Ijesm
Consulting, help, relaxation
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES
&
MANAGEMENT

TWO DIMENSIONAL LOW LATENCY NOC ROUTER BY WORMHOLE SWITCHING

Rajesh Nema[*1], Teena Raikwar[2],Prerna Suryavanshi3
Dept. E&E, NRI institute of science of technology, Bhopal

Abstract
Network on Chip (NoC) is an approach to designing communication subsystem between intelligent property (IP) cores in a system on chip (SoC). Packet switched networks are being proposed as a global communication architecture for future system-on-chip (SoC) designs. In this project, we propose a design and implement a wormhole router supporting multicast for Network-on-chip. Wormhole routing is a network flow control mechanism which decomposes a packet into smaller flits and delivers the flits in a pipelined fashion. It has good performance and small buffering requirements. The implementations are at the RT level using VHDL and they are synthesizable. First, based on virtual cut through router model, a unicast router is implemented and validated and based on the wormhole switching mode the multicast router architecture is designed and implemented. A Wormhole input queued 2-D mesh router is created to verify the capability of our router.
Key words- Network on chip, router architecture, wormhole switching

Introduction

System-on-chip (SoC) designs provide integrated solutions to challenging design problems in the telecommunications, multimedia, and consumer electronics domains. However, the designs of SoCs encounter some challenges with the advanced process technologies and SoC complexity scales. The challenges include the negative effect of technology scaling on global interconnects, growing system complexity, the need to construct flexible multi-use designs and platforms and so on. The Network-on-Chip (NoC) has been recognized to solve these challenges.

Networks-on-Chips (NoC) is a bridge concept from Systems on-Chip (SoCs) into Multiprocessor System-on-Chip (MPSoC). A SoC design approach uses sometimes more than one processing element (PE) to implement an integrated circuit for a certain system application. The PEs send messages to other PEs for sharing computational processes to complete tasks. A sophisticated communication structure is needed for inter-processor data exchange. Rather than using a bus for single communication among PEs, or using point-to-point communication, a concept of shared segmented communication infrastructures is proposed to support application-scalability.

*** Corresponding Author**
E. mail: rajeshnema2010@rediffmail.com

Research studies with the Arteris NoC [1] have demonstrated the feasibility and advantages of NOCs over traditional bus-based architecture but have not focused on compatible communication standards.

Computer bus-based communication architectures do not easily handle the real-time data flows associated with networking, telecommunication, and multimedia data streams [2]. On-chip networks will meet the distinctive challenges of providing functionally correct and reliable operation of interacting system-on-chip components [3]. The effectiveness of NoC platforms for MPSoCs is more and more significant when a huge number of PEs is used in the MPSoC. Therefore, NoCs enable promising concepts for the design of supercomputers with multiprocessor cores. NoCs have also potential applications in integrated control systems, e.g. in automotive electronic control and entertainment systems. The NoC concept has potential to provide sustainable platforms and proposes a new paradigm in SoC architecture and multiprocessor systems [4].

The Network topology could influence the scalability and performance of the NoC. The architecture and routing decision must meet bandwidth requirements and should be scalable for wide range of applications. Some NoCs that have been developed with Mesh topology are NOSTRUM [5], SoCBUS [6], RAW [7], PNoC [8], CLICH´E [9] and HiNoC [10]. OCTAGON NoC

[11] uses octagon topology. Fat tree topology is used in SPIN [12], and its extended version DSPIN [13] uses mesh distribution of clusters. Flexible regular and irregular topology is presented in [14]. Xpipes NoC [15] supports a customized topology. ASNoC [16] is an application specific NoC, where the design methodology supports the development of NoCs with 2-D mesh and hierarchical irregular topologies.

Message transmission from a resource to another resource through the intermediate router nodes can be divided into synchronous and asynchronous methods. In synchronous designs, global clock-trees are distributed, which leads to electromagnetic interference effect and clock power consumption. Asynchronous communication design is a promising concept, but lacks of industrial standard tools support, especially with respect to testability issues. Synchronous NoCs can also support GALS (globally asynchronous, locally synchronous) concept by implementing asynchronous input/output queues in network interfaces. The NoC is considered as a synchronous island, where resources clocked with different frequencies are connected through network interface.

A resource may send a short, medium, long messages, or even
a data stream in a certain time duration to other resources. In real-time applications, some messages require lower completion bounds or higher bandwidth. These messages are e.g. run-time reconfiguration data (small until medium size messages), or video stream packets (very large size messages), or message having critical time constraint. Therefore, a NoC providing two priority level for packet services, i.e. for low and high priority messages are introduced in this paper that supports applications running normal traffics and time-critical traffics.

On-Clip Network Feature Characteristic

A. Two-Level Priority Message Delivery Service

The early prototype of our NoC has provided only a single priority with *Best-Effort* (BE) service. This paper proposes a new prototype with additional services for different level of priority. Therefore new modules i.e., a *HP FIFO buffer*, *TypD* unit and *VCSC* are inserted in each port to support the new service as shown in Fig. 2(b).

1) Low Priority Message Service (LP): Our proposed LP service guarantees lossless packet completion and in-order message delivery, but provides no commitment to latency bound or data throughput because the messages are sent with a packet-based approach. The LP packets are routed using a minimal west-first adaptive routing algorithm, where the packets will not be routed away from their target nodes. The routing is made on flit-by-flit basis, where different packets from different input ports share the link wires using wormhole switching, and can be interleaved in the FIFO. The incoming packet flits, which require the same link, are selected by an arbiter unit using a fair round-robin arbitration.

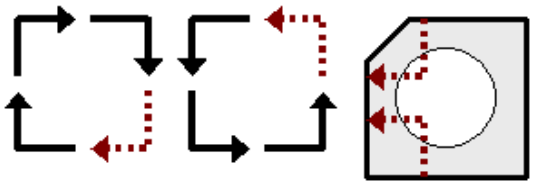


Fig.1. The turn model of adaptive West-First routing algorithm (the solid lines are allowed turns, and the dashed lines are prohibited turns).

2) Higher Priority Message Service (HP): The HP packets will reserve absolutely the selected links to route the packets into their target nodes. The HP packets are routed using a misrouting or non-minimal west-first routing algorithm, where the packets can be routed away from their target nodes to find an optimal path. The HP packets will firstly be routed into the possible output links, which have not been used by another packet. Non minimal routing will be undertaken as long as the routing does not violate the prohibited turn-models as shown in Fig. 1. Otherwise, the HP packet will select the output link
which has more free IDs and more available free registers in the downstream FIFO. The HP packet will not share the reserved links with other HP packets. These packets will be buffered into the FIFO in virtual HP channels. The arbiter will always prioritize the flits in the HP buffer (higher priority) as a winner to access the output ports. The LP packets must wait until the HP buffer is empty or the last flit of the HP packet has been forwarded from the HP buffer. In this situation, the HP packet uses the full bandwidth of its reserved links.

B. Packet interleaving and Identity-Slot Divison Multiplexing

The contention-free routing can be implemented by using a time division multiple access technique. This approach uses a pipelined circuit switching method. Contention in our NoC is handled by using an identity-slot division multiple access technique for wormhole packet interleaving. Each wormhole packet is injected into the network with the same identity-tag (ID-tag). But each time the flits of the

packets are forwarded into the next router, their local ID-tag will be updated. Each flit belonging to the same message will have the same local IDtag in a certain communication link to differentiate it from other flits of other messages. Therefore dynamic packet identity management (IDM) modules are implemented over the link to map old local ID-tag of each flit into new local ID-tag.

C. Congestion-Aware Adaptive Routing

Deadlock is a situation where all packets in the deadlock configuration cannot be forwarded to the next network nodes. Deadlock is formed by a cycle, where there are channel dependencies between packets in that cycle. By introducing one prohibited turn in one turn model, deadlocks can be prevented. By introducing a small number of virtual channels deadlock can be avoided. It seems that implementing virtual channel is as good solution for deadlock free router design. However, the use of the virtual channel leads to high area and logic utilization. As shown in Fig. 1, the turns from North to West and South to West are prohibited. Hence, packets are routed firstly to West when the target nodes are located in North-West or South-West quadrants. Packets can be adaptively routed when destination nodes are located in North-East or South-East quadrants. Our NoC uses one-hop congestion measurement, i.e. by considering the congestion of the adjacent nodes to select adaptively two possible directions. The adaptiveness of the routing algorithms is also based on the number of free identity-slots provided by identity-slot manager (IDM) modules. The availability of ID-slots in a communication link segment denotes actually that there is still available bandwidth that can be allocated for new incoming messages. When there are two possible output directions to route a packet, then the router will consider firstly the number of free ID-slot provided by the IDM units at both output ports. Secondly, the router will consider the number of free registers in the FIFO buffer at the two adjacent mesh nodes.

The packet will be routed to the output direction, where more free ID-slots are available in the output direction. If the numbers of free ID-slots are the same for two possible directions, then the router will select the direction, where more free registers are available in the next downstream FIFO buffer.

D. Special Packet Format with Extra Control Bits

The 38-bit packet consists of a header flit followed by payload flits. Two additional 3-bit heads are Type and ID (Identity) bits. The Type can be a header, a data body, and the end of data body (last flit). The 3-D source and target address of the packet are asserted in the header flit. The Z-address is reserved for further development of 3-D NoC topology. Each message is associated as single packet even if the size of message is very large. It means that each message will have only one header flit for one target node. The message body will travel in the NoC to follow links set up by the header flit, and the end flit of the message will close the link reservation. This approach will guarantee in-order message delivery even if adaptive routing algorithm is used, because the header flit is the only flit, which is routed adaptively to find an optimal link. Each of its flits has the same local identity number (ID-tag) to differentiate it from flits of other packets, when it passes through a communication segment of the NoC. The ID-tag of the data flits of one packet will vary over different communication segments in order to provide a scalable concept. Fig. III-D represents bit encodings for packet types. The LP packet is encoded with binary code ’001’ and the HP packet with binary code ’100’. The remaining binary codes can be used for other types of packet for future investigation, e.g. for connection-oriented guaranteed-throughput packets.

Generic Low Latency Router Architecture

For the proposed NoC architecture a reconfigurable wormhole router architecture is used, the details of which are presented below. This was chosen over other options (e.g. storeand- forward, virtual-cut-through) because of its low routing latency, low complexity and high buffer utilization. Two key attributes that have allowed this to be achieved are 1) high scalability, mainly in router radix in order to accommodate highly connected topologies, and 2) optimized pipeline organization in order to reduce hop delay.

A. Components

The router mainly comprises input ports, switch arbiters, Finite State Machines (FSM) and the crossbar. This comprises a head flit, a body flit and a tail flit. Here the packet length is unfixed and longer packets can be formed by adding more data flits between the head and the tail. The Output Channel (OC) field stores the output channel used by the packet. This is pre-computed using look-ahead routing, one hop ahead. The width of the address field and output channel field both vary with network size. The router is fully pipelined allowing flits to pass through it in this manner.

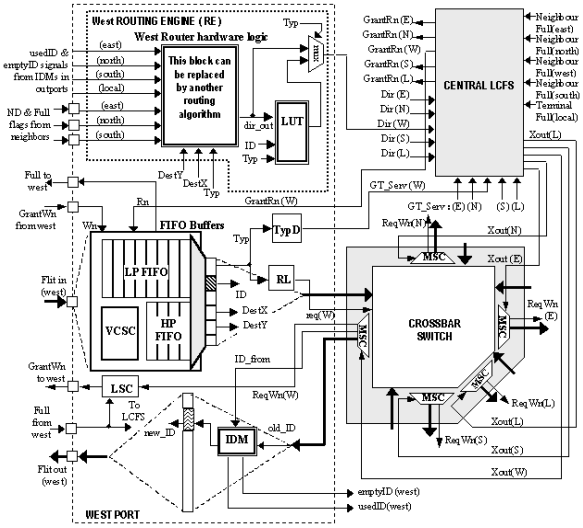


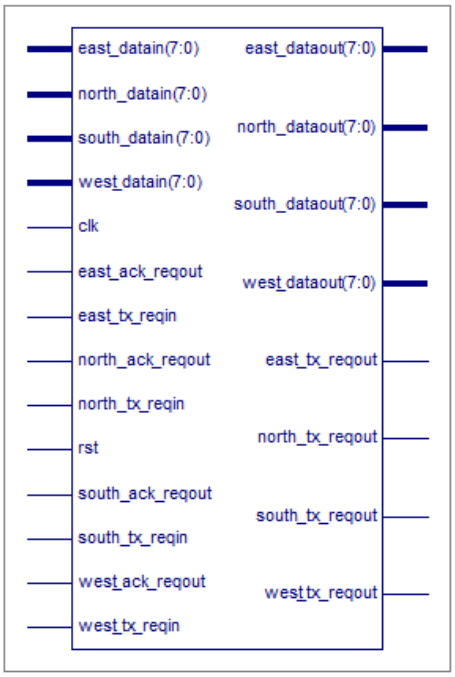
Fig.2. Architecture

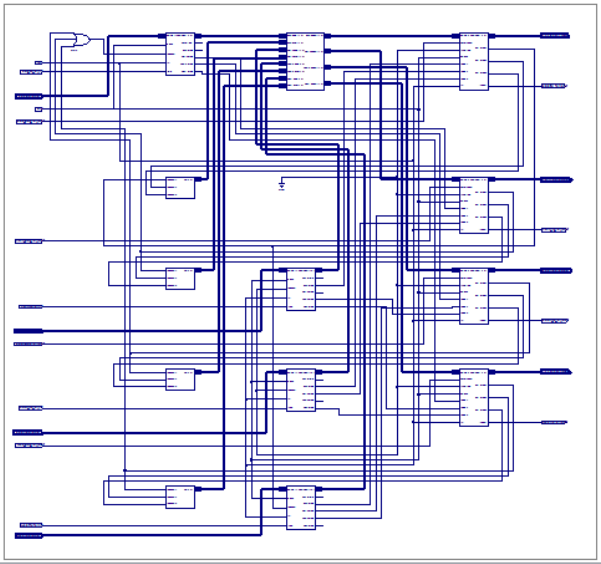
Fig. 2. Modular architecture and the special packet format of our NoC with extra 6 control bits (Flit type and ID-tag fields).

The router can be categorized in following parts according to their functionality.

1. In channel: This block is responsible for data handling from network. It take cares of the identification of data and header information by packet divider. Further the headers can be considered as flits. Which further propagate through crossbar circuitry to destination. FIFO is used for data storage in such a way that it will not wait for entire data to be buffered, depends upon flit requirements and controller value. It will take decision when to buffer the information and release the information to destination node.
2. FSM Controller: It monitors the operation between source and destination and also responsible for handshaking establishment. It also controls the FIFO information depends upon FIFO full or FIFO empty.
3. CrossBar Switch: It establish the routing path depends upon the grant and request from 4 nodes i.e. east node, west node, south node, north node. And decides which information will pass to which node depends upon priority given by arbiter.
4. Arbiter: In our design we have implemented distributed arbitration scheme i.e. each nodes having its own architecture block. It takes input as a request from various nodes and the request which will be in highest priority. Based on that it will get grant.
5. Output Channel: It is a destination interface between other nodes. It takes input from MUX and passes the information to the output channel of other node.
6. Decoder: It is a header decoder, based on that node selection can be decided.

Result





Conclusion

Design has been developed with verilog code and implemented in xillinx ise 9.2 with spartan3 device. The functionality has been verified by testbench I n modelsim. Our design has performed in two dimensional routing features. Even though it is parallel but because of wormhole deadlocks can be there so thorough testing has been done for deadlock operation.

References

1. P. Martin, "Design of a Virtual Component Neutral Network-on- Chip Transaction Layer," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'05)*, pp. 336-337, 2005.
2. D. Wingard, "MicroNetwork-Based Integration for SOCs," *Proc. Design Automation Conf. (DAC'01)*, pp. 673-677, 2001.
3. L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, vol. 35, pp. 70-78, Jan. 2002.
4. A. Jantsch and H. Tenhunen, *Networks on Chip*, Kluwer Academic Publisher, Hingham, MA, USA, 2003..
5. M. Millberg, E. Nilsson, R. Thid and A. Jantsch, "Guaranteed Bandwidth using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'04)*, pp. 890-895, 2004.
6. D. Wiklund and D. Liu, "SoCBUS: Switched Network on Chip for Hard Real TIme Embedded Systems," *Proc. IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS'03)*, 8 pp., 2003.
7. M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, et. al., "The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs," *IEEE Micro*, vol. 22, issue 2, pp. 25-35, Mar-Apr. 2002.
8. C. Hilton and B. Nelson, "PNOC: a flexible circuit-switched NoC for FPGA-based systems," *IEE Proc. Computers and Digital Techniques*, vol. 153, no.3, pp. 181-188, May 2006.
9. S. Kumar, A. Jantsch, J. -K. Soininen, M. Forsell, M. Millberg, J. O¨ berg, K. Tiensyrja and A. Hemani, "A Network on Chip Architecture and Design Methodology," *Proc. IEEE Computer Society Annual Symposium on VLSI*, pp. 105-112, 2002.
10. M. K. -F. Sch¨afer, T. Hollstein, H. Zimmer, M. Glesner, "Deadlock- Free Routing and Component Placement for Irregular Mesh-based Network-on-Chip," *IEEE/ACM Int'l Conf. on CAD (ICCAD'05)*, pp. 238-245, 2005.
11. F. Karim, A. Nguyen and S. Dey, "An Interconnect Architecture for Networking Systems on Chips," *IEEE Micro*, vol. 22, issue 5, pp. 36-45, Sept-Oct. 2002.
12. P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnection," *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'00)*, pp. 250-256, 2000.
13. I. M. Panades, A. Greiner and A. Sheibanyrad, "A Low Cost Network-on-Chip with Guaranteed Service Well Suited to the GALS Approach," *Proc. the 1st Int'l Conf. and Workshop on Nano-Networks)*, pp. 1-5, 2006.
14. T. A. Bartic, J. -Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde and R. Lauwereins, "Topology adaptive network-onchip design and implementation," *IEE Proc. Computers and Digital Techniques*, vol. 152, no.4, pp. 467-472, July 2005.
15. L. Benini and D. Bertozzi, "Network-on-chip architectures and design methods," *IEE Proc. Computers and Digital Techniques*, vol. 152, no.2, pp. 261-272, Mar. 2005.\
16. J. Xu, W. Wolf, J. Henkel and S. Chakradhar, "A Design Methodology for application-Specific Networks-on-Chip," *ACM Trans. On Embedded Computing Systems*, vol. 5, no. 2, pp. 263-280, M