RROCN: An On-Chip Network with Regular Reconfigurable Topology for Chip-Multiprocessors

Hong-Yin Luo[1] Shao-Jun Wei[2] Dong-Hui Guo[1*]

[1] Department of Electronic Engineering, Xiamen University

Xiamen, Fujian 361005, China

{hyluo, dhguo}@xmu.edu.cn

[2] Institute of Microelectronics, Tsinghua University

Beijing 100084, China

wsj@public3.bta.net.cn

Received 17 March 2013; Revised 02 April 2013; Accepted 15 April 2013

Abstract: On-chip network is a promising solution for the on-chip communication problem of large-scale CMPs and a major factor in the performance, area, and power consumption of the overall system. This paper proposes an on-chip network with regular reconfigurable topology (RROCN). The RROCN is a reconfigurable and hybrid communication structure contained routed network with 2D mesh topology and shared bus. The reconfiguration is implemented by disabling and bypassing the unwanted nodes of routed network and then organizing them as shared buses. To achieve this goal, a constructive algorithm, reconfiguration scheme, and modified XY routing algorithm with self-adaptive feature are proposed. We evaluate the RROCNs with four reconfiguration topologies and compare them with a regular on-chip network. Our results show that, with 4x4 2D mesh reconfigurable topology, the RROCN requires 41.3% less power, provides 34.5% lower zero-load latency, but provides 34.6% lower maximum throughput compared with the regular on-chip network.

Keywords: CMPs, on-chip network, reconfiguration, 2D mesh topology, XY routing algorithm

1 Introduction

On-chip network (OCN) is a promising scheme for the interconnection of many processors on a single chip [1-4] and has potential to provide low-power, low-latency, and high-bandwidth communications. To achieve this potential, an OCN should have the ability to effectively handle applications with varying bandwidth demands [5]. OCN with regular topology [6] is extensively used in the general-purpose chip-multiprocessors (CMPs), such as Intel 48-cores SCC [7] and Tilera TILEPro64 [8]. This regular OCN is normally designed to meet the highest throughput requirement of applications. For the application with lower bandwidth demand using a few processors, only parts of the regular OCN is used and thus this regular OCN is oversized. Under this circumstance, the regular OCN is used inefficiently. To enable an OCN to provide suitable bandwidth for applications, the fully customized OCN is proposed [9-11]. However, since their topologies are optimized for a specific application characteristic and generated at design time, the fully customized OCN is unpractical.

Because no single topology can provide optimal performance for all applications [12], OCN with reconfigurable topology is proposed for finding the compromise between flexibility and efficiency (performance and power) [13-16]. The reconfiguration OCNs proposed by reference [15] and [16] use the partial reconfiguration technique of FPGA [17] to reconfigure the topology. In this scheme, the specific topologies for various applications are compiled as separate bitstreams. These bitstreams are then loaded into FPGA separately at runtime according to the application requirement, such that the old topology is replaced by the new one. Since the partial reconfiguration technique of FPGA is required, this scheme is implemented only in the FPGA based system. On the other hand, a technology-independent scheme that can be implemented in FPGA or ASIC is also proposed in reference [13] and [14]. In this scheme, a set of reconfigurable structures, such as the configuration switch in [13] and the topology switch in [14], are used to construct a reconfiguration topology that could be any shape. However, the topology in [13] and [14] is fully customized for a specific traffic pattern, and some interconnections are inefficient or even inexistent for other applications. Thus, several independent applications cannot run at the same time.

In this work, a reconfigurable OCN (RROCN) is proposed to provide suitable throughput and power consumption for application with different bandwidth demands. The RROCN is a reconfigurable and hybrid scheme

contained routed network with 2D mesh topology and shared bus. For applications with the highest bandwidth demand, a routed network is constructed. For applications with the lowest bandwidth demand, the shared bus is constructed. For application with modest bandwidth demand, a hybrid structure contained both routed network and shared bus is constructed. This reconfigurable communication architecture is placed between two limits: shared bus and routed network, and thus benefits from both worlds, where the routed network provides high throughput and the shared bus provides low cost.

The RROCN is a reconfigurable network with regular topology. In the RROCN, only 2D mesh topology is considered for the reconfiguration because of the simplicity consideration of the routing algorithm. Arbitrary or other topologies are not considered in this work because their routing algorithms are complicated. Furthermore, the reconfiguration is processed at runtime. To achieve this goal, a constructive algorithm, reconfiguration scheme, and modified XY routing algorithm with self-adaptive feature are proposed. Finally, we evaluate the RROCNs with four reconfiguration topologies and compare them with a regular on-chip network.

The remainder of this paper is organized as follow. In Section 2, the architecture of RROCN is described. Section 3 describes the evaluation methodology. The results and discussions are described in Section 4. Finally, Section 5 concludes this paper.

2 Architecture of RROCN

RROCN is a regular reconfigurable OCN to provide suitable bandwidth with low cost. The reconfiguration is processed at runtime. In this section, we introduce the RROCN in the following four aspects: (1) reconfigurable topology, (2) reconfiguration scheme, (3) reconfiguration process, and (4) routing algorithm with self-adaptive feature.

2.1 Reconfigurable Topology

Architecture of the RROCN with 8x8 2D mesh topology is illustrated in Fig. 1, where a CPU core is attached to the network through the local port of routers and the peripherals are located around the network. Thus, the RROCN illustrated in Fig. 1 can attach 64 CPU cores and 32 peripherals at most. The node contains a router.

The N x N 2D mesh topology, such as the 8x8 2D mesh topology shown in Fig. 1, is the largest topology that the RROCN can construct, which is called *original topology* in this paper. The topology of RROCN could be configured into any shape under constrains of the original topology. However, with the consideration of the simplicity of routing algorithm, only 2D mesh topology is considered in this work. Therefore, based on the original topology, the RROCN is capable of construct an M x H mesh topology, where the M and H could be different but must be less than N. The original network is a routed network. After reconfiguration, the network is a hybrid communication structure contained routed network and shared bus. Several examples of the RROCN with regular reconfiguration topology are illustrated in Fig. 2. To construct a new topology, the unwanted nodes are disabled, bypassed, and then organized as shared buses. How to connect these unwanted nodes will be discussed in Section 2.3.

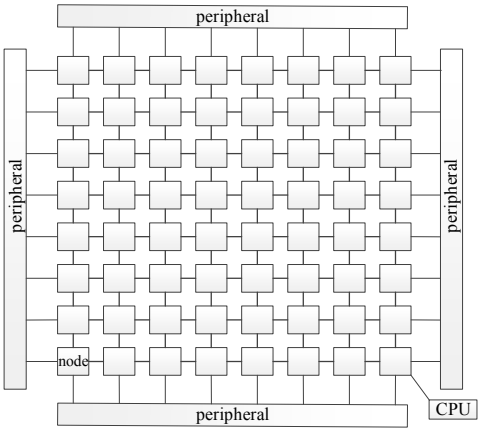


Fig. 1. RROCN with 8x8 2D mesh topology

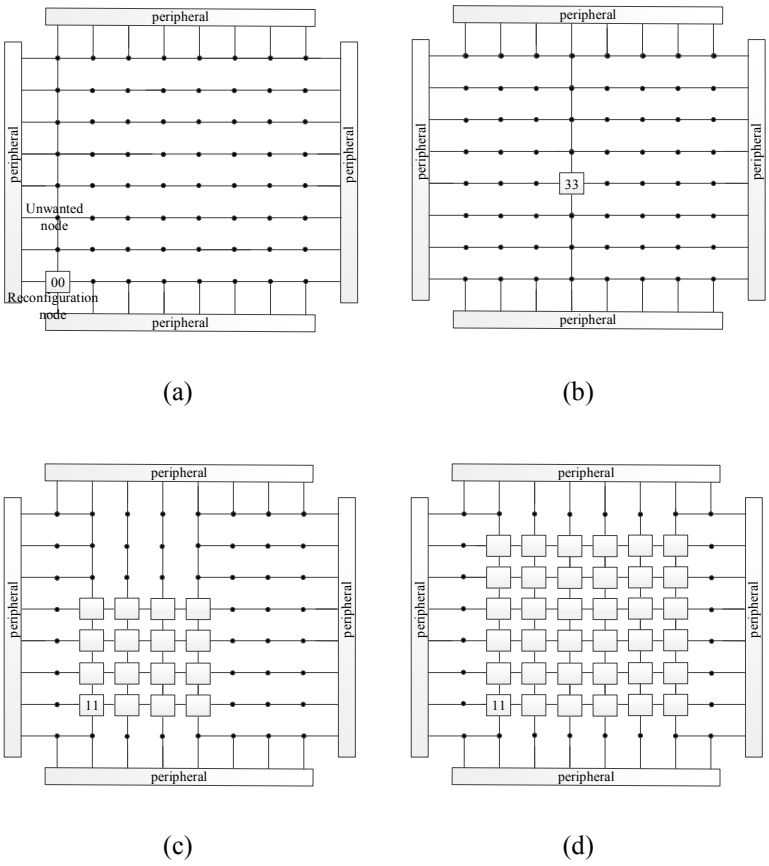


Fig. 2. RROCNs with reconfiguration topologies. The dark spot represents the unwanted node. The square represents the reconfiguration node. The unwanted nodes are disabled and bypassed. The reconfiguration nodes are activated

2.2 Reconfiguration Scheme

In an unwanted node, the router is disabled and bypassed according to a reconfiguration scheme. The block diagram of the router is illustrated in Fig. 3, where only the key modules are illustrated. The reconfiguration controller receives configuration info from the previous router, configures the crossbar and multiplexers according to a reconfiguration scheme, and then launches new configuration info to the next routers. Therefore, in an unwanted node, a packet bypass the buffers and other logics, go through the crossbar without arbitration, and then reach the next router. Furthermore, an unwanted node is also disabled. The clock of most modules within the unwanted node, including the input buffers, routing module, arbiter and other logics, is turn off for low-power consideration.

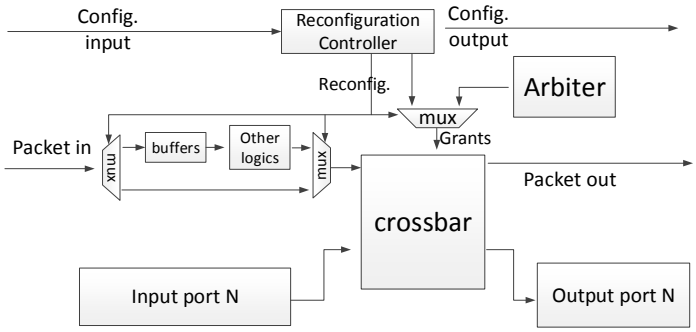


Fig. 3. Block diagram of router. Only one input port and one output port are illustrated

With the ability to bypass and disable the unwanted nodes, the network becomes to be reconfigurable. However, the routing module and arbiter in an unwanted node are disabled. In order to guarantee every peripheral is

reachable, the crossbar of the switch within an unwanted node should be carefully configured. A reconfiguration scheme for the configuration of crossbar is proposed.

The reconfiguration scheme defines how the crossbar is configured. As shown in Fig. 2, the port number of a reconfiguration topology, such as the 4x4 mesh topology in Fig. 2(c), is less than that of the original topology. To deal with this mismatch in port number, we use a broadcast scheme to configure the crossbar in an unwanted node. This broadcast scheme is illustrated in Fig. 4. The port 4 is the local port. The other four ports are divided into two groups: one is the processor group, which contains one port; another is the peripheral group, which contains three other ports. The crossbar is configured as follow: the input port in the processor group is connected to the three output ports in the peripheral group in broadcast manner. On the other hand, the three input ports in the peripheral group are all connected to the output port in the processor group through an arbiter that determines the priority and handles only the requests from peripherals.

With the broadcast scheme, a packet from a processor bypass the unwanted nodes and reaches corresponding peripherals in a broadcast manner. The peripheral then compares the destination address in the packet with its own network address and determines whether to take this packet. Furthermore, when multiple peripherals compete for the path resource, the arbiter determines the priority and grants one of them. This transmission scheme in the unwanted nodes is very similar as the shared bus. Thus, we consider these unwanted nodes are organized as the shared bus.

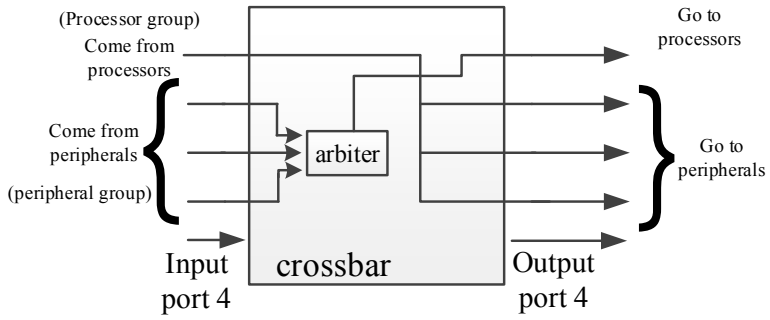


Fig. 4. Broadcast configuration scheme of the crossbar in an unwanted node. Port 4 is the local port of the router, which is used to attach a CPU core

2.3 Reconfiguration Process

The topology of RROCN can be reconfigured at runtime. A reconfiguration process is launched by a processor and then handled by the reconfiguration controllers of switches. In this section, how to build a network is presented.

Two steps are needed for building a network. First, an original node is selected as the starting point. The processor attached to the original node then starts the reconfiguration process by launching configuration info. The configuration info is actually an n-bits signal that represents the coordinate of a reconfiguration topology in four direction (+x, -x, +y, -y) based on the original node. The coordinate also represents the distance between the original node and the boundary of the reconfiguration topology. For example, in Fig. 5, the coordinate of the reconfiguration topology based on the original node 00 is (1, 0, 1, 0). Second, the configuration info is spread from the original node, reaches every node based on a constructive algorithm, and then is used by the reconfiguration controllers to configure the nodes. Finally, a new network is constructed.

In the spreading process of configuration info, an YX constructive algorithm is used. The YX constructive algorithm determines how the configuration info is spread, whether a node is enabled or disabled, and which port of the crossbar within the unwanted node is in the processor group. The Pseudocode of the YX constructive algorithm is given as follows, where configuration info is the coordinate of the reconfiguration topology (+x, -x, +y, -y).

YX Constructive algorithm(configuration info (a, b, c, d).

1: if (this is the original node) **then**
2: This node is enabled. Configuration info is spread to four directions: +x, -x, +y, -y.
3: **else if** (configuration info comes from -y direction) **then**
4: **if (**c is 0**) then**
5: This node is disabled. Configuration info become to (0, 0, 0, 0). Configuration info is spread to +x, -x, +y.

Port -y is in the processor group
6: **else**
7: This node is enabled. Configuration info become to (a, b, c-1, d). Configuration info is spread to +x, -x, +y.
8: **else if (**configuration info comes from +y direction**) then**
9: **if (**d is 0**) then**
10: This node is disabled. Configuration info become to (0, 0, 0, 0). Configuration info is spread to +x, -x, -y. Port +y is in the processor group
11: **else**
12: This node is enabled. Configuration info become to (a, b, c, d-1). Configuration info is spread to +x, -x, -y.
13: **else if (**configuration info comes from -x direction**) then**
14: **if (**a is 0**) then**
15: This node is disabled. Configuration info become to (0, 0, 0, 0). Configuration info is spread to +x. Port -x is in the processor group
16: **else**
17: This node is enabled. Configuration info become to (a-1, b, c, d). Configuration info is spread to +x, -y, +y.
18: **else if (**configuration info comes from +x direction**) then**
19: **if (**b is 0**) then**
20: This node is disabled. Configuration info become to (0, 0, 0, 0). Configuration info is spread to -x. Port +x is in the processor group
21: **else**
22: This node is enabled. Configuration info become to (a, b-1, c, d). Configuration info is spread to -x, -y, +y.

As shown in the Pseudocode, in the spreading process of configuration info, the Y direction has higher priority than the X direction. Thus, in the RROCN with (1, 0, 1, 0) coordinate illustrated in Fig. 5, the reconfiguration info from node 11 to node 10 is not used by node 10. Furthermore, because the packet from processor cannot go through the arbiter of the unwanted node, the path from node 11 to node 10 through node U0 is blocked. These types of paths are logically deleted in Fig. 5 for better observation.

The RROCN is a hybrid communication structure. Shared bus and routed network are both used to construct a reconfiguration network. As shown in Fig. 5, the nodes 00, 01, 10, and 11 constitute a 2x2 2D mesh network. The broadcast wires with gray arrows in the +y direction of node 11 is a shared bus with the distributed arbiter that is shown in Fig. 4. Thus, the reconfiguration network illustrated in Fig. 5 is composed of one routed network and eight shared bus (the wire in the +x direction of node 11 is also considered as a simple bus).

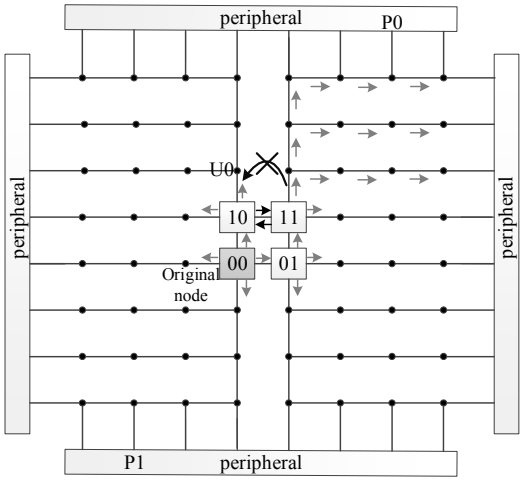


Fig. 5. RROCN with coordinate (1, 0, 1, 0) based on the original node 00

2.4 Routing Algorithm

XY routing algorithm [18] is used for a 2D mesh network with fixed topology and thus cannot be used in the RROCN. For adapting with different reconfiguration topologies, a self-adaptive XY routing algorithm is proposed.

The modified XY routing algorithm is a self-adaptive algorithm that handles all reconfiguration topologies in the RROCN. To achieve this goal, the boundary of a reconfiguration topology in x direction, which is indicated by *shape info* generated by the reconfiguration controller, is recognized by the routing algorithm. If a node is located at the boundary of a reconfiguration topology in +x or -x direction, the modified XY routing algorithm stops the routing computing to +x or -x direction and then sets this direction with lowest priority.

For example, as shown in Fig. 5, nodes 01 and 11 are located at the boundary in +x direction. Thus, the routing to +x direction has lowest priority, but that to –x direction still has the highest priority. With the modified XY routing algorithm, a packet in node 11 to peripheral P0 is forwarded to +y direction because the routing to +y direction has higher priority than that to +x direction. Then, this packet reaches its destination through the broadcast wires. Moreover, a packet in node 11 to peripheral P1 is forwarded to -x direction as normal, because the routing to –x direction still has higher priority than that to –y direction.

3 Evaluation Methodology

We used the HCS network [19] with 8x8 2D mesh topology as the regular on-chip network for comparison. For fairness, the router of the RROCN is similar as that of this regular OCN except the reconfiguration corresponding logics shown in Fig. 3. We evaluated the RROCN with four reconfiguration topologies shown in Fig. 2, and compared them with the HCS network. For a reconfiguration topology, the coordinate (1, 1, 1, 1) based on the original node 44 is defined as 1111_44. Thus, the coordinates of the reconfiguration topologies shown in Fig. 2 are 0000_00, 0000_33, 3030_11 and 5050_11.

We synthesized the RROCN and HCS network by using Synopsys Design Compiler and performed the place and route by using Synopsys IC Compiler. We used a commercial 55 nm general technology library under the worst-case condition. The floor plan utilization was set to 70%. The implementation results (area, clock period, and power consumption) were obtained by using Synopsys IC Compiler. The power consumption was estimated under the worst case.

The RTL-based hardware models are used to measure the performance (latency and throughput) using Mentor Modelsim. For the measure scheme with equal frequency, performance is measured in clock cycles and thus the packet/cycle*100 is used. For the measure scheme with maximum frequency, performance is measured in absolute time. For fairness, we use the slowest cycle time (5.06 ns) as the unit time, and thus the packet/5.06ns*100 is used. For measuring the throughput and latency, traffic pattern generators, which generate packets under uniform random traffic, are attached to the local port of every router. In the unwanted node of the RROCN, this traffic pattern generator is shut down. For fairness, the corresponding traffic pattern generator in the HCS network is also shut down. Furthermore, packet monitors are attached to the peripheral ports of the networks.

4 Results and Discussion

The RROCN is compared with the HCS network in terms of area, power consumption, clock period, latency, and maximum throughput. Four reconfiguration topologies shown in Fig. 2 are considered. Fig. 2(a) represents the worst case in timing and maximum throughput. Fig. 2(b) is used for the comparison with Fig. 2(a) to show how the timing is impacted by the position of reconfiguration nodes. Fig. 2(c) represents a case with moderate usage. Fig. 2(d) represents a case with heavy usage.

4.1 Implementation Results

For evaluating the overload of the reconfiguration logics compared with the HCS network, the RROCN with coordinate 7070_00 is also considered. The RROCN with coordinate 7070_00 has the same topology with the HCS network and thus provides the same performance in latency and throughput. The normalized implementation results are illustrated in Fig. 7, where the HCS network is set as the reference case. Because of the additional hardware logics for the reconfiguration, the RROCN occupies 31.0% more area, and requires 9.3% more power compared with the HCS network. Furthermore, the critical path in the RROCN and HCS networks are same. However, because the EDA tools use heuristic algorithm to optimize the design, clock period of the RROCN and HCS network could be different. As shown in Fig. 7, although the RROCN has 2.0% smaller clock period than the HCS network, we consider their clock period is approximately equal.

Clock period and power consumption of the RROCN with four reconfiguration topologies are illustrated in Fig. 8(a) and (b). For coordinate 0000_00, 0000_33, 3030_11 and 5050_11, the RROCN has 104.0%, 35.9%, 37.4%, and 9.7% larger clock period and 52.7%, 59.3%, 41.3%, and 14.1% less power consumption compared

with the HCS network. The RROCN constructs a reconfiguration network by disabling and bypassing the unwanted nodes. Disabling a node means that the clock of this node is turn off. Bypassing a node means that the input buffers and other logics are bypassed. A signal, which is launched from a register, then needs to go through more logics to reach the next register. As a result, the RROCN with a reconfiguration topology, such as the topology shown in Fig. 2 (a), has less power consumption but larger clock period compared with the HCS network.

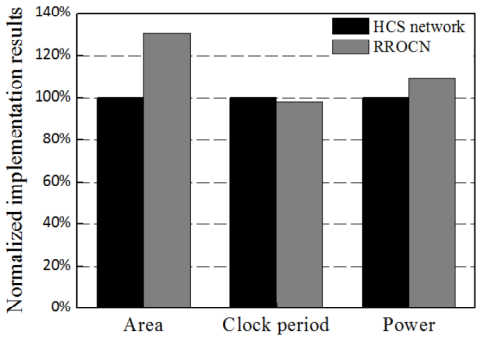
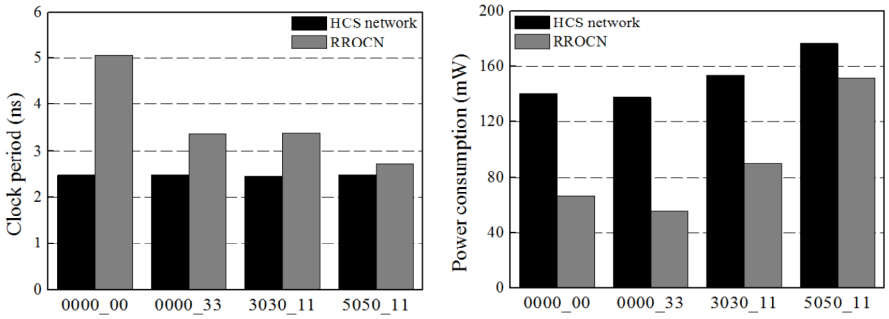
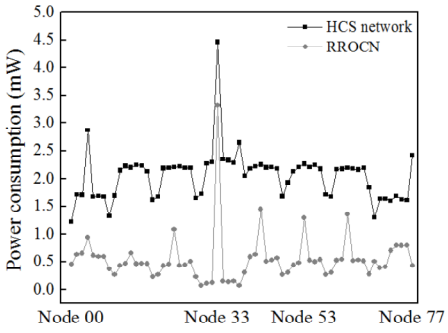


Fig. 7. Normalized implementation results for the RROCN with coordinate 7070_00



(a) (b)



(c)

Fig. 8. Implementation results. (a) Normalized clock period for four reconfiguration topologies. (b) Normalized power consumption for four reconfiguration topologies. (c) Power breakdown for coordinate 0000_33.

For investigating the source of the power saving in the RROCN, power breakdown for the RROCN with coordinate 0000_33 is illustrated in Fig. 8(c). In this RROCN, all nodes except node 33 are disabled. In these disabled nodes, most modules are bypassed and their clock is shut down, while the node 33 is fully functional. Thus the node 33 has the highest power consumption. All these disabled nodes consume less power than the corresponding nodes in the HCS network. Furthermore, in the RROCN, the unwanted node in the main path, such as

the node 53, consumes more power than other disabled nodes because it has more fanout than other nodes. More fanout means that the logic cell drives more load and thus consumes more power.

Clock period of the RROCN is varied with the variation of the topology. It is impacted by the position of the reconfiguration nodes. For example, in the RROCN with coordinate 0000_00, a packet goes through 14 unwanted nodes at most to reach its destination, which is the critical path. However, in the RROCN with coordinate 0000_33, a packet goes through 8 unwanted nodes at most to reach its destination, which is also the critical path. The path with 14 unwanted nodes is obviously longer than that with 8 unwanted nodes. Thus, the RROCN with coordinate 0000_00 has larger clock period than the RROCN with coordinate 0000_33.

4.2 Latency

The injection rate versus latency with equal and maximum frequencies is plotted in Fig. 9. We define the latency under 1% injection rate as the approximate zero-load latency. With equal frequency, for coordinate 0000_00, 0000_33, 3030_11 and 5050_11, the RROCN has 77.8%, 73.4%, 34.5% and 14.1% lower zero-load latency compared with the HCS network. On the other hand, with maximum frequency, for coordinate 0000_00, 0000_33, 3030_11 and 5050_11, the RROCN has 54.7%, 63.8%, 10.0% and 5.8% lower zero-load latency compared with the HCS network.

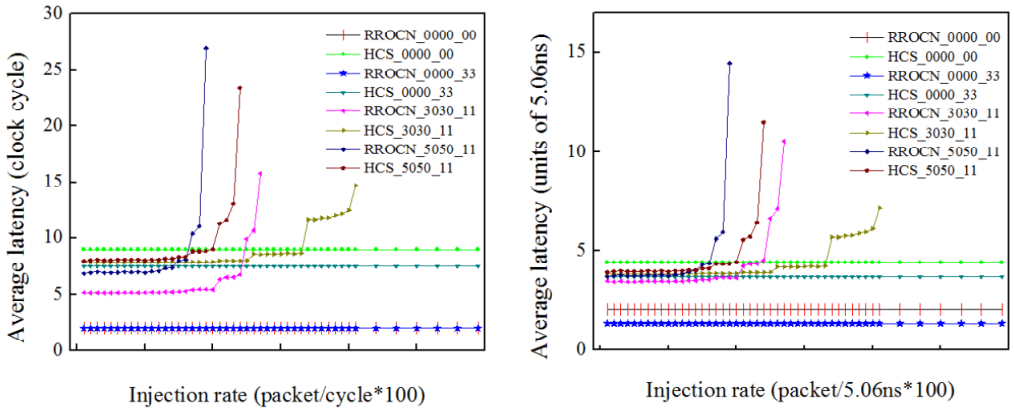


Fig. 9. Injection rate versus latency with (a) equal frequency and (b) maximum frequency

In the HCS network, a packet spends one clock to go through a node in the path to its destination. On the other hand, in the RROCN, all unwanted nodes are bypassed. A packet spends only one clock to go through these unwanted nodes. Thus, with equal frequency, the RROCN with reconfiguration topology provides lower zero-load latency than the HCS network. Moreover, this improvement in zero-load latency becomes better when the number of the unwanted node increases. Extremely, in the RROCN with coordinate 0000_00 or 0000_33, only one reconfiguration node exists. A packet spends only two clocks to reach its destination: one clock to go through the node 00 or 33 and one clock to go through rest unwanted nodes.

The bypassing path through the unwanted nodes in the RROCN is obviously longer than the path between adjacent nodes in the HCS network. Clock period of the RROCN with reconfiguration topology is larger than that of the HCS network, which is shown in Fig. 8(a). For fairness, the clock period is considered for the comparison by letting the RROCN and HCS network both run at their maximum frequency. Then, the latency is measured in absolute time (unit of 5.6 ns). Even so, the RROCN with reconfiguration topology still has lower zero-load latency than the HCS network.

With equal frequency, the RROCNs with coordinate 0000_00 and coordinate 0000_33 have the same zero-load latency. As discussed before, clock period of the RROCN with coordinate 0000_33 is smaller than that of the RROCN with coordinate 0000_00. When the maximum frequency is considered, the RROCN with coordinate 0000_33 actually has lower zero-load latency. Thus, for getting better performance, constructing a reconfiguration topology in the center of the network is a better option.

As shown in Fig. 9, for coordinate 3030_11 and 5050_11, the latency becomes higher with the increasing of the injection rate. The reason of this is clear: packets begin to block each other when the injection rate is big enough. The buffers of the RROCN is less than that of the HCS network because the unwanted nodes are disabled. Thus, the RROCN with reconfiguration topology is more sensitive about the packet blocking. As a result, the latency in the RRCON with reconfiguration topology increases faster than that in the HCS network. Due to

this rapid latency increasing, the topology size of a RROCN needs to be carefully determined. The reconfiguration topology is normally constructed for satisfying the throughput requirement. However, for avoiding this latency increasing, the topology also needs to be big enough to make the network non-congested.

4.3 Maximum Throughput

The maximum throughput with equal and maximum frequencies is plotted in Fig. 10. With equal frequency, for coordinate 0000_00 and 0000_33, the RROCN has the same maximum throughput as the HCS network. For coordinate 3030_11 and 5050_11, the RROCN has a 34.6% and 22.0% lower maximum throughput compared with the HCS network. On the other hand, with maximum frequency, for coordinate 0000_00, 0000_33, 3030_11 and 5050_11, the RROCN has a 51.0%, 26.4%, 52.4% and 28.9% lower maximum throughput compared with the HCS network.

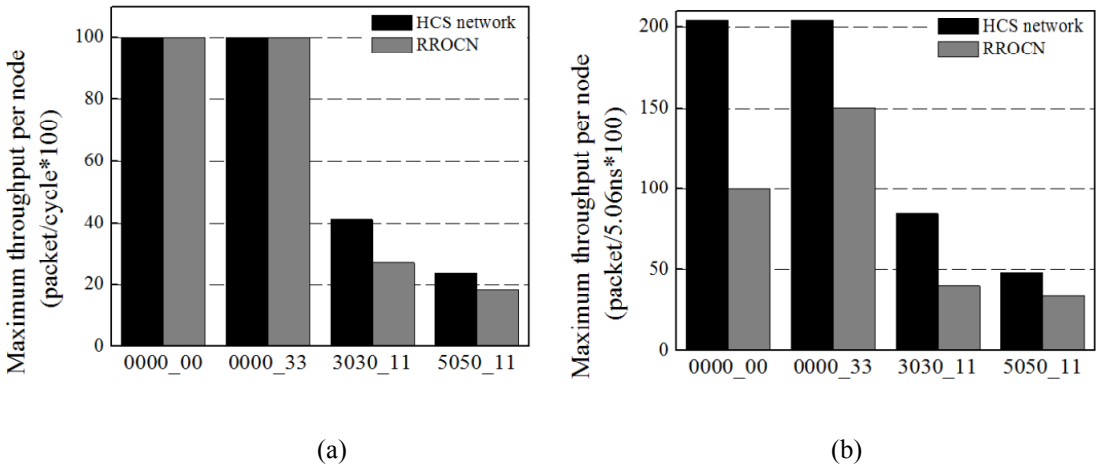


Fig. 10. Maximum throughput with (a) equal frequency and (b) maximum frequency

As shown in Fig. 10, for coordinate 0000_00 and 0000_33, the RROCN has the same maximum throughput with the HCS network. In these RROCNs, only one node (the original node) has the ability to generate packets. Since all packets come from the same one original node and move one step at one clock, these packets cannot block each other. Thus, one packet can be injected into the network at one clock at most in the RROCN with only one original node. This is also same for the HCS network with only one packet generator. This maximum throughput is the maximum limitation that one node can achieve. With the increasing of the active nodes, the maximum throughput per node decreases. The reason of this decreasing is as follows: when the number of the active node increases, the packet generator increases. More packets could block each other and thus wait longer in the network. As a result, less packets inject into the network. The maximum throughput per node then becomes lower.

The maximum throughput shown in Fig. 10 is presented in the unit of node. Furthermore, for measuring the total maximum throughput that one network can provide, this throughput per node should be multiplied by the node number. In contrast to the decreasing trend in the maximum throughput per node, the total maximum throughput actually increases with the increasing of the active nodes. This means that larger topology can provides larger total maximum throughput. This is identical to the purpose of the RROCN.

The RROCN has the ability to provide suitable throughput and power consumption for applications with varying throughput demands. To achieve this goal, the topology in the RROCN is reconfigurable. In general, smaller reconfiguration topology, has less power consumption and lower zero-load latency, but lower total maximum throughput and larger clock period than the larger reconfiguration topology. Overall, for a demanded throughput that is less than the maximum limitation that one network can provide, the RROCN with reconfiguration topology is better than the regular on-chip network in terms of power consumption and zero-load latency.

5 Conclusion

In this work, we propose the RROCN for chip-multiprocessors for reducing the power consumption under a demanded throughput. With the constructive algorithm and reconfiguration scheme, the topology of RROCN could be configured into any 2D mesh topology under constrains of the original topology at runtime. The modi-

fied XY routing algorithm is also proposed to guarantee a packet can reach its destination. The RROCN is composed of shared bus and routed network. It is a hybrid scheme and thus takes advantages from both worlds.

We evaluated the RROCN with four specific reconfiguration topologies and compared them with a regular on-chip network. The results show that, for a reconfiguration topology, the RROCN has less power consumption and lower zero-load latency, but equal or lower maximum throughput compared with the regular on-chip network. Furthermore, with the increasing of the reconfiguration nodes, the total maximum throughput of the RROCN increases at the cost of the degradation in zero-load latency and power consumption. Thus, for getting better power efficiency, the reconfiguration topology needs to be carefully chosen for a traffic pattern.

For an application with specific throughput demand, the RROCN could be configured with a topology that provides suitable throughput with less power consumption and lower zero-load latency. Furthermore, for acquiring lower latency, the RROCN could be configured with a topology that provides sufficient throughput to make the network non-congested. For acquiring less power consumption, the RROCN could be configured with a topology with less reconfiguration nodes.

Due to the reconfiguration ability, the RROCN could be reconfigured for the optimization purpose in throughput, latency or power consumption or for finding the compromise between throughput, latency and power consumption. Therefore, the RROCN is a flexible scheme for chip-multiprocessors.

Acknowledgements

The authors would like to thank Mengrao Tang for the technical assistance and valuable suggestion about the improvement of the work. The authors would also like to thank their laboratory team member's assistance. The work is supported by the National Natural Science Foundation of China (No. 61274133).

References

[1] S. Borkar, "Thousand Core Chips: A Technology Perspective," in *Proceedings of 44th Annual Design Automation Conference (DAC 2007)*, pp. 746-749, ACM Press, 2007.

[2] L. Benini, G.D. Micheli, "Networks on Chips: A New SoC Paradigm," *Computer*, Vol. 35, No. 1, pp. 70-78, 2002.

[3] W.J. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proceedings of 38th Annual Design Automation Conference (DAC 2001)*, pp. 684-689, ACM Press, 2001.

[4] M. Forsell, "A Scalable High-Performance Computing Solution for Networks on Chips," *IEEE Micro Magazine*, Vol. 22, No. 5, pp. 46-55, 2002.

[5] J. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, L. S. Peh, "Research Challenges for On-chip Interconnection Networks," *IEEE Micro Magazine*, Vol. 27, No. 5, pp. 96-108, 2007.

[6] T. Bjerregaard, S. Mahadevan, "A Survey of Research and Practices of Network-on-chip," *ACM Computing Surveys*, Vol. 38, No. 1, pp. 1-51, 2006.

[7] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De1, R.V. Wijngaart, T. Mattson, "A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS," in *Proceedings of 2010 IEEE Solid-State Circuits Conference (ISSCC 2010)*, pp. 108-109, IEEE Press, 2010.

[8] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. Mackay, M. Reif, L. Bao, J. Brown, M. Mattina, C. C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, J. Zook. "TILE64 TM Pro-cessor: A 64-Core SoC with Mesh Interconnect," in *Proceedings of 2008 IEEE Solid-State Circuits Conference (ISSCC 2008)*, pp. 88-598, IEEE Press, 2008.

[9] U. Y. Ogras, R. Marculescu, "Energy- and Performance-driven Customized Architecture Synthesis Using a Decomposition Approach," in *Proceedings of 2005 Design, Automation and Test in Europe Conference and Exposition (DATE 2005)*, pp. 352-357, IEEE Press, 2005.

[10] J. Chan, S. Parameswaran, "NoCOUT : NoC Topology Generation with Mixed Packet-Switched and Point-to-point Networks," in *Proceedings of 13th Asia and South Pacific Design Automation Conference (ASP-DAC 2008)*, pp. 265-270, 2008.

[11] U. Y. Ogras, R. Marculescu, "Application-specific Network-on-chip Architecture Customization via Long-Range Link Insertion," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2005)*, pages 246-253, 2005.

[12] M. Kim, J. Davis, M. Oskin, T. Austin, "Polymorphic on-chip networks," Proc. ISCA, pp. 101-112, 2008.

[13] M. Modarressi, "Application-aware Topology Reconfiguration for On-chip Networks," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 19, No. 11, pp. 2010-2022, 2011.

[14] M. B. Stuart, M. B. Stensgaard, J. Sparsø, "The ReNoC Reconfigurable Network-on-Chip," *ACM Transactions on Embedded Computing Systems*, Vol. 10, No. 4, pp. 1-26, 2011.

[15] V. Rana, D. Atienza, M. Santambrogio, "A reconfigurable network-on-chip architecture for optimal multi-processor SoC communication," *VLSI-SoC: Design Methodologies for SoC and SiP*, Vol. 313, pp. 232-250, 2010.

[16] L. Devaux, S. Ben Sassi, S. Pillement, D. Chillet, D. Demigny, "Flexible Interconnection Network for Dynamically and Partially Reconfigurable Architectures," *International Journal of Reconfigurable Computing*, vol. 2010, pp. 1-15, 2010.

[17] XILINX, *Partial Reconfiguration User Guide*, http://www.xilinx.com/support/documenttion/sw_manuals/xilinx13_1/ug702.pdf, 2011.

[18] L. M. Ni and P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *IEEE Transactions on Computers*, Vol. 26, No. 2, pp. 62-76, 1993.

[19] H. Luo, S. Wei, D. Chen and D. Guo, "Hybrid Circuit-Switched NOC for Low Cost On-Chip Communication," in *Proceedings of 6th IEEE International Conference on Anti-Counterfeiting, Security and Identification (ASID 2012)*, pp. 180-184, IEEE Press, 2012.