



Ve370 Introduction to Computer Organization

Homework 5

1. Exercise 4.16.1
2. Exercise 4.16.2
3. Exercise 4.16.3

Exercise 4.16

The first three problems in this exercise refer to the following MIPS instruction:

	Instruction
a.	SW R16, -100(R6)
b.	OR R2, R1, R0

4.16.1 [5] <4.6> As this instruction executes, what is kept in each register located between two pipeline stages?

4.16.2 [5] <4.6> Which registers need to be read, and which registers are actually read?

4.16.3 [5] <4.6> What does this instruction do in the EX and MEM stages?

4. Exercise 4.16.4
5. Exercise 4.16.5
6. Exercise 4.16.6

The remaining three problems in this exercise refer to the following loop. Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, and that the pipeline has full forwarding support. Also assume that many iterations of this loop are executed before the loop exits.

	Loop
a.	Loop: ADD R1,R2,R1 LW R2,0(R1) LW R2,16(R2) SLT R1,R2,R4 BEQ R1,R9,Loop
b.	Loop: LW R1,0(R1) AND R1,R1,R2 LW R1,0(R1) LW R1,0(R1) BEQ R1,R0,Loop

4.16.4 [10] <4.6> Show a pipeline execution diagram for the third iteration of this loop, from the cycle in which we fetch the first instruction of that iteration up to (but not including) the cycle in which we can fetch the first instruction of the next iteration. Show all instructions that are in the pipeline during these cycles (not just those from the third iteration).

4.16.5 [10] <4.6> How often (as a percentage of all cycles) do we have a cycle in which all five pipeline stages are doing useful work?

4.16.6 [10] <4.6> At the start of the cycle in which we fetch the first instruction of the third iteration of this loop, what is stored in the IF/ID register?

7. Exercise 4.17.3

Exercise 4.17

Problems in this exercise assume that instructions executed by a pipelined processor are broken down as follows:

	ADD	BEQ	LW	SW
a.	40%	30%	25%	5%
b.	60%	10%	20%	10%

4.17.3 [5] <4.6> Assuming there are no stalls, how often (percentage of all cycles) do we use the data memory?

8. Exercise 4.18.1

9. Exercise 4.18.3

Exercise 4.18

The first three problems in this exercise refer to the execution of the following instruction in the pipelined datapath from Figure 4.51, and assume the following clock cycle time, ALU latency, and Mux latency:

	Instruction	Clock Cycle Time	ALU Latency	Mux Latency
a.	LW R1, 32(R2)	50ps	30ps	15ps
b.	OR R1, R5, R6	200ps	170ps	25ps

4.18.1 [10] <4.6> For each stage of the pipeline, what are the values of the control signals asserted by this instruction in that pipeline stage?

4.18.2 [10] <4.6, 4.7> How much time does the control unit have to generate the ALUSrc control signal? Compare this to a single-cycle organization.

4.18.3 What is the value of the PCSrc signal for this instruction? This signal is generated early in the MEM stage (only a single AND gate). What would be a reason in favor of doing this in the EX stage? What is the reason against doing it in the EX stage?

10. Exercise 4.20.1

11. Exercise 4.20.3

12. Exercise 4.20.5

13. Exercise 4.20.6

Exercise 4.20

Problems in this exercise refer to the following instruction sequences:

	Instruction Sequence
a.	ADD R1, R2, R1 LW R2, 0(R1) LW R1, 4(R1) OR R3, R1, R2
b.	LW R1, 0(R1) AND R1, R1, R2 LW R2, 0(R1) LW R1, 0(R3)

4.20.1 [5] <4.7> Find all data dependences in this instruction sequence.

4.20.3 [10] <4.7> To reduce clock cycle time, we are considering a split of the MEM stage into two stages. Repeat 4.20.2 for this 6-stage pipeline.

The remaining three problems in this exercise assume that, before any of the above is executed, all values in data memory are zeroes and that registers R0 through R3 have the following initial values:

	R0	R1	R2	R3
a.	0	-1	31	1500
b.	0	4	63	3000

4.20.5 [10] <4.7> If we assume forwarding will be implemented when we design the hazard detection unit, but then we forget to actually implement forwarding, what are the final register values after this instruction sequence?

4.20.6 [10] <4.7> For the design described in 4.20.5, add NOPs to this instruction sequence to ensure correct execution in spite of missing support for forwarding.