# CS 382 Fall 2023 — Computer Architecture and Organization — Homework 3

**Requirement (READ FIRST!)**

You have to **type** the solutions. **Handwritten homework will not be graded and will receive zero credit.** You can annotate on this document directly (you might need to know how to insert a picture into a PDF file); or you can submit a separate PDF, but with solutions clearly marked with question numbers.

## 1 (15 points)

When silicon chips are fabricated, defects in materials (*e.g.,* silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get "broken" and always register a logical 0. This is often called a "stuck-at-0" fault. Answer the following questions based on Figure 3.27 in the textbook.

### 1.1 (5 points)

Which instructions fail to operate correctly if the `Br` wire is stuck at 0?

### 1.2 (5 points)

Which instructions fail to operate correctly if the `ALUsrc` wire is stuck at 0?

### 1.3 (5 points)

Which instructions fail to operate correctly if the `RegWrite` wire is stuck at 0?

## 2 (20 points)

Consider adding a new instruction, `SWAP Rd, Rn`, to our architecture, which will swap the data stored in `Rd` and `Rn`. Discuss the necessary changes that must be made to the datapath above. There is no need to design a new datapath - just discuss what should be added to the existing one.

# 3 (15 points)

Consider the addition of a multiplier to the CPU shown in Figure 3.27 in the textbook. This addition will add 200 ps to the latency of the ALU, but will reduce the number of instructions by 20% (because there will no longer be a need to emulate the multiplication instruction). Answer the following questions based on **Chapter 3.3** of the textbook **(no pipelining)** and the following table for the latencies of the stages.

| IF | ID | EX | ME | WB |
|---|---|---|---|---|
| 200 ps | 250 ps | 150 ps | 300 ps | 200 ps |

## 3.1 (5 points)

What is the clock cycle time with and without this improvement?

## 3.2 (5 points)

What is the speedup achieved by adding this improvement? *Hint:* $speedup = 1 - \frac{new\ time}{old\ time}$.

## 3.3 (5 points)

What is the slowest the new ALU can be and still result in improved performance?

# 4 (20 points)

In this exercise, we examine how pipelining affects the clock cycle time of the processor. Questions in this problem assume that individual stages of the datapath have the latencies shown in Problem 3 above. Answer the following questions.

## 4.1 (5 points)

What is the clock cycle time in a pipelined and non-pipelined processor?

## 4.2 (5 points)

What is the total latency of an `LDR` instruction in a pipelined and non-pipelined processor?

## 4.3 (10 points)

If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

## 5 (20 points)

Consider the following instructions executed in a pipeline with full forwarding support (including WB write/read in the same cycle). Identify the value of which register is forwarded from a stage of an instruction to a stage of a subsequent instruction. (Completely impossible example: "The value of X5 is forwarded from the IF stage of instruction 1 to the WB stage of instruction 2.") Submit a pipeline execution diagram as shown in the textbook.

```
LDR X20, [X19, 0]
LDR X21, [X19, 8]
ADD X22, X21, X20
SUB X23, X23, X22
```

# 6 (20 points)

Consider the following instructions.

```
LDR   X1, [X6, 8]
ADD   X0, X1, X0
STR   X0, [X10, 4]
LDR   X2, [X6, 12]
SUB   X3, X0, X2
STR   X3, [X8, 24]
CBZ   X2, 40
```

## 6.1 (10 points)

Add `NOP` instruction to the code above so that it will run correctly on a pipeline **without forwarding**, but WB write/read in the same cycle. (A pipeline execution diagram is not needed for this problem. Sketching it may help, but it will not be graded.)

## 6.2 (10 points)

Optimize the code execution by re-arranging the instructions to get the same correct result faster.