International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-6, January 2012

Bursty Communication Performance Analysis of Network-on-Chip with Diverse Traffic Permutations

Naveen Choudhary

***Abstract*— To satisfy the increasing communication demands of complex VLSI circuits, Network on Chip (NoC) has been introduced as a new paradigm, where processing and communication can be independently catered by communication infrastructure design. Network on Chip proposes to establish a communication infrastructure for the complex VLSI circuit in such a way that communication between any nodes in the circuit is possible even if the circuit blocks are not directly connected by a direct channel. Each circuit block of the whole circuit can be assumed as an Intellectual Property (IP) which may be a microprocessor, memory or ASIC, etc. In this paper the performance of standard 2D mesh NOC is analyzed for bursty communication traffic for various traffic or topology mapping patterns such as butterfly, transpose etc over a NOC simulation framework. The routing for the NoC is assumed to be XY and OE.**

***Index Terms*—NoC, Simulation, VLSI, Transpose, Traffic latency**

I. INTRODUCTION

The leap and bound progress in the VLSI design technology and the large number of transistors obtainable on a single chip permit designers to fabricate complex SoCs with hundreds of IP blocks. The IPs can be CPU or DSP cores, memory blocks, or ASICs. The affluence of the computational resources on a chip also produces a huge demand on the communication requirement with the various blocks of the chip. Moreover the decreasing feature size in VLSI circuit design makes interconnect delay and power consumption the major factors for optimization of modern VLSI systems. Another consequence of these advancements in VLSI is the complexity in optimizing the interconnect due to the deterioration effects such as crosstalk, electro-magnetic interference and soft errors. Till now the shared-bus systems were the system communication infrastructure but such bus structure raises may problems as the number of processing blocks per chip increases such as large capacity load for the bus drivers leading to large delays and huge power consumption. Moreover the shared bus architecture is not scalable. The scalability of switch-based networks and packet-based communication in parallel computing and Internet has stimulated the researchers to propose the Network-on-Chip (NoC) architecture as a viable solution to the complex on-chip communication problems [1]. NoC is the emerging and promising solution for complex VLSI chip design due to the reality that the traditional design techniques have faced grave challenges and restrictions as the number of on-chip VLSI components increases. NoC tries to bring macro network communication methodologies to the on-chip communication. The NoC design methodology is to establish the communication infrastructure in advance and then plot the computational resources to it may be with the help of resource dependent interfaces. Intellectual Property (IPs) in a NoC are connected by switches/routers and network channels, and information over the NoC is communicated in the form of packets.

In Section 2 basic NOC model is described. Section 3 describes the routing in 2D-mesh NOC and also describes the various Traffic or mapping permutations used in the experimental analysis. Section 4 presents some simulation experimental results for 2D-mesh NoC for bursty traffic for varying traffic permutations and in Section 5 we conclude.

II. NETWORK ON CHIP MODEL

Chip design has four distinct aspects: computation, memory, communication, and I/O. As processing power has increased and data intensive applications have emerged, the challenge of the communication aspect in single-chip systems, Systems-on-Chip (SoC), has attracted increasing attention. NoC does not constitute an explicit new alternative for intra-chip communication but is rather a concept which presents a unification of on-chip communication solutions. Figure 1 shows a sample NoC structured as a 4-by-4 grid which provides global chip level communication. Instead of busses and dedicated point-to-point links, a more general scheme is adapted, employing a grid of routing nodes spread out across the chip, connected by communication links. Figure 1 shows a simplified perspective of NoC which contains the following fundamental components.

- Core/IP is responsible for carrying out computation and for generating traffic for communication with other IPs in the network.
- Network adapters implement the interface by which cores (IP blocks) connect to the NoC. Their function is to decouple computation (the cores) from communication (the network).
- Routing nodes/Switch route the data according to chosen protocols. They implement the routing strategy.
- Links/Channels connect the nodes, providing the raw

Dr. Naveen Choudhary, Department of Computer Science and Engineering, College of Technology and Engineering, MPUAT, Udaipur, India. (e-mail: naveenc121@yahoo.com).

bandwidth. They may consist of one or more logical or physical channels.

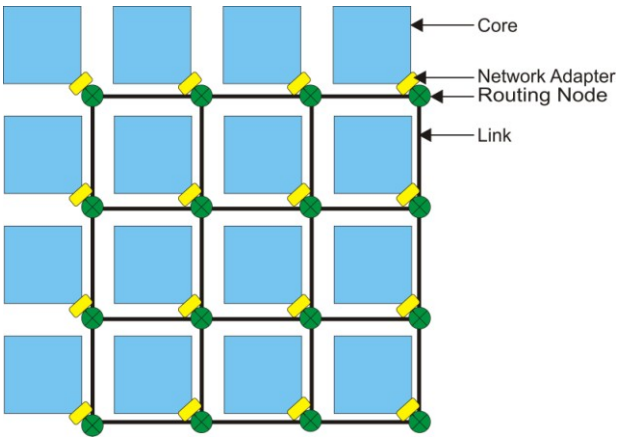

Fig 1: Topological illustration of a 4\times4 grid structured NoC, indicating the fundamental components

NoC architecture can be characterized with the help of network topology, routing methodology, flow control schemes, switching and the technique applied to ensure quality-of-service for data transmission.

The **topology** of a NoC specifies the physical organization of the interconnection network. It defines how nodes, switches and links are connected to each other. Topology for NoCs can be classified into two broad categories: 1) direct network topologies, in which each node (switch) is connected to at least one core (IP), and 2) indirect network topologies, in which we have a subset of switches (nodes) not connected to any core (IP) and performing only network operation. Both direct and indirect topology can be regular like meshes, tori, k-ary n-cubes and fat trees or irregular customized application-specific topology. Most NoCs implement regular forms of network topology that can be laid out on a chip surface (a 2-dimensional plane) for example, k-ary 2-cube (where k is the degree of each dimension and 2 is the number of dimensions) commonly known as grid-based topologies. Besides the form, the nature of links adds an additional aspect to the topology. In k-ary 2-cube networks, popular NoC topologies based on the nature of link are the mesh which uses bidirectional links and torus which uses unidirectional links. For a torus, a folding can be employed to reduce long wires. In the NOSTRUM NoC presented in Millberg et al. [4], a folded torus is discarded in favor of a mesh with the argument that it has longer delays between routing nodes. Generally, mesh topology makes better use of links (utilization), while tree-based topologies are useful for exploiting locality of traffic. In this work 2D-Mesh topology is assumed for the experimental analysis.

The NoC **switching strategy** determines how data flows through the routers in the network. It defines the granularity of data transfer and the switching technique. NoCs use packet switching as the fundamental transportation mode. Packet switching is a communications paradigm in which packets are routed between nodes over data links shared with other traffic. In each network node, packets are queued or buffered, resulting in variable delay. This contrasts with the other principal paradigm, circuit switching, which sets up a limited number of constant bit rate and constant delay connections between nodes for their exclusive use for the duration of the communication [5]. In packet switching, instead of establishing a path before sending any data, the packets are transmitted from the source and make their way independently to the receiver, possibly along different routes and with different delays. There are mainly three kinds of switching schemes [5]: store-and-forward, virtual cut-through and wormhole switching.

Store-and-forward is a telecommunications technique in which information is sent to an intermediate station where it is kept and sent at a later time to the final destination or to another intermediate station. The intermediate station or node in a networking context, verifies the integrity of the message before forwarding it. In general, this technique is used in networks with intermittent connectivity, especially in the wilderness or environments requiring high mobility. It may also be preferable in situations when there are long delays in transmission and variable and high error rates, or if a direct, end-to-end connection is not available.

Virtual cut-through switching is a switching method for packet switching systems, wherein the intermediate switch starts forwarding a frame (or packet) before the whole frame has been received if there is ample space for the whole packet in the later switch (switch where the packet is being forwarded), normally as soon as the destination address is processed. This technique reduces latency through the switch, but decreases reliability.

Wormhole switching combines packet switching with the data streaming quality of circuit switching to attain minimal packet latency. The node looks at the header of the packet to determine its next hop and immediately forwards it. The subsequent flits are forwarded as they arrive. This causes the packet to worm its way through the network, possibly spanning a number of nodes, hence the name. The latency within the router is not that of the whole packet. A stalling packet, however, has the unpleasant expensive side effect of occupying all the links that the worm spans. Because of the limited silicon resources and the low-latency requirements for typical NoC applications, most NoC architectures use wormhole switching scheme for the on-chip routers. In this work for experimental analysis the wormhole switching is assumed.

Peh and Dally [6] have defined **flow control** as the mechanism that determines the packet movement along the network path. Thus it encompasses both global and local issues. Flow control mainly addresses the issue of ensuring correct operation of the network. In addition, it can be extended to also include issues on utilizing network resources optimally and providing predictable performance of communication services [5]. In the following, first the concept of virtual channels and their use in flow control is discussed and later buffering issues are briefly discussed.

Virtual channels (VCs): VCs are the sharing of a physical channel by several logically separate channels with individual and independent buffer queues. Generally 2 to 16 VCs per physical channel have been proposed for NoCs. Their implementation results in an area and possibly also power and

latency overhead due to the cost of control and buffer implementation. There are however a number of advantageous of using VCs as shown below.

- Avoiding Deadlocks: Since VCs are not mutually dependent on each other, by adding VCs to links and choosing the routing scheme properly, one may break cycles in the resource dependency graph and thus deadlocks can be avoided.
- Optimizing Wire Utilization: In future technologies, wire costs are projected to dominate over transistor costs. Letting several logical channels share the physical wires, the wire utilization can be greatly increased. Advantages include reduced leakage power and wire routing congestion.
- Improving Performance: VCs can generally be used to relax the inter-resource dependencies in the network, thus minimizing the frequency of stalls.
- Providing QoS Services: Quality-of-service (QoS) can be used as a tool to optimize application performance. VCs can be used to implement such services by allowing high priority data streams to overtake those of lower priority or by providing guaranteed service levels on dedicated connections .

Buffers are an integral part of any network router. In by far the most NoC architectures, buffers account for the main part of the router area. As such, it is a major concern to minimize the amount of buffering necessary under given performance requirements. There are two main aspects of buffers (i) their size and (ii) their location within the router. In Kumar et al. [3], it is shown that increasing the buffer size is not a solution towards avoiding congestion. At best, it delays the onset of congestion since the throughput is not increased. The performance improves marginally in relation to the power and area overhead. On the other hand, buffers are useful to absorb bursty traffic, thus leveling the bursts.

Generally speaking, QoS in NoCs refers to the level of commitment for packet delivery. Such a commitment is mainly in the form of bounds on performance (bandwidth, delay and jitter) since correctness of the transfer and completion of the transition (packet transmission) is often the basic requirements of on-chip packet transfers. Transaction correctness is concerned with packet integrity (corruption-less) and in order transfer of packets from the source to the intended destination. Transaction completion is concerned with ensuring freedom from deadlocks or livelocks. In terms of bounds on performance, QoS requirements can be classified into three basic categories as best-effort, guaranteed and differentiated. In best effort (BE), only the correctness and completion of communication is guaranteed and no other commitments can be made. Packets are delivered as quickly as possible, over a connectionless (i.e. packet switched) network, but worst case times cannot be guaranteed. A guaranteed service (GS) such as guaranteed throughput (GT), make a tangible guarantee on performance, in addition to the basic guarantees of correctness and completion for communication. Guaranteed service is typically implemented using connection-oriented switching (i.e., virtual circuit switching). A differentiated service prioritizes communication according to different categories and the NoC switches employ priority based scheduling and allocation policies. Unlike guaranteed services, such priority based approaches can enable higher resource utilization, but cannot provide strong guarantees.

Poplavko et al. [7] proposed a guaranteed service model for reconfigurable NoCs. Hansson et al. [8] also proposes a guaranteed service technique that depends on buffer dimensioning. Hansson et al. [9, 10] and Murali et al. [11] propose QoS control methodologies during architecture reconfiguration by use case switches.

III. Routing and Traffic Permutations

Routing is the process of selecting paths in the computer network, along which data or physical traffic is sent. Routing algorithms are responsible for correctly and efficiently routing packets or circuits from the source to destination [2]. In other words if switching is mere transport of data than routing is the intelligence behind it, that is, it determines the path of the data transport.

Routing schemes are usually categorized into two folds: deterministic routing and adaptive routing. Deterministic routing means routing paths are completely determined statically and the packets follow the same path for a given source-destination pair, while in adaptive routing, the paths are determined dynamically depending on network congestion conditions. Deterministic routing has the design simplicity and low latency under low network traffic, but performance throughput degrades when there is network congestion. Adaptive routing uses alternative paths when network is congested, which provides higher throughput, although it will experience higher latency if network congestion is low. In NoCs, the routing scheme usually selects candidates among the routing paths that have minimum distance between the source and destination nodes.

In switch-based networks, packets usually traverse several switches before reaching the destinations. However, it may happen that some packets are not able to reach their destination, even if there exists, a fault-free path connecting the source and destination for every packet. Assuming that the routing algorithm is able to use those paths, there are several situations that may prevent packet delivery like livelock, starvation and deadlock to name the few. Among these the deadlock is the most significant. Deadlock can be defined as a situation where each packet in the network whose header has not already arrived at its destination are waiting for the resources (channels, buffers) to be freed by other packets in the network while keeping the resources currently storing the packet. This can lead to indefinite waiting of packets without any movement in the network, leading to a deadlock situation. The deadlock situations arise from cyclic wait dependencies caused by typical flow-control schemes in order to prevent buffer overflow. Starvation may be defined as a situation when a packet may be permanently stopped if traffic is intense and the resources requested by it are always granted to other packets. Similarly livelock may be defined as a situation when some packets are not able to reach their destination, because the channels required to do so are always occupied by other packets. Livelock can be avoided by always using only the minimal path or by limiting the number of misrouting

operation.

The XY routing [5, 12] and odd-even routing [13] are the most used deadlock free routing algorithms for the popular 2D-mesh based NoCs. They are both theoretically guaranteed to be free of deadlock and livelock. The XY routing strategy can be applied to regular two-dimensional mesh topologies without obstacles. The position of the mesh nodes and their nested network components is described by coordinates, the x-coordinate for the horizontal and the y-coordinate for the vertical position. A packet is routed to the correct horizontal position first and then in vertical direction. XY routing produces minimal paths without redundancy, assuming that the network description of a mesh node does not define redundancy. The odd-even turn model is a shortest path routing algorithm that restricts the locations where some types of turns can take place such that the algorithm remains deadlock-free. More precisely, the odd-even routing prohibits the east to north and east to south turns at any tiles located in an even column. It also prohibits the north to west and south to west turns at any tiles located in an odd column.

The following traffic permutations or topology mappings were used in this paper to analyze the performance of 2D-Mesh NoC with bursty traffic and XY and OE routing function.

In this work, we are using the following traffic Patterns to analyze the performance of 2D-Mesh NoC with XY and OE routing.

The ith **cube permutation** complements the i^{th} bit of the index. In this paper we have assumed the butterfly permutations for i = 0, 1 and 2.

The i^{th} **butterfly permutation** interchanges the 0^{th} and the i^{th} digit of the index. In this paper we have assumed the butterfly permutations for i = 0, 1 and 2.

The i^{th} **baseline permutation** performs the cyclic shifting of the i+1 least significant digits in the index from left to the right for one position. In this paper we have assumed the baseline permutations for i = 0, 1 and 2.

IV. Experimental Results

For analysis and comparison of the performance of the NoC on various Traffic Patterns, a discrete event, cycle accurate NoC simulator NIRGAM [14] is used. NIRGAM allows to experiment with various options available at every stage of the design be it topology, switching techniques, virtual channels, buffer parameters, routing mechanisms or traffic generation applications. The simulator can generate performance metrics such as latency and throughput for a given set of choices. In this paper we are trying to analyze the performance of 2D Mesh topology with bursty traffic with uniform burst length 1o clock cycles and off period of 120 clock cycles for various trafiic permutations with chosen routing function as XY and OE. A 4 × 4 2D Mesh NoC is assumed for all the experimental results. The results presented in Figure 2 shows all the cubic permutations (0th, 1^{st} and 2^{nd}) for XY and OE routing for 4 × 4 2D Mesh NoC. As is evident from the graph with uniform bursty traffic the XY routing tends to perform better and exhibits on average lower per flit latency in comparison to OE routing. However 0^{th} cube permutation shows least average per flit latency of 55.33 clock cycles.

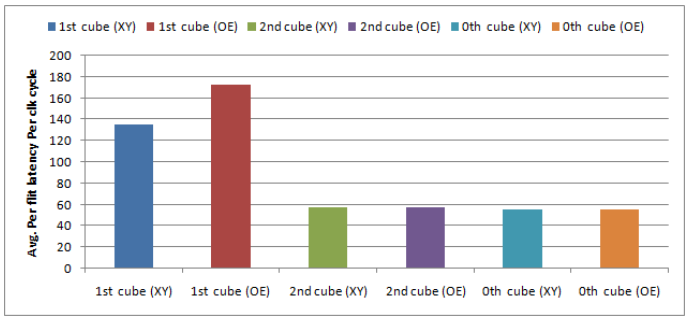

Fig 2: Cubic permutation performance results with bursty traffic for 4 × 4 2D-Mesh NoC with XY and OE routing

The results presented in Figure 3 shows all the butterfly permutations (0th, 1st and 2nd) for XY and OE routing for 4 × 4 2D Mesh NoC. As is evident from the graph with uniform bursty traffic the XY routing tends to perform better and exhibits on average lower per flit latency in comparison to OE routing. However 0th butterfly permutation shows least average per flit latency of 35.73 clock cycles.

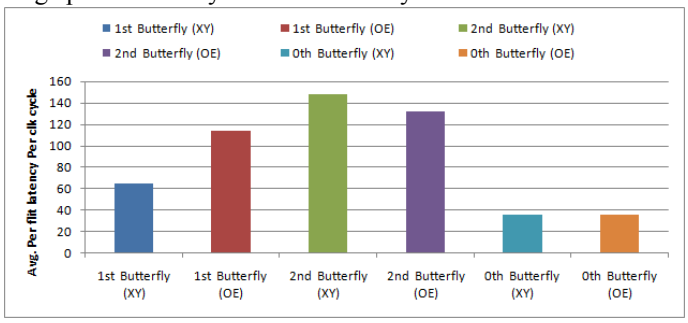

Fig 3: Butterfly permutation performance results with bursty traffic for 4 × 4 2D-Mesh NoC with XY and OE routing

The results presented in Figure 4 shows all the baseline permutations (0th, 1st and 2nd) for XY and OE routing for 4 × 4 2D Mesh NoC. As is evident from the graph with uniform bursty traffic the XY routing tends to perform better and exhibits on average lower per flit latency in comparison to OE routing. However 0th baseline permutation shows least average per flit latency of 35 clock cycles.

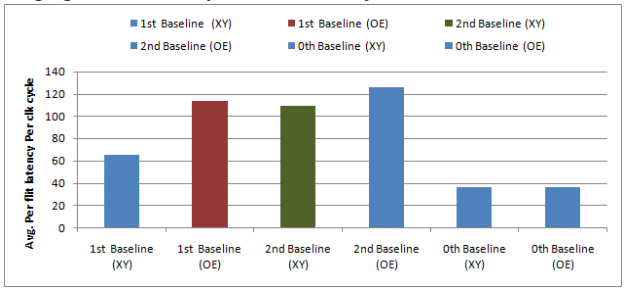

Fig 4: Baseline permutation performance results with bursty traffic for 4 × 4 2D-Mesh NoC with XY and OE routing

V. Conclusion

The paper presents the Network on Chip communication performance for bursty traffic patterns on various traffic permutations for 2D mesh NOC with XY and OE routing functions. We have observed that the communication performance of the 2D-mesh based NoC for bursty traffic is deeply affected by the varying traffic permutation for the used routing function. Which basically helps us conclude that if appropriate traffic permutation are chosen for the bursty

traffic in accordance to the routing function may lead to major gain in communication performance of the NoC.

REFERENCES

[1] W. J. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks", in Proceedings DAC, pp. 684-689, June 2001.
[2] L. Benini, G. DeMicheli., "Networks on Chips: A New SoC Paradigm", In IEEE Computer Vol. 35, No. 1 pp. 70–78, January 2002.
[3] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A Network on Chip Architecture and Design Methodology", In Proceedings of VLSI Annual Symposium (ISVLSI 2002), pp. 105–112, 2002
[4] M. Millnerg, E. Nilsson, R. Thid, A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network-on-chip," in Proceedings of Design, Automation and Testing in Europe Conference (DATE). IEEE, pp. 890–895, 2004.
[5] J. Duato, S. Yalamanchili, L. Ni, Interconnection Networks : An Engineering Approach, Elsevier, 2003.
[6] L.-S. Peh, W. J. Dally, "A delay model for router microarchitectures," in IEEE Micro 21, pp. 26–34, 2001.
[7] P. Poplavko, T. Basten, M. Bekooij, J. van Meerbergen, B. Mesman, "Task level timing models for guaranteed performance in multiprocessor networks-on-chip," in CASES '03: Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded systems, New York, USA, pp. 63-72, ACM, 2003.
[8] A. Hansson et al., "Applying data flow analysis to dimension buffers for guaranteed performance in networks on chip," in Proceedings of the International Symposium on Networks-on-Chip (NOCS), April 2008.
[9] A. Hansson, M. Coenen, and K. Goossens, "Undisrupted quality-of-service during reconfiguration of multiple applications in networks on chip," in Proceedings of Design, Automation & Test in Europe Conference & Exhibition, pp. 1–6, DATE 16-20, April 2007.
[10] A. Hansson and K. Goossens, "Trade-offs in the configuration of a network on chip for multiple use-cases," 1^{\text{st}} International Symposium on NoC (NOCS 2007), pp. 233–242, 7-9May 2007.
[11] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "A methodology for mapping multiple use-cases onto networks on chips," in Proceedings Design, Automation and Test in Europe, 2006 (DATE '06), vol. 1, pp. 1–6, 6-10 March 2006.
[12] W. Zhang, L. Hou, J. Wang, S. Geng, W. Wu, "Comparison research between XY and odd-even routing algorithm of a 2-dimension 3\times3 mesh topology network-on-chip," in GCIS'09, pp. 329–333, 2009.
[13] G. M. Chiu, "The odd-even turn model for adaptive routing," in IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 7, pp. 729–738, Jul 2000.
[14] Lavina Jain et al., "NIRGAM: A Simulator for NoC Interconnect Routing and Application Modelling, Proc. DATE 2007, 2007

Dr. Naveen Choudhary received his B.E, M.Tech and PhD degree in Computer Science & Engineering. He completed his M.Tech from Indian Institute of Technology, guwahati, India and PhD from Malviya National Institute of technology, Jaipur, India in 2002 and 2011 respectively. Currently he is working as Associate Professor and Head, department of Computer Science and Engineering, College of Technology and Engineering, Maharana Pratap University of Agriculture and Technology, Udaipur, India.
His research interest includes Interconnection Networks, Network on Chip, Distributed System and Information Security. He is a life member The Indian Society of Technical Education, Computer Society of India and The Institution of Engineers, India. E-mail: naveenc121@yahoo.com