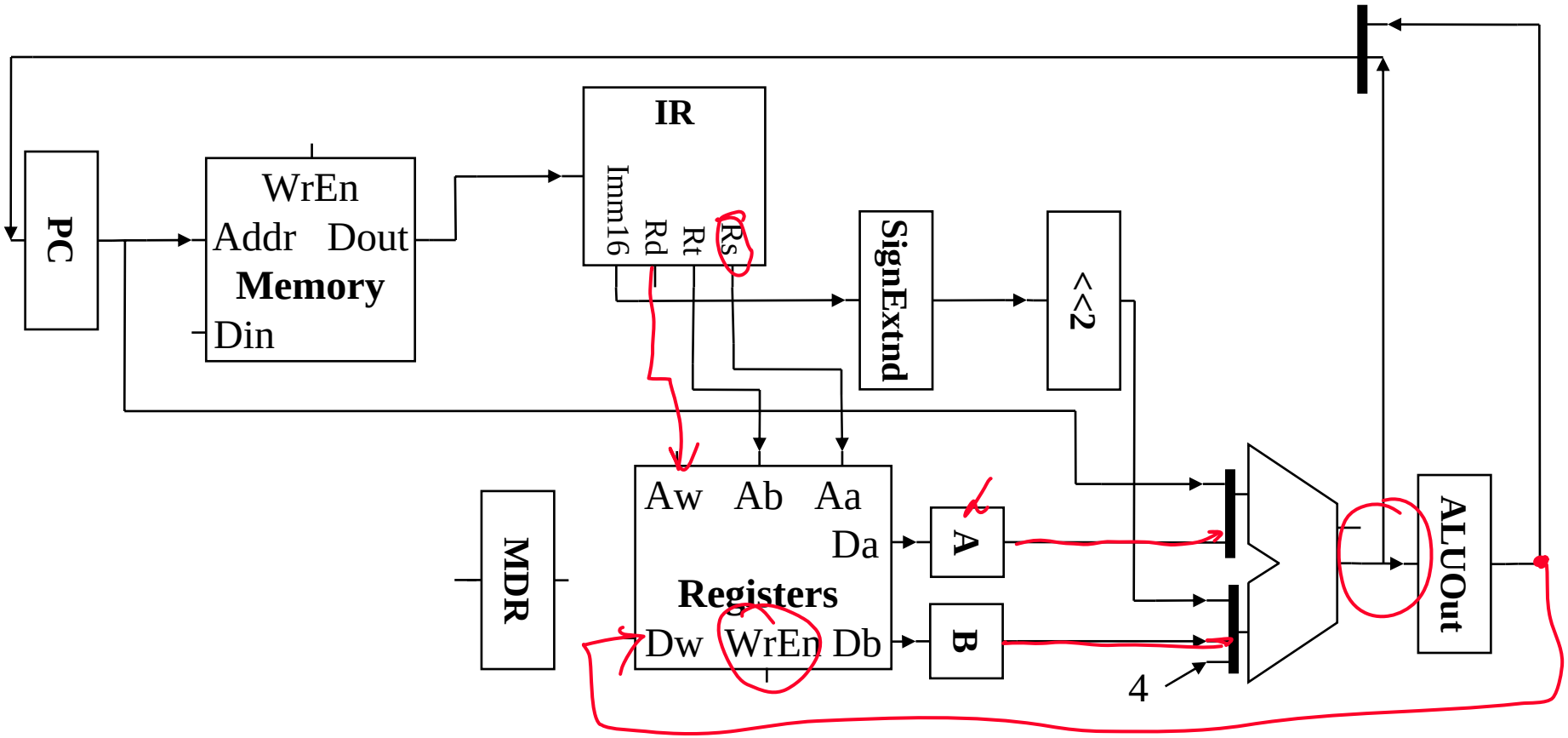
Branch, R-Type Datapath

(4)

(3) ALUout = A op B , Reg[Rd] = ALUOUT



Cycle 3-4 (Store)

Cycle 3: compute address from operand A and IR[15-0], put into ALUout

RTL: ALUout ← A + SE(imm16)

sw $t0, 16[$t1]

A

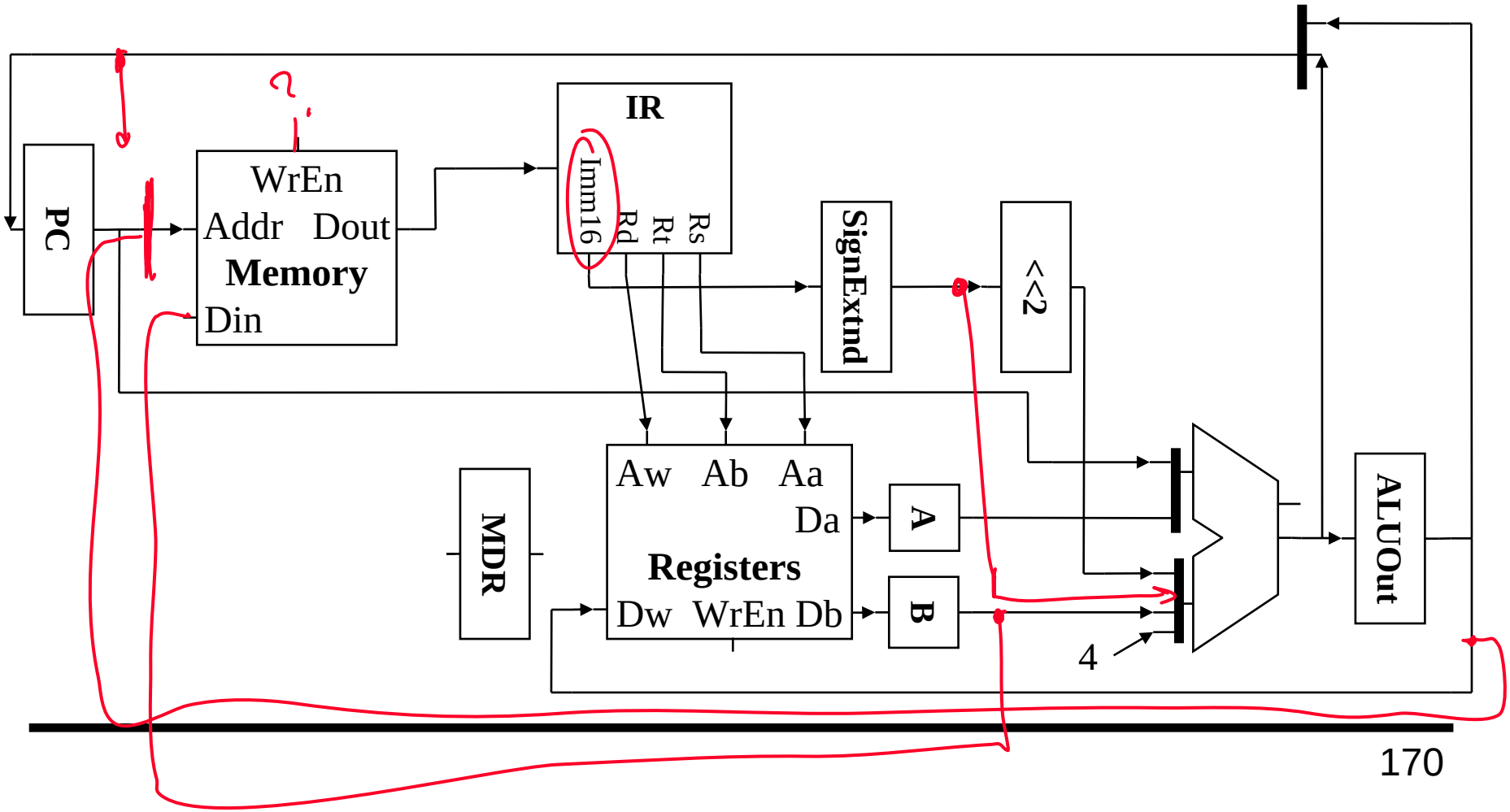
imm16

Cycle 4: Store value from operand B to address specified in ALUout

RTL: Mem[ALUout] ← B

Branch, R-Type, Store Datapath

3 ALUout = A + SE(imm16) 4 Mem[ALUout] = B



Cycle 3-5 (Load)

lw $t0, 16[$t1]

Cycle 3: compute address from operand A and IR[15-0], put into ALUout

RTL: ALUout = A + SE(imm16)

Cycle 4: Load value from address specified in ALUout to MDR

RTL: MDR = Mem[ALUout]

Cycle 5: Write value from MDR to destination register

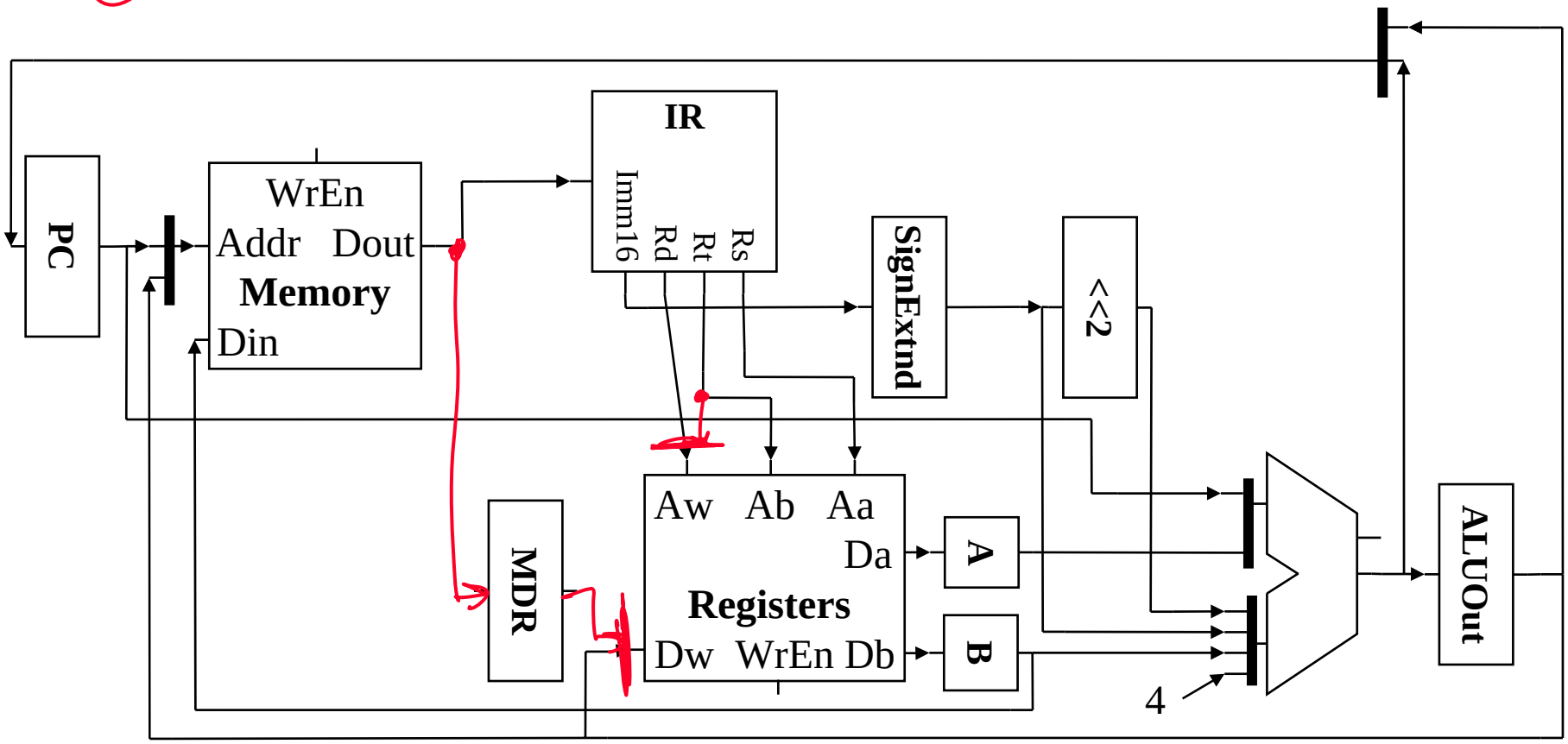
RTL: Reg[Rt] = MDR

Multicycle Processor Datapath

③ ALUout = A + SE(i16)

④ MDR = Mem(ALUout)

⑤ Reg[Rt] = MDR



Multicycle Processor Control

Need to control data path to perform required operations

Multiple cycles w/different control values each cycle, so control is an FSM.

```
Cycle 1:          IR = Mem[PC]; PC = PC + 4
Cycle 2:          A = Reg[RS]; B = Reg[Rt];
                  ALUOut = PC + (Sign-extend(Imm16)<<2)
Cycle 3 Branch:   Zero = (A-B); If (zero) PC = ALUout
Cycle 3 R-Type:   ALUout = A op B
Cycle 4 R-Type:   Reg[Rd] = ALUout
Cycle 3 Store:    ALUout = A + sign-extend(Imm16)
Cycle 4 Store:    Mem[ALUout] = B
Cycle 3 Load:     ALUout = A + sign-extend(Imm16)
Cycle 4 Load:     MDR = Mem[ALUout]
Cycle 5 Load:     Reg[Rt] = MDR
```

Control FSM

Opcodes: `LW 35, SW 43, BEQ 4, add/sub 0`