HiRA: Hidden Row Activation
for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

Abdullah Giray Yağlıkçı
Ataberk Olgun   Minesh Patel   Haocong Luo   Hasan Hassan
Lois Orosa   Oğuz Ergin   Onur Mutlu

SAFARI

ETH Zürich   CESGA   TOBB ETÜ
Executive Summary

**Problem:** DRAM Refresh
- is a **fundamental operation** to avoid bit flips due to **leakage** and **RowHammer**
- incurs **increasingly large performance overhead** with DRAM chip **density scaling**

**Goal:** Reduce the **performance overhead** of DRAM Refresh

**Key Idea:** Hide **refresh latency** by **refreshing** a DRAM row **concurrently with activating** another row in a **different subarray** of the **same bank**

**HiRA:** Hidden Row Activation – a new DRAM operation that
- Issues **DRAM commands** in **quick succession** to concurrently open two rows in **different subarrays**
- Works on **real off-the-shelf DRAM chips** by violating timing constraints
- **Significantly reduces** (51.4%) the time spent for refresh operations

**HiRA-MC:** HiRA Memory Controller – a new mechanism
- Leverages **HiRA** to perform **refresh requests concurrently with DRAM accesses and other refresh requests**
- **Significantly improves** system performance by **hiding refresh latency** for both **regular periodic** and **RowHammer-preventive** refreshes
DRAM Organization

DRAM Chip

DRAM Bank

DRAM Subarray

Chip I/O

Bank

Subarray

Bitline

DRAM Cell

Wordline

DRAM Row

Row Buffer

SAFARI
**DRAM Operations**

1. **ACTIVATE (ACT):**
   - Fetch the row’s content into the **row buffer**

2. **Column Access (RD/WR):**
   - Read/Write the target column and drive to I/O

3. **PRECHARGE (PRE):**
   - Prepare the array for a new ACTIVATE
DRAM Refresh

DRAM Refresh **is the key maintenance operation** to **avoid bit flips** due to charge leakage.

DRAM Refresh **activates** a row and **precharges** the bank.

**Problem:** DRAM Refresh **blocks** accesses to the **whole bank / rank**.

---

**DRAM Subarray**

- **Row Buffer**
- **DRAM Refresh**
- **Fully charged**
- **DRAM cells leak charge** over time
Two Main Types of DRAM Refresh

1. **Periodic Refresh**: Periodically restores the charge. DRAM cells leak over time.

2. **RowHammer**: Repeatedly accessing a DRAM row can cause bit flips in other physically nearby rows.

   - **Preventive Refresh**: Mitigates RowHammer by refreshing physically nearby rows of a repeatedly accessed row.
Periodic Refresh with Increasing DRAM Chip Density

A larger capacity chip has more rows to be refreshed

A smaller cell stores less charge

More periodic refresh operations incur larger performance overhead as DRAM chip density increases
RowHammer and Preventive Refresh with Increasing DRAM Chip Density

RowHammer vulnerability worsens as DRAM chip density increases

Preventive refresh operations need to be performed more aggressively as DRAM chip density increases
Outline

Background and Problem

Goal and Key Idea

HiRA: Hidden Row Activation

HiRA in Real DRAM Chips

HiRA-MC: HiRA Memory Controller

Performance Evaluation

Conclusion
Our Goal

Reduce the \textit{performance overhead} of DRAM Refresh (both \textit{periodic} and \textit{preventive})
Hide refresh latency by refreshing a DRAM row concurrently with activating another row in a different subarray of the same bank.
Outline

Background and Problem

Goal and Key Idea

HiRA: Hidden Row Activation

HiRA in Real DRAM Chips

HiRA-MC: HiRA Memory Controller

Performance Evaluation

Conclusion
HiRA: Hidden Row Activation – Key Insight

Activating two rows in **quick succession** that are in **different subarrays** in the **same bank** can **refresh one row** concurrently with **activating the other row**.

![Diagram](image)

- **ACT**
  - Subarray X
  - Row A
  - Refreshes Row A concurrently with
  - Subarray Y
  - Row B
  - Activating Row B

**DRAM Bank**
HiRA: Hidden Row Activation – Key Benefit

Refresh RowA concurrently with Activating RowB

Without HiRA

With HiRA

Saved time using HiRA
HiRA: Hidden Row Activation – Closer Look

HiRA refreshes RowA concurrently with activating RowB by issuing \textit{ACT-PRE-ACT} commands in quick succession.
## Outline

- Background and Problem
- Goal and Key Idea
- HiRA: Hidden Row Activation
- HiRA in Real DRAM Chips
- HiRA-MC: HiRA Memory Controller
- Performance Evaluation
- Conclusion
DRAM Testing Infrastructure

FPGA-based SoftMC (Xilinx Virtex UltraScale+ XCU200)

Xilinx Alveo U200 FPGA Board (programmed with SoftMC*)

DRAM Module with Heaters

PCIe Host Interface

MaxWell FT200 Temperature Controller

Fine-grained control over **DRAM commands**, timing parameters ($\pm 1.5\text{ns}$), and **temperature** ($\pm 0.1\degree \text{C}$)

HiRA in Off-the-Shelf DRAM Chips: Key Result 1

- HiRA works in **56 off-the-shelf DRAM chips** from **SK Hynix**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>G.SKILL</td>
<td>DWCW (Partial Marking)* F4-2400C17S-8GNT [39]</td>
<td>2400</td>
<td>42-20</td>
<td>4Gb</td>
<td>B</td>
<td>x8</td>
<td>24.8%</td>
<td>1.75</td>
</tr>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24.9%</td>
<td>1.72</td>
</tr>
<tr>
<td>B0</td>
<td>Kingston</td>
<td>H5AN8G8NDJR-XNC KSM32RD8/16HDR [87]</td>
<td>2400</td>
<td>48-20</td>
<td>4Gb</td>
<td>D</td>
<td>x8</td>
<td>25.1%</td>
<td>1.71</td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25.0%</td>
<td>1.74</td>
</tr>
<tr>
<td>C0</td>
<td>SK Hynix</td>
<td>H5ANAG8NAJR-XN HMAA4GU6AJR8N-XN [109]</td>
<td>2400</td>
<td>51-20</td>
<td>4Gb</td>
<td>F</td>
<td>x8</td>
<td>25.3%</td>
<td>1.47</td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.2%</td>
<td>1.09</td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.5%</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The chip identifier is partially removed on these modules. We infer the chip manufacturer and die revision based on the remaining part of the chip identifier.

- HiRA performs a given row’s **refresh concurrently with activating** any of the **32% of the rows** in the same bank
HiRA in Off-the-Shelf DRAM Chips: Key Result 2

- HiRA works in **56 off-the-shelf DRAM chips** from **SK Hynix**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 A1</td>
<td>G.SKILL</td>
<td>DWCW (Partial Marking)*</td>
<td>2400</td>
<td>42-20</td>
<td>4Gb</td>
<td>B</td>
<td>x8</td>
<td>24.8%</td>
<td>25.0%</td>
<td>25.5%</td>
<td>1.75</td>
<td>1.90</td>
<td>2.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F4-2400C17S-8GNT [39]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24.9%</td>
<td>26.6%</td>
<td>28.3%</td>
<td>1.72</td>
<td>1.94</td>
<td>2.55</td>
</tr>
<tr>
<td>B0 B1</td>
<td>Kingston</td>
<td>H5AN8G8NDJR-XNC KSM32RD8/16HDR [87]</td>
<td>2400</td>
<td>48-20</td>
<td>4Gb</td>
<td>D</td>
<td>x8</td>
<td>25.1%</td>
<td>32.6%</td>
<td>36.8%</td>
<td>1.71</td>
<td>1.89</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25.0%</td>
<td>31.6%</td>
<td>34.9%</td>
<td>1.74</td>
<td>1.91</td>
<td>2.51</td>
</tr>
<tr>
<td>C0 C1 C2</td>
<td>SK Hynix</td>
<td>H5ANAG8NAJR-XN HMAA4GU6AJR8N-XN [109]</td>
<td>2400</td>
<td>51-20</td>
<td>4Gb</td>
<td>F</td>
<td>x8</td>
<td>25.3%</td>
<td>35.3%</td>
<td>39.5%</td>
<td>1.47</td>
<td>1.89</td>
<td>2.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.2%</td>
<td>38.4%</td>
<td>49.9%</td>
<td>1.09</td>
<td>1.88</td>
<td>2.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.5%</td>
<td>36.1%</td>
<td>42.3%</td>
<td>1.49</td>
<td>1.96</td>
<td>2.58</td>
</tr>
</tbody>
</table>

* The chip identifier is partially removed on these modules. We infer the chip manufacturer and die revision based on the remaining part of the chip identifier.

- **51.4% reduction** in the time spent for refresh operations

HiRA **effectively reduces the time spent** for refresh operations in **off-the-shelf** DRAM chips
HiRA in Off-the-Shelf DRAM Chips: Key Results

HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

A. Giray Yağlıkçı¹ Ataberk Olgun¹ Minesh Patel¹ Haocng Luo¹ Hasan Hassan¹
Lois Orosa¹,³ Oğuz Ergin² Onur Mutlu¹
¹ETH Zürich ²TOBB University of Economics and Technology ³Galicia Supercomputing Center (CESGA)

DRAM is the building block of modern main memory systems. DRAM cells must be periodically refreshed to prevent data loss. Refresh operations degrade system performance by interfering with memory accesses. As DRAM chip density increases with technology node scaling, refresh operations also increase because: 1) the number of DRAM rows in a chip increases; and 2) DRAM cells need additional refresh operations to mitigate bit failures caused by RowHammer, a failure mechanism that becomes worse with technology node scaling. Thus, it is critical to enable refresh operations at low performance overhead. To this end, we propose a new operation, Hidden Row Activation (HiRA), and the HiRA Memory Controller (HiRA-MC) to perform HiRA operations.

As DRAM density increases with technology node scaling, the performance overhead of refresh also increases due to three major reasons. First, as the DRAM chip density increases, more DRAM rows need to be periodically refreshed in a DRAM chip [55, 57–61]. Second, as DRAM technology node scales down, DRAM cells become smaller and thus can store less amount of charge, requiring them to be refreshed more frequently [10, 20, 67, 102, 103, 118, 122–124]. Third, with increasing DRAM density, DRAM cells are placed closer to each other, exacerbating charge leakage via a disturbance error mechanism called RowHammer [79, 84, 119, 120, 133, 134, 167, 180, 183], and thus requiring additional refresh operations (called preventive refreshes) to avoid data corruption due to RowHam-

HiRA-MC: HiRA Memory Controller

- **Goal**: Leverage HiRA’s parallelism as much as possible

- **Key Insight**: A *time slack* is needed to find a *row activation* and a *refresh* to perform HiRA

---

**RowA** and **RowZ** are in two *electrically disconnected subarrays*
HiRA-MC: HiRA Memory Controller

Generates each **periodic refresh** and **RowHammer-preventive refresh with a deadline**

1. **Buffers** each **refresh request** and **performs** the refresh request until the **deadline**

2. Finds if it can **refresh a DRAM row concurrently with a DRAM access** or **another refresh**
HiRA-MC: HiRA Memory Controller

HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

A. Giray Yağlıkçı¹ Ataberk Olgun¹ Minesh Patel¹ Haocong Luo¹ Hasan Hassan¹
Lois Orosa¹,³ Oğuz Ergin² Onur Mutlu¹
¹ETH Zürich ²TOBB University of Economics and Technology ³Galicia Supercomputing Center (CESGA)

DRAM is the building block of modern main memory systems. DRAM cells must be periodically refreshed to prevent data loss. Refresh operations degrade system performance by interfering with memory accesses. As DRAM chip density increases with technology node scaling, refresh operations also increase because: 1) the number of DRAM rows in a chip increases; and 2) DRAM cells need additional refresh operations to mitigate bit failures caused by RowHammer, a failure mechanism that becomes worse with technology node scaling. Thus, it is critical to enable refresh operations at low performance overhead. To this end, we propose a new operation, Hidden Row Activation (HiRA), and the HiRA Memory Controller (HiRA-MC) to perform HiRA operations.

As DRAM density increases with technology node scaling, the performance overhead of refresh also increases due to three major reasons. First, as the DRAM chip density increases, more DRAM rows need to be periodically refreshed in a DRAM chip [55, 57–61]. Second, as DRAM technology node scales down, DRAM cells become smaller and thus can store less amount of charge, requiring them to be refreshed more frequently [10, 20, 67, 102, 103, 118, 122–124]. Third, with increasing DRAM density, DRAM cells are placed closer to each other, exacerbating charge leakage via a disturbance error mechanism called RowHammer [79, 84, 119, 120, 133, 134, 167, 180, 183], and thus requiring additional refresh operations (called preventive refreshes) to avoid data corruption due to RowHam-

Outline

Background and Problem

Goal and Key Idea

HiRA: Hidden Row Activation

HiRA in Real DRAM Chips

HiRA-MC: HiRA Memory Controller

Performance Evaluation

Conclusion
Performance Evaluation

• Cycle-level simulations using **Ramulator** [Kim+, CAL 2015]

**System Configuration:**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>3.2 GHz, 8 core, 4-wide issue, 128-entry instr. window</td>
</tr>
<tr>
<td>Last-Level Cache</td>
<td>64-byte cache line, 8-way set-associative, 8 MB</td>
</tr>
<tr>
<td>Memory Scheduler</td>
<td>FR-FCFS</td>
</tr>
<tr>
<td>Address Mapping</td>
<td>Minimalistic Open Pages</td>
</tr>
<tr>
<td>Main Memory</td>
<td>DDR4, 4 bank group, 4 banks per bank group (16 banks per rank)</td>
</tr>
<tr>
<td>Timing Parameters</td>
<td>$t_1=t_2=3\text{ns}$, $t_{\text{RC}}=46.25\text{ns}$, $t_{\text{FAW}}=16\text{ns}$</td>
</tr>
</tbody>
</table>

**Workloads:** **125** different **8-core** multiprogrammed workloads from the SPEC2006 benchmark suite

**DRAM Chip Capacity:** {2, 4, 8, 16, 32, 64, 128} Gb

**RowHammer Threshold:** {1024, 512, 256, 128, 64} activations

The **minimum number of row activations** needed to induce the **first RowHammer bit flip**
HiRA for Periodic Refreshes

- **No-Refresh**: No periodic refresh is performed (Ideal case)
- **Baseline**: Auto-Refresh (using conventional REF commands)

### Weighted Speedup (Normalized to No-Refresh)

- **No Refresh**: Performance slowdown
- **Baseline**: No significant change
- **HiRA**: Performance improvement

Periodic refreshes cause **significant (26%) performance overhead**

**HiRA improves** system performance by **12.6%** over the baseline
HiRA for Preventive Refreshes

- **No Defense**: No RowHammer mitigation employed (i.e., no preventive refresh)
- **PARA [Kim+, ISCA’14]**: the RowHammer defense with the **lowest hardware overhead**

**Diagram Description**

- Bar chart comparing RowHammer thresholds (number of activations) for No-Defense, PARA, and HiRA.
- **No-Defense** shows the highest weighted speedup.
- **HiRA** improves system performance by 3.7x over PARA.
- **PARA** significantly reduces (by 96%) system performance.

**Key Points**

- PARA reduces system performance significantly by 96%.
- HiRA improves system performance by 3.7x over PARA.
More in the Full Paper

• **Real DRAM Chip** Experiments
  - Verification of **HiRA's functionality**
  - **Variation** in HiRA’s characteristics **across banks**

• **Sensitivity to**
  - length of **time slack** for refreshes
  - **number of channels**
  - **number of ranks**

• **Hardware Complexity Analysis**
  - Chip **area cost of 0.0023%** of a processor die per DRAM rank
  - **No additional latency** overhead

• **Experimental Methodology**
  - **Detailed algorithms** for each set of **real chip** experiments
  - Extensive **security analysis** for RowHammer-preventive refreshes

• **Detailed Algorithm of Finding Concurrent Refreshes**
HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

A. Giray Yağlıkçı\textsuperscript{1} Ataberk Olgun\textsuperscript{1} Minesh Patel\textsuperscript{1} Haocong Luo\textsuperscript{1} Hasan Hassan\textsuperscript{1}
Lois Orosa\textsuperscript{1,3} Oğuz Ergin\textsuperscript{2} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \textsuperscript{2}TOBB University of Economics and Technology \textsuperscript{3}Galicia Supercomputing Center (CESGA)

\textbf{DRAM is the building block of modern main memory systems. DRAM cells must be periodically refreshed to prevent data loss. Refresh operations degrade system performance by interfering with memory accesses. As DRAM chip density increases with technology node scaling, refresh operations also increase because: 1) the number of DRAM rows in a chip increases; and 2) DRAM cells need additional refresh operations to mitigate bit failures caused by RowHammer, a failure mechanism that becomes worse with technology node scaling. Thus, it is critical to enable refresh operations at low performance overhead. To this end, we propose a new operation, Hidden Row Activation (HiRA), and the HiRA Memory Controller (HiRA-MC) to perform HiRA operations.}

As DRAM density increases with technology node scaling, the performance overhead of refresh also increases due to three major reasons. First, as the DRAM chip density increases, more DRAM rows need to be periodically refreshed in a DRAM chip [55, 57–61]. Second, as DRAM technology node scales down, DRAM cells become smaller and thus can store less amount of charge, requiring them to be refreshed more frequently [10, 20, 67, 102, 103, 118, 122–124]. Third, with increasing DRAM density, DRAM cells are placed closer to each other, exacerbating charge leakage via a disturbance error mechanism called RowHammer [79, 84, 119, 120, 133, 134, 167, 180, 183], and thus requiring additional refresh operations (called \textit{preventive} refreshes) to avoid data corruption due to RowHam-

\url{https://arxiv.org/pdf/2209.10198.pdf}
Outline

1. Background and Problem
2. Goal and Key Idea
3. HiRA: Hidden Row Activation
4. HiRA in Real DRAM Chips
5. HiRA-MC: HiRA Memory Controller
6. Performance Evaluation
7. Conclusion
Conclusion

**HiRA:** Hidden Row Activation – a new DRAM operation
- First technique that refreshes a DRAM row concurrently with activating another row in the same bank in off-the-shelf DRAM chips
- Real DRAM chip experiments:
  - HiRA works on 56 real off-the-shelf DRAM chips
  - 51.4% reduction in the time spent for refresh operations

**HiRA-MC:** HiRA Memory Controller – a new mechanism
- Leverages HiRA to perform refresh requests concurrently with DRAM accesses and other refresh requests
- HiRA-MC provides:
  - 12.6% speedup by hiding periodic refresh latency
  - 3.7x speedup by hiding RowHammer-preventive refresh latency
HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

Abdullah Giray Yağlıkçı
Ataberk Olgun   Minesh Patel   Haocong Luo   Hasan Hassan
Lois Orosa   Oğuz Ergin   Onur Mutlu

SAFARI

ETH Zürich   CESGA   TOBB ETÜ
HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

Backup Slides

Abdullah Giray Yağlıkçı
Ataberk Olgun   Minesh Patel   Haocong Luo   Hasan Hassan
Lois Orosa   Oğuz Ergin   Onur Mutlu

SAFARI

ETH zürich   CESGA   TOBB ETÜ
The RowHammer Vulnerability

Repeatedly opening (activating) and closing (precharging) a DRAM row in real DRAM chips causes RowHammer bit flips in nearby cells.
Activating a DRAM row refreshes the row and prevents RowHammer bit flips
Mitigating RowHammer

ACT ➔ Row 0 ➔ Row 1 ➔ Row 2 ➔ Row 3 ➔ Row 4

⚠️ Row 2 is being hammered

Refresh neighbor rows

Preventive Refresh

Activating potential victim rows mitigate RowHammer by refreshing them
RowHammer and Preventive Refresh

- **RowHammer**: Repeatedly accessing a DRAM row can cause bit flips in other physically nearby rows.

- **Preventive Refresh**: Refresh a DRAM row when a physically nearby row is activated based on activation counts or probabilistic processes.

Preventive refresh mitigates RowHammer bit flips.
HiRA: Hidden Row Activation

- **HiRA concurrently activates** two rows in a DRAM bank
  - **Challenge 1**: Only one row can be activated in a DRAM bank at a given time
  - **Solution 1**: HiRA violates timing constraints for concurrent row activations

- **HiRA issues two row activation (ACT) commands in quick succession**
  - **Challenge 2**: DRAM chips ignore the second activation before precharge
  - **Solution 2**: HiRA issues a precharge (PRE) command between two ACTs

- **HiRA activates two DRAM rows in the same bank**
  - **Challenge 3**: The two rows can override each other’s data via shared bitlines
  - **Solution 3**: HiRA uses rows from two electrically disconnected subarrays

**HiRA violates DRAM timing constraints** by issuing a sequence of ACT-PRE-ACT commands that target two rows in two electrically disconnected subarrays.
HiRA: Hidden Row Activation

Refreshing *RowA* concurrently with **Activating RowB**

---

**Without HiRA**

- **ACT RowA**
- **PRE**
- **ACT RowB**
- **RD**
- **RD**

**With HiRA**

- **HiRA**
- **PRE**
- **ACT RowA**
- **ACT RowB**
- **RD**
- **RD**

- The time saved using HiRA
- Reduction in the time spent for two refreshes

---

**HiRA**:

- **Hidden Row Activation**

---

**SAFARI**

---

41
HiRA Operation

HiRA refreshes RowA concurrently with activating RowB by issuing \textit{ACT-PRE-ACT} commands in quick succession.
HiRA in Off-the-Shelf DRAM Chips: Key Results

- HiRA works in **56 off-the-shelf DRAM chips** from **SK Hynix**

- **51.4% reduction** in the time spent for refresh operations

- HiRA performs a given row’s **refresh concurrently with activating** any of the **32% of the rows** in the same bank

HiRA effectively reduces the time spent for **refresh** operations in **off-the-shelf** DRAM chips
HiRA Support in Off-the-Shelf DRAM Chips

• 56 off-the-shelf DDR4 DRAM chips support HiRA (from SK Hynix)
• HiRA Coverage of a given DRAM row:
  - Refresh a given DRAM row while activating other rows in the same bank
  - We sweep two timing parameters: $t_1$ and $t_2$

HiRA can refresh a DRAM row concurrently with 32% of any of the other DRAM rows in the same bank

$t_1$ and $t_2$ can be as small as 3ns
HiRA’s Second Row Activation

- Does performing HiRA in between refresh the victim row?
  - If HiRA’s second row activation is performed, more activations are needed to induce RowHammer bit flips
  - If HiRA’s second row activation is ignored, RowHammer threshold should not change

![Graphs showing absolute and normalized RowHammer thresholds with and without HiRA]
Variation across DRAM Banks

• Coverage: Identical across banks
• The effect of second row activation
HiRA-MC: HiRA Memory Controller

• **Goal**: Leverage HiRA’s parallelism as much as possible

• **Periodic** and **preventive** refresh controllers generate each refresh request **with a deadline**

• **Refresh Table** buffers a refresh request until its **deadline**

• **Concurrent Refresh Finder** finds if HiRA can refresh a row
  - **Concurrently with a memory request**
  - **Concurrently with another refresh request**
The Concurrent Refresh Finder

**Case 1:** Executes when a precharge is issued (completes before the precharge completes)

**Case 2:** Periodically executes after every $t_{RC}$ (completes before $t_{RC}$)
HiRA-MC Example

• Case 1: **Refresh – Access** Parallelism

Memory Request Queue

| RD SA:B Row:0 |
| RD SA:A Row:0 |

Refresh Table

| REF SA:C Row:2 |
| REF SA:B Row:1 |
| REF SA:B Row:6 |

HiRA(Row SA:B Row:6, SA:A Row:0)

- ACT SA:B Row:6
- ACT SA:A Row:0

PRE

6ns

• Case 2: **Refresh – Refresh** Parallelism

Memory Request Queue

| RD SA:B Row:0 |

Refresh Table

| REF SA:C Row:2 |
| REF SA:B Row:1 |

HiRA(Row SA:B Row:1, SA:C Row:2)

- ACT SA:B Row:1
- ACT SA:C Row:2

PRE

6ns

HiRA-MC provides **refresh-access** and **refresh-refresh** parallelism
HiRA-MC Hardware Complexity

• We use CACTI with 22nm technology node

<table>
<thead>
<tr>
<th>HiRA-MC Component</th>
<th>Area (mm²)</th>
<th>Area (% of Chip Area)</th>
<th>Access Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh Table</td>
<td>0.00031</td>
<td>&lt;0.0001%</td>
<td>0.07ns</td>
</tr>
<tr>
<td>RefPtr Table</td>
<td>0.00683</td>
<td>0.0017%</td>
<td>0.12ns</td>
</tr>
<tr>
<td>PR-FIFO</td>
<td>0.00029</td>
<td>&lt;0.0001%</td>
<td>0.07ns</td>
</tr>
<tr>
<td>Subarray Pairs Table</td>
<td>0.00180</td>
<td>0.0005%</td>
<td>0.09ns</td>
</tr>
<tr>
<td>Overall</td>
<td>0.00923</td>
<td><strong>0.0023%</strong></td>
<td><strong>6.31ns</strong></td>
</tr>
</tbody>
</table>

HiRA-MC consumes only **0.0023%** of CPU chip area per DRAM rank

HiRA-MC does not increase memory access latency
Estimating Periodic Refresh Overhead

\[ t_{RFC} = 110 \times C_{\text{chip}}^{0.6} \]

Latency of a REF command

DRAM Chip Capacity

Nonblocking Memory Refresh

Kate Nguyen, Kehan Lyu, Xianze Meng  
Department of Computer Science  
Virginia Tech  
Blacksburg, Virginia  
katevy@vt.edu, kehan@vt.edu, xianze@vt.edu

Vilas Sridharan  
RAS Architecture  
Advanced Micro Devices, Inc  
Boxborough, Massachusetts  
vilas.sridharan@amd.com

Xun Jian  
Department of Computer Science  
Virginia Tech  
Blacksburg, Virginia  
xunj@vt.edu
Reducing Overall Latency of Two Refreshes

• Refreshing two rows using nominal timing parameters:

Using HiRA:

Overall latency of refreshing two rows reduces by 51.4% from 78.25ns down to 38ns
# Tested DRAM Chips

## Table 4: Characteristics of the tested DDR4 DRAM modules.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>G.SKILL</td>
<td>DWCW (Partial Marking)* F4-2400C17S-8GNT [39]</td>
<td>2400</td>
<td>42-20</td>
<td>4Gb</td>
<td>B</td>
<td>x8</td>
<td>24.8%</td>
<td>25.0%</td>
<td>25.5%</td>
<td>1.75</td>
<td>1.90</td>
<td>2.52</td>
</tr>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24.9%</td>
<td>26.6%</td>
<td>28.3%</td>
<td>1.72</td>
<td>1.94</td>
<td>2.55</td>
</tr>
<tr>
<td>B0</td>
<td>Kingston</td>
<td>H5AN8G8NDJR-XNC KSM32RD8/16HDR [87]</td>
<td>2400</td>
<td>48-20</td>
<td>4Gb</td>
<td>D</td>
<td>x8</td>
<td>25.1%</td>
<td>32.6%</td>
<td>36.8%</td>
<td>1.71</td>
<td>1.89</td>
<td>2.34</td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25.0%</td>
<td>31.6%</td>
<td>34.9%</td>
<td>1.74</td>
<td>1.91</td>
<td>2.51</td>
</tr>
<tr>
<td>C0</td>
<td>SK Hynix</td>
<td>H5ANAG8NAJR-XN HMMA4GU6AJR8N-XN [109]</td>
<td>2400</td>
<td>51-20</td>
<td>4Gb</td>
<td>F</td>
<td>x8</td>
<td>25.3%</td>
<td>35.3%</td>
<td>39.5%</td>
<td>1.47</td>
<td>1.89</td>
<td>2.23</td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>29.2%</td>
<td>38.4%</td>
<td>49.9%</td>
<td>1.09</td>
<td>1.88</td>
<td>2.27</td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.5%</td>
<td>36.1%</td>
<td>42.3%</td>
<td>1.49</td>
<td>1.96</td>
<td>2.58</td>
</tr>
</tbody>
</table>

* The chip identifier is partially removed on these modules. We infer the chip manufacturer and die revision based on the remaining part of the chip identifier.

HiRA-MC: HiRA Memory Controller

• **Periodic** and **preventive** refresh controllers generate each refresh request **with a deadline**
• **Refresh Table** buffers a refresh request **until its deadline**
• **Concurrent Refresh Finder** finds if HiRA can refresh a row
  - **Concurrently with a DRAM access**
  - **Concurrently with another refresh request**
HiRA for Periodic Refreshes

a) HiRA's perf. overhead, compared to No Refresh

b) HiRA's perf. improvement compared to Baseline

- Baseline
- HiRA-0
- HiRA-2
- HiRA-4
- HiRA-8

Weighted Speedup (Norm. to No Refresh)

Weighted Speedup (Norm. to Baseline)

DRAM Chip Capacity (Gb)
RowHammer Thresholds

a) PARA's probability threshold ($p_{th}$) for different values of $N_{RH}$ and $t_{RefSlack}$

b) Overall RowHammer success probability for different values of $N_{RH}$ and $t_{RefSlack}$
HiRA for Preventive Refreshes

a) PARA's perf. overhead with and without HiRA

b) HiRA's perf. improvement compared to PARA
HiRA for Periodic Refresh

Sensitivity to Number of Channels and Ranks

[Diagrams showing normalized weighted speedup for different numbers of channels and ranks for 2Gb, 8Gb, and 32Gb DRAM chips.]
HiRA for Preventive Refresh
Sensitivity to Number of Channels and Ranks
Workload Memory Access Characteristics

- 125 different 8-core multiprogrammed workloads

- Three histograms showing MPKI, RBCPKI, and RBHPKI respectively
RowHammer Mitigation across Generations

Refresh Delay

• DDRx protocols allow a REF command to be **postponed** for ~70us

• HiRA-MC’s current design **does not** leverage this flexibility

• A **longer time slack** allows
  - the baseline to **better utilize** DRAM idle time to perform refresh operations
  - HiRA to find **more opportunities** to perform a refresh operation **concurrently with** a DRAM access

• **Future sensitivity study:** the effect of long refresh delays

SAFARI
Energy

• **HiRA does not change** the **number of refresh operations** at a given time window
  - Overall energy consumed for refresh operations is the same

• **HiRA improves system performance**
  - Reduces the background energy consumption

• Evaluation requires an **accurate power model** based on real system measurements, similar to VAMPIRE [Ghose+ SIGMETRICS’17], but for HiRA operations
HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

Abdullah Giray Yağlıkçı
Ataberk Olgun  Minesh Patel  Haocong Luo  Hasan Hassan
Lois Orosa  Oğuz Ergin  Onur Mutlu

SAFARI