Fault-injection-Enhanced Virtual Prototypes Enable Early SW Development for Automotive Applications

Mohammad Badawi, Javier Castillo, Andreas Mauderer, Jan-Hendrik Oetjens

BOSCH
VPs Enable Early SW-Safety Development

• Effort to comply to safety standards increases as SoC’s complexity increase.
• System dependability (HW + SW) \(\rightarrow\) evaluate SW before HW is available.
• SW-based FDM & FRM to enhance overall system dependability, but:
  • Increase SW complexity.
  • Further SW quality assurance is needed \(\rightarrow\) more effort and time.
  • More reason for SW to start earlier.
• Using VPs to enable SW development very early, but need to
  • Provide high degree of flexibility to ease integration with SW.
  • Raise abstraction level \(\rightarrow\) correct functionality can be viewed from SW perspective.
Fault Injection Framework

• Generalize fault injection and reporting to provide the needed flexibility.

• Models faults in registers, communication and computation:
  • Transient faults (SEU and MEU).
  • Permanent faults.

• Use of Fault Scenarios
  • Ease traceability.
  • Enable creating a set of related faults.
  • Scenario identifier, Number of packets involved, Reference to first packet.
  • Envelope for faulty use cases.
Fault Payload Packets

- Register Access Callback
  - Inject fault in a register.

- TLM Socket Callback
  - Communication fault at interconnect, interface or register port.

- Function Corruption Callback
  - Hook customized functions to targeted data processing functions in the VP.
Case Studies: Integration

• Simulation Based FMUs

• Multi-Process Simulation
Case Studies: Results

- Fault injection reports
- Fault injection overhead
- Overhead caused to applications

<table>
<thead>
<tr>
<th>Fault Type</th>
<th>Num Faults</th>
<th>Overhead [ms]</th>
<th>Overhead Per Fault [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg_W</td>
<td>50</td>
<td>0.518</td>
<td>0.010</td>
</tr>
<tr>
<td>Reg_W</td>
<td>500</td>
<td>4.103</td>
<td>0.008</td>
</tr>
<tr>
<td>Reg_W</td>
<td>1000</td>
<td>8.370</td>
<td>0.008</td>
</tr>
<tr>
<td>Reg_W</td>
<td>2000</td>
<td>17.776</td>
<td>0.009</td>
</tr>
<tr>
<td>Reg_W_M</td>
<td>50</td>
<td>0.699</td>
<td>0.014</td>
</tr>
<tr>
<td>Reg_W_M</td>
<td>500</td>
<td>7.288</td>
<td>0.015</td>
</tr>
<tr>
<td>Reg_W_M</td>
<td>1000</td>
<td>14.761</td>
<td>0.015</td>
</tr>
<tr>
<td>Reg_W_M</td>
<td>2000</td>
<td>27.705</td>
<td>0.014</td>
</tr>
<tr>
<td>Link_W</td>
<td>10</td>
<td>0.003</td>
<td>0.0003</td>
</tr>
<tr>
<td>Link_SAF</td>
<td>10</td>
<td>0.006</td>
<td>0.0006</td>
</tr>
</tbody>
</table>
Summary and Future Work

• SW-based EDM and ECM start early before RTL and GL available.
• Simulating complex real-life fault scenarios becomes possible using VP.
• Flexibility to address different integration with SW.
• Traceability and observation; fault scenarios and comprehensive reposting.
• Minor overhead.
• Further issues to address:
  • Tracing fault propagation.
  • Fault dependency analysis and better understanding of masked faults.
  • Improved and standardized fault reporting.
Questions