Computer Architecture and Programming Languages
Jacobs University Bremen
Neeha Hammad

Course: CO20-320241
18 Novemeber, 2019

# Homework 10

## Problem 10.1

**Solution:**

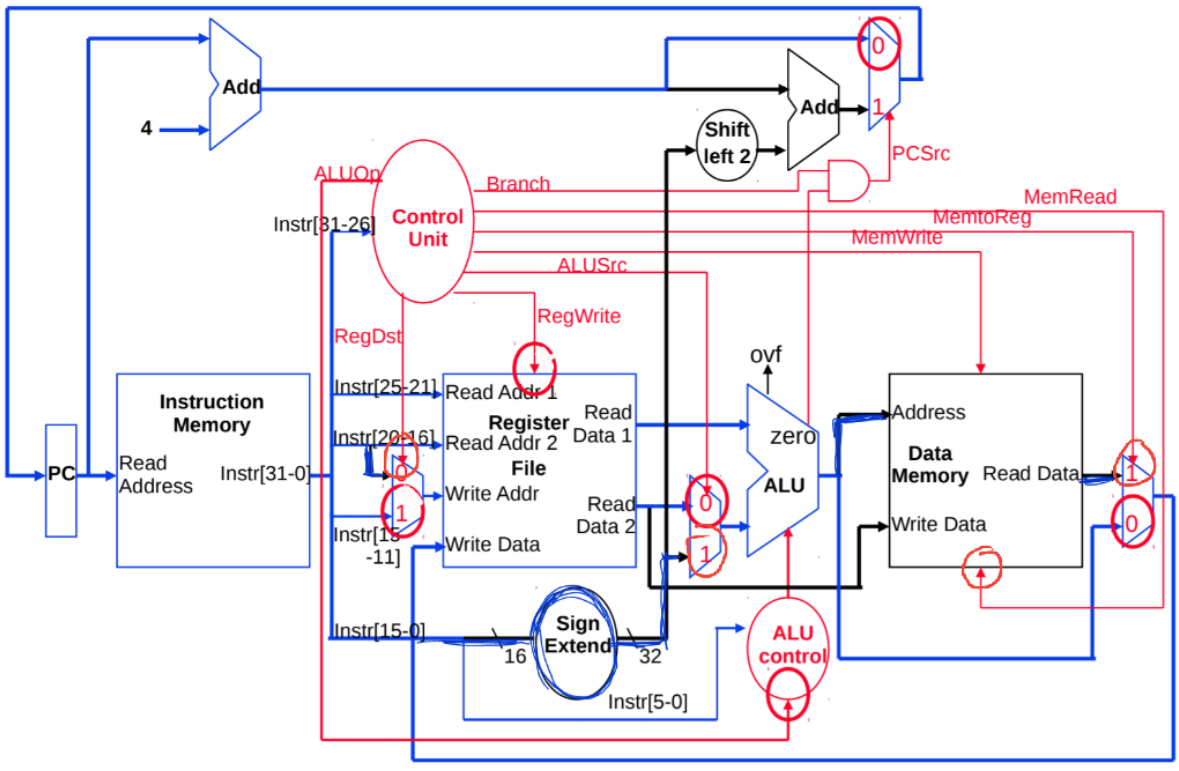
The selection of each input line in a multiplexer is controlled by an additional set of inputs (control lines) and according to the binary condition of these control inputs, either HIGH or LOW the appropriate data input is connected directly to the output. Control lines from the control unit also activate/deactivate memory and registrar manipulations. This is to decide whether we have to read from the memory/registrar or, write to memory/registrar.

In other words, control lines are required within a single cycle datapath so that the multiplexers can determine which input lines should be included in the instruction. Opcode of the instructions also play a part in this decision making.

The interaction between control lines and multiplexers revolves around deciding which of the inputs from the multiplexers will be sent as an output.

## Problem 10.2

**Solution:**




---

This is like a union for R-type single data path and lw single data path.

The add instruction is first read from the memory using the Program Counter and is then executed. During add, data is being read from two registers (rs and rt) and is being written to one other register (rd). Thus, ResDst and WriteReg are activated. At this particular point, ALUSrc (which determines the adequate data path) is at 0. Function field is 100000 while the ALUOp is 10.

The addi instruction is an I type instruction with an immediate value of 16 bits. The instruction is first read from the memory using the Program Counter and is then executed. During addi, data is being read from only 1 register. Thus, RegDst goes to 0 and is deactivated. In the image, it can be seen as the small blue line branching out from Instr[20-16] and going to 0.

Next, the 16 bit immediate value is passed via Sign Extended, which converts it into a 32 bit number that is suitable for the ALU. This new value goes to the multiplexer. At this point, ALUSrc is at 1, which means it is activated. Now, this value, along with the value read from a register earlier, are being evaluated within the ALU. ALUOp is 10. The addition happens here and the resulting value is written back to destination register.

The circles are representing both cases (add and addi). For instance, during add, MemtoReg is deactivated (at 0) or but in case of addi, it is activated. Sorry for the untidy image!

## Problem 10.3

**Solution:**

Depending on the latencies given, we need to select the longest path in terms of time.

(a) For R type instructions, there are four possible paths. We calculate the clock cycle times for all of them and then decide the longest/critical path. The longest possible path includes:
I-mem - RegF - Mux2 - ALU - Mux3 - WBack to RegF

Clock Cycle Time: $450 + 250 + 30 + 120 + 30 + 0 =$ **880ps**

Here, we assume that Mux2 and Mux3 are just Mux and have the same value of 30. If we multiply the value with 2 and 3 respectively, the answer would be different. Additionally, since we are not given a value for WBack to Reg, we take it as 0.

(b) For sw instructions, there are 4 possible paths. The longest one is the following:
I-Mem - RegF - ALU - D-Mem

Clock Cycle Time: $450 + 250 + 120 + 350 =$ **1170ps**

Note that those which appear longer do not always have the longest time. For instance, the path: I-Mem - Sign-Extend - Mux2 - ALU - D-Mem has a clock time of 1000 ps only.

(c) In this case we calculate the clock times for all paths for all types of instructions and find out that the following path for lw is the longest one and thus, we select that:

I-Mem - RegF - ALU - D-Mem - Mux3 - WBack to RegF

Clock Cycle Time: $450 + 250 + 120 + 350 + 30 + 0 =$ **1200ps**

Since we are not given a value for WBack to RegF, we take it as 0..