Stanford University Concurrent VLSI Architecture Memo 124

Elastic Buffer Networks-on-Chip

George Michelogiannakis[*], James Balfour and William J. Dally

Concurrent VLSI Architecture Group

Computer Systems Laboratory

Stanford University, Stanford, CA 94305

Email: {mihelog,jbalfour,dally}@stanford.edu

Abstract

This paper presents elastic buffers (EBs), an efficient flow-control scheme that uses the storage already present in pipelined channels in place of explicit input virtual-channel buffers (VCBs). With this approach, the channels themselves act as distributed FIFO buffers under congestion. Without VCBs, and hence virtual channels (VCs), deadlock prevention is achieved by duplicating physical channels. We develop a channel occupancy detector to apply universal globally adaptive load-balancing (UGAL) routing to load balance traffic in networks using EBs. Using EBs results in up to 12 % (17 % for low-swing channels) improvement in peak throughput per unit power compared to a network using VC flow control. The power saved by eliminating VCBs is used to make channels wider, providing increased throughput that more than offsets the degradation caused by increased blocking without VCs. EB networks have identical zero-load latency to VC networks and have considerably simpler router logic as a VC allocator is not required.

1 Introduction

Semiconductor technology scaling allows more processing and storage elements to the integrated on the same die. Networks-on-chip (NoCs) provide a scalable communication infrastructure [5, 7] for connect-

[*]George Michelogiannakis, Robert Bosch Stanford Graduate Fellow, Department of Electrical Engineering, Stanford University.

ing these elements. The on-chip environment makes buffering and flip-flops (FFs) costly. Thus, reducing buffering or making its implementation more efficient increases design efficiency.

This paper presents elastic buffers (EBs), an efficient flow-control scheme that uses the storage already present in pipelined channels in place of explicit input virtual channel buffers (VCBs). Removing VCBs removes the area and power cost they impose. However, this also removes virtual channels (VCs) from the network. Duplicate physical channels must be used in the same way as VCs to prevent deadlocks and to differentiate between traffic classes. Dividing into sub-networks has been shown to increase performance and power efficiency [1], which we find to hold for elastic buffered networks (EBNs).

We examine two router architectures – the buffered and the un-buffered crossbar routers – and conclude that the latter is the optimal choice for packets which do not fit into crosspoint buffers. We use universal globally adaptive load-balancing (UGAL) routing in a flattened butterfly (FBFly) [11] topology to balance network load. For this purpose, we develop a channel load detector to be used in place of credit count.

For a FBFly with UGAL routing, performance and power efficiency is almost equal for EBNs and virtual channel networks (VCNs) for full-swing channels, with a 4 % gain for EBNs for low-swing channels. EBN gains in 2D mesh networks reach 12 %, increasing to 17 % for low-swing channels. EBN gains depend on the fraction of cost represented by VCBs. Since EBN channels need to be made wider to make up for the removal of VCBs, EBNs tend to occupy more area. For a fixed area budget, channel widths are very close thus making VCNs more area efficient. Zero-load latency is equal between VCNs and EBNs.

The rest of this paper is organized as follows. Section 2 discusses the basic building blocks of EBNs as well as deadlock, topology and adaptive routing. Section 3 presents performance, area and power evaluations. Section 4 discusses EBNs and analyzes trade-offs. Finally, section 6 concludes and discusses related work.

2 Elastic Buffer Networks

In this section we present EB channels and routers, the fundamental EBN building blocks. We then discuss channel duplication, deadlock avoidance and adaptive routing.

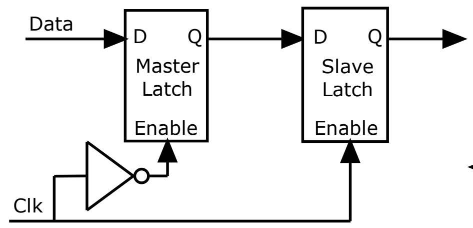
2.1 Elastic Channels

Figure 1(a) illustrates a D FF implemented with master-slave latches. By adding a control logic to drive the two latch enable inputs separately to use latches for buffering, the FF becomes an EB FIFO with two storage slots. This is illustrated in Figure 1(b). FFs of other types will require a different design. However, master-slave FFs have been found to be an efficient and popular design [18].

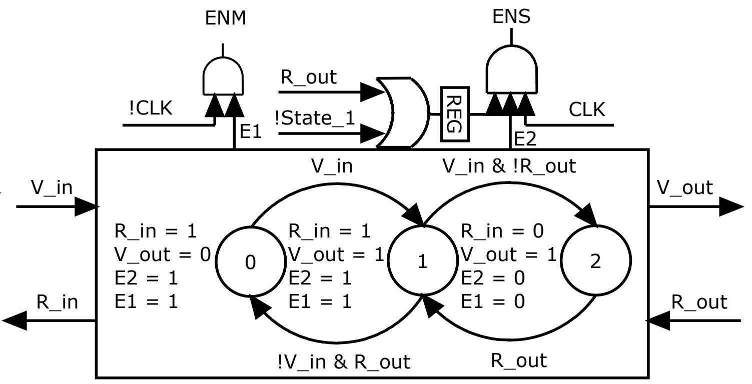
EB channels use many EBs to form a distributed FIFO. FIFO storage can be increased by adding additional latches to EBs or by using repeater cells for storage [14]. EBs use a ready-valid handshake to safely advance data. Ready outputs indicate that the EB has at least one empty storage slot to latch more data. Valid outputs indicate that the data currently being driven is valid. The control logic samples incoming control signals at rising clock edges to determine if data has advanced to or from the EB. Asserted ready and valid signals between two EBs indicate that data has advanced. This timing mandates having at least two storage slots per clock cycle's delay to avoid creating unnecessary pipeline bubbles.

The three-state finite state machine (FSM) implementing the EB control logic is illustrated in Figure 1(c). A logic diagram implementing it is shown in Figure 1(d). The control logic generates signals which qualify the clock to the latches in the same manner as FFs. Data is latched in the master latch at the end of the cycle, and advances to the slave at the beginning of the next cycle. If during the previous cycle the incoming ready was de-asserted and the state remains at 1, data is already in the slave latch and its enable input needs to be disabled. The extra logic handling this case needs to have its output set up before E2 arrives, to prevent glitches that could corrupt the slave latch. EB control logic cost is 3 FFs and 11 logic gates. Since flow-control is applied at a flit granularity, control logic is amortized over the width of the channel. The channel datapath is unaffected since no transistors are added to it.

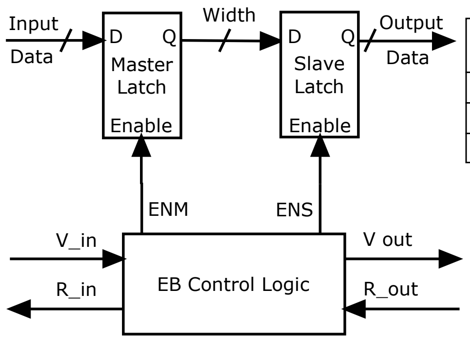
EB control logic adds overhead to the setup and clock-to-q time. If it proves to be a problem in reaching the desired network clock frequency, the ready and valid wires can be optimized for delay by promoting them to higher metal layers which have a smaller RC delay, or by engineering them for delay using more aggressive repeaters. If we assume 1 mm wires, a 2 fanout-of-four (FO4) FF clock-to-q delay and total of 3 FO4 delay for the FSM critical path composed of three gates, in our environment explained in section 3.1, that 5 FO4 overhead can be hidden by targeting a ready and valid delay of 400 ps/mm instead of 500 ps/mm, with the minimal being 150 ps/mm. Furthermore, hiding the control logic's extra clock-to-q delay is achieved by skewing the ENM and ENS gate clock to arrive earlier than the clock channel EBs are



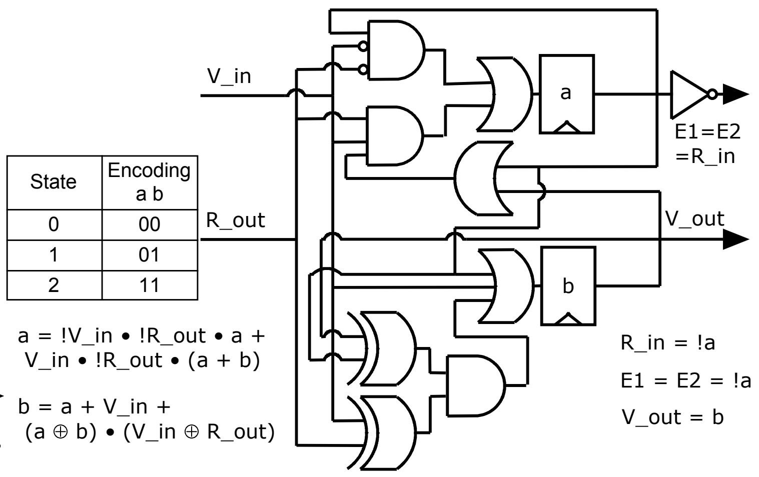
(a) A D FF implemented with a master and a slave D latch.



(c) Expanded view of the EB control logic as a synchronous FSM for two-slot EBs. ENM and ENS cannot be asserted at the same time. If ready_out was de-asserted during state 1, at the next cycle the slave enable input must be de-asserted. E2 in state 0 can be 0 to save power, but 1 simplifies logic.



(b) An EB is a FF whose latches are controlled by the EB control logic and used for buffering.



State	Encoding a b
0	00
1	01
2	11

$a = !V\_in \bullet !R\_out \bullet a + V\_in \bullet !R\_out \bullet (a + b)$

$b = a + V\_in + (a \oplus b) \bullet (V\_in \oplus R\_out)$

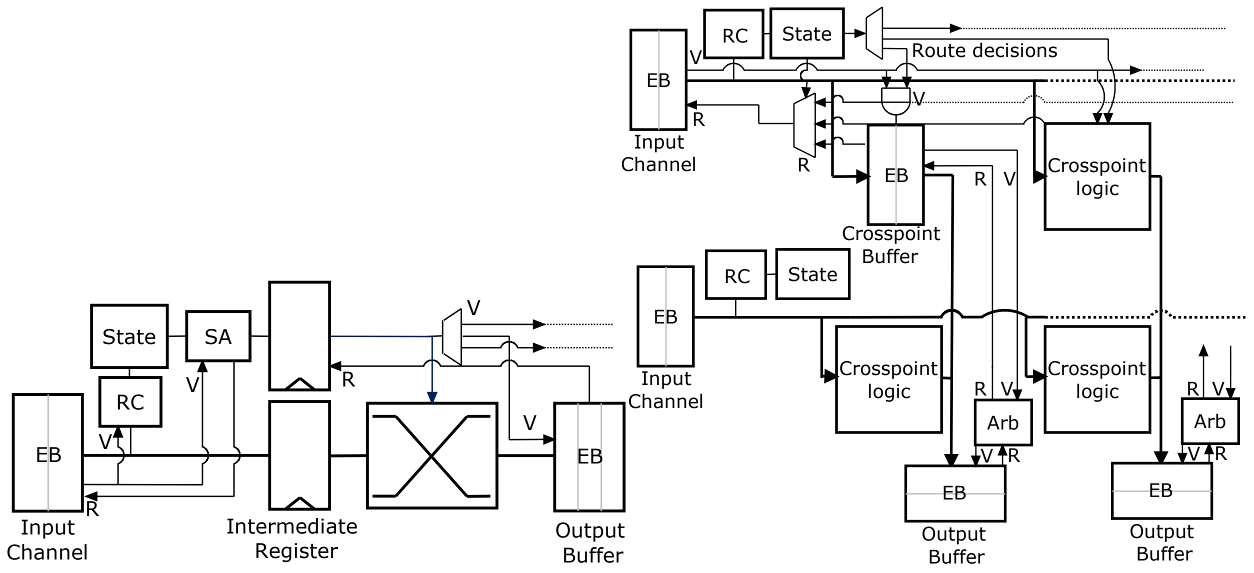
(d) Logic diagram of the EB control FSM with 2 FFs and 8 gates.

Figure 1: EB channels.

synchronized to, to provide enough time to the AND gates. The FSM clock needs to arrive even earlier to account for the FSM's clock-to-q delay in driving its outputs.

2.2 Router Architecture

In this work we focus on the unbuffered crossbar router. We also consider the buffered crossbar router. They are illustrated in Figure 2. Unlike input-queued VC routers [15], EBN routers lack input buffers and thus do not have VCs and VC allocators, simplifying their design. Credits are not used to track occupancy at the downstream router. The ready-valid handshake is used instead to determine when output EBs are ready to receive flits as well as to detect valid flits arriving at the inputs. Both EBN routers consist of two stages. Allocation or arbitration in both EBN routers is performed on a per-packet basis, to prevent the interleaving deadlock described in section 2.5.



(a) Unbuffered crossbar elastic router. Routing decisions are stored in state registers to be used for non-head flits. Only one input and one output are illustrated.

(b) Buffered crossbar router. All crosspoints are the same, but only the upper left is illustrated in detail. Crossbar allocation is replaced with output arbiters. Two inputs and two outputs are illustrated.

Figure 2: Two approaches for EB routers.

The unbuffered crossbar router is presented in Figure 2(a). The first stage is composed of routing computation and switch allocation. Routing decisions are stored when head flits are routed and used to forward non-head flits. When a flit wins allocation, a ready signal is driven back to the input channel and the flit advances to the intermediate pipeline register. During the next cycle, the flit traverses the switch and is stored into the output EB. From there it proceeds along the output channel as already described in section 2.1. The switch allocator observes the output EB ready signals to determine which output can accept flits. An extra storage slot in output EBs is required to cover for the latency due to allocation being performed one cycle in advance of switch traversal. Outputs de-assert their ready signals if they have less than two free slots.

The buffered crossbar router is shown in Figure 2(b). Incoming flits are stored at the appropriate crosspoint for their desired output. At the second stage, outputs arbitrate among all the valid crosspoints and store flits into the output's EB. The ready-valid handshake is used to facilitate movement in both stages.

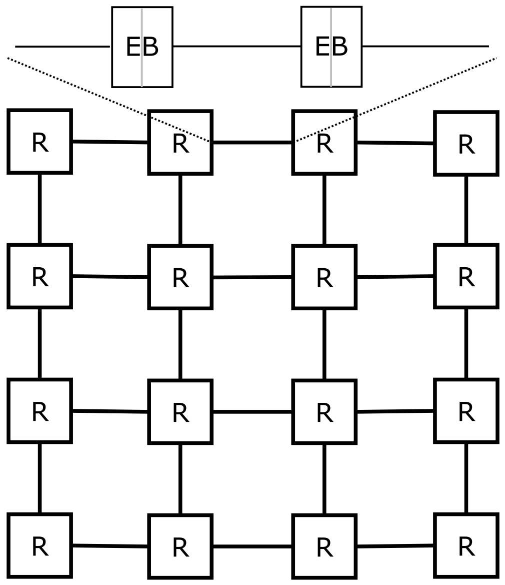
The buffered crossbar router appears attractive, performing considerably better if crosspoints can fully fit packets to prevent blocked packets from blocking preceding packets travelling through the same input. However, this is an unrealistic assumption. Moreover, while this router features arbiters instead of an allocator, thus simplifying router design, designing and building the crossbar is challenging and costly due to the large and expensive crosspoints, as well as the large number of ready-valid control signals throughout the crossbar. This effect becomes worse when trying to design a low-swing crossbar. Therefore, even though the buffered crossbar router can be beneficial for short packets or if allocators dominate the critical path, this study focuses on the unbuffered crossbar router.

2.3 Topologies

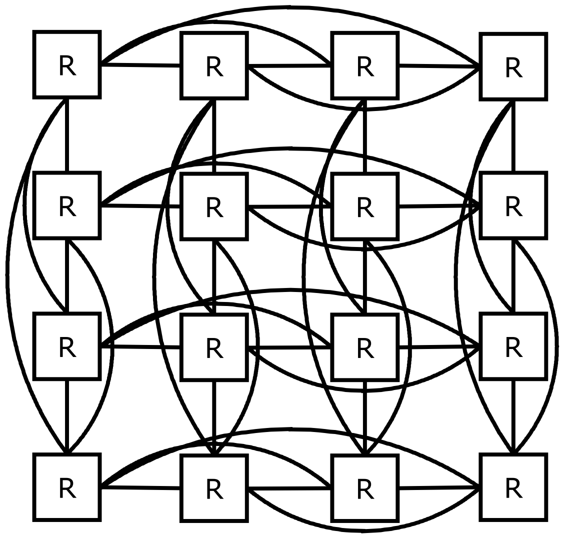
This work focuses on the 2D mesh and 2D FBFly [11] topologies. The 2D mesh represents a common topology choice due to its simplicity. The FBFly topology is similar to the 2D mesh, except that every router connects to every other router in the same axis. The two topologies are illustrated in Figure 3. Topologies with unequal channel lengths, such as the FBFly, force VCN routers to either have a different amount of buffering according to their location to account for the round-trip delay thus increasing their design complexity, or have large enough buffers to accommodate for the longest channels. EBNs avoid this because they do not use credits.

To use adaptive routing in EBNs, we define traffic classes to prevent cyclic dependencies, as discussed in section 2.5. Section 2.6 explains that the most efficient choice is by duplicating sub-networks. This way, we apply UGAL routing in the FBFly, as analyzed in section 2.4.

A single FBFly router is replaced by a router for each of the two sub-networks, as illustrated in Figure 4(a). Non-minimal routers connect to other non-minimal routers in the same Y axis by Y' channels. They connect to the minimal routers in the same X axis by X' channels. This way, flits can traverse to the minimal sub-network without taking a hop they would not otherwise take. There is also a self-channel which connects a non-minimal router with the minimal router with the same coordinates. Minimal routers connect to other minimal routers in a FBFly fashion. Traffic sources are connected to the sub-network used for non-minimal traffic, whereas destinations are connected to the minimal traffic sub-network. A complete 3×3 UGAL FBFly is illustrated in Figure 4(b).

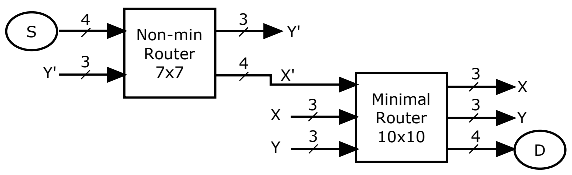


(a) A 2D 4×4 mesh. Channels have a latency of 2 clock cycles, thus two EBs.

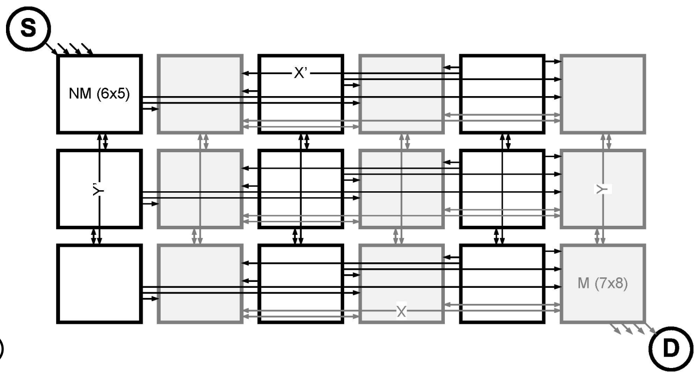


(b) A 2D 4×4 FBFly. Every router connects to every other in the same axis. Short channels have a latency of 2 clock cycles, medium of 4, and long 6.

Figure 3: The two topologies we focus on for this study.



(a) A single router is replaced by one for each sub-network. X’ channels connect non-minimal routers to minimal. The self-channel connects the two routers with the same coordinates. Illustrated for a 4×4 UGAL FBFly.



(b) A 3×3 UGAL FBFly. Y’ are non-minimal router (NM) channels in the Y axis. X and Y connect minimal routers (M – shaded).

Figure 4: The UGAL FBFly. Two traffic classes are needed to distinguish between traffic routed minimally to their destination (minimal), and traffic travelling towards their randomly chosen intermediate destination (non-minimal).

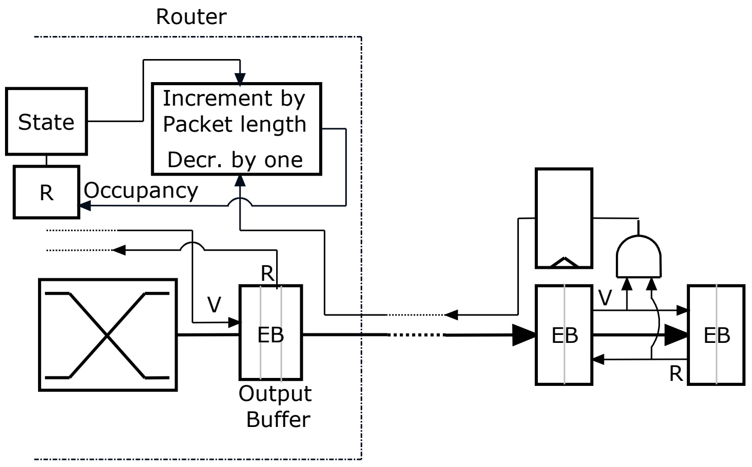


Figure 5: An AND gate to detect flits departing from the region of interest. One output is illustrated. The occupancy is incremented by the length of each packet routed to that output.

2.4 Routing

Adaptive routing algorithms balance network load to avoid performance degradation under adversarial traffic patterns. As an application of this, we use UGAL [17] adaptive routing on the FBFly topology presented in section 2.3. UGAL requires a congestion metric. We evaluate five different congestion metrics for EBNs to be used instead of credit count in VCNs. We also find progressive adaptive routing to improve load balancing.

The first congestion metric we consider is a running average of the amount of cycles outputs remain blocked for (output block time). Once an output is blocked, a counter keeps track of the amount of clock cycles until it gets unblocked. At that point it updates the running average. The new value for running averages is calculated as $newvalue = 0.3 \times oldvalue + 0.7 \times newsample$. These factors were chosen for best performance after evaluating the alternatives.

The second metric is a ratio of the amount of cycles an output has spent blocked, divided by its unblocked cycles (blocked/unblocked). We calculate this for the 20 most recent clock cycles.

The third congestion metric is output channel occupancy (occupancy route). After all packets arriving in input ports are routed, the occupancy counters for the outputs they chose are incremented by the packet's length in flits. Output counters are decremented by one for each flit leaving the region of interest. A region of interest is the output channel segment for which we keep track of flits in transit.

Detecting flits leaving the region of interest is done by an AND gate whose inputs are the ready and valid signals between the last EB of the region of interest and the next. The circuit is shown in Figure 5.

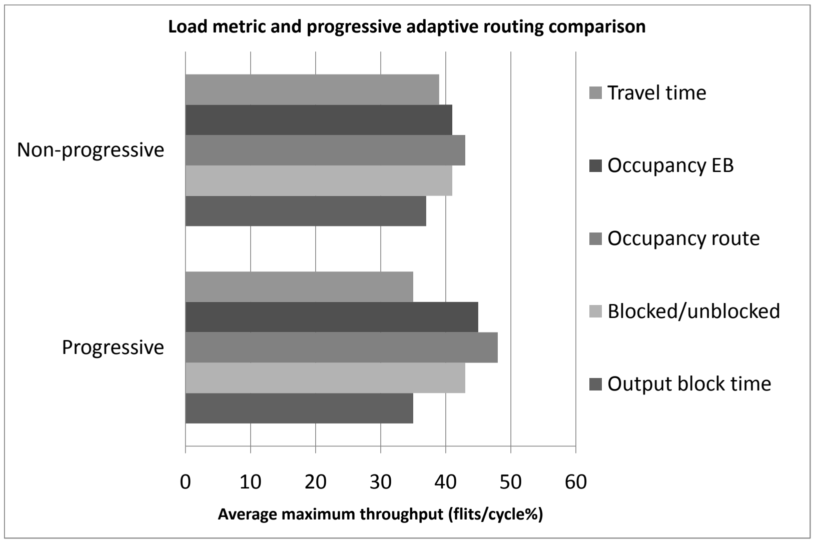


Figure 6: Evaluation of EBN congestion metrics and progressive adaptive routing. Results are for a single UGAL FBFly, with one non-minimal and one minimal sub-network. Simulation setup is explained in section 3.1.

Since that output has to be routed back to the router, there is a propagation delay which makes increasing the length of that region excessively unfavorable. That length represents a trade-off by dictating how fast routers can sense congestion at the next hop router. As discussed in section 2.1, that wire can be optimized for delay to accommodate larger regions. For fairness, the region of interest in our study is as long as the shortest network channel. Propagation delay is equal to the channel's propagation delay.

The fourth congestion metric is channel occupancy which instead increments each output's occupancy counter for each flit entering output EBs (occupancy EB).

Finally, the firth congestion metric uses the same mechanism as above but measures the average number of cycles needed by flits to leave the region of interest (travel time). A FIFO at each output stores timestamps of flits entering output EBs. Once a flit has been detected to leave the region, the timestamp at the head is removed and the current clock cycle count is used to determine the delay and update the running average.

Figure 6 shows comparison results. The metric of choice is channel occupancy incrementing counters after routing decisions. Occupancy metrics are also the simplest. Metrics based on running averages fail to adapt quickly to the current network status. Moreover, it is important for a metric to record the amount of flits that have been routed and waiting for an output as well as flits that have advanced to it. Otherwise, inputs are not aware of other inputs' choices. Thus, an output channel which is never blocked appears as not loaded even though many flits may be blocked waiting for that output because all inputs are making the same choice.

After flits get injected into the non-minimal network, UGAL is applied. UGAL chooses a random intermediate destination. It calculates the dimension-ordered routing (DOR) path to it as well as to the packet's final destination. It decides to route to the intermediate destination (non-minimally to the final destination) if the load of the output port to the appropriate next hop, multiplied by the overall hop count to the final destination, is larger than the same metrics for routing directly (minimally) to the packet's final destination. That decision is never revisited.

Flits are routed minimally, in an Y'X' manner in the non-minimal network, and XY in the minimal. Flits deciding to route minimally to their final destination are routed X'Y. The X' hop places them into the minimal network. If there is no X' hop and a non-minimal route has been decided, the X hop becomes an X' hop since flits would have reached their intermediate destination in the non-minimal network. Therefore, routing becomes Y'X'Y. The self-channel is used when a traversal to the minimal network is required, but there is no X' hop. Thus, an extra hop is created. Routing Y'X'YX may result in a more balanced load between rows. However, this routing causes an increased performance only for one of our traffic patterns – transpose –, while the rest have an equal performance to Y'X'XY, or smaller due to a more severe load imbalance in the minimal network.

Although UGAL attempts to balance the load, routers do not have congestion information for channels not attached to them. Therefore, there might be congestion at channels further away. To solve this, we apply progressive adaptive routing to allow the random intermediate destination decision to be revisited. For every non-minimal hop towards the current intermediate destination, another random intermediate destination is chosen that would result in an output in the same axis as the output towards the current intermediate destination, to preserve dimension ordering. UGAL is applied to choose the best option among the previous intermediate destination and the new. The best option then becomes the new intermediate destination and the output port towards it is used instead. If the intermediate destination is not reached with that hop, routing proceeds in the next dimension. Thus, there is still only up to one hop in every dimension. As soon as the most recently chosen intermediate destination is reached, routing proceeds to the minimal network. In addition, when taking an X' channel to the minimal network, another X' channel is randomly chosen and UGAL is applied to keep the best option. Routing in the minimal network is deterministic.

Progressive adaptive routing is not applied in the minimal network since the final destination is constant. An evaluation of progressive adaptive routing is included in Figure 6. Disabling it drops the maximum throughput by 2.5 % by average among all the congestion metrics, and by 10.4 % for the best one. That is due to an increased load imbalance and congestion in hops after the initial UGAL decision.

Because non-minimal routers have no X' channels to other non-minimal routers, they have a smaller radix. Therefore, they have shorter critical paths and more cycle time to make the complex adaptive routing decisions. Having separate networks makes it hard to accurately balance load across sub-networks under all traffic patterns. Traffic from sources and destinations connected to same-coordinate routers now need to change to the minimal network. Thus, they use network channels to make the traversal. This makes them contend with other network traffic, which is not true without sub-networks. This issue can be solved by increasing network cost by adding more channels, such as ejection ports in the non-minimal network.

2.5 Deadlocks

EBNs use duplicate physical channels, discussed in section 2.6, in the same manner as VCs to define disjoint traffic classes and prevent deadlocks. Moreover, a unique type of deadlock, the interleaving deadlock, appears in EBNs.

Protocol (request-reply) deadlocks [8] are solved by guaranteeing that replies are always able to reach network destinations, bypassing blocked requests to the same destinations. Destinations may be waiting for those replies before they can serve more requests. More complex protocols may require more traffic classes. Cyclic dependencies [4] are formed by a series of packets each depending on one another in a cycle in order to progress. They are broken by enforcing DOR in topologies without physical channel cycles, such as our chosen 2D mesh and 2D FBFly. If that is not the case, enough disjoint traffic classes must be formed such that no cyclic dependency is possible within and across classes.

EBNs suffer from a unique type of deadlock, the interleaving deadlock. Due to the FIFO nature of EB channels, flits at the head must be sent to their outputs before the next flit coming to the same input can be processed. If packets are allowed to interleave, a blocked packet also blocks tails of other packets that have interleaved with it. Therefore, a cyclic dependency can be formed between the interleaved packets. For instance, since router input ports have a finite number of routing computation latches to store routing decisions made for head flits, a new head flit may arrive at an input port with all its routing computation latches occupied. Thus, that head flit depends on tail flits from the packets occupying those latches and interleaving with the new packet, to have routing computation latches released. However, those tail flits also depend on that head flit in order to advance.

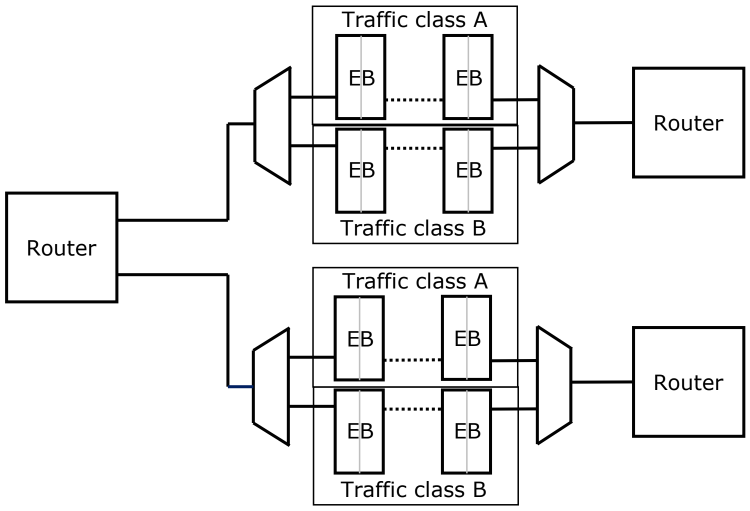


Figure 7: Duplicating channels and multiplexing them into a single router port.

This deadlock can be easily prevented by disabling packet interleaving. This guarantees that a head flit is followed by all the remaining flits of the same packet before another packet's flits arrive. Therefore, only one routing computation status latch is required for each input port. 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 to have been delivered once their tails arrive, this may decrease packet latency. Interleaving may result in tails arriving late, behind body flits from a number of other packets. This is analogous to VCNs which do not allow interleaving within the same VC.

2.6 Duplicating Physical Channels

Duplicating physical channels is the means for EBNs to differentiate between traffic classes and also provide a larger amount of bandwidth for increased performance. We evaluate three ways of duplicating channels and conclude that using duplicate sub-networks is the preferred choice.

The first option is duplicating physical channels between routers but multiplexing them into router ports, as shown in Figure 7. To maintain non-interleaving, multiplexers and de-multiplexers must select on a per-packet basis. Moreover, routers need to have duplicate output EBs to prevent flits of different classes from interacting. We find this option to yield a minimal performance gain since router ports remain the same, but there is a disproportional increased overhead due to having a multiplied channel clock and output EB area and power cost.

The second option is duplicating physical channels and router ports. However, we find this to yield an excessive overhead due to the crossbar cost increasing quadratically with the number of router ports. For the same reason, dividing into sub-networks –the third option– proves to be an efficient choice, as already shown for VCNs [1]. It doubles the available bisection bandwidth and therefore cost. However, when narrowing channels down to meet the same power or performance (maximum throughput) budget as the single network, router cost decreases quadratically. This results in a more performance and power efficient overall network, enabling us to maintain a higher bisection bandwidth and performance with the same power budget.

3 Evaluation

This section presents evaluation results for EBNs. The methodology is explained in section 3.1. Results are presented in section 3.2.

3.1 Methodology

Evaluation is performed with a modified version of booksim [6] for EBNs. We compare EBNs and VCNs using a 2D mesh with DOR, and the UGAL FBFly described in section 2.3. For a fair comparison, EBNs and VCNs have two sub-networks to support a request-reply protocol. 2D mesh networks have 16 routers in each sub-network, organized as 4×4, without a concentration factor. Injection and ejection channels have a latency of 1 clock cycle. Network channels have a latency of 2. For the UGAL FBFly, short channels also have a latency of 2. FBFly medium-length channels have a latency of 4 and the long channels a latency of 6 clock cycles. Self-channels have a latency of 1 clock cycle. Reply and request sub-networks have 32 routers each, 16 for the non-minimal sub-network and 16 for the minimal. They are also organized in a 4×4 fashion. The concentration factor is set to 4. Non-minimal routers have 7 input and output ports whereas minimal have 10. Progressive adaptive routing, as described in section 2.4, is applied to both FBFly VCNs and EBNs.

Packets are assumed to be a constant size of 512 bits. Sources generate packets according to their injection rate and enqueue them in the proper sub-network's network interface buffer according to the packet's type. One flit can be injected to each sub-network from the appropriate network interface buffer each cycle. The cost of having higher radix network interfaces to connect to more sub-networks at sources and destinations is not included.

The set of traffic patterns [6] used for the evaluation is uniform random, random permutations, shuffle, bit comparison, tornado and neighbor traffic for the 2D mesh, extended to include transpose and an adversarial traffic pattern for the UGAL FBFly. Presented results average over that set of traffic patterns for each sampling point. The maximum throughput is the average of the maximum throughput of each traffic pattern. The consumed power at that average maximum throughput is the average of the power consumptions at the maximum throughput of each traffic pattern. Since only the UGAL FBFly has a concentration factor, the adversarial traffic pattern aims to load specific network channels by making all sources in a router send to destinations in a single other router. Moreover, transpose illustrates the effect of traffic destined to the same-coordinate router now contenting with other network traffic to switch to the minimal network, as explained in section 2.4.

Area and power models for routers and channels are based on those described in [1]. We model channel wires as being routed above other logic and report only the area of the repeaters and flip-flops. Router area is estimated using detailed floorplans, and input buffers are implemented as SRAM. Device and interconnect parameters are for a 65 nm technology. We use a clock frequency of 2 GHz, about 20 FO4 inverter delays. Critical devices in the channels and router datapaths, such as the repeaters used to drive large wire capacitances, are sized to ensure circuits will operate at the clock frequency. The power model includes all of the major devices in the channels and routers, and includes leakage currents. The flip-flops in the channels are clock-gated locally. Aggressive low-swing channel designs can achieve up to a 10x traversal power per bit reduction compared to full-swing [9]. As a conservative estimation, our low-swing channel model has 30 % of the full-swing repeated wire traversal power, and double the channel area.

VCNs use a two-stage router design. The first stage consists of routing, VC and switch allocation. The second stage is switch traversal. We decide the number of VCs and buffer slots to maximize VCN performance and power efficiency. For this, we sweep the number of VCs and buffer slots and compare the ratio of the maximum throughput and total power consumption for a baseline design of 64-bit channels. We find that for the UGAL FBFly, a total of 8 VCs (4 for each traffic class) with 8 buffer slots each –enough to fully fit a packet– are the optimal choice. Increasing any of the two parameters has a minimal benefit of 1-2 % to the maximum throughput, but a power consumption increase of 8-11 %. For the 2D mesh, the optimal choice is 6 VCs with 8 buffer slots each.

3.2 Results

Figures 8 – 15 show Pareto optimal curves for the UGAL FBFly and the 2D mesh for full and low-swing channel models. The curves were generated by sweeping channel width from 28 to 192 bits in multiples of the packet size. Each point represents an optimal design point, associating power consumption and maximum throughput, or occupied area and maximum throughput. Thus, they illustrate the characteristics of a network with a certain area, power or performance budget.

Figures 16(a) and 16(b) present throughput-latency curves for the two topologies. Uniform traffic was assumed since averaging each sample point produced a counter-intuitive graph near each traffic pattern's saturation point. Channel width is equal across networks.

Figure 17(a) shows a breakdown of the consumed power into network components for the full-swing model. Figure 18 presents the same breakdown for the low-swing model. Finally, Figure 17(b) presents an area breakdown for the full-swing model. All results are for a single DOR FBFly network, making EBN and VCN flits traverse the same paths. This provides a fair comparison given the different congestion metrics for UGAL routing of the two flow-control schemes. Low-swing networks have the same area breakdown except for the channel area which is doubled. Channel width is equal across networks. There are four major network components: channels, input buffers (VCBs), crossbars and output modules. For each, power for clocking or control wires is added where applicable, as well as dynamic FF traversal power. Channel traversal refers to the power to traverse a segment with repeaters.

Table 1 summarizes the percentage gains for each of the two topologies. One aspect, shown in the first column, is normalized for each comparison. The percentage gains compare VCNs and EBNs against the two other aspects. That percentage is calculated by calculating the distance at each sampling point between the Pareto curve that associates each aspect under comparison with the normalized aspect, dividing by the value of the aspect under comparison, and averaging among all sampling points. A positive percentage means that EBNs provide gains for that comparison, and thus the percentage was calculated against the aspect under comparison value of VCNs. Likewise, negative percentages indicate gains for VCNs.

4 Discussion

As the results show, EBNs yield larger gains in a 2D mesh. In a mesh, each flit takes many router hops. At every hop, it spends energy to be buffered. Thus, each flit consumes more energy to reach its final

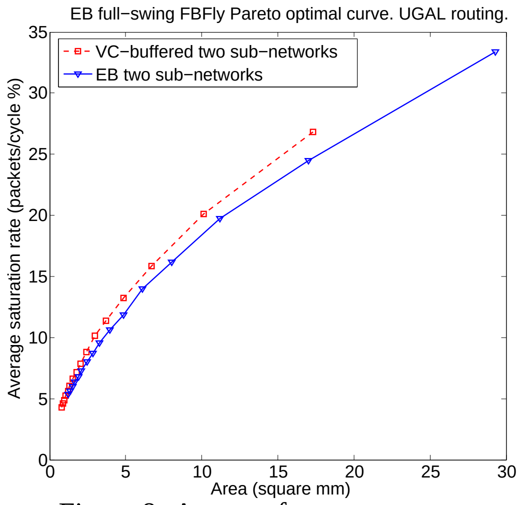


Figure 8: Area-performance curve.

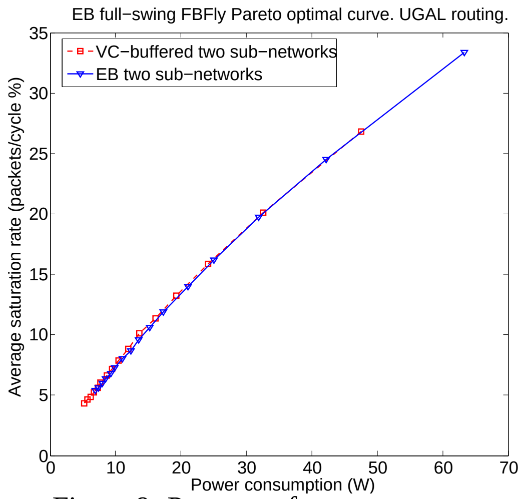


Figure 9: Power-performance curve.

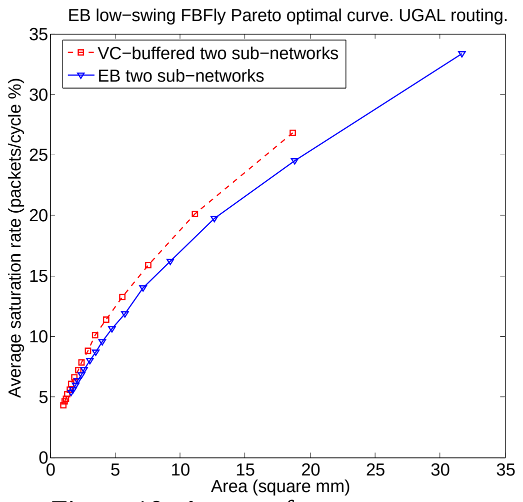


Figure 10: Area-performance curve.

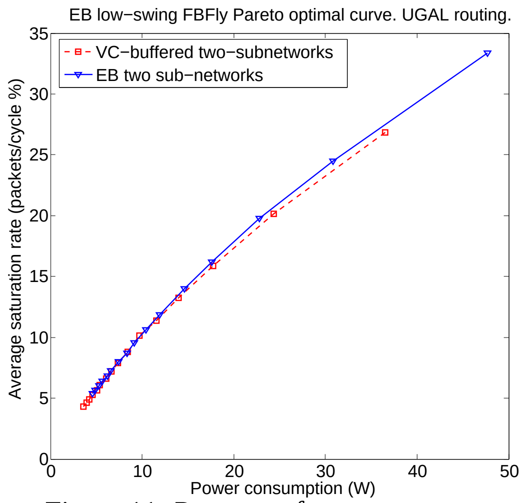


Figure 11: Power-performance curve.

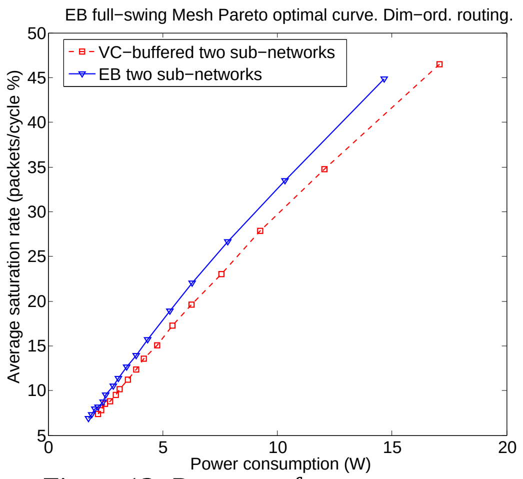


Figure 12: Power-performance curve.

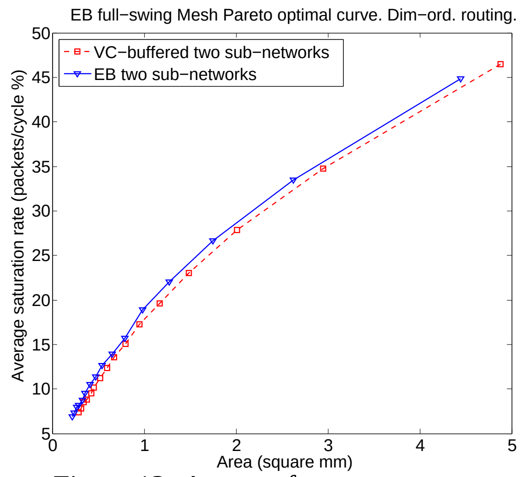


Figure 13: Area-performance curve.

destination compared to a FBFly, with VCB energy being a more significant portion. Therefore, 2D mesh EBNs provide larger power savings when removing VCBs. The added power for the intermediate EBN

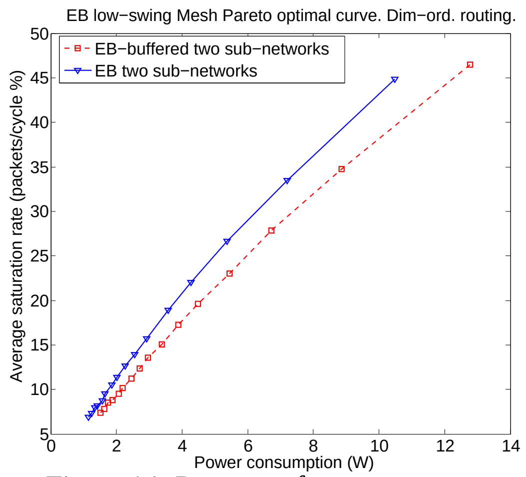


Figure 14: Power-performance curve.

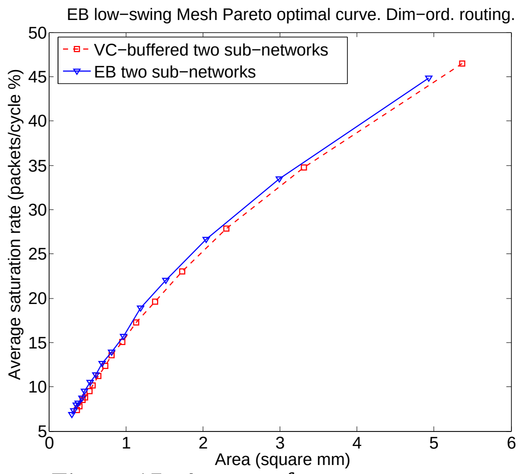
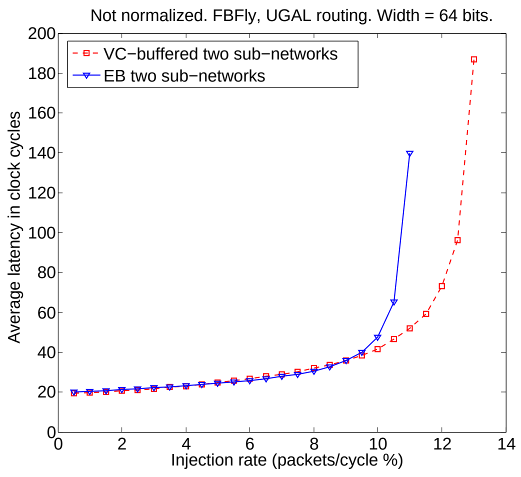
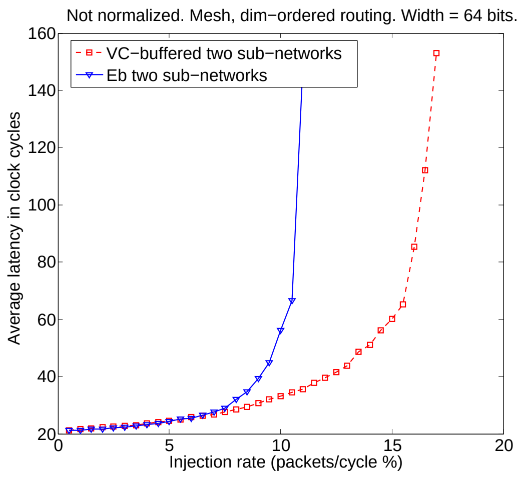


Figure 15: Area-performance curve.



(a) FBFly with UGAL routing.

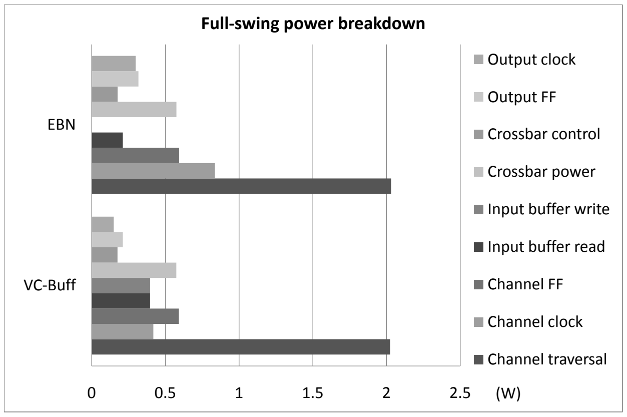


(b) 2D mesh with DOR.

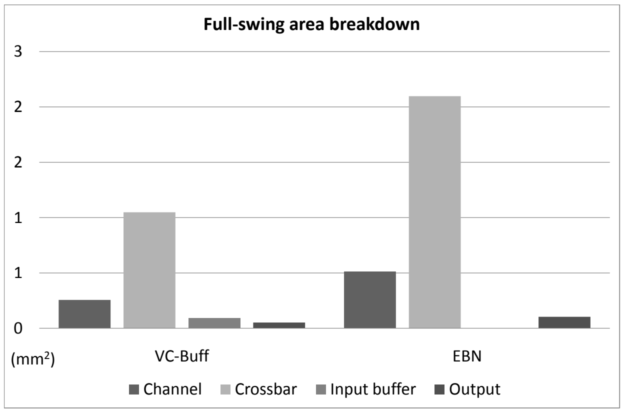
Figure 16: Throughput-latency curves for uniform random traffic.

router pipeline register is much less. This shows that topology hop count is a factor which affects EBN gains.

The dominant portion of the overall power for the full-swing model is the power to traverse a repeated wire, which makes VCB power less significant. Low-swing channels reduce the overall power and make VCB power a more significant portion of it. Thus, EBNs yield a larger power gain by removing VCBs in low-swing channel networks. Our design assumes efficient VCBs implemented with SRAM cells. Implementations with FF or latch arrays are less efficient, and therefore also increase VCB overhead. Doubling channel area due to differential wiring has a small impact because the vast majority of the area is occupied



(a) Power breakdown (W) at a 4% packet injection rate. For EBNs, input buffer read power is the traversal power for the intermediate router register presented in Figure 2(a).



(b) Area breakdown (mm^2).

Figure 17: Power and area breakdowns for a DOR FBFly with full-swing channels under uniform traffic.

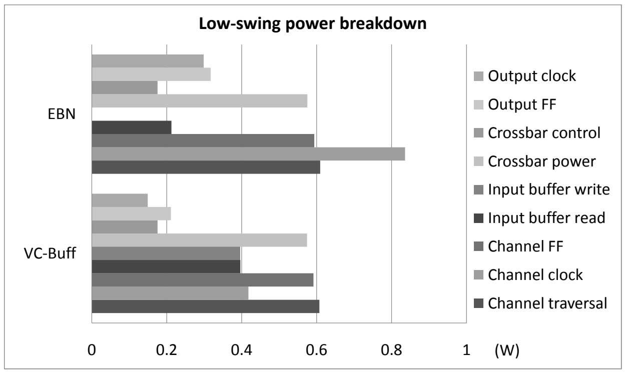


Figure 18: Power breakdown (W) at a 4% packet injection rate for a DOR FBFly with low-swing channels under uniform traffic. For EBNs, input buffer read power is the traversal power for the intermediate router register presented in Figure 2(a).

by the router crossbar.

EBNs power gains constitute a power budget to be spent on widening channels. By choosing the proper channel width, power between EBNs and VCBs is normalized. In that case, EBNs can support a higher maximum throughput, as shown in the Pareto curves. If normalizing for performance, another channel width is chosen for EBNs, making them consume less power. SRAM VCBs occupy a small amount of area. Thus, area normalization makes channel widths equal or almost equal, resulting in similar networks as bisection bandwidth normalization.

Due to the FIFO nature of EB channels, packets flowing in the same sub-network share the same resources. Therefore, blocked flits block all flits behind them. Thus, EBNs are more sensitive to contention

Table 1: EBN gain percentages according to a normalized aspect, shown in the first column, against a comparison aspect. Negative percentages indicate gains for VCN. Gains are average distance between the corresponding Pareto optimal curves.

Full-swing						
Normalization.	2D Mesh, DOR			FBFly, UGAL routing		
Comparison in:	Area	Performance	Power	Area	Performance	Power
Area	-	3%	8%	-	-8%	9%
Performance	7%	-	12%	-13%	-	0%
Power	-12%	10%	-	-12%	-1%	-
Low-swing						
Area	-	3%	12%	-	-10%	15%
Performance	6%	-	17%	-15%	-	4%
Power	-19%	14%	-	-19%	3%	-

since they cannot allow flits destined to other output ports to advance, bypassing blocked flits. As a solution, more duplicate sub-networks can be provided, unless that would decrease overall efficiency. This also makes providing quality of service (QoS) guarantees only possible across duplicate sub-networks.

Due to this, the effect of having VCBs in networks with equal-width channels to normalize for area is a higher maximum throughput with a higher maximum power consumption, but with a favorable performance/power ratio compared to EBNs. Designs with fixed area budgets and the need for the highest performance possible with that budget still have VCNs as their best choice.

As explained in section 2.6, dividing into sub-networks is more performance and power efficient. However, this is only true up to a certain number of sub-networks. After that, each sub-network's control overhead starts dominating and serialization latency becomes significant. Moreover, duplicating networks increases source and destination network interface radix. Thus, for each network configuration there is a unique number of sub-networks which optimizes performance and power efficiency. VCNs are a favorable choice for complex protocols or adaptive routing algorithms requiring more sub-networks than that number.

The amount of buffering in EB channels scales directly with channel length, affecting EBN performance. Topologies with double the channel length have an increased performance of 8-10% in the UGAL FBFly, but on the other hand almost double the channel power for a total power increase of approximately 60%. Topologies with half the channel length show a similar trend. VCN performance is affected by the different size of VCBs, due to the different round-trip time. The 2D mesh shows a similar trend. This shows that the dominant factor affecting maximum throughput in EBNs is the contention in the bufferless routers.

However, designers might still find it beneficial to add storage slots in channels, as explained in section 2.1, depending on their topology, layout and router architecture.

EB channels have the same zero-load traversal latency, since the ready-valid protocol described in section 2.1 does not impose any additional latency. Moreover, both of the router schemes described in section 2.2 consist of two stages, as the assumed VCN router model. In practice, EBNs routers are more likely to be fewer cycles to traverse due to their simplified design. Based on this assumption, zero-load latency is equal between VCNs and EBNs of equal width channels. As injection rate increases, latency increases in a similar manner between the two flow-control schemes. However, EBNs saturate earlier for equal-width channels due to their sensitivity to contention. Modifying channel width affects serialization latency, thus zero-load latency.

We have shown that designs benefit from EBNs by receiving a higher throughput for the same power budget, or consuming less power for the same throughput. However, designs with strict area budgets which prioritize performance find VCNs to be the preferable choice. Designs which need other features from the network should investigate how to implement those in EBNs and compare according to their priorities. Low-swing channels or any other efficient circuit implementations which reduce the power consumption of various components make VCB cost more significant compared to the overall, and thus provide a greater gain margin for EBNs. For fairness, this study examines SRAM VCBs, which is the most efficient design. Assuming latch or FF arrays would significantly increase VCB overhead, thus EBN gains.

5 Related Work

EBs channels first appeared as unpipelined, using repeater cells for storage [14]. Past work has already examined EBs in NoCs [12], proposing a hybrid VCB – EB scheme. However, due to the FIFO nature of channels, the interleaving deadlock, presented in section 2.5, can still occur in these networks. Preventing it requires transmitting flits without a credit (which thus may be stored in the EB channel) only if they are non-head, preventing the network from achieving a high performance.

For the same reason, there is no real isolation between flits of different VCs. Blocked flits in EB channels block flits from other VCs, even though there may be available buffer space in the router. We have found such hybrid schemes to be beneficial only for very small VCBs. However, such small VCBs represent an inefficient design choice since they can be made larger for a small power cost, but with a significant performance gain. This assumes SRAM VCB implementations, which are the most efficient implementation choice. Finally, using repeater cells for storage adds transistors to them, affecting the channel datapath. Such designs rely on parasitic capacitances which makes them challenging during the validation process.

Compressionless routing also does not use VCs [10]. It relies on feedback from the network sent back to network interfaces. Time division circuit-switching flow-control has also been proposed [13]. Connection-based routing can also be used for deadlock-free adaptive routing [19] by allowing intermediate nodes to tear down and modify end-to-end virtual circuits. A hybrid packet-connected circuit has also been proposed, requiring the routers to have buffer space only for request packets [20]. Bufferless networks can avoid dropping packets by emitting packets in a non-ideal direction, also called deflective routing [16].

While there has been a number of flow-control schemes, the vast majority of network-on-chip (NoC) implementations focus on packet switched networks with a per input VCB scheme [3]. Such networks enable easy deadlock avoidance, optimized wire utilization, improved performance and QoS [2]. These characteristics along with a reasonable design complexity make VCs the current dominant NoC flow-control scheme and VCNs the comparison metric for this work.

6 Conclusion

This work presented EBNs, an efficient NoC flow control scheme. EBNs make use of already-existent pipeline FFs in channels for storage, thus using channels as distributed FIFOs. With the removal of VCBs, EBNs gain a power cost margin spent on widening channels, essentially trading VCBs for increased bandwidth. Duplicate physical channels are used instead of VCs to prevent cyclic dependency and protocol deadlocks. Dividing into sub-networks increases performance and power efficiency when normalizing for a fixed performance or power budget. This way, adaptive routing can be applied in EBNs. As an application of this, we presented a scheme to apply UGAL in EBNs.

For a FBFly with UGAL routing, performance and power efficiency is almost equal in the full-swing model, with a 4 % gain for EBNs in the low-swing case, compared to UGAL VCNs. Gains in 2D mesh networks reach 12 % for the full-swing model, and 17 % for the low-swing. On the other hand, due to having wider channels, EBNs occupy more area when normalized for power or performance. Zero-load latency is equal between EBNs and VCNs.

An unexplored dimension of this study, left as future work, is the effect of the simpler router design due to the removal of the VC allocator. Depending on router design, clocking demands and device technology, its critical path may be shortened allowing the network to be clocked at a higher frequency or reducing the clock cycle delay per router. This provides higher throughput per absolute time and a smaller zero-load latency. This constitutes a subtle EBN gain. Additional future work includes an improved FBFly UGAL scheme to balance load more evenly and thus improve efficiency.

Overall, the flow-control scheme choice must be made by system designers with their priorities, requirements and constraints in mind. Many designs constrained in power or performance will find EBNs the means to increase network efficiency.

References

[1] James Balfour and William J. Dally. Design tradeoffs for tiled cmp on-chip networks. In *ICS '06: Proceedings of the 20th annual international conference on Supercomputing*, pages 187–198, 2006.

[2] Tobias Bjerregaard and Shankar Mahadevan. A survey of research and practices of network-on-chip. *ACM Comput. Surv.*, 38(1):1, 2006.

[3] W. J. Dally. Virtual-channel flow control. *IEEE Trans. Parallel Distrib. Syst.*, 3(2):194–205, 1992.

[4] W. J. Dally and H. Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *IEEE Trans. Parallel Distrib. Syst.*, 4(4):466–475, 1993.

[5] Wiliam J. Dally and Brian Towles. Route packets, not wires: On-chip interconnection networks. In *Proceedings of the 38th Design Automation Conference (DAC)*, pages 684–689, June 2001.

[6] William Dally and Brian Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[7] G. de Micheli and L. Benini. Networks on chip: A new paradigm for systems on chip design. In *DATE '02: Proceedings of the conference on Design, automation and test in Europe*, page 418, 2002.

[8] Andreas Hansson, Kees Goossens, and Andrei Rădulescu. Avoiding message-dependent deadlock in network-based systems on chip. *VLSI Design*, 2007:Article ID 95859, 10 pages, May 2007. Hindawi Publishing Corporation.

[9] R. Ho, K. Mai, and M. Horowitz. Efficient on-chip global interconnects. In *VLSI Circuits, 2003. Digest of Technical Papers. 2003 Symposium on*, pages 271–274, June 2003.

[10] J. H. Kim, Z. Liu, and A. A. Chien. Compressionless routing: a framework for adaptive and fault-tolerant routing. *SIGARCH Comput. Archit. News*, 22(2):289–300, 1994.

[11] John Kim, William J. Dally, and Dennis Abts. Flattened butterfly: a cost-efficient topology for high-radix networks. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 126–137, 2007.

[12] Avinash Kodi, Ashwini Sarathy, and Ahmed Louri. Design of adaptive communication channel buffers for low-power area-efficient network-on-chip architecture. In *ANCS '07: Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems*, pages 47–56, 2007.

[13] Jian Liu, Li-Rong Zheng, and H. Tenhunen. A guaranteed-throughput switch for network-on-chip. In *System-on-Chip, 2003. Proceedings. International Symposium on*, pages 31–34, Nov. 2003.

[14] M. Mizuno, , W. J. Dally, and H. Onishi. Elastic interconnects: repeater-inserted long wiring capable of compressing and decompressing data. In *2001 IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, pages 346–347, 464, 2001.

[15] Robert Mullins, Andrew West, and Simon Moore. Low-latency virtual-channel routers for on-chip networks. *SIGARCH Comput. Archit. News*, 32(2):188, 2004.

[16] Erland Nilsson, Mikael Millberg, Johnny Oberg, and Axel Jantsch. Load distribution with the proximity congestion awareness in a network on chip. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, page 11126, 2003.

[17] Arjun Singh. *Load-Balanced Routing in Interconnection Networks*. PhD in electrical engineering, Stanford University, 2005.

[18] V. Stojanovic and V.G. Oklobdzija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. *Solid-State Circuits, IEEE Journal of*, 34(4):536–548, Apr 1999.

[19] Yoshio Turner and Yuval Tamir. Deadlock-free connection-based adaptive routing with dynamic virtual circuits. *J. Parallel Distrib. Comput.*, 67(1):13–32, 2007.

[20] Daniel Wiklund and Dake 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*, page 78.1, 2003.