An Easy-to-Implement and Efficient Flow Control for Deadlock-free Adaptive Routing

Yi Dai, Kai Lu*, Sheng Ma, Junsheng Chang
Department of Computer Science
National University of Defense Technology
Changsha, China
{daiyi, kailu, masheng, changjunsheng}@nudt.edu.cn

***Abstract*—Deadlock-free adaptive routing is extensively adopted in interconnection networks to improve communication bandwidth and reduce latency. However, existing deadlock-free flow control schemes either underutilize memory resources due to inefficient buffer management for simple hardware implementations, or rely on complicated coordination and synchronization mechanisms with high hardware complexity. In this work, we solve the deadlock problem from a different perspective by considering the deadlock as a lack of credit. With minor modifications of the credit accumulation procedure, our proposed full-credit flow control (FFC) ensures atomic buffer usage only based on local credit status while making full use of the buffer space. FFC can be easily integrated in the industrial router to achieve deadlock freedom with less area and power consumption, but 112% higher throughput, compared to the critical bubble scheme (CBS). We further propose a credit reservation strategy to eliminate the escape virtual channel (VC) cost for fully adaptive routing implementation. The synthesizing results demonstrate that FFC along with credit reservation (FFC-CR) can reduce the area by 29% and power consumption by 26% compared with CBS.**

***Index Terms*—deadlock-free flow control, adaptive routing, bubble atomic usage, full-credit flow control**

I. Introduction

With the rapid evolution and significant growth of artificial intelligence (AI) and machine learning (ML), the volume of data communicated between computation and storage resources has increased exponentially in both data centers and high performance computing (HPC) systems [1]. The interconnection network, which mainly dominates the communication bandwidth and latency, is increasingly becoming the bottleneck of system performance due to the ever-increasing volume of transferred data and variety of communication patterns. By taking full use of all available equivalent paths and avoiding communication hot spots, adaptive routing can effectively reduce the network latency and improve network throughput. However, adaptive routing may lead to cyclic channel dependency (termed deadlock) in both off-chip and on-chip networks. In order to ensure network availability, deadlock-free flow control mechanisms of adaptive routing become indispensable. Most flow control mechanisms use the VCs to process the deadlock incurred by adaptive routing. The VC splits a physical link into multiple independent channels and allocate a dedicated buffer to each of them at each port of routers and end-nodes. The multi-VC design has a great impact on the memory overhead and arbitration complexity since the number of buffers and arbitration requests is proportional to the number of VCs per port. More seriously, most deadlock-free flow control mechanisms allocate a separate VC set for each message type to avoid protocol-level deadlock, which further increase the number of VCs in networks.

Bubble flow control (BFC) [2] is widely used to avoid cyclic VC dependency by maintaining a bubble in the ring of the torus, or k-ary n-cube, topologies. A bubble is typically defined as an empty buffer of a fixed size. Hence packets can advance within the ring via the bubble. A lot of optimizations based on BFC have been proposed to achieve deadlock freedom for fully adaptive routing. However, due to inefficient buffer management, most of these optimizations underutilize memory resources, resulting in high latencies and low throughput.

In this paper, we propose a simple but efficient full-credit flow control (FFC) with a novel credit accumulation mechanism to guarantee deadlock freedom without any global coordination or restriction on buffer allocation. The basic idea of FFC is to guarantee the atomic usage of the bubble when it moves passively between routers. By accumulating locally released credits until the accumulated credits plus the upstream credits reaches the size of an entire bubble, our scheme ensures the integrity of the bubble during the bubble movement. Moreover, FFC only needs to be performed by the router that accesses a bubble, and other routers can still apply normal flow control with no extra processing for deadlock avoidance. In contrast, existing designs need to apply inefficient buffer management or packet arbitration to every router to achieve deadlock freedom, which causes significant performance degradation.

We implement the proposed FFC with synthesizable register transfer level (RTL) Verilog HDL, and integrate it into an industrial level router. We evaluate the FFC with a industrial HDL simulator to get more accurate and precise performance. It shows significant throughput improvement, but with much less area and power cost compared to mainstream approaches.

II. Background and motivation

Most deadlock-free flow control mechanisms for adaptive routing have been developed upon two main theories: one was proposed by William J. Dally in 1987 [3] and the other

This work is financially supported by the National Key R&D Program of China (2018YFB2202603), the Foundation of State Key Lab of HPC(202101-07), NSF of China(62172430), and NSF of Hunan Province(2021JJ10052).

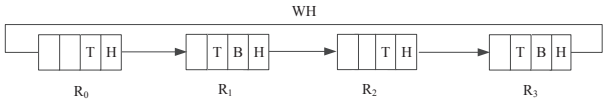

Fig. 1. Deadlock configuration with atomic buffer allocation in wormhole switching.

by *José* Duato in 1993 [4]. According to Dally's theory, a necessary and sufficient condition for deadlock-free routing is the absence of cycles in a channel dependency graph (CDG). By prohibiting certain turns to break all cyclic dependencies among channels, turn models [5] enable Dally's theory to be adapted for deadlock-free routing algorithms. However, due to the complexity of checking all cyclic channel dependencies, adaptive routing based on turn models is limited to networks with a small number of dimensions and VCs [6]. By utilizing more VCs to construct cycle-free partitions, EbDa [6] extends Dally's theory to the high-dimensional k-ary n-cube topologies. However, the minimum number of VCs for EbDa to provide fully adaptive routing in an *n*-dimensional network is $(n+1)\times 2^{(n-1)}$, which is unacceptable in a hardware resource-constrained environment. For example, the minimum number of VCs for fully adaptive routing in a 2D network is 6.

Duato's theory provides a deadlock avoidance methodology to design fully adaptive routing via a cycle-free subset of channels. Consequently, Duato's theory reduce the minimal number of VCs for deadlock-free adaptive routing to two. One serves as the adaptive VC, and the other serves as the escape VC. Each adaptive VC has a corresponding escape VC as its deadlock-free channel. Packets following the routing in adaptive VCs should be able to switch to cycle-free escape VCs whenever blockage or deadlock occurs. Duato's theory has been widely leveraged by commercial routers with virtual cut-through (VCT) switching to implement fully adaptive routing, such as the routers in the 3D torus network of the IBM Blue Gene/L supercomputers [7].

A. Atomic Buffer Allocation for Wormhole Switching

The main challenge of using Duato's theory for wormhole switching is that when a packet spans multiple routers, additional channel dependencies are introduced. To address this challenge, the atomic buffer allocation scheme [8] that assigns the whole VC buffer only to flits belonging to the same packet is typically assumed in wormhole switching networks. With atomic buffer allocation, the head flit of each packet is always in the head of buffer so that the packet can be switched to the escape VC whenever deadlock or network blocking occurs. In essence, atomic buffer allocation only addresses the head blocking problem rather than the deadlock itself. As depicted in Fig. 1, each rectangle represents a flit-sized buffer, a packet consists of a head flit denoted as H, multiple body flits denoted as B, and a tail flit denoted as T. After atomic buffer allocation, none of the head flits can move forward to the downstream VC without violating atomic buffer allocation, and the deadlock can still occur. To sum up, deadlock-free flow controls for wormhole switching are memory-underutilized due to atomic buffer allocation used to eliminate extra dependencies caused by cross-router packet storage. Ma et al. [9] argued that multiple packets can reside in one VC buffer as long as the whole packet can be accommodated, extending atomic allocation but implying VCT switching.

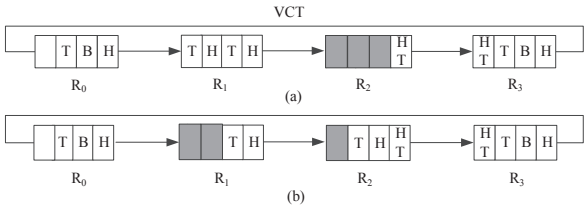

Fig. 2. Bubble fragmentation caused by variable-size packets.

B. The Bubble Fragmentation Issue in VCT Switching

Duato's theory is intensively applied in VCT switching to design fully adaptive routing algorithms. To ensure deadlock freedom, deadlock-free flow control along with deterministic routing, such as dimension-order routing (DOR), are typically applied to escape VCs. Ideally, the flow control should use a minimum number of VCs to avoid deadlock. BFC maintains at least one bubble in the ring especially after out-of-ring packet injection. This bubble prevents the packets in the ring to be blocked forever, which avoid the deadlock in one dimension. Along with the DOR algorithm, which removes the cyclic dependency across dimensions, BFC can avoid deadlock in k-ary n-cube topologies with only one VC.

However, the hardware implementation of BFC is not as simple as it appears. Although VCT flow control naturally comply with atomic buffer allocation, existing VCT deadlock-free flow controls have to regard each packet as the longest packet to avoid deadlock resulting from bubble fragmentation. A bubble is generally an empty buffer that can accommodate the longest packet. Fig. 2 demonstrates how the bubble is fragmented by variable-size packets. Assuming the length of the longest packet is 3, as shown in Fig. 2 (a), the three buffer slots marked in gray in Router 2 represent an initial bubble in the ring. When the 2-flit packet at the head of the VC buffer in router R_1 is switched to R_2, as depicted in Fig. 2 (b), two buffer slots of the bubble are replaced by this packet. Consequently, the bubble spans across R_1 and R_2. For VCT switching, a packet can be forwarded to the next router only if there is enough space for the entire packet. Therefore, the head packets with lengths of 3 and 2 in R_0 and R_1, respectively, cannot move forward because the free buffer sizes are less than the packet lengths, which causes a cyclic dependency.

C. LBFC and CBS flow control in VCT switching

The bubble fragmentation problem was first observed by the Blue Gene/L team when applying BFC to the 3D torus network [7]. To address this issue, the Blue Gene/L regards each packet as the longest one and allocates a bubble-size (also called full-size which is equal to the maximum length of a packet) VC buffer to the packet, regardless of the actual length of the data packet. In practice, this scheme is implemented by modifying the credit-based flow control to operate at the granularity of a longest packet rather than a flit.

In order to avoid global bubble tracing, a local BFC (LBFC) scheme was proposed in [2], which requires the minimum depth

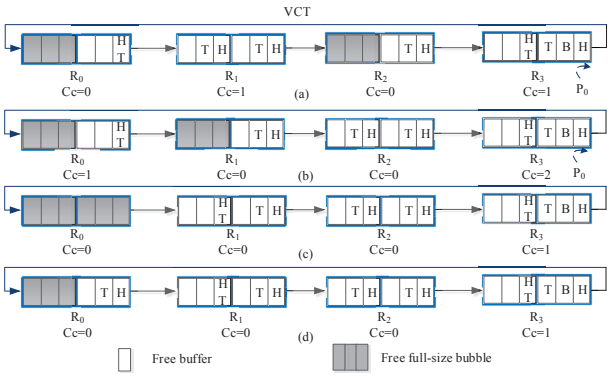

Fig. 3. Local Bubble flow control with the credit operation at the granularity of the full-size packet.

of the VC buffer to be two full-sizes. This is because the packet injection needs at least two free buffer units, one for the injected packet and the other for bubble maintenance. As shown in Fig. 3, each full-size VC buffer with a blue outline is essentially a basic credit unit and must be used as a whole, hence the initial credit value is 2. Particularly, each free full-size buffer should be treated as a bubble in LBFC. The head packet in R_1 (Fig. 3(a)) can be forwarded to R_2 as the credit counter $C_c > 0$. The forwarding of this packet moves the bubble into router R_1. Then, the head packet in R_0 can be forwarded to R_1 (Fig. 3(c)), similarly. In Fig. 3(a) the out-of-ring packet P_0 at R_3 cannot enter R_0, in case it might occupy the only bubble in the ring, as there is only one free buffer unit in R_0. In contrast, Fig. 3(b) shows that P_0 with a length of 2 is forwarded to R_0, when the available credits $C_c = 2$ which means there will be another free buffer reserved as a bubble as shown in Fig. 3(d). With the credit operating at the packet granularity, the bubble is always replaced as a whole, hence avoiding the bubble fragmentation deadlock shown in Fig. 2.

A more efficient approach for BFC implementation is the critical bubble scheme (CBS) [10], which propagates the bubble information via a control line between routers. CBS can reduce the buffer size to one longest packet since the critical bubble can be traced and recognized. However, CBS cannot efficiently handle variable-size packets, as one full-size buffer can only be used by one packet.

III. Our Approach

The above sections analyze the mainstream deadlock-free flow control schemes from the perspective of credit usage. Based on the observation of deadlock caused by the lack of credits, we propose a novel technique termed full-credit flow control (FFC) to avoid bubble fragmentation while improving buffer utilization. Our scheme considerably simplifies the hardware implementation of deadlock-free flow control with minor modifications of credit processing. In order to describe a more complicated scenario of bubble usage in an environment with diverse packet lengths carrying multiple type of messages, in the following figures, we use numbered rectangles to indicate the length of the packet residing in the VC buffer.

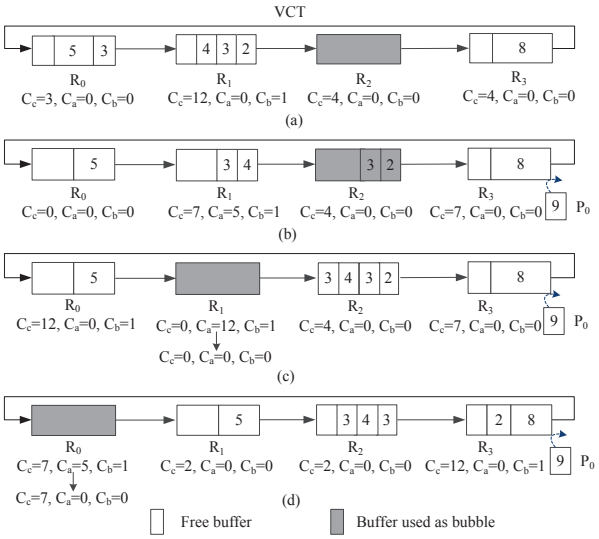

Fig. 4. A walk-through example of full-credit flow control operation.

A. Full-credit Flow Control

The main idea of FFC is to guarantee the bubble atomic swap between two adjacent routers, thus ensuring the bubble's integrity and recirculation. There are two types of buffers participate in the FFC operation. One is the buffer where the bubble lies, which is called the bubble buffer; and the other is the bubble's upstream buffer, which is called the swap buffer. A bubble swap operation refers to the packets moving forward from the swap buffer to the bubble buffer until the bubble is successfully conveyed backward to the swap buffer. For more efficient packet forwarding, FFC is only performed at the router where there exists a bubble in the downstream. As shown in Fig. 4, besides the credit counter C_c, two counters C_b and C_a are set to record the number of bubble and accumulate credits. C_c is generally used in the credit-based flow control to count the credits returned from the downstream buffer, while C_a temporarily accumulates the credits released by forwarding packets to the downstream bubble. Eventually, C_a will be returned to the upstream router once the local VC buffer become a new bubble. The bubble indicator C_b will be incremented when receiving a bubble assertion from the downstream router. The real magic exists in the introduction of C_a which prevents upstream packets from bubble usage simply by delaying the credit returning, hence avoiding bubble fragmentation while maximizing utilization of the bubble space.

Assuming that the bubble size is 12 flits and Fig. 4(a) shows a bubble marked as gray resides in R_2 and FFC should be performed at R_1. When packets with lengths of 2 and 3 are advancing to R_2, the credits released by R_1 from sending packets to the bubble buffer are counted into C_a rather than being returned to the upstream R_0. Hence, the value of C_a of R_1 is 5, and the credit counter of R_0 still remains 0; this is depicted in Fig. 4(b). This trivial modification of the credit returning behaviour prevents the packet with a length of 5 in R_0 from entering R_1 and breaking the atomic bubble swap.

Let's have a look at what would happen if this packet entered

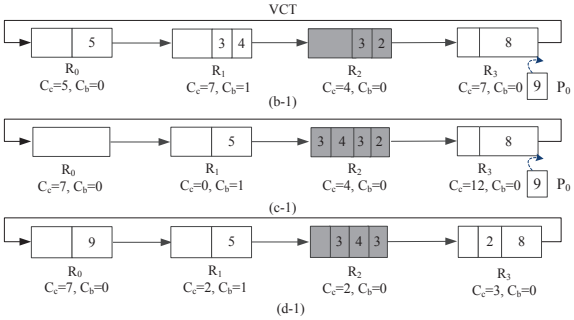

Fig. 5. A hypothetical situation of FFC with regular credit returning.

R_1 (shown in Fig. 5(c-1)) via the released 5 credits by R_1 under a general credit-based flow control. Consequently, the whole bubble could not move backward to R_1, since the free buffer slots of R_1 would be 7 which is less than a bubble size, hence the bubble was still stuck in R_2. The worst part is after R_3 receiving the returned credits of 5 from R_0, C_c of R_3 would be 12, which made the out-of-ring packet P_0 advance to R_0, since the available credit count of 12 is greater than the packet length of 9 and the downstream buffer is not used as a bubble. (refer to the packet injection rules described in Section III-C) Finally, a cyclic dependency chain was formed and no packets in the ring could make any progress as shown in Fig. 5(d-1).

In contrast, Fig. 4(b) to Fig. 4 (d) show how the deadlock shown in Fig. 5 (d-1) is avoided simply by the introduction of C_a. As discussed above, the released credits from bubble usage should be accumulated in C_a temporarily instead of returning them to the upstream router. This key step blocks the packet with the length of 5 in R_0 during the bubble swap operation. In Fig. 4 (c), after R_1 forwards the packets with lengths of 4 and 3 to R_2, the local VC buffer is empty, C_a changes from 5 to 12, and $C_c = 0$, which means the credits consumed for bubble access is up to 12 and there are no credits left. At this point the FFC operation is finished and the accumulated credits C_a and bubble counter C_b are returned to R_0; thus, the values of C_c and C_b of R_0 become 12 and 1, respectively. The counter C_a and C_b of R_1 are cleared after the completion of bubble swap between R_1 and R_2.

In essence, the number of flits eligible to use the bubble buffer and counted in C_a will eventually be returned to upstream C_c, when the sum of the above C_a and C_c equals full-credit which is a bubble size, and the bubble is successfully conveyed to the swap buffer. This full-credit condition can be satisfied by forwarding all packets from the swap buffer to the bubble buffer until the swap buffer is empty. In the case of bubble movement from R_2 to R_1, C_a at R_1 shown in Fig. 4 (c) equals full-credit and C_c at R_0 equals 0 before receiving the accumulated credits from R_2, thus the full-credit condition is satisfied and the swap buffer at R_1 becomes the new bubble. Another bubble swap example is shown in Fig. 4 (c) and Fig. 4 (d) with $C_a = 5$ and upstream $C_c = 7$. Besides, the normal credit returning is recovered in R_2 since there is no bubble in the downstream router as shown in Fig. 4 (d). Therefore, FFC is only used for bubble access hence the throughput loss resulted from it is very limited.

B. Full-credit Flow Control with Credit Reservation

Although FFC achieves deadlock freedom using only one VC by minor modification of the credit returning mechanism. For fully adaptive routing, the minimum VC cost is still 2, or rather, 2 times the original number of VCs. The number of VCs dominates the hardware complexity of control logics, which has a great proportion in chip area and power consumption.

Therefore, we further propose a credit reservation scheme, along with which FFC eliminates the escape VC cost when applying Duato's theory. Unlike the adaptive buffer dynamically allocated and shared by all adaptive VCs, the escape buffer is fixed-sized and dedicated for a specific adaptive VC, to ensure the availability of the deadlock-free escape channel at any time. By reserving a fixed amount of credits used as the escape buffer, FFC-CR (Full-Credit Flow Control with Credit Reservation) logically divides each VC buffer into two parts: one is the variable-size buffer for adaptive packets and other is the fixed-size buffer for escape packets. Moreover, the packet residing in the adaptive buffer can use the reserved credits to access the deadlock-free escape buffer when blockage or deadlock occurs.

A key step for FFC-CR to reduce the hardware complexity is to combine the arbitration requests, respectively from the adaptive and escape packets, into one request to implement fully adaptive routing with a minimum cost of just one VC. For quick deadlock recovery of the adaptive network, the packet blocked in the adaptive buffer has the highest scheduling priority. However, the unblocked packet of the adaptive buffer has a lower priority than that of the escape buffer. After singling out a higher priority request, for each VC, either the packet in escape buffer or that in adaptive buffer is granted. In this way, FFC-CR reduces the number of VCs participating in the arbitration contention to 50% of FFC, thus considerably reducing the hardware complexity of the arbitration logic. Moreover, this compare-merge operation for FFC-CR is implemented solely by the request generation module, which is independent and transparent to the arbitration logic.

Last but not least, to relieve the bottleneck caused by the very limited escape buffers, and maximize the routing flexibility, the packet in the escape buffer can be switched back to the adaptive buffer if the available adaptive credits exceed a preset threshold. The threshold value should be appropriately set to ensure that blockage does not occur frequently, which is initialized to the longest packet length for FFC.

C. Modifications to Router Microarchitecture

FFC guarantees the correctness and high efficiency of bubble flow control merely by local credit accumulation and return. So it can be easily integrated in commercial routers with less hardware cost but higher performance compared with prior work. This section details the hardware modification of the router to support adaptive routing by using FFC-CR.

Fig. 6 shows the microarchitecture of a typical VCT router. The dynamically allocated multi-queue (DAMQ) buffer shared by all v VCs is widely used in commercial routers to improve the buffer utilization [11]. The routing computation module computes the adaptive port and the escape port for each

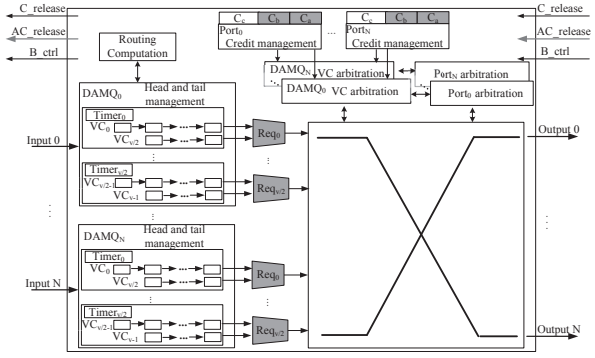

Fig. 6. The typical router architecture with FFC-CR integration.

arriving packet. The escape port follows DOR routing and the adaptive port is generated by the adaptive routing algorithm. For FFC-CR, the arbitration requests from the adaptive and its corresponding escape buffer are merged by generating one valid request according to the priority order of blocked adaptive packets, escape packets and unblocked adaptive packets, thus reducing the arbiter radix by half. Also, a timer is added for each adaptive VC to detect the network deadlock or blockage. The timer monitors the credit available signals of the adaptive buffer. When it reaches a predetermined threshold, which means the adaptive packet has not been scheduled for a while due to the lack of credits, this blocked packet must be switched to the deadlock-free escape buffer. Then the corresponding escape credits will be used by the adaptive packet to generate a valid arbitration request with the escape port, instead of the adaptive one. Besides, the adaptive packets should stop escaping and recover adaptive routing, whenever the adaptive credit is available again.

Fig. 6 shows N credit management modules, each maintaining the credit status of the input buffer for a downstream port. The counter C_c counts the downstream credit for each VC. Another two counters C_a and C_b used for local credit accumulating and bubble indication, are added only for each escape buffer to support FFC. FFC is completely implemented by the credit management module without any modification to the arbitration logic. Whenever $C_b > 0$, FFC should be executed that can be described as follows:

1) locally released credits will be counted into C_a instead of returning to the upstream router via the C_release signal, until the corresponding escape VC buffer is empty which means the full-credit is achieved;

2) assert bubble presence via the control signal B_ctrl, meanwhile return the value of C_a to the upstream router by signals AC_release;

3) clear the counters C_b and C_a while recovering normal credit returning operation.

In sum, with the trivial hardware cost of a few registers, comparators and AND/OR gates, FFC provides a simple but efficient implementation method to achieve deadlock freedom without any complicated synchronization [12] or precise coordination at a clock-cycle accuracy, such as [13], [14].

IV. Evaluation

A. Simulation Methodology

We first evaluate FFC and other deadlock-free flow control mechanisms by integrating them into synthesizable routers with different injection port configurations. For the low-radix on-chip networks, each router is configured with one injection port connecting to one end node, and four interconnection ports each for one direction of each dimension. Hence, the 4×4 and 8×8 2D torus topologies respectively with 16 nodes and 64 nodes are constructed to apply Duato's theory for adaptive routing. Another alternative topology is the concentrated [15] torus, each router is configured with 8 injection ports connected to 8 end nodes, and 16 interconnection ports providing 4 equivalent paths in each direction. As a result, a 128-node 4×4 concentrated torus network is built with lower hop count.

The baseline router is implemented with 4 VCs that is a typical VC number in most commercial designs of on-chip networks to avoid protocol-level deadlock [9]. Hence, as we discussed in Section II, the minimum number of VCs to support fully adaptive routing with LBFC or CBS is two times the original number of VC, that is, $2 \times VC = 8$ VCs. We integrate six flow control mechanisms in a synthesizable router design. LBFC-1xVC, CBS-1xVC and FFC-1xVC follow DOR algorithm to achieve deadlock freedom with only one VC. As discussed in Section II-C, the VC buffer depth of LBFC-1xVC is two longest packet lengths, which is two times that of CBS-1xVC and FFC-1xVC. Furthermore, LBFC-2xVC, CBS-2xVC and FFC-CR are implemented to evaluate the adaptive routing with an additional escape buffer for each adaptive VC. Although the escape buffer is dedicated by each VC, the adaptive buffers can be shared by all adaptive VCs.

We compare throughput and latency of these six flow control mechanisms under synthetic traffic models. The uniform traffic is generated with evenly distributed destination nodes. Hotspot traffic steers half of its load to one-third of all of the destination nodes. The packet destinations of exponential traffic follow an exponential distribution.

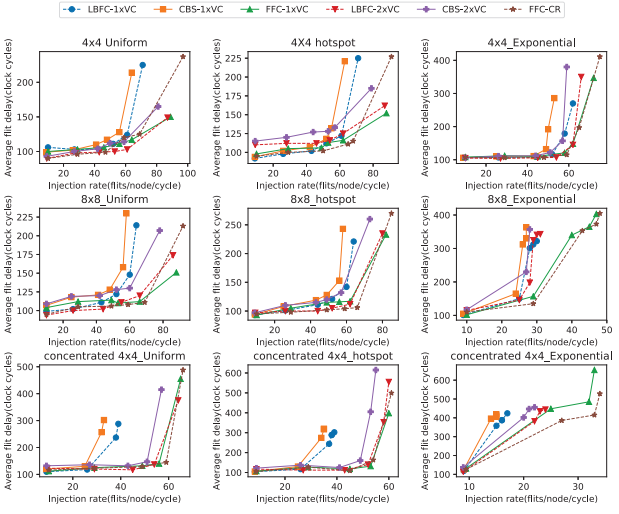

Fig. 7. Average throughput and flit delay in different network configurations.

B. Performance Analysis

We model three different network sizes to carry out the simulation, namely, an 16-node on-chip 4x4 torus, 64-node on-chip 8x8 torus, and 128-node concentrated 4x4 torus. We use the RTL verilog simulator to carry out all experiments, each running 250us. Fig. 7 shows the average throughput of all of the injection nodes and the corresponding average flit latency, under different traffic models and network configurations. FFC-1xVC exhibits the highest throughput and lowest latency among all 1xVC flow control schemes. It even performs better than CBS-2xVC, and competes with LBFC-2xVC under all of the traffic models. This distinguishing performance of FFC is derived from the efficient packet scheduling and higher buffer utilization. FFC achieves a significant 74% and 112% improvement of saturation throughput compared with CBS-1xVC, for the 8x8 on-chip torus and 4x4 concentrated torus, respectively, under unbalanced exponential traffic.

For the adaptive networks with 2xVC, FFC, CBS and LBFC only apply to the deadlock-free escape VCs. Therefore, FFC demonstrates less performance improvement in adaptive networks than it does in the single-VC (1xVC) networks. Besides, LBFC outperforms CBS by a small margin in terms of throughput and delay, but doubles the buffer depth.

C. Area and Power Comparison

We further compare the area and power consumption of the above six flow control mechanisms by synthesizing the RTL code with a 28 nm cell library. The VC number and buffer depth of each VC have been described in Section IV-A. As shown in Fig. 8, the logic including combination logic and sequential logic mainly dominates the router area and power. FFC-1xVC and CBS-1xVC almost have the same area for the same buffer consumption and arbitration complexity. However, LBFC-1xVC has 24% more SRAM area, which leads to a 9% higher overall area for doubling the buffer size. To support adaptive routing, LBFC and CBS need to double the number of VCs. However, FFC-CR considerably reduces the crossbar area by 40% compared with CBS-2xVC, resulting in an overall area reduction of 36% and 29%, respectively, compared with LBFC-2xVC and CBS-2xVC. The logic mainly consists of arbitration logic largely determines the on-chip router area. Hence, FFC-CR has the lowest area consumption among all flow control mechanisms for adaptive routing, by halving the number of arbitration requests of the VC arbiters. We also notice that although CBS reduces the memory depth by 50% compared with LBFC, the buffer area savings is only 19% and 25%, respectively, for 1xVC and 2xVC configurations. This is because SRAM depth has limited effect to area consumption since each SRAM are shared by all VCs for a DAMQ buffer.

In terms of power evaluation, the synthesis report demonstrates the dynamic power has a substantial proportion of the overall power consumption. LBFC-1xVC, CBS-1xVC, and FFC-1xVC have almost the same power consumption due to the similar SRAM structure and arbitration complexity. However, when increasing the number of VCs from 4 to 8, the logic dynamic power increases considerably. As shown in Fig. 8, LBFC-2xVC and CBS-2xVC, respectively, exhibit 38% and 40% higher dynamic power than FFC-CR. FFC-CR also achieves a 28% and 26% reduction of overall router power compared with LBFC-2xVC and CBS-2xVC, respectively.

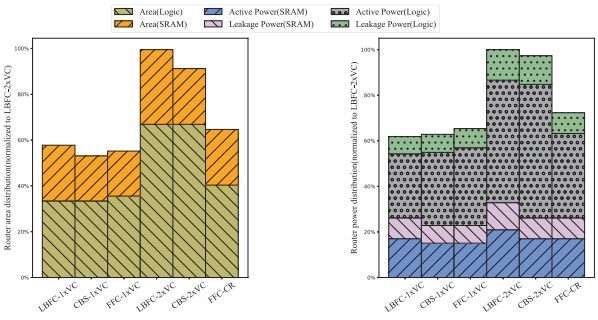

Fig. 8. Router area and power distribution with different flow controls.

V. Conclusion

In this paper, we first analyze the deadlock formation from the credit perspective, and propose a novel technique termed full-credit flow control (FFC) to efficiently solve the deadlock issue by breaking the credit loop while taking full use of the bubble space. Without any restrictions on packet arbitration or buffer allocation, FFC ensures atomic bubble usage to achieve deadlock freedom by minor modifications of the credit management mechanism. Compared with the traditional CBS, FFC achieves 74% and 112% higher throughput under exponential traffic model with heavy loads. Moreover, with an optimized arbitration strategy, FFC-CR reduces the area and power consumption by 36% and 28%, respectively, compared with LBFC; it also reduces the area by 29% and power consumption by 26% compared with CBS.

References

[1] S. Chari and M. R. Pamidi, "The Intel Omni-Path Architecture (OPA) for Machine Learning," White Paper, Dec. 2017.

[2] V. e. a. Puente et al., "The adaptive bubble router," Journal of Parallel and Distributed Computing, vol. 61, no. 9, pp. 1180-1208, 2001.

[3] W. Dally et al., "Deadlock-free message routing in multi- processor interconnection networks," IEEE TC, vol. 36, no. 5, pp. 547-553, 1987.

[4] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," IEEE TPDS, vol. 4, no. 12, pp. 1320-1331, 1993.

[5] C. J. Glass et al., "The turn model for adaptive routing," ACM SIGARCH Computer Architecture News, vol. 20, no. 2, pp. 278-287, 1992.

[6] M. Ebrahimi et al., "Ebda: A new theory on design and verification of deadlock-free interconnection networks," ISCA, 2017.

[7] A. Gara et al., "Blue gene/l torus interconnection network," IBM J. Res. Dev., vol. 49, no. 2.3, pp. 265-276, 2005.

[8] F. Verbeek et al., "On necessary and sufficient conditions for deadlock-free routing in wormhole networks," IEEE TPDS, vol. 22, no.12, 2011.

[9] S. Ma et al., "Whole packet forwarding: Efficient design of fully adaptive routing algorithms for networks-on-chip," in HPCA, pp. 1-12, 2012.

[10] L. Chen et al. "Critical bubble scheme: An efficient implementation of globally aware network flow control," in IPDPS, 2011.

[11] H. Zhang et al., "A fast and fair shared buffer for high-radix router," Journal of Circuits, Systems, and Computers, vol. 23, no. 01, 2014.

[12] M. Parasar et al., "Swap: Synchronized weaving of adjacent packets for network deadlock resolution," in MICRO 52, 2019.

[13] A. Ramrakhyani et al., "Synchronized progress in interconnection networks (spin): A new theory for deadlock freedom," in ISCA, 2018.

[14] M. Parasar et al., "Drain: Deadlock removal for arbitrary irregular networks," in HPCA, 2020.

[15] J.D. Balfour and W. J. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks," in Int. Conference on Supercomputing, June 2006.