Credit where credit is due:
Slides contain original artwork (© Jacob, Wang 2005)
Move Data on and off

1. Command transport and decode
2. In bank data movement
3. In device data movement
4. System data transport
Basic Commands

- Row Access
- Column Read
- Column Write
- Precharge
- Refresh
- Need other commands to manage special structures. (i.e. Write buffers, virtual channels)
Row Access

cmd & addr bus
bank utilization
device utilization
data bus

addr & cmd
data restored to DRAM cells
t<sub>RCD</sub>
t<sub>RAS</sub>

1. cmd
2. data sense
3. data restored to DRAM cells

address and command bus
sense amplifiers
data bus

time
Column Read

Generic DRAM device (one rank)

1. addr & cmd
2. cmd & addr bus
3. bank access
4. data burst

I/O gating

bank utilization

device utilization

data bus

cmd

tCAS

tBurst

time

address and command bus

data bus

SLIDE 5
### Column Write

**Generic DRAM device (one rank)**

- **cmd & addr bus**
- **bank utilization**
- **device utilization**
- **data bus**

**Diagram:**
- **tCWD**: Time for command word delay
- **tBurst**: Burst time
- **tWR**: Write time
- **I/O gating**: Input/output gating
- **data burst**: Data burst
- **bank access**: Bank access

**SDRAM**
- **cmd**: Command
- **data burst**: Data burst
- **tCWD = 0**

**DDR SDRAM**
- **cmd**: Command
- **data burst**: Data burst
- **tCWD = 1** (full clock cycle)

**DDR2 SDRAM**
- **cmd**: Command
- **data burst**: Data burst
- **tCWD = tCAS - 1** (full cycle)
Precharge

Generic DRAM device (one rank)

cmd & addr bus
data bus

bank utilization
device utilization

Row access to same bank
(previous command)

address and command bus
data bus

sense amplifiers

Decodor

Time

tRC
tRAS
tRP

1

2

cmd

bank precharge
Refresh I

Generic DRAM device (one rank)

address and command bus
data bus

sense amplifiers

Row access to all banks  all banks precharge

cmd
ttRFC
ttRC
ttRAS
ttRP

additional time needed by DRAM device to recover from current spike (all bank refresh)

Basic Refresh Command: All Banks

* Refresh cycle time dominated by device power consideration, not resource usage. Model is only for resource usage. In real devices, refresh cycle times take longer than tRC. Getting much worse due to larger rows. Parallel refresh hitting all banks draws large current spike. Takes longer time to recover for higher density devices.
Refresh II

<table>
<thead>
<tr>
<th>Density</th>
<th>Bank Count</th>
<th>Row Count</th>
<th>Row Size (bits)</th>
<th>Row cycle time (ns)</th>
<th>Refresh Cycle time (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 Mbit (x8)</td>
<td>4</td>
<td>8192</td>
<td>8192</td>
<td>55</td>
<td>75</td>
</tr>
<tr>
<td>512 Mbit (x8)</td>
<td>4</td>
<td>16384</td>
<td>8192</td>
<td>55</td>
<td>105</td>
</tr>
<tr>
<td>1 Gbit (x8)</td>
<td>8</td>
<td>16384</td>
<td>8192</td>
<td>55</td>
<td>127.5</td>
</tr>
</tbody>
</table>

Trend: Refresh getting more expensive

Advanced Refresh? Per Bank, Hidden?
A Read Cycle

- cmd & addr bus
- bank utilization
- device utilization
- data bus
- row access
- column read
- precharge
- time
- tRC
- tRAS
- tRP
- tRCD
- tCAS (tCL)
- tBurst
- row act
- col read
- data sense
- bank access
- I/O gating
- data burst
- prec.
- data restore
- array precharge
- row act
- row access column read precharge
Read and Precharge

Diagram showing the timing and operations involved in read and precharge operations in memory systems. The diagram includes:

- `tRCD`: Row Access to Column Data
- `tRC`: Row Precharge Delay
- `tRAS`: Row Access Delay
- `tRP`: Row Precharge
- `tCAS`: Column Access Time
- `tCL`: Column Load Time
- `tBurst`: Burst Time
- `read & prec`: Read and Precharge
- `I/O gating`: I/O Gating
- `data burst`: Data Burst
- `implicit precharge`: Implicit Precharge

The diagram illustrates the sequence of events and timing parameters involved in memory operations, including row activation, data sense, data restore, and precharge processes.
Posted CAS

CAS timing remains the same, but delay CAS command internally for X cycles.
Simplifies controller design.
How do DRAM command interactions look like in system context?
Reads to Same Rank

Can be pipelined consecutively in SDRAM/DDRx SDRAM/DRDRAM memory systems
Reads to Diff. Rows of Same Bank

Best Case: \( t_{RAS} \) already satisfied

Worst Case: \( t_{RAS} \) not yet satisfied
Read to Diff. Banks/Conflict

Two reads, second read to different bank, but has bank conflict.

Support Re-ordering?
If not... Best case shown above

If yes... Precharge ASAP
Read to Different Ranks

- If $i \neq j$
  - $n \neq m$
  - Command and address
  - Bank "i" of rank "m"
  - Bank "j" of rank "n"
  - Rank "m" utilization
  - Rank "n" utilization
  - Data bus
  - Time
  - Data burst
  - I/O gating

---

SDRAM: $t_{DQS} = 0$

DDR$x$ SDRAM: $t_{DQS} \geq 1$ full cycle

Direct RDRAM: $t_{DQS} = 0$
Consecutive Writes

- cmd\&addr
- bank “i” of rank “m”
- bank “j” of rank “n”
- rank “m” utilization
- rank “n” utilization
- data bus
  - i ? j Does not matter
  - n ? m Does not matter
- addr & cmd

- bank i access
- I/O gating
- data restore
- I/O gating
- data burst
- data burst
- tBurst
- tCWD
- tBurst

- decode
- Rank n
- data bus
- decode
- Rank m

- time
Writes W/Bank Conflict

Bank conflict to same bank: best case

Bank conflict to different ranks: No re-ordering
Write following Read

Open Banks: Best Case

Bank Conflict: Best Case
Read following Write I

To same rank:
SDRAM/DDRrx
no write buffer,
writes must complete before reads can begin.

D-RDRAM: has write buffer
Read following Write II

To different ranks: read can proceed “immediately”, subject to data bus sharing constraint.
# Minimum Command Distances

<table>
<thead>
<tr>
<th>Previous Rank</th>
<th>Rank</th>
<th>Minimum scheduling distance between DRAM commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Command Re-Ordering</td>
<td></td>
<td>Best Case</td>
</tr>
<tr>
<td>Worst Case</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $R_{ss} o$: $t_{Burst}$
- $R_{ss} c$: $t_{Burst} + t_{RP} + t_{RCD}$
- $R_{sd} o$: $t_{Burst}$
- $R_{sd} c$: $t_{CMD} + t_{RP} + t_{RCD}$
- $R_{d} o$: $t_{DQS} + t_{Burst}$
- $R_{d} c$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{ss} o$: $t_{CAS} + t_{Burst} + t_{DQS} - t_{CWD}$
- $W_{ss} c$: $t_{Burst} + t_{RP} + t_{CMD}$
- $W_{sd} o$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{sd} c$: $t_{Burst} + t_{WR} + t_{RP} + t_{RCD} - t_{CMD}$
- $W_{d} o$: $t_{DQS} + t_{Burst} - t_{CMD}$
- $W_{d} c$: $t_{Burst} + t_{WR} + t_{RP} + t_{RCD} - t_{CMD}$
- $W_{s} o$: $t_{Burst}$
- $W_{s} c$: $t_{Burst} + t_{RP} + t_{RCD}$
- $W_{d} o$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{d} c$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{s} o$: $t_{Burst}$
- $W_{s} c$: $t_{Burst} + t_{RP} + t_{RCD}$
- $W_{d} o$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{d} c$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{d} o$: $t_{CMD} + t_{RP} + t_{RCD}$
- $W_{d} c$: $t_{CMD} + t_{RP} + t_{RCD}$
Power Constraint

Current profile of a read cycle

Row Activation draws a lot of current
Pipelined Read Cycles

Start to pipeline DRAM commands, and it can get really bad
Dumb Solution: $t_{RRD}$ & $t_{FAW}$

$t_{RRD}$ and $t_{FAW}$ spec’ed to keep row activation commands far apart.

Limits peak current draw, but random row access performance suffers.
Worse with Larger Rows

<table>
<thead>
<tr>
<th>Device configuration</th>
<th>512 Mbit x 4</th>
<th>256 Mbit x 8</th>
<th>128 Mbit x 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data bus width</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Number of banks</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Number of rows</td>
<td>16384</td>
<td>16384</td>
<td>8192</td>
</tr>
<tr>
<td>Number of columns</td>
<td>2048</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Row size (bits)</td>
<td>8192</td>
<td>8192</td>
<td>16384</td>
</tr>
<tr>
<td>( t_{RRD} ) (ns)</td>
<td>7.5</td>
<td>7.5</td>
<td>10</td>
</tr>
<tr>
<td>( t_{FAW} ) (ns)</td>
<td>37.5</td>
<td>37.5</td>
<td>50</td>
</tr>
</tbody>
</table>

1 Gbit DDR2 SDRAM Devices

Larger Rows = longer \( t_{RRD} \) & \( t_{FAW} \)
Same issue as \( t_{RFC} \)

Power consumption IS important for DRAM
Summary

- More than you ever wanted to know about DRAM command scheduling.
- DRAM commands are quite simple, but command combinations increase complexity.
- Minimum scheduling distances based on resource usage/availability model, but additional constraints exist. i.e. Power limit.
- Additional resources (i.e. write buffers, caches, virtual channels) require additional commands.