Implementation and Analysis of History Based Output Channel Selection Strategies for Adaptive Routers in Mesh NoCs

JOHN JOSE and MADHU MUTYAM, Indian Institute of Technology, Madras

The efficiency and effectiveness of an adaptive router in an NoC based multicore system is evaluated by the performance it achieves under varying inter-core communication traffic. A well designed selection strategy plays an important role in an adaptive router to act upon dynamic traffic variations. The effectiveness of a selection strategy depends on what metric is used to represent congestion, how precisely that metric captures the actual congestion, and how much is the cost involved in capturing the congestion on a real time scale. Congestion is formed over a period of time due to cumulative and chain reaction effects. We propose novel history based selection strategies that could be used with any adaptive, deadlock free, minimal routing in mesh NoCs. Buffer occupancy time and rate of flit flow across reachable ports of neighboring routers in the recent past are captured, propagated and maintained in a cost effective way to compute the selection metric. Experiments results on real and synthetic workloads show that our proposed selection strategies significantly outperform state of the art techniques.

Categories and Subject Descriptors: B.4.3 [**Input/Output and Data-Communications**]: Interconnections (Subsystems); C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms: Design, Performance

Additional Key Words and Phrases: adaptive routing, selection strategy, load balancing, congestion

1. INTRODUCTION

Over the last decade, technological innovations in System-on-chip (SoC) have been focusing on empowering the computational efficiency of computing cores [Lotfi-kamran et al. 2010]. Diminishing feature sizes and shrinking wire widths expose the limitations of traditional bus based interconnection systems. Technology scaling has increased the performance gap between computation efficiency and communication efficiency in multicore chips [Marculescu et al. 2009].

Apart from high speed computing cores, efficient and reliable communication is also essential for achieving high performance in multicore processors. Network-on-chip (NoC) is an emerging paradigm that can efficiently support integration of a massive number of cores on a chip by decoupling the on-chip computation and communication infrastructure, thereby overcoming the scalability issues faced by conventional buses. NoC architectures are well suited for both homogeneous and heterogeneous mutlicore systems. NoCs employ a grid of routers organized across the chip, connected by communication links [Dally and Towles 2001].

In a conventional 2D mesh topology, the processing cores are organized as rectangular tiles as shown in Figure 1. Each core consists of a processing element (out-of-order super-scalar processor), a private L1 cache, and a distributed shared L2 cache bank. The main memory controllers are generally attached to the cores in the corner of the mesh. Each core is attached to a local router that connects the core to other cores via a well structured set of routers and links. Inter-core communication is needed during cache miss, coherence and synchronization transactions.

When a core wants to communicate with another, it creates a packet with the required data or control information. Additional information needed to process and route the packet is organized as a header and is prefixed with the packet. Once the packet is ready, the core injects it to the network via a network interface controller (NIC). Wormhole routing [Dally and Towles 2001] is one of the most commonly used approach in 2D mesh NoCs where each packet is serialized into a sequence of flow control units, called *flits*. Typically a flit size is the width of a physical channel connecting a pair of

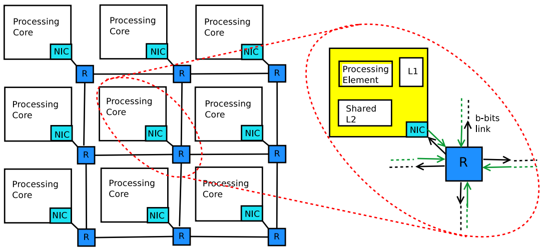

Fig. 1: Interaction of an NoC router with a tile

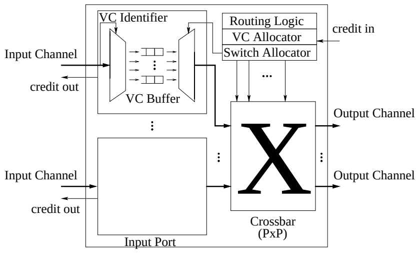

Fig. 2: Conventional baseline VC router

routers. Modest buffering requirements make wormhole routing a best design choice in NoC platform [Schwiebert and Bell 2002].

NoC Routers: An Overview- In a 2D mesh topology, the router attached to a tile consists of five input ports and five output ports; one for each direction and one for the local tile. A block diagram of a VC based router is given in Figure 2. Every input port of a router is associated with a set of virtual channels (VC). Once a flit reaches a router, buffers in the VCs provide storage space for it [Dally 1992]. The control logic embedded in the router performs routing operation, allocation of VC in the next router, and switch arbitration. The credit signals to and from a router carry the availability of buffer space. Buffering within the routers and two way handshaking between routers enable the smooth flow of the packets. Routers are typically pipelined and several speculation or pre-computation techniques are used to reduce the critical path latency [Matsutani et al. 2009; Kim et al. 2005; Ogras and Marculescu 2006].

Routing techniques used in underlying NoC play a vital role in the overall system performance. In deterministic routing, the path from source to destination is determined exclusively by the address of source and destination. Dimension Order Routing (DOR) is one of the most commonly used deadlock free deterministic routing algorithm. Here each packet first travels along the X direction and then along Y direction to reach the destination. Even though DOR is a simple and easy to implement, it is not adaptable to traffic variations [Lotfi-kamran et al. 2010].

Adaptive routing algorithms use information about the state of the network to make routing decisions [Dally and Towles 2003]. In adaptive routing, the path taken by a particular packet depends on the dynamic network conditions. They exploit the possibility of routing packets along alternative paths in order to avoid congested areas. Adaptive routing algorithms can be either minimal that allow only shortest path (in terms of the number of hops) or non-minimal that do not enforce message forwarding through only shortest paths.

Scope of Output Channel Selection Strategy- Once a packet reaches an adaptive router, the route function identifies a set of possible output channels. If the route function returns more than one admissible output channel, an output channel selection strategy is needed. The selection strategy makes use of an appropriate congestion metric for selecting an output channel that allows the packets to be routed through minimally congested nodes. The effectiveness of a selection strategy depends on the metric used to identify congestion and how precisely that metric captures congestion.

Our study on various synthetic traffic patterns on 4×4 mesh network with *Minimal Odd Even* (MOE) routing shows that, on an average, in 27% of cases, the route function returned multiple admissible output channels. For larger meshes the percentage is even higher. So at least for every one in four routing decisions, selection strategy determines which output port is to be selected. This emphasizes the fact that a well formed selection strategy can significantly impact the overall packet latency.

Majority of the existing adaptive routers use the availability of free buffers in downstream nodes as the congestion metric [Kim et al. 2005; Ascia et al. 2008] for channel selection decisions. Our experimental observations show that the real congestion status of a router cannot be fully represented by the count of free buffers alone. We feel that real congestion is not well captured by conventional congestion metrics. There is always a potential research scope for developing a better, cost effective, selection metric in adaptive routers. Hence for building an efficient selection strategy, a proper feedback mechanism that monitors the packet flow in the network is needed. Such mechanism should foresee congestion situations and take necessary steps that facilitate congestion relief. Through this paper we make an effort to address this issue.

The following are our main contributions:

— We propose an extended model of Buffer Occupancy Factor based Adaptive Router (BOFAR) [Jose et al. 2011] for 2D mesh topology that uses the cumulative history of buffer occupancy time of flits in neighboring routers as the congestion metric for output port selection.

— We propose a hybrid selection strategy that incorporates the above mentioned cumulative buffer occupancy time and cumulative flit flow proposed in [Jose et al. 2012].

— On a baseline router with MOE routing, we compare the performance of the proposed selection strategies with other state of the art selection techniques like Neighbors on Path (NOP) [Ascia et al. 2006; 2008], and Fluidity of downstream neighbors (FON) [Lan et al. 2008].

2. RELATED WORK

First generation NoC designs use simple routers employing deterministic routing algorithms. Later the turn models [Glass and Ni 1992] and odd-even routing [Chiu 2000] provide the flexibility of having multiple possible paths between a pair of source and destination. Since they are not using any specific selection strategy, they make a random selection from the available output ports. DyAD smart routing [Hu and Marculescu 2004] switches between odd-even and deterministic routing based on the network congestion conditions. DyAD and LATEX [Azampanah et al. 2012] incorporates a selection strategy based on the free buffer slots in the current and neighboring router.

Proximity Congestion Awareness [Nilsson et al. 2003] uses stress value, defined as the number of flits a switch handles in a cycle, of neighboring switches for channel selection decisions. Congestion Aware Deterministic Routing [Kiasari et al. 2010] estimates the congestion level in the network based on the past flow pattern and computes the optimized routing paths for all traffic flows. This is suited only for systems that generate repetitive traffic patterns. Path Based Randomized Oblivious Minimal Routing [Cho et al. 2009] proposes a load balancing routing scheme by routing packets through a random node in the minimal path.

VC flow control [Dally 1992] and low latency adaptive router [Kim et al. 2005] use the number of free VCs (FVC) in downstream nodes as the congestion metric. Neighbors-on-Path (NOP) [Ascia et al. 2008] strategy explores the free buffer status of reachable two hop away neighbors of the current node. This allows each router to monitor the router buffer status of reachable two-hop neighbors, which helps in detecting potential congestion earlier. But in NOP, channel selection is based on two cycle old free buffer status of two-hop neighbor. This does not guarantee the availability of these free buffers by the time the flit arrives at that node. Since NOP technique acts upon congestion status beyond neighboring nodes, it is more effective than FVC technique.

Fluidity of Neighbors (FON) metric proposed in Congestion Avoidance Scheme [Lan et al. 2008] defines congestion in a path in terms of buffer fluidity. A buffer is said to be fluid when a flit is moving out from it in the current clock cycle. Fluid buffers indicate easy flow of flits. Selection strategy based on power dissipation related to the self and coupling switching activity for each admissible output link is explored in [Salemi et al. 2011]. A congestion aware selection method with special cluster agent and a separate agent feedback network is employed in [Ebrahimi et al. 2011]. In [Lin and Xiang 2010] congestion metric is computed based on the aggregation of historical bandwidth usage and recent bandwidth usage.

HARAQ [Ebrahimi et al. 2012] uses a non-minimal routing with Q-learning based channel selection strategy. It uses a mad-y routing with a static turn control model and a quadrant based congestion cost model for selection decisions. Since the congestion estimate of paths to various nodes in the same quadrant is different, the quadrant based Q-table approach lacks the real-time congestion accuracy. Regional Congestion Awareness (RCA) [Gratz et al. 2008] and Destination Based Adaptive Routing (DBAR) [Sheng et al. 2011] present a comprehensive evaluation and usage of non-local congestion information for improving the dynamic load balancing properties of fully adaptive minimally routed networks. In RCA a composite metric that consists of number of virtual channels, free buffers, and crossbar demand is used.

The commonly used metrics like FVC and NOP are not looking into the real cause for congestion, but they simply act upon count of unassigned VCs. Congestion level of two routers cannot be compared by the present value of free buffers in them. It could be possible that a smooth path can occasionally have few traffic bursts making the router buffers empty in those paths. As per FVC and NOP techniques, such paths are been noted as congested the moment their buffers are occupied. FON metric also acts on instantaneous fluidity value.

If output port selection is based on such instantaneous snapshot based greedy decisions, long term goal of reducing average packet latency may not be met. Considering this, we have proposed an adaptive router, TRACKER [Jose et al. 2012] that employs a cumulative flit flow rate based output channel selection which reduces the possibility of congestion formation by distributing the packets evenly across possible available minimal paths. Techniques that look into real cause of congestion and taking remedial steps that alleviate the situation only will bring down the average packet latency.

3. HISTORY BASED SELECTION STRATEGIES

We propose an adaptive router, BOFAR that provides congestion relief by reducing flit flow to congested nodes. To facilitate this it employs a cumulative buffer occupancy time based output channel selection. It is a modified energy efficient model of the primitive BOFAR design [Jose et al. 2011]. We also propose a Hybrid Adaptive Router (HAR) that effectively combines the history of buffer occupancy time of flits used in the modified BOFAR and history of flit flow rate used in TRACKER [Jose et al. 2012] to form a novel selection strategy. HAR is able to provide the right combination of congestion relief and congestion avoidance, thereby providing the best design choice for multicore processors operating on applications with reasonably high network traffic.

The working principle of BOFAR, TRACKER, and HAR are similar. We attach few additional circuitry with every output port of a conventional VC-router except the ejection port. We capture the statistics about flits moving out through various output ports, encode into a meaningful *feedback metric*, and send it to all neighboring routers. This feedback updation happens in every cycle through a special control network. Every router forwards *feedback metric* of three of its output ports to each of its neighbor. For instance, *feedback metric* of east, north, and west ports of a router are forwarded to the neighbor located in its south. Every router captures these propagated *feedback metrics* from its neighbors, computes the history based *congestion metric* for each output port of the neighboring routers, and stores it in the respective *Status Registers (SR)*.

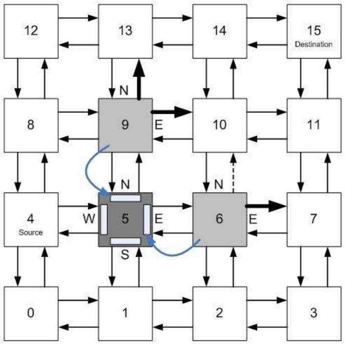

Fig. 3: Illustration of congestion metric exchange and selection function.

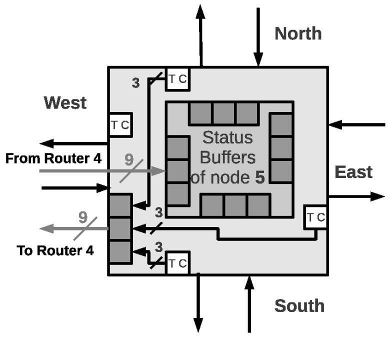

Fig. 4: Architecture of selection strategy of BOFAR

We illustrate the working of history based selection strategy with the following example. Consider a 2D mesh given in Figure 3, where each node represents an NoC router. Assume a flit F, sourced at node 4 and destined to node 15 reaches node 5. The MOE route function chooses east port (link to node 6) and north port (link to node 9) as possible output channels. In the mean time, the *feedback metrics* (shown by the blue curved lines) captured from node 9 and node 6 update the *congestion metric* stored in the respective SRs of node 5 (shown as buffers in the node 5). The *feedback metric* from node 9 to node 5 contains *feedback metric* through the east, north, and west output ports of node 9. Similarly the *feedback metric* from node 6 to node 5 contains *feedback metric* through the north, west, and south output ports of node 6. Each of these *feedback metrics* is connected to the unique SR in node 5.

As per MOE routing, a flit from node 5 destined to node 15, upon reaching node 9 has two possible output links. They are the north and east ports (shown by thick arrows) of node 9. Hence we consider the *selection parameter* for F along the north output channel of node 5 is the average of the *congestion metric* through east port and north port of node 9. Since we have only one possible path for F upon reaching node 6 (the north port of node 6 is not a reachable port in MOE routing), the *selection parameter* for F along east output channel of node 5 is *congestion metric* along east port of node 6 itself. The output channel of node 5 with smaller *selection parameter* value is given higher priority at the time of output port selection for flit F. The following subsections will describe in detail how the *feedback metric* and *congestion metric* are computed in our proposed selection strategies.

3.1. Selection Logic Architecture for BOFAR

In a 2-cycle VC-baseline router, if two or more flits residing in the buffers compete for the same output port, only one can win and the others have to stay back in the buffers and try their chance in the next cycle. Similarly flits stay back in a router buffer due to unavailability of buffers (lack of credit) in the downstream router. We consider both these situations as indications of local congestion. Since congestion increases the buffer occupancy time of the flits, we quantify the intensity of congestion in terms of the number of extra cycles flits stay back in the router buffers.

An architectural block diagram of the selection strategy of BOFAR is given in Figure 4. The *Tracking Circuit* (TC) attached to each output port captures the buffer occupancy time of every outgoing flit. Buffer occupancy time of a flit is considered as the total time (in cycles) a flit spent in the router buffer. The 3-bit buffer occupancy time is the *feedback metric* of the respective port. If no flit is moving out through a port in the current cycle, then the *feedback metric* is zero. Every router captures and propagates this *feedback metric* to its neighboring routers and they compute the *Cumulative Buffer Occupancy Count* (CBOC).

In Figure 4, we can see that the 3-bit *feedback metric* of the north, east and south output port of node 5 is combined and is forwarded to the west neighbor (node 4). Similar operation is done for all ports and respective neighbors. At the same time, node 5 receives a 9 bit feedback metric from node 4 (3-bit *feedback metric* of the south, west, and north output port of node 4) and updates the CBOC values kept in the west port SRs of node 5.

In every cycle, CBOC is updated by adding the corresponding *feedback metric*. At the end of *Refresh Interval* (RI), we perform a weight reduction of CBOC by multiplying with a weight factor α = 0.125. This is same as right shifting the present SR contents thrice. By this operation, the captured CBOC values are fairly accounted, the relative history of CBOC is still preserved, and the overflow of SR is prevented. The RI is chosen in such a way that the CBOC value is sufficient to distinguish the CBOC of the two candidate paths. We consider a bit wide SR and 16 for RI.

Let us see how CBOC based output channel selection reduces congestion. Since congestion at any port in a router increases the buffer occupancy of flits moving out through that port, the CBOC of the respective port will be high. Since the output channel selection of flits arriving at the neighbors of the congested node is based on this CBOC, a high CBOC prevents further flit flow to the congested node. This reduces the traffic from neighbors to a congested node. When the flow of flits through a congested node is reduced for sometime, blocked flits make forward movement. Thus selection strategy using CBOC provides congestion relief.

3.2. Selection Logic Architecture for TRACKER

Our previous work TRACKER [Jose et al. 2012] has proved that a selection strategy based on history of flit flow rate on neighbor router ports can effectively balance the load in the network by evenly distributing the packets along admissible minimal paths. By its unique, cost effective feedback system, TRACKER regulates flit flow in the network. Every router keeps track of flow of flits through all its ports and exchanges this instantaneous flit flow information to its neighbors. Routers make use of these flit flow estimates to compute *Cumulative Flit Count* (CFC).

The working principle of TRACKER is similar to that of BOFAR. Instead of a TC, here a flip flop (FF) is attached to each output port. At the end of every clock cycle, the FF is updated based on whether a flit flows through that port in the current cycle. Here we have a 1 bit FF value as the *feedback metric*. Hence between a pair of routers we need only a 3 bit control network. The (CFC) is computed exactly in the same way like CBOC computation. We consider α value as 0.25, SR as 4 bit wide and RI as 16 cycles. Low CFC represents the link which is less used in the past and is assigned the highest priority for the output port selection.

We now present the intuition for considering CFC as a meaningful congestion metric. One of the reasons of congestion formation is the irregular traffic flow. Consider Figure 3. Assume a flit F, currently at node 5 wants to move to node 15 through a minimal path. One path is through nodes 9, 13, and 14. Another option is through nodes 9, 10, and 11. Yet another option is through nodes 6, 7, and 11. From node 5, if we want to select the minimally congested path, we need to obatin the congestion levels of each of the nodes in the available paths. Obtaining the real time congestion information of nodes 14 and 11 (more than 2-hop away neighbors) in node 5 is difficult. One option is to have an estimate of contention level at nodes 10, 13 and 7 (2-hop away neighbor). One of the major factor that affects the contention of these nodes is the count of flits moving in through the input links of these nodes. So if we get the flit flow estimate through the output ports of nodes that are connected to nodes 10, 13 and 7, (in this case, the east port of node 9, north port of node 9, and east port of node 6, respectively) it will be an indirect indication of the contention level.

The next hop for flit F residing at node 5 should be either node 9 (northward) or node 6 (eastward). So we make channel selection decisions by considering the past flit flow rate through the possible reachable output ports of node 9 and node 6. Heavy flow of flits through a port indicates that the contention level associated neighbor of the port is high. This is derived from the fact that more number of flits reaching a router in a given time frame can increases port contention and delay in forwarding flits. CFC is an indication of the level of contention of a neighbor router port. Higher the CFC, higher the number of flits passed through that router output port in the past. So if we have a choice in selecting an output port, always opt for a router which has minimum CFC.

3.3. Selection Logic Architecture for Hybrid Adaptive Router (HAR)

BOFAR acts only after congestion formation and then it provides congestion relief by flit flow regulation to high CBOC channels. It is not taking any remedial steps to prevent irregular flow. At the same time, TRACKER regulates the flit flow evenly across available links. But congestion can still occur even though flit flow is regulated. It could be due to traffic contributions from other end of the network which could not be captured from one end. A feedback system that checks the existence of congestion is not there in TRACKER. If flits are not moving through a channel (due to congestion in downstream routers or lack of credits), then its CFC values will be low. But unfortunately TRACKER prefers such low CFC channels which will worsen the situation. So if CBOC based checking is also there along with flit flow balancing with CFC, it will provide better performance. Our HAR realizes such a combination.

HAR is an adaptive router that employs a novel selection strategy which uses CBOC and CFC. So the TC and the FF specified in BOFAR and TRACKER, respectively are attached to each output port. Between a pair of routers, we keep two control networks: a 3 bit flit flow feedback channel that carries the FF values and 9 bit buffer occupancy feedback channel that carries the buffer occupancy time. We have three 10 bit SRs attached to each port of a router. Each 10 bit SR is divided into two segments: a 6 bit SR that store the CBOC and a 4 bit SR that stores the CFC. Computation of CFC and CBOC is done as mentioned earlier. We attach a comparison logic to compare CBOC and CFC of candidate paths in making a channel selection.

The maximum value of CBOC is 63 and that of CFC is 15 since the SR size is 6 bit and 4 bit, respectively. Consider that, for a given flit F, let A and B be the admissible output channels as per the MOE routing. If both CBOC and CFC of A are less than that of B, then channel A is selected. When the difference in CBOC of A and B is greater than 15 (25% of the maximum value), then that channel with lower CBOC is selected. On the other hand if the difference in CBOC of A and B is less than or equal to 15, then (1) channel with lower CFC is selected if the difference in CFC is greater than 4, (2) channel with lower CBOC is selected if the difference in CFC is greater than 4. This algorithm gives preference to congestion relief (due to priority given to CBOC over CFC). For admissible channels without much difference in CBOC (both channel have almost same level of buffer occupancy), HAR evenly distributes packets among the available minimal paths by employing CFC as the *selection parameter*. Whereas if the level of buffer occupancy time of admissible channels is too varying, then HAR first regulate flit flow to bring down this wide difference in CBOC. In this way HAR effectively handles congestion relief and at the same time provide flit flow regulation to reduce future congestion formation.

Our techniques are not affected by the presence or absence of available VCs on the next router. If the neighbor along the port chosen by the selection strategy is not having any free VCs, the flit stays back in the current router in the present clock cycle and

Table I: Percentage of different network injection intensity applications in various SPEC 2006 benchmark mixes.

Benchmark Mix	M1	M2	M3	M4	M5	M6	M7
% of Low	100	0	0	50	0	50	31
% of Medium	0	100	0	0	50	50	31
% of High	0	0	100	50	50	0	38

Table II: Representative benchmarks (SPEC 2006 suite) for each MPKI set.

Low MPKI	Medium MPKI	High MPKI
calculix	*bwaves*	*hmmer*
gromacs	*gcc*	*leslie3d*
h264ref	*bzip2*	*lbm*
gobmk	*gamess*	*mcf*

checks the VC state in the next cycle. If we choose the selection strategy for neighbors with available VCs we may be able to avoid an extra cycle delay in current router. But results show that such greedy local decisions increase the overall packet latency.

4. EXPERIMENTAL ANALYSIS

We compare our history based selection strategies with state of the art baseline selection metrics: FON [Lan et al. 2008] and NOP [Ascia et al. 2008]. We analyze the results for 16-core (4×4 mesh), 25-core (5×5 mesh), and 36-core (6×6 mesh) networks with synthetic workloads. For real workloads we use 64-core (8×8 mesh). We also examine the sensitivity of the network to various design parameters.

4.1. Experimental Setup and Workloads

We use *Booksim* [Dally and Towles 2003], a cycle accurate NoC simulator, that models a 2-cycle VC-router using MOE routing in sufficient detail and accuracy. On top of this, we implement various selection metrics like FON, NOP, CBOC, CFC, and that of HAR. We use 4, 8, and 12 VCs per router port in 4×4, 5×5, and 6×6 mesh networks, respectively. VC depth is taken as 3-flit buffer. We consider variable length packets with 60% single-flit packets and the rest larger packets of size up to four flits. We use a 128-bit wide flit channel and a 4-bit wide credit channel. We use Multi2sim [Ubal et al. 2007] simulator that accurately models multicore setup with processing units, cache hierarchy and coherence protocols. We model a 64 core CMP set up where each core consists of an out-of-order x86 processing unit with a 64KB, 4-way set-associative, 32B block, private L1 cache and a 512KB, 16-way set associative, 64B block, shared distributed L2 cache.

Application Workloads: We Each core is mapped with a SPEC CPU 2006 [@ONLINE 2011] benchmark application. Based on the MPKI values calculated on a 64KB L1 cache, we classify the benchmarks into *Low* (MPKI less than 5), *Medium* (MPKI between 5 and 25) and *High* (MPKI greater than 25). We construct 50 multiprogrammed workloads, each with 64 single threaded benchmark instances. We categorize these workloads into 7 mixes (M1 to M7) based on the proportion of the network injection intensity *(Low/Medium/High)* of the constituent benchmarks. The workloads are created using various combination of the benchmarks given in Table II and as per the proportion mix given in Table I. After sufficient fast forwarding, we capture the L1 cache miss requests and reply packets that generate network traffic and feed them to the modified Booksim simulator to simulate the network operations.

Synthetic Workloads: In order to show the robustness of our proposed techniques we also evaluate our design using five standard synthetic traffic patterns: *uniform*, *transpose*, *tornado*, *bit-complement*, and *bursty* for 4×4 mesh network. Average packet latency, and link utilization values are collected for each traffic pattern with injection rate varying from zero to saturation. Since *uniform* and *tornado* traffic are defined for 5×5, and 6×6 networks, we analyze the results for larger networks as well for these traffic patterns.

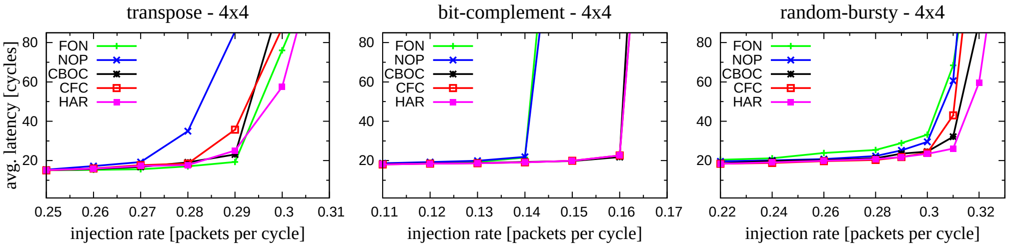

Fig. 5: Average packet latency using various synthetic traffic patterns in 4x4 mesh network.

4.2. Evaluation of Average Packet Latency

Figure 5 contains a set of plots of injection rate vs average packet latency using synthetic traffic patterns in a 4×4 mesh topology. We use adaptive routers with MOE routing algorithm employing various selection parameters. In all the plots we can see that, at low injection rates, all the selection parameters experience the same average packet latency. This is because the network is free from congestion and the selection strategy used is irrelevant. But as the traffic in the network increases, the effect of the selection strategy adopted becomes more predominant. Both CBOC and CFC show relatively lower average packet latency than other baseline selection parameters. CFC shows late saturation due to its effective load balancing approach and regulated flit flow. But CBOC outperforms CFC in random-bursty traffic. Bursty traffic creates packet flooding in certain routers resulting in congestion. CBOC effectively captures this increased buffer occupancy of flits and explore alternate paths. In 4×4 mesh network, HAR shows the best performance in all traffic patterns except tornado. Similarly the plots clearly show that HAR extends the injection rate at which the network saturates.

Figures 6 and 7 contain the injection rate vs average packet latency plots in varying network sizes with uniform and tornado traffic patterns respectively. HAR shows the best performance in uniform traffic (in all network sizes). In tornado traffic, CBOC is not effective and hence HAR also cannot reach its true potential. The effect of CFC is more predominant in tornado traffic as the source and destination pairs are diagonally oriented and the possibility of having multiple paths to spread the traffic is high. It can be seen that CFC has the least pre-saturation latency.

For larger networks CBOC is not performing as good as CFC. The buffer occupancy time of flits in larger networks is comparatively higher than that of smaller networks due to traffic contention as the number of packets injected into the network is high. This leads to early saturation of SRs where CBOC is kept. Once SRs of candidate paths are saturated before RI, it leads to random selection of output channels thereby making selection strategy unproductive.

The relative performance improvement of CFC over CBOC is due to the fact that CBOC targets at congestion relief whereas CFC targets at congestion avoidance. Selection strategy with CBOC operates after congestion formation. Congestion is recognized by long buffer occupancy of flits in routers. It then takes up remedy actions by regulating flit flow to congested nodes. By this time already some packets are penalized and their latencies will be more that average packet latency of the system. On the contrary, CFC operates under congestion prevention mechanism that oversees the possibility of congestion formation occurring through irregular flit flow. Due to this reason, CFC performs better than CBOC in larger networks.

We analyze the average packet latency using real workloads consist of SPEC CPU 2006 benchmark mixes as specified earlier. The percentage reduction in packet latency of various selection metrics with respect to NOP metric is given in Figure 8. We can see considerable reduction in the packet latency using CBOC, CFC and HAR across all

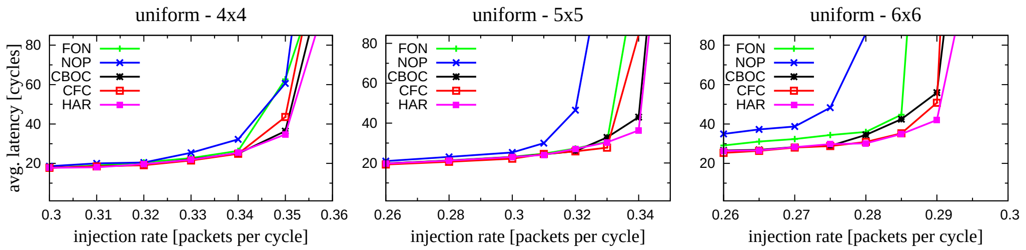

Fig. 6: Average packet latency using uniform traffic pattern in various network size.

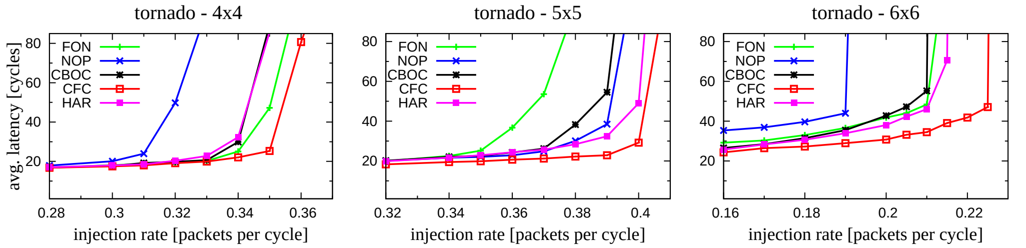

Fig. 7: Average packet latency using tornado traffic pattern in various network sizes.

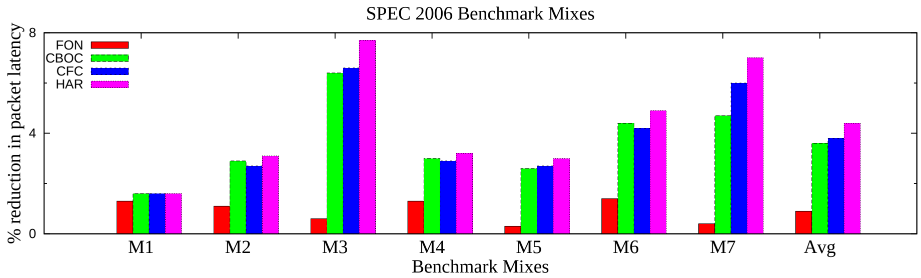

Fig. 8: Percentage reduction in average packet latency with respect to NOP metric using SPEC CPU 2006 benchmark mixes on 8×8 mesh network with 16 VCs per port.

benchmark mixes. For very low traffic injection mix M1, all our techniques are showing the same level of performance. For mixes M2, M4, and M6, CBOC is performing better than CFC, whereas for mixes M3, M5, and M7, CFC is performing better than CBOC. Since HAR combines both congestion relief and congestion avoidance, it is giving better results in all mixes. On an average, FON shows hardly 1% reduction in average packet latency with respect to NOP metric, whereas the % reduction using CBOC, CFC, and HAR are 3.6, 3.8, and 4.4, respectively. The performance improvement for HAR is more in mixes M3 and M5 as they have high MPKI applications and hence more traffic is there in the network. Synthetic traffic results also clearly show that our proposed techniques are performing much better at higher injection rates. Results in mixes M3 and M5 prove that our proposed metrics are excellent design choices for systems with high MPKI applications running on it.

4.3. Evaluation of Link Utilization Fairness

Output channel selection strategy using CFC as the congestion metric is balancing the load in the network. This can be observed by checking the link usage of the network. We compute the total number of flits flowing through each link under a particular injection rate and define *Link Utilization Fairness* (LUF) as the ratio of *average number* *of flits per link* to *standard deviation of flits per link*. Using CFC, flits are evenly distributed across available minimal paths. This reduces the standard deviation of flits per links and hence LUF increases. Increase in LUF leads to uniform wear and tear and hence more life to links.

With uniform traffic in 4×4 mesh network, at normal load, CFC exhibits better LUF by an average increase of 12% and 9% with respect to FON and NOP, respectively. At saturation load the corresponding average increase is 23% and 15%, respectively. The average increase in LUF using CFC in 6×6 mesh is 31% and 22% at normal load and 44% and 25% at saturation load with respect to FON and NOP, respectively. From the results it is clear that at higher loads, when more flits are injected into the network, CFC exhibits higher LUF. This combination of reduced average packet latency along with increased LUF makes TRACKER an excellent choice for an adaptive router design. Since CBOC is not balancing the flit flow like the way CFC does, its LUF is only 73% (in 4×4 network) and 75% (in 5×5 network) of that of CFC.

Since HAR also uses the CFC metric partially, its LUF also is more than FON and NOP (even though not up to the level of LUF of CFC). In 4×4 network, at saturation load, LUF of HAR is 84% of that of CFC whereas it is 89% in 6×6 network. So HAR is able to reduce average packet latency and at the same time increases the LUF.

5. SENSITIVITY ANALYSIS

Both BOFAR and TRACKER use a number of design parameters like weight factor α, Refresh Interval (RI), width of SR and the Compute Interval (CI) of SR updation. These key design parameters play a significant role in the performance achieved by the proposed adaptive routers. In this section we describe how we come up with optimal values for these design parameters. We would also like to have a sensitivity analysis on the router performance with respect to changes in these design parameters.

α is a key design factor in computing the *congestion metric* (CBOC and CFC). It determines how significant is the *congestion metric* residing in SR in the previous RI period. Lower values of α (less than 0.3) assign higher weightage to the *feedback metric* in the current RI period. We conduct experiments with different values of α ranging from 0.0 to 0.5. If α is 0, then it results in total loss of *congestion metric* history at the beginning of a new RI period. In such cases the SR contents cannot represent the real congestion in the initial few cycles of an RI period, leading to poor port selection decisions. Based on this we conclude that, for real time representation of congestion, *feedback metric* updates in the current RI period should be given dominance over cumulative past estimates. But at the same time the relative history of past *congestion metric* should be preserved.

Different synthetic traffic patterns require different values of α for giving the best average packet latency. Since α is a fraction, we need a floating point multiplier to calculate the *congestion metric*. This makes extra overhead in power, area, and computation delay. A complex floating point multiplier can be replaced with a simple shifter circuit if we choose α values like 0.25 or 0.125. Hence to realize the multiplication of α and incrementing of SR, a shifter and incrementer circuit is attached to the SR. We compromise on the average packet latency and saturation throughput for incorporating this change in α to keep router power and area within acceptable limits. Table III shows the average packet latency values for varying α values for uniform traffic pattern in 4×4 mesh. The best case (shown by bold entries) is selected for HAR implementation. For better understanding of the effect of α only injection rates near saturation is shown. For lower packet injection rates (not shown in table) latency values are almost same irrespective of α value.

Next we explore the *Refresh Interval* (RI) of SRs. For a given size of SR, high RI value results in early saturation of SR. On the other hand, for low RI values (less than

Table III: Average packet latency using CFC and CBOC metrics under varying α values in 4×4 mesh with uniform traffic. (PIR is packet injection rate in packets/cycle/router)

PIR	**Average Packet Latency (cycles)**														
	CFC Metric (RI=16 cycles, SR=4 bit)							CBOC Metric (RI=16 cycles, SR=6 bit)							
	α values							α values							
	0.0	0.1	0.15	0.2	**0.25**	0.3	0.5	0.0	0.1	**0.125**	0.15	0.2	0.25	0.3	0.5
0.33	22	24	23	22	**21**	23	22	22	21	**22**	22	21	22	21	22
0.34	31	29	34	26	**25**	25	33	26	25	**25**	28	25	27	25	23
0.35	82	55	61	54	**44**	72	48	63	51	**36**	41	45	42	36	47
0.36	151	186	205	177	**161**	157	153	139	159	**133**	132	100	100	109	117

Table IV: Average packet latency using CFC and CBOC metrics under varying RI values and SR sizes in 4×4 mesh with uniform traffic. In the table, row RI corresponds to Refresh Interval (cycles), row SR corresponds to size of Status Register (bits), column PIR corresponds to packet injection rate (packets/cycle/router).

parameters	**Average Packet Latency (cycles)**																	
	α **= 0.25**									α **= 0.125**						α **= 0.0625**		
RI	8	16	**16**	32	32	64	64	128	16	16	**16**	32	32	64	64	64	128
SR	3	3	**4**	4	5	5	6	6	4	5	**6**	5	6	6	7	8	8
PIR	**CFC Metric**								**CBOC Metric**									
0.33	23	23	**21**	21	23	22	22	21	22	22	**22**	22	21	23	21	23	21
0.34	29	29	**25**	40	31	34	29	29	27	25	**25**	27	27	28	25	25	26
0.35	80	49	**44**	77	34	71	60	65	54	44	**36**	57	37	42	37	40	55
0.36	155	151	**161**	135	156	182	172	180	131	120	**133**	161	154	139	113	129	165

10 cycles), the *congestion metric* kept in the respective SR is insufficient to distinguish between candidate paths for a given flit. If we keep larger SRs to keep it safe from saturation, it incurs more power and area overhead. A fast saturating SR is a representation of heavy contention as well as large buffer occupancy for flits through that port (in the case of BOFAR) or heavy flow of flits through that port (in case of TRACKER). This results in a high value of *congestion metric* thereby making the port associated with this SR a less preferred one. Thus we make sure that no further flits will be moving to this port which is already having a large value of *congestion metric*. All these factors prompt us to choose an RI value large enough to capture the variations of buffer occupancy or flit flow and small enough so that the design overhead of SR is minimum. For TRACKER, BOFAR and HAR we fix the RI to 16 cycles, since performance is not much improving for larger RI values. Table IV shows the average packet latency for varying RI and SR combinations for uniform traffic pattern for pre-saturation and saturation injection rates.

Yet another design parameter is the *compute interval* (CI) of the congestion metric. We examine the results by reducing the frequency of updation of CFC and CBOC to once in 4, 8, 16, and 32 cycles under varying RI. But results show that it leads to degradation of LUF. This is due to the fact that a constant value of CFC for few cycles sets the priority of output ports of a router to be constant within this CI. So during this interval all flits destined to a quadrant move through one path only, which disturbs the principle of LUF. Similarly an infrequent updation of CBOC also sets the priority of output ports of a router to be constant within this CI leading to non-response to congestion updates from neighbors. All the results presented in this paper are based on using CI of 1 cycle, i.e., CFC and CBOC are updated in every cycle.

Table V: Overhead comparison in terms of size of SRs, width of control network (CN) for various selection metrics. Router power and link area overhead values are with respect to traditional VC router with 128-bit flit channel and 4-bit credit channel.

Selection Metric	Size of SRs / port	Width of CN	Router Power Overhead (%)	Link Area Overhead (%)
FON	16	16	5.44	9.03
NOP	12	12	3.57	6.02
CBOC	18	9	5.83	4.53
CFC	12	3	3.59	1.50
HAR	30	12	9.36	6.03

6. HARDWARE OVERHEAD AND POWER DISSIPATION

We use Orion 2.0 [Kahng et al. 2009] for area and power computation of various selection strategies. We assume $65nm$ technology at $1\ GHz$ operating frequency. Using the Predictive Technology Model [Zhao and Cao 2007], we find that signal propagation on the channel takes one cycle at $1\ GHz$ clock frequency. Since all the baseline architectures use some sort of calculation and comparison of congestion values, power overhead of the combinational circuits that do these operations are almost same in all baseline routers. Significant hardware overhead difference is due to width of Control Network and size of Status Register.

All the selection metrics require some control wiring to transmit congestion information. Apart from a data channel a VC router has a 4 bit credit channel for communicating downstream router VC status. The control network consists of the number of additional control wires required to propagate necessary congestion information from a router to its adjacent ones. A comparative study of the SR size and width of CN is given in Table V. All power and area overhead values given in the table are with respect to MOE router with 128-bit flit channel and 4-bit credit channel for VC status updation of downstream router.

We implement the pipelined design of BOFAR, TRACKER, and HAR in Verilog and synthesize using Synopsys Design Compiler with $65nm$ library to obtain the timing latency for each unit in the pipeline stage. In a standard two staged pipelined VC router, the first stage is the routing function and the second stage is for VC allocation, switch arbitration and switch traversal. Our latency analysis on the synthesized Verilog design shows that the second stage determines the critical path of the router pipeline. The proposed selection strategies that are initiated after the routing function is operating well with in the time slack available between the routing function latency and the clock cycle time of the router. Hence we argue that our proposed selection strategies have no effect on the critical path of the router.

7. CONCLUSION

We proposed a set of history based selection strategies that make the best use of link bandwidth and spread traffic across less frequently used links to balance the load. Experimental results showed that our techniques are scalable on larger networks and that cumulative history based selection strategies are excellent design choices for future NoC adaptive router designs.

ACKNOWLEDGMENTS

This work is supported in part by grant from Defence Research and Development Organization (DRDO) under IITM- DRDO MoC.

REFERENCES

Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi, and Davide Patti. 2006. Neighbors-on-Path: A New Selection Strategy for On-Chip Networks. In *Proceedings of the IEEE-ACM-Workshop on Embedded Systems for Real Time Multimedia*. 79–84.

Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi, and Davide Patti. 2008. Implementation and Analysis of a New Selection Strategy for Adaptive Routing in NoC. *IEEE Trans. Comput.* 57, 6 (2008), 809–820.

S. Azampanah, A. Khademzadeh, N. Bagherzadeh, M. Janidarmian, and R. Shojaee. 2012. LATEX: New Selection Policy for Adaptive Routing in Application-Specific NoC. In *Proceedings of Euromicro International Conference on Parallel Distributed and Network-Based Processing*. 515 –519.

G M Chiu. 2000. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems* 11, 7 (July 2000), 729–738.

Myong Hyon Cho, Mieszko Lis, Keun Sup Shim, Michel Kinsy, and Srinivas Devadas. 2009. Path-Based, Randomized, Oblivious, Minimal Routing. In *Proceedings of the 2nd International Workshop on Networks-on-Chip Architectures*. 23–28.

William Dally. 1992. Virtual-Channel Flow Control. *IEEE Transactions on Parallel and Distributed Systems* 3, 2 (March 1992), 194–205.

William Dally and Brian Towles. 2001. Route packets, Not wires: On-chip Interconnection Networks. In *Proceedings of the Design Automation Conference*. 684–689.

William Dally and Brian Towles. 2003. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., USA.

Masoumeh Ebrahimi, Masoud Daneshtalab, Fahimeh Farahnakian, Juha Plosila, Pasi Liljeberg, Maurizio Palesi, and Hannu Tenhunen. 2012. HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks. In *Proceedings of International Symposium on Networks-on-Chip*. 19–26.

M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen. 2011. Agent-based on-chip network using efficient selection method. In *Proceedings of International Conference on VLSI and System-on-Chip*. 284 –289.

Christopher J. Glass and Lionel M. Ni. 1992. The turn model for adaptive routing. In *Proceedings of the Annual International Symposium on Computer Architecture*. 278–287.

Paul Gratz, Boris Grot, and Stephen W. Keckler. 2008. Regional congestion awareness for load balance in Networks-on-chip. In *Proceedings of the International Symposium on High Performance Computer Architecture*. 203–214.

J Hu and R Marculescu. 2004. DyAD: Smart Routing for Networks-on-Chip. In *Proceedings of the Design Automation Conference*. 260–264.

John Jose, K V Mahathi, J Shiva Shankar, D, and Madhu Mutyam. 2012. TRACKER: A Low Overhead Adaptive NoC Router with Load Balancing Selection Strategy. In *Proceedings of International Conference on Computer Aided Design*. 564–568.

John Jose, J Shiva Shankar, K V Mahathi, D Kranthikumar, and Madhu Mutyam. 2011. BOFAR : Buffer Occupancy Factor based Adaptive Router for Mesh NoCs. In *Proceedings of the 4th International Workshop on Networks-on-Chip Architectures*. 23–28.

Andrew B Kahng, Li Bin, Li-Shiuan Peh, and Kambiz Samadi. 2009. Orion 2.0: A fast and accurate NoC power and area model for early stage design space exploration.. In *Proceedings of the Design, Automation and Test in Europe Conference*. 423–429.

Abbas Eslami Kiasari, Axel Jantsch, and Zhonghai Lu. 2010. A Framework for Designing Congestion-Aware Deterministic Routing. In *Proceedings of the 3rd International Workshop on Networks-on-Chip Architectures*. 45–50.

Jongman Kim, Dongkook Park, T Theocharides, and N Vijaykrishnan. 2005. A Low Latency Router Supporting Adaptivity for On-Chip Interconnects. In *Proceedings of the Design Automation Conference*. 559–564.

Ying Cherng Lan, Michael C Chen, Alan P Su, Yu Hen Hu, and Sao Jie Chen. 2008. Fluidity Concept for NoC: A Congestion Avoidance and Relief Routing Scheme. In *Proceedings of Annual IEEE International SoC Conference*. 65–70.

Yanbin Lin and Dong Xiang. 2010. An Effective Congestion-Aware Selection Function for Adaptive Routing in Interconnection Networks. In *Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies*. 156–165.

P Lotfi-kamran, A M Rahmani, M Daneshtalab, A Afzali-kusha, and Z Navabi. 2010. EDXY A low cost congestion-aware routing algorithm for network-on-chips. *Journal of Systems Architecture* 56 (2010), 256–264. Issue 7.

Radu Marculescu, Umit Y. Ogras, Li Shiuan Peh, Natalie Enright Jerger, and Yatin Hoskote. 2009. Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28 (January 2009), 3–21. Issue 1.

Hiroki Matsutani, Michihiro Koibuchi, Hideharu Amano, and Tsutomu Yoshinaga. 2009. Prediction router: Yet another low latency on-chip router architecture. In *Proceedings of International Symposium on High Performance Computer Architecture*. 367–378.

Erland Nilsson, Mikael Millberg, Johnny Oberg, and Round Robin. 2003. Load distribution with the Proximity Congestion Awareness in a Network-on-Chip. In *Proceedings of the Design, Automation and Test in Europe Conference*. 1126–1127.

U Y Ogras and R Marculescu. 2006. Prediction based flow control for network-on-chip traffic. In *Proceedings of the Design Automation Conference*. 839–844.

Standard Performance Evaluation Corporation-2006 @ONLINE. 2011. (2011). http://www.spec.org/

D. Salemi, M. Palesi, and V. Catania. 2011. Power-aware selection policy for networks on chip. In *Proceedings on International Conference on Design Technology of Integrated Systems in Nanoscale Era*. 1 –4.

Loren Schwiebert and Renelius Bell. 2002. Performance Tuning of Adaptive Wormhole Routing through Selection Function Choice. *IEEE Transactions on Parallel and Distributed Systems* 62, 7 (July 2002), 1121–1141.

Ma Sheng, Natalie Enright Jerger, and Zhiying Wang. 2011. DBAR: An Efficient Routing Algorithm to Support Multiple Concurrent Applications in Networks-on-Chip. In *Proceedings of Annual International Symposium on Computer Architecture*. 413–424.

R. Ubal, J. Sahuquillo, S. Petiti, and P. Lopez. 2007. Multi2Sim: A Simulation Framework to Evaluate Multicore-Multithreaded Processors. In *Proceedings of the International Symposium on Computer Architecture and High Performance Computing*. 62–68.

Wei Zhao and Yu Cao. 2007. Predictive technology model for nano-CMOS design exploration. *ACM Journal on Emerging Technologies in Computing Systems* 3 (April 2007), 1–17. Issue 1.