COSC175 Day21: Multi-Cycle Processor Performance

Start by **assigning roles** in your group! If you have only 3 members, the Ambassador+Recorder roles are combineable, and for groups of 5, multiple Ambassadors or Sketchers will help!

- ***Ambassador*** should have the textbook, lecture materials, and other examples handy, and is the point-person for raising their hand to ask a question.
- ***Recorder*** should take detailed notes, either on the handout or in a digital format, and will submit notes on behalf of the group at the end of class.
- ***Sketcher*** should have a marker in hand, drawing at the board as your group brainstorms and works through the exercises.
- ***Executive*** should keep an eye out that all group members are staying in-the-loop, as well as an eye on the clock in case you are stuck on a point, would benefit from help, or would benefit from moving on to another question.

With your group **at the whiteboard**, work through the following exercises to brainstorm multi-cycle processor performance (adapted from DDCA Chapter 7 Exercises)!

Part 0: Multicycle Performance Estimation

For each of the following example programs, determine the performance by answering the following questions:

1. How many instructions will execute?
2. How many of each *type* of instruction will execute (e.g., branch vs. R-type vs. load word vs. store word)?
3. How many total cycles does this program take to execute?
4. What is the average CPI for this program?
5. For a cycle time of 375ps, how long does this program take to execute?

Example Program A

```
     addi a0, zero, 5       # result = 5
L1: bge zero, a0, done      # if result <= 0, exit
     addi a0, a0, -1            # result = result - 1
     jal zero, L1          # equivalent to `j L1`
done:
```

Example Program B

```
        addi s0, zero, 0    # i = 0
        addi s1, zero, 0    # sum = 0
        addi t3, zero, 10   # t3 = 10
loop:  beq s0, t3, L2       # if i == 10, jump to L2
        add s1, s1, s0      # sum = sum + i
        addi s0, s0, 1      # i = i + 1
        jal zero, loop      # equivalent to `j loop`
L2:
```

Part 1: Brainstorming About Multicycle Performance

1. In which cases (if any) will the multicycle processor provide better program performance than our single-cycle design? Explain your answer.
2. If you were provided an alternative register file design that uses 50% less power, but increases the register read time by 80ps, would you prefer to use this register file in your design? What would be the advantages and disadvantages with respect to program execution time, resources, power, etc.?
3. If you could reduce the delay of any one component in the datapath, which component would you choose and why?

At the end of class be sure your recorder hands in a copy of your notes with all group members listed! You can hand in notes on paper or via email with [COSC175] in the subject line. I use these notes to check your understanding and your engagement, not to grade for correctness.