J. Basic. Appl. Sci. Res., 1(10)1594-1602, 2011

ISSN 2090-424X
Journal of Basic and Applied Scientific Research
www.textroad.com

Simulation and Performance Evaluation of Network on Chip Architectures and Algorithms using CINSIM

Sheraz Anjum*, Ehsan Ullah Munir and Muhammad Wasif Nisar

COMSATS Institute of Information Technology, Quaid Avenue, The Mall, Wah Cantt, Pakistan
*drsheraz@comsats.edu.pk, ehsanmunir@comsats.edu.pk and wasifnisar@gmail.com

ABSTRACT

The continuous saturation problems of today's buses introduced due to large number of resources has lead the System on Chip (SoC) designers to research a new SoC paradigm known as Networks-on-Chip (NoC). The research related to NoC is still in its infancy and researchers are trying to propose, design and explore different architectures, algorithms and simulation tools related to NoC. In this paper we have explored a tool named as CINSIM (Component Based Interconnection Network Simulator) that is designed for linux systems by a research team at Real-Time Systems and Robotics of Technische Universität Berlin. First we introduced the simulation components and environment related to the tool and then applied the tool on a heterogeneous 2D-Mesh Network on Chip architecture. We showed how the different combinations of routing algorithms, switching techniques, simulation types, traffic sources and measurement types can be used to check the performance evaluation of the underline heterogeneous NoC which in turn shows the general methodology of simulation and performance evaluation of any NoC platform under consideration. We hope this effort will help many NoC researchers for the quick design, analysis and selection of particular NoC platforms.

KEY WORDS: Component Based Interconnection Network Simulator (CINSIM), Switching Techniques, Routing Strategies, Traffic Sources, Flow Control Unit (Flit).

1 INTRODUCTION

According to ITRS [1], we are now able to fabricate billions of transistors on a single chip using 45nm or lower process technologies. Current SoC design methodologies are not scaling well with the advancement of process technologies. The use of buses in today's SoCs for interconnecting heterogeneous resources is becoming a bottleneck due to contention and congestion problems. More over global wire delays, reusability, less modularity and scalability issues have added to the problems of current bus based SoCs. Consequently some research groups [2], [3] and [4] have proposed to adopt the more modular and scalable design methodologies known as Networks on chip, a new SoC paradigm. The use of Globally Asynchronous and Locally Synchronous concept in NoCs has disintegrated the design of resources from the rest of the network. Its use could enhance the scalability, modularity and reusability of IP.

Design and selection of appropriate architecture, routing algorithm and switching technique for on chip communication has a key role in the design and implementation of the complete platform for NoC. An appropriate NoC tool can accelerate the process of design and selection of suitable NoC architectures and algorithms. Due to lack of tools available for NoC Simulations, many researchers as in [5] [6], [11] and [12]have used publicly available Network Simulator 2 (NS-2) [7], [8], [9] for the simulation and evaluation of their NoC architectures and algorithms, but NS-2 is less suitable for the simulation and performance evaluation of networks on chip. Therefore we decided to explore, introduce and apply CINSIM [10] tool that is more suitable for the simulation and performance evaluation of Networks-on-Chip architectures and algorithms. In the following section we will briefly discuss the specification and use of CINSIM tool.

***Corresponding Author:** Sheraz Anjum, COMSATS Institute of Information Technology, Quaid Avenue, The Mall, Wah Cantt, Pakistan. Email: drsheraz@comsats.edu.pk

2 CINSIM

CINSIM is a Component-based Interconnection Network Simulator and is designed for Linux environments. The simulation is performed by the simulation core (cinsim) to be controlled at the command-line. The core is widely written in C++ and capable of executing performance analysis of regular and irregular interconnection networks with some boundary conditions to be satisfied. The simulation setup, including the network description, must be specified by an XML file based on an XML schema. For this purpose, the CINSIM provides a fully schema driven editor (cinsim-gui) that can visualize and edit the simulation parameters of the XML document shown in Figure 1 to easily describe interconnection networks. The editor is written in JAVA that makes it platform independent.

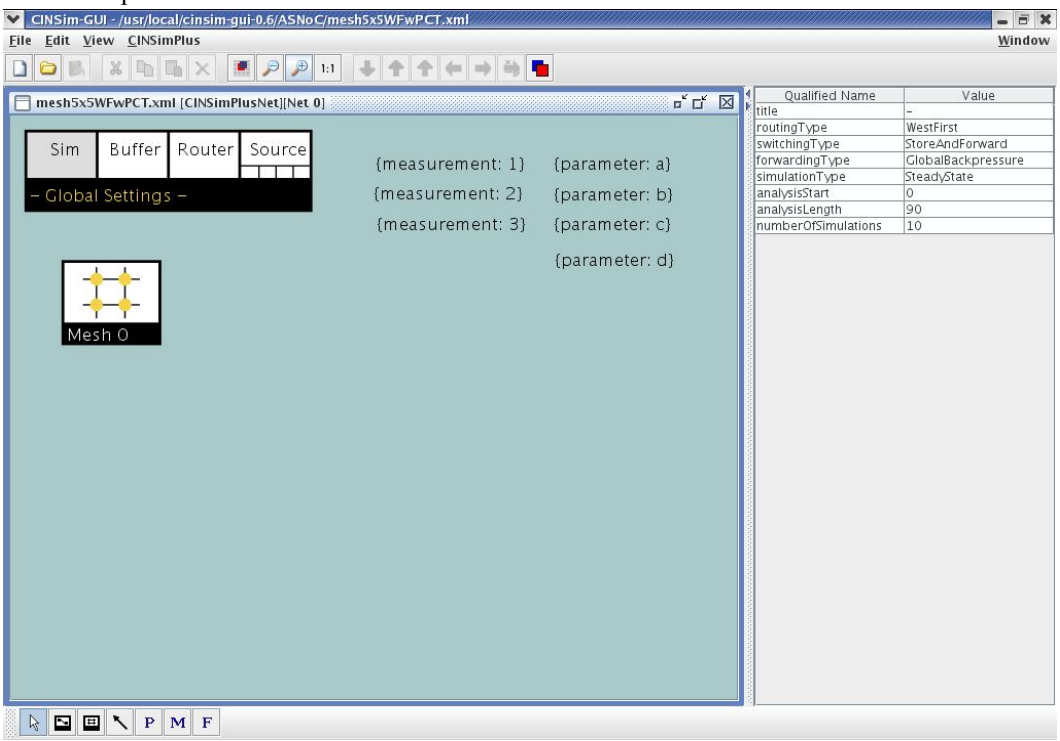

Figure 1: The Global Setting Menu of 5x5 Heterogeneous 2D-Mesh XML Document

The performance analysis of interconnection networks that can be executed using the simulator CINSIM includes mean packet delay, mean queue length, mean flit delay, target and source throughputs. These properties can be investigated using steady-state or terminating simulations. Confidence levels and estimated precisions are observed for each measure. If the desired termination criteria are met, the simulation stops. The component-based approach of the simulator CINSIM leads to a distinction of several network and simulation components that can be put together in many ways. The following Network components can be used to describe regular or irregular interconnection networks:

- Source Buffers – Used to specify traffic sources
- Non-Shared Buffers – Used as intermediate data (Flit) holders
- Routers – Used to route packets
- Target Buffers – Used to analyze received packets

The simulation of an interconnection network invokes several independent simulation components that can also be set up using the provided XML editor. The simulation components are listed below:

- Routing Strategies – Includes Bitmask, Shortest Path, XY and West First routing
- Switching Techniques – Packet switching including Store-and-Forward, Virtual/Partial Cut Through and Wormhole switching
- Terminating and Steady State Simulations
- Scheduling Algorithms - To resolve routing conflicts
- Measure Routines - Flit Delay, Target Throughput, Source Throughput, Mean Queue Length

- Backpressure Mechanisms – Local or Global

In the following sections we briefly describe the important network and simulation components that can be setup by the provided schema driven editor for the XML document under consideration.

2.1 Source Buffer Component

A source buffer does not receive packets but creates them on demand invoking various distribution functions such as geometric distribution, periodic distribution, pareto distribution and random-burst distributions. In some aspects a source buffer can be considered as the counterpart of a client connected to a network, like a single processor. A client usually transfers a message to a network addressed to a specific client or to a set of clients also connected to the network. However, a source buffer component in CINSIM can not receive messages, defined by constant-size packets; therefore an additional target buffer component is needed to simulate a client sending and receiving messages.

2.2 Buffer Component

Buffer components are used to store packets along their way through an interconnection network. Within every single clock cycle only one packet fragment (Flit) can be received and stored while on the other hand one fragment can be sent and removed.

2.3 Target Buffer Component

The target buffers are special buffers that are used as destinations for packets passed through an interconnection network simulated with CINSIM. Each target buffer is represented by a single bit within the target bitmasks of packets and network components. Targets can receive one packet fragment (Flit) per clock cycle and their queue will be cleared at the end of each clock cycle. Analyzer components connected to target buffers can analyze the received Flits.

2.4 Router Component

The router component is an abstract switch using an I/O matrix to route packets. Routers are responsible for routing, scheduling and switching and contain input and output ports. Input ports serve incoming packets while an output port is the contentious resource. The operation of a router can be divided up into the routing, the scheduling and the switching. The contentious resource can be solved by selecting one of many scheduling algorithms such as random, global round robin, local round robin, fixed order, most recently used, least recently used, most frequently used, least frequently used, oldest packet first, longest waiting packet first, priority and deadline scheduling. All arbiters call the random number generator, if the scheduling algorithm cannot solve the conflict.

2.5 Routing Strategies

The layout of an interconnection network sets up one or, in case of redundancy, more paths between a given source/destination pair. A routing strategy determines the paths to the destination for a given message. CINSIM supports four different routing strategies that includes Bitmask, Shortest Path, XY and West First routing out of which one can be set up for each network configuration in the first section of the local/global settings menu by choosing the related value for the attribute routingType.

2.6 Switching Techniques

Apart from determining valid paths between sources and destinations within an interconnection network, a switching technique is needed that specifies how messages are to be fragmented before passing them to the network and how the resources along the path are to be allocated. Furthermore, a switching technique gives preconditions to be fulfilled before a fragment can be moved on to the next network component. CINSIM features several packet switching techniques such as Store and Forward, Virtual Cut Through, Partial Cut Through and Wormhole switching technique.

2.7 Simulation Type

CINSIM provides two different types of simulation runs. A steady-state simulation can be used to determine the performance characteristics for the steady state of an interconnection network. The simulation invokes consecutive clock cycles until the results of all measurements reach steady states. In contrast to this approach, the transient behavior of an interconnection network within a given range of clock cycles can be investigated using a terminating simulation. A terminating simulation run stops and restarts after the last clock cycle of the desired range has been simulated. For every single clock cycle within the specified range, a result for all set up measurements will be determined. Both, steady-state and terminating simulation, are observed by CINSIM and stops, if all performance characteristics to be investigated reach steady states according to a given confidence level and a desired precision.

2.8 Observation Type

This analyzer component has been introduced to provide some kind of measurement device that can be connected to other

components of a network but it is actually not a network component. An analyzer refers to one or more measurement variables defining measurements. A measurement variable belongs to one parameter observed by CINSIM or, in case of terminating simulations, to a sequence of parameters. Multiple analyzers may share the same measurement variables and therefore the same parameters. The observation types supported by CINSIM include delay and latency at targets, delay and latency at buffers, source throughput, target throughput and mean queue length.

3 Implementation of a Heterogeneous 2D-Mesh NoC using CINSIM

To apply the CINSIM tool and check the performance evaluation of different parameters, we decided to choose 5x5 heterogeneous 2D-Mesh NoC architecture as shown in the Figure 2. Where 'r' represents a resource and 'Rt' represents a router. The resource 'r' can be a DSP, a processor, RAM, ROM, an ASIC or an FPGA etc. The link between any two routers is a duplex link.

By selecting 'Sim' in the global setting menu of Figure 1 and choosing different combinations of routing type, switching type and simulation type (refer to top-right corner of Figure 1), we created different XML documents related to heterogeneous 2D-Mesh to be simulated and analyzed later by the cinsim core. We defined three measurement variables 1, 2 and 3 to measure the delay at targets, source throughput and destination throughput respectively and four parameters 1, 2, 3 and 4 to be used with different simulation components detailed in Table 1. Where the start value is used by first simulation run and the add values are used for the next simulation runs in sequence.

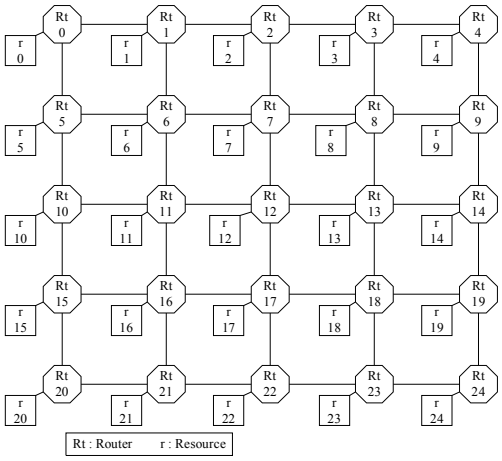

Figure 2: Architecture of 5x5 2D-Mesh NoC

Table 1: Parameter Values and their Relationship

Parameter	Start Value	Add Value	Related To
a	1	1	**Buffer Size**
b	1	1	**Packet Size**
c	0.1	0.1	**Geometric Distribution**
d	**1**	**1**	**Periodic Distribution**

By choosing the 'Mesh 0' component in the cinsim-gui editor (see Figure 1), we designed the 5x5 heterogeneous 2D-Mesh NoC architecture as shown in Figure 3. To model the NoC as containing heterogeneous resources, we applied geometric distribution on the odd numbered sources and periodic distribution on the even numbered source nodes.

The detail of components and connections comprised by a single Mesh node is shown in Figure 4. Each Mesh node contains four input and four output ports which is also clear from Figure 3. In addition it consists of a source buffer S0 to generate packets according to selected distribution function, a target buffer T0 to receive traffic in the form of packets, a router R0 to route and schedule Flits depending on selected routing technique and scheduling algorithm, five buffer components B0 to B4 to hold intermediate Flits during their flight to the destinations and two analyzer components A0 and A1 to measure the delay, target throughput and source throughput respectively.

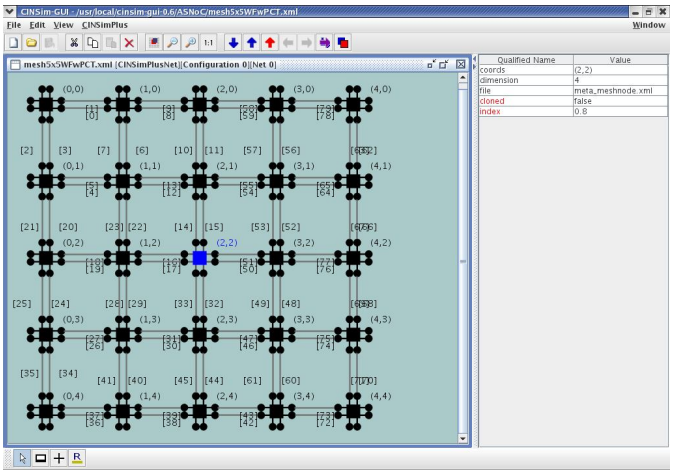

Figure 3: 5x5 Heterogeneous 2D-Mesh NoC Architecture using cinsim-gui

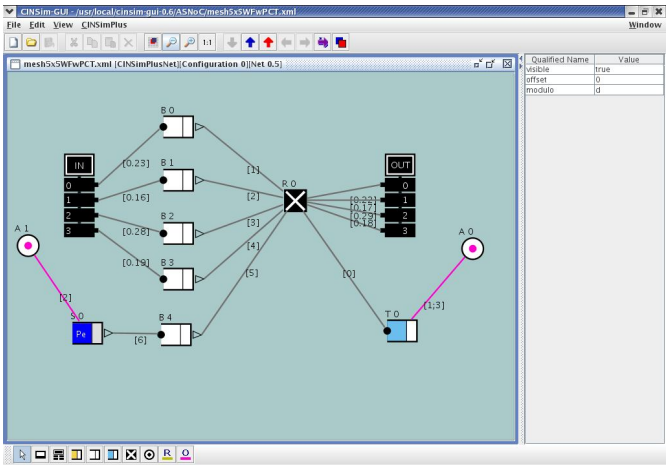

Figure 4: Components comprised by a Mesh Node

4 SIMULATION RESULTS

We analyzed the architecture of Figure 3 using different combinations of routing, switching, simulation and measurement types without disturbing the local information at each node. The detail of combinations used for the generation of different XML files is listed in Table 2. The generated XML files along with 90% confidence level and 10% precision were used one by one as an input to the 'cinsim' core to evaluate the heterogeneous 2D-Mesh architecture for parameters such as delay at targets, source throughput and destination throughputs. In the Figures from 5 to 10, we will use some abbreviations as given in Table 3.

Table 3: Abbreviations Used in Figures 5 to 10

Name	Abbreviation
West First	WF
Store and Forward	SaF
Partial Cut Through	PCT
Source Throughput	STP
Target Throughput	TTP
Steady State Simulation	SSS
Terminating Simulation	TS

Table 2: Detail of Combinations for the Generation of Different XML Files

Combination	Routing Type	Switching Type	Simulation Type	Measurement Type
1	West First	Store and Forward	Steady State	Delay at Targets
2	XY	Store and Forward	Steady State	Delay at Targets
3	West First	Partial Cut Through	Steady State	Delay at Targets
4	XY	Partial Cut Through	Steady State	Delay at Targets
5	West First	Store and Forward	Terminating	Delay at Targets
6	XY	Store and Forward	Terminating	Delay at Targets
7	West First	Partial Cut Through	Terminating	Delay at Targets
8	XY	Partial Cut Through	Terminating	Delay at Targets
9	West First	Store and Forward	Steady State	Source Throughput
10	XY	Store and Forward	Steady State	Source Throughput
11	West First	Partial Cut Through	Steady State	Source Throughput
12	XY	Partial Cut Through	Steady State	Source Throughput
13	West First	Store and Forward	Terminating	Source Throughput
14	XY	Store and Forward	Terminating	Source Throughput
15	West First	Partial Cut Through	Terminating	Source Throughput
16	XY	Partial Cut Through	Terminating	Source Throughput
17	West First	Store and Forward	Steady State	Target Throughput
18	XY	Store and Forward	Steady State	Target Throughput
19	West First	Partial Cut Through	Steady State	Target Throughput
20	XY	Partial Cut Through	Steady State	Target Throughput
21	West First	Store and Forward	Terminating	Target Throughput
22	XY	Store and Forward	Terminating	Target Throughput
23	West First	Partial Cut Through	Terminating	Target Throughput
24	XY	Partial Cut Through	Terminating	Target Throughput

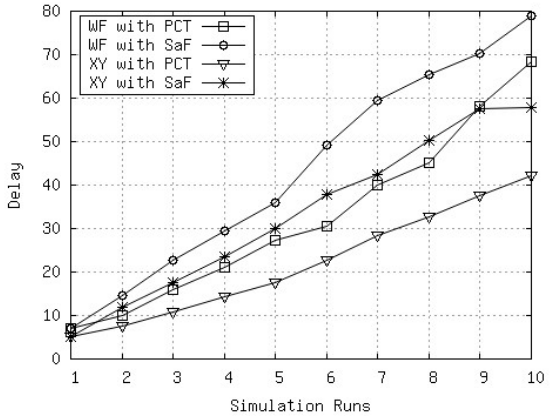

Figure 5: Delay Vs Simulation Runs Using SSS

Figures 5 and 6 with combinations from 1 to 4 and 5 to 8 (see Table 2) shows the delay at targets Vs simulation runs using steady state simulation and clock cycle using terminating simulation respectively. In both Figures XY with PCT has the lowest delay and proves to be the best combination for the heterogeneous 2D-Mesh NoC.

Figure 7 and 8 with combinations from 9 to 12 and 13 to 16 shows the source throughput Vs simulation runs using steady state simulation and clock cycle using terminating simulation respectively. On the average the source throughput for the steady

state simulation increases for the next simulation runs on the other hand it decreases for the terminating simulation up to cycle 15 and then randomly varies between 0.3 and 0.45.

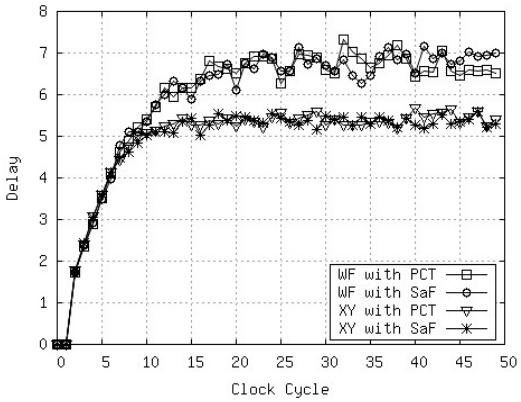

Figure 6: Delay Vs Clock Cycle Using TS

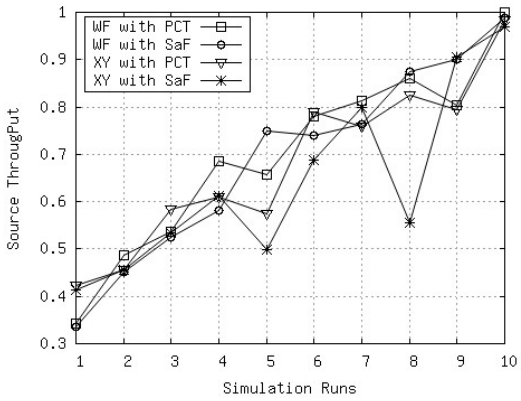

Figure 7: STP Vs Simulation Runs Using SSS

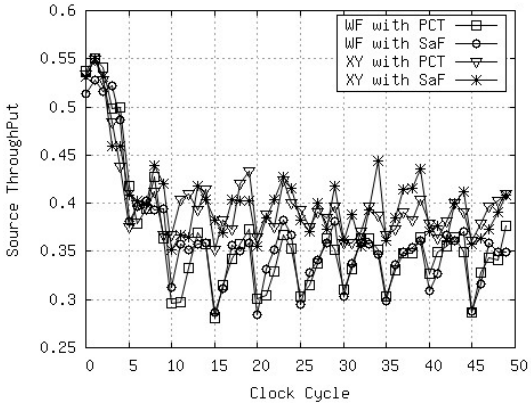

Figure 8: STP Vs Clock Cycle Using TS

Similarly Figure 9 and 10 with combinations from 17 to 20 and 21 to 24 shows the target throughput Vs simulation runs using steady state simulation and clock cycle using terminating simulation respectively. Again in this case XY with PCT has the largest target throughput. Keeping in view of these results we can conclude that XY routing with PCT switching technique can be selected as a best choice for 5x5 heterogeneous 2D-Mesh NoC. Similarly any other NoC architecture can be analyzed for different combinations and the best one can be selected for the implementations.

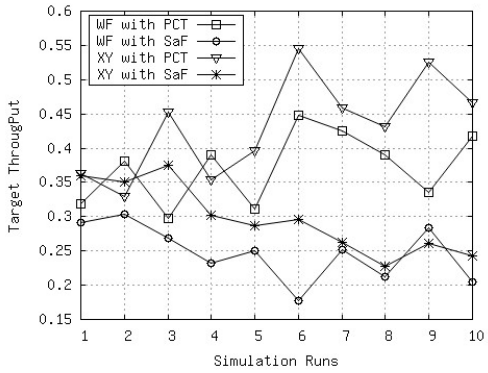

Figure 9: TTP Vs Simulation Runs Using SSS

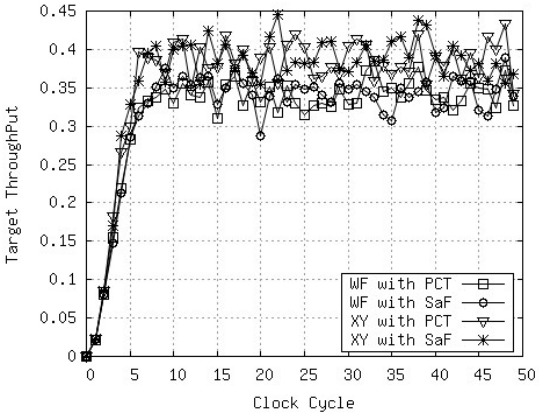

Figure 10: TTP Vs Clock Cycle Using TS

5 Conclusions

In this paper we introduced the reader to a new tool named as Component Based Interconnection Network Simulator (CINSIM) that is designed for linux platforms and proves to be a suitable choice for network-on-chip simulations and performance evaluation. We showed the methodology of designing the XML files using the GUI based schema driven editor 'cinsim-gui' taking a particular 5x5 heterogeneous 2D-Mesh NoC architecture into consideration by selecting different combinations of routing strategies, switching techniques and simulation types provided by the editor. Later we applied the core cinsim to the designed XML files of the 2D-Mesh NoC to get the simulation results related to the delay at targets, source throughput and target throughputs. The results show that the XY routing algorithm with Partial Cut Through switching technique proves to be the best choice for this particular architecture. Hence the CINSIM tool can effectively be used for the design, analysis and selection of different kind of NoC architectures and algorithms before their actual implementations which in turn helps to accelerate the overall process of NoC platform design.

Acknowledements

This work is jointly sponsored by COMSATS Institute Of Information Technology, Pakistan and 国家自然科学基金项目 under the No. 60425413.

REFERENCES

[1] ITRS. International technology roadmap for semiconductors - 2004 edition, http://public.itrs.net/

[2] L. Benini, G. D. Micheli, 2002. Networks on Chip: A New SoC Paradigm. IEEE Computer., 35(1): 70-78.

[3] Kumar S et al., 2002. A Network on Chip Architecture and Design Methodology. In the Proceedings of the 2002 IEEE Computer Society Annual Symposium on VLSI, pp: 117-124.

[4] Jantsch A, Tenhunen H, 2003. Networks on Chip. Kluwer Academic Publishers, pp: 85-106.

[5] Sun Y R, Kumar S, Jantsch A. Simulation and Evaluation for a Network on Chip Architecture Using Ns-2. In the Proceedings of 2002 20th NORCHIP Conference, pp: 6.

[6] Vahdatpour A, Tavakoli A, Falaki M H. Hierarchical Graph: A New Cost Effective Architecture for Network on Chip. In the Proceedings of 2005 EUC(IFIP) , LNCS 3824, pp. 311-320.

[7] Fall K, Varadhan K, 2006. The NS Manual. The VINT Project, available at http://www.isi.edu/nsnam/ns/ns-documentation.html

[8] Altman E, Jimenez T., 2003. NS Simulator for Beginners: Lecture Notes, Univ. de Los Andes, Merida, Venezuela, France.

[9] The network simulator - NS-2, available at http://www.isi.edu/nsnam/ns/

[10] Walter A, Kühm M, Tutsch D, Lüdtke D, Zimmermann C., 2004. CINSim Handbook: Installation and User's Guide. Technische Universität Berlin, Real-Time Systems and Robotics Copyright © 2004-2007, pp: 18-68.

[11] Sheraz Anjum, Jie Chen, Pei-pei Yue and Jian Liu, 2009. A Delay Optimized Architecture for On-Chip Communication. Journal of Electronic Science and Technology of China., 7(2): 104-109.

[12] Sheraz Anjum, Chen Jie, Yue Pei-pei and Liu Jian, 2008. Traffic Modeling and Mapping of H.264 Encoder on 2D-Mesh Vs Application Specific NoC. Journal of System Simulation., 20(10): 2782-2788.