

Disseny del camí de dades multicycle per l'arquitectura MIPS

MIPS multicicle

- Introducció
- Descripció del camí de dades
 - Anàlisis del repertori d'instruccions a implementar
 - Requisits del camí de dades
 - Selecció de components
 - Construcció del camí de dades
 - Anàlisi de la implementació de cada instrucció
 - Punts de control

MIPS multicicle

- Implementació Multicicle

- Divisió de l'execució de les instruccions en varies etapes.

- S'executa una etapa de la instrucció a cada cicle de rellotge

- Avantatges

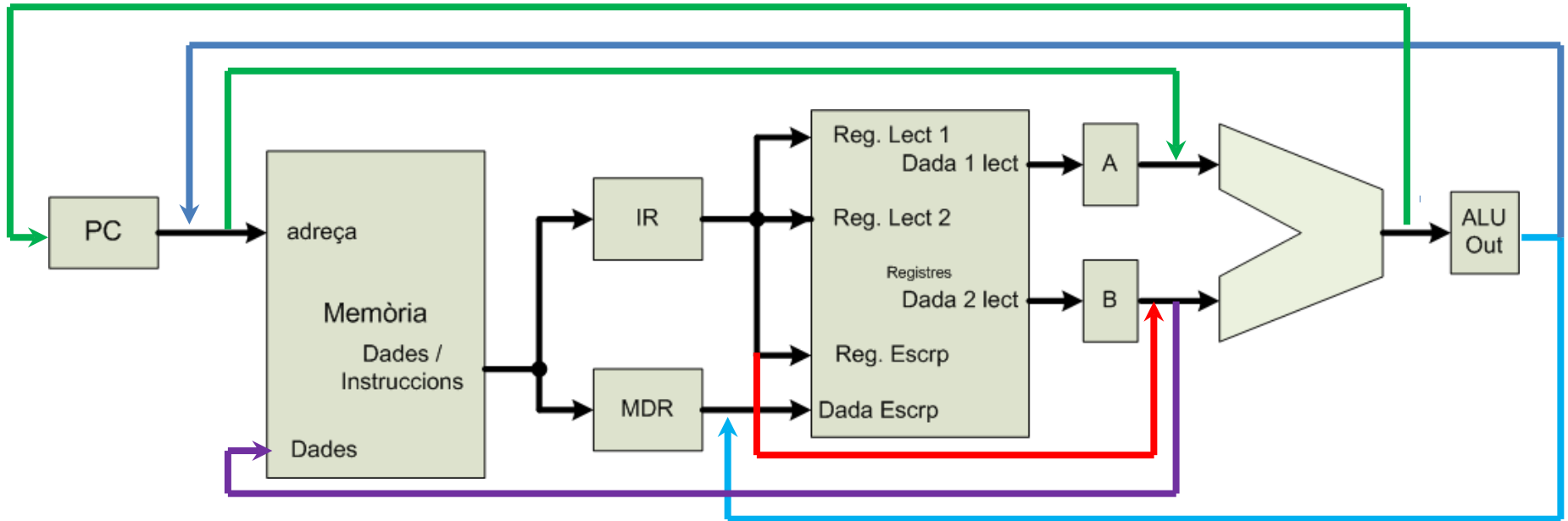
- Reutilització de les unitats funcionals

- Una unitat pot utilitzar-se més d'un cop per instrucció sempre que es faci en diferents cicles de rellotge

- CPI (cicles per instrucció) variable

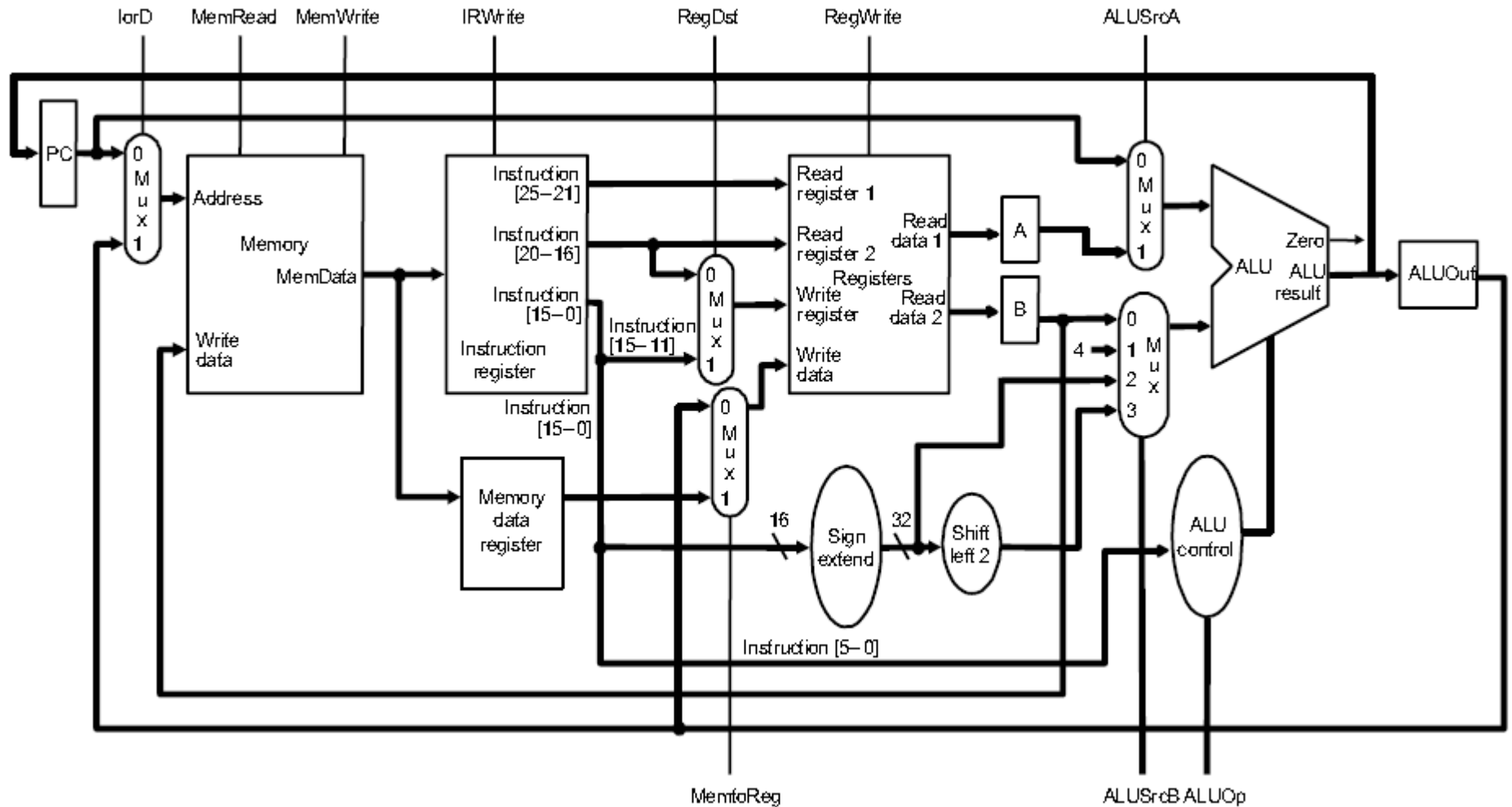
- Instruccions senzilles → pocs cicles per executar-se
 - Instruccions complexes → més cicles de rellotge

MIPS multicicle: Components bàsics



1. S'ha d'incrementar el PC
2. S'ha de poder guardar el resultat de l'ALU als registres. (add)
3. Ús de registres temporals a les sortides de les unitats funcionals principals.
La sortida de l'ALU, pot ser una adreça calculada (lw)
4. S'han de poder sumar desplaçaments i donar valors immediats (addi)
 - Actualitzen el seu contingut a cada instrucció (IR i PC)
5. S'han de poder guardar valors a la memòria (sw)
 - Actualitzen el seu contingut a cada cicle de rellotge (MDR, A, B, ALUOut)

MIPS multicicle: Components bàsics



- Compartir les unitats funcionals → Multiplexors
- Execució en múltiples cicles → Cada cicle requereix un conjunt de senyals de control
- Senyals de control

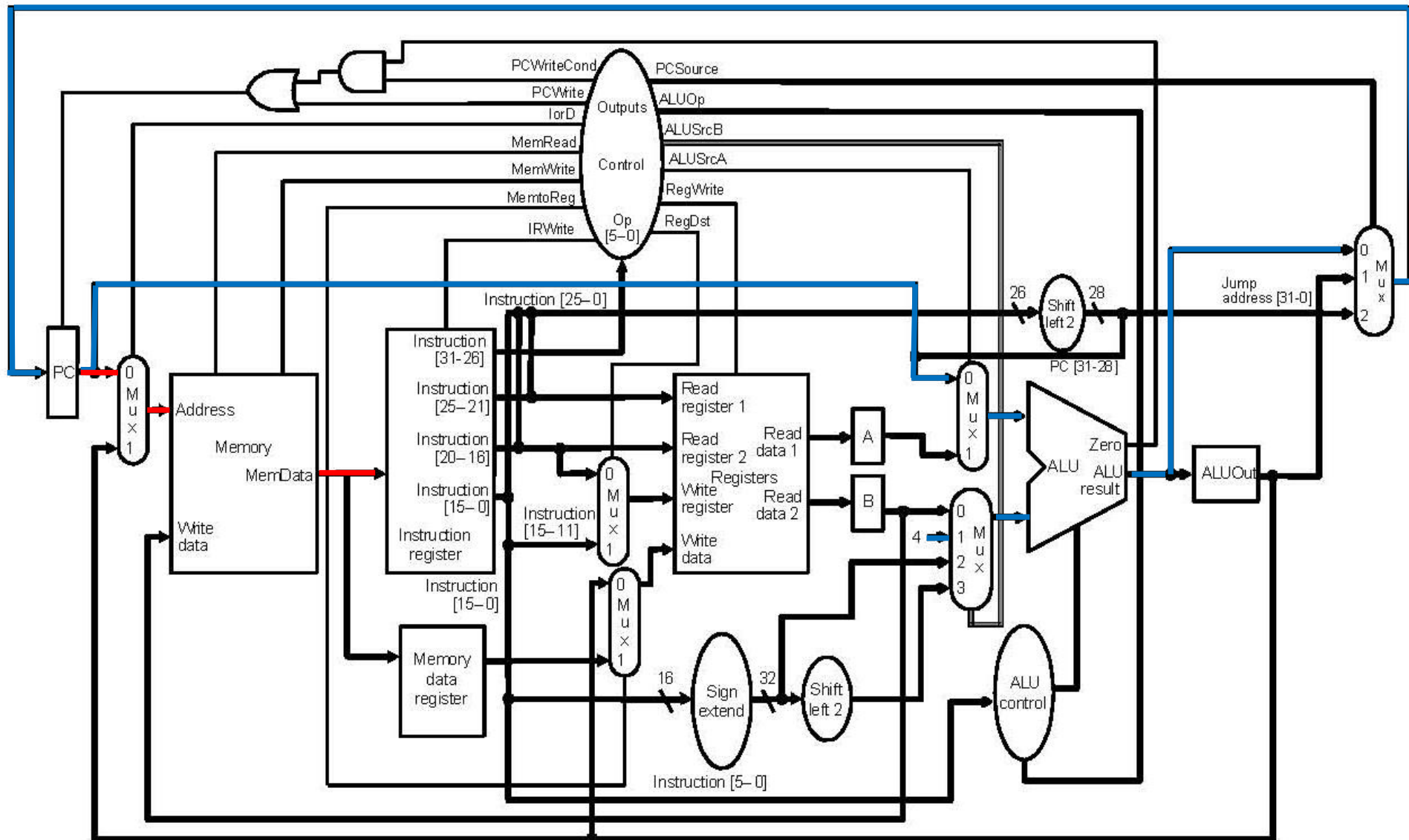
Etales d'execució de les instruccions

- Quines operacions s'han de realitzar en cada cicle de rellotge?
 - Objectiu: Equilibrar la càrrega computacional de cada etapa → Minimitzar el temps de cicle
 - En cadascuna de les etapes s'ha de realitzar alguna de les següents operacions:
 - Un accés a un registre
 - Actualització a cada cicle (registres temporals: MDR, A, B, ALUOut)
 - Actualització segons un senyal d'escriptura (PC, IR)
 - Un accés a memòria
 - Una operació d'ALU
 - Com es pot determinar el temps de cicle mínim?
 - Totes les operacions d'un sol cicle s'executen en paral·lel
 - Les etapes de la instrucció s'executen en sèrie

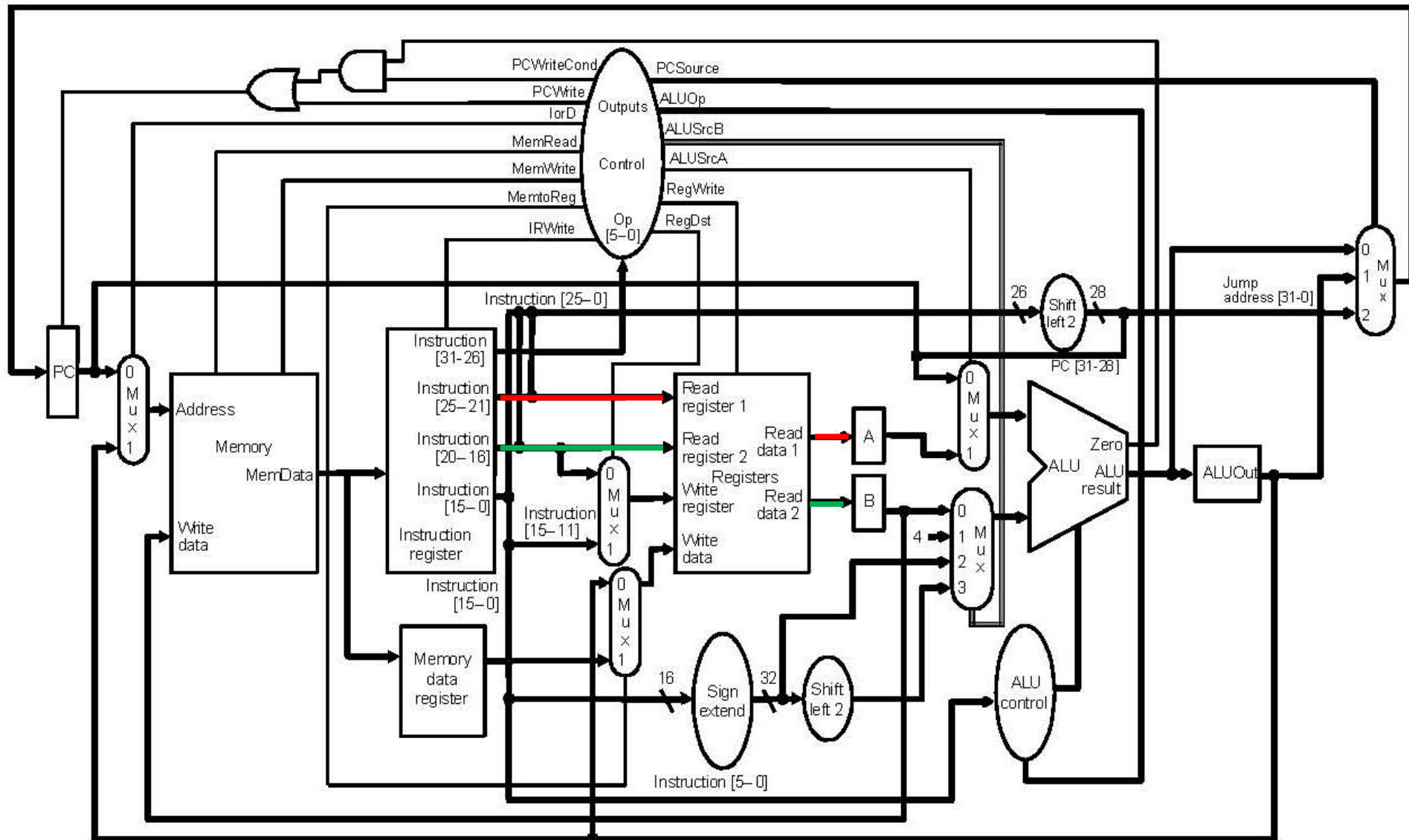
MIPS multicicle: Etapes d'execució de les instruccions

- Etapa 1: Cerca del codi d'operació (fetch)
 - $IR = \text{Memory}[PC]$ i $PC = PC + 4$
- Etapa 2: Descodificació i accés als operants
 - $A = \text{Reg}[rs]$, $B = \text{Reg}[rt]$ i $ALUOut = PC + \text{extensió-signe}(IR[15-0]) \ll 2$
- Etapa 3: Execució, càlcul d'adreces o finalització del salt
 - **Instrucció tipus R (and, or, add, sub, slt)**
 - $ALUOut = A \text{ op } B$
 - **Referència a memòria (lw/sw)**
 - $ALUOut = A + \text{extensió} - \text{signe}(IR[15-0])$
 - **Salt (beq)** (Salt condicional)
 - if (A == B) $PC = ALUOut$
 - **Bifurcació (j)** (Salt)
 - $PC = PC[31-28] \ || \ IR[25-0] \ll 2$
- Etapa 4: Accés a memòria / fi d'execució d'instrucció del tipus R
 - **Referència a memòria**
 - $MDR = \text{Memory}[ALUOut]$ o $\text{Memory}[ALUOut] = B$
 - **Fi d'execució instrucció tipus R**
 - $\text{Reg}[rd] = ALUOut$
- Etapa 5: Fi de lectura a memòria
 - $\text{Reg}[rt] = MDR$

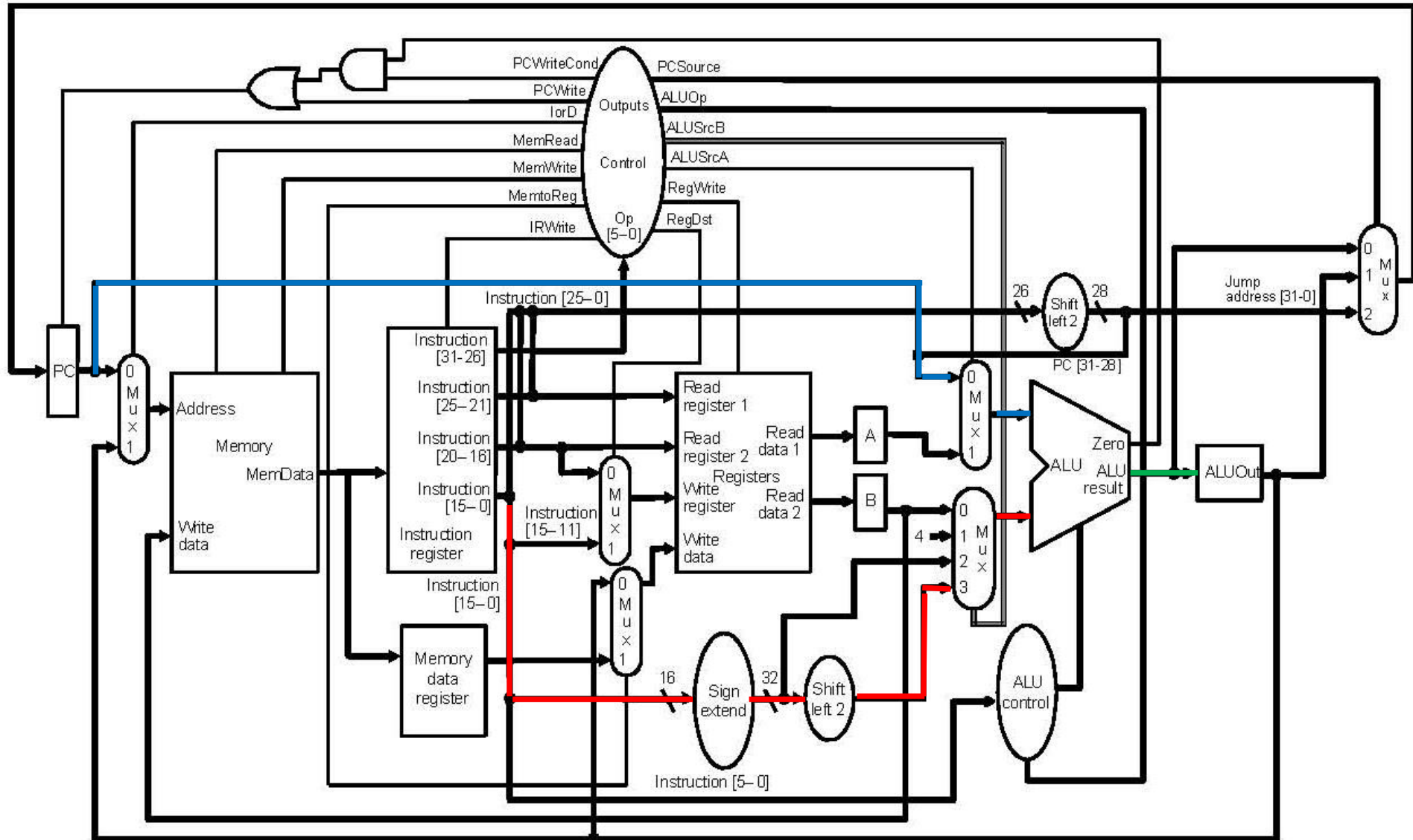
MIPS multicycle: Etapa 1; $IR = \text{Memory}[PC]$ i $PC = PC + 4$



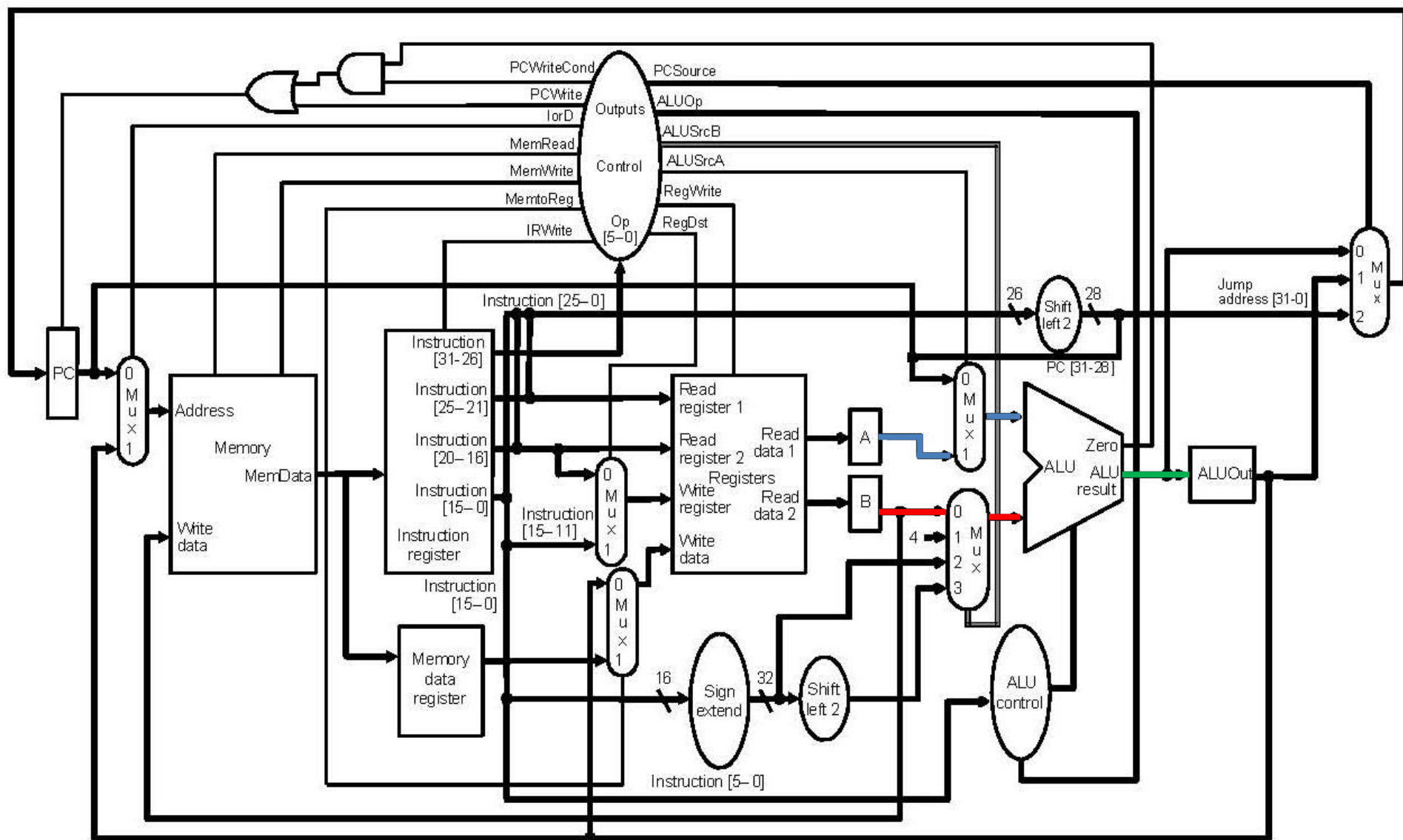
MIPS multicycle: Etapa 2 ($A = \text{Reg}[rs]$, $B = \text{Reg}[rt]$)



MIPS multicicle: Etapa 2 $(ALU_{out} = PC + \text{extensió-signe}(IR[15-0]) \ll 2)$

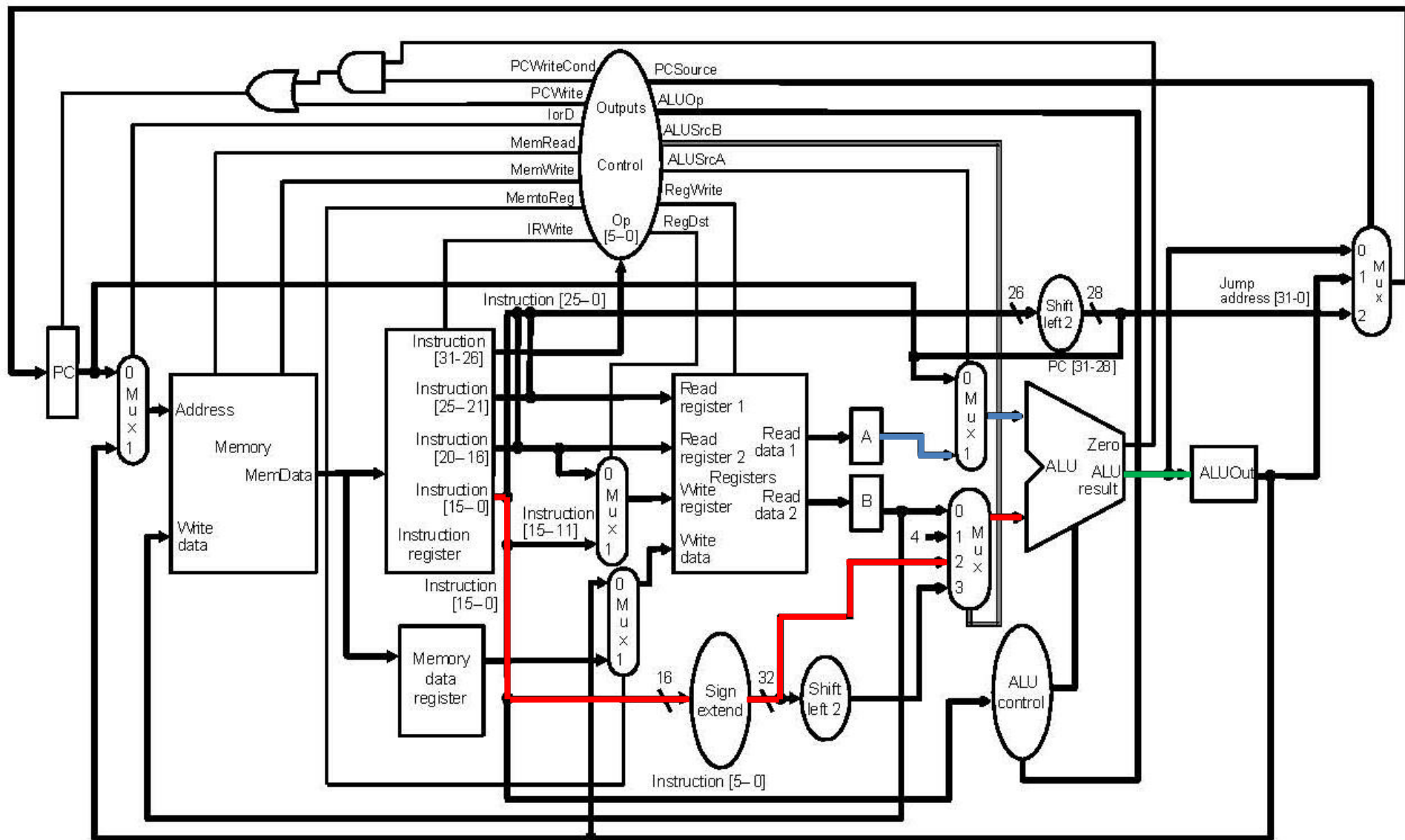


MIPS multicicle: Etapa 3 (Tipus R) -> $ALUOut = A \text{ Op } B$

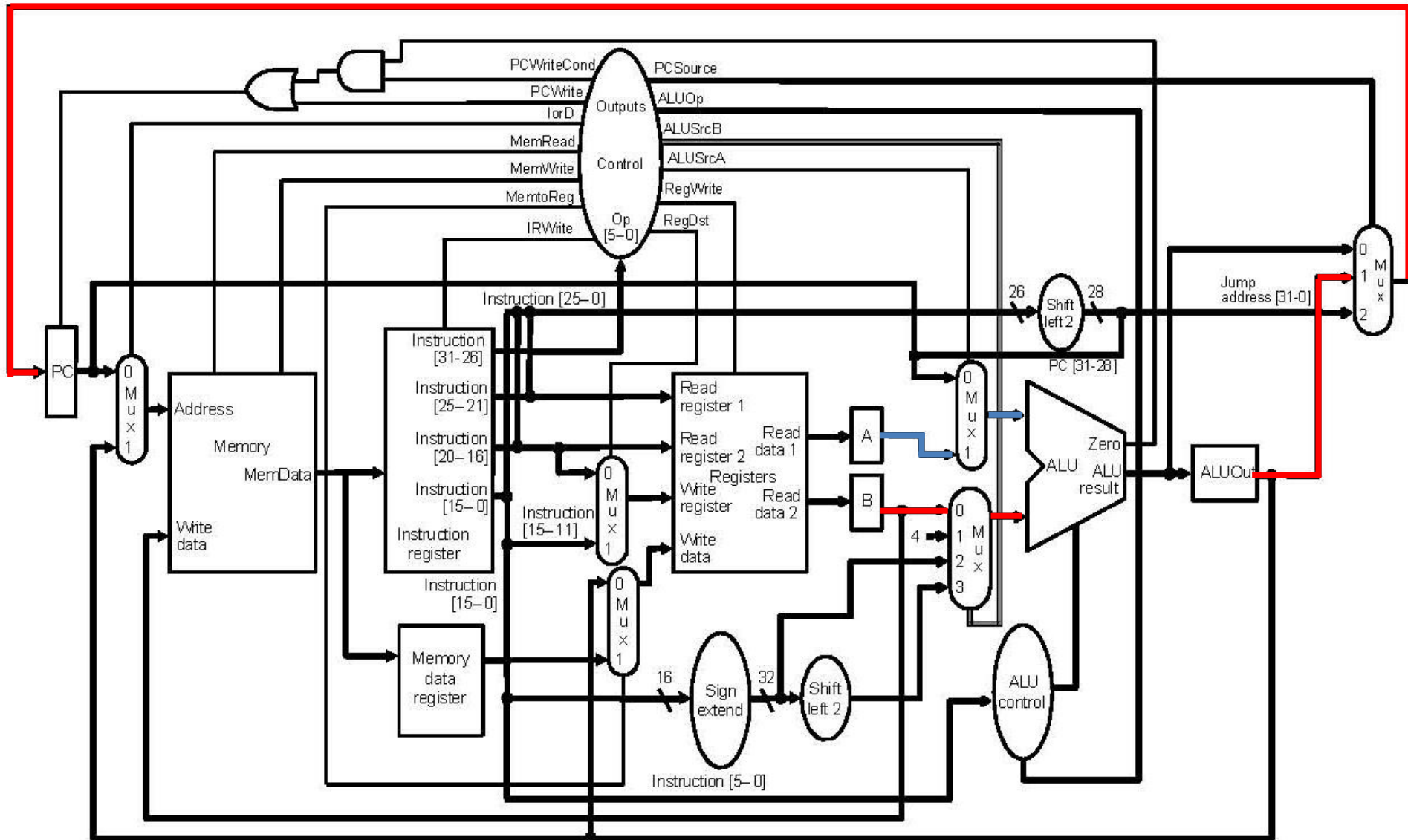


MIPS multicicle: Etapa 3 (Referència a memòria)

$$ALUout = A + \text{extensió-signe}(IR[15-0])$$

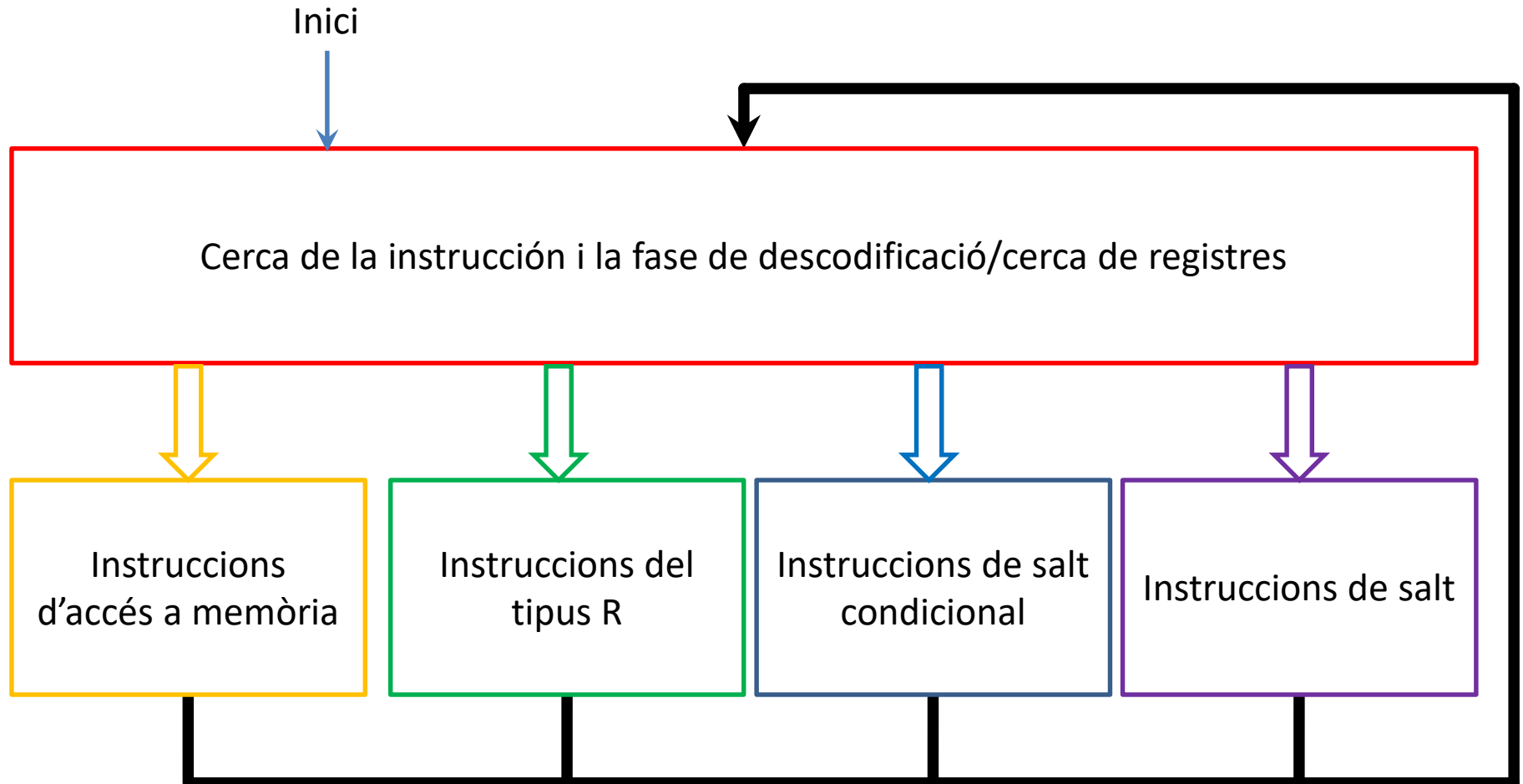


MIPS multicycle: Etapa 3 (Salt condicional)



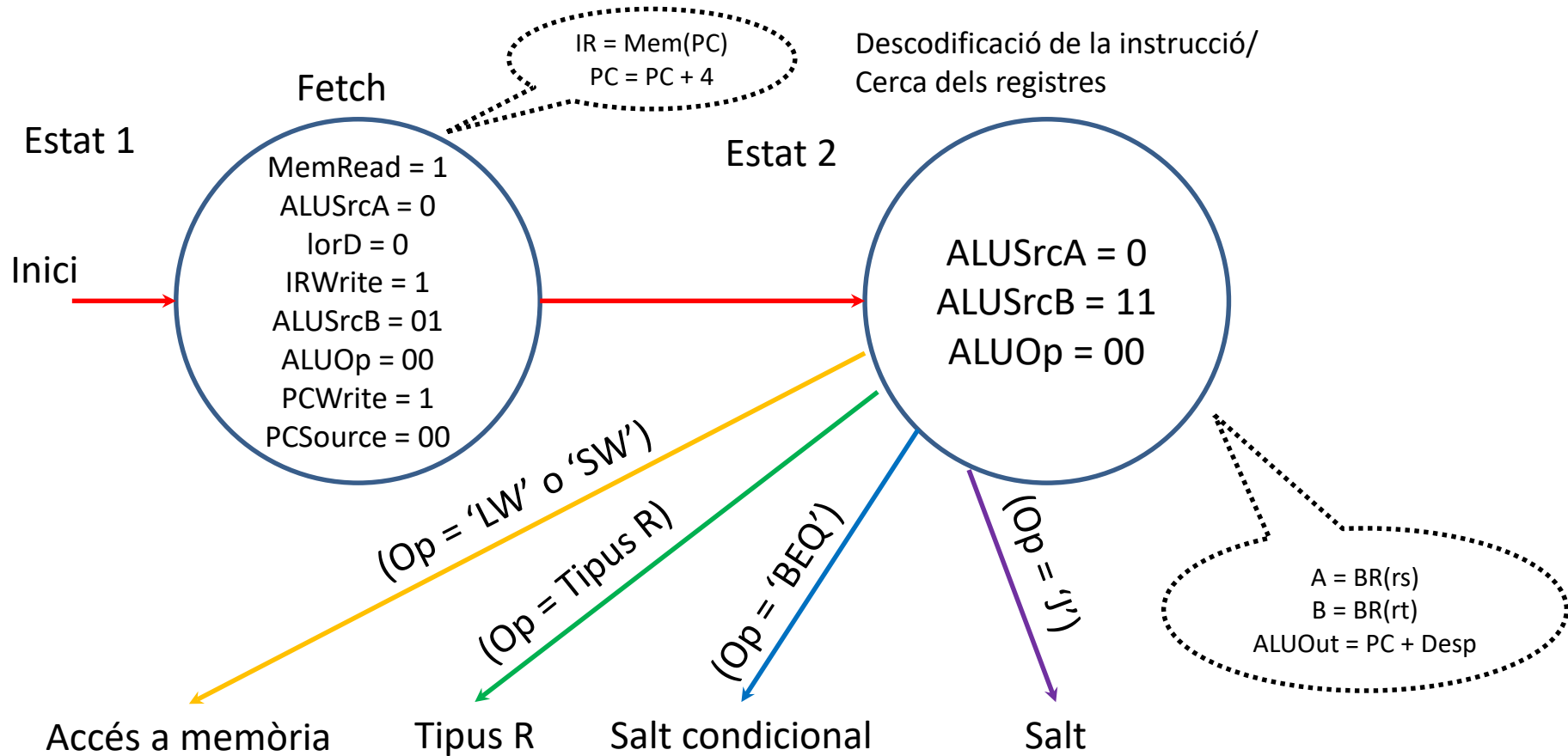
MIPS multicicle: Controlador

- Implementació amb una màquina d'estats finits

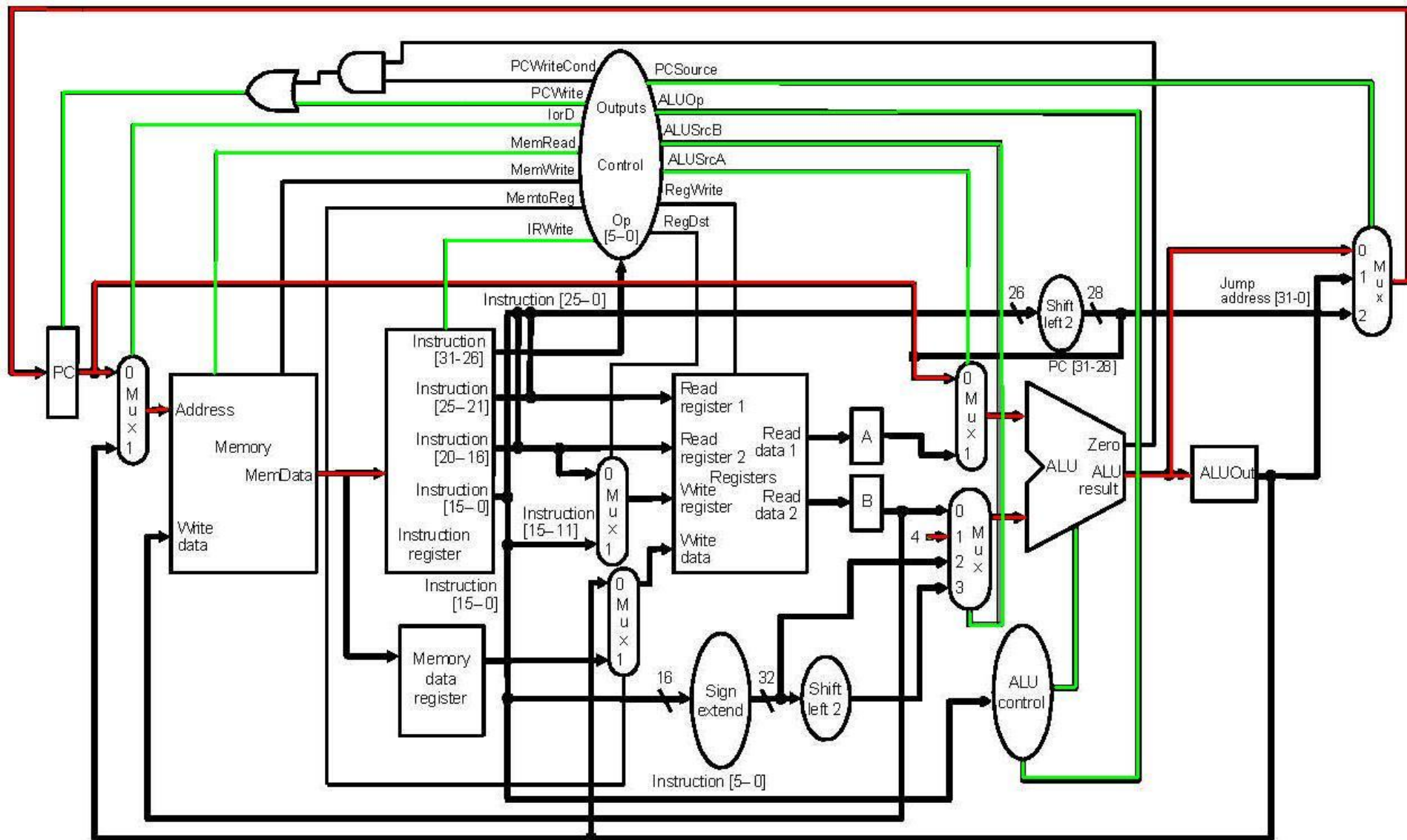


MIPS multicicle: Controlador

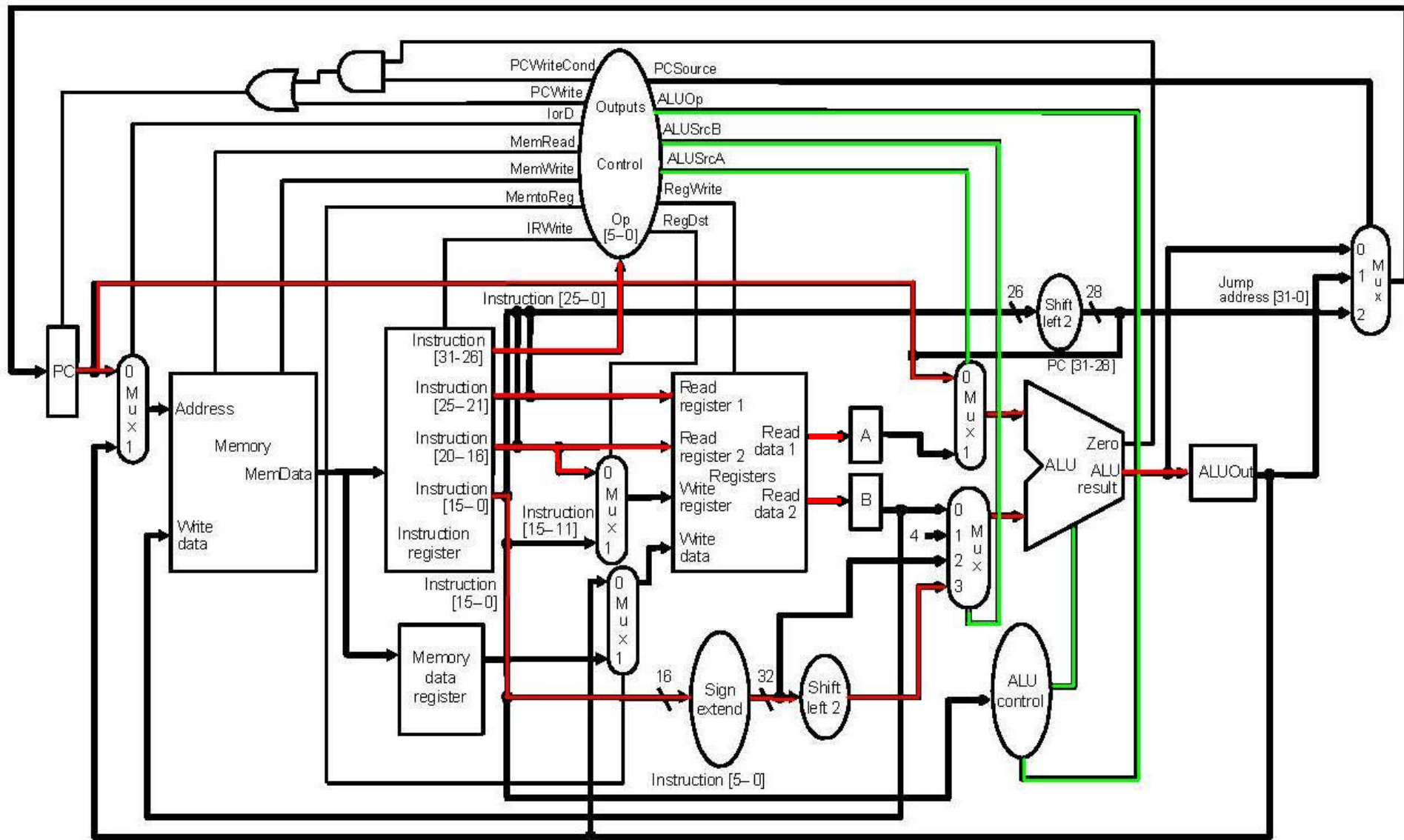
- Cerca de la instrucció i la fase de descodificació/cerca de registres



MIPS multicycle: Fetch



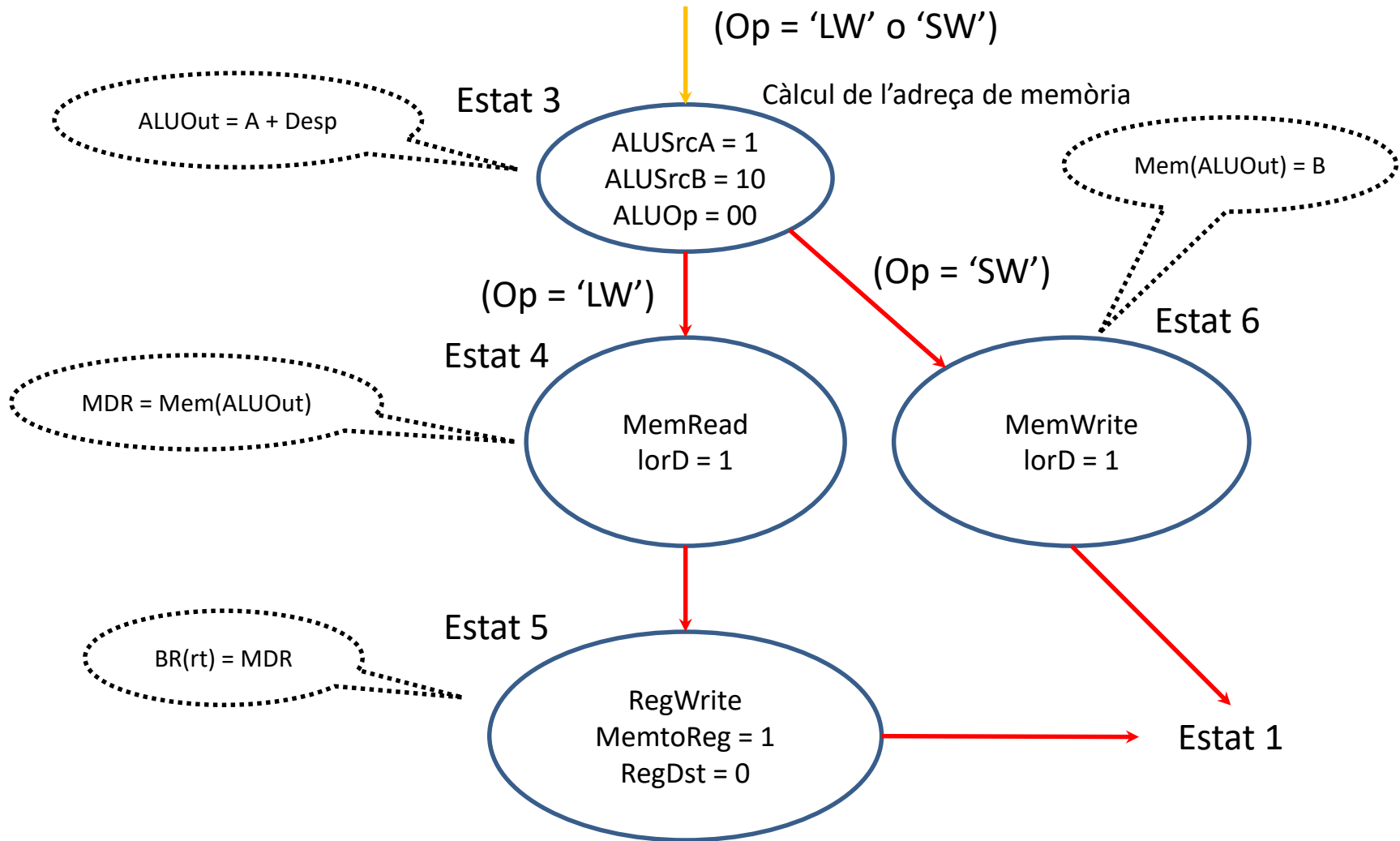
MIPS multicicle: Descodificació i càlcul adreça



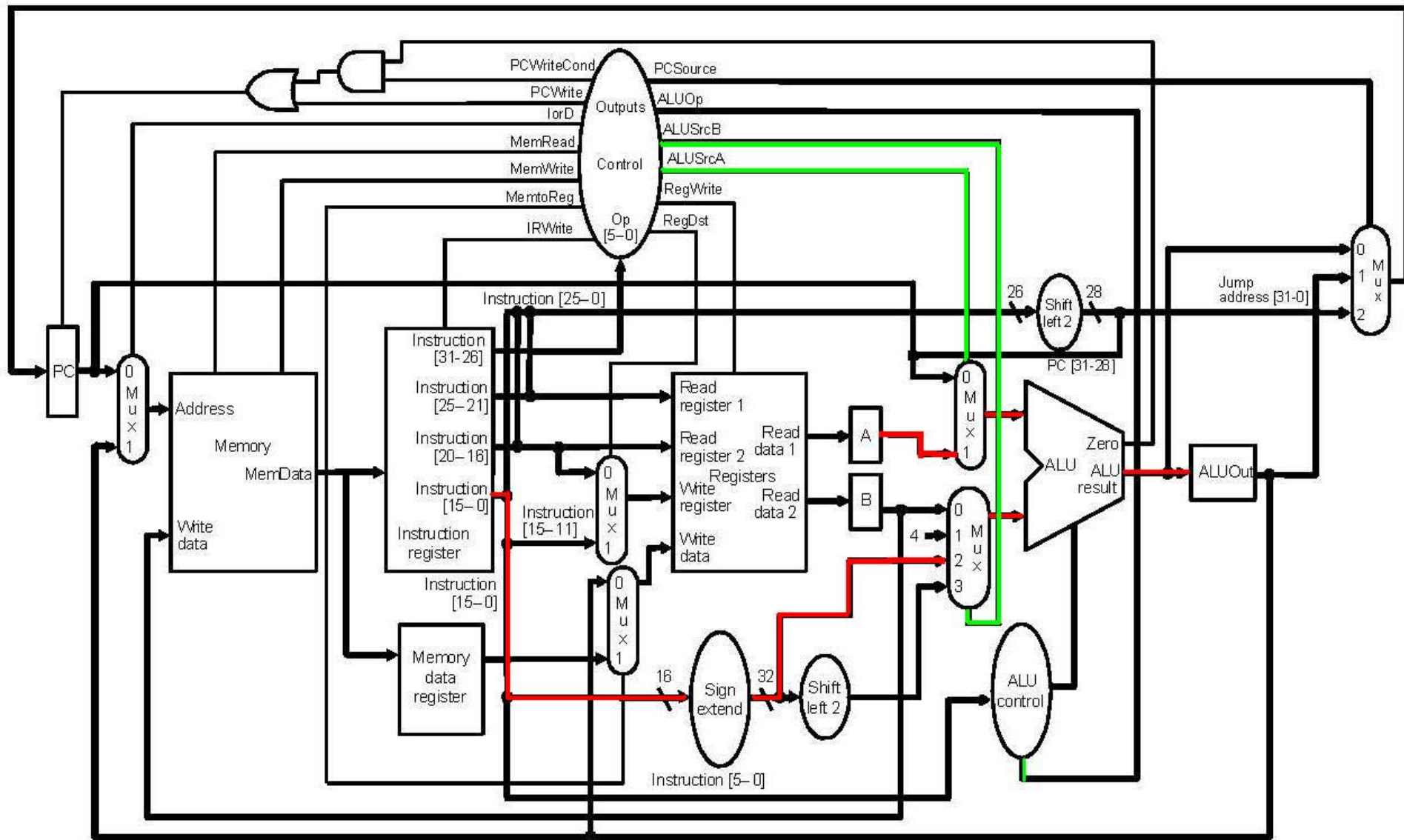
$$\text{ALUout} = \text{PC} + \text{extensió_signe}(\text{IR}[15-0]) \ll 2$$

MIPS multicicle: Controlador

- Instruccions d'accés a memòria (4 o 5 estats)

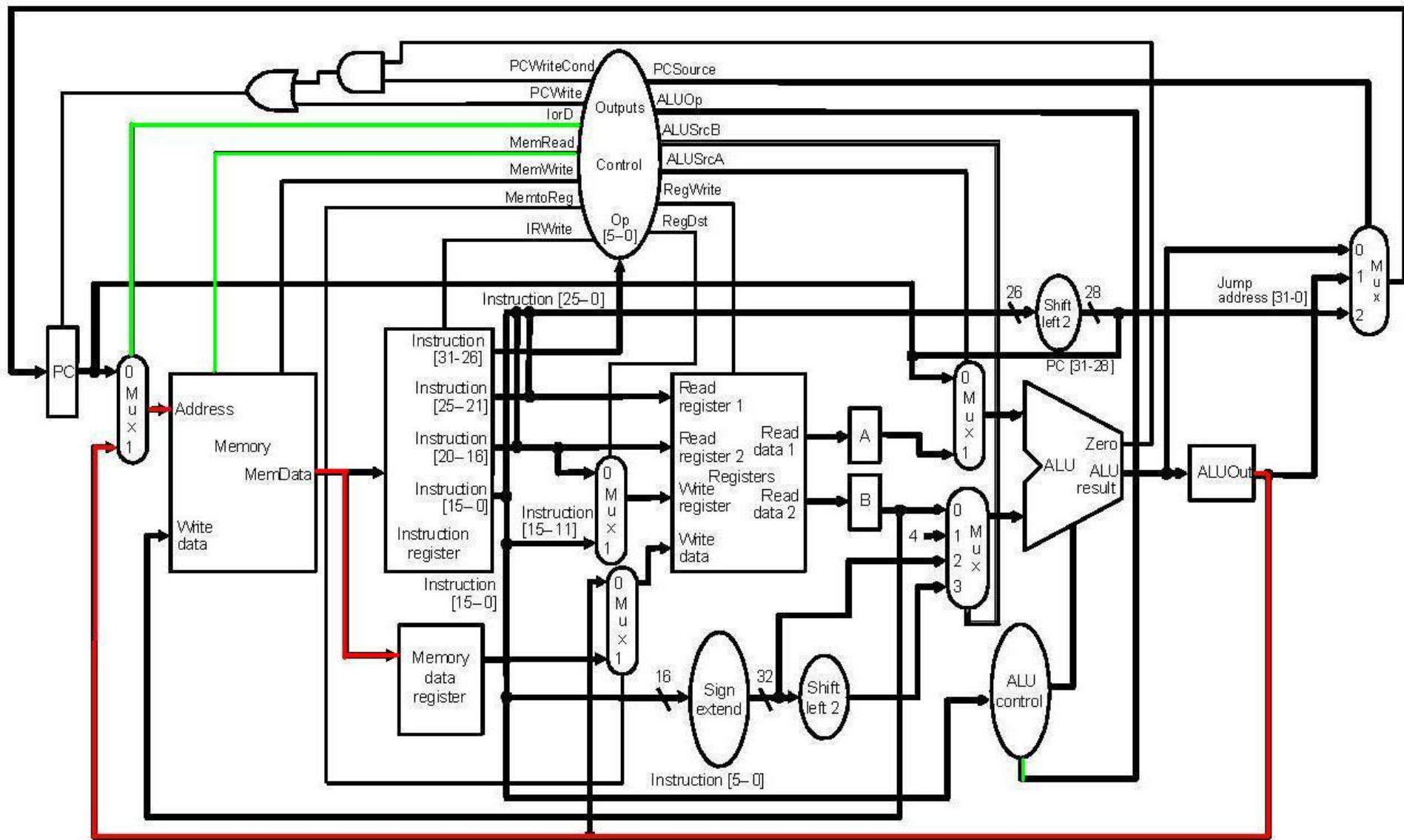


MIPS multicicle: Accés a memòria (Estat 3)

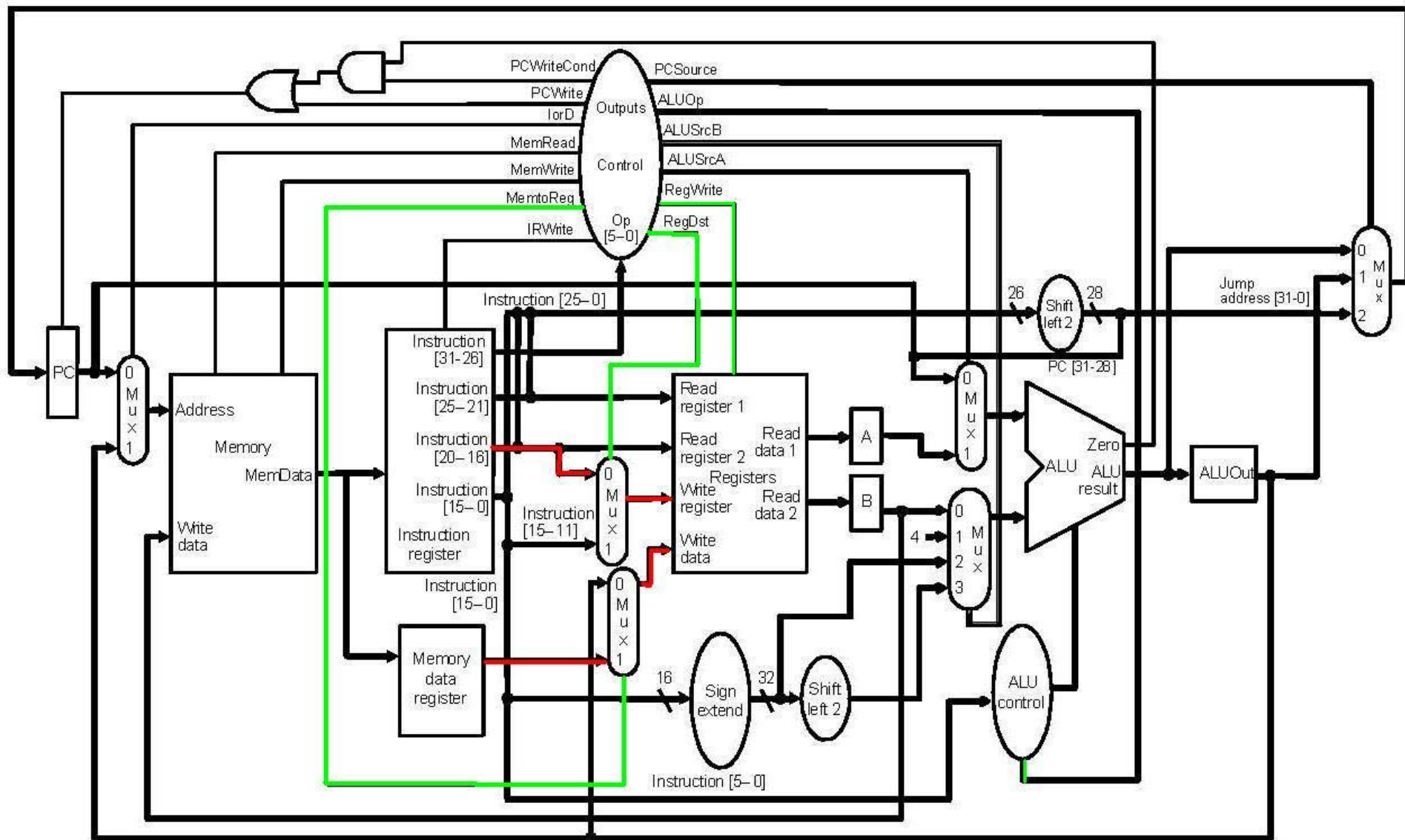


$$\text{ALUout} = \text{PC} + \text{extensió_signe}(\text{IR}[15-0]) \ll 2$$

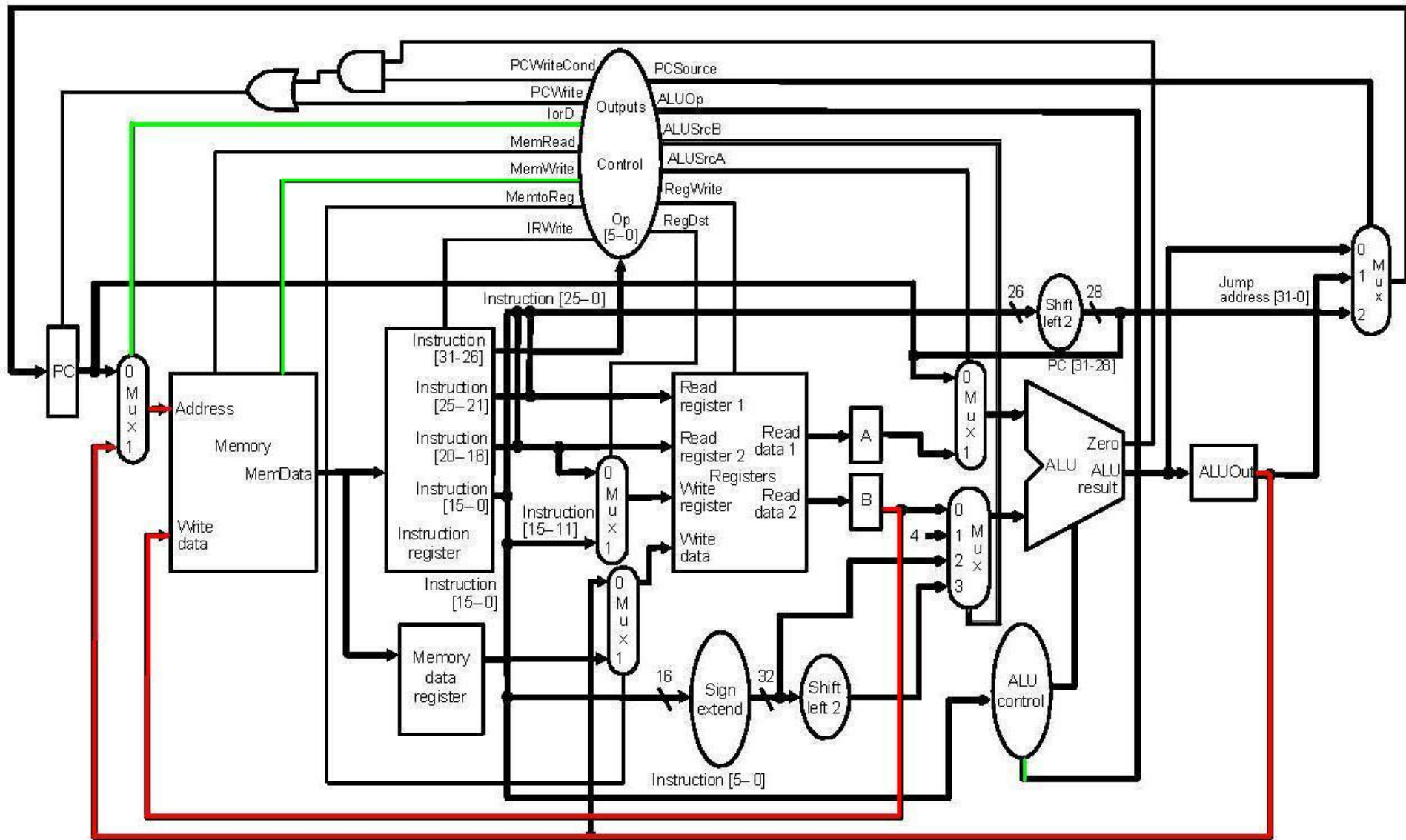
MIPS multicycle: Accés a memòria (Estat 4) lw



MIPS multicycle: Accès a memòria (Estat 5) lw

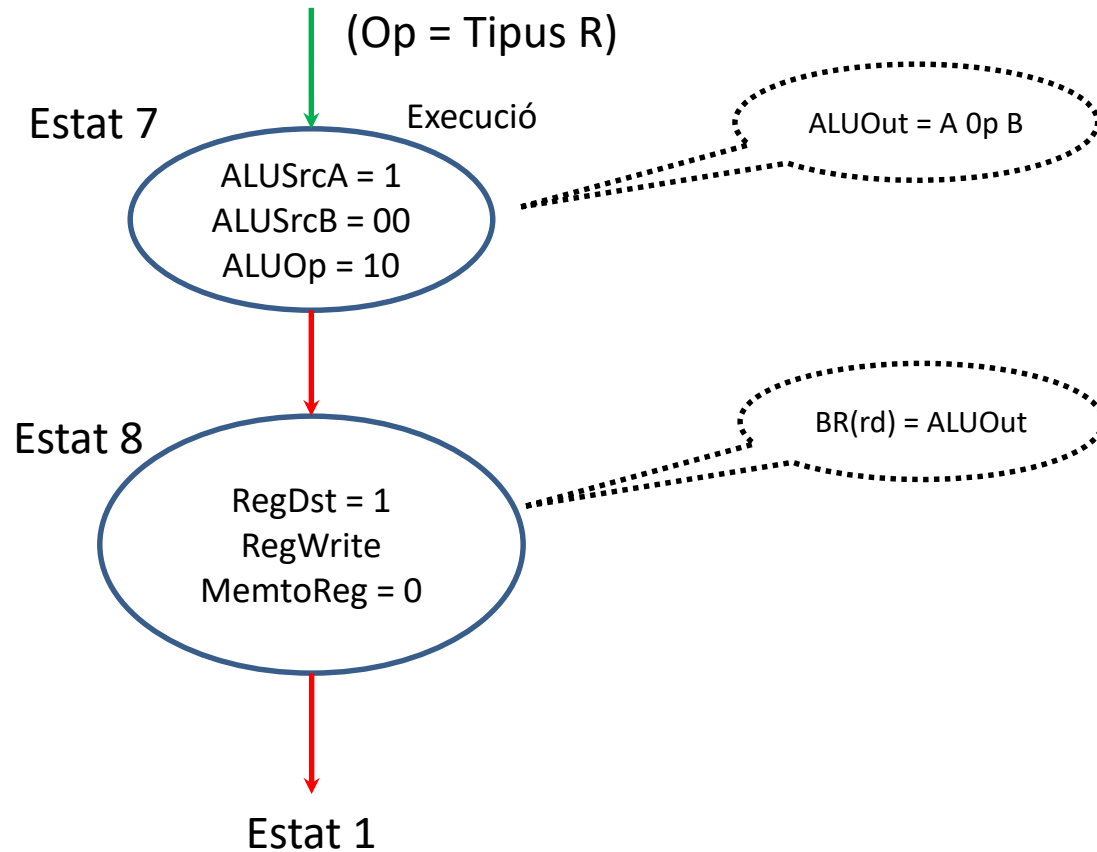


MIPS multicycle: Accès a memòria (Estat 6) sw

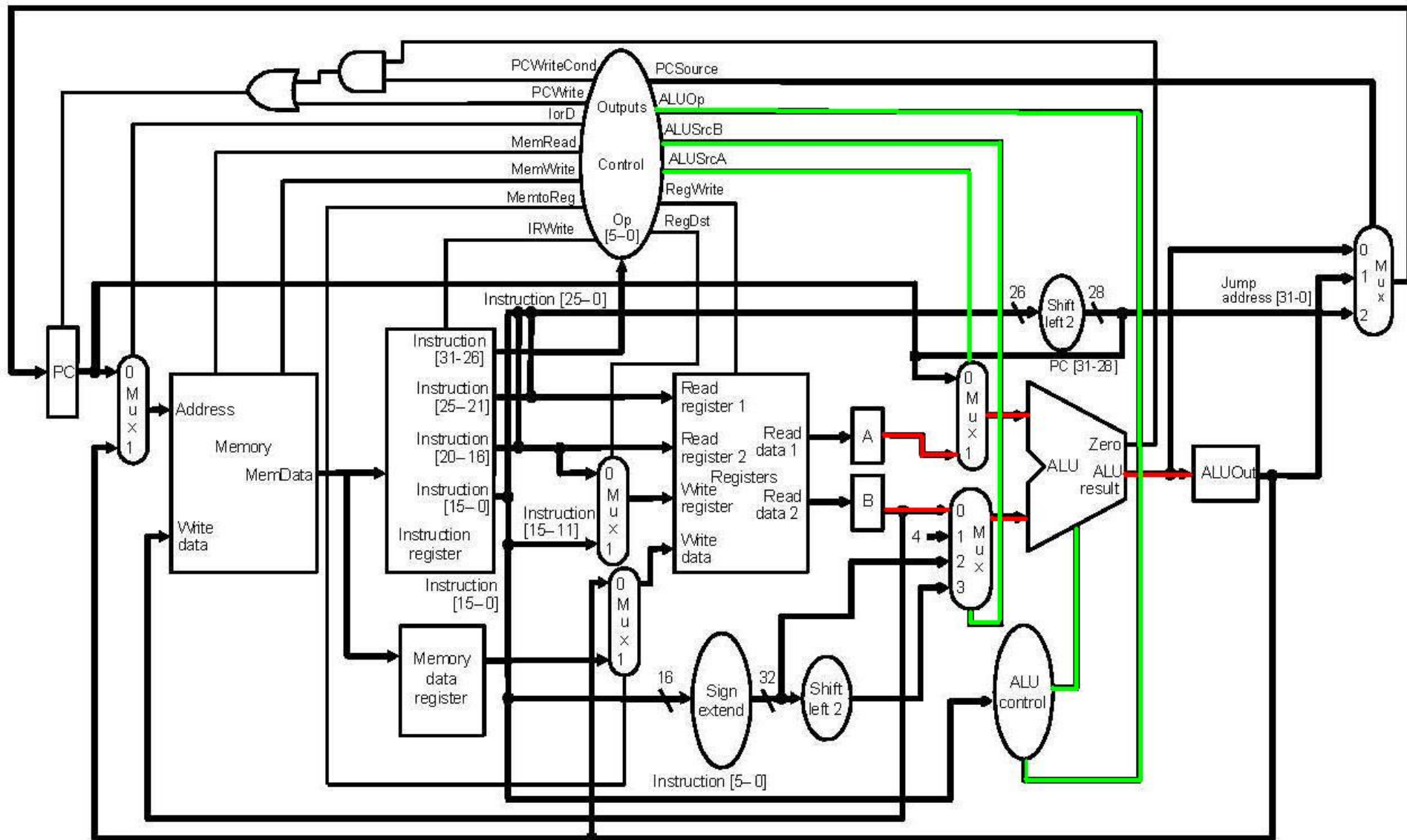


MIPS multicicle: Controlador

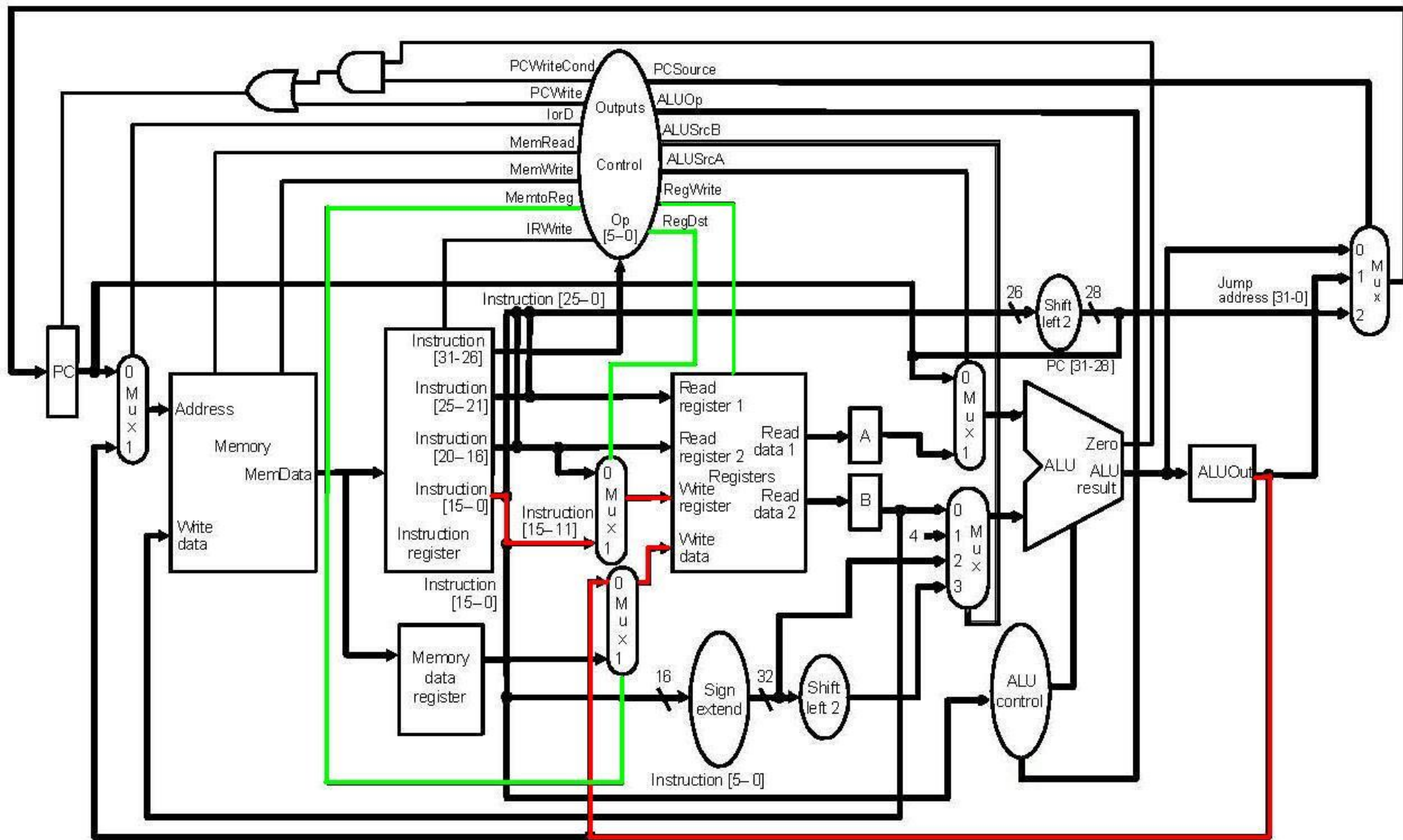
- Instruccions del tipus R (4 estats)



MIPS multicycle: Tipus R (Estat 7)

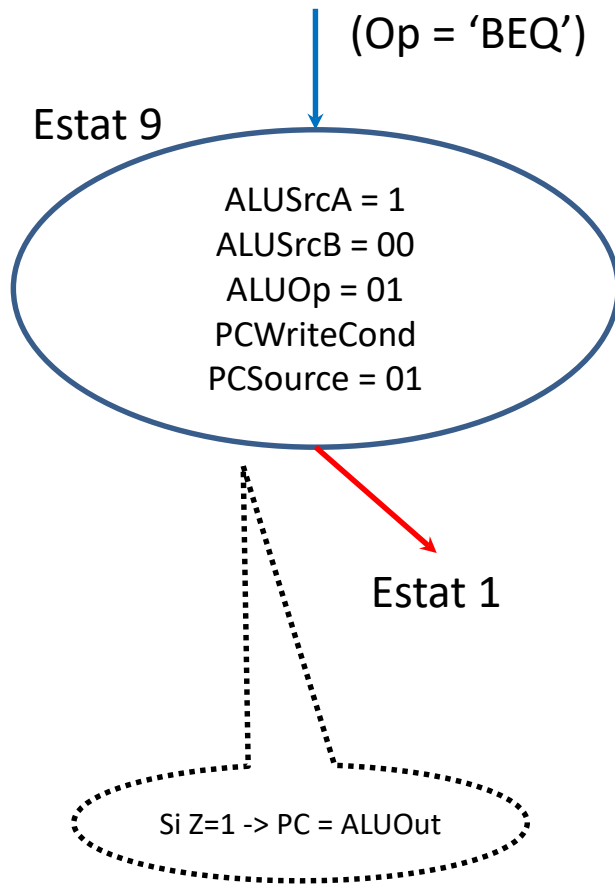


MIPS multicycle: Tipus R (Estat 8)

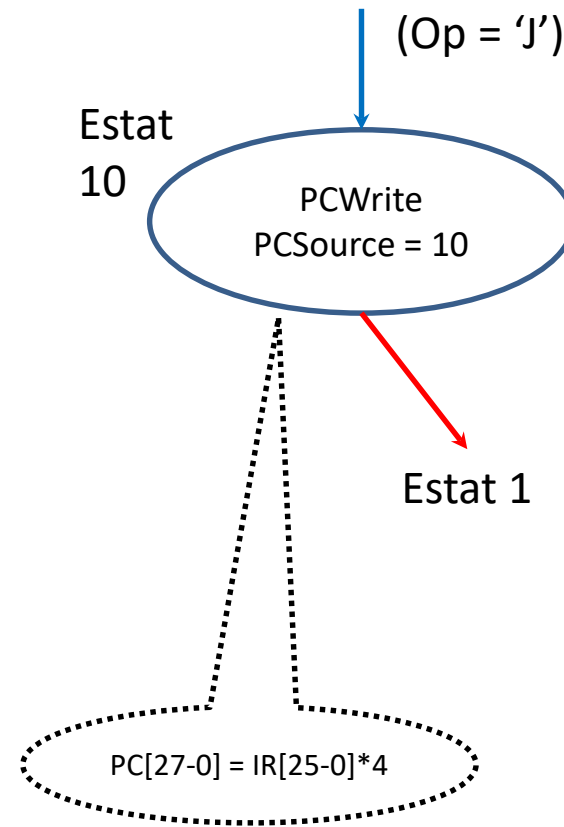


MIPS multicicle: Controlador

- Instruccions del tipus salt condicional (3 estats)

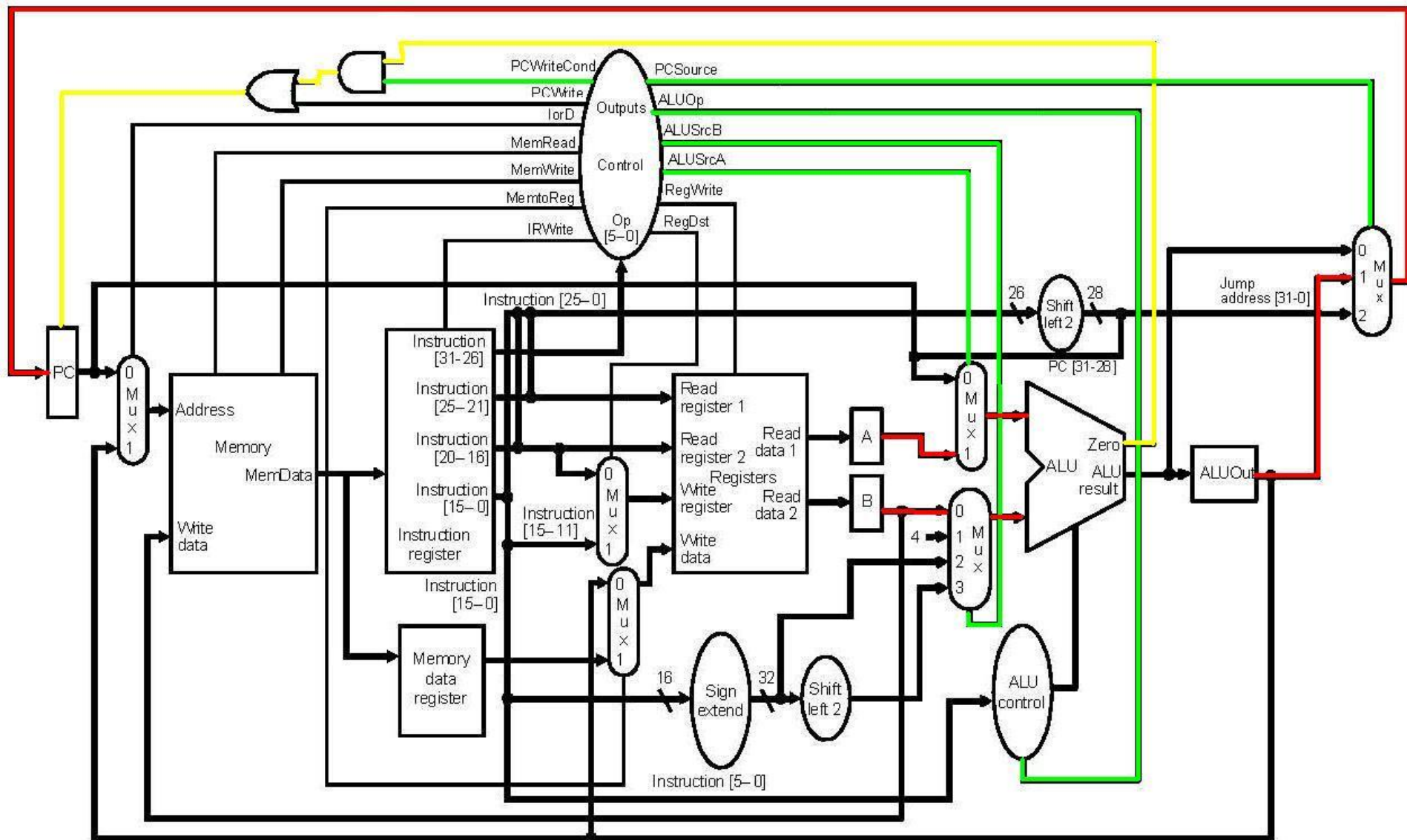


- Instruccions del tipus salt (3 estats)



MIPS multicycle: Salt condicional

(Estat 9)



MIPS multicycle: Salt incondicional (Estat 10)

