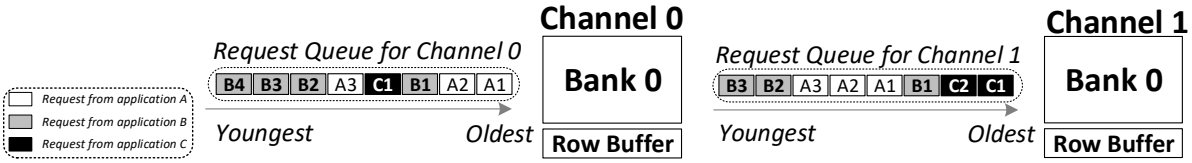
4 Memory Scheduling [40 points]

4.1 Basics and Assumptions

To serve a memory request, the memory controller issues one or multiple DRAM commands to access data from a bank. There are four different DRAM commands as discussed in class.

- `ACTIVATE`: Loads the row (that needs to be accessed) into the bank's row-buffer. This is called opening a row. **(Latency: 15ns)**
- `PRECHARGE`: Prepares the bank for the next access by closing the row in the bank (and making the row buffer empty). **(Latency: 15ns)**
- `READ/WRITE`: Accesses data from the row-buffer. **(Latency: 15ns)**

The diagrams below show the snapshots of memory controller's *request queues* at time 0, i.e., t_0, when applications A, B, and C are executed together on a multi-core processor. Each application runs on a separate core but shares the memory subsystem with other applications. Each request is color-coded to denote the application to which it belongs. Additionally, each request is annotated with a number that shows the order of the request among the set of enqueued requests of the application to which it belongs. For example, A3 means that this is the third request from application A enqueued in the request queue. Assume all memory requests are reads and a read request is considered to be served when the `READ` command is complete (i.e., 15 ns after the request's `READ` command is issued).



Assume also the following:

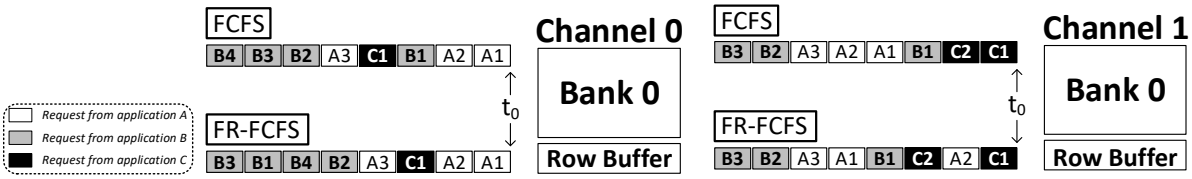
- The memory system has two DRAM channels, one DRAM bank per channel, and four rows per bank.
- All the row-buffers are closed (i.e., empty) at time 0.
- All applications start to stall at time 0 because of memory.
- No additional requests from any of the applications arrive at the memory controller.
- An application (A, B, or C) is considered to be stalled until *all* of its memory requests (across all the request buffers) have been served.

4.2 Problem Specification

The below table shows the stall time of applications A, B, and C with the FCFS (First-Come, First-Served) and FR-FCFS (First-Ready, First-Come, First-Served) scheduling policies.

Scheduling	Application A	Application B	Application C
FCFS	195 ns	285 ns	135 ns
FR-FCFS	135 ns	225 ns	90 ns

The diagrams below show the scheduling order of requests for Channel 0 and Channel 1 with the FCFS and FR-FCFS scheduling policies.



What are the numbers of row hits and row misses for each DRAM bank with either of the scheduling policies? Show your work.

Channel 0, hits:	FCFS: 3, FR-FCFS: 5
Channel 0, misses:	FCFS: 5, FR-FCFS: 3
Channel 1, hits:	FCFS: 2, FR-FCFS: 4
Channel 1, misses:	FCFS: 6, FR-FCFS: 4

Extra space for explanation (use only if needed):

To calculate the number of hits and misses we should consider the following facts:

- The first request of each channel will be always a row-buffer miss and it requires one `ACTIVATE` and one `READ` command which lead to a 30 ns delay.
- For all requests in each channel, except for the first one, a row-buffer miss requires one `PRECHARGE`, one `ACTIVATE`, and one `READ` command which lead to a 45 ns delay.
- A row-buffer hit requires one `READ` command which leads to a 15 ns delay.
- The stall time of each application is equal to the maximum service time of its requests in both Channel 0 and Channel 1.
- When using the FR-FCFS policy, the requests will be reordered to exploit row buffer locality. For example, the B1 request in Channel 0 is reordered in FR-FCFS with respect to FCFS and executed after C1, A3, B2, and B4 requests. This means that A2, C1, A3, B2, and B4 are all accessing the same row.