

Assigned: Tuesday 3 November

Due: Monday 9 November 10:30 pm

Collaboration: This assignment must be completed individually.

Submission: Upload a PDF with your name, box number, and all answers to PioneerWeb.

1. Assume that logic blocks needed to implement a MIPS processor's datapath (as shown in Figures 4.11 and 4.17) have the following latencies. Explain each answer in complete sentences.

I-mem	Add	MUX	ALU	Regs	D-Mem	Sign-extend	Shift-left-2
400 ps	100 ps	30 ps	120 ps	200 ps	350 ps	20 ps	2 ps

- (a) What is the time required to update the PC?
 - (b) What is the time required to execute the `addi` instruction?
 - (c) What is the time required to execute the `lw` instruction?
 - (d) What is the smallest cycle time allowing all supported instructions to complete within one cycle?
 - (e) How much time does the control unit have to generate the `MemRead` signal, without slowing down the critical path for the `lw` instruction?
 - (f) How much time does the control unit have to generate `ALUSrc` signal, without slowing down the critical path for *any* instruction?
2. Consider the following MIPS instruction word: 0001 0000 0010 0011 0000 0000 0000 1100
 - (a) Give and briefly explain the seven control signal settings for this instruction (cf. Figure 4.16).
 - (b) Give and briefly explain the four ALU control line values for this instruction (cf. p. 316).
 3. Give and briefly explain the control line settings for the MIPS `addi` and `j` (jump) instructions needed in the truth table shown below. See textbook Figures 4.23–4.24 for a discussion of adding the jump instruction to the datapath.

	Signal name	R-format	<code>addi</code>	<code>lw</code>	<code>sw</code>	<code>beq</code>	<code>j</code>
Inputs	<code>Op5</code>	0	0	1	1	0	0
	<code>Op4</code>	0	0	0	0	0	0
	<code>Op3</code>	0	1	0	1	0	0
	<code>Op2</code>	0	0	0	0	1	0
	<code>Op1</code>	0	0	1	1	0	1
	<code>Op0</code>	0	0	0	1	1	0
Outputs	<code>RegDst</code>	1		0	X	X	
	<code>ALUSrc</code>	0		1	1	0	
	<code>MemToReg</code>	0		1	X	X	
	<code>RegWrite</code>	1		1	0	0	
	<code>MemRead</code>	0		1	0	0	
	<code>MemWrite</code>	0		0	1	0	
	<code>Branch</code>	0		0	0	1	
	<code>Jump</code>	0		0	0	0	
	<code>ALUOp0</code>	1		0	0	0	
	<code>ALUOp1</code>	0		0	0	1	

Acknowledgments This derivative work of Janet Davis, used under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License was developed in collaboration with Charlie Curtsinger. The table in Problem 3 is adapted from Patterson & Hennessy (2008), Figure 4.22 (p. 327).¹

¹ Patterson, D. A., & Hennessy, J. L. (2008). *Computer organization and design: The hardware/software interface* (Fourth Edition). Morgan Kaufmann: Burlington, MA.