APPLICATION-BASED DYNAMICALLY RECONFIGURABLE OPTICAL NETWORK ON CHIP ARCHITECTURE

Wenzhe Li[1,2]* , Bingli Guo[2], Guojun Yuan [1], Shanguo Huang[2], George N Rouskas[3]

[1]Institute of Computing Technology, Chinese Academy of Science, Beijing, China
[2]State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunication, Beijing, China
[3]Department of Computer Science, North Carolina State University, Raleigh, U.S.
**E-mail: lwz@bupt.edu.cn*

Keywords: RECONFIGURABLE OPTICAL NETWORK ON CHIP, RECONFIGURABLE OPTICAL ROUTER STRUCTURE, RECONFIGURATION ALGORITHM

Abstract

We propose an application based dynamically reconfigurable optical network on chip architecture and a power efficient reconfigurable optical router structure. The simulation results show the reconfigurable architecture can reduce path-setup delay by 70% compared with Torus and 43% compared with Firefly under a dynamic traffic.

1 Introduction

With the recent advances in silicon nanophotonic technology, optical network-on-chip (ONoC), which provide enormous bandwidth and higher power efficiency, becomes a promising architecture for chip multiple processors (CMP)[1]. Real-world applications running on ONoC may exhibit varied behaviors and highly dynamic traffic patterns. However, the majority of previous ONoC structures have not considered the time-varying character of applications. An ONoC with a fixed topology may achieve good performance under some traffic patterns, but perform poorly under other traffic patterns [2]. A reconfigurable ONoC has the potential to overcome this limitation by adjusting its network topology and resource reservation scheme in real time depending on the communication pattern of the application.

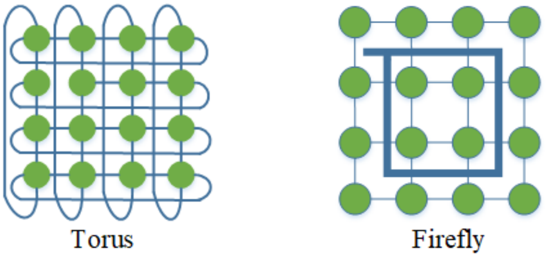


Fig. 1. Torus and Firefly

In this paper, we present a reconfigurable ONoC architecture called TFONoC, which can be configured between the Torus and Firefly topologies (shown in Fig 1) dynamically to route various traffic patterns. The topology of Firefly [3] is local mesh with global rings, while Torus is a typical mesh-like topology. Since about 50% of the Firefly and Torus topology overlap, we reuse a portion of the Torus topology to build the Firefly topology in a manner that reduces network complexity and saves footprint. To implement the TFONoC and maintain a good power efficiency, we propose a novel reconfigurable optical router (OR) structure, which can implement bidirectional transmission in the optical rings of Firefly and only add a few extra microring resonators (MR) compared with common optical router of Torus. Bidirectional transmission within the ring guarantees high link utilization and shorter transmission distance. In addition, in order to achieve better performance under different time-varying applications, we design a time-sensitive configuration algorithm for TFONoC, aiming at optimizing the overall performance of an application.

2 TFONoC Architecture

Many classic architectures have been proposed for ONoC, such as the Torus, CMesh, Butterfly, Firefly and so on, which may achieve high performance under some traffic patterns, but perform poorly under others. In this paper, TFONoC can be configured as either a Torus or Firefly network dynamically based on real-time application so as to adapt to the requirements of a wide variety of traffic patterns. The architecture of 4×4 TFONoC is shown in Fig 2. The Torus subnetwork is an electronic-controlled architecture [4], which employs an optical transmission network and an electrical control network. It is a classic and widely used topology due to its high connectivity and low network diameter. Besides, Torus has high flexibility and scalability because of its electronic-controlled communication structure. The Firefly subnetwork[3] is a hybrid ONoC, in which an electrical mesh network is used for local communication, while an optical ring network is used for global communication. In TFONoC, there already exists an optical mesh interconnection in Torus subnetwork, hence we do not have to build an additional electrical transmission network for

Firefly, making the overall structure highly power efficient. In other words, both Firefly and Torus employ optical layer for transmitting the payload.

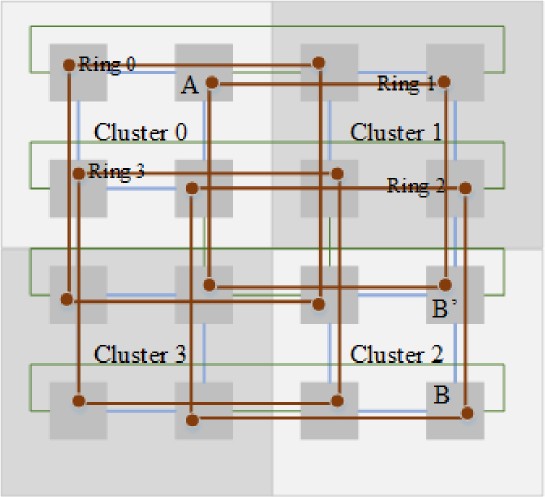
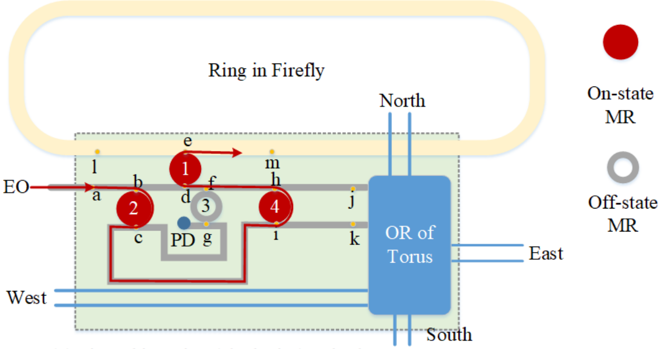


Fig. 2. 4×4 TFONoC

According to the characteristic of the topology, we employ simple XY dimension routing and first-fit wavelength allocation scheme in Torus; while we employ a grouping wavelength allocation in Firefly subnetwork to avoid contentions. Specifically, in a network with N clusters, each cluster having N nodes, all channels are divided into N groups, and each node on a ring manages a group of wavelengths. The routing in Firefly follows first global ring then XY routing in small mesh. All inter-cluster traffic only can request a wavelength from the wavelength group managed by the destination in the ring. As shown in Fig 2, when node A (Cluster 0, Ring 1) requests node B (Cluster 2, Ring 2), the node must request a wavelength from the wavelength group managed by the node B' (Cluster 2, Ring 1).

3 Compound Reconfigurable OR Structure

To implement the reconfiguration of TFONoC and maintain a good power efficiency as well, we design a novel reconfigurable compound OR, shown in Fig 3. Our proposed OR structure is built on a common OR of Torus, whose structure has been studied thoroughly in previous works, and it only needs $3w + w/n$ extra MRs (w represents the number of wavelengths in WDM and n represents the number of clusters) to implement the bidirectional transmission in optical rings. Specifically, three sets of w MRs are used to switch the signal from or to rings. On the other hand, due to wavelength grouping, only w/n MRs are needed to receive and switch the signal from the rings, also resulting in a decrease in power consumption. All channels in the rings are bidirectional, which can reduce transmission distance and improve link utilization. All 6 routes inside the compound OR for Firefly network is shown in Fig 3. When all these added MRs are off state, the compound OR can act as a Torus OR. Tuning MRs can be implemented



(1)Injected into ring (clockwise):a-d-e-l
(2)Injected into ring (anticlockwise):a-b-c-i-h-d-e-m
(3)From ring to PD (clockwise):m-e-d-f-g-PD
(4)From ring to PD (anticlockwise):l-e-d-b-c-PD
(5)From ring to OR(clockwise):m-e-d-j
(6)From ring to OR(anticlockwise):l-e-d-b-c-g-f-j

Fig. 3. The structure of reconfigurable OR

by applying a voltage on MR. With a applied voltage, the MRs switch from off-state into on-state. As an example shown in Fig 3, when TFONoC is reconfigured as Firefly topology and the signal needs to be injected into the ring anti-clockwise, the control units apply a voltage on MR$_1$, MR$_2$, and MR$_3$, so the injected signal will be coupled into these MRs and enter the ring of Firefly.

4 Reconfiguration Algorithm and Control

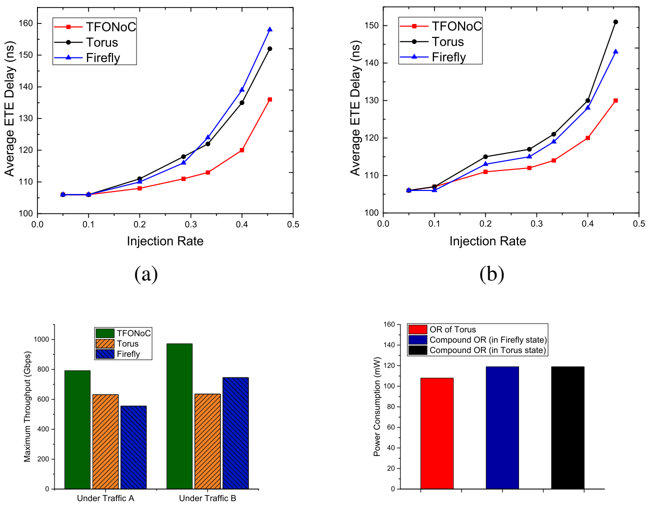
TFONoC will rebuild paths in a new subnetwork for all ongoing communications and migrate all communications to new network during reconfiguration. Considering that the setup time of new path will increase end-to-end (ETE) delay, the path in previous network continues transmitting the payload until the new one has been set up. When the new path has been set up, the previous one is torn down, and the communication is migrated into the new network.

We propose a novel reconfiguration algorithm which can maximize the performance of TFONoC on delay dynamically, but does not need to collect current network status so as to reduce control complexity. We evaluated the performance of several synthetic traffic patterns including Bit Complement, Bit Reverse, Perfect Shuffle, Tornado, Matrix Transpose and uniform traffic pattern. The simulation results show that under some of these traffic patterns the Firefly subnetwork outperforms Torus in term of ETE delay, while under other traffic patterns, the Torus topology performs better than Firefly. We use the simulation results to analyze the performance of real traffic running on TFONoC. We find a known traffic pattern that is most similar as current running traffic so as to estimate which topology will perform better under current traffic. First, the reconfiguration controller updates the current traffic matrix R periodically and then calculates its matrix distance $d(R, E)$ with each known traffic matrix E, respectively, using expression 1, in which r_{ij} and e_{ij} denote the communication volume between source i and destination j in the matrix R and E respectively. Then it determines the matrix E_s with the smallest

$d(R, E_s)$ and reconfigures the network to the topology (Torus or Firefly) that is best for this matrix if it is currently in the other topology.

$$d(R, E) = \sum_{i=1}^{n} \sum_{j=1}^{n} |r_{ij} - e_{ij}| \tag{1}$$

5 Simulation Experiment and Results Analysis



(a) (b)

(c) (d)

Fig. 4 (a) ETE delays under traffic A. (b) ETE delays under traffic B. (c) Maximum throughputs. (d) Power consumption of ORs

We developed our ONoC simulator using OMNeT++[5]. We compare our reconfigurable TFONoC with non-reconfigurable Torus and Firefly on 64 clusters (256 cores). In our simulation, 16 wavelengths are multiplexed into a waveguide, while the data rate of each wavelength is set at 10 Gbps. The payload size is assumed to be 125 Bytes, so the transmission delay of payload is a constant value (100 ns) in this simulation. To demonstrate that effectiveness of the proposed reconfiguration algorithm and the performance of TFONoC, we simulated three ONoCs (Firefly only, Torus only, and TFONoC) on two dynamic traffic patterns. Traffic pattern A is a sequence of 6 synthetic patterns, Bit Complement, Bit Reverse, Perfect Shuffle, Tornado, Matrix Transpose and Uniform, each of the six patterns lasting for a given period of time. Traffic pattern B starts as one synthetic pattern and it slowly and randomly evolves to another synthetic pattern, such that at any given point in time it represents the mix of several synthetic patterns.

Fig 4(a) shows the delay comparison under traffic A. With the increasing injection rate, the difference in ETE delay between TFONoC with Torus and Firefly becomes larger. End-to-end delay consists of path-setup delay and transmission delay of payload. Since the transmission delay is constant, we can focus on the difference of setup delay. At the injection rate of 0.45, TFONoC reduces the path-setup delay by about 39% compared with Torus and 61% compared with Firefly. This indicates that TFONoC can achieve better performance by reconfiguring the network dynamically so as to adapt to real-time traffic. The simulation results also demonstrate that the TFONoC can accurately identify the current pattern and reconfigure the network when the current topology does not match current traffic pattern. Fig 4(b) shows that TFONoC achieves good performance in terms of delay under traffic B as well. TFONoC can reduce setup delay 70% compared with Torus and 43% compared with Firefly. The TNONoC is aware when the current traffic is closer to another synthetic traffic pattern so as to make a decision whether to reconfigure. Fig 4(c) shows the TFONoC achieves the maximum throughput at 792 Gbps and 972 Gbps under traffic A and B respectively. Under traffic B, TFONoC improves the maximum throughput by 30% compared with Firefly and 53% compared with Torus, which indicates TFONoC has better performance in terms of throughput as well.

To evaluate the performance of TFONoC in terms of power efficiency, we make a simplistic analysis of the power consumption of the proposed compound optical router. We assume the injection rate to be 0.45. The power consumption of optical router includes dynamic energy 375 fJ/bit, static energy 200 μW/ring and Tuning power 100 μW/ring[6]. Fig 4(d) shows that the power consumption of Torus's OR[7] is around 108 mW, while the power consumption of compound OR is around 119 mW. TFONoC increases the power consumption of OR by 10% compared to Torus network. On the other hand, the power consumed on buffer is one of largest contributors to the overall power budget. When the network is heavily loaded, the setup delay performance mainly depends on the contentions in ONoC. So the delay comparisons indicate that there are fewer contentions in TFONoC with the injection rate of 0.45, which means that TFONoC can dramatically reduce the power consumed on buffering blocked packets. Hence, TFONoC has a competitive power performance compared with Torus network.

6 Conclusion

In this paper, we propose a reconfigurable ONoC called TFONoC, which can dynamically reconfigure the network into Torus and Firefly according to real-time traffic. We also introduce reconfigurable optical router and reconfiguration algorithm. The simulation results show that the reconfiguration algorithm can recognize the traffic pattern and the proposed architecture can achieve lower ETE delay and higher maximum throughput.

7 Acknowledgements

This work is supported in part by CAS SPRP (XDB24050200), NSFC (No.61572464, No.61771074, No.61622102), National Key R&D Program (No.2016YFB0200205), Innovative Project (No.20166060), and Open Fund of State Key Laboratory of Information Photonics and Optical Communications (BUPT) China.

8 References

[1] Wenzhe Li, Bingli Guo, Xin Li, Yu Zhou, Shanguo Huang, and George N. Rouskas. A large-scale nesting ring multi-chip architecture for manycore processor systems. *Optical Switching and Networking*, 31:183 – 192, 2019.

[2] Huaxi Gu, Zheng Chen, Yintang Yang, and Hui Ding. Ronoc: A reconfigurable architecture for application-specific optical network-on-chip. *IEICE TRANSACTIONS on Information and Systems*, 97(1):142–145, 2014.

[3] Yan Pan et al. Firefly: Illuminating future network-on-chip with nanophotonics. In *ACM SIGARCH Computer Architecture News*, volume 37, pages 429–440. ACM, 2009.

[4] Assaf Shacham, Keren Bergman, and Luca P Carloni. Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Transactions on Computers*, 57(9):1246–1260, 2008.

[5] András Varga. Discrete event simulation system. In *Proc. of the European Simulation Multiconference (ESM'2001)*, 2001.

[6] Johnnie Chan, Gilbert Hendry, Keren Bergman, and L. P. Carloni. Physical-layer modeling and system-level design of chip-scale photonic interconnection networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(10):1507–1520, 2011.

[7] Yiyuan Xie, Mahdi Nikdast, Xu Jiang, Zhang Wei, Li Qi, Xiaowen Wu, Yaoyao Ye, Wang Xuan, and Weichen Liu. Crosstalk noise and bit error rate analysis for optical network-on-chip. In *Design Automation Conference*, 2010.