Co-Design of Channel Buffers and Crossbar Organizations in NoCs Architectures

Avinash Kodi†, Randy Morris†, Dominic DiTomaso†, Ashwini Sarathy‡ and Ahmed Louri‡

†*Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701*

‡*Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721*

kodi@ohio.edu, louri@ece.arizona.edu

Abstract— **Network-on-Chips (NoCs) have emerged as a scalable solution to the wire delay constraints, thereby providing a high-performance communication fabric for future multicores. Research has shown that power, area and performance of Network-on-Chips (NoCs) architecture are tightly integrated with the design and optimization of the link and router (buffer and crossbar). Recent work has shown that adaptive channel buffers (on-link storage) can considerably reduce power consumption and area overhead by reducing or replacing the power hungry router buffers. However, channel buffer design can lead to Head-of-Line (HoL) blocking which eventually reduces the throughput of the network. In this paper, we explore the design space of organizing channel buffers and router crossbars to improve the performance (latency, throughput) while reducing the power consumption. Our proposed designs analyze the power-performance-area trade-off in designing channel buffers for NoC architectures while overcoming HoL blocking through crossbar optimizations. Our simulation and NoC design synthesis shows that for a 8 × 8 mesh architecture, we can reduce the power consumption by 25-40%, improve performance by 10-25% while occupying 4-13% more area when compared to the baseline architecture.**

I. INTRODUCTION

In order to address the growing wire delay problems and improve the performance of future multi-cores, a growing number of designs have adopted the flexible and scalable packet switched architecture, called Network-on-Chips (NoCs) [1], [2]. Power dissipation in NoCs is the most important technology constraint and is rapidly affecting performance (latency and throughput) of multicores [3]. With NoC, researchers have shown that 46% of the router power is consumed by the input buffers, while the crossbar occupies more than 54% of the router area [4]. Therefore, with the need for low-power architectures, researchers have initiated several efforts into optimizing and minimizing the power and area overhead while improving performance of NoC architectures [5], [6], [7], [8], [9], [10].

As router buffers consume substantial power, researchers have analyzed several techniques to optimize and minimize the impact of router buffers. Recently, iDEAL (inter-router Dual-function Energy and Area-efficient Links) [11] proposed to reduce the size of the router 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. While the single channel combined with static virtual channel (VC) allocation created head-of-line (HoL) blocking, iDEAL design resorted to dynamic buffer allocation to sustain performance which in turn increased complexity. Another approach utilizing channel buffering is the Elastic Channel Buffers (ECB), which replaces the repeaters with flip-flops, and eliminates the router buffers altogether [8]. HoL problem was eliminated by creating two separate sub-networks which reduced power consumption and limited area overhead, however the performance (throughput) was significantly affected. Other bufferless networks such as Flit-BLESS [9] and SCARAB [10] adopt either deflecting or dropping conflicting packets, thereby reducing the latency and power, while sustaining throughput at low network loads. However, at high network loads, these networks suffer from excessive deflection/dropping leading to an increase in power consumption. The crossbar within the NoC has also received a lot of attention due to area overhead and power consumption [6], [12], [5]. Researchers have proposed segmented and split crossbars to reduce the power and area overhead. While crossbar optimizations have reduced the power consumption and area overhead, there has been no co-design analysis of channel buffer and crossbar organization.

In this paper, we propose to uniquely co-design channel buffers and router crossbars with the goals of minimizing power consumption, eliminating HoL blocking, and further improving network performance. HoL blocking can be eliminated with dual inputs per router port and to facilitate this design, we analyze three different channel buffer organizations; they include dual channel (dc), dual channel multi-input (dcM), and single channel multi-input (scM) organizations. With dual input ports, we propose multiple ways of organizing the crossbars to take advantage of the speed-up offered with different routing and allocation mechanisms; they include dual input single crossbar (dsx), dual crossbar (dx) and multi-crossbar (mx) organizations. Each of these organizations improve performance from the speed-up and provide varying power savings. 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 while occupying 5% to 13% excess area for different design configurations. Cycle accurate network simulation on a 8 × 8 mesh network topology shows 10-25% improvement

in performance for different synthetic traffic traces when compared to the baseline with identical router buffers. The major contributions of this work are as follows:

- We uniquely identify channel buffer organizations that avoid Head-of-Line (HoL) blocking, thereby prevent performance degradation.
- While various crossbar organizations reduce the power consumption, we further show techniques to improve performance with minimal adaptive routing.
- We evaluate the proposed buffer and crossbar organizations on synthetic (uniform and permutation) and real applications (PARSEC [13] and SPEC2006 benchmarks) showing a performance improvement of 10-25%, power savings of 25-40% with an area overhead of 5-13%.

II. 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 (see inset). 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. The adaptive dual-function links hence enable a decrease in the number of buffers within the router and saves appreciable power and area.

Control Block Implementation: The design shown in Figure 1(a) requires a single control block per inter-router link in order to control all the repeater stages along the link, unlike the design in [8] which uses one control block per stage along the link. Therefore, our proposed control technique is power-efficient and has a lesser area overhead. In addition, the proposed control block outputs one control signal per repeater stage and can thereby tri-state or release each stage independent of the other stages. Figures 1(b) and 1(c) show the control logic and the state diagram for one stage within the control block. The control logic to generate one control signal output, *CTRL* consists of only one flip-flop and three gates. The flip-flop delays the incoming control signal by one clock, while the gates determine the *next state* of the control signal based on the inputs received from the router. The control block operates with two logic states : 'Release' and 'Hold'. In the hold state, the control block delays the incoming congestion signal by one clock cycle before transmitting it to each successive repeater stage. Hence each repeater stage is successively tri-stated to hold the data in position, until the congestion signal is released. During congestion, the router may request the control block to release any given repeater stage, by setting the corresponding bit in the 'release_stage' signal. The control block then moves to the release state and resets the control signal to the particular stage whose release_stage bit has been set by the router. In order to reduce the power consumption due
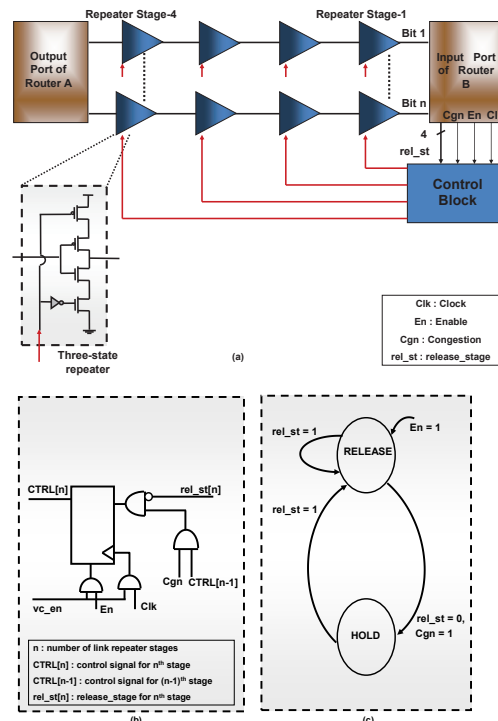

Fig. 1. (a) A link using three-state repeaters that function as channel buffers during congestion, (b) Control block implementation details and (c) State transition diagram.

to the control block, it is enabled by the router only during congestion, using the 'enable' signal. The vc_{en} signal is used in conjunction with the switching control to indicate to the control block the onset of a flit into the repeater stage.

III. CHANNEL BUFFER ORGANIZATIONS

In this section, we propose three channel buffer organizations - dual channel (dc), dual channel multi-input (dcM) and single channel multi-input (scM) organizations. Figure 2 shows the configurations for one link between the upstream and downstream router. Each packet is composed of 4 flits with each flit being 128 bits.

A. Dual Channel Organization

Figure 2(a) 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 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

220

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. 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.

B. Dual Channel Multi-Input Organization

Figure 2(b) shows the dual channel multi-input (dcM) organization. Here, with the goal of increasing the throughput, we organize the channel buffers such that we have 4 VCs but with two channel buffers per VC. This organization will reduce congestion for the same input port. As there are 4 VCs, we have 4 separate control blocks to control the channel buffers. All the congestion signal is fed into the switching control that manages the flow of flits into the different channel buffers. We use two sets of 2-to-1 DEMUXes to reduce the area overhead due to aligning the channel buffers as shown. The VC allocation includes two control bits directed to two sets of DEMUXes to direct the flit to the correct VC. The objective of this organization is to relieve congestion while saving power and minimizing the increase in area overhead. The area overhead of this organization is higher due to the stacking of the channel buffers. We reduce area overhead by increasing the number of repeater stages and resizing the repeaters. This results in increase in the power consumption slightly, as more repeater stages are included, however, the resizing reduces the area overhead.

C. Single Channel Multi-Input Organization

Figure 2(c) shows the single channel multi-input organization. Here, we stack the channel buffers towards the entry point into the downstream router. This organization reduces the congestion only at the entry into the downstream router, however, the HoL blocking is not completely eliminated. The control blocks CB0 to CB3 transmit the congestion signal to the control block CB4. CB4 releases the flit along the single channel buffer only if CB0 to CB3 release the congestion and the head of the line matches the VC identifier. Therefore, this design does not completely eliminate the HoL blocking, however, the design reduces the area overhead as compared to dcM design above. The switching control sends multiple VC allocation information as there can be potentially three in-transit channel buffers. This is needed at the CB4 to determine where each flit held in the channel
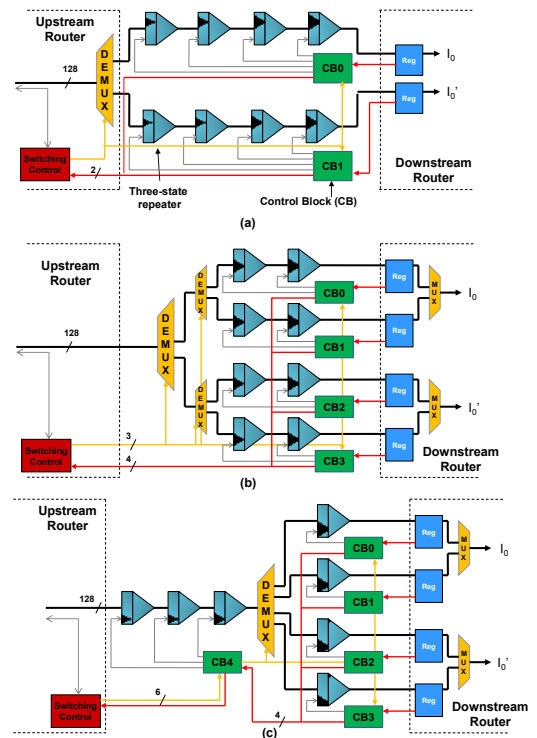

Fig. 2. (a) Dual channel (dc) buffer organization, (b) Dual channel multi-input (dcM) buffer organization and (c) Single channel multi-input (scM) buffer organization.

buffer is directed. This design provides a trade-off between performance and area overhead due to the stacking of the channel buffer at the end of the link.

IV. CROSSBAR ORGANIZATIONS

The dual inputs from the buffer should be utilized to further increase the throughput of the network. To that end, we propose three crossbar organizations with different routing and allocation mechanisms; they include dual input single crossbar (dsx), dual crossbar (dx) and multi-crossbar (mx) organizations.

A. Dual Input Single Crossbar

Figure 3(a) shows the dsx crossbar (1-bit). Dsx is constructed by placing transmission gates between output lines of a matrix crossbar. These transmission gates allow or block an electrical signal from crossing from one side to the other. For example, if a high voltage signal is placed on the transmission gate, there is a conduction path from one side to another. On the other hand, if a low voltage signal is placed on the transmission gate, the electrical current is blocked creating a segmentation of the crossbar input. By correctly controlling these transmission gates, it is possible to segment the matrix crossbar to allow for multiple flits from the same input port to traverse the crossbar. In Figure 3(c), an example of multiple flits traversing the crossbar at the same time is shown. From the figure, I_0 has one flit traversing the crossbar to O_2 and also has another flit traversing the crossbar
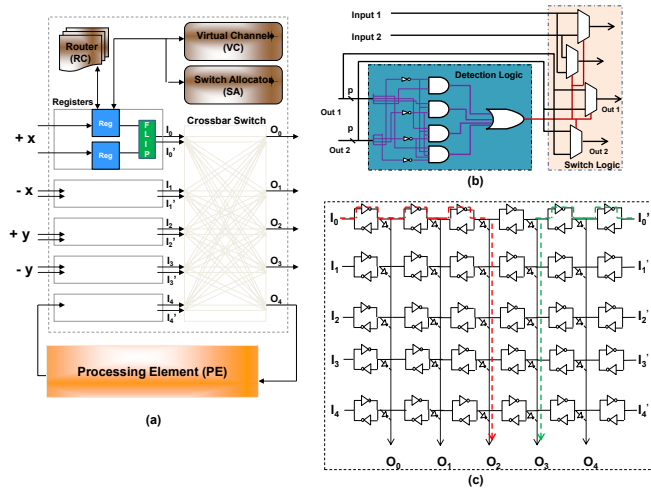
221

Fig. 3. (a) Dual input single matrix crossbar (dsx) organization, (b) Flip logic and (c) Example communication from dual inputs.

to O_3. This is accomplished by having the transmission gate off that is between the two output ports and all the other transmission gates along the input to be on. From the figure, the transmission gate between O_2 and O_3 for input I_0 will be deactivated by placing a value of 0 on the transmission gate. All other transmission gates along I_0 will have a value of 1 as this is require for I_0 to be connected to O_2 and O_3.

Switch Allocator Implementation: As each input port has the potential for two different packets traversing across a crossbar, the standard switch allocation found in most routers needs to be augmented. In a separable output-first switch allocator, flits will proceed through two stages of arbitration [14]. During the first stage, all output ports are combined together (OR logic) into a P bit value, where each bit corresponds to an output port. Then the P bits from each input port are routed to the correct P:1 arbiters. Next, each P:1 arbiter independently selects which input port is granted the right to traverse across the crossbar to the given output port. Afterwards one bit from each of the P:1 arbiters are combined together and progress to the second stage of each input port. In the second stage, the output ports from each input port won compete among the multiple flits inside each input port to see who will traverse the crossbar. To accomplish this, the P bits are logic AND and then OR together with the requesting input port flits to see if there is a match. If there is a match the corresponding bit for the selected requesting flit will be high and will proceed to the V:1 arbiter. Finally, the V:1 arbiter will select a flit to traverse the crossbar. It should be mentioned from the figure, dsx will have a value of 5 for p and a valve of 2 for V as dsx has 5 input/output ports and 2 incoming flits. We add another V:1 arbiter in series with the first arbiter. This second V:1 arbiter is used to select an additional packet for a different output port if the given input port was granted to two or more output ports. The reason the second V:1 arbiter is designed in series and not in parallel with the first V:1 arbiter is we do

not want the second arbiter to select the same input buffer as the first arbiter.

Conflict Free Allocator: Each of the two V:1 arbiters can select a combination of output ports that will cause a conflict. For example (Figure 3(b)), the first V:1 arbiter for input 1 can select output 4 and the second V:1 arbiter can select output 2. As this creates a conflict, only input 1 will have one packet traverse the crossbar. To compensate for these situations, we add addition logic after the switch allocation to detect if a packet conflict arises. Figure 3(b) shows the logic used to evaluate and detect a conflict between the two inputs. From the figure, the conflict detection logic is divided into two different stages. In the first stage, conflict detection takes place by having the two selected output ports from the two inputs enter the detection logic circuitry. After the detection logic, the signal will be an input for four multiplexors which will select the correction conflict free combination. The single crossbar design has more overhead; power as well as latency for additional logic. However, single crossbar design with dual input can provide consistently better performance and different routing algorithms can be easily implemented due to full connectivity.

B. Dual Crossbar Organization

In this organization as shown in Figure 4(a), we split the monolithic crossbar into two, each with smaller number of output ports. This proposed dual crossbar has been well researched in several architectures [6], [15], [16]. The dual 2×2 crossbar used in RoCo is aligned along x and y dimensions, thereby reducing the area and power consumption. Another high-radix router [15] has similar functionality with the dual-input port feeding into two separate crossbars. The dual crossbar organization shown here is slightly different from the previous work as we have a single register connected to the crossbars. This makes the VC allocation more restrictive with the direction in which we expect the packet to turn. For example, with dimension order routing (DOR), the lower VC will be always allocated to x direction until there are more hops in the x direction. At the turn router (from x to y), the higher VC should be allocated. Once the turn is completed, lower VC should be allocated. With 2 VC organization such as dual channel (dc), the number of VCs are limited which cannot support minimal or fully adaptive networks. The other two organizations, dcM and scM will be able to leverage additional VCs to support adaptive routing topologies with some restrictions. The dual crossbar organization reduces the power consumption and area overhead while delivering performance proportional to the dual input crossbar. Due to the single register storage, this design limits the VC allocation during turns.

C. Multi Crossbar Organization

Figure 4(b) shows the multi-crossbar organization which splits the crossbar into 4 smaller crossbars to reduce area and power consumption. The division of the 4 crossbars are along the 4 quadrants: (+x, +y) [North-East], (-x, -y)
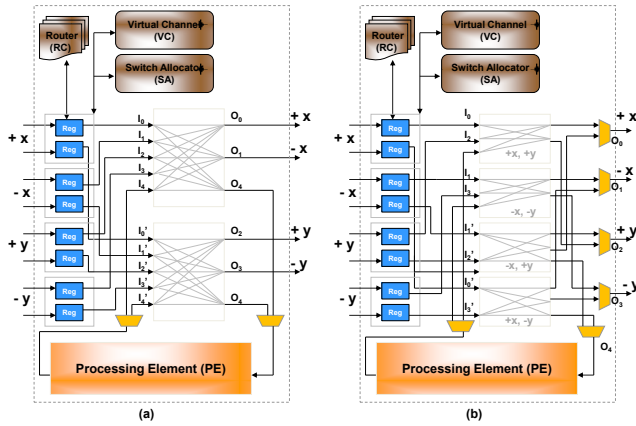
222

Fig. 4. (a) Dual Crossbar (dx) organization and (b) Multi Crossbar (mx) Organization.

[South-West], (-x, +y) [North-West] and (+x, -y) [South-East]. We adaptively route the packet on the quadrant which hosts the destination, assuming the source is located at the origin. Suppose, the packet arrives from +x direction into I_0, indicating that the quadrant is (x+, y+). 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, indicating that the destination quadrant is (x+, y-). Therefore, by limiting the crossbar connections and combining select crossbar outputs, we adaptively provide more opportunities for the output ports to be occupied than a conventional crossbar. The VC allocation is more flexible than the previous approach. 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 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.

V. PERFORMANCE EVALUATION

In this section, we evaluate our proposed channel buffer and router crossbar organizations in terms of power dissipation, area overhead and overall network performance and compare to a baseline VC router. We consider each router with a 4-stage router pipeline (baseline and all proposed approaches) as discussed before. Each router has P = 5 input ports (4 for each cardinal direction; North, South, East and West and 1 for the PE). For a fair comparison, we consider two baseline designs with 2 VCs and 4 VCs per input port with each VC having 4 flit buffers in the router for a total of 40 and 80 flit buffers respectively. Each packet consists of 4 flits where each flit is 128 bits for a total of 512 bits per packet. Every combination of channel buffer and crossbar organization was synthesized and optimized using the Synopsys Design Compiler tool using the TSMC 65 nm technology library. The power dissipation and the area in the links and the routers are obtained for each case at a nominal supply voltage of 1.0 V and an operating frequency of 2 GHz.

A. 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. 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 1 mm long for the mesh network. The buffer organizations considered are (1) dual channel (dc), (2) dual channel multi-input (dcM) and (3) single channel multi-input (scM); and the crossbar organizations considered are (1) dual-input single crossbar (dsx), (2) dual crossbar (dx) and (3) multiple crossbar (mx). Therefore this provides us with 9 different architectures with different naming conventions, (Eg. dc-dx implies dual channel with dual crossbar) and is compared to the baseline which is the 2 VC router. This keeps the number of buffers the same across different designs. Table 1 shows the power and area overhead of each router design in 65 nm technology.

As Figure 5 shows, the majority of the power consumption is in the links. This power is equal in all designs due to the fixed wire length of 1 mm. The baseline input buffers were implemented with 128-bit FIFO registers that were found to have a power of 2.78 mW using Synopsys. Overall, the channel buffers consumed about 40% less power because of the low power three-stage repeaters which were found to have a power of 0.1325 mW each. This difference in power, shown as registers (reg) in Figure 5, is the cause of the large power savings of the channel buffer designs. The dc design with the mx crossbar showed the best reduction at 39.1% compared to the baseline, where as the scM with the dsx crossbar had the least power reduction at 25.1%. The small difference in power between the different channel buffer designs was due to the different number of multiplexers and demultiplexers used. For the crossbars, the power values calculated by Synopsys were lower for the mx crossbar because the total distance for a flit to travel is smaller in the mx compared to the larger dx and dsx crossbars. The large savings in power allowed the channel buffers to have more flexibility with the crossbars while maintaining a significantly lower overall power compared to the baseline.

The latency for the baseline, dc, dcM, and scM designs was found to be 0.47 ns, 0.37 ns, 0.44 ns, and 0.46 ns, respectively. These latencies which were due to the buffering and all were within our specified clock period of 0.50 ns. The small differences in the critical paths of the channel buffer
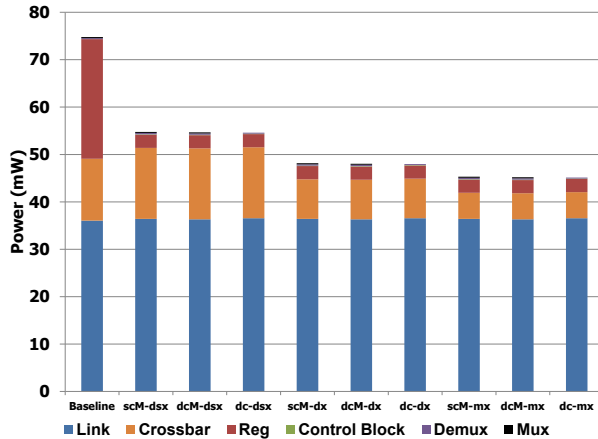
223

Fig. 5. Dynamic power breakdown for different design choices.

Design	Power (mW) Buf + xbar	% Change	Total Area (mm^2) Buf + xbar	% Change
Baseline	61.32 + 13.56	-	0.248 + 0.0356	-
dc-dsx	39.63 + 16.10	-25	0.272 + 0.0471	+12
dc-dx	39.63 + 8.19	-35	0.272 + 0.0246	+4
dc-mx	39.63 + 5.95	-39	0.272 + 0.0237	+4
dcM-dsx	39.71 + 16.10	-25	0.274 + 0.0471	+13
dcM-dx	39.71 + 8.19	-35	0.274 + 0.0246	+5
dcM-mx	39.71 + 5.95	-39	0.274 + 0.0237	+5
scM-dsx	39.81 + 16.10	-25	0.274 + 0.0471	+13
scM-dx	39.81 + 8.19	-34	0.274 + 0.0246	+5
scM-mx	39.81 + 5.95	-38	0.274 + 0.0237	+5

designs was due to the different number of repeaters, demultipexers and multiplexers that a flit had to travel through in each design. The latency of four three-stage repeaters was found to be 0.20 ns and the latency of the demultipexer and multiplexers was found to be 0.08 ns each. Additionally, the latency for the baseline, dx, and mx crossbars alone were 0.35 ns, 0.39 ns, and 0.39 ns respectively. These were due to the critical path of the logic in the VA stage. The latency for the dsx crossbar was larger at 0.47 ns due to the SA stage. The latency for the dsx crossbar was largest at 0.47 ns due to the extra logic needed to switch the vc input flits.

Area overhead of the baseline vc2 router obtained from Synopsys is 0.283 mm^2 which includes the buffer and crossbar. All proposed designs occupy slightly more area compared to the baseline due to the increase in link width. The total area for each channel buffers design is due to the wires, registers, and control blocks because the repeaters and wires use different metal layers [17], [18]. For area optimization of the channel buffers, the link will not be split into separate channels or inputs until the end of the link. This optimization causes the wire to remain a single 128-bit wire for most of the link. However, an increases in the number of repeaters on the link will occur. This will slightly add to the overall power but allows a significant reduction in area. The dc was assumed to be a single 128-bit wire for 0.5 mm then split into two parallel channels for the remaining 0.5 mm causing the total wire length be 1.5 mm. Similarly, the dcM and scM were assumed to be single 128-bit wire for the first 0.875 mm. The lengths were determined in order to offer the best area optimization while also limiting the additional power added by the repeaters. In the dual channel buffer, the two registers and control blocks on the two channels reduced the area overhead. The combination of this channel buffer and the mx crossbar had the least area overhead of only 0.295 mm^2. The smaller 2 × 3 and 3 × 2 crossbars in the mx crossbar results in a lower area for all channel buffer designs. The multiple inputs in the dcM along with the size of the dsx crossbar resulted in an area of 0.322 mm^2, which

was the largest.

B. Simulation Methodology

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. We consider 5 designs out of 9 as they represent the best design choices: dc-dx, dc-mx, dcM-mx, dc-dsx and scM-mx. The proposed designs were compared to a 2 VC and 4 VC router buffer with a standard 5 × 5 crossbar. The network load is varied from 0.1-0.9 of the network capacity. The simulator was warmed up under load without taking measurements until steady state was reached. Then a sample of injected packets were labelled during a measurement interval. The simulation was allowed to run until all the labelled packets reached their destinations. All designs were tested with different synthetic traffic traces such as (1) Uniform Random, where each node randomly selects its destinations with equal probability, (2) Permutation Patterns, where each node selects a fixed destination based on the permutations and (3) PARSEC [13] and SPEC2006 benchmark traces collected using SIMICS simulator with GEMS enabled [19]. For permutation traffic, we evaluated the performance on: Bit-Reversal, Butterfly, Matrix Transpose, Complement and Perfect Shuffle. We consider six PARSEC applications with medium inputs (blackscholes, facesim, fluidanimate, freqmin, streamcluster, ferret and swaptions) and two workloads from SPEC2006 (bzip and hmmer). For collecting the traces, we assumed a 2 cycle latency to access the L1 cache (64KB, 4-way), a 4 cycle latency to access the L2 cache (4MB, 16-way, MOESI cache coherence protocol), and a 160 cycle latency to access the main memory (16 memory controllers, 4 GB main memory).

C. Simulation Results and Discussion

Figure 6(a) shows the throughput plot for UR traffic. From the figure, dcM-mx (dual channel multi-input with multiple crossbars) is the best performing network with a saturation throughput of about 0.37 or a 15% improvement over the baseline VC2. This results from the dual input of dcM where multiple flits from the same input port can traverse to separate output ports. In addition, there are four potential flits

224

that are available to traverse the crossbar instead of only two flits found in the VC2 design. Also dcM slightly outperforms the baseline VC4 design, where VC4 has two times more buffer space. The increase in performance is due to the dual-input nature of dcM as both networks have the same number of flits available (4 flits) to traversal the crossbar. scM, dsx, and dc network designs have a saturation throughput of about 0.35 and have a performance improvement of about 10% over VC2 due to the dual inputs found in each router design. Lastly, dx has the least increase in performance over VC2 with a 6% improvement in performance. This reduction in overall performance over the other designs is due to the restricted dual input crossbar found in dx. In dx, two flits can traverse the crossbars from the same input if the two flits are required to traverse to two different crossbars. Figure 6(b) shows the latency plot for UR traffic. From the figure, dcM has the lowest zero load latency of about 34 clock cycles followed by ScM with a latency of 39 clock cycles.

Figure 6(c) shows the throughput plot for complement (CP) traffic. From the figure, both dcM and dc are able to significantly outperform VC2 with an improvement of about 25% and have similar saturation throughput of VC4. This large increase in performance is mainly due to the restrictive natural of the VC allocation found in the proposed networks. In the baseline case (VC2 and VC4), flits are free to occupy any VC, therefore for complement traffic as more packets travel in the same direction, they see more contention. In the proposed crossbar designs, restrictive VC allocation reduces the contention as they can occupy VCs in both directions, thereby relieving congestion and increasing throughput. dsx and dx have about the same saturation throughput as VC2 because both designs do not use the restricted VC allocation found in dcM and dc designs. Between dsx and dx, dsx is able to slightly outperform dx as dx has no restriction for two flits from the same input port wanting to traverse to two different output port. Figure 6(d) shows the latency plot for CP traffic. From the figure, dcM has the lowest zero load latency of about 35 clock cycles followed by dc with a latency of 36 cycles.

Figure 7 shows the saturation throughput for all traffic traces. From the figure, at least one of our proposed router design is able to out perform both the VC2 and VC4 router design. For Matrix Transpose (MT) traffic, the propose networks dc, scM, and dcM perform the worst. This is due to the restriction in VC allocation causing flits to be stalled in a upstream router which greatly reduces the performance. The best performing networks are dsx and dx. dsx and dx are able to outperform VC2 by about 5% because the dual input crossbars allow for an increase in throughput as more output ports are occupied. For NUR traffic both dcM and scM as the highest throughput and outperforms VC2 by 12%. For BR traffic, dsx is able to outperform V2 by about 5% and dx has the same saturation throughput as VC4. As for perfect-shuffle traffic, both dcM and dc have about a 15% improvement in performance over VC2 and about 10% improvement in performance over VC4.

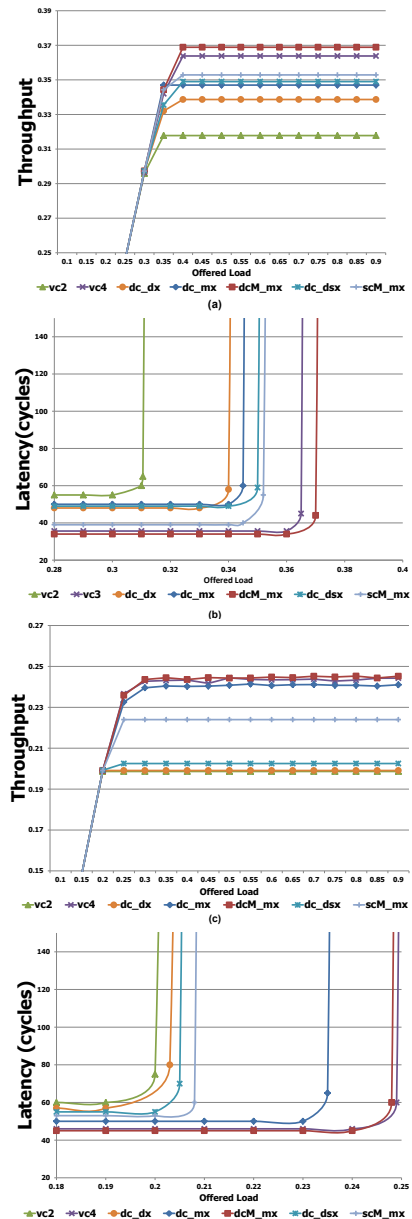PARSEC and SPEC2006 Results: Figure 8 shows the exe-

Fig. 6. Throughput and latency for different design for (a)(b) Uniform Traffic and (c)(d) Complement Traffic.

cution time speed-up when normalized to VC2 configuration. The majority of PARSEC benchmarks (blackscholes, facesim, fluidanimate, ferret and swaptions) show performance improvement of 10-12% speed-up when compared to VC2 baseline. It should be noted that the performance jump obtained from the real benchmarks is equivalent and in some cases even more than a VC4 configuration. This clearly shows that with half the number of buffers (and virtual channels) and smaller crossbars, we can obtain the performance equivalent to what can be obtained with twice the number of buffers. For SPEC2006 benchmarks, the performance jump from most of the combinations is above 10% and outperforms the baseline VC2. Clearly, the combined

Fig. 7.   Throughput at an offered load = 0.5 for all synthetic traces
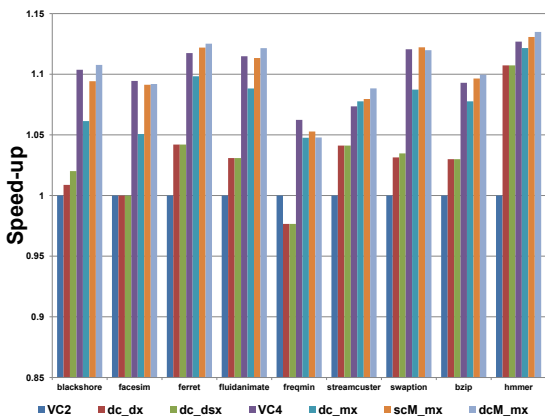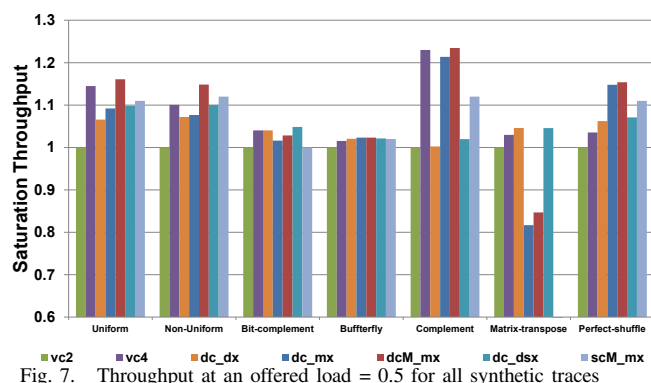

Fig. 8.   PARSEC and SPEC2006 speed-up when normalized to the execution of vc2 router design.

effects of channel buffer organizations and crossbar designs improve the performance for both synthetic as well as real applications.

VI. CONCLUSION

In this paper, we evaluated different organizations of channel buffers and crossbars with the twin objectives of reducing power dissipation while improving performance at the cost of slight area increase. Our best designs show power savings of 39% while improving performance from 10-20% at the cost of 4-13% area overhead. The dual channel design combined with multiple crossbar organization showed that we can achieve high throughput and minimize power while expending some area. The single crossbar design consumes more area and power while yielding better performance across all traffic patterns. Our dual link designs reduce the HoL blocking of traditional channel buffers and increase throughput with restrictive VC allocation with multiple cross-bars. 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 same number of VC router buffer based NoC architecture.

REFERENCES

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

[2] 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.

[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] P. Kundu, "On-die interconnects for next generation cmps," in *2006 Workshop on On- and Off-Chip Interconnection Networks for Multi-core Systems*, Stanford, CA, USA, December 6-7 2006.

[5] 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)*, Cairns, Australia, June 28-30 2006, pp. 187–198.

[6] 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)*, Boston, MA, USA, June 17-21 2006, pp. 4–15.

[7] 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.

[8] 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*, 2009, pp. 151–162.

[9] 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.

[10] 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.

[11] 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)*, Beijing, China, June 2008, pp. 241–250.

[12] 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.

[13] 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.

[14] D. U. Becker and W. J. Dally, "Allocator implementations for network-on-chip routers," in *SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 2009, pp. 1–12.

[15] G. Mora, J. Flich, J. Duato, P. Lopez, E. Baydal, and O. Lysne, "Towards an efficient switch architecture for high-radix switches," in *ACM/IEEE Symposium on Architecture for Networking and Communications systems*, December 2006, pp. 11 –20.

[16] J. Kim, D. Park, C. Nicopoulos, N. Vijaykrishnan, and C. Das, "Design and analysis of an noc architecture from performance, reliability and energy perspective," in *ACM/IEEE Symposium on Architecture for Networking and Communications systems*, 26-28 Oct 2005, pp. 173–182.

[17] 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.

[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, November 2002.

[19] M. Martin, D. Sorin, B. Beckmann, M. Marty, M. Xu, A. Alameldeen, K. Moore, M. Hill, and D. Wood, "Multifacet's genreal execution-driven multiprocessor simulator (gems) toolset," *ACM SIGARCH Computer Architecture News*, no. 4, pp. 92–99, November 2005.