Computer Architecture

Lecture 23: Research and Beyond

Prof. Onur Mutlu
ETH Zürich

Fall 2017

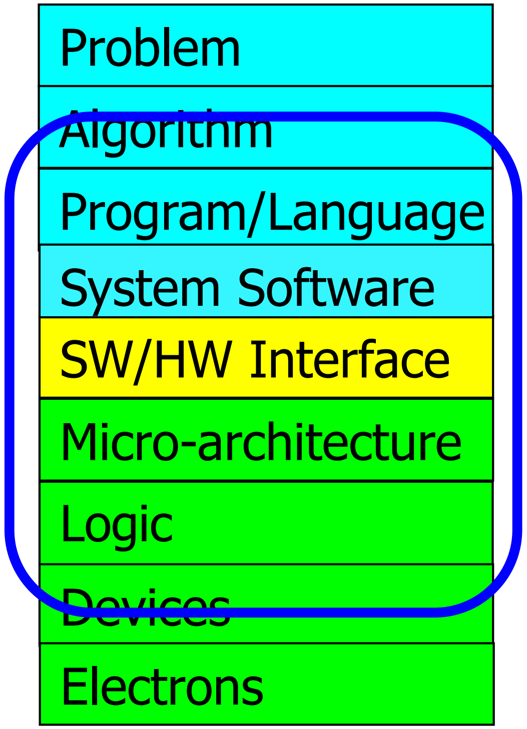
20 December 2017

Research in Computer Architecture

State of the Art

- This is a great time to be a computer architect

- Circuits strained
- Applications strained
- Multiple possible emerging technologies
- Many requirements, many systems



Problem

Algorithm

Program/Language

System Software

SW/HW Interface

Micro-architecture

Logic

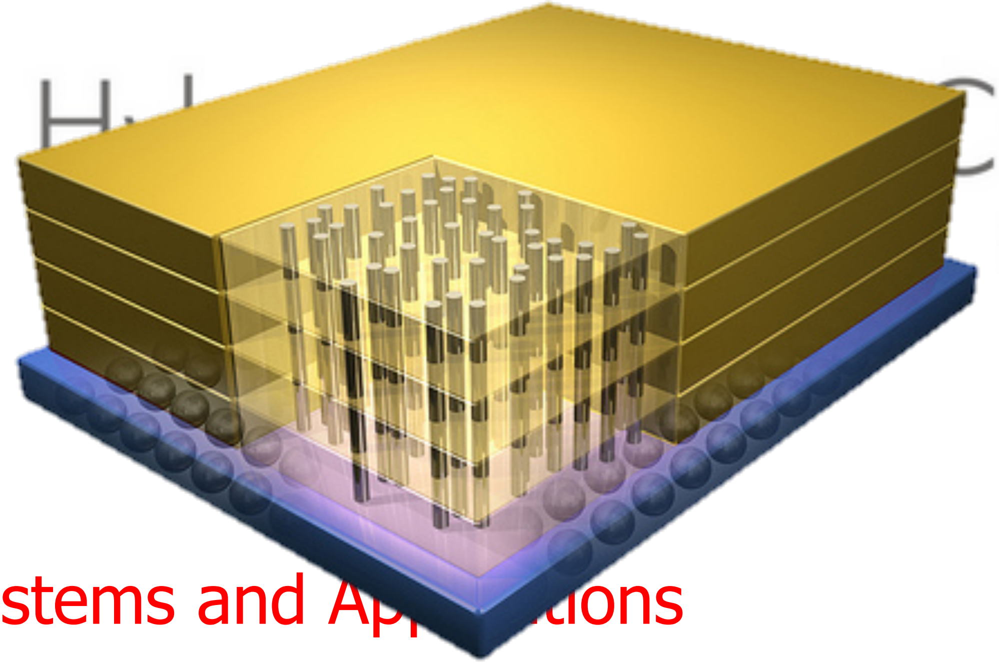
Devices

Electrons

- Many big innovations require computer architecture

Example: Why In-Memory Computation Today?

- Push from Technology

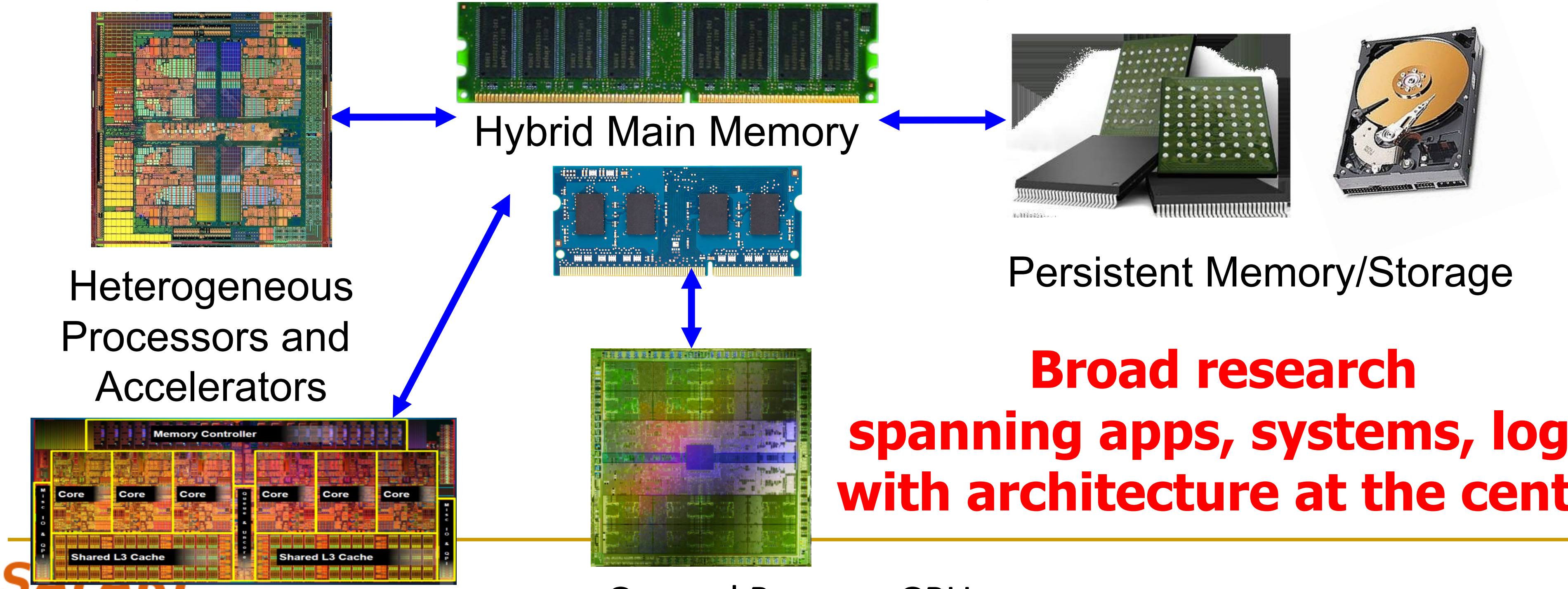


- Pull from Systems and Applications
  - Data access is a major system and application bottleneck
  - Systems are energy limited
  - Data movement much more energy-hungry than computation

Current Research Focus Areas

Research Focus: ***Computer architecture, HW/SW, bioinformatics***

- *Memory and storage (DRAM, flash, emerging), interconnects*
- *Heterogeneous & parallel systems, GPUs, systems for data analytics*
- *System/architecture interaction, new execution models, new interfaces*
- *Energy efficiency, fault tolerance, hardware security, performance*
- *Genome sequence analysis & assembly algorithms and architectures*
- *Biologically inspired systems & system design for bio/medicine*



Hybrid Main Memory

Heterogeneous Processors and Accelerators

Persistent Memory/Storage

General Purpose GPUs

Broad research spanning apps, systems, logic with architecture at the center

Four Key Current Directions

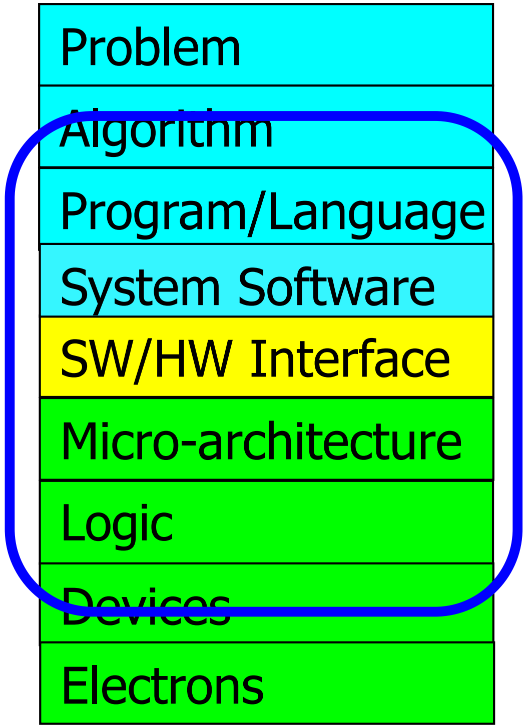
- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency Architectures

- Architectures for Genomics, Medicine, Health

Research Across the Stack



Problem

Algorithm

Program/Language

System Software

SW/HW Interface

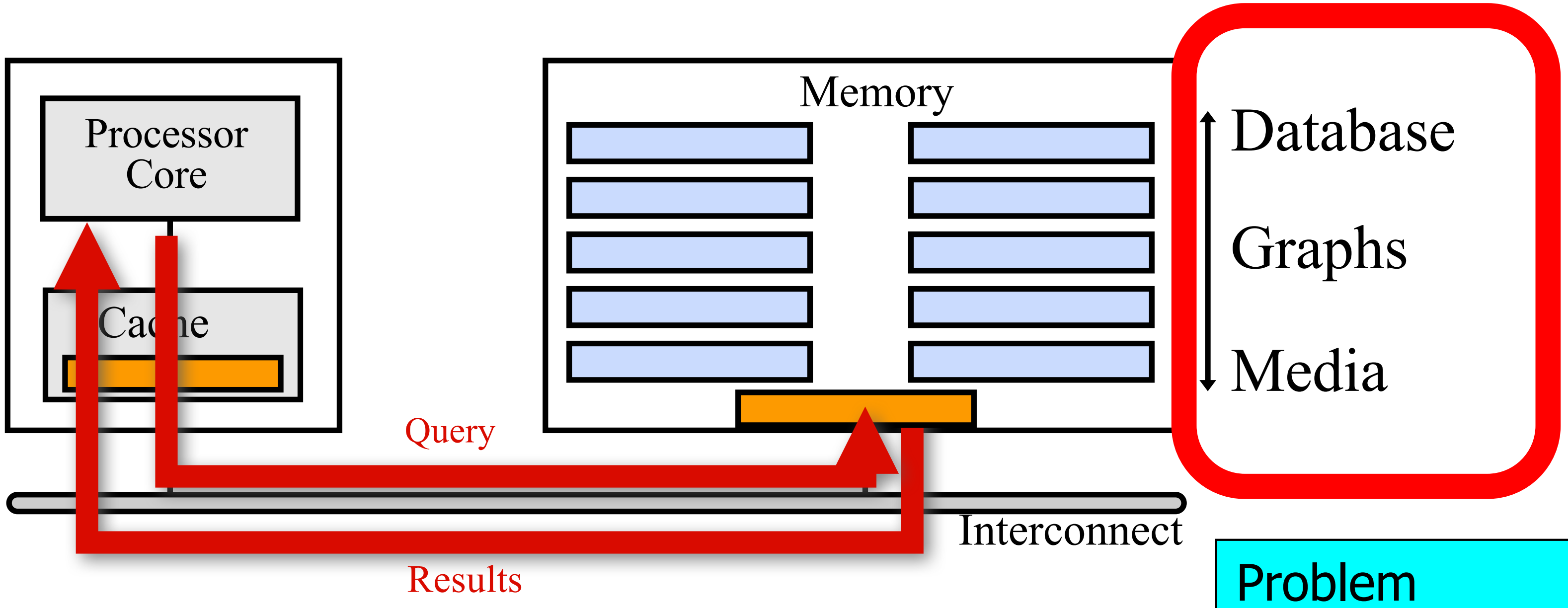
Micro-architecture

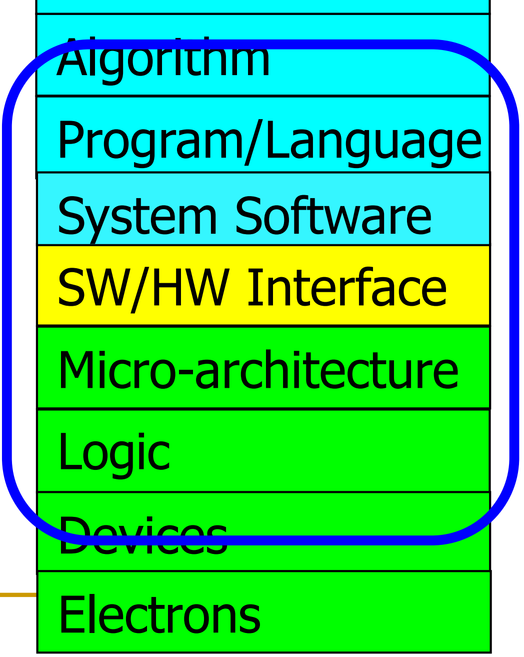
Logic

Devices

Electrons

One Example: Processing Inside Memory





- Many questions … How do we design the:
  - compute-capable memory & controllers?
  - processor chip?
  - software and hardware interfaces?
  - system software and languages?
  - algorithms?

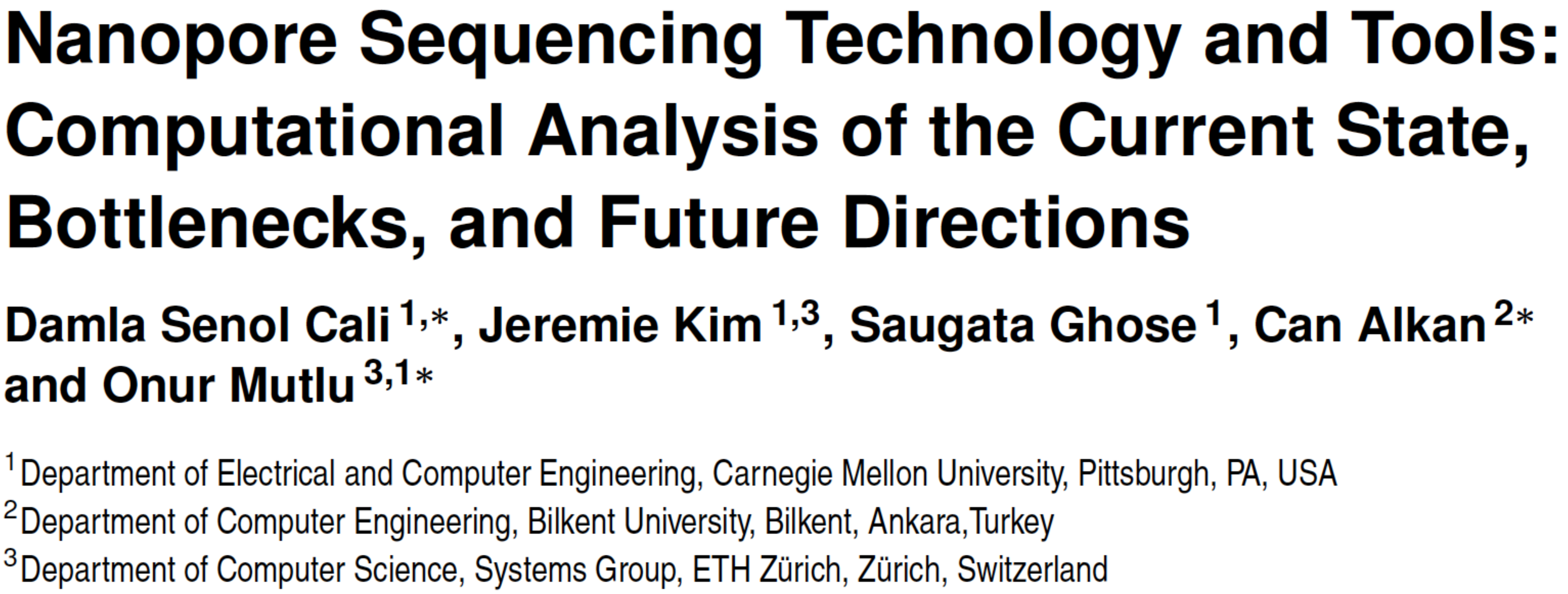
In-Memory DNA Sequence Analysis

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu, **"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"**
  to appear in ***BMC Genomics***, 2018.
  *to also appear in Proceedings of the 16th Asia Pacific Bioinformatics Conference (**APBC**)*, Yokohama, Japan, January 2018.
  arxiv.org Version (pdf)

GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies

Jeremie S. Kim[1,6*], Damla Senol Cali[1], Hongyi Xin[2], Donghyuk Lee[3], Saugata Ghose[1], Mohammed Alser[4], Hasan Hassan[6], Oguz Ergin[5], Can Alkan*[4], and Onur Mutlu*[6,1]

New Genome Sequencing Technologies



Nanopore Sequencing Technology and Tools: Computational Analysis of the Current State, Bottlenecks, and Future Directions

Damla Senol Cali[1,*], Jeremie Kim[1,3], Saugata Ghose[1], Can Alkan[2*] and Onur Mutlu[3,1*]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
[2] Department of Computer Engineering, Bilkent University, Bilkent, Ankara,Turkey
[3] Department of Computer Science, Systems Group, ETH Zürich, Zürich, Switzerland

Some Basics of Research

How To Do Research & Advanced Dev.

- We will talk a lot about this in this course

- Learning by example
  - Reading and evaluating strong and seminal papers & designs

- Learning by doing
  - Semester-long research/design projects, masters' projects, PhD thesis

- Learning by open, critical discussions
  - Paper reading groups, frequent brainstorming and discussions
  - Design sessions
  - Collaborations

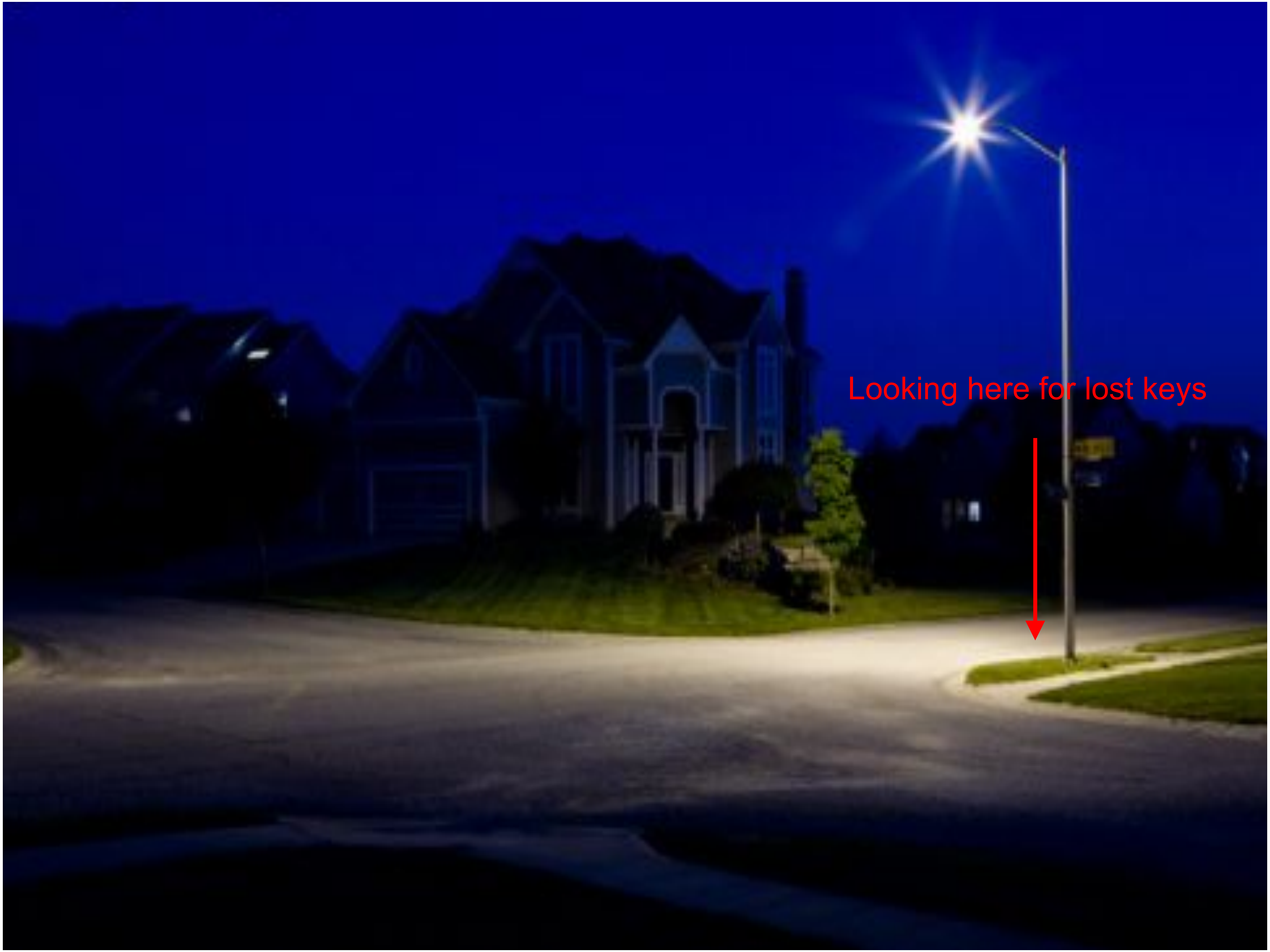
What Is The Goal of Research?

- To generate new insight
    - that can enable what previously did not exist

- Research is a hunt for insight that can eventually impact the world

Some Basic Advice for Good Research

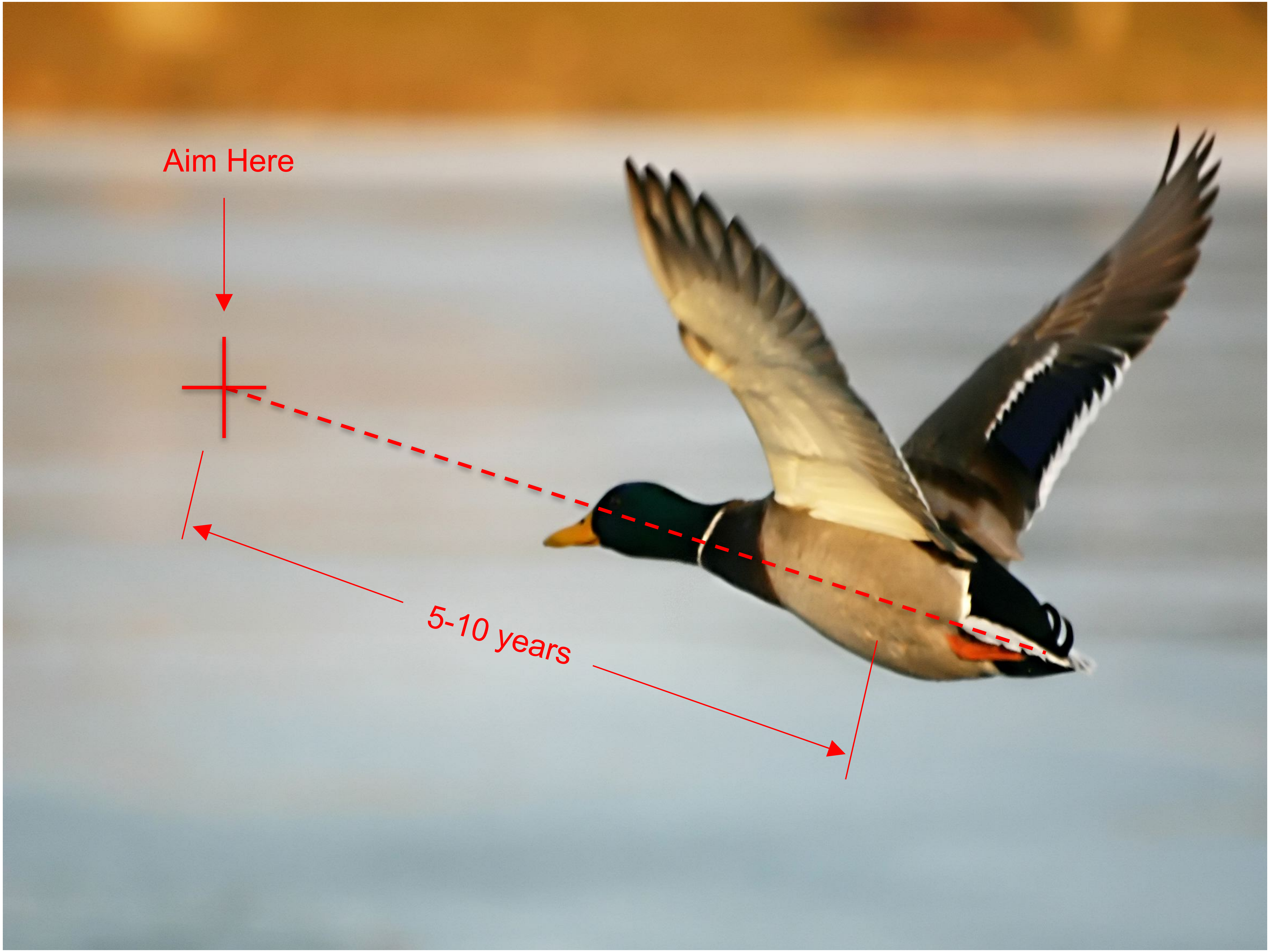
- Choose great problems to solve: Have great taste
  - Difficult
  - Important
  - High impact
- Read heavily and critically
- Think big (out of the box)
  - Do not restrain yourself to tweaks or constraints of today
  - Yet, think about adoption issues
- Aim high
- Write and present extremely well

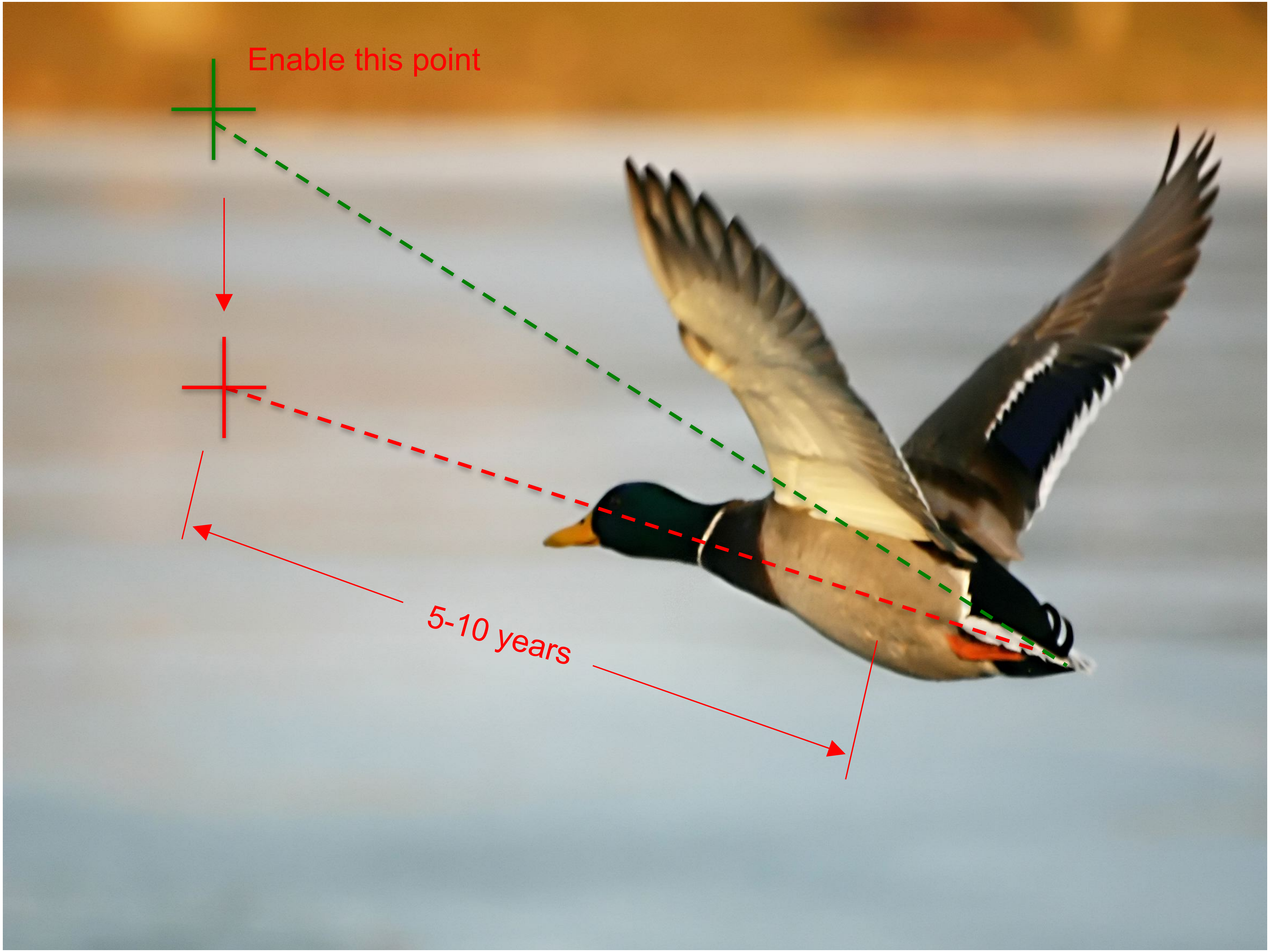


Looking here for lost keys

Lost keys here
Looking here

Current Architecture Practice


Aim Here
5-10 years


Enable this point
5-10 years

The Research Formula

$$ROI = \frac{reward}{risk \times effort}$$

Reward

If you are wildly successful, what difference will it make?

$$ROI = \frac{reward}{risk \times effort}$$

Effort

Learn as much as possible with as little work as possible

$$ROI = \frac{reward}{risk \times effort}$$

Effort

Do the minimum analysis and experimentation necessary to make a point

$$ROI = \frac{reward}{risk \times effort}$$

Research is a
hunt for insight

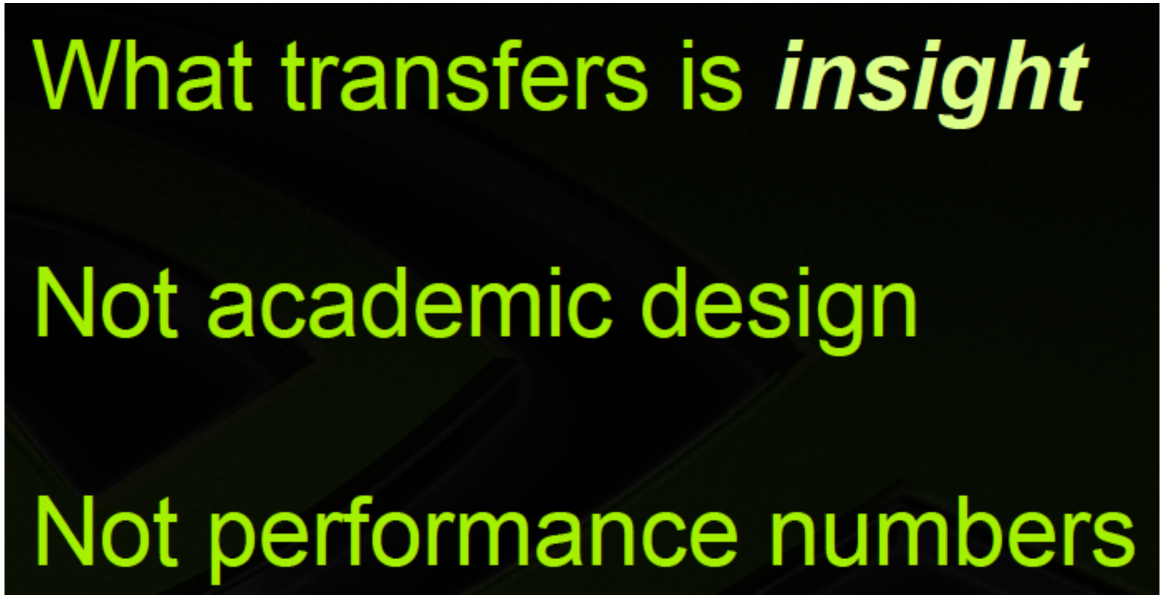
Need to get off the beaten path to find new insights

Recommended Talk

- Bill Dally, Moving the needle: Effective Computer Architecture Research in Academy and Industry

  ISCA 2010 Keynote Talk.

- Acknowledgment: Past few slides are from this talk

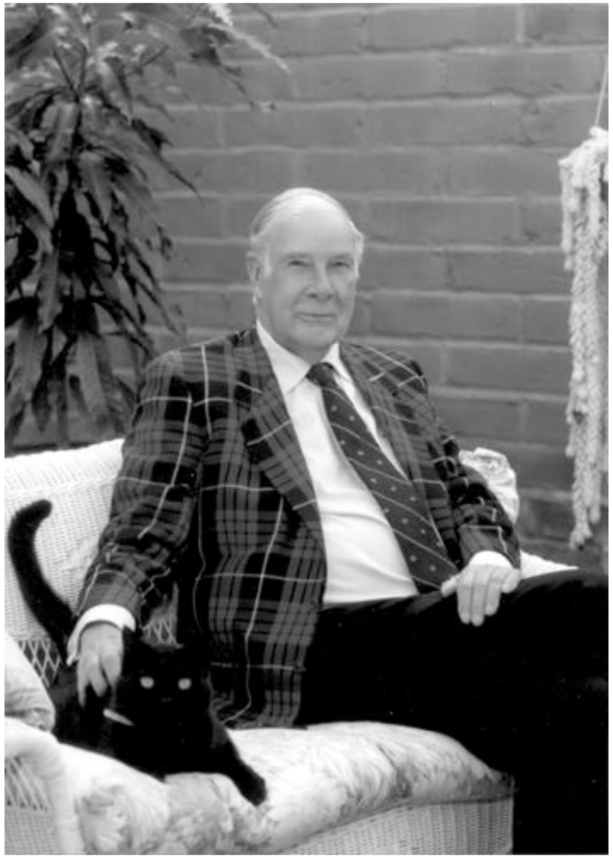


What transfers is ***insight***

Not academic design

Not performance numbers

More Good Advice



“The purpose of computing is insight, not numbers”

Richard Hamming

Some Personal Examples

Questions & Discussion

Computer Architecture

Lecture 23: Research and Beyond

Prof. Onur Mutlu

ETH Zürich

Fall 2017

20 December 2017

We did not cover the following slides in lecture. These are for your benefit.

Personal Journey (I)

- Runahead execution
  - PhD thesis (HPCA 2003, ISCA 2005, MICRO 2005, …)
  - Started with discussions on how to build large windows efficiently → to tolerate memory latencies

- Memory controllers, interference, QoS
  - patent on reordering memory accesses (US Patent filed 2001)
  - started with Memory Perf. Attacks (USENIX Security 2007)
  - later STFM (MICRO 2007), PARBS (ISCA 2008), ATLAS, TCM
  - later: many ways of tackling the problem
    - Source throttling: FST (ASPLOS 2010)
    - Data partitioning: MCP+IMPS (MICRO 2011)
    - Heterogeneous systems: SMS (ISCA 2012), DASH (TACO 2016)
    - More insights into scheduling: BLISS (ICCD 2014, TPDS 2016)
    - Thread scheduling: A2C (HPCA 2013), A-DRM (VEE 2015)

Personal Journey (II)

- Non-volatile memory, persistent memory
  - started with Architecting Phase Change Memory (ISCA 2009)
  - hybrid memories (ICCD 2012, CAL 2012, CLUSTER 2017)
  - crash consistency (MICRO 2015)
  - continues with many issues, including programming and security

- Overcoming the DRAM scaling issues by rethinking DRAM
  - started with RAIDR (ISCA 2012) and SALP (ISCA 2012)
  - Tiered-Latency DRAM (HPCA 2013)
  - built FPGA-based infrastructure to truly understand (ISCA 2013)
  - Rowhammer (ISCA 2014)
  - Latency issues (HPCA 2013, HPCA 2015, SIGMETRICS 2016, …)
  - continues with all aspects of DRAM design and use (HW/SW)

Personal Journey (IV)

- Processing in memory
  - early thoughts: intelligent memory controller do anything
  - started with RowClone (MICRO 2013)
  - Ambit (MICRO 2017)
  - Tesseract and PEI (both in ISCA 2015)
  - Enhanced Memory Controller (ISCA 2016)
  - PIM for Google Consumer Workloads (ASPLOS 2018)
  - Many issues and systems…

- Memory compression
  - started with BDI (PACT 2012)
  - linearly-compressed pages (MICRO 2013)
  - many other issues…

Personal Journey (V)

- Bufferless deflection-based networks
  - started with BLESS (ISCA 2009)
  - aimed to make it implementable and high performance
    - CHIPPER (HPCA 2011), MinBD (NOCS 2012), …
  - hierarchical rings (SBAC-PAD 2014, PARCO 2016)
  - throttling mechanisms (HotNets 2010, SIGCOMM 2012, ...)
  - ...

- QoS-aware on-chip interconnects
  - started with STC (MICRO 2009) and PVC (MICRO 2009)
  - Aergia (ISCA 2010)
  - Kilo NoC (ISCA 2012)
  - …

Personal Journey (VI)

- Online self-test for fault tolerance and bug tolerance
  - Started with introspection (DSN 2015) and ACE (MICRO 2007)
  - Bug detection (MICRO 2008)
  - OS scheduling for online self test (ICCAD 2009)
  - Online self test for uncore (VTS 2010)
  - many works on doing this for DRAM/memory now…
- GPUs
  - started with 2L-scheduling and large warp uarch (MICRO 2011)
  - other warp scheduling mechanisms (ASPLOS 2013)
  - Scheduling and memory interactions (ISCA 2013)
  - Assist warps (ISCA 2015)
  - MeDiC: handling memory divergence (PACT 2015)
  - …

Personal Journey (VII)

- Genome sequence analysis
  - Started with discussions on next-generation sequencing
  - First work to speed up a comprehensive read mapper (Nature Genetics 2009)
  - First pre-alignment/filtering mechanism (BMC Genomics 2013)
  - SIMD-friendly Filtering (Bioinformatics 2015)
  - Gatekeeper: FPGA acceleration of filtering (Bioinformatics 2017)
  - GRIM-filter: in-memory filtering (BMC Genomics 2018)

Personal Journey (VIII)

- Caching and Prefetching
  - Caching issues in runahead (various works in 2003-2006)
  - MLP-aware cache replacement (ISCA 2006)
  - Feedback directed prefetching (HPCA 2007)
  - Evicted address filter (PACT 2012)
  - Dirty block index (ISCA 2014)
  - …
- Branch handling
  - Wrong path events (MICRO 2004)
  - Wish branches (MICRO 2005)
  - Dynamic predication (MICRO 2006), indirect jmps (ASPLOS 2008)
  - VPC Prediction (ISCA 2007)
  - …

Personal Journey (IX)

- Flash Memory
  - started with Error Patterns in Flash Memory (DATE 2012)
  - flash correct and refresh (ICCD 2012) (invited Intel Tech J 2013)
  - many works on characterization and modeling and prediction for bit error reduction and lifetime improvement
    - threshold voltage modeling (DATE 2013)
    - read reference voltage prediction (ICCD 2013)
    - neighbor assisted correction (SIGMETRICS 2014)
    - data retention (HPCA 2015)
    - read disturb, program interference (DSN 2015, HPCA 2017)
    - WARM (MSST 2015)
    - accurate online read reference voltage prediction (JSAC 2016)
  - 3D NAND characterization, modeling and prediction (HPCA 2018)
  - Summary in (Proc IEEE 2017, Book Chapter 2018)

Personal Journey (X)

- Infrastructure
  - Ramulator (CAL 2015) – Result of 10+ year effort now
  - SoftMC (HPCA 2017) – Result of 5+ year effort now
  - NAND Flash Infrastructure – Result of 6+ year effort now
  - NoCulator (PARCO 2016) – Result of 9+ year effort now
  - MemSchedSim, ASMSim (MICRO 2015) – 10+ year effort now
  - Many simulators over time…
    - Implemented Runahead Execution on at least 4 simulators
  - Many different methods of evaluation over time
    - Testing and analysis of real systems
    - Very high level simulation to explore big tradeoffs
    - Models in-between
  - https://github.com/CMU-SAFARI
  - http://www.ece.cmu.edu/~safari/tools.html

Understanding DRAM Scaling Issues

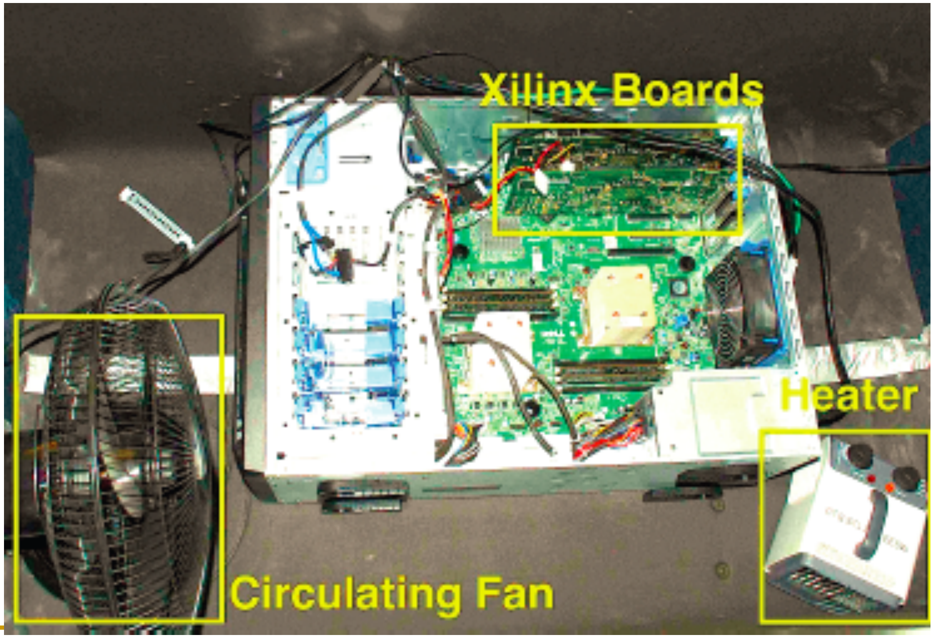
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

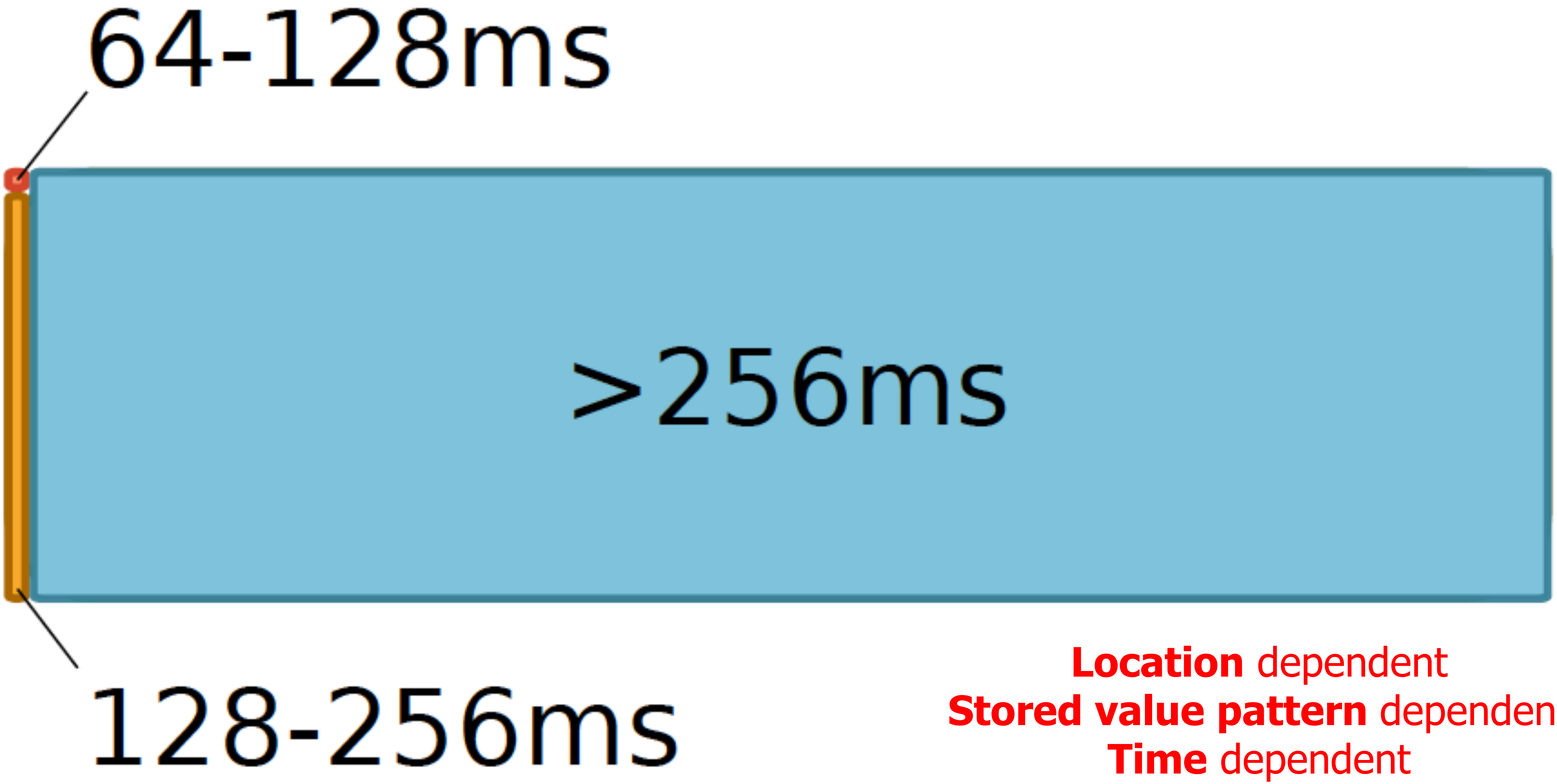
Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)



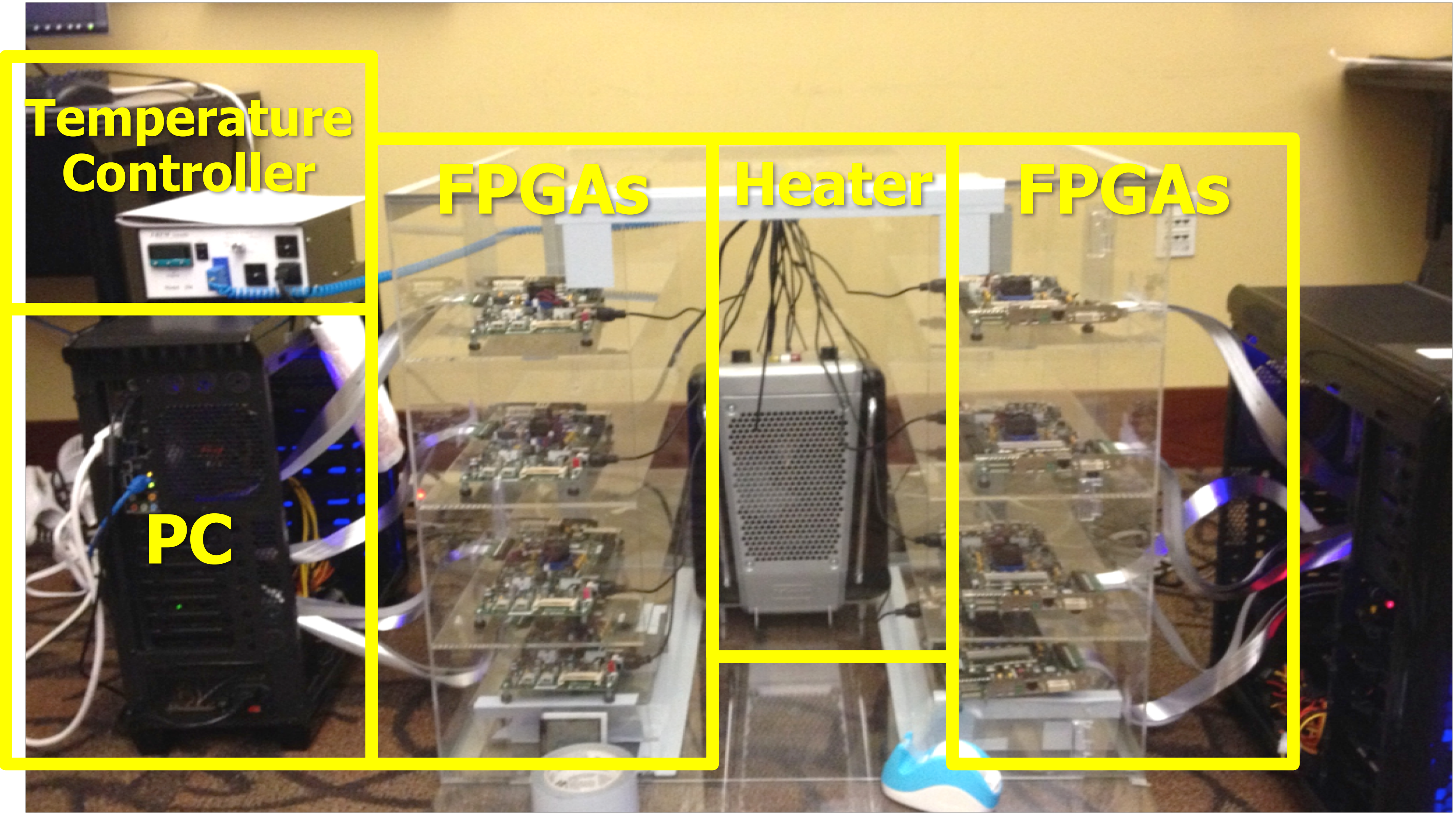
Data Retention in Memory [Liu et al., ISCA 2013]

- Retention Time Profile of DRAM looks like this (RAIDR, ISCA 2012):



Location dependent
Stored value pattern dependent
Time dependent

Understanding DRAM Scaling Issues



SAFARI

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

A Curious Discovery **[Kim et al., ISCA 2014]**

One can

predictably induce errors

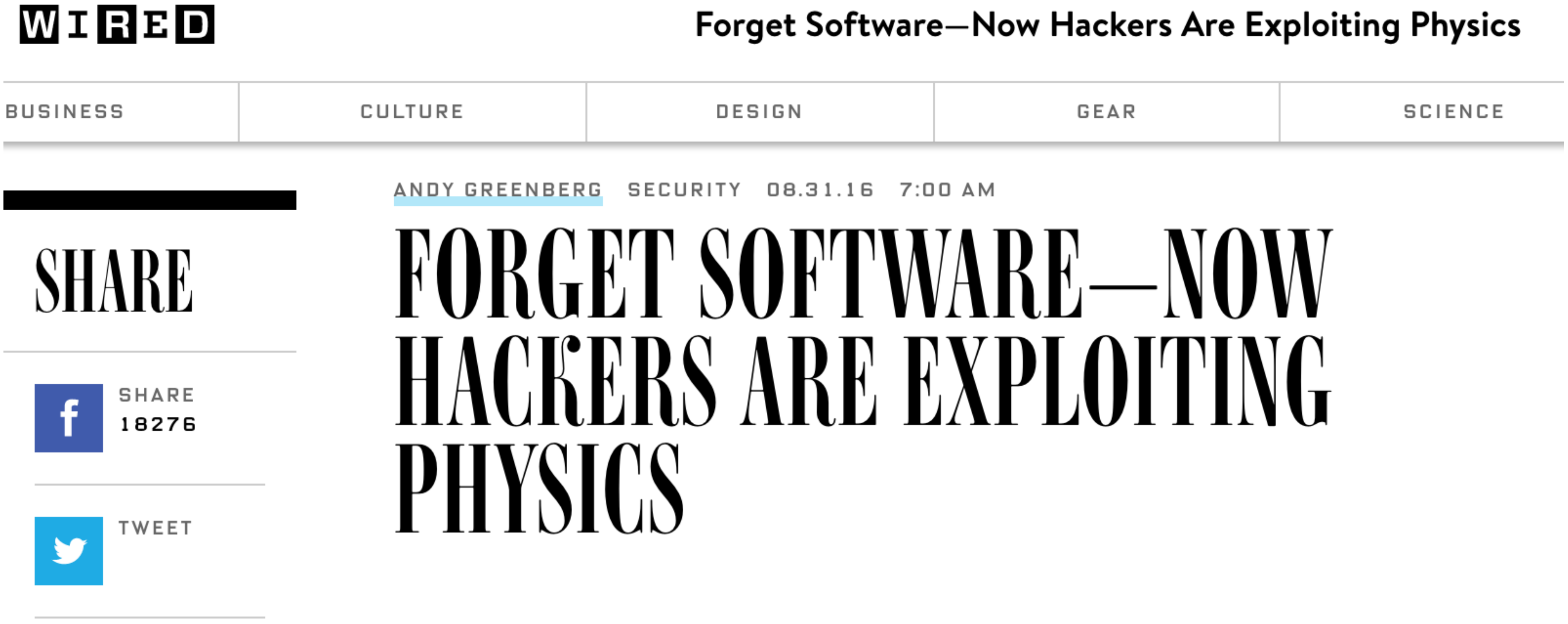
in most DRAM memory chips

DRAM RowHammer

A simple hardware failure mechanism

can create a widespread

system security vulnerability



WIRED

Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

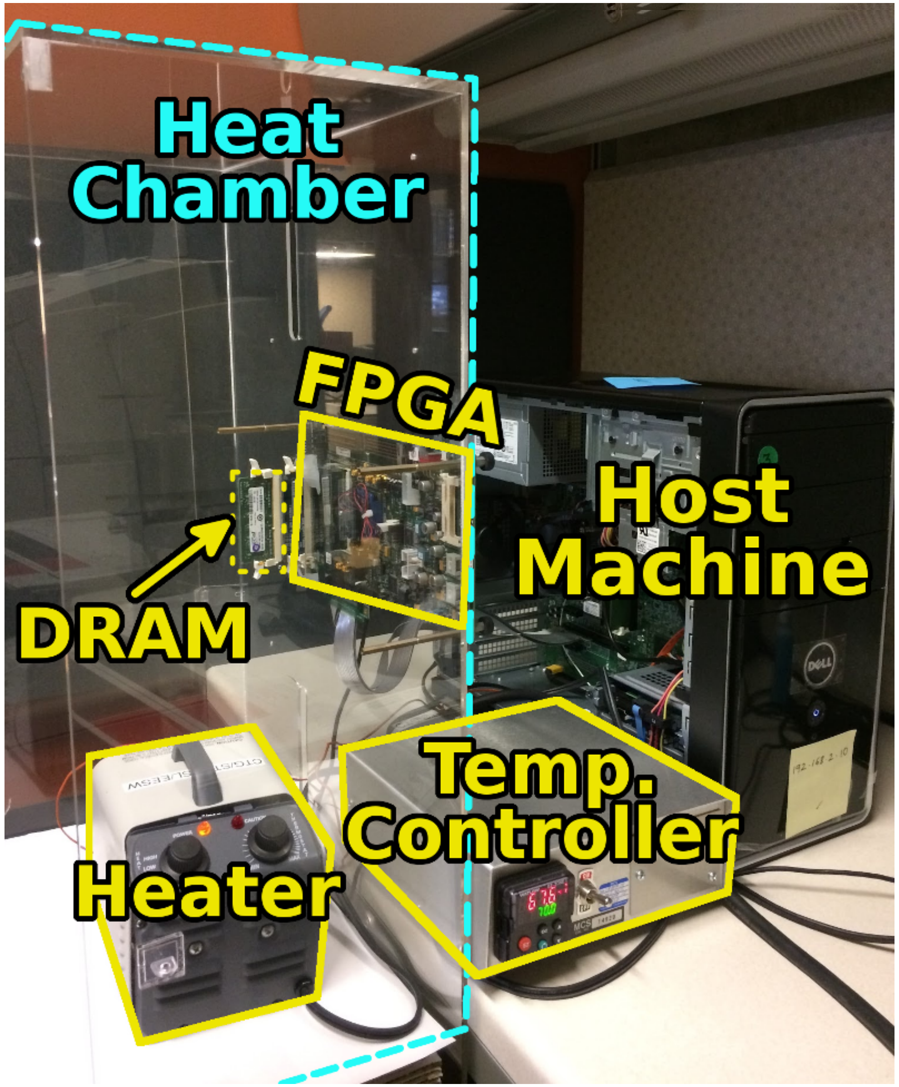
FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., "**SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies**," HPCA 2017.

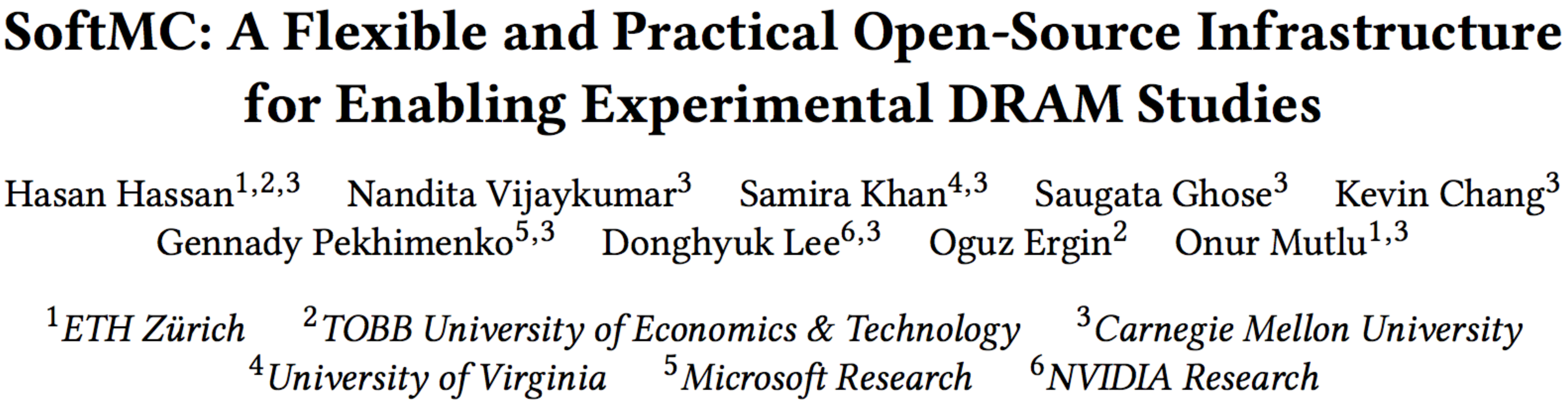
- **Flexible**
- **Easy to Use (C++ API)**
- **Open-source**

  github.com/CMU-SAFARI/SoftMC



SoftMC

- https://github.com/CMU-SAFARI/SoftMC

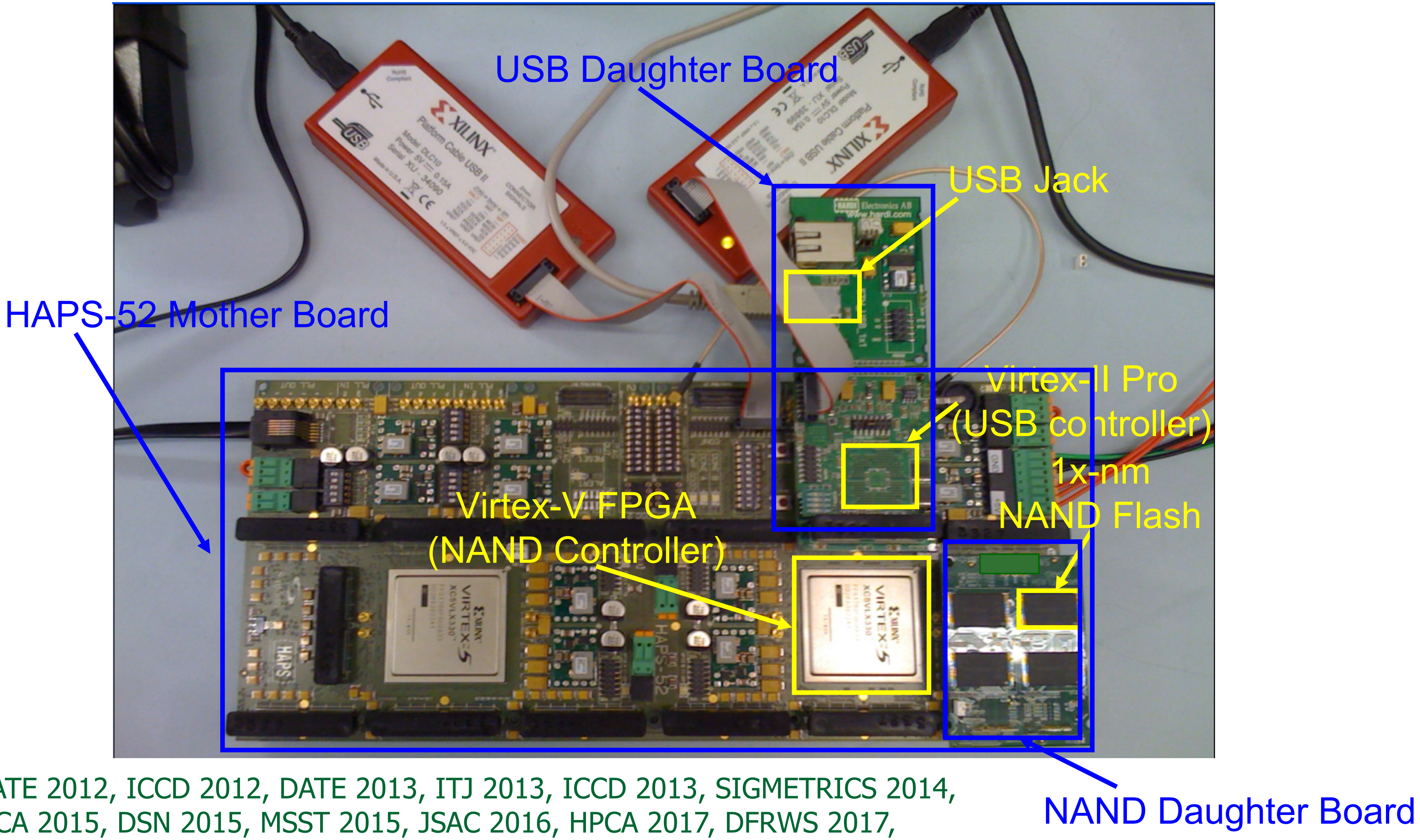


SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan[1,2,3] Nandita Vijaykumar[3] Samira Khan[4,3] Saugata Ghose[3] Kevin Chang[3]
Gennady Pekhimenko[5,3] Donghyuk Lee[6,3] Oguz Ergin[2] Onur Mutlu[1,3]

[1] *ETH Zürich* [2] *TOBB University of Economics & Technology* [3] *Carnegie Mellon University*
[4] *University of Virginia* [5] *Microsoft Research* [6] *NVIDIA Research*

Understanding NAND Flash Scaling Issues

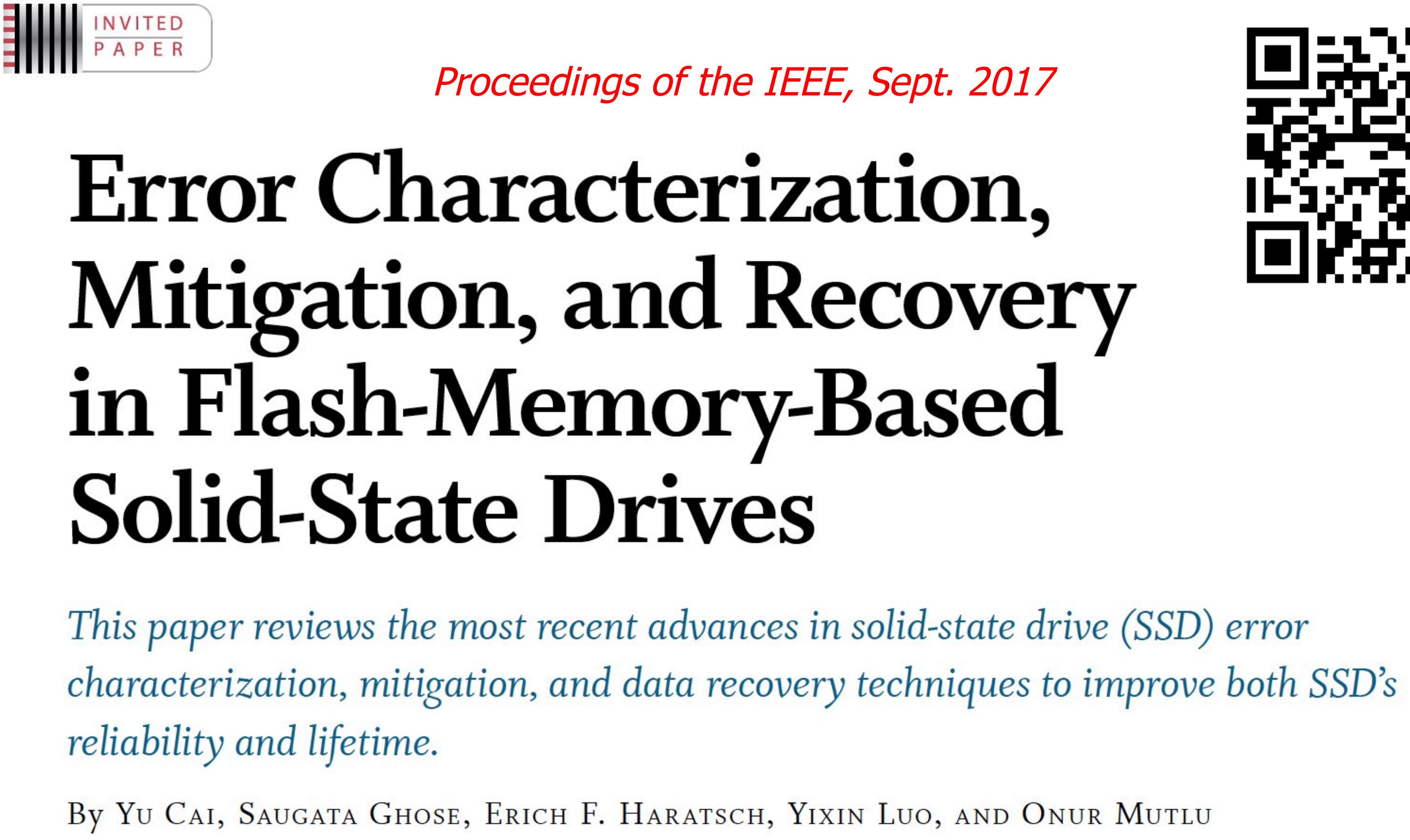


[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018]

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.

Understanding NAND Flash Scaling Issues

Proceedings of the IEEE, Sept. 2017



INVITED PAPER

Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642