# Utilizing NoC Switches as BIST-structures in 2D Mesh Network-on-Chip

**Kim Petersén, Johnny Öberg**

*Royal Institute of Technology (KTH), School of Information and Communication Technology (ICT), Dept. of Electronic, Computer and Software Systems (ECS), Email: {kimp, johnny}@imit.kth.se*

## Abstract[1]

This poster proposes a test methodology and presents a method for carrying out an automatic go/no-go BIST operation at start up of a 2D-mesh NoC Network. It executes in functional mode at full clock speed. Only minor area penalty is introduced in the NoC-network itself; the BIST is placed in the Network-Interface inside the computational resources.

### 1. Introduction

The Nostrum NoC from KTH is a packet switched Network-on-Chip (NOC) architecture designed to be scalable from a few dozens to several hundreds or even thousands of heterogeneous resources of different kinds, for instance processor cores, DSP cores, FPGA blocks, dedicated HW blocks, memory blocks, etc. To be able to structurally test such a vast and complex chip in a reasonable amount of time, a test methodology for NoCs has to be designed that heavily relies on the usage of Built-In-Self-Test (BIST) structures. This poster outlines a test methodology for carrying out an automatic BIST operation at start-up of a 2-D Mesh NoC. The methodology is general enough to be portable to any NoC structure.

### 2. Structure to explore

The Nostrum NoC Backbone consists of Resources (R) and Switches (S) organised in a 2D Mesh Manhattan-like structure. A switch is connected to a single one resource in a one-to-one correspondence. The Resources are logic modules that implement various functionality. The signalling between any two Switches and between Switch and Resource is packet based. Each packet is 128 bits wide and consists of Header (32 bits) and User Data (96 bits). The Switches makes an independent routing decision for each individual packet in a single clock cycle. No internal buffering in the switch is used.

All Resources are equipped with a Network Interface (NI) to communicate between the Resource core and the Network. The NI handles the incoming packets from the Switch. A Resource-Network-Interface (RNI) is included for adapting the network to various different local communication protocol standards like, for instance, the AMBA-bus.

### 3. Start-up test methodology

The boot strap start up test sequence for the Nostrum NoC goes through the following phases: 1) Reset; 2) The BIST sequence start; 3) The NIs log results of its own Switch-Link test; 4a) The (Test) Operating System ((T)OS) Node starts collect test result from all nodes; 4b) NIs wait for initiation from the node containing the (T)OS-node; 5a) NIs respond to the (T)OS request by sending its test log; 5b) the OS-node collect results from the NIs; 6) Normal Operation.

After the BIST sequence has been completed, the (T)OS-node starts booting the system by sending out a query to each node. This query function as a time marker to synchronise the NI's and mark which of the clock cycle windows that should function as a control channel. Each NI answers with its test log. The (T)OS starts querying its neighbourhood, and move its way outward detecting and circumventing broken links on the way. In case all four links out from the (T)OS-node is broken, i.e., the (T)OS-node is not reachable, the chip is considered broken beyond repair.

### 4. Results

The BIST sequence is comprised of two main steps, called phase 1 and 2. During phase 1 the Switches themselves work as BIST engines; test packets are sent into the switches by choosing an appropriate register reset value. The NIs collects the test result when packets arrive. During phase 2, the NI sends values to the Switch and collects test data.

The total length of the BIST execution was varied to see how fast a good test coverage could be achieved:

| | Whole Switch | Datapath |
|---|---|---|
| - After 20 clock cycles | 46% | 68% |
| - After 100 clock cycles | 67% | 87% |
| - After 213 clock cycles | 80% | 97% |

To improve the controllability of the datapath and the observability of the controller, five 2to1-muxes were added to the switch. This makes it possible to achieve a reasonably good fault-coverage very fast for the datapath. A deeper analysis of the controller in the switch and its testpatterns needs to be performed in order to achieve a better total fault coverage. This will be done in the future.

For the links, no extra test logic is added. The switch is executed in functional mode. For the NIs, test logic is added, i.e. a signature register and some logic to keep track of test time and packets to be loaded during phase 2. The NI is running in test mode during execution of the BIST. The result from the BIST is saved as a signature, collected by the NI.

Phase 1 and 2 create a fast go/ no-go test of the NoC. A repair functionality can be added by introducing a phase 3 of the BIST. Phase 1 and 2 can be used to decide if a phase 3 is needed or not and can be triggered by the (T)OS when needed. The presented BIST assumes that reset can be released in the same clock cycle for all switches and NIs.

### 5. Conclusion and Future Work

The proposed BIST methodology enables a fast go/no-go BIST, with minor extra area in the NoC itself. A test coverage of 97% was obtained for the datapath, and 80% for the whole switch, in only 213 clock cycles.

1.This work is supported by Industrial Research School in Electronic Design.