

FAULT-TOLERANT ROUTING SCHEME FOR NOCS

USING DYNAMIC RECONFIGURATION OF PARTIAL-FAULTY

ROUTING RESOURCES

ABSTRACT:

A NoC architecture offers high reliability since it has multiple routes from the host to clients. A fault tolerant NoC framework is proposed which achieves maximum performance under fault. NoCs under fault become totally unfunctional. Hence the faulty components are handled separately which ensure that the network is not partitioned and results in high connectivity even under high fault.

NoCs are a very complex network and are susceptible to a lot of faults. Faults can be temporary and permanent. Temporary faults occur due to the on-chip cross-talks and coupling noise. Permanent faults mostly occur due to worn-out devices, manufacturing defects and accelerated aging effects. These faults can be handled using router level dynamic reconfiguration schemes such as Dynamic Buffer swapping and Dynamic Mux Swapping. These methods deal with the crossbar and buffer faults which are the main sources of fault in the buffer level.

The aim of these methods is to fully utilize the healthy resources in a router to mitigate faults. In the proposed DBS method, there is a threat of deadlocks. This can be handled by using the proposed recovery scheme. These methods result in high packet acceptance rate and lower latency when compared to other routing schemes.

FAULTS IN A NOC:

A NoC is an on-chip communication infrastructure that implements multi-hop and predominantly packet-switched communication. Through pipelined packet transmission, NoCs permit a more efficient utilization of communication resources than traditional on-chip buses. Regular NoC structures reduce VLSI layout complexity compared to custom routed wires [5].

In future chip generations, faults will appear with increasing probability due to the susceptibility of shrinking feature sizes to process variability, age-related degradation, crosstalk, and single-event upsets. To sustain chip production yield and reliable operation, very large numbers of faults will have to be tolerated [3][10].

Forecasts for future technologies suggest that component failure rates will be much higher than 0.1%. If computers are to benefit from future advances in technology then there lie major

challenges ahead, involving understand how to build reliable systems on increasingly unreliable technology and how to exploit parallelism increasingly effectively, not only to improve performance, but also to mask the consequences of component failure.

Rapid development in silicon technology is enabling the chips to accommodate billions of transistors. It has been observed however, that the current on-chip interconnects –buses – are becoming a bottleneck as they are unable to cope with growing number of participating cores on a chip.

Shrinking silicon die size will lead to enhanced levels of cross talks, high field effects and critical leakage currents which, in turn, will lead to more temporary and permanent errors on-chip.

Crash or permanent failures can occur due to electromigration of a conductor or a connection failure permanently halting the operation of some modules. On the other hand, faults like Gaussian noise on a channel and alpha particles strikes on memory and logic can cause one or more bits to be in error but do not cause permanent failures. Firstly, transient faults can corrupt individual packets causing them to be mis-routed or invalid, in which case a retransmission is required. Secondly, due to electromigration, cracks, or dielectric breakdowns, links and/or routers become permanently unavailable causing them to stop functioning.

Transient:

Transient failures can occur on a chip for many reasons: alpha particles emitted by trace uranium and thorium impurities in packages and high-energy neutrons from cosmic radiations can cause soft errors in semi-conductor devices. Similarly, low energy cosmic neutrons interacting with isotope boron-10 can cause soft errors.

Permanent:

Permanent faults, as the name shows, cause permanent damage to the circuit. These faults cause physical changes in the circuits whose behavior does not change with time. Electromigration of a conductor, broken wires, dielectric breakdowns etc. are a few examples of permanent failures on chip. Permanent faults, at one level, are modeled as stuck-at faults, or as fail-stop model where a complete module malfunctions and informs its neighbors about its out-of-order status. NoCs, hence we consider a permanent failure as a fail-stop fault where a module after having permanent damage shuts itself down and inform the neighbors about it.

EXISTING SYSTEMS:

1) The faulty router is treated as a node fault and the routing algorithm can be dynamically reconfigured to re-route the packets following the contour around the faulty nodes. This region-

based routing algorithm is only suitable for one-faulty-router topology. Also, the node fault model may cause the faulty router totally isolated from the network, while in real situation, the faults may only affect a certain part of the router and the rest of it can still function correctly.

Reference:

Zhen Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip," in Proc DAC, 2008, pp. 441 -446.

2) After the built-in-self-test procedure and a static port swapping, the remaining faults (both link faults and unswapped buffer/MUX faults) are all treated as link level hard failures

Reference:

D. Fick, A. DeOrion, Jin Hu, V. Bertacco, D. Blaauw, and D. Sylvester, "Vicis: A reliable network for unreliable silicon," in Proc DAC, 2009, pp. 812 - 817.

D. Fick, A. DeOrion, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant nocs," in Proc DATE, 2009, pp. 21 - 26.

3) A resilient routing algorithm is designed to reconfigure the routing table in an offline process so as to bypass these fault links. Both the port swapping operation and routing table reconfiguration in are static in nature.

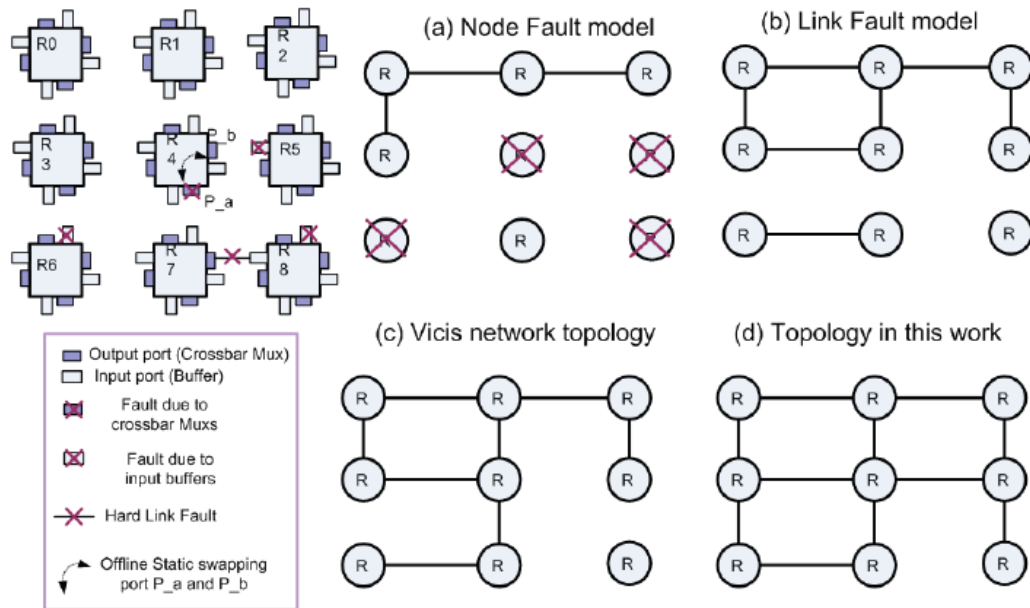
Different from that, our DBS and DMS operation will dynamically share the healthy resources among different ports in run time and hence improve the performance. A deflection routing algorithm is proposed to dynamically re-route the packets towards the destinations.

Reference: A. Hosseini, T. Ragheb, and Y. Massoud, "A fault-aware dynamic routing algorithm for on-chip networks," in Proc ISCAS, 2008, pp. 2653 - 2656.

PROPOSED SYSTEM:

A faulty router with an appropriate reconfiguration strategy may be able to tolerate the buffer or crossbar errors and still support the full connectivity in the NoC with reduced capacity. The previously suggested fault tolerant models such as the static offline port swap model, generic node or link fault model result in the network being partitioned. Hence the dynamic swapping scheme is suggested which shares the healthy resources in the router for different ports at run time.

Fault models:



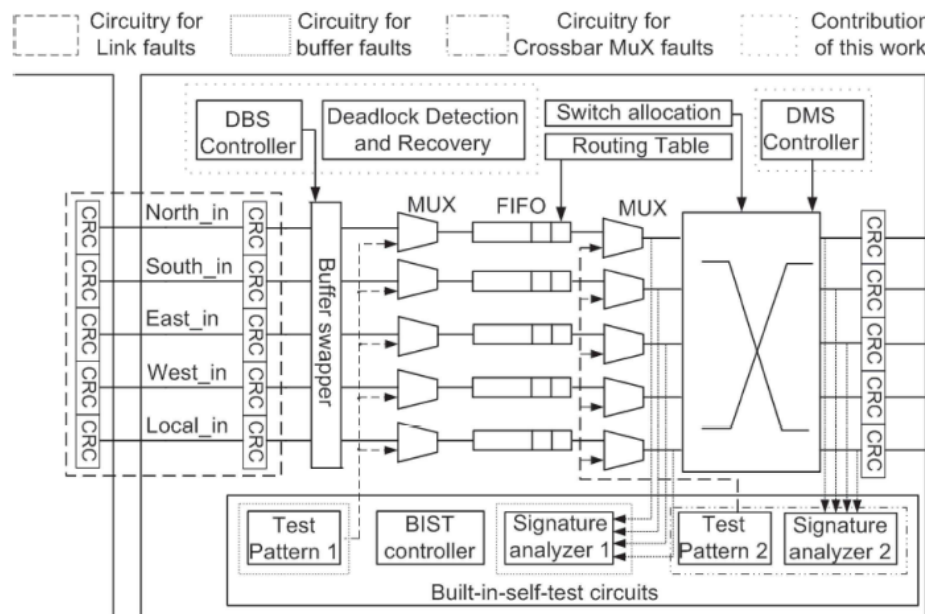
By leveraging the other healthy buffers or multiplexers (Muxes) and dynamically re-allocating the resources, the network connectivity can remain intact which result in better performance in terms of packet acceptance rate. Routers working in the DBS mode may face potential deadlocks. These deadlocks can be prevented using the efficient deadlock recovery scheme proposed.

Faults In routers can be detected with built-in-self-test (BIST) and self-diagnose circuits. Since the input buffers and the crossbar occupy the majority of the router area, most of the faults in the router may occur there. A hybrid fine-grained fault model is proposed to distinguish the nature and consequence of faults that have occurred in the NoCs and suggest fault-diagnosis technique to determine the type of fault. Following this, the DBS and DMS techniques are introduces. These dynamically reconfigure the routers to handle buffer faults and crossbar faults while still maintaining the maximum service of the router.

FAULT DIAGNOSIS STRUCTURE:

This architecture is used to determine the occurrence, type and position of the faults. The packet correctness is verified by the cyclic redundancy check module associated with each input port. The link transmission errors are characterized by a checksum mismatch. Switch-to-switch transmission error is detected by the BIST controller in the upstream router which will start the link diagnosis by sending the test patterns to the downstream neighbor and find out the specific link wires that have faults. Similarly, the intra-router error will be discovered if the packets pass

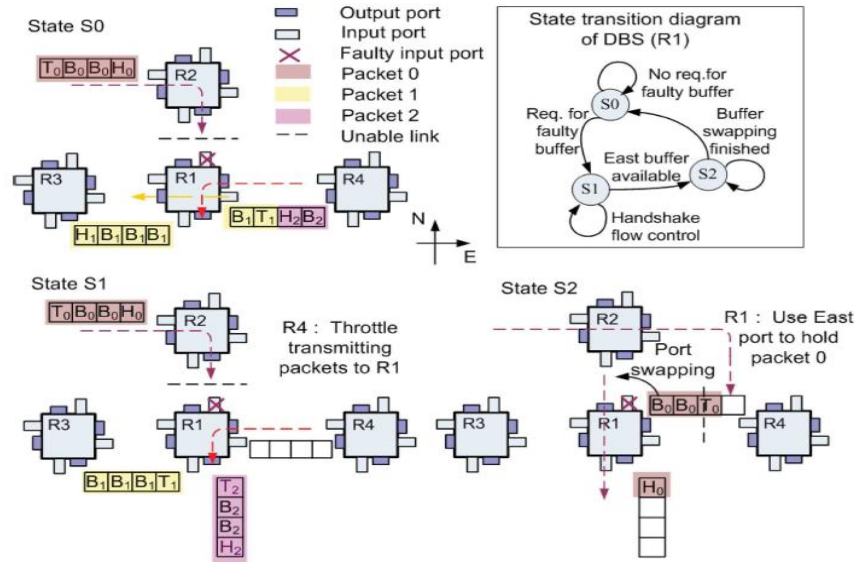
the CRC check at the input of the router but fail at the output side. On the detection of a fault, the router switches to a test mode to diagnose [4].



DYNAMIC BUFFER SWAPPING (DBS):

Dynamic buffer swapping (DBS) algorithm is proposed to handle the input buffer faults [1]. Through proper swapping of another substitute buffer to hold the packets destined for the faulty buffer, graceful performance degradation can be achieved by fully utilizing the link [2]. By adopting DBS algorithm, no extra buffers or virtual channels are needed in the router, which satisfies the stringent area constraint of NoC.

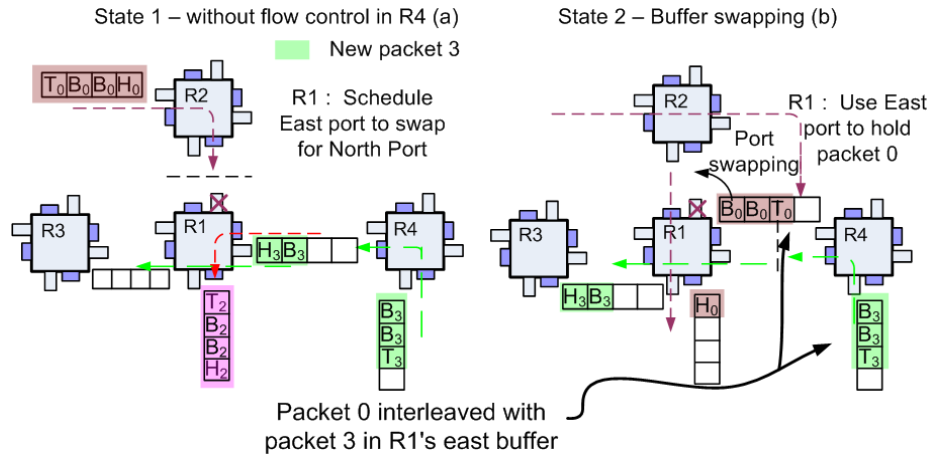
A healthy port must be used to replace the faulty ports. A buffer swapper is required to match the physical link connections with the swapped healthy FIFO input. A round-robin DBS scheduling can be adopted by dynamically using different buffers to swap with the faulty port so to even out the bandwidth penalty for each input. This creates a more area-efficient implementation.



Avoid packet interleaving:

Interference of packets from 2 routers is called packet interleaving. Two handshake signals, current transmit status (CTS) and current port status (CPS) for each port to avoid this situation. The router can send the packets across the link only when it receives a '1' in the CTS.

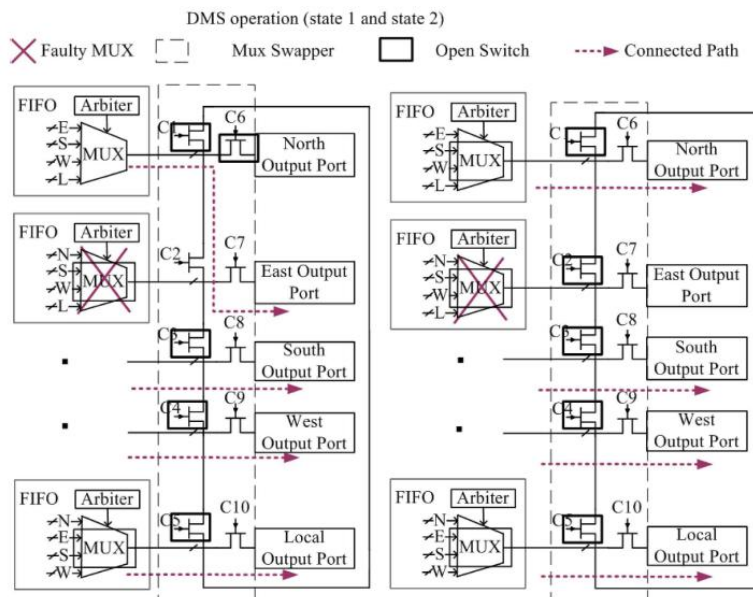
States	Signals status (North port of R1 is faulty)	Operations / representation on tokens
S0	R1 North writes: CPS "10"; CTS "0" R1 Reg: FP_R: "N"; SP_R: "φ"; DP_R: "N"	R2 blocks packet 0 R1 north port is faulty
S1	R1 North writes: CPS "10"; CTS "0" R1 East writes: CPS "01"; CTS "1" R1 Reg: FP_R "N"; SP_R: "E"; DP_R: "N"	R2 blocks packet 0 R4 monitors request to the west R1 prepares east to substitute north
S2	R1 North writes: CPS "00"; CTS "1" R1 East writes: CPS "10"; CTS "0" R1 Reg: FP_R "N"; SP_R: "E"; DP_R: "E"	R2 transmits packet 0 to east R4 blocks transmission to R1 R1 east port buffer is used for north link



DYNAMIC MUX SWAPPING (DMS)

The crossbar in the router is usually implemented in the form of five 4×1 MUXs. If one of the MUX malfunctions, the data will not be presented to the output link correctly. The ring topology allows each MUX to be shared among its neighboring output ports [1].

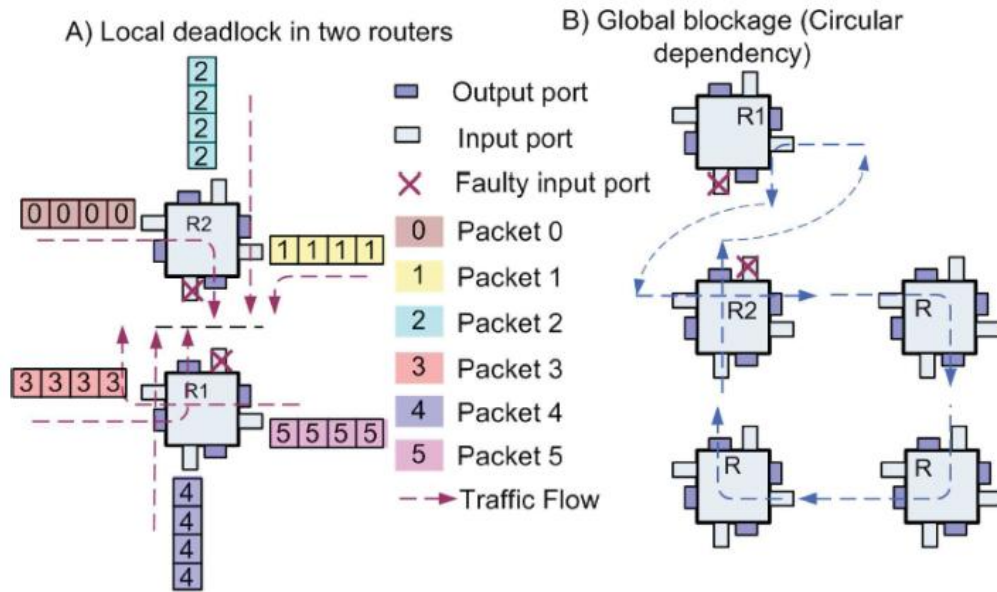
For the link wire faults, since the physical connectivity between the routers has been damaged, a robust routing algorithm is used to bypass the faulty links. This is called handling hard link faults.



DEADLOCK ISSUE AND HANDLING STRATEGY:

To ensure deadlock free routing, in normal operation, Odd-even (OE) routing algorithm is used in this framework [21]. When faults occur, due to the DBS or the re-routing, a deadlock may be introduced. These faults can be handled using an efficient deadlock recovery strategy.

When the router receives the header flit of a packet, an embedded counter is triggered to count the number of cycles it stays in the buffer. The counter continues to increment until the tail flit has been successfully transmitted or its value exceeds a predefined threshold [14] [22].



In DBS, A local deadlock will occur between the two nodes R1 and R2 if the south port of R2 and north port of R1 are faulty and all the packets in other ports choose these two ports as outputs. In this case, R1 (R2) will be stuck at state S1 waiting for the original packets in the substitute port of R2 (R1) being completely transmitted. If the scheduler in R2 arranges its west input buffer to hold packets from north and the scheduler in R1 arranges its east input buffer to substitute the faulty south buffer, a circular dependency on buffer resources can thus be generated as in the figure. This might bring deadlocks to the system.

For DMS operation, we only time-share the MUX between two output ports and do not change the dependency of the buffers in the routing algorithm. Hence it will not create deadlock if the original routing algorithm is deadlock-free.

A set of strategies is proposed to relieve the deadlock situation. Packets in the faulty routers in DBS mode are mainly dealt with because DMS and the routing algorithm will not generate circular dependency. This breaks the original dependency and prevents blockage. If all the ports in the router are not available for swapping, we will reset the routing computation (RC), switch allocation (SA) information of these packets to request a different output port.

Another strategy is to use the leader election algorithm. Once the nodes in deadlock are detected, the arbiters' decisions of each node are saved. The node's output channels are reused for selecting the leader among those nodes. After the leader node being elected, all nodes reconstruct its previous arbiter decision and wait for the incoming leader packets together with a preemption indicator. This preemption indicator sent by the elected leader node will make the entire message bypass the deadlocked nodes up to its destination. When a buffer becomes available after this process, all the other nodes will leave their awaiting state to transmit its deadlocked messages. This is similar to the prioritization of messages in computer networks.

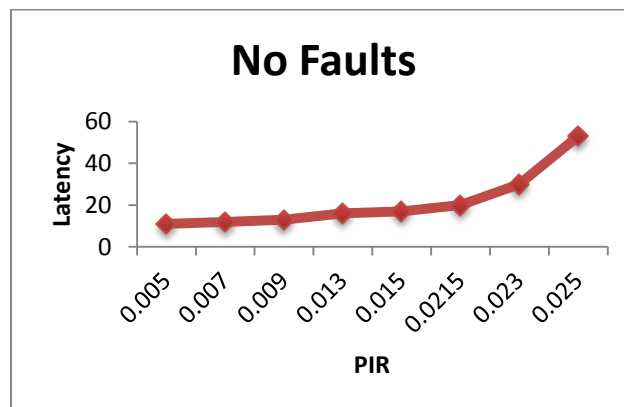
If the re-computation step fails to relieve the blockage, and the local processing element (PE) has available memory spaces in network interface (NI), next, we will send these blocked packets to the local PE, and make the PE re-send these packets to the destination. In order to avoid creating a new circular dependency between the PE and other routers, this operation is carried out only when NI has free memory slots.

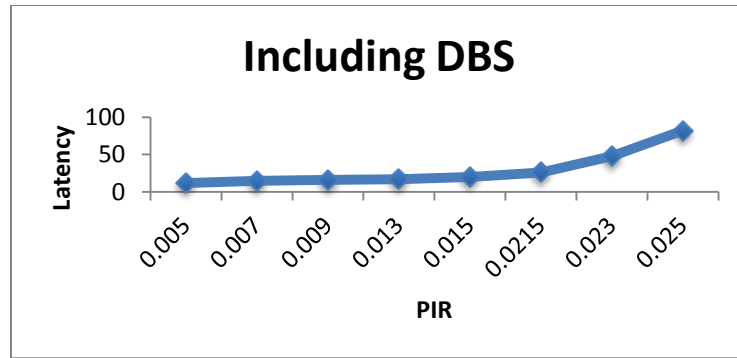
SIMULATION:

An NoC network was constructed using Noxim Simulator and systemC.

A single simulation topology of NoC using a 8 * 8 mesh network is shown. The router buffer size is 8 and packet size is (2,8). Simulation cycle is 50,000 with warm up cycle 20,000.

% noxim -dimx 8 -dimy 8 -dimz 4 -buffer 8 -size 2 8 -sim 50000 -warmup 20000 -pir 0.005





FUTURE WORK:

- Perform Analysis for Dynamic MUX swapping.
- Perform Analysis with the introduction of deadlocks.
- Compare DBS and DMS with respect to throughput by introducing faults into a NoC.

REFERENCE:

- [1] Zhiliang Qian, Ying-Fei Teh, and Chi-Ying Tsui, “A fault-tolerant network-on-chip design using dynamic reconfiguration of partial-faulty routing resources,” in VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on, Oct 2011, pp. 192–195.
- [2] A. Kohler, G. Schley, and M. Radetzki, “Fault tolerant network on chip switching with graceful performance degradation,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 29, no. 6, pp. 883–896, 2010.
- [3] D. Fick, A. DeOrio, Jin Hu, V. Bertacco, D. Blaauw, and D Sylvester, “Vicus: A reliable network for unreliable silicon,” in Design Automation Conference, 2009. DAC ’09. 46th ACM/IEEE, 2009
- [4] Shu-Yen Lin, Wen-Chung Shen, Chan-Cheng Hsu, Chih-Hao Chao, and An-Yeu Wu, “Fault-tolerant router with built-in self-test/self-diagnosis and fault-isolation circuits for 2d-mesh based chip multiprocessor systems,” in VLSI Design, Automation and Test, 2009. VLSI-DAT ’09.
- [5] L. Benini and G. De Micheli, “Networks on chips: a new soc paradigm,” Computer, vol. 35, no. 1, pp. 70 –78, jan 2002.
- [6] <http://www.noxim.org/>
- [7] <http://www.cse.iitd.ac.in/~panda/SYSTEMC/LangDocs/UserGuide20.pdf>

- [8] <http://ihome.ust.hk/~qianzl/Network%20on%20chip%20routing/a%20highly%20resilient%20routing%20algorithm.pdf>
- [9] <http://andrewdeorio.com/research/assets/DAC09Vicis-slides.pdf>
- [10] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, Nov. 2005.
- [11] S. Furber, "The future of computer technology and its implications for the computer industry," *Comput. J.*, vol. 51, no. 6, pp. 735–740, 2008
- [12] J. Hu and R. Marculescu, "Dyad: Smart routing for networks-on-chip," in *Proc. Design Autom. Conf. (DAC)*, 2004, pp. 260–263
- [13] A. Kohler and M. Radetzki, "Fault-tolerant architecture and deflection routing for degradable NoC switches," in *Proc. Symp. Netw. Chip (NOCS)*, 2009, pp. 22–31.
- [14] J. Wu and D. Wang, "Fault-tolerant and deadlock-free routing in 2-D meshes using rectilinear-monotone polygonal fault blocks," in *Proc. Int. Conf. Parallel Process.*, 2002, p. 247.
- [15] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2-D-mesh network-on-chip," in *Proc. Design Autom. Conf. (DAC)*, 2008, pp. 441–446.
- [16] M. Ali, M. Welzl, S. Hessler, and S. Hellebrand, "An efficient fault tolerant mechanism to deal with permanent and transient failures in a network on chip," *Int. J. High Performance Syst. Architecture*, vol. 1, no. 2, pp. 113–123, 2007.
- [17] Wu, M.S. and Lee, C.L. (2005) 'Using a periodic square wave test signal to detect cross talk faults', *IEEE Design and Test of Computers*, Vol. 22, No. 2, pp.160–169.
- [18] D. Fick, A. DeOrion, G. Chen, V. Bertacco, D Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant nocs," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09*, 2009, pp. 21–26.
- [19] Maurizio Palesi and Masoud Daneshtalab (Eds.), *Routing Algorithms in Networks-on-Chip*, Springer, 2014.
- [20] U.Y. Ogras, P. Bogdan, and R. Marculescu, "An analytical approach for network-on-chip performance analysis," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Trans. on*, vol. 29, no. 12, pp. 2001–2013, dec. 2010.
- [21] Anuradha K.shrimawale and Mahendra A Gaikwad. Article: "Review of Odd Even Routing Algorithm for 2D Mesh Topology of Network-on-Chip Architecture for Bursty Traffic." *IJCA Special Issue on Recent Trends in Engineering Technology RETRET:9-12*, March 2013
- [22] Verbeek, F. and Schmaltz, J., "A Decision Procedure for Deadlock-Free Routing in Wormhole Networks", *Parallel and Distributed Systems*, *IEEE Transactions on*, Volume:25 Issue:8, pp.1935 – 1944, 2014