Fault Detection, Isolation and Recovery Schemes for Spaceborne Reconfigurable FPGA-Based Systems

Felix Siegle, PhD

Supervisor: Prof Tanya Vladimirova
Department of Engineering, University of Leicester, UK

ESA Support: J. Ilstad, M. Suess, C. Poivey

Airbus Support: O. Emam

Sponsorship from ESA under the NPI Programme, Airbus Defence and Space, UK and the University of Leicester.
Outline

• Literature Survey
• FDIR Hardware Framework: Distributed Failure Detection
• Availability Analysis Method
• Proton Irradiation Test Campaign
• Publications
Literature Survey

- No FDIR scheme exists that specifically targets multi-FPGA systems.
- No availability analysis method exists for such an FDIR scheme, which also takes Block RAMs into account.
- Is on its own a novel contribution to knowledge.
- Together with the included design recommendations, it can serve as a tutorial for both scientists and engineers who are novices in this field.
Distributed Failure Detection

- Data is independently processed by “stream processors”.
- Data is provided via a Network-on-Chip (NoC)
- Based on modular redundancy + voting/comparison:
- Allows a real FDIR approach.
- Allows the distribution of redundant stream processors over several FPGAs.
- Redundant processors can be added / removed during operation, depending on the criticality of the current mission phase.
Distributed Failure Detection

- Intercommunication is done via a switched fabric NoC architecture.
- Failure detection and isolation mechanisms are embedded into NoC switches.
- Compared to the state of the art, this approach scales much better with the size of the application.
- Can easily be applied to multi-FPGA systems.
- Is well suited for high-performance payload data processing.
Distributed Failure Detection

Example application and network topology:
Availability Analysis Method

Question: Which failure rates do we have to expect for a stream processor in a specific FDIR configuration in a specific radiation environment?
Availability Analysis Method

Step 1: SEU Rate Calculation / Determination of MTBF

- Cross-sections (as provided e.g. from Xilinx/NASA) can be used to calculate the failure rates for a particular stream processor design in a particular orbit / radiation environment.

- Tools like OMERE simplify the computation according to ECSS standards.
Availability Analysis Method

Step 2: Block RAM Profiling

- Main novelty: Block RAM profiling tool.

- Allows a much better estimate of the number of susceptible Block RAM bits.

- It therefore increases the overall prediction precision of MTBF and availability figures.
Availability Analysis Method

Step 3: Fault Injection Experiments

- Second novelty: Fault injection algorithm.

- Can classify sensitive bits depending on how a system can recover from failures triggered by upsets in these bits.

- Allows more advanced availability models.
Availability Analysis Method

Random fault injection can provide accurate results
Availability Analysis Method

Random fault injection can provide accurate results
Availability Analysis Method

Random fault injection can provide accurate results

- SEU rate per bit-day is known. Thus, we need to determine the number of sensitive configuration memory elements.
- Random fault injection preferred, full campaign would take too much time.

<table>
<thead>
<tr>
<th># Tested</th>
<th># Sensitive</th>
<th>95% confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>134</td>
<td>11.35%</td>
</tr>
<tr>
<td>10,000</td>
<td>1,382</td>
<td>13.15%</td>
</tr>
<tr>
<td>100,000</td>
<td>13,899</td>
<td>13.69%</td>
</tr>
<tr>
<td>150,000</td>
<td>20,870</td>
<td>13.74%</td>
</tr>
<tr>
<td>3,735,264</td>
<td>523,543</td>
<td>14.02%</td>
</tr>
</tbody>
</table>

Random fault injection can provide accurate estimates!

\[ F_C = 0.1409 \cdot 3735264\text{bits} = 526299\text{bits} \approx 526300\text{bits} \]
Availability Analysis Method

Step 4: Stochastic modelling to determine availability for different redundancy configurations (Duplication with comparison, TMR)

- The steady-state availability gives an indication of how much downtime must be expected.
- It is also a good figure to compare different FDIR approaches.
- Stochastic Petri nets are used for modelling, which can analytically be solved by the TimeNET tool.

<table>
<thead>
<tr>
<th>Mission</th>
<th>DWC</th>
<th>TMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentinel-3</td>
<td>0.999996</td>
<td>0.999999999...</td>
</tr>
<tr>
<td>Sentinel-3 (SPE)</td>
<td>0.998</td>
<td>0.9999998</td>
</tr>
<tr>
<td>Galileo</td>
<td>0.999998</td>
<td>0.999999999...</td>
</tr>
<tr>
<td>Galileo (SPE)</td>
<td>0.995</td>
<td>0.999998</td>
</tr>
</tbody>
</table>
Demonstration System

- Complex demonstration system comprising hardware, embedded software and workstation software components.
- Very similar to flight systems since most components are available as space-qualified versions.
Proton Irradiation Test Campaign

Validation of both FDIR Hardware Framework and Availability Analysis method.

- Test campaign was conducted at PSI.
- Three experiments:
  - $4.2 \times 10^6$ p/cm$^2$-s @ 200 MeV
  - $8.3 \times 10^6$ p/cm$^2$-s @ 200 MeV
  - $8.5 \times 10^6$ p/cm$^2$-s @ 100 MeV
- DUT: Virtex-4 SX55.
Proton Irradiation Test Campaign
Proton Irradiation Test Campaign

First interesting outcome: Static cross-sections also gained during dynamic testing by reading back the bitstream after each failure detection.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Fluence$^1$</th>
<th>Runtime</th>
<th>SEUs</th>
<th>X-Section / bit$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 MeV /A</td>
<td>3.82E+10</td>
<td>9,176 s</td>
<td>10,728</td>
<td>1.83E-14</td>
</tr>
<tr>
<td>200 MeV /B</td>
<td>2.67E+10</td>
<td>3,207 s</td>
<td>7,472</td>
<td>1.83E-14</td>
</tr>
<tr>
<td>100 MeV</td>
<td>3.42E+10</td>
<td>4,028 s</td>
<td>8,052</td>
<td>1.54E-14</td>
</tr>
</tbody>
</table>

Cross-sections: Configuration memory

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Fluence$^1$</th>
<th>Runtime</th>
<th>SEUs</th>
<th>X-Section / bit$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 MeV /A</td>
<td>3.82E+10</td>
<td>9,176 s</td>
<td>6,225</td>
<td>3.69E-14</td>
</tr>
<tr>
<td>200 MeV /B</td>
<td>2.67E+10</td>
<td>3,207 s</td>
<td>4,290</td>
<td>3.64E-14</td>
</tr>
<tr>
<td>100 MeV</td>
<td>3.42E+10</td>
<td>4,028 s</td>
<td>4,808</td>
<td>3.19E-14</td>
</tr>
</tbody>
</table>

Cross-sections: Block RAMs

1 [p/cm²]  
2 [cm²/bit]
Proton Irradiation Test Campaign

Dynamic Test Results #1 (@ 200 MeV)

• Measured during beam test:
  – Failures detected and recovered: 439
  – Average proton flux: 4.16E+06 p/cm²-s
  – Average Mean Time Between Failures (MTBF): 20.23 sec

• Estimation:

\[
300 \text{ bits} + \frac{3.69 \times 10^{-14} \text{ cm}^2/\text{bit} \times (67858 + 27351) \text{ bits}}{[\frac{1}{18.275 \text{ s/p}}]} = 18.275 \text{ s/p}
\]

• Error: 9.7%
Proton Irradiation Test Campaign

Dynamic Test Results #2 (@ 200 MeV)

- **Measured during beam test:**
  - Failures detected and recovered: 343
  - Average proton flux: $8.31 \times 10^6$ p/cm$^2$-s
  - Average Mean Time Between Failures (MTBF): 9.147 sec

- **Estimation:**

  $$MTBF = \left[ 8.31 \times 10^6 \text{p/cm}^2\text{s} \cdot (1.83 \times 10^{-14} \text{cm}^2/\text{bit} \cdot 526300 \text{bits} + 3.64 \times 10^{-14} \text{cm}^2/\text{bit} \cdot (67858 + 27351) \text{bits}) \right]^{-1} = 9.185 \text{s/p}$$

- **Error:** 0.4%
Proton Irradiation Test Campaign

Dynamic Test Results #3 (@ 100 MeV)

- **Measured during beam test:**
  - Failures detected and recovered: 309
  - Average proton flux: $8.48 \times 10^6$ p/cm$^2$-s
  - Average Mean Time Between Failures (MTBF): 12.518 sec

- **Estimation:**

  \[
  MTBF = \left[ 8.48 \times 10^6 \text{p/cm}^2\text{s} \cdot (1.54 \times 10^{-14} \text{cm}^2/\text{bit} \cdot 526300 \text{bits} + 3.19 \times 10^{-14} \text{cm}^2/\text{bit} \cdot (67858 + 27351) \text{bits}) \right]^{-1} = 10.586 \text{s/p}
  \]

- **Error:** 15.4%
# Proton Irradiation Test Campaign

## Availability Prediction (based on stochastic Petri nets)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TX Img</th>
<th>RX Img</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 MeV /A</td>
<td>80,724</td>
<td>77,621</td>
<td>0.9616</td>
</tr>
<tr>
<td>200 MeV /B</td>
<td>28,586</td>
<td>26,266</td>
<td>0.9188</td>
</tr>
<tr>
<td>100 MeV</td>
<td>35,845</td>
<td>33,741</td>
<td>0.9413</td>
</tr>
</tbody>
</table>

Measured availability during beam test

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Availability (Error) using measured MTBF</th>
<th>Availability (Error) using predicted MTBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 MeV /A</td>
<td>0.9639 (0.2%)</td>
<td>0.9602 (0.1%)</td>
</tr>
<tr>
<td>200 MeV /B</td>
<td>0.9235 (0.5%)</td>
<td>0.9238 (0.5%)</td>
</tr>
<tr>
<td>100 MeV</td>
<td>0.9429 (0.2%)</td>
<td>0.9332 (0.9%)</td>
</tr>
</tbody>
</table>

Predicted steady-state availability using Petri nets
Proton Irradiation Test Campaign

Summary:

- It was demonstrated that the proposed FDIR framework withstands a real radiation environment.
- It was shown that the availability analysis method could predict the measured MTBF value with a maximum error of 15.4% and the availability figure with a maximum error of only 0.9%.
- Static cross-sections at 200 MeV were measured for the Virtex-4 SX55 device (NASA/Xilinx documents only provide data up to 60 MeV).
Publications


- “Adaptive FDIR framework for payload data processing systems using reconfigurable FPGAs,” in Proc. of 8th NASA/ESA Conference on Adaptive Hardware and Systems, June 2013

- “New voter design enabling hot redundancy for asynchronous network nodes,” in Proc. of 9th NASA/ESA Conference on Adaptive Hardware and Systems, July 2014


- “Mitigation of radiation effects in SRAM-based FPGAs for space applications,” in ACM Computing Surveys, 2014


- Thank you -