A Case for Refresh Pausing in DRAM Memory Systems

Prashant Nair
Chia-Chen Chou
Moinuddin Qureshi
Dynamic Random Access Memory (DRAM) used as main memory
• DRAM stores data as charge on capacitor
Introduction

• Dynamic Random Access Memory (DRAM) used as main memory
• DRAM stores data as charge on capacitor
Dynamic Random Access Memory (DRAM) used as main memory

• DRAM stores data as charge on capacitor

DRAM is a volatile memory ➔ Charge leaks quickly
Refresh: Restoring Data in DRAM

DRAM maintains data by Refresh operations
Refresh: Restoring Data in DRAM

DRAM maintains data by Refresh operations

Charge on cells restored
DRAM maintains data by Refresh operations.

Charge on cells restored.

JEDEC specified DRAM retention time:
- 64ms (< 85°C)
- 32ms (> 85°C)

Time between Refresh ≤ Retention Time.

DRAM relies on Refresh for data integrity.
Refresh: A Growing Problem

Time spent in Refresh proportional to number of Rows

Increasing memory capacity ➔ More time spent in Refresh

The time for doing Refresh is increasing with chip density
Refresh Blocks Reads

Memory unavailable for Read/Write during Refresh
Refresh Blocks Reads

Memory unavailable for Read/Write during Refresh

No Refresh

A

B
Refresh Blocks Reads

Memory unavailable for Read/Write during Refresh

No Refresh

Interference due to Refresh
Refresh Blocks Reads

Memory unavailable for Read/Write during Refresh

- No Refresh
- Interference due to Refresh

Wait

- Refresh blocks reads ➔ Higher read latency
Impact of Refresh

Increase in Read Latency

- 8Gb
- 16Gb
- 32Gb

Impact of Refresh
Impact of Refresh

![Bar chart showing the increase in read latency for 8Gb, 16Gb, and 32Gb]
Impact of Refresh

Increase in Read Latency

Performance Loss

- 8Gb
- 16Gb
- 32Gb
Impact of Refresh

Increase in Read Latency

- 8Gb: 0%
- 16Gb: 30%
- 32Gb: 60%

Performance Loss

- 8Gb: 5%
- 16Gb: 25%
- 32Gb: 40%
Impact of Refresh

Increase in Read Latency

- 8Gb: 0%
- 16Gb: 30%
- 32Gb: 50%

Performance Loss

- 8Gb: 5%
- 16Gb: 25%
- 32Gb: 40%
**Impact of Refresh**

Impact of Refresh is significant, and increasing.

Our Goal: Reduce the Read Latency impact of Refresh
Outline

- Introduction & Motivation
- Refresh Operation: Background
- Refresh Pausing
- Evaluation
- Alternative Proposals
- Summary
Refresh Operation

A DRAM Bank

- Row 1
- Row 2
- Row 3
- Row 4
- Row 5
- Row n-1
- Row n

Refresh operates on a Row granularity
Refresh Operation

A DRAM Bank

Row 1
Row 2
Row 3
Row 4
Row 5
Row n-1
Row n

Refresh operates on a Row granularity
Refresh Modes

- **Burst Mode:**
  
  Memory unavailable until all rows finish refresh

- **Distributed Mode:**
Refresh Modes

- **Burst Mode:**

  Memory unavailable until all rows finish refresh

- **Distributed Mode:**

  8K refresh pulses in 64ms
Refresh Modes

- **Burst Mode:**
  Memory unavailable until all rows finish refresh

- **Distributed Mode:**
  8K refresh pulses in 64ms
  Distributed mode reduces contention from Refresh
Every pulse refreshes a ‘Bundle of rows’

<table>
<thead>
<tr>
<th>Chip Size</th>
<th>Rows in a Refresh bundle (per bank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 Mb</td>
<td>1</td>
</tr>
<tr>
<td>1Gb</td>
<td>2</td>
</tr>
<tr>
<td>2Gb</td>
<td>4</td>
</tr>
<tr>
<td>4Gb or 8Gb (Twin 4Gb die)</td>
<td>8</td>
</tr>
</tbody>
</table>
Every pulse refreshes a ‘Bundle of rows’

<table>
<thead>
<tr>
<th>Chip Size</th>
<th>Rows in a Refresh bundle (per bank)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 Mb</td>
<td>1</td>
</tr>
<tr>
<td>1Gb</td>
<td>2</td>
</tr>
<tr>
<td>2Gb</td>
<td>4</td>
</tr>
<tr>
<td>4Gb or 8Gb (Twin 4Gb die)</td>
<td>8</td>
</tr>
</tbody>
</table>

Refresh Bundle currently have upto 8 rows, and increasing
The Latency Wall of Refresh

$T_{RFC}$ is the time to do refresh for every refresh pulse
The Latency Wall of Refresh

$T_{RFC}$ is the time to do refresh for every refresh pulse
The Latency Wall of Refresh

$T_{RFC}$ is the time to do refresh for every refresh pulse

- **8Gb:** Available
- **16Gb:** Available
- **32Gb:** Available


**The Latency Wall of Refresh**

$T_{RFC}$ is the time to do refresh for every refresh pulse

Current 8Gb chips have $T_{RFC}$ of 350ns >> read latency

High $T_{RFC}$ $\Rightarrow$ Read waits for refresh for long time
Outline

- Introduction & Motivation
- Refresh Operation: Background
- **Refresh Pausing**
- Evaluation
- Alternative Proposals
- Summary
Refresh Pausing

Insight: Make Refresh Operations Interruptible
Refresh Pausing

Insight: Make Refresh Operations Interruptible

Baseline system

time
Refresh Pausing

Insight: Make Refresh Operations Interruptible
Refresh Pausing

Insight: Make Refresh Operations Interruptible

Request B arrives

Baseline system
Refresh Pausing

Insight: Make Refresh Operations Interruptible

Request B arrives

Baseline system
Insight: Make Refresh Operations Interruptible
Insight: Make Refresh Operations Interruptible

Baseline system

Request B arrives

Refresh Pausing
Insight: Make Refresh Operations Interruptible

Refresh Pausing

Baseline system

Request B arrives

Refresh Pausing

Request B arrives
Refresh Pausing

Insight: Make Refresh Operations Interruptible

Baseline system

Request B arrives

Refresh Pausing

Request B arrives

Interrupted
Refresh Pausing

Insight: Make Refresh Operations Interruptible

Baseline system

A Refresh Pausing B

Request B arrives

Request B arrives

Interrupted

Refresh Pausing
Insight: Make Refresh Operations Interruptible
Insight: Make Refresh Operations Interruptible

Pausing Refresh reduces wait time for Reads
Insight: Make Refresh Operations Interruptible

Pause at arbitrary point can cause data loss

Pausing Refresh reduces wait time for Reads
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing

Bank

Chip

Rows

Read X

Row Buffer

a
b
c
d
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing

Bank

Chip

Rows

Read X

Row Buffer
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

Without Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing

Bank

Chip

Rows

Row Buffer
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing

Pause

Read X
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing

Bank

Chip

Rows

Row Buffer

a
b
c
d
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing
Refresh Pausing: When to Pause?

Refresh Pulse
(4 rows in a bundle)

With Refresh Pausing

Bank

Chip

Rows

Row Buffer

Refresh Pausing at Row boundary to service read
Refresh Pausing: Interface Details

• Memory Controller generates a Refresh Enable (RE) signal
• Pausing requires ‘active low’ detection of RE
• One way communication only
Refresh Pausing: Interface Details

- Memory Controller generates a Refresh Enable (RE) signal
- Pausing requires ‘active low’ detection of RE
- One way communication only
Refresh Pausing: Interface Details

- Memory Controller generates a Refresh Enable (RE) signal
- Pausing requires ‘active low’ detection of RE
- One way communication only
Refresh Pausing: Interface Details

- Memory Controller generates a Refresh Enable (RE) signal
- Pausing requires ‘active low’ detection of RE
- One way communication only
Refresh Pausing: Track a Paused Row

- Row Address Counter increments the addresses

[Diagram of DRAM with Address Generator, Row Address Counter, and Incremener]
Refresh Pausing: Track a Paused Row

- Row Address Counter increments the addresses
- Stop the increment using a simple AND gate
Refresh Pausing: Track a Paused Row

- Row Address Counter increments the addresses
- Stop the increment using a simple AND gate
- Active Low Refresh Enable as ‘Refresh Pause’
Scheduler schedules: Read, Write, and Refresh

Responsible for Pausing Refresh for Read

Keeps track of refresh time done before Pause
Forced Refresh

- Pausing can delay Refresh

- JEDEC allows delay of up-to 8 pending refresh
Forced Refresh

- Pausing can delay Refresh

- JEDEC allows delay of up-to 8 pending refresh

- If 8 pending refresh, then issue ‘Forced Refresh’
Forced Refresh

- Pausing can delay Refresh

- JEDEC allows delay of up-to 8 pending refresh

- If 8 pending refresh, then issue ‘Forced Refresh’

- Forced Refresh cannot be Paused

Forced Refresh for data integrity
Outline

- Introduction & Motivation
- Refresh Operation: Background
- Refresh Pausing
- Evaluation
- Alternative Proposals
- Summary
Experimental Setup

• **Simulator:** uSIMM from Memory Scheduling Championship (MSC)

• **Workloads:** MSC Suite
  COMMERCIAL(5), PARSEC(9), BIOBENCH(2) and SPEC(2)

• **Configuration:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cores</td>
<td>4</td>
</tr>
<tr>
<td>Last Level Cache</td>
<td>1MB</td>
</tr>
<tr>
<td>DRAM (DDR3)</td>
<td>8 Chips/Rank, 8Gb/Chip</td>
</tr>
<tr>
<td>Channels, Ranks, Banks</td>
<td>4,2,8</td>
</tr>
<tr>
<td>Refresh (Baseline)</td>
<td>Distributed (JEDEC)</td>
</tr>
</tbody>
</table>

• Results presented for temperature > 85C (paper also has <85C)
Results: Read Latency

Normalized Read Latency

- **COMMERCIAL**
- **SPEC**
- **PARSEC**
- **BIOBENCH**
- **GMEAN**

- **Refresh Pausing**
- **No Refresh**
- Refresh Pausing gives ~7% read latency reduction for an 8Gb chip
Results: Performance

Performance Comparison

<table>
<thead>
<tr>
<th>Speedup</th>
<th>COMMERCIAL</th>
<th>SPEC</th>
<th>PARSEC</th>
<th>BIOBENCH</th>
<th>GMEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh Pausing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Refresh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- Refresh Pausing gives ~5% performance improvement for an 8Gb chip
Results: Impact of Chip Density

- **8Gb**
  - Refresh Pausing: Speedup 1.0
  - No Refresh: Speedup 1.1

- **16Gb**
  - Refresh Pausing: Speedup 1.2
  - No Refresh: Speedup 1.3

- **32Gb**
  - Refresh Pausing: Speedup 1.4
  - No Refresh: Speedup 1.4
Results: Impact of Chip Density

[Bar chart showing speedup for different chip densities (8Gb, 16Gb, 32Gb) with and without refresh pausing.]
Results: Impact of Chip Density

Impact of Density on Refresh Pausing

Results: Impact of Chip Density
Results: Impact of Chip Density

Refresh Pausing more effective as chips density increases
Outline

- Introduction & Motivation
- Refresh Operation: Background
- Refresh Pausing
- Evaluation
- **Alternative Proposals**
- Summary
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

![Diagram showing request and refreshes](image-url)
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

No Refreshes

Request A

Request B

time
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request

![Diagram showing requests A and B with and without refreshes.]

No Refreshes

With Refreshes

Request A ↔ B

3 units

time

time
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request

No Refreshes

With Refreshes

Request A 3 units Request B

A B

Request A

A
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

No Refreshes

With Refreshes

Request A

Request B

A

B

3 units

Refresh

time

time

[MICRO’10]
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request

No Refreshes

With Refreshes

[MICRO’10]
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request

No Refreshes

With Refreshes
Elastic Refresh waits for idle period before issuing a refresh

- Estimates average inter-arrival time of memory request

No Refreshes

- Request A to Request B: 3 units

With Refreshes

- Request A to Request B: 4 units

Elastic Refresh

- Refreshing process with consideration of time units
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

No Refreshes

With Refreshes

Elastic Refresh
Elastic Refresh waits for idle period before issuing a refresh

Estimates average inter-arrival time of memory request
Elastic Refresh for Scheduling Refresh

[MICRO’10]

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

No Refreshes:
- Request A to B in 3 units

With Refreshes:
- Request A to Refresh in 4 units

Elastic Refresh:
- Wait before Refresh
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

![Diagram showing refresh scenarios](image)

**No Refreshes**
- Request A to B: 3 units

**With Refreshes**
- Request A to B: 4 units

**Elastic Refresh**
- Request A to Wait: 7 units
Elastic Refresh for Scheduling Refresh

- Elastic Refresh waits for idle period before issuing a refresh
- Estimates average inter-arrival time of memory request

The “Wait and Watch” policy can increase wait times
Comparison with Elastic Refresh

Refresh Pausing outperforms Elastic Refresh
DDR4 proposals: x2 and x4 modes

Reduce bundles size and have more bundles
DDR4 proposals: x2 and x4 modes

Reduce bundles size and have more bundles

- In x2 mode, $T_{REFI}$ is reduced by 2 (x4 mode by 4)
- In x2 mode $T_{RFC}$ is reduced by 2 (x4 mode by 4)
DDR4 proposals: x2 and x4 modes

Reduce bundles size and have more bundles

- In x2 mode, $T_{REFI}$ is reduced by 2 (x4 mode by 4)
- In x2 mode $T_{RFC}$ is reduced by 2 (x4 mode by 4)

Fine Grained Refresh to reduce contention of Refresh
Comparison with DDR4

DDR4 modes (x2 and x4) useful but not enough
Comparison with DDR4

DDR4 modes (x2 and x4) useful but not enough
Outline

- Introduction & Motivation
- Refresh Operation: Background
- Refresh Pausing
- Evaluation
- Alternative Proposals
- Summary
Summary

• DRAM relies on Refresh for data integrity
• Time for Refresh increases with chip density
• Refresh blocks read, increases read latency
• Refresh Pausing: make Refresh Interruptible
• Pausing provides 5% improvement for 8Gb, increases with higher density
• Applicable also to DDR4 (fine grained refresh)