

# NoC with Near-Ideal Express Virtual Channels Using Global-Line Communication

Tushar Krishna<sup>1</sup>, Amit Kumar<sup>1</sup>, Patrick Chiang<sup>2</sup>, Mattan Erez<sup>3</sup>, Li-Shiuan Peh<sup>1</sup>

<sup>1</sup>Dept. of Electrical Engineering, Princeton University, Princeton, NJ 08544

<sup>2</sup>School of EECS, Oregon State University, Corvallis, OR 97331

<sup>3</sup>Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712

<sup>1</sup>{tkrishna, amitk, peh}@princeton.edu, <sup>2</sup>{pchiang}@eecs.oregonstate.edu,

<sup>3</sup>{mattan.erez}@mail.utexas.edu

## Abstract

*As processor core counts increase, networks-on-chip (NoCs) are becoming an increasingly popular interconnection fabric due to their ability to supply high bandwidth. However, NoCs need to deliver this high bandwidth at low latencies, while keeping within a tight power envelope. In this paper, we present a novel NoC with hybrid interconnect that leverages multiple types of interconnects—specifically, conventional full-swing short-range wires for the datapath, in conjunction with low-swing, multi-drop wires with long-range, ultra-low-latency communication for the flow control signals. We show how this proposed system can be used to overcome key limitations of express virtual channels (EVC), a recently proposed flow control technique that allows packets to bypass intermediate routers to simultaneously improve energy-delay-throughput. Our preliminary results show up to a 8.2% reduction in power and up to a 44% improvement in latency under heavy load compared to the original EVC design that only uses the conventional full-swing interconnects.*

## 1 Introduction

Future microprocessors are limited by power consumption and interconnect latency, such that achieving higher performance requires an increasing number of processor cores. As the number of cores increases, the performance of the networks on chip (NoC) that connects them together is critical. Traditional shared-bus architectures typically do not scale effectively to these large core counts. For example, multi-core processors such as the Cell Broadband Engine [12] and the AMD Radeon HT 2900 [1] have adopted a multi-hop NoC with a ring topology, while Intel has chosen a mesh network for its Teraflops processor, an 80-core research prototype [8]. Incorporating a packet-switched fab-

ric within a chip to interconnect the processor cores places strict power and area constraints on the NoC routers, yet requiring high performance in terms of sustained bandwidth and short packet delivery latencies. In this paper we advocate a novel *network on chip with hybrid interconnect* (NOCHI) design approach that utilizes multiple interconnect circuit types to improve latency while simultaneously reducing power. We present an instance of the NOCHI approach and apply the method to enhance a state-of-the-art NoC with express virtual channels [16]. Our preliminary results show a reduction of up to 8.2% in total network power, 29% reduction in network power used for buffers, and up to 44% in latency near saturation compared to the conventional implementation.

Our approach is driven by two observations. First, unlike traditional off-chip interconnection networks, VLSI circuits for NoCs offer a wider range of possible parameter optimizations, especially in terms of the underlying interconnects. Different design points offer trade-offs in terms of density, bandwidth, and latency, from full-swing short-range wires at minimum pitch for high bandwidth to interconnects with transmission-line properties for low-latency communication. Our second observation is that sophisticated flow control techniques, such as express virtual channels, can significantly improve the behavior of the NoC and approach the efficiency of an ideal, dedicated point-to-point interconnect. We explore a hybrid interconnection network, which consists of two network planes, one for carrying high-bandwidth data payloads and a second for providing timely control information to improve network efficiency. The *data plane* uses state-of-the-art on-chip routers with dense, high-bandwidth, full-swing links to provide the required network bandwidth. The control of this data plane is improved upon by introducing a separate *control plane* that is comprised of a collection of ultra low-latency, multi-drop on-chip global lines (*G-lines*), providing instantaneous global information to the routers and enabling a flow control technique that significantly reduces router power while

improving delivery latency. The design of the control plane is optimized for latency and exchange of control information via broadcast. Hence it can only support limited bandwidth and communication patterns and is not suitable for carrying data directly. In this paper, we propose a NoC architecture that uses some of the advantages of low-latency, control interconnect to improve upon conventional express virtual channels.

Express virtual channels (EVCs) are a network control optimization technique that enable some network packets to entirely bypass buffering, arbitration, and crossbar switching within a single dimension of the on-chip routers, thus approaching the latency and power characteristics of point-to-point interconnects. The design and analysis of EVCs using a conventional network was implemented only with short, dense, full-swing wires [16]. The authors of that paper demonstrate the impact of utilizing this flow control optimization but point out two deficiencies of the approach, both of which relate to the many-cycle latency required for signaling and exchanging of control information with the traditional link design, coupled with the delivery guarantees expected of the NoC. The first problem is that buffers at the end point of an EVC must be managed very conservatively to allow for traffic to bypass switching and arbitration and ensure that the destination (end point of the EVC) can accept the traffic. This leads to over-provisioning and under-utilization of buffers, which adversely impacts the power dissipation of the network. In fact, multiple researches have shown that the power associated with NoC buffers accounts for 30–40% of the total power required for the NoC [8, 25]. The second problem is that virtual channels must be partitioned between different express paths statically, and the control latency limits this number. As a result, the EVC design limits the number of nodes that an EVC can span to four or fewer, and thus the opportunity to improve performance is reduced and is not applied to packets that need to traverse a larger number of nodes in a dimension.

Using our NOCHI approach and newly proposed interconnect circuits, we enable single-cycle control communication across all nodes in a row or column of a mesh network alleviating both limitations outlined above. Using timely information reduces the demand on router buffers and significantly reduces the number of buffers needed to sustain a specific bandwidth, reducing the critical buffer leakage power. At the same time, allowing distant nodes to instantaneously claim EVCs increases the applicability of the EVC technique, enabling more bypassing of nodes, reducing the traversal latency and dynamic power required to deliver packets across large distances on the chip.

To summarize, we make three important contributions to the field of NoC design:

- We introduce the two-plane NOCHI network design approach and demonstrate its potential for improving performance and reducing power, through a sample application to a 49-core chip with EVCs.

- We design a novel multi-drop on-chip global interconnect line with collision detection capabilities that provides instantaneous cross-chip control information to facilitate flow control decisions.
- We develop a flow control mechanism that uses the NOCHI method and extends express virtual channels, increasing their applicability while decreasing the amount of required buffering, resulting in latency and power reductions simultaneously.

The rest of the paper is organized as follows: Section 2 details the designs and properties of the interconnect circuits used in our example NOCHI; Section 3 provides background on express virtual channels and describes our extended implementation with NOCHI; Section 5 presents our results and discusses their implications; Section 6 presents related work; and Section 7 concludes the paper.

## 2 Global interconnect circuits

Recently there have been a number of papers showing the possibility of communicating at the speed-of-light across several millimeters on a silicon substrate. We give an overview of these techniques in Subsection 6.2. Here we describe our efforts on circuits that enable broadcast-capable, single-cycle latency, global communication (*G-lines*). In this work, we use capacitive feed-forward circuits [7, 18] with two extensions: multi-point broadcast ability and collision detection with node quantity determination.

For our simulations, we use a real 65nm, Vdd=1V standard CMOS process with 8 metal layers. Figure 1 shows a block diagram of one column of the  $7 \times 7$  chip multi-processor we are evaluating for the sample NOCHI design point. Assuming a chip edge of 7mm in top-level thick M8 metal, resistance/square is  $0.2m\Omega/sq$ . Placed within a low-K dielectric, the lumped 1mm wire resistance is  $20\Omega$  with a total coplanar capacitance of approximately 400fF. Given these dimensions, the feedforward capacitor is sized to be 300fF, requiring a medium-size inverter buffer to drive it. ( $W_p=8\mu m$ ,  $W_n=4\mu m$ ). Note that due to the capacitive feed-forward driver, the common-mode voltage of the differential wire is set by large  $30k\Omega$  termination resistors.

Figure 2 shows the pulse response at 1mm locations from one end of the global line to the other. Notice that the feed-forward capacitance not only increases bandwidth, but provides a pre-emphasis capability that helps to compensate for high-frequency signal attenuation. The delay from core0 to core6 is shown to be 193ps, or within a single clock cycle.

Our proposed circuit design enables not only speed-of-light, multi-drop capability, but also the ability for the receiver to sense the number of transmitters utilizing the line on a per-bit granularity. We refer to this technique as *smart carrier sense multiple access*, or S-CSMA. This procedure is done by implementing a flash, analog-to-digital converter (ADC), which implements voltage amplitude sensing and

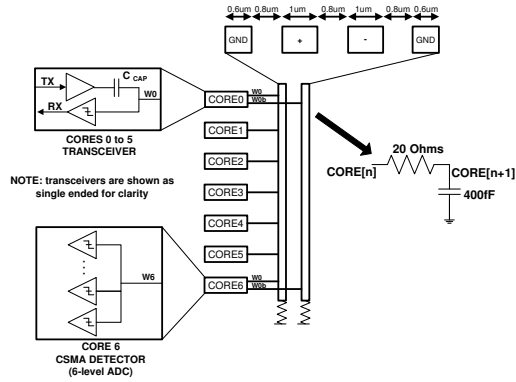


Figure 1: Block diagram and schematic of a 7-core, global interconnect

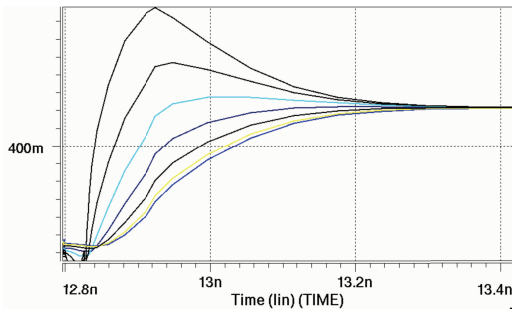


Figure 2: Step response a pulse propagation at all seven core locations

thereby determines the number of transmitters at any one instance. The worst case situation for multiple transmitters colliding with each other is the longest shared, multi-drop bus, where six cores simultaneously communicate with the seventh core, the farthest core. In this situation, six voltage levels are possible, requiring a 6-level ADC running at 2.5GHz to determine the number of simultaneous transmitting cores.

Figure 3 shows the eye diagram over 2k cycles of the six voltage levels. The minimum eye opening is approximately 79mV large, which is sufficiently large enough to overcome any quantizer offset and input sensitivity limitations. The current design utilizes receiver offset cancellation [17] to improve the minimum eye opening sensitivity to approximately 20mV.

In summary, the simulated power dissipation is: 0.6mW/transmitter; 0.4mW/receiver-quantizer (note: a single receiver uses 0.4mW, while a 6-level CSMA receiver uses 2.4mW).

### 3 Background: Express Virtual Channels

Current state-of-the-art packet-switched on-chip networks multiplex multiple packet flows on the same physical links in the network. This enables high bandwidth but results in delay, energy and area overheads due to complex

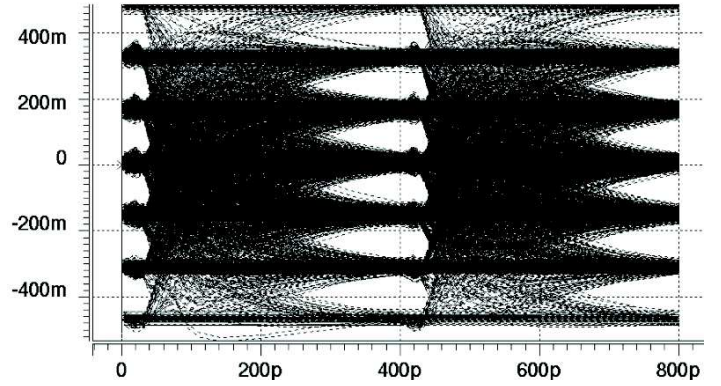
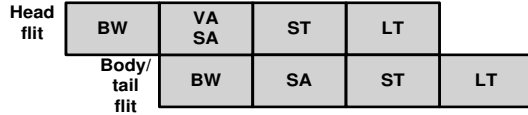


Figure 3: Six-level eye diagram for voltage determination of the number of simultaneous transmissions

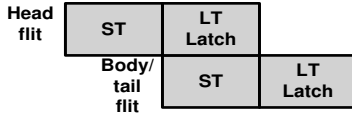
routers at every intermediate node. Each of the packets needs to compete for resources while going through the typically 4-5 stage router pipelines at each hop [4]. This dominates packet energy and delay, widening the gap between the ideal interconnect (dedicated point-to-point links between all nodes) and the state-of-the-art NoC design. Moreover, throughput is also degraded due to inefficient allocation of the network bandwidth. *Express Virtual Channels (EVCs)* were introduced as a flow-control mechanism and router micro-architecture to overcome some of these limitations [16]. The key idea behind EVCs is to provide *virtual* express lanes in the network which can be used to bypass intermediate routers by skipping the router pipeline. The EVC flits are forwarded as soon as they reach an intermediate router, without any buffering or arbitration, thereby resulting in significant reduction in packet latency, router dynamic energy (as buffer access is skipped), and router leakage energy (as the number of buffers required to sustain the same bandwidth is lowered). There is also an improvement in network throughput as contention at intermediate routers is lowered. In short, EVCs enable network performance and energy to approach that of an ideal interconnection fabric.

**EVC working:** In the EVC design, there are two kinds of VCs at each port of a router: NVCs (Normal Virtual Channels), which are the traditional VCs that carry flits through one hop at a time; and  $k$ -hop EVCs, which are the VCs that carry flits  $k$ -hops at a time. Here, we consider the more flexible and symmetric dynamic EVC design [16], in which all routers act as sources/sinks of EVCs, thereby allowing packets to acquire EVCs at any node in the network. Moreover, each router supports EVCs of varying lengths ranging from two hops up to  $l_{max}$  hops.

The head flit of a packet can choose either an NVC or an EVC of appropriate length depending on its path and the availability of VCs. When traveling on a  $k$ -hop EVC, the flit is allowed to bypass the router pipeline at the next intermediate  $k - 1$  nodes. This is done by sending a look-ahead signal once cycle in advance of the EVC flit, which sets up the intermediate switches, thus ensuring the EVC



(a) Normal speculative non-express pipeline (BW: Buffer Write, RC: Route Computation, SA: Switch Allocation, VA: VC Allocation, ST: Switch Traversal, LT: Link Traversal)



(b) Express pipeline – Flits go through this when bypassing a router using EVCs

Figure 4: Router pipelines

flit direct access to its desired output port when it arrives at these intermediate nodes. Hence, flits crossing a router using EVCs are given priority over any locally-buffered flits at that router which allows them to skip buffering and allocation and go through a much shorter express pipeline (shown in Fig. 4(b)) as opposed to the normal speculative pipeline (shown in Figure 4(a)). In other words, a head flit using a  $k$ -hop EVC, can bypass  $k - 1$  intermediate routers before getting buffered again at the sink node of that EVC. The body and tail flits follow on the same EVC and release it when the tail flit leaves the sink node.

It should be noted that in order to avoid conflicts, EVCs are not allowed to turn which ensures that multiple EVC flits arriving from different input ports and asking for the same output port do not arrive at a router simultaneously in the same cycle. However, multiple EVCs can cross a router along the same straight dimension because all such overlapping EVCs share the same physical link in a sequential fashion. From a router’s perspective, only one EVC flit can arrive at it asking for a particular output port in a given cycle in which case it is prioritized over any locally-buffered flits waiting for that output port and directly forwarded to the switch. [16] discusses in detail the router micro-architecture to incorporate EVCs.

### 3.1 Limitations of EVCs

On-chip networks have to be loss-less; packet drops are not allowed. As a result, an upstream node can send flits to a downstream node only if it knows that the flit is ensured of a free buffer slot. This information is exchanged between the routers using various techniques including credit-based signaling and on-off signaling [4]. For NVCs, this infor-

mation needs to be communicated between routers that are one-hop away, while for EVCs, this information needs to be communicated between routers that are  $k$ -hops away ( $k$  can be variable between 2 and  $l_{max}$  in dynamic EVCs). [16] uses on/off signaling to reserve downstream buffers for the EVC/NVC flits. This works as follows. Each router maintains a pool of buffers which can be allocated to a NVC or EVC flit (considering dynamic EVCs where each node can be a *source/sink*). When the number of free buffers falls below a calculated threshold  $Thr_k$ , the downstream node sends a *stop* token to its upstream node  $k$  hops away, which might be sending it flits (either physically through NVCs or virtually through EVCs). On receiving this signal, the upstream nodes stop sending any more flits. Similarly, when the number of free buffers at the downstream node exceeds the threshold  $Thr_k$ , a *start* token is sent to the upstream nodes to allow them to start sending more flits. [16] shows that the threshold value is given by

$$Thr_k = c + 2k - 1 \quad (1)$$

where  $c$  is the number of cycles taken by the token to propagate to the upstream EVC source while  $2k - 1$  is the maximum number of flits from the EVC source that might already be in flight and need a buffer downstream (for NVCs,  $k=1$ ). This is also referred to as buffer turnaround time. (The factor of 2 comes because it takes 2 cycles, for ST and LT, at the intermediate routers). In our design,  $c$  is equal to  $k$ .

There are two problems with this. The first is that these threshold values limit the maximum length  $l_{max}$  that the EVCs can take. From the threshold values, the number of free buffers  $n_{buf}$  of the downstream node should be greater than the maximum possible threshold value, that is

$$n_{buf} > Thr_k(k = l_{max}, c = k) \quad (2)$$

which gives

$$n_{buf} > 3(l_{max}) - 1 \quad (3)$$

in order to account for all flits in flight from all upstream nodes before they receive a *stop* token. Thus the minimum number of buffers required grows as we increase the length of the EVCs. The second problem is that these threshold values are highly conservative taking into account the worst case, i.e. the maximum number of flits in flight from all the nodes that would need downstream buffers. In an average scenario, this would result in under-utilization of longer EVCs due to the high threshold of free buffers required for their operation. Thus not only do we need more buffers, we might also encounter a situation where the upstream nodes are not able to send EVC flits to the downstream node, despite the latter having free buffers due to the conservative on/off signaling. Another issue with longer EVCs is that the amount of wiring overhead for reverse signals increases [16]. The original EVC design thus restricts  $l_{max}$  of the EVCs to three to four.

Another limitation of the original EVC design is that head flits at upstream nodes can only arbitrate for a fixed

number of VCs of each type (1-hop NVC, 2-hop EVC, ...  $l_{max}$  hop-EVC). Suppose that there are  $n$  nodes per dimension, and each router has  $v$  VCs per port. There is a static partition among the  $v$  VCs into  $l_{max}$  bins, one bin for each type of VC. Thus each upstream node is allowed to arbitrate for only a fixed subset of a downstream routers' VCs (per port), depending on the EVC type it wants. If an upstream node wants to send multiple packets to a  $k$ -hop away downstream node, the former will not be able to do so if the fixed number of  $k$ -hop EVCs to the latter are not free, even if the latter node might have other free VCs. Thus VC allocation will fail, and the flits have the option of either waiting for the EVCs to get free (which might take many cycles because the EVC is allocated by the head flit of a packet at this upstream node, but it is freed only when the tail flit of the packet departs from the *downstream* node); or they have an option of retrying for allocation of smaller-length EVCs (which degrades performance). This was not a major issue in the original EVC design, as there were a total of eight VCs which had to be partitioned into only two to three bins (as the maximum EVC length was mostly three). But this problem is enhanced if EVC lengths become longer, because a) the number of VCs per bin go down as  $l_{max}$  increases (which increases VC contention), and b) the VC turnaround time increases as it takes longer for flits to hop along on the EVC to the downstream node, leave from there, and then free the EVC (thus forcing other packets to choose smaller-length EVCs).

#### 4 Flow control for EVCs using global interconnects

We introduce a flow control mechanism for EVCs, where Global Interconnect Lines (G-lines) broadcast the control signals. This can help overcome the EVC length limitations, potentially reducing power and improving performance. This scheme differs from the original EVC scheme in the following ways:

- We allow EVCs of arbitrary lengths
- We allow a downstream router to signal an *ON*, even if it has only one buffer left, rather than the conservative method of sending an *OFF* signal when the buffers reach the threshold. This allows us to reduce the buffers in each router, thereby reducing power
- We allow flexible binding of EVCs, allowing a node to allocate multiple EVCs to the same downstream node if the traffic desires, rather than a fixed binding position at design time, where a node is forced to try and allocate a lower-hop EVCs if the few fixed longer EVCs are not available
- We thus potentially allow a packet to zoom through the entire route, getting buffered only at the intermediate router where it turns.

The following sections elaborate on the various issues.

#### 4.1 How to use G-lines?

The G-lines provide a one cycle broadcast across the chip. This can be exploited by sending the *start/stop* tokens across them. This potentially allows nodes to send *start* tokens upstream up to the point where they have only one empty buffer left, enabling EVCs of arbitrary lengths. For a  $n \times n$  chip, EVCs can have maximum lengths  $l_{max}$  up to  $n - 1$  (The maximum possible hops per direction). A flit can thus bypass all routers along its path in one direction, get buffered at the last router in its path that direction, turn, and zoom all the way until it reaches its destination node. (We are assuming XY routing like the original EVC design).

The G-lines can also be used to signal the availability of free EVCs at each node. This allows nodes to dynamically arbitrate for *all* EVCs at a particular router port, instead of a few statically assigned ones (which was done in the original EVC scheme). This allows an upstream node to select multiple VCs to the same downstream node, thus enabling flits to travel on longer EVCs as far as possible, thereby reducing latency.

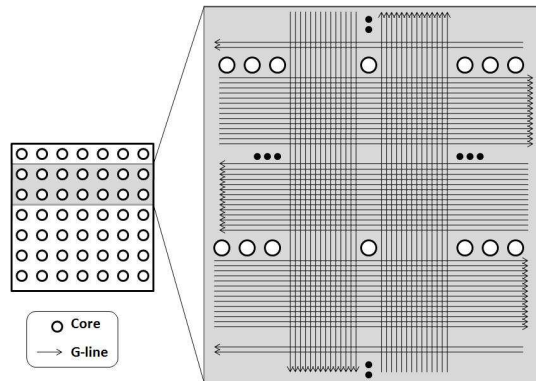


Figure 5: 7x7 chip with 14 G-lines per direction per row/column

The flow control mechanism is described next. To be consistent in comparing with the baseline EVC design, we choose a 49 core CMP with a  $7 \times 7$  packet switched mesh network. We assume dynamic EVCs, where every router can either be bypassed or can buffer the flits. Along a particular direction, for a particular node, only its upstream nodes can send flits to it. We assign 14 one-bit G-lines per direction (i.e. N-S, S-N, E-W, and W-E) as shown in Figure 5. We allow an upper limit of 6-hop EVCs ( $l_{max}$ ), allowing flits to potentially bypass all routers in a direction (unlike the original EVC design). This is shown in Figure 6.

The G-lines are divided into two sets: one for VC signaling and one for buffer signaling. Each G-line is statically assigned to one node. The requirements are: (1) a downstream node is only allowed to transmit on this line if it wants to indicate free buffers/VCs to its upstream nodes,

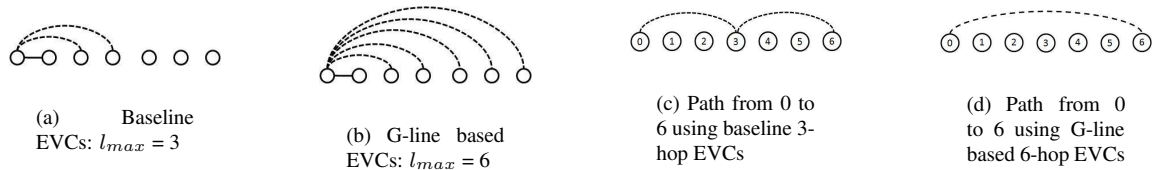


Figure 6: Comparison of baseline and G-line based EVC designs. In the baseline EVC design, a flit wanting to go from 0 to 6 takes two 3-hop EVCs in the best case, but still has to be buffered at node 3. In the G-line based EVCs, a flit can bypass all routers between 0 and 6.

and (2) any signal transmitted on this G-line by its upstream nodes is meant for this node. As there are two sets of G-lines, each node has a G-line that it uses to broadcast information about its free VCs, and another G-line that it uses to indicate its free buffers. The upstream routers can send flits to it and arbitrate for free VCs and buffers over its G-lines. Each downstream router has a special receiver that can count the *number* of signals transmitted on the G-line that cycle. This Smart-CSMA (S-CSMA) property of the G-line receivers can be used by the downstream router node to calculate how many of the upstream nodes are requesting for its VCs or buffers, and grant requests accordingly. (Note that upstream nodes reserve downstream buffers before transmitting flits because dropping of flits is not permitted).

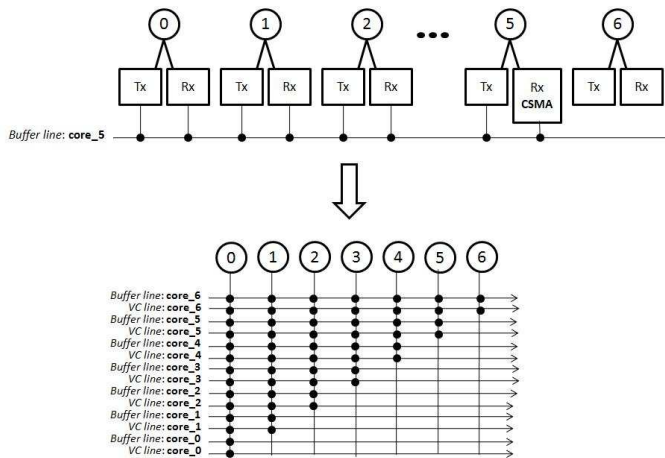


Figure 7: VC and Buffer signaling Each pair of G-lines (one for buffers and one for VCs) is statically assigned to one node. This node can only receive flits from its upstream nodes. For example, *core\_5* transmits on its buffer line if it has a free buffer, and all its upstream nodes snoop the line. Then the upstream nodes (*core\_0* to *core\_4*) can all place requests for the buffers of *core\_5* by transmitting on this line. *core\_5* receives and performs S-CSMA to calculate the number of requests

## 4.2 VC and Buffer Signaling

We now explain the buffer and VC signaling for one particular downstream router, for example *core\_5*, without loss

of generality. Figure 7 illustrates this scenario. All the upstream routers of *core\_5* in the W-E direction, namely *core\_0* to *core\_4*, are allowed to send flits to it via the 5-hop to 1-hop EVCs respectively. The downstream node and the upstream nodes are allowed to transmit on the G-line every alternate cycle. For instance, to signal for free buffers, *core\_5* transmits a 1 (*ON*) on its G-line if it has more than one free buffer. In the next cycle, the upstream nodes that want a buffer at this node for their flits *all* transmit a 1, indicating a buffer request on this G-line. The receiver of *core\_5* performs S-CSMA to calculate *how* many cores want to send flits to it, decreasing its free buffer count accordingly (implying the reservation of buffers for the flits that will arrive from the requesting upstream nodes). The node then transmits a count of the number of requests it granted to its upstream nodes, via normal wires. Each upstream node then receives the message, checking if it had issued a request to *core\_5*. If no, it forwards it upstream. If yes, it decrements this count and again forwards the flit upstream. It also signals to its switch allocator that the buffer request was granted, and that the flit can proceed.

In case the number of requests at the downstream node *core\_5* are greater than the number of free buffers that are available, the number of requests granted is equal to the number of free buffers. The free buffer count is thus made zero and *core\_5* does not transmit a 1 on its buffer G-line in the next cycle. The count of the number of requests granted is again forwarded upstream using the normal wires. As this count propagates upstream, the requesting nodes keep decrementing the value. Thus, if a node receives this count as 0, it knows that its request was not granted and places a new request once *core\_5* transmits a 1 again. *core\_5* meanwhile keeps updating its free buffer count as flits leave, transmitting a 1 once it has even one free buffer.

VC signaling works in the same manner. In the general scenario where there are enough buffers and VCs, *core\_5* transmits a 1 on its G-lines every alternate cycle while past requests by the upstream nodes get granted by the normal wires. Thus multiple requests for buffers and EVCs at *core\_5* can potentially be granted every alternate cycle.

### 4.3 TX and RX operation

It should be noted that the number of upstream nodes is different for each downstream node. For instance, the upstream nodes in the W-E direction for  $core_6$  are  $core_0$  to  $core_5$ , for  $core_5$  are  $core_0$  to  $core_4$ , for  $core_4$  are  $core_0$  to  $core_3$ , and so on. Thus the total number of G-line transmitters per direction (N-S, S-N, W-E and E-W) are  $6 + 5 + 4 + 3 + 2 + 1 = 21$  for each type of G-line (buffer and VC signaling). Therefore, the total number of G-line transmitters per direction is 42. We observed that multiple upstream nodes can request for EVCs and buffers at the same downstream node in one cycle. However, an upstream node is not allowed to send buffer/VC requests to multiple nodes in the same cycle. When the downstream nodes transmit buffer/VC availability on their respective G-lines every alternate cycle, no upstream node is allowed to transmit a request on that particular G-line. This means that in every cycle, only a maximum of 12 transmissions can occur on a G-lines per row per direction, during the situation when all nodes transmit. These transmissions may be the downstream nodes signaling buffer/VC availability or upstream nodes transmitting requests in alternate cycles. Thus the total number of active transmitters in every cycle is  $12/42 = 28.5\%$ .

Analogous to the transmitters calculation, there are 42 G-line receivers per row per direction. However, the upstream receivers snoop on *all* the G-lines every alternate cycle looking for buffer and VC availability. All of them are thus active every alternate cycle. During request signaling by upstream nodes (every alternate cycle), only the downstream routers have active receivers. Thus there are 12 active receivers in this cycle. However, these receivers are the S-CSMA receivers, which are more complex than the normal upstream receivers. The receiver for  $core_6$  has to perform a 6-count S-CSMA (as  $core_0$  to  $core_5$  can all be transmitting, such that the receiver needs to calculate the number of transmissions), while the receiver for  $core_5$  has to perform a 5-level S-CSMA, and so on. Thus each receiver is not of the same complexity. These estimates of the maximum possible transmitters and receivers active every cycle has been taken into account when calculating the G-line TX/RX power.

### 4.4 Buffer signaling optimization

An extra optimization added is that for EVCs less than or equal to length 3, in addition to the G-line based signaling, we use traditional threshold based on-off signaling as described in [4]. This is added to decrease starvation of neighboring nodes due to the long EVCs. When the number of buffers goes below the threshold, EVCs switch off like the conventional design except that G-line signaling continues until no buffer is available. This hybrid flow control scheme using G-lines in conjunction with the normal wires

ensures that this new implementation will achieve at least the baseline EVC performance.

Like the original EVC scheme, we have assumed both a free pool and a reserved pool of buffers, the latter being used for deadlock avoidance. Starvation signaling is also implemented similar to the original EVC scheme. The signaling of free reserved slots, as well as starvation, can proceed hop-by-hop using normal wires.

## 5 Results

In this section, we present a limit study evaluating the potential of NOCHI-based flow control for EVCs. The network performance was evaluated using an in-house commercial cycle-accurate simulator that models all the major components of the router pipeline at clock granularity. Table 1 presents the microarchitecture and process parameters used in this study. Router power was calculated based on extrapolations from [8] for our design point. We compare NOCHI-EVC with a baseline aggressive EVC design described in [16] using synthetic traffic. In our results, saturation throughput is chosen to be the point at which packet latency becomes three times the no-load latency.

Table 1: Process and network parameters

Technology	65 nm
$V_{dd}$	1 V
$V_{threshold}$	0.17 V
Frequency	2.5 GHz
Topology	7-ary 2-mesh
Routing	Dimension-ordered (DOR)
Number of router ports	5
Vcs per port	8
Flit size/channel width ( $c_{width}$ )	128 bits
Link length	1 mm
Wire pitch ( $W_{pitch}$ )	0.45 $\mu$ m
<b>EVC-specific parameters</b>	
EVC pipeline	aggressive express pipeline
$l_{max}$	3
NVCs per port	2
EVCs per port	6
<b>Global interconnect circuit parameters</b>	
differential pair area	5 $\mu$ m <sup>2</sup>
capacitive-feedforward inverter power (TX)	0.6mW
offset-canceled quantizer power (RX)	0.4mW
7mm-wire pulse-propagation latency	192ps

### 5.1 Assumptions in design

For our evaluations, we have made certain assumptions about the flow-control design to estimate the potential of our scheme, which are currently not supported by the actual design. For buffer signaling, we assume that the free-buffer signaling by the downstream node, and the arbitration for these buffers by the upstream nodes happens within a cycle. In case of contention for the buffers (i.e. the number of requests being greater than the number of free buffers), there is a static priority: the requesting node that is farthest from

the downstream node has the highest priority for getting a buffer, then the next farthest, and so on. Thus there is no conservative buffer management and an ON signal can be sent over the T-line even if there is just one buffer left. The situation is similar for the VC allocation. The nodes get the updated free VC information every cycle and allocate EVCs accordingly. Multiple nodes can thus get EVCs to the same node every cycle. In case of contention, the same priority scheme is followed like the buffers wherein the upstream nodes that are farthest have the highest priority.

## 5.2 Synthetic traffic

We used both uniform random traffic (in which each node sends packets to randomly chosen destinations) as well as tornado traffic (in which each node sends packets halfway around the mesh along the X-dimension) to evaluate NOCHI-EVC. The lower average hop count along each dimension in uniform random traffic led to less utilization of long express paths, resulting in almost similar performance for both NOCHI-EVC and conventional EVC.

On the other hand, for tornado traffic in which packets need to travel more hops along a dimension, the utilization of longer express paths is high leading to a significant performance gain. Fig 8 plots flit latency as a function of network load for tornado traffic for both NOCHI-EVC and EVC assuming the same amount of buffering (25 buffers per port). As shown, NOCHI-EVC is able to reduce latency by 44% near the EVC saturation point and 9.4% at no-load. This is mainly due to enabling of longer express paths in NOCHI-EVC which allow packets to bypass 53.7% of the routers along their path on average as compared to 41.3% in the baseline EVC case.

For the same saturation throughput, NOCHI-EVC requires significantly fewer buffers than EVC. Specifically, a NOCHI design with 15 buffers per port exhibits the same saturation throughput as an EVC design with 25 buffers per port. To evaluate the reduction in buffer power as a result of this, we used extrapolated data based on the Intel Teraflops NoC router [8]. Specifically, the router power of 924mW at 5GHz reported in [8] was first scaled down to 2.5GHz at 1.0V, deriving a router power of 500mW. There are 16 38-bit wide buffers per lane and 2 lanes per port in the Teraflops router, giving a total of 6080 buffer bit cells per router, with these buffers consuming 22% of the total router power. Assuming 60% of the total power being dynamic power, the power per buffer cell is 0.0108mW. With NOCHI leading to a reduction of 10 buffers per port as compared to EVC and packets bypassing 12.4% more nodes on average, this leads to a dynamic buffer power reduction of 8.3mW per router. Similarly, assuming 40% of the total power being leakage, a reduction of 10 buffers per port results in a buffer leakage power reduction of 46.3mW per router. Thus, the reduction in buffer power for NOCHI-EVC is 54.6mW per router (2.67W for the entire 49-node network) or 29.2% over the

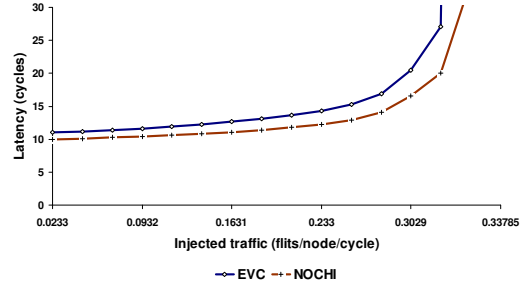


Figure 8: Network latency

conventional EVC design. However, our circuit-level simulations show the G-lines described in Section 2 consume a total of 0.67W. Hence, the net power reduction of NOCHI-EVC over EVC is 2W or 8.2% of the total network power.

## 5.3 G-Line Network Power Calculation

Our initial circuit design, which we later abandoned, used on-die  $50\Omega$  transmission lines, with  $10\mu\text{m}$  differential wire pitch ( $2.5\mu\text{m}$  width,  $2.5\mu\text{m}$  spacing). While the transmission line exhibited a near-speed-of-light 10ps/mm phase velocity, a critical problem was the series DC wire resistance. For a 7mm long wire, the series resistance was on order of the characteristic impedance, resulting in significant signal attenuation. This series resistance, coupled with the low impedance attained on-die, resulted in very large current dissipation of 5mA/transmitter. In addition, the series resistance meant that it would be difficult to distinguish the exact number of TXs that were simultaneously transmitting.

Fortunately, the proposed forward-C capacitive transceiver exhibits little power, since the global interconnect swing is reduced by a factor of 8 – 10. In addition, no static power is consumed using the G-Line circuit. With a simulated 0.6mW/transmitter, and a total 334 of TXs operating at one time, the total chip transmit power is 0.2W. The receiver power consists of the switching power to quantize a small differential input voltage. The speed and power of such quantizers scale well with CMOS process scaling, such that in our 65nm CMOS process, quantization power (including clock power) is 0.4mW/receiver. This quantizer power is the same whether for a 1-bit receive or for a 1-level decision within one of the S-CSMA analog-to-digital converters. With a total of 1176 quantizers operating on a given cycle, the total chip RX power is 0.47W. In summary, the entire power for the both the TXs and the RXs is 0.67W.



## 6 Related Work

### 6.1 Network flow control techniques that leverage advanced interconnects

Flow control, the mechanism that allocates resources to packets within the network, is a key determinant of communication energy/delay as well as network throughput. As a result, there has been a vast body of work in the past. Here, we will focus specifically on those that leverage advanced interconnect circuits. It should be noted, though, that the baseline router which we use as a comparison target in all experiments already incorporates many recently proposed techniques for improving network energy-delay, such as speculation [20, 23], bypassing [19], lookahead pipelines [6, 15], simplified virtual-channel allocation [15], and lookahead routing [5]. All these techniques drive towards ultra-low router latency, but they only succeed in bypassing the pipeline at low loads, performing poorly when network contention is high. Express virtual channels [16], however, bypasses the router pipeline at all traffic loads. G-line EVCs thus enable bypassing at all loads, lowering delay and dynamic energy at all loads, as well as leakage energy due to the reduction in buffer count.

The most relevant work is that by Kim and Stojanovic [13], who modeled the use of equalization in on-chip interconnects and investigated the impact of such equalized interconnects on on-chip network designs, through incorporating detailed models of the interconnect in a design-space exploration framework of on-chip networks. The study, however, focuses on exploring existing on-chip network topologies, while our work extends and modifies flow control (EVCs) in order to leverage transmission line characteristics. Another flow control technique that is co-designed with interconnects is flit-reservation flow control [22]. Here, no sophisticated interconnect circuits is used; this work just harnesses the faster upper metal wires for sending control flits out in advance to schedule resources for subsequent data flits, so data flits can zoom through the pipeline when they arrive. However, it needs large reservation tables and a complex router microarchitecture. More disruptive interconnect technologies have been explored in conjunction with on-chip network designs: On-chip photonics [14, 24] and RF interconnects [2] both enable very high bandwidth global communications, mandating a rethinking of on-chip network designs. In contrast, the G-line interconnect explored in this paper is a nearer-term technology that can be readily fabricated in today's VLSI technologies. Another important difference of our approach is the use of a two-plane network where the circuits are optimized separately for control and data transfer.

### 6.2 Related work with global interconnect circuits

A number of papers show the potential for communicating at the speed-of-light across several millimeters on a silicon substrate. Chang [3] and [11, 10] showed early point-to-point circuits that allowed for transmission-line, wave-like velocity for 10mm of interconnect. While transmission-line circuits achieve 10PBS/mm speed in silicon dioxide, these implementations suffer from two main disadvantages. First, area per differential pair is quite large—for example, in [11], a single differential pair with width= $8\mu\text{m}$  and spacing= $4\mu\text{m}$  would yield a  $32\mu\text{m}$  per transmission-line pitch. Second, power consumption is still very large, greater than  $2\text{mW/Gbps}$ , or  $8\times$  more power than a normal inverter repeater [21]. Even worse, due to the use of current-mode signaling, significant static power consumption exists even with little activity factor. Reduction in static current consumption can be achieved if the characteristic impedance can be increased. However, it is difficult to obtain transmission line impedances greater than  $80\Omega$  in a standard CMOS process because adjacent metal layers are in close proximity, increasing capacitance and preventing large inductance. This requirement for keeping the inductance large also prevents transmission lines from being routed over other metal or transistor layers, since these lower level layers increase the capacitance. Therefore, the ability integrate a useful number of these transmission lines on a single die becomes limited.

While the previous works above illustrate point-to-point, speed-of-light capability, recent work by Ito, et al. [9], enable both low-latency and multi-drop ability on a transmission line with low-power dissipation of  $1.2\text{mW/transceiver}$ . While this work still exhibits integration density issues (greater than  $20\mu\text{m}$  per transmission-line, with no allowable wiring metal layers below), it suggested that broadcast, multi-drop, bidirectional ability is achievable for network-on-a-chip applications. However, while this work allows for multi-drop ability, these circuits do not show a way to arbitrate or schedule such information.

Recently, it has been shown that a capacitive feed-forward method of global interconnect [7, 18] achieves nearly single-cycle delay for long RC wires, but with voltage-mode signaling. By using a simple inverter driving a feed-forward capacitance, voltage gain can be exchanged for bandwidth. For example, assuming a 1V supply voltage, adding a feed-forward capacitor 1/10 the size of the global interconnect capacitance reduces the voltage swing by 10x but also increases the bandwidth by 10x. Because these wires are inherently still capacitive, they relax the difficult constraints of requiring large inductance, resulting in higher signal density as well as enabling metal layers to be routed underneath. In addition, the use of voltage-mode driving eliminates the problem of static power dissipation associated with current-mode signaling. Our proposed G-Line cir-

cuits use this type of bandwidth extension technique, but extend this further with the concepts of multi-drop connectivity and S-CSMA collision detection and measurement techniques.

## 7 Conclusions

In this paper, we have motivated one potential use of NoCs with hybrid interconnects (NOCHI), where ultra-fast, low-swing, multi-drop interconnects are used to enable long EVCs, enabling packets to bypass most intermediate routers on their way from source to destination. Bypassing routers lowers latency (obviates the need to traverse the pipeline), and pushes throughput (as contention at intermediate routers is removed). Bypassing also saves power in two ways: (1) saving the dynamic power incurred in buffer access; and (2) reducing the number of buffers needed to sustain a specific bandwidth level, thereby reducing the buffer leakage power. Our preliminary investigations show that using this NOCHI approach with EVCs can lead to significant latency and throughput benefits, as well as power savings.

## References

- [1] AMD. ATI Radeon HD 2900 Graphics Technology. <http://ati.amd.com/products/Radeonhd2900/specs.html>.
- [2] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S. Tam. CMP network-on-chip overlaid with multi-band RF-interconnect. In *International Conference on High-Performance Computer Architecture*, February 2008.
- [3] R. Chang, N. Talwalkar, C. Yue, and S. Wong. Near speed-of-light signaling over on-chip electrical interconnects. *IEEE Journal of Solid State Circuits*, 38:834–838, May 2003.
- [4] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2004.
- [5] M. Galles. Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip. In *Proc. Hot Interconnects 4*, Aug. 1996.
- [6] P. Gratz *et al.* Implementation and evaluation of on-chip network architectures. In *Proc. Int. Conf. Computer Design*, Oct. 2006.
- [7] R. Ho, T. Ono, F. Liu, R. Hopkins, A. Chow, J. Schauer, and R. Drost. High-speed and low-energy capacitively-driven on-chip wires. *IEEE Solid-State Circuits Conference*, pages 412–413, February 2007.
- [8] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-GHz mesh interconnect for a teraflops processor. *IEEE Micro*, 27(5):51–61, Sept. 2007.
- [9] H. Ito, M. Kimura, K. Miyashita, T. Ishii, K. Okada, and K. Masu. A bidirectional- and multi-drop-transmission-line interconnect for multipoint-to-multipoint on-chip communications. *IEEE Journal of Solid-State Circuits*, pages 1020–1029, April 2008.
- [10] A. Jose, G. Patounakis, and K. L. Shepard. Pulse current-mode signalling for nearly speed-of-light intrachip communications. *IEEE Journal of Solid State Circuits*, 41:772–780, April 2006.
- [11] A. P. Jose and K. L. Shepard. Distributed loss-compensation techniques for energy-efficient low-latency on-chip communications. *IEEE Journal of Solid State Circuits*, 42:1415–1424, June 2007.
- [12] J. A. Kahle *et al.* Introduction to the Cell multiprocessor. *IBM Journal of Research and Development*, 49(4/5), 2005.
- [13] B. Kim and V. Stojanovic. Equalized interconnects for on-chip networks: Modeling and optimization framework. In *International Conference on Computer-Aided Design*, pages 552–559, November 2007.
- [14] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi. Leveraging optical technology in future bus-based chip multiprocessors. In *International Symposium on Microarchitecture*, pages 492–503, December 2006.
- [15] A. Kumar, P. Kundu, A. P. Singh, L.-S. Peh, and N. K. Jha. A 4.6Tbits/s 3.6GHz single-cycle noc router with a novel switch allocator in 65nm CMOS. In *Proc. Int. Conf. Computer Design*, Oct. 2007.
- [16] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha. Express virtual channels: Towards the ideal interconnection fabric. In *Proc. Int. Symp. Computer Architecture (and IEEE Micro Top Picks 2008)*, June 2007.
- [17] M.-J. E. Lee, W. Dally, and P. Chiang. Low-power area-efficient high-speed i/o circuit techniques. *IEEE Journal of Solid-State Circuits*, 35(11):1591–1591, November 2000.
- [18] E. Mensink, D. Schinkel, E. Klumperink, E. van Tuijl, and B. Nauta. A 0.28pf/b 2gb/s/ch transceiver in 90nm cmos for 10mm on-chip interconnects. *IEEE Solid-State Circuits Conference*, pages 412–413, February 2007.
- [19] S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb. The Alpha 21364 network architecture. *IEEE Micro*, 22(1):26–35, Jan./Feb. 2002.
- [20] R. Mullins, A. West, and S. Moore. Low-latency virtual-channel routers for on-chip networks. In *Proc. Int. Symp. Computer Architecture*, June 2004.
- [21] J. D. Owens, W. J. Dally, D. N. J. Ron Ho, S. W. Keckler, and L.-S. Peh. Research challenges for on-chip interconnection networks. *IEEE Micro*, pages 96–108, Sep./Oct. 2007.
- [22] L.-S. Peh and W. J. Dally. Flit-reservation flow control. In *Proc. Int. Symp. High Performance Computer Architecture*, pages 73–84, Jan. 2000.
- [23] L.-S. Peh and W. J. Dally. A delay model and speculative architecture for pipelined routers. In *Proc. Int. Symp. High Performance Computer Architecture*, pages 255–266, Jan. 2001.
- [24] A. Shacham, K. Bergman, and L. P. Carloni. The case for low-power photonic networks on chip. In *Design Automation Conference*, pages 132–135, June 2007.
- [25] H.-S. Wang, L.-S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proc. Int. Symp. Microarchitecture*, pages 105–116, Nov. 2003.