Post-Silicon Bug Diagnosis with Inconsistent Executions

Andrew DeOrio
Daya Shanker Khudia
Valeria Bertacco

University of Michigan

Impact of errors

- Functional bugs

$475 M

17 Jan 1995	**FDIV bug:** Intel announces a pre-tax charge for re

- Electrical failures

- Transistor faults

$1 B

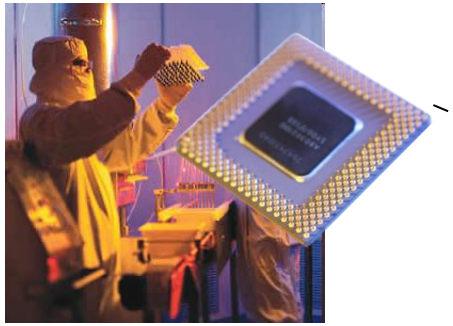
tom's hardware

Sandy Bridge Bug 2X Costly as Pentium FDIV Bug

Post-silicon validation

Pre-Silicon

Post-Silicon



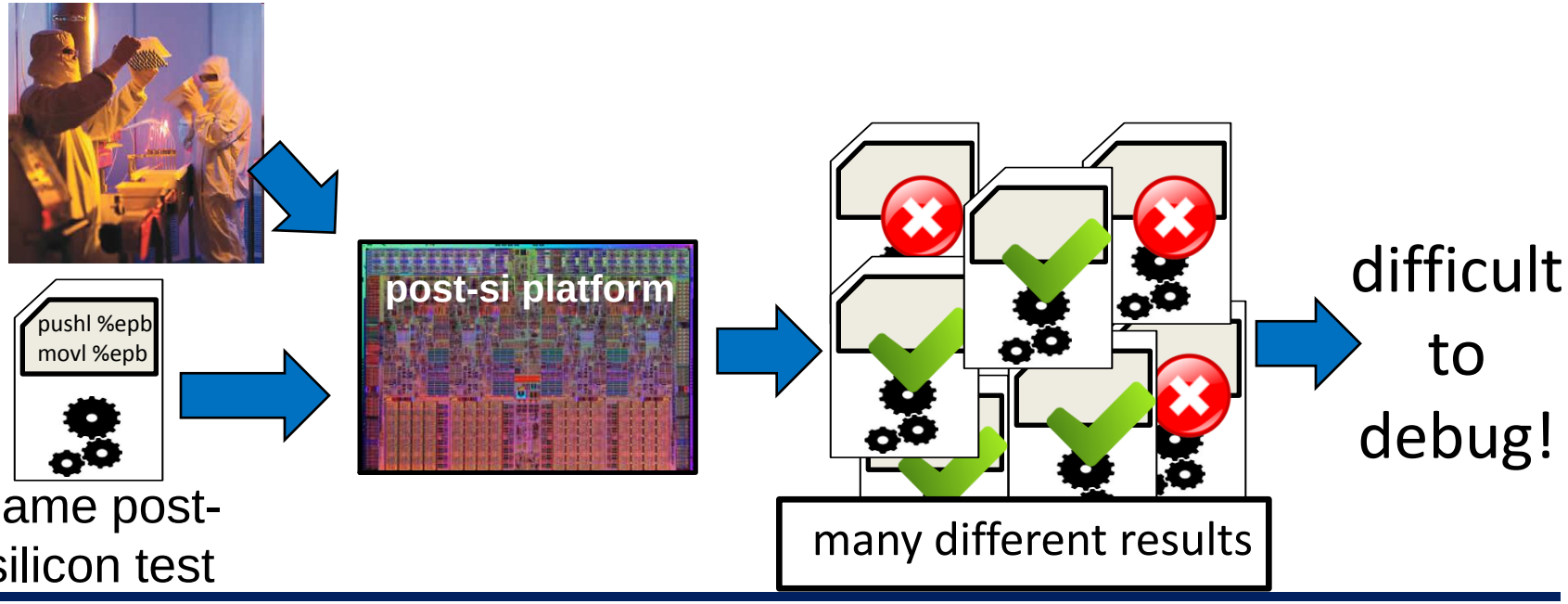
Product

Debug prototypes before shipment

+ Fast prototypes	- Poor observability
+ High coverage	- Slow off-chip transfer
+ Test full system	- Noisy
+ Find deep bugs	- Intermittent bugs

Post-silicon bugs

- Intermittent post-silicon bugs are challenging
  - A same test does not expose the bug in every run
  - Each run exhibits different behaviors
- **Our goal: locate intermittent bugs**

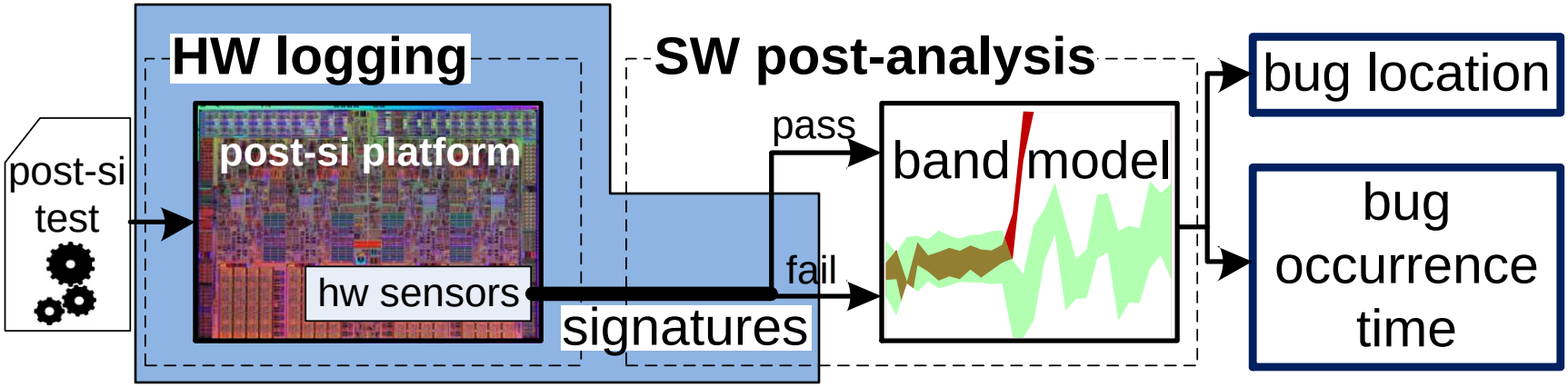


Post-silicon debugging

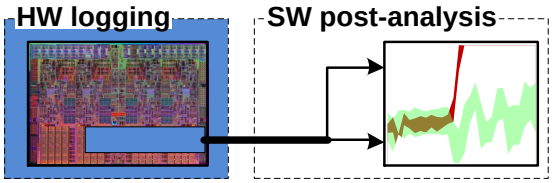
- Scan chains, logic analyzers
  [Whetsel 1991, Abramovici 2006, Dahlgren 2003]
  - Limited observability
  - Large manual effort
- Processor-core specific debugging
  [Park 2009]
  - Limited areas of chip
  - Limited time to catch bug
- Deterministic replay
  [Gao 2009, Li 2010, Yang 2008]
  - HW/performance overhead
  - Perturbation may prevent bug manifestation

BPS: “Bug Positioning System”

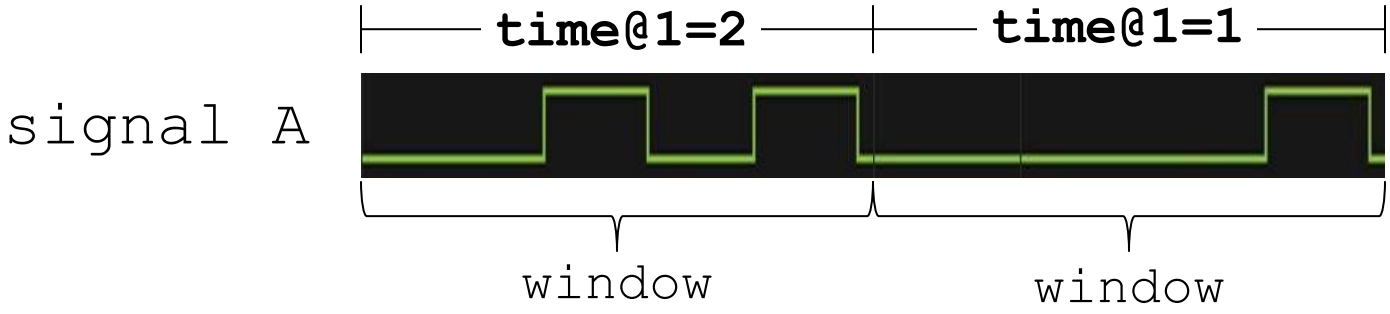
- Localize failures
  - Time (cycle) and space (signals)
- Tolerate non-repeatable executions
  - Statistical approach
- Scalable, adaptable to many HW subsystems



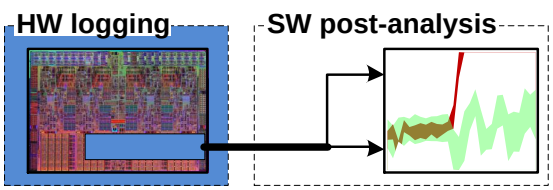
Signatures



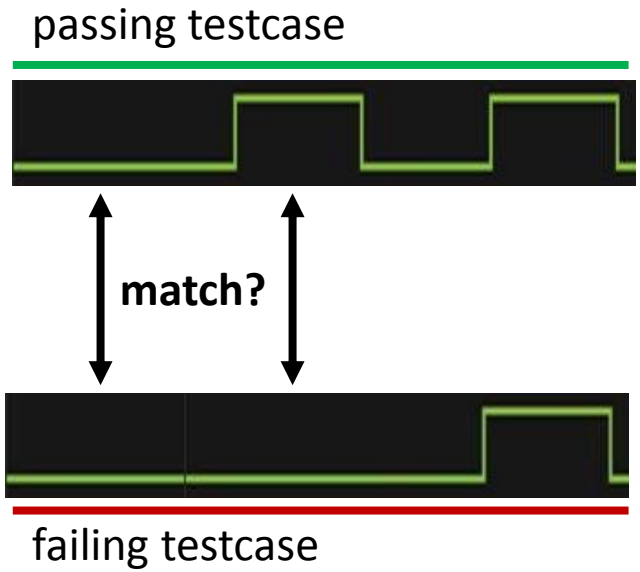
- Goal: summarize signal value
- Encodings (hamming, CRC, etc.)
  - Large hardware
  - Small change in input -> large change in output
- Counting schemes (time@1, toggles)



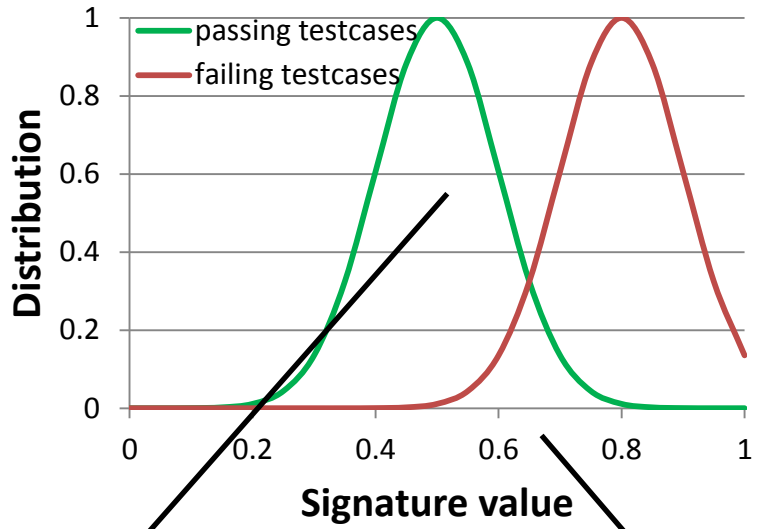
Statistical approach



traditional debugging



statistical debugging



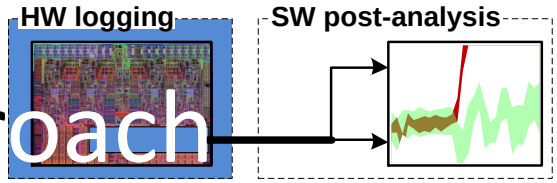
distribution of signature values: same test can yield different results

time@1

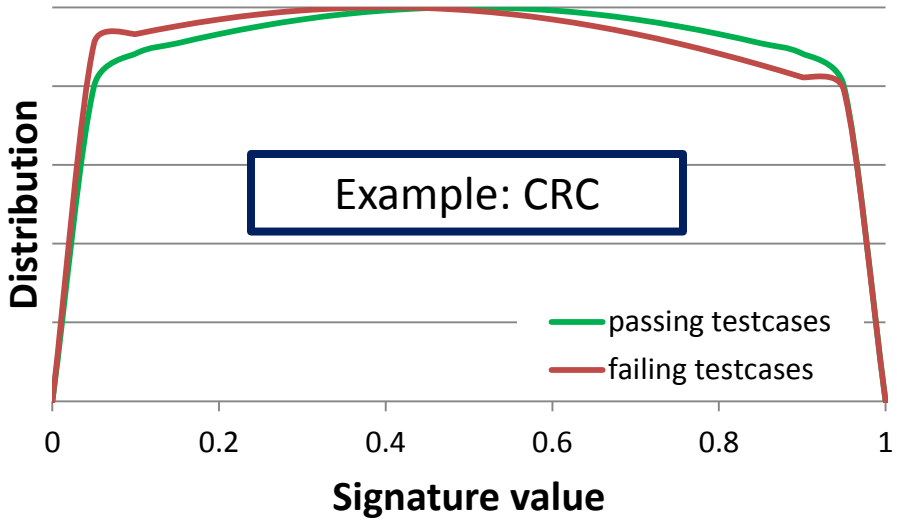
---

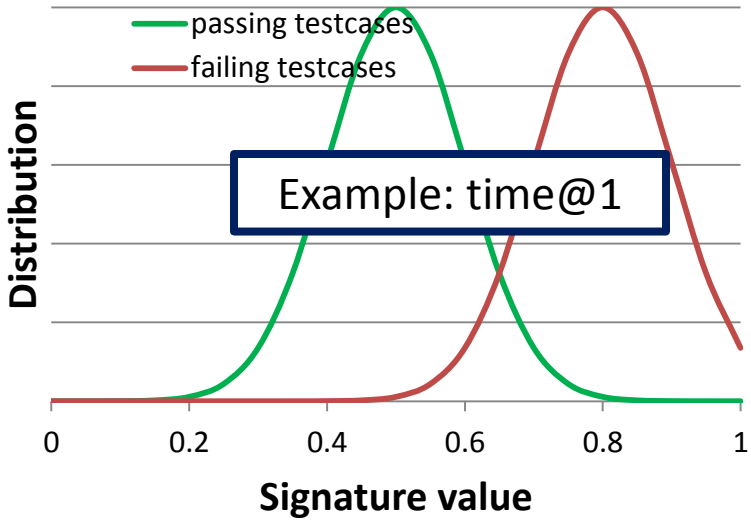
window size

Signatures for statistical approach

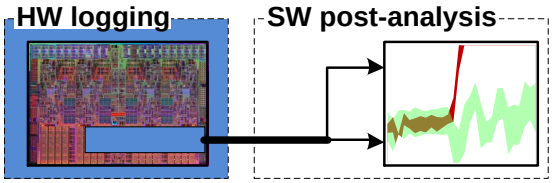


- Characterize populations of signatures
- Statistical separation between noise and bug

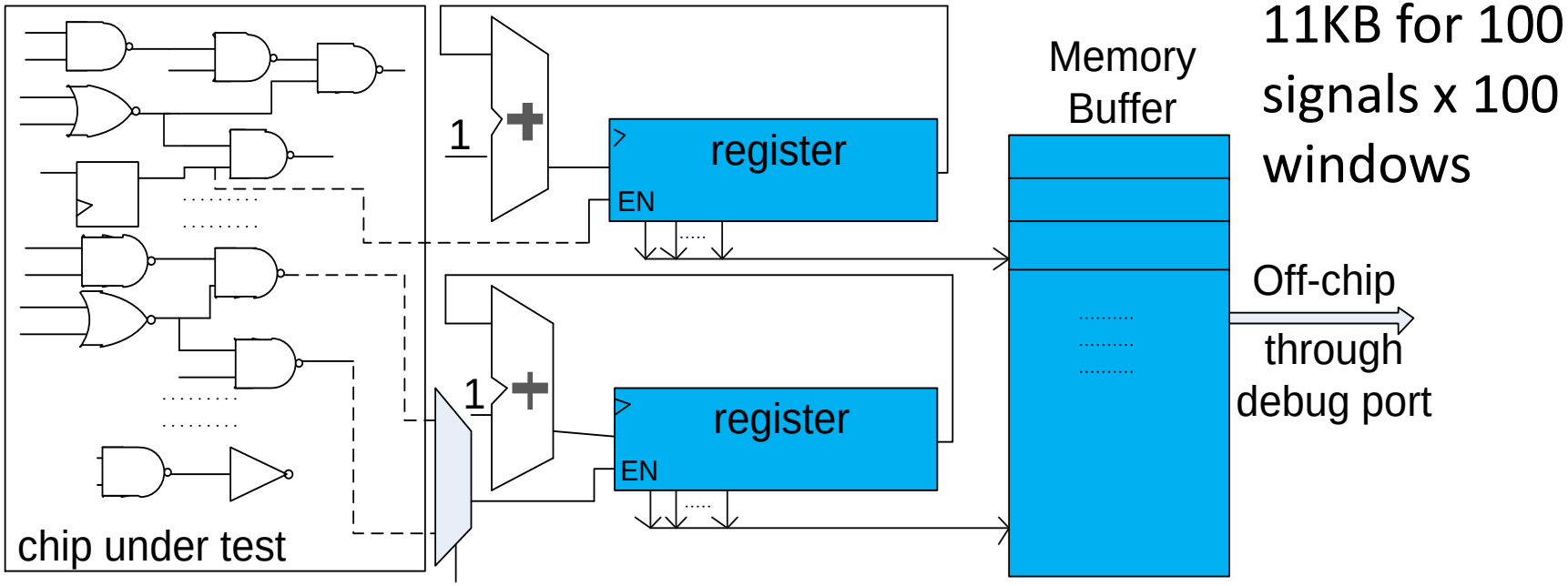




Signature hardware

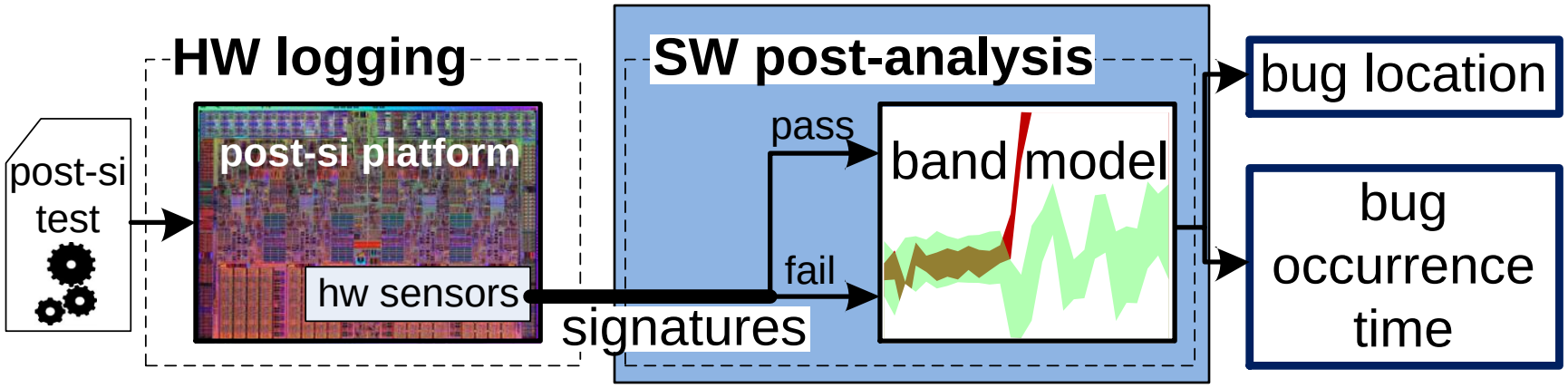


- Measure `time@1`
- Use custom hardware or reuse existing debug infrastructure

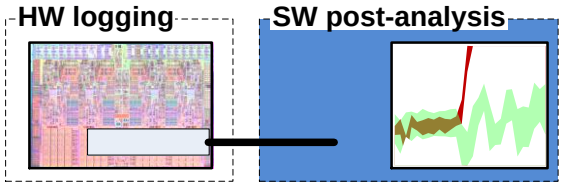


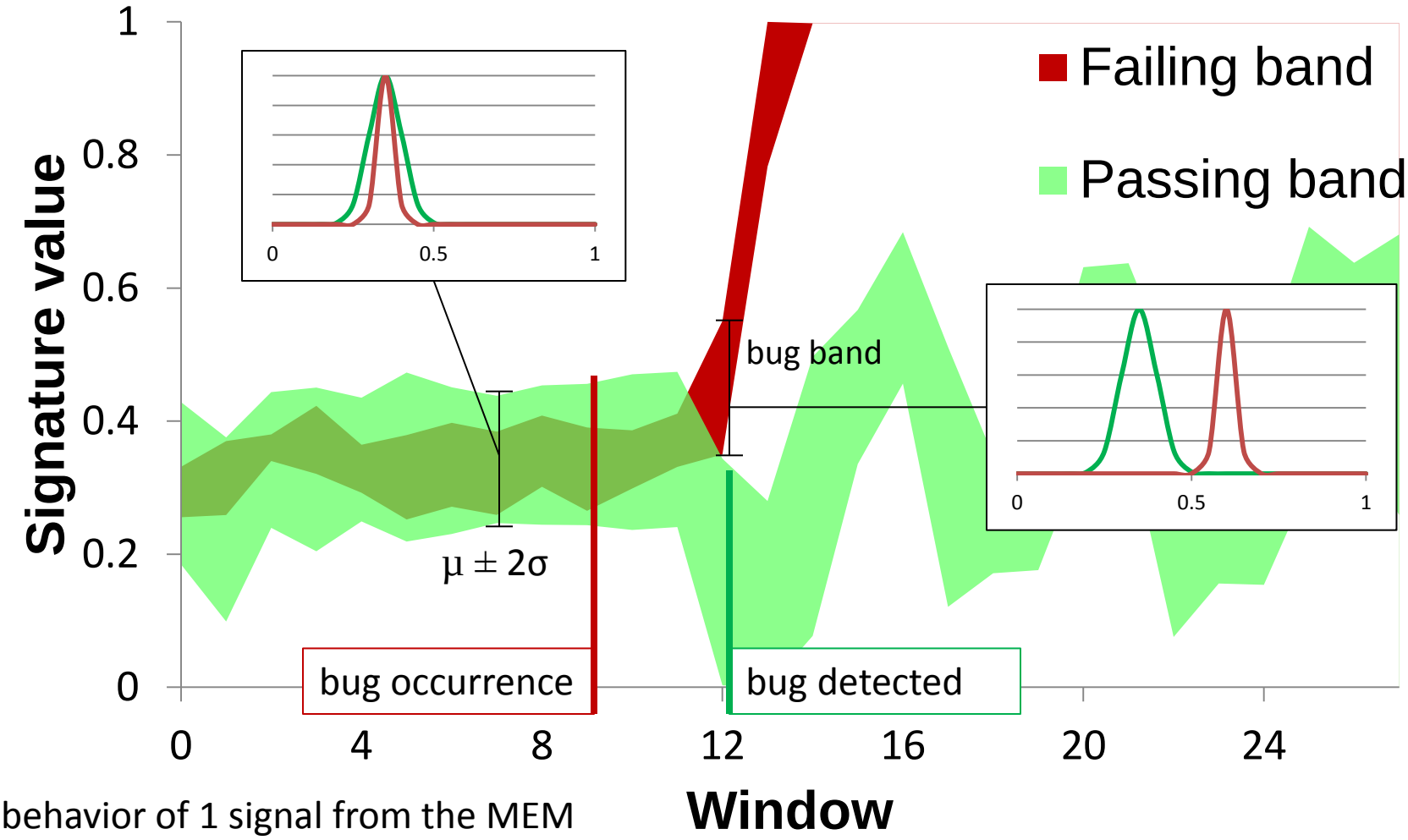
BPS: “Bug Positioning System”

1. Hardware logging
2. Software post-analysis



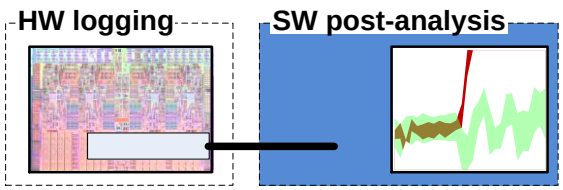
Bug band model

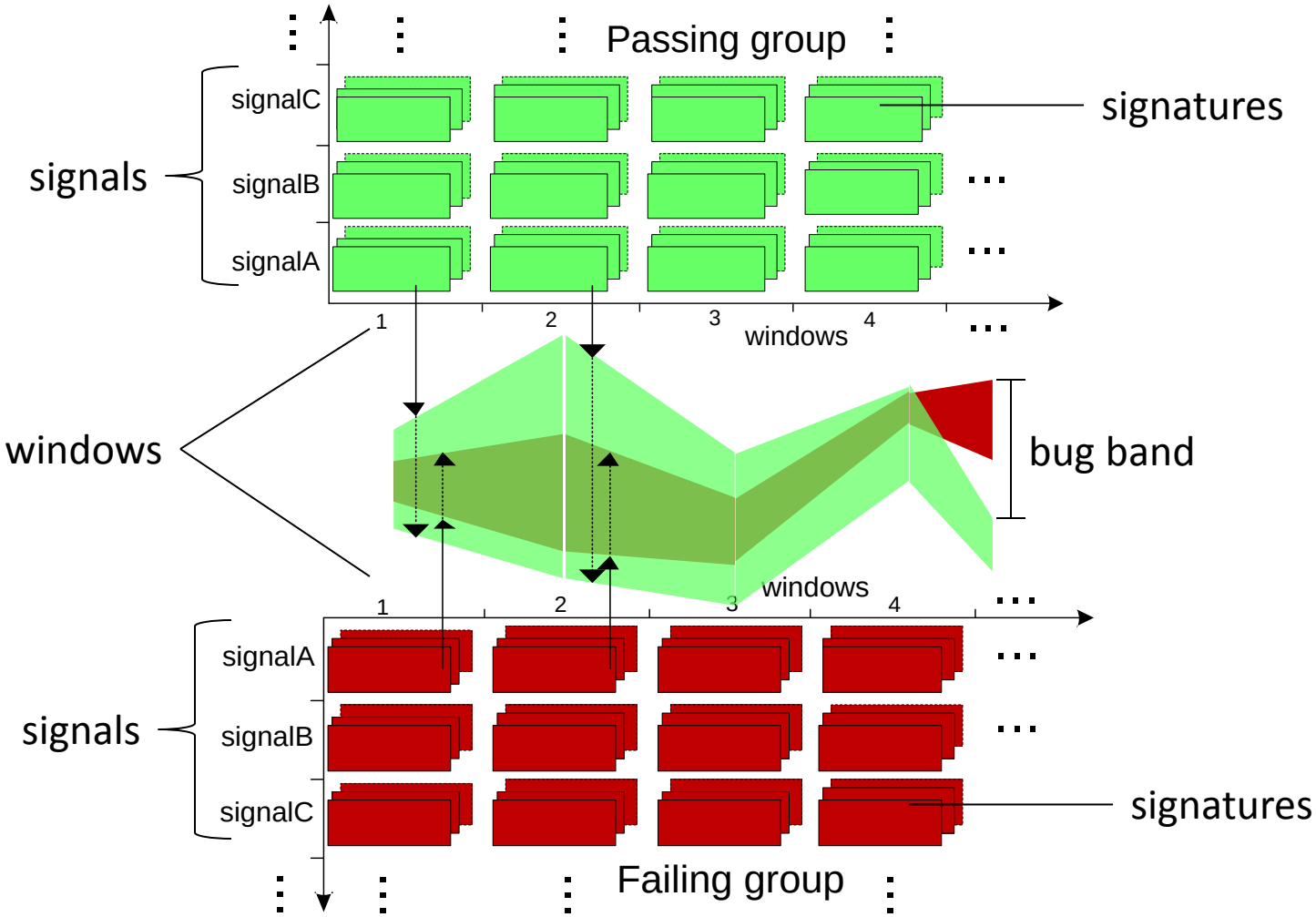




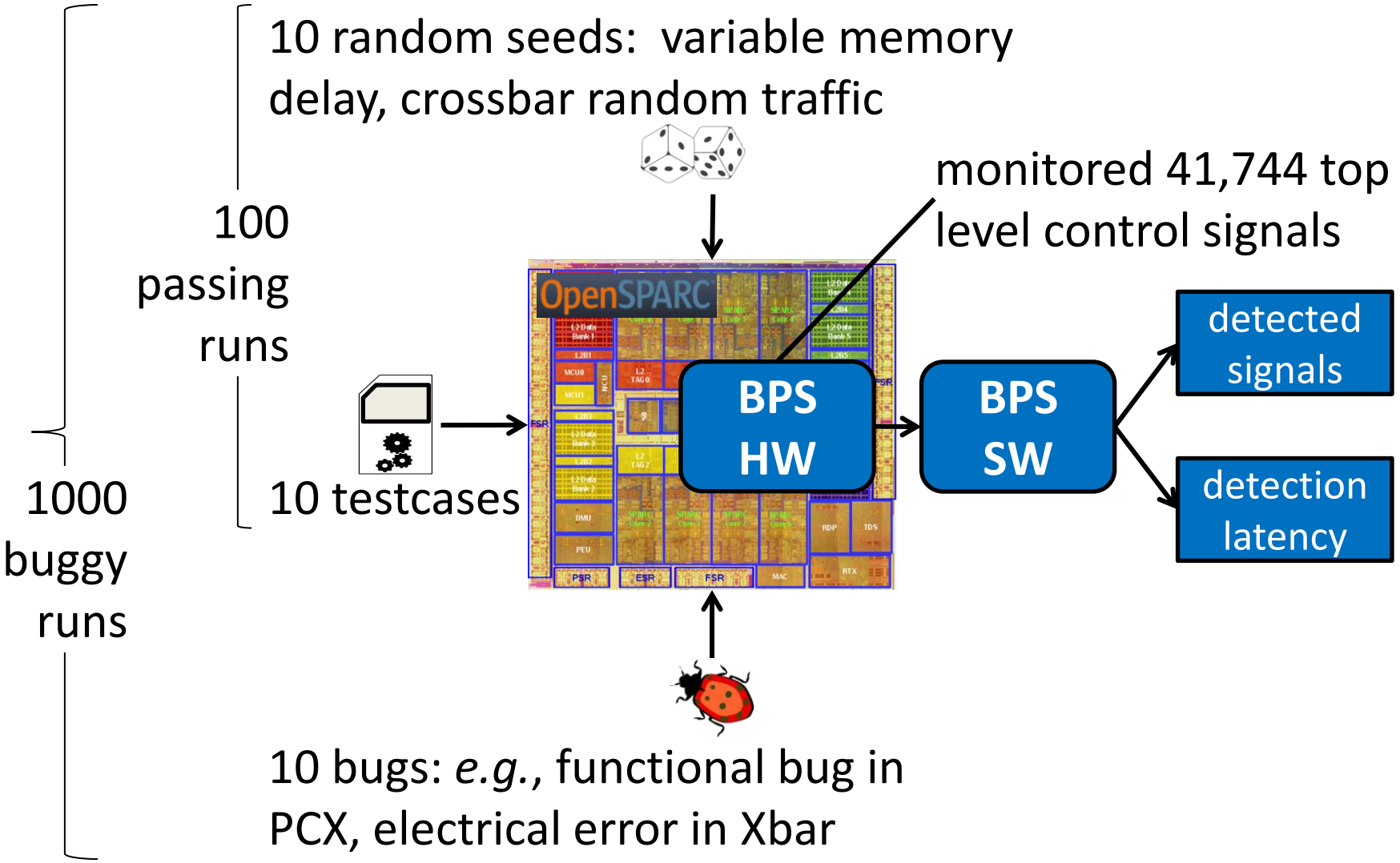
behavior of 1 signal from the MEM stage of a 5-stage pipeline processor

SW post-analysis

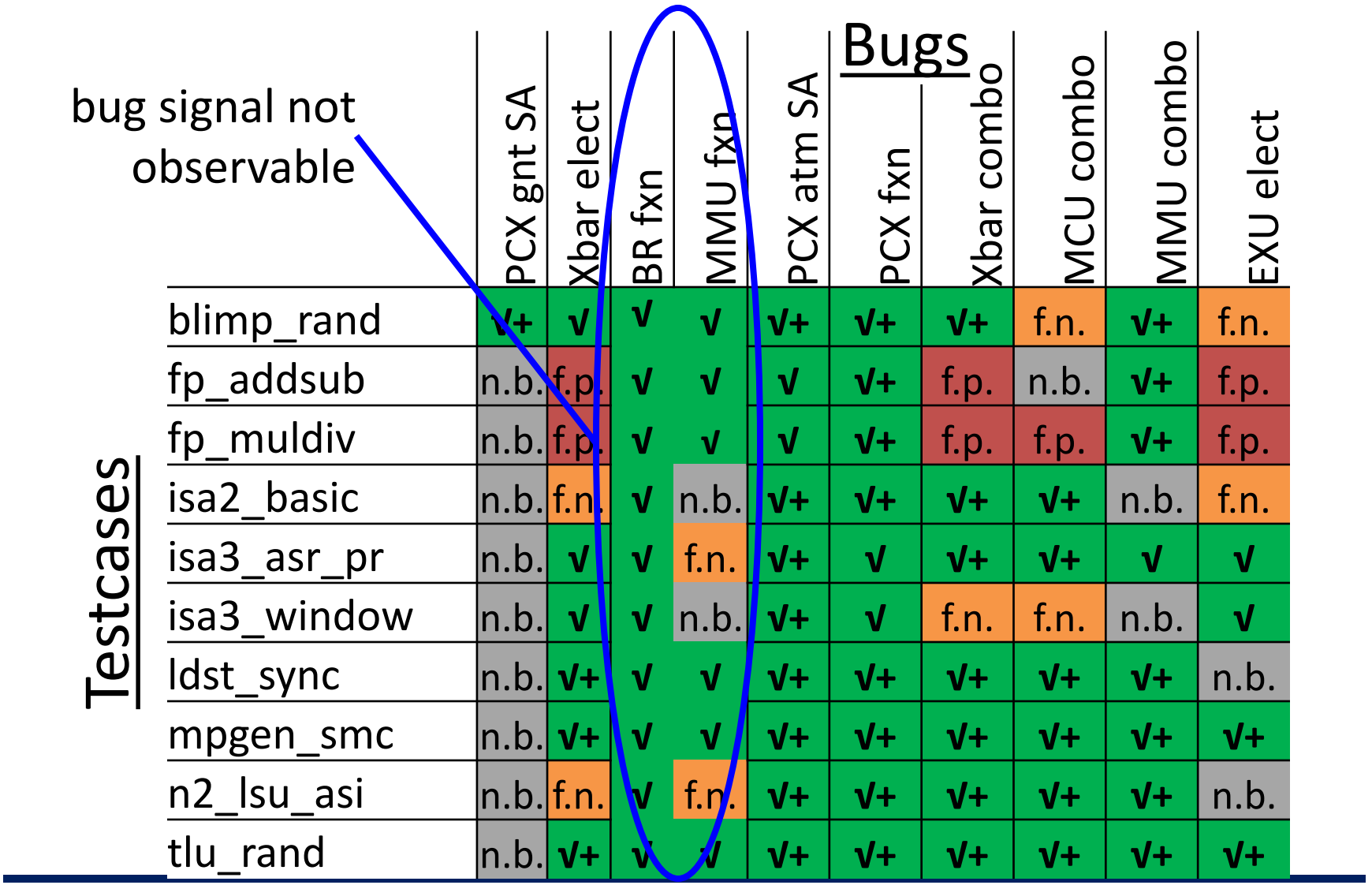




Experimental setup



Signal Localization



Signal Localization

3 noisy signals excited by floating point benchmarks

Testcases	Bugs: PCX gnt SA	Xbar elect	BR fxn	MMU fxn	PCX atm SA	PCX fxn	Xbar combo	MCU combo	MMU combo	EXU elect
blimp_rand	√+	√	√	√	√+	√+	√+	f.n.	√+	f.n.
fp_addsub	n.b.	f.p.	√	√	√	√+	f.p.	n.b.	√+	f.p.
fp_muldiv	n.b.	f.p.	√	√	√	√+	f.p.	f.p.	√+	f.p.
isa2_basic	n.b.	f.n.	√	n.b.	√+	√+	√+	√+	n.b.	f.n.
isa3_asr_pr	n.b.	√	√	f.n.	√+	√	√+	√+	√	√
isa3_window	n.b.	√	√	n.b.	√+	√	f.n.	f.n.	n.b.	√
ldst_sync	n.b.	√+	√	√	√+	√+	√+	√+	√+	n.b.
mpgen_smc	n.b.	√+	√	√	√+	√+	√+	√+	√+	√+
n2_lsu_asi	n.b.	f.n.	√	f.n.	√+	√+	√+	√+	√+	n.b.
tlu_rand	n.b.	√+	√	√	√+	√+	√+	√+	√+	√+

n.b. no bug √ found √+ exact signal f.p. false pos. f.n. false neg.

Signal Localization

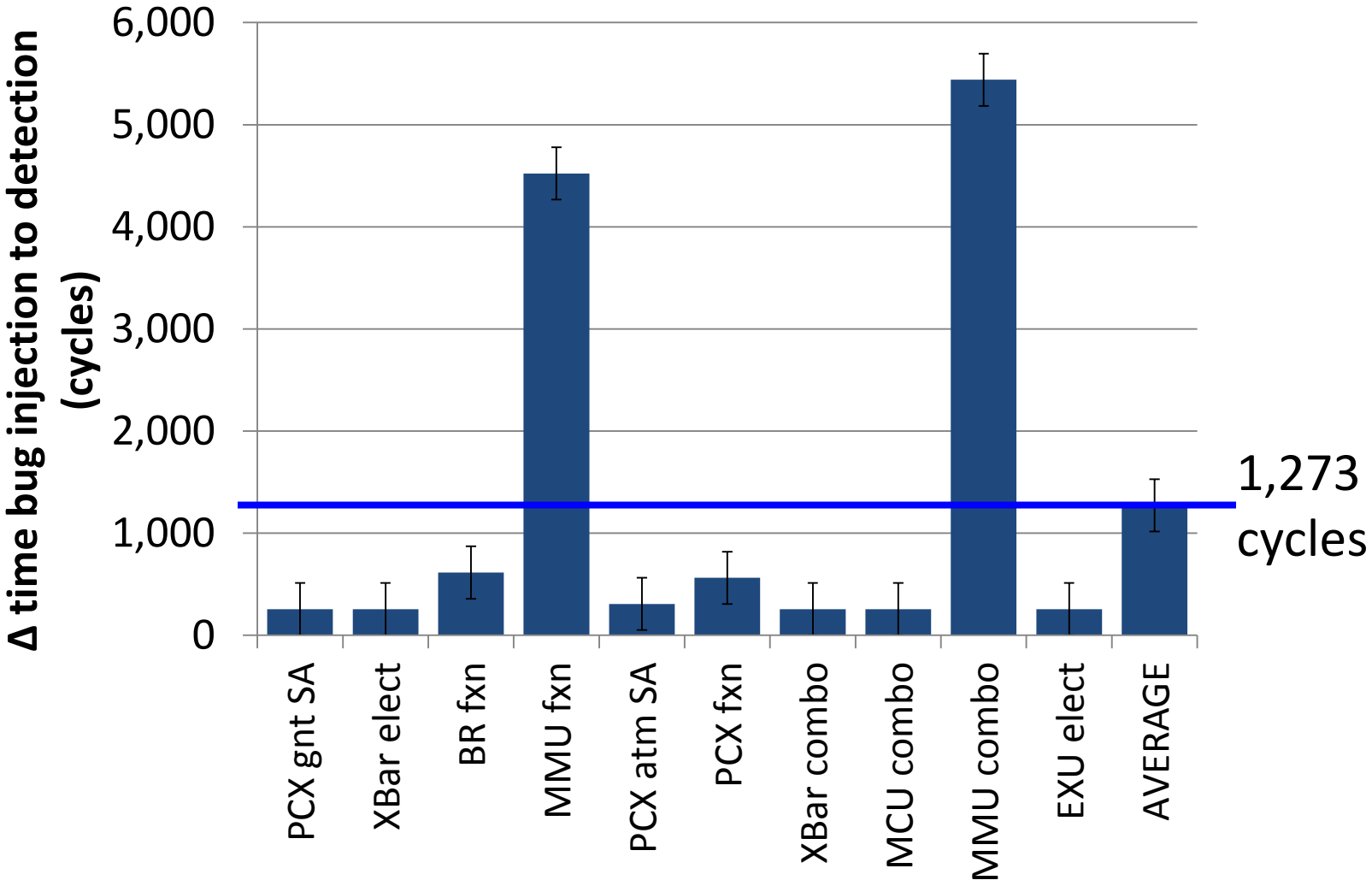
wider effects, easier to catch

Bugs

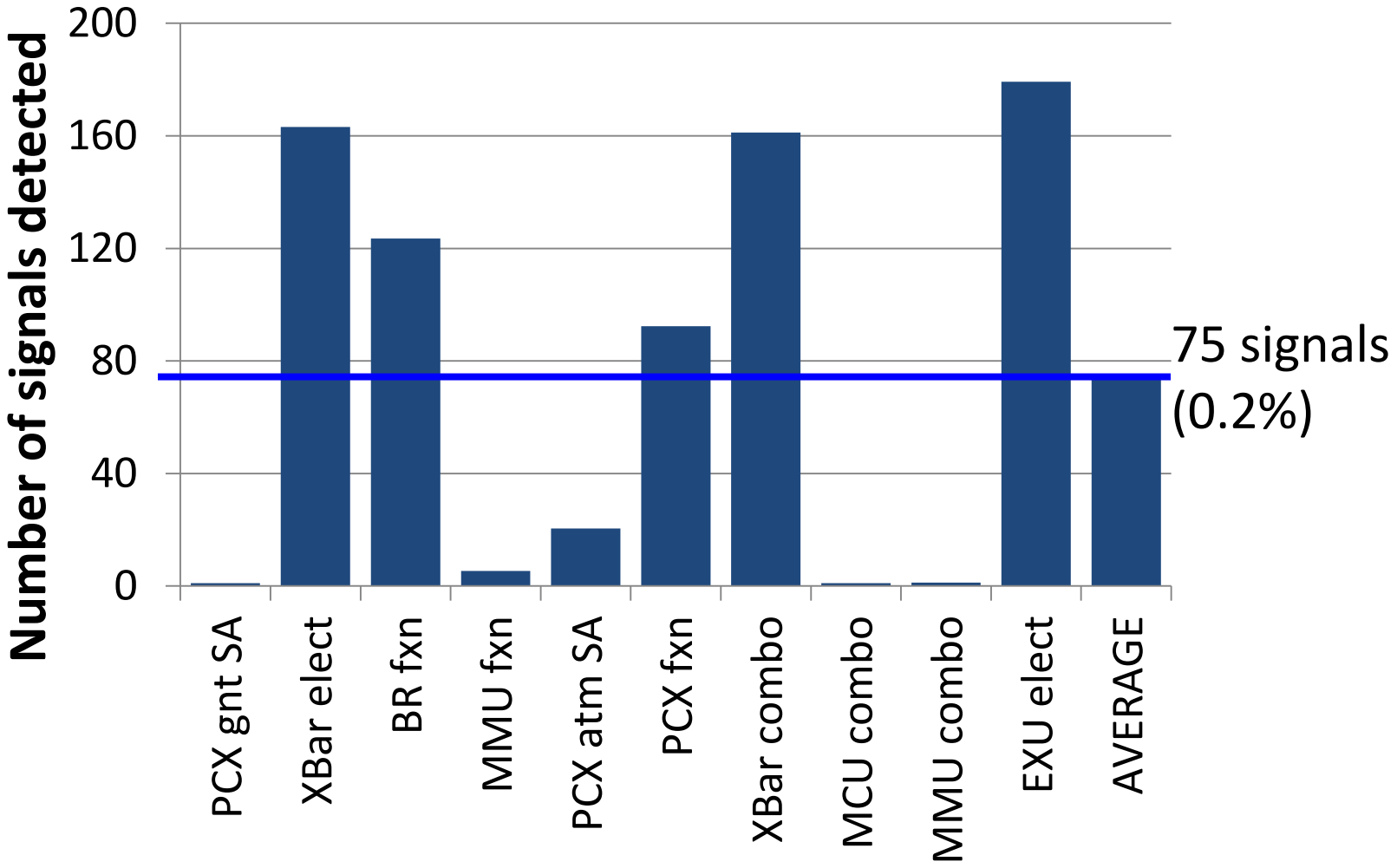
Testcases	PCX gnt SA	Xbar elect	BR fxn	MMU fxn	PCX atm SA	PCX fxn	Xbar combo	MCU combo	MMU combo	EXU elect
blimp_rand	√+	√	√	√	√+	√+	√+	f.n.	√+	f.n.
fp_addsub	n.b.	f.p.	√	√	√	√+	f.p.	n.b.	√+	f.p.
fp_muldiv	n.b.	f.p.	√	√	√	√+	f.p.	f.p.	√+	f.p.
isa2_basic	n.b.	f.n.	√	n.b.	√+	√+	√+	√+	n.b.	f.n.
isa3_asr_pr	n.b.	√	√	f.n.	√+	√	√+	√+	√	√
isa3_window	n.b.	√	√	n.b.	√+	√	f.n.	f.n.	n.b.	√
ldst_sync	n.b.	√+	√	√	√+	√+	√+	√+	√+	n.b.
mpgen_smc	n.b.	√+	√	√	√+	√+	√+	√+	√+	√+
n2_lsu_asi	n.b.	f.n.	√	f.n.	√+	√+	√+	√+	√+	n.b.
tlu_rand	n.b.	√+	√	√	√+	√+	√+	√+	√+	√+

n.b. no bug √ found √+ exact signal f.p. false pos. f.n. false neg.

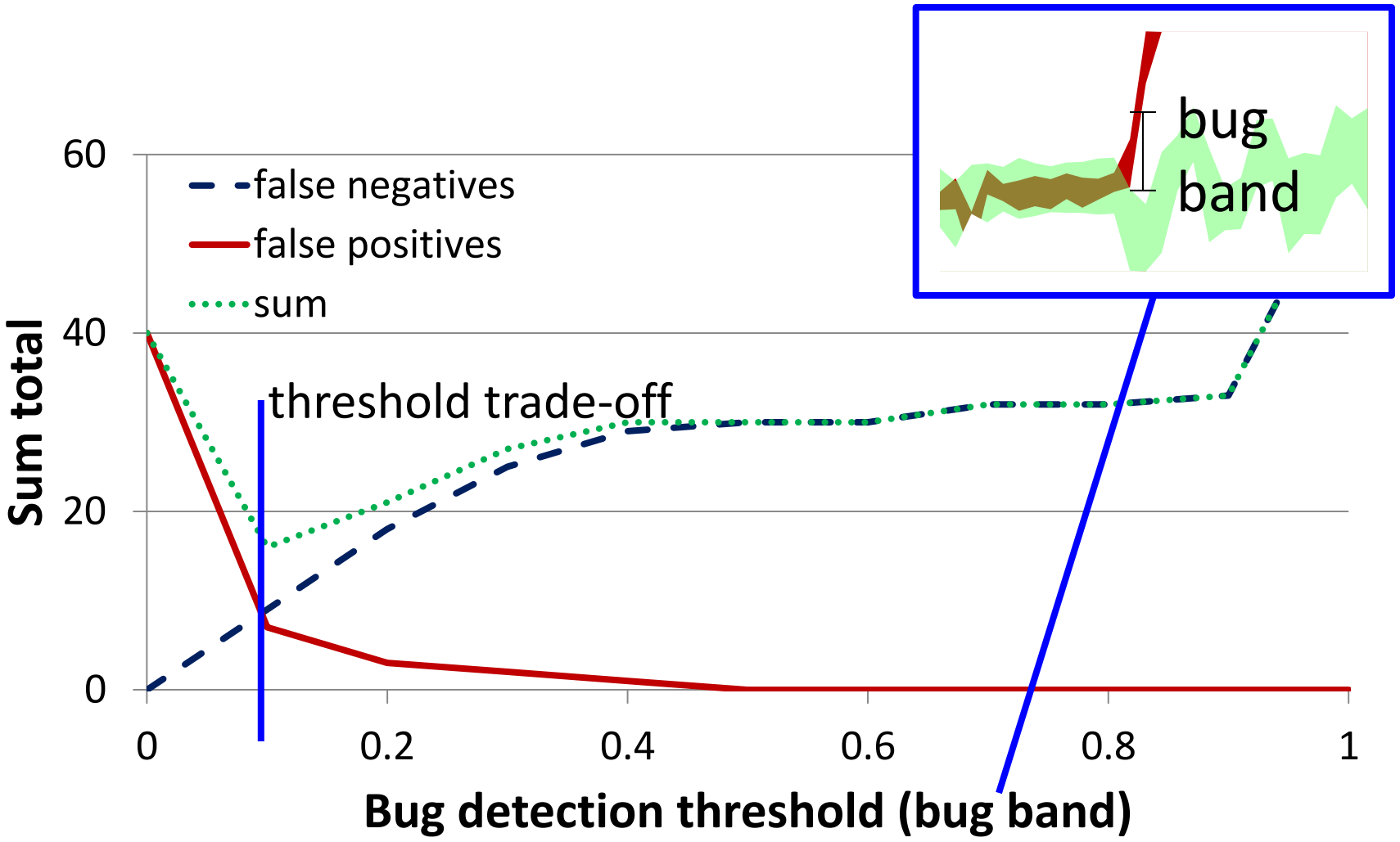
Time to detect bug



Number of signals detected

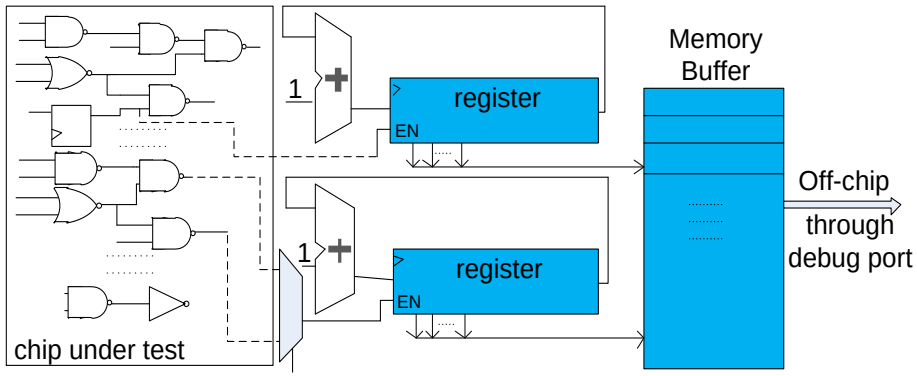


Threshold selection



Area overhead

- Option 1: reuse existing debug structures
- Option 2: add counters and memory buffer
  - Record a few signals at a time
  - 11KB for 100 signals x 100 windows @9bit precision
  - 1.35mm^2 with 65nm library
  - **0.4% of OpenSPARC**



Conclusions

- BPS automatically localizes **bug time and location**
- Leverages a **statistical approach** to tolerate noise
- Effective for a **variety of bugs**: functional, electrical and manufacturing
  - 1,273 cycles, 75 signals on average