A 2D topology for Network-on-Chip

Pankaj Choudhury1, Delphin Bampire2

1, 2Department of Computer Science & Engineering, Assam University, Silchar, Assam, India, E-mail: 1alleypach@gmail.com, 2badelly2020@yahoo.fr

Keywords: Network-on-chip; 2D topology; performance; scalability; load balance;

Abstract

The numbers of intellectual properties (IP’s) are increasing in a single chip day by day. Earlier shared bus based communication system was used between the IPs for intercommunication. But due to increase in number of IPs it becomes infeasible. A new communication paradigm called Network-On-Chip was introduced to fulfill the growing demand of electronics system design which integrates a large number of computational resources into a single chip, with the aim to get more efficiency and high performances. In this paper we proposed a new NoC topology, which is scalable in two dimensions (X and Y). In our proposed topology we introduced diagonal connections through which we try to reduce the number of hop counts and latency, which is not present in existing two-dimensional topologies. The proposed topology is described with a packet based adaptive routing algorithm, which is dead lock and live lock free. The routing algorithm provides proper load balancing, path diversity and maximum link utilization. Our proposed topology has minimum latency and high throughput than existing topologies. The proposed topology is simulated and tested using open source simulator named OMNET++.

1 Introduction

System on chip (SoC) is a communication paradigm which was used in earlier chip design, where IPs viz CPU, memory, DSPs were interconnected using a shared bus [7]. As we know day-by-day the size of a chip is reducing and number of IP cores inside a chip are growing rapidly. Since the shared bus systems have limitation in terms of scalability, bandwidth, latency, throughput etc. Researcher introduced a new interconnection approach known as Network on Chip to overcome the limitation of SoC [2].

The Network on Chip is like a general purpose networks where clients are connected using router and communication channel. But the main difference is, here clients are computational resources viz CPU, memory, DSP etc. and the network is inside a chip.

There are many topologies that had been proposed viz mesh, butterfly, fat tree, ring etc. and this topologies had been tested using different routing algorithm, but there are still lots of problems exists in terms of efficiency, scalability, throughput.

In this paper we propose a new highly scalable topology for Network on Chip along with a suitable routing algorithm.

The remaining part of this paper is organized as follows, section II describes related works. In section III, a brief idea about proposed topology and routing algorithm. In section IV simulation results is compared and performance is analysed. Finally in section V describes future work and concludes this paper.

2 Related work

There are already some popular topologies present. For example star, ring, mesh, torus, binary tree, butterfly, and fat tree. However among all these, fat tree, butterfly and mesh show some interesting results in scalability and performance than star or ring.

A mesh based topology is proposed by S. Kumar et al. in [8]. The IPs and cores are placed in M X N matrix form, which takes a grid like structure as shown in Figure 1.

![Figure 1: Mesh Topology [6]](image)

There each switch is connecting to four neighbouring switches and one IP core. It is highly scalable in both X and Y direction. But as the size of the network increases, the network diameter and latency also increase. Clients that are present in the edges of the network have very high latency.

To reduce the high latency between end switches in mesh, torus is proposed. In torus end switches are joined by wrapped around wire as shown below in Figure 2.
But torus has some disadvantages. As the network size increases the length of end-to-end connecting wire also increases, which can lead to excessive delay.

A tree based topology is proposed by Charles E. Leiserson in [5] known as fat tree. Fat tree is highly scalable and the most efficient network topology in terms of scalability and performance. Also it is efficient in terms of hardware use. As shown in Figure 3 clients are in the leaf node. The main disadvantage of fat tree is that it uses large number of switches for few clients and these switches consume most of the chip area. Again bandwidth requirement increase when going up closer to the root from leaf.

A butterfly topology [9] is a multistage network on chip topology, which uses MIN structure of a fat tree, where clients are placed in both end of the network and switches are placed in middle. It contains double the number of clients using same number of switches as fat tree and reduces chip area consumption. In butterfly topology routing is unidirectional, so it uses deterministic routing. The disadvantage of butterfly topology [3] is that it does not have path diversity and uses long wirings.

3 Proposed work

As we have already seen that existing topologies have many limitations. We have proposed a topology which we name as Topology A in this text to overcome the scalability problem. the proposed topology i.e. “Topology A” although shows promising results but is also not free from other problems such as network congestion at gateway routers as explained in the later sections. To solve the congestion problem found in our proposed topology A we proposed a modifies version of “Topology A” which we name here in this text as “Topology B”. In our proposed topologies we tried to solve problems about scalability, multiple path diversity, latency etc.

3.1 Proposed Topology A

The main advantages of proposed topology are the path diversity and it reduces the conflicts in the network as it increases the number of paths from one node to another. Below figure shows the basic structure (a single group) of the proposed topology.

The proposed topology have another significant advantage, that is clients are divided into group of 8 and within a group there exist direct communication channel between routers connected to clients. In one group for a packet to reach its destination must travel at least maximum two hops. The topology has been designed in such a way that it can be scalable in both X and Y direction and number of clients can vary from 8 to 16, 24, 32, 48, 64 and so on. Below figure describe how proposed topology is scalable in both X and Y direction with 32 clients.
3.2 Proposed Topology B

Proposed topology is the modified version of “topology A”. The basic structure of the proposed topology B is borrowed from topology A. In proposed topology A there is only one gateway router present as shown in Figure 7. Because of that if the topology is extended more, gateway routers will get heavily congested. For example router number 7 in figure 7.

In our proposed topology we try to reduce the congestion by adding more channel as shown in figure 8. There are 2 advantage of adding more channels, firstly if one channel fails other can still work, which gives reliable communication and secondly reduces congestion in gateway router by balancing the load.

In our proposed topology we introduced diagonal connection to reduce hop count and latency as shown in figure 9. Our proposed topology is highly scalable in both X and Y direction like a mesh topology. The number of clients can be increased from 8, 32, 72, up to 128. Topology forms 2X2, 3X3 and 4X4 matrices of groups. A single group contains 8 clients with 8 routers. We have expanded the topology for 8, 32, 72, 128 clients. As we have discussed earlier the proposed topology B is scalable in both X and Y direction, figure 9 describes it for 32 clients.

3.3 Router design

The router design for both of the topology is same. There are 2 types of routers (based on their usage), one that connects the clients and another connects between groups known as gateway router. But both routers are homogeneous and have 10 ports. Routers are assigned with a unique router ID for identification. Client routers have odd numbers 1, 3, 5, 7... and so on as router ID. Gateway routers have even numbers 2, 4, 6, 8... and so on as router ID. Again ports are interconnected using gates. All ports have three modules namely scheduler, inport, opcal[4].

The scheduler [4] module only forwards packets. Scheduler checks whether the packet has come from internal gate or external gate connecting client and other router. If the packet comes from internal gate scheduler directly forwards the packet to proper output gate. And if the packet comes from external gate, scheduler forwards the packet to opcal via inport for routing calculation.

Inport [4] work as a mediator between scheduler and opcal. It only forward packets between opcal and scheduler.

Opical [4] execute the necessary part of routing algorithm. It calculates the out port through which the packet need to send. After the routing decision opical send packet with output port number to scheduler via inport.

3.4 Proposed routing algorithm for topology A

The proposed topology is tested with a new adaptive routing algorithm. The proposed routing algorithm is described below.

destination_ID: Destination client ID.
destination_group_ID: Destination clients group ID.
destination_router_ID: Destination router ID where the destination client is attached.
current_group_ID: Current routers group ID, which currently have the packet.
current_router_ID: Current router ID, which currently have the packet.
column_no: As the topology is scalable in X and Y direction it forms a grid, column_no is how many numbers of column in the grid.

Input: New packet from external router or client.
Output: Desired output port number through which packets needs to be send.

Steps:
1) Preprocessing:
   a) Receive the packet and get the destination_ID
   b) Calculate destination_group_ID and destination_router_ID using following methods
      i) destination_group_ID = destination_ID/8
      ii) temp = destination_ID – 8*(destination_group_ID)
         If temp = 0 or 1 then
         destination_router_ID = 0
         If temp = 2 or 3 then
         destination_router_ID = 2
         If temp = 4 or 5 then
         destination_router_ID = 1
         If temp = 6 or 7 then
         destination_router_ID = 3

2) Depending on the destination_group_ID, destination_router_ID and destination_ID do the following:
   a) If current_group_ID = destination_group_ID then do the following:
      i) If current_router_ID = destination_router_ID
         Then deliver the packet to the destination client
      ii) If current_router_ID ≠ destination_router_ID
         Then forward the packet to destination router through the desired port
   b) If current_group_ID ≠ destination_group_ID then do the following:
      i) If current_group_ID/column_no = destination_group_ID/column_no
         And current_group_ID < destination_group_ID, then forward the packet to right side of the current group through the desired port
      ii) If current_group_ID/column_no = destination_group_ID/column_no
         And current_group_ID > destination_group_ID, then forward the packet to left side of the current group through the desired port
   iii) If current_group_ID/column_no ≠ destination_group_ID/column_no
         And current_group_ID/column_no < destination_group_ID/column_no then
         forward the packet above the current group through the desired port
   iv) If current_group_ID/column_no ≠ destination_group_ID/column_no
         And current_group_ID/column_no > destination_group_ID/column_no then
         forward the packet below the current group through the desired port

3.5 Proposed routing algorithm for topology B

The proposed routing algorithm is an adaptive routing algorithm, which is different from Topology A's routing algorithm. The routing algorithm is described below.

destination id = dest_id
destination router id = dest_router_id
current router id = curr_router_id
current group id = curr_group_id
destination group id = dest_group_id
current group column no = curr_col_no
current group row no = curr_row_no
destination group column no = dest_col_no
destination group row no = dest_row_no

RADIUS is no of column present in the topology.

Step 1.
Receive the packet and get the destination_ID from the packet

Step 2.
Calculate dest_router_id, dest_group_id, curr_col_no, curr_row_no, dest_col_no, dest_row_no

dest_group_id = dest_id/8
IF dest_id%2 = 0
   dest_router_id = dest_id + 1
ELSE
   dest_router_id = dest_id
curr_col_no = curr_group_id%radius
curr_row_no = curr_group_id/radius
dest_col_no = dest_group_id%radius
dest_row_no = dest_group_id/radius
Step 3.
IF dest_group_id = curr_group_id
    destination group is in the current group

IF dest_router_id = curr_router_id
    destination is attached to current router.

Step 4.
IF dest_group_id ≠ curr_group_id AND curr_router_id%2!=0
    destination is in other group, current router is a client router, send the packet to a gateway router.

Step 5.
IF dest_group_id ≠ curr_group_id AND curr_router_id%2=0
    destination is in other group, current router is a gateway router

IF curr_col_no<dest_col_no AND curr_row_no<dest_row_no
    destination is in straight downward direction from current group

IF curr_col_no<dest_col_no AND curr_row_no>dest_row_no
    destination is in straight upward direction from current group

IF curr_col_no<dest_col_no AND curr_row_no=dest_row_no
    destination is straight rightward direction from current group

IF curr_col_no>dest_col_no AND curr_row_no=dest_row_no
    destination is straight leftward direction from current group

IF curr_col_no<dest_col_no AND curr_row_no<dest_row_no
    destination is in diagonally down rightward direction from current group

IF curr_col_no<dest_col_no AND curr_row_no>dest_row_no
    destination is in diagonally up rightward direction from current group

IF curr_col_no>dest_col_no AND curr_row_no<dest_row_no
    destination is in diagonally down leftward direction from current group

IF curr_col_no>dest_col_no AND curr_row_no>dest_row_no
    destination is in diagonally up leftward direction from current group

3.5 Packet structure
The topologies interprets messages as packets of size 20 Bytes. Packet has 5 fields; each field is of size 4 Bytes. Those Fields are as follows.

<table>
<thead>
<tr>
<th>packetID</th>
<th>sourceID</th>
<th>destinationID</th>
<th>outputline</th>
<th>data</th>
</tr>
</thead>
</table>

After calculation of routing decision opcal put output port number in output line field.

4 Simulation results and performance analysis
In this section we have shown a comparative analysis of both of the proposed topologies. The simulation and results collection is done in open source simulator OMNET++.

4.1 End to end delay
The End to end latency is calculated by the difference between the creation time of a message and arrival time of that message in the destination. Here table 1 describes the average end to end latency of packets that are send to random destination. Table 2 show result of end to end latency when packets are send to a single destination.

Table 1: Average end to end delay for random destination

<table>
<thead>
<tr>
<th>No of packet</th>
<th>Average end to end delay in 10⁻⁹ for random destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed Topology A</td>
</tr>
<tr>
<td>8</td>
<td>58.625</td>
</tr>
<tr>
<td>32</td>
<td>100.5</td>
</tr>
<tr>
<td>72</td>
<td>155.5833333</td>
</tr>
<tr>
<td>128</td>
<td>181.171875</td>
</tr>
</tbody>
</table>

Figure 10: Average end to end delay for random destination
### Table 2: Average end to end delay for static destination

<table>
<thead>
<tr>
<th>No of packet</th>
<th>Average end to end delay in $10^9$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed topology A</td>
</tr>
<tr>
<td>8</td>
<td>161</td>
</tr>
<tr>
<td>32</td>
<td>3339.156</td>
</tr>
<tr>
<td>72</td>
<td>4779.36</td>
</tr>
<tr>
<td>128</td>
<td>1113328</td>
</tr>
</tbody>
</table>

Figure 11: Average end to end delay for Static destination

4.2 Path diversity

Both the proposed topologies have high path diversity. Again because of multiple path packets get transfer even if there is a link failure. Hence the proposed topology provides reliability.

4.3 Hop count

The proposed topologies has less number of hops count than mesh topology and fat tree. As the network size increases proposed 2D topologies uses less number of hops compared to other 2D topology like mesh.

5 Conclusion and future work

In this paper we proposed two new network on chip topology which is scalable in both X and Y direction along with suitable adaptive routing algorithms for both of the topologies. Topology A solve the scalability problem into some extent. Again topology B, which is a modified version of topology A, solve the congestion problem present in topology A. Also in the topology B diagonal connections are used, which is not even present in mesh topology. Diagonal connections help to reduce number of hop count and latency.

In future we will try to improve our proposed topology such a way that it can give better performance in latency and we will try to reduce hardware usage and static destination value.

References


