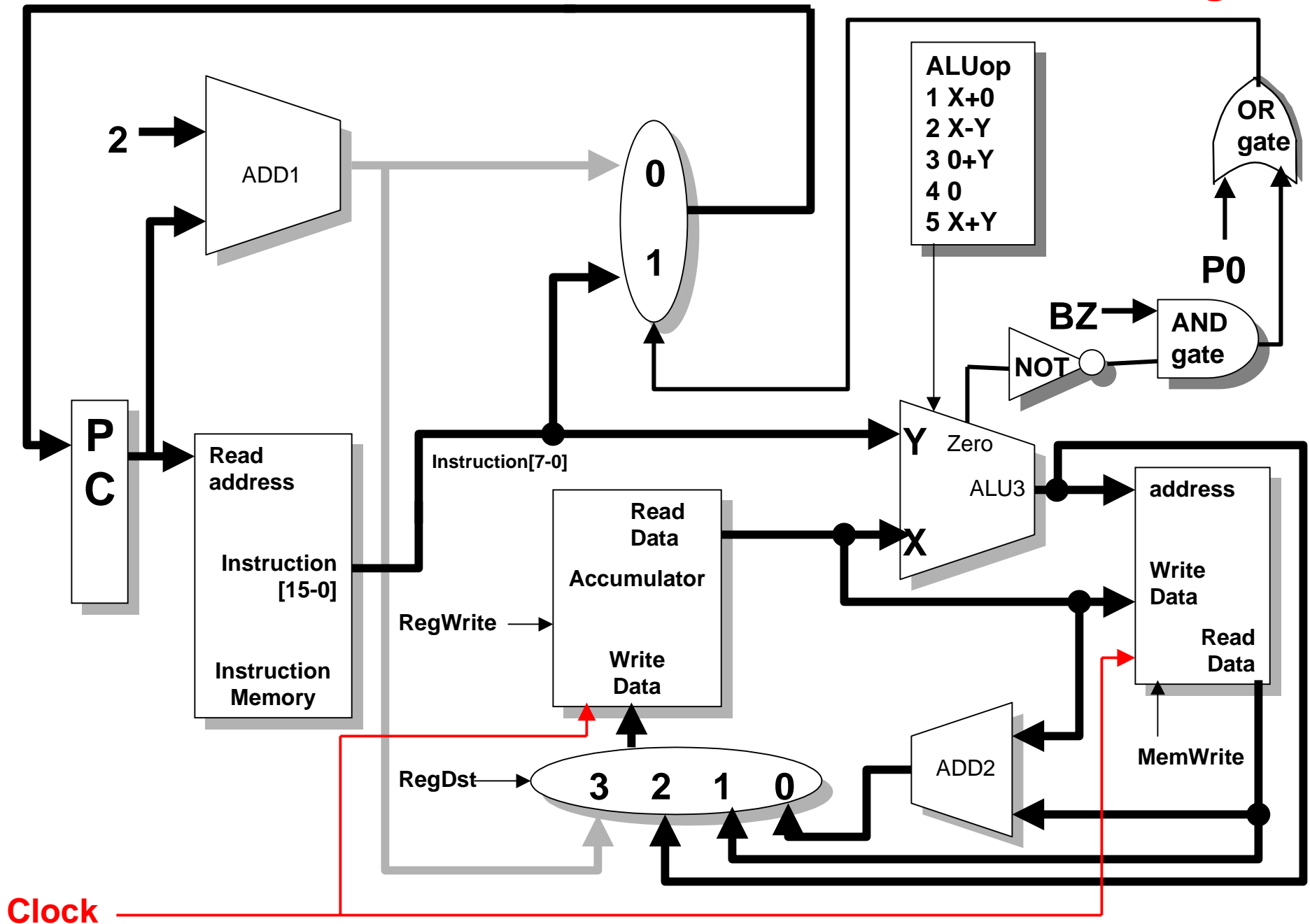


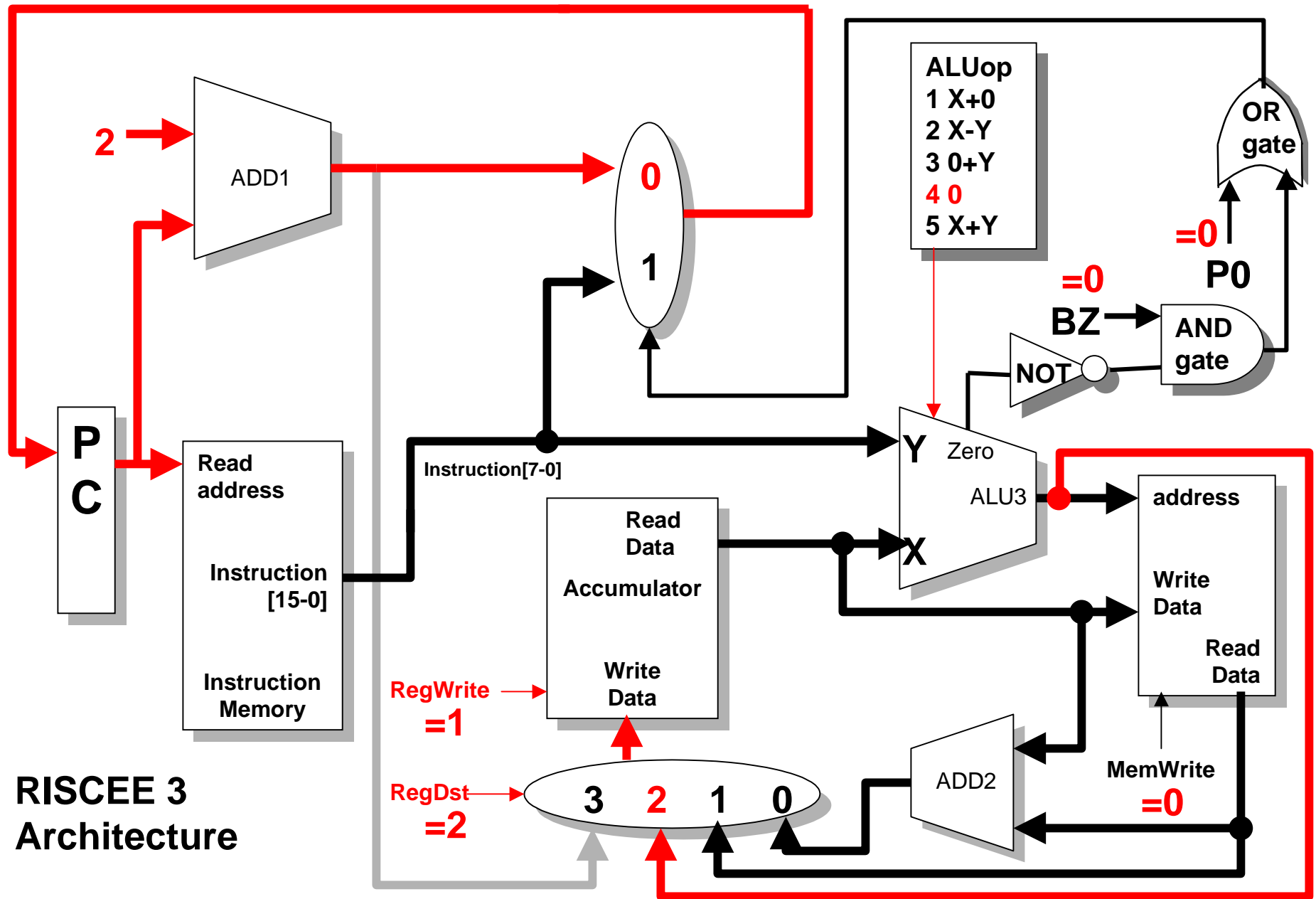
RISCEE 3 Architecture

Clock = load value into register



Instruction: clear

Operation: $A=0;$



RISCEE 3
Architecture

RegDst=2

ALU=4

MemWrite=0

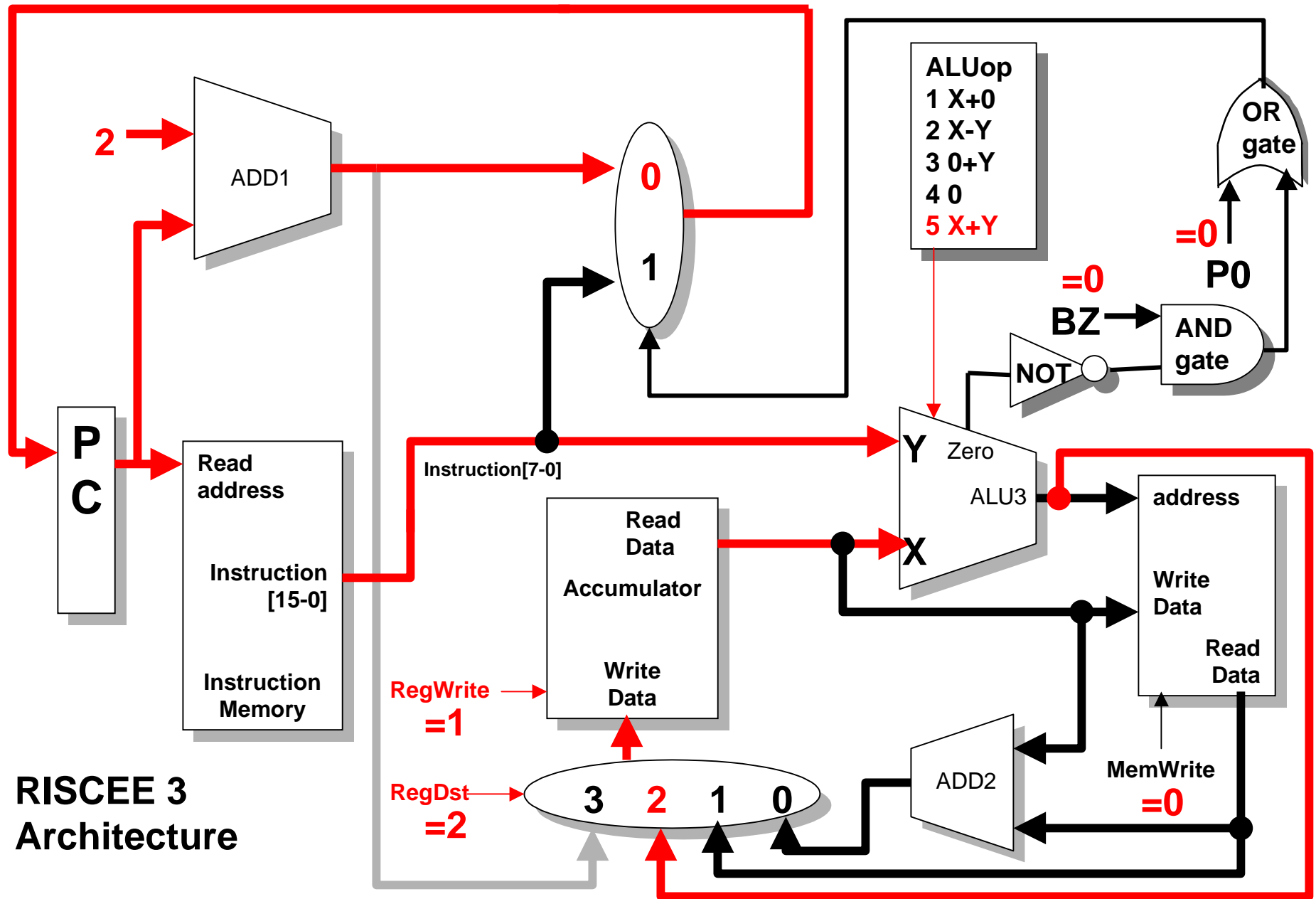
RegWrite=1

BZ=0

P0=0

Instruction: `addi data8`

Operation: $A = A + \text{data8};$



RISCEE 3
Architecture

RegDst=2

ALU=5

MemWrite=0

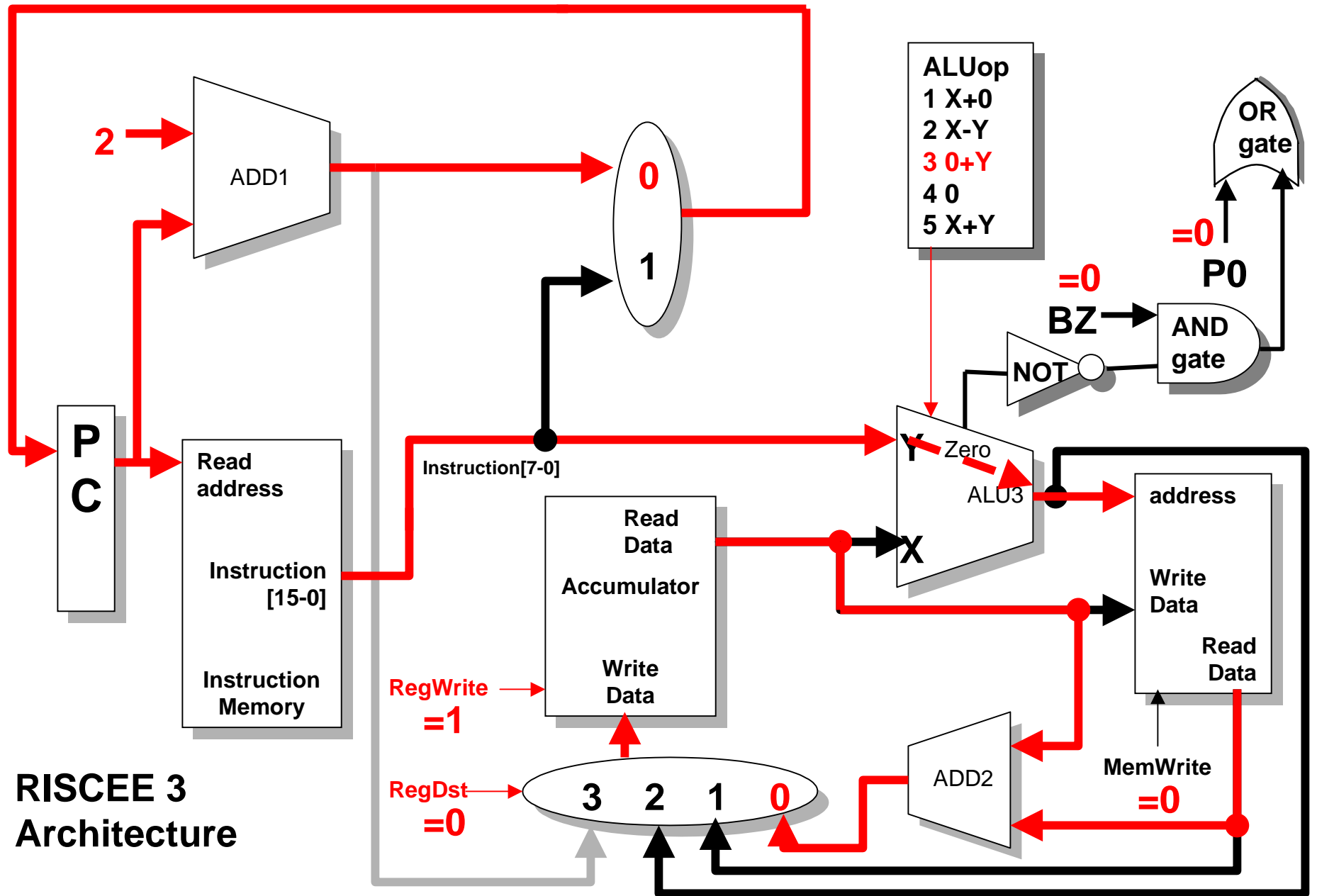
RegWrite=1

BZ=0

P0=0

Instruction: add addr8

Operation: $A = A + \text{Memory}[\text{addr8}]$;



RISCEE 3
Architecture

RegDst=0

ALU=3

MemWrite=0

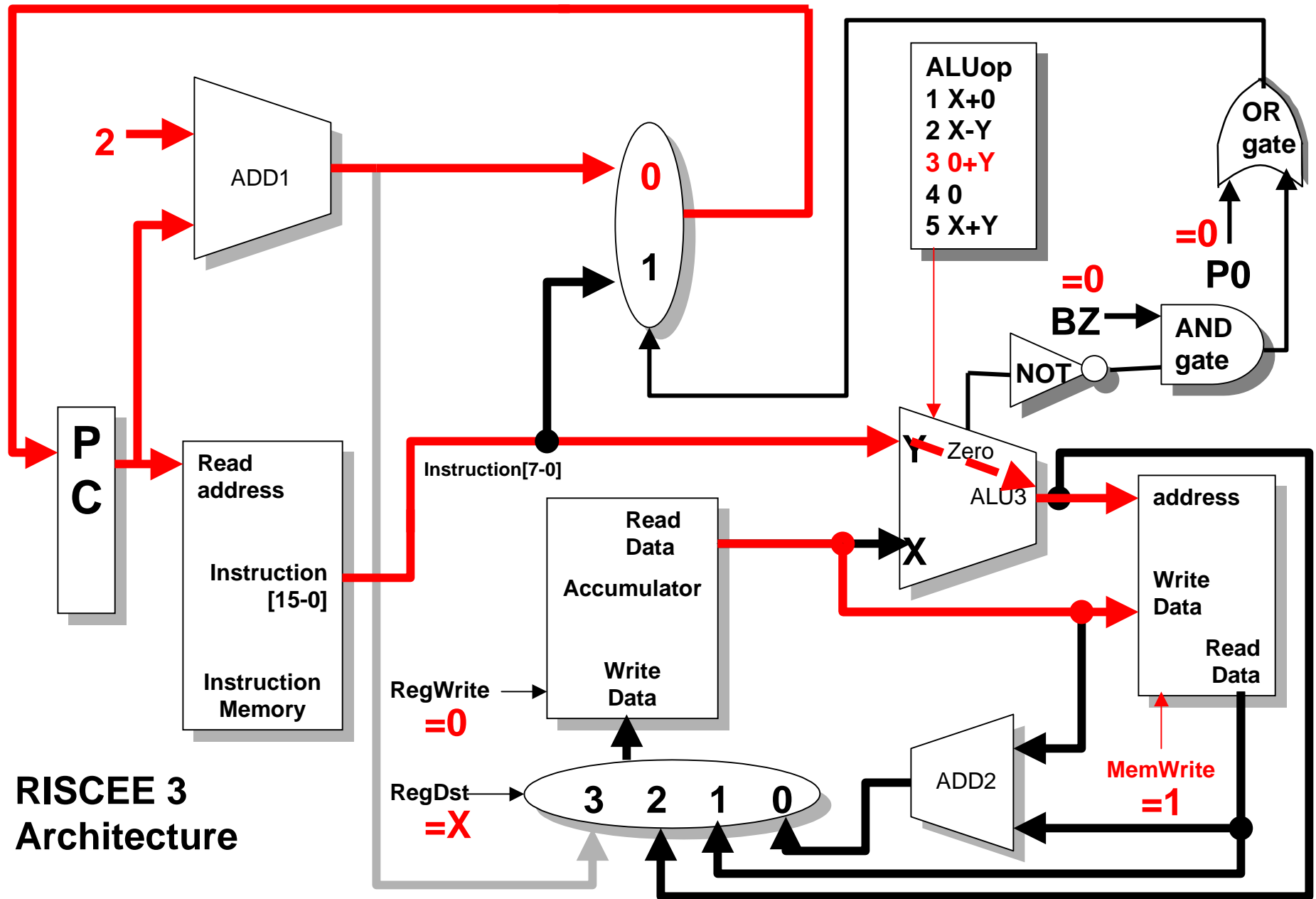
RegWrite=1

BZ=0

P0=0

Instruction: store addr8

Operation: Memory[addr8]=A;



RISCEE 3
Architecture

RegDst=X

ALU=3

MemWrite=1

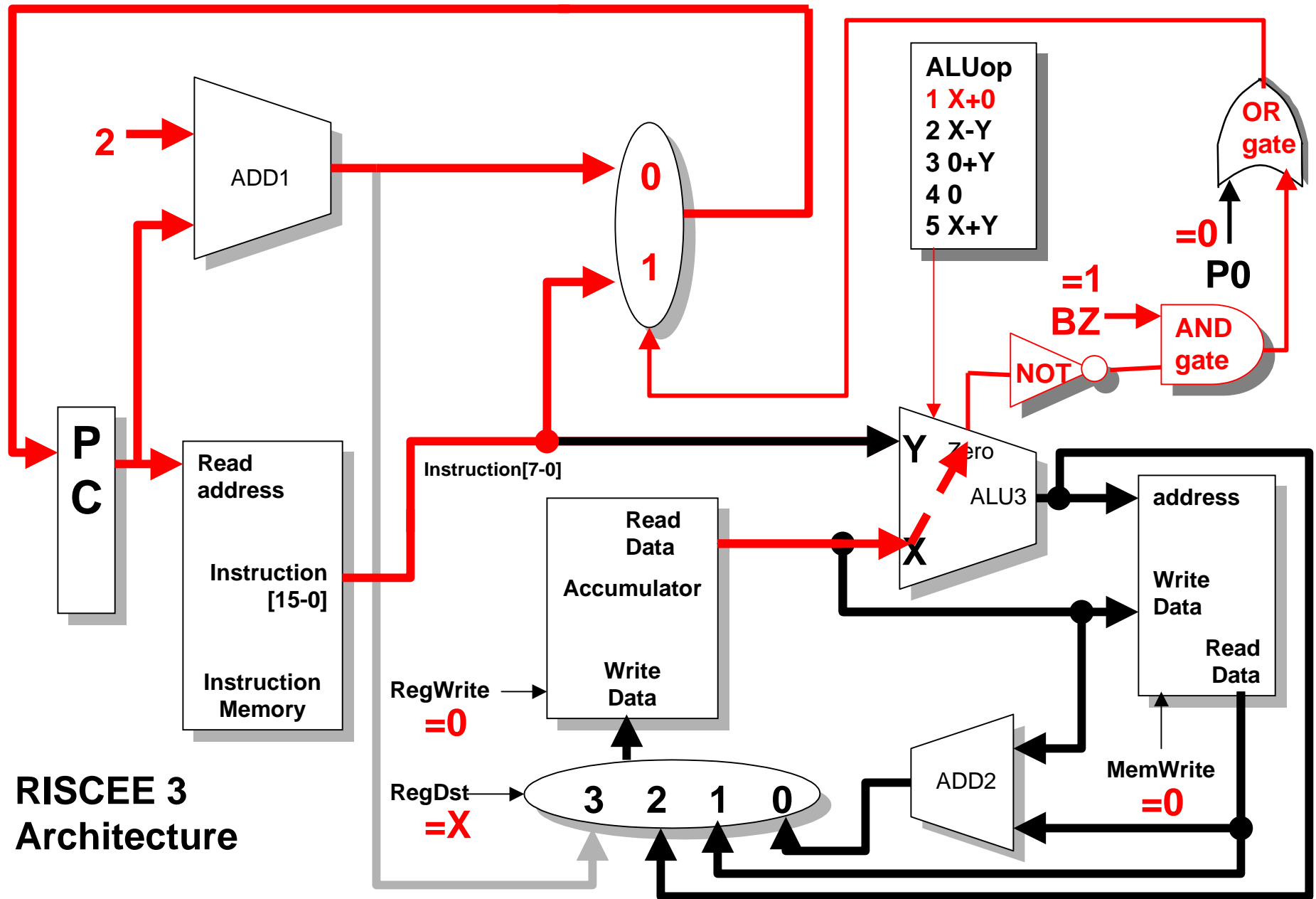
RegWrite=0

BZ=0

P0=0

Instruction: bne addr8

Operation: if (A != 0) { pc=addr8; }



RISCEE 3
Architecture

RegDst=X

ALU=3

MemWrite=1

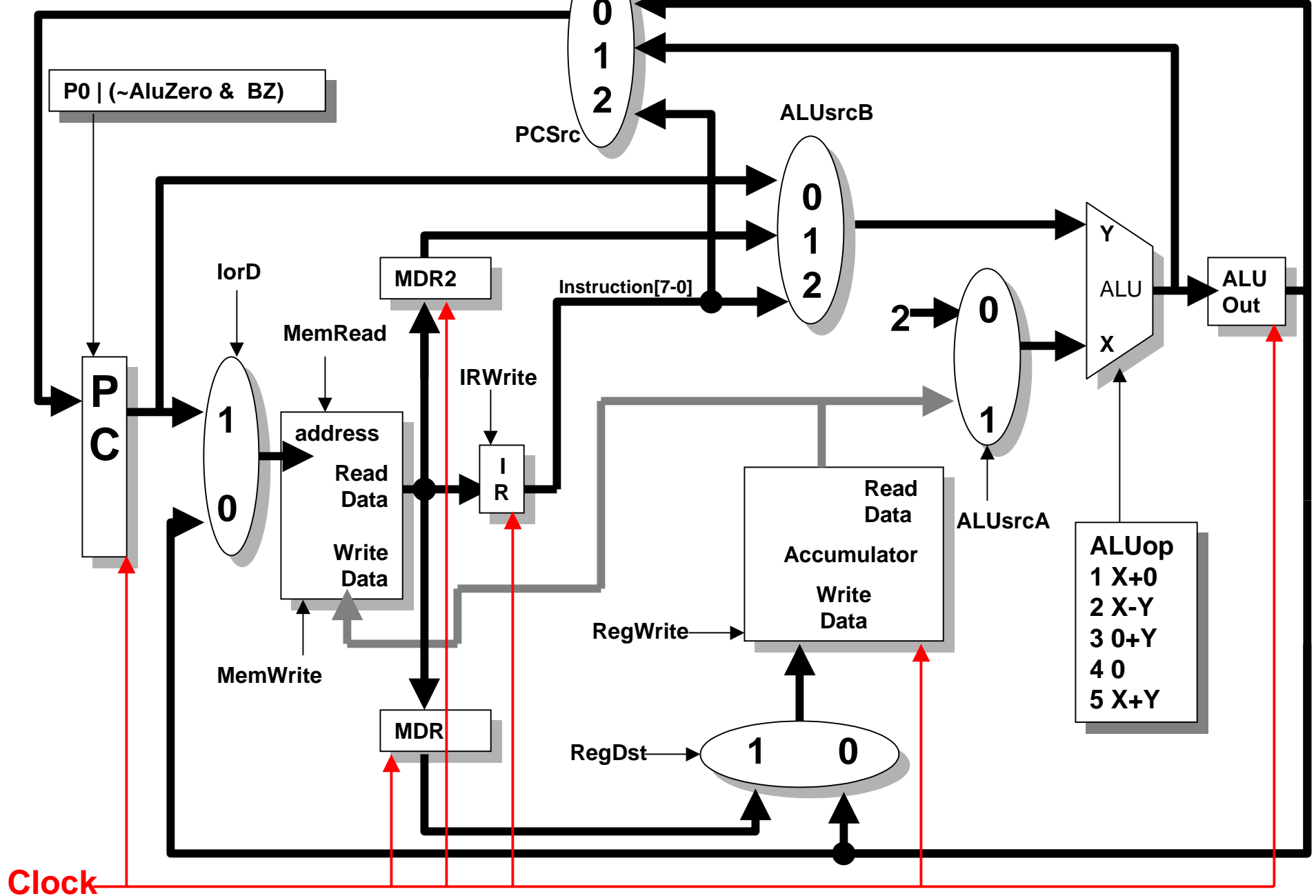
RegWrite=0

BZ=0

P0=0

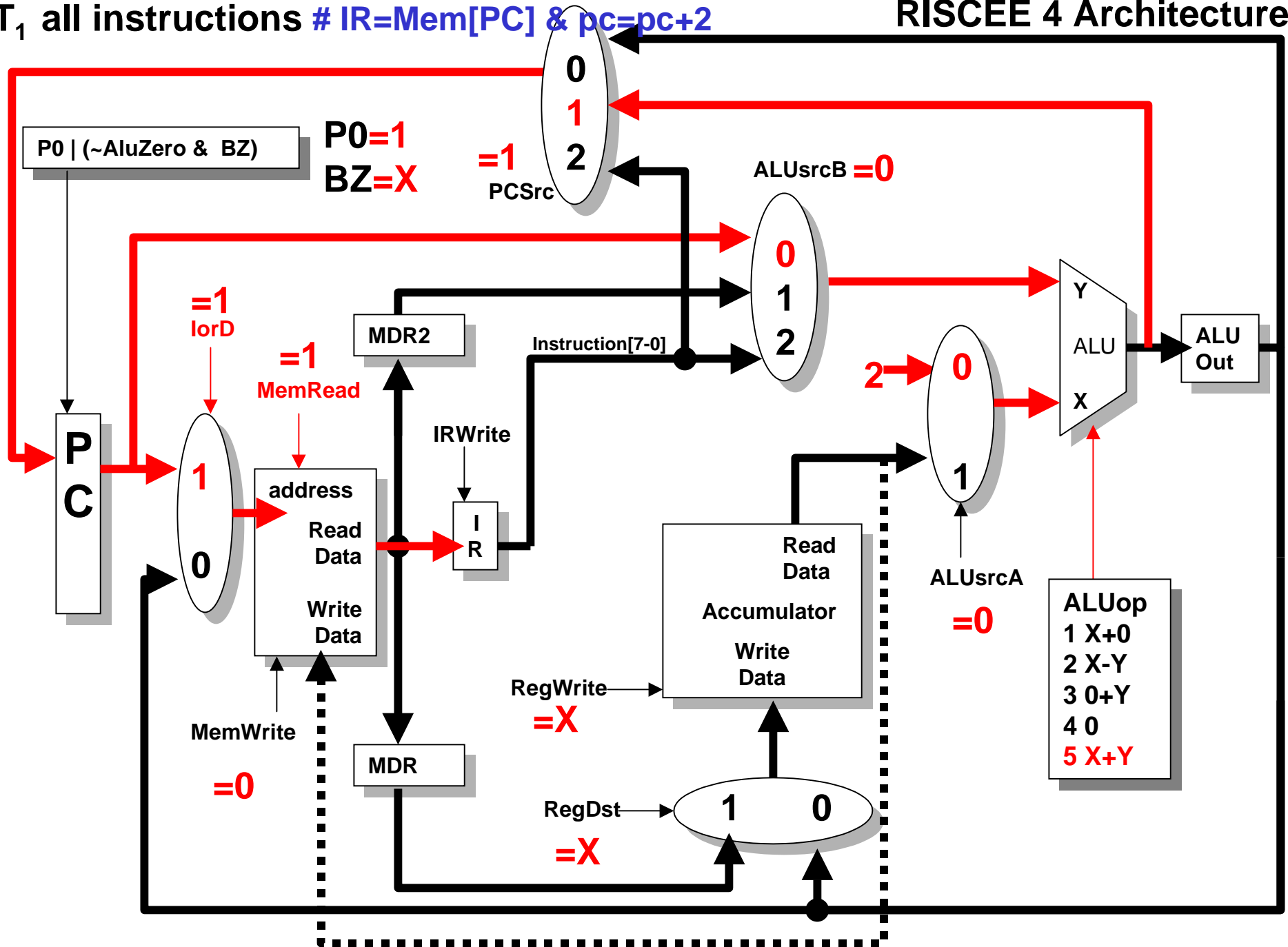
RISCEE 4 Architecture

Clock = load value into register



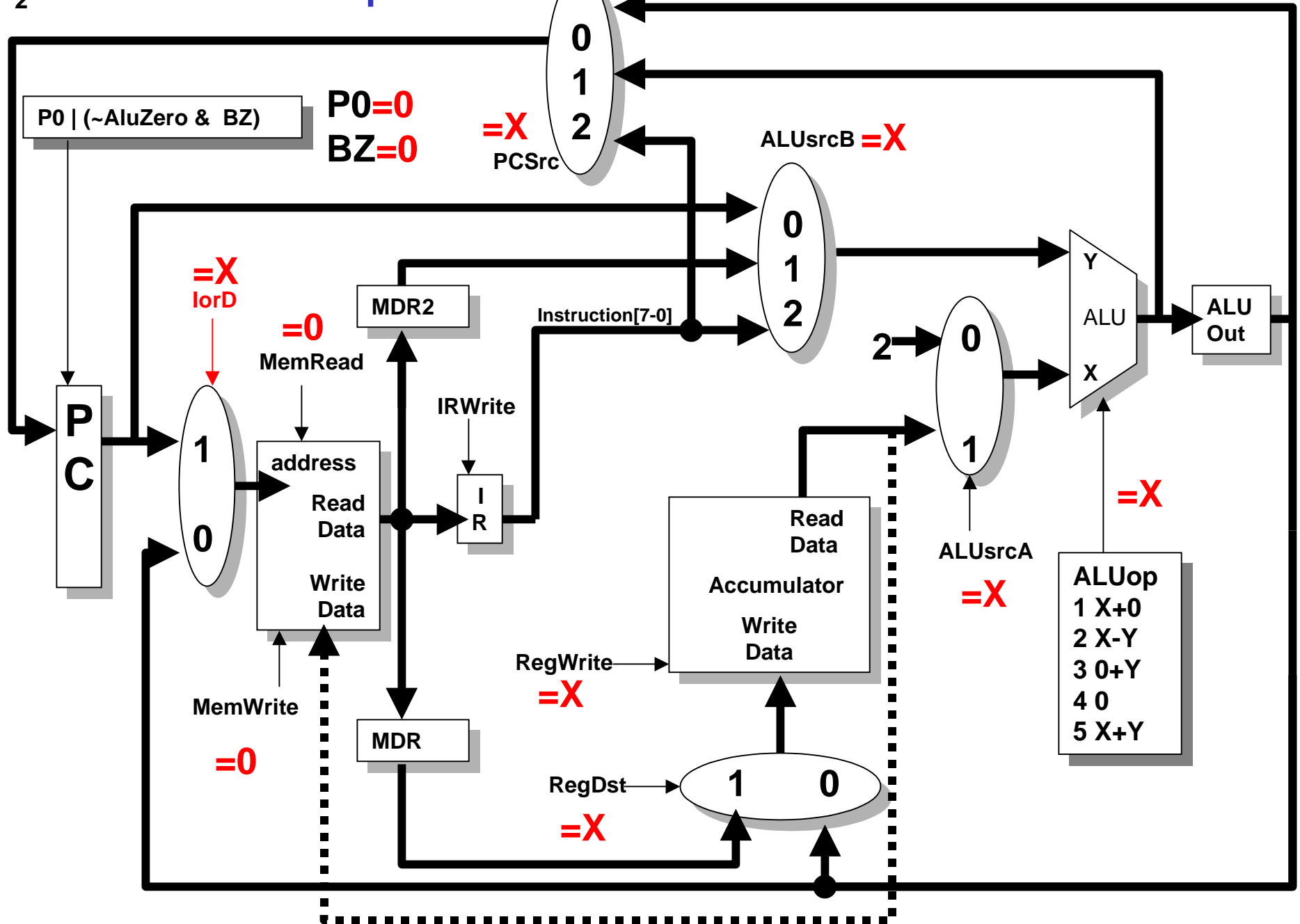
T₁ all instructions # IR=Mem[PC] & pc=pc+2

RISCEE 4 Architecture



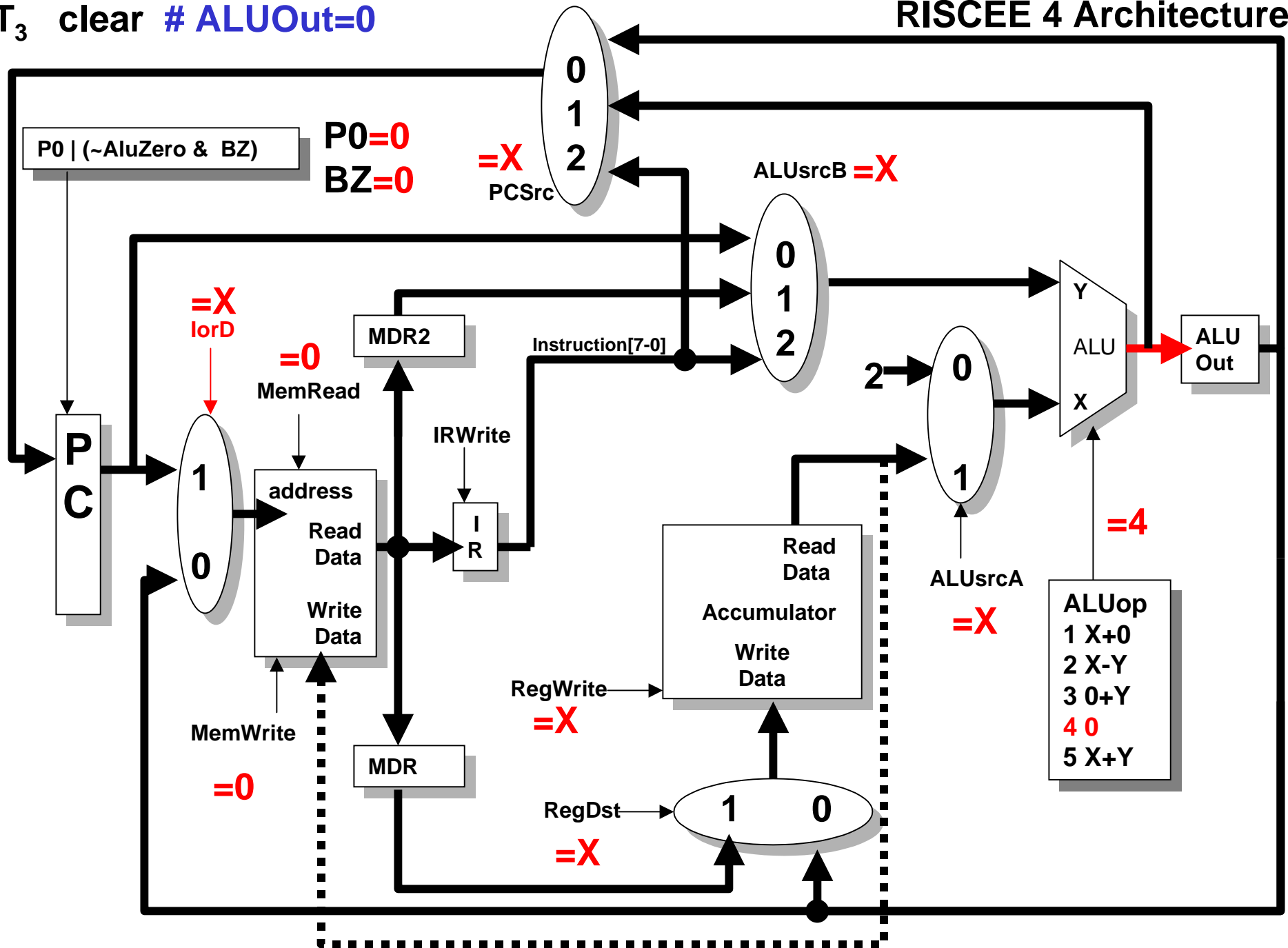
T₂ all instructions #opcode decode

RISCEE 4 Architecture



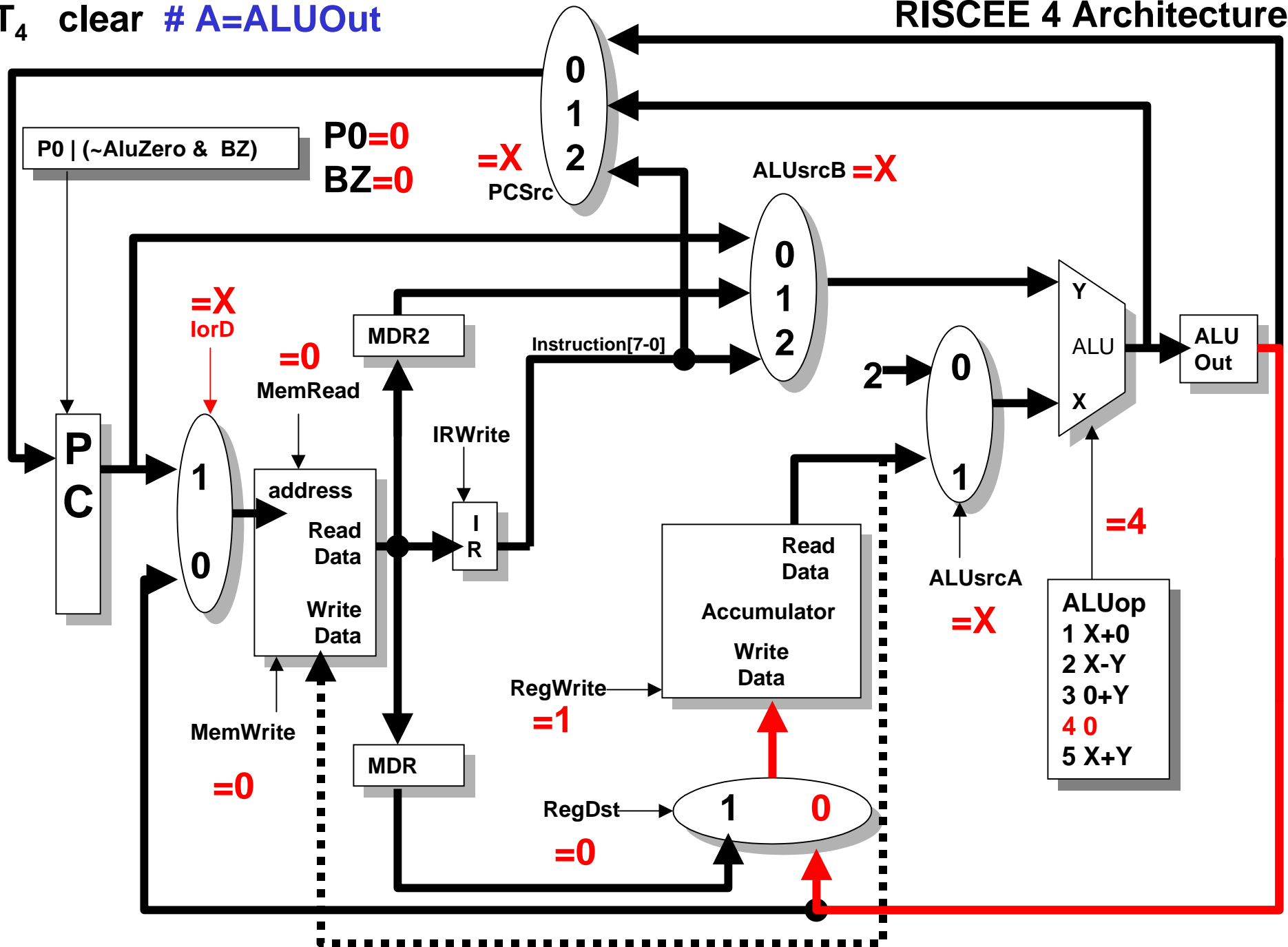
T₃ clear #ALUOut=0

RISCEE 4 Architecture



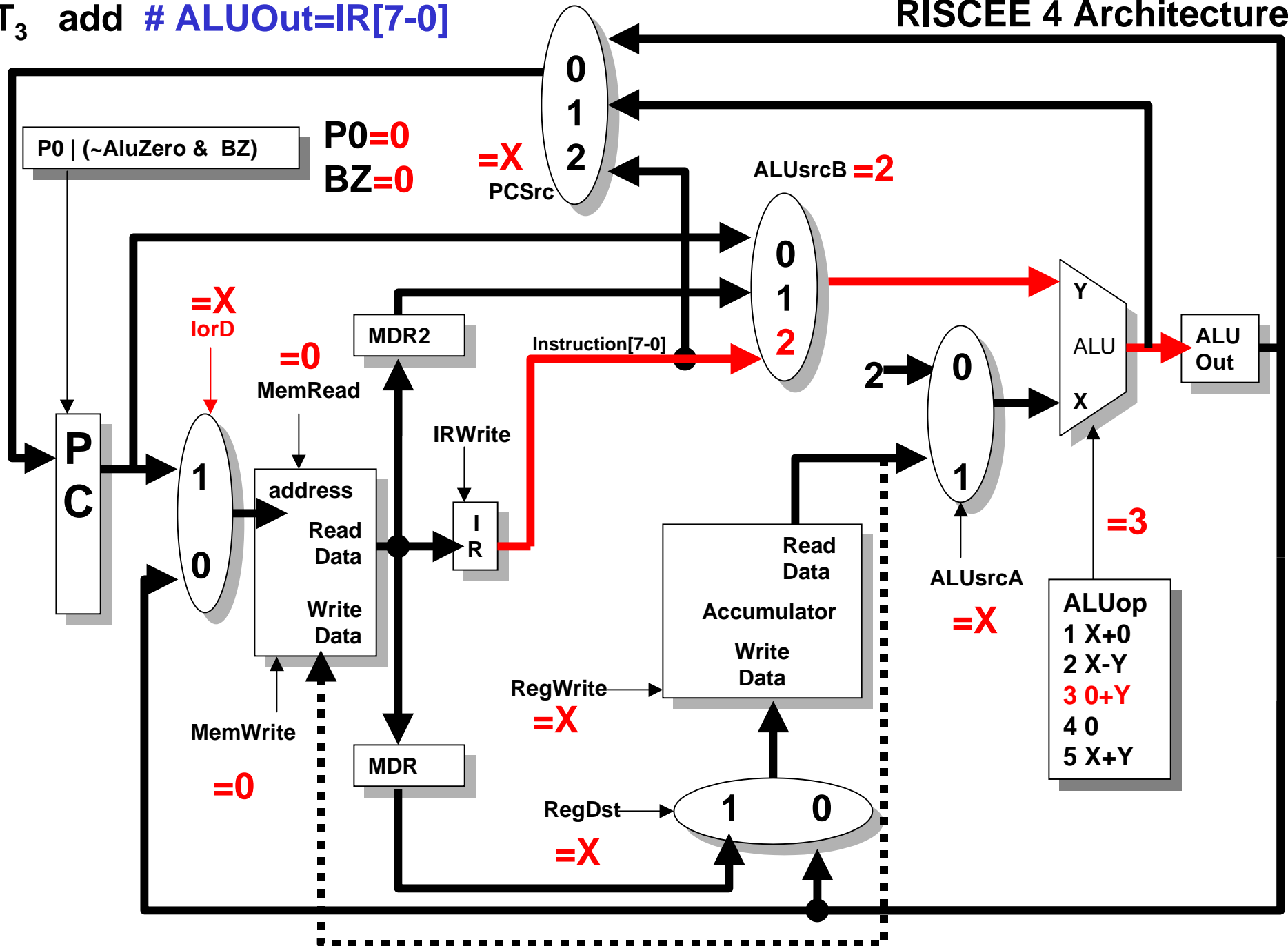
T₄ clear # A=ALUOut

RISCEE 4 Architecture



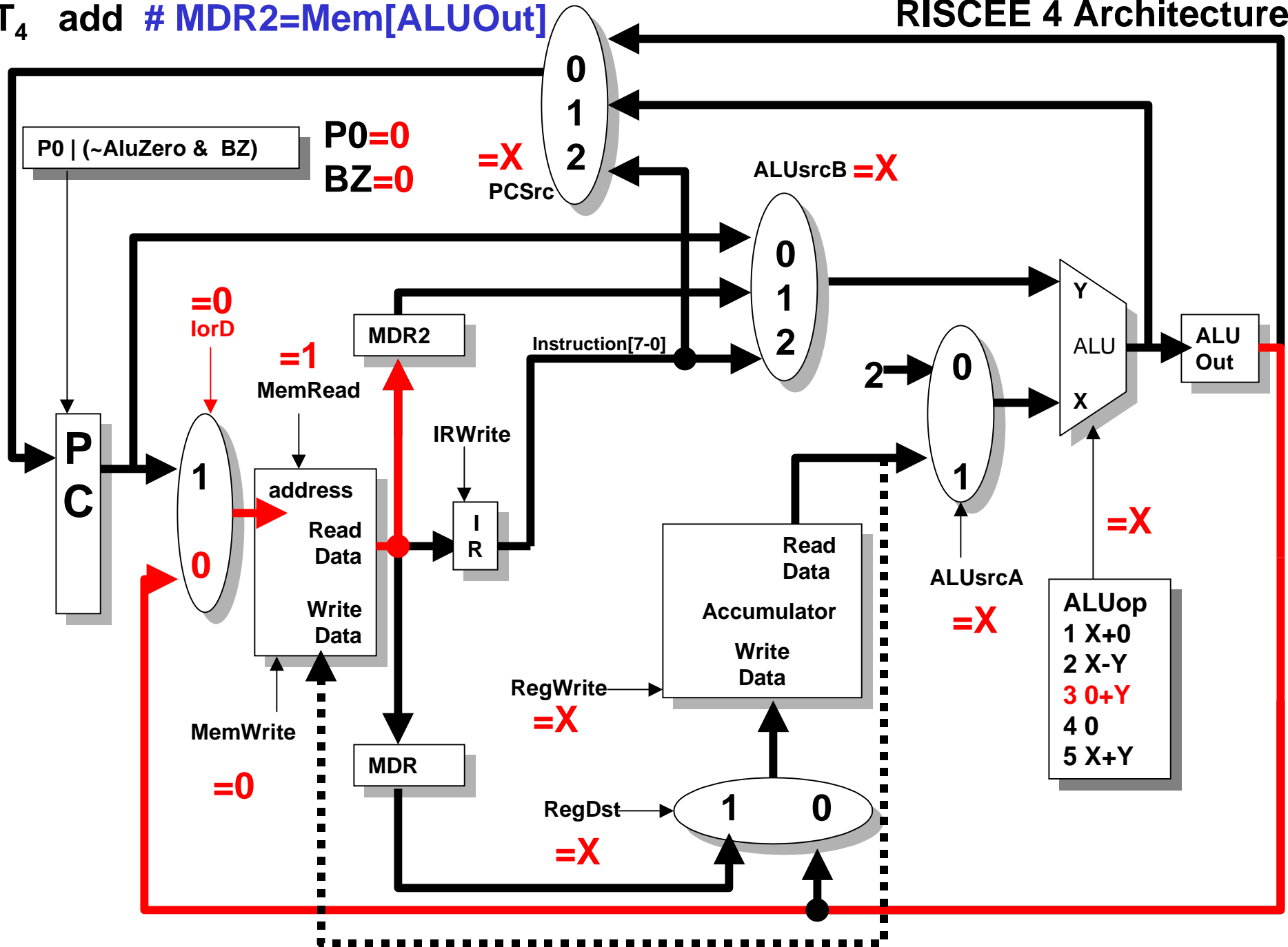
T₃ add # ALUOut=IR[7-0]

RISCEE 4 Architecture



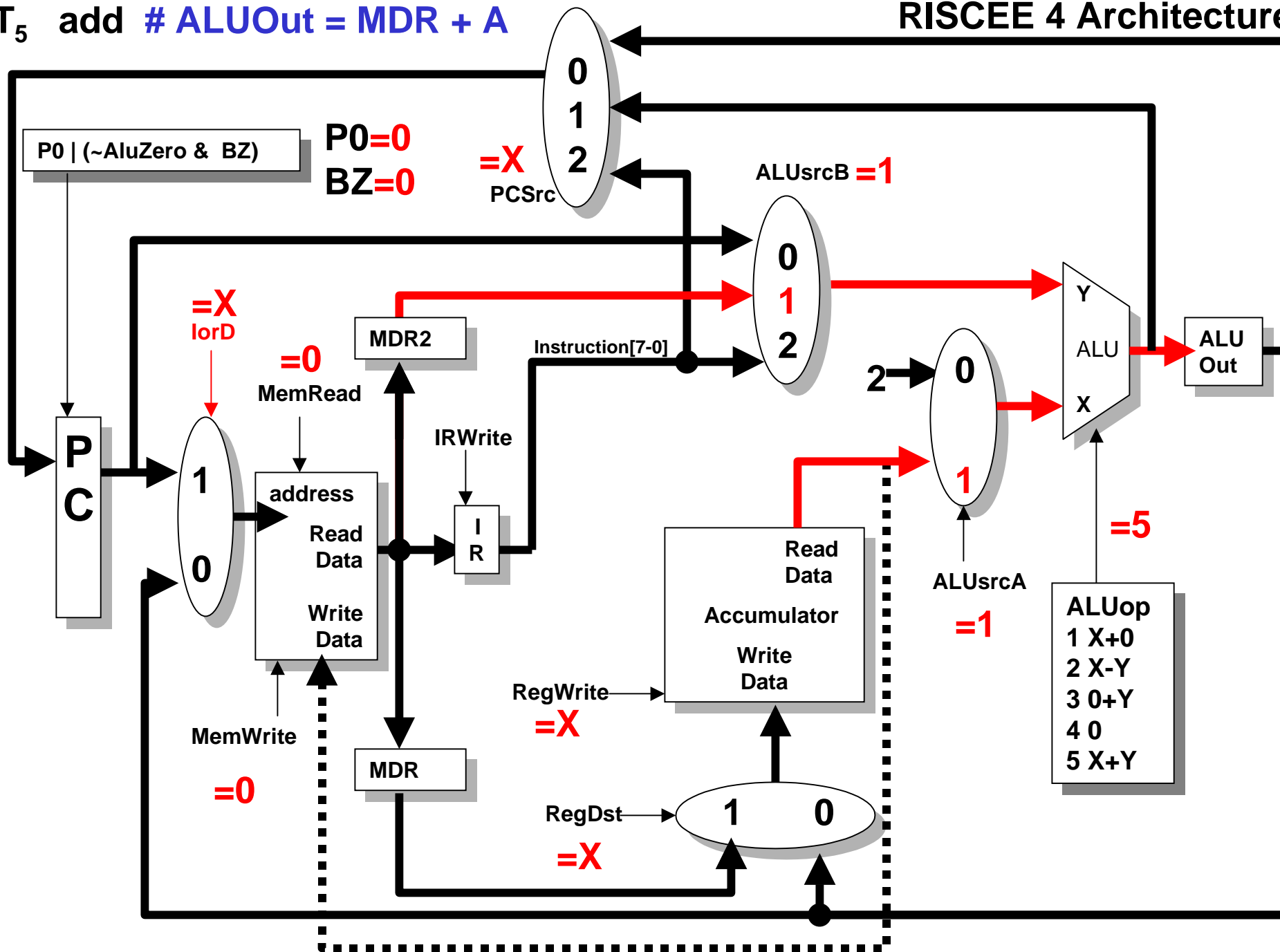
T₄ add # MDR2=Mem[ALUOut]

RISCEE 4 Architecture



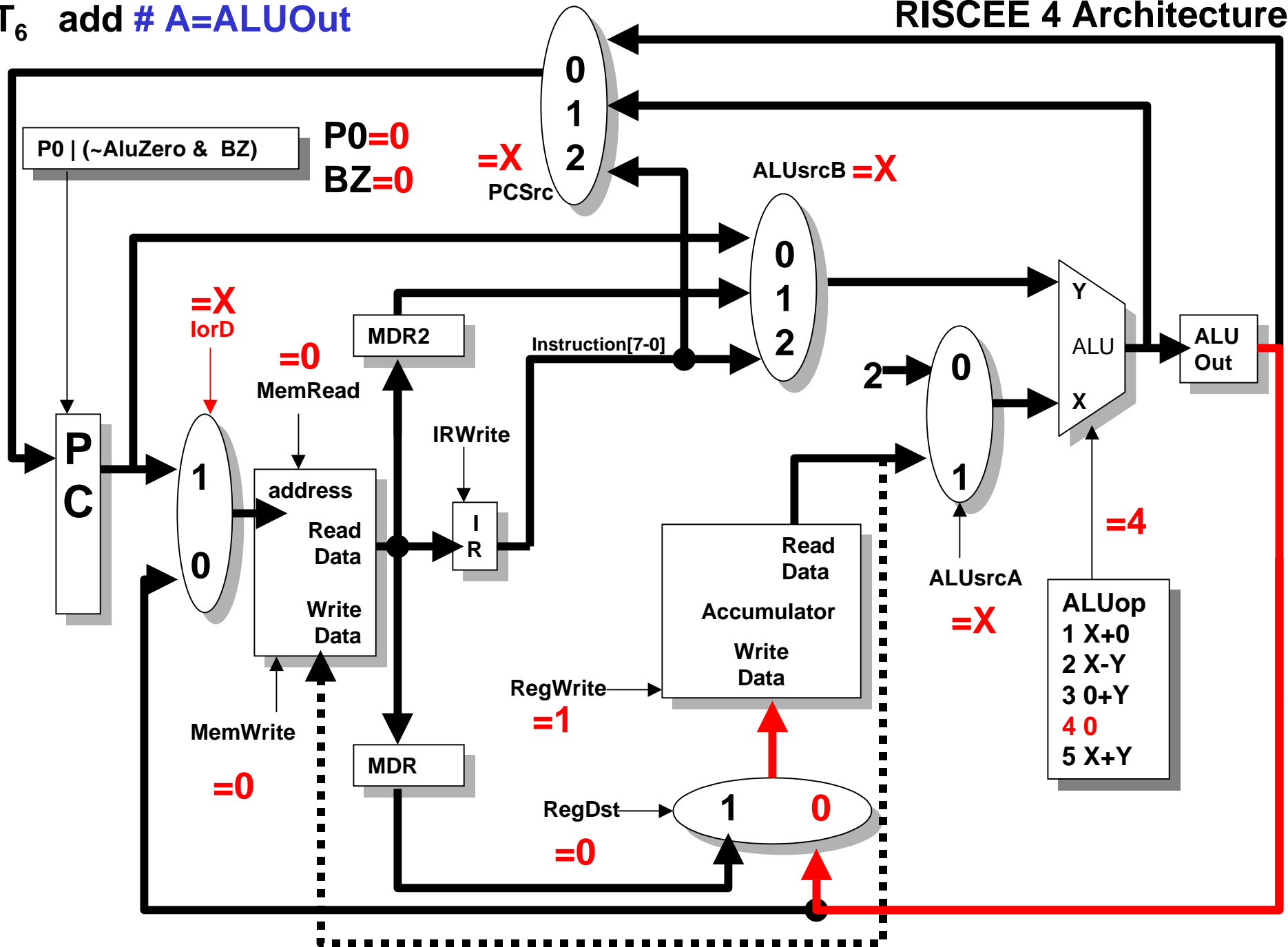
T₅ add # ALUOut = MDR + A

RISCEE 4 Architecture



T₆ add # A=ALUOut

RISCEE 4 Architecture



T₃ bne # if(A!=0) { PC=IR[7-0] }

RISCEE 4 Architecture

