Computer Architecture

Final Exam

Spring 2012

2012/06/13
09:30 A.M. – 10:45 A.M.

Total: 130 points

Read Me First

0. 답은 한글로 적어도 좋습니다**.**

1. If you cheat on exam, you will get an "F" for the course.

2. Some questions have several subproblems. Make sure that you put all the answers for the subproblems in one place on your answer sheet.

Visit the course homepage for the final exam scores and grades.
The exact schedule will be announced on the homepage.

Problem 1. [30]

(a) [5] When a processor is based on 20-bit addresses, if you are designing a 64-KB 4-way set-associative cache, how many bits out of the 20-bit address are allocated to the cache tag field?

(b) [10] Suppose a computer's address size is *k* bits (using byte addressing), the cache size is *S* bytes, the block size is *B* bytes, and the cache is *A*-way set associative. Assume that *B* is a power of 2, so $B = 2^b$. Express the number of bits needed to implement the cache in terms of *S*, *B*, *A*, *b* and *k*.

(c) [10] We are going to run the following C program fragment using a data cache U of a unknown configuration with 32-byte cache blocks:

```
#define SIZE (32 * 1024)
int A[SIZE], B[SIZE], C[SIZE];
for (i = 0; i <SIZE; i++)
        A[i] = B[i] + C[i];
```

Assume that `int` maps to a 4-byte data quantity, and the arrays `A`, `B`, and `C` are located at the address `0x66340000`, `0x66440000`, and `0x66540000`, respectively. If you would like to maximize the number of data cache misses for this program, what should be the configuration of the cache U? For the cache U, how many cache misses occur while executing this program fragment? Assume that we are using 32-bit addresses.

(d) [5] What is the main difference between a blocking cache and a non-blocking cache?

Problem 2. [10]

We have a program segment consisting of two conditional branches, B1 and B2, which are executed multiple times during a single execution of the program segment. Below are the outcomes of each branch when the program segment was executed (T for taken and N for not taken):

B1: T-T-T-N-T

B2: T-T-N-T-T-N-T

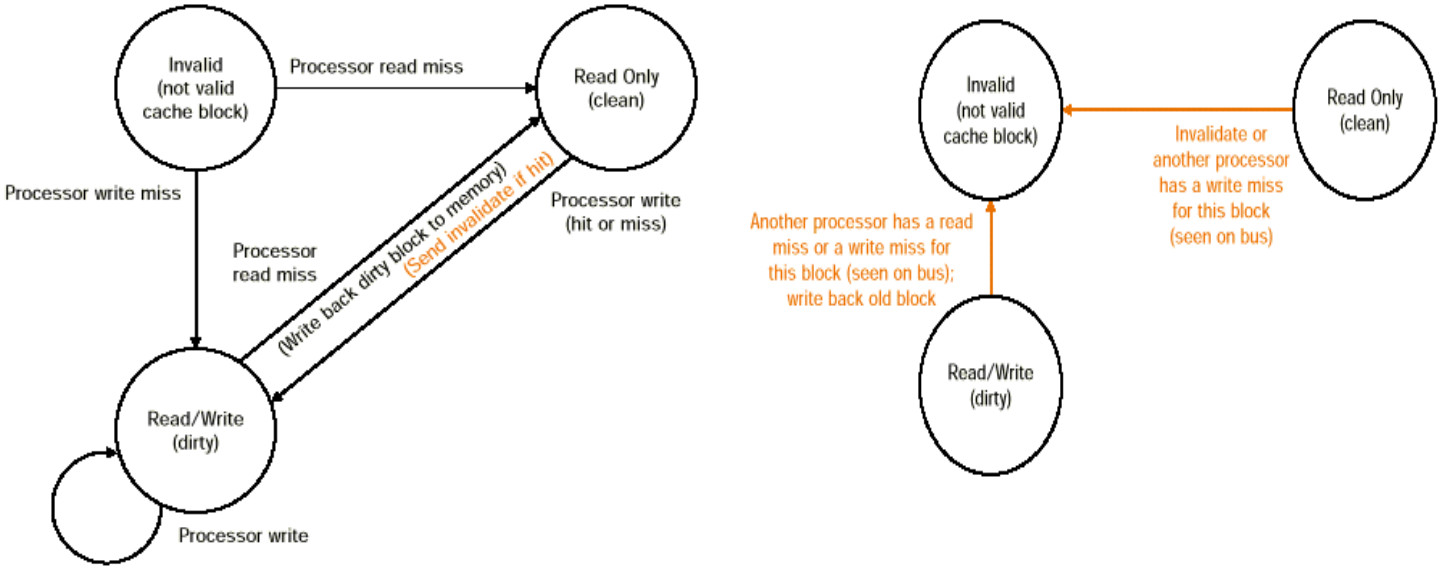
Assume that each branch has its own prediction buffer. Compare the prediction results of two branch prediction schemes: (1) 1-bit predictor, initialized to predict taken and (2) 2-bit predictor, initialized to weakly predict taken. For each scheme, count the number of correct predictions and the number of incorrect predictions.

Problem 3. [10]

(a) [5] Explain how ephemeral history registers can be used in implementing a bypass register file.

(b) [5] What is a scoreboard? Why is it useful?

Problem 4. [10]



The above diagrams show state transitions for a write-invalidate cache coherence protocol based on a write-back policy. Assuming that you are using the above protocol with three processors, P0, P1, and P2, fill in the state for a cache block X in each of processors for the memory operations listed below. Use the same table for your answer.

Operations	P0	P1	P2
Initially	Invalid	Invalid	Invalid
(1) P2: loads X			
(2) P0: loads X			
(3) P1: stores X			
(4) P2: stores X			
(5) P1: loads X			

Problem 5. [10]

Suppose that you are about to execute 'load word' instruction on a computer with a translation-lookaside buffer, an L1 data cache, an L1 instruction cache, and an L2 unified cache. Explain in detail when the execution of 'load word' becomes the slowest.

Problem 6. [10]

(a) [5] What is the main disadvantage of virtually addressed caches?

(b) [5] What is weak scaling?

Problem 7. [10]

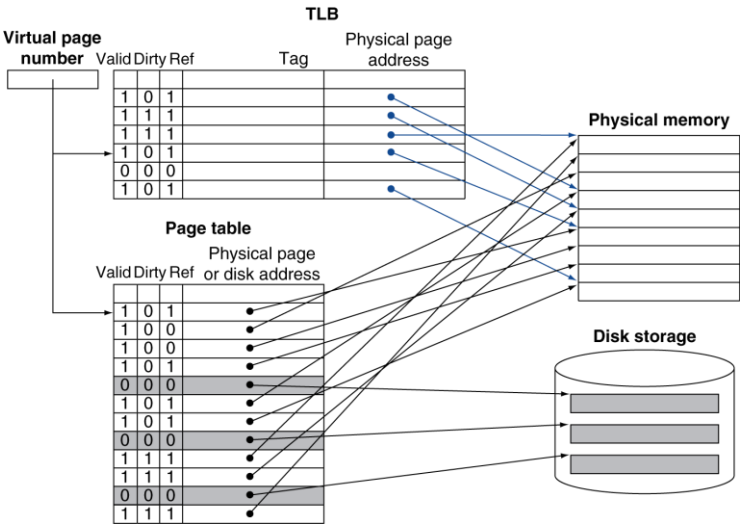
One of the key design principles of designing high-performance systems is to make the common case fast. Give one case where this principle was successfully applied. Give your explanation based on the following framework:

1. The definition of the common case
2. How to improve the performance for the common case
3. Impact of the proposed improvement on non-common cases

Problem 8. [10]

(a) [5] What is the main role of the Dirty bit below?

(b) [5] What is the main role of the Ref bit below?



Problem 9. [30]

Decide if the following statement is *true* or *false*. If your answer is false, *argue* why the statement is not true. (If the answer is false but you didn't provide enough argument to justify your answer, you will get no point.) If the statement is neither *true* nor *false*, explain why it is *undecidable*.

(a) [5] Increasing the depth of pipelining always increases performance.

(b) [5] For data-intensive applications with a large number of loads and stores, the write-through cache write policy with a write buffer is more useful than the write-back policy.

(c) [5] Every bus arbitration scheme does not allow any kind of bus collisions.

(d) [5] DMA delegates more I/O responsibility from the CPU than IOP does.

(e) [5] Forwarding can completely resolve the data hazard problem.

(f) [5] All the statements of Problem 9 are false.