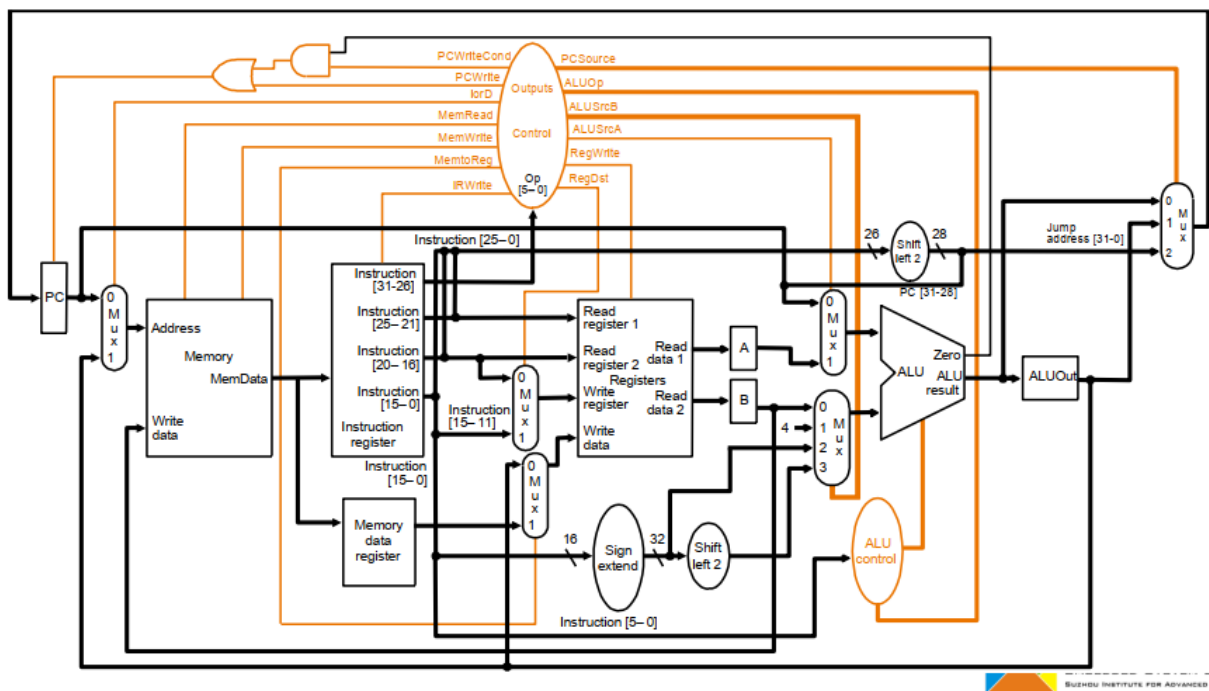
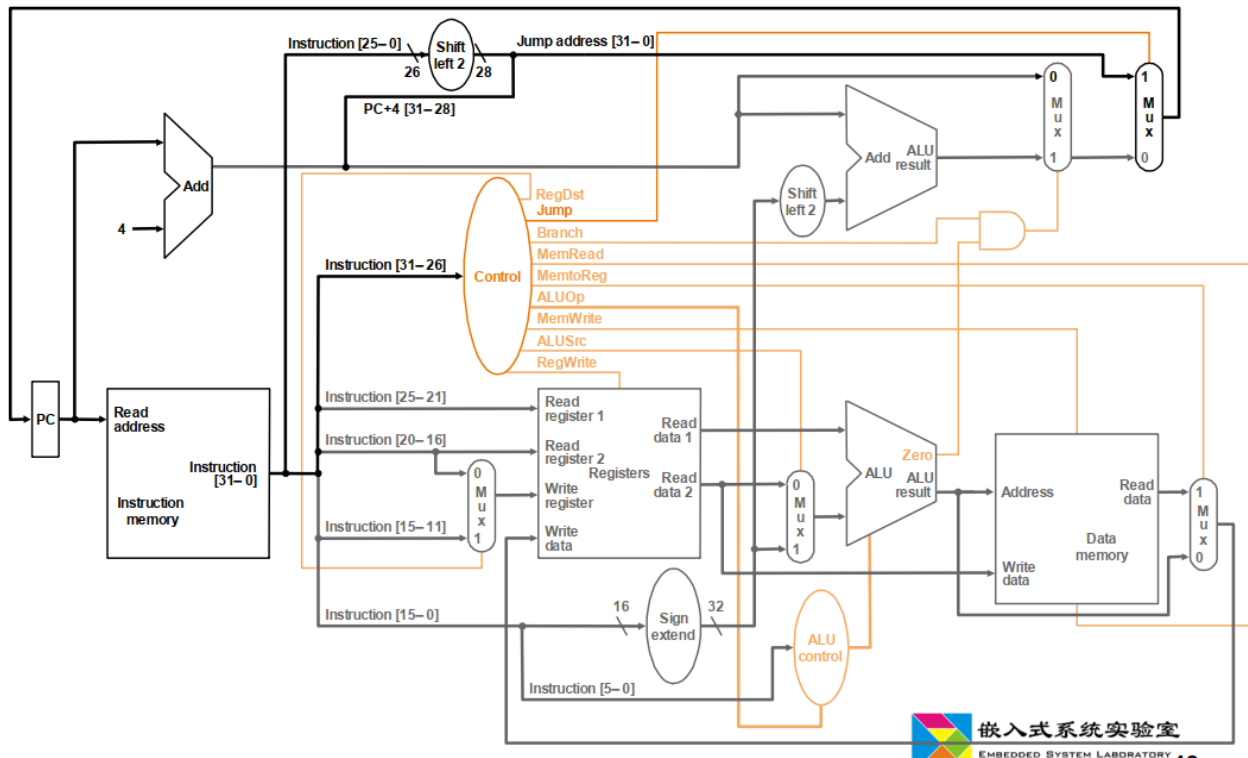


由于课本上找不到所谓图 4-2，此次作业参照老师 PPT 上的图。



Week 6.

4.1.

4.1.1

1 表示使能或高电平, 0 表示失能或低电平

	RegPst	RegWrite	Branch	MemRead	MemWrite	MemtoReg	ALUOP	ALUSrc
a	1	1	0	0	0	0	10	选 Reg, 0
b	0	1	0	1	0	1	00	选 Imm, 1

4.1.2.

a. PC, 指令存储器, 寄存器堆, ALU, Add, Mux, 控制

b. PC, 指令存储器, 寄存器堆, ALU, Add, 数据存储器, Mux, 控制

4.1.3. 输出而不用到:

a. 右边的 Add, shift left 2 (for jump), shift

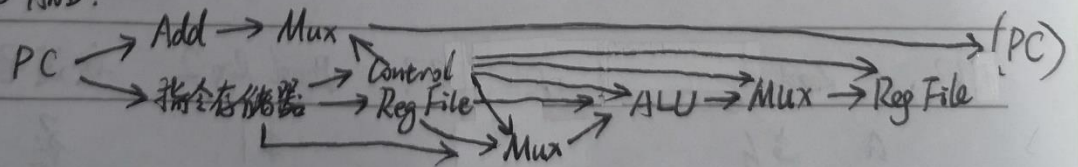
b. Registers 的 Read Data 2, shift left 2 (for jump)

不产生输出:

a. Data Memory, ALU's Zero

b. 无

4.1.4. 对于 AND:

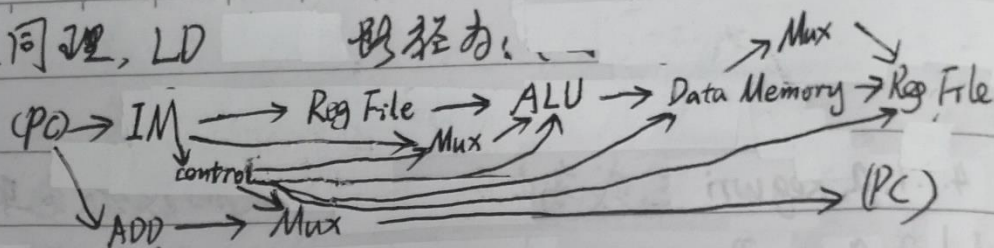


显然 a, b 关键路径均为 (PC) → 指令存储器 → 寄存器堆 → Mux → ALU

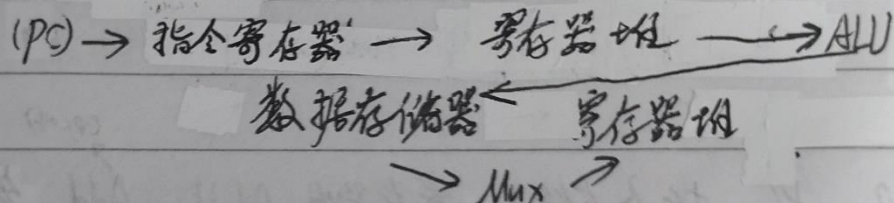
Mux ← 寄存器堆

4.1.5. 同理, LD

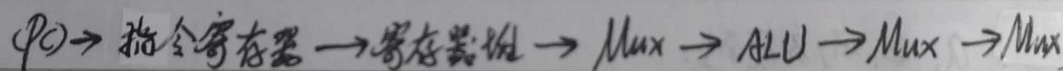
路径为:



经比较, a, b. 关键路径均为:



4.1.6. a, b. 均为:



4.9.1 a. 值为 100011 00110 00001 0000 0000 0010 1000

即 0x8CC10028

b. 值为 000101 00001 00010 1111 1111 1111 1111

即 0x1422FFFF

4.9.2. Read Reg 1 被读吗? Read Reg 2 被读否?

a. \$6

是

\$1

否

b. \$1

是

\$2

是



4.9.3.

Write Reg

被写否?

a. \$1

是

b. \$2

否

4.9.4.

控制信号 1

控制信号 2

a. 0

1

b. 0

0

4.9.5. 和 4.9.6.

[31:26]

输入指令  
32

101 3 0x23 (lw)  
000 3 0x26 (sw)  
000 3 0x04 (beq)  
110 3 0x00 (add)  
000 3 0x02 (j)  
⋮  
MUX

[2] → Reg Write

(均为大端数据)

[1] → Reg Dst

[0] → MemRead

(注: 4.9.5 只需去掉 MemRead 部分  
此图为 4.9.6 答案)

2.	R-type	I-type			J-type
		lw	sw	branch	
IF	PC, IR, Memory, ALU				
ID	Control, Registers, ALU, A,B, ALUOut				
EX	ALU, ALUOut				shifter, PC
MEM	Registers	MDR Memory	Memory		
WB		Registers			