113 計算機組織 Homework 4-1　　Due date: YYYY/MM/DD.

- You must write down / type the calculations, or you won't get the scores.
- 請用手寫於 A4 紙上,並掃描上傳至 ZUVIO (提供雲端連結)
- 請勿抄襲，抄襲者與被抄襲者單一分數取平均

一、Given a processor implemented by a single-cycle control and datapath shown in Figure 1. Assume that the functional blocks adopted to implement the datapath have the following latencies in Table 1, and we need to design a processor that can support the MIPS instructions listed in Table 2:

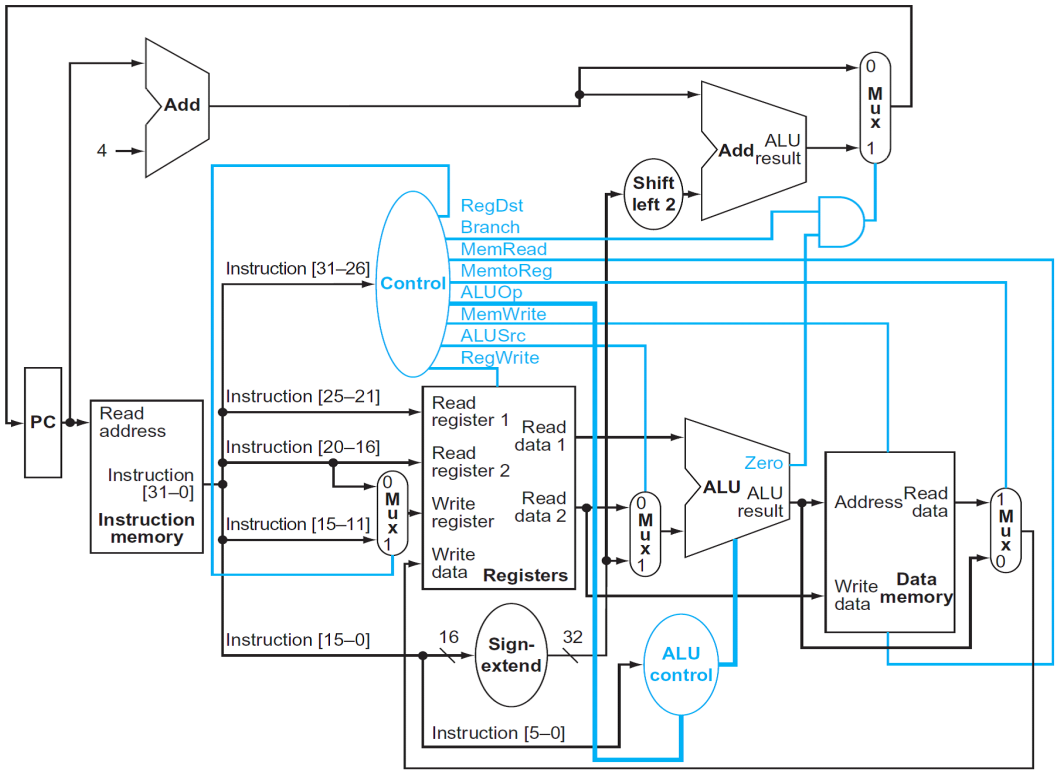


Figure 1. The single-cycle datapath with the control unit

Table 1. Latencies of the functional units on the datapath

PC's Clk.-to-Q	I-Mem.	Add	Mux	ALU	Ctrl. Unit	ALU-Ctrl.	Regs. Access	Data-Mem.	Sign-extend	AND-Gate	Shift-left-2
0ps	180ps	80ps	30ps	100ps	200ps	60ps	100ps	200ps	20ps	60ps	20ps

Table 2. MIPS Instruction Opcode & Funct fields

Instr.	Opcode	Funct
add	0_{10}	32_{10}
sub	0_{10}	34_{10}
beq	4_{10}	
lw	35_{10}	
sw	43_{10}	

1.1 If the only thing we need to do in a processor is fetch consecutive instructions (Figure 4.6), what would the cycle time be?

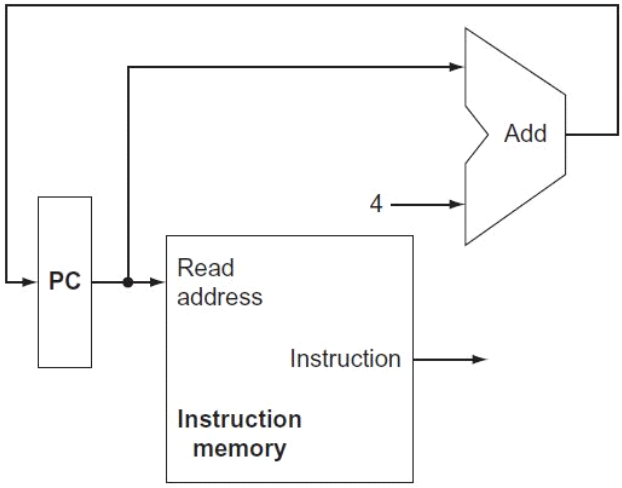


FIGURE 4.6 A portion of the datapath used for fetching instructions and incrementing the program counter. The fetched instruction is used by other parts of the datapath.

1.2. If this processor only need to support **sw** and **beq** and **add**instructions, what is the **clock cycle time**? (Hint: You should analyze the critical path of the required instructions)

1.3. What is the **clock cycle time** of this processor that support the instructions listed in Table 2? (Hint: lw 最長 path 請使用下面公式計算)

I-Mem + Ctrl Unit + ALU Ctrl + ALU + Data-Mem + Mux + Regs. Write

二、In this exercise we examine in detail how an instruction is executed in a single-cycle datapath. Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction word:

0000 0001 0110 1001 0101 0000 0010 0010

Assume that **data memory is all zeros** and **that the processor's registers have the following values** at the beginning of the cycle in which the above instruction word is fetched (Please refer to the MIPS Instruction Opcode & Funct fields listed in ***Table 2***.):

r0	*r1*	*r2*	*r3*	*r4*	*r5*	*r6*	*r8*	*r9*	*r10*	*r11*	*r31*
0	-1	2	-3	-4	10	6	8	3	8	4	-16

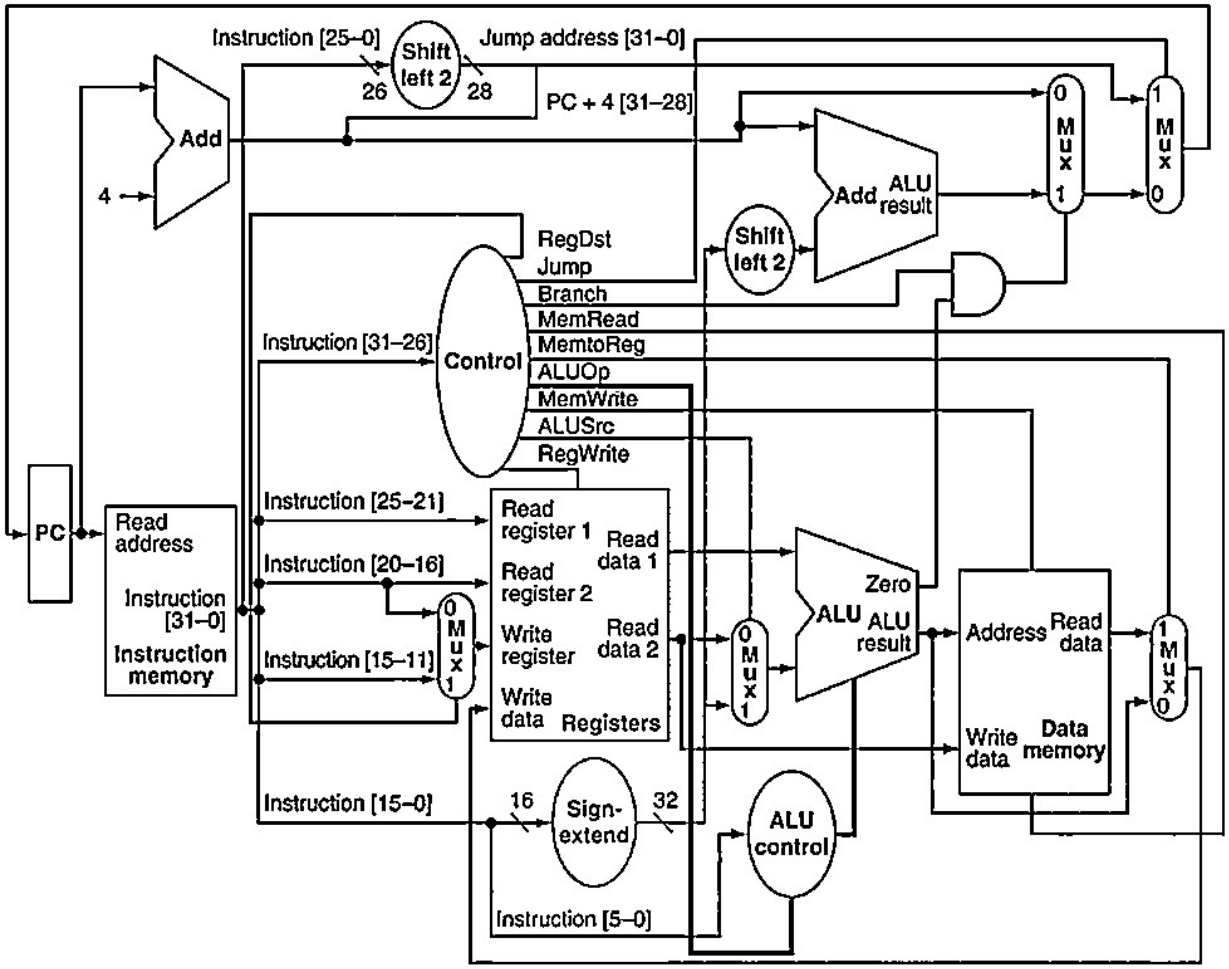


Figure 4.24

2.1. What are the outputs of the sign-extend and the jump “Shift left-2” unit (near the top of Figure 4.24) for this instruction word?

2.2. What are the values of the ALU control unit’s inputs for this instruction?

2.3. What is the new PC address after this instruction is executed? Highlight the path through which this value is determined.

2.4. For each Mux, show the values of its data output during the execution of this instruction and these register values.

RegDst Mux	ALUsrc Mux	PCsrc Mux	MemtoReg Mux
(1)	(2)	(3)	(4)

2.5. For the ALU and the two add units, what are their data input values?

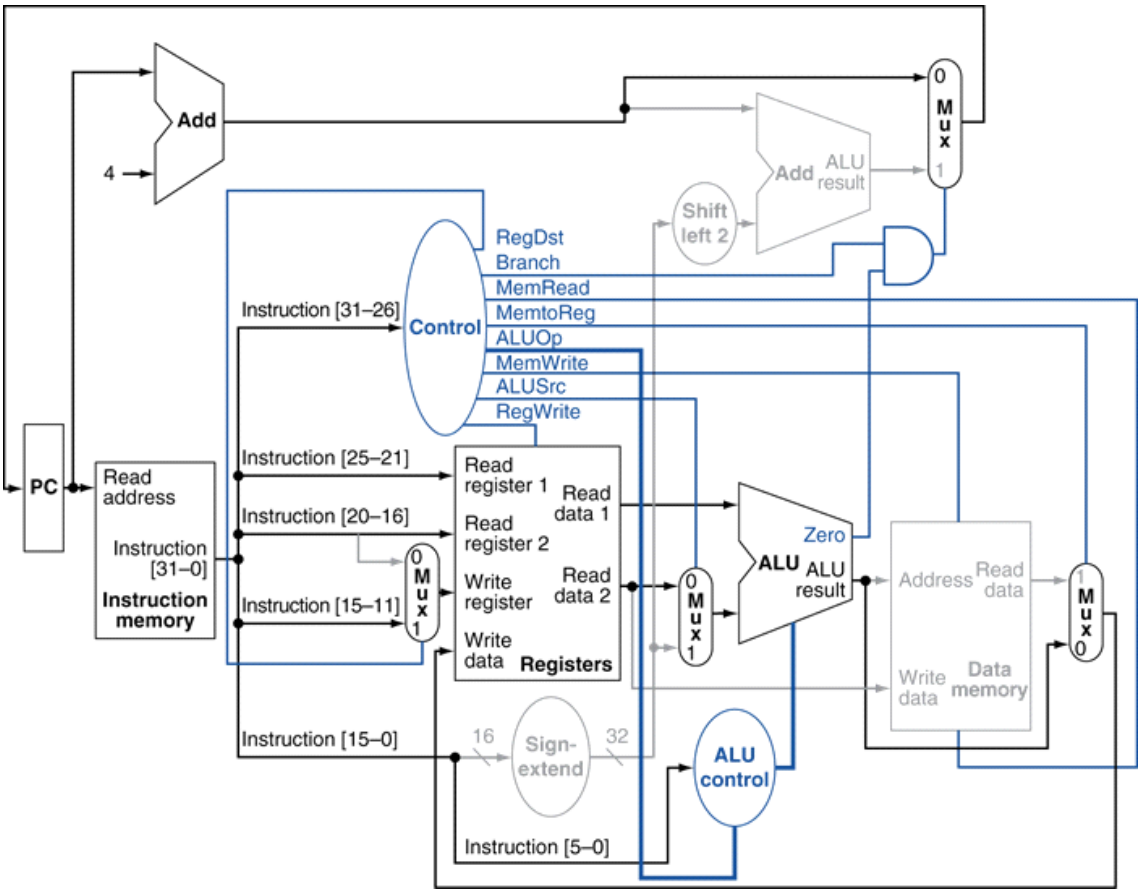
ALU(ALU result)	Add	Add (ALU result)
(1)	(2)	(3)

2.6. What are the values of all inputs for the “Registers” unit?

Read Register 1	Read Register 2	Write Register	Write Data	RegWrite
(1)	(2)	(3)	(4)	(5)

三. Given a processor implemented by a single-cycle control and datapath shown in follow. Assume that the functional blocks adopted to implement the datapath have the following latencies:

Inst.-Mem.	Add	Mux	ALU	ALU-Ctrl	Regs. Access	Data-Mem.	Sign-extend	Shift-left-2
500ps	350ps	100ps	300ps	100ps	350ps	600ps	100ps	50ps



3.1. For add instruction's critical path shown in Figure, **which control signal is the most critical** that is needed to be quickly generated, and **what is the maximum time** that the control unit can generate this signal to avoid lengthening the critical path? (Hint: the critical path of add instruction)

3.2. Assume that the control unit shown in Figure requires the times to generate the individual control signals as the following table. What is the clock cycle time of this processor?

RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1300ps	1000ps	1000ps	600ps	1000ps	500ps	900ps	350ps	1300ps

四. When processor designers consider a possible improvement to the processor data path, the decision usually depends on the cost/performance trade-off . In the following three problems, assume that we are starting with a data path from Figure 4.2, where I-Mem, Add, Mux, ALU, Registers, D-Mem, and Control blocks have latencies of 500ps, 200ps, 50ps, 150ps, 250ps, 400ps, and 100ps, respectively, and costs of 1100, 30, 20, 80, 200, 1600, and 500 (dollars), respectively. Consider the addition of a multiplier to the ALU. This addition will add 300ps to the latency of the ALU and will add a cost of 500 dollars to the ALU. The result will be 5% fewer instructions executed since we will no longer need to emulate the MUL instruction.

4.1 What is the clock cycle time with and without this improvement?

4.2 What is the speedup achieved by adding this improvement?

4.3 Compare the cost/performance ratio with and without this improvement.

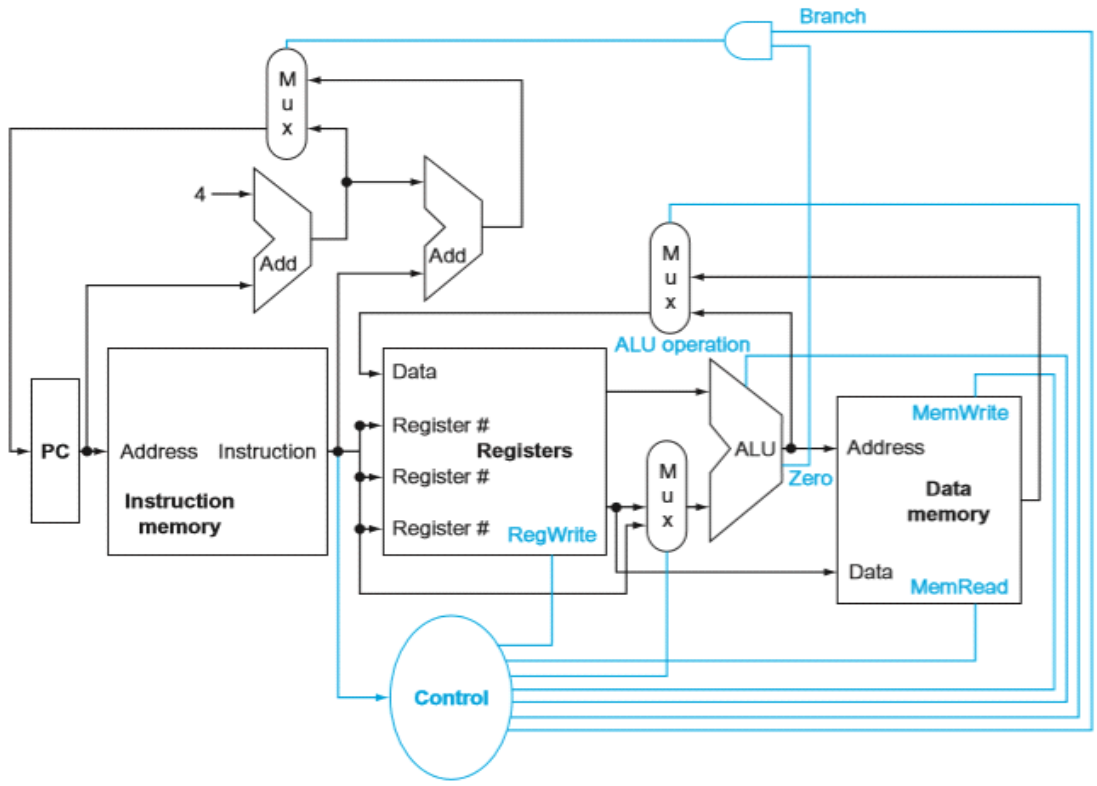


FIGURE 4.2 The basic implementation of the MIPS subset, including the necessary multiplexors and control lines. The top multiplexor ("Mux") controls what value replaces the PC (PC + 4 or the branch destination address); the multiplexor is controlled by the gate that "ANDs" together the Zero output of the ALU and a control signal that indicates that the instruction is a branch. The middle multiplexor, whose output returns to the register file, is used to steer the output of the ALU (in the case of an arithmetic-logical instruction) or the output of the data memory (in the case of a load) for writing into the register file. Finally, the bottommost multiplexor is used to determine whether the second ALU input is from the registers (for an arithmetic-logical instruction or a branch) or from the offset field of the instruction (for a load or store). The added control lines are straightforward and determine the operation performed at the ALU, whether the data memory should read or write, and whether the registers should perform a write operation. The control lines are shown in color to make them easier to see.