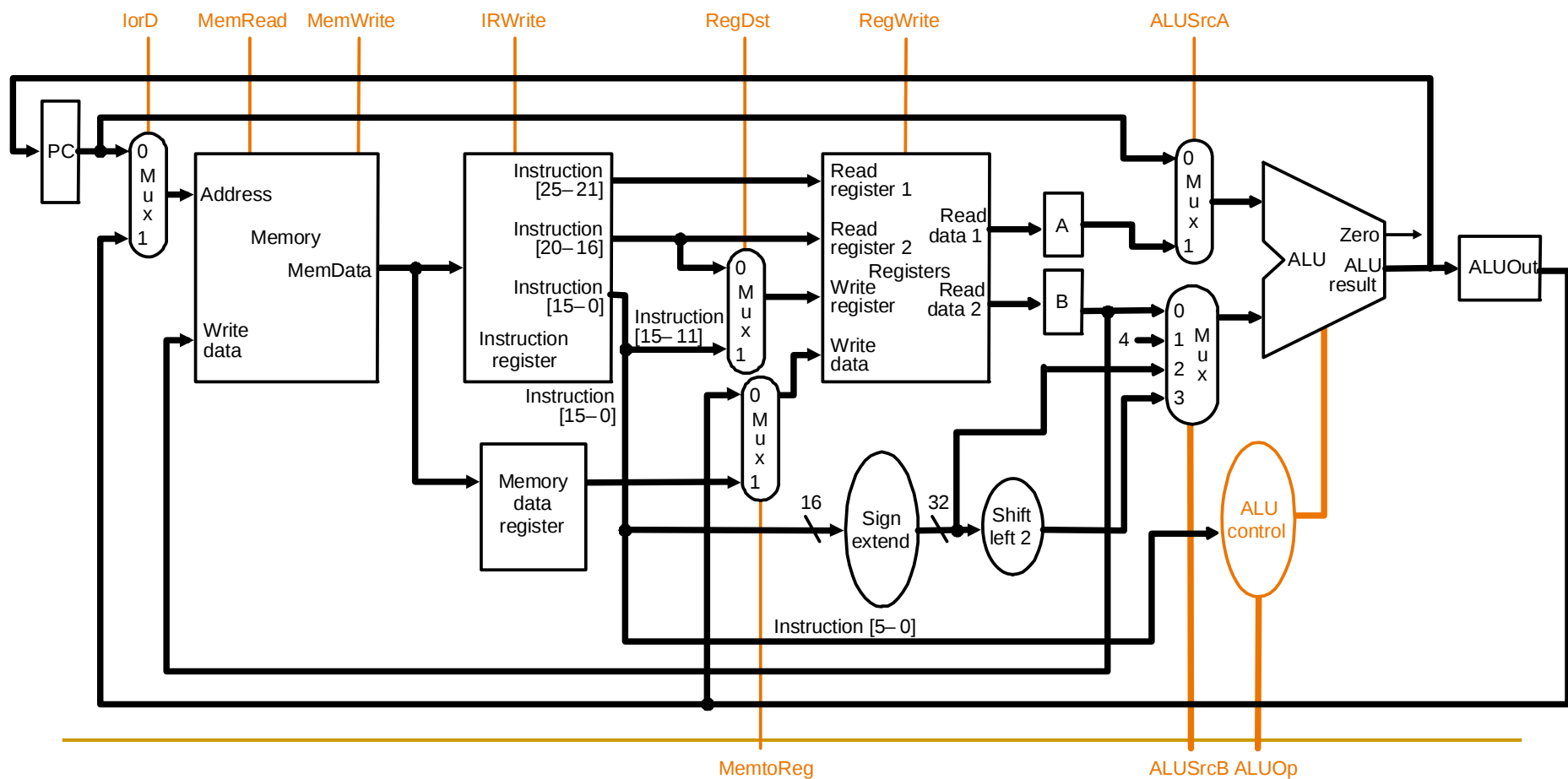


Arquiteturas de Computadores

Implementação de MIPS multiclo (cont.)

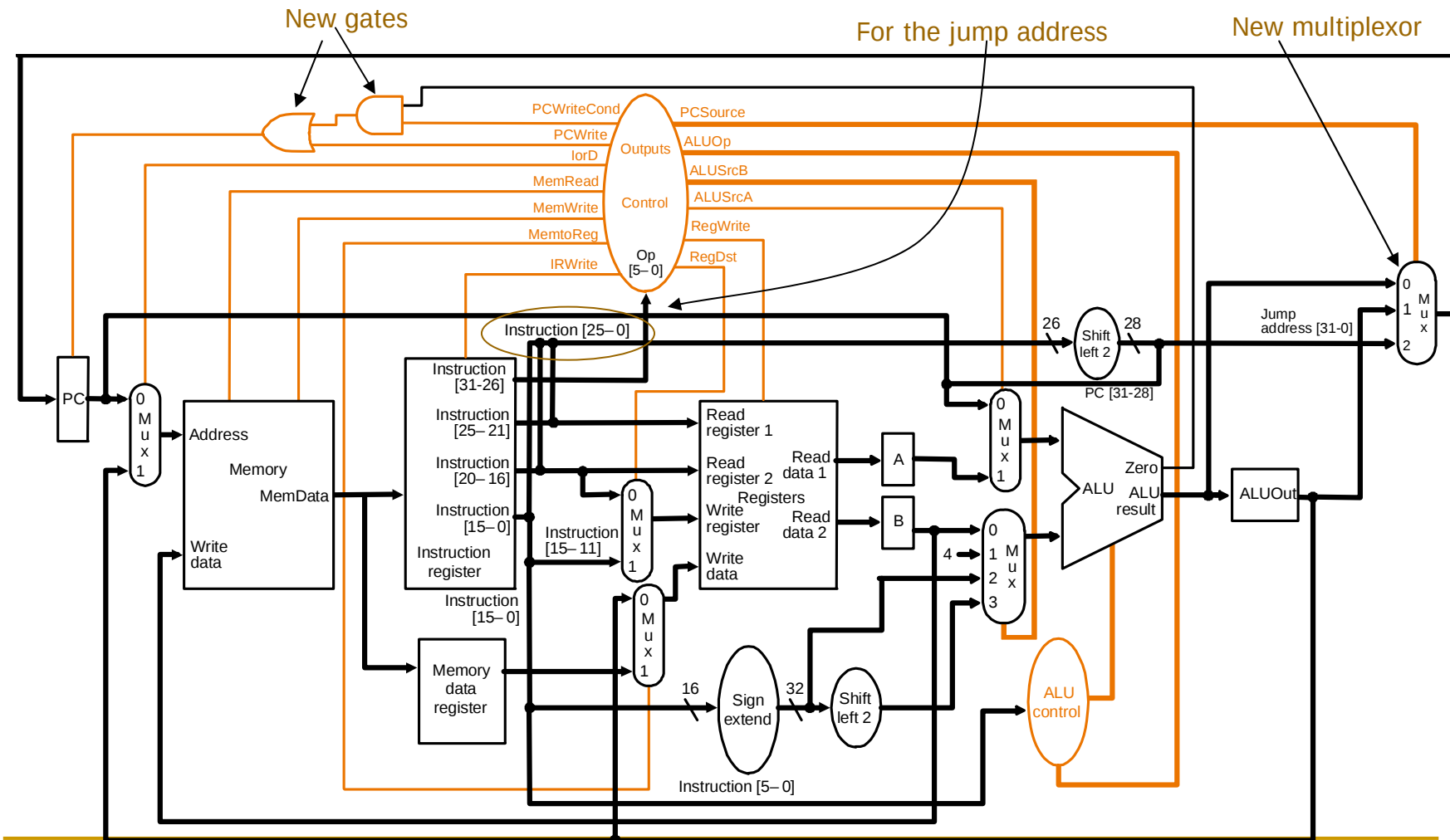
Fontes dos slides: Patterson &
Hennessy book website
(copyright Morgan Kaufmann)
e Dr. Sumanta Guha

Caminho de dados com controle I



...somente linhas de controle e da ALU mostradas

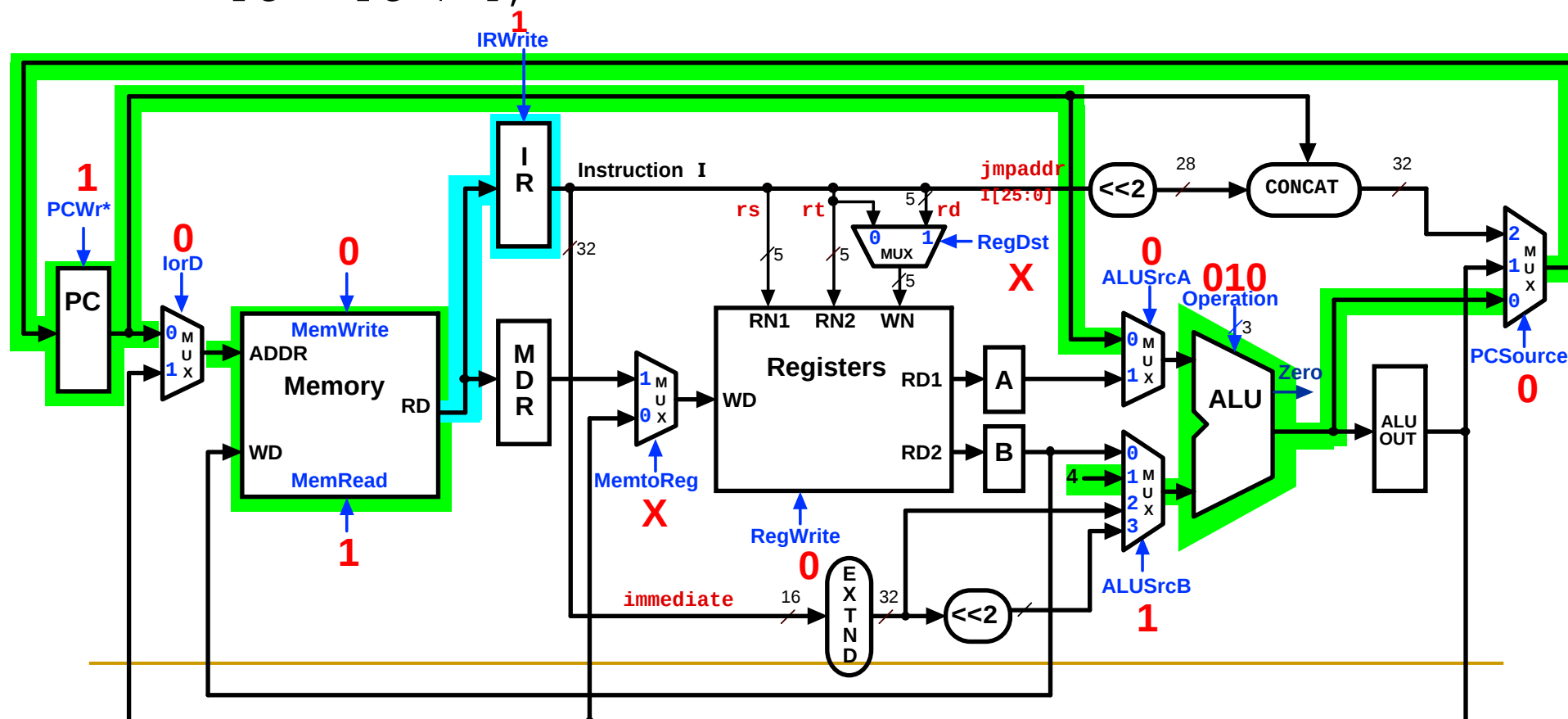
Caminho de dados com controle II



Caminho de dados para MIPS multiciclo

Passo 1 para busca de instrução

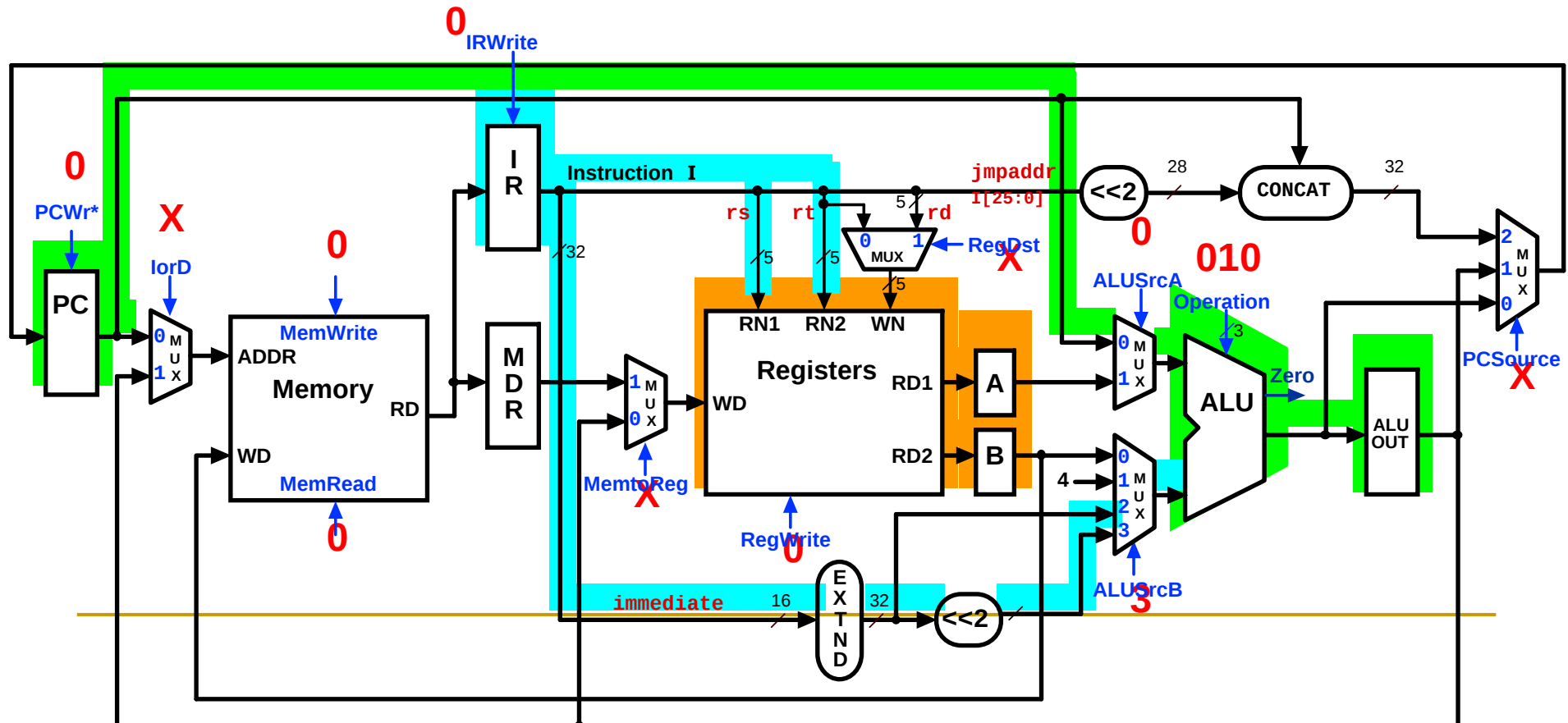
```
IR = Memory[PC];
```

$$\text{PC} = \text{PC} + 4;$$


Passo 2 para decodificação e busca no registrador

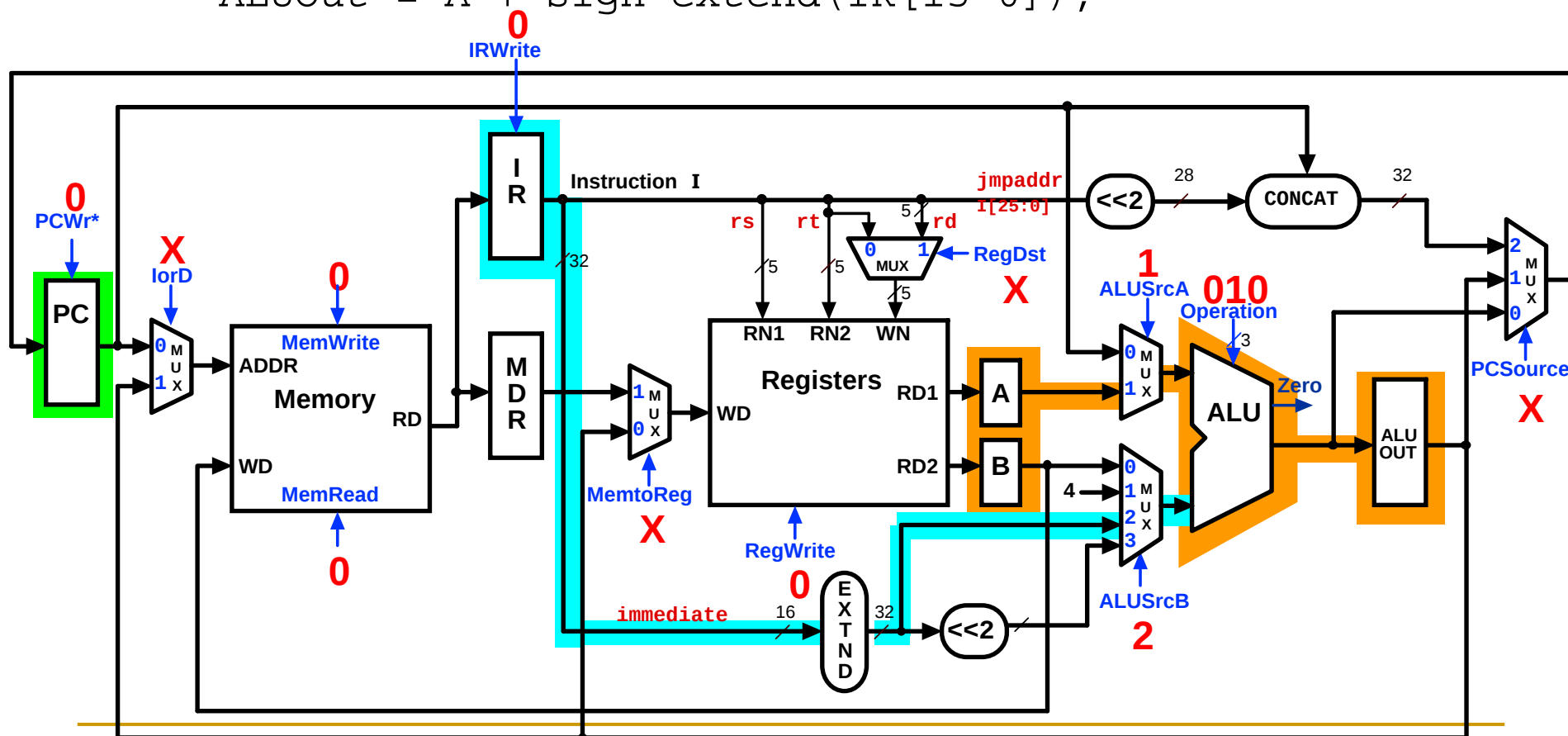
```

A = Reg[IR[25-21]];          (A = Reg[rs])
B = Reg[IR[20-15]];          (B = Reg[rt])
ALUOut = (PC + sign-extend(IR[15-0]) << 2);
    
```



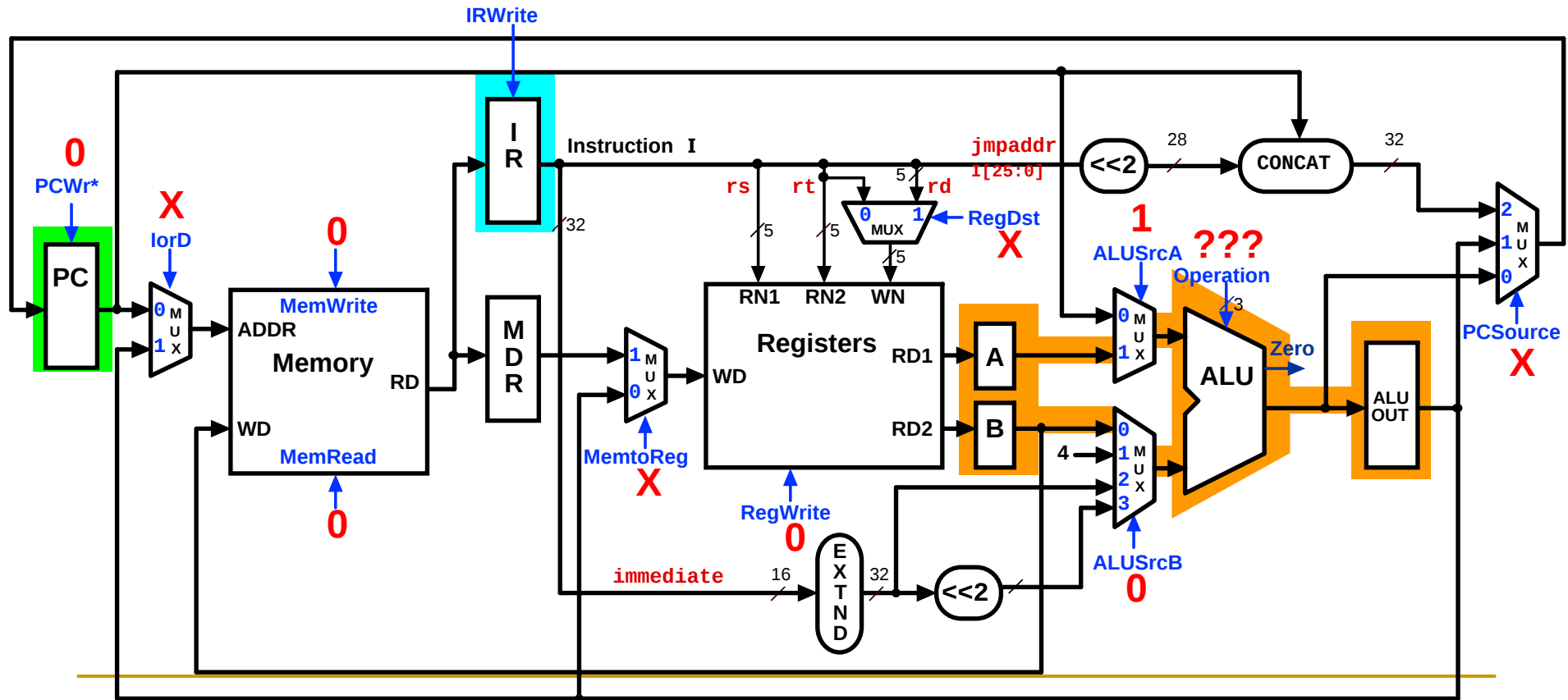
Passo 3 para referência à memória

$ALUOut = A + \text{sign-extend}(IR[15-0])$



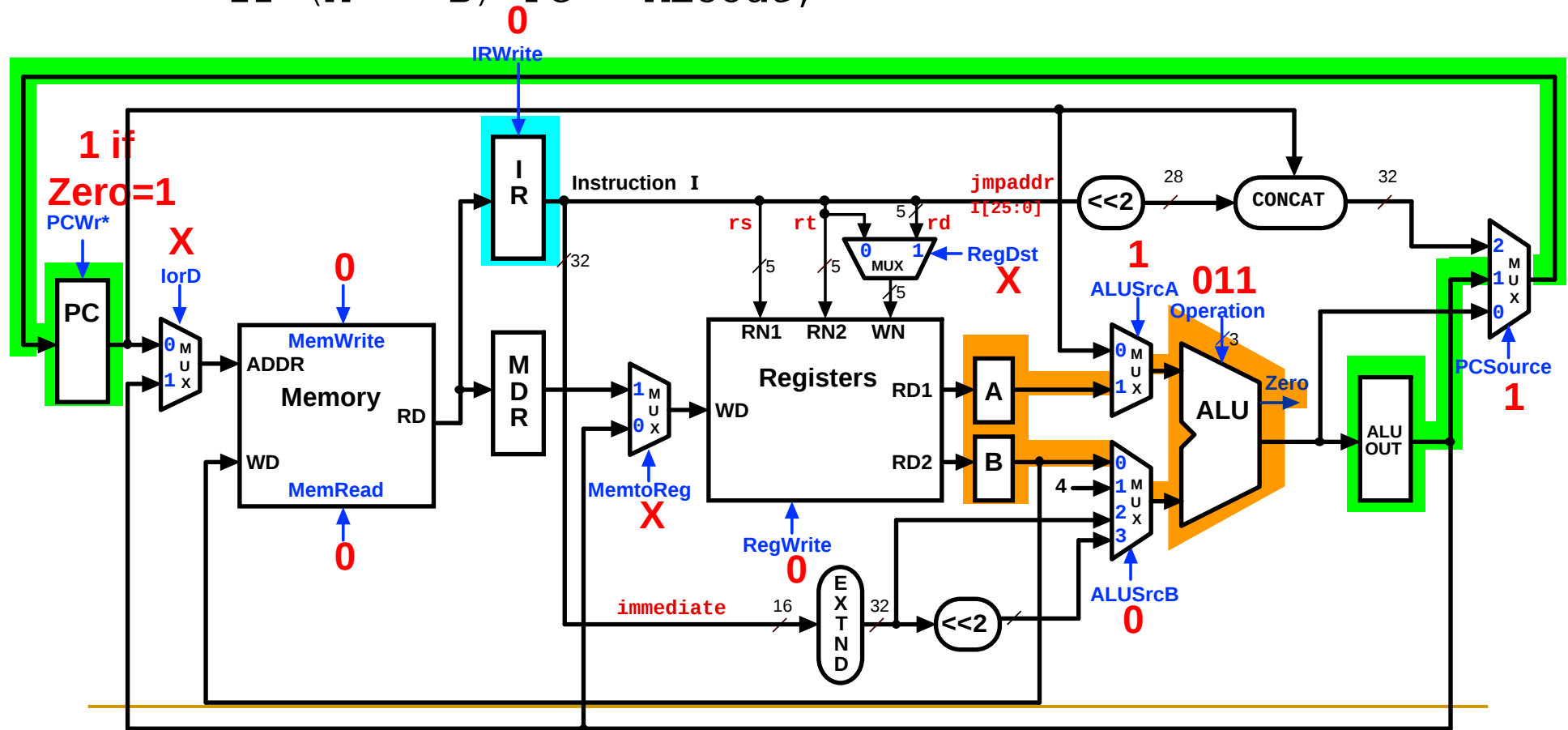
Passo 3 para instruções R com acesso à ALU

$$\text{ALUOut} = A \text{ op } B;$$



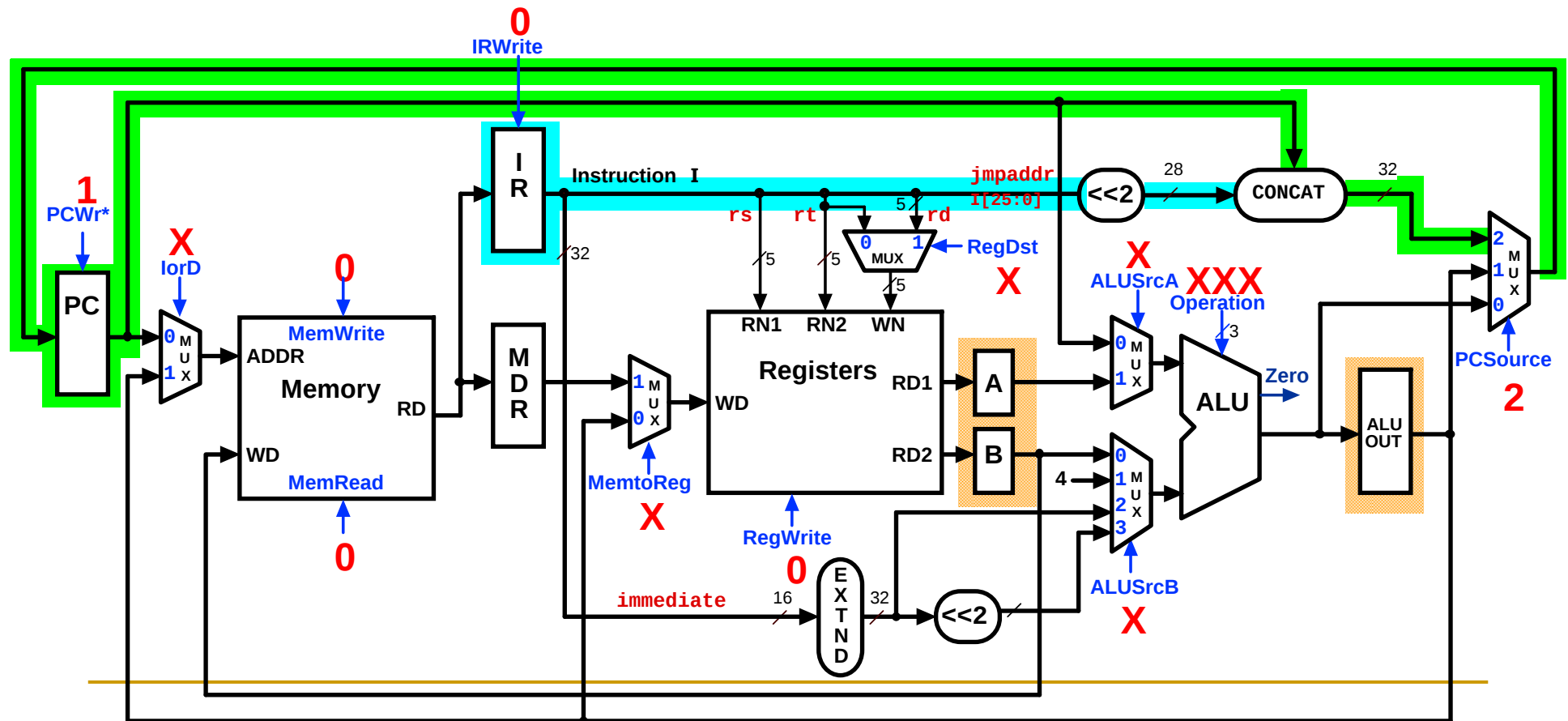
Passo 3 para instruções de desvio

```
if (A == B) PC = ALUOut;
```



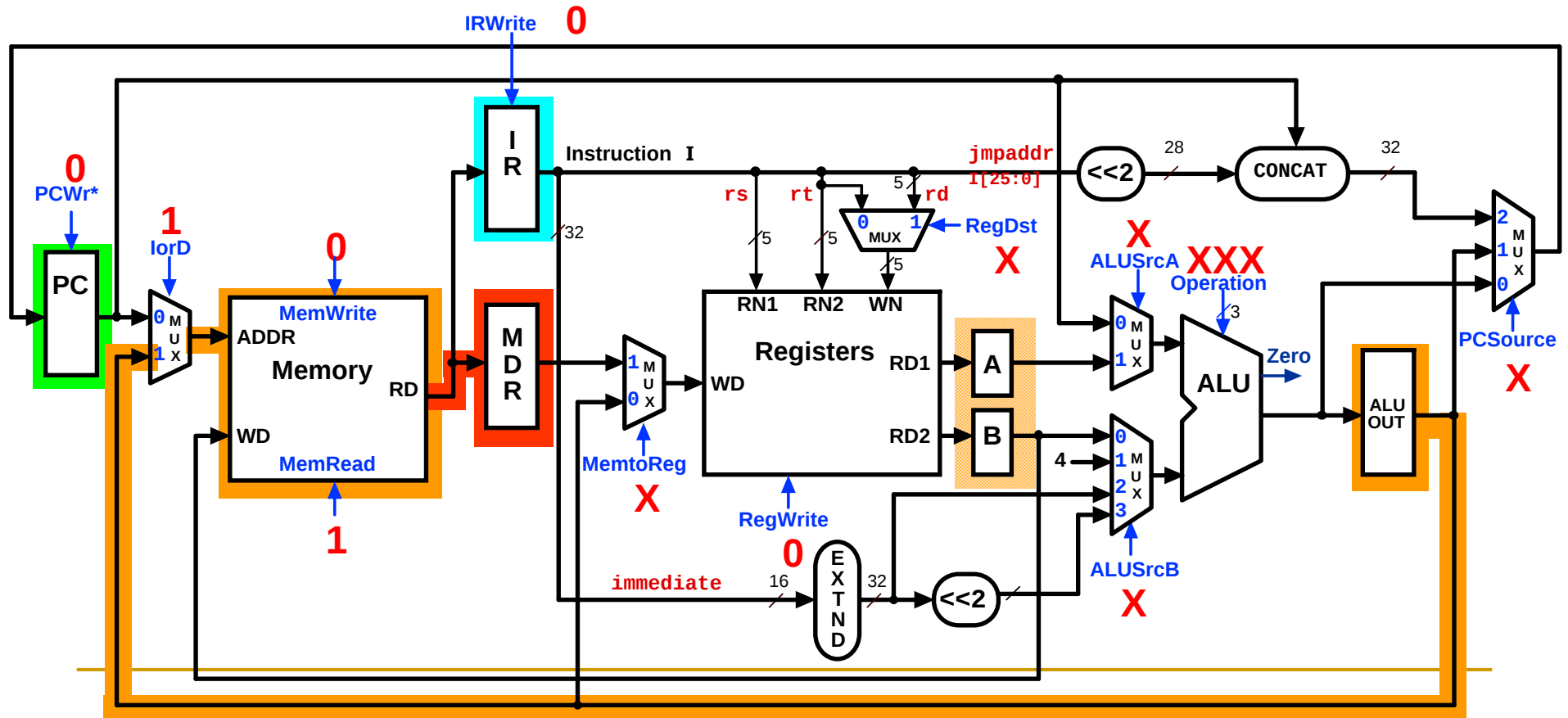
Passo 3 para instruções de salto

$PC = PC[21-28] \text{ concat } (IR[25-0] \ll 2);$



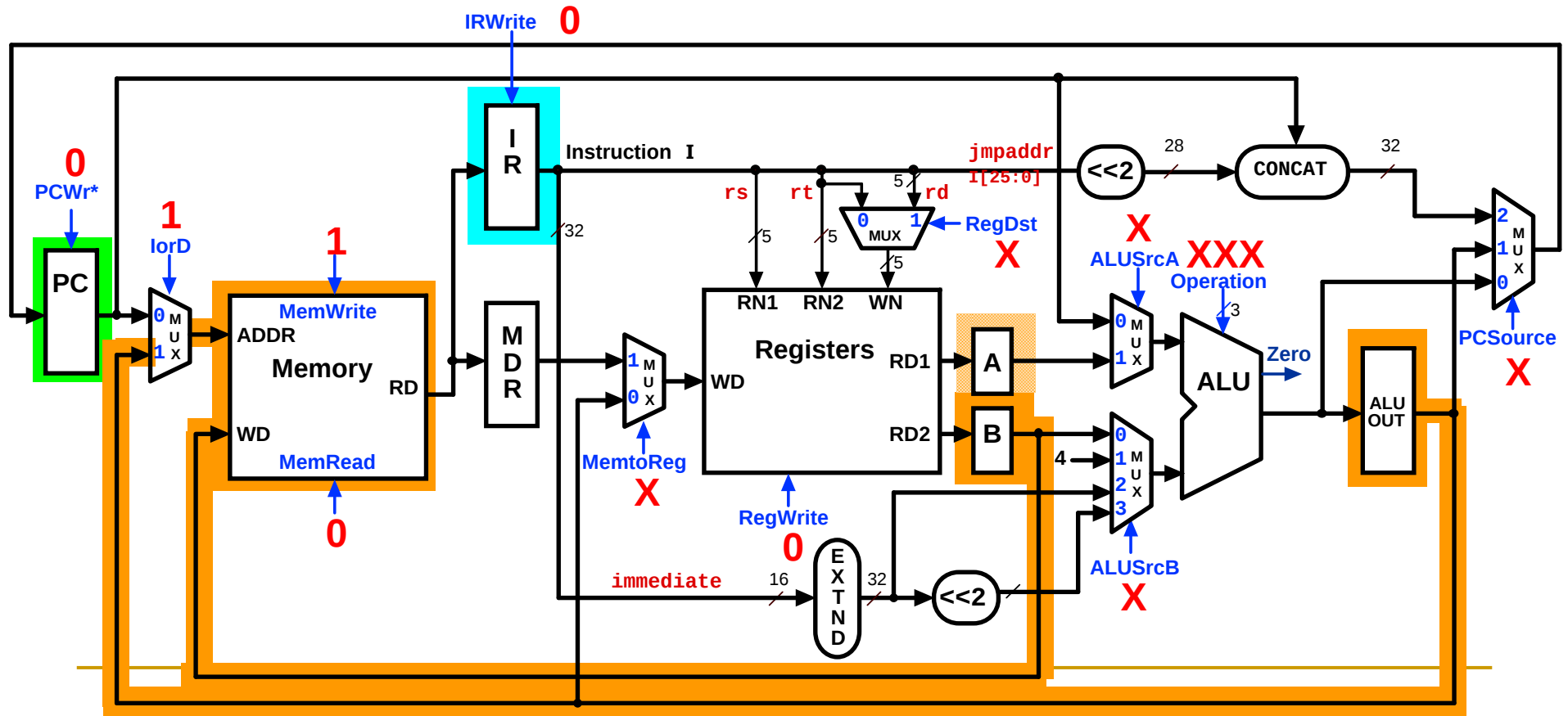
Passo 4 com acesso de leitura à memória (1w)

MDR = Memory[ALUOut];



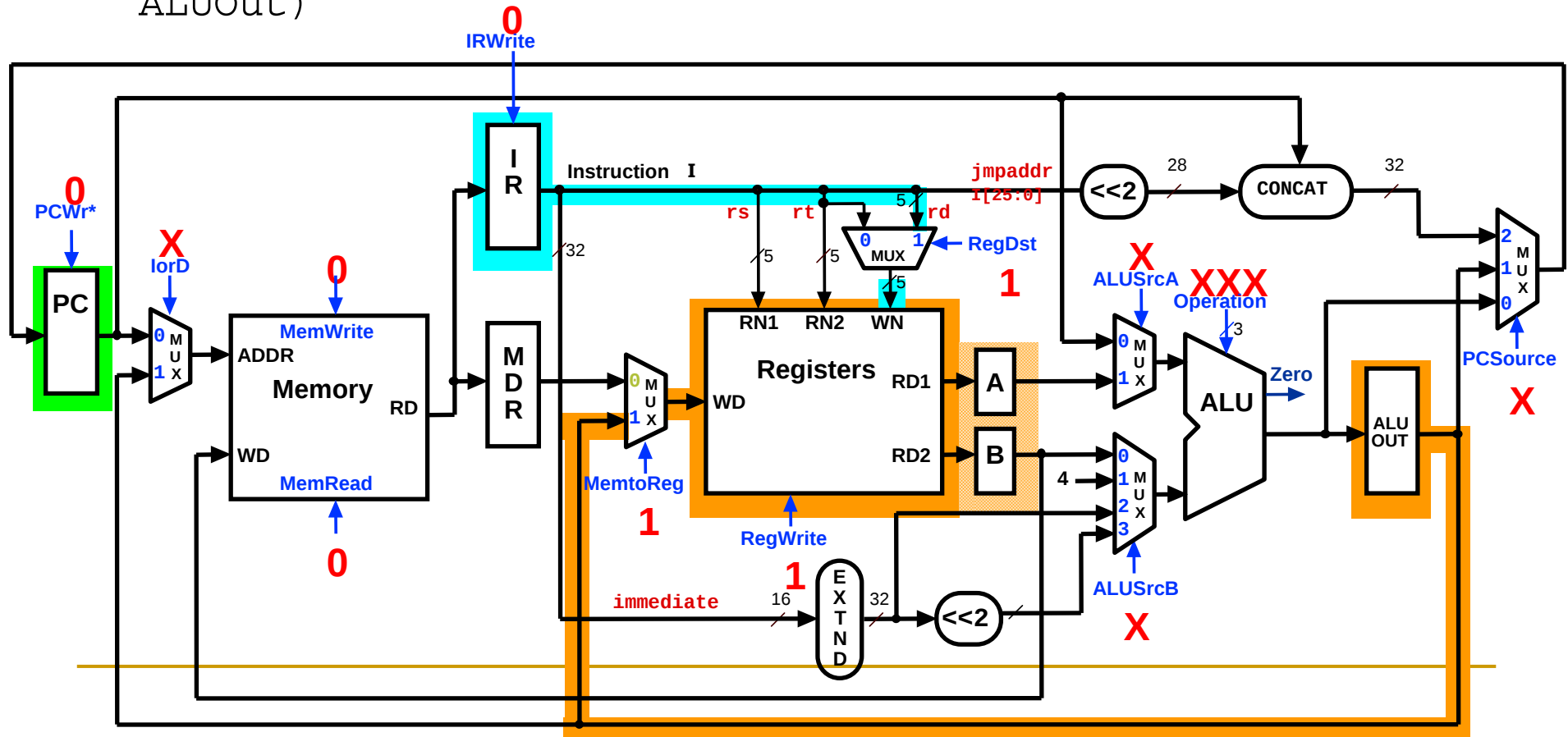
Passo 4 com acesso de escrita à memória (sw)

Memory[ALUOut] = B;



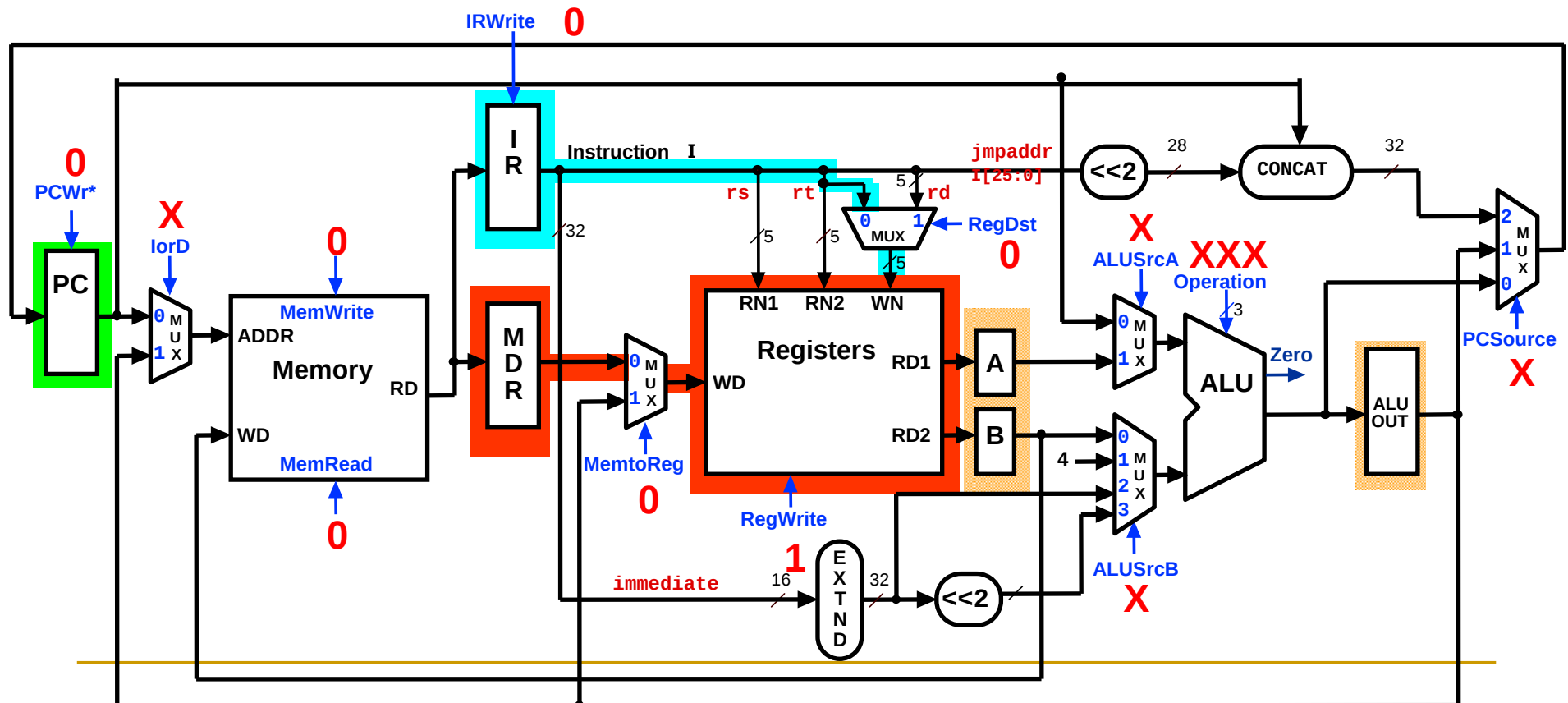
Passo 4 com acesso à ALU (tipo R)

$\text{Reg}[\text{IR}[15:11]] = \text{ALUOut};$ $(\text{Reg}[\text{Rd}] =$
 $\text{ALUOut})$



Passo 5 com finalização de leitura de memória (tipo R)(lw)

$\text{Reg}[\text{IR}[20-16]] = \text{MDR};$



Perguntas

- *Quantos ciclos serão utilizados pelo programa abaixo?*

```
        lw $t2, 0($t3)
        lw $t3, 4($t3)
        beq $t2, $t3, Label #assume not equal
        add $t5, $t2, $t3
        sw $t5, 8($t3)
Label:   ...
```

- *O que acontece no oitavo ciclo?*



- *Em que ciclo o conteúdo de \$t2 é somado ao de \$t3 ?*

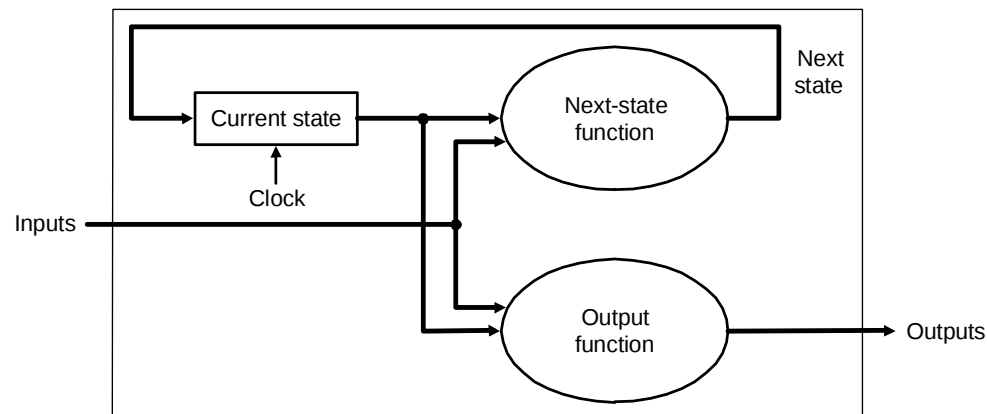
Implementando o controle

- Os valores dos sinais de controle dependem:
 - Qual instrução está sendo executada
 - Qual passo está sendo executado
 - Deve-se especificar uma máquina de estados utilizando
 - gráficos, ou
 - Microprogramação
 - A implementação é derivada da especificação
-

Máquinas de estado finito

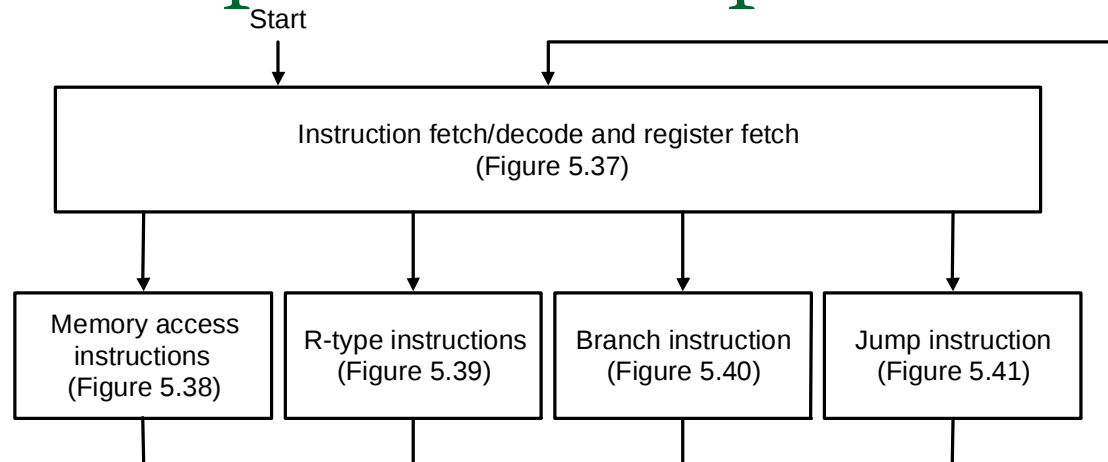
■ Máquinas de estado finito (FSMs):

- Um conjunto de estados e
- Função de próximo estado, determinada pelo estado corrente e entrada
- Função de saída, determinada pelo estado corrente e possivelmente entrada



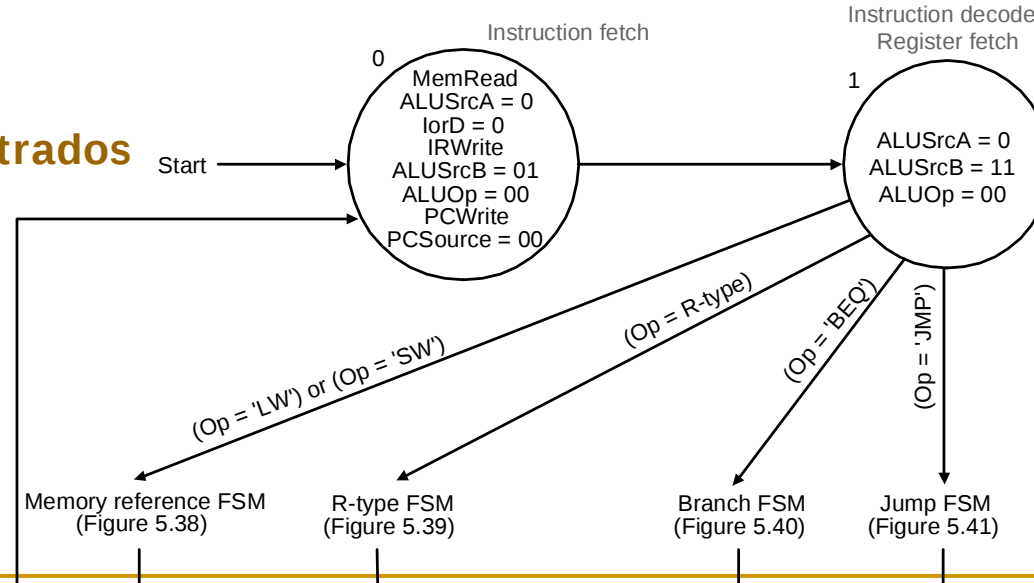
- *Máquina de Moore*— saída baseada somente no estado corrente

Controle representado por FSM



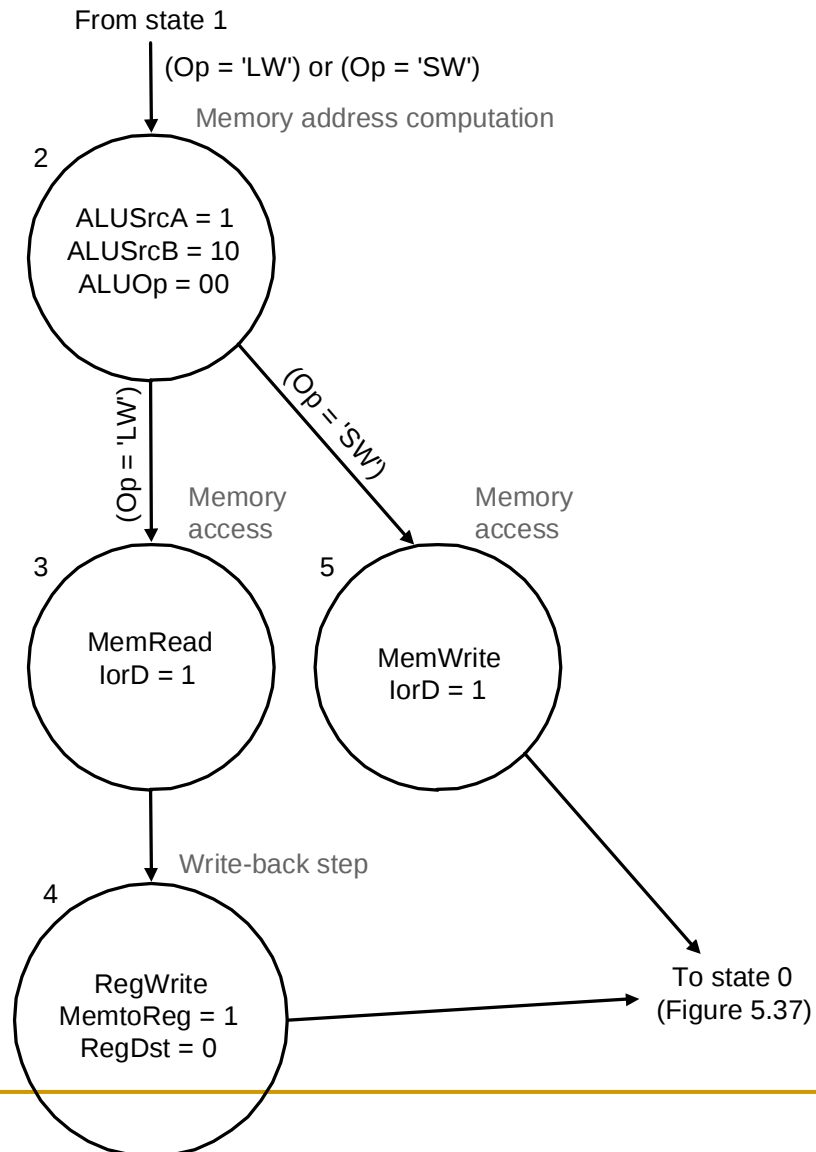
Visão de alto nível

Sinais são mostrados nos círculos



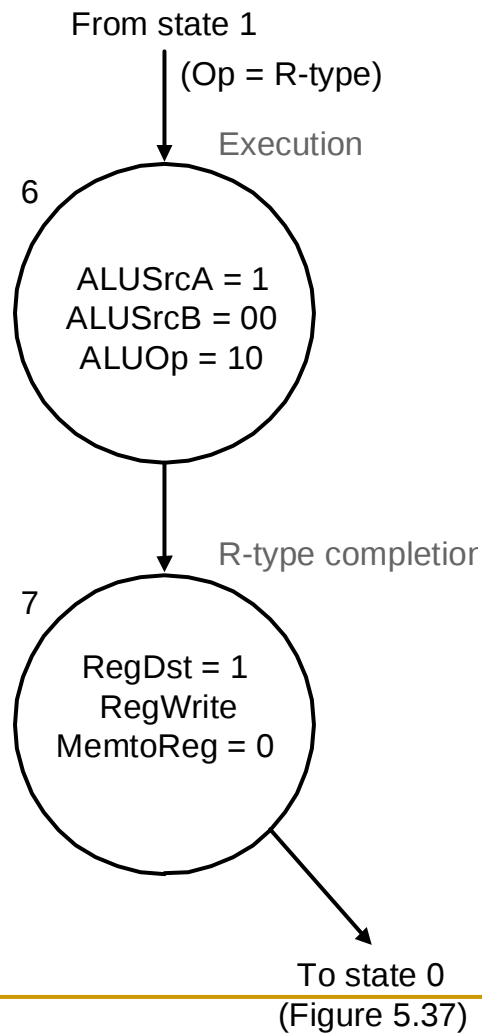
Os passos de busca e decodificação de instrução são idênticos para todas instruções

Referência à memória



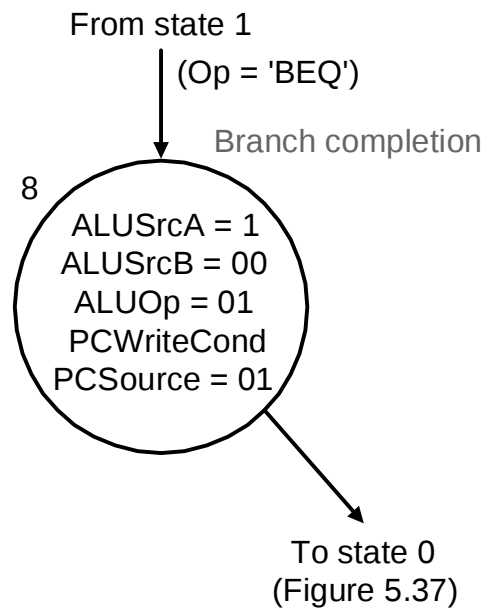
Possui 4 estados

Instrução do tipo R



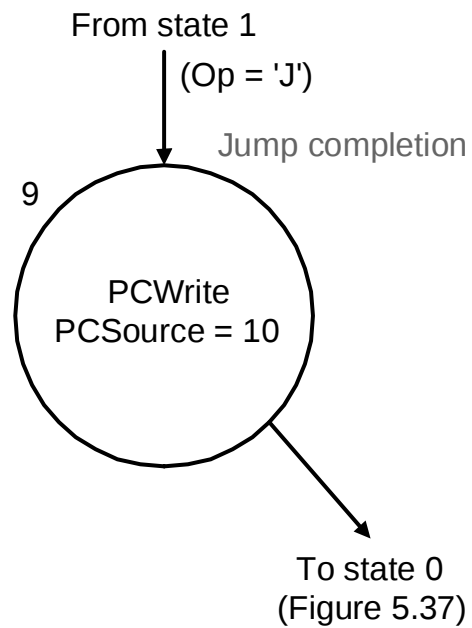
Possui 2 estados

Instrução de desvio



Possui 1 estado

Instrução de salto



Possui 1 estado

Controle FSM

