Maxwell Jung 706531056
June 15, 2025
ECE 209AS L1
Advanced and Secure Computer Architecture

CA4 Report

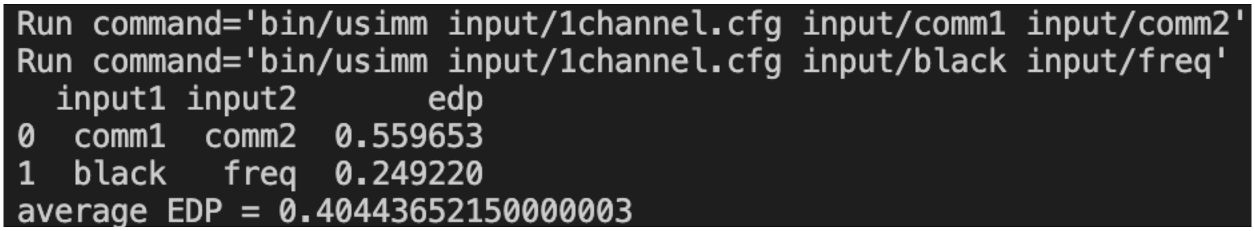
Objective

The goal of this Computer Assignment was to implement a DRAM scheduling policy that achieves a lower energy delay product (EDP) than the baseline First Come First Serve (FCFS) policy. The scheduling policy I chose is First Ready First Come First Serve policy augmented with a very simple precharge logic that will be referred to as FRFCFS-P.

Results

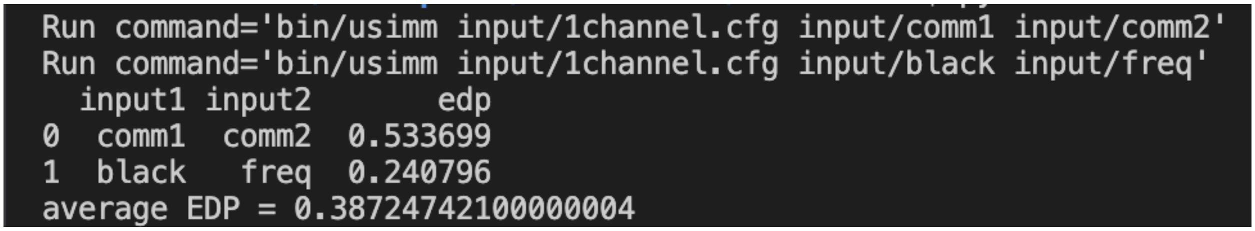
Below screenshots show the average EDP of the FCFS policy and my FRFCFS-P policy. My FRFCFS-P policy achieves ~4.25% lower average EDP than the baseline FCFS policy. I also noticed that running the benchmarks on MacOS results in higher EDP than linux. On MacOS, the baseline FCFS policy results in 0.471 EDP and my FRFCFS-P policy results in 0.439 EDP. The final results uploaded to the leaderboard were achieved with a Ubuntu Linux docker container.

FCFS results (baseline) on Linux



```
Run command='bin/usimm input/1channel.cfg input/comm1 input/comm2'
Run command='bin/usimm input/1channel.cfg input/black input/freq'
  input1 input2       edp
0  comm1  comm2  0.559653
1  black   freq  0.249220
average EDP = 0.40443652150000003
```

FRFCFS-P results on Linux



```
Run command='bin/usimm input/1channel.cfg input/comm1 input/comm2'
Run command='bin/usimm input/1channel.cfg input/black input/freq'
  input1 input2       edp
0  comm1  comm2  0.533699
1  black   freq  0.240796
average EDP = 0.38724742100000004
```

Methodology

FRFCFS-P is basically FRFCFS policy learned in lecture augmented with a very simple precharge logic. For a given read/write queue, the scheduler picks the first memory request that results in a row buffer hit. In the simulator, this is achieved by checking if the next_command of the request is equal to column read or write. If none of the requests in the queue result in a row buffer hit, the first request in the queue is chosen as per FCFS policy.

For precharging memory banks, the bank is precharged whenever the row buffer is reused at least once. This is achieved by keeping a counter called row_hit_count for each memory bank that increments whenever a memory request with row buffer hit is chosen. If there were no requests with row buffer hit, the counter is reset to 0 to indicate that the row is discarded and a new row is activated. Contrary to my intuition, raising the row_hit_count threshold condition for precharging the memory bank actually worsens average EDP. I expected keeping the row open as long as possible would minimize unnecessary energy consumption, but the benchmark results prove otherwise.

Storage Overhead

The only storage overhead in FRFCFS-P policy is the row_hit_count data structure that counts the number of row buffer hits for each memory bank. Since there are NUM_CHANNELS * NUM_RANKS * NUM_BANKS = 16*16*32 = 8192 memory banks in the system and each counter is a 4 byte integer, the total storage overhead is 32 KiB. However, because the threshold value I selected for row buffer hit is 1, each counter value can be substituted with a 1-bit boolean. This decreases the storage overhead by a factor of 32 to a total of 1 KiB.