Design and Implementation of Mesh Topology for Network On Chip

A V Aswin
Department of Computer Science and Engineering,
Muthoot Institute of Technology and Science,
Kochi, India
hi@aswinavofficial.com

Anu S Alunkal
Department of Computer Science and Engineering,
Muthoot Institute of Technology and Science,
Kochi, India
anualunkal10@gmail.com

Merin Joseph
Department of Computer Science and Engineering,
Muthoot Institute of Technology and Science,
Kochi, India
merinj785@gmail.com

Dr. Sanju V
Department of Computer Science and Engineering,
Muthoot Institute of Technology and Science,
Kochi, India
sanjuv21@gmail.com

***Abstract*—Network on Chip (NoC) is another paradigm to detail the interconnections in a System on Chip (SoC) system. With the fast utilization of NoC technology the bus structure which regularly brings about activity clog, is supplanted with an integrated network.Today a NoC comprises of gadgets that use the network routers controlling traffic among gadgets and wires closely resembling the Internet. Thus efficient network topology and routing algorithm are crucial for the NoC design. There are different topologies available today; this paper presents design and implementation of Mesh topology with XY routing algorithm.The proposed strategy is a quick approach for transferring data via a specific path between two nodes in the network. As Mesh size increases latency and number of hops increases accordingly**

Keywords—-Network on Chip; NoC Simulator; Design and Implementation; Mesh Topology

I. INTRODUCTION (*NETWORK ON CHIP*)

The network topology is an arrangement of nodes in a network. Network on chip is an embedded switching network. Introduction of the network on chip could possibly solve the various uses like reliability and efficient communication between the ever-increasing numbers of modules on chips that are faced by the chip designers. There are many methods of arrangement of different nodes inside a chip. One such topology that is under consideration is this paper which is Mesh Topology.

II. MESH TOPOLOGY

A. Definition

In mesh topology, the nodes connect directly, to as many other nodes as possible. It resembles a matrix with n rows and m columns. Thus the total number of nodes in a n×m mesh network is nm. Here the connection between the nodes are made and then the data propagate along a predefined path and reaches its destination. Mesh topology is one of the most commonly used topologies for an NoC design due to its easy mapping onto an SoC.

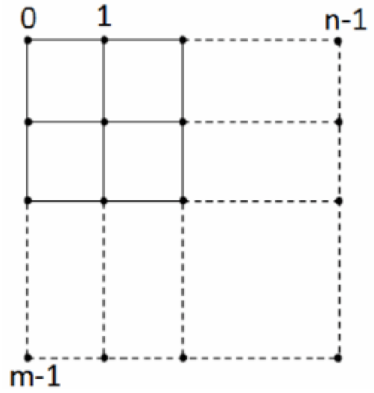


Fig 1: Mesh Topology

B. Node Structure

Node is the basic building block of the topology. Class definition of mesh topology is as follows:

```
Class MeshNode
{
public:
Interface interfaces[4];
int x,y;
int Routing_Algorithm(struct packet);
void Control_Logic();
void Acquire_Data_Packet(int,int,char[]);
}
```

Each node contains a packet generator, routing algorithm, and a control logic. Packet generator accepts addresses and data, and stores it in the form of packets. The Routing algorithm is responsible for transferring data from source to destination through a predefined path (i.e. connected nodes). Control logic generates control signals which determine the working of a node. In addition to these, nodes have interferences to communicate with adjacent nodes. The send and receive register present in the node transmits and stores the data respectively.

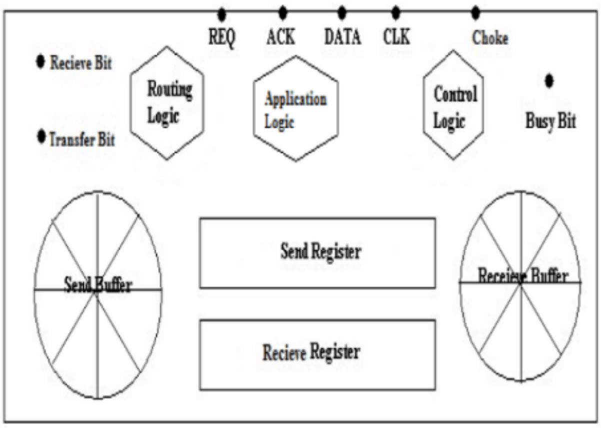


Fig 2 : Interface Structure

C. *Node with Interface*

Communication between nodes is made possible with the help of an interface. In the mesh topology, a node has 4 interfaces to communicate with the neighboring nodes.

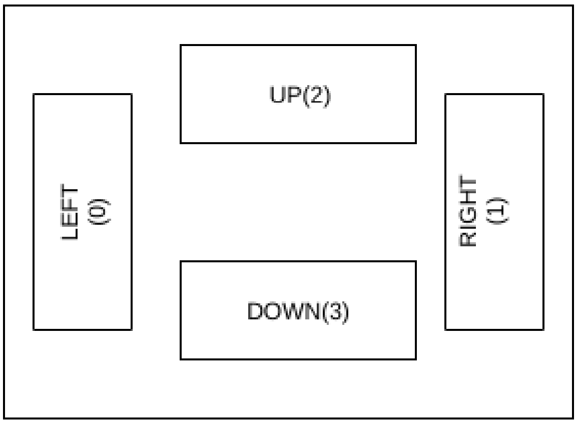


Fig 3 : Node Structure

III. Interfaces

The connection of nodes is as follows:

1. For each row, created each row pointer.
2. For each row, that much column nodes are created.
3. For each node create four interfaces.
4. Creating link objects

Number of objects = row*column

5. Creating connection

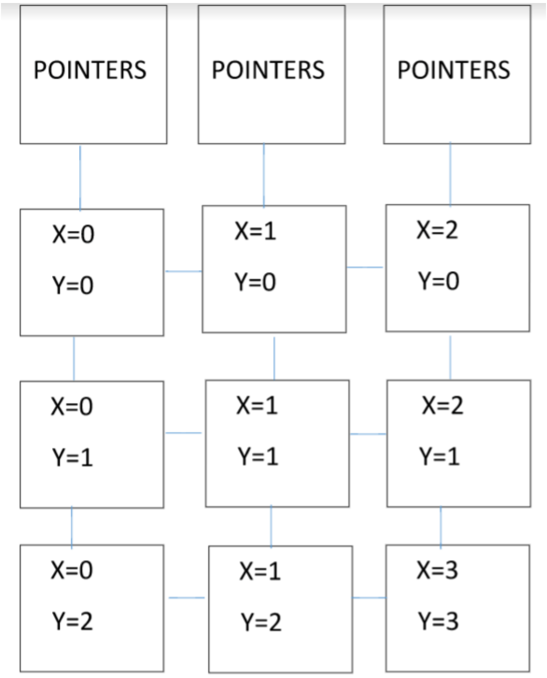


Fig 4: Connection

IV. IMPLEMENTATION

The simulator was implemented in C++ programming language. The simulator has to be compiled and the binary must be executed to start the simulation.

1. *Packet Format*

Trade of packets that stores information is the fundamental wellspring of correspondence between nodes. Each packet has three fields. The first field stores the source address. It has the addressing nodes that generated the packet. The Second field has the destination address, i.e. the address of the node to which data to be sent. The third field is the data field that stores the data to be transmitted.

SOURCE ADDRESS		DESTINATION ADDRESS		DATA
ROW	COL	ROW	COL	

Fig 5 : Packet Format

The correspondence procedure begins with the information in the send register. when a request signal is generated the busy bit is set.The following stage is to check whether the receive register is fit for putting away the whole data. Assuming this is the case, an acknowledgment flag is sent back to signal that it is prepared by setting the comparing bit high. On getting this signal, a clock is created by the sending interface. The information is transmitted from send register to the receive register of the proper receive register in a serial way.

We have implemented packet using the structure in C++ which stores source address, destination address as an integer type, data as a string of size 30 characters and start time, end time, number of hops as integer.

Structure definition of packet is as follows :

```
struct packet {
    int source_address_row, source_address_col;
    int dest_address_row, dest_address_col;
    char data[30];
    int start_timer, end_timer;
    int nhops;
};
```

2. Routing Algorithm

The routing algorithm is in charge of deciding the shortest route from the source address to the destination address.The shortest route will guarantee a minimum number of hops. The logic checks the address of the current node and also that of the destination node in order to select the best possible path.

In this paper, we use XY Routing Algorithm. In Mesh Topology, each node can be identified by its coordinate (x, y) The XY routing algorithm compares the current node address (CuX, CuY) to the destination node address (DstX, DstY) of the packet.

XY Algorithm is implemented in C++ as follows:

```
int MeshNode:: Routing_Algorithm(struct packet p)
{
        int dy=p.dest_address_col;
        int dx=p.dest_address_row;
        if(dy<y)
        {
            return LEFT;
        }
        else if(dy>y)
        {
            return RIGHT;
        }
        else if(dx<x)
        {
            return UP;
        }
        else
        {
            return DOWN;
        }

}
```

Algorithm takes packet as input returns integer corresponding to direction.

(LEFT=0,RIGHT=1,UP=2,DOWN=2)

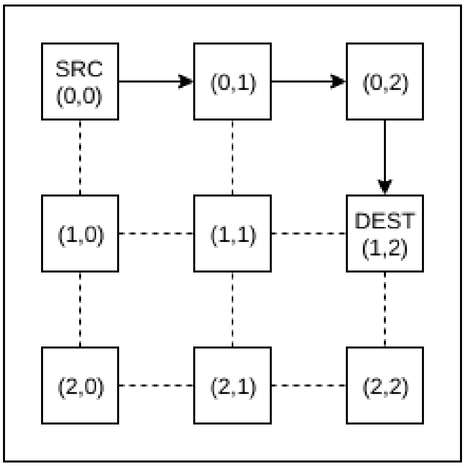


Fig 6 : Routing in 2X2 Mesh

3. Control Logic

All the internal working is controlled by it by generating appropriate control signal.On gathering of the packet in the suitable receive register, the control logic along with the routing logic checks the destination address in the packet and derives the following node to be navigated keeping in mind the destination to achieve the goal. The control logic is also responsible for generating the request signal mentioned earlier. the two main operations of the control logic are the sending and receiving of data. The control logic is called when the data is received in order to determine the next node in the path.

The control logic checks whether the choke and busy is set. If the choke is set, it resets the choke bit and the cycle continues. If both the signals are low then the sending interface raises request signal along with setting of the busy bit and transfer bit and waits for an acknowledgment. A priority resolver is used to resolve conflicts if both interfaces raise its request pin simultaneously. If the request is not acknowledged by the adjacent interface within a timeout period then the request pin and status bits are reset. The acknowledgment is returned by the adjacent interface if there is a space in its receive buffer. This is accompanied by resetting of the request line at the interface and setting of the receive bit and the busy bit.

On reception of the acknowledgment at the sending interface, its request line is reset and the packet is placed in its send register which is sent serially through the data pin along with generation of internal clock. The packet is received serially in the receive register through the data pin.Choke bit is used to integrate fairness in transmission of packets alternately between interfaces. During reception of data, if the send buffer contains packets then the choke bit is set. The interface checks the choke bit and if it is set it allows the adjacent interface to set the request providing fairness. After the packet is fully received, the status bits are reset at both the interfaces. The packet is given to the application logic if the packet reaches destination else the routing algorithm is performed on the packet to determine the next interface along with its destination. Now the control logic checks whether there is a space available in the send buffer along its destination. If there is a space it is transferred to send buffer else it is placed in the receive buffer at the receiving interface. Now the cycle continues till the packet reaches the destination where the packet is taken out of the system.

V PERFORMANCE PARAMETERS

Latency is an important factor that contributes to network speed.Latency refers to the delays incurred during the processing of data. It is also described as the time required to transfer the information from source to destination. Latency is calculated in units of time.It can also be expressed in terms of simulator clock cycles.

Hops required to reach the destination from source is another parameter for the comparison of the different topologies. A better topology will take less number of hopes and small amount of time to reach the destination.

VI SIMULATION RESULTS & ANALYSIS

The simulation is performed on a Pentium Dual Core 2.8 GHz System with 4 GB of RAM.

The different cases simulated were:

1. *One node sending data to all the other nodes*
2. *All Source to one Destination Node*
3. *One Source and One Destination*
4. *All source nodes to all Destination nodes*

Case 1: *One node sending data to all the other nodes*
Source : 0,0 Data:10101

ROW	COL	SrX	SrY	DstX	DstY	Time(s)	hops
3	3	0	0	0	2	0.08	1
3	3	0	0	2	2	0.22	4
4	4	0	0	1	0	0.12	1
4	4	0	0	3	3	0.76	6
5	5	0	0	1	0	0.14	1
5	5	0	0	4	4	1.49	8
8	8	0	0	1	0	0.60	1
8	8	0	0	7	7	6.10	14
10	10	0	0	1	0	1.00	1
10	10	0	0	9	9	12.5	18

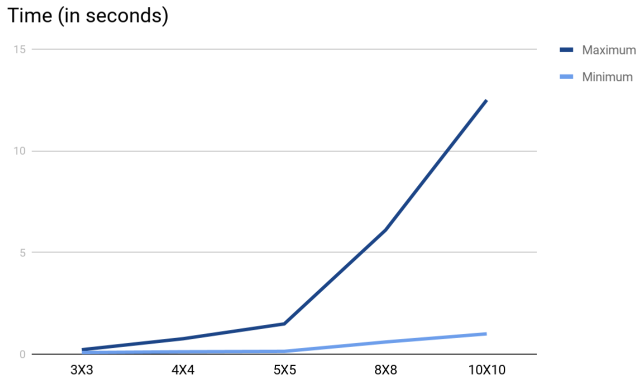
Case 2:*All Source to one Destination Node*

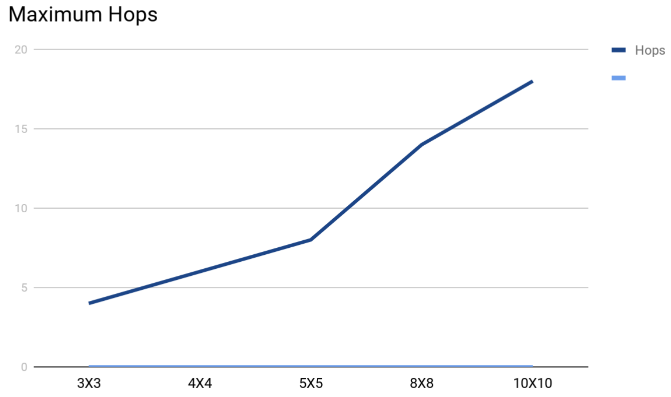
Destination:Last Node Data:10101

ROW	COL	SrX	SrY	DstX	DstY	Time(s)	hops
3	3	0	0	2	2	0.28	4
3	3	2	1	2	2	0.06	1
4	4	0	0	3	3	0.69	6
4	4	3	2	3	3	0.27	1
5	5	0	0	4	4	1.60	8
5	5	4	3	4	4	0.19	1
8	8	0	0	7	7	7.52	14
8	8	7	6	7	7	0.63	1
10	10	0	0	9	9	13.0	18
10	10	9	8	9	9	0.69	1

Case 3: One Source and One Destination
Data:10101

ROW	COL	SrX	SrY	DstX	DstY	Time(s)	hops
3	3	0	0	2	2	0.28	4
4	4	0	0	3	3	0.69	6
5	5	0	0	4	4	1.60	8
8	8	0	0	7	7	7.52	14
10	10	0	0	9	9	13.0	18





VII FUTURE WORK

Future work incorporates the augmentation of the NoC simulation system to support more network topologies and the implementation of the simulator to multi-core processors.And we like to implement the technique of multithreading which includes parallel execution of several threads.

VII CONCLUSION

The simulator has helped in studying the characteristics and internal working of Mesh Topology.It was found that time latency and the number of hops increases exponentially as the mesh size increases.

Acknowledgment

This project is the consequence of our diligent work wherein we have been helped and bolstered by a few people straightforwardly and in a roundabout way. Presently it is an ideal opportunity to recognize their commitments.. I would like to sincerely thank our guide Dr. Sanju V., for his support and valuable guidance. His convenient guidance, fastidious examination, the academic and logical approach has helped us finish this undertaking on time.

We would like to express our sincere gratitude to our principal, Dr. Ram Kumar, Head of the Department Dr. Tripti S Warrier and the management of Muthoot Institute Of Technology and Science for their support and to all those people who helped us complete throughout the entire project. Last however not the slightest we are appreciative to every one of our companions and guardians for their valuable inspiration and support.

References

[1] William J. Dally , Brian Towles, Route Packets Not Wires : on chip interconnection network, DAC 2001 June 2001.

[2] Ville Rantala , Teijo Lehtonen, Juha Plosila, Network On Chip Routing Algorithms, TUCS Technical Report No 779, August 2006

[3] Shashi Kumar, Axel Jantsch, Juha Pekka Soininen, Martti Forssell, Mikael Mellberg, Johnny berg, Kari Tiensyrja And Ahmed Hemani, Deadlock Free Routing Algorithms for Mesh Topology NoC Systems with Regions.

[4] Nurmi Network on chip: a new paradigm for system on chip design, Proceedings 2005 International Symposium on System on Chip, 1517 November.

[5] L. Benini and G. De Micheli, Networks on chips: A new SoC paradigm, Computer, vol. 35, no. 1, 2002, pp. 7078.

[6] Lalit Kishore Arora, et.al, Performance Evaluation of mesh with source routing for Packet Loss. IJ ARCSSET Volume-1, Issue-5, August 2012 .

[7] R. Mullins, A. West, and S. Moore, Low-latency virtual channel routers for on-chip networks, in Proc. Int. Symp. Comput. Architecture, Jun. 2004, pp. 188197.

[8] L. Peh and W. J. Dally, Flit-reservation flow control, in Proc. Int. Symp. High-Performance Comput. Architecture, Jan. 2000, pp. 7384.