Disseny del camí de dades multicicle per l'arquitectura MIPS

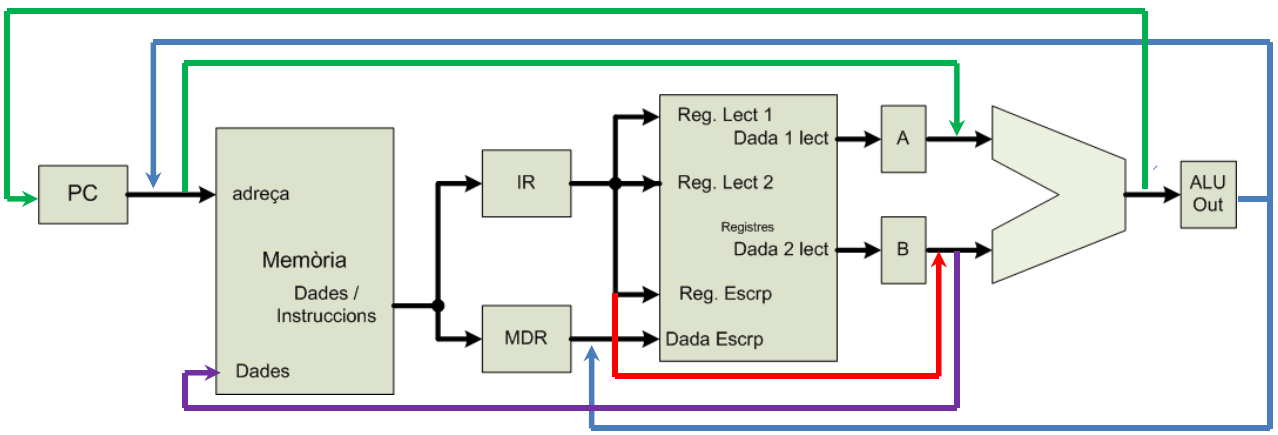
MIPS multicicle

- 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

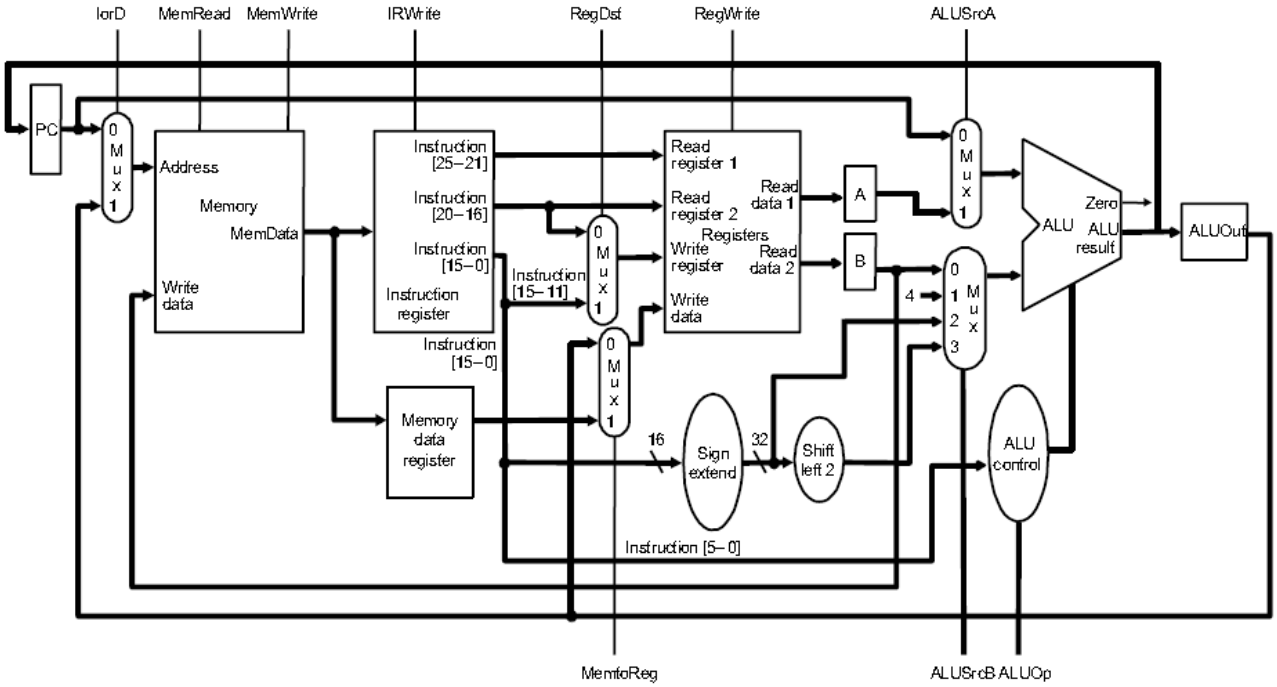
- Implementació Multicicle
  - Divisió de l'execució de les instruccions en varies etapes.
    - S'executa una etapa de la instrucció a cada cicle de rellotge
- Avantatges
  - Reutilització de les unitats funcionals
    - Una unitat pot utilitzar-se més d'un cop per instrucció sempre que es faci en diferents 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



- Una sola memòria per dades i instruccions
- Una sola ALU enlloc d'una ALU i dos sumadors
- Ús de registres temporals a les sortides de les unitats funcionals principals. Dos tipus de registres:
  - Actualitzen el seu contingut a cada instrucció (IR)
  - Actualitzen el seu contingut a cada cicle de rellotge (MDR, A, B, ALUOut)

MIPS multicicle: Components bàsics

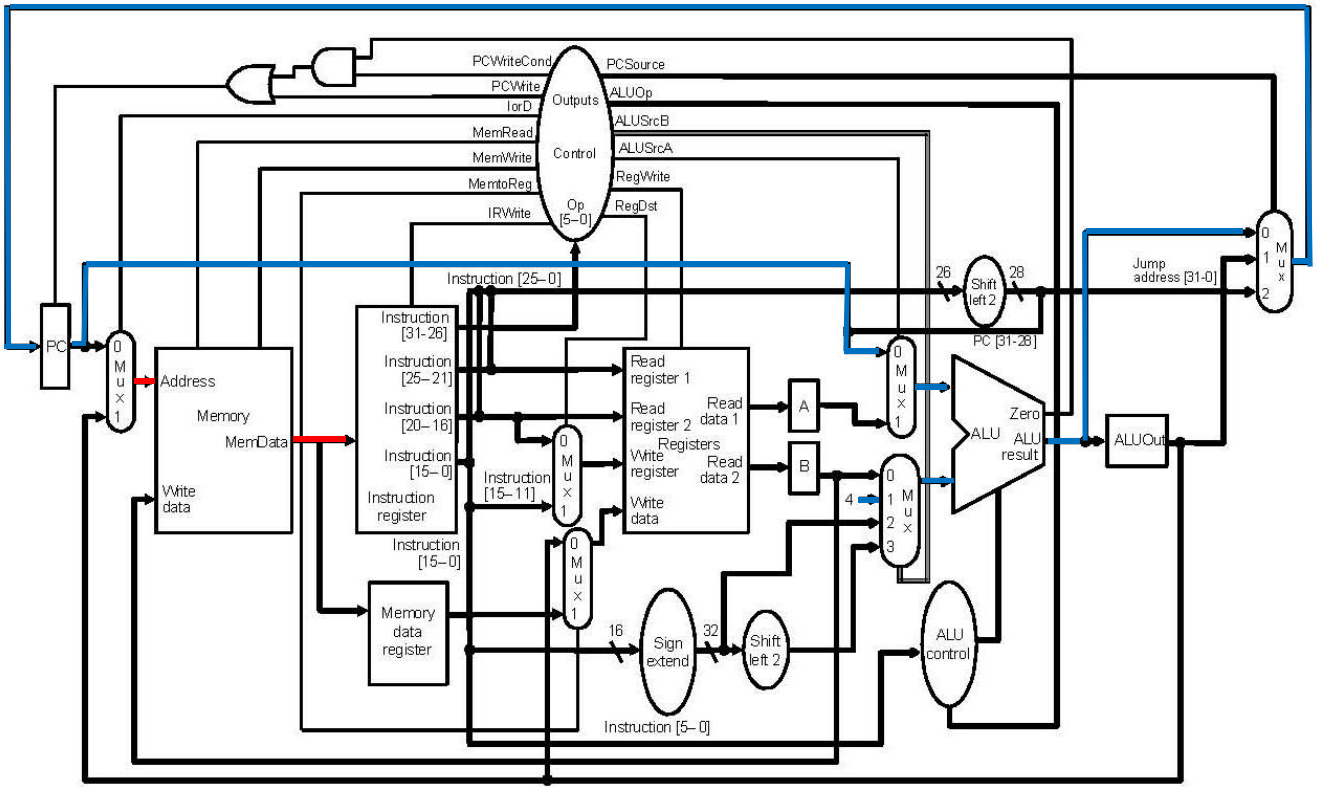


- Compartir les unitats funcionals ⟶ Multiplexors
- Execució en múltiples cicles ⟶ Cada cicle requereix un conjunt de senyals de control
- Senyals de control

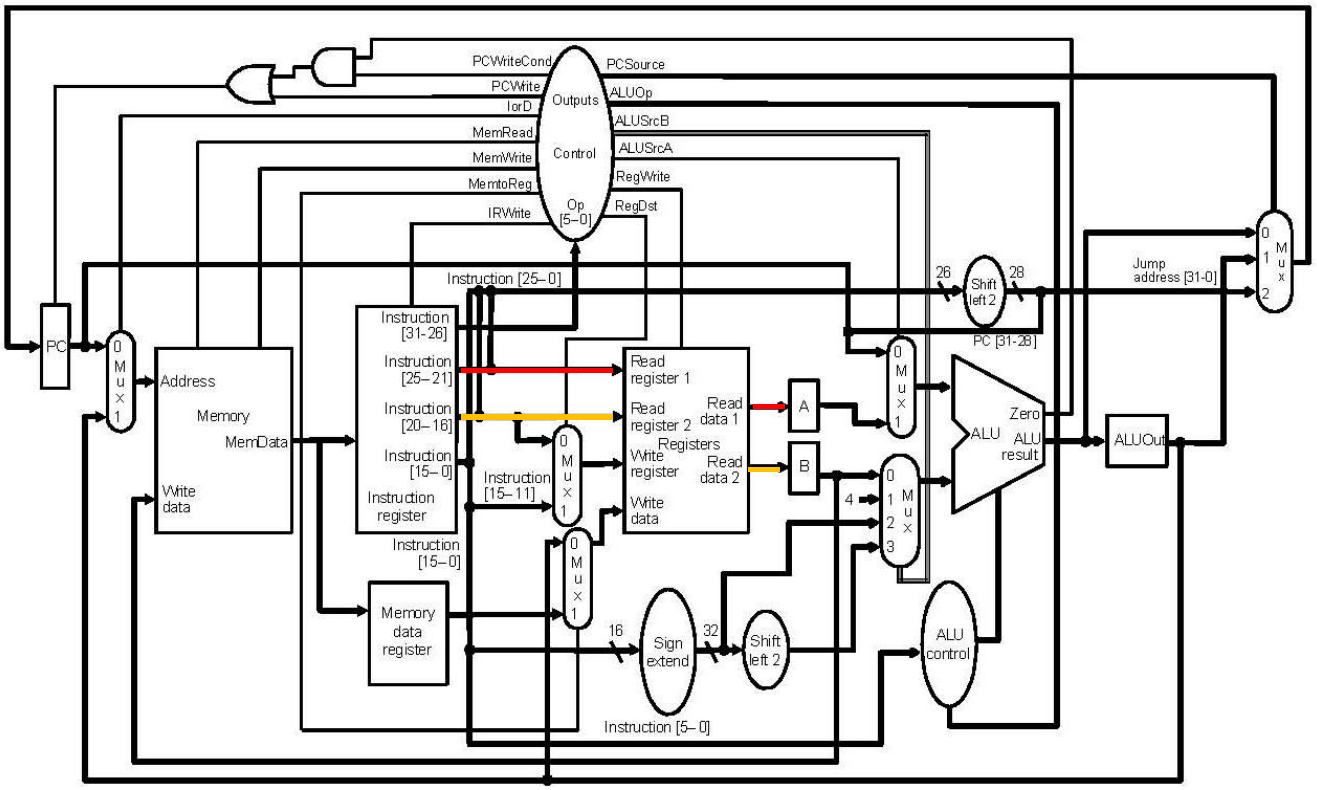
Etapes d'execució de les instruccions

- 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 es sèrie

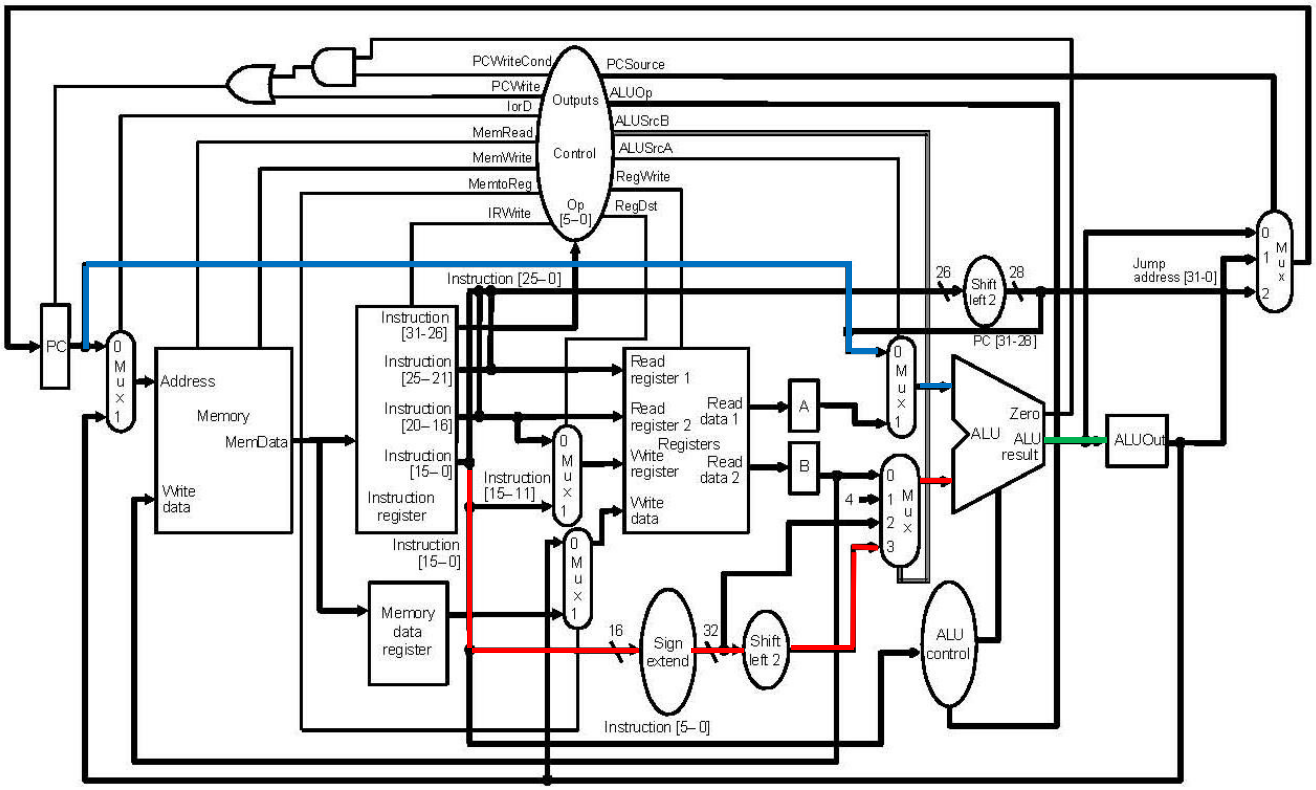
MIPS multicicle: Etapa 1



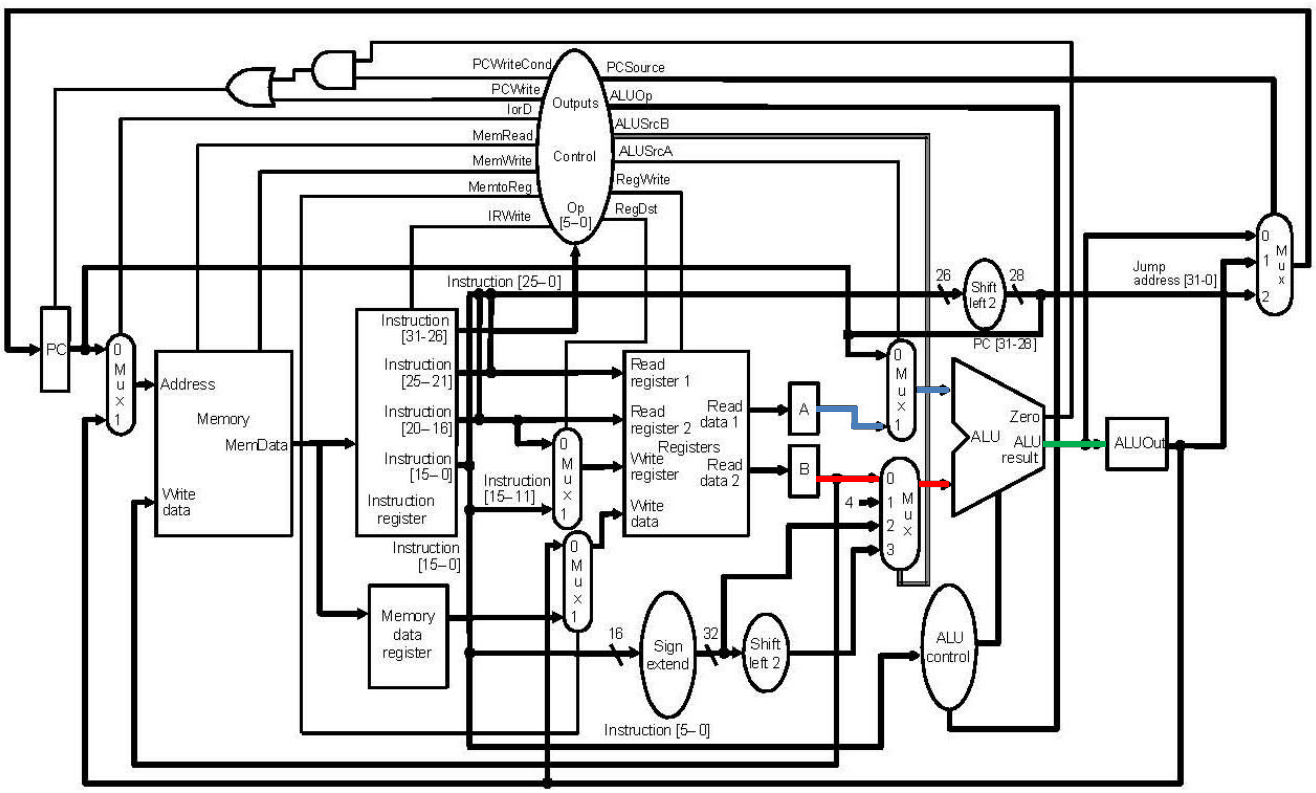
MIPS multicicle: Etapa 2 (A = Reg[rs], B = Reg[rt])



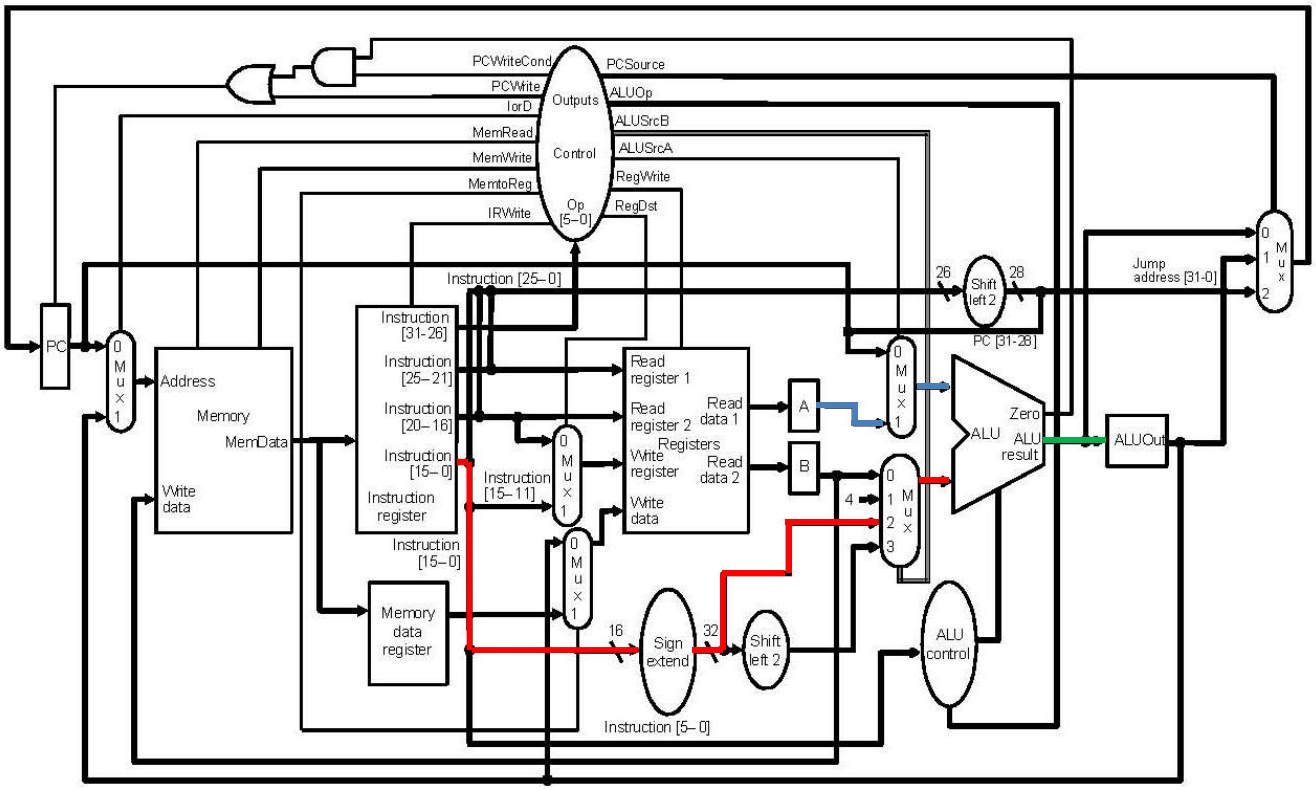
MIPS multicicle: Etapa 2 (ALUout = PC + extensió-signe(IR[15-0]) << 2)



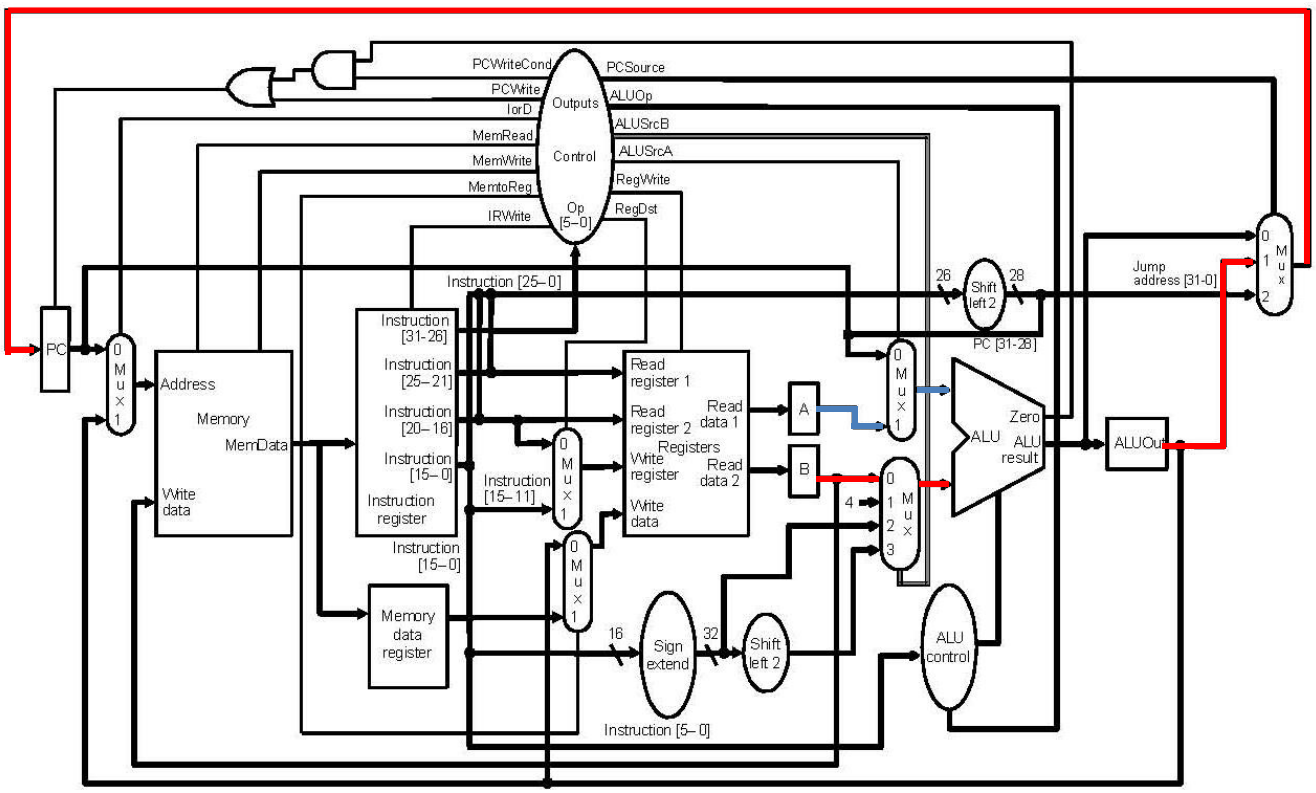
MIPS multicicle: Etapa 3 (Tipus R)



MIPS multicicle: Etapa 3 (Referència a memòria)



MIPS multicicle: Etapa 3 (Salt condicional)

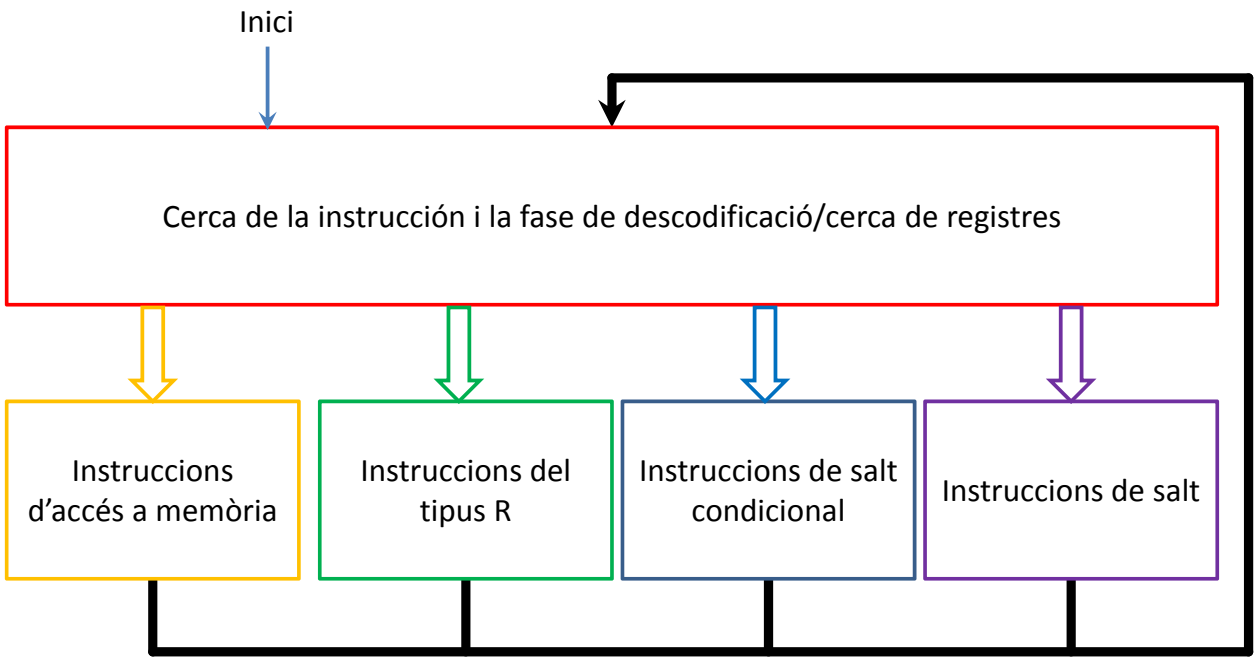


MIPS multicicle: Etapes d'execució de les instruccions

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

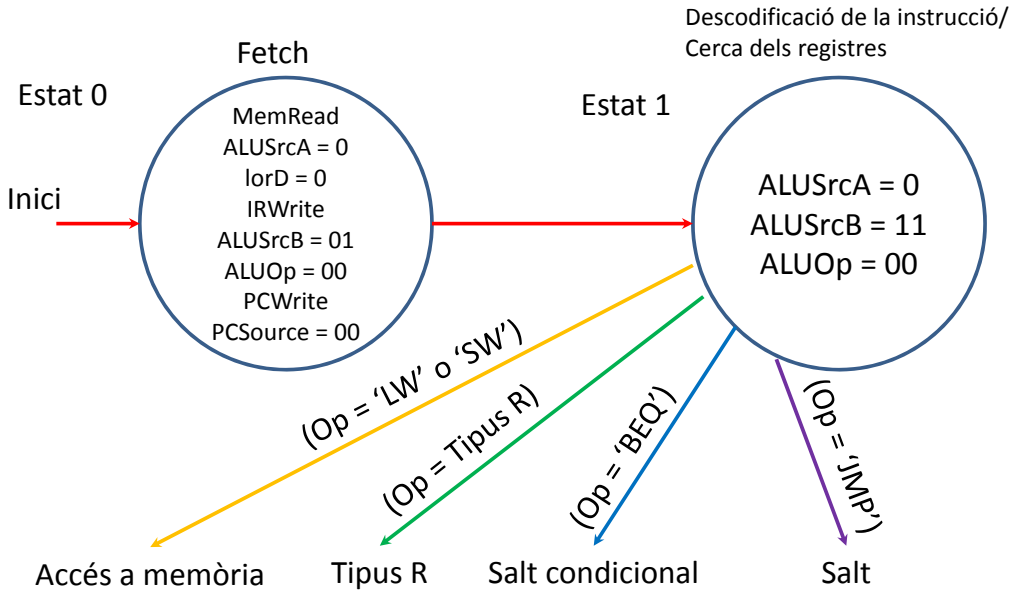
MIPS multicicle: Controlador

- Implementació amb una màquina d'estats finits

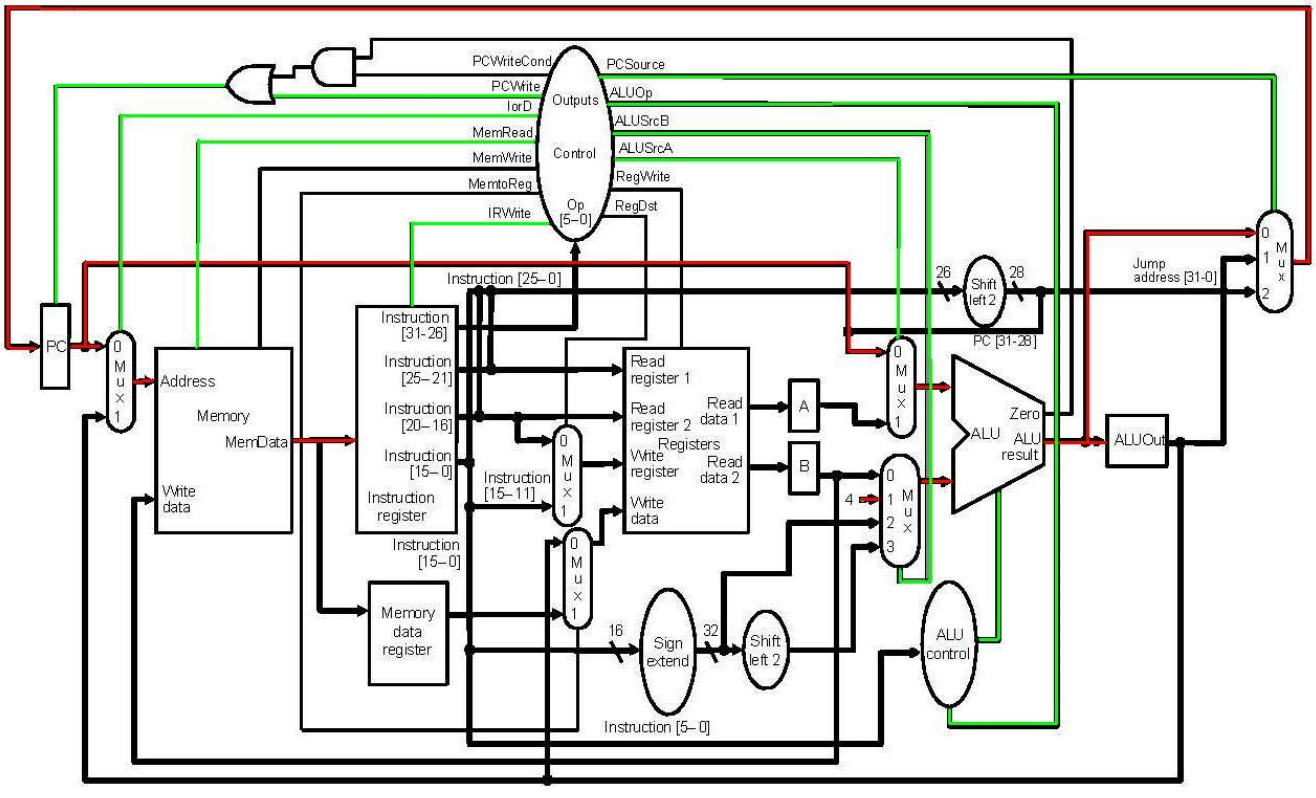


MIPS multicicle: Controlador

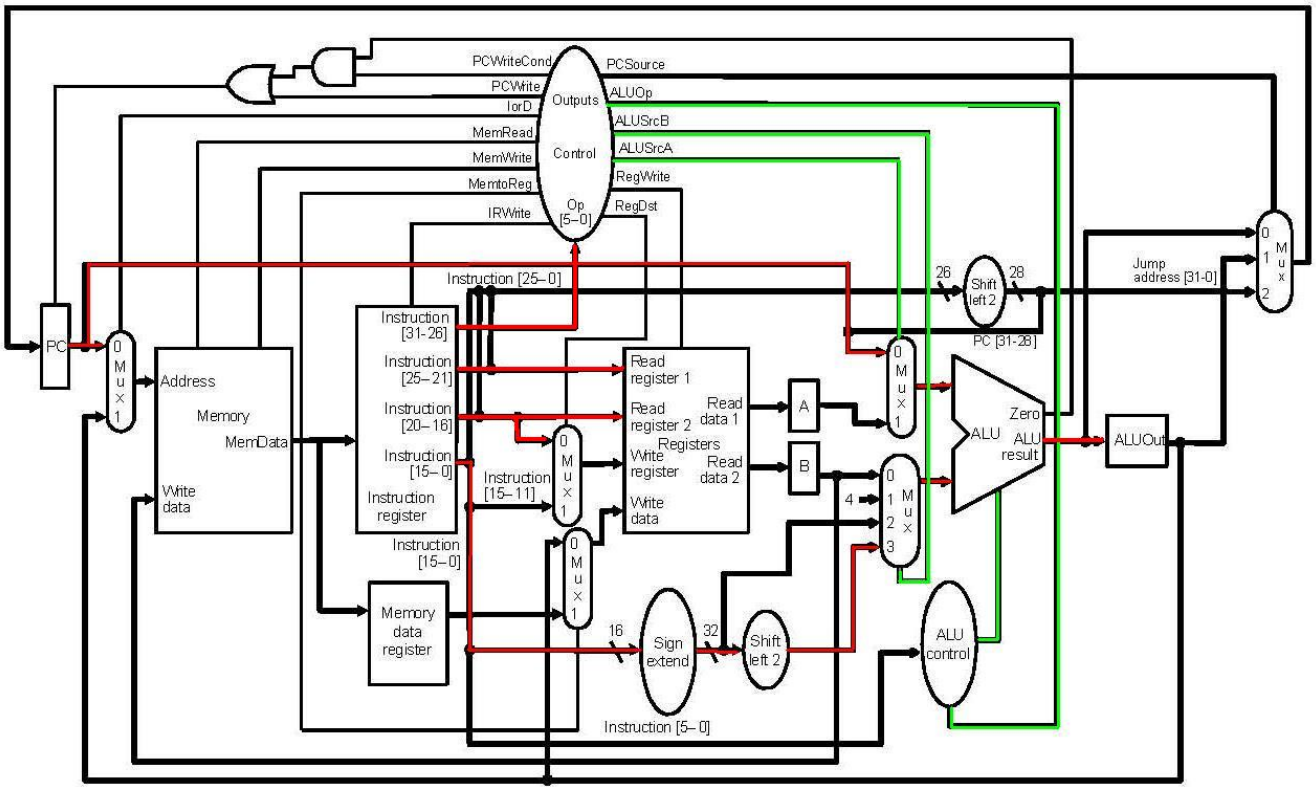
- Cerca de la instrucción i la fase de descodificació/cerca de registres



MIPS multicicle: Fetch



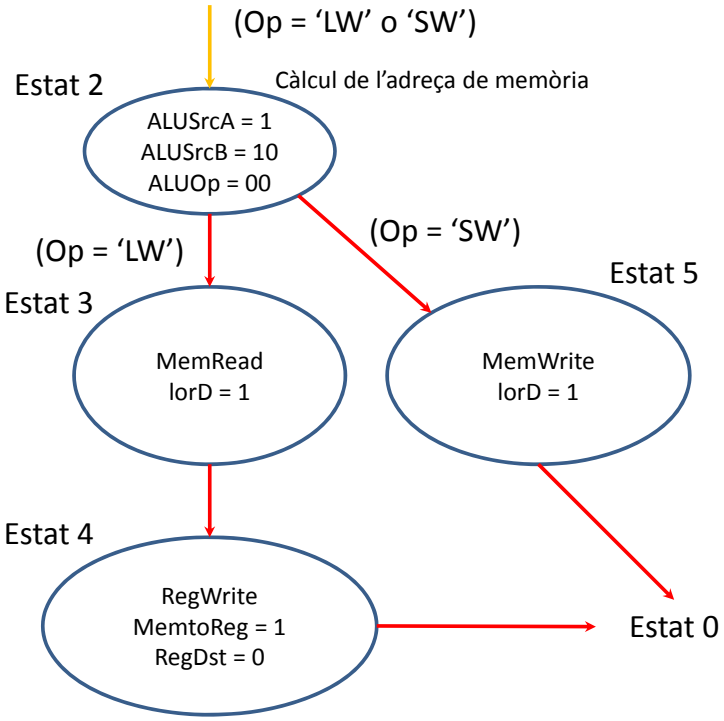
MIPS multicicle: Descodificació i càlcul adreça



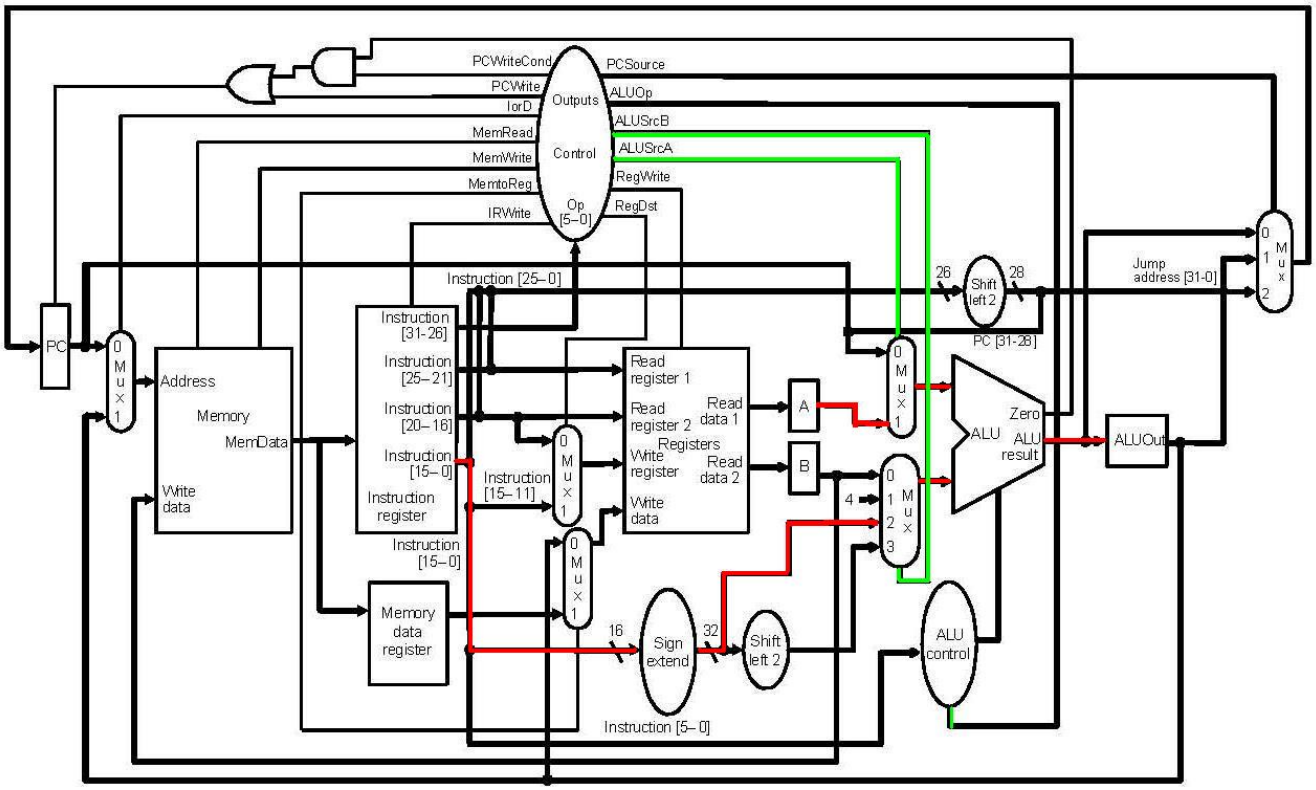
ALUout = PC + extensió_signe(IR[15-0]) << 2

MIPS multicicle: Controlador

- Instruccions d'accés a memòria (4 estats)

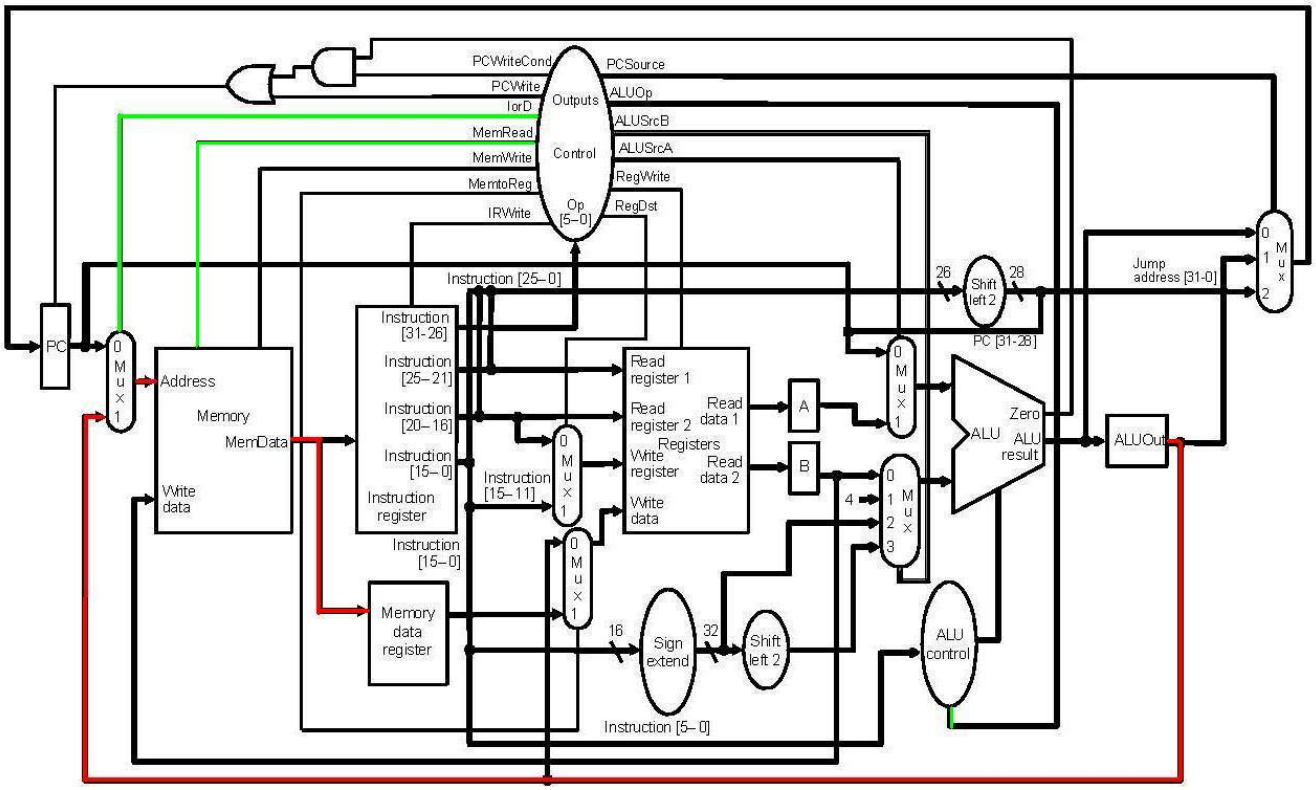


MIPS multicicle: Accés a memòria (Estat 2)

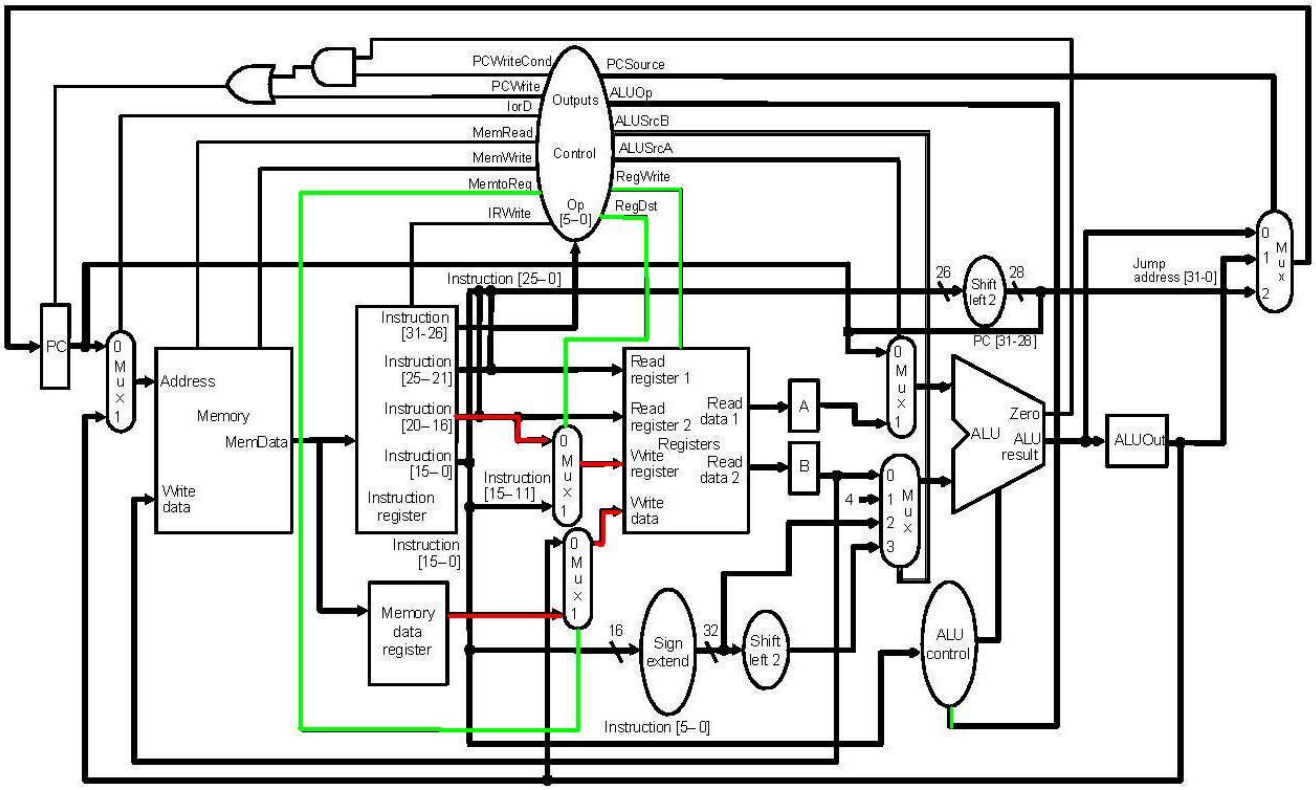


ALUout = PC + extensió_signe(IR[15-0]) << 2

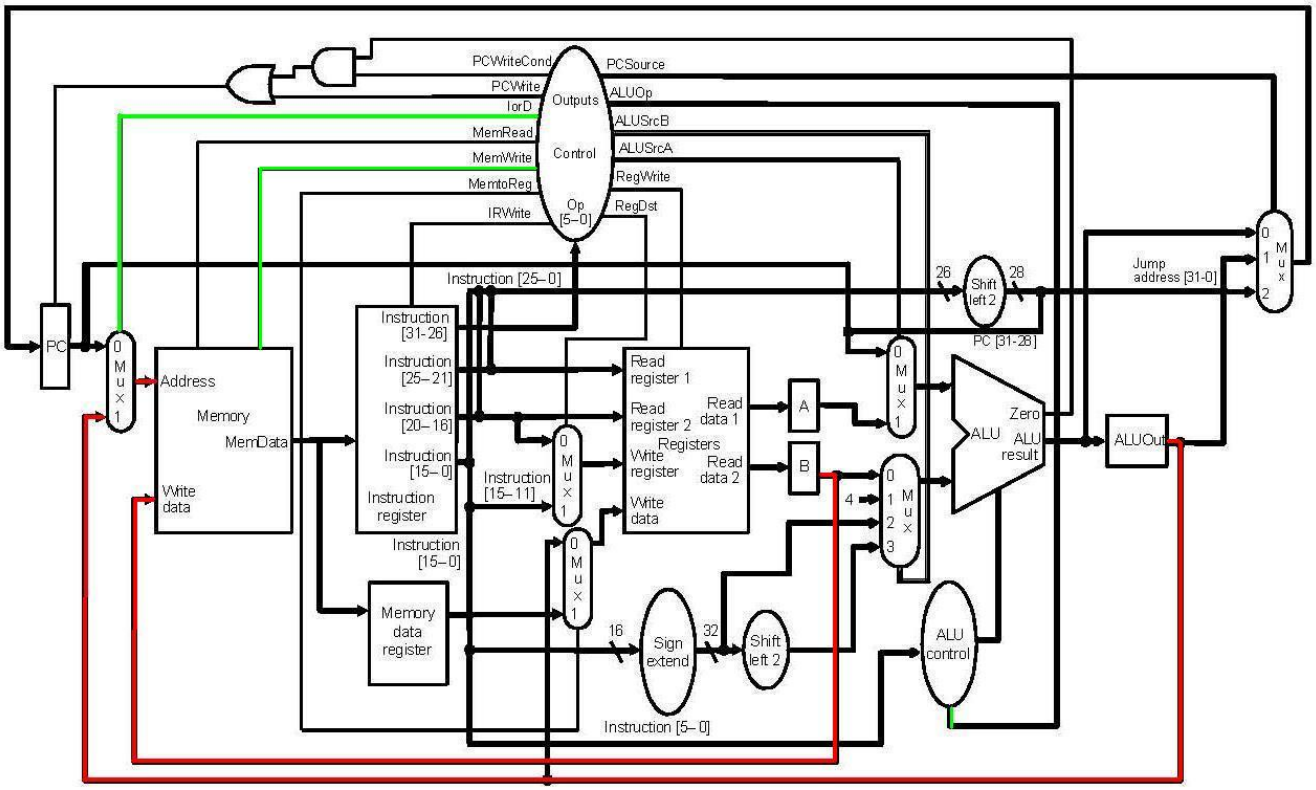
MIPS multicicle: Accés a memòria (Estat 3) lw



MIPS multicicle: Accés a memòria (Estat 4) lw

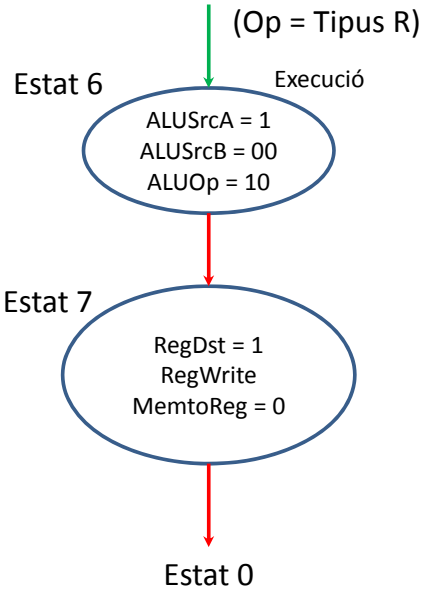


MIPS multicicle: Accés a memòria (Estat 5) sw

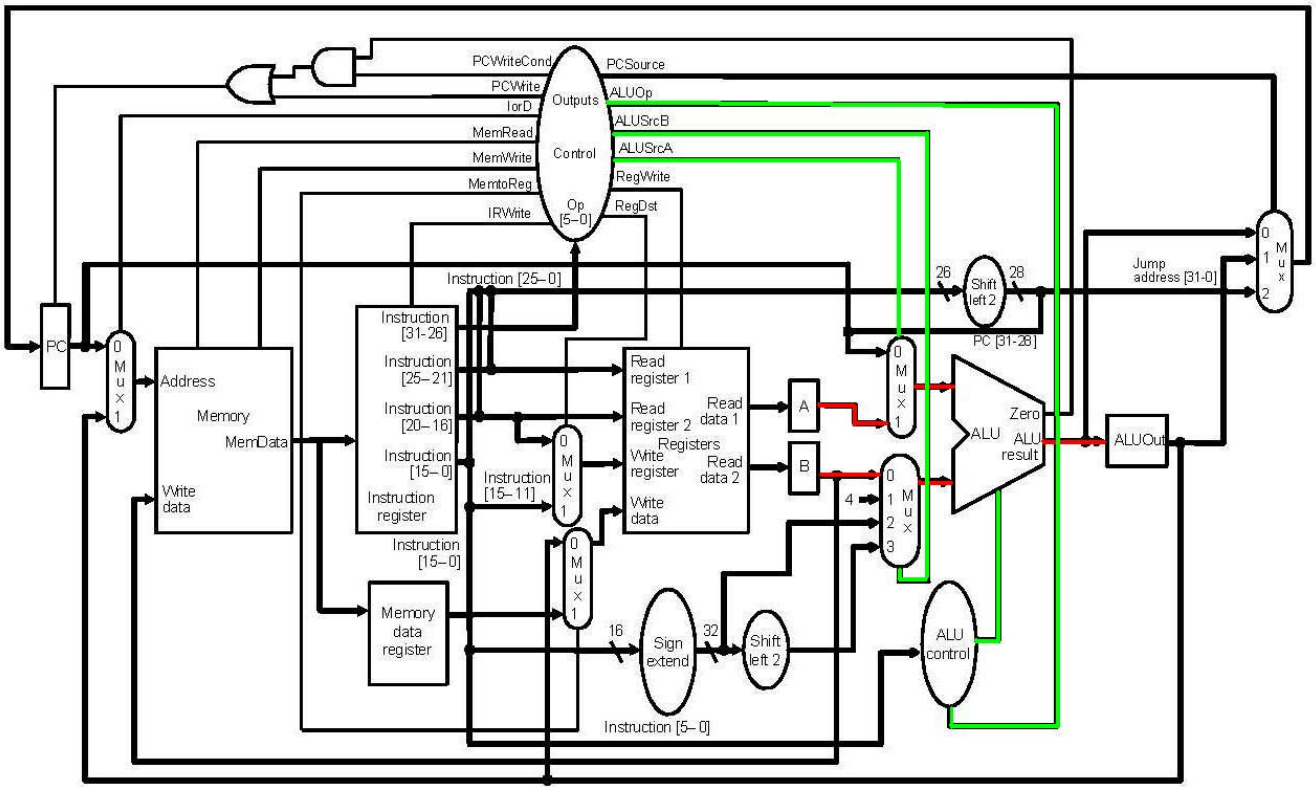


MIPS multicicle: Controlador

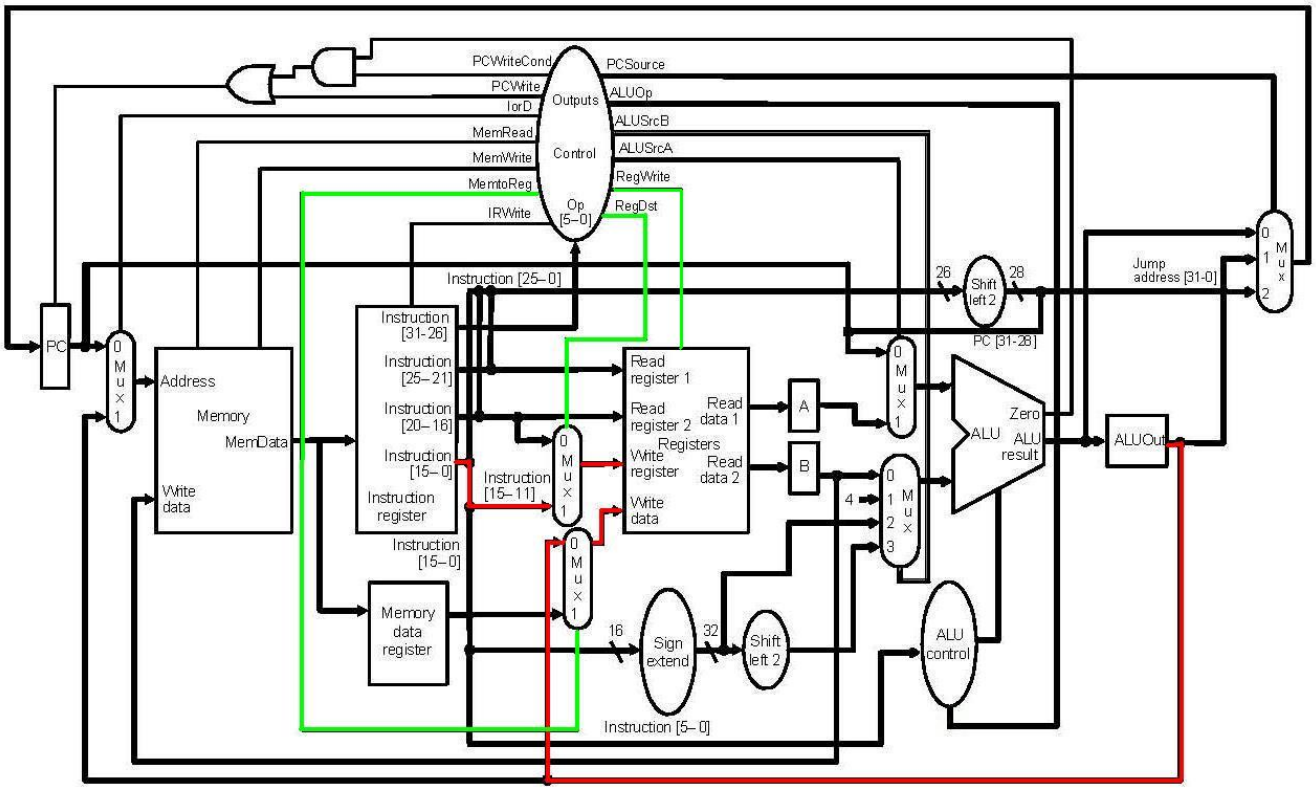
- Instruccions del tipus R (2 estats)



MIPS multicicle: Tipus R (Estat 6)

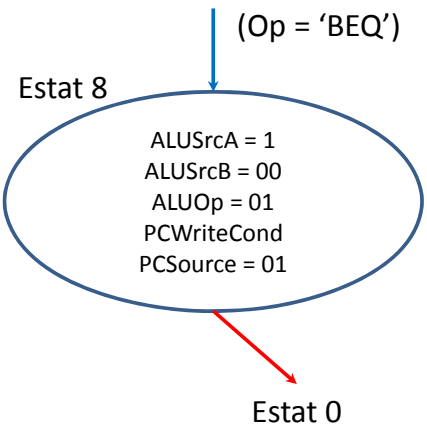


MIPS multicicle: Tipus R (Estat 7)

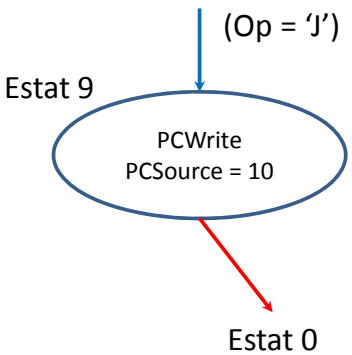


MIPS multicicle: Controlador

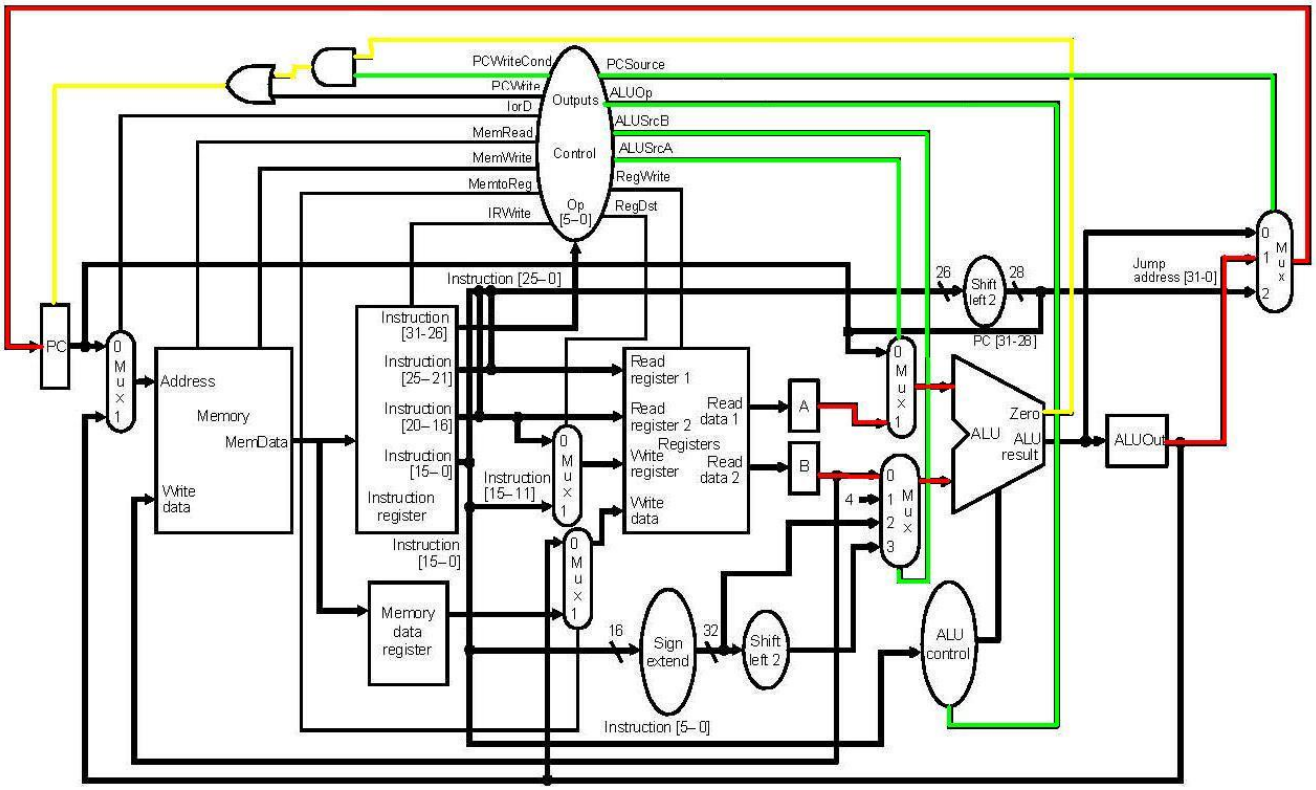
- Instruccions del tipus salt condicional (1 estat)



- Instruccions del tipus salt (1 estat)



MIPS multicicle: Salt condicional (Estat 8)



MIPS multicicle: Salt incondicional (Estat 9)