Congestion Aware Routing Algorithm for NOC

Anshu Dhariwal[#1], **D.S.Gangwar**[#2]
[1] MTech Scholar, [2] Assistant Professor, [#] Faculty of Technology, Uttarakhand Technical University,
Dehradun, Uttarakhand, INDIA
[1]anshudhariwal.ec@gmail.com,
[2]dsgangwar@gmail.com

Abstract— **Network on chip is a communication subsystem on an integrated circuit. Network on chip has become a promising solution for multi-core architecture now a days as NOCs are able to scale communication links with the increasing number of cores in the system. In this paper we use adaptive routing algorithm for network on chip so that congestion in the network can be reduced considerably. Adaptive routing is used as it reduces the time for travelling the data between source and destination and circuit will respond accordingly faster. Furthermore, adaptivity requires a comprehensive, hardly intrusive, runtime, observability infrastructure, i.e., monitoring components, in order to gather data on the system.**

Keywords— **Adaptive Routing, Network on Chip, Verilog.**

I. INTRODUCTION

The routing algorithm is the main key factor of designing network on chip. The primary aim of using a routing algorithm in network is to achieve high performance. Different routing algorithms are used for removing the congestion in the network. Some of them are illustrated in this paper. The **Destination-Based Adaptive Routing** method (DBAR) is described under the assumption of an 8×8 mesh network. In this method, each switch has two 9-bit registers, called congestion-X and congestion-Y. These registers are used to store the congestion information received from the X and Y dimensions. Each entry in the congestion-X (congestion-Y) register is associated with one switch in that row (column) so that the register represents the statuses of all input buffers along that row (column). The congestion information of an input buffer is transferred through a wire to all the other switches along that row or column. In an 8×8 mesh network, eight congestion wires are used along each row and column. The DBAR method only considers the congestion value of the switches that are locate5d in the minimal path between source and destination switches.

In **Dynamic XY** (DyXY) algorithm, which is based on the static XY algorithm, a packet is sent either to the X or Y dimension depending on the congestion condition. It uses local information, which is the number of free buffer slots in an input buffer of the neighboring switches, to decide on the next hop. One of the drawbacks of DyXY is that the use of local information in the routing decision may forward a packet to a switch which is not locally congested while the surrounding regions of this switch are highly congested. Such non-optimal routing decisions increase the network latency in NoC.

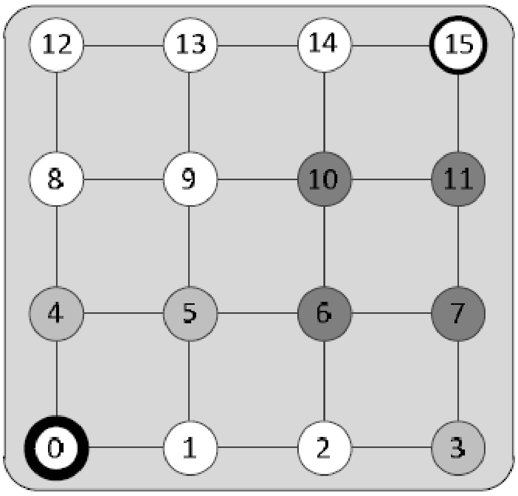

Fig. 1 DyXY Routing Algorithm

Figure 1 shows an example of the DyXY method where the routing decision based on local congestion information leads to deliver a packet through a congested region. In this figure, colors show the congestion level at each switch such that the darker color indicates the more congested switch. In this example, the switches 0 and 15 are the source and destination of the packet, respectively. In the DyXY method, the source switch 0 compares the occupied slots at the west input buffer of the switch 1 and that of the south input buffer of the switch 4. Since the switch 1 is less congested, the packet is sent to this switch. When the packet arrives at the switch 1, it has to be delivered through the switch 2 or 5. According to the congestion condition shown in Figure 1, the switch 2 is less congested and thus the packet is delivered to the switch 2. At the switch 2, the packet has to pass through the most congested region (i.e. the switches 6, 7, 10, and 11) in the network to reach the destination switch. This non-optimal path is as a result of making routing decisions based on local information. The packet could pass through the less congested region (i.e. the switches 8, 9, 12, and 13) if non-local information was considered.

In **the Enhanced Dynamic XY** (EDXY) method, the network is provided with auxiliary wires along each row and column to carry the buffer statuses located in that row or column. This information is propagated to the switches in the adjacent row and column. In this method, every switch first looks at the destination address of the packet. If the destination switch is not located in the adjacent row or column, the packet is routed similar to the DyXY method. However, if the destination address is just one hop apart from the switch in either the X or Y dimension, not only the queue length of the buffer in the neighboring switches are considered, but also congestion wires are used for the decision making.

Consider the example of Figure 2 where a packet is delivered from the source switch 0 to the destination 15 and it is already at the switch 2. Based on the DyXY method, since the switch 3 is less congested than the switch 6, the switch 3 is selected as the next hop. However, by this decision the packet has to pass the switches 7 and 11 which are highly congested. In contrast, in the EDXY method, in a similar situation (i.e. when a packet is at the switch 2), the congestion conditions of the third and fourth columns are compared together; since the third column is less congested, the packet is sent to the switch 6, thus avoiding packets to be routed through the highly congested switches.

In a similar example as in Figure 1, when the switches 6, 7, 10, and 11 are congested, EDXY behaves similar to DyXY and the packet has to be routed through the congested region due to the lack of global congestion information.

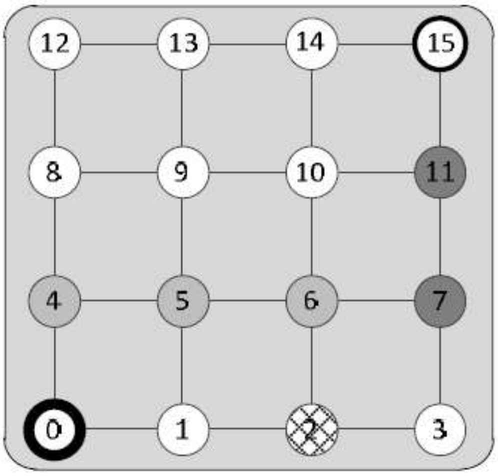

Fig.2 EDXY Routing Algorithm

Regional Congestion Awareness (RCA)

A well-known method called Regional Congestion Awareness (RCA-quadrant) is proposed to utilize non-local congestion information in the routing decision. In the RCA method, in order to collect global congestion value in switches, the locally computed congestion value of the corresponding input buffers of a switch is combined with those global signals propagated from upstream switches and the newly-aggregated value is transmitted to the downstream switches and so on. In this method, non-local congestion information is used to send a packet through a less congested direction. Even though RCA collects global information, in fact this information cannot be efficiently used in routing decisions.

Figure 3 shows an example of the RCA method where the switch 0 wants to communicate with the switch 15. Firstly, the packet should be sent to either the switch 1 or the switch 4 depending on global congestion information received from X and Y dimensions. According to the RCA method, the congestion value in the Y dimension is calculated by weighting sum (i.e. 50-50 weighting of local and propagated congestion values) of the values of the corresponding input buffers of all switches above the first row. Similarly, for the X dimension, the congestion value is determined by aggregating the values of corresponding input buffers of all switches except the first column. As can be seen in Figure 3 both calculated congestion values in the X and Y dimensions contain the congestion information of several common switches, i.e. 5, 6, 7, 9, 10, 11, 13, 14, and 15.

In as much as the comparison is made between the values in X and Y dimensions and both values include the congestion information of common switches, only the values of the switches in the row 0 and column 0 have a strong impact on routing decisions. Another shortcoming of the RCA method is that it considers the statuses of all switches along each direction whether they are located in the minimal path or not. Therefore, this method introduces excessive information, which may degrade performance, especially when traffic is mostly local.

According to the above explanation, the packet is sent to the X dimension from the source switch since the congestion statuses of the switches 1, 2, and 3 are less than those of the switches 4, 8, and 12. The congestion statuses of the other switches (i.e. 5, 6, 7, 9, 10, 11, 13, 14 and 15) are compared with each other, thus not affecting the comparison result considerably. On the other hand, for an instance, the routing decision is the same wherever the destination is located in the northeast position of the source switch (i.e. the switches 5, 6, 7, 9, 10, 11, 13, 14, and 15). Therefore, when the destination is at the switch 5, the routing unit not only compares the congestion values of the switches 1 and 4, but also considers the congestion values of all the other switches which are not located between the source and destination switches.

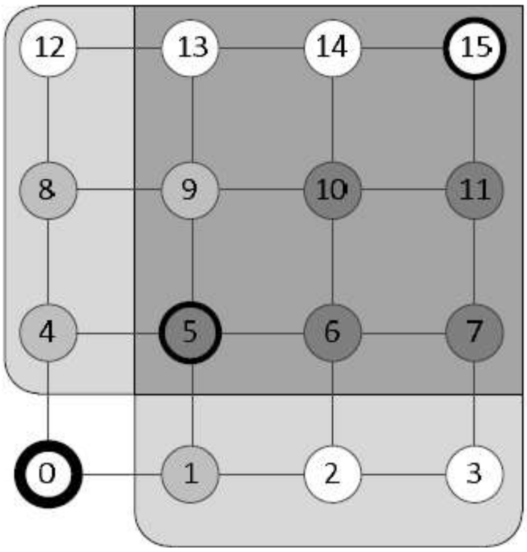

Fig.3 Regional Congestion Awareness

II. SIMULATION RESULTS

First we force clock, then apply reset =1, after that give enable value to 1then we put enable flag at high level & then we give the data in write data, we run this program and the data will be store in write data then we reset the write enable i.e., put wr_en = 0 & give the high value to rd_en i.e., rd_en =1. Now the data will store in rd_data variable. Write pointer and read pointer will show the status of how many times read & write operation will occur. Count will show the no. of counts, fifo-full, fifo-empty & memory are the internal variables.

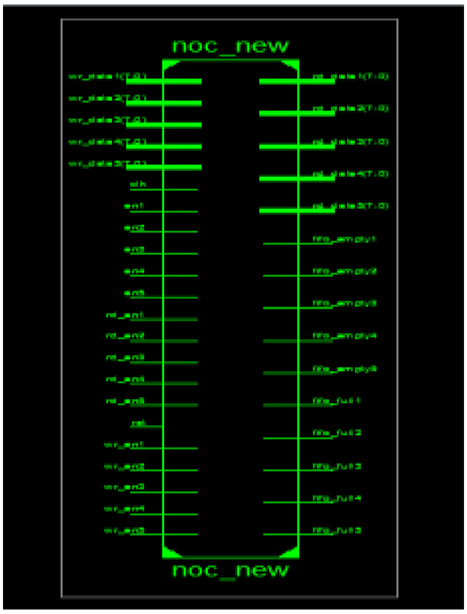

Fig. RTL of the NOC Router

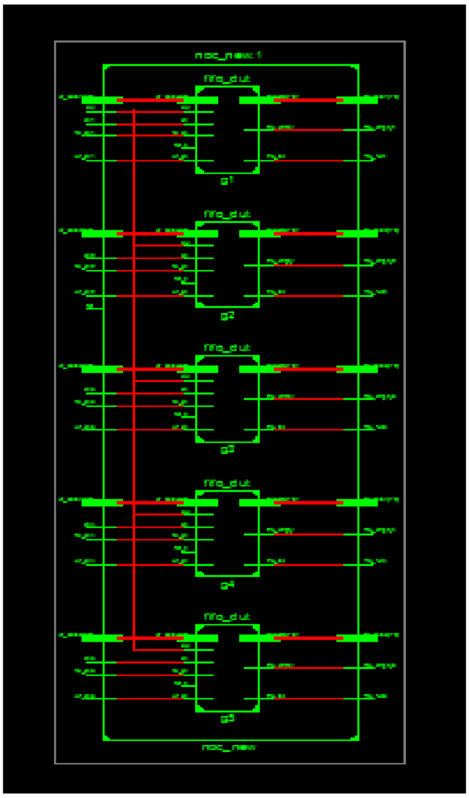

Fig. Expanded RTL of the NOC Router

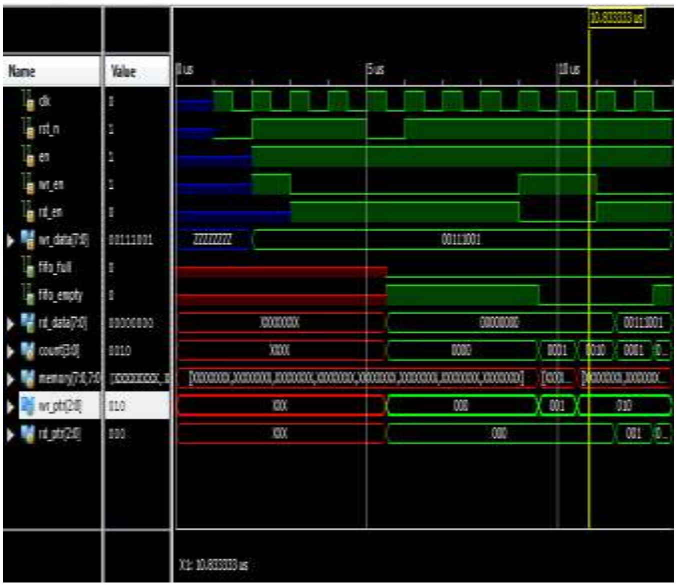

Fig. Output of the Router

III. Conclusions

In this paper adaptive routing algorithm is used which is helpful in minimizing the congestion of the network. This become feasible as congestion of the path is predetermined in the routing algorithm so we can deviate the path of the data or information to the other path with the help of which congestion is minimized or removed completely.

The another aim of this paper is to design and implement fault tolerant adaptive routing algorithm characterised by a unique combination of routing which is use to make a network deadlock and livelock free.

References

- Rabab Ezz-Eldin, Magdy A. El-Moursy, and Hesham F. A. Hamed, "Process Variation Delay and Congestion Aware Routing Algorithm for Asynchronous NoC Design", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2015.
- Mohammad Abdullah Al Faruque, Thomas Ebi, Jorg Henkel, "AdNoC: Runtime Adaptive Network-on-Chip Architecture", IEEE Transactions On Very Large SCALE Integration (VLSI) Systems, 2015.
- Parag Parandkar, Jayesh kumar Dalal and Sumant Katiyal, "Performance Comparison of XY, OE and DY Ad Routing Algorithm by Load Variation Analysis of 2-Dimensional Mesh Topology Based Network-on-Chip", BIJIT - BVICAM's International Journal of Information Technology, Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi, January 2012.
- Hamed S. Kia, and Cristinel Ababei, "A New Fault-tolerant and Congestion-aware Adaptive Routing Algorithm for Regular Networks-on-Chip".
- Somashekhar, S Rekha, "Design of a 5 Port Router for Noc Using Verilog", IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163, pISSN: 2321-7308.
- Masoud Seyedmohammadzadeh, "A Distributed Adaptive Routing Algorithm for Regular Networks-on-Chip", International Journal of Scientific & Engineering Research, Volume 5, Issue 6, June-2014 ISSN 2229-5518.
- Liang Wang, Xiaohang Wang, Terrence Mak, "Dynamic Programming-Based Lifetime Aware Adaptive Routing Algorithm for Network-on-Chip".
- Sheng Ma, Natalie Enright Jerger, Zhiying Wang, "Whole Packet Forwarding: Efficient Design of Fully Adaptive Routing Algorithms for Networks-on-Chip", 18th IEEE International Symposium on High Performance Computer Architecture (HPCA- 2012).
- Zaheer Ahmed, " Fault Adaptive Routing", Haupt- Seminar, Reliable Network-on- Chip in Many Core- Era, June 2009.
- Danny Ly, " Design of a Network- on- Chip Adaptive Routing Algorithm for Multi- Core Computing", March 2013.
- Leonel Tedesco, Thiago Rosa, Febien Clermidy, Ney Calazans, Fernando Gehm Moraes, " Implementation and Evaluation of a Congestion Aware Routing Algorithm for Network – on- Chip, ResearchGate, January 2010.