The Roce-Bush Router: A Case for Routing-centric Dimensional Decomposition for Low-latency 3D NoC Routers

Miguel Salas
Intel Corporation
Fort Collins, CO, USA
miguel.o.salas@intel.com

Sudeep Pasricha
Electrical and Computer Engineering Dept.
Colorado State University, Fort Collins, CO, USA
sudeep@colostate.edu

Abstract

As 3D System-On-Chips (SoCs) come ever closer to becoming the standard for high performance ICs, 3D Networks on Chips (NoCs) have emerged as a key component in meeting performance constraints and ensuring power-efficiency. Among the proposed 3D router architectures, dimensionally-decomposed routers are widely accepted as an efficient solution to deal with the increased port count and the accompanying exponential power and area increases. All decompositions proposed thus far have however been dimensionally static, that is, they have set in stone a particular bias among the three dimensions. This paper presents a novel router with a routing-centric decomposition and virtual channel buffer sharing called the *Roce-Bush* router. To our knowledge, this is the first work that integrates routing-awareness in the context of dimensional decomposition and buffer resource allocation for NoC routers. Experimental results involving RTL level implementations of our router and synthesis at 45nm show that compared to a dimensional-agnostic decomposed router, the Roce-Bush router can achieve up to 14% better performance and 5% lower power.

Categories and Subject Descriptors
B.5 **[Register-Transfer-level implementation]** - Data-path design
C.1.2 **[Multiple Data Stream Architectures]** - Interconnection architectures

Keywords: Network-on-chips, NoCs, System-on-chips, 3D ICs, router architecture

I. INTRODUCTION

As silicon chips continue to scale into the deep submicron era, the performance bottleneck of Chip Multiprocessors (CMPs) has shifted to their interconnection network. The ever-increasing latency and throughput requirements on CMPs that must stay within a stringent power envelope are slowly making bus-based interconnects impractical due to their limited throughput, power inefficiencies, and poor scaling. More importantly, the number of cores is increasing rapidly, as can be easily seen in the current state-of-the-art CMPs, with some commercial offerings integrating around 100 cores [23].

CODES+ISSS'12, October 7–12, 2012, Tampere, Finland.

Network-on-Chips (NoCs) are quickly becoming the de-facto standard for communication in CMPs with such a large number of cores [1]. However, *one of the main challenges* with NoCs is long packet latencies that can become a major performance bottleneck. To this end, a NoC router is required to not only be extremely power and area efficient, but must also allow for a large throughput and low latency. Figure 1 shows a typical NoC router [2] that supports virtual channels, which is a common way to dramatically lower transfer latencies. It has five key components: Port Buffers, Route-Computation (RC), Virtual-channel Allocator (VA), Switch-Allocator (SA), and a Crossbar Switch. The specific router architecture chosen in a NoC plays a vital role in its performance, and thus much research has focused on designing a highly efficient router across the latency-throughput-area-power spectrum [4]-[11].

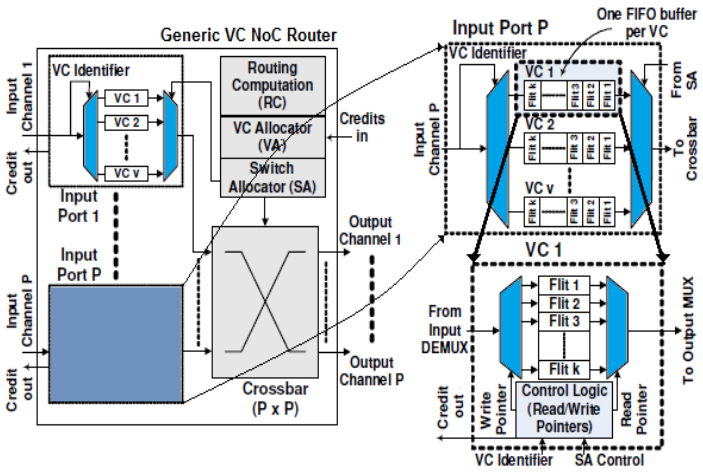


Figure 1. A Generic Virtual Channel (VC) Router [2]

The use of multiple layers of silicon within a die, coined a 3D IC, has very recently emerged as one solution to mitigate the high latencies due to long interconnects as illustrated in Figure 2 [3]. By using order of magnitude shorter vertical Through Silicon Vias (TSVs) to replace long global interconnects, the inter-core communication latency can be dramatically reduced. Since each layer is simply a 2D IC, most circuit design concepts still apply. Thus, it is no surprise that first-pass 3D NoCs proposed in literature [24] are simply made up of 2D routers with two extra ports – the up and down ports for inter-die communication (Z-direction). However, such extension to NoC routers that have an $n \times n$ structure (n is the number of ports) can significantly increase fanout, area, and complexity, *resulting in very inefficient 3D routers*.

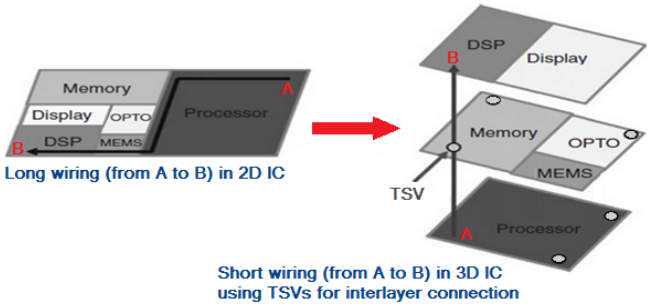


Figure 2. Motivation for migrating from 2D to 3D ICs [3]

In this paper, we propose a novel 3D dimensionally decomposed router (the *Roce-Bush* Router) that is optimized for low-latency and when compared to the state-of-the-art decomposed routers, can achieve significantly better performance with lower power dissipation. Our Routing-Centric Buffer-Sharing (*Roce-Bush*) router supports decompositions that exploit the patterns of a given routing scheme. Furthermore, it allocates shared buffer resources to exploit these patterns to further enhance communication performance. Finally, it also decomposes the arbitration modules within a router. Using all of these enhancements results in a novel 3D router that is power and area efficient while also meeting low-latency design requirements and supporting virtual channels.

II. RELATED WORK

A. 3D Decomposed Routers

Coming up with efficient routers for 3D NoCs has been a topic of avid research. One of the most popular solutions that have recently emerged is a *dimensionally-decomposed* router, that is, a router that is decoupled among certain directions in the XYZ Cartesian plane. A typical decomposed router avoids a 7×7 crossbar and other router components that have $n \times n$ complexities, depending on the actual router implementation, where n is the number of router ports (radix). A decomposed router breaks up these components into a set of smaller components, thereby avoiding the prohibitive area and power increase of a large crossbar and other $n \times n$ components.

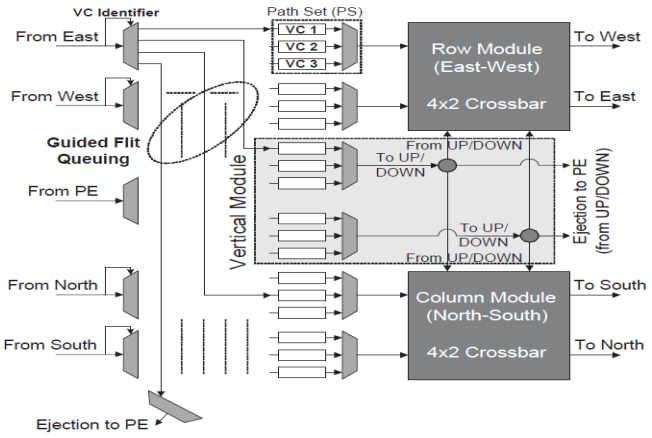


Figure 3. A Fully dimensionally decomposed router [4]

One well known solution in dimensionally decomposed 3D NoC routers is shown in Figure 3 [4]. This router is fully decomposed among each of the 3 dimensions: there is a module for X-routing, one for Y-routing, and one for Z-routing. Implied across each direction are 2 stages of arbitration and a crossbar. For example, for a flit to go to west or east (X-module), it must first go through two 3×2 crossbar stages to select 2 out of the 5 inputs (east, west, north, south, core). Then, it must couple the two winners with the Z-direction via a 4×2 crossbar. The same applies to the Y and Z directions, resulting in a design that sacrifices latency and throughput (due to arbitration as will be discussed in section IV) in favor of a smaller and more power-efficient design.

While the router decomposed in all directions is simple and efficient, it is not practical for latency and throughput-sensitive designs. A more recent and popular decomposed 3D NoC router is shown in Figure 4 [5], [6]. This decomposed router keeps the latency and throughput characteristics of a 2D design and decouples one of the directions, ending up with two 4×4 routers that still finish within 1 full clock cycle, thereby making it fall under the single cycle router category. Thus, the router functions as a single synchronous router with some latency increase due to the two stages of arbitration and logic. The final result is a router with a decomposition that partitions in *core-Zdir* and *Xdir-Ydir* paths, and hence results in a compromise between the area-power efficient decomposition-across-all-directions and a low-latency but implementation inefficient "composed" design. However, this particular implementation has an inherent bias towards the packets traveling in the *Z* direction, which see lower latency as they bypass the *X-Y* arbitration, and just see a net latency provided by its 4×4 crossbar (up, down, core and "other"). It can be argued that this is because all other things equal, the *Z* direction might see the overhead of TSV propagation and thus it might make sense to bias the router towards Z-direction in hopes of cancelling the TSV delay effect. That brings up another question: *What if we chose the X-direction as the preferred direction? Or what if we chose Y?* Our *Roce-Bush* router attempts to answer these specific questions, proving that indeed, there is a notable performance improvement if we decompose the NoC router in the appropriate direction. To this end, we implemented the dimensionally static 3D router from Figure 4 as a comparison point against our *Roce-Bush* router.

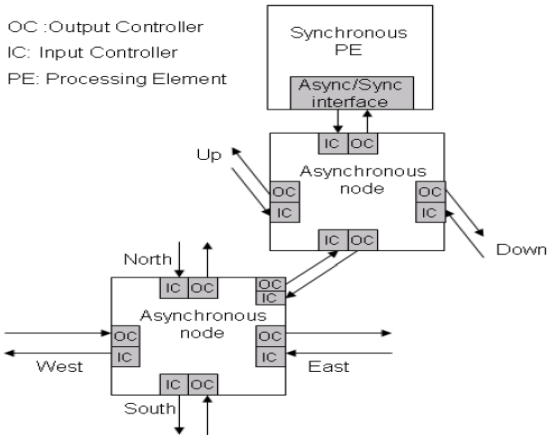


Figure 4. Z-biased 2-level decomposed 3D NoC router [5]

Note that the previous two 3D decomposed routers are not an exhaustive list by any means. Other notable decomposed routers have also been proposed. For example, the MIRA router [7] proposes having a 3D router span among multiple layers, thereby reducing latencies for packets travelling across each

layer (Z-direction travel). Nevertheless, routers such as MIRA understate the difficulty in physical design and validation complexity by not being able to keep a given design on a layer. For example, a physical designer of the router might require special synthesis tools to incorporate across-layer information, where small inaccuracies can lead to very low yields. Furthermore, silicon validation across layers would require a very complex interlayer validation scheme [8].

B. Other high performance 3D routers

Decomposed routers are by no means the only proposed high performance NoC routers. Figure 5 shows a speculative router, which is one of the most popular low-latency router designs [9]. The speculative router tries to request crossbar access without knowing whether a packet has reserved a virtual channel or not. Speculative routers have demonstrated good performance in 2D routers. Nevertheless, it has been pointed out that their clock period and power dissipation costs are extremely high, not only due to the abort-logic overhead, but also due the wasted dynamic power every time the speculation does not work [9]. To compare the *Roce-Bush* router performance against other state-of-the-art 3D routers, we also implemented a 3D speculative NoC router.

Finally, large amounts of research on 3D NoC routers have focused on application-specific routers, e.g., [10], [11]. However, the *Roce-Bush* router attempts to improve on generic 3D router architectures for NoCs, and thus avoids an application specific implementation.

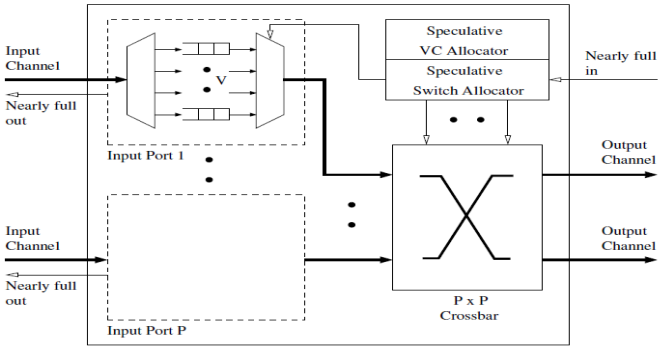


Figure 5. Speculative NoC router design [9]

III. BACKGROUND: 3D NOC ROUTER DECOMPOSITION

In most literature on router decomposition, typically the only part of a NoC router that is considered for decomposition is the crossbar, as it is a fully composed $n \times n$ structure whose complexity grows significantly with increasing n. Some of the proposed decomposed crossbars have been shown to have a large impact on performance [22]. However, decomposing only the crossbar neglects the fact that low latency routers make use of very efficient and more complex arbitration modules for the switch and virtual channel allocation. One of the most popular of these arbitrations is explored in [12], [13], where switch arbitration is done through matrix arbiters and virtual channel arbitration through a tree arbiter made up of parallel matrix arbiters. Switch arbiters are simple $n \times n$ arbiters that through fair arbitration can achieve low latencies. What follows in this section describes the decomposition in each of the typically *composed* router components, for an $n \times n$ complexity.

A. Decomposed Crossbar

A simple extension of the 5×5 crossbar found in typical 2D NoC routers would yield a very large 7×7 crossbar in 3D. Figure 6 shows three efficient ways to decompose a 7×7 crossbar. Figure 6(a) shows a decomposition that, while balancing the 7 loads evenly into two 4×4 crossbars, does not lend itself to any dimensional decomposition. Figure 6(b) shows the type of crossbar that allows for dimensional decomposition. This results in a two-stage 3×3 and 5×5 set of crossbars. Another type of decomposition is explored in figure 6(c), with two-stage 3×3 and 4×4 crossbars in *parallel*, and a 2:1 mux to decide between the two. Note that while Figure 6(c) shows the most power and area efficient design among them all, resulting in the smallest number of gates, Figure 6(b) also has an advantage: because the critical path of an ultra-low latency router is always its control logic, there is a slight latency improvement by using this implementation over the parallel decomposed crossbar. Whichever decomposition is ultimately used in a dimensionally decomposed router depends on whether area, leakage power or latency is the more important design constraint.

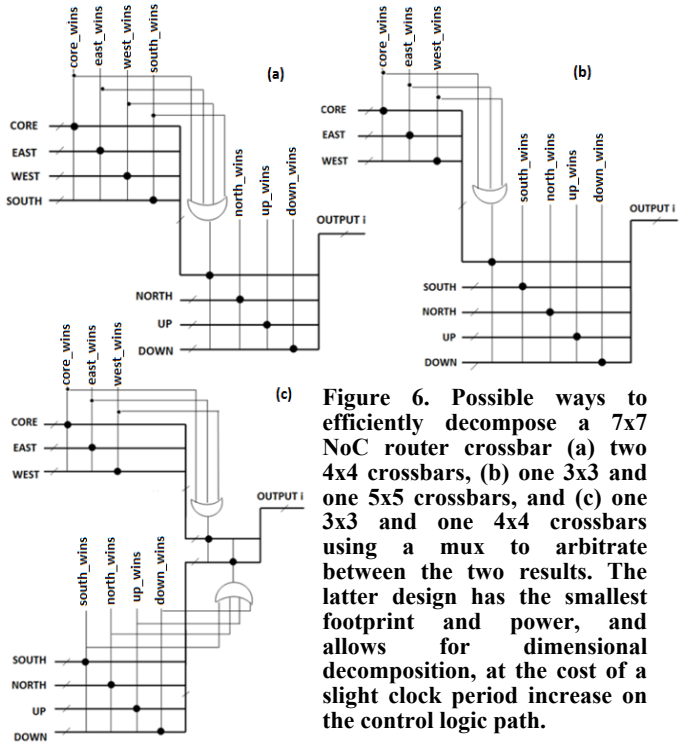


Figure 6. Possible ways to efficiently decompose a 7x7 NoC router crossbar (a) two 4x4 crossbars, (b) one 3x3 and one 5x5 crossbars, and (c) one 3x3 and one 4x4 crossbars using a mux to arbitrate between the two results. The latter design has the smallest footprint and power, and allows for dimensional decomposition, at the cost of a slight clock period increase on the control logic path.

B. Decomposed Switch Arbitration

What is rarely discussed in literature on router decomposition is decomposition of non-crossbar router elements, primarily because the baseline-router for decomposition so far has been extremely simple. Some of the studied routers are simply made out of a naive switch allocator, a crossbar, and a routing module. However, for ultra-low latency, a better complexity-performance compromise is achieved by the matrix arbiters, shown in Figure 7(a). A matrix arbiter keeps track of the requests by setting the corresponding bit in a matrix to 1 and setting the entire row of the requester back to 0 whenever a particular requester wins the current arbitration [13].

However, the matrix arbiter is of $n \times n$ complexity as can be seen in Figure 7(b), resulting in a rather large 49-element matrix for n=7, with fairly complex corresponding control logic. To overcome this challenge, one must decompose the switch arbitration as well. Decomposing the arbiter in a dimensionally-exploitable manner is shown in Figure 7(c), which accomplishes power and area savings from the control logic reduction. The remaining challenge involves deciding how to arbitrate in the case that there is a request from the *preferred* directions (X direction in the figure 7(c)) and the *non-preferred* directions (Y and Z directions in Figure 7(c)) to keep the savings. We define the *preferred* dimension in the decomposition scheme as the dimension that lives next to the core, contending only within the dimension and the core. The other two dimensions are thus called *non-preferred*. This challenge is addressed in section IV.

C. Decomposed Virtual Channel Allocation

Finally, in low-latency routers, FIFO-based virtual channels are a common practice to avoid performance bottlenecks such as head-of-the-line blocks (HOL blocking), and to avoid deadlocks [14]. An efficient implementation is presented in Figure 8(a) [13]. It consists of a tree of $n \times (n \times V+1)$ parallel matrix arbiters, where V is the number of virtual channels. The tree functions as follows: each output port contains n arbiters deciding which VC of the given input port wins the arbitration, calculated in parallel to another arbiter that determines which input port gets to win the requested output port. This results in a scheme that contains $n \times V+1$ matrix arbiters per output port. As VC allocation is critical for low-latency routers, this level of VC complexity is often excused with the performance gain.

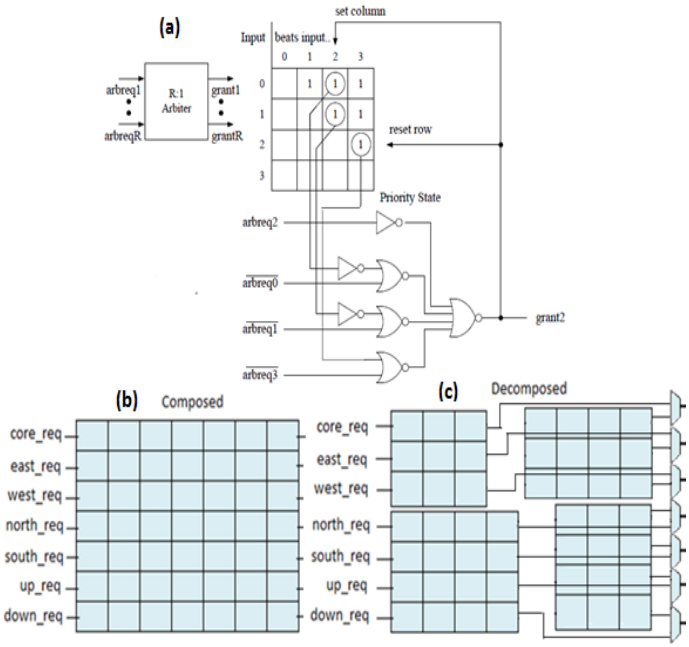


Figure 7. Matrix arbitration for switch allocation (a) 4x4 matrix of arbiters for switch allocation (b) composed 7x7 switch allocation for 3D routers; each block is 1 element of the matrix corresponding to the requestor (row) and requested (column) (c) decomposed switch allocation that decomposes each of the directions and adds a mux per output port for the last stage arbitration.

However, extending this implementation to 3D results in a non-trivial complexity hike. For example, a scheme with 2 VCs, which is the minimum complexity and size required to take advantage of minimal HOL blocking and deadlock freedom provided by VCs, results in a scheme with 15×7 matrix arbiters. Figures 8(b) and 8(c) elaborate on the actual implementation differences. The darker lines indicate the separation among each input port, in this case 2 column-wide to represent 2 Virtual Channels per input port. It can be seen that in the decomposed scheme, we again end up with the dilemma of how to arbitrate among the preferred and non-preferred directions. Furthermore, as in the case of the switch arbitration, all the n-dependant logic of the tree (such as the matrix-arbiter control logic) has exponentially decreased complexity through the use of a 3×3-4×4 decomposed schemes over a 7×7 composed scheme.

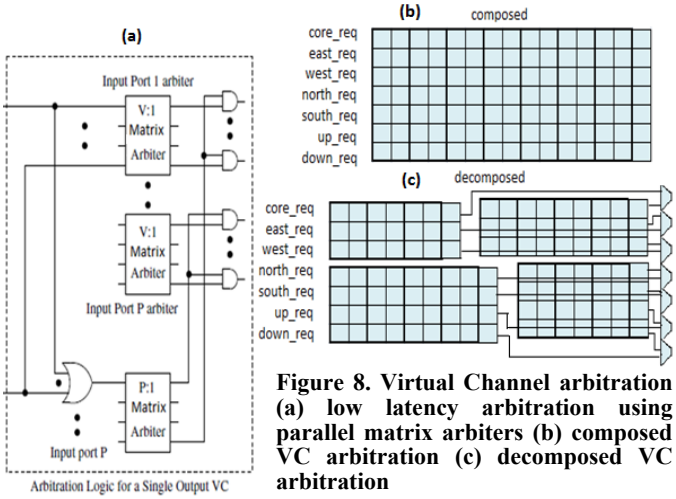


Figure 8. Virtual Channel arbitration (a) low latency arbitration using parallel matrix arbiters (b) composed VC arbitration (c) decomposed VC arbitration

IV. THE ROCE-BUSH 3D NOC ROUTER

In this section, we describe the key components of our Roce-Bush 3D NoC router: routing-centricity, physical optimizations, decomposed arbitraton, and directionality-aware buffering.

A. Routing centricity

Although there is literature on routing centric NoC design such as the recent multimedia application-specific design presented in [11], the *Roce-Bush* router is the first work to focus on routing centricity towards decomposed routers. We conjecture that biasing the decomposition in the Roce-Bush router based on the routing scheme chosen can lead to better performance. The motivation for routing centricity is that since dimensionally-decomposed routers rely on some static assumptions (such as TSV effects on 3D SoCs) for choosing a certain dimensional decomposition over another, a systematic routing scheme may present a strong case for providing a better and different dimensional bias. The routing schemes considered for routing-centric decomposition in our router are:

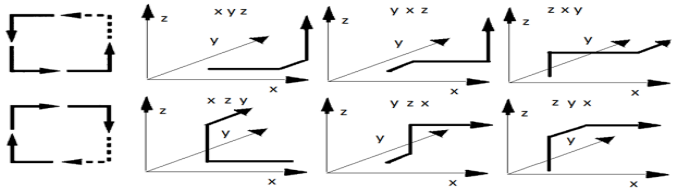
1) Dimension-ordered Routing

One of the most popular routing schemes for 3D NoCs is dimension-ordered routing (e.g., XYZ) because of its simple routing logic implementation. However, dimension-ordered routing presents a very systematic way of routing flits through the network: first route in dimension A, then in dimension B, and finally in dimension C. Qualifying a router in terms of how

it performs across each direction on a fully symmetrical network often shows differences that can be classified as statistical noise if all the ports and router logic are symmetrical with respect to the *X, Y* and *Z* directions. However, by adding a dimensional-bias, which is the heart of the decomposed routers presented thus far, the original assumption is eliminated altogether. This opens up the question of *which direction to decompose on and how to dimensionally bias that decomposition in the most efficient manner.*

2) Partially adaptive Routing

While simple dimension-order routing remains a popular choice, it certainly is not the de-facto standard for routing. Several enhanced routing schemes for mesh networks have been proposed [15]-[18]. Among them, partially adaptive turn-model based routing schemes such as the west-first routing algorithm, *can exploit dimensional decomposition*. This is because there is a pattern to the routing scheme. For the west-first algorithm, first proposed in [15], flits are routed in the west direction first, and only then is routing in other directions allowed (Figure 9(a)). Such behavior puts a higher load on *X*-direction travel, making it an inherently dimensional biased routing and thus leads itself to routing-aware decomposition. Exploiting these routing scheme patterns, however, is limited by the actual pattern inherent in the chosen routing scheme. For example, in a 3D toggling scheme between XYZ and its other 5 permutations in Figure 9(b), there is no pattern to exploit, and hence, this kind of dimensional bias based decomposition is not beneficial.



(a) **(b)**

Figure 9. Non dimension-ordered routing schemes (a) 2D West-first turn-based routing, (b) 3D dimension-based toggle routing

B. Physical Optimizations

One of the advantages of directionality-based routing-centric decomposition is that it lends itself to optimizations in certain directions. For example, in an XYZ-routing router, all decomposed logic related to Y going to X, Z to X, or Z to Y can be removed altogether. This results in dimensionally decomposed routers whose parts are dramatically more efficient than their composed counterparts. Compare Figure 7(c) with Figure 10. The latter shows the concept as it is applied to the decomposed switch arbitration logic on a router using XYZ routing.

In Figure 10, since the first stage of arbitration contains the intra-arbitration among the decomposed parts, the diagonal logic is superfluous and thus removed, as it represents requests for a port going to itself. Moreover, the directional optimizations mean that for an XYZ scheme, an X-port must only consider requests from the other X direction and from the core. Similarly, the Y-direction need only consider requests from X, the other Y direction, and the core. These observations can allow us to remove additional logic that is not needed, further reducing router footprint.

The same logic pruning optimization applies to the crossbar and virtual channel allocation decompositions. This optimization makes our proposed dimensional decomposition very appealing. Note that the redundant logic in the decomposition, however, will not be auto-optimized away by commercial synthesis tools, because at a router-level, synthesis is unaware that this logic will remain unused. This implies that all these optimizations must be hard-coded in the RTL code. Finally, note that on schemes that must support routing across all directions, this physical optimization reduces to only pruning of the self-sending directions.

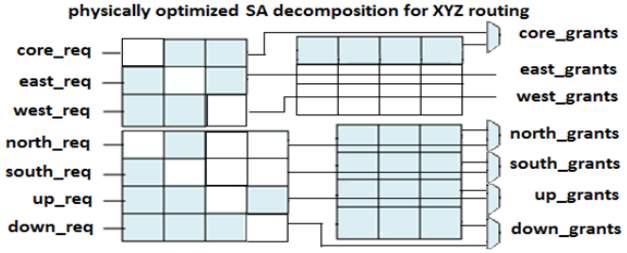


Figure 10. Decomposed SA arbitration with physical optimizations for XYZ routing. Compared to figure 7(c), the logic that will remain unused with XYZ routing has been removed, as shown by the white squares.

C. Efficient Decomposed Arbitration

While asynchronously decomposed crossbars, whose data passes through both stages within the same clock cycle do not change the cycle count (although they affect the clock period), decomposed arbitration does. One of the unanswered questions in decomposed arbitrations is how to efficiently reconcile the arbitration results of the two-stage arbitration scheme. The two inputs to arbitrate are the winner of the *preferred* direction and the winner of the *non-preferred* directions. This arbitration is represented by 2:1 muxes in Figures 7(c), 8(c) and 10. While matrix arbiters provide an efficient way to deal with fair arbitration of multiple inputs, given the amount of inputs to arbitrate, it may be overkill for this problem. Below are four novel simpler schemes that we propose to explore:

- ***Preferred-Takes-All:*** In this default scheme, the preferred direction always wins the arbitration.
- ***Preferred-biased***: Here, the preferred direction wins 66% of the contention times. This scheme tries to provide some relief to the non-preferred contender, to prevent complete starvation.
- ***Preference-Fair****:* In this scheme, when there is contention, half of the time the preferred direction wins, and the other half, the non-preferred directions win. This scheme tries to be fully fair among the preferred and non-preferred contenders. Note that as the first stage of the preferred direction only has 3 requestors (core and dimension A), and the first stage of the non-preferred direction has 4 requestors (dimensions B and C), this scheme still favors the preferred direction, but in a more fair manner. The net effect is that arbitration is slightly biased towards the preferred direction.
- ***Non-Preferred-biased****:* Here, the preferred direction only wins 33% of the contention times. The purpose of this scheme is to have a bias towards the non-preferred directions, to provide a contrast to the other approaches with preferred-direction biasing.

D. Directionality aware buffering

While buffer resources in NoC routers increase throughput and overall performance, they are a very limited resource due to their large power consumption. In a recent NoC power analysis study [19], almost 25% of power dissipated in a router is spent on the FIFO buffers. Hence sharing buffer resources among and within input ports has been proposed as a way to make power-efficient use of them. Among the many sharing permutations, sharing across two of the four directions in a 2D router was recently proposed after a study evaluated each of the permutations [20].

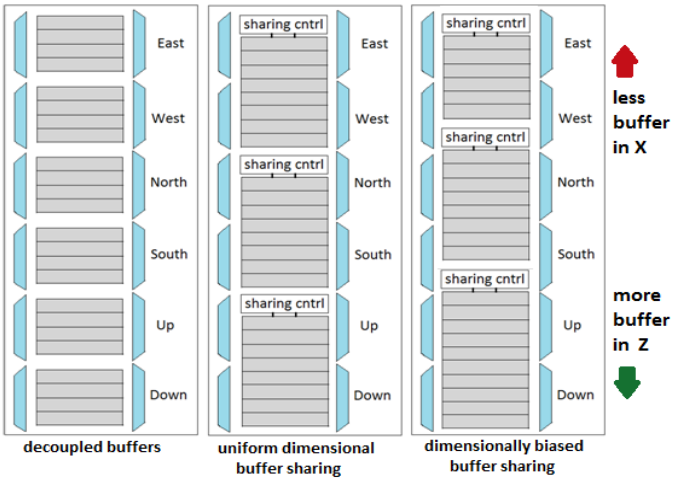


Figure 11. Buffer resource allocation in VC routers. (a) baseline resource allocation in VC routers is uniform and isolated (b) uniform dimensional buffer sharing (c) dimensionally-biased dimensional buffer sharing.

The *Roce-Bush* router attempts to share buffer resources while exploiting the dimensional bias of decomposed routers. We conjecture that if we can have two ports along a dimension sharing their buffer resources, and allocate these buffers unevenly among each of the directions, we can achieve better performance than a nominal decomposed router that uses a dimensionally-uniform buffer distribution. Figure 11 compares different possible FIFO buffering configurations. Figure 11(a) is the typical scheme where no buffer sharing occurs among ports. Figure 11(b) shows buffer sharing across two input ports along the same dimension. Figure 11(c) shows the same buffer sharing with uneven buffer allocation among each of the dimensional buffer pools. The motivation for dimensionally biasing buffer resources comes from the fact that in a dimensionally decomposed router, there is some inherent bias in the decomposed arbitration. Coupled with the bias on the buffer sharing, a fine-tuned router architecture that outperforms dimensionally oblivious schemes can be developed.

E. Putting it all together: The Roce-Bush 3D NoC router

Combining dimensional biasing, efficient decomposed arbitration and efficient buffer resource utilization leads us to the Roce-Bush router architecture, whose architecture is shown in Figure 12 (for an XYZ routing scheme). Note the optimal location of the decomposed crossbars, facing one another. The shared buffer pool in each direction can be lined up with the ports for an efficient physical implementation.

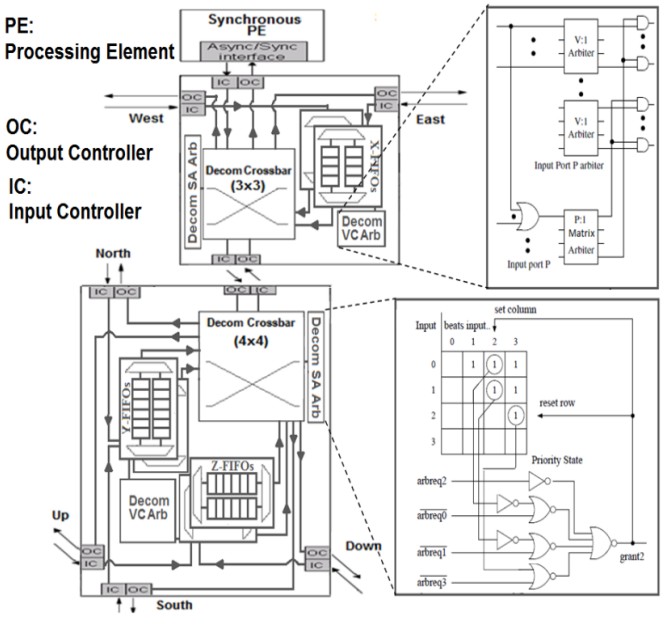


Figure 12. Proposed Roce-Bush router architecture (shown here with an implementation biased towards an XYZ routing scheme).

V. EXPERIMENTAL RESULTS

We implemented our Roce-Bush 3D NoC router at the RTL level using System Verilog, with simulation analysis using VCS and performed logic synthesis using Synopsys Design Compiler. The baseline code used was derived from the 2D NoC Netmaker library [20]. The following results are based on a 2 GHz single-cycle router synthesized using the standard 45nm Synopsys library. Unless otherwise noted, a router is a 7-port 3D router with 2 virtual channels of 4-flit buffers each, and a fixed packet length of 8 flits, running under a *Uniform Random* traffic pattern with a flit injection rate of 0.3 flits per router per cycle. Other simulated traffic patterns include *Neighbor*, *Hotspot-25* [25], *Bit-complement,* and *Tornado*. Each simulation result is based on a symmetrical 3 layer 4×4×4 (64 core) 3D network with 100,000 injected packets. Finally, the datapath/flit width across the routers is 64 bits. The results below measure latency in terms of the average cycle count taken by an 8-flit packet travelling across the NoC, as well as the worst-case cycle count for the slowest packet in the NoC. We measure the worst-case latencies on top of the average latencies because it gives a rough idea of the stability of the chosen design, an important aspect in embedded designs. For example, two designs might yield similar average latencies, but one of them might produce a much larger worst-case latency, which is undesirable for embedded designs.

Figure 13 shows the router performance (average and maximum flit latency) assuming XYZ, YXZ, and ZXY routing respectively, and after dimensionally decomposing a 3D NoC router across the three different dimensions, for five synthetic traffic patterns. *The results reveal a large performance advantage in decomposing the router in the first direction of the dimension-order routing.* This backs our conjecture that router decompositions biased based on the routing scheme can lead to improved performance. By decomposing in the first direction of the chosen dimension-order routing, our router helps reduce congestion. All other things equal, in a fully symmetrical network with an *ABC* dimension-ordered routing, our analysis

has shown that by choosing to minimize congestion on the first-dimension *A*, we allow more packets into the network, increasing the probability of higher throughput, and diminishing the probability of wasted cycles on the following dimensions, as the packet travels through the *B* and *C* downstream. If instead, we choose to route on *B*, then we ease some congestion on *B*, but without increasing the amount of packets coming into the network because *A* remains congested. To generalize from our results, *a dimensionally-decomposed router will have the highest performance if its dimensional bias aligns with the first direction of the dimensional routing scheme*, i.e., by giving preference to direction *A* in *ABC* routing.

Note that the larger the latency, whether average or maximum, the larger the spread among the decompositions. This makes intuitive sense, since for best-case traffic patterns such as *neighbor* and *tornado* for the XYZ or ZXY routing schemes, where there is very little congestion in the network (and thus latencies are low), routing-centricity during router decomposition, which aims to reduce congestion, does not have a major performance impact. However, as can be seen for the YXZ routing scheme, an inappropriate choice of decomposition direction bias (e.g., Z) can actually create congestion and drive down performance. In general, choosing a proper decomposition in the appropriate direction gives a larger performance gain when the congestion and latency increases, as can be seen for a majority of the traffic scenarios.

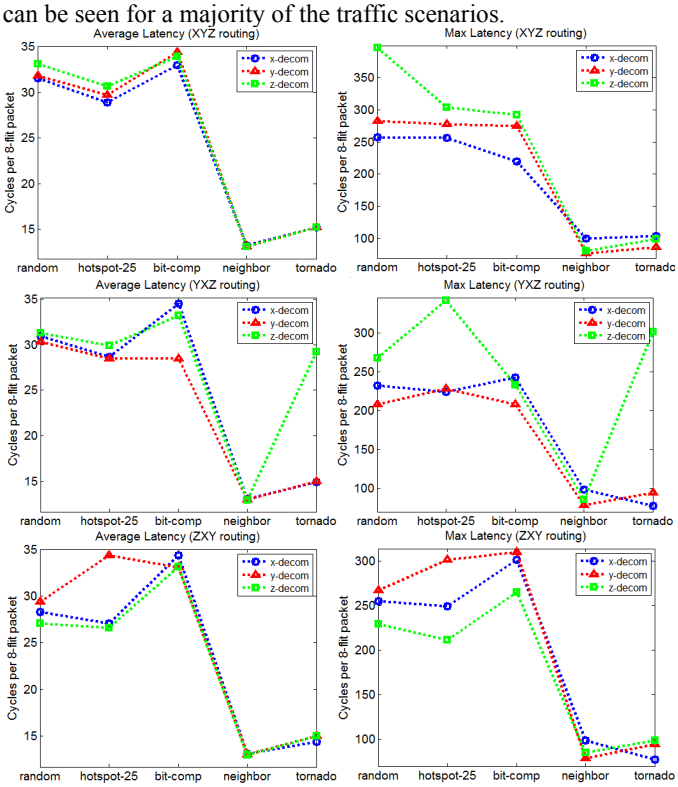


Figure 13. Performance comparison of three different decomposition configurations of the Roce-bush router for five synthetic traffic patterns under XYZ, YXZ and ZXY routing.

Figure 14 shows the performance of different arbitration techniques. In the figure, *pref-bias*, *fair*, *nonPref-bias*, and *Always-Pref* corresponds to the arbitration types "Preferred-bias", "Preference-fair", "Non-Preferred bias", and "Preferred-takes-all", respectively, described in section IV. The figure shows that using the simple preferred-takes-all scheme consistently results in the worst performance, by a significant margin. This is because by choosing a preferred takes all arbitration, we are starving the traffic in the non-preferred dimensions, which can lead to increased congestion along those dimensions due to a lack of load balancing. This is akin to having green lights only on east-west street lights in your drive home: even if your drive consists of mostly east or west roads, at some point you are likely to need to head south or north, and you will be stuck waiting, along with everyone else who was already waiting while you were going east or west.

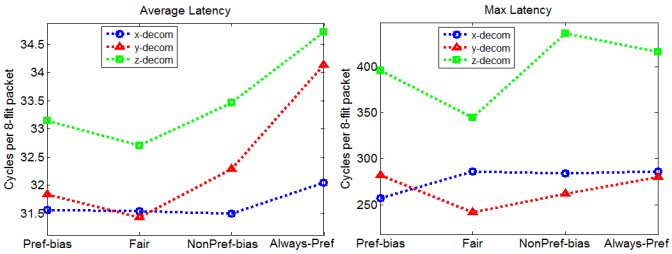


Figure 14. Performance comparison for different decomposed arbitration schemes in Roce-Bush, with XYZ routing. The implemented arbitration schemes (x-axis) correspond to preference-bias, preference-fair, non-preference-bias and preferred-takes-all.

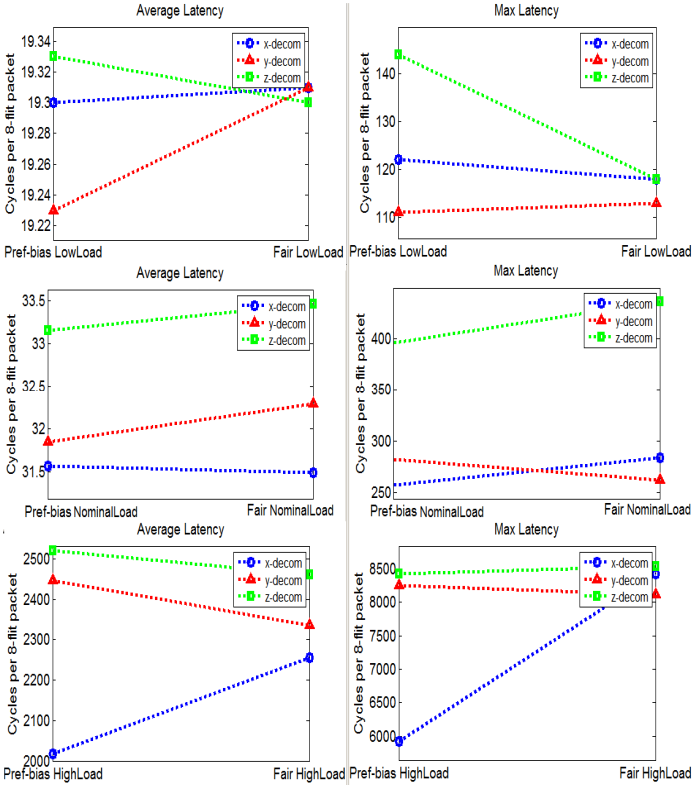


Figure 15. Performance of the three Roce-Bush decomposition configurations under different network loads (injection rates) for uniform random traffic and XYZ routing: Low load of 0.15 flits arriving at a router per cycle, Nominal load of 0.3 flits/cycle/router, and High load of 0.5 flits/cycle/router

On the other hand, a load-balancing arbitration, just as our regular street traffic lights, does a better job balancing the load and thus results in lower latencies. However, *this load balancing should not be direction agnostic. Our results show*

that a scheme that is slightly biased towards the preferred direction performs best. This is equivalent to the waiting time on a main street vs. a side street, where an optimal traffic light will give slightly more green time to the main street traffic, though it will not fully starve the side street traffic. Thus, we conclude that *by dimensionally biasing the arbitration towards the dimension we choose to prioritize (based on the directional bias of the chosen routing scheme), we get better performance.* Also, note that we consider the cases favoring the Y-decomposition over X to be within statistical noise due to the proximity of their results.

Figure 15 shows the latencies of each of the decompositions for the XYZ routing scheme under three different network loads for Uniform Random traffic: 0.15 flits/router/cycle (light loading), 0.3 flits/router/cycle (nominal loading), and 0.5 flits/route/cycle (heavy loading), under two different arbitration schemes (*Pref-bias* and *Fair*). In the chosen case of XYZ routing, the Roce-Bush router decomposes with X-direction bias. From the figure, we can see that the *Roce-Bush* router can attain an up to 25% increase in performance under high loads for the XYZ routing scheme. Note that as the load of the network increase, congestion goes up, and as discussed previously, this is where the decomposition used matters most, hence resulting in a much better performance for a *Roce-Bush* router over a statically decomposed router. Finally, although not shown for brevity, similar performance improvements were obtained in experiments with the Roce-Bush router when a YXZ or YZX routing scheme was chosen and a Y-direction bias was employed, and when a ZYX or ZXY routing scheme was chosen and a Z-direction bias was employed.

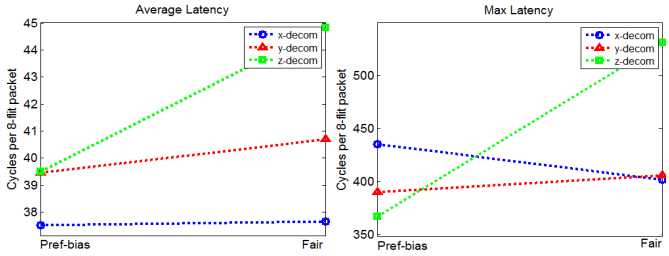


Figure 16. Performance comparison of different decomposition configurations in Roce-Bush for 3D west-first routing scheme.

The Roce-Bush router provides routing-centric performance optimization that can extend beyond simple dimension-ordered routing schemes. Figure 16 shows a comparison of different dimensional decompositions in Roce-Bush for a turn model-based 3D west-first routing algorithm, where the XY plane (typical 2D plane implementation) forbids north-west and south-west turns, the YZ plane forbids up-south and down-south turns, and finally, the ZX plane forbids up-west, and down-west turns. This routing scheme always puts more pressure on the X direction (as implied by its name all flits must go west-first) hence a Roce-Bush optimization with an X-direction decomposition bias yields roughly a 20% increase in performance (average latency) as compared to the static Z-decomposed router on an average loaded network. Note that the Y-preferred design has a slightly lower latency than the Z-preferred design, because the 3D west-first routing first goes west, then it alternates among the Y directions, and then among Z, thus making it more efficient to bias along the Y over the Z.

Figure 17 shows the results of biasing the amount of buffer resources in each of the directions, in a Roce-Bush implementation with an XYZ routing scheme and Uniform Random (nominal injection rate) traffic. Notice that in XYZ routing, regardless of each of the direction of the decomposition chosen or the arbitration scheme, the latency is minimized by adding more buffer resources to either the Y or Z dimensions. This makes sense because these two dimensions are further along the packet's travel where there is a higher chance of seeing HOL blocking.

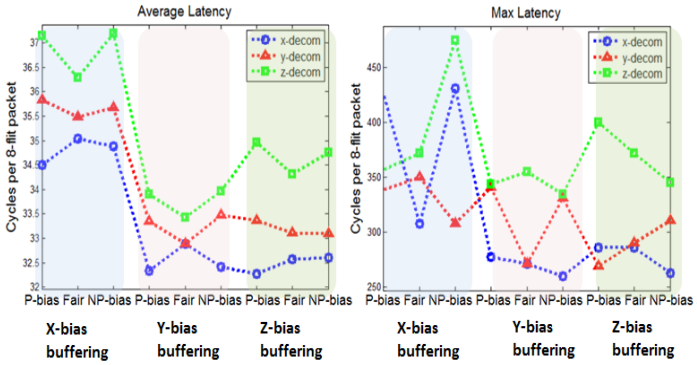


Figure 17. Performance comparison with buffer biasing for each of the dimensional decompositions in Roce-Bush for XYZ routing

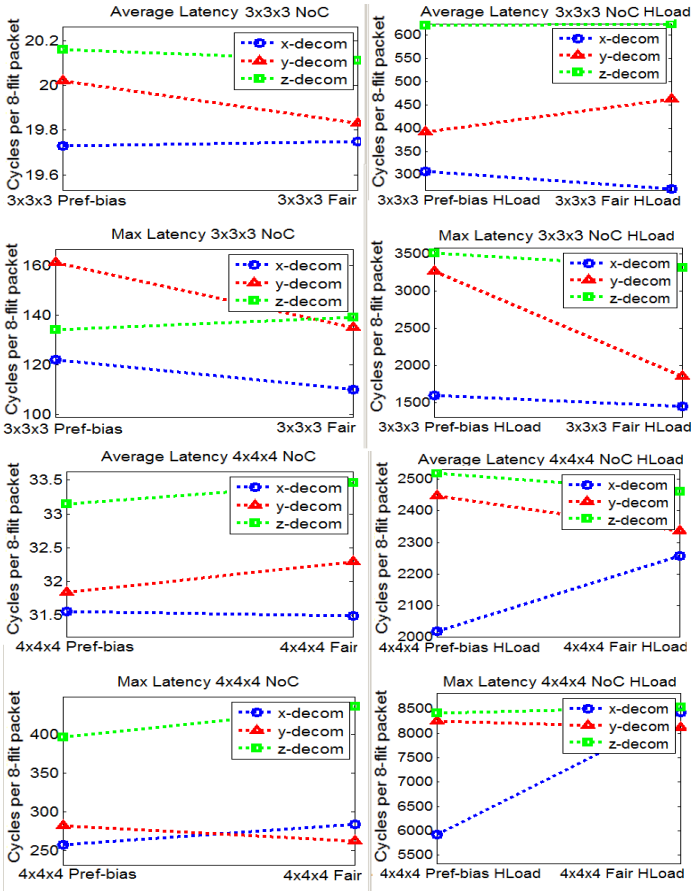


Figure 18. Performance comparison between different dimensional decompositions in Roce-Bush under different network sizes using a random traffic pattern.

Figure 18 shows the promising future of this optimization as we increase the number of cores in the die. On a nominal load, the results show that a 4x4x4 (64-core) network has *3 times the performance increase with respect to different directional decompositions* of a 3x3x3 (27-core) network, where in the 27 core network there is a 0.5 cycle improvement per packet, while in the 64 core network, there is roughly 1.5 cycles improvement. Note that both networks show much higher performance gain using the Roce-Bush router under high-load conditions as expected.

Finally, Figure 19 provides a comprehensive comparison of the *Roce-Bush* router against various state-of-the-art 3D NoC routers for different traffic patterns, for the XYZ routing scheme and a 4x4x4 3D NoC, using Prime Power to extract power estimates. The routers compared are: 3D Composed VC router based on the VC router of [12], a 3D speculative router of [13], a 3D static Z-decomposed hierarchical router [4], and the proposed Roce-Bush router. The *speculative router* extension to 3D results in the highest power dissipation and does not result in a decrease in latency when compared to a state-of-the-art router that uses a minimal amount of virtual channels. A single cycle *composed router* with virtual channels uses the second largest power, but achieves very low cycles to completion. By decomposing the router on the Z-direction, the *static Z-decomposed hierarchical router* takes a performance penalty but has lower power dissipation. However, using a strategic decomposition and buffer allocation using our *Roce-Bush router*, we beat the cycle count of the composed router while also using lower power than all other routers. Our results show power savings of 25% against the speculative router, 12.5% against the composed router and 5% over the statically z-decomposed router, as well as a latency reduction of 39% against the speculative router, 3% against a composed router, and 14% against a statically Z-decomposed hierarchical router.

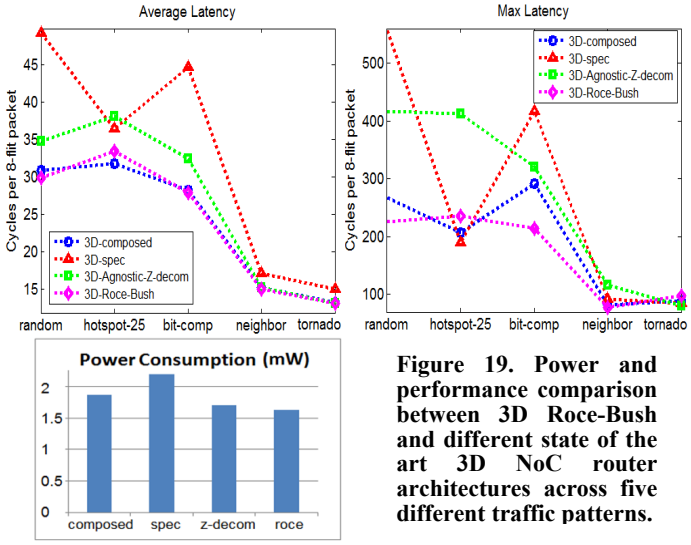


Figure 19. Power and performance comparison between 3D Roce-Bush and different state of the art 3D NoC router architectures across five different traffic patterns.

In summary, based on our experimental results, we have shown that our novel routing-centric decomposed *Roce-Bush* 3D NoC router can achieve significant performance gains while still staying below the power envelope of other state-of-the-art 3D NoC routers.

VI. CONCLUSION

This paper presented a novel dimensionally decomposed 3D router for NoCs, with the ability to exploit inherent dimension biases in routing schemes used with 3D CMPs. The Roce-Bush router highlights the important concept of routing-centric dimensionality awareness in 3D NoC router design by exploiting it in terms of resource decomposition and buffer allocation. While in 2D NoC routers, dimensional decomposition was excused due to the decomposition overhead, in 3D routers, it becomes a necessity. However, as shown in our experimental results, statically decomposed routers that are agnostic to the chosen routing scheme may not achieve desired results. As we migrate from conventional 2D ICs to multilayer CMPs, exploiting the decoupling of dimensional patterns in routers based on routing-centricity will become a handy technique for designers to efficiently tailor to fit more stringent performance needs. Our experimental comparisons for RTL level implementations of Roce-Bush with various state-of-the-art 3D NoC routers confirm this observation.

VII. REFERENCES

[1] L. Benini et al., "Networks on Chip: A New SoC Paradigm," Proc. Computer, 49(1):70-71, Jan 2002.

[2] C. Nicopoulos et al., "Network-on-Chip Architectures: A Holistic Design Exploration", ISBN-9048130301, 2009.

[3] J. Lau, "Advanced MEMS Packaging", ISBN-10: 0071626239, chapter 2, pp 50, 2009.

[4] J. Kim, "A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architectures", Proceedings of the 34th annual international symposium on Computer architecture (ISCA), 2007.

[5] W. Lafi, "An efficient hierarchical router for large 3D NoCs", 21st IEEE International Symposium on Rapid System Prototyping (RSP), 2010.

[6] F. Darve, "Physical Implementation of an Asynchronous 3D-NoC Router using Serial Vertical Links", Proc. IEEE ISVLSI, 2011.

[7] D. Park, "MIRA: A Multi-layered On-Chip Interconnect Router Architecture", Proc. ISCA, 2008.

[8] E. Marinissen, "Testing 3D chips containing through-silicon vias", International Test Conference, 2009.

[9] A. Banerjee, "A Power and Energy Exploration of Network-on-Chip Architectures", Proceedings of the First International Symposium on Networks-on-Chip (NOCS), 2007.

[10] A. G. Wassal, "Novel 3D memory-centric NoC architecture for transaction-based SoC applications", Saudi International Electronics, Communications and Photonics Conference (SIECPC), 2011.

[11] N. Choudhary, "Routing Centric NoC Design for High Performance Multimedia Application", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-1, Issue-5, 2011.

[12] R. Mullins, "Low-Latency Virtual-Channel Routers for On-Chip Networks", Proc. ISCA, 2004.

[13] R. Mullins, "The Design and Implementation of a Low-Latency On-Chip Network", Asia and South Pacific Conference on Design Automation, 2006.

[14] W. J. Dally and B. P. Towles, "Principles and Practices of Interconnection Networks", ISBN 0-12-200751-4, 2004.

[15] C. Glass, "The Turn Model for Adaptive Routing", The 19th Annual International Symposium on Computer Architecture, 1992.

[16] H. Zhu, "Performance Evaluation of Adaptive Routing Algorithms for achieving Fault Tolerance in NoC Fabrics", IEEE International Conf. on Application -specific Systems, Architectures and Processors, 2007.

[17] R. Manevich, “A Cost Effective Centralized Adaptive Routing for Networks-on-Chip”, 14th Euromicro Conference on Digital System Design (DSD), 2011.
[18] K. Shim, “Static virtual channel allocation in oblivious routing”, 3rd ACM/IEEE International Symposium on Networks-on-Chip (NoCs), 2009.
[19] A. B. Kahng, “ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration”, Conference on Design, Automation and Test in Europe (DATE), 2009.
[20] K. Latif, “PVS-NoC: Partial Virtual Channel Sharing NoC Architecture”, 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2011
[21] R. Mullins, “Netmaker – 2D NoC simulation and synthesis library”, http://www-dyn.cl.cam.ac.uk/~rdm34/wiki/index.php, 2009.
[22] S. Y. Nam, “Decomposed crossbar switches with multiple input and output buffers”, Global Telecommunications Conference, 2001
[23] Tilera Corp., “Tilera TILE-Gx 3000”, http://www.tilera.com/, 2011.
[24] F. Darve et al., “3D Embedded multi-core: Some perspectives”, Proc. DATE, 2011.
[25] A. Rahmani et al., “Negative Exponential Distribution Traffic Pattern for Power/Performance Analysis of Network on Chips”, International conference on VLSI design, pp 157-162, Jan 2009.