



A Deadlock free Algorithm for NoC based System Using Distance Class Approach

Sudeep Ghosh ,Mandira Banik

*Gurunanak Institute of Technology,
Kolkata, West Bengal, India*

Abstract— Similar to the interconnection architectures, deadlock occurs in NoC architectures when a group of packets are unable to make progress because they are waiting on one another to release resources. Despite its effectiveness in reducing buffer requirements, wormhole-switching technique is susceptible to deadlock. In wormhole switching, the resources are virtual channel and their corresponding buffers. Here, cyclic dependency on flit buffers occurs among several packets. As a result, the network is clogged with packets without making any progress. To prevent this catastrophic problem, we mainly adopt deadlock-avoidance policy in NoCsim. The user can provide a deadlock-free algorithm for the network or use our methods of eliminating cycles in the resource dependence curve—*distance class approach* [13]. Distance class approach is particularly useful for the networks for which deterministic algorithms succumb to deadlock. We Implemented it NoC simulator [1].

Keywords—system-on-chip; network-on-chip; buffer depth; odd-even routing algorithm

I. INTRODUCTION.

A SoC consists of several system components, for example processors, programmable logic, or memories. These system components are known as Resources. Today's SoCs use a means of communication, which may become problematic in the future. This is why the NoC[1,2] concept is being developed. The NoC concept comprises many different ideas, something like IP, memories, and how to connect various parts of a SoC. Among these various NoC concepts distinction is made between Resources, which do the computation, and the Network, which handles the communication. 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 [1]-[5]. The infrastructure of network tail determines system performance and cost. The First-In-First-Out (FIFO) buffer depth of tail's input channel is one of the key design problems [6]. The input channel buffer depth that is at each tail in the NoC has a serious impact on the overall area. At the same time, depending on the network workload, designers can reduce the network latency by orders of magnitude with the method of increasing the buffer size. Therefore, it needs to be optimized when developers design a special NoC. In this paper we have given an overview of NoC design approach in section two

at first. And then we have described our proposed algorithm.. At last, conclusions are provided.

II. OVERVIEW OF DESIGN APPROACH

NoC is means of communications among different resources which is built up by following components:

- Switch
- Resource Block (called Intellectual Property (IP).
- Resource Network Interface (RNI)
- Dedicated wire for communications

The IP cooperate in performing one or several tasks. In the OSI model, IP implement the Application Layer. All IPs are connected to a network which consists of Network Interfaces and Switches. Above figure shows how the various parts are interconnected.

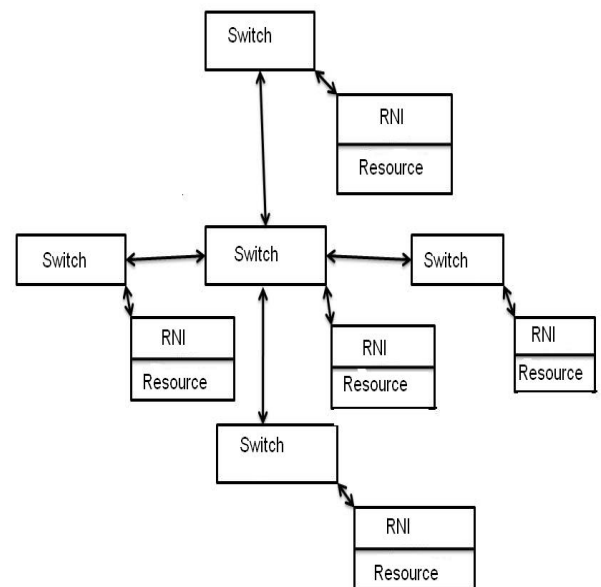


Figure 1. Block diagram of Noc

The Network Interface provides networking services to an IP. The layers that are implemented in the Network Interface

are the presentation, session, and transport layers. Some of the responsibilities of the network layer may be taken care of here as well.

Each Network Interface is connected to a Switch, which itself is connected to a number of other Switches. The responsibility of the Switches is to deliver packets from the source to its destination. One of the problems with future silicon technologies is that it becomes hard to distribute one clock throughout the entire chip to work all the components synchronously.

Global wires are the physical implementation of the communication channels. The trend toward faster and lower power communication may decrease reliability as an unfortunate side effect. In NoC design approach, physical layer design finds a compromise between competing quality metrics and provide a clean and complete abstraction of channel characteristics to above micro-network layers [2]. The data-link layer abstracts the physical layer as an unreliable digital link. The main purpose of data-link protocols is to increase the reliability of the link up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable [3].

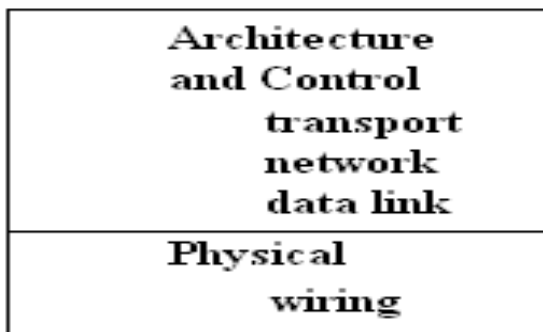


Figure 2. The Protocol Stack from which the stack paradigm of network on chip can be adapted.

At the data-link layer, a key task is to achieve the specified communication reliability level [5]. At the network layer, packet data transmission can be customized by the choice of switching and routing algorithms. The former (e.g., circuit, packet, and cut-through switching) establishes the type of connection and the latter determine the path followed by a message through the network to its final destination. At the network layer, the communication abstraction is an end-to-end connection.

The transport layer decomposes messages into packets at the source. It also decides the order of packets and reassembles the messages at the destination. Another transport-layer task is flow control. Flow control can mitigate the effect of congestion by regulating the amount of data that enters the network, at the price of some throughput penalty [6].

III. ODD-EVEN ROUTING ALGORITHM

An odd-even routing is a adaptive algorithm used in dynamically adaptive and deterministic (DyAD) Network on Chip. The odd-even routing is a deadlock free turn model which prohibits turns from east to north and from east to south at tiles located in even columns and turns from north to west and south to west at tiles located in odd columns. The DyAD system uses the minimal odd-even routing which reduces energy consumption and also removes the possibility of livelock.

In a two-dimension mesh with dimensions $X*Y$ each node is identified by its coordinate (x, y) . In this model, a column is called even if its x dimension element is even numerical column. Also, a column is called odd if its x dimension element is an odd number. A turn involves a 90-degree change of traveling direction. A turn is a 90-degree turn in the following description. There are eight types of turns, according to the traveling directions of the associated channels. A turn is called an ES turn if it involves a change of direction from East to South. Similarly, we can define the other seven types of turns, namely EN, WS, WN, SE, SW, NE, and NW turns, where E, W, S, and N indicate East, West, South, and North, respectively. As a whole, there are two main theorems in odd-even algorithm:

Theorem1: NO packet is permitted to do EN turn in each node which is located on an even column. Also, No packet is permitted to do NW turn in each node that is located on an odd column.

Theorem 2: NO packet is permitted to do ES turn in each node that is in an even column. Also, no packet is permitted to do SW turn in each node which is in an odd column.

Algorithm OE

```

/*Source router: (Sx,Sy);destination router: (Dx,Dy); current
router: (Cx,Cy).*/
begin
avail_dimension_set<-empty;
Ex<-Dx-Cx;
Ey<-Dy-Cy;
if (Ex=0 && Ey=0) //current router is destination
return C;
if (Ex=0){ //current router in same column as
//destination
if (Ey<0)
add S to avail_dimension_set;
else
add N to avail_dimension_set;
}
else{
if (Ex>0){ //eastbound messages
if (Ey=0){ //current in same row as destination
add E to avail_demision_set;
}
else{

```

```

if(Cx % 2 != 0 or Cx=Sx) //N/S turn allowed only in
odd//column.
if(Ey < 0)
add S to avail_dimension_set;
else
add N to avail_dimension_set;
if(Dx% 2 != 0 or Ex != 1) {

//allow to go E only if destination
// is odd column
add E to avail_dimension_set;
//because N/S turn not allowed in even column
}
}
else { // westbound messages
add W to avail_dimension_set;
if(Cx%2=0) //allow to go N/S only in even
//column, because N->W and S->W
//not allowed in odd column
if(Ey<0)
add S to avail_dimension_set;
else
add N to avail_dimension_set;
}
}
select a dimension from avail_dimension_set to forward the
packet.
End
    
```

OE routing algorithm is a complex routing algorithm. However, it is one kind of adaptive routing algorithm. For a pair of source and destination, it can provide a group of routing paths and it can prevent from dead lock appearance. Compared to XY routing algorithm, it fits better to 2-dimension mesh topology NoC with Constant Bit Rate (CBR) traffic condition [11].

IV. ARCHITECTURE 2-DIMENSION MESH TOPOLOGY

A mesh-shaped network consists of m columns and n rows. The routers are situated in the intersections of two wires and the computational resources are near routers. Addresses of routers and resources can be easily defined as x-y coordinates in mesh. Regular mesh network is also called as Manhattan Street network.

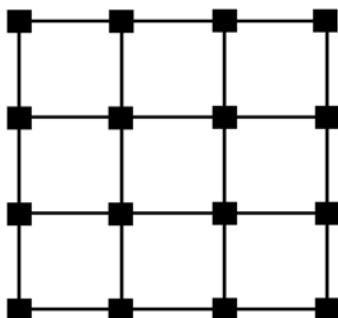


Figure 3: Mesh network.

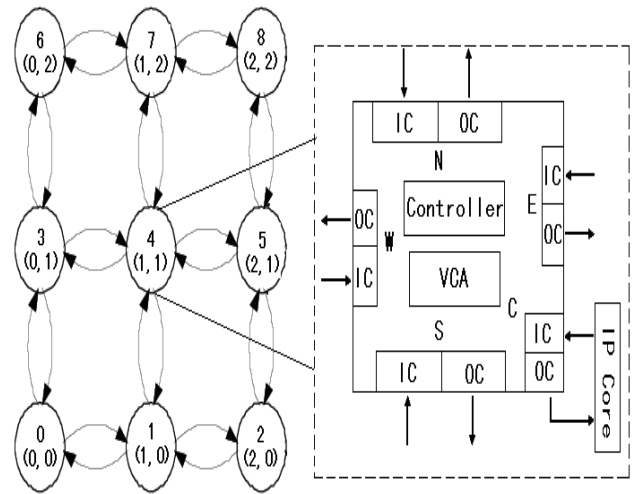


Figure 4. The Architecture of 2-Dimension 3X3 Mesh Topology Network on Chip

Fig. 4 is the graph of a 2-Dimension 3X3 mesh topology NoC example, in which each circle represents a tile of the network. A typical tile is connected to neighbor tiles by four bidirectional channels (N, E, S and W). Each tile is identified by a unique integer ID. Also, each tile can be identified by a pair x-coordinate and y-coordinate.

The major components of each tile are Input Channel Controller (IC), Output Channel Controller (OC), Controller, Virtual Channel Allocator (VCA) and IP core (Fig. 4):
 Input Channel Controller (IC): Each tile consists of one IC for each neighbor, and an IC for IP core. For example, a tile having 4 neighbors (one in each of the directions North (N), South (S), East (E) and West (W)) will have 5 ICs: N IC, S IC, E IC, W IC and core IC. Each IC consists of one virtual channel (VC). Each VC consists of a FIFO buffer.

Output Channel Controller (OC): Each tile consists of one OC for each neighbor, and an OC for IP core.

Controller: Each tile consists of one Controller which implements router to service routing requests from all ICs.

Virtual Channel Allocator (VCA): Each tile consists of one VCA that services requests for virtual channel allocation from all ICs.

IP core: Each tile consists of an IP core (IP element) to which an application or traffic generator can be attached.

V. DEADLOCK PREVENTION

Similar to the interconnection architectures, deadlock occurs in NoC architectures when a group of packets are unable to make progress because they are waiting on one another to release resources. Despite its effectiveness in reducing buffer requirements, wormhole-switching technique is susceptible to deadlock. In wormhole switching, the resources are virtual channel and their corresponding buffers. Here, cyclic dependency on flit buffers occurs among several packets. As a result, the network is clogged with packets without making any progress. To prevent this catastrophic problem, we mainly adopt deadlock-avoidance policy in NoC. The user

can provide a deadlock-free algorithm for the network or use our methods of eliminating cycles in the resource dependence curve—distance class approach [13]. Distance class approach is particularly useful for the networks for which deterministic algorithms succumb to deadlock.

Distance class is an approach to group the virtual channel and the associated buffers into numbered classes and restrict allocation of resources so that packets acquire resources in ascending order. In this approach, a packet at distance i from its source is allocated virtual channel from class i . Initially from the source node to the parent switch, the packet takes virtual channel 0. Then to transfer to the next switch, it takes virtual channel 1. With this system, a packet holding an input buffer from an input virtual channel waits on an output buffer on higher numbered output virtual channel. Since, there is no dependency on the lower or same numbered virtual channels, no cycle in the resource dependence graph can occur; hence deadlock cannot occur. Although, this approach is a general way to order resources, the number of virtual channels per physical link is proportional to the diameter of the network. For some networks, we can take advantage of the topology to reduce the number of virtual channel and associated buffer classes significantly.

Distance class is an approach to group the virtual channel and the associated buffers into numbered classes and restrict allocation of resources so that packets acquire resources in ascending order. In this approach, a packet at distance i from its source is allocated virtual channel from class i . Initially from the source node to the parent switch, the packet takes virtual channel 0. Then to transfer to the next switch, it takes virtual channel 1. With this system, a packet holding an input buffer from an input virtual channel waits on an output buffer on higher numbered output virtual channel. Since, there is no dependency on the lower or same numbered virtual channels, no cycle in the resource dependence graph can occur; hence deadlock cannot occur. Although, this approach is a general way to order resources, the number of virtual channels per physical link is proportional to the diameter of the network. For some networks, we can take advantage of the topology to reduce the number of virtual channel and associated buffer classes significantly.

Figure 5 gives a concrete example of distance classes, where a network using buffer classes based on distance. From switch i , a flit in an input buffer $\text{Bin-}i,k$ moves to an output buffer $\text{Bout-}i,k+1$ to reach the input buffer $\text{Bin-}i+1,k+1$ of the switch $i+1$. In this figure, none of the packets flowing between switches is generated in the children nodes of these four switches, otherwise there would have been flow between $\text{Bin-}i,k$ to $\text{Bout-}i,k$.

The *Distance Class* approach guarantees that the deadlock won't occur in a topology even if the routing algorithm does not ensure deadlock avoidance. This approach is more useful for the some architecture for which the packets inherent fall into the prey of circular dependence. Since any deterministic algorithm would fail to balk the disastrous situation, the *Distance Class* approach will be able to come as a remedy for the topology at the cost of throughput.

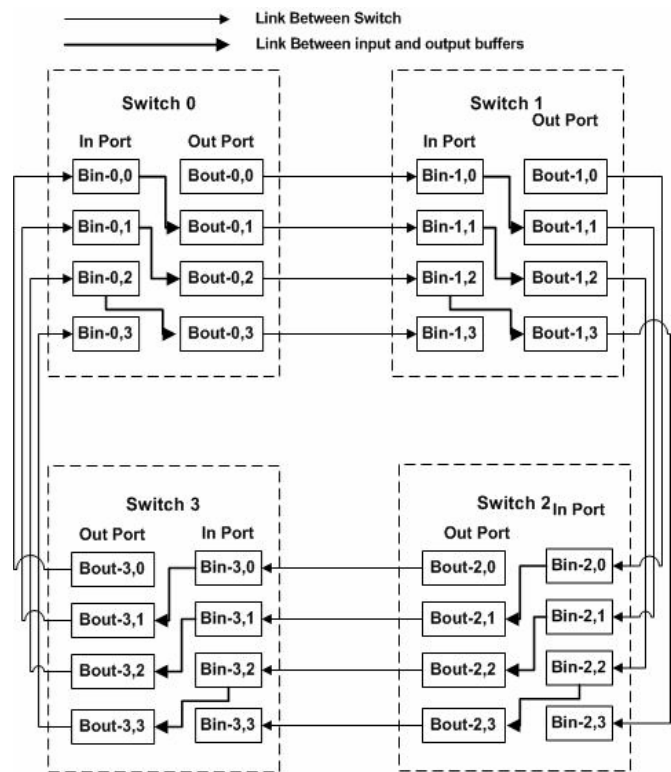


Figure 5: Implementation of Distance Class

VI. SIMULATION RESULT ANALYSIS

For our evaluation purposes we have developed a model of 7x7 mesh topology NoC with regions. We have implemented wormhole switching with a packet size of 10 flits. Every router has two flit input and one flit output buffer. The router can simultaneously route packets destined to non-conflicting output ports. The minimal link delay is three cycles / flit and the maximum link bandwidth is 0.5 flits / cycle (1 packet / 20 cycles).

The following parameters were used to study the performance of a NoC platform. Performance values were collected over 60 000 packets, after a warm-up session of 30 000 packets.

- **Average Latency:** The average delay of a packet from source (when the header leaves) to the destination (when the tail has reached).
- **Blocked Routing Cycles/Router:** The total number of routing cycles when packets were blocked in a router. Latency values were averaged over 5 random traffic scenarios to get an overall view about how the performance in the network is affected by changes in network configuration and packet injection rate. Blocked Routing Cycles/Router can give information where the network is most congested. We have compared our simulation result with the work[4]. Here we are providing the comparison graph of avg_throughput vs num_Buffer with work[4]. This graph shows that our proposed scheme produce the better throughput than the work[4].

BUFFER DEPTH (LEVEL)	AVERAGE THROUGHPUT FOR EACH CHANNEL (GBPS)	
	Result of work[4]	Result of Our proposed Scheme
1	8.7328	8.5673
2	9.4494	8.5796
3	9.4912	9.5232
4	9.5209	9.5541
5	9.5338	9.5594
6	9.5712	9.5869
7	9.5712	9.6979
8	9.5712	9.7877
9	9.5712	9.7878
10	9.5712	9.8174
11	9.5712	9.8174
12	9.5712	9.8275
13	9.5712	9.8278
14	9.5712	9.8278
15	9.5712	9.8278
16	9.5712	9.8278

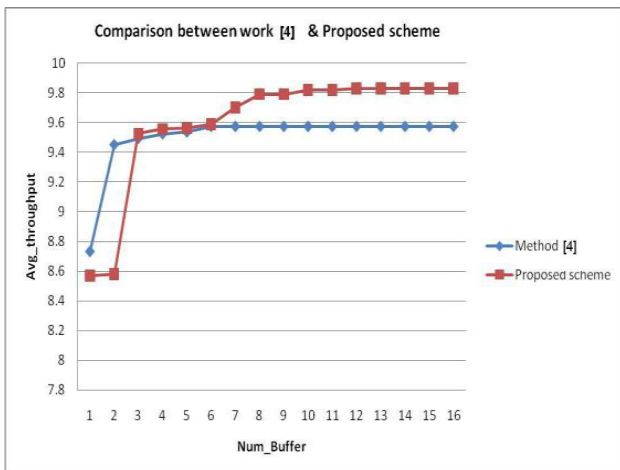


Figure 6: Comparison graph between [4] and ours proposed scheme

The above figure depicts the comparison graph between work[61] and our proposed scheme .We can state the fact that at a given number of buffer our scheme gives enhanced average throughput than work[4]. The Reason is work [4] have not used any deadlock avoidance technique as a result their deadlock occur. But in our experiment we have used distance vector technique to avoid deadlock i.e circular dependency.

VII. CONCLUSION

In this paper we have discussed only to prevent circular dependency to avoid deadlock. We have not considered congested route. We have not consider region based technique. the area of a NoC router required by the APSRA based algorithm is expected to be larger than the router for the other algorithm. This is because APSRA requires tables within each router to store routing information, whereas the other algorithm can be implemented as an optimized FSM. The table based implementation of the APSRA based algorithms could also be a blessing because it allows configurability (and even dynamic re-configurability) of routing algorithms to efficiently handle modifications in communication requirements in the running applications. Future developments will mainly address the definition of design space exploration strategies to optimally determine region placement, shape, and number of access points.

REFERENCES

- [1] Sudeep Ghosh,Hafizur Rahaman,"A General Purpose Simulator For Network-on-chips", Journal of Information Systems and Communication (JISC), Volume 3 issue 1,2012 pp-174-179.
- [2] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg, etal, "Network on a chip: an architecture for billion transistor era." Proc.IEEE NorChip, 2009.
- [3] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," Proc. Design Automation Conference, Jun.2005, pp. 684-689.
- [4] Wang Zhang, Wuchen Wu, Lei Zuo, Xiaohong Peng "The Buffer Depth Analysis of 2-Dimension Mesh Topology Network-on-Chip With Odd-Even Routing Algorithm"
- [5] S. Kumar, A. Jantsch, J. Soinenen, M. Forsell, M. Millberg, J. Oberg . etal, "A network on chip architecture and design methodology," Proc.IEEE Computer Society Annual Symposium on VLSI, Apr. 2002, pp.105-112, doi: 10.1109/ISVLSI.2002.1016885.
- [6] R. Marculescu, H. Jingcao, U. Y. Ogras, "Key research problems in NoC design: a holistic perspective," Proc. Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'05) Sept. 2009, pp.69-74, doi:10.1145/1084834.1084856.
- [7] D. Wiklund, S. Sathe, D. Liu, "Network on Chip Simulations for Benchmarking,"iwsoc, pp. 269-274, System-on-Chip for Real-Time Applications, 4th IEEE International Workshop on (IWSOC'04), 2004
- [8] J. Duato, S. Yalamanchili, L. Ni, Interconnection networks: an engineering approach, Morgan Kaufmann, Revised Edition, 2002.
- [9] Z. Lu, A. Jantsch. "Traffic Configuration for Evaluating Networks on Chips". iwsoc,pp. 535-540, Fifth International Workshop on System-on-Chip for Real-Time Applications (IW-SOC'05), 2005.
- [10] L. Jain. NIRGAM Version 1.1 Manual, Sep 2007, pp. 1-2.
- [11] Z. Wang, H. Ligang, W. Jinhui, G. Shuqin, W. Wuchen. "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip", 2009 Global Congress on Intelligent Systems (GCIS09), submitted for publication.
- [12] Luca. Benini, Giovanni De Micheli. Networks on chips: technology and tools. Elsevier Morgan Kaufmann, Boston, 2006.
- [13] W. J. Dally, and B. Towels, "Principles and Practices of Interconnection Networks", Morgan Kaufman Publishers, 2004