

A Novel Routing Algorithm for On-Chip Communication in NoC on Diametrical 2D Mesh Interconnection Architecture

Prasun Ghosal¹ and Tuhin Subhra Das²

¹ Department of Information Technology
Bengal Engineering and Science University, Shibpur
Howrah 711103, WB, India

² Purabi Das School of Information Technology
Bengal Engineering and Science University, Shibpur
Howrah 711103, WB, India
prasun@ieee.org, tuhinbcrec@gmail.com

Abstract. To meet the design productivity, scalability and signal integrity challenges of next-generation system designs, a structured and scalable interconnection architecture, network on chip (NoC), has been proposed recently to solve the complex on-chip communication problem. In an NoC, the communication among different nodes is achieved by routing packets through a pre-designed network fabric according to some architecture rather than wires. In Diametrical 2D Mesh architecture, advantages of routing include smaller network diameter as well as larger bisection width. Again, fast network access requires a well-constructed optimal routing algorithm. Our proposed routing algorithm ensures that the packet will always reach the destination through the shortest path and it is deadlock free.

Keywords: NoC routing, Diametrical 2D mesh routing, On-chip communication.

1 Introduction

Network-On-Chip(NoC) is a new paradigm used for making inter-connections between IP cores on a system-on-chip(SoC) [4]. It brings notational improvement over conventional Bus and Crossbar interconnection. In NoC [2], system module such as processor, memories and specialized IP exchange data using network as a public transportation sub-system for the information traffic.

In NoC, a pre-designed network-fabric consisting of routers and links connecting them in a definite topology is used. The intellectual property (IP) cores are connected to the routers using Network Interfaces(NIs). The cores exchange messages between themselves through this on-chip network. Network Interfaces(NIs) modules are put as interface between an IP core and the corresponding router. This on-chip network architecture plays a very significant role in the performance of the overall NoC.

A high level of parallelism is achieved in NoC, because all links in the NoC can operate simultaneously on different data packets. Therefore, as the complexity of integrated systems keeps growing, a NoC provides enhanced performance (such as throughput, power consumption) and scalability in comparison with previous communication architectures (e.g., dedicated point-to-point signal wires, shared buses, or segmented buses with bridges).

1.1 Routing on NOC

In the network design of NoC the most essential things are network topology and its related routing algorithm. This on-chip network should have both low latency and high bandwidth communication fabrics that efficiently support a diverse variety of workloads. The algorithms must be designed in such a way that they offer large parallelism and can hence utilize the potential of NoC.

Routing on NoC is similar to routing on any network. A routing algorithm determines how the data is routed from sender end to receiver end. In NoC, routing algorithm should be implemented by simple logic and require size of data buffers should be minimal. Network topologies [8] [9] [10] and properties may be application specific [11].

1.2 General Problem on NoC Routing

There are couples of requirements that every Network on Chip implementation has to achieve. Performance requirements are small latency [5] [12], guaranteed throughput [12], path diversity, sufficient transfer capacity and low power consumption [6]. Architectural requirements are scalability, generality and programmability. Fault and distraction tolerance as well as valid operation are major on Quality of Service.

However, selecting an appropriate topology [8] [9] [10] is one of the most critical decisions in the design of an interconnection network because it is related to some performance metrics such as the network's zero load latency and its capacity [7], and directly influences implementation costs, both in terms of the on-chip resources and implementation complexity.

2 Background and Motivation

2.1 Related Works

2D Mesh is a very popular well known topology use in Network on Chip due to its facility in implementation, simplicity of the XY routing strategy and the network scalability. But, 2D Mesh has some disadvantages such as long network diameter as well as energy inefficiency because of the extra hops. Due to this, in this work, we use a proposed novel NoC topology called Diametrical 2D Mesh [1]. Proposed architecture reduces the network diameter [8] [12] 50 percent compared to 2D mesh by using 8 extra bypass diametrical channel. It reduces the network average latency [12] and power consumption [6] considerably.

2.2 Drawbacks in Existing Works

But the existing algorithm called Extended XY Routing algorithm [1] have some drawbacks like sometimes the packet never reaches to its destination (because of livelock) and sometime packet follows XY path though there exists a diametrical path. So, existing algorithm is not able to totally exploit the benefit of diametrical path. But our proposed deterministic routing ensures that the packet will always reach the destination through the shortest path and it is deadlock free [13] as no circular waiting path is generated from the proposed routing algorithm.

3 Problem Description

Assume a Diametrical 2D Mesh consisting of 25 IP cores (see Figure 1(b)) having 5 nodes in each row and each column. We illustrate two cases, where in case 1 message never reaches to its destination and in case 2 message is not able to use the shortest path though there exists a shortest path.

Case 1: If we consider a packet whose current node is (3,0) and destination node is (4,4), then according to the existing Extended XY algorithm [1] packet will move through diametrical path not follow simple XY routing path and it will reach to node (0,3). As $X_{offset} = (4 - 3) = 1$ (using $X_{offset} = X_{dest} - X_{current}$) and $Y_{offset} = (4 - 0) = 4$ (using $Y_{offset} = Y_{dest} - Y_{current}$) and $d=5$ (using $d = \text{square root of } 25$) and so it does not meet the necessary criteria for XY Routing as described in [1], which is $X_{offset} + Y_{offset} \leq (d - 1)$, rather it will follow diametric path and will reach to node (0,3). At the node (0,3) $X_{offset} = (4 - 0) = 4$ and $Y_{offset} = (4 - 3) = 1$, so packet at this node packet will again follow diametric path as condition $X_{offset} + Y_{offset} \leq d - 1$ becomes false and will reach to node (3,0). Thus a livelock will appear and the packet will never reach to destination node (4,4).

Case 2: Again, if we consider the current node is (3,0) and destination node is (0,4), then $X_{offset} = (0 - 3) = -3$ and $Y_{offset} = (4 - 0) = 4$ and $d = 5$ (using $d = \text{square root of } 25$) and according to the existing extended XY algorithm [1] packet will move through XY path as $X_{offset} + Y_{offset} = 1$ which is less than $(d-1)$, where $d=5$. But it should follow the existing diametrical path which will carry the packet to the node (0,3) in terms of utilizing proper facility of diametrical path and to reduce the number of hop count.

4 Proposed Modified Extended XY Routing

We propose a Modified extended XY routing for Diametrical 2D mesh Topology. This routing algorithm is very similar to Extended XY routing and XY routing, Modified Extended XY is very simple due to simple addressing scheme and

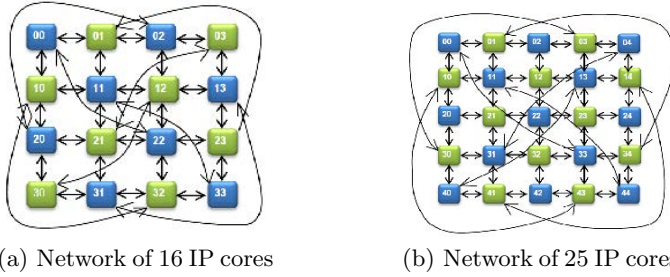


Fig. 1. Diametrical 2D Mesh network

structural topology. Algorithm 1 shows the details of the routing pseudo code. We define X_{diff} and Y_{diff} values in the pseudo code, which are calculated as follows respectively:

$$X_{diff} = \text{difference between } (x_{dest}, y_{current}) \tag{1}$$

$$Y_{diff} = \text{difference between } (y_{dest}, y_{current}) \tag{2}$$

$$d = \sqrt{\text{number of nodes}} \tag{3}$$

where,

x_{dest} =X coordinate of destination node

y_{dest} =Y coordinate of destination node

x_{curr} =X coordinate of current node

y_{curr} =Y coordinate of current node

X_{diff} and Y_{diff} are the values indicating the number of rows and columns between a current and a destination node respectively. If both X_{diff} and Y_{diff} values are zero, it means that the current node is the destination and the packet reaches the destination node. The flow of Modified Extended XY routing is described as given below:

- If a current node and a destination node are located in the same row or column or $(X_{diff} + Y_{diff}) < (d - 1)$
 - Then conventional XY routing is performed (condition 3 of pseudo code).
- Else
 - If X_{diff} and Y_{diff} both are greater than or equal to $diff$ (condition 5 of pseudo code).
 - * Then flits are forwarded by diameter channel;
 - Else
 - * If X_{diff} is less than $diff$ (condition 7 of pseudo code)
 - Then depending on some conditions as stated in pseudo code under condition 7 either X_{shift} routing or Y_{shift} routing or diameter channel is used for routing.
 - * Else (condition 8 of pseudo code)
 - Then depending on some conditions as stated in pseudo code under condition 8 either X_{shift} routing or diameter channel is used for routing.

In the proposed algorithms, the following notations have been used.

x+ indicates movement of flit in positive direction of x axis (i.e. in eastward direction)

x- indicates movement of flit in negative direction of x axis (i.e. in westward direction)

y+ indicates movement of flit in positive direction of y axis (i.e. in northward direction)

y- indicates movement of flit in negative direction of y axis (i.e. in southward direction). Also, in the pseudo code represented in Algorithm 1, $C < \# >$ represents the condition numbers.

4.1 Illustration of Algorithm with Example

We are discussing here the flow of the proposed algorithm with few cases.

Case1: If current node of packet is (3,0) and destination node is (4,4) of Fig1(b). Then according to our proposed algorithm $X_{diff} = 4 - 3 = 1$ and $Y_{diff} = 4 - 0 = 4$. Now as condition 3 (i.e. $X_{diff} + Y_{diff} < (d - 1)$ for $d=5$) in pseudo code becomes false. It check for condition 5 which is not true as $X_{diff} < diff$ (where $diff=2+\text{number of rows not having any diametrical connection} = 2+1=3$). Now condition 7 becomes true as $X_{diff} < diff$ is true. As x_{dest} is equal to $d-1$, so condition 7a becomes true and a Y_{shift} routing will take place which will move packet to node (2,0). At this node again a Y_{shift} routing will take place because of above reason and packet will move to node (1,0). Now $X_{diff} = 4 - 1 = 3$ and $Y_{diff} = 4 - 0 = 4$. So condition 5 becomes true as value of $diff = 3$ and packet will follow the diametrical path to move node (4,3). At node (4,3) packet will flow simple XY routing and will reach to node (4,4). (see Table 2 Sl. no 8)

Case2: If current node of packet is (3,0) and destination node is (0,4) of Fig1(b). Then according to our proposed algorithm $X_{diff} = 3 - 0 = 3$ and $Y_{diff} = 4 - 0 = 4$. So condition 3 (i.e. $X_{diff} + Y_{diff} < (d - 1)$ for $d = 5$) in pseudo code becomes false. Now condition 5 is true as both X_{diff} and Y_{diff} have value which is greater than or equal to $diff$ (as $diff=3$ here). So packet will follow diametrical path and hence move to node (0,3). At this node $X_{diff} = 0 - 0 = 0$ and $Y_{diff} = 4 - 3 = 1$. So condition 3 becomes true and packet will follow simple XY routing and will reach to node (0,4). (see Table 2 Sl. no 7)

5 Experimental Results

The proposed algorithm is implemented in a standard desktop environment running Linux operating system on a chipset with Intel Pentium processor running at 3 GHz using GNU GCC compiler. The important section of a sample run is shown below.

```

Input : Diametric 2D matrix of nodes,  $x_{dest}, y_{dest}, x_{curr}, y_{curr}$ 
Output: Optimum routing path between source and destination nodes

 $X_{diff}$ =difference between( $x_{dest}, x_{curr}$ );
 $Y_{diff}$ =difference between( $y_{dest}, y_{curr}$ );
 $X_{offset} = x_{dest} - x_{curr}$ ;
 $Y_{offset} = y_{dest} - y_{curr}$ ;
d=size of diametric matrix having same X and Y size. /*(e.g. for  $5 \times 5$  matrix size is 5)*/;
if  $d \geq 4$  then
    |  $diff=2$ +no of rows not having diametrical connection;
end
else
    |  $diff=1$ ;
end
C1: if  $X_{diff} = 0$  and  $Y_{diff} = 0$  then
    | channel=internal;
end
C2: else
    C3: if  $X_{diff} = 0$  or  $Y_{diff} = 0$  or  $X_{diff} + Y_{diff} < d - 1$  then
        | XY routing;
    end
    C4: else
        C5: if  $X_{diff} \geq diff$  and  $Y_{diff} \geq diff$  then
            | channel = diameter;
        end
        C6: else
            C7: if  $X_{diff} < diff$  then
                C7a: if  $x_{dest} = 0$  or  $x_{dest} = (d - 1)$  then
                    Yshift Routing;
                C7b: else
                    | if ( $x_{curr} = 0$  or  $x_{curr} = 1$  or  $x_{curr} = (d - 2)$  or  $x_{curr} =$ 
                    | ( $d - 1$ ) and ( $y_{curr} = 0$  or  $y_{curr} = 1$  or  $y_{curr} =$ 
                    | ( $d - 2$ ) or  $y_{curr} = (d - 1)$ ) then
                    | | channel=diameter;
                    end
                end
                C7c: else
                    | if  $x_{curr} = 0$  or  $x_{curr} = (d - 1)$  then
                    | | Xshift Routing;
                    end
                end
                C7d: else
                    | Yshift Routing;
                end
            end
            C8: else
                C8a: if  $y_{dest} = 0$  or  $y_{dest} = (d - 1)$  then
                    Xshift Routing;
                C8b: else
                    | if ( $x_{curr} = 0$  or  $x_{curr} = 1$  or  $x_{curr} = (d - 2)$  or  $x_{curr} =$ 
                    | ( $d - 1$ ) and ( $y_{curr} = 0$  or  $y_{curr} = 1$  or  $y_{curr} =$ 
                    | ( $d - 2$ ) or  $y_{curr} = (d - 1)$ ) then
                    | | channel = diameter;
                    end
                end
                C8c: else
                    | Xshift Routing;
                end
            end
        end
    end
end
end
end
end

```

Algorithm 1: Algorithm for proposed routing in diametrical 2D mesh NoC

```

XY_route();


---


if  $Y_{diff} \neq 0$  then
  | if  $Y_{offset} > 0$  then
  | | channel=x+;
  | | else
  | | | channel=x-;
  | | end
  | end
  | else
  | | if  $X_{offset} > 0$  then
  | | | channel=y-;
  | | | else
  | | | | channel=y+;
  | | | end
  | | end
  | end
end

```

Algorithm 2: Procedure for XY routing

```

Xshift_route();


---


if  $Y_{offset} > 0$  then
  | channel=x-;
  | else
  | | channel=x+;
  | end
end

```

Algorithm 3: Procedure for X shift routing

```

Yshift_route();


---


if  $X_{offset} > 0$  then
  | channel=y+;
  | else
  | | channel=y-;
  | end
end

```

Algorithm 4: Procedure for Y shift routing

```
[student@localhost26 ~]$ gcc diametric_algo.c
[student@localhost26 ~]$ ./a.out
give size of diametric matrix
5
give value of xdest 4
give value of ydest 4
give value of xcurr 0
give value of ycurr 0
current node is(0 0)
current Xdiff is 4
current Ydiff is 4
After diametr routing current node is (3 3)
current Xdiff is 1
current Ydiff is 1
After xy routing current node is(3 4)
current Xdiff is 1
current Ydiff is 0
After xy routing current node is(4 4)
current Xdiff is 0
current Ydiff is 0
Packet has reached the detination
```

Table 1. A comparative study of few Interconnection parameter between D-2D Mesh and 2D Mesh. $d=No\ of\ nodes\ in\ each\ side\ of\ Diametrical\ 2D\ Mesh$, $D-2D= Diametrical\ 2D\ Mesh$.

Parameters ↓	No of IP cores →	4 IP cores d=2	9 IP cores d=3	16 IP cores d=4	25 IP cores d=5	36 IP cores d=6	49 IP cores d=7
Number of links	D-2D Mesh	6	20	32	48	68	92
	2D Mesh	4	12	24	40	60	84
D-2D Mesh link Redundancy		33%	30%	25%	16%	11%	8%
Diameter	D-2D Mesh	1	2	3	4	5	6
	2D Mesh	2	4	6	8	10	12
D-2D Mesh Diameter Improvements		50%	50%	50%	50%	50%	50%

Table 2. Routing path with required hope count of a matrix size 5. $d=diametrical\ routing$, $xy=simple\ xy\ routing$, $Yshift=Yshift\ routing\ of\ pseudo\ code$.

SL No.	# Cores	Matrix Size(d)	Src node	Dest node	Routing Path	Hop count
1	25	5	00	23	00-d→33-xy→23	2
2	25	5	00	04	00-xy→01- xy →02- xy →03- xy →04	4
3	25	5	00	40	00- xy →10- xy →20- xy →30- xy →40	4
4	25	5	00	14	00-d→33- xy→34- xy →24- xy →14	4
5	25	5	21	04	21-yshift→31- d →04	2
6	25	5	21	44	21- yshift →11- d →44	2
7	25	5	30	04	30- d→03- xy →04	2
8	25	5	30	44	30- yshift→20- yshift →10- d→43- xy →44	4
9	25	5	44	00	44- d→11- xy→10- xy→00	3
10	25	5	44	20	44- d→11- xy →10- xy →20	3

Table 3. Shows required no of hop count to route from source 00 to different destination node of matrix size $d=2,3, \dots, 9$.

SL No.	# Cores	Matrix Size(d)	Src node	Dest node	Hop count
1	4	2	00	11	1
2	4	2	00	10	1
3	9	3	00	11	1
4	9	3	00	20	2
5	9	3	00	22	2
6	16	4	00	11	2
7	16	4	00	03	3
8	16	4	00	13	3
9	16	4	00	33	3
10	25	5	00	22	2
11	25	5	00	04	4
12	25	5	00	44	3
13	36	6	00	23	4
14	36	6	00	33	3
15	36	6	00	05	5
16	36	6	00	55	3
17	49	7	00	34	2
18	49	7	00	44	3
19	49	7	00	06	6
20	49	7	00	66	3
21	64	8	00	77	3
22	64	8	00	34	6
23	64	8	00	07	7
24	81	9	00	88	3
25	81	9	00	03	3
26	81	9	00	43	7
27	81	9	00	08	8

Table 1 shows a comparison of few important parameters between Diametrical 2D Mesh and simple 2D Mesh and from the above table it is clear that number of extra links in Diametrical 2D Mesh does not grow by the growth of IP cores. Number of extra links constantly equals to 8. This link redundancy is decreasing gradually by the growth of IP cores. For example, the link redundancy in 16 IP cores 2D Mesh is 1/3 and in 25 IP cores 2D Mesh, this ratio is 1/5. Furthermore, the network diameter is decreased to 50% with any number of IP cores in comparison to 2D Mesh. This leads to less average hop count and network average latency and power consumption considerably.

Table 2 shows different routing path with number of required hop count for different source and destination node pair of a matrix size $d = 5$ (where, $d = \sqrt{\text{Total number of nodes}}$). It clearly shows that required number of hop count for any path does not exceed 4 (i.e. in general case $(d-1)$), which is very less and required time is also negligible(required routing time < 0.005 sec).

6 Conclusion

Diametrical 2D Mesh was proposed to cope up with the disadvantages of 2D Mesh architecture such as large network diameter and high power consumption. But the algorithm proposed in [1] was not able to fully utilize the benefit of Diametrical 2D Mesh and in some cases the packet is not reaching to its destination node. In our proposed algorithm those issues have been solved and the proposed deterministic algorithm is able to fully exploit the benefit of Diametrical 2D Mesh architecture. Our proposed algorithm is a deterministic one. So, future works may be to make it adaptive in order to address the issues like network congestion and faulty network issues.

References

1. Reshadi, M., Khademzadeh, A., Reza, A., Bahmani, M.: A Novel Mesh Architecture for On-Chip Networks. D & R Industry Articles, <http://www.design-reuse.com/articles/23347/on-chip-network.html>
2. A survey of Architectural Design and Implementation Tradeoffs in Network on Chip System, <http://www.danmarconett.com/downloads/NoCSurveyPaper.pdf/>
3. A comparison of Network-on-Chip and Buses, <http://www.arteris.com/nocwhitepaper.pdf/>
4. Benini, L., De Micheli, G.: Networks on Chips: A new SoC Paradigm. *IEEE Computer*, 70–78 (2003)
5. Michelogiannakis, G.: Approaching Ideal NoC Latency with Pre-Configured Routes, Master's Thesis, Department of Computer Science, University of Crete
6. Kaisari, A.E., Rahmati, D., Sarbazi-Azad, H., Hessabi, S.: A Markovian Performance Model for Network-on-Chip. In: *Proceedings 16th Euromicro Conference on Parallel, Distributed and Network-Based Proceedings*, pp. 157–164 (2008)
7. Dally, W.J., Towles, B.: *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, San Francisco (2004)
8. Kundu, S., Chattopadhyay, S.: Network-on-chip architecture design based on mesh-of-tree deterministic routing topology. *International Journal of High Performance Systems Architecture* 1(3), 163–182 (2008)
9. Ezhumalai, P., Chilambuchelvan, A., Arun, C.: Novel NoC Topology Construction for High-Performance Communications, <http://www.hindawi.com/journals/jcnc/2011/405697/>
10. Benini, L., De Micheli, G.: *Networks on Chips: Technology and Tools*. Morgan Kaufmann, San Francisco (2006)
11. Palesi, M., Holsmark, R., Kumar, S., Catania, V.: APSRA: A methodology for design of application specific routing algorithms for NoC systems, Technical Report DIIT-TR-01-060406, Dip. di Ingegneria Informatica e delle Telecomunicazioni, Univ. di Catania (2006)
12. Kundu, S., Dasari, R.P., Chattopadhyay, S., Manna, K.: Mesh-of-Tree Based Scalable Network-on-Chip Architecture. In: *IEEE Region 10 and the Third International Conference on Industrial and Information Systems, ICIIS 2008*, pp. 1–6 (2008)
13. Kundu, S., Chattopadhyay, S.: Mesh-of-Tree Deterministic Routing for Network-on-Chip Architecture. In: *ACM Great Lake Symposium on VLSI (GLSVLSI)*, pp. 343–346 (2008)