

# SoftMC

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

*Presented at HPCA 2017*

Hasan Hassan,

Nandita Vijaykumar, Samira Khan,

Saugata Ghose, Kevin Chang,

Gennady Pekhimenko, Donghyuk Lee,

Oguz Ergin, Onur Mutlu

**ETH** zürich

**Carnegie Mellon**



TOBB  
UNIVERSITY OF  
ECONOMICS AND TECHNOLOGY

# 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**

# Outline

## 1. DRAM Basics & Motivation

## 2. SoftMC

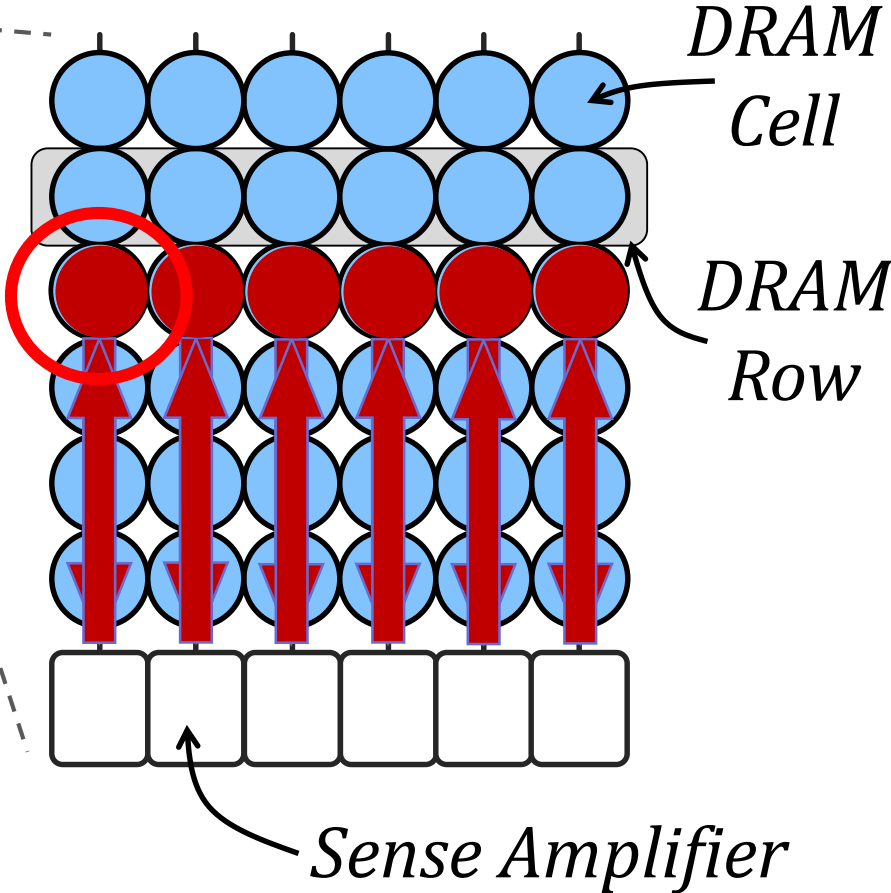
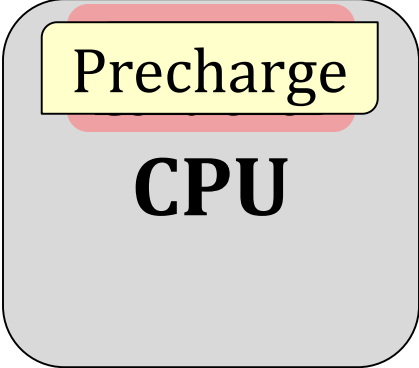
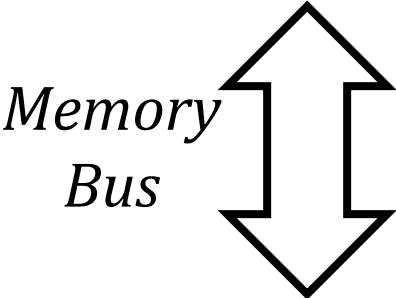
## 3. Use Cases

- Retention Time Distribution Study
- Evaluating Recently-Proposed Ideas

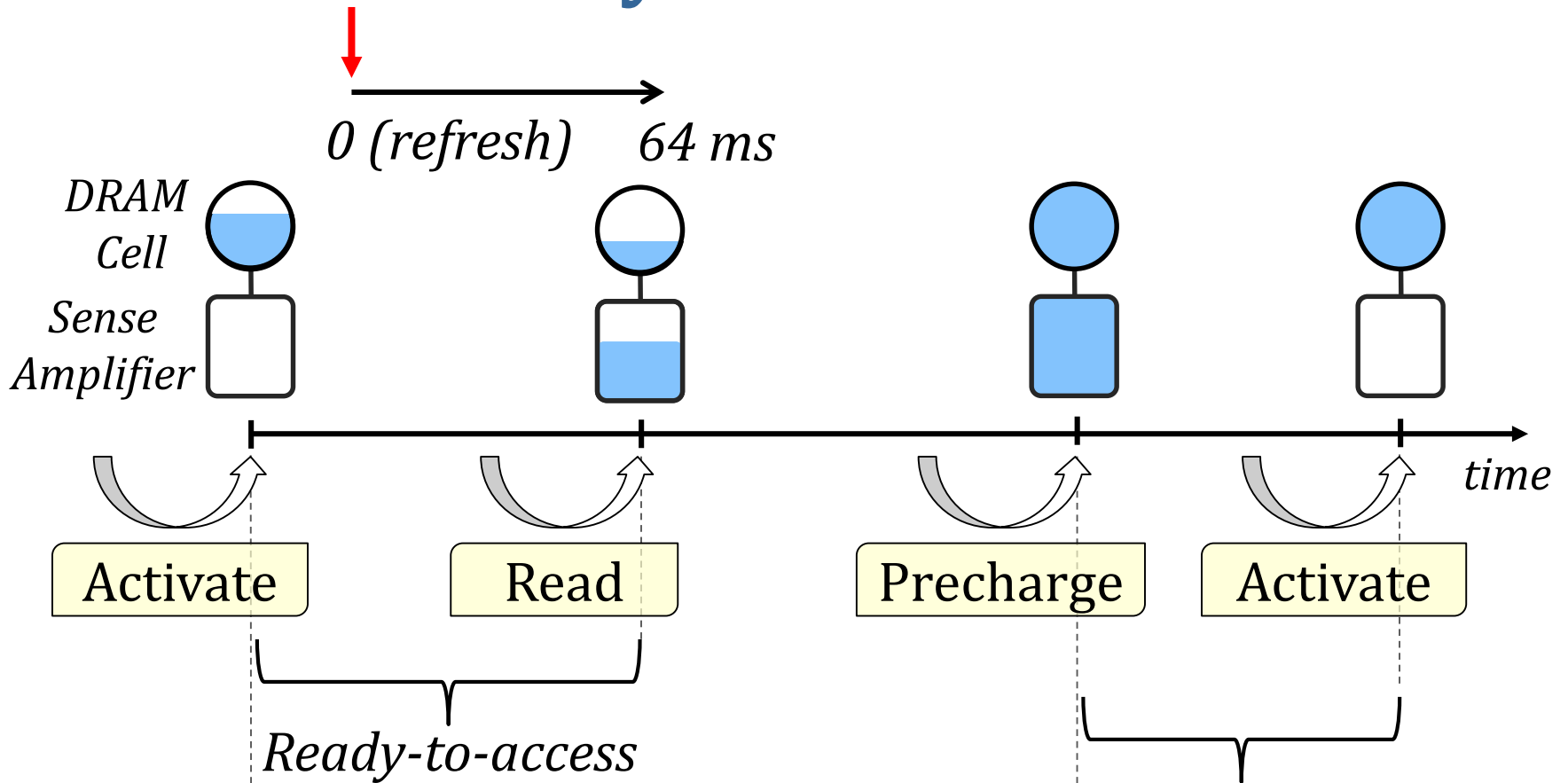
## 4. Future Research Directions

## 5. Conclusion

# DRAM Operations



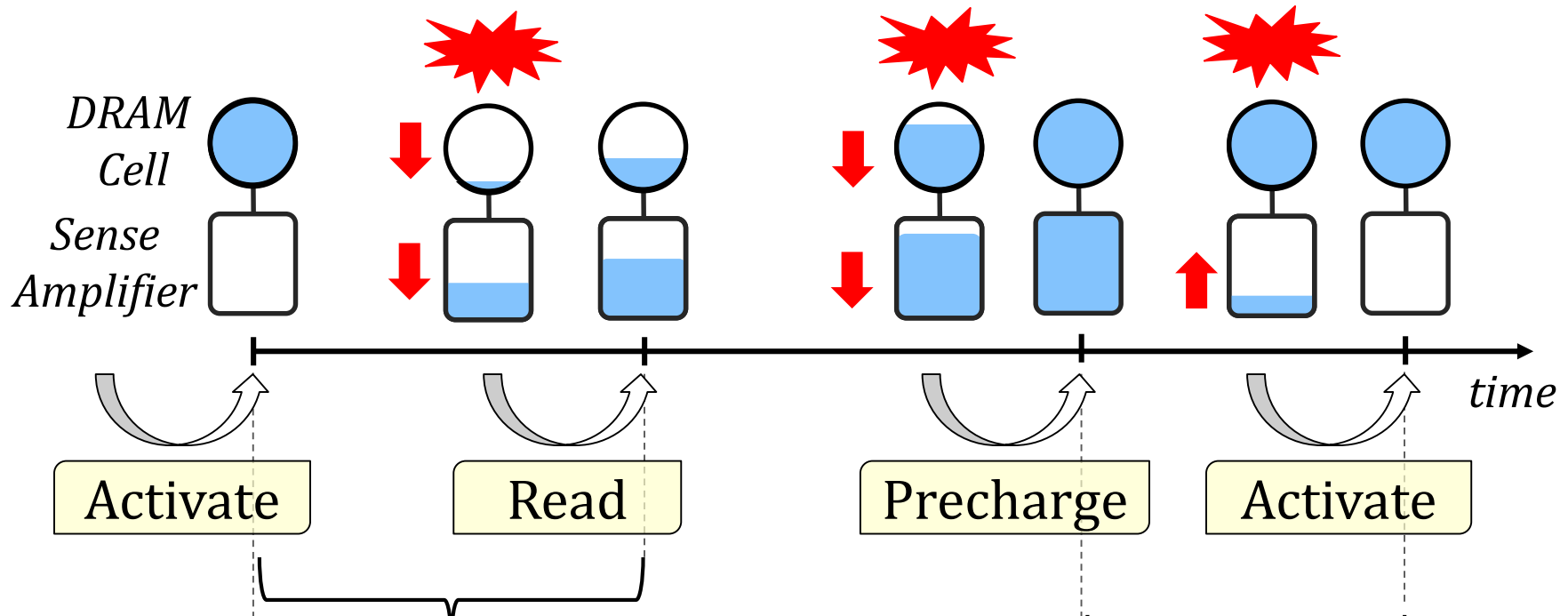
# DRAM Latency



**Retention Time:** The interval during which the data is retained correctly in the DRAM cell without accessing it

**Latency:** The interval during which the data is not available for access

# Latency vs. Reliability



**Violating latencies negatively affects DRAM reliability**

## Other Factors Affecting Reliability and Latency

- Temperature
- Voltage
- Inter-cell Interference

**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

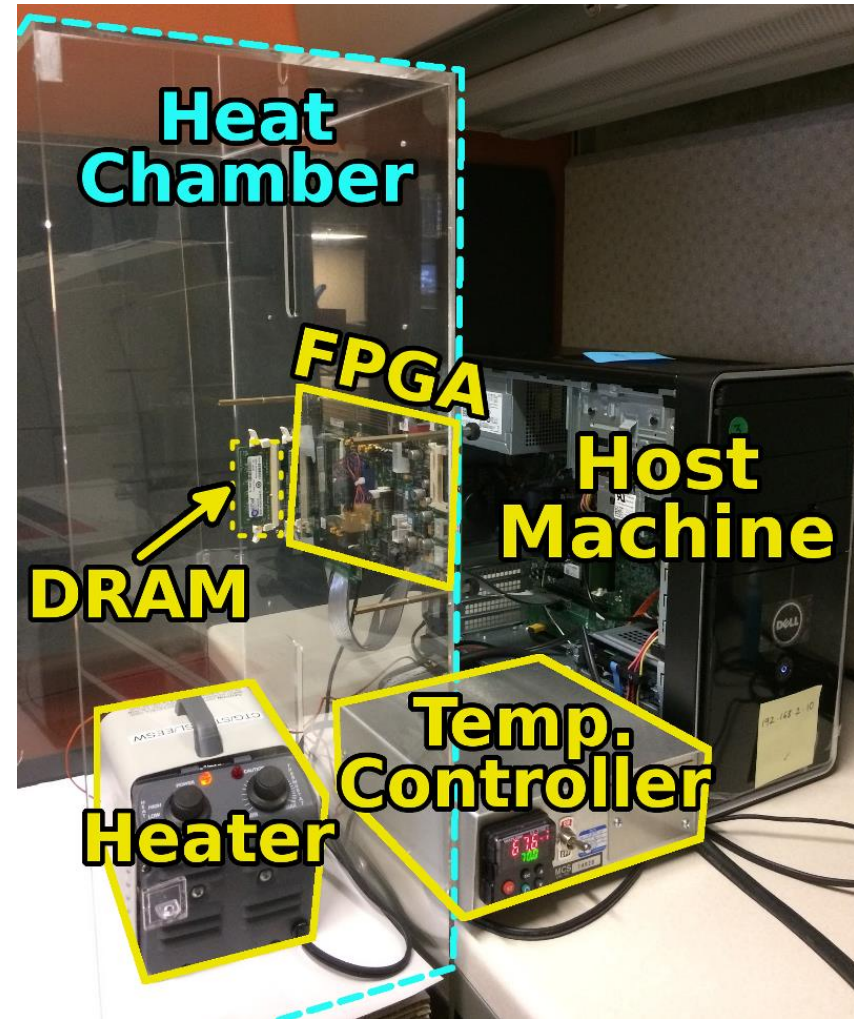
- Flexibility
  - Ability to test *any* DRAM operation
  - Ability to test *any* combination of DRAM operations and *custom* timing parameters
- Ease of use
  - **Simple** programming interface (C++)
  - **Minimal** programming effort and time
  - **Accessible** to a wide range of users
    - *who may 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

# SoftMC API

## Writing data to DRAM:

```
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);
```

*Instruction generator  
functions*



# SoftMC: Key Components

1. SoftMC API

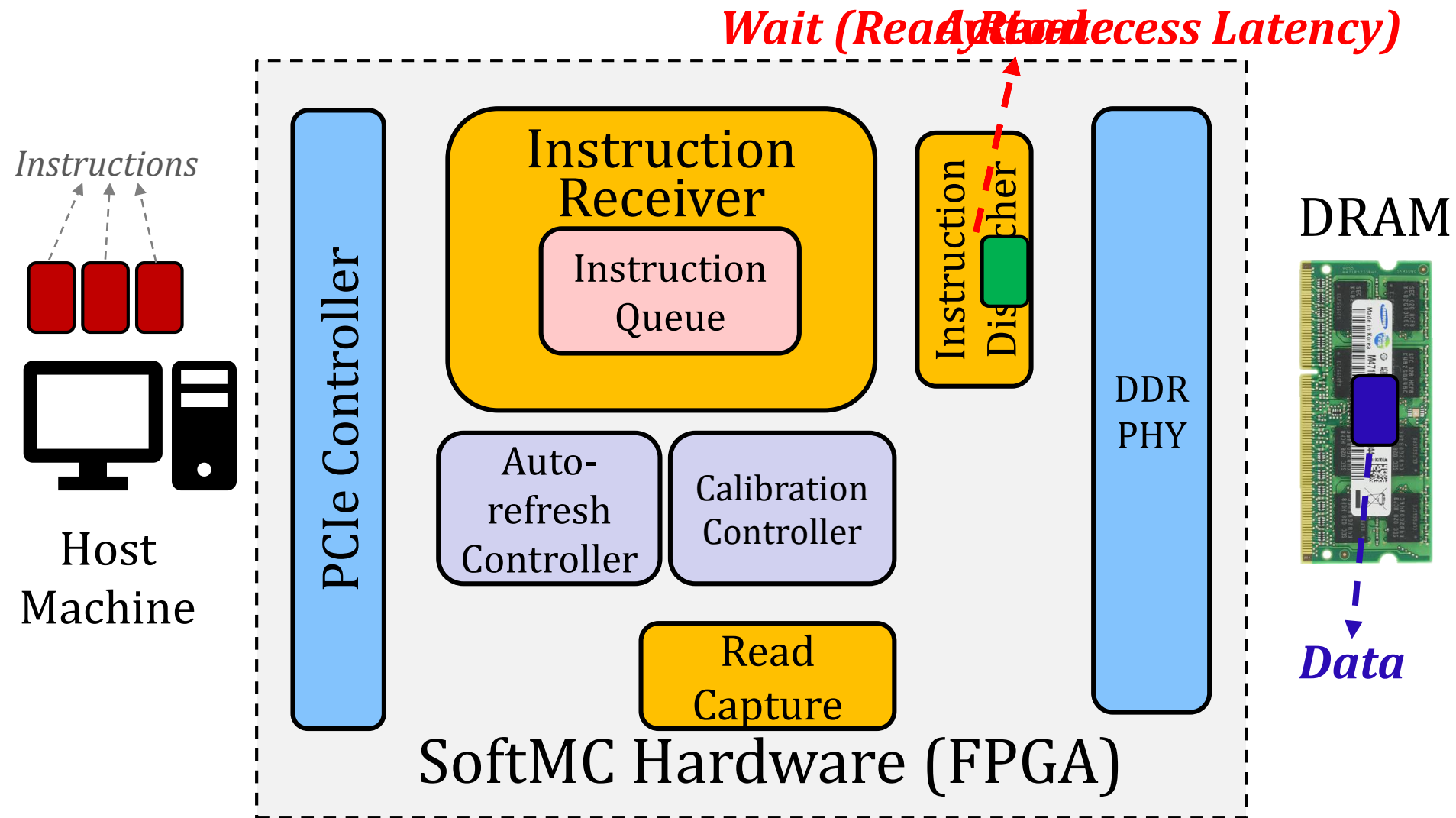
**2. PCIe Driver\***

↳ Communicates raw data with the FPGA

3. SoftMC Hardware

*\* Jacobsen, Matthew, et al. "RIFFA 2.1: A reusable integration framework for FPGA accelerators." TRETTS, 2015*

# SoftMC Hardware



# 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

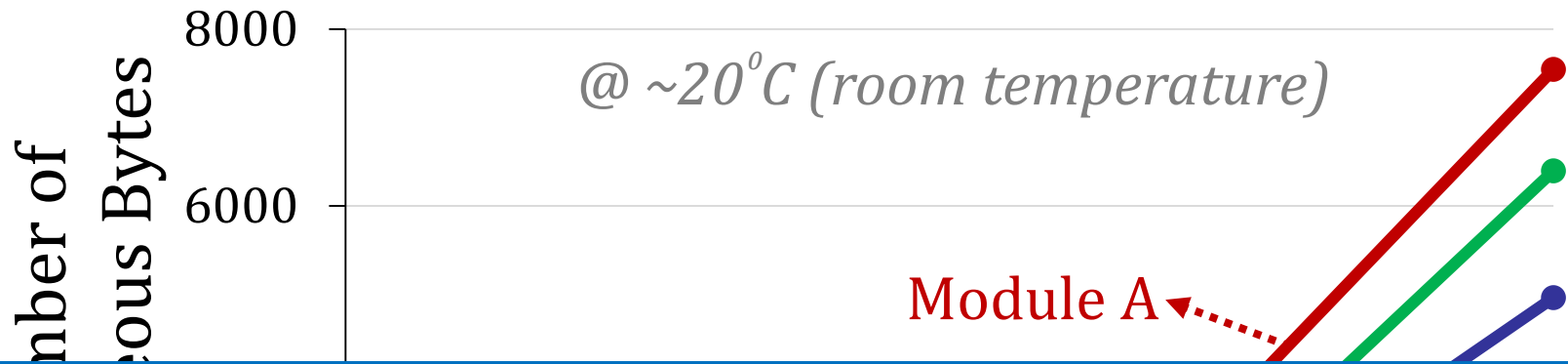
Write Reference  
Data to a Row

(Re

```
1  InstructionSequence iseq;
2  iseq.insert(genACT(bank, row));
3  iseq.insert(genWAIT(tRCD));
4  for(int col = 0; col < COLUMNS; col++){
5      iseq.insert(genWR(bank, col, data));
6      iseq.insert(genWAIT(tBL));
7  }
8  iseq.insert(genWAIT(tCL + tWR));
9  iseq.insert(genPRE(bank));
10 iseq.insert(genWAIT(tRP));
11 iseq.insert(genEND());
12 iseq.execute(fpga);
```

Can be implemented with just ~100 lines of code

# Retention Time Test: Results



**Validates the correctness of the SoftMC Infrastructure**

# 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

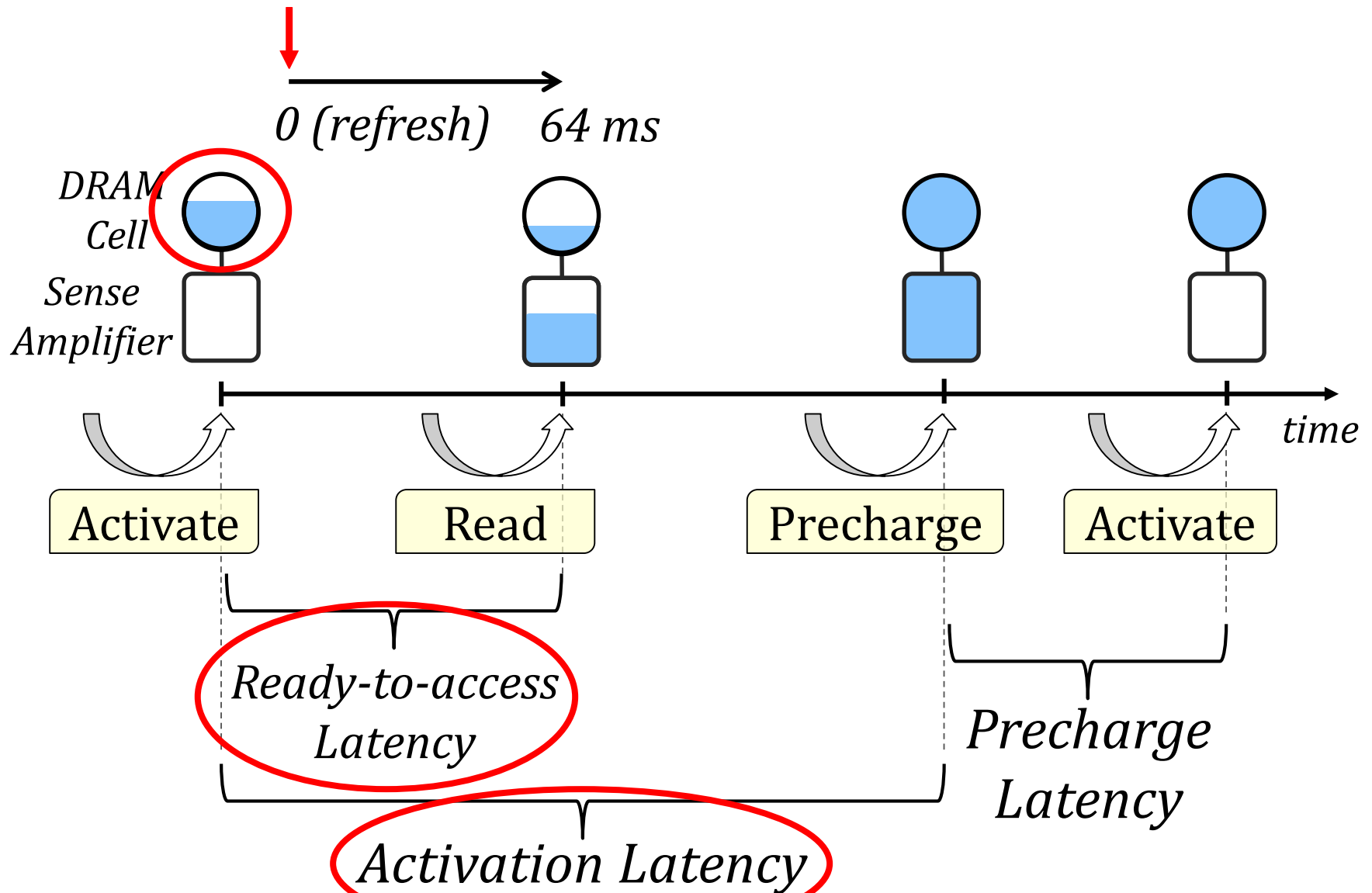
**NUAT**  
*(Shin+, HPCA 2014)*

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

---

A **highly-charged** cell can be  
accessed with **low latency**

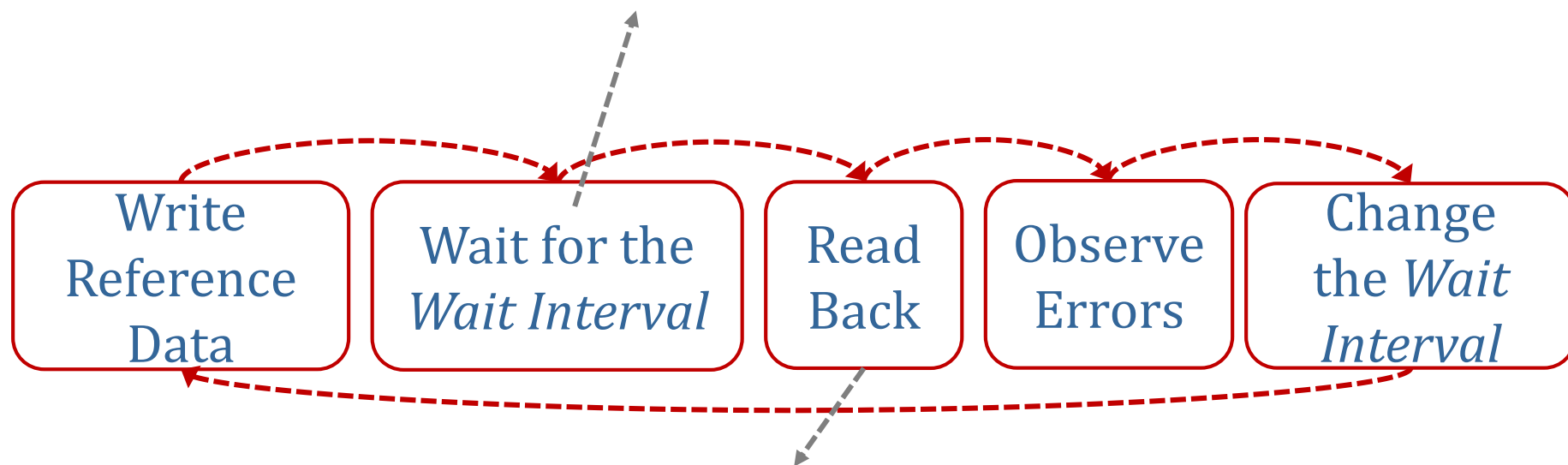
# How a Highly-Charged Cell Is Accessed Faster?



# Ready-to-access Latency Test

Longer wait → Lower cell charge

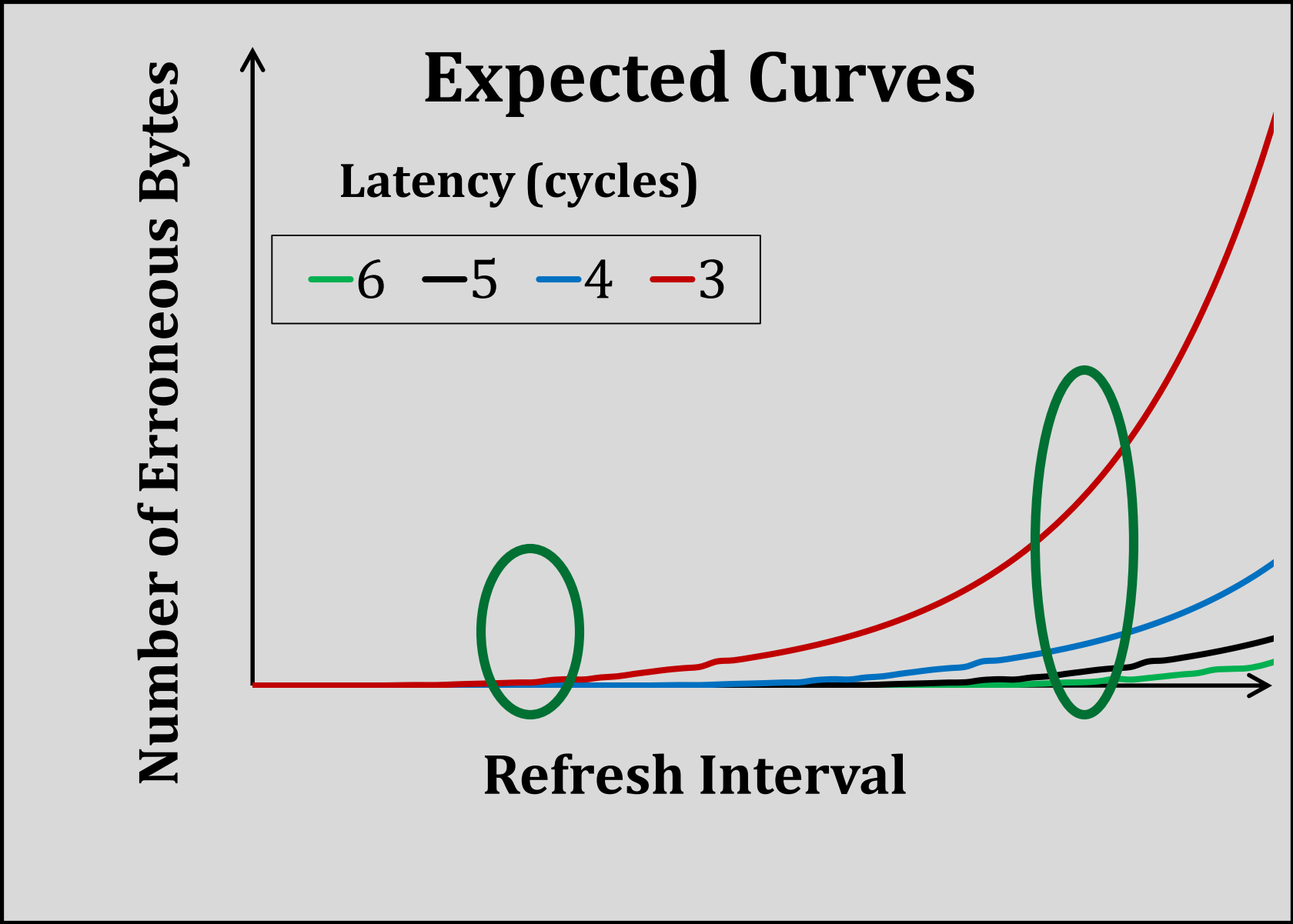
Shorter wait → Higher cell charge



With **custom** ready-to-access latency parameter

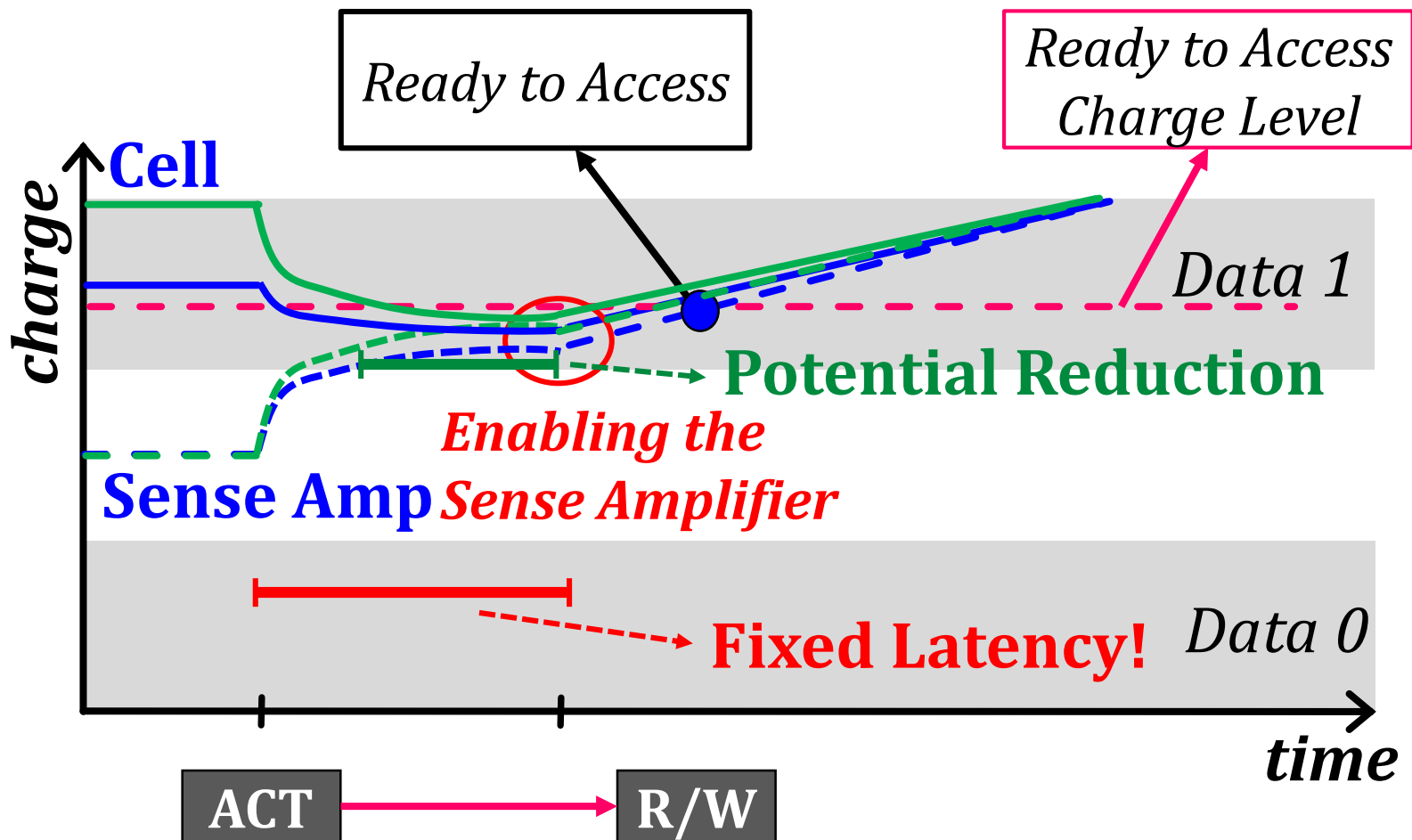
Can be implemented with just ~150 lines of code

# Ready-to-access Latency: Results



# 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**





# 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](https://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

**ETH** zürich

**Carnegie Mellon**



TOBB  
UNIVERSITY OF  
ECONOMICS AND TECHNOLOGY

# Backup Slides

# Key SoftMC Instructions

## InstrType

DDR (4)	<i>unused</i> (3)	CKE, CS (2), RAS, CAS, WE (6)	Bank (3)	Addr (16)
WAIT (4)	cycles (28)			
BUSDIR (4)	<i>unused</i> (27)			dir (1)
END (4)	<i>unused</i> (28)			

# SoftMC @ Github

The screenshot shows the GitHub repository page for CMU-SAFARI / SoftMC. At the top, there is a search bar and navigation links for Pull requests, Issues, and Gist. The repository name is CMU-SAFARI / SoftMC, with 3 Unwatch, 0 Star, and 0 Fork buttons. Below this, there are tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Pulse, Graphs, and Settings. A message states "No description, website, or topics provided." with an Edit button. A "New" button is used to add topics. A summary bar shows 3 commits, 1 branch, 0 releases, and 1 contributor. Below this, there are buttons for "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A commit history table follows, listing the latest commit by arthasSin (fix readme;) and several initial commits for folders (hw/boards/ML605, prebuilt, sw) and files (LICENSE, README.md). The README.md file is selected, showing the title "SoftMC v1.0" and a description: "SoftMC is an experimental FPGA-based memory controller design that could be used to develop tests for DDR3 SODIMMs. SoftMC currently supports only the Xilinx ML605 board. Soon, we will port SoftMC on other popularly used boards (e.g., Xilinx VC709)."

This repository Search

Pull requests Issues Gist

CMU-SAFARI / SoftMC

Unwatch 3 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description, website, or topics provided. Edit

New Add topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Commit	Message	Time
arthasSin	fix readme;	Latest commit ac75ef1 an hour ago
	hw/boards/ML605	initial commit; an hour ago
	prebuilt	initial commit; an hour ago
	sw	initial commit; an hour ago
	LICENSE	initial commit; an hour ago
	README.md	fix readme; an hour ago

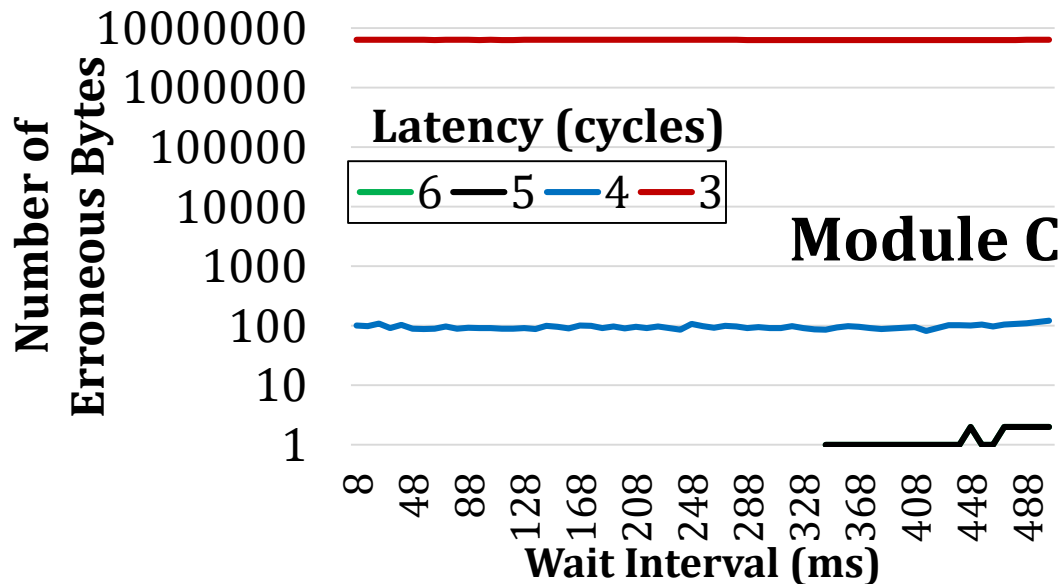
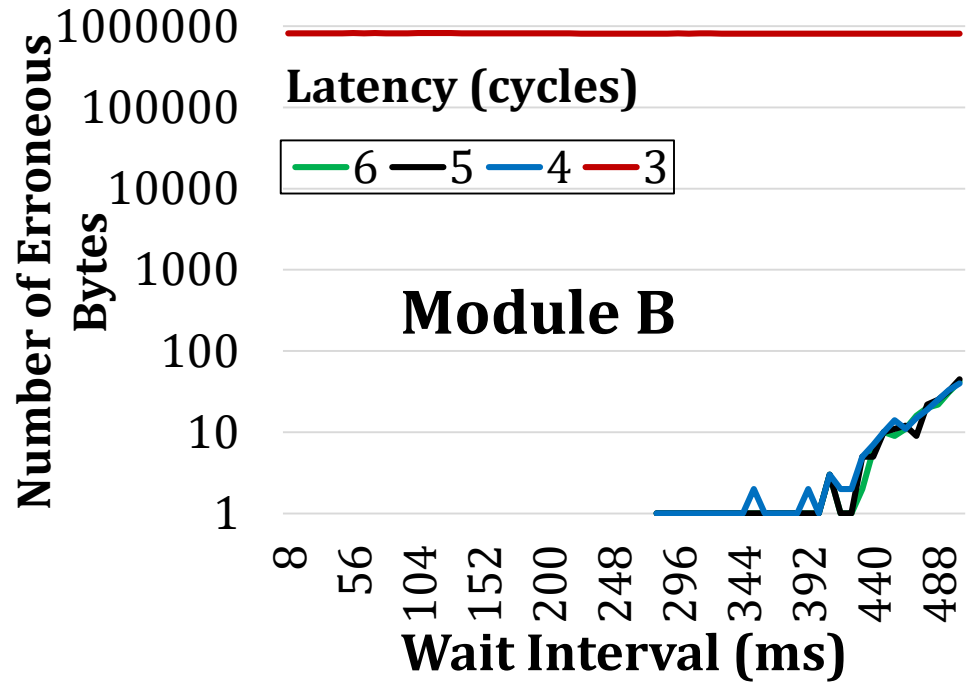
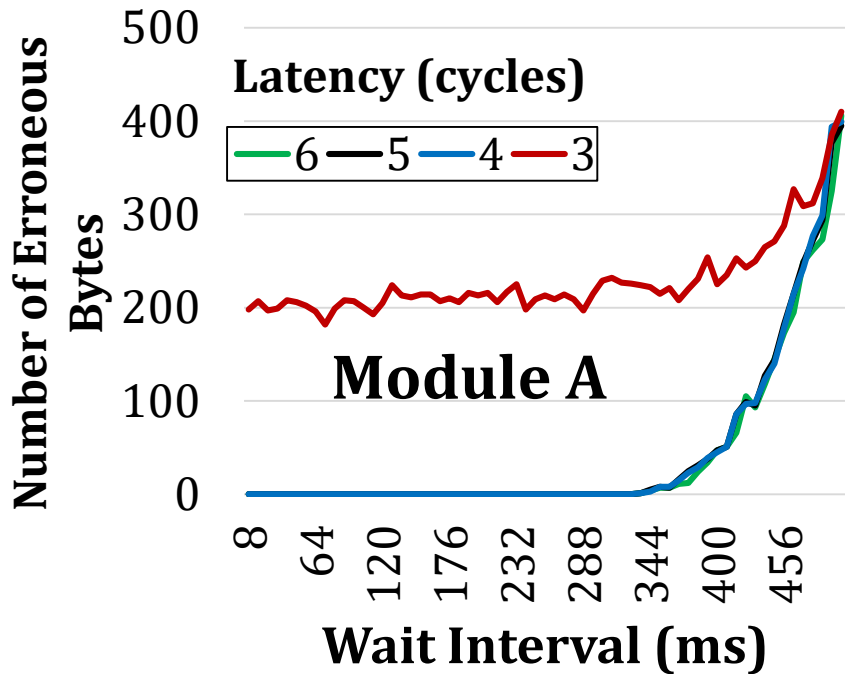
README.md

## SoftMC v1.0

SoftMC is an experimental FPGA-based memory controller design that could be used to develop tests for DDR3 SODIMMs. SoftMC currently supports only the Xilinx ML605 board. Soon, we will port SoftMC on other popularly used boards (e.g., Xilinx VC709).

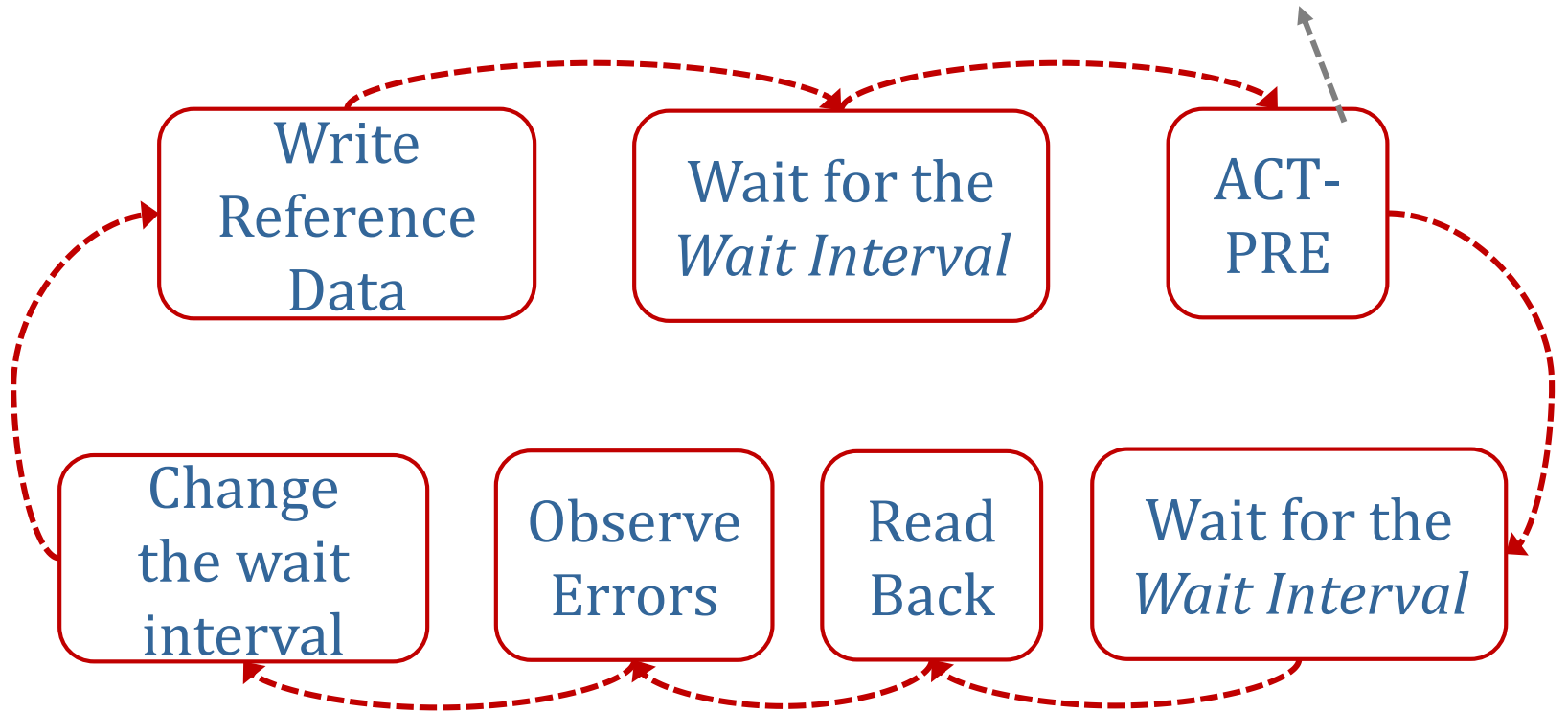


# Ready-to-Access Latency Test Results



# Activation Latency Test

With **low** activation latency parameter



# Activation Latency Test Results

