

OE+IOE: A Novel Turn Model Based Fault Tolerant Routing Scheme for Networks-on-Chip

Sudeep Pasricha^{†*}, Yong Zou[†], Dan Connors[‡], Howard Jay Siegel^{†*}

[†] Department of Electrical and Computer Engineering
* Department of Computer Science
Colorado State University, Fort Collins, CO
{sudeep.pasricha, yong.zou, h.j.siegel}@colostate.edu

[‡] Department of Electrical Engineering
University of Colorado Denver, CO
dan.connors@ucdenver.edu

ABSTRACT

Network-on-chip (NoC) communication architectures are increasingly being used today to interconnect cores on chip multiprocessors (CMPs). Permanent faults in NoCs due to fabrication challenges in ultra deep submicron (UDSM) technology nodes and due to wearout have led to an increased emphasis on fault tolerant design techniques. To ensure fault tolerant communication in NoCs, several fault tolerant routing algorithms have been proposed in recent years with the goal of routing flits around faults. A majority of these algorithms are based on the turn model approach due to its simplicity and inherent freedom from deadlock. However, existing turn model based fault tolerant routing algorithms are either too restrictive in the choice of paths that flits can traverse, or are tailored to work efficiently only on very specific fault distribution patterns. In this paper, we propose a novel low overhead fault tolerant routing scheme that combines the odd-even (OE) and inverted odd-even (IOE) turn models to achieve much better fault tolerance than traditional turn model based schemes. The proposed scheme uses replication opportunistically to optimize the balance between energy overhead and arrival rate. Our experimental results indicate that the proposed OE+IOE routing scheme provides better fault tolerance than existing turn model, N-random walk, and dual virtual channel based routing schemes that have been proposed in literature.

Categories and Subject Descriptors: C.5.4 [VLSI Systems]

General Terms: Design, Experimentation, Reliability.

Keywords: Networks-on-chip, Fault-tolerant routing.

1. INTRODUCTION

With CMOS technology aggressively scaling into the ultra-deep sub-micron (UDSM) regime and application complexity growing rapidly in recent years, processors today are being driven to integrate multiple cores on a chip. Such chip multiprocessor (CMP) architectures offer unprecedented levels of computing performance for highly parallel emerging applications in the era of digital convergence. However, a major challenge facing the designers of

these emerging multicore architectures is the increased likelihood of failure due to the rise in transient, permanent, and intermittent faults caused by a variety of factors that are becoming more and more prevalent with technology scaling. *Transient faults* (also called soft errors) occur when an event such as high-energy cosmic neutron particle strike, alpha particle strike due to trace uranium/thorium impurities in packages, capacitive and inductive crosstalk, and electromagnetic noise. causes the deposit or removal of enough charge to invert the state of a transistor, wire, or storage cell [1][2]. The inverted value may propagate to cause an error in program execution. These errors occur for a very short duration of time and can be hard to predict. *Permanent faults* occur due to manufacturing defects, or after irreversible wearout damage due to electromigration in conductors, negative bias temperature instability (NBTI), dielectric breakdown, etc. [3][4]. A third class of faults, called *intermittent faults*, occurs frequently and irregularly for several cycles, and then disappears for a period of time [5]. These faults commonly arise due to process variations combined with variation in the operating conditions (such as voltage and temperature fluctuations).

On-chip interconnect architectures are particularly susceptible to faults that can corrupt transmitted data or prevent it from reaching its destination [6]. Reliability concerns in UDSM nodes have in part contributed to the shift from traditional bus based communication fabrics to network-on-chip (NoC) architectures that provide better scalability, performance, and utilization than buses [7][8]. The inherent redundancy in NoCs due to multiple paths between packet sources and sinks can greatly improve communication fault resiliency. Several multicore chip designs are emerging that make use of NoCs as interconnection fabrics [9]-[12]. To ensure reliable data transfers in these communication fabrics, utilizing fault tolerant design strategies is essential. Traditionally, error detection coding and retransmission has been a popular means of achieving resiliency towards transient and intermittent faults [13][14]. Alternatively, forward error correction (FEC) schemes can provide better resiliency against these faults, but usually at a higher performance and energy overhead [15]. Circuit and layout optimizations such as shield insertion and wire sizing to reduce crosstalk induced transient faults have also been proposed [16][17]. To overcome permanent faults in NoCs, fault tolerant routing schemes are a critical requirement and the focus of several research efforts over the last few years [20]-[38]. In the presence of intermittent or permanent faults on NoC links and routers, routing schemes can ensure error free data flit delivery by using an alternate route that is free of faults.

In this paper, we propose a novel low overhead fault tolerant routing scheme (OE+IOE) that combines the odd-even (OE) and inverted odd-even (IOE) turn model based routing schemes to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or fee
CODES+ISSS'10, October 24–29, 2010, Scottsdale, Arizona, USA.
Copyright 2010 ACM 978-1-60558-905-3/10/10...\$10.00.

achieve better fault tolerance than existing fault tolerant routing schemes. The target topology for the scheme is the 2D mesh, which is the most popular interconnection topology in NoCs today because of its layout efficiency, predictable electrical properties, and simple core addressing. The proposed scheme uses replication and dual virtual channels (VCs) at ports opportunistically to optimize the balance between energy overhead and flit arrival rate. Our extensive experimental results and comparison studies indicate that the proposed fault tolerant routing scheme outperforms existing turn model, N-random walk, and dual VC based routing schemes proposed in literature.

2. RELATED WORK

NoC routing schemes can be broadly classified as either static (also called deterministic) or dynamic (also called adaptive). While static routing schemes [18][19] use fixed paths and offer no fault resiliency, dynamic (or adaptive) routing schemes [20]-[38] can alter the path between a source and its destination over time as traffic conditions and the fault distribution changes. The design of adaptive routing schemes is mainly concerned with increasing flit arrival rate, avoiding deadlock, and trying to use a minimal hop path from the source to the destination to decrease transmission energy. Unfortunately, these goals are often conflicting, requiring a complex trade-off analysis that is rarely addressed in existing literature. In general, fault tolerant routing schemes can be broadly classified into three categories: (i) stochastic, (ii) fully adaptive, and (iii) partially adaptive.

Stochastic routing algorithms provide fault tolerance through data redundancy by replicating packets multiple times and sending them over different routes [20]. For instance, the probabilistic gossip flooding scheme [21] allows a router to forward a packet to any of its neighbors with some pre-determined probability. Directed flooding refines this idea by preferring hops that bring the packet closer to its addressed destination [22]. N-random walk [22] limits the number of packet copies by allowing N replications at the source only. In the rest of the network, these copies stochastically take different routes without further replication. The major challenges with these approaches are their high energy consumption, strong likelihood of deadlock and livelock, overhead of calculating, storing, and updating probability values, and poor performance even at low traffic congestion levels.

Fully adaptive routing schemes make use of routing tables in every router or network interface (NI) to reflect the runtime state of the NoC and periodically update the tables when link or router failures occur to aid in adapting the flit path. For instance, [23] proposes using source routing, with routing tables stored at NIs instead of in the routers. But the scheme is only useful for design time faults and cannot adapt to runtime faults. In [24][25], routing tables are used in every router to determine the best path to take in the presence of faults at runtime in mesh topologies. There are several issues with these approaches that prevent their widespread use. First, these schemes require that the routing schemes be updated with global fault and route information frequently, which can take hundreds to thousands of cycles at runtime during which the network state is unstable. In addition, obtaining a global view of the system may not be practically feasible from an energy and performance point of view. Second, the cost and scalability of these schemes can make them prohibitive. As the size of the NoC grows, the table sizes also increase rapidly, increasing router (or NI) area, energy, and latency. Finally, deadlock is a strong possibility that can bring the entire system to a halt, unless high

overhead deadlock recovery mechanisms such as escape channels or distributed timeout counters are used.

Partially adaptive routing schemes enable a limited degree of adaptivity, placing various restrictions and requirements on routing around faulty nodes, primarily to avoid deadlocks. Turn model based routing algorithms such as negative-first, north-last, and south-last [26]-[29] are examples of partially adaptive routing, where certain flit turns are forbidden to avoid deadlock. The odd-even (OE) turn model [29] classifies columns in a mesh topology as either odd or even and prevents different turns in the odd and even columns to prevent cyclic dependencies that can cause deadlocks. In [30], the XY and YX schemes are combined to achieve fault resilient transfers. In [31], routing tables are combined with a customized turn model to avoid deadlock. The work is primarily aimed at design time permanent faults. Unfortunately, the degree of adaptivity provided by these routing algorithms is highly unbalanced across the network, which in some cases results in poor performance.

A few works have proposed using convex or rectangular fault regions with turn models [32]-[38]. Routing detours around these regions can be selected so that deadlock-prone cyclic dependencies are impossible, such as by using turn models or cycle free contours. For instance, [33] builds a fault ring around faults and introduces rules to route flits around the fault region to avoid deadlock. But the scheme can only handle design time permanent faults (i.e., fault regions are built only at design time and cannot handle runtime faults) and also requires up to four VCs, which can quickly become cost ineffective as the NoC size scales. In [34], the OE turn model was used together with fault regions. However, the work was based on assumptions that no fault exists in the two columns that are adjacent to the west and east edges of the mesh. In general, fault tolerant routing schemes that make use of fault regions are either too conservative (disabling fully functional routers to meet region shape requirements), have restrictions on the locations of the faults that can be bypassed, or cannot adapt to runtime intermittent and permanent faults.

In contrast to existing work, our partially adaptive OE+IOE fault tolerant routing scheme achieves fault resilient transfers by balancing data replication, resource redundancy, and path redundancy to achieve a high arrival rate with a low energy and cost overhead. Data replication is limited to environments with high fault rates, which ensures a low cost operation in the absence of faults (or for scenarios with very few faults). Path diversity as a result of utilizing dual turn models ensures that even if one packet hits a dead end, the other has a possibility of arriving at the destination. The scheme can adapt to handle not only design time permanent faults, but also runtime intermittent and permanent faults. Experimental results in Section 4 that compare the OE+IOE routing scheme with existing fault tolerant routing schemes indicate that our proposed scheme has several advantages and can be a viable fault tolerant routing scheme for emerging CMPs.

3. OE-IOE ROUTING SCHEME

In this section, we present an overview of our proposed OE+IOE fault tolerant routing scheme for 2D mesh NoCs. Section 3.1 describes the odd-even (OE) and inverted odd-even (IOE) turn models. Section 3.2 presents details of the implementation and operation of the proposed routing scheme that combines OE and IOE turn models. Finally, Section 3.3 describes the router architecture and the control network that supports our OE+IOE routing scheme in a NoC fabric.

3.1 OE and Inverted OE Turn Models

The odd-even (OE) turn model for deadlock-free routing in 2D meshes was introduced by Chiu [29]. A turn in this context refers to a 90-degree change of traveling direction for a flit. There are eight types of turns that are possible in a 2D mesh based on the traveling directions of the associated channels. To facilitate the explanation, we label the four sides of the 2D mesh as East, West, South, and North (Figure 1). Then a turn is called a NE turn if it involves a flit traveling in the North direction and attempting to turn East. Similarly, we can define the other seven types of turns, namely EN, WS, WN, SE, SW, ES, and NW turns, where E, W, S, and N indicate East, West, South, and North, respectively.

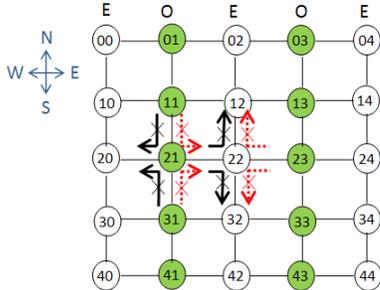


Figure 1. OE and IOE prohibited turns based on column

Deadlock in wormhole routing is caused by flits waiting on each other in a cycle. Unlike traditional turn models [26] that avoid deadlock by prohibiting certain turns, the OE turn model restricts the locations at which certain turns can occur to ensure that a circular wait does not occur. In OE turn model based routing, columns in a 2D mesh are alternately designated as odd (O) and even (E), as shown in Figure 1 for a 5x5 2D mesh. Then the following two forbidden turn rules ensure cycle- and deadlock-free routing: (i) a packet is not allowed to take an EN or ES turn at any node located in an even column; and (ii) a packet is not allowed to take a NW or SW turn at any node located in an odd column. A deadlock-free minimal path route can then be found for any source destination pair as described in [29], and we refer the reader to that paper for details. The prohibited turns for the OE turn model are depicted in black in Figure 1. No other restrictions need to be applied to ensure freedom from deadlock. The inverted odd-even (IOE) turn model can be understood by rotating the mesh by 180 degrees, while preserving the odd-even column designations and corresponding prohibited turns. Figure 1 shows the prohibited turns in the odd and even columns for the IOE turn model in red, which are required to ensure deadlock free operation.

3.2 OE+IOE Routing Scheme: Overview

To ensure robustness against faults, redundancy is a necessary requirement, especially for environments with high fault rates and unpredictable fault distributions. As the level of redundancy is increased, system reliability improves as a general rule. However, redundancy also detrimentally impacts other design objectives such as power and performance. Therefore in practice it is important to limit redundancy to achieve a reasonable trade-off between reliability, power, and performance. Unlike directed and random probabilistic flooding algorithms that propagate multiple copies of a packet to achieve fault tolerant routing in NoCs, the proposed OE+IOE routing algorithm sends only one redundant packet for each transmitted packet, and only if the fault rate is above a replication threshold δ . The value of δ is a designer-

specified parameter that depends on the application characteristics, routing complexity, and power-reliability trade-off requirements. The original packet is sent using the odd-even (OE) turn model while the redundant packet is propagated using an inverted odd-even (IOE) turn model based routing scheme. This packet replication happens only at the source. Two separate VCs, one dedicated to the OE packets and the other for the IOE packets ensure deadlock freedom. If the fault rate is low (i.e., below δ), replication is not utilized and only the original packet is sent using the OE scheme while power/clock gating the IOE virtual channel to save power. The proposed routing algorithm prioritizes minimal paths that have higher probabilities of reaching the destination even if faults are encountered downstream. Minimal paths and the replication threshold ensure low power dissipation under a diverse set of fault rates and distributions. No restriction on the number or location of faults is assumed, but the routers must know which of their adjacent (neighbor) links/nodes are faulty, which is accomplished using minimal control signaling. The proposed routing approach can be combined with error control coding (ECC) techniques for transient fault resiliency and optimizations such as router buffer reordering [39] and router/NI backup paths [40] to create a comprehensive fault tolerant NoC fabric. In the following subsections, we describe the implementation of the proposed OE+IOE routing scheme.

3.2.1 Turn Restriction Checks

Whenever a packet arrives at a router in the OE+IOE scheme, three scenarios are possible: (i) there is no fault in the adjacent links and the packet is routed to the output port based on the OE (or IOE) scheme that selects a minimal path to the destination, (ii) there are one or more faults on adjacent links that prevent the packet from propagating in a valid direction (i.e., an output port with a fault-free adjacent link that does not violate turn model routing rules) in which case the packet must be dropped as a back turn is not allowed, and (iii) one or more adjacent links have faults but a valid direction exists to route the packet towards, based on the OE (or IOE) rules. A packet traversing an intermediate router must choose among four output directions (N, S, W, E). However, choosing some directions may lead to a violation of a basic OE (or IOE) turn rule immediately or downstream based on the relative location of the destination. Figure 2 shows the SystemC [41] pseudocode of the turn restriction check phase for the OE turn model in our router model to aid in detecting an invalid routing direction. The reference to XY coordinates follows the convention as shown in Figure 1 where each node is labeled with its XY coordinates. Note that the pseudocode for the IOE scheme is similar with the directions inverted, and is not shown here for brevity.

First, we check whether the output port direction has a fault in its attached adjacent link (Steps 2-4), in which case this is an invalid direction. Next, we check the forbidden turn rules for the OE turn model as discussed in Section 3.1 (Steps 5-10) based on the router location (in an odd or even column), the input port of the packet, and its output port direction. If the packet is attempting a forbidden turn, then the direction is invalid. If the direction has no adjacent faults and does not violate the basic OE routing rules, we then check if the direction will lead to a turn rule violation downstream based on the location of the destination. We start by checking for the scenario when the packet is attempting to go in the East direction (Steps 11-18). If the destination is on an even column ($\text{dest_yc}\%2 = 0$) and the current packet is on its left neighbor column but not in the same row as the destination, then the packet should not be routed to the East direction (Steps 12-14). This is

because after it arrives at the destination column, it has to route in the North or South direction to reach the destination row, which however breaks the basic OE turn rules and will not be allowed. Alternatively, if the destination is to the West of the current node, then the East direction is also forbidden (Steps 15-17). This is because once the packet is routed in the East direction, it cannot go back in the West direction without violating the basic OE turn rules (if it arrives at an even column and wants to go West, it must go North or South which breaks the basic OE turn rules; if it arrives at an odd column then after turning North or South it must go West which also breaks the basic OE turn rules).

Pseudocode: Turn Restrictions for OE Implementation

```

1: RESULT check_neighbor(UI ip_dir, UI direct, ULL dest_id) {
//ip_dir: direction where the packet comes from
//direct: direction we want to check
//cur_id: current node position
//dest_id: destination node position
//dest_yco, dest_xco: y,x coordinates of destination node
//cur_yco,cur_xco: y,x coordinates of current node
//dif_yco=dest_yco - cur_yco
//RESULT: LINK_FAULT, TURN_FAULT, BACK_TURN, OK
2:   if (check_linkfault(direct)==LINK_FAULT){
3:     return LINK_FAULT;
4:   }
5:   if ((cur_yco%2==0)&&(ip_dir==E)&&(direct==N||direct==S)) {
6:     return TURN_FAULT;
7:   }
8:   if ((cur_yco%2==1)&&(ip_dir==N||ip_dir==S)&&(direct==W)) {
9:     return TURN_FAULT;
10:  }
11:  if (direct==E) {
12:    if ((dest_xco!=cur_xco)&&(dif_yco==1)&&(dest_yco%2==0)) {
13:      return TURN_FAULT;
14:    }
15:    if (dest_yco<=cur_yco) {
16:      return TURN_FAULT;
17:    }
18:  }
19:  if ((dif_yco==1)&&(dest_yco%2==0)) {
20:    if ((dest_xco<cur_xco)&&(direct==S)) {
21:      return TURN_FAULT;
22:    } else if ((dest_xco>cur_xco)&&(direct==N)) {
23:      return TURN_FAULT;
24:    } else if ((dest_xco==cur_xco)&&(direct==N||direct==S||direct==W)) {
25:      return TURN_FAULT;
26:    }
27:  }
28:  if ((dest_yco<cur_yco)&&(cur_yco%2!=0)&&(direct==N)||direct==S))
29:    return TURN_FAULT;
30:  if (((borderW(cur_id)||cur_yco%2!=0)&&(cur_yco==dest_yco)) {
31:    if ((cur_xco<dest_xco)&&(direct==N)) {
32:      return TURN_FAULT;
33:    } else if ((cur_xco>dest_xco)&&(direct==S)) {
34:      return TURN_FAULT;
35:    }
36:  }
37:  if (ip_dir==direct) {
38:    return BACK_TURN;
39:  }
40:  return OK;
41: }

```

Figure 2. SystemC pseudocode for OE turn restrictions

Next we check for the scenario where the destination is on the even column and the current packet is on its left neighbor column and attempting to go in a direction other than the East direction (Steps 19-27). In this case, when the current node is not on the same row as the destination, then the only choice is to go North or South towards that destination row. If the current node is on the same row as the destination, then it can only choose the East direction. The next check is for the scenario where the current node is on an odd column and the destination is to its West direction (Steps 28-

29). In this case, the North and South directions are both forbidden. This is because if a packet goes in the North or South directions, it will ultimately need to go West, which is not possible without violating the basic OE turn rules. Next, we check for the scenario where the current node is on the same column as the destination, but the current node is also on the West border of the mesh or on an odd column (Steps 30-36). In this case if the packet moves North or South in a direction away from the destination, it will not be able to turn West on the West border or make a subsequent West turn when in an odd column to reach the destination without violating basic OE turn rules. Finally, we check for a back turn, which is not allowed (Steps 37-39). If all these checks pass, then the given direction is valid for sending the packet (Step 40).

The turn restriction check procedure described above is invoked in parallel for all four possible directions in an intermediate node for a header flit. We created a gate level implementation of the circuit and synthesized it to obtain power and latency overhead of the scheme. At the TSMC 65nm process technology, we observed that our implementation has low overhead, dissipating a nominal 2.03 μ W of power on average, and with a critical path latency of 0.54 ns.

3.2.2 Prioritized Valid Path Selection

After the turn restriction checks, it is possible for a packet to have multiple valid directions that it can follow from its current location. To ensure low energy transfers, directions in minimal paths are always given higher priority in our scheme. If, however, two valid directions in minimal paths exist to the destination from the current router, the goal is to give higher priority to the direction that allows for greater path diversity so that the packet can still route around faults downstream and reach its destination. In the OE (or IOE) turn models, path diversity is strongly dependent on the location of destination. In our proposed OE+IOE scheme, given two minimal paths to the destination, *we prioritize the North and South directions over the East and West directions.*

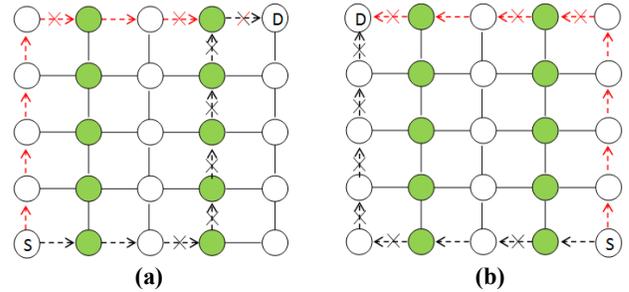


Figure 3. Choosing minimal paths (a) destination to the North-East of source, (b) destination to the North-West of source

Consider Figure 3 (a) which shows a case where the destination is to the North-East of the current (in this case source) node. Two valid minimal paths exist from source S to destination D – one to the North (red dashed line) and the other to the East (black dashed line). We use the convention of white nodes lying in the even column and green nodes lying in the odd column. If the East minimal path encounters any one of the six faults shown with an ‘X’, the packet will have to be dropped as it cannot reach the destination. In contrast, with a North minimal path, there are only three possible faults that will cause the packet to be dropped; faults in any of the other links can be routed around using another valid alternate path. Thus, there is a greater probability of the packet reaching its destination if the North direction is given greater priority over the East direction. Figure 3 (b) shows the case where the destination is to the North-West of the source node, with the

North minimal path again having higher probability of delivering the packet to the destination compared to the West minimal path. Similar arguments hold for cases when the destination is to the South-East and South-West of the source, with South minimal paths providing greater probability of successful packet arrival than if the West or East minimal paths are chosen.

3.2.3 Stochastic Valid Path Selection

An alternative to using prioritized minimal paths during OE or IOE routing is to stochastically choose one of the available valid paths. In this method, every outgoing channel in a router is assigned a probability value: P_N , P_S , P_E , and P_W , where the sum of all probabilities is 1. We assume that valid paths have an equal probability of being selected. For instance, if a packet enters a router from the South port and a West turn is forbidden, then $P_S = 0$ (back turn not allowed), $P_W = 0$ (forbidden turn), and therefore $P_E = P_N = 0.5$. Thus the North or the East directions can be chosen with equal probability. The usefulness of this technique is in enabling the selection of non-minimal valid paths, which in some cases may lead to higher arrival rate, at the cost of higher energy consumption. We contrast the prioritized and stochastic path selection techniques in our experiments, presented in Section 4.

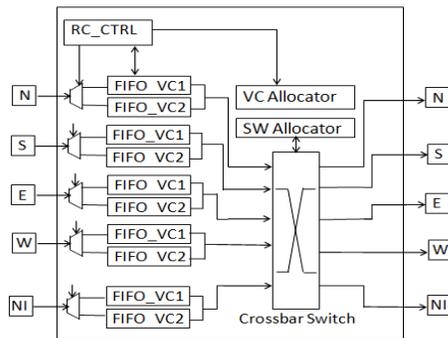


Figure 4. Router architecture

3.3 Router Architecture

Figure 4 depicts our virtual channel (VC) router architecture. Packets traversing the network have a header flit with a destination ID field that is used to determine the output port at each router. The router datapath consists of the buffers and the switch. The input FIFO buffers store up to 16 flits waiting to be forwarded to the next hop. There are two input FIFO buffers each dedicated to a VC, with one VC for the OE routed packets and the other for the IOE routed packets. When a flit is ready to move, the switch connects an input buffer to an appropriate output channel. To control the datapath, the router contains three major control modules: a route compute control unit (RC_CTRL), a virtual-channel (VCA) allocator, and a switch allocator (SA). These control modules determine the next hop direction, the next VC, and when a switch is available for each packet/flit. The routing operation takes four phases: route computation (RC), virtual-channel allocation (VCA), switch allocation (SA), and switch traversal (ST). When a head flit (the first flit of a packet) arrives at an input channel, the router stores the flit in the buffer for the allocated VC and determines the next hop for the packet using the checks and priority rules described in the previous sections (RC phase). Given the next hop, the router then allocates a virtual channel (VCA phase). Finally, the flit competes for a switch (SA phase) if the next hop can accept the flit, and moves to the output port (ST phase). In our implementation, a packet carries its virtual

channel (OE or IOE) information in a bit in the header flit, therefore the output port and the destination VC can be obtained at the end of the RC stage.

To reduce energy consumption, if the fault rate is below the replication threshold δ , the RC_CTRL unit shuts down the IOE virtual channels in the router. We assume that dedicated control signals connected to a lightweight fault detection unit (FDU) can allow estimation of the fault rate at runtime in the NoC, and the decision to change the replication status can be communicated to the nodes and implemented with a delay of a few hundred cycles. Typically, such a change in status would be a rare occurrence as permanent faults do not appear at runtime very frequently.

An important requirement for any fault tolerant NoC fabric is a control network that can be used to send acknowledgement signals to inform the source whether the packet successfully arrived at the destination. We assume a lightweight and fault-free control network that is topologically similar to the data network for routing ACK signals from the destination to the source node on successful packet arrival, or NACK signals from an intermediate node to the source node if the packet must be dropped due to faults that make it impossible to make progress towards the destination. In our implementation, a source can re-send a packet at most twice after receiving NACK signals, before assuming that the packet can never reach its destination. While this can signal unavoidable failure for certain types of applications, other applications may still be able to operate and maintain graceful degradation. For instance, a multiple use-case CMP can shut down certain application use-cases when their associated inter-core communication paths encounter fatal faults, but there might be other use cases that can still continue to operate on fault-free paths. We assume that a higher level protocol (e.g. from the middleware or OS levels) can step in to initiate thread migration and other strategies to ensure useful operation in the presence of unavoidable system faults.

4. EXPERIMENTAL STUDIES

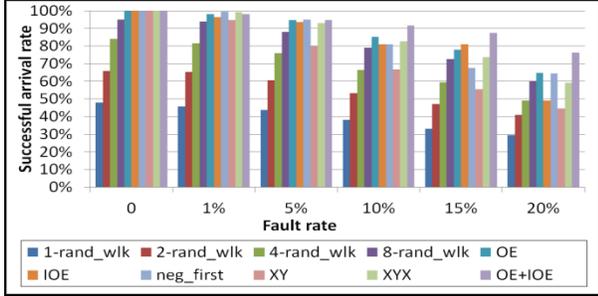
In this section we present the results of our experimental studies. Section 4.1 describes our experimental setup. Section 4.2 compares the proposed OE+IOE routing scheme with several fault tolerant routing schemes proposed in literature. Finally, Section 4.3 contrasts variants of our proposed fault tolerant routing scheme.

4.1 Experimental Setup

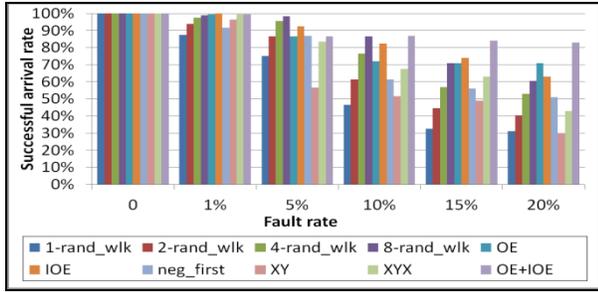
We implemented and evaluated our proposed algorithm using a SystemC based cycle-accurate 2D mesh NoC simulator that was created by extending the open source Nirgam simulator [42] to support faults and different fault tolerant routing schemes. In addition to communication performance estimates, we were interested in NoC power and energy estimation as well. For this purpose we incorporated dynamic and leakage power models from Orion 2.0 [43] into the simulator. The NoC fabric was clocked at 1 GHz. The target implementation technology was 65nm, and this node was used to determine parameters for delay and energy consumption. Two different core complexities were considered for our analysis: 6×6 (36 cores) and 9×9 (81 cores). The stimulus for the simulations was generated for specific injection rates using three different traffic models: uniform random, transpose, and hotspot. A fixed number of flits (3000/core) were transmitted according to the specified traffic patterns in the simulations, to enable comparisons for not only successful arrival rates, but also for overall communication energy. Faults in the NoC were modeled as randomly distributed link-level hard failures that can represent design time (fabrication related) or runtime (wearout

related) failures. Ten different random fault distributions were generated and analyzed for each fault rate, and for each traffic type to obtain a more accurate estimate of fault resiliency for the routing schemes. The faults can cause routers to be only partially functional, or entirely non-functional if all of the links of a router are faulty. Both of these scenarios were tested in the simulations.

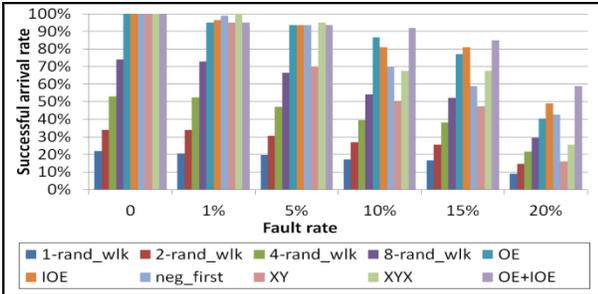
energy consumption is an order of magnitude higher than other schemes and frequent deadlocks hamper overall performance. For our OE+IOE scheme, we used a prioritized valid path selection (Section 3.2.2) and a replication threshold value of $\delta = 6\%$, based on our extensive simulation studies that showed a good trade-off between arrival rate and energy for $8\% \geq \delta \geq 2\%$.



(a)



(b)

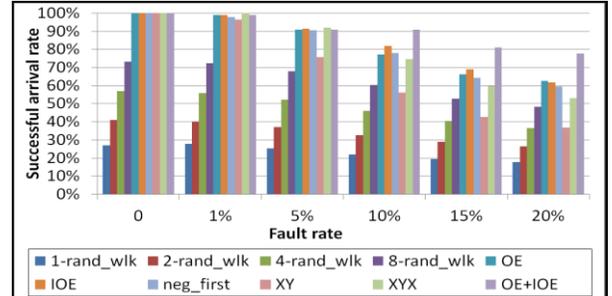


(c)

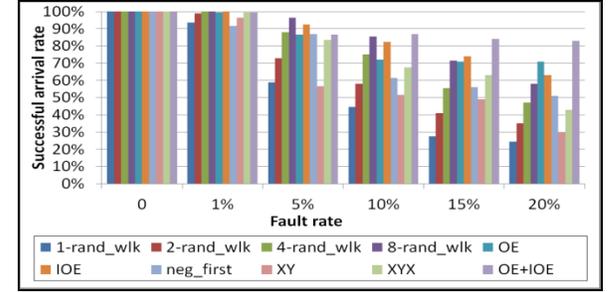
Figure 5. Packet arrival rate for 6×6 NoC (a) Uniform random traffic, (b) Hotspot traffic, (c) Transpose traffic

4.2 Comparison with Existing FT Routing Schemes

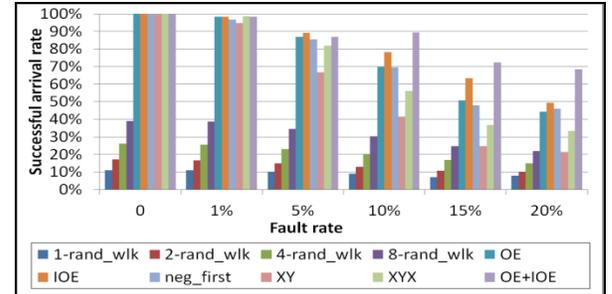
To demonstrate the effectiveness of our proposed OE+IOE fault tolerant routing scheme for NoCs, we compared it to several existing fault tolerant routing schemes from literature. The following schemes were considered in the comparison study: (i) XY dimension order routing [18] (although not inherently fault tolerant, it is considered here because of its widespread use in 2D mesh NoCs), (ii) N-random walk [22] for $N = 1, 2, 4, 8$, (iii) adaptive negative first turn model [27], (iv) adaptive OE turn model [29], (v) adaptive IOE turn model, and (vi) XYX [30] which combines replication with the XY and YX routing schemes on two VCs. Other fault tolerant routing schemes such as probabilistic flooding [21] and fully adaptive table based routing [24] were considered but their results are not presented because these schemes have major practical implementation concerns - their



(a)



(b)



(c)

Figure 6. Packet arrival rate for 9×9 NoC (a) Uniform random traffic, (b) Hotspot traffic, (c) Transpose traffic

Figure 5 (a)-(c) compares the successful packet arrival rate for various fault tolerant routing schemes under different fault rates. Results are shown for the three traffic types, a 6×6 NoC, and a 20% injection rate (0.2 flits/node/cycle). It can be seen that the N-random walk schemes have a low arrival rate even in the absence of faults, for the uniform and transpose traffic types. This is due to frequent deadlocks during simulation which the scheme is susceptible to, causing packets to be dropped. As hotspot traffic has fewer actively transmitting cores than the other traffic types, the arrival rates for N-random walk schemes are a little better for low fault rates (as there is lower probability of deadlock) but get worse as the fault rate increases. The best arrival rate for the N-random walk scheme is achieved for a value of $N=8$. Out of the turn/dimension-order based routing schemes, XY not surprisingly performs worse than the other schemes due to the lack of any built

in fault tolerance capability. For low fault rates, OE, IOE, and negative first have a comparable successful arrival rate, but for higher fault rates OE and IOE on average perform better than the negative first scheme. Interestingly, the dual virtual channel XYX scheme performs better than other schemes for low fault rates ($\sim 1\%$), but for higher fault rates, its successful arrival rates are on average lower than those for single virtual channel OE, IOE, and negative first turn model based schemes! The lack of path diversity in the XYX scheme leads to this dramatic reduction in arrival rate as the number of faults in the network increase. Our proposed OE+IOE fault tolerant routing scheme can be seen to have a much higher successful packet arrival rate compared to the other schemes, especially for higher fault rates. The results indicate that the OE+IOE scheme can scale well with the higher fault rates that are expected to be the norm in future CMP designs.

To explore how the fault tolerant schemes fare as the size of the CMP increases, we repeated the experiments on a 9×9 NoC. Figure 6 (a)-(c) shows the successful packet arrival rates for the same set of routing schemes under different fault rates and a 20% injection rate, for a 9×9 NoC. Once again it can be seen that the OE+IOE scheme has a greater fault tolerance than the other schemes, showing how well the scheme scales with increasing levels of core integration. Results for other injection rates are not shown here because as long as the fault distribution and source-destination pairs remain the same, changing the injection rate does not impact the successful packet arrival rate (although it does impact energy and average packet latency).

In addition to arrival rate, an increasingly important metric of concern for designers is the energy consumption in the communication network. Figure 7 (a)-(c) shows the energy consumption (in Joules) for the various routing schemes under different fault rates, for a 9×9 NoC with a 20% injection rate. Results for a smaller 6×6 NoC are omitted for brevity as they follow a similar trend to the 9×9 NoC. It can be seen that the N-random walk fault tolerant routing schemes have significantly higher energy consumption compared to the other schemes. This is due to the high level of replication with increasing values of N , as well as because of the non-minimal paths chosen to route the packets on. For single VC schemes (OE, IOE, XY, and negative first), the XY scheme has the lowest energy dissipation which explains its popularity in NoCs today. But as it is lacking in any fault tolerance, the scheme may not be a viable option for future CMPs. The negative first scheme has lower energy consumption than the OE and IOE schemes on average for hotspot traffic, and higher energy consumption on average for uniform random and transpose traffic scenarios. The XYX scheme makes use of two VCs and replication, therefore it does not come as a surprise to see that it has higher energy consumption than the single VC schemes. Our proposed OE+IOE scheme uses the replication threshold δ to trade-off energy consumption with fault tolerance. For the chosen value of $\delta=6\%$, our scheme utilizes a single VC and no replication for low fault rates ($< 6\%$) and then switches to its dual VC mode with replication for higher fault rates. This explains the low energy consumption for low fault rates for the OE+IOE scheme in Figure 7. For higher fault rates ($\geq 10\%$), the OE+IOE energy consumption is comparable to that of XYX for hotspot traffic, and somewhat higher than XYX for uniform random and transpose traffic. But the higher energy consumption for OE+IOE is an artifact of its much higher successful packet arrival rate. The overall energy consumption for the scheme is still significantly lower than other fault tolerant schemes such as stochastic flooding, N-random walk and table based source or distributed routing. If lower energy is

desirable, the value of the replication threshold can be increased by designers. For instance, for multimedia applications, occasionally dropped pixel data may not be perceivable by viewers, and in such scenarios the resulting lower energy consumption may lead to a much longer battery life and better overall user experience. Conversely, if fault resiliency is paramount, the value of δ can be reduced to close to 0, which will enable the highest possible successful packet arrival rate.

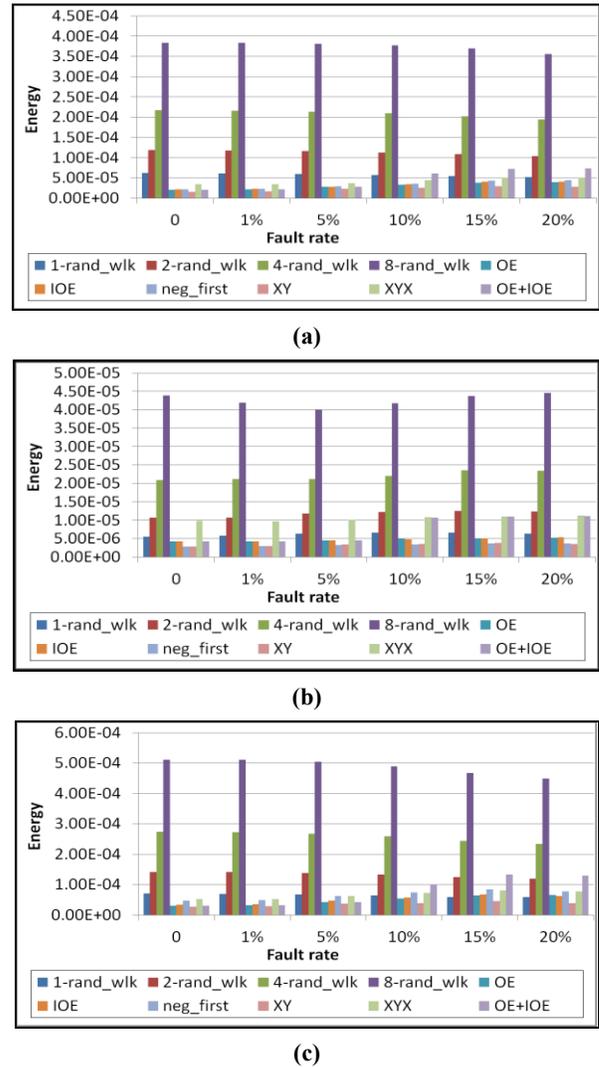
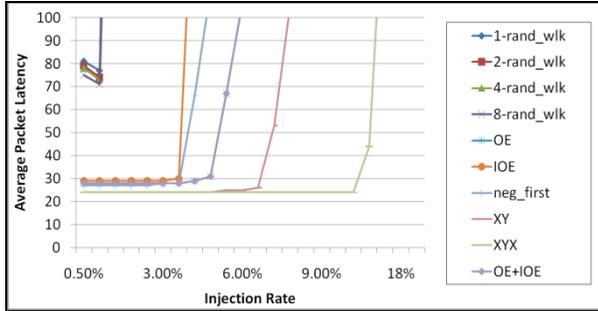


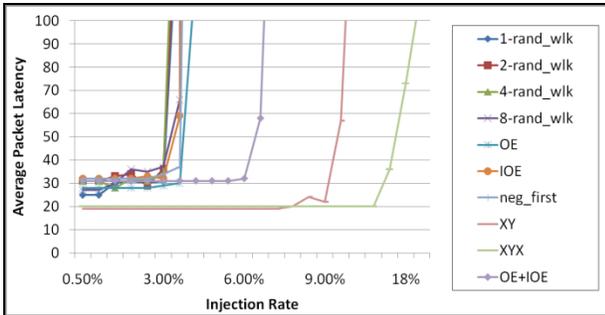
Figure 7. Communication energy for 9×9 NoC (a) Uniform random traffic, (b) Hotspot traffic, (c) Transpose traffic

Figure 8 (a)-(b) summarizes the average packet latency (in cycles) for the routing schemes for various injection rates under the uniform random traffic model. The results are shown for a 9×9 NoC with low fault rate (1%) and high fault rate (20%). The lowest packet latency is achieved for the XY scheme for low injection rates and the XYX scheme for higher injection rates, which are both minimal path schemes. The latency of the XYX scheme is lower than that of the XY scheme for high injection rates because the XY scheme encounters more faults at high injection rates that cause it to drop packets more frequently than the XYX scheme which in some cases still manages to deliver packets using its YX

route even if the XY route has a fault in it. The OE+IOE scheme has the next lowest latency on average, which coupled with its higher successful arrival rate compared to the XY, XYX, and the rest of the schemes makes it an extremely viable alternative for fault tolerant routing in future NoC based CMPs.



(a)



(b)

Figure 8. Average packet latency for 9x9 NoC with uniform random traffic (a) 1% fault rate, (b) 20% fault rate

4.3 Comparison with OE+IOE Variants

Next, we compare our OE+IOE scheme that uses prioritized path selection and selective replication with two of its variants: (i) OE+IOE with prioritized path selection and without selective replication (OE+IOE_{no_δ}), and (ii) OE+IOE with stochastic valid path selection and with selective replication (OE+IOE_S). Figure 9 shows the successful packet arrival rate for our proposed scheme (OE+IOE) compared to the OE+IOE_S and OE+IOE_{no_δ} variants for different fault rates and the three traffic types, for a 9x9 NoC and 20% injection rate. The highest successful packet arrival rate on average is obtained for the OE+IOE_{no_δ} scheme which always uses replication and dual VCs. The OE+IOE scheme has a similar behavior and hence packet arrival rate compared to the OE+IOE_{no_δ} scheme for fault rates higher than the δ threshold. But for lower fault rates, our emphasis on lower energy consumption with the OE+IOE scheme translates into a slightly lower arrival rate than OE+IOE_{no_δ}. Using stochastic valid path selection in the OE+IOE_S scheme leads to higher arrival rates than the OE+IOE scheme only for low fault rates below δ . For higher fault rates, it can be clearly seen that stochastic valid path selection is not as efficient as prioritized valid path selection.

Figure 10 compares the energy consumption (in Joules) for the OE+IOE, OE+IOE_S, and OE+IOE_{no_δ} schemes under different fault rates for the same 9x9 NoC and traffic scenarios as above. The extremely low energy consumption of the OE+IOE scheme compared to its variants for low fault rates is a strong motivation for the existence of the replication threshold parameter. In general,

the OE+IOE_S scheme employs more non-minimal paths than the other schemes which leads to longer packet latencies and higher overall energy consumption. For high fault rates under transpose traffic, the OE+IOE_S energy consumption is lower, but only because it has a lower packet arrival rate with many more dropped packets before they can complete their routes. Thus, it can be seen that OE+IOE not only provides much higher fault tolerance compared to existing fault tolerant routing schemes, but also enables a trade-off between reliability and energy consumption that can be tuned on a per-application basis by the designer.

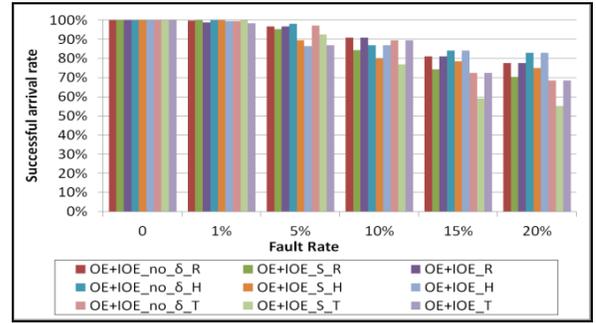


Figure 9. Packet arrival rate for 9x9 NoC for three OE+IOE variants under uniform random traffic (R), hotspot traffic (H), and transpose traffic (T)

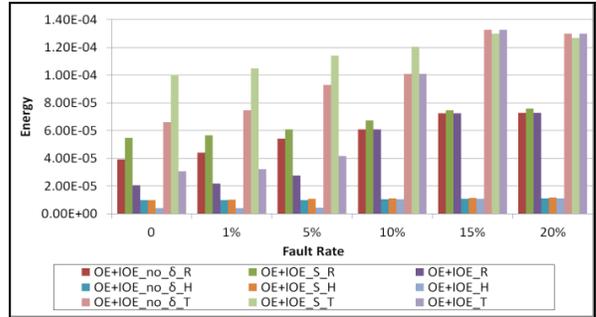


Figure 10. Communication energy for 9x9 NoC for three OE+IOE variants under uniform random traffic (R), hotspot traffic (H), and transpose traffic (T)

5 CONCLUSION AND FUTURE WORK

Rapidly scaling CMOS technology and ever increasing levels of core integration on chip multiprocessors (CMPs) have led to an increase in transient, intermittent, and permanent faults that can lead to chip failure. There is thus a critical need today to focus on fault tolerant design techniques to overcome the detrimental impact of faults on emerging CMP designs. In this paper we proposed a novel fault tolerant routing scheme (OE+IOE) for 2D mesh NoCs that can adapt to design time and runtime permanent link faults, as well as potential intermittent faults in NoC communication architectures. Our scheme uses replication opportunistically based on fault rate, and combines the odd-even (OE) and inverted odd-even (IOE) turn models to achieve deadlock free packet traversal. Experimental results show that our proposed OE+IOE scheme can provide better fault tolerance (i.e., higher successful packet arrival rates) than traditional fault tolerant routing schemes such as N-random walk and turn model based schemes. Our ongoing and future work is exploring novel fault tolerant routing schemes for irregular NoC topologies.

ACKNOWLEDGEMENTS

This research was supported in part by the NSF under grant CNS-0905399.

REFERENCES

- [1] S. S. Mukherjee, J. Emer, S. K. Reinhardt, "The soft error problem: An architectural perspective," Proc. HPCA, 2005.
- [2] E. Normand, "Single event upset at ground level," *IEEE Trans. on Nuclear Science*, 43(6):2742–2750, Dec 1996.
- [3] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," *IEEE Micro*, 23(4):14-19, July-Aug 2003.
- [4] S. Nassif, "Modeling and analysis of manufacturing variations," Proc. CICC, May 2001.
- [5] C. Constantinescu, "Intermittent faults in VLSI circuits," Proc. SELSE, 2007.
- [6] S. Pasricha and N. Dutt, *On-Chip Communication Architectures*, Morgan Kaufman, Apr 2008.
- [7] L. Benini and G.D. Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, pp. 70-78, Jan. 2002.
- [8] W. J. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks," Proc. DAC, pp. 684-689, 2001.
- [9] D.C. Pham, et al., "Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor," *IEEE J. Solid-State Circuits*, 41(1):179-196, 2006
- [10] Intel Teraflops, http://download.intel.com/research/platform/terascale/terascale_overview_paper.pdf.
- [11] Picochip PC102. <http://www.picochip.com/highlights/pc102>.
- [12] S. Bell et al., "TILE64 processor: A 64-core SoC with mesh interconnect," Proc. ISSCC, 2008.
- [13] D. Bertozzi, L. Benini, G. De Micheli, "Error control schemes for on-chip communication links: the energy–reliability tradeoff," *IEEE Trans. CAD*, 24(6):818-831, 2005.
- [14] S. Murali et al., "Analysis of error recovery schemes for networks on chips," *IEEE Design & Test of Computers*, 22(5): 434-442, 2005.
- [15] S. Lin and D. J. Costello, *Error control coding: fundamentals and applications*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [16] D. Bertozzi, L. Benini, G. De Micheli, "Low power error resilient encoding for on-chip data buses," Proc. DATE, pp. 102-109, 2002.
- [17] M. Lajolo, "Bus guardians: an effective solution for online detection and correction of faults affecting system-on-chip buses," *IEEE Trans. VLSI*, 9(6):974-982, Dec. 2001
- [18] W.J. Dally, B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufman, 2004.
- [19] M. Dehyadgari, M. Nickray, A. Afzali-kusha, Z. Navabi, "Evaluation of pseudo adaptive XY routing using an object oriented model for NoC," Proc. MICRO, pp. 13–15, 2005.
- [20] H. Zhu, P. P. Pande, C. Grecu, "Performance evaluation of adaptive routing algorithms for achieving fault tolerance in NoC fabrics," Proc. ASAP 2007.
- [21] T. Dumitras, R. Marculescu, "On-chip stochastic communication," Proc. DATE, 2003.
- [22] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect," Proc. ISVLSI, 2004.
- [23] Y. B. Kim, Y.-B. Kim, "Fault tolerant source routing for network-on-chip," Proc. DFT, 2007.
- [24] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Fully adaptive fault tolerant routing algorithm for network-on-chip architectures," Proc. DSD, pp. 527–534, Aug. 2007.
- [25] T. Schonwald, O. Bringmann, W. Rosenstiel, "Region-based routing algorithm for network-on-chip architectures," Proc. Norchip 2007.
- [26] C. J. Glass, L. M. Ni, "The turn model for adaptive routing," Proc. ISCA, pp. 278 – 287, 1992.
- [27] C. J. Glass and L. M. Ni, "Fault-tolerant wormhole routing in meshes without virtual channels," *IEEE Trans. Parallel and Distributed Systems*, 7(6):620-635, 1996.
- [28] C. M. Cunningham, D. R. Avresky, "Fault-tolerant adaptive routing for two-dimensional meshes," Proc. HPCA, 1995.
- [29] G.-M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel and Distributed Systems*, 11(7), pp.729–738, 2000.
- [30] A. Patooghy, S.G. Miremadi, "XYX: A power & performance efficient fault-tolerant routing algorithm for network on chip," Proc. ICPDNP, pp. 245 – 251, 2009
- [31] D. Fick, A. DeOrio, G.K. Chen, V. Bertacco, D. Sylvester, D. Blaauw, "A highly resilient routing algorithm for fault-tolerant NoCs," Proc. DATE, 2009.
- [32] R.V. Boppana and S. Chalasani, "Fault-tolerant wormhole routing algorithms for mesh networks," *IEEE Trans. on Computers*, 44(7):848–864, 1995.
- [33] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model," *IEEE Trans. on Computers*, 52(9):1154-1169, Sep 2003.
- [34] A. Rezazadeh, M. Fathy, A. Hassanzadeh, "If-cube3: An improved fault-tolerant routing algorithm to achieve less latency in NoCs," Proc. IACC, pp. 278-283, 2009.
- [35] S. Jovanovic, C. Tanougast, S. Weber, C. Bobda, "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections," Proc. FPLA, pp. 326 – 331, 2009.
- [36] M. Andres, P. Maurizio, F. José, "Region-Based Routing: A Mechanism to Support Efficient Routing Algorithms in NoCs," *IEEE Trans. VLSI*, 17(3):356-369, 2009 .
- [37] J. Hu, R. Marculescu, "Dyad - smart routing for networks-on-chip," Proc. DAC, 2004.
- [38] Z. Zhang, A. Greiner, S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-mesh network-on-chip," Proc. DAC, 2008.
- [39] W.-C. Kwon, S. Yoo, J. Um, S.-W. Jeong, "In-network reorder buffer to improve overall NoC performance while resolving the in-order requirement problem," Proc. DATE, pp. 1058-1063, 2009.
- [40] M. Koibuchi, H. Matsutani, H. Amano, T. M. Pinkston, "A lightweight fault-tolerant mechanism for network-on-chip", Proc. NOCS 2008.
- [41] SystemC initiative. <http://www.systemc.org>.
- [42] Nirgam simulator <http://nirgam.ecs.soton.ac.uk/>.
- [43] A. Kahng, B. Li, L.-S. Peh and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," Proc. DATE, 2009.