SWAT-Sim: Accurate Microarchitecture-Level Fault Models

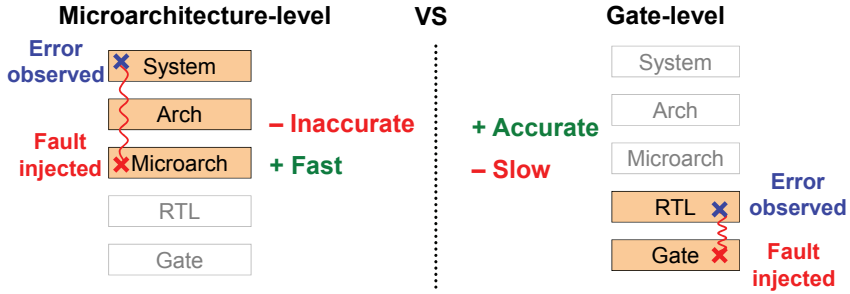
Manlap Li, Pradeep Ramachandran, Rahmet Ulya Karpuzcu, Siva Kumar Sastry Hari, Sarita Adve
University of Illinois, Urbana Champaign

Resilient Theme, Task # 1.2.2.5

GSRC Annual Symposium
Sep 29-30, 2008

Need for Accurate Microarchitecture-Level Permanent Fault Models

Current microarchitecture-level (latch) fault model insufficient for logic



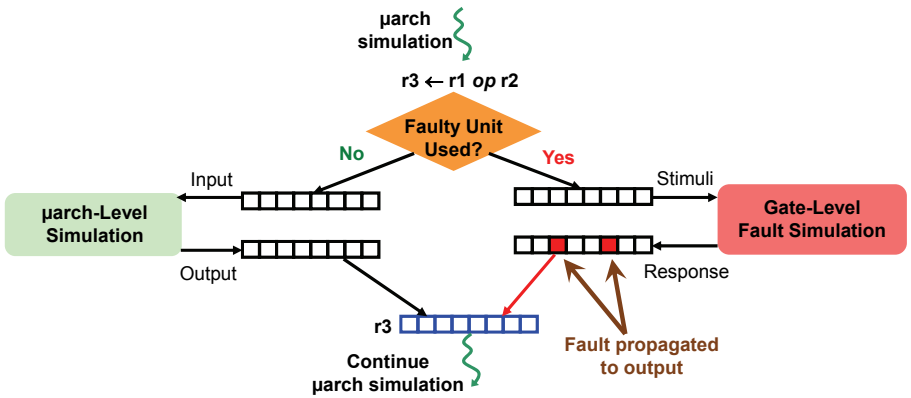
SWAT-Sim: Accurate and Fast Hierarchical Fault Simulator

On-demand* gate-level simulation of faulty component *online

On-demand – Gate-level sim when faulty unit utilized ⇒ low overhead

Online – Hard errors affect future activations thru software (feedback loop)

– Coupled gate-level & µarch sim ⇒ real-world system behavior



Experimental Methodology to compare Fault Models using SWAT-Sim

Fault injection platform: full-system (Simics) simulation of superscaler µarch (GEMS)

Workloads: SPEC2K running on top of Solaris 9

Fault models: µarch-level stuck-at faults, gate-level (with NC-Sim) stuck-at and delay faults

Faulty components: Int ALU (ALU) and Address Generation Unit (AGEN) (from OpenSPARC)

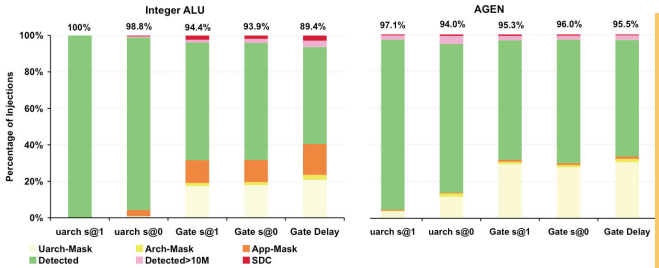
Metric for accuracy: Coverage & latency (not shown) of SWAT detectors (system-level effects)

Are There Speed Advantages to SWAT-Sim?

100k times speed of pure gate-level simulation, similar fault modeling accuracy

< 2x avg slowdown from µarch-level simulation, but higher accuracy

Are Existing µarch-level Fault Models Accurate?



Existing Fault Models inaccurate

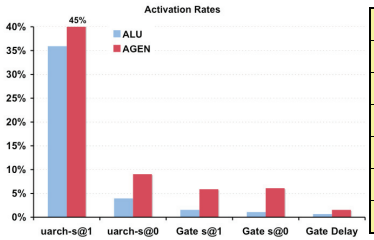
Coverage:

- µarch s@1 too aggressive
- ALU µarch s@0 inaccurate
- AGEN µarch s@0 close to gate-level

Latency:

- µarch s@ inaccurate

What are Reasons for the Differences?



Bit-flips	µarch	Gate s@0	Delay
Int ALU			
Single	100%	84.4%	90.4%
Multiple	0%	15.6%	9.6%
AGEN			
Single	100%	75.5%	90.5%
Multiple	0%	24.5%	9.5%

µarch vs. Gate-Level

- Different activation rates
- Ability to corrupt multi bits

Can Accurate µarch-Level Fault Models be Derived?

If so, use only µarch sim ⇒ fast!

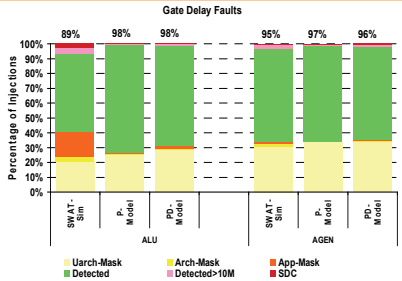
Probabilistic models from SWAT-Sim data

Collect *number*, *pattern*, and *direction of flips*

More accurate than µarch stuck-at models

Only AGEN delay models do well

Rest are inaccurate



Conclusions + Future Work

SWAT-Sim propagates faults using real-world functional vectors

Observe system-level effects of gate-level faults at low overheads

Difficult to derive accurate µarch fault models ⇒ need SWAT-Sim!

Future work: complex gate-level timing faults (e.g., path delay), interfacing larger modules