CS 2410

Computer Architecture

Spring 2022

Distributed: Mar 14th, 2022

Due: 11:59pm Mar 28th, 2022

Points: 100

1. (20 pts) Fill the table below

TLB	Page Table	Cache	Possible? (yes or no)	Under what circumstances?
Hit	Hit	Hit	Yes	This is what we want!
Hit	Hit	Miss	Yes	Although the page table is not checked after the TLB hits
Miss	Hit	Hit	Yes	TLB missed, but PA is in page table and data is in cache
Miss	Hit	Miss	Yes	TLB missed, but PA is in page table, data not in cache
Miss	Miss	Miss	Yes	Page fault
Hit	Miss	Hit/ Miss	No	TLB translation is not possible if the page is not present in main memory
Miss	Miss	Hit	No	Data is not allowed in the cache if the page is not in memory

2. (20 pts) Consider the following code for a nested loop and assume that the inner loop executes 50 iterations and the outer loop executes 10 iterations.

```
L:  lw $1, 100($5)
    lw $2, 800($6)
    addi $5, $5, -4
    add $3, $1, $2
    sw $3, 804($5)
B1: bneq $5, $10, L
    addi $6, $6, -4
B2: bneq $6, $10, L
```

Complete the following sentences assuming that i) B1 and B2 do not collide in the Branch Target Buffer, ii) B1 and B2 have separate branch predictors, and iii) All the 1 bit predictors start with the initial state as predict-taken and all the 2-bits predictors below start with the initial state as predict-strongly-not taken.

In the case you think 50 times of inner loop for the first outer loop iteration
- A 1-bit branch predictor will correctly predict the branch at B1 58 times out of 59 times
- A 1-bit branch predictor will correctly predict the branch at B2 9 times out of 10 times
- A 2-bit branch predictor will correctly predict the branch at B1 55 times out of 59 times
- A 2-bit branch predictor will correctly predict the branch at B2 7 times out of 10 times

In the case you think 50 times of inner loop for each outer loop
- A 1-bit branch predictor will correctly predict the branch at B1 481 times out of 500 times
- A 1-bit branch predictor will correctly predict the branch at B2 9 times out of 10 times
- A 2-bit branch predictor will correctly predict the branch at B1 488 times out of 500 times
- A 2-bit branch predictor will correctly predict the branch at B2 7 times out of 10 times

3. (20 pts) Assume a system with a 4-way set associative L1 cache and a 8-way set associative L2 cache, both with block size of 8 words. The hit times for the L1 cache and the L2 cache are 1 and 5 cycles, respectively. Assume also that the L1 miss rate is 10% and that 80% of the total misses in L1 are found in L2. On a L2 miss, a block is fetched from memory and it takes 72 cycles for the first word of the block to reach the L2 and additional 4 cycles for each of the 7 following words to reach the cache (i.e., each word except the first one takes additional 4 cycles). That is, the L2 miss penalty is 100 cycles. Ignore any time needed to transfer blocks (or words) from L2 to L1 and to the CPU.

a) Compute the average memory access time (in cycles)
AMATa = 1 + 0.1 * (5 + 0.2 * 100) = 1 + 0.1 * (5 + 20) = 3.5

b) What would be the L2 miss penalty and the average memory access time if early restart and critical word first are used when a miss occurs in L2 and a block is fetched from memory.
Miss penalty = 72
AMATb = 1 + 0.1 * (5 + 0.2 * 72) = 2.94

c) Assume that early restart and critical word first are not used, but that way prediction is used for L1. With correct prediction, the L1 hit time is still 1 cycle, but a pseudo hit (way misprediction) takes 2 cycles (one cycle after the discovery that the predicted way resulted in a miss). What is the AMAT if the way prediction is correct 60% of the time?
AMATc = (0.6 * 1 + 0.4 * 2) + 0.1 * (5 + 0.2 * 100) = 3.9

d) In what circumstances a way prediction would be beneficial? And why?
Open question and you should discuss about the variance in accuracy and the hit time

4. (20 pts) The table below shows two cores and their read/write operations on two different words of a two-word cache block X (initially X[0] = X[1] = 0). Assume each core has private L1 caches (write-through for the L1D$) and that they share a single L2 cache.

Core 0	Core 1
X[0] = 10; X[1] = 3	X[0] = 5; X[1] += 2

List the possible values of cache block X in the L1D$ for a correct cache coherence protocol implementation. List at least one more possible value of cache block X if the protocol doesn't ensure cache coherency. (note that the answer in this question is regardless of what coherence protocol being used).

X[0] and X[1]: 5-5, 10-5, 5-3, 10-3

If non-coherent, a combination apart from prior 4 with explanation

5. (20 pts) Recall the cache optimizations and TLB optimizations we have discussed in class. Assuming that you are designing the memory system (including TLB and cache) for the following two types of application domains, please choose the optimization strategies for TLB and cache and justify why you choose them. Note that you may choose multiple cache optimizations and multiple TLB optimizations as long as you provide sound reasons.
   a) Graph applications dealing with irregular graphs (e.g., twitter user connection diagrams). **Hint:** random memory access pattern
   b) Streaming application dealing with regular images and videos. **Hint:** regular memory access pattern.

Open question, as long as you can justify your answer with reasons.