Chapter 4

The Processor

Lecture 8

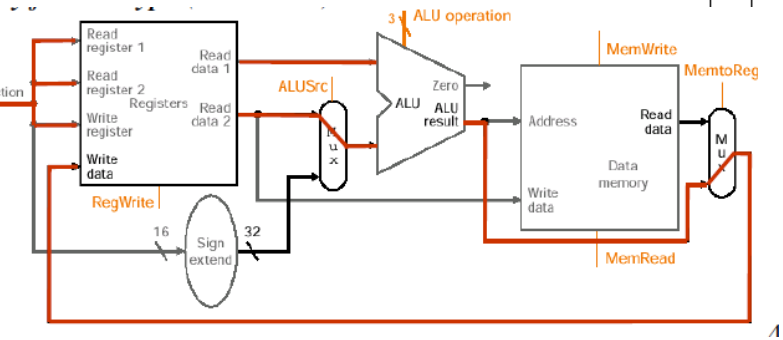
Dr. Karim Emara

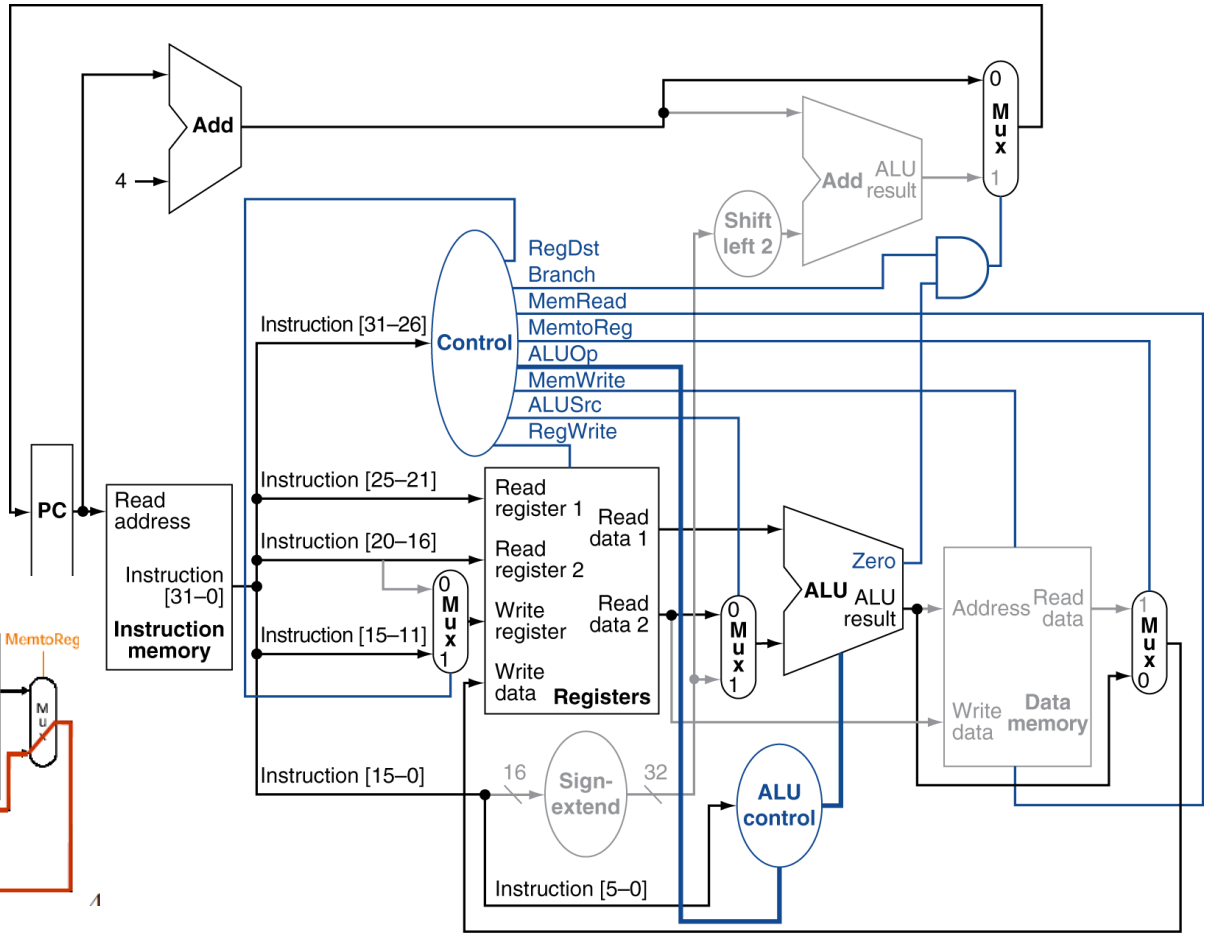
Agenda

- MIPS Revision
- Exercises

R-Type Instruction

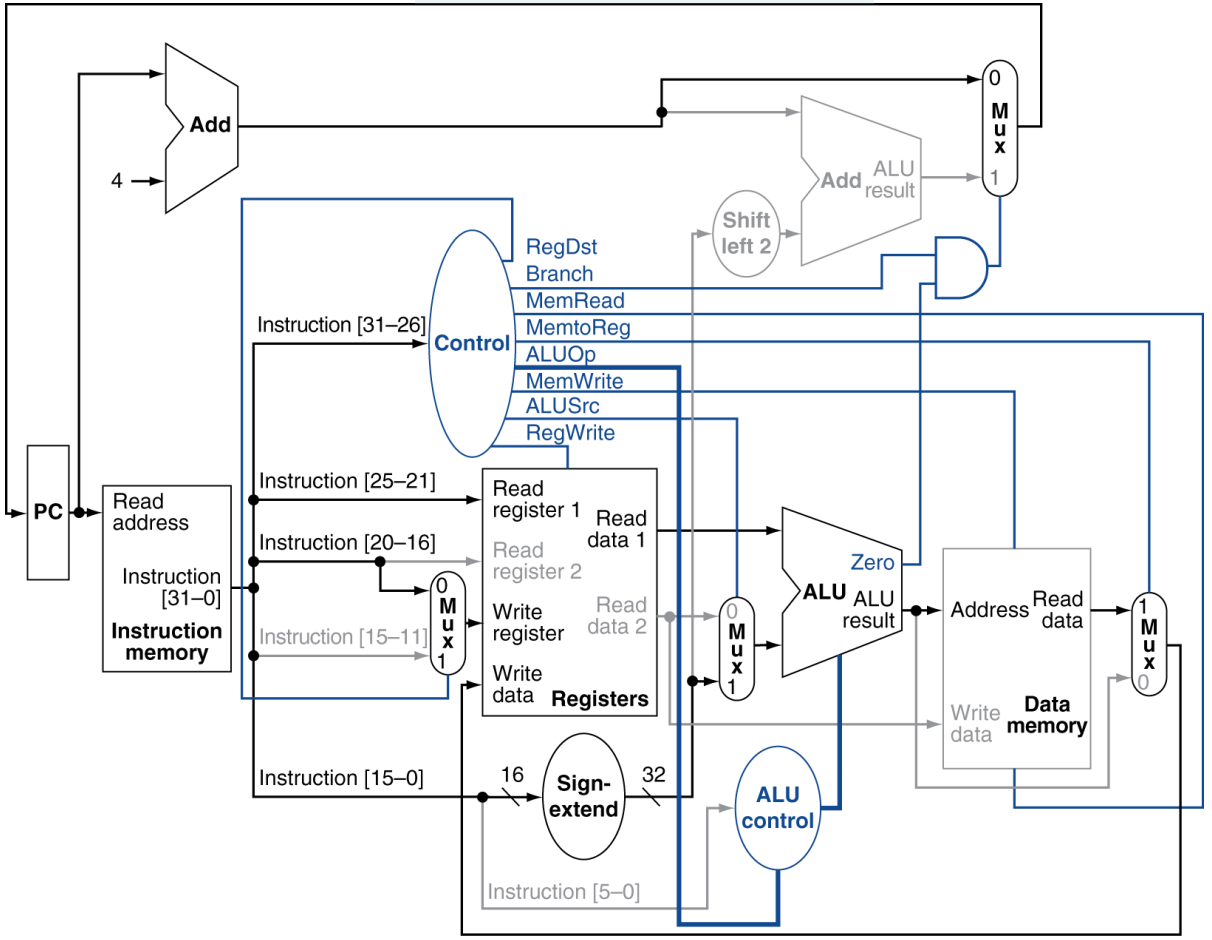
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0



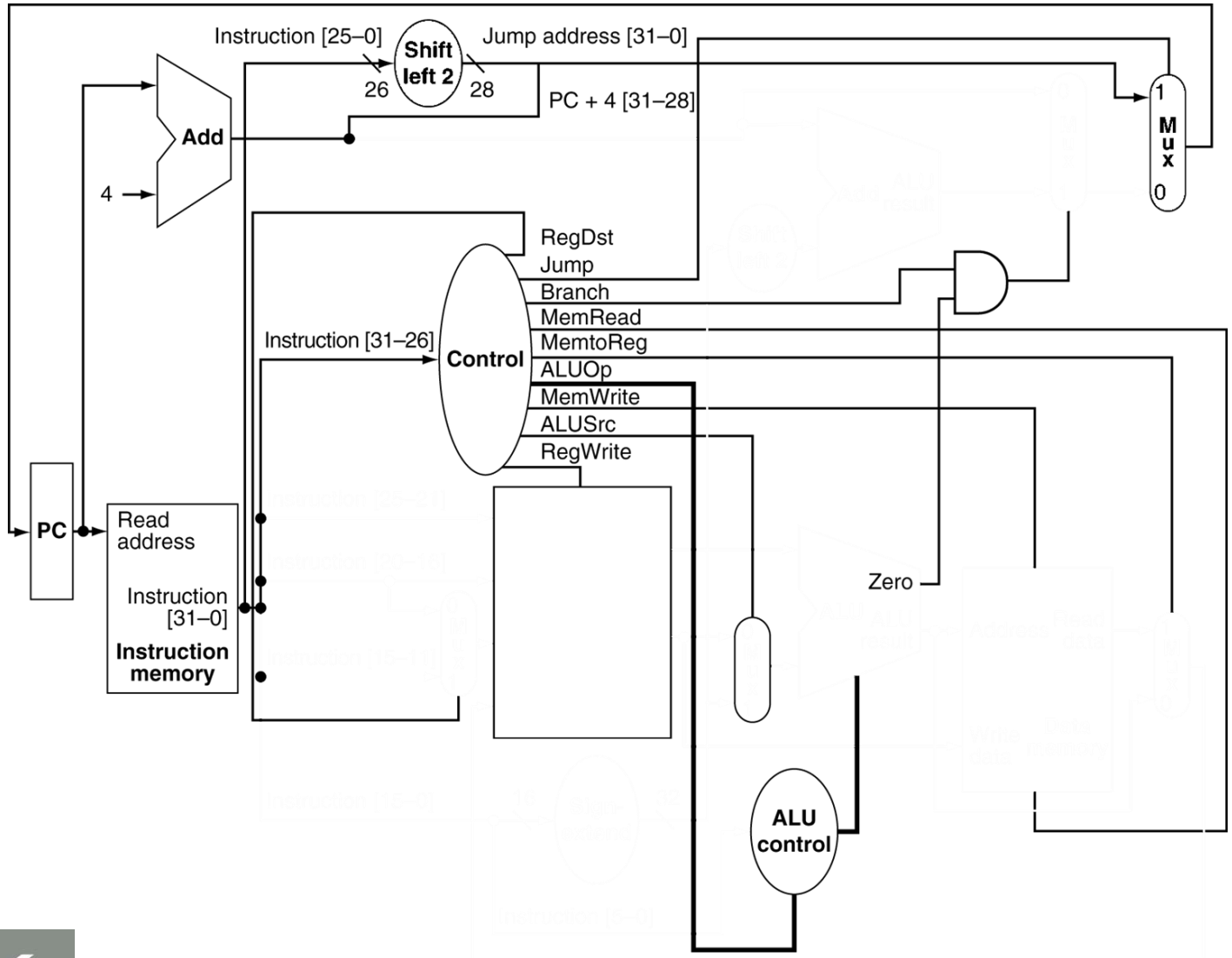


Load Instruction

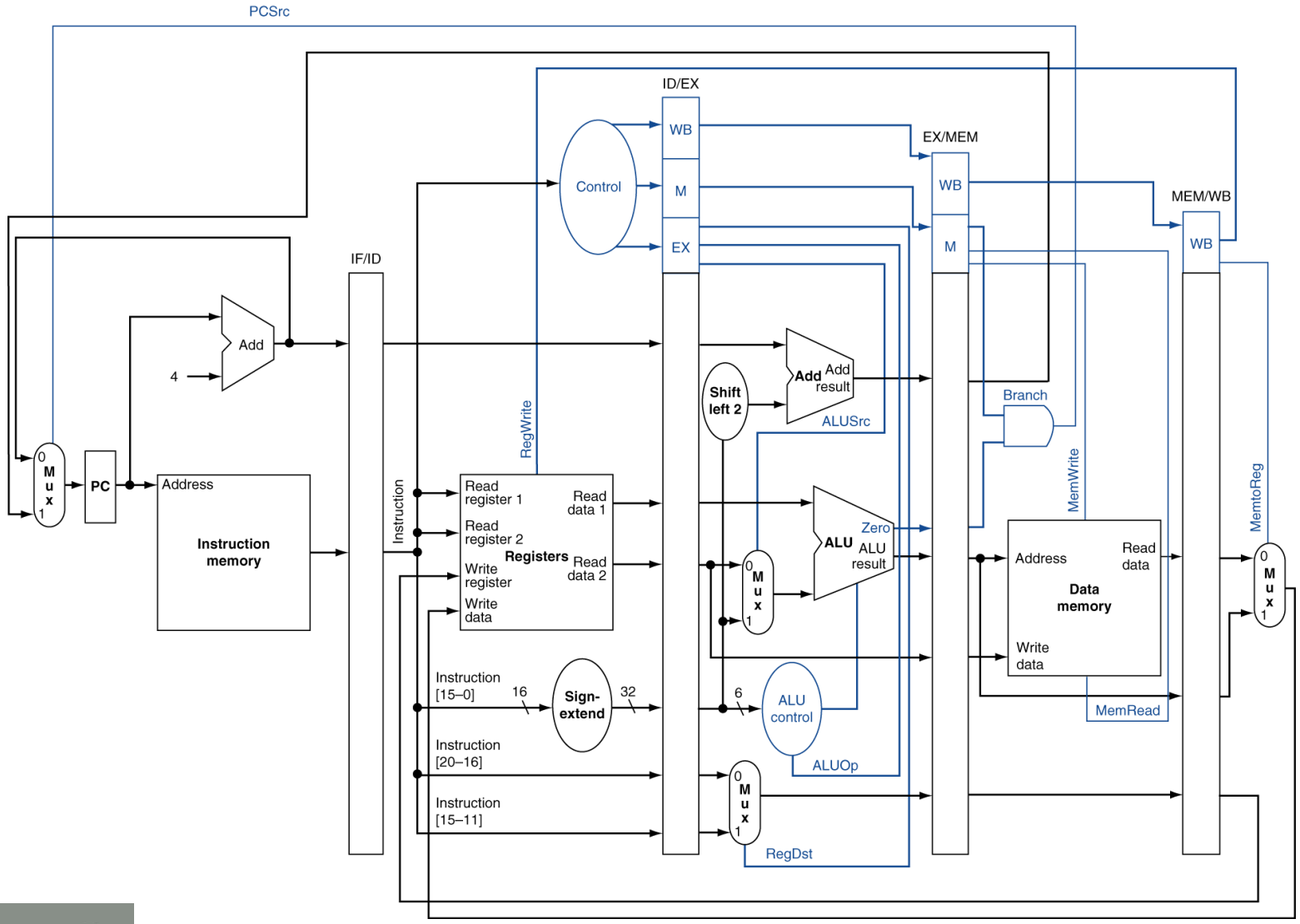
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0



Single Cycle MIPS



Pipelined MIPS



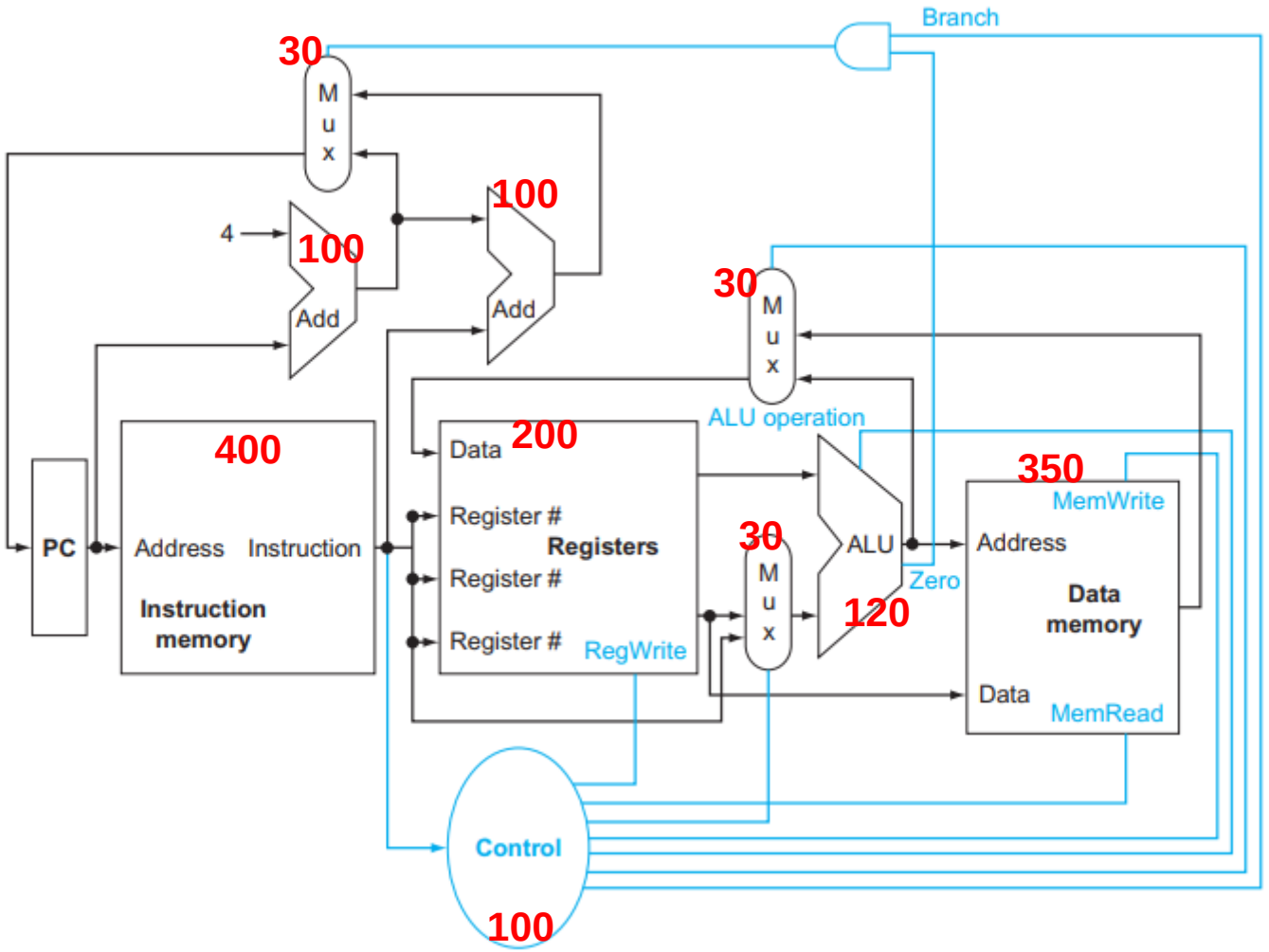
Agenda

- Pipelining Revision
- Exercises

Exercise 4.3

- Assume that we are starting with a datapath from Figure 4.2, 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.

Figure 4.2


Branch
30
M
u
x
100
4
100
Add
Add
30
M
u
x
ALU operation
400
200
Data
PC
Address
Instruction
Instruction
memory
Register #
Registers
Register #
Register #
RegWrite
30
M
u
x
ALU
Zero
120
350
MemWrite
Address
Data
memory
Data
MemRead
Control
100

Exercise 4.3.1

- What is the clock cycle time with and without this improvement?
- Clock cycle time is determined by the critical path (load instruction): I-Mem (read instruction), Regs (read registers and write data), Mux (select ALU input), ALU, Data Memory, and Mux (select value from memory to be written into Registers).

	I-MEM	Regs	Mux	ALU	D-Mem	Mux	
	Add	Control	Mux				
		Add					
W/O	400	200	30	120	350	30	=1130ps
Wt Improve	400	200	30	420	350	30	=1430ps

Exercise 4.3.2

- What is the speedup achieved by adding this improvement?

- Speedup = CPU Time$_{wo\text{-}imp.}$/CPU Time$_{w\text{-}imp.}$
- CPU Time = # Cycles x Cycle Time

- S = 1.0 C x 1130 / 0.95 C x 1430
- = 0.83 (Slowdown)

Exercise 4.4.1

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

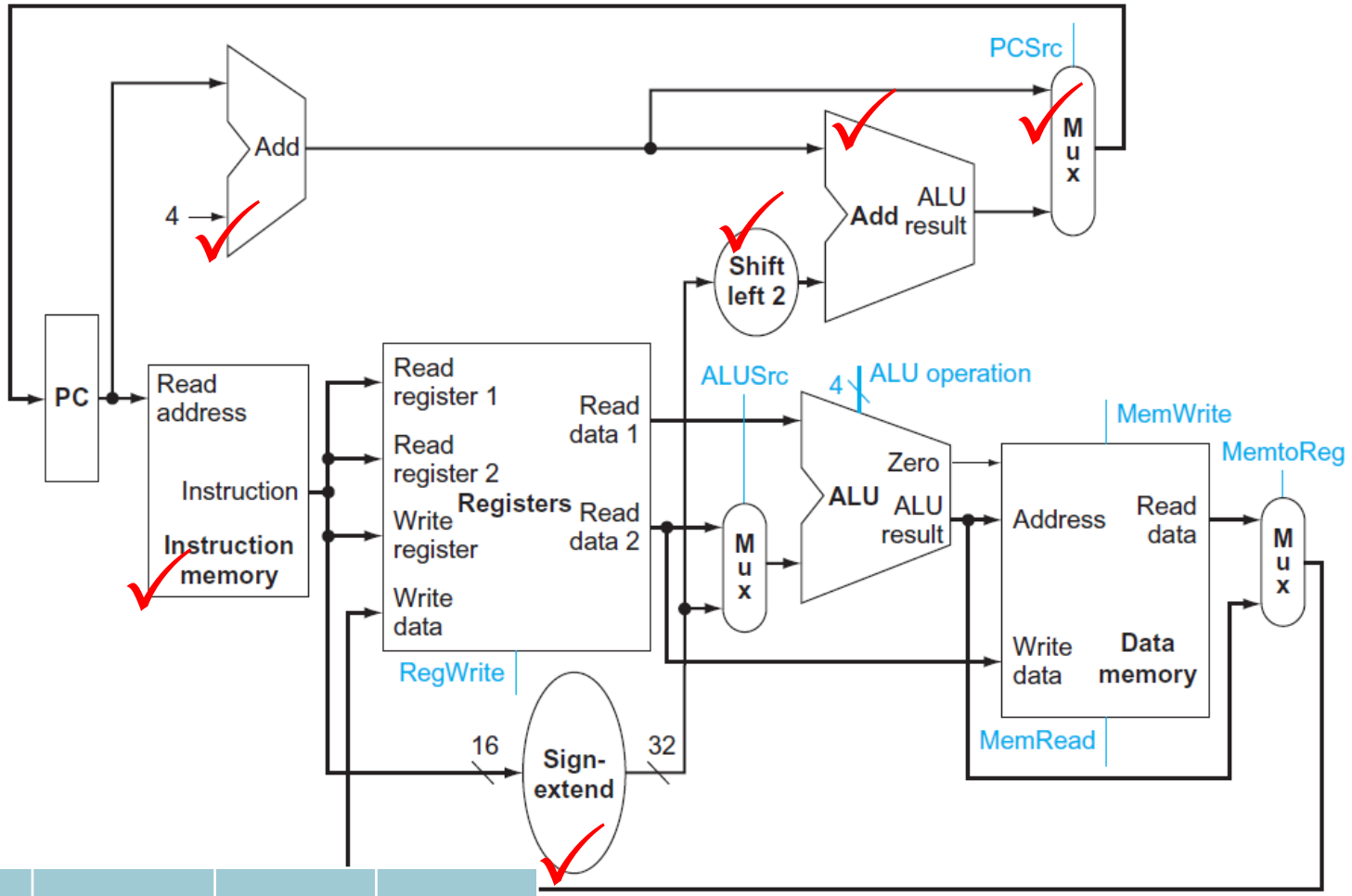
- If the only thing we need to do in a processor is fetch consecutive instructions what would the cycle time be?
- Fetch requires I-MEM access and PC +4 (ADD)
- Both components run in parallel, so the cycle time = max(I-MEM, ADD) = 200 ps

Exercise 4.4.2

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

- Assume processor has only one type of instruction: unconditional PC-relative branch. What the cycle time?

Single-Cycle MIPS



I-MEM	Sign	Shift 2	ADD	MUX
Add				

Exercise 4.4.2

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

- The critical path for this instruction is:
- I-MEM, Sign-Extend, Shift-Left-2, Add, MUX
- Cycle time = 200 + 15 + 10 + 70 + 20 = 315 ps

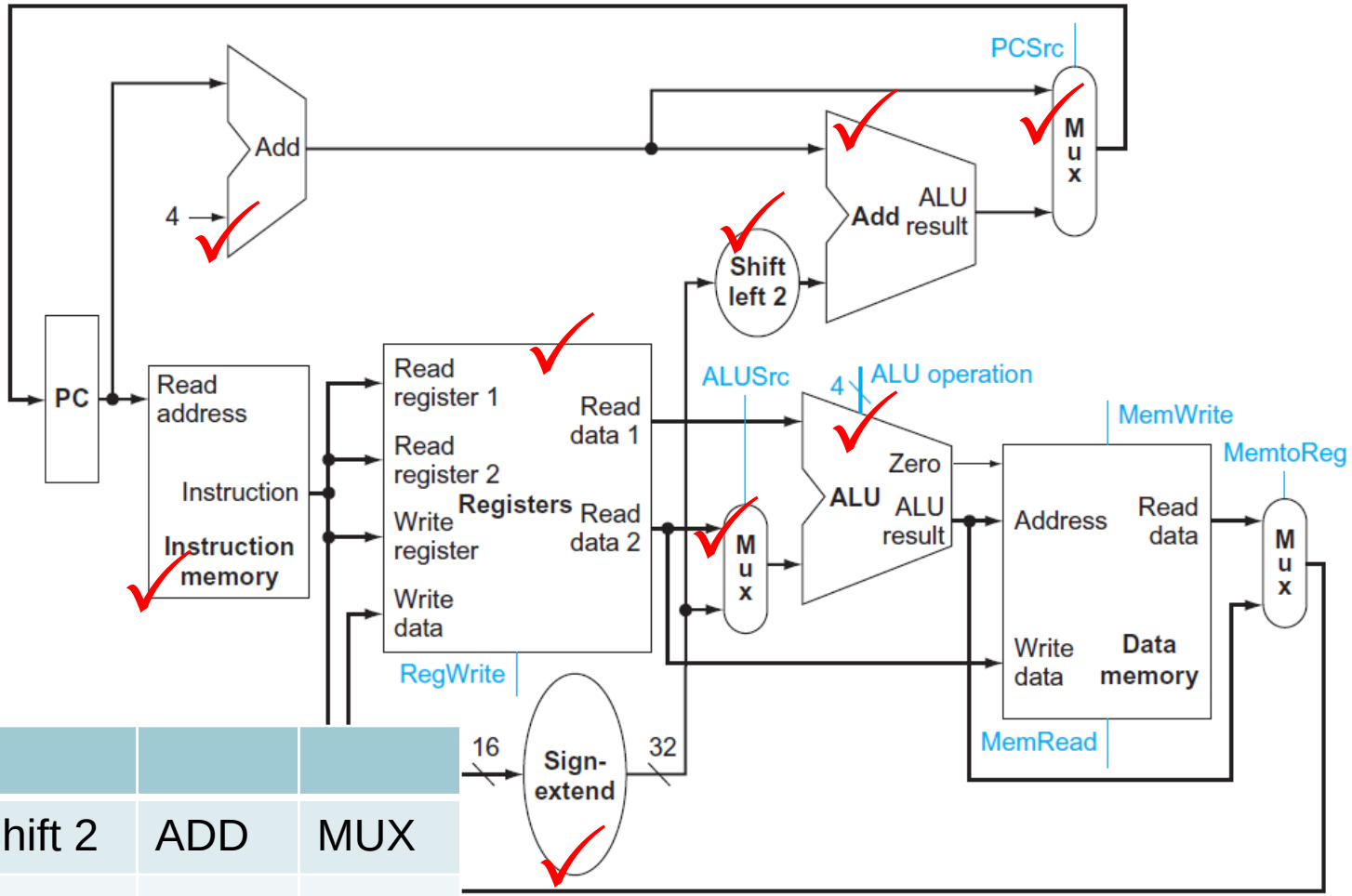
Exercise 4.4.3

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

- Assume processor has only one type of instruction: **conditional** PC-relative branch. What the cycle time?

Single-Cycle MIPS

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



I-MEM	Sign	Shift 2	ADD	MUX
Add	Reg	MUX	ALU	

Exercise 4.4.3

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

- The critical path for this instruction is:
- I-MEM, Regs, MUX, ALU, MUX (ignore the other parallel path of shorter delays)
- Cycle time = 200 + 90 + 20 + 90 + 20 = 420 ps

Exercise 4.4.6

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

- Assuming that we only support **bne** and **add**, discuss how changes in **Shift-left-2** latency affect the cycle time of the processor.

BNE				
I-MEM	Sign	Shift 2	ADD	MUX
Add	Reg	MUX	ALU	

Exercise 4.4.6

- Add instruction has shorter path than Bne. So it does not affect the cycle time
- In Bne, delay of the shift-left component is not included in the critical path of Bne execution
- Shift-left will make difference if its delay is increased so that **its path becomes the critical path**
- i.e.: (SignExtend+ShiftLeft+Add) > (Regs + MUX + ALU)
- ShiftLeft > (90+20+90)-(15+70)

  > 115ps (increased by 105ps more)

Exercise 4.7

- instruction word: 101011 00011 00010 00000 00000010100.
- Assume that data memory is all zeros and that the processor's registers have the following value

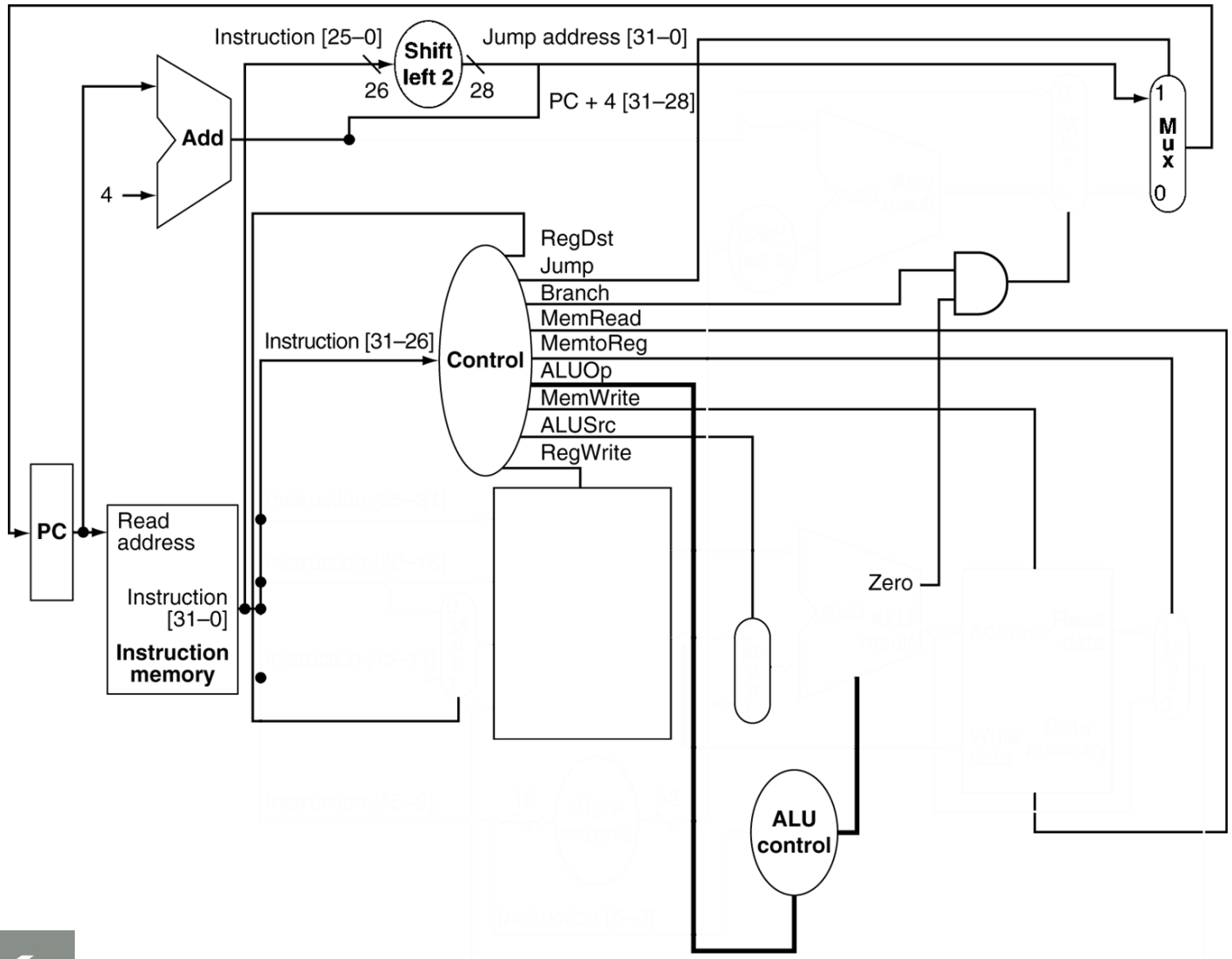
r0	r1	r2	r3	r4	r5	r6	r8	r12	r31
0	–1	2	–3	–4	10	6	8	2	–16

- 101011 ➔ sw, rs = 3, rt = 2, Imm = 20

4.7.1

- What are the outputs of the sign-extend and the jump “Shift left 2” unit for this instruction word?
- Sign-Extend: 0…000 00010100
- Jump Shift-left: 00011 00010 00000 00000010100 **00**

Single Cycle MIPS



4.7.2, 3

- What are the values of the ALU control unit's inputs for this instruction?
- ALU control = 00 → add
- What is the new PC address after this instruction is executed? Highlight the path through which this value is determined.
- PC + 4
- Path: Add (PC4) to branch Mux to jump Mux to PC

4.9

- Assume the following sequence of instructions:
- or r1,r2,r3
- or r2,r1,r4
- or r1,r1,r2
- Also, assume the following cycle times for each of the options related to forwarding

Without Forwarding
250ps

4.9.1

- Indicate dependences and their type.
- I2 and I3 depend on I1
- I3 depends on I2

- All dependencies cause data hazards

4.9.2

- Assume there is **no forwarding** in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

CC	1	2	3	4	5	6	7
I1	IF	ID R2, R3	EX	MEM	WB R1		
I2		IF	ID R1, R4	EX **2 CC**	MEM	WB R2	
I3			IF	ID R1, R2	EX **2 CC**	MEM	WB R1

4.9.2

- OR R1,R2,R3
- NOP
- NOP
- OR R2,R1,R4
- NOP
- NOP
- OR R1,R1,R2

Reference

- Book sections: 4.1 – 4.6 (except parts not mentioned in the lectures)
- Exercises: 4.1 – 4.10 except 4.6