CSE 3038 – Computer Organization - QUIZ #4.

Problem 1. (48 PTS)

Assume that cache size is 2048KB and the size of the physical memory is 32 GB. Assume that the block size of 32 words, a 2-way set associative cache, and a 32-bit address. (Assume that we consider byte addressing)

a) (21) How many bits are needed for the following? **Show all your work.**

Tag	Index	Offset (total)
12	13	7

32 words x 4 = 128 bytes = 2^7 bytes
(7-bit offset)

2048 KB = $2^1 \times 2^{20} = 2^{21}$

$2^{21} / 128 = 2^{14}$ $\quad 2^{14} / 2\text{-wy} = 2^{13}$ $\quad$ index = 13 bits

Tag ⇒ 32 − (13 + 7) = 12

b) (27) Rewrite the above fields for each of the following changes. (You should consider each case separately). **Show all your work.**

Change	Tag	Index	Offset (total)
Increase the cache size from 2048 KB to 8192 KB	10	15	7
Increase the block size from 32 words to 128 words	12	11	9
Change the 2-way set associativity to 8-way set associativity	14	11	7

* incr. cache size → No chge in offset (index ↑ 2 bits)
* incr. block size 4 times ⇒ offset ↑ 2 bits, index ↓ 2 bits)
* 2 wy → 8 wy ⇒ No chge in offset. index ↓ 2 bits

Problem 2 (28 PTS)

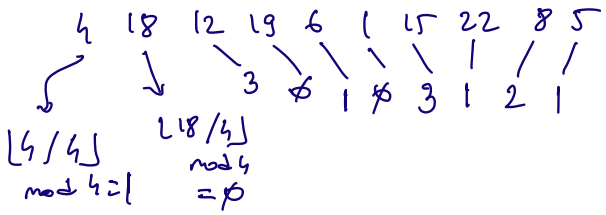
Suppose you have the following direct mapped cache system: cache size = 4 blocks, 1 block = 4words, cache hit time = 50ns, time to read a block from main memory to cache = 500ns. After running a certain program, the observed string of **word** addresses for instruction fetch is:

4, 18, 12, 19, 6, 1, 15, 22, 8, 5

M M M H H M H M M M

a. (16) Show the final content of the cache.

Set # = $\left\lfloor \frac{\text{word addr}}{\text{\# of words per block}} \right\rfloor \bmod N$ ← # of sets for set assoc.



~~16~~ ~~17~~ ~~18~~ ~~19~~ 0 1 2 3			
~~84~~ ~~85~~ ~~86~~ ~~87~~ ~~20~~ ~~21~~ ~~22~~ ~~23~~ 4 5 6 7			
8 9 10 11			
12 13 14 15			

0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15

b. (12) Compute the total access time for instruction fetches for the above program.

3 Hits & 7 Misses

Total Time = Time for accessing cache + Time for Accessing main memory (Miss penalty)
(Hit / Hits)

= 10 × 50 + 7 × 500

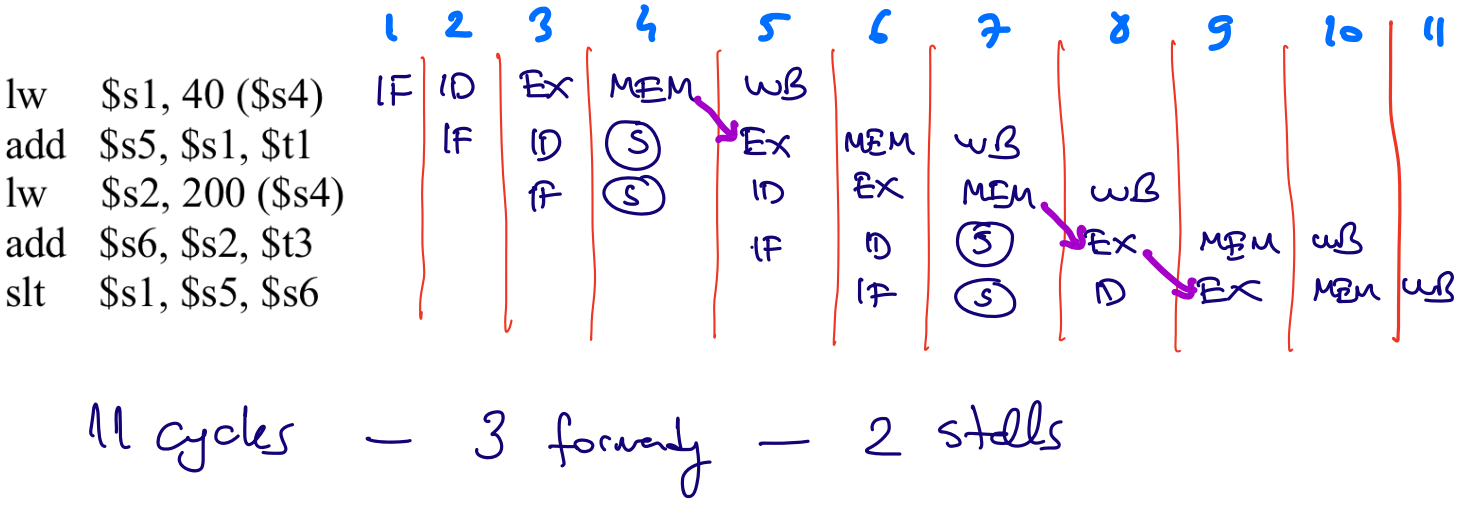
= 4000

Problem 3 (24 points)

Assume that our processor handles hazards by forwarding results whenever possible. If forwarding used, forward from the earliest stage at which the results is ready. It is allowed to write and read registers at the same cycle.

```
lw    $s1, 40 ($s4)
add   $s5, $s1, $t1
lw    $s2, 200 ($s4)
add   $s6, $s2, $t3
slt   $s1, $s5, $s6
```

a. (12) What is the total number of cycles required to execute the above code on a pipelined MIPS? Draw the pipeline chart and show all necessary stalls and/or forwarding.



11 cycles — 3 forwards — 2 stalls

b. (12) Assume that you can reorder the above code by preserving the program correctness. Write down the order of instructions *which minimizes total number of cycles required to execute the given set of instructions.*

Change "seond lw" instruction with "first add"

```
lw $s1, 40($s4)
lw $s2, 200($s4)
add $s5 $s1,$t1
add $s6, $s2, $t3
slt $s1, $s5, $s6
```

* There will be no stall cycles.

# of cycles = 9 cycles

1 forward (add $s6 → slt)