Exercise 4.1

4.1.1.

For a.

1: RegDst, RegWrite

0: ALUSrc, MemtoReg, MemRead, MemWrite, Branch

For b.

1: ALUSrc, MemtoReg, RegWrite, MemRead

0: RegDst, MemWrite, Branch

4.1.2.

For a. Registers, ALU

For b. Registers, ALU, Data memory

4.1.3.

For a.

Output produced: Registers, ALU

No output produced: Data memory

For b.

Output produced: Registers, ALU, Data memory

No output produced: Branch

4.1.4

For a.

Total latency = 400PS(I-Mem) + 200PS(Regs-read) + 120PS(ALU) + 200PS(Regs-write) = 920PS

For b.

Total latency = 500 + 220 + 180 + 220 = 1120PS

4.1.5.

For a.

Total latency = 400PS(I-Mem) + 200PS(Regs) + 120PS(ALU) + 350PS(D-Mem) + 200PS(Regs) = 1270 PS

For b.

Total latency = 500 + 220 + 180 + 1000 + 220 = 2120PS

4.1.6.

For a.

Total latency = 400PS(I-Mem) + 200PS(Regs) + 120PS(ALU) = 720PS

For b.

Total latency = 500 + 220 + 180 = 900PS

Exercise 4.7

4.7.1

For a.

Total Clock Cycle Time = 400PS (I-Mem) + 200PS (Regs) + 120PS (ALU) + 200PS (Regs) = 920PS

For b.

Total Clock Cycle Time = 300 + 220 + 180 + 220 = 1120PS

4.7.2.

For a

Total Clock Cycle Time = 400PS (I-Mem) + 200PS (Regs) + 120PS (ALU) + 350PS (D-Mem) + 200PS (Regs) = 1270PS

For b.

Total Clock Cycle Time = 500 + 220 + 180 + 1000 + 220 = 2120PS

4.7.3.

For this, we consider the slowest of all instruction paths (it is always `lw`)

For a

Total Clock Cycle Time = 1270PS (the same as 4.7.2)

For b.

Total Clock Cycle Time = 2120PS (the same as 4.7.2)

4.7.4. The data memory is used in `lw` and `sw`

For a. 20% + 10% = 30%

For b. 35% + 15% = 50%

4.7.5. `addi`, `lw`, `sw`, `beq`

For a. 15% + 20% + 10% + 20% = 65%

For b. 5% + 35% + 15% + 15% = 70%

Extend the immediate values from their typical 16-bit width to the 32-bit width to ensure the immediate values are correctly interpreted, whether as offsets for memory operations or as operants in arithmetic (I-type)

4.7.6.

For both a and b:

The component is `D-Mem`

$$\text{Speed-up} = \frac{\text{Excution Time}_{old}}{\text{Excution Time}_{new}} - 1$$

$$= \frac{1}{1-0.1} - 1 \approx 11\%$$

Exercise 4.9

4.9.1

For a. $(8CC10028)_H$

For b. $(1422 \langle label\_offset \rangle)_H$

4.9.2

For a.

Read Register 1: The register supplied is `$6`, which is register number `6`

Read Register 2: No second register is read.

For b.

Read Register 1: This register to be read is `$1`, thus the register number supplied to its input is `1`.

Read Register 2: The register is `$2` and the number is `2`

4.9.3.

For a. The target register is `$1`, which is register number `1`.

For b: No Write register is needed.

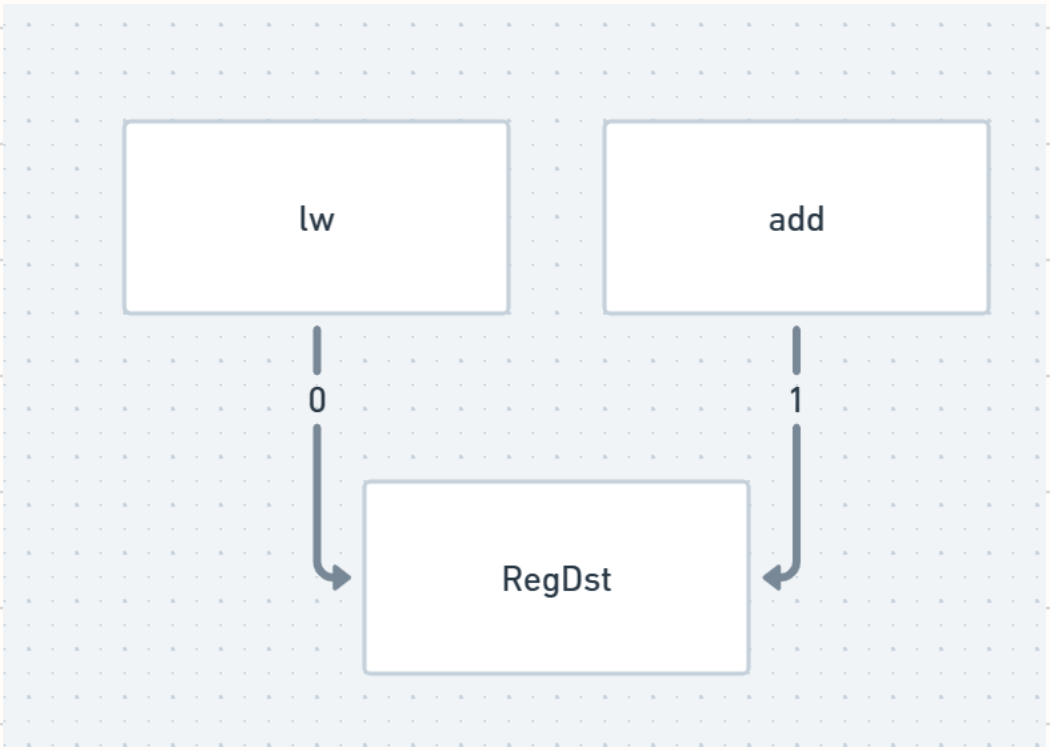
4.9.4.

For a. RegDst: `0`; MemRead: `1`

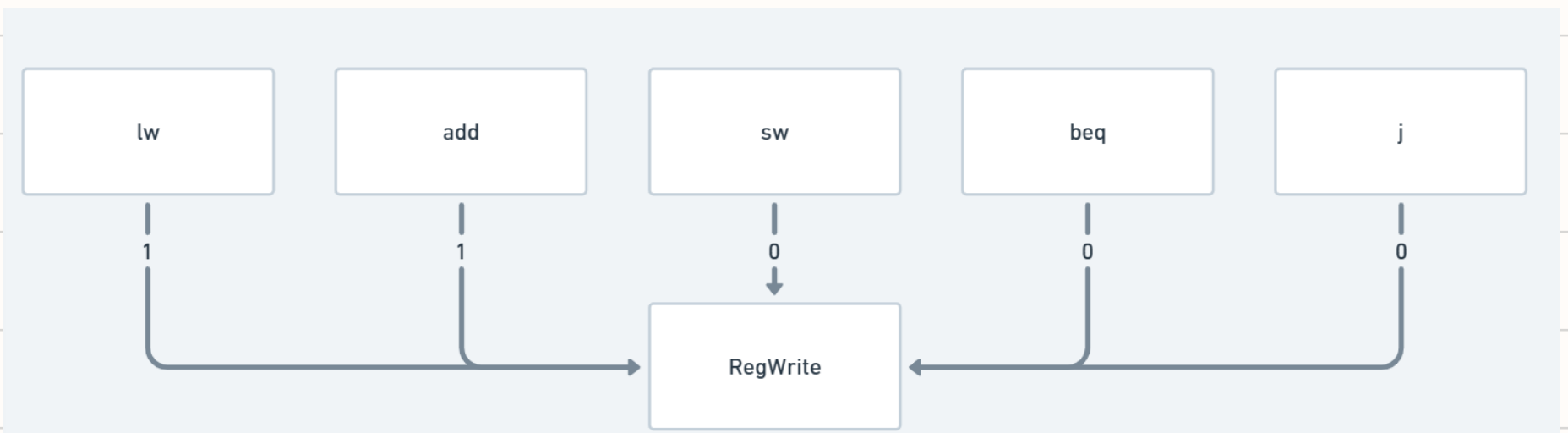
For b. RegWrite: `0`; MemRead: `0`

4.9.5.

For a.

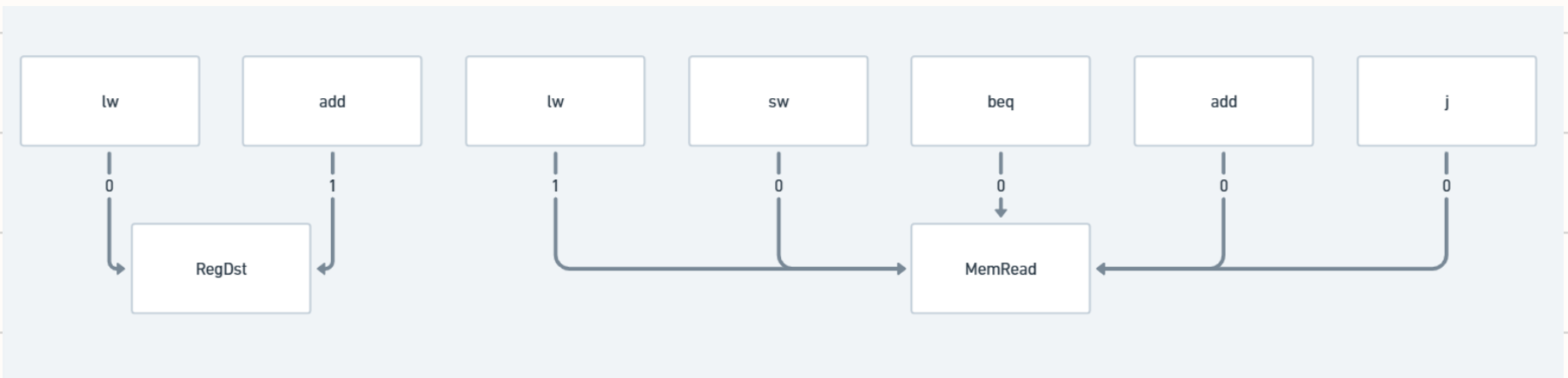


For b.

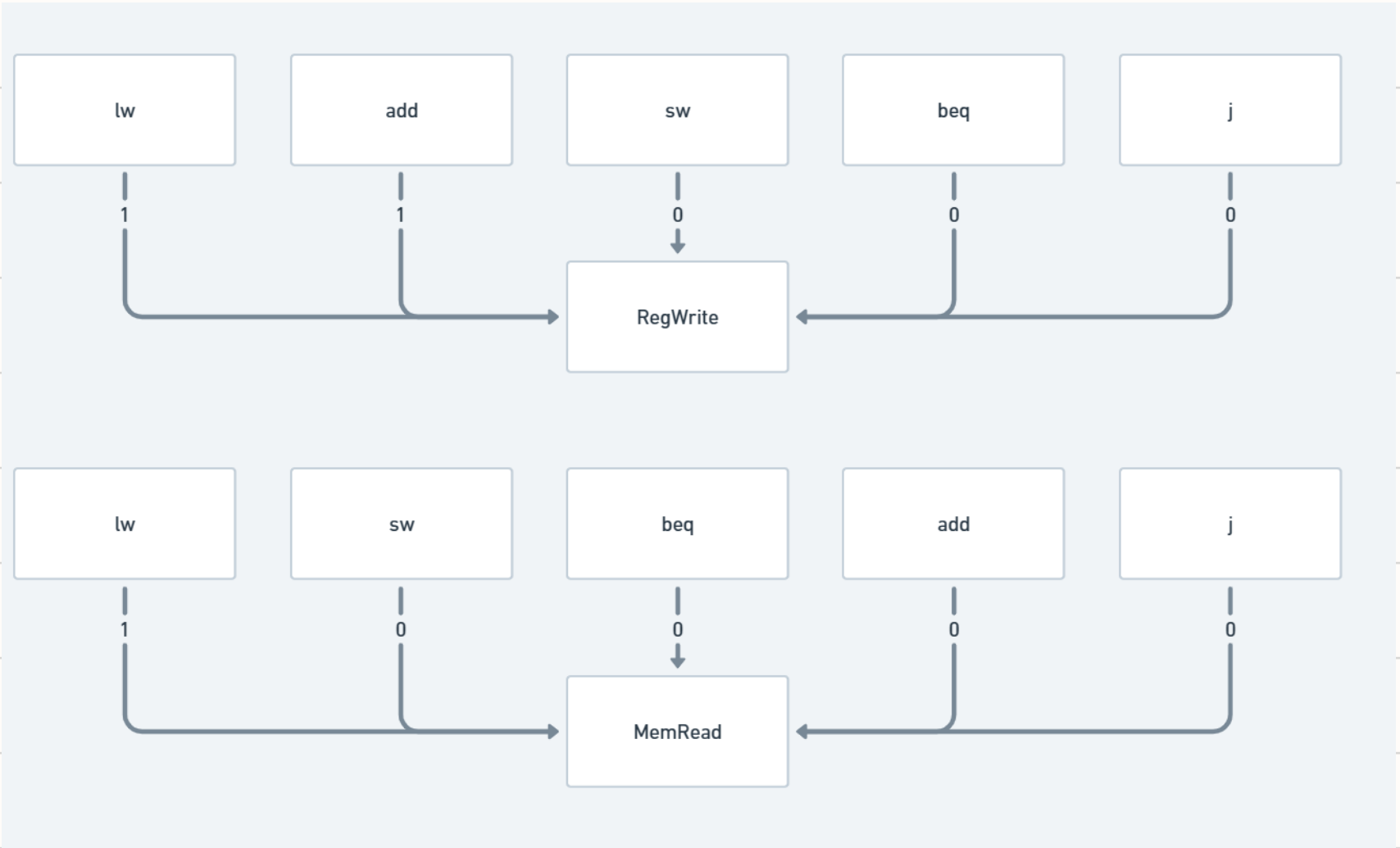


4.9.6

For a.



For b.



Exercise 4.11.

4.11.1.

For a. lw $3 0x1a($2)

Sign-extend: $(00000010)_H$ (32-bit binary indeed)

jump "shift left 2": Not used

for b: beq $1, $3, 12

Sign-extend: $(00000000c)_H$ (32-bit binary indeed)

jump "shift left 2": Not used.

4.11.2.

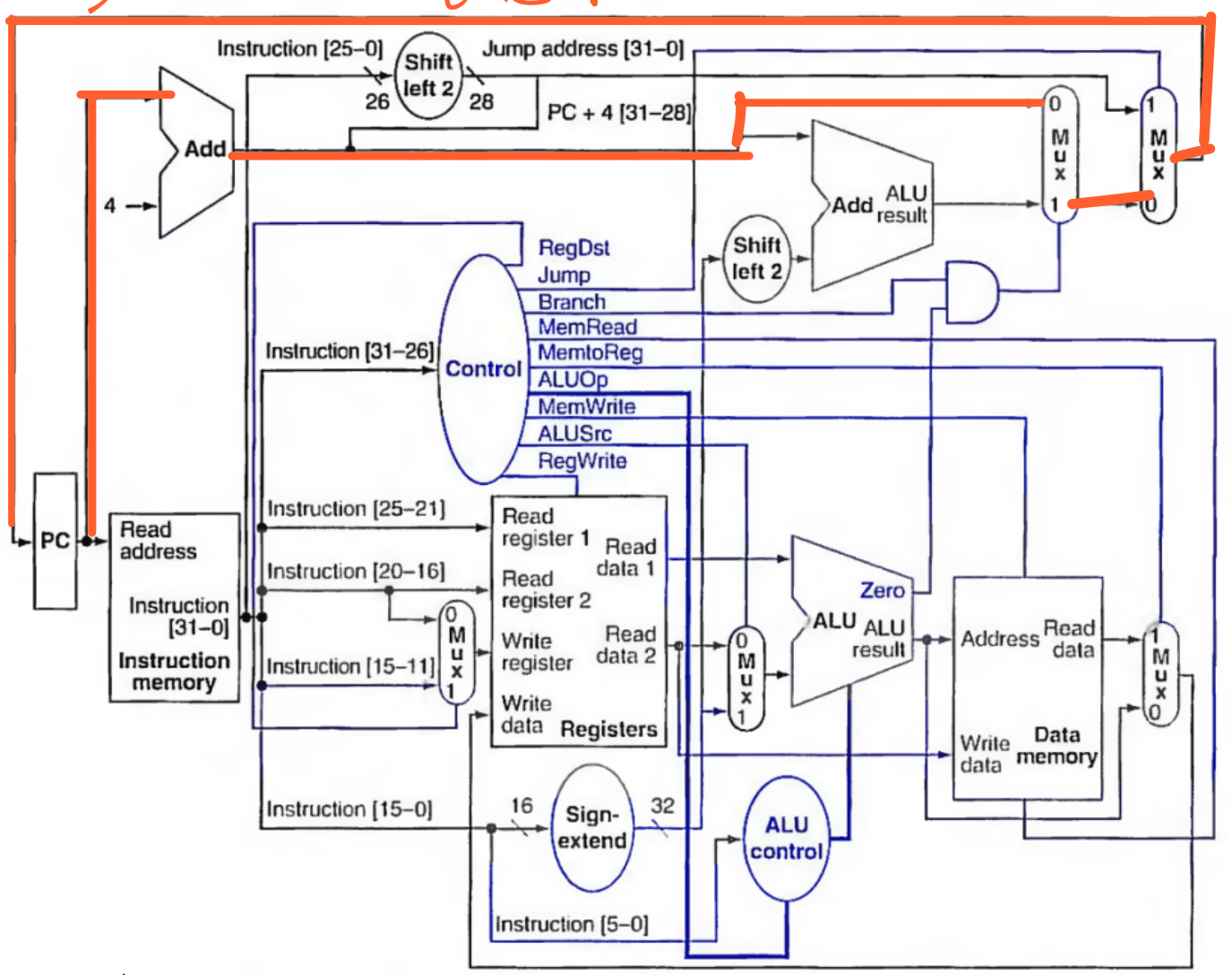
For a. 00 (for load/store operations)

For b. 01 (for branch operation)

4.11.3.

For a. New PC address: PC+4

Below is the Path

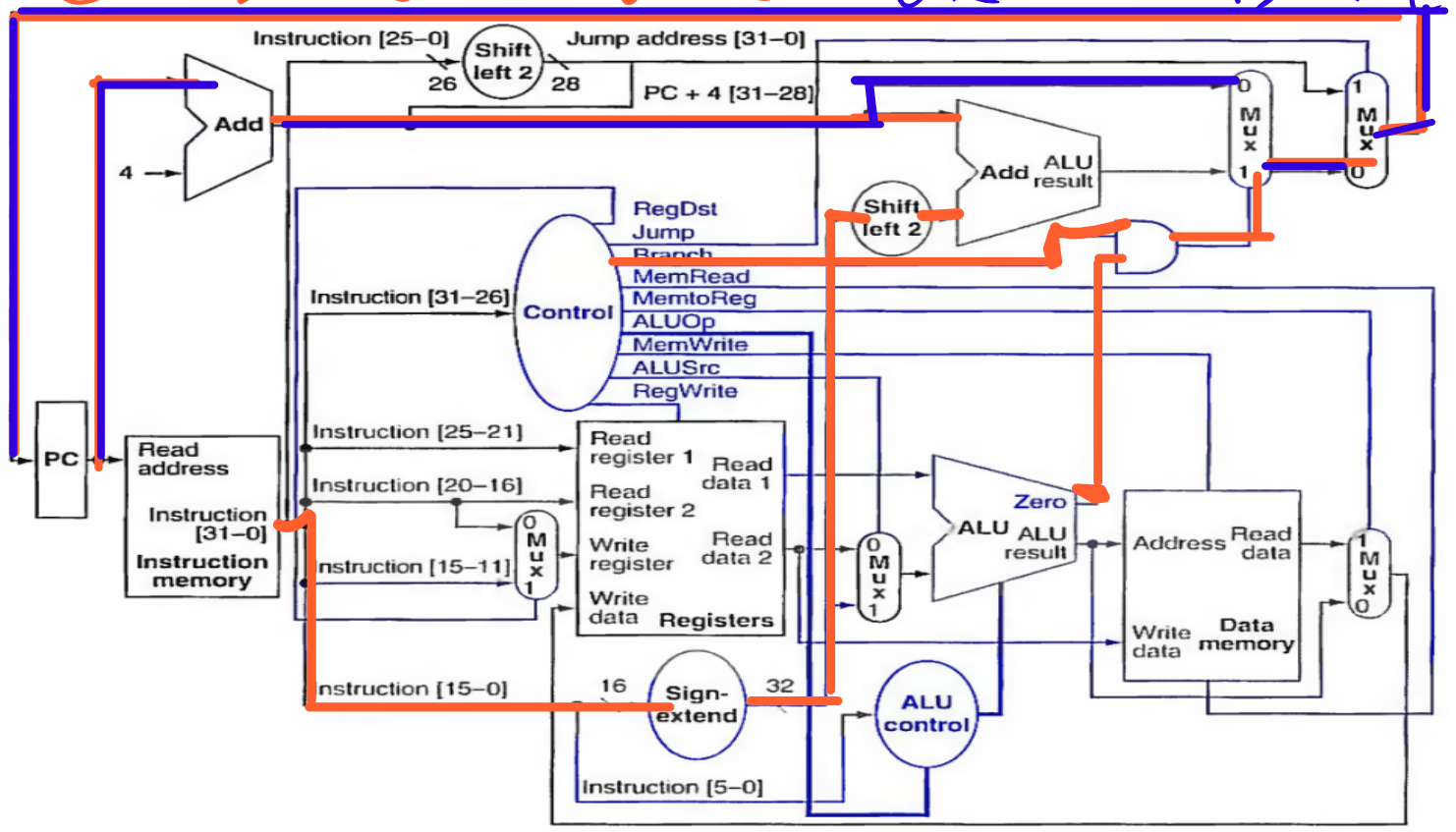


For b.

① if the value in register $1 and $3 are equal, new PC address is: (PC+4 + immediate≪2).

② if they are not equal, new PC address is: (PC+4).

The red is branch taken the blue is not taken

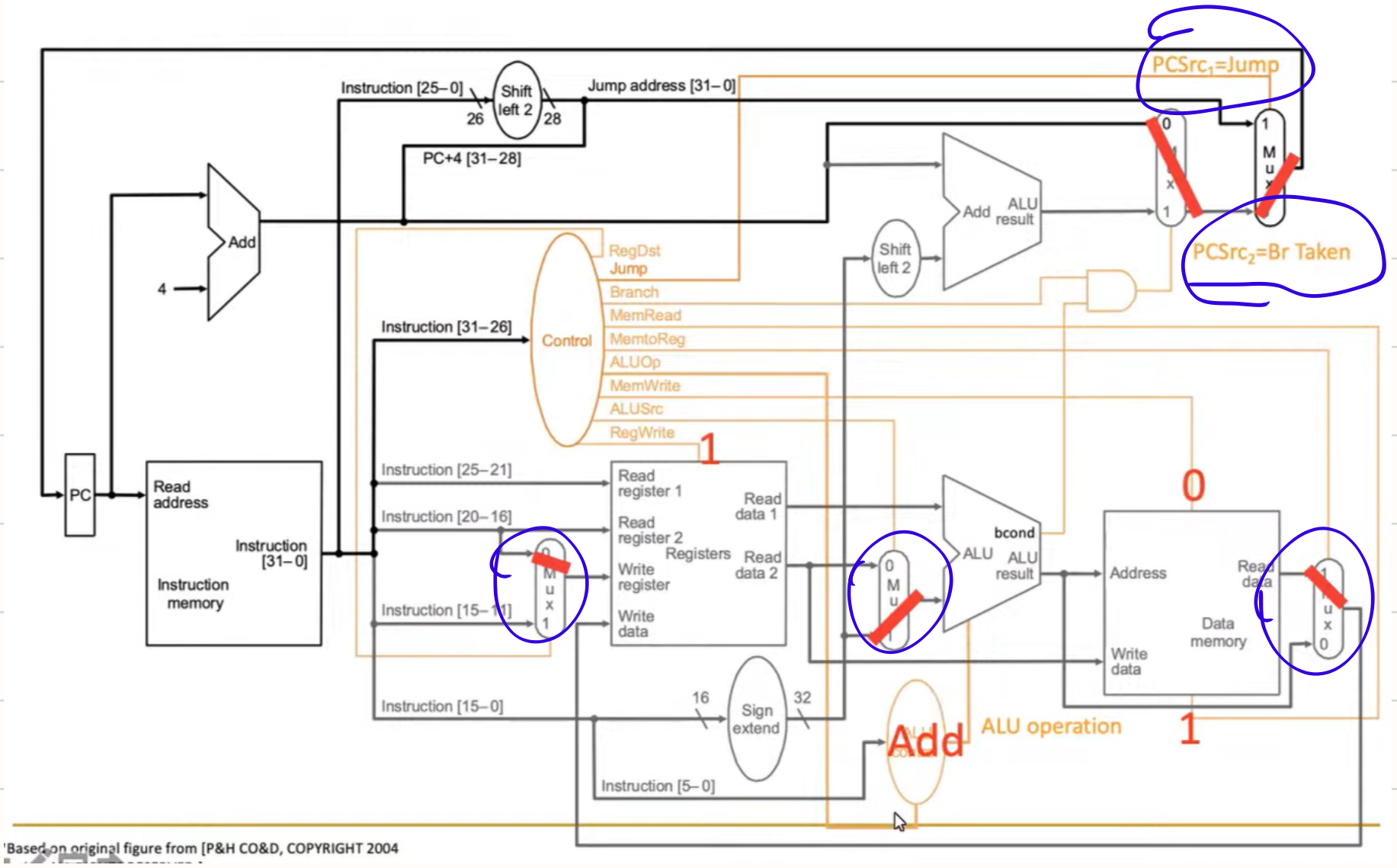


4.11.4

For a.

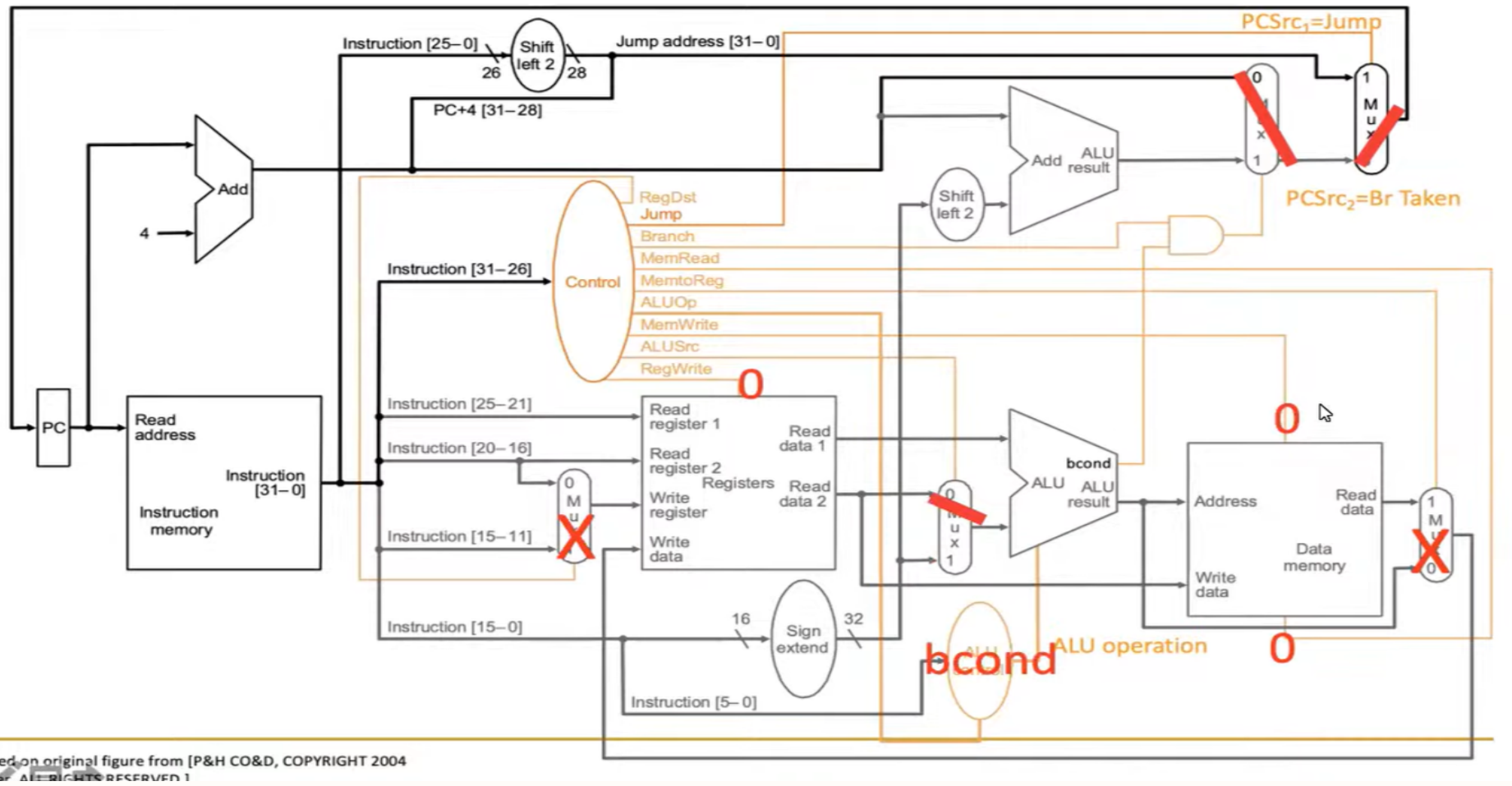
	Data output	Register value
RegDst Mux:	0	3
ALUSrc Mux:	1	16
MemtoReg Mux:	1	0
$PCSrc_1$ Mux:	0	PC+4
$PCSrc_2$ Mux:	0	PC+4
MemWrite Mux:	0	Disable
RegWrite Mux:	1	Enable
MemRead Mux:	1	Data from memory

LW



For b.

Branch (Not Taken)



Because in our scenario, the branch is not taken

RegDst Mux: Not used

MemtoReg Mux: Not Used

ALUSrc MuX: 0 values in $3

MemWrite/MemRead/RegWrite: 0 Disable

PCSrc2 Mux: 0 PC+4

PCSrc1 Mux: 0 PC+4

4.11.5.

For a.

ALU Inputs: Input 1: Value in $2 (2)

Input 2: Immediate Value (16)

First Add Unit: Input 1: PC

Input 2: 4

For b.

ALU Inputs: Input 1: Value in $1 (-16)

Input 2: Value in $3 (-3)

First Add Unit: Input 1: PC

Input 2: 4

Second Add Unit: Input 1: PC +4

Input 2: Immediate value shifted left by 2 (48)

4.11.6.

For a.

Read Register 1: $2 (2)

Read Register 2: $3 (3)

Write Register: $3 (3)

Write Data: Data from memory (0)

Read Data 1: Value in $2 (2)

Read Data 2: Value in $3 (3)

For b.

Read Register 1: $1 (-16)

Read Register 2: $3 (-3)

Write Register: Not used

Write Data : Not used

Read Data 1: Value in $1 (-16)

Read Data 2: Value in $3 (-3)

Additional Question:

Final Table

Iteration	Instruction	IS	EX	WB	Comment
1	L.D F0, 0(R1)	1	2	3	First issue
1	MUL.D F0, F0, F2	2	4-8	9	Wait for F0
1	L.D F4, 0(R2)	3	4	5	
1	ADD.D F0, F0, F4	4	10-11	12	Wait for MUL.D, L.D
1	S.D 0(R2), F0	5	13	14	Wait for ADD.D
1	DSUBUI R1, R1, #8	6	7	8	
1	DSUBUI R2, R2, #8	7	8	9	
1	BNEZ R1, loop	8	9	10	Wait for DSUBUI
2	L.D F0, 0(R1)	9	10	11	
2	MUL.D F0, F0, F2	10	12-16	17	
2	L.D F4, 0(R2)	11	12	13	
2	ADD.D F0, F0, F4	12	18-19	20	Wait for MUL.D, L.D
2	S.D 0(R2), F0	13	21	22	Wait for ADD.D
2	DSUBUI R1, R1, #8	14	15	16	
2	DSUBUI R2, R2, #8	15	16	17	
2	BNEZ R1, loop	16	17	18	Wait for DSUBUI

Excution Time per Iteration

Iteration 1: Max(WB) = 14

Iteration 2: Max(WB) = 22