Homework 5 Due: Nov 25, 2025

Please write legibly or type. Show your work to the extent possible. **Question 1**(60 points): Given the following code:

```
Loop:
    lw    $t0, 0($s1)
    add   $t0, $t0, $s2
    sub   $t0, $t0, $s3      # extra dependency
    and   $t4, $t0, $s4      # added dependent instruction
    or    $t5, $t4, $s5      # added dependent instruction
    sw    $t5, 0($s1)
    addi  $s1, $s1, -4
    bne   $s1, $zero, Loop
```

(a) Arrange the loop in the two-issue Slot 1 / Slot 2 table format shown in slide 20, with respect to all hazards. Compute the IPC.

(b) Unroll the loop three times (as opposed to four in slide 23). Show the three copies.Rescheduled loop unrolled using the Slot 1 / Slot 2 table and compute IPC.

(c) Apply register renaming following the method shown in slide 26. Use distinct temporaries for each unrolled copy and highlight index changes. Reschedule and compute the IPC.

(d) Compute the speedup achieved from part (b) to part (c).

Question 2 (40 points):

Code Block for Question 2

```
1: addi $s1, $s0, 16
2: lw   $t0, 0($s1)
3: addi $s2, $s0, 200
4: lw   $t1, 0($s2)
5: mul  $t2, $t0, $t1     # hazard chain extension
6: add  $t3, $t2, $t4
7: and  $t5, $t3, $t6
8: or   $t7, $t1, $t5
```

(a) Identify the hazards and draw the dependency structure in the format used in slide 37.

(b) Assume both load instructions experience cache misses. Rewrite the code and annotate each line as in slide 38: "cache miss", "on hold", or "OK to execute."

(c) Before the misses resolve, show the out-of-order execution that can occur, and list the instructions in the reorder buffer with destination, ready/not-ready, and commit status.