Design of a Concentrated Torus Topology with Channel Buffers and Efficient Crossbars in NoCs

Dominic DiTomaso†, Randy Morris†, Evan Jolley†, Ashwini Sarathy‡, Ahmed Louri‡,
and Avinash Kodi†
†Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701
‡Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
dd292006@ohio.edu, kodi@ohio.edu

Abstract

Excess power dissipation along with increased leakage currents in router buffers and crossbars are becoming a major constraint that is affecting the performance of Network-on-Chips (NoCs) architectures. In this paper, we design channel buffers and router crossbars in a concentrated torus topology (CTorus) which is a dual network without the additional area overhead. When compared to other dual networks, CTorus improves saturation throughput by 11-20% for synthetic traffic and improves speedup by 1.78-2.15X for real benchmark traces such as PARSEC and SPEC CPU2006. When the energy-efficient buffer and crossbar organization was inserted into our CTorus topology, we reduced energy dissipation by 32% and area by 53% on average over mesh2X, CMesh2X and FBfly2X.

1 Introduction

Network-on-Chips (NoCs) [2, 5] design paradigm overcomes the dual problem of global wire delay and scalability in Chip Multiprocessors (CMPs) by (i) matching or reducing the wire lengths to network topology and (ii) increasing the bandwidth with more links and switches. As NoCs architecture (combination of *links* for communication and *routers* for storage and switching) gains traction with increasing number of cores on a chip, power dissipation combined with excess leakage currents is already a major technology constraint which affects both performance (throughput and latency) and area overhead. While the previous design of 80-core Intel TeraFlops consumed more than 28% of the total chip power [9], more recent 48-core Intel SCC design [4] reduced the overall communication power to 10% of the total power budget by implementing several power optimization techniques. Clearly, energy-efficient and high-performance NoCs architectures are required to sustain and continue the performance gains achieved by increasing the number of cores on a single chip with every successive generation.

Of the several research directions that improves the energy-efficiency and performance in NoCs [16], we focus on three critical inter-related components which are: (a) buffering, (b) switching and (c) topology. As buffers consume a substantial router power, several techniques to minimize the impact of the router buffers have been proposed including (i) replacing the repeaters along the link to duplicate as hold and store (channel buffers) when desired [13] and (ii) replacing all buffers with elastic buffers along the link by replacing repeaters with flip-flops and implementing a handshaking protocol between buffers [14]. Crossbars have been the subject of evaluation for NoCs and researchers have proposed smaller, segmented and split crossbars for improved energy and area-efficiency [6]. Lastly, there have been several topologies that have improved throughput, latency while reducing power. Concentrating cores has shown to be an effective way to maximize the performance by trading off serialization latency for higher radix routers [1]. Flattened Butterfly (FBfly) is another high-radix NoC router architecture which reduces any extra hops along a dimension, thereby restricting the diameter of the network to two at the cost of increased router radix [11]. While prior work has shown the performance benefits of reducing hop count with topologies, there has not been an integrated evaluation that takes router optimizations (buffers and crossbars) into topology evaluation while overcoming NoC performance limitations such as Head-of-Line (HoL) blocking and router complexity (power and area overhead).

In this paper, we propose an integrated NoC architecture with channel buffers and router crossbars on a concentrated Torus (CTorus) with the goals of minimizing power consumption, reducing HoL blocking, and further improving network performance. While channel buffers provide power savings, HoL blocking is not alleviated as the packet

978-0-7695-4979-8/13 $26.00 © 2013 IEEE
DOI 10.1109/IPDPSW.2013.273

at the head of the router can block subsequent packets. We propose a dual channel (dc) configuration that increases the number of inputs and provides speedup without the inputs blocking each other. Further, to take advantage of dual inputs into the router, we re-design the monolithic crossbar into multiple crossbar organizations (mx) along the four quadrants. The multi-crossbar design shows the twin objectives of saving power with smaller crossbars along with increasing the throughput/performance with adaptive routing techniques. Finally, we incorporate the energy-efficient channel buffers (dc) and crossbars (mx) on a concentrated Torus (CTorus) topology. The implementation of the dual channel dc and mx organizations has the performance of a dual network without the additional area overhead. Therefore, we compare CTorus to the following dual networks: mesh2X, CMesh2X, and FBfly2X. We used the Synopsys Design Compiler to evaluate the power, area and router pipeline latencies for various configurations. Our results indicate that the router pipeline to be within the design tolerances for 2 *Ghz* router clock at 1.0 *V* and consuming 25% to 40% lesser power. CTorus improves saturation throughput by 11-20% for synthetic traffic and improves speedup by 1.78-2.15X for real benchmark traces such as PARSEC and SPEC CPU2006. Moreover, the proposed CTorus topology shows up to 56% power savings and occupies approximately 47% to 64% lesser area while improving energy-delay product (EDP) from 29% to 37% over CMesh2X and FBfly2X topologies. The major contributions of this work are as follows: (1) We utilize an adaptive channel buffer design along with a channel buffer organization that alleviates Head-of-Line (HoL) blocking, thereby preventing performance degradation without duplicating networks. (2) We show the design of a multi-crossbar that can take advantage of the speed-up offered while maximizing the port occupancy and improving performance with minimal adaptive routing. (3) We evaluate the proposed buffer and crossbar organizations on synthetic and real applications (PARSEC [3] and SPEC CPU2006 [8] benchmarks) showing a performance improvement of 10-25%, power savings of 25-40% with an area overhead of 5-13%. Using the best of channel buffer and multiple crossbar organizations in a CTorus topology, we show an average saturation throughput improvement of approximately 17%, an average power reduction of 32%, and an average total area reduction of 53% when compared to other NoC topology such as CMesh2X and FBfly2X.

2 Related Work

Dynamic VC allocation improved performance for both short and long packets, however the table-based control increases the complexity due to a large number of VCs. iDEAL design combined channel buffers with router buffers to reduce power consumption and area overhead, however HoL blocking was not addressed in iDEAL design [13]. Elastic channel buffer solved the HoL blocking with duplicate sub-networks and provided design solutions for both Flattened Butterfly and mesh interconnection networks. Duplicate networks is an effective solution for HoL problem, however elastic channel buffer (ECB) do not discuss the crossbar organizations. Moreover, ECB save power and area, but do not address performance limitations due to channel buffers [14]. Bufferless networks (FlitBLESS [15] and SCARAB [7]) reduce power consumption with no buffers by deflecting/dropping conflicting packets. These networks will need high-speed route computation logic to determine which inputs should obtain the preferred direction and which ones should be deflected or dropped, leading to increase in complexity and stringent timing problems. Moreover, these networks are not designed for high network load due to excessive power consumption with extra deflections or dropping.

Low-radix switch organizations have been extensively analyzed starting with RoCo [12] which restricts the direction (row or column) and limit the radix (simple two, 2×2), leading to significant area and power savings. Our approach requires more crossbars, however, with careful combining and restrictive VC allocation, we can obtain performance similar to that of 4VC routers. Low-cost approach reduces the radix and splits into row and column, thereby reducing the crossbar complexity for ring networks [10].

3 Adaptive Channel Buffers and Multi-Crossbar

In this section, we detail the implementation of the dual-function links and the associated control logic. A single stage of the three-state repeaters, shown in the inset of Figure 1(a), comprises of a three-state repeater inserted segment along all the wires in the link. When the control input to a repeater stage is low, the three-state repeaters in that stage function like the conventional repeaters transmitting data. When the control input to the repeater stage is high, the repeaters in that stage are tri-stated and hold the data bit in position. The adaptive dual-function links hence enable a decrease in the number of buffers within the router and saves appreciable power and area. The design requires a single control block per inter-router link in order to control all the repeater stages along the link, unlike the design in [14] which uses one control block per stage along the link. Therefore, our proposed control technique is power-efficient and has a lesser area overhead compared to the design in [14].

In addition to the low-power, area-efficient implementation and the independent control of each repeater stage, the control block, shown in Figure 1(b), provides the following

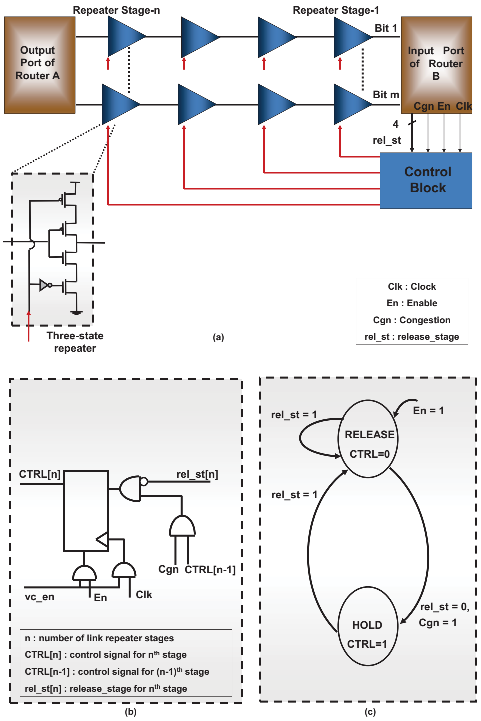

Figure 1: (a) A link using three-state repeaters that function as channel buffers during congestion, (b) Control block implementation details and (c) State transition diagram.

advantages over the control block design in [14]: (1) The design in [14] requires a careful monitoring of three different clock skews for the correct operation of the circuit. In our design, the *clock-to-q delay* of the flip-flop does not limit the correct operation of the three-state repeaters. The control signal may arrive at a repeater stage at any time prior to the next clock cycle, as the data that is to be held will be lost or overwritten only at the next clock edge. This provides a more relaxed timing requirement, making our technique more preferable and easier to implement. (2) Unlike the design in [14] which uses two acknowledgement signals in addition to two control signals, our proposed technique employs only one control signal per stage making it more scalable and simpler to implement, along with significantly less power and area overhead. Figure 1(c) shows the state diagram for one stage within the control block. The router can then request that the control block release any given repeater stage, by setting the corresponding bit in the 'rel_stage' signal.

3.1 Dual Channel Buffer Organization

Figure 2 shows the dual-channel buffer configuration. In this configuration, we duplicate the channel buffers to avoid HoL blocking as shown. Each channel has a dedicated input port (register) at the downstream router to read the flit

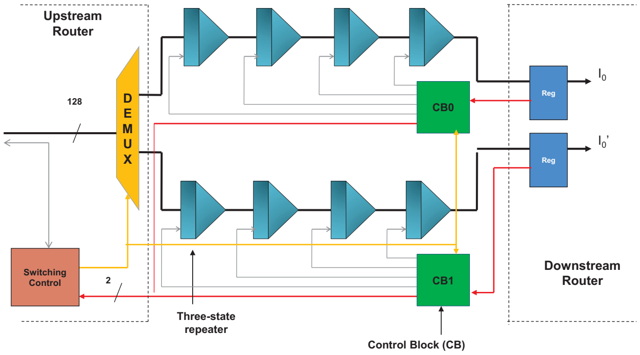

Figure 2: Dual channel (dc) buffer organization.

before it will be written into the crossbar. The two inputs are shown as I_0 and $I_0^{'}$. When the flit is read into the register, it activates the control block (CB0) or (CB1) to indicate a full register. As explained before, the control block will then hold flits one cycle after another into different channel buffers associated with the particular control block. To ensure that the channel buffers are ready to store the flit, the DEMUX information is also transmitted to the control block to indicate that a flit will be arriving via the vc_{en} signal. When all the channel buffers are occupied, it will then signal the upstream switching control to indicate a full channel or congestion.

The flit read into the register undergoes the standard router pipeline stages of RC (route computation), VC (virtual channel) allocation, SA (switch allocation) and then switch traversal (ST), before moving on to link traversal (LT). Here, we combine RC and VC into a single stage, giving us a 4-stage router pipeline. Look-ahead routing can be employed for deterministic routing while adaptive routing schemes require RC to be computed to find the best downstream router. Prior elastic buffer designs have eliminated

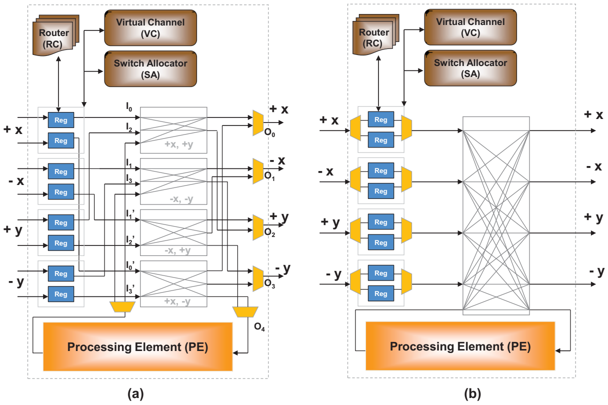

Figure 3: (a) Multi Crossbar (mx) Organization and (b) Baseline Crossbar Organization.

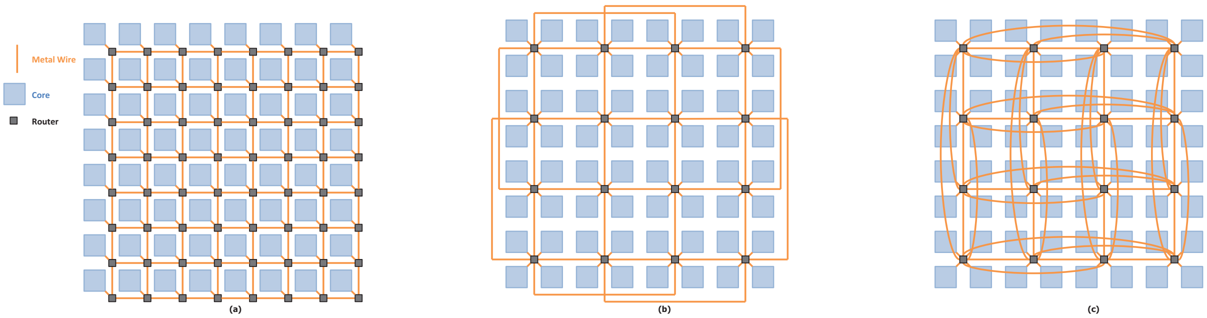

Figure 4: (a) Mesh, (b) Concentrated Mesh (CMesh), and (c) Flattened Butterfly (FBFly) topologies.

the VC stage, simplifying the channel buffer design and reducing the router pipeline. However, we retain the VC stage as we have two channel buffer links to choose from. Moreover, this provides an opportunity to have different classes of service for different packets. We do not include speculative switch allocation as this increases switch allocation complexity (latency) while consuming more power. Once, the flit is in the ST stage, we transmit the VC allocation information (0 or 1 as there are 2 VCs) along with the flit to the switching control to set the DEMUX to the appropriate channel buffer link. When all the channel buffers are occupied for a particular VC, the switching control will deactivate the channel buffer from receiving any more flits until the control block releases the congestion. The dual channel buffer organization reduces the HoL blocking, providing differentiated classes of service while also ensuring sufficient buffering to improve the throughput.

3.2 Multi-Crossbar Organization

Figure 3(a) shows the multi-crossbar organization which splits the crossbar into 4 smaller crossbars to reduce area and power consumption and Figure 3(b) shows the baseline design with 2VCs. The division of the 4 crossbars are along the 4 quadrants: (+x, +y) [North-East], (-x, -y) [South-West], (-x, +y) [North-West] and (+x, -y) [South-East]. The four quadrants represent the four directions with dedicated channels such that adaptive packet flow can be implemented. Communication along the quadrant chooses the crossbar designed for the direction. Suppose, the packet arrives from +x direction into I_0. This packet can be routed to either O_0 (+x direction) or O_2 (+y direction) using the North-East crossbar. Similarly, if the packet arrives from +x direction from $I_0^{'}$ direction into the South-East crossbar, then the possible outgoing directions will be O_0 and O_3. The VC allocation is based on how many hops away the packet is from the destination. If the packet is more than one hop away from the destination in either dimensions, then the packet can be allocated to either VC. If the packet is exactly one hop away from the destination in a particular dimension, then always the lower VC should be allocated. With this simple restriction, we can use both the VCs and connect using different crossbars to get to the same direction. The availability of VC guarantees that the load maybe lower in the specified direction. This also allows the packet to be adaptively routed along the minimal dimension. The packet will always select the route that offers the VC. In case of low loads when the VC may become available in both dimensions, the packet will then randomly choose the direction. Deadlocks are avoided naturally as there are always two VCs available. Further as the packets traverse specific quadrants (+x, +y), (-x, -y), (-x, +y) and (+x, -y) to reach the destination, there are no circular dependencies that could potentially lead to deadlocks. The packets progress always in the forward direction towards the destination and never return along the same path. Therefore, the multi-crossbar configuration provides the best of the three worlds - lower area due to split crossbars, lower power dissipation due to shorter path lengths and higher throughput due to selective merging of different output ports.

4 Topology

Some leading topologies for NoCs include mesh, Concentrated Mesh (CMesh), and Flattened Butterfly (FBfly), as shown in Figure 4. The mesh network topology has a router at each processing core. The routers are connected in a grid fashion where each router is connected to four neighboring routers. Each router, except those around the edges of the grid, has one input and output port for the cores as well as four ports for the four directions: +x, -x, +y, and -y. The mesh topology allows quick communication between neighboring cores, however, there is a high hop count which increases the network diameter. The CMesh topology has four cores concentrated to one router. The routers are also connected in a grid fashion, however, there are extra links around the edges which skip over one router. These links are added so that CMesh and mesh have the same bisectional bandwidth. The CMesh routers have four ports for

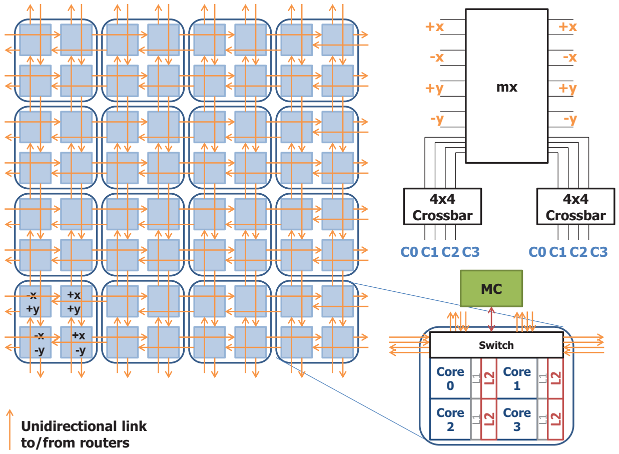

Figure 5: CTorus Topology and router design using the mx crossbar organization.

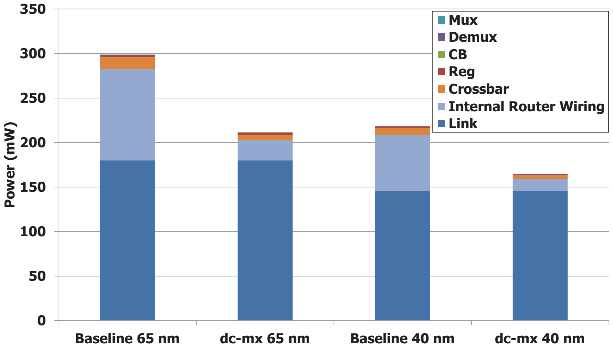

Figure 6: Power of baseline and dc-mx organizations at 65 nm and 40 nm.

the four cores as well as four ports for the four cardinal directions. CMesh offers a lower hop count allowing lower packet latency. However, the multiple cores connected to the same router may cause contention as packets enter and leave the same ports. The FBfly topology also uses a concentration of four cores, although, routers in the same x and y dimension are fully connected. This further reduces the hop count of the network. However, the router area does not scale well since there are four ports for the cores and ports for each of the other routers in the x and y dimensions. In addition to this high radix router, the cost in wires increases area and power dissipation.

We propose a CTorus topology using the dual channel (dc) buffer organization and multi-crossbar (mx) organization. The CTorus topology balances the traffic load better than a mesh due to wrap-around links allowing packets to travel in both directions, thereby reducing the traffic contention at the center of the network. Concentration of the cores provides the added advantage of reduced hop count thereby leading to savings in power and area overhead. Moreover, due to the reduced crossbar complexity, we can further reduce the router complexity when compared to FBfly topology. The CTorus topology is shown in Figure 5 and uses a concentration of four cores. The arrows around the edges of the topology are links which wrap around the opposite edge and were shown like this for simplicity. Each router has four inputs and outputs for each of the four directions: +x, -x, +y, and -y. Since the dual channel buffer organization is used, each router has two links for each direction. To accommodate for the concentration of four cores, the mx crossbar organization changes slightly. Instead of 2 to 1 multiplexers and demultiplexers at the cores, we must use two 4 × 4 crossbars at the cores as shown in Figure 5. The figure also shows the logical connection between the cores. Each core in the concentration is given one of four quadrants: (+x, +y) [North-East], (-x, -y) [South-West], (-x, +y) [North-West] and (+x, -y) [South-East]. CTorus represents a dual network in that it has two redundant links between routers. Dual networks are created by duplicating the NoC routers and links so that packets have more resources. For this reason, we compare CTorus to mesh2X, CMesh2X, and FBfly2X which duplicate routers and links. The inset of Figure 5 shows how the cores, caches, and memory controllers (MC) are connected. Each core has a private L1 and private L2 cache. Each L2 cache is connected to the switch. From the switch, communication can go to the MC or to other core routers.

5 Performance Evaluation

In this section, we evaluate our proposed channel buffer and router crossbar organization in terms of power dissipation, area overhead and overall network performance. We consider each router with a 4-stage router pipeline. Each packet consists of 4 flits where each flit is 128 bits for a total of 512 bits per packet. The dc buffer and mx organization were synthesized and optimized using the Synopsys Design Compiler tool using the TSMC 65 nm and 40 nm technology libraries with a nominal supply voltage of 1.0 V and an operating frequency of 2 GHz. We also evaluate our CTorus topology and compare to mesh2X, Concentrated mesh2X (CMesh2X), and Flattened Butterfly2X (FBfly2X). For a fair comparison, every architecture uses channel buffers. The mx crossbar is implemented only in the CTorus design because it cannot be implemented in high radix routers such as FBfly2X. For equal comparison, the bisectional bandwidth was maintained equal for all designs by adjusting the link widths.

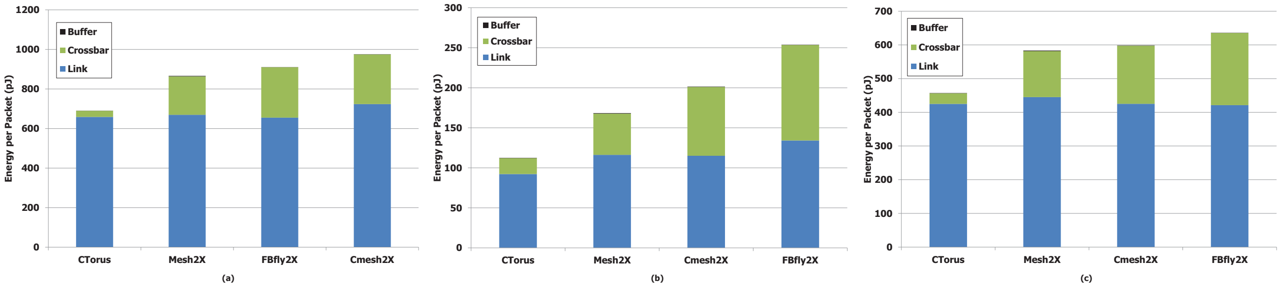

Figure 7: Energy per packet for different traffic loads. (a) Complement (b) Butterfly, and (c) Uniform Random.

5.1 Power, Timing and Area Estimation

The power per segment of the repeater-inserted link is given by, $P_{segment}$ = $P_{dynamic}$ + $P_{leakage}$ + $P_{short-ckt}$ where $P_{dynamic}$ is the switching power, $P_{leakage}$ is the power due to the subthreshold leakage current and $P_{short-ckt}$ is the power due to the short-circuit current. The power per segment is multiplied by the number of segments and the link width to obtain the total link power dissipation for a flit traversal. When a conventional repeater is replaced by a three-state repeater, there is an additional capacitance due to the added transistors. The increase in the switching capacitance increases the total power consumed by the links. Power is also dissipated in the control blocks controlling the dual-function repeater stages, when they are enabled during congestion. In calculating the power values, the inter-router links are assumed to be 5 *mm* long for the concentrated networks.

5.1.1 Power

Figure 6 shows how the power changes with different technology nodes of 65 nm and 40 nm. The total power dissipation of the dc-mx organization is 164.8 mW 40 nm and is due to: (i) dc - the link (145.39 mW), register (1.74 mW), control block (0.054 mW) and demux (0.077 mW) and (ii) mx - the crossbar switching power (3.38 mW), the VA arbiter (0.298 mW), the SA arbiter (0.0908 mW), and the additional wiring between the inputs/outputs and the crossbar switches (13.8 mW). The baseline power dissipation is 232.22 mW for a 29% improvement. Additionally, the total leakage power of the dc-mx design was found to be 1.525 μW and the baseline has a leakage power of 1.659 μW.

For topology, Figure 7 shows the energy per packet for a certain traffic pattern for the CTorus, mesh2X, CMesh2X and FBfly2X topologies. The energy is broken down to link, crossbar, and buffer energy dissipation. The three traffic patterns shown are Complement, Butterfly, and Uniform Random. For each load, the link energy per packet is similar across all topologies. This is because the distance a packet must travel from source to destination is independent of topology. Each link consumes 6.65 pJ/mm for a 128 bit link in 65 nm technology. In each traffic pattern, CTorus has a lower total energy dissipation per packet. This savings is due to the smaller crossbars used and long wrap around links which skip over intermediate routers. For example, CTorus uses one 3 × 3 crossbar at intermediate routers and one 3 × 3 crossbar plus one 4 × 4 crossbar when the packet is at the source and destination. This corresponds to a crossbar power of 7.0 pJ per packet at intermediate routers and 17.4 pJ at the source and destination. The energy dissipation for an equivalent 8 × 8 crossbar is 28.44 pJ. Therefore, a four hop packet in CTorus saves 86.3 pJ in crossbar traversals compared to the 8 × 8 crossbar in CMesh2X.

Figure 8 shows the average energy-delay product (EDP) per packet. The EDP allows us to analyze how both the latency and power effect each network. Since each topology used the same number of bits and clock frequency, the power and energy are directly related for each topology. The results shown are normalized to the mesh2X topology. Mesh2X has a high EDP for some cases but not all with an average EDP 30% higher than CTorus and an average EDP 18% lower than FBfly2X. This is due to the large network diameter causing high latency and high power of mesh2X. CTorus has the same network diameter as CMesh2X, however, the low power from the crossbar design allows CTorus to have a lower EDP for many cases with an average of 29% less than CMesh2X. For Complement, CTorus has an EDP 38% less than CMesh2X.

5.1.2 Timing

The latency for the dc design was found to be 0.31 ns in 45 nm technology. This latency was due to the channel buffer latency of 0.17 ns, the register buffering of 0.07 ns, and the demux latency of 0.07 ns for a total of 0.31 ns which is within our specified clock period of 0.50 ns. The critical latency for the mx crossbar is 0.24 ns and the latency for the baseline was 0.22 ns. These were due to the critical path of the logic in the VA stage.

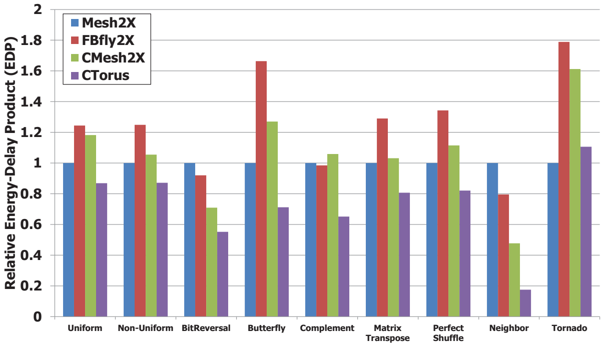

Figure 8: Relative Energy-Delay Product (EDP) of 64 core topologies for all synthetic traffic.

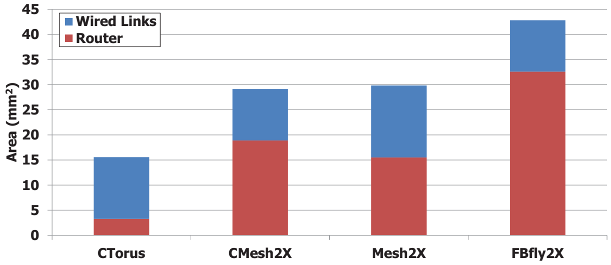

Figure 9: Router and link area overhead of each topology.

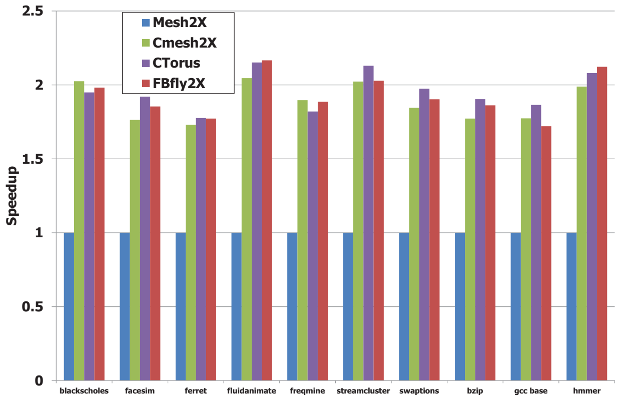

Figure 10: Speedup of 64 core topologies for SPEC2006 and PARSEC benchmarks.

5.1.3 Area

Area overhead of the baseline 4VC router obtained from Synopsys is 0.283 mm^2 which includes the buffer and crossbar. The dc-mx design area is 0.295 mm^2 which is slightly more compared to the baseline due to the increase in link width. For topology, Figure 9 shows the router and link area overhead of each topology. The torus saves approximately 47% area over CMesh2X, 48% area over mesh2X, and 64% area over FBfly2X. The 8 × 8 crossbar used in CMesh2X was estimated from Synopsys to have an area of 0.590 mm^2. Since CMesh2X is a dual network, the network requires two 8 × 8 crossbar. Using four 3 × 3 crossbars and two 4 × 4 crossbars in CTorus reduces the crossbar area overhead by 83%. This large savings is due to the dual links in the dc buffer design and the mx crossbar which creates a dual network without the overhead of doubling router and link components. With equal bisectional bandwidth, each topology has similar link area overheads. Each 128-bit link occupies 0.0256 mm^2 for every 1 mm length, estimated from Synopsys.

5.2 CTorus Results

A cycle-accurate on-chip network simulator was used to conduct a detailed evaluation of the proposed channel buffer and router crossbar designs in a 8 × 8 mesh network. For open-loop measurement, the packet injection rate is varied from 0.1 to 0.9 of the network capacity, and packets are injected according to the Bernoulli process based on the given network load. The simulator was warmed up and allowed to run until all the packets reached their destinations. For closed-loop measurement, we collect traces from real applications using the full execution-driven simulator SIMICS from WindRiver, with the memory package GEMS enabled. We evaluate the performance on PARSEC [3] and SPEC CPU2006 [8] workloads. We assume a 2 cycle latency to access the L1 cache, a 4 cycle latency to access the L2 cache, and a 160 cycle latency to access the main memory. In addition, there are 16 memory controllers used to access main memory and each processor can issue two threads.

Different topologies were evaluated on the PARSEC and SPEC CPU2006 benchmarks. Figure 10 shows the speedup of the total number of clock cycles compared to mesh2X. For PARSEC benchmarks, the CTorus improvement over mesh2X ranges from 1.78 for ferret to 2.15 for fluidanimate benchmarks. For SPEC CPU2006, the communication pattern of gcc base gives an improvement of 1.86 whereas the communication in hmmer allows for a speedup of 2.08 over mesh2X. This improvement over mesh2X is due to the 14 hop diameter of mesh2X compared to 4 hops for CMesh2X and CTorus and 2 hops for FBfly2X. CMesh2X and CTorus have the same network diameter, however, the long wrap around links of CTorus allows packets to have a lower average hop count for most traffic traces. Therefore, skipping over more immediate routers in CTorus lowers the average packet latency.

Figure 11 shows the saturation throughput on the synthetic traffic patterns Uniform Random, Non-uniform Random, Bit Reversal, Butterfly, Complement, Matrix Transpose, Perfect Shuffle, Neighbor, and Tornado for the dual topologies. CTorus has a saturation throughput approximately 1.5× higher than CMesh2X and FBfly2X for the

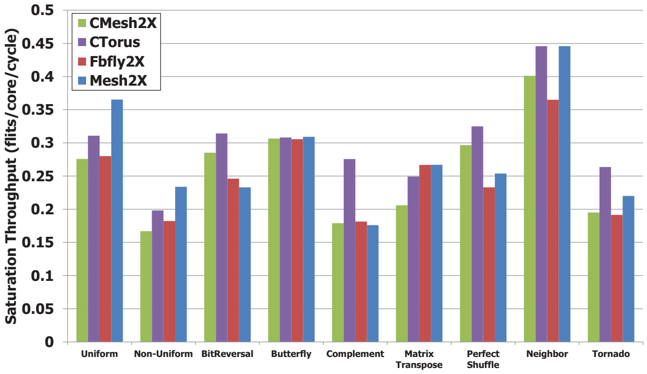

Figure 11: Saturation throughput of 64 cores for synthetic traffic patterns.

Complement traffic pattern. The dual links in the torus and long wrap around links reduce contention for packets traveling all the way across the chip as in Complement traffic. In traffic such as Butterfly, where traffic travels halfway across the chip, the 10 mm links in CMesh2X and FBfly2X allow for a similar saturation throughput compared to CTorus. CTorus improves the saturation throughput by an average of approximately 20% over CMesh2X, 21% over FBfly2X, and 11% over mesh2X.

6 Conclusions

In this paper, we propose a channel buffer and crossbar organization with the objectives of reducing HoL blocking, reducing power dissipation and simultaneously improving performance at the cost of slight area increase. Our results conclude that it is possible to improve performance of channel buffers with some area overhead while saving substantial power when compared to the virtual channel router based NoC architectures. In addition, we compare leading dual topologies such as mesh2X, Concentrated Mesh2X (CMesh2X), and Flattened Butterfly2X (FBfly2X) to a CTorus topology which implements a dual network without the additional area overhead and improves performance by up to 44%.

Acknowledgements

This research was partially supported by NSF awards, ECCS-0725765, CCF-0915537, CCF-0915418, CCF-1054339 (CAREER) and ECCS-1129010

References

[1] J. Balfour and W. J. Dally. Design tradeoffs for tiled cmp on-chip networks. In *Proceedings of the 20th ACM International Conference on Supercomputing (ICS)*, pages 187–198, Cairns, Australia, June 28-30 2006.

[2] L. Benini and G. D. Micheli. Networks on chips: A new soc paradigm. *IEEE Computer*, 35:70–78, 2002.

[3] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The parsec benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, October 2008.

[4] S. Borkar. Will interconnect help or limit the future of computing? Presented at the 19th Annual IEEE Symposium on High-Performance Interconnects (Hot Interconnects), Santa Clara, California, 2011.

[5] W. J. Dally and B. Towles. Route packets, not wires. In *Proceedings of the Design Automation Conference (DAC)*, Las Vegas, NV, USA, June 18-22 2001.

[6] F. Gilabert, M. Gomez, S. Medardoni, and D. Bertozzi. Improved utilization of noc channel bandwidth by switch replication for cost-effective multi-processor systems-on-chip. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 165 –172, May 2010.

[7] M. Hayenga, N. E. Jerger, and M. Lipasti. Scarab: A single cycle adaptive routing and bufferless network. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, December 2009.

[8] J. L. Henning. Spec cpu suite growth: an historical perspective. *SIGARCH Comput. Archit. News*, 35:65–68, March 2007.

[9] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-ghz mesh interconnect for a teraflops processor. *IEEE Micro*, pages 51–61, September/October 2007.

[10] J. Kim. Low-cost router microarchitecture for on-chip networks. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 255–266, 2009.

[11] J. Kim, W. J. Dally, and D. Abts. Flattened butterfly: Cost-efficient topology for high-radix networks. In *Proceedings of 34th Annual International Symposium on Computer Architecture(ISCA)*, pages 126 – 137, June 2007.

[12] J. Kim, C. A. Nicopoulos, D. Park, N. Vijaykrishnan, M. S. Yousif, and C. R. Das. A gracefully degrading and energy-efficient modular router architecture for on-chip networks. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture (ISCA)*, pages 4–15, Boston, MA, USA, June 17-21 2006.

[13] A. K. Kodi, A. Sarathy, and A. Louri. ideal: Inter-router dual-function energy- and area-efficient links for network-on-chip (noc). In *Proceedings of the 35th International Symposium on Computer Architecture (ISCA'08)*, pages 241–250, Beijing, China, June 2008.

[14] G. Michelogiannakis, J. Balfour, and W. J. Dally. Elastic-buffer flow control for on-chip networks. In *Proceedings of the Fifteenth International Symposium on High-Performance Computer Architecture*, pages 151–162, 2009.

[15] T. Moscibroda and O. Mutlu. A case for bufferless routing in on-chip networks. In *Proceedings of the 36th annual International Symposium on Computer Architecture*, June 2007.

[16] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh. Research challenges for on-chip interconnection networks. *IEEE Micro*, 27(5):96–108, September-October 2007.