A high-throughput network on-chip in full-mesh architecture

Hongyu Meng^{1,2a}, Lei Yang¹, Zijun Liu¹, and Donglin Wang¹

¹ Institute of Automation, Chinese Academy of Sciences,
95 Zhongguancundonglu, Haidian, Beijing, China

² University of Chinese Academy of Sciences,
19 Yuquanlu, Shijingshan, Beijing, China

a) menghongyu2014@ia.ac.cn

Abstract: Just as Moores law dictates, technology scaling still continues in recent years. Thus increasing the number of on-chip cores continues to be the facto strategy to scale performance while interconnection networks remain important in multi-core chip systems. In this paper, we present a brand-new network-on-chip (NoC) based on full-mesh architecture - Full-Mesh NoC (FMN) that can provide high connectivity with low-radix routers. We give an example of implementation for a 64-node FMN system in 28 nm process. Results show that this FMN can achieve high frequency of 700 MHz while area requirement of each router is less than 0.035 mm². At last, we simulate a no-buffered FMN and a buffered FMN under uniform random traffic. Results show that the throughput of the no-buffered FMN can achieve the level of high-radix crossbar. The buffered FMN can achieve 2X throughput compared with the traditional buffered 2D-mesh NoC.

Keywords: network on-chip, full-mesh, high connectivity, multi-core

Classification: Integrated circuits

References

- [1] K. Goossens, *et al.*: “AEthereal network on chip: Concepts, architectures, and implementations,” IEEE Des. Test Comput. **22** (2005) 414 (DOI: [10.1109/MDT.2005.99](https://doi.org/10.1109/MDT.2005.99)).
- [2] T. Nguyen, *et al.*: “High-performance adaption of ARM processors into network-on-chip architectures,” IEEE International SOC Conference (2014) 222 (DOI: [10.1109/SOCC.2013.6749691](https://doi.org/10.1109/SOCC.2013.6749691)).
- [3] W. J. Dally and B. Towles: “Route packets, not wires: On-chip interconnection networks,” Proc. 38th Design Automation Conference (2001) 684 (DOI: [10.1109/DAC.2001.156225](https://doi.org/10.1109/DAC.2001.156225)).
- [4] B. Grot, *et al.*: “Express cube topologies for on-chip interconnects,” 2009 IEEE 15th International Symposium on High Performance Computer Architecture (2009) 163 (DOI: [10.1109/HPCA.2009.4798251](https://doi.org/10.1109/HPCA.2009.4798251)).
- [5] G. Chrysos: “Intel® Xeon Phi coprocessor (codename Knights Corner),” 2012 IEEE Hot Chips 24 Symposium (2012) 1 (DOI: [10.1109/HOTCHIPS.2012.7476487](https://doi.org/10.1109/HOTCHIPS.2012.7476487)).
- [6] B. Bowhill, *et al.*: “The Xeon® processor E5-2600 v3: A 22 nm 18-core product family,” IEEE J. Solid-State Circuits **51** (2016) 92 (DOI: [10.1109/JSSC.2016.2575000](https://doi.org/10.1109/JSSC.2016.2575000)).

- JSSC.2015.2472598).
- [7] A. Sodani, *et al.*: “Knights landing: Second-generation Intel Xeon Phi product,” IEEE Micro **36** (2016) 34 (DOI: [10.1109/MM.2016.25](https://doi.org/10.1109/MM.2016.25)).
 - [8] G. Passas, *et al.*: “A $128 \times 128 \times 24$ Gb/s crossbar interconnecting 128 tiles in a single hop and occupying 6% of their area,” 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip (2010) 87 (DOI: [10.1109/NOCS.2010.37](https://doi.org/10.1109/NOCS.2010.37)).
 - [9] C.-L. Wu and T.-Y. Feng: “On a class of multistage interconnection networks,” IEEE Trans. Comput. **C-29** (1980) 694 (DOI: [10.1109/TC.1980.1675651](https://doi.org/10.1109/TC.1980.1675651)).
 - [10] J. Balfour and W. J. Dally: “Design tradeoffs for tiled CMP on-chip networks,” ACM International Conference on Supercomputing 25th Anniversary Volume (2014) 390 (DOI: [10.1145/2591635.2667187](https://doi.org/10.1145/2591635.2667187)).
 - [11] J. Kim, *et al.*: “Flattened butterfly topology for on-chip networks,” 40th Annual IEEE/ACM International Symposium on Microarchitecture (2007) 172 (DOI: [10.1109/MICRO.2007.29](https://doi.org/10.1109/MICRO.2007.29)).
 - [12] G. Passas, *et al.*: “Crossbar NoCs are scalable beyond 100 nodes,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **31** (2012) 573 (DOI: [10.1109/TCAD.2011.2176730](https://doi.org/10.1109/TCAD.2011.2176730)).
 - [13] Y.-H. Kao, *et al.*: “Design of high-radix clos network-on-chip,” Fourth ACM/IEEE International Symposium on Networks-on-Chip (2010) 181 (DOI: [10.1109/NOCS.2010.27](https://doi.org/10.1109/NOCS.2010.27)).
 - [14] M. Lin and N. McKeown: “The throughput of a buffered crossbar switch,” IEEE Commun. Lett. **9** (2005) 465 (DOI: [10.1109/LCOMM.2005.1431173](https://doi.org/10.1109/LCOMM.2005.1431173)).

1 Introduction

As technology advances, Very Large Scale Integration (VLSI) and ultra-deep sub-micron manufacturing technologies have been improved continuously. Due that traditional integrated circuits are no longer applying to the needs of information technology, full-featured System on-Chip (SoC) appears. After decades of development in electronic systems and the emergence of diverse-compute intensive applications, SoC designers has started adding more and more general-purpose/application-specific Intellectual Property (IP) cores in designs [1]. The system design has changed from the computation centric design to the communication centric design [2]. The traditional shared bus has led to low efficient transfers due to their scalability issues in multi-core chips. The Network on-Chip (NoC) paradigm has been emerging as a promising solution to satisfy the communication needs for high performance designs [3]. The demand for system scalability, reusability and the decoupling between computation and communication have motivated the growth of NoCs in the recent years.

Most existing NoC architectures are based on rings or two-dimensional (2D) mesh as they have low design complexity and a good fit to planar silicon substrates [4]. While cores can always be added, their effectiveness is only as good as the on-chip network that binds them. The first-generation Intel Xeon Phi product family of which the code name is Knights Corner use a pair of rings to connect the cores, cache and memory controller [5] while the E5-2600 product family employ three pairs of rings to bring higher data transfer bandwidth [6]. Knights Landing (KNL) is the second-generation Intel Xeon Phi product family, which

introduces a new 2D, cache-coherent mesh interconnect [7]. Cores, vector processing units, level-2 (L2) cache, memory controllers, I/O controllers and other agents on the chip are connected by this interconnection.

The performance of interconnection networks is determined by throughput and latency while these two factors are mainly determined by blocking probability. A good routing algorithm or a good topology connectivity can make blocking probability decline greatly. Several studies have been conducted to investigate the connectivity merits of different topologies for NoCs. High-radix crossbar can be considered as one of the typical NoCs topologies which can provide excellent connectivity [8]. Multistage interconnection networks (MINs) are topologies driven by indirect networks in which high-radix crossbars are broken into several low-radix switches [9]. Though there are several stages between hosts which will lead to long network diameter, the frequency of switches can be improved very much as their radix and complex reduce. Concentrated mesh (C-mesh) is a typical orthogonal architecture and it can reduce the average latency by two-level interconnection [10]. Flattened butterfly topology [11] and Multi-drop Express Channels (MEC) topology [4] are two enhanced architectures based on C-mesh, and they both use high-radix routers to bring a further down to the network latency.

Network based on crossbar architecture can easily provide 100% throughput in total arrangement transactions while MIN can also achieve high bandwidth with excellent routing algorithms. However, planar silicon technologies are best matched to 2D networks. Previous 2D networks cannot offer good connectivity or achieve high throughput easily. In order to improve the performance of networks and make it fit the chip layout, we introduce Full-Mesh NoC (FMN) of which the architecture belongs to orthogonal topologies. Compared with other NoCs based on orthogonal topologies, the FMN can provide a high degree of connectivity as much as crossbar while the complex of FMN is much lower than high-radix crossbars. We implement a 64-node FMN system through designing hosts and routers at Register Transfer Level (RTL). After synthesis and Place and Route (P&R) in TSMC 28 nm process, we consider that FMN can operate at high frequency of 700 MHz with 0.035 mm² area overhead. At last, we give a simulation of a no-buffered FMN system and a buffered FMN system. The results show that the no-buffered FMN can provide the same throughput as the no-buffered high-radix crossbar. The buffered FMN can achieve 82% throughput which is 2X compared with the buffered 2D-mesh NoC.

Following the above discussion, the remainder of this paper is organized as follows. We briefly introduce related work in Section 2. In Section 3, we describe our topology architecture, and we give an implementation of the FMN in Section 4. Section 5 describes the evaluation and Section 6 concludes our work.

2 Related work

Crossbar networks are typically used for high-performance computing multiprocessors solutions and in the design for switches in networks (Fig. 1). Though the crossbar topology dose not scale well as system grows, there are still some researches working on how to place and route the logic cells, nets and pins in high-radix crossbars. In [12], the authors firstly compare three different floorplans

of the crossbar including side crossbar, distributed crossbar and centralized crossbar. Then they work on the design of data path. They compare several multiplexors architectures and two ways of how to place the data path. In addition, they also propose a novel scheduler microarchitecture that inverts the locality of wires by orthogonally interleaving the input arbiters with the output arbiters. With the help of high gate densities, abundant wiring resources and studious layout techniques, they conclude that it is possible to build high-radix crossbar. Though the high-radix crossbars have been implemented in silicon, it is still difficult to achieve high frequency or increase their channel width due to the quadratic requirements of wires.

Fig. 1. Crossbar network topology.

Different from the implementation of high-radix crossbar networks in direct connections, MINs are driven by several low-radix switches in indirect network. In [13], the authors propose a high-radix router architecture with Virtual Output Queue buffer structure and Packet Mode Dual Round-Robin Matching scheduling algorithm to achieve a high bandwidth Clos NoC as shown in Fig. 2. They synthesized an implementation of their network which contains 64 nodes in 3 stages. The switches in every stage are radix-8 and the width of every link is 36 bits. The results show that the delay of the crossbar and allocator can be limited to 0.556 ns and 0.931 ns. Though the topologies of MINs are preferable than high-radix crossbars, mapping of them including networks and hosts in the 2D surface of

Fig. 2. Clos network topologies.

the chip is still a big challenge. In addition, the routing algorithms can affect the throughput of MINs sometimes which cause the performance lower than high-radix crossbars.

Compared with crossbar networks of which the switches are distributed unevenly or centralized in floorplans, orthogonal networks consist of switches with homogeneous distribution. Thus the compact layouts provided by orthogonal networks can make them easy to be implemented at P&R phase. In C-mesh network [10], four hosts are co-located by a 4-way concentration at each network interface to reduce the number network nodes which can be seen in Fig. 3(a). Thus this architecture can provide smaller diameter compared to the traditional 2D-mesh network. They also augment the mesh with expressing channels along the perimeter of the network to restore the bisection channel count lost to the reduction in the radix of the mesh. In addition, they propose a C-meshX2 network which contains two independent C-mesh sub-networks to further improve the bandwidth while area utilization can also be improved. In [11], authors propose a flattened butterfly network based on the sub-networks of C-mesh architecture. Compared with the C-mesh network, this network introduces more channels to fully connect the sub-networks. The flattened butterfly network also makes a significant improvement to minimize the overall impact of router delay by reducing the maximum number of hops to two. However, the flattened butterfly itself is not truly scalable as the physical channel count in each dimension grows quadratically with the number of nodes in the dimension. In [4], the MEC network are introduced in order to improve the utilization of channels between the sub-networks based on the C-mesh as shown in Fig. 3(b). Compared with quadratic growth in routers of the flattened butterfly network, the radix of routers in the MEC networks grows linearly with the number of nodes in the dimension. In order to maintain the bandwidth, the MEC networks double the width of each channel between two sub-networks compared with flattened butterfly networks. In addition, the MEC networks can also get improvement in channel utilization and maintain the network diameter to 2. Though the above three orthogonal networks can provide low network diameter, it is difficult for them to provide high throughputs.

Fig. 3. Mesh-based topologies.

3 Full-mesh network on-chip architecture

3.1 Overview

FMN employs multiple unidirectional links to connect each pair of neighboring routers which are based on XY routing algorithm. Thus FMN can provide the same bisection bandwidth compared with crossbars or MINs. In addition, it can afford preferable connectivity at each direction than common 2D-mesh NoCs due to its well-designed topology. Fig. 4 shows an example of a 16-node FMN architecture. They key characteristics of FMN are as follows:

1. Bisection channel count per each row/column is equal to the network radix.
2. The west input port count of a router in each row is equal to its left node count and the east input port count is equal to its right node count.
3. The west output port count of a router in each row is equal to its right node count plus one. The east output port count of a router in each row is equal to its left node count plus one. It should be noted that the left routers do not have west ports and right routers do not have east ports.
4. The north input port count of a router in each column is equal to its lower node count plus one. The south input port count of a router in each column is equal to its upper node count plus one. It should be noted that the upper routers do not have north ports and the lower routers do not have south ports.
5. The north output port count of a router in each column is equal to its upper node count and the south input port count is equal to its lower node count.
6. Each router connects its local host using a pair of links like the common NoCs.

Fig. 4. A 16-node FMN.

The well-designed topology can make the FMN handle transactions as crossbar networks do. That means the FMN can handle conflict-free transactions as long as the destinations are different.

3.2 Router architecture

As the above, the highest radix (R) of routers in a k -node FMN can be calculated as Eq. (1).

$$R = 2 \times \sqrt{k} + 1. \quad (1)$$

As shown in Fig. 4, the highest radix of routers in 16-node FMN is 9. For a 64-node FMN, the highest radix is 17 which is very low compared with a 64-node crossbar network. However, this radix is very high compared with a 64-node MIN and much higher than 2D-mesh NoC. So we should take steps to optimize the router design by taking advantage of FMN topology and XY routing algorithm.

Due that the FMN is based on XY routing algorithm, the transaction is firstly delivered in a row and then in a column. If the source host and the destination host are in the same column, then the transaction will be only delivered in a column. So the input links in Y-scale of a router do not need to be connected to the output links in X-scale. As there are multiple links between each pair of routers, we define that a host can only use a specified link to transmit information. Then we number hosts from west to east in each row and from up to down in each column. We number links from west to east in each column and from north to south in each row. Thus a transaction can only be delivered in the row link which is numbered as its source host or in the column link which is numbered as its destination host. For example, the router of HOST5 as shown in Fig. 4 is numbered 2 both in the row and column. So HOST5 can only use the w2_o link and e2_o link to transmit information in this row as shown in Fig. 5. When transactions are in column and destination is HOST7, HOST5 can only use the n1_o link. When destination is HOST6 or HOST4, n2_o link or s4_o link can only be used. Though the number of ports on this router is high, there is no need to connect all its input ports to output ports. As a result, we find that this 9-radix router only needs four 5-radix decode modules, four 5-radix arbiter modules and four 5-radix AND-OR combinational modules. That is much lower than a common switch with the same radix. So this router can be designed in low latency which is mainly determined by the radix of decode modules, arbiter modules and AND-OR modules. It can also be designed in low area which is mainly determined by the number of these modules and links between ports. Compared with a 5-radix switch which usually consists of five 5-radix decode modules, five 5-radix arbiter modules, five 5-radix AND-OR modules, five input ports and five output ports, this 9-radix router has the same delay and almost same area requirements. In other words, we can consider that the router of HOST5 is a 5-radix router. Similarly, routers of HOST6, HOST9 and HOST10 in this 16-node FMN can also be considered as 5-radix routers. Thus the real highest radix of routers in a k -node FMN is calculated as Eq. (2).

$$R = \sqrt{k} + 1. \quad (2)$$

3.3 Analysis

Table I compares different 64-node network topologies using several metrics. As it can be seen, Clos and FMN present the maximum bisection bandwidth. Clos can be considered as the optimal topology because it can provides the highest bisection

Fig. 5. The router microarchitecture of HOST5.

bandwidth using a few and low-radix switches. However, Clos relies too much on the scheduling algorithms of allocators in chip systems to provide high throughput. In addition, the area requirements of the allocators and the links between multistage switches make these networks difficult to be mapped in the 2D surface of the chip. C-mesh and MEC can be implemented with greater ease than MINs while they can provide higher bandwidth and lower latency than 2D-mesh. However, they can only provide half bisection bandwidth of MINs and they also rely too much on the routing algorithms. FMN can not only be implemented easily, but also it can provide high bisection bandwidth and do not need complex routing algorithms. However, FMN use the most switches and the diameter of it is also the highest. As the Manhattan distance between the two farthest hosts are all the same, the network diameter is not the main reason causing the network delay. Not only that, the routers in FMN can operate independently with each other that means the complex of its routers is much simpler than others. Though the FMN looks like MEC in architecture, their connectivity is different. The focus of MEC is to reduce the diameter of the network while FMN is to improve the throughput. Compared with the complex algorithms of MINs, the XY routing algorithm used by FMN is much more simple which has been integrated in the decode modules and links of routers. Based on this, we think the 9-radix routers in FMN can be designed as concisely as the 8-radix switches in MINs.

Table I. Comparison of different 64-node topologies.

Topologies	Switches	Bisection bandwidth	Diameter	Switch radix
Clos	24	64	3	8
2D-mesh	64	16	15	5
C-meshX2	32	32	6	8
MECX2	32	32	2	10
FMN	64	64	15	9

4 Implementation flow

In this section, we implement a 64-node FMN in 28 nm process. Because FMN is based on XY routing algorithm, we place these hosts by column and then row. Thus the design of decode modules in routers can be simplified. In this paper, we assume that the host modules are all same except the address map while the hosts can be high-speed peripheral controllers, cores and memories in real chip systems. In addition, the most complex routers in this 64-node system are the middle four routers. So in this paper, we just implement one host and the 27-th router which locates at the fourth row and the fourth column.

4.1 Host implementation

Based on our existing APE core which stands for Algebraic Processing Engine, we give a simplified version named as Simple APE (SAPE) and use it as the master device in this host. It can achieve 1.2 GHz using AXI 4.0 interface in 28 nm process with 128 bits data width while area requirements is 4.5 mm². We integrate two memory banks which contains 8 SRAM blocks and the total storage capacity of these banks are 2 MB. Because our focus is the bandwidth of the interconnection network and the performance of routers, we should assume that the host has enough capacity and the chip is able to cover 64 hosts. Then the length of nets between routers can be long enough to verify the performance of the network. In addition, we design an adapter module which contains an issuing outstanding transaction module, an Asynchronous FIFO module and a converter module. We put this module in the host so that the area and critical timing paths of routers can be reduced. At the ports of the host, we add a register stage module to improve the timing. After P&R, the area requirements of the host is 9 mm² while the width and height are both 3 mm. The furthest port of the host is less than 1.5 mm from the bottom edge. The frequency of the host can easily achieve 800 MHz.

4.2 Router implementation

In order to improve the performance of routers, we have integrated some modules in the hosts, such as issuing outstanding transaction modules and buffer modules. We do not insert any pipeline registers in routers either. There are only three types of modules in routers. The decode modules are used for the source ports to generate request signals to their destination ports. As mentioned above, we only need to integrate eight 9-radix decode modules for the X-scale ports and hosts ports. Other ports can be directly to the logics of their destination ports. The arbiter modules which are based on Round-Robin strategy are used to handle these request signals and send grant signals to both source ports and destination ports. Similarly, we only integrate eight 9-radix arbiter modules for the Y-scale ports and hosts ports. At last, we use eight 9-radix AND-OR combinational modules to select the signals which can be send to the destination ports according to grant signals.

We use Design Compiler and IC Compiler to finish the implementation of this router. The layout of the router can be seen in Fig. 6. The results and others work are shown in the Table II. Though the Clos NoC is implemented in 65 nm process, the improvements of our routers performance are much greater than that brought by

the advanced technology. We also implement a set of register stage module which can be used to divide the long nets between two neighboring routers. The maximum setup of register stage 0.08 ns while the maximum delay is 0.25 ns. Thus we can calculate the total delay of the register stage add the router is less than 0.9 ns. We also conclude that the net delay is 100 ps/mm in 28 nm from previous experiments. Given that the handshake signals need to return to the sources, we consider that the signal nets delay between hosts and routers is less than 0.3 ns. As the routers are directly connected to their hosts without register stages and there are only one set of register stage between two neighboring routers, we calculate that the maximum delay between two register stages is 1.2 ns. Assuming that the clock uncertainty is 0.2 ns, our FMN can achieve a frequency of 700 MHz.

Fig. 6. Layout of the router.

Table II. Results of different switches.

Parameters	Clos NoC after synthesis	FMN after synthesis	FMN after P&R
Decode delay	N/A	0.05 ns	0.07 ns
Arbiter delay	N/A	0.15 ns	0.15 ns
AND-OR delay	N/A	0.15 ns	0.25 ns
Total delay	1.4872 ns	0.35 ns	0.57 ns
Area	0.04 mm ²	0.016 mm ²	0.035 mm ²

5 Evaluation

In this section, we present simulation-based performance evaluation for two 64-node FMN systems. One is not buffered and the other is buffered. In the no-buffered FMN system, there are only no more than 144 registers in each routers. In the buffered FMN system, we integrate one FIFO module at each path from decode modules to arbiter modules. Due that other researchers set the length of the packet

to eight in previous works, we use this figure to simulate our networks and the depth is of the FIFO modules is also set to eight. For example, the 27-th router is integrated nine 8-depth FIFO modules after each decode module. That means there are 72 8-depth FIFO modules altogether in this router. Fig. 7 shows the delay performance results of different NoCs under uniform traffic. The transmission delay in networks based on C-mesh and MEC is lower than others under low injection rate. However, as the injection rate increases, their latency get much higher than others. At last the C-mesh network can achieve about 38% throughput and the MEC network can only achieve about 29% throughput. Though the delay in the no-buffered FMN is higher than others under low injection rate, it can get higher throughput than networks based on C-mesh, MEC and 2D-mesh. Like the no-buffered crossbar-based network, the no-buffered FMN can achieve 58% throughput which is roughly consistent with the theory calculation in [14]. Similarly, the buffered FMN can achieve 82% throughput which is also approximate to the calculation value. Thus we consider that the buffered FMN can get 2X performance compared with buffered 2D-mesh NoC.

Fig. 7. Latency vs. injection rate for different NoCs.

However, in real systems especially in the parallel processing systems, hosts are usually assigned to access different destinations at the same time. So we only need to integrate FIFOs in host modules to buffer the congestion which occurs in rare cases. In addition, for the master devices which do not support for out-of-order transaction, it is difficult to get lower latency with buffered routers or switches.

6 Conclusion

In this paper, we present a new kind of Network on-Chip, Full-Mesh NoC. This network can provide high connectivity using low-radix routers. Compared with high-radix crossbar networks, FMN can get the same throughput while the radix of routers is much lower. We also prove that both the area and delay of routers in FMN are modest after implementing in 28 nm process. In the end, we give a latency evaluation experiment of the FMN system. Results show that it can indeed achieve high throughput.

However, we do not implement the buffered FMN as our intension is to handle balanced transmission and we consider that the no-buffered FMN is able to handle it. For the conflicting transmission, it is difficult for the no-buffered FMN to get high throughput. In addition, the FMN is not suitable for many-core systems as the area and delay of routers in FMN will be very large under high radix. For chip systems containing 100 hosts or less, we consider that FMN is the best choice for interconnection networks.