Main Memory Background

- **Performance of Main Memory:**
  - **Latency:** Cache Miss Penalty
    - **Access Time:** time between request and word arrives
    - **Cycle Time:** time between requests
  - **Bandwidth:** I/O & Large Block Miss Penalty (L2)

- **Main Memory is DRAM:** Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (8 ms, 1% time)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - **RAS** or **Row Address Strobe**
    - **CAS** or **Column Address Strobe**

- **Cache uses SRAM:** Static Random Access Memory
  - No refresh (6 transistors per bit vs. 1 transistor + 1 capacitor per bit)
  - **Size:** SRAM/DRAM 4-8,
  - **Cycle time:** DRAM/SRAM 8-16
Memory subsystem organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column
Breaking down a DIMM

DIMM (Dual in-line memory module)

Serial presence detect ( SPD)
- Stored in EEPROM on module
- Has info to configure mem controllers

Rank 0: collection of 8 chips
Rank 1
### Dimm & Rank (from JEDEC)

#### Unbuffered Dimm Details

<table>
<thead>
<tr>
<th>Row Card Version</th>
<th>Dimm Capacity</th>
<th>Dimm Organization</th>
<th>Sdram Density</th>
<th>Sdram Organization</th>
<th>Number of Sdram</th>
<th>Number of Physical Ranks</th>
<th>Number of Ranks in Sdram</th>
<th>Number of Address Bits/Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>512MB</td>
<td>64 M x 64</td>
<td>512 Mbit</td>
<td>64 M x 8</td>
<td>8</td>
<td>8</td>
<td>15/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1GB</td>
<td>128 M x 64</td>
<td>1 Gbit</td>
<td>128 M x 8</td>
<td>8</td>
<td>1</td>
<td>14/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2GB</td>
<td>256 M x 64</td>
<td>2 Gbit</td>
<td>256 M x 8</td>
<td>8</td>
<td>1</td>
<td>15/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4GB</td>
<td>512 M x 64</td>
<td>4 Gbit</td>
<td>512 M x 8</td>
<td>8</td>
<td>1</td>
<td>16/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8GB</td>
<td>1 G x 8</td>
<td>8 Gbit</td>
<td>1 G x 8</td>
<td>8</td>
<td>1</td>
<td>16/11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16GB</td>
<td>1 G x 8</td>
<td>8 Gbit</td>
<td>1 G x 8</td>
<td>8</td>
<td>2</td>
<td>16/11</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1GB</td>
<td>128 M x 64</td>
<td>512 Mbit</td>
<td>64 M x 8</td>
<td>16</td>
<td>2</td>
<td>13/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2GB</td>
<td>256 M x 64</td>
<td>1 Gbit</td>
<td>128 M x 8</td>
<td>16</td>
<td>2</td>
<td>14/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4GB</td>
<td>512 M x 64</td>
<td>2 Gbit</td>
<td>256 M x 8</td>
<td>16</td>
<td>2</td>
<td>15/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8GB</td>
<td>1 G x 8</td>
<td>4 Gbit</td>
<td>512 M x 8</td>
<td>16</td>
<td>2</td>
<td>16/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16GB</td>
<td>2 G x 8</td>
<td>8 Gbit</td>
<td>1 G x 8</td>
<td>16</td>
<td>2</td>
<td>16/11</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>32GB</td>
<td>32 M x 64</td>
<td>512 Mbit</td>
<td>32 M x 16</td>
<td>4</td>
<td>1</td>
<td>12/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64GB</td>
<td>64 M x 64</td>
<td>1 Gbit</td>
<td>64 M x 16</td>
<td>4</td>
<td>1</td>
<td>15/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1GB</td>
<td>128 M x 64</td>
<td>2 Gbit</td>
<td>128 M x 16</td>
<td>4</td>
<td>1</td>
<td>14/10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2GB</td>
<td>256 M x 64</td>
<td>4 Gbit</td>
<td>256 M x 16</td>
<td>4</td>
<td>1</td>
<td>15/10</td>
<td></td>
</tr>
</tbody>
</table>
Breaking down a Rank

Rank 0

Chip 0

Chip 1

... Chip 7

Data <0:63>

<0:7> <8:15> <56:63>

Breaking down a Chip

Chip 0

8 banks

<0:7> <8> <56> ... <63>

<0:7>
Breaking down a Bank

Example: Transferring a cache block

Physical memory space

Channel 0

DIMM 0

Rank 0

Mapped to

cache block

64B
Example: Transferring a cache block

Physical memory space

Chip 0 Chip 1
<0:7>
<8:15>
<56:63>

Rank 0

Data <0:63>

Row 0 Col 0

64B cache block
Example: Transferring a cache block

Physical memory space

0xFFFF...F

0x40

0x00

64B cache block

8B

8B

Row 0
Col 0

<0:7>

<8:15>

<56:63>

8B

Data <0:63>

Chip 0

Chip 1

Rank 0

Chip 7

Row 0
Col 1

...
A 64B cache block takes 8 I/O cycles to transfer.

During the process, 8 columns are read sequentially.
**DRAM Overview**

- Bits stored in 2-dimensional arrays on chip
- Modern chips have around 4 logical banks on each chip
  - each logical bank physically implemented as many smaller arrays
1-T Memory Cell (DRAM)

- **Write:**
  - 1. Drive bit line
  - 2. Select row

- **Read:**
  - 1. Precharge bit line to Vdd/2
  - 2. Select row
  - 3. Storage cell shares charge with bitlines
    » Very small voltage changes on the bit line
  - 4. Sense (fancy sense amp)
    » Can detect changes of ~1 million electrons
  - 5. Write: restore the value

- **Refresh**
  - 1. Just do a dummy read to every cell.

**SRAM vs. DRAM**

- *Static Random Access Mem.*
  - 6T vs. 1T1C
    - Large (~6-10x)
    - Bitlines driven by transistors
      - Fast (~10x)

- *Dynamic Random Access Mem.*
  - Bits stored as charges on node capacitance (non-restorative)
    - Bit cell loses charge when read
    - Bit cell loses charge over time
  - Must periodically refresh
    - Once every 10s of ms
DRAM Operation: Three Steps

- **Precharge**
  - charges bit lines to known value, required before next row access

- **Row access (RAS)**
  - decode row address, enable addressed row (often multiple Kb in row)
  - Contents of storage cell share charge with bitlines
  - small change in voltage detected by sense amplifiers which latch whole row of bits
  - sense amplifiers drive bitlines full rail to recharge storage cells

- **Column access (CAS)**
  - decode column address to select small number of sense amplifier latches (4, 8, 16, or 32 bits depending on DRAM package)
  - on read, send latched bits out to chip pins
  - on write, change sense amplifier latches. which then charge storage cells to required value
  - can perform multiple column accesses on same row without another row access (burst mode)

DRAM: Memory-Access Protocol

- **5 basic commands**
  - ACTIVATE
  - READ
  - WRITE
  - PRECHARGE
  - REFRESH

- **To reduce pin count, row and column share same address pins**
  - RAS = Row Address Strobe
  - CAS = Column Address Strobe
DRAM Bank Operation

Access Address:
- (Row 0, Column 0)
- (Row 0, Column 1)
- (Row 0, Column 85)
- (Row 1, Column 0)

Row decoder

Rows

Columns

Data

Row address 0

Row address 1

Column address 0

Column address 85

Columns

Rows

Column mux

Row Buffer CONFLICT!

Commands
- ACTIVATE 0
- READ 0
- READ 1
- READ 85
- PRECHARGE
- ACTIVATE 1
- READ 0

DRAM: Basic Operation

- Access to an “open row”
  - No need for ACTIVATE command
  - READ/WRITE to access row buffer

- Access to a “closed row”
  - If another row already active, must first issue PRECHARGE
  - ACTIVATE to open new row
  - READ/WRITE to access row buffer
  - Optional: PRECHARGE after READ/WRITEs finished
**DRAM Read Timing (Example)**

- Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to read: early or late v. CAS

**DRAM: Burst**

- Each READ/WRITE command can transfer multiple words (8 in DDR3)
- DRAM channel clocked faster than DRAM core

- Critical word first?
**DRAM: Banks**

- Modern DRAM chips consist of multiple **banks**
  - Address = (Bank x, Row y, Column z)

- Banks operate independently, but share command/address/data pins
  - Each can have a different row active
  - Can overlap ACTIVATE and PRECHARGE latencies!
    (i.e. READ to bank 0 while ACTIVATING bank 1)

---

**DRAM: Banks**

- Enable concurrent DRAM accesses (overlapping)
  
  ![Diagram](image-url)
2Gb x8 DDR3 Chip [Micron]

Observe: bank organization

Quest for DRAM Performance

1. Fast Page mode
   - Add timing signals that allow repeated accesses to row buffer without another row access time
   - Such a buffer comes naturally, as each array will buffer 1024 to 2048 bits for each access

2. Synchronous DRAM (SDRAM)
   - Add a clock signal to DRAM interface, so that the repeated transfers would not bear overhead to synchronize with DRAM controller

3. Double Data Rate (DDR SDRAM)
   - Transfer data on both the rising edge and falling edge of the DRAM clock signal ⇒ doubling the peak data rate
   - DDR2 lowers power by dropping the voltage from 2.5 to 1.8 volts + offers higher clock rates: up to 400 MHz
   - DDR3 drops to 1.5 volts + higher clock rates: up to 800 MHz
   - DDR4 drops to 1.2 volts + higher clock rates: up to 1600 MHz
1. Fast Page Mode Operation

- **Regular DRAM Organization:**
  - N rows x N column x M-bit
  - Read & Write M-bit at a time
  - Each M-bit access requires a RAS / CAS cycle

- **Fast Page Mode DRAM**
  - N x M “SRAM” to save a row
  - After a row is read into the register
    - Only CAS is needed to access other M-bit blocks on that row
    - RAS_L remains asserted while CAS_L is toggled

```
<table>
<thead>
<tr>
<th></th>
<th>1st M-bit Access</th>
<th>2nd M-bit</th>
<th>3rd M-bit</th>
<th>4th M-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A   Row Address</td>
<td>Cas_L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Col Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

2. SDRAM timing (Single Data Rate)

- Micron 128M-bit dram (using 2Meg×16bit×4bank ver)
  - Row (12 bits), bank (2 bits), column (9 bits)
3. Double-Data Rate (DDR2) DRAM

Row Column Precharge Row

[ Micron, 256Mb DDR2 SDRAM datasheet ]

[ 400Mb/s Data Rate ]

Memory Organizations

<table>
<thead>
<tr>
<th>Production year</th>
<th>Chip size</th>
<th>DRAM Type</th>
<th>Slowest DRAM (ns)</th>
<th>Fastest DRAM (ns)</th>
<th>Column access strobe (CAS)/ data transfer time (ns)</th>
<th>Cycle time (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>64K bit DRAM</td>
<td>180</td>
<td>150</td>
<td>75</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>1983</td>
<td>256K bit DRAM</td>
<td>150</td>
<td>120</td>
<td>50</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>1M bit DRAM</td>
<td>120</td>
<td>100</td>
<td>25</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>4M bit DRAM</td>
<td>100</td>
<td>80</td>
<td>20</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>16M bit DRAM</td>
<td>80</td>
<td>60</td>
<td>15</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>64M bit SDRAM</td>
<td>70</td>
<td>50</td>
<td>12</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>128M bit SDRAM</td>
<td>70</td>
<td>50</td>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>256M bit DDR1</td>
<td>65</td>
<td>45</td>
<td>7</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>512M bit DDR1</td>
<td>60</td>
<td>40</td>
<td>5</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>1G bit DDR2</td>
<td>55</td>
<td>35</td>
<td>5</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>2G bit DDR2</td>
<td>50</td>
<td>30</td>
<td>2.5</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>4G bit DDR3</td>
<td>36</td>
<td>28</td>
<td>1</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>8G bit DDR3</td>
<td>30</td>
<td>24</td>
<td>0.5</td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.13 Times of fast and slow DRAMs vary with each generation. (Cycle time is defined on page 95.) Performance improvement of row access time is about 5% per year. The improvement by a factor of 2 in column access in 1986 accompanied the switch from NMOS DRAMs to CMOS DRAMs. The introduction of various burst transfer modes in the mid-1990s and SDRAMs in the late 1990s has significantly complicated the calculation of access time for blocks of data; we discuss this later in this section when we talk about SDRAM access time and power. The DDR designs are due for introduction in mid-to late 2012. We discuss these various forms of DRAMs in the next few pages.
Memory Organizations

<table>
<thead>
<tr>
<th>Standard</th>
<th>Clock rate (MHz)</th>
<th>M transfers per second</th>
<th>DRAM name</th>
<th>MB/sec/DIMM</th>
<th>DIMM name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR</td>
<td>133</td>
<td>266</td>
<td>DDR266</td>
<td>2128</td>
<td>PC2100</td>
</tr>
<tr>
<td>DDR</td>
<td>150</td>
<td>300</td>
<td>DDR300</td>
<td>2400</td>
<td>PC2400</td>
</tr>
<tr>
<td>DDR</td>
<td>200</td>
<td>400</td>
<td>DDR400</td>
<td>3200</td>
<td>PC3200</td>
</tr>
<tr>
<td>DDR2</td>
<td>266</td>
<td>533</td>
<td>DDR2-533</td>
<td>4264</td>
<td>PC4300</td>
</tr>
<tr>
<td>DDR2</td>
<td>333</td>
<td>667</td>
<td>DDR2-667</td>
<td>5336</td>
<td>PC5300</td>
</tr>
<tr>
<td>DDR2</td>
<td>400</td>
<td>800</td>
<td>DDR2-800</td>
<td>6400</td>
<td>PC6400</td>
</tr>
<tr>
<td>DDR3</td>
<td>533</td>
<td>1066</td>
<td>DDR3-1066</td>
<td>8528</td>
<td>PC8500</td>
</tr>
<tr>
<td>DDR3</td>
<td>666</td>
<td>1333</td>
<td>DDR3-1333</td>
<td>10664</td>
<td>PC10700</td>
</tr>
<tr>
<td>DDR3</td>
<td>800</td>
<td>1600</td>
<td>DDR3-1600</td>
<td>12800</td>
<td>PC12800</td>
</tr>
<tr>
<td>DDR4</td>
<td>1066–1600</td>
<td>2133–3200</td>
<td>DDR4-3200</td>
<td>17,056–25,600</td>
<td>PC25600</td>
</tr>
</tbody>
</table>

Figure 2.14 Clock rates, bandwidth, and names of DDR DRAMs and DIMMs in 2010. Note the numerical relationship between the columns. The third column is twice the second, and the fourth uses the number from the third column in the name of the DRAM chip. The fifth column is eight times the third column, and a rounded version of this number is used in the name of the DIMM. Although not shown in this figure, DDRs also specify latency in clock cycles, as four numbers, which are specified by the DDR standard. For example, DDR3-2000 C1 9 has latencies of 9-9-9-28. What does this mean? With a 1 ns clock (clock cycle is one-half the transfer rate), this indicates 9 ns for row to column address (RAS time), 9 ns for column access to data (CAS time), and a minimum read time of 28 ns. Closing the row takes 9 ns for precharge but happens only when the read from that row is finished. In burst mode, transfers occur on every clock on both edges, when the first RAS and CAS times have elapsed. Furthermore, the precharge in not needed until the entire row is read. DDR4 will be produced in 2012 and is expected to reach clock rates of 1600 MHz in 2014, when DDR5 is expected to take over. The exercises explore these details further.

Graphics Memory

- Achieve 2-5 X bandwidth per DRAM vs. DDR3
  - Wider interfaces (32 vs. 16 bit)
  - Higher clock rate
    - Possible because they are attached via soldering instead of socketted DIMM modules
  - E.g. Samsung GDDR5
    - 2.5GHz, 20 GBps bandwidth
DRAM Power: Not always up, but…

Power @ Maximum Frequency

<table>
<thead>
<tr>
<th>DRAM Type</th>
<th>DDR3</th>
<th>DDR3</th>
<th>DDR3</th>
<th>DDR4</th>
<th>DDR4</th>
<th>DDR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>1.5V</td>
<td>1.35V</td>
<td>1.25V</td>
<td>1.2V</td>
<td>1.1V</td>
<td>1.08V</td>
</tr>
</tbody>
</table>

Power Relative to PC-133

DRAM Modules

DIMMs

CPU

MC

Bus

128

64

Channel 1

Channel 2

Rank 0

Rank 1

DRAM Chip

Arrays

DRAM Cell
DRAM Modules

- DRAM chips have narrow interface (typically x4, x8, x16)
- Multiple chips are put together to form a wide interface
  - DIMM: Dual Inline Memory Module
  - To get a 64-bit DIMM, we need to access 8 chips with 8-bit interfaces
  - Share command/address lines, but not data

Advantages
- Acts like a high-capacity DRAM chip with a wide interface
  - 8x capacity, 8x bandwidth, same latency

Disadvantages
- Granularity: Accesses cannot be smaller than the interface width
  - 8x power

A 64-bit Wide DIMM (physical view)
A 64-bit Wide DIMM (logical view)

DRAM Ranks

- A DIMM may include multiple Ranks
  - A 64-bit DIMM using 8 chips with x16 interfaces has 2 ranks

- Each 64-bit group of chips is called a rank
  - All chips in a rank respond to a single command
  - Different ranks share command/address/data lines
    - Select between ranks with “Chip Select” signal
  - Ranks provide more “banks” across multiple chips (but don’t confuse rank and bank!)
Multiple DIMMs on a Channel

Advantages:
- Enables even higher capacity

Disadvantages:
- Interconnect latency, complexity, and energy get higher
- Addr/Cmd signal integrity is a challenge

Fully Buffered DIMM (FB-DIMM)

- DDR Problem
  - Higher capacity → more DIMMs → lower data rate (multidrop bus)
- FB-DIMM approach: use point to point links
  - introduces an advanced memory buffer (AMB) between memory controller and memory module
  - Serial interface between mem controller and AMB
  - enables an increase to the width of the memory without increasing the pin count of the memory controller
FB-DIMM challenges

- AMB is big, expensive, and power hungry
  - Low volume, so FB-DIMM modules are expensive

- Daisy chain adds significant latency
  - 8 slot FB-DIMM channels are slow

- Requires FB-DIMM memory controller
  - Incompatible with on-chip DDR3 controllers!

• As of Sep 2006, AMD has taken FB-DIMM off their roadmap

• In 2007 it was revealed that major memory manufacturers have no plans to extend FB-DIMM to support DDR3 SDRAM
  - Instead, only registered DIMM for DDR3 SDRAM had been demonstrated
  - In normal registered/buffered memory, only the control lines are buffered whereas in fully buffered memory, the data lines are buffered as well
  - Both FB and “registered” options increase latency and are costly

Intel Scalable Memory Buffer

- High-speed serial link from CPU to SMB
  - Two DDR3 channels behind SMB (2 slots per channel)
  - Can use commodity DDR3 modules
  - Mitigates pin-count on CPU

- On-chip MC manages DRAM access protocol

- Jury still out on the right design
DRAM Channels

Channel: a set of DIMMs in series
- All DIMMs get the same command, one of the ranks replies

System options
- Single channel system
- Multiple dependent (lock-step) channels
  - Single controller with wider interface (faster cache line refill!)
  - Sometimes called “Gang Mode”
  - Only works if DIMMs are identical (organization, timing)
- Multiple independent channels
  - Requires multiple controllers

Tradeoffs
- Cost: pins, wires, controller
- Benefit: higher bandwidth, capacity, flexibility
**DRAM Channel Options**

**Lock-step**
- CPU
- MC

**Independent**
- CPU
- MC

**Multi-CPU (old school)**
- CPU
- MC

- External MC adds latency
- Capacity doesn’t grow w/ # of CPUs
NUMA Topology (modern)

- Capacity grows w/ # of CPUs
- NUMA: “Non-uniform Memory Access”

Memory Controller

- Diagram showing the interaction between CPU, Memory Controller (MC), DIMMs, and DRAM chips.
DRAM: Timing Constraints

- Memory controller must respect physical device characteristics
  - tRCD = Row to Column command delay
    - How long it takes row to get to sense amps
  - tCAS = Time between column command and data out
  - tCCD = Time between column commands
    - Rate that you can pipeline column commands
  - tRP = Time to precharge DRAM array
  - tRAS = Time between RAS and data restoration in DRAM array (minimum time a row must be open)
  - tRC = tRAS + tRP = Row “cycle” time
    - Minimum time between accesses to different rows

- There are dozens of these...
  - tWTR = Write to read delay
  - tWR = Time from end of last write to PRECHARGE
  - tFAW = Four ACTIVATE window (limits current surge)

- Makes performance analysis, memory controller design difficult

- Datasheets for DRAM devices freely available
Latency Components: Basic DRAM Operation

- CPU → controller transfer time
- Controller latency
  - Queuing & scheduling delay at the controller
  - Access converted to basic commands
- DRAM bank latency
  - tCAS is row is “open” OR
  - tRCD + tCAS if array precharged OR
  - tRP + tRCD + tCAS (worst case: tRC + tRCD + tCAS)
- DRAM data transfer time
  - BurstLen / (MT/s) 
  - Controller → CPU transfer time

DRAM Addressing

LD R1, Mem[foo]

0xBA5EBA77

Miss!
DRAM Controller Functionality

- Translate memory requests into DRAM command sequences
  - Map “Physical Address” to DRAM Address
  - Obey timing constraints of DRAM, arbitrate resource conflicts (i.e. bank, channel)

- Buffer and schedule requests to improve performance
  - Row-buffer management and re-ordering

- Ensure correct operation of DRAM (refresh)

- Manage power consumption and thermals in DRAM
  - Turn on/off DRAM chips, manage power modes

A Modern DRAM Controller
Row Buffer Management Policies

- **Open row**
  - Keep the row open after an access
  - Pro: Next access might need the same row → row hit
  - Con: Next access might need a different row → row conflict, wasted energy

- **Closed row**
  - Close the row after an access
    - (if no other requests already in the request buffer need the same row)
  - Pro: Next access might need a different row → avoid a row conflict
  - Con: Next access might need the same row → extra activate latency

- **Adaptive policies**
  - Predict whether or not the next access to the bank will be to the same row

DRAM Controller Scheduling Policies (I)

- **FCFS (first come first served)**
  - Oldest request first

- **FR-FCFS (first ready, first come first served)**
  - 1. Row-hit first
  - 2. Oldest first
  - Goal: Maximize row buffer hit rate → maximize DRAM throughput
DRAM Controller Scheduling Policies (II)

- A scheduling policy is a prioritization order

Prioritization can be based on
- Request age
- Row buffer hit/miss status
- Request type (prefetch, read, write)
- Requestor type (load miss or store miss)
- Request criticality
  - Oldest miss in the core?
  - How many instructions in core are dependent on it?

DRAM Refresh (I)

- DRAM capacitor charge leaks over time

The memory controller needs to read each row periodically to restore the charge
- Activate + precharge each row every $N$ ms
- Typical $N = 64$ ms

Implications on performance?
- DRAM bank unavailable while refreshed
- Long pause times: If we refresh all rows in burst, every 64ms the DRAM will be unavailable until refresh ends
DRAM Refresh (II)

- Distributed refresh eliminates long pause times
- How else we can reduce the effect of refresh on performance?
  - Can we reduce the number of refreshes?

DRAM Controllers are Difficult to Design

- Need to obey DRAM timing constraints for correctness
  - There are many (50+) timing constraints in DRAM
- Need to keep track of many resources to prevent conflicts
  - Channels, banks, ranks, data bus, address bus, row buffers
- Need to handle DRAM refresh
- Need to optimize for performance (in the presence of constraints)
  - Reordering is not simple
  - Predicting the future?
DRAM Power Management

- DRAM chips have power modes
- Idea: When not accessing a chip power it down

- Power states
  - Active (highest power)
  - All banks idle (i.e. precharged)
  - Power-down
  - Self-refresh (lowest power)

- State transitions incur latency during which the chip cannot be accessed

DRAM Reliability

- DRAMs are susceptible to soft and hard errors
- Dynamic errors can be
  - detected by parity bits
    - usually 1 parity bit per 8 bits of data
  - detected and fixed by the use of Error Correcting Codes (ECCs)
    - E.g. SECDED Hamming code can detect two errors and correct a single error with a cost of 8 bits of overhead per 64 data bits

- In very large systems, the possibility of multiple errors as well as complete failure of a single memory chip becomes significant
  - Chipkill was introduced by IBM to solve this problem
  - Similar in nature to the RAID approach used for disks
  - Chipkill distributes data and ECC information, so that the complete failure of a single memory chip can be handled by supporting the reconstruction of the missing data from the remaining memory chips
  - IBM and SUN servers and Google Clusters use it
  - Intel calls their version SDDC
Looking Forward

- Continued slowdown in both density and access time of DRAMs → new DRAM that does not require a capacitor?
  - Z-RAM prototype from Hynix
- MRAMs → use magnetic storage of data; nonvolatile
- PRAMs → phase change RAMs (aka PCRAM, PCME)
  - use a glass that can be changed between amorphous and crystalline states; nonvolatile