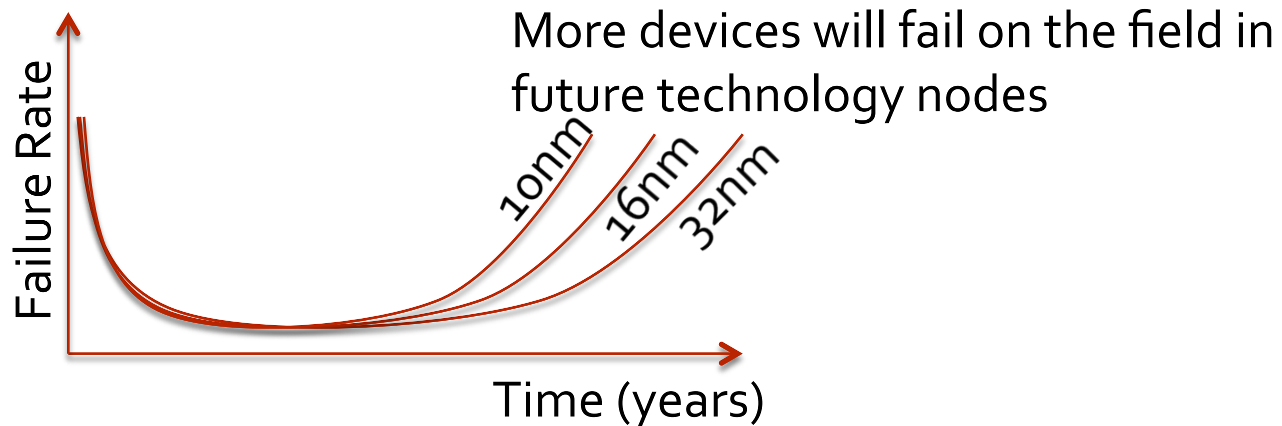
Virtually-Aged Sampling DMR
Unifying Circuit Failure Prediction and Detection

Raghuraman Balasubramanian
Karthikeyan Sankaralingam

Microprocessor Reliability ☹

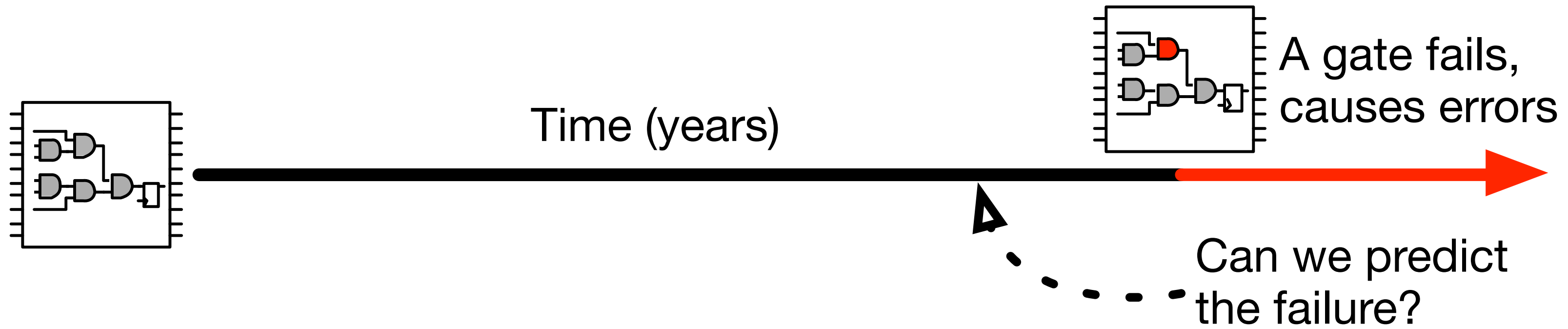


- A lot of research on how to…
  - Mitigate / Recover / Repair …

  - Detect : DMR, Diva, Argus, BIST, SWAT…
  - Predict : Canaries, Razor, WearMon…

- Coverage, detection latency, fault type…

Circuit Failure Prediction



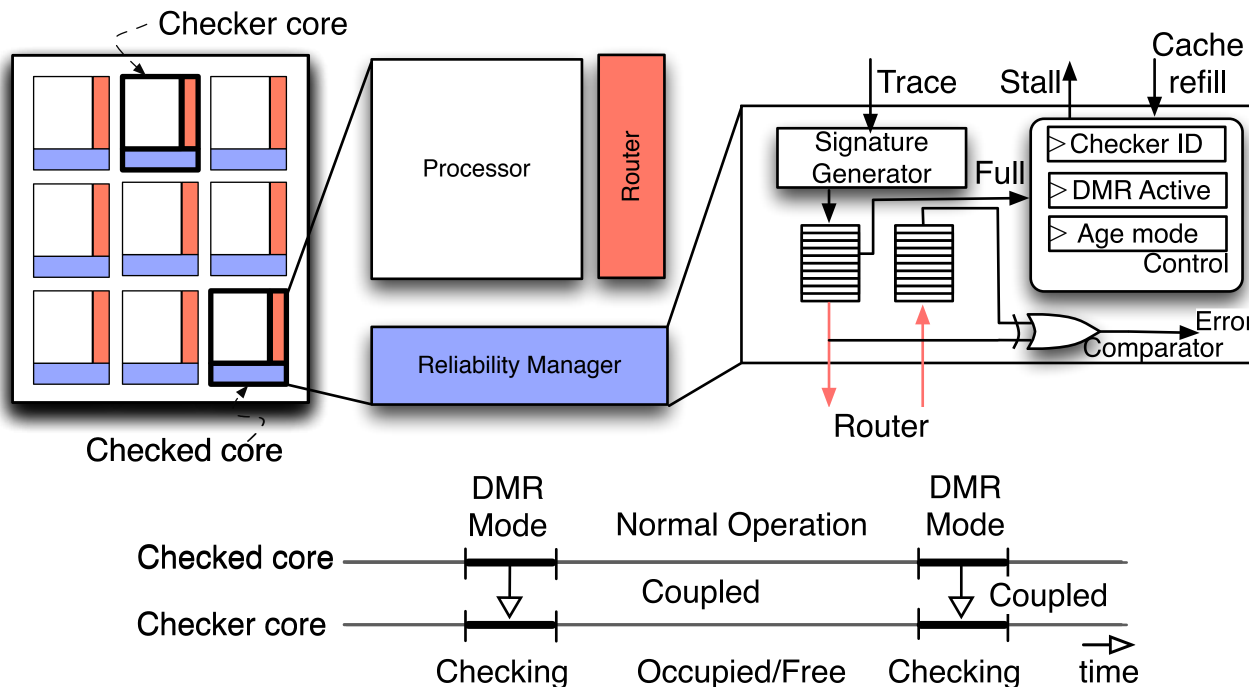
- Our goals
  - Low Design Complexity
  - Low Overheads
  - High Accuracy
  - Full Coverage

To get there…

Lets start from a good baseline
Sampling-DMR

Sampling+DMR

- Permanent fault detection
  - 100% coverage
  - < 2% Energy overheads



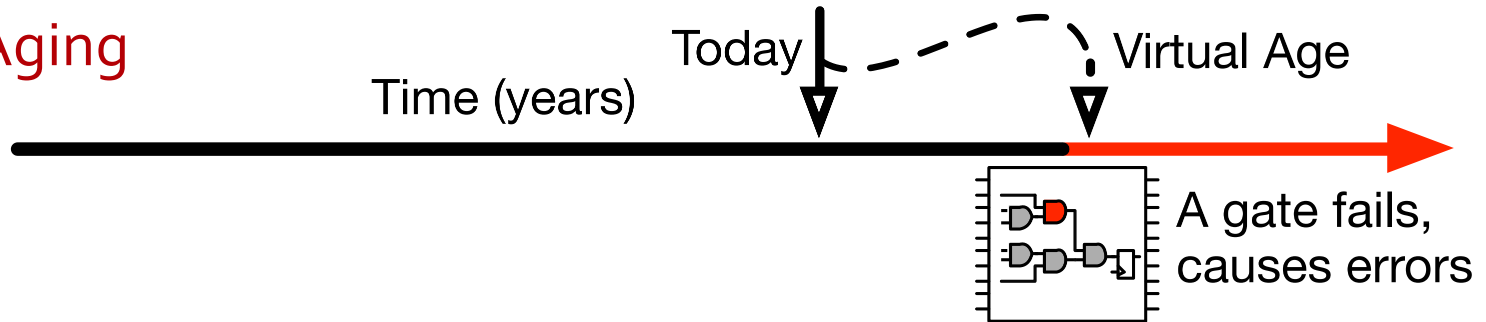
Nomura, Shuou, et al. "Sampling+dmr: practical and low-overhead permanent fault detection." *International Symposium on Computer Architecture (ISCA), 2011*

But There is a Problem

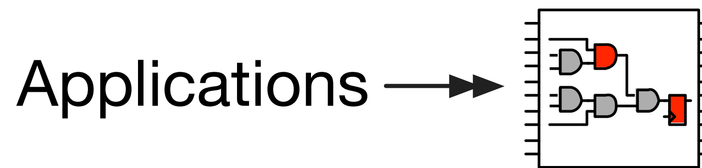
Virtual aging makes the gates behave as if they were 6 months older

Virtually Aged Sampling DMR

Virtual Aging

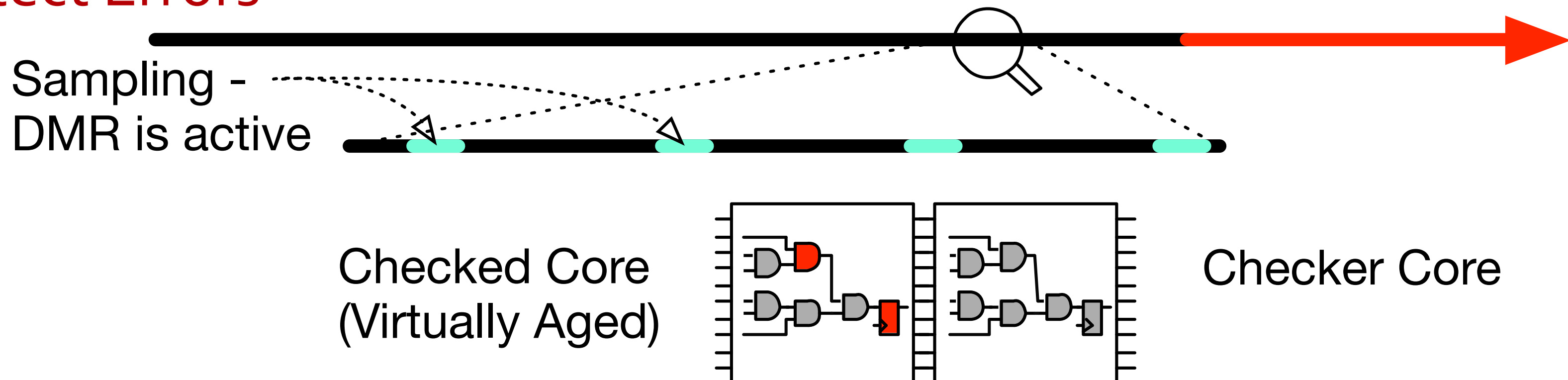


Fault Exposure



- In most gates the faults are automatically exposed
- A new mechanism to expose faults in other gates

Detect Errors



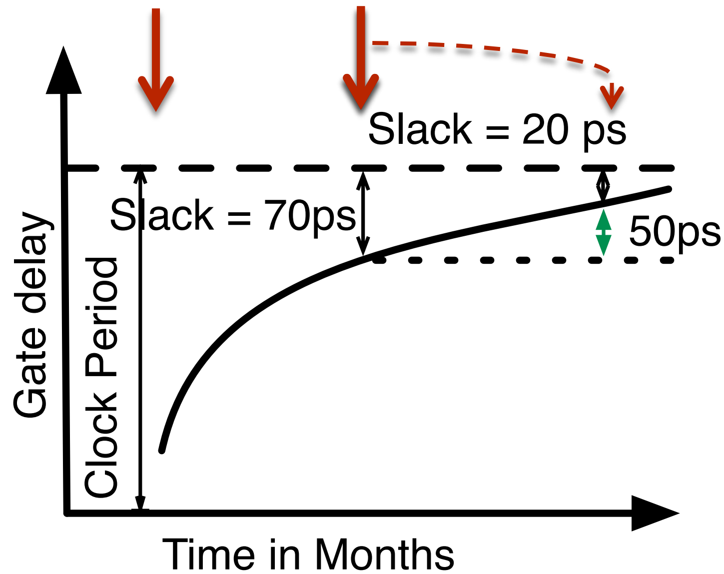
Executive Summary

- Virtually Aged Sampling-DMR
  - Microprocessor Failure Prediction
  - Full logic coverage
  - With < 0.7% energy overhead
  - Negligible performance overhead

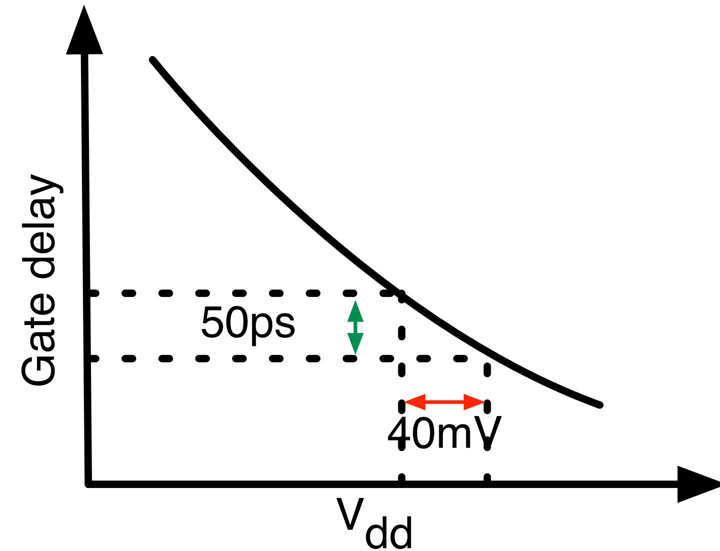
Outline

- Motivation and Overview
- Virtual aging?
- Are all gates covered?
- Evaluation Methodology
- Results
- Related work
- Questions

Virtual Aging



As a chip wears out, the gates become slower



As we decrease V_{dd}, the gates become slower

Virtual aging => Reducing V_{dd} == 6-month Delay Degradation

Outline

- Motivation and Overview
- Virtual aging
- **Are all gates covered?**
- Evaluation Methodology
- Results
- Related work
- Questions

Are all gates covered?

- Most gates (near-critical paths) ✔
  - Initial worst-case propagation delay ~ clock period
  - Wearout ➔ propagation delay ↑ > clock period
  - **Delay fault is naturally exposed**
- Some gates (non-critical paths) ✗
  - Initial worst-case propagation delay << clock period
  - Wearout ➔ propagation delay ↑ < clock period ⇒ Fault is not manifested
  - **Delay degradation is benign**
  - Eventually catastrophic breakdown ☹

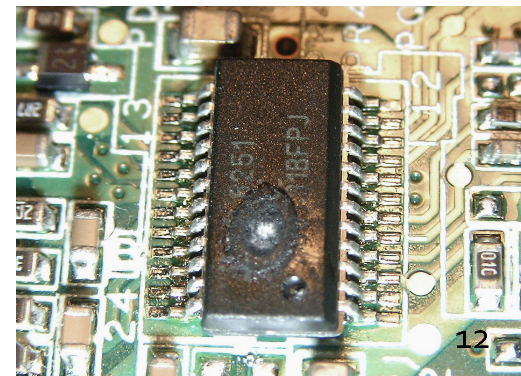
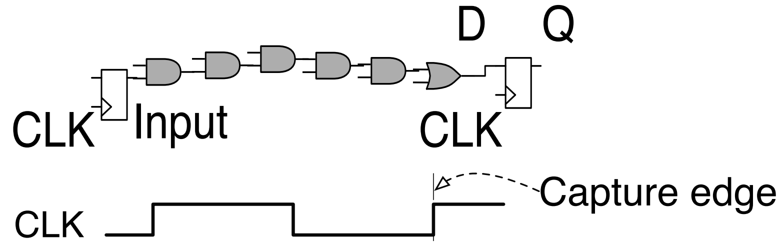
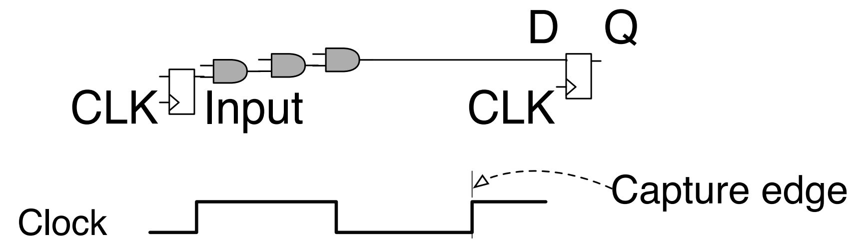
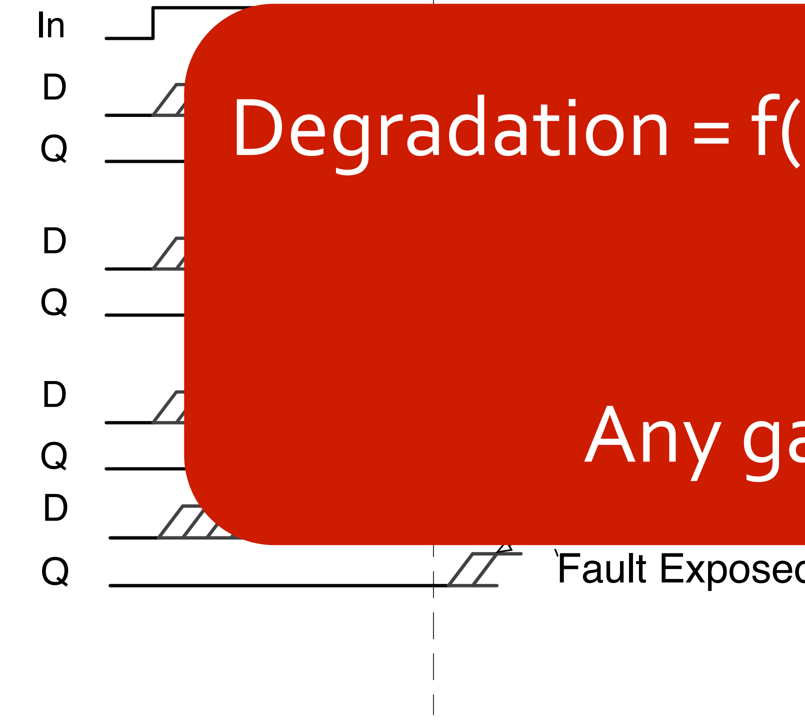


Photo credit : Wikimedia Commons

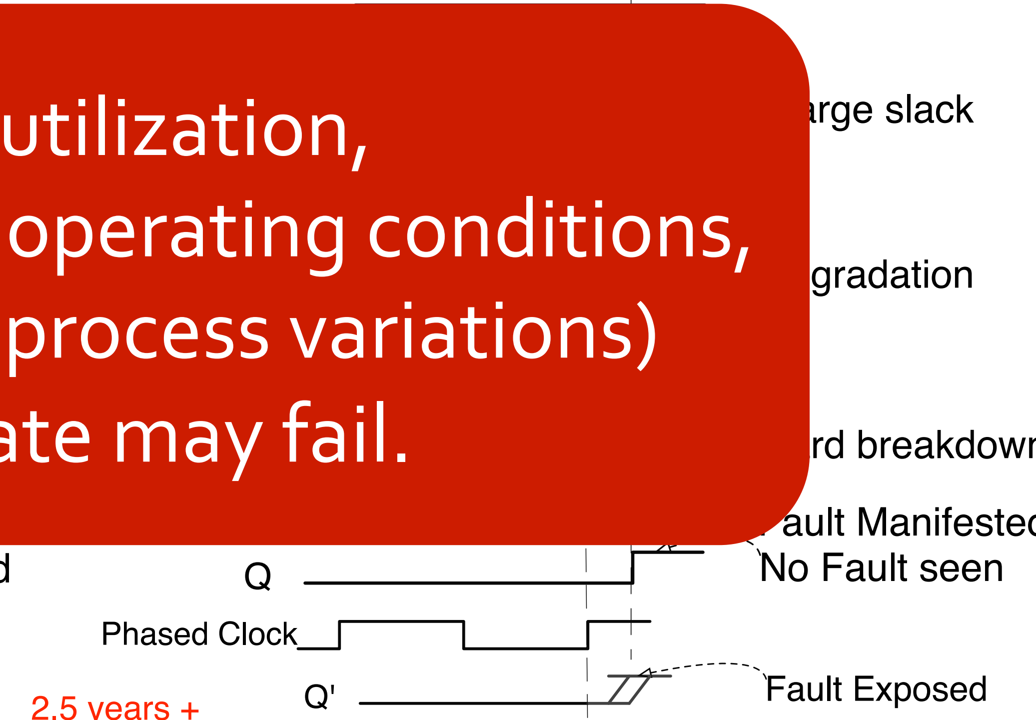
Soft and Hard breakdown







(a) Near-Critical Paths



(b) Non-Critical Paths

Degradation = f(utilization,
operating conditions,
process variations)

Any gate may fail.

Fault Capture Logic for Non-Critical Paths

Near-critical paths

fast gate

CLK

Non-critical path

Fault Capture Logic for Non-Critical Paths

Comprehensive Logic coverage

Virtually Aged SDMR

Low Overheads ● Low Design-Complexity ● High Accuracy ●

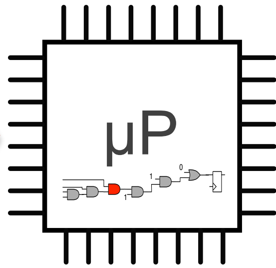
Generality: { Soft Breakdown ● Hard Breakdown ● }

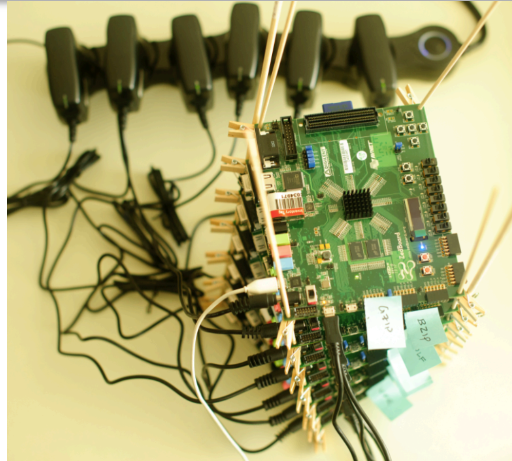
Outline

- Motivation and Overview
- Virtual aging??
- Are all gates covered??
- **Evaluation Methodology**
- Results
- Related work
- Questions

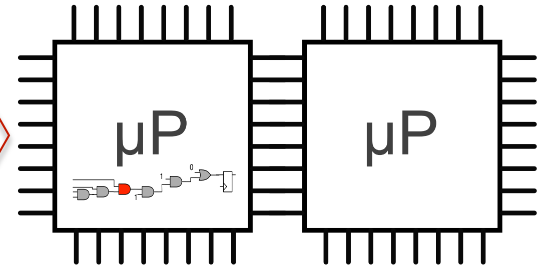
Evaluation Methodology

Applications



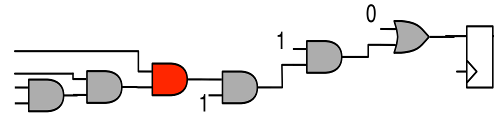


Applications



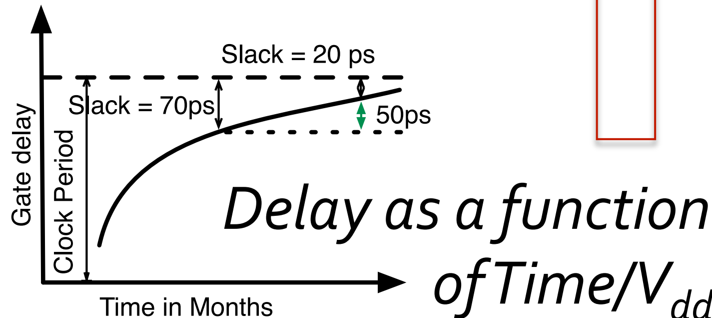
DMR
Error??

Input Sequences



Fault Vector

Delay Aware Simulation



Synopsys
HSPICE +
MOSRA

Delay as a function of Time/V_{dd}

- Full SPEC benchmarks
- OpenRISC Processor
- ~400,000 Fault Injection Experiments

Outline

- Motivation and Overview
- Virtual aging?
- Are all gates covered?
- Evaluation Methodology
- **Results**
- Related work
- Questions

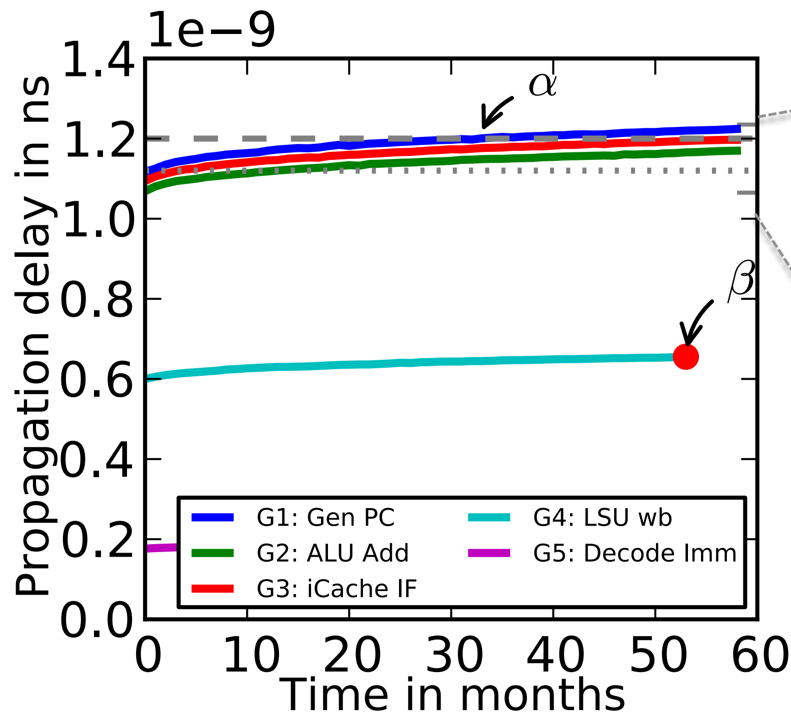
Results

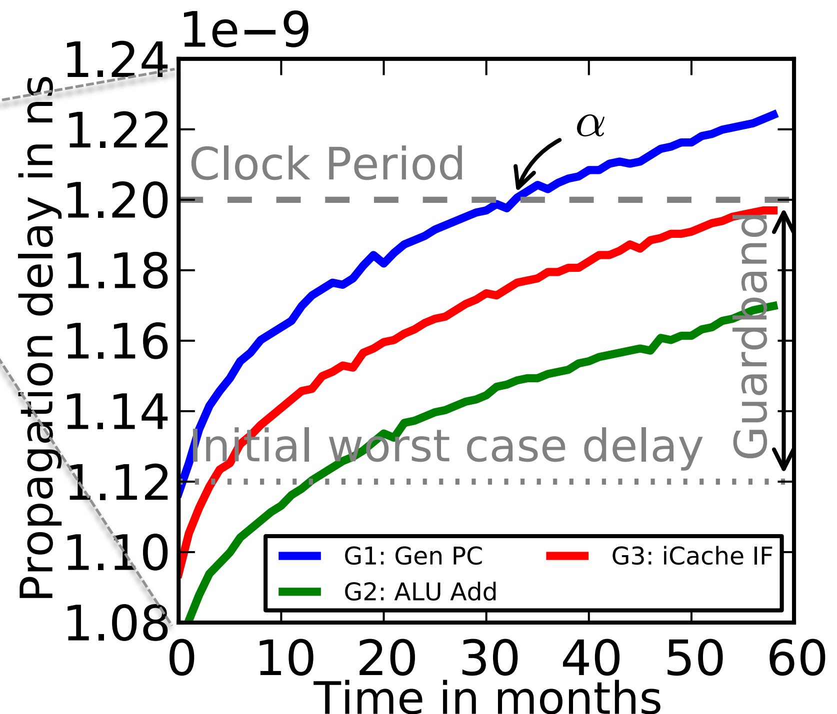
1. Is delay degradation measurably observable?
2. Can voltage reduction mimic virtual aging?
3. Do the manifested faults get exposed to the µ arch and cause timing faults?
4. Do the faults exposed to the microarchitecture translate to architectural errors, then detected?
5. What are the overheads?

Paper includes results on running 10 SPEC benchmarks to completion spanning almost 400,000 experimental runs

1. Is delay degradation measurably observable?

- 5 gates represent fault sites
- Model paths through these gates in HSPICE
- MOSRA wearout models





2. Can voltage reduction mimic virtual aging?

- HSPICE @ V_{dd} = 1.2 V, V_{dd} = 1.15V

1e−9

1.24
1.22
1.20
1.18
1.16
1.14
1.12
1.10
1.08

Propagation delay in ns

9 months

$\alpha_{V_{dd}=1.2V}$

$\alpha_{V_{dd}=1.15V}$

Guardband

Initial worst case delay

V_{DD} = 1.2V V_{DD} = 1.15V

0 10 20 30 40 50 60

Time in months

5. What are the overheads?

- Synthesized with 32nm Synopsys process
- Implemented additional logic for fast paths

	OpenRISC		OpenSPARC	
	Logic	Processor	Logic	Processor
Gates on Fast Path	39%		30%	
Area Overhead	28.9%	8.9%	22.2%	6.8%
Peak Power Increase	3.2%	2.54%	2.21%	0.99%
Energy Increase	0.9%	0.7%	1.02%	1.07%

Results - Summary

- Experimental Result

  Predict failures 9 months in advance using a V_{dd} reduction of 50mV

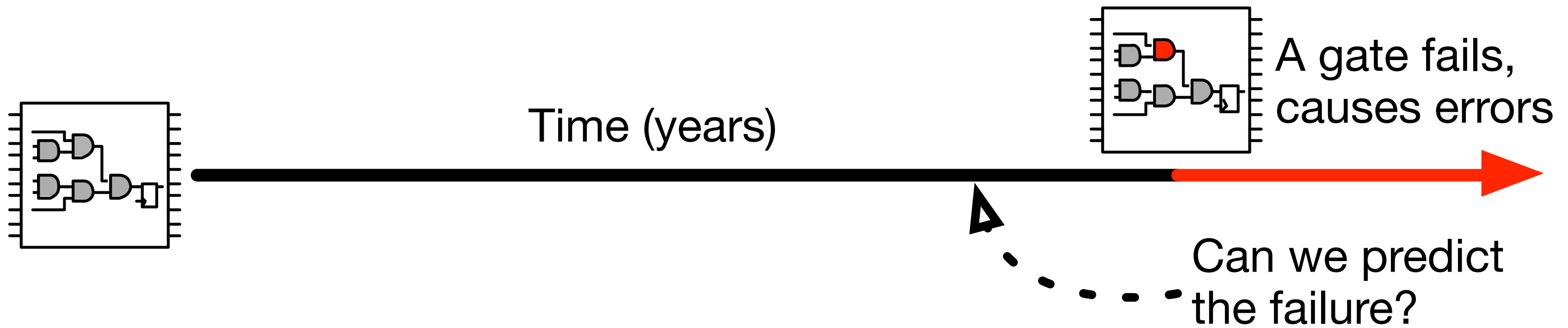
- Empirical result + Mathematical modeling

  Can predict failure within 0.4 days in all but 1 of 1 billion chips

Outline

- Motivation and Overview
- Virtual aging?
- Are all gates covered?
- Evaluation Methodology
- Results
- **Related work**
- Questions

Circuit Failure Prediction



- Predict the onset of failures
  - Low Design Complexity
  - Low Overheads
  - High Accuracy
  - Full Coverage

Related Work

Technique	Complexity	Overheads	Accuracy	Coverage
Canary circuits	✓	✓	✗	✗

On-chip test circuits

Related Work

Technique	Complexity	Overheads	Accuracy	Coverage
Canary circuits	✓	✓	✗	✗
Age Detection (Shadow) Latches	✗	✗	✓	✗

Detect aging in select near-critical paths

Related Work

Technique	Complexity	Overheads	Accuracy	Coverage
Canary circuits	✓	✓	✗	✗
Age Detection (Shadow) Latches	✗	✗	✓	✗
BIST/DFT Aging Analysis	✗	✗	✓	✗

Periodic testing (offline) using on-chip test vectors

Related Work

Technique	Complexity	Overheads	Accuracy	Coverage
Canary circuits	✓	✓	✗	✗
Age Detection (Shadow) Latches	✗	✗	✓	✗
BIST/DFT Aging Analysis	✗	✗	✓	✗
Continuous Delay Tracking	✗	✗	✓	✗

Measure + Analyze (online)

Related Work

Technique	Complexity	Overheads	Accuracy	Coverage
Canary circuits	✓	✓	✗	✗
Age Detection (Shadow) Latches	✗	✗	✓	✗
BIST/DFT Aging Analysis	✗	✗	✓	✗
Continuous Delay Tracking	✗	✗	✓	✗
Virtually Aged Sampling DMR	✓	✓	✓	✓

Reduce V_{dd} + Expose Faults

Contributions

- Virtually Aged Sampling-DMR
  - Microprocessor Failure Prediction
  - Full logic coverage
  - With < 0.7% energy overhead
  - Negligible performance overhead
- A new state-of-the-art in evaluation
  - Accurate wearout models at the gate level
  - And impact on full system (running full benchmarks)

Thank You

How Devices Degrade

- NBTI, HCI, TDDB
- Over time, Threshold Voltage Increases
  $\Longrightarrow$ Propagation Delay Increases

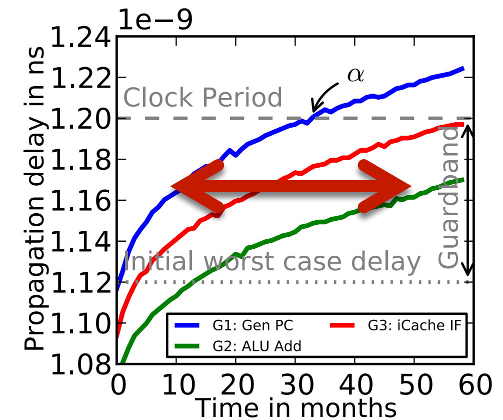
$$t_d = \frac{2LC}{W \mu_{eff} C_{ox} (V_{dd} - V_{th})^2}$$

Target failure mechanisms for which delay degradation is a symptom

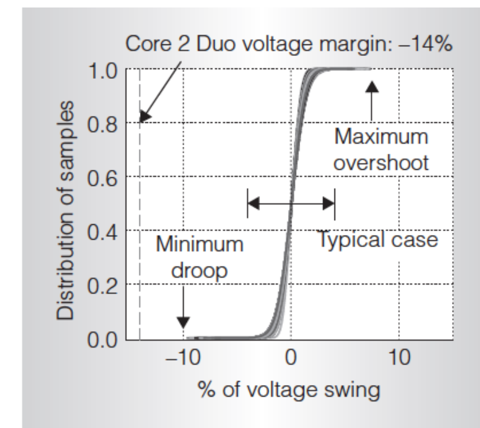
- NOT covered: Electromigration, thermal runaway

Variations

- Process variations (*Static)*
  - Some processors are more susceptible



- Voltage variations *(Dynamic)*
  - Variations ~1 order of magnitude smaller compared to degradation
  - Similar conditions in actual failure & virtual aging



Reddi, Vijay Janapa, et al. "Voltage noise in production processors." *Micro, IEEE* 31.1 (2011).

When does this not work?

- Only when the *conditions* change drastically between prediction and actual failure
  - Change in program behavior
  - Operating conditions (Temperature, Voltage etc.,)
  - Program hides fault exposure (but stresses it)

- As long as the fault is manifested 0.4 days before the actual failure – Aged-SDMR works.

Evaluation setup

Manifesting Faults

OpenRISC RTL

32nm lib

Synopsys Design Compiler

Netlist

Gate under test

Worst case path

32nm lib

HSPICE + MOSRA

Time
Voltage
Switching Activity

Delay

Time

@ different gates
@ supply voltage reduction
@ switching activity variation

Q1 : Is delay degradation in CMOS logic measurably observable? Is this deterministic?
Q2 : Can reducing supply voltage virtually manifest wearout faults? ***A1, A2 : Figure 8***

Exposing Faults

Xilinx Zynq FPGA

Sensitivity vector extraction

OpenRISC processor

SPEC2000

Sensitivity vector for fan in for each gate during a 100000 cycle sampling window

VCS

Delay Aware Simulation

Gate subcircuit

9 months from degradation,SS-DMR mode

Timing Fault Rate

Fault Vectors

Q3 : Do these faults get exposed to the micro-architecture and cause timing faults? ***A3 : Table 4***

Estimating Overheads

Netlist

Synopsys Design Compiler- STA

Fast Gates

Script to insert capture logic

Modified Netlist

Area, Power, Energy overheads

Q5 : What are the overheads? ***A4 : Table 6***

Detecting Architectural Errors

SPEC2000

Xilinx Zynq FPGA

Aged SDMR Emulation

OpenRISC Processor

Architecture state

Ager

Test controller

Checker

OpenRISC Processor

Architecture state

Architectural Error Rate

Q4 : Do these timing faults translate to architectural errors, then detected? ***A4 : Table 5***

3. Do the manifested faults get exposed to the μ-arch and cause timing faults?

- Delay Aware Simulation
- Input sequences from OpenRISC FPGA
  - 10 benchmarks (6 SPEC INT, 4 SPEC FP)
  - 5 million cycle traces x 3 phases of the program

We saw timing faults appear during the sampling windows

- Cycle accurate fault vectors

4. Do the faults exposed in the microarchitecture translate to architectural errors, then detected?

- Fault vector from delay aware simulation
- Injected on OpenRISC on FPGA + DMR emulation

Appln	G1	G2	G3	G4	G5
ammp	1.60%	3.10%	5.10%	1.40%	1.40%
art	0.02%	2.70%	0.01%	2.60%	0.01%
bzip	2.30%	1.20%	0.90%	0.20%	0.07%
gzip	1.50%	0.03%	0.40%	0.04%	0.01%
mcf	3.40%	3.10%	0.90%	0.70%	0.02%
mesa	2.20%	1.00%	1.20%	0.09%	0.80%
parser	4.30%	1.30%	1.90%	0.50%	1.50%
quake	1.90%	0.90%	0.80%	0.20%	1.30%
twolf	3.30%	1.10%	0.02%	4.30%	1.90%
vpr	2.60%	0.80%	2.10%	0.70%	1.60%

Architecture error rate using 100000 cycle sampling windows

Canary based

Processor circuit

CLK

Representative test circuit

Delay measurement

Failure predicted?

Low Overheads ● Low Design-Complexity ● High Accuracy ○

Generality: { Soft Breakdown ● Hard Breakdown ● }

Age Detection Latches

Processor circuit

Age detecting latch

CLK

Gates covered

Gates missed

Circuit level failures

Low Overheads ◐ Low Design-Complexity ○ High Accuracy ●

Generality: { Soft Breakdown ◐ Hard Breakdown ○ }

BIST/DFT Based (Offline)

Low Overheads ◐ Low Design-Complexity ○ High Accuracy ●

Generality: { Soft Breakdown ●† Hard Breakdown ○ }

Continuous Degradation Tracking

Processor circuit

CLK

Degradation measurement and Analysis

Failure predicted?

Low Overheads ● Low Design-Complexity ○ High Accuracy ●

Generality: { Soft Breakdown ●† Hard Breakdown ○ }

Evaluation Methodology : Key Challenges

- Aged-SDMR is a Cross-layered Approach
  - Wearout is a gate-level phenomenon
  - Sampling-DMR works at the architecture level
- Application dependency
  - Technique relies on the application to expose faults

Run full applications on a full system simulator & model wearout at the device level