Computer Architecture Homework Assignment #3

Grading Policy

TA in charge: Myeongjae Jang
E-mail: myeongjae0409@kaist.ac.kr
Office: N1 - #922

I. Solution and Grading Policy

Q1.

A. Average memory access time
= (Hit rate) * (Hit access time) + (Miss rate) * (Miss access time)
= (1 - 0.05) * 1 + 0.05 * 105
= 0.95 + 5.25
= 6.20 cycles.

Scores are given as follows. **(10 points total)**
For Hit, (1 - 0.05) * 1 = 0.95 cycles. **(4 points)**
For Miss, 0.05 * 105 = 5.25 cycles. **(4 points)**
Average memory access time is 6.20 cycles. **(2 points)**

B. Because accesses are generated by the uniform random number generator, we can consider cache hit rate with size of cache and main memory.

Cache hit rate
= (Cache size) / (Main memory size)
= 64 KB / 256 MB
= 65,536 / 268,435,456
= 1 / 4,096

(For ease of calculation, 1 / 4,000 = 0.00025 also can be an answer.)
Now, we can compute average memory access time like (A).

Average memory access time
= 1 / 4,096 * 1 + (1 - 1 / 4,096) * 105
≅ 0.00024 + 104.97437
≅ 104.97461 cycles.

(If you compute cache hit rate as 0.00025, average memory access time will be 104.974 cycles.)
Scores are given as follows. **(10 points total)**
Cache hit rate is 1 / 4,096 or 1 / 4,000. **(5 points)**
Average memory access time is about 104.974 cycles. **(5 points)**

C. By the question, main memory access with cache disabled takes 100 cycles. It is lower than the result of (B). It means that the cache can affect worse for performance or can be useless if there are not any appropriate locality on process or data access pattern.

(To induce the answer above, the result of (B) should be higher than 100 cycles. It is also considered as grading policy.)
Scores are given as follows. **(13 points total)**

The result of (B) is higher than main memory access with cache disabled, 100 cycles. **(5 points)**

If there are not any appropriate locality on process or data access pattern. **(5 points)**

Cache can affect worse for performance or can be useless. **(3 points)**

D. We can measure usefulness of cache with G and L. When we use a cache, we can gain (Hit rate) * G but also lose (Miss rate) * L. As the result, we only get (Hit rate) * G - (Miss rate) * L by using cache. If it is lower than 0, it means that there are not any performance gain, and the cache use would be disadvantages.

If the cache use would be disadvantages,
(Hit rate) * G - (Miss rate) * L $\leq$ 0
(1 - Miss rate) * G - (Miss rate) * L $\leq$ 0
G - (Miss rate) * (G + L) $\leq$ 0
- (Miss rate) * (G + L) $\leq$ -G
(Miss rate) $\geq$ G / (G + L).

Therefore, if Miss rate is equal or higher than G / (G + L), the cache use would be disadvantages. Based on observation in the question, when G = 99 cycles and L = 5 cycles, Miss rate will be 99 / (99 + 5) = 99 / 104.
(Since the question asks Miss rate using G and L, you do not need to calculate 99 / 104 explicitly. However, if you calculate it without G / (G + L), it also can be an answer.)
Scores are given as follows. **(13 points total)**

The total gain by using cache is (Hit rate) * G - (Miss rate) * L or (Hit rate) * 99 - (Miss rate) * 5. **(5 points)**

The limitation of Miss rate is G / (G + L) or 99 / 104. **(8 points)**

Q2.

A. The fully associative 128-byte instruction cache can have whole 64-byte loop. It means that there are not misses, and miss rate will be 0.

Scores are given as follows. **(10 points total)**

128-byte cache can have whole 64-byte loop. **(5 points)**

There are not any misses, so miss rate is 0. **(5 points)**

B. For 192-byte and 320-byte loops, 128-byte instruction cache is too small. With LRU replacement policy, the oldest instruction will be evicted and the newest instruction will fill the cache. However, since they are loops, the evicted instruction will be accessed again. It will causes miss again and again. With a large number of iterations, it continuously causes cache miss. It means that miss rate will be 1 for both 192-byte and 320-byte loops.

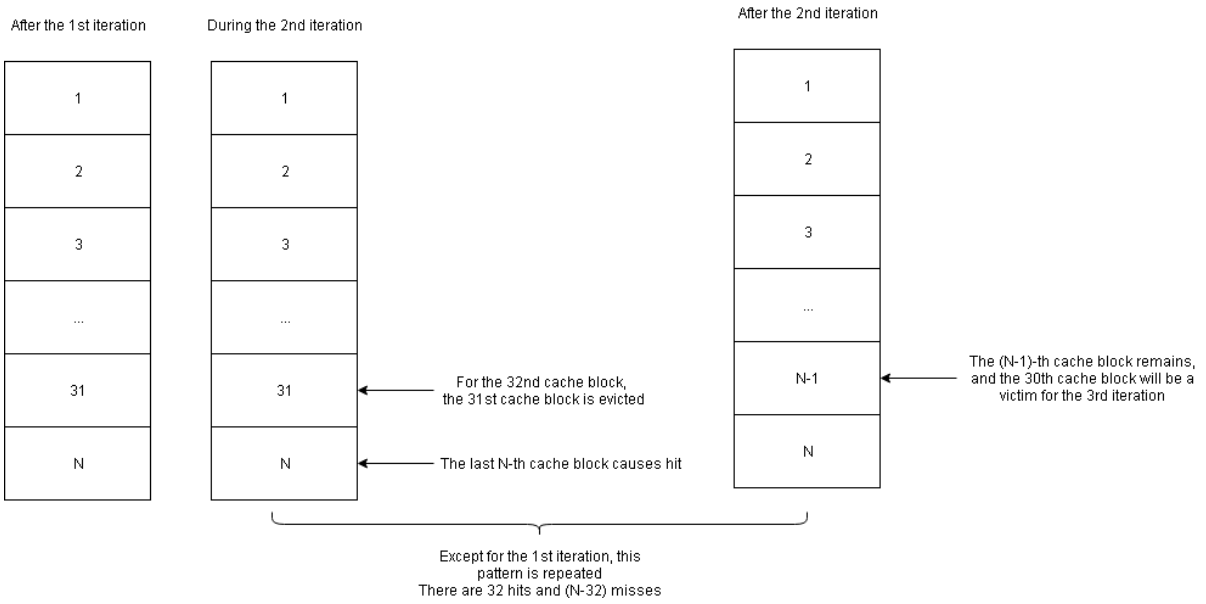
Scores are given as follows. **(10 points total)**

Both 192-byte and 320-byte loops are larger than the 128-byte instruction cache. **(3 points)**

Based on LRU replacement policy, larger loop than the cache size causes continuous miss. **(5 points)**

With continuous misses, miss rate will be 1. **(2 points)**

C. For the 64-byte loop, replacement policy cannot affect to performance since whole 64-byte loop are stored in the 128-byte cache. Its miss rate is still 0.
For the 192-byte and 320-byte loops, the newest 4-byte cache block will continuously evicted and replaced as the next cache block. The 128-byte cache has 32 cache blocks. We can draw the status of the cache as follows. The figure assumes the loop which has N blocks, (4 * N)-byte loop.



As the result, we can compute miss rate for the 192-byte and 320-byte loops.

Miss rate for the 192-byte loop
= (192 / 4 - 32) / 48
= (48 - 32) / 48
= 16 / 48
= 0.33

Miss rate for the 320-byte loop
= (320 / 4 - 32) / 80
= (80 - 32) / 80
= 48 / 80
= 0.6

For the 192-byte and 320-byte loops, miss rate is lower than before. It means that MRU replacement policy can be better than LRU replacement policy for the 192-byte and 320-byte loops.
In the question, it does not ask detailed miss rate for each loops. Therefore, above figure and calculation are additional explanation. You will get a point even though there are not any detailed computation. However, there should be explanation how MRU replacement policy affects to the pattern of eviction, and miss rate can be lower than 1.

Scores are given as follows. **(14 points total)**

For the 64-byte loop, there are not any changes and miss rate is still 0. **(4 points)**

For the 192-byte and 320-byte loops, since the oldest cache blocks are not evicted, there can be hits. **(5 points)**

For the 192-byte and 320-byte loops, their miss rate will be lower than 1. It means that MRU replacement policy can be better than LRU replacement policy. **(5 points)**

Q3. You can fill the table as follows. Since the question does not mention about replacement in TLB, you can assume that there are not any replacement. If you assume replacement and show appropriate replacement based on the replacement policy that you choose, it can be an answer. If there are not any mentions about replacement policy, it assumes that you did not consider replacement.
If the valid bit in TLB is 0, it should be dealt as like TLB miss. However, even though you consider it as TLB hit, it can be an answer if you explain the valid bit and it should be access the page table.

Virtual page accessed	TLB (hit or miss)	Page table (hit or fault)
1	Miss	Fault
5	Hit	X
9	Miss	Fault
14	Miss	Fault
10	Hit	X
6	Miss	Hit
15	Hit	X
12	Miss	Hit
7	Miss	Hit
2	Miss	Fault

Scores are given as follows. **(20 points total)**
For each blanks, **(1 points)**