Electronic Design

By STEVEN WOO, Fellow and Distinguished Inventor, Rambus

Understanding Memory's RowHammer Challenge

The DRAM industry has long been plagued by the security risk of RowHammering, and mitigation techniques have done little to stop attackers. But a new approach to remapping DRAM addresses could be a novel solution to this security challenge for memory.

Rapid growth in the world's digital information has driven continued improvements in computing to process, organize, and monetize insights gathered from this data. The semiconductor industry has enabled these advances with smaller process geometries that deliver more transistors per die, faster processing, and more memory to store the data being used. However, as processing gets faster and devices shrink, the hardware has become increasingly susceptible to errors.

Each new generation of DRAM offers many advantages, including higher performance, better power efficiency, and higher capacity, which has led to it becoming entrenched as the choice for main memory. Packing bit cells more closely together on a die enables the industry to develop higher capacities, but it's also introduced and exacerbated a vulnerability and attack vector for malicious actors to exploit.

This DRAM vulnerability, known as RowHammer, takes advantage of the increasingly closer proximity of bit cells in the memory array. When bits are repeatedly accessed, neighboring bit cells can flip, causing their values to unknowingly change.

RowHammer has been a concern for many years, with academic studies demonstrating this phenomenon on DDR3 DRAMs since 2014 and showing that newer generations have only become more susceptible to such attacks. At smaller process geometries, it becomes easier for bits to flip due to both normal operations and malicious attacks. Researchers have proposed and implemented defense measures, such as Target Row Refresh (TRR), but many such mechanisms have been reverse-engineered and bypassed with targeted attacks.

Solutions that address RowHammer must not only be difficult to circumvent, but they also need to balance area, cost, and performance impact to be useful for future memory systems.

How RowHammer Attacks Work

A RowHammer attack occurs when a row in the DRAM bank is repeatedly activated, leading to bit flips in neighboring rows. An unintended consequence of the DRAM architecture and smaller process geometries is that neighboring bit cells can interact electrically, causing charge leaks and changing the contents of cells that aren't being accessed.

The values in these neighboring bit cells can flip from 0 to 1 or vice versa, leading to silent data corruption that may result in uncorrectable or undetectable errors. Bad actors can exploit RowHammer to trigger many errors that overwhelm a system and cause denial-of-service (DoS), as well as gain access to data in privileged areas of the system.

Traditional attacks focus on activating (or hammering) an "aggressor" row in a DRAM bank. The repeated activation aims to cause bit flips in neighboring rows before their data is refreshed.

Since discovering this vulnerability, DRAM makers have implemented multiple solutions to disrupt RowHammer attacks, including targeted memory refreshes for the affected bit cells. While refreshing these "victim" rows can restore charge and counteract the effects of a RowHammer attack, researchers have demonstrated that frequent refresh operations can be leveraged for a different and newer type of RowHammer attack.

In 2021, Google shared the discovery of the Half-double, or victim-focus mitigation, attack. This technique takes advantage of these refreshes by repeatedly triggering mechanisms to create a domino effect. Repeated activations of an aggressor row R will lead to multiple refreshes to the neighboring rows R-1 and R+1 to restore charge and avoid bit flips. However, these refresh operations act as activations to rows R-1 and R+1, resulting in a potential downstream effect on their neighboring rows R-2 and R+2, leading to

flipped bits in these rows.

Taking advantage of the growing restrictions and worsening physics on new generations of DRAM chips, researchers used the mechanisms put in place to mitigate attacks on row R to conduct a new type of attack on rows R-2 and R+2, adding another layer of difficulty in mitigating the problem *(Fig. 1)*.

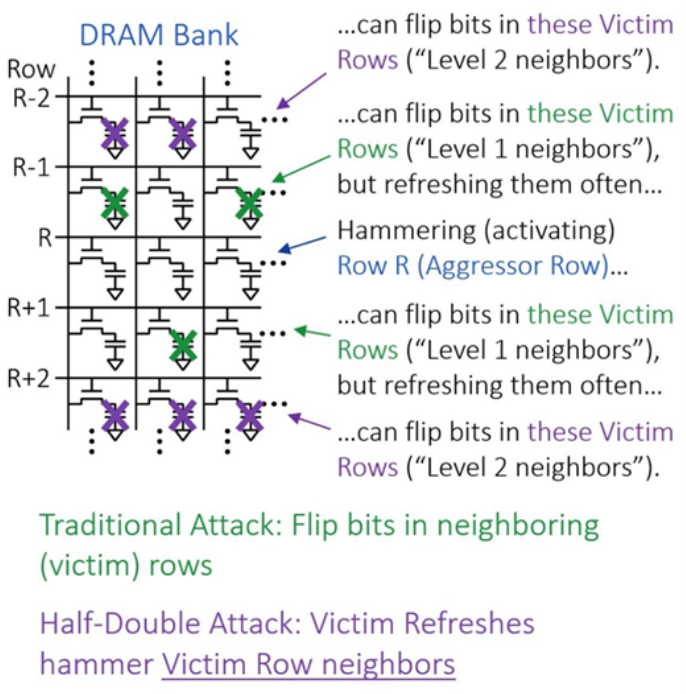
A Growing Concern for Memory Security

The RowHammer problem is especially concerning for data centers that can house tens of thousands of servers. Errors can go undetected when successful attacks flip bits and overwhelm the detection and correction capabilities of error correction codes (ECC). This causes a program to perform calculations on incorrect data that result in false conclusions, execution of the wrong instruction, denial-of-service, or access to protected data.

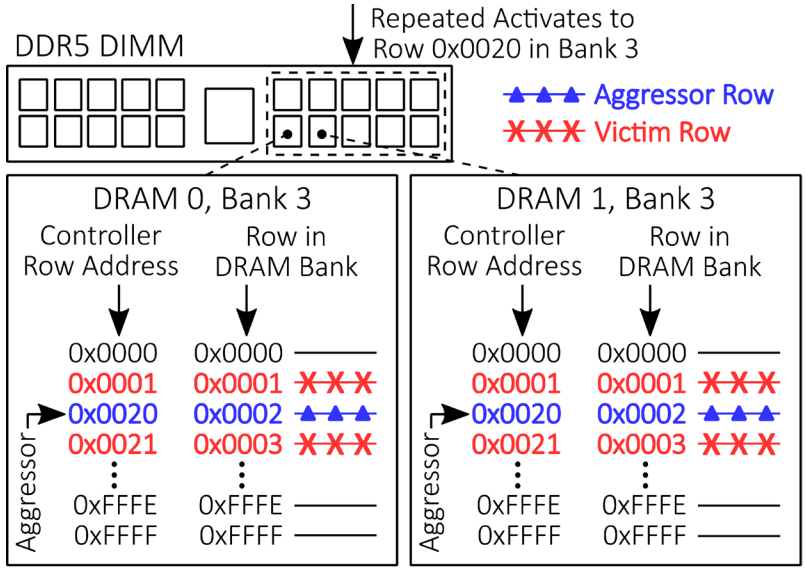
As processing geometries become smaller, the hammer count (HC, number of activations to cause bit flips in victim rows) falls and the number of neighboring victim rows affected by repeated activations to an aggressor row rises. The hammer count has fallen by more than a factor of 10 over the past decade and will continue to fall at more advanced process nodes. Extrapolations from recent data show that HC can fall to as low as 1K-3K in future process nodes, a steep decline from the 50K in previous studies.

In modern servers, dual in-line memory modules (DIMMs) are used to store data. When a processor accesses memory, multiple DRAMs on the DIMM respond, each of which holds a portion of the requested data. A RowHammer attack in a server is especially concerning because it can flip bits in multiple DRAMs on a DIMM, making it even harder to detect and correct errors due to bit flips.

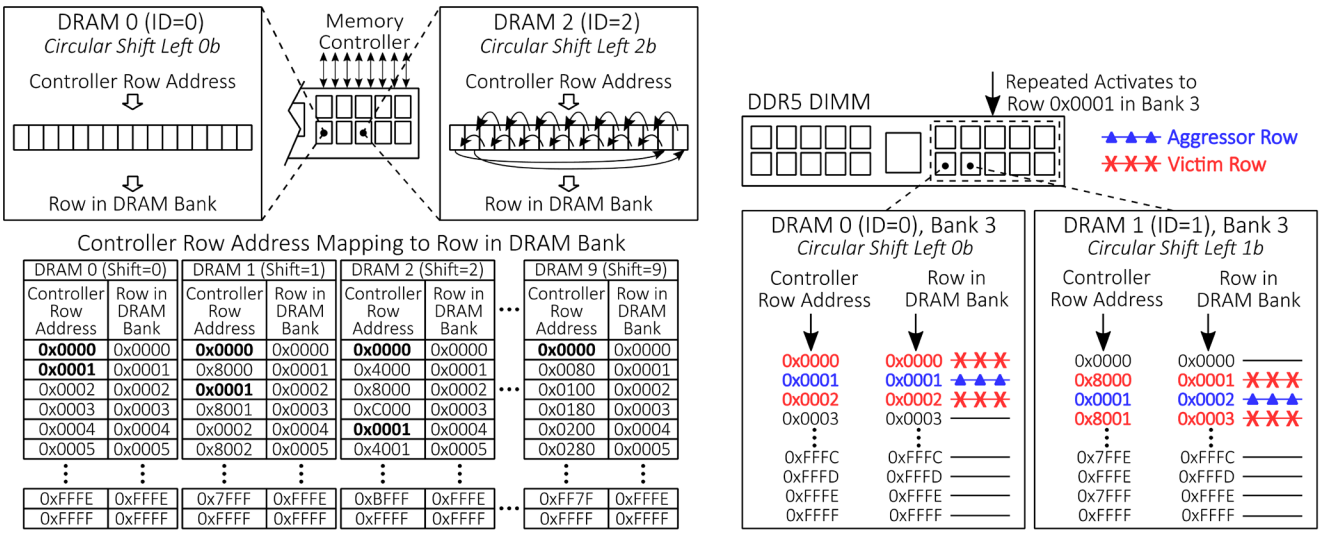
The fundamental issue is that all DRAMs on the DIMM map addresses from the controller to rows inside the



1. Attacking a single row by "Hammering" it with activates can impact the neighboring rows. A RowHammer attack on Row R can cause bit flips in Rows R-1 and R+1. Refreshing Rows R-1 and R+1 can restore charge that's leaking away because of hammering Row R, but these refreshes can act as hammers on their neighboring Rows R-2 and R+2.



2. RowHammer attacks on a DDR5 DIMM can impact multiple DRAMs due to the identical mapping of controller addresses to internal rows in the DRAMs. Hammering controller row address 0x0020 will impact controller row addresses 0x0001 and 0x0021 in each DRAM, potentially leading to many simultaneous errors that can overwhelm the ECC correction capabilities of the DRAMs and memory system.



3. RAMPART's simple shifting of the Row Address Mapping has a profound impact on mitigating RowHammer attacks when different rows (0x0000 and 0x0002 vs 0x8000 and 0x8002 in this example) are impacted by an attack on Controller Row Address 0x0001. This allows for standard error-correction tools to be leveraged to effectively mitigate the effects of one successful RowHammer attack.

DRAM in the same way. A successful attack on one DRAM address makes all other DRAMs vulnerable to the same attack at the same victim row address, because all DRAMs have the same neighboring addresses *(Fig. 2)*.

Malicious actors can repeatedly target the same address across multiple DRAMs on the same DIMM and overwhelm the error-correction mechanisms in the DRAM and memory system.

Addressing RowHammer with Available Tools

Although the possibility of a RowHammer attack can't be fully eliminated, we're able to build on existing tools to improve resistance to these attacks. We next describe a method that requires two successful attacks to cause irreparable damage to the data, drastically improving DRAM and memory system reliability.

The method relies on enabling each DRAM to have a unique mapping of controller row addresses to internal DRAM rows, preventing addresses from being neighbors in more than one DRAM. By enabling unique address mappings, which result in unique neighboring row addresses in each DRAM, we can limit bit flips from a successful attack to a single DRAM in the rank for any given victim row address. Chipkill ECC can then correct errors limited to a single DRAM, making it possible for the system to continue running while reversing the effects of the attack.

Implementing this unique address mapping per DRAM, which we call RAMPART (Row Address Map Permutation and Reassignment Technique), improves resistance to RowHammer attacks. It requires two successful attacks against the same victim row address in different DRAMs to cause irreparable damage to the data, even if all bits in the neighboring victim rows flip.

RAMPART takes the address from the memory controller and circularly shifts it left by a different number of bits equal to the unique ID assigned to each DRAM, ensuring that every address has unique neighbors. Bit flips in any victim row address from a successful attack will be confined to one DRAM, enabling Single Device Data Correction (also referred to as chipkill) ECC to correct these bit flips, reversing any damage caused by one successful RowHammer attack *(Fig. 3)*.

Building Better System Resistance to Data Corruption

Memory performance and capacity will continue to be a focus for the industry as demanding applications like AI become more prevalent. DRAM designs will evolve to meet the new demands of more challenging future workloads. New designs, which will increase DRAM density by placing bit cells closer together in future generations, will be increasingly susceptible to RowHammer attacks, motivating the development of new mitigation and prevention techniques.

Modeling work shows that RAMPART, in combination with existing error-correction mechanisms, offers 4 to 17 orders of magnitude better system resistance to data corruption over a year of continuous attacks for a range of parameters representative of next-generation systems. RAMPART is simple to integrate into DRAM designs, has no performance impact, lowers the chances of data being corrupted by a RowHammer attack, and is compatible with many existing tracking and probabilistic selection techniques.

To learn more about RAMPART, check out the white paper, "RAMPART: RowHammer Mitigation and Repair for Server Memory Systems."