Alert Refresh System for Mitigating RowHammer

Nayeon Kim, Kwangrae Kim, Ki-Seok Chung*
Dept. of Electronic Engineering
Hanyang University
Seoul. Republic of Korea
wingedna@hanyang.ac.kr, kksilver91@hanyang.ac.kr, kchung@hanyang.ac.kr*

2023 20th International SoC Design Conference (ISOCC) | 979-8-3503-2703-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISOCC59558.2023.10396283

Abstract— **Modern DRAM cells are known to be vulnerable to RowHammer attacks because they are sensitive to electrical interference between adjacent rows. Most existing approaches to mitigate Rowhammer require complicated hardware structures. This paper proposes an effective protection scheme called Alert Refresh System. The proposed scheme has a low implementation overhead because it utilizes a pin called ALERT_n, which is a part of the standard DRAM interface. In addition, it applies hardware-friendly algorithms to track complicated RowHammer attacks. Experimental results show that the proposed method achieves a high average RowHammer protection rate of 99.81% against maliciously crafted RowHammer attacks, which is 16.94% higher than PRoHIT, one of the state-of-the-art probabilistic RowHammer mitigation methods.**

Keywords; RowHammer, Security, DRAM

I. Introduction

RowHammer is a security attack that induces undesirable bit flips in some adjacent rows by intensively accessing a small set of rows. Various studies have proposed methods to protect DRAM cells from the RowHammer attack. The number of intensive accesses to a specific row that may cause an undesirable bit flip is called the *RowHammer threshold (RH threshold)*. Rows accessed frequently are called *aggressor rows*, and adjacent rows whose cell values may change are called *victim rows*. One practical concern with existing approaches is that most approaches require some modification of the DRAM memory controller and DRAM PHY interface, which is not realistically possible. Furthermore, in most schemes, the memory controller is supposed to transfer information on the victim rows to be refreshed to DRAM. However, the memory controller does not know the exact physical address of the DRAM, and even if it were possible, it would cause performance degradation.

In this paper, we propose an effective method called an Alert Refresh System (ARS) that protects DRAM cells from RowHammer without requiring modification of the DRAM interface or the memory controller. ARS uses ALERT_n pin, one of the interfaces specified in the DDR4 standard, as a key means of alerting the RowHammer threat. ARS mitigates RowHammer by employing a probability-based method using the Misra-Gries algorithm [1]. In ARS, we modify the reset policy of the Misra-Gries algorithm in order to efficiently track complicated RowHammer attack patterns such as double-sided attacks.

II. Proposed Alert Refresh System

ARS has two operation modes: normal and Alert Refresh modes. In the normal mode, the rows in the DRAM cell arrays are sequentially refreshed by Auto Refresh. On the other hand, in the Alert Refresh mode, only the victim rows are additionally refreshed. Fig. 1 describes the ARS architecture. The ACTIVE Count module determines the operation mode based on the count of the issued ACTIVE commands. The Alert_n pin can be used to enable alert refresh mode and change the state in the DRAM mode register. One of the key components of ARS is the Misra-Gries module which efficiently determines aggressor candidates among the accessed rows. The Misra-Gries module decides the 1st aggressor row and the 2nd aggressor row and transfers information to a module called Victim Row Decision. The Victim Row Decision module regards the rows on both sides of the two aggressor rows as victim rows, and they will be refreshed.

Fig. 2 illustrates an example to show how the Misra-Gries module tracks a candidate set of aggressor rows. The Misra-Gries table stores the information on the frequently accessed rows. A register called Spill Counter is added to determine rows that have not been frequently accessed. When access to a new row whose information is not included in the table is issued, we need to update the table by adding the information of the newly accessed row. The Spill Counter value is used to determine the row that the newly accessed row will replace. If there is an entry whose access count equals the Spill Counter value, the corresponding entry will be replaced by the newly accessed row. When a new entry is added, the initial access count of the newly added entry is set to one bigger than the Spill Counter value. Fig. 2 (a) shows how a newly accessed row (0x50000) replaces an entry (0x1000) whose access count is equal to the

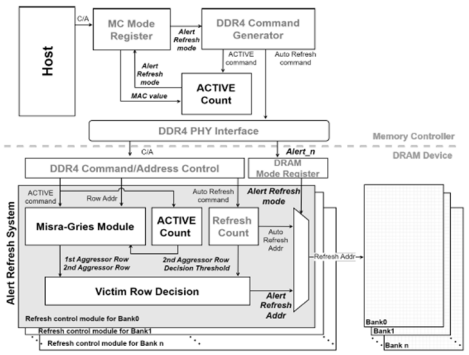

Fig. 1 Architecture of ARS

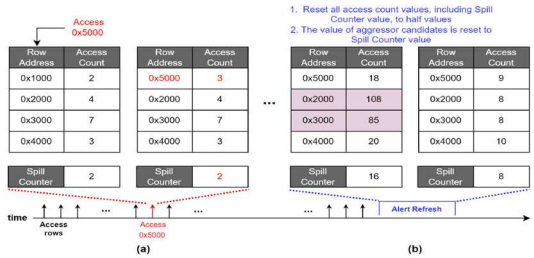

Fig. 2 How the Misra-Gries table is managed (the number of table entries is four in this example) (a) an example of the case where a new row is accessed (b) an example of updating the table after two aggressor rows are determined.

Spill Counter value and how the initial access count of the newly added entry is set. This policy makes it possible to track a set of frequently accessed rows. The details of how the Misra-Gries table works can be found in [1].

Fig. 2 (b) describes how the table updates after the two aggressors are determined. Suppose two addresses (0x2000, 0x3000) are selected as the aggressor rows. Since the victim rows of the selected aggressor rows will be refreshed, we need to reset the access count values of the rows in the table and the Spill Counter value. The reset policy of the Misra-Gries table after an Alert Refresh is issued is as follows: 1) All the access counts in the Misra-Gries table, including the Spill Counter value, are reset to half of the current access count value. This is to avoid the situation where unnecessarily many refreshes are issued if all the access counts are monotonically increasing. 2) The access counts of the aggressor rows are reset to the new Spill Counter value because victims of the aggressor rows are supposed to be refreshed recently. This policy to update the access account after an Alert Refresh is issued differs from the original Misra-Gries algorithm, and experimental results show that it works better.

Fig. 3 Comparison between the proposed scheme and prior works for RowHammer mitigation performance. They are evaluated with the RowHammer attack patterns while increasing the number of aggressor rows from 2 by 2 to 20.

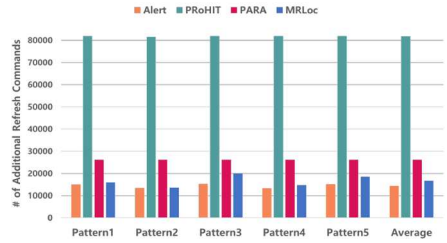

Fig. 4 Comparison between the proposed scheme and prior works for the number of additional refresh commands.

III. Evaluations

We use DRAMSim2 [2] with some modifications to implement both the proposed method and some existing works [4], [5], [6] for comparison. We measure RowHammer mitigation performance by self-made RowHammer attack patterns made with reference to [3]. The RowHammer reduction ratio is the ratio between the numbers of RowHammer occurrences with and without the application of the proposed RowHammer mitigation scheme when self-made attack patterns are applied to DDR4. Fig 3 shows the comparison results of ARS when compared with existing probabilistic methods in terms of the mitigation rate for RowHammer. PARA [4] and MRLoc [5] barely protect the DRAM cells against all RowHammer attack patterns. PRoHIT [6] fails to protect against complicated attack patterns (i.e., Pattern 3, 4, and 5). On the other hand, the proposed work almost perfectly protects the DRAM cells against all RowHammer attack patterns. On average, ARS's RowHammer reduction ratio is 99.81%, which is 16.94% higher than that of PRoHIT. As described in Fig. 4, ARS issues fewer additional refresh commands than others. (18% compared to PRoHIT).

IV. Conclusion

In this paper, we propose a novel RowHammer mitigating scheme called Alert Refresh System (ARS). ARS does not require any change in the standard DRAM interface or the memory controller by utilizing the ALERT_n pin. Experimental results show that the proposed method achieves the highest average RowHammer reduction ratio of 99.81% among all the compared probability-based RowHammer mitigation schemes.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00153, Development of O-RU for in-building based on AI network management using beamforming path)

References

[1] Y. Park, W. Kwon, E. Lee, T. J. Ham, J. Ho Ahn and J. W. Lee, "Graphene: Strong yet Lightweight Row Hammer Protection," *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Athens, Greece, 2020, pp. 1-13

[2] Rosenfeld, Paul, Elliott Cooper-Balis, and Bruce Jacob, "DRAMSim2: A cycle accurate memory system simulator." *IEEE computer architecture letters* 10.1 (2011): 16-19.

[3] K. Kim, J. Woo, J. Kim and K. -S. Chung, "HammerFilter: Robust Protection and Low Hardware Overhead Method for RowHammer," 2021 IEEE 39th International Conference on Computer Design (ICCD), Storrs, CT, USA, 2021, pp. 212-219

[4] Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J. H., Lee, D, & Mutlu, O, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors." ACM SIGARCH Computer Architecture News 42.3 (2014): 361-372.

[5] Jung Min You, and Joon-Sung Yang, "MRLoc: Mitigating Row-hammering based on memory Locality." Proceedings of the 56th Annual Design Automation Conference 2019. 2019.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[6] Mungyu Son, Hyunsun Park, J. Ahn and Sungjoo Yoo, "Making DRAM stronger against row hammering," *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, 2017, pp. 1-6