

Lab4 Report_2019-10737_CHOI

전기정보공학부
2019-10737
최동호

🔗 Computer Architecture

Lab 4: Memory System Simulation

Part1

16384, 2, 64

```
-----  
L1 Hit Rate: 96.4078 %  
-----  
number of accesses: 1000000  
number of hits: 964078  
number of misses: 35922  
number of writes: 80669  
number of writebacks: 4377
```

16384, 4, 64

```
-----  
L1 Hit Rate: 96.8525 %  
-----  
number of accesses: 1000000  
number of hits: 968525  
number of misses: 31475  
number of writes: 80669  
number of writebacks: 2950
```

16384, 8 64

```
-----  
L1 Hit Rate: 97.1274 %  
-----  
number of accesses: 1000000  
number of hits: 971274  
number of misses: 28726  
number of writes: 80669  
number of writebacks: 2783
```

Part1 에서 기본적인 access 함수를 구현하였고, 결과는 각각 위와 같다.
1-B 에서는 LRU 스택을 std::list를 써서 구현하였다.

Part2

16384 4 64

```

-----
Performance Stats
-----
CPI: 10.7347
number of cycles: 8367886
number of insts: 779515
number of memory insts: 220485
-----
L1D Hit Rate: 96.8525 %
-----
number of accesses: 1000000
number of hits: 968525
number of misses: 31475
number of writes: 80669
number of writebacks: 2950
number of back invalidations: 0
number of writebacks due to back invalidations: 0

```

Part2 에서는 L1 - Memory의 hierarchy를 구현하였다.

이때 주의했던 점은

1. cache_base::access() 함수의 접근하는 것을 중복으로 세지 않도록 하는 것이었다.
2. dram의 done_func로 l1의 fill()을 호출하였다.

Part3-A

```

-----
Performance Stats
-----
CPI: 13.0714
number of cycles: 10189390
number of insts: 779515
number of memory insts: 220485
-----
L1D Hit Rate: 89.8372 %
-----
number of accesses: 1000000
number of hits: 898372
number of misses: 101628
number of writes: 80669
number of writebacks: 18412
number of back invalidations: 0
number of writebacks due to back invalidations: 0
-----
L2 Hit Rate: 68.9711 %
-----
number of accesses: 101628
number of hits: 70094
number of misses: 31534
number of writes: 0
number of writebacks: 0
number of back invalidations: 0
number of writebacks due to back invalidations: 0

```

Part3-A 에서는 L1U - L1D - Memory의 hierarchy를 구현하였다.

이를 위해

1. Write Miss 인 경우, 하위 레벨에는 Read로 접근 및 레벨에 따라 Fill을 달라해주었다.
2. L2 miss인 경우 DRAM에서 L1, L2 동시에 fill을 해주었다.
 1. dram의 done_func로 l2의 fill()을 호출하였다.
3. L2에서 eviction 시에 inclusiveness policy를 지키기 위해 back invalidation을 구현하였다.

Part3-B & Part4

구현하지 못하였다.