Design and Optimization of System-on-chip (SOC)

[1] D.Naresh Kumar, [2] Geethu Mohan
[1,2]Assistant Professor, ECE Department,
MLRIT, Dundigal, Hyderabad, Telangana

Abstract*:* The design complexity in consumer embedded systems has changed to new techniques. The System –on-chip consists of many integrated components of high speed data rates. Intercommunication requirements of SOC uses shared bus or a hierarchy of buses because of their poor scalability, System size, energy efficiency requirements and their shared bandwidth between all the attached hardware cores of final products. To overcome the problems of scalability and complexity we proposed the Networks-On-Chip (NoCs).TheSoCs, overheads of buses are connected by general-purpose communication architectures. This paper describes that NoCs as a major role in optimizing power consumption, the performance The performance of can be Enhanced by using data compression technique.

Keywords: System on chip, System on chip, Network on chip, Network interface, Network Interface compression, Electromagnetic interference, Non Uniform Cache Architecture.

I. INTRODUCTION

In the last years there has been an increase in computation requirements for embedded systems due to the increasing complexity of new communication and multimedia standards. This has fostered the development of high-performance embedded platforms that can handle the computational requirements of recent complex algorithms, which cannot be executed in traditional embedded mono-processor architectures. To meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. Although SoCs promise to significantly improve the processing capabilities and versatility of embedded systems, one major problem in their current and future design is the effectiveness of the interconnection mechanisms between the internal components, as the amount of components grows with each new technological node. Bus-based designs are not able to cope with the heterogeneous and demanding communication requirements of SoCs. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics.

However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered. Such scalable bandwidth requirement can be satisfied by using on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC) architecture. The scalable and modular nature of NoCs and their support for efficient on-chip communication lead to NoC-based system implementations. Even though the current network technologies are well developed and their supporting features are excellent, their complicated configurations and implementation complexity make it hard to be adopted as an on-chip interconnection methodology.

Hence, new paradigms and methodologies that can design power-effective and reliable interconnect for SoCs are a must nowadays. Networks-on-Chip (NoCs) have been suggested as a promising solution to the aforementioned scalability problem of forthcoming SoCs. NoCs also help in tackling design complexity and verification issues. Using NoCs the interconnect structure and wiring complexity can be controlled well. When the interconnect is structured, the number of timing violations that occur during the physical design (floor planning and wire routing) phase are minimal. Such design predictability is critical for today's MPSoCs to achieve timing closure. It leads to faster design cycle and faster time-to-market.

II. SYSTEM-ON- CHIP

Multi-processor System–on–Chip (MPSoC) are SoC that may contain one or more types of computing subsystems, memories, input/output devices (I/O), and other peripherals, as in [1]. The MPSoC architecture is made of three types of components: software subsystems, hardware subsystems, and inter-subsystem communication .The hardware subsystems (HW-SS) represent custom hardware subsystems that implement specific functionality of an application or global memory sub systems. The HW-SS contain two types of components: intra-subsystem communication and specific hardware components. The hardware components implement specific functions of the target application or represent global memories accessible by the computing subsystems. The intra-subsystem communication represents the communication inside the HW-SS between the different hardware components. This can be in form of a small bus (collection of parallel wires for transmitting address, data, and control signals) or point-to-point communication links.

The software subsystems (SW-SS) represent programmable subsystems, also called processor nodes of the architecture. The SW-SS include computing resources, intra-subsystem communication, and other hardware components, such as local memories, I/O components, or hardware accelerators. The computing resources represent the processing units or CPUs. The CPU (central processing

unit) also known asprocessor core, processing element, or shortly processor executes programs stored in the memory by fetching their instructions, examining them, and then executing them one after another as in [1],[5]. There are two types of SW-SS: single core and multi-core. The single-core SW-SS includes a single processor, while the multi core SW-SS can integrate several processor cores in the same subsystem, usually of same type. The intra-subsystem communication represents the communication inside the SW-SS, e.g., local bus, hardware FIFO, point-to-point communication links, or other local interconnection network used to interconnect the different hardware components inside the SW-SS. Homogeneous MPSoC as in [8] architectures are made of identical software subsystems incorporating the same type of processors. In the heterogeneous MPSoC architectures, different types of processors are integrated on the same chip, resulting in
different types of software subsystems. These can be GPP (general-purpose processor) subsystems for control operations of the application; DSP (digital signal processor) subsystems specially tailored for data-intensive applications such as signal processing applications; or ASIP (application-specific instruction set processor)subsystems with a configurable instruction set to fit specific functions of the application.

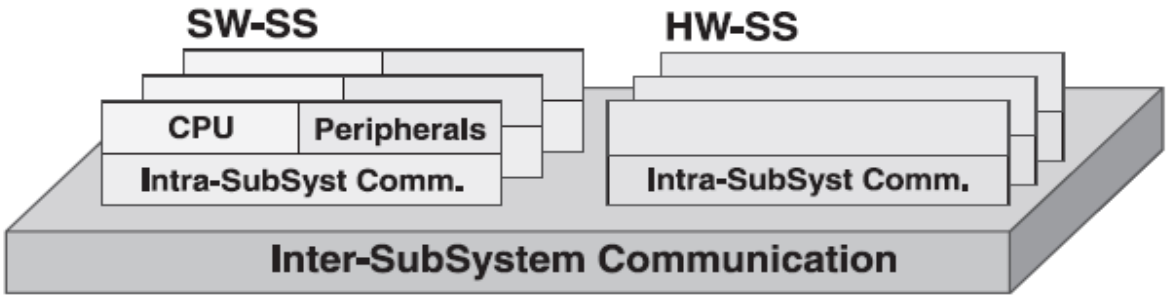

Figure 1: MPSoC

The different subsystems working in parallel on different parts of the same application must communicate each other to exchange information. There are two distinct MPSoC designs that have been proposed and implemented for the communication models between the subsystems: shared memory and message passing. The shared memory communication model characterizes the homogeneous MPSoC architecture. The key property of this class is that communication occurs implicitly. The communication between the different CPUs is made through a global shared memory. Any CPU can read or write a word of memory by just executing LOAD and STORE instructions. Besides the common memory, each processor code may have some local memory which can be used for program code and those items that need not be shared. In this case, the MPSoC architecture executes a multithreaded application organized as a single software stack. The message-passing organization assumes multiple software stacks running on identical or non-identicalsoftware subsystems. The communication between different subsystems is generally made through message passing. The key property of this class is that the communication between the different processors is explicit through I/O operations. The CPUs communicate by sending each other message by using primitives such as send and receive.

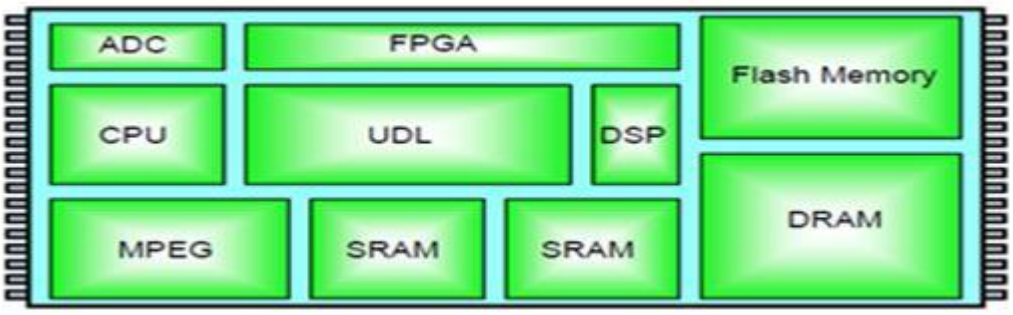

Figure 2: System on chip

Multiprocessor architectures and platforms have been introduced to extend the applicability of Moore's law. They depend on concurrency and synchronization in both software and hardware to enhance the design productivity and system performance. These platforms will also have to incorporate highly scalable, reusable, predictable, cost and energy-efficient architectures. With the rapidly approaching billion transistors era some of the main problems will arise from non-scalable wire delays, errors in signal integrity and unsynchronized communications.

The MPSoC architecture is made of three types of components: software subsystems, hardware subsystems, and inter-subsystem communication. The hardware subsystems (HW-SS) represent custom hardware subsystems that implement specific functionality of an application or global memory subsystems. As in figure no.1 the HW-SS contain two types of components: intra subsystem communication and specific hardware components. The hardware components implement specific functions of the target application or represent global memories accessible by the computing subsystems.

A. Traditional SoC nightmare

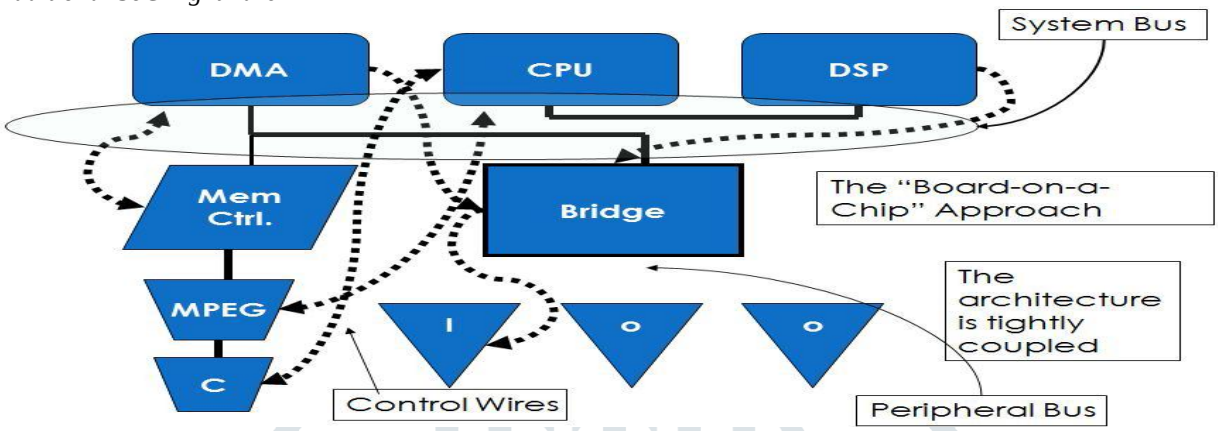

Figure 3: Traditional SoC nightmare

As seen in figure no.3 the point to point communication results in communication bottleneck. Bus based communication is also seen in the figure .Both the types of communication between the bocks results in communication bottleneck. As a result communication and computation cannot be separated. Thus the performance, speed and power consumption are adversely affected.

B. Limitations of SoC

1) Needs variety of interfaces
2) Poor separation between computation and communication
3) Design complexity

III. NETWORK-ON-CHIP

On a billion transistors chip, it may not be possible to send a global signal across the chip within real-time bounds. As the number of IP modules in Systems-on-Chip (SoCs) increases, bus-based interconnection architectures may prevent these systems to meet the performance required by many applications. For systems with intensive parallel communication requirements buses may not provide the required bandwidth, latency, and power consumption. A solution for such a communication bottleneck is the use of an embedded switching network, called Network-on-Chip (NoC), to interconnect the IP modules in SoCs. NoCs design space is considerably larger when compared to a bus-based solution, as different routing and arbitration strategies can be implemented as well as different organizations of the communication infrastructure. If the SoC (System-on-Chip) is synchronized by a global clock signal, the circuit will be more prone to EMI (electromagnetic interference). The traditional system designs are usually based on critical paths and clock trees. These critical paths and clock trees contribute to an increased amount of power consumption. Therefore, SoCs are not power efficient. Besides, it is difficult to manage these clock trees due to clock skew problems.

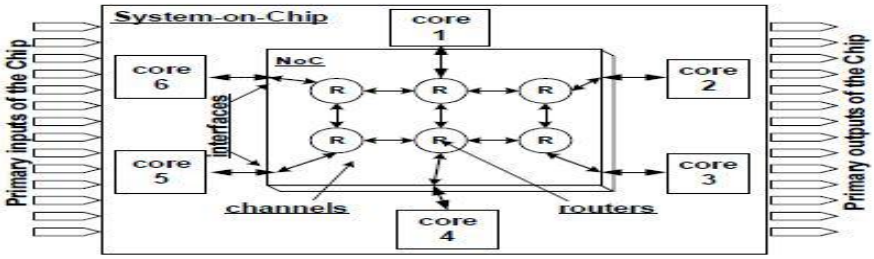

Figure 4: NoC Design space of Network on chip

Researchers have combined the ideas of synchronous and asynchronous designs. One such strategy is GALS (globally asynchronous and locally synchronous) solution. GALS divides a system into smaller, locally decoupled synchronous regions and then composes a few of them to yield a localized subsystem. These synchronous regions and subsystems would be easier to integrate into a global solution and verify. There will be an asynchronous way in which all the local synchronous regions will communicate at the system

level. Therefore, these different synchronous regions need not have to be synchronized to a single global clock. This approach will reduce the requirement for chip-wide clock trees; the designers could focus on local synchronous regions only, which would be far less complex than the complete system. Since one has the flexibility to reduce the clock speed of a given synchronous region (or node) independent of other such regions, the amount of power consumption in a system can be managed better and reduced. One GALS solution is NOC (Network-on-Chip). NOC can improve design productivity by supporting modularity and reuse of complex cores.

A. *Basic properties of the NoC paradigm*

The basic properties of the NoC paradigm are listed below
1) Separates communication from computation.
2) Avoids global, centralized controller for communication.
3) Allows arbitrary number of terminals.
4) Topology allows the addition of links as the system size grows (offers scalability).
5) Customization (link width, buffer sizes, even topology).
6) Allow multiple voltage and frequency domains.
7) Delivers data in-order either naturally or via layered protocols.
8) To support system testing.
Thus, the NoC plays a critical role in optimizing the performance and power consumption of such non-uniform cache based multicore architectures.

B. Difference between bus based and NOC Architectures

TABLE I
DIFFERENCE BETWEEN BUS BASED AND NETWORK ON CHIP ARCHITECTURES

Bus Based	Network on chip
Longer connections → higher parasitic capacitance	Performance does not downgrade with network scaling
Arbitration grows and becomes a bottleneck	Arbitration and routing are distributed
Bandwidth is limited and shared by all cores	Aggregated bandwidth scales with network size
Latency is wire-speed once arbitration granted control	Multiple hops increase latency

C. Basic building blocks of Network on chip
A network-on-chip is composed of three main building blocks.
1) Links
2) Router
3) Network interface or Network adapter

1) Links: A communication link is composed of a set of wires and connects two routers in the network. Links may Consist of one or more logical or physical channels and each channel is composed of a set of wires. In the remaining chapters, unless stated otherwise, the words net, wire, and line mean a single wire interconnecting two entities (routers and/or IP cores). The words channel and link mean a group of wires connecting two entities. Typically, a NoC link has two physical channels making a full-duplex connection between the routers (two unidirectional channels in opposite directions).The number of wires per channel is uniform throughout the network and is known as the channel bit width. The implementation of a link includes the definition of the synchronization protocol between source and target nodes. This protocol can be implemented by dedicated wires set during the communication or through other approaches such as FIFOs. Asynchronous links are also an interesting option to implement globally asynchronous locally synchronous (GALS) systems where local handshake protocols are assumed. The links ultimately define the raw performance (due to link delays) and power consumption in an NoC and designers are supposed to provide fast, reliable, and low-power interconnects between nodes in the network.

2) Routers: The design and implementation of a router requires the definition of a set of policies to deal with packet collision, the routing itself, and so on. A NoC router is composed of number of input ports (connected to shared NoC channels), number of output ports (connected to possibly other shared channels), switching matrix connecting the input ports to the output ports, and local port to access the IP core connected to this router. In addition to this physical connection infrastructure, the router also contains a logic block that implements the flow control policies (routing, arbiter, etc.) and defines the overall strategy for moving data though the NoC. A flow control policy characterizes the packet movement along the NoC and as such it involves both global (NoC-level) and local (router-level) issues. One can ensure a deadlock-free routing, for instance, by taking specific measures in the flow control policy (by avoiding certain paths within the NoC for example). Control can be of two types -centralized and distributed. NoCs typically use a distributed control, where each router makes decisions locally. Deadlock occurs when network resources are fully occupied and waiting for each other to be released to proceed with the communication, that is, when two paths are blocked in a cyclic fashion when the status of the resources keep changing (there is no deadlock) but the communication is not

completed. Routing algorithm is the logic that selects one output port to forward a packet that arrives at the router input. Routing algorithms can lead to or avoid the occurrence of deadlocks and live locks. While the routing algorithm selects an output port for a packet, the arbitration logic implemented in the router selects one input port when multiple packets arrive at the router simultaneously requesting the same output port.

3) Network Interface: The third NoC building block is the network adapter (NA) or network interface (NI). This block makes the logic connection between the IP cores and the network, since each IP may have a distinct interface protocol with respect to the network. This block is important because it allows the separation between computation and communication. This allows the reuse of both, core and communication infrastructure independent of each other. The adapter can be divided into two parts: front end and back end. The front end handles the core requests and is ideally unaware of the NoC. The back end part handles the network protocol (assembles and disassembles the packet, reorders buffers, implements synchronization protocols, helps the router in terms of storage, etc.).

D. NoC Performance Parameters

The performance of a network-on-chip can be evaluated by three parameters:

1) Bandwidth
2) Throughput
3) Latency

1) *Bandwidth:* Bandwidth refers to the maximum rate of data propagation once a message is in the network. The unit of measure for bandwidth is bit per second (bps) and it usually considers the whole packet, including the bits of the header, payload and tail.

2) *Throughput:* Throughput is defined as the maximum traffic accepted by the network, that is, the maximum amount of information delivered per time unit. The throughput measure is messages per second or messages per clock cycle. One can have a normalized throughput (independently from the size of the messages and of the network) by dividing it by the size of the messages and by the size of the network. As a result, the unit of the normalized throughput is bits per node per clock cycle (or per second).

3) *Latency:* Latency is the time elapsed between the beginning of the transmission of a message (or packet) and its
Complete reception at the target node. Latency is measured in time units and mostly used as comparison basis among different design choices. In this case, latency can also be expressed in terms of simulator clock cycles.
Normally, the latency of a single packet is not meaningful and one uses the average latency to evaluate the network
Performance.

E. Data compression technique for NoC

The trend towards integrating multiple cores on the same die has accentuated the need for larger on-chip caches. Large caches are constructed as a multitude of smaller cache banks interconnected through a packet-based Network-on-Chip (NoC) communication fabric. NoC plays a critical role in optimizing the performance and power consumption of non-uniform cache-based multicore architectures. We examine the data compression technique Compression in the NIC (NC).Higher frequencies lead to higher power consumption, which, in turn, spawned cooling and reliability issues. By utilizing a number of simpler, narrower cores on a single die, architects can now provide Thread-Level Parallelism (TLP) at much lower frequencies. The presence of several processing units on the same die necessitates oversized L2 and, where applicable, L3 caches to accommodate the needs of all cores. Larger cache sizes are easily facilitated by the aforementioned explosion in on-chip transistor counts. However, the implementation of such large cache memories could be impeded by excessive interconnect delays. While smaller technology nodes are associated with shorter gate delays, the former are also responsible for increasing global interconnect delays [1]. Therefore, the traditional assumption that each level in the memory hierarchy has a single, uniform access time is no longer valid. Cache access times are transformed into variable latencies based on the distance traversed along the chip, and has spurred the Non-Uniform Cache Architecture (NUCA) concept as in [10]. A large, monolithic L2 cache is divided into multiple independent banks, which are interconnected through an on-chip interconnection network. The network fabric undertakes the fundamental task of providing seamless communication between the processing cores and the scattered cache banks. NoCs are be-coming increasingly popular because of their well-controlled and highly predictable electrical properties and their scalability. However, research in NoC architectures clearly points to some alarming trends; the chip area and power budgets in distributed, communication-centric systems are progressively being dominated by the interconnection network . The ramifications of this NoC dominant design space have led to the development of sophisticated router architectures with performance enhancements, area-constrained methodologies, power-efficient and thermal-aware designs, and fault-tolerant mechanisms. Data compression as in [5], [6] is used for additional gain in performance and power.

Advantages of Data compression are

1) Reduce network load
2) Lower average network latency
3) Lower power consumption
4) Application level performance improvement.

1) Network Interface Compression: As in figure no.5 this scheme needs a compressor and de compressor in each NIC, and thus, adds to the area and timing overheads. Some of the compression and decompression latency can be hidden by utilizing a pipelined router design. Frequent-pattern compression can be realized in five cycles.

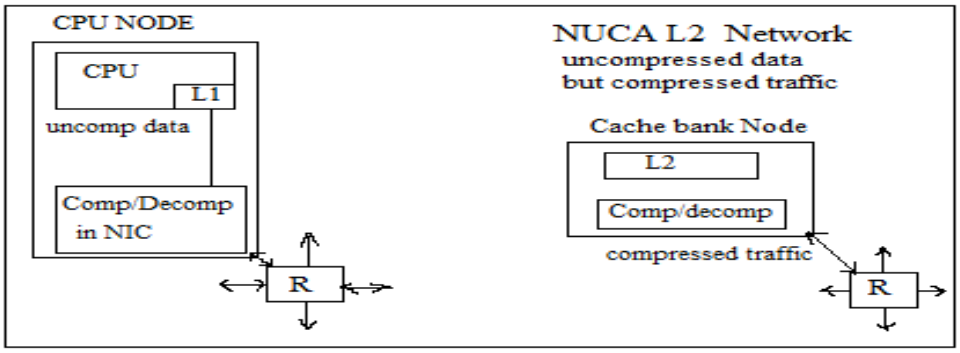

Figure 5: NIC Compression

1) In the first cycle, the length of each word is decoded using the prefix tags.
In the next two cycles, the starting bit address of each word is computed by cumulatively adding the length bits of each word.
2) In the fourth and the fifth cycles, a parallel pattern decoder decodes the contents of the compressed data bits into
Uncompressed words using their respective prefix bits.The decompression pipeline can be designed such that we need not wait for the entire cache line to arrive before we can start the decompression phase. Since at the ejection port we can get at most one flit per cycle in a wormhole switched network, decompression may commence as soon as the header flit of the compressed cache line arrives. This flit contains the prefix tags and all the necessary information for the first three stages of the decompression pipeline. After the tail flit arrives at a NIC, we can complete the final two steps of the decompression stage and commit the data to the L1 cache and/or processor. This technique will save three clock cycles per decompression operation, effectively reducing the overhead from 5 cycles to 2 cycles. The three cycle saving can be accrued from the proposed scheme.

Advantages of NIC Compression
1) The NIC scheme on an average provides network latency reduction of 20%, (32% maximum),
2) Power consumption reduces on an average by 10% (21% maximum).

IV. CONCLUSIONS

Intercommunication requirements of MPSoCs made of hundreds of cores will not be feasible using a single shared bus or a hierarchy of buses due to their poor scalability with system size, their shared bandwidth between all the attached cores and the energy efficiency. NoC separates communication from computation, centralized controller for communication, allows arbitrary number of terminals. It has a topology that allows the addition of links as the system size grows (offers scalability), does not utilize long, global wires spanning the whole chip .As compared to bus based communication NoC provides many advantages like performance does not downgrade with network scaling , arbitration and routing are distributed, multiple hops increase latency etc. Data compression is employed to reduce network load, to lower the power consumption. The NIC scheme on an average provides reduction in power consumption.

REFERENCES

[1] System- on -chip basics Springer-“Embedded Software Design and Programming of Multiprocessor System-on-Chip”, Embedded Systems, Springer Science+ Business Media.
[2] Chapter 2-Network–on–Chip basics “Reliability, Availability and Serviceability of Networks-on-Chip”, © Springer Science Business Media.
[3] Sonal S. Bhople M. A. Gaikwad, *A Comparative Study of Different Topologies for Network-On-Chip Architecture,* International Journal of Computer Applications “Recent Trends in Engineering Technology-2013”.
[4] Erik Fischer and Gerhard P. Fettweis*An Accurate and Scalable Analytic Model for Round-Robin Arbitration in Network-on-Chip* Vodafone Chair Mobile Communications Systems TechnischeUniversit¨at Dresden, Dresden, Germany ©2013 IEEE.
[5] Comparative Analysis of Different Topologies based on Network-on-Chip Architectures International Journal of Electronics and Communication Engineering.ISSN 0974-2166 Volume 6, Number 1(2013).
[6] Performance Optimization for Networks-on-Chip Architectures using Multi-Level Network Partitioning EnCon 2012, 5th Engineering Conference, "Engineering Towards Change - Empowering Green Solutions"10-12th July 2012, Kuching, Sarawak.
[7] S. Borkar, “Thousand core chips - a technology perspective,” in Proc.of DAC, 2007.
[8] W. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in Proc. of DAC, 2001.
[9] L. Benini and G. De Micheli, “Networks on chips: a new socparadigm,”*Computer*, vol. 35, no. 1, Jan 2002.
[10] U. Ogras, P. Bogdan, and R. Marculescu, “An analytical approach for network-on-chip performance analysis,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 29, no. 12, pp. 2001 –2013, Dec. 2010.
[11] P. Pande, C. Grecu, A. Ivanov, R. Saleh, and G. De Micheli, “Design,synthesis, and test of networks on chip,” *IEEE Des. Test Compute*, vol. 22,no. 5, pp. 404–413, Sep./Oct. 2005.