Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future
Main Memory Background

- Performance of Main Memory:
  - **Latency**: affects cache Miss Penalty
  - **Bandwidth**: I/O & Large Block Miss Penalty (L2 or L3)

- 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

- While a lot is done in terms of cache organization (to reduce processor-DRAM performance gap), innovations in main memory is needed as well

---

Memory subsystem organization

- Memory subsystem organization

  - Channel
  - DIMM
  - Rank
  - Chip
  - Bank
  - Row/Column
Memory subsystem

“Channel”  DIMM (Dual in-line memory module)

Processor

Memory channel  Memory channel

Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view

Front of DIMM  Back of DIMM

Serial Presence Detect (SPD)
- Stored in EEPROM on module
- Has info to configure mem controllers
Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view

Front of DIMM

Rank 0: collection of 8 chips

Back of DIMM

Rank 1

Rank

Rank 0 (Front) -> Rank 1 (Back)

<0:63>

Addr/Cmd CS <0:1>

Data <0:63>

Memory channel
Breaking down a Rank

Rank 0

Chip 0  Chip 1  ...  Chip 7

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

Data <0:63>

Breaking down a Chip

Chip 0

8 banks

<0:7>  <8:7>  <56:7>

Page 5
Breaking down a Bank

Example: Transferring a cache block
Example: Transferring a cache block

Physical memory space

Chip 0 Chip 1 Chip 7

Rank 0

Data <0:63>

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

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

Row 0 Col 0

Data <0:63>

64B cache block
Example: Transferring a cache block

Physical memory space

Chip 0  Chip 1  Rank 0  Chip 7
Row 0  Col 0

Row 0  Col 1

Data <0:63>

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

64B cache block
Example: Transferring a cache block

A 64B cache block takes 8 I/O cycles to transfer.
During the process, 8 columns are read sequentially.
Outline

• Main memory organization
• DRAM basics
• Quest for DRAM Performance
• Memory controller
• Future

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. Cell and bit line share charges
    - Minute 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

*The primary difference between different memory types is the bit cell*

- **SRAM Cell**
  - Larger cell $\Rightarrow$ lower density, higher cost/bit
  - No dissipation
  - Read non-destructive
  - No refresh required
  - Simple read $\Rightarrow$ faster access
  - Standard IC process $\Rightarrow$ natural for integration with logic

- **DRAM Cell**
  - Smaller cell $\Rightarrow$ higher density, lower cost/bit
  - Needs periodic refresh, and refresh after read
  - Complex read $\Rightarrow$ longer access time
  - Special IC process $\Rightarrow$ difficult to integrate with logic
**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, charge 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: 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 Bank Operation

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

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

**Row Buffer CONFLICT!**

- Row decoder
- Columns
- Rows
- Column address 0
- Column address 85
- Column mux
- Data
Outline

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future
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 \(\Rightarrow\) 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 1-1.2 volts + higher clock rates: up to 1600 MHz

Memory Optimizations

<table>
<thead>
<tr>
<th>Production year</th>
<th>Chip size</th>
<th>DRAM type</th>
<th>RAS time (ns)</th>
<th>CAS time (ns)</th>
<th>Total (ns) Precharge needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>256M bit</td>
<td>DDR1</td>
<td>21</td>
<td>21</td>
<td>42</td>
</tr>
<tr>
<td>2002</td>
<td>512M bit</td>
<td>DDR1</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>2004</td>
<td>1G bit</td>
<td>DDR2</td>
<td>15</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>2006</td>
<td>2G bit</td>
<td>DDR2</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>2010</td>
<td>4G bit</td>
<td>DDR3</td>
<td>13</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>2016</td>
<td>8G bit</td>
<td>DDR4</td>
<td>13</td>
<td>13</td>
<td>26</td>
</tr>
</tbody>
</table>
Memory Optimizations

<table>
<thead>
<tr>
<th>Standard</th>
<th>I/O clock rate</th>
<th>M transfers/s</th>
<th>DRAM name</th>
<th>MiB/s/DIMM</th>
<th>DIMM name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR1</td>
<td>133</td>
<td>266</td>
<td>DDR26</td>
<td>2128</td>
<td>PC2100</td>
</tr>
<tr>
<td>DDR1</td>
<td>150</td>
<td>300</td>
<td>DDR30</td>
<td>2400</td>
<td>PC2400</td>
</tr>
<tr>
<td>DDR1</td>
<td>200</td>
<td>400</td>
<td>DDR40</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>10.664</td>
<td>PC10700</td>
</tr>
<tr>
<td>DDR3</td>
<td>800</td>
<td>1600</td>
<td>DDR3-1600</td>
<td>12.800</td>
<td>PC12800</td>
</tr>
<tr>
<td>DDR4</td>
<td>1333</td>
<td>2666</td>
<td>DDR4-2666</td>
<td>21.300</td>
<td>PC21300</td>
</tr>
</tbody>
</table>

• 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 on 32-bit bus (160GBps on 256-bit bus)
Stacked/Embedded DRAMs

- Stacked DRAMs in same package as processor
  - High Bandwidth Memory (HBM)
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)

Increasing Capacity: 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
**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**
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

Independent
Multi-CPU (old school)

- External MC adds latency
- Capacity doesn’t grow w/ # of CPUs

NUMA Topology (modern)

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

• Main memory organization
• DRAM basics
• Quest for DRAM Performance
• Memory controller
• Future

Memory Controller

[Diagram showing the relationship between CPU, memory controller (MC), DIMMs, and DRAM chips]
DRAM Controller Functionality

• Obey timing constraints of DRAM
• Map physical addresses to DRAM addresses
• Row buffer management policies
• DRAM request scheduling
• DRAM refresh strategies
• DRAM power management
• DRAM reliability

→ DRAM controllers are difficult to design!

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
- DRAM data transfer time
  - BurstLen / (MT/s)
- Controller → CPU transfer time

500 MHz DDR = 1000 MT/s
DRAM Addressing

- Memory Controller has a significant impact on access latency

`LD R1, Mem[foo]`

![Diagram of DRAM addressing](image_url)

DRAM: Timing Constraints

- Memory controller must respect physical device characteristics
  - \( t_{RCD} \) = Row to Column command delay
    - How long it takes row to get to sense amps
  - \( t_{C\text{AS}} \) = Time between column command and data out
  - \( t_{CCD} \) = Time between column commands
    - Rate that you can pipeline column commands
  - \( t_{RP} \) = Time to precharge DRAM array
  - \( t_{\text{RAS}} \) = Time between RAS and data restoration in DRAM array (minimum time a row must be open)
  - \( t_{\text{RC}} \) = \( t_{\text{RAS}} + t_{RP} \) = Row “cycle” time
    - Minimum time between accesses to different rows
DRAM: Timing Constraints

- 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

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?
Problem: Memory Request Interference

- Problem: Threads share the memory system, but memory system does not distinguish threads' requests
  - Memory system algorithms thread-unaware and thread-unfair
- Existing memory systems
  - Free-for-all, demand-based sharing of the memory system
  - Aggressive threads can deny service to others
  - Do not try to reduce or control inter-thread interference
- Solution #1: Smart resources: Design each shared resource to have a configurable fairness/QoS mechanism
  - Fair/QoS-aware memory schedulers, interconnects, caches, arbiters
- Solution #2: Dumb resources: Keep each resource free-for-all, but control access to memory system at the cores/sources
  - Fairness via Source Throttling; Estimate thread slowdowns in the entire system and throttle cores that slow down others; Coordinated Prefetcher Throttling
A Modern DRAM Controller

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 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
Mobile DRAM characteristics

- Same core as DDR3 devices
  - Same capacity per device, same access latency, same active currents
- IO interface optimized for very low static power
  - Including faster powerdown modes, no termination
- Same chip bandwidth
  - Wider interface operating at slower clock rate

<table>
<thead>
<tr>
<th>Technology Parameter</th>
<th>DDR3</th>
<th>LPDDR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing (tCAS, tRAS, tRC)</td>
<td>15, 38, 50ns</td>
<td>15, 42, 57ns</td>
</tr>
<tr>
<td>Active current (Read, Write)</td>
<td>180, 185mA</td>
<td>210, 175mA</td>
</tr>
<tr>
<td>Idle current (Powerdown, Standby)</td>
<td>35, 45mA</td>
<td>1.6, 23mA</td>
</tr>
<tr>
<td>Powerdown exit latency</td>
<td>24ns</td>
<td>7.5ns</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>1.5V</td>
<td>1.2V</td>
</tr>
<tr>
<td>Typical operating frequency</td>
<td>800MHz</td>
<td>400MHz</td>
</tr>
<tr>
<td>Device width</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

LPDDR2 advantages

- Energy proportional
- Energy efficient ~ 40pJ/bit
  - 2x to 5x reduction over DDR3
**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
  - 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
  - Often combined with dynamic bit steering (Similar to RAID)
  - IBM and SUN servers and Google Clusters use it
  - Intel calls their version SDDC

**Outline**

- Main memory organization
- DRAM basics
- Quest for DRAM Performance
- Memory controller
- Future
Looking Forward: Future of DRAM

• Molecular RAMs → e.g., use special compounds such as porphyrin-based polymers which are capable of storing electric charge
  – Once a certain voltage threshold is achieved the material oxidizes, releasing an electric charge. The process is reversible, in effect creating an electric capacitor.

• PRAMs → phase change RAMs
  – use a glass that can be changed between amorphous and crystalline states
  – nonvolatile

Micron Hybrid Memory Cube (HMC)

• 3D-stacked device with memory+logic
• High capacity, low power, high bandwidth

  • All links are between host CPU and HMC logic layer
  • Maximum bandwidth per GB capacity

Figure from: T. Pawlowski, HotChips 2011
Recent News

• Crossbar memory

• Conferences, companies
  – http://isscc.org/index.html
  – http://www.monolithic3d.com/

• ISSCC (memory) trends