DESIGN OF A MULTICAST ROUTER FOR NETWORK-ON-CHIP ARCHITECTURES WITH IRREGULAR TOPOLOGIES

Hsi-Che Tseng, Zhi-Hong Ye, and Hsin-Chou Chi
National Dong Hwa University, Taiwan, hcchi@mail.ndhu.edu.tw

ABSTRACT. As chip complexity keeps increasing in system-on-chip (SoC), the on-chip interconnect has become a critical issue for large-scale chip design. It has been proposed that the packet-switched network exchanging messages between intellectual property (IP) cores is a viable solution for the SoC interconnect problem. The design of the router in such network-on-chip (NoC) architectures is the key to high-performance communication for the IP cores in SoC. In this paper, we present the design and implementation of a multicast router for NoC with irregular topologies. The router employs our previously proposed tree-based routing algorithm for irregular networks. Our experiment results show that the multicast router has a slightly lower clock rate and moderately larger chip area than the unicast router in NoC. Since multicasting is a technique providing superior network performance, especially for large networks, such multicast router design is an effective routing solution for large-scale network-on-chip architectures.

Keywords: network-on-chip, router, system-on-chip, VLSI design

INTRODUCTION

As modern VLSI systems become more complex, designers are faced with many challenges. One of the major challenges is the design of the communication infrastructure between homogeneous or heterogeneous cores having different characteristics. Conventional on-chip communication uses dedicated wires or shard buses to connect on-chip IP cores. However, these approaches suffer from many drawbacks in modern system-on-chip (SoC) designs (Benini, 2005; Dally, 2001).

Recently, network-on-chip (NoC) architectures have been proposed as the communication framework for SoC design (Nurmi, 2002; Hu, 2003). Packet switching is usually employed for data exchange between IP cores for such on-chip networks. One of the critical issues for designing on-chip networks is deadlock-free packet routing, especially when there are faulty routers or links. When some of the routers or links are faulty, the network topology is no longer regular. Thus, the original deadlock-free routing scheme for regular networks cannot be applied.

To solve the packet routing problem for irregular networks, several schemes have been proposed (Koibuci, 2001; Qiao, 1996; Zhou, 2006). However, most of them have many restrictions, which either do not allow all healthy nodes to be in the routing network or require complex, costly routing hardware in the router. Deadlock-free tree-based routing has been proposed to solve the routing problem of irregular networks. However, these proposed routing schemes still require costly routing tables in the router (Koibuci, 2001; Qiao, 1996; Zhou,

2006). As the size of the on-chip network increases, the chip area for the routing table increases significantly. Such disadvantage severely limits the size of practical on-chip networks.

In this paper, we present the design of the multicast router based on our previously proposed tree-based routing scheme called TRAIN (Tree-based Routing Architecture for Irregular Networks) for faulty mesh on-chip networks (Chi, 2010). TRAIN is deadlock-free, and does not require a routing table in the router and hence can be efficiently implemented in NoC architectures. When part of the components, including links and routers, are faulty and cannot be used, our reconfigured routing network can still deliver the injected packets to their destinations successfully. Furthermore, our multicast router can transmit packets with more than one destination and provide better utilization of network bandwidth.

TREE-BASED ROUTING FOR IRREGULAR NETWORKS

An irregular network consists of an arbitrary number of nodes and has an irregular topology. Figure 1 shows the example of an irregular network consisting of 9 nodes. Each node includes an IP core and an associated router in the network. We assume that a router connects to only one local IP core in this paper since it is the typical case for most on-chip networks. The efficient design and implementation of the router and its routing function have significant impact on the performance of the network.

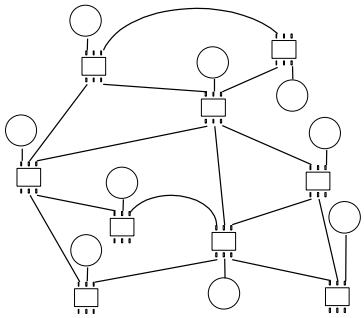

Figure 1. An irregular network. Each circle and each rectangular indicate an IP core and a router, respectively

We assume that each port in the router is associated with a pair of input and output channels. The link connected to a port in the router leads to either the local IP core or a neighboring router. A router often includes buffers at the input ports for storing incoming packets that cannot be forwarded immediately due to output port contention or blocking. The crossbar in the router provides unblocking paths between the input ports and output ports. A hardware routing decision unit in the router determines during run-time which output port an arriving packet is destined to according to the information in the packet header.

Many deadlock-free routing algorithms have been proposed for regular networks such as mesh, torus, and hypercube. While many proposed routing schemes for regular networks have been shown effective, there have been relatively few counterparts for irregular networks. Most of these routing algorithms achieve deadlock freedom by avoiding the possibility of deadlock situations. A well-designed routing algorithm should achieve deadlock freedom without significantly reducing the network performance or increasing system complexity.

TRAIN Routing

Our proposed TRAIN routing uses a subset of the network to construct a tree. With these previously proposed three routing schemes, some of the links cannot be used for packet transmission. However, with our TRAIN, all links not belonging to the tree become shortcuts and can be used. Furthermore, there is an algorithmic routing decision unit in the TRAIN

router, which determines the destined output port for the arriving packets. TRAIN does not require a routing table in the router, and hence can reduce router complexity significantly, especially for large networks.

The basic idea of TRAIN is as follows. A packet sets out from the source node and tries to follow the tree route to reach the destination node. When a transmitted packet is in any router, including the source, it tries to take advantage of the shortcut if shorter distance than the "planned" tree route is provided. These shortcuts help reduce the number of hops a packet traverses from the source to the destination. In addition, more bandwidth is offered in the network and traffic congestion in the area near the root of the tree is mitigated. Note that a packet may take advantage of more than one shortcut along its journey, and the path is not necessarily the shortest.

When a packet arrives in a router, the routing decision unit checks if there is a shortcut leading to a router "closer" to the destination compared with the tree route. If such a shortcut exists, a packet is routed to the neighboring router associated with the most profitable shortcut. If the desired shortcut does not exist or is currently blocked, the packet simply continues to follow the planned tree route. A key to the effectiveness of TRAIN is that distance calculation of tree route between any two nodes should be simple enough. Such distance calculation function should be efficiently implemented in an algorithmic hardware and hence no routing table needed. We have developed a distance calculation algorithm for TRAIN.

Figure 2 shows an example network with TRAIN. The number beside each node is the node ID used for distance calculation of tree route. The labeling method is as follows: The root node is labeled 000...0. The tree is not necessarily a binary tree. The children of the root are labeled 100...0, 200...0, 300…0, and so on. From source node 121 to destination node 220, for instance, the planned tree route takes five hops. However, when the packet arrives in node 120, it finds that the shortcut leading to node 200 provides shorter route. Hence the eventual distance that the packet traverses from node 121 to node 220 is reduced from 5 to 3. Note that in order for the routing decision unit to work, a router is required to store the neighboring nodes' IDs for the shortcuts. This simple information requires very little storage and is set up through ID exchange between neighboring routers during initialization.

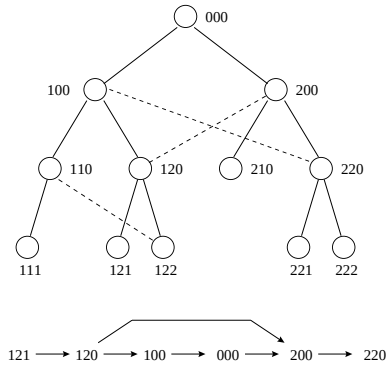

Figure 2. An irregular network. Each circle and rectangular indicate an IP core and a router, respectively

DESIGN OF A MULTICAST ROUTER FOR IRREGULAR NETWORKS

We have designed a multicast TRAIN router for NoC architectures, as well as a unicast TRAIN router for comparison of their cost and performance. The routers are reconfigurable when the network is faulty. The designs have been implemented as IPs for NoC platforms. Figure 3 illustrates the block diagram for the organization of the multicast TRAIN router. The

router inside is a two-stage pipeline, and thus it takes at least two clock cycles for a packet to traverse the router, not including the transmission delay on the link. Assuming that the transmission on the link takes one clock cycle, each hop takes three clock cycles if there is no queueing delay. There is a latching register at each input port and a latching register at each output port. A packet is latched in the input register when it arrives in the router, and is latched in the output register before it crosses the link. The width of each link is 34 bits, and one flit can be transmitted across the link in one clock cycle.

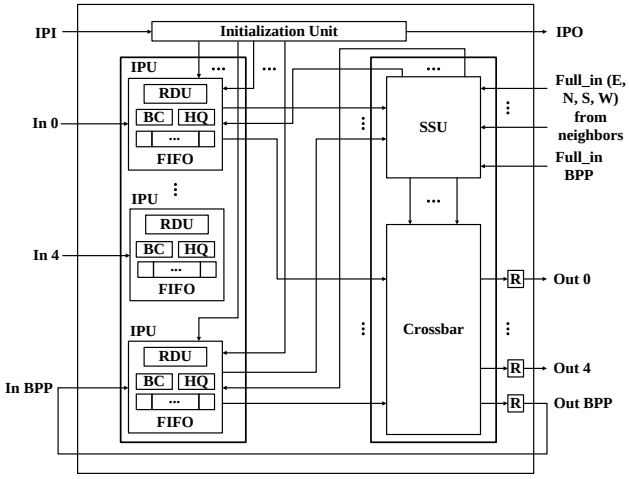

Figure 3. Organization of our multicast router

There are five external ports and one internal port in the multicast router. The five external ports can connect to four neighboring nodes and one local node with the IP core. In a mesh network, the five ports can be assigned as the East (E), North (N), South (S), West (W), and Local (L) directions. The only internal port is called the blocked packets port (BPP). The BPP is used to accommodate those packets that cannot be forwarded to the next hop entirely. In our multicast router, a multicast packet can be duplicated with more than one destination output ports. If some of the destined output ports are blocked, the packet will be forwarded back to the BPP, waiting in the FIFO queue there. Only the duplicated packets with free output ports are forwarded to the neighboring nodes. In the first stage, the input port unit (IPU) at each input port include the packet buffer (FIFO), the header buffer (HQ), the routing decision unit (RDU), and the buffer control unit (BC). A packet is received in the packet buffer, and the buffer is a FIFO queue. Packet headers are separately stored in the header buffer as a queue too. Tree distance is calculated by the RDU for the tree route and shortcuts for the head packet in the queue. The results of distance calculation for the tree route and shortcuts are latched, respectively. The BC is responsible for coordinating the writing and reading of packets of the packet buffer.

In the second stage, the basic blocks for the router include the crossbar switch and the switch scheduling unit (SSU). At each clock cycle, the SSU decides which crosspoints in the crossbar should be connected based on the requests from the six input ports, including the internal BPP. One of the keys to the design of our router is the processing speed of distance calculation unit. Distance calculation has to be fast enough such that it does not slow down the clock frequency. The simplicity of our distance calculation algorithm facilitates efficient processing. Our multicast router can process packets with up to five destinations. In the packet header, there can be more than one header flits for different destination nodes. Figure 4 and Figure 5 show the packet format and the format of a header flit, respectively, for the multicast network. In each header flit, 32 bits are used for the destination node IDs, and two bits are for

the flit type (no flit, header flit, last header flit, or data flit). The length of the packet body for data is 170 bits.

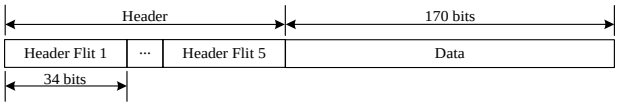

Figure 4. Packet format

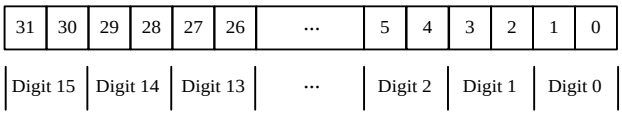

Figure 5. A packet header flit has 16 digits and each digit consists of 2 bits. Two extra bits besides the 32 bits are used to indicate the flit type

When a packet is at the front of the FIFO buffer in the input port, the SSU will decide, according the header flits, if the packet should be duplicated and the routing path should be forked. If some destined ports are free and the others are blocked, the packet will be forwarded back to the BPP for the blocked destined output ports. For the free destined output ports, the packet is duplicated and sent to the neighboring nodes in forked paths.

In the design of our router, the FIFO buffer has four slots and can accommodate at most four packets. In the router, the neighboring node IDs and the local ID are stored and ready for use in routing decision. After the RDU processes the packet header, the destined output ports are stored separately in the header queue at each input port. Inside the RDU, five hardware routing distance calculation units can calculate the distance from each neighboring node (E, N, S, W, and L) to each destination node. Owing to the simplicity of our distance calculation algorithm, these can be done in one clock cycle.

IMPLEMENTATION OF THE MULTICAST ROUTER

We have implemented our multicast router in two process technologies, including TSMC 180nm and 90nm using cell-based design flow. For comparison, we also implemented a unicast router to evaluate the cost of incorporating the multicast capability. From Table 1 and Table 2, we can see that, the chip area and power consumption increase moderately for incorporating the multicast capability. However, the clock rate of the multicast router only decreases a little at 3.3% and 2.6% for the increased complexity with 180nm and 90nm process technologies, respectively. A multicast router can reduce the load of the network significantly and provides much better utilization of the network bandwidth. Hence, with reasonable extra chip area cost and slightly lower clock rate for the router, the multicast router is a feasible and practical component for high-performance network-on-chip architectures.

Table 1. Implementation results for the unicast router and the multicast router with TSMC 180-nm technology

	Unicast Router	Multicast Router	Comparison
Process Technology	TSMC 180-nm		
Frequency	62 MHz	60 MHz	3.3% ↓
Core Area	1807877.9 μm^2	2619771.7 μm^2	44.9%↑
Power Dissipation	18.4294 mW	25.2644 mW	37.1%↑
Gate Count	181131	262965	45.1%↑

Table 2. Implementation results for the unicast router and the multicast router with TSMC 90-nm technology

	Unicast Router	Multicast Router	Comparison
Process Technology	TSMC 90-nm		
Frequency	117 MHz	114 MHz	2.6% ↓
Core Area	446800.4 μm^2	644613.4 μm^2	44.2%↑
Power Dissipation	8.3130 mW	11.4319 mW	37.5%↑
Gate Count	183847	266383	44.8%↑

SUMMARY AND CONCLUSIONS

In this paper, we presented the design of a reconfigurable pipelined multicast router for mesh on-chip networks with irregular topologies. The router is based on our previously proposed TRAIN routing algorithm and requires no routing table. Owing to the simple routing decision algorithm, the routing decision can be made rapidly and done in one clock cycle. From the implementation results of our multicast router based on the two process technologies, the chip area and power consumption increase moderately, compared to the unicast router. Furthermore, the clock rate only decreases slightly at 3% or so. The experiment results validate the idea that a multicast router for the irregular network-on-chip architectures with TRAIN routing is feasible. With such multicast routers incorporated in the network, the load of the network can be significantly reduced. Hence, network-on-architecture architectures with the design of the multicast router can provide high-performance communications between IP cores for many applications.

REFERENCES

Benini, L., & Bertozzi, D. (2005). Network-on-chip Architectures and Design Methods. *Proceedings of IEE Computers and Digital Techniques.*

Chi, H. C., Jhang, Y. H., & Chen, W. S. (2010) Tree-Based Routing for Faulty On-Chip Networks with Mesh Topology. *Proceedings of IEEE Computer Society Annual Symposium on VLSI*.

Dally, W. J., & Towles, B. (2001). Route Packets, Not Wires: On-Chip Interconnection Networks. *Proceedings of Design Automation Conference.*

Hu, J., & Marculescu, R. (2003). Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures. *Proceedings of Design Automation and Test in Europe Conference and Exhibition.*

Koibuci, M., Funahashi, A., Jouraku, A., & Amano, H. (2001). L-turn Routing: An Adaptive Routing in Irregular Networks. *Proceedings of IEEE International Conference on Parallel Processing*.

Nurmi, J, Saastamoinen, I., & Siguenza-Tortosa, D. (2002). Interconnect IP Node for Future Sytem-on-Chip Designs. *Proceedings of IEEE International Workshop on Electronic Design, Test and Applications.*

Qiao, W., & Ni, L. M., (1996). Adaptive Routing in Irregular Networks Using Cut-Through Switches. *Proceedings of International Conference on Parallel Processing*.

Schroeder, M. D., et al. (1990). Autonet: a High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links. *SRC Research Report 59, DEC*.

Zhou, J., Lin, X. Y., & Y. C. Chung (2006). A Tree-Turn Model for Irregular Networks. *Proceedings of IEEE International Symposium on Network Computing and Applications*.