DRAM Row-Hammer Attack Reduction Using Dummy Cells

Hector Gomez, Andres Amaya and Elkim Roa
Design Group of Integrated Systems CIDIC - Universidad Industrial de Santander
Bucaramanga, Santander, Colombia
{hector.gomez,andres.amaya1}@correo.uis.edu.co, efroa@uis.edu.co

***Abstract*—This paper describes a low-cost and low-complexity alternative to reduce the occurrence of Row-Hammer attacks. The detection of an undesired attack is based on the use of an additional memory cell —called dummy cell—, with a larger leakage current and thus a higher sensitivity to crosstalk and coupling noise. This characteristic is achieved due to the use of a wider pass transistor and a smaller storage capacitor. One of the most relevant aspects of this solution is the involved additional low-complexity hardware, occupying less than 0.1% of the whole memory. In addition, the dummy cells can be distributed across the whole memory to hinder hackers identification. Simulations on a 65nm CMOS process were done in order to validate the proposed alternative. Process variations for coupling and interconnections were taken into account in a 64x64 memory array, so that the results keep congruence with a memory dedicate, state-of-art, 28nm process.**

***Index Terms*—Security, DRAM, hardware attacker.**

I. Introduction

Devices miniaturization have increased the susceptibility to crosstalk in integrated capacitors have increased due to the close proximity between devices. As consequence, high severity computer failures can occur such as rebooting systems and lost or corrupted data [1].

DRAM memories are some of the devices affected by technology shrinking. Despite miniaturization allows DRAM circuits to have more cells in the same area, increased density has brought also negative impacts in memory performance and reliability. First, reduced memory cell capacitance implies a low noise margin due to the low capacity to hold charge. Second, the proximity between cells increases the electric coupling effects. Finally, the effects of these issues are exacerbated by process variations causing a larger number of cells with higher susceptibility to inter-cell crosstalk [2]–[4].

Coupling noise in adjacent rows, increases the leakage current of memory cells when a specific memory address is repeatedly opened (or activated), read and closed. These increased leakage current can cause disturbance errors in memory devices; specifically, coupling between wordlines is exploited to cause bit flips or data corruption. The latter disturbance is often considered to be the worst defect in regard to computer system quality expectations [1].

The phenomenon of increasing leakage current in cells of adjacent rows due to the consecutive opening of a given row is called row hammering. Recently, many works have proven how row hammering causes bit flips in modern DRAM chips. In Kim's published work [2], 129 DRAM modules fabricated between 2012 and 2013 were studied, discovering that 110 of these modules presented high vulnerability to row-hammering. Kim constructed a user level-program that flushes the cache-line while an specific address is accessed, proving the existence of disturbance errors based on a field programmable gate array (FPGA) platform.

A team of security analysts employed by Google (Project Zero) [5] studied the row hammering phenomenon in a x86 platform. In this case, Project Zero used a refined technique based on Kim's work, demonstrating that is possible to achieve privilege escalation either to escape from native client sandbox or to access all physical memory.

In literature another method of exploiting row hammer bug was exposed by Gruss [6]. This author proposes to induce remote attacks based on software using a website with JavaScript. Also in his research, Gruss achieves a fully automated attack that gains unrestricted access to website visitors systems.

Dealing with taking control of vulnerable computer, a recent publication [7] exposes the possibility of using videos and documents. This possibility has been discovered based on the previous publications concluding that a way to trigger row hammer attacks is implementing booby-trapped videos and documents. These alternatives enable to create bitflips in order to exploit buffer overflows and other software vulnerabilities. This discovery means that an attacker can manipulate a video to carry out the attack as shown the Fig. 1.

Originally, row hammer bug has been reported in DDR3 silicon since 2010. Because of the threat of this attack, companies tried to fix or mitigate the problem in this memory generation by proposing new solutions. Furthermore, computer industry affirmed that row hammer bug had been resolved in DDR4 memories. However, Lanteigne [1] studied many of the previous works in row hammering, introducing a test used as a brute force attack that could be applied to any computer system. The same writer found faults in DDR3 memories and also designed a test for DDR4 memories, finding flaws in 8 of 12 DRAM modules.

In regard to this matter, some mitigation strategies have been reported in the literature. Software-based strategies compose solutions such as increasing refresh rate or reducing the time to activate error correction algorithms. The first solution, where the refresh interval is generally 64ms, consist in halving this

978-1-5090-1095-0/16/$31.00 ©2016 IEEE

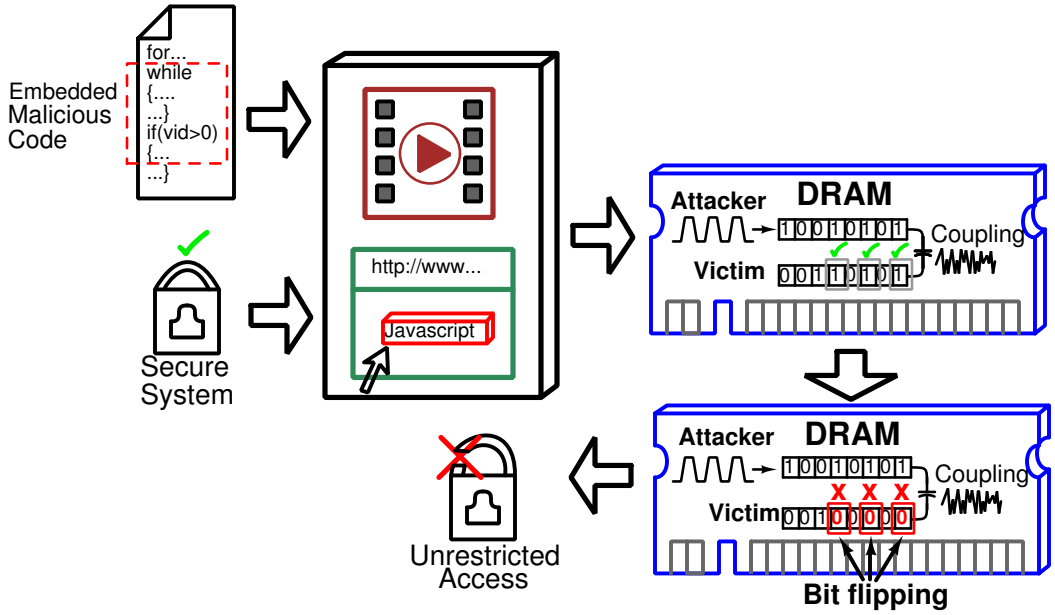

Fig. 1. Illustration of a DRAM Hammering attack

interval by the memory controller. In the latter solution, the errors in memories that include correction algorithms, varies with the error correction code (ECC) used. Moreover, the interval in which the errors are corrected, also affects the error rate, thus reducing the time interval can mitigate induced errors by the row hammer bug. However, these software strategies can induce a large performance overhead, reducing their effectiveness in practical applications.

Hardware strategies are also found in some works and are divided in two approaches: counter based and probabilistic row refresh. Counter based approach, such as target row refresh (TRR), and counter based row activation (CRA) [4], repose on counting the number of times a row (aggressor) is opened. When a threshold is achieved, a control signal indicates that some rows (victims) must be opened in order to refresh the stored charge. This threshold can be calculate based on the row hammering threshold (RH_{th}), which deals with the technology used and can be expressed as follows:

$$RH_{th} = \frac{\beta - 1}{\alpha - 1} \times M_{max} \quad (1)$$

where β is a factor used to guarantee an adequate refresh time, α is a constant that represents the number of times the leakage current of a memory cell is increased under row hammer attack, and M_{max} corresponds to the total possible number of activations in a refresh rate. Considering a refresh rate of 64ms, M_{max} can be 1.3 million. Furthermore, assuming $\alpha = 11$ (α can be in range from 4 to 11.7) and $\beta = 2$, the RH_{th} results in 130K which agrees with Kim's work [2]. However, Kim [4] states that for future technologies this threshold can be in the range of tens of thousands.

The main disadvantage of counting to activate a control signal is the high performance overhead included. To reduce this overhead, probabilistic approaches were proposed. Proposals such as pseudo TRR (pTRR), probabilistic row activation (PRA) [4] and probabilistic adjacent row activation (PARA) [2] based their operation on the activation of the victim rows with low probability each time an aggressor row is activated. The activation of an aggressor row launches a random generator and the resulting number determines if activation of victim rows occurs. Probabilistic operation avoids the need of storage information of the counters, reducing the hardware needed.

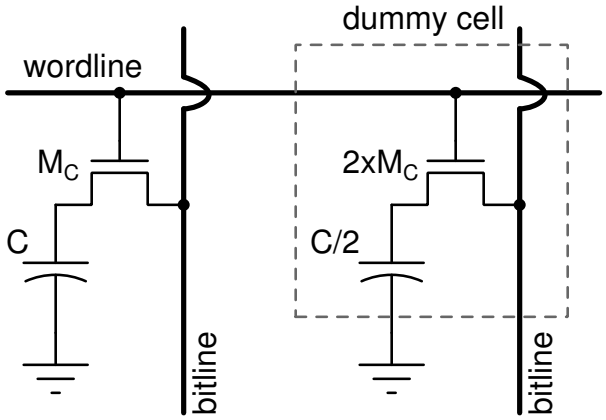

Fig. 2. Dummy cell with reduced capacitance and increased transistor size.

II. Dummy Cell Based Mitigating Strategy

Despite the reported work in the literature to mitigate row hammering, bit flipping is still present in modern DRAM circuits. This work proposes an alternative at circuit level to reduce the effectiveness of this attack.

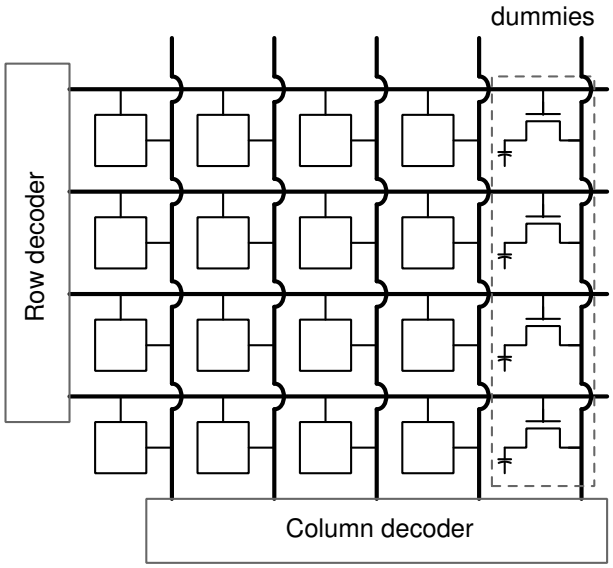

Fig. 3. Dummy cells within a DRAM array.

The proposed strategy consists in connecting a dummy cell to the wordline of the victim row, as Fig. 2 shows. The additional cell is similar to a standard memory cell but more susceptible to leakage. This characteristic can be achieved by implementing a transistor of twice its size regarding its nominal value, and a capacitor of a half its capacity. A wider transistor has a larger leakage current that, combined with the reduced storage capacity, results in a cell with a decreased retention time.

A cell with this particular attribute can be used as an indicator of a possible attack as follows: first, the dummy cell must be pre-charged to logic level one (1) during the refresh phase; in this case the dummy cell in the attacker wordline is ignored. Then, if a malicious memory access is carried-out, the dummy cell in the victim row will experiment a larger leakage current than a standard cell, thus its capacitor will be discharged with a increased rate. Therefore, the information stored in the dummy cell will be corrupted before the data of standard cells.

if the memory controller senses the output data of dummy cells, the respective victim row or near ones can be refreshed, when controller detects a zero logic. This new refresh avoids a possible bit-flipping, preventing data-corruption. In addition, memory controller can read the information stored in dummy cells at a higher rate than the refreshing speed, which demands a lower power consumption.

The Fig. 3 shows an alternative to include dummy cells in a DRAM memory array. For instance, the dummy cells should share the respective bitline reducing complexity in control algorithm as memory controller only have to monitor one column address. Nonetheless, dummy cells can be placed in a sparse way across the array, in order to avoid that an attacker can identify easily their position.

One important characteristic of the proposed alternative is the fact that it does not include additional complex hardware, because it uses only slightly-modified cells. Also, dummy cells can be placed in specific memory locations where critical information is stored, or randomly across the whole memory. It is important to highlight that the memory structure is not altered, then it is difficult for a hacker to find unusual hardware that arouses suspicions.

III. Results

A DRAM array was simulated in a 65nm CMOS standard technology. The purpose of this simulation was to prove the concept of accelerated discharge rate in a dummy cell. The block consisted of a 64×64 cells including a dummy cell per row similar to Fig. 3. The simulation application used a CMOS 65nm process, whose results could be extended to a state-of-art RAM dedicated process [8], [9]. One characteristic of a memory process is its low-leakage devices, implying a reduced current-capability and performance. It is important to highlight that a conventional 65nm process has a larger parasitic capacitance than a 28nm process, however its higher current density leads to an increment of the total performance.

As mention before, the main phenomenon that exploits a row hammer Attack is coupling between wordlines. Therefore, the simulated array included the RC coupling model proposed by [10]. In this manner, each connection between cells used a distributive model in each wordline. The value of coupling elements was obtained from technology files.

The Fig. 4 presents a simulation of the discharge process of a victim row for a standard memory and a dummy cell under process and temperature variations. The aggressor row is being accessed with a frequency of 1GHz and one of the adjacent rows is being monitored. The threshold for memory controller alert is set to $V_{DD}/2$. The results show that the dummy cell loses its charge with an increased rate, ensuring that capacitor voltage crosses the threshold before the standard cell one does. It means that the information stored in dummy cells is corrupted with enough time in advance, thus the memory controlled can refresh the row (or the whole array) preventing information loses.

The Fig. 5 was constructed based on the temperature variation. The Fig. 5 shows the voltage difference between capacitor voltage of dummy and memory cell under different process corners and a temperature range from -40 to 120 °C. In all cases, the voltage difference is greater than 260mV ensuring that dummy cell is corrupted faster than memory cell.

In order to validate the robustness of the proposed alternative, process variations were included in the simulation models for both coupling and interconnection resistance. These process simulations were used to construct a surface using the time difference of the discharge process. The Fig. 6 shows the surface for time difference including the variation of coupling capacitor and resistance. It can be seen that this time difference varies from 1µs to 5µs but in all cases the capacitor voltage of dummy cell crosses the threshold in less than memory cell. The range of time difference is enough to perform a refresh operation by memory controller. These figures validate the

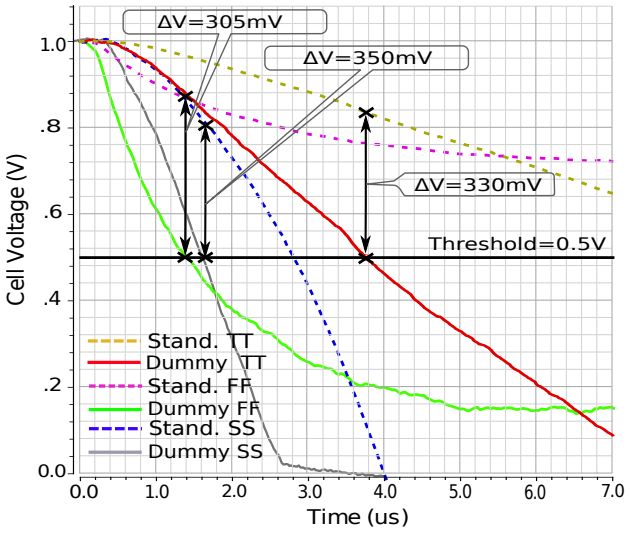

Fig. 4. Discharge process under the influence of coupling noised in a DRAM array.

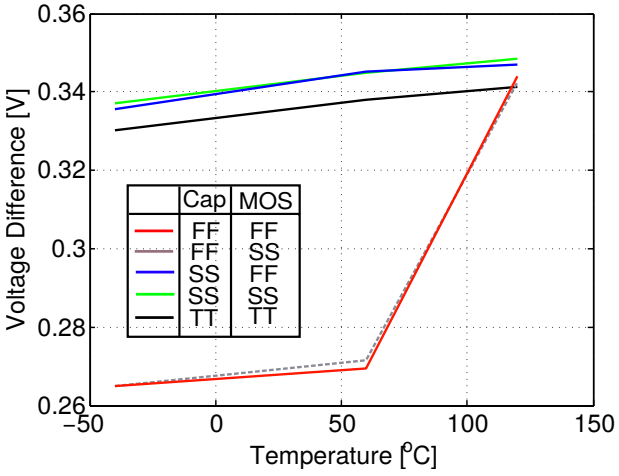

Fig. 5. Voltage difference between standard and dummy cell when the latter one crosses the threshold voltage.

concept of dummy cell as flag indicator for a Row Hammer attack in a DRAM array.

A highlight of the proposed strategy is the null impact on the regular memory operation due to possible generation of fake alerts. A fake alert could be generated if the dummy cell loses its charge during normal operation. Even though the leakage current of the dummy cells is higher, the conditions to discharge its storage capacitor faster than a standard cell are proper of a malicious attack; while in normal operation the memory refresh rate is smaller than capacitor discharge.

Another characteristic of this solution is the associate low-hardware overhead. A typical DDR4 memory has 512 bytes (4096 cells) per row [11]. So, if only a dummy cell is added to each row, an increment of less than 0.1% in the whole memory area will be achieved . Moreover, adding one dummy cell each 4096 makes difficult for a hacker to identify extra security.

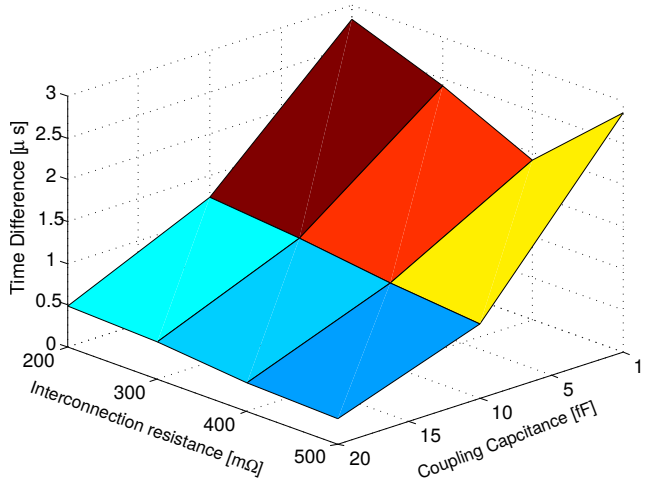

Fig. 6. Time difference between standard and dummy cell when both cross the threshold voltage.

IV. Summary

This document describes a strategy to increment chip security against row hammering attacks on DRAM. This kind of attack can be detected with the use of additional dummy-cells, whose data will be corrupted before information on the standard memory cells. The main benefit of this solution is the additional low-complexity circuits, adding only 0.1% of extra area. Also, a variety of distributions of dummy cells across the memory can be used, in order to avoid hacker identification.

References

[1] M. Lanteigne, " How Row-Hammer Could Be Used to Exploit Weaknesses in Computer Hardware," 2016.

[2] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bBts in Memory without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, June 2014, pp. 361–372.

[3] Y. Kim, "Architectural Techniques to Enhance DRAM Scaling," Ph.D. dissertation, Carnegie Mellon University, 2015.

[4] D. H. Kim, P. J. Nair, and M. K. Qureshi, "Architectural Support for Mitigating Row Hammering in DRAM Memories," *IEEE Computer Architecture Letters*, vol. 14, no. 1, pp. 9–12, Jan 2015.

[5] P. Zero, "Exploiting the DRAM Row-Hammer Bug to Gain Kernel Privileges." [Online]. Available: http://googleprojectzero.blogspot.com.co/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

[6] D. Gruss, C. Maurice, and S. Mangard, "Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript," in *arXiv*, 2016.

[7] D. Goodin, "DRAM Bitflipping Exploits that Hijack Computers just Got Easier." [Online]. Available: http://arstechnica.com/security/2016/04/dram-bitflipping-exploits-that-hijack-computers-just-got-easier/

[8] K. C. H. et. al, in *Electron Devices Meeting (IEDM), 2011 IEEE International*.

[9] S. K. h. Fung et. al, "65nm SOI CMOS Technology for High Performance Microprocessor Application," in *2006 International Symposium on VLSI Technology, Systems, and Applications*, April 2006, pp. 1–2.

[10] P. Heydari and M. Pedram, "Analysis and reduction of capacitive coupling noise in high-speed VLSI circuits," in *Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Conference on*, 2001, pp. 104–109.

[11] D. Wang, "Why migrate to DDR4?" [Online]. Available: http://www.eetimes.com/document.asp