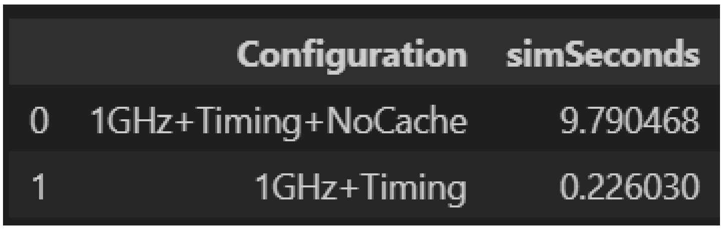
Lab Report-1 Part 2

Anannya Mathur

$$\text{CPU Execution Time} = \frac{\text{No of instructions} \times cpi}{\text{clock frequency}}$$

Adding L1 Cache:

The introduction of a cache makes the simulation faster.



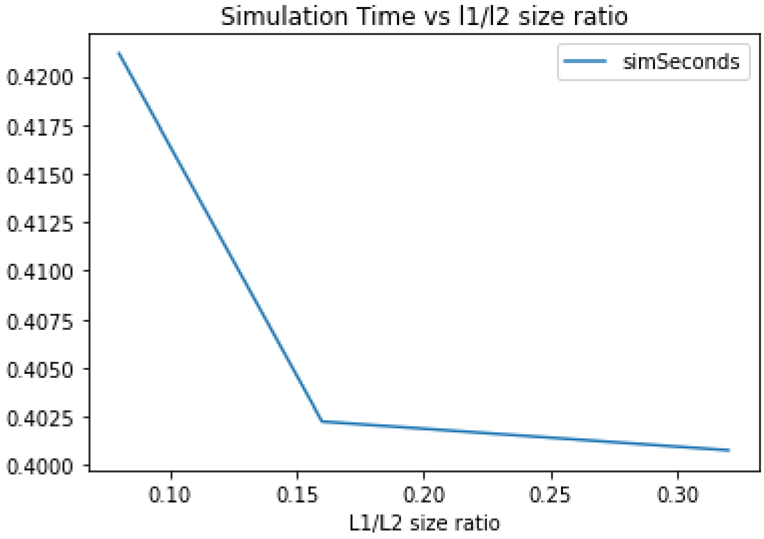
	Configuration	simSeconds
0	1GHz+Timing+NoCache	9.790468
1	1GHz+Timing	0.226030

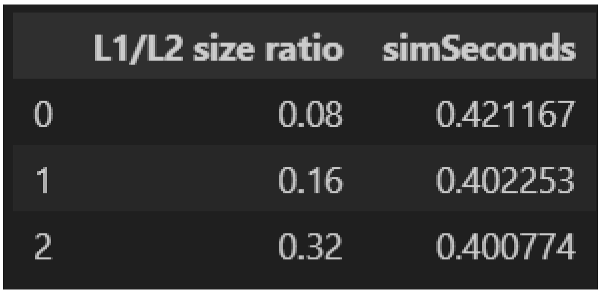
It can be observed that cache inclusion brings down the simulation time from 9.79s to 0.226s.

Changing Cache Parameters:

Varying L1/L2 size ratio:

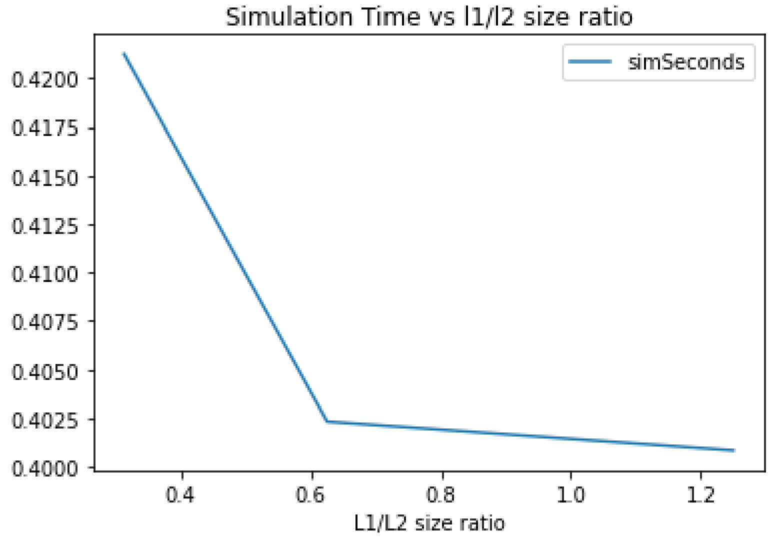
L2 size = 1MB→

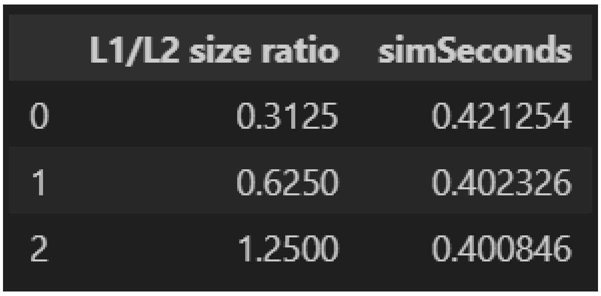




	L1/L2 size ratio	simSeconds
0	0.08	0.421167
1	0.16	0.402253
2	0.32	0.400774

L2 size = 256kB→

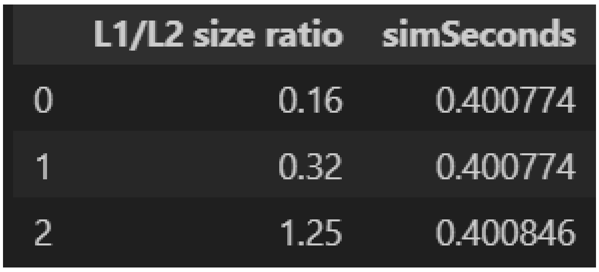




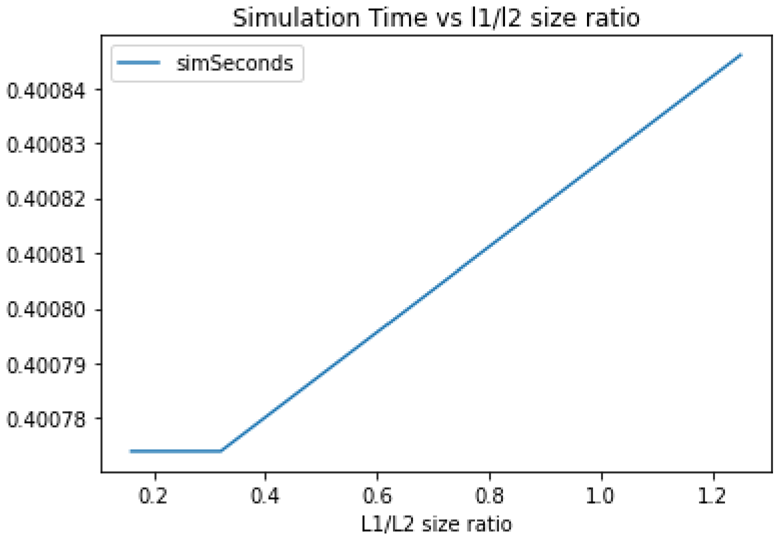
	L1/L2 size ratio	simSeconds
0	0.3125	0.421254
1	0.6250	0.402326
2	1.2500	0.400846

It can be observed that increasing the size of L1 while keeping L2 size constant brings down the time taken by the simulation to complete.

L1i size=256kB, L1d size=64kB→

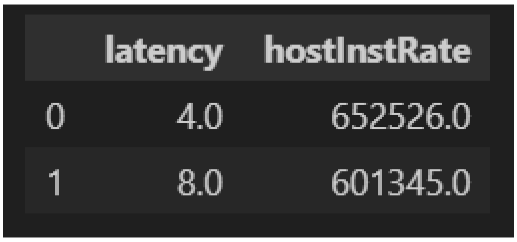


	L1/L2 size ratio	simSeconds
0	0.16	0.400774
1	0.32	0.400774
2	1.25	0.400846



It can be observed that decreasing the size of L2 while keeping L1 size constant slows down the simulation.

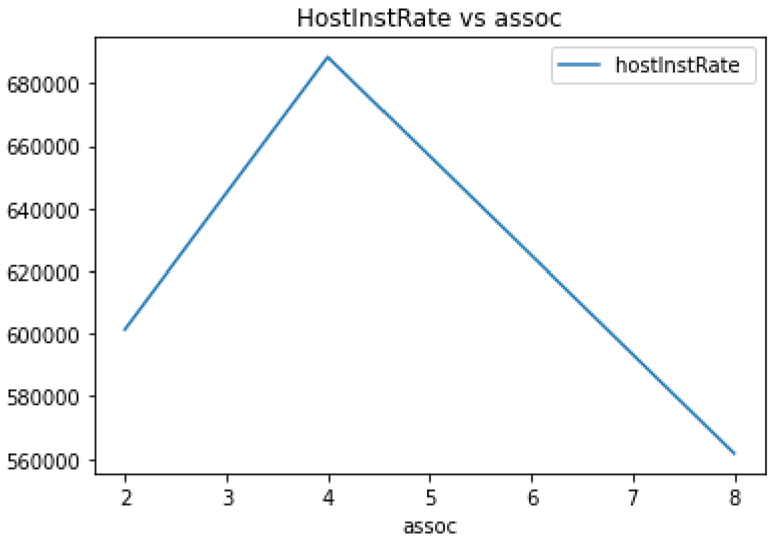
Varying data latency of L1 cache:



	latency	hostInstRate
0	4.0	652526.0
1	8.0	601345.0

“hostInstRate” denotes Simulator Instruction Rate and, therefore, gives an estimate of the simulator performance. Increasing data latency from 4 to 8 reduces the “hostInstRate”. Data latency of L2 cache was fixed at 20.

Varying associativity of L1 cache:

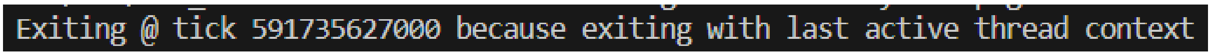


Associativity often allows better utilisation of cache resources, but increasing it beyond a point may not always lead to better performance, especially if the cache size is small.

Unified vs Split L1 Cache:

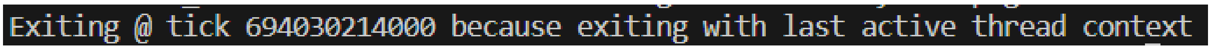
I tried to play around with config files to convert split L1 cache to unified. These were the results I observed:

Split L1 Cache:



```
Exiting @ tick 591735627000 because exiting with last active thread context
```

Unified L1 Cache:



```
Exiting @ tick 694030214000 because exiting with last active thread context
```

It can be observed that the simulation took longer to complete for the latter, thus, showing that split L1 cache works better.