P&S SoftMC

Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments

Hasan Hassan
Prof. Onur Mutlu
ETH Zürich
Spring 2022
8 March 2022
P&S SoftMC: Content

- We will learn in detail how modern DDR4 DRAM operates.

- You will learn how to characterize DRAM using an FPGA-based DRAM characterization infrastructure (SoftMC).

- You will use SoftMC to develop your own DRAM experiments and gain hand-on experience in studying DRAM characteristics.
P&S SoftMC: Key Takeaways

- This P&S is aimed at improving your
  - Knowledge in Computer Architecture and Memory Systems
  - Technical skills in running DRAM experiments using real devices
  - Critical thinking and analysis
  - Interaction with a nice group of researchers
  - Familiarity with key research directions
  - Technical presentation of your project
P&S SoftMC: Key Goal

(Learn how to) study real memory devices using an FPGA-based DRAM infrastructure to gain new insights on DRAM behavior
Prerequisites of the Course

- Digital Design and Computer Architecture (or equivalent course)
- Familiarity with FPGA programming
- Interest in low-level hacking and memory
- Interest in discovering why things do or do not work and solving problems
Course Info: Who Are We? (I)

- Onur Mutlu
  - Full Professor @ ETH Zurich ITET (INFK), since September 2015
  - Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-
  - PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
  - https://people.inf.ethz.ch/omutlu/
  - omutlu@gmail.com (Best way to reach me)
  - https://people.inf.ethz.ch/omutlu/projects.htm

- Research and Teaching in:
  - Computer architecture, computer systems, hardware security, bioinformatics
  - Memory and storage systems
  - Hardware security, safety, predictability
  - Fault tolerance
  - Hardware/software cooperation
  - Architectures for bioinformatics, health, medicine
  - ...
Course Info: Who Are We? (II)

- **Lead Supervisor:**
  - Hasan Hassan

- **Supervisors:**
  - Giray Yaglikci
  - Ataberk Olgun
  - Haocong Luo
  - Minesh Patel
  - Jeremie Kim

- **Get to know us and our research**
  - [https://safari.ethz.ch/group-members](https://safari.ethz.ch/group-members)
Onur Mutlu’s SAFARI Research Group

**Computer architecture, HW/SW, systems, bioinformatics, security, memory**

https://safari.ethz.ch/safari-newsletter-april-2020/

Think BIG, Aim HIGH!

https://safari.ethz.ch
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

**Broad research spanning apps, systems, logic with architecture at the center**
Course Info: How About You?

- Let us know your background, interests
- Why did you join this P&S?
- Please submit HW0
Course Requirements and Expectations

- Attendance required for all meetings
- Study the learning materials
- Each student will carry out a hands-on project
  - Build, implement, code, and design with close engagement from the supervisors
- Participation
  - Ask questions, contribute thoughts/ideas
  - Read relevant papers

We will help in all projects!
If your work is really good, you may get it published!
Course Website

- [https://safari.ethz.ch/projects_and_seminars/doku.php?id=softmc](https://safari.ethz.ch/projects_and_seminars/doku.php?id=softmc)

- Useful information about the course

- Check your email frequently for announcements
Meeting 1

- **Required materials:**
  
  SoftMC Tutorial Video:  [https://youtu.be/909uTQu0lbA](https://youtu.be/909uTQu0lbA)

  SoftMC lecture: [https://www.youtube.com/watch?v=tnSPEP3t-Ys](https://www.youtube.com/watch?v=tnSPEP3t-Ys)


  - **Recommended materials:**
  


Meeting 2 (TBD)

- We will announce the projects and will give you some description about them.

- You will have a week to submit your project preferences.

- The supervisors would like to help you with selecting a project that matches your interests, skills, and background.

- It is important that you study the learning materials before our next meeting!
Tentative Weekly Schedule

- **Week 1** – Logistics & Intro to DRAM and SoftMC [HPCA’17]
- **Week 2** – Revisiting RowHammer [ISCA’20] | Available Projects
- **Week 3** – Deeper Look Into RowHammer’s Sensitivities [MICRO’21] | PU
- **Week 5** – QUAC-TRNG [ISCA’21] | PU
- **Week 6** – The Reach Profiler (REAPER) [ISCA’17] | PU
- **Week 7** – PiDRAM [arXiv’21] | PU
- **Week 8+** - PU

- **PU** = Project Updates

- **How about meeting every Tuesday at 11:00?**
Performance Assessment

We expect you to:

- **Learn** how DRAM operates and how to perform DRAM characterization using FPGAs

- **Achieve the goals** of your project

- **Deliver** your code and results with sufficient documentation

- Prepare a **final presentation** and present your work to SAFARI
An Introduction to DRAM and SoftMC
SoftMC
A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, Onur Mutlu

HPCA 2017
Executive Summary

- Two critical problems of DRAM: **Reliability** and **Performance**
  - Recently-discovered bug: *RowHammer*

- **Characterize, analyze, and understand** DRAM cell behavior

- We design and implement **SoftMC**, an FPGA-based DRAM testing infrastructure
  - Flexible and Easy to Use (C++ API)
  - Open-source ([github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC))

- We implement two use cases
  - A retention time distribution test
  - An experiment to validate two **latency reduction** mechanisms

- **SoftMC enables a wide range of studies**
1. DRAM Basics & Motivation

2. SoftMC

3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas

4. Future Research Directions

5. Conclusion
DRAM Operations
DRAM Latency

Retention Time: The interval during which the data is retained correctly in the DRAM cell without accessing it.
Latency vs. Reliability

Violating latencies negatively affects DRAM reliability
Other Factors Affecting Reliability and Latency

- Temperature
- Voltage
- Inter-cell Interference
- Manufacturing Process
- Retention Time
- ...

To develop new mechanisms improving reliability and latency, we need to better understand the effects of these factors.
Characterizing DRAM

Many of the factors affecting DRAM reliability and latency cannot be properly modeled.

We need to perform experimental studies of real DRAM chips.
Outline

1. DRAM Basics & Motivation
2. SoftMC
3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas
4. Future Research Directions
5. Conclusion
Goals of a DRAM Testing Infrastructure

- **Flexibility**
  - Ability to test *any* DRAM operation
  - Ability to test *any* combination of DRAM operations and *custom* timing parameters

- **Ease of use**
  - Simple programming interface (C++)
  - Minimal programming effort and time
  - Accessible to a wide range of users
    - who may lack experience in hardware design
SoftMC: High-level View

FPGA-based memory characterization infrastructure

Prototype using *Xilinx ML605*

Easily programmable using the C++ API
SoftMC: Key Components

1. SoftMC API
2. PCIe Driver
3. SoftMC Hardware
Writing data to DRAM:

```cpp
InstructionSequence iseq;
iseq.insert(genACT(bank, row));
iseq.insert(genWAIT(tRCD));
iseq.insert(genWR(bank, col, data));
iseq.insert(genWAIT(tCL + tBL + tWR));
iseq.insert(genPRE(bank));
iseq.insert(genWAIT(tRP));
iseq.insert(genEND());
iseq.execute(fpga);
```
SoftMC API (New)

- **SoftMCPlatform:**
  - `execute()` – starts execution of a SoftMC program
  - `receiveData()` – gets data from PCIe

- **Program:**
  - `add_inst()` – adds an instruction
  - `add_branch()` – adds a branch instruction
  - `add_label()` – adds a branch target

- **SoftMC Instructions:**
  - Arithmetic & Logic: AND, OR, XOR, ADD, SUB, LI, MV, ...
  - Scratchpad Memory: LD, ST
  - DRAM Commands: ACT, PRE, READ, WRITE, REF, ...
SoftMC: Key Components

1. SoftMC API

2. PCIe Driver*
   - Communicates raw data with the FPGA

3. SoftMC Hardware

SoftMC Hardware (Old)

- PCIe Controller
- Instruction Receiver
- Instruction Queue
- Instruction Dispatcher
- DDR PHY
- Auto-refresh Controller
- Calibration Controller
- Read Capture
- Host Machine
- Instructions

SoftMC Hardware (FPGA)

Wait (Read Access Latency)
SoftMC Hardware (New)

PCIe Controller

Simple Processor

Program

Instruction Dispatcher

Auto-refresh Controller

Calibration Controller

Read Capture

SoftMC Hardware (FPGA)

Host Machine

SoftMC Program

Instructions

DRAM

DDR PHY
Outline

1. DRAM Basics & Motivation
2. SoftMC
3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas
4. Future Research Directions
5. Conclusion
Retention Time Distribution Study

Can be implemented with just ~100 lines of code

```java
InstructionSequence iseq;
iseq.insert(genACT(bank, row));
iseq.insert(genWAIT(tRCD));
for(int col = 0; col < COLUMNS; col++){
    iseq.insert(genWR(bank, col, data));
    iseq.insert(genWAIT(tBL));
}
iseq.insert(genWAIT(tCL + tWR));
iseq.insert(genPRE(bank));
iseq.insert(genWAIT(tRP));
iseq.insert(genEND());
iseq.execute(fpga));
```
Retention Time Test: Results

Validates the correctness of the SoftMC Infrastructure

Number of Erroneous Bytes vs. Refresh Interval (s)

- Module A
- Module B
- Module C

@ ~20°C (room temperature)
Outline

1. DRAM Basics & Motivation
2. SoftMC
3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas
4. Future Research Directions
5. Conclusion
Accessing Highly-charged Cells Faster

**NUAT**  
(Shin+, HPCA 2014)

**ChargeCache**  
(Hassan+, HPCA 2016)

A highly-charged cell can be accessed with low latency
How a Highly-Charged Cell Is Accessed Faster?

- **Activate DRAM Cell**
- **Sense Amplifier**

**Latencies**:
- **Precharge Latency**
- **Activation Latency**
- **Ready-to-access Latency**

**Timelines**:
- **0 (refresh)**: 64 ms

**Actions**:
- **Activate**
- **Read**
- **Precharge**

**Related Concepts**:
- **Precharge**
- **Activate**
Ready-to-access Latency Test

Longer wait  ➔  Lower cell charge
Shorter wait  ➔  Higher cell charge

Write Reference Data
Wait for the \textit{Wait Interval}
Read Back
Observe Errors
Change the \textit{Wait Interval}

With \textbf{custom} ready-to-access latency parameter

\textit{Can be implemented with just \sim 150 lines of code}
Ready-to-access Latency: Results

We do not observe the expected latency reduction effect in existing DRAM chips.

Expected Curves

Latency (cycles)

Number of Erroneous Bytes

Refresh Interval

Latency: Results

42
Why Don’t We See the Latency Reduction Effect?

- The memory controller **cannot externally control** when a sense amplifier gets enabled in **existing DRAM chips**

![Diagram showing the concept of latency reduction and fixed latency](image)

**Ready to Access**

**Sense Amplifier**

**Fixed Latency!**
Outline

1. DRAM Basics & Motivation
2. SoftMC
3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas
4. Future Research Directions
5. Conclusion
Future Research Directions

- More Characterization of DRAM
  - How are the cell characteristics changing with different generations of technology nodes?
  - What types of usage accelerate aging?
- Characterization of Non-volatile Memory
- Extensions
  - Memory Scheduling
  - Workload Analysis
  - Testbed for in-memory Computation
Outline

1. DRAM Basics & Motivation
2. SoftMC
3. Use Cases
   - Retention Time Distribution Study
   - Evaluating Recently-Proposed Ideas
4. Future Research Directions
5. Conclusion
Conclusion

- **SoftMC**: First publicly-available FPGA-based DRAM testing infrastructure
- **Flexible and Easy to Use**
- Implemented two use cases
  - Retention Time Distribution Study
  - Evaluation of two recently-proposed latency reduction mechanisms
- SoftMC can enable many other studies, ideas, and methodologies in the design of future memory systems
- **Download** our first prototype

github.com/CMU-SAFARI/SoftMC
SoftMC
A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, Onur Mutlu

HPCA 2017
P&S SoftMC

Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments

Hasan Hassan
Prof. Onur Mutlu
ETH Zürich
Spring 2022
8 March 2022