Accurate Microarchitecture-Level Fault Modeling for Studying Hardware Faults

Man-Lap Li, Pradeep Ramachandran, Ulya R. Karpuzcu, Siva Kumar Sastry Hari, Sarita V. Adve
Department of Computer Science
University of Illinois at Urbana-Champaign

HPCA 2008
Motivation

- To evaluate the efficacy of uarch-level fault tolerance solutions, it is essential to capture the expected behavior of the fault at the level at which the solution is implemented.
- Gate-level simulations: Accurately capture low-level faults but slow
- uarch-level simulations: Less accurate but fast
- Best of both the worlds ??
Contributions

• Swat Sim:
  – a novel fault injection infrastructure for studying system-level effects of gate-level permanent faults
    • selective and on-demand gate-level simulation
    • repeated invocations of gate-level simulator during run-time
    • gate-level timing simulator coupled with arch-level simulator to model gate-level timing faults

• Show that, in general, arch-level stuck-at faults do not result in similar system-level fault manifestation as gate-level stuck-at or delay faults

• Derive two probabilistic fault models, the \textit{P-model and the PDmodel}, for gate-level stuck-at and delay faults (not covered in this presentation)
SWAT-SIM

(a) μarch-Level Fault Simulation

(b) SWAT-Sim Fault Simulation
Experimental Methodology

• Swat- Sim coupled with NC Verilog
• Faulty Components:
  – ALU, AGEN, Decoder
• Fault models:
  – uarch stuckat 0 and 1
  – gate level stuckat 0 and 1
  – gate level delay
    • gate delay increased by 1 cycle
    • gate delay increased by 0.5 cycle
• Possible outcomes: uarch-mask; arch-mask; app-mask; detected; detected > 10M; SDC
• Coverage = percentage of unmasked faults that are detected within 10 million instructions
Results - Performance Overhead

<table>
<thead>
<tr>
<th>Unit</th>
<th>Fault Model</th>
<th>Max</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>Gate Stuck-At Gate Delay</td>
<td>2.20</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.65</td>
<td>1.93</td>
</tr>
<tr>
<td>AGEN</td>
<td>Gate Stuck-At Gate Delay</td>
<td>1.59</td>
<td>1.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.89</td>
<td>1.35</td>
</tr>
<tr>
<td>Decoder</td>
<td>Gate Stuck-At Gate Delay</td>
<td>2.91</td>
<td>2.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.10</td>
<td>2.91</td>
</tr>
</tbody>
</table>

Table 2. Slowdowns of SWAT-Sim when compared to pure $\mu$arch-level simulation.
Results – Accuracy of Uarch-Level Model

- Lower coverage for gate-level models
- High masking rate for gate-level models
  - fault may not be activated
  - fault may not propagate
- Uarch-level models directly change the latch value

Figure 2. Efficacy of the SWAT fault detection scheme [12] under different fault models for the ALU, AGEN, and Decoder. Depending on the fault model and the structure, the Uarch-level fault may or may not capture the system-level effects of gate-level faults accurately, as indicated by the differences in coverage.
Differences Between Fault Models

Fault Activation Rate

Fault activation rate: % instructions that are corrupted by the fault among all instructions using the hardware.

- Activation rate lower for gate-level
  - Excite and propagate
  - uarch-level fault directly injected
- Delay gate-level activation rate lower than stuck-at gate level
  - Lower excitation probability.

**Figure 4.** Mean fault activation rate for the different fault models as a percentage of the number of instructions.
Differences Between Fault Models
Bit Corruption Pattern

<table>
<thead>
<tr>
<th>Bits</th>
<th>ALU</th>
<th>AGEN</th>
<th>Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>μarch</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Gate s@1</td>
<td>91.1%</td>
<td>4.7%</td>
<td>1.2%</td>
</tr>
<tr>
<td>Gate s@0</td>
<td>84.4%</td>
<td>4.6%</td>
<td>2.8%</td>
</tr>
<tr>
<td>Delay 1cyc</td>
<td>90.4%</td>
<td>3.9%</td>
<td>1.4%</td>
</tr>
<tr>
<td>Delay 0.5cyc</td>
<td>75.0%</td>
<td>5.8%</td>
<td>2.2%</td>
</tr>
</tbody>
</table>

Table 3. Percentage of bits incorrect at the output latch.

For ALU and AGEN:
μarch-level faults injected in atmost 1 output latch bit but gate-level model can corrupt multi-bits. Most common case: only 1 bit in the output latch is corrupted.

For Decoder:
μarch-level faults injected in input latch corrupt 8+ bits in >22% cases due to large output cone.
Another source of discrepancy of the arch-level model to represent gate-level faults:

- uarch-level model corrupts an output bit with much different probability than gate-level model
• Higher activation rate implies more chances of detection => better coverage. Therefore, uarch-level models have better coverage.

• Multi-bit corruption improves the coverage even for low activation rates. Therefore, coverage of gate-level model can be similar to uarch-level model