HW7.3. Cache Properties

How do the following caches compare in hit time and hit rates to a 64 KiB, 8-way set associative cache with 1 KiB blocks?

Q1: A 64 KiB, direct-mapped cache with 1 KiB blocks
How does cache associativity affect the hit time/hit rate?

Q1.1: Hit Time:

○ (a) Increased Hit Time

○ (b) Minimal effect/Could increase or decrease the Hit Time

○ (c) Decreased Hit Time

Q1.2: Hit Rate:

○ (a) Increased Hit Rate

○ (b) Minimal effect/Could increase or decrease the Hit Rate

○ (c) Decreased Hit Rate

Q2: A 128 KiB, 16-way set associative cache with 1 KiB blocks
How does cache capacity affect the hit time/hit rate?

Q2.1: Hit Time:

○ (a) Slightly Increased Hit Time

○ (b) Minimal effect/Could increase or decrease the Hit Time

○ (c) Slightly Decreased Hit Time

Q2.2: Hit Rate:

○ (a) Increased Hit Rate

○ (b) Minimal effect/Could increase or decrease the Hit Rate

○ (c) Decreased Hit Rate

Q3: A 64 KiB, 8192-way set associative cache with 1 B blocks
What happens if we have a very large cache associativity but very small block size?

Q3.1: Hit Time:

○ (a) Increased Hit Time

○ (b) Minimal effect/Could increase or decrease the Hit Time

○ (c) Decreased Hit Time

Q3.2: Hit Rate:

○ (a) Most likely increased Hit Rate

○ (b) Minimal effect/Could increase or decrease the Hit Rate

○ (c) Most likely decreased Hit Rate

Recall the formula for the average memory access time:

$$AMAT = HitTime + MissRate * MissPenalty$$

The processor always checks the cache first. If there is a hit, it will only need to access the cache (thus the inclusion of hit time in the formula). If there is a miss, then the cache has to access the next level of memory first (the main memory if there's only one cache, or the next level of cache L2, L3, and so on) to fetch the block (the miss penalty is the penalty in accessing the next level of memory during misses).

Q4: For this question, assume that we have 1 MiB of main memory, and that accessing main memory takes 100 clock cycles.

Q4.1: We add a 1 KiB cache which has a hit time of 5 clock cycles; our miss penalty is still the 100 clock cycles needed to access main memory. To test this cache, we then run a program that accesses **random memory addresses**. To the nearest clock cycle, what does the AMAT converge to as the program runs indefinitely?

Homework 7

Assessment overview

Total points: 0/100

Score: 0%

Question

Value: 20

History:

Awarded points: 0/20

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

Hint: If the memory accesses are purely random, are you exploiting the spatial/temporal locality offered by caches? What happens to your miss rate?

integer ?

Q4.2: We test the above cache on another program, which exhibits an 80% hit rate. What is our AMAT when running this program on this cache?
Hint: How does hit rate relate to miss rate which is part of the AMAT calculation?

integer ?

Q4.3: In order to "break even" (having a cache is more efficient than the naive solution of directly accessing the main memory), what would you need your hit rate to be?
Hint: This is the point where AMAT = main memory access time (miss penalty).

number (rtol=0.01, atol=1e-08) ?

Q4.4: We add an L2 cache (the earlier cache is the L1 cache), which has a hit time of 20 cycles. We run a program that exhibits an 80% L1 hit rate, and a 95% L2 hit rate (note that the L2 hit rate only counts accesses that miss on the L1 cache). Our miss penalty is still the 100 clock cycles needed to access main memory. What is the AMAT for this cache setup on this program?
Hint: L1 cache is closer to the processor. If it misses, it accesses the L2 cache. If L2 cache misses, it accesses the main memory. How would AMAT calculation change in this configuration?

integer ?

Save & Grade *20 attempts left* Save only *Additional attempts available with new variants* ?