FMEA-Based Analysis of Networks-on-Chip for Mixed-Critical Systems

Eberle A. Rambo, Alexander Tschiene, Jonas Diemer, Leonie Ahrendts, Rolf Ernst
29.10.2014
Introduction

Multi- and many-core processors
- Adopted by server and consumer electronics markets
  - Servers, laptops, digital media players, smartphones, …
- Now being evaluated for embedded markets

Real-time mixed-critical systems: e.g. CERTAINTY, ARAMiS projects
- Automotive: Real-time Image Processing
- Aviation: Flight Management Systems (FMS)

Safety Standards
- IEC 61508
- ISO 26262: Automotive domain
- DO-254: Aviation domain

- Safety requirements
- Certification process
Introduction

Safety Standards: requirements
- Task isolation
  - No error propagation
  - No violation of guarantees
- Fault tolerance
  - Transient faults
    Single Event Upsets (SEUs): electro-magnetic radiation → bit-flip

The Network-on-Chip
- Central part of the system: must be certified
- Concentrates all communication: essential
- SEUs: how can the requirements be violated
A Network-on-Chip for Mixed-Critical Systems

- 2D mesh topology
- Nodes: switch + up to 4 tiles
- Switch: up to 8 ports
- XY source routing
- Credit-based flow control
- Wormhole switching
  - Variable packet length: 1..n flits
- Virtual channels: distinct traffic classes
  - Best effort
  - Guaranteed service (throughput)

No fault-tolerance mechanisms implemented
Motivation

- **Transient faults** (SEUs) in the NoC
  - Why worry? Just retransmit the packet
  - What if it causes a static effect?

- Mixed-critical systems → general systems

- **Transient fault**: what exactly happens in the NoC?
  - What are the effects?
  - How long do they last?
Existing approaches

- Off-chip networks: reliability studies [5]
  - Usually aim at increasing availability of the network
  - Different constraints: can not be directly applied to NoCs

- On-chip networks: many studies on fault-tolerance
  - Fault-tolerance approaches [12]-[16]
  - Surveys [6],[7]
  - Trade-off analyses [17]-[19]
  - Address specific cases: e.g. packet corruption or loss
    - No guarantee that all possible errors are covered
  - Focus on the links
  - Transient faults = retransmission mechanisms
  - Switches?
Our approach

FMEA-based analysis

✓ Uncover all potential errors caused by transient faults (SEUs) and their effects

✓ Error characterization
  ✓ Duration of the effects
  ✓ Isolation violation

✓ Relative occurrence probabilities of failures and effects
Failure Mode and Effects Analysis (FMEA)

- Powerful, thorough bottom-up analysis
- Systematically captures all potential errors and their effects

Analysis:
- For each component type
  - For each instance
    - For each failure mode (bit-flip)
      - For each state of system
        - Analyze local effects
        - Analyze global effects

- Analysis effort grows fast
Outline

- Introduction & motivation
- A Network-on-Chip for mixed-critical systems
- FMEA-based analysis
- Methodology
- Results
- Transient fault: static effects
- Conclusion
Methodology

Keeping the analysis effort acceptable:

- Error abstraction
- Exploiting symmetries
- Worst-case effect on a test packet
Error abstraction

- Signals are grouped logically by function
  - Conservative: every single-bit error has immediate effect

- Errors are equivalent if the effects are identical

- Errors are considered on a block level
  - Conservative: errors inside a block always show up on the interface

- We need to be conservative.
  But is it too conservative?
Exploiting symmetries

- Identical behavior of input and output ports in a switch
  - Analyzing one path is enough

- Identical behavior of the switches in the NoC
  - Minimal network configuration
    - XY source routing
      1. Straight: packet continues in the same direction, i.e. a “run”
      2. Turn: packet turns right or left
      3. Exit: packet goes to an up-link connected to a NI

- All behaviors are seen in a 2x2 mesh
- Larger NoCs will present repeated behaviors
Errors have a multitude of different consequences
- Depending on the current state of a switch

State of the switch is defined by:
- Packets being transferred
- State of those transfers

Analyze the transmission of a test packet
- Different compositions (e.g. 1 or 3 flits)
- Effects on the packet and on background traffic
  - Symmetry
Effects characterization

- **Duration**
  - **Transient:** effect vanishes with the affected packet
    E.g. packet payload corruption
  - **Degraded:** effect degrades performance, accumulates and manifests
    E.g. credit counter does not increment
  - **Intermittent:** effect remains but can eventually vanish
    E.g. virtual channel blocking
  - **Static:** effect remains, affecting subsequent transmissions
    E.g. virtual channel blocking

- **Task isolation**
  - **Isolation violation:** error affects the transmission of other streams
    E.g. causing packet corruption or loss
  - Can also be **static**
Results of the FMEA

- 161 different errors (failure modes)
  - Switch: 107
  - Link: 54

The switch can not be treated as a black box!

How important are the errors?
Probabilities

Technical report available with detailed results

Switch: block diagram

Switch Fabric

Input Buffer for Input Port 0

VC 0

VC 7

Input Port 0

Link 0

Output Port 0

Link 0

Output Port 7

Link 7

Input Buffer for Input Port 7

VC 0

VC 7

Input Port 7

Link 7

Output Port 7

Link 7

VC Access Controller

Credit Counter

Switch Arbiter

Register Bank

RB Port

Credits

Credits
Relative probabilities from a real NoC implementation

The NoC was implemented in VHDL

- Prototype with HAPS-62
  - 2 Virtex 6 FPGAs

Synthesis results for 2x2 mesh

- Switch S1
  - Register usage: 3886
  - Look-up tables usage: 24628
  - + 3 Block RAM: 3360 bits (ECC)

Register usage in the switch:

- Input Buffers 46.55%
- Register Bank 39.94%
- VC Controllers 1.65%
- Switch Fabric 0.54%
- Switch Arbiters 3.09%
- Credit Counters 8.23%

Probability of an error is relative to the number of registers
Global effects: relative probability

- QoS violation
- Packet loss
- Packet corruption
- Return route corruption
- VC buffer blockage

Transients faults resulting in static effects!

- CAUSE: Virtual Channel Controller unable to handle incomplete packets
Virtual Channel Controller

- Wormhole switching
Virtual Channel Controller

- Handling a packet without Head Flit

![Diagram of Switch with Virtual Channel Controller](image)
Virtual Channel Controller

- Handling a **packet without Tail Flit**

Sources of flit loss:
- Link
- Credit Counter
- Switch Arbiter
- Input Buffer
...
Preventing static effects

- Virtual Channel Controller: handling incomplete packets
  - Improving the state machine

![Diagram showing state transitions and packet handling](image-url)
Global effects: relative probability

- Baseline (B) and Improved (I)

- QoS violation
  - Baseline (B)
  - Improved (I)

- Packet loss
  - Baseline (B)
  - Improved (I)

- Packet corruption
  - Baseline (B)
  - Improved (I)

- Return route corruption
  - Baseline (B)
  - Improved (I)

- VC buffer blockage
  - Baseline (B)
  - Improved (I)

[Diagram showing relative probability for different global effects and conditions: Static, Degrading, Intermittent, Transient, Isolation violation]
Conclusion

FMEA-based analysis

- Compliant with certification processes of Safety Standards

- The switch is more susceptible to errors

- Many cases where transient faults cause static effects

- Static effects block the network: prevent them

- Isolation violation and undesired effects have to be addressed separately