System-Level Mitigation of SEFIIs in Data Handling Architectures, A Solution for Small Satellites

Mrs. Shazia Maqbool
Supervised By Dr. Craig I Underwood
Overview

- Context
- Mitigation Scheme
  - Architecture - Top Level Description
  - Realization - Protocol Gets Defined
  - Implementation - Test Cases
- Conclusions
Single Event Functional Interrupts (SEFIs)

- A type of anomaly in microcircuits caused by a single ion strike
- Occurs in sensitive cross-section of the device
- User doesn’t have direct access to fault location
- Signatures
  - An upset rate higher than expected
  - Non responding device
  - In a communication network SEFI is an event, which stops communication
  - Variations in device current consumption
- During a SEFI, device is unavailable to the system
- Device is potentially recoverable
  - Recovery involves resetting or power cycling
  - System recovery requires restoring the device functionality followed by its state recovery
System Architecture

• A fast data network interlinks all units
  ➢ Scalable
  ➢ Distributed
  ➢ Reusable

• A system level SEFI mitigation

• A diagnosis and recovery (DAR) packet from each unit acts as an indicator of health status for the unit

• The supervisor intervenes when a packet does not arrive or it does not match expectation
On-Board Computer

Possible source of fault
- Processor
- Memory
- Network interface

Required underlying mitigations
- EDAC
- OPC
## SEFI Signatures

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microprocessor</td>
<td>Screech</td>
</tr>
<tr>
<td></td>
<td>Non responsive</td>
</tr>
<tr>
<td></td>
<td>Calculation errors</td>
</tr>
<tr>
<td></td>
<td>Current Variations</td>
</tr>
<tr>
<td>Memory</td>
<td>Upset rate higher than expected</td>
</tr>
<tr>
<td>Network interface</td>
<td>Invalid packet</td>
</tr>
<tr>
<td></td>
<td>Non responsive</td>
</tr>
<tr>
<td></td>
<td>Link establishment time-out</td>
</tr>
<tr>
<td></td>
<td>Transmit and receive time-out</td>
</tr>
</tbody>
</table>
# Diagnosis And Recovery (DAR) Packet Flow

<table>
<thead>
<tr>
<th>OBC-Processor</th>
<th>OBC- Interface FPGA</th>
<th>Supervisor</th>
<th>Code Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAR Process Starts Disable Interrupts</td>
<td>SODARP Marker Start Sampling Current</td>
<td>Collect Current Value</td>
<td>DAR Packet Received Compare with Stored Values</td>
</tr>
<tr>
<td>Perform Test</td>
<td>Collect SEU Count Send DAR Packet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable Interrupts</td>
<td></td>
<td></td>
<td>Update Memory</td>
</tr>
<tr>
<td>Waiting for Supervisor Response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command to Update Program Memory</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25/08/2005

SSC-VII-3
# Recovery Method

<table>
<thead>
<tr>
<th>Fault Type</th>
<th>Recovery Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screech</td>
<td>Reload program memory</td>
</tr>
<tr>
<td>Packet time-out (Network problems)</td>
<td>Detect faulty interface and apply recovery</td>
</tr>
<tr>
<td>Packet time_out (Processor Problem)</td>
<td>Maskable interrupt NMI, Reset</td>
</tr>
<tr>
<td>Current consumption variations</td>
<td>Power cycle and reload memory</td>
</tr>
<tr>
<td>SEU count exceeding threshold</td>
<td>Reload memory</td>
</tr>
<tr>
<td>Test task result mismatch</td>
<td>Reset and reload memory</td>
</tr>
</tbody>
</table>
Recovery Method (2)

In case of a processor reset and power cycle, the OBC should be allowed sufficient time for reinitialization.

The supervisor needs to keep a record of recoveries applied.

Consecutive recovery cycles need to be avoided.
Synchronization

OBC
- Supervisory protocol initialized
- Wait for supervisor ready
- Supervisor ready packet received
- Supervisor ready packet every invocation period

Supervisor
- Supervisory protocol initialized
- Wait for OBC ready packet
- OBC ready packet received
- Supervisory protocol established, DAR packets every invocation period
- OBC ready packet received

25/08/2005

SSC-VII-3
Test Case 1

Router

OBC

Supervisor

Analog Voltage

25/08/2005

SSC-VII-3

11
Test Cases

• Test Case 1
  - Supervisor time slice
  - Invocation period
  - OBC computational overhead
  - Worst case detection and recovery latencies for each fault mode

• Test Case 2
  - Demonstration of the synchronization protocol
  - DAR packet generation and processing time
  - Time required by CODEC FPGA
Conclusions

A system-level approach has been presented to mitigate SEFIs in data handling architectures

- Upset detection is not straightforward, limits effectiveness of currently available mitigation techniques

- Increasing SEFI susceptibility in all major data handling device technologies

- A system level intelligent supervisor allows monitoring of a wide range of devices with minimal overhead

- Synchronization is straightforward

- Requires significantly low processing capabilities, requires at least two timers for interface and one timer for each underlying unit

- Two test cases are produced to evaluate the mitigation scheme performance

- Future work will require investigation into adding resource management, redundancy management and scheduling capabilities to the supervisor, thus leading to an adaptive, autonomous and highly capable data handling architecture
Thank You!