CSEE W3827

Fundamentals of Computer Systems

Homework Assignment 5

Profs. Stephen A. Edwards & Martha Kim

Columbia University

Due December 10th, 2012 at 5:00 PM
Turn in at CSB 469.

Write your name **and UNI** on your solutions

Show your work for each problem; we are more interested in how you get the answer than whether you get the right answer.

1. (25 points) Pipelined software execution.

   (a) Assuming a fully bypassed (i.e., both W-E and M-E operand forwarding), 5-stage MIPS pipeline with early branch resolution, indicate all of the stalls and operand forwarding that is used to execute the following ten instructions:

```
bnez $a0, tree_sum_recurse
addi $sp, $sp, -12
sw $ra, 0($sp)
sw $s0, 4($sp)
sw $s1, 8($sp)
move $s0, $a0
lw $s1, 0($s0)
lw $a0, 4($s0)
```

   (b) How many cycles will it take for this code to execute *completely* on the pipelined processor?

   (c) Will this code run faster on a 100MHz single cycle processor or a 400MHz pipelined processor?

2. (25 points) Assuming a pipeline with no bypassing, reorder the instructions such that they compute the same function, but require *no* pipeline stalls due to data hazards. Branch stalls are fine. You should reorder instructions as effectively as you can before inserting as few nops as possible. Of the nops you inserted, indicate which could be avoided with bypassing. FYI: This function takes a pointer to a string and a character, and counts how many times the character appears in the string.

```
count:
  addi  $sp, $sp, -4             # save registers
  sw    $ra, 0($sp)
  li    $v0, 0                   # $v0 stores count
count_top:
  lb    $t0, 0($a1)              # char B from from string into $t0
  beq   $t0, $zero, count_done   # if B == NULL, done
  bne   $t0, $a0, count_advance  # if B != X, move on
  addi  $v0, $v0, 1              # else, inc counter, then advance
count_advance:
  addi  $a1, $a1, 1              # advance pointer in string
  j     count_top
count_done:
  lw    $ra, 0($sp)              # restore registers & return
  addi  $sp, $sp, 4
  jr    $ra
```

3. (25 pts.) Suppose the MIPS processor were divided into 10 stages, s1 through s10. The key operations are distributed across the stages as follows:

s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
	branch resolution		register decode	ALU			data memory		register writeback

Hint: You can ignore conditional branches for this problem.

(a) What is the ideal CPI of this pipeline?

(b) Assuming no operand fordwarding, what is the CPI of producer and consumer instructions?

(c) Consider an instruction mix of 25% loads, 10% stores, 2% jumps, and 52% R-type instructions. Assuming that 50% of the loads and R-type instructions are followed immediately by their consumers, and that there is no data forwarding, what is the average CPI of this workload?

(d) What is the average CPI on the standard 5 stage pipeline, also without operand forwarding?

(e) How much faster does the 10-stage pipe's clock need to be than the 5-stage in order for the two processors to have the same performance on this workload?

4. (25 pts.) Consider the **direct mapped** cache that interprets an 8-bit address according to the `{tag:setIdx:byteOffset}` format specified.

   (a) Complete the table on the following page, indicating whether the each access in the reference stream will hit (H) or miss (M) in each cache.

   (b) If the L1 access time is 7ps, and memory has a 100ps access time, what is the expected access time for the references above on the two level hierarchy?

   (c) Will the addition of an L2 cache with a 50% miss rate and 20ps access time improve or degrade the hierarchy?

Address Format →	L1 {2:2:4}
Block Size (Bytes):	
Cache Size (Blocks):	

Address References (in binary)	
01000000	
00100000	
00010000	
00001000	
00000100	
00000010	