Latency Performance of Chordal Ring Perfect Difference Network Topology for Network on Chip Architecture using NS-2

Mahendra Gaikwad
Professor
B. D. College of Engineering, Sevagram (BDCoE) Dist. Wardha - 442102 , Maharashtra, INDIA

Rajendra Patrikar
Professor
Visvesvaraya National Institute of Technology (VNIT), Nagpur - 440010, Maharashtra, INDIA

Abhay Gandhi
Associate Professor
Visvesvaraya National Institute of Technology (VNIT), Nagpur - 440010, Maharashtra, INDIA

ABSTRACT
The emerging platform-based design paradigm poses enormous challenges to conceptualizing, implementing, verifying and programming today's complex System on Chip (SoC) design. Network-on-Chip (NoC) architecture provides a scalable, high performance and robust infrastructure for on-chip communication. Perfect Difference Network (PDN) based on the mathematical notion of Perfect Difference Set as an asymptotically optimal method for connecting IP cores with network diameter D=2 so that any IP core is reachable from any other IP core in one or maximum two hops. In this paper, we have proposed the Network on Chip architecture based on the mathematical notion of Perfect Difference Network (PDN), which is referred as Chordal Ring PDN topology for Network on Chip architecture. PDN based Network-on-Chip (NoC) architecture is currently viewed as a 'revolutionary' approach as a solution for addressing the design challenges of future high performance nanoscale architecture to provide a scalable, modular and robust infrastructure. In this paper, we present the latency performance of Chordal Ring PDN topology for NoC architecture using simulating tool NS-2.

General Terms
Network on Chip Architecture, Chordal Ring Perfect Difference Network Topology.

Keywords
Perfect Difference Network (PDN), Network on Chip (NoC), Perfect Difference Set (PDS).

1. INTRODUCTION
The Network on Chip has been proposed as a viable alternative for the traditional design-specific buses of today's System on Chips. A Network on Chip is viewed as a collection of computational resources connected through a network where they communicate using packets. In this paper, we have proposed a generic interconnect architecture template for Network on Chip based on Perfect Difference Network (PDN). Perfect Difference Network is based on the mathematical notion of Perfect Difference Set (PDS). In this paper, we have evaluated the performance of Chordal ring PDN topology for Network on Chip architecture in terms of latency and throughput using NS-2, which is an open source, object-oriented and discrete event driven network simulator written in C++ and OTcl (Object Tool Command Language). Most of the researchers and designers extensively use NS-2 for simulation of computer network and evaluate the performance of the computer network. As the feature size in chips scale down, it is becoming increasingly difficult to maintain global synchrony among various components of the chip with a single clock [1]. NoC works on the notion of Globally Asynchronous Locally Synchronous (GALS) [2] where self-timed blocks (IP cores) will communicate with each other. This scenario is analogous to distinct computers in a traditional network. This, in turn, paves the way for decoupling communication from computation, which is one of the key goals of NoCs. In order to address the challenge of growing number of components on a chip, a 5-layered protocol model is proposed to govern the communication in Network on Chip in comparison to the OSI model in [1]. Network on Chip resembles a traditional computer network in the sense that it has resources, routers and links connecting resources to routers and routers to routers, routing strategies (static or dynamic) and protocols, packet forwarding mechanism (flow control), and finally data in the form of packets to be transported on the network. All these characteristics of NoCs are influenced by computer networks making them a perfect candidate to utilize the facilities provided by NS-2 simulator [3]. Network on Chip is viewed as a regular, tightly coupled network of resources on a silicon chip using a packet based communication infrastructure making them identical to a regular network; but on a much smaller geographical scale. Since NoC is composed of IPs (Intellectual Property), a network interface (NI) is needed which can act as a middle layer transforming streams of bits from the resource into packets before sending them to router and vice versa [4], [5]. When a resource has something to send, it starts transmitting the data to the NI, which creates data packets of predefined sizes and then forwards them to the connected router. Each router along the path checks the destination address; and forwards the pack accordingly until it reaches the destination. Likewise, on receiving packets from the network, the last router simply passes them to the connected NI which extracts the bits from it and forwards it to the connected resource. Computer networks have been at the core of research for decades and are still progressing at a very high pace. Similarly, a significant amount of work is being done in parallel computing area which is considered as the parent domain for NoCs. Design paradigms and protocols that are obsolete in networks (or the Internet) however can be a good

choice for NoCs, because of their size and regular structure [6]. Besides the similarities, there are certain differences between the computer networks and NoCs due to which various design choices for NoCs need to be re-evaluated.

2. CHORDAL RING PDN TOPOLOGY FOR NETWORK ON CHIP ARCHITECTURE

In this paper, we have proposed a generic interconnect architecture template for Network on Chip architecture based on Perfect Difference Network [7] referred as Chordal Ring PDN topology. A family of regular graphs of degree three is called Chordal Rings [8]. The number of switches is equal to the number of IP cores or nodes in the network. Figure 1 shows Chordal Ring PDN topology for n=7 based on PDS {0, 1, 3} with PDS order δ=2. Each switch is connected to 2δ neighboring switches and one IP core or node. The number of IP cores or nodes is the function of PDS order (δ) [9], [10]. Total number of IP cores or nodes is equal to $n= \delta^2 + \delta + 1$, where n is the number of IP cores and δ is the order of PDS [11]. The number of switches in Chordal Ring topology is always equal to number of IP cores or nodes in the network. The degree of the switch is equal to (2δ+1) whereas the degree of IP core or node is always equal to one [12]. Any two switches in a PDN are either connected by a ring link directly or via skip link or chord with either one or maximum two hops.

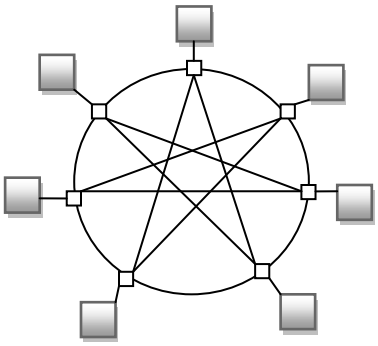
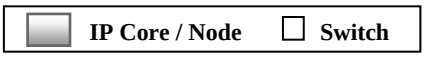


Figure 1: Chordal Ring PDN Topology for NoC Architecture for n=7 based on PDS {0, 1, 3} with PDS order δ=2.



3. MODELING OF CHORDAL RING PDN TOPOLOGY FOR NETWORK ON CHIP ARCHITECTURE

The Chordal Ring PDN topology was modeled and simulated by NS-2 with only built-in option. However, the simplest cases of routing algorithms, routing strategies, queuing algorithms and transmission protocols were defined as simulation constraints. The Chordal Ring PDN topology is easily scaled to different network sizes like $n= \delta^2 + \delta + 1$, where δ is the order of PDS and should be the order of prime. Each IP core or node has unique communication address. Each IP core or node connects with a switch through network interface (NI). The IP cores or nodes and switches are also treated similarly; except that traffic generator is attached to the IP cores or nodes. Switch, IP core or node and link are the three basic elements in the topology. We assume that the buffer size in switches is finite. The overall NoC simulation model consists of four parts: topology, traffic generator, simulation controller and traffic monitor as shown in Figure 2.

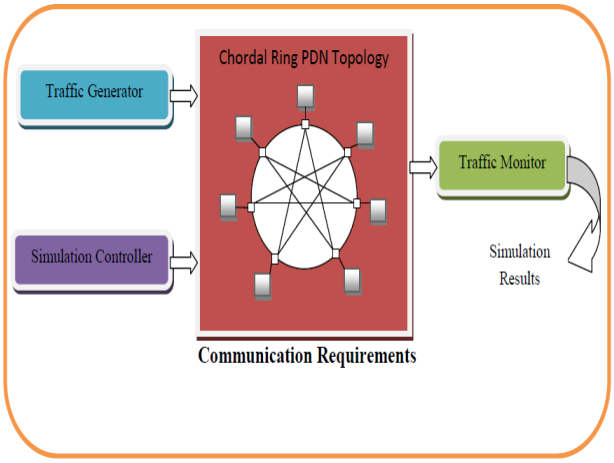


Figure 2: NoC Simulation Model for Chordal Ring PDN Topology

3.1 Traffic Generator

Each source or node was attached to a traffic generator. Every node starts to generate packets with the behavior defined by the attached traffic generator. A traffic generator is set for Constant Bit Rate (CBR) behavior. It is mainly defined through two parameters, start sending packet and stop sending packet. CBR traffic object generates traffic with a deterministic rate. Packets are of a fixed size.

3.2 Simulation Controller

Simulation controller helps to generate the appropriate events. It also helps in selecting subsets of switches for monitoring and tracing.

3.3 Traffic Monitor

Traffic monitor is required for dynamically and instantaneously collecting the simulation results.

4. CHORDAL RING PDN TOPOLOGY GENERATION FOR NETWORK ON CHIP ARCHITECTURE

There are two types of links in NoC, switch-to switch and switch-to-IP core. Switch-to-switch link is further categorized into skip link and ring link for Chordal ring PDN. Switch-to-source link is also called as the local link. A communication path is composed of a set of links identified by the routing table of PDN connectivity. Link delay and maximal bandwidth are two architecture parameters for modeling the links. All the topology parameters are described as a script file in Tcl.

4.1 Chordal Ring PDN Generation Module

The Chordal Ring PDN topology generation consists of five modules. Each module is responsible to perform a predefined task. The flow diagram for generation of Chordal Ring PDN topology is as shown in Figure 3.

- **Module 1:** Generate the number of nodes in the Chordal Ring PDN topology on the basis of PDS

order. It also checks the appropriate value of PDS order.

- **Module 2:** This module finds the corresponding PDS string for a particular PDN node.
- **Module 3:** This module is responsible for determining the routing strategy as per PDN connectivity. It generates the routing table for each switch in the network.
- **Module 4:** Tcl script for showing the data transition in PDN topology as per predefined events.
- **Module 5:** The Chordal Ring PDN topology is displayed in Network Animator (NAM) after compiling the tcl script of NS-2.

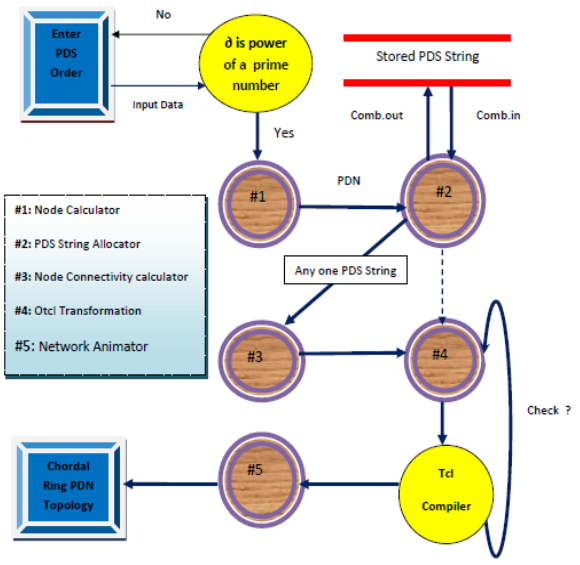


Figure 3: Flow Diagram for generation of Chordal Ring PDN Topology

5. SIMULATION ENVIRONMENT

Network Simulator (Version 2), widely known as NS-2, is an object-oriented, event driven network simulation tool has been developed by University of California, Berkeley to study the dynamic nature of communication networks. University of California and Cornell University has developed the REAL Network Simulator-1 which is the basic foundation for NS-2. Defense Advanced Research Projects Agency (DARPA) supported development of NS through the Virtual Inter-Network Test bed (VINT) project [13]. Currently the National Science Foundation (NSF) has joined the ride in development [13]. The network simulator, NS-2, with its capabilities of describing network topologies, network protocols, routing algorithms, packet scheduling schemes, as well as traffic generation methods, have been broadly used in the field of computer network design and simulation. Moreover, NS-2 also provides routing strategies and the basic network transmission protocols. The NS-2 is written in C++ and OTcl [14]. Tcl script is the front-end interpreter for NS-2 used for constructing commands and configuration interfaces. The NS-2 is open source code, it allows designers to modify the built-in components or add their own designed components to carry out certain functions. Along with these features of NS-2, Network AniMator (NAM) is also provided with NS-2 in order to visualize the flow of messages and the entire system simulation at run time [15]. NS-2 implements network protocol such as UDP and TCP; traffic source behavior such as exponential ON/OFF behavior, CBR and VBR; router queue management mechanism such as Drop Tail, RED (Random Early Detection), FIFO and FQ(Fair Queuing); routing algorithm such as Dijkstra and a lot more.

6. EXPERIMENTAL RESULTS

While making an evaluation of latency performance of Chordal Ring PDN topology for NoC architecture, we have defined the communication scenario in which many traffic source-sink pairs are selected manually and are concurrently active. We have assumed that shorter the distance between two nodes or maximum two-hop path between the nodes, higher is the probability that communication will take place between them. A traffic generator is set for Constant Bit Rate (CBR) behavior. It is mainly defined through two parameters, start sending packet and stop sending packet. CBR traffic object generates traffic according to a deterministic rate. Packets are of a constant size. The static routing strategy was used in each switch. We have considered network protocol UDP for transmission the packets. Drop Tail queue management mechanism was used. A shortest path with maximum two hops has been selected for each traffic pair. We have considered a packet size of 500 bytes and maximum link bandwidth of 1 Mbps in our experiments. For better understanding, we have considered the Chordal ring PDN topology for NoC architecture with n=7 based on PDS {0, 1, 3} having PDS order δ=2 as shown in Figure 4.

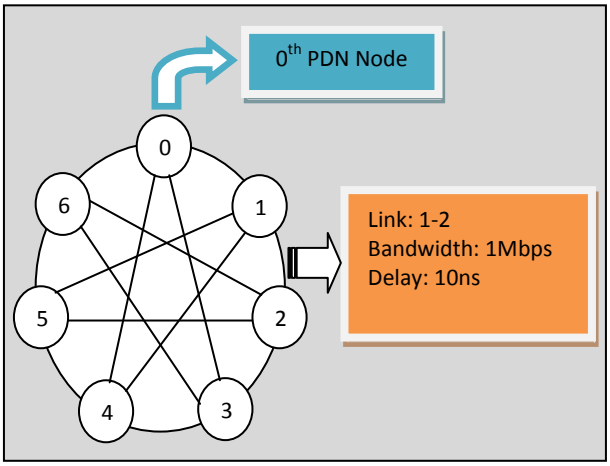


Figure 4: Chordal Ring PDN Topology for n=7 for NoC

Figure 5 shows the communication scenario for Node-0 as a source and Node-5 as a destination.

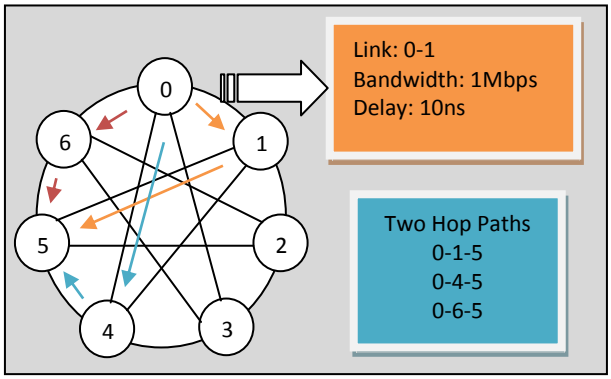


Figure 5: Communication scenario for Node-0 as a source and Node-5 as a destination for Chordal Ring PDN Topology

Flow of packets from Node-0 to Node-5 on one of the PDN path (0-1-5) with two hops is shown in Figure 6. We can also monitor the details of data packets, CBR traffic the time step at which the packets are sending from one node to the other on the link.

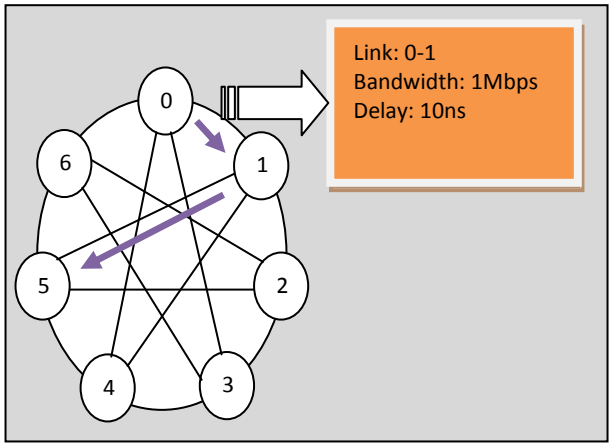


Figure 6: Packet flow from Node-0 to Node-5 for Chordal Ring PDN Topology through path (0-1-5)

In case of PDN n=7 based on PDS {0, 1, 3}, node-0 has the routing table to connect to the other node. The routing table is generated at each node based on the connectivity rule of PDN. Node-0 has three connecting paths to node-5 with two hops. The connecting paths from node-0 to node-5 are (0-1-5), (0-4-5) and (0-6-5) has the maximum double hop connectivity. In case of link failure, particular node is responsible to send the packets on the other available paths having maximum double hop connectivity. The routing of packet from node-0 to node-5 after failure of link between node-1 to node-5 is depicted in Figure 7. Failed link is indicated by the red line and the packets flow through another path according to the routing table of node-0 having maximum two hops from node-0 to node-5 as shown in Figure 7.

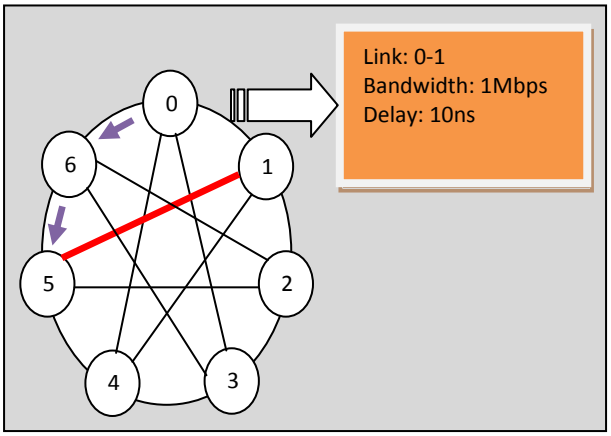


Figure 7: Packet flow from Node-0 to Node-5 for Chordal Ring PDN Topology through (0-6-5) due to link failure

Latency for sending the packets from one node to another either in single hop or in double hop is shown in Figure 8.

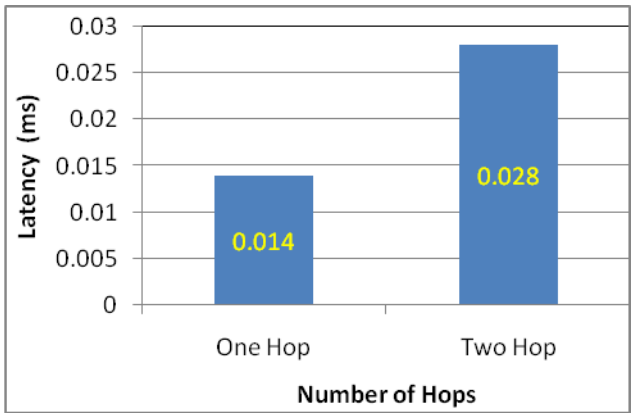


Figure 8: Latency versus number of hops for Chordal Ring PDN Topology for n=7 based on PDS {0, 1, 3} with PDS order δ=2.

7. CONCLUSION

In this paper, we have analyzed the latency performance of Chordal Ring PDN topology for NoC architecture with certain constraints. Constant Bit Rate (CBR) behavior is adopted by traffic generator. Packets are of a fixed size of 500 bytes. The static routing strategy was used in each switch. We have considered network protocol UDP for transmission of the packets. Drop Tail queue management mechanism was used. We have also carried out the high level simulation on Chordal Ring PDN topology using NS-2. We have presented the effective use of NS-2 network simulator for simulating the Chordal Ring PDN topology for NoC architecture. Our argument is that NS-2 is a feasible solution for simulating the Chordal Ring PDN topology at a higher abstraction level. The In-built facilities of NS-2 can effectively facilitate in the design of new protocols other than UDP for Chordal Ring PDN topology for Network on Chip architecture as a future work. The latency performance for Chordal Ring PDN topology has been observed for single and double hop transmission between each traffic pair.

8. REFERENCES

[1]. L. Benini and G. De Michelli, "Networks on Chip: A new paradigm for component-based MPSoC design", Multiprocessors Systems on Chips, edited by A. Jerrraya and W. Wolf, Morgan Kaufman, 2004, pp. 49-80.

[2]. J. Muttersbach, T. Villiger, H. Kaeslin, N. Felber and W. Fichtner, ``Globally-Asynchronous Locally Synchronous Architectures to Simplify the Design of On-Chip Systems", Proc. ASIC/SOC pp. 317- 321, 1999.

[3]. Yi-Ran, Shashi Kumar, Axel Jantsch, "Simulation and Evaluation for a Network on Chip Architecture using NS-2", Proceeding of 20th IEEE Norchip Conference, November 2002.

[4]. Partha Pratim Pande, Cristian Grecu, Andre Ivanov, Resve Saleh, and Giovanni De Michelli, ``Design, Synthesis, and Test of Networks on Chips", IEEE Design and Test, September/October 2005 (Vol. 22, No. 5), pp. 404 - 413.

[5]. Luca Benini and Giovanni De Micheli (2002), "Networks on Chips: A New SoC Paradigm", IEEE Computer, January 2002, pp. 70-78.

[6]. L. Benini and D. Bertozzi (2002), "Network-on-Chip architectures and design methods", IEE Proceeding on Computer Digit. Tech., Vol. 152, No. 2, March 2002, pp. 261-272.

[7]. B. Parhami, M. A. Rakov, "Performance, Algorithmic, and Robustness Attributes of Perfect Difference Networks," IEEE Transactions on Parallel and distributed systems, Vol. 16, No. 8, pp. 725-736, August 2005.

[8]. Arden B. W. & H. Lee, "Analysis of Chordal Rings Networks", IEEE Trans. Computers, Vol. No. 30, No. 4, pp. 291-295, April 1981.

[9]. B. Parhami, M. A. Rakov, " Perfect Difference Networks and Related Interconnection Structures for Parallel and Distributed Systems," IEEE Transactions on parallel and distributed systems, Vol. 16, No. 8,pp. 714-724 August 2005.

[10]. B. Parhami, "Introduction to Parallel Processing: Algorithms and Architectures", Plenum Press, 1999.

[11]. Mahendra Gaikwad and Rajendra Patrikar, "Perfect Difference Network for Network-on-Chip Architecture", International Journal of Computer Science and Network Security (IJCSNS), Vol. 9, No. 12, December 2009, pp. 286-290.

[12]. Mahendra Gaikwad, Rajendra Patrikar and Abhay Gandhi, "Energy-aware Network-on-Chip architecture using Perfect Difference Network", International Conference on Advances in Computer Engineering-ACE 2010, IEEE Computer Society, DOI 10.1109/ACE.2010.90, June 21-22, 2010, pp. 134-138.

[13]. Virtual Inter-Network Test Bed (VINT) Project and National Science Foundation http://www.isi.edu/nsnam/ns

[14]. Object oriented TCL script and NS Manual http://www.isi.edu/nsnam/ns/ns-documentation.html

[15]. Network AniMator (NAM) and Tcl Tutorial http://jan.netcomp.monash.edu.au/ProgrammingUnix/tcl/tcl_tut.html

[16]. OTcl Tutorial http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/ -oTcl tutorial

[17]. Tcl Tutorial: http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html

[18]. Tcl Manual: http://www.tcl.tk/man/tcl8.6/TclCmd/contents.html