209AS CA3

Memory Scheduling Policy- Nathan Portillo, Benjamin Cruz

[1]

***Abstract*—The aim of this paper is to introduce an implementation of RLDP and FCFS**

I. Introduction

This paper is to introduce a method that utilizes Row Buffer Locality-based Drain Policy(RLDP) but still uses FCFS as a fall-back method. This is done in an effort to keep the fairness of the scheduling system while also trying to exploit RLDP's more efficient handling of energy and latency.

II. RLDP Implementation

Other implementations of RLDP have found that this approach can yield extremely positive results when it comes to reducing EDP. So, this was going to be the foundation for this approach. RLDP attempts to prioritize requests that access a currently open row; this may help reduce the number of row activations, saving us the aforementioned energy and latency issues.

The implementation of RLDP is a somewhat 'naive' approach. We build upon the current scheduler code similarly, where we check the read queue for a given channel and see if the read request is issuable. The difference is that if it is, we check if that request hits the currently open row. If it does, the current request is 'promoted' or made the priority. We then issue the prioritized request.

III. FCFS Implementation

Like in previous assignments, we thought a hybrid approach may be a viable solution. This implementation is simple; if we never hit an open row, and therefore never prioritize, we First-Come, First-Served and issue the first issuable read request found.

IV. Trade-Offs and Results

Finally, the implementation of the RLDP and FCFS yielded a somewhat good result compared to the baseline. The established baseline, .475 was reduced to .395 or a 17% decrease in the EDP.

However, this decrease in EDP has its disadvantages. We lose the fairness aspect that FCFS offers. Although we try to retain it by keeping FCFS as a backup, the inherent use of RLDP neglects that. Additionally, RLDP has an increase in memory consumption compared to FCFS. We now have to keep track of open rows and their prioritization.

Results:

```
Total memory system power = 4.775661 W
Miscellaneous system power = 10 W  # Processor un
Processor core power = 9.356592 W  # Assuming tha
Total system power = 24.132254 W # Sum of the pre
Energy Delay product (EDP) = 0.550201058 J.s
```

```
Total memory system power = 3.281376 W
Miscellaneous system power = 10 W  # Processor un
Processor core power = 9.788097 W  # Assuming tha
Total system power = 23.069473 W # Sum of the pre
Energy Delay product (EDP) = 0.241698295 J.s
```

[1]