Concepts for Robust NoC Communication

Martin Radetzki
Department of Embedded Systems Engineering
Institute of Computer Architecture and Computer Engineering
Universität Stuttgart

www.iti.uni-stuttgart.de/ese.phtml
martin.radetzki@informatik.uni-stuttgart.de
Future Challenge: Reliability

“Within a decade we will see 100 billion transistor chips. That is the good news. The bad news is that 20 billion of those transistors will fail in manufacture and a further 10 billion will fail in the first year of operation.”

Reliability Issues

Single Event Effects
- e.g. ion, neutron impact
- causing transient faults

Variability
- manufacturing-induced variations, e.g. in wire thickness, transistor parameters
- causing permanent and intermittent faults that must be tolerated for acceptable yield

Stress & Ageing
- e.g. Electromigration, HCI, NBTI
- causing permanent faults that appear during system operation

Environmental Impacts
- make everything worse...
- e.g. accelerated ageing under high temperatures, high levels of radiation on space missions
**Failure – Fault - Error**

**Terminology**

**Failure mechanism**
- Physical cause

**Fault (model)**
- Formal description

**Error**
- Effect on information

**Examples**
- Single-event effect
- Fabrication defect
- Chip ageing
- Transient fault
- Permanent fault (e.g. stuck-at)
- Delay fault
- Soft error
- Recurring data corruption
- NoC: Packet loss

Concepts for Robust NoC Communication
Steps for Fault-Tolerance

**Testing**

- Error detection

**Diagnosis** (model)

- Fault location
- Fault classification

**Correction**

- Methods depend on the class of fault

**Examples on various layers**

- **Offline**
  - DL: Syndrome ≠ 0
  - NL: Packet corruption
  - TL: Packet loss

- **Online**
  - DL: Analyze syndrome
  - NL: Observe the regular traffic
  - TL: Inject test packets

- **Correction**
  - DL: Toggle erroneous bits
  - NL: Adapt routing to bypass fault
  - TL: Re-send packet (ARQ)
Outline

- Fault Models and Fault Diagnosis
- Fault-Adaptive Routing
- Experimental (Simulation) Results
Distributed Error Detection and Diagnosis

e.g. [Grecu, Pande 2006]

- Parity checks at switch inputs and outputs
- Output error => diagnosis of switch fault
- Input error => diagnosis of switch fault
- Drop and retransmit packet in case of error
- Does not help against permanent faults
Traditional Fault Models

- Switch faults: the complete switch is assumed to be unavailable
- Link faults: refers to a link between two switches (and/or a directional output of a switch)
- Switch faults: Throughput breaks down even at low failure rates
Structural Fault Model

- One multiplexer, part of a crossbar switch:

- Structural defect causes a single connection to fail (E → W)
- Structural defect invalidates a complete link (→ W)
One approach [Raik, Ubar 2007]:

- Sending directed test patterns
- Assumption: Links are fault-free
- Diagnostic access at the network boundaries
- Central control, not distributed (scalability?)
- Regular NoC operation needs to be interrupted (offline)
Proposed Diagnostic Infrastructure

Nostrum switch

- Packet format extended by a CRC checksum:
  - Input: Detecting Inter-Switch-Faults
  - Output: Detecting Intra-Switch-Faults
- Model of individual crossbar connection faults
- Implemented on chip to store online diagnosis results
Outline

- Fault Models and Fault Diagnosis
- Fault-Adaptive Routing
- Experimental (Simulation) Results
Deflection Routing

- Packets are routed with a preferred direction, e.g. following XY route
- If that is not possible, the packet is deflected into another direction

- No blocking
- No buffers
- Free of deadlocks
- Non-minimal paths possible
- Livelock avoidance through priority for „old“ packets

S: Switch  R: Ressource  ø: Schleife  ---: Verbindungs paar
Implementing Deflection in Switches

1.) Choose input based on priority

2.) Select output according to deflection policy

N, E, S, W: directions to neighbour switches
L: connection with the local resource

(1) Routing in preferred direction (here: x towards destination)
(2) Routing in other direction towards destination
(3) Routing in other direction away from destination
(4) Reflection to the packet source of the previous hop
Routing Adaptation

- In order to find feasible routes in case of faults, all incoming packets have to be considered as a whole:

<table>
<thead>
<tr>
<th>packet</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>max (4)</td>
<td>high (3)</td>
<td>low (2)</td>
<td>min (1)</td>
</tr>
<tr>
<td>from</td>
<td>N</td>
<td>E</td>
<td>S</td>
<td>W</td>
</tr>
<tr>
<td>preferred route</td>
<td>S, W</td>
<td>N, W</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>veer away</td>
<td>E</td>
<td>S</td>
<td>W, E</td>
<td>N, S</td>
</tr>
<tr>
<td>Reflection</td>
<td>N</td>
<td>E</td>
<td>S</td>
<td>W</td>
</tr>
</tbody>
</table>

Diagram showing packet routing and preferred routes with nodes labeled P1, P2, P3, and P4.
Cost Driven Routing (1)

- Assign a cost to a routing decision for one packet:
  - Reflects the progress of the packet
  - Cost is weighted by priority

<table>
<thead>
<tr>
<th>packet</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>max (4)</td>
<td>high (3)</td>
<td>low (2)</td>
<td>min (1)</td>
</tr>
<tr>
<td>from</td>
<td>N</td>
<td>E</td>
<td>S</td>
<td>W</td>
</tr>
<tr>
<td>preferred route</td>
<td>S, W</td>
<td>N, W</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>veer away</td>
<td>E</td>
<td>S</td>
<td>W,E</td>
<td>N,S</td>
</tr>
<tr>
<td>reflection</td>
<td>N</td>
<td>E</td>
<td>S</td>
<td>W</td>
</tr>
</tbody>
</table>

\[
C_{ij} = \begin{cases} 
\infty & \text{if } f_{ij} > 001_2 \vee \overline{avl}_{ij}, \\
2 & \text{elif } i = j, \\
0 & \text{elsif } i \rightarrow j \text{ reduces } |P_i.addr|, \\
1 & \text{else} 
\end{cases}
\]

- connection unavailable
- reflection
- approach destination
- veer away
Cost Driven Routing (2)

- In order to consider all packets at once, build the sum of weighted costs

\[ c(\pi) = \sum_i r_{i\pi(i)} \]

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>E</th>
<th>S</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>6</td>
<td>(\infty)</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>W</td>
<td>1</td>
<td>(\infty)</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

- Objective: Find routing permutation \(\pi_{\text{opt}}\) with minimal cost sum in a combinational way
  - Construct permutation tree
  - Bottom-up traversal of the permutation tree
  - Use minimum operations to discard “bad” routes early

- \(c(\pi_{\text{bad}}) = 9\)
- \(c(\pi_{\text{good}}) = 4\)
Routing Implementation

Example:

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
</tr>
</tbody>
</table>

Concepts for Robust NoC Communication
Switch Area and Timing

<table>
<thead>
<tr>
<th>Area $[\mu m^2]$</th>
<th>Standard Nostrum switch</th>
<th>Fault-tolerant version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13570</td>
<td>16790.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max. operating frequency [MHz]</th>
<th>Standard Nostrum switch</th>
<th>Fault-tolerant version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>106.84</td>
<td>77.4</td>
</tr>
</tbody>
</table>
Outline

- Fault Models and Fault Diagnosis
- Fault-Adaptive Routing
- Experimental (Simulation) Results
Simulation Setup

- 8x8 mesh with uniformly distributed and transposed data traffic
- Incremental injection of faults according to a fixed, randomly selected fault pattern
- Measurement of latency and throughput
- Comparison of three models:
  - Diagnosis and adaptation as presented
  - Automated Repeat Request (ARQ) at receiving end
  - Automated Repeat Request (ARQ) on switch level
Fault-Free Case

- Average network latency reduced by ~12%

Injection rate [Packets per Core and Cycle]

<table>
<thead>
<tr>
<th>Hop count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.15</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.25</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.35</td>
</tr>
</tbody>
</table>

- Classic
- Cost based
Packet Throughput with Single Crossbar Faults

- Minor throughput reduction even at large failure rates
- Fault duration has no significant impact
Routing around faults is better than accepting faults and performing retransmission
Conclusions

- Routing method using information on NoC permanent fault status

- Distributed implementation (scalable)

- Can be implemented with acceptable cost

- Current research: combined with retransmission to handle transient faults

- Future work: research optimal combination of techniques on different layers
## Hierarchical Approach to Fault Tolerance

<table>
<thead>
<tr>
<th>Fault Models</th>
<th>Monitoring</th>
<th>Diagnosis</th>
<th>Reconfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport</strong></td>
<td>Packets</td>
<td>Congestion</td>
<td>Packet Loss</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
<td>Packet Loss</td>
<td>Error</td>
</tr>
<tr>
<td>Distributed Online-Algorithms, Regular Structures</td>
<td></td>
<td>Packet-Fault</td>
<td></td>
</tr>
<tr>
<td><strong>Data Link</strong></td>
<td>Word-stream</td>
<td>Bandwidth</td>
<td>Fault</td>
</tr>
<tr>
<td>Encoding (Logic)</td>
<td></td>
<td>Fault Rate</td>
<td></td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td>Bitstream</td>
<td>Fault Rate</td>
<td>Defect</td>
</tr>
<tr>
<td>Encoding (electrical), Drivers, Monitoring</td>
<td></td>
<td>Electr. Param.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Temperature</td>
<td></td>
</tr>
</tbody>
</table>

### Error Detection
- Bit-Faults
- Delay Faults
- Voltage-, Current-, Temperature-Monitoring

### Error Correction
- Adjust Voltage, Current
- Use multiple lines
- Spare-Lines

---

**Fault Models**
- Fault
- Monitoring
- Diagnosis
- Reconfiguration

**Transport**
- Packets
- Congestion
- Packet Loss
- Packet-Fault

**Network**
- Word-stream
- Bandwidth
- Fault Rate

**Data Link**
- Bitstream
- Fault Rate
- Electr. Param.
- Temperature

**Physical**
- Bitstream
- Fault Rate
- Electr. Param.
- Temperature
Parameter Interaction

Performance

- Delay faults
- Correction overhead

Error rate

- Electromigration
- NBTI, HCI, ...
- Temperature
- LC-Coupling
- Radiation (SEU)
- Noise / SNR

Power

- Switching activity
- Frequency
- Supply voltage

Concepts for Robust NoC Communication