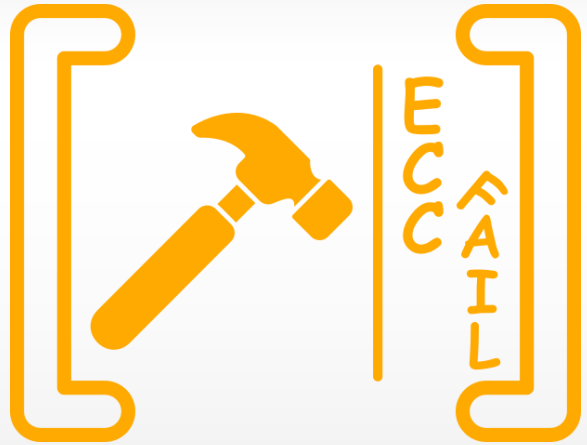
ECC.fail: Mounting Rowhammer Attacks on DDR4 Servers with ECC Memory

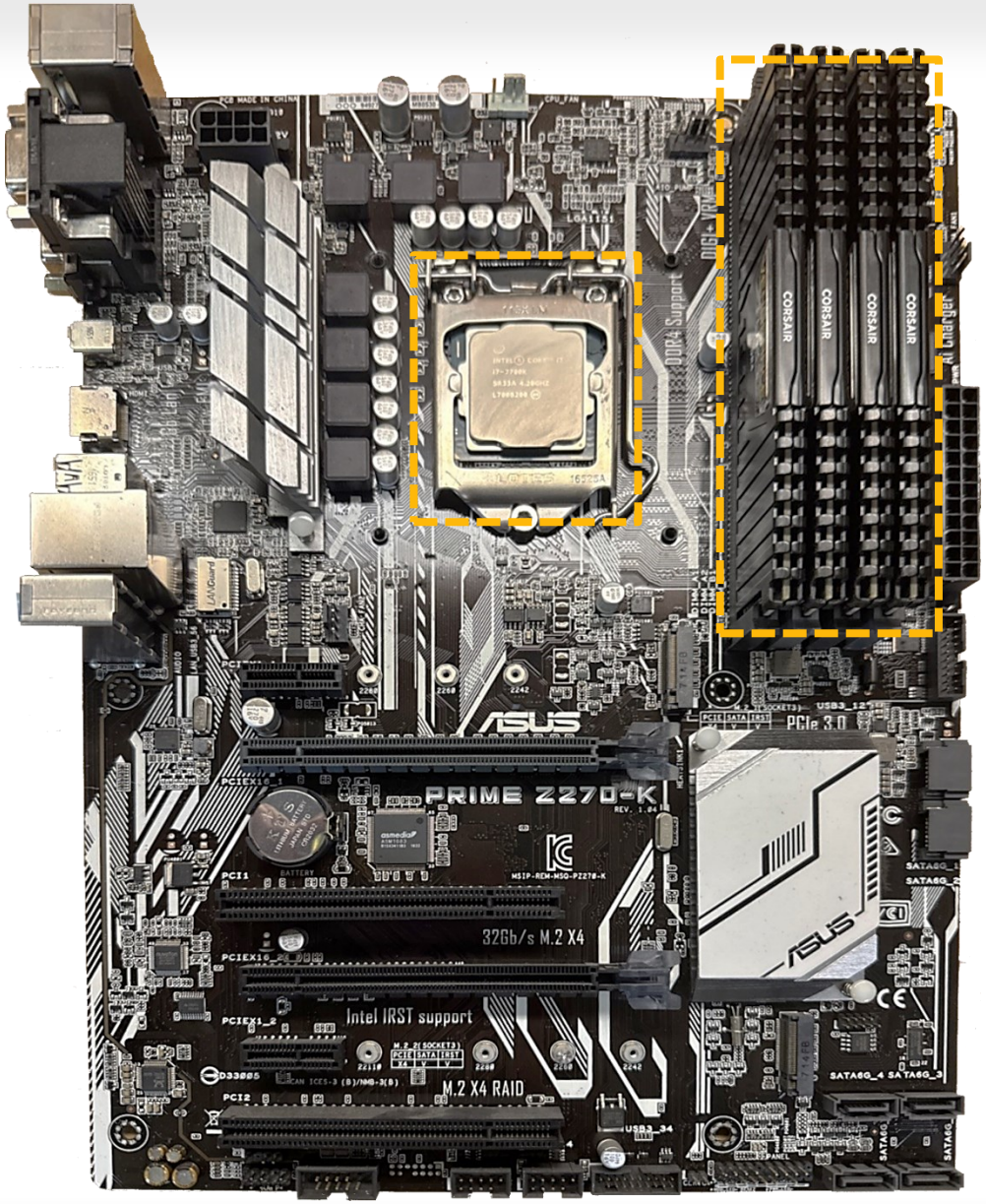


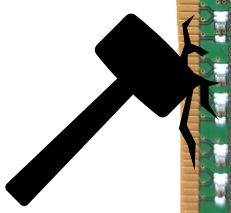
Nureddin Kamadan, **Walter Wang**, Stephan van Schaik,
Christina Garman, Daniel Genkin, Yuval Yarom

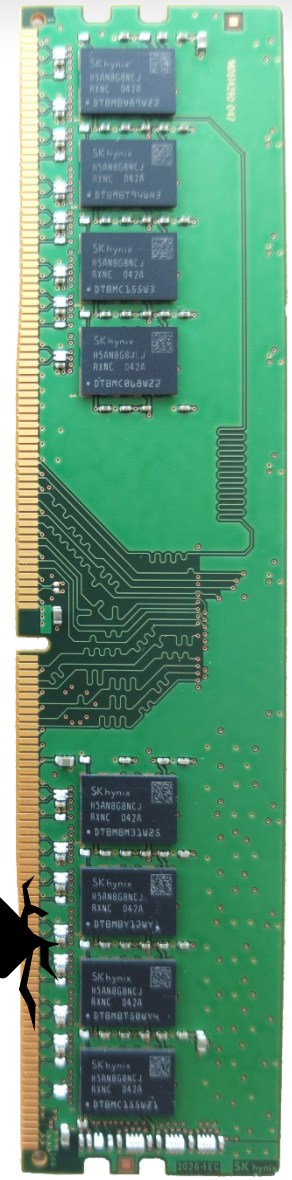
Georgia Institute of Technology

Inside Your Computer…

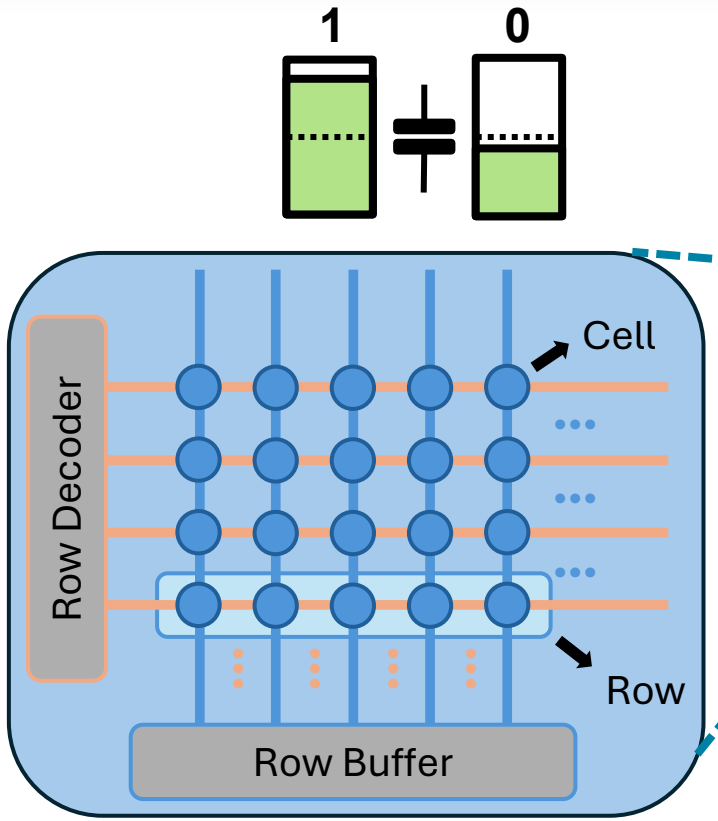
Inside Your Computer…

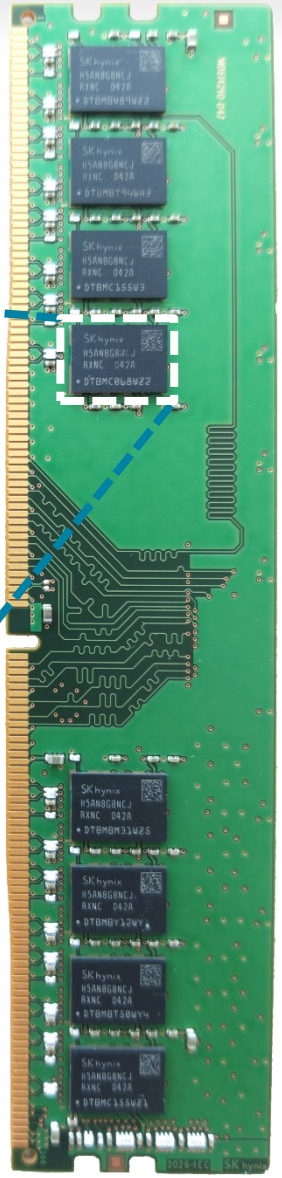






How DRAM works?

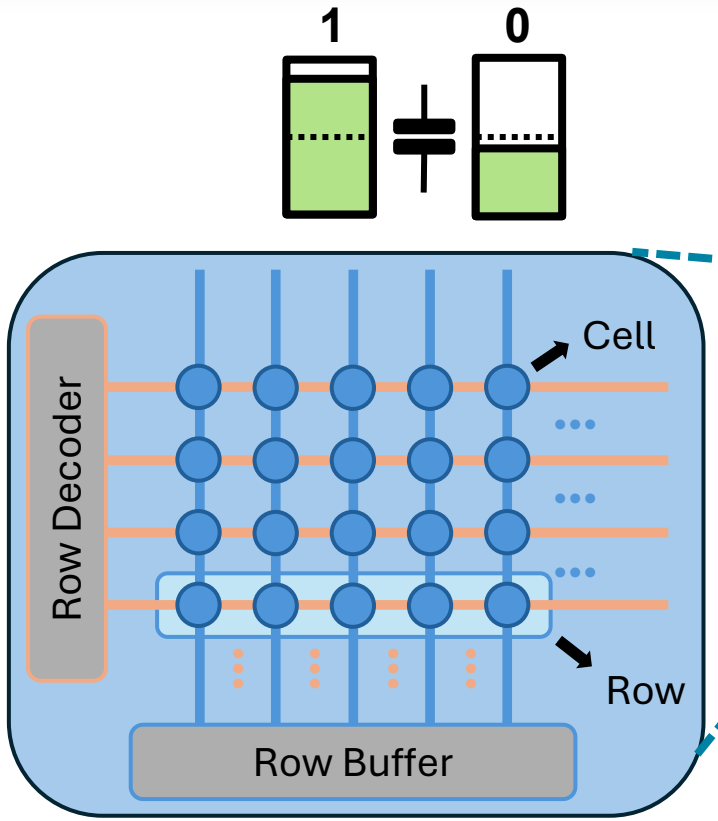


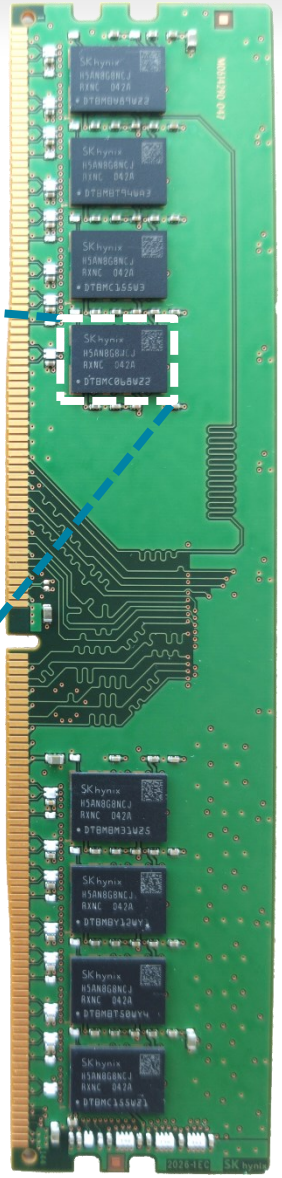


How DRAM works?

A Row activate command

Ref Refresh command

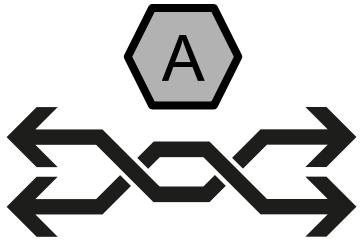


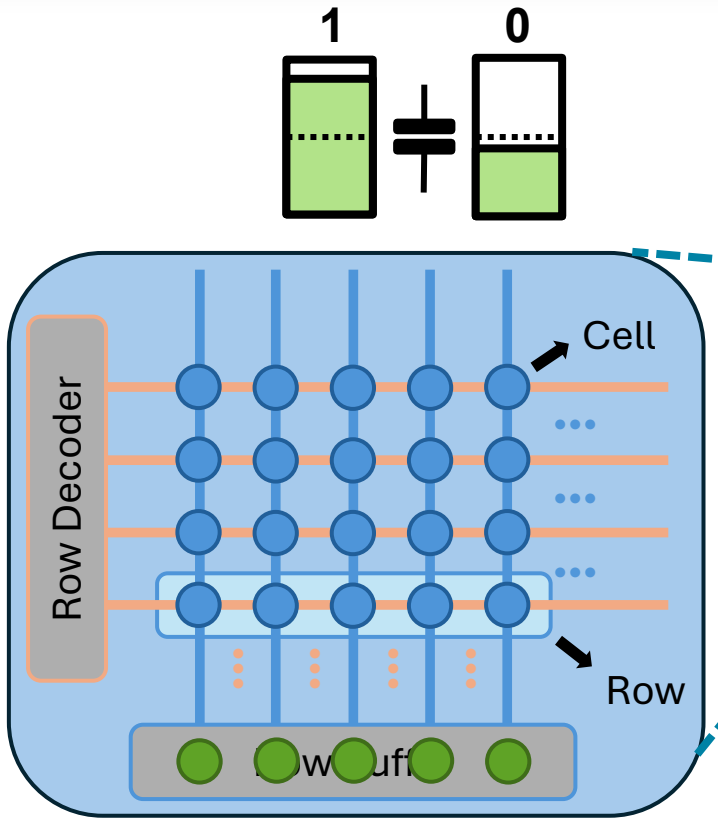


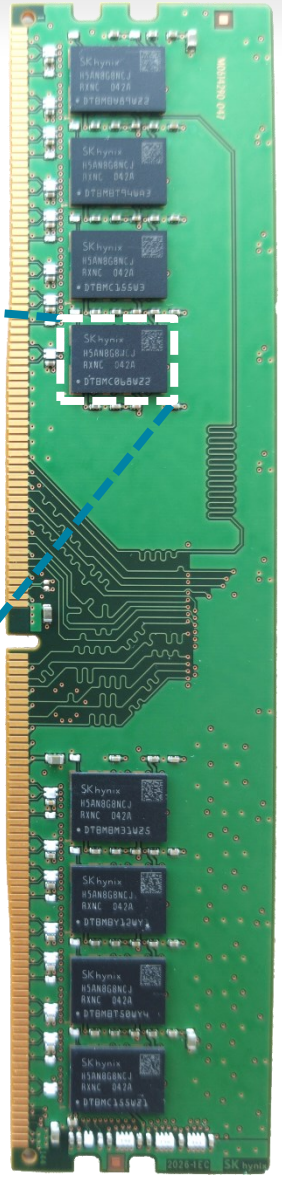
How DRAM works?

A Row activate command

Ref Refresh command



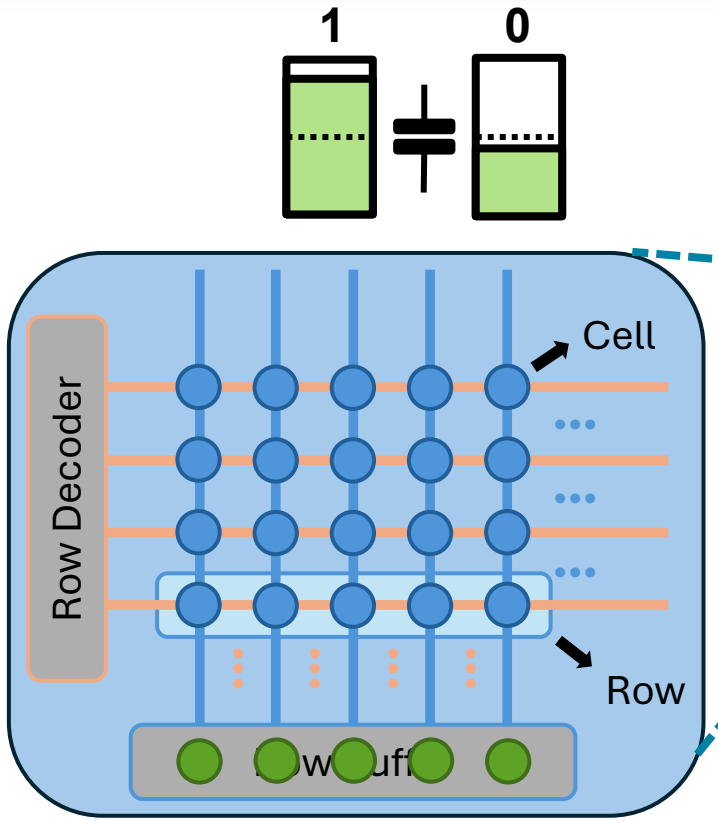


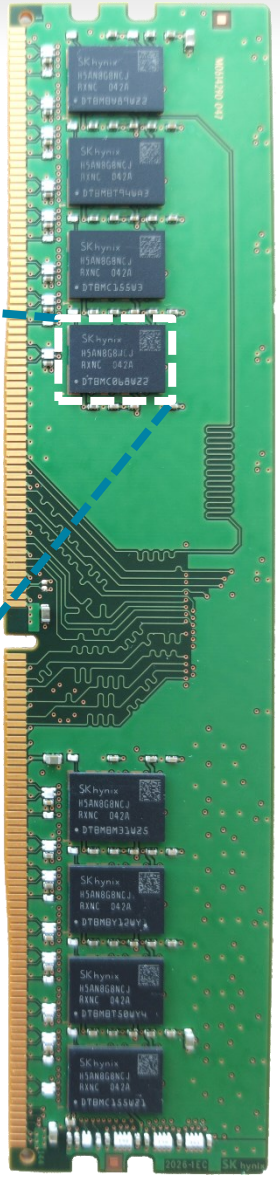


How DRAM works?

A Row activate command

Ref Refresh command

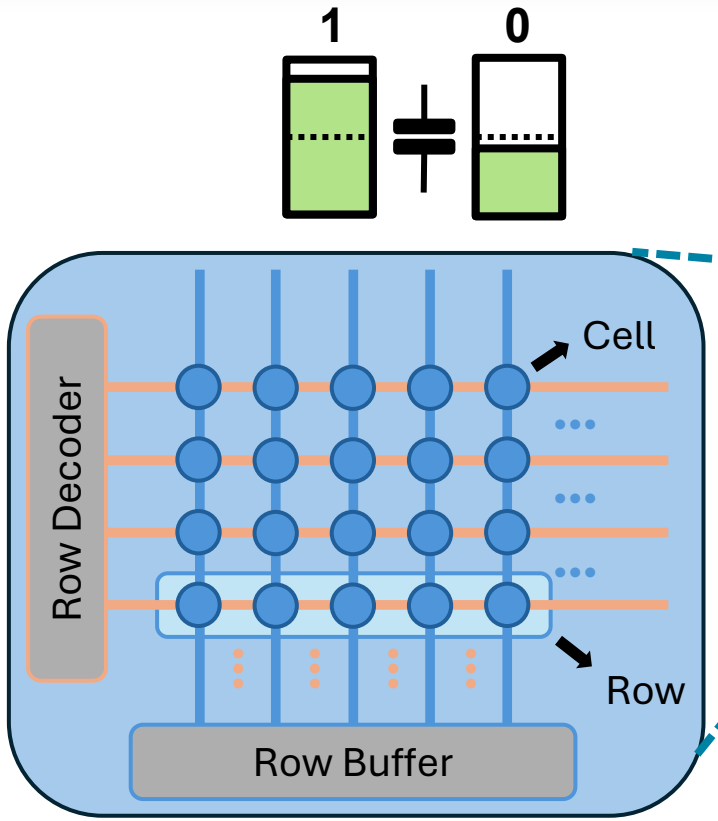


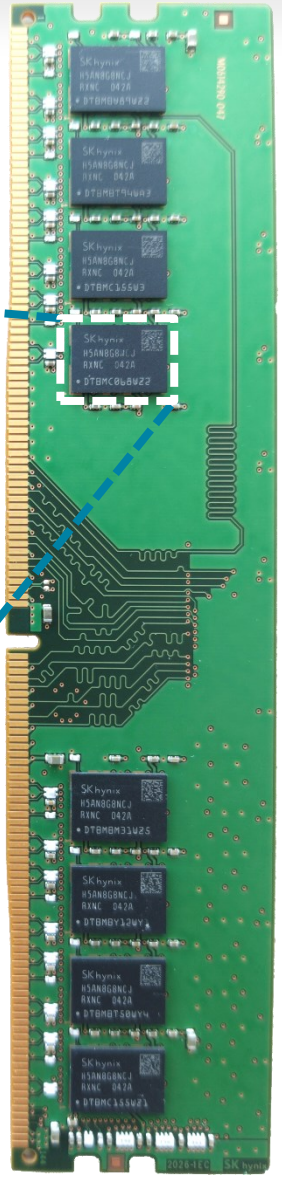


How DRAM works?

A Row activate command

Ref Refresh command

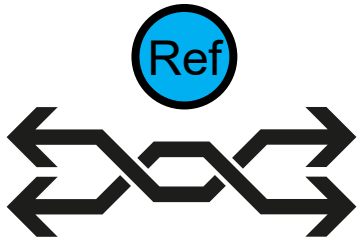


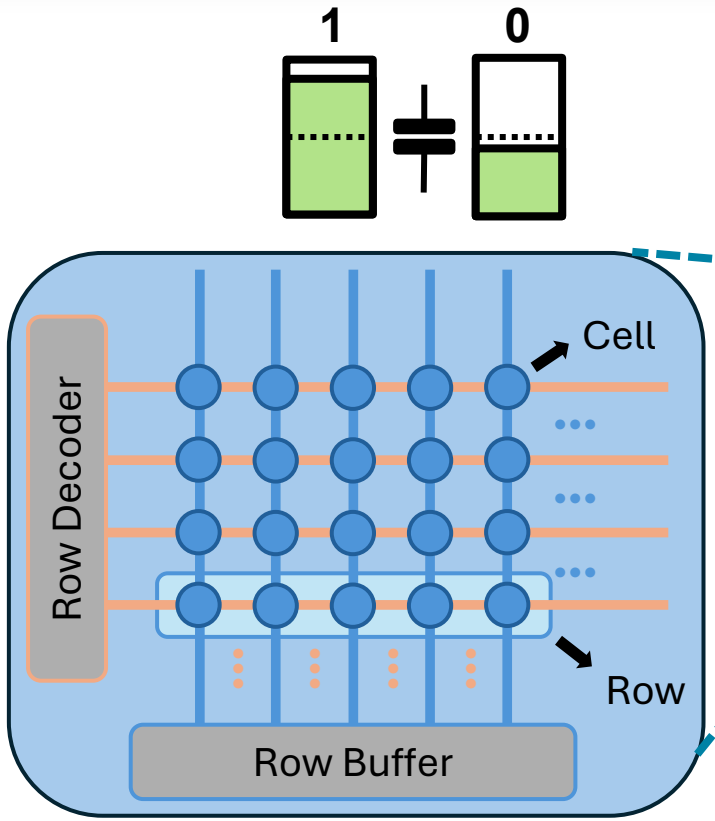


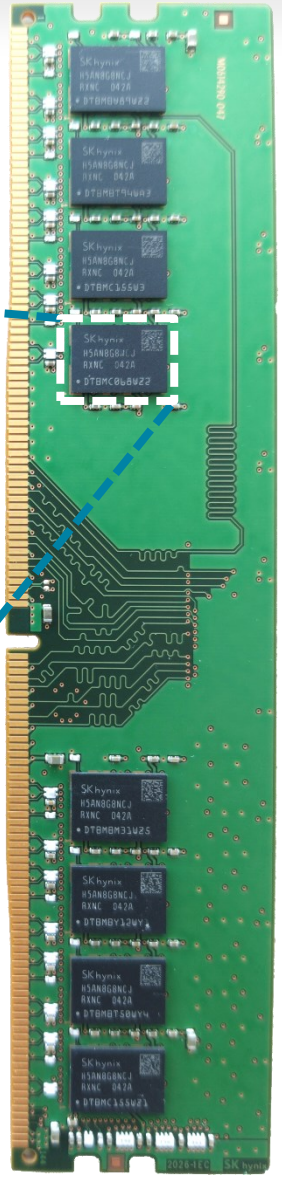
How DRAM works?

A Row activate command

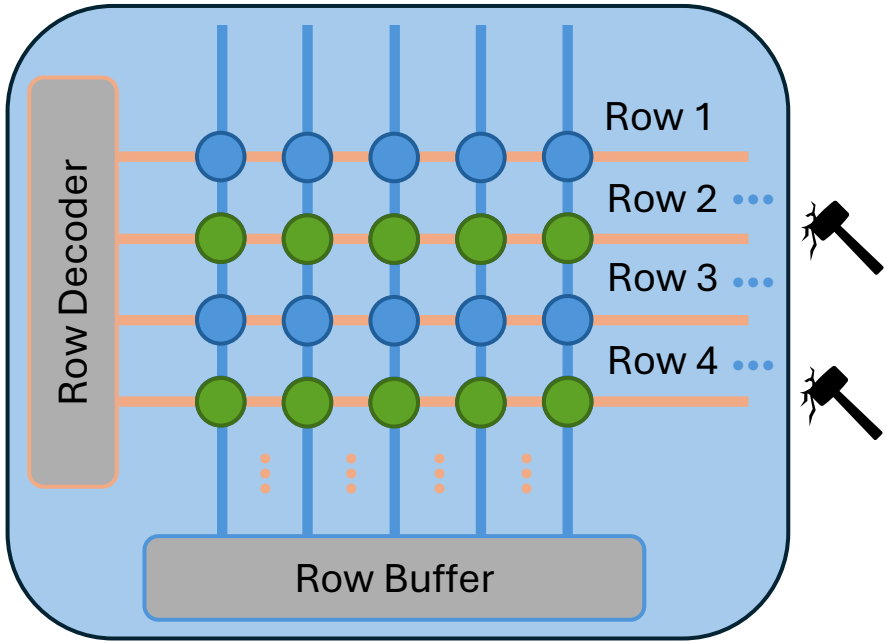
Ref Refresh command



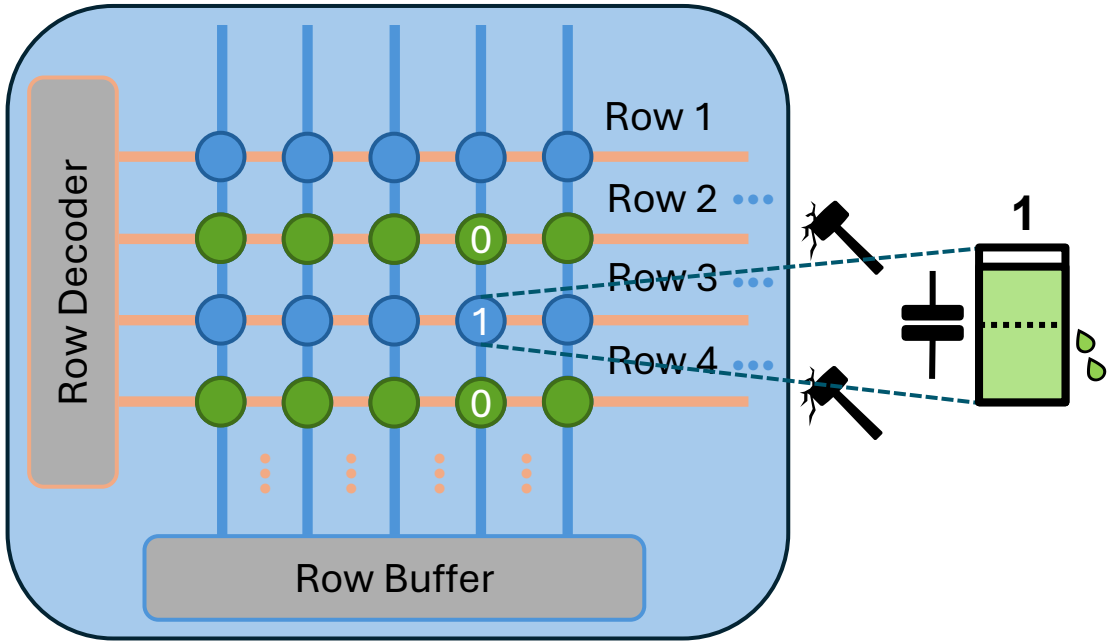




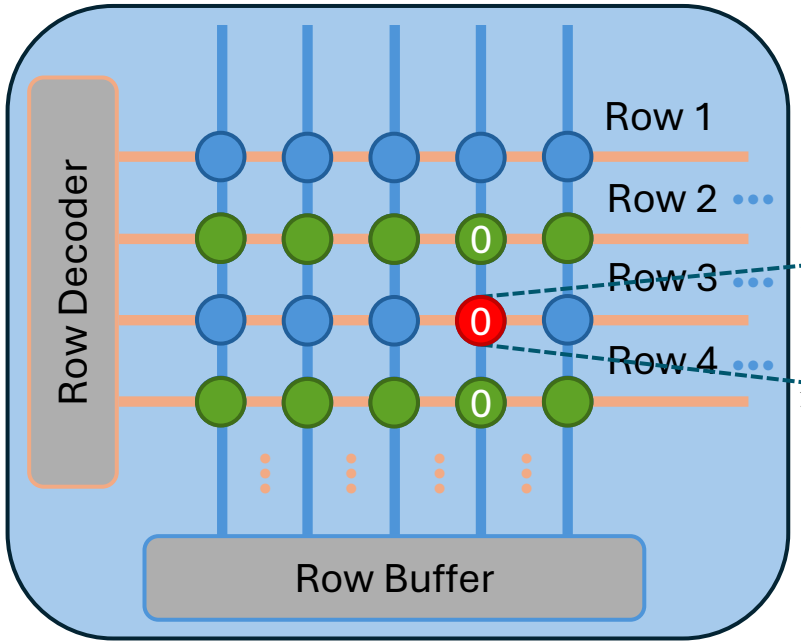
Rowhammer Attack

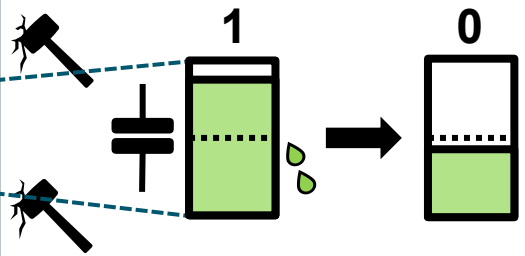


Rowhammer Attack



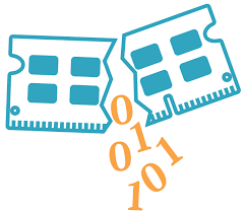
Rowhammer Attack





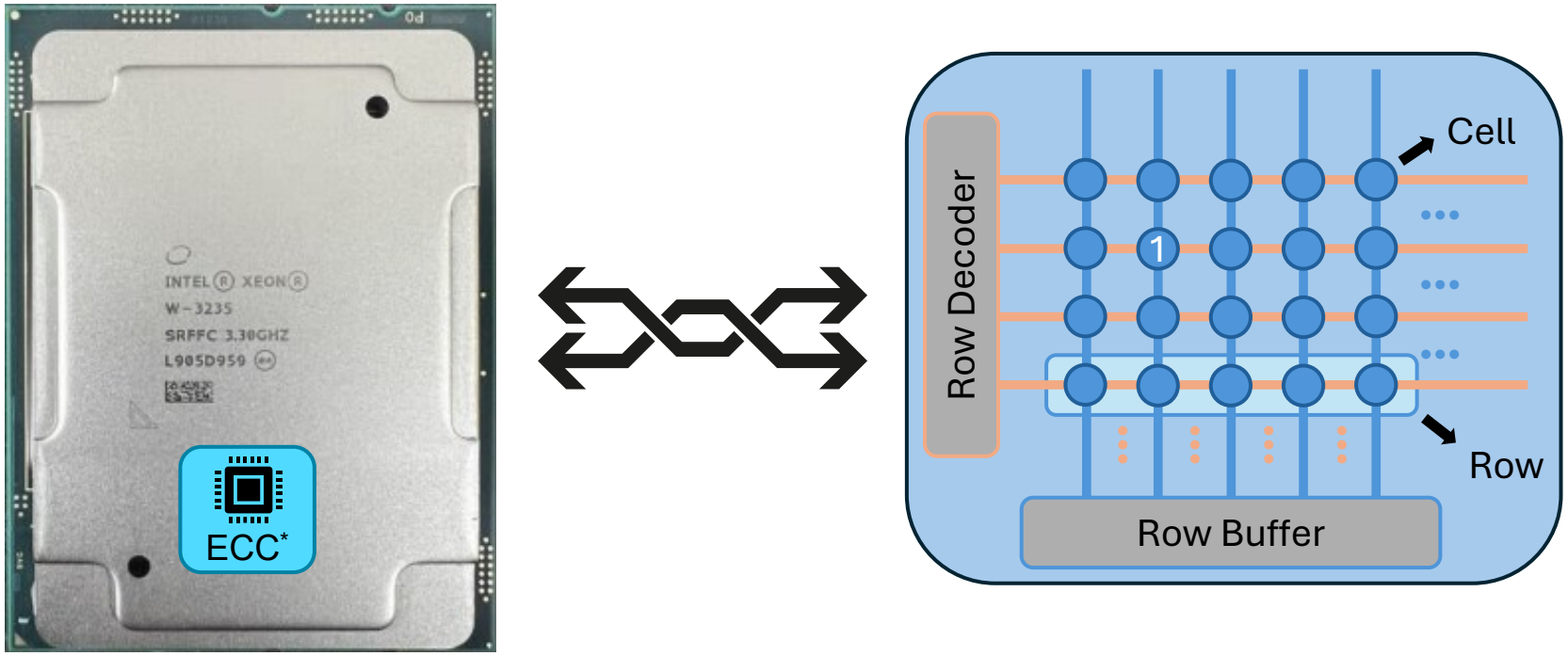
root

Rambleed



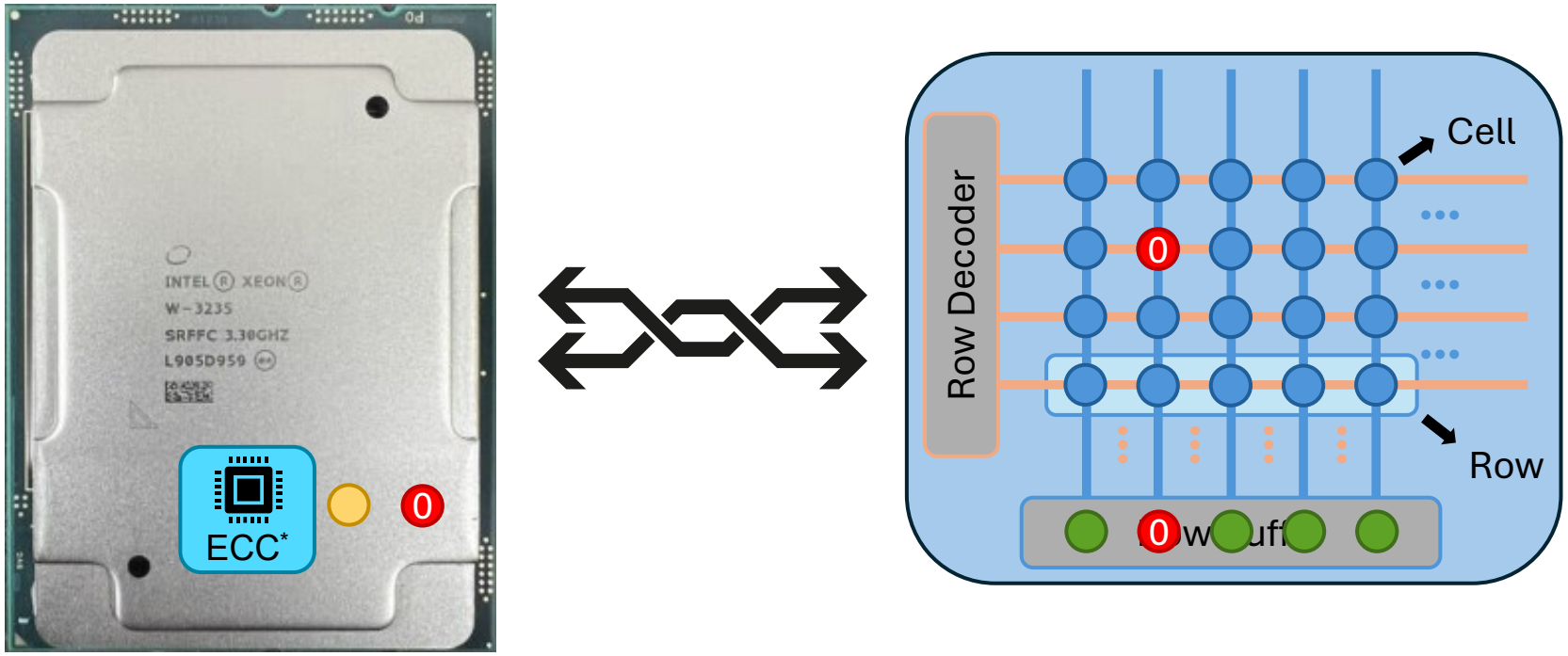
What about servers?

Rowhammer Attack on Servers?



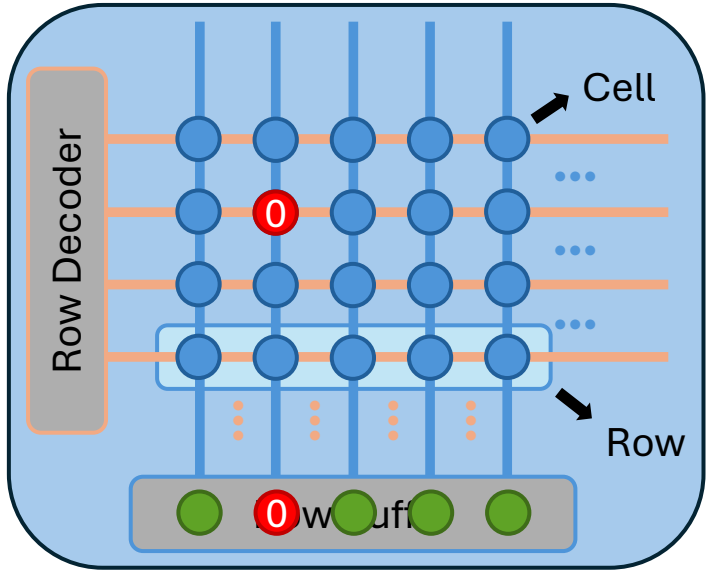
ECC*: Error Correcting Code

Rowhammer Attack on Servers?



ECC*: Error Correcting Code

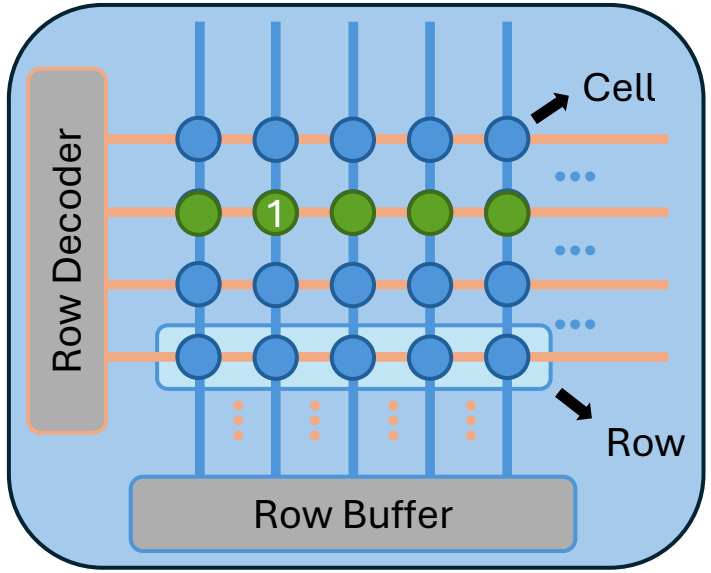
Rowhammer Attack on Servers?



Correctable

ECC*: Error Correcting Code

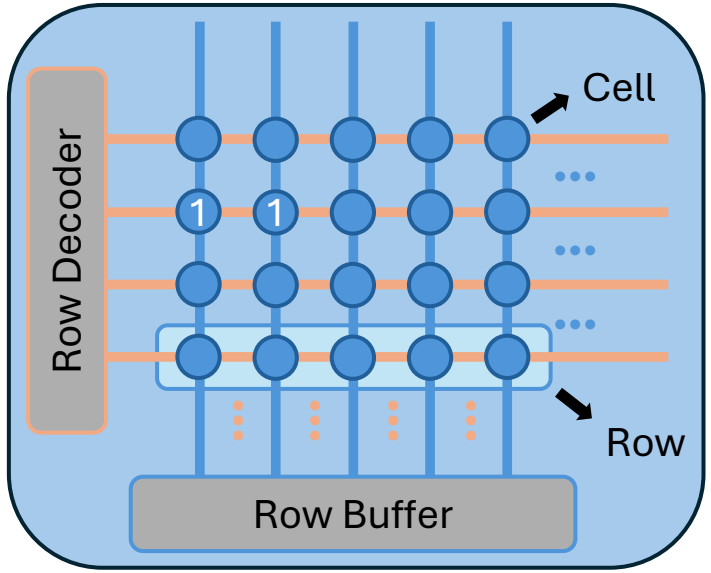
Rowhammer Attack on Servers?



Correctable

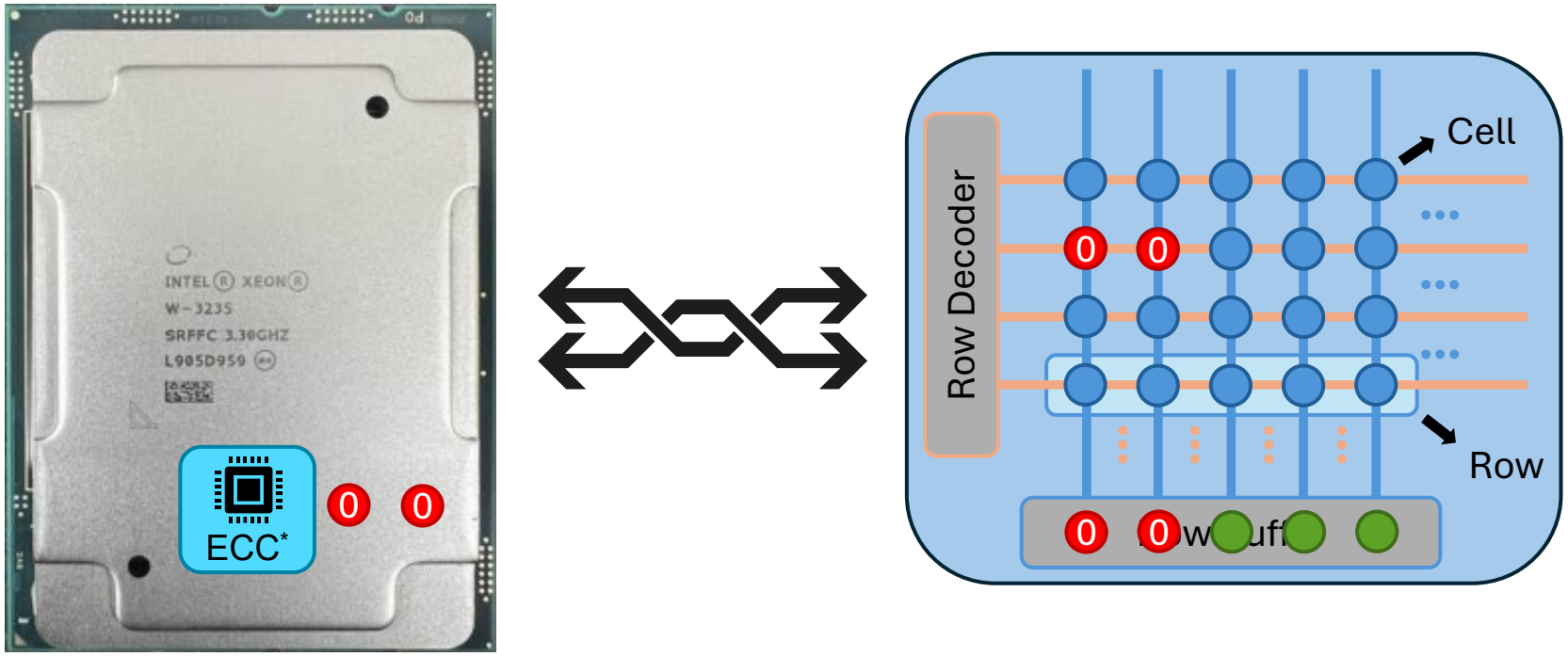
ECC*: Error Correcting Code

Rowhammer Attack on Servers?



ECC*: Error Correcting Code

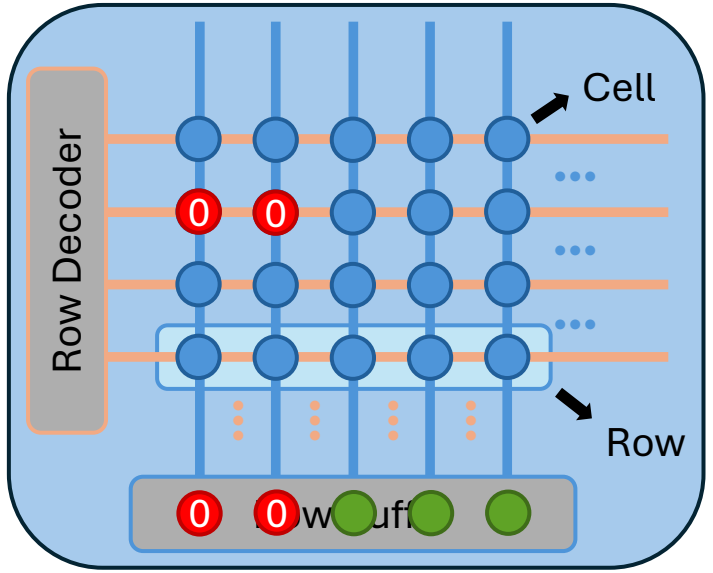
Rowhammer Attack on Servers?



ECC*: Error Correcting Code

Rowhammer Attack on Servers?

Crashed



ECC*: Error Correcting Code

Rowhammer Attack on Servers?

ECC*: Error Correcting Code

Our Results

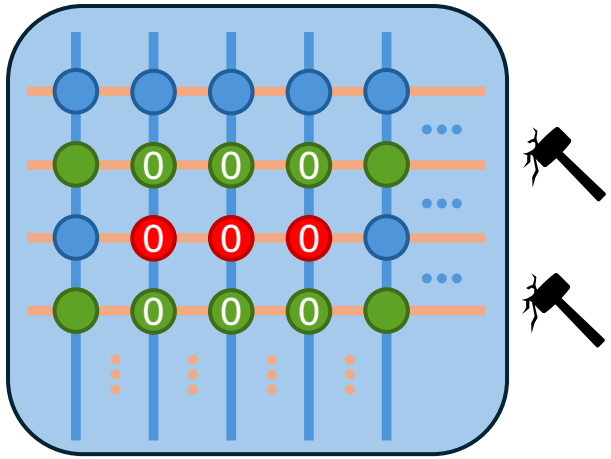
First Rowhammer Attack on DDR4 Servers with Default Configuration!

- **Get Rowhammer bit flips: avg. 2.5h**
- **Break RSA signatures: avg. 10h**

ECC*: Error Correcting Code

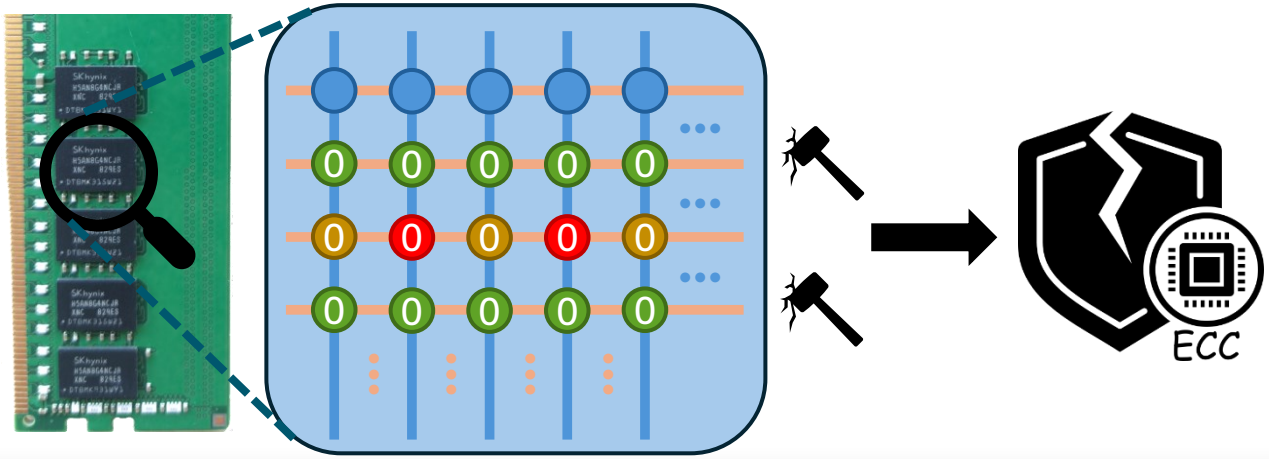
Preview: 4 Challenges

C1: Rowhammer on Server DIMMs



C2: Recover the ECC Matrix

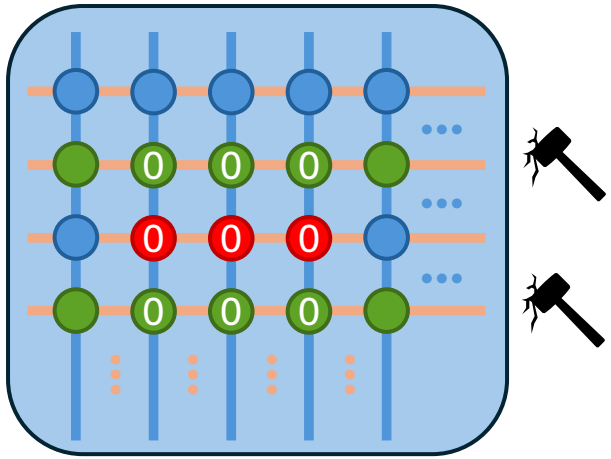
C3: Template DIMMs



C4: End-to-End Attack

Index: 4 Challenges

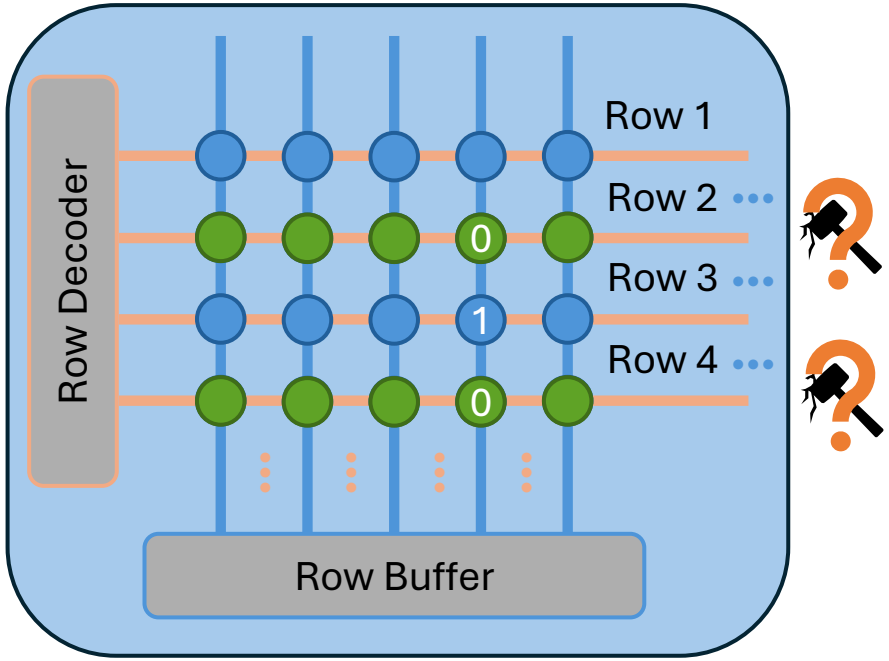
C1: Rowhammer on Server DIMMs

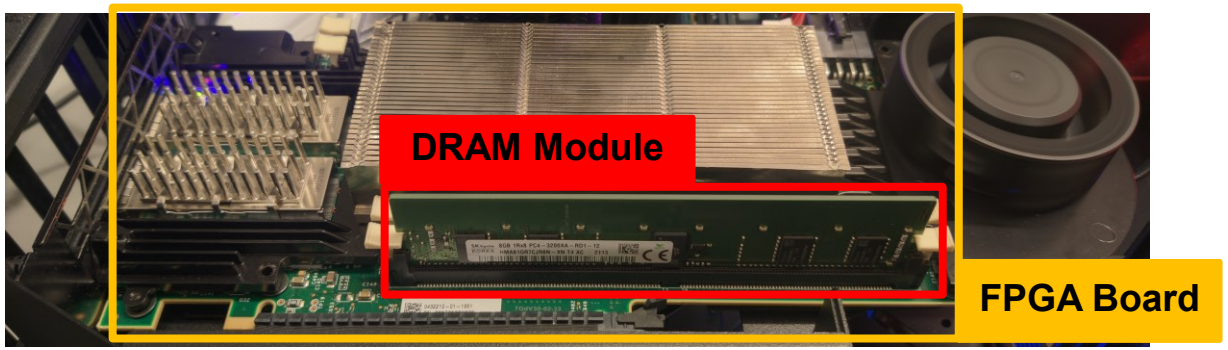


C1: Rowhammer on Server DIMMs

Challenge 1:

No Rowhammer research on DDR4 server DIMMs



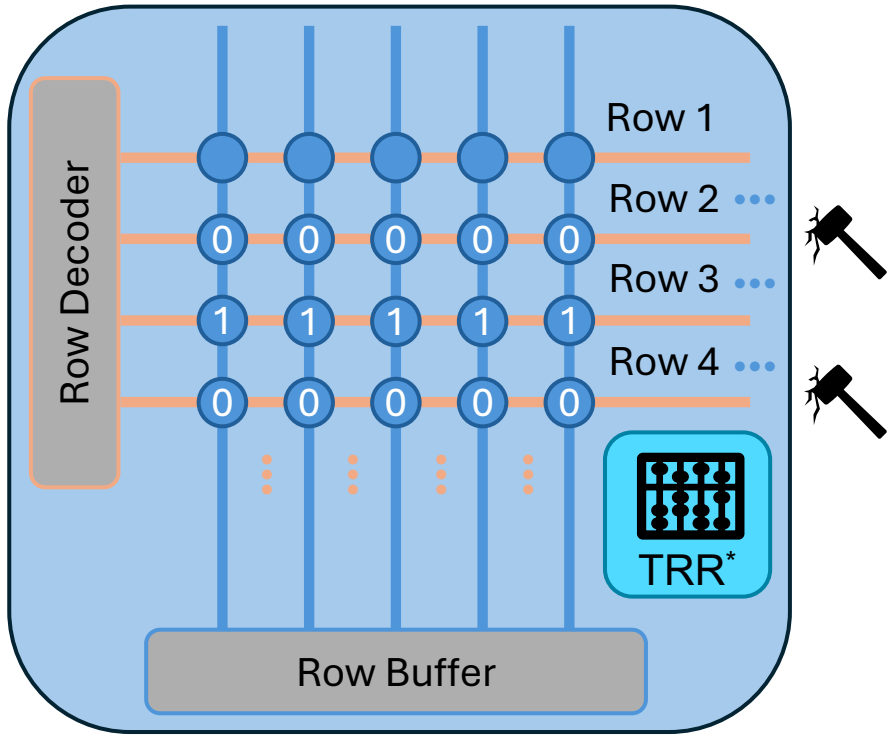


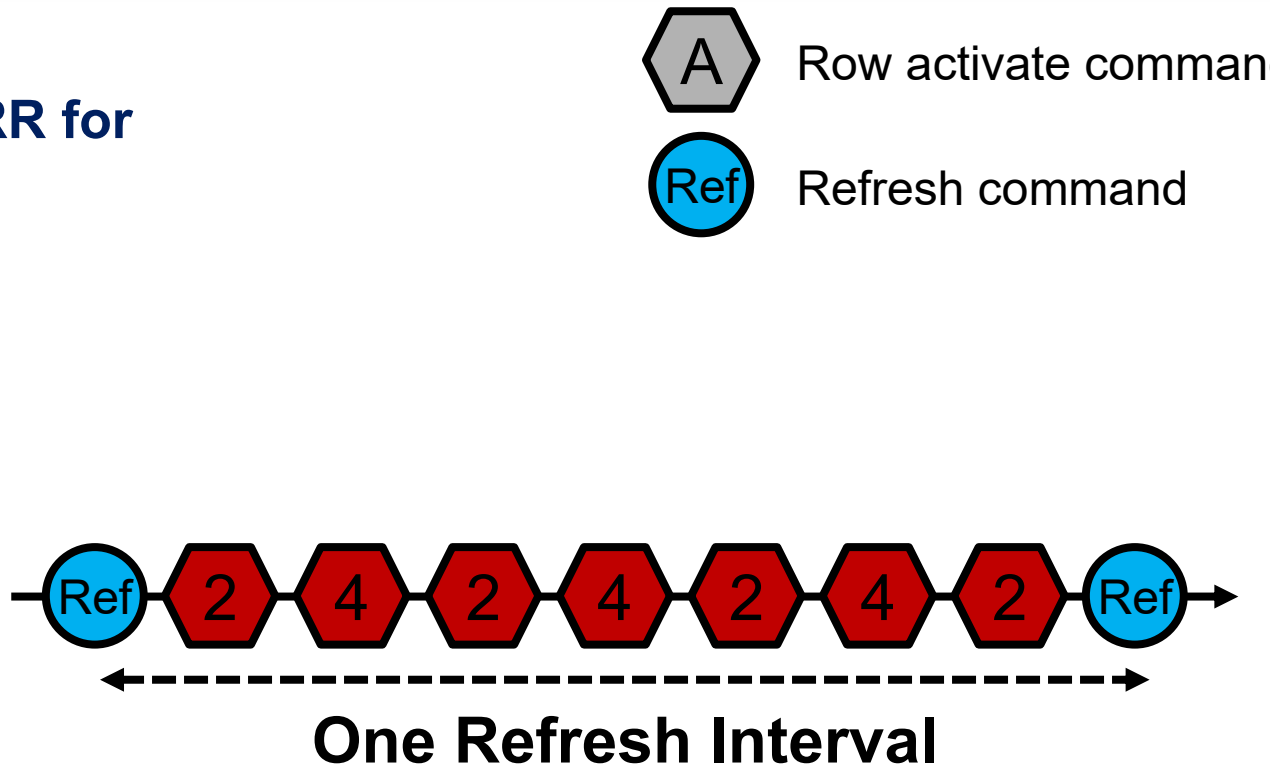
Solution: Analyze Hynix server DIMMs on FPGA to find effective hammer pattern

S1: Server DIMM Analysis on FPGA

Observation 1:

Hynix DIMMs use sampling-based TRR for Rowhammer mitigation



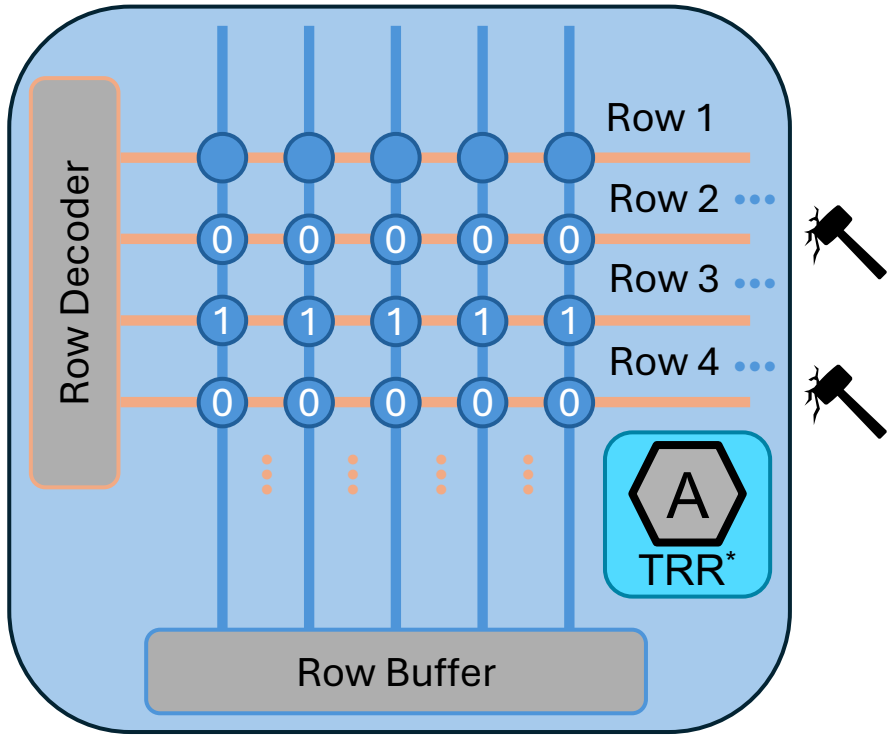


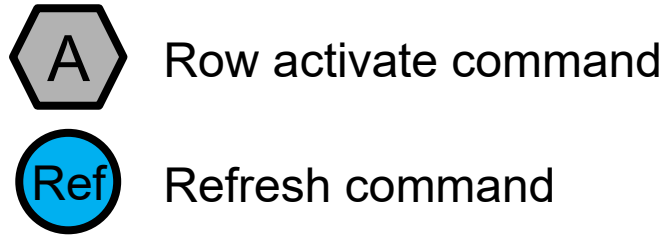
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 1:

Hynix DIMMs use sampling-based TRR for Rowhammer mitigation



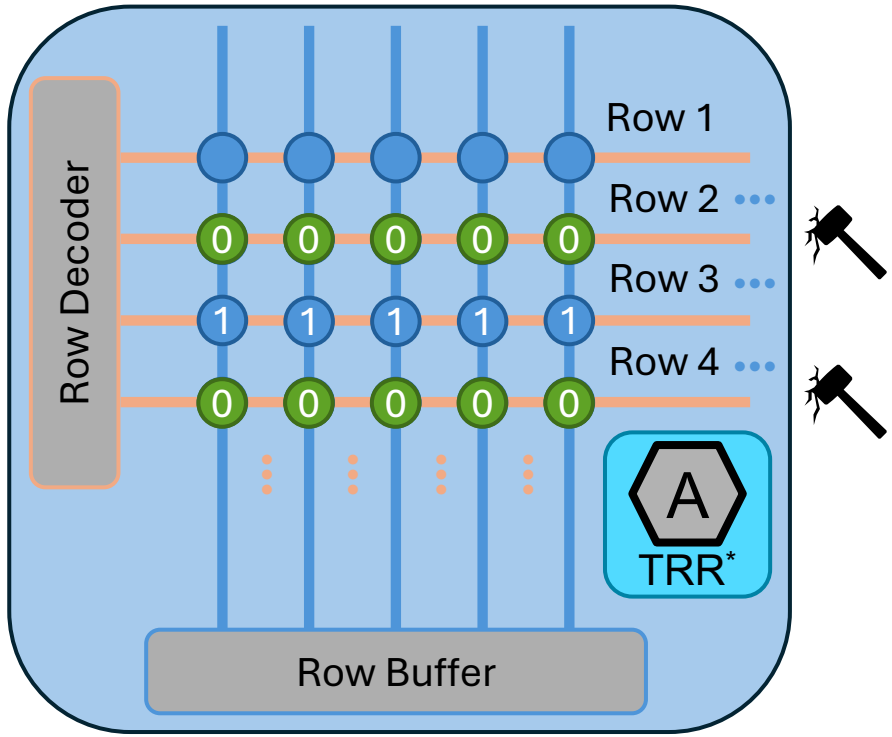


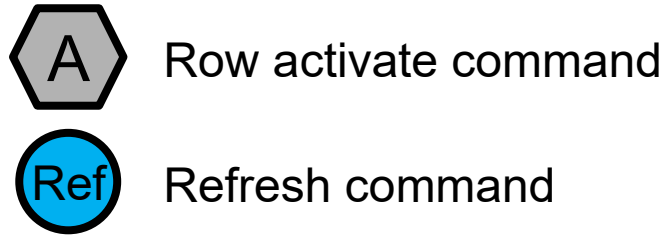
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 1:

Hynix DIMMs use sampling-based TRR for Rowhammer mitigation



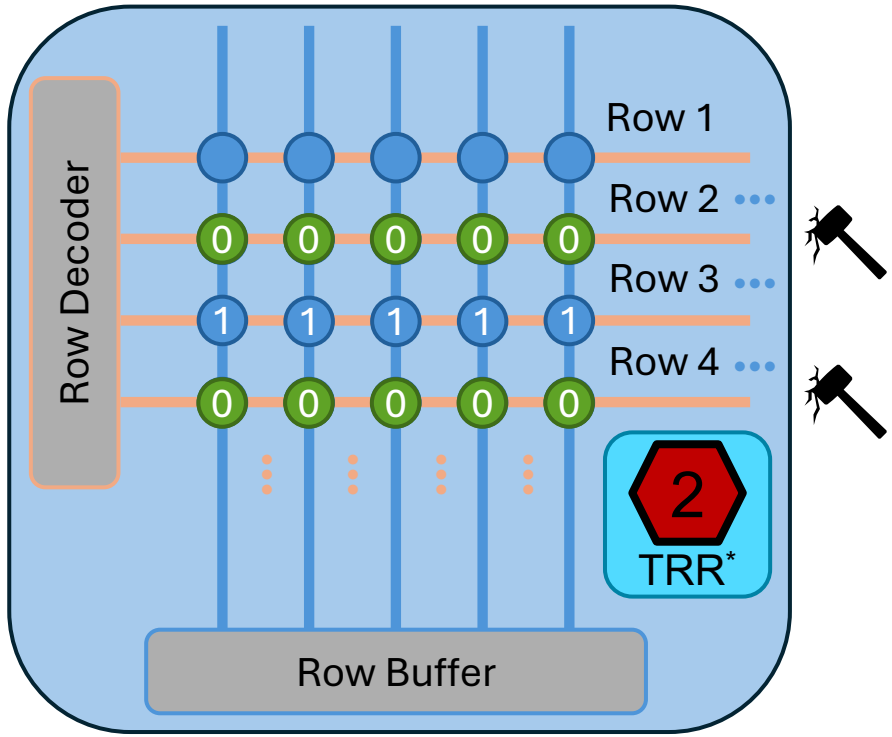


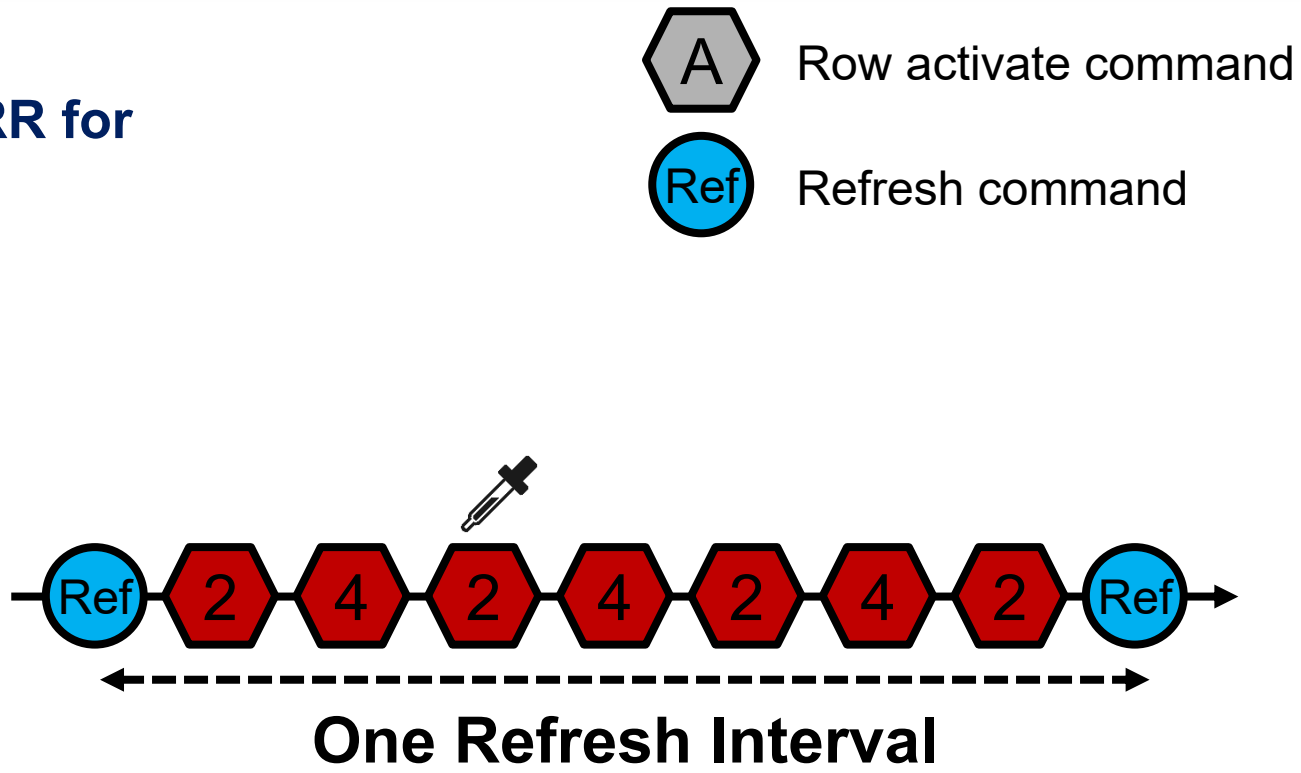
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 1:

Hynix DIMMs use sampling-based TRR for Rowhammer mitigation



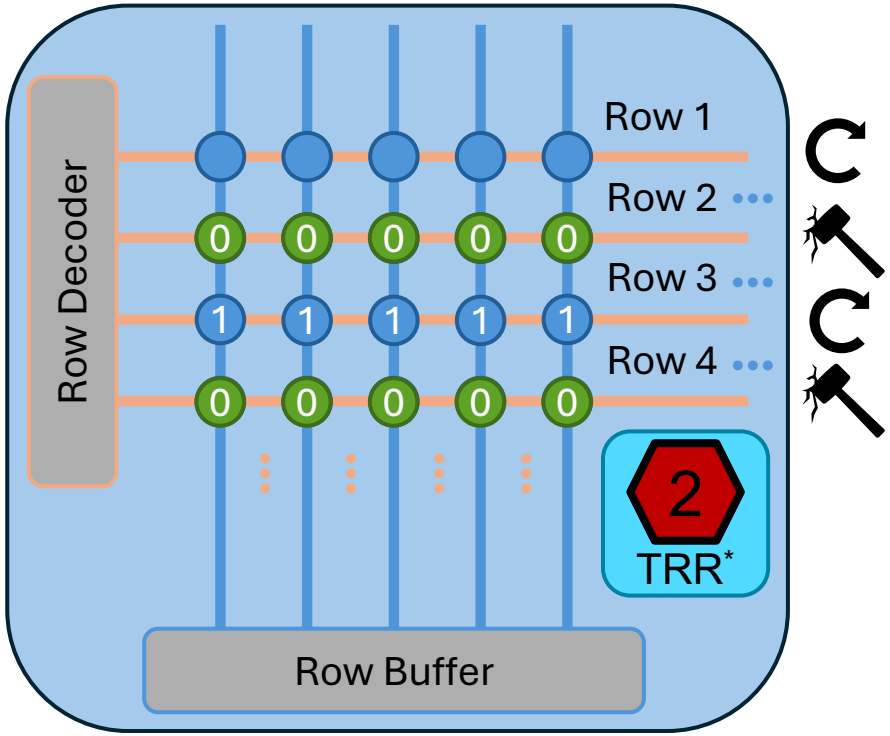


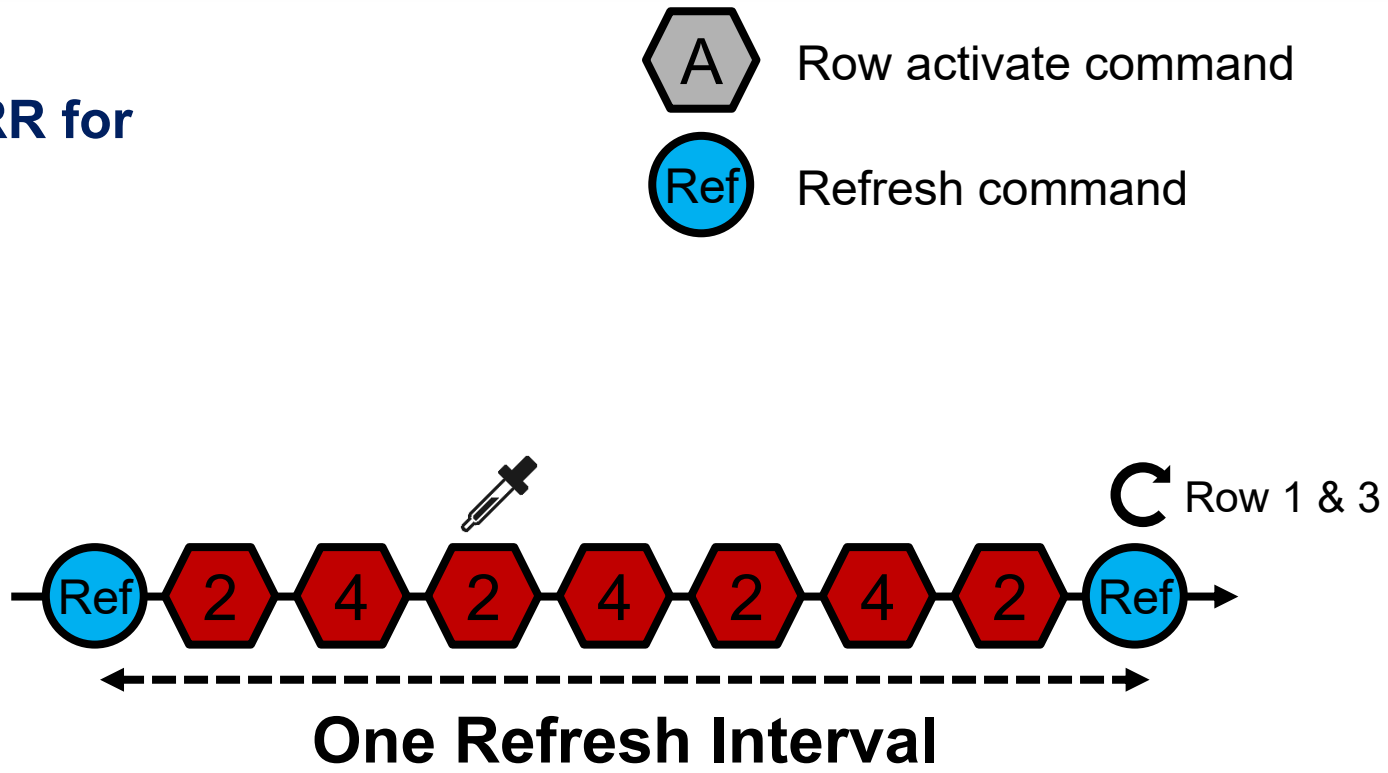
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 1:

Hynix DIMMs use sampling-based TRR for Rowhammer mitigation



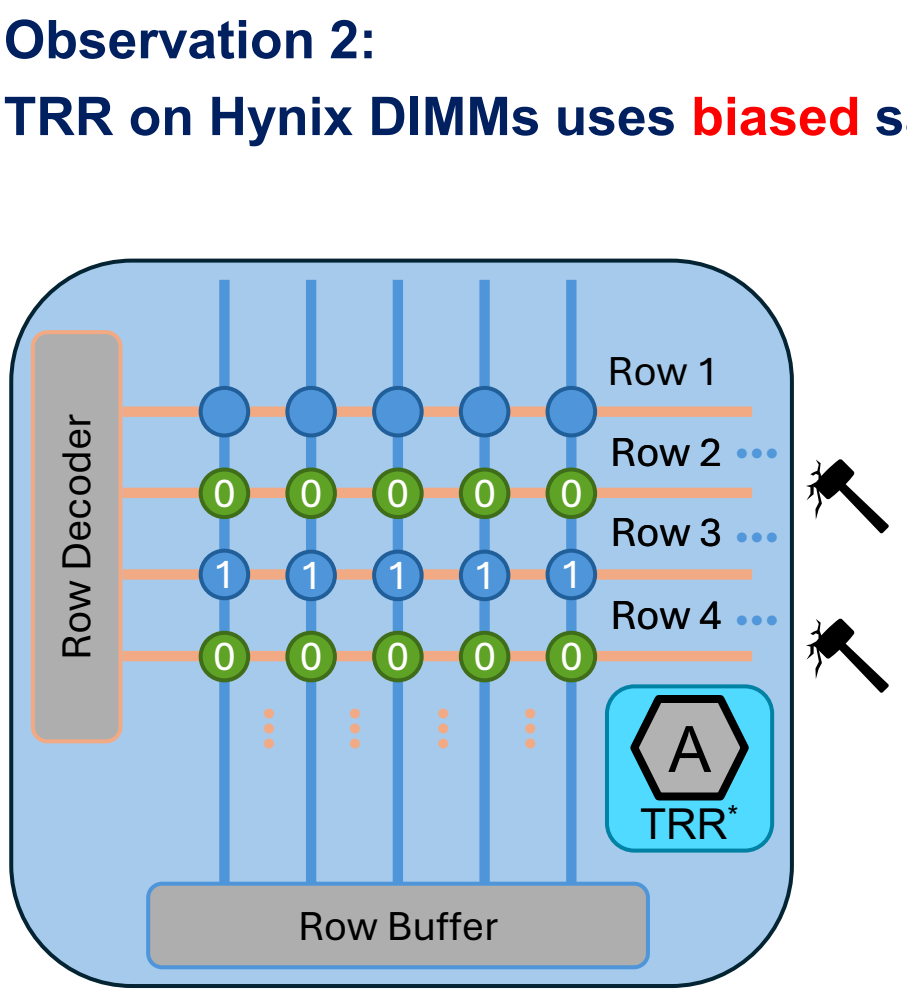


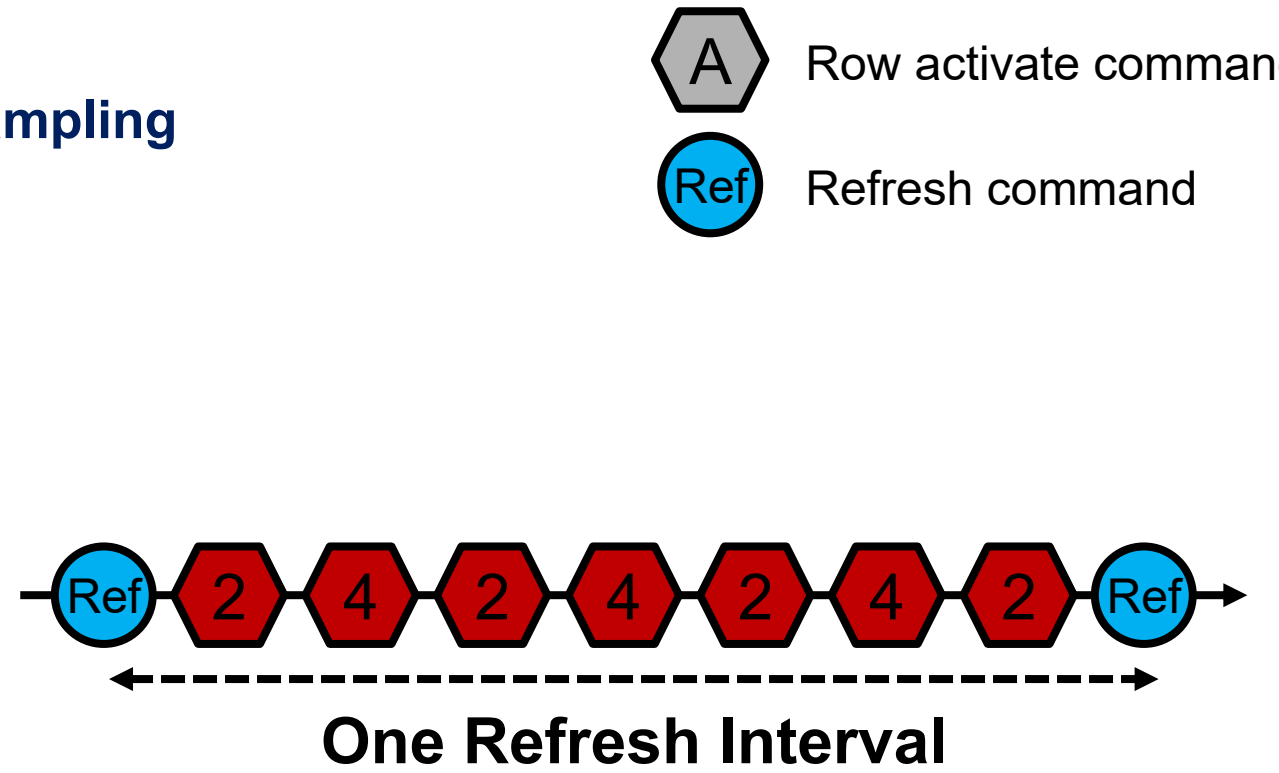
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 2:

TRR on Hynix DIMMs uses biased sampling



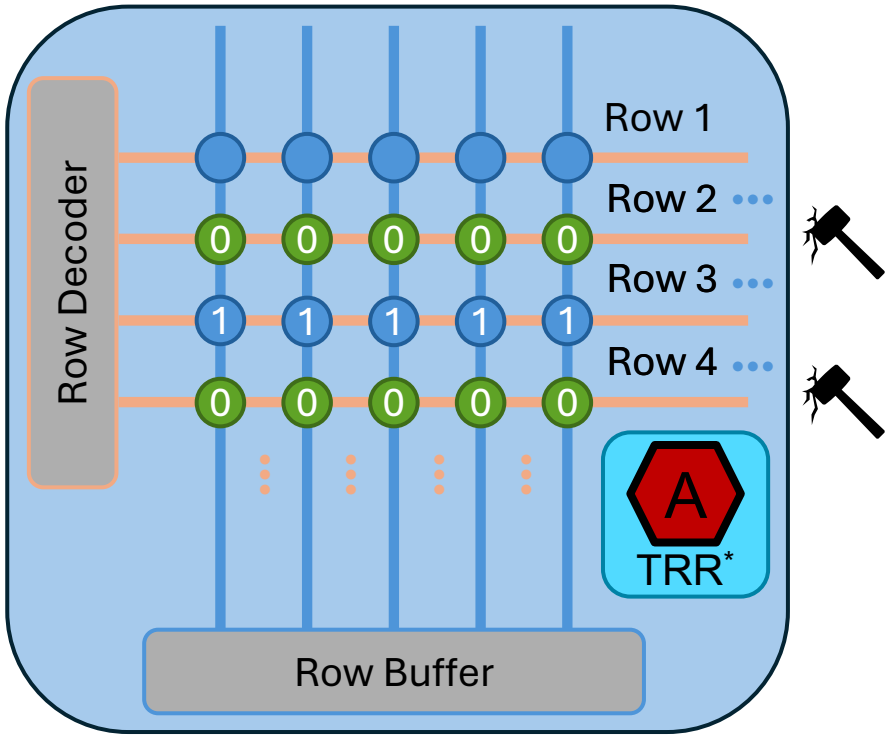


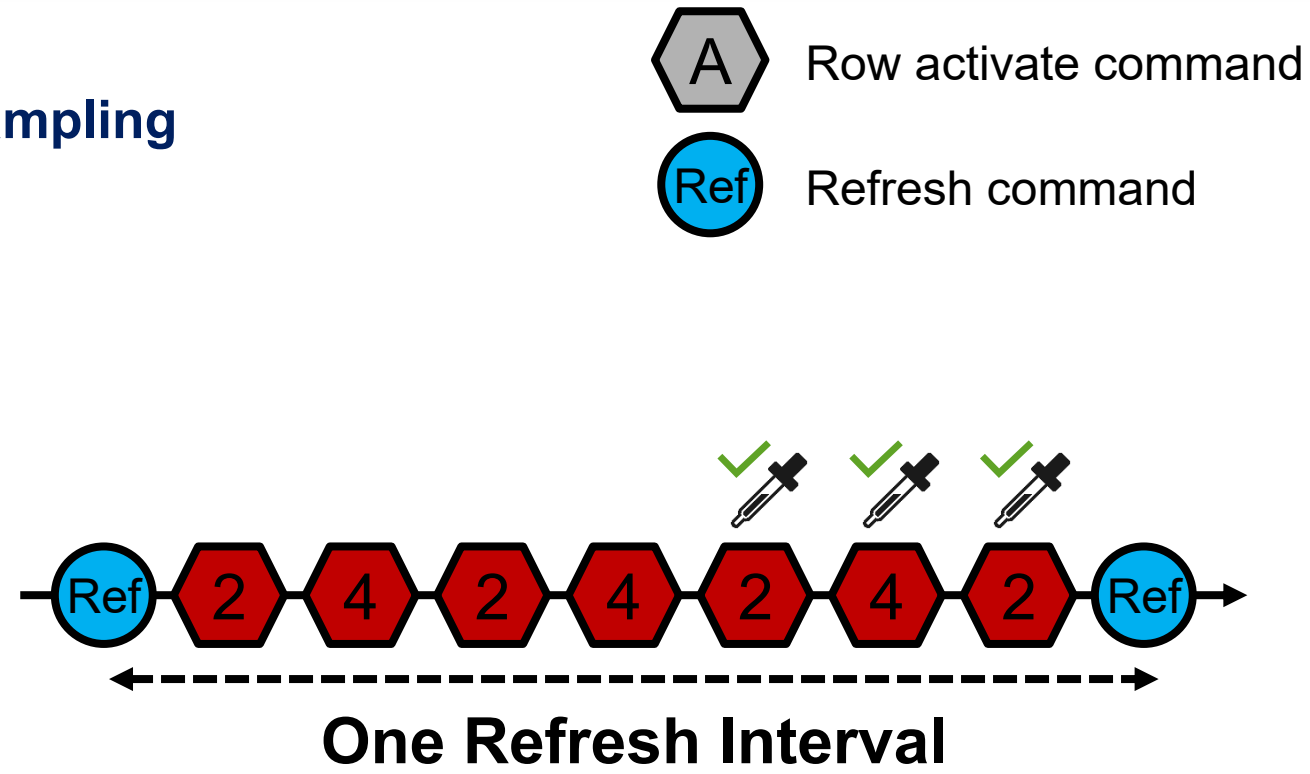
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 2:

TRR on Hynix DIMMs uses biased sampling



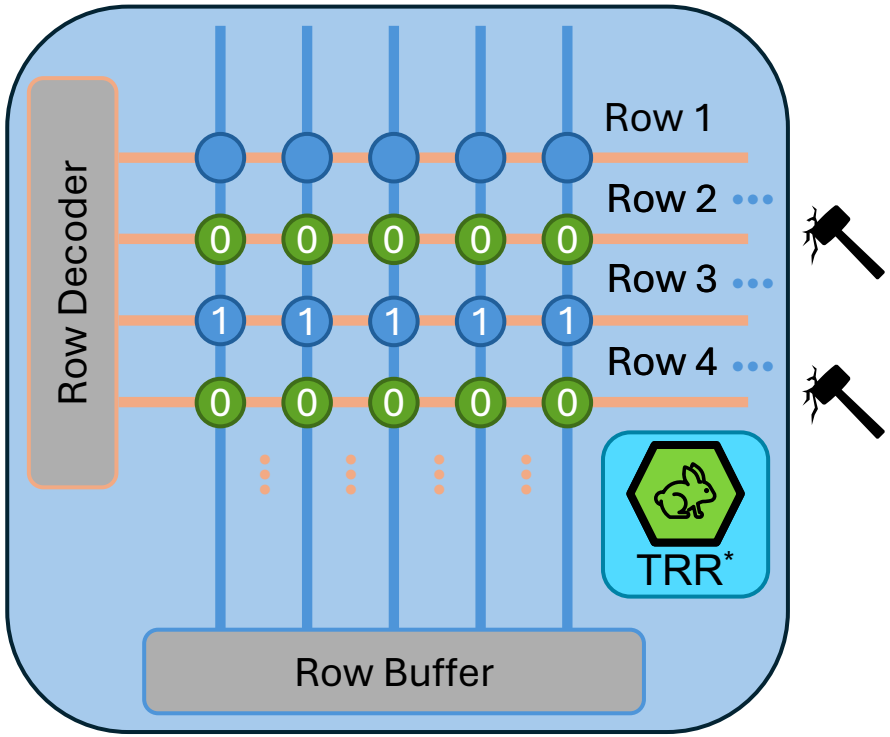


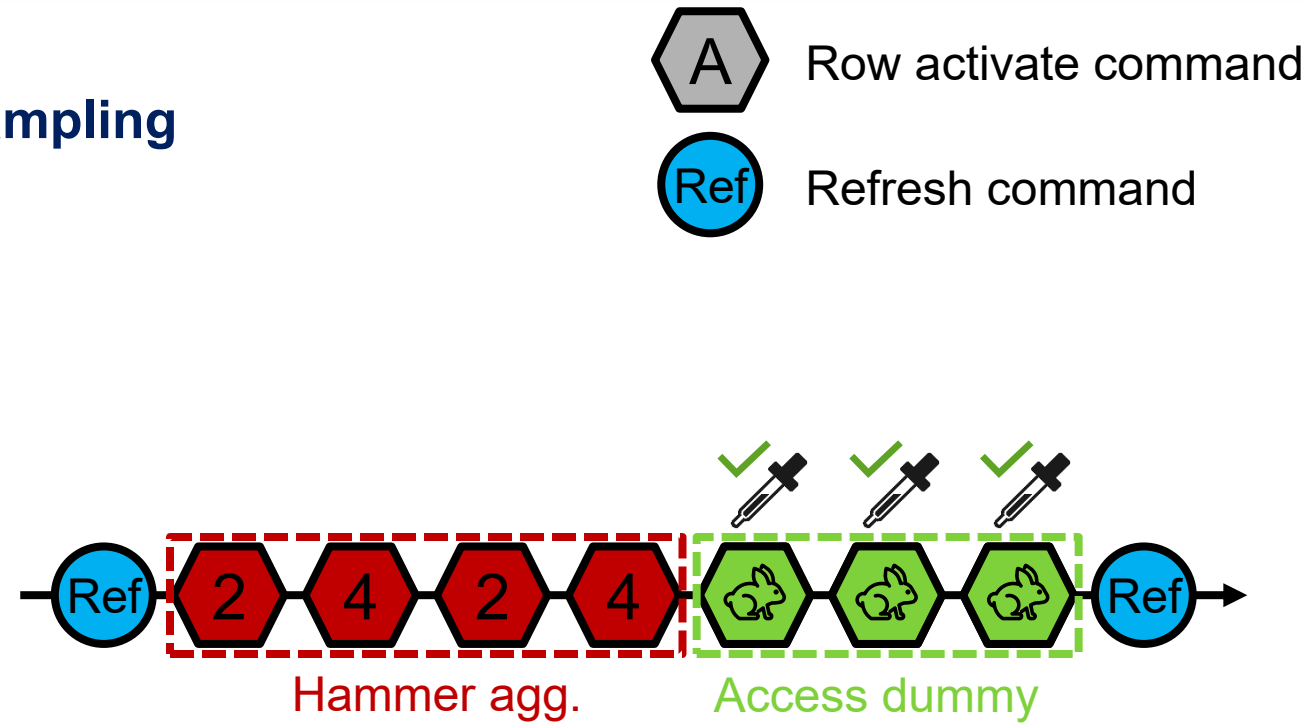
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 2:

TRR on Hynix DIMMs uses biased sampling



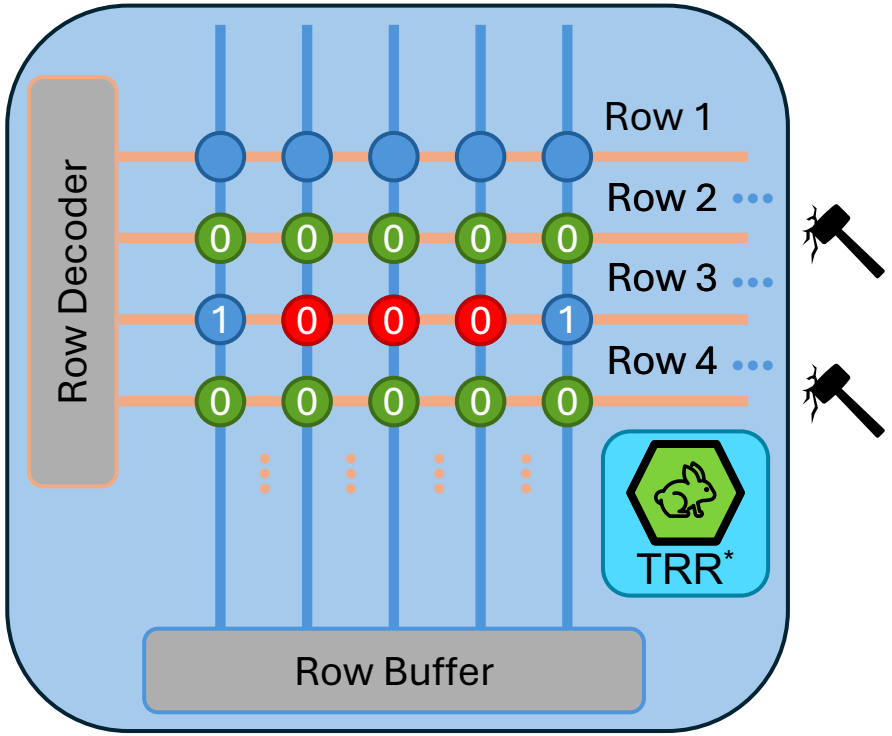


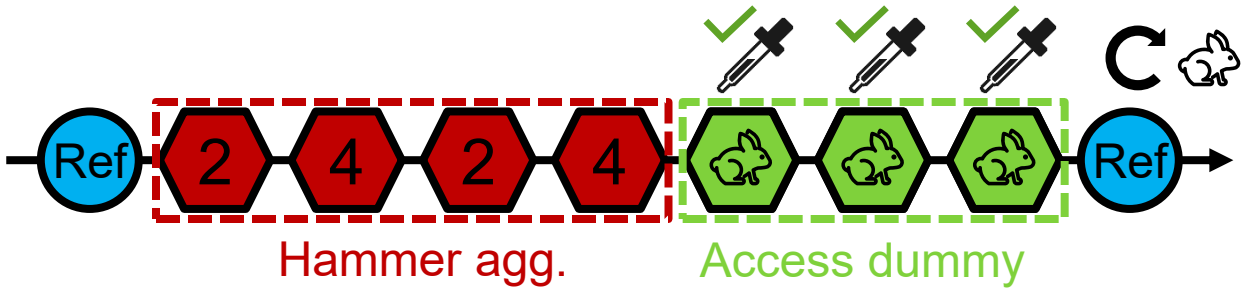
TRR*: Target Row Refresh

S1: Server DIMM Analysis on FPGA

Observation 2:

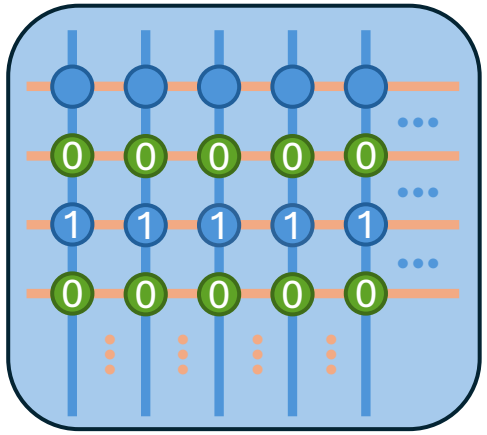
TRR on Hynix DIMMs uses biased sampling



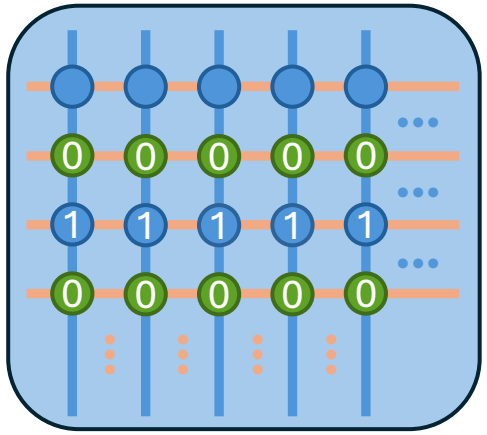


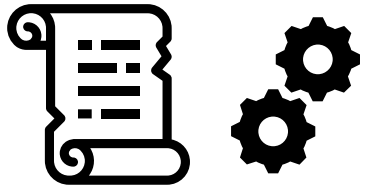
TRR*: Target Row Refresh

C1: Rowhammer on Server DIMMs



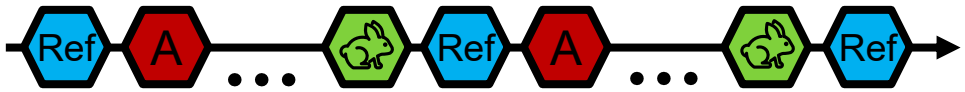
C1: Rowhammer on Server DIMMs





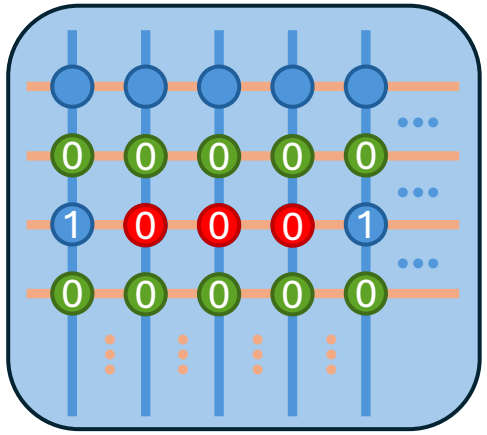
Native Hammer Code

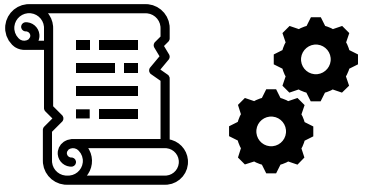
ECC.fail



Refresh Synchronization

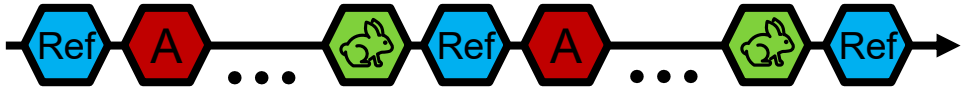
C1: Rowhammer on Server DIMMs





Native Hammer Code

ECC.fail

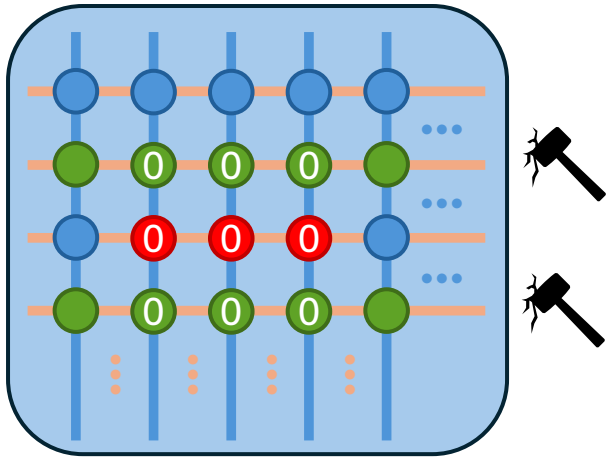


Refresh Synchronization

Result: 100+ Flips per Row on Real Machine

Index: 4 Challenges

C1: Rowhammer on Server DIMMs

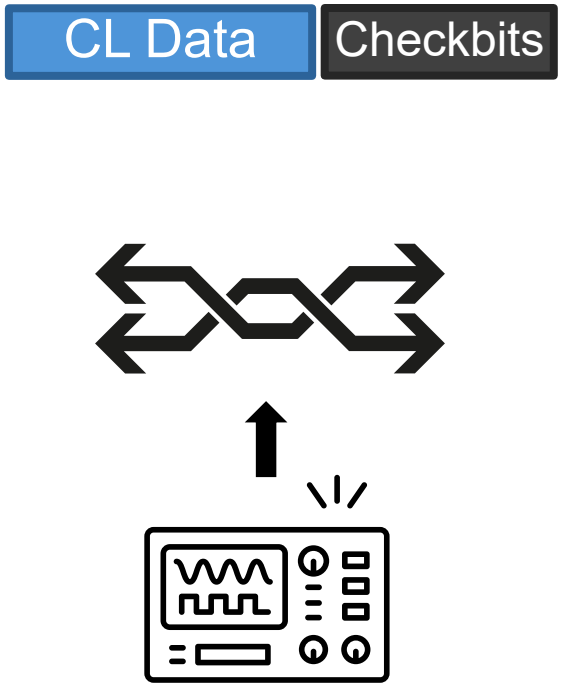


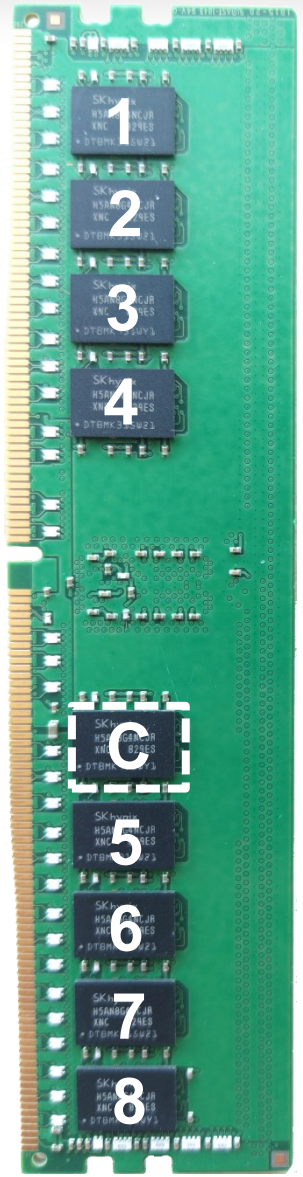
C2: Recover the ECC Matrix

C2: Recover the ECC Matrix

Challenge 2:

ECC checkbits are invisible to software





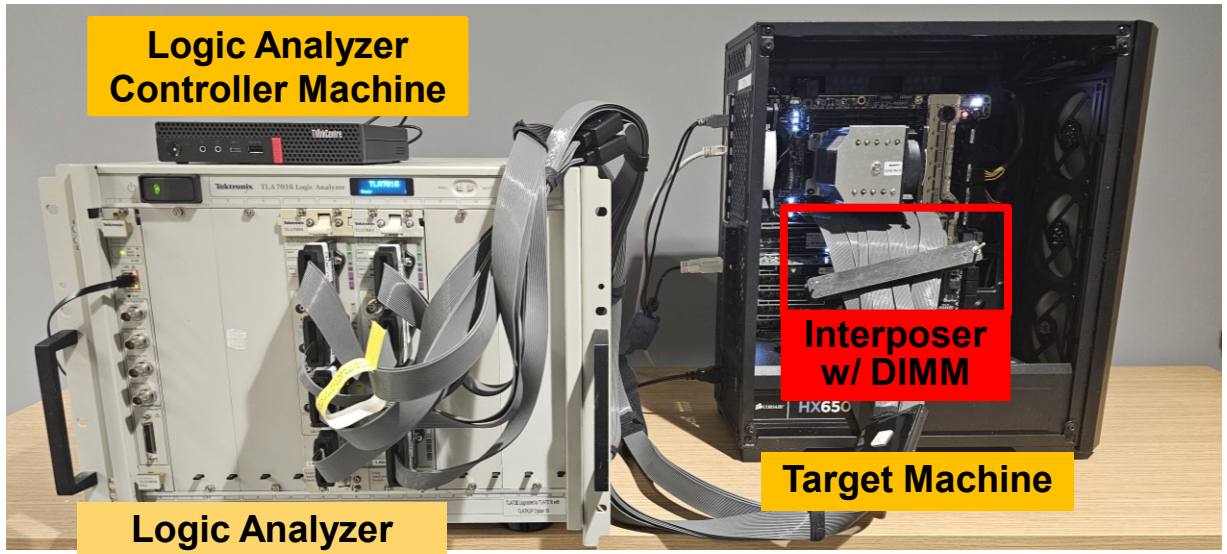
Solution:

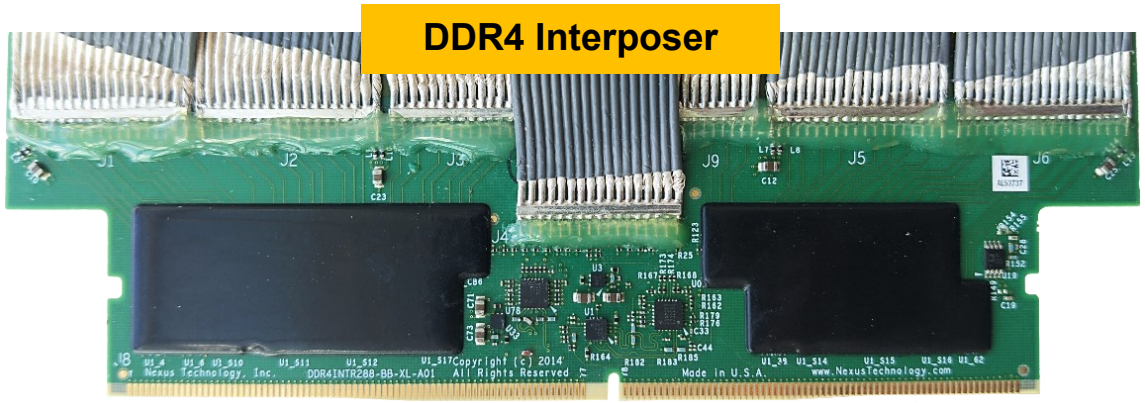
Use a logic analyzer to capture DRAM bus traffic

S2: Recover the ECC Matrix with Logic Analyzer

Solution:

Use a logic analyzer to capture DRAM bus traffic





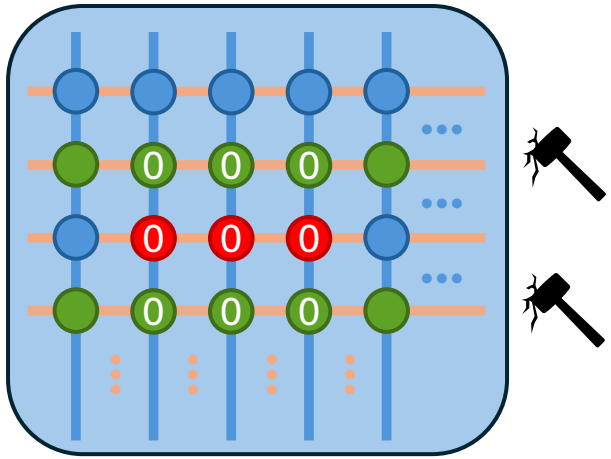
R_DDR4A_2B Address	R_DDR4A_2B Mnemonics		R_DDR4A_2B DataHi	R_DDR4A_2B DataLo	R_DDR4A_2B ChekBits
000208	RD - READ (S0#)	BG:1 BA:0 Logical Bank: 1.0	----------	----------	----------
----------	READ DATA		96689012	24ED541F	30
----------	READ DATA		A033BFD0	CC0BE881	62
----------	READ DATA		E5738D55	1CBC7EC6	B7
----------	READ DATA		C21322F1	BE0139B5	87
----------	READ DATA		DE67D7F8	AC0E54B8	7F
----------	READ DATA		192FE467	813742B2	41
----------	READ DATA		C8528611	62204AFF	A4
----------	READ DATA		284B52EC	F6FCFC54	BE

Result:

- **Code size: 64bytes to 72bytes**
- **Code distance: 4**
- **See our paper for details!**

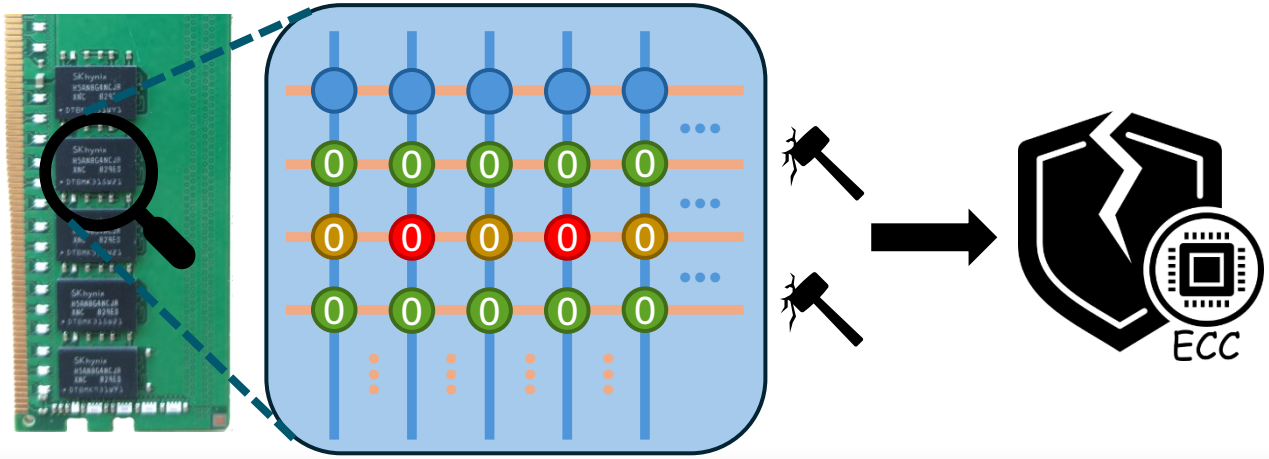
Index: 4 Challenges

C1: Rowhammer on Server DIMMs

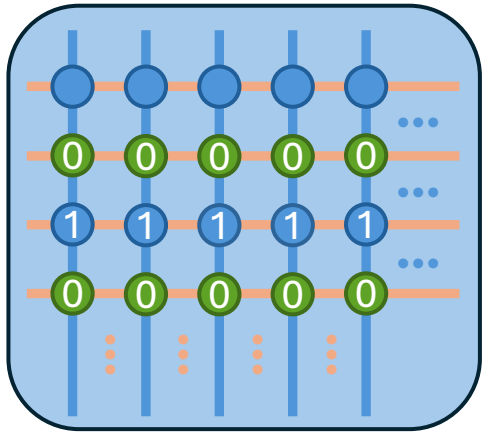


C2: Recover the ECC Matrix

C3: Template DIMMs

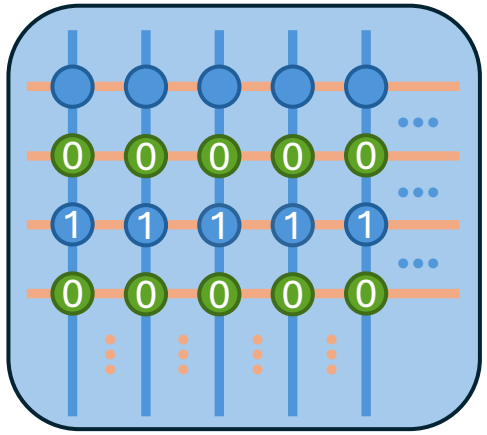


C3: Template DIMMs w/o Crashing



ECC.fail

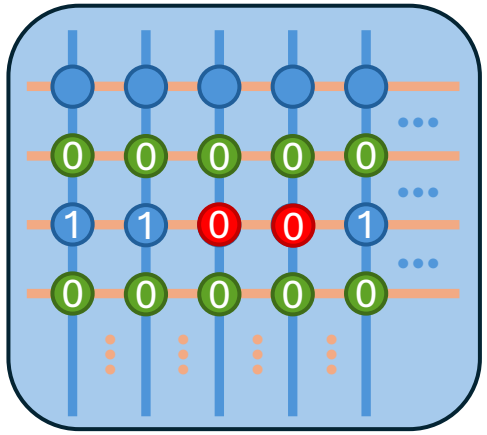
C3: Template DIMMs w/o Crashing



ECC.fail

C3: Template DIMMs w/o Crashing

Crashed



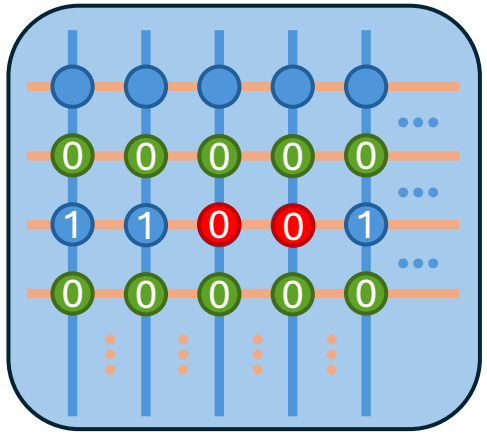
ECC.fail

Challenge 3.1:

Random two or more bit flips will crash

C3: Template DIMMs w/o Crashing

Crashed



ECC.fail

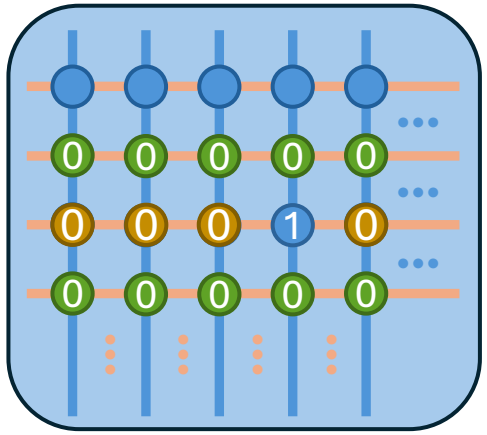
Challenge 3.1:

Random two or more bit flips will crash

Observations:

- **ECC is good at correcting single bit flip**
- **Rowhammer bit flips depend on data**

C3: Template DIMMs w/o Crashing



ECC.fail

Challenge 3.1:

Random two or more bit flips will crash

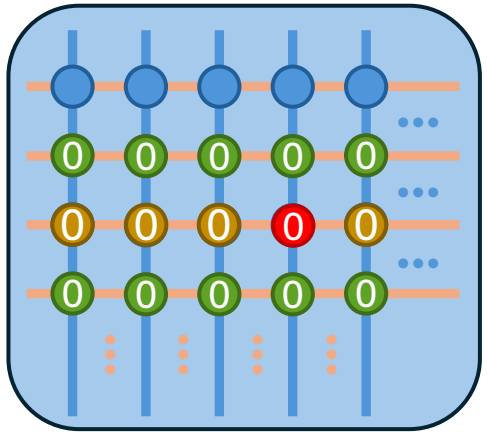
Observations:

- **ECC is good at correcting single bit flip**
- **Rowhammer bit flips depend on data**

Solution 3.1:

Template only one bit at a time

C3: Template DIMMs w/o Crashing



ECC.fail

Challenge 3.1:

Random two or more bit flips will crash

Observations:

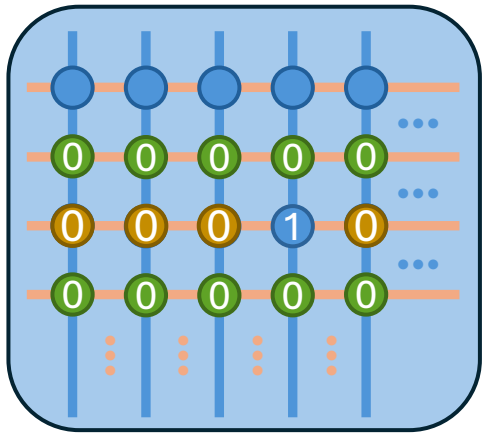
- **ECC is good at correcting single bit flip**
- **Rowhammer bit flips depend on data**

Solution 3.1:

Template only one bit at a time

C3: Template DIMMs w/o Crashing

Correctable



ECC.fail

Challenge 3.1:

Random two or more bit flips will crash

Observations:

- **ECC is good at correcting single bit flip**
- **Rowhammer bit flips depend on data**

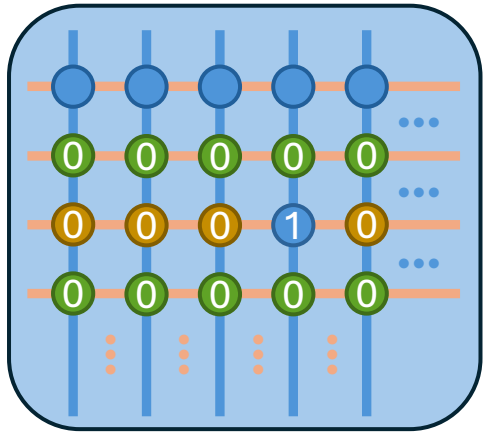
Solution 3.1:

Template only one bit at a time

C3: Template DIMMs and Detect Bit Flips

Challenge 3.2:

Cannot detect bit flip from data read back



ECC.fail

Observation:

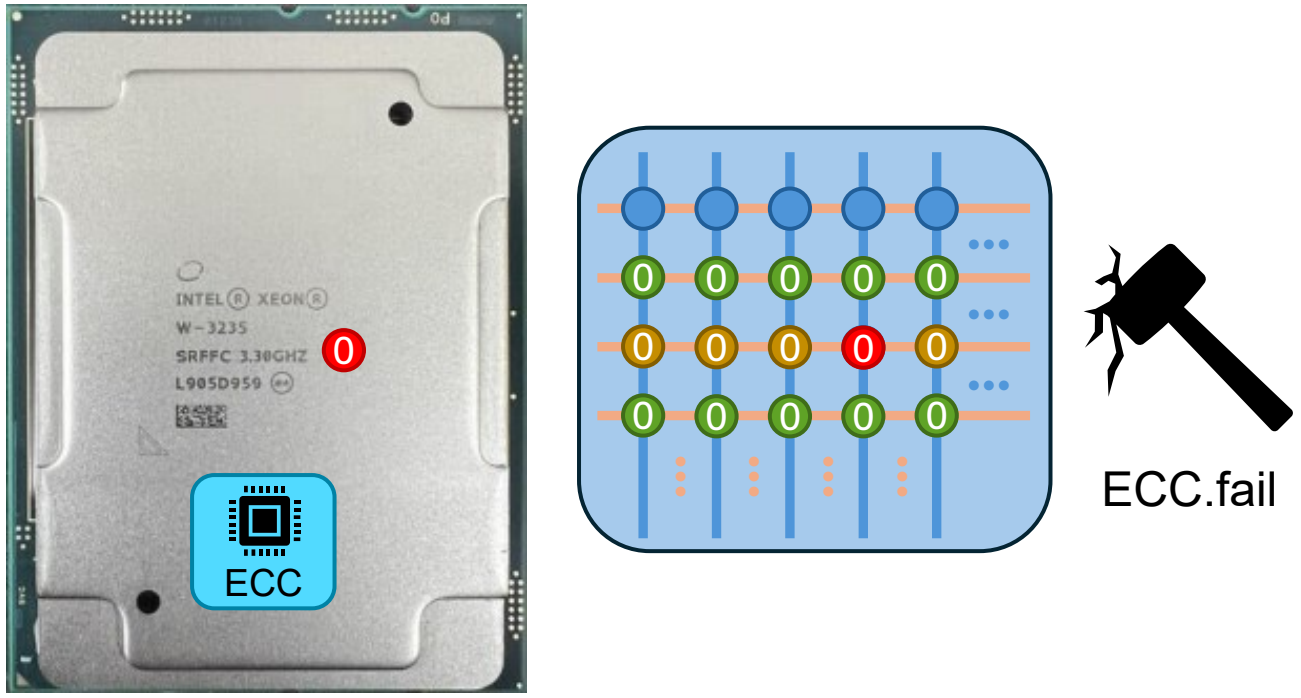
Intel memory controller replays DRAM read

when encounters data corruption

C3: Template DIMMs and Detect Bit Flips

Challenge 3.2:

Cannot detect bit flip from data read back



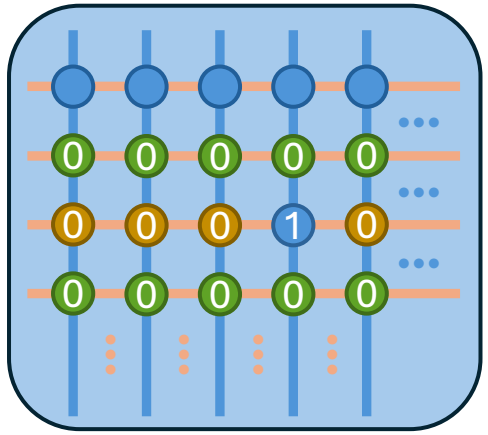
Observation:

Intel memory controller replays DRAM read when encounters data corruption

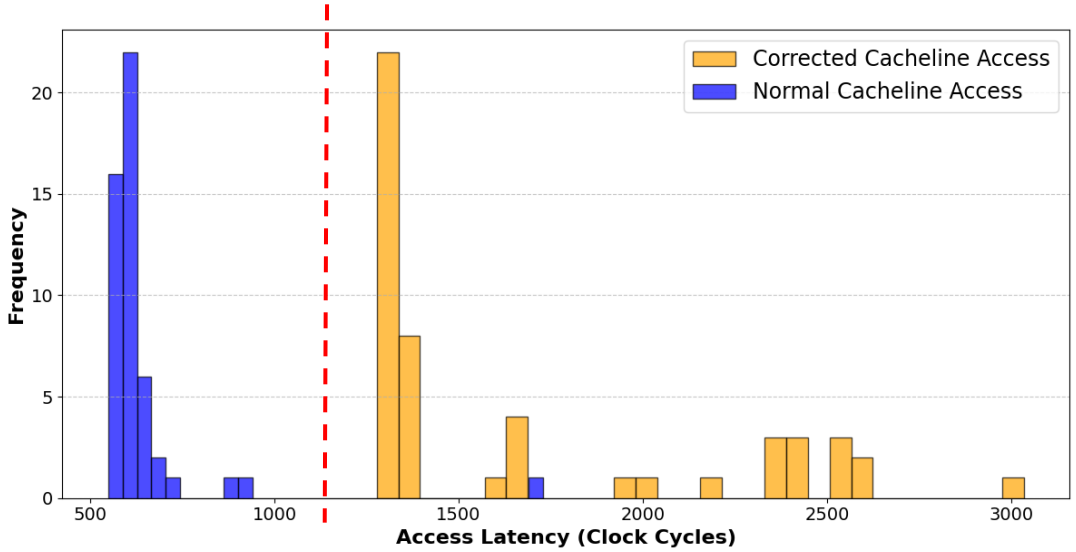
C3: Template DIMMs and Detect Bit Flips

Challenge 3.2:

Cannot detect bit flip from data read back



ECC.fail



Observation:

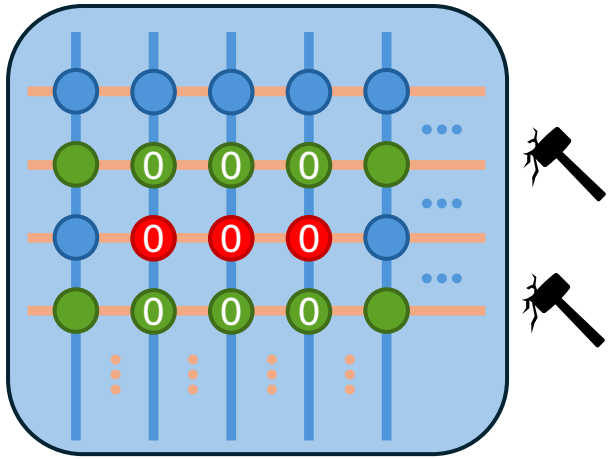
Intel memory controller replays DRAM read when encounters data corruption

Result:

Use access latency side-channel for DIMM templating

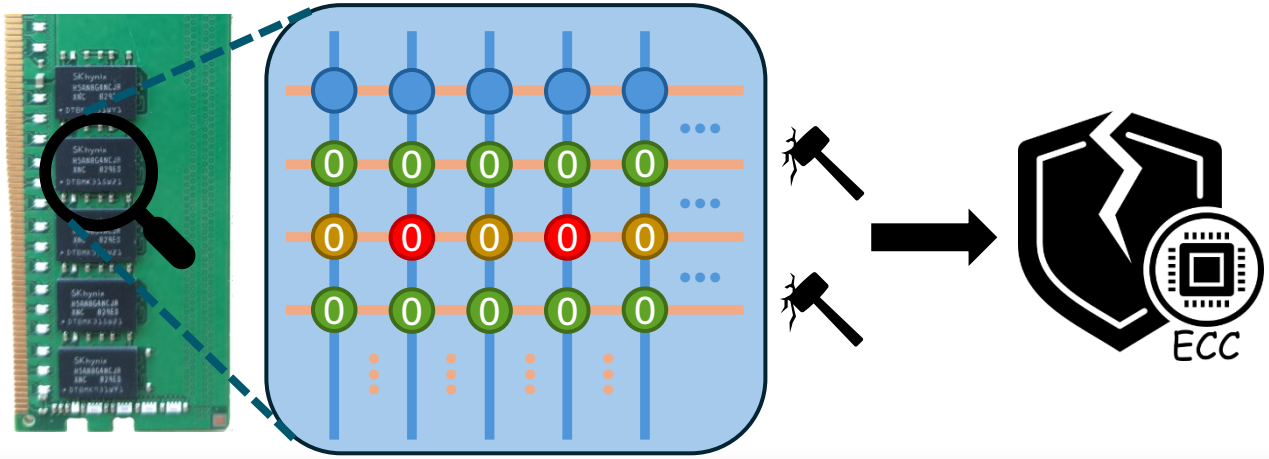
Index: 4 Challenges

C1: Rowhammer on Server DIMMs



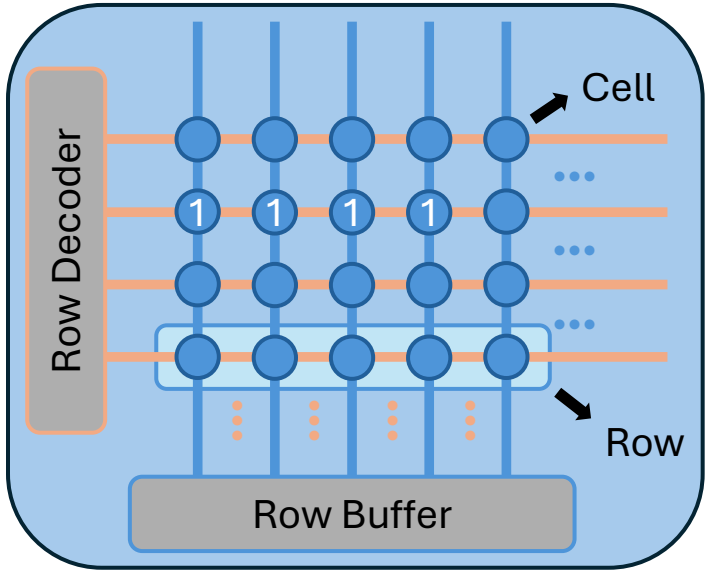
C2: Recover the ECC Matrix

C3: Template DIMMs

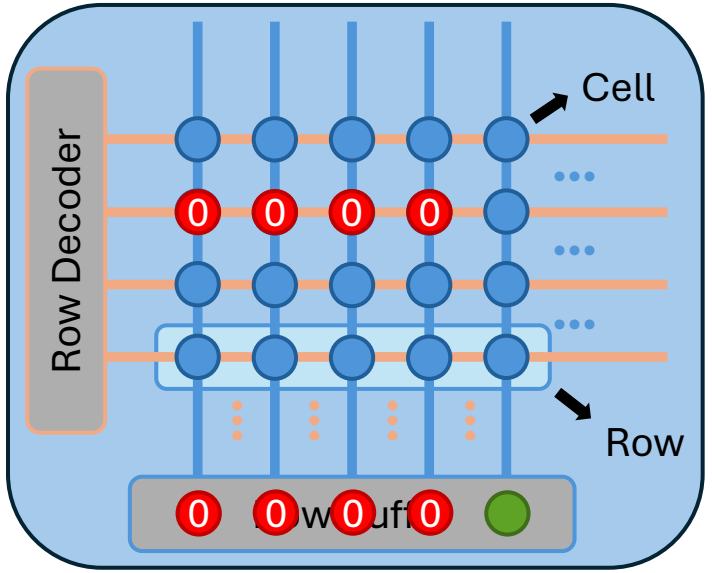


C4: End-to-End Attack

C4: End-to-End Attack



C4: End-to-End Attack



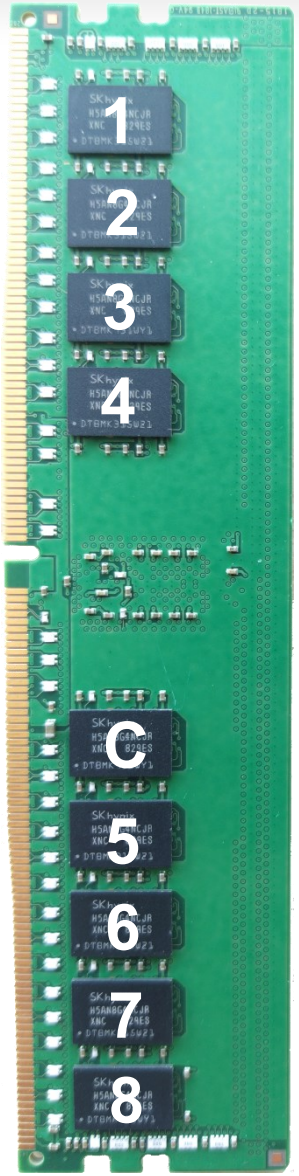
No Error

Challenge 4: Hard to find 4 bit flips at usable location

C4: End-to-End Attack

Observation:

Intel server supports ChipKill error correction capability

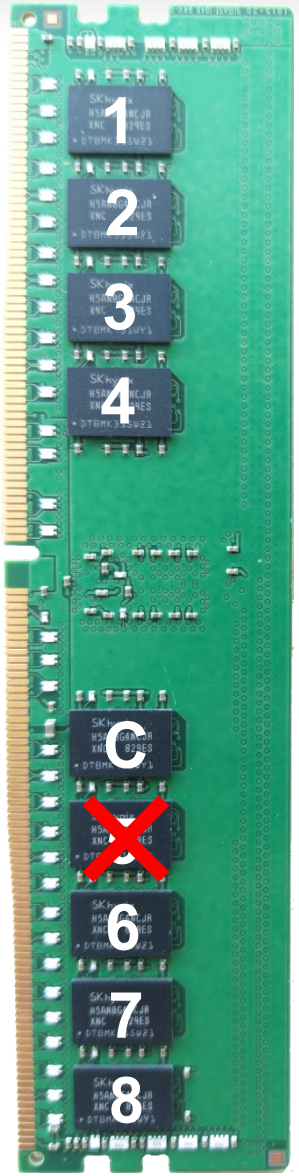


C4: End-to-End Attack

Observation:

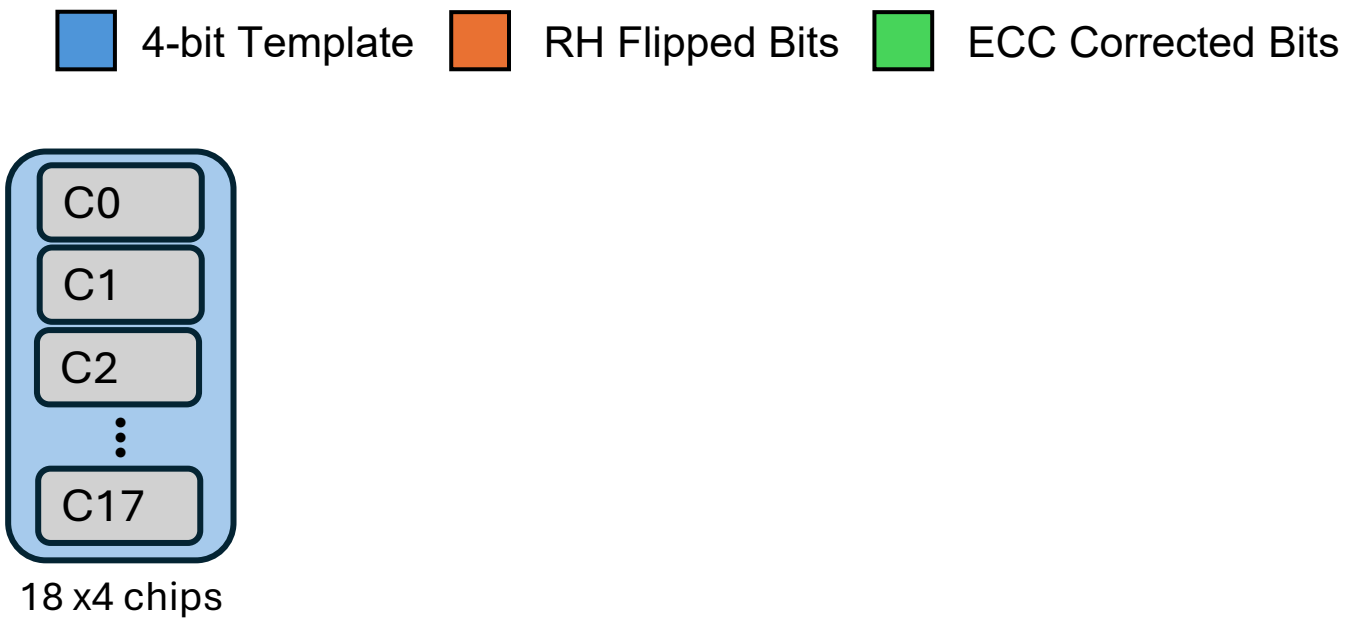
Intel server supports ChipKill error correction capability

Correctable



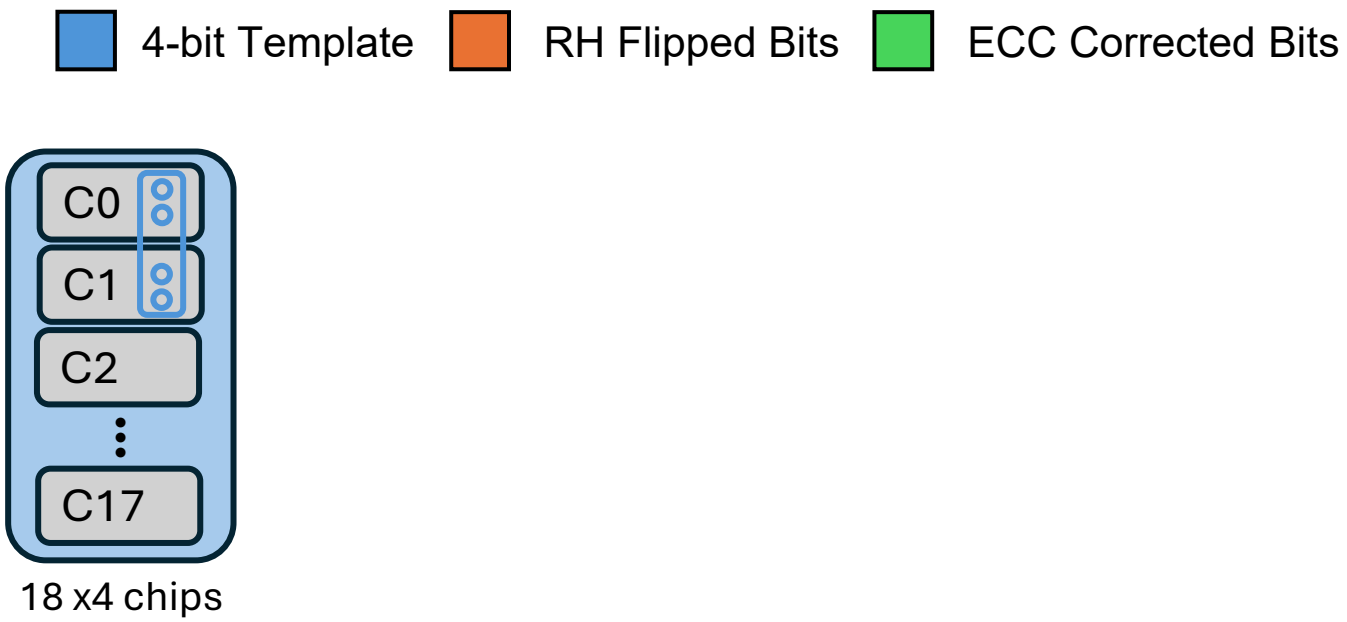
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



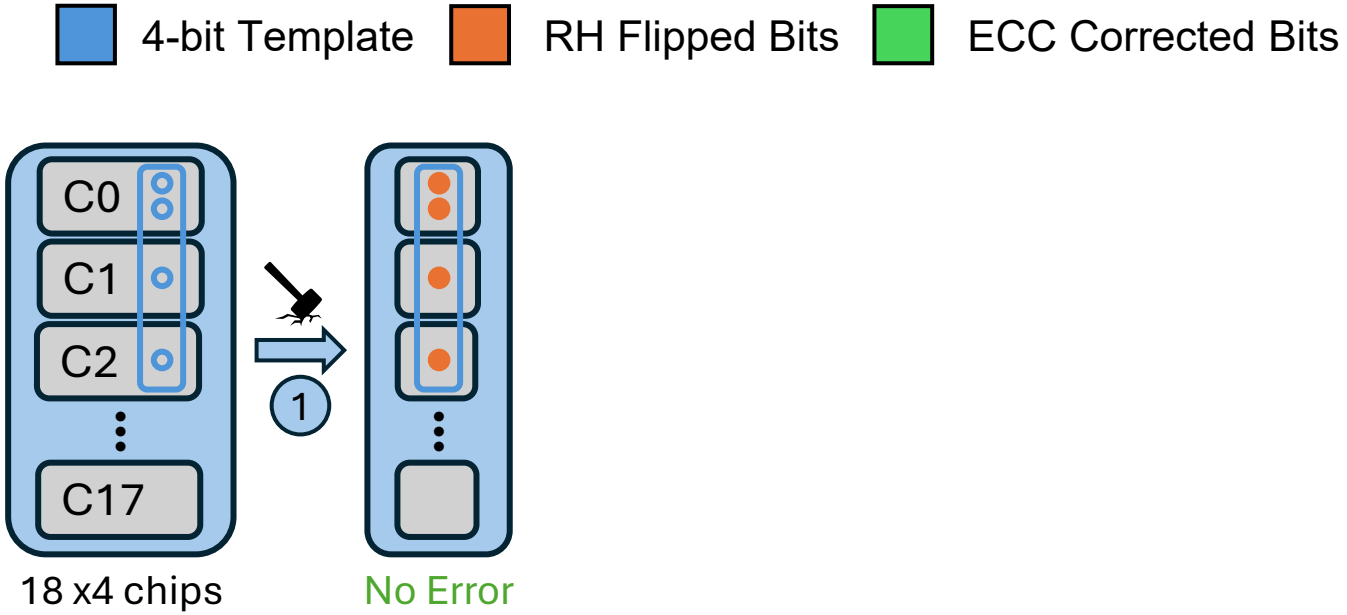
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



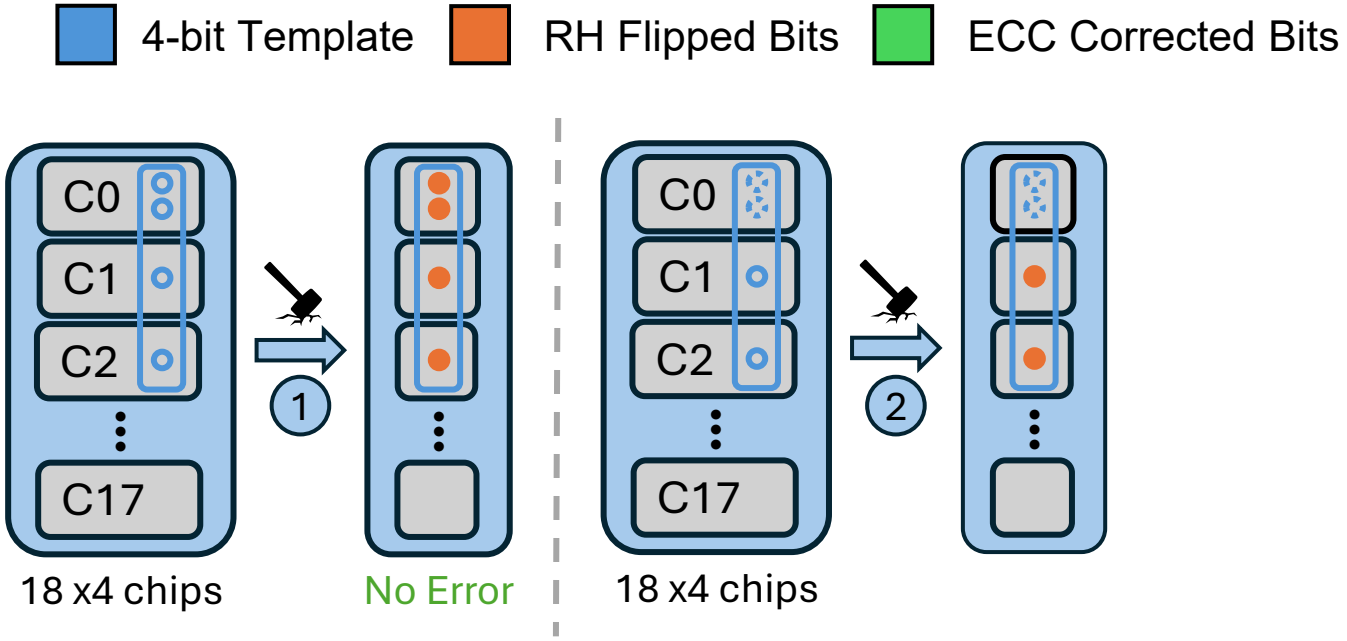
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



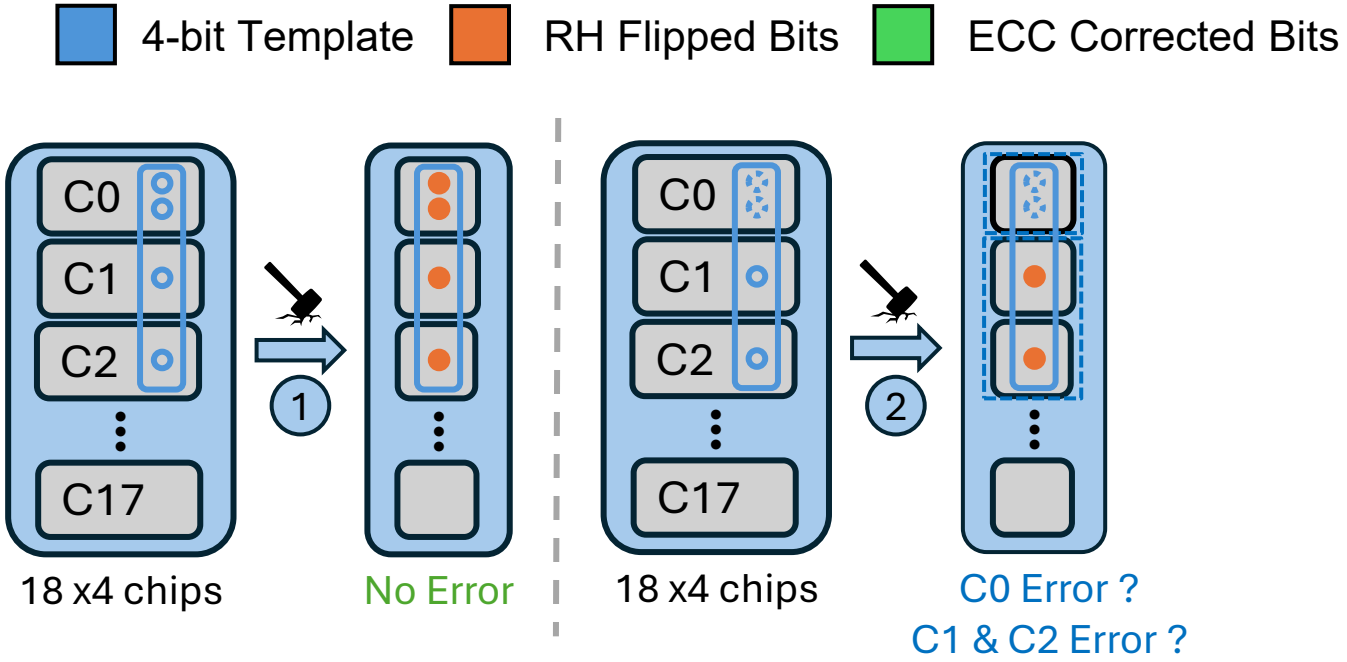
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



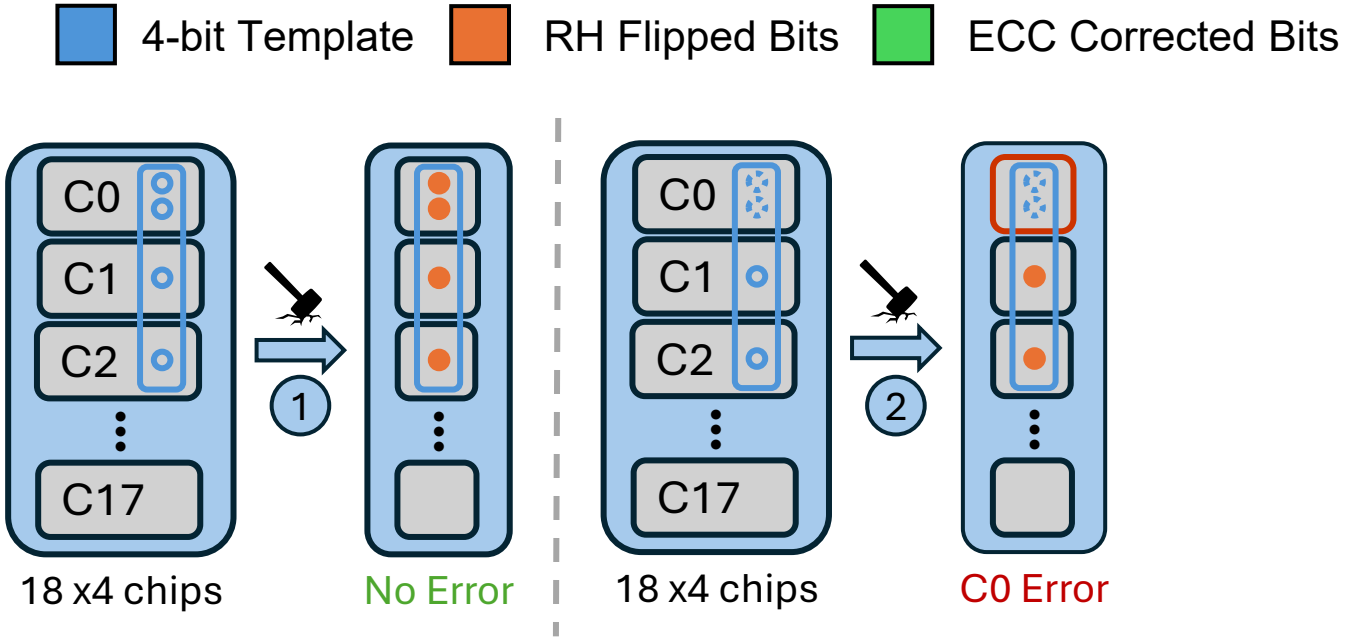
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



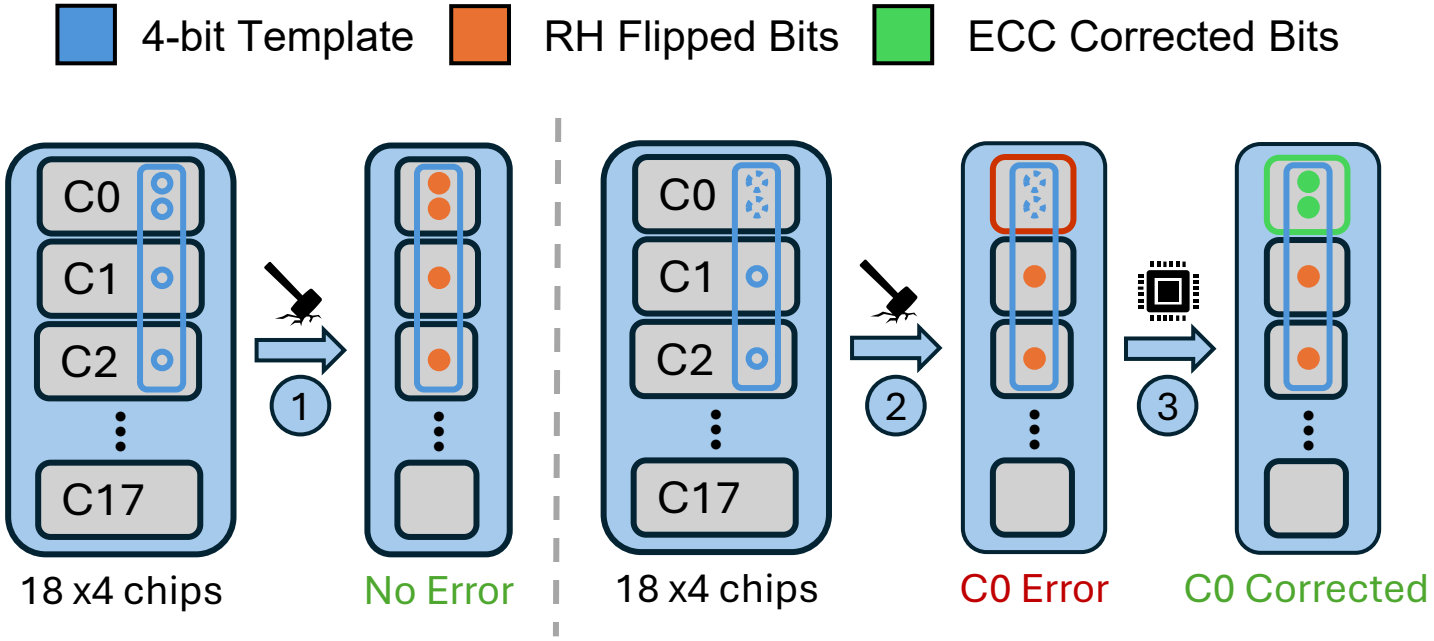
S4: End-to-End Attack

Solution: Exploit ChipKill error correction capability on Intel CPUs



S4: End-to-End Attack

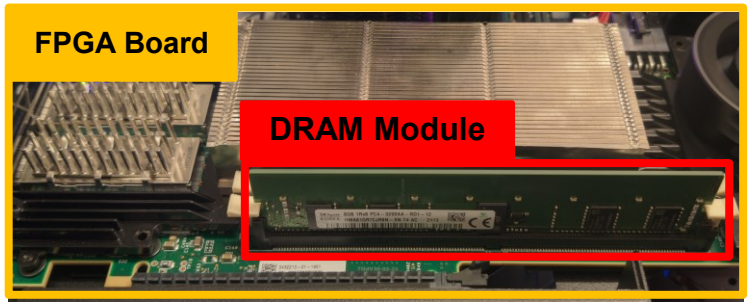
Solution: Exploit ChipKill error correction capability on Intel CPUs



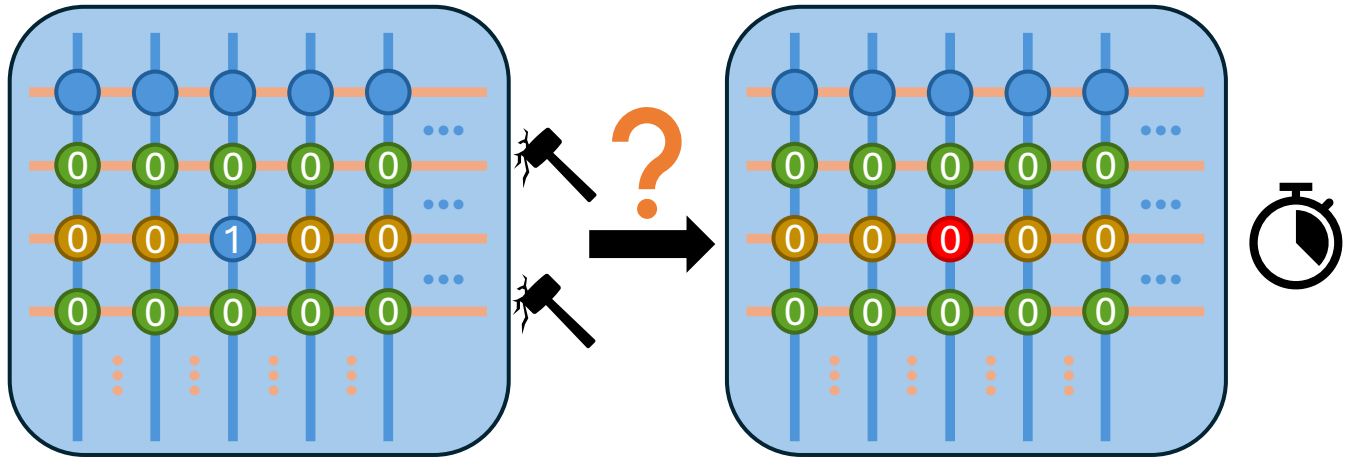
Result: Flipping 2 bits leads to ECC bypass

End-to-End Attack Overview

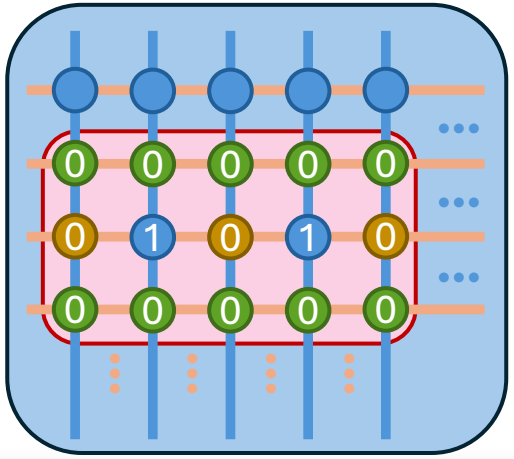
Step 1: Recover DRAM TRR and CPU ECC matrix (offline)



Step 2: Template memory for ECC bypass



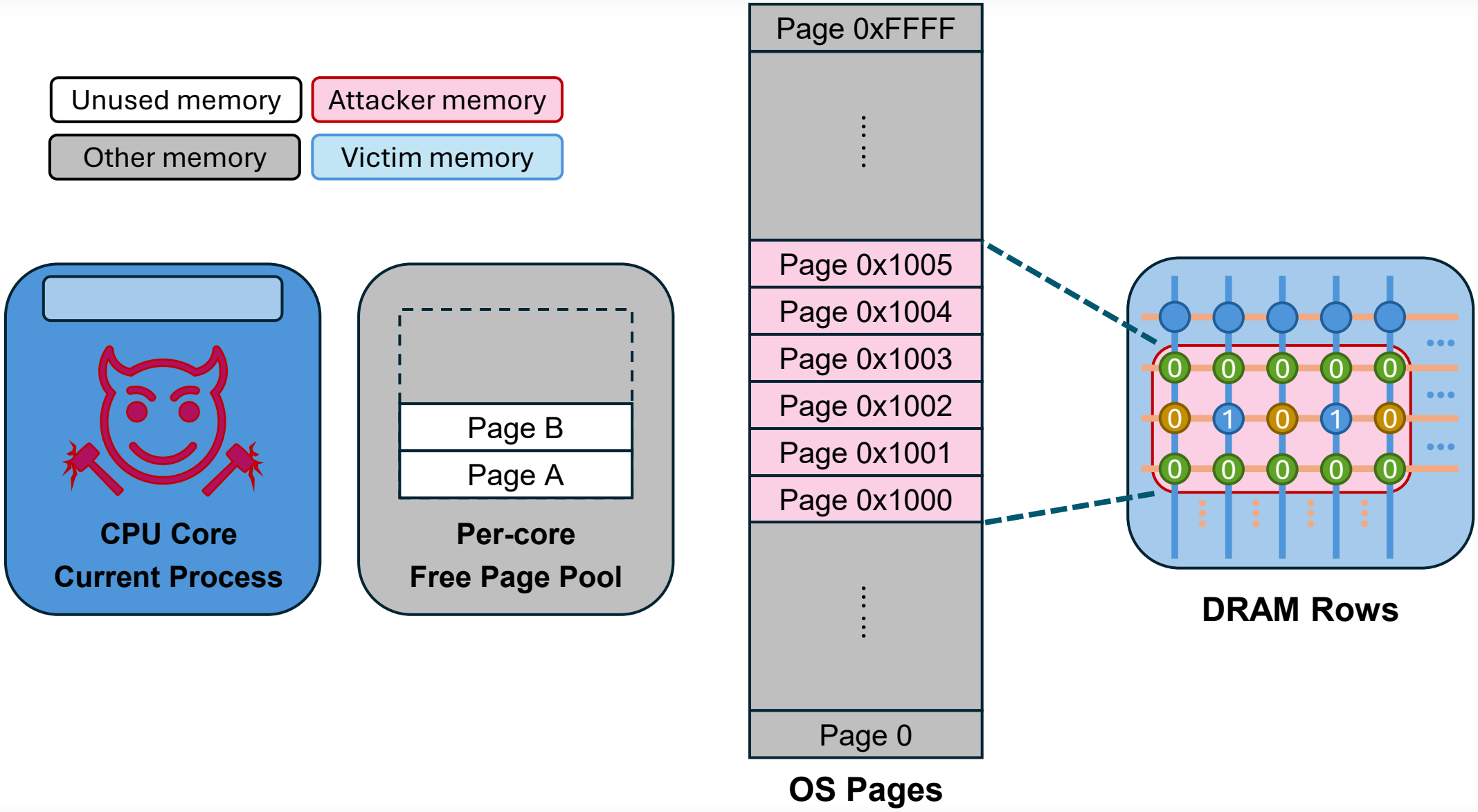
Step 3: Massage memory allocator



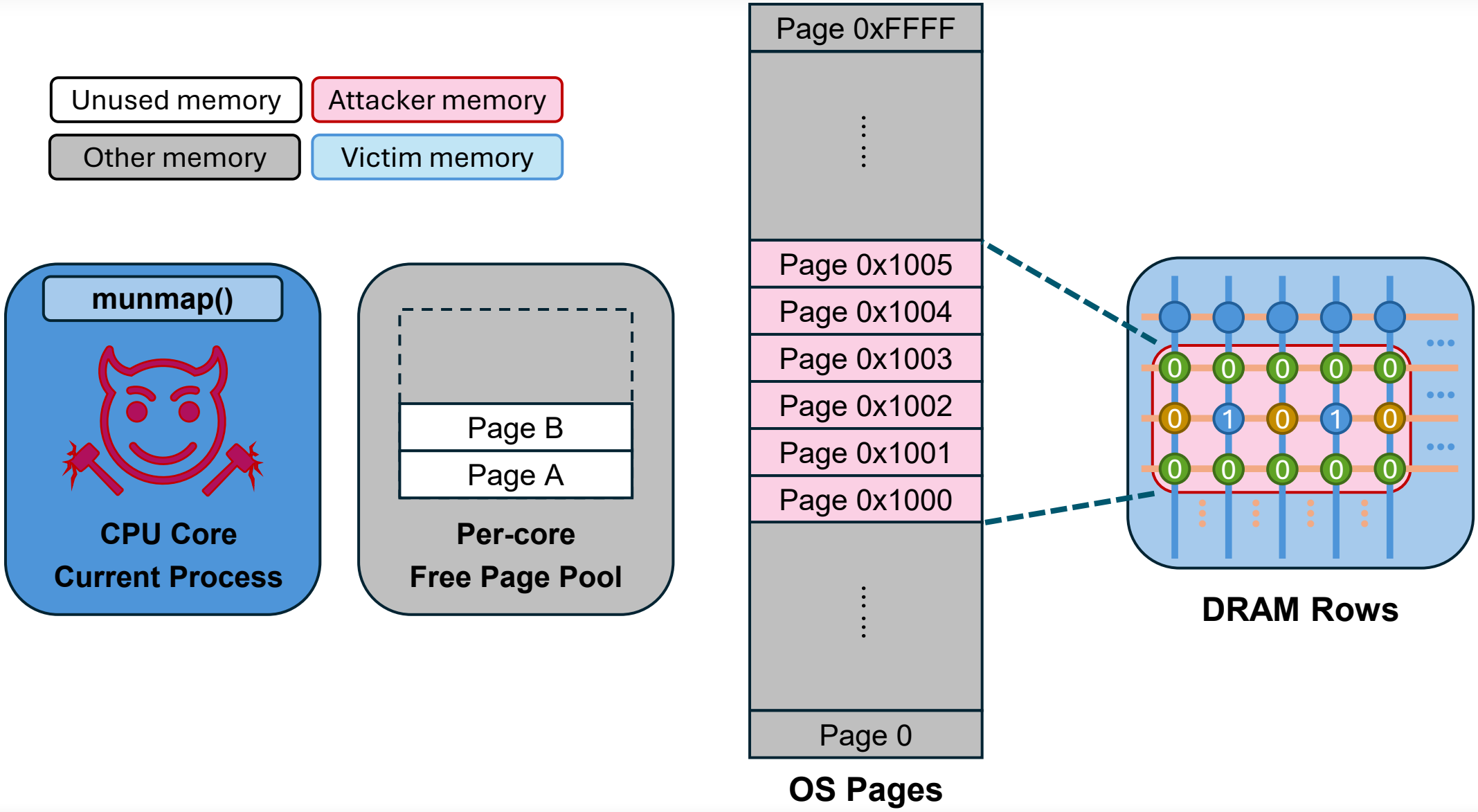
Attacker memory

Victim memory

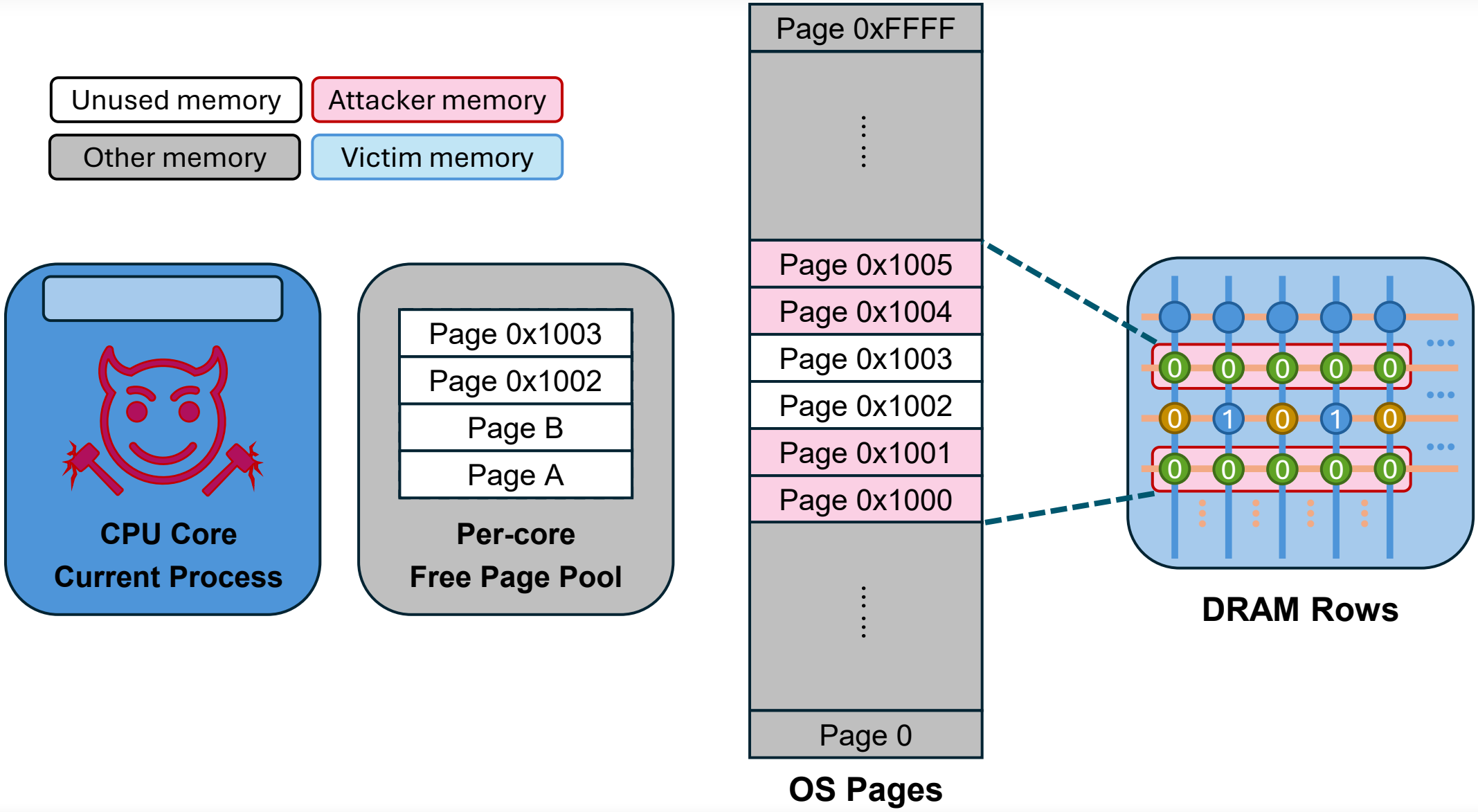
E2E Attack Step 3: Memory Allocator Massaging



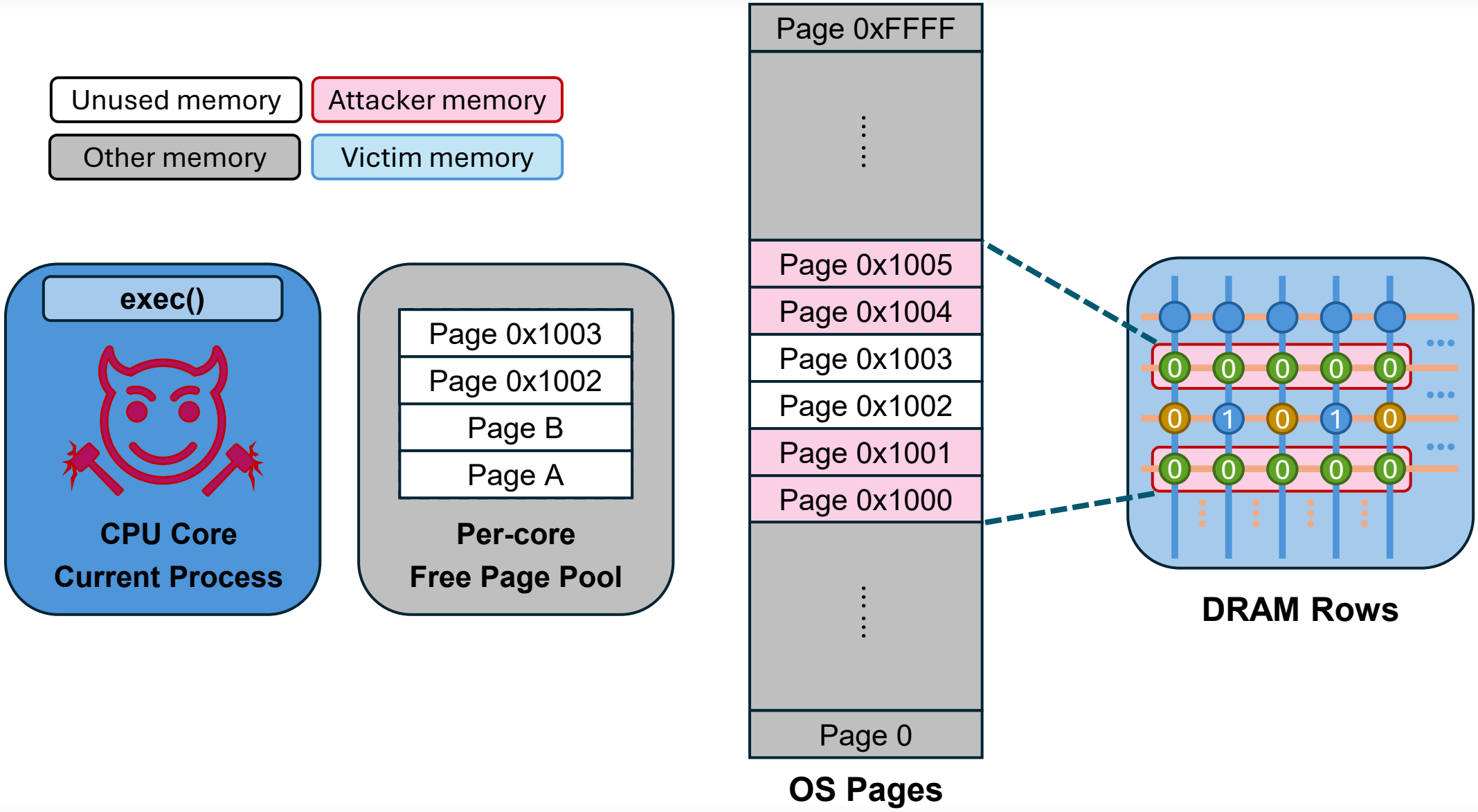
E2E Attack Step 3: Memory Allocator Massaging



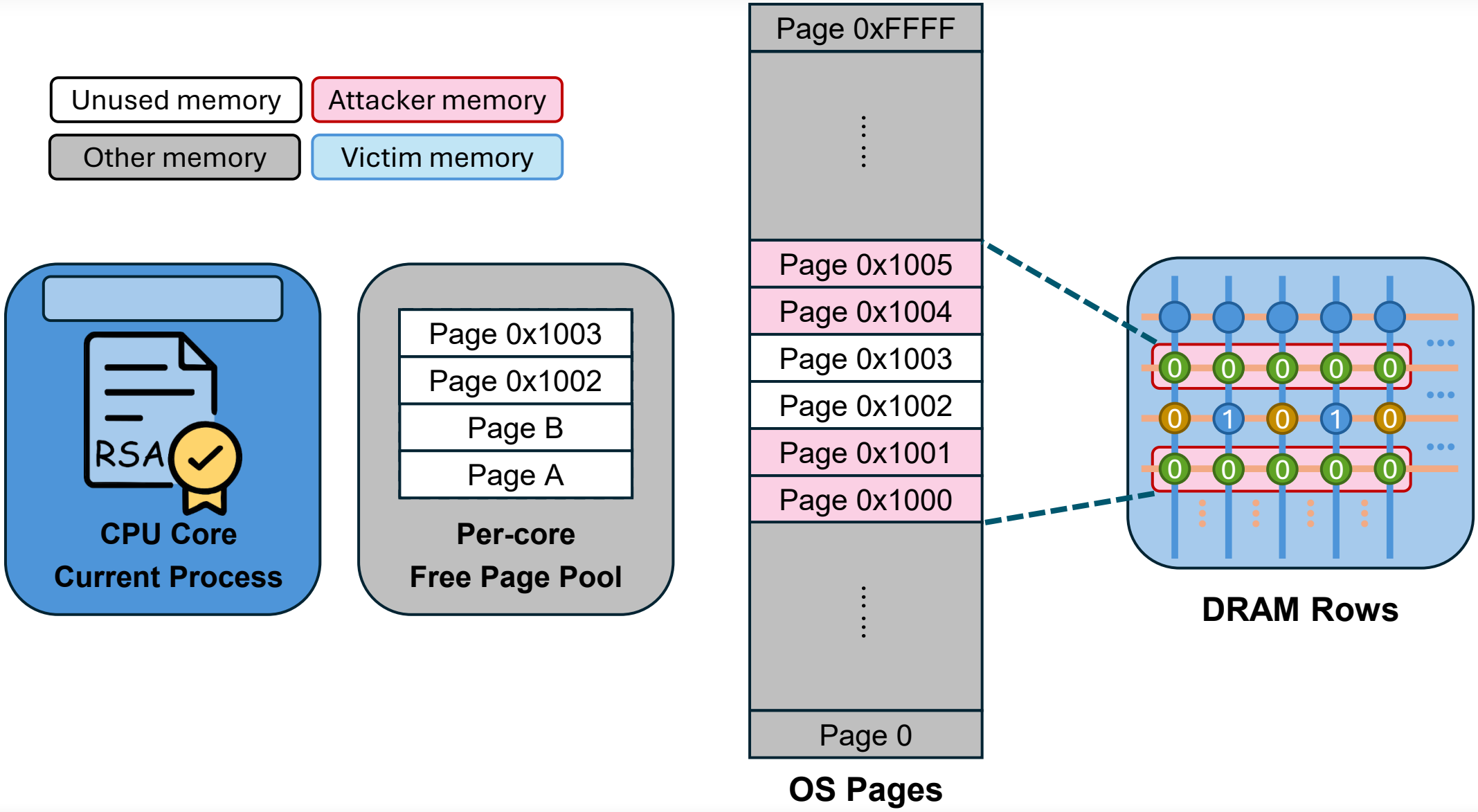
E2E Attack Step 3: Memory Allocator Massaging



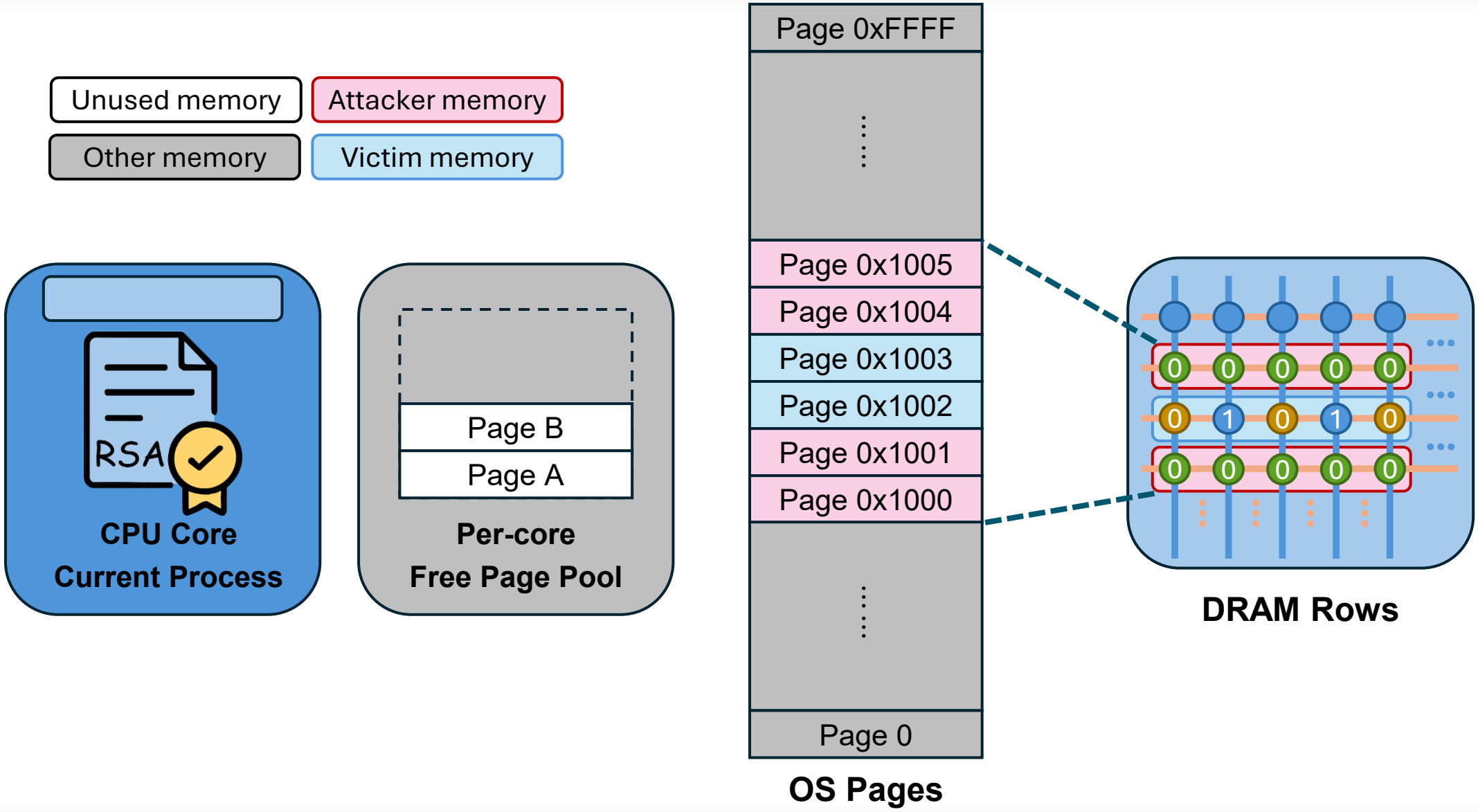
E2E Attack Step 3: Memory Allocator Massaging



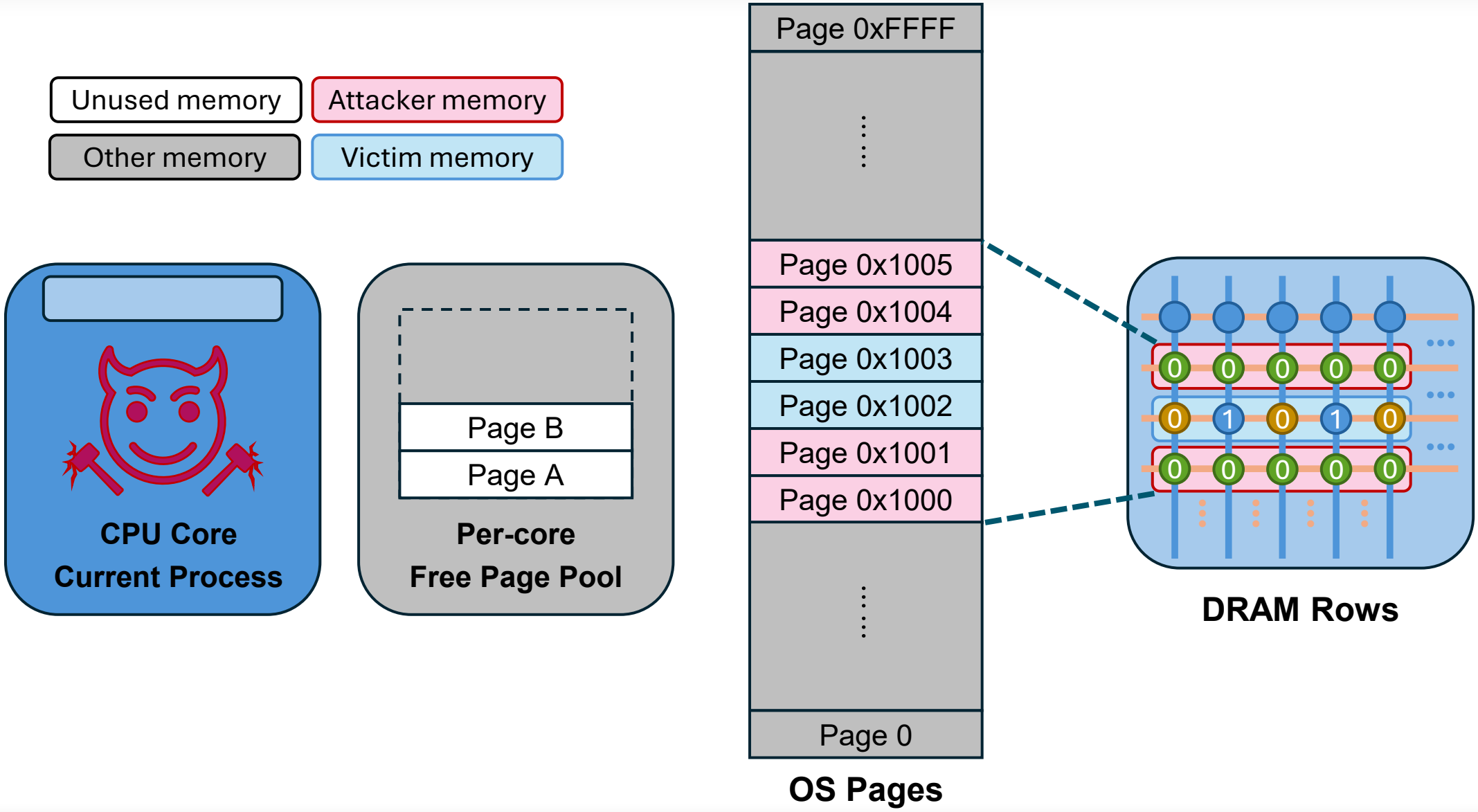
E2E Attack Step 3: Memory Allocator Massaging



E2E Attack Step 3: Memory Allocator Massaging

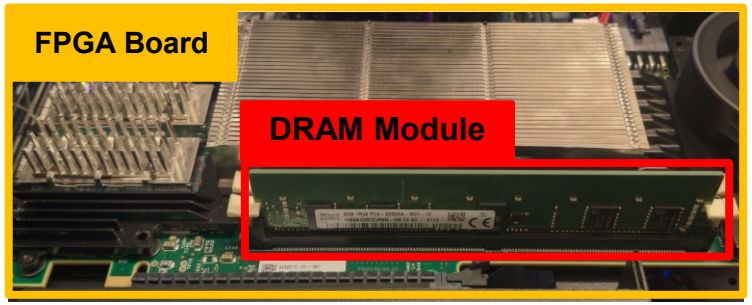


E2E Attack Step 3: Memory Allocator Massaging

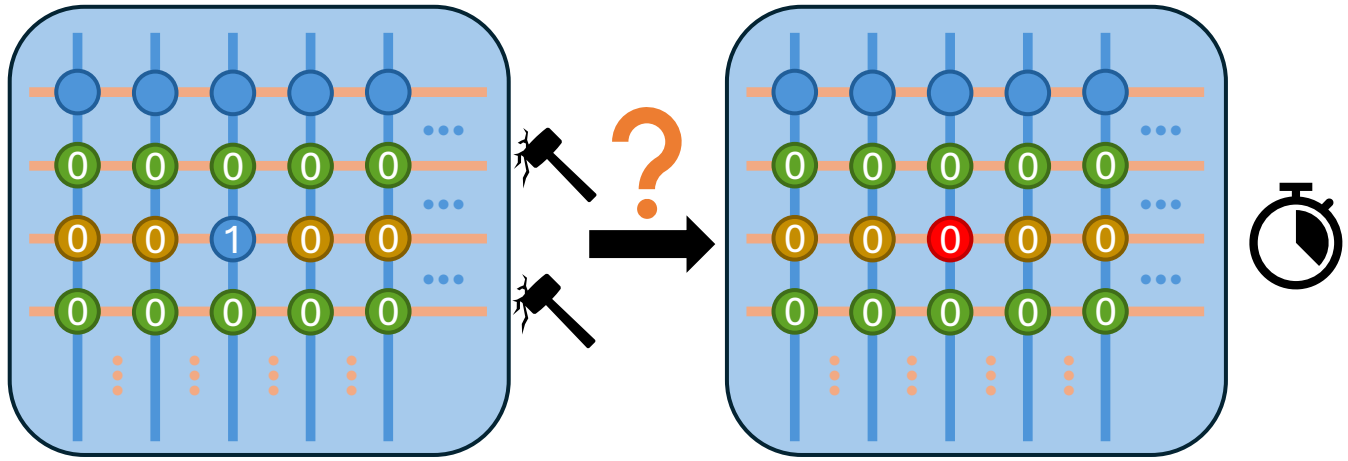


End-to-End Attack Overview

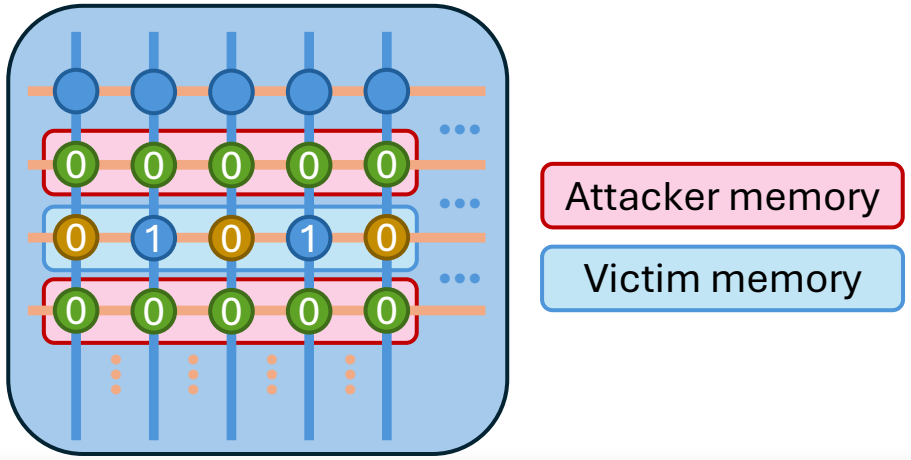
Step 1: Recover DRAM TRR and CPU ECC matrix (offline)



Step 2: Template memory for ECC bypass



Step 3: Massage memory allocator

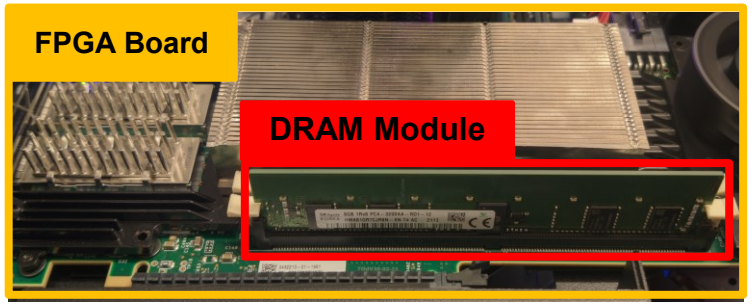


Attacker memory

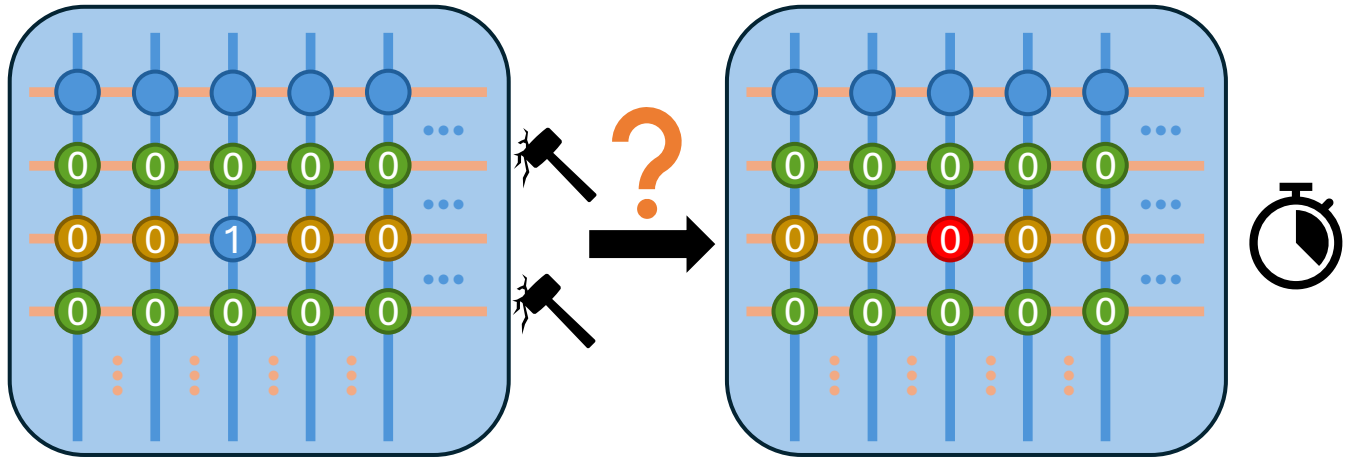
Victim memory

End-to-End Attack Overview

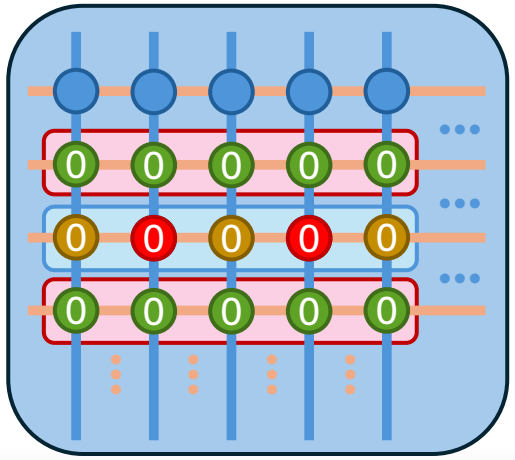
Step 1: Recover DRAM TRR and CPU ECC matrix (offline)



Step 2: Template memory for ECC bypass



Step 3: Massage memory allocator



Attacker memory

Victim memory

Step 4: Rowhammer to exploit RSA signature

Results:

- **Get Rowhammer bit flips: avg. 2.5h**
- **Break RSA signatures: avg. 10h**

Thank you for listening!

Walter Wang
walwan@gatech.edu

https://ecc.fail
https://architecture.fail