NoC Reconfiguration for Utilizing the Largest Fault-free Connected Sub-structure

Armin Alaghi, Mahshid Sedghi, Naghmeh Karimi and Zainalabedin Navabi
Electrical and Computer Engineering Department
University of Tehran, Tehran, Iran

Abstract

This paper proposes an offline test strategy for finding the largest fault-free connected sub-structure of a mesh-based NoC. Faulty switch ports are found by flooding the NoC with test packets. Then, NoC routers are reconfigured according to the degraded NoC structure to route incoming packets.

1. Introduction

The fast paced growing of the complexity and shrinking feature size resulting in significant decrease of the manufacturing yield of the recent integrated circuits forces the designers to utilize fault-tolerant techniques and graceful degradable architectures. In this paper, we propose a test strategy and a reconfiguration method to make a faulty mesh-based NoC architecture degradable.

For the test phase, we used a simplified version of a system-level fault model proposed by Bengtsson et al. [1]. This model includes two types of port faults; faults that drop the packets (*Dropped*) and faults that corrupt the packets' contents (*Corrupt*). Our proposed strategy is able to detect and locate multiple faults in a NoC.

To evaluate our test strategy and re-routing algorithm, we developed a platform for our simulations at the Transaction Level. Transaction-Level Modeling (TLM) is a promising solution to improve simulation performance and modeling efficiency for early design space exploration [2].

2. Discussion

Our TLM NoC model is a 2-D mesh of switches defined as SystemC *module*s. The inputs and outputs of our NoC are placed at PI and PO nodes relatively. The router of the switch is a SystemC *thread* inside the switch. The strategy presented here locates *Dropped* and *Corrupt* faults in ports. We assume a small fault-free router inside the switch responsible for routing packets during the test mode.

For fault injection, a FIFO of length one is placed inside each port. Upon arrival of a packet, it is *put* into the FIFO and a controller decides to throw it away or just corrupt it according to the type of the fault.

At the start of the test session, the NoC PI generates a test packet and sends a copy of the packet to all of its neighbors. This is called flooding. Upon receiving a test packet, each switch sends an acknowledgement (*ack*) to the port from which the test packet arrived and floods it. Figure 1 shows the first three steps of the testing phase.

Each switch keeps the status of all of its ports. All links are suspected to be faulty at the beginning, and only when an *ack* is received from an input port, and its parity check succeeds, the switch marks the port as fault-free. A failure in parity check implies existence of a *Corrupt* fault where a failure in reception of the *ack* indicates a *Dropped* fault.

After detecting the faulty ports, each switch gets the information about its neighboring ports and reconfigures its own router with a simple local algorithm. This routing algorithm keeps no history of the received packets and routes each packet with only the local information available using simple rules. Experimental results show that the proposed routing algorithm delivers 97% of the possible packets to their destination successfully. A simple XY-algorithm delivers only 64% of the packets. The set of the usable switches of the degraded NoC can be extracted by running a simple iterative graph *Reachability* algorithm.

3. References

[1] T. Bengtsson, S. Kumar, and Z. Peng, "Application Area Specific System Level Fault Models: A Case Study with a Simple NoC Switch", *Proceedings (electronic) Int. Design and Test Workshop (IDT)*, Dubai, UAE, 2006.

[2] D. Shin, A. Gerstlauer, J. Peng, R. Dömer, and D. D. Gajski, "Automatic Generation of Transaction-Level Models for Rapid Design Space Exploration", *Proceedings Int. Conference on Hardware/Software Codesign and System Synthesis*, Seoul, Korea, 2006, pp. 64-69.

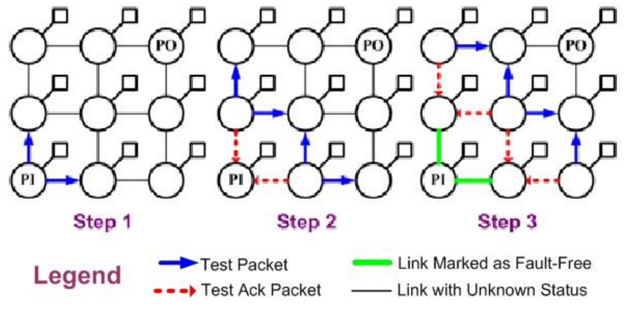

Figure 1. The first three steps of the test phase

1-4244-4203-0/08/$20.00 ©2008 IEEE