Implementation of Rowhammer Effect in gem5

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

Implementation of Rowhammer Effect in gem5

Loïc France, Florent Bruguier, Maria Mushtaq, David Novo, and Pascal Benoit
LIRMM, University of Montpellier, CNRS, Montpellier, France
E-mail: {firstname}.{lastname}@lirmm.fr

Abstract

Modern computer memories have been shown to have reliability issues. The main memory is the target of a security attack called Rowhammer, which causes bit flips in adjacent victim cells of aggressor rows. Existing architectures simulator don't provide any implementation of unintended memory modifications like bit-flips. In this paper, we propose an implementation of the Rowhammer effect in the gem5 architecture simulator.

1. Introduction

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. As computer technologies became more efficient and smaller, manufacturers have been able to put more memory in much smaller spaces, resulting in better performances and lower cost [3]. However, making DRAM smaller resulted in higher vulnerability to what Kim et al. [5] depicted as disturbance errors: activating a row slightly disturbs adjacent rows due to electromagnetic coupling between adjacent wordlines, and repeated activation of neighbors of a victim row can make the capacitors lose their charge, effectively deleting the data.

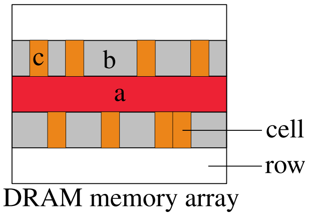

Figure 1. cell-to-cell disturbance in DRAM: hammering row (a) disturbs adjacent rows (b) and delete data stored in victim cells (c)

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 [4].

To counter the RH attack, multiple mitigation techniques have been proposed, with a lot of them implying a modification of the memory controller and/or a new hardware component in the architecture. However, the most efficient solutions require architectural modifications, e.g on the memory controller. Therefore, most of these solutions only propose a concept and simulate its behavior mathematically.

For those solutions, a simulation environment would help to prove their efficiency against existing and future attacks.

gem5 [1] 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 accurate regarding timings and does not simulate memory corruption due to RH attacks. Ramulator [6] was introduced to bring timing-accurate main memory simulation to gem5 but is not capable of simulating RH effects.

In this paper, we propose a modification of gem5 to introduce the Rowhammer effect to architectural simulations. This modification will allow us to develop new attack methods and associated counter-measures.

2. Integration of Rowhammer effect in gem5

In order 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 some existing components.

The M-C module stores an associative map where the key is the DRAM row and the value is a counter. When a DRAM row is activated, the concerned row is removed from the map; for each of its neighbor rows, if it is not yet referenced in the map, it is added and its counter initialized to 1, otherwise its counter is incremented. When a counter is incremented, if the value is above the RH threshold, every bit of the line has a low probability of flipping from 1 to 0.

In order to integrate the M-C module, we had to modify existing gem5 and Ramulator components to let the M-C module know when rows are activated: we added the

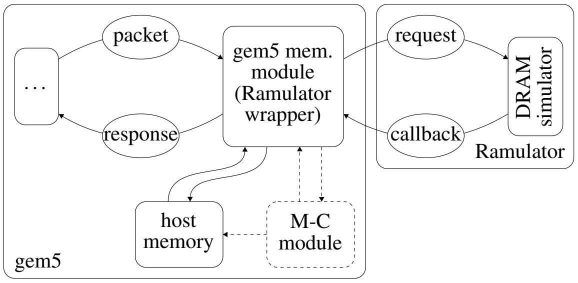

Figure 2. Integration of the M-C module in gem5

row activation information to the request callback when accessing the main memory, and made periodic refresh events, originally generated and handled by Ramulator alone, notify gem5 when they happen.

3. Usage

The M-C module is used to generate bit-flip in the host memory when the simulated memory is the target of a Rowhammer attack. This module can be used to test if attacks achieve memory corruptions and if defense mechanisms can prevent the attack from doing any harm to the memory.

3.1. Configuration

The M-C module can be configured to simulate the RH effect as precisely as possible, and to add information to the log in order to compare and improve the attacks' efficiency. Multiple parameters can be passed to the M-C module:

- The threshold at which the rows start to have bit-flips;
- Enable or disable the modification of the host memory (leaving only a log when the threshold is exceeded);
- The address row to physical placement mapping, used to check which row is adjacent to the activated one;
- A set of target addresses, to limit the possible corruptions to those addresses.

3.2. Limitations

There is some limitations regarding the fidelity of the simulated RH effect.

- The refresh events used by Ramulator are fired with a period corresponding to the refresh interval, which is usually $7.8\mu s = 64ms/8196$. This refresh interval is used by the DRAM controller to refresh a small subset of all DRAM rows. We considered that the activate counters are reset once every 8196 refresh events
- The mapping has to be the same across all DRAM banks.
- The data pattern of the victim and aggressor rows has no impact on the corruption speed
- All bit-flips happen in the same direction (1 to 0).

3.3. Perspectives

We recently published an article where we used traces generated with gem5 to train machine learning (ML) models into detecting RH attacks by classifying hardware event traces [2]. At this time, we had to consider that some programs are harmful even if the simulated memory was not corrupted. Using this new contribution, we are able to create programs that perform memory corruption on the simulated memory, generate datasets from the simulations and train ML models correctly.

This contribution will also allow designing RH mitigation mechanisms with proof of work, and providing working simulations of architecture integrating state of the art mitigation techniques.

4 Conclusion

In this paper, we presented the Memory-Corruption gem5 module which brings the Rowhammer effect simulation to gem5. By generating DRAM bit-flips during Rowhammer attacks, this gem5 module allows researchers and designers to simulate a system under attack and securely create attacks targeting not-yet-available architectures, and test defense mechanisms that cannot be tested on existing platforms.

Acknowledgements

The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-19-CE39-0008 (project ARCHI-SEC). They also acknowledge the French Ministère des Armées – Agence de l'innovation de défense (AID) under grant ID-UM-2019 65 0036.

References

[1] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al. The gem5 simulator. *ACM SIGARCH computer architecture news*, 2011.

[2] L. France, M. Mushtaq, F. Bruguier, D. Novo, and P. Benoit. Vulnerability assessment of the rowhammer attack using machine learning and the gem5 simulator-work in progress. In *Proceedings of ACM SaT-CPS*, 2021.

[3] J. L. Hennessy and D. A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[4] J. S. Kim, M. Patel, A. G. Yağlıkçı, H. Hassan, R. Azizi, L. Orosa, and O. Mutlu. Revisiting rowhammer: An experimental analysis of modern dram devices and mitigation techniques. In *Proceedings of ISCA*, 2020.

[5] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In *Proceedings of ISCA*, 2014.

[6] Y. Kim, W. Yang, and O. Mutlu. Ramulator: A fast and extensible dram simulator. *IEEE Computer architecture letters*, 2015.