International Journal of Electronics and
Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 5, Issue 1, Dec – Jan 2016, 33-44

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

DESIGN OF ROUTING AND MESSAGE-DEPENDENT DEADLOCK AVOIDANCE SCHEME FOR NoC

DIVYA D. S.

Project Engineer II, Centre for Development of Advanced Computing, Trivandrum, India

ABSTRACT

Network-on-Chip (NoC) is the most advanced technology to address the communication bottleneck of System-on-Chip (SoC). Routing algorithms using wormhole packet switching is best for NoC as it provides high throughput, low latency and requires small buffers. But it can cause deadlock during the system operation. Ensuring deadlock-free operation of NoC is a major challenge. This paper presents a combined method to make the network routing-dependent deadlock-free by using hold-release tagging method and message-dependent deadlock-free by adding virtual channels. This uses wormhole switching and supports adaptive and multicast routing. The data body flits of multicast packets will follow the same path made by the multicast headers which are sending in advance. An additional Identification (ID) field is packetized together with the flits belongs to the same packet. These methods are supported by the NoC router which controls the deadlock in tree-based multicast routing.

KEYWORDS: Deadlock-Free Routing, Network-on-Chip (NoC), Wormhole Packet Switching, VLSI Technology

INTRODUCTION

Network-on-Chip is the most advanced technology for the communication between different computing resources such as CPU, DSP and specific IPs etc in System-on-Chip. Thus Networks-on-Chip (NoCs) are a promising and flexible communication infrastructure to address the communication bottleneck of Systems-on-Chip (SoCs) which has emerged as a result of growth of VLSI technology. The NoCs are characterized by high level of parallelism, high performances and scalability. In NoC technology, we are inserting some routers in between each communication object which helps to shorten the wiring. With this approach, system modules communicate by sending packets to one another over the network with the help of different routing techniques. The paths that the data packets take in the network depend on two basic NoC parameters; routing algorithm and topology. The frequently used topologies are mesh, torus, hypercube and butterfly. Depending on the degree of adaptiveness, there are three categories of the routing algorithms - non adaptive, partially-adaptive and fully adaptive. The non adaptive routing algorithms or often called deterministic routing algorithms which route the data packet via only one path between the source and destination components. In the partially-adaptive routing algorithms, between the source and destination components, a set of available paths can be used for the packet routing but not for all packets. The partially routing algorithms produce a routing with limited adaptivity and establish a balance between performances and router complexity. In the fully adaptive algorithms, communication between the components takes place via any of the available paths for all packets. The fully adaptive routing algorithms make the potential for the better use of all available network resources to the detriment of the router complexity. Another criterion to distinguish routing algorithms is according to the packet transport principle like store and forward, wormhole-routing, virtual cut-through or circuit switching in which wormhole switching is commonly used. In the wormhole switching method, messages are divided into a number of flow control digit or commonly called as flit. Every flit may bring a data word. The main advantage of the wormhole switching is that the buffer size can be set as small as possible to reduce the buffering area cost. The commonly used communication service is multicast in which the same message is sent from a source node to an arbitrary number of destination nodes.

Multicast routing algorithm can be classified as unicast-based, path based and tree-based. In unicast-based, the multicast operation is performed by sending a separate copy of message from the source to every destination or, alternatively, by sending the unicast message to subset of destinations. The drawback of this scheme is on account of the fact that multiple copies of the same message are injected into the network, the traffic of the network will be increased. Furthermore, each copy of message loses considerable startup latency at the source. In path-based multicast routing, the Processing Elements (PEs) that injects message has to set up the order of headers containing address of all multicast destination nodes, in order to find optimum paths from the PEs to the destination nodes.

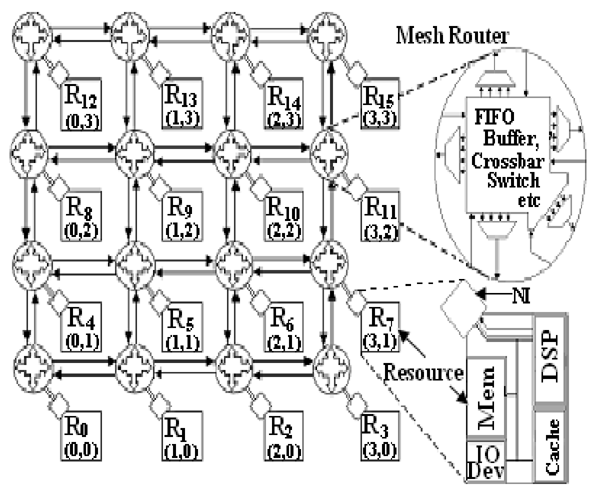

Figure 1: NoC in 2D Mesh 4x4 Topology

In the tree-based multicast routing, the header ordering in source nodes is not required. The multicast routing will form communication paths like branches of trees connecting the source node with the destination nodes at the end points of the tree branches. A higher probability that multicast deadlock occurs in intermediate nodes is the disadvantage of the tree-based multicast routing. Tree-based routing incurs high congestion in wormhole networks. This paper presents an efficient and effective method to solve the routing-dependent multicast deadlock problem in the intermediate nodes when using adaptive tree-based multicast routing. For this we are using an extendable router having five input-output ports and the router in 4x4 mesh topology is shown in figure 1. Even if we remove the routing-dependent deadlock, message-dependent deadlock can still occur due to the dependencies and interactions created between different messages types at network endpoints, when they share resources in the network. This paper also presents a method to remove message-dependent deadlock by adding virtual channels for different message types.

RELATED WORKS

The most important of the NoC problems is freedom from deadlock and livelock, since deadlock and livelock detection and recovery mechanisms are expensive and they may lead to unpredictable delays. Deadlock occurs when the network resources are suspended, waiting for each other to be released in a cyclic fashion. If the NoC has no support to either avoid or recover from deadlocks, then correct functionality of the system cannot be guaranteed. This can lead to system crashes and unexpected system behavior, which is clearly unacceptable for SoCs.

Designing efficient methods that avoid such a situation with minimum power and area overhead is an important research area in the NoC domain. The deadlocks that can occur in NoCs can be broadly categorized into two classes; routing-dependent deadlocks and message-dependent deadlocks [17]. Routing dependent deadlocks occur when there is a cyclic dependency of resources created by the packets on the various paths in the network. For regular topologies such as the mesh and torus, the use of restricted routing functions based on turn models is an effective way to avoid routing-dependent deadlocks. Message-dependent deadlocks occur when interactions and dependencies are created between different message types at network endpoints, when they share resources in the network.

Several deadlock-free routing methods have been presented with different techniques which overcome some of the difficulties but with some limitations. There are different deadlock-free methods which use different algorithms. It includes Application Specific Routing Algorithm (APSRA) which provides deadlock-freeness, reach ability and maximum routing adaptivity [9], Module Proximity Algorithm (MPA) which provides partially adaptive deadlock-free routing for 2D mesh without VC and reroute packets in presence of faulty nodes and regions which are not rectangular[16], and deadlock-free routing algorithms using turn models such as static XY routing algorithms.

A connection-oriented deadlock-free multicast routing in wormhole switched networks-on-chip was proposed in [23]. It is a path-oriented multicast routing for mesh networks. In this deadlock is avoided with the help of XY routing and including virtual channels. But it gives priority to multicast message in mixed unicast & multicast traffic if the network is not saturated. Considering multicast traffic takes only a portion of total network traffic in the network. In order to overcome these difficulties, a new method is proposed in this paper.

MULTICAST DEADLOCK

Deadlock occurs when a packet waits for an event that cannot happen. In tree-based multicast routing, if one branch of the tree is blocked, then all are blocked. Branches will proceed forward in lockstep, which may cause a message to hold many channels for extended periods, thereby increasing network contention. Blockage of any branch of the tree can prevent delivery of the message even to those destination nodes to which paths have successfully been established.

A practical multicast routing algorithm must be deadlock-free and should transmit the message to each destination node in a little time and using as few communication channels as possible. Adaptive routing dynamically selects the route of a packet in order to make the best use of bandwidth in interconnection networks. In adaptive routing, when a packet encounters a faulty or congested path, another bypassing path can be selected. Since this allows for a better balance of network traffic, adaptive routing improves throughput and latency.

Multicast deadlock configuration will lead to the blocking of entire distribution of the multicast packet when any of its branches are blocked, especially if the message is not very short. If the deadlock problem can be solved on every router, then the network is free from multicast deadlock problem as long as the routing algorithm used to route unicast and multicast packets does not form cyclic dependencies. Figure 2 shows the occurrence of multicast configuration. Turn models which prohibits some turns in the routing function can be used to remove the deadlock occurring by cyclic dependency.

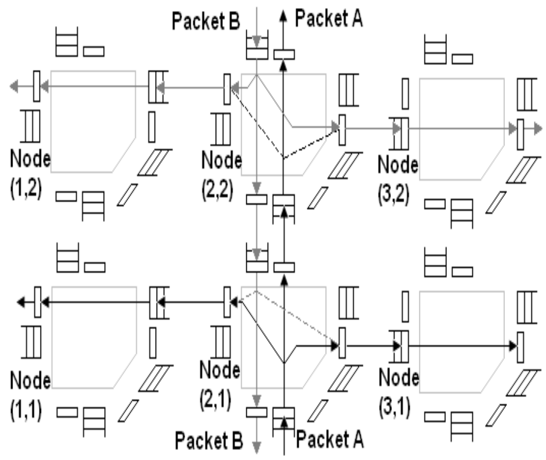

Figure 2: Multicast Deadlock Conflict

The multicast deadlock problem presented in Figure 2 emerges due to two reasons. First one is that once a flow acquires a port, it will not release it until the entire packet has been forwarded, and the second is that, a multicast packet that should be forwarded to multiple output ports is not forwarded unless all the output ports have been acquired. A flit-by-flit routing technique can be used to solve the problem. However, this technique comes with a new problem; because once we split the flits of a packet then we need to keep track of which flit belongs to which packet. We solve the new problem by inserting an additional ID tag field on each flit.

For a unicast message, the packet will have only one header flit, even if the size of the message is very large. Each packet consists of a header containing routing information or the address of the destination node and a few payload flits. The message is classified into three flit types i.e., header flit (Head), databody or payload data flit (DBod) and tail flit (Tail). Databody flit is identified as the payload data of a message/stream. Tail flit is used to mark the end of a message/stream. The header and ID consist of 3 bits each and data payload is of 32 bits. Flits belonging to the same message have the same local identity number (ID-tag) to differentiate it from other flits of different messages, when it passes through a communication link of the NoC. The ID-tag of the data flits of one message will vary over different communication links allowing different messages are interleaved each other at flit-level while being routed with wormhole switching.

REMOVING ROUTING-DEPENDENT DEADLOCK

A deadlock configuration is a situation where packet cannot be forwarded because of cyclic dependency. By prohibiting certain turns in a network, a routing algorithm might prevent deadlock altogether. The routing algorithm would have to prohibit at least one turn in each of the many possible cycles. At the same time, the algorithm would have to leave a path between every pair of nodes. In addition, it should not prohibit more turns than necessary; otherwise, the adaptiveness of the algorithm would be reduced.

To solve this problem of designing wormhole routing algorithms that are deadlock free and maximally adaptive for a network, we can use the turn model. The model involves analyzing the directions in which packets can turn in the network and the cycles that the turns can form. It prohibits just enough turns to break the entire cycles. The turn model is a partially adaptive routing algorithm, widely investigated for multi processor environments. Compared to fully adaptive routing algorithms, turn model algorithm is a partially adaptive algorithm because two turns out of eight are forbidden in order to avoid deadlock.

Adaptive West-First (WF) Routing Algorithm

The turn model is based on analyzing the directions in which packets can turn in a network and the cycles that the turns can form. Prohibiting just enough turns to break all of the cycles produces routing algorithms that are deadlock free, livelock free, minimal or non-minimal, and maximally adaptive for the network. In a 2-D mesh network, there are eight possible turns that can be used to route packets in the network. By committing only two prohibited turns, the partially adaptive routing algorithms can be performed. The dashed arrows denote the prohibited turns. The implemented routing function is minimal, i.e., a packet will not be routed away from its destination node. Therefore, the routing algorithms are also livelock-free.

In the West-First turn model, the prohibited turns are the two to the west. Therefore, to travel west, a packet must start out in that direction. This suggests the west-first routing algorithm. That is route a packet first west, if necessary, and then adaptively south, east, and north. Figure 3 represents the turn models of adaptive west-first (WF) algorithm.

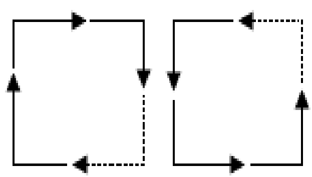

Figure 3: West-First Turn Model

A.Hold and Release Tagging Method

If more than one flit requests the same output port, the deadlock can occur. In order to avoid this Hold and Release Tagging Method is applied. If a multicast flits from an input port n has an N number of requests at any instant time t, then each single request to an output port m can be forwarded from the input port n to the output port m in the next time stage only if it receives a grant by an arbitration unit at input port n, while the other requests must be held in the input port if it is not granted by their requested output ports. This is shown in six snapshots of figure 6.

In each next time stage, a single request, which has been granted before, must be reset to prevent improper flit replication. This can be done with the help of arbitration unit. If all requests have been granted, then the multicast flit can be released from the queue in input port n. The flits A_1^0, B_1^0 and C_1^0 represent the flits with the local ID-tag 0. Snapshot 1 shows three multicast packets, i.e., A from EAST, B from WEST, and C from the SOUTH incoming ports, request different and the same outgoing links. Flits of all packets can be granted one by one as a winner to access the outgoing link at every stage as shown in Snapshot 2. In the next stage, as presented in Snapshot 3, all granted flits are accepted in the outgoing links.

In the next stage as shown in Snapshot 4, by using the flit-by-flit round arbitration method, NORTH and WEST outgoing ports change their selection to other flits, which also request these ports. In the next stage as presented in Snapshot *5*, flits A_1, B_2 and C_1 are transferred to the outgoing links, and can be released from EAST, WEST, and SOUTH input buffers respectively. Their request is now replaced by the requests of new incoming flits. Snapshot 6 shows generally the same mechanism with the situation shown in Snapshot 2.

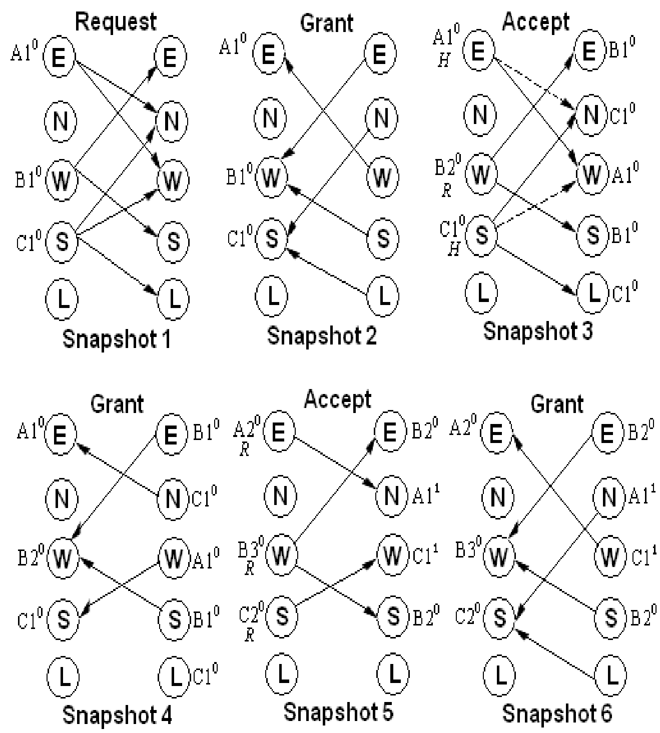

Figure 6: Hold and Release Tagging Method

REMOVING MESSAGE-DEPENDENT DEADLOCK

Even when the underlying network is designed to be free from routing-dependent deadlocks, the message-level deadlocks can block the network indefinitely, thereby affecting the proper system operation. A message-dependent deadlock occurs, if there are at least two kinds of messages; for example request and response.

In order to avoid message dependent deadlocks we can use separate logical networks for the different message types. This would ensure that the different message types do not share the network components, thereby guaranteeing freedom from message-dependent deadlocks. The most common method to achieve separate logical networks is the use of separate virtual channels for the different message types. For this each router input will need two virtual channels: one for the request messages and the other for the response messages. This separation of message types is maintained at all the routers in the network.

ROUTER ARCHITECTURE

The components of the router should be consistent with the techniques applying for deadlock-free routing. For this the east, north, west, south and local input-output ports of the router is designed with extendable components. The router consists of two main component groups, i.e. component groups at input and output ports. At every input port, there are a First-In First-Out (FIFO) buffer/queue and Routing Engine with Data Buffering (REB) components. The REB components consist of three modules, i.e. a Route Buffer, a Routing Engine (RE) and a Grant Controller (GC). The RE module consists of a Routing State Machine (RSM) unit and a Routing Reservation Table (RRT). Separate virtual channel is added to each router.

The *FIFO* Buffer is used to buffer data coming from a neighbor to the input of the NoC router. *RE* module is used to make a routing decision such that a message can be routed from an input port to an output port. The Routing Engines (RE) is a combination of a routing state machine (RSM) and a routing reservation table (RRT). When a header flit is coming from an input port n, a routing direction is computed by the RSM, and it is concurrently written in the slot number k in the RRT. When a databody or tail flit is coming to the input port n, then the routing direction is fetched directly from the slot number $k=ID$ in the RRT. The Route Buffer is used to buffer a message flit soon after the routing decision has been made for the flit. The GC module is a combinatorial logic used to control the data read operation of the FIFO buffer.

At each output port, there are two main modules, i.e. an Arbiter (A) unit and a Crossbar Multiplexor with ID Management (IDM) Unit (MIM). The Arbiter or Arbitration unit is used to select a message flit from an input port that will be switched out to the output port. The MIM module is used to multiplex message flits from input ports and concurrently used to update the ID-tag of each message and manages ID Slot table.

IDM-LUT Unit for ID-Based Routing

With the technique of ID based routing, flits of different messages can be interleaved in the same queue. This scheduling is done by the ID-tag mapping and management (IDM) unit. When a packet flit is switched to an outgoing port, the IDM unit will identify the local ID-tag and the flit type of a flit that flows through an outgoing link. If the flit is a header, the IDM unit will find a free local ID-slot to replace the old local ID-tag of the packet with a new one. In the next periods, whenever payload flits belonging to the packet are switched through the outgoing link, they will get the new local ID-tag from the ID-slot in the IDM unit by identifying their old local ID-tag and from which port they come. Then the routing direction of payload flits will be getting directly from the LUT unit. Figure 7 presents the IDM unit at the EAST outgoing port in node (3, 3) and ID updating. It shows also the LUT of the routing engine at the WEST incoming port in node (3, 4) and routing assignment.

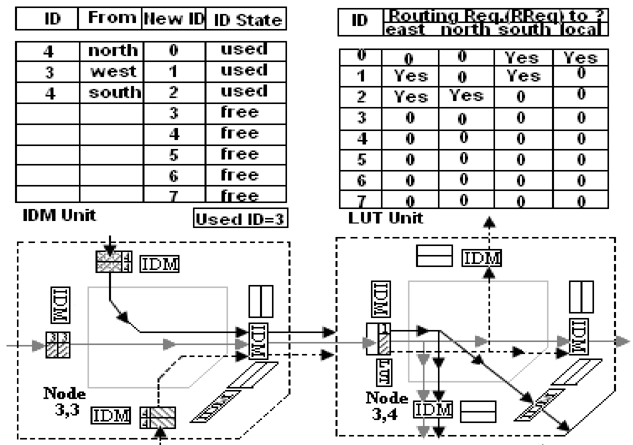

Figure 7: IDM-LUT Working

The new ID-slot is indexed based on the previous old local ID-tag and direction value from which the flit comes. Then the current ID-tag state is set from "free" to "use" state, and the number of used IDs (UIDs) is incremented. When a tail flit is passing through the outgoing port, then the related ID-tag state is set from "used" to "free" state, the UID is decremented, and the information related to the tail flit ID number is deleted from the ID Slot Table. Then the ID-slots of the IDM unit which have been used, is sent to router hardware logics in incoming ports for routing adaptivity.

The Arbiter Unit

It plays a very important role to implement the multicasting control and multiple data conflict management, which can guarantee the deadlock-free tree-based multicast routing. They are used to select a winner flit to access an outgoing link. A fair flit-by-flit round arbitration for multiple routing requests from all incoming ports is implemented in this module. If the full flag from the neighbor or NI is set, then the arbiter unit will not select a winner flit to access the related outgoing link of the neighbor or to access the input queue of the NI.

For instance, if two flits from different incoming ports request the same output port, then the two signals are set in the same cycle and sent to the same arbiter unit. In the next cycle, one of them will be granted by the arbiter unit and will reset it.

RESULTS AND DISCUSSIONS

In order to provide the flit level packet routing mechanism for NoC, at first, a router is designed with the required characteristics and then the packets are applied through it in a 4x4 mesh network, i.e., having 16 routers. It provides routing adaptivity and avoided routing-dependent deadlock using West-First routing algorithm and hold and release tagging method.

As shown in the fig. 8 initially we set the write signal and the packets are sending to north, south, west, east and local input ports of router and we are requesting the west input port. Hence data is stored as flits in the west FIFO buffer. Then data read signal is set and data body packets are send. The data packet consists of 3 bit type of header, databody or tail flit, ID of 3 bit and data payload of 32 bits. The data payload is now stored in the west route buffer. We are using the Xilinx ISE 8.1i synthesis tool and simulated using Model Sims 6.3f. Xilinx ISE 8.1i provides design entry and synthesis supporting VHDL or Verilog, place-and-route, completed verification and debug.

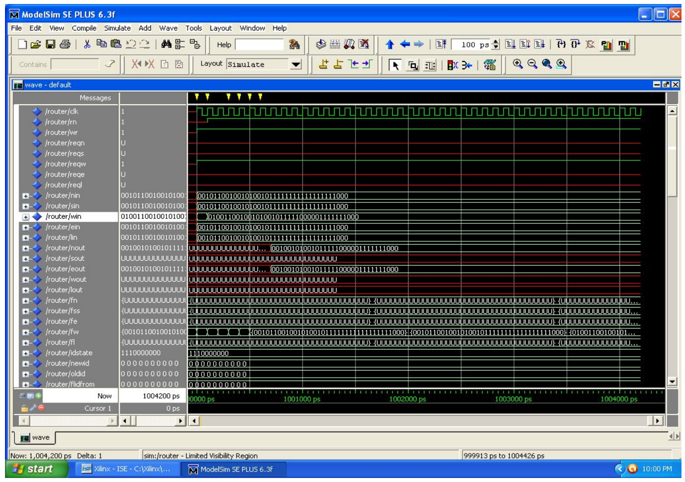

Figure 8: Send Packets to Five Input Ports and Data Body Flits Stored in FIFO Buffer of West Input Port by requesting it and Multicast Routing was Done to Two Output Ports

Then to provide adaptive routing, ID tag, source and destination address of packets are identified. From these offset values are calculated. This is done by taking the difference of source and destination address. These offset values are used in west-first algorithm to route packets adaptively to destination avoiding deadlock and it is shown in figure 9. Thus the multicast routing was done through the north and east output ports as shown in Figure 8.

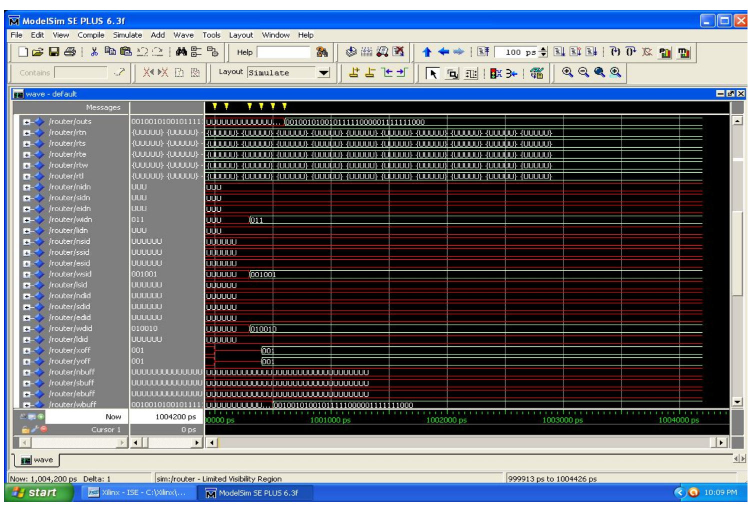

Figure 9: Identifying the Source and Destination Address and Calculating Offset Values to Route Packets to the Destination

CONCLUSIONS

We have presented the routing-dependent deadlock-free method for multicast routing with the help of turn models and ID-tag technique. The adaptive routing algorithms will avoid the cyclic dependencies and hold and release tagging method using flit-by-flit arbitration will avoid the deadlock due to more than one flit requesting the same output port at the same time. In order to avoid the message-dependent deadlocks, we have presented use of virtual channels to avoid the dependencies among the resources shared by different message types so as to make the network deadlock-free.

REFERENCES

1. *Ciprian Seiculescu, Srinivasan Murali, Luca Benini, Giovanni De Micheli,*"A Method to Remove Deadlocks in Networks-on-Chips with Wormhole Flow Control", *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010.*
2. *Everton Alceu Carara, Fernando Gehm Moraes*, "Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh networks-on-Chip", *IEEE Computer Society Annual Symposium on VLSI, 2008. ISVLSI '08.*
3. *Hosseingholi, M.; Sharif Ahmadian, A.; Sarbazi-Azad, H*,**"** Improving the Performance of Deadlock Recovery Based Routing in Irregular Mesh NoCs Using Added Mesh-Like Links", *IEEE International Symposium on **Circuits** and Systems (ISCAS), Proceedings of JUNE 2010.*
4. *Hsin-Chou Chi; Yu-Hong Jhang; Wen-Shu Chen*," Tree-Based Routing for Faulty On-Chip Networks with Mesh Topology ", *IEEE Computer Society Annual Symposium on VLSI (ISVLSI), JULY 2010.*
5. *L. Benini and G. D. Micheli,* "Networks on chips: A new SoC paradigm," *Computer, vol. 35, no. 1, pp. 70–78, 2002.*
6. *Masoumeh Ebrahimi, M. Daneshtalab, M. Neishaburi, M.H. Mohammadi, S. Afzali-Kusha, A. Plosila, J. Tenhunen*, "An Efficient Dynamic Multicast Routing Protocol for Distributing Traffic in NOCs", *Design, Automation & Test in Europe Conference & Exhibition, 2009.*
7. *Ruizhe Wu; Yi Wang; Dan Zhao,*" A Low-Cost Deadlock-free Design of Minimal-Table Rerouted XY-Routing for Irregular Wireless NoCs", *Fourth ACM/IEEE International Symposium on Networks-on-Chip (NOCS), MAY 2010.*
8. *Markovsky, Y.; Patel, Y.; Wawrzynek, J*," Using Adaptive Routing to Compensate for Performance Heterogeneity", *3rd ACM/IEEE International Symposium on Networks-on-Chip, MAY 2009.*
9. *Maurizio Palesi, Shashi Kumar, Rickard Holsmark, and Vincenzo Catania*, **"**Exploiting Communication Concurrency for Efficient Deadlock Free Routing in Reconfigurable NoC Platforms", *IEEE International Parallel and Distributed Processing Symposium, 2007.*
10. *Nickray, M.; Dehyadgari, M.; Afzali-kusha, A*,**"** Adaptive Routing Using Context-Aware Agents for Networks on Chips", *4th International Conference on Design and Test Workshop (IDT), NOV 2009.*
11. *Nishikawa, Y.; Koibuchi, M.; Matsutani, H.; Amano, H*," A Deadlock-free Non-minimal Fully Adaptive Routing Using Virtual Cut-through Switching", *IEEE Fifth International Conference on Networking, Architecture and Storage (NAS), JULY 2010.*
12. *Ruizhe Wu; Yi Wang; Dan Zhao,*" A Low-Cost Deadlock-free Design of Minimal-Table Rerouted XY-Routing for Irregular Wireless NoCs", *Fourth ACM/IEEE International Symposium on Networks-on-Chip (NOCS), MAY 2010.*

13. *Samman, F.A. Hollstein, T. Glesner, M.*, "Adaptive and Deadlock-Free Tree-Based Multicast Routing for Networks-on-Chip", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, July 2010.*
14. *Schonwald, T.; Zimmermann, J.; Bringmann, O.; Rosenstiel, W.* "Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures", *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, Aug 2007.*
15. *Shafiee, Ahmad M. Sadr, Parinaz Dehghani, Fatemeh* ,"Congestion Resistant Routing Algorithm for a Congestion Resistant Ring Road Network on Chip", *Fifth International Conference on Digital Telecommunications (ICDT), JUNE 2010.*
16. *Slaviˇsa Jovanovi´c, Camel Tanougast, Serge Weber, Christophe Bobda* "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections" *International Conference on Field Programmable Logic and Applications, SEP-2009.*
17. *Srinivasan Murali, Paolo Meloni, Federico Angiolini, David Atienza, Salvatore Carta, Luca Benini, Giovanni De Micheli, Luigi Raffo,* "Designing Message-Dependent Deadlock Free Networks on Chips for Application-Specific Systems on Chips", *International Conference on Very Large Scale Integration, 2006.*
18. *Umit Y. Ogras, Jingcao Hu, Radu Marculescu*, "Key Research Problems in NoC Design: A Holistic Perspective", *Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 2005.*
19. *Valinataj, M. Mohammadi, S. Plosila, J. Liljeberg*, "A Fault-Tolerant and Congestion-Aware Routing Algorithm for Networks-on-Chip", *IEEE 13th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), APRIL 2010.*
20. *W.J. Dally, B. Towles*, "Route packets, not wires: on-chip interconnection network", Proceedings of *Design Automation Conference, 2001.*
21. *Yan.S, Lin, B*," Joint multicast routing and network design optimization for networks-on-chip", *Computers & Digital Techniques, IET,SEP 2009*
22. *Yonghui Li; Huaxi GU; Peibo Xie; Keqiang He*; "Performance Modeling of Fully Adaptive Wormhole Routing in 2D-Mesh Network-on-Chip with MMPP (2) Input Traffic", *International Symposium on Information Science and Engineering, Dec 2008.*
23. *Zhonghai Lu, Bei Yin and Axel Jantsch*, "Connection-oriented Multicasting in Wormhole-switched Networks on Chip", *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, March 2006.*