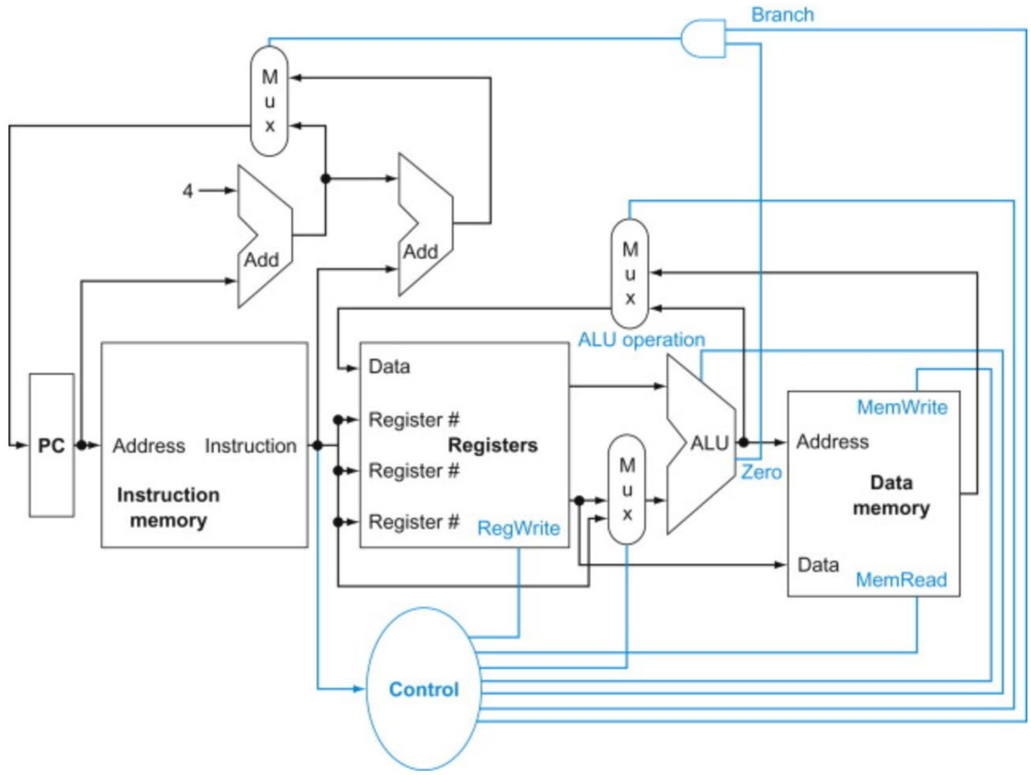
CECS341-practise problems



COD Figure 4.2. The basic implementation of the MIPS subset, including the necessary multiplexors and control lines:

- ALUMux is the control signal that controls the Mux at the ALU input, 0 (Reg) selects the output of the register fi le, and 1 (Imm) selects the immediate from the instruction word as the second input to the ALU.
- RegMux is the control signal that controls the Mux at the Data input to the register fi le, 0 (ALU) selects the output of the ALU, and 1 (Mem) selects the output of memory.
- A value of X is a “don’t care” (does not matter if signal is 0 or 1)

1. COD 4.1 Consider the following instruction:
   Instruction: AND Rd, Rs, Rt
   Interpretation: Reg[Rd] = Reg[Rs] AND Reg[Rt]

   4.1.1 [5] <COD §4.1> What are the values of control signals generated by the control in COD Figure 4.2 (below) for the above instruction?

4.1.2 [5] <COD §4.1> Which resources (blocks) perform a useful function for this instruction?

4.1.3 [10] <COD §4.1> Which resources (blocks) produce outputs, but their outputs are not used for this instruction? Which resources produce no outputs for this instruction?

2. COD 4.3 When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. In the following three problems, assume that we are starting with a datapath from COD Figure 4.2 (The basic implementation of the MIPS subset …), where I-Mem, Add, Mux, ALU, Regs, D-Mem, and Control blocks have latencies of 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps, and 100 ps, respectively, and costs of 1000, 30, 10, 100, 200, 2000, and 500, respectively.

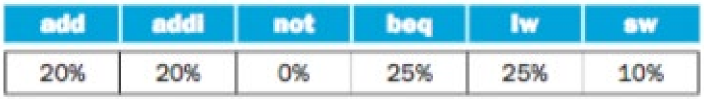
   Consider the addition of a multiplier to the ALU. This addition will add 300 ps to the latency of the ALU and will add a cost of 600 to the ALU. The result will be 5% fewer instructions executed since we will no longer need to emulate the MUL instruction.

   4.3.1 [10] <COD §4.1> What is the clock cycle time with and without this improvement?

   4.3.2 [10] <COD §4.1> What is the speedup achieved by adding this improvement?

   4.3.3 [10] <COD §4.1> Compare the cost/performance ratio with and without this improvement.

3. COD 4.5 For the problems in this exercise, assume that there are no pipeline stalls and that the breakdown of executed instructions is as follows:

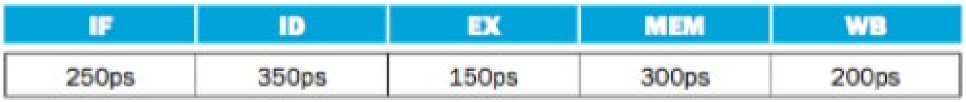


add	addi	not	beq	lw	sw
20%	20%	0%	25%	25%	10%

   4.5.1 [10] <COD §4.3> In what fraction of all cycles is the data memory used?

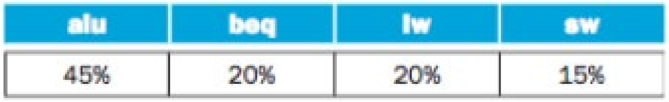
   4.5.2 [10] <COD §4.3> In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its input is not needed?

4. 4.8 In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:



IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

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



alu	beq	lw	sw
45%	20%	20%	15%

4.8.1 [5] <COD §4.5> What is the clock cycle time in a pipelined and non-pipelined processor?

4.8.2 [10] <COD §4.5> What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

4.8.3 [10] <COD §4.5> 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?

4.8.4 [10] <COD §4.5> Assuming there are no stalls or hazards, what is the utilization of the data memory?

4.8.5 [10] <COD §4.5> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?