Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 330 – 337

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Comparative Design and Analysis of Mesh, Torus and Ring NoC

Arpit Jain[a]*, Adesh Kumar[b] , Sanjeev Sharma[c]

[a.] *Research Scholar, Department of Computer Science, Teerthanker Mahaveer University, Moradabad, India*
[b] *Department of Electronics, Instrumentation and Control Engineering, University of Petroleum & Energy Studies, Dehradun India*
[c]*Department of Computer Science & Engineering, JP Institute of Engineering & Technology, Meerut, India*

Abstract

The research article presents the simulation and FPGA synthesis of mesh, torus and ring Network on Chip (NoC). The network is based on the Multiprocessor System on Chip (MPSoC) structure for a network cluster of 256 nodes. The paper focuses on the comparative analysis based on hardware design parameters, memory utilization and timing parameters such as minimum and maximum period, frequency support. The interprocess communication among nodes in verified using Virtex-5 FPGA with an arbitration logic. The designs are developed in Xilinx ISE 14.2 and simulated in Modelsim 10.1b with the help of VHDL programming language. Network topological structures help for on chip intercommunication, routing, switching, flow control, queuing, scheduling and to communicate among different networks.

Keywords: Network on Chip (NoC), Multiprocessor System on Chip (MPSoC), Field Programmable Gate Array (FPGA)

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

1. Introduction

Today, the designers are facing the problem of on chip interconnects apart from increasing the no. of nodes. The systems using traditional bus system are facing the problem of scalability are not capable to fulfill the requirement

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: arpit.record@gmail.com

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.190

for future SoC in terms of power, timing parameters, hardware utilization, performance and predictability. To overcome the design productivity gap, cost, and signal integrity for future SoC, a scalable NoC structure is helpful to realize the on chip communication problems. Chip processors (CMP)[2] and MPSoC[1,5] uses bus structure for on chip communication and integrates nodes on a single die to meet the requirement of transistor density, more throughput, less delay, less time to market, and operating frequency.

The internode communication in multiprocessor system is based on the concept of memory sharing or message passing. The message passing among nodes is dependent on APIs such as *transmit ( ), receive ( ).* The APIs need some protocol to connect each other. In many multiprocessor NoC system shared memory architecture concept is used and data transfer is possible trough memory access points. MPSoC architecture is based on processor memory hierarchy and topological structure helpful for interprocess communication in network. The shared memory based architecture provide high throughput because of shared or cache memory between processors and pipelined processing for data transactions. Lucent developed a single chip multiprocessor called Daytona[1].It was having 64 bits processing elements targeted for DSP applications with scalable structures and performs transactions with different sizes. On chip was there to perform the on chip communication that split transactions and different targets. MIPS based processor was developed by Stanford Hydra[5] chip. It used shared level-2 cache memory to perform interprocess communication. DEC developed the project Piranha[6] to perform on chip communication based on packet routing. In the work eight alpha processors were integrated on a single chip multiprocessor.

A shared memory multiprocessor[7] consists of several nodes/processors or processing elements form an on chip interconnected network. All PEs[4] have their own CPU or hierarchy of their memory, may be one or two level of cache memory. The multiprocessor system [3,4] has a big memory unit physically but it has shared memory accessed by different processors globally. The data packet arrives at a particular node is based on the request by the node. The memory will return a reply packet to requested node containing the data of the requested node. Read the data of the requested node and write data to destination node is accessed through cache reference. In MPSoC system, the major problem is cache coherence[7,8] because the data is saved by the different caches should be updated otherwise one data can have multiple copies. The problem of cache can be resolved with the help of cache updating that updates all node memories whenever there is new data in memory.

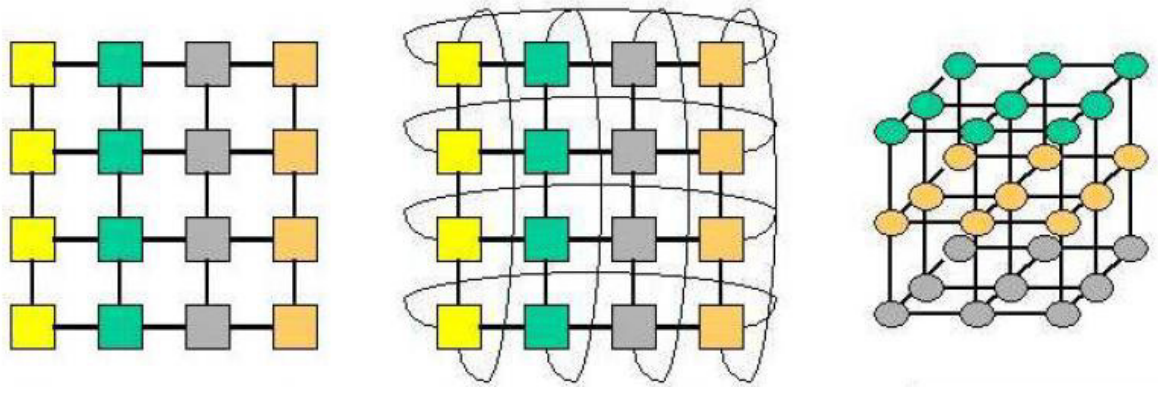

Fig. 1(a) 4 ary 2 dim mesh [7] (b) 4 ary 2 dim torus (c) 4 ary 3 dim mesh

Multiprocessor system consist of different network topologies may be targeted to specific application to enhance the NoC performance and throughput. The MPSoC network structure can form direct network and indirect network structure. In direct form all the nodes are connected directly with each other with the help of network only. The arbitration and data flow is possible with the help of each node. In indirect network structure data flow is possible by an intermediate switch. The switching and routing is performed with the help of switch between the processors. Multistage network configurations are formed using indirect networks. Orthogonal topological [7,10] structures are the examples of direct topologies. The nodes in the orthogonal topologies can form mesh structure (with k ary and n dimensional or k ary n cube) or torus (with k ary and n dimensional). The pipelined operations and

parallel processing can be performed with the help of mesh or torus [9,10] structures because the structures provide easy connection and simple routing and interconnection length between nodes can be same.

2. NoC Design Consideration

The design considerations for the mesh and torus structure for (256 x 256) is shown in fig. 2(a) and (b) in which 256 nodes can process intercommunication. Each node is identified with its address assigned N_0 (00000000), N_1(00000001), N_2(00000010), N_3(00000011), N_4(00000100),N_5(00000101).....N_{255}(11111111). There is also row and column address assigned for node identification based on row and column processing having 8 bits addresses because (2^8= 256). The functionality of mesh and torus NoC structure is understood with the help of table. For an example node, the identification of node 18 is based on row address (00000001) and column address (00000010) but it has the probability to communicate with any node in NoC.

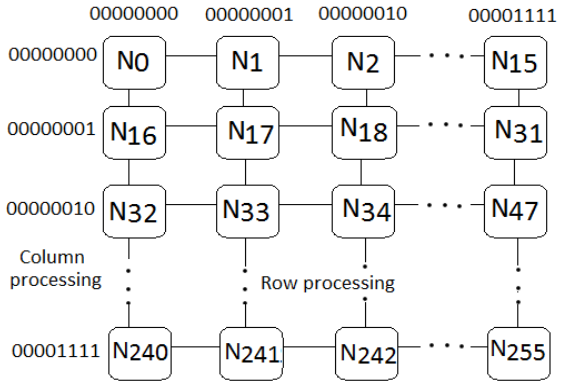

Fig. 2(a) Mesh NoC (256 x 256)

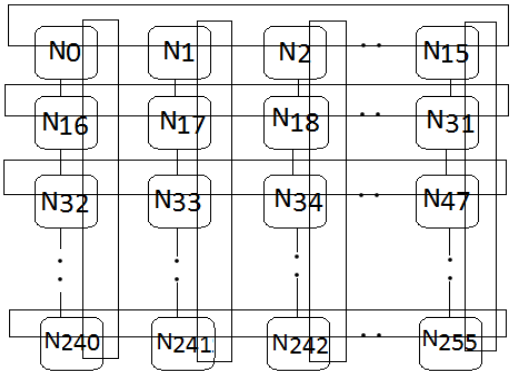

Fig.2 (b) Torus NoC (256 x 256)

Table 1 Node selection in mesh and torus ring topological NoC

Row_address(8 bit)	Column_address (8bit)	Node Selection
00000000	00000000 : 00001111	Node 0 : Node 15
00000001	00000000 : 00001111	Node 16 : Node 31
00000010	00000000 : 00001111	Node 32 : Node 47
00000011	0000000 : 00001111	Node 48 : Node 63
: :	: :	: :

00001111	00000000 : 11111111	Node 240 : Node 255

The topological structure of ring NoC for 256 nodes is shown in fig.2(c). The structure has 256 nodes, arranged in a ring configuration. The functionality of the ring NoC can be understood with the help of table 2. All 256 nodes are counted from N_0 to N_{255} sequentially counted with their node address of 8 bits starting from "00000000" to "11111111" Let node N_0 is assigned a source_address "00000000", Node N_1 has address "00000001". In the same way, all the nodes can be assigned their 8 bits of address and node N_{256} is assigned source_address "11111111". Moreover, nodes have the priority mechanism to communicate in multiprocessor system. The data packet arrival to source and delivery to destination node is considered with the help of arbiter which assigns the priority for interconnection of destination node in mesh, torus and ring NoC.

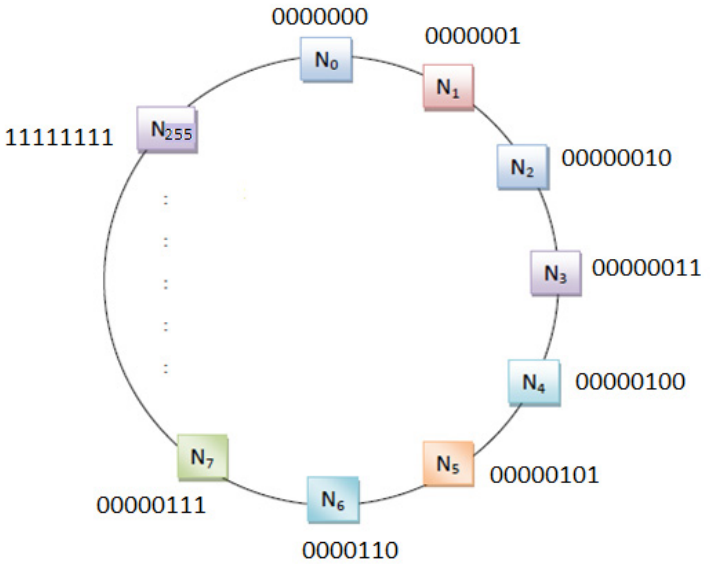

Fig. 3 Ring NoC (256)

Table 2 Node selection in ring topological NoC

Source_node_address (8 bit)	Destination_node_address (08bit)	Node Selection
00000000	00000000 : 11111111	Node 0 : Node 255
00000001	00000000 : 11111111	Node 0 : Node 255
00000010	00000000 : 11111111	Node 0 : Node 255
00000011	0000000 : 11111111	Node 0 : Node 255
:	:	:

:	:	:
11111111	00000000	Node 0
	:	:
	11111111	Node 255

3. Results & Discussions

The RTL view is the description of input and outputs of the developed chip. The RTL view of the NoC is shown in fig. 4. The functionality of the individual pin is described in table 3. The functional modelsim simulation shown in fig. 5, shows the data transfer scheme from node N_3 to node N_4. The functional simulation depends on the following steps input.

Step input 1: Reset = '1' and run, all node data will contains zero output.

Step input 2: Reset = '0', Apply rising edge clock pulse, source_address and destination_address value and data of destination node with input_data_packet, then run.

Step input 3: Apply the source address and destination address of another nodes and data on input_source.and run

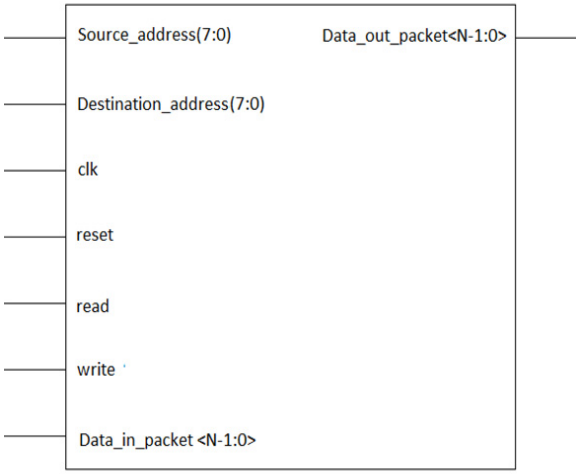

Fig. 4 RTL view of NoC (Common to mesh, torus and ring NoC)

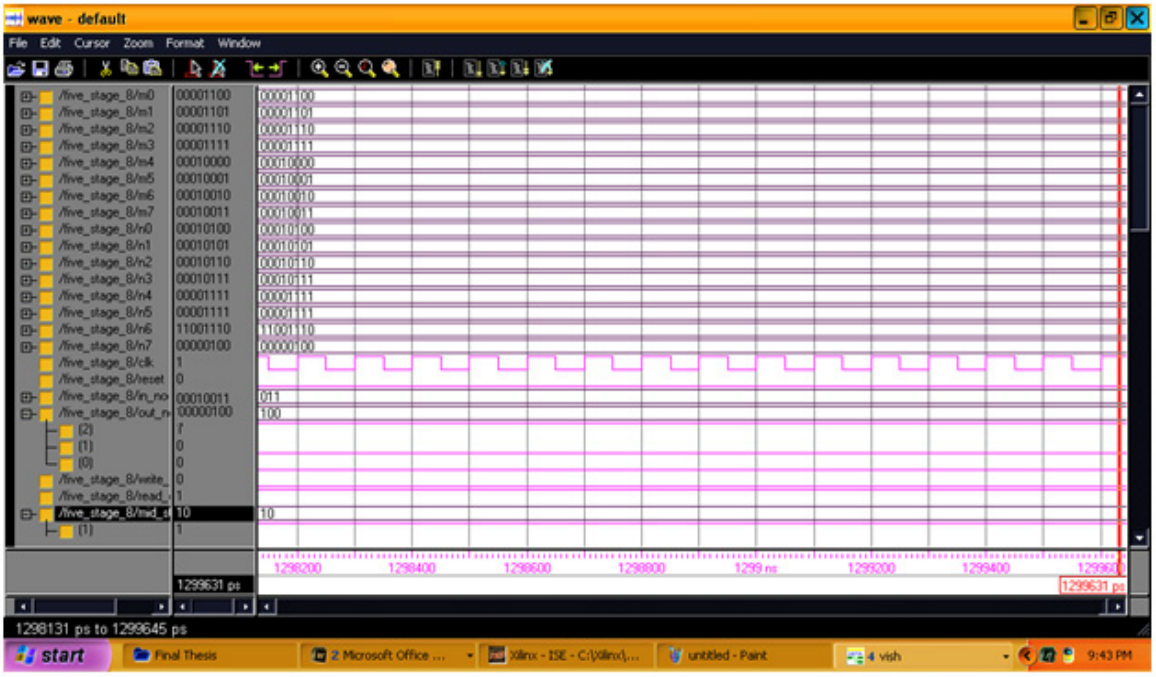

Fig.5 Modelsim simulation for NoC from N_3 to N_4 (Common to mesh, torus and ring)

Table 3 Pin details of NoC

Pins	Description
reset	Default input signal used to reset the memory contents zero for synchronization of the components by using clk of std_logic (1 bit)
clk	Synchronized input for sequential logic to work on rising edge of clock pulse of std_logic.(1 bit)
source_address [7:0]	Address of input source nodes of std_logic_vector (8 bit)
destination_address [7:0]	Address of output destination nodes of std_logic_vector (8 bit)
input_data_packet [N-1:0]	Input data of 'N' bits of the source node of std_logic_vector (N-1:0)
output_data_packet [N-1:0]	Output data of 'N' bits of the source node of std_logic_vector (N-1:0)
write	Memory control signal to perform write operation with respect to individual node of std_logic(1 bit)
read	Memory control signal to perform read operation with respect to individual node of std_logic(1 bit)

4. FPGA Synthesis Results

Device utilization report gives the percentage utilization of device hardware for the chip implementation. Device hardware includes no. of slices, no. of flip flops, no. of input LUTs, no. of bounded IoBs, and no of gated clocks (GCLKs) used in the implementation of design. Timing details provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock. Table 4 and table 5 show the synthesis results as device utilization and timing parameters for mesh, torus and ring NoC. Total memory utilization required to complete the design is also listed for individual stage. The target device is: xc5vlx20t-2-ff323 synthesized with Virtex-5 FPGA.

Table 4 Device utilization in NoC structures

Device Part	Utilization		
	Mesh (256 x 256)	Torus (256 x 256)	Ring (256)

Number of Slices	344 out of 12480,	3%	324 out of 12480,	3%	131 out of 12480,	1%
Number of Slice Flip Flops	362 out of 12480,	3%	358 out of 12480,	3%	142 out of 12480,	1%
Number of 4 input LUTs	120 out of 362,	33%	115 out of 358,	32%	87 out of 362,	24%
Number of bonded IOBs	86 out of 172,	50%	81 out of 172,	47%	53 out of 172,	34%
Number of GCLKs	1 out of 32,	3%	1 out of 32,	3%	1 out of 32,	3%

Table 5 Timing parameters for NoC structures

Timing Parameter	Utilization		
	Mesh (256 x 256)	Torus (256 x 256)	Ring (256)
Minimum period	1.457ns	1.497 ns	0.987ns
Maximum frequency	600.00 MHz	589.00 MHz	780.00 MHz
Minimum input arrival time before clock	3.190 ns	3.460 ns	1.189 ns
Maximum output required time after clock	2.230 ns	2.490 ns	2.150 ns
Total memory usage	221985 kB	207895 kB	126740 kB

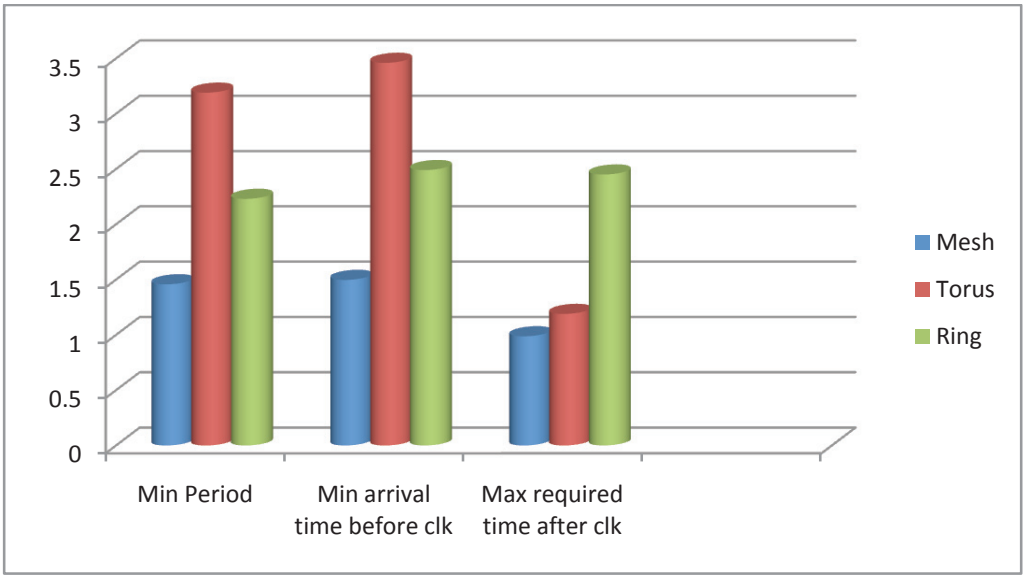

Fig. 6 Comparative graph for mesh, torus and ring NoC structure

Fig. 6 describes the timing variations in the design of mesh, torus and ring NoC. From the device utilization and timing parameters, it clarified that ring NoC has optimized parameters. In torus structure min period 2.74 %, minimum input arrival time before clock 8.45 % and maximum output required time after clock 11.65 %, is greater than in comparison to mesh structure. In ring NoC structure min period 32.25 %, minimum input arrival time before clock 62.72 % and maximum output required time after clock 3.58 %, is less than in comparison to mesh structure. The hardware and memory utilization in torus and ring NoC is less than mesh NoC. The frequency support for the same targeted device is 600.00 MHz, 589.00 MHz and 780 MHz dor mesh, torus and ring NoC respectively, which

signifies that ring NoC is faster in comparison to mesh and torus and has significant less hardware optimization to support a particular network configuration.

5. Conclusions

The NoC design for mesh (256 x 256) torus (256 x 256) and ring (256) is implemented on Virtex 5 FPGA successfully. The architecture is based on shared memory architecture and optimal routing scheme is suggested. The design is tested for the different test cases. In each NoC configuration, the data transfer with arbitration scheme is verified on modelsim 10.1 b and FPGA successfully. The synthesis report is generated and contains the information for hardware utilization in terms of No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs and No of gated clocks (GCLKs) used in the implementation of design. Timing analysis is also carried out for the staged network which provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock. A comparative study is carried out for the mesh, torus and ring NoC structure hardware and timing parameters and estimated that ring NoC has optimized results.

References

1. A. G. Wassal, M. A. Hasan, “Low-power system-level design of VLSI packet switching fabrics”, IEEE Transactions on CAD of Integrated Circuits and Sys- tems, June 2001. pp. 723-738.
2. B. Ackland; et.al, “A single Chip, 1.6-Billion, 16-MAC/s Multiprocessor DSP”, IEEE J. Solid-State Circuits, March 2000, pp. 412-424.
3. E. Rijpkema, K. G. W. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, ”Tradeoffs in the design of a router with both guaranteed and best-effort services for networks on chip”, Proceedings of Design Automation and Test Conference in Europe, March 2003, pp. 350-355
4. Ganghee Lee, Kiyoung Choi, and Nikil D. Dutt, “Mapping Multi-Domain Applications onto Coarse-Grained Reconfigurable Architectures” *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 5, pp (637-650) , May 2011.
5. L. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, B. Verghese, “Piranha: A Scalable Architecture Based on Single- Chip Multiprocessing”, Proceedings of 27th Annual International Symposium on Computer Architecture, 2000, pp. 282-293.
6. L. Hammond, B Hubbert, M. Siu, M. Prabhu, M. Chen, K. Olukotun, “The Stanford Hydra CMP”, IEEE MICRO Magazine, March-April 2000, pp.71-84
7. Terry Tao Ye Ph.D thesis On-Chip Multiprocessor Communication Network Design AND Analysis, ch-1,ch-2, December 2003 pp (1-35)
8. Teijo Lehtonen, Pasi Liljeberg, and Juha Plosila “Online Reconfigurable Self-Timed Links for Fault Tolerant NoC” *VLSI Design, Hindawi Publishing Corporation* (2007), pp (1-13)
9. Vasilis F. Pavlidis, Eby G. Friedman “3-D Topologies for Networks-on-Chip” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 15, No. 10, October 2007, pp (1081-1091),
10. Wen-Chung Tsai, Ying-Cherng, Lan,Yu-Hen Hu, and Sao-Jie Chen “Networks on Chips: Structure and Design Methodologies” Hindawi Publishing Corporation., Journal of Electrical and Computer Engineering pp (1-13), 2012.