Addressing Emerging Fault Modes with Testing and Reliability

Vilas Sridharan
Senior Fellow, RAS Architecture
Advanced Micro Devices, Inc.

With credit to: Sankar Gurumurthy, Sudhanva Gurumurthi, Jeff Rearick, Steve Hesley

AMD
together we advance_

Our Mission

Public Trusts Compute

How

Security **Privacy** **Integrity** **Reliability**

Source: “Trustworthy Computing,” *Microsoft White Paper 2002*

The Next Five Years

Computing Market Transformation

Data Center and Cloud

Insatiable Performance Demands

Workload Optimized Compute/Networking

Edge Compute: Distributed DC

Security from Core to Edge

Efficiency and Sustainability Focus

Explosion of AI

AI Workloads Proliferating

Dominating the Data Center

Expanding to Edge and Endpoint

Increasingly Large Models

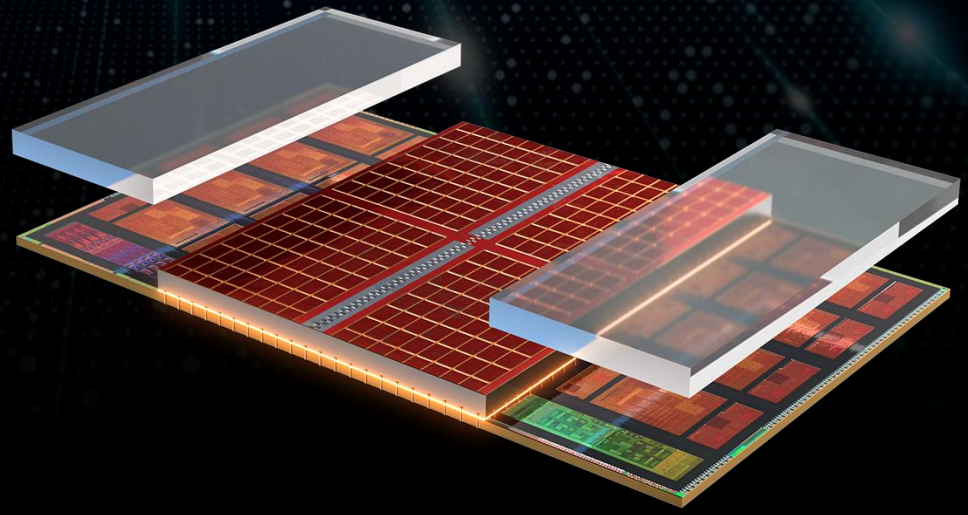
PCs & Gaming

Hybrid Work Focused on Improving Collaboration, Battery Life, Security

Billions of Gamers Gaming Anywhere and at Anytime

AI-powered Productivity, Creativity and Gaming

Technology Scaling



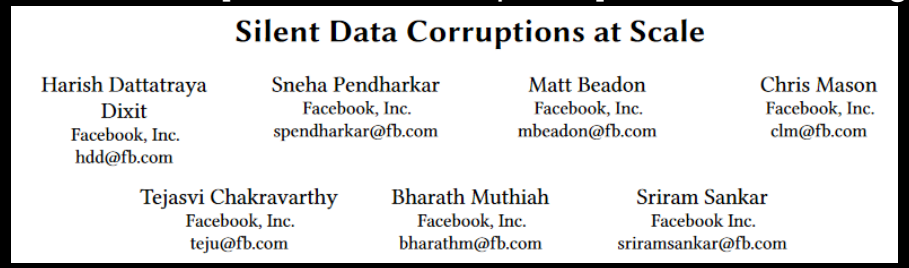
Process Technology is not scaling at Moore's Law

New approaches are required

Chiplets & Die Stacking are becoming ubiquitous

Is the industry achieving the mission?

"CPU SDCs [silent data corruptions] are orders of magnitude higher than soft-error based FIT simulations" [1]



Silent Data Corruptions at Scale

Harish Dattatraya Dixit
Facebook, Inc.
hdd@fb.com

Sneha Pendharkar
Facebook, Inc.
spendharkar@fb.com

Matt Beadon
Facebook, Inc.
mbeadon@fb.com

Chris Mason
Facebook, Inc.
clm@fb.com

Tejasvi Chakravarthy
Facebook, Inc.
teju@fb.com

Bharath Muthiah
Facebook, Inc.
bharathm@fb.com

Sriram Sankar
Facebook Inc.
sriramsankar@fb.com

"On the order of a few mercurial cores per several thousand machines" [2]

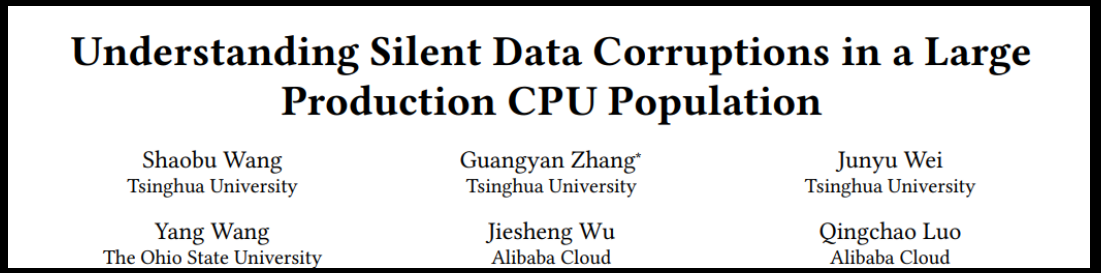
Cores that don't count

Peter H. Hochschild
Paul Turner
Jeffrey C. Mogul
Google
Sunnyvale, CA, US

Rama Govindaraju Parthasarathy Ranganathan
Google
Sunnyvale, CA, US

David E. Culler
Amin Vahdat
Google
Sunnyvale, CA, US

"CPU SDCs occur at a low but non-negligible frequency" [3]



Understanding Silent Data Corruptions in a Large Production CPU Population

Shaobu Wang
Tsinghua University

Guangyan Zhang*
Tsinghua University

Junyu Wei
Tsinghua University

Yang Wang
The Ohio State University

Jiesheng Wu
Alibaba Cloud

Qingchao Luo
Alibaba Cloud

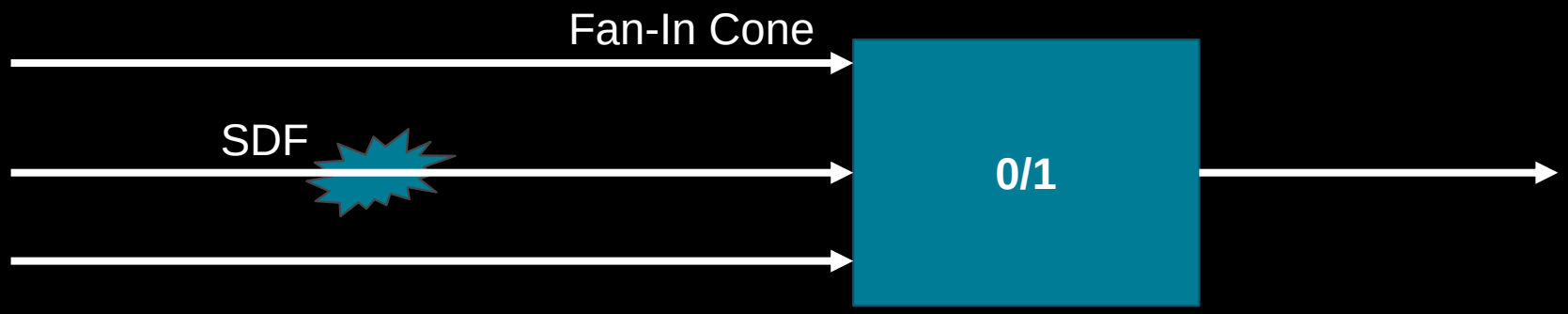
[1] Meta: "Silent Data Corruptions at Scale"
[2] Google: "Cores that don't Count"
[3] Alibaba: "Understanding Silent Data Corruptions in a Large Production CPU Population"

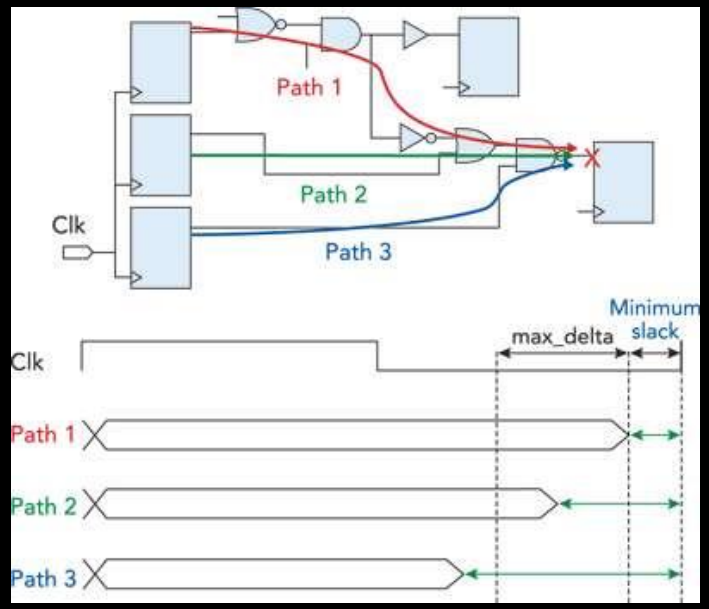
What the industry has learned

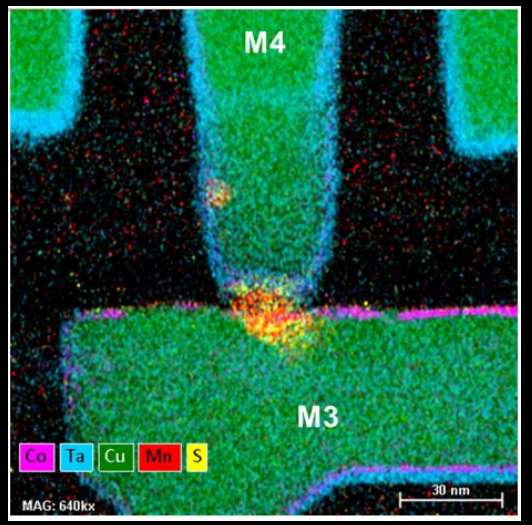
Root causes

Small delay faults (SDFs) due to marginal defects [4] [5] [6]

[4] VTS 2023: “Silent data errors: Sources, Detection, and Modeling”
[5] SIGARCH CAT 2023: “Emerging Fault Modes: Challenges and Research Opportunities”
[6] IRPS 2024: “Defect Mechanisms Responsible for Silent Data Errors”

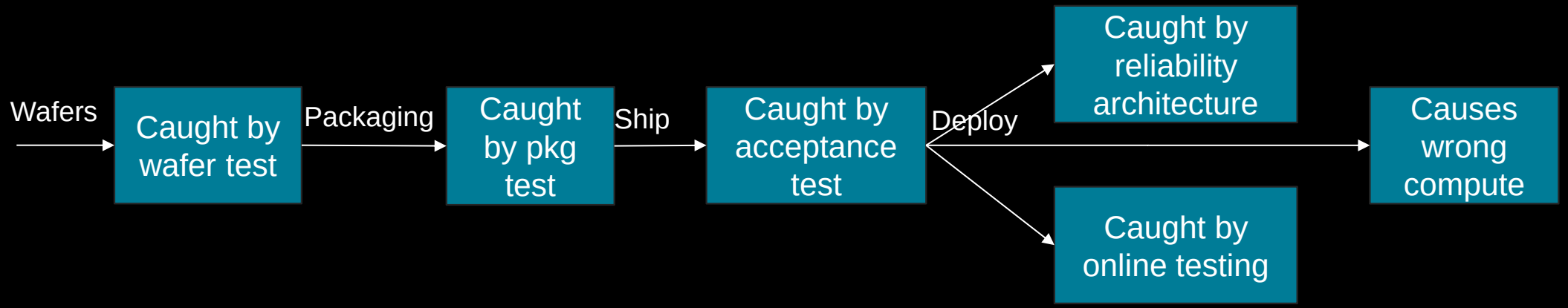






How SDFs Affect Product Lifecycle

Increasing cost of detection →

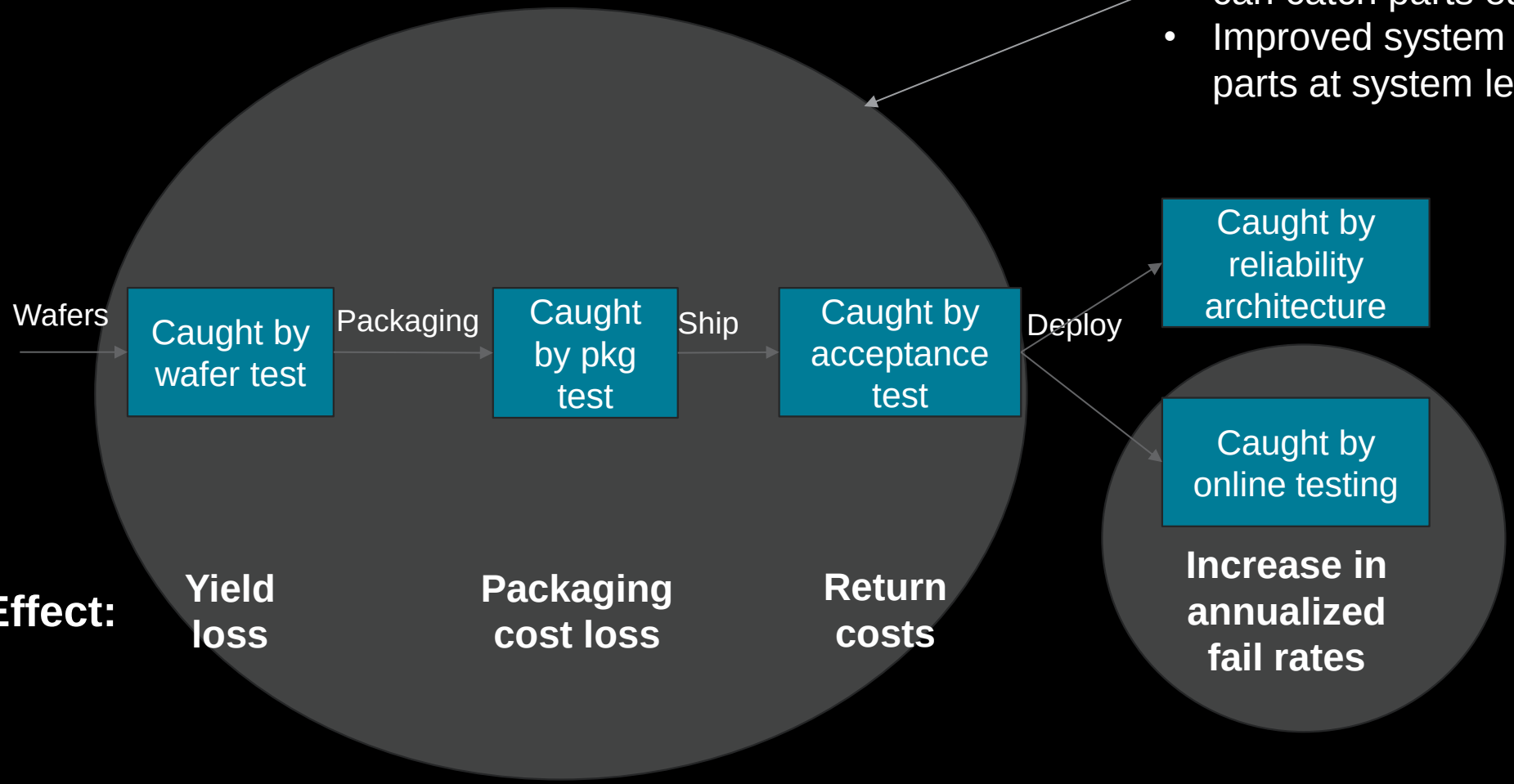


Effect:	Yield loss	Packaging cost loss	Return costs	Increase in annualized fail rates	SDC

← Imperative to move detection

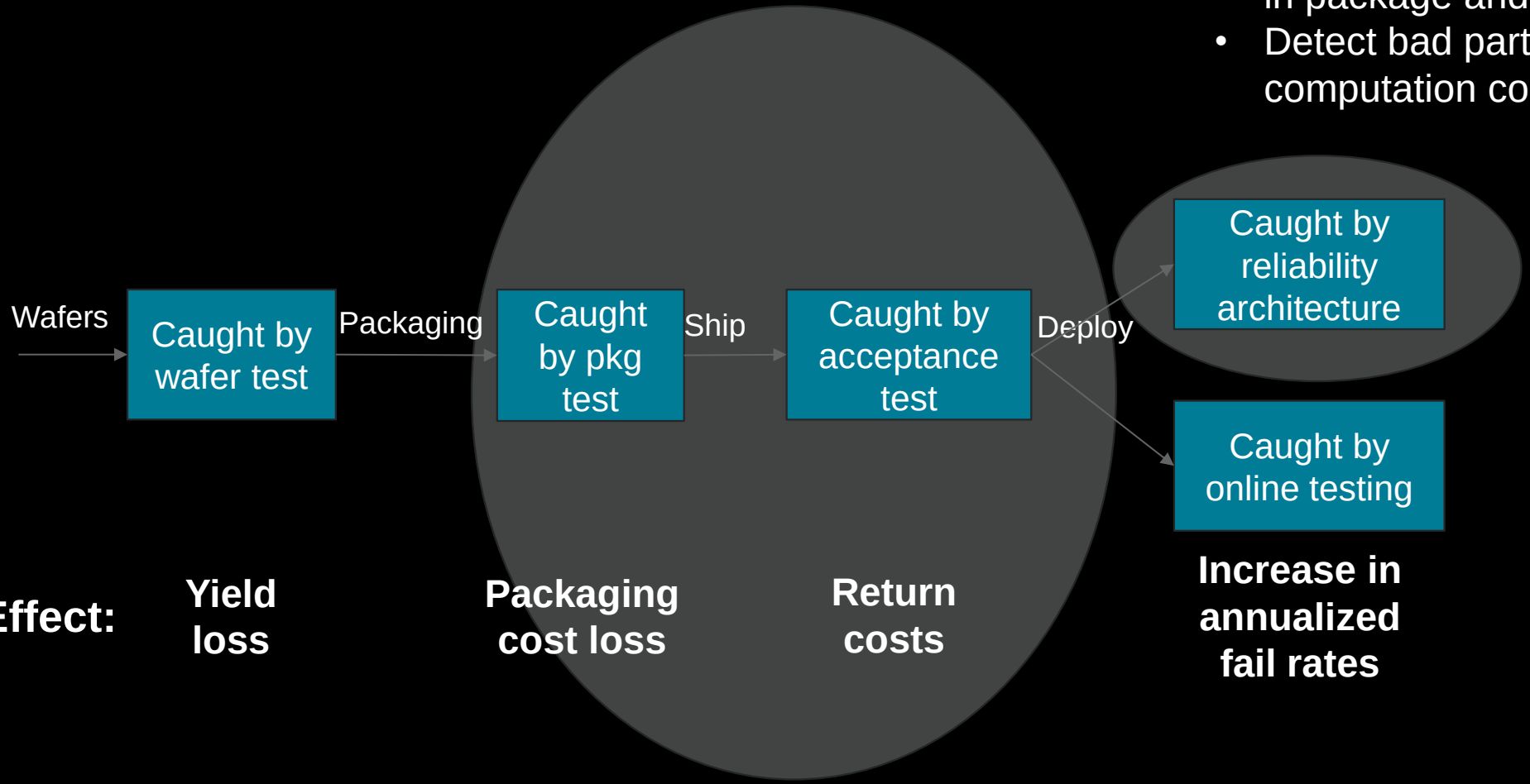
Testing improvements

- Improved tests for wafer and packaging can catch parts early in manufacturing
- Improved system tests that help catch parts at system level testing and in field



Reliability architecture improvements

- Improve observability of system tests used in package and acceptance tests
- Detect bad parts in field before real computation corruption



Effect:

Yield loss

Packaging cost loss

Return costs

Increase in annualized fail rates

Areas for Innovation

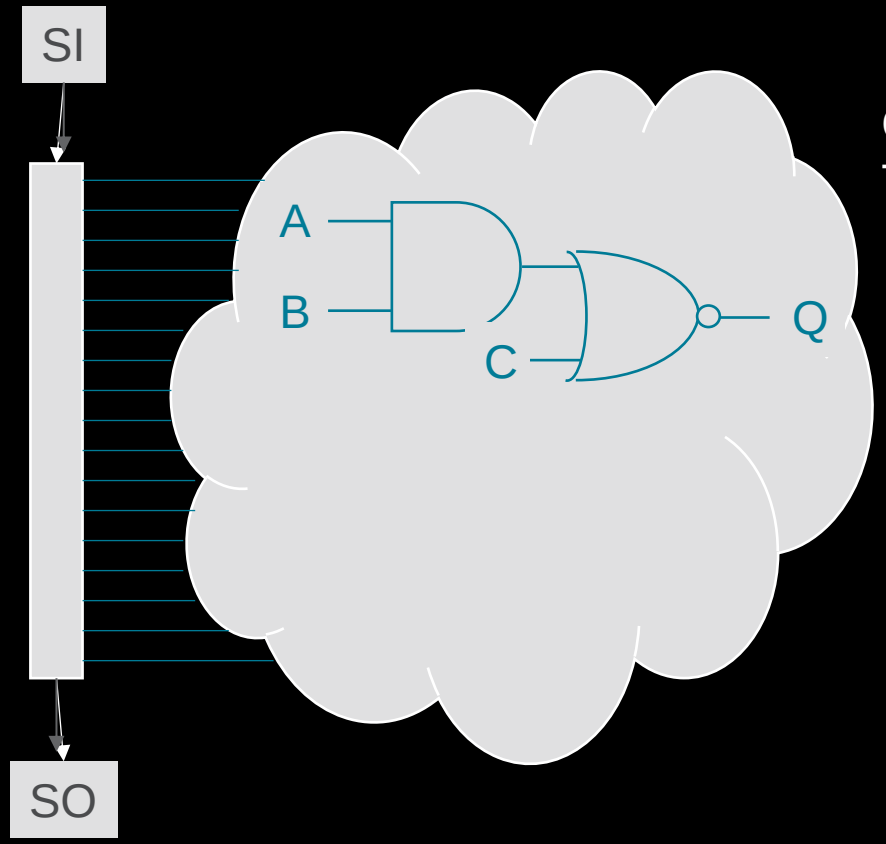
Testing

Burn-in techniques and coverage to accelerate latent defects

Structural tests that can better mimic mission mode conditions

Improved functional tests (manufacturing and online)

Burn-in

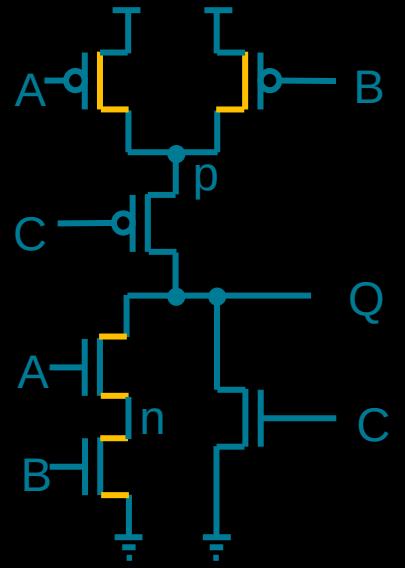


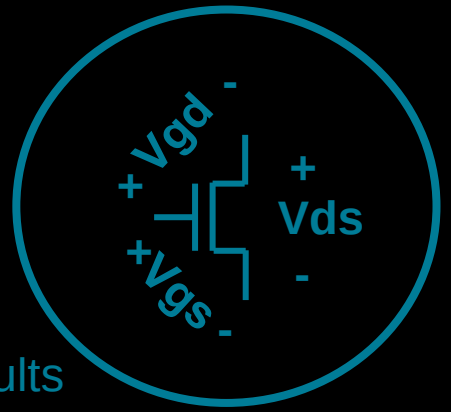
Original method:
Toggle fault model

A	B	C	Q
0	0	0	1
1	1	1	0

New method:
UDFM based on E-fields

A	B	C	Q
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	0





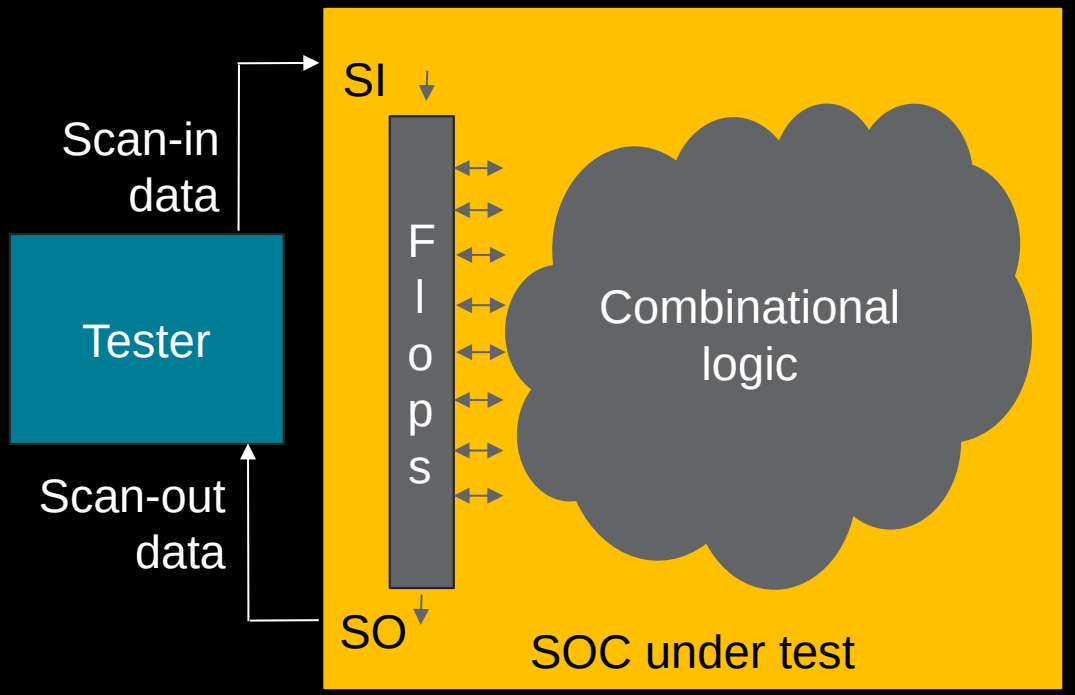
Electric field across
dielectrics and channels
+
Node-to-node bridging faults

Traditionally, scan toggle used in burn-in

Toggle coverage at the nodes used as the metric

Does that satisfy the requirement of getting electric field across dielectrics and channels?

Structural Testing



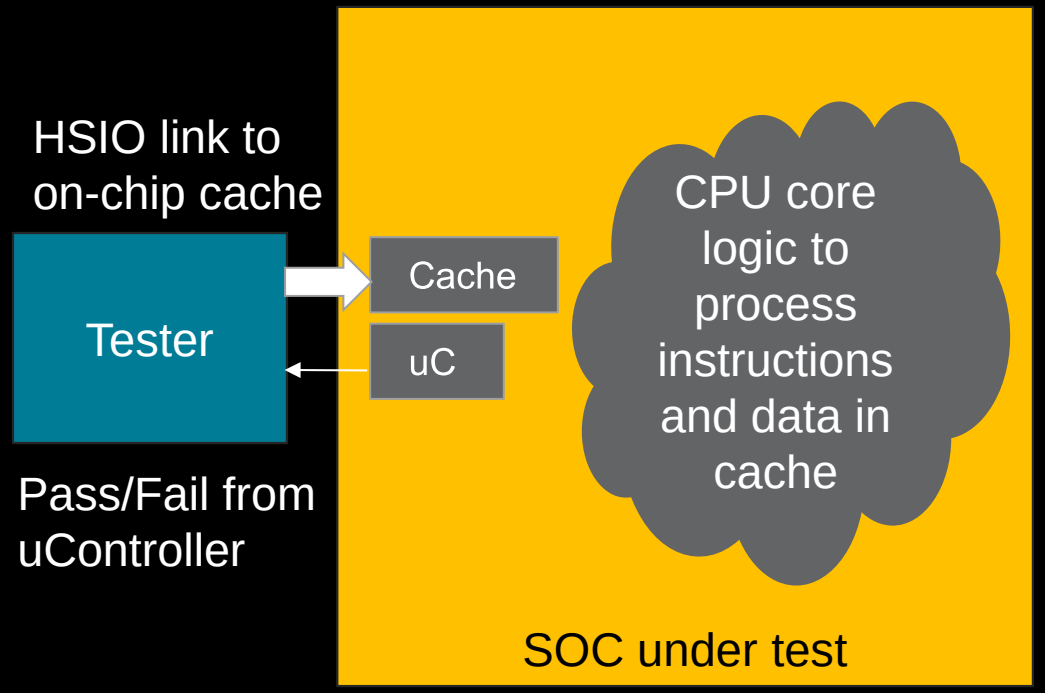
A simplistic view of ***Scan based test*** application

Combinatorial explosion of # of faults for path delay fault models

Electrical environment during a scan test does not replicate mission mode

Functional Testing

Manufacturing Test



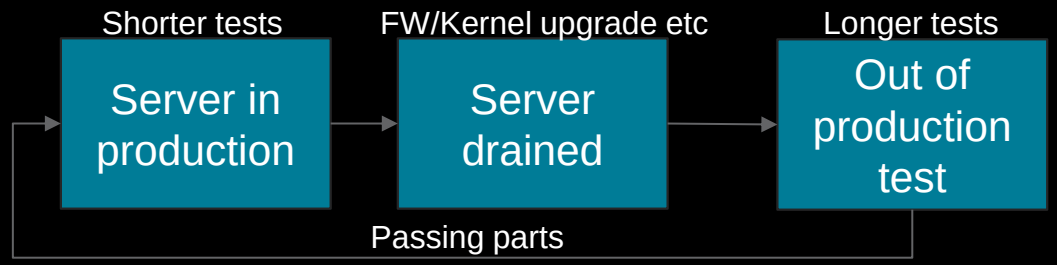
A simplistic view of ***functional self-test*** application

Limited generation of functional tests targeting fault models

Increased design complexity means system level tests don’t fully represent mission mode

Coverage evaluation and other toolsets for functional tests are lagging

Online Test



A simplistic view of ***online functional test*** lifecycle

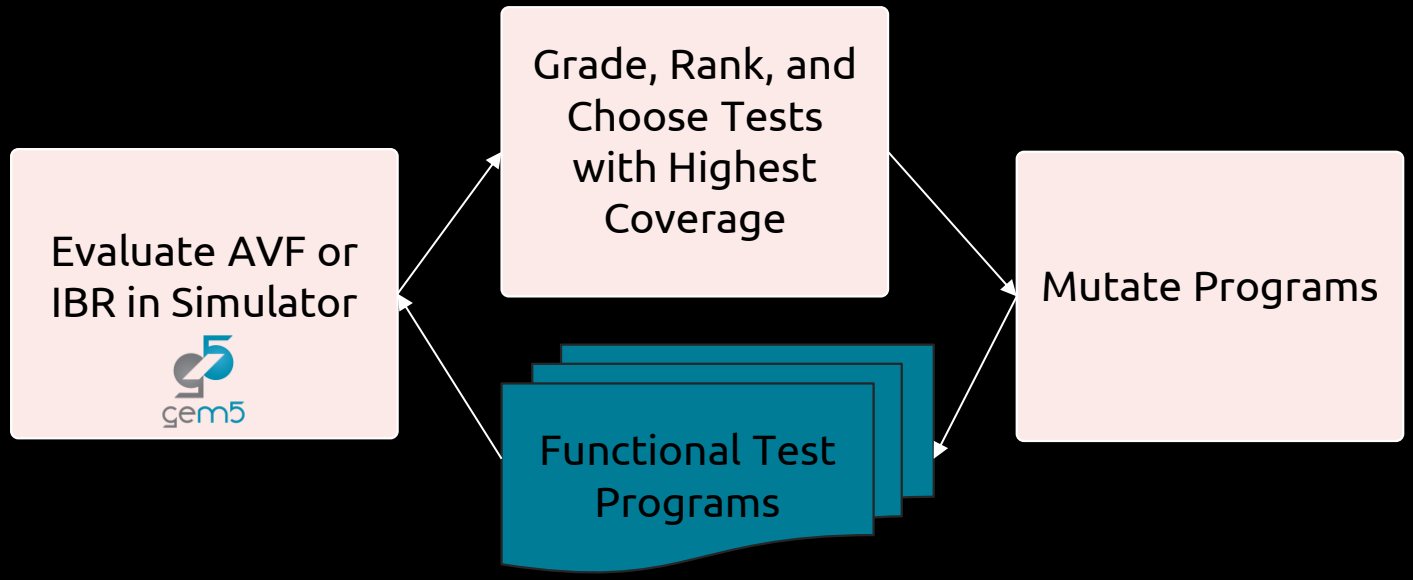
Periodicity of the tests short enough to catch degraded parts before affecting real compute

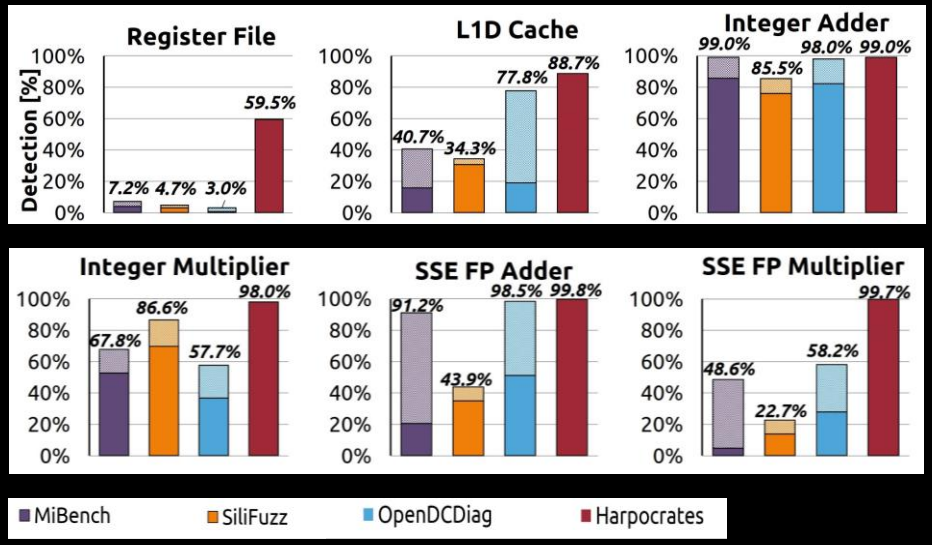
Testing should not affect the overall utilization of the servers

Harpocrates

Can we craft **high-coverage** functional tests, **targeted** at specific hardware blocks and **specific fault models**, in an **automated** manner?

Adapts hardware fuzzing techniques to automatically generate functional tests.
Hardware Coverage metrics for grading tests: AVF: transient faults in arrays; IBR: stuck-at faults in functional units
Maximizing hardware coverage ➔ Higher likelihood of catching a defect that manifests with given fault model





 N. Karystinos, O. Chatzopoulos, G. Fragkoulis, G. Papadimitriou, D. Gizopoulos, S. Gurumurthi, Harpocrates: Breaking the Silence of CPU Faults through Hardware-in-the-Loop Program Generation, ISCA 2024

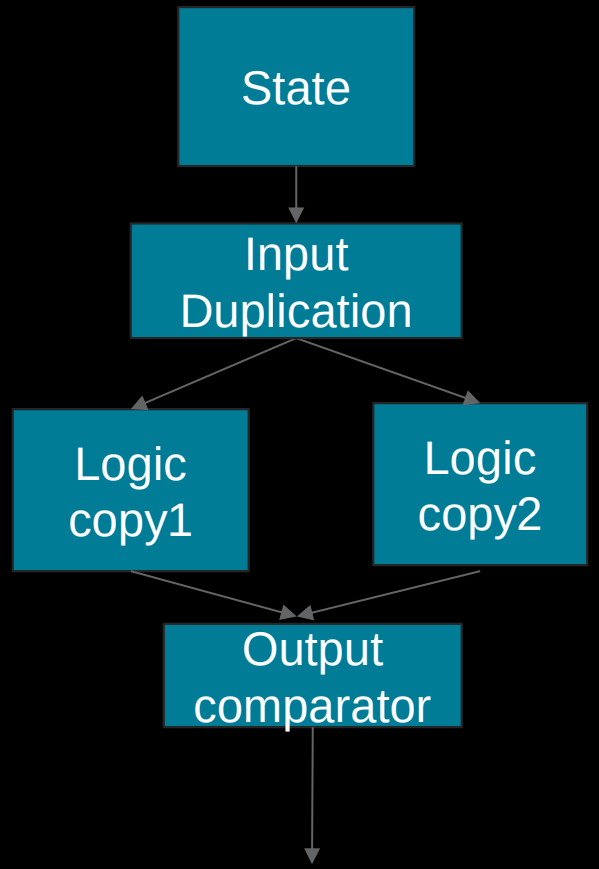
Reliability Architecture

Precise techniques that approach coverage of “big hammer” techniques

Metrics that quantify ROI for protection of design blocks

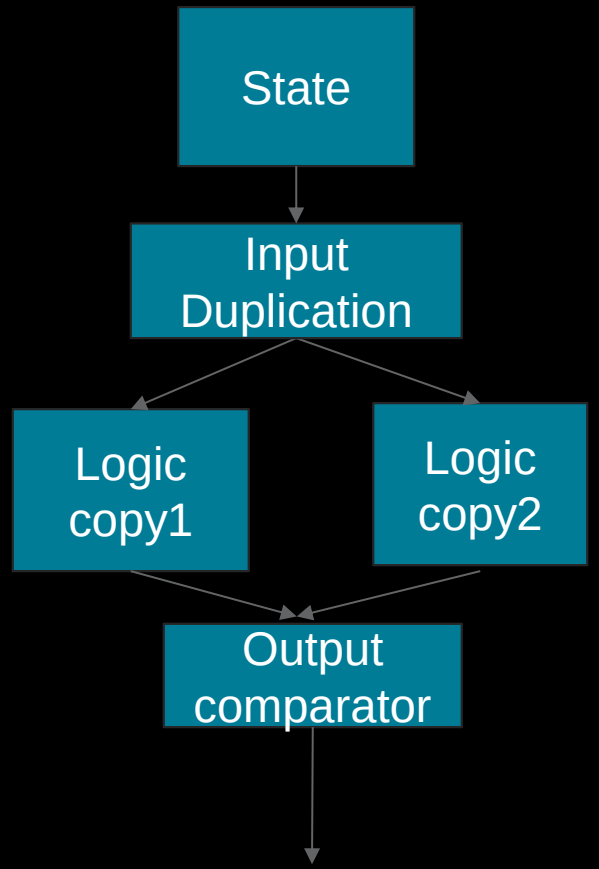
Techniques to target reliability architecture at small delay faults

Big Hammer Techniques



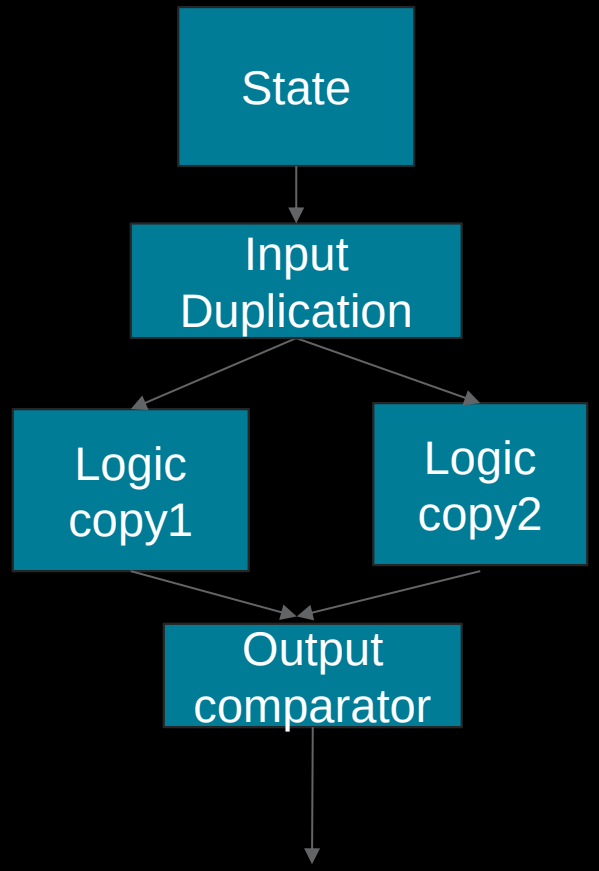
Use multiple copies of logic to check each other

Examples: lockstep, redundant multi-threading



Theoretically can cover a large portion of a design

Can be run during regular operation in fleet

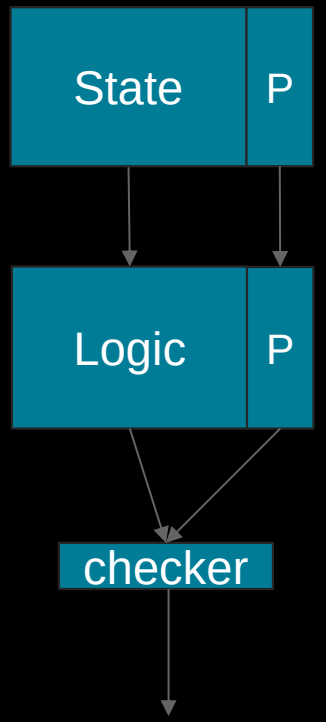


Intrusive to design

Will not be replicating the electrical conditions seen in mission mode

Cost (performance/power/area)

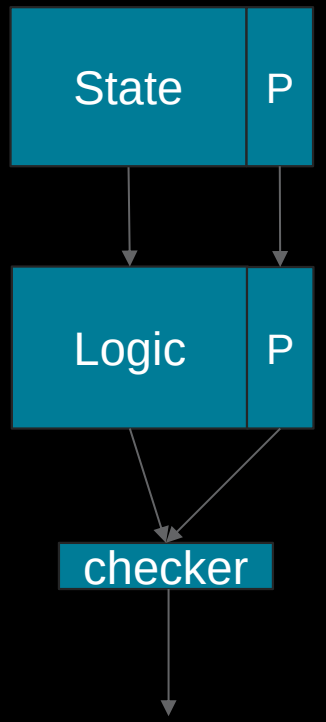
Precise Techniques



Information redundancy

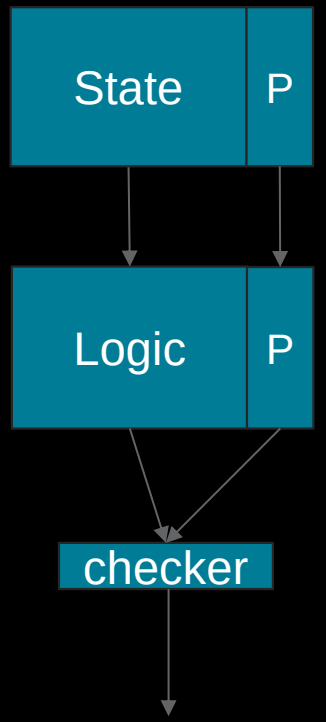
Protect smaller sections of logic with lower-overhead protection techniques

Examples: parity checking, error correction codes (ECCs), parity prediction



Lower overhead to design

Better diagnosability

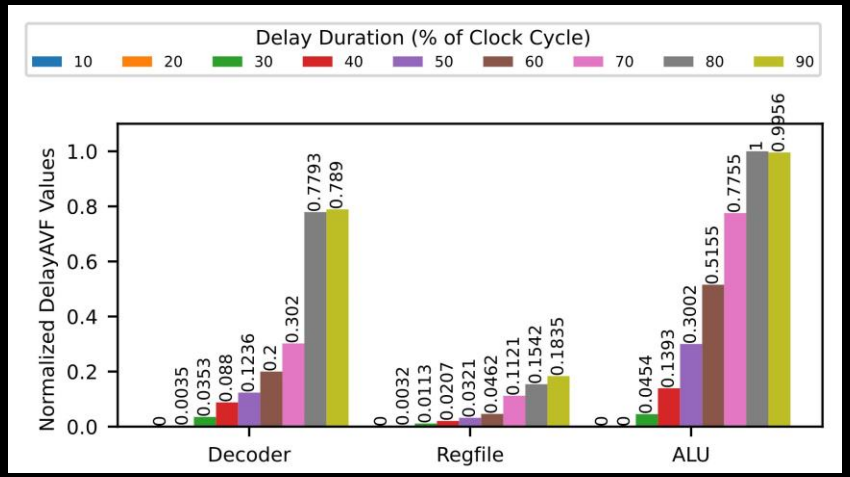


Covers only portions of the design and hence requires identification of logic to protect

Difficult to reason about the ROI of protecting different sections of the design

DelayAVF

DelayAVF	The probability that a small delay fault in a microarchitectural structure propagates to a program-visible error
DelayACE	A circuit element *e* is $DelayACE_d$ in cycle *i* if an added fixed-length propagation delay *d* results in a program-visible error
Calculating DelayAVF	$$DelayAVF_d(T) = \sum_{\forall e \in T} \sum_{i=1}^{N} \frac{DelayACE_d(e,i)}{N \cdot \lvert E \rvert}$$



P. Deutsch, V. Ulitzsch, S. Gurumurthi, V. Sridharan, J. Emer, M. Yan, DelayAVF: Calculating Architectural Vulnerability Factors for Delay Faults, MICRO 2024 (To Appear)

Industry efforts

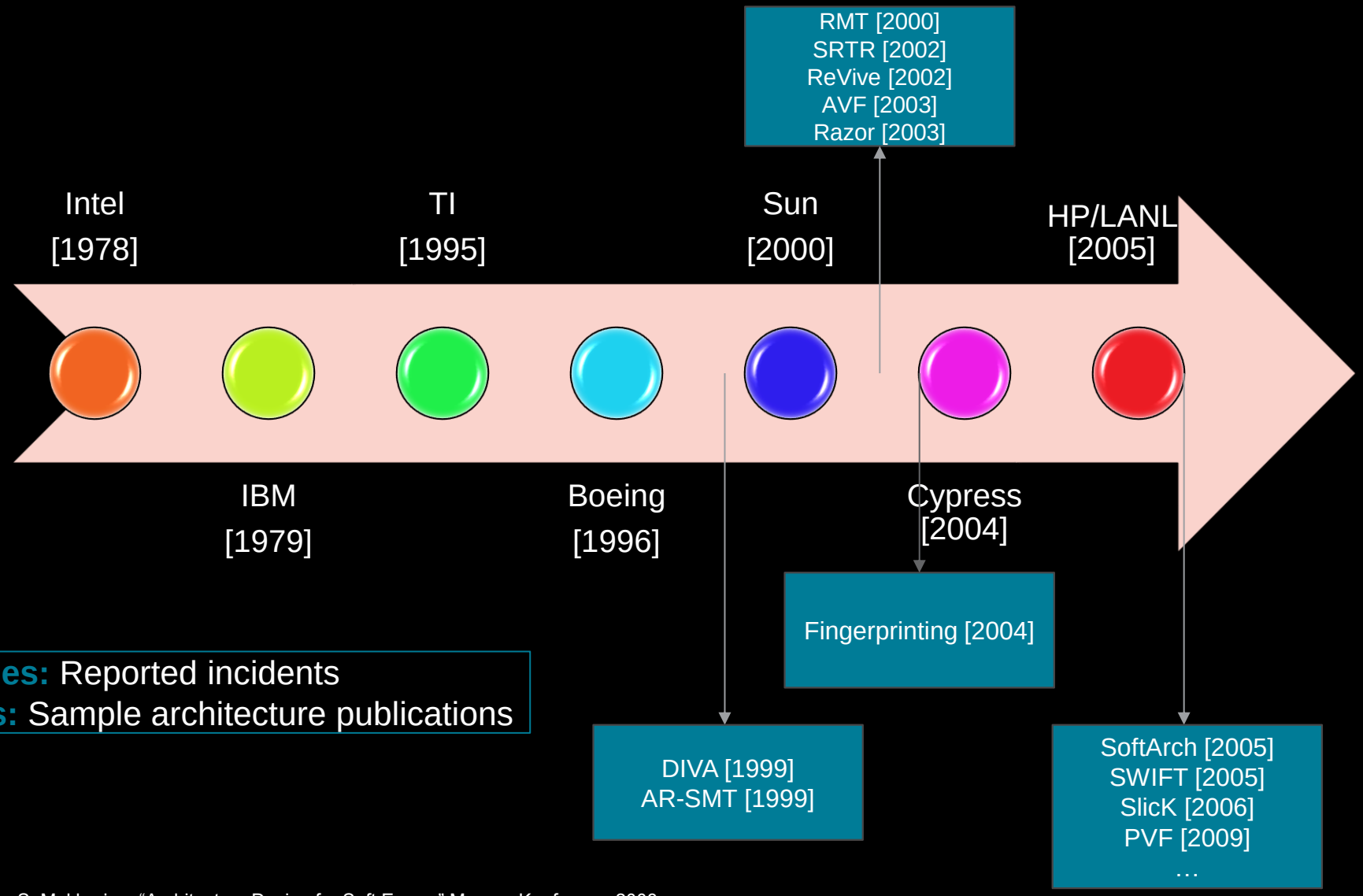
OCP Server Component Resilience: Research Grant Awards

University	Topic
Arizona State	MOTION: Probabilistic Fault Modeling and Test Generation using On-chip Telemetry IntegratION & Generative AI
Auburn	Understanding Test Escapes and SDC Failures in ICs Caused by Transistors with Extreme Device Parameters from Random Manufacturing Variations
Carnegie Mellon	SDC Detection and Correction In Software via Application-level Coding Techniques
Stanford	Mobilizing Hardware and Software Towards SDC Testing, Detection, and Correction
U of Athens	Grade Early and Detect Fast – Tackling Silent Data Corruption through the Power of Microarchitectural Modeling
U of Chicago	Formal Verification of HW Failures & Understanding Impact on Accelerators

6 winning proposals with wide-ranging solutions proposed

Demonstrates strong industry and academic commitment to solving SDC

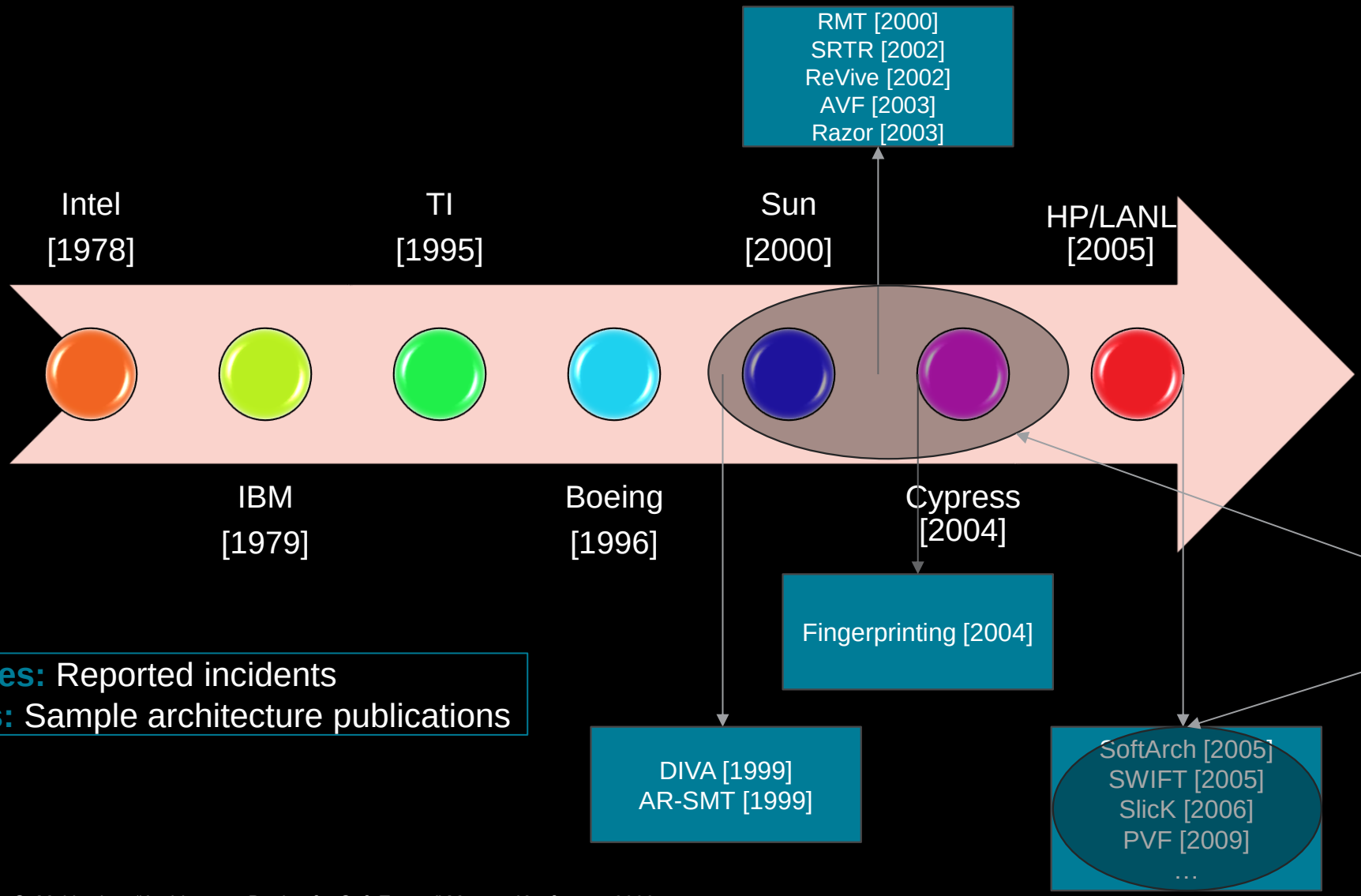
Learning from the Past



Best Practices for Soft Error Remediation

Bubbles: Reported incidents
Boxes: Sample architecture publications

Reference: S. Mukherjee, “Architecture Design for Soft Errors,” Morgan Kaufmann, 2008



Best Practices for Small Delay Fault Remediation

SDFs:
We are here
Still scope for significant innovation

Bubbles: Reported incidents
Boxes: Sample architecture publications

Reference: S. Mukherjee, “Architecture Design for Soft Errors,” Morgan Kaufmann, 2008

Thank You

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Third-party content is licensed to you directly by the third party that owns the content and is not licensed to you by AMD. ALL LINKED THIRD-PARTY CONTENT IS PROVIDED “AS IS” WITHOUT A WARRANTY OF ANY KIND. USE OF SUCH THIRD-PARTY CONTENT IS DONE AT YOUR SOLE DISCRETION AND UNDER NO CIRCUMSTANCES WILL AMD BE LIABLE TO YOU FOR ANY THIRD-PARTY CONTENT. YOU ASSUME ALL RISK AND ARE SOLELY RESPONSIBLE FOR ANY DAMAGES THAT MAY ARISE FROM YOUR USE OF THIRD-PARTY CONTENT.

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc.


AMD
together we advance_