A Low-Radix and Low-Diameter 3D Interconnection Network Design *

Yi Xu†, Yu Du‡, Bo Zhao†, Xiuyi Zhou†, Youtao Zhang‡, Jun Yang†
† Dept. of Electrical and Computer Engineering
‡ Dept. of Computer Science
University of Pittsburgh, Pittsburgh, PA 15621
†{yix13, boz6, xiz44, juy9}@pitt.edu, ‡{fisherdu,zhangyt}@cs.pitt.edu

Abstract

Interconnection plays an important role in performance and power of CMP designs using deep sub-micron technology. The network-on-chip (NoCs) has been proposed as a scalable and high-bandwidth fabric for interconnect design. The advent of the 3D technology has provided further opportunity to reduce on-chip communication delay. However, the design of the 3D NoC topologies has important distinctions from 2D NoCs or off-chip interconnection networks. First, current 3D stacking technology allows only vertical inter-layer links. Hence, there cannot be direct connections between arbitrary nodes in different layers — the vertical connection topology are essentially fixed. Second, the 3D NoC is highly constrained by the complexity and power of routers and links. Hence, low-radix routers are preferred over high-radix routers for lower power and better heat dissipation. This implies long network latency due to high hop counts in network paths.

In this paper, we design a low-diameter 3D network using low-radix routers. Our topology leverages long wires to connect remote intra-layer nodes. We take advantage of the start-of-the-art one-hop vertical communication design and utilize lateral long wires to shorten network paths. Effectively, we implement a small-to-medium sized clique network in different layers of a 3D chip. The resulting topology generates a diameter of 3-hop only network, using routers of the same radix as 3D mesh routers. The proposed network shows up to 29% of network latency reduction, up to 10% throughput improvement, and up to 24% energy reduction, when compared to a 3D mesh network.

1. Introduction

The technology driven integration of many cores into a single chip is facing critical challenges such as high power dissipation, resource management etc. In particular, the interconnection network starts to play a more and more important role in determining the performance and power of the entire chip [24]. In packet-switched on-chip network, which is the predominant network-on-chip (NoC) for tiled multicore processors, communication among cores at distance experiences long latencies because packets need to compete for resources on a hop-by-hop basis. To provide low latency and high bandwidth communication in NoCs, many researches have been carried to optimize the network in various approaches such as developing fast routers [19, 20, 23, 28] and designing new network topologies [9, 18, 30].

The emerging three-dimensional (3D) stacking technology has provided a new horizon for NoC designs. 3D stacking is a technology that stacks multiple active silicon layers on top of each other, and connect them through wafer bonding. It reduces interconnect delay via much shorter vertical wires between the dies. It is estimated that 3D architectures reduce wiring length by a factor of the square root of the number of layers used [16]. The main benefits of 3D stacking over traditional 2D designs are higher performance and lower interconnect power due to reduced wire length. Such an advantage enables higher transistor packing density, which is particularly suitable for chip multiprocessor (CMP) designs, and has created unique opportunities and challenges to design low latency, low power and high bandwidth interconnection network in 3D. Lately, there has been an increasing interest in interconnection network designs for 3D stacked CMPs [21, 25, 31]. Since the major difference between 3D and 2D NoC is the presence of vertical links that connect different layers, existing researches have focused mainly on optimizing the vertical communication including choosing good vertical link architecture [25], designing more efficient routers to reduce vertical hop count [21], and reducing power consumption of routers via a multi-layered 3D technology [31]. However, despite the previous efforts in minimizing the vertical communication time, the 3D NoC of a CMP still incurs long network latency mainly due to the the large network *diameter*, much like in a 2D NoC.

The purpose of this paper is to develop a 3D NoC topology of low latency. This is achieved through a low-diameter network using long wires and low-overhead routers. There have been a myriad of research decades ago on off-chip interconnection network topology designs, especially in the parallel processing community. However, there are essen-

*This work is supported in part by NSF 0747242, 0641177, 0720595, 0734339 and Intel.

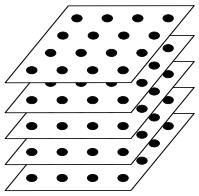


Figure 1. A sample 4×4×5 3D chip architecture. Each node represents either a core or a cache memory bank.

tial differences between on-chip interconnection networks and their off-chip counterparts. The distinctions between this paper and the past efforts mainly come from the physical constraints in 3D chips. Specifically,

- Nodes in 3D interconnect represent either cores or cache banks which are typically tiled in 2D and stacked in 3D regularly, as depicted in Figure 1. The links between different layers, at current fabrication stage, are only vertical links connecting nodes directly atop or below. It is not feasible to directly connect nodes on different layers with an angle. This restriction eliminates a large portion of topologies containing angled links such as trees. In fact, the only freedom in placing links lies within each layer. Therefore, it appears that our problem can be reduced to using a low-diameter 2D topology, which has also been extensively studied in the past. However, this brings another distinction between this paper and the prior art.
- A low diameter 2D topology such as the flattened butterfly [18], or any other topology that can be flattened, entails high-radix routers. For example, a fully connected 2D network (diameter=1) for each layer in Figure 1 requires radix-18 routers (15 level + 2 vertical + 1 local port). Using flattened butterfly of diameter-2 requires radix-9 routers (without concentration). Lower radix routers, such as those in Express Virtual Channel [23], varietal tori or hypercubes, will continue to increase the diameter of the network. Hence, to keep the diameter in each layer very low, e.g., 1 or 2, one must incorporate high-radix routers. Unfortunately, high-radix routers accompanied with long wires imposes great concerns in their area and power overhead in both routers and wires, which are the first-order constraints in a 3D stacked chip. Simply replicating those 2D designs in the layers of our 3D network would generate prohibitive area overhead and power surge.
- Finally, even though we have freedom in placing links in each layer, long wires occupy 4× to 8× the width of short wires [8]. Therefore, we may not be able to place all desired long wires in one metal layer. This is a critical restriction in designing our topology. Therefore, we need to seek an alternative that does not compromise the network diameter.

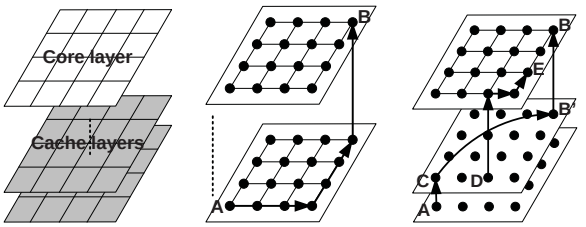
In this paper, we develop a methodology for designing a *low-diameter* 3D NoC using *low-radix routers* to achieve low network latency. Our design is suitable for a prevalent 3D CMP architecture where all cores are placed in the layer closest to the heat sink (for best heat dissipation), and the cache memories are stacked in the remaining layers [4, 17, 26]. Our topology adopts the one-hop router design in vertical vias [21], and replaces the level 2D mesh with a network of long links connecting nodes that are at least m mesh-hops away, where m is a design parameter. The mesh for the core layer is preserved for short distance communication less than m hops. In such a topology, communication that requires more than m horizontal hops will leverage the long physical wire and vertical links to reach destination, achieving low total hop count. Long-range links have been used on-chip for improving the performance of critical paths [29]. Long links have also been inserted into an application-specific 2D mesh to reduce its average packet hop count [30]. Although the main challenges in using long links are 1) they may limit the clock frequency of the network; and 2) they may consume higher power than shorter links, we demonstrate through our experiments that we still obtain positive gains.

To use low-radix routers while incorporating long wires as many as possible, we leverage the great advantage of 3D stacking to distribute long wires onto different layers, reducing the radix pressure on each router. Intra-layer long distance communication may utilize the point-to-point long wire at a different layer via vertical hops. We develop a mechanism to automatically generate such a network topology using Integer Linear Programming. Using this method, we can distribute a fully connected 4×4 2D network onto 5 layers forming a diameter-3 3D network using routers with same degree as a 3D mesh router. That is, any point-to-point communication requires at most 3 hops. We also present methods to scale our design with different layer and core numbers. Our experimental results on synthetic traffic, SPLASH2, OpenMP, and SpecJbb 2005 benchmark traces show up to 29% reduction in zero-load packet latency, reducing the network energy by up to 24%, as compared to the 3D mesh network.

The remainder of this paper is organized as follows. Section 2 gives an illustrative example of our proposed topology and its advantages over a conventional 3D mesh. Section 3 discusses the details of our design methodology. Section 4 explains the wire models. Section 5 shows the experimental results. Section 6 explains the scalability of our proposed network. Section 7 discusses related works. Finally, Section 8 concludes this paper.

2. An Illustrative Example

In this section, we highlight the advantages of our proposed topology over a conventional 3D mesh network using an illustrative example. First, we introduce the 3D CMP architecture that our network is developed upon. A 3D CMP can be built in a number of ways. The first design is to place cores and caches in alternating locations, both horizontally and vertically forming a staggered layout [1, 25]. This design avoids direct contact between active cores and interleaves cool cache banks evenly with hot active cores. The second scheme redesigns the entire core and



(a) 3D chip architecture (b) Routing path in a 3D mesh network (c) Routing path in our proposed network

Figure 2. A motivating example.

other logic into a 3D circuit to span them across all layers of the chip [4, 31, 32]. This design reduces the wire latency within logic to improve performance. The third design places cores in one layer, closest to the heat sink, and cache memories in all remaining layers [4, 17, 26], as illustrated in Figure 2(a). This design is the best in terms of heat dissipation and scalability in number of layers. Our network topology will be developed based on this layout, but the principle is equally applicable to other architectures.

The mostly adopted interconnection network in 3D chip is a regular 3D mesh network [21, 25, 31], as shown in Figure 2 (b). Here we omit some layers and the vertical links for clarity. Each node represents a router associated with a core or a cache bank. In such a network, packets are typically routed using dimension-order routing (DOR) such as x-y-z. For example, when node A sends a packet to node B in Figure 2(b), the routing path will follow the arrows shown in the graph, traversing through 8 routers, or in 7 hops. Every hop involves certain delay in the router, creating long latencies between point-to-point communication. Previous contributions have optimized the vertical link and router architecture such that data transfer between any two layers can complete in a single hop [21, 25]. Therefore, to further reduce the network latency, optimizations must be carried in each layer.

Our proposed topology aims to place long links between remote nodes in a layer. For example, in Figure 2(c), when node C sends a packet to B, it can directly go through the long link CB', and then B'B, taking 2 hops altogether which is a significant reduction from 7. We will call CB' a 6-hop link since C and B' are 6 hops away in the original mesh with DOR. However, due to the wire area and node degree restrictions (a node cannot take too many long links), similar long links may not exist in all layers. For example, when node A communicates with B, the similar 6-hop link may not be available in A's layer. Hence, A will borrow link CB' via a vertical hop, leading to a 3-hop path to B. This is still shorter than the 7-hop path in a mesh. In an extreme case where an m-hop ($m > 1$) link cannot be found in any cache layers, we direct the request to the core layer using the mesh that is preserved for core-to-core communication. This is illustrated in Figure 2(c) for traffic from D to E. It is easy to see that when $m \leq 2$, the routing distance in the long-link topology is identical to that in a 3D mesh.

3 Design of the Proposed Topology

Knowing the advantages of long links, we now elaborate the design of a network for a 3D architecture using long wires. First of all, the core layer has many intra-layer performance critical traffic such as the cache coherence messages. We preserve the mesh for the core layer because the one hop traffic are most efficiently handled here. Taking them down to the cache layers would triple their hop count and introduce more traffic in the entire network. Our focus is to design a mechanism to connect the routers in the cache layers with long wires. Adding long wires in the cache layers reduces the non-uniformity of the shared cache accesses, which implies that we can avoid complex cache management techniques such as data migration since all cache banks are of approximately the same distance to the requesting core. Therefore, we only need to consider the inter-layer traffic between the cores and the cache nodes.

3.1 The Rationale

The ideal diameter of such a 3D network is 2, meaning that every pair of core-cache nodes are at most 2 hops away: 1 horizontal plus 1 vertical hop. This would require the intra-layer routing distance between every pair of nodes be 1, indicating a clique (fully connected) network per layer. This is clearly too expensive. Therefore, we relax the network diameter to 3, allowing one more hop between any pair of core-cache nodes to exist either horizontally or vertically. If this hop is horizontal, then the routing distance between any intra-layer pair of nodes is ≤ 2. This is still very expensive in total link count and router ports required per layer. For example, the 4×4 flattened butterfly topology has a diameter 2, but requires routers of radix 9 which is quite high especially in a 3D chip. Hence, we are left with the choice of letting the extra hop be a vertical one. This implies that the only horizontal hop in a 3-hop path is still a long jump. However, it does not have to exist in every layer. As long as it is in some layer, we can always use a vertical hop to reach that layer, and then finish the route in 2 hops. An intuitive example was given in Figure 2(c) for the path between node A and B. This seems a feasible approach as it does not require a long link to exist in every layer.

To achieve a true network diameter of 3, we need to ensure that every pair of core-cache nodes are within 3 hops away, i.e., 2 vertical hops and 1 horizontal hop at most. That is, for every node pair (i, j) in a layer, there must be a link between them either in this layer, or in a different layer with end nodes that are vertically aligned with i and j. This means that we are implementing a clique, but the links are in different layers. *Essentially, we are slicing a clique onto different layers with the top layer being a mesh, and each of the remaining layers being a subgraph with smaller node degrees and fewer links*. Such a topology will be practical for implementation in 3D. Note that we may not need intra-layer 2-hop links as their 3D path length is 3, which would not generate any hop reduction. Therefore, the subgraphs can be even thinner without the 2-hop links. However, we will leave this as an option, as having them in the subgraphs can help to distribute the network traffic more evenly.

The reason we are able to achieve a low-diameter and

low-radix topology lies in the great advantage of the vertical links, i.e., pillars that connect nodes on different dies. We are taking advantage of the single hop in vertical direction between any layers, but the connection among the vertical routers does not form a clique, leveraging the recent contributions in 3D router designs [21, 31]. Every router uses the same number of vertical ports as in a 3D mesh, but inter-layer links can be connected through a "connection box" of only a few transistors (see Figure 5) that can be dynamically turned on and off in every cycle to connect and disconnect inter-layer links. This enables a single hop between any layers using only one link between adjacent layers. Such a technology leverage the short distance in the vertical direction of a 3D chip, so it is difficult to implement for off-chip interconnection networks.

3.2 The Design Space

Our topology design is subject to the radix of the routers (or the number of links) allowed in each layer. In this section, we examine the design space for embedding a clique into a 3D topology. Let L be the number of cache layers, N be the node count per layer, and R be the router's port count for non-local horizontal links. Then every layer can host $RN/2$ links maximally, and the total number of horizontal links accumulating all cache layers is $LRN/2$. The total number of links in an N-node clique is $N(N-1)/2$. Removing the $2(N-\sqrt{N})$ mesh links, the total links we should distribute to the L layers are $N(N-1)/2 - 2(N-\sqrt{N})$. Therefore, to accommodate all the links, we must have:

$$\frac{LRN}{2} \geq \frac{N(N-1)}{2} - 2(N-\sqrt{N}) \quad (1)$$

or

$$LR \geq N + \frac{4}{\sqrt{N}} - 5 \quad (2)$$

Also, our topology should have the same number of links per layer as in a mesh network to keep the same network bandwidth. This gives:

$$\frac{RN}{2} = 2(N-\sqrt{N}) \quad or \quad R = 4 - \frac{4}{\sqrt{N}} \quad (3)$$

Hence, $3 \leq R \leq 4$, meaning that some routers use 4 ports and the rest use 3 ports. Combining equation (2) and (3), we can obtain a set of practical solutions to our design space, as listed in Table 1. The L column shows the number of layers

N	$LR \geq$	R	L
16	12	3	≥ 4
25	21	3.2	≥ 7
36	32	3.3	≥ 10

Table 1. Solutions space for a 3D N-node clique.

required to include all the necessary links. If the layer count is greater than the given number, links can be replicated in different layers, which could potentially offload some traffic from the single long link design with minimum layer count. The stacking depth is also limited by the current and future technology in 3D integration. The ITRS projected that by year 2011, the number of dies that can be stacked will reach 11 [37]. Hence, for $N \leq 36$, the L's fall within a practical region for implementation. If the layer count is smaller than the number given in Table 1, we can only incorporate a subset of the links, which will be discussed in the next section. When $N > 36$, the layer count or the ports per router required become too large. We will discuss the method for scaling the topology for larger networks in Section 6.

3.3 Subgraphing a Clique Using ILP

The design space analysis shows that in theory, our topology can be developed given the router port and link count constraints. In practice, there is another important constraint that must be considered for on-chip interconnection network design. This is the area overhead and wiring complexity of those long links. If the space for global wires is limited, we may not be able to place enough long links in the cache layer, and would have to push the traffic to the top mesh layer. This would reduce the attainable average hop reduction, and impose the concerns on imbalanced traffic distribution.

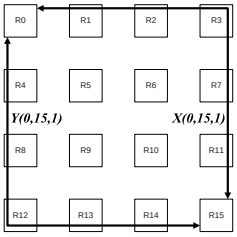


Figure 3. The Boolean variables for wire routing.

Therefore, our goal is to select the wires that satisfy the port, link, and area constraints while achieving the largest reduction in hop count. This selection process can be carried systematically through Integer Linear Programming (ILP) which is a powerful method for maximizing (minimizing) certain objectives through determining a set of decision variables, subject to some constraints. We will now discuss these three main ILP components: variables, objective function, and constraints.

3.4 Decision variables

Our decision variables are all boolean variables representing whether a wire connecting node i and j on layer k should be selected. However, we need to take into account how the wire will be routed on-chip as their layout will affect the area and wiring density around the routers. Since wires are laid out in horizontal and vertical directions only, we use two directional variables $X_{i,j,k}$ and $Y_{i,j,k}$ to indicate whether the wire is routed first in the x or y direction. For example, in Figure 3, the wire between node 0 and 15 on layer 1 is represented by $X_{0,15,1}$, indicating that the wire is first routed in horizontal direction, and $Y_{0,15,1}$, indicating that it is first routed vertically. As a wire can be only routed in one direction first, we have

$$X_{i,j,k} + Y_{i,j,k} \leq 1, \quad 0 \leq X_{i,j,k},\ Y_{i,j,k},$$
$$X_{i,j,k} = X_{j,i,k}, \quad Y_{i,j,k} = Y_{j,i,k} \quad (4)$$

3.5 Objective function

Our objective is to maximize the latency reduction in the network. Different wires have different latencies. As

we will discuss in Section 4, we will pipeline long wires if higher network frequency is required. We take this into consideration for calculating the latency. Let us first consider the hop count between a core node with coordinate $(x, y, 0)$, and a cache node with coordinate (u, v, w). Let i be this cache node, and j be another cache node on the same layer with coordinate (x, y, w). As we can see, in a 3D mesh, the hop count between $(x, y, 0)$ and (u, v, w), denoted as $H_{mesh}(i, j)$ is the Manhattan distance between i and j plus 1:

$$H_{mesh}(i, j) = Manhatten(i, j) + 1 \quad (5)$$

In our long wire topology, the hop count between $(x, y, 0)$ and (u, v, w) is 2 if the wire between i and j is in layer w, and 3 otherwise. We assume that the traffic is uniform random since our study is general purpose. Thus, there are $1/L$ (there are L layers of cache) chances that the hop count is 2. Hence, the hop count between $(x, y, 0)$ and (u, v, w) in our long wire topology is:

$$H_{long}(i, j) = 2 \times \frac{1}{L} + 3 \times \frac{L-1}{L} = 3 - \frac{1}{L} \quad (6)$$

The latency for packet transmission can be expressed as [23]: $L = D/v + L/b + H \times T_{router} + T_{contention}$, where D is average Manhattan distance, v is signal propagation velocity, L is packet size, b is channel bandwidth, H is hop count, T_{router} is router delay, and $T_{contention}$ is the delay due to network contention. We assume uncontended network to formulate our latency calculation (i.e. $T_{contention} = 0$). For example, the 6-hop long wire with pipelined design requires 3 cycles under 3GHz frequency (see Section 4). The value D is the total wire length of a path (1 level 6-hop wire and up to 2 vertical pillars). The vertical pillars are significantly shorter than the long wire, and thus can be neglected for simplicity. Thus, D/v only depends on the clock frequency of the long wire, e.g. 1ns for the 6-hop long wire and $\frac{1}{3}$ns for a 1-hop mesh link under 3GHz. The value b is 1 flit per cycle. The value H can be obtained from (5) for mesh and (6) for long link network. L and T_{router} are both constants. The packet latency can be then computed by summing up the above values. Let $L^m_{i,j}$ and $L^l_{i,j}$ be the packet latency in mesh network and long-link network respectively. The objective function can be expressed as:

$$MAX\left(\sum_{k=0}^{L-1}\sum_{i=0,i\neq j}^{N-1}\sum_{j=0}^{N-1}(X_{i,j,k} + Y_{i,j,k})(L^m_{i,j} - L^l_{i,j})\right) \quad (7)$$

3.6 Constraints

We first discuss the area constraints for wiring long links, and then summarize mathematically the rest constraints that we discussed earlier.

Wiring Area. The number of wires that can be routed in a interconnection network is limited by the size of the router as well as the area taken by the wires. The *wiring density* for interconnection network is referred as the maximum number of tile-to-tile wires routable across a tile edge [15]. Therefore, given the size of a router, the number of global wires that can be routed through the router is fixed. Studies show that global wires can consume 4 to 8 times the area (width+spacing) of short local wires [8]. The link in a regular mesh spanning one tile is a short wire. In our topology, we have wires spanning from 2 to $\sqrt{N} - 1$ hops. The 2 and 3-hop wires are considered as short and medium wires, and rest are considered as long wires.

Let $W_{i,j}$ denote the area taken by the wire from node i to j, A_{max} be the wiring density in unit of the area for one short wire. Let s denote a one hop segment between two neighboring routers, and S denote the union of them. Then the area constraint for wiring can be expressed as:

$$\sum_{passing\ thru.\ s}^{all\ wires} W_{i,j} \times (X_{i,j,k} + Y_{i,j,k}) \leq A_{max},\ \forall k, 0 < k < L, \forall s \in S \quad (8)$$

The sum can be expanded by examining the routing paths of all wires. The constant $W_{i,j}$ and A_{max} are determined as follows. If the Manhattan distance between i and j is larger than 3, $W_{i,j}$ is $4\times$ the area of a short wire. Otherwise it is $1\times$. For A_{max}, we define that the total pitch (width+spacing) of wires in one hop does not exceed the edge of a 3D mesh router. Note that the routers in our topology are no larger than that of a 3D mesh router because of the port constraint we defined. Therefore, our area overhead does not create pressure in routing the introduced long wires. In our experimental setting, the wire bandwidth is 130 bits (data + control). Every such bundle is doubled to support bidirectional communication. According to the ITRS prediction in 2006 [37], the semi-global wire width is 180nm. Previous studies on 3D mesh routers of 5 horizontal ports show that its area is $\sim 0.37mm^2$ [25]. Hence, we have $130\times2\times$x$\times$180nm$\leq\sqrt{0.37}mm$, where x indicates how much area, in multiple of the semi-global wire width, is allowable per 3D mesh router's edge. Therefore, $A_{max} = x \leq 12$. That is, we can arrange up to $12\times$ the area of a semi-global wire per hop. This can accommodate, for example, 3 long wires, or 2 long wires plus 4 short wires etc., depending on the global gain calculated from our objective function defined in equation (7).

Router Port. We have discussed earlier that we use only low-radix routers in our network. Hence, we define that the number of ports per router should not exceed the maximal ports per mesh router, denoted as P_{max}, which is typically 7 (4 intra-layer, 1 local, and 2 vertical) unless more pillars are used. Hence,

$$\sum_{j\neq i}^{0\leq j<N} (X_{i,j,k} + Y_{i,j,k}) \leq P_{max}, \forall i, 0 \leq i < N,\ \ \forall k, 0 \leq k < L \quad (9)$$

Total Links. We have discuss in equation (3) that we keep the total number of links in the same amount as in a mesh, which is $2(N - \sqrt{N})$, to provide the same network bandwidth. This can be defined mathematically as:

$$\sum_{i=0}^{N-1}\sum_{j\neq i}^{N-1}(X(i, j, k) + Y(i, j, k)) \leq 2(N - \sqrt{N}) \quad (10)$$

3.7 Summary and ILP efficiency

Putting everything together, we have defined the boolean variables of our ILP problem as $X(i, j, k)$ and $Y(i, j, k)$ to indicate the wires and their routing direction. The results of these variables are determined by evaluating the objective

function specified in equation (7) subject to the constraints defined in equation (4), (8), (9), and (10).

The constraints and the objective functions were formulated using the front-end AMPL language [10]. The system was solved using the state-of-the-art ILP solver *lpsolve* [3]. Due to the presence of large number of decision variables and complex constraints, directly solving the entire system of equations requires several days to get the final results on a 2.4GHz dual-core Intel Xeon workstation.

To improve the efficiency, we divided the decision variables into two groups – one for short wires, and the other for long wires. Since long wires are more effective in reducing hop counts, they are given higher priority. Therefore we first use ILP solver to map long wires. Considering the wire area constraints, the number of long wires between any single-hop node pair is no more than 3. Other constraints on link and port count remain unchanged. After obtaining the routing of long wires, we then apply the solver again to find the routing for short wires. With this two-phase optimization, we can generate result within 1 minute for a $4 \times 4 \times (4 \; or \; 5)$ topology. Note that even though our network size, objective function, variables and constraints are all fixed, changing the ILP solving procedure will result in different topologies. Our experimental results show that the difference in the produced topologies before and after the optimization does not have impact on the average network latency.

3.8 A sample topology generated

We now present a sample 3D topology generated using ILP. A $4 \times 4 \times 5$ (16 cores and 4 cache layers) 3D chip's network was solved. The resulting topologies for all cache layers are shown in Figure 4. In the figures, the bold lines stand for long wires that are thicker than short wires.

The constraints we used here are in line with the discussion in Section 3.2. Apart from the vertical ports and local port, each router is restricted to a maximum of 4 intra-layer ports. For each 16-node layer, the total number of wires allowed is 24, which is equal to that of a mesh. For a 16 (nodes)×4(layer) network, there are 96 links that are equal to or longer than 2 hops. The ILP solver was able to place all those links into the network under the port, link and area constraints.

As we can see from the resulting topologies, the wire densities in the center of the topology tend to be higher than those along the edges. However, all segments between neighboring routers are limited by A_{max} (12). We will show in our experiments later that our total network energy is less than that of a mesh because of the hop count we saved.

3.9 Routing Algorithm

We choose the deterministic routing algorithm in our topology as it produces the minimal hop count. Once the topology is generated, our routing algorithm is also determined: when a core with coordinate $(x, y, 0)$ generates a request to a cache bank (u, v, w) on the w^{th} layer, the router of the core checks whether the long link between (x, y) and (u, v) exists in any layer l. If so, the routing path is computed as $(x, y, 0) \rightarrow (x, y, l) \rightarrow (u, v, l) \rightarrow (u, v, w)$ with 3 hops. Note that if $l = w$, the route is completed in 2 hops. When the data bank responds to the core's request, the same path in a reversed order is used. If the long link between (x, y) and (u, v) does not exist. The request will follow the DOR algorithm such as XYZ or YXZ. The cache-to-core traffic will also follow the same path, but in a reversed order. Also, in addition to deterministic routing algorithm, we used 3 VC per port to avoid deadlocks in our network. Our current deterministic routing has generated encouraging improvement in network latency (see Section 5). We will use an adaptive routing in the future to balance the traffic load and further improve the performance.

3.10 Routing Table

We use routing tables to implement our routing algorithm. Each router with coordinate (x, y, z) has a lookup table that contains the information for the location of the long link from router $(x, y, *)$ to any router $(u, v, *)$, where $*$ represents any layer, and u, v represent coordinates other than x, y. This is because a packet arriving at a router will first search for the long link that reaches the destination. If it is in the current layer, the packet will be routed across. Otherwise, the packet will be first brought to the layer that has the long link, and then routed across. Therefore, the routing tables for a column of routers are identical. For our 4×4×4 network, there are 15 entries in each table. Each entry of the table (in a router (x, y, z)) contains the location of the long link from $(x, y, *)$ to $(u, v, *)$: layer ID and output port ID which require 2 bits for each. Hence, the routing table size for the 4×4×4 network is $4 \times 15 = 60$ bits.

3.11 Discussion

Although our design results in different interconnections in different layers, the maximum radix of routers of all layers and the total number of links in every layer are the same as in a mesh. The only difference among layers is the wiring. Fortunately, our proposed ILP methodology can help to automate the process of generating the topology and wire layout. Also, additional constraints can be added to satisfy manufacture and technology requirements. When the topology is modified, the only change to the network is the routing table contents. The routing algorithm remains the same.

4 Modeling Long Wires

In section 3.6, we discussed the handling of global wire area. In this section, we focus on its delay and energy consumption. Global wires are placed in the top metal layers, e.g. metal 7 or metal 8 layer. They are thicker and wider than local wires that are placed near the device layer. Because of their size, global wires take longer time and more energy to carry signals than local (short) wires. If our single-hop long link takes only one clock cycle to complete, the delay on the longest link in our topology will have an impact on the clock frequency of the network. For this reason, we will model the long wires with two different clock frequencies: one that can allow the longest link to transmit signals in one clock cycle, and another that requires pipelined wire designs for higher clock frequencies. Their impact on the overall network latency and energy consumption will be shown in the experimental section.

(a) first cache layer (b) second cache layer (c) third cache layer (d)forth cache layer

Figure 4. A 3D topology for a $4 \times 5 \times 4$ 3D chip (1 core layer and 3 cache layers) generated using ILP.

For all types of wires (long or short, pipelined or none), we use HSPICE with 45nm BSIM4 technology model from PTM [35]. The supply voltage Vdd is 0.8V and the operating temperature is assumed to be 70°C.

Single-cycle long wires. There has been an incessant effort in minimizing the delay of global wires, especially with the shrinkage of process dimension. Successful researches have been carried to lower the resistance of electrical wires and provide high speed (near velocity-of-light) and high bandwidth [5, 7, 14] global wires. New technologies such as optical [22] and radio frequency [6] signaling have also been proposed for on-chip communication. These innovations can be leveraged for designs utilizing global wires [18, 29, 30]. However, we will use the conventional global wires with repeaters to illustrate the applicability of our design even in the current technology.

We modeled the delay for both long and short wires. For long wires, we obtain a delay-optimized design through carefully inserting the wire repeaters. We first refer to ITRS 2007 to obtain the latest dimension of wires, e.g. 1.5mm per hop, which are then used to calculate the wire parasitics (capacitance and resistance) in the PTM interconnection model [36]. Using these parameters, we calculate distances between the inverters and the size of them for a delay-optimized wire of certain length. Since there should be even number of inverters on every wire, we round the delay-optimized repeater number up to the nearest even integer. Then we shrink the size of each inverter to reduce its power without affecting the delay. To model the wire power, we used the bus coupling model in [13]. To simulate the cross coupling effect between wires, we connected every adjacent wire pair with coupling capacitor using the PTM interconnect model [36]. We simulated the wires under the worst case scenario where every wire has an inverse voltage level change with its two neighbors.

The resulting delay and energy for wires of 1-6 hops are listed in the "sng" columns of Table 2. Note that the 1-hop wire is simply the short links in a mesh network. The slowest long wire is the 6-hop wire with a delay of 957ps. This is sufficient to sustain a 1GHz network such that every link requires only 1 clock cycle to transmit a signal. We consider this frequency reasonable because 3D chips have high constraint in heat dissipation and high clock frequency may not be preferred. Furthermore, the power and energy of n-hop long wires are less than n times the power and energy for a 1-hop wire. Our later results will show that using single-cycle long wires will achieve energy reductions in routers and wires, in addition to latency reduction.

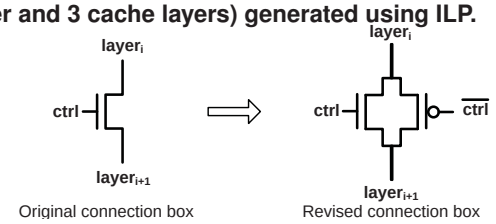


Figure 5. Connection box transistor logic.

Pipelined long wires. The single-cycle long wires can sustain a network frequency of no more than 1GHz. To obtain higher clock frequencies, the long wires must be segmented and pipelined to accommodate smaller cycle time. We implemented pipelined long wires that can sustain a 3GHz network. The long wires are divided into N segments by $N-1$ flip-flops (FF). For example, the N for the longest 6-hop wire is 3. Note that we still need repeaters to drive the wire load. The repeater distance and sizes are designed in the same way as in single-cycle long wires. However, the FFs are more expensive in both power and area than the repeaters. Therefore, we decreased their size for lower power. This shrinkage will positively impact the wire delay. Hence, we increased the repeater size to offset the delay increase while making sure we still save power. The results for repeaters, FF, delay and power/energy are given in the "ppl" columns of Table 2. As we can see, the total energy for transmitting one bit has increased noticeably for all long wires. We will show later that such increase does have an impact on the overall network energy for some workloads.

Modeling pillars. For vertical pillars, we followed the scheme proposed in [21], paying special attention to the connection box between adjacent wire segments. In our setting, the nMOS-only design with an inter-layer wire length of $50\mu m$ shows a decrease in output when passing logic 1. After several stacked nMOS stages, the output reduces to nearly the Vth (0.3V in our 45nm transistor model), which is difficult to be recognized by the next stage logic. Therefore, we revised the connection box with a pair of nMOS and pMOS —transmission gate, as illustrated in Figure 5. Simulation results show that this design solves the voltage drop problem. However, using transmission gate requires the complementary control signal, which increases the control wire numbers. We report the results based on this revised model. We obtained the energy for transmitting one bit on a pillar that goes through 2 layers ($50\mu m$), 3 and 4 layers are 111, 211, and 293fJ respectively.

-hop	repeaters		flip flops		delay(ps)		clock cycles		Dynamic P (mW)		Static P (μW)		Total E (pJ)	
link	ppl	sng	ppl	sng	ppl	sng	ppl	sng	ppl	sng	ppl	sng	ppl	sng
6	9	7	2	0	958	957	3	1	4.91	1.09	11.87	5.37	1.641	1.094
5	7	5	1	0	644	951	2	1	6.51	0.94	9.68	2.91	2.182	0.945
4	6	4	1	0	636	959	2	1	4.32	0.82	8.58	1.71	1.449	0.823
3	4	3	1	0	629	826	2	1	2.24	0.35	3.56	0.91	0.750	0.349
2	3	2	0	0	318	505	1	1	0.75	0.32	3.23	0.56	0.256	0.324
1	1	1	0	0	221	221	1	1	0.70	0.70	2.89	2.89	0.238	0.238

Table 2. Modeling results of single-cycle and pipelined long wires. "ppl" stands for "pipelined". "sng" stands for "single-cycle". "P", "E" stands for power and energy respectively. The power and energy results are for transmitting one bit on one wire. The total E is calculated as (dynamic + static power)$\times$clock cycles$\times$cycle time.

5 Performance Evaluation

In this section, we present simulation-based performance evaluation of our proposed 3D topologies, and the 3D mesh with the state-of-the-art router designs developed in previous researches [21].

5.1 Simulation Infrastructure

To model and compare different network designs, we extended a cycle-accurate 2D NoC simulator Noxim [41] developed in SystemC into a 3D network. The simulator models all major components of the NoC: routers, wires, and pillars down to the level of details such as a signal or a switch. We also augmented the energy model in the original Noxim to characterize the 3D behavior. The technology and energy parameters were obtained from Orion [34] for routers. The power model for wires and vertical pillars were from Section 4. Other essential parameters used in our simulator are listed in Table 3.

We used both synthetic and real workload traces to test different networks. For the 3D mesh, we evaluated DOR routing algorithms. The routing for our topology is deterministic as explained earlier. The synthetic traffic uses 1 flit for request messages and 5 flits for data messages. We tested Uniform Random traffic (each node uniformly injects packets into the network with random destinations) and HotSpot traffic (different processors generate requests to the same region of cache banks with high probability). The real workload traffic traces include SPLASH-2 [40], OpenMP [38] and Specjbb 2005 [39]. They are gathered from the full-system simulator Simics [27] configured into a multicore processor with large shared last level cache to mimic our 3D chip. Each workload was simulated for 50M instructions per core. This generated > 500K packets per workload. The detailed processor configurations are listed in Table 4.

The router microarchitecture has typical components as in a state-of-the-art NoC router. We have described the connection box in building vertical pillars. The intra-layer components are input buffers, a VC allocator, a routing unit, a switch allocator and a crossbar. Each router has 5 intra-layer ports, and each port of the router has 3 VCs. The buffer depth of each VC is 5 flits, the size of a packet. Packets of different message types are assigned to corresponding VCs to avoid message deadlock. The arbitration scheme of switch allocator is round-robin. We use determined routing algorithm to avoid routing deadlock, since packet could access the horizontal destination in one-hop with long wires. Both the mesh and long-link network use the same router architecture except for their port numbers.

5.2 Network Latency Reduction

Figure 6 plots the average flit latencies for a $4 \times 4 \times 4$ 3D chip (3 cache layers) using two different traffic. The curves labeled with "long" and "long_pipeline" are the results from our topology with single-cycle long wires (1GHz) and pipelined long wires (3GHz) respectively. The rest curves are results for a 3D mesh using different routing algorithms: ZXY, ZXY-XYZ, and XYZ. The ZXY-XYZ means that the communication initiated by the core is first routed down, and then across the destination layer. Among the three algorithms, our experiments show that XYZ outperforms the other two most of the time. Therefore, we only show the XYZ and pipelined long wire results for the 3GHz network.

The results show that both long-link based topologies outperform the 3D mesh in terms of latency and throughput (for HotSpot traffic) under almost all injection rates. For the HotSpot traffic, the cores may generate high volume of traffic to the same region of cache banks (4 in our setting) in one layer. As we can see, the no-load latency of the 3D mesh using XYZ routing is improved by 25.7% and 20.2% for single-cycle and pipelined long link topology respectively. Also, the network saturation point of our topology is 10.8% later than the XYZ-mesh, resulting a noticeable throughput increase under our topology. The reason for the improvements is that the long-link network distributes congested intra-layer hotspot traffic onto different layers, because the long links to the hotspot are not placed in the same layer. This effectively balances the traffic congestion in the entire network, proving the advantages of our design.

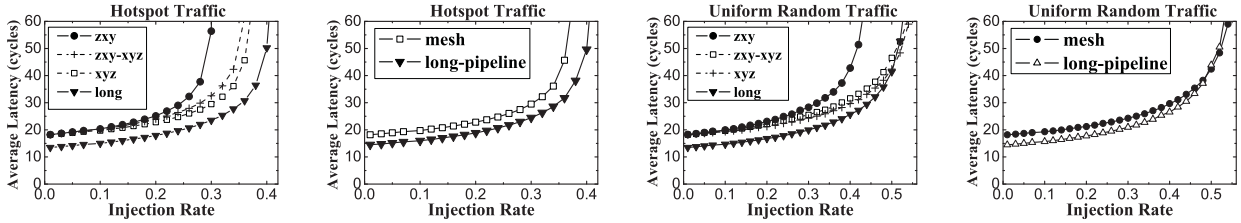
For the Uniform Random traffic, the zero-load latency sees a 25.9% and 20.4% improvement for 1GHz and 3GHz respectively. However, the network saturation point is 3.5% earlier than the 3D mesh. This is mainly due to the imbalanced traffic distribution in the topology under high uniform injection rates. Recall that the network topology of our modeled 3D chip was shown in Figure 4. The 3-layer cache design cannot accommodate all the links in a clique excluding the 1-hop links. Therefore, for those missing links, all those traffic are routed to the top core layer, creating contention when the traffic injection rate is high. In fact, the 4 clique has 120 links in total. They can be subgraphed into 5

Characteristic	Parameters
Technology	45nm
Vdd	0.8v
Network size	4×4×(4 or 5)
Routing	DOR / Determined Routing
Router delay	2 cycles
virtual channels/port	3
flit size	128 bits
number of pillars	2∼4 per node
frequency	1GHz
simulation warmup(cycles)	20,000
Analyzed packets	100,000

Table 3. Baseline network configuration.

Number of Processors	16
Issue Width	4
ISA	SPARC
L1 conf.	32K-I/D, 4 way, 64B/line, 2 cycles
L2 size	512KB/bank, 48 banks, 16 way, 64B/line, 10 cycles
Cache coherence protocol	MESI_SCMP
Memory	DDR2-800, 55ns (220 cycles)
Processor frequency	4GHz

Table 4. Architecture parameters for trace generation.



(a) 1GHz (b) 3Ghz (c) 1GHz (d) 3GHz

Figure 6. Average network latency for a 4×4×4 network with 1GHz and 3GHz frequencies.

layers, each having 24 links. The 3 cache layer design discarded 24 links, and distributed the rest (24×4) links onto 4 layers including the mesh layer. Hence the mesh layer is loaded with double the traffic of any cache layer because it absorbs both one-hop and the traffic that cannot be routed in the cache layers. If we increase the the cache layer number to 4, and allow the 4^{th} layer to also use 24 links, then the traffic will be perfectly distributed onto 5 layers, resulting a more balanced design. The latency results using 4 cache layers are plotted in Figure 7.

The results show noticeable improvement in both network latency and throughput. The no-load latency for the Uniform Random traffic is improved by 29.6% (1GHz) and 23.9% (3GHz) for long wire designs. The latencies for HotSpot traffic are also increased by 29.5% and 23.9% for 1GHz and 3GHz respectively. The saturation point of both long wire designs are 3.5% (Uniform Random) and 10% (HotSpot) later than the XYZ routing in mesh, indicating a throughput improvement as well. At this time, the latency of long wires has 86% improvement over the mesh. Also the latency gap between the two topologies does not narrow as quickly as the 3-layer design. We will use the 4-layer cache design to present the subsequent results.

Impact of pillar number. It has been noted by previous researches that the vertical communication in 3D chip is critical to the network performance [21]. This is also the case in our design, as we use vertical hops to reduce the complexity of routers. Previous works have shown that Through-Silicon-Vias (TSV) have pitches of 4∼10μm [12, 21]. With this dimension, the area consumed by a bundle of 161 wires (128 bit data + control signals) is around 0.01mm^2. With this area, the state-of-the-art routers can have more than 30 [25] to 100 [11] pillars. Hence, increasing the vertical pillar seems feasible and will likely not be a limiting factor for several generations [21, 26]. Figure 8 shows the latency

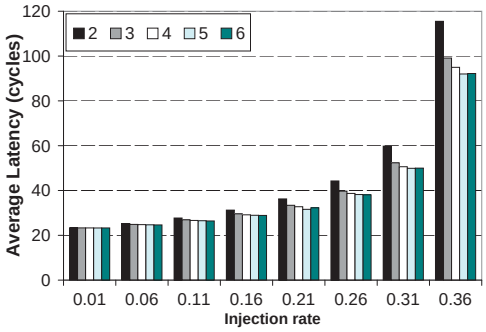
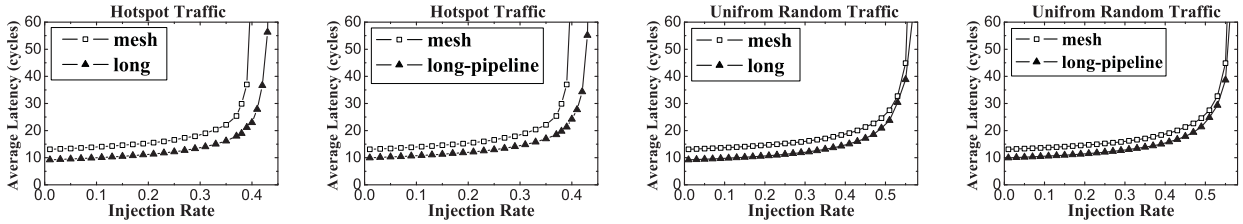


Figure 8. Latency reduction with number of pillars.

reduction as we increase the pillar count from 2 to 6, with increasing flit injection rate. As we can see, although the pillar density is not a concern, increasing the pillars gives diminishing returns. From 2 pillars to 3 pillars, the latency improvements are clearly observed; but beyond 4, the gains are negligible. Therefore, we choose 4 as our pillar count for each router.

5.3 Energy Reduction

We have introduced our wire energy and delay model in Section 4. The router energy is modeled in Orion [34]. We then measured the energy reduction of our topology compared to the 3D mesh using synthetic traffic of the same number of packets, at an injection rate close to their network saturation points to fully activate all components of the networks. Figure 9 shows the measured energy reductions in



(a) 1GHz (b) 3GHz (c) 1GHz (d) 3GHz

Figure 7. Latency improvement for a 4×4×5 network of 1GHz and 3GHz.

router, wires and their sum for the single-cycle long wire design. As we can see, the router energy reduction is from 46-55% mainly because the significant hop count reductions achieved in our topology. The wires also have some energy reduction, ranging from 14-15%. This is because: 1) We applied a leakage reduction technique, described in section 5.4, for links that are idle. We observed more leakage reduction than dynamic energy reduction from the mesh. 2) The energy of n-hop wires is less than n times the energy of 1-hop wires, for single-cycle long wires. Hence, it pays off to use long wires to reduce hop count. The total energy reduction range from 46-55% because the dominant energy is dissipated in routers.

Figure 9. Energy reduction compared with 3D mesh using XYZ routing. Results are collected near the network saturation point.

5.4 Results for Real Workload Traces

Figure 10 shows the average network latencies and energy consumption of the real workload traces we collected for the 4-layer cache configuration. The single-cycle and pipelined wire results are normalized to the 3D mesh using XYZ routing with 1GHz and 3GHz clock frequency respectively. We can see that no matter which wire frequency is used, the long link based topology consistently reduces the latency across all tested benchmarks. They show a fairly steady reduction amount: 8.8%∼24.2% (single-cycle) and 6.7%∼19.9% (pipelined long wire), with the largest seen in benchmark *raytrace* and *water-spatial* (SPLASH-2), and average of 20.5% (single-cycle) and 15.9% reduction (pipelined long wire). This is mainly because these real workload traces present near Uniform Random characteristic with relatively low flit injection rate.

The energy reductions from these benchmarks however are not as significant as those with synthetic traffic. The results are 1.6%∼16.8% and -1.7%∼9.6% for single-cycle and pipelined long wires respectively. The pipelined wire design, unsurprisingly, increased the total energy for *fft* and *volrend*. Other benchmarks show a mild reduction. We observed that the leakage energy becomes dominant in the total energy consumption because the flit injection rates of these benchmarks are much lower than the network saturation point. Hence, many components are idle, consuming only leakage energy. We developed a leakage reduction method that can turn off the long links while they are idle. We turn off the all the links of a router when all its internal buffers are empty. We turn the links on whenever there is a flit enters the router. This is conservative, but simple to implement. A more aggressive router with techniques such as pre-routing can further lower the leakage by accurately turning on and of the long links. On average, the single-cycle and pipelined long wires achieve 8.5% and 4% energy reduction over a baseline 3D mesh network.

5.5 Hardware Overhead Comparison

In this section, we compare the hardware savings in routers of our topology, and the overhead introduced by long wires.

The area of a router, denoted as S_R, can be modeled as S_R=H_1n+H_2n^2. Here H_1 and H_2 are area coefficients related to dimensions of a channel, input and output buffers, crossbar etc. [2]. n is the radix of a router. Since both our topology and the 3D mesh use the same ports for vertical links, we will only count the lateral ports for clarity. In a 4×4×4 3D mesh, each layer has 4 5-port routers, 4 3-port routers, and 8 4-port routers. Hence, $S_R(mesh)$=3×(4×5+4×3+8×4)H_1+3×(4×5^2+4×3^2+8×4^2)H_2=192H_1+792H_2. In our 3-cache layer long link based topology, every router has 4 ports except for the last layer, 2 routers there have only 3 ports due to the area constraints. We can come up with a similar equation and get $S_R(long)$=190H_1+754H_2. As we can see, $S_R(long)$¡$S_R(mesh)$. In the 4×4×5 network, $S_R(mesh) = 256H_1+1056H_2$, and $S_R(long) = 256H_1+1032H_2$. As we can see $S_R(long) < S_R(mesh)$ still holds. Hence, the router area in our long-link based topology is less than that of a 3D mesh network.

The overhead we pay in this design is the longer and wider wires. Here we quantify such an increase. In a 4×4×4 3D mesh, the total number of links is $S_{wtotal}(mesh)$=24×3=72S_w without the top mesh layer since it is the same for both networks. The S_{wtotal} denotes the total wire area, and S_w denotes the area of a 1-hop link. With our long wire design, adding the hop counts of all our wires and considering the 4× the area for 3-6 hop links, we get $S_{wtotal}(long)$=640S_w. Hence, $S_{wtotal}(long)$=8.9$S_{wtotal}(mesh)$. Using the same method for the 4×4×5 network, we can obtain $S_{wtotal}(long)$=7.2$S_{wtotal}(mesh)$. We remark that al-

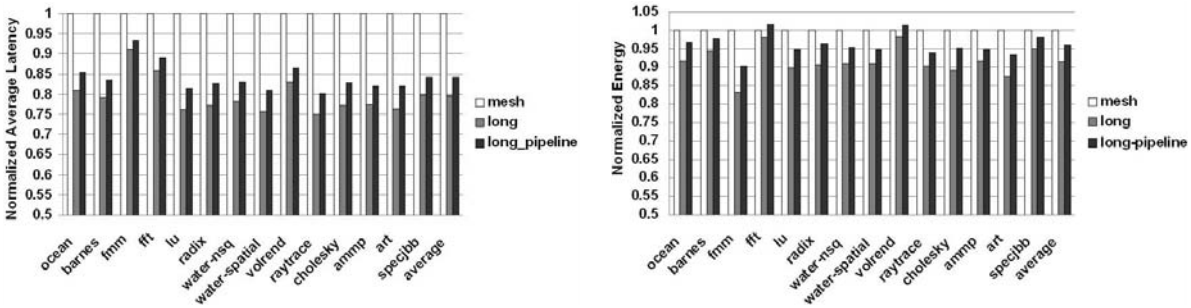


Figure 10. Latency and energy reduction for SPLASH2, OpenMP, and SpecJbb 2005 workloads.

though the area of the long wires is 7∼9× of the mesh topology, global wires are placed on top metal layers and do not take the space of the device layer. Hence, such an area "increase" is not a burden to the chip real estate. Also, as we have defined the routing and area constraints in developing the topology, the increased wire area will be unlikely to impact the layout of the rest of the chip.

6 Scalability

Our proposed topology using long links can be scaled in different ways. We have shown in Section 3.2 that our design is workable for core count up to 36 (6×6), with current and near future 3D stacking technology. Beyond this number, we have to increase either the radix of the router or stacking depth in order to fit a clique into all layers in 3D.

One method of scaling the topology is to use the Concentrated mesh [2] (CMesh) as our baseline for improvement. The CMesh network reduces a regular mesh to a radix-4 mesh in which each router services four processors. For example, a 64-node mesh can be concentrated into a 16-node CMesh with the same topology. Although a single CMesh's router is larger compared to a single router of a regular mesh, it is not four times larger. Hence, the total router area of a CMesh is smaller than that of a regular mesh. Concentration can also reduce the hop count of the network. It was studied that concentration improves both area efficiency and energy efficiency of a network [2]. Using concentration, our design on a 4×4×L network (L is number of layers) can be expanded to an 8×8×L network, with each router servicing 4 cores or cache banks. Similarly, a 6×6×L can be expanded to 12×12×L, which is already on the high-end of multicore designs. Naturally, when a wire is overly long, we can apply the pipelined design as described in Section 4. Such a technique has also been investigated recently in topologies using long wires [18].

Another way of scaling is to relax the diameter further. Up till now we have been focusing on a clique for an N-node network (N≤36), where a node can be either a regular mesh router or a concentrated router. We can apply the same principle to subgraphing a 2-hop diameter network such as the flattened butterfly [18] onto different layers with still low-radix routers. For example, for a 7×7 network, the radix of each router is 6×2=12 (not counting the local port). Then the total number of links in the flattened butterfly network is 49×12/2=294. If we use a radix-4 router in our 3D topology, each layer can host 4×49/2=98 links. Then we only need to stack 294/98=3 layers of cache to incorporate all links. Similarly, subgraphing the flattened butterfly of a 10×10-node network requires only 5 layers of 3D stacking with radix-4 routers (not counting the local port). Note that we can continue to scale using concentration here.

7 Related Work

There have been many researches lately on improving the network performance. They can be broadly categorized into topology optimization and router enhancement.

For 2D network topology, the "Small-World" design [30] inserts extra long links into a 2D mesh to exploit the application-specific bandwidth requirement. Our design is general purpose and the traffic was assumed to be rather uniform across the entire network. Also, inserting links into a mesh increases the complexity of the router. The flattened butterfly topology uses high-radix routers with concentration to achieve a diameter of 2 NoC for a 64-node network. Again, one of our goals is to use low-radix routers in a 3D network to meet the power and area constraint. Express cubes [9], using physical express channels to connect distant nodes, loses performance when a number of neighboring nodes compete for the same express channel. Also it takes more than one hop for data to traverse from local router to interchange units, which means that the diameter of express cubes is larger than our proposed topology. The Express Virtual Channel (EVC) [23] design for a planar mesh uses a novel flow control mechanism and router microarchitecture renovation to allow packets to virtually bypass intermediate routers along their path. However, the diameter of the 2D network is strictly larger than 1.

Linear programming has also been used in custom 2D NoC designs for SoCs where the bandwidth requirement are known ahead of time, but the number of routers and the power consumption of the network need to be minimized, subject to LP solving time [33]. We are targeting a more general purpose network rather than an application specific custom network.

A 3D dimensionally-decomposed router was designed to provide a good tradeoff between circuit complexity and performance benefits. With this design, the communication between any two layers requires only a single hop, and non-overlapping vertical communication can carry in parallel – a significant improvement over the bus architecture [25]. We are already using the properties of this router in our topology design.

8 Conclusion

We presented a new network topology using long-range links for 3D CMPs. The new topology has a low diameter, but requires only low-radix routers to implement. We modeled the long-range links and showed that they have la-

tency advantages even when we increase the network frequency and pipeline the wires. Also, their power/energy increase is sub-linear to their increase in length. We experimented a 4×4×4 and a 4×4×5 3D network. Our experiments show that a 1GHz network frequency is more suitable for 3D chips because it achieves more latency and energy reductions. A higher clock frequency for a 3D chip brings the concern of high heat dissipation, and therefore is not recommended for implementation (though its latency and energy results are still positive). Our topology can be scaled to larger networks using techniques such as network concentration.

References

[1] M. Awasthi, V. Venkatesan, and R. Balasubramonian. Understanding the Impact of 3D Stacked Layouts on ILP. *J. of Instruction-Level Parallelism,* Vol. 9: 1-27, 2007.

[2] J. Balfour and W. Dally. Design Tradeoffs for Tiled CMP On-Chip Networks. *the 20th Int. Conf. on Supercomputing,* pp. 187-198, 2006.

[3] M. Berkelaar, K. Eikland, and P. Notebeat. LP_solve: Opern Source (Mixed-Integer) Linear Programming System (2007). http://lpsolve.sourceforge.net/5.5/

[4] B. Black, *et al.* Die Stacking (3D) Microarchitecture. *Proc. of the 39th Int. Sym. on Microarchitecture,* pp. 469-479, 2006.

[5] P. Caputa and C. Svensson. A 3Gb/s wire Global On-Chip Bus with Near Velocity-of-Light Latency. *the 19th Int. Conf. on VLSI Design,* pp.117-122, 2006.

[6] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, S.-W. Tam. CMP Network-on-Chip Overlaid with Multi-Band RF-Interconnect. *the 14th Int. Sym. on High-Perf. Comp. Arch.,* pp. 191-202, 2008.

[7] R. T. Chang, N. Talwalkar, C. P. Yue, S.S. Wong. Near Speed-of-light Signaling over On-chip Electrical Interconnects. *IEEE J. of Solid-State Circuits,* 38(5):834-838, May 2003.

[8] L. Chen, N. Muralimanohar, K. Ramani, R. Balasubramonian, J. Carter. Interconnect-Aware Coherence Protocols for Chip Multiprocessors. *Proc. of the 33rd Int. Sym. on Comp. Arch.,* pp. 339-351, 2006.

[9] W. J. Dally. Express Cubes: Improving the Performance of k-ary n-cube Interconnection Networks. *IEEE Trans. on Computers,* 40(9):1016-1023, 1991.

[10] R. Fourer, D. M. Gay, and B. W. Kernighan. AMPL: A Modeling Language for Mathematical Programming. *Duxbury Press Publishing Company,* 2nd ed. 2002.

[11] P. Gratz, C. Kim, K. Sankaralingam, H. Hanson, P. Shivakumar, S. W. Keckler, D. C. Burger. On-Chip Interconnection Networks of the TRIPS Chip. *IEEE Micro,* 27(5):41-50, September/October 2007.

[12] S. Gupta, M. Hilbert, S. Hong, and R. Patti. Techniques for Producing 3D ICs with High-Density Interconnect. *Proc. of the 21st Intl. VLSI Multilevel Interconnection Conference,* 2004.

[13] R. Ho, K. Mai, and M. Horowitz. The Future of Wires. *Proc. of the IEEE,* 89(4):490-504, 2001.

[14] D. Ingerly, *et al.* Low-K Interconnect Stack with Thick Metal 9 Redistribution Layer and Cu Die Rump for 45nm High Volume Manufacturing. *Interconnect Tech. Conf., IITC* pp. 216-218, 2008.

[15] D. N. Jayasimha, Bilal Zafar, Yatin Hoskote. On-Chip Interconnection Networks: Why They are Different and How to Compare them. *Tech. Rep., Intel* http://blogs.intel.com/research/terascale/ODI_why-different.pdf

[16] J. W. Joyner, P. Zarkesh-Ha, and J. D. Meindl. A Stochastic Global Net-Length Distribution for a Three-Dimensional System-on-Chip (3D-SoC). *The 14th IEEE Int. ASIC/SOC Conf.,* pp. 147-151, 2001.

[17] T. Kgil, *et al.* PicoServer: Using 3D Stacking Technology to Enable a Compact Energy Efficient Chip Multiprocessor. *Proc. of the 12th Int. Conf. on Arch. Support for Prog. Lang. & Operating Sys.,* pp. 117-128, 2006.

[18] J. Kim, J. Balfour, and W. J. Dally. Flatterned Butterfly Topology for On-Chip Networks. *Proc. of the 40th Int. Sym. on Microarchitecture,* pp. 172-182, 2007.

[19] J. Kim, D. Park, T. Theocharides, V. Narayanan, C. Das. A Low Latency Router Supporting Adaptivity for On-Chip Interconnects. *Proc. of the 42nd Conf. on Design Auto.,* pp. 559-564, 2005.

[20] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, and C. R. Das. A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks. *Proc. of the 33rd Int. Sym. on Comp. Arch.,* pp. 138-149, 2006.

[21] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. Das. A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architecture. *Proc. of the 34th Int. Sym. on Comp. Arch.,* pp. 4-15, 2007.

[22] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi. Leveraging Optical Technology in Future Bus-Based Chip Multiprocessors. *Int. Sym. on Microarchitecture,* pp. 492-503, 2006.

[23] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha. Express Virtual Channels: Towards the Ideal Interconnection Fabric. *Proc. of the 34th Int. Sym. on Comp. Arch.,* pp. 150-161, 2007.

[24] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling. *Proc. of the 32nd Int. Sym. on Comp. Arch.,* pp. 408-419, Madison, USA, 2005.

[25] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, M. Kandemir. Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. *Proc. of the 33rd Int. Sym. on Comp. Arch.,* pp. 130-141, 2006.

[26] G. H. Loh. 3D-Stacked Memory Architecture for Multi-Core Processors. *the 35th Int. Sym. on Comp. Arch.,* pp. 453-464, 2008.

[27] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner. Simics: A Full System Simulation Platform. *IEEE Computer,* 35(2):50-58, 2002.

[28] R. Mullins, A. West, and S. Moore. Low-Latency Virtual-Channel Routers for On-Chip Networks. *Proc. of the 31st Int. Sym. on Comp. Arch.,* pp. 188-197, 2004.

[29] N. Muralimanohar, R. Balasubramonian. Interconnect Deign Considerations for Large NUCA Caches. *Proc. of the 34th Int. Sym. on Comp. Arch.,* pp. 369-380, 2007.

[30] U. Y. O. and R. Marculescu. It's a Small World After All: NoC Performance Optimization via Long-Range Link Insertion. *IEEE Trans. on VLSI Sys.,* 14(7):693-706, July 2006.

[31] D. Park, S. Eachempati, R. Das, A. K. Mishra, Y. Xie, V. Narayanan, C. Das. MIRA: A Multi-Layered On-Chip Interconnect Router Architecture. *Proc. of the 35th Int. Sym. on Comp. Arch.,* pp. 251-261, 2008.

[32] K. Puttaswamy and G. H. Loh. Thermal Herding: microarchitecture techniques for controlling hotspots in high-performance 3D-integrated processors. *Int. Sym. on High-Perf. Comp. Arch.,* pp. 193-204, 2007.

[33] K. Srinivasan, K. S. C., and G. Konjevod. Linear-Programming-Based Techniques for Synthesis of Network-on-Chip Architectures. *IEEE Trans. on VLSI Sys.,* 14(4):407-420, 2006.

[34] Hang-Sheng Wang, Xinping Zhu, Li-Shiuan Peh, Sharad Malik. Orion: A Power-Performance Simulator for Interconnection Networks *Proc. of Int. Sym. on Micro.,* pp. 294-305, 2002.

[35] 45nm BSIM4 model card for bulk CMOS: V1.0, Feb 22, 2006, http://www.eas.asu.edu/ ptm/latest.html

[36] PTM interconnect model. http://www.eas.asu.edu/ ptm/ interconnect.html

[37] ITRS roadmap. *Tech. Rep.,* 2005, and 2006.

[38] OpenMP. http://openmp.org/wp

[39] SPECjbb 2005. http://www.spec.org/jbb2005

[40] SPLASH-2. http://www-flash.stanford,edu/apps/SPLASH/.

[41] Noxim, An Open Network-on-Chip Simulator. http://noxim.sourceforge.net