CS2610 Lab #6: DRAM Row Buffer Management

Aditya C (CS20B003)
Kruthic Vignesh M (CS20B045)

Objective:

- To implement an adaptive-page DRAM row-buffer management policy and compare it with open-page and closed-page policies in run-cycles & read/write latencies.
- To compare DRAM performances of the two address mappings for the best policy among adaptive, open and closed page policies.

Implementing Adaptive DRAM row buffer management policy:

A 4-bit saturation counter was implemented, whose value is updated as described in the problem statement (decrement if page-hit on closed-page policy, increment if page-miss on open-page policy). If the counter goes below LOW_COUNT (= 6), the open-page policy is used. If the counter goes above HIGH_COUNT (= 11), the closed-page policy is used.

A 3D array called prev_active_row was introduced, to keep track of the last accessed row in each bank. Whenever a COL_WRITE_CMD or COL_READ_CMD is issued, the currently accessed row and the prev_active_row of that bank are compared; if they match then it is a hit, else it is a miss. After comparison, the prev_active_row of that bank is updated.

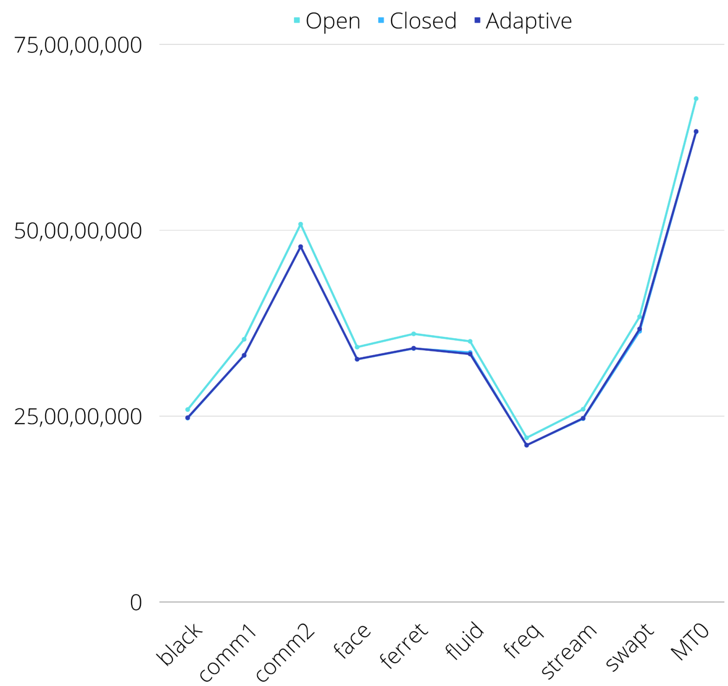
The implementations of the open and closed paging policy were derived from the simulator.

Open-Paging Policy: We keep the last used row in the row_buffer itself. If the next read/write query also deals with the same row, then we don't need to do PRE or ACT, we can directly do COL_READ/COL_WRITE. In case it deals with a different row, we have to do PRE + ACT + COL_READ/COL_WRITE.

Closed-Paging Policy: We close a row once we access it using COL_WRITE/COL_READ, irrespective of which row the next query deals with. In this case, for any query, the average latency is ACT + COL_READ/COL_WRITE. The pre-charging is done during access of other rows and hence doesn't contribute to the average time latency of the closed-paging policy.

Number of cycles for each paging policy: (in million cycles):

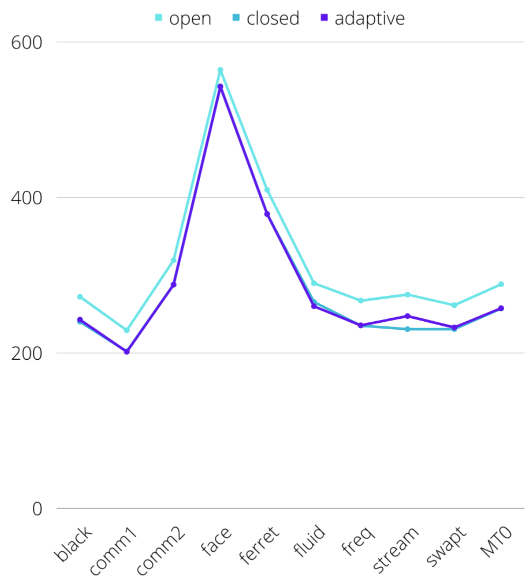
Trace Name	Open	Closed	Adaptive	% Accesses via Open Policy
black	258.87	247.40	247.95	6.659005
comm1	353.40	331.65	331.88	8.059191
comm2	508.19	477.86	478.01	1.181465
face	342.86	326.45	326.63	7.312764
ferret	360.68	341.02	341.33	4.430303
fluid	350.62	335.59	333.52	5.791407
freq	220.88	210.92	210.98	6.726139
stream	259.31	246.61	247.24	6.739906
swapt	383.74	364.16	367.35	6.747916
MT0	677.08	632.51	632.94	1.802428



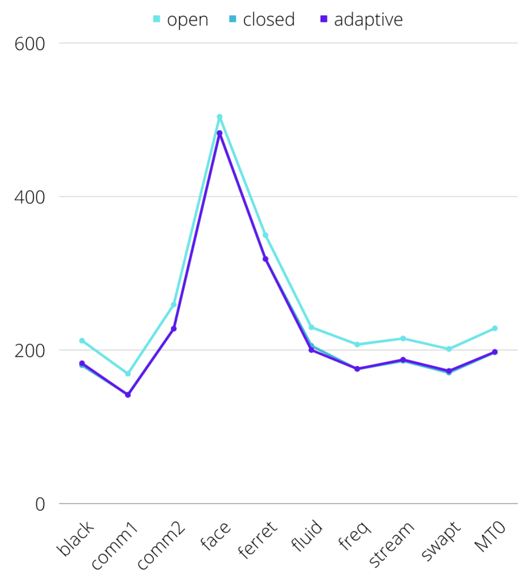
Read-Write Latencies for different paging policies:

	Read Latency			Write Latency		
Trace Name	**Open**	**Closed**	**Adaptive**	**Open**	**Closed**	**Adaptive**
black	272.30957	240.19647	242.7667	2423.88796	2107.10373	2110.63572
comm1	229.12357	202.00012	201.57495	3270.82317	2740.79704	2835.0818
comm2	319.32932	288.06129	287.83254	2499.96374	2289.21022	2289.96725
face	564.06197	543.05073	542.5537	1631.30905	1520.57116	1516.58166
ferret	409.71111	378.49501	378.70829	2404.39613	2196.60147	2209.11407
fluid	289.73389	265.6545	260.04417	2220.18759	1991.79149	1990.3223
freq	267.31074	235.279	235.55733	2367.19713	2073.25443	2085.67482
stream	275.11834	230.61289	247.46276	2381.17843	2119.08407	2124.34417
swapt	261.45022	230.61289	232.85375	2300.24206	2001.3246	2024.99204
MT0	288.33736	257.06585	257.63625	2624.7777	2376.89776	2378.10819

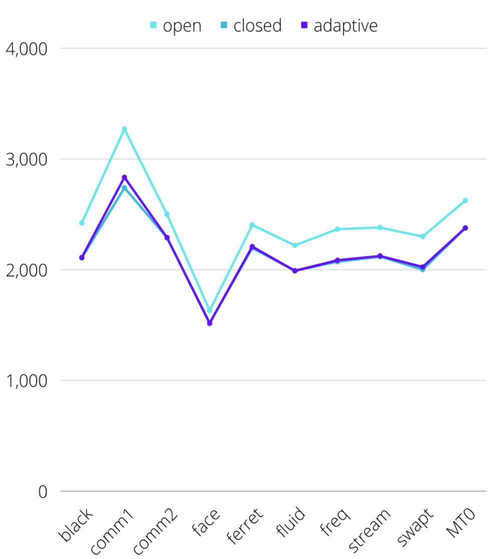
Plots of Latency in different paging modes:



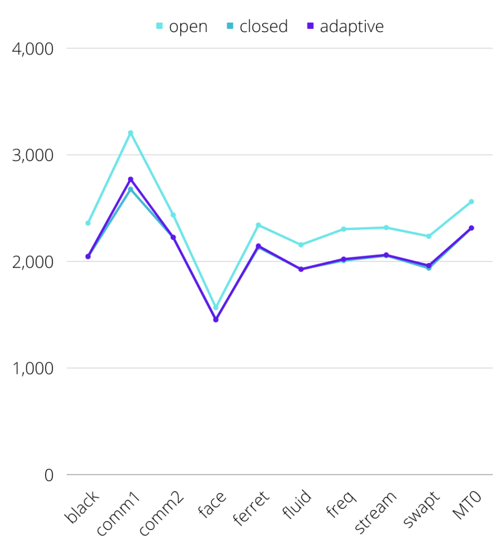
Read Latency (# cycles)



Read-queue Latency (# cycles)



Write Latency (# cycles)



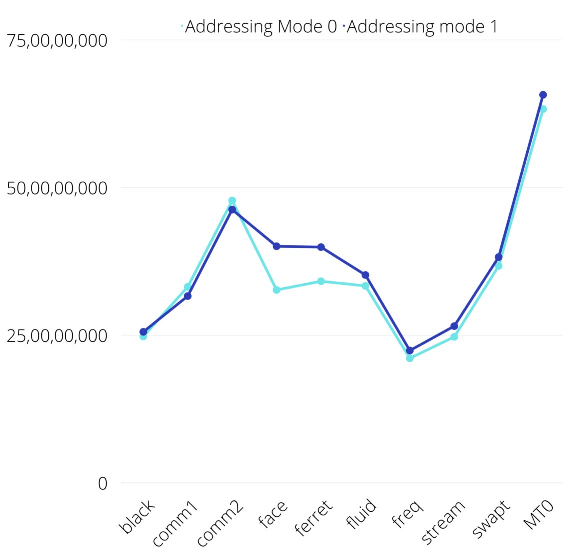
Write-queue Latency (# cycles)

Observation:

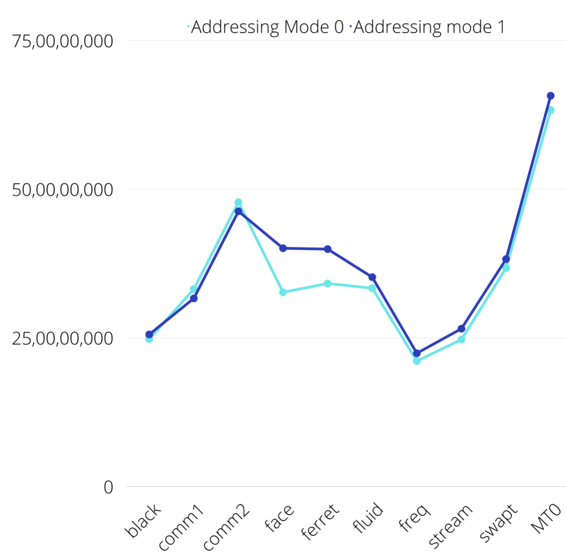
- The closed page policy and adaptive paging policy are almost equally optimal, with the closed page policy very marginally beating the adaptive policy in many cases.
- Hence, for the next part, both adaptive & closed paging policies have been used for address mapping comparison.
- The open page policy takes more cycles, more read and write latency in all traces when compared to the adaptive page policy.

Comparing DRAM performances of the two addressing modes (in millions):

	Adaptive		Closed	
Trace Name	**Addressing Mode 0**	**Addressing Mode 1**	**Addressing Mode 0**	**Addressing Mode 1**
black	247.95	255.70	247.40	252.97
comm1	331.88	316.27	331.65	314.15
comm2	478.01	462.71	477.86	462.19
face	326.63	400.60	326.45	399.70
ferret	341.33	399.19	341.02	392.91
fluid	333.52	352.00	335.59	348.77
freq	210.98	224.06	210.92	222.94
stream	247.24	265.40	246.61	265.01
swapt	367.35	382.32	364.16	380.63
MT0	632.94	657.06	632.51	654.85



Adaptive Paging Policy



Closed Paging Policy

Observation:

- Address mapping scheme 0 is better for most cases, taking around 7% lesser cycles than address mapping scheme 1.
- Address mapping scheme1 takes lesser cycles for comm1 & comm2 files; for the other traces, address mapping scheme 0 works better.

Results:

- The closed & adaptive paging policy perform with very close efficiency, while the open paging policy takes more cycles and has higher read & write latency.
- In both adaptive & closed paging policies, address mapping scheme 0 performs better than address mapping scheme 1 for most of the traces.