A Low Latency Router Supporting Adaptivity for On-Chip Interconnects *

Jongman Kim, Dongkook Park, T. Theocharides, N. Vijaykrishnan and Chita R. Das
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{jmkim, dpark, theochar, vijay, das}@cse.psu.edu

ABSTRACT

The increased deployment of System-on-Chip designs has drawn attention to the limitations of on-chip interconnects. As a potential solution to these limitations, Networks-on -Chip (NoC) have been proposed. The NoC routing algorithm significantly influences the performance and energy consumption of the chip. We propose a router architecture which utilizes adaptive routing while maintaining low latency. The two-stage pipelined architecture uses look ahead routing, speculative allocation, and optimal output path selection concurrently. The routing algorithm benefits from congestion-aware flow control, making better routing decisions. We simulate and evaluate the proposed architecture in terms of network latency and energy consumption. Our results indicate that the architecture is effective in balancing the performance and energy of NoC designs.

Categories and Subject Descriptors: B.4[I/O and Data Communications] Interconnections(Subsystems):B.8[Perfromance and Reliability] Performance Analysis and Design Aids

General Terms: Design, Performance.

Keywords: Networks-On-Chip, Adaptive Routing, Interconnection Networks.

1. INTRODUCTION

With the growing complexity of System-on-Chip (SoC) architectures, the on-chip interconnects are becoming a critical bottleneck in meeting performance and power consumption budgets of the chip design. The ICCAD 2004 Keynote Speaker [17] emphasized the need for an interconnect centric design by illustrating that in a 65nm chip design, up to 77% of the delay is due to interconnects. Packet-based on chip communication networks [10, 5, 4] (a.k.a network-on-chip (NoC) designs) have been proposed to address the challenges of increasing interconnect complexity.

The design of NoC imposes several interesting challenges as compared to traditional off-chip networks. The resource limitations - area and power limitations - are major constraints influencing NoC designs. Early NoC designs used dimension order routing, due to its simplicity and deadlock avoidance. However, traditional networks enjoy complicated routing algorithms and protocols, which provide adaptivity to various traffic topologies, handling congestion as it evolves in the network. The challenge in using adaptive routing in NoC designs, is to limit the overhead in implementing such a design.

In this work, we present a low-latency two-stage router architecture suitable for NoC designs. The router architecture uses a speculative strategy based on lookahead information obtained from neighboring routers, in providing routing adaptation. A key aspect of the proposed design is its low latency feature that makes the lookahead information more representative than possible in many existing router architectures with higher latencies. Further, the router employs a pre-selection mechanism for the output channels that helps to reduce the complexity of the crossbar switch design.

We evaluated the proposed router architecture by using it in 2D mesh and torus NoC topologies, and performing cycle-accurate simulation of the entire NoC design using various workloads. The experimental results reveal that the proposed architecture results in lower latency than when using a deeper pipeline router. This results from the more up to date congestion information used by the proposed low-latency router. We also demonstrate that the adaptivity provides better performance in comparison to deterministic routing for various workloads. We also evaluate our design from an energy standpoint, as we designed and laid out the router components, and obtained both dynamic and leakage energy consumption of the router. Our results indicate that for non-uniform traffic, our adaptive routing algorithm consumes less energy than dimension order routing, due to the decrease in the overall network latency.

This paper is organized as follows. First, we give a short background of existing work in Section 2. We present the proposed router architecture and the algorithm in Section 3, and we evaluate our architecture in Section 4. Finally we conclude our paper in Section 5.

2. RELATED WORK

The quest for high performance and energy efficient NoC architectures has been the focus of many researchers. Fine-tuning a system into maximizing system performance and minimizing energy consumption includes multiple trade-offs that have to be explored. As with all digital systems, energy consumption and system performance tend to be contradictory forces in the design space of on-chip networks. Router architectures have dominated early

*This research was supported in part by NSF grants CCR-0093085, CCR-0098149, CCR-0208734, CCF-0429631, EIA-0202007, and MARCO/DARPA GSRC:PAS.

DAC 2005, June 13–17, 2005, Anaheim, California, USA

NoC research, and the first NoC designs [5, 9] proposed the use of simplistic routers, with deterministic routing algorithms. Gradually researchers have explored multiple router implementations, and ongoing research such as [12, 2, 7, 3] explores implementations where pipelined router architectures utilize virtual channels and arbitration schemes to achieve Quality of Service (QoS) and high-bandwidth on-chip communication. Mullins, et. al. [14] propose a single stage router with a doubly speculative pipeline to minimize deterministic routing latency. Among the disadvantages however of such approach, is an increased contention probability for the crossbar switch, given the single-stage switching. The results given in [14] do not seem to take contention into consideration. Additionally, emphasis on the intra-router delay does not imply adaptivity to the network congestion. As such, a better approach should combine both low intra-routing latency, and adaptivity to the network traffic.

Under non-uniform traffic, or application-specific (i.e. real time multimedia) traffic, deterministic routing might not be able to react to congestion due to network bursts, and consequently results in an increase in network delay. Adaptive routing algorithms employed in traditional networks as a solution to congestion avoidance are more suitable for NoC implementations. As a result, adaptive routing algorithms have recently surfaced for NoC platforms. Such examples include thermal aware routing [18], where hotspots are avoided by using a minimal-path adaptive routing function, and a mixed-mode router architecture [11] which combines both adaptive and deterministic modules, and employs a congestion control mechanism. Both these adaptive schemes do not use up-to-date congestion information to make the routing decision. A preferred method is to use real time congestion information about the destination node, and concurrently utilize adaptive routing to handle fluctuation. A disadvantage, however, that adaptive routing algorithms might suffer is the increased hardware complexity, and possibly higher power consumption. Energy consumption is a primary design constraint, and power driven router design and power models for NoC platforms, have been investigated in [19, 8]. Energy consumption depends on the number of hops a packet travels prior to reaching its destination, We believe that by reducing the overall network latency and increasing the throughput, the minor energy penalty paid when migrating from deterministic to adaptive routing is nullified by the performance of adaptive routing in non-uniform traffic. The motivation for our work, therefore, focuses on supporting adaptivity, while maintaining low latency and low energy consumption.

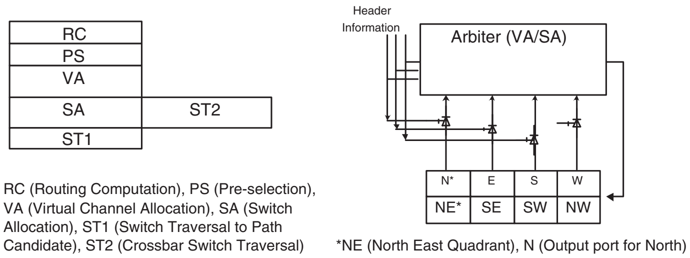
3. PROPOSED ROUTER MODEL

The NoC latency impacts the performance of many on-chip applications. Minimization of message latency by optimizing the intra-node delay and utilizing organized wiring layout with regular topologies has been targeted in NoC designs. The proposed router, designed with this objective, consists of a two-stage pipelined model with look ahead routing and speculative path selection [6, 16]. In this section, we present a customized router architecture that can support deterministic, and adaptive routing in 2-D mesh and torus on-chip networks.

3.1 Proposed Router Architecture

A typical state-of-the-art, wormhole-switched, virtual channel (VC) flow control router consists of four major modules: routing control (RC), VC allocation (VA), switch allocation (SA), and switch transfer (ST). In addition, it may have an extra stage at each input port for buffering and synchronization of arriving flits. Pipelined router architectures typically arrange each module as a pipeline stage. It is possible to reduce the critical path latency by reducing the pipelined stages through look-ahead routing [14]. Figure 1(a) illustrates the logical modules of our two-stage pipelined router incorporating look ahead routing and speculative allocation. The first stage of the router performs look ahead routing decision for the next hop, pre-selection of an optimal channel for the incoming packet (header), VA and SA in parallel. The actual flit transfer is essentially split in two ”stages”, the preliminary semi-switch traversal through the VC selection (ST1) and the decomposed crossbar traversal (ST2). In contrast to the prior router architectures, the proposed model incorporates a pre-selection (PS) unit as shown in Figure 1. The PS unit uses the current switch state and network status information to decide a physical channel (PC) from among the possible paths, computed during the previous stage look-ahead decision. Thus, when a header flit arrives at a router (stage *i*), the RC unit decides a possible set of output PCs for the next hop (*i+1*). The possible paths depend on the destination tag and the routing algorithm employed. Instead of using a routing table for path selection, we use a hardware control that computes a 3-bit direction vector, called Virtual Channel ID (VCID), (based on the four destination quadrants (NE, NW, SE, and SW) and four directions (N, E, S and W)), which can be sent along with the header, for deciding an optimal path using the PS unit in the next hop. The VA and SA are then performed concurrently for the selected output, before the the transfer of flits across the crossbar.

The detailed design of the router is shown in Figure 2. We describe its functionality with respect to a 2-D mesh or torus. The router has five inputs, marked for convenience inputs from the four directions and one from the local PE. It has four sets of VCs, called path sets; one set for possible traversal in each of the four quadrants; NE, SE, NW, SW. Each path set has three groups of VCs to hold flits from possible directions from the previous router.



RC (Routing Computation), PS (Pre-selection), VA (Virtual Channel Allocation), SA (Switch Allocation), ST1 (Switch Traversal to Path Candidate), ST2 (Crossbar Switch Traversal)

*NE (North East Quadrant), N (Output port for North)

(a) Router Pipeline

(b) Direction Vector Table in PS

Figure 1: The Two-Stage Pipelined Router

This grouping is customized for a 2-D interconnect. For example, a flit will traverse in the NE quadrant only if is coming from the west, south or the PE itself. Similarly, it will traverse the NW quadrant if the entry point is from south, east or the local PE. Thus, based on this grouping, we have 3VCs per PC. Note that it is possible to provide more VCs per group if there is adequate on-chip buffering capacity, and the VA selects one of the VCs in a group.

The MUX in each path set selects one of the VCs for crossbar arbitration. In the mean time, the PS generates the pre-selection enable signals to the arbiter based on the credit update, and congestion status information of the neighboring routers. The arbiter, in turn, handles the crossbar allocation.

Another novelty of this router is that since we are using topology tailored routing, we use a 4×4 decomposed crossbar with half the connections of a full crossbar as shown in Figure 2. Usually a 5×5 crossbar is used for 2-D networks with one of the ports assigned

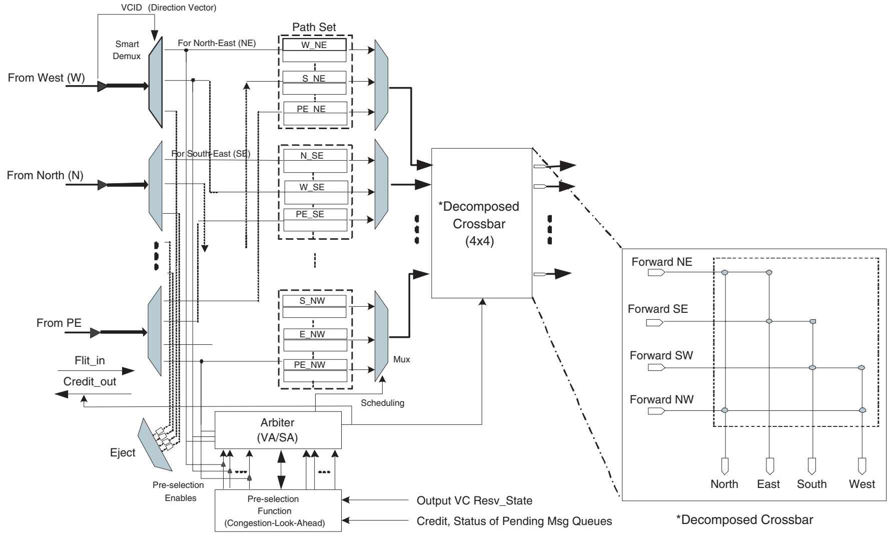


Figure 2: Proposed Router Architecture

to the local PE. In our architecture, a flit destined for the local PE, does not traverse the crossbar. Utilizing the look-ahead routing information, it is ejected after the DEMUX. Thus, the flits save two cycles at the destination node by avoiding the switch allocation and switch traversal. This provides a significant advantage for nearest-neighbor traffic, and can take advantage of NoC mapping which places frequently communicating PEs close to each other [13].

The decomposed crossbar offers two advantages for on-chip design. First, it needs less silicon and second, it should consume less energy compared to a full crossbar. In addition, because of less number of connections, the output contention probability is reduced. This, in turn, should help in reducing the mis-speculation of the VA and SA stages.

3.2 Pre-selection Function

The pre-selection function, as outlined earlier, is responsible for selecting the channel for a packet. It maintains a direction vector table for the path selection and works one cycle ahead of the flit arrival. The direction vector table, as shown in Figure 1 (b), selects the optimal path for each of the four quadrant path sets. As an example, if a flit is in the W_NE VC, the PS decides whether it will use the N or E direction and inputs this enable signal to the arbiter. The PS logic uses the congestion look-ahead information and crossbar status to determine the best path, and also immediately updates the credit priorities into the arbitration. The congestion look-ahead scheme provides adaptive routing decisions with the best VC (or set of VCs) and path selection from the corresponding candidates based on the congestion information of neighboring nodes.

3.3 Adaptive Routing Algorithm

An adaptive routing algorithm either needs a routing table or hardware logic to provide alternate paths. Our proposed router can support deadlock-free fully adaptive routing for 2-D mesh and torus networks using hardware logic. Note that a flit can use any minimal path in one of the four quadrants, as discussed in the router model in Section 3.1. The final selection of an optimal channel is done by the PS module. It can be easily proved that the adaptive routing is deadlock free for a 2-D mesh because the four path sets, shown in Figure 2 are not involved in cyclic dependency. Whenever two flits from two quadrants are selected to traverse in the same direction (for example a flit from NE and a flit from SE contend for an east channel), they use separate VC in the NE and SE path sets, thereby avoiding channel dependency.

For a 2-D torus network, we use an additional VC to support deterministic and fully adaptive routing similarly to other adaptive routing algorithms in torus. One of the VCs in each subset is used for crossing dimensions in ascending order, and the other VC is used for descending order, which is possible for wrap-around links in a torus. Note that prior adaptive routing schemes need at least 3 VCs per PC to support adaptivity.

Although an adaptive routing algorithm helps in achieving better performance specifically under non-uniform traffic patterns, the underlying path selection function has a direct impact on performance. Most adaptive routing algorithms take little account of temporal traffic variation and other possible traffic interferences.

The proposed adaptive routing algorithm, utilizes look-ahead congestion detection for the next router's output links using credit-based system. The PS module keeps track of credits for each neighboring router on the candidate paths. Neighboring routers send credits indicating congestion (VC state), with the amount of free buffer space available for each VC. In addition, time between successive transmissions to neighboring routers is kept minimal due to the low latency of our architecture, and consequently our proposed architecture captures congestion traffic in short spurts and reacts accordingly. This helps in capturing congestion fluctuation and picking the right channel from amongst the available candidate channels.

3.4 Contention Probabilities

To illustrate the benefits of our decomposed crossbar architecture, we compare the contention probabilities of a full crossbar versus our decomposed crossbar design. We derive the contention probability as a function of the offered load λ, where λ is the probability that a flit arrives at an arbitrary time slot for each input port [15], for a generic (NxN) full crossbar and an (N-1xN-1) decomposed crossbar, assuming uniform distribution. Let λ_f, λ_{PE} denote load directed to an output port and ejection channel respectively. Let P_s be the probability that a port is busy serving a request. Let Q_n be the probability that an input port has n flits for a specific output port at the time of request. Q_n can be solved by discrete time Markov-chain (DTMC) as follows:

$$Q_0 = 1 - \frac{\lambda_f}{P_s}, Q_{0\_PE} = 1 - \frac{\lambda_{PE}}{P_s}$$

The contention probabilities, P_{con_full} for the full crossbar and P_{con_dec} the decomposed crossbar, can be formulated as:

Full Crossbar:

$$P_{con\_full} = \sum_{j=2}^{N-1} \frac{1}{j} \left(\begin{array}{c} N-1 \\ j \end{array} \right) (1-Q_0)^j Q_0^{N-1-j} Q_{0\_PE}$$
$$+ \sum_{j=2}^{N-1} \frac{1}{j} \left(\begin{array}{c} N-1 \\ j-1 \end{array} \right) (1-Q_0)^{j-1} Q_0^{N-j} (1-Q_{0\_PE})$$

Decomposed Crossbar:

$$P_{con\_dec} = \sum_{j=2}^{\frac{N-1}{2}} \frac{1}{j} \left(\begin{array}{c} \frac{N-1}{2} \\ j \end{array} \right) (1-Q_0)^j Q_0^{\frac{N-1}{2}-j}$$

The contention probability is shown in Figure 3. It is evident that the decomposed crossbar outperforms the full crossbar. The tree structure of the decomposed crossbar (see Figure 2) provides flits with pre-selected path sets towards the destination, thus reducing contention.

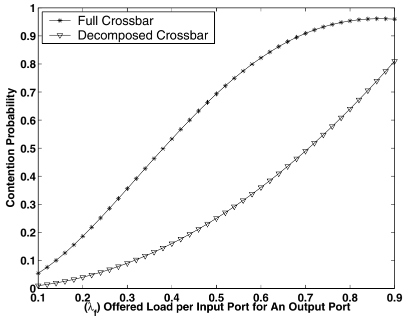


Figure 3: Comparison of Contention Probability

4. PERFORMANCE EVALUATION

In this section, we present simulation-based performance evaluation of our architecture in terms of network latency and energy consumption, under various traffic patterns. We describe our experimental methodology, and detail the procedure followed in evaluation of our proposed architecture.

4.1 NoC Simulator

To evaluate our architecture, we have developed an architectural level cycle-accurate NoC simulator using C, and we used simulation models for various traffic patterns. The simulator inputs are parameterizable allowing us to specify parameters such as network size and topology, routing algorithm, the number of VCs per PC, buffer depth, PE injection rate, flit size, and number of flits per packet. Our simulator breaks down the router architecture into individual components, modeling each component separately and combining the models in unison to model the entire router and subsequently the entire network architecture. We measure network latency from the time a packet is created to the network interface of the PE, to the time its last flit arrives at the destination PE, including network queuing time. We assume that link propagation happens within a single clock cycle. In addition to the network-specific parameters, our simulator accepts hardware parameters such as power consumption (leakage and dynamic) and clock period for each component, and utilizes these parameters to provide results regarding the network energy consumption.

4.2 Energy Estimation

Energy consumption for on-chip networks consumes a large chunk of the total chip energy. While no absolute numbers are given, researchers [19] estimate the NoC energy consumption to approximately 45% of the total chip power. The generic estimation in [19] relates the energy consumption for each packet to the number of hop traversals per flit per packet, times the energy consumed for each flit per router. We took this estimation a step further, decomposing the router into individual components which we laid out in 70nm technology, using a power supply voltage of 1V, and 200MHz clock frequency. We used the Berkeley Predictive Technology model [1], and measured the average dynamic power using 50% switching activity on the inputs. Additionally, we measured the leakage power of each component when no input activity is recorded. We imported these values into our architectural level cycle-accurate NoC simulator and simulated all individual components in unison to estimate both dynamic and leakage power in routing a flit. Doing so we were able to identify the individual utilization rates of each component, thus accurately modeling the leakage and dynamic power consumption.

A particular detail which we emphasized was to distinguish between each flit type (header, data, tail) so as to correctly model the power consumed by each flit type. For instance, a header flit will be accessed by all components, where as a data flit will simply stay in the buffer until propagated to the output port.

4.3 Experimental Setup

We evaluated our architecture using two 8x8 networks, one using 2D torus and the other using 2D mesh topologies. We use 4 flits per packet and wormhole routing, with 4-flit deep buffers per VC. We use 3 VCs per PC for 2D mesh and 6VCs per PC for 2D torus (as required by our algorithm). Every simulation initiates a warm-up phase of 20,000 packets. Thereafter, we inject 1,200,000 packets and simulate our architecture until all the packets are received at the destination nodes. We log the simulation results and gather network performance metrics based on our simulations for various traffic patterns. For packet injection rates, we used three workload traces: self similar web traffic, uniform and MPEG-2 video multimedia traces. We simulated each trace using four different traffic permutations: normal-random, transpose, tornado and bit-complement [6].

4.4 Simulation Results

First we compare our proposed 2-stage router architecture to a 3-stage pipelined router, in order to evaluate the average network latency. In the 3-stage router, we implement both a dimension-order (DOR) algorithm, as well as our adaptive routing algorithm. We separate the crossbar arbitration from the routing decision stage, resulting in three stages in the router pipeline when using our adaptive algorithm. A direct effect of this is a full crossbar switch instead of the decomposed crossbar our 2-stage architecture employs. We compare both routers (2-stage vs. 3-stage) using both DOR and our adaptive algorithm, and show the results in Figure 6. We partition the average latency in three parts: contention free transfer latency, blocking time, and queuing delay in source node. In both zero load latency, and congestion avoidance, the 2-stage router performs better than the 3 stage. Figure 6 shows also the impact of adaptivity, indicating poor performance when compared to DOR under uniform random traffic. Spatial traffic evenness benefits load balancing in DOR, even if we improve adaptivity. In the presence of non-uniform traffic however, the adaptive algorithm outperforms the DOR algorithm as shown in Figure 7 for transpose (TP), self-similar(SS) and Multimedia(MM) traffic patterns. As bursty traffic is more representative of a practical on-chip communication environment, we expect our adaptive algorithm to be very useful.

An additional NoC property is spatial locality. PEs that communicate often with each other, are usually placed in proximity to each other. In our architecture, when a packet arrives its destination router, instead of traversing the crossbar to the PE, the VC selection mechanism ejects the packet into the PE immediately, bypassing the router traversal as shown in Figure 4. We also show a 3-stage router without the bypass modification in Figure 4, to illustrate the difference. Even though in Figure 4 we assume no blocking inside the router, as the injection rate increases, the blocking latency inside the router increases as well and the bypassing technique results in significant benefits. We simulated the 2-stage and 3-stage router architectures for nearest-neighbor traffic in order to investigate the latency, which we show in Figure 5, illustrating that early ejection and intra-router latency reduction jointly decrease the overall network latency. The increased hardware complexity in implementing the adaptive routing algorithm results in a higher power consumption than when implementing the simpler, DOR algorithm. However, the latency reduction when using adaptive algorithm for non-uniform, more practical traffic environments, more than compensates for this power increase, reducing the overall energy. Figure 8(a)shows the energy consumption per packet for uniform and transpose traffic, when comparing adaptive routing vs. DOR. In Figure 8(b), we also see that adaptive routing is beneficial when we consider the Energy-Delay Product for non-uniform traffic.

5. CONCLUSIONS

We present a low-latency router architecture suitable for on-chip networks which utilizes an adaptive routing algorithm. The architecture uses a novel path selection scheme resulting in a 2-stage switching operation with a decomposed crossbar switch. We evaluate our architecture using various traffic patterns, and we show that under non-uniform traffic our architecture reduces the overall network latency by a significant factor. Due to the low latency of our architecture, congestion information about each router's neighboring routers is swiftly updated, resulting in almost real-time updates. We also show that the energy consumption when using adaptive routing is also reduced due to the reduction in the network latency. Results also show a better performance in the presence of nearest-neighbor traffic, a particular benefit for NoC designs which emphasize spatial locality. In the future we are planning to explore architectural optimizations in reducing the latency to a single stage thus obtaining better congestion information, and introduce more features in our adaptive algorithm implementation. In particular, we can expand the lookahead information for the destination router from VC state only, to include the link state and other parameters such as hotspots and permanent faults in the links and routers in order to provide hotspot avoidance and fault tolerance.

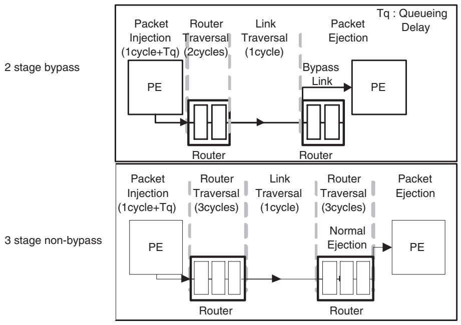


Figure 4: Latency Analysis of Nearest-Neighbor Traffic

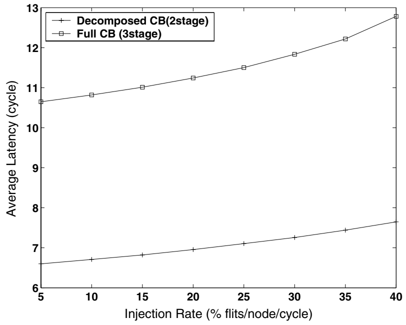


Figure 5: Comparison of Nearest-Neighbor Traffic

6. REFERENCES

[1] Berkeley predictive technology model. http://www-device.eecs.berkeley.edu/ ptm/.

[2] The nostrum backbone project. http://www.imit.kth.se/info/FOFU/Nostrum/.

[3] Stanford-bologna netchip project. http://akebono.stanford.edu/users/nanni/research/net/netchip/.

[4] L. Benini and G. D. Micheli. Networks on Chips: A New SoC Paradigm. *IEEE Computer*, 35(1):70–78, 2002.

[5] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proceedings of the 38th Design Automation Conference*, June 2001.

[6] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[7] J. Dielissen and et. al. Concepts and implementation of the Philips network-on-chip. In *IP-Based SOC Design*, Nov. 2003.

[8] N. Eisley and L.-S. Peh. High-level power analysis for on-chip networks. In *Proceedings of CASES*, pages 104–115. ACM Press, 2004.

[9] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections. In *Proc. of DATE*, pages 250–256. ACM Press, 2000.

[10] M. Horowitz, R. Ho, and K. Mai. The future of wires. In *Proc. of SRC Conference*, 1999.

[11] J. Hu and R. Marculescu. DyAD - Smart Routing for Networks-on-Chip. In *In Proceedings of the 41'st ACM/IEEE DAC*, 2004.

[12] A. Jalabert, S. Murali, L. Benini, and G. D. Micheli. xpipes compiler: a tool for instantiating application specific networks on chip. In *Proc. of DATE*, pages 884–889, 2004.

[13] R. M. Jingcao Hu. Energy-aware mapping for tile-based noc architectures under performance constraints. In *Proc. of ASPDAC*, January 2003.

[14] R. Mullins and et. al. Low-latency virtual-channel routers for on-chip networks. In *Proc. of the 31st ISCA*, page 188. IEEE Computer Society, 2004.

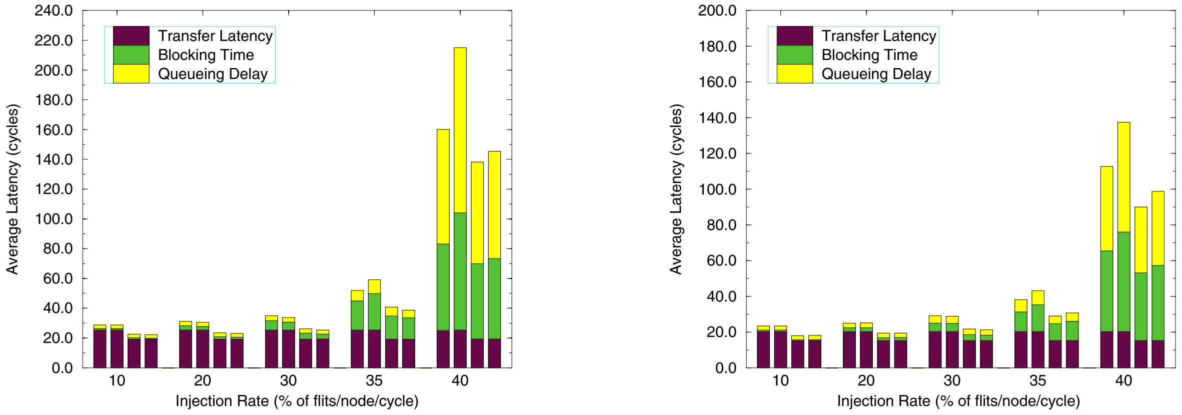
[15] S. Y. Nam and D. K. Sung. Decomposed crossbar switches with multiple input and output buffers. In *Proc. of IEEE GLOBECOM*, 2001.

[16] L.-S. Peh and W. J. Dally. A Delay Model and Speculative Architecture for Pipelined Routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, 2001.

[17] P. Rickert. Problems or opportunities? beyond the 90nm frontier, 2004. ICCAD - Keynote Address.

[18] L. Shang, L.-S. Peh, A. Kumar, and N. K. Jha. Thermal Modeling, Characterization and Management of On-Chip Networks. In *Proc. of the 37th MICRO*, 2004.

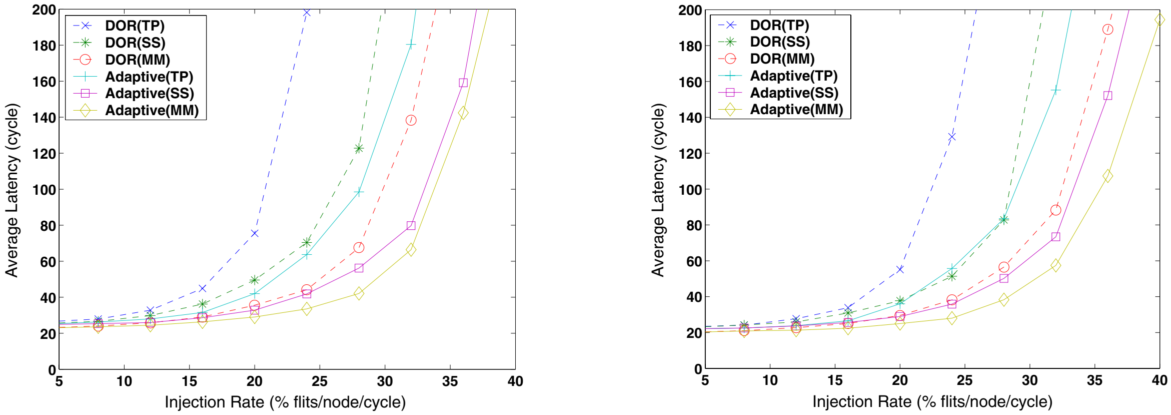
[19] H.-S. Wang, L.-S. Peh, and S. Malik. Power-Driven Design of Router Microarchitectures in On-Chip Networks. In *Proceedings of the 36th MICRO*, November 2003.



(a) 2D Mesh Network

(b) Torus Network

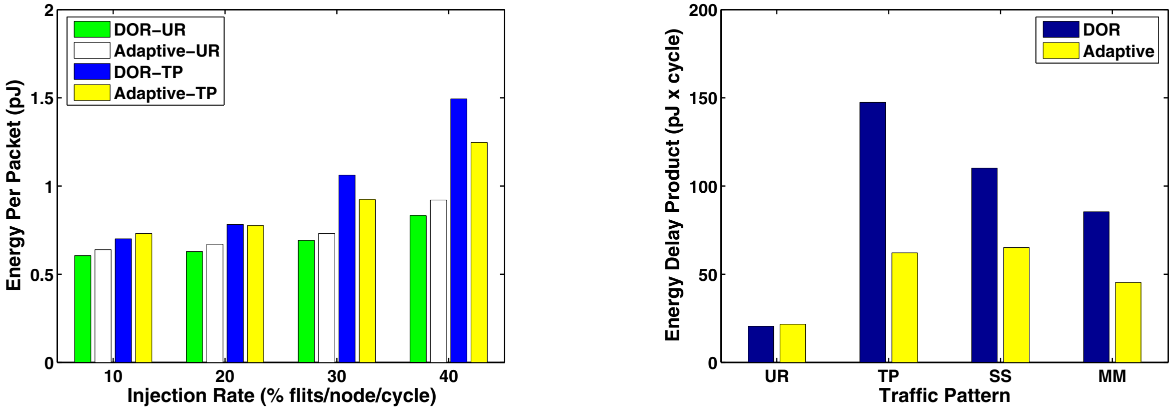
Figure 6: Comparison of 2-stage vs. 3-stage routers utilizing both DOR and adaptive routing algorithms. [For each injection rate, from left to right, columns correspond as follows: 3-Stage DOR, 3-Stage Adaptive, 2-Stage DOR, 2-Stage Adaptive]



(a) 2D Mesh Network

(b) Torus Network

Figure 7: Performance Comparison under Non Uniform Traffic



(a) Energy Consumption in Uniform and Transpose Traffic Patterns

(b) Energy Delay Product

Figure 8: Energy Consumption