

Pledge: I pledge my honor that I have abided by the Stevens Honor System.

Name: Haig Emirzian

CS-382 Computer Architecture and Organization

Fall 2022

Homework 3 · Microprocessors

Lecturer: Shudong Hao

Date: See Canvas

All update on the homework will be announced on Canvas. Please make sure to check it out often!

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 3.26 in the textbook (or the downloadable datapath diagram on Canvas).

1.1 (5 points)

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

All B, CBZ, and BL would fail

1.2 (5 points)

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

LDR, STR, ORR, ADD, MUL, LSL, LRS, SUB, and all immediate value instructions would fail

1.3 (5 points)

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

All B, LDR, ADD, AND, ORR, BL, and MOV would fail

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.

I would add a new WriteReg and a control signal into the register file to swap the registers. This would happen because once the data in the registers is in the data path, then all I have to do is rewire the inputs into the other register.

3 (15 points)

Consider the addition of a multiplier to the CPU shown in Figure 3.26 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?

Without: $200 + 250 + 150 + 300 + 200 = 1100 \text{ ps}$ **With:** $200 + 250 + 150 + 300 + 200 + 200 = 1300 \text{ ps}$

3.2 (5 points)

What is the speedup achieved by adding this improvement? *Hint: $speedup = 1 - \frac{new\ time}{old\ time}$.*
 $1 - (1040/1100) = \textbf{5\% faster}$

3.3 (5 points)

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

$$\begin{aligned}
 0.8 * x &< 1100 & 1375 - 1100 &= 275 \\
 x &= 1375 \text{ ps} & 275 + 150 &= \textbf{425 ps}
 \end{aligned}$$

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. Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	LDR	STR
45%	20%	20%	15%

Answer the following questions.

4.1 (5 points)

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

Pipelined: 300 ps (The highest latency stage in the non-pipeline) **Non-pipelined: 1100 ps**

4.2 (5 points) (From problem 3)

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

Pipelined: $300 * 5 = 1500$ ps (Multiplied for each stage) **Non-pipelined: 1100 ps**

4.3 (10 points) (From problem 3)

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?

I would split the ME stage because it has the highest latency. The new clock cycle would be 250 ps because that's the next highest latency.

5 (20 points)

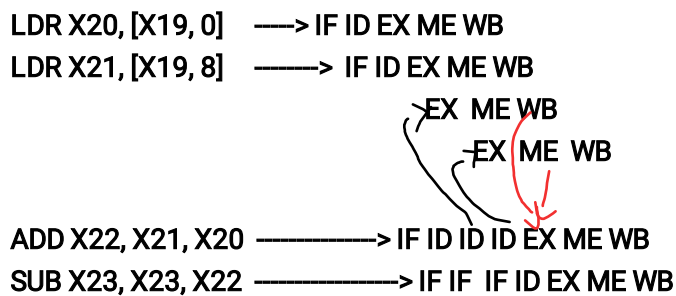
Consider the following instructions executed in a pipeline with full forwarding support (including WB write/read in the same cycle). Identify the value of which register is forwarded from a stage of an instruction to a stage of a subsequent instruction. (Completely impossible example: "The value of X5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

```

1  LDR X20, [X19, 0]
2  LDR X21, [X19, 8]
3  ADD X22, X21, X20
4  SUB X23, X23, X22

```

The value of X21 is forwarded from the ME stage of instruction 2 to the EX stage of instruction 3.



6 (20 points)

Consider the following instructions.

```

1  LDR  X1, [X6, 8]
2  ADD  X0, X1, X0
3  STR  X0, [X10, 4]
4  LDR  X2, [X6, 12]
5  SUB  X3, X0, X2
6  STR  X3, [X8, 24]
7  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.

6.1

```

1 LDR X1, [X6, 8]
2 NOP
3 NOP
4 ADD X0, X1, X0
5 NOP
6 NOP
7 STR X0, [X10, 4]
8 LDR X2, [X6, 12]
9 NOP
10 NOP
11 SUB X3, X0, X2
12 NOP
13 NOP
14 STR X3, [X8, 24]
15 CBZ X2, 40

```

6.2

```

1 LDR X1, [X6, 8]
2 LDR X2, [X6, 12]
3 NOP
4 ADD X0, X1, X0
5 NOP
6 NOP
7 SUB X3, X0, X2
8 NOP
9 STR X0, [X10, 4]
10 STR X3, [X8, 24]
11 CBZ X2, 40

```