Performance Evaluation of Traffic Permutations for Mesh On-Chip Networks

Naveen Choudhary
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

Dharm Singh
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

Abhilasha Sharma
Department of CSE
College of Technology and Engineering, MPUAT
Udaipur, India

ABSTRACT

Networks-on-Chip (NoC) is recently proposed as an alternative to the on-chip bus to meet the increasing requirement of complex communication needs in Systems-on-Chip (SoC). Using on-chip interconnection networks in place of ad-hoc global wiring, structures the top level wires on a chip and facilitates modular design. The structured network wiring gives well-controlled electrical parameters that eliminate timing iterations and enable the use of high-performance circuits to reduce latency and increase bandwidth. Using a network to replace global wiring has advantages of structure, performance, and modularity. With this approach, system modules (processors, memories, peripherals, etc.) communicate by sending packets to one another over the network. In NoC, nodes are arranged in the topology such that communication between any nodes is possible even though they are not directly connected. Each node is a IP core which can be a DSP, Microprocessor, Memory along with routing function which is responsible for forwarding the data packet to the neighboring node.

Keywords

Network on Chip, XY routing, Traffic Patterns, Intenerconction Networks, Simulation.

INTRODUCTION

With the development of IC manufacture technology and the increasing demand for more processing power in consumer electronics, On chip designs are becoming more and more complex and the number of processing cores are on the rise to support evolving standards and new applications. Computation and communication complexity is skyrocketing, and scalability-centric design paradigms are critically needed. How to improve the communication efficiency between the IP/PEs (processing elements) has become one of the key challenge to be addressed in such scenario. Traditionally, bus based and point-to point (P2P) on chip communication architectures has been most widely used in System-on-chip (SoC) designs.

Bus based or hierarchy bus based architecture use simple communication protocols to reduce design complexity and can only connect a very few i.e. tens of IP cores in a cost-efficient manner. However it lacks the scalability in terms of performance and energy efficiency due to the large capacitive load of the bus drivers that cause large delay and energy consumption. P2P architectures improves the system performance at the expense of dedicated channels between every pair of IPs, but are not scalable with exponential increase in number of nodes leading to high design complexity, cost and performance degradation. To address such communication infrastructure design challenges of on chip communication, the Network on Chip (NoC) was proposed as an alternative solution [1].

NoCs are based on networks instead of using traditional buses or peer to peer connections. It uses packet based scheme to forward messages in an on-chip communication networks instead of dedicated connection. Using a network to replace global interconnects has the benefits of structure, performance, and modularity. In NoC, nodes are arranged in the topology such that communication between any nodes is possible even though they are not directly connected.

Each node consist of a IP core which can be a DSP, Microprocessor, Memory and/or a Router, which may be responsible for forwarding the data packet to the neighbouring node [2]. Each IP is placed in a rectangular tile on the chip and can communicate with any other IP on the chip including its immediate neighbors. The network.(NoC) research addresses the on chip communication issue in ultra lagre scale SoC, and basically involves a move from computation-centric to communication-centric scalable design in the nano scale regime.

A NoC is described by 5 main characteristic: Topology (arrangement of the network elements), Switching (the way the data is transferred from the input port to the output port), Routing (Determine message path), Flow Control (Dynamic allocation of the channels and flow control of communication), Buffering (The way the packets are stored while waiting for forward transmission), Arbitration (plans the use of channels and buffers). NoC stands out in design modularity, low power consumption reusability, scalability and parallelism in communications [3].
A common workflow of networks-on-chip design includes network topology synthesis, communication channel width and buffer size selection, IP core mapping, packet routing, switching, and real-time scheduling of the task executions and communications. Most researchers advocate the use of traditional regular networks like meshes, tori or trees as architectural templates which have gained a high popularity in general-purpose parallel computing.
NoC are the design of the topology or structure of the network and setting of various design parameters (such as frequency of operation or link-width). The communication infrastructures based on NoC design broadly has Network Elements(NE) and Network Interfaces(NI) as major components. The packet travel across the Network Elements while the Network Interfaces provide an interface with the processing units of IP.

1. NoC COMMUNICATION MODEL

NoC communication paradigm proposed a Layered Design on chip communication which are unlike the ISO/OSI model. NoC architecture consists of Data link(Switching) layer, Network(Routing) layer and Physical layer(Contention issues, reliability issues, grouping of physical layer bits, e.g. "flits). The layered approach helps NoC to address various communication challenges in modular and efficient manner.

1.1 Architecture of NoC

From a component –based view, there are several basic building blocks of a typical NoC system, namely, processing node (also called resources), routing node (representing network on chip) and resource network interface (RNI) [4]. Figure1 is a sample NoC structured as a 4-by-4 grid which shows the interconnection architecture of these blocks providing global chip level communication.

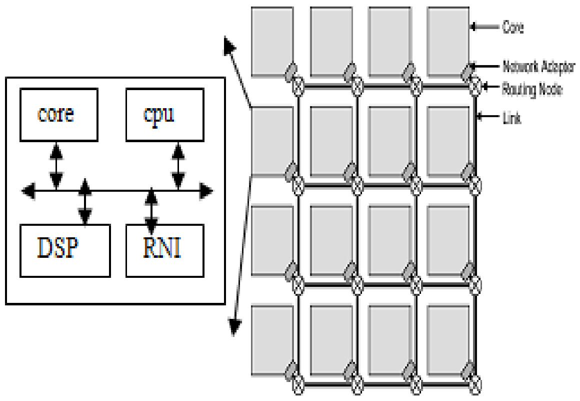

Fig 1: Topological illustration of a 4-by-4 grid structured NoC, indicating the fundamental components

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. A simplified perspective of NoC containing the fundamental components can be described as follows. 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 route the data according to chosen protocols. They implement the routing strategy.

Processing Node implements the computing functionalities of CPU, DSP or other specific applications. Links connect the nodes, providing the raw bandwidth. They may consist of one or more logical or physical channels. Resource Network Interface connect the processing node and routing node. It can transmit packets from processing nodes to routing nodes or receive them from routing node to processing node. In designing the NoC, there are some other important issues such as network topology, routing algorithm and packet formats. In current work, a popular 2-dimension mesh topology is chosen along with XY routing algorithm to analyze its performance with various traffic scenarios.

1.2 Switching

The benefits of switching vary from network to network. Understanding traffic patterns is very important to switching as the efficient flow traffic is the responsibility of switching layer. Network response times (the user-visible part of network performance) suffers as the load on the network increases, and under heavy loads small increases in user traffic often results in significant decreases in performance.

Wormhole flow control, also called wormhole switching or wormhole routing is a system of simple flow control in computer networking based on known fixed links. It is a subset of flow control methods called Flit-Buffer Flow Control. Although wormhole switching and wormhole routing are used to describe the same phenomenon, this technique does not direct any path or route to reach some specific destination over the network. However, it only generates a decision about the timing for routing packets from the router. The term wormhole switching is sometimes confused with cut-through switching but they are different in the sense that cut-through flow control assigns channel bandwidth and buffers on packet level, while wormhole flow control allocates them on flit level. When the head flit arrives at a node, it must acquire three resources before it can be forwarded to the next node along a route. A virtual channel (channel state) for the packet, flit buffer, bandwidth of communication channel according to the flit size. Various flits of a packet the virtual channel acquired by the head flit and have to acquire flit buffers and bandwidth for flit transmission. Tail flits behave like body flits but also act as a releaser of the reserved resources.

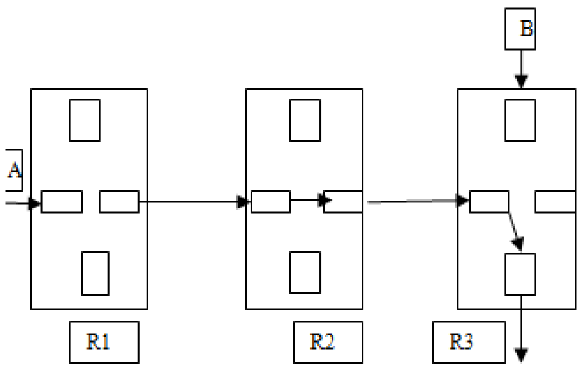

Fig 2: An example of a blocked wormhole switched message

In wormhole switching, message packets are pipelined through the network. However the buffer requirement within the routers are substantially reduced over the requirements for VCT switching [5]. A message packet is broken up into flits. The flit is the unit message flow control and input and output buffers at the routers are typically large enough to store few flits. The header flit again builds a path in the network, which the other flits follow in pipeline. The sequence of buffers and links occupied by flits of a given packet forms the wormhole. However, the length of the worm here is proportional to the number of flits in the packet. Typically, it may spans the whole path between the source and destination if packet size is very large. If the header cannot proceed due to busy output channels, the whole chain of flits gets stalled, occupying flit buffers in routers on the path traversed so far and blocking other possible communications. In presence of blocking the flits of the packet are pipelined through the network. During blocking that is when the output channel is busy the packet is blocked "in place" for example Figure 2 shows that packet being transmitted from router R1, R2, R3. Assuming input and output buffers to be 2 flit deep. At router R3, Packet A requires an output channel that is being used by some other packet. Therefore packet A blocks in place. The small buffer sizes at each node cause the flits of the packet to occupy buffers at multiple routers leading to blocking of other packets.

The base latency of a wormhole switched message can be computed as follows:

$$T_{wormhole} = D(t_r + t_s + t_w) + \max(t_s + t_w)[L/W]$$

The expression assumes flit buffers at the routers inputs and outputs.[8] Once the header flit arrives at the destination, the message pipeline cycle time is determined by the maximum of switch delay and wire delay. For an input-only or output-only buffering this cycle time would be given by the sum of the switch and wire delays.

1.3 Virtual Channel (VCs)

VCs are the mechanism for sharing of a physical channel by several logically separate channels with individual and independent buffer queues. Buffers are commonly operated as FIFO queues. Therefore, once a message occupies a buffer for a particular channel number other message can access this physical channel, even if the message is not blocked. Alternatively, a physical channel may support several logical or virtual channels multiplexed across the physical channel. Each unidirectional virtual channel is realized by an independently managed pair of message buffer.

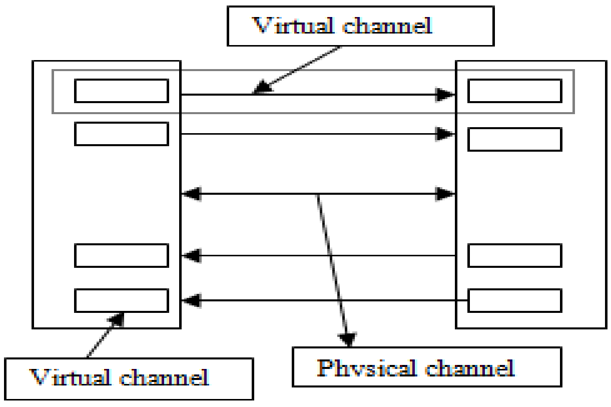

Fig 3: Virtual Channel and Physical Channel for NoC Communication Infrastructure

The Figure3 shows two unidirectional Virtual channels in each direction across the physical channel. Logically each virtual channel operates as if each were using distinct physical channel operating at half the speed. Virtual channels were originally introduced to solve the problem of deadlock [5]. Deadlock is a state where no message can advance because each message requires a channel occupied by another message. Since VCs are not mutually dependent on each other, by adding virtual channel to links and choosing the routing scheme properly, one may break cycles in the resource dependency graph. Virtual channel can also be of great use to improve message latency and network throughput. VCs are use for optimizing wire utilization by letting several logical channels share the same physical wires increases the wire utilization thus reducing leakage power and wire routing congestion. virtual channel can generally be used to relax the inter-resource dependencies in the network, thus minimizing the frequency of blockages. Virtual channel are also used to implements quality of service by allowing high priority data streams to overtake those of lower priority or by providing guaranteed service levels on dedicated connections.

1.4 Routing

The network-on-chip (NoC) has been recognized as a paradigm to solve system-on-chip (SoC) design challenges. The routing algorithm is one of key research area of a NoC design. XY routing algorithm, which is a kind of distributed deterministic routing algorithms is a popular routing algorithm for standard mesh NoCs as it is simple to understand and implement although it may suffers from the problem of hotspot [5]. The main objective of XY routing algorithm is to distribute network load. A simple XY progressive routing algorithm consists of reducing an offset to zero before considering the offset in the next direction. This routing algorithm is also known as dimension order routing. This routing algorithm route packets by crossing dimensions in strictly increasing (or decreasing) order i.e. reducing to zero the offset in one dimension before routing in the next one.

For n-dimensional meshes, dimension order routing produces deadlock free routing algorithm [5, 6]. Although XY routing algorithms (Algorithm 1) assume that the packet header carries the absolute address of the destination node, the first few sentences in the algorithm computes the offset from the current node to the destination node. The offset is the value carried by the header when relative addressing is used. So, the remaining sentences in the algorithm describe the operation for routing using relative addressing. This type of addressing would also require updating the header at each intermediate node.

Algorithm1: XY Routing algorithm for 2-D Mesh NoC

Inputs: Coordinates of current node (Xcurrent, Ycurrent) and destination node (Xdest, Ydest)

Output: Selected output channel

```
Procedure XY :
    Xoffset := Xdest –Xcurrent;
    Yoffset := Ydest –Ycurrent;;

    If Xoffset<0 then
        Channel :=X-;
    endif

    if Xoffset>0 then
        Channel :=X+;
    endif

    if Xoffset=0 and Yoffset<0 then
        Channel := Y-;
    endif

    if Xoffset=0 and Yoffset>0 then
        Channel := Y+;
    endif

    if Xoffset=0 and Yoffset=0 then
        Channel := Internal;
    endif
```

3. TRAFFIC PATTERN

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

Permutation: The perfect k-shuffle permutation σ^k is defined by

$$\sigma^k(X) = (kX + [kX/N]) \bmod N$$

The perfect k-shuffle permutation performs a cyclic shifting of the digits in X to the left for one position. We have assumed k=2 in this work. The inverse perfect shuffle does the opposite of the perfect shuffle permutation. The bit reversal permutation ρ^k is defined by

$$\rho^k(X_{n-1}X_{n-2}.......X_1X_0) = ...X_0X_1......X_{n-2}X_{n-1}$$

This permutation is usually referred to as bit reversal. For the experimental results, We have used the values of K and N as 2 and 9 respectively.

Butterfly Permutation: The ith butterfly permutation interchanges the Zeroth and the ith digit of the index. In this paper we have assumed the butterfly permutations for k=2 and i = 0 ,1 and 2 with N = 9.

Baseline Permutation: The ith baseline permutation performs the cyclic shifting of the i+1 least significant digits in the index from left to the right for one position.

2. EXPERIMENTAL RESULTS

For analysis and comparison of the performance of the NoC on various Traffic Patterns, a discrete event, cycle accurate simulator NIRGAM [7] is used. Network on chip Interconnect Routing and Application Modelling (NIRGAM) is generic modular and extensible simulation framework providing substantial support to experiment with NoC designs in terms of routing Algorithms and applications on various topologies 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 [8]. For this paper we are calculating only the average overall Latency (in clk cycles per flit) of the network for various traffic patterns. We have calculated the overall average latency (in clk cycles per flit) for 3*3 2D-Mesh topology with XY Routing for various traffic patterns as exhibited in Figure 4, Figure 5 and Figure 6. The results presented in Figure 4 shows that *Bit reversal* has best overall average latency (in clk cycles per flit) of 3.66384 in comparison to perfect, inverse perfect and transpose traffic patterns. However when the average of the Latency of various permutations is compared to the latency of the transpose, we found that overall average latency of the various permutations is better than the transpose as shown in Figure 4.

The Figure 5 shows that the Butterfly Permutation for i=2 has the best overall average latency among various butterfly permutations and the transpose. We also found that Butterfly Permutation for i=0 has the poor latency due to longer XY paths resulting in poor performance in terms of latency. However when the average of the latencies of butterfly permutation for i =0 ,1, 2 is compared with the average latency of the Transpose, we can easily establish that butterfly permutation in general performs better in comparison to the transpose with the exception of zeroth permutation

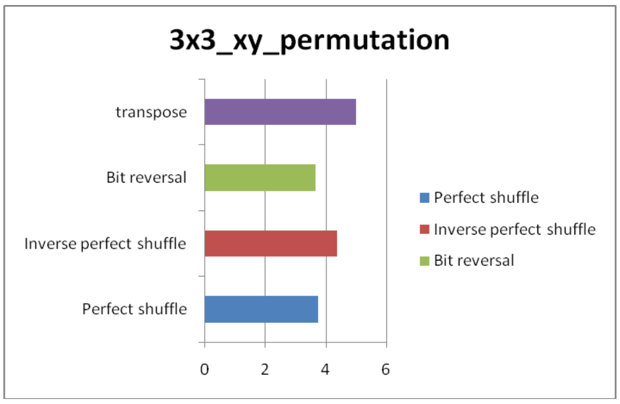

Fig 4: Comparison of overall average Latency of various permutations such as The perfect shuffle, The inverse perfect shuffle, The bit reversal and the transpose

.

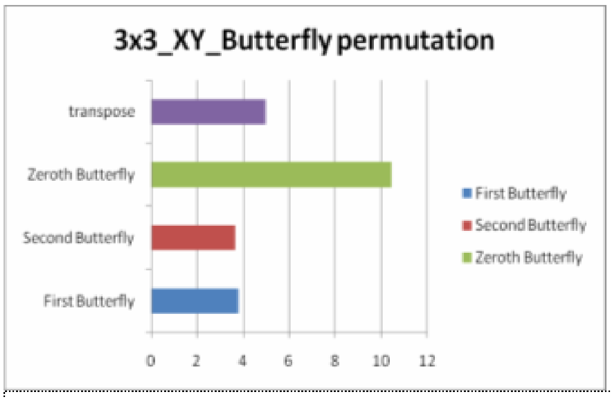

Fig 5: Comparison of overall average latency (in clk cycles per flit) of transpose permutation and butterfly permutation for k=2, and i = 0, 1, 2 with N=9

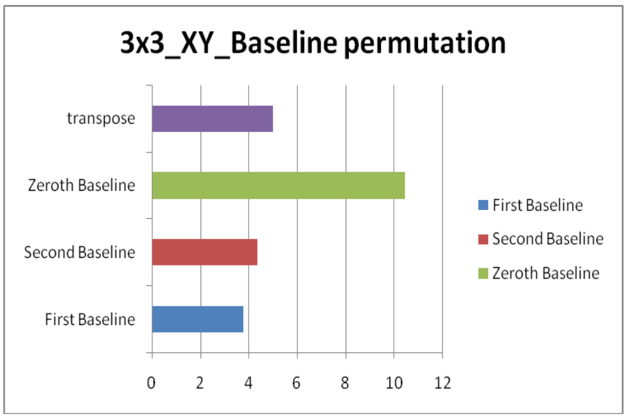

Fig 6: Comparison of overall average latency (in clk cycles per flit) of transpose permutation and baseline permutation for k=2, and i =0, 1, 2 with N=9

The Figure 6 presents the performance comparison of baseline permutations along with transpose permutation. The figure exhibits that baseline permutation for i=1 has the best overall average latency in comparison to the other baseline permutation and the transpose permutation. Moreover when the average of the latencies of baseline permutation for i =0 ,1, 2 is compared with the average latency of the Transpose, we can easily establish that baseline permutation in general performs better in comparison to the transpose with the exception of zeroth permutation.

As the results we have found that in Figure 4. Bit reversal has best Overall Average Latency(in clk cycles per flit) of 3.66384.However when the average of the Latency of various permutations is compared to the latency of the transpose , we found that Overall Average Latency of the various permutations is better than the transpose.

3. CONCLUSION

The paper presents the Network on Chip communication performance for various traffic patterns. The communication performance of the NoC is greatly affected by the traffic permutation and chosen routing, which in turn highlights that if appropriate traffic permutation is chosen according to the chosen routing than communication performance of the NoC can be greatly enhanced. The work presented can be of great help in choosing the appropriate application mapping to the cores of the NoC for improved performance.

4. REFERENCES

[1] Dally, W.J., Towles, B. 2001 Route Packets, Not Wires: On-Chip Interconnection Networks. In IEEE Proceedings of the 38th Design Automation Conference (DAC), 684–689

[2] Benini, L., DeMicheli, G. 2002 Networks on Chips: A New SoC Paradigm. In IEEE Computer Vol. 35, No. 1, 70–78

[3] Choudhary, N., Gaur, M.S., Laxmi, V. Irregular NoC Simulation Framework :IrNIRGAM

[4] Du, G., Zhang, D., Song, Y., Gao, M., Geng, L. 2008 Scalability Study on Mesh based Network on chip. In IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application

[5] J. Duato, S. Yalamanchili, L. Ni, Interconnection Networks: An Engineering Approach, Elsevier, 2003

[6] Bjerregaard, T., Mahadevan, S. 2006 A Survey of research and practices of network-on-chip. In Acm computing Surveys, vol.38, No.11-51.

[7] Jain, Lavina, Al-Hashimi, B.M, Gaur, M.S, Laxmi V and Narayanan, A, "NIRGAM: A Simulator for NoC Interconnect Routing and Application Modelling, Proc. DATE 2007, 2007

[8] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.