FMS
the Future of Memory and Storage

DRAM Simulation and Testing Infrastructures

Presenter: Haocong Luo

PhD Student, ETH Zurich

SAFARI Research Group

SAFARI

ETH zürich

Motivation

- **Robustness issues** in DRAM
  - Data retention
  - Read disturbance (RowHammer, RowPress, etc.)
  - …
- **Performance issues** in main memory system
  - Performance overhead analysis of read disturbance mitigation techniques
  - Processing-in-Memory architectures
  - Emerging memory technologies
  - …

DRAM simulation & testing infrastructures are needed to understand, characterize, and evaluate the robustness and performance of DRAM

Executive Summary

Ramulator 2.0: Modern, modular, and extensible DRAM simulator

- Unified functional and timing modeling of DRAM based on hierarchical state-machines
- Modular and extensible software architecture
- Models a wide range of DRAM standards and memory controller functionalities

DRAM Bender: Extensible and versatile FPGA-based commodity DRAM testing infrastructure

- Programmatically issue DRAM commands in arbitrary order and fine-grained timings
- Easy-to-use C++ and Python programming interface
- Enables a large body of works in DRAM read disturbance, random-number generation, processing-using-DRAM, etc.

Outline

1. Motivation
2. Ramulator 2.0
   - 2.1 Simulator Design & Key Features
   - 2.2 Case Studies
3. DRAM Bender
   - 3.1 Infrastructure Design & Key Features
   - 3.2 Case Studies
4. Conclusion & Future Work

Outline

1. Motivation
2. Ramulator 2.0
   - 2.1 Simulator Design & Key Features
   - 2.2 Case Studies
3. DRAM Bender
   - 3.1 Infrastructure Design & Key Features
   - 3.2 Case Studies
4. Conclusion & Future Work

Ramulator 2.0

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

Abstract—We present Ramulator 2.0, a highly modular and extensible DRAM simulator that enables rapid and agile implementation and evaluation of design changes in the memory controller and DRAM to meet the increasing research effort in improving the performance, security, and reliability of memory systems. Ramulator 2.0 abstracts and models key components in a DRAM-based memory system and their interactions into shared *interfaces* and independent *implementations*. Doing so enables easy modification and extension of the modeled functions of the memory controller and DRAM in Ramulator 2.0. The DRAM specification syntax of Ramulator 2.0 is concise and human-readable, facilitating easy modifications and extensions. Ramulator 2.0 implements a library of reusable templated lambda functions to model the functionalities of DRAM commands to simplify the implementation of new DRAM standards, including DDR5, LPDDR5, HBM3, and GDDR6. We showcase Ramulator 2.0's modularity and extensibility by implementing and evaluating a wide variety of RowHammer mitigation techniques that require *different* memory controller design changes. These techniques are added modularly as separate implementations *without* changing *any* code in the baseline memory controller implementation. Ramulator 2.0 is rigorously validated and maintains a fast simulation speed compared to existing cycle-accurate DRAM simulators. Ramulator 2.0 is open-sourced under the permissive MIT license at https://github.com/CMU-SAFARI/ramulator2.

IEEE CAL Paper

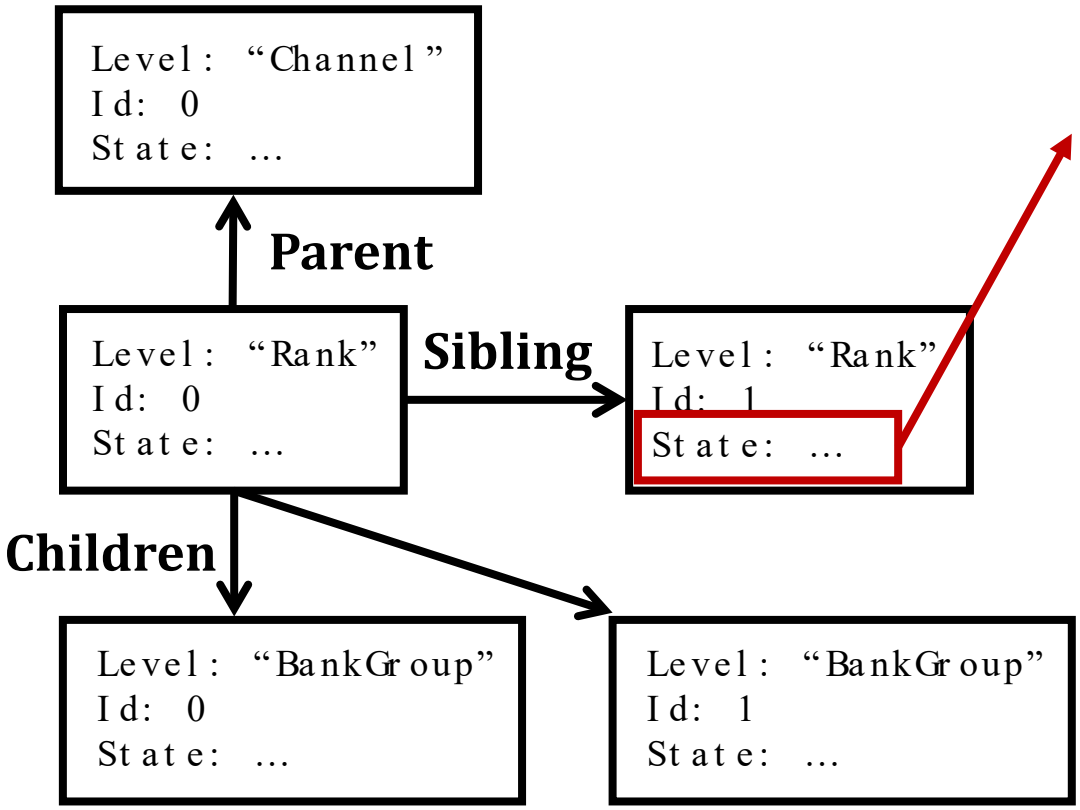
Open-source version Github repo:
CMU-SAFARI/ramulator2

Outline

1. Motivation
2. Ramulator 2.0
   - 2.1 Simulator Design & Key Features
   - 2.2 Case Studies
3. DRAM Bender
   - 3.1 Infrastructure Design & Key Features
   - 3.2 Case Studies
4. Conclusion & Future Work

Ramulator 2.0 Design and Features (I)

- **Hierarchical state-machine** based modeling of DRAM



State of a DRAM node:

- **Current state:** open, close, activating, etc.
- **Timing constraints:** Earliest time in the future that each DRAM command is allowed to be issued
- **Energy & power:** Time spent in each state and the number of DRAM commands served (DRAMPower model)
- **Can be extended to include more**

Ramulator 2.0 Design and Features (II)

- DRAM commands implemented as lambda functions that hierarchically traverses and updates the states of the nodes
  1. Checks the current states of the nodes to decode which DRAM command to issue
  2. Programmatically apply state changes
  3. Updates the timing constraints, power metrics, etc.
- Templated library of generalized DRAM command lambda functions allow reuse of command implementations across different DRAM standards

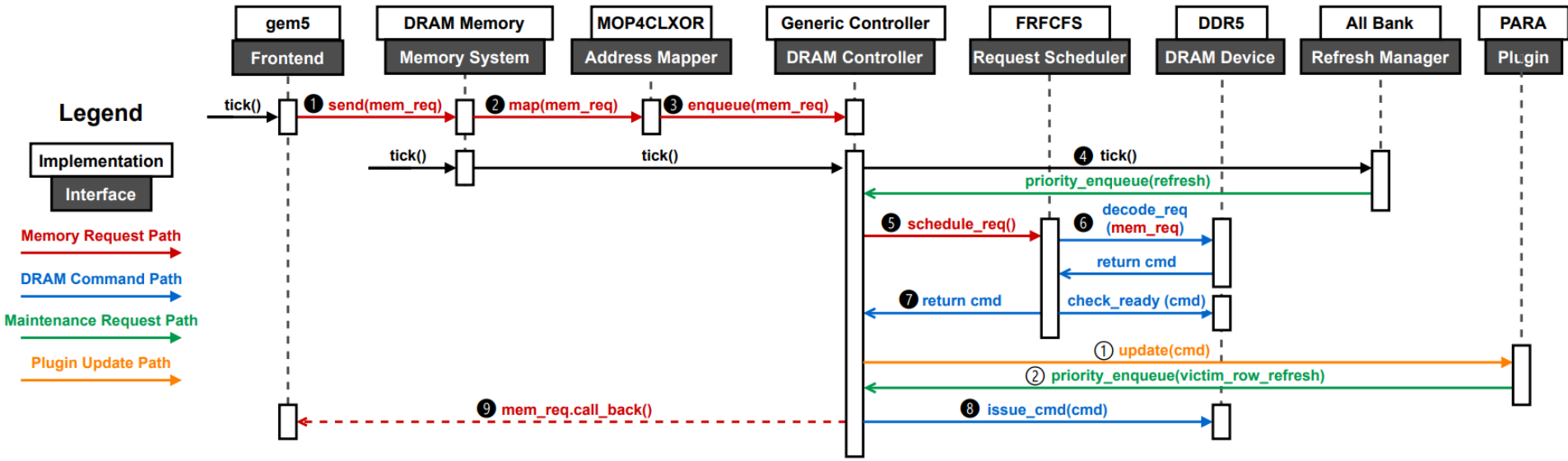
```
template <class DRAM_t>
int RequireAllBanksClosed(typename DRAM_t::Node* node,
     int cmd, int target_id, Clk_t clk) {
  // for all banks {
  // ...
    if (bank->m_state == DRAM_t::m_states["Closed"]) {
      continue;
    } else {
      return T::m_commands["PREab"];
    }
  // }
  return cmd;
};
```

Example DRAM Command Decode Function

Applicable to:
DDR3, DDR4, DDR5
LPDDR4, LPDDR5
HBM (1/2/3), GDDR6

Ramulator 2.0 Design and Features (III)

- Modular and extensible software architecture
  - All components in the memory system modeled with the same interface and different implementations
  - Example: The memory controller include:
    - Address Mapper, Request Scheduler, Refresh Controller, Row Policy, etc.
    - Each can be flexibly changed without hardcoding other parts



Ramulator 2.0 Design and Features (IV)

- More in the paper
  - More detailed explanation of modeling methodology
  - Authoring of DRAM specifications (organization, timings, etc.)
  - Memory controller plugin & RowHammer mitigations
  - Performance comparison with other DRAM simulators

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

IEEE CAL Paper

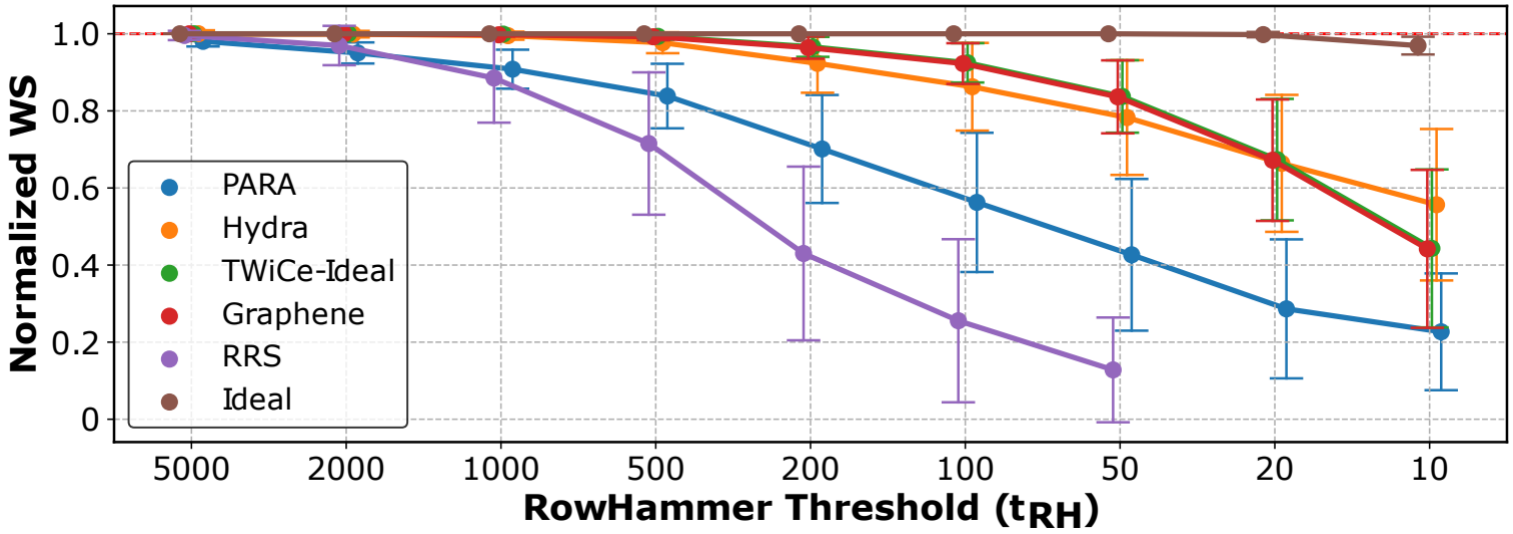
Open-source version Github repo:
CMU-SAFARI/ramulator2

Outline

1. Motivation
2. Ramulator 2.0
    - 2.1 Simulator Design & Key Features
    - 2.2 Case Studies
3. DRAM Bender
    - 3.1 Infrastructure Design & Key Features
    - 3.2 Case Studies
4. Conclusion & Future Work

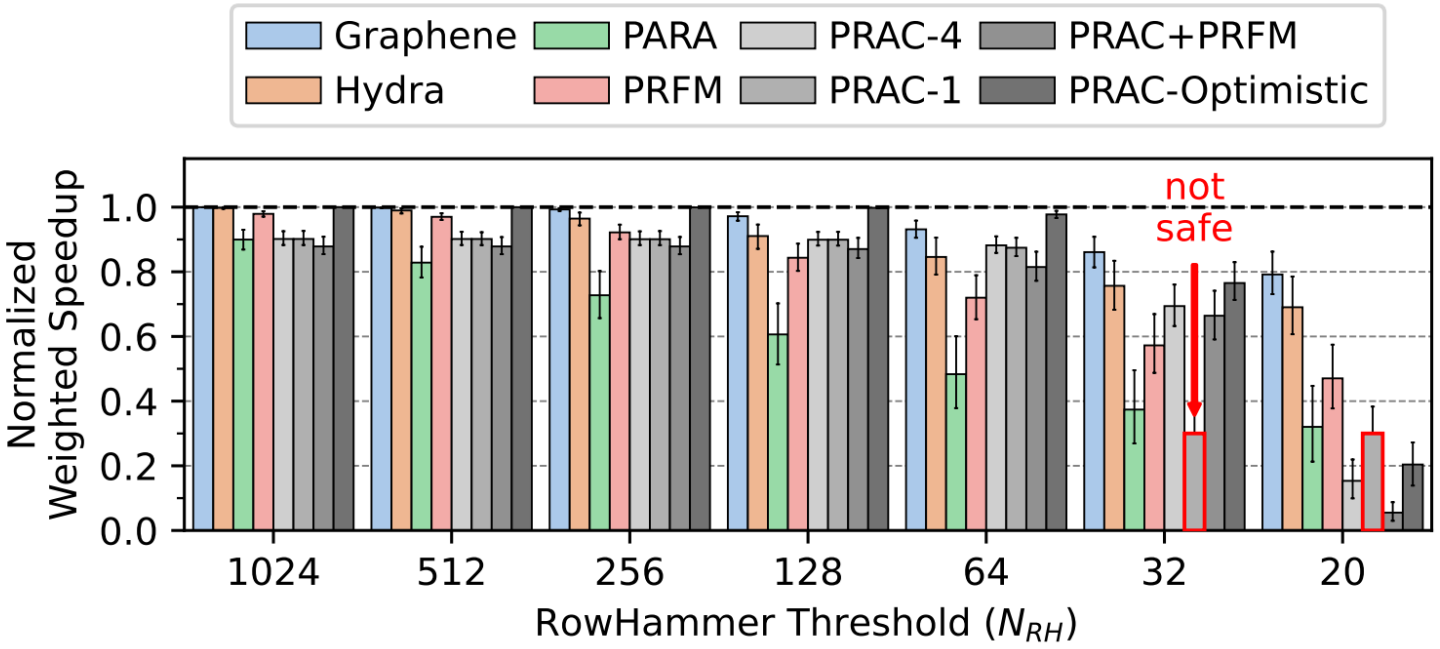
Ramulator 2.0 Case Studies (I)

- **Cross-section performance overhead evaluation of different RowHammer mitigation techniques [Luo+, IEEE CAL]**
  - Six different RowHammer mitigation techniques all implemented as plugins to the same memory controller implementation



Ramulator 2.0 Case Studies (II)

- **Performance evaluation of DDR5 Per Row Activation Counting (PRAC) [Canpolat+, DRAMsec'24]**
  - Memory controller implementation extended with support for per-row activation count tracking and back-off signal



Outline

1. Motivation
2. Ramulator 2.0
   - 2.1 Simulator Design & Key Features
   - 2.2 Case Studies
3. DRAM Bender
   - 3.1 Infrastructure Design & Key Features
   - 3.2 Case Studies
4. Conclusion & Future Work

DRAM Bender

DRAM Bender: An Extensible and Versatile FPGA-Based Infrastructure to Easily Test State-of-the-Art DRAM Chips

Ataberk Olgun, *Graduate Student Member, IEEE*, Hasan Hassan, A. Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, *Member, IEEE*, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu, *Fellow, IEEE*

IEEE TCAD Version

arXiv Version

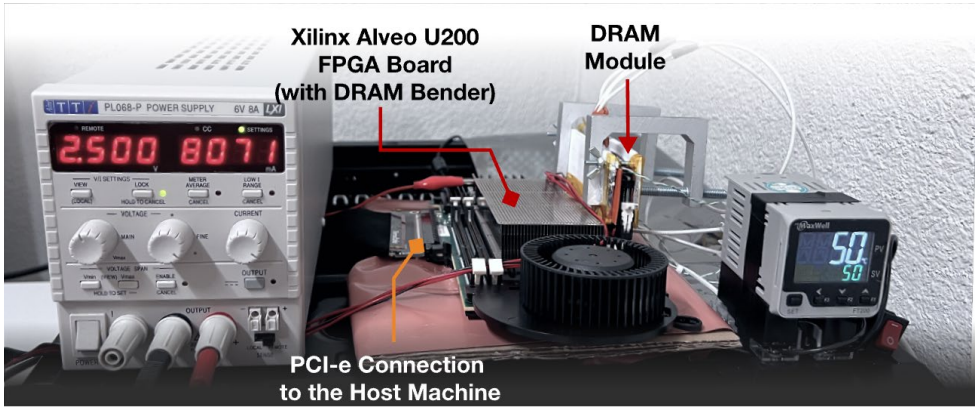
Github repo:
CMU-SAFARI/DRAM-Bender

Outline

1. Motivation
2. Ramulator 2.0
    - 2.1 Simulator Design & Key Features
    - 2.2 Case Studies
3. DRAM Bender
    - 3.1 Infrastructure Design & Key Features
    - 3.2 Case Studies
4. Conclusion & Future Work

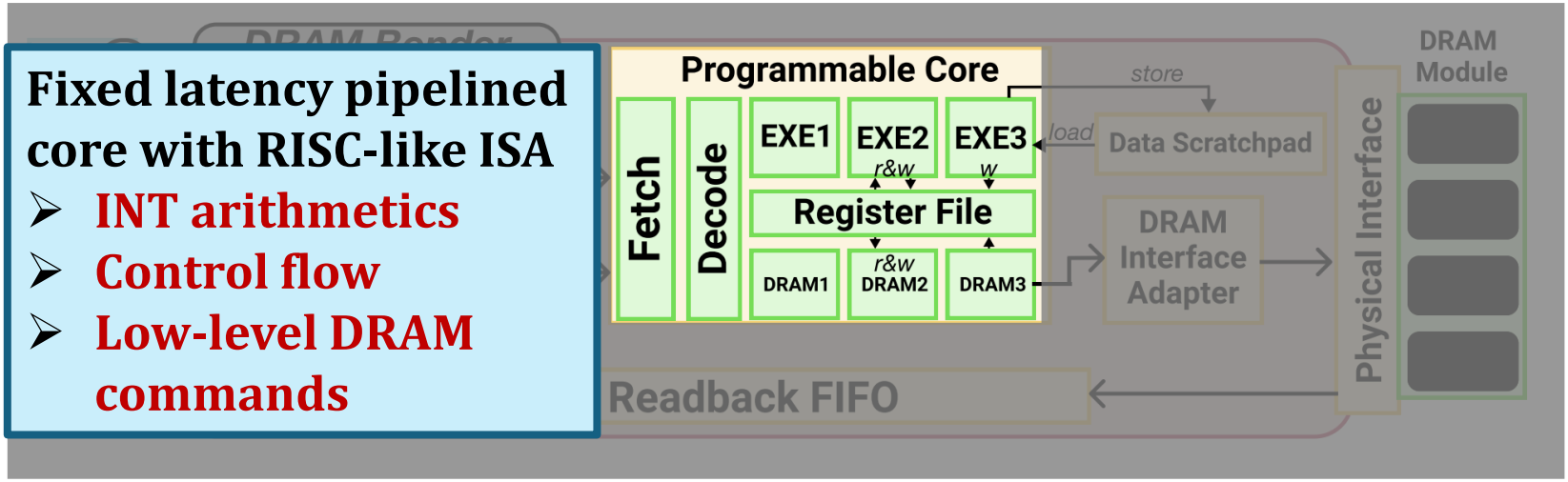
DRAM Bender Design & Key Features (I)

- An extensible and versatile FPGA-based infrastructure to easily test commodity DRAM



Fixed latency pipelined core with RISC-like ISA

- **INT arithmetics**
- **Control flow**
- **Low-level DRAM commands**



DRAM Bender Design & Key Features (II)

- Five FPGA boards supported out-of-the-box

Vendor	Model	DRAM Standard
Xilinx	XCU200	DDR4 DIMM/SODIMM
BittWare	XUPS3S	DDR4 SODIMM
BittWare	XUPP3R	DDR4 DIMM
BittWare	XUPVVH	DDR4 DIMM
Xilinx	XCU50	HBM2

- Optional control for
  - DRAM temperature (external heater pad)
  - Voltage (V_{PP} for DDR4 through DDR4 riser board)

DRAM Bender Design & Key Features (III)

- An easy-to-use and flexible high-level API (C++ and Python)

```
1   p.appendLI(hammerCount, 0);
2   p.appendLabel("HAMMER1");
3   p.appendACT(bank, false, A1, false, tRAS);
4   p.appendPRE(bank, false, false, tRP);
5   p.appendADDI(hammerCount, hammerCount, 1);
6   p.appendBL(hammerCount, T, "HAMMER1");
7   p.appendLI(hammerCount, 0);
8   p.appendLabel("HAMMER2");
9   p.appendACT(bank, false, A2, false, tRAS);
10  p.appendPRE(bank, false, false, tRP);
11  p.appendADDI(hammerCount, hammerCount, 1);
12  p.appendBL(hammerCount, T, "HAMMER2");
```

Example DRAM Bender program: Double-sided RowHammer

Easy to devise new experiments to uncover new insights.

DRAM Bender Design & Key Features (IV)

- More in the paper and Github repository
  - Detailed hardware design and DRAM Bender ISA
  - How to extend the DRAM Bender ISA
  - More case studies
  - …

DRAM Bender: An Extensible and Versatile FPGA-Based Infrastructure to Easily Test State-of-the-Art DRAM Chips

Ataberk Olgun, *Graduate Student Member, IEEE*, Hasan Hassan, A. Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, *Member, IEEE*, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu, *Fellow, IEEE*

IEEE TCAD Version

arXiv Version

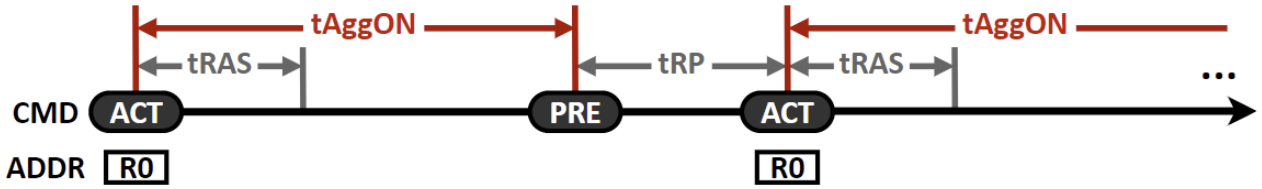
Github repo:
CMU-SAFARI/DRAM-Bender

Outline

1. Motivation
2. Ramulator 2.0
    - 2.1 Simulator Design & Key Features
    - 2.2 Case Studies
3. DRAM Bender
    - 3.1 Infrastructure Design & Key Features
    - 3.2 Case Studies
4. Conclusion & Future Work

DRAM Bender Case Studies (I)

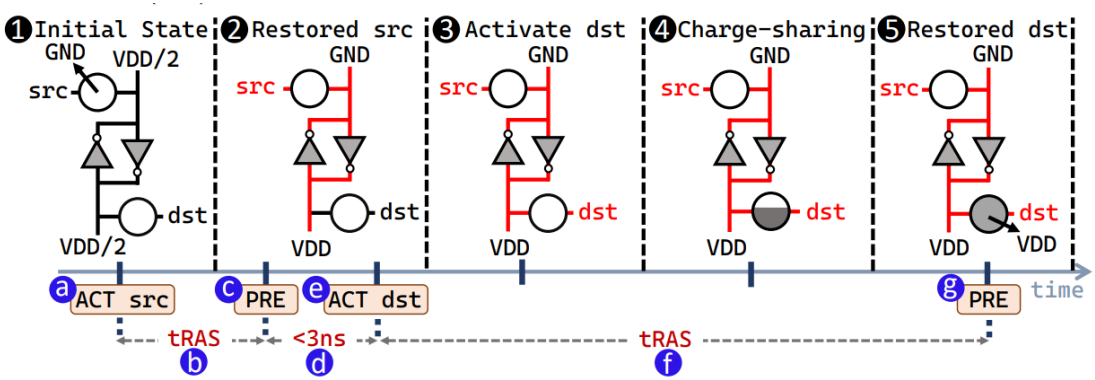
- **RowPress Vulnerability in Modern DRAM Chips [Luo+, ISCA'23, IEEE MICRO Top Picks 2024]**
  - Keeping a DRAM row open for a long period of time induces bitflips *without* as many row activations as RowHammer
  - Different underlying error mechanism than RowHammer
  - Insights from DRAM Bender experiments transferred to real system demonstration that breaks TRR
- **Key DRAM Bender Features Used**
  - Flexible and accurate timing control of DRAM commands
  - Programmatical control flow and data pattern generation



Example RowPress DRAM Access Pattern

DRAM Bender Case Studies (II)

- **Functionally-Complete Boolean Logic in Real DRAM Chips [Yüksel+, HPCA'24]**
  - Leverage the differential sensing of BLSA to complement MAJ-based Compute-Using-DRAM with NOT operations
  - Violates JEDEC DDR4 command sequence and timings to trigger multi-row activation in commodity DRAM



- **Key DRAM Bender Features Used**
  - Direct issuing of DRAM commands
  - Flexible and accurate timing control of DRAM commands

DRAM Bender Case Studies (III)

- **Large body of works enabled by DRAM Bender**

  - Ismail Emir Yuksel, Yahya Can Tugrul, F. Nisa Bostanci, Geraldo F. Oliveira, A. Giray Yaglikci, Ataberk Olgun, Melina Soysal, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu, **"Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis"**, *Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Brisbane, Australia, June 2024.
  - Ataberk Olgun, Majd Osseiran, Abdullah Giray Yaglikci, Yahya Can Tugrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez Luna, and Onur Mutlu, **"Read Disturbance in High Bandwidth Memory: A Detailed Experimental Study on HBM2 DRAM Chips"**, *Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Brisbane, Australia, June 2024.
  - Haocong Luo, Ismail Emir Yüksel, Ataberk Olgun, A. Giray Yağlıkçı, Mohammad Sadrosadati, and Onur Mutlu, **"An Experimental Characterization of Combined RowHammer and RowPress Read Disturbance in Modern DRAM Chips"** , *Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Disrupt Track (**DSN Disrupt**)*, Brisbane, Australia, June 2024.
  - Nam, Hwayong, Seung Hyup Baek, Minbok Wi, Michael Jaemin Kim, Jaehyun Park, Chihun Song, Nam Sung Kim and Jung Ho Ahn. "DRAMScope: Uncovering DRAM Microarchitecture and Characteristics by Issuing Memory Commands." *Proceedings of the* ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), Buenos Aires, Argentina, July 2024.
  - Abdullah Giray Yaglikci, Geraldo Francisco de Oliveira, Yahya Can Tugrul, Ismail Yuksel, Ataberk Olgun, Haocong Luo, and Onur Mutlu, **"Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions"** ,*Proceedings of the 30th International Symposium on High-Performance Computer Architecture (**HPCA**)*, April 2024.
  - Ismail Emir Yuksel, Yahya Can Tugrul, Ataberk Olgun, F. Nisa Bostanci, A. Giray Yaglikci, Geraldo F. Oliveira, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu, **"Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis"** , *Proceedings of the 30th International Symposium on High-Performance Computer Architecture (**HPCA**)*, April 2024.
  - Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu, **"PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"**, *ACM Transactions on Architecture and Code Optimization (**TACO**)*, March 2023.
  - A. Giray Yağlıkçı, Haocong Luo, Geraldo F. de Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S. Kim, Lois Orosa, and Onur Mutlu, **"Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices"**, *Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Baltimore, MD, USA, June 2022.
  - Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu, **"A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"**, *Proceedings of the 54th International Symposium on Microarchitecture (**MICRO**)*, Virtual, October 2021.
  - Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, and Onur Mutlu, **"Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications"**, *Proceedings of the 54th International Symposium on Microarchitecture (**MICRO**)*, Virtual, October 2021.
  - Ataberk Olgun, Minesh Patel, A. Giray Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu, **"QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips"**, *Proceedings of the 48th International Symposium on Computer Architecture (**ISCA**)*, Virtual, June 2021.
  - **...**

Outline

1. Motivation
2. Ramulator 2.0
    - 2.1 Simulator Design & Key Features
    - 2.2 Case Studies
3. DRAM Bender
    - 3.1 Infrastructure Design & Key Features
    - 3.2 Case Studies
4. Conclusion & Future Work

Conclusion

Ramulator 2.0: Modern, modular, and extensible DRAM simulator

- Unified functional and timing modeling of DRAM based on hierarchical state-machines
- Modular and extensible software architecture
- Models a wide range of DRAM standards and memory controller functionalities

DRAM Bender: Extensible and versatile FPGA-based commodity DRAM testing infrastructure

- Programmatically issue DRAM commands in arbitrary order and fine-grained timings
- Easy-to-use C++ and Python programming interface
- Enables a large body of works in DRAM read disturbance, random-number generation, processing-using-DRAM, etc.

Future Works

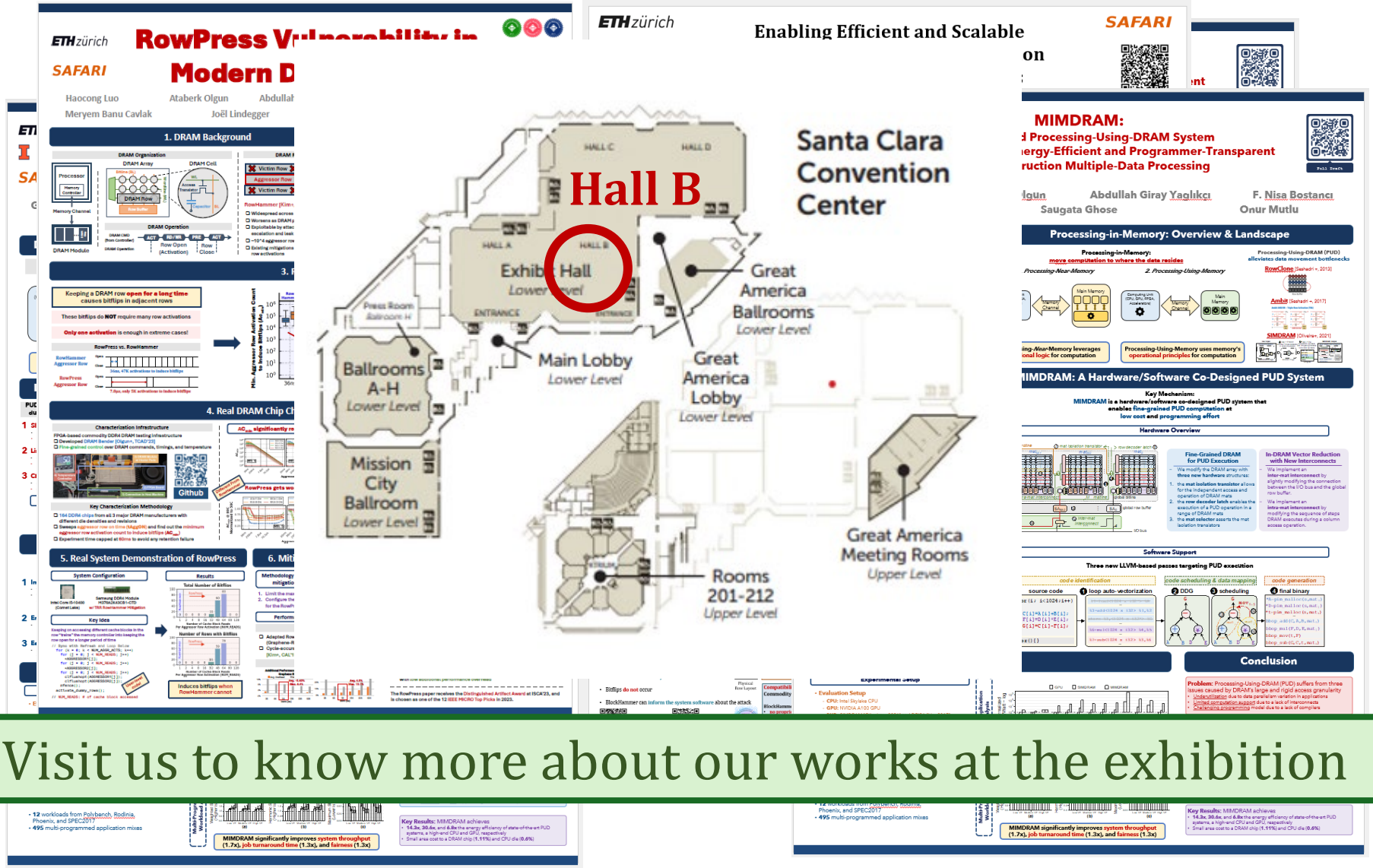
Ramulator 2.0

- ❑ Unit & regression test coverage
- ❑ More DRAM standards and emerging technologies
- ❑ More detailed memory controller modeling (i.e., pipelined scheduler and gear ratio)
- ❑ Generalizable modeling for PuM/PnM architectures
- ❑ …

DRAM Bender:

- ❑ DDR5 support
- ❑ Automatic reverse engineering of DRAM microarchitecture
- ❑ Better and more stable voltage control
- ❑ …

Posters



Visit us to know more about our works at the exhibition

FMS
the Future of Memory and Storage

DRAM Simulation and Testing Infrastructures

Presenter: Haocong Luo

Ramulator 2 Paper

Github repo:
CMU-SAFARI/ramulator2

DRAM Bender Paper

Github repo:
CMU-SAFARI/DRAM-Bender

SAFARI

ETH zürich