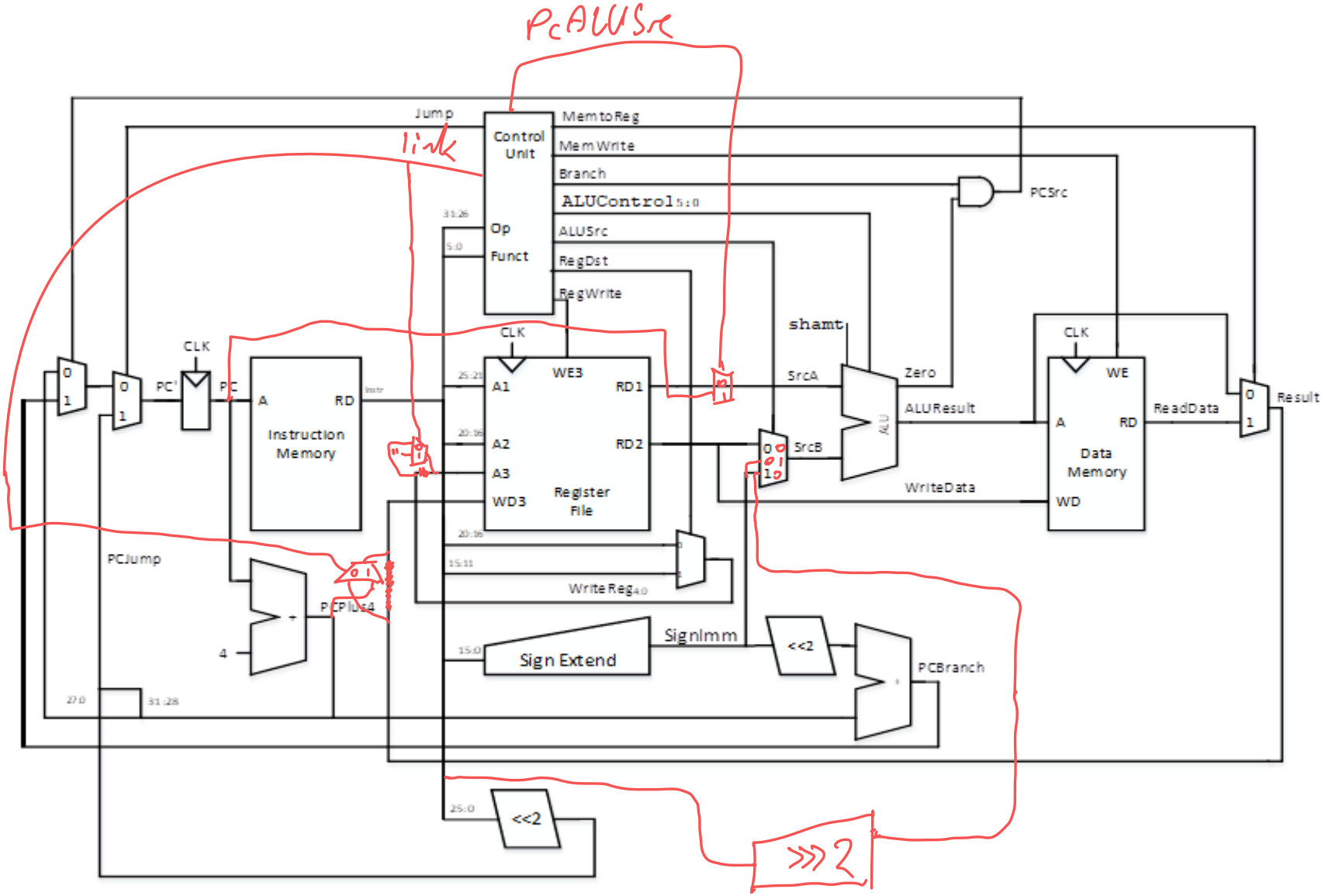
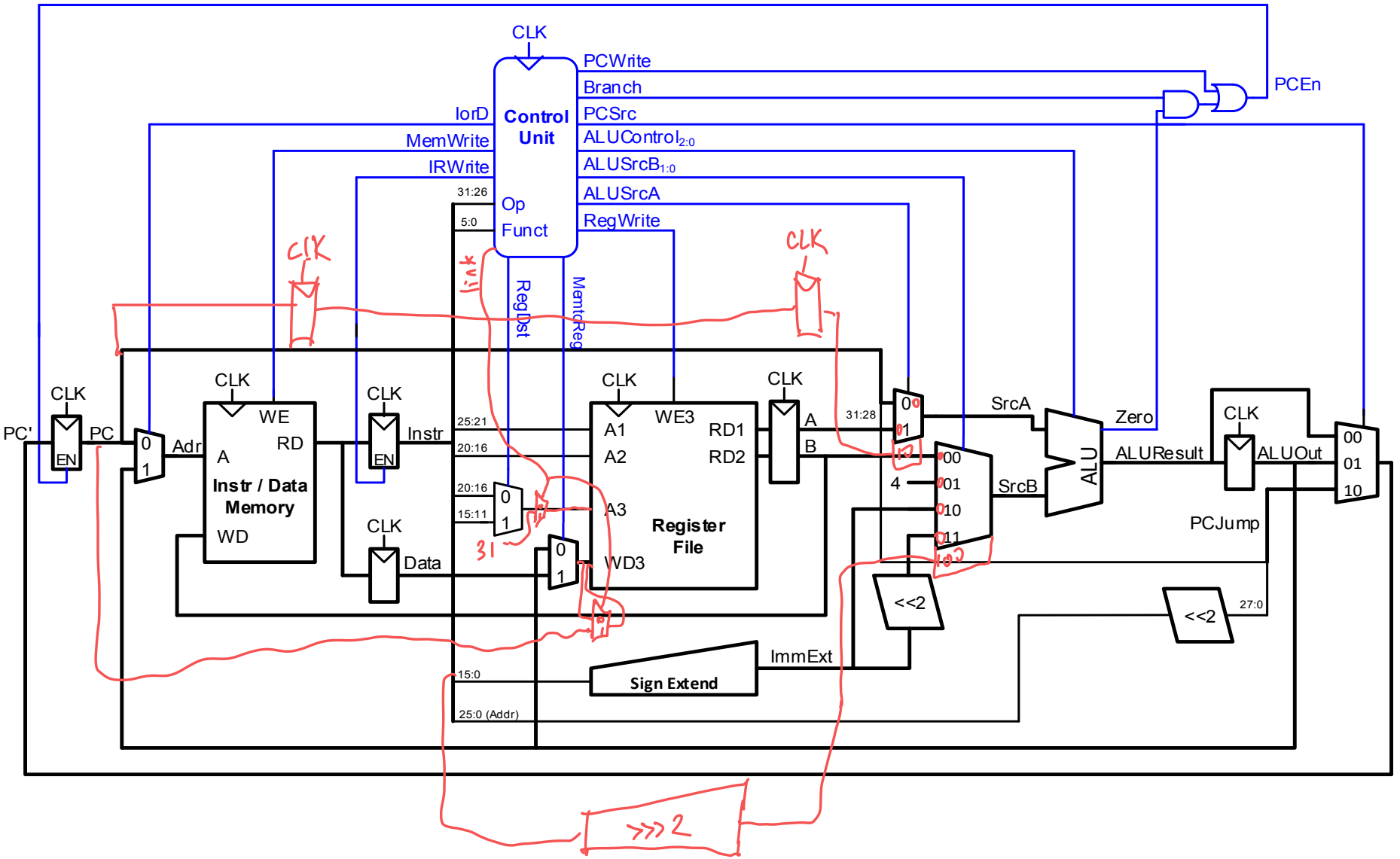
1. a)


PcALUSrc
Jump
link
Control Unit
MemtoReg
MemWrite
Branch
ALUControl5:0
ALUSrc
RegDst
RegWrite
Op
Funct
31:26
5:0
PCSrc
shamt
CLK
PC'
PC
A
RD
Instruction Memory
WE3
A1
A2
A3
WD3
RD1
RD2
Register File
SrcA
SrcB
ALU
Zero
ALUResult
WriteData
WE
Data Memory
WD
ReadData
Result
PCJump
PCPlus4
4
WriteReg4:0
SignImm
Sign Extend
15:0
<<2
PCBranch
27:0
31:28
25:0
>>2

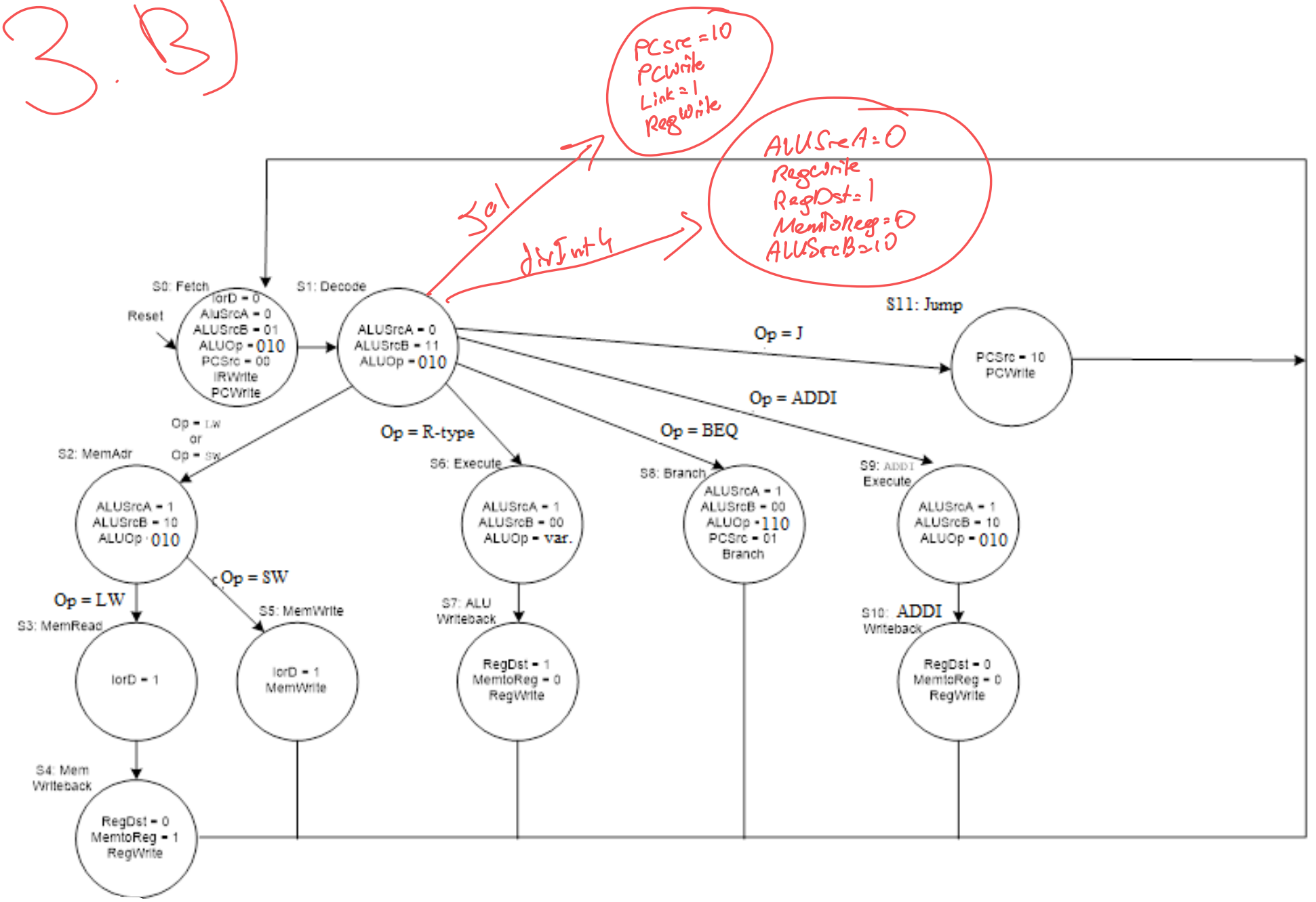
1.B)

Instr	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUControl_{2:0}$	Jump	PcAluSrc	link	
R-type	000000	1	1	0	0	0	0	Diff.	0	0	0	
lw	100011	1	0	1	0	0	1	010 (add)	0	0	0	
sw	101011	0	X	1	0	1	X	010 (add)	0	0	0	
beq	000100	0	X	0	1	0	X	110 (sub)	0	0	0	
j	000010	0	X	X	X	0	X	XXX	1	X	0	
addi	001000	1	0	1	0	0	0	010 (add)	0	0	0	
divInt4	000000	1	1	0	0	0	0	011	0	1	0	
jal	000011	1	X	X	0	0	X	XXX	1	X	1	

3. A)


CLK
PCWrite
Branch
PCSrc
ALUControl2:0
ALUSrcB1:0
ALUSrcA
RegWrite
IorD
MemWrite
IRWrite
Control Unit
Op
Funct
31:26
5:0
PCEn
RegDst
MemtoReg
link
PC'
PC
EN
Adr
WE
RD
A
Instr / Data Memory
WD
Instr
Data
25:21
20:16
15:11
31
A1
A2
A3
WD3
WE3
RD1
RD2
Register File
A
B
31:28
4
SrcA
SrcB
ALU
Zero
ALUResult
ALUOut
PCJump
<<2
27:0
ImmExt
Sign Extend
15:0
25:0 (Addr)
>>2

3.B)



PCSrc = 10
PCWrite
Link = 1
RegWrite
ALUSrcA = 0
RegWrite
RegDst = 1
MemToReg = 0
ALUSrcB = 10
Jal
divInt4
S0: Fetch
Reset
IorD = 0
AluSrcA = 0
ALUSrcB = 01
ALUOp = 010
PCSrc = 00
IRWrite
PCWrite
S1: Decode
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 010
S11: Jump
Op = J
PCSrc = 10
PCWrite
Op = ADDI
Op = BEQ
Op = R-type
Op = LW
or
Op = SW
S2: MemAdr
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 010
S6: Execute
ALUSrcA = 1
ALUSrcB = 00
ALUOp = var.
S8: Branch
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 110
PCSrc = 01
Branch
S9: ADDI
Execute
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 010
Op = LW
Op = SW
S3: MemRead
IorD = 1
S5: MemWrite
IorD = 1
MemWrite
S7: ALU
Writeback
RegDst = 1
MemtoReg = 0
RegWrite
S10: ADDI
Writeback
RegDst = 0
MemtoReg = 0
RegWrite
S4: Mem
Writeback
RegDst = 0
MemtoReg = 1
RegWrite