Barcelona Supercomputing Center
Centro Nacional de Supercomputación

@Barcelona Supercomputing Center

Silent Data Corruptions in Computing Systems – Modeling, Measuring, Mitigating Across the Layers

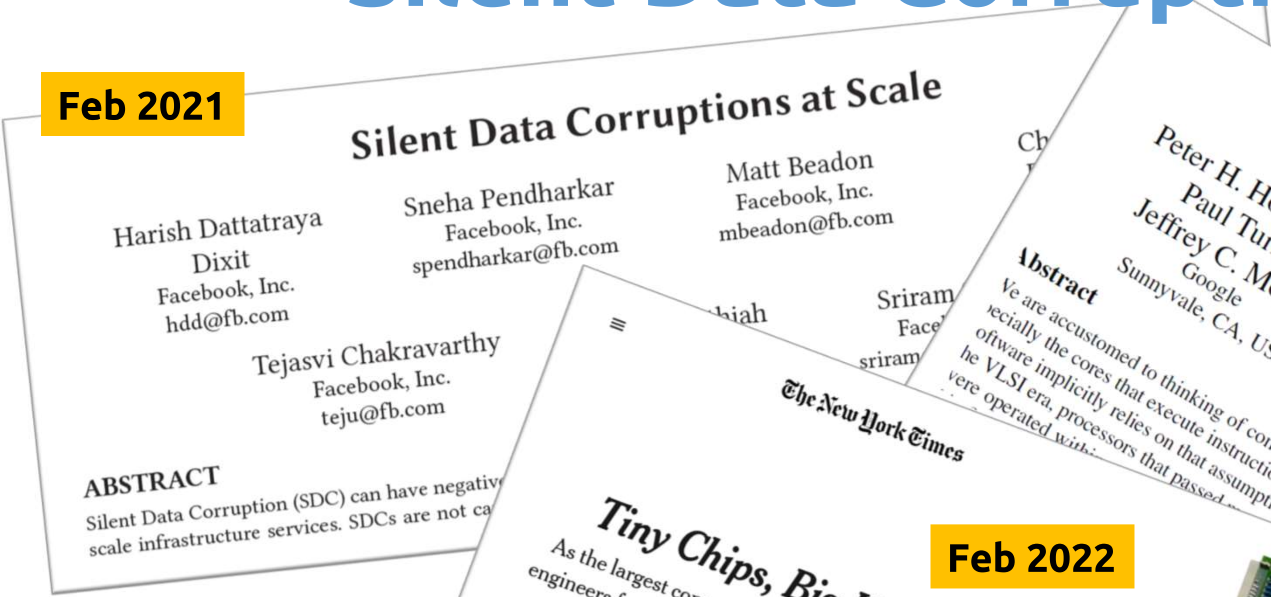
Dimitris Gizopoulos
University of Athens
http://www.di.uoa.gr/~dgizop – http://cal.di.uoa.gr

HELLENIC REPUBLIC
National and Kapodistrian University of Athens
EST. 1837
Computer Architecture Lab

February 1, 2023

Silent Data Corruptions (SDCs)

Feb 2021



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

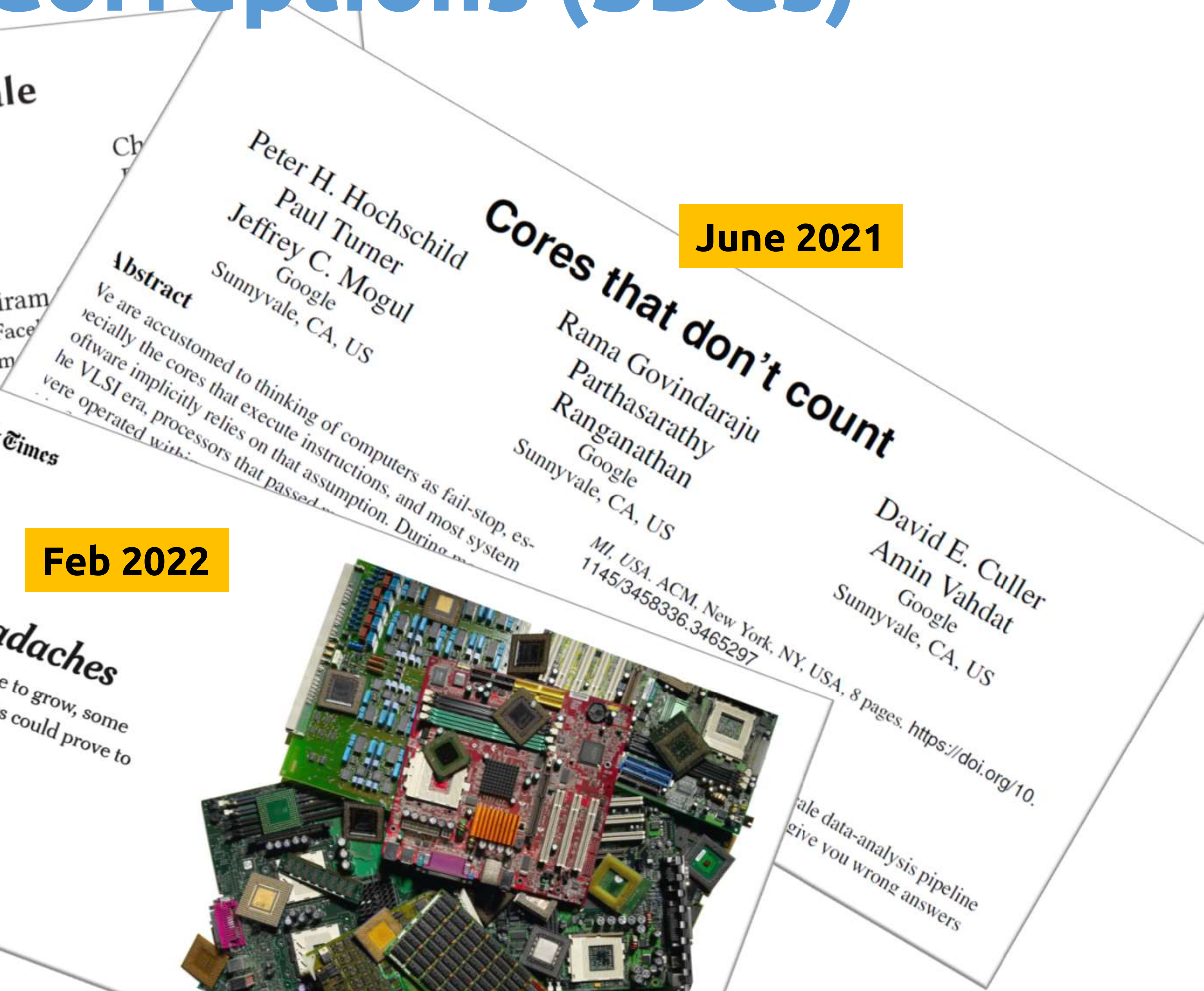
Tejasvi Chakravarthy
Facebook, Inc.
teju@fb.com

Sriram
Face
sriram

ABSTRACT

Silent Data Corruption (SDC) can have negativ
scale infrastructure services. SDCs are not ca

June 2021



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

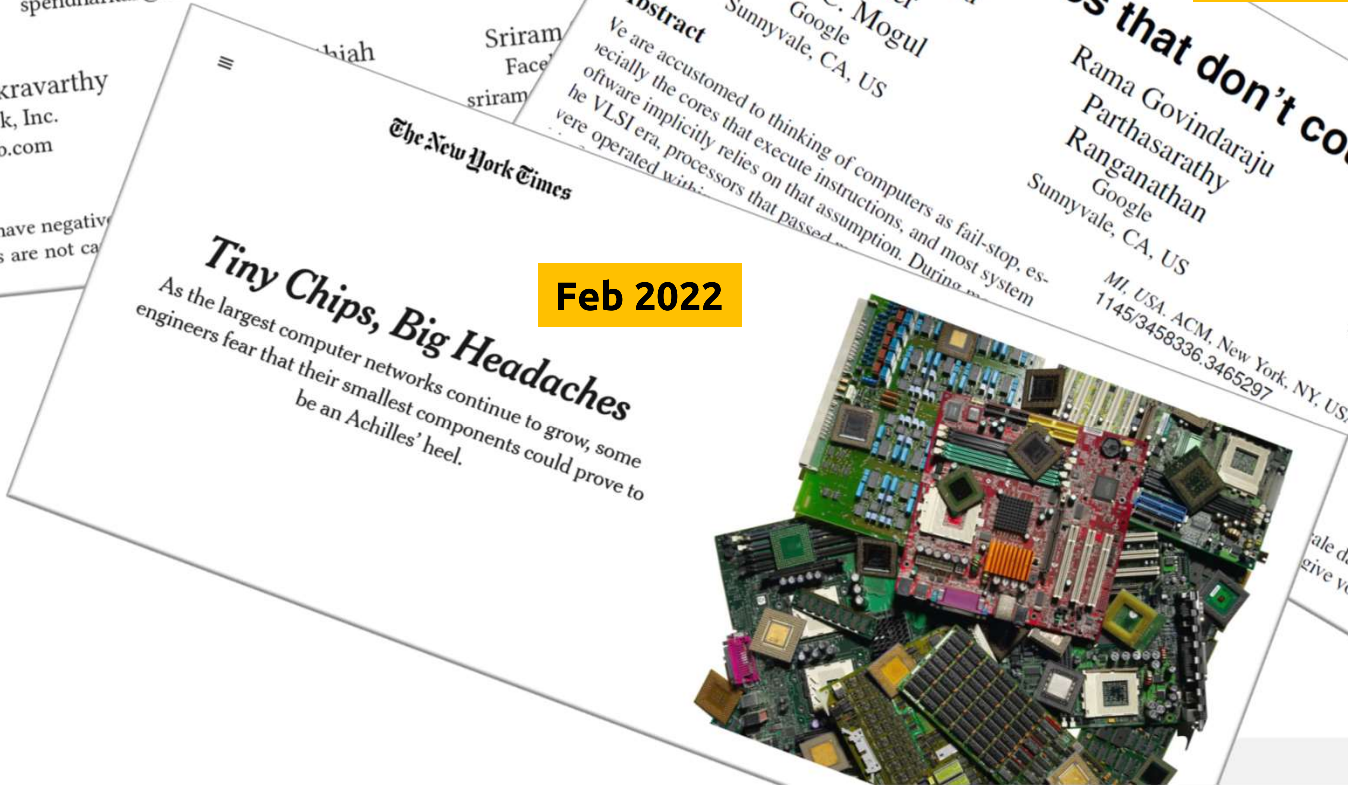
Abstract

Ve are accustomed to thinking of computers as fail-stop, es-
pecially the cores that execute instructions, and most system
oftware implicitly relies on that assumption. During m
he VLSI era, processors that passed
vere operated with

ale data-analysis pipeline
give you wrong answers

MI, USA. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3458336.3465297

Feb 2022



The New York Times

Tiny Chips, Big Headaches

As the largest computer networks continue to grow, some engineers fear that their smallest components could prove to be an Achilles' heel.

Meta Request for Proposals

Feb 2022

JUNE 16, 2022

Announcing the winners of the 2022 Silent Data Corruptions at Scale request for proposals

June 2022

By: Meta Research

Research award winners

Principal investigators are listed first unless otherwise noted.

Hardware failures root causing: Harnessing microarchitectural modeling
Dimitris Gizopoulos (National and Kapodistrian University of Athens)

Lightweight in-production SDC detection tools inspired by coding theory
Rashmi Vinayak (Carnegie Mellon University)

Quarantine and vaccination framework for SDC mitigation at-scale
Devesh Tiwari (Northeastern University)

Software-hardware strategies for enhancing ML application resilience
Prashant Nair (University of British Columbia), Karthik Pattabiraman (University of British Columbia), Sathish Gopalakrishnan (University of British Columbia)

Testing for corrupt execution errors
Caroline Trippel (Stanford University)

SDCs – as simple as this

- Operation: addition 5 + 6
  - **Correct result should be: 11**
  - **CPU generating SDC says: 13**

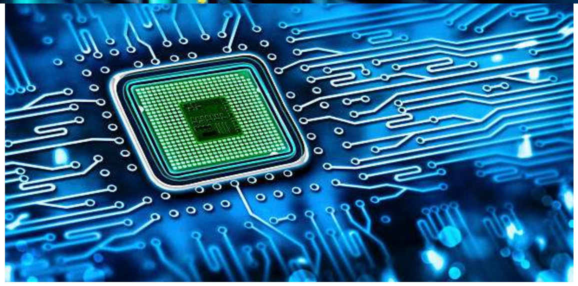
- All subsequent calculations continue with the wrong 13 value

- When is difference noticed?
  - When something goes grossly wrong!
  - Maybe never

What is a Silent Data Corruption or Silent Error?

- Program runs to end (hours, days)
- Output is produced
- System fully responsive
- No detection
  - No ECC, exception, …

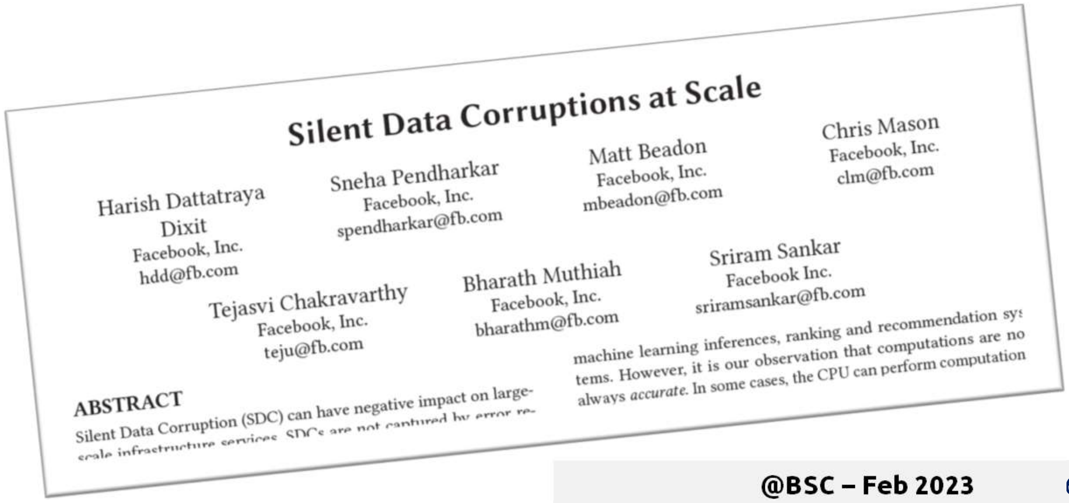
- Output is wrong (*data corruption*)
- **Nobody knows !** ☹ *(silent)*



Meta report: example CPU generating SDCs

- Exponentiation function gives wrong result on Core 59 of a certain CPU in the Meta fleet
  - *Int*(1.1^{53}) = 0 *(should be 156)*
- Result is a file size; zero means no file exists
- Other exponents work fine
  - *Int*(1.1^{52}) = 142

- **Systematic but very rare**



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

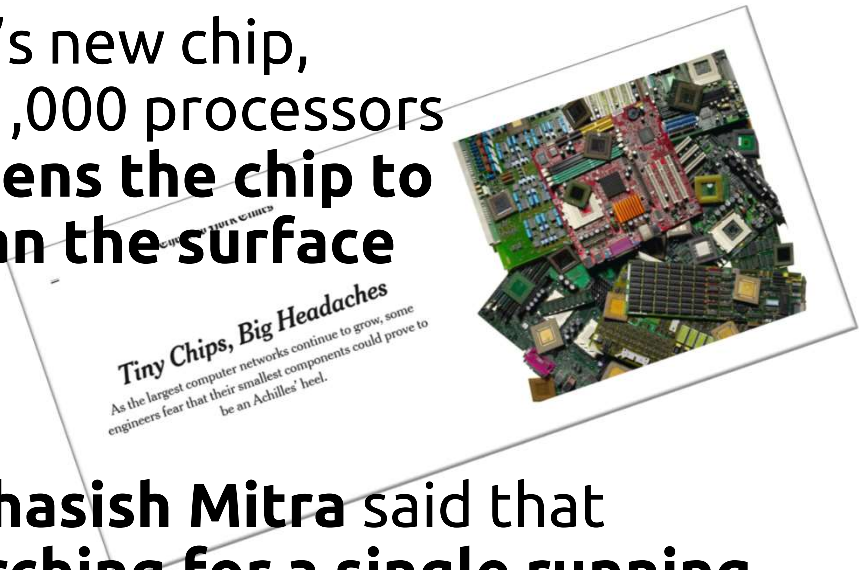
Sriram Sankar
Facebook Inc.
sriramsankar@fb.com

ABSTRACT
Silent Data Corruption (SDC) can have negative impact on large-
scale infrastructure services. SDCs are not captured by error re-

machine learning inferences, ranking and recommendation sys
tems. However, it is our observation that computations are no
always *accurate*. In some cases, the CPU can perform computation

How rare? A metaphore

- “Tracking down these errors is challenging, said **David Ditzel**, chairman and founder of Esperanto Technologies. […] He said his company’s new chip, which is just reaching the market, had 1,000 processors made from 28 billion transistors. **He likens the chip to an apartment building that would span the surface of the entire United States.**



Tiny Chips, Big Headaches

As the largest computer networks continue to grow, some engineers fear that their smallest components could prove to be an Achilles' heel.

- Using Mr. Ditzel’s metaphor, **Prof. Subhasish Mitra** said that finding new errors was a little **like searching for a single running faucet, in one apartment in that building, that malfunctions only when a bedroom light is on and the apartment door is open.**

Silent Corruptions/Errors at Scale

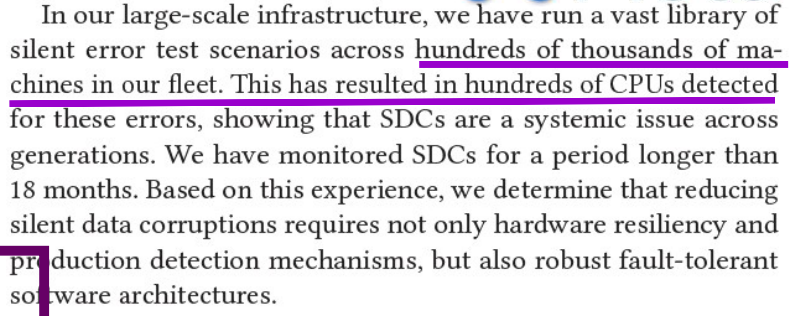
- Silence means results:
  - Are considered correct
  - Are distributed at scale
- Data centers, Cloud, Supercomputers
- Effects of SDCs may take weeks or months before getting noticed (if ever) ☹

Who needs to know about SDCs ?

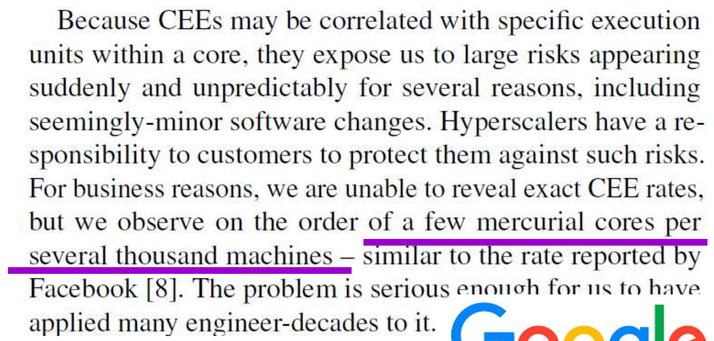
- **Hyperscalers, cloud services providers** – detect faulty CPUs and move out of production
- **Software developers** – “harden” software to bypass/recover from faulty hardware structures
- **CPU vendors** – root cause, feedback for design, manufacturing, testing improvements
- **Users** – better not know ☺

SDC rates/behaviors by Hyperscalers

- **Meta and Google report ~1 CPU in a 1000 generates SDCs**
  - Or 100 to 1000 DPPM (defective parts per million)
- Info we know:
  - Systematic/reproducible events
  - Affect the same instruction all the time
  - Data dependent
  - Chip age dependent
  - Voltage/frequency dependent
  - Attributed to manufacturing/design defects



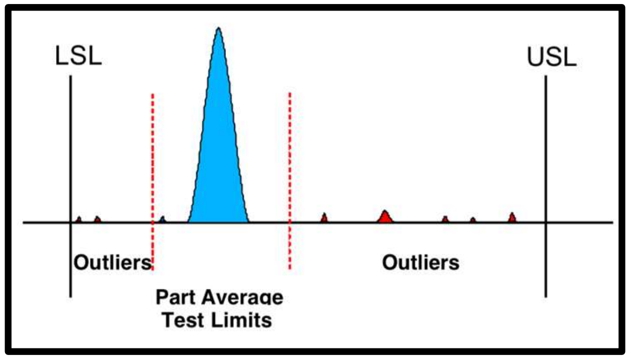
In our large-scale infrastructure, we have run a vast library of silent error test scenarios across hundreds of thousands of machines in our fleet. This has resulted in hundreds of CPUs detected for these errors, showing that SDCs are a systemic issue across generations. We have monitored SDCs for a period longer than 18 months. Based on this experience, we determine that reducing silent data corruptions requires not only hardware resiliency and production detection mechanisms, but also robust fault-tolerant software architectures.

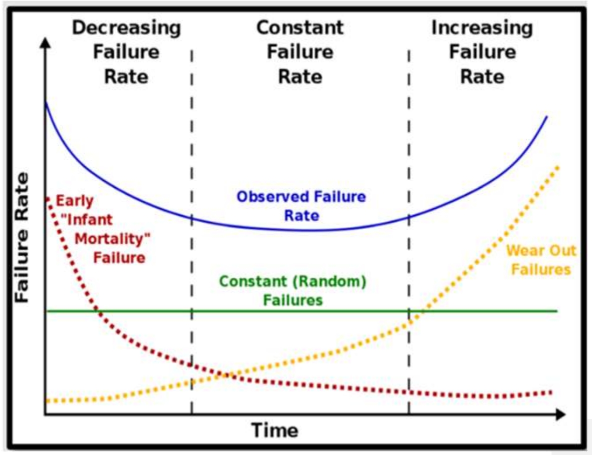


Because CEEs may be correlated with specific execution units within a core, they expose us to large risks appearing suddenly and unpredictably for several reasons, including seemingly-minor software changes. Hyperscalers have a responsibility to customers to protect them against such risks. For business reasons, we are unable to reveal exact CEE rates, but we observe on the order of a few mercurial cores per several thousand machines – similar to the rate reported by Facebook [8]. The problem is serious enough for us to have applied many engineer-decades to it.

Likely causes for SDCs (1)

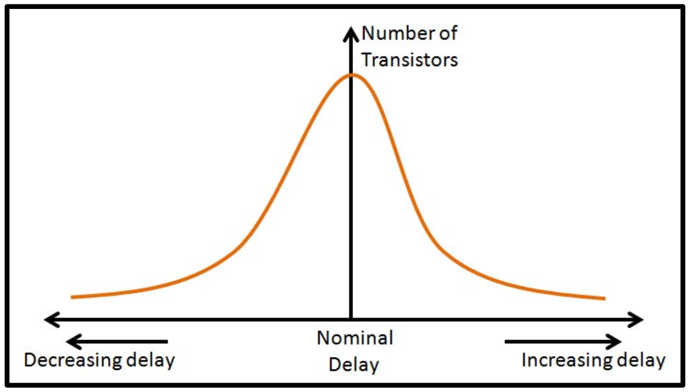
- "I see three subproblems — time zero defects, which represent test escapes, early mortality defects (a.k.a. latent defects), and aging related defects that occur later during the lifecycle of the product in its system."
  - **Janusz Rajski**, vice president of engineering for Tessent at Siemens EDA (https://semiengineering.com/screening-for-silent-data-errors/; Jan 2023)





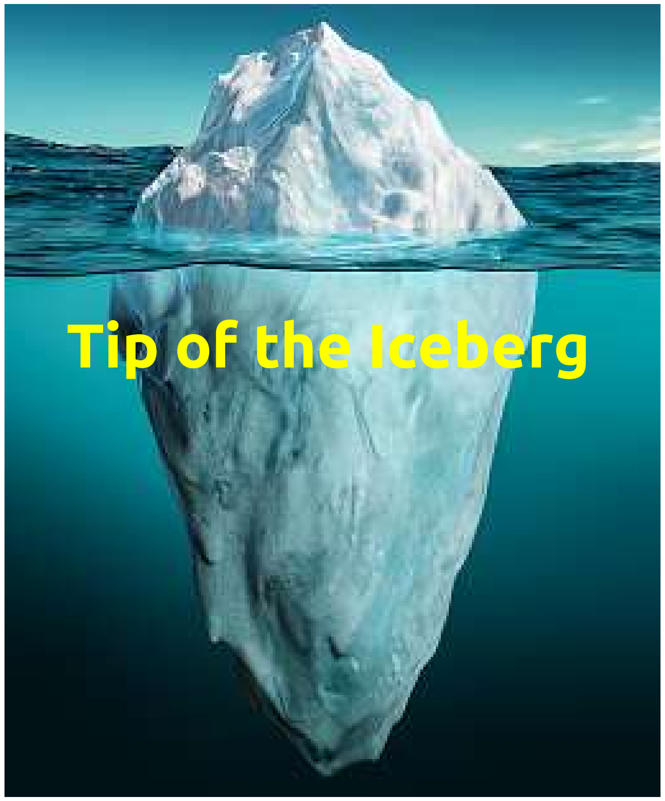
Likely causes for SDCs (2)

- "Are Timing Marginalities due to Process Variations the Source of Silent Data Corruption?"
- Transistor delays and delay variability from process variations is greatly accentuated in low voltage power saving modes.
  - **Adit Singh** (Auburn U), Keynote at IEEE VLSI Test Symposium, April 2022.



Do we know the true rates ?

- **~1 CPU every 1000**
- But these rates came only after
  - Customers/users noticed and complained
  - Months/years of debug
- What about **SDC** **Escapes** ?
  - They are everywhere around us
  - We don't know how many
  - We don't know where
    - Which calculation is affected – Which result is corrupted



Need to know to mitigate SDCs

- **Real rates of SDCs**
  - 1 in 1000 or more ?
- **Guilty hardware structures**
  - Vulnerable to faults & bugs
- **Suspect software codes**
  - Susceptible instructions & calculations

- Thus, we need:
  - **Root cause (origin) should exist** ("faulty", "buggy" CPUs!)
  - **Full system evaluation** (hardware, architecture, OS, software)
  - **Complete, end-to-end execution** (need to know the output to check corruption)
  - **Fine-grain observability** (need to know internal activity of hardware while software runs)

Collecting SDCs information (#1)
Large Fleets (Hyperscalers)

- Collect data from large fleets
  - > 2.5M servers – year 2016, Gardner
  - > 100K – year 2022
  - > 4M – year 2021
  - > 500 K – year 2012
  - >1 500 000 ? – year 2020
  - *need to own such fleets*
  - *does hardware provide information ?*
- How long does it take?
  - Years before results are collected, processed and made public
  - *results still valid ?*
  - *hw+sw always evolve*

Collecting SDCs information (#2)
Own/Design the CPU

- Intel – AMD – Arm
  - Actively investigating the problem
- Detailed RTL (pre-synthesis), gate-level (post-synthesis), layout (post-P&R) chip models
  - Very close to real manufactured chips
  - Best case to analyze faults and bugs
  - *no such models available to research community*
  - *even if they were, it's impossible to simulate long executions*
  - *no full-system, no end-to-end execution*
- RISC-V
  - RTL available but not yet the CPUs that hyperscalers use

Microarchitectural modeling for SDCs analysis and mitigation

- **Microarchitectural, performance simulators to the rescue**

gem5 · sniper · ChampSim/ ChampSim · s5z / zsim

- Model faults and bugs (describe it)
- Full system simulation (OS and application)
- End-to-end simulation (high throughput)
- Fine-grain observability (hardware and software)

SDCs Assessment Options – Speed

- Layer of abstraction
  - Software – **too high level, no hardware info**
  - Architecture/ISA – **no hardware info**
  - Microarchitecture – **early & flexible hw model**
  - RTL – **late hw model, extremely slow**
  - Silicon – **too late, limited observability**

Abstraction Layer	Performance (cycles/sec) *
Software	3×10^9
Architecture	6×10^7
Microarchitecture	**3×10^6 (simple CPU)** **2×10^5 (detailed CPU)**
Flip-flop	6×10^2

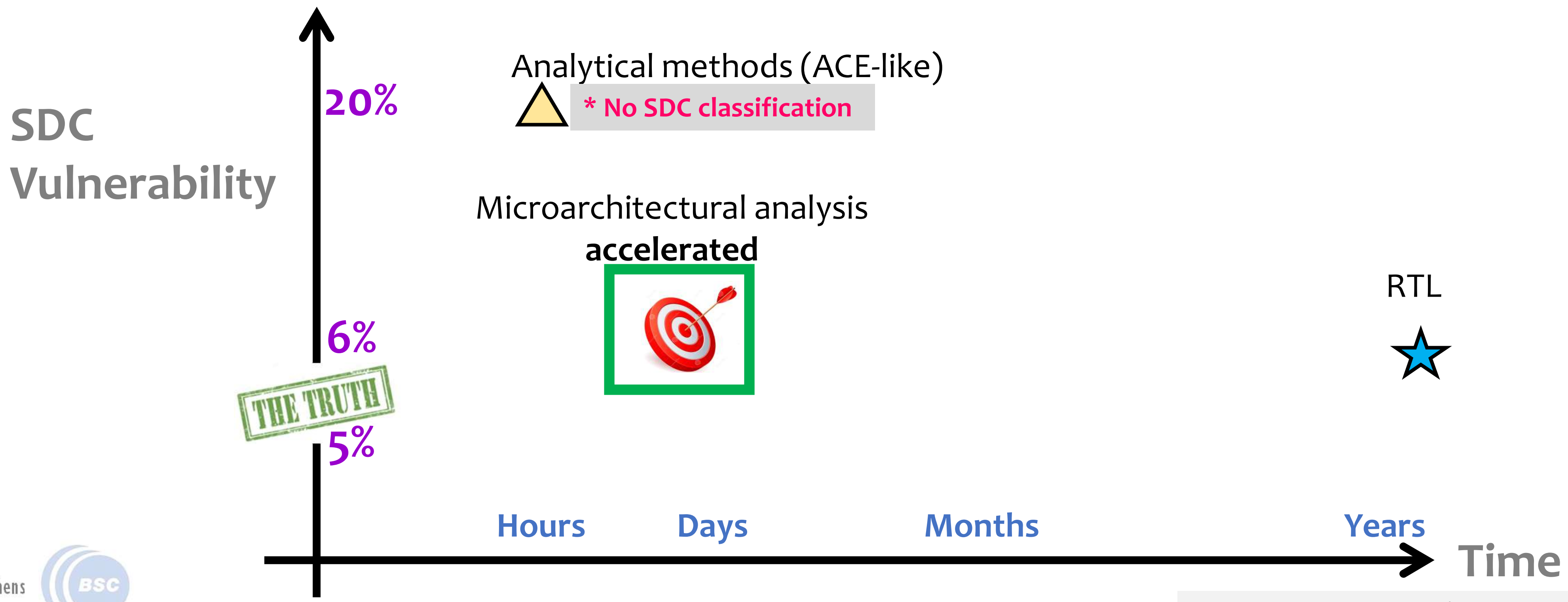
~10 000x

ONE HOUR HERE IS SEVEN YEARS ON EARTH
GREAT! WE WILL RUN CHIP LEVEL SIMULATIONS ON THIS PLANET
imgflip.com

* J.Goodenough, R.Aitken, "Post-Silicon is Too Late – avoiding the $50 Million Paperweight Starts with Validated Designs, ACM/IEEE DAC 2010.
* H.Cho, S.Mirkhani, C.-Y.Cher, J.A.Abraham, S.Mitra, "Quantitative Evaluation of Soft Error Injection Techniques for Robust System Design", ACM/IEEE DAC 2013.

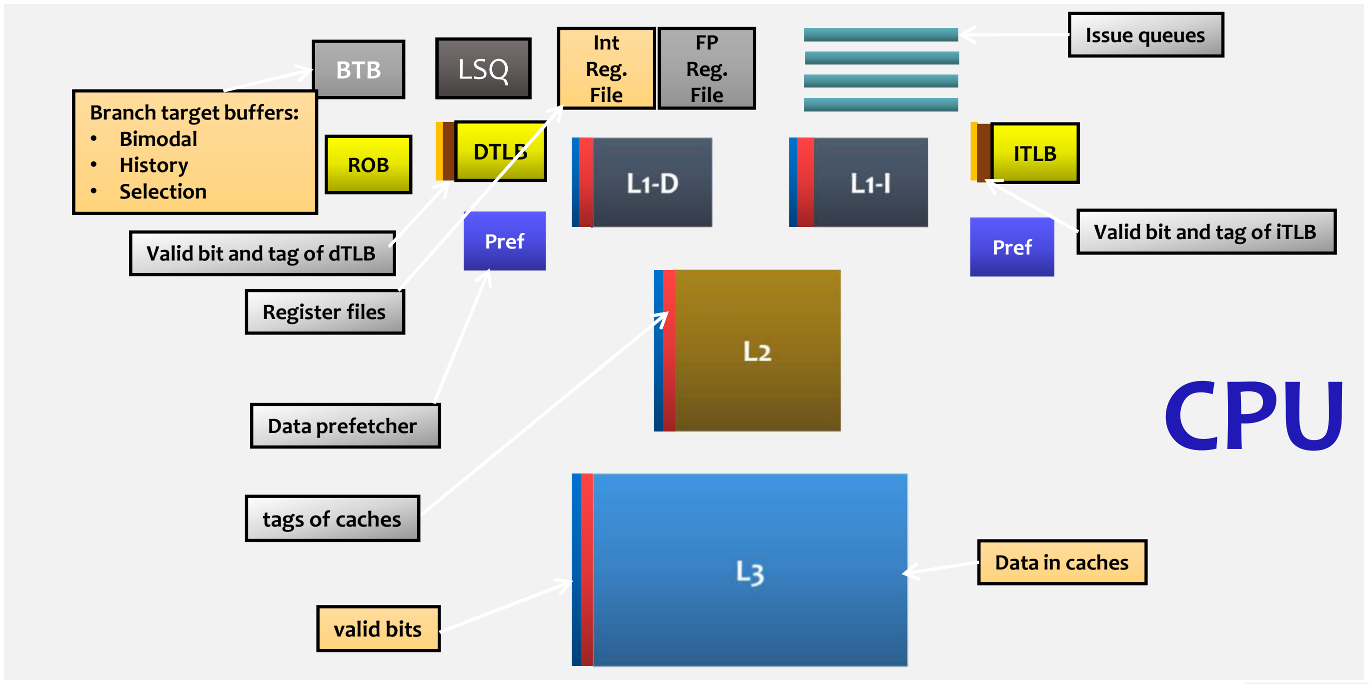
The objective visualized

- **Fast & Accurate** SDC measure/predict and root cause

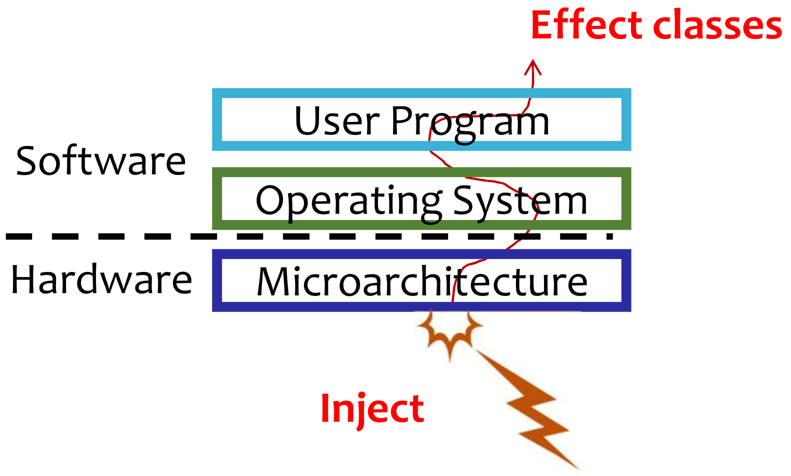


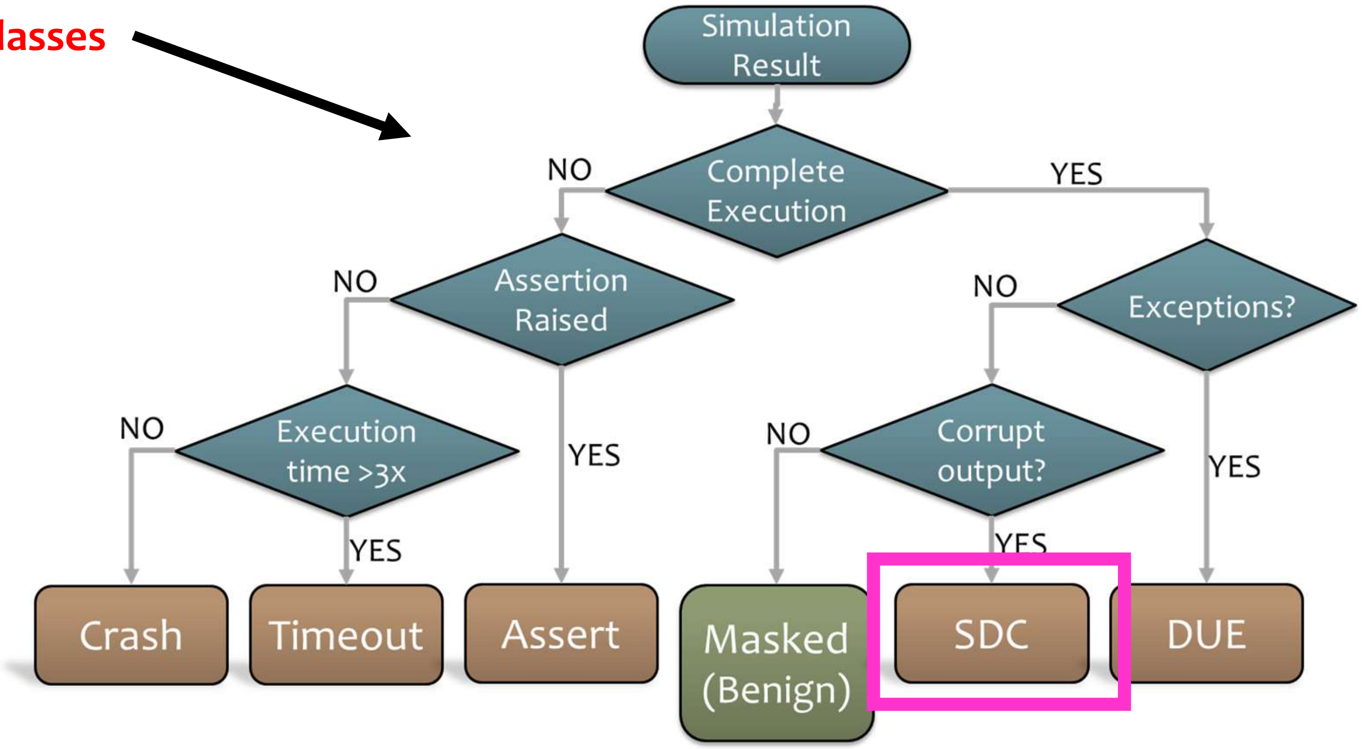
Microarchitectural modeling (example: gem5)

- **More than 50 hardware structures for root cause injection (bugs, faults)**: Caches, Registers, Register Files, Buffers, Queues, BPUs, BTBs, TLBs, Translation caches, etc.



Injection Infrastructure on gem5 (full system, cycle accurate)



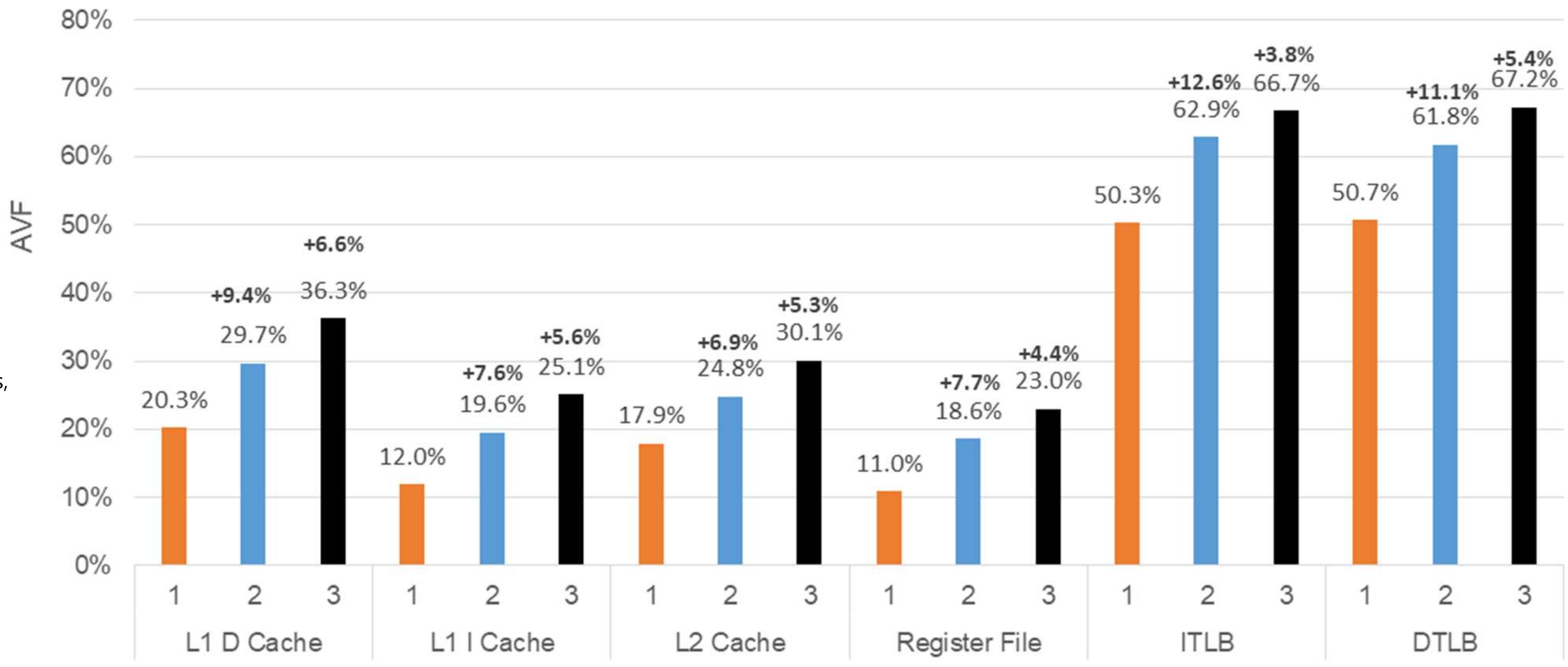


- Soft errors (SEUs, sMBUs)
- Hard errors
- Timing errors
- Design bugs

100x – 1000x speedup over baseline gem5 simulation

Soft errors – SEU and sMBU Case Study

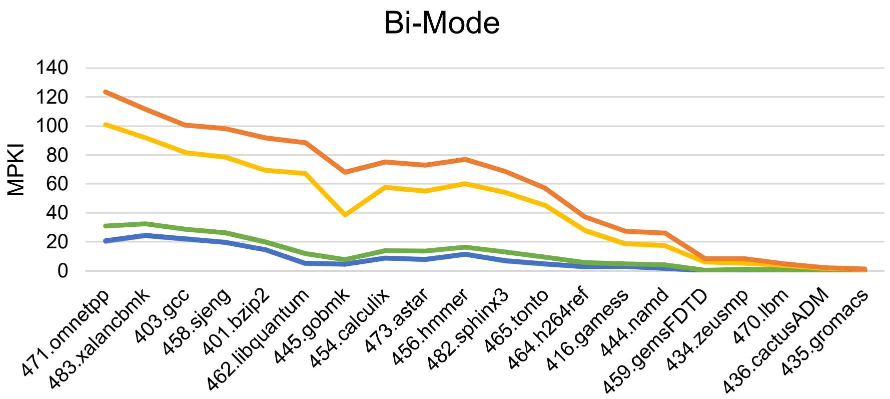
- SEUs and sMBUs – 1/2/3 bits flipped per component

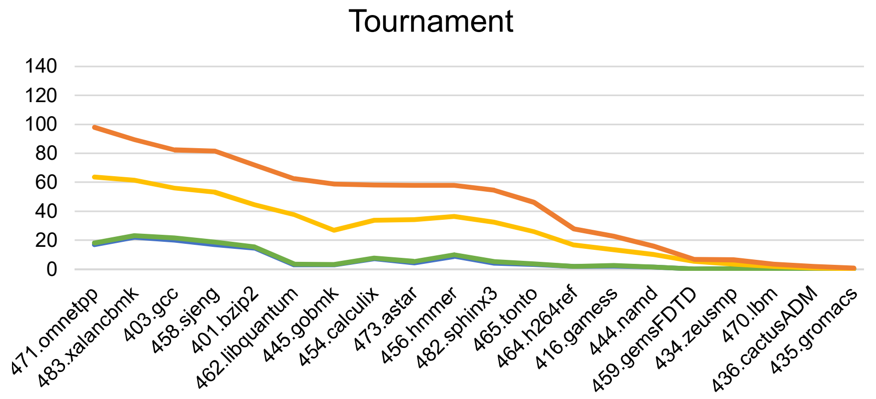


A.Chatzidimitriou, G.Papadimitriou, C.Gavanas, G.Katsoridas, and D.Gizopoulos, "Multi-Bit Upsets Vulnerability Analysis of Modern Microprocessors", IEEE International Symposium on Workload Characterization (IISWC 2019), Orlando, Florida, USA, November 2019.

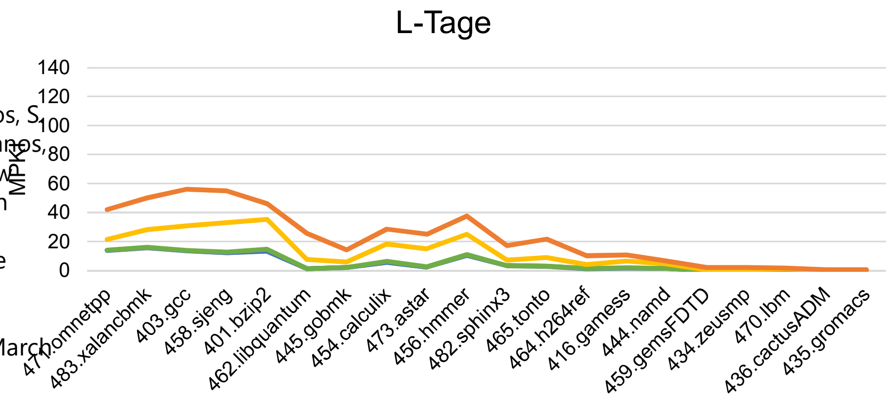
Case Study – Undervolted Predictors (SRAM permanent faults)

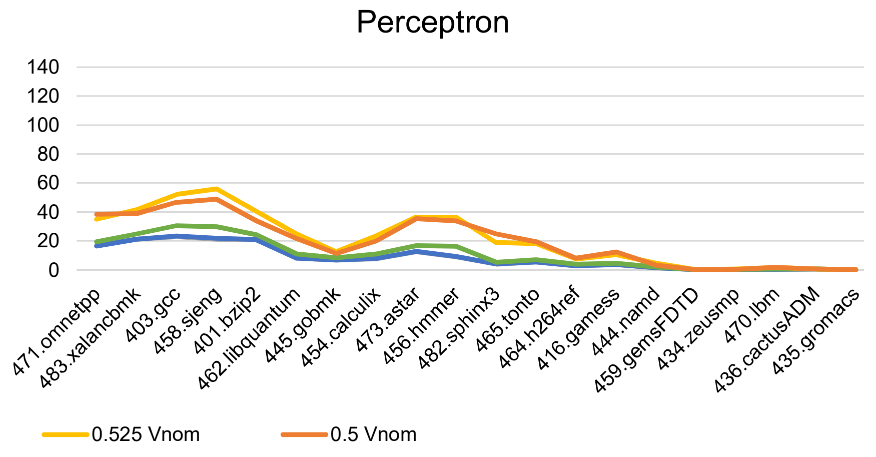
MPKI (mispreds per K instructions)



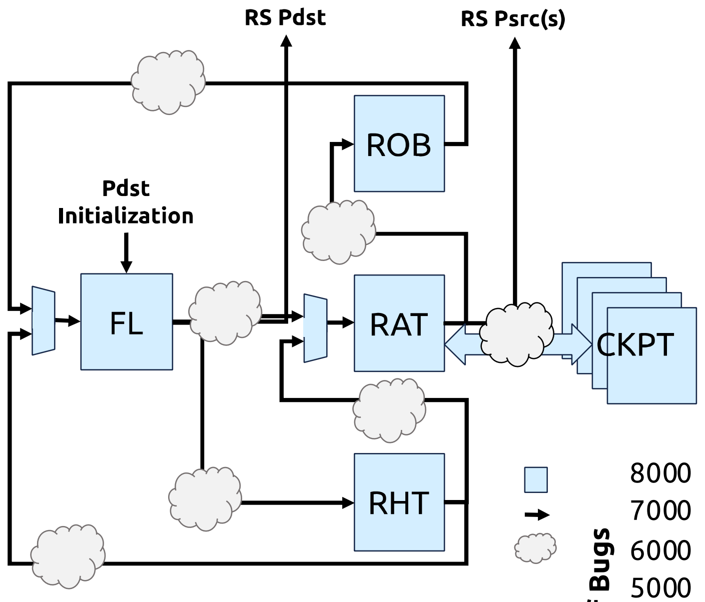


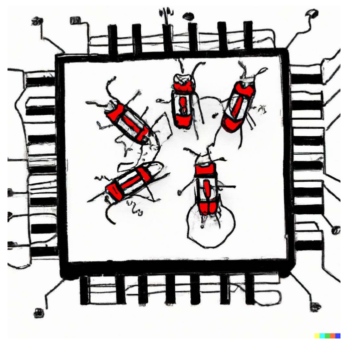
A. Chatzidimitriou, G. Papadimitriou, D. Gizopoulos, S. Ganapathy, and J. Kalamatianos, "Assessing the Effects of Low Voltage in Branch Prediction Units", IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2019), Madison, Wisconsin, USA, March 2019.

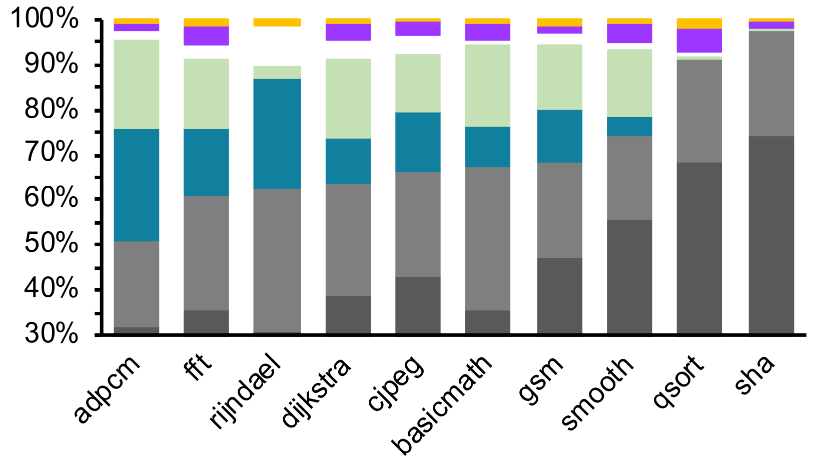


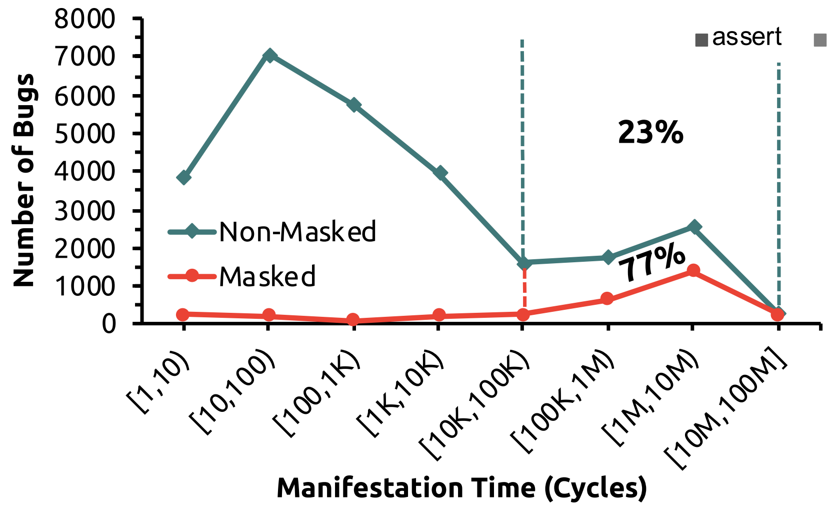


Design Bugs (OoO core, renaming logic)





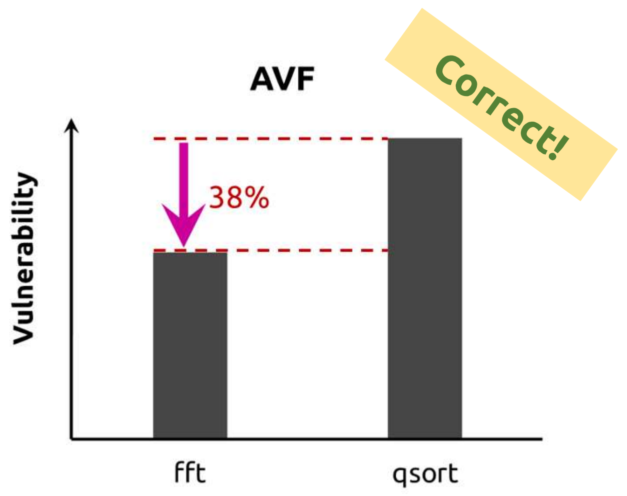


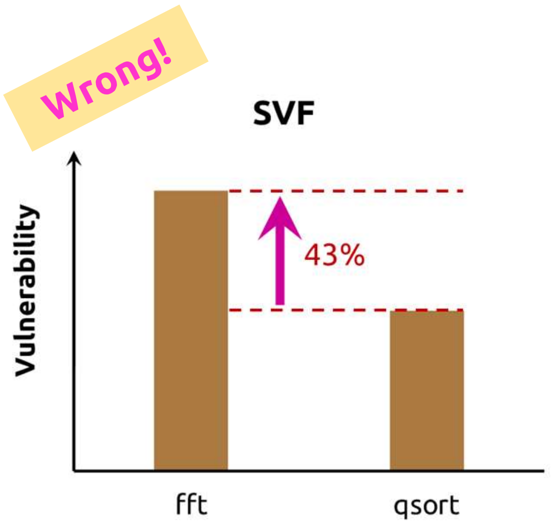


Y. Sazeides, A. Gerber, R. Gabor, A. Bramnik, G. Papadimitriou, D. Gizopoulos, C. Nicopoulos, G. Dimitrakopoulos, and K. Patsidis, "IDLD: Instantaneous Detection of Leakage and Duplication of Identifiers used for Register Renaming", ACM/IEEE International Symposium on Microarchitecture (MICRO 2022), Chicago, Illinois, USA, October 1-5, 2022.

Why not at the Software Layer ?

- Because software-only analysis is **fast but wrong** ☹
- (full system) Architectural Vulnerability Factor (**AVF**) vs.
- (partial) Software Vulnerability Factor (**SVF**)

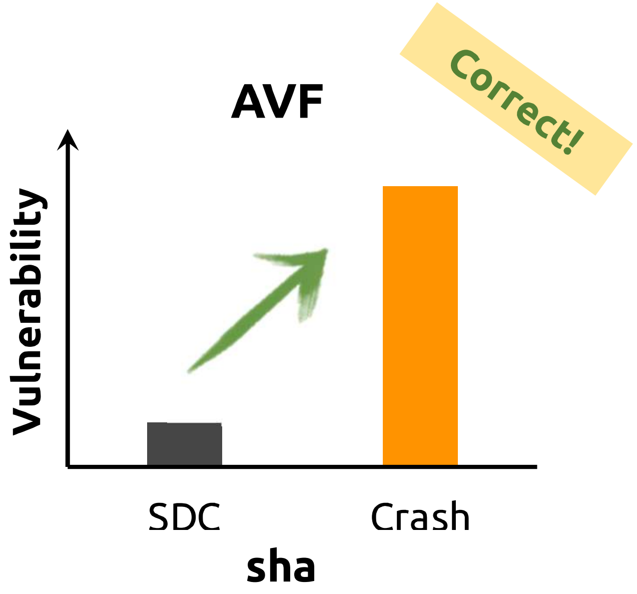


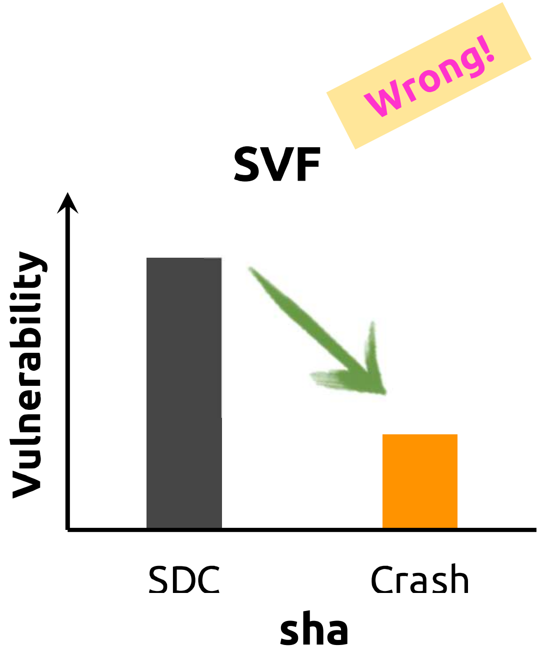


G. Papadimitriou, D. Gizopoulos, "Demystifying the System Vulnerability Stack: Transient Fault Effects Across the Layers", IEEE International Symposium on Computer Architecture (ISCA 2021), June 2021.

SDC measurements at the Software Layer

- Still wrong …



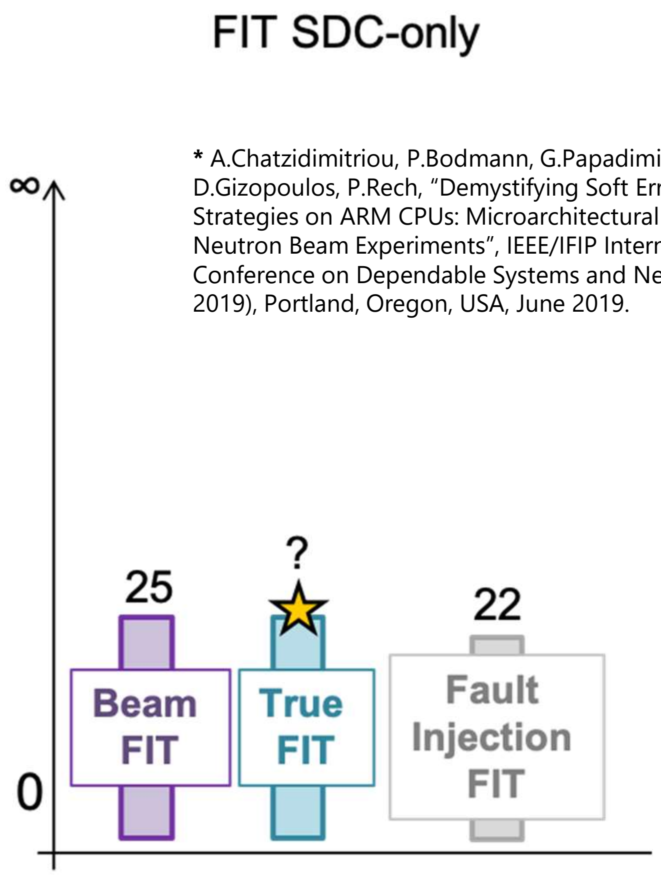


G. Papadimitriou, D. Gizopoulos, "Demystifying the System Vulnerability Stack: Transient Fault Effects Across the Layers", IEEE International Symposium on Computer Architecture (ISCA 2021), June 2021.

Is it Accurate? Validation to Chips Beaming (1)

- **ARM Cortex-A9 CPU core**
  - gem5 SEU fault injections vs. neutron beaming
  - 11 applications
  - full system (Linux)
  - End-to-end workload execution (to record SDCs)
- **SDCs FIT rates very close**

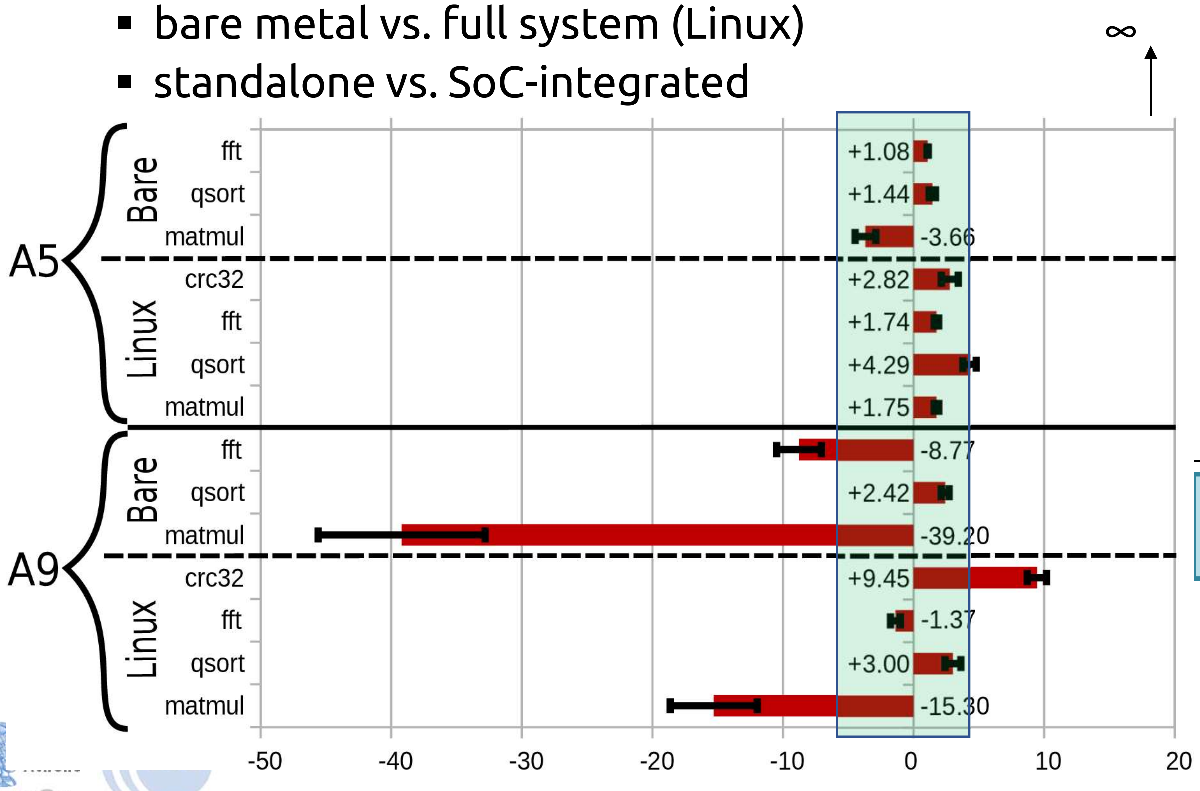
* A.Chatzidimitriou, P.Bodmann, G.Papadimitriou, D.Gizopoulos, P.Rech, "Demystifying Soft Error Assessment Strategies on ARM CPUs: Microarchitectural Fault Injection vs. Neutron Beam Experiments", IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2019), Portland, Oregon, USA, June 2019.

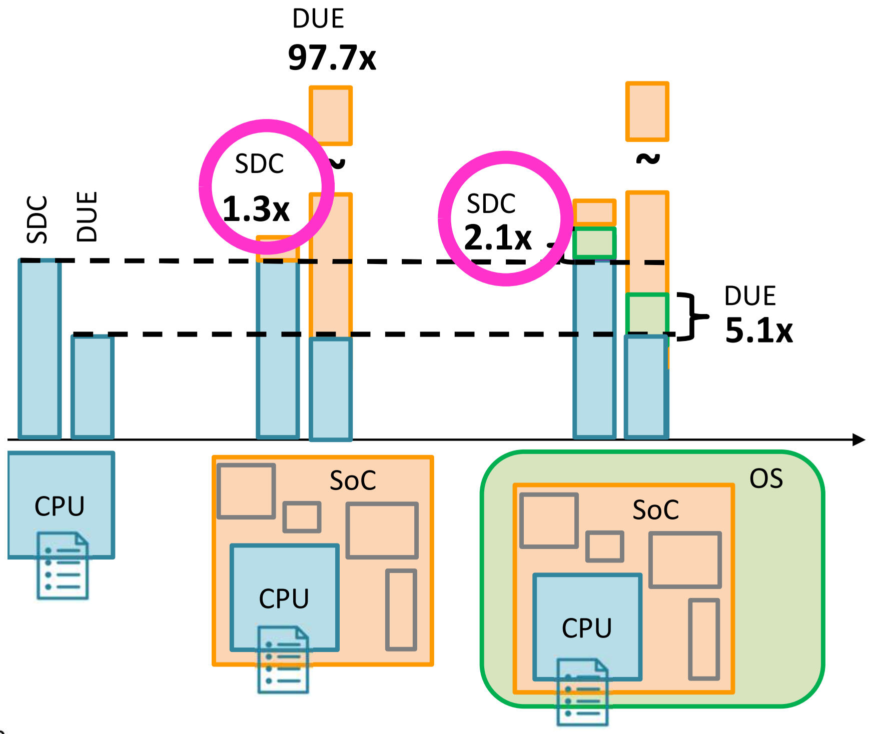


Is it Accurate? Validation to Chips Beaming (2)

- **ARM Cortex-A5 and Cortex-A9 CPU cores**
  - gem5 SEU fault injections vs. neutron beaming
  - bare metal vs. full system (Linux)
  - standalone vs. SoC-integrated

* P.Bodmann, G.Papadimitriou, R.L.Rech Junior, D.Gizopoulos, P.Rech, "Soft Error Effects on Arm Microprocessors: Early Estimations vs. Chip Measurements", IEEE Transactions on Computers, October 2022 *(featured article)*.





Conclusion

- **Silent Data Corruptions**
  - Significant problem at any computing scale
- To detect and provide mitigation, **we need to know**
  - True SDC rates
  - Suspect hardware blocks
  - Vulnerable software pieces
- **Microarchitectural modeling** is an important piece of the puzzle
  - Along with silicon + system measurements
  - Along with finer granularity models

@Barcelona Supercomputing Center

Thank you