Elastic Buffer Flow Control for On-Chip Networks

George Michelogiannakis, *Student Member, IEEE,* and William J. Dally, *Fellow, IEEE*

Abstract—Networks-on-chip (NoCs) were developed to meet the communication requirements of large-scale systems. The majority of current NoCs spend considerable area and power for router buffers. In our past work, we have developed elastic buffer (EB) flow control which adds simple control logic in the channels to use pipeline flip-flops (FFs) as EBs with two storage locations. This way, channels act as distributed FIFOs and input buffers are no longer required. Removing buffers and virtual channels (VCs) significantly simplifies router design. Compared to VC networks with highly-efficient custom SRAM buffers, EB networks provide an up to 45% shorter cycle time, 16% more throughput per unit power or 22% more throughput per unit area. EB networks provide traffic classes using duplicate physical subnetworks. However, this approach negates the cost gains or becomes infeasible for a large number of traffic classes. Therefore, in this paper we propose a hybrid EB-VC router which provides an arbitrary number of traffic classes by using an input buffer to drain flits facing severe contention or deadlock. Thus, hybrid routers operate as EB routers in the common case, and as VC routers when necessary. For this reason, the hybrid EB-VC scheme offers 21% more throughput per unit power than VC networks and 12% than EB networks.

Index Terms—On-chip interconnection networks, Interconnection architectures

1 Introduction

Networks-on-chip (NoCs) have been developed to address the communication requirements of large-scale systems enabled by semiconductor technology scaling [1]. Past work has attributed approximately up to 40% [2] of the power and 11% [3] of the area of the overall chip to the NoC. A significant part of the above costs is in the router buffers, quoting numbers as high as 75% [3] of the area and 22% [4] of the power. At the same time, channels come at little cost in the on-chip environment because wires can usually be routed above other logic [1], [5]. Thus, the area they occupy is only that of repeaters and retiming elements. This has motivated research to reduce or eliminate buffer cost.

To eliminate buffer cost, we have proposed elastic buffer (EB) flow control in [6]. EB flow control adds a simple control logic block to drive the enable inputs of the master and slave latches of master-slave pipeline flip-flops (FFs) separately, as illustrated in Fig. 1. This allows FFs to use each latch independently for buffering because each latch can simultaneously contain different data. Therefore, each FF becomes an EB and behaves as a two-slot FIFO. Consecutive EBs make network channels act as distributed FIFOs. Flits advance to the next EB using a ready-valid handshake. An EB asserts its *ready* signal routed upstream to indicate that it has at least one free storage slot. Furthermore, an EB asserts its *valid* signal routed downstream to indicate that it is driving a valid flit. When ready and valid are asserted between two EBs at a rising clock edge, a flit has advanced. This timing requires at least two storage slots per EB to avoid unnecessary pipeline bubbles. With EB flow control, channels are used for buffering in lieu of input buffers.

Removing input buffers removes virtual channels (VCs). This increases head-of-line blocking and therefore reduces performance. However, area and power are also reduced from removing the buffers. Therefore, the datapath can be made wider such that performance or cost is equalized compared to a NoC with input buffers. This produces a fair comparison because it clearly shows the performance per unit cost efficiency of the compared networks. Removing VCs also removes the VC allocator. Consequently, because an input may only request a single output, the switch allocator is replaced by output arbiters. Furthermore, credits are not used. Therefore, routers are simplified compared to VC routers [7].

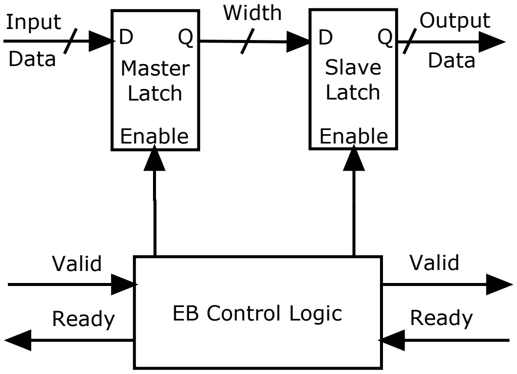


Fig. 1. An elastic buffer.

Two router designs for EB networks were presented in [8]. They improve on the baseline design of [6]. The two-stage EB router emphasizes on throughput by reducing cycle time. Compared to a VC router with speculative switch allocation [7], the two-stage EB router offers 8% more throughput per unit power in a 2D mesh with dimension-order routing (DOR), assuming equal clock frequencies. It also reduces cycle time by 45%. The second EB router design merges the two-stages of the two-stage router to avoid pipelining overhead and prioritize latency. This single-stage router requires 29% less energy per transferred bit, but also has a 33% longer cycle time compared to the two-stage router. By using the optimal EB router and shortest cycle time for each comparison, a 2D mesh EB network provides 16% more throughput per unit power 22% more throughput per unit area or has an up to 45% shorter cycle time compared to a similar VC network with highly-optimized custom SRAM buffers. Furthermore, because EB networks have a wider datapath when equalizing performance or cost, they reduce zero-load latency.

• *George Michelogiannakis and William J. Dally are with the Electrical Engineering Department, Stanford University, CA 94305. E-mails: {mihelog,dally}@cs.stanford.edu*

A comparison of wormhole (without VCs) and EB routers has shown that EB routers remain more area and power efficient [9]. Because the complexity of wormhole routers is directly comparable to EB routers since wormhole routers also do not have VCs and use output arbiters instead of allocators, this shows that the dominant factor dictating area and power savings for the EB network is the ratio of the overall network area and power cost made up by the buffers. That ratio determines the area and power that can be traded for a wider datapath in the EB network to equalize performance or cost. On the other hand, simplifying router logic is the primary contributor for reducing cycle time.

Credit count, which is typically used for sensing congestion in VC networks, is not available in EB networks. Thus, EB networks use a congestion sensing mechanism which counts flits bidding for and traversing output channels.

To provide complete traffic separation, multiple *physical channels are used in the same way as multiple* virtual channels (VCs). An efficient way to provide multiple physical channels is by using duplicate physical networks, all of which connect to the network endpoints (sources and destinations). This provides separation between traffic classes and increases throughput per unit power [6]. However, this is only true up to a number of traffic classes. That number depends on implementation technology and details. Above that number, the cost at the network endpoints outweighs the benefits, because endpoints have to connect to each network in the general case. Also, the significance of the control overhead and serialization latency both increase as the datapath width of each network decreases to equalize the overall cost with the single-network design. Providing separation for an arbitrary number of traffic classes has been identified as the primary weakness of EB flow control.

To provide a large number of traffic classes, in this paper we propose a hybrid EB-VC scheme which reinstates input buffers with VCs but only to drain flits at router inputs that are blocked for a predefined number of cycles, as well as other flits belonging to the same class. Therefore, head-of-line blocking is alleviated by removing blocked flits from EB channels. If buffer occupancy for the congested class exceeds a predefined threshold, a control signal is transmitted upstream to halt further transmission of flits of that class. While there is still interaction between flits of different traffic classes in the EB channels, deadlocks are prevented and performance increases.

A properly-chosen predefined threshold makes hybrid EB-VC routers more energy efficient than VC routers because buffers are not used in the common case, as well as more energy efficient than EB routers because hybrid routers reduce head-of-line blocking. Specifically, in a 2D mesh, networks with EB-VC routers offer 21% more throughput per unit power than VC routers, and 12% than EB routers. However, buffers still occupy area and lengthen the timing path of the first pipeline stage. Moreover, each input in a hybrid EB-VC router may request multiple outputs, which necessitates a switch allocator. This makes hybrid routers comparable to VC routers in area and cycle time, but they still allow limited interaction between flits in different VCs. Thus, VC networks offer 41% more throughput per unit area compared to hybrid EB-VC networks, while EB networks offer 49%.

The rest of this paper is organized as follows: Section 2 outlines the two EB router designs. Section 3 presents deadlock avoidance schemes. Section 4 discusses congestion sensing and adaptive routing in EB networks. Section 5 presents our results. Section 6 provides an overview of related work. Finally, Section 7 concludes this paper.

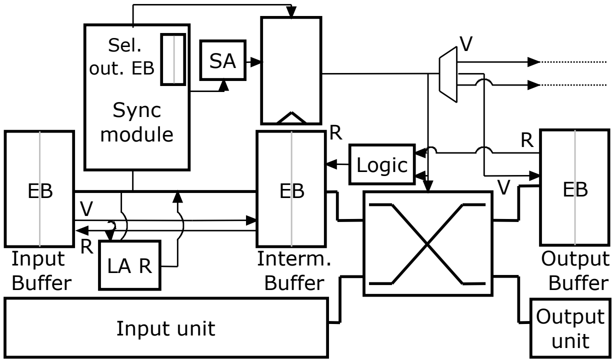


Fig. 2. The two-stage router.

2 Elastic Buffer Routers

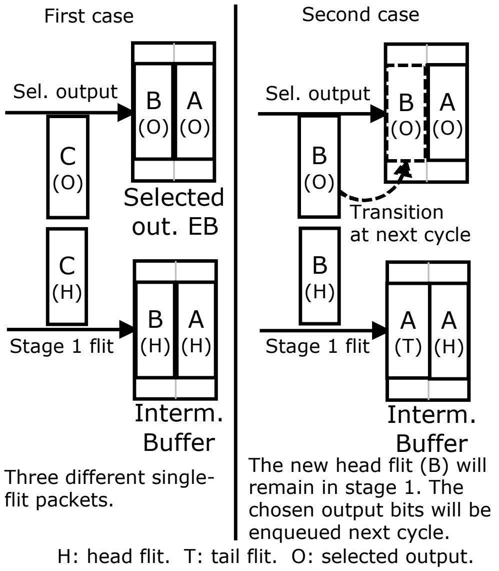
This Section presents the two EB router designs.

2.1 Two-Stage Router

The *two-stage* router is illustrated in Fig. 2. Only one input and one output are shown in detail. EBs are used between pipeline stages for pipelining and storage. Therefore, *valid* (V) and *ready* (R) signals are used to facilitate the transfer of flits from one EB to the next. The two-stage router emphasizes on throughput by reducing cycle time. This router was initially proposed in [8] as the *enhanced two-stage router*, improving on the *baseline two-stage router* of [6].

The input EB receives flits from the channel and drives them to the first pipeline stage. Look-ahead routing (LA R) [10] is used at routers to calculate the output at the next hop for all incoming packets. Therefore, head flits already contain their desired output when entering a router. In parallel with routing, flits send a request to their desired output's arbiter. Therefore, switch arbitration (SA) and routing occur in the first pipeline stage. The output arbiters deassert their grants as long as their output EB is non-ready (full). However, flits advance to the intermediate EB using the ready-valid handshake as soon as it has a free storage slot, without waiting for a switch grant. Therefore, each input port can buffer up to four flits in the input and intermediate EBs.

Because flits advance to the intermediate EB as soon as it is ready, extra care must be taken to maintain alignment between flits arriving at the switch traversal stage and their grants. That is because flits may get granted before or after they are stored into the intermediate EB, depending on contention, and may have flits of other packets ahead of them in the intermediate EB traversing the crossbar. Alignment is maintained by the synchronization module (sync module). It maintains the selected output ports for the flits stored in the intermediate EB in a separate *selected output EB*. The selected output ports are used to route non-head flits. The output port contained in the head of the selected output EB is always that of the current packet (oldest to have flits remaining in the first stage or the intermediate EB). The selected output EB may also contain in its tail (master latch) the selected output port of the next packet as shown in Fig. 3 (left). In the worst case, only the tail of the current packet remains and there are two more single-flit packets. The most recent to arrive is contained in the intermediate EB, and the other is driven in the router's first stage. In that case, the selected output EB stores



Three different single-flit packets.

The new head flit (B) will remain in stage 1. The chosen output bits will be enqueued next cycle.

H: head flit. T: tail flit. O: selected output.

Fig. 3. Synchronization module operation.

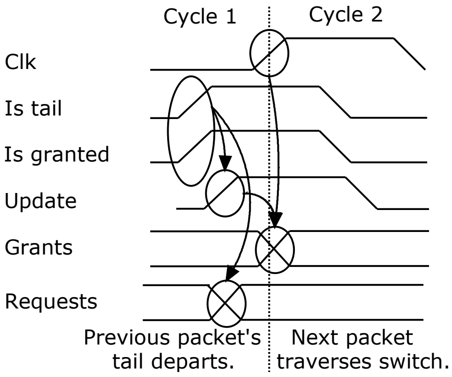

Cycle 1
Cycle 2
Clk
Is tail
Is granted
Update
Grants
Requests
Previous packet's tail departs.
Next packet traverses switch.

Fig. 4. Updating output grants.

the current packet's selected output port at its head and the next packet's selected output port at its tail. The third packet's selected output port is driven as an input. It will be enqueued when the third packet is enqueued into the intermediate EB, a cycle after the tail flit of the current packet departs.

The synchronization module detects when the tail of the current packet is about to depart from the intermediate EB and propagates the selected output port of the next packet to the switch arbiters. Arbiter outputs are registered to shorten the critical path such that it does not extend past the first pipeline stage. Thus, propagating the selected output port is done one cycle in advance to avoid inserting bubbles, as shown in Fig. 4.

Depending on the next packet's time of arrival, it may either have its head flit stored in the intermediate EB, or driven in the first router stage. In the former case, the next packet's selected output port will be stored in the selected output EB's master latch. In the latter case, the selected output port will be driven as an input to the selected output EB. However, if the next packet's head flit has remained in the first stage for more than one cycle because the intermediate EB is full, the selected output EB is non-empty because there is no intervening packet, and the next packet's selected output port will be stored in the selected output EB. This is illustrated in Fig. 3 (right). The synchronization module propagates the selected output port of the next packet from the selected output EB's master latch, slave latch, or the input to the master latch, knowing if the intermediate and selected output EBs are non-ready (full) or non-valid (empty).

Flits at the head of each input's intermediate EB traverse the switch if they have a grant from their selected output and that output's EB is ready (non-full). In that case, a *ready* signal is asserted to the intermediate EB to release the flit. In addition, the *valid* output of the intermediate EB is demultiplexed to the proper output EB. Grants are made on packet boundaries and then gated by the selected output EB's *ready* output. When a tail flit is traversing the switch, that input's synchronization logic asserts an *update* signal to all outputs. An output which receives an update signal from the input it is granting has its grant registers clocked at the next clock edge, thus updating the grants driven to the other router components. This is illustrated in Fig. 4. Arbiters also have their grant register clocking enabled if they are currently granting no input, to assure that grants for newly-arrived packets will be propagated. An extra storage slot in the output EBs is not required because the decision to have the flit traverse the switch is made in the same cycle as the flit would arrive at the output EB.

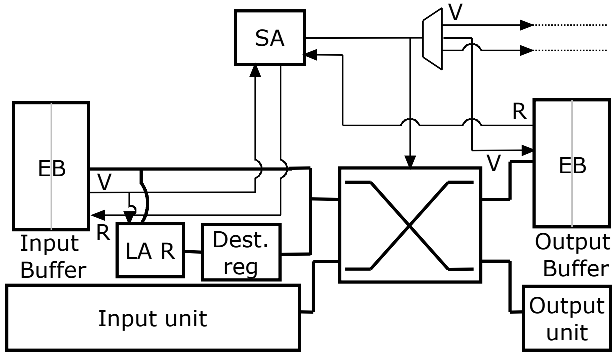


Fig. 5. The single-stage router.

2.2 Single-Cycle Router

The *single-stage* router is shown in Fig. 5 [8]. It prioritizes latency instead of throughput and avoids pipelining overhead by merging the two stages of the two-stage router. Incoming flits request their outputs from the arbiters, calculated in their previous hop by the look-ahead routing logic and stored in the destination register, and in parallel compute the output at their next hop. Only output arbiters with ready (non-full) output EBs can assert grants. Flits traverse the switch and are stored in the output EB in the same cycle that they are granted.

3 Deadlock Avoidance and Traffic Classes

This Section discusses traffic separation in EB networks and motivates the hybrid EB-VC design proposed in this paper and described in Section 3.3.

3.1 The Interleaving Deadlock

With the removal of VCs, tail flits may get blocked behind head flits of other packets due to the FIFO nature of EB channels. The same is true for wormhole networks because their input buffers are single-lane FIFOs. Therefore, packet interleaving is infeasible in both networks [11]. This is illustrated in the example of Fig. 6. A head flit of a new packet requires a free register to store its desired output (dependency *A*). However, a register cannot be released until a tail flit arrives (dependency C). On the other hand, no tail flit may bypass the blocked head

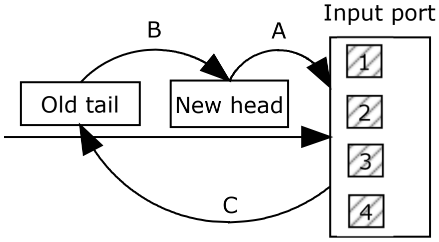


Fig. 6. The interleaving deadlock cyclic dependency.

flit to arrive at the input (dependency *B*). Disabling packet interleaving does not degrade network performance assuming that sources transmit flits of the same packet contiguously. To the contrary, since packets are considered delivered once their tails arrive, this may decrease average packet latency. However, disabling packet interleaving may raise fairness issues in the presence of long packets. Interleaving of packets in different VCs is allowed for the hybrid EB-VC router described in Section 3.3, similarly to VC routers.

3.2 Duplicating Physical Channels

Duplicate physical channels define traffic classes in the same way as duplicate virtual channels (VCs). All relevant literature on VCs is applicable to prevent cycling dependencies within or across traffic classes. Network destinations have to be able to eject traffic from all classes required to prevent protocol deadlocks [11], [12].

Duplicate channels can be efficiently provided by instantiating separate physical subnetworks. Each subnetwork carries traffic from a single traffic class. In the general case, the subnetworks are independent; network endpoints connect to all of them. Hierarchical approaches are possible to reduce the radix of the endpoints. To facilitate packets switching traffic classes, channels can connect parts of some subnetworks to other subnetworks. Such a design should be fully customized for a specific topology and routing algorithm to allow only the required transitions between traffic classes. Such an example is described in Section 4.2.

With duplicate subnetworks, each subnetwork should have a reduced datapath width such that the design with the duplicate subnetworks has the same energy or area cost as a single network. This ensures a fair comparison. Reducing the datapath width results in a more cost efficient network because of the crossbar's quadratic cost relationship with the number of incoming and outgoing wires. For instance, doubling the number of subnetworks and halfing the datapath width will result in a net reduction in area because both the input side and the output side of the crossbar now have half the length, therefore the crossbar occupies a quarter of the area. This assumes that crossbar area is dictated by wire pitch, which is most often the case for not overly narrow datapaths. Therefore, using multiple subnetworks both provides traffic classes and increases cost efficiency.

Cost efficiency is increased by duplicating subnetworks up to a certain number of subnetworks. Above that number, the radix of the network endpoints becomes significant, or the datapath becomes narrow enough such that control overhead makes the design inefficient. That number of subnetworks depends heavily on implementation technology, endpoint design and network constraints and therefore needs to evaluated in specific chip designs and implementation technologies. Furthermore, narrowing the datapath increases serialization latency which can be an important consideration. However, narrowing the datapath may reduce router cycle time if the switch is in the critical path. Finally, care must be taken to load the duplicate subnetworks equally to avoid idling network resources while others are oversubscribed. Load balancing requires properly assigning traffic classes to subnetworks, or adjusting the datapath width of each subnetwork individually. However, balancing load becomes a harder problem as the number of traffic classes increases.

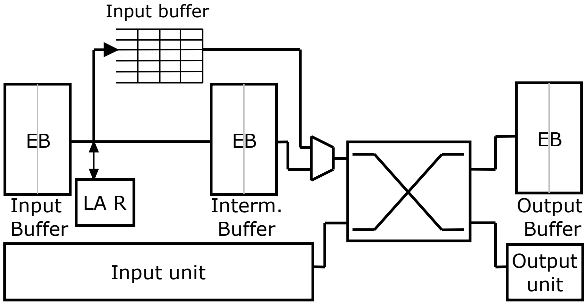


Fig. 7. Hybrid EB-VC router.

Other approaches are possible but are less cost efficient [6]. Duplicating channels but multiplexing and demultiplexing them into single switch ports provides a minimal benefit for a disproportional channel cost increase. Also, duplicating channels and switch ports increases the crossbar cost quadratically, outweighing the performance benefits. Recent work has also proposed duplicating just the switches in the routers, but not channels [13].

3.3 Hybrid EB-VC Networks

To provide an arbitrary number of traffic classes, in this paper we propose adding input buffers slotted per traffic class to EB networks. We base this hybrid EB-VC design, shown in Fig. 7, on the two-stage EB router, because the buffer introduces logic complexity which necessitates pipelining.

The buffer drains flits that are blocked in the input EB for a predefined number of cycles, *BL CYCL*. *BL CYCL* is a design-time parameter that affects how often the buffer is used. Each input has a counter per traffic class. A flit blocked in the input EB will cause the counter of that traffic class to be incremented by one. A flit leaving the input EB to be stored into the intermediate EB causes the same counter to be decremented by one. Counters are also decremented if the flit at the head (slave latch) of the input EB is from a different class. If a counter is equal or greater than *BL CYCL*, flits of that class leave the input EB and are stored into the buffer. When the counter is incremented and becomes equal to *BL CYCL*, it is further incremented by a predefined value. This creates a bias towards draining. Otherwise, interleaved flits from other classes will cause the counter of the class under contention to be decremented. Therefore, when the next flit of the class under contention arrives, it will be blocked for a few more cycles. In our implementation, counters are incremented by the number of traffic classes as soon as they are incremented and become equal to *BL CYCL*. This efficiently handles the case of having a flit from every other class interleaved before the next flit from the class under contention.

To prevent buffer overflow, a backpressure signal for each channel is routed upstream via dedicated wires. The backpressure signal instructs the traffic source upstream to pause sending flits of the class under congestion. That occurs as soon as the free buffer slots barely suffice to drain the channel from flits of the class under congestion in the worst case. The worst case is all flits in the upstream channel belonging to that class, with more to be transmitted until the congestion signal arrives. Therefore, the buffer always has enough free slots to drain incoming flits of the class under congestion, similar to a skid buffer. Once the number of free slots for a class increases above the threshold, a signal is sent upstream to resume transmission of that class.

Flits bid for the switch from the intermediate EB and the buffer. Therefore, a switch allocator is required because each input may request multiple outputs. The allocator ignores requests from classes to outputs that have received a backpressure signal for those classes. A VC allocator may be required if flits are permitted to transition to different classes, or if multiple VCs are provided per class.

Routers cannot predict at transmission time if flits will be stored into the buffer of the downstream router. Thus, using credit flow control and consuming a credit when transmitting a flit would be pessimistic because it would assume that all flits would use the buffer. Therefore, hybrid EB-VC routers do not use credits but only use the ready-valid handshake, which is required for EBs.

The presence of buffers makes the occupied area almost identical to VC routers for equally-sized buffers. Moreover, the hybrid routers need an allocator and extra control channels for the backpressure signal. Furthermore, flits may momentarily block flits of other traffic classes in the EB channels. Therefore, while packet priorities can be provided by the hybrid EB-VC router, in the worst case there can be considerable interaction with lower-priority packets.

The worst case blocking latency in the hybrid EB-VC network for a flit which just traversed the switch occurs if the flit at the head of the router's output EB (slave latch) and every flit ahead of it blocks for *BL CYCL* cycles. For a single hop, the worst-case blocking latency solely because of flits of other classes is $((EBs + 2) \times 2 - 1) \times (BL\ CYCL)$, where "EBs" is the number of EBs in the channel. This takes into account the downstream router's input EB and the output EB the flit is currently in. This theoretical worst case is extremely rare, thus not affecting performance in practice.

The hybrid EB-VC router consumes almost the same dynamic energy as the two-stage EB router in the common case. EBs remain the primary means of storing flits. Buffers are only used to to alleviate head-of-line blocking and resolve situations which would otherwise result in a deadlock. These situations form when packets of different traffic classes contend in the EB channels and form either cyclic dependencies in the network or at the endpoints due to the communication protocol [12], [14]. The buffer is used to resolve those dependencies by draining flits. The choice of *BL CYCL* represents a tradeoff. With a low value the buffer is used similarly to VC networks. With a high *BL CYCL* value the buffer is used rarely, resembling EB networks.

A proper choice of *BL CYCL* makes the hybrid EB-VC router more energy efficient than VC routers because buffers are not used in the common case, and more energy efficient that EB routers because it reduces head-of-line blocking. However, the buffers still extend the timing path of the first pipeline stage and impose an area overhead compared to EB routers. Also, in certain buffer implementations or high-performance technology libraries, leakage power in the buffers may become a concern.

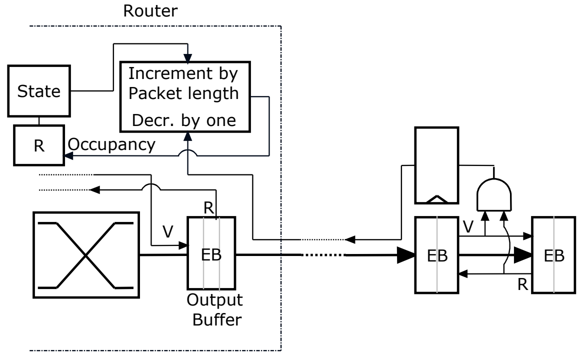


Fig. 8. Measuring output occupancy.

The hybrid EB-VC router provides an arbitrary number of traffic classes. If only a few traffic classes are needed, duplicating subnetworks is more efficient as explained in Section 3.2; duplicating subnetworks also avoids buffer area and timing overhead as well as does not allow interaction between flits from different classes in EB channels. However, the hybrid EB-VC design can be combined with duplicate subnetworks. Choosing among the hybrid design, duplicate subnetworks or a combination thereof requires an evaluation of the different performance and cost metrics specific to each network design and implementation technology.

4 Congestion Sensing

This Section describes congestion sensing mechanisms for EB networks. It then outlines an application of adaptive routing.

4.1 Congestion Sensing Mechanism

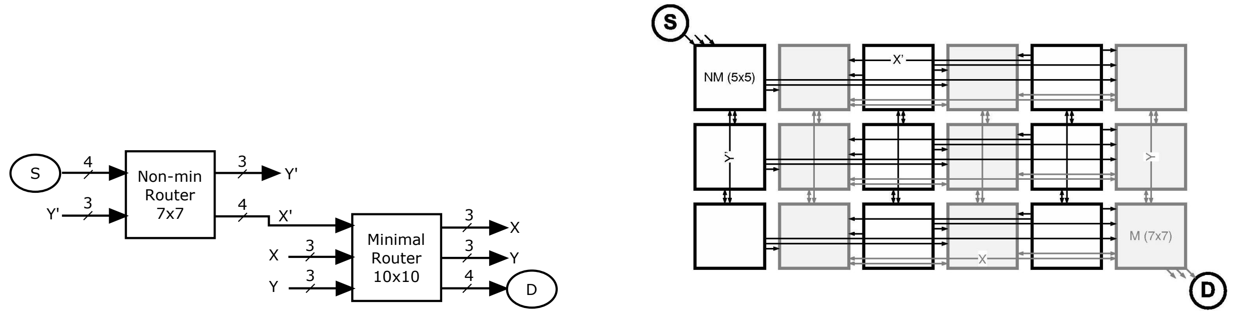
Congestion sensing in EB networks must measure channel occupancy to estimate congestion because credits are not available. To further alleviate contention, the mechanism must take into account packets that have been routed to an output but are still bidding for the switch. Otherwise, all inputs wanting to send to an unblocked output will be considered a low congestion scenario. Moreover, the mechanism must adapt quickly to current network status.

We find the *output occupancy* metric to be optimal because it satisfies the above properties [6]. This metric counts the flits currently in a segment of each output channel, called the observation region. When the head flit of a packet is routed, the occupancy counter for the chosen output is incremented by the packet length in flits. Output counters are decremented by one for each flit leaving the observation region.

The logic for tracking the number of flits in the observation region is shown in Fig. 8. Flits leaving the observation region are detected using an AND gate whose inputs are the *ready* and *valid* signals between the last EB of the region and the next EB. The AND gate's output propagation delay affects the reaction time to congestion. Like the *ready* and *valid* wires, that wire can also be engineered more aggressively for delay. For this study, the observation region is the same length as the shortest network channel. The AND gate's output wire propagation delay is half of that channel's delay in cycles, rounded up.

4.2 Adaptive Routing

Any adaptive routing algorithm is applicable to EB networks by using the congestion sensing mechanism described in Section 4.1. Depending on the number of the required traffic



(a) The two routers, one for each subnetwork. Illustrated for a 4×4 UGAL FBFly.

(b) A 3×3 UGAL FBFly. Non-minimal router (NM) connect to minimal routers (M – shaded) via X' channels.

Fig. 9. The UGAL FBFly. Two traffic classes are defined by two subnetworks.

classes, network designs should use duplicate subnetworks, the hybrid EB-VC router, or a combination. As an example of customizing to the routing algorithm in EB networks, we apply universal globally adaptive load-balancing (UGAL) [15] to a flattened butterfly (FBFly) [16] topology. To define the *minimal* and *nonminimal* traffic classes required for UGAL, we use two subnetworks interconnected as shown in Fig. 9.

Packets are injected into the *nonminimal* subnetwork. Then, a random intermediate destination *I* is chosen. A packet is routed to *I* (nonminimally to the final destination *D*) if the load of the output port on the route to *I*, multiplied by the nonminimal hop count to *D*, is larger than the load of the output port on the route to *D* multiplied by the hop count of the minimal route to *D*. That decision is never revisited. When routing to *D* or *I*, packets take up to one hop in each dimension and follow dimension order. In our implementation, output load is measured by the *output occupancy* metric described in Section 4.1.

Packets in the *nonminimal* network are routed in Y′X′ dimension order to *I*, and then in YX dimension order to *D* in the *minimal* subnetwork. This prevents cyclic dependencies. Packets that do not choose to route to an intermediate destination proceed directly to *D* in X′Y dimension order. The X′ hop places them in the *minimal* subnetwork. Because flits traverse from the nonminimal class to the minimal but not vice-versa, X′ channels are not bidirectional. If a nonminimal route has been chosen but there is no X′ hop, the first hop after reaching *I* is X′ instead of X since flits would have reached *I* in the *nonminimal* subnetwork. Therefore, routing becomes Y′X′Y. The channel connecting a nonminimal router to the adjacent minimal router is used when a traversal to the *minimal* subnetwork is required but there is no X′ hop. Thus, an extra hop is created. Routing Y′X′YX results in better column load balancing than routing Y′X′XY, since flits in the *minimal* subnetwork traverse the column of the randomly-chosen *I*. Routing Y′X′XY makes flits use *D*'s column.

Because routers are unaware of distant congestion, we extend UGAL by applying progressive adaptive routing (PAR) [17]. For every hop towards *I*, another intermediate destination is randomly chosen that would result in an output in the same axis as the output towards *I*, to preserve dimension order. UGAL is applied to choose the best option between *I* and the new intermediate destination; that option then becomes *I*. Packets take up to one hop in each dimension to ensure forward progress. Also, when taking an X′ channel to traverse to the minimal network, another X′ channel is randomly chosen and UGAL is applied to choose among the two. PAR is not applied in the *minimal* subnetwork because the final destination is constant. Disabling PAR reduces maximum throughput by 14% with our chosen congestion metric [6].

Routers in the *nonminimal* subnetwork have a smaller radix. Thus, they have shorter critical paths and more cycle time to make the complex adaptive routing decisions. On the other hand, adaptive routing decisions are made in the *nonminimal* subnetwork and cannot consider the load of the *minimal* subnetwork because it is distant. Adaptive routing in the minimal subnetwork would violate dimension order.

VC networks share channels between the different traffic classes, and thus average out channel utilization. For the EB network, PAR is a cause of load imbalance in the *minimal* subnetwork, because it makes the intermediate destination choice not truly random. Finally, traffic that needs to be ejected from the same router it was injected to needs to traverse to the *minimal* subnetwork, using a X′ link. This makes that traffic contend with other traffic. This issue can be alleviated by increasing network cost to add more channels, such as ejection ports in the *nonminimal* subnetwork.

5 Evaluation

This Section presents evaluation results.

5.1 Methodology

We use a modified version of Booksim [12] for EB networks. We use two topologies: a 2D mesh with DOR and a 2D FBFly [16] with UGAL routing [15], as described in Section 4.2 for the EB network. For fairness, all comparisons assume the same number of subnetworks, each carrying a single traffic class. We assume the same amount of load for each traffic class.

The mesh is configured as a 4×4 or an 8×8 grid. Each router has a single node attached to it. The FBFly has four nodes attached to each router (concentration factor of four), and routers arranged in a 4×4 grid. Injection and ejection channels have a single cycle of latency. The 4×4 mesh channels have two cycles of latency, while the 8×8 mesh channels have one cycle of latency. For the FBFly, short, medium and long channels have 2, 4 and 6 cycles of latency, respectively. The channel connecting two adjacent routers of different subnetworks for the EB UGAL FBFly has one cycle of latency. Also, *nonminimal* routers are 7×7 while *minimal* ones are 10×10. PAR is applied to both the EB and VC FBFly. In all networks, each cycle of latency represents 2mm of physical distance.

Sources generate fixed-size 512-bit packets and enqueue them into the injection buffer of the proper subnetwork. Each buffer can inject and eject up to a single flit per cycle. The set of traffic patterns [12] used for the evaluation is uniform random, random permutations, shuffle, bit complement, tornado and neighbor traffic for the mesh. For the FBFly we also include transpose and an adversarial traffic pattern. The adversarial traffic pattern aims to load a small subset of network channels by making all sources connected to a router send to destinations connected to a single other router.

For the Pareto-optimal curves shown, the datapath width is swept from 29 to 171 bits such that packets consist of 3-18 flits. Flits consist of 1 phit. Also for the Pareto-optimal curves, the maximum throughput is the average of the maximum throughput of each traffic pattern; the consumed power is the average of the power consumptions at the maximum throughput of each traffic pattern. Percentage summaries are calculated by calculating the average distance between sampling points of different networks, dividing by the normalized aspect, and averaging among all sampling points.

VC networks use a two-stage router design [7]. The first stage consists of input buffering, look-ahead routing [10], VC and speculative switch allocation [7]. The second stage is switch traversal. VC and switch allocators are separable input-first with round-robin arbiters, executing a single iteration per cycle. EB networks also use round-robin arbiters. We do not assume input buffer bypassing [18]. Our energy evaluation focuses at the saturation points at which this has minimal effect. When comparing the VC and the hybrid EB-VC networks, all data points assume one VC per traffic class and an equal number of VCs between the two networks. Eight buffer slots are statically assigned to each VC for both networks. For each datapath width used when comparing EB to VC or hybrid EB-VC networks, we sweep the number of VCs and buffer slots statically assigned to each to maximize throughput per unit power. We only consider buffer depths that cover the credit round trip latency to avoid penalizing latency. For a 64-bit datapath, 4 VCs is the optimal choice for the UGAL FBFly, of 10 slots each for full-swing and 8 slots for low-swing channels. For the mesh, the optimal choice is 4 VCs of 9 slots each for full-swing and 8 for low-swing channels.

Area and power results are based on cost models or placement and routing. Comparisons among EB routers but not with the VC and hybrid EB-VC routers use placement and routing. Comparisons with the VC or hybrid EB-VC networks use the cost models described in [5]. Comparisons between the VC and the two-stage EB routers use device and interconnect parameters from a 65nm general-purpose CMOS technology in the typical case. However, comparisons among the EB routers as well as the hybrid EB-VC router use device and interconnect parameters from a commercial 45nm low-power technology library, under worst case conditions for both energy and timing. However, to clearly illustrate the reduced complexity of EB routers compared to VC routers, we also include placement and routing results. All placement and routing comparisons are performed using the same 45nm library.

The area and power models route wires above other logic and report only the area of the repeaters and FFs. Therefore EB channels have the same area and power as pipelined channels with FFs because they have the same number of latches in the datapath. Router area is estimated using detailed floorplans, and input buffers are implemented as custom SRAMs. Critical devices in the channels and router datapaths, such as repeaters used to drive large wire capacitances, are sized to ensure circuits will operate at the clock frequency. The power model includes the major devices in the channels and routers, and includes leakage currents. The FFs in the channels are clock-gated locally. We also present results for low-swing channels. Aggressive low-swing channel designs can achieve up to a 10× traversal power per bit reductions compared to full-swing [19]. As a conservative estimate, our low-swing channel model has 30% of the full-swing repeated wire traversal power, and double the channel area.

TABLE 1
Two-stage EB network percentage gains compared to VC.

Norm	DOR Mesh			UGAL FBFly		
Comp:	Area	Thr.	Power	Area	Thr.	Power
Full-swing						
Area	-	1%	7%	-	-10%	10%
Throughput	2%	-	8%	-20%	-	-3%
Power	-11%	8%	-	-16%	-2%	-
Low-swing						
Area	-	2%	10%	-	-11%	15%
Throughput	2%	-	12%	-23%	-	0%
Power	-15%	10%	-	-24%	0%	-

To perform the placement and routing comparisons, we synthesize a single router instance using Synopsys Design Compiler and place and route the synthesized netlist using Cadence Silicon Encounter. Placement and routing captures cost not regarded by the cost models, such as the cost for arbitration or allocation, credits, the synchronization logic presented in Section 2.1 as well as the ready-valid handshake logic. Clock frequencies are determined by static timing analysis using post place and route parasitics. Routers are optimized for minimum cycle time in the place and route flow. Energy per transferred bit results are calculated by simulating the placed and routed netlists under an equal cycle time and flit injection rates. The initial floorplan utilization is set to 70%. Primary input and output driving strengths, loads and timing constrains are specified to realistically assume network channels at router ports. Router ports are placed in the floorplan according to the inter-router connections of the assumed network. The switch is implemented using multiplexers.

To illustrate the effects of removing the buffers, simulations comparing VC routers against EB or hybrid EB-VC routers assume a clock frequency of 2GHz. Comparisons among EB routers use each router's minimum cycle time. However, to illustrate the influence of cycle time, we also present throughput curves assuming an equal clock frequency for the EB routers. These curves use a cycle time of 4.45ns. However, the routers are still optimized for minimum cycle time in the place and route flow. Throughput and latency are measured in absolute time for curves using different cycle times.

5.2 EB and VC Network Comparison

First, we compare the VC router with the two-stage EB router. Fig. 10 shows Pareto-optimal curves for the 4×4 2D mesh and the UGAL FBFly, with routers operating at an equal clock frequency. Therefore, these curves ignore benefits from the reduced cycle time of EB routers, discussed later. These curves illustrate Pareto-optimal design points which show the maximum throughput achieved by the two networks given a certain area or power budget, as well as the area or power required to achieve a certain maximum throughput. Table 1 summarizes the percentage gains. Rows indicate which aspect was equalized between the EB and VC networks. Columns

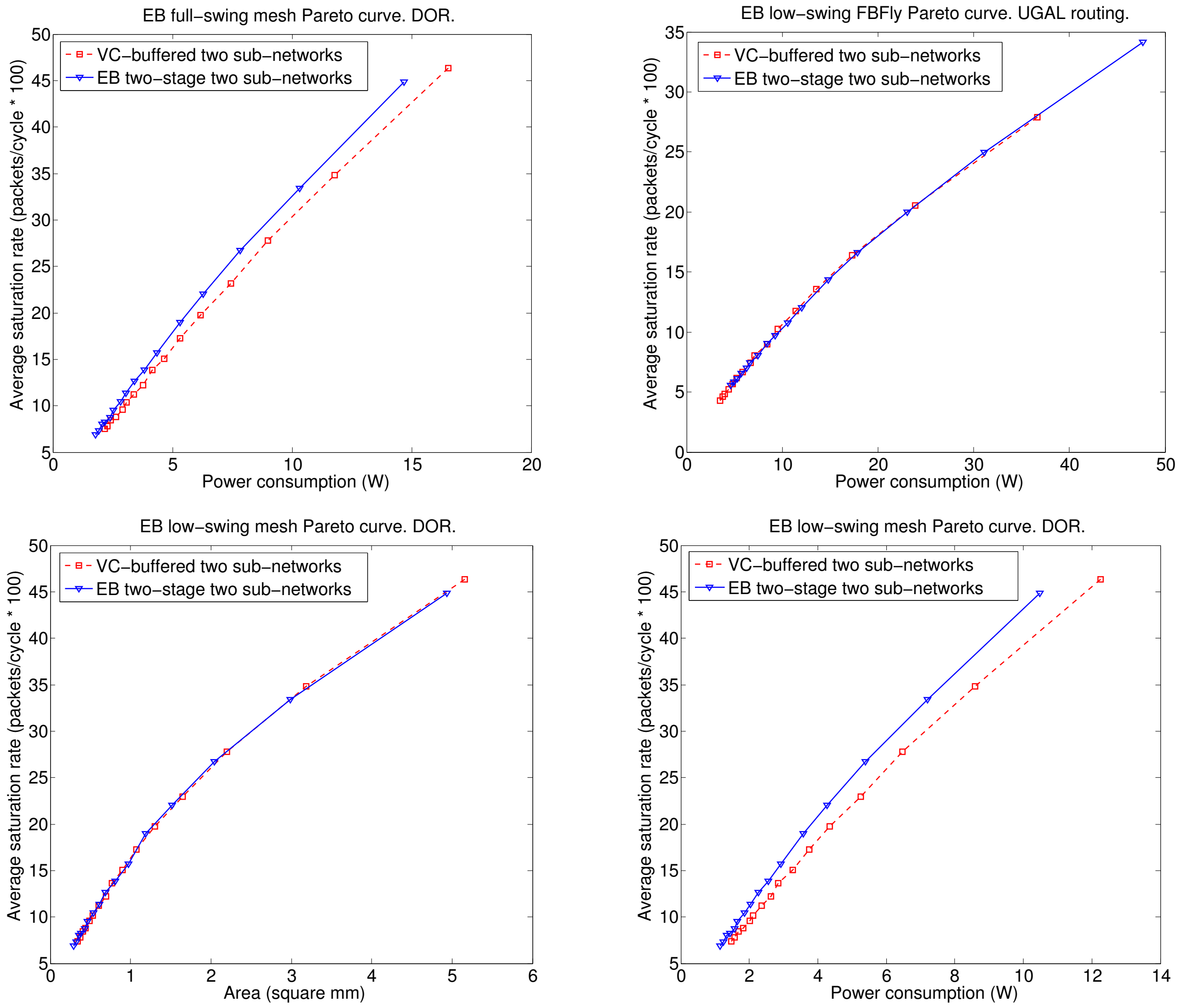


Fig. 10. Pareto-optimal curves for full and low-swing channel networks.

show the percentage gains for the mesh and the FBFly. Positive percentages indicate gains for the EB network. For fairness, both networks have one subnetwork for requests and one for replies. This increases the efficiency of both networks as explained in Section 3.2. Section 5.3 offers more insight for buffer costs which result in the performance per unit cost increase of EB networks.

Fig. 11(a) shows latency as injection rate increases for a 64-bit datapath. Zero-load latency is equal because the two routers have the same number of pipeline stages and EB channels have the same latency as channels with an equal number of FFs. In the UGAL FBFly, zero-load latency is 3% higher for the EB network due to using the channels to transition from the *nonminimal* to the *minimal* network. Due to the lack of input buffers and VCs which results in increased head-of-line blocking, the EB network is not able to reach the VC network's maximum channel utilization rate. Therefore, the EB network saturates at a 34% lower injection rate than the VC network.

EB networks consume less power than VC networks for equal injection rates, as shown by Fig. 11(b). At the EB network's saturation rate, the VC network consumes 14% more power. VC networks which bypass input buffers when they are empty and under no contention [18] would still not consume less power than EB networks, because flits would traverse as many pipeline FFs as EBs in the two-stage EB router. The single-stage EB router would consume less power because it lacks pipelining overhead. Furthermore, buffer bypassing would add logic and cost overhead to the VC router under high load, where bypassing buffers is rare.

TABLE 2
Two-stage EB and VC router implementation comparison.

Aspect	VC router	EB router (2-stage)
Number of ports	703	683
Number of nets	16202	6117
Number of gates	60010	12269
Number of cells	15943	5691
Area (μm^2)	63515	15080
Cycle time	3.3ns (41FO4)	1.8ns (22FO4)

As explained in Section 5.3, EB networks trade cost savings for wider datapaths to increase throughput. Therefore, EB networks that have equal throughput or cost with a VC network will have a lower serialization latency. EB networks using the single-stage design would further reduce zero-load latency.

Table 2 presents place and route results for 5×5 mesh routers using DOR. The VC router has 2 VCs of 8 buffer slots each. Due to technology constraints, buffers are implemented as FF arrays. The VC router uses look-ahead routing [10]. Results show a 76% decrease in occupied area and an 45% decrease

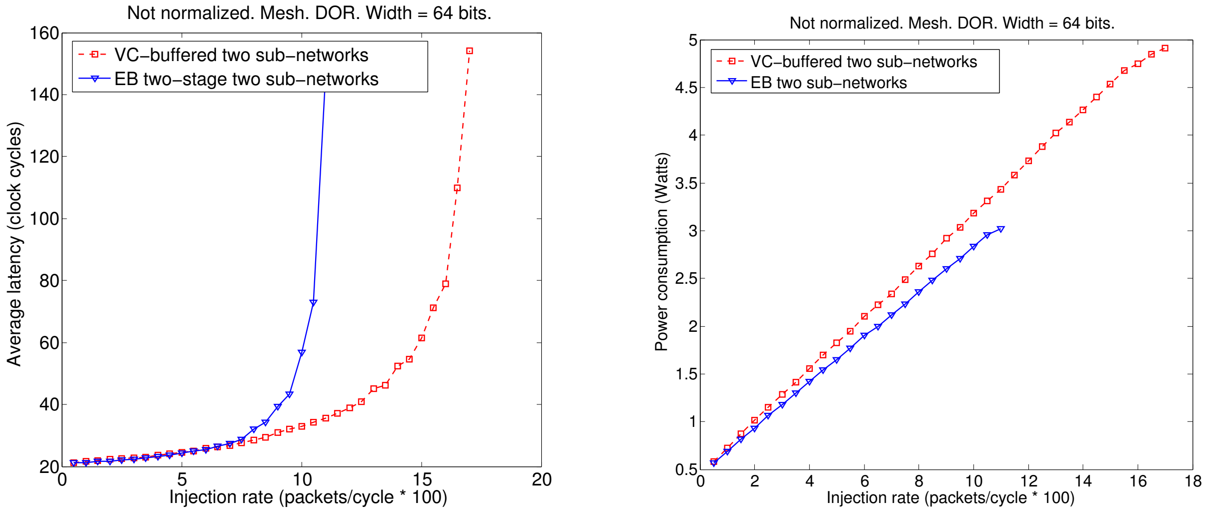


Fig. 11. Latency and power by increasing the injection rate for the 4×4 mesh under uniform random traffic.

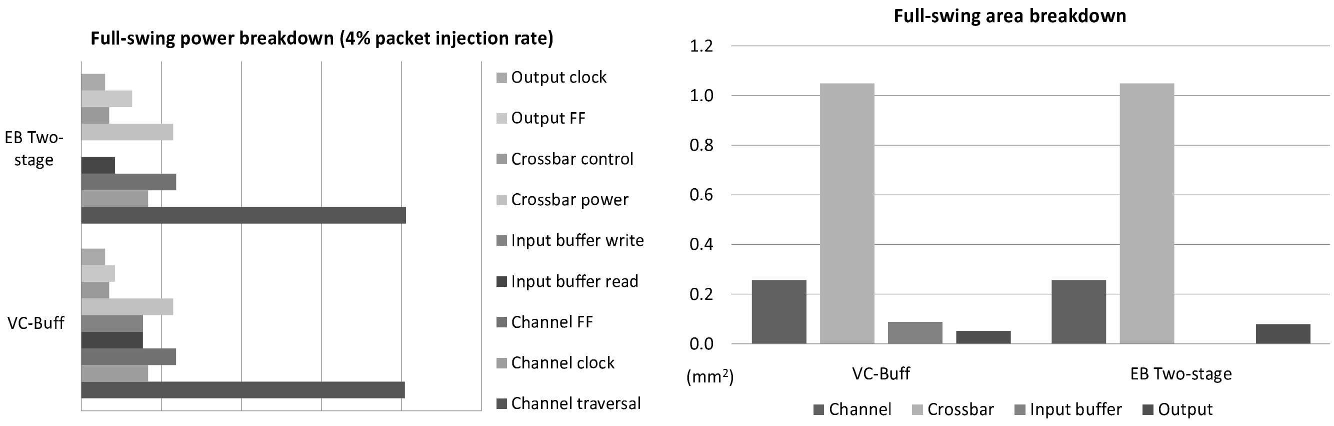


Fig. 12. Cost breakdowns for a DOR FBFly with 64-bit full-swing channels under uniform traffic.

in cycle time. The reduced cycle time enables the network to be clocked at a higher frequency, thus achieving higher throughput in absolute time, or lower zero-load latency if the pipeline stages in the VC router are increased.

The VC router cycle time is constrained by the VC and switch allocators. Increasing the number of VCs increases the complexity of the first stage. This shows the gain in cycle time of reducing router complexity by removing VCs and allocation. This is only slightly affected by the buffer implementation since the critical path begins at buffer read. However, that timing overhead remains considerably larger compared to EB read.

The amount of buffering in EB channels scales directly with channel length. Topologies with double the channel length have an increased throughput of 8-10% in the UGAL FBFly using the two-stage EB router. On the other hand, they have almost double the channel power for a total power increase of approximately 60%. This small change in throughput shows that the dominant factor affecting maximum throughput in EB networks is the contention in the bufferless routers. However, designers might still find it beneficial to add storage in EB channels depending on their topology, layout and router radix. This can be done by adding EBs, adding latches to existing EBs, or using repeaters for additional storage [20].

5.3 Buffer Cost Impact

Fig. 12 shows a cost breakdown for a FBFly using DOR and full-swing channels. DOR is chosen for fairness to have a single EB subnetwork and ensure that flits traverse the same paths in the EB and VC networks. Fig. 13 presents the same power breakdown for low-swing channels. *Channel traversal* refers to the power to traverse a segment with repeaters. For the EB network, *input buffer read* power is the traversal power for the intermediate EB shown in Fig. 2. Channel and switch area and power remain the same. The difference between the buffer power and the intermediate EB power is the amount saved by removing the buffers. Buffer power in the VC network is 15.5% with full-swing channels and 21.5% with low-swing channels of the overall power. Low-swing channels double the channel area due to differential signaling, making buffer area a smaller percentage of the overall area. They also reduce channel traversal power, making the buffers a more significant ratio of the network power. This increases the EB network's power gains from removing the buffers.

Removing router buffers provides power and area savings for EB networks. Those savings can be traded for a wider datapath. This way, power or area can become equal to the VC network. Then, performance and other aspects can be compared. Therefore, the increase in performance per unit area or power of EB networks is dictated by the area or power of the overall network that is consumed by the buffers.

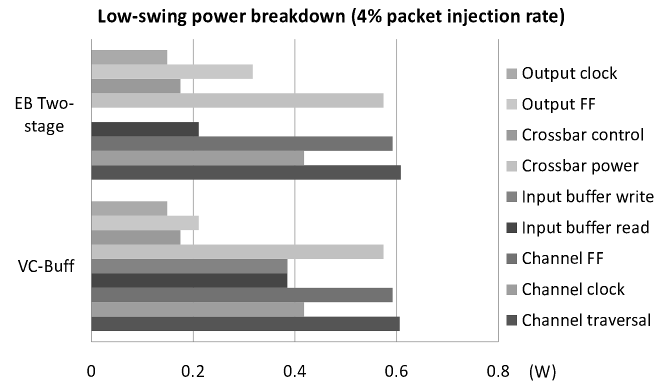


Fig. 13. Low-swing power breakdown. Same case as Fig. 12.

A comparison of EB and wormhole routers has shown that removing the buffers is the dominant factor for the EB network cost savings [9]. Since EB and wormhole routers have directly comparable complexity due to the lack of VCs, this isolates the contribution of buffer cost from the extra complexity of VC flow control. On the other hand, simplifying allocation primarily reduces cycle time.

The ratio of area and power consumed by the buffers depends on every part of the network. Therefore, optimizing various network components will make EB networks more attractive. Architectural choices also affect buffer power. For example, flits in multi-hop topologies traverse more routers by average than in topologies with express links.

Finally, buffer cost heavily depends on the implementation. The comparisons performed in our study would be more in favor of EB networks if we had assumed more expensive buffer implementations than our efficient custom SRAMs. However, we note that due to implementation or technology library constraints, some designers may be forced to use costly buffers. Finally, buffer cost can be reduced if the design is focused at some end of the operating spectrum. For example, empty buffer bypassing saves the majority of buffer dynamic power under low loads [18], [21].

5.4 EB Router Design Comparison

This Section compares the two EB router designs. Fig. 14 shows place and route results. These curves are not smooth because the software tools for the place and route flow use randomized algorithms with heuristics (such as simulated annealing) to perform optimizations on discrete values (such as cell sizing). The two-stage router has a cycle time reduced by 26% compared to the single-stage router. The single-stage router requires 19% less energy per transferred bit compared to the two-stage router. Finally, the single-stage router occupies 30% less area than the two-stage router.

The reduced area and energy of the single-stage router is due to the lack of pipeline overhead. Since the synchronization module of the two-stage router operates only on chosen output port bits, its contribution is small compared to the datapath. The increased energy cost of the two-stage router is attributed to the addition of the synchronization logic, splitting the intermediate register cell into two latch cells to implement the intermediate EB, and to the increased clock frequency, which forces the cells to have greater driving strength. This also affects the switch. The single-stage router is much simpler, reducing its energy consumption.

The lack of pipelining is also the reason that the single-stage router has a larger cycle time than the two-stage router. However, the increase is still less than a factor of two. The difference in cycle time has a small effect on occupied area because it only affects cell sizing and placement. Instead, the dominant factor is component complexity.

The two-stage router is constrained by the first stage only for datapath widths smaller than 47 bits. This means that further optimizations should focus on other aspects of the router for larger datapath widths. As the datapath width increases, the cycle times of all routers converge. This is because the switch dominates the cycle time at large widths and all routers use identical switches. Techniques such as switch slicing thus will allow the two-stage router to be clocked at smaller cycle times for large datapath widths. However, routers with large critical paths on the first stage will have their cycle times unaffected.

Fig. 15 shows a gate and cell count for the EB routers. The cell count of the intermediate EBs of the two-stage router includes the synchronization module logic. As shown, the various components of the two routers have a directly comparable number of gates and cells. However, the two-stage router has 39% more gates and 44% more cells due to the intermediate EB and synchronization module logic. Furthermore, the gate and cell count of the switch arbiters is low (61% fewer gates than the mux-based crossbar), illustrating the low arbitration complexity of EB routers.

As shown in Fig. 16(a), the single-stage router offers the smallest zero-load latency per unit throughput, on average. This is because of the single pipeline stage. Its effect is directly dependent on the average number of hops of our network, therefore especially important in our multi-hop 2D 8×8 mesh. However, the difference with the two-stage router significantly decreases when clocking each router at its maximum frequency, shown in Fig. 16(b), compared to clocking them at an equal frequency. This is because the network with the two-stage router also has higher-frequency channels, which have lower latency in absolute time. However, these results rely on the channel latency in clock cycles remaining equal when increasing the clock frequency (the number of retiming elements remains constant). While this is true for our clock frequencies and physical channel lengths, other network settings might find that increasing the clock frequency also increases the channel latency in clock cycles. In that case, the single-stage router will provide an additional reduction in latency compared to the two-stage router. Networks with a small average hop count or channels with many pipeline stages will have their latency influenced more by channels than by routers.

Since the routers were placed and routed for maximum clock frequency, their occupied areas remain constant regardless of the clock frequency they operate at. At an equal clock frequency, the single-stage router provides the most throughput per unit area because it occupies the least area. Therefore, the single-stage router has an increased datapath width compared to the two-stage router occupying the same area and provides a higher throughput. However, if the routers operate at their maximum frequencies, the two-stage router provides more throughput per unit area due to its reduced cycle time.

The two-stage router is the optimal choice for area and consumes the least energy. However, it is closely followed by the single-stage router which carries cycle time, latency and area benefits. Designs with zero-load latency in mind should take into account the average number of hops and the effect on channel latency in clock cycles when applying the two-stage router's maximum clock frequency. Network designs which would clock all routers under the same frequency have the

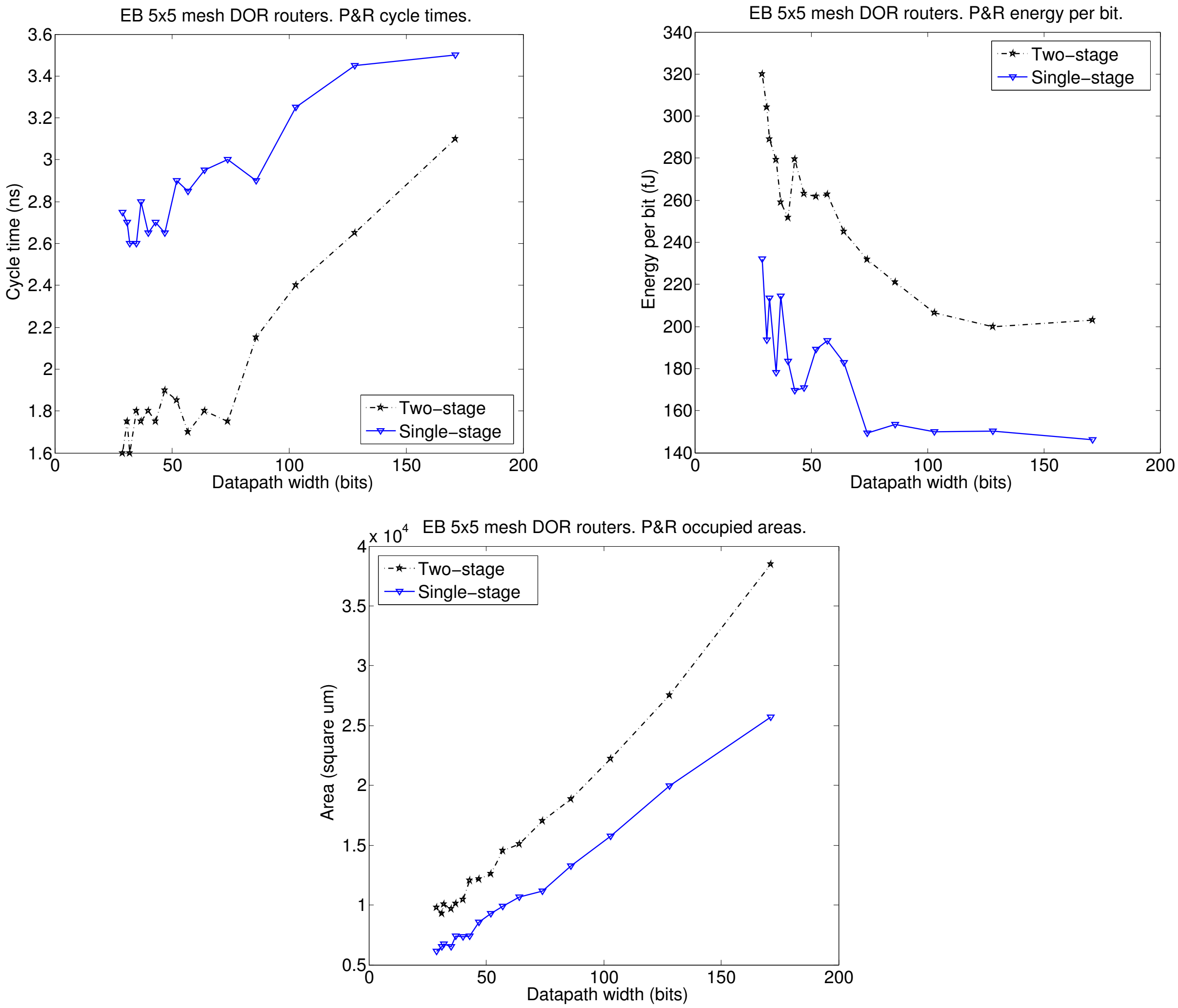


Fig. 14. Place and route EB router implementation results.

TABLE 3
Optimal router choice for our 8×8 2D mesh depending on design priority.

Priority	Router choice
Operate at maximum frequencies	
Area	Two-stage
Energy	Single-stage
Latency	Single-stage (depends on effect on channels)
Operate at the same frequency	
Area	Single-stage
Energy	Single-stage
Latency	Single-stage

single-stage router as their optimal choice for area and latency. Examples of such designs can be systems-on-chip, which may not require a higher clock frequency or may keep the network clock synchronized to a slower system-wide clock to avoid multiple clock domains. Table 3 summarizes which router is the optimal choice depending on design priorities.

Using the optimal EB router and the shortest cycle time for each comparison, EB networks provide an up to 45% shorter cycle time, 16% increase in throughput per unit power, or 22% increase in throughput per unit area compared to VC networks.

5.5 Hybrid EB-VC Networks

Fig. 17 shows Pareto-optimal curves for a network with the hybrid EB-VC router. The number of blocking cycles before a flit is drained (*BL CYCL*) is set to 25. There are 8 traffic classes with 8 buffer slots per class. Clock frequencies are equal. Only results for the single-stage EB router are included because it's the optimal choice for equal clock frequencies, as shown in Table 3. The hybrid router provides 21% more throughput per unit power compared to the VC router, and 12% more compared to the single-stage EB router. On the other hand, the VC router provides 41% more throughput per unit area while the single-stage provides 49% more. Zero-load latency is equal for the hybrid and VC routers.

Smaller *BL CYCL* values reduce the throughput per unit power of the hybrid router. For instance, with *BL CYCL* set to 4, the EB router provides 5% more throughput per unit power. On the other hand, increasing *BL CYCL* above 25 has no effect on throughput per unit power, but increases the worst-case blocking latency. Different values of *BL CYCL* do not considerably affect maximum throughput, because blocking a flit from a different class even for a few cycles penalizes throughput significantly. However, they do affect how often the buffer of the hybrid router is used. For small values, the buffer is used even in the common case, while increasing *BL*

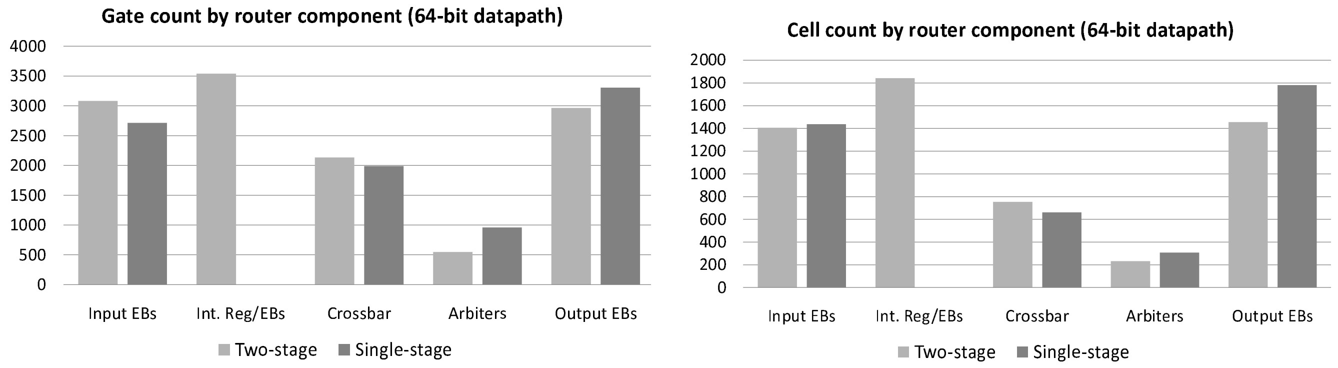


Fig. 15. Router place and route gate and cell breakdown.

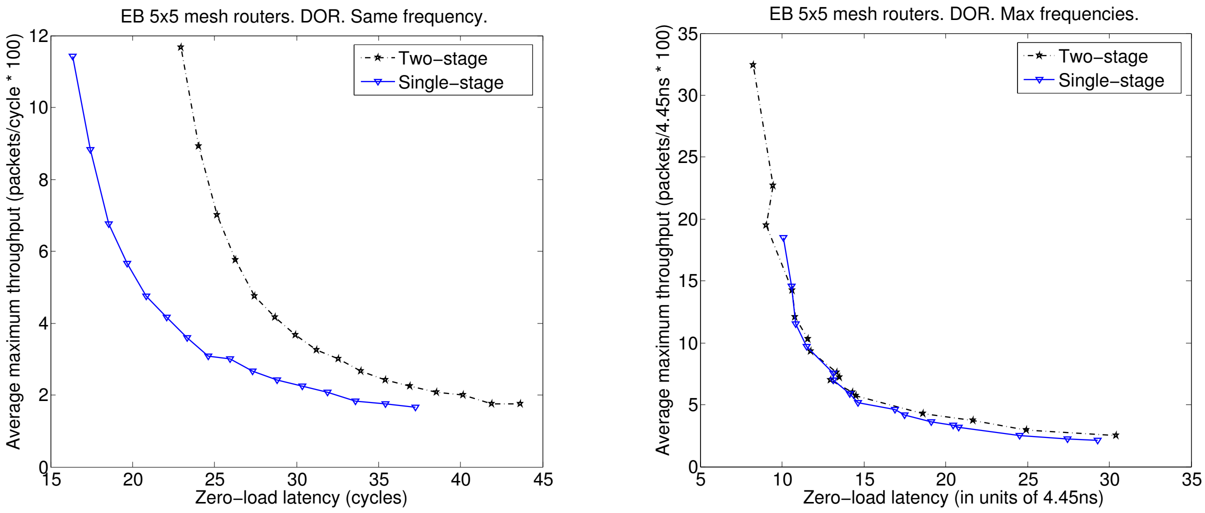


Fig. 16. EB router latency-throughput comparison.

CYCL above 25 makes no difference because buffers are still used only in situations which would otherwise result in a deadlock and to alleviate severe contention. Therefore, small *BL CYCL* values make the hybrid router's power consumption comparable to the VC router's, but without a proportional increase in throughput due to head-of-line blocking.

Increasing the number of buffer slots per traffic class to 16 makes the EB router marginally (2%) more throughput per unit power efficient than the hybrid router. While buffers can hold more flits, this rarely happens because by the time 15 other flits of the class being drained arrive, the oldest drained flit has traversed the switch. Also, increasing the buffer size increases the energy cost for accessing it. On the other hand, increasing the number of traffic classes to 16 reduces throughput per unit power of the hybrid router only marginally (2%) compared to the VC. This percentage increases to 8% for small values of *BL CYCL*. This is because increasing traffic classes increases the probability of blocking in the EB channels. This causes more flits to be drained, increasing power. However, area efficiency for the hybrid router is also marginally increased (2%) because more draining increases throughput without affecting area.

The hybrid EB-VC router increases throughput per unit power because it uses buffers only to alleviate head-of-line blocking and to resolve cases which would otherwise result in a deadlock. In the common case, buffers are not used and so the hybrid router is almost as energy efficient as the two-stage EB router. To retain the power efficiency of EB networks, EBs remain the primary means of buffering. To accomplish this, *BL CYCL* should not be small. We note that our experiments used a low-power library, which has negligible leakage power. Increasing leakage would reduce the power efficiency of the hybrid router compared to EB routers. However, this effect would be very small because in a similar network as our 8×8 2D mesh, buffer leakage was only 1.5% of the overall network power, which was dominated by channel power [21]. However, the buffers occupy area and may extend the critical path.

6 Related Work

Various control logic blocks have been proposed for EBs. A few implementations have been proposed for the control logic and the ready-valid handshake used by EB NoCs described in this paper [6], [22], [23]. Alternative and slightly more costly EB implementations add an auxiliary FF to every pipeline FF [24], therefore doubling the amount of latches in the channels. The auxiliary FF is used to store incoming data when the pipeline is stalled. Furthermore, EB implementation variations for various kinds of pipelines have been studied, such as without a valid signal routed downstream [25]. These studies shown how EBs can be made scannable [25] and how to connect one EB to multiple EBs [23]. Finally, another EB design uses repeaters to store data [20]. This design requires very careful implementation and faces charge sharing and leakage current issues originating from storing data using repeater parasitic capacitance for an indefinite amount of time.

Past NoCs have used the EB design with the main and auxiliary FFs [26] as well as the EBs based on repeaters [27].

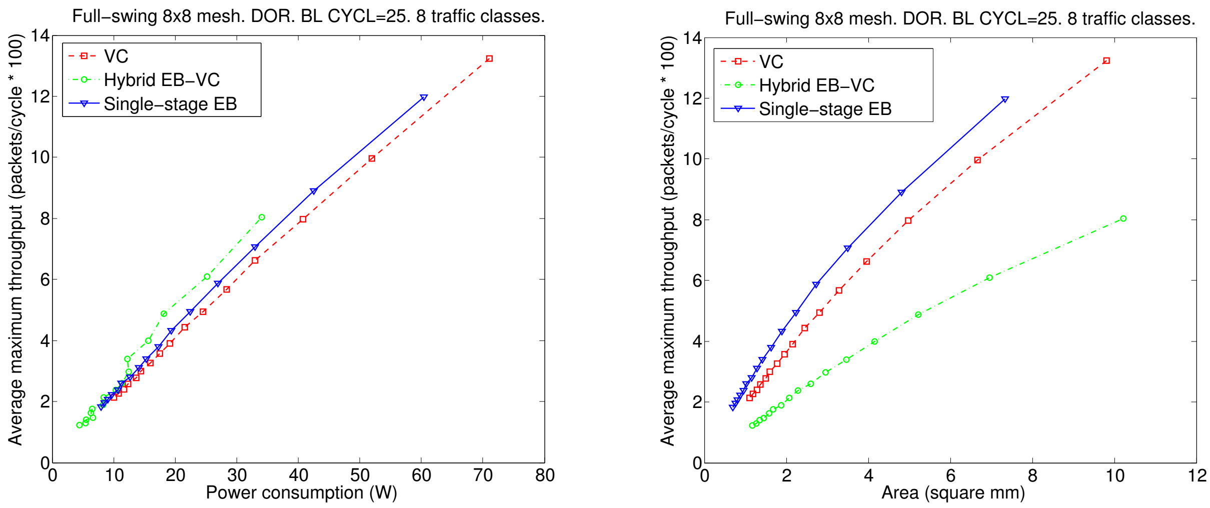


Fig. 17. Hybrid EB-VC Pareto-optimal curves.

Both of these proposals make full use of router input buffers in all cases, and therefore retain the associated complexity and costs. However, the EB flow control described in this paper uses EBs instead of input buffers, and significantly simplifies router design by removing credits and VCs. If a large number of traffic classes are required, input buffers and VCs are reintroduced but the buffers are still not used in the common case.

Past research has proposed other bufferless flow control techniques. Compressionless routing relies on feedback from the network sent back to the network interfaces [28]. Circuit-switched networks [29] allocate resources along the path leading to the destination before transmitting the packet. A hybrid packet-connected circuit has also been proposed, requiring the routers to have buffer space only for request packets [30]. Connection-based routing allows intermediate nodes to tear down and modify end-to-end virtual circuits [31]. Finally, deflection [32], [33] or dropping [34], [35] networks handle contention by dropping or deflecting flits to a free output. We improve on them by providing buffering to the network without adding router buffers, to increase performance and avoid complications such as large latency variance [21].

While UGAL was used in this study as an example, other adaptive routing algorithms are equally applicable. Furthermore, EB networks can use regional congestion awareness [36], as well as propagating control information to create reservations or pre-compute arbiter decisions [7], [37]. Applying these and other orthogonal optimizations to EB networks is beyond the scope of this study.

7 Conclusions

EB flow control uses the pipeline FFs in the channels for buffering flits. Therefore, channels act as distributed FIFOs and router buffers are no longer required. Switch and VC allocators are replaced by a switch arbiter for every output. This significantly simplifies router design. EB flow control uses separate physical subnetworks for traffic separation. However, this option becomes inefficient for a large number of traffic classes. To make EB networks efficient for a large number of classes, in this paper we propose a hybrid EB-VC router which has input buffers only used to drain flits in case of deadlock or heavy head-of-line blocking. Thus, in the common case hybrid routers operate as EB routers, and as VC routers otherwise.

By using the optimal EB router and shortest cycle time for each comparison, a 2D mesh EB network provides 16% more throughput per unit power, 22% more throughput per unit area or has an up to 45% shorter cycle time compared to a similar VC network. Gains for EB networks are proportional to the area and power cost of the buffers in VC networks. Design simplicity from removing VCs, allocators and credits primarily affects cycle time, but is not a major contributor to the area and power efficiency increase [9]. Hybrid EB-VC routers offer 21% more throughput per unit power than VC routers, and 12% more than single-stage EB routers, because input buffers are used only to resolve deadlocks and alleviate head-of-line blocking. However, hybrid EB-VC routers carry the area and timing overheads of input buffers.

The choice of router among the three described in this paper should be based on design constraints. Table 2 identifies the optimal EB router without input buffers, depending on design priorities. To provide traffic classes, EB networks without input buffers and duplicate physical subnetworks should be considered first, because of their small area and complexity. The number of traffic classes above which this option becomes infeasible depends significantly on a variety of factors and therefore should be studied for specific chip designs and implementation technologies. If more classes than that number are desired, the hybrid EB-VC router should be used. Finally, note that the hybrid EB-VC router can be more energy efficient compared to EB routers without input buffers, because input buffers alleviate head-of-line blocking. Therefore, designs focusing on energy with few constraints on other cost factors should consider the hybrid router.

The simplicity of EB networks makes it easier to reach an optimal design. Therefore, EB flow control provides a simple and cost-efficient network to satisfy modern demands.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant CCF-0702341, in part by the National Security Agency under Contract H98230-08-C-0272 and in part by the Robert Bosch Stanford Graduate Fellowship.

References

[1] W. J. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” in *DAC ’01: Proceedings of the 38th annual design automation conference*, 2001.

[2] M. B. Taylor, W. Lee, J. Miller, D. Wentzlaff, I. Bratt, B. Greenwald, *et al.*, “Evaluation of the RAW microprocessor: An exposed-wire-delay architecture for ILP and streams,” in *ISCA ’04: Proceedings of the 31st annual international symposium on Computer architecture*, 2004, p. 2.

[3] P. Gratz, C. Kim, R. McDonald, S. Keckler, and D. Burger, “Implementation and evaluation of on-chip network architectures,” in *ICCD ’06: international conference on computer design*, 2006, pp. 477–484.

[4] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, “Orion 2.0: A fast and accurate NoC power and area model for early-stage design space exploration,” in *DAC ’09: Proceedings of the 46th annual conference on design automation*, 2009, pp. 423–428.

[5] J. Balfour and W. J. Dally, “Design tradeoffs for tiled CMP on-chip networks,” in *ISCA ’06: Proceedings of the 20th annual international conference on supercomputing*, 2006.

[6] G. Michelogiannakis, J. Balfour, and W. J. Dally, “Elastic buffer flow control for on-chip networks,” in *HPCA ’09: Proceeding of the 15th international symposium on high-performance computer architecture*, 2009, pp. 151–162.

[7] R. Mullins, A. West, and S. Moore, “Low-latency virtual-channel routers for on-chip networks,” in *ISCA ’04: Proceedings of the 31st annual international symposium on computer architecture*, 2004, p. 188.

[8] G. Michelogiannakis and W. J. Dally, “Router designs for elastic buffer on-chip networks,” in *SC ’09: Proceedings of the conference on high performance computing networking, storage and analysis*, 2009, pp. 1–10.

[9] G. Michelogiannakis, D. Becker, and W. Dally, “Evaluating elastic buffer and wormhole flow control,” *IEEE transactions on computers*, vol. 60, no. 6, pp. 896 –903, june 2011.

[10] M. Galles, “Spider: A high-speed network interconnect,” *IEEE Micro*, vol. 17, no. 1, pp. 34–39, 1997.

[11] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip,” *ACM computing surveys*, vol. 38, no. 1, p. 1, 2006.

[12] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[13] F. Gilabert, M. E. Gomez, S. Medardoni, and D. Bertozzi, “Improved utilization of NoC channel bandwidth by switch replication for cost-effective multi-processor systems-on-chip,” in *NOCS ’10: Proceedings of the fourth international symposium on networks-on-chip*, 2010, pp. 165–172.

[14] L. Ni and P. McKinley, “A survey of wormhole routing techniques in direct networks,” pp. 492–506, 2000.

[15] A. Singh, “Load-balanced routing in interconnection networks,” PhD in Electrical Engineering, Stanford University, 2005.

[16] J. Kim, W. J. Dally, and D. Abts, “Flattened butterfly: a cost-efficient topology for high-radix networks,” in *ISCA ’07: Proceedings of the 34th annual international symposium on computer architecture*, 2007.

[17] J. Kim, W. J. Dally, S. Scott, and D. Abts, “Technology-driven, highly-scalable dragonfly topology,” in *ISCA ’08: Proceedings of the 35th annual international symposium on computer architecture*, 2008, pp. 77–88.

[18] H. Wang, L.-S. Peh, and S. Malik, “Power-driven design of router microarchitectures in on-chip networks,” in *MICRO ’03: Proceedings of the 36th annual international symposium on Microarchitecture*, 2003.

[19] R. Ho, K. Mai, and M. Horowitz, “Efficient on-chip global interconnects,” in *Symposium on VLSI circuits*, 2003, pp. 271–274.

[20] M. Mizuno, , W. J. Dally, and H. Onishi, “Elastic interconnects: repeater-inserted long wiring capable of compressing and decompressing data,” in *ISSCC ’01: Proceedings of IEEE international solid-state circuits conference*, 2001, pp. 346–347, 464.

[21] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis, “Evaluating bufferless flow control for on-chip networks,” in *NOCS ’10: Proceedings of the fourth international symposium on networks-on-chip*, 2010, pp. 9–16.

[22] J. Cortadella, M. Kishinevsky, and B. Grundmann, “Synthesis of synchronous elastic architectures,” in *DAC ’06: Proceedings of the 43rd annual design automation conference*, 2006, pp. 657–662.

[23] H. Jacobson, P. Kudva, P. Bose, P. Cook, S. Schuster, E. Mercer, *et al.*, “Synchronous interlocked pipelines,” in *Proceedings of the 8th international symposium on asynchronous circuits and systems*, 2002, pp. 3 – 12.

[24] L. P. Carloni, “The role of back-pressure in implementing latency-insensitive systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 146, pp. 61–80, 2006.

[25] D. Harris and T. P. S. U. C. Archives, “Local stall propagation,” 2000. [Online]. Available: http://citeseer.ist.psu.edu/506008.html

[26] N. Concer, M. Petracca, and L. P. Carloni, “Distributed flit-buffer flow control for networks-on-chip,” in *Proceedings of the 6th IEEE/ACM/IFIP international conference on hardware/software codesign and system synthesis*, ser. CODES+ISSS ’08, 2008, pp. 215–220.

[27] A. Kodi, A. Sarathy, and A. Louri, “Design of adaptive communication channel buffers for low-power area-efficient network-on-chip architecture,” in *ANCS ’07: Proceedings of the 3rd symposium on architecture for networking and communications systems*, 2007, pp. 47–56.

[28] J. H. Kim, Z. Liu, and A. A. Chien, “Compressionless routing: a framework for adaptive and fault-tolerant routing,” *SIGARCH computer architecture news*, vol. 22, no. 2, pp. 289–300, 1994.

[29] J. Liu, L.-R. Zheng, and H. Tenhunen, “A guaranteed-throughput switch for network-on-chip,” in *SOC ’03: Proceedings of the international symposium on system-on-chip*, 2003, pp. 31–34.

[30] D. Wiklund and D. Liu, “SoCBUS: Switched network-on-chip for hard real time embedded systems,” in *IPDPS ’03: Proceedings of the 17th international symposium on parallel and distributed processing*, 2003, p. 78.1.

[31] Y. Turner and Y. Tamir, “Deadlock-free connection-based adaptive routing with dynamic virtual circuits,” *Journal of parallel and distributed computing*, vol. 67, no. 1, pp. 13–32, 2007.

[32] T. Moscibroda and O. Mutlu, “A case for bufferless routing in on-chip networks,” *SIGARCH computer architecture news*, vol. 37, no. 3, pp. 196–207, 2009.

[33] Z. Lu, M. Zhong, and A. Jantsch, “Evaluation of on-chip networks using deflection routing,” in *Proceedings of the 16th GLSVLSI*, 2006.

[34] C. Gómez, M. E. Gómez, P. López, and J. Duato, “Reducing packet dropping in a bufferless NoC,” in *Proceedings of the 14th international Euro-par conference on parallel processing*, 2008.

[35] M. Hayenga, N. E. Jerger, and M. Lipasti, “Scarab: a single cycle adaptive routing and bufferless network,” in *MICRO 42: Proceedings of the 42nd annual international symposium on microarchitecture*. New York, NY, USA: ACM, 2009, pp. 244–254.

[36] P. Gratz, B. Grot, and S. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in *HPCA ’08: Proceeding of the 14th international symposium on high-performance computer architecture*, 2008, pp. 203 –214.

[37] L.-S. Peh and W. J. Dally, “Flit-reservation flow control,” in *HPCA ’00: Proceeding of the 6th international symposium on high-performance computer architecture*, 2000.

George Michelogiannakis is a Ph.D. candidate. He received his B.Sc. and M.Sc. with honors from the University of Crete, Greece. At Stanford, he was selected for the Stanford Graduate Fellowship. His main research interests and his research publications are on flow control and buffer usage for on-chip networks.

William J. Dally is Chief Scientist and Senior Vice President of Research at NVIDIA and the Willard R. and Inez Kerr Bell Professor of Engineering at Stanford University. He is a Member of the National Academy of Engineering, a Fellow of the IEEE and the ACM, and a Fellow of the American Academy of Arts and Sciences. He has received numerous honors including the IEEE Seymour Cray Award and the ACM Maurice Wilkes award. He has published over 200 papers and holds over 50 issued patents.