DRAM CONTROLLER

Mahdi Nazm Bojnordi
Assistant Professor
School of Computing
University of Utah
Recall: DRAM Operations

- Main DRAM operations are:
  - **Precharge** bitlines to prepare subarray for activating a wordline
  - **Activate** a row by connecting DRAM cells to the bitlines and start sensing
  - **Read** the contents of a data block from the row buffer
  - **Write** new contents for data block into the row buffer
  - **Refresh** DRAM cells
    - can be done through a precharge followed by an activate
DRAM Control

- DRAM chips have no intelligence
  - An external controller dictates operations
  - Modern controllers are integrated on CPU

- Basic DRAM timings are
  - $t_{\text{CAS}}$: column access strobe (RD $\rightarrow$ DATA)
  - $t_{\text{RAS}}$: row active strobe (ACT $\rightarrow$ PRE)
  - $t_{\text{RP}}$: row precharge (PRE $\rightarrow$ ACT)
  - $t_{\text{RC}}$: row cycle (ACT $\rightarrow$ PRE $\rightarrow$ ACT)
  - $t_{\text{RCD}}$: row to column delay (ACT $\rightarrow$ RD/WT)
DRAM Timing Example

- **Access time**
  - Row hit: $t_{CAS}$
  - Row empty: $t_{RCD} + t_{CAS}$
  - Row conflict: $t_{RP} + t_{RCD} + t_{CAS}$

![Diagram of DRAM timing example with labels for Rd, Pr, X, Y, and Data Array with requests RD B and RD A](image)
Memory Channels

- Memory channels provide fully parallel accesses
  - Separate data, control, and address buses

Not scalable due to pin overhead
Memory Banks

- Memory banks provide parallel operations
  - Shared data, control, and address buses
- The goal is to keep the data bus fully utilized

Shorter data transfer time to reduce bus conflicts
Double data rate vs. single rate
DRAM Organization

- DRAM channels are independently accessed through dedicated data, address, and command buses
  - Physically broken down into DIMMs (dual in-line memory modules)
  - Logically divided into ranks, which are a collection of DRAM chips responding to the same memory request
Memory Controller

- Memory controller connects CPU and DRAM
- Receives requests after cache misses in LLC
  - Possibly originating from multiple cores
- Complicated piece of hardware, handles:
  - DRAM Refresh management
  - Command scheduling
  - Row-Buffer Management Policies
  - Address Mapping Schemes
DRAM Refresh Management

- DRAM requires the cells’ contents to be read and written periodically
  - **Burst refresh:** refresh all of the cells each time
    - Simple control mechanism
  - **Distributed refresh:** a group of cells are refreshed
    - Avoid blocking memory for a long time
- Recently accessed rows need not to be refreshed
  - **Smart refresh**

![Diagram of DRAM Refresh Management](image)
Command Scheduling

- Write buffering
  - Writes can wait until reads are done

- Controller queues DRAM commands
  - Usually into per-bank queues
  - Allows easily reordering ops. meant for same bank

- Common policies:
  - First-Come-First-Served (FCFS)
  - First-Ready First-Come-First-Served (FR-FCFS)
Command Scheduling

- **First-Come-First-Served**
  - Oldest request first

- **First-Ready First-Come-First-Served**
  - Prioritize column changes over row changes
  - Skip over older conflicting requests
  - Find row hits (on queued requests)
    - Find oldest
    - If no conflicts with in-progress request ⇒ good
    - Otherwise (if conflicts), try next oldest
## FCFS vs. FR-FCFS

- **FCFS**
  
  READ(B0,R0,C0) READ(B0,R1,C0) READ(B0,R0,C1)

- **FR-FCFS**

  ![Diagram showing FR-FCFS scheduling](#)

  The diagram shows the sequence of commands and addresses for FCFS and FR-FCFS, highlighting the savings achieved by FR-FCFS.
Row-Buffer Management Policies

- **Open-page Policy**
  - After access, keep page in DRAM row buffer
  - If access to different page, must close old one first
    - Good if lots of locality

- **Close-page Policy**
  - After access, immediately close page in DRAM row buffer
  - If access to different page, old one already closed
    - Good if no locality (random access)