

# QUAC-TRNG

*High-Throughput True Random Number Generation  
Using Quadruple Row Activation in Real DRAM Chips*

**Ataberk Olgun**

Minesh Patel   A. Giray Yağlıkçı   Haocong Luo

Jeremie S. Kim   F. Nisa Bostancı   Nandita Vijaykumar

Oğuz Ergin   Onur Mutlu

**SAFARI**

 **kasirga**

**ETH** zürich



**TOBB ETÜ**  
University of Economics & Technology



UNIVERSITY OF  
**TORONTO**

# Executive Summary

- **Motivation**: DRAM-based true random number generators (TRNGs) provide **true random numbers at low cost** on a **wide range** of computing systems
- **Problem**: Prior DRAM-based TRNGs are slow:
  1. Based on fundamentally slow processes → **high latency**
  2. Cannot effectively harness entropy from DRAM rows → **low throughput**
- **Goal**: Develop a **high-throughput** and **low-latency** TRNG that uses **commodity DRAM** devices
- **Key Observation**: Carefully engineered sequence of DRAM commands can activate **four DRAM rows** → **QUadruple ACtivation (QUAC)**
- **Key Idea**: Use **QUAC** to activate DRAM rows that are initialized with **conflicting data** (e.g., two '1's and two '0's) to generate random values
- **QUAC-TRNG**: DRAM-based TRNG that generates true random numbers at **high-throughput** and **low-latency** by **repeatedly performing QUAC operations**
- **Results**: We evaluate QUAC-TRNG using **136** real DDR4 chips
  1. **5.4 Gb/s** maximum (**3.4 Gb/s** average) TRNG throughput per DRAM channel
  2. Outperforms existing DRAM-based TRNGs by **15.08x** (base), and **1.41x** (enhanced)
  3. Low TRNG latency: **256-bit RN** in **274 ns**
  4. Passes **all 15** NIST randomness tests

# Outline

## **True Random Numbers in DRAM**

DRAM Organization and Operation

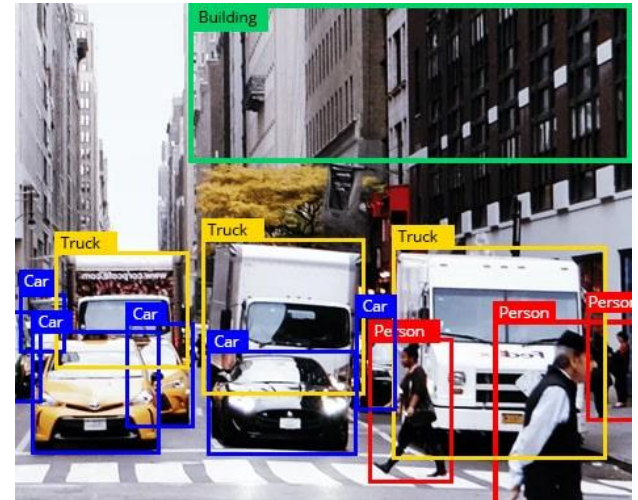
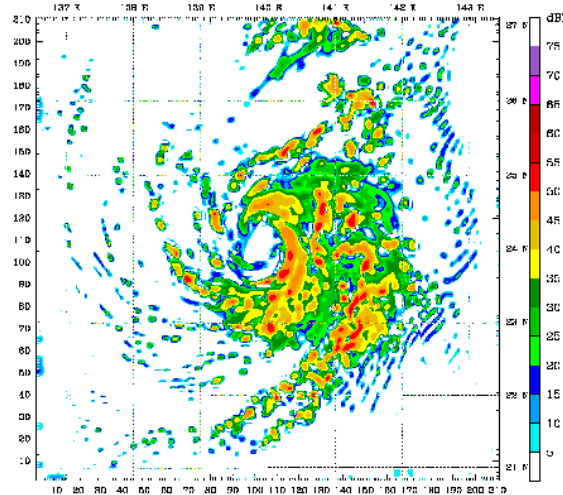
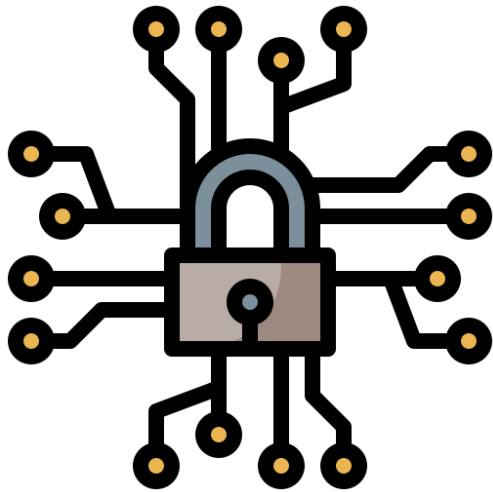
QUadruple ACtivation (QUAC)

QUAC-TRNG

Evaluation

# Use Cases of True Random Numbers

High-quality true random numbers  
are **critical** to **many applications**



True random numbers can **only** be obtained  
by sampling random physical processes

Unfortunately, **not all computing systems** are equipped  
with **TRNG hardware** (e.g., dedicated circuitry)

# DRAM-Based TRNGs

DRAM chips are ubiquitous in modern computing platforms

DRAM-based TRNGs enable true random number generation within DRAM chips

**Low-cost:** No specialized circuitry for RNG

- Beneficial for constrained systems

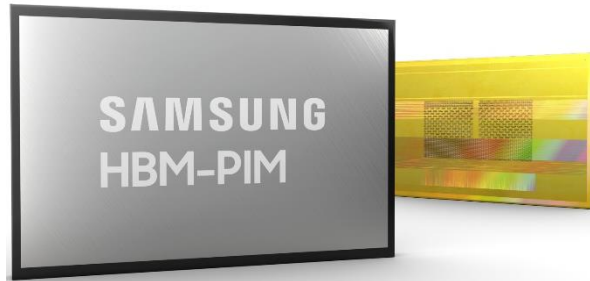
**High-throughput:** > Gb/s throughput

- Open application space that require high-throughput TRNG

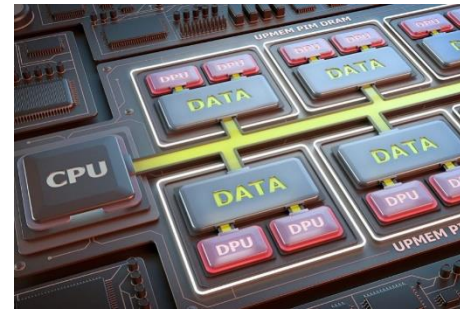
# Synergy with Processing-in-Memory

## Processing-in-Memory (PIM) Systems

- Perform **computation** directly **within** a **memory chip**
- Improve system performance by avoiding off-chip **data movement**



[Samsung]



[UPMEM]

## True random number generation within DRAM

- Enables PIM workloads to sample true random numbers **directly within the memory chip**
- Avoids inefficient communication to other possible **off-chip TRNG sources**, **enhances security & privacy**

# Outline

True Random Numbers in DRAM

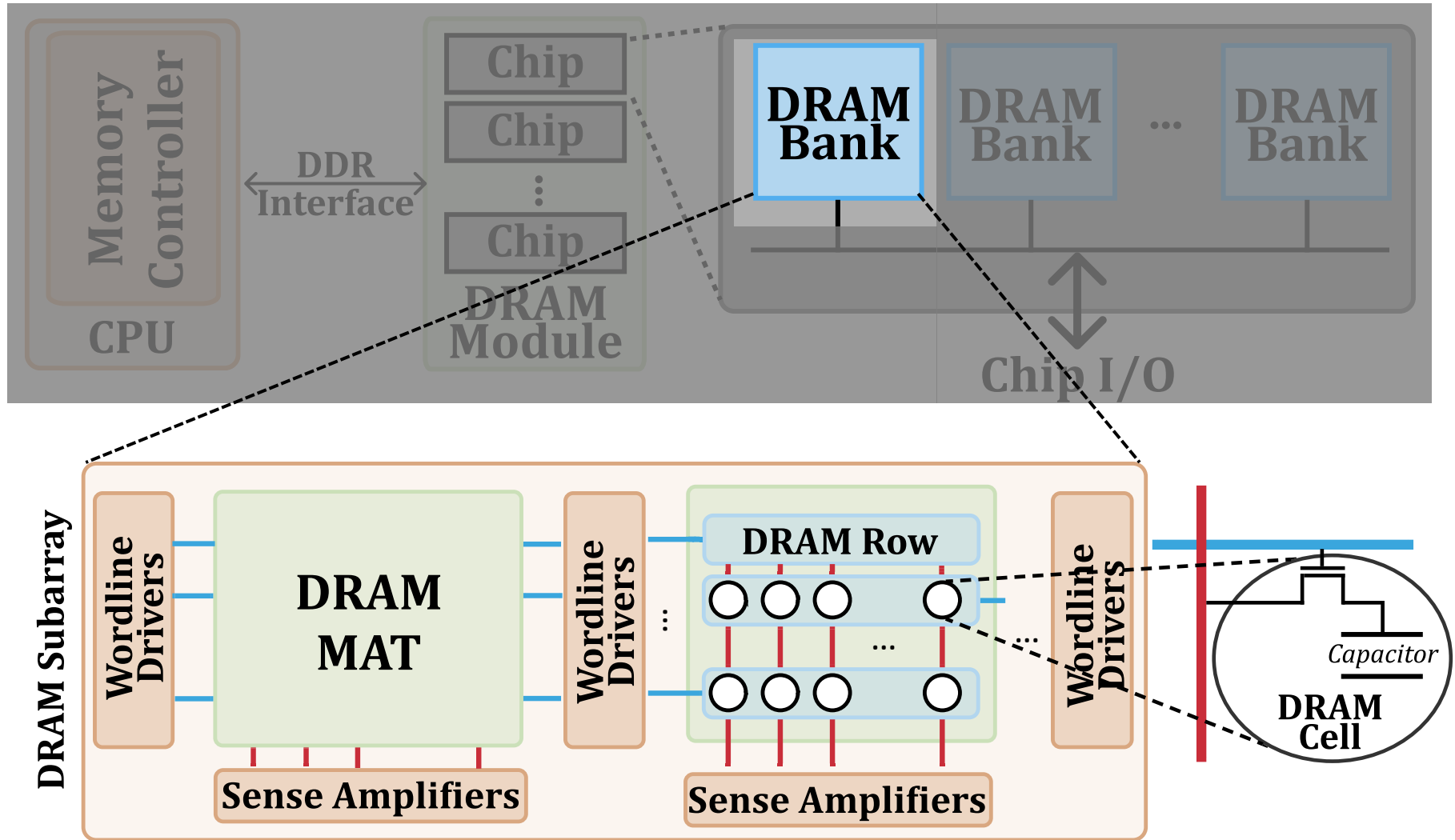
**DRAM Organization and Operation**

QUadruple ACtivation (QUAC)

QUAC-TRNG

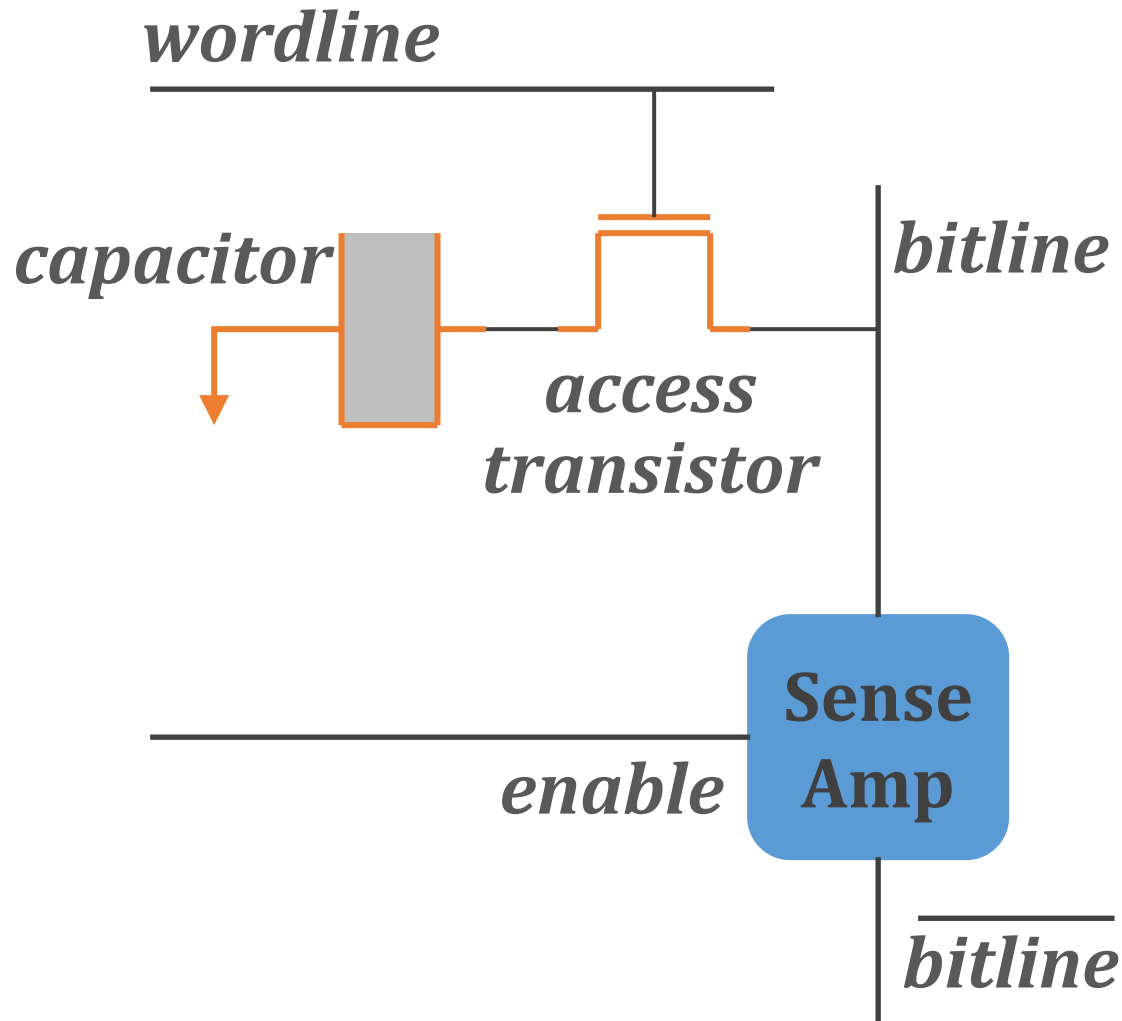
Evaluation

# DRAM Organization

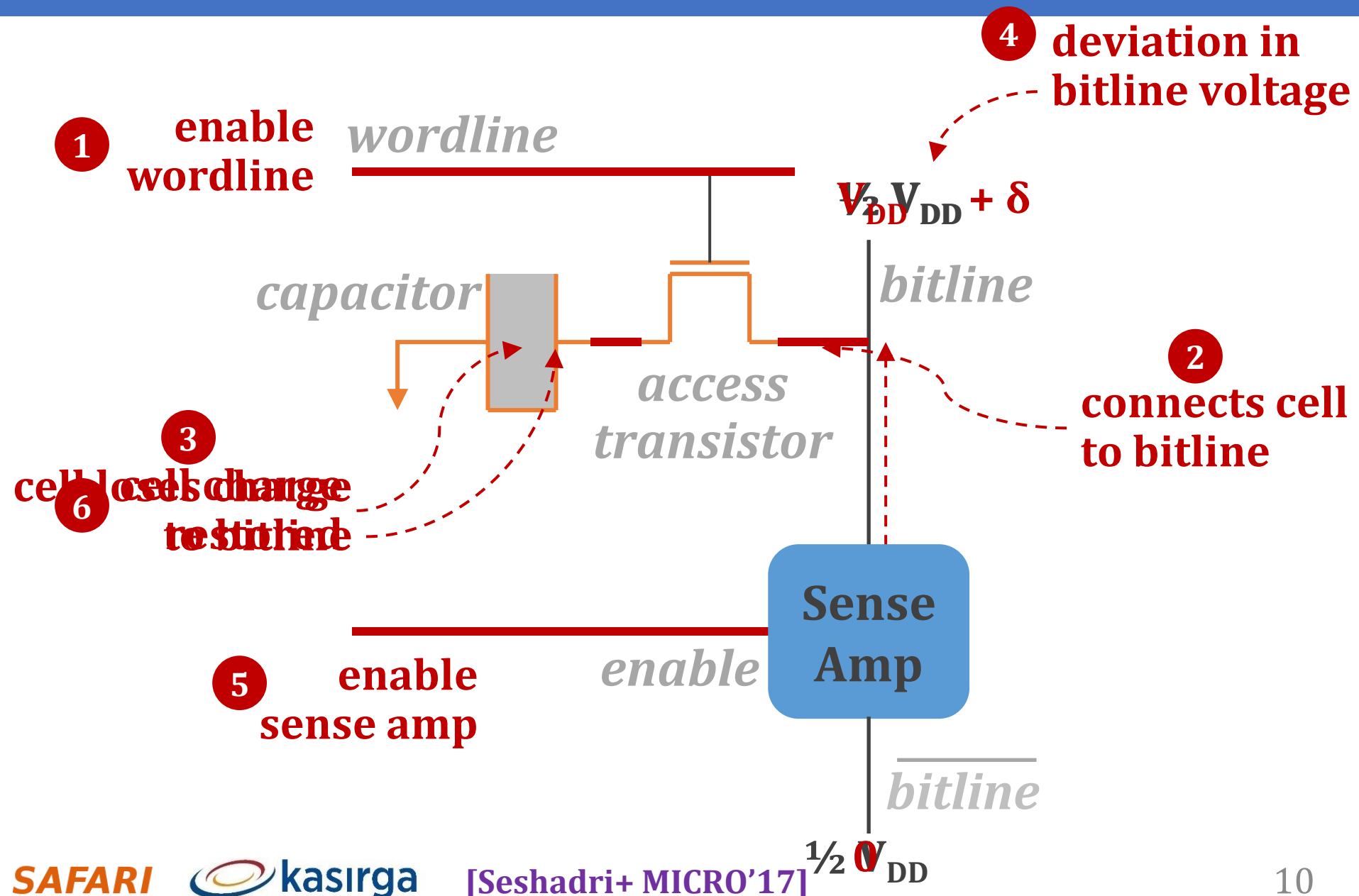




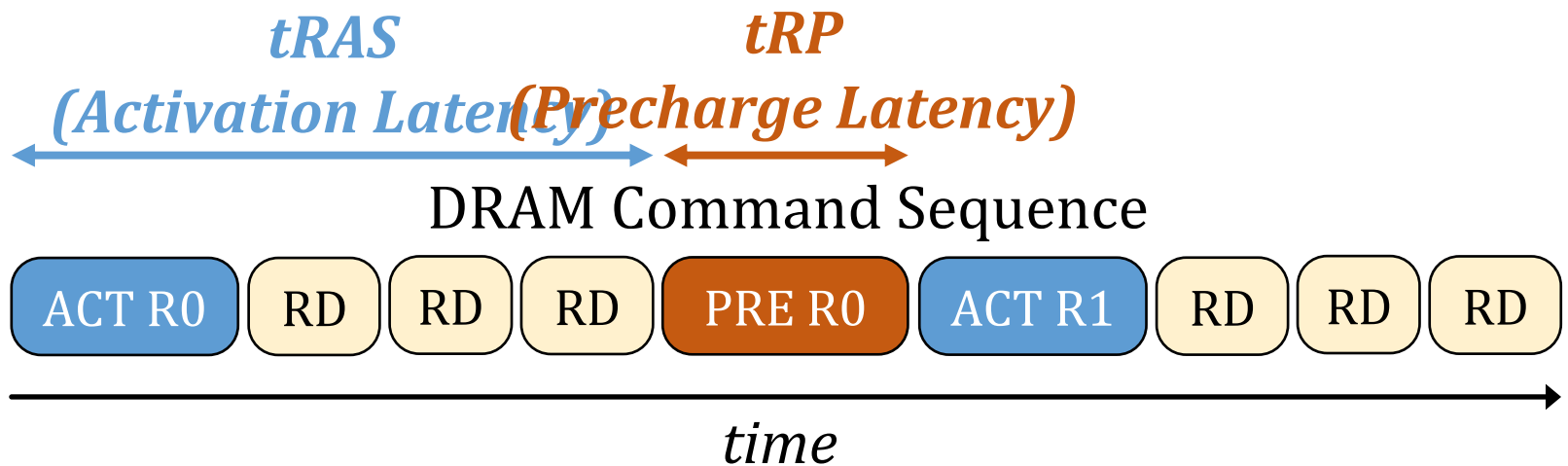
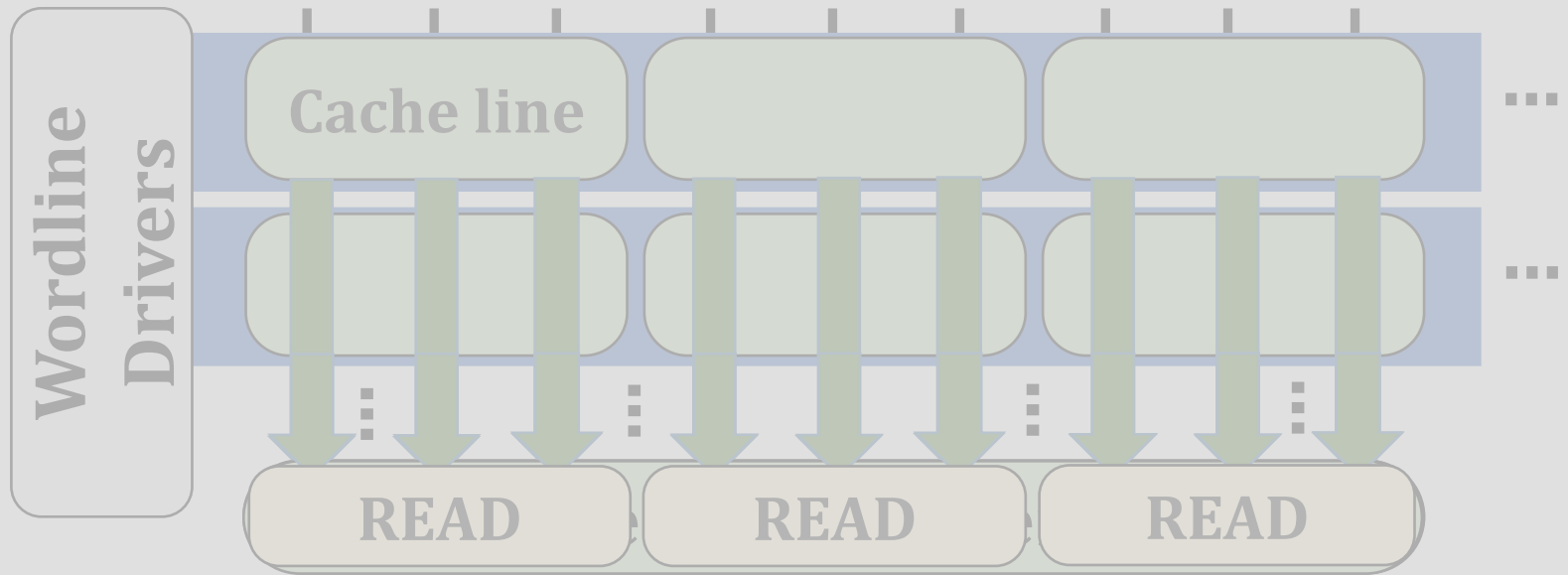
# Accessing a DRAM Cell



# Accessing a DRAM Cell



# DRAM Operation



# Outline

True Random Numbers in DRAM

DRAM Organization and Operation

**QUadruple ACtivation (QUAC)**

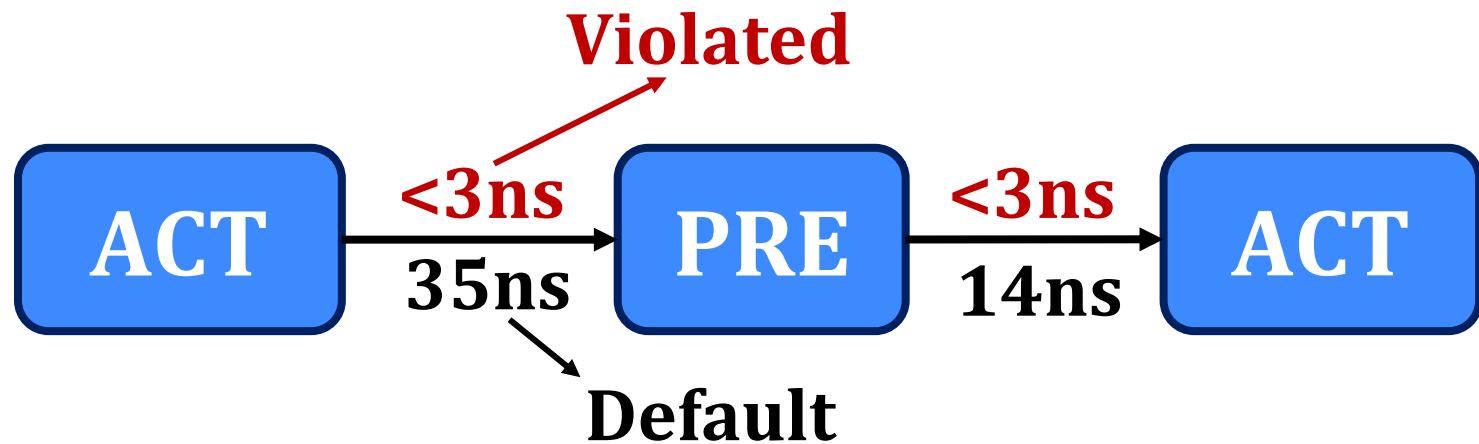
QUAC-TRNG

Evaluation

# Quadruple Activation (QUAC)

## New Observation

Carefully-engineered DRAM commands can activate four rows in real DRAM chips

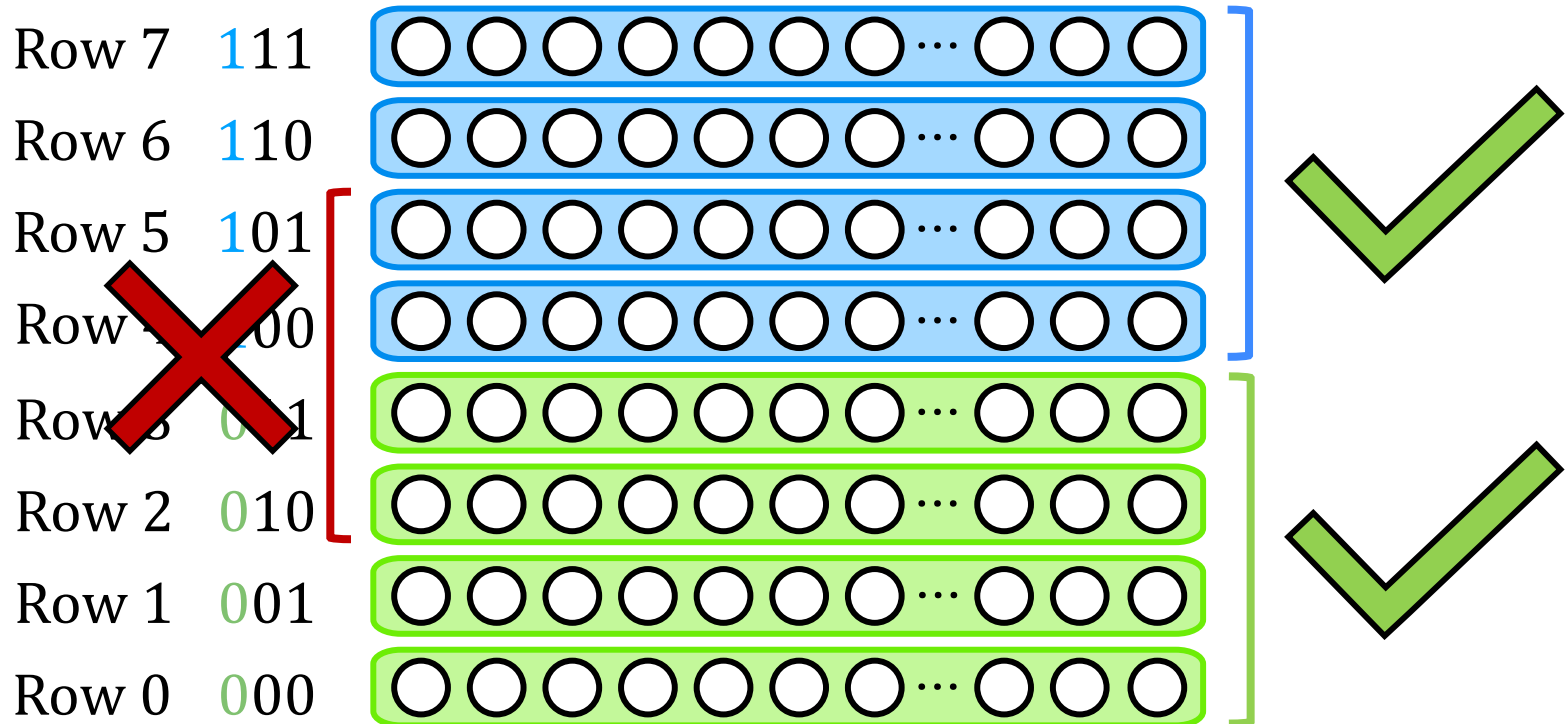


Activate four rows with two ACT commands

# Quadruple Activation (QUAC)

## Characteristic 1

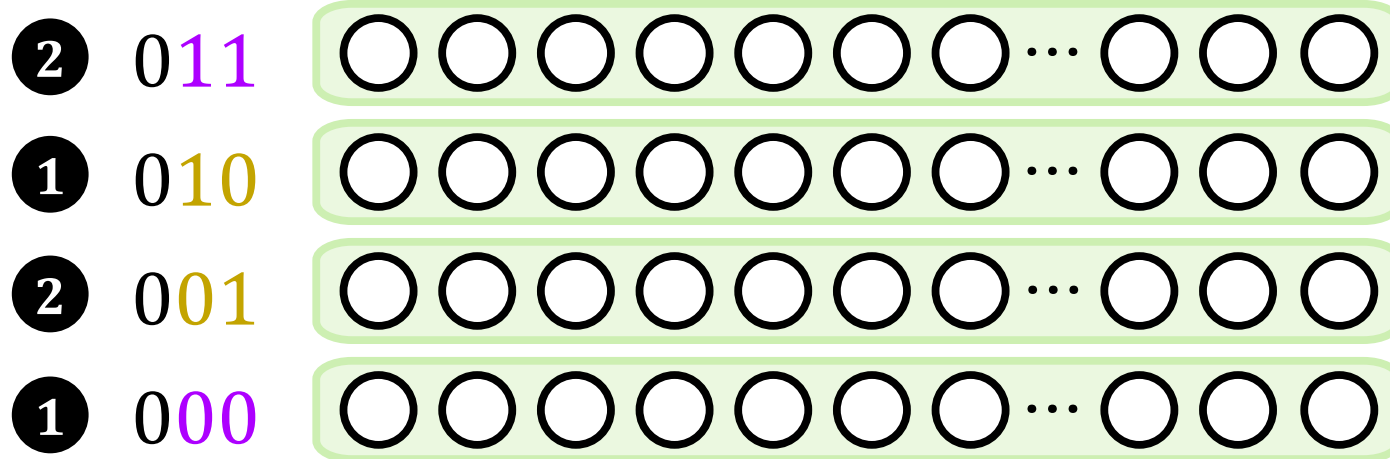
Activates a set of four DRAM rows whose addresses **differ only** in their **two LSBs**



# Quadruple Activation (QUAC)

## Characteristic 2

First and second ACT's addresses must have their **two LSBs inverted**



# QUAC on Real DRAM Chips

b. FPGA Board

a. DRAM Module

c. PCIe Host Interface

d. Temperature Controller

Valid QUAC behavior on  
**136 DDR4 chips**



# Why Does QUAC Work?

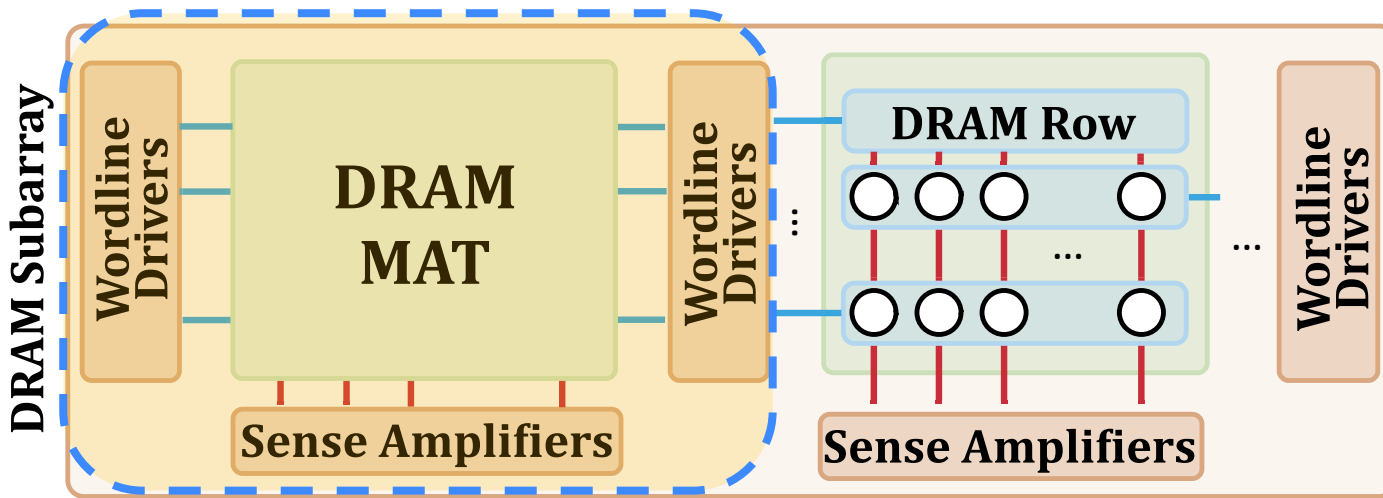
## Hypothetical circuit to explain QUAC

### Hierarchical Wordlines

- High *density* and *performance* requirements
- Hierarchical organization of DRAM wordlines *enable high-density* and *low-latency* DRAM operation

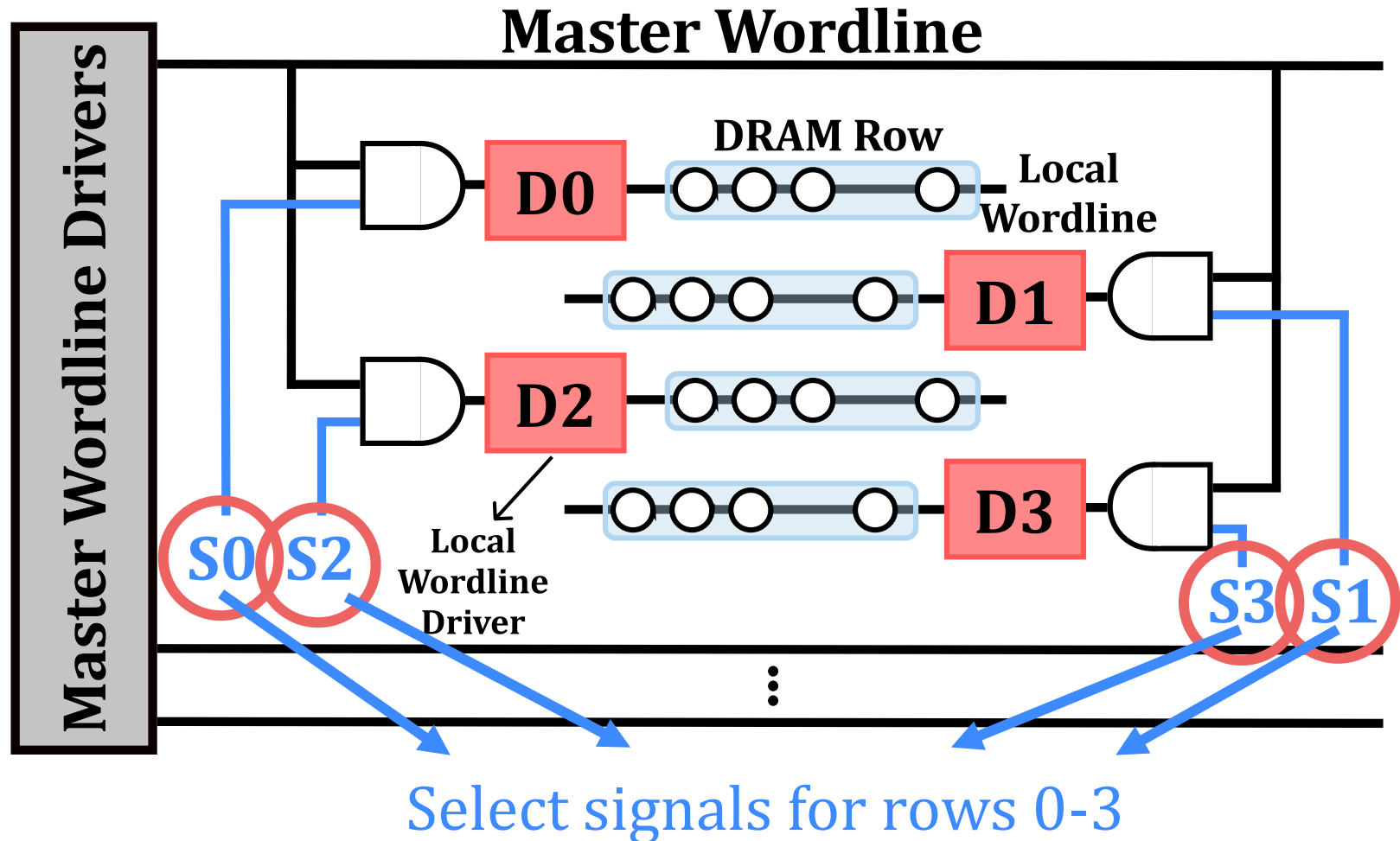
# Hierarchical Wordlines

A **master** wordline drives multiple **local** wordlines



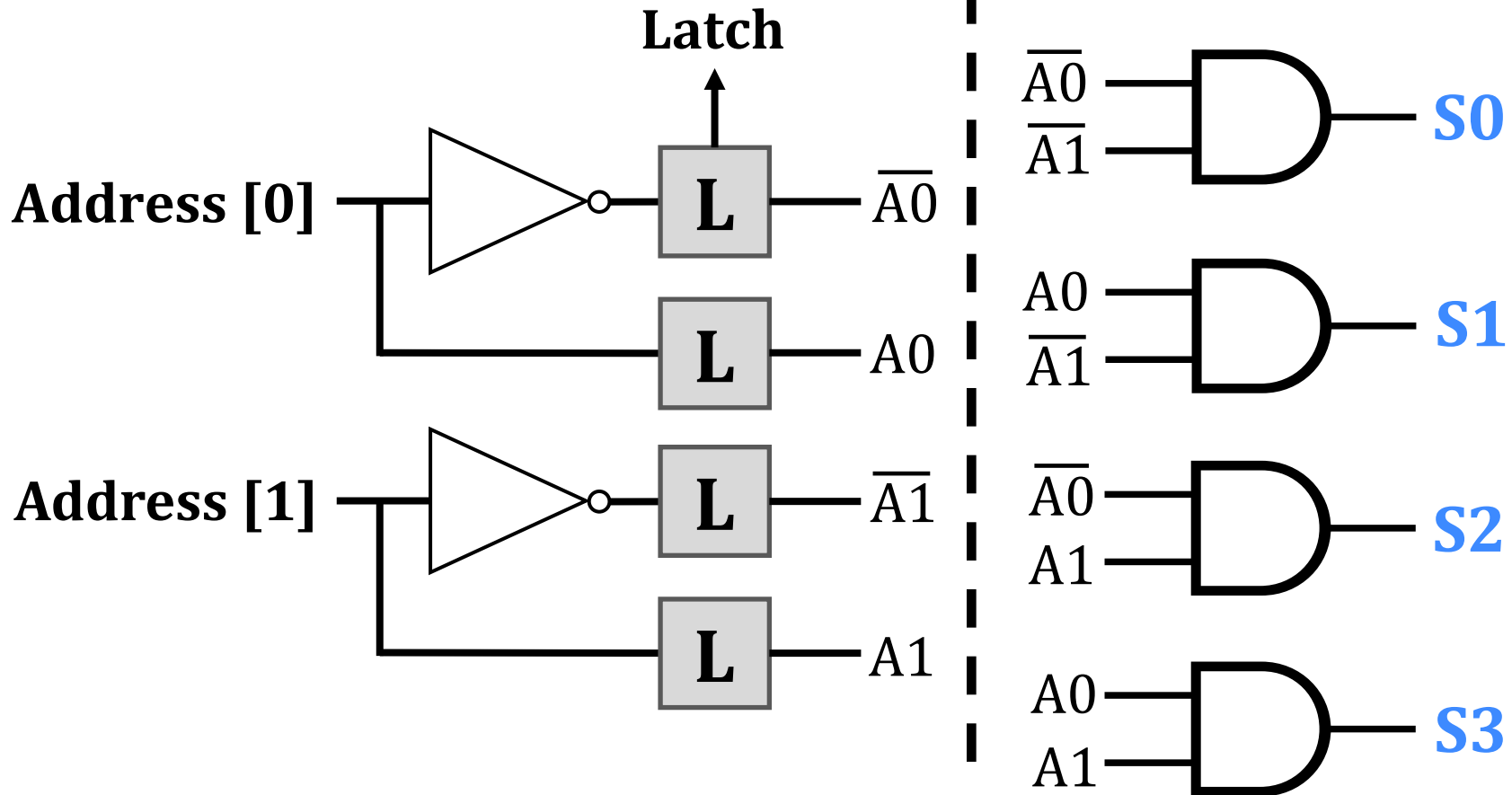
# Hierarchical Wordlines

A **master** wordline drives multiple **local** wordlines

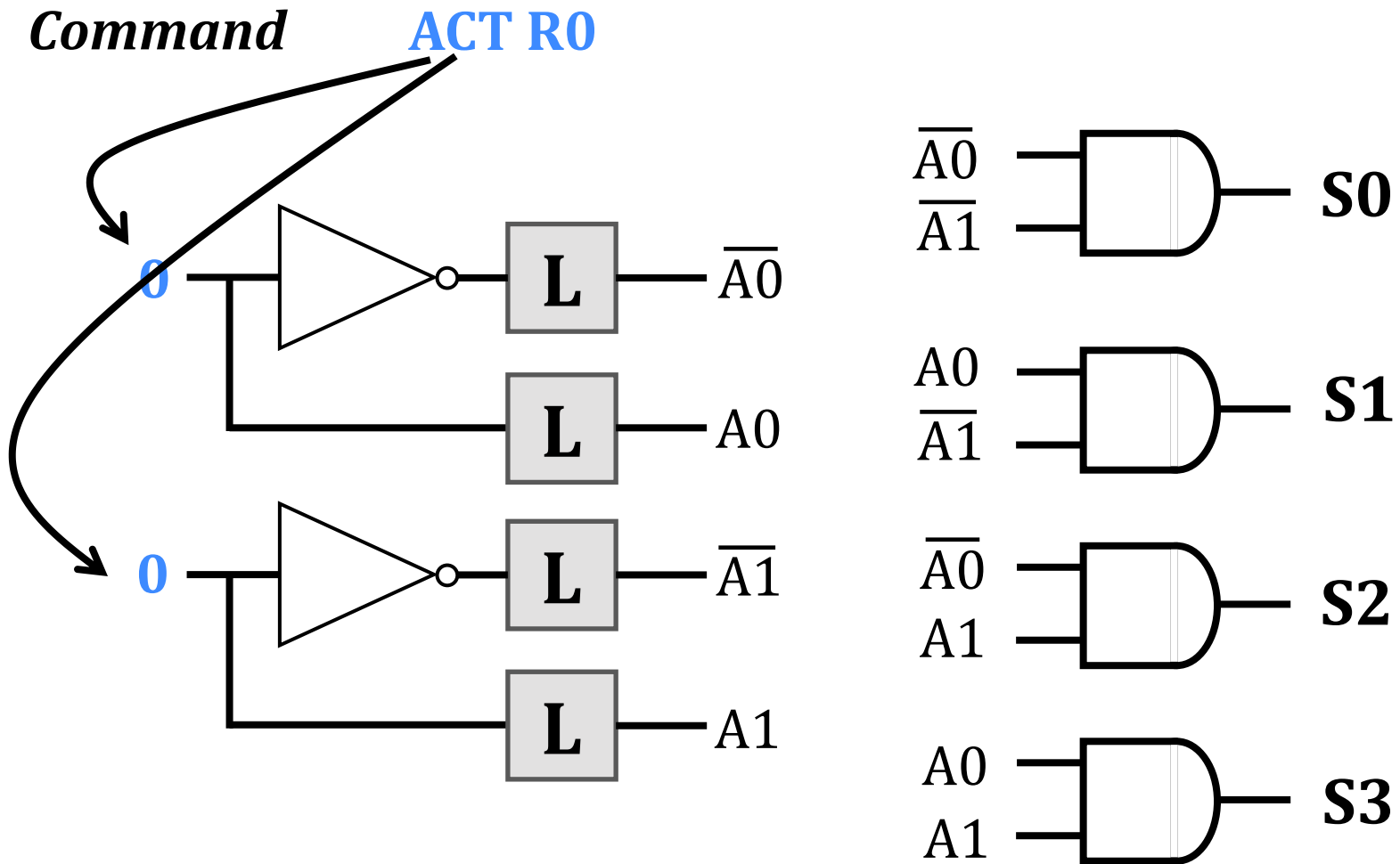


# Hypothetical Row Decoder

*Predecode the least significant two bits* | *Drive control signals*



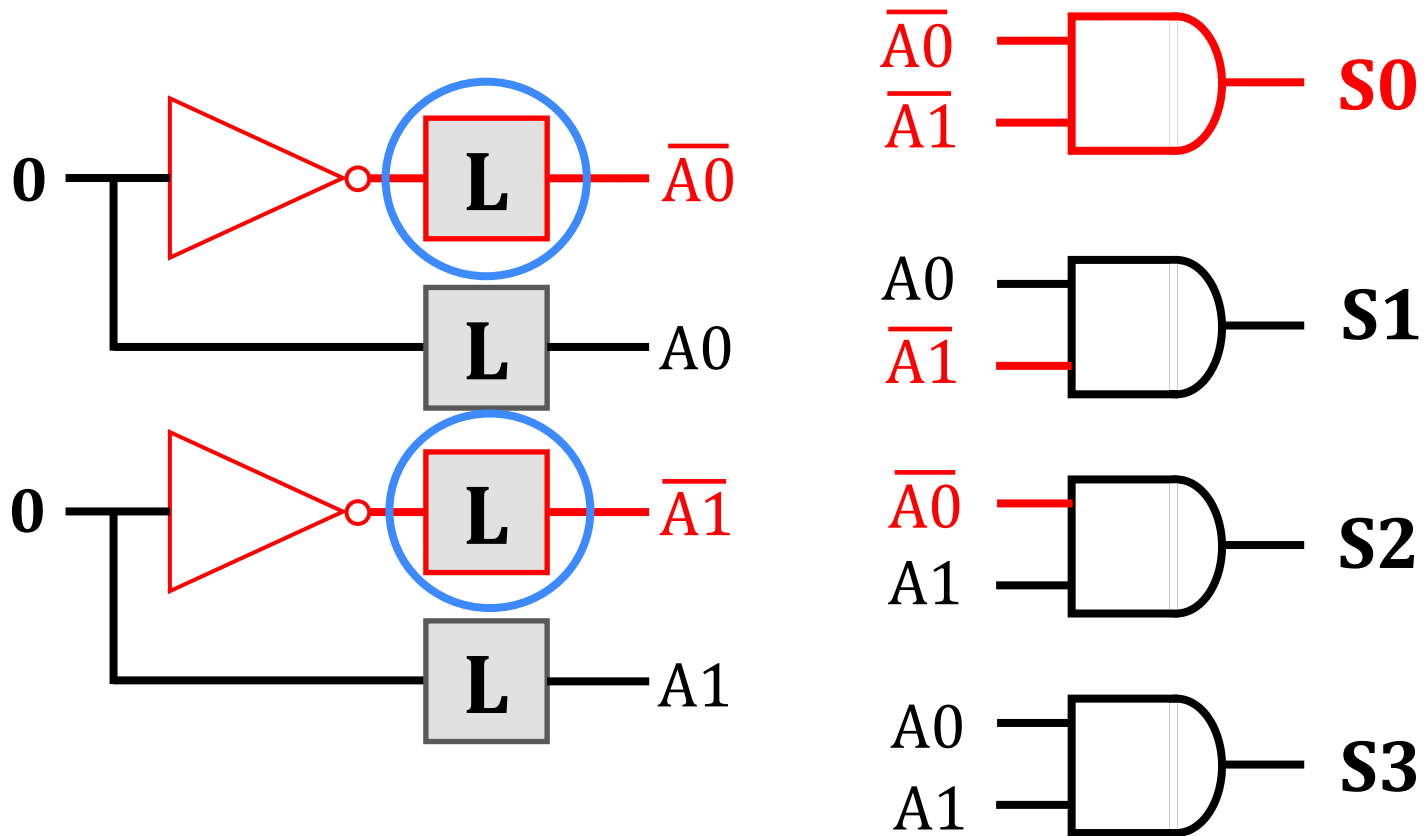
# Hypothetical Row Decoder



First ACT command drives a single wordline

# Hypothetical Row Decoder

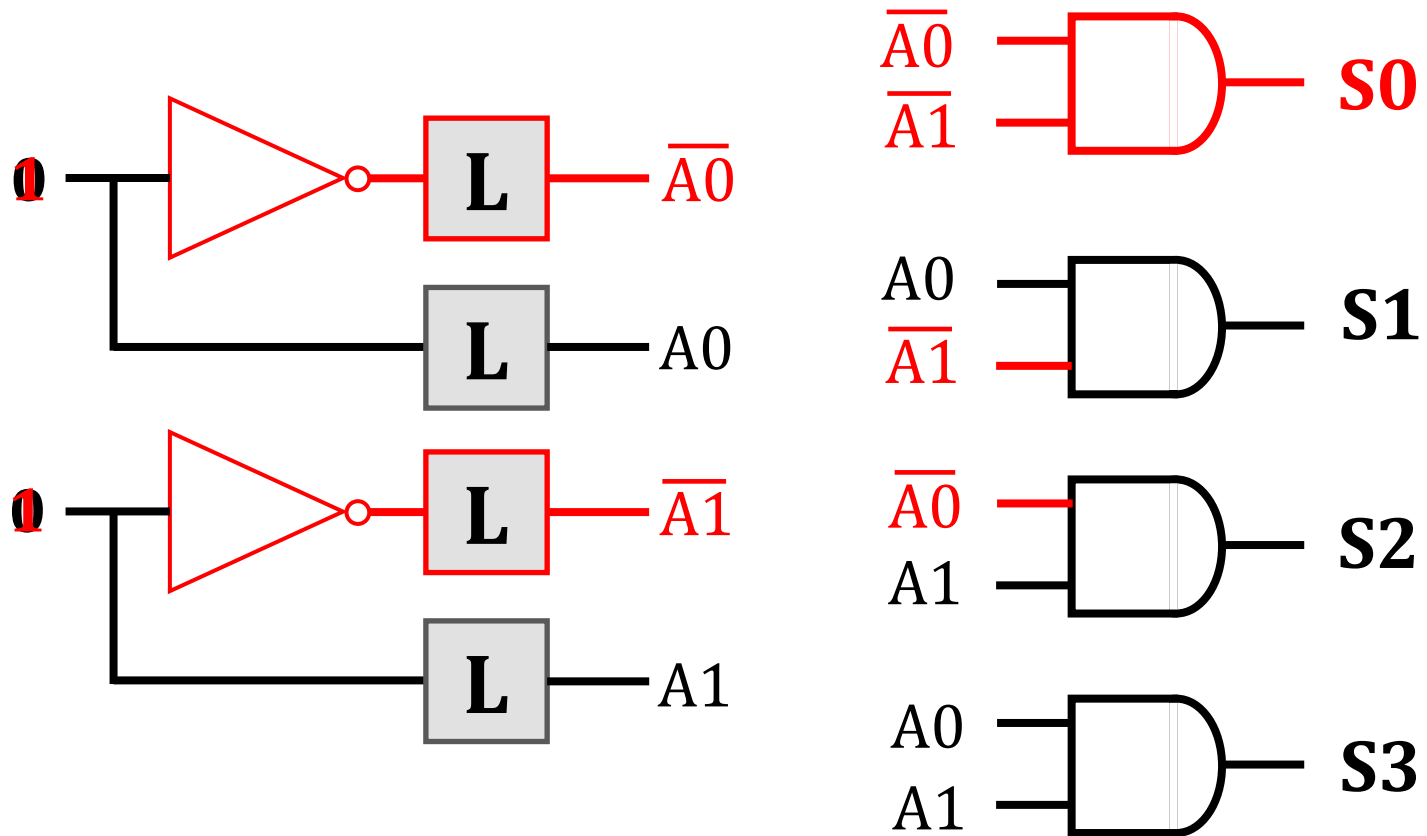
*Command*      ACT R0  $\xrightarrow[\text{Timing}]{\text{Violate}}$  PRE



PRE command **cannot** disable latches

# Hypothetical Row Decoder

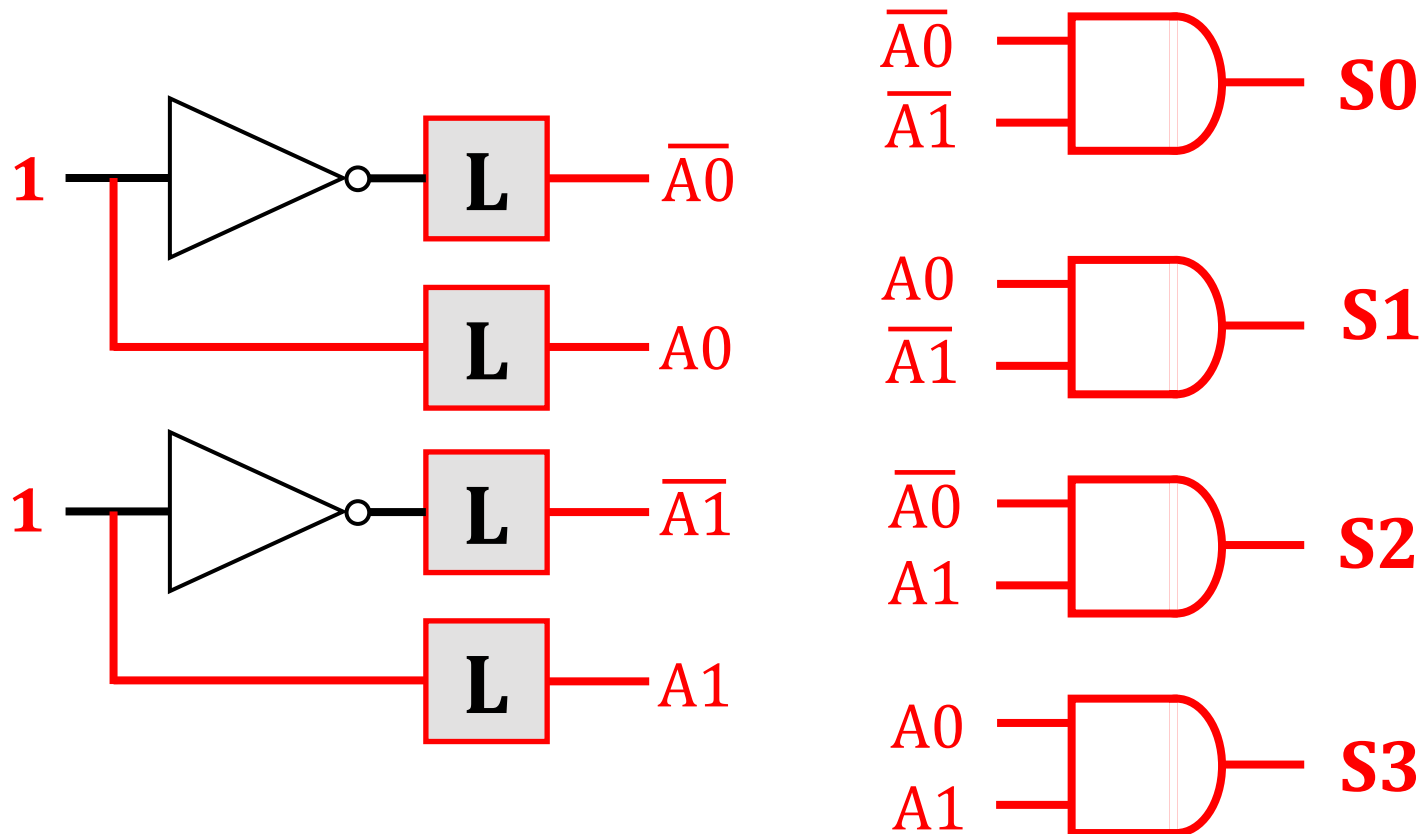
*Command*      ACT R0  $\xrightarrow{\text{Violate Timing}}$  PRE  $\xrightarrow{\text{Violate Timing}}$  ACT R3



Second ACT drives the **remaining three** wordlines

# Hypothetical Row Decoder

*Command*      ACT R0  $\xrightarrow[\text{Timing}]{\text{Violate}}$  PRE  $\xrightarrow[\text{Timing}]{\text{Violate}}$  ACT R3



**All four wordlines are enabled**  
Quadruple Activation



# Outline

True Random Numbers in DRAM

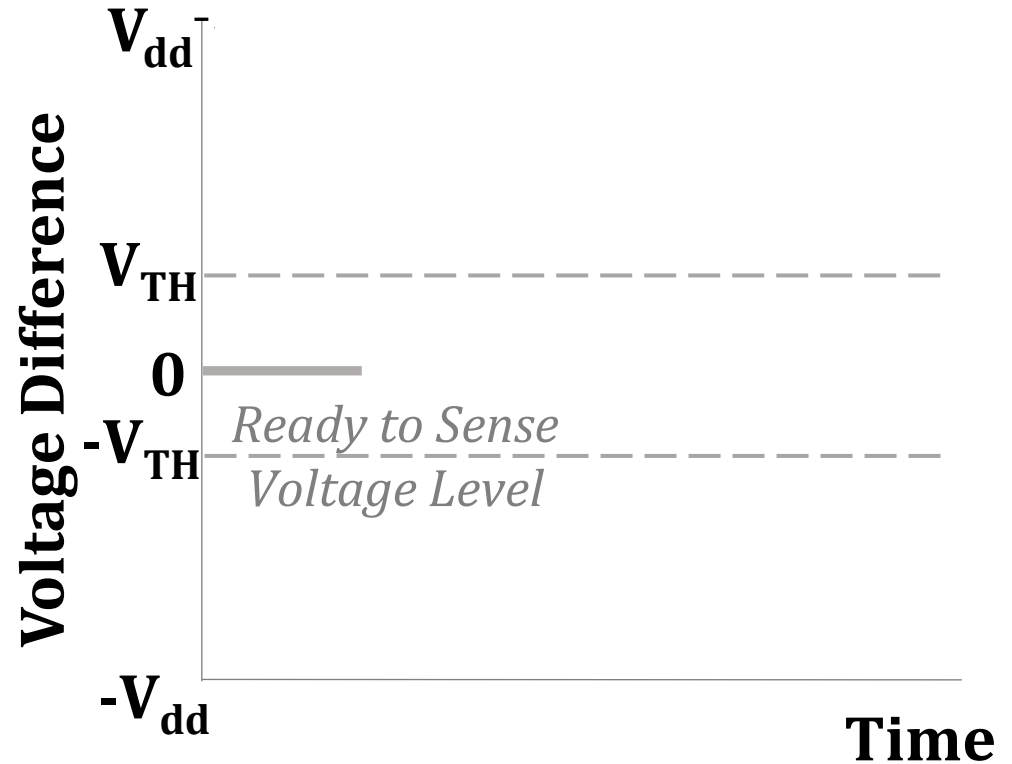
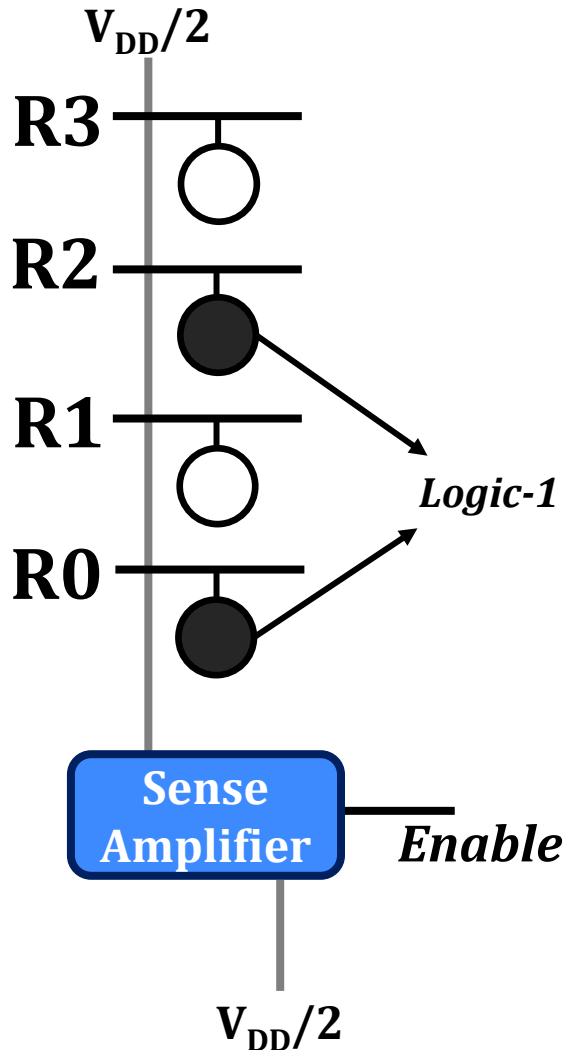
DRAM Organization and Operation

QUadruple ACtivation (QUAC)

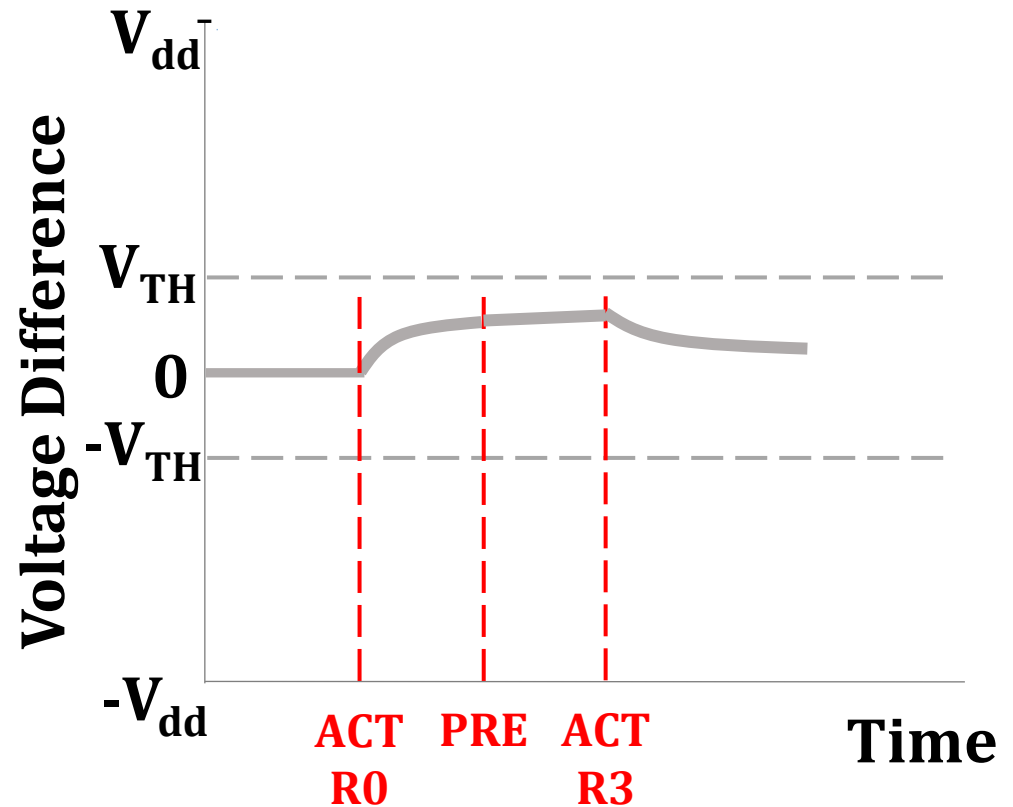
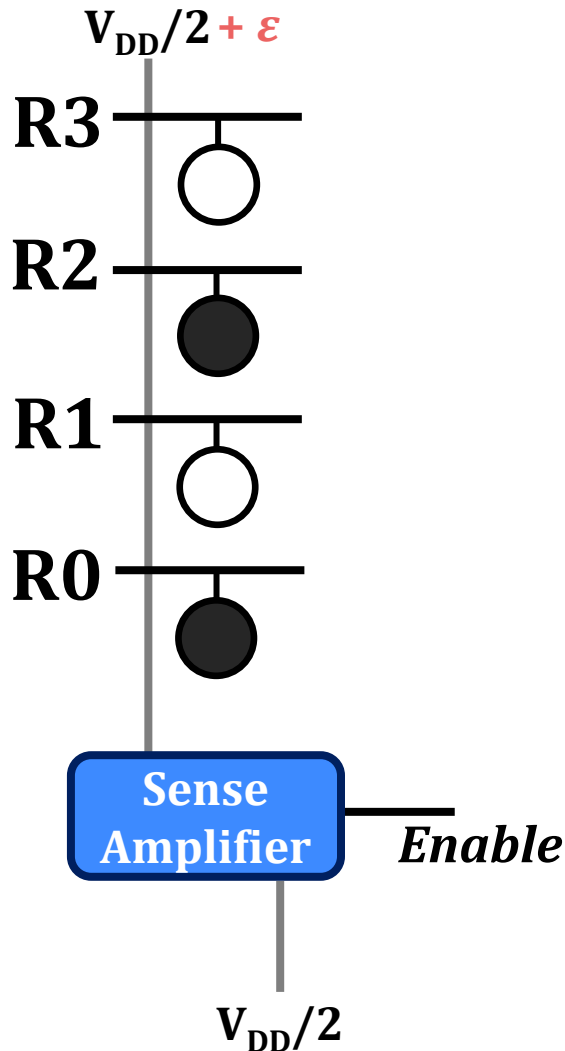
**QUAC-TRNG**

Evaluation

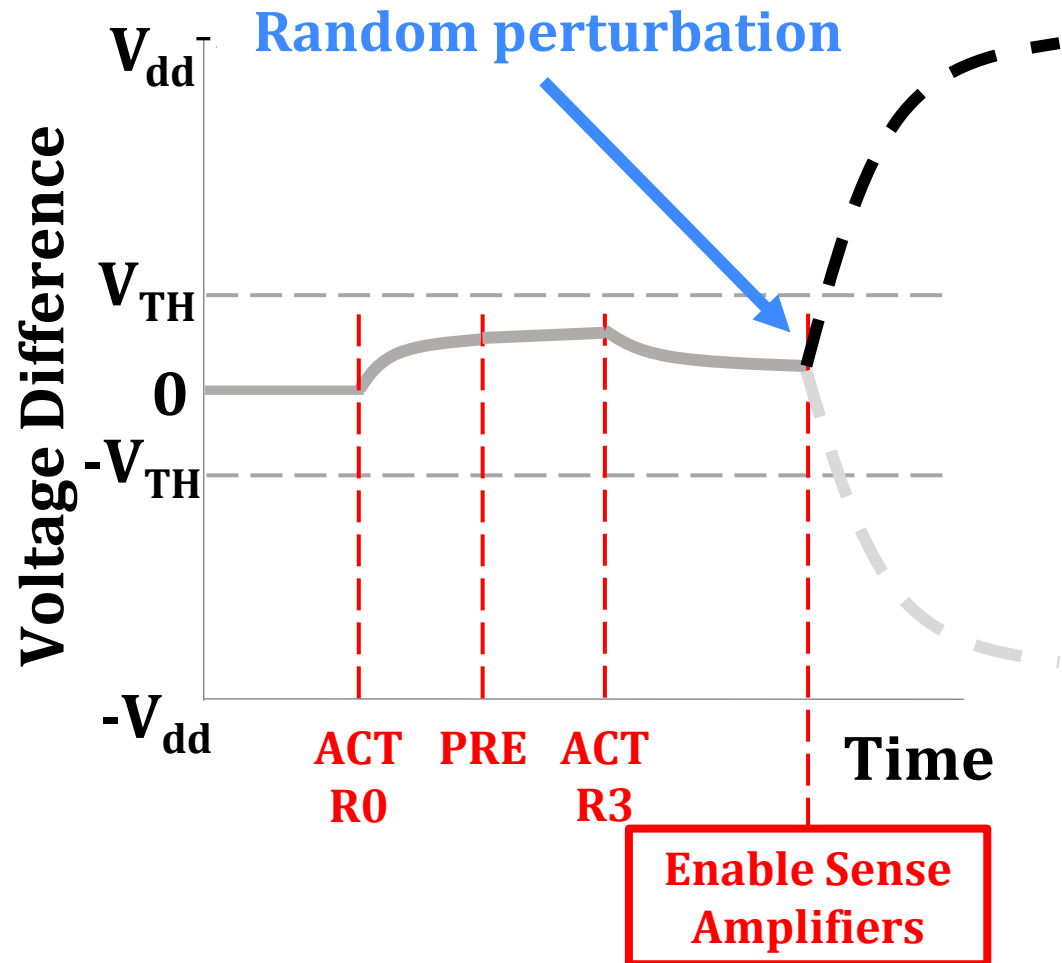
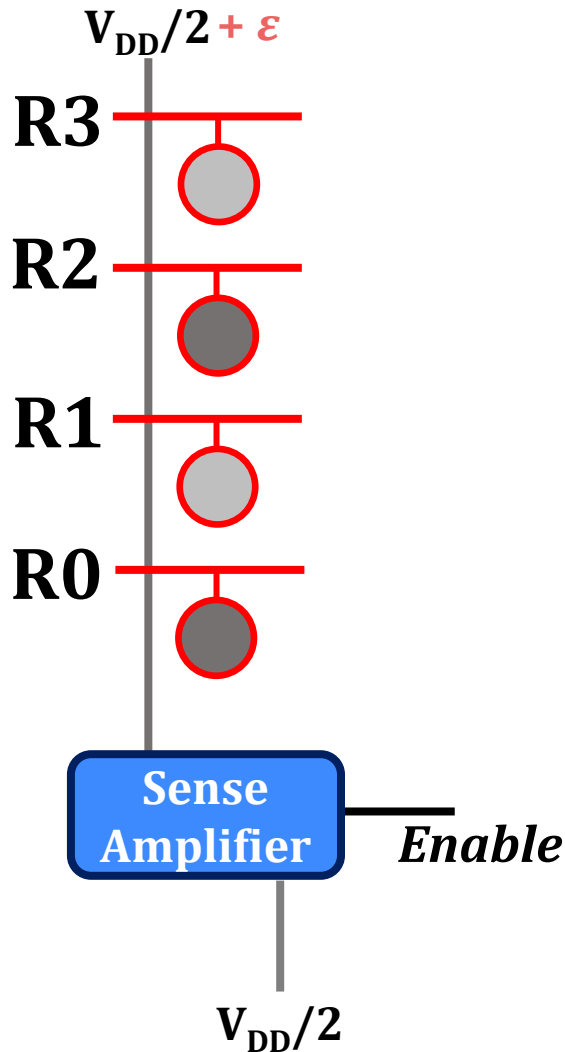
# Generating Random Values via QUAC



# Generating Random Values via QUAC

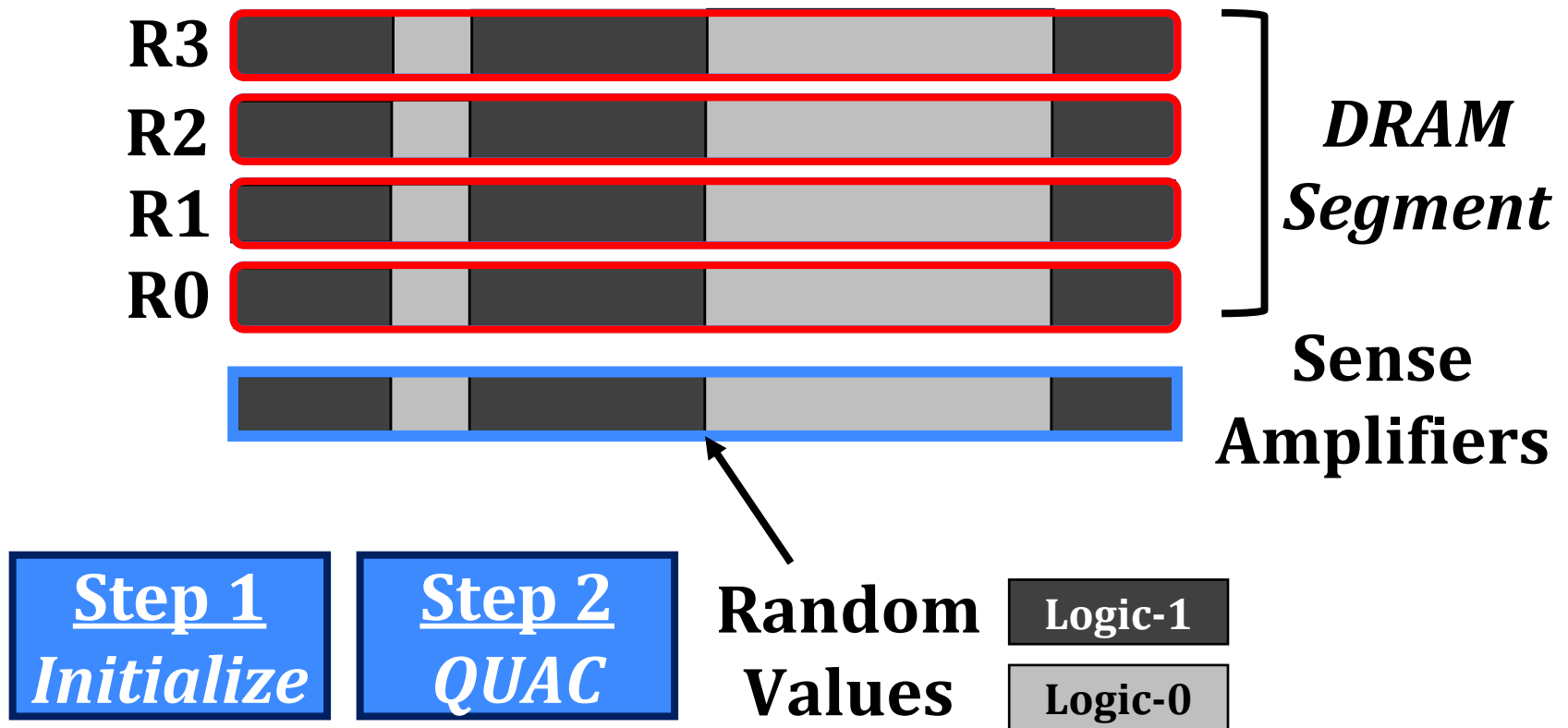


# Generating Random Values via QUAC



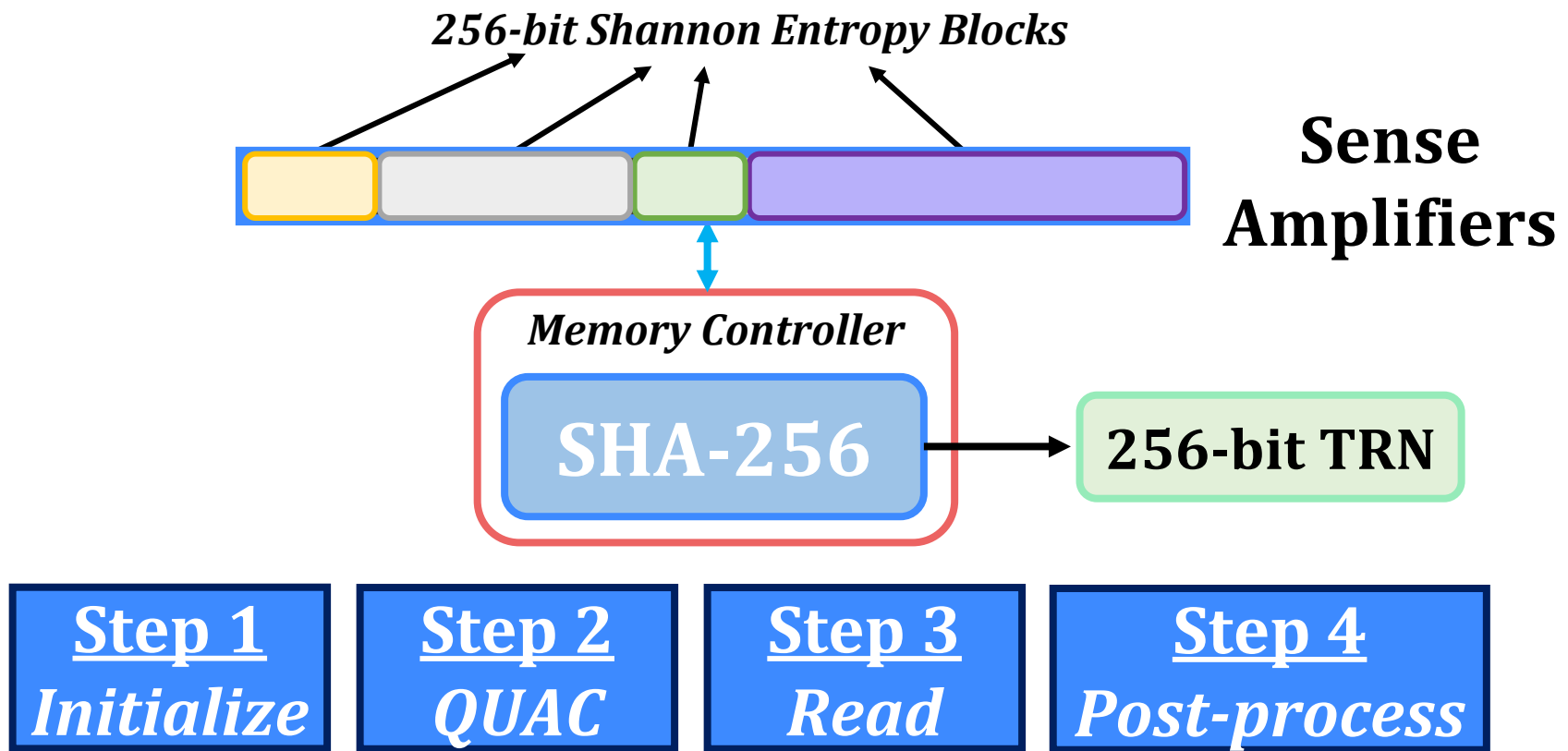
# QUAC-TRNG

**Key Idea:** Leverage **random values** on sense amplifiers generated by **QUAC** operations as **source of entropy**



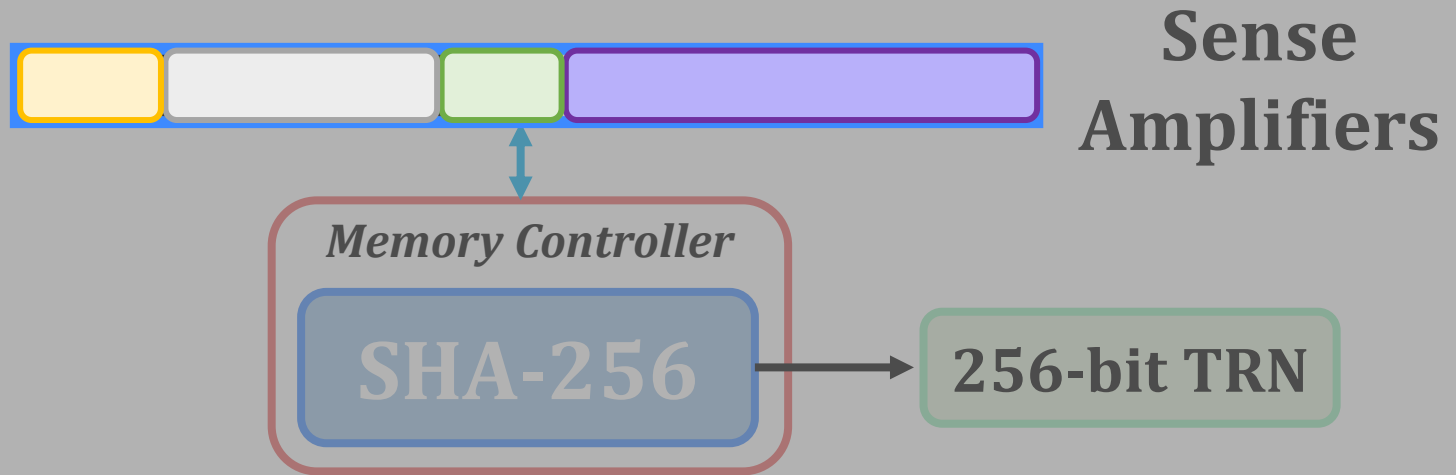
# QUAC-TRNG

**Key Idea:** Leverage **random values** on sense amplifiers generated by **QUAC** operations as **source of entropy**



# QUAC-TRNG

**Key Idea:** Leverage random values on sense amplifiers generated by QUAC operations as source of entropy



Generates a 256-bit random number  
for every 256-bit Shannon Entropy block

# Outline

True Random Numbers in DRAM

DRAM Organization and Operation

QUadruple ACtivation (QUAC)

QUAC-TRNG

**Evaluation**

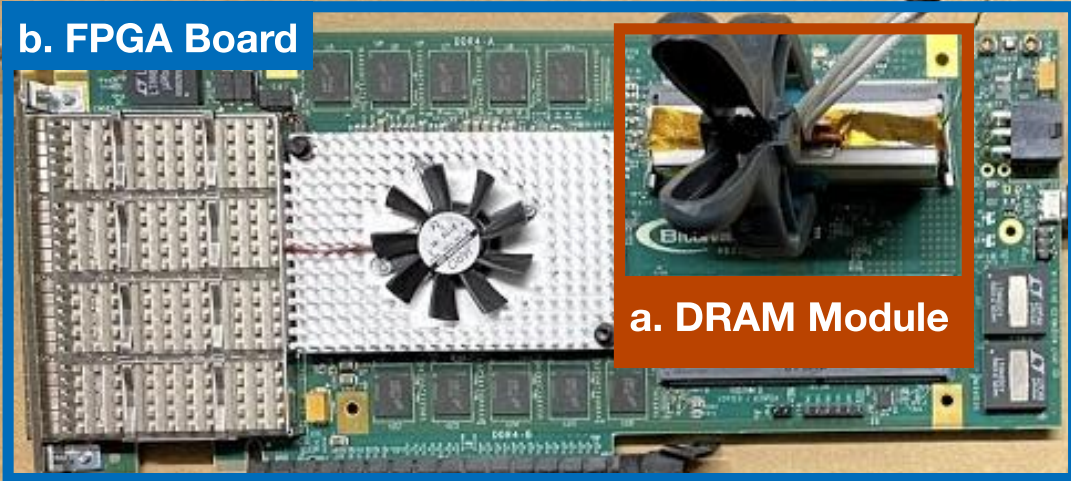


# Real Chip Characterization

Experimentally study QUAC and QUAC-TRNG using 136 real DDR4 chips from SK Hynix

*DDR4 SoftMC* → DRAM Testing Infrastructure

b. FPGA Board



a. DRAM Module

c. PCIe Host Interface



d. Temperature Controller

# Real Chip Characterization

Measure randomness of bitstreams using  
**Shannon Entropy**

$$H(x) = - \sum_{i=1}^2 p(x_i) \log_2 p(x_i) \rightarrow \text{Calculating probabilities:}$$

Proportion of *logic-1* and *logic-0* values in the random bitstream

Sample each bitline following QUAC *1000 times* and calculate the bitline's Shannon Entropy

$$\text{SE}(1111...111) = 0$$

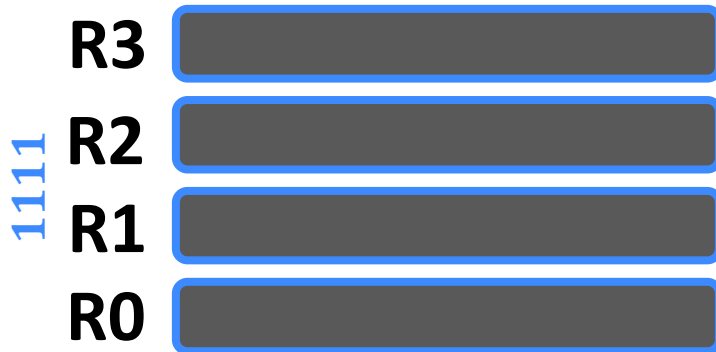
$$0 < \text{SE}(1001...010) < 1$$

# Real Chip Characterization

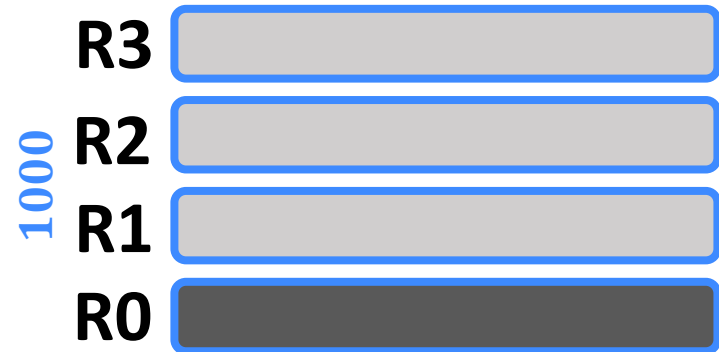
At 50°C and nominal voltage:

Repeatedly perform QUAC 1000 times and measure the Shannon Entropy of each bitline in 8K DRAM Segments (32K DRAM Rows), using all 16 different four-bit data patterns

Data pattern: 1111 (four ones)



Data pattern: 1000



# Data Pattern Dependence

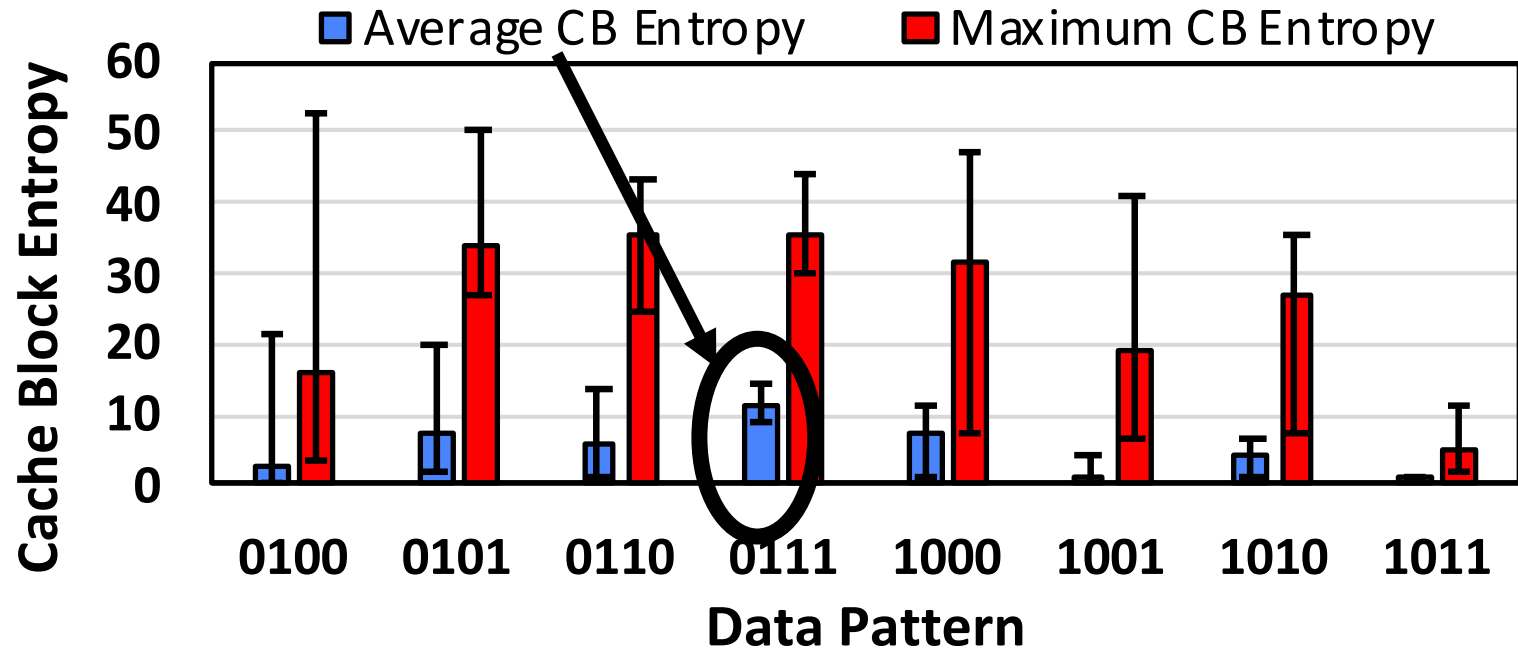
Calculate *cache block entropy (CBE)*

$$\sum \text{all bitline entropies in the cache block}$$

Metrics based on CBE:

- 1. Average CBE:** *Average* entropy across all cache blocks in a module
- 2. Maximum CBE:** *Maximum* of the cache block entropies in a module

# Data Pattern Dependence

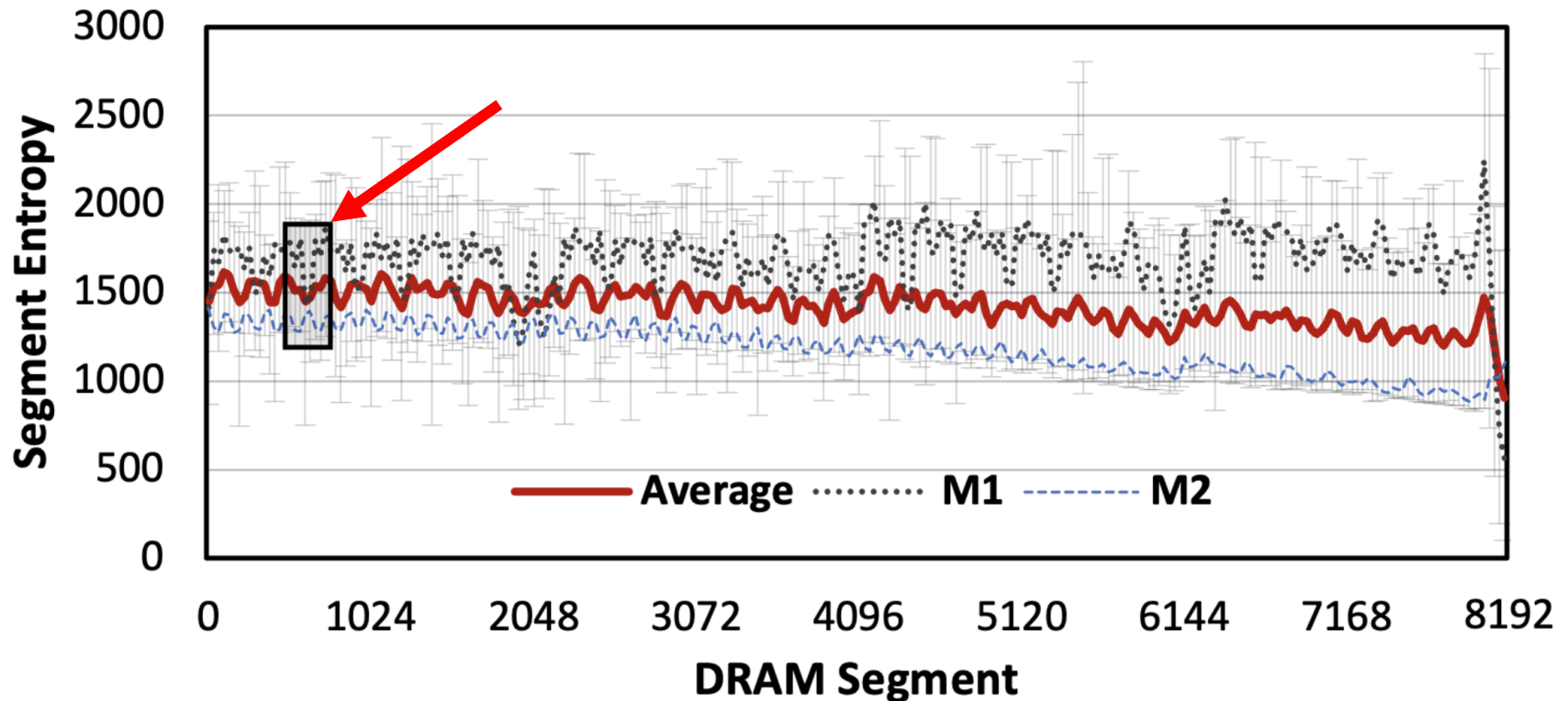


Entropy varies with data pattern

Highest average entropy with pattern "0111"

# Spatial Distribution

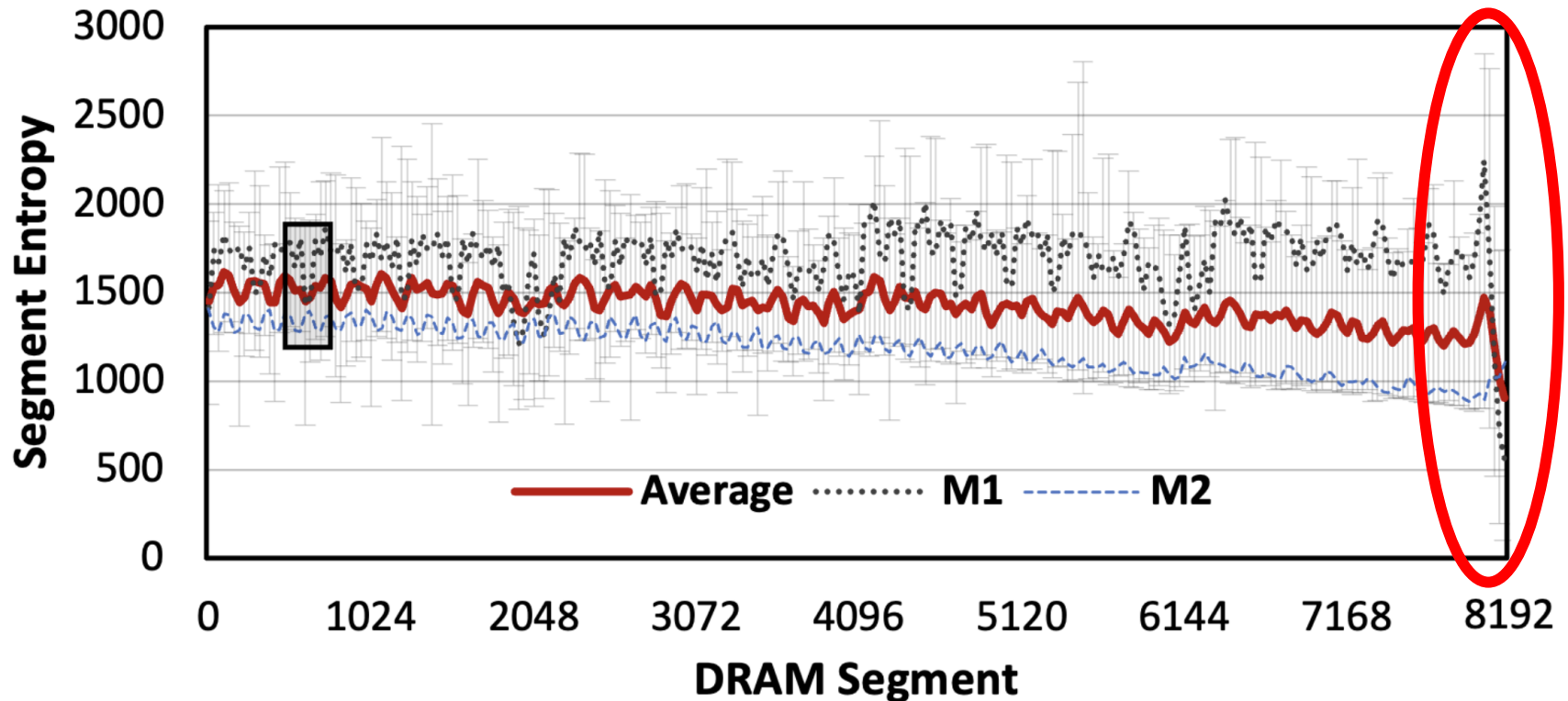
Segment entropy =  $\sum$  all bitline entropies in the segment



Segment entropy behavior is different  
for different modules

# Spatial Distribution

Segment entropy =  $\sum$  all bitline entropies in the segment



Entropy significantly increases towards the end of the DRAM bank

# Takeaways: QUAC Entropy

We observe that entropy resulting from QUAC operations changes according to the

- data pattern used in initialization
- physical location of DRAM segments

attributed to:

- systematic manufacturing process variation
- design-induced variation



# QUAC-TRNG's Quality

Two experiments:

1. Collect bitstreams by repeatedly sampling bitlines after QUAC operations
  - 1 Mb bitstreams
  - Post-processing: Von Neumann Corrector
2. Collect bitstreams using QUAC-TRNG
  - 1 Gb bitstreams
  - Post-processing: SHA-256

# QUAC-TRNG's Quality

Two experiments:

1. Collect bitstreams by repeatedly sampling bitlines after QUAC operations
  - 1 Mb bitstreams
  - Post-processing: Von Neumann Corrector

**QUAC and QUAC-TRNG bitstreams  
pass all 15 NIST randomness tests**

- Post-processing: SHA-256

# QUAC-TRNG Throughput Estimation

Estimate QUAC-TRNG's throughput according to:

$$(256 \times SIB) / (L \times 10^{-9}) \text{ bps}$$

**SIB:** # of **SHA Input Blocks** in the **highest-entropy segment**

**L:** Latency of one QUAC operation in nanoseconds

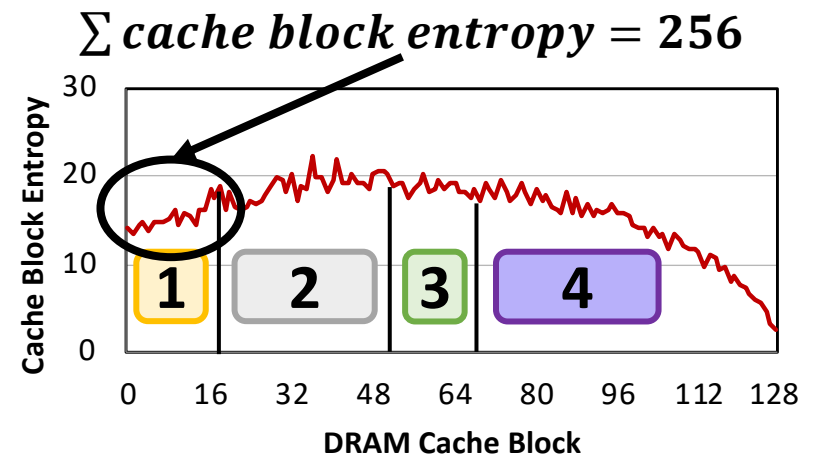
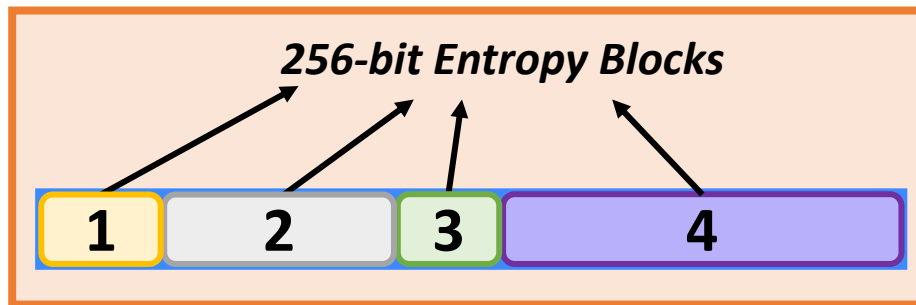
# QUAC-TRNG Throughput Estimation

Estimate QUAC-TRNG's throughput according to:

$$(256 \times SIB) / (L \times 10^{-9}) \text{ bps}$$

**SIB:** # of **SHA Input Blocks** in the **highest-entropy segment**

**L:** Latency of one QUAC operation in nanoseconds



# QUAC-TRNG Throughput Estimation

Estimate QUAC-TRNG's throughput according to:

$$(256 \times SIB) / (L \times 10^{-9}) \text{ bps}$$

**SIB:** # of **SHA Input Blocks** in the **highest-entropy segment**

**L:** Latency of one QUAC operation in nanoseconds

# QUAC-TRNG Throughput Estimation

Estimate QUAC-TRNG's throughput according to:

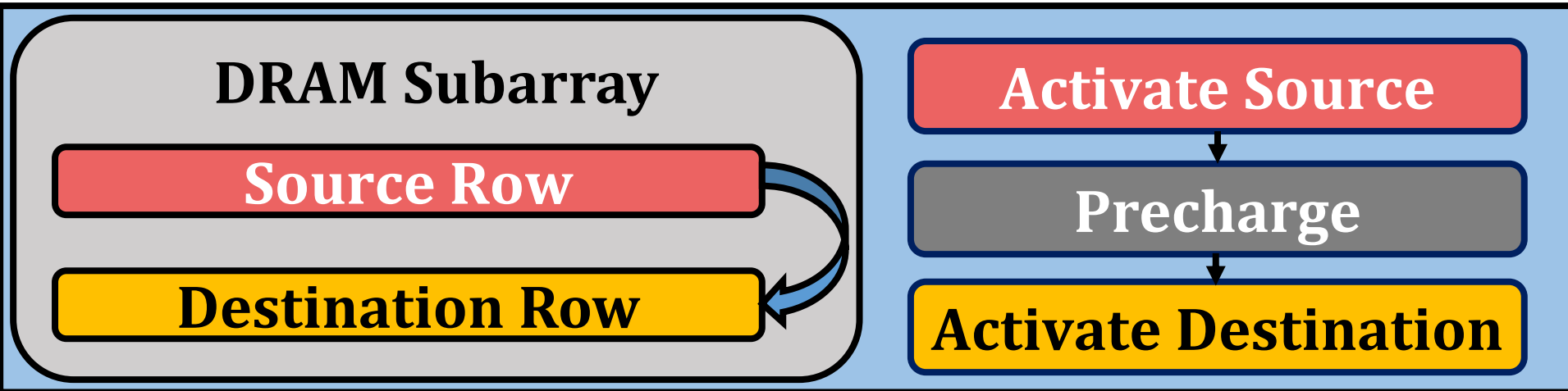
$$(256 \times SIB) / (L \times 10^{-9}) \text{ bps}$$

**SIB:** # of **SHA Input Blocks** in the **highest-entropy segment**

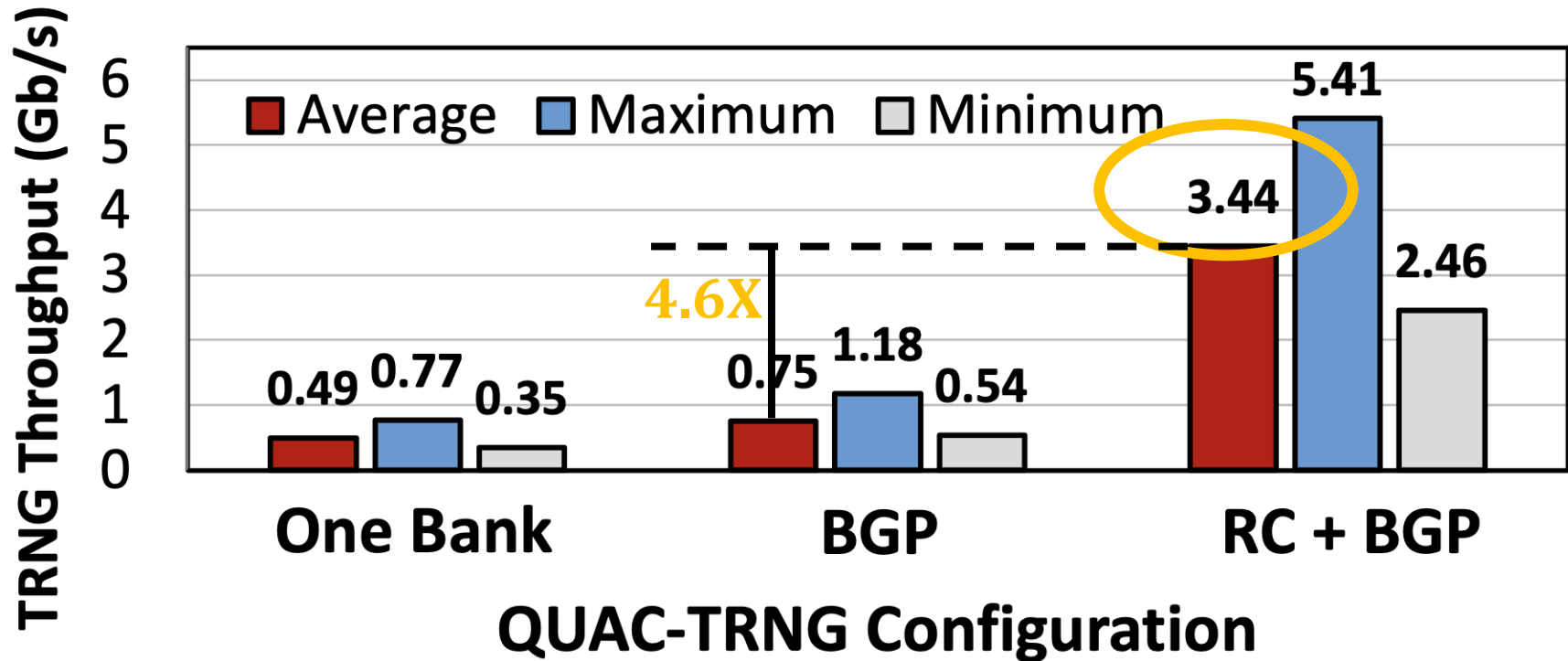
**L:** Latency of one QUAC operation in nanoseconds

# QUAC-TRNG Configurations

- 1 **One Bank** Use a *single DRAM bank*
- 2 **BGP** **Bank Group-Level Parallelism**  
Use *four banks from different bank groups*
- 3 **RC + BGP** **RowClone + BGP**  
Use *in-DRAM copy* to initialize DRAM rows *and* use four banks from *different bank groups*



# QUAC-TRNG Throughput



Achieves **3.44 Gb/s** throughput **per DRAM channel** on average across all modules

In-DRAM initialization greatly improves throughput



# QUAC-TRNG vs State-Of-The-Art

High-throughput DRAM-based TRNGs:

- **D-RaNGe**: Activation latency failures
- **Talukder et. al**: Precharge latency failures

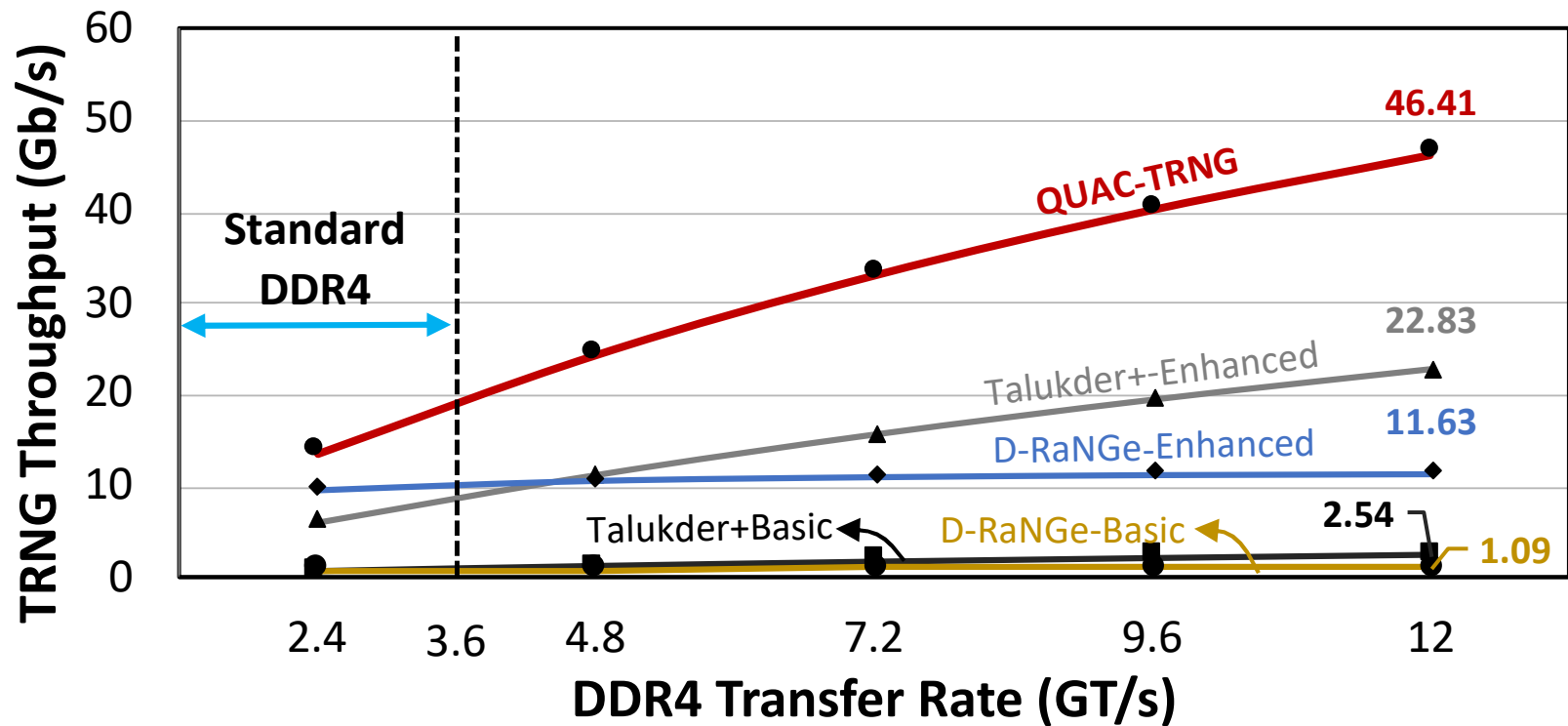
Calculate throughput by **tightly scheduling the DDR4 commands** required to induce failures

Evaluate two versions of these past two works:

- **Base**: As proposed
- **Enhanced (Fair)**: Throughput-optimized (SHA-256)

Assume four-channel DDR4 memory

# QUAC-TRNG vs State-Of-The-Art



Outperforms best prior DRAM-based TRNG

(i) “base” by **15.08x** at **2.4 GT/s**

(ii) “enhanced” by **2.03x** at **12 GT/s**

# More in the Paper

- NIST randomness tests results
- Throughput & latency comparison against four other DRAM-based TRNGs
- System Integration
  - How QUAC-TRNG can be implemented in real systems
  - System performance study
    - QUAC-TRNG's throughput with concurrently running applications
  - Area overhead: 0.04% of a contemporary CPU (7 nm)
  - Memory overhead: 0.002% of an 8 GiB DRAM module
- Sensitivity Analysis
  - Effect of temperature on QUAC's entropy
    - Entropy changes with temperature
  - Time dependence study
    - Entropy remains stable for at least up to a month

# Executive Summary

- **Motivation**: DRAM-based true random number generators (TRNGs) provide **true random numbers at low cost** on a **wide range** of computing systems
- **Problem**: Prior DRAM-based TRNGs are slow:
  1. Based on fundamentally slow processes → **high latency**
  2. Cannot effectively harness entropy from DRAM rows → **low throughput**
- **Goal**: Develop a **high-throughput** and **low-latency** TRNG that uses **commodity DRAM** devices
- **Key Observation**: Carefully engineered sequence of DRAM commands can activate **four DRAM rows** → **QUadruple ACtivation (QUAC)**
- **Key Idea**: Use **QUAC** to activate DRAM rows that are initialized with **conflicting data** (e.g., two '1's and two '0's) to generate random values
- **QUAC-TRNG**: DRAM-based TRNG that generates true random numbers at **high-throughput** and **low-latency** by **repeatedly performing QUAC operations**
- **Results**: We evaluate QUAC-TRNG using **136** real DDR4 chips
  1. **5.4 Gb/s** maximum (**3.4 Gb/s** average) TRNG throughput per DRAM channel
  2. QUAC-TRNG has low TRNG latency: **256-bit RN** in **274 ns**
  3. Outperforms existing DRAM-based TRNGs by **15.08x** (base), and **1.41x** (enhanced)
  4. QUAC-TRNG passes **all 15** NIST randomness tests

# QUAC-TRNG

*High-Throughput True Random Number Generation  
Using Quadruple Row Activation in Real DRAM Chips*

**Ataberk Olgun**

Minesh Patel   A. Giray Yağlıkçı   Haocong Luo

Jeremie S. Kim   F. Nisa Bostancı   Nandita Vijaykumar

Oğuz Ergin   Onur Mutlu

**SAFARI**

 **kasirga**

**ETH** zürich



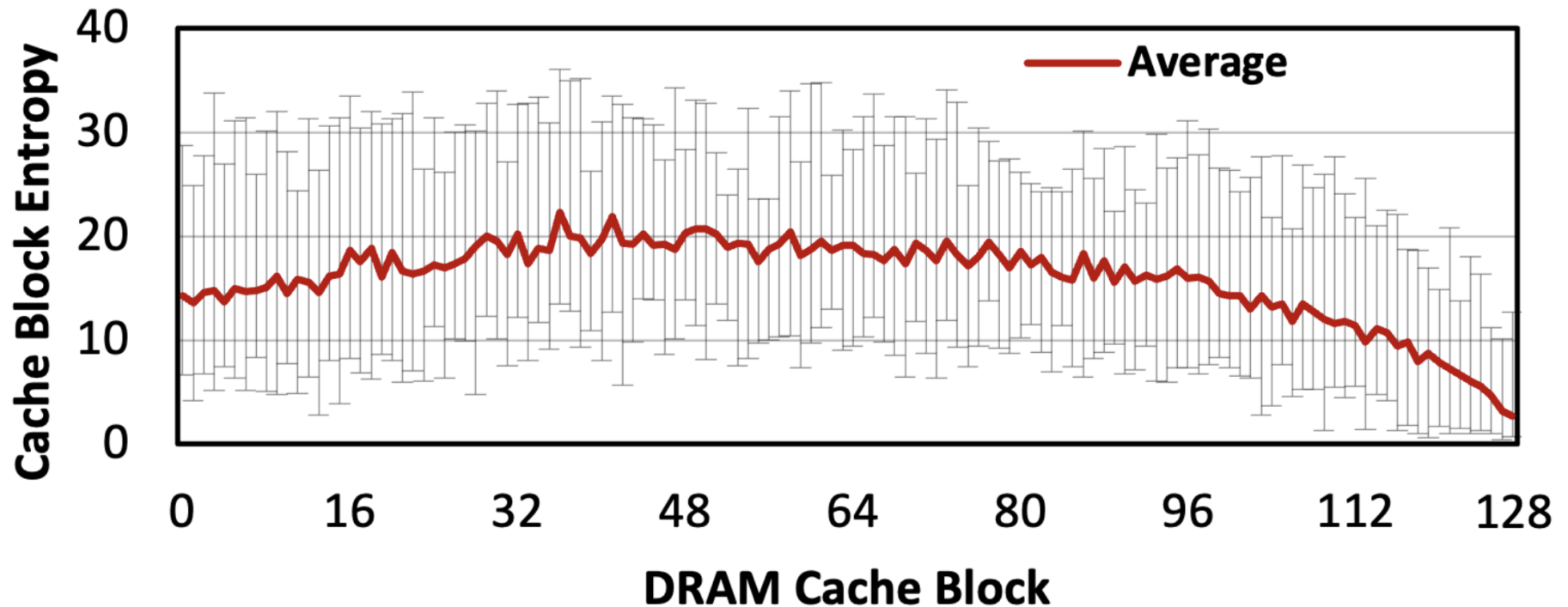
**TOBB ETÜ**  
University of Economics & Technology



UNIVERSITY OF  
**TORONTO**

***BACKUP SLIDES***

# Spatial Distribution



Cache block entropy is the highest around the middle of the DRAM segment

# NIST Results

**Table 1: NIST STS Randomness Test Results**

<b>NIST STS Test</b>	<b>VNC* (p-value)</b>	<b>SHA-256 (p-value)</b>
monobit	0.430	0.500
frequency_within_block	0.408	0.528
runs	0.335	0.558
longest_run_ones_in_a_block	0.564	0.533
binary_matrix_rank	0.554	0.548
dft	0.538	0.364
non_overlapping_template_matching	>0.999	0.488
overlapping_template_matching	0.513	0.410
maururs_universal	0.493	0.387
linear_complexity	0.483	0.559
serial	0.355	0.510
approximate_entropy	0.448	0.539
cumulative_sums	0.356	0.381
random_excursion	0.164	0.466
random_excursion_variant	0.116	0.510

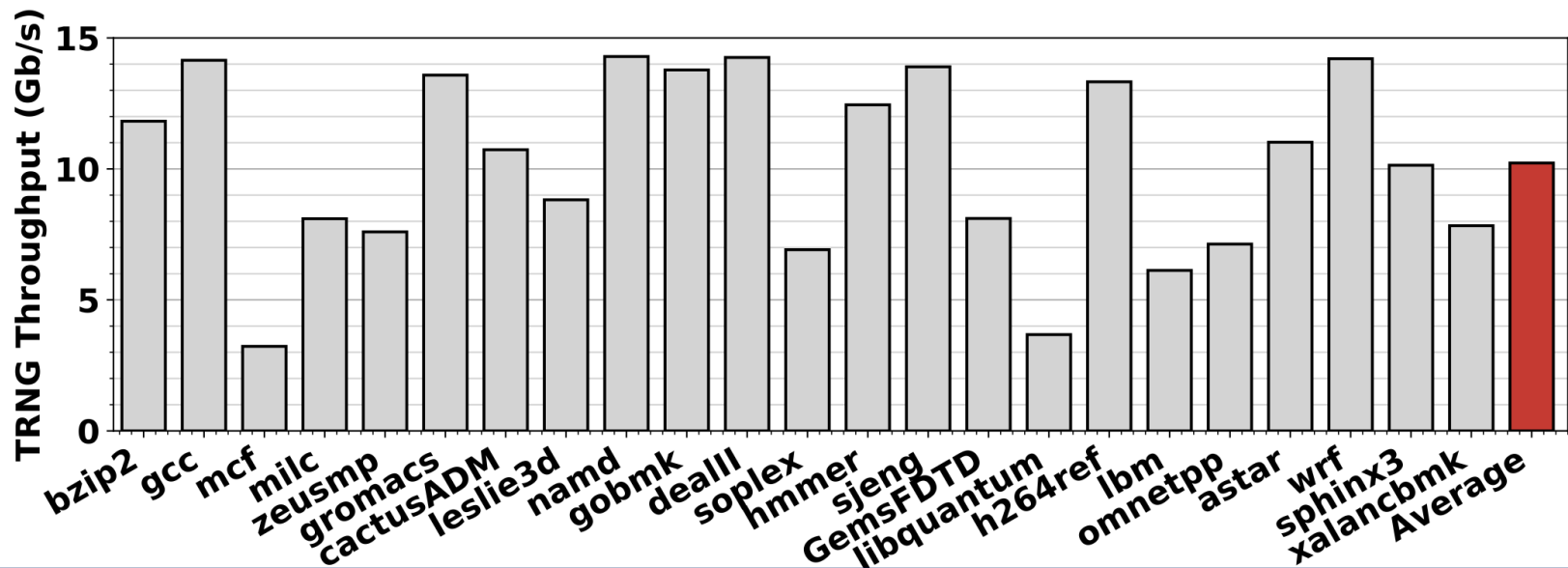
\*VNC: Von Neumann Corrector



# System Performance Study

The maximum throughput QUAC-TRNG provides without reducing the total off-chip memory bandwidth

**Ramulator:** 3.2 GHz core, four-channel DDR4 memory



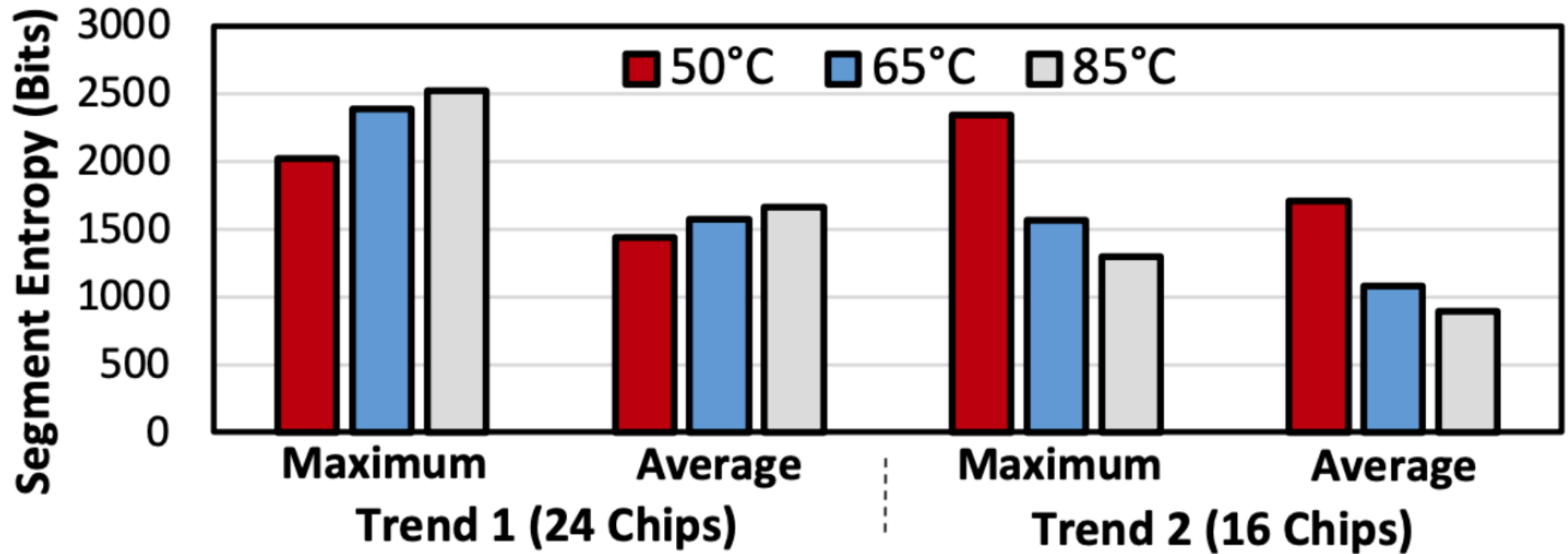
QUAC-TRNG achieves **74.13%** of the empirical average throughput

# Throughput & Latency Comparison

**Table 2: Summary of prior DRAM-TRNGs vs QUAC-TRNG**

Proposal	Entropy Source	TRNG Throughput	256-bit TRNG Latency
<b>QUAC-TRNG</b>	Quadruple ACT	13.76 Gb/s	274 ns
Talukder+ [15]	Precharge Failure	0.68 - 6.13 Gb/s	249 ns - 201 ns
D-RaNGe [88]	Activation Failure	0.92 - 9.73 Gb/s	260 ns - 36 ns
D-PUF [150]	Retention Failure	0.20 Mb/s	40 s
DRNG [47]	DRAM Start-up	N/A	700 $\mu$ s
Keller+ [81]	Retention Failure	0.025 Mb/s	40 s
Pyo+ [126]	DRAM Cmd Schedule	2.17 Mb/s	112.5 $\mu$ s

# Temperature Dependence



**Figure 14: Maximum and average segment entropy at different temperatures.**

# DDR4 Modules

Module	Module Identifier	Chip Identifier	Freq. (MT/s)	Organization			Segment Entropy		
				Size (GB)	Chips	Pins	Avg.	Max. <sup>†</sup>	Avg. (after 30 days)
M1	Unknown	H5AN4G8NAFR-TFC	2133	4	8	x8	1688.1	2247.4	–
M2	Unknown	Unknown	2133	4	8	x8	1180.4	1406.1	–
M3	Unknown	H5AN4G8NAFR-TFC	2133	4	8	x8	1205.0	1858.3	1192.9
M4	76TT21NUS1R8-4G	H5AN4G8NAFR-TFC	2133	4	8	x8	1608.1	2406.5	1588.0
M5	Unknown	T4D5128HT-21	2133	4	8	x8	1618.2	2121.6	–
M6	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1211.5	1444.6	–
M7	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1177.7	1404.4	–
M8	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1332.9	1600.9	1407.0
M9	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1137.1	1370.9	–
M10	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1208.5	1473.2	1251.8
M11	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1176.0	1382.9	1165.1
M12	TLRD44G2666HC18F-SBK	H5AN4G8NMFR-VKC	2666	4	8	x8	1485.0	1740.6	–
M13	KSM32RD8/16HDR	H5AN4G8NAFA-UHC	2400	4	8	x8	1853.5	2849.6	–
M14	F4-2400C17S-8GNT	H5AN4G8NMFR-UHC	2400	8	8	x8	1369.3	1942.2	–
M15	F4-2400C17S-8GNT	H5AN4G8NMFR-UHC	3200	8	8	x8	1545.8	2147.2	–
M16	KSM32RD8/16HDR	H5AN8G8NDJR-XNC	3200	16	8	x8	1634.4	1944.6	–
M17	KSM32RD8/16HDR	H5AN8G8NDJR-XNC	3200	16	8	x8	1664.7	2016.6	–

<sup>†</sup>The maximum possible entropy in a DRAM segment is 64K (65,536) bits.