

Design of Energy-Efficient Channel Buffers with Router Bypassing for Network-on-Chips (NoCs)

Avinash Kodi¹, Ahmed Louri², Janet Wang²

¹School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701

²Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721

¹E-mail: kodi@ohio.com

Abstract

Network-on-chip (NoC) architectures are fast becoming an attractive solution to address the interconnect delay problems in chip multiprocessors (CMPs). However, increased power dissipation and limited performance improvements have hindered the wide-deployment of NoCs. In this paper, we combine two techniques of adaptive channel buffers and router pipeline bypassing to simultaneously reduce power consumption and improve performance. Power consumption can be decreased by reducing the size of the router buffers. However, as reducing router buffers alone will significantly degrade performance, we compensate by utilizing the newly proposed dual-function channel buffers that allow flits to be stored on wires when required. Network bypassing technique, on the other hand, allows flits to bypass the router pipeline and thereby avoid the router buffers altogether. We combine the two techniques and attempt to keep the flits on the wires from source to destination. Our simulation results of the proposed methodology combining the two techniques, yield a overall power reduction of 62% over the baseline and improve performance (throughput and latency) by more than 10%.

Keywords

Network-on-Chips (NoCs), Channel Buffers, Router Bypassing.

1. Introduction

As the industry builds multi-core architecture involving tens and hundreds of cores in the future, on-chip interconnection networks have emerged as a promising candidate for solving the wire-delay problem facing current chip multiprocessors (CMPs) [1], [2]. However, one of the major research challenges currently faced by on-chip interconnection network designers is that of power dissipation [3]. For example, in the Intel TeraFLOPS processor architecture, the interconnect consumes more than 28% of the total power budget, when the expected power budget should be less than 10% [4]. NoC architectures are characterized by the *links* for data transmission and the *routers* for storing, arbitration and switching functions performed by input buffers, arbiters and the crossbar respectively. Power is dissipated both for communicating data across links as well as for switching and storage within the routers [3]. With the increasing need for low power architectures, NoC research has focused on optimizing buffer design [5], [6], [7], minimizing crossbar power [4], [8], and utilizing 3D interconnects [9].

Modular router design ensures that the network bandwidth and storage is shared evenly among all the input channels and packets. This effective sharing of resources (buffer and channel) is achieved by implementing routing, virtual channel (VC) and switch allocation functionalities within the router on a hop-by-hop basis. While the sharing of resources improves the utilization, it also leads to excessive delays seen by every packet/flit traversing from source to destination. Recently, Express Virtual Channel (EVCs) [10] based flow control allowed some network packets to bypass buffering, arbitration and crossbar switching within a single dimension of the on-chip routers, thereby improving latency and reducing power consumption. However, buffer availability through credit based system and VC information has to be explicitly communicated across multiple EVC nodes, which in turn increases the complexity of the design. Recent NOCHI [11] design extended EVCs by allowing buffer/VC information to be broadcast to all nodes using low-swing multi-drop wires which overcomes some of the shortcomings of EVC design. However, NOCHI relies on the use of global wires which requires a separate control plane. This extra control plane adds excessive area. Additionally, broadcasting of communication information across every node adds power (0.6 mW/TX and 0.4 mW/RX). Given the tight power budget, this design maybe suitable where performance is more critical than power consumption such as real-time systems.

Reducing the size of the input router buffers is a natural approach to reduce the power to read/write a flit and area overhead of the router. However, the network performance and flow control is primarily characterized by the input buffers [12]. Recently, iDEAL (inter-router Dual-function Energy and Area-efficient Links) [7], [13] proposed to reduce the size of the buffer and to minimize the performance degradation due to the reduced buffer size, the already existing repeaters along the inter-router channels are doubled as buffers along the channel when required. Research initiatives into optimizing the performance of the repeaters have shown that the repeaters can also be designed to sample and hold data values thereby storing values on the channels [14]. In addition, iDEAL makes use of dynamic buffer allocation to enable a higher buffer occupancy where space is reserved on a per flit basis.

In this paper, we combine iDEAL technique with effective bypassing to reduce both power and improve performance simultaneously. We adopt iDEAL's power and area saving approach by combining circuit and architectural

techniques to reduce power consumption without significant performance degradation. At the links, we deploy circuit level enhancements to the existing repeaters so that they double as buffers when required. At the router buffer, we deploy architectural techniques such as dynamic buffer allocation to prevent performance degradation. To further reduce power and improve performance, we enable bypassing of the router pipeline. When a flit traverses directly to the output of the router via bypassing, it avoids the read/write energy of the input buffers, which saves considerable power. In addition, as it heads directly to the output of the router, it improves latency of the packet as it avoids the router pipeline. As we have additional storage on the wires because of iDEAL design, we can bypass packet on a per-hop basis, unlike EVC which requires explicit control spanning multiple hops or NOCHI which requires broadcasting of control information. This enables packets to be routed wire-to-wire from the source node to destination node at low loads. However, at high loads, blocking probability increases due to wire-to-wire transfers. Therefore, we design a larger crossbar 10×5 to provide bypass path at all loads. Although a larger crossbar occupies more area, recent work on high-radix routers show that these designs are feasible for on-chip networks [15]. Moreover, double-pumped crossbar designed for Intel Teraflops which reduces the size of the crossbar by 57% could be adopted for our design. Synthesized designs using Synposys Power Compiler in the 90nm technology at 500 *Mhz* and 1.0 *V*, show a power reduction of 75% at low loads and 62% at high network loads as compared to the baseline. Cycle accurate network simulation on a 8×8 mesh network topology show a throughput improvement of 10% for all network traffic by combining adaptive channel buffers with bypassing. In what follows, we briefly describe the adaptive channel buffers and effective bypassing used in the proposed design.

2. Adaptive Channel Buffers

In this section, we detail the implementation of the dual-function links and the associated control logic. Figure 1(a) shows a repeater-inserted interconnect, with the conventional repeaters replaced by three-state repeaters [14]. While the tri-state repeater design has been adopted from [14], we significantly differ in the implementation of the control logic as will be explained next. A single stage of the three-state repeaters 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. Once congestion is alleviated, the control logic is disabled and the three-state repeaters return to the conventional mode of operation. The adaptive dual-function links hence enable a decrease in the number of buffers within the router and save appreciable power and area.

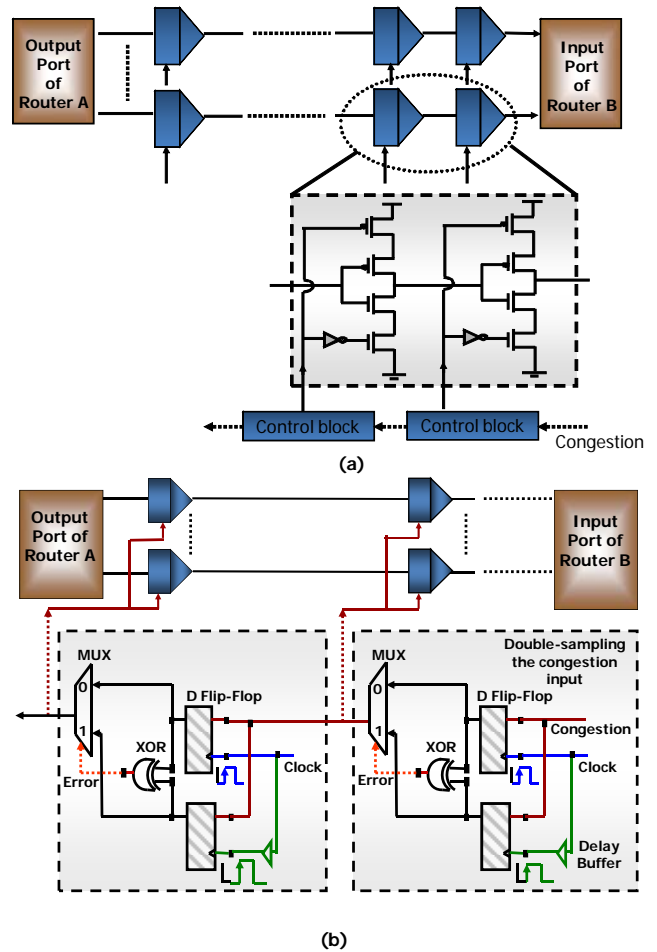


Figure 1: (a) A link using three-state repeaters that function as channel buffers during congestion. (b) Control block using a self-checking double-sample technique.

The control block enables the tri-state repeater inserted link to function as a dual-function link during congestion. A single control block is sufficient to control the functionality of all the repeaters in one stage. Thus the overhead of the control circuitry is negligible compared to the savings in power and area obtained by reducing the router buffer size. In Figure 1(b), the control block is implemented using a self-checking double-sampling technique that enables reliable operation at high frequencies. The incoming congestion signal is sampled by two flip-flops operating at the same clock speed. But the two clocks are slightly offset with respect to each other, such that the data is ensured to be correctly detected at the offset clock edge in spite of any timing errors on the data signal. The multiplexer selects the data sampled at the offset clock edge, in the event of an error. Although this circuit consumes slightly greater area and power, it offers a reliable error-free operation under varying frequencies.

The control block in Figure 1(b) is more efficient than the design using a conventional repeater-inserted control line [14], as the control block provides the following advantages: (1) The control circuit behaves as a delay module as well as a repeater for the congestion signal.

Unlike conventional repeaters, the control circuit shown in Figure 2 operates accurately at variable clock speeds and enables error-recovery in case of timing errors. (2) The control block can be turned OFF by the clocking circuitry when there is no congestion, thus reducing the power consumption along the congestion control line.

3. Router Architecture

In this section, we first describe a generic router architecture. Then we propose the extensions for implementing router bypass and dynamic buffer management.

3.1 Generic NoC Router

Figure 2(a) shows a generic packet-switched NoCs in which every processing element (PE) is connected to a NoC component (router), with most NoCs commonly adopting network topologies such as mesh, or folded torus for regularity and modularity [1], [2], [4], [10]. In wormhole switching, each packet that arrives on the input port progresses through router pipeline stages (routing computation (RC), virtual channel allocation (VA), switch allocation (SA), switch traversal (ST)) before it is delivered to the appropriate output port [12]. At each intermediate router, only the header flit of every packet is responsible for the first two pipeline stages of RC and VA, where as individual flits arbitrate for the SA stage. Each router pipeline stage requires a single clock cycle for every operation. After ST, the flit is transferred on the channel between the routers in the Link Traversal (LT) stage.

For a router architecture with P ports, v VCs/port and r flit buffers/VC the total number of buffers/port is $z = vr$. In Figure 2(a), a generic 5×5 router architecture with 4 VCs and 4 flit buffers ($P = 5$, $v = 4$ and $r = 4$). Each input VC is associated with a VC state table [6], [12]. It maintains the state for each incoming packet and ensures that the body flits are routed to the correct output port. It includes VCID (VC Identifier) of the incoming flit that allows the DEMUX to switch to the correct input VC, RP (read pointer) and WP (write pointer) to read the flit into buffer and write the flit to the crossbar. OP (output port) and OVC (output VC) are provided by the RC and VA stages for the head flit. Each VC is associated with r credits, and for every flit transmitted downstream a credit is consumed.

3.2 Router Bypass Implementation

Figure 2(b) shows the proposed router bypass implementation. To implement bypassing of the router pipeline, every flit of the packet is associated with a lookahead signal. This signal will reserve resources at the next router before the flit arrives at the downstream router. This will allow flits to bypass the router pipeline stages of RC, VA and SA, thereby reducing the latency. Bypassing will also reduce the read/write energy of the buffer as the flit heads directly to the crossbar. This signal will represent the destination router address for the packet. For a $N = 64$ core NoC architecture, we will need 6 bits ($\log_2(N)$) to implement the lookahead. Let us first consider the head flit of the packet. When the lookahead signal for the head flit

arrives one cycle ahead of the packet, it accesses the routing information, VC allocation, switch allocation and credit availability in parallel. Recent designs have shown that VA stage presents the longest critical path and can be accommodated within a single clock cycle [10]. Although credit availability is not critical for the proposed design as we (1) dynamically allocate buffers which ensures higher occupancy and (2) implement congestion signal that will trigger whenever we run out of buffers that will start storage on the wires. This backpressure will be felt by the upstream router that will eventually stop transmitting.

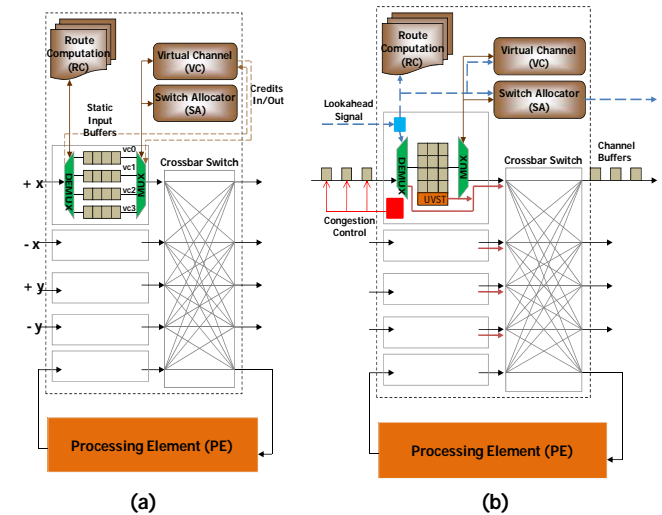


Figure 2: (a) A generic 5×5 NoC router architecture (b) The proposed bypass implementation.

If VC and switch access are both successful, the lookahead signal will (1) set the demux to the bypass path (shown in red) and (2) add the entry into the VC state table to indicate that the flits for the allocated VC should be bypassed. The lookahead signal will then be transmitted to the downstream router in the next cycle. When the head flit arrives at the router, it will be immediately switched to the crossbar and bypass the entire router pipeline and arrive directly at the ST stage. While it is bypassing, the VCID (VC identifier) will be overwritten by the VC state table. This ensures that the packet has the correct VCID when it arrives at the downstream router. During bypassing, the VC state table will also return a credit to the upstream router to indicate a free buffer location and consume a credit for the downstream router. The lookahead signal associated with the body and tail flits will arrive at the router one cycle ahead of the actual flit arrival. Now, the lookahead signal will access the VC state table to determine the output port (switch allocation) and credit availability. If they are both available, once again the body and tail flits will be switched to the crossbar directly. The VC state table will overwrite the VCID and return the credit to the upstream router while consuming the credit for the downstream router. In this way, the flits are routed wire-to-wire thereby bypassing the router pipeline which results in power savings from not having to read/write the flit into the buffers.

In case the bid is successful for bypassing by the head flit, but the switch or the credits are unavailable for the tail or body flits, the lookahead signal will hold the flit on the router buffer by enabling the congestion signal. When the congestion signal is enabled, the flit is stored on the channel buffers and does not enter the router. Once the switch is allocated or the credit is available, the lookahead will release the congestion signal enabling the flit to bypass the router. In case the bid for the VC or the switch is unsuccessful for the head flit due to lack of VC, switch or credit count, the packet is switched based on the VCID into the router buffer. Here, the normal router pipeline accesses of RC, VA, and SA stages occur. Once the output switch is available, the lookahead signal is sent to the next router ahead of the ST stage to bypass the next router. The tail and the body flits will follow the head flit into the router buffer and will enter the SA stage for traversing the router.

When the congestion signal is enabled the bypass path is blocked due to the flit on the wire. However, the MUX of the router can still service the flits within the input buffers. This is possible due to the crossbar design that has two inputs per input port. The proposed design is based on a per router bypass and can bypass to any output port, unlike other bypass designs that can bypass the routers along a single dimension. The added input ports for the crossbar allow flits from both within the router as well as the bypass paths to be switched simultaneously to different output ports. Although this increases the area overhead, we aggressively pursue the reduction of power consumption in NoC architectures.

3.3 Dynamically Allocated Router Buffers

The proposed dynamically allocated router buffer design with bypassing and congestion control is shown in Figure 3. In designing dynamically allocated router buffers, our goal is to maximize the throughput of the network without increasing the router latency. Link list [16] and circular buffers [17] have either the latency penalty or the crossbar scaling issue. As ViChaR's [6] table based approach had solved issues pertaining to latency, we have adopted a similar idea but limited the number of VCs to prevent excess control overhead.

Figure 3 explains the dynamically allocated router buffer. We adopt the unified buffer architecture and augment the architecture with a 'Unified VC State Table' (UVST). In this case, there are v VCs/port, z buffer slots/port and c channel buffers, with r approximately z/v . When a new flit arrives, if the flit is to be bypassed, then the lookahead will set the DEMUX to the appropriate output and access the UVST. In case, the flit is supposed to enter the router buffers, then its VCID cannot be used to switch as all buffer slots are unified. For that purpose, we use the 'Buffer Slot Availability' (BSA) tracking system. BSA allocates/deallocates arriving/departing flits with buffer slots. Therefore, the DEMUX switches to the buffer slot provided by the BSA at the input flit tracking. BSA keeps track of all buffer slots currently available and allocates the first buffer slot found to be free. If the buffer slot number

points to NULL, then such a slot can be selected for the newly arriving flit. After allocating the buffer slot to the incoming flit, BSA then searches for the next free slot to be allocated. Similarly, for a departing flit, BSA will de-allocate the buffer slot using the output flit tracking and add the free slot to the list of free slots maintained in the table (shown in the inset). This allocation and de-allocation occurs only for the flits that enter the router buffer.

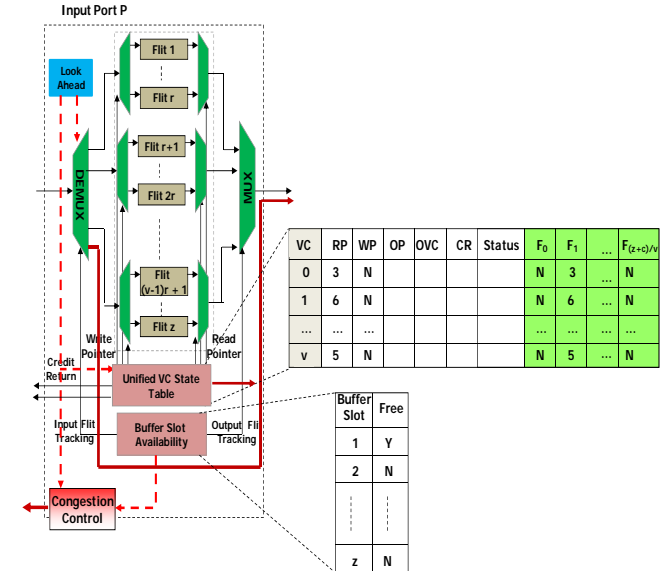


Figure 3: Dynamic buffer allocation with congestion control and the proposed bypass implementation.

Once the flit is associated with the input flit tracking number identifying which flit buffer it is destined to, the flit now arrives at the second DEMUX. Here, the WP logic writes the flit to the buffer slot allocated by the BSA. In the same cycle, UVST identifies the VCID of the newly arriving flit and accordingly updates the UVST. If the newly arriving flit is the header flit, then it will undergo the usual stages of RC, VA, SA, and ST. The UVST table contains buffer slots $F_0, F_1, \dots, F_{(z+c)/v}$ in addition to the regular fields of RP, WP, OP, OVC, CR and Status fields. Here, the status field is used to indicate if the flit will be bypassed or not. The total number of credits is limited to $(z+c)/v$ per VC slot. The buffer slots are used to identify the location of the flit assigned to the particular VC. For fairness purposes, the number of credits is equally divided between all the different VCs. The responsibility for congestion detection rests with the BSA. When BSA finds only a single non-null pointer in its base table, it will trigger the congestion signal. To determine whether the input buffers are full, a small counter that counts the number of free slots is maintained and when this counter reaches one, we trigger the congestion signal. Similarly a departing flit will create a free buffer slot releasing the congestion signal. A single buffer slot combined with a dynamic spare VC for every output port is maintained to ensure deadlock recovery [6]. This congestion signal is implemented as a OR-wired signal as both the lookahead and the BSA can enable this signal.

4. Performance Evaluation

In this section, we evaluate the router bypass and the proposed dual-function links in terms of power dissipation, area overhead and overall network performance. We consider 8×8 mesh with 4-stage pipelined router design. Each router has $P = 5$ input ports (4 for each direction and 1 for the PE). The baseline design considered has 4 VCs per input port, with each VC having 4 flit buffers in the router, for a total of 80 flit buffers ($= 5 \times 4 \times 4$). Each packet consists of 4 flits and each flit is 128 bits long. For the design, we consider 6 different cases where some of the repeaters along the link are replaced by the link buffers and we implement dynamic buffer allocation and bypassing. For a fair comparison with the baseline, the number of flit buffers eliminated from the router is added to the set of link buffers. In each case, the design is implemented in Verilog and synthesized using the Synopsys Design Compiler tool and the TSMC 90 nm technology library at a supply voltage of 1 V and an operating frequency of 500 MHz.

4.1 Power Estimation

For the inter-router channel buffers, we assume the links to be 2 mm long for the mesh network. In the baseline design, there are 8 optimally spaced conventional repeaters along each wire of the 128-bit wide links. The total power consumed by the link per flit traversal is 2.45 mW for the 8×8 mesh [18], [19]. When all the 8 conventional repeaters are replaced by channel buffers, the total power consumed in the link for every flit traversal is found to be 3.55 mW. In the presence of congestion, the power dissipated by the control block with double sampling technique is found to be 6.1 μ W. When the number of VCs or the buffer depth per VC is changed, the size and number of components within the buffer changes, altering the power and area consumption. Considering both the write and read operations in the buffer, the total dynamic power consumed for a 128-bit flit in the buffer is estimated to be 19.28 mW, for the baseline design with 16 buffer slots. The corresponding leakage power is found to be 0.26 mW, giving a total power of 19.54 mW per flit. Decreasing the buffer size by 4 buffer slots (25%) leads to a power savings of 25.72% compared to the baseline. Power reduces by 40.77% when the buffer size is reduced to 50% of the baseline. The switch in the router consumes 0.31 mW per flit traversal, in case of the design with 4 VCs per port and 0.27 mW per flit traversal in the case of 3 VCs per port [20]. However, with additional inputs for the crossbar, the power dissipated doubles per flit traversal. As we utilize a higher radix crossbar to achieve higher throughput, the area of the crossbar increases by 40%. Although this is a substantial increase in area, we believe that double-pumped crossbar designed for Intel Teraflops [4] could be adopted for our proposed work that could decrease the area overhead.

4.2 Throughput, Latency and Power

Figure 4 shows the power, throughput and latency for a 64 node NoC for uniform and non-uniform (bit reversal (BR), butterfly (BU), bit complement (CO), matrix

transpose (MT), perfect shuffle (PS), and tornado (TO)) traffic traces for different designs. Here, the notation followed for the different cases is of $vnv-rnr-cnc$, where nv is the number of VCs per input port, nr is the number of router flit-buffers per VC and nc is the number of link buffers. For example, the baseline is denoted as $v4-r4-c0$, implying 4 VCs per input port, 4 router flit-buffers per VC and 0 link buffers. In addition, we add 'D' to indicate dynamic buffer management and 'B' to indicate bypass, therefore, a 434D-B indicates 4 VCs, 3 flit buffers, 4 channel buffers and both dynamic buffer allocation and bypassing are implemented.

Figure 4(a) shows the power consumed for various configurations at low network load of 0.2. As the figure shows the power consumed for a baseline router is 4.5 W. For 434D and 428D configurations that reduce the size of the buffer and implement only dynamic buffer management, we achieve 22% and 33% reduction in power. For the remaining 3 configurations of 434D-B, 433D-B and 432D-B, we achieve almost 75% power reduction. This is achieved by not only bypassing the router pipeline, but also ensuring that the flits are stored on the wires. We achieve a maximum of wire to wire transfer that translates into substantial power savings for the proposed design. Figure 4(b) shows the power consumed for various configurations at a high network load of 0.5. As the network load increases, the network saturates leading to an increase in flits entering the router buffers. This directly leads to increase in power consumption for the techniques that implement both dynamic buffer management and bypass the flits. For example, in this case, for the 428D, we achieve almost 38% reduction in power consumption as compared to the baseline, where as for the 434D-B, we achieve almost 62% reduction in power consumption as compared to the baseline.

Figure 4(c) and 4(d) show the throughput and latency for various configurations. As seen the performance improves by 10% for 434D-B configuration over baseline and configurations that implement only dynamic buffer allocation. We are able to sustain and even improve performance primarily due to the crossbar design that allows flits to traverse the crossbar when the bypass flit may be stored at the input of the router. This prevents HoL blocking and ensures that the packets flow through the router even at high loads. From Figure 4(d), we can see that the latency at low loads is significantly lower for the configurations that implement both dynamic buffering and bypassing due to the wire-to-wire transfer. In Figures 4(e) and 4(f), we evaluate the throughput and power consumption for different traffic patterns and observe similar trends of reduction in power consumption and improvement in throughput.

5. Conclusion

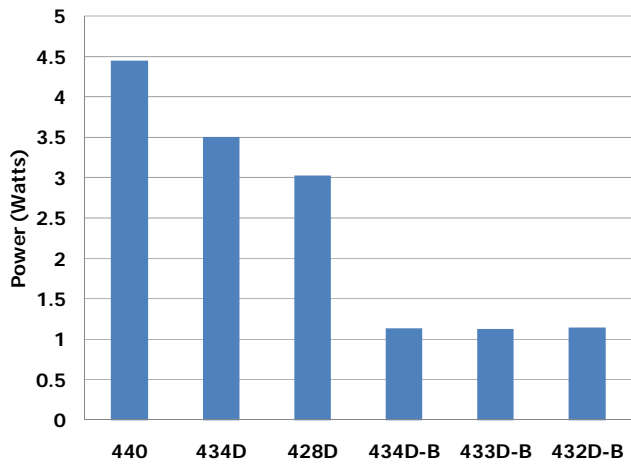
In this paper, we combine two techniques of adaptive channel buffers and router pipeline bypassing to reduce the power consumption and improve performance simultaneously. Power consumption can be decreased by

reducing the size of the router buffers. However, as reducing the number of router buffers alone will degrade performance, we compensate by utilizing the recently proposed dual-function channel buffers that allow flits to be stored on wires when required. Network bypassing technique, on the other hand, allows flits to bypass the router pipeline and thereby avoid the router buffers altogether. We combine the two techniques when appropriate and attempt to keep the flits on the wires from source to destination. We simulated the proposed combined strategy. The results show an overall power reduction of 62% over the baseline at high network loads and improve performance (throughput and latency) by more than 10%.

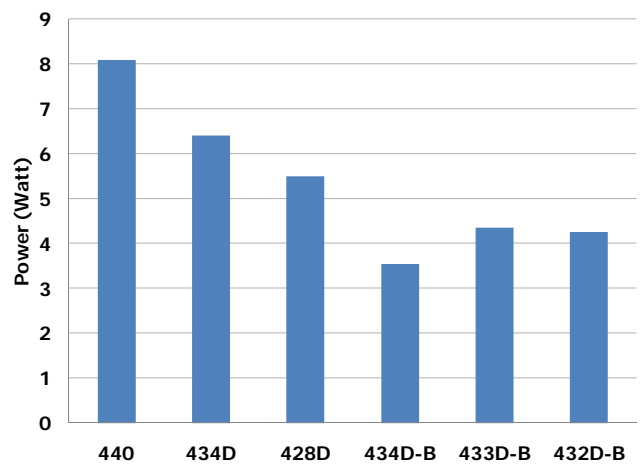
Acknowledgement: This research was partially supported by NSF grants CCF-0538945 and ECCS-0725765.

6. References

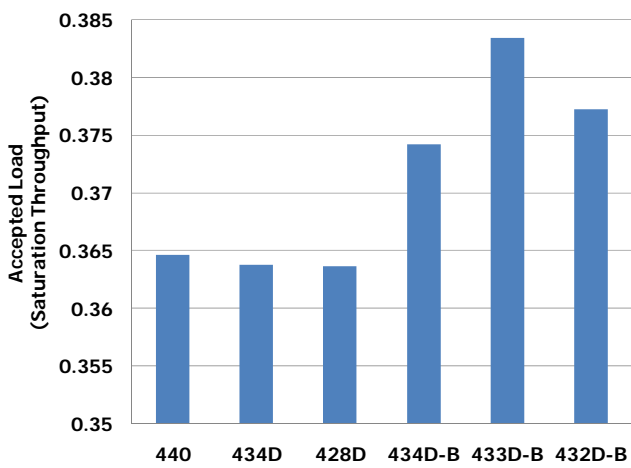
- [1] W. J. Dally and B. Towles, "Route packets, not wires," in *Proceedings of the Design Automation Conference (DAC)*, Las Vegas, NV, USA, June 18-22 2001.
- [2] L. Benini and G. D. Micheli, "Networks on chips: A new soc paradigm," *IEEE Computer*, vol. 35, pp. 70–78, 2002.
- [3] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh, "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, September-October 2007.
- [4] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-ghz mesh interconnect for a teraflops processor," *IEEE Micro*, pp. 51–61, Sept/Oct 2007.
- [5] J. Hu and R. Marculescu, "Application-specific buffer space allocation for network-on-chip router design," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, San Jose, CA, USA, November 7-11 2004, pp. 354–361.
- [6] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "Vichar: A dynamic virtual channel regulator for network-on-chip routers," in *Proceedings of the 39th Annual International Symposium on Microarchitecture (MICRO)*, Orlando, FL, USA, December 9-13 2006, pp. 333–344.
- [7] A. K. Kodi, A. Sarathy, and A. Louri, "ideal: Inter-router dual-function energy and area-efficient links for network-on-chip (noc) architectures," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2008, pp. 241–250.
- [8] H. S. Wang, L. S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in *Proceedings of the 36th Annual ACM/IEEE International Symposium on Microarchitecture*, Washington DC, USA, December 03-05 2003, pp. 105–116.
- [9] S. E. Dongkook Park, R. Das, A. K. Mishra, Y. Xie, N. Vijaykrishnan, and C. R. Das, "Mira: A multi-layered on-chip interconnect for router architecture," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2008, pp. 251–261.
- [10] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 9 - 13 2007.
- [11] T. Krishna, A. Kumar, P. Chiang, M. Erez, and L.-S. Peh, "Noc with near-ideal express virtual channels using global-line communication," in *Proceedings of the Proceedings of Hot Interconnects (HOTI'08)*, Stanford, California, August 2008.
- [12] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, USA: Morgan Kaufmann, 2004.
- [13] A. K. Kodi, A. Sarathy, and A. Louri, "Adaptive channel buffers in on-chip interconnection networks - a power and performance analysis," *IEEE Transactions on Computers*, vol. 57, pp. 1169 – 1181, September 2008.
- [14] M. Mizuno, W. J. Dally, and H. Onishi, "Elastic interconnects: Repeater-inserted long wiring capable of compressing and decompressing data," in *Proceedings of the IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, February 5-7 2001, pp. 346–347.
- [15] J. Kim, W. Dally, B. Towles, and A. Gupta, "Microarchitecture of a high-radix router," in *Proceedings of the 32th Annual International Symposium on Computer Architecture (ISCA'05)*, June 2005, pp. 420–431.
- [16] Y. Tamir and G. L. Frazier, "High-performance multiqueue buffers for vlsi communication switches," in *Proceedings of the 15th Annual International Symposium on Computer Architecture (ISCA)*, Honolulu, Hawaii, USA, May-June 1988, pp. 343–354.
- [17] N. Ni, M. Pirvu, and L. Bhuyan, "Circular buffered switch design with wormhole routing and virtual channels," in *Proceedings of the International Conference on Computer Design (ICCD)*, Austin, TX, USA, October 1998, pp. 466–473.
- [18] K. Banerjee and A. Mehrotra, "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *IEEE Transactions on Electron Devices*, vol. 49, no. 11, pp. 2001–2007, Nov 2002.
- [19] A. K. Kodi, A. Sarathy, and A. Louri, "Design of adaptive communication channel buffers for low-power area-efficient network-on-chip architecture," in *Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, Orlando, Florida, December 3-4 2007.
- [20] H. S. Wang, X. Zhu, L. S. Peh, and S. Malik, "Orion: A power-performance simulator for interconnection networks," in *Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture*, Istanbul, Turkey, November 18-22 2002, pp. 294–305.



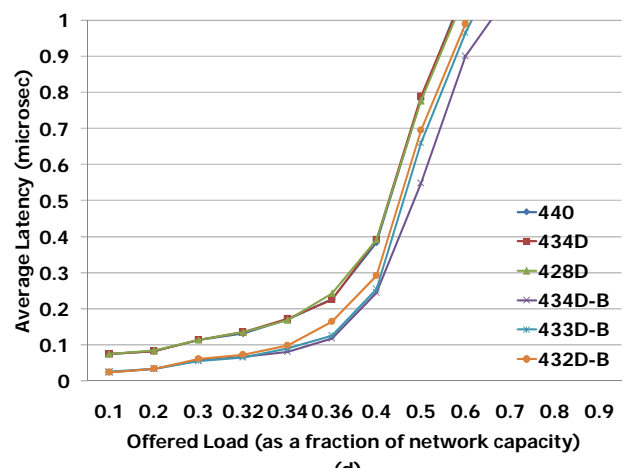
(a)



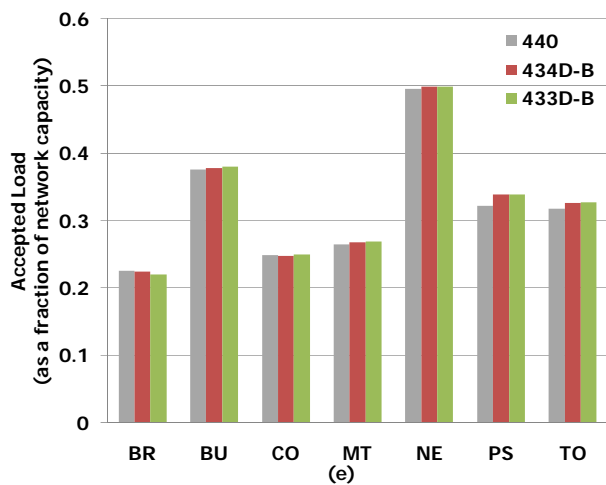
(b)



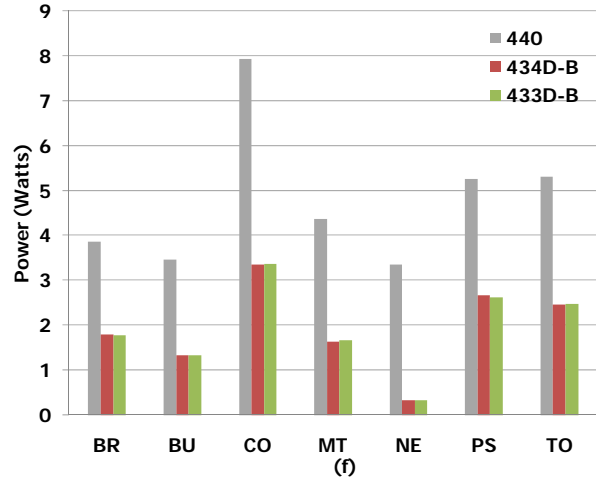
(c)



(d)



(e)



(f)

Figure 4: (a) Power consumption at low network load of 0.2 for Uniform traffic and (b) high network load of 0.5 for Uniform traffic, (c) throughput for Uniform traffic, (d) latency for Uniform traffic, (e) throughput and (f) power for Bit Reversal (BR), Butterfly (BU), Complement (CO), Matrix Transpose (MT), Neighbor (NE), Perfect Shuffle (PS) and Tornado (TO) for 8×8 mesh for various configurations identified as $nV - nrR - ncC$, where nV is the number of VCs per input port, nrR is the number of router flit-buffers per VC and ncC is the number of link buffers with 'D' (dynamic buffer allocation) and 'B' (bypassing).