Design as You See FIT: System-Level Soft Error Analysis of Sequential Circuits

Dan Holcomb    Wenchao Li    Sanjit A. Seshia

Department of EECS
University of California, Berkeley

Design Automation and Test in Europe, 2009
Soft errors in VLSI circuits

- Spurious radiation-induced flip of one or more stored bits
- Does not permanently damage devices
- Measured in units of FIT; 1 FIT is 1 failure in $10^9$ hrs
- Strikes to logic or directly to memory
Soft errors in VLSI circuits

- Spurious radiation-induced flip of one or more stored bits
- Does not permanently damage devices
- Measured in units of FIT; 1 FIT is 1 failure in $10^9$ hrs
- Strikes to logic or directly to memory

- Logic FIT may increase 9 orders of magnitude from 1992-2011 [Shivakumar 02]
- Logic FIT approaching memory FIT around 100nm [Shivakumar 02, Baumann 05]

Circuit-level hardening techniques exist, but have costs
Our contribution
the verification guided error resilience methodology

- Use formal specifications to capture system-level correctness
Our contribution
the verification guided error resilience methodology

- Use formal specifications to capture system-level correctness
- Use verification to analyze system-level impact of circuit-level upsets

VGER Toolkit

Formal Specification
Seq. Circuit
Workload
FIT Target

System FIT
Latches/gates to harden against soft errors

Holcomb, Li, Seshia  Design as You See FIT
Our contribution
the verification guided error resilience methodology

- Use formal specifications to capture system-level correctness
- Use verification to analyze system-level impact of circuit-level upsets
- Guide efficient circuit hardening techniques
Our contribution
the verification guided error resilience methodology

- Use formal specifications to capture system-level correctness
- Use verification to analyze system-level impact of circuit-level upsets
- Guide efficient circuit hardening techniques

Particularly suited for communication protocols and on-chip networks
Outline

**Introduction**
- Single Event Upset
- Circuit-level masking
- System-level masking

**VGER Toolkit**
- BFIT: Circuit-level soft error analysis
- Sequential Simulation with Monitors

**Case Study: CMP Router**
- Analysis
- Efficient Hardening
Basic mechanisms of single event upset (SEU) in logic

- Strike near sensitive diffusion
  - Sensitive diffusion is a function of gate input
  - If sufficient charge, glitch results at gate output
  - Glitch propagates downstream toward sequential element(s)
Circuit-level masking
not all glitches are equally likely to flip bits

Logical Masking

▶ Is there a sensitized path from strike to latch(es)?

Timing Masking

▶ Does the glitch arrive at latch(es) while open?

Electrical Masking

▶ Is the strike magnitude sufficient to cause upset?

SEU can lead to single (SBU) or multi-bit upset (MBU)
Related work
Circuit-level masking

Related work

- Static analysis of circuit structure [Miskov-Zivanov 06, B. Zhang 06]
Circuit-level masking

Related work

- Static analysis of circuit structure [Miskov-Zivanov 06, B. Zhang 06]
- Electrical masking using input vectors [M. Zhang 06, Rao 07]
  - Path-based methods
  - Use only a subset of paths
  - Lose accuracy in timing masking
Related work

- Static analysis of circuit structure [Miskov-Zivanov 06, B. Zhang 06]
- Electrical masking using input vectors [M. Zhang 06, Rao 07]
  - Path-based methods
  - Use only a subset of paths
  - Lose accuracy in timing masking
- Timing masking, with no electrical masking [Krishnaswamy 08]
Circuit-level masking

Related work

- Static analysis of circuit structure [Miskov-Zivanov 06, B. Zhang 06]
- Electrical masking using input vectors [M. Zhang 06, Rao 07]
  - Path-based methods
  - Use only a subset of paths
  - Lose accuracy in timing masking
- Timing masking, with no electrical masking [Krishnaswamy 08]
- No analysis of MBU
Related work

- Static analysis of circuit structure [Miskov-Zivanov 06, B. Zhang 06]
- Electrical masking using input vectors [M. Zhang 06, Rao 07]
  - Path-based methods
  - Use only a subset of paths
  - Lose accuracy in timing masking
- Timing masking, with no electrical masking [Krishnaswamy 08]
- No analysis of MBU

Our approach

- An efficient method for estimating electrical/timing masking
  - Analysis of circuits up to 20k gates
  - Able to handle multiple sensitized paths
  - Analysis of both SBU and MBU
System level masking
not all bit flips are equally likely to cause system failure

Related work
Introduction
VGER Toolkit
Case Study: CMP Router

System level masking
not all bit flips are equally likely to cause system failure

Related work

- Architectural Vulnerability Factor [Mukherjee 03]
  - Find probability of a bit flip leading to incorrect future execution
  - Requires detailed architecture model
System level masking
not all bit flips are equally likely to cause system failure

Related work

▶ Architectural Vulnerability Factor [Mukherjee 03]
  ▶ Find probability of a bit flip leading to incorrect future execution
  ▶ Requires detailed architecture model

▶ Output equivalence
  ▶ Find probability of a bit flip leading to incorrect outputs
  ▶ Model state using Markov Chain theory [Miskov-Zivanov 08]
System level masking
not all bit flips are equally likely to cause system failure

Related work

- **Architectural Vulnerability Factor** [Mukherjee 03]
  - Find probability of a bit flip leading to incorrect future execution
  - Requires detailed architecture model

- **Output equivalence**
  - Find probability of a bit flip leading to incorrect outputs
  - Model state using Markov Chain theory [Miskov-Zivanov 08]

- **Verification Guided** [Seshia 07]
  - Find probability possibility of a bit flip leading to bad behavior
  - Bad behavior formalized using specifications
  - Model checking identifies non-critical latches
  - High confidence but binary
  - Can apply to individual functional blocks
System level masking
not all bit flips are equally likely to cause system failure

Related work

- **Architectural Vulnerability Factor** [Mukherjee 03]
  - Find probability of a bit flip leading to incorrect future execution
  - Requires detailed architecture model

- **Output equivalence**
  - Find probability of a bit flip leading to incorrect outputs
  - Model state using Markov Chain theory [Miskov-Zivanov 08]

- **Verification Guided** [Seshia 07]
  - Find probability possibility of a bit flip leading to bad behavior
  - Bad behavior formalized using specifications
  - Model checking identifies non-critical latches
  - High confidence but binary
  - Can apply to individual functional blocks

Our approach

- Verification guided, but produce a refined ranking
Outline

Introduction
Single Event Upset
Circuit-level masking
System-level masking

VGER Toolkit
BFIT: Circuit-level soft error analysis
Sequential Simulation with Monitors

Case Study: CMP Router
Analysis
Efficient Hardening
VGER toolkit

VGER Toolkit

- Formal Specification
- Seq. Circuit
- Workload
- FIT Target

System FIT

Latches/gates to harden against soft errors

Holcomb, Li, Seshia  Design as You See FIT
VGER toolkit

BFIT
Circuit Analysis

Formal Specification
Seq. Circuit
Workload
FIT Target

VGER Toolkit

Sequential Sim.
w/ Monitors

System FIT

Latches/gates to harden against soft errors

Holcomb, Li, Seshia
Design as You See FIT
VGER toolkit

BFIT: Circuit-level soft error analysis
Sequential Simulation with Monitors

Holcomb, Li, Seshia
Design as You See FIT
VGER toolkit

BFIT: Circuit-level soft error analysis
Sequential Simulation with Monitors
**VGER toolkit**

**BFIT Circuit Analysis**

- Formal Specification
- Seq. Circuit
- Workload
- FIT Target

**VGER Toolkit**

- Errors
- State Vectors
- System FIT

**Sequential Sim. w/ Monitors**

- Latches/gates to harden against soft errors

**Introduction**

*VGER Toolkit*

*Case Study: CMP Router*

*BFIT: Circuit-level soft error analysis*

*Sequential Simulation with Monitors*
VGER toolkit

BFIT: Circuit-level soft error analysis
Sequential Simulation with Monitors

VGER Toolkit

Sequential Sim. w/ Monitors

Formal Specification
Seq. Circuit
Workload
FIT Target

Comb. Ckt
State Vectors
Errors
FIT Errors

System FIT
Latches/gates to harden against soft errors

VGER Toolkit

Introduction
VGER Toolkit
Case Study: CMP Router

Holcomb, Li, Seshia
Design as You See FIT
VGER toolkit

BFIT: Circuit-level soft error analysis
Sequential Simulation with Monitors

- Introduction
  - VGER Toolkit
  - Case Study: CMP Router

Holcomb, Li, Seshia
Design as You See FIT
BFIT circuit level analysis tool

- Open source C++ simulation tool for combinational logic circuits
- Based on Nangate 45nm open cell library

**Inputs**

- Comb. circuit and sampled states

**Outputs**

- FIT of all events: $FIT_{g\rightarrow E}$
  - Struck gate $g$
  - Set of upset sequential elements $E$
  - some $E$ are SBU, others are MBU
What determines the FIT of an event?

- Every possible strike is represented by a collected charge and time \((q, t)\)

\[
FIT_{g \rightarrow E} \propto \int \int R_g(q, t) \, N_{g \rightarrow E}(q, t) \, dt \, dq
\]

\(R_g(q, t) \in \mathbb{R}\)

probability of observing strike \(q, t\) at gate \(g\)

\(N_{g \rightarrow E}(q, t) \in \{0, 1\}\)

conditional probability of upset in set \(E\) of latches, given a strike \(q, t\) at gate \(g\)

- Encompasses logical, electrical, timing masking
BFIT approach to Masking

$N(q, t)$ of single path can be characterized using path delay and gate input

- Inputs determine drive strength

- Input determines shape of $N(q, t)$

**Table: FIT vs gate input**

<table>
<thead>
<tr>
<th>Input</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIT</td>
<td>5.0E-6</td>
<td>7.5E-6</td>
<td>16.1E-6</td>
<td>2.9E-6</td>
</tr>
</tbody>
</table>

Holcomb, Li, Seshia  
Design as You See FIT
BFIT approach to Masking

\( N(q, t) \) of single path can be characterized using path delay and gate input

- Inputs determine drive strength
- Input determines shape of \( N(q, t) \)
- Path delay is time-shift to \( N(q, t) \)

### Table: FIT vs gate input

<table>
<thead>
<tr>
<th>q</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.0E-6</td>
<td>7.5E-6</td>
<td>16.1E-6</td>
<td>2.9E-6</td>
</tr>
</tbody>
</table>

### Table: FIT vs path length

<table>
<thead>
<tr>
<th>q</th>
<th>15</th>
<th>5</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.2E-6</td>
<td>6.4E-6</td>
<td>6.4E-6</td>
<td>6.7E-6</td>
</tr>
</tbody>
</table>
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
2. Dynamic programming back trace in reverse levelized order

▶ If an input can flip current gate, back propagate delays
3. For each gate $g$, find all possible $N_{g \rightarrow E}(q, t)$ using:
   ▶ List of path delays and terminating latches
   ▶ Gate input state and load capacitance
   ▶ Cell precharacterization
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
2. Dynamic programming back trace in reverse levelized order
   ▶ If an input can flip current gate, back propagate delays
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
2. Dynamic programming back trace in reverse levelized order
   ▶ If an input can flip current gate, back propagate delays
3. For each gate $g$, find all possible $N_{g\rightarrow E}(q, t)$ using:
   ▶ List of path delays and terminating latches
   ▶ Gate input state and load capacitance
   ▶ Cell precharacterization
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
2. Dynamic programming back trace in reverse levelized order
   - If an input can flip current gate, back propagate delays
3. For each gate $g$, find all possible $N_{g \rightarrow E}(q, t)$ using:
   - List of path delays and terminating latches
   - Gate input state and load capacitance
   - Cell precharacterization

\[ N(q, t) \]

\[ d_i = [8] \]

\[ d = \{8: \text{L}_1, 26: \text{L}_2\} \]
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG
2. Dynamic programming back trace in reverse levelized order
   ▶ If an input can flip current gate, back propagate delays
3. For each gate \( g \), find all possible \( N_{g \rightarrow E}(q, t) \) using:
   ▶ List of path delays and terminating latches
   ▶ Gate input state and load capacitance
   ▶ Cell precharacterization
BFIT results

<table>
<thead>
<tr>
<th>Circuit</th>
<th>INPUTS</th>
<th>LATCHES</th>
<th>GATES</th>
<th>RUNTIME (s/1k vectors)</th>
<th>FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>214</td>
<td>179</td>
<td>3232</td>
<td>35</td>
<td>3.27e-3</td>
</tr>
<tr>
<td>s9234</td>
<td>247</td>
<td>228</td>
<td>7230</td>
<td>68</td>
<td>1.06e-2</td>
</tr>
<tr>
<td>s13207</td>
<td>700</td>
<td>669</td>
<td>10277</td>
<td>136</td>
<td>1.77e-2</td>
</tr>
<tr>
<td>s15850</td>
<td>611</td>
<td>597</td>
<td>12712</td>
<td>207</td>
<td>2.24e-2</td>
</tr>
<tr>
<td>s38417</td>
<td>1664</td>
<td>1636</td>
<td>28223</td>
<td>451</td>
<td>5.62e-2</td>
</tr>
<tr>
<td>s38584</td>
<td>1464</td>
<td>1452</td>
<td>28854</td>
<td>1311</td>
<td>4.85e-2</td>
</tr>
<tr>
<td>s35932</td>
<td>1763</td>
<td>1728</td>
<td>23012</td>
<td>204</td>
<td>3.99e-2</td>
</tr>
</tbody>
</table>
BFIT results
multiple bit upsets

- Do MBU occur?
BFIT results
multiple bit upsets

Do MBU occur?

<table>
<thead>
<tr>
<th>Circuit</th>
<th>BFIT</th>
<th>HSPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>8.6e-7</td>
<td>7.4e-7</td>
</tr>
<tr>
<td>MBU</td>
<td>9.0e-7</td>
<td>12.6e-7</td>
</tr>
<tr>
<td>total FIT</td>
<td>1.7e-6</td>
<td>2.0e-6</td>
</tr>
</tbody>
</table>
BFIT results
multiple bit upsets

▶ Do MBU occur?

<table>
<thead>
<tr>
<th>Circuit</th>
<th>%SBU</th>
<th>%MBU</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>0.8388</td>
<td>0.1612</td>
</tr>
<tr>
<td>s9234</td>
<td>0.8625</td>
<td>0.1375</td>
</tr>
<tr>
<td>s13207</td>
<td>0.9379</td>
<td>0.0621</td>
</tr>
<tr>
<td>s15850</td>
<td>0.9148</td>
<td>0.0852</td>
</tr>
<tr>
<td>s38417</td>
<td>0.8812</td>
<td>0.1188</td>
</tr>
<tr>
<td>s38584</td>
<td>0.9538</td>
<td>0.0462</td>
</tr>
<tr>
<td>s35932</td>
<td>0.9960</td>
<td>0.0030</td>
</tr>
</tbody>
</table>

▶ Is MBU a concern?

<table>
<thead>
<tr>
<th></th>
<th>BFIT</th>
<th>HSPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>8.6e-7</td>
<td>7.4e-7</td>
</tr>
<tr>
<td>MBU</td>
<td>9.0e-7</td>
<td>12.6e-7</td>
</tr>
<tr>
<td>total FIT</td>
<td>1.7e-6</td>
<td>2.0e-6</td>
</tr>
</tbody>
</table>
BFIT results
multiple bit upsets

▶ Do MBU occur?

▶ Is MBU a concern?

<table>
<thead>
<tr>
<th>Circuit</th>
<th>%SBU</th>
<th>%MBU</th>
</tr>
</thead>
<tbody>
<tr>
<td>s5378</td>
<td>0.8388</td>
<td>0.1612</td>
</tr>
<tr>
<td>s9234</td>
<td>0.8625</td>
<td>0.1375</td>
</tr>
<tr>
<td>s13207</td>
<td>0.9379</td>
<td>0.0621</td>
</tr>
<tr>
<td>s15850</td>
<td>0.9148</td>
<td>0.0852</td>
</tr>
<tr>
<td>s38417</td>
<td>0.8812</td>
<td>0.1188</td>
</tr>
<tr>
<td>s38584</td>
<td>0.9538</td>
<td>0.0462</td>
</tr>
<tr>
<td>s35932</td>
<td>0.9960</td>
<td>0.0030</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BFIT</th>
<th>HSPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>8.6e-7</td>
<td>7.4e-7</td>
</tr>
<tr>
<td>MBU</td>
<td>9.0e-7</td>
<td>12.6e-7</td>
</tr>
<tr>
<td>total FIT</td>
<td>1.7e-6</td>
<td>2.0e-6</td>
</tr>
</tbody>
</table>
Verification Guided Error Resilience using sequential simulation with monitors

- Correctness captured in specifications
- Hardware monitors synthesized from specifications
- Estimate failure probabilities using random fault injections
- Accurately incorporate workload
- Produces refined ranking

Inputs
- List of formal specifications
- List of circuit errors $E$

output
- $FP_E$ - the probability that upset $E$ will lead to a violated specification
Outline

**Introduction**
- Single Event Upset
- Circuit-level masking
- System-level masking

**VGER Toolkit**
- BFIT: Circuit-level soft error analysis
- Sequential Simulation with Monitors

**Case Study: CMP Router**
- Analysis
- Efficient Hardening
Chip Multiprocessor (CMP) Router

- Simplified 2 port version of 5 port design\[^{Peh\ 01}\]
- 174 latches, \(\approx 1300\) gates

**Specification**
Every incoming flit must be routed correctly within 11 cycles

**Target**
Select gates or latches to harden to reduce combinational FIT to \(FIT_{TARGET}\)
BFIT analysis of CMP router

Output from BFIT tool:

- Thousands of E observed, but over 94% are SBU
BFIT analysis of CMP router

- Output from BFIT tool:

- Thousands of E observed, but over 94% are SBU
System level masking in CMP

94% of SEU are SBU

- 94% of SEU are SBU
- Find failure probability ($FP_E$) for all SBU

![Graph showing failure probability ($FP_Li$) vs Latch Number]
CMP - all masking factors

Figure: System-level SBU FIT
CMP
Achieving FIT target by hardening gates or latches

Hardening gates
Hardening gates

Assume a hardened gate contributes no FIT
CMP
Achieving FIT target by hardening gates or latches

Hardening gates

Assume a hardened gate contributes no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 81/1300 gates
**Hardening gates**

Assume a hardened gate contributes no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 81/1300 gates

- Pareto optimal coverage if all MBU causes system failure
CMP
Achieving FIT target by hardening gates or latches

Hardening gates

Assume a hardened gate contributes no FIT

▶ To achieve $FIT \leq FIT_{TARGET}$:
  ▶ Harden 81/1300 gates

▶ Pareto optimal coverage if all MBU causes system failure

Hardening latches
CMP
Achieving FIT target by hardening gates or latches

**Hardening gates**

Assume a hardened gate contributes no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 81/1300 gates

  ![Graph showing MBU and SBU distribution]

- Pareto optimal coverage if all MBU causes system failure

**Hardening latches**

Assume a hardened latch captures no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 39/174 latches

- Coverage is Pareto optimal with respect to SBU.
**CMP**
Achieving FIT target by hardening gates or latches

**Hardening gates**

Assume a hardened gate contributes no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 81/1300 gates

- Pareto optimal coverage if all MBU causes system failure

**Hardening latches**

Assume a hardened latch captures no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 39/174 latches

Holcomb, Li, Seshia  
Design as You See FIT  
23
Hardening gates

Assume a hardened gate contributes no FIT

- To achieve \( \text{FIT} \leq \text{FIT}_{\text{TARGET}} \):
  - Harden 81/1300 gates

- Pareto optimal coverage if all MBU causes system failure

Hardening latches

Assume a hardened latch captures no FIT

- To achieve \( \text{FIT} \leq \text{FIT}_{\text{TARGET}} \):
  - Harden 39/174 latches

- Coverage is Pareto optimal with respect to SBU
**Hardening gates**

Assume a hardened gate contributes no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 81/1300 gates

  ➤ Pareto optimal coverage if all MBU causes system failure

**Hardening latches**

Assume a hardened latch captures no FIT

- To achieve $FIT \leq FIT_{TARGET}$:
  - Harden 39/174 latches

  ➤ Coverage is Pareto optimal with respect to SBU

... design as you see FIT
A new method for efficiently analyzing system level impact of circuit level upsets

Can guide cost-effective hardening of circuit level features

Provide designer flexibility to harden either gates or latches

MBU poses threat to reliability

Holcomb, Li, Seshia

Design as You See FIT
A new method for efficiently analyzing system level impact of circuit level upsets
A new method for efficiently analyzing system level impact of circuit level upsets

Can guide cost-effective hardening of circuit level features
A new method for efficiently analyzing system level impact of circuit level upsets
Can guide cost-effective hardening of circuit level features
Provide designer flexibility to harden either gates or latches
A new method for efficiently analyzing system level impact of circuit level upsets

Can guide cost-effective hardening of circuit level features

Provide designer flexibility to harden either gates or latches

MBU poses threat to reliability
Thank You

BFIT

is available at www.eecs.berkeley.edu/~holcomb/bfit.htm
N. Miskov-Zivanov and D. Marculescu.
Modeling and Optimization for Soft-Error Reliability of Sequential Circuits.

N. Miskov-Zivanov and D. Marculescu.
Circuit Reliability Analysis Using Symbolic Techniques.

R. Baumann.
Radiation-induced soft errors in advanced semiconductor technologies.

P. Hazucha and C. Svensson.
Impact of CMOS technology scaling on the atmospheric neutron soft error rate.

S. Krishnaswamy, et al.
On the role of timing masking in reliable logic circuit design.

L.-S. Peh.

Computing the soft error rate of a combinational logic circuit using parameterized descriptors.

S. S. Mukherjee et al.
A systematic methodology to compute the architectural vulnerability factors for a high-perf. microprocessor.
In MICRO 2003, pages 29–40.

S. A. Seshia, et al.
Verification-guided soft error resilience.
In DATE 2007, pages 1442–1447.

P. Shivakumar, et al.
Modeling the effect of technology trends on soft error rate of combinational logic.
DSN’02, pp. 389-398.

B. Zhang, et al.
FASER: fast analysis of soft error susceptibility for cell-based designs.

M. Zhang and N. R. Shanbhag.
Soft-error-rate-analysis (SERA) methodology.