conditions can be used for the routing restrictions on each NoT, such as dimension-order routing (DOR) for meshes and tori[5], and up*/down* routing for arbitrary irregular topologies. Notice that the packet transfer across tiers is beyond the scope of the restrictions, because the restrictions work for the packet transfer within a single NoT.

b) Restrictions on Inter-NoT Communications

Packet transfer from a lower-numbered NoT to a higher-numbered NoT is prohibited in order to prevent deadlocks across tiers, unless the next hop is a pillar router directly connected to the destination.

c) Search Shortest Paths

Search for the shortest paths between every pair of cores by using Dijkstra’s algorithm under the conditions that (i) each channel has the same constant cost, and (ii) all channel transitions on prohibited turns are forbidden.

Alternative shortest paths can be found at this step. In the case of deterministic routing, a single path is selected from alternative paths based on a certain policy, such as *random*[9]. Such a path selection algorithm is commonly required to implement a deterministic routing when the routing algorithm provides alternative paths.

Theorem *The routing designed with the above procedure guarantees deadlock-freedom.*

Proof *The routing guarantees deadlock-freedom, because no dependency occurs as follows.*

1. *No cyclic dependency is formed in each tier, because a packet must follow the restrictions for deadlock-freedom, as long as the packet is transferred on the single tier.*
2. *No cyclic dependency is formed across tiers, because a packet is passed between tiers only in the descending order.*
3. *No cyclic dependency is formed within a pillar, because a pillar router is a crossbar switch.* ■

This idea to guarantee deadlock-freedom and connectivity between all cores is similar to that of Descending Layer routing[10] for system area networks.

The symmetric XNoTs topologies, in which all tiers employ the same planar topology, would be selected in order to simplify the 3-D IC design. Such a symmetric structure of XNoTs can further simplify the routing algorithm, as follows: since every NoT can provide shortest paths between

any pair of cores, each pillar router selects a path by indicating a tier to be used, based on a certain policy. The path selection algorithm is also used for this selection.

A routing design is also extended in the special case where only a part of source-destination pair requires in-order packet delivery in an XNoTs topology. In such a case, adaptive routing and deterministic routing for different tiers can be used in the XNoTs. For example, only the bottom NoT employs DOR as a deterministic routing which can guarantee in-order packet delivery, whereas the other NoTs use Duato’s protocol as an adaptive routing for higher channel utilization. The decision about which NoT to use for transferring a packet is made by pillar routers in this case.

3.3 Layout of Pillar Routers

The above discussion introduced XNoTs from the viewpoint of topology. In this section, we discuss how to place pillar routers on them.

We propose to integrate a pillar router and all network interfaces (NIs) into a same pillar. That is, a single network interface is shared by all vertically arranged cores in an XNoTs topology as shown in Figure 7(b), unlike typical direct networks such as 3D-mesh, in which every node has its own network interface that connects its local core and router as shown in Figure 7(a). Such network interfaces are used as many as pillar routers in an XNoTs topology, while each of them requires $2n$ -channel, where n is the number of tiers stacked in the XNoTs.

Since a crossbar switch is formed across tiers in a pillar, inter-tier links rapidly increase as tiers increase, and considerable area would be required for them. However, symmetric XNoTs topologies can further simplify such a crossbar switch and reduce its inter-tier connections, because the network interface does not forward packets from a router to the other routers in symmetric cases, as mentioned previously.

Based on a detailed design of routers and network interfaces in a $0.18\mu\text{m}$ CMOS, we estimated the total network logic area including inter-tier vias (see Section 4.5).

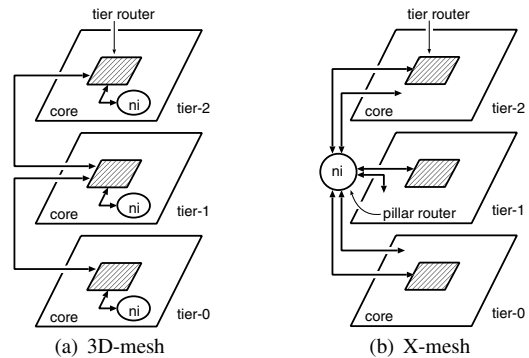


Figure 7. Single pillar in 3D-mesh and X-mesh

4 Evaluations

As typical XNoTs topologies, we evaluated X-mesh, X-torus, X-ft141, X-ft241, and X-ft441 in terms of ideal throughput, average hop count, simulated throughput, component count, network logic area, and energy consumption. The pros and cons of these topologies are discussed at the end of this section.

4.1 Ideal Throughput

The ideal throughput of a network is the data acceptance rate that would result from perfectly balanced routing and flow control with no idle cycles; it is calculated as [5]

$$\Theta_{ideal} \leq \frac{2bB_c}{N} \quad (1)$$

where N is the number of cores, b is the channel bandwidth, and B_c is the channel bisection of the network.

Array sizes for horizontal and vertical directions are not always the same (e.g., a 3-D IC that has two tiers, each of which consists of a 4×4 mesh). Such a network sometimes offers different bandwidths for the horizontal direction and the vertical one. Here, the bisection bandwidth is naturally extended into three dimensions; that is, the horizontal channel bisection, B_{ch} , is defined as the minimum channel count over all vertical bisections of a network. Similarly, the vertical channel bisection, B_{cv} , is defined as well. Even if a network offers different B_{ch} and B_{cv} , its channel bisection is limited by the smaller one; thus $B_c = \min(B_{ch}, B_{cv})$.

Table 1. Channel bisection B_c ($N = 2^i \times 2^i$)

	N -core $\times n$	16-core $\times 1$	16-core $\times 4$
X-ft141	$\min(4n, nN)$	4 (4, -)	16 (16, 64)
X-ft241	$\min(2^{i+1}n, nN)$	8 (8, -)	32 (32, 64)
X-ft441	$\min(4^i n, nN)$	16 (16, -)	64 (64, 64)
X-mesh	$\min(2^{i+1}n, nN)$	8 (8, -)	32 (32, 64)
X-torus	$\min(2^{i+2}n, nN)$	16 (16, -)	64 (64, 64)
3D-mesh	$\min(2^{i+1}n, 2N)$	8 (8, -)	32 (32, 32)
3D-torus	$\min(2^{i+2}n, 4N)$	16 (16, -)	64 (64, 64)

Table 1 lists the B_c of 3-D topologies. It also shows their B_{ch} and B_{cv} in parentheses. Note that B_{cv} is not available in the single tier case. The channel bisections of X-ft441, X-torus, and 3D-torus are the same; thus they offer equivalent ideal throughput. Similarly, those of X-ft241, X-mesh, and 3D-mesh are comparable. We confirmed their throughputs by using a flit-level network simulator (see Section 4.3).

Some topologies have unbalanced B_{ch} and B_{cv} . For example, X-ft141 has a larger B_{cv} than B_{ch} ; thus it cannot exploit the vertical bandwidth due to the horizontal one in the case of uniform traffic, in which each source sends equally to each destination. However, if traffic pattern has locality,

the penalty could be compensated by the task mapping that tries to place a pair of tasks which frequently communicate each other on the same pillar. This is advantageous to the energy consumption, because required energy for traveling in the vertical direction is smaller than that for the horizontal one such as in core-to-core communications on a tier.

4.2 Average Hop Count

Average hop count affects communication latency. It also affects energy consumption, which is one of the most crucial factors in modern embedded devices. We define the average router hop count, H_{rt} , as the number of routers a packet passes through on average in the case of uniform traffic. Similarly, we define H_{ni} as the average number of network interfaces a packet passes through.

Table 2. Average router hop count H_{rt} (H_{ni} is shown in parentheses)

	routing	16-core $\times 1$	16-core $\times 4$
X-ft †	up*/down*	2.60 (2.00)	2.48 (1.95)
X-mesh	DOR	3.67 (2.00)	3.54 (1.95)
X-torus	DOR	3.13 (2.00)	3.03 (1.95)
3D-mesh	DOR	3.67 (2.00)	4.81 (2.00)
3D-torus	DOR	3.13 (2.00)	4.05 (2.00)

† X-ft refers X-ft141, X-ft241, and X-ft441.

Table 2 shows the H_{rt} of 3-D topologies. The H_{rt} of 3D-mesh and 3D-torus increased as the number of tiers increased, whereas the H_{rt} of XNoTs topologies slightly decreased. As for XNoTs topologies, if the source and destination cores are in the same pillar, the packet can reach the destination via a network interface; thus it does not go through any routers in such cases. This is why the H_{rt} of the XNoTs topologies was reduced when the number of tiers increased.

Table 2 shows H_{ni} of these topologies in parentheses. The H_{ni} of 3D-mesh and 3D-torus was always two, because each packet goes through the network interfaces of the source and destination cores. As for XNoTs topologies, on the other hand, a packet can reach the destination via a network interface when the source and destination cores are in the same pillar; otherwise it passes through two network interfaces as in the 3D-mesh and 3D-torus cases. Thus, the H_{ni} of the XNoTs topologies is less than two in the cases of multiple tiers.

As shown, the average hop counts of the XNoTs topologies are smaller than those of 3D-mesh and 3D-torus; thus we can expect that X-mesh and X-torus offer better energy efficiency than 3D-mesh and 3D-torus. We confirm this in Section 4.6.

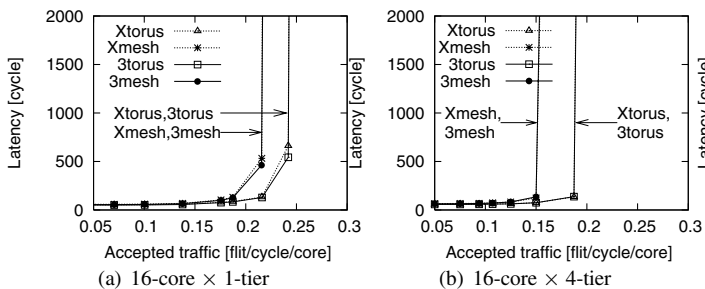


Figure 8. Performance of grid-based XNoTs

4.3 Simulated Throughput

A flit-level simulator written in C++ was used to confirm deadlock-freedom and measure the throughput on XNoTs topologies. A simple model of a wormhole router, which corresponds to the router used in the area evaluation (see Section 4.5), was used as a switching fabric in the simulator. A header flit requires at least three clock cycles to be transferred to the next router or core; one cycle for the routing computation, one cycle for allocating a virtual-channel and a crossbar, and the remaining cycle for transferring the flit to the next router or core. Wormhole switching was used as a switching technique on the router. The nodes inject packets independently of each other, and we set the packet length for 16-flit including one header flit. We used uniform synthesis traffic as a baseline.

Table 3. Routing algorithm in each topology

	algorithm	# of VCs	path selection
X-ft †	up*/down*	1	N/A
X-mesh	DOR	1	random
X-torus	DOR	2	random
3D-mesh	DOR	1	N/A
3D-torus	DOR	2	N/A

† X-ft refers X-ft141, X-ft241, and X-ft441.

The performance depends on the routing algorithm used in a topology. In this simulation, we selected dimension-order routing for 3D-mesh and 3D-torus. As for XNoTs topologies, dimension-order routing was also used in each tier in grid-based XNoTs, and up*/down* routing was used in each tier in tree-based ones, as listed in Table 3. These routing algorithms are popularly used in meshes, tori, and trees, respectively. Deterministic routing on XNoTs provides alternative shortest paths when multiple tiers are available and source and destination are not in the same pillar. As a simple path selection policy, we used *random*, which randomly selects a path from alternative paths.

First, we compare the grid-based XNoTs topologies with 3D-mesh and 3D-torus. Figure 8 shows the throughput (ac-

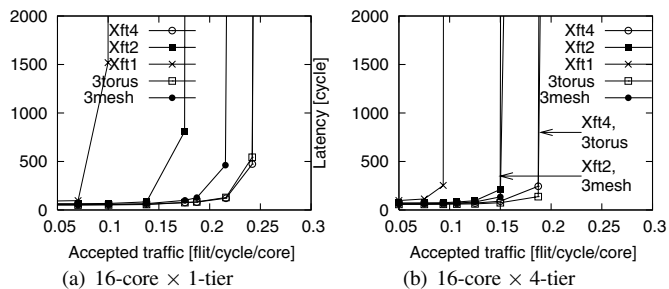


Figure 9. Performance of tree-based XNoTs

cepted traffic) versus the latency in the case of uniform traffic. As shown in the graphs, X-mesh and X-torus achieve equivalent performance to those of 3D-mesh and 3D-torus, respectively.

Figure 9 shows the results on tree-based XNoTs topologies. The performance of 3D-mesh and 3D-torus are also shown in the graphs, for comparison. As we expected, X-ft441 offers the highest performance among the tree-based ones, and it also achieves equivalent throughput to those of 3D-torus and X-torus. These results are consistent with the channel bisection of each topology analyzed in Section 4.1.

4.4 Component Count

The number of routers and network interfaces in a chip affects the network logic area and the implementation cost.

Table 4 lists the number of routers in 3-D topologies. The node degree of routers, which means the maximum number of channels a router has, is shown in parentheses. The numbers of routers required for 3D-mesh and X-mesh are the same, but the size of each router in X-mesh is smaller than that in 3D-mesh, because the node degree of X-mesh is five while that in 3D-mesh is seven. The same can be said for 3D-torus and X-torus. As for tree-based XNoTs, they require less than or equal to half the routers that grid-based topologies such as meshes and tori need.

Table 4 does not consider network interfaces (or pillar routers in XNoTs), which require considerable area in the case of XNoTs topologies. Number of network interfaces and their node degree are shown in Table 5. As shown, the number of network interfaces in an XNoTs topology is the same as that of pillars, while each network interface requires $2n$ channels, where n is the number of tiers stacked in the XNoTs topology.

From the above discussion, we can expect that the network logic area of an XNoTs topology will be smaller than (or at least equal to) that of the corresponding 3-D topology such as 3D-mesh or 3D-torus, unless the network interfaces for XNoTs take up a considerable area. In addition, we need to calculate the area overhead of vertical-links. In

Table 4. Number of routers ($N = 2^i \times 2^i$; node degree is shown in parentheses)

	$N\text{-core} \times n$	16-core \times 1	16-core \times 4
X-ft141	$n(4^i - 1)/3$	5 (5)	20 (5)
X-ft241	$n(4^i - 2^i)/2$	6 (6)	24 (6)
X-ft441	$n(4^{i-1}i)$	8 (8)	32 (8)
X-mesh	nN	16 (5)	64 (5)
X-torus	nN	16 (5)	64 (5)
3D-mesh	nN	16 (7)	64 (7)
3D-torus	nN	16 (7)	64 (7)

Table 5. Number of network interfaces (node degree is shown in parentheses)

	$N\text{-core} \times n$	16-core \times 1	16-core \times 4
XNoTs	N	16 (2)	16 (8)
3D-mesh	nN	16 (2)	64 (2)
3D-torus	nN	16 (2)	64 (2)

the next section, we evaluate them by using a detailed design of routers and network interfaces.

4.5 Network Logic Area

The network logic area in a 3-D NoC is composed of routers, network interfaces, and vertical links. To obtain the network logic area for each topology, we firstly estimated the area used in routers and network interfaces, and then calculated the area for vertical links.

To estimate the size of routers and network interfaces in a topology, we implemented a wormhole router that supports various node degrees. We also developed an NoC generator that automatically connects the routers and network interfaces in the arbitrary network topologies. Using the Synopsys Design Compiler, we synthesized the generated NoC design with a TSMC 0.18 μm standard cell library and estimated the network logic area. The behavior of the synthesized NoC routers was confirmed through a gate-level simulation assuming an operating frequency of 250MHz.

The router architecture was fully pipelined, and it transferred a header flit through four pipeline stages that consisted of a routing computation, virtual-channel allocation, crossbar allocation, and crossbar traversal. The flit-width was set to 32-bit, and each pipeline stage had a buffer for storing one flit. The routing decisions were stored in the header flit prior to packet injection (i.e., source routing); thus routing tables that require register files for storing routing paths were not needed in each router, resulting a low cost router implementation.

A network interface in XNoTs topologies was also implemented as a $2n$ -channel router, where n is the number of tiers. As for 3D-mesh and 3D-torus, on the other hand,

we implemented a simple network interface that employs a 2-flit FIFO buffer for both the core-to-network and network-to-core interfaces, for a fair comparison.

The XNoTs topologies evaluated in this section are all symmetric XNoTs, in which every tier can provide the shortest paths. The pitches of a through-wafer via can range from 1 μm to 10 μm square[6, 7, 12], depending on the manufacturing process (e.g., accuracy of wafer-to-wafer alignment). In this evaluation, the size of a through-wafer via was set to 5 μm square, and the flit-width was set to 32-bit. Then, we calculated the through-via area according to the number of all unidirectional 1-bit links between tiers.

Figure 10(a) and 11(a) show the total network logic area of 3-D topologies. In the graphs, areas for routers, network interfaces, and through-vias are identified by different colors. The ratio of the through-via area to the total is shown in parentheses. In the case of 4-tier (11(a)), XNoTs topologies require a larger network interface area compared with 3D-mesh and 3D-torus, because their network interfaces were implemented as routers connecting all tiers. On the other hand, X-mesh and X-torus require a smaller router area than 3D-mesh and 3D-torus, since their node degree is smaller than 3D-mesh and 3D-torus (see Table 4).

Although XNoTs topologies need more through-vias than 3D-mesh and 3D-torus do, the ratio of through-via area is several percent when the via pitch is 5 μm square, as shown in the graphs. Notice that the through-via area would be further reduced when the pitch becomes 1 μm square. The area overhead of vertical links in XNoTs can be compensated by their router area reduction. Actually, in the 4-tier case (Figure 11(a)), X-torus requires 12.3% smaller total area than 3D-torus, and X-mesh's total area is 3.4% smaller than that of 3D-mesh.

4.6 Energy Consumption

The average energy consumed to transmit a single flit from source to destination can be estimated as [16]

$$E_{flit} = wH(E_{sw} + E_{link}) \quad (2)$$

where w is the flit-width, H is the average total hop-count (i.e., $H = H_{rt} + H_{ni}$), E_{sw} is the average energy to switch a 1-bit data inside a router (or a network interface), and E_{link} is the 1-bit energy consumed in a link.

We used the Synopsys Power Compiler to extract E_{sw} of the router synthesized with the 0.18 μm standard cell library. The switching activity of the running router was captured through the gate-level simulation of the synthesized router. The gate-level power analysis based on this switching activity shows that E_{sw} is 1.13pJ when the router is operating at 250MHz with a 1.8V supply voltage.

E_{link} can be calculated as

$$E_{link} = dV^2C_{wire}/2 \quad (3)$$

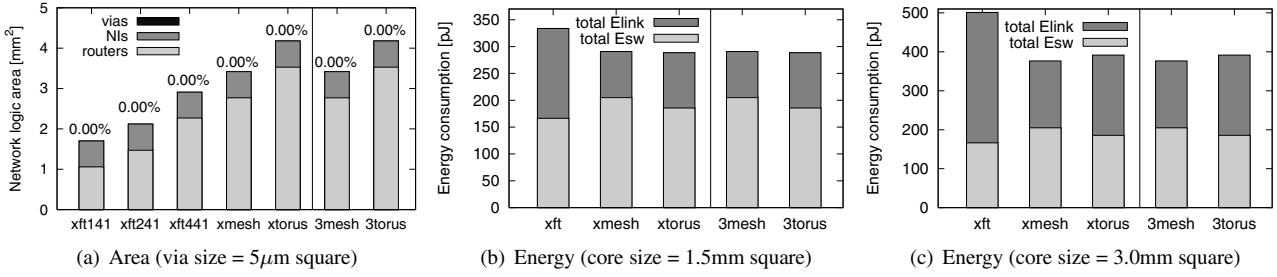


Figure 10. Network logic area and energy consumption to transmit a single flit (16-core \times 1-tier)

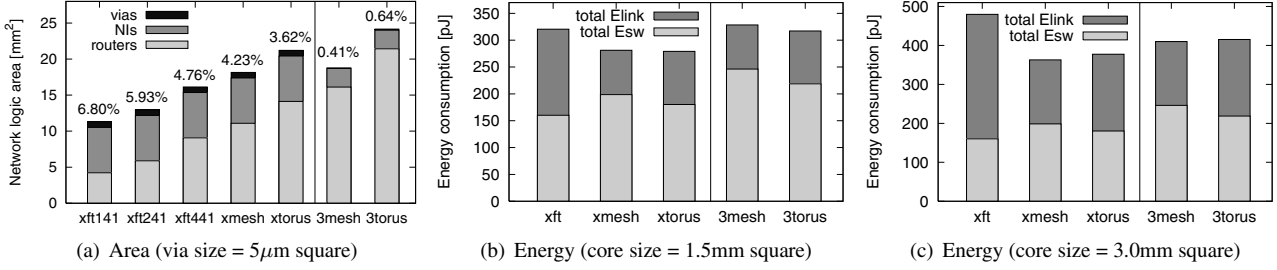


Figure 11. Network logic area and energy consumption to transmit a single flit (16-core \times 4-tier)

where d is the average 1-hop distance (in millimeters), V is the supply voltage, and C_{wire} is the wire capacitance per millimeter. C_{wire} can be estimated using the method proposed in [8], and is $414\text{fF}/\text{mm}$ in the case of a semi-global interconnect in the $0.18\mu\text{m}$ CMOS technology. For instance, E_{link} is 0.67pJ when the 1-hop distance is 1mm on average. The average 1-hop distance, d , depends on the distance between neighboring cores (i.e., granularity of core). We assumed two cases: the fine-grain case (core size = 1.5mm square) and the coarse-grain case (core size = 3.0mm square). In addition to E_{link} for the horizontal wires, we calculated E_{link} for the vertical links, assuming that the capacitance of an inter-tier via is 4.34fF [7], which is approximately the capacitance of a $10\mu\text{m}$ wire.

We derived E_{flit} based on Equation 2 with the various parameters mentioned above. Figure 10(b) and 11(b) shows E_{flit} of each topology in the case of fine-grain cores. Total energies consumed in routers (total E_{sw}) and links (total E_{link}) are identified by different colors. Although d of 3D-torus is longer than that of 3D-mesh due to the folded layout, its average hop count, which affects both total E_{sw} and total E_{link} , is fewer than that of mesh; thus 3D-torus requires a slightly smaller E_{flit} compared with 3D-mesh in the case of four tiers (Figure 11(b)). As for X-mesh, its total E_{link} is almost the same as that of 3D-mesh, but its average hop count is smaller than that of 3D-mesh; thus X-mesh requires 14.3% smaller E_{flit} than 3D-mesh, as shown in Figure 11(b). Similarly, X-torus consumes 12.0% less total energy than 3D-torus does.

Figure 10(c) and 11(c) shows the results for coarse-grain cores. As the core size enlarges, the impact of E_{link} increases; therefore 3D-torus, which has long wires, consumes slightly more energy compared with 3D-mesh. As well as the fine-grain case mentioned above, X-mesh and X-torus require less energy than 3D-mesh and 3D-torus.

4.7 Discussion

We discuss the pros and cons of the XNoTs topologies we created, in terms of channel bisection, average hop count, network logic area, and energy consumption in the case of 16 cores \times 4 tiers. For simplicity, we use channel bisection as a performance metric, because the actual throughput is highly dependent on the environments, such as I/O performance, traffic pattern, and routing algorithm.

Tree-based XNoTs topologies are advantageous in terms of performance per cost. Actually, the required silicon budget for X-ft441 is smaller than that of 3D-torus while it achieves the torus-level performance as we confirmed in Section 4.3. On the other hand, their downside is energy consumption when their core is enlarged. Although they consume energy as much as 3D-mesh and torus in the fine-grain case, their energy consumption is increased as their core becomes large, because much energy is consumed in their long wires around the root of tree. As for grid-based XNoTs, X-torus achieves good performance per cost and performance per energy compared with 3D-torus. Although X-mesh offers almost the same performance per cost as 3D-

mesh, it also has an advantage in energy consumption.

Many other XNoTs topologies can be created by combining various planar topologies such as meshes, tori, rings, and/or trees. For such topologies, their characteristics such as performance, cost, and energy consumption could be estimated based on the evaluation results provided here.

Although network topologies should be carefully selected so as to meet the requirements of embedded applications, the topological exploration for 3-D NoCs is still an emerging research topic and there is currently only a narrow range of choices, except for 3D-mesh and vertical buses. Therefore, the XNoTs topologies proposed here would be attractive alternatives to 3D-mesh when they meet the requirements of the target application.

5 Conclusions

As a new class of network topologies for 3-D NoCs, we proposed Xbar-connected Network-on-Tiers (XNoTs), which consist of multiple network layers tightly connected via crossbar switches. The planar topology on every layer can be independently customized so as to meet the cost-performance requirements, as far as network connectivity is at least guaranteed with the bottom layer. As typical forms of XNoTs topologies, we created X-mesh, X-torus, X-ft141, X-ft241, and X-ft441 and evaluated them in terms of performance, cost, and energy consumption. The evaluation results show that 1) X-torus achieves good performance per cost and performance per energy compared with 3D-torus; 2) X-mesh offers the equivalent performance per cost of 3D-mesh; 3) X-ft441 also outperforms 3D-torus in terms of performance per cost, but it consumes more energy when the core size becomes large. Thus, we confirmed that XNoTs topologies achieve at least as high throughput as existing 3-D topologies for equivalent chip sizes, even though XNoTs provide the flexibilities mentioned above.

Although only a small number of tiers is currently considered as feasible for the 3-D IC integration due to the heat-dissipation and yield limitations, we are planning to develop an efficient inter-tier network interface architecture, which can keep vertical links reasonable even when the number of tiers grows. One idea is to impose routing restrictions that prohibit a part of the packet transfer between cores and tier routers in order to simplify the crossbar switch of the network interface. Another idea is to partition the single crossbar into smaller crossbars. We will carefully consider the both possibilities as a future work.

References

- [1] C. Addo-Quaye. Thermal-Aware Mapping and Placement for 3-D NoC Designs. In *Proceedings of the International System-on-Chip Conference*, pages 25–28, Sept. 2005.
- [2] B. Black, D. W. Nelson, C. Webb, and N. Samra. 3D Processing Technology and Its Impact on iA32 Microprocessors. In *Proceedings of the International Conference on Computer Design*, pages 316–318, Oct. 2004.
- [3] D. Burger and et. al. Scaling to the End of Silicon with EDGE Architectures. *IEEE Computer*, 37(7):44–55, July 2004.
- [4] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proceedings of the Design Automation Conference*, pages 684–689, June 2001.
- [5] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
- [6] S. Das, A. Fan, K.-N. Chen, C. S. Tan, N. Checka, and R. Reif. Technology, Performance, and Computer-Aided Design of Three-Dimensional Integrated Circuits. In *Proceedings of the International Symposium on Physical Design*, pages 108–115, Apr. 2004.
- [7] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3D ICs: The Pros and Cons of Going Vertical. *IEEE Design and Test of Computers*, 22(6):498–510, Nov. 2005.
- [8] R. Ho, K. W. Mai, and M. A. Horowitz. The Future of Wires. *Proceedings of the IEEE*, 89(4):490–504, Apr. 2001.
- [9] M. Koibuchi, A. Jouraku, and H. Amano. Path Selection Algorithm: The Strategy for Designing Deterministic Routing from Alternative Paths. *PARALLEL COMPUTING*, 31(1):117–130, Jan 2005.
- [10] M. Koibuchi, A. Jouraku, K. Watanabe, and H. Amano. Descending Layers Routing: A Deadlock-Free Deterministic Routing using Virtual Channels in System Area Networks with Irregular Topologies. In *Proceedings of the International Conference on Parallel Processing*, pages 527–536, Oct. 2003.
- [11] C. E. Leiserson. Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, Oct. 1985.
- [12] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. In *Proceedings of the International Symposium on Computer Architecture*, pages 130–141, June 2006.
- [13] T. H. Szymanski. $O(\log N / \log \log N)$ Randomized Routing in Degree- $\log N$ "Hypermeshes". In *Proceedings of the International Conference on Parallel Processing*, pages 443–450, Aug. 1991.
- [14] N. Tanabe, S. Nakamura, T. Suzuoka, and S. Oyanagi. Base-n-Cube: High Performance Interconnection Networks for Highly Parallel Computer PRODIGY. In *Proceedings of the International Conference on Parallel Processing*, pages 509–516, Aug. 1991.
- [15] M. B. Taylor and et. al. The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs. *IEEE Micro*, 22(2):25–35, Apr. 2002.
- [16] H. Wang, L.-S. Peh, and S. Malik. A Technology-Aware and Energy-Oriented Topology Exploration for On-Chip Networks. In *Proceedings of the Design, Automation and Test in Europe Conference*, pages 1238–1243, Mar. 2005.