CDA 3101: Spring 2019
Homework 2

Total Points: 50
Due: Thursday 03/14/2018, in class (by 4:50 PM)

Submissions are due by the beginning of class on the specified due date. Handwritten or typed solutions are acceptable. If you do write your solutions by hand, please write clearly. If the TA's cannot read your answer, they cannot give you the points.

Late submissions will be accepted with a 10% penalty by 4:50 PM on Friday, 03/15/2019. Late submissions should be turned in in my mailbox or under my door.

You must show how you arrived at the answer and circle your final answer where applicable!

1 Problem 1 - 20 points

Squidward has decided to branch out into hardware design. Being new to the field, he's asking for your help in designing a pipeline of a new processor. Once the processor has been built, he's going to test it with a sample program that contains $2x10^9$ instructions.

(a). If the new processor were a non-pipelined, single cycle design and each instruction took 860 ps to finish, how long would it take to execute the sample program?

(b). If the new processor were a non-pipleined multi cycle design with the longest stage taking 140 ps to finish, how long would it take to execute the sample program? (You may assume instructions average out at 4 cycles each)

(c). Assume the current state-of-the-art pipeline has 15 stages. Assume also that the stages are perfectly balanced. How much speedup will it achieve compared to the non-pipelined single cycle processor? Be exact here!

(d). Realistically, we cannot achieve ideal speedup due to the overhead of implementing pipelining stages (e.g. imperfectly balanced stages, adding pipeline registers, etc). Does this overhead affect the instruction latency, instruction throughput, or both? For each metric affected, is the effect significant? Why or why not?

2 Problem 2 - 20 points

Consider the following instructions. Complete the pipeline diagram, indicating the cycle in which each instructions stages are executed. Assume that forwarding is used to avoid stalls, when possible. Draw lines between stages to indicate forwarded values. If necessary, stalls may be indicated with an S. You will certainly need more cycles than are included below. You only need to show one run through these lines of code. Assume the branch is NOT TAKEN at the beq. This is a loop, but you don't have to go on until the program halts.

Cycle	1	2	3	4	5	6	7	8	9	10
ori $s0,$0,24										
lw $t0, 8 ($s0)										
sll $s2, $t0, 3										
LOOP: lw $s1, 12($s2)										
addi $t0,$s1,1										
sub $t3, $s0, $t0										
beq $0,$t3,LOOP										
sw $s5, 16($s2)										

3 Problem 3 - 10 points

Suppose that a machine with a 5-stage pipeline uses branch prediction. 12% of the instructions for a given test program are branches, of which 84% are correctly predicted. The other 16% of the branches suffer a 4-cycle mis-prediction penalty. (In other words, when the branch predictor predicts incorrectly, there are four instructions in the pipeline that must be discarded.) Assuming there are no other stalls, develop a formula for the number of cycles it will take to complete n lines of this program