One Bit is (Not) Enough: An Empirical Study of the Impact of Single and Multiple Bit-Flip Errors

Behrooz Sangchoolie, Chalmers
Karthik Pattabiraman, UBC
Johan Karlsson, Chalmers

http://blogs.ubc.ca/karthik/
Soft Error Problem

- Soft errors are increasing in computer systems

Source: Shekar Borkar (Intel) - Stanford talk in 2005
Error Resilience

• Ability of a program to **NOT** produce an **SDC** (Silent Data Corruption) upon a hardware fault

• **SDC**: Deviation of output from golden output
Our Groups’ Research: Application-level Selective Fault-Tolerance

- Add error detectors to applications to detect SDCs
  - Much more efficient than “all-or-nothing” techniques

![Diagram showing application execution with error detectors and SDCs]
Software Fault Injection

- Inject faults to iteratively improve coverage
- Find which errors result in SDCs
- Find errors that are missed by detectors
Hardware Vs. Software Injectors

Accuracy

Hardware Injectors

Gap

Ease of Analysis and Configurability

Software Injectors
Main Problem

• Software Fault Injectors use the **single bit-flip** fault model to abstract the effect of soft errors

• But a single soft error is likely to manifest as **multiple-bit errors** at the **application level**

---

DAC 2013

Quantitative Evaluation of Soft Error Injection Techniques for Robust System Design

Hyungmin Cho¹, Shahrzad Mirkhani³, Chen-Yong Cher⁴, Jacob A. Abraham³, Subhasish Mitra¹,²

¹Department of EE and ²Department of CS ³Computer Engineering Research Center ⁴IBM T. J. Watson Research Center,
Stanford University, Stanford, CA, USA The University of Texas at Austin, Austin, TX, USA Yorktown Heights, NY, USA
Fault Model

• Faults in the processor
  • Register file
  • Computational elements

• Faults in memory
  • Assume memory is ECC protected

• Faults in control logic
  • Assumed to be protected by other means
Multiple Bit Flip Errors

- Single Soft error → Multiple bit-flips in software
- Error propagation in the micro-architectural level

Source: Chen-Yong Chen, IBM Research
This Paper: Main Question

• Does the *multiple bit-flip* model result in significantly different *error resilience* results compared with the *single bit-flip* model?

  • For different kinds of injection techniques

  • For different kinds of fault distributions
Challenge

• **Multiple-bit injection space is extremely large**
  • Multiple bit-flips in a single register
  • Multiple bit-flips in multiple registers
  • Any combination of the above
Main Insight

• Effect of multiple-bit faults are confined within a (small) dynamic instruction window from fault
• Sufficient to consider multiple-bit faults within the window for injecting faults into application
Why does this hold in practice?

- Soft errors manifest as multiple bit-flips in the program through propagation in the hardware.
- Hardware error propagation is confined to the instruction window in superscalar processors.
Outline

• Motivation and Goal

• Experimental Setup

• Results

• Conclusion
Experimental Setup: LLFI tool

Works at LLVM compiler’s intermediate (IR) [Wei14]
Experimental Setup: FI techniques

- **Inject on Read**: Inject fault *before* reading a source register
- **Inject on Write**: Inject fault *after* writing to a destination register
- Models faults that occur in the **register file**
- Models faults that occur during the **computation**

<table>
<thead>
<tr>
<th>21bc</th>
<th>mr</th>
<th>r2, r1</th>
</tr>
</thead>
<tbody>
<tr>
<td>21c0</td>
<td>or</td>
<td>r3, r2, r1</td>
</tr>
<tr>
<td>21c4</td>
<td>neg</td>
<td>r5, r4</td>
</tr>
<tr>
<td>21c8</td>
<td>and</td>
<td>r7, r5, r6</td>
</tr>
<tr>
<td>21cc</td>
<td>and</td>
<td>r8, r3, r7</td>
</tr>
</tbody>
</table>
Experimental Setup: Parameters

- **window_size**: max distance between faults
  - Varies from 1 to 1000 (random and fixed values)
- **number_of_bits**: max number of faults/run
  - Varies from 1 to 30 (1-10 and 30 as an extreme)

Dynamic Instructions Executed

```
Multiple Bit Flip Faults Possible ?
```

```
Fault injection
```

```
window_size
```

```
Dynamic Instructions Executed
```
Experimental Setup: Benchmarks

• **11 MiBench** programs – embedded systems
• **4 Parboil** programs – parallel computing
Experimental Setup: Approach

We perform over **27 Million fault injection** experiments for each combination of the benchmark, parameters, and FI technique!

**15 benchmarks * 91 parameter values * 2 techniques * 10,000 fault injections/combination**

= **27,300,0000**
Experimental Setup: Outcome Classification

- **Input**
  - Hardware
  - Software

- **Output**
  - ?

**Computer system**

- **Benign**
  - Detected by hardware exceptions

- **No output**
  - Hang

- **Incorrect Output**
  - a.k.a.
  - Silent data corruption (SDC)

Detected by hardware exceptions
Outline

• Motivation and Goal
• Experimental Setup
• Results
• Conclusion
Research Questions (RQs)

• **RQ1**: How many multi-bit errors are *activated* in a program?
• **RQ2**: Does the single bit-flip error model result in *pessimistic* percentage of SDCs compared with multiple bit-flip error model?
• **RQ3**: Is there an *upper bound* to the maximum number of multiple bit-flips needed to cause pessimistic percentage of SDCs?
• **RQ4**: Is there a *maximum dynamic window size* that causes pessimistic percentage of SDCs?
• **RQ5**: Can we use *single bit-flip* results to prune the multiple bit flip fault injection space?
RQ1: Activation of Multiple Bit Flips

Most programs have fewer than 10 activated multiple bit-flips
RQ2: Single Bit-flip Vs. Multiple Bit Flips

Single Bit-Flip Model provides pessimistic SDC results or results close to multiple bit-flip model
RQ3: Upper bound on multiple bit flips

At most 3 bit flips are sufficient to get pessimistic SDC results in most applications (in the few cases where single bit-flip model is not sufficient)

<table>
<thead>
<tr>
<th>Program</th>
<th>inject-on-read</th>
<th>inject-on-write</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max-MBF</td>
<td>win-size</td>
</tr>
<tr>
<td>FFT</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
<tr>
<td>IFFT</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
<tr>
<td>CRC32</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>dijkstra</td>
<td>single bit-flip</td>
<td>3</td>
</tr>
<tr>
<td>sha</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
<tr>
<td>stringsearch</td>
<td>2</td>
<td>RND(2-10)</td>
</tr>
<tr>
<td>bfs</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
<tr>
<td>histo</td>
<td>single bit-flip</td>
<td>6</td>
</tr>
<tr>
<td>sad</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
<tr>
<td>spmv</td>
<td>single bit-flip</td>
<td>single bit-flip</td>
</tr>
</tbody>
</table>
RQ4: Effect of window_size

Window sizes do not have a significant effect on SDC rates. However, smaller window sizes (<5) result in higher percentages of SDCs.
RQ5: Multiple bit-flip (MB) Error Space Pruning from Single Bit-Flip (SB)

Sufficient to inject multiple bit-flips into locations where single bit-flips result in benign outcomes to get pessimistic SDC results
Takeaways

• In most cases, single bit flip fault model yields comparable resilience to multiple bit fault model
• To get pessimistic estimates of resilience, we need at most 3 multiple bit flips across applications
• Smaller window sizes result in (slightly) higher percentages of SDCs for a given no. of bit flips
• Sufficient to inject multiple-bit errors into locations where single bit-flips result in Benign outcomes
Outline

• Motivation and Goal

• Experimental Setup

• Results

• Conclusion
Conclusion

Does the *multiple bit-flip* model result in significantly different error resilience results compared with the *single bit-flip* model?

- No, in most cases
- Yes, in a few cases

Based on a total of 27 million fault injection experiments

Bottom line: One bit is *Often* Enough!