

QoS architecture and design process for cost effective Network on Chip

Evgeny Bolotin, Israel Cidon, Ran Ginosar and Avinoam Kolodny

Electrical Engineering Department,
Technion—Israel Institute of Technology
Haifa 32000, Israel

Abstract—Our design process characterizes and verifies the inter-module traffic, places the modules so as to minimize the system spatial traffic density on a generic network grid, and optimizes the grid by trimming links, routers and buffers while maintaining the required QoS. We classify the system traffic into four classes of service: Signaling (replacing inter-module control signals); Real-Time (representing delay constrained bit streams); RD/WR (modeling short data access) and Block-Transfer (providing for large data bursts). We model traffic behavior for each class and define the Quality of Service requirements of each class in terms of delay, throughput and relative priority. Based on the traffic model we derive a NoC architecture comprising network protocols, routers, buffers and links that support these four classes. This generic architecture is subsequently optimized to minimize cost (area and power) while maintaining the required QoS.

The network architecture is based on the following principles: The network topology is a planar grid of switches that route the traffic according to fixed shortest path (X-Y based) discipline, thus minimizing hardware tables and traffic overheads. Buffer requirements are reduced by employing multi-class wormhole forwarding while allowing inter-class priorities. The layout of the network is customized and bandwidth is allocated to links according to their relative load so that the utilization of links in the network is balanced. During customization unnecessary resources (links, routers, buffers) are trimmed where possible, resulting in a low cost customized layout for the specific SoC. Analytic calculations and traffic simulations are used in the optimization steps to ensure that QoS is strictly met.

Index Terms— Network on Chip, QoS architecture, wormhole switching, NoC design process

I. INTRODUCTION

On-chip packet-switched networks [1]-[11] have been proposed as a solution for the problem of global interconnect in deep sub-micron VLSI Systems-on-Chip (SoC). Networks on Chip (NoC) can address and contain major physical issues such as synchronization, noise, error-correction and speed optimization. NoC can also improve design productivity by supporting modularity and reuse of complex cores, thus enabling a higher level of abstraction in architectural modeling of future systems [4], [5]. However, VLSI designers must be ensured that the benefits of NoC do not compromise system performance and cost [8], [10]. Performance concerns are associated with latency and throughput. Cost concerns are primarily chip-area and power dissipation. This paper presents a design process and a network architecture that satisfy Quality of Service (performance) requirements at a measurable cost which is favorably compared with alternative on-chip interconnection approaches.

Traditionally, on-chip global communication has been addressed by shared-bus structures and ad-hoc direct interconnections. Non-scalability of these approaches was discussed in [1], [6], [9]. However, modern on-chip buses have evolved to multi-layered and segmented structures, supporting split transactions, burst transfers and parallel operations [12]-[14]. From several aspects they can be considered as networks but still, they don't provide effective spatial reuse of resources and do not utilize packet or wormhole switching associated with distributed routing and congestion/flow control. Therefore, they are inefficient and require centralized arbitration mechanisms.

Advantages of spatial-reuse packet/wormhole switched networks were analyzed in comparison with buses by several authors [1], [3], [5], [8], [9]. A hybrid approach, supporting both NoC and on-chip buses has been proposed in [10]. Switched networks and techniques for their design have been developed for computer networks and for multiprocessor systems [15]-[21]. However, a unique set of resource constraints and design considerations exists for an on-chip environment. As described in [1], [9], memory and computing resources are relatively more expensive on-chip, while relatively more wires are available. The need to combine several types of service, such as "best effort" and "guaranteed throughput" was noted by [1], [8]. In [9] it was suggested to support

several access paradigms such as request-response (for compatibility with bus-based approaches) and connection-oriented services for extended functionality. A mesh network topology was proposed in [4], a torus topology was proposed in [1], while [6] used a fat tree. Different routing schemes and router architectures have been proposed [1], [4], [6], [7], [10], [11].

Unlike computer networks which are built for on-going expansion, future growth and standards compatibility, on-chip networks can be designed and customized for an a-priori known set of computing resources, given pre-characterized traffic patterns among them. These imply that various components of the network architecture including addressing fields and QoS classification can be modified between implementations. Moreover, placement of the computing resources can be made simultaneously with the design of the network. Dynamic changes of links (link upgrades or failures) are not expected on-chip. Also, highly reliable link operation can be assumed, at least in the early generations of NoCs.

Based on the above considerations, given the cost sensitivity and the need to support various services and access paradigms, we suggest a NoC architecture and a process for its design, using the following characteristics: The modules are interconnected by a network of multi-port switches connected to each other by links composed of parallel point-to-point lines. The physical layer is optimized to take care of deep-submicron issues such as delay and repeater optimization, synchronization, noise immunity etc. The network applies a mesh topology and employs wormhole packet forwarding with hop-by-hop credit-based backpressure flow-control (for lossless buffer operation and minimal buffer requirements). The packets are forwarded using a static shortest path, X-Y coordinates-based routing (for minimal routing table operations, deadlock avoidance and no reordering at end-points). Packets can belong to different classes of service and packets of different classes are forwarded in an interleaved manner according to the QoS definitions (packets priorities). As a typical guideline we classify system traffic into four common classes of service: Signaling (replacing inter-module control signals); Real-Time (representing delay constrained bit streams); R/W (modeling short data access) and Block-Transfer (providing for large data bursts and DMA operations). We model traffic behavior for each class and define the Quality of Service requirements of each class in terms of throughput, end-to-end delay and relative priority. Unlike other wormhole routing systems these requirements are recorded at all switches and different packet forwarding is interleaved according to the QoS rules. For example a high priority Signaling packet will preempt the transmission of a long Block-Transfer packet. Similar to [4] we employ a design process, starting from a generic topology and proceeding to a customized network. The layout of the network is customized and links bandwidth is allocated according to their relative load so that the overall utilization of links in the network is balanced. During customization unnecessary resources (links, routers and buffers) are trimmed where possible, resulting in a low cost customized layout for the specific SoC. Traffic simulations are used in the cost optimization steps to ensure that QoS is satisfied. We also introduce a simple methodology to evaluate the cost in area and power of the resulting network. The area cost is based on total wire length and the amount of packet switch logic (buffers, tables etc.). The power cost is based on summation of the traffic that traverses each wire length and is received by input stages. It is easy to realize that for a given traffic metric and network topology, the power consumption will be reduced by employing shortest path routing and by the elimination of packets losses within switches.

The rest of this paper is organized as follows: Section II presents the network architecture, Section III describes the network design process, Section IV discusses simple design examples, and Section V provides simulation results for the network examples along with observations and conclusions.

II. NOC ARCHITECTURE

Our QoS network architecture is based on a grid topology and wormhole packet routing, following [1], [6], [7]. Wormhole routing [1], [22] reduces latency and buffer requirements in the routers. Circuit switching [7] is avoided in our architecture due to the high cost of establishing and managing circuit connections. Similarly, store-and-forward routing techniques [7] are also avoided as they may incur high buffer requirements and consequently a high penalty in silicon area of the router. The network does not drop packets, and links are assumed reliable so that no retransmissions are required. Packets traverse the network along the shortest route, thus minimizing power dissipation and maximizing network resource utilization. The architecture has served as a platform for developing the NoC design process and cost metrics (Section III), and has been modeled and simulated, as described in Section IV. In this section we describe the NoC topology, service levels, link communications, router design and interface to the system modules of the chip.

1. NoC Topology

Networks on chip comprise routers interconnected by point-to-point links. Topology can vary depending on system needs and module sizes and placement. Fat tree [6], folded torus [1] and regular mesh [4], [7] topologies have been proposed for NoC (Figure 1). We propose an irregular mesh topology as a best match for the typically irregular planar and rectangular structure of common SoC floorplans. Each system module is connected to a router (Figure 1c) via a standard interface, where the bandwidth is adapted to the communication needs of the module. The bandwidth of each inter-router link is similarly adjusted to accommodate the

expected traffic and fulfill QoS requirements at the specific link. Link and interface bandwidth is adjustable by changing either the number of wires or the data frequency. In addition, a module may be connected to the network through more than one interface.

Routing is performed over fixed shortest paths, employing a symmetric X-Y discipline whereby each packet is routed first in an “X” direction and then along the perpendicular dimension or vice versa. This scheme leads to a simple, cost-effective router implementation. Network traffic is thus distributed non-uniformly over the mesh links, but each link’s bandwidth is adjusted to its expected load, achieving an approximately equal level of link utilization across the chip.

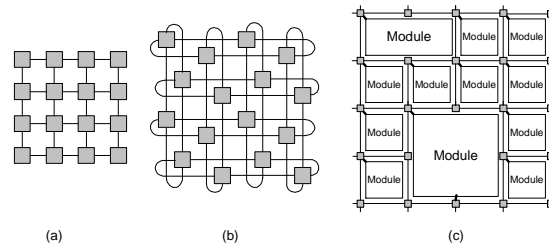


Figure 1. NoC topologies: (a) regular mesh; (b) folded torus; (c) irregular mesh-custom topology

2. NoC Service Levels

The principal goal of an on-chip interconnection network is to provide for all communication demands of heterogeneous modules within the chip. A NoC should replace not only shared buses but also other types of dedicated inter-modular wires and interfaces. We identify four different types of communication requirements and define appropriate service levels (SL) to support them: Signaling, Real-Time, Read/Write and Block-Transfer. Two classes of service were previously proposed in this context in [7]: best-effort and guaranteed throughput.

Signaling covers urgent messages and very short packets that are given the highest priority in the network to assure shortest latency. This service level is suitable for interrupts and control signals and alleviates the need for dedicating special, single-use wires for them.

Real-Time service level guarantees bandwidth and latency to real-time applications, such as streamed audio and video processing. This service (like all other ones) is packet based (and does not employ virtual circuits); a certain maximal level of bandwidth may be allocated to each real-time link and it should not be exceeded. This is achieved either by the design of each module or by enforcement circuits in the network.

Read/Write (RD/WR) service level provides bus semantics and is hence designed to support short memory and register accesses.

Block-Transfer service level is used for transfers of long messages and large blocks of data, such as cache refill and DMA transfers.

A linear priority ranking is established among the four service levels, where Signaling is given the highest priority and Block-Transfer the lowest. Below we describe a preemptive communication scheduling where data of a higher priority packet is always transmitted before that of a lower service level (a round-robin is employed within each service level). Thus, service levels are simply implemented by means of a priority mechanism. Additional service levels may be defined if desired, as long as a linear priority ranking is adhered to. For instance, the RD/WR service level may be split into normal and urgent RD/WR sub-levels.

3. NoC Communication

Packets carry routing information, command and payload. Figure 2 shows the basic packet format. The TRA (Target Routing Address) field contains the address required for routing. The command field identifies the payload, specifying the type of operation. The rest is an arbitrary length payload, including operation-specific control information such as sender identification.

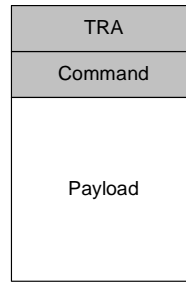


Figure 2. Packet format

The packet is divided into multiple *flits*[21] and transmitted over the *Data_o* signals (see Table1). Flit transfer over the link is controlled by handshake. The flits are classified into the following types:

- FP (Full Packet): A one-flit packet
- EP (End of Packet): Last flit in a packet
- BDY (Body): A non-last flit

Thus, all but the last flit in a packet are tagged BDY. The first flit of a packet can be detected as the first valid flit following either a FP or EP flit (this identification triggers the routing mechanism). Flit type and service level are indicated on separate *out-of-band* control wires (Table1).

4. NoC Routers

Routers connect to up to five links (Figure 3), designed for planar interconnect to four mesh neighbors and to one chip module. The router forwards packets from input ports to output ports. Data is received in flits. Every arriving flit is first stored in an input buffer. On the first flit of a packet, the router determines to which output port that packet is destined. The router then schedules the transmission for each flit on the appropriate output port. Figure 4 demonstrates data flow in a 4-link router.

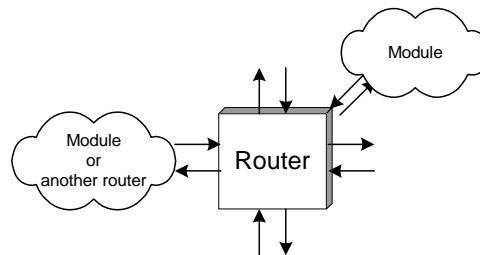


Figure 3. The router has up to five links and may connect to neighbor mesh routers or to chip modules.

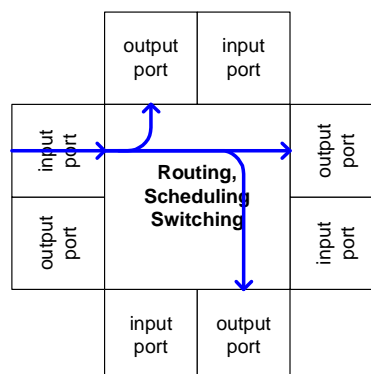


Figure 4. Router – data flow

There are separate buffers for each of the four service levels (“direct buffer mapping”). Relatively small buffers are allocated to

each service level, capable of storing only a few flits (this is a tunable design parameter). The routing algorithm is invoked when the first flit of a packet is received. The algorithm uses a simple routing function. For instance, relative routing is employed for X-Y routing. Routing information per each service level per each input port is stored in the *Current Routing Table* (CRT; Figure 5), until the tail flit of the packet is received, processed and delivered. When a flit is forwarded from an input to an output port, one buffer becomes available and a *buffer-credit* is sent back to the previous router on separate out-of-band wires (Table2).

Each output port of a router is connected to an input port of a next router via a communication link. The output port maintains the number of available flit slots per each service level in the buffer of the next input port. These numbers are stored in the *Next Buffer State* (NBS) table (Figure 5). The number is decremented upon transmitting a flit and incremented upon receiving a buffer-credit from the next router. When a space is available, the output port schedules transmission of flits that are buffered at the input ports and waiting for transmission through that output port, as detailed below.

We describe a simple handshake interface to each of the links. Other interfaces, such as asynchronous, are also possible. The same interface is employed whether the link connects to a chip module or to another router. The output port transmits a flit on the rising edge of the link clock (identifying the flit with a non-idle type, Table 1), and the input port samples a new flit on the falling edge. The clock could be replaced by a *valid* signal, which toggles only when a flit needs to be transmitted, alleviating the need for an “idle” flit type. Tables 1,2 summarize the output and input signals of the output port.

<i>Output Signals</i>	<i>Width [bit]</i>	<i>Description</i>
<i>Clk</i>	1	Source-synchronous clock indicating flit transmission
<i>Data_o</i>	Parameter	Data out of the router
<i>Type</i>	2	Type of flit: 00: IDLE 01: EP – end of packet 10: BDY - packet body 11: FP - full packet Note: start of packet is implied by non-idle flit following {EP, FP} per each SL
<i>SL</i>	2	Flit service level

TABLE1: INTERFACE SIGNALS OF OUTPUT PORT: OUTPUT DIRECTION

<i>Input Signals</i>	<i>Width [bit]</i>	<i>Description</i>
<i>Buffer_Credit_SL</i>	4	A buffer space for one flit at each specified service levels has become available.
<i>Buffer_Credit_valid</i>	1	Indicates that <i>Buffer_Credit_SL</i> lines carry a valid credit.

TABLE2: INTERFACE SIGNALS OF OUTPUT PORT: INPUT DIRECTION

We now turn to the mechanics of flit transfer inside the router. Flits are buffered at the input ports, awaiting transmission by the output ports. Flit routing (namely, to which output port each flit is targeted) is resolved upon arrival of the first flit of a packet and the output port number is stored in CRT for the pending flit per each input port and per each service level. Each output port schedules transmission of the flits according to the availability of buffers in the next router, the priority (namely service level) of the pending flits, and the round-robin ordering of flits within the same service level. The numbers of available flit slots in the

buffers at the next routers are stored in the NBS tables for each service level at each output port. Service level priorities are ranked with Signaling having the highest priority, Real-Time being second, RD/WR third and Block-Transfer ranked last. A round-robin arbitration is performed on input ports of the same service level. The present state of round-robin scheduling is stored in the *Currently Served Input Port* number (CSIP) table for each service level at each output port (Figure 5). This number is advanced when transmission of a complete packet is finished or if there is nothing to transmit from a particular input port and service level.

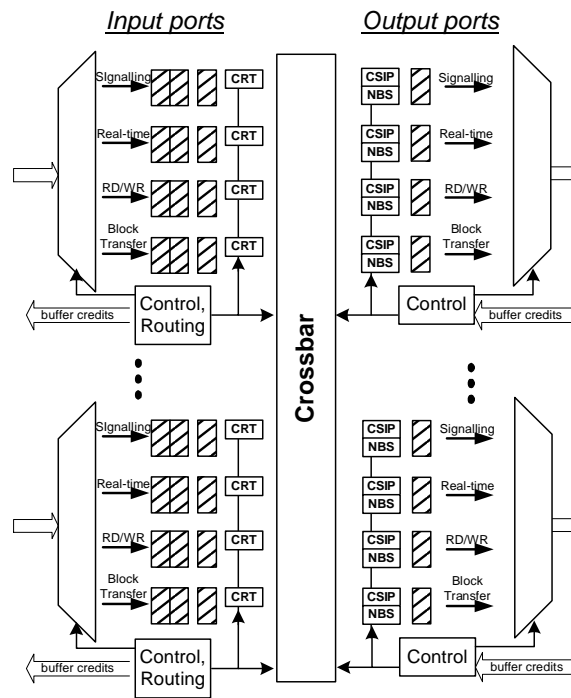


Figure 5. Router Architecture

This scheduling discipline implies that a particular flit gets transmitted on an output port as long as there is buffer space available on the next router and there is no packet with a higher priority pending for that particular output port. Once a higher priority packet appears on one of the input ports, transmission of the current packet is preempted and the higher priority packet gets through. Transmission of the lower priority packet is resumed only after all higher priority packets are serviced. In the example of Figure 6, a Real-Time packet is preempted by a Signaling packet. The network is designed with bounded traffic requirements in the higher service levels, to avoid starvation of RD/WR and Block-Transfer communications.

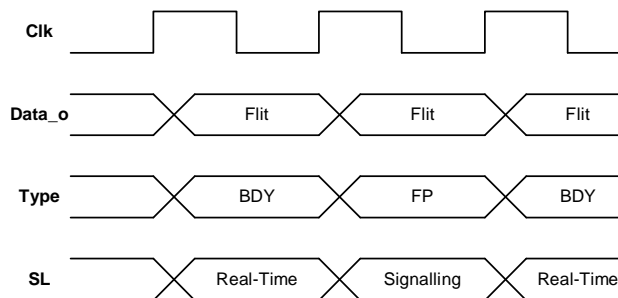


Figure 6. Tx waveforms: A **Real-Time** packet is preempted by a single-flit **Signalling** packet

5. NoC Interface

The network interface connects modules to the network. It maps a variety of transactions onto the four structured NoC service levels. For instance, it hides the packet switching character of the network when one module needs to access another using bus and

conventional read/write semantics.

This section has presented the NoC architecture: Simple and efficient routers are interconnected with short point-to-point links and arranged as an irregular mesh. Communication is organized in four service levels; a preemptive priority scheduling is established in the router among the levels, while round-robin scheduling is employed within each level. Network resources are adjusted to fit a given traffic pattern, as described in the following section.

III. NOC DESIGN PROCESS

In this section we present a design process for constructing a low cost NoC. In traditional communication networks the topology and link capacities are given, and the routing and congestion control processes balance the a-priori unknown traffic loads. NoC design process has more degrees of freedom as the topology of the network, network resources and protocols can be changed and tuned by the network designer for a particular SoC with particular requirements. Design effort shifts to adapting the network to given traffic flows and QoS requirements and optimizing it for low cost in terms of area and power. The block diagram in Figure 7 summarizes the NoC design flow.

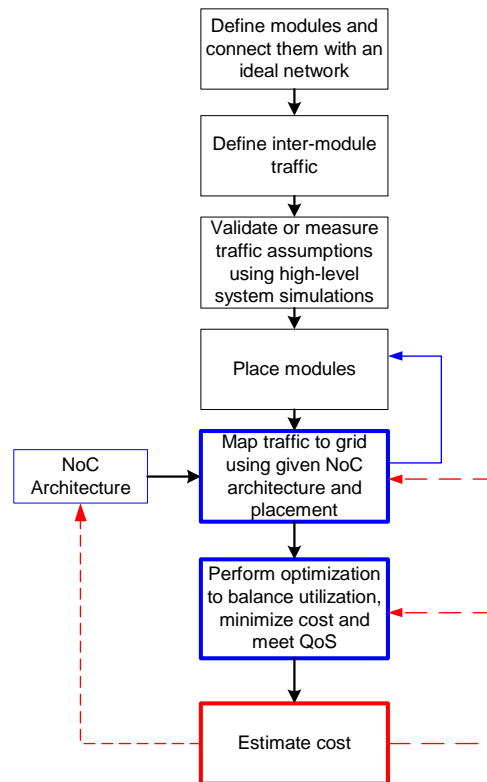


Figure 7. NoC Design flow chart

First, the functional system modules are defined and assumed to be connected by an ideal interconnection infrastructure with unlimited bandwidth and programmable delay. Then, inter-module traffic is characterized. This characterization is conducted by analyzing the interconnected modules and their traffic specification. To verify the assumptions or as an alternative way for characterization, the inter-module traffic is measured and partitioned into traffic classes using a high-level multi-module operational simulation. Similarly, QoS requirements are derived for each traffic class by observing the actual performance as well as evaluating via simulation the effect of delays and throughput. The characterization and requirements derivation stage should take into account that NoC cost will increase if safety margins are excessive. Once traffic patterns are specified, modules are placed so as to minimize the system spatial traffic density.

Only after system module placement and inter-modular communication requirements are determined, the NoC can be constructed. The NoC architecture is finalized, and architectural parameters are set according to the number of modules, their spatial placement, and the QoS service levels to be supported. The initial topology is set to a mesh grid and the required traffic is mapped onto the mesh grid according to the routing algorithm, such as X-Y routing. As parts of the grid are not fully utilized, some vertices and links can be eliminated as shown in Figure 1.c. Once the routing algorithm is selected, communication paths between all pairs of modules can be determined and link bandwidth optimization can be performed. Average traffic load at each

link can be calculated since routing is fixed and traffic patterns are known in advance. Link bandwidth can be assigned proportionally to the calculated load on that link. In that way the designer calibrates the system resources so that average utilization of all links in the network is approximately equal. At this point, the average load calculation provides only relative link bandwidths. To finalize the design, the NoC can be simulated and analyzed more precisely by a network simulator. Actual bandwidth can then be assigned to the links according to QoS requirements and the supporting simulation results. Further optimizations can be performed: Buffers and routers can be trimmed where possible while maintaining the required QoS. The entire design process may be iterated if hardware cost of the resulting NoC is too high, or if other NoC architectures need to be investigated.

NoC Cost Estimation

It is important to estimate accurately the cost of the NoC, as it directly influences the cost of the entire system. By having a good measure of cost, the system architect can compare different solutions that provide the same performance and choose the most cost-effective one. We employ area and power cost functions, as is common for VLSI systems, by comparing architectures having the same performance (in terms of delay and throughput) and quantifying their area and power requirements.

The cost of NoC architecture consists of two main factors: the cost of routers and module interfaces (logic cost), and the cost of wires of the links that interconnect them. We assume that the logic and links are designed in an effective way such that power is consumed only when information is flowing through these components (only when logical transitions happen). For power saving, packets traverse the network on the shortest path. In addition, no retransmissions of information are needed since the transmission on a link is reliable and hop-by-hop flow control (back pressure) prevents losing or dropping any packet. These facts result in a power efficient network architecture.

Wire cost: Since the distance between two adjacent wires is fixed, the area occupied by link wires on a chip is proportional to the total wire length:

$$Cost_{wire-area} = A_0 \cdot \sum_{i \in \{\text{NoC links}\}} W_i \cdot l_i \quad (1)$$

where: A_0 - constant, W_i - width of link i (number of bits), and l_i - length of link i . The dynamic power consumed by wires is proportional to wire length and thus wire length is a good estimator of power dissipated on wires. Dynamic power dissipation in switching circuits is:

$$P_d = C_L \cdot V_{dd}^2 \cdot f_p; \quad (2)$$

where: C_L - load capacitance, V_{dd} - supply voltage and f_p - switching frequency.

Switching frequency of every link is link frequency multiplied by the link utilization. C_L is the total load capacitance, consisting of wire capacitance (C_{wire}) and gate capacitance of the transistor driven by that wire (C_{gate}). We assume that C_{gate} can be neglected and the dominant factor is C_{wire} , which is directly proportional to the length of the wire:

$$Cost_{wire-power}(P_d) = P_0 \cdot U \cdot \sum_{i \in \{\text{NoC links}\}} f_i \cdot W_i \cdot l_i \quad (3)$$

where: P_0 - constant coefficient, U - utilization of the links, f_i - frequency of the link i .

Logic cost:

Logic cost consists of the cost of the routers and the cost of network interfaces of the system modules. In all interconnection architectures, a bus or a network, an interface logic must be implemented. It is beyond the scope of this paper to evaluate the exact interface mechanisms that need to be developed. However, it is clear that in a shared media interconnection (like a bus), each module interface must accommodate higher speed bursts of information as compared to a switched architecture and consequently the cost is increased. The cost of router is affected by several parameters: number of ports ($\#Port$), number of service levels ($\#SL$), flit size ($FlitSize$), buffer size for each service level ($BufSize$). A good estimation for the area cost of the router is flip-flop count. Generally the cost of a router is very much architecture specific. We give an estimate for the cost of the router in the architecture that was presented in Section II.4. The number of flip-flops in router is dominated by the flip-flops used for storing data and control information:

$$\#FF \approx \#Port \cdot \#SL \cdot \left[(FlitSize + 2) \cdot BufSize + \log_2 (BufSize \cdot (\#Port)^2) \right] \quad (4)$$

Total logic cost of NoC is summation of costs of all routers in the network:

$$Cost_{logic-area} \approx \sum_{i \in \{\text{Routers}\}} \#FF_i \quad (5)$$

IV. DESIGN EXAMPLE

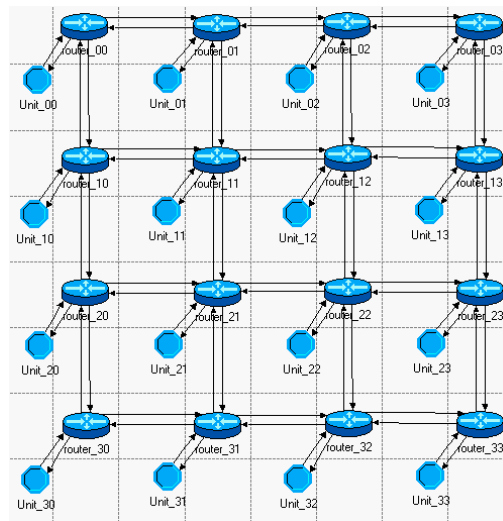


Figure 8. Simulated NoC: 16 system modules interconnected in a 4x4 mesh

As a simple example of NoC design for a given SoC, let us consider an array of 16 communicating units arranged in a 4X4 mesh. Each unit is connected to the NoC via a router that is a part of the network (see Figure 8). The routing mechanism is a symmetric X-Y routing when the destination x-coordinate is greater than source x-coordinate; otherwise, it is Y-X routing. In that way the traffic between each pair of nodes in both directions traverses the network on the same routing path. We assume that links operate at a frequency of 1GHz, which is a reasonable assumption for short links at current technology. We allocate the final link's bandwidth by adding or trimming wires from it. We define flit size to be 16 bits, and use minimal buffering requirements - buffers capable of storing two flits at each input port (so small Signaling or Real-Time packets will not be stretched beyond a single stage and create excessive blocking). The delay of packets in the network consists of the delay of the links and queuing delay in the network. We neglect the delay of routers logic, which should add only a few cycles to the overall delay. During the design process we assign traffic characteristics to the units and simulate the NoC behavior. OPNET[23] was chosen as a simulation framework for performing this task. OPNET provides a convenient tool for hierarchical modeling of a network, including processes (state machines), network topology description and simulation of different traffic scenarios. The NoC architecture described in Section II was fully modeled in the OPNET environment and simulated for two scenarios as follows:

1) Uniform Traffic scenario

Each unit communicates uniformly with all other units in the system. Each unit contains four traffic sources that correspond to four classes of system traffic: Signaling, Real-Time, RD/WR and Block-Transfer. Each source creates packets with specific distribution of packet size and packet inter-arrival time. In our simulations we changed the traffic load of each source to obtain different scenarios and we also changed the bandwidth allocation of each link in order to balance the utilization of all links. Representative cases are shown in the next section. We suppose that Signaling packets are of constant size of two flits and packets inter-arrival time is distributed uniformly between 50-150 ns. In other words we assume that each module in the system sends on average one Signaling packet every 100 ns to a random target in the mesh, and thus Signaling packets are sent from each module at an average rate of 40MB/s. For Real-Time traffic we have to mimic periodic connection between each pair of modules. We assume that each module transfers to another module about 320 voice channels of 64kb/s. At the beginning each module has a list of all target modules in the system (15 modules). It picks a random target and sends the first packet to it, then it increments the target index and the next packet is sent to the next target on the list and so on, so that periodic connections between all modules are created. A Real-Time source creates packets with size distributed uniformly in the range 20-60 flits, and packet inter-arrival time is distributed exponentially with an average value of $2\mu\text{s}$, so that we obtain a Real-Time source rate of 40MB/s. RD/WR packets are sent to random targets; packet size is distributed uniformly between 2-6 flits, packet inter-arrival time is distributed exponentially with a mean value of 25ns, so that an average RD/WR packet load of 320MB/s is obtained. Finally, a Block-Transfer source creates packets with constant size of 2000 flits with exponential inter-arrival time having a mean value of $12.5\mu\text{s}$, so that total Block-Transfer load is 320MB/s. According to that benchmark we obtain average traffic load from each module in the system of about 720MB/s, or a total load of 11.52GB/s for the entire 16-modules SoC. This is only one representative example; in our simulations we also checked cases with higher and lower traffic loads.

2) Non-uniform Traffic scenario

Uniform traffic distribution is unrealistic. More realistic traffic exhibits a non-uniform communication patterns with higher traffic locality. Moreover, according to the proposed design process of the network (Section III), system modules are placed considering their inter-module traffic so as to minimize the system spatial traffic density. In our non-uniform benchmark the network topology and traffic load of the sources is the same as in the uniform-traffic case (Section IV.1), but the probability that a module will send a packet to one of its adjacent neighbors is twice the probability to send the packet to any other module in the network.

In order to analyze the results of our benchmarks we define QoS requirements in terms of packet end-to-end (ETE) delay for each class of service. ETE delay is defined as the sum of the queuing time at the source and travel time through the network experienced by 99% of the packets. The final NoC configuration must meet those requirements, which are typically defined by the system architect. In our example, we have chosen the maximum ETE delay of a Signaling packet to be no more than 20-30ns, for Real-Time packets we require ETE delay to be the order of magnitude of 125 μ s, since our Real-Time is voice connection and it shouldn't be more than several frames of a 8KHz clock, and for RD/WR packets we allow ETE delay \sim 100 nanoseconds. In order to obtain QoS requirements for Block-Transfer packets we consider an alternative solution of a typical system bus that traverses the chip and interconnects all modules on the chip, the bus width is 32 bits and it operates at 50MHz so that its total bandwidth is 1.6Gbps. Just transmission time of one Block-Transfer packet (32000 bits) on such a bus lasts 20 μ s. Hence we allow ETE delay of a Block-Transfer packet in the NoC to be no more than several times its transmission time on a typical system bus.

V. OBSERVATIONS AND CONCLUSIONS

1. Uniform Traffic Scenario Results

We used the design process described in Section IV and applied a uniform traffic load. The modules were placed in a full mesh. Relative traffic load on all the links of the mesh is shown in Figure 9, where bar height stands for a link's relative load. For example, links (0,0) \rightarrow (1,0) and (2,0) \rightarrow (3,0) have the smallest load in the system, denoted by 1 unit. Other link loads are measured relative to the load on those two links. The highest relative load in the mesh is on link (1,3) \rightarrow (2,3), reaching 9.3. This load distribution originates from traffic distribution and module locations (which are symmetric in our case) and from X-Y coordinates routing, as described in Section IV.

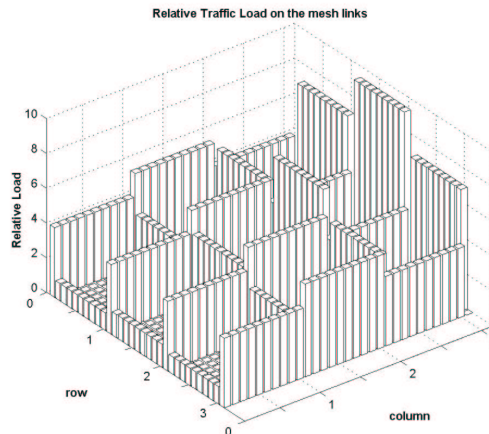


Figure 9. Uniform traffic scenario: Relative load on mesh links, also relative link bandwidth allocation

Next, link bandwidth was allocated according to the ratios shown in Figure 9. That allocation led to balanced utilization of the mesh links. We applied the uniform traffic load described in Section IV (11.52GB/s from each module) and simulated several total network bandwidth levels. ETE delay was measured at each destination module according to packet service levels. ETE delay was measured in clock cycles of the link (since we assume that links operate at 1GHz, each cycle represents a delay of one nanosecond). Total network bandwidth allocations of 2560Gbps, 1280Gbps, 850Gbps and 512Gbps resulted in different distributions of packet ETE delay, and network utilization of 10.3%, 20%, 30% and 44% respectively (see Figure 10 - Figure 13). In the first two cases (Figure 10 and 11) the network is underutilized and delivers better performance than required. By reducing bandwidth (and thus reducing cost) we obtain a network that operates at 30.4% utilization (Figure 12). It can be seen that this network configuration delivers the required QoS. Specifically, 99.9% of the Signaling packets arrived with ETE delay of less than

20ns (as required), 99.9% of Real-Time packets arrived with ETE delay of less than 250ns (over-performing, we only required less than 125 μ s), 99% of RD/WR packets arrived with ETE delay of less than 80ns (as required) and 99.9% of Block-Transfer packets arrived with ETE delay of less than 50 μ s. That is 2.5 times the transmission time of this packet on an assumed system bus. If we try to reduce the cost any further, the network will not be able to satisfy our QoS requirements as shown in Figure 13, where requirements for delay of Signaling and Block-Transfer packets are not met.

In order to estimate the cost of NoC systems we use the cost metrics described in Section III. Total wire-length of the links considering data and control wires is \sim 3.7m. The cost of the routers is estimated by flip-flop count which results in \sim 10K flip-flops. Power dissipation is calculated using equation (3): $P_{NoC,uniform} = 1.11 P_0$

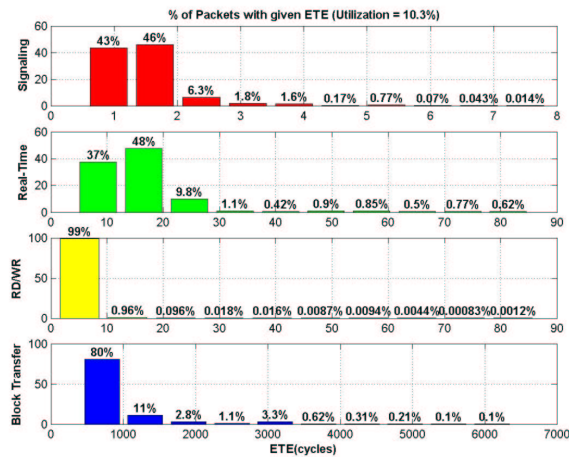


Figure 10. Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 2560Gbps (10.3% utilization). 99.7% of Signaling packets arrived with ETE delay less than 6ns, 99.9% of Real-Time packets arrived with ETE delay less than 80ns, 99.9% of RD/WR packets arrived with ETE delay less than 20ns and 99% of Block-Transfer packets arrived with ETE delay less than 4 μ s.- Performance is better than required.

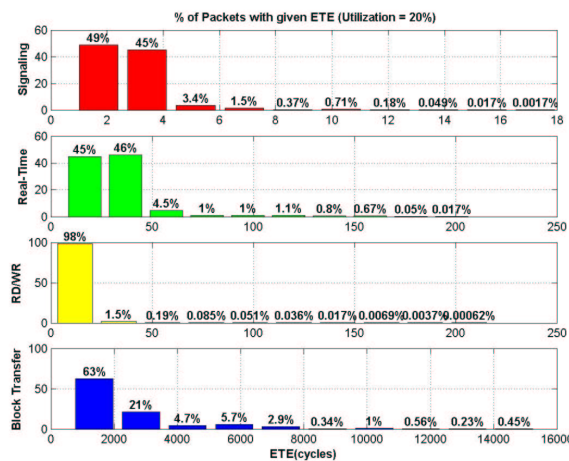


Figure 11. Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 1280Gbps (20% utilization). 99.7% of Signaling packets arrived with ETE delay less than 11ns, 99.9% of Real-Time packets arrived with ETE delay less than 150ns, 99.9% of RD/WR packets arrived with ETE delay less than 50ns and 99% of Block-Transfer packets arrived with ETE delay less than 12 μ s.- Performance is better than required.

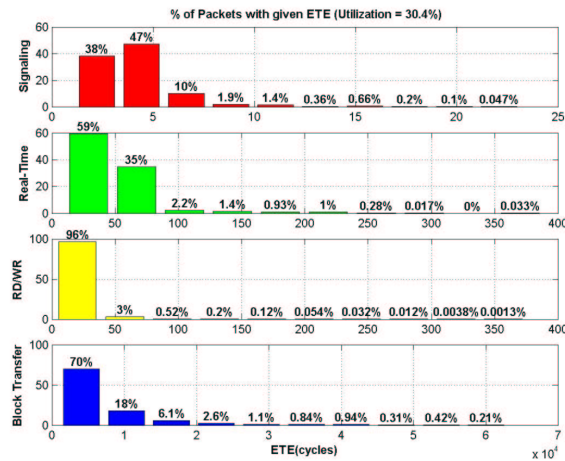


Figure 12. Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 850Gbps (30.4% utilization). 99.7% of Signaling packets arrived with ETE delay less than 20ns, 99.9% of Real-Time packets arrived with ETE delay less than 250ns, 99.9% of RD/WR packets arrived with ETE delay less than 80ns and 99% of Block-Transfer packets arrived with ETE delay less than 50 μ s-Delivers required QoS.

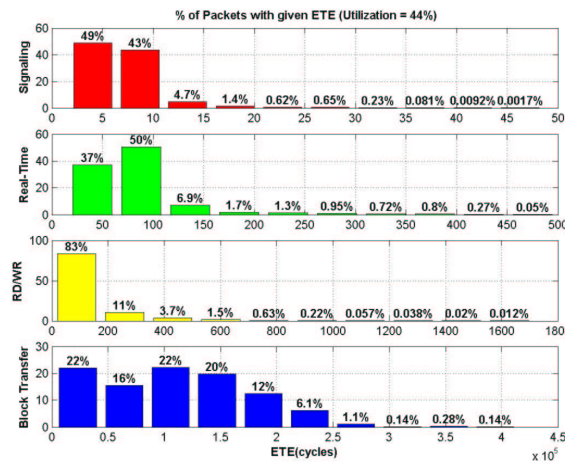


Figure 13. Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 512Gbps (44% utilization). 99.7% of Signaling packets arrived with ETE delay less than 35ns, 99.9% of Real-Time packets arrived with ETE delay less than 450ns, 99.9% of RD/WR packets arrived with ETE delay less than 1 μ s and 99% of Block-Transfer packets arrived with ETE delay less than 300 μ s-QoS requirements are not satisfied.

Another important issue is network behavior in terms of delay as a function of traffic load. We chose a fixed network configuration and bandwidth allocation and applied various traffic loads by reducing and expanding packet inter-arrival time for each service level. Figure 14 shows the mean ETE delay of packets at each service level as a function of traffic load in the network. One can observe that while the traffic load is growing, ETE delay of Block-Transfer and RD/WR packets grows exponentially, but the delay of delay-constrained traffic (Real-Time and Signaling) remains nearly constant. Since network resources are kept constant, network utilization grows when higher traffic load is applied (from 16% to 42% in the Figure).

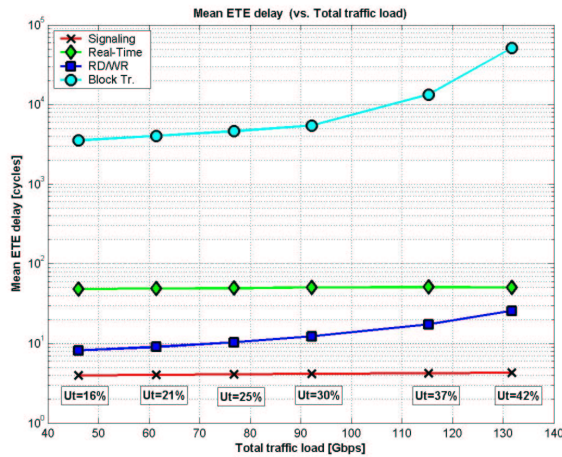


Figure 14. Uniform traffic: Mean ETE delay of packets at each service level vs. total load, using constant network bandwidth allocation.

2. Non-Uniform Traffic Scenario Results

Results for non-uniform traffic are shown in Figure 15. It can be observed that the ratios between links loads are smaller than in the uniform scenario and the overall traffic distribution is more balanced because of the higher locality in network traffic.

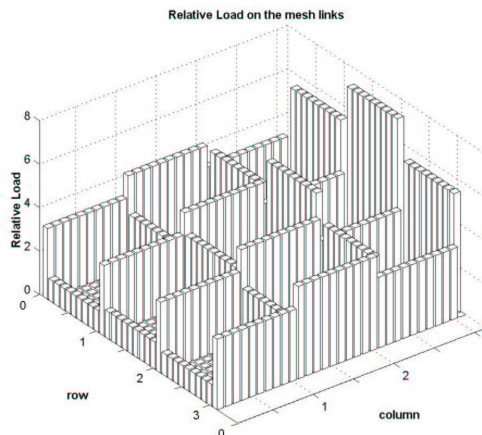


Figure 15. Non-Uniform traffic: Relative load on mesh links, also relative link bandwidth allocation

We used several network bandwidth allocations while applying the non-uniform traffic presented in Section IV. Total network bandwidth allocations of 2752Gbps, 1376Gbps, 688Gbps and 459Gbps resulted in different packets ETE delay distributions and network utilization of 8.2%, 16.5%, 33.5% and 44% respectively (see Figure 16-Figure 19). Again, the network was underutilized in the first two cases. Thus we reduced network bandwidth further, and it can be seen that the network operating at 33.5% utilization (Figure 18) was delivering the required QoS. In particular, 99.9% of Signaling packets arrived with ETE delay of less than 20ns (as required), 99.9% of Real-Time packets arrived with ETE delay of less than 270ns, 99% of RD/WR packets arrived with ETE delay of less than 120ns and 99.9% of Block-Transfer packets arrived with ETE delay less than 45 μ s. That is 2.3 times the transmission time of the same packet on a system bus. If we try to reduce the cost any further, the network will not be able to satisfy our QoS requirements, for example for Signaling and Block-Transfer packets (see Figure 19).

The fact that network traffic in the non-uniform scenario is more local makes it possible to provide the required QoS using less network resources compared with the uniform scenario. Indeed, total wire length of the links considering data and control wires in this case is \sim 3.2m, compared with 3.7m in the uniform scenario. This is a 14% reduction in the wire cost of the links. Power dissipation is calculated using equation (3): $P_{NoC, non-uniform} = 1.056 P_0$, compared with $1.11 P_0$ in the uniform traffic case.

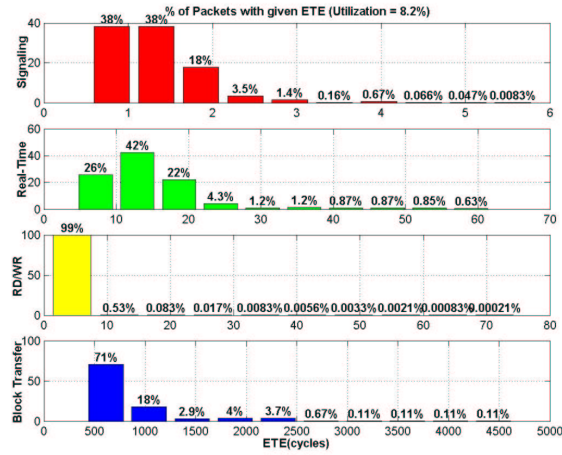


Figure 16. Non-Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 2752Gbps (8.2% utilization). 99.9% of Signaling packets arrived with ETE delay less than 5ns, 99.9% of Real-Time packets arrived with ETE delay less than 60ns, 99.9% of RD/WR packets arrived with ETE delay less than 20ns and 99.9% of Block-Transfer packets arrived with ETE delay less than 4.5µs- Performance is better than required

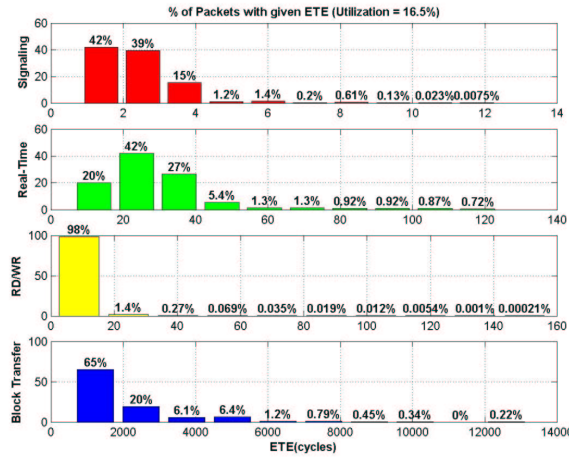


Figure 17. Non-Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 1376Gbps (16.5% utilization). Network operates with average link utilization of 16.5%. 99.9% of Signaling packets arrived with ETE delay less than 10ns, 99.9% of Real-Time packets arrived with ETE delay less than 120ns, 99.9% of RD/WR packets arrived with ETE delay less than 50ns and 99.9% of Block-Transfer packets arrived with ETE delay less than 13µs -Performance is better than required

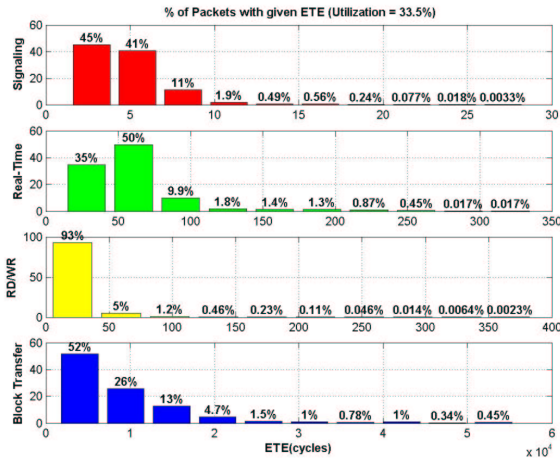


Figure 18. Non-Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 688Gbps (33.5% utilization). 99.9% of Signaling packets arrived with ETE delay less than 20ns, 99.9% of Real-Time packets arrived with ETE delay less than 270ns, 99.9% of RD/WR packets arrived with ETE delay less than 120ns and 99.9% of Block-Transfer packets arrived with ETE delay less than 45 μ s-Delivers required QoS.

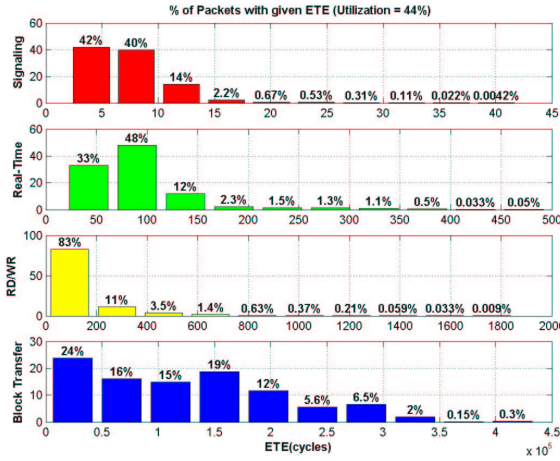


Figure 19. Non-Uniform traffic: Distribution of ETE delay for total NoC bandwidth of 459Gbps (44% utilization).. 99.9% of Signaling packets arrived with ETE delay less than 35ns, 99.9% of Real-Time packets arrived with ETE delay less than 400ns, 99.9% of RD/WR packets arrived with ETE delay less than 1.3 μ s and 99% of Block-Transfer packets arrived with ETE delay less than 350 μ s-QoS requirements are not satisfied.

Figure 20 shows mean ETE delay of packets at each service level as a function of traffic load in the network. These results are similar to the uniform traffic case.

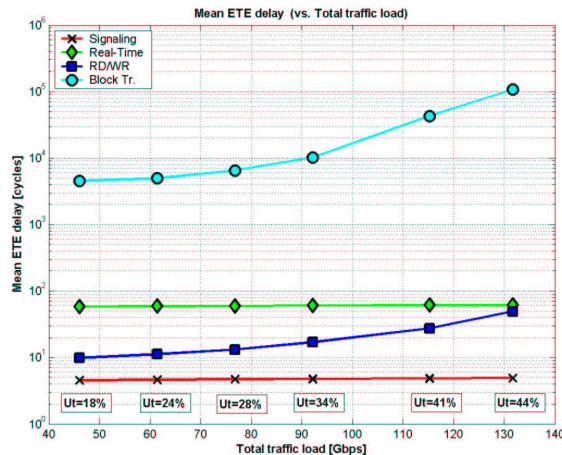


Figure 20. Non-Uniform traffic: Mean ETE delay of packets at each service level vs. total traffic load from each source using constant network bandwidth allocation.

3. Comparison with alternative solutions

In this section we compare the cost of NoC architecture in terms of area and power with the cost of alternative interconnection solutions that provide the same QoS: a system bus and dedicated point-to-point (PTP) links. We assume chip size of 12mm×12mm and we place 16 modules on it.

1) System Bus

A system bus in the uniform traffic load design example would have to deliver total traffic load of ~11.52GB/s. Let's also assume that this bus operates at 50 MHz and that it will deliver the required QoS under utilization of 50% (a very optimistic assumption for the given QoS requirements). In order to be competitive with NoC performance, such a bus would require at least 3700 wires. The bus has to connect to all modules on the chip, and as a result its length would be ~25mm. In practice, system buses are multiplexed and there are actually two unidirectional buses. Even if we neglect the significant cost of the multiplexing logic, we obtain a total wire length of ~180m for such a bi-directional bus, as compared with the 3.7m of the NoC. Power dissipation on such a bus is calculated using equation (3) again: $P_{bus,uniform} = 4.5 P_0$, as compared with $\sim 4.5 P_0$ of the NoC.

2) Dedicated Point-To-Point (PTP) links

We assume that each module is connected to all other modules by dedicated wires. We further assume that point-to-point links operate at 100 MHz. In order to provide the required performance (several times the transmission time of Block-Transfer packet on a system bus), the PTP link should consist of ~6 wires (five data wires and one control wire) and should operate with 80% utilization. Total length of wires that interconnect all 16 modules on chip is ~11.4m. Power dissipation is: $P_{ptp,uniform} = 0.9 P_0$.

The comparison of the alternative interconnection architectures for the uniform traffic example is summarized in Figure 21. It should be noted that the cost of NoC is several times lower than the cost of bus, both in terms of power dissipation and wire length. The PTP area is also higher than that of the NoC. Theoretically, a PTP interconnect should consume the same power as the NoC, because the same traffic is transmitted along the same Manhattan distances and no power is wasted on idle links. However, because of smaller overhead of control wires, the power dissipation of point-to-point solution is slightly lower than in NoC.

For the non-uniform example, the cost of the bus remains the same, because in the bus each transaction is propagated all over the chip and it cannot benefit from higher traffic locality. NoC cost is reduced (14% reduction in our example) because it benefits directly from traffic locality since less traffic has to be transferred for longer distances. PTP interconnect will also benefit from traffic locality, but its cost remains higher.

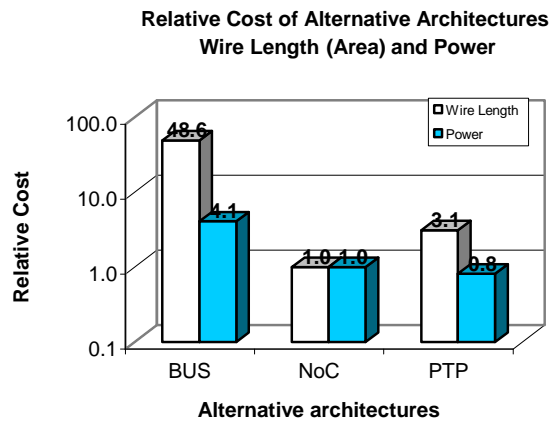


Figure 21. Relative cost of three compared interconnection architectures (system bus, NoC and point-to-point interconnect) in terms of area and power for uniform traffic design example. Relative cost of NoC is one and the cost of system bus and PTP interconnect is measured relatively to NoC costs.

Bus and PTP solutions cost will rise rapidly in more complicated design examples (with more communicating modules). Buses have no parallelism, hence capacitance will grow, frequency will degrade and many more wires will be needed to compensate for the frequency degradation and to satisfy the growing communication demands. The same is true for PTP solution: wire cost will grow quadratically with the number of modules and the power cost will be similar to the power cost of the NoC. On the other hand, NoC is more scalable and it benefits from the parallelism and spatial reuse of the network links and from the fact that links will still be short and cheap and would be still able to operate at a high frequency.

VI. CONCLUSIONS

In this paper we presented a novel NoC design process, targeted to achieve a pre-defined Quality of Service. The NoC design process characterizes inter-module traffic, places the system modules and constructs a customized, cost-effective NoC. We have identified several QoS classes and introduced a QoS-based wormhole routing architecture that supports these classes. We showed design examples demonstrating that it is possible to construct cost-effective, scalable Network on Chip that provides the required quality of service with area and power cost lower than alternative simple interconnection solutions.

REFERENCES

- [1] William J. Dally and Brian Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA.
- [2] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, A. Sangiovanni-Vincentelli, "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design", Design Automation Conference, DAC '01, June, 2001.
- [3] Luca Benini, Giovanni De Micheli, "Networks on Chips: A New SoC Paradigm", IEEE Computer 35(1): 70-78 (2002)
- [4] Shashi Kumar, Axel Jantsch, Juha-Pekka Soininen, Martti Forsell, Mikael Millberg, Johnny Oberg, Kari Tiensyrja and Ahmed Hemani, "A Network on Chip Architecture and Design Methodology", Proceedings of the IEEE Computer Society Annual Symposium on VLSI 2002 (ISVLSI.02)
- [5] Ahmed Hemani, Axel Jantsch, Shashi Kumar, Adam Postula, Johnny Oberg, Mikael Millberg, Dan Lindqvist, "Network on a Chip: An architecture for billion transistor era", In Proceeding of the IEEE NorChip Conference, November 2000.
- [6] Pierre Guerrier, Alain Greiner, "A generic architecture for on-chip packet-switched interconnections", Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, 2000 Page(s): 250-256
- [7] E.Rijpkema, K. Goossens and P.Wielage, "A Router Architecture for Networks on Silicon", Proceedings of Progress 2001, 2nd workshop on embedded systems
- [8] K. Goossens, J. van Meerbergen, A. Peeters, and P. Wielage, "Networks on Silicon: Combining Best-Effort And Guaranteed Services", DATE 2002, Design automation and test conference, March, 2002.
- [9] Andrei Radulescu and Kees Goossens. In Shuvra Bhattacharyya and Ed Depretere and Juergen Teich, editors, "Communication Services for Networks on Silicon", Domain-Specific Processors: Systems, Architectures, Modeling, and Simulation. Marcel Dekker, 2003.
- [10] Paul Wielage and Kees Goossens, "Networks on Silicon: Blessing or Nightmare?", Euromicro Symposium On Digital System Design (DSD 2002), Dortmund, Germany, September 2002.
- [11] W.J. Bainbridge, S.B. Furber, "Chain: A Delay Insensitive Chip Area" Micro, IEEE, Volume: 22 Issue: 5, Sep/Oct 2002 Page(s): 16-23
- [12] "AMBA Specification", Arm Inc, May 1999
- [13] "The CoreConnect Bus Architecture", IBM, 1999
- [14] Drew Wingard, "MicroNetwork-based integration of SOCs", In Proceedings of the 38th Design Automation Conference, June 2001.

- [15] C.H.Sequin and R.M.Fujimoto:"X-Tree and Y-Components",VLSI architecture, prentice hall international,1983 pp. 70-78
- [16] Jennifer Rexford, John Hall and Kang G. Shin: "A Router Architecture for real-time communication in Multicomputer networks", IEEE trans. On computers, vol. 47, no.10, october1998.
- [17] Mukherjee, S.S.; Bannon, P.; Lang, S.; Spink, A.; Webb, D. - Compaq Computer Corp., "The alpha 21364 network architecture", IEEE micro, pp.26-35,January-February 2002
- [18] W. J. Dally, C.L. Seitz, "The Torus Routing Chip", Distributed Computing pp.187-196, 1986
- [19] InfiniBand™ Architecture Specification, Volume 1, Release 1.0, October 24, 2000
- [20] C.B. Stunkel, J.Herring, B. Abali, R.Sivaram, " A New Switch Chip or IBM RS/6000 SP Systems", Proceedings of the 1999 conference on Supercomputing, January, 1999
- [21] W. J. Dally, "A VLSI Architecture for Concurrent Data Structures", Kluwer Academic Publishers,1987.
- [22] L.M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", IEEE Computer, pp.62-75, February 1993
- [23] OPNET Modeler, www.opnet.com