Protect Non-volatile Memory from Wear-out Attack based on Timing Difference of Row Buffer Hit/Miss

Summary

Submitted by: Sharanya Kamath (16CO140)

Non-volatile Memories (NVMs), such as PCM and ReRAM, have been widely proposed for future main memory design because of their low standby power, high storage density, fast access speed. However, these NVMs suffer from the write endurance problem. In order to prevent a malicious program from wearing out NVMs deliberately, researchers have proposed various wear-leveling methods, which remap logical addresses to physical addresses randomly and dynamically. However, the paper discovers that side channel leakage based on NVM row buffer hit information can reveal details of address remappings. Consequently, it can be leveraged to side-step the wear-leveling. The simulation done by the authors shows that the proposed attack method in this paper can wear out a NVM within 137 seconds, even with the protection of state-of-the-art wear-leveling schemes. To counteract this attack, the paper further introduces an effective countermeasure named Intra-Row Swap (IRS) to hide the wear leveling details. The basic idea is to enable an additional intra row block swap when a new logical address is remapped to the memory row. Experiments demonstrate that IRS can secure NVMs with negligible timing/energy overhead, compared with previous works.

Row buffer hit time attack is a 4 step process:

- Identify LAs in the same row as that of target LA. This can be done by observing their latencies. All LAs that have buffer hit latency belong to the same row.
- Attack continuously on target LA until swapping occurs. When swapping happens there are 3 possibilities, first it can happen outside the attack set, second LA other than target is swapped out, target LA is swapped out (with some LA outside attack set).
- For the first two possibilities target LA does not change, but for the third attack set is updated to incorporate the swapped in LA and it is made the target.
- Repeat the above steps until the PA location of the target LA is weared out.

To counteract the attack, the authors propose a mechanism called intra row swap.
The scheme enforces a compulsory swap after the block swap of original wear leveling. To implement this, an intra row swap vector storing M/2 bits (M=Number of Row Buffer Blocks in NVM). The bits are reversed after two blocks are swapped.
This swapping prevents the attacker from knowing about the new logical address to write to after the last one was stopped.

Questions:

1. What will be the behavior of the approach on other types of attacks?
2. What is the computational overhead caused by the intra row swap?