Modeling Rowhammer in the gem5 simulator

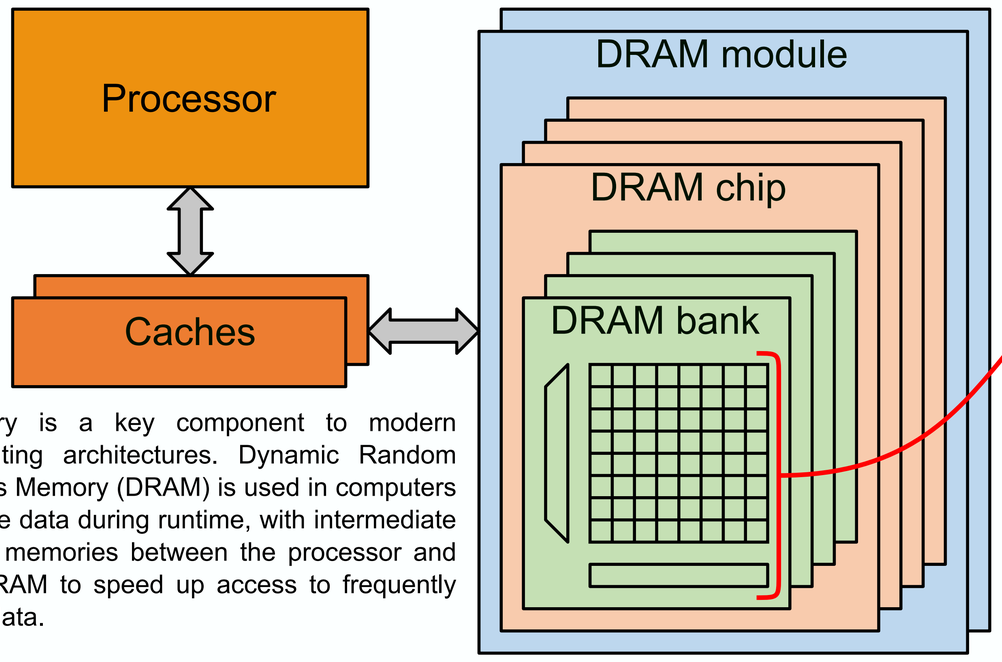
Loïc France, Florent Bruguier, Maria Mushtaq, David Novo, Pascal Benoit

Modeling Rowhammer in the gem5 simulator.

Loïc France, Florent Bruguier, Maria Mushtaq, David Novo, Pascal Benoit

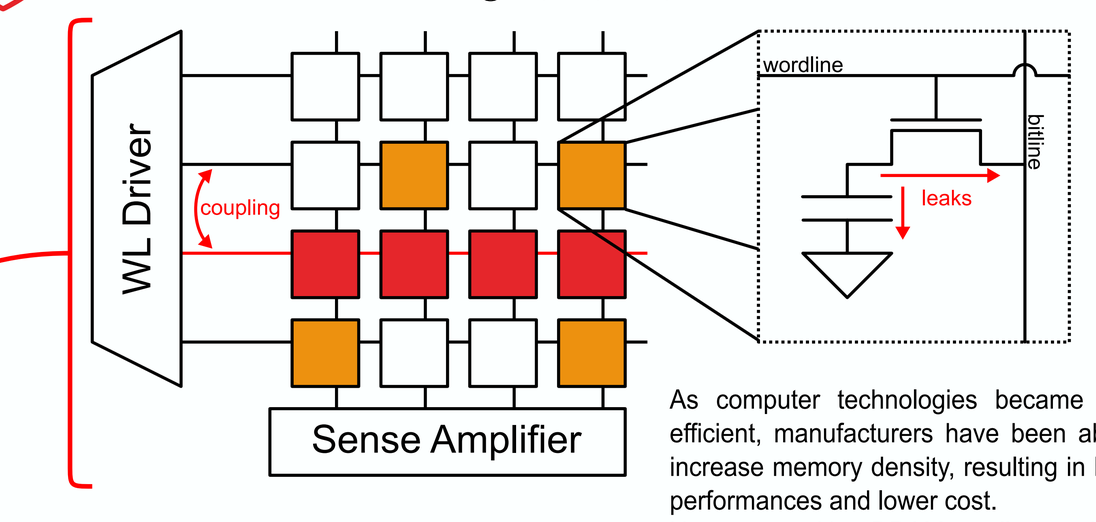
CHES 2022

① DRAM

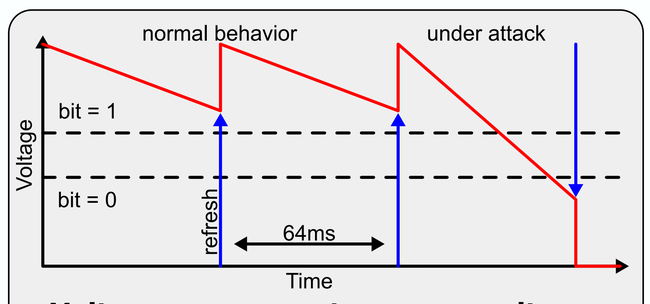


Memory is a key component to modern computing architectures. Dynamic Random Access Memory (DRAM) is used in computers to store data during runtime, with intermediate cache memories between the processor and the DRAM to speed up access to frequently used data.

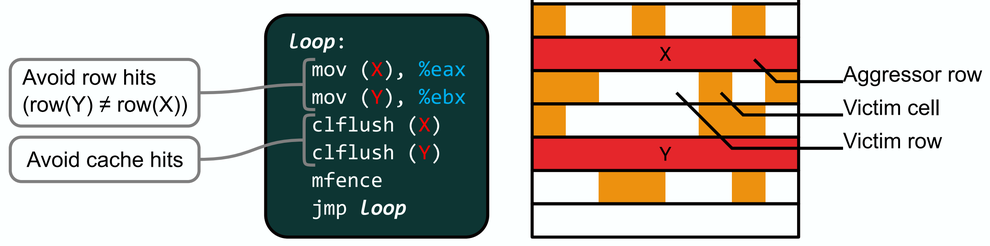
② Rowhammer



As computer technologies became more efficient, manufacturers have been able to increase memory density, resulting in better performances and lower cost.
However, making DRAM smaller resulted in higher vulnerability to disturbance errors: activating a row slightly disturbs adjacent rows due to electromagnetic coupling between wordlines. Repeated activation of neighbors of a victim row can empty the charge of capacitors, effectively deleting the data (bit-flip).

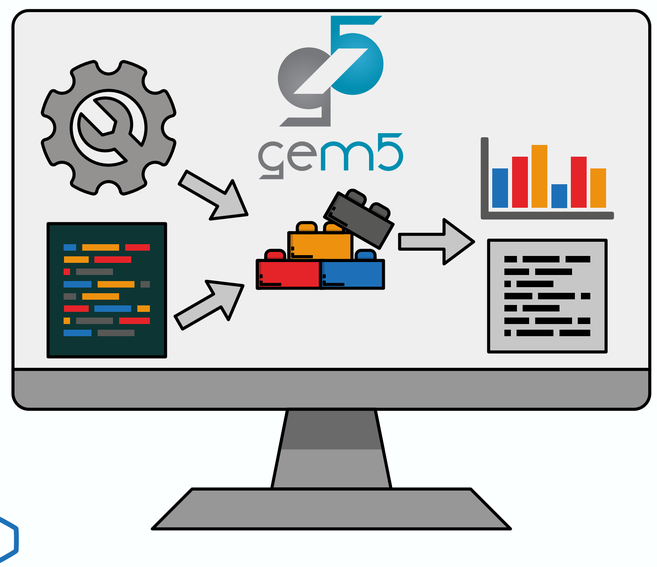


Voltage accross a storage capacitor



This error rapidly became a major threat as attacks exploiting this error, named Rowhammer (RH) attacks, appeared to precisely flip bits to gain kernel privileges, even from sandboxes or without a malicious program running on the victim system.

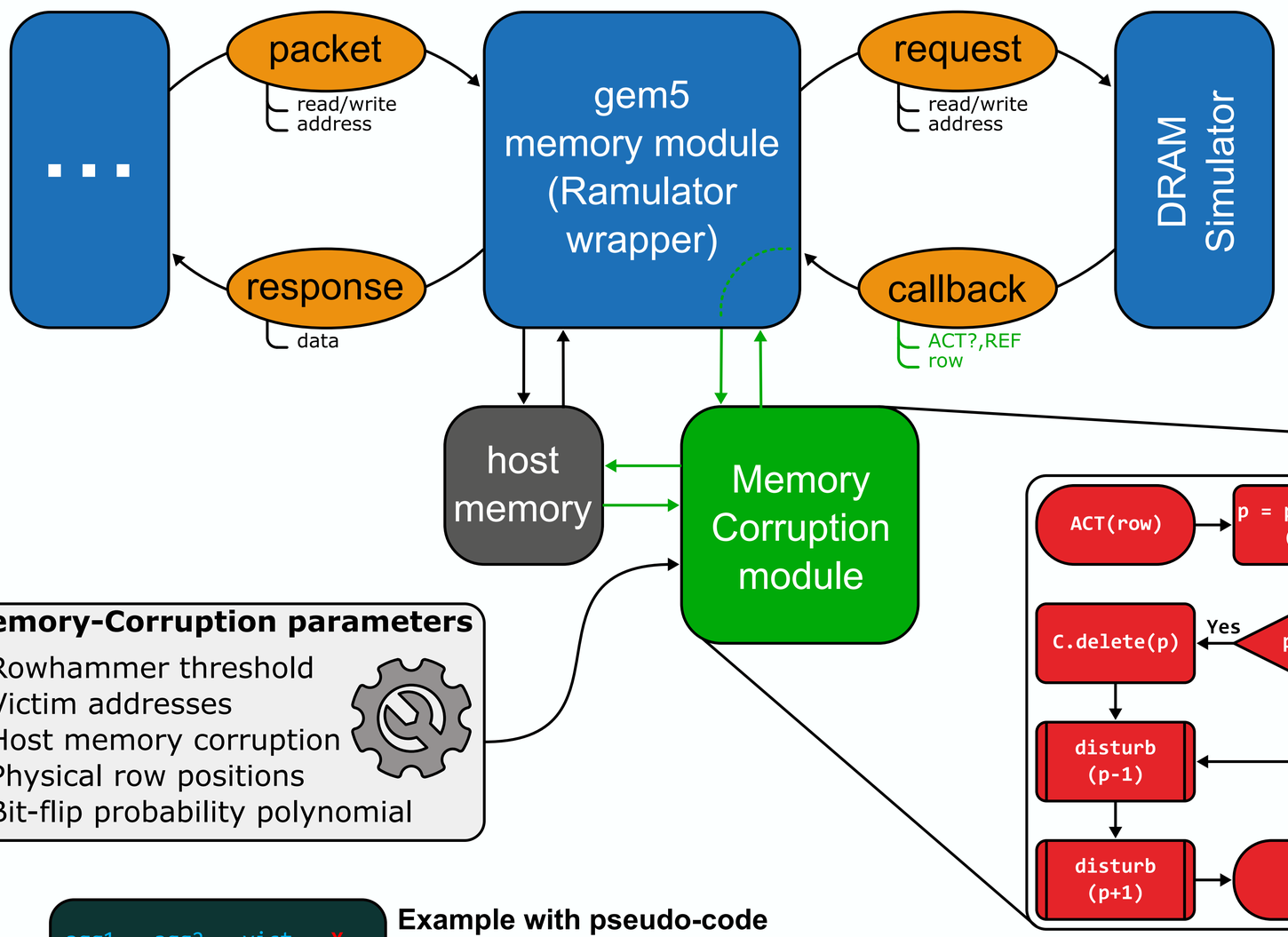
③ gem5 & Ramulator



gem5 is a modular computer architecture simulator used for research. It is used to define a custom architecture with cores and memories, and simulate this architecture running programs and operating systems.
However, its memory simulator is not timing-accurate.
Ramulator was introduced to bring timing-accurate main memory simulation to gem5.

Despite a timing-accurate main memory simulation with Ramulator, gem5 does not provide any implementation of unintended memory modifications like bit-flips caused by Rowhammer attacks. We created a gem5 module to introduce memory corruption to architectural simulations. This modification will allow the development of new attack methods and associated counter-measures.

④ Memory-Corruption Module

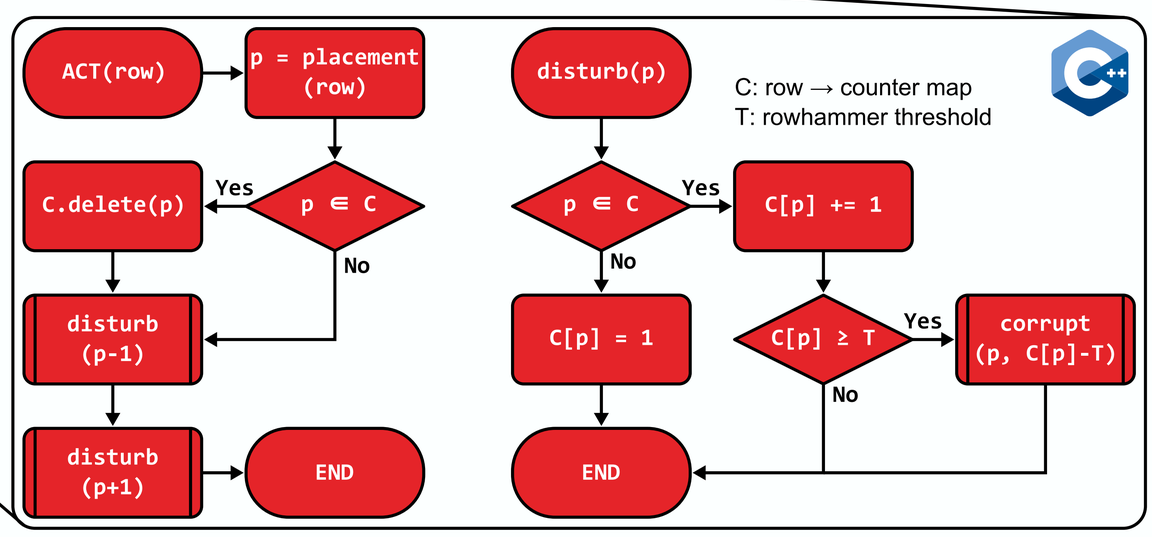


To simulate the memory corruption due to the RH effect, we created a Memory-Corruption (M-C) module, and integrated it inside gem5 by modifying Ramulator and its integration in gem5. The M-C module stores a map to associate DRAM rows to counters.
When the counter for a row exceeds the threshold, the flip probability is calculated as P_f = clamp(f(C[p]-T),[0,1]), with f being the specified polynomial. All bits set to 1 from the row have a probability P_f to flip.

The algorithm is described below. p is the physical placment of the row, p-1 and p+1 are its two neighbor rows. C is the associative map and T is the corruption threshold.

Memory-Corruption parameters

- Rowhammer threshold
- Victim addresses
- Host memory corruption
- Physical row positions
- Bit-flip probability polynomial



Example with pseudo-code

```
agg1 = agg2 = vict = X
agg1.row--
agg2.row++
*vict = FFFFFFFFh
print(time, *vict)
attack(agg1, agg2, 1000)
print(time, *vict)
```

```
[code]1.197ms: victim = FFFFFFFFh
[gem5]...
[gem5]1.523ms: reached hammer threshold (2000)
[gem5]...
[code]1.538ms: victim = 0
```

When the DRAM is refreshed (every 64ms) the associative map C is cleared, effectively resetting all counters.

Recherche soutenue financièrement par le Ministère des armées – Agence de l'innovation de défense

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
University of Montpellier, CNRS
161, rue Ada, 34095 Montpellier cedex 5
{firstname}.{lastname}@lirmm.fr