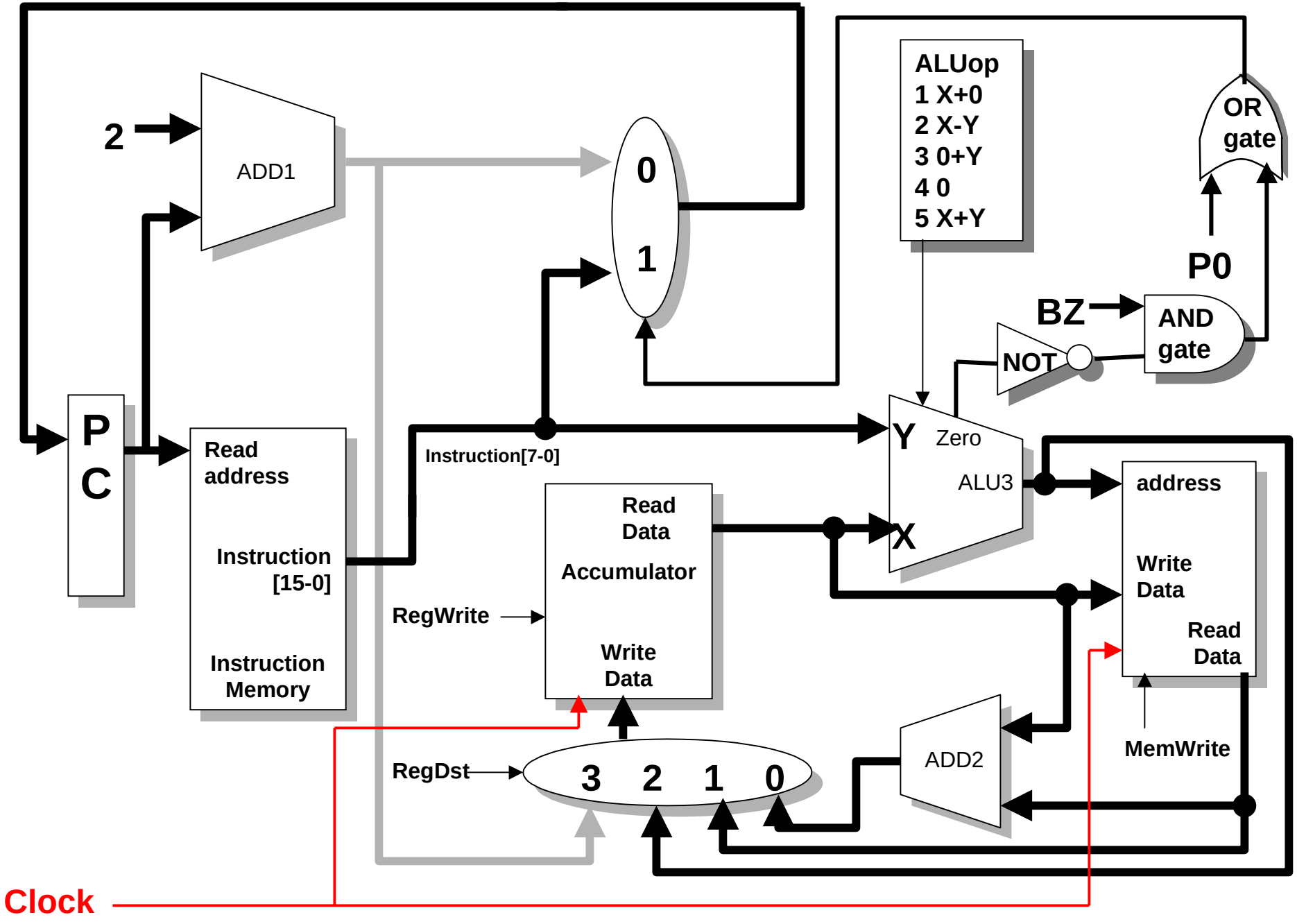
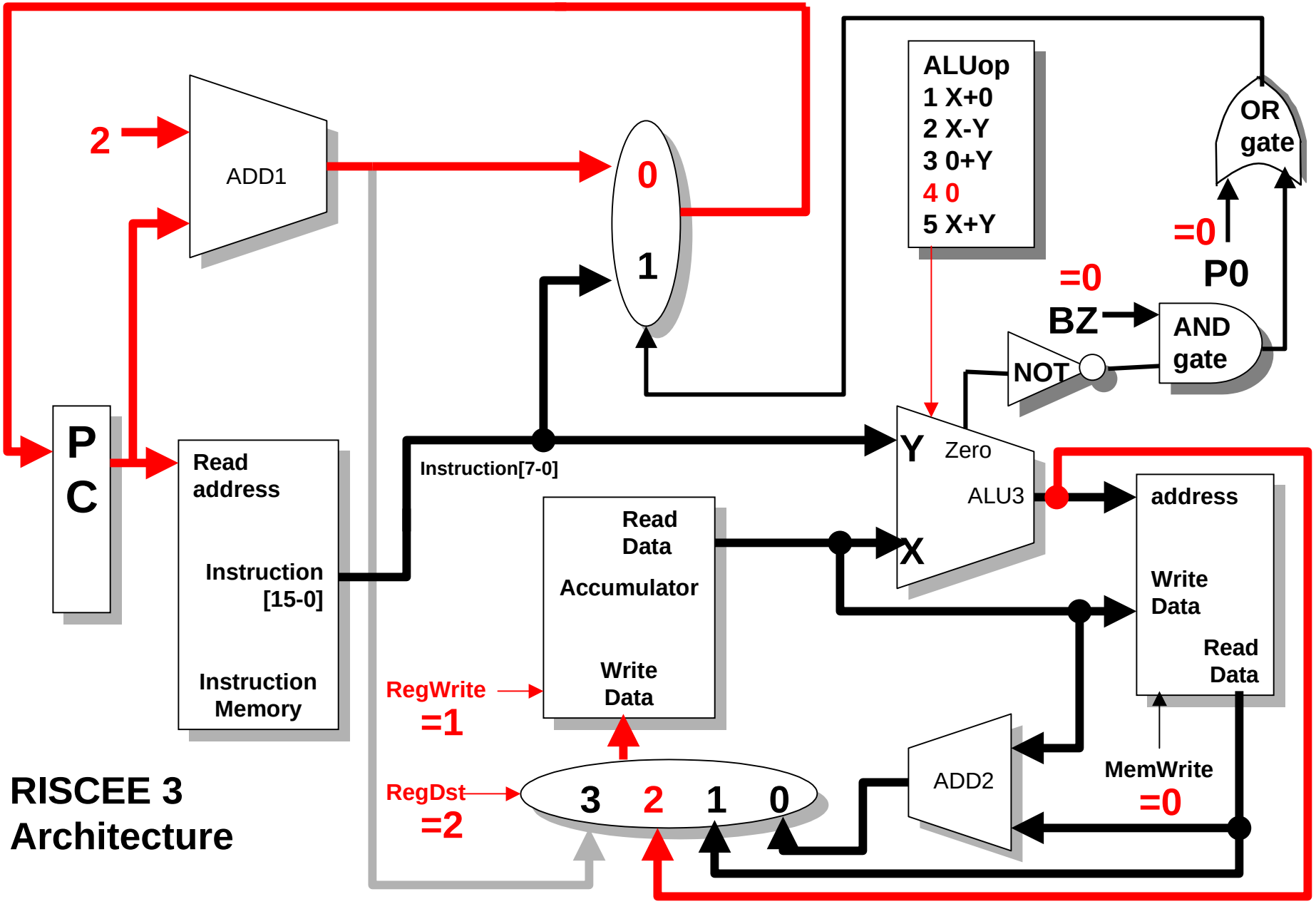
RISCEE 3 Architecture

Clock = load value into register



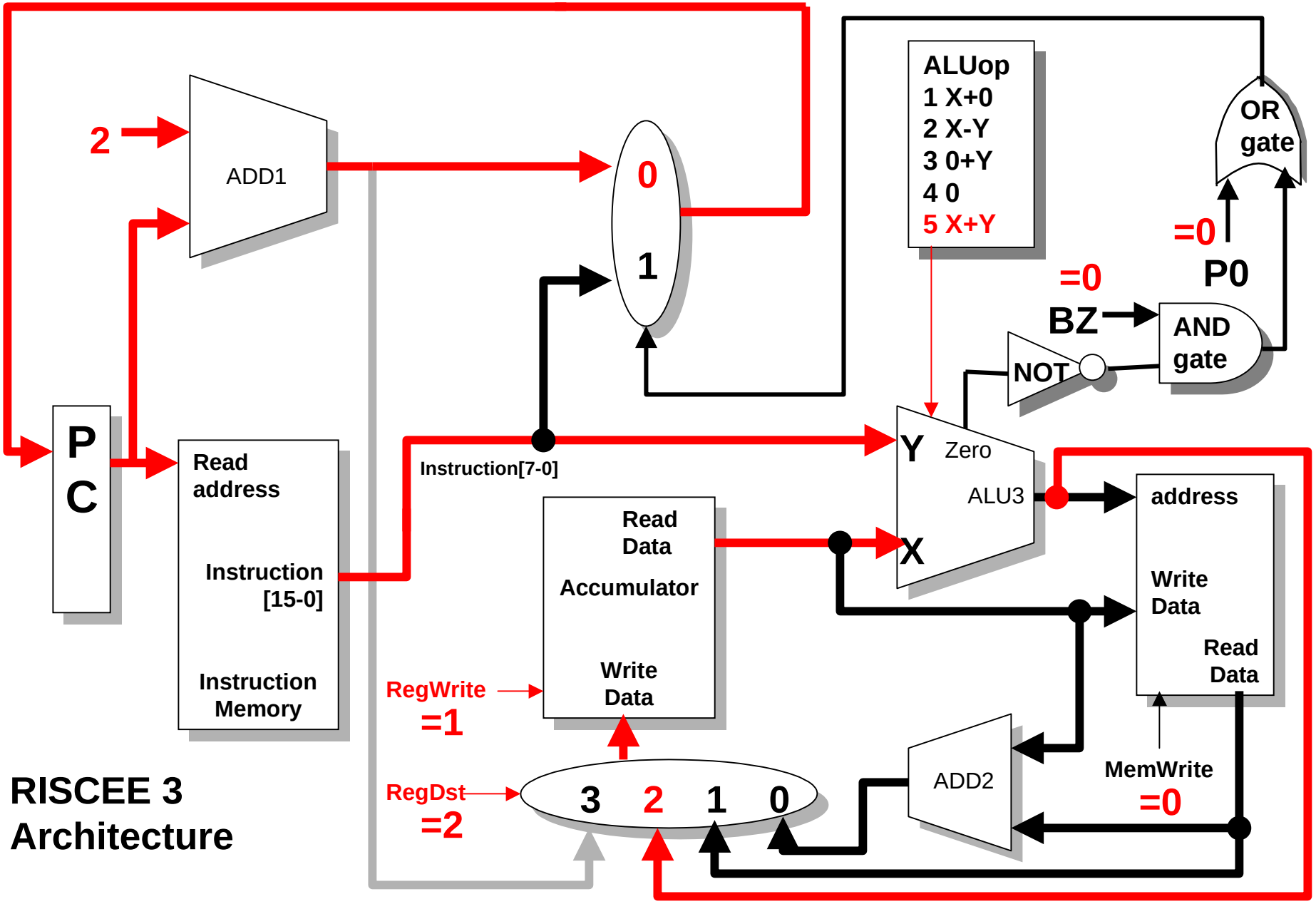
Instruction: clear **Operation: A=0;**



RISCEE 3 Architecture

RegDst=2 ALU=4 MemWrite=0 RegWrite=1 BZ=0 P0=0

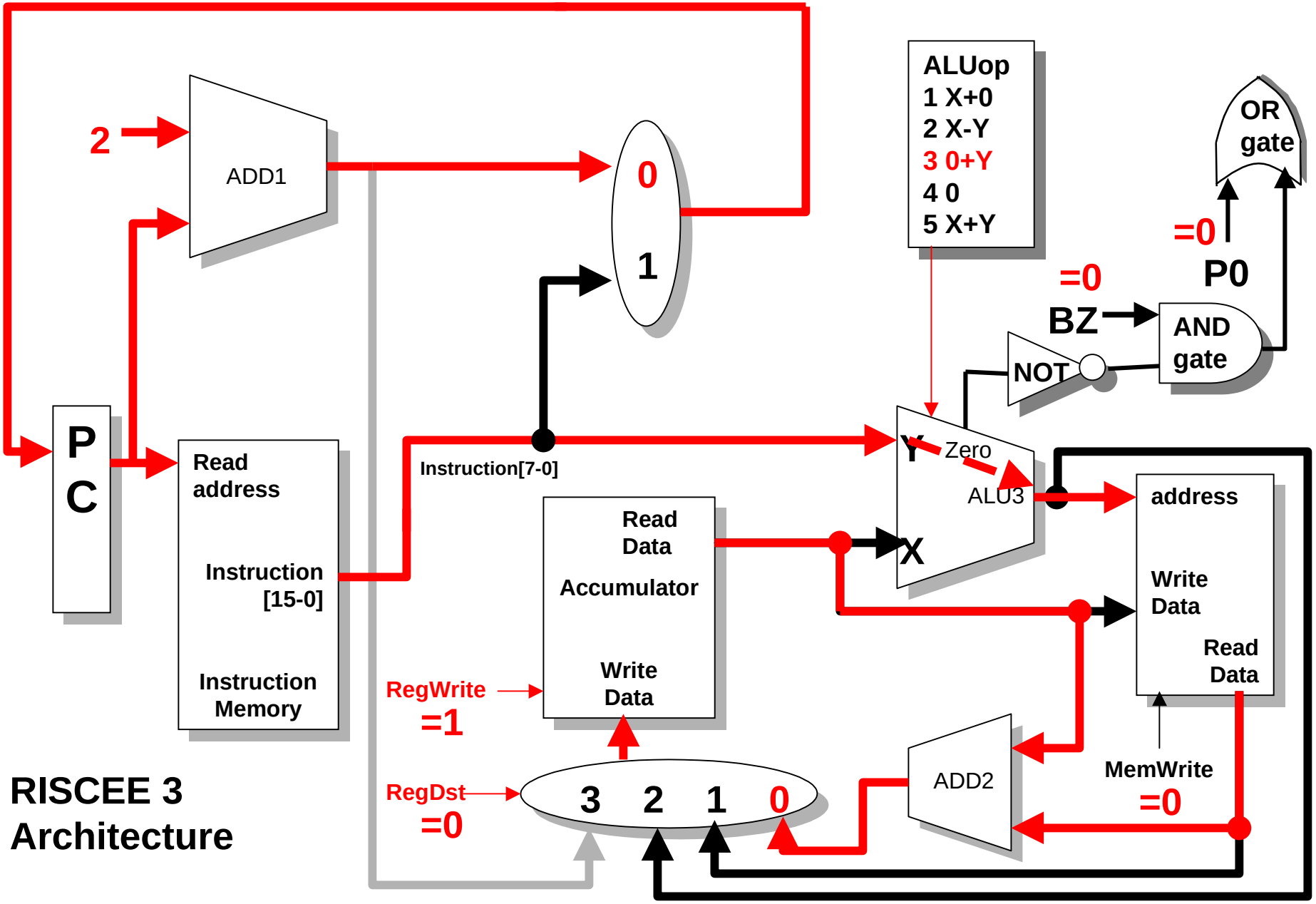
Instruction: addi data8 **Operation: A = A + data8;**



RISCEE 3 Architecture

RegDst=2 ALU=5 MemWrite=0 RegWrite=1 BZ=0 P0=0

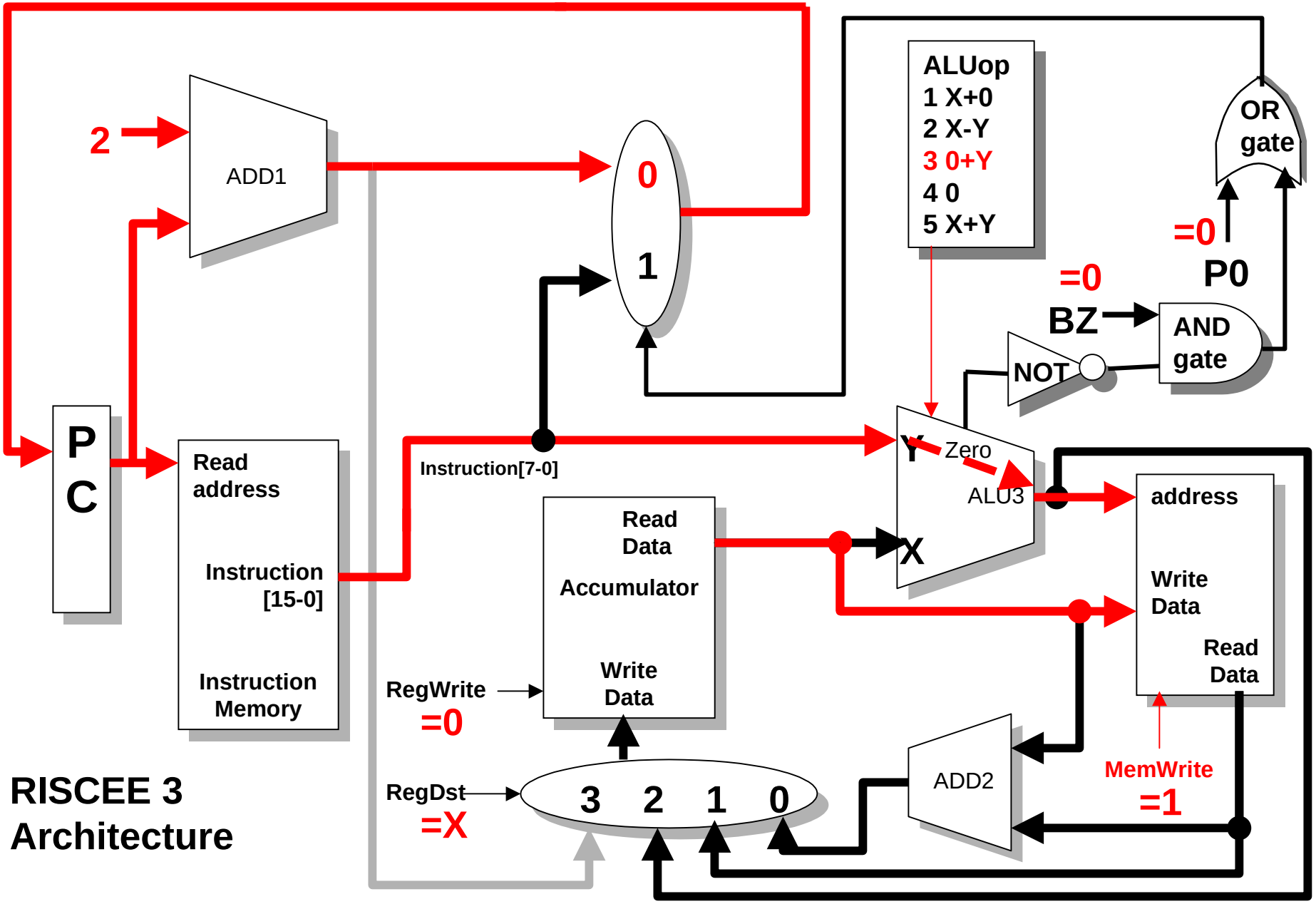
Instruction: add addr8 **Operation: A = A + Memory[addr8];**



RISCEE 3 Architecture

RegDst=0 ALU=3 MemWrite=0 RegWrite=1 BZ=0 P0=0

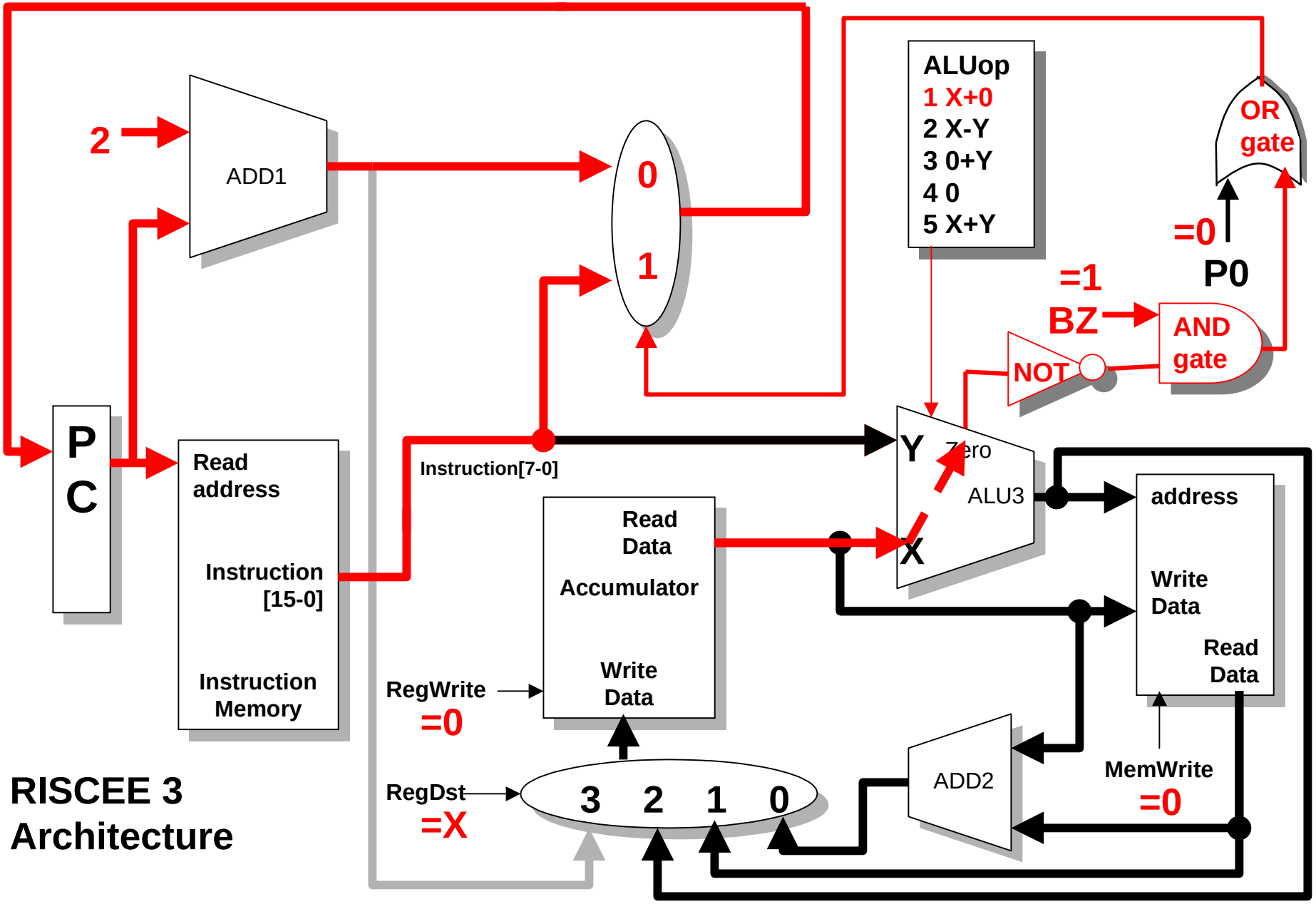
Instruction: store addr8 **Operation: Memory[addr8]=A;**



RISCEE 3 Architecture

RegDst=X ALU=3 MemWrite=1 RegWrite=0 BZ=0 P0=0

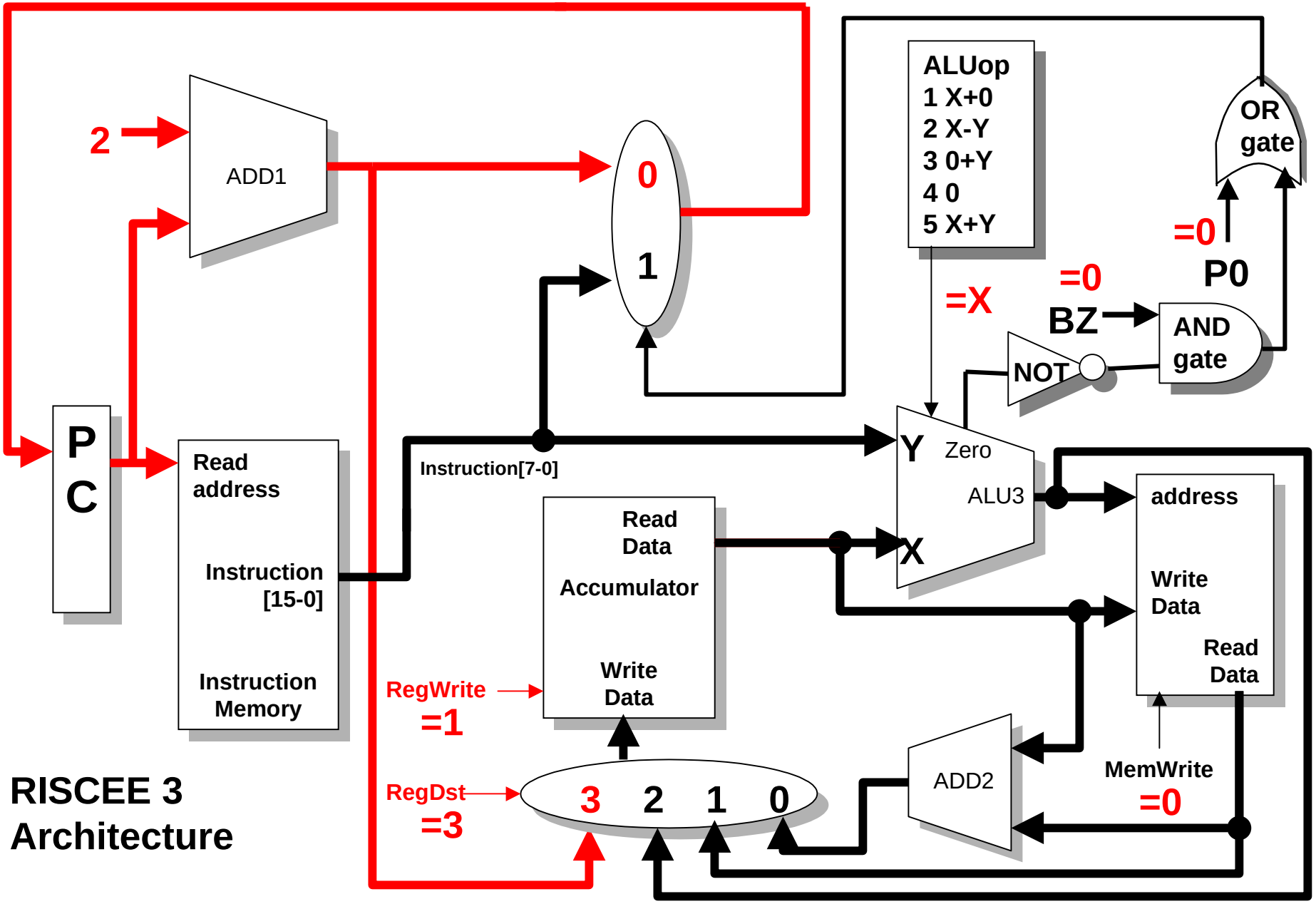
Instruction: bne addr8 **Operation: if (A != 0) { pc=addr8; }**



RISCEE 3 Architecture

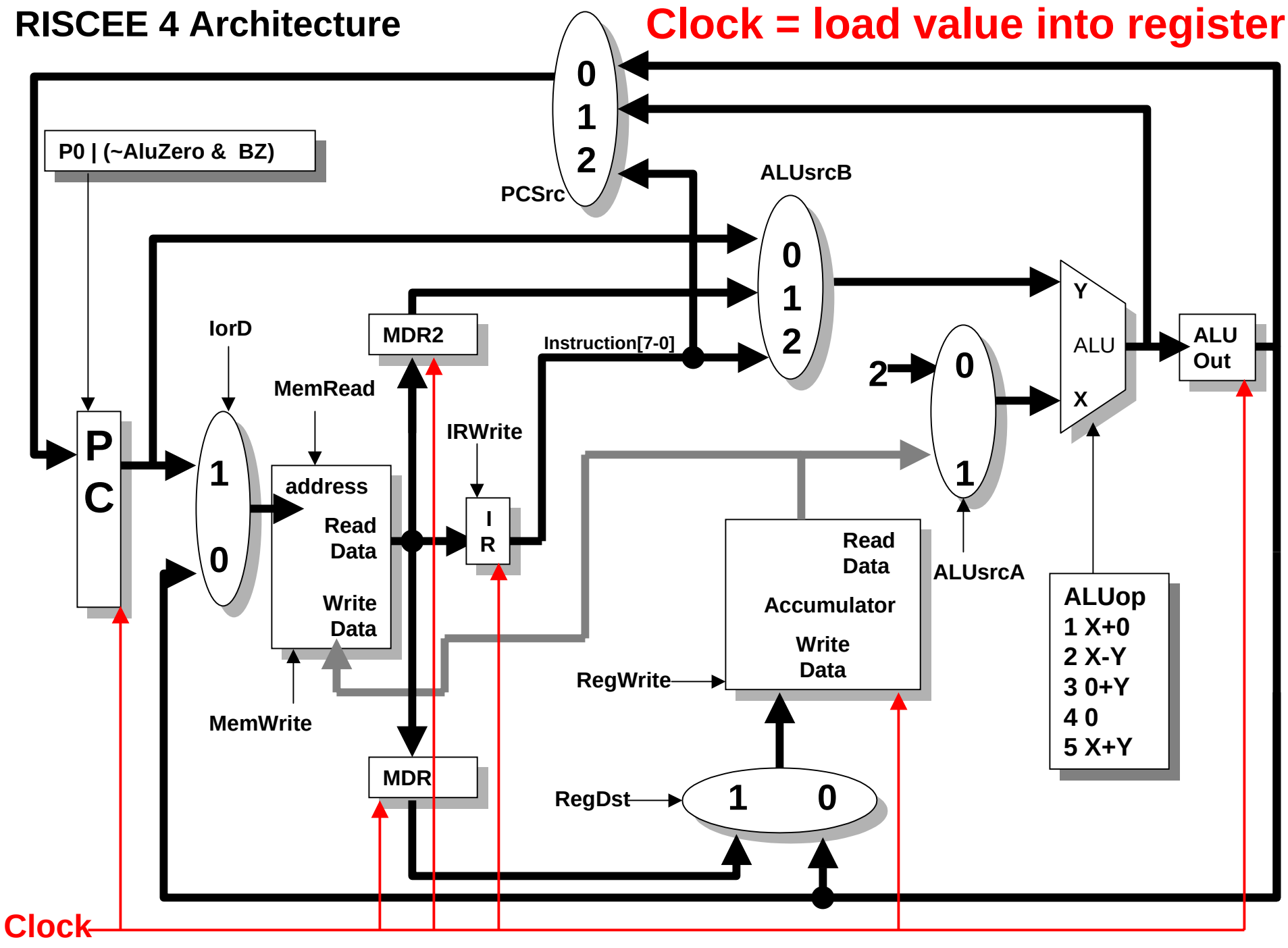
RegDst=X ALU=3 MemWrite=1 RegWrite=0 BZ=0 P0=0

Instruction: apc **Operation: A=pc+2;**



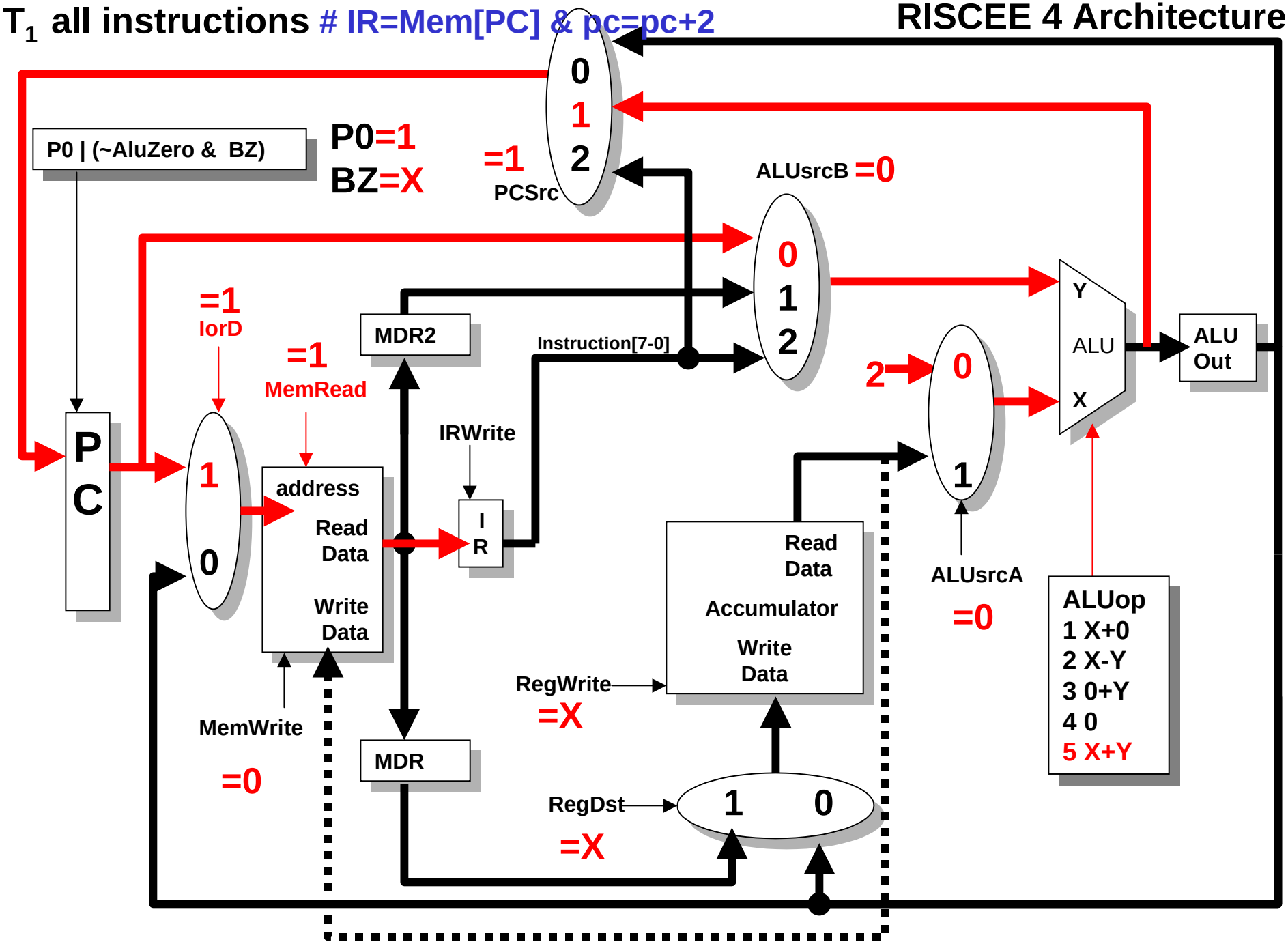
RISCEE 3 Architecture

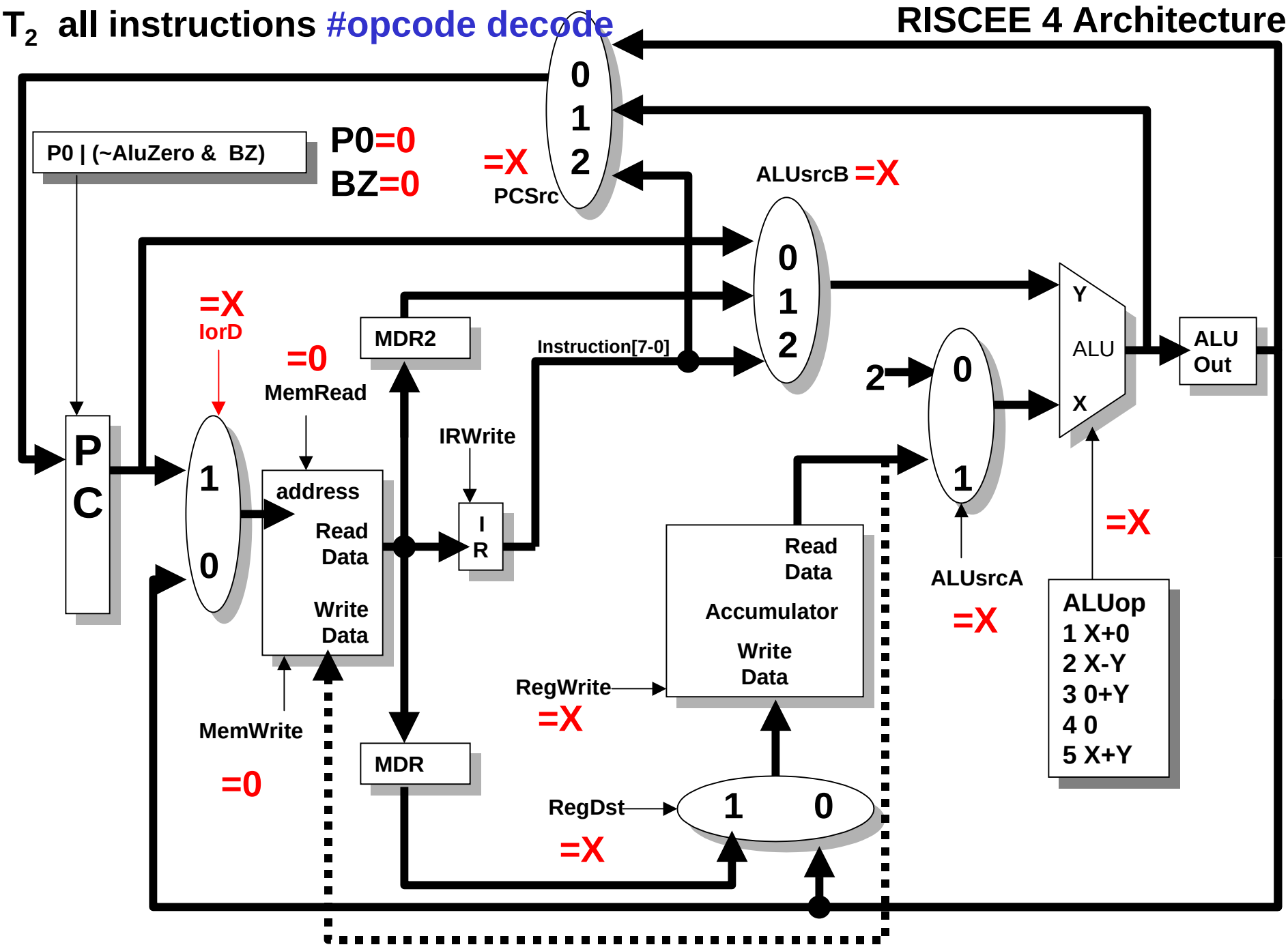
RegDst=X ALU=X MemWrite=1 RegWrite=0 BZ=0 P0=0

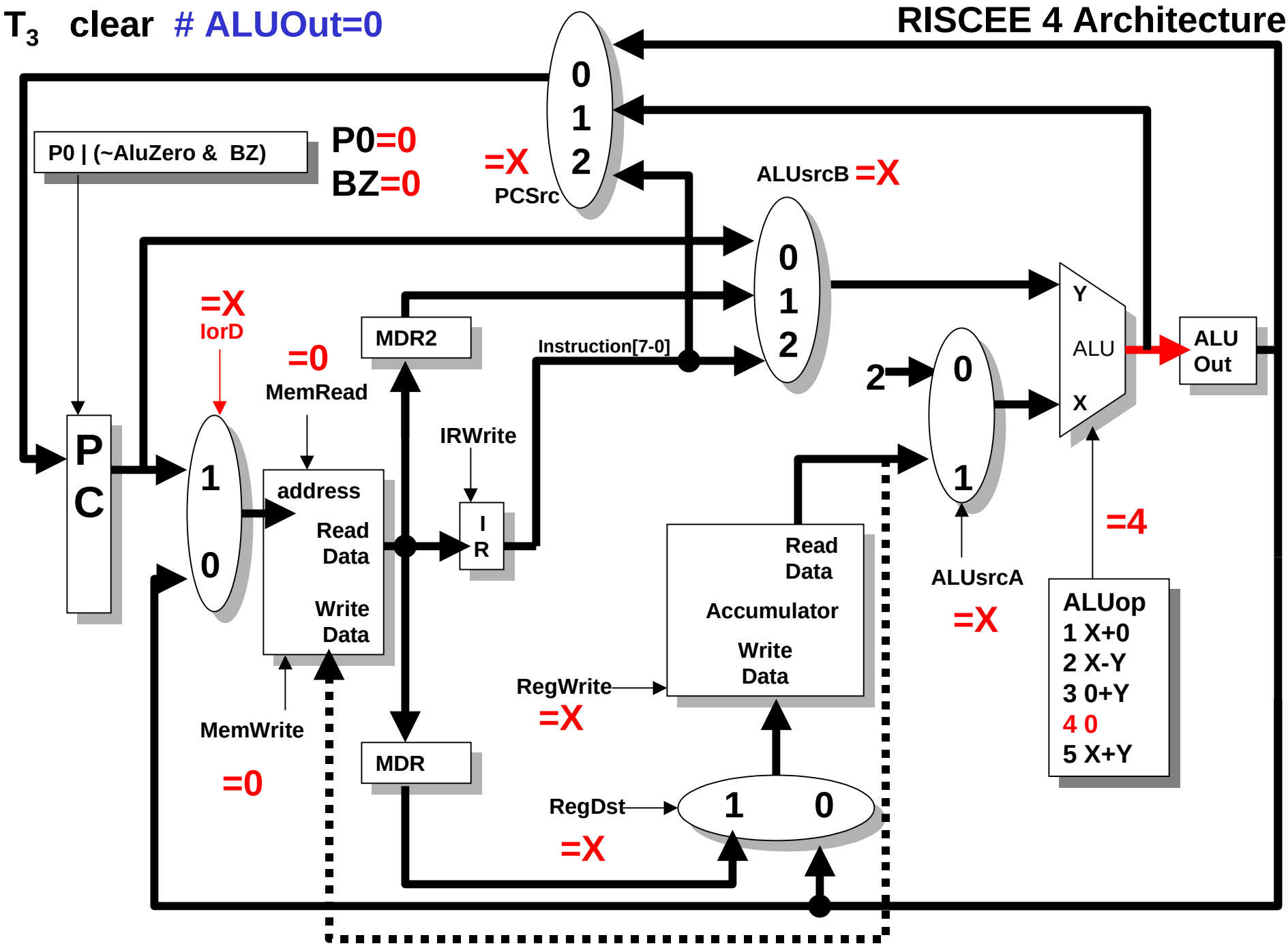

RISCEE 4 Architecture
Clock = load value into register
P0 | (~AluZero & BZ)
PCSrc
0
1
2
ALUsrcB
0
1
2
Y
ALU
X
ALU Out
IorD
MDR2
Instruction[7-0]
MemRead
IRWrite
2
0
1
P
C
1
0
address
Read Data
Write Data
I
R
Read Data
Accumulator
Write Data
ALUsrcA
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
MemWrite
MDR
RegDst
1
0
Clock

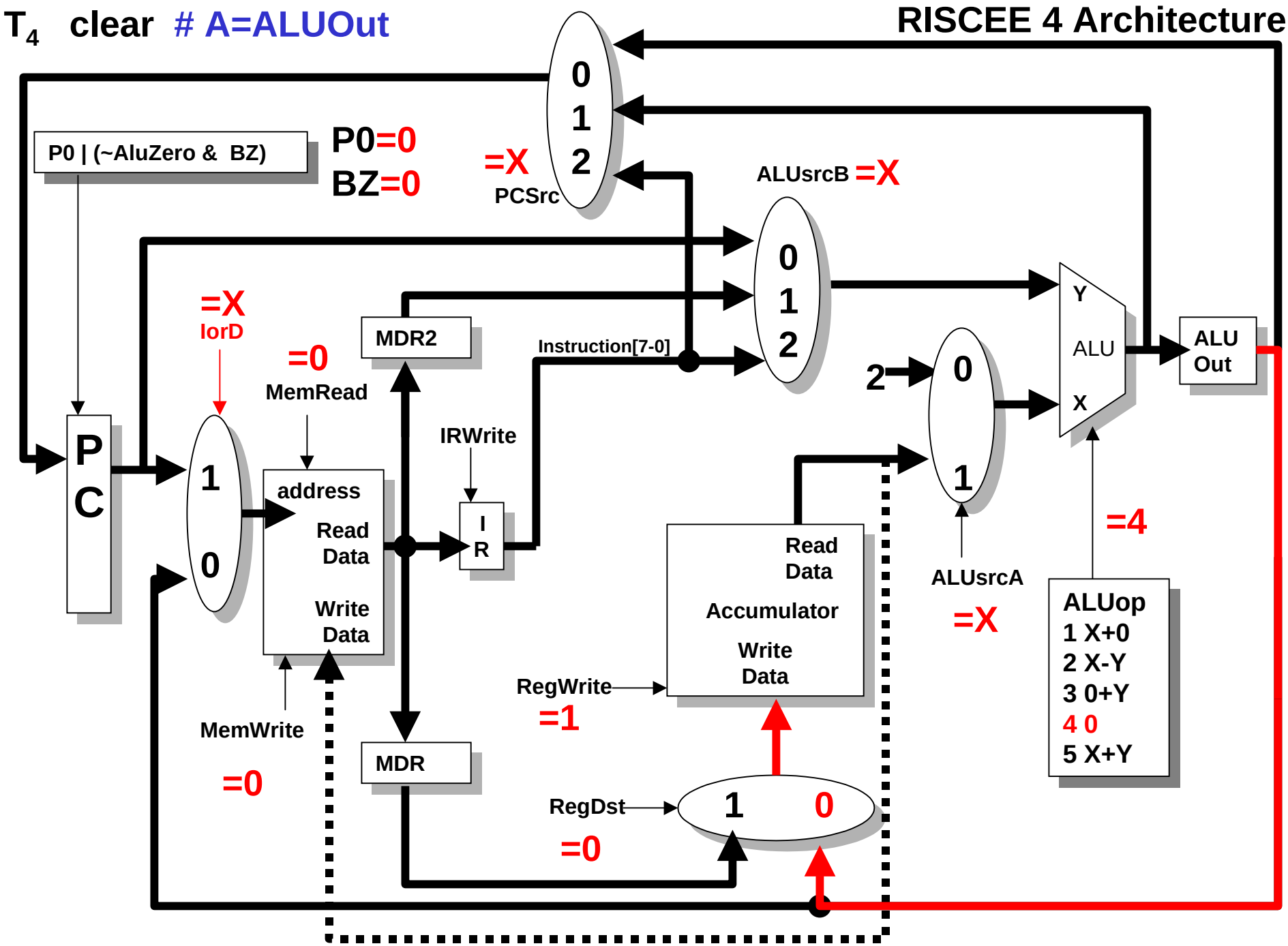
T_1 all instructions # IR=Mem[PC] & pc=pc+2

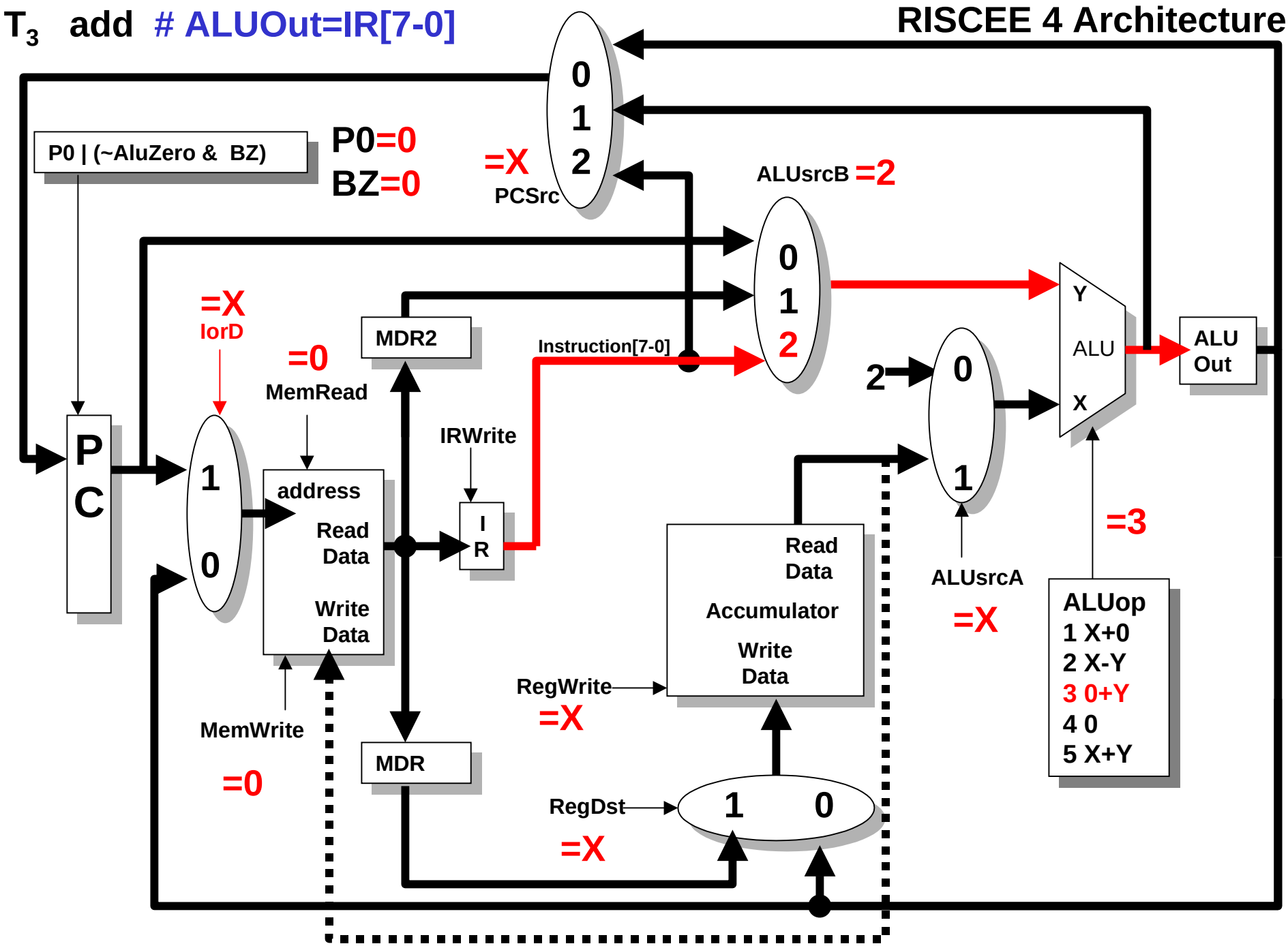
RISCEE 4 Architecture

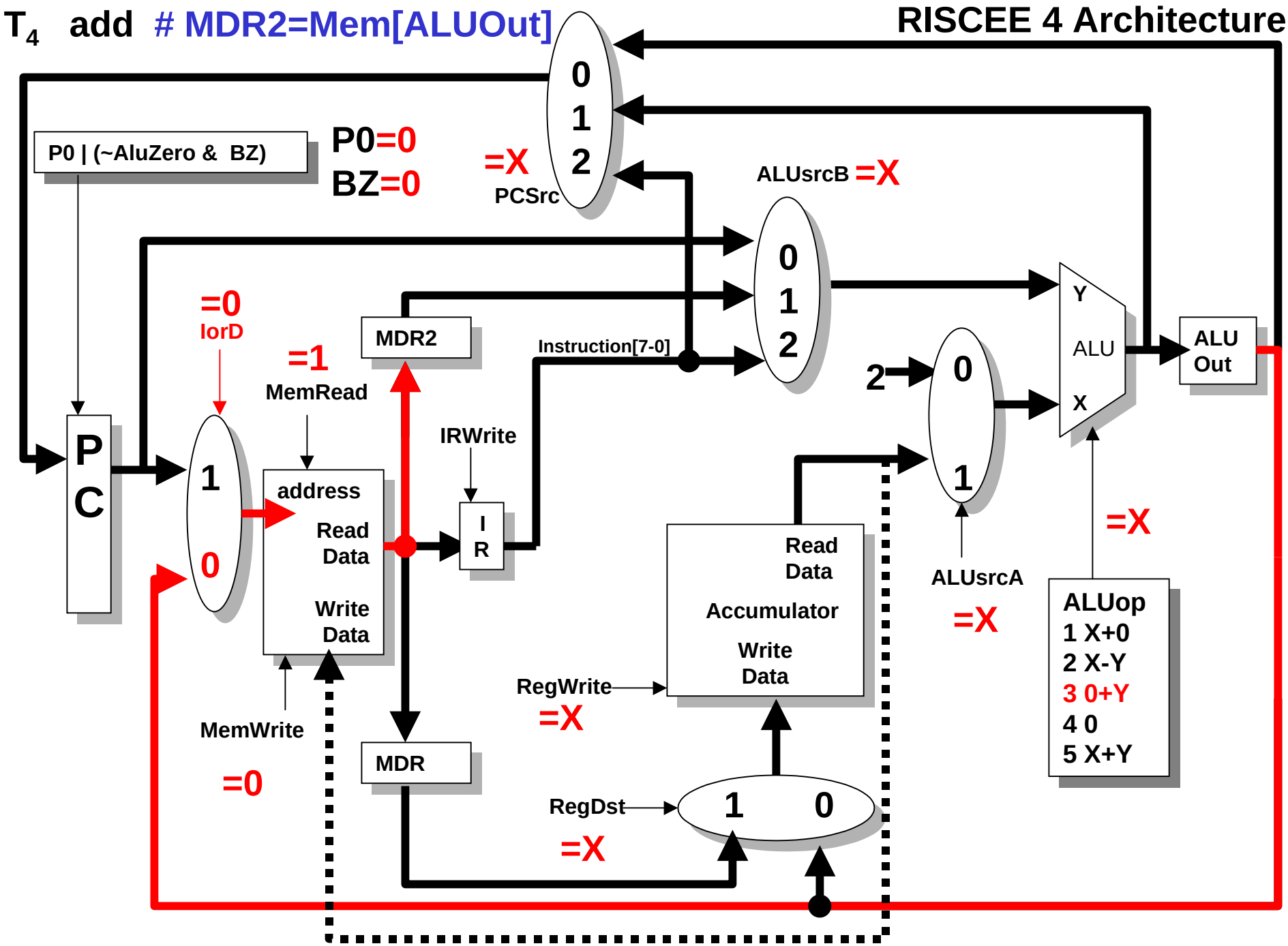


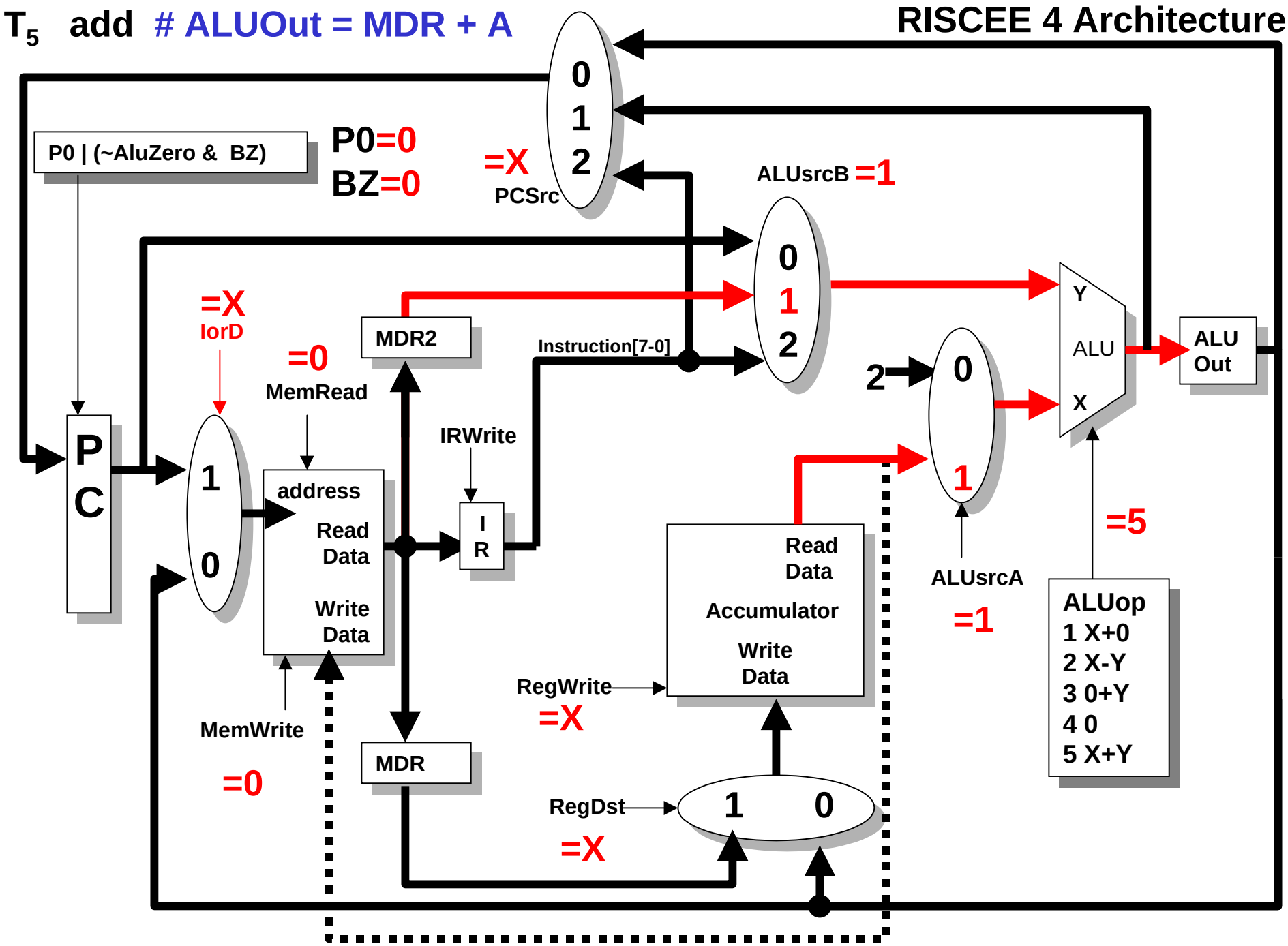

T2 all instructions #opcode decode
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =X
0
1
2
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
2
0
1
Y
ALU
X
ALU Out
P C
1
0
address
Read Data
Write Data
IRWrite
I R
Read Data
Accumulator
Write Data
ALUsrcA
=X
=X
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=X
MemWrite
=0
MDR
RegDst
1
0
=X

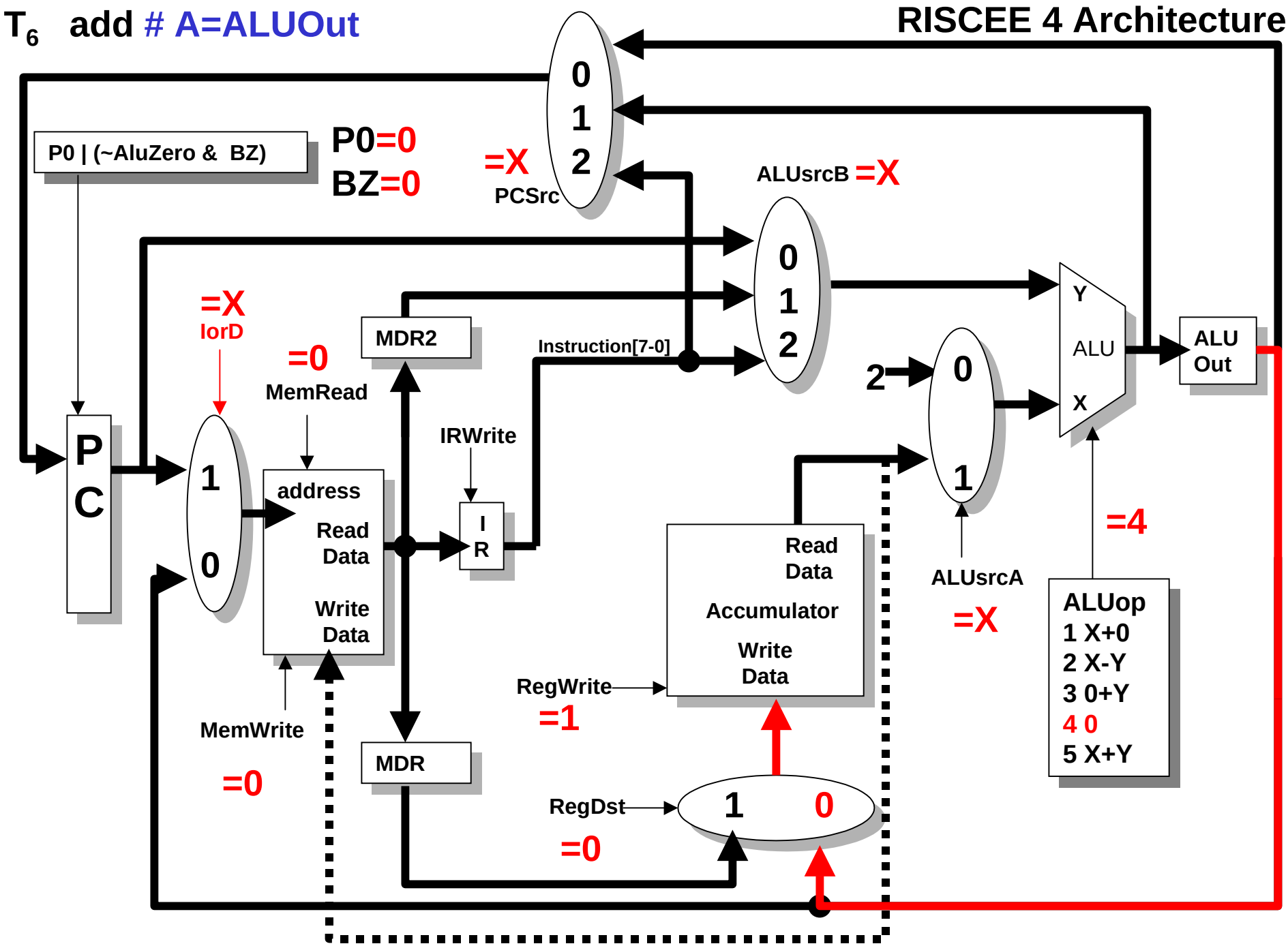

T_3 clear # ALUOut=0
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =X
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
IRWrite
P
C
1
0
address
Read
Data
Write
Data
I
R
MemWrite
=0
MDR
RegWrite
=X
Read
Data
Accumulator
Write
Data
RegDst
=X
1
0
2
0
1
ALUsrcA
=X
Y
ALU
X
ALU
Out
=4
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y

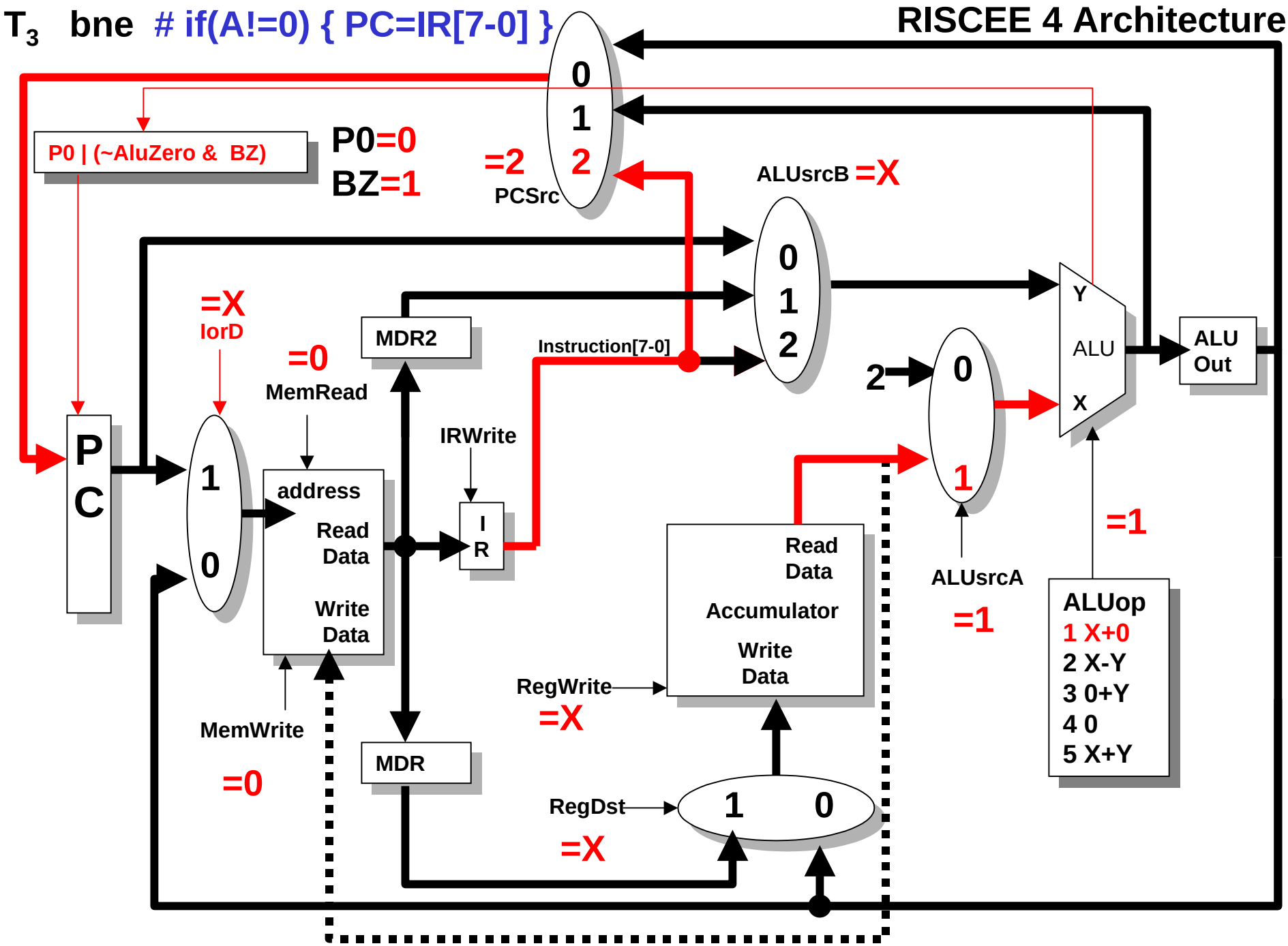

T4 clear # A=ALUOut
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =X
0
1
2
Y
ALU
X
ALU Out
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
2
0
1
P
C
1
0
address
Read Data
Write Data
IRWrite
I
R
Read Data
Accumulator
Write Data
ALUsrcA
=X
=4
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=1
MemWrite
=0
MDR
RegDst
1
0
=0


T3 add # ALUOut=IR[7-0]
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =2
0
1
2
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
IRWrite
P
C
1
0
address
Read
Data
Write
Data
I
R
MemWrite
=0
MDR
RegWrite
=X
Read
Data
Accumulator
Write
Data
RegDst
=X
1
0
2
0
1
ALUsrcA
=X
Y
ALU
X
ALU
Out
=3
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y


T4 add # MDR2=Mem[ALUOut]
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =X
0
1
2
Y
ALU
X
ALU Out
=0
IorD
=1
MemRead
MDR2
Instruction[7-0]
2
0
1
P
C
1
0
address
Read Data
Write Data
IRWrite
I
R
Read Data
Accumulator
Write Data
ALUsrcA
=X
=X
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=X
MemWrite
=0
MDR
RegDst
=X
1
0


T5 add # ALUOut = MDR + A
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =1
0
1
2
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
2
0
1
Y
ALU
X
ALU Out
P C
1
0
address
Read Data
Write Data
IRWrite
I R
Read Data
Accumulator
Write Data
ALUsrcA
=1
=5
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=X
MemWrite
=0
MDR
RegDst
=X
1
0


T6 add # A=ALUOut
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=0
=X
PCSrc
0
1
2
ALUsrcB =X
0
1
2
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
2
0
1
Y
ALU
X
ALU Out
P
C
1
0
address
Read Data
Write Data
IRWrite
I
R
Read Data
Accumulator
Write Data
ALUsrcA
=X
=4
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=1
MemWrite
=0
MDR
RegDst
1
0
=0


T3 bne # if(A!=0) { PC=IR[7-0] }
RISCEE 4 Architecture
P0 | (~AluZero & BZ)
P0=0
BZ=1
=2
PCSrc
0
1
2
ALUsrcB =X
0
1
2
=X
IorD
=0
MemRead
MDR2
Instruction[7-0]
2
0
1
Y
ALU
X
ALU Out
P C
1
0
address
Read Data
Write Data
IRWrite
I R
Read Data
Accumulator
Write Data
ALUsrcA
=1
=1
ALUop
1 X+0
2 X-Y
3 0+Y
4 0
5 X+Y
RegWrite
=X
MemWrite
=0
MDR
RegDst
=X
1
0