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

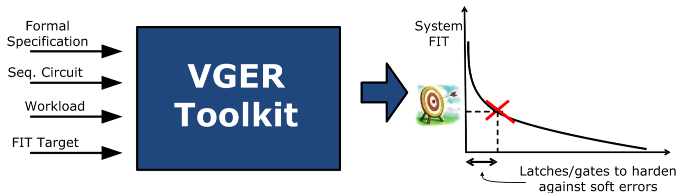
- Logic FIT may increase 9 orders of magnitude from 1992-2011 [Shivakumar 02]
- Logic FIT approaching memory FIT around 100nm [Shivakumar 02, Baumann 05]

Circuit-level hardening techniques exist, but have costs

Our contribution

the verification guided error resilience methodology

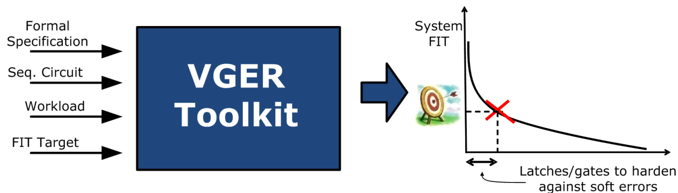
- Use formal specifications to capture system-level correctness



Our contribution

the verification guided error resilience methodology

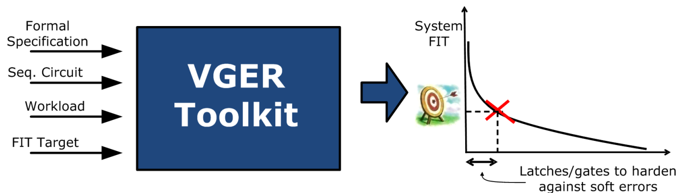
- Use formal specifications to capture system-level correctness
- Use verification to analyze system-level impact of circuit-level upsets



Our contribution

the verification guided error resilience methodology

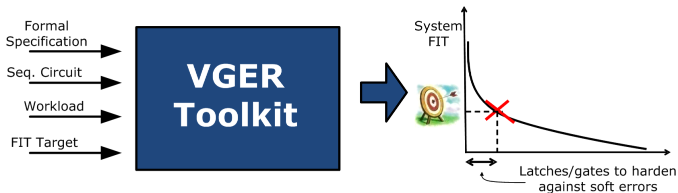
- Use formal specifications to capture system-level correctness
- Use verification to analyze system-level impact of circuit-level upsets
- Guide efficient circuit hardening techniques



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

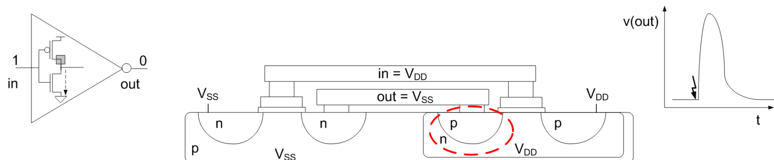
Introduction
- Single Event Upset
- Circuit-level masking
- System-level masking

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)

Circuit-level masking

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

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]

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

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

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

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

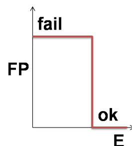
Related work

- 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

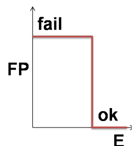
- Architectural Vulnerability Factor [Mukherjee 03]
  - Find probability of a bit flip leading to incorrect future execution
  - Requires detailed architecture model
- Output equivalence
  - Find probability of a bit flip leading to incorrect outputs
  - Model state using Markov Chain theory [Miskov-Zivanov 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

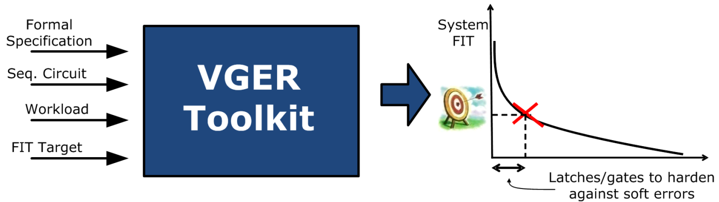
Outline

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

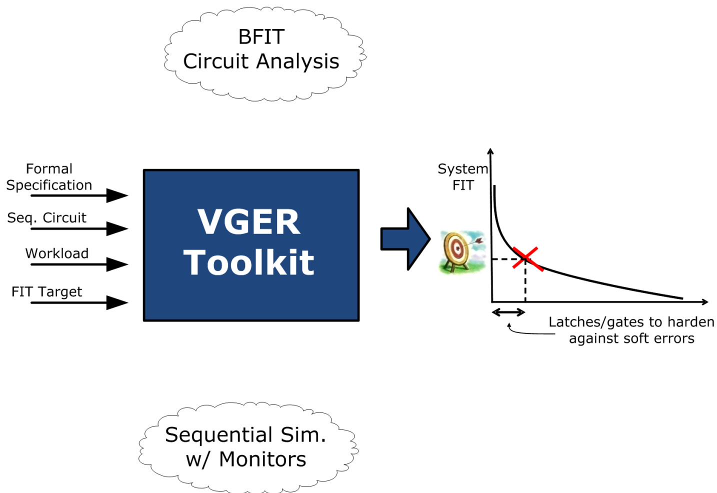
VGER Toolkit
- BFIT: Circuit-level soft error analysis
- Sequential Simulation with Monitors

Case Study: CMP Router
- Analysis
- Efficient Hardening

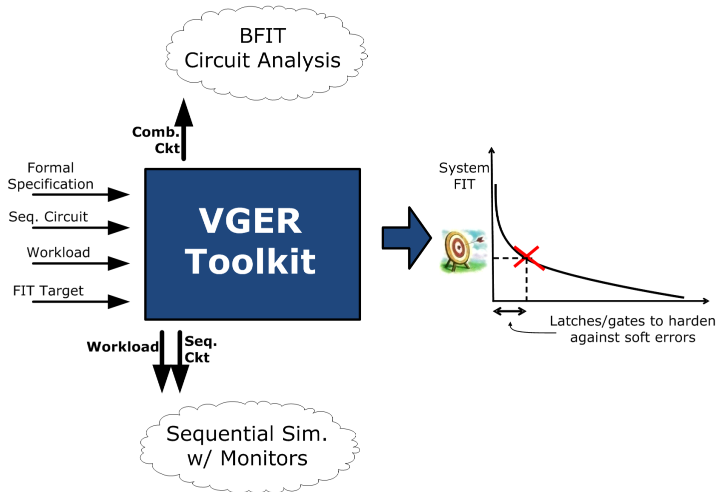
VGER toolkit



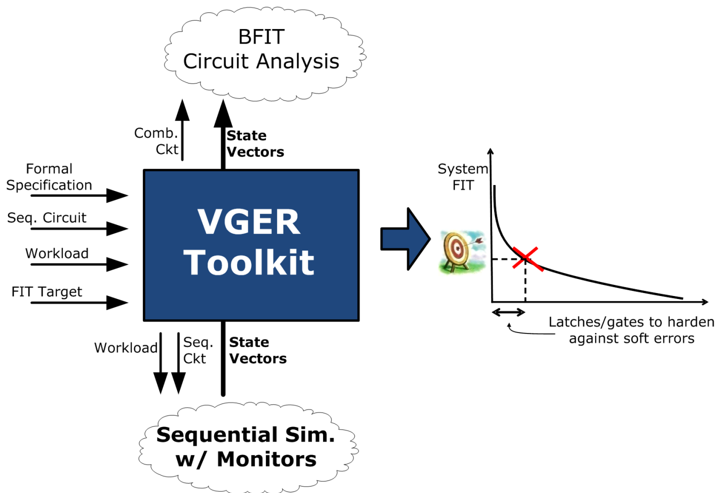
VGER toolkit



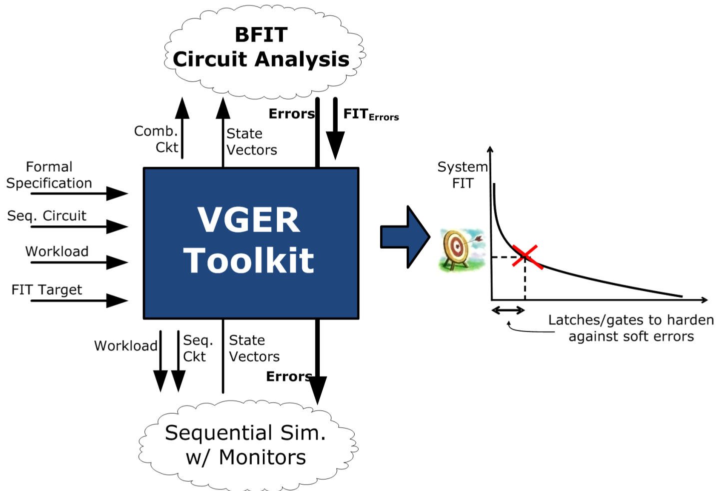
VGER toolkit



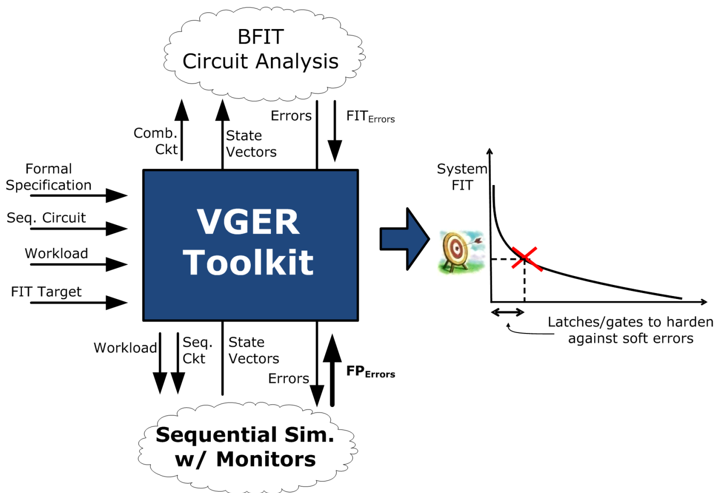
VGER toolkit



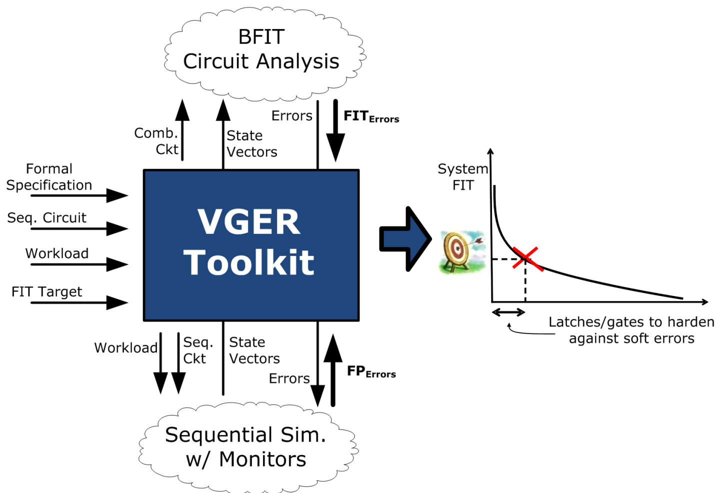
VGER toolkit



VGER toolkit



VGER toolkit



BFIT circuit level analysis tool

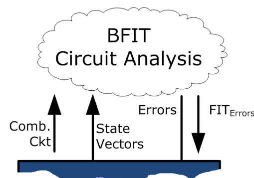
- Open source C++ simulation tool for combinational logic circuits
- Based on Nangate 45nm open cell library

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)

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

$R_g(q,t) \in \Re$

probability of observing strike q, t at gate g

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

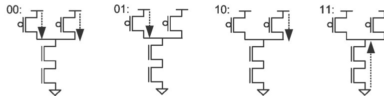
conditional probability of upset in set E of latches, given a strike q, t at gate g

  - **Encompasses logical, electrical, timing masking**

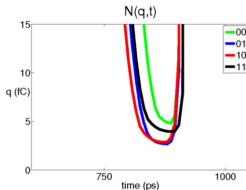
BFIT approach to Masking

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

- Inputs determine drive strength



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



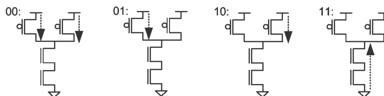
00	01	10	11
5.0E-6	7.5E-6	16.1E-6	2.9E-6

Table: FIT vs gate input

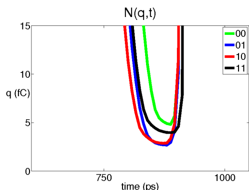
BFIT approach to Masking

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

- Inputs determine drive strength



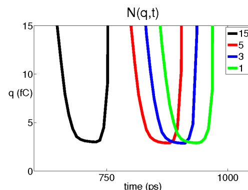
- Input determines shape of $N(q, t)$



00	01	10	11
5.0E-6	7.5E-6	16.1E-6	2.9E-6

Table: FIT vs gate input

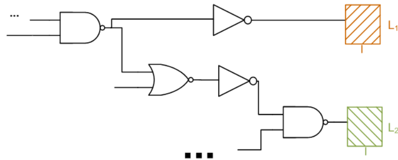
- Path delay is time-shift to $N(q, t)$



15	5	3	1
6.2E-6	6.4E-6	6.4E-6	6.7E-6

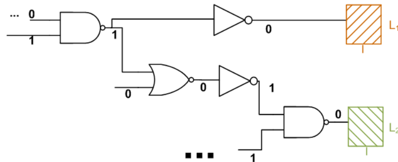
Table: FIT vs path length

Demonstration of BFIT algorithm



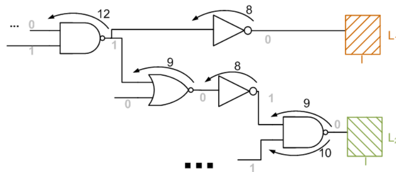
Demonstration of BFIT algorithm

1. Forward propagate input vector in levelized DAG



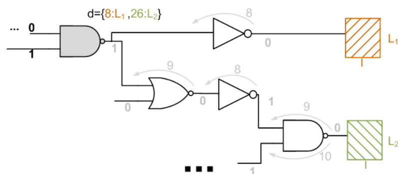
Demonstration of BFIT algorithm

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



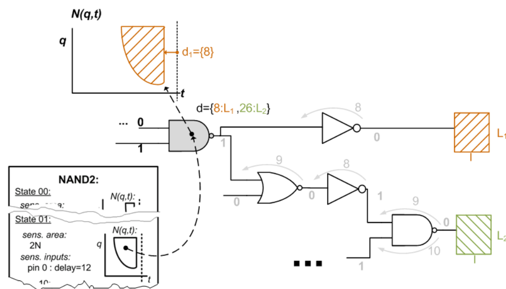
Demonstration of BFIT algorithm

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 $N_{g \to E}(q, t)$ using:
   - List of path delays and terminating latches
   - Gate input state and load capacitance
   - Cell precharacterization



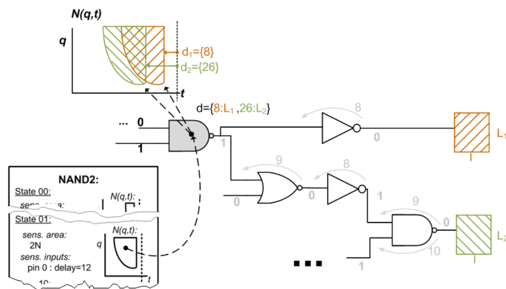
Demonstration of BFIT algorithm

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 $N_{g \to E}(q, t)$ using:
   - List of path delays and terminating latches
   - Gate input state and load capacitance
   - Cell precharacterization

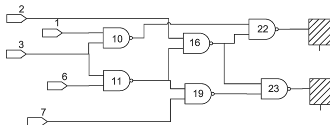


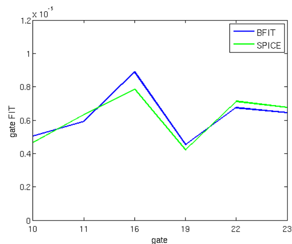
Demonstration of BFIT algorithm

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 $N_{g \to E}(q, t)$ using:
   - List of path delays and terminating latches
   - Gate input state and load capacitance
   - Cell precharacterization



BFIT results





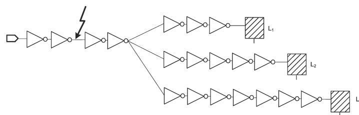
Circuit	INPUTS	LATCHES	GATES	RUNTIME (s/1k vectors)	FIT
s5378	214	179	3232	35	3.27e-3
s9234	247	228	7230	68	1.06e-2
s13207	700	669	10277	136	1.77e-2
s15850	611	597	12712	207	2.24e-2
s38417	1664	1636	28223	451	5.62e-2
s38584	1464	1452	28854	1311	4.85e-2
s35932	1763	1728	23012	204	3.99e-2

BFIT results
multiple bit upsets

- Do MBU occur?

BFIT results
multiple bit upsets

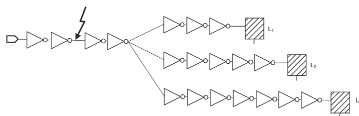
- Do MBU occur?



	BFIT	HSPICE
SBU	8.6e-7	7.4e-7
MBU	9.0e-7	12.6e-7
total FIT	1.7e-6	2.0e-6

BFIT results
multiple bit upsets

- Do MBU occur?

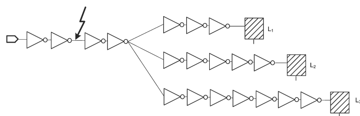


	BFIT	HSPICE
SBU	8.6e-7	7.4e-7
MBU	9.0e-7	12.6e-7
total FIT	1.7e-6	2.0e-6

- Is MBU a concern?

BFIT results
multiple bit upsets

- Do MBU occur?



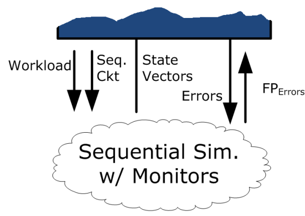
	BFIT	HSPICE
SBU	8.6e-7	7.4e-7
MBU	9.0e-7	12.6e-7
total FIT	1.7e-6	2.0e-6

- Is MBU a concern?

Circuit	%SBU	%MBU
s5378	0.8388	0.1612
s9234	0.8625	0.1375
s13207	0.9379	0.0621
s15850	0.9148	0.0852
s38417	0.8812	0.1188
s38584	0.9538	0.0462
s35932	0.9960	0.0030

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

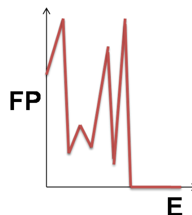


Inputs

- List of formal specifications
- List of circuit errors E

output

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



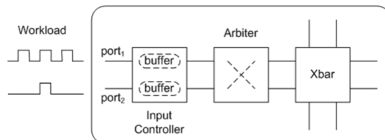
Outline

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, ≈1300 gates

Specification

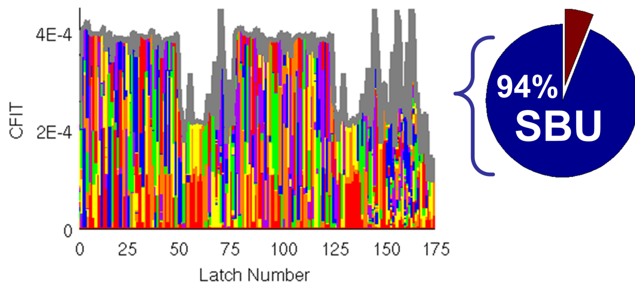
Every incoming flit must be routed correctly within 11 cycles

Target

Select gates or latches to harden to reduce combinational FIT to FIT_{TARGET}

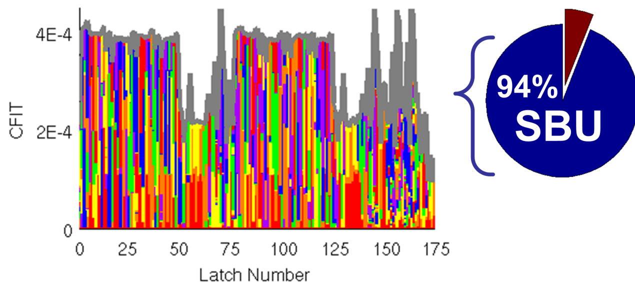
BFIT analysis of CMP router

- Output from BFIT tool:



BFIT analysis of CMP router

- Output from BFIT tool:

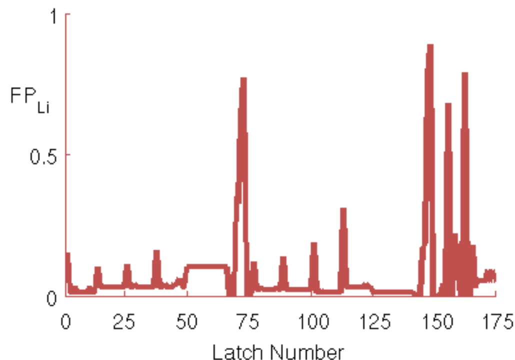


- Thousands of E observed, but over 94% are SBU

System level masking in CMP

94% of SEU are SBU

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



CMP - all masking factors

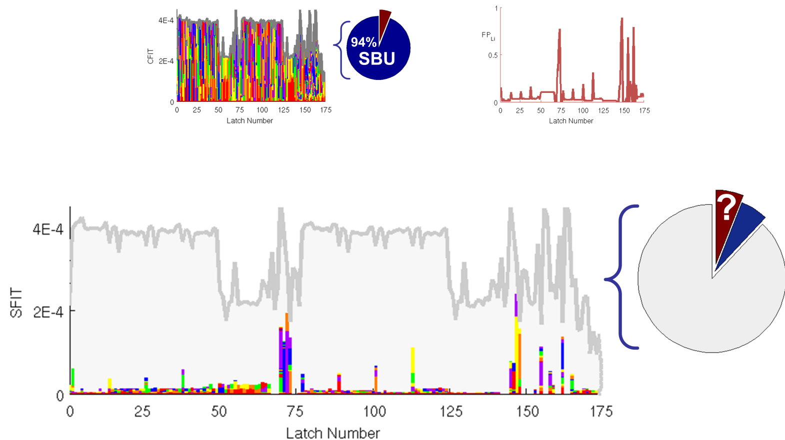


Figure: System-level SBU FIT

CMP
Achieving FIT target by hardening gates or latches

Hardening gates

CMP
Achieving FIT target by hardening gates or latches

Hardening gates

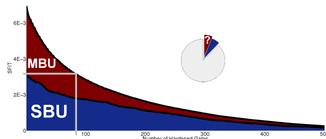
Assume a hardened gate contributes no FIT

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

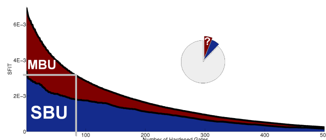


CMP
Achieving FIT target by hardening gates or latches

Hardening gates

Assume a hardened gate contributes no FIT

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



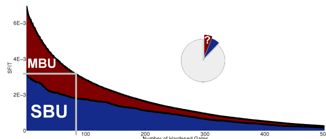
- Pareto optimal coverage if all MBU causes system failure

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

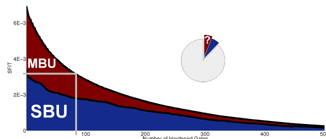
Hardening latches

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



- Pareto optimal coverage if all MBU causes system failure

Hardening latches

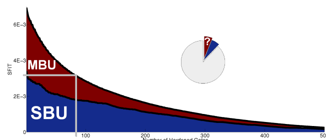
Assume a hardened latch captures no FIT

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

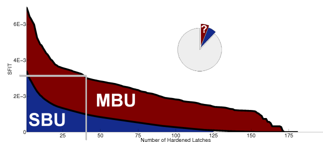


- Pareto optimal coverage if all MBU causes system failure

Hardening latches

Assume a hardened latch captures no FIT

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

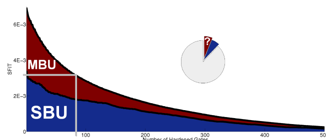


CMP
Achieving FIT target by hardening gates or latches

Hardening gates

Assume a hardened gate contributes no FIT

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

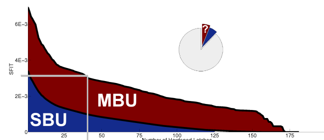


- Pareto optimal coverage if all MBU causes system failure

Hardening latches

Assume a hardened latch captures no FIT

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



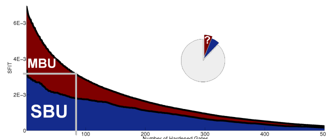
- Coverage is Pareto optimal with respect to SBU

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

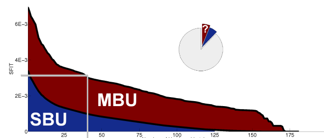


- Pareto optimal coverage if all MBU causes system failure

Hardening latches

Assume a hardened latch captures no FIT

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



- Coverage is Pareto optimal with respect to SBU

. . . design as you see FIT

Summary

Summary

- A new method for efficiently analyzing system level impact of circuit level upsets

Summary

- A new method for efficiently analyzing system level impact of circuit level upsets
- Can guide cost-effective hardening of circuit level features

Summary

- 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

Summary

- 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.
IEEE Trans. on CAD of Integ. Circ. and Sys., pages 803–816, May 2008.

N. Miskov-Zivanov and D. Marculescu.

Circuit Reliability Analysis Using Symbolic Techniques.
IEEE Trans. on CAD of Integ. Circ. and Sys., pages 2638–2649, Dec. 2006.

R. Baumann.

Radiation-induced soft errors in advanced semiconductor technologies.
IEEE Trans. Device and Materials Reliability, 5(3):305–316, Sept. 2005.

P. Hazucha and C. Svensson.

Impact of CMOS technology scaling on the atmospheric neutron soft error rate.
IEEE Trans. Nuclear Science, 47(6):2586–2594, Dec 2000.

S. Krishnaswamy, et al.

On the role of timing masking in reliable logic circuit design.
DAC 2008, pages 924–929.

L.-S. Peh.

Flow Control and Micro-Arch. Mechanisms for Extending the Performance of Interconnection Networks.
PhD thesis, Stanford University, August 2001.

R. R. Rao, et al.

Computing the soft error rate of a combinational logic circuit using parameterized descriptors.
IEEE Trans. on CAD of Integ. Circ. and Sys., 26(3):468–479, 2007.

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.
ISQED 2006, pages 755-760.

M. Zhang and N. R. Shanbhag.

Soft-error-rate-analysis (SERA) methodology.
IEEE Trans. on CAD of Integrated Circuits and Systems, 25(10):2140–2155, 2006.