D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim[‡§] Minesh Patel[§] Hasan Hassan[§] Lois Orosa[§] Onur Mutlu[§‡]
[‡]Carnegie Mellon University [§]ETH Zürich

HPCA2019

Presented by Manuel Hässig

Based on: https://people.inf.ethz.ch/omutlu/pub/drange-dram-latency-based-true-random-number-generator_hpca19-talk.pptx

Executive Summary

- **Motivation**: High-throughput true random numbers enable system security and various randomized algorithms.
  - Many systems (e.g., IoT, mobile, embedded) do not have dedicated **True Random Number Generator (TRNG)** hardware, but have DRAM devices
- **Problem**: Current TRNGs either
  1) need **dedicated hardware**
  2) do **not** sample a fundamentally non-deterministic entropy source or are **too slow** for continuous high throughput operation
- **Goal**: A novel and effective TRNG that uses existing commodity DRAM to provide random values with 1) **high-throughput**, 2) **low latency** and 3) no adverse effect on concurrently running applications

Executive Summary (contd.)

- **D-RaNGe**: Reduce DRAM access latency below reliable values and **exploit the failure probability of DRAM cells** to generate random numbers
- **Evaluation**:
  1) Experimentally characterize **282 real LPDDR4 DRAM devices**
  2) **D-RaNGe (717.4 Mb/s)** has significantly higher throughput **(211x)**
  3) **D-RaNGe (100ns)** has significantly lower latency **(180x)**

Outline

- **Motivation, Problem, Goal**
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

Motivation

- High-throughput **True Random Numbers** are required for many real-world applications (i.e., **cryptography** for encryption, randomized algorithms, scientific simulation)
- True random numbers can only be generated by sampling a physical process
  - Most systems rely on **dedicated TRNG hardware**
  - Smaller devices (e.g., IoT, mobile, embedded) **require, but often lack**, a high-throughput TRNG
- DRAM devices are available on most systems

Problem

- In spite of widespread need for TRNGs, they are **not widely used**
  - Dedicated hardware is **costly** and needs **a lot of room**
  - TRNGs implemented in DRAM are either **too slow** or unable to provide continuous high-throughput streams of true random numbers

Goal

Implement an **effective TRNG** in commodity DRAM:

1) Fully non-deterministic
2) Continuous high-throughput stream of true random numbers
3) Low latency
4) Low system interference
5) Low energy overhead
6) Low implementation cost

Outline

- Motivation, Problem, Goal
- **Background**
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

True Random Number Generator

- Samples a non-deterministic physical phenomenon to construct a bitstream of random data. It typically consists of
  1) **Entropy source**: physical phenomenon (e.g., jitter, Brownian motion, metastable circuit). Should not be visible nor be modifiable by an adversary.
  2) **Randomness extraction technique**: harvests random data from the entropy source. Should have high throughput and not disturb the entropy source.
  3) **Post-processing**: de-bias bits and correct for correlations in the bitstream
- Quality of a random bit is quantified by its entropy

Shannon Entropy

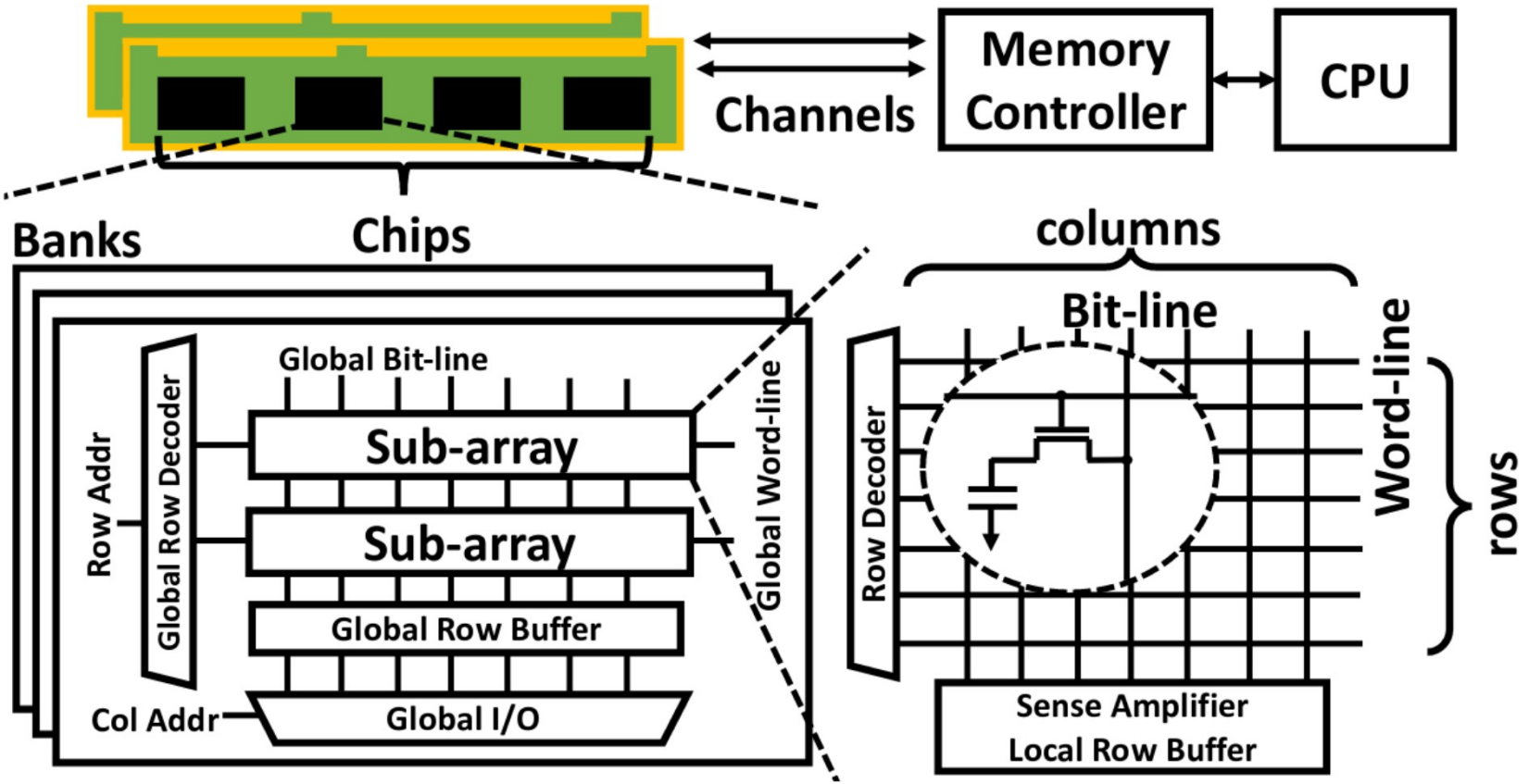
- Notion from Information Theory [Shannon, 1948]
- For a random variable $X \in \{0,1\}$ we define

$$H(X) := -\sum_{x \in \{0,1\}} \Pr[X = x] \log \Pr[X = x].$$

- It describes the amount of information in bits we gain (in expectation) by looking at the value of X.
- For X uniformly distributed over {0,1} (ideal random bit) we have

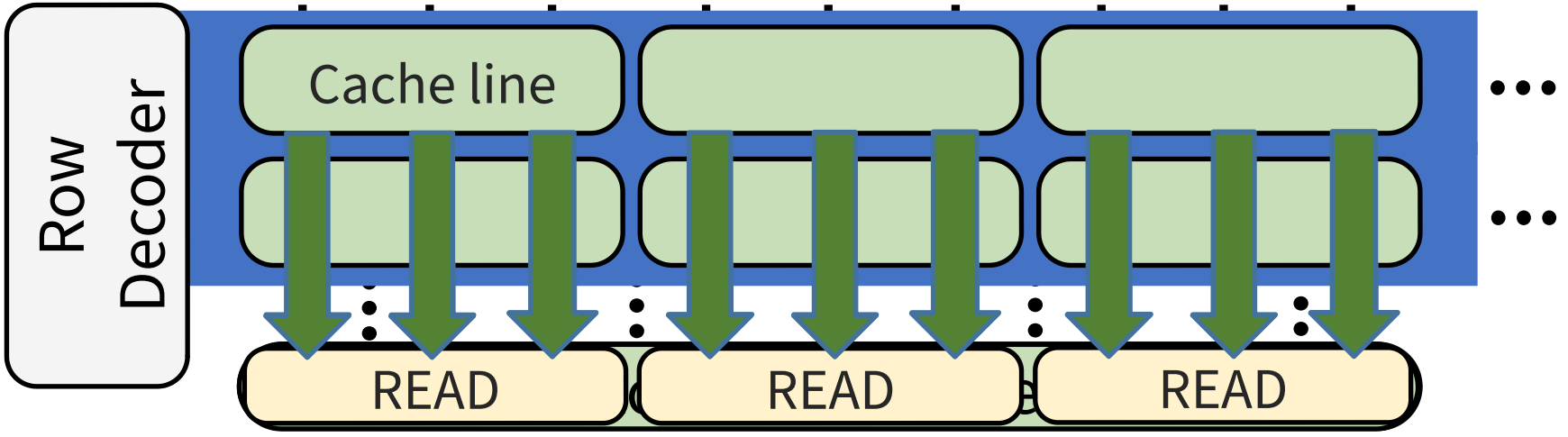
$$H(X) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 0.5 + 0.5 = 1.$$

DRAM organization

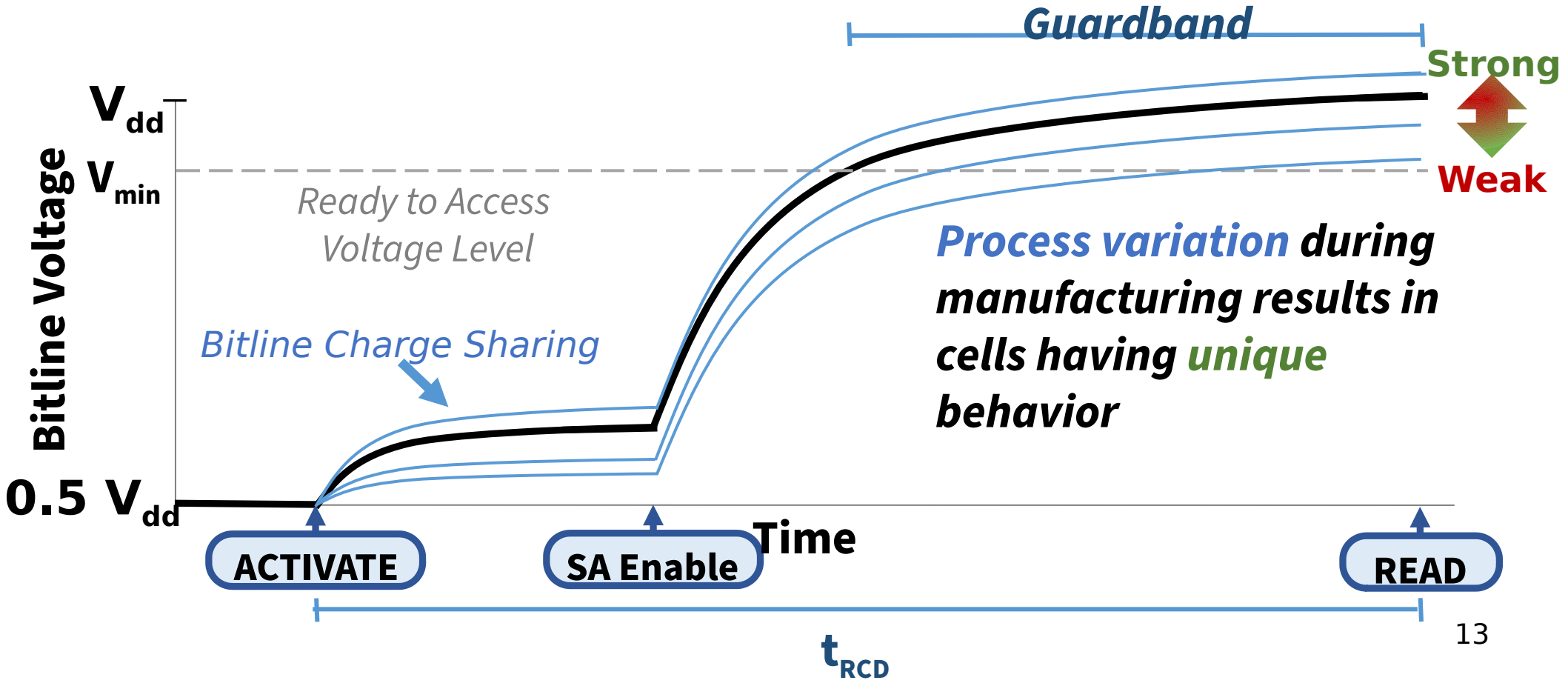


Source: Source: Gao+, ComputeDRAM: In-Memory Compute using off-the-shelf DRAMs

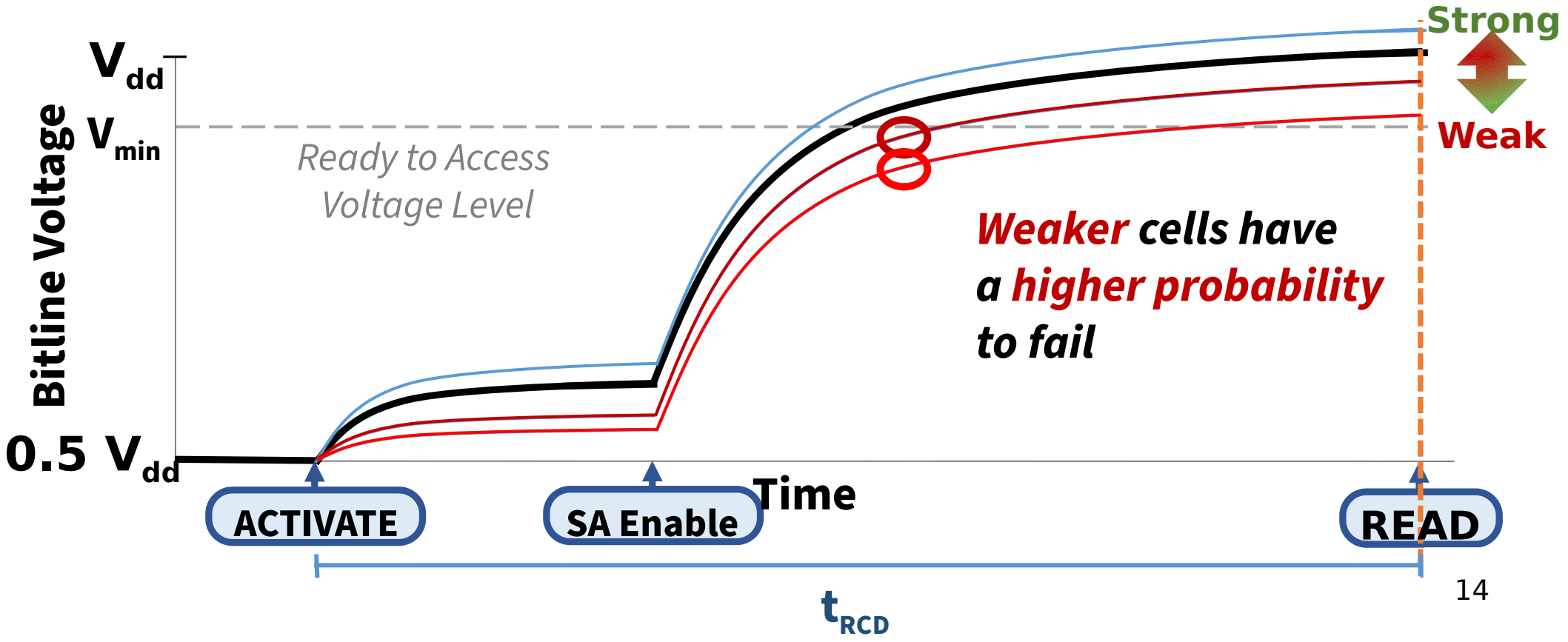
DRAM Operation



DRAM Access and Failures

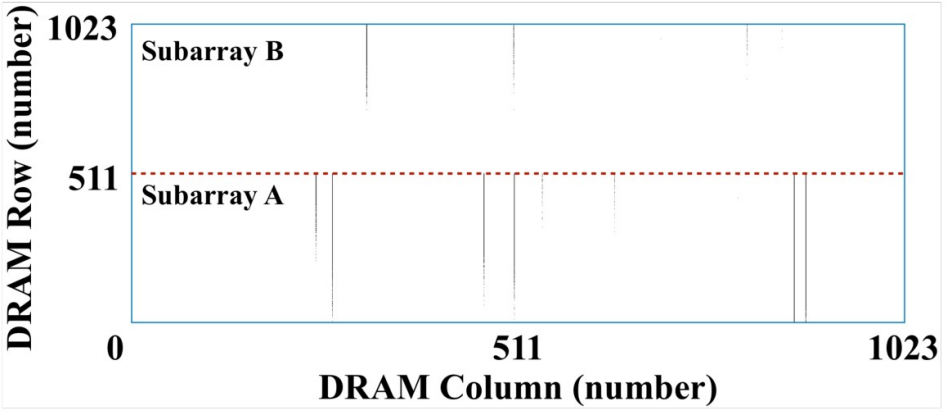


DRAM Accesses and Failures



Activation Failures

- Location dependence
  - Activation failure only on first row access
  - A weaker local sense amplifier => cells on bitline with higher F_{prob}
- Data pattern dependent
- Temperature dependent
  - Higher temperature => higher F_{prob}
- Not time dependent



Outline

- Motivation, Problem, Goal
- Background
- **Key Idea**
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

Key Idea

Based on two **key observations**:

1) The latency failure probability of a cell is inherently related to **random process variation** from manufacturing. Thus, we can extract **true random values**.
2) An activation failure can be induced very quickly.

The **key idea** is to extract random values from DRAM cells that **fail truly randomly**.

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- **Methodology**
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

Testing environment

Experimental test setup:

- 282 2y-nm **LPDDR4 DRAM** devices with **2GB** size from **3 major manufacturers**
- **Thermally controlled** testing chamber
  - Ambient temperature range: **{40°C – 55°C} ± 0.25°C**
  - DRAM temperature held at **15°C above ambient**
- **Fine grained control** over DRAM command and timing parameters with SoftMC
  - t_{RCD} reduced from 18ns to 10ns

Outline

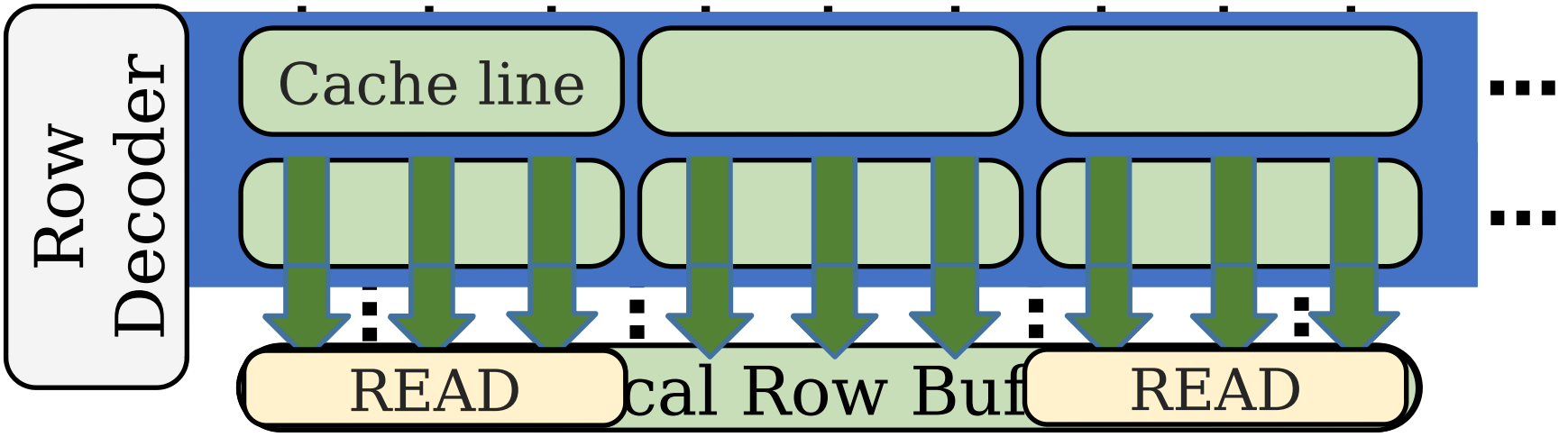
- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- **D-RaNGe**
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

D-RaNGe: Identification of RNG cells

- Finding cells that **fail randomly** when accessed with reduced t_{RCD} (**RNG cell**)
  1) Read cell **1000 times** with reduced t_{RCD}
  2) Approximate **Shannon Entropy**
  3) Select **good enough** ($F_{prob} \approx 0.5$) cells
- Repeated access to RNG cells with reduced t_{RCD} yields a **bitstream of true random numbers**
- Identification step needs to be repeated
  - For different temperatures
  - After some time

D-RaNGe Access Pattern

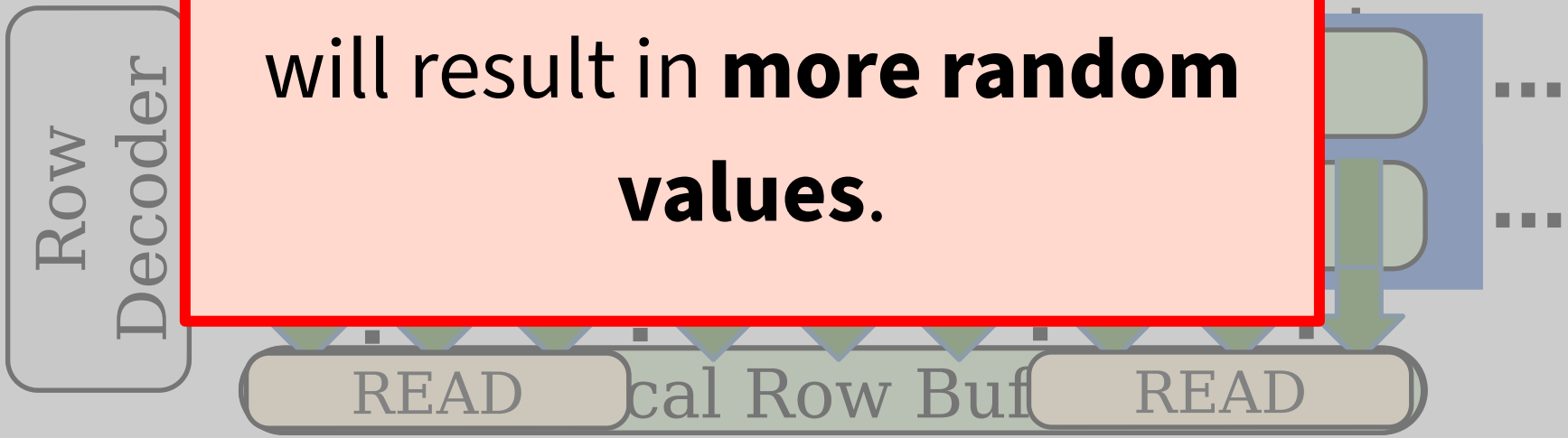
- To maximize the bits that are accessed immediately following activation, we alternate accesses to distinct rows in each bank
  - quickly generate t_{RCD} failures within cache lines in two rows
  - maximizes the number of access failures per activation



D-RaNGe Access Pattern

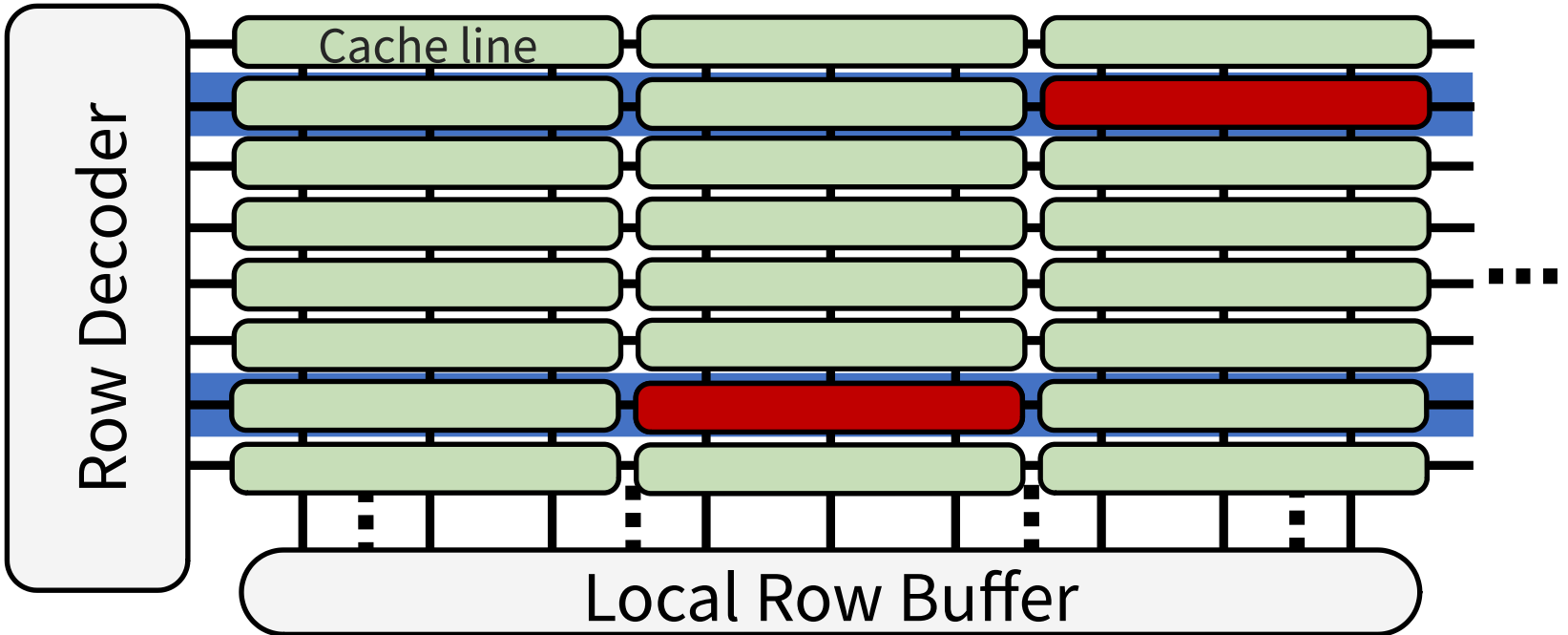
- To maxi ivation, we alternat
  - quic
  - max

Accessing cache lines containing **more RNG cells** will result in **more random values**.



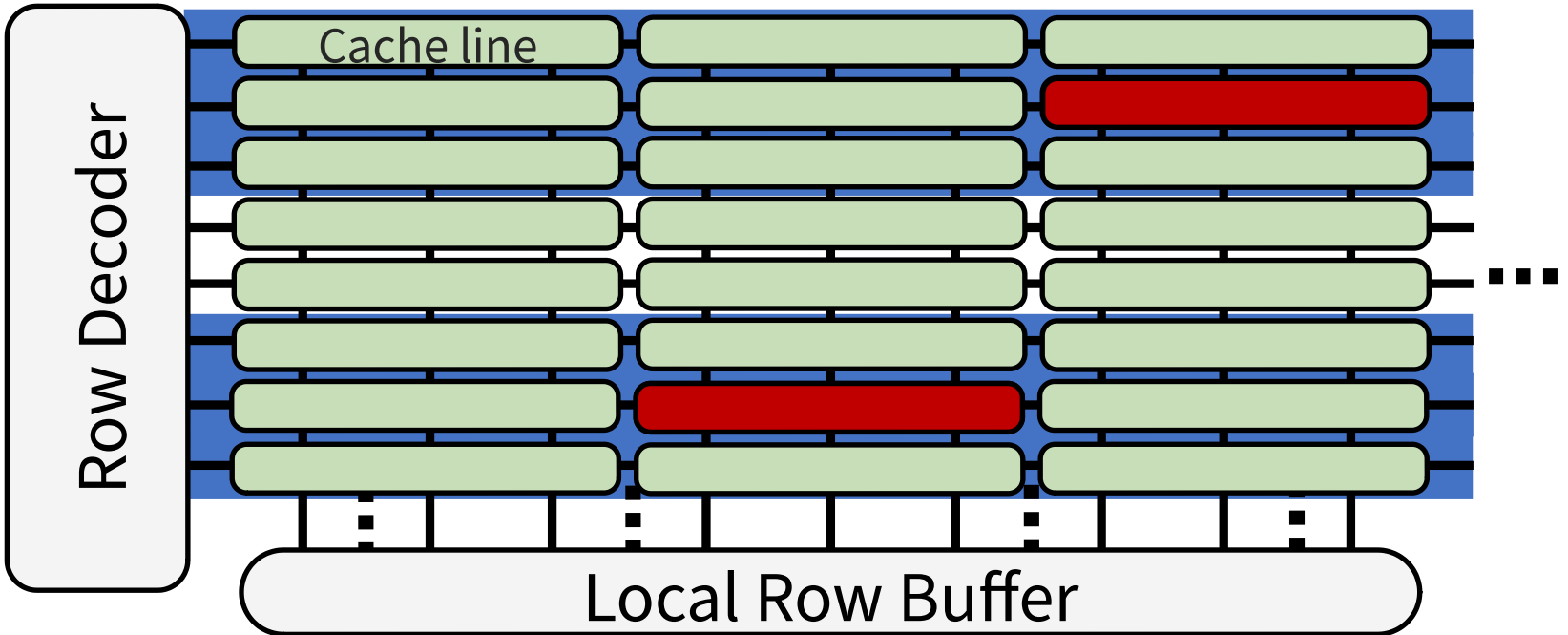
D-RaNGe: Exclusive Access

- **In a bank**, find the **two cache lines** in distinct rows with the most RNG cells
- **Reserve rows containing selected cache lines** exclusively for D-RaNGe to minimize interference



D-RaNGe: Exclusive Access (cont.)

- Write **suitable data pattern** in adjacent rows to **maximize number of RNG cells per cache line**
- **Also reserve adjacent rows** exclusively for D-RaNGe to minimize pattern and read interference



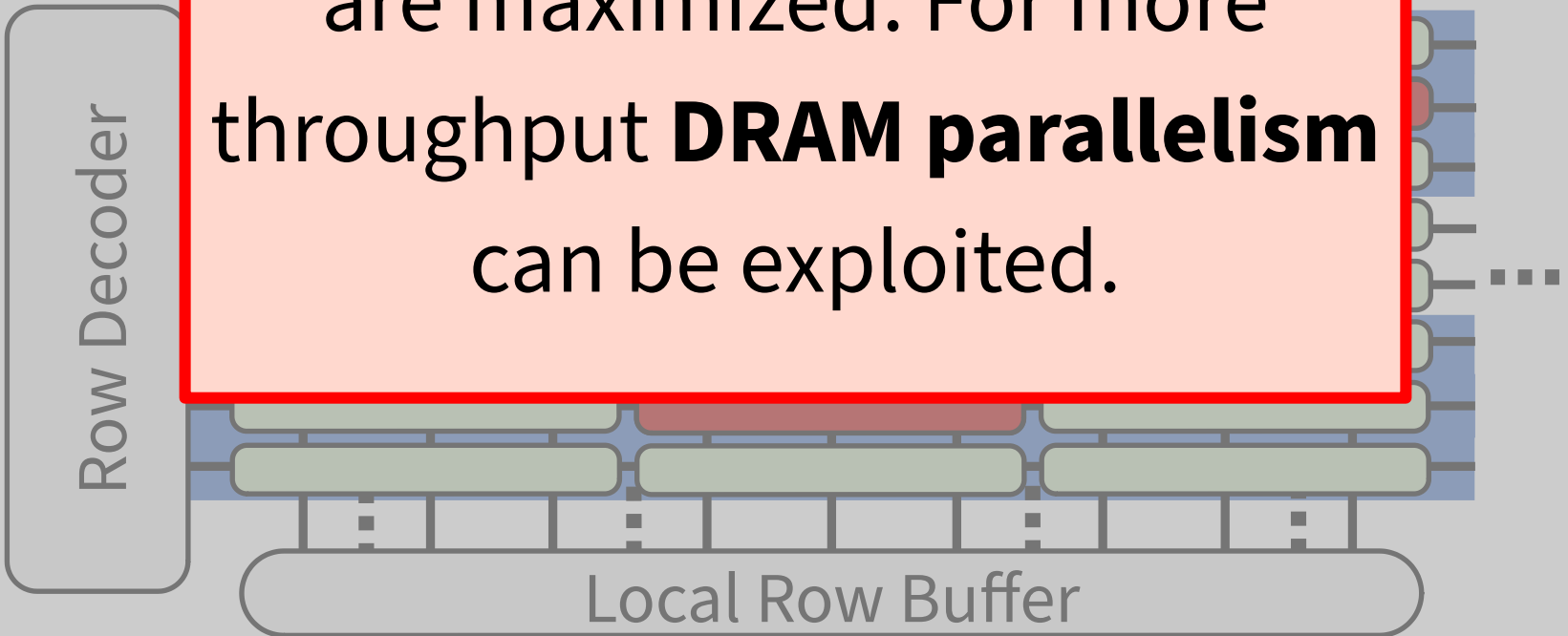
D-RaNGe: Exclusive Access (cont.)

- Write **su**
- **Also res**

er of RNG cells

pattern and read int

Activation failures in a bank are maximized. For more throughput **DRAM parallelism** can be exploited.



D-RaNGe: Implementation

- D-RaNGe can be fully implemented in the **memory controller firmware**
  - Fine-grained control over **DRAM timing parameters** needed (must be provided by hardware)
- Provide a small queue (hardware or software) of harvested random data
- Expose an API to the user
  - New ISA instruction (e.g. Intel **RDRAND**)
  - Request/Receive interface with memory mapped configuration status registers
  - DMA-like access to queue of random data

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- **Results**
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

Results: Non-Determinism

- Verified with NIST statistical test suite [Rukhin+, Tech report, 2001]
  - Passes all tests
- Provides unbiased output
  - No post processing needed
- Minimum cell entropy of 0.9507 bits

NIST Test Name	P-value	Status
monobit	0.675	PASS
frequency_within_block	0.096	PASS
runs	0.501	PASS
longest_run_ones_in_a_block	0.256	PASS
binary_matrix_rank	0.914	PASS
dft	0.424	PASS
non_overlapping_template_matching	>0.999	PASS
overlapping_template_matching	0.624	PASS
maurers_universal	0.999	PASS
linear_complexity	0.663	PASS
serial	0.405	PASS
approximate_entropy	0.735	PASS
cumulative_sums	0.588	PASS
random_excursion	0.200	PASS
random_excursion_variant	0.066	PASS

Results: Non-Determinism

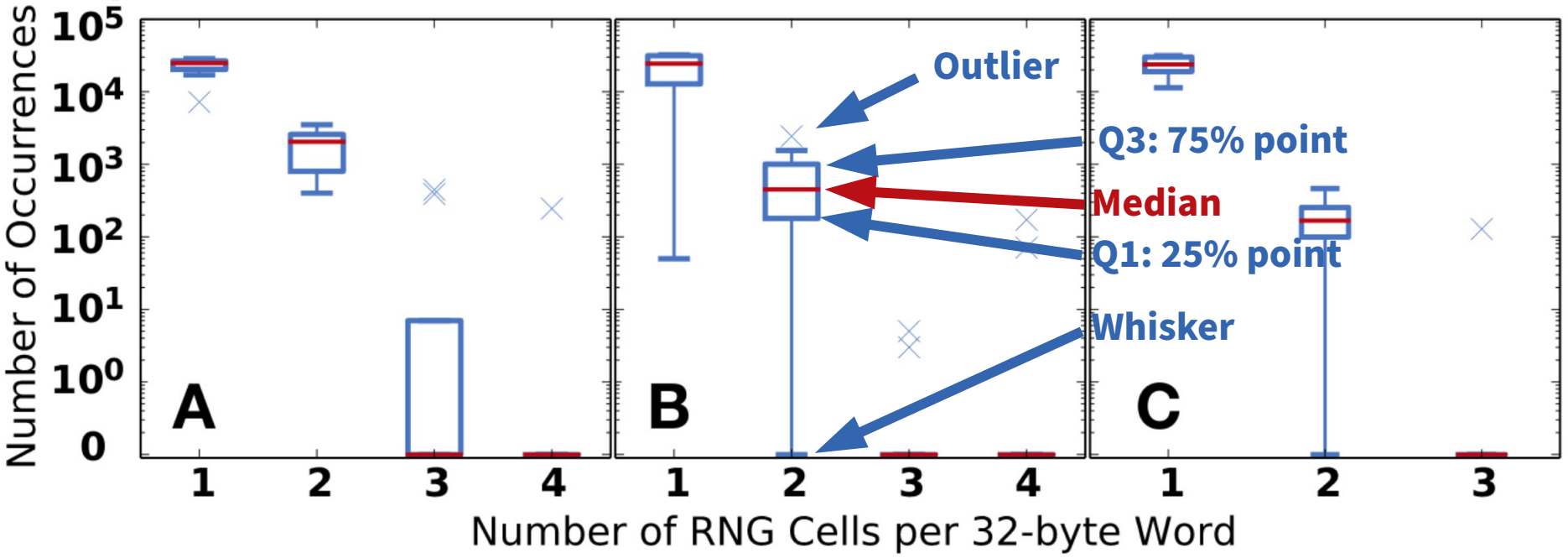
- Ver
  sui
  –
- Pro
  –
- Mi

It is highly likely a **metastable state** is induced in the sense amplifiers.

A non-deterministic physical phenomenon is sampled

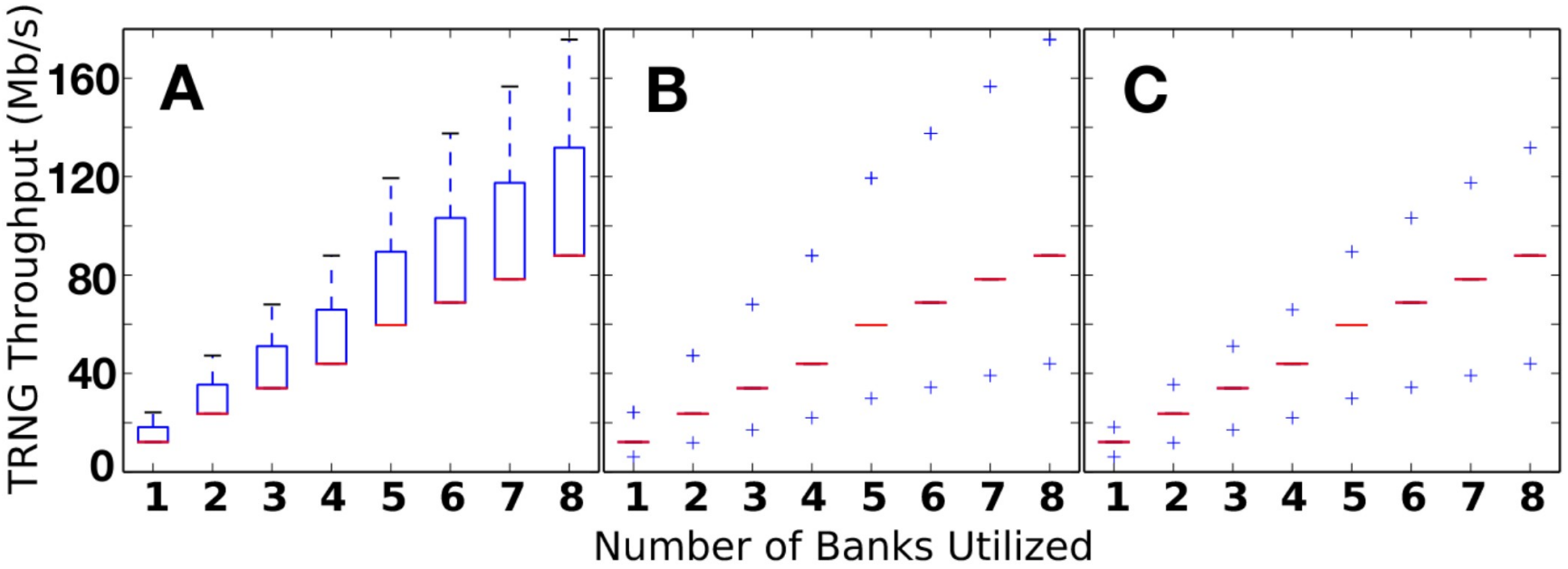
	alue	Status
	675	PASS
	096	PASS
	501	PASS
	256	PASS
	914	PASS
	424	PASS
	.999	PASS
	624	PASS
	999	PASS
	663	PASS
	405	PASS
	735	PASS
	588	PASS
random_excursion	0.200	PASS
random_excursion_variant	0.066	PASS

Results: Latency



- Latency directly related to **density of available RNG cells** per cache line
- **Maxium latency**: **960ns** (1 RNG cell / cache line, 1 bank)
- **Minimum empirical latency**: **100ns** (4 RNG cells /cache line, 32 banks)

Results: Single channel throughput



- Rate simulated in **Ramulator**
- Minimum of **40Mb/s** with all **8 banks**
- Maximum for **A/B/C: 179.4/179.4/134.5 Mb/s**
- 4-channel max (avg) throughput: **717.4 (435.7) Mb/s**

Further Results

- System Interference:
  - Capacity overhead: 6 DRAM rows per bank (**~0.018%**)
  - Flexible level of interference
  - Average throughput with SPEC CPU2006 workloads, issuing D-RaNGe accesses only during **idle** times: **83.1 Mb/s**
- Energy consumption:
  - **4.4 nJ/bit**
- Implementation cost:
  - Can be implemented in **software only**
  - Certain hardware prerequisites (control over DRAM timing parameters)

Further Results

- Syste
  - C
  - Fl
  - Av aNGe
    ac
- Energ
  - **4.**
- Imple
  - C
  - Certain hardware prerequisites (control over DRAM timing parameters)

D-RaNGe implements an

effective TRNG

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- **Comparison to Prior Work**
- Key Contributions
- Strengths, Weaknesses
- Related Work
- Discussion

Prior Work

Proposal	Year	Entropy Source	True Random	Streaming Capable	64-bit TRNG Latency	Energy Consumption	Peak Throughput
Pyo+ [116]	2009	Command Schedule	✗	✓	$18\mu s$	N/A	$3.40Mb/s$
Keller+ [65]	2014	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
Tehranipoor+ [144]	2016	Startup Values	✓	✗	$> 60ns$ (optimistic)	$> 245.9p\mathcal{J}/bit$ (optimistic)	N/A
Sutar+ [141]	2018	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
D-RaNGe	2018	Activation Failures	✓	✓	$100ns < x < 960ns$	$4.4n\mathcal{J}/bit$	$717.4Mb/s$

Table 2: Comparison to previous DRAM-based TRNG proposals.

Command Schedule:

- Does **not** sample **inherently random** phenomenon (lower bits of cylcle timer)
- Best performing proposal before D-RaNGe

Prior Work

Proposal	Year	Entropy Source	True Random	Streaming Capable	64-bit TRNG Latency	Energy Consumption	Peak Throughput
Pyo+ [116]	2009	Command Schedule	✗	✓	$18\mu s$	N/A	$3.40Mb/s$
Keller+ [65]	2014	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
Tehranipoor+ [144]	2016	Startup Values	✓	✗	$> 60ns$ (optimistic)	$> 245.9p\mathcal{J}/bit$ (optimistic)	N/A
Sutar+ [141]	2018	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
D-RaNGe	2018	Activation Failures	✓	✓	$100ns < x < 960ns$	$4.4n\mathcal{J}/bit$	$717.4Mb/s$

Table 2: Comparison to previous DRAM-based TRNG proposals.

Data Retention:

- Inherently **too slow** (latency of 40s)
- **Low throughput** even at high DRAM capacity overhead (0.05 Mb/s at 32GB)
- High energy consumption due long idle periods

Prior Work

Proposal	Year	Entropy Source	True Random	Streaming Capable	64-bit TRNG Latency	Energy Consumption	Peak Throughput
Pyo+ [116]	2009	Command Schedule	✗	✓	$18\mu s$	N/A	$3.40Mb/s$
Keller+ [65]	2014	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
Tehranipoor+ [144]	2016	Startup Values	✓	✗	$> 60ns$ (optimistic)	$> 245.9p\mathcal{J}/bit$ (optimistic)	N/A
Sutar+ [141]	2018	Data Retention	✓	✓	$40s$	$6.8m\mathcal{J}/bit$	$0.05Mb/s$
D-RaNGe	2018	Activation Failures	✓	✓	$100ns < x < 960ns$	$4.4n\mathcal{J}/bit$	$717.4Mb/s$

Table 2: Comparison to previous DRAM-based TRNG proposals.

Startup Values:

- DRAM startup values are random due to interaction between precharge and row decoder logic and column select lines
- Requires a DRAM power cycle to extract the random values
  - Unsuitable for continuous high-throughput operation

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- **Key Contributions**
- Strengths, Weaknesses
- Related Work
- Discussion

Key Contributions

- Novel approach for **extracting true random numbers from commodity DRAM devices** at hight throughput and low latency.
- D-RaNGe uses DRAM cells as entropy sources to generate true random numbers by accessing them with an **access latency lower than manufacturer recommendations**.
- Using experimental data from state of the art LPDDR4 DRAM modules a **rigorous characterization of randomness** in errors induced by **accessing DRAM with low activation latency** is presented.
- The quality of D-RaNGe's bitstream is evaluated using the NIST statistical suite and is found to **pass every test**.
- The performance of D-RaNGe is **compared to other DRAM based TRNG's**.

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- **Strengths, Weaknesses**
- Related Work
- Discussion

Strengths

- Novel mechanism to extract high quality true random data at high performance
- Can be applied to many real world applications
- Very flexible mechanism with respect to system integration and system interference
- Repurposing of widely available hardware to completely different use
- Step towards processing in memory
- Well written paper, good background provided

Weaknesses

- Profiling at different temperatures is necessary
- No consideration of security implications
- Only one DRAM failure mode considered as entropy source
- Profiling overhead is not considered
- Implementation is only easy if hardware supports control over DRAM timing parameters and the firmware of the memory controller can be modified
- Testing only done at temperatures of 40°C to 55°C

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- **Related Work**
- Discussion

Prior work

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case

Donghyuk Le
Samira Khan

Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization

Kevin K. Chang[1] Abhijith Kashyap[1] Hasan Hassan[1,2]
[1] Tianshi Li[1,3]
ur Mutlu[5,1]
versity of Virginia [5]ETH Zürich

Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines

Jeremie

The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim[†§] Minesh Patel[§] Hasan Hassan[§] Onur Mutlu[§†]
[†]Carnegie Mellon University [§]ETH Zürich

Related Work

Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices

Minesh Patel[†] Jeremie S. Kim[‡†] Hasan Hassan[†] Onur Mutlu[†‡]

ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs

Fei Gao Georgios Tziantzioulis David Wentzlaff
wentzlaf@princeton.edu
Department of Electrical Engineering
Princeton University

Exploiting DRAM Latency Variations for Generating True Random Numbers

B. M. S. Bahar Talukder, Joseph Kerns, Biswajit Ray, Thomas Morris, and Md Tauhidur Rahman
Electrical and Computer Engineering Department
University of Alabama in Huntsville, Huntsville, Alabama 35899, USA
Email: {bt0034, jck0012, biswajit.ray, tommy.morris, tauhidur.rahman}@uah.edu

Outline

- Motivation, Problem, Goal
- Background
- Key Idea
- Methodology
- D-RaNGe
- Results
- Comparison to Prior Work
- Key Contributions
- Strengths, Weaknesses
- Related Work
- **Discussion**

Discussion

- Can you think of an improvement for D-RaNGe (i.e., for higher throughput)
  - Combination with other proposals
- Will this problem become more relevant in the future? Is the proposed solution future proof?
- What are possible security implications?
  - Temperature attack
  - RNG implemented in software (potentially modifiable by an adversary)
  - Separate module (channels trusted)