Reservation Cut-through Switching Allocation for High-Radix Clos Network on Chip

Yang Xu, Tang Jiang, Ming Yang and H. Jonathan Chao
Department of Electrical and Computer Engineering
Polytechnic Institute of New York University
Brooklyn, NY, USA 11201

Abstract—Clos Network on Chip (CNOC) is a promising Network on Chip (NOC) topology because of its low hop counts and good load-balancing characteristics. The throughput of a CNOC depends on the specific router design, which includes at least two major components: buffering structure and switching allocation (SA). In this paper, we propose three cut-through SA schemes for Virtual Output Queue (VOQ)-based NOC routers. The proposed cut-through SA schemes have two major advantages over the flit-mode SA scheme: (1) eliminate the Tail of Line (TOL) blocking suffered by the VOQ structure, and (2) improve the quality of matching obtained in each SA without increasing the iterations. Besides that, the proposed reservation cut-through SA scheme can cure the potential starvation problem involved in cut-through switching. The performances of the proposed cut-through SA schemes are evaluated based on a 64-node CNOC as well as a 36-node 2D Mesh network. Simulation results show that the cut-through SA schemes significantly improve the performance of CNOC and 2D Mesh network in terms of transmission latency and throughput, even in those extreme scenarios when there are long packets that cannot be switched using cut-through or the input buffer size is very small. Compared to the baseline router, the router with cut-through SA can still achieve a frequency of 1GHz using 65nm technology with only 7.5% increase in area and 11% increase in power consumption.

I. INTRODUCTION

Large-scale systems-on-chips (SOC), consisting of many complex computational blocks on a single chip, have been enabled by an increasing transistor budget per silicon die. Along with advances in technology scaling, chip multiprocessors (CMP) have become favored over traditional superscalar processors because of their power efficiency and scalability. In a CMP system, the chip area is normally divided into a number of tiles, each containing
The effectiveness of CMP is often determined by the performance of the NOC, which is affected by many factors, including topologies [1][2][3], routing algorithms [7], switching strategies [16], and flow-control mechanisms [4][5][6]. Among these factors, topology is the one determining the upper-bound performance (e.g., throughput and communication latency) that can be achieved by an NOC. After the topology is determined, the routing algorithm, switching strategy, and flow control mechanism need to be developed to efficiently make use of the physical resources on the NOC to approach the performance upper bound. Most NOC topologies proposed in literature are based on regular interconnection structures and low-radix routers, such as 2D Mesh and Torus, for ease of implementation. However, as more tiles or network nodes are put on a single chip, the latency and throughput achievable by the NOC are getting worse because of the large network diameter [7].

[8] proposed a high-radix NOC architecture named Clos Network-on-Chip (CNOC), which is based on the well-known Clos network [9], as shown in Figure 1. Compared to a low-radix NOC, CNOC can provide much better scalability in terms of latency and throughput, and can easily accommodate several hundred nodes with a reasonable router radix. Results in [8] show that CNOC has the smallest zero-load latency, highest throughput, and best power efficiency compared to other high-radix topologies and the 2D Mesh network.

Although CNOC has the potential to achieve high throughput, the real throughput achieved depends on the specific router design, which includes at least two major factors: buffering structure and switching allocation scheme. In traditional 2D Mesh and 2D Torus NOCs, routers usually employ Virtual Channels (VCs) as their input buffering structures, and use separated or integrated VC Allocation (VA) and Switching Allocation (SA) to schedule flits of packets from the input sides to the output sides of the router. Actually, the VC-based buffering structure still partially suffers from the Head of Line (HOL) blocking problem, which degrades the throughput of the router. One major reason that low-radix routers in 2D Mesh and Torus NOCs are still using VC-based buffering structures is that the performance bottleneck in 2D Mesh and Torus topologies is the link capacity rather than the router capacity. Therefore there is no strong demand for redesigning the buffering structures and SA schemes.

As opposed to 2D Mesh and Torus, CNOC is able to achieve 100% throughput if we can provide perfect load-balancing and perfect router architecture. It is therefore worth studying how to improve the efficiency of router architecture to provide high throughput and high switch frequency.

---

1 Assuming traffic is uniform and there is a perfect traffic load-balancing scheme, the maximum throughput of a 2D Mesh/Torus is a decreasing function of the network size, which is less than 50% when the network size is larger than 8×8 [7].
Motivated by this, in this paper we propose three cut-through SA schemes based on iSLIP scheme [11] for high-radix routers, which can significantly improve the performance of CNOC without increasing too much of the implementation complexity. The contributions of the paper are summarized as follows.

(1) The proposed cut-through SA schemes can eliminate the Tail of Line (TOL) blocking suffered by the Virtual Output Queue (VOQ) buffering structure. The proposed reservation cut-through SA scheme can further cure the potential starvation problem involved in cut-through switching.

(2) Besides the TOL blocking elimination, we point out another important property of cut-through SA: its ability to improve the quality of matching obtained in every SA. To the best of our knowledge, we are the first to point out this property of cut-through switching in the NOC area.

(3) The proposed cut-through SA schemes are prototyped based on an 8×8 VOQ router. Compared to the baseline router, the router with cut-through SA can still achieve a frequency of 1GHz using 65nm technology with only 7.5% increase in area and 11% increase in power consumption.

(4) The performances of the cut-through SA schemes are evaluated based on a 64-node CNOC and a 36-node 2D Mesh network. Simulation results show that the proposed cut-through SA schemes can significantly improve the performance of CNOC/2D Mesh even in those extreme scenarios when there are long packets that cannot be switched using cut-through or the input buffer size is very small.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the architecture of CNOC. Section 4 presents three cut-through SA schemes and their advantages over the flit-mode SA schemes. Section 5 presents the hardware architecture of the VOQ router with cut-through SA along with the hardware evaluation results. In Section 6, we present the simulation results. Section 7 concludes the paper.

II. RELATED WORK

A. High-radix NOCs vs. Low-radix NOCs

Low-radix NOCs, such as the 2D Mesh and Torus networks, have the advantage of modest design complexity with a regular interconnection structure and short wires, but suffer several disadvantages including large network
diameter and energy inefficiencies because of high hop counts. Several high-radix NOC topologies have, therefore, been proposed during the past few years to overcome these disadvantages. Balfour and Dally [2] proposed Concentrated Mesh (CMesh) with express channels, which adapts the 2D Mesh by allowing several tiles to connect to the same router, and adding express channels on the edge of the Mesh. Kim et al. [1] proposed Flattened Butterfly, which has a regular floor plan as CMesh, but with an even higher router radix to achieve lower zero-load latency and better throughput. Some studies have been done to evaluate Fat Tree [13], which could be configured as a high-radix network as well.

B. High-Radix Clos Network on Chip

[8] proposed a high-radix NOC architecture named Clos Network-on-Chip (CNOC), which is based on the well-known Clos network [9]. A high-radix CNOC provides smaller zero-load latency as compared to a low-radix NOC, since the number of hops a packet traverses in the CNOC is limited to three or five. Besides low latency, another advantage of CNOC is its good load-balancing nature from the multiple paths available between any pair of PEs. Therefore, CNOC can provide much better scalability in terms of latency and throughput, and can easily accommodate several hundred nodes with a reasonable router radix. One major concern for the Clos network is its large number of long interconnection wires, which may lead to an increased routing area and power dissipation. In [8], the authors apply the “Routing over logic” layout scheme [10] to eliminate the area overhead caused by the long wires, and propose a heuristic floor-planning algorithm to minimize the power consumption caused by the long wires. Results show that CNOC has the smallest zero-load latency, highest throughput, and best power efficiency compared to other high-radix topologies and the 2D Mesh network [8].

Besides CNOC, there are several other NOC designs based on Clos network topology, such as the Scalable Programmable Integrated Network-on-chip (SPIN) [3] and Reduced Unidirectional Fat Tree (RUFT) [14]. In a SPIN network, local traffic between PEs connected to the same router does not need to traverse through the entire network. While in a 3-stage CNOC, every packet needs to traverse through the entire network to reach its destination port. [14] showed that by replacing Fat Tree with RUFT, which does not allow local traffic, hardware complexity and power consumption are significantly reduced.

C. Switching Allocation in VC-Based NOC Router

Different switching allocation schemes, such as input-first/output-first separable allocator and wavefront allocator, are proposed to increase the matching quality in a VC router. However, without any speedup in the switch or multiple iterations in SA, none of these methods can give satisfactory throughput with the conventional single-crossbar, input-
buffered VC router design. According to our simulation results, the throughput of a router with 8 VCs can only achieve 65% throughput with an input-first (or output-first) separable allocators and round-robin arbiters. There have been sophisticated designs proposed in the past few years that aim at high throughput router design [15][16][17]. These methods require a lot more hardware resources compared to the canonical VC-based, input-buffered, single-crossbar router design with separable allocators.

D. Virtual Cut-through Switching

The concept of virtual cut-through was originally proposed in the interconnection network of computer systems [18][19] to reduce the packet latency. Its main idea is to start delivering a packet to the next-hop buffer before the whole packet is received if the next-hop buffer has enough space to hold the entire packet. It is pointed out that virtual cut-through switching can provide higher throughput than wormhole switching because virtual cut-through switching can avoid a packet blocking multiple routers along the traversal path. Most NOC designs, however, prefer wormhole switching to cut-through switching because cut-through switching tends to use more memory space to store the entire packet. In [20] a layered switching is proposed for NOC (2D Mesh) as a tradeoff between cut-through and wormhole switching. Its main idea is to separate each packet into several fixed-length groups and perform cut-through switching within each group, while performing wormhole switching among different groups.

In this paper, although we use a concept similar to virtual cut-through to design the SA scheme for high-radix NOC routers, it should be noted that our research significantly differs from the previous work of cut-through switching. Previous papers study virtual cut-through switching more from a high-level point of view. Their explanations of the higher throughput achieved by virtual cut-through switching is that virtual cut-through switching can avoid a packet blocking multiple routers along the traversal path. In this paper, however, we study the virtual cut-through switching from a bottom-up approach. We integrate the concept of virtual cut-through switching into the router arbitrator design, and reveal several important properties of virtual cut-through switching that are not easily observed without the detailed design of the system.

1. The ability to develop better-quality matching in each clock cycle without using multiple iterations;
2. The potential starvation problem involved in cut-through switching;
3. The ability to cure TOL blocking in the VOQ structure (this is similar to the packet blocking problem observed in previous virtual cut-through switching papers).

Furthermore, we study cut-through switching in the context of the high-radix NOC. A sophisticated SA scheme is more important in a high-radix NOC router than in a low-radix NOC router to achieve a good performance.
III. CNOC ARCHITECTURE

Figure 1 shows a general three-stage CNOC from the logical point of view. The switch modules (SMs) in the first, second, and third stages are denoted as input modules (IMs), center modules (CMs), and output modules (OMs). The first stage consists of \( k \) IMs, each with \( n \) inputs and \( m \) outputs. The second stage consists of \( m \) CMs, each with \( k \) inputs and \( k \) outputs. The third stage consists of \( k \) OMs, each with \( m \) inputs and \( n \) outputs. Each SM has input buffers to store the packets/flits that lost the contention when they compete with other packets/flits for the same output port.

To avoid overflowing the limited buffer in the SM, there is a link-level credit-based flow-control mechanism applied between any two sides of a link. The upstream entity has to keep track of the available buffer space in the downstream entity to prevent buffer overflow. To route a packet in CNOC with the destination address decided, there are \( m \) different paths, each corresponding to a CM. In this paper, we forward the packets from each IM in a round-robin manner to different CMs for the ease of implementation and good load-balancing.

In this paper, our target is to design a CNOC with 64 PEs, so we use 24 8x8 routers in the CNOC (i.e., \( k=m=n=8 \)). Each SM in the CNOC is an input-buffered router. The input buffer structure usually is one of two types: VC and VOQ. [21] first proposed the concept of VC, which improves performance by decoupling the physical channels from the buffer resource. In the CNOC, however we prefer the VOQ structure to the VC structure because VOQ structure can eliminate the head-of-Line (HOL) blocking problem [11], from which the VC structure still partially suffers. In Figure 2, we show three 3×3 input-buffered routers with VOQ structure.

IV. SWITCHING ALLOCATION WITH CUT-THROUGH

In the rest of this paper, we assume that the NOC router uses VOQ as the input buffer structure. In an \( n \times n \) router, each input buffer is divided into \( n \) VOQs, each corresponding to one output port of the router. To use memory resources more efficiently, we adopt the sharing VOQ structure, i.e., one VOQ can take up all the space of an input buffer if all other VOQs are empty. To avoid deadlock, we reserve one flit space for each empty VOQ if its last arrived flit is not a tail flit.

Most of SA allocator in NOCs use flit-mode arbitration, and are very similar to the well-studied cell-based scheduling in IP routers, such as iSLIP [11] and DRRM [12]. Just like the difference between iSLIP and DRRM, SA schemes in NOCs have two major implementations: input-first and output-first separable allocators. In this paper, we use output-first separable allocator since it is easier to be extended to support cut-through switching.

Flit-mode SA with VOQ buffering structure causes two problems. First, although the VOQ structure eliminates the HOL blocking, flit-mode SA with VOQ structure suffers from the Tail of Line (TOL) blocking, which happens
when an input has multiple packets partially transmitted to the downstream VOQs. When this happens, these downstream VOQs are not available to be used by other inputs (since we don’t allow flits of different packets interleaved in a VOQ), resulting in switching performance degradation. In Figure 2, we show an example of TOL blocking. Packet 1 comes from input 1 of IM_1 and is destined for output 3 of CM_1. Packet 2 comes from input 3 of IM_1 and is also destined for output 3 of CM_1. Packet 3 comes from input 1 of IM_1 and is destined for output 2 of CM_1.

In the figure, VOQ_{13} of CM_1 and VOQ_{12} of CM_1 are both grabbed by input 1 of IM_1 (the partial packets in these VOQs are from input 1 of IM_1), and are not available for use by other inputs of IM_1. As a result, packet 2 is blocked at the input side of IM_1, although both input 3 and output 1 of IM_1 are free at present. The second problem is that the quality of matching obtained by the flit-mode SA with one iteration is poor. The performance of a high-radix NOC router depends on the quality of matching obtained in each cycle. Theoretically, a router needs log_2 n iterations of SA to obtain a maximal matching [11], where n is the router size. However, in a high-radix NOC router, the critical path usually resides in the arbitrator. It is infeasible for a high-radix NOC router to perform multiple iterations of SA in each cycle due to its very tight timing constraint.

In this paper, we propose using virtual cut-through switching to overcome the two problems mentioned above, so that a better switching performance can be achieved. In this section, we first introduce the flit-mode SA scheme based on iSLIP (FM-iSLIP). After that, the cut-through version of iSLIP (CT-iSLIP) is presented.

![Figure 2. TOL blocking. (The packet in VOQ_{13} of IM_1 is blocked because the downstream VOQ_{13} of CM_1 is exclusively grabbed by input 1 of IM_1)](image)

**A. FM-iSLIP**

We first give the definition of eligible VOQ in FM-iSLIP: we say a VOQ V of router R is eligible if it meets the following three conditions.

1. V is not empty (suppose the first flit of V is f);
2. The downstream buffer destined by flit f is not full;
(3) The downstream VOQ destined by flit $f$ doesn’t have a partial packet from other input of $R$.

Condition (3) given here is to guarantee there are no flits belonging to different packets interleaved in a VOQ.

FM- iSLIP uses a round-robin policy to serve inputs and outputs in the input/output arbitration. Each output has an output arbiter, which maintains a round-robin pointer to record the current highest-priority input; similarly, each input has an input arbiter, which maintains a round-robin pointer to record the current highest-priority output.

Each iteration of FM-iSLIP consists of three steps.

Step 1: input request. Each unmatched input sends a request to every output associated with the nonempty VOQ;

Step 2: output arbitration. Each unmatched output identifies the eligible VOQ requests among all received requests, and grants the first eligible VOQ request in a fixed round-robin order, starting from the current position of the pointer in the output arbiter. The pointer of the output arbiter is incremented by one location beyond the selected input if and only if the grant is accepted in step 3;

Step 3: input arbitration. If an unmatched input receives one or more grants, it accepts the one that appears next in a fixed round-robin order, starting from the current position of the pointer in the input arbiter. The pointer of the input arbiter is incremented to one location beyond the accepted output. If there is no grant received, the pointer remains where it is.

Figure 3 shows an example of FM-iSLIP with one iteration in each cycle. It shows the input buffers of a 4x4 router, as well as the input buffers of the downstream routers. Initially, all pointers in input and output arbiters are pointing to 1, and all nonempty VOQs at the input buffers of the router are eligible. In clock cycle 1, after one iteration of SA, three input/output pairs get matched (input 1-output 1, input 3-output 3, input 4-output 4). At the end of clock cycle 1, each of the matched inputs/outputs updates its pointer to the next location of the matched partner. At the beginning of clock cycle 2, VOQ$_{21}$ becomes ineligible (however, its request will still be sent out since the input
doesn’t know the status of the output), because the VOQ in the downstream router destined by the first flit of VOQ
is already grabbed by the packet from input 1. It is easy to see that in the second clock cycle, there are only two
input/output pairs matched. VOQ21 and VOQ22 will suffer from the TOL blocking (and become ineligible) in the next
few clock cycles because their associated next-hop VOQs are grabbed by packets from input 1, although in each
clock cycle input 1 can only match with one next-hop VOQ.

It is worth mentioning that a well-known phenomenon called de-synchronization effect [12][11] makes iSLIP
achieve 100% throughput under uniform traffic with one iteration of SA in each clock cycle when input buffer size is
large enough and packet size is 1 flit/cell. In the NOC context, however, due to the very limited buffer space and
variable packet lengths, the de-synchronization property is not in effect, so that flit-mode SA with one-iteration is not
enough to provide satisfactory performance.

B. Cut-through iSLIP (CT-iSLIP)

The main idea of cut-through SA is that we will not start the transmission of a packet before the downstream
router gets enough buffer space to store the whole packet. Once the transmission of a packet starts, the connection
between input and output will be preserved until the transmission of the whole packet finishes. This way, we can
guarantee that each input only grabs one next-hop VOQ at a time, and therefore the TOL blocking is eliminated.
Another reason that motivated us to use cut-through SA is to improve the matching quality without using multiple
iterations in each clock cycle.

A major concern about the feasibility of cut-through SA is that the input buffer size should not be less than the
maximum packet length. Many research efforts have revealed the distributions of packet lengths on NOCs [22][23].
Generally speaking, there are two major types of packets on the NOC of a SOC [22]. The first type is short signaling
packets, such as memory-access request packets and cache-coherence packets (invalidate approach). These short
signaling packets usually have no payloads, and consists of only header and tail flits (a length of 2 flits). The second
type is long data packets, such as data fetch/update packets and cache-coherence packets (update approach).
Normally, the payload size of the data packets equals the cache block size. According to the result in [22], the
optimal cache block size in a SOC is 64 bytes. If we take flit size to be 8 bytes, the optimal data packet length would
be 8 flits. Although there is still the chance that longer packets appear on an NOC because of some special
applications (such as real-time audio/video applications), these applications only take a very small percentage of the
total traffic load [23]. According to state-of-the-art VLSI techniques, it is easy to implement routers with input
buffers larger than eight flits. For simplicity, in this paper we assume that all packets appeared on an NOC are with
lengths no more than the input buffer size, and the header flit of every packet contains the packet length information
[24]. Later in Section VI, we will present a mixed mode SA scheme to deal with the small percentage of super-long packets which cannot be switched using cut-through.

In CT-iSLIP, the definition of eligible VOQ and pointer update policies used in input and output arbitrations are modified as follows. In CT-iSLIP, we say a VOQ $V$ of router $R$ is eligible if it meets the following three conditions.

1. $V$ is not empty (suppose the first flit of $V$ is $f$);
2. The free buffer space in the downstream buffer is large enough to hold the whole packet that flit $f$ belongs to;
3. The downstream VOQ destined by flit $f$ doesn’t have a partial packet from other input of $R$.

![Figure 4. Example of CT-iSLIP. (Notations are the same as in Figure 3; thick grey lines represent already matched input/output connections)](image)

In CT-iSLIP, we say an input/output is matched if either condition below is satisfied

1. The input/output is in the middle of a cut-through transmission;
2. The input/output is matched to an output/input early in the current clock cycle (if there are multiple iterations in one clock cycle).

A matched input/output becomes unmatched when the tail flit of the associated packet finishes its transmission. Each iteration of CT-iSLIP includes three steps.

Step 1: input request. Each unmatched input sends a request to every output associated with the nonempty VOQ with the length information of the head packet enclosed.

Step 2 and 3 are exactly the same as those in FM-iSLIP, with only the definitions of eligible VOQ and matched input/output changed.

Consider the example in Figure 4 which uses CT-iSLIP as the SA scheme (the initial statuses of input buffers and pointers are the same as those in Figure 3). The operation in clock cycle 1 is the same as that of FM-iSLIP in clock cycle 1. At the beginning of clock cycle 2, only input 2 and output 2 are unmatched (the others are in the cut-through
transmission). Therefore only input 2 will send requests to outputs. In the end of clock cycle 2, there are four input/output pairs matched, and the matching quality is better than that of FM-iSLIP.

Besides the TOL elimination, another important reason for this better matching quality is that in CT-iSLIP, once an input and an output are matched, the connection between them lasts for a relatively long period until the transmission of the whole packet finishes. So in each clock cycle, there are just partial inputs effectively joining the switch allocation (in clock cycle 2 of Figure 4, only input 2 join the allocation), which gives us an effect similar to multiple iterations of SA [11], and therefore results in better matching quality.

![Figure 5. The starvation problem in cut-through switching allocation](image)

C. **Lookahead Cut-through iSLIP (LACT-iSLIP)**

CT-iSLIP guarantees that the transmission of a packet between two routers won’t be interrupted by other packets. We can take this property into consideration when measuring the available buffer space in the downstream router, i.e., we can count the buffer occupied by the departing packet into the available buffer space. The potential benefit of this is that we can achieve a better buffer utilization, especially when the buffer size is small.

Motivated by this, we propose Look-ahead CT-iSLIP (LACT-iSLIP). Its only difference from CT-iSLIP is in condition (2) of the eligible VOQ definition, which is modified as follows.

(2) The free buffer space plus the buffer space occupied by the departing packet (if applicable) in the downstream buffer is large enough to hold the whole packet that flit $f$ belongs to.

D. **Starvation Problem in Cut-Through Switching Allocation**

Besides the good properties, cut-through switching also brings a side-effect, which is the potential starvation problem due to its intrinsic characteristic. Consider the example in Figure 5, where we show a $3 \times 3$ SM and the downstream VOQ connected to its output port 1. Suppose there are only two active flows. The first flow is from input 1 to output 1, destined for output 2 of the downstream router. Every packet in the first flow has a length of 4 flits. The second flow is from input 3 to output 1, destined for output 3 of the downstream router. Every packet in the second flow has a length of 2 flits. Suppose the arriving rates of both flows are equal to the link capacity. It is easy to see that
after the downstream buffer is filled up, packets of the second flow get into the downstream buffer immediately once the downstream buffer gets two flits of free space. Packets of the first flow, however, never get a chance to go into the downstream buffer, since the free-buffer size of the downstream buffer never goes beyond 2. We call this phenomenon the starvation problem of cut-through SA. A good SA scheme should not starve any flows.

E. Reservation Cut-through iSLIP (RevCT-iSLIP)

In the previous cut-through SA schemes, the request of a packet is granted only when the downstream buffer has enough space to store the whole packet. To resolve the starvation problem, we have to modify the cut-through SA scheme to guarantee that all packets at the input of an SM can eventually get served. Therefore, we add an exception to condition (2) of VOQ eligibility: a VOQ with a full packet arrived is eligible, even if the downstream buffer doesn’t have enough space to store the whole packet at present. We call grants sent to inputs before the downstream buffers are ready reserve grants, and call the other grants normal grants.

Based on the new definition of VOQ eligibility, we propose Reservation Cut-through iSLIP (RevCT-iSLIP), in which each iteration consists of the following three steps.

Step 1: Input Request. Each unmatched input sends a request to every output associated with the nonempty VOQ with the following information enclosed: (a) the length of the head packet of the VOQ and (b) whether the entire head packet is arrived.

Step 2: Output Arbitration. Each unmatched output identifies the eligible VOQ requests among all received requests, and grants the first eligible VOQ request in a fixed round-robin order, starting from the current position of the pointer in the output arbiter. The pointer of the output arbiter is, (a) updated to the selected input if the grant is a reserve grant (no need to be accepted), or (b) incremented to one location beyond the selected input if the grant is a normal grant and accepted in step 3.

Step 3: Input Arbitration. If an unmatched input receives one or more grants, it selects the one that appears next in a fixed round-robin order, starting from the current position of the pointer in the input arbiter. The acceptance policy and pointer update policy are given as follows. (a) If the selected grant is a normal grant, the input accepts the grant and forwards the pointer of the input arbiter to one location beyond the accepted output; (b) If the selected grant is a reserve grant, the input updates the pointer to the output corresponding to the selected grant. Nothing else is performed (No acceptance is sent). (c) If there is no grant received, the pointer remains where it is.

In RevCT-iSLIP, after a reservation between an input and an output is made, the output keeps sending reserve grants to the input, while the input does nothing but updates its pointer to the output. Once the output releases enough buffer space to hold the entire packet, it changes to send normal grant, which will be accepted immediately by the
input. After the cut-through transmission starts, it won’t be interrupted, so the look-ahead cut-through can still be applied in RevCT-iSLIP for better buffer utilization.

V. IMPLEMENTATION OF CUT-THROUGH VOQ ROUTER

In this section, we first describe the pipeline and the micro-architecture of the proposed cut-through VOQ router. Then the structure of the cut-through switch allocator (including the input arbiter and output arbiter) is discussed. Figure 6 shows our proposed VOQ router pipeline, which consists of three stages: (1) Switch Allocation (SA) and Buffer Write (BW), (2) Buffer Read (BR), and (3) Switch Traversal (ST). Since we eliminate routing computation from the pipeline, SA can be done in the first cycle. In parallel with SA, BW is also performed where the flit is written into the buffer. The next pipeline stage is the BR stage where the flit is read out of the buffer, followed by the ST stage where the flit traverses the crossbar switch.

Figure 7 shows the micro-architecture of the proposed VOQ router, which includes \( n \) input VOQ buffers, a crossbar, and a cut-through switch allocator. Credit-based flow control is used in the design. When one credit is received, it reflects that a flit-size buffer becomes available at the downstream buffer. Each time a flit sent from the input side to the output side of the switch fabric, one credit is sent to the corresponding upstream router. The credit counter is incremented upon receiving a credit from the downstream router and decremented by one after each flit sent to the downstream router.

A. Implementation of Cut-through Switch Allocator

The cut-through switch allocator performs one iteration (with three steps) of SA in each cycle, which includes three major stages. (1) Each output port first filters VOQ requests according to the available space size in the downstream buffer. A request can pass the filter if its packet length is not greater than the available space size in the

![Figure 6. Cut-through router pipeline](image)

![Figure 7. Cut-through router micro-architecture (“Look ahead Credit in” is used only in LACT-iSLIP)](image)
downstream buffer; (2) Each output port grants one of at most n VOQ requests, using an n:1 output arbiter, which is implemented using an Hierarchical Round Robin Arbiter (HRRA), as described later in this section; (3) Each input port accepts one of at most n grants using an n:1 input arbiter which is also implemented using an HRRA.

The difference between CT-iSLIP and LACT-iSLIP is that in the latter SA scheme, the available buffer space in the downstream router includes the current free buffer and the space occupied by the departing packet. Since the transmission of a packet will not be interrupted in the cut-through switching, we use look-ahead credit from the downstream router to indicate the (complete) length of the departing packet at the moment that the packet starts to leave. The remaining length of the departing packet in the downstream router can be easily calculated based on the look-ahead credit (complete packet length) and the flow control credit.

Here we use three bits to deliver the look-ahead credit (i.e., the length of the departing packet), as shown in Figure 8. The first bit is the valid signal, used to indicate the start and end of the look-ahead credit transmission for a packet. The other two bits are used to deliver the value of the look-ahead credit starting from the two least significant bits. The upstream router starts to update the estimation of the buffer space occupied by the departing packet once it receives the two least significant bits of the look-ahead credit, so that the transmission of the next packet can start as early as possible. It should be noted that bit “0”s in the most significant bits will not be delivered. Apparently, the delivery of look-ahead credit always finishes before the whole packet departs from the downstream router.

B. Hierarchical Round Robin Arbiter (HRRA)

Since the input arbiter has the similar implementation as the output arbiter, here we only consider the design of an output arbiter. A traditional round-robin arbiter consists of a programmable priority encoder [25] and a pointer indicating which input has the highest priority. The delay of the programmable priority encoder and the pointer update logic increases linearly with the arbiter size, which is not scalable.

HRRA [26] divides the arbitration process into simpler consecutive steps for better arbitration performance and scalability. In this paper, we use an 8-input HRRA, which utilizes high-performance and light-weight 4-input Sub RRAs (SRRAs), as shown in Figure 9. The operation of HRRA can be summarized as follows. The 8 requests are
divided into two groups of 4 local requests. In the first stage, each SRRA is responsible for selecting one of the 4 local requests it receives. In the second stage, another RRA (global RRA) does the arbitration between these two selected requests (one request selected per SRRA in the first stage).

HRRA provides high performance, since it eliminates certain complex operations from the traditional RRA. Only one SRRA (denoted as active SRRA) at any given time needs to deal with changing priority and selecting grant decisions between up requests (i.e., requests above the pointer’s location) and down requests (i.e., requests at or below the pointer’s location). The other SRRA works with the fixed priority, which simplifies its operation. For any SRRA, the down request output (DRQ), indicating whether there is a down request granted or not, is used as the input of the RRA at the second stage. With the Pass output indicating last request in the SRRA is just granted, the pass mechanism is used to ensure the smooth transition of active state from one SRRA to the other in the round-robin order. Further details of the operation of the HRRA can be found in [26].

C. Post-layout Simulation Result

We designed an $8 \times 8$ VOQ router with cut-through SA. The VHDL code of the router design is synthesized and analyzed by the Cadence Encounter RTL Compiler on the ST Microelectronics Company 65nm technology. Then we use SOC Encounter to do automatic place and route. Power evaluation is performed based on an activity factor of 10%. For memory read/write operations, we use the popular memory model CACTI 5.3 from HP Labs to characterize the memory delay, and area and power consumption.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Output Arbiter</th>
<th>Input Arbiter</th>
<th>Update Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>45ps</td>
<td>369ps</td>
<td>369ps</td>
<td>148ps</td>
</tr>
</tbody>
</table>

Table 1 shows the critical path of the proposed router. As we use 8-input HRRA in the router design, the delay on the critical path is less than 1000ps and hence make the router’s frequency achieve 1 GHz. To the best of our knowledge, this is the best timing performance for $8 \times 8$ NoC routers. In 2D mesh network, as the router only needs to use 4-input arbiters in its allocator, a high-frequency router was proposed under Intel technology [27]. Table 2 summarizes the evaluation results (area and power) for the proposed 8x8 cut-through router as well as the baseline router (with FM-iSLIP). Compared to the baseline router, the proposed router with cut-through SA has only 7.5% increase in area and 11% increase in power consumption.

<table>
<thead>
<tr>
<th>Router</th>
<th>Area (mm²)</th>
<th>Power (mw) @ 1GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.160</td>
<td>33.9</td>
</tr>
<tr>
<td>Cut Through</td>
<td>0.172</td>
<td>37.8</td>
</tr>
</tbody>
</table>

Table 2. Area and Power Required by an $8 \times 8$ Cut-through Router
VI. Simulation Evaluation

In this section, we evaluate the performances of CNOC and 2D Mesh with different SA schemes used.

A. Performance Evaluation on CNOC

The simulation of CNOC is performed based on a 64-PE CNOC using a cycle accurate simulator. Each SM is an 8×8 input-buffered router, in which each input buffer is divided into 8 sharing VOQs. The default input buffer size is configured as 64 flits. The default packet size is set as 8 flits. In default, all SA schemes perform one iteration in each clock cycle. The default traffic pattern is uniform traffic with Bernoulli i.i.d. arrival.

Besides the cut-through SA schemes proposed in the paper, we simulate several other SA schemes for comparison purposes:

- FM_VC_iSLIP: flit-mode iSLIP SA based on VC routers (to make fair comparison, the VC number of each input buffer is set to 8);
- FM_VOQ_iSLIP: flit-mode iSLIP SA based on VOQ routers;
- PM_DRRM [8]: packet-mode DRRM based on VOQ routers;

Figure 10 shows the average packet latency in CNOC with different SA schemes under uniform traffic pattern. We can see that the three cut-through SA schemes perform much better than the other three schemes in terms of both latency and throughput. The cut-through SA schemes provide 25% more throughput than FM_VC_iSLIP does, and 19% more than FM_VOQ_iSLIP does. Although the performances of the three cut-through SA schemes are close, we can still find a slight performance lead on LACT_iSLIP because it makes use of buffers more efficiently than the other two cut-through SA schemes.

In Figure 11, we evaluate the average packet latency in CNOC under non-uniform traffic pattern, which is defined as follows. Suppose \( \rho \) is the injection rate, and \( \rho_{i,j} \) is the traffic rate from PE \( i \) to PE \( j \). Under the non-uniform traffic pattern, \( \rho_{i,j} \) is defined in the following equations:
\[
\rho_{ij} = \begin{cases} 
\rho \cdot \left( w + \frac{1-w}{N} \right) & \text{if } j = (i+1) \mod N \\
\rho \cdot \frac{1-w}{N} & \text{otherwise}
\end{cases}
\]

where \( w \) is the unbalanced probability, and \( N \) is the total number of PEs. It is easily seen that non-uniform traffic becomes uniform traffic when \( w=0 \), and transposed traffic when \( w=1 \). In the simulation, we set \( w=0.5 \). It can be seen that the three cut-through SA schemes are still better than the other SA schemes. As a matter of fact, all SA schemes achieve performances similar as what they achieved under uniform traffic thanks to the excellent load-balancing property of CNOC.

In Figure 12, we show the influence of multiple iterations (4 vs. 1) on the performances of SA schemes. As we expected, FM_VOQ_iSLIP and PM_DRRM can achieve higher throughput with four iterations used in each clock cycle. In contrast, the throughput of the cut-through SA scheme remains almost unchanged when the iteration number is increased (We only show one cut-through SA scheme in the figure; the other two cut-though SA schemes have the same characteristic). This confirms our previous conclusion that cut-through SA schemes can achieve very good performance even with one iteration in each clock cycle. This is a desirable property for NOC routers because of the very tight timing constraint.
We tested the robustness of the cut-through SA scheme in a scenario with 1% (in packet number) 80-flit super-long packets (the traffic rate of the super-long packets is about 9.2%). Figure 13 illustrates the results. Since the long packets are larger than the input buffer size, they cannot be switched using cut-through directly. Therefore we use PM_DRRM [8] to switch packets longer than the input buffer size, while still employ RevCT_iSLIP to switch the short packets. We call this scheme Mixed-mode Cut-through iSLIP (MCT_iSLIP). It can be seen that because of the existence of super-long packets, the throughputs achieved by these SA schemes are all decreased. However, there is still 20% throughput improvement of MCT_iSLIP over the other SA schemes.

We further test the robustness of the cut-through SA schemes using packets with variable lengths, and the results are shown in Figure 14. In the simulation, we generate packets with exponentially distributed packet lengths (the average packet length is 8) and randomly selected destinations (i.e., uniform traffic). It is clear that the proposed cut-through SA schemes still work very well, and are much better than the other SA schemes.

Next, we evaluate the performance of different SA schemes under another extreme scenario: the input buffer size is very small. Figure 15 shows the packet delay in CNOC with different SA schemes when the input buffer size is only 16 flits. It is easy to see that LACT_iSLIP and RevCT_iSLIP are still the best schemes. However, the performance of CT_iSLIP becomes worse. This is due to the relatively low buffer utilization caused by the switching policy used in CT_iSLIP: a packet can start to be forwarded to the downstream buffer only after the downstream buffer frees up enough space. From the figure, we can see that LACT_iSLIP improves the buffer utilization by counting in the buffer space currently occupied by the departing packet, and therefore leads to a better performance.

B. Performance Evaluation on 2D Mesh Network

The simulation of 2D Mesh network is performed based on a 36-PE 6x6 2D Mesh. Each SM is a 5 × 5 input-buffered router, in which each input buffer is divided into 5 sharing VOQs (or VCs). Other simulation settings are the same as those in the CNOC simulation. Besides the SA schemes used in the evaluation of CNOC, we consider one more SA scheme named SPAROFLO, which is proposed in [27]. Since SPAROFLO requires a large number of VCs to work efficiently, in our simulation we give each input buffer 16 VCs when SPAROFLO is used (Please note that the other SA schemes use only 5 VOQs/VCs).
Figure 16 shows the average packet latency in 2D Mesh network with different SA schemes under the uniform traffic pattern. The cut-through SA schemes outperform the other schemes. However, the performance gain of cut-through SA schemes is not that obvious as we observed in CNOC. This is because the maximum throughput achievable by the 6x6 2D Mesh network is 66% even with a perfect SA scheme. The link capacity is the most severe bottleneck in a 2D Mesh network, rather than the SA scheme.

VII. CONCLUSION

The performance of routers is important to the overall performance of a high-radix NOC. In this paper, three cut-through SA schemes are developed. Their advantages in improving the performance of CNOC are analyzed. It should be noted, however, besides the performance, the implementation cost is also an important evaluation metric when we design a NOC router since we have very limited on-chip resources and tight timing constraint. Our hardware prototype shows that the router with the proposed cut-through SA schemes has very similar implementation costs compared to the canonical router design. The major extra cost occurs when we implement the look ahead cut-through SA, where extra communications are required between every two adjacent routers to notify the upstream router about the remaining length of the departing packet at the downstream router. The hardware cost of this communication mechanism is very low and negligible as opposed to the overall hardware cost of the NOC router.

REFERENCES


