DDR Memory Errors Caused by Row Hammer

Barbara P. Aichinger
Vice President New Business Development

FuturePlus Systems Corporation
15 Constitution Drive
Bedford NH 03110 USA
Outline

• What is Row Hammer?
• What Research has been done?
  – CMU
  – Google Project Zero
  – Java Script
  – Third IO
• ECC
• Mitigation Strategies
• Software that creates Row Hammer
• Summary
What is Row Hammer?

- Disturbance Errors: Row to Row Coupling

Excessive ACTIVATE commands apply repeated charge to the memory cells.

Electromagnetic field induced by applied voltage.

Cells lose charge by repeated nearby electromagnetic field, causing a coupled bit.

Source: https://www.eurosoft-uk.com/eurosoft-test-bulletin-testing-row-hammer/
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim¹  Ross Daly*  Jeremie Kim¹  Chris Fallin*  Ji Hye Lee¹  Donghyuk Lee¹  Chris Wilkerson²  Konrad Lai  Onur Mutlu¹

¹Carnegie Mellon University  ²Intel Labs

July 2014

```
loop:
  mov  (X), %eax
  mov  (Y), %ebx
  clflush  (X)
  clflush  (Y)
  mfence
  jmp  loop
```

```
X →
0011101111
1111 . 1111
101111101
110001011
1100011101

Y →
011011110
1111 . 1111
```

0 = Disturbed bit
Security Implications
Source: CMU: Flipping bits in Memory without accessing them

• Breach of memory protection
  – OS page (4KB) fits inside DRAM row (8KB)
  – Adjacent DRAM row is a different OS page

• Vulnerability: disturbance attack
  – By accessing its own page a program can corrupt pages belonging to another program
CMU then induced errors with an FPGA based system.
Results

1. Most Modules Are at Risk

- **A company**: 86% (37/43) errors up to $1.0 \times 10^7$
- **B company**: 83% (45/54) errors up to $2.7 \times 10^6$
- **C company**: 88% (28/32) errors up to $3.3 \times 10^5$

How many errors did CMU find?

Figure 3. Normalized number of errors vs. manufacture date

CMU Study Summary

• Temperature did not play a role in bit flips

• “Disturbance errors are widespread in DRAM chips sold and used today”

• “Due to difficulties in DRAM scaling, new and unexpected types of failures may appear”

• Recommends a mitigation strategy of ACT or REF adjacent rows when accesses are made
  – Requires changes to memory controllers
  – Knowledge of DRAM layout
Google: Project Zero
Exploiting the DRAM rowhammer bug to gain kernel privileges

Mark Seaborn and Thomas Dullien

• March 2015
• Demonstrated using the Row Hammer failures as an exploit to gain kernel privileges
• Used CLFLUSH instruction
Double Sided Hammering

• Increases bit flips in row n by hammering row n+1 and n-1
• Produced failures much faster
  – One machine had 25 bit flips in a single row using this technique
• Need to understand the physical geometry
  – Need to know physically adjacent row addresses
  – Was able to figure this out by hammering rows 256K below and above and observing increased bit flips
Row Hammer Exploit #1

- Native Client Sandbox in Google Chrome escape
  - Uses the CLFLUSH instruction
  - Escape from the sandbox
  - Exploit works by triggering bit flips in the indirect jump instructions
Row Hammer Exploit #2

- Page Table Entry method
  - Use Row Hammer to flip bits in PTEs
  - Causes the PTE to point to another PTE
    - To increase probability of jumping to attacking code
      - Find bits that have failed and use them since they most likely will fail again
      - Spray Memory with page tables so if you jump you will most likely hit a PTE that points to attacking code
Bit Flips in Laptops

• Google tested 29 laptops (without ECC)
  – 8 different models
  – 5 different memory vendors
  – Laptops ranged in date from 2010 to 2014
  – 15 of 29 laptops showed bit flips
Project Zero Summary

• Many bugs that appear to be difficult to exploit have turned out to be exploitable
  – The poisoned NUL byte, 2014 edition
    • a off-by-one NUL byte overwrite could be exploited to gain root privileges from a normal user account
Project Zero Summary

• Recommends disallowing the CLFLUSH for use in unprivileged code
  – Was removed from Google Chrome Native Client

• Points out several less likely methods that might succeed even without CLFLUSH

• Points at the need for more research especially using a JavaScript….this proves to be fortuitous…..
• Replaces CLFLUSH with a cache eviction strategy that gives a 99.99% successful eviction rate
• First remote software-induced hardware fault attack
  – Does not require physical access to the machine
  – Does not use native code or special instructions
• Via a web page can be performed on millions of users simultaneously without their knowledge
• Demonstration used Firefox v39 on a Linux machine
Eviction Strategy

• Replaces the CLFLUSH instruction
• Uses large pages
  – Developed a tool to convert JavaScript array indices to physical addresses
  – This helps find the weak locations where the bit flips can be exploited
• Verified its eviction strategy for Sandy Bridge, Ivy Bridge and Haswell CPUs
Results

• JavaScript code performed as well as native code, but not as well as CLFLUSH
  – Haswell
    • No success unless the refresh rate was lowered
  – Ivy Bridge laptop
    • Produced significant bit flips
Row Hammer Failures in Servers

• Third IO Mark Lanteigne
  – Memesis: A enterprise memory test
    • It is Linux Kernel Embedded, Lower Overhead and Is Closer to Hardware
    • Uses the e820 memory map and ACPI NUMA for precise memory targeting
    • HPC Parallel Processing – uses the full power of all CPU cores
    • Provides higher DRAM bandwidth versus the Stream benchmark (verified)
Row Hammer Failures resulting in Machine Checks

```
rd10: 3 10:56:59 Third10 kernel: Memesis cmdline: numa=0 cm=1 test=31 hits=5000000 aux=1 pat
rd10: 3 10:57:39 Third10 kernel: Memesis cmdline: numa=0 cm=1 test=31 hits=5000000 aux=1 pat=3
rd10: 3 10:57:40 Third10 kernel: cpumap
rd10: 3 10:57:41 Third10 kernel: TEST #31 Starting aux
rd10: 3 10:59:49 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:00:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:01:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:02:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:03:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:04:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:05:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:06:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:07:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:08:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:09:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:10:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:11:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:12:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:13:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:14:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:15:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:16:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:17:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:18:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:19:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:20:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:21:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:22:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:23:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:24:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:25:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:26:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:27:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:28:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:29:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:30:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:31:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:32:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:33:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:34:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:35:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:36:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:37:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:38:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:39:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:40:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:41:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:42:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:43:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:44:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:45:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
rd10: 3 11:46:31 Third10 kernel: mce: [Hardware Error]: Machine check events logged
```

```
ECC Systems are Vulnerable

- Third I/O often encounters ECC protected servers that can be extremely vulnerable to Row Hammer
- Even after 2X refresh mitigation in place
- ECC errors (with thresholding, means hundreds of errors before first reporting)
- CMCI Storms (too many ECC errors reported)
  - ECC is broken! No reporting standards
- Performance problems, reboots, lockups, halts
ThirdIO Contact Information

• Mark Lanteigne
• lant@thirdio.com
• (512) 422-4254
Row Hammer Software

- Passmark MemTest86 Test 13
- Github: Mark Seaborn’s code
  - https://github.com/google/rowhammer-test
- Github: CMU code
  - https://github.com/CMU-SAFARI/rowhammer
- Github: Rowhammer.js
  - https://github.com/IAIK/rowhammerjs
- ThirdIO: Memesis for Servers
Testing the system for excessive ACTIVATE commands

• Repurposed our DDR Detective® Protocol Analyzer to count ACT commands to unique Row Addresses (Row Hammer feature)
Running the Google code
ECC helps but will not prevent undetected data corruption

- **Error Correction Codes** on DDR3 are Single Error Detection and Correction and Double Error Detection
- Research showed more than 2 bits on a single 64 bit access
  - However the rate of failures was much less
- Multibit errors will not be detected or erroneously flagged as SEDC
Mitigation Strategies

- **Row Activate Counters**: Counts Activates to Rows and issues dummy ACT to neighboring Rows
  - Requires significant changes to the memory controller/DRAM
- **Probabilistic Row Activation (CMU)**: Memory controller issues dummy ACT commands to neighboring Rows
  - Requires changes to the memory controller and knowledge of the DRAM layout
- **Targeted Row Refresh**
  - Requires special DRAM and changes to memory controller
- **Double the Refresh Rate**
  - Best Solution for existing hardware: Performance and power penalty
Row Hammer failures on DDR4?

Rowhammer mitigation

My i7-5820K/GA-X99-UD4/2400MHz Crucial Ballistix DDR4 system was failing rowhammer (a few hundred errors per pass) until I reduced the refresh interval timing from the default of 7.8ms, in spite of the fact that DDR4 is supposed to include rowhammer mitigation (source: https://en.wikipedia.org/wiki/Row_hammer#Mitigation)

In my board's BIOS, the two settings were tREFI (default of 9360) and tREFIX9 (default of 82).

refresh interval (ms) = tREFI / (RAM clock (MHz) / 2)

\[ tREFIX9 = \frac{8.9 \times tREFI}{1024} \]

so...

9360/(2400/2)=7.8ms

(Source: page 123 of http://www.intel.com/content/dam/www....-datasheet.pdf)

The standard recommendation is to reduce the refresh interval to 3.9ms and thereby double the refresh rate (source: http://support.lenovo.com/us/en/produ...ity/row_hammer). Doing that gave me one error per pass at the same address both times, so I reduced the interval to 75% of 3.9ms (i.e. tREFI=3510, tREFIX9=31) and it's now error free over 8 passes overnight.
Summary

• DDR3 Memory is everywhere!

• Critical Applications need to be aware of this issue
  – It's both a reliability issue and a security issue

• ECC protected memory should be used but is not a fix for this problem
Contact Information

Barbara P. Aichinger
FuturePlus Systems
Barb.Aichinger@FuturePlus.com
603-472-5905
www.FuturePlus.com
www.DDRDetective.com