Full System Simulation: Network On Chip

Deepak Siddharth Parthipan, *Student Member*

***Abstract*—Design of multicore chip has evolved into complex task, integrating different cores on chip, hardware threads, memories, and network on chip. There is always a need for measuring the performance of newer design so that the development phases get rampant. Network on chip scalable communication centric-interconnect are becoming famous because of the emerging trend in SoC designs. The NoC provides feature such as parallelism and scalability which helps in increasing the throughput of the integrated system. Improving the network performance is essential to increase the number of processing cores on a single chip. In this paper, we present the Multi2Sim simulation framework, which models simple 3x3 mesh and torus architecture and is intended impact on network traffic performance when its running on a SoC.**

***Index Terms*—NoC, multi2sim, mesh, torus.**

I. Introduction

Multicore environment approach to build systems on a computer architecture provides additional benefits in terms of the performance. The use of additional core helps to reduce the clock speed in comparison to that of multicore environment. The multicore design has mainly three components microprocessor, cache hierarchy and interconnection networks. The demand for the complexity of network architecture is growing and the interest has developed globally. Interconnection networks is responsible for communicating information between main memory or another processor. Caches from different processors share memory blocks, the interconnect oversees transmitting coherence messages generated by the cache controllers. Research options in this filed includes increasing network performance by focusing on new topologies, switching and flow control mechanisms, routing algorithms or fault tolerance techniques.

Different network on chip topologies has been proposed such as Mesh, Torus, SPIN, Star, Octagon, Folded Torus [1]. This paper uses a simulation approach to study the performance on 3x3 mesh and 3x3 torus architecture multicore network on chip architectures using multi2Sim. Section 2 describes the network on chip architecture with processor core, cache, switch and main memory. Section 3 talks about the multi2sim software methodology. Section 4 provides the evaluations and the simulation results. Section 5. Finally, conclusions and directions for future research.

II. Architecture

A. Core-Cache Design

The simulated Multi-Core Architecture with Internal Network as described below. Intel x86 processing core was selected for the simulation aspect. Fig. 1 The x86 processor consists Internal Network that has a single private L1 cache per core, to unify the instruction and data requests. Also, single L2 cache for each of the cores. Connectivity of L1-to-L2 network consists of switches and network nodes, which further communicate with main memory through others switches. One of the most challenging aspect of the cache design is that for example, a L2 cache must ensure data consistency among L1 cache. MOESI is the most widely used cache coherence protocol. Multi2sim also emulates this in the software. The x86 in the multi2sim has 6 stage pipeline with configurable latencies.

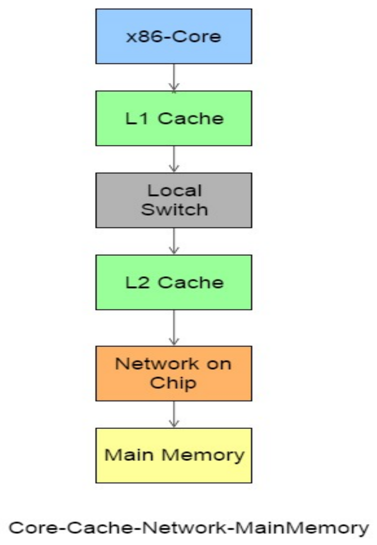

Figure 1. Multicore Architecture

Network-on-chip (NoC) is a packet switched on-chip communication network designed using a layered methodology. NoCs use packets to route data from the source to the destination PE via a network fabric. NoCs are an attempt to scale down the concepts of largescale networks, and apply them to the embedded system-on-chip (SoC) domain. Mesh and torus network architectures are the most widely used on chip topologies. So, the performance of 3x3 mesh and 3x3 torus topology is analyzed. If no routes are specified in the network configuration file, the simulator uses the Floyd-Warshall algorithm to initialize the routing table based on the existing links between the nodes. The routing table contains the shortest paths for every pair of nodes in the network.

B. Mesh Topology

Mesh Topology is the most popular networks which is used in current parallel supercomputer architecture. The 2D mesh network topology is a 2D grid of nodes where each node is

Deepak Siddharth is with the Department of Electrical Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA (e-mail: dp9040@rit.edu).

connected to its own switch. The latency to send a message from one node to another node is non-uniform. Traveling from one node to an adjacent node travels over one link while going from a node from one corner to a node in the opposite corner travels over 2n-2 links. The latency on average is O(sqrt(N)). Since there is one switch per node, the cost is O(N). This is better than the crossbar network topology but the obvious tradeoff is higher latency. Other characteristics are that the layout is very intuitive. It is easy to lay out on a chip. The topology is considered direct since nodes are connected to switches that are within the network. There are also multiple ways for messages to travel from one node to the other. However, the mesh topology is considered blocking. Consider a node in the corner of a 2D mesh trying to send 3 messages to 3 different locations. There are only two paths connected to the switch for this node so at least one message is blocked. Fig. 2 shows the complete block view of 8 core 3x3 mesh network with core, cache and memory.

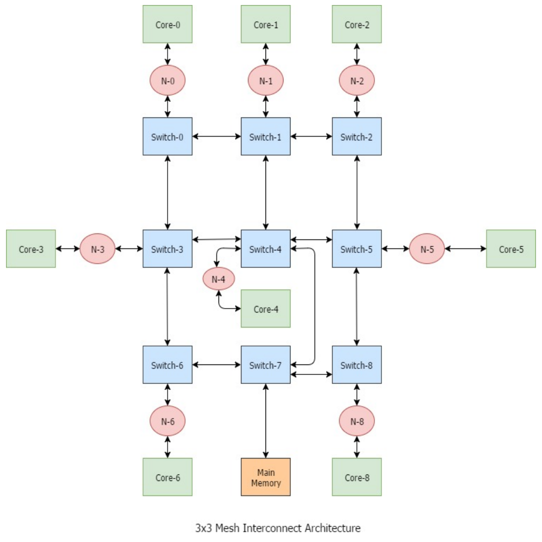

Figure 2. 3x3 Mesh Full Network Topology

C. Torus Topology

Torus networks are frequently utilized on top-performing supercomputers. They have certain drawbacks, but their main benefit is cost and that makes them appropriate for very big installations. Besides, in some computational algorithms compute nodes mainly exchange data with their closest neighbors. The 2D torus is an extension of the 2D mesh. A problem with 2D meshes is that nodes near the edge of the mesh have different characteristics than nodes at the center. The 2D torus avoids this problem by having a link between a switch on an edge and the switch on the opposite edge. Like a 3D mesh, the 2D torus has slightly lower latency while higher cost and complexity than a 2D mesh. Torus and folded torus network architectures are the most commonly used for network on chip topologies. Folded torus has similar layout as torus, in which the links are arranged in a folded manner to equalize the wire length. This eliminates wrap around links which are usually found in torus topology. Fig. 3 shows the complete block view of 8 core 3x3 torus network with core, cache and memory.

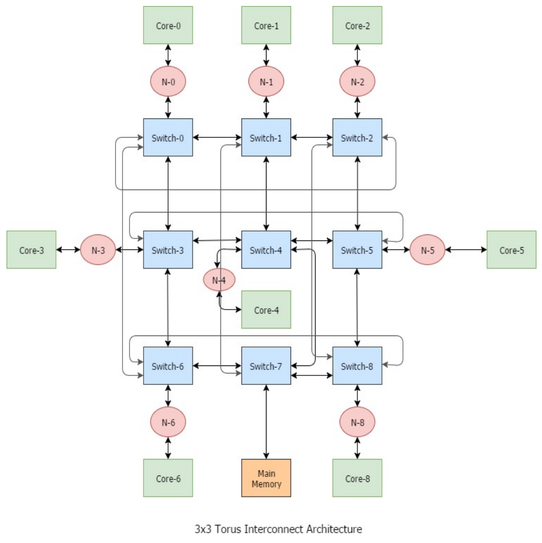

Figure 3. 3x3 Torus Full Network Topology

III. Methodology

Multi2sim[2] provides a Simulation framework for heterogeneous CPU-GPU systems. It supports CPU architectures such as: x86, ARM, MIPS. Full system simulation consists of guest application program running on a simulated core environment. There is a slight difference between full system simulation and application level simulation is that the application simulation does not emulate OS on system but uses system call interfaces. There are four isolated software modules (simulator, disassembler, functional simulator, and stand-alone simulator) for each x86 architecture and each module has a command-line interface for stand-alone execution, or an API for interaction with other modules.

Multithreading The major problem in using multithreading is to acquire great performance through the multi-core processor. The application have capability to run subroutine on various cores. This signifies that the data dependence ought to be handled in a very synchronized and structured way. The simulated program runs test threads program on all the 8 cores.

The configuration file is a plain-text file in the IniFile format. The user Init files is updated along with the m2s command line options to compile and run the program. The memory system is formed of a set of cache modules, main memory modules, and interconnects. '--x86-config': The parameters of the CPU model used for a detailed simulation. This file defines the system frequency, cores, and the number of threads the systems is running. '--ctx-config': This file contains the test execution details of that program. It also redirects the stdout to the different file location. '--mem-config': Is used to configure the memory system. It basically sets the Processor, L1 Cache, Switch1, L2 Cache, Switch2 and Main Memory

IV. Evaluations and Result

Different results of the simulations are compared in this section. The simulated 3x3 mesh and 3x3 torus network are given in the fig.5 and fig.6 respectively. All the nodes in the network except for core except for one core main memory.

Both the network was simulated standalone for 1000 cycle, all 8 nodes inject with a message of size 72 in the network.

Throughput defines the rate at which the output is being produced. The throughput can be defined as follows:

$$Throughput = \frac{TotalCompletedMessage * LengthofMessage}{NumberofblocksofIP * TotalTime}$$

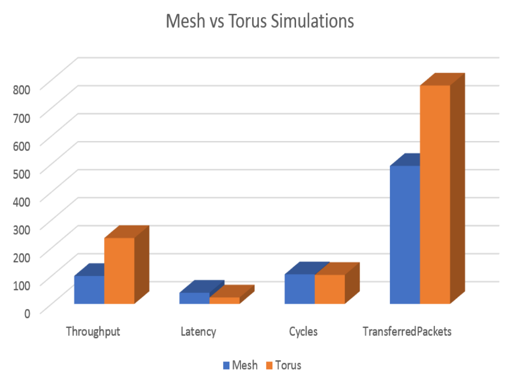

Figure 4. Simulation Results

The above equation, the number of bytes transferred and called as total completed message. The number of function IP blocks which in involved in the communication is known as number of block of IP. Length of message is Average Message Size typically it's the flits. Time is taken is the average latency. Latency is the time that elapsed between the packets being reached to the destination on the defined NoC architecture. For the sake of display purpose the charts were scaled equally for throughput and cycles values and shown as results in fig.4.

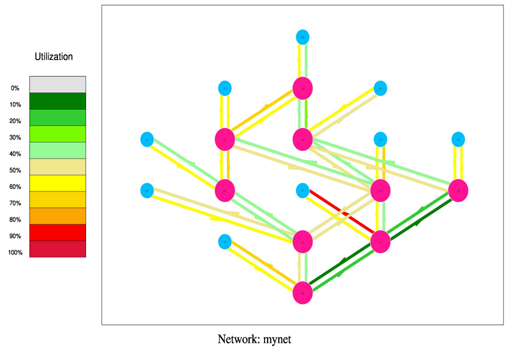

Figure 5. Network output graph-Mesh

V. Conclusions

Network on chips has lots has lots of different issues associated with it. Mesh and torus are the widely-used network topologies. Using multi2sim core-memory-network environment comparisons of mesh and torus was done for latency, throughput, packet received. Torus has better latency and through put. It simply because of the way the structure is designed by connecting the end nodes across all rows and columns. The fall side to this higher power consumption however the third design which was simulated but done explained folded torus has lesser power consumption than torus.

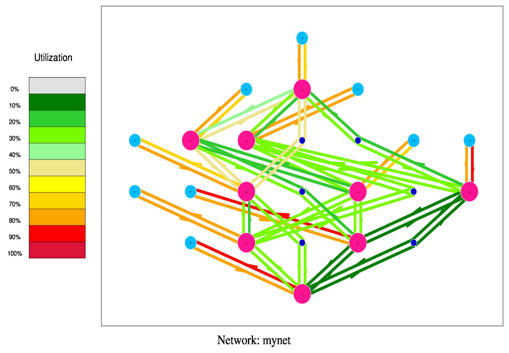

Figure 6. Network output graph-Torus

Basically, the routing protocol, number of nodes and links in a network determine the total power consumed. The conclusion is that torus architecture is better than a mesh topology. For future work, the cores can be multiplied to higher number like 16, 32 or 64 for further study. More exploitation can be done towards understanding the dead lock situation and virtual buffer within the multi2sim tool environment. Also, automation of the scripts can be future scope of work especially when we need intelligent environment to setup different models for study and comparison.

VI. References

[1] Performance Evaluation and Design Trade-Offsfor Network-on-Chip Interconnect Architectures, Partha Pratim Pande, CristianGrecu, MichaelJones, Andre´Ivanov and Resve Saleh.

[2] Multi2sim guidebook

ACKNOWLEDGEMENT

This research paper work has been done as part coursework CMPE731- Design and Test of Multicore chips. This project was carried out under the guidance and support of Professor Amlan Ganguly.

Deepak Siddharth received the BE degree in Electronics and Communication from Karunya University, India in 2010. He then completed PG Diploma in Embedded System Design from National Institute of Electronics & Information Technology, India in 2011. From 2011 to 2016 he worked as Design Engineer for BPL Telecom Pvt Ltd and Embedded Software Developer for Robert Bosch, India. He is currently working as a graduate student in the department of Electrical Engineering at Rochester Institute of Technology, New York. His Research interests include Computer Architecture, Digital Design and Embedded systems.