Simulation-based Fault Injection on Ibex Core with UVM Environment

Li Lu †, Junchao Chen †, Markus Ulbricht †, Milos Krstic †
† IHP-Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany
‡ University of Potsdam, Potsdam, Germany

Introduction

• Simulation-based fault injection enables us to identify the vulnerable parts of a circuit and employ fault mitigation techniques to improve the reliability of circuits before manufacturing.
• Testbenches based on UVM support coverage-driven verification. It can balance verification completeness with minimum verification effort and time.
• This work presents a procedure to conduct simulation-based fault injection at the gate level on the ibex core with its UVM testbench [1] to identify critical flip-flops which determine the core’s correct functioning. The workflow could reduce the time involved in the fault simulation.

Methodologies

• We use the Xcelium fault simulator provided by Cadence for the fault simulation. This simulator enables us to reuse the UVM functional verification testbench to build the fault simulation testbench.

• Testcase Selection

   Fault injection at RTL

   Identifying critical flip-flops

   Mapping the results from RTL to the gate level

   Permanent fault simulation

   Detected?

   SEU fault simulation

   Record the number of detected faults

   Remove the detected critical flip-flops from the fault list of the following testcase.

   Fault Injection:

   • Fault simulation is conducted in the order of the testcase’s contribution to the functional coverage from high to low.
   • For each testcase, we first inject permanent faults (i.e., stuck-at-1 or stuck-at-0) into all flip-flops. If the fault is not detected on the outputs, we can interpret that the fault is masked or the flip-flop is not covered by the testcase. The flip-flop is therefore considered non-critical.
   • Otherwise, SEU faults are injected several times in a specified time window. The flip-flop will be identified as critical if the number of detected SEU faults is larger than the threshold we define.
   • If the flip-flop is annotated as critical, it will be removed from the fault list of the following testcase.

• Integrated Metrics Center (IMC) provided by Cadence is used to analyze the contribution of each test case provided by the UVM testbench to the functional coverage.
• We remove the testcases which don’t or less contribute to the coverage and merge some of them. The function coverage of the selected testcases is around 98.2%.
• The fault simulation is implemented at the RTL first to identify the critical flip-flops. These flip-flops are mapped from RTL to the gate level eventually.

• We use the Xcelium fault simulator provided by Cadence for the fault simulation. This simulator enables us to reuse the UVM functional verification testbench to build the fault simulation testbench.

Discussion

The procedure could reduce the time involved in the fault simulation. However, there are some limitations to this approach.
• We need to specify the threshold to identify critical flip-flops correctly at the beginning. Otherwise, the simulation of the following testcase might be based on an incorrect fault list.
• The testcases should also be selected based on the relevant applications.
• The simulation time of testcases could be reduced if we could use more efficient constraints. The efficiency could be improved further with these considerations.

• We investigate the difference between the fault simulation results at the RTL and GL to prove the process is feasible. The results of an example is shown below.

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Difference</th>
<th>Number of runs</th>
<th>Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTL - GL</td>
<td>0.7 (≈ 3.3%)</td>
<td>2277</td>
<td>2126</td>
</tr>
<tr>
<td>RTL - GL</td>
<td>6.2%</td>
<td>2277</td>
<td>2126</td>
</tr>
<tr>
<td>SEU</td>
<td>4.7%</td>
<td>1800</td>
<td>1800</td>
</tr>
</tbody>
</table>

Fault Injection:

• Fault simulation is conducted in the order of the testcase’s contribution to the functional coverage from high to low.
• For each testcase, we first inject permanent faults (i.e., stuck-at-1 or stuck-at-0) into all flip-flops. If the fault is not detected on the outputs, we can interpret that the fault is masked or the flip-flop is not covered by the testcase. The flip-flop is therefore considered non-critical.
• Otherwise, SEU faults are injected several times in a specified time window. The flip-flop will be identified as critical if the number of detected SEU faults is larger than the threshold we define.
• If the flip-flop is annotated as critical, it will be removed from the fault list of the following testcase.

The project underlying this report is funded by the German Federal Ministry of Education and Research under grant number 16MEO134 and 16KIS1339K.


innovations for high performance microelectronics | Im Technologiepark 25 | 15236 Frankfurt (Oder) | Germany | www.ihp-microelectronics.com

Legend

• RTL

• SEU

• Critical

• Non-critical

• < threshold

• >= threshold

• Testcase

• Simulation

• Functional

• Coverage

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Simulation Coverage</th>
<th>Functional Coverage</th>
<th>Number of Flips Injection</th>
<th>Permanent Faults</th>
<th>Number of Flips Injection</th>
<th>SEU Faults</th>
<th>Number of Detected Critical Flips</th>
<th>Number of Runs</th>
<th>Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_jump</td>
<td>54,400/3,816</td>
<td>41/96</td>
<td>2277</td>
<td>943</td>
<td>199</td>
<td>2331</td>
<td>5186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prep_seq</td>
<td>3,013/44</td>
<td>177</td>
<td>1037</td>
<td>958</td>
<td>0</td>
<td>941</td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_branch_jump</td>
<td>21,075/704</td>
<td>10</td>
<td>1575</td>
<td>349</td>
<td>53</td>
<td>9940</td>
<td>386</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non_seq</td>
<td>9,651/314</td>
<td>0</td>
<td>1722</td>
<td>442</td>
<td>327</td>
<td>7342</td>
<td>280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_shoemake</td>
<td>11,274/244</td>
<td>3</td>
<td>1495</td>
<td>179</td>
<td>104</td>
<td>2627</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>test_seq</td>
<td>2,545/214</td>
<td>2</td>
<td>1094</td>
<td>11</td>
<td>0</td>
<td>2239</td>
<td>23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Federal Ministry of Education and Research

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Simulation Coverage</th>
<th>Functional Coverage</th>
<th>Number of Flips Injection</th>
<th>Permanent Faults</th>
<th>Number of Flips Injection</th>
<th>SEU Faults</th>
<th>Number of Detected Critical Flips</th>
<th>Number of Runs</th>
<th>Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_jump</td>
<td>54,400/3,816</td>
<td>41/96</td>
<td>2277</td>
<td>943</td>
<td>199</td>
<td>2331</td>
<td>5186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prep_seq</td>
<td>3,013/44</td>
<td>177</td>
<td>1037</td>
<td>958</td>
<td>0</td>
<td>941</td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_branch_jump</td>
<td>21,075/704</td>
<td>10</td>
<td>1575</td>
<td>349</td>
<td>53</td>
<td>9940</td>
<td>386</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non_seq</td>
<td>9,651/314</td>
<td>0</td>
<td>1722</td>
<td>442</td>
<td>327</td>
<td>7342</td>
<td>280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_shoemake</td>
<td>11,274/244</td>
<td>3</td>
<td>1495</td>
<td>179</td>
<td>104</td>
<td>2627</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>test_seq</td>
<td>2,545/214</td>
<td>2</td>
<td>1094</td>
<td>11</td>
<td>0</td>
<td>2239</td>
<td>23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The project underlying this report is funded by the German Federal Ministry of Education and Research under grant number 16MEO134 and 16KIS1339K.

innovations for high performance microelectronics | Im Technologiepark 25 | 15236 Frankfurt (Oder) | Germany | www.ihp-microelectronics.com

Legend

• RTL

• SEU

• Critical

• Non-critical

• < threshold

• >= threshold

• Testcase

• Simulation

• Functional

• Coverage

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Simulation Coverage</th>
<th>Functional Coverage</th>
<th>Number of Flips Injection</th>
<th>Permanent Faults</th>
<th>Number of Flips Injection</th>
<th>SEU Faults</th>
<th>Number of Detected Critical Flips</th>
<th>Number of Runs</th>
<th>Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_jump</td>
<td>54,400/3,816</td>
<td>41/96</td>
<td>2277</td>
<td>943</td>
<td>199</td>
<td>2331</td>
<td>5186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prep_seq</td>
<td>3,013/44</td>
<td>177</td>
<td>1037</td>
<td>958</td>
<td>0</td>
<td>941</td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_branch_jump</td>
<td>21,075/704</td>
<td>10</td>
<td>1575</td>
<td>349</td>
<td>53</td>
<td>9940</td>
<td>386</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non_seq</td>
<td>9,651/314</td>
<td>0</td>
<td>1722</td>
<td>442</td>
<td>327</td>
<td>7342</td>
<td>280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>debug_shoemake</td>
<td>11,274/244</td>
<td>3</td>
<td>1495</td>
<td>179</td>
<td>104</td>
<td>2627</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>test_seq</td>
<td>2,545/214</td>
<td>2</td>
<td>1094</td>
<td>11</td>
<td>0</td>
<td>2239</td>
<td>23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>