MEMORY SYSTEMS

Mahdi Nazm Bojnordi

Assistant Professor
School of Computing
University of Utah
Main Memory System

- A critical component of all computing systems
  - server, mobile, embedded, desktop, sensor

- Must scale to maintain performance growth
  - size, technology, efficiency, cost, and control algorithms
Why Main Memory is Important?

- **Shared resource**
- **Multiple applications running on different processor cores**
- **Different objectives and requirements**
- **Highly contented resource**

Complex control policies and microarchitectures are required.
Scalability Challenges

- Increasing need for memory capacity, bandwidth, and quality-of-service maintenance
  - increasing number of cores (multicores)
  - increasing demand for data (big data processing)
  - cloud computing, GPUs, mobile (consolidation)

- AMD Barcelona
  - 8 cores

- Intel Core i7
  - 8+1 cores

- IBM Cell BE
  - 8+1 cores

- IBM POWER7
  - 8 cores

- Sun Niagara II
  - 8 cores

- Nvidia Fermi
  - 448 “cores”

- Intel SCC
  - 48 cores, networked

- Tilera TILE Gx
  - 100 cores, networked
CPU-DRAM Gap

- Core count doubling ~ every 2 years
- DRAM DIMM capacity doubling ~ every 3 years

Memory capacity per core expected to drop by 30% every two years

[Lim’09]
DRAM: Design Challenges

- Main memory energy/power is a key system design concern
  - Energy spent in off-chip memory hierarchy is about 40-50% [Lefurgy’03]
  - DRAM consumes power even when not used
    - periodic refresh

- DRAM technology scaling is ending
  - stops gaining higher capacity, lower cost, lower energy
DRAM: Logical Organization

- Five DRAM coordinates
  - Channel, rank, bank, row, column

[Kim’12]
DRAM: Physical Organization

“Channel”

DIMM (Dual in-line memory module)

Processor

Memory channel

Memory channel
DIMM Structure

DIMM (Dual in-line memory module)
DIMM Structure

DIMM (Dual in-line memory module)

Rank 0: collection of 8 chips
Rank Organization

Address/Command (Addr/Cmd) <0:1> CS <0:1> Memory Channel

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

Data <0:63>
Rank Breakdown
DRAM DIMM and Rank

- Multiple chips operated together to form a wide interface
  - All chips within a rank are controlled at the same time
  - Respond to a single command
  - Share address and command; different data bits

- A DRAM module consists of one or more ranks
  - e.g., DIMM (dual inline memory module)
  - If we have chips with 8-bit interface, to read 8 bytes in a single access, use 8 chips in a DIMM
Chip Structure
A “DRAM row” is also called a “DRAM page” “Sense amplifiers” also called “row buffer”
DRAM Page Access

- Access to a closed row
  - Activate command opens row (placed into row buffer)
  - Read/write command reads/writes column in the row buffer

- Precharge command closes the row and prepares the bank for next access

- Access to an “open row”
  - No need for activate command
DRAM Page Access

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

Row address 0

Row decoder

Columns

Row 1

Row Buffer

CONFLICT!

Column address 05

Column mux

Data

[ref: Mutlu]
DRAM: Parallel Access

- DRAM subsystem comprises multiple banks
  - Organized under independent channels and ranks

Before: No Overlapping
Assuming accesses to different DRAM rows

After: Overlapped Accesses
Assuming no bank conflicts
DRAM Controller

- Ensure correct operation of DRAM (refresh and timing)
- Service DRAM requests while obeying timing constraints of DRAM chips
  - Constraints: resource conflicts (bank, bus, channel), minimum write-to-read delays
  - Translate requests to DRAM command sequences
- Buffer and schedule requests to improve performance
  - Reordering, row-buffer, bank, rank, bus management
- Manage power consumption and thermals in DRAM
  - Turn on/off DRAM chips, manage power modes
DRAM Controller

Ensuring DDRx timing constraints

Figure 5. Three Phases of DRAM Access

Table 2. Timing Constraints (DDR3-1066) [43]

<table>
<thead>
<tr>
<th>Phase</th>
<th>Commands</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACT → READ, ACT → WRITE</td>
<td>tRC</td>
<td>15ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tRP</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>READ → data, WRITE → data</td>
<td>tCL</td>
<td>15ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tCWL</td>
<td>11.25ns</td>
</tr>
<tr>
<td></td>
<td>data burst</td>
<td>tBL</td>
<td>7.5ns</td>
</tr>
<tr>
<td>3</td>
<td>PRE → ACT</td>
<td>tRP</td>
<td>15ns</td>
</tr>
<tr>
<td>1 &amp; 3</td>
<td>ACT → ACT</td>
<td>tRC</td>
<td>52.5ns</td>
</tr>
<tr>
<td></td>
<td>(tRAS+tRP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Kim’12]