CDRing: Reconfigurable Ring Architecture by Exploiting Cycle Decomposition of Torus Topology

Liang Wang[1,2], Leibo Liu[1*], Xiaohang Wang[3], Jie Han[4], Chenchen Deng[1], Shaojun Wei[1]
[1]Institute of Microelectronics, Tsinghua University, China [2]Beihang University, China
[3]South China University of Technology, China [4]University of Alberta, Canada
[*]Corresponding author: liulb@tsinghua.edu.cn

Abstract—**Future NoCs should be highly flexible to adapt to communication demands to achieve high scalability and low power consumption. However, the flexibility is still quite limited by the high complexity of reconfiguration for globally reconfigured channels. In this paper, we propose to augment a router-based buffered NoC with a reconfigurable ring architecture by exploiting cycle decomposition of a torus bufferless network. At runtime, the topologies of the rings can be reconfigured according to the workloads by choosing different cycle decompositions of the torus network. Because the shapes of the rings are restricted to a specified regular shape, the reconfiguration time can be reduced to a linear complexity with respect to network size, and the reconfiguration algorithm can be implemented in a distributed hardware. The experimental results show that the reconfigurable rings provide 54% and 26% improvements on packet latency and static power saving, respectively, for realistic workloads.**

Index Terms—**Reconfigurable ring, cycle decomposition, NoCs**

I. Introduction

Networks-on-Chip (NoCs) are becoming widely adopted as the interconnects of many-core systems, CPU-GPU heterogeneous systems, etc. Traditional router-based NoCs are still facing serious challenges of limited scalability and high power consumption. These issues are further aggravated by continuous technology scaling from two aspects. First, as hundreds or even thousands of cores integrated on a chip [1], the high packet latency limits the scalability. Second, with smaller feature size, the static power consumption is becoming more dominant while the network resources are underutilized. The two reasons necessitate reconfigurable NoCs that are highly flexible to adapt to dynamic traffic variations with the goal to achieve scalability and low power consumption.

Small-world NoCs have been proposed to provide additional global bypass channels for distant nodes by inserting long-range wires [2]. However, these global channels are fixed and cannot adapt to traffic variations. Current reconfigurable NoCs still face high complexity of reconfiguration for globally reconfigured channels. The main reason is that the globally reconfigured channels are generally based on complex reconfiguration algorithms (e.g., more than 10K cycles) [3], [4]. Therefore, the globally reconfigured channels cannot adapt to fine-grained traffic variations. Another reason comes from the intrinsic characteristics of the router architectures, which are more applicable to a relatively fixed topology. In contrast, rings have a much lower complexity and power consumption, and have higher potential to be globally reconfigured.

On the other hand, power gating is an effective solution to significantly reduce static power. However, power gating of NoCs suffers from high wakeup latency (6-12 cycles per router), non-negligible power overhead of frequent power gating and disconnection of the network [5], [6]. As a result, power gating is usually combined with bypass channels to address these challenges such that the packets can be transmitted through the bypass paths instead of routers. NoRD [6], which proposes a node-router decoupling technique, connects the escape virtual channels of all routers as a single ring. At a low network load, the bypass channels are locally configured such that packets can be are deflected to the ring-based path without waking up the routers. However, the single ring is not scalable and incurs high latency in a large network. Bypassing techniques, such as TooT [7] and SPONGE [8], set up bypass channels according to local traffic flows. However, the bypass channels are locally reconfigured, making the power saving efficiency sensitive to local traffic congestion. Moreover, the power saving is limited by the additional powered-on buffers or routers. Globally reconfigured bypass channels are promising to save more static power because the bypass channels can be set up according to global traffic information. However, due to the high complexity of reconfiguration, the globally reconfigured channels are less exploited for power gating.

In this paper, we propose a multi-NoC architecture that augments a buffered NoC with a novel reconfigurable ring architecture named CDRing. By exploiting cycle decomposition of the torus topology, a bufferless torus network is decomposed into a set of bufferless reconfigurable rings. At runtime, the topologies of the rings can be flexibly reconfigured according to traffic workloads by choosing different cycle decompositions of the torus network. To reduce the complexity of the cycle decomposition, the shapes of the rings are restricted to a specific regular shape, such that the complexity of the reconfiguration algorithm can be reduced to a linear complexity with respect to the network size. Thus, CDRing can quickly adapt to communication demands. The reconfiguration algorithm can be implemented in a low-overhead distributed hardware. The globally reconfigured rings can accommodate the traffic loads by reconfiguration of the bypass channels, which not only provide additional bandwidth for current intensive traffic flows, but also offer more opportunities for the power gating of routers.

II. Related Work

A. Reconfigurable NoCs

To reduce packet latency and power consumption, extensive research efforts have been devoted to reconfigurable

NoCs to adapt to traffic workload. Reconfigurable NoCs can be classified into locally reconfigurable NoCs and globally reconfigurable NoCs. For locally reconfigurable NoCs, the microarchitectures of the routers are reconfigured by setting up bypass paths, which include circuit-switching paths [9], [10], express virtual channels [11], single-cycle multi-hop paths [12] and the bypass paths for power gating [6], [7], [8]. The microarchitecture can be reconfigured cycle by cycle. However, they are unaware of the global traffic information, and only achieve limited performance improvement or power saving. Globally reconfigurable NoCs that provide global bypass channels have been proposed in [3], [13], [4], [14]. However, these reconfigurable designs rely on centralized and complex algorithms to solve optimization problems for topology reconfiguration. Although reconfigurable rings have been proposed in [15], they incur long packet detours and are only applicable to mesh topology. Runahead [16] is a multi-NoC design that augments a buffered NoC with a bufferless NoC. However, all packets are sent by both NoCs. The bufferless network allows packet dropping, thus it can only be used to reduce zero-load latency. In contrast to Runahead, we make smarter use of the bufferless network by decomposing it into a set of rings, which are used as the globally bypass channels to improve both the network latency and static power saving.

B. Power Gating of NoCs

Power gating is usually combined with reconfigurable NoCs to mitigate the high blocking latency and high overhead of power gating. Researchers [6], [7], [8] have proposed to set up bypass paths for routers to maximize the idle time of the routers and to reduce the power gating overhead. NoRD [6], a node-router decoupling technique, connects the escape VCs of all routers as a single bypass path. At a low network load, all nodes can communicate through the bypass path instead of routers. However, the single bypass path is not scalable because packets are detoured to a long non-minimal path. TooT [7] introduces a bypass technique that sets up the bypass paths for straight packets and ejection packets. SPONGE [8] keeps one ore more columns of routers powered-on and sets up bypass paths for other routers to connect them to the powered-on columns. Both of them use locally reconfigured bypass paths, which are sensitive to traffic congestion. Moreover, the power saving is limited due to the always-on buffers or rows of routers to maintain normal packet transmission. Powerpunch [5], an early-wakeup technique, exploits hop slack to hide wakeup latency and sends power control signal ahead of packets. Because the routers are powered on in advance, the packet blocking is mitigated. Topology reconfiguration or routing reconfiguration approaches that rely on complex reconfiguration algorithms are proposed in [17], [18]. However, because of the high complexity of the algorithms, they are not applicable to fine-grained variations.

C. Ring-Based NoCs

Existing research works on bufferless rings mainly focus on hierarchical rings and multiple overlapped rings. Hierarchical ring NoCs have been proposed in [19], [20], in which local rings are connected by global rings. However, because the switching between the local and global rings incurs high overhead, they are more applicable to the traffic patterns with high locality. In [21], [22], multiple overlapped bufferless rings are proposed. Compared to traditional buffered NoC, they incur much lower area overhead. However, to make sure that all nodes are connected to each other via at least a ring, there would be a large number of overlapped rings in a large network. The number of overlapped rings is limited by the overlap cap [22]. Thus, the multiple overlapped rings still have limited scalability.

Bufferless rings have a much lower hardware complexity than routers. Therefore, ring topology has the advantages of low area overhead, low power consumption and low latency (1 cycle per hop). In addition, the low complexity makes it easier to reconfigure the topology.

III. Proposed Approach

In this section, we first present how the torus network can be decomposed into a set of cycles or rings. Then we present the proposed NoC architecture that is based on the cycle decomposition of torus topology.

A. Cycle Decomposition of a 2-D Torus Network

A 2-D torus network is formally defined as a k-ary 2-cube network, in which k is the radix and 2 denotes the number of dimensions. It has been proved that a k-ary n-cube network can be decomposed into edge disjoint Hamiltonian Cycles [23]. Rather than Hamiltonian Cycle that visits every node of the original network, we focus on a more general form of cycle decomposition of a k-ary 2-cube network. The cycle decomposition has no restriction on the length or the shape of the cycles. A theorem is presented as follows,

Theorem 1. *A k-ary 2-cube network can be arbitrarily decomposed into a set of edge-disjoint cycles, and each edge of the network is embedded in one of the cycles.*

The theorem can be simply proved by assuming the edge pairing in each node. We use edge pairing to indicate that two edges or ports of the same node are directly connected. As shown in Fig. 1. Each node (router) has 4 edges (ports). When having a cycle decomposition, 2 of the 4 edges are paired and the remaining 2 edges are paired. Each edge is also paired with 2 edges of its 2 connected nodes, respectively. When two ports are paired, the messages can be directly transmitted between the two ports without arbitration. Because the number of edges is limited ($2k^2$), it guarantees to form a cycle no matter how the edges are paired for any other nodes. In other words, a message sent from a port can be guaranteed to return to current port through a cycle. Obviously, it can be proved that all edges are embedded in a cycle. A set of edge-disjoint cycles are obtained from the network.

It is also shown in Fig. 1 that there are at most 3 edge pairing patterns in each node. Fig. 2 presents three examples of cycle decomposition for a 3-ary 2-cube network. The figure shows that by arbitrarily choosing an edge pairing pattern (a or b or c) for each router, the network is decomposed into a set of edge-disjoint cycles. In the figure, *a, b, c* are corresponding to the edge pairing patterns in Fig. 1. For a k-ary 2-cube network with $M = k^2$ nodes, it is obvious that the maximum number of cycle decompositions for the

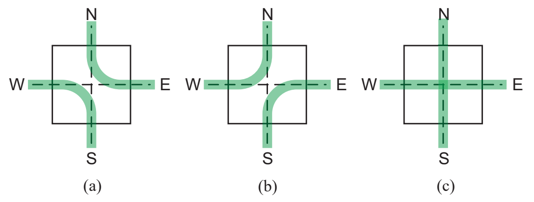


Fig. 1. Edge pairing of a node.

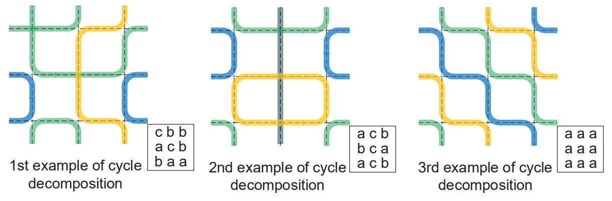

c b b
a c b
b a a
1st example of cycle decomposition
a c b
b c a
a c b
2nd example of cycle decomposition
a a a
a a a
a a a
3rd example of cycle decomposition

Fig. 2. Examples of cycle decomposition for a 3-ary 2-cube network. a, b, c denote the three edge pairing patterns in Fig. 1.

network is 3^M. This also indicates there exists a large number of variations of the reconfigurable rings. For a 2-D bufferless torus network, the cycle decomposition provides us a chance to achieve reconfigurable rings on a torus network by using different cycle decompositions.

B. Proposed Architecture

We proposed a multi-NoC architecture, which augments a router-based buffered NoC with a reconfigurable ring architecture CDRing. CDRing is actually a bufferless NoC. Both NoCs adopt torus topology. Because of the torus topology, the bufferless NoC can be decomposed into a set of reconfigurable rings. By choosing different cycle decompositions, the rings can be reconfigured according to the traffic workloads. If two nodes are connected by the same ring, the messages between the two nodes can be directly transmitted by the bufferless ring with 1 cycle per hop. Otherwise, the messages are transmitted via the buffered NoC with relatively high latency and power consumption. Because the rings are bufferless, the packet transmission is managed by deflections rather than buffering flow control. In other words, the packet continues circulating in the ring when a deflection occurs.

Different from traditional bufferless NoC [24] and bufferless rings [19], [22], CDRing can provide reconfiguration capability by choosing a cycle decomposition of a torus network. The decomposed rings provide low latency communication channels and additional bandwidth for the nodes within the same ring. On the other hand, because more packets can be transmitted through the rings without routers, the idle time of the buffered routers is maximized, allowing more power gating opportunities without incurring high blocking latency and power gating overhead.

Another concern is that the long wraparound links of the torus network could introduce high wire delay. In this paper, folded torus topology [25] is adopted for both the buffered and the bufferless network. The folded torus topology and the general torus topology share the same topological properties (i.e., the relative positions of the nodes are the same) but differ in the physical locations of the nodes. Thus, the long wraparound links are eliminated without affecting the cycle decomposition of the torus topology.

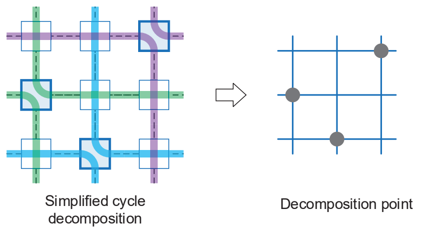


Fig. 3. Simplified cycle decomposition and decomposition point.

IV. Reconfiguration

A. Simplified Cycle Decomposition

As previously discussed, the maximum number of cycle decompositions is 3^M, in which M denotes the total number of nodes. For an 8×8 network, there exist more than 3.4×10^{30} decompositions. Thus, there is a large number of variations of the reconfigurable rings. In other words, the complexity of the cycle decompositions is extremely high, making it almost infeasible to obtain the optimal solution for the reconfiguration. On the other hand, the traffic behaviors of chip-multiprocessors exhibit high spatial and temporal variations, necessitating a fast reconfiguration to adapt to fine-grained variations.

To reduce the complexity of reconfiguration or cycle decomposition, we simplify the cycle decomposition by restricting the shape of each ring to a specified regular shape as shown in Fig. 3. Each ring connects the nodes on an entire row and an entire column, and only has 2 turns in the same node. Therefore, a $N \times N$ torus network is decomposed into N equal-length rings, and the length of each ring is $2N$ hops. As shown in Fig. 3, the node that contains 2 turns is defined as the **decomposition point**, which determines how the nodes in a column and a row are exclusively connected by a ring. Each set of decomposition points is corresponding to a cycle decomposition. The number of cycle decompositions is $N!$. The cycle decomposition problem is similar as the switch allocation in a network switch, thus it can be transformed into an allocation problem for the decomposition points. The problem can be implemented in a linear time complexity with respect to the total number of nodes by using a modified Parallel iterative matching (PIM) algorithm [25], [15].

B. Reconfiguration Algorithm and Implementation

Because the rings can provide bypass channels and additional bandwidth for the buffered network, the problem of the reconfiguration is to maximize the utilization of the rings according to current traffic behaviors. $f_{i,j}$ is used to denote the volume of the traffic flows that originate from the nodes in the i-th column to the nodes in the j-th row. In a $N \times N$ network, the problem can be formulated as follows,

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{i,j} f_{i,j} \tag{1}$$

where $\lambda_{i,j}$ indicates if the node (i, j) is a decomposition point (i, j).

Because each row and each column are exclusively connected by a decomposition point, the problem is transformed into an allocation problem. The allocation of the decomposition points is similar to the allocation problems of switch allocation in a router or a network switch. We adopt a similar allocation algorithm as [15], which uses a modified PIM algorithm for allocation. A modified two-stage allocator that supports multi-iteration allocations is designed. Each stage has N N:1 arbiters. In each arbiter, the grant is asserted to the input with the maximize $f_{i,j}$. In the first stage, the unmatched decomposition point with the maximum $f_{i,j}$ in each row is selected. Then, the grant results of the first stage are sent to the second stage as requests. The second stage performs the arbitration in each column by selecting the maximum $f_{i,j}$ from the inputs. And the end of the iteration, the rows and the columns of the granted decomposition points are set matched. After N iterations, it can be guaranteed that all decompositions points are allocated. Because the arbitrations in different rows or columns are performed in parallel, the complexity of the allocation can be reduced to linear with respect to the network size. The allocator can be implemented in a distributed manner, in which the components of the arbiter on each row/column are distributed to each node [15]. Assuming that the propagation delay of the signal within the arbiters is 1 cycle per row or per column, the delay per iteration is $2N$. Therefore, the maximum delay to obtain the solution is $2N^2$, achieving a linear time complexity with the total number of processing nodes N^2. Thus, for an 8×8 network, the maximum time to obtain a reconfiguration solution is 128 cycles.

V. Integration with Router-Based Network

XY routing is adopted in the router-based buffered torus NoC. To avoid routing deadlocks in torus, 2 virtual channels in each virtual network are adopted in the routers [25]. There are 3 virtual networks to avoid protocol deadlock. The buffered NoC is assumed to adopt traditional power gating techniques, which put the router into sleep state when it is idle for consecutive 4 cycles (break-even time [5]) and takes 8 cycles to wake up a router when to transmit a message.

A. Packet Injection Process

When a new packet arrives, the packet can either be injected to a ring or a router. If the destination can be reached by a ring and the selected output port of the ring is available, then the packet is transmitted by the ring. Otherwise, the packet is transmitted by the routers. If messages from 2 or more virtual networks can be transmitted by the ring, the virtual channel with a smaller ID is selected.

To determine if a node can be reached by a ring and which port to inject (the shortest path to destination), a routing function is adopted instead of a routing table. For a source node in the position (s_x, s_y), let (r_x^h, r_y^h) and (r_x^v, r_y^v) denote the decomposition points on the the same row and the same column, respectively. (d_x, d_y) denotes the destination node. If $d_x == r_x^h$ or $d_y == r_y^v$, then the destination can be reached by a ring. To determine the direction of the routing on the rings, the routing function is presented as follows. The decomposition point of the connected ring is denoted as (r_x^*, r_y^*). The position of the node, which is on the ring, is denoted as (id_x, id_y). Then the number on the ring (NR) of the node is calculated by the following equation,

$$\text{NR} = \begin{cases} (id_x - r_x^*) \bmod N & \text{If } id_y == r_y^* \\ (r_y^* - id_y) \bmod N + N & \text{If } id_x == r_x^* \end{cases} \tag{2}$$

where N denotes the radix of the torus network. Fig. 4(a) gives an example of the calculation, in which the NRs of the nodes increase sequentially on the ring. The incremental direction of NRs is defined as the positive direction. If NR_{dest}-$\text{NR}_{src} \leq N$, then the packet is sent to the positive direction along the ring. Otherwise, the packet is sent to the negative direction along the ring. By using this routing function, the routing tables widely used in previous ring-based interconnects [21], [22], [15] are no longer required. Thus, there is no need to update a routing table. Moreover, the routing function provides a lower area overhead for packet injection in the rings.

B. Packet Ejection Process

When multiple flits from different interconnects (2 bidirectional rings and 1 router-based NoC) can be ejected by current node, only one flit is allowed to be ejected because multiple ejection links incur a high hardware overhead. If the packets from the routers cannot proceed to the ejection buffers, they will be stalled in the input buffers of routers. If the packets from the rings cannot proceed to their ejection buffers, they will be deflected to the next node. The deflected packets will continue to circulate on the current rings and circle back to the destination later. To determine the ejection priority, we adopt the Oldest-First mechanism [24], [21] for the arbitration of the flits. The number of cycles from injection is stored in the timestamp field of the header. The packet with a higher timestamp field has a higher priority. The Oldest-First mechanism guarantees that a packet can finally arrive its destination, i.e., there is no livelock concern if there is no reconfiguration.

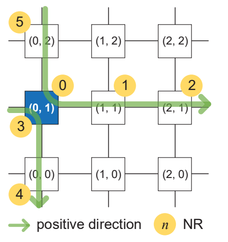
For multi-flit packets, packet contentions possibly occur when a packet arrives while the corresponding output port is occupied by an injection packet. We adopt a similar technique as [22] that uses a packet-size buffer (5-flit data packet) named Extension Buffers (EXB) to store the arriving packet temporarily when contention happens. As analyzed in [22], one EXB is enough for multiple rings.

C. Packet Draining

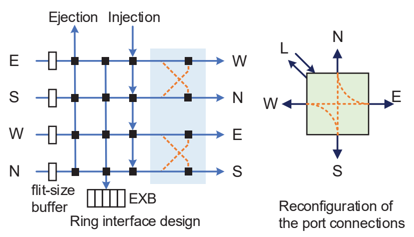
Before reconfiguration of the rings, the packets in the rings should be drained to avoid the livelock issue. When in the packet draining, the packet injection to the rings is forbidden. The packet draining time is configured as $4N$ cycles, the value of which is equal to the double length of the ring. It is possible that some packets cannot be ejected within the packet draining time due to contentions from different rings. If the deflection of a packet occurs more than 1 times within the packet draining time, the decomposition points are not allowed to be reconfigured. However, this case is rare due to the low probability of contentions in the packet draining.

D. Ring Interface Design

Fig. 4(b) presents the architecture of the ring interface (or bufferless router). Different from traditional bufferless routers

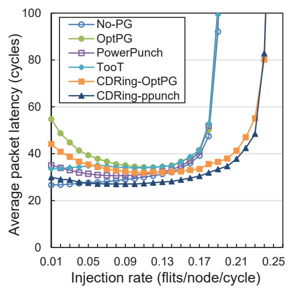


(a) Routing function example

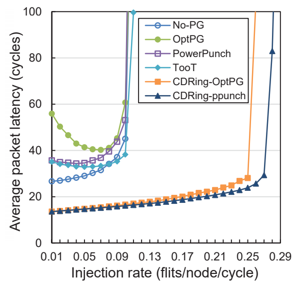


(b) Design of router interface

Fig. 4. Design of ring integration with the router-based NoC

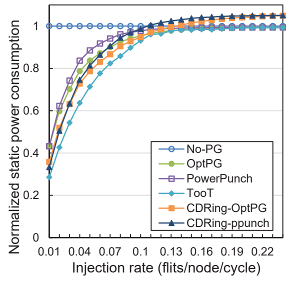


(a) Uniform traffic

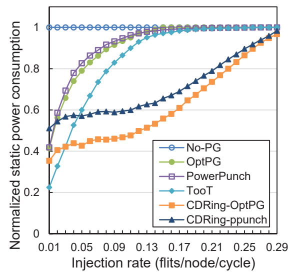


(b) Transpose traffic

Fig. 5. Average packet latency for synthetic traffic patterns a 8×8 NoC



(a) Uniform traffic



(b) Transpose traffic

Fig. 6. Normalized static power consumption for synthetic traffic patterns a 8×8 NoC

with a 5×5 crossbar, the ring interface consists of two 2×2 crossbars. The ring interface has two states, normal state and reconfiguration state. If the node is a decomposition point, then the ring interface is in the reconfiguration state, and the crossbar is enabled with two adjacent ports connected (i.e., E and N; W and S). Otherwise, the ring interface is in the normal state, and the opposite ports are connected (i.e., S and N; W and E).

VI. Experimental Results

A. Experimental Setup

The evaluations are performed in the Garnet simulator [26]. DSENT is adopted to model the power consumption [27]. The topologies of both the buffered and bufferless networks are 8×8 2D folded torus. The router-based NoC adopts the traditional input-buffered router model. There are 3 virtual networks (VNs) for protocol deadlock freedom. Each VN has 2 virtual channels (VCs). The 2 VCs are used for deadlock freedom of XY routing in the torus topology [25]. The buffer depth is 4 flits. The router latency is 3 cycles and the link latency is 1 cycle.

The proposed multi-NoC is compared with NoCs that incorporate the following power gating techniques (1) **No-PG**, router-based NoC with no power gating, (2) **OptPG**, optimized conventional power gating that uses an early wakeup signal to partially hide the wakeup latency [28], [5], (3) **PowerPunch**: an early wakeup technique [5], (4) **TooT**, a locally reconfigured power gating technique [7].

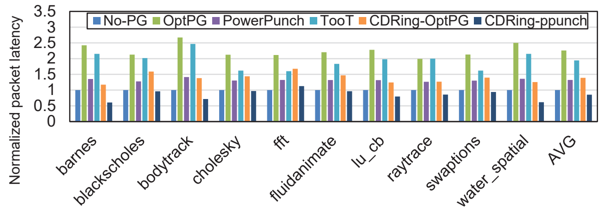
In the proposed multi-NoC architecture, the buffered NoC adopts OptPG and Powerpunch as the power gating schemes, which are named **CDRing-OptPG** and **CDRing-ppunch**, respectively. The reconfiguration epoch of CDRing is configured as 2000 cycles.

B. Packet Latency and Static Power Consumption with Synthetic Traffic Patterns

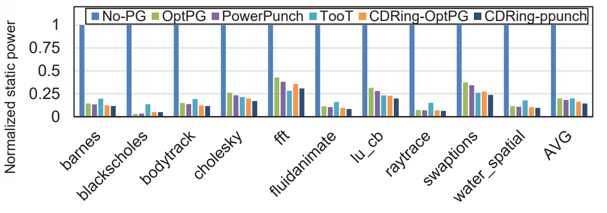
We evaluate the average packet latency and static power consumption of the network using uniform and transpose traffic patterns. The evaluation results of packet latency and power consumption are shown in Fig. 5 and Fig. 6, respectively. In Fig. 5(a) and Fig. 6(a), the traffic pattern is uniform, i.e. each node has an equal probability to send packets to another node. Therefore, CDRing does not show obvious advantages over others in terms of static power consumption. However, because CDRing provides additional bandwidth for the network, it has a higher saturation point than others. When the network gets saturated, CDRing also exhibits an additional 5.1% static power consumption due to the augmented bufferless network. The static power overhead is analyzed in Section VI-D. In Fig. 5(b) and Fig. 6(b), CDRing shows significant advantages in terms of zero-load latency, saturation point and static power consumption for the transpose traffic pattern. The main reason is that the traffic pattern consists of a set of diagonal communication pairs. By adaptively selecting diagonally arranged decomposition points, CDRing can well provide channels for the communication pairs. Therefore, the saturation point is significantly improved, and the static power consumption is much lower than others even the injection rate reaches 0.2 flits/node/cycle. In contrast, OptPG, PowerPunch and TooT are much more sensitive to the network congestion.

C. Packet Latency and Static Power Consumption for Realistic Benchmarks

Fig. 7 shows the comparisons of the packet latency and static power consumption for realistic benchmark traces, which are adopted from Synfull [29]. The configurations of Synfull in 8×8 networks are the same as [16]. Fig. 7(a) and Fig. 7(b) present the normalized packet latency and normalized static power consumption, respectively. Fig. 7(a) shows that CDRing-OptPG improves the packet latency and static power saving by 62.5% and 23.2%, respectively, over OptPG by augmenting a buffered NoC with CDRing. Moreover, the results of CDRing-ppunch show that the CDRing achieves 54.7% improvements on packet latency and 26.5% improvements on static power saving over the baseline NoC with PowerPunch. Compared to No-PG, CDRing-ppunch can incur over 86% static power reduction while the network latency is 14.5% lower. Compared to TooT, CDRing-OptPG and CDRing-ppunch provide 39.9% and 127.3% improvements in



(a) Normalized network packet latency



(b) Normalized static power consumption

Fig. 7. Packet latency and static power consumption.

terms of packet latency, and 23.6% and 39.1% improvements in terms of static power saving. The results also indicate the advantages of the globally reconfigured rings over the locally reconfigured channels.

D. Hardware Overhead

The hardware overhead includes the ring interfaces and the distributed allocator for the reconfiguration algorithm. They are implemented with Verilog HDL and synthesized using Synopsys Design Compiler with 45nm TSMC library. The results show that the areas of the bufferless network and the allocator in each node are 5956 μm^2 and 819 μm^2, respectively. Compared to a baseline router [30] with area 106972 μm^2, CDRing incurs additional 6.33% area overhead. In addition, CDRing incurs additional 5.1% static power overhead compared to a baseline router. The static power is also put into account in the previous evaluations on the static power consumption.

VII. Conclusion

In this paper, we propose a reconfigurable NoC design based on the cycle decomposition of a torus network. The reconfiguration is combined with power gating to reduce the static power consumption of NoCs. The reconfiguration can be achieved through a fast algorithm that has a linear time complexity. The experimental results show that both the latency and power consumption can be greatly reduced by using the proposed architecture. The reconfigurable rings provide over 54% and 26% improvements on packet latency and static power saving for realistic workloads. In the possible future works, more complex shapes of the rings will be designed to better exploit the potential of the cycle decomposition.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China No.61834002, in part by the National Key R&D Program of China No. 2018YFB2202101, in part by the Natural Science Foundation of China No. 61971200, the Natural Science Foundation of Guangdong Province No. 2018A030313166, Pearl River S&T Nova Program of Guangzhou No. 201806010038 and the Fundamental Research Funds for the Central Universities No. 2019MS087, the Natural Science Foundation of Tianjin No. 16JCYBJC41400.

References

[1] S. Borkar, "Thousand core chips: A technology perspective," in *DAC*, June 2007, pp. 746–749.
[2] U. Y. Ogras *et al.*, ""it's a small world after all": Noc performance optimization via long-range link insertion," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 7, pp. 693–706, July 2006.
[3] Y. Xue *et al.*, "Improving noc performance under spatio-temporal variability by runtime reconfiguration: a general mathematical framework," in *NOCS*, Aug 2016, pp. 1–8.
[4] Y. Jin *et al.*, "Communication-aware globally-coordinated on-chip networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 2, pp. 242–254, Feb 2012.
[5] L. Chen *et al.*, "Power punch: Towards non-blocking power-gating of noc routers," in *HPCA*, 2015, pp. 378–389.
[6] ——, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *MICRO*, 2012, pp. 270–281.
[7] H. Farrokhbakht *et al.*, "Toot: an efficient and scalable power-gating method for noc routers," in *NOCS*, 2016, pp. 1–8.
[8] H. Farrokhbakht *et al.*, "Sponge: A scalable pivot-based on/off gating engine for reducing static power in noc routers," in *ISLPED*, 2018, pp. 17:1–17:6.
[9] N. E. Jerger *et al.*, "Circuit-switched coherence," *IEEE Comput. Archit. L.*, vol. 6, no. 1, pp. 5–8, Jan 2007.
[10] J. Cong *et al.*, "On-chip interconnection network for accelerator-rich architectures," in *DAC*, 2015, pp. 8:1–8:6.
[11] A. Kumar *et al.*, "Express virtual channels: Towards the ideal interconnection fabric," in *ISCA*, 2007, pp. 150–161.
[12] T. Krishna *et al.*, "Breaking the on-chip latency barrier using smart," in *HPCA*, Feb 2013, pp. 378–389.
[13] M. Modarressi *et al.*, "Application-aware topology reconfiguration for on-chip networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 11, pp. 2010–2022, Nov 2011.
[14] X. Wang *et al.*, "On self-tuning networks-on-chip for dynamic network-flow dominance adaptation," in *NOCS*, April 2013, pp. 1–8.
[15] L. Wang *et al.*, "Achieving flexible global reconfiguration in nocs using reconfigurable rings," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 611–622, 2020.
[16] Z. Li *et al.*, "The runahead network-on-chip," in *HPCA*, 2016, pp. 333–344.
[17] A. Samih *et al.*, "Energy-efficient interconnect via router parking," in *HPCA*, Feb 2013, pp. 508–519.
[18] R. Parikh *et al.*, "Power-aware nocs through routing and topology reconfiguration," in *DAC*, June 2014, pp. 1–6.
[19] R. Ausavarungnirun *et al.*, "Design and evaluation of hierarchical rings with deflection routing," in *SBAC-PAD*, Oct 2014, pp. 230–237.
[20] H. Kim *et al.*, "Transportation-network-inspired network-on-chip," in *HPCA*, Feb 2014, pp. 332–343.
[21] S. Liu *et al.*, "Imr: High-performance low-cost multi-ring nocs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 6, pp. 1700–1712, June 2016.
[22] F. Alazemi *et al.*, "Routerless network-on-chip," in *HPCA*, Feb 2018, pp. 492–503.
[23] M. M. Bae *et al.*, "Gray codes for torus and edge disjoint hamiltonian cycles," in *IPDPS*, May 2000, pp. 365–370.
[24] T. Moscibroda *et al.*, "A case for bufferless routing in on-chip networks," in *ISCA*, 2009, pp. 196–207.
[25] W. Dally *et al.*, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.
[26] N. Agarwal *et al.*, "Garnet: a detailed onchip network model inside a full-system simulator," in *ISPASS*, 2009, pp. 33–42.
[27] C. Sun *et al.*, "Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *NOCS*, May 2012, pp. 201–210.
[28] Hiroki Matsutani *et al.*, "Run-time power gating of on-chip routers using look-ahead routing," in *ASP-DAC*, March 2008, pp. 55–60.
[29] M. Badr *et al.*, "Synfull: Synthetic traffic models capturing cache coherent behaviour," in *ISCA*, June 2014, pp. 109–120.
[30] D. U. Becker, "Efficient microarchitecture for network-on-chip routers," in *PhD thesis, Stanford University*, 2012.