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

Authors: Jeremie S. Kim‡§ Minesh Patel§ Hasan Hassan§ Lois Orosa§ Onur Mutlu§‡

‡Carnegie Mellon University §ETH Zürich

First presented: 25th IEEE International Symposium on High-Performance Computer Architecture 2019

Presented by: Axel Schwarzenbach

Executive Summary

Motivation: High throughput and low latency True Random Number Generators (TRNGs) are a key component for encryption and randomized algorithms. Many commodity devices do not posses dedicated True Random Number Generator hardware but have DRAM.

Current Problem: Prior approach to TRNG designs based in DRAM either 1) exploit a fundamentally non-deterministic entropy source or 2) are too slow for continuous high-throughput operations.

Goal: A novel approach to TRNGs that uses existing DRAM devices with 1) low implementation cost, 2) low latency and 3) high throughput

Key Idea: Exploit non-determinism in DRAM cells' activation failures to generate true random numbers.

Evaluation: D-RaNGe was implemented and tested on 282 real LPDDR4 DRAM devices showing a remarkably high peak throughput (717.4 Mb/s) and very low latency (100ns).

Overview

Overview

Motivation

- Low latency, high throughput true random numbers (TRNs) are required for many applications
  - Encryption algorithms and standard protocols (i.e. TLS,SSL,RSA,VPN keys) require TRN
  - Other purposes include randomized algorithms, simulation and complex modelling
- A TRNG requires a physical process (e.g. radioactive decay, thermal noise, clock jitters)
- Most devices lack the dedicated hardware for a high throughput TRNG
- DRAM is widely available in most modern devices
- A widely available TRNG would allow applications requiring True Random Numbers to run on most devices

Overview

Goal

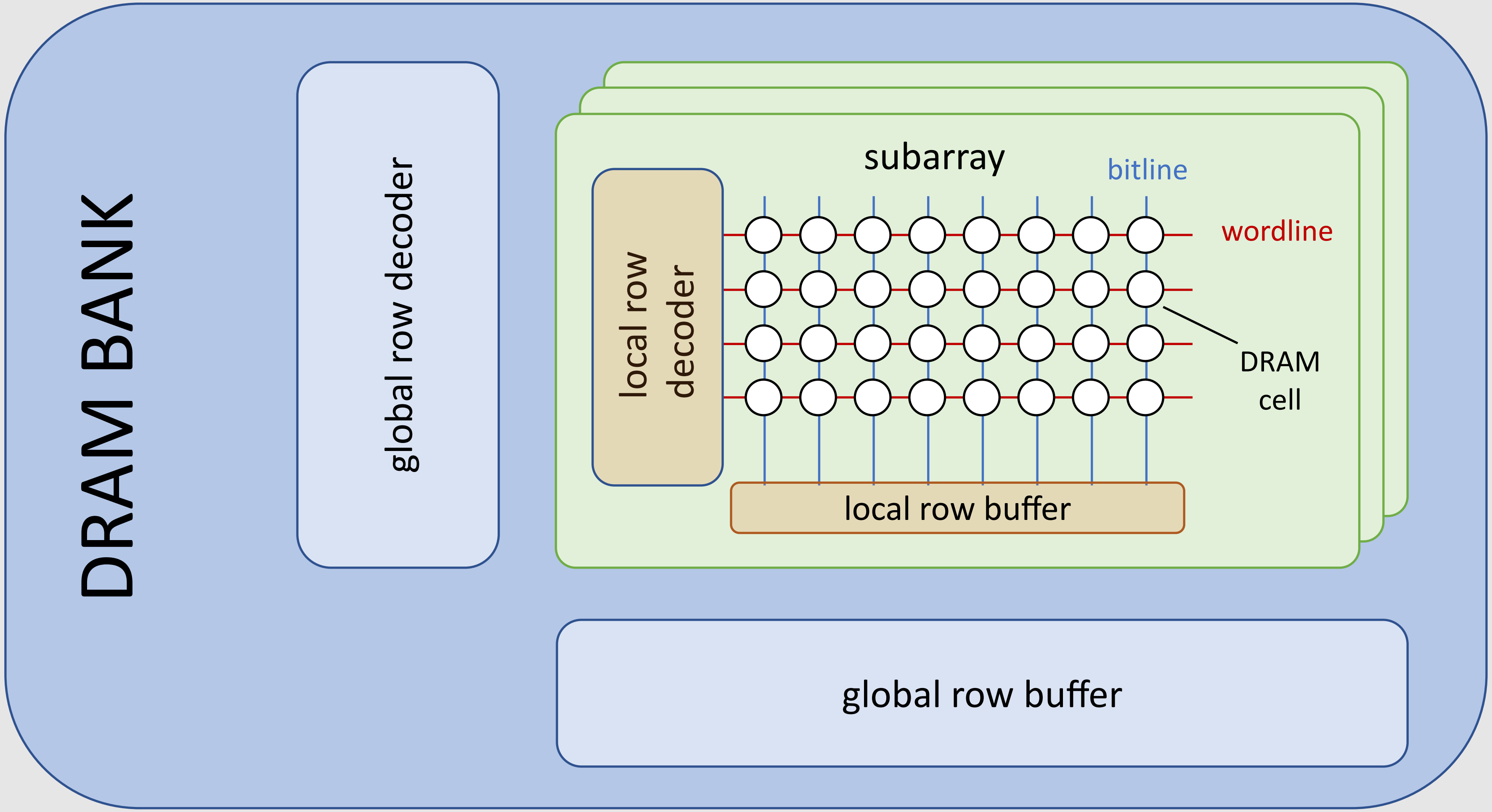
The goal is to devise a TRNG in DRAM device that satisfies the six key properties of an effective TRNG:

1. Low implementation cost
2. Fully non-deterministic
3. Provide a continuous stream of random numbers with high throughput
4. Provide random numbers with low latency
5. Exhibit low system interference
6. Generate random values with low energy overhead

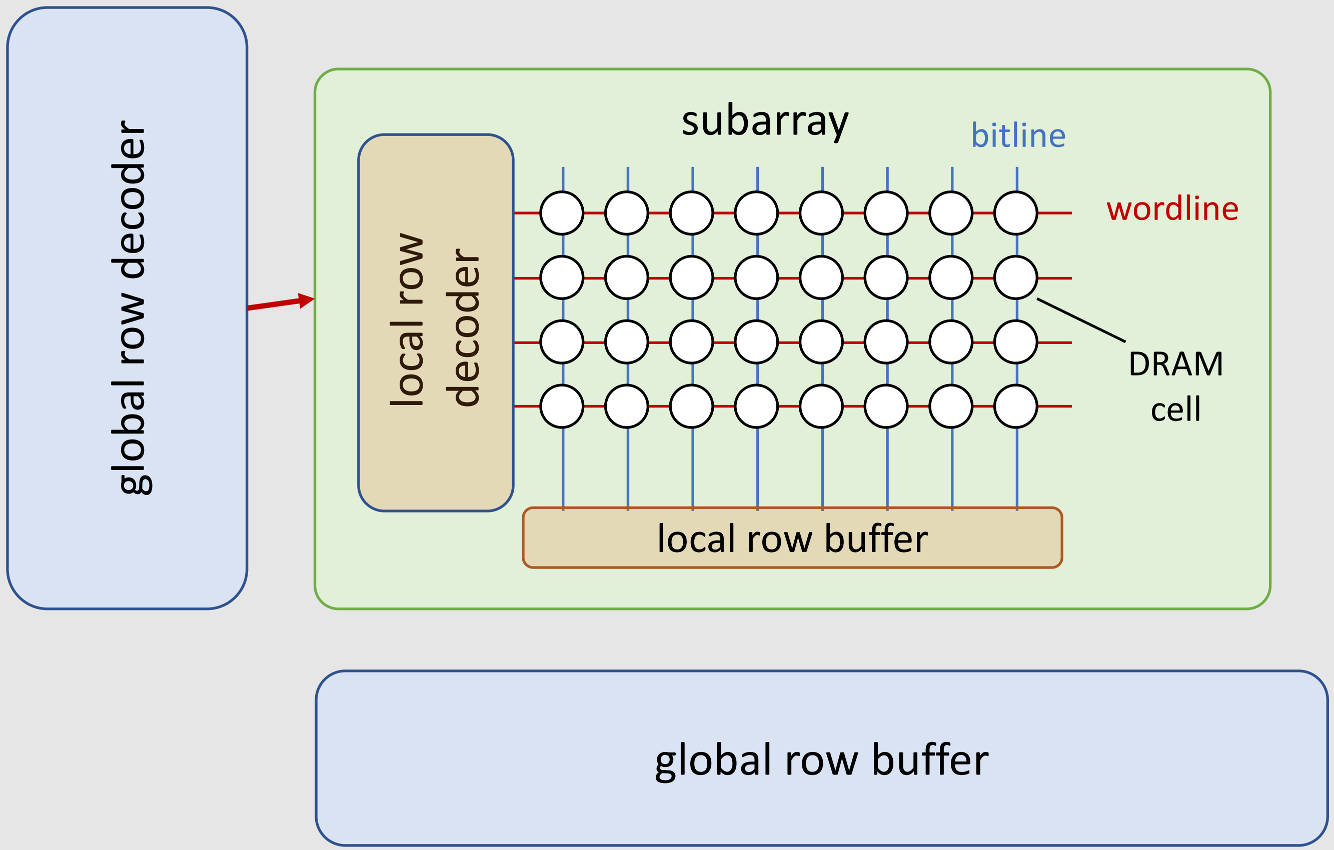
Overview

- Motivation
- Goal
- **DRAM Background**
- D-RaNGe
- Testing Environment
- Results
- Comparison to Prior Work
- Summary and Conclusion
- Strengths
- Limitations
- Discussion

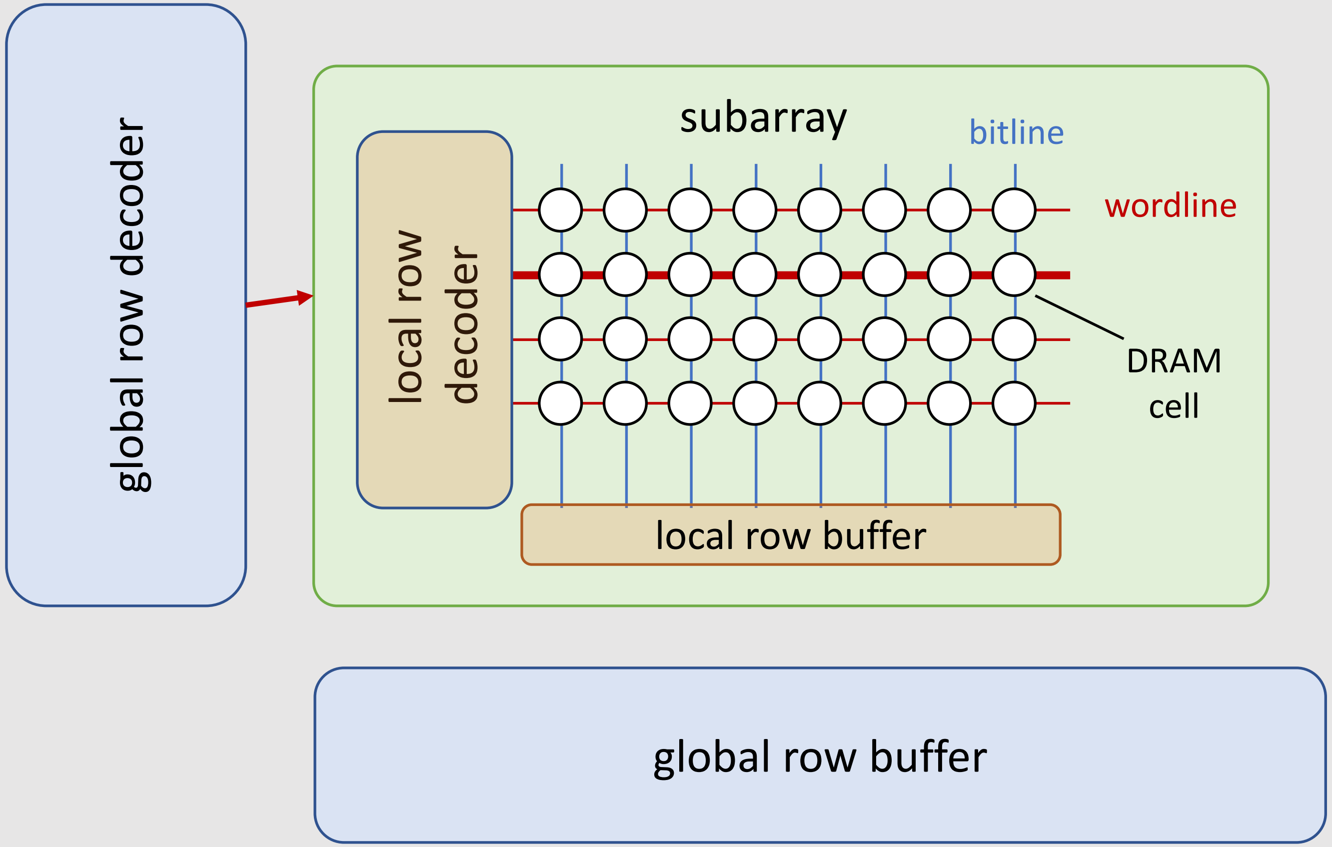
DRAM Background



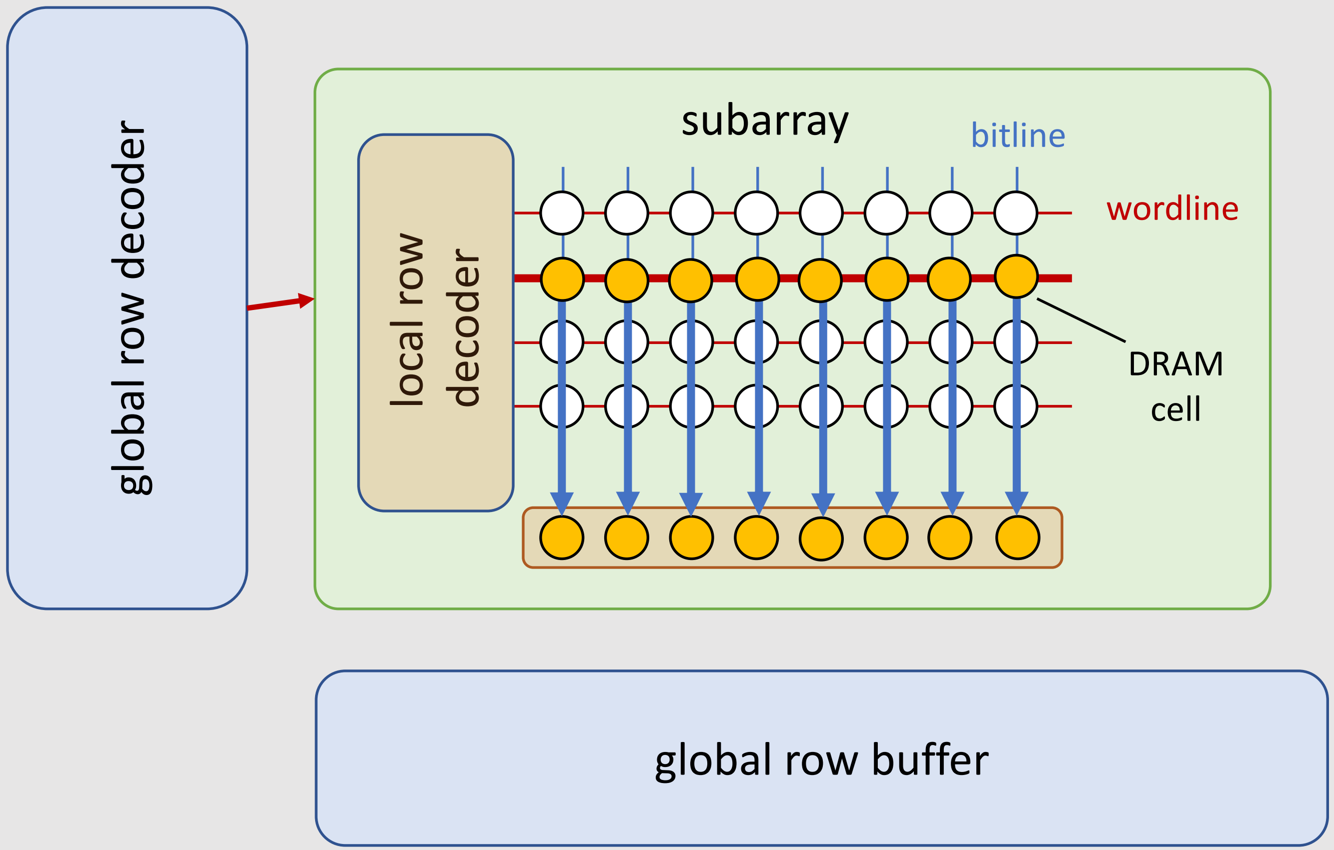
DRAM Background



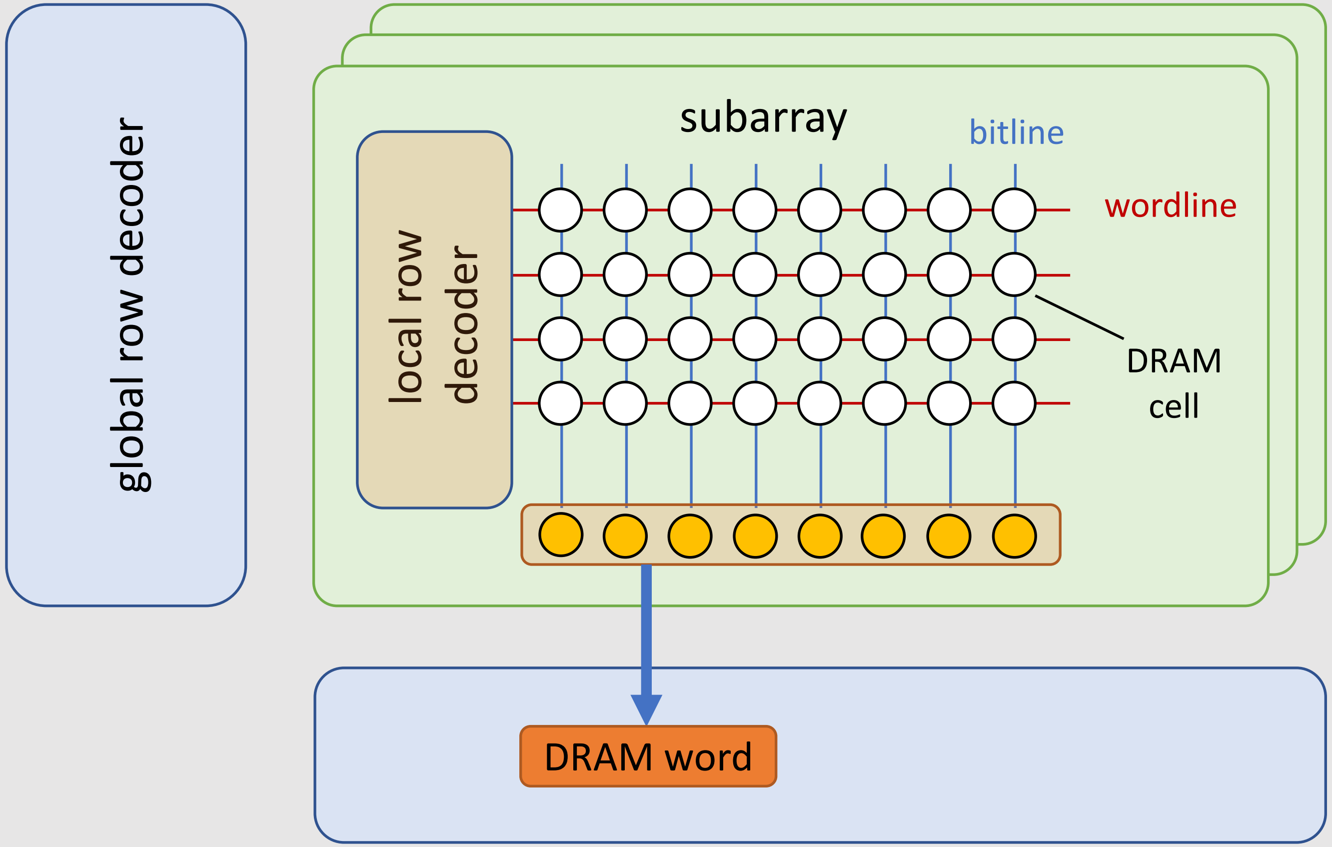
DRAM Background



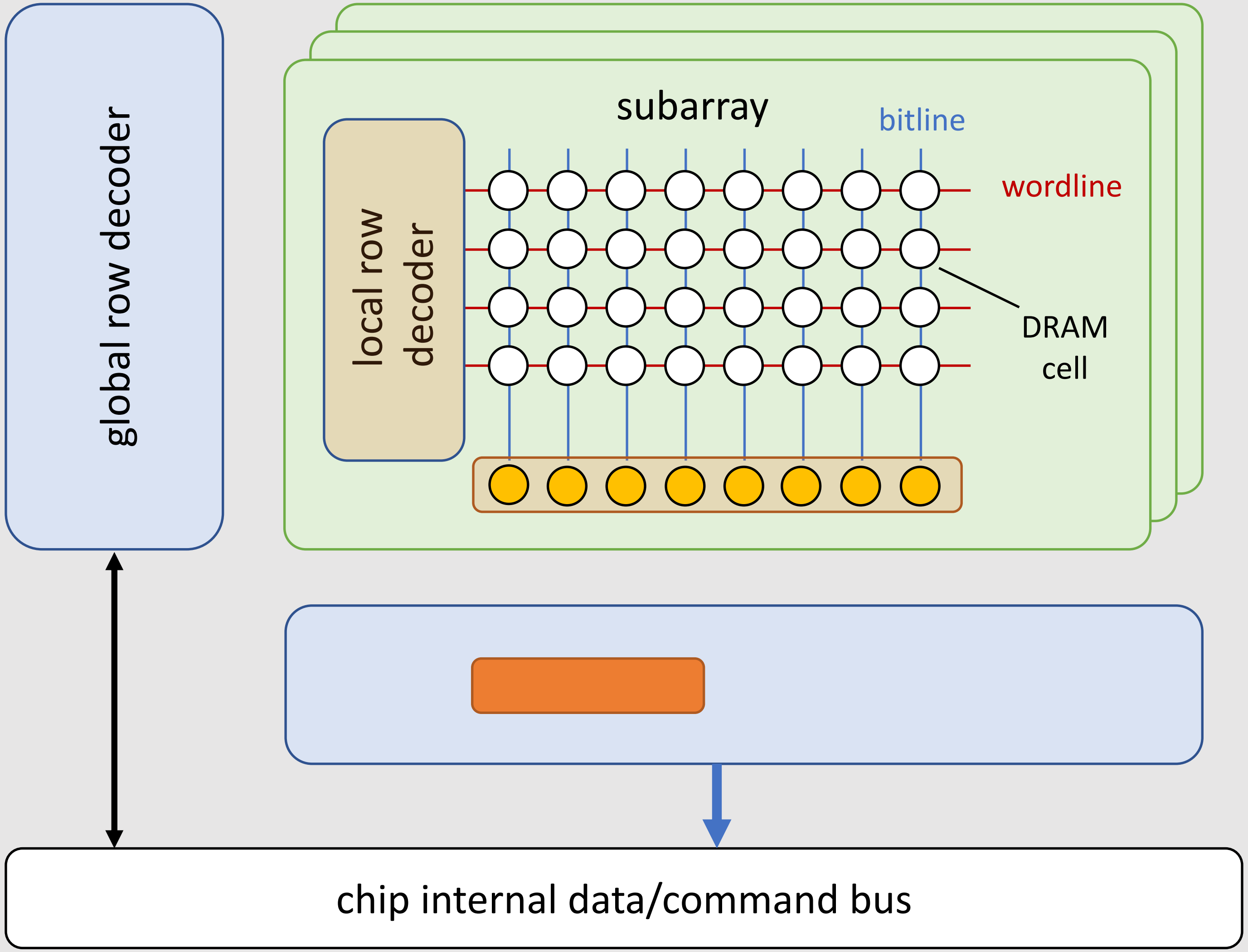
DRAM Background



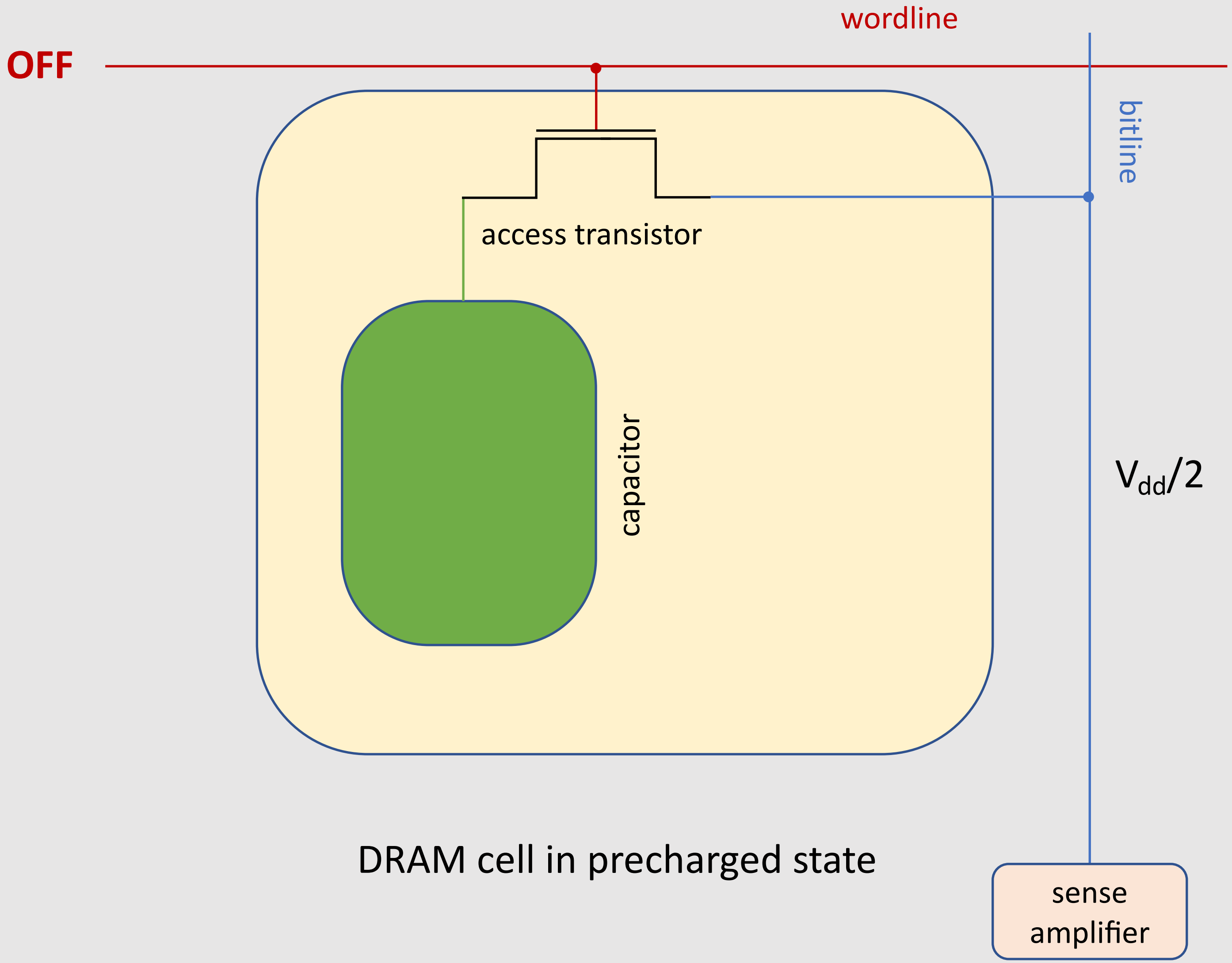
DRAM Background



DRAM Background

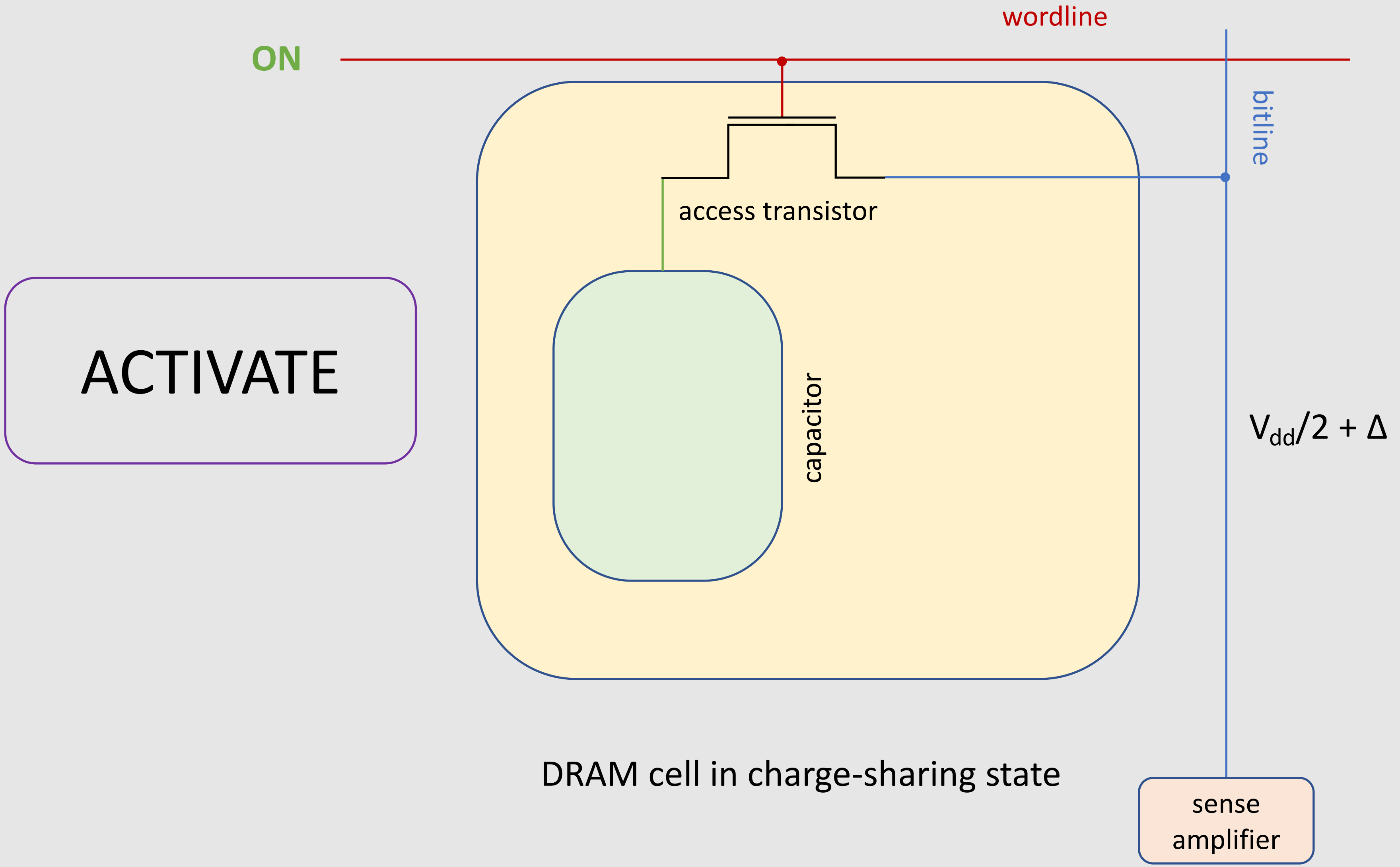


DRAM Background



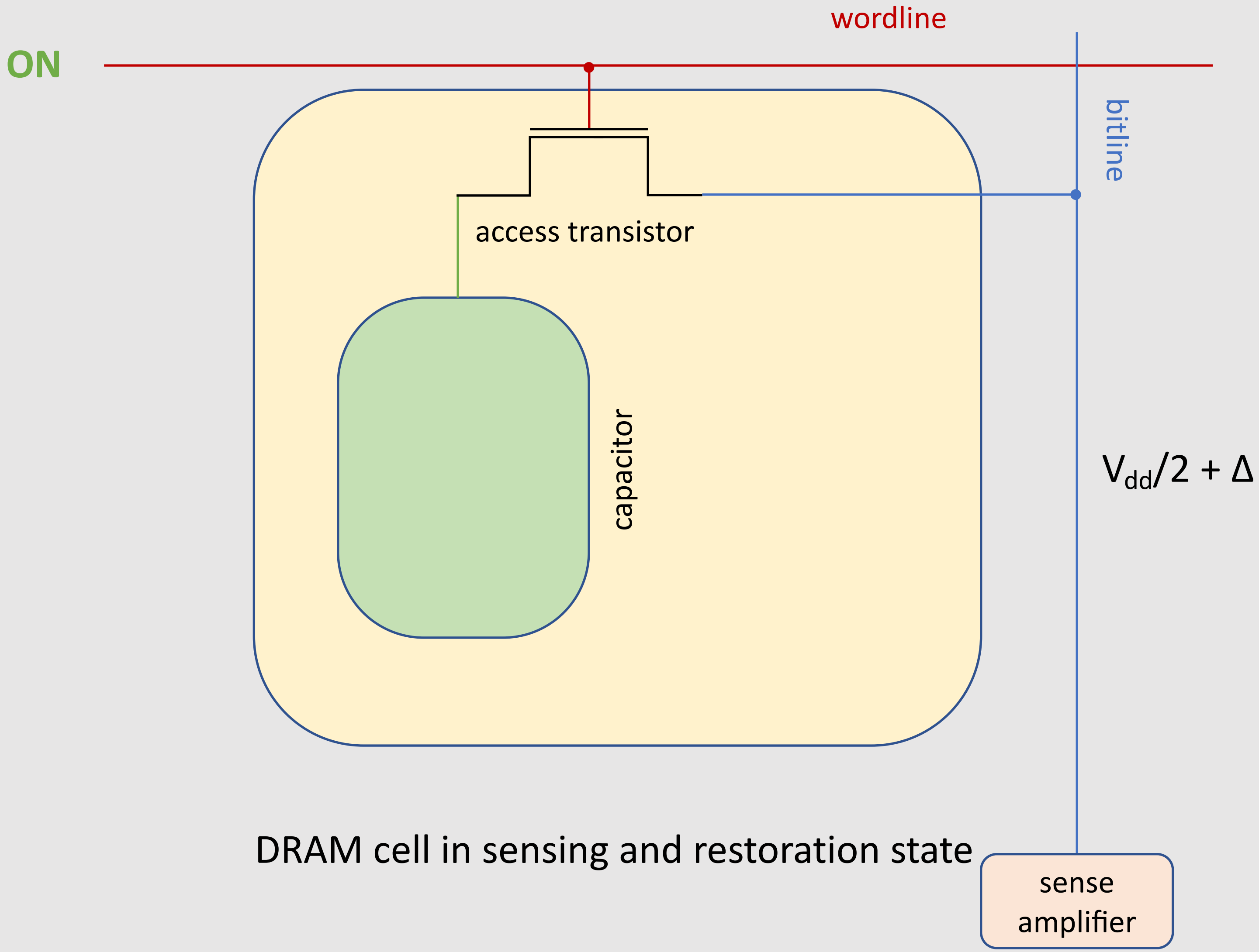
DRAM cell in precharged state

DRAM Background



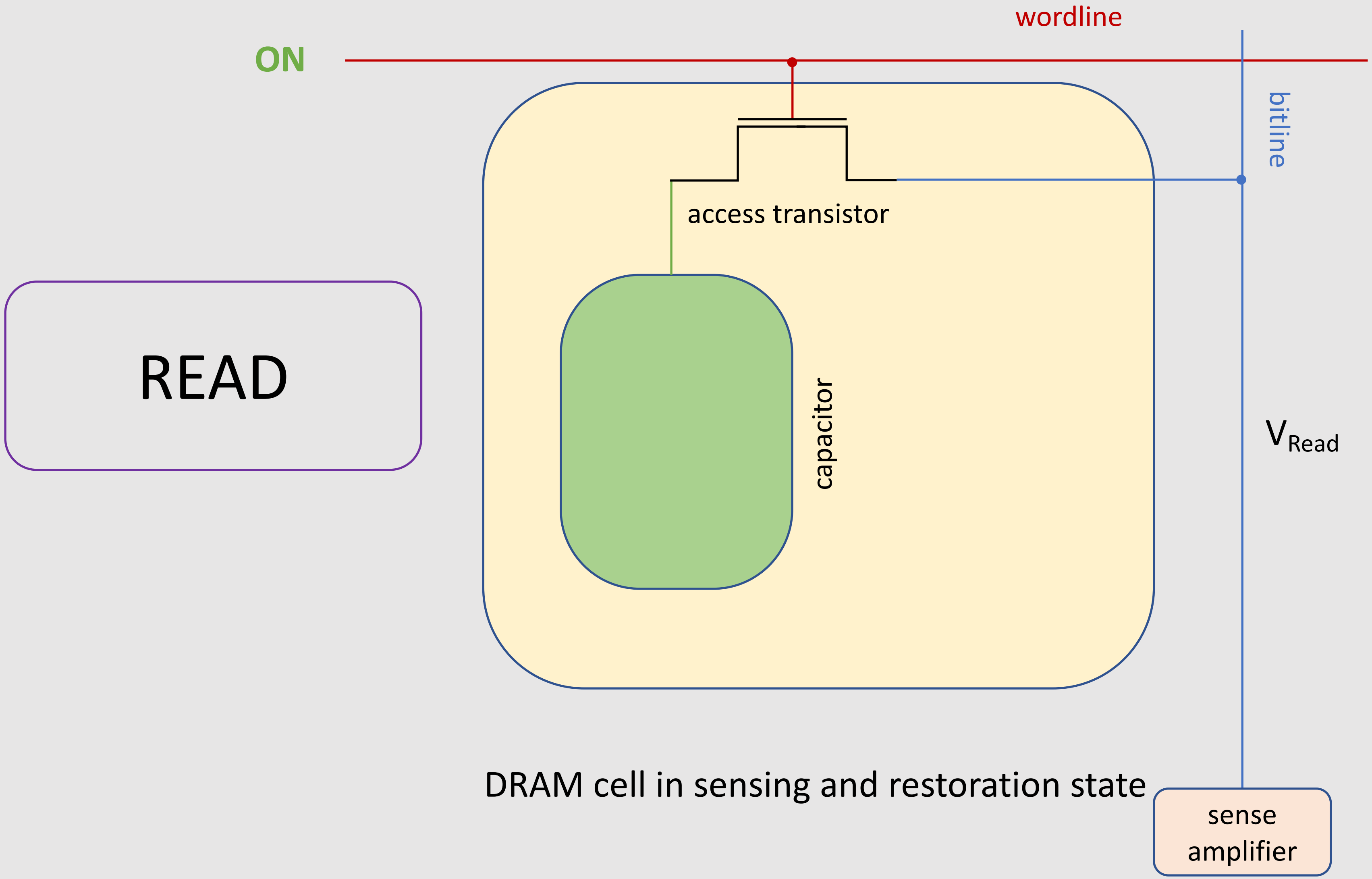
DRAM cell in charge-sharing state

DRAM Background



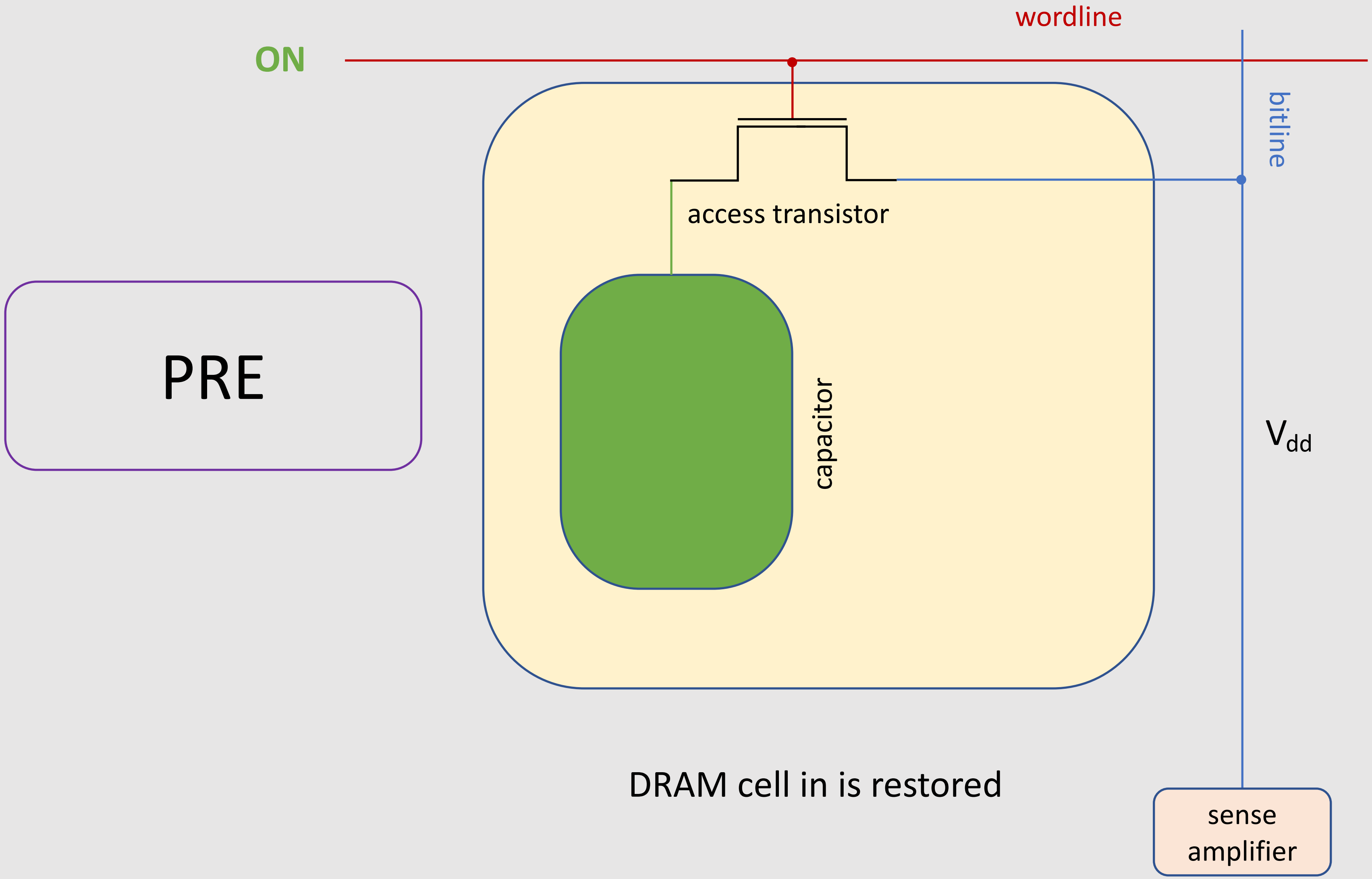
DRAM cell in sensing and restoration state

DRAM Background



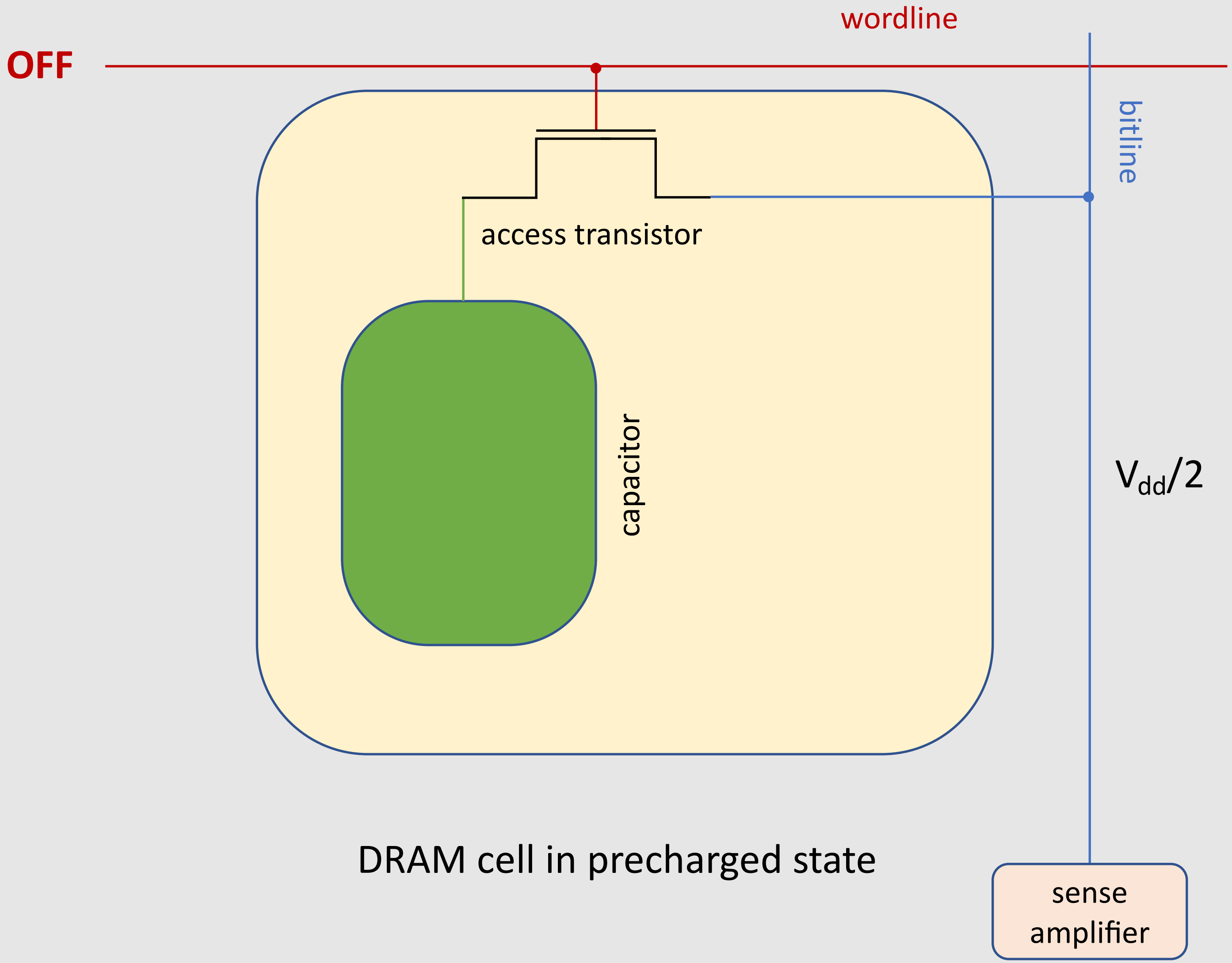
DRAM cell in sensing and restoration state

DRAM Background



DRAM cell in is restored

DRAM Background



DRAM cell in precharged state

Overview

D-RaNGe

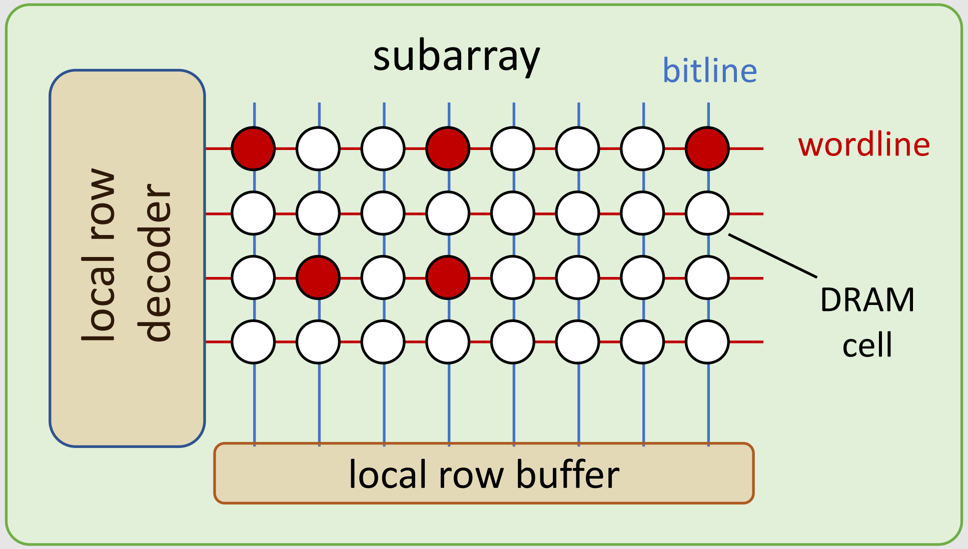
- Observation: Reducing the time interval between the ACTIVATE and the READ (t_{RCD}) command leads to random errors
- Idea: Sampling DRAM cells that fail with a probability of 50% and high entropy to generate truly random data (RNG cells)

D-RaNGe: Finding RNG Cells

- Goal: Finding DRAM cells that have a failure probability of 50% and high entropy
- Each cell in a DRAM bank is read 1M times with reduced t_{RCD} parameter
- The NIST statistical suite for randomness is run on the resulting bitstreams
- The cells that pass the NIST tests are chosen as RNG cells
- RNG cell location in memory, operating temperature and t_{RCD} value are stored in the memory controller

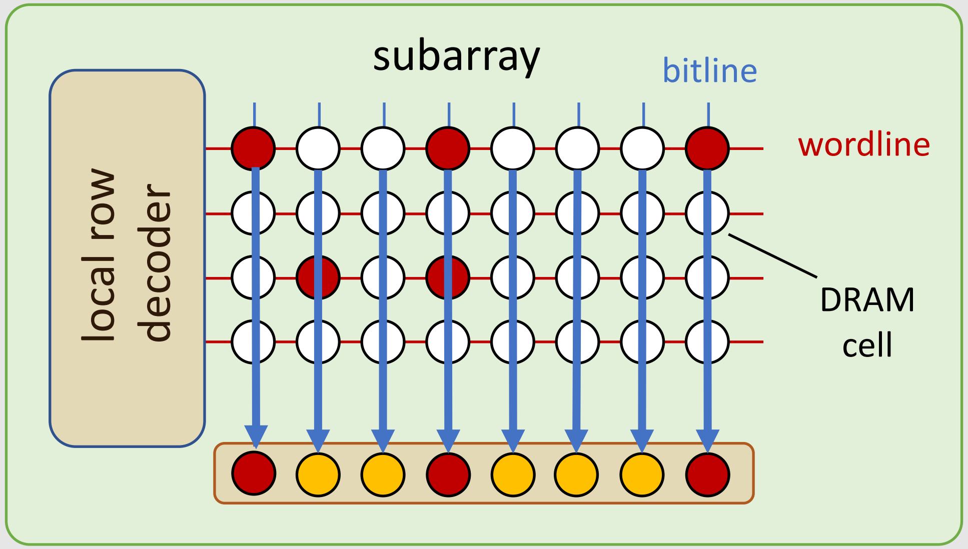
D-RaNGe: Sampling RNG Cells

- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures



D-RaNGe: Sampling RNG Cells

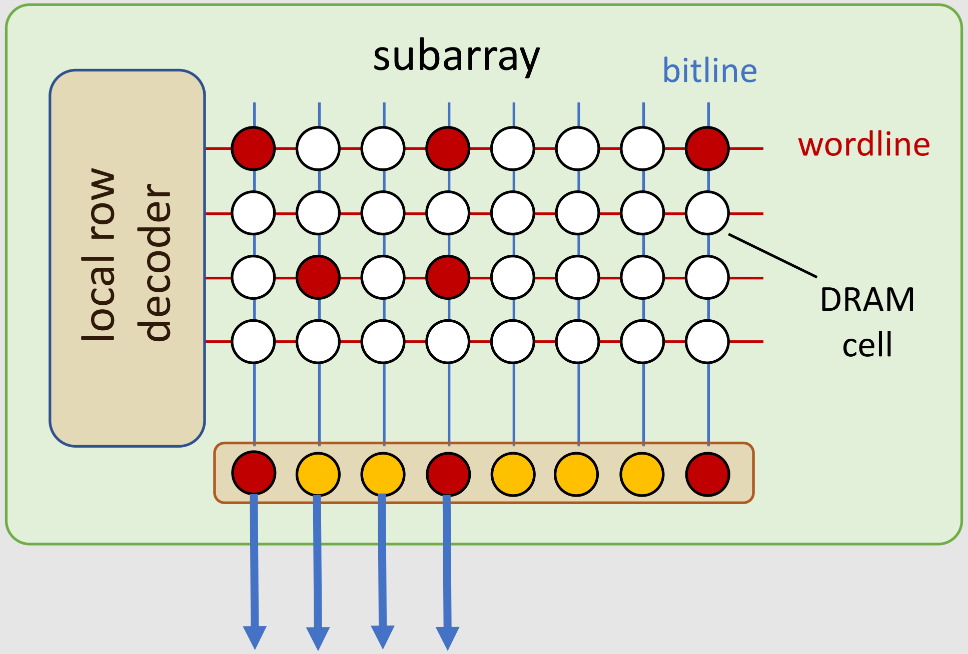
- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures

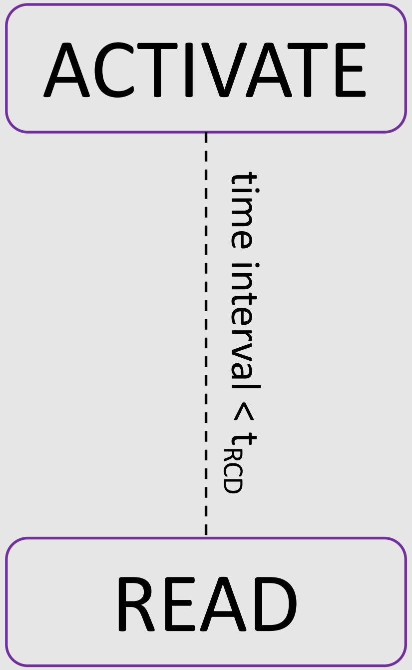


ACTIVATE

D-RaNGe: Sampling RNG Cells

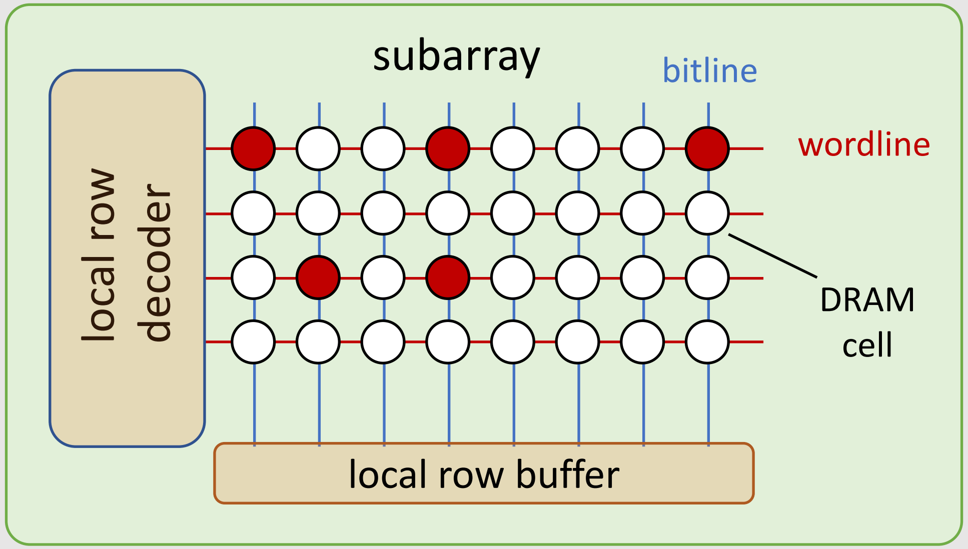
- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures





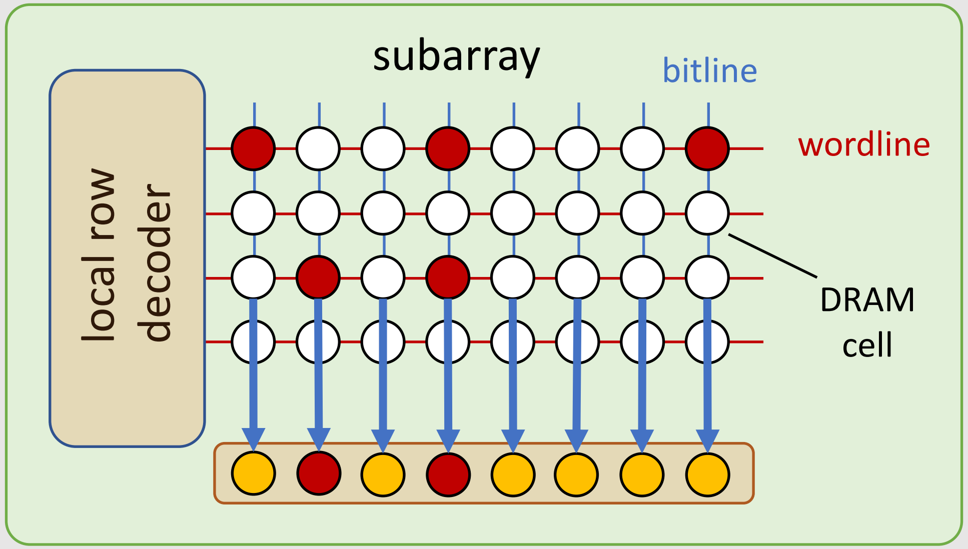
D-RaNGe: Sampling RNG Cells

- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures



D-RaNGe: Sampling RNG Cells

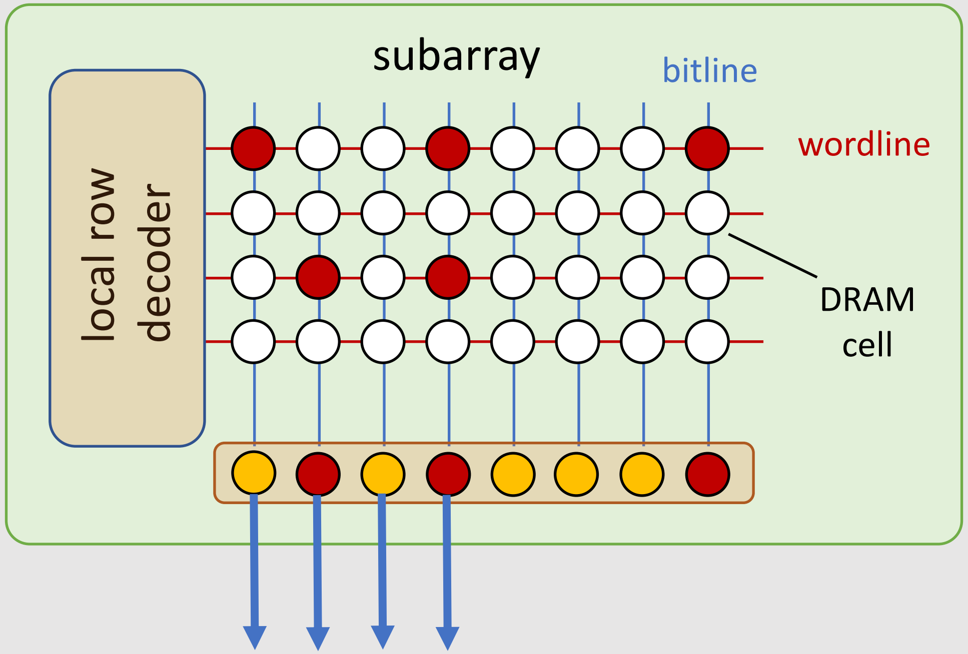
- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures

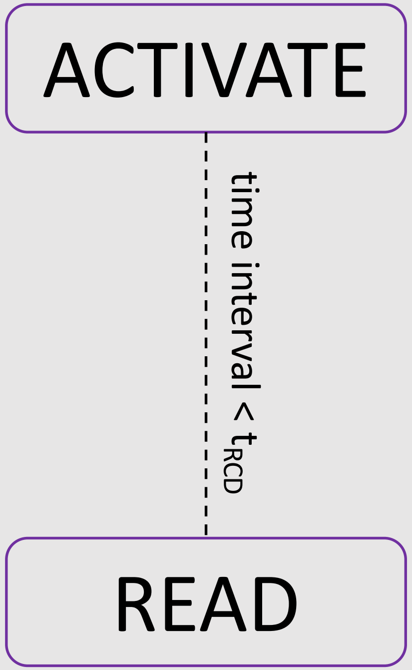


ACTIVATE

D-RaNGe: Sampling RNG Cells

- Reading an RNG cell with reduced t_{RCD} results in random output
- Inducing bank conflicts maximizes the number of activation failures





D-RaNGe: System Integration

- D-RaNGe obtains exclusive access for target rows and cells adjacent to RNG cells
- Can be implemented without any hardware modifications in many existing architectures
- Implemented with firmware running exclusively in the memory controller
- Performance overhead can be reduced by maintaining a queue of already-harvested random data
- Could be integrated in existing architectures by adding a new ISA instruction (i.e. RDRAND from Intel)

Overview

Testing Environment

- All tests were performed:
  - on a total pf 282 2y-nm LPDDR4 DRAM chips from three major manufacturers
  - in a thermally-controlled chamber with a reliable temperature range of 40°C to 55°C and an accuracy of 0.25°C
- DRAM temperature was maintained at 15°C above ambient temperature using a separate heating source
- A separate infrastructure allowed precise control and testing with different timing parameters
- DRAMPower and Ramulator were used to compute energy consumption

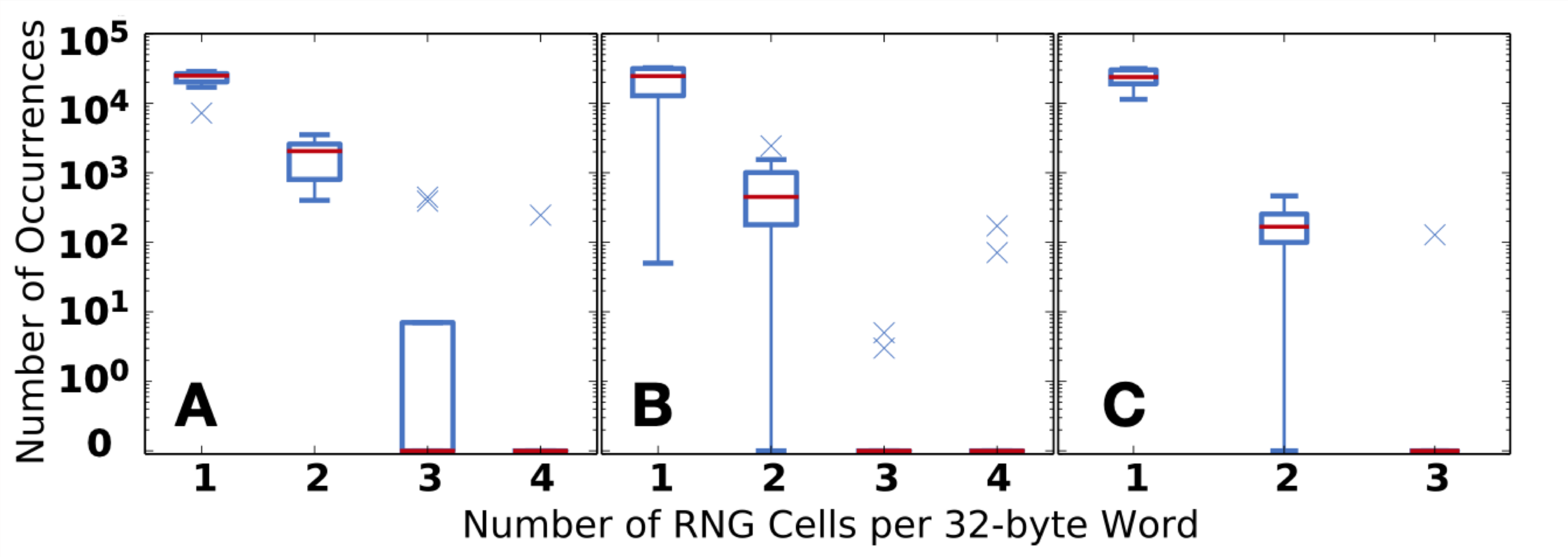
Overview

Results: NIST Tests

- 4 RNG cells from each of 59 DRAM chips were sampled 1M times
- The entropy of each 1Mb bitstream was evaluated with the NIST test suite for randomness
- NIST test suite for randomness includes:
  - A frequency test across the whole bitstream
  - A frequency test for blocks of the bitstream
  - Runs test
- All sampled RNG cells passed all tests

Results: RNG Cell Distribution

- The throughput of D-RaNGe depends on
  1. The density of RNG cells per DRAM word
  2. The bandwidth at which DRAM words can be accessed while inducing activation failures



Source: https://people.inf.ethz.ch/omutlu/pub/drange-dram-latency-based-true-random-number-generator_hpca19.pdf

Results: Key Properties of a TRNG

- Low implementation cost:
  - To induce activation failure, timing parameters must be modifiable below manufacturer-specified values
  - Some processors already allow software to change memory controller registers
  - Most processor only need minimal software changes to expose an interface for changing memory controller registers
  - A few minimal hardware changes would have to be implemented for all other chips
- Fully non-deterministic:
  - The NIST test suite suggests that the RNG cells are a fully non-deterministic entropy source

Results: Key Properties of a TRNG

- High throughput of random data:
  - Throughput is linearly correlated with the number of banks utilized
  - A minimum throughput of 40 Mb/s of random numbers can be sustained regardless of manufacturer when using all 8 banks in a single channel
  - A maximum throughput of A: 179.4, B: 134.5, C: 179.4 Mb/s was observed
  - Average throughput across all manufacturers: 108.9 Mb/s
  - Maximum throughput achieved (in a device with 4 DRAM channels): 717.4 Mb/s

Results: Key Properties of a TRNG

- Low Latency:
  - D-RaNGe latency is directly related to DRAM access latency
  - Maximum latency for 64 bits of random data: 960 ns
  - Minimum latency for 64 bits of random data: 220 ns
  - Empirical minimum latency for 64 bits of random data: 100ns
- Low system interference:
  - D-RaNGe is highly flexible in terms of system interference versus high throughput
  - The overhead of acquiring exclusive access rights to DRAM rows results in 0.018% storage overhead cost
  - Maximum average throughput with no significant impact of system performance: 83.1 Mb/s
- Low energy consumption:
  - Cost of generating a random data: 4.4 nJ/bit

Overview

Motivation

Goal

DRAM Background

D-RaNGe

Testing Environment

Results

Comparison to Prior Work

Summary and Conclusion

Strengths

Limitations

Discussion

Comparison to Prior Works: DRAM Command Scheduling

- Idea: Use latency of DRAM accesses as source of randomness
- Problem: DRAM access latency is not fully non-deterministic
- Maximum throughput: 3.4 Mb/s
- D-RaNGe outperforms this approach by 211x in terms of throughput
- Latency for 64 bits of random data: 18µs

Comparison to Prior Works: DRAM Data Retention

- Idea: Exploit DRAM cell leakage by increasing the refresh interval
- Data Retention Errors are non-deterministic
- Latency: 40s
- Throughput: 0.05 Mb/s
- Energy consumption: 6.8 mJ/bit

Comparison to Prior Works: DRAM startup values

- Idea: Sample start-up values of DRAM cells
- Non-deterministic entropy source
- Not capable of continuous throughput
- Latency and power consumption are very hard to estimate

Overview

Summary and Conclusion

Motivation: High throughput and low latency True Random Number Generators are a key component for encryption and randomized algorithms. Many commodity devices do not posses dedicated TRNG hardware but have DRAM.

Current Problem: Prior approach to TRNG designs based in DRAM either 1) exploit a fundamentally non-deterministic entropy source or 2) are too slow for continuous high-throughput operations.

Goal: A novel approach to TRNGs that uses existing DRAM devices with 1) low implementation cost, 2) low latency and 3) high throughput

Key Idea: Exploit non-determinism in DRAM cells' activation failures to generate true random numbers.

Evaluation: D-RaNGe was implemented and tested on 282 real LPDDR4 DRAM devices showing a remarkably high peak throughput (717.4 Mb/s) and very low latency (100ns).

Overview

- Motivation
- Goal
- DRAM Background
- D-RaNGe
- Testing Environment
- Results
- Comparison to Prior Work
- Summary and Conclusion
- **Strengths**
- Limitations
- Discussion

Strengths

- No extra hardware is required to implement D-RaNGe in most cases
- D-RaNGe can be scaled according to application requirements
- No postprocessing is required as RNG cells return unbiased output
- RNG cells maintain high entropy and activation failure probability across system reboots
- Shifts the current computing paradigm towards a data centric architecture

Overview

Limitations

- The effect of long term ageing on RNG cells was not analyzed
- D-RaNGe was only tested in a narrow range of operating temperatures
- Effects of different voltages on RNG cells were not considered
- Memory channels could become a bottleneck for memory intensive applications
- Each DRAM device has to be profiled individually

Overview

- Motivation
- Goal
- DRAM Background
- D-RaNGe
- Testing Environment
- Results
- Comparison to Prior Work
- Summary and Conclusion
- Strengths
- Limitations
- Discussion

Do you see some other limitations with D-RaNGe?
How can we improve it?

Could we exploit some other widely available hardware to host a TRNG? What would the advantages and disadvantages be?

What does it take for D-RaNGe to be commercially available? What must happen for D-RaNGe to become a standard service on every computer?