

CS219 Worksheet 2 (Ch4)

Purpose: Become familiar with the MIPS architecture instruction formats, control signals, Datapath elements.

Points: 75

Reading/References: Chapter 4, Class Lecture / Lecture Notes, Textbook

- Explain and trace/highlight the Datapath for the given R and I format instructions - **beq**, **ori**, **lw**, **sw**, Update the control signal table for each instruction. Rewrite the instruction using register format, for example change add into add rd, rs, rt. Use fig 4.17 for this question and you will need a separate picture for each instruction.
(Note: Use different color sketch/pen to easily identify the path of each instruction.)

[40 pts, 10 pts each]

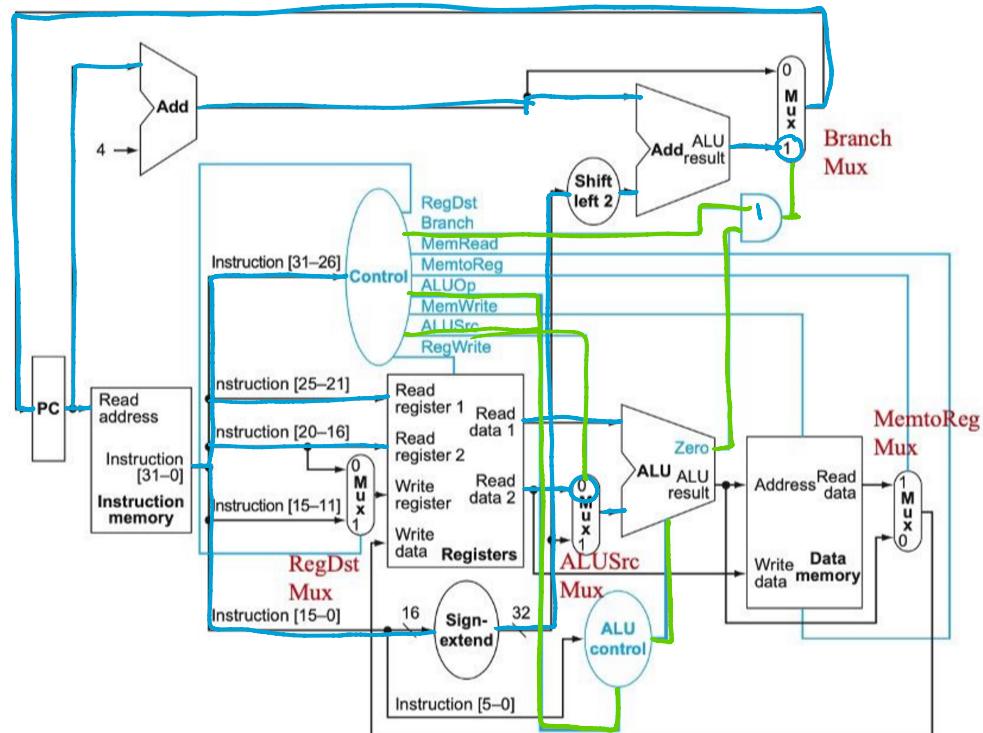
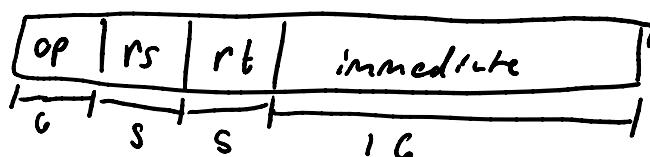


FIGURE 4.17 The simple datapath with the control unit. The input to the control unit is the 6-bit opcode field from the instruction. The outputs of the control unit consist of three 1-bit signals that are used to control multiplexors (RegDst, ALUSrc, and MemtoReg), three signals for controlling reads and writes in the register file and data memory (RegWrite, MemRead, and MemWrite), a 1-bit signal used in determining whether to possibly branch (Branch), and a 2-bit control signal for the ALU (ALUOp). An AND gate is used to combine the branch control signal and the Zero output from the ALU; the AND gate output controls the selection of the next PC. Notice that PCSrc is now a derived signal, rather than one coming directly from the control unit. Thus, we drop the signal name in subsequent figures.

beq rs, rt , immediate



CS219 Worksheet 2 (Ch4)

Purpose: Become familiar with the MIPS architecture instruction formats, control signals, Datapath elements.

Points: 75

Reading/References: Chapter 4, Class Lecture / Lecture Notes, Textbook

- Explain and trace/highlight the Datapath for the given R and I format instructions - **beq, ori, lw, sw**, Update the control signal table for each instruction. Rewrite the instruction using register format, for example change add into add rd, rs, rt. Use fig 4.17 for this question and you will need a separate picture for each instruction.
(Note: Use different color sketch/pen to easily identify the path of each instruction.)

[40 pts, 10 pts each]

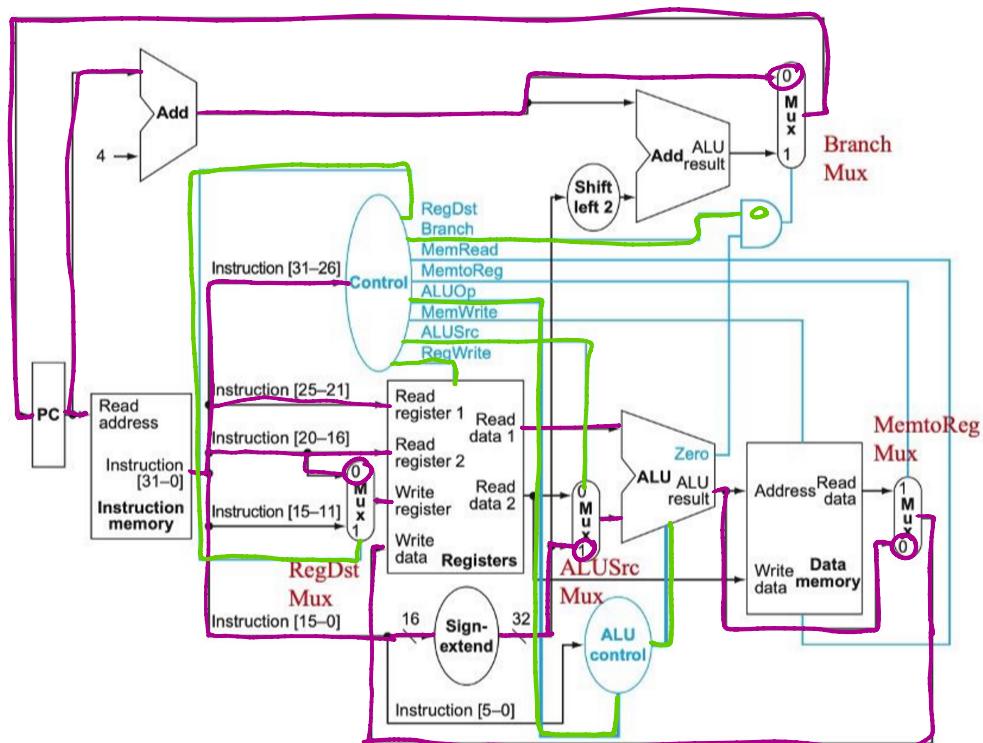
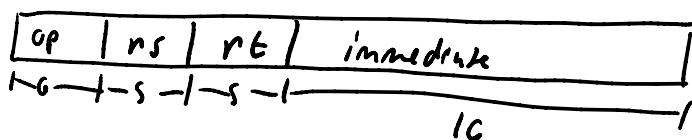


FIGURE 4.17 The simple datapath with the control unit. The input to the control unit is the 6-bit opcode field from the instruction. The outputs of the control unit consist of three 1-bit signals that are used to control multiplexors (RegDst, ALUSrc, and MemtoReg), three signals for controlling reads and writes in the register file and data memory (RegWrite, MemRead, and MemWrite), a 1-bit signal used in determining whether to possibly branch (Branch), and a 2-bit control signal for the ALU (ALUOp). An AND gate is used to combine the branch control signal and the Zero output from the ALU; the AND gate output controls the selection of the next PC. Notice that PCSrc is now a derived signal, rather than one coming directly from the control unit. Thus, we drop the signal name in subsequent figures.

ori rt, rs , immediate



CS219 Worksheet 2 (Ch4)

Purpose: Become familiar with the MIPS architecture instruction formats, control signals, Datapath elements.

Points: 75

Reading/References: Chapter 4, Class Lecture / Lecture Notes, Textbook

- Explain and trace/highlight the Datapath for the given R and I format instructions - **beq**, **ori**, **lw**, **sw**, Update the control signal table for each instruction. Rewrite the instruction using register format, for example change add into add rd, rs, rt. Use fig 4.17 for this question and you will need a separate picture for each instruction.
(Note: Use different color sketch/pen to easily identify the path of each instruction.)

[40 pts, 10 pts each]

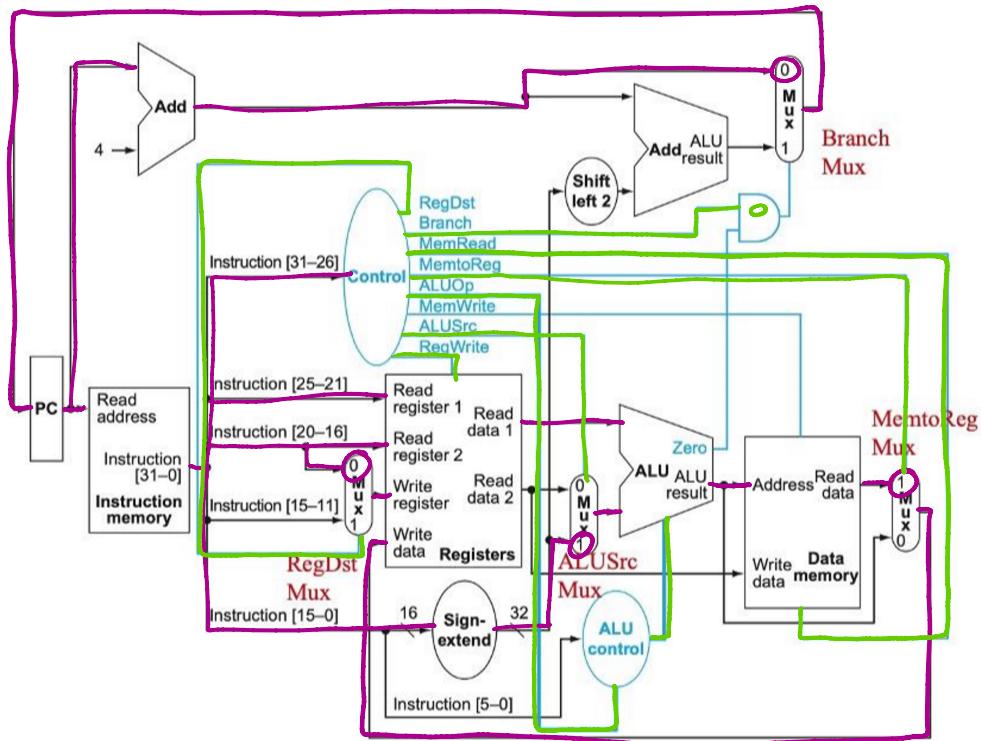


FIGURE 4.17 The simple datapath with the control unit. The input to the control unit is the 6-bit opcode field from the instruction. The outputs of the control unit consist of three 1-bit signals that are used to control multiplexors (RegDst, ALUSrc, and MemtoReg), three signals for controlling reads and writes in the register file and data memory (RegWrite, MemRead, and MemWrite), a 1-bit signal used in determining whether to possibly branch (Branch), and a 2-bit control signal for the ALU (ALUOp). An AND gate is used to combine the branch control signal and the Zero output from the ALU; the AND gate output controls the selection of the next PC. Notice that PCSrc is now a derived signal, rather than one coming directly from the control unit. Thus, we drop the signal name in subsequent figures.

lw rt, offset(rs)



CS219 Worksheet 2 (Ch4)

Purpose: Become familiar with the MIPS architecture instruction formats, control signals, Datapath elements.

Points: 75

Reading/References: Chapter 4, Class Lecture / Lecture Notes, Textbook

- Explain and trace/highlight the Datapath for the given R and I format instructions - **beq, ori, lw, sw**, Update the control signal table for each instruction. Rewrite the instruction using register format, for example change add into add rd, rs, rt. Use fig 4.17 for this question and you will need a separate picture for each instruction.
(Note: Use different color sketch/pen to easily identify the path of each instruction.)

[40 pts, 10 pts each]

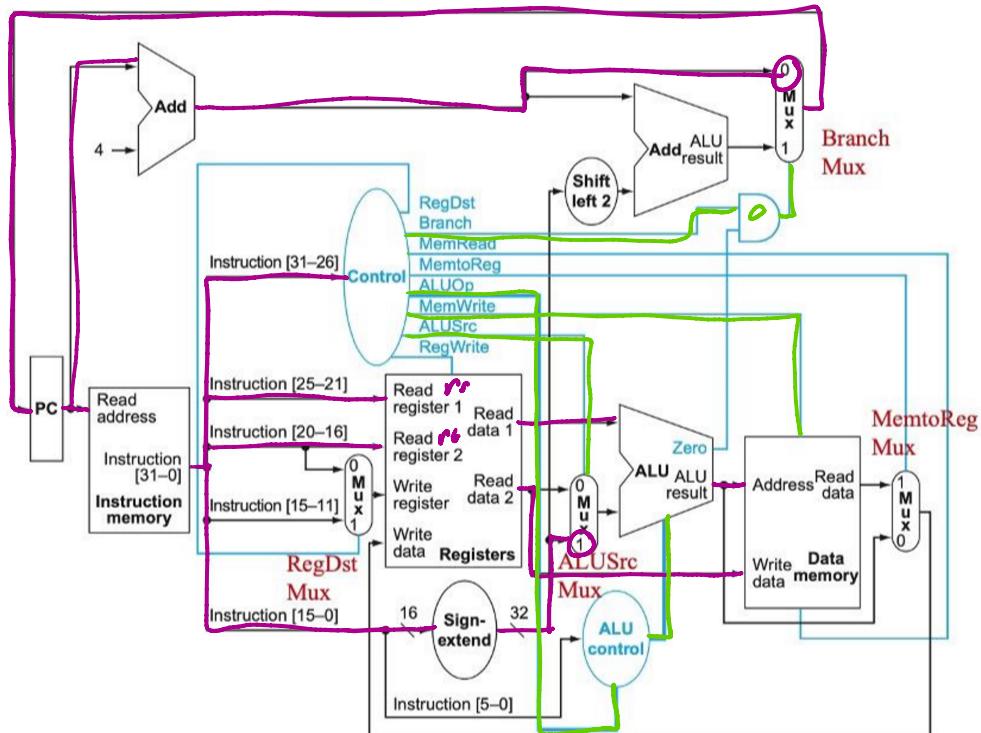
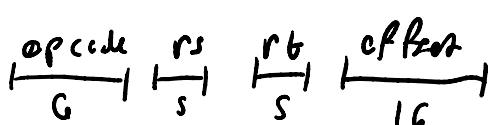


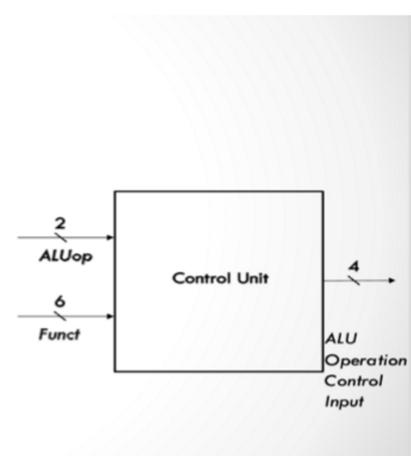
FIGURE 4.17 The simple datapath with the control unit. The input to the control unit is the 6-bit opcode field from the instruction. The outputs of the control unit consist of three 1-bit signals that are used to control multiplexors (RegDst, ALUSrc, and MemtoReg), three signals for controlling reads and writes in the register file and data memory (RegWrite, MemRead, and MemWrite), a 1-bit signal used in determining whether to possibly branch (Branch), and a 2-bit control signal for the ALU (ALUOp). An AND gate is used to combine the branch control signal and the Zero output from the ALU; the AND gate output controls the selection of the next PC. Notice that PCSrc is now a derived signal, rather than one coming directly from the control unit. Thus, we drop the signal name in subsequent figures.

sw rt, offset(rs)



ALU CONTROL LINES

Opcode	ALU op	Operation	Funct	ALU action	ALU Control Input
lw	00	Load word	N/A	add	0010
sw	00	Store word	N/A	add	0010
beq	01	Branch equal	N/A	subtract	0110
R-type	10	Add	100000	add	0010
R-type	10	Subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	Set on less than	101010	slt	0111



Instruction	RegDst Mux	ALUSrc Mux	MemtoReg Mux	RegWrite	MemRead	MemWrite	Branch	Branch Mux	ALUop1 (MSB)	Aluop0 (LSB)
beq	X	0	X	0	0	0	1	1	0	1
ori	0	1	0	1	X	X	0	0	1	0
lw	0	1	1	1	1	0	0	0	0	0
sw	X	1	X	0	0	1	0	0	0	0

2) Explain and trace/highlight the Datapath for the given instructions:

- (i) j Label
- (ii) ori rt, rs, imm

Use fig 4.24 for the above questions and you will need a separate picture for each instruction. (Note: Use different color sketch/pen to easily identify the path of each instruction.) [20 pts, 10 pts each]

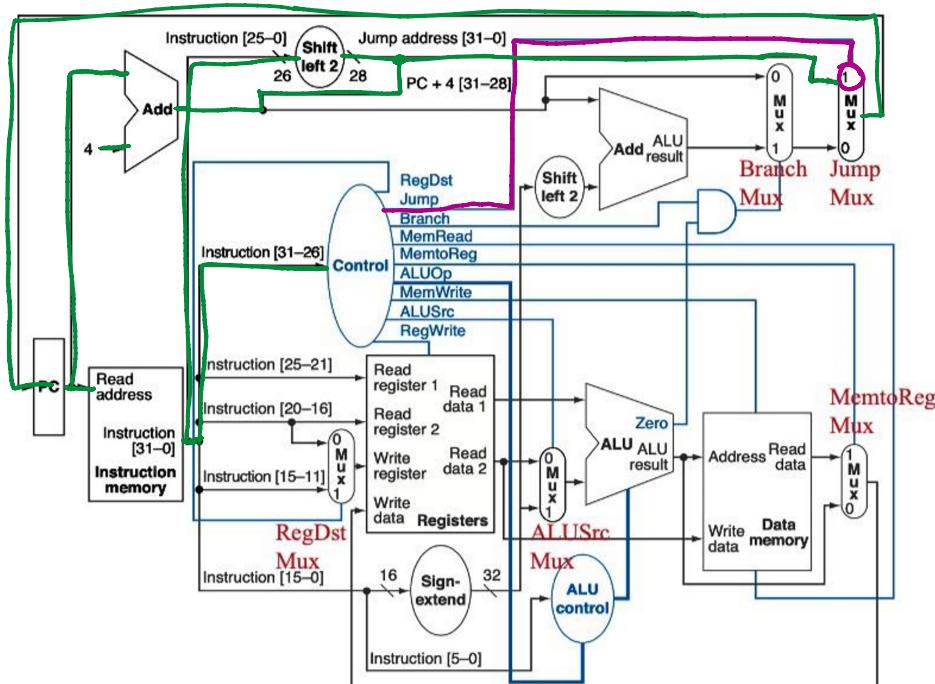
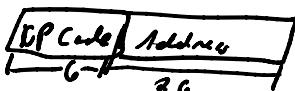


FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. An additional multiplexer (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexer is controlled by the jump control signal. The jump target address is obtained by shifting the lower 26 bits of the jump instruction left 2 bits, effectively adding 00 as the low-order bits, and then concatenating the upper 4 bits of PC + 4 as the high-order bits, thus yielding a 32-bit address.

J Address



2) Explain and trace/highlight the Datapath for the given instructions:

- (i) j Label
- (ii) ori rt, rs, imm

Use fig 4.24 for the above questions and you will need a separate picture for each instruction. (Note: Use different color sketch/pen to easily identify the path of each instruction.) [20 pts, 10 pts each]

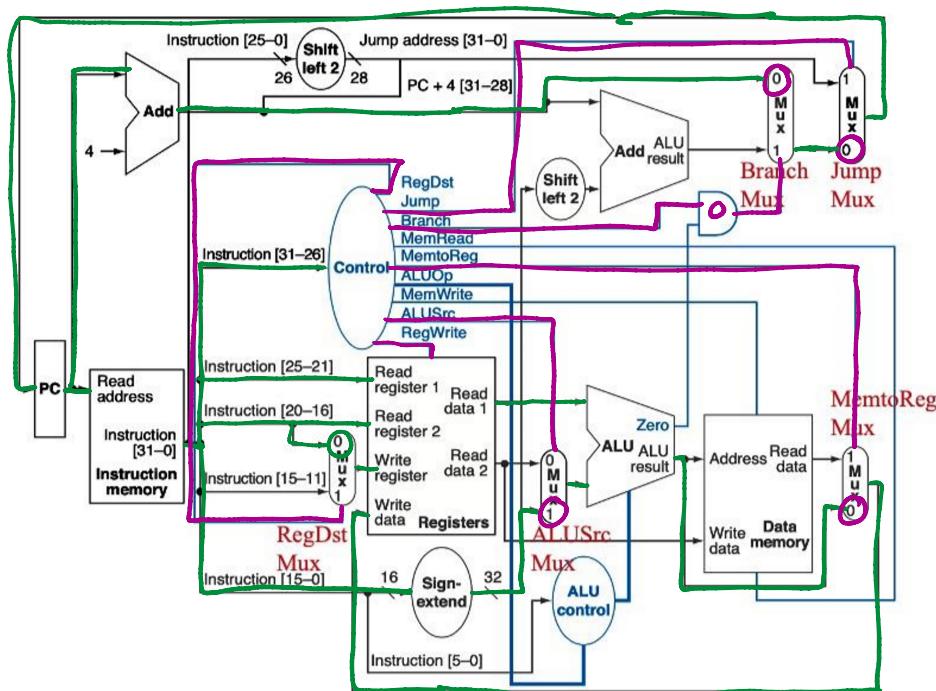
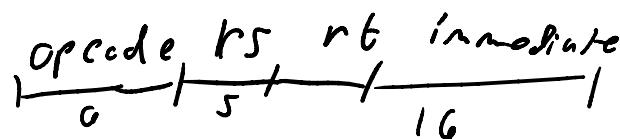


FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump control signal. The jump target address is obtained by shifting the lower 26 bits of the jump instruction left 2 bits, effectively adding 00 as the low-order bits, and then concatenating the upper 4 bits of PC + 4 as the high-order bits, thus yielding a 32-bit address.

Ori rt, rs, immediate



Note: This question(screenshot) is from textbook, however for the assignment we are considering a different diagram and values. I am sharing to let you know that, this is from textbook exercise and I have not changed any wording except numerical values. You no need to answer this sample(4.4, 4.4.1, 4.4.2, 4.4.3)

4.4 Problems in this exercise assume that logic blocks needed to implement a processor's datapath have the following latencies:

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps

4.4.1 [10] <§4.3> If the only thing we need to do in a processor is fetch consecutive instructions ([Figure 4.6](#)), what would the cycle time be?

4.4.2 [10] <§4.3> Consider a datapath similar to the one in [Figure 4.11](#), but for a processor that only has one type of instruction: unconditional PC-relative branch. What would the cycle time be for this datapath?

4.4.3 [10] <§4.3> Repeat 4.4.2, but this time we need to support only conditional PC-relative branches.

- 3) In this problem(you have to calculate critical path), we examine how latencies of individual components of the Datapath affect the clock cycle time of the entire datapath, and how these components are utilized by instructions. For problems in this question, assume the following latencies for logic blocks in the datapath: [15 pts, 5 pts each]

Processor	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
A	300ns	100ns	30ns	150ns	80ns	500ns	20ns	2ns

- a) If the only thing we need to do in a processor is fetch consecutive instructions ([Figure 4.24](#)), what would the cycle time be?
- b) Consider a Datapath similar to the one in [Figure 4.24](#), but for a processor that only has one type of instruction: unconditional branch instruction (b label == beq \$0, \$0, offset). What would the cycle time/critical path be for this Datapath?
- c) Consider a Datapath similar to the one in [Figure 4.24](#), but for a processor that only has one type of instruction: jump instruction (j Loop). What would the cycle time/critical path be for this Datapath?

a) $\boxed{I\text{-Mem} + ADD + MUX}$
 $300 + 100 + 30 = \boxed{430 \text{ ns}}$

b) $I\text{-MEM} + ADD + MUX + R\text{egs} + S\text{ign-Ext} + S\text{hift-Left-2} + MUX + ALU$
 $= 300 + 100 + 30 + 80 + 20 + 2 + 30 + 150$
 $= \boxed{712 \text{ ns}}$

c) ADD + Shift Left - 2 + MAX + IMem
= 300 + 100 + 2 + 70
= 432 ns