Microprocessors and Microsystems 35 (2011) 98–109

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

A practical low-latency router architecture with wing channel for on-chip network

Mingche Lai, Lei Gao *, Sheng Ma, Xiao Nong, Zhiying Wang

Department of Computer, National University of Defense Technology, Changsha, Hunan, China

ARTICLE INFO

Article history:
Available online 24 September 2010

Keywords:
On-chip Network
Router
VLSI design

ABSTRACT

With increasing number of cores, the communication latency of Network-on-Chip becomes a dominant problem due to complex operations per node. In this paper, we try to reduce communication latency by proposing single-cycle router architecture with wing channel, which forwards the incoming packets to free ports immediately with the inspection of switch allocation results. Also, the incoming packets granted with wing channel can fill in the time-slots of crossbar switch and reduce the contentions with subsequent ones, thereby pushing throughput effectively. We design the proposed router using 65 nm CMOS process, and the results show that it supports different routing schemes and outperforms express virtual channel, prediction and Kumar's single-cycle ones in terms of latency and throughput. When compared to the speculative router, it provides 45.7% latency reduction and 14.0% throughput improvement. Moreover, we show that the proposed design incurs a modest area overhead of 8.1% but the power consumption is saved by 7.8% due to less arbitration activities.

1. Introduction

With the continual shift advancement of device technology towards nanometer region, it will be allowed to introduce thousands of cores on a single chip in the future. There is a wide consensus both in industry and academia, the many cores are the only efficient way for utilizing the billions of transistors and represent the trend of future processor architecture. More recently, several commercial or prototype many-core chips, such as TeraScale [1], Tile [2] and Kilocore [3], have been delivered. To connect such many cores, the traditional bus-based or crossbar structure has been seen more and more incapable of meeting the challenges of intolerant wire delays or poor scalability in deep submicron conditions. Network-on-Chip as an effect way of on-chip communication has introduced a packet-switched fabric that can address the challenges of the increasing interconnection complexity [4].

Although NoC provides a preferable solution towards the long wire delay compared to traditional structures, the communication latency still becomes a dominant problem with the increasing number of cores. For example, the average communication latencies of 80-core TeraScale and 64-core Tile are close to 41 and 31 cycles, since their packets forwarded via many cores must perform complex operations at each node through 5-stage or 4-stage routers. In this way, the communication latency tending to be larger with increasing number of cores will become the bottleneck of application performance on the future many cores.

There have been significant works to reduce communication latencies of NoCs in various approaches such as designing new topologies and developing fast routers. Bourduas et al. [5] has combined mesh and hierarchical rings to form a hybrid topology to provide fewer transfer cycles. In theory, the architects prefer to adopt many high-radix networks to further reduce average hop counts; however, for the complex structures such as flattened butterfly [6], finding the efficient wiring layout during the backend design flows is a challenge on its own right. Recently, many aggressive router architectures with single-cycle transfers have also been developed. Kumar et al. [7] proposes the express virtual channel (EVC) to reduce the communication latency by bypassing intermediate routers in a completely non-speculative fashion. This method is efficient to close the gap between speculative router and ideal ones; however, it does not work well at some non-intermediate node and only suits for the deterministic routing. Moreover, it sends a starvation token upstream every fixed n cycles to stop the EVC flits and prevent the normal flits of high-load node being starved. And it results that many packets at the EVC source node have to be forwarded via normal virtual channel (NVC), thereby increasing average latencies. In [8,9], another predictive switching scheme is proposed, where the incoming packets are transferred without waiting the routing computation and switch allocation if the prediction hits. Matsutani et al. [10] analyzes the prediction rates of six algorithms and finds that the average hit rate of the best one only achieves 70% under different traffic patterns. It means that many packets still require at least three cycles to go

* Corresponding author. Tel.: +86 13787314163.
E-mail addresses: mingchelai@nudt.edu.cn (M. Lai), angela_nudt@yahoo.com (L. Gao).

0141-9331/$ - see front matter Crown Copyright © 2010 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2010.09.002

through a router when prediction misses or packet conflicts. Then, Kumar et al. [11] presents a single-cycle router pipeline which uses advanced bundles to remove control setup overhead. However, the design in [11] only works well at low traffic rate because it emphasizes no flit exists in the input buffer when the advanced bundle arrives. At last, the preferred path [12] is also pre-specified to offer the ideal latency, but it can not adapt to the different network environments.

Besides the single-cycle transfer exhibited by all the techniques listed above, we additionally emphasize on three aspects in the proposed low-latency router. Firstly, a preferred technique that accelerates a specific traffic pattern should also work well for other patterns and it would be best to suit for different routing schemes involving deterministic and adaptive ones. Secondly, besides for zero load latency, the high throughput and low latencies under different loads are also important since the traffic rate is easily changed on a NoC. At last, some complex hardware mechanisms should be avoided to realize the cost-efficiency of our design, such as prediction, speculation, retransmission and abort detection logics. In this paper, the main contribution is that we propose a low-latency router architecture with wing channel (Section 2). Regardless of what the traffic rate is, the proposed router inspects the switch allocation results, and then selects some new packets without port conflict to enter the wing channel and fill the time-slots of crossbar ports, thereby bypassing the complex 2-stage allocations and directly forwarding the incoming packets downstream in the next cycle. Here, no matter what traffic pattern or routing scheme is, once there is no port conflict with others, the new packet at current router can be delivered within one cycle, which is the optimal case in our opinions. Moreover, as the packets of wing channel make full use of the time-slots of crossbar and reduce contentions with subsequent ones, the network throughput is also pushed effectively. We then modify the traditional router with little additional cost constraint, and present the detailed micro-architecture and circuit schematics of proposed one in Section 3. Section 4 estimates the timing and power consumption via commercial tools and evaluates the network performance using a cycle-accurate simulator considering different routing schemes under various traffic rates or patterns. Our results indicate that the proposed router outperforms EVC, prediction and Kumar's single-cycle ones in terms of latency and throughput metrics and then provides 45.7% latency reduction and 14.0% throughput improvement averagely when compared with state-of-the-art speculative one. The evaluation results of proposed router also show that although router area is increased by 8.1%, the average power consumption is saved by 7.8% due to less arbitration activities in low rates. Finally, Section 5 concludes the paper.

2. Proposed router design

In this section, the single-cycle router architecture supporting different routing schemes is proposed, where the incoming packets forwarded to the free ports are selected for the immediate transfers at wing channel without waiting their VA and SA operations based on the inspection of switch allocation. Hence, it can reduce communication latency and improve network throughput under various network environments. Through the analysis of the original router, Section 2.1 first presents the single-cycle router architecture and describes its pipeline structure, and then we explain details of wing channel in Section 2.2.

2.1. Proposed router architecture

It is well known that the wormhole flow control is first introduced to improve performance through fine-granularity buffer at flit level. Here, the router with single channel, playing a crucial role in architecting the cost-efficient on-chip interconnects, always supports the low latency due to its little hardware complexity, but is prone to the head of line blocking which is a significant performance limiting factor. To remedy this predicament, the virtual channel provides an alternative to improve the performance, but is not amenable to the cost-efficient on-chip implementation. Some complex mechanisms, e.g. virtual channel arbitration, 2-phase switch allocation, increase the normal pipeline stage. By the detailed analysis of router with virtual channel, it can be found that the switch allocation and switch traversal are necessary during the transfer of each packet but the pipeline delay of former always exceeds that of latter [13]. Hence, we believe that the complex 2-phase switch allocation may be preferred attributing to the arbitration among multiple packets at high rates, but it would increase the communication latencies at low rates, where the redundant arbitrations which increase the pipeline delay is unwanted because no contention happens.

Given the aforementioned analysis, we thereby introduce another alternative to optimize the single-cycle router. When an input port receives a packet, it computes the state of the forwarding path based on the output direction and switch allocation results. As the forwarding path is free, this port grants the packet with wing channel, and then bypasses the 2-phase switch allocation pipeline stage to forward the packet directly. For the purpose of illustration, we first introduce the original speculative router [13] and then describe the change details. The main components of original router include input buffer, next routing computation (NRC), virtual channel arbitration (VA), switch allocation (SA) and crossbar units. When a header flit comes into the input channel, the router at the first pipeline stage parallelizes the NRC, VA and SA using speculative allocation, which performs SA based on the prediction of VA winner, otherwise cancels SA operation regardless of its results when the VA operation is failed due to conflicts with other packets. Then, at the second pipeline stage the winner packets will be transferred through the crossbar to the output port.

Fig. 1 illustrates our proposed low-latency router architecture changed from the original one mentioned above. First, each input port is configured with a single wing channel, which uses the same number of flit slots to replace a certain normal VC reference, thus keeping the buffer overhead as a constant. In our proposed architecture, the wing channel as a special type of VC has its own simple mechanism. By inspecting the forwarding path of arriving packets to be free, the wing channel only holds them and asserts their request signals to fast arbiter immediately to implement the single-cycle transfers. Note that since the packet transfer at wing channel is performed when the switch allocation results of other normal channels are failed at previous cycle, it has lower priority than the normal transfers. Second, we add the fast arbitration logic which is of low complexity to handle the requests from wing channels of all inputs. Here, the extra one-stage fast arbitration logic implemented by five 4:1 arbiters incurs little hardware overhead, in contrast with 25 arbiters of the two-stage virtual channel arbitration. Through the fast arbitration, the winner will traverse the crossbar switch right now. Third, each input introduces a channel dispenser unit. Distinguished with the original router, the proposed router just allocates the logical identifier of free channel at neighborhood. According to the logical identifier stored in the header flits, the dispenser unit grants the physical channel identifier to the new packets. Besides, this unit is also responsible for selecting proper packets to use the wing channel, whose detail is described later in Section 2.2. At last, we use the advanced request signal to perform the routing computation in advance. In original router, the routing computation or next routing one must be completed before the switch traversal, and we find that it is difficult to