A New Fault Injection Approach for Testing Network-on-Chips

L. Sterpone, D. Sabena, M. Sonza Reorda
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, ITALY
Motivations and Goals

• Modern generation of Multi-processor Systems-on-Chip (MP-SoCs) are increasingly sensitive to faults
  – Shrinking technology
  – Lower voltage margins

• There is the need of accurate test solutions
  – Effective fault model
  – Fast elaboration

• To provide a method for effectively evaluate the fault tolerance capability of Networks on Chip (NoCs)
Outline

• Introduction
• NoC general architecture
• Routing and logic fault model
• The fault injection method
• Experimental results
• Conclusions
• Future works
Introduction

• NoCs are characterized by high performances and low power consumption
• Faults evaluation is an open problem, increasingly affecting NoC capabilities
• Technology scaling results in an increase fault sensitivity
  – Single Event Upsets
  – Cross-Talk
  – Age-related degradation
  – Process variability
Previous works

- Functional IP cores using **Test Access Mechanism (TAM)**
- **Specific fault model** for NoC fabrics
- Dedicated TAM on specific on-chip network is adopted by functional test solutions on SoCs multicore
NoC general architecture

• NoC architecture consists of a 2-D mesh
  – Routers
  – Set of interconnection resources

• NoCs differ by several factors
  – Routing algorithm
  – Switching
  – Flow control

• Communication schemes
  – Communication switching
  – Virtual cut-through
  – Wormhole switching
Router architecture

- FIFO buffers in the input and output data ports
- Crossbar switch controlled by logic circuitry
  - Transportation methodology
- Possible links
  - Same Row (SR)
  - Same Column (SC)
  - Other locations (OL)
- Crossbar switch is the router core
  - It performs links between input and output queues
  - Generally it can generate all possible combinations
Crossbar switch fault analysis on FPGA

FPGA configuration memory

Routing and Logic cells

Look-up tables (LUTs)
Flip – Flops (FFs)
Routing switch matrices
Crossbar switch fault analysis on FPGA

• NoC mapped on SRAM-based FPGAs
Crossbar switch fault analysis on FPGA

- NoC mapped on SRAM-based FPGAs

Fault affects crossbar switch functionality
## Routing fault model

- Detailed routing fault model

<table>
<thead>
<tr>
<th>Description</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original wire segments</td>
<td><img src="image1.png" alt="Original Wire Segments" /></td>
</tr>
<tr>
<td>Logic value on wires is inverted</td>
<td><img src="image2.png" alt="Logic Value Inverted" /></td>
</tr>
<tr>
<td>Wire forced to logic 0</td>
<td><img src="image3.png" alt="Wire Forced to Logic 0" /></td>
</tr>
<tr>
<td>Wires A forced to value of wire B</td>
<td><img src="image4.png" alt="Wires A Forced" /></td>
</tr>
<tr>
<td>Wire forced to a propagation Delay D</td>
<td><img src="image5.png" alt="Wire Forced to Delay" /></td>
</tr>
</tbody>
</table>
The fault injection method

• The architecture: implemented on single SRAM-based FPGA
The fault injection method

• The fault injection execution flow
Case study: NoCem

- Network on Chip emulator (NoCem)
  - Open source HDL model of a NoC
- NoCem used configuration
  - 32 bits dataword size
  - Packet length 16 datawords
  - Router FIFO buffer length of 8
  - Square grid configuration (4x4 2D mesh) 16 IP ports

<table>
<thead>
<tr>
<th></th>
<th>Routing Slices [#]</th>
<th>FF Slices [#]</th>
<th>LUT Slices [#]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoCem</td>
<td>2,612</td>
<td>4,236</td>
<td>7,369</td>
</tr>
<tr>
<td>Test Environment</td>
<td>1,790</td>
<td>12,484</td>
<td>13,210</td>
</tr>
</tbody>
</table>
Experimental results

• Four fault injection campaigns
  – Average injection time of a single fault 12 µm
  – Total pattern applications and observation time 73 ms

• Stuck-at 0/1 fully compliant with Tetramax \textregistered commercial fault simulator
  – Fault simulator around 13 hours
  – Proposed platform 5.6 hours

<table>
<thead>
<tr>
<th></th>
<th>Injection Faults [#]</th>
<th>Detected Faults [#]</th>
<th>Injection Time [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire Faults</td>
<td>2,834</td>
<td>2,780</td>
<td>3.45</td>
</tr>
<tr>
<td>Stuck-at</td>
<td>278,430</td>
<td>261,278</td>
<td>338.8</td>
</tr>
<tr>
<td>Bridge</td>
<td>6,829</td>
<td>6,740</td>
<td>8.31</td>
</tr>
<tr>
<td>Delay</td>
<td>279,834</td>
<td>143,439</td>
<td>340.6</td>
</tr>
</tbody>
</table>
Conclusions

• A novel test approaches based on dual-processor system has been implemented on SRAM-based FPGAs

• Effectively applicable to mesh-based NoCs
  – Flexibility
  – Observability
  – Test Speed

• Experimental results demonstrate the feasibility of the proposed approach
  – Applicable to large scale NoCs
Future works

• Apply to large NoC models
• Extend emulation platform to analyze the effects of delay faults
• Investigate possible hardening solution
  – Fault Tolerance
  – Testability
Thank you...

Any questions?

Luca Sterpone

luca.sterpone@polito.it