Crypto 2021

A Logarithmic Lower Bound for Oblivious RAM (for all parameters)

August 20, 2021

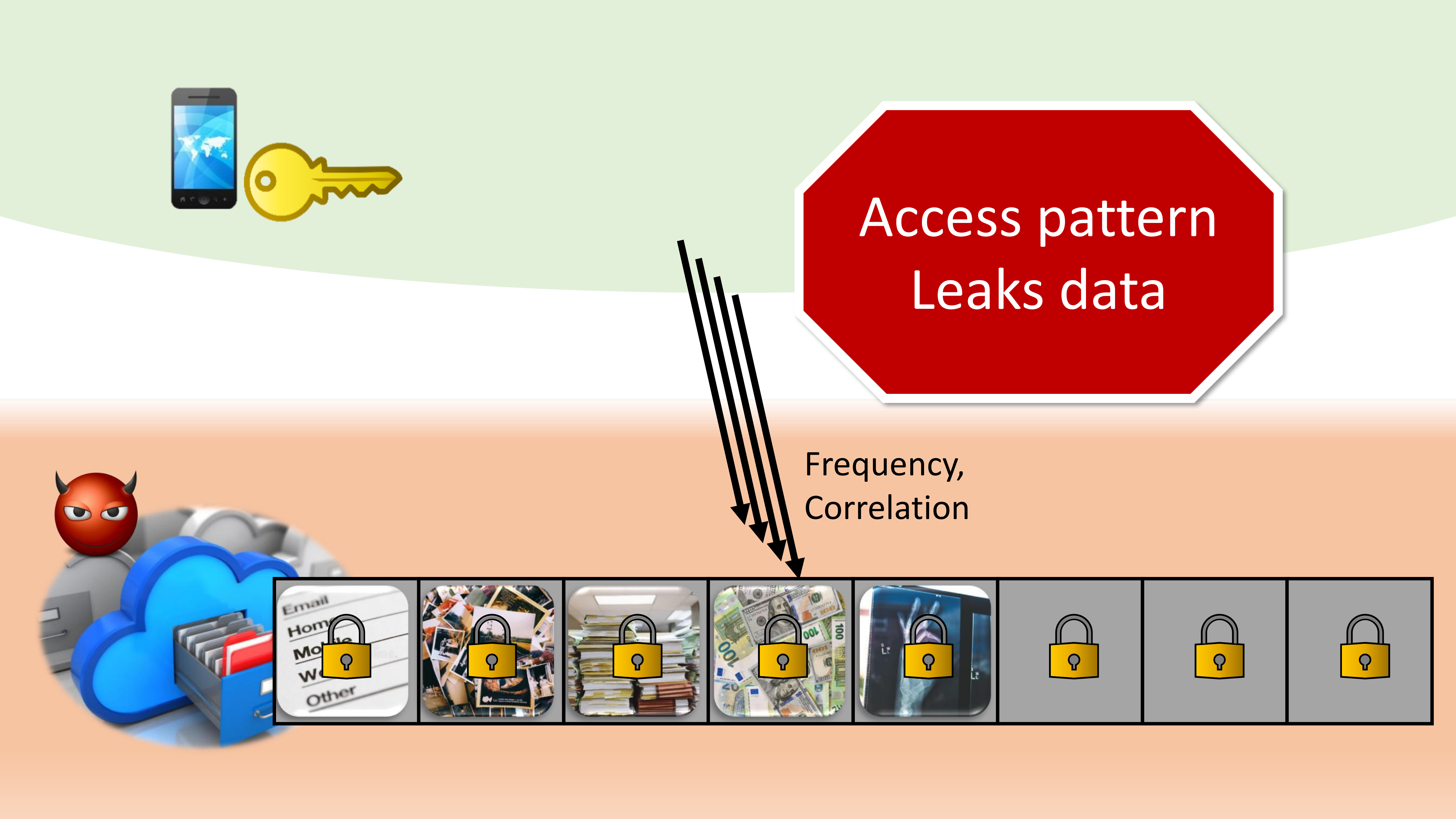
Wei-Kai Lin

Cornell University

Ilan Komargodski

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

NTT Research


Access pattern
Leaks data
Frequency,
Correlation

ORAM, Correctness

[Goldreich-Ostrovsky ’87,96]

“Online”:
Answer a query
before next

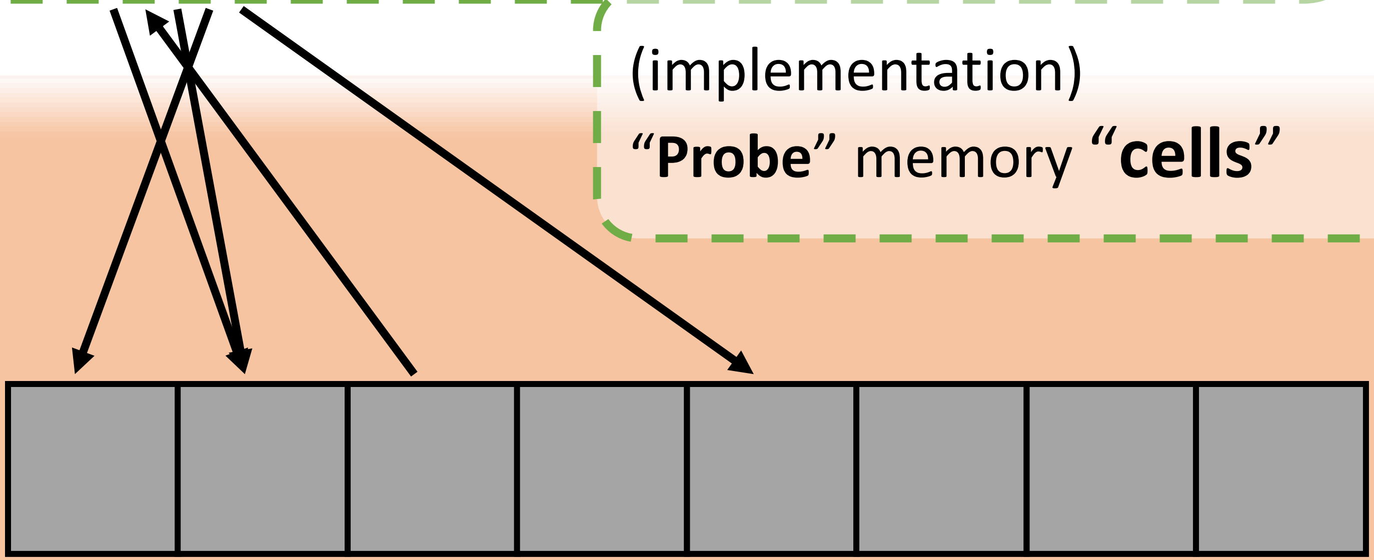
ORAM operations (array):

* Update(i, )
* Query(i)

(implementation)

“**Probe**” memory “**cells**”

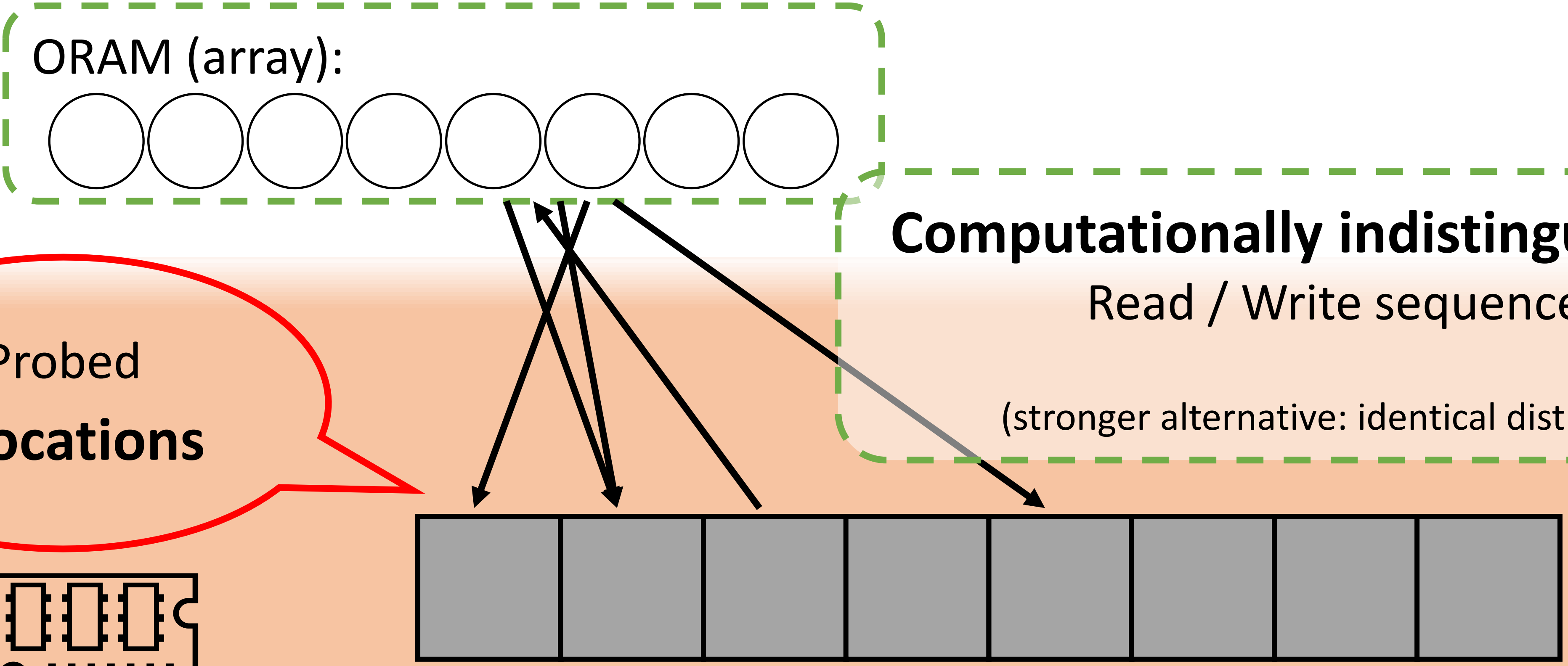
[Yao ‘81]



ORAM, Security

[Goldreich-Ostrovsky ’87,96]

Any sequence of
Update / Query

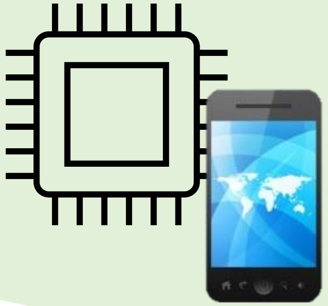


Computationally indistinguishable
Read / Write sequence

(stronger alternative: identical distribution)

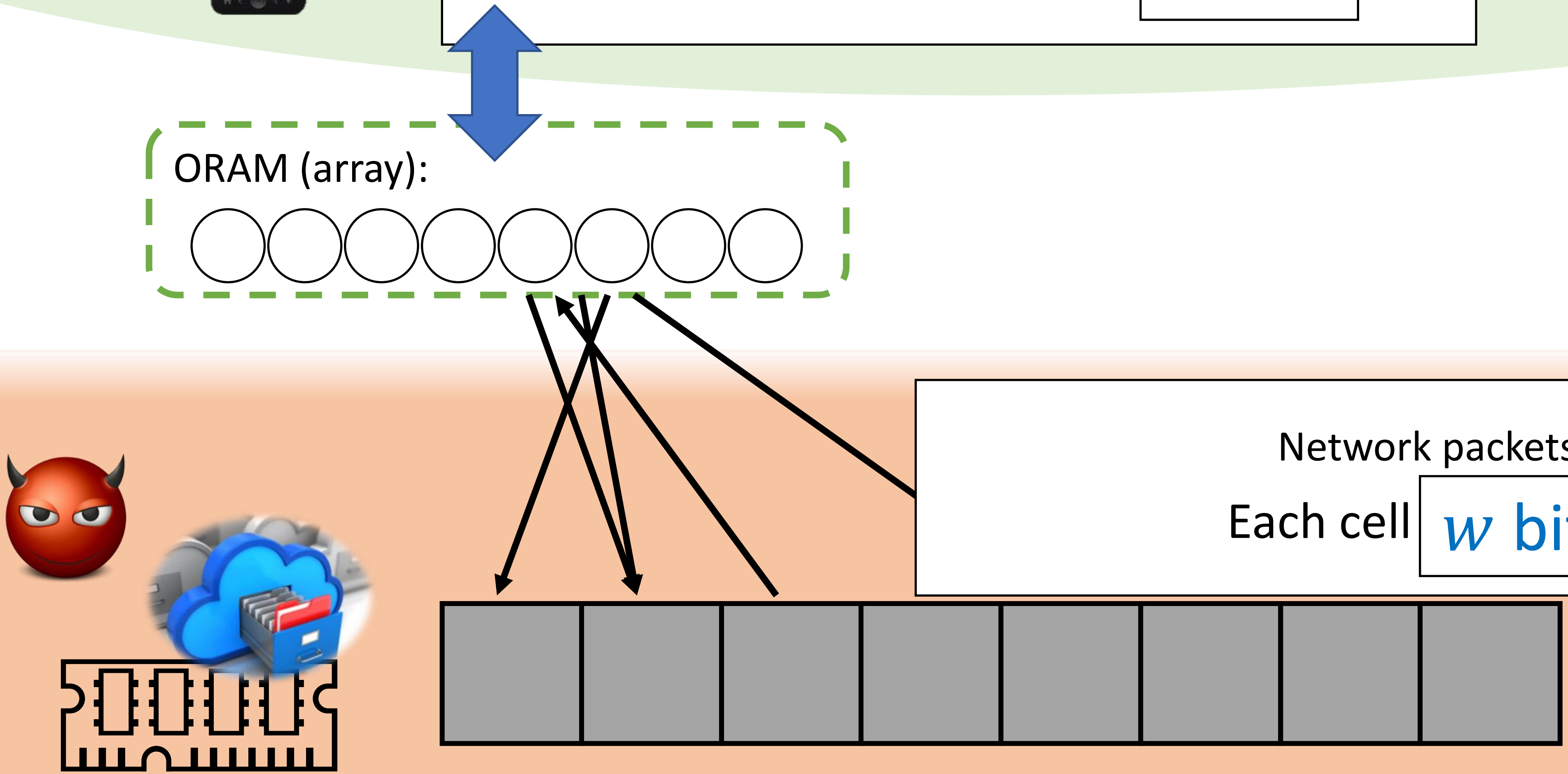
Probed
locations

ORAM, Parameters



64-bit program:

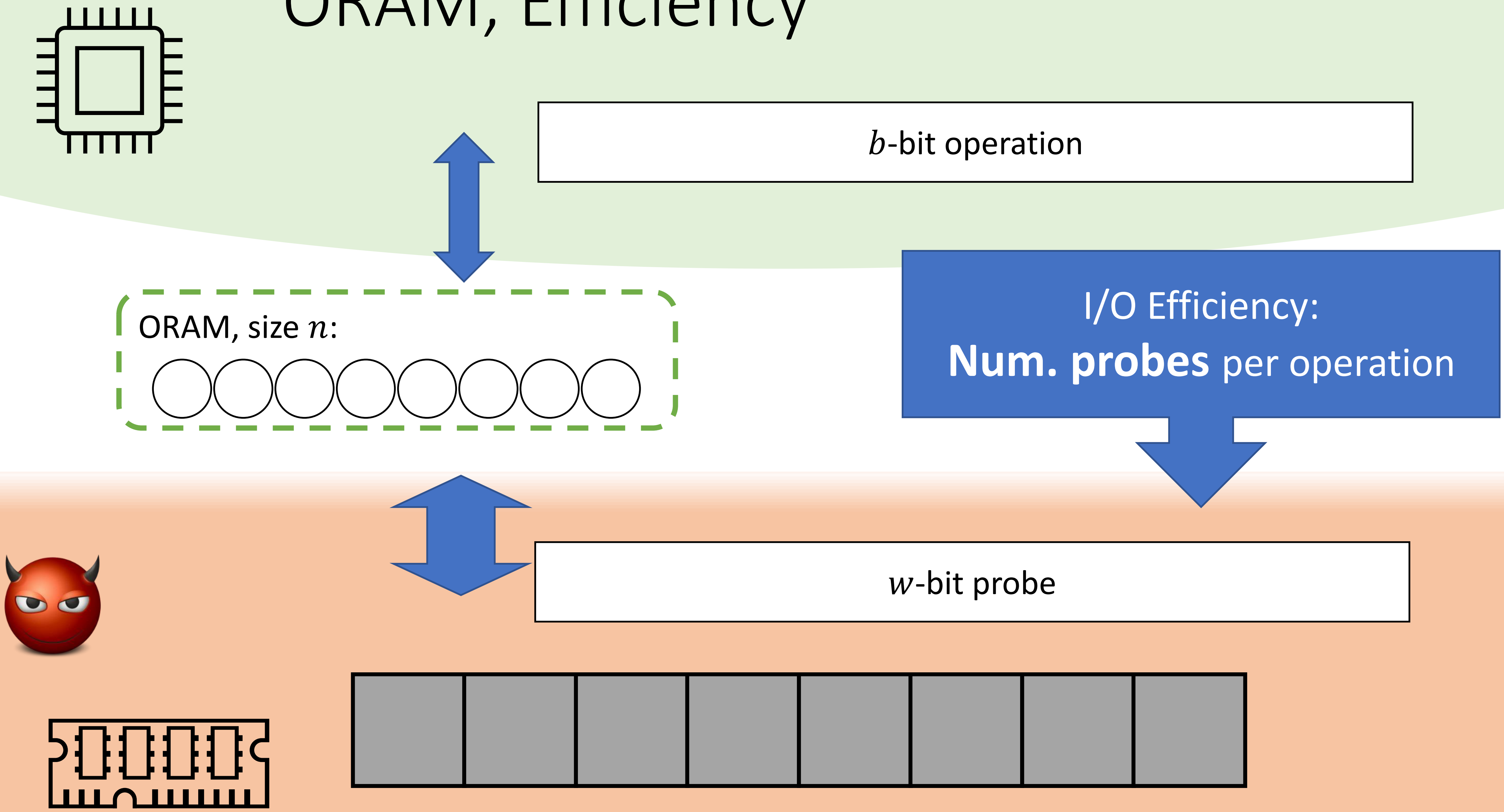
Array of n entries, each b bits

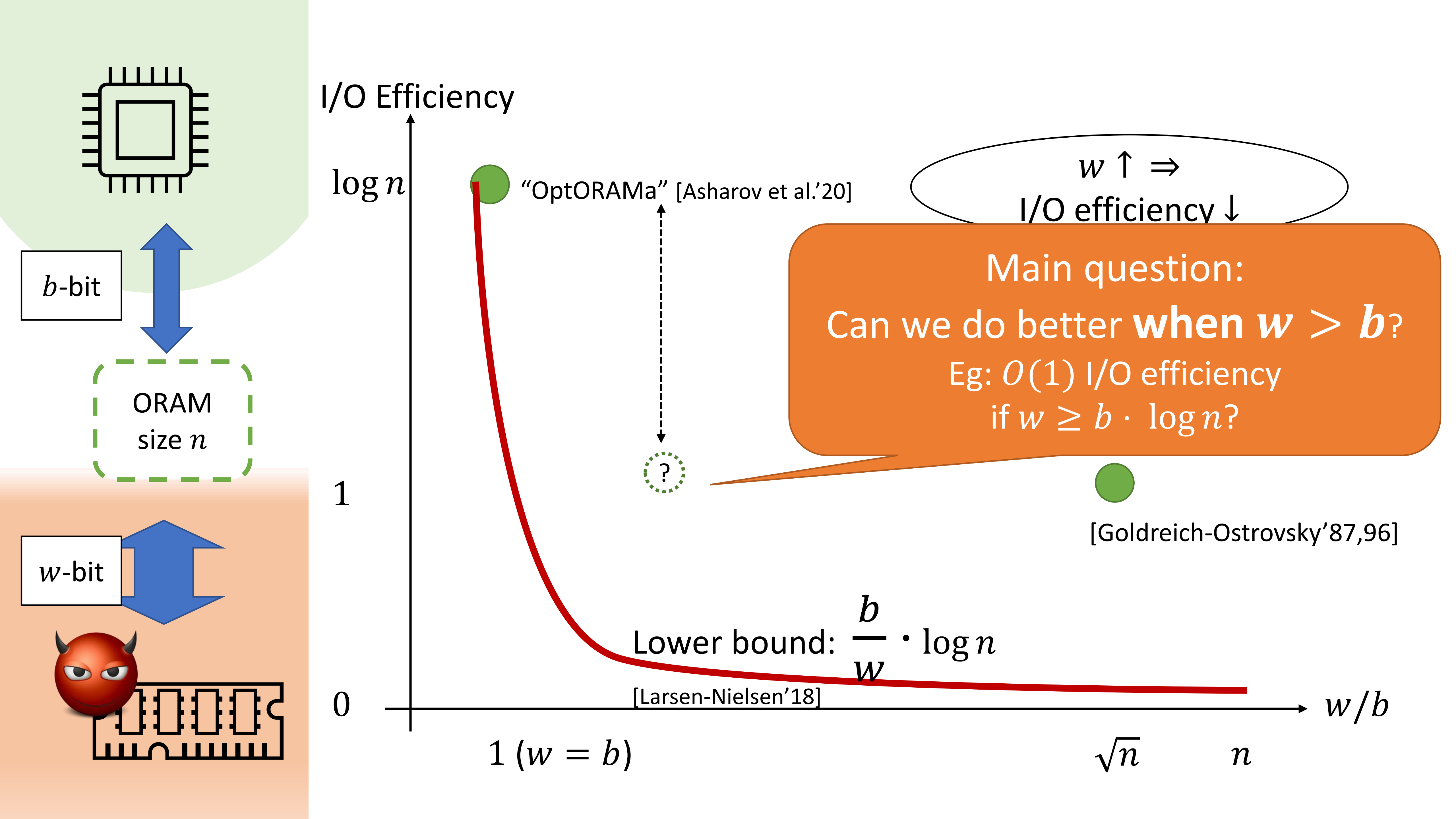


Network packets:

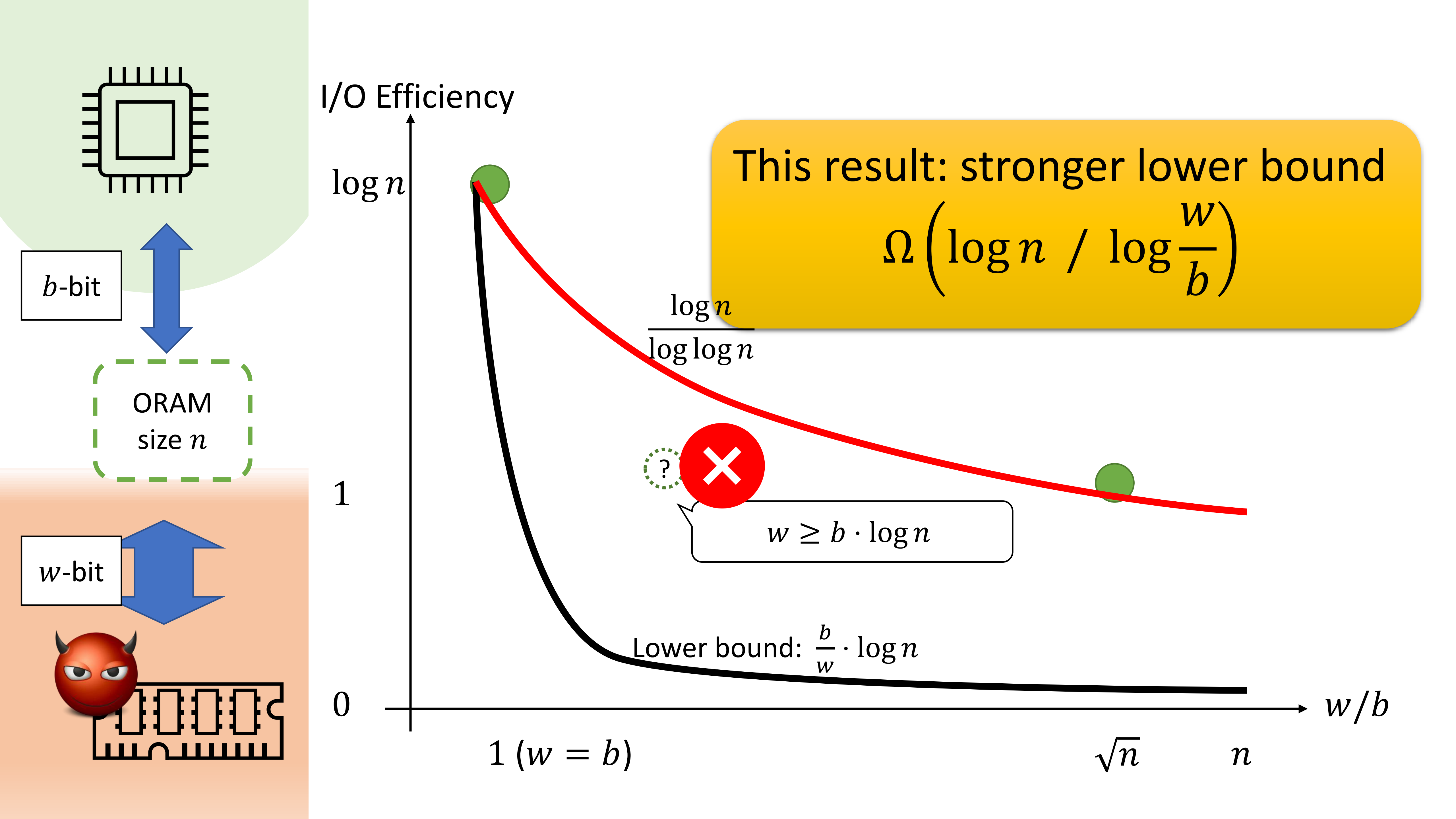
Each cell w bits

ORAM, Efficiency

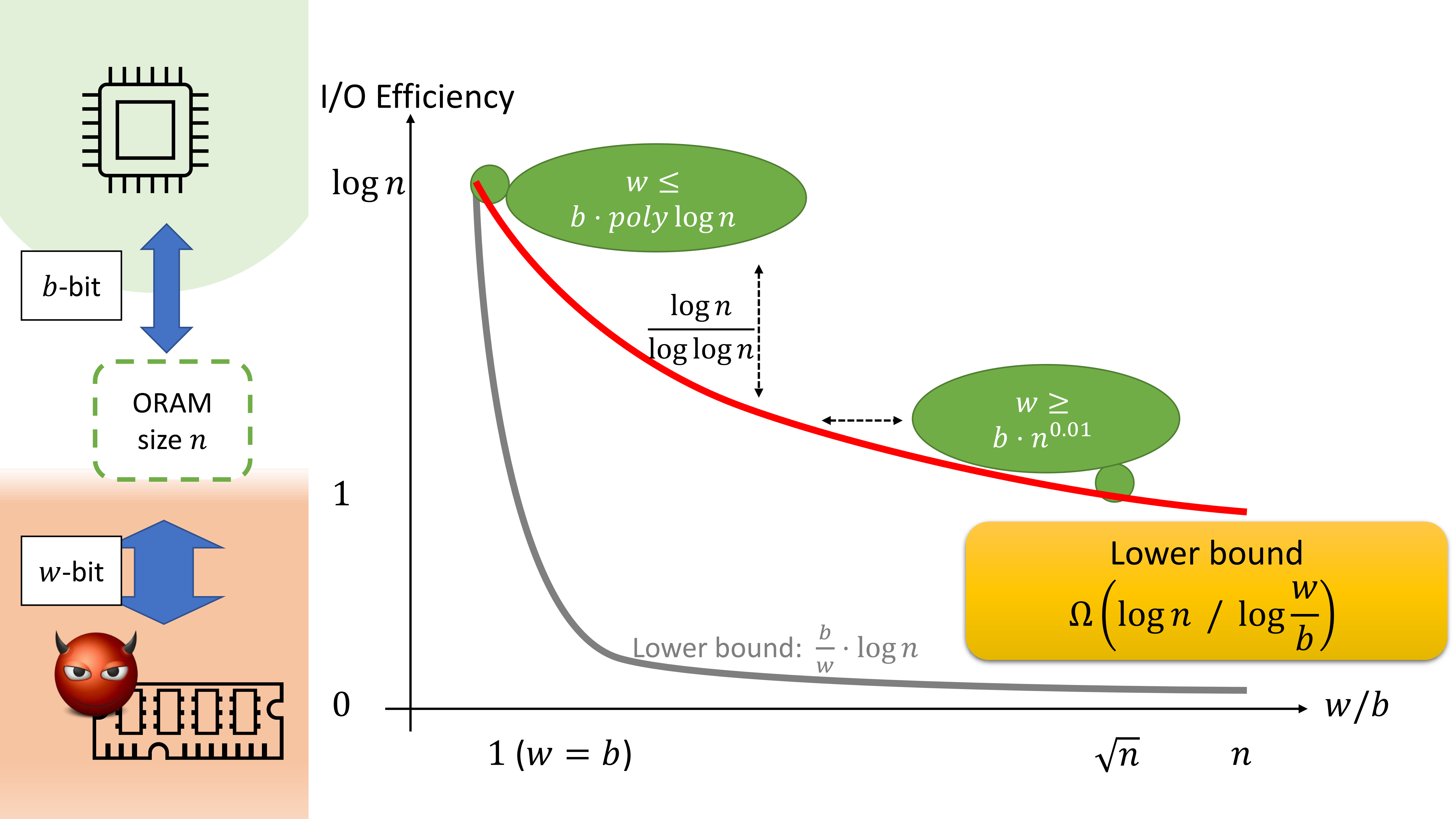




b-bit
ORAM
size n
w-bit
I/O Efficiency
$\log n$
1
0
"OptORAMa" [Asharov et al.'20]
?
$w \uparrow \Rightarrow$
I/O efficiency $\downarrow$
Main question:
Can we do better **when $w > b$**?
Eg: $O(1)$ I/O efficiency
if $w \geq b \cdot \log n$?
[Goldreich-Ostrovsky'87,96]
Lower bound: $\frac{b}{w} \cdot \log n$
[Larsen-Nielsen'18]
w/b
$1\ (w = b)$
$\sqrt{n}$
n

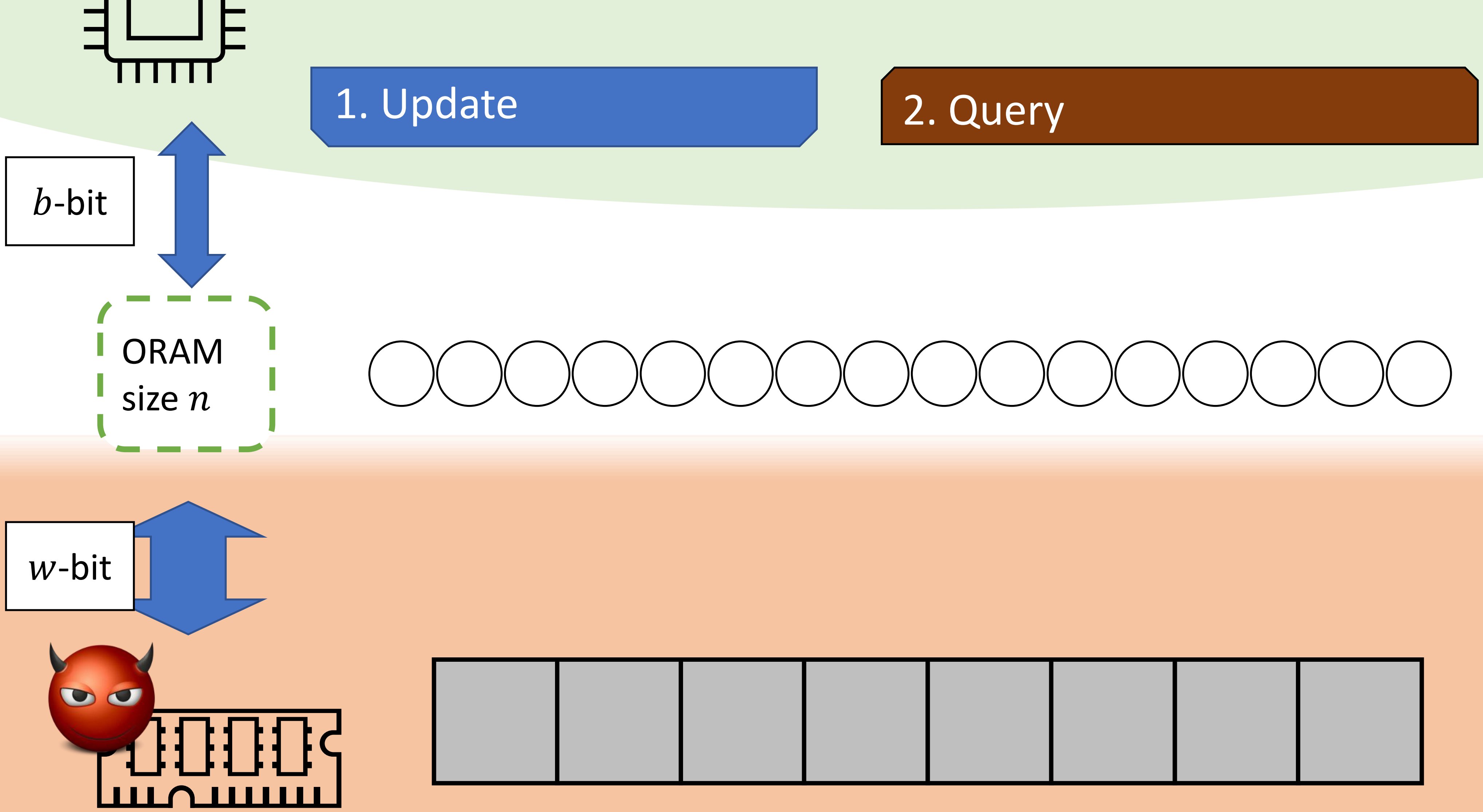


I/O Efficiency
$\log n$
1
0
b-bit
ORAM size n
w-bit
This result: stronger lower bound
$\Omega\left(\log n \ / \ \log \frac{w}{b}\right)$
$\frac{\log n}{\log \log n}$
?
$w \geq b \cdot \log n$
Lower bound: $\frac{b}{w} \cdot \log n$
w/b
$1 \ (w = b)$
$\sqrt{n}$
n

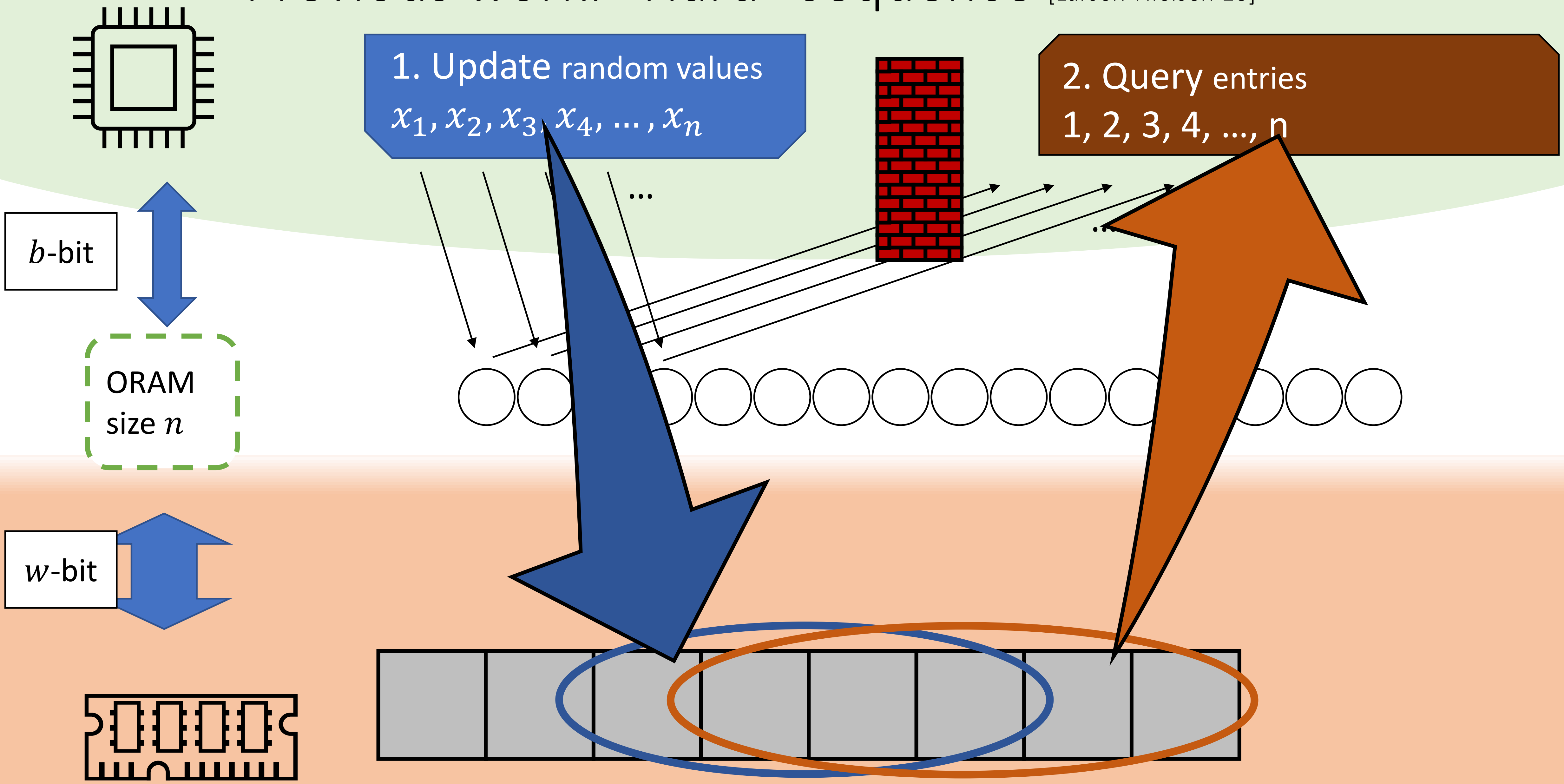


b-bit
ORAM size n
w-bit
I/O Efficiency
$\log n$
1
0
$w \leq b \cdot poly \log n$
$\frac{\log n}{\log \log n}$
$w \geq b \cdot n^{0.01}$
Lower bound: $\frac{b}{w} \cdot \log n$
Lower bound $\Omega\left(\log n \ / \ \log \frac{w}{b}\right)$
w/b
$1 \ (w = b)$
$\sqrt{n}$
n

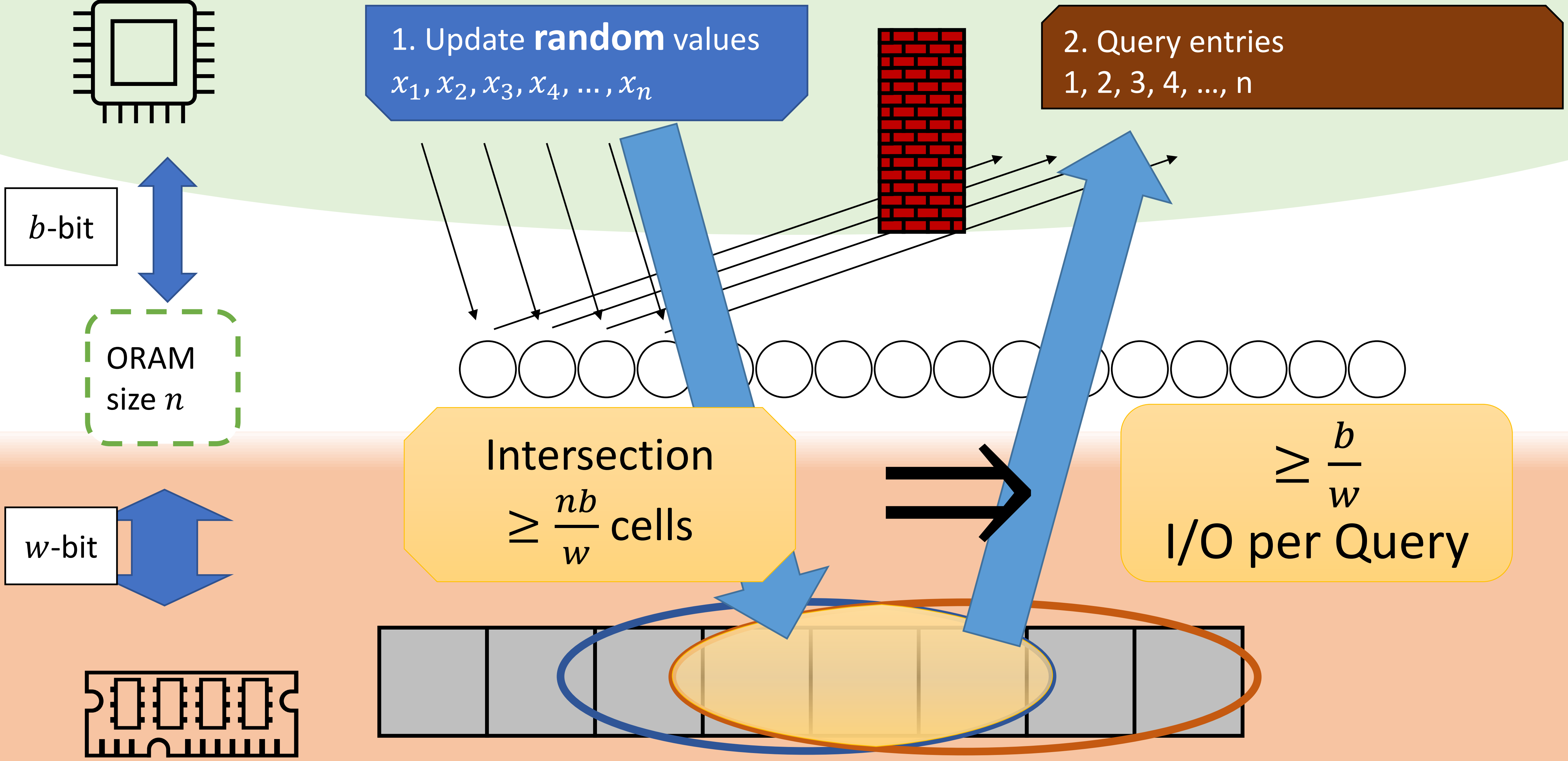
Lower Bound Proof



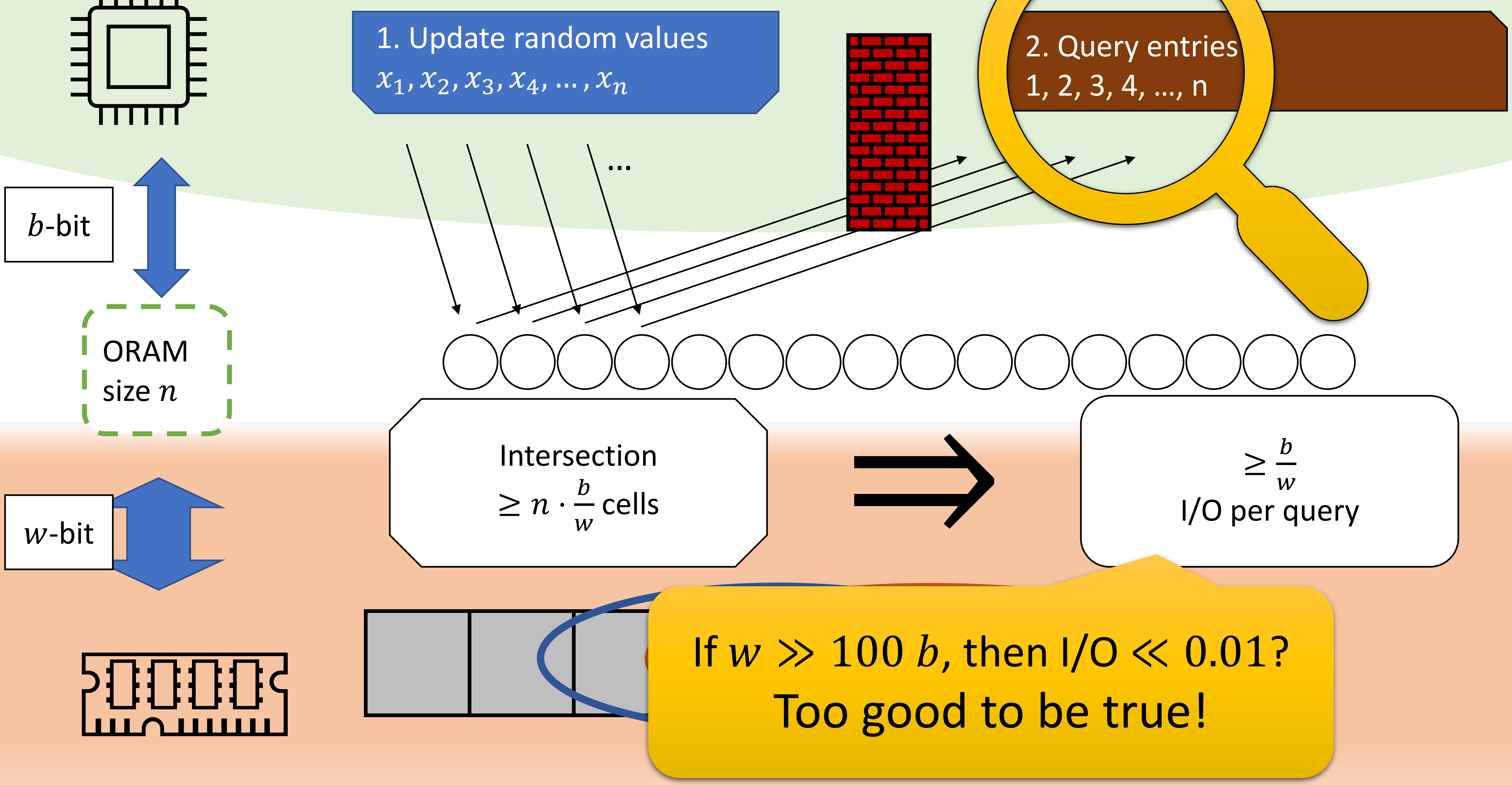
Previous work: “Hard” sequence [Larsen-Nielsen’18]



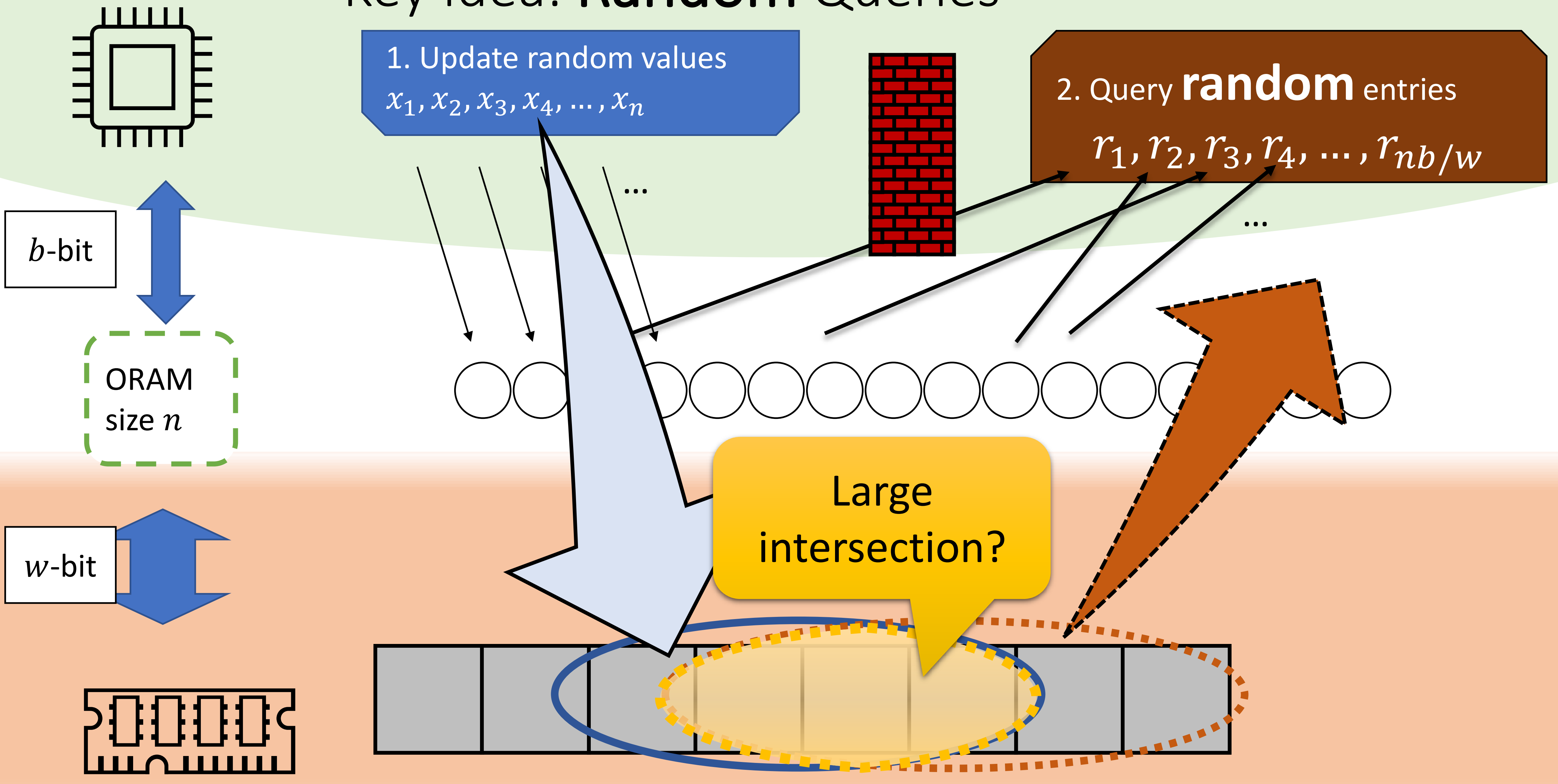
Previous work: “Hard” sequence



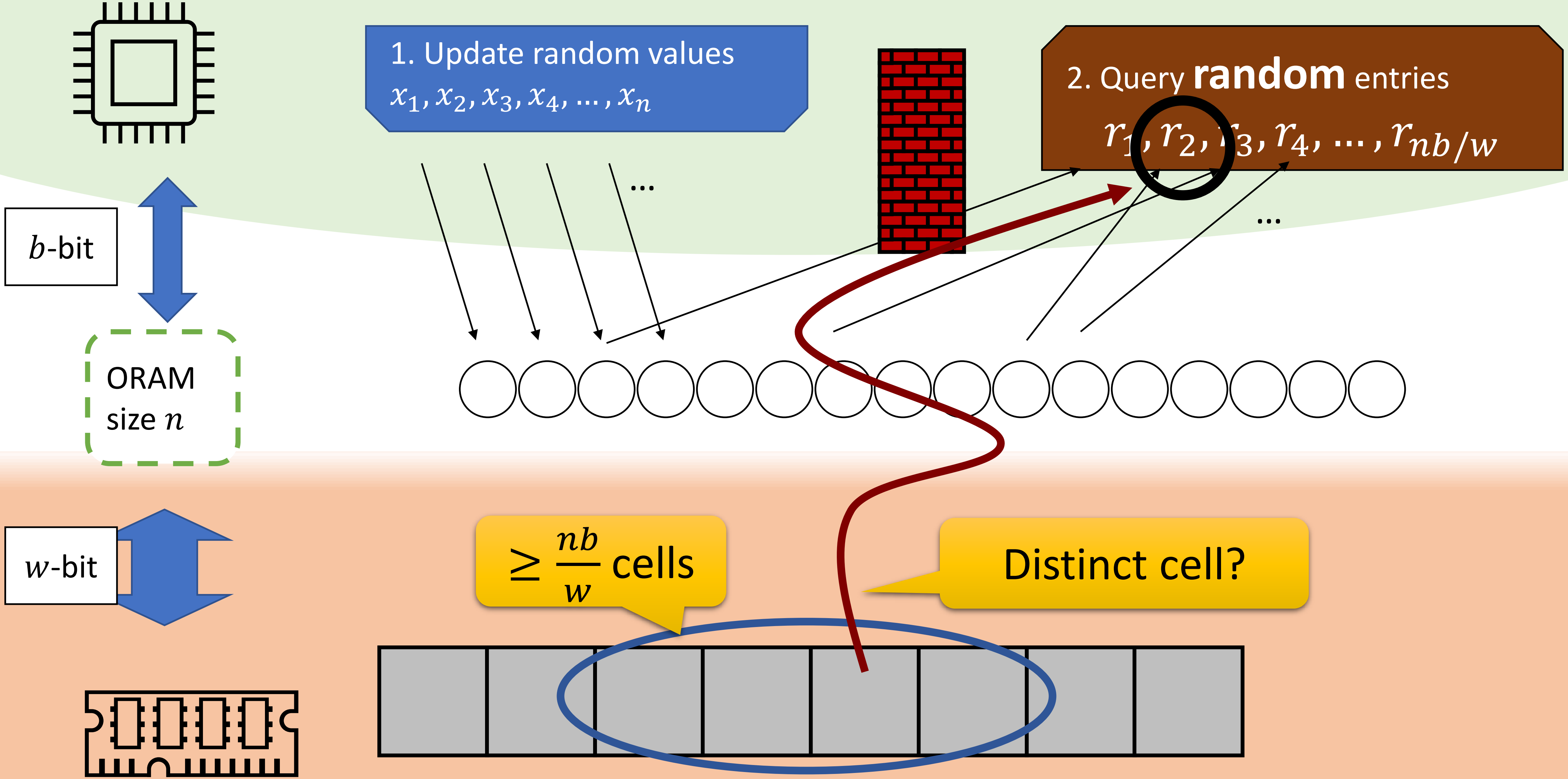
Key Idea: **Random** Queries



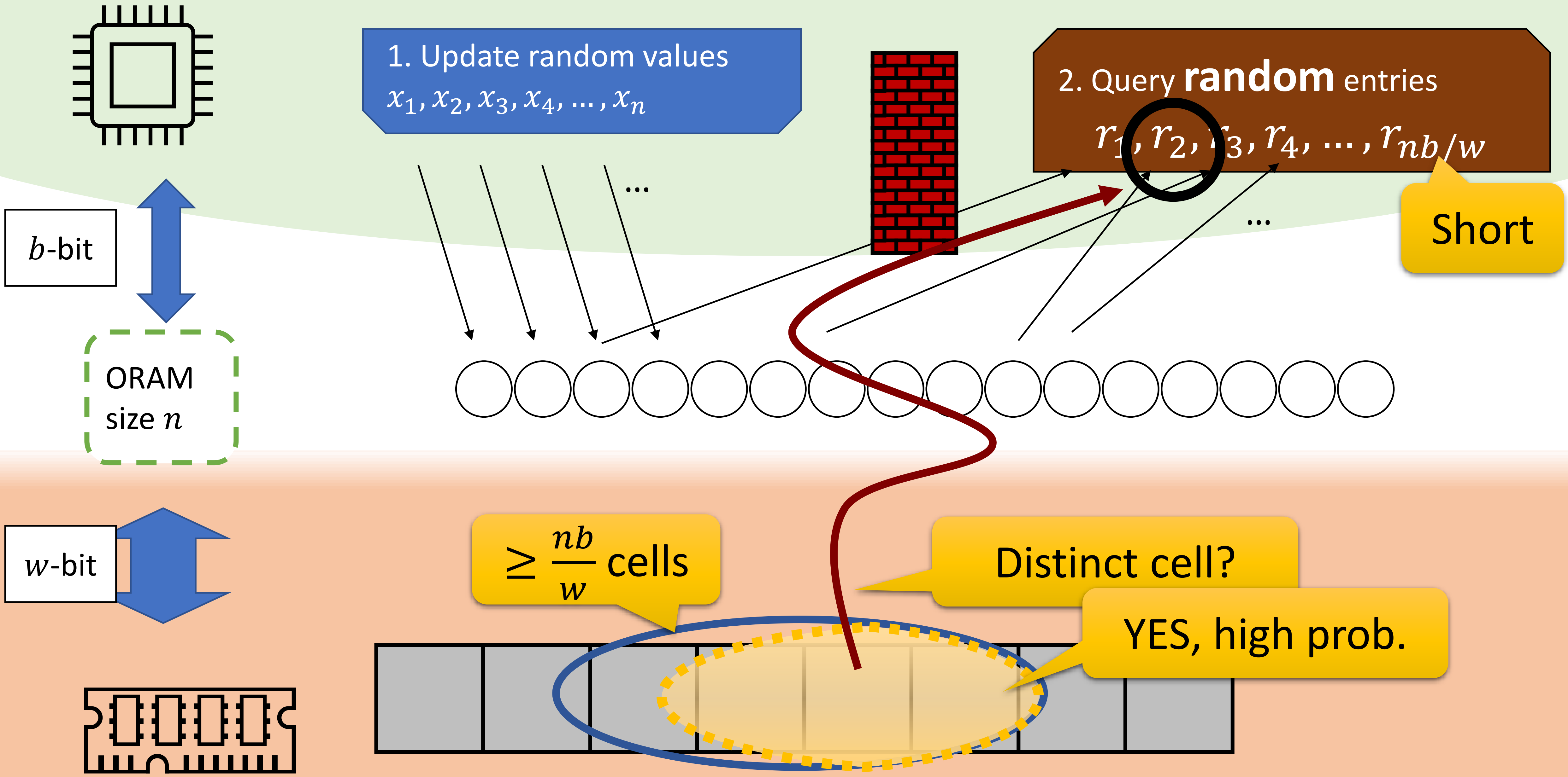
Key Idea: **Random** Queries



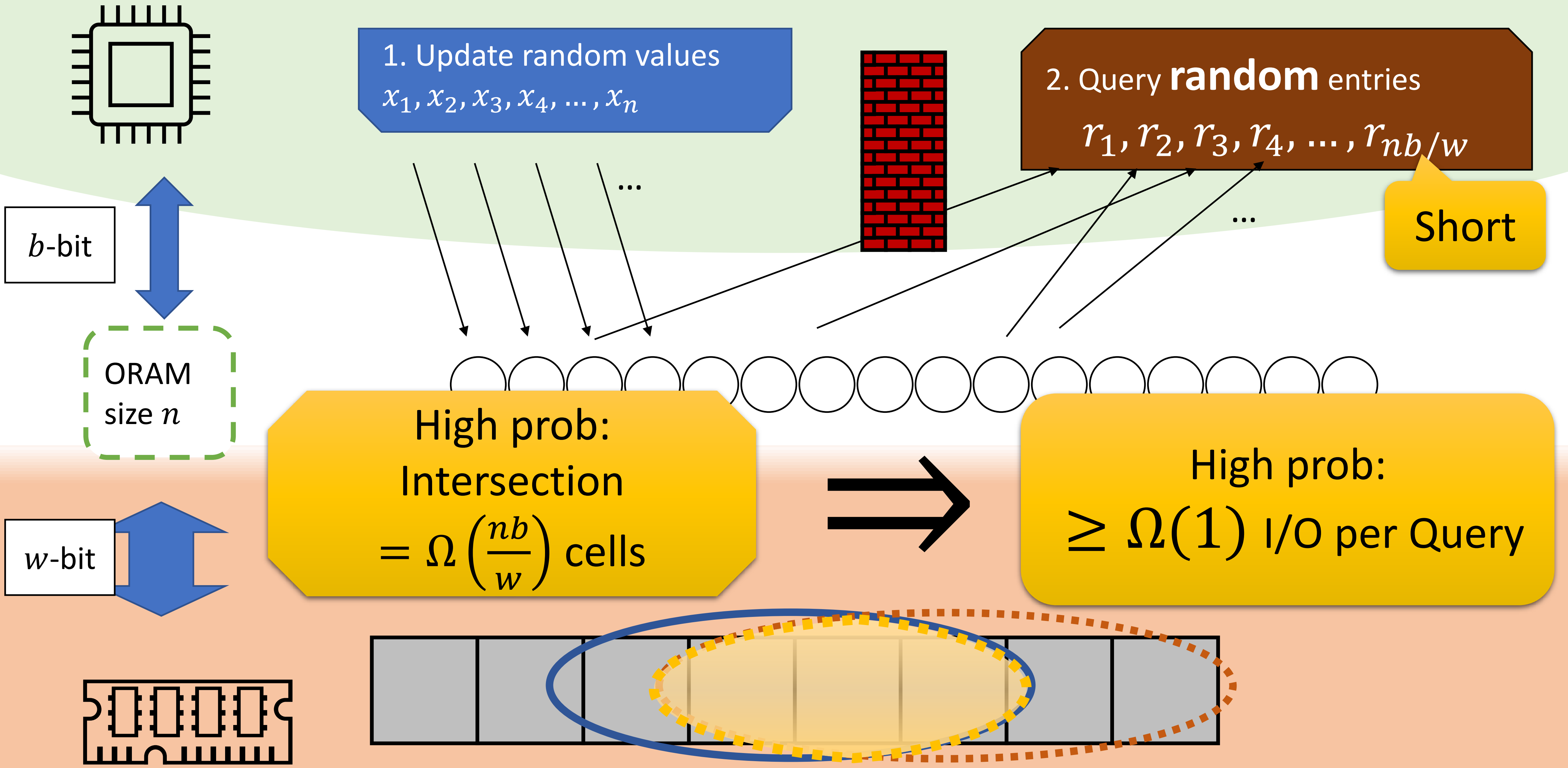
Main Technical **Lemma**



Main Technical **Lemma**

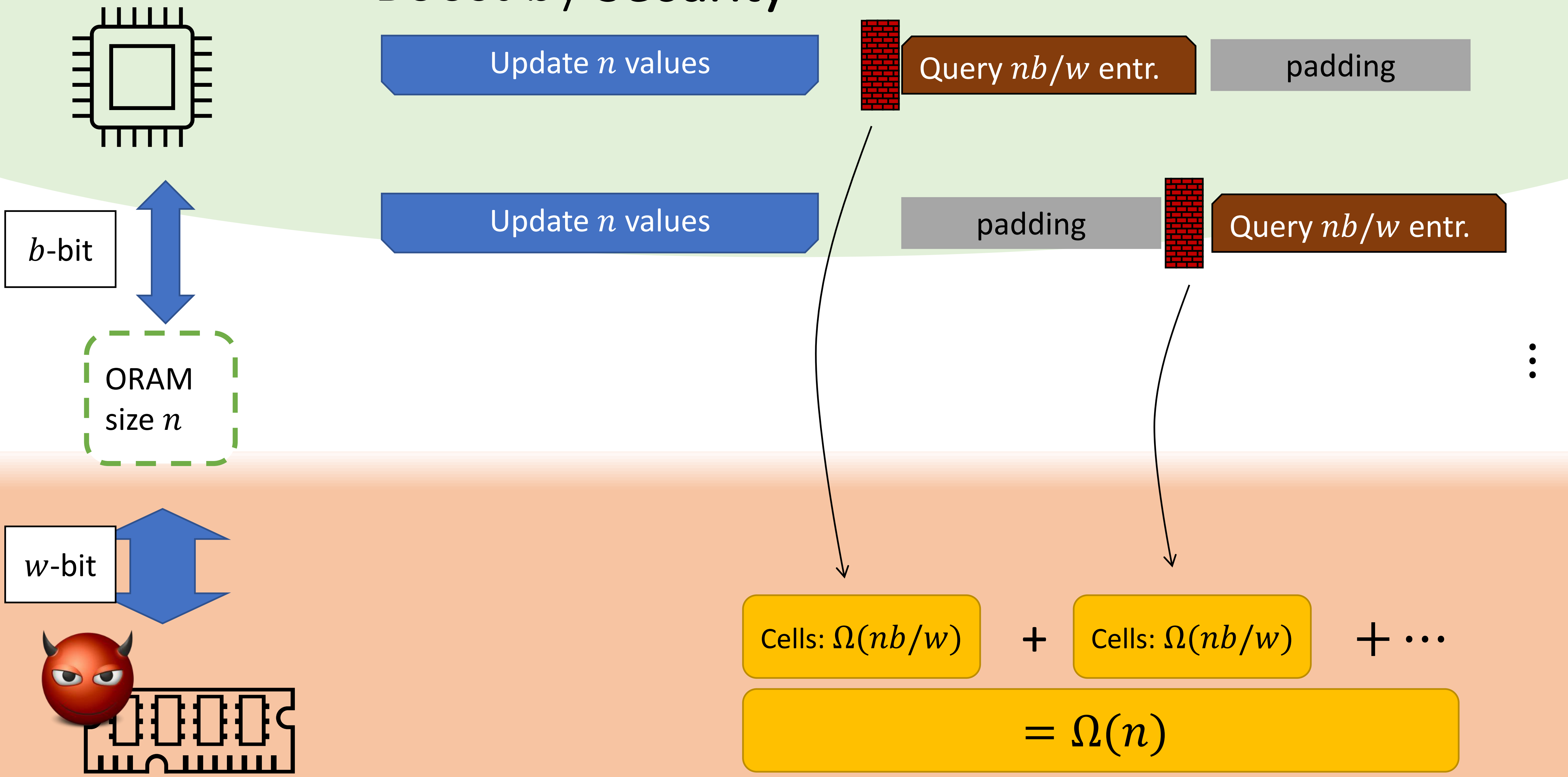


Main Technical **Lemma**

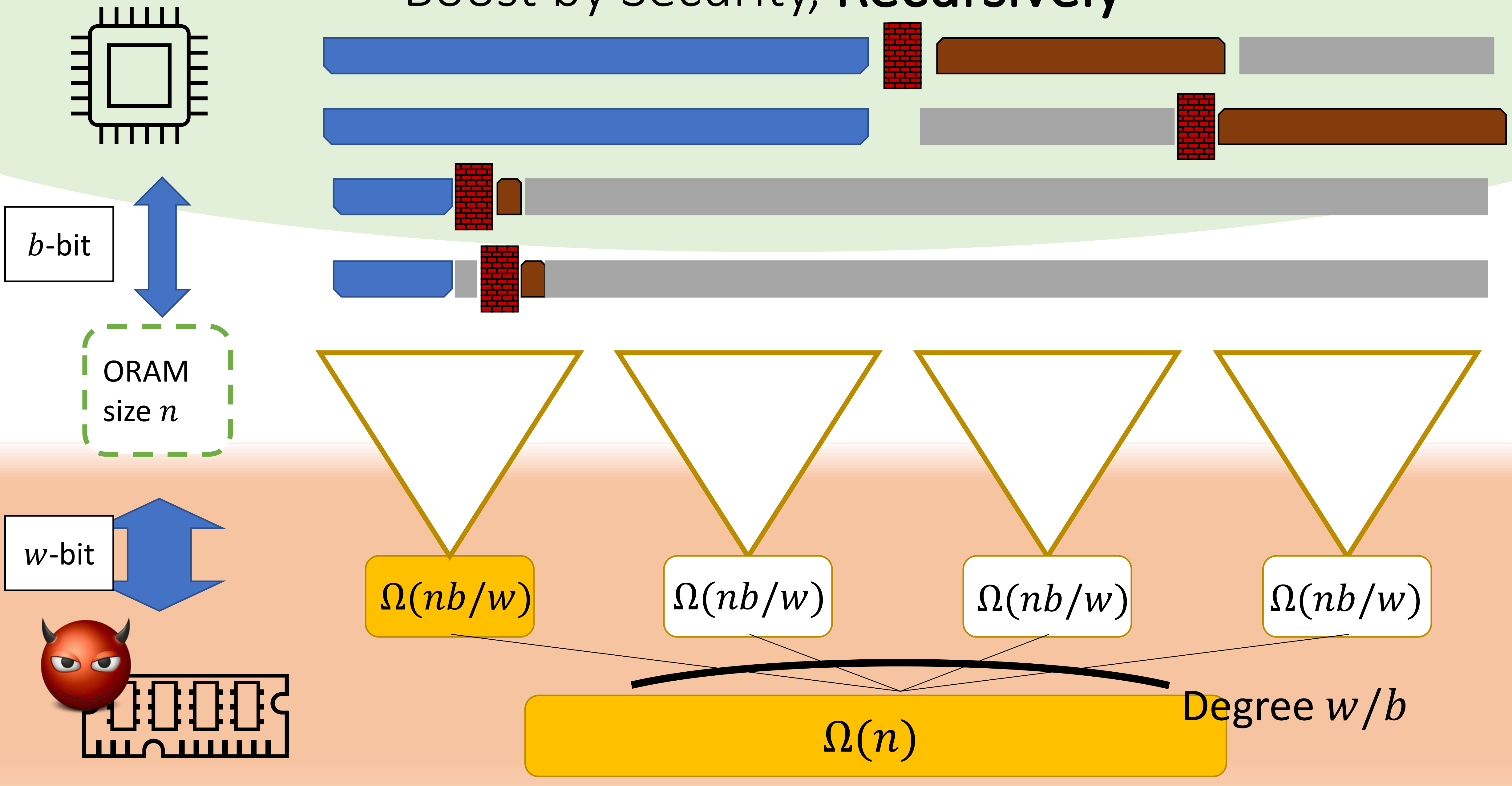


Proved by compression argument

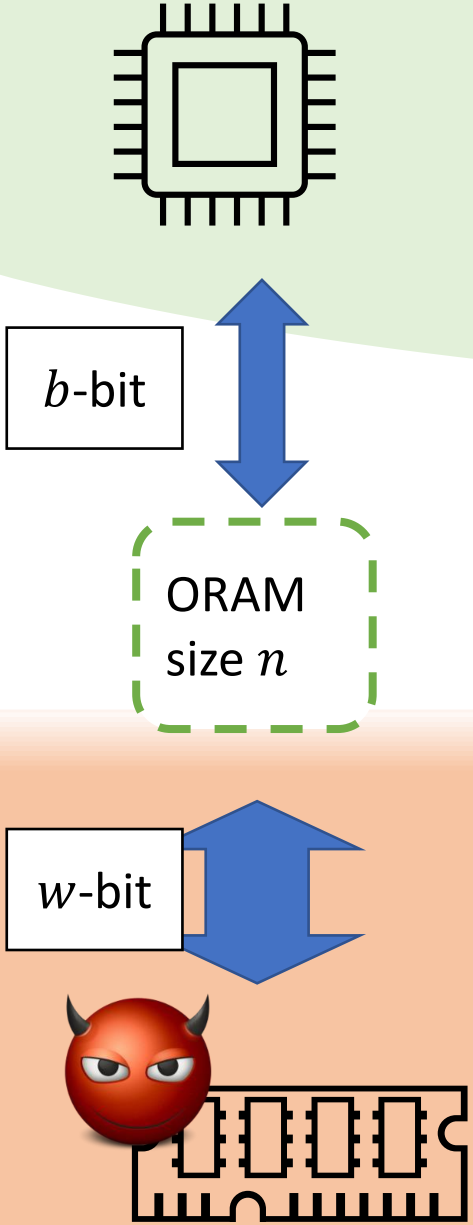
Boost by **Security**



Boost by Security, **Recursively**



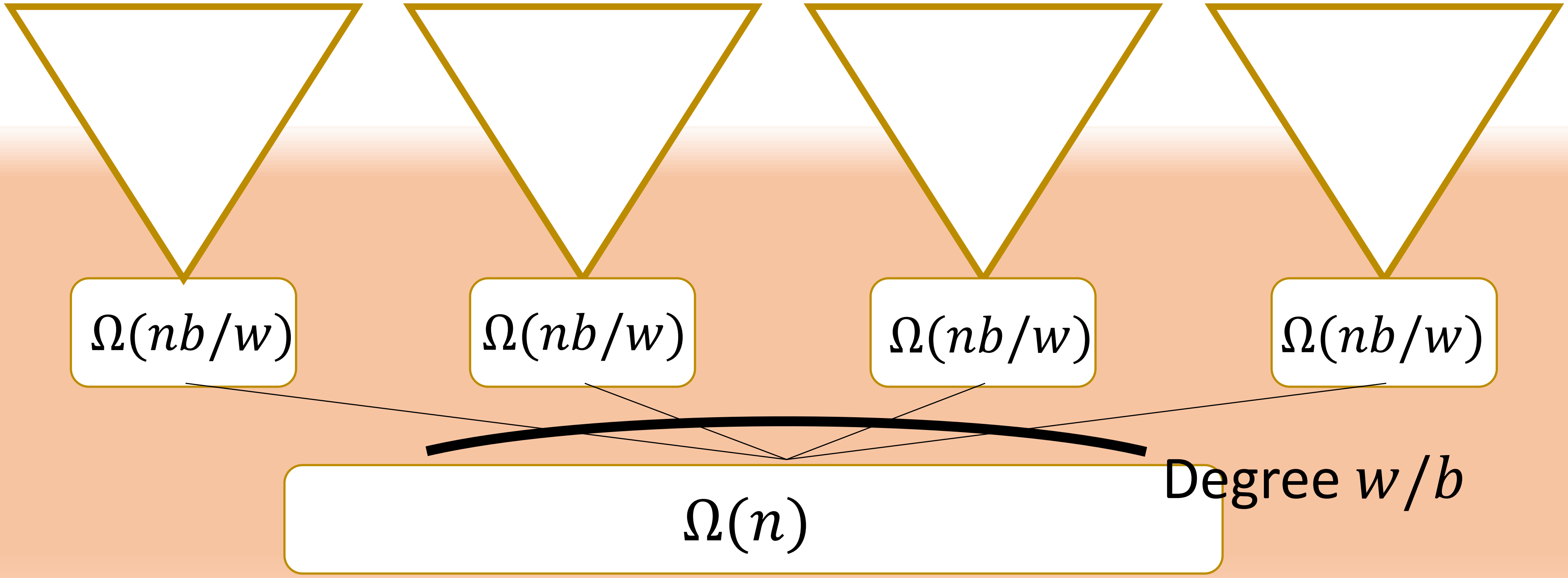
Boost by Security, **Recursively**

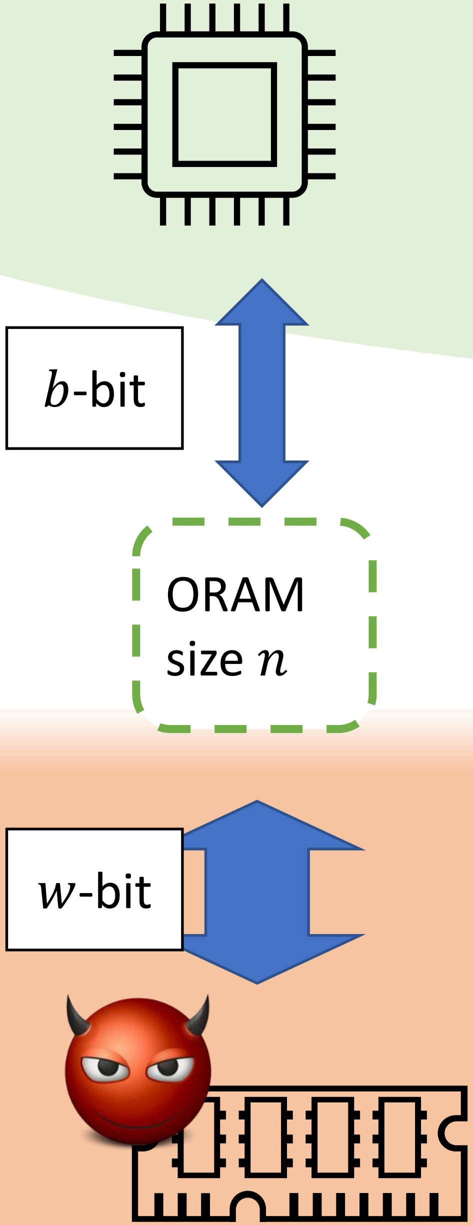


Tree height = $\log_{w/b} n$

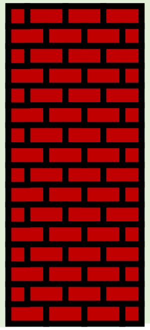
$\Rightarrow$ #I/O = $\Omega(n \cdot \log_{w/b} n)$

per n Query/Update





1. Update random values $x_1, x_2, x_3, x_4, \dots, x_n$



2. Query **random** entries $r_1, r_2, r_3, r_4, \dots, r_{nb/w}$

Main lemma (this hard sequence):

With high prob: intersection $= \Omega\left(n \cdot \frac{b}{w}\right)$ cells

Main result (any ORAM):

Any $b \geq w$, I/O = $\Omega\left(\log n \;/\; \log \frac{b}{w}\right)$

- Unconditional (not “balls-and-bins” model)
- Computational (ORAM may use any crypto)

Challenge to main lemma

1. Update random values
$x_1, x_2, x_3, x_4, \dots, x_n$

2. Query **random** entries
$r_1, r_2, r_3, r_4, \dots, r_{nb/w}$

With high prob:
intersection $= \Omega\left(n \cdot \frac{b}{w}\right)$ cells

Suppose not, then exists ORAM:
Intersection $\leq 0.01\, n \cdot \frac{b}{w}$
$\Rightarrow$ Can compress random values
$x_1, x_2, x_3, x_4, \dots, x_n$
To $< \mathbf{0.99}\, \boldsymbol{nb}$ bits (impossible)

Alice (impossible compress)

[Pătraşcu,Demaine'06]

1. If Intersection of $(x_1, \dots, x_n\ ; r_1, r_2, \dots, r_{nb/w})$ is large, then output $(x_1, \dots, x_n)$ directly;
   Else, continue.
2. **Write** small Intersection (of cell contents, $0.01nb$ bits)
3. Pick random t from 1 to nb/w.
4. For each $\boldsymbol{i}$ from 1 to n:
   If Query$(r_1, r_2, \dots, r_{t-1}, \boldsymbol{i})$
   can NOT be answered by small Intersection,
   then **Write** x_i

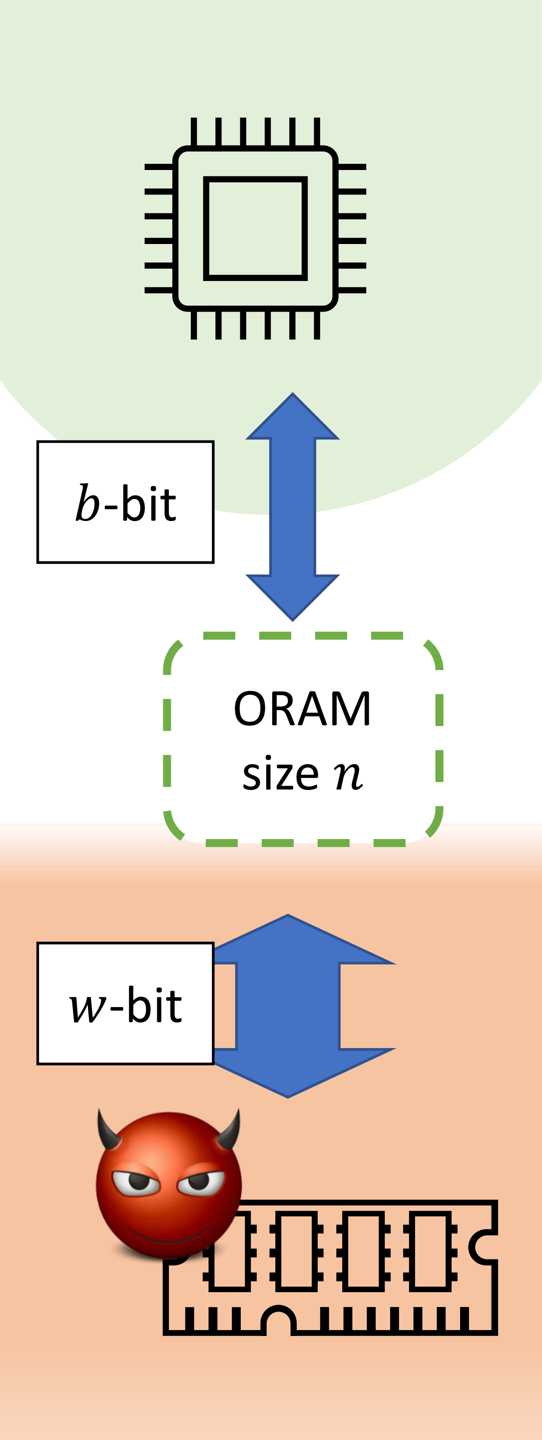
Analysis, simplified

- X, Y independent random variables
- Y^* random variable, independent and identically distributed to Y
- $f(x, y)$ arbitrary Boolean function

Then:

$$\Pr[f(X, Y^*) = 1 \mid f(X, Y) = 1] \quad \geq \quad \Pr[f(X, Y) = 1]$$

A “win” makes it more likely to “win”



Main result (any ORAM):

Any $w \geq b$, I/O = $\Omega\left(\log n \;/\; \log \frac{w}{b}\right)$

(extends to multi-server setting)

Open problems:

- Remaining gap (for computational security)
- Lower/upper bound for
  - Weaker notions (eg, differential-private ORAMs)
  - Stronger notions (eg, statistical security)

Related **new results**:

- ORAM with Worst-Case Logarithmic Overhead (Crypto2021)
- Optimal Oblivious *Parallel* RAM

Thank you!