

A comprehensive comparison between virtual cut-through and wormhole routers for cache coherent Network on-Chips

Peng Wang, Sheng Ma^{a)}, Hongyi Lu, and Zhiying Wang

State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, 410000, China

a) masheng@nudt.edu.cn

Abstract: A basic design aspect of cache coherent Networks-on-Chip (NoCs) is the flow control mechanism. Since the minimum buffer size of virtual cut-through (VCT) switching is larger than that of the wormhole one, the VCT switching is traditionally regarded as an inefficient NoC flow control type. Yet, the scaling of semiconductor technology shrinks the transistor size, and reduces the criticality of buffer amount for NoC designs; the VCT switching may become a promising NoC flow control candidate. This paper performs a comprehensive comparison between the VCT and wormhole switching. Based on detailed RTL-level implementations, we evaluate the hardware costs with both deterministic and adaptive routing. Compared with the wormhole switching, the VCT one shortens the critical path by up to 27%, and induces less area and power overheads. Furthermore, the allocator in VCT routers exhibits a better scalability in area overheads. Thus, the VCT router is an efficient NoC flow control type.

Keywords: wormhole, virtual cut-through, cache coherent NoCs

Classification: Integrated circuits

References

- [1] S. Ma, Z. Wang, N. Enright Jerger, L. Shen and N. Xiao: IEEE Trans. Parallel Distributed Syst. (2013) PrePrints(99). DOI:10.1109/TPDS.2013.166
- [2] S. Ma, Z. Wang, Z. Liu and N. Enright Jerger: IEEE Trans. Comput. (2013) PrePrints(99). DOI:10.1109/TC.2013.2295523
- [3] W. J. Dally: IEEE Trans. Parallel Distributed Syst. **3** (1992) 194. DOI:10.1109/71.127260
- [4] L.-S. Peh and W. J. Dally: The Seventh International Symposium on High-Performance Computer Architecture, 2001. HPCA (2001) 255. DOI:10.1109/HPCA.2001.903268
- [5] W. J. Dally and B. P. Towles: *Principles and Practices of Interconnection Networks* (Elsevier, 2004).
- [6] P. Kermani and L. Kleinrock: Computer Networks (1976) **3** (1979) 267. DOI:10.1016/0376-5075(79)90032-1

- [7] J. Duato: IEEE Trans. Parallel Distributed Syst. **7** (1996) 841. DOI:10.1109/71.532115
- [8] H. Wang, L.-S. Peh and S. Malik: Proc. of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (2003) 105. DOI:10.1109/MICRO.2003.1253187
- [9] X. Chen and L.-S. Peh: Proc. of the 2003 International Symposium on Low Power Electronics and Design (2003) 90. DOI:10.1109/LPE.2003.1231841
- [10] A. Roca, J. Flich, F. Silla and J. Duato: Proc. of the International Conference on High-Performance Computing (HiPC) (2010) 1. DOI:10.1109/HIPC.2010.5713170
- [11] R. Das, S. Narayanasamy, S. K. Satpathy and R. G. Dreslinski: Proc. of the 40th Annual International Symposium on Computer Architecture (2013) 320. DOI:10.1145/2485922.2485950
- [12] D. U. Becker and W. J. Dally: Proc. of the Conference on High Performance Computing Networking, Storage and Analysis (2009) 1. DOI:10.1145/1654059.1654112
- [13] M. S. Abdelfattah and V. Betz: FPT (2012) 95. DOI:10.1109/FPT.2012.6412118

1 Introduction

As the functional unit count increases on a single chip, the coherence and communication substrate are going to become the bottleneck to improve the overall processor performance. Optimizing Networks-on-Chip (NoCs) based on the characteristics of coherence traffic is an effective way to improve the efficiency of many-core coherence protocols [1, 2]. The router microarchitecture and the flow control play important roles for the NoC performance, and they directly and largely decide the power consumption, latency and area overhead of the entire NoC.

The wormhole and virtual cut-through (VCT) are the two main switching types. The wormhole switching allocates buffers and channel bandwidths on the flit granularity. When a packet is blocked, routers only need to store a few flits, instead of an entire packet. Thus, a small amount of buffers satisfies the switching requirement. With the deployment of multiple virtual channels (VCs) [3], the wormhole switching can mitigate the network blocking and improve the channel utilization. But VCs adds router complexities and introduces additional delays [4]. Moreover, since the wire delay between neighboring routers is continually increasing, the wormhole switching must configure more buffers to cover the credit round-trip delay [5]. All these factors reduce the advantage of wormhole switching in the NoC environment.

In contrast, the virtual cut-through (VCT) switching [6] allocates buffers and channel bandwidths on the packet granularity. The VCT switching allows packets to apply for routing as soon as the head flit arrives, without waiting for all flits. When the requested channel is busy, VCT routers have to provide a large storage capacity to buffer one or more packets. The area and power overheads of large buffers lead previous NoC designers to prefer the wormhole switching. However, as the CMOS feature size decreases, more

transistors are available and the capacity of buffers may no longer be the most critical factor for NoC routers. Furthermore, previous work shows that in off-chip networks with a slightly more storage capacity, the VCT switching can achieve significantly higher throughput than the wormhole one at the same frequency [7]. Similar trends can be observed for on-chip networks as well. This may makes the VCT switching be an appropriate NoC flow control candidate. Thus, it is meaningful and imperative to compare the wormhole and VCT switching for NoC routers with the consideration of emerging technology trends.

In this paper, we explore the design space of NoC routers for the wormhole and VCT switching, based on detailed RTL-level implementations. For fair comparison, the wormhole and VCT routers implement the same arbiter type and crossbar structure. In addition to applying the deterministic routing algorithm, we also implement an adaptive VCT router and an adaptive wormhole router for thorough comparison. We use the Synopsys design compile (DC) to synthesize our designs. At the same environment, the VCT router consumes less area and power overheads than the wormhole router with both the deterministic and the adaptive routing. With simpler allocator structures, the VCT router reduces the critical path delay by up to 27% compared with the wormhole router. Moreover, the allocator structure in VCT router exhibits a better scalability in the area overhead, which is helpful to implement high-radix routers.

The remainder of the paper is organized as follows. Section 2 discusses the research motivation. Section 3 presents the implementation of wormhole router. Section 4 describes the VCT router. section 5 introduces the adaptive routing rule. Section 6 gives the experimental results, including the area overhead, critical path delay and power consumption. Finally, Section 7 concludes the paper.

2 Motivation

Although the NoCs or on-chip networks originate from the off-chip multi-processor interconnection networks, the design targets and constraints of NoCs are significantly different with these of the off-chip networks. For example, the area and power overheads of the router buffer are primary design constraints for NoCs. The buffer's dynamic power consumption can be as high as 50% of the dynamic power consumption of a NoC router [8], and its leakage power consumption also takes about 64% for the whole router [9]. Besides, larger buffers induce high area overheads. Even though the larger buffers can support a higher performance, its significant area and power overheads make previous NoC designers prefer to wormhole switching.

However, several factors cause the virtual cut-through (VCT) switching to become a promising NoC flow control type. First, the advanced semiconductor technology can integrate more transistors to offer more buffers to NoC routers. This reduces the criticality of the VCT switching's large buffer size. The development of off-chip networks has already experienced similar proce-

dures [5]. In the very first, the off-chip router buffer size was a concern. As more transistors are integrated into the off-chip routers, the buffer size quickly was not a constraint any more and, therefore, the VCT switching was easily and widely adopted in off-chip networks [10]. Second, the wide on-chip channel size reduces the largest flit count of the NoC packet [1, 2], which shortens the difference between the wormhole and VCT switching, and improves the applicability of VCT flow control for the NoCs. Third, sophisticated low-power designs [11] allow the NoC to utilize the buffers more efficiently, rather than limiting the buffer capacity. Previous work shows that the VCT switching can achieve significantly higher performance than the wormhole one for off-chip networks [7]. Yet, whether the VCT switching is superior to the wormhole one for on-chip network is still an open issue. It is meaningful to perform thorough comparisons between these two flow control types for NoCs, with the consideration of different routing schemes.

3 Wormhole router

As an efficient switching technology, the wormhole flow control is widely applied in NoCs. Fig. 1 shows components in a typical wormhole router. The input port (IP) is responsible for buffering the input data and performing the routing computation. The VC allocator (VA) and switch allocator (SA) allocate VCs and channels. The output port (OP) keeps the flow control information and monitors the status of downstream routers to avoid the buffer overflow. In our design, the wormhole router adopts the virtual channel technology, the lookahead routing and speculative switch allocation. Besides, the X-Y routing algorithm keeps the networks deadlock free. The below is the detailed implementation.

3.1 Input port unit

Fig. 2 shows the IP structure. As we adopt lookahead routing, the head flit carries the routing information, which is computed by the upstream router. When a head flit of a packet has arrived at the router, the input port encodes the routing information and allocates a input VC to the packet. The input port will keep the routing information to guide the rest flits into the input VC, until the tail flit has arrived.

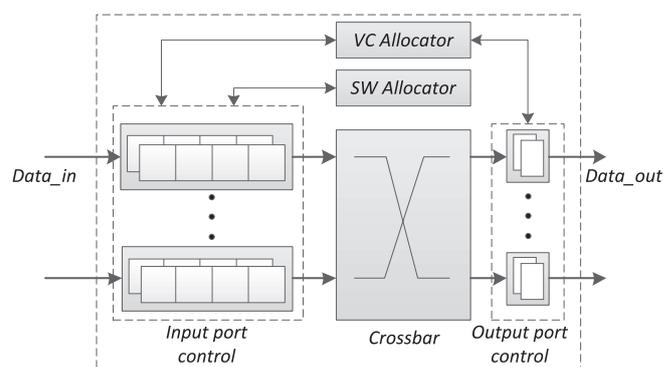


Fig. 1. The wormhole router.

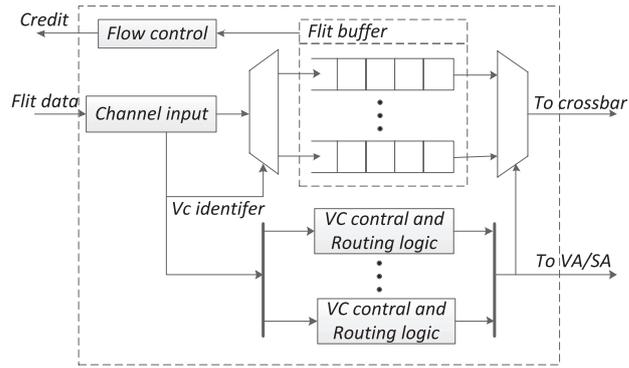


Fig. 2. Input port.

The routing logic computes the routing information for the downstream router, based on the current router address and destination address. Even though one routing unit can meet with our requirement, we design an individual routing unit for each VC. In this way, we can simplify control units, with a little impact on the area consumption. The flow control unit monitors the VC status. When a flit leaves the router, the flow control unit sends out a credit to the upstream router.

3.2 Virtual channel allocator and switch allocator

3.2.1 Virtual channel allocator

The virtual channel allocator matches $P \times V$ requests from input VCs with $P \times V$ resources from output VCs [12]. As shown in the Fig. 3, we adopt the separable input first structure. In order to guarantee that each input VC only requests one output VC in a time, we set $P \times V$ arbiters for input VCs [13]. First, the input arbiter selects a request for each input VC. Next, we group the requests for the same output VC together. At last, each output VC arbiter selects a request from the input VC arbiter results. In this way, one output VC is granted to a unique input VC.

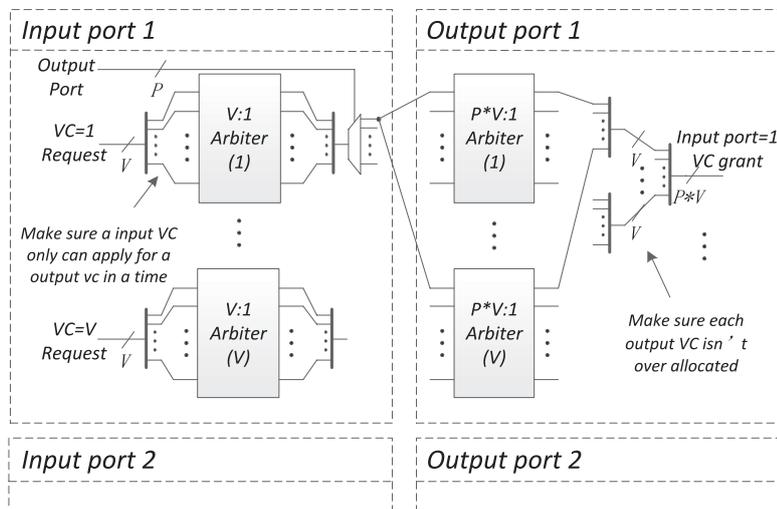


Fig. 3. Virtual allocator.

3.2.2 Switch allocator

The switch allocator (SA) matches the output ports (physical channels) with input VCs, according to the principle that each output port is granted to at most one input VC, and each input port is allocated to at most one output port. Like the VA structure, as shown in Fig. 4, the SA adopts the separable input first structure as well. First, each IP arbiter selects a VC request from the input ports. Next, we group the results applying for the same output port together. At last, each OP arbiter selects a request from the IP arbiter results. So each output port is granted to at most one unique input VC.

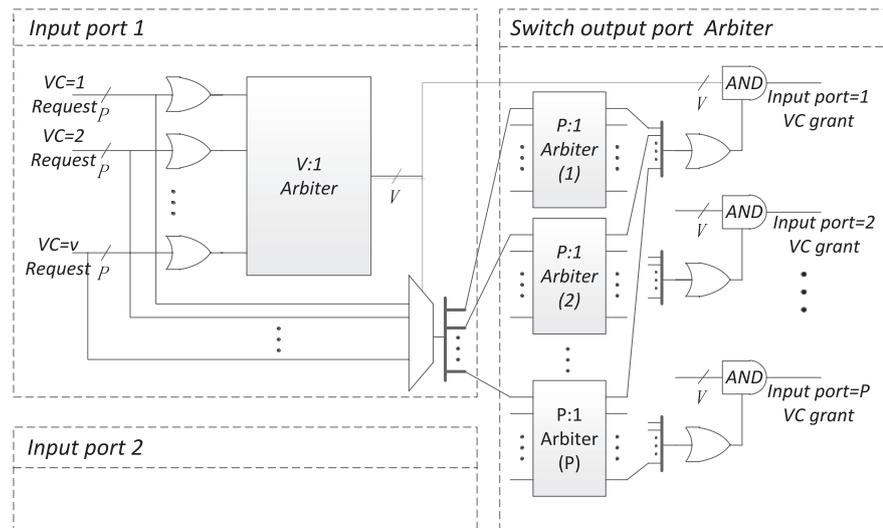


Fig. 4. Switch allocation

In order to achieve the same pipeline depth as the VCT router, we use the speculative switch allocation. We have designed two switch allocators. One uses speculative requests and the other one uses non-speculative requests. We give non-speculative requests a higher priority. When there are conflicts between two allocations, non-speculative requests win the conflict.

3.3 Output port

The output port mainly monitors the buffer capacity in the downstream router, based on credits sending between neighbor routers. In the wormhole router, a credit indicates a flit buffer. When a flit leaves the current router, the credit in the output port decreases, meaning that the buffer capacity in the downstream router decreases. At the same time, a credit is sent to the upstream router to indicate that the buffer in current router is released. In order to avoid the channel resource waste caused by the speculative switch allocation (a flit has been granted a channel, but the input VC in the the downstream router has no a free buffer), we need to check the buffer status when performing the speculative switch allocation.

4 VCT router

4.1 Overall structure

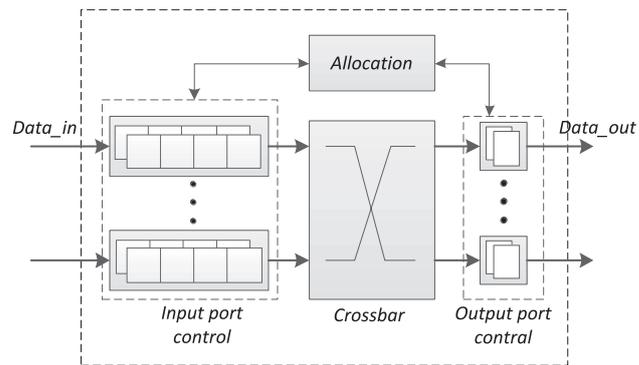


Fig. 5. The VCT router.

As shown in Fig. 5, the VCT router is similar to the wormhole one, except for that the VCT router allocates channels and buffers at the packet granularity. In addition, the VCT router has simpler structure due to its simple allocators. In the VCT router, we also adopt the X-Y routing to avoid deadlock and the lookahead routing to reduce pipeline depths.

4.1.1 Input port

In the VCT router, a packet is broken into constant-sized flits. Flits from the same packet must be delivered continuously and cannot be interrupted by other packets. Besides, a credit releases a packet-size buffer, instead of a flit-size buffer. When a head flit begins to transfer, the current router sends out a credit to inform the upstream router that a packet buffer has been vacated. Furthermore, the packet applies for routing as soon as the head flit arrives, without waiting for the rest flits. So, the VCT router can also efficiently utilize buffer resources to achieve a high throughput.

4.1.2 Allocator

Unlike the wormhole router, both the VA and SA in the VCT router take the packet as the smallest unit to arbitrate. As shown in Fig. 6, the allocator fairly selects a input VC for each output port in every cycle. But the packet transmission in the VCT router must be continuous and cannot be interrupted. Therefore, we need to generate a hold signal to select the SA result between the previous arbitration results and new arbitration results. In our design, the *Hold* signal is asserted when an output VC is allocated to the input VC, and it is de-asserted when the tail flit leaves the input VC.

As we allocate the output virtual channel and output channel at the same time, we need to ensure that the allocated output VC has enough available buffers and has not been allocated to other input VCs. Also, the allocated input port and output port are not occupied by other packets. The *Elig* signal is used to mark these available output VCs.

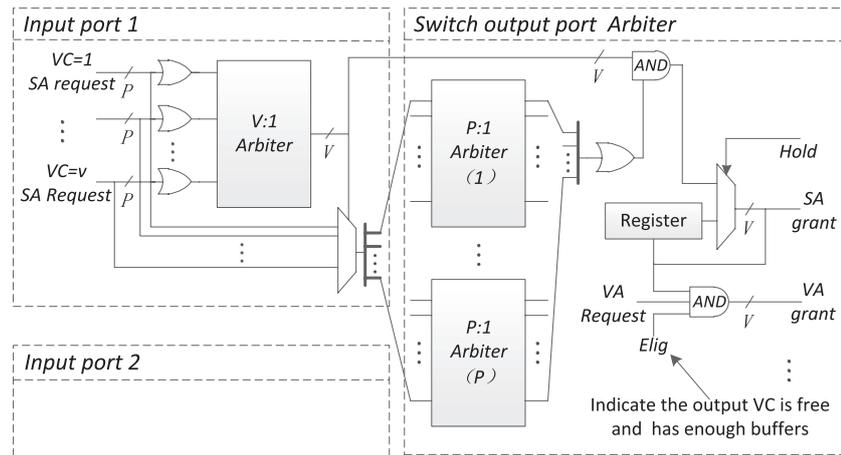


Fig. 6. The allocator. *Elig* output VC indicates which output VCs can be allocated. *Hold* indicates whether keep the switch allocation result.

4.1.3 Output port

Since each packet is broken down into the same number of flits (according to the maximum length packet), each credit represents a fixed length buffer capacity. When a head flit leaves the current router, the credits decrease. At the same way, credit is sent to upstream routers when a head flit leaves the current router, while the remaining flits will not impact on the number of credits. This is critical for the VCT switching.

5 Adaptive routing

In our adaptive routers, each node allocates output ports, based on the current router status and minimum transmission path. In the escape path, we use the X-Y deterministic routing to avoid deadlock. In order to keep the router structure simple, sending packets from the escape path to adaptive VC is not allowed, but sending packet from adaptive VCs to the escape path is allowed. In the adaptive VC, we give the highest priority in the X direction output port. When the X direction output port has no free buffers, we send packets to the Y direction output port. When there is no free buffers in both X and Y directions, we send the packet to the escape path.

6 Results

Our evaluation of router architectures is based on detailed RTL-level implementations. We take a two dimensional 8×8 mesh topology, with a physical network, and multiple virtual resource classes (RC). Each virtual resource class contains multiple message classes (MC). In our design, each message has a single virtual channel, so the number of VC equals to $MC \times RC$. Routers have the same input port count ($IP = 5$) and output port count ($OP = 5$). We use round robin arbiters for switch allocator and VC allocator. We take the same crossbar structure (5×5) in all of routers to make a fair comparison. The designs are synthesized using the Synopsys Design Compiler with the 45 nm NanGate open cell library.

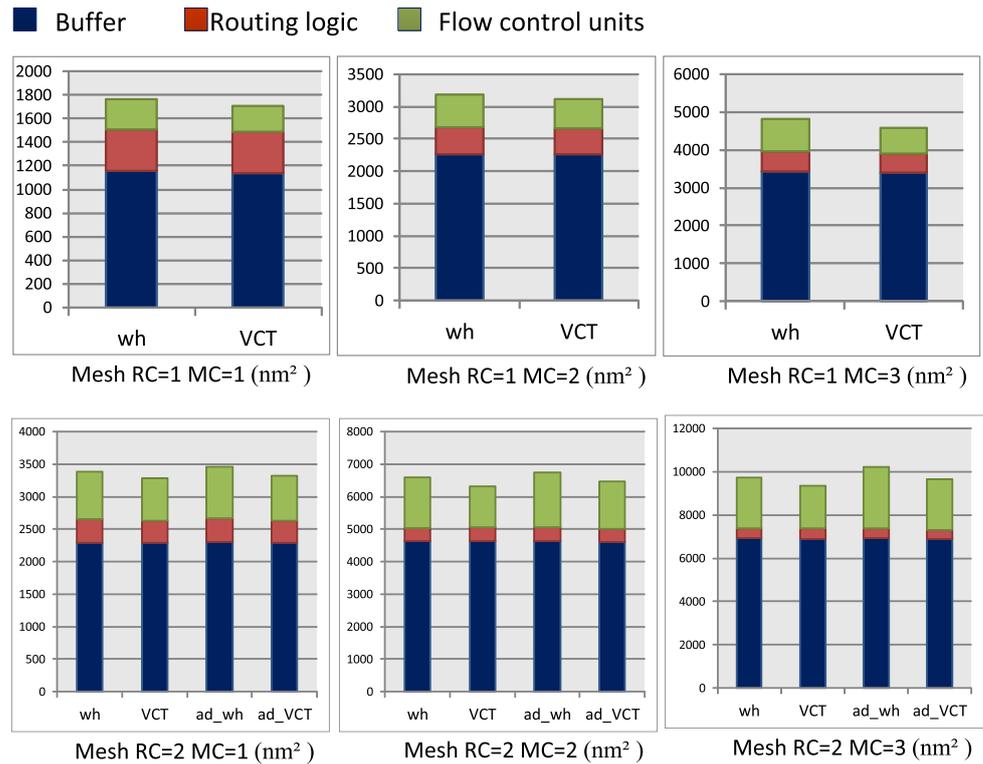


Fig. 7. Area of input port. Each VC buffer depth is 4 flits, and the flit width is 32 bits.

6.1 Area

6.1.1 Area of input port

Fig. 7 (“wh” stands for wormhole and “ad” stands for adaptive) shows the area for input port. Since the VCT router’s control mechanism is simpler, its VC control and the flow control area is about 11.9% and 2.2% (MC = 1 and RC = 1) less than that of the wormhole router. On the one side, the wormhole router needs to monitor the status of each flit buffer, while the VCT router only needs to monitor the packet status. On the other side, it is unnecessary for VCT routers to distinguish the speculative requests with non-speculative requests. The VCT router allocate virtual channels and switching at the same time. The VC control area in the adaptive VCT router has increased by 17.72% (RC = 2 and MC = 2) compare with the VCT router. But it consumes 13.2% less area than the adaptive wormhole router.

6.1.2 Area of allocator

Fig. 8 shows the allocator area with When each input port has only one input VC, the SA and the VA allocate the same number of resources to the same number of requests. But the SA consumers 81.47% more area than the VA. It is due to that the speculate SA leverage each allocators for speculative requests and non-speculative requests, respectively.

With VCs increase, the area of the VA increases rapidly but the area of the SA increases slowly. Even though the VA and the SA have the similar structures, the VA grants output VCs to input VCs, so it needs $P \times V$ arbiters

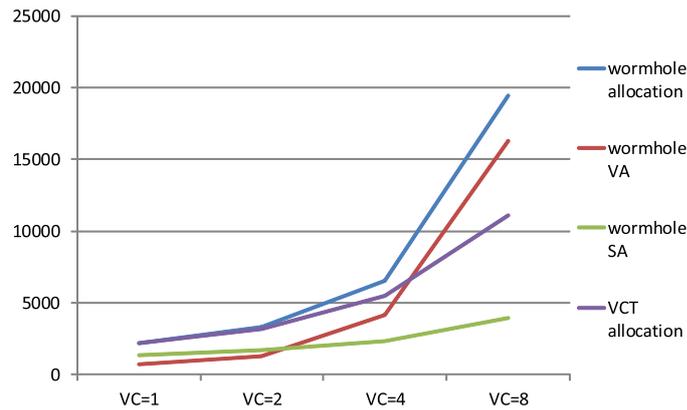


Fig. 8. Area of allocator.

to guarantee that each input VC is allocated at most one output VC and need $P \times V$ arbiters for output VCs to prevent the output VC from being over allocated. The SA allocates output ports to input VCs, so it need $2 \times P$ arbiters to select input VC requests and needs $2 \times P$ arbiters for output ports to select input VCs (considering speculative allocation). Therefore, the area increase in the VA is mainly caused by the number of arbiters, and the area increase in the SA is caused by the number of requests.

The allocator area (including VA and SA) in VCT router is 2.36%, 5.3%, 15.31%, 43.07% ($VC = 1, VC = 2, VC = 4, VC = 8$) less than that of the wormhole router. Thus, the allocators in the VCT router displays a better scalability in the area, which can support the allocation of more VCs.

6.1.3 Area of output port

As shown in Fig. 9, the credit count unit in the VCT router only consumes 20% area of the credit count unit in the wormhole router. A credit in the VCT router represents a packet (multi-flits) storage capacity, while a credit in the wormhole router represents a flit storage capacity. With the same amount of buffers, the credit count in the wormhole router requires more bits wide and larger number of registers to record credit information.

6.2 Critical path delay and power consumption

As shown in Table I, the critical path in the VCT router is 24.31%~27.08% shorter than the wormhole router. The critical path in wormhole router begins at the request signal from the input VC control, then through the VA for allocating an output VC, at last back to the input VC control unit. The critical path in the VCT router is reduced in two aspects. First, the input VC unit in the VCT router does not have to distinguish the non-speculative or speculative requests. In contrast, the wormhole router has to produce the speculative request and make sure the requested output VC is free, which induce some delay on the critical path.

Second, comparing Fig. 3 with Fig. 6, the VA in wormhole router adopts $P \times V:1$ arbiters for output VC allocation, while the VCT router adopts $P:1$ arbiters for output port allocation. More requests induce more delay in the

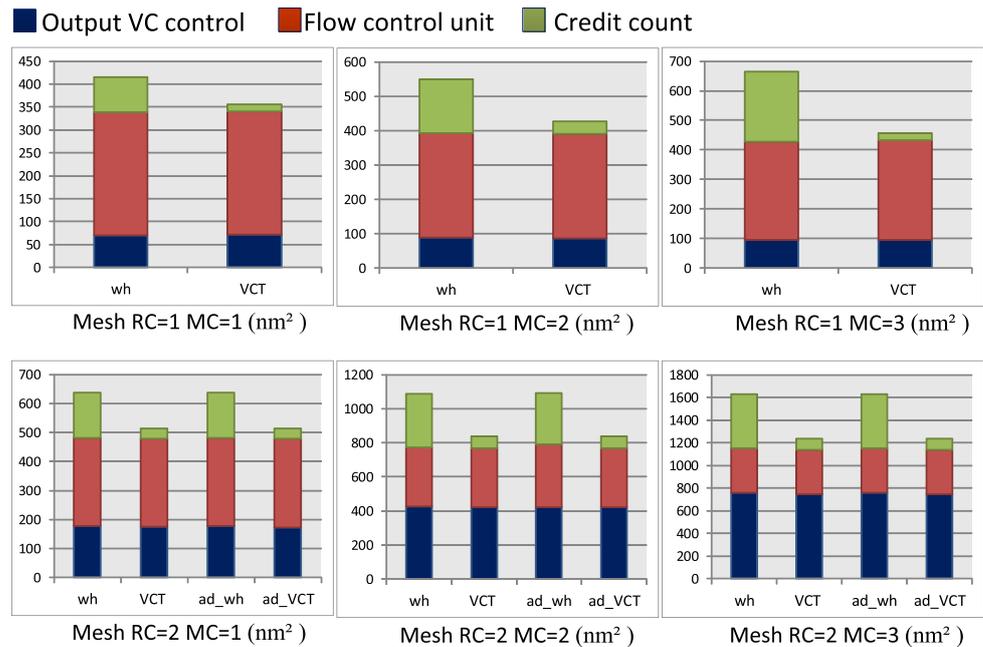


Fig. 9. Area of output port. Each VC buffer depth is 4, and the bit width is 32 bits.

Table I. The critical path delay.

num vcs	wormholens	VCTns	adaptive wormholens	adaptive VCTns
1	1.44	1.05		
2	1.59	1.15	2.05	1.6
4	1.81	1.37	2.33	1.82

Table II. The power consumption (The number of VC is 4vc buffer depth is 4, and bitwidth is 32 bits, frequency is 500 Mhz)

	wormhole	VCT	adaptive wormhole	adaptive VCT
Cell Internal Power (mW)	12.3199	11.6584	12.2049	11.5895
Net Switching Power (mW)	1.4969	1.6601	1.3023	1.5824
Total Dynamic Power (mW)	13.8169	13.3185	13.5072	13.1720
Cell Leakage Power (mW)	0.8880	0.7792	0.9310	0.9067

arbiters. The shorter critical path can support more complex designs, which is critical to realize more efficient flow control and routing algorithms.

As shown in Table II, with both the deterministic and adaptive routing algorithm, the VCT router achieves lower total dynamic power consumption than the wormhole one at the same frequency. Besides, with the same amount buffers, the VCT router consumes lower leakage power.

7 Conclusion

In this paper, we have explored the design space of the wormhole router and the VCT router. Based on detailed RTL-level implementations, we estimate their critical path delays, area and power consumption. With the same allocator structure, the VCT router achieves a 27% shorter critical path than the wormhole router. This shorter critical path offers more design spaces to realize efficient flow control mechanisms and routing algorithms. Furthermore, the allocator in the VCT router also displays a better area scalability with the increase of VC counts; this can support the efficient allocation of more VCs. Finally, with the same amount buffers, the VCT router consumes lower power and its input and output port units consume smaller areas. These comparisons illustrate that the VCT router is an efficient and promising router candidate for NoC designs.

Acknowledgment

This work is supported by 863 Program of China (2012AA010905), NSFC (61272144, 61303065), Research Project of NUDT (JC13-06-02), Doctoral Fund of Ministry of Education (20134307120028), and Hunan Provincial Natural Science Foundation (14JJ3002).