Computer Architecture and Programming Languages
Jacobs University Bremen
Jovan Shandro

Course: CO20-320241
November 25, 2019

Homework 10

Problem 10.1

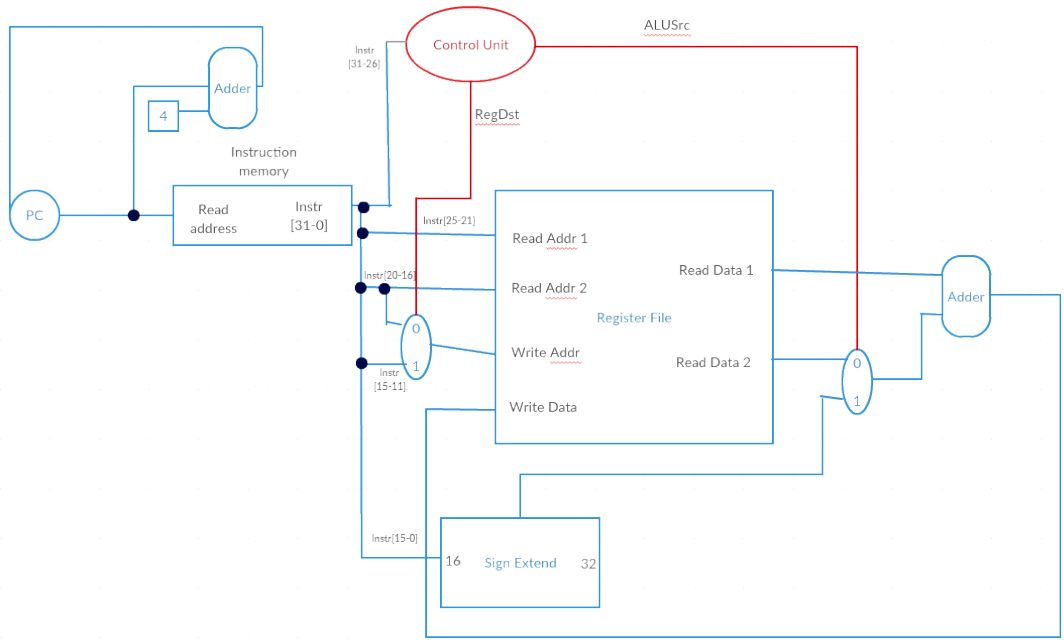
Solution:

Control lines are necessary in a single cycle datapath as they control which path in the whole datapath is followed by a certain instruction. This includes controlling the operation of the ALU, checking if the there will be branching or not, if the instruction requires or not to write the output in the register, if the instruction requires to read from or write to the memory and so on. The way through which the control lines make this possible is through the Multiplexors, with which the control lines interact by sending a single input which decides which of the two inputs of the Multiplexor will be chosen as an output. This helps shorten the implementation of the datapath as many instruction paths have certain parts in common. (0 selects the first input and 1 selects the second).

Problem 10.2

Solution:

One such datapath would be:



Now concerning the explanation. Since one of the instructions is of type R and the other of type I, the first writes the result in the destination register (rd), whereas the other in a source register (rt), so a multiplexor is needed for deciding this and the control line RegDst is also needed to make the decision. Also since both instructions use only addition as the operation we can easily substitute the ALU with a parallel adder (for 32 bit numbers). ALso, a second multiplexor is needed to decide whether the second value to be added is found in a source register (for add) or as an immediate value extended to 32 bits (for addi). So we also need the ALUSrc control line to decide on the output of this multiplexor. Since no further reading or writing from memory takes place for both instructions, these are the only 2 control lines needed. At last, since there is no branching, the PC just changes by 4 to go to the next instruction.

Problem 10.3

Solution:

a) As mentioned in the slides there are 4 possible paths for ALU operations :

1. I-Mem → RegF → ALU → Mux3 → WBack to RegF
2. I-Mem → RegF → Mux2 → ALU → Mux3 → WBack to RegF
3. I-Mem → Mux1 → RegF (Write address)
4. Add → Mux4 → PCWrite (update PC)

So, according to the given table of latencies we calculate the clock cycle time in ps for the 4 paths above :

Path	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2	Total
1	450	0	30	120	250 ·2	0	0	0	1100 ps
2	450	0	30 ·2	120	250 ·2	0	0	0	1130 ps
3	450	0	30	0	250	0	0	0	730 ps
4	0	110 ·2	30	0	0	0	0	0	250 ps

From the table above we see that the longest possible path takes 1130 ps, which decides the clock cycle time.

b) Again as mentioned in the slides there are 4 possible paths for the sw instruction :

1. I-Mem → RegF → ALU → D-Mem
2. I-Mem → RegF → D-Mem
3. I-Mem → Sign-Extend → Mux2 → ALU → D-Mem
4. Add → Mux4 → PCWrite (update PC)

So, according to the given table of latencies we calculate the clock cycle time in ps for the 4 paths above :

Path	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2	Total
1	450	0	0	120	250	350	0	0	1170 ps
2	450	0	0	0	250	350	0	0	1050 ps
3	450	0	30	120	0	350	20	0	970 ps
4	0	110 ·2	30	0	0	0	0	0	250 ps

From the table above we see that the longest possible path takes 1170 ps, which decides the clock cycle time.

c) From the two previous points we know that the clock cycle times for the add and sw instructions are 1130 and 1170 ps respectively. So we only need to find the clock cycle times of beq and lw to find the clock cycle time in case we want to support all four instructions. From the slides we know that the beq instruction can take the following paths:

1. I-Mem → RegF → ALU → Mux4 → PCWrite (update PC)
2. I-Mem → RegF → Mux2 → ALU → Mux4 → PCWrite (update PC)
3. I-Mem → Sign-Extend → Shift-left-2 → Add → Mux4 → PCWrite (update PC)
4. Add → Mux4 → PCWrite (update PC)

So, according to the given table of latencies we calculate the clock cycle time in ps for the 4 paths above :

Path	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2	Total
1	450	110	30	120	250	0	0	0	960 ps
2	450	110	30 ·2	120	250	0	0	0	980 ps
3	450	110 ·2	30	0	0	0	20	0	700 ps
4	0	110 ·2	30	0	0	0	0	0	250 ps

From the table above we see that the longest possible path takes 980 ps, which decides the clock cycle time for the beq instruction.

Now for the lw instruction. From the slides we know that the lw instruction can take the following paths:

1. I-Mem → RegF → ALU → D-Mem → Mux3 → WBack to RegF
2. I-Mem → Mux1 → RegF (Write address)
3. I-Mem → Sign-Extend → Mux2 → ALU → D-Mem → Mux3 → WBack to RegF
4. Add → Mux4 → PCWrite (update PC)

So, according to the given table of latencies we calculate the clock cycle time in ps for the 4 paths above :

Path	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2	Total
1	450	0	30	120	250 ·2	350	0	0	1450 ps
2	450	0	30	0	250	0	0	0	730 ps
3	450	0	30 ·2	120	250	350	20	0	1250 ps
4	0	110 ·2	30	0	0	0	0	0	250 ps

From the table above we see that the longest possible path takes 1450 ps, which decides the clock cycle time for the lw instruction.

So the clock cycle time is 1450 ps, as it is determined by the longest path out of the 4 instructions.