cin.ufpe.br

Centro de Informática

U · F · P · E

Universidade Federal de Pernambuco

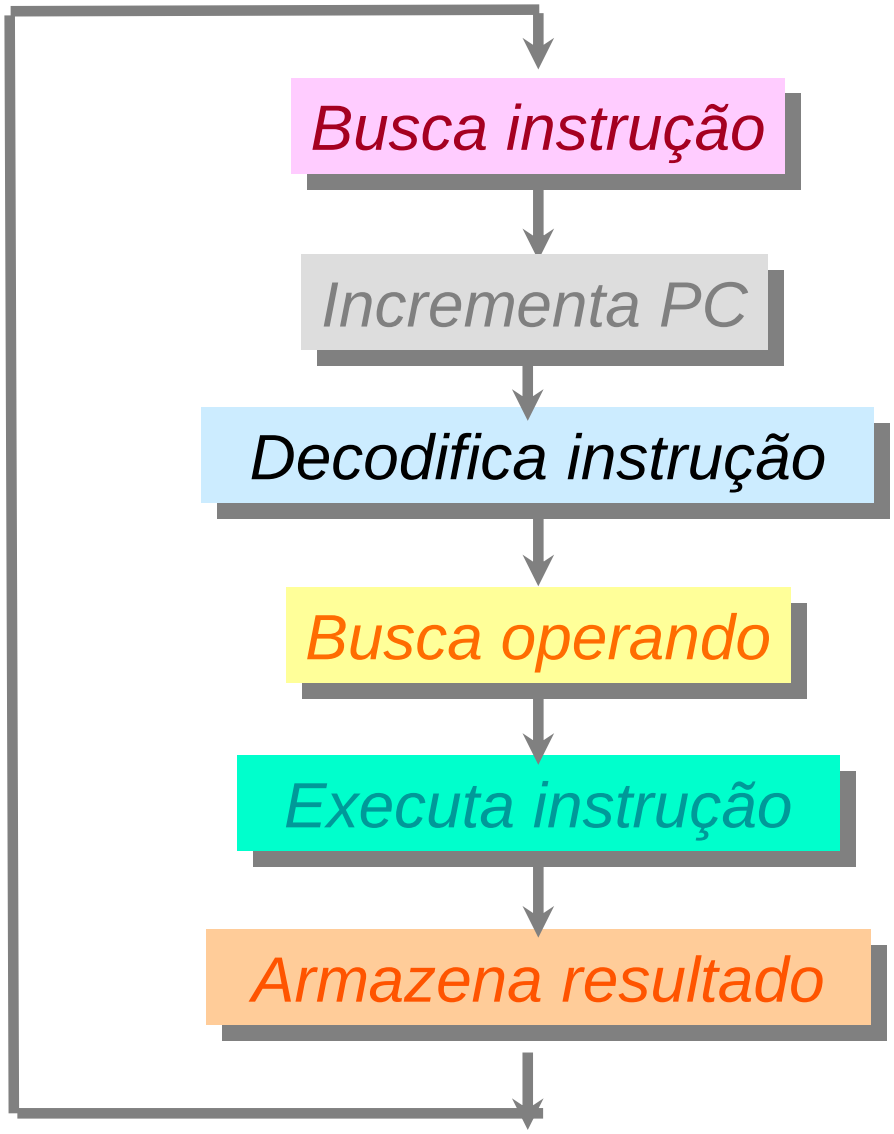
CPU: Estrutura e Funcionalidade

Implementação Multi-ciclo

Roteiro da Aula

- Projeto de uma CPU simples
- Unidade de Processamento – Via de Dados
  - Compartilhamento de unidades funcionais e memória
  - Registradores adicionais
  - Multiplexes
- Implementação Multi-ciclo
  - Unidade de controle
    - Leitura da Instrução
    - Operações Aritméticas
    - Leitura + Operação entre registradores
    - Acesso à Memória
    - Desvio Condicional
  - Máquina de Estado
- Exceções
  - Instrução indefinida
  - overflow

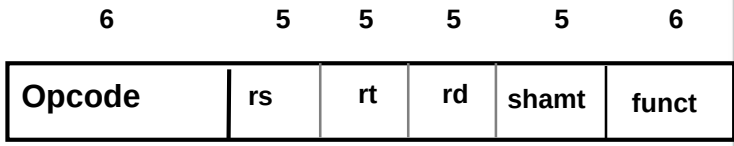
Ciclo de Instrução



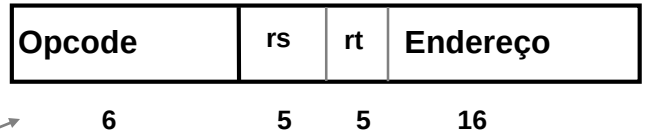
Instruções

- Tipos de instruções
  - Processamento:
    - aritméticas e lógicas
  - Armazenamento
  - E/S
  - Controle:
    - teste e desvio

F1



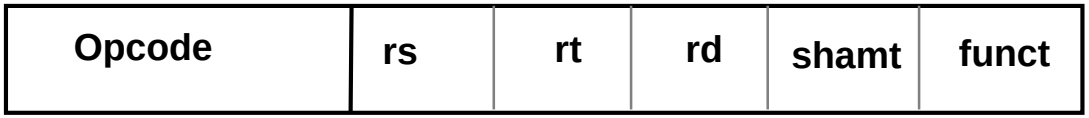
F2



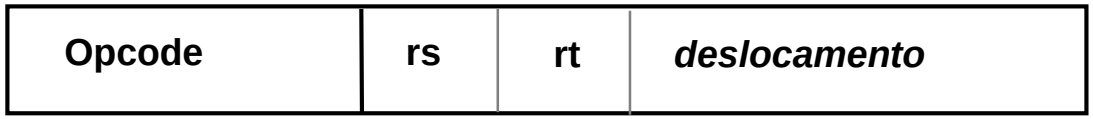
Projeto: uma CPU simples...

Instrução	Descrição
LW rt, desl(rs)	Carrega palavra de mem em rs
SW rt, desl(rs)	Armaz. Reg. na memória
ADD rd, rs, rt	rd <- rs + rt
SUB rd, rs, rt	rd <- rs – rt
AND rd, rs, rt	rd <- rs and rt
BEQ rs, rt, end	Desvio se rs = rt
J end	PC = end

Aritm

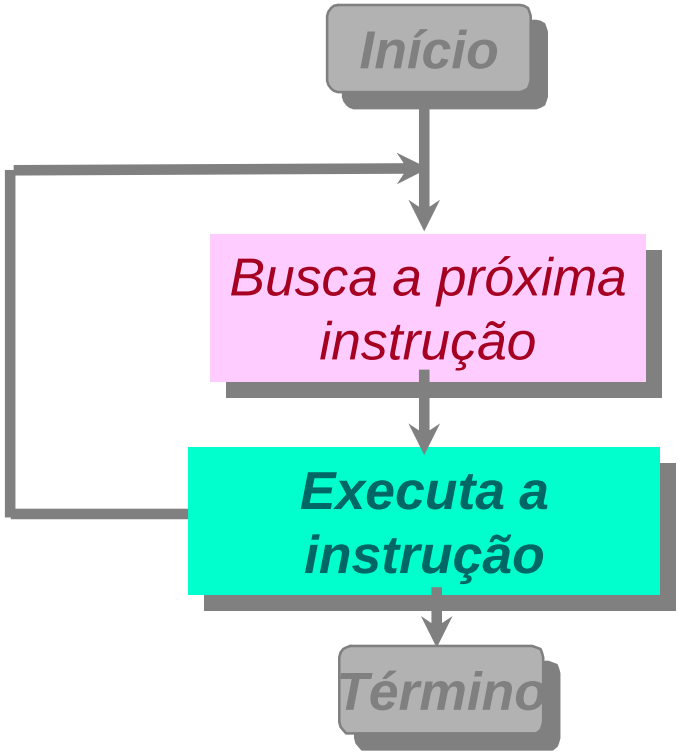


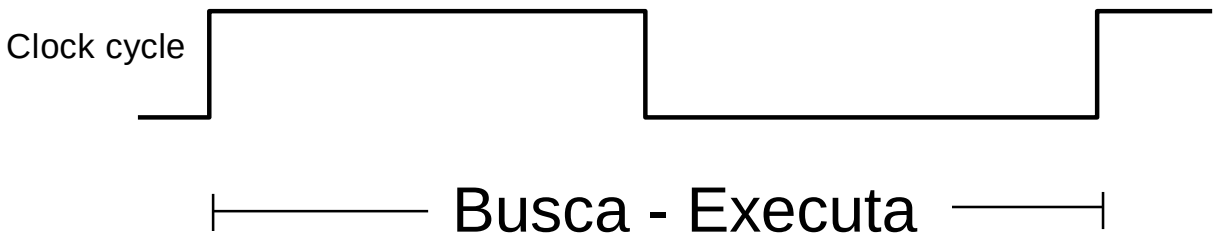
lw/sw e beq



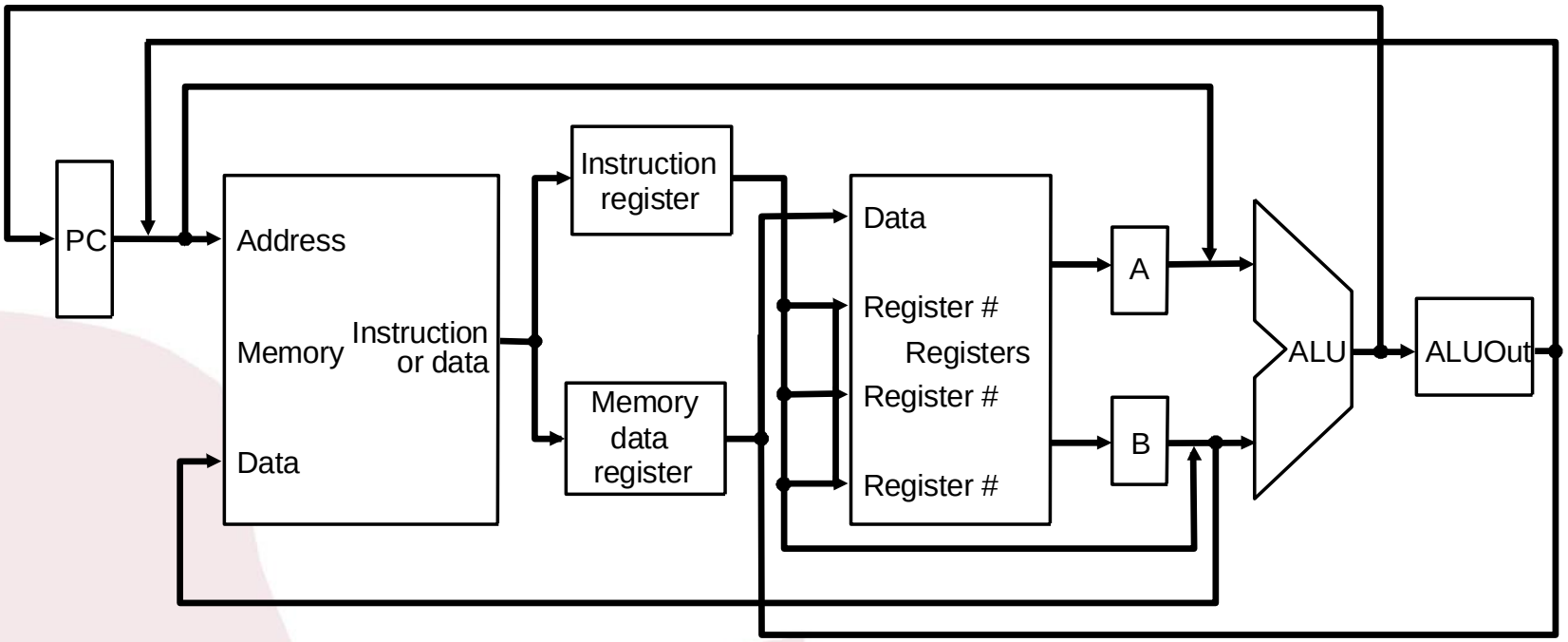
Jump

Mono-ciclo

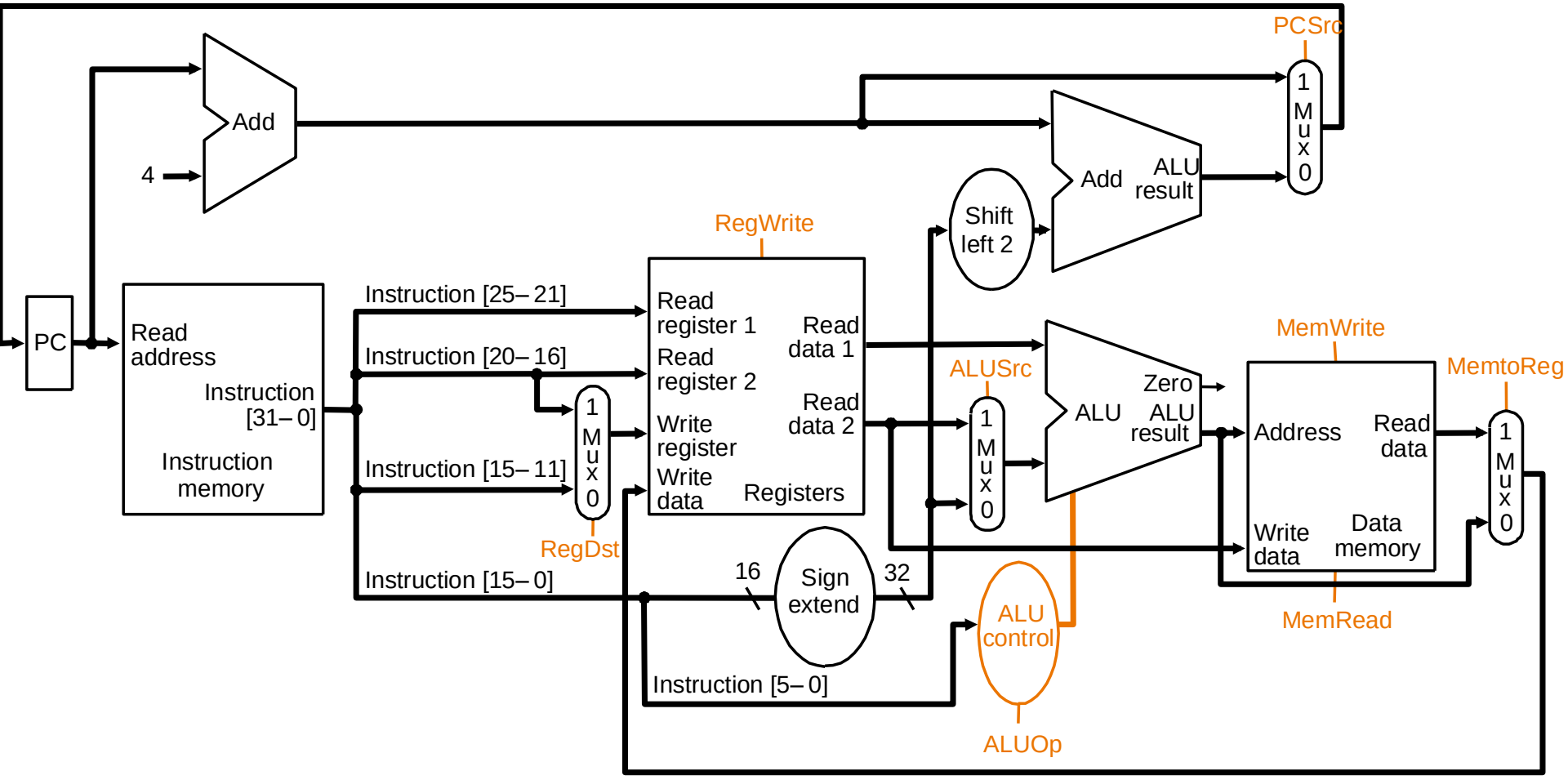

Início
Busca a próxima instrução
Executa a instrução
Término


Clock cycle
Busca - Executa

Implementação Multi-ciclo

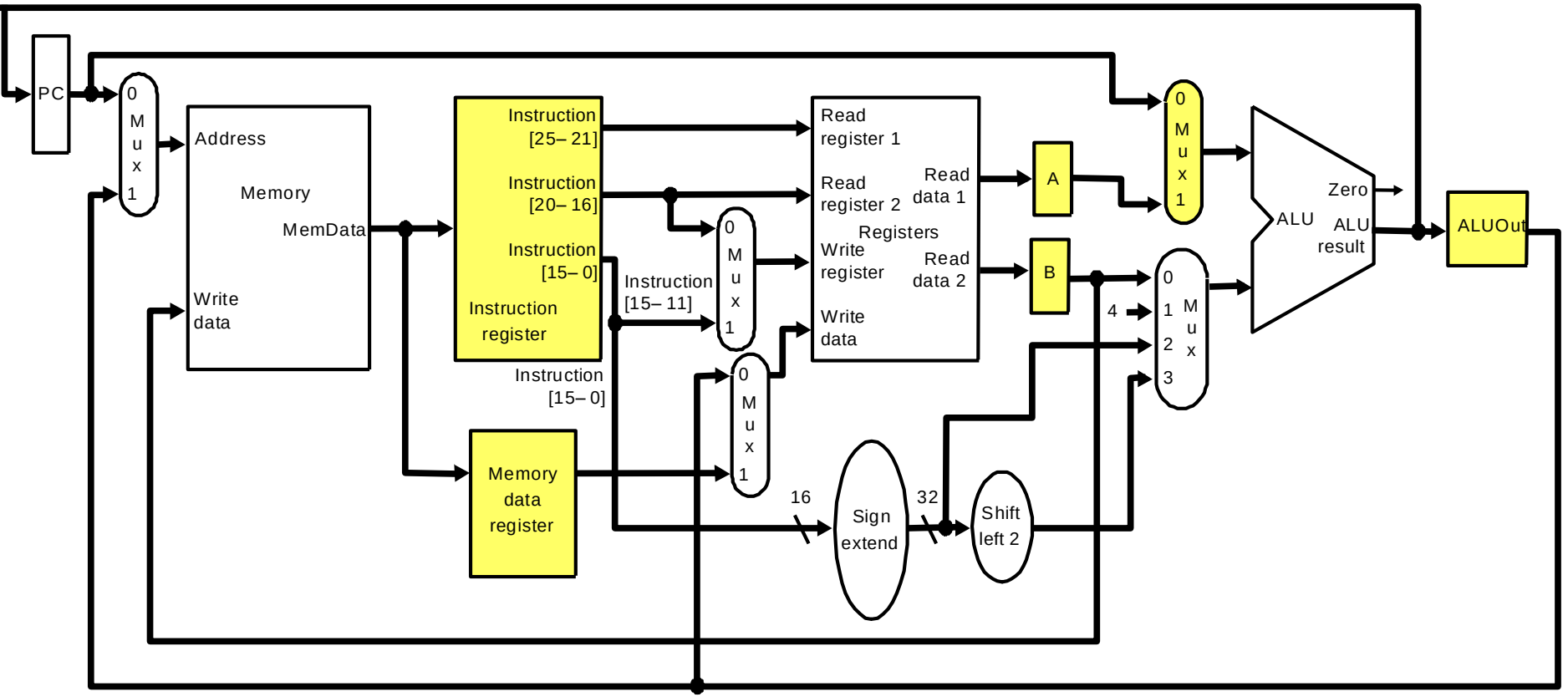


Implementação Mono-ciclo



Implementação Multi-ciclo

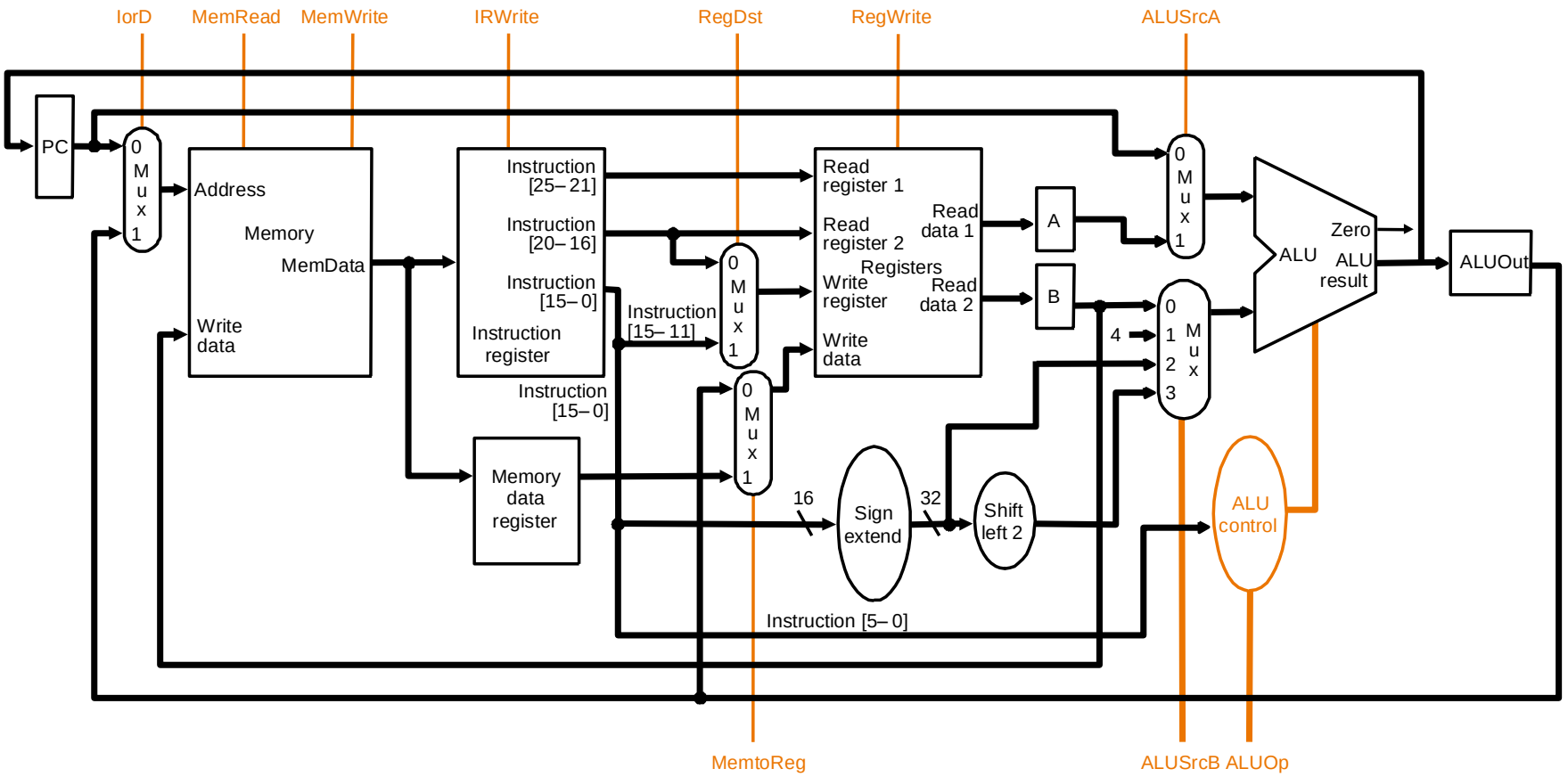
Unidade de Processamento – Via de Dados



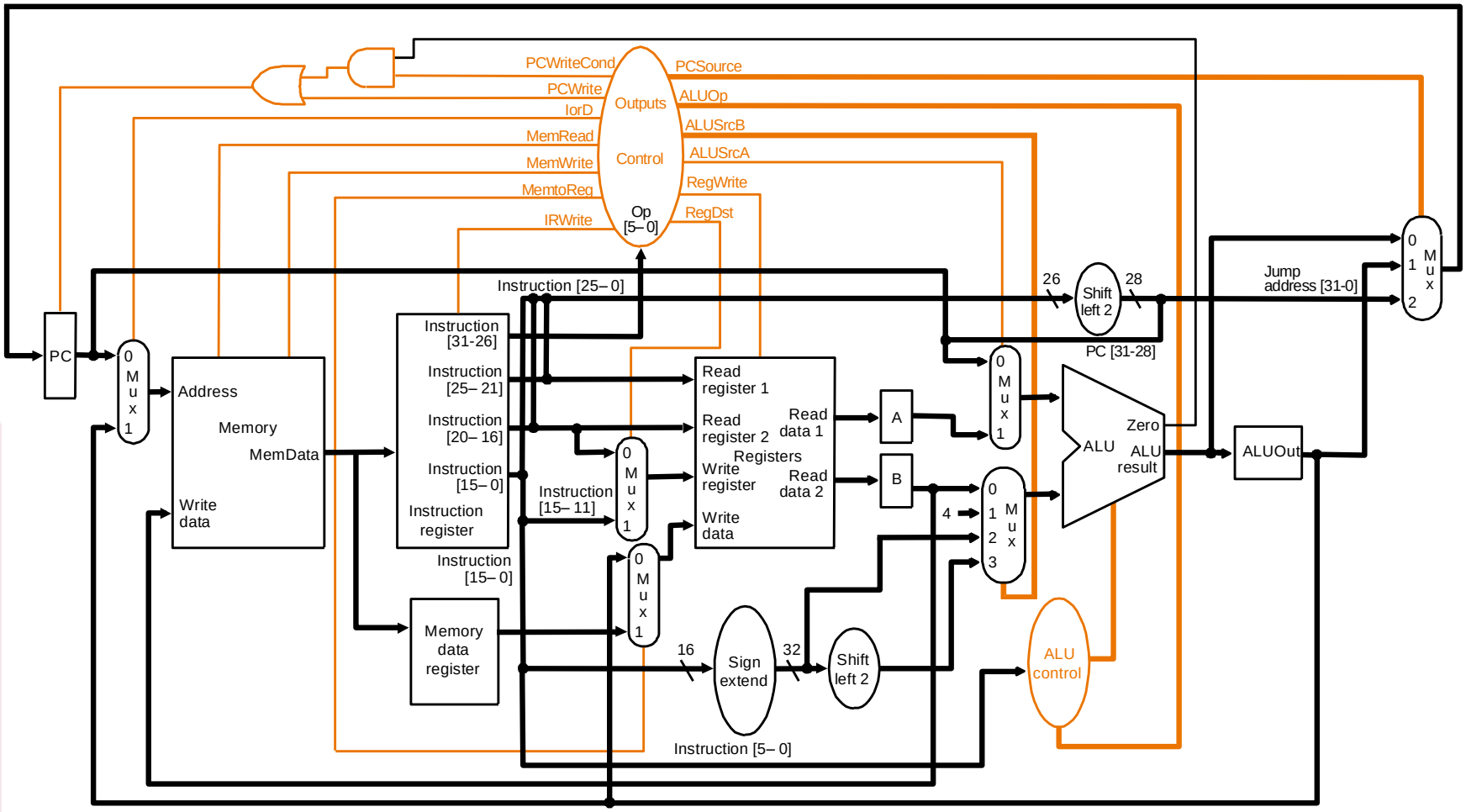
Implementação Multi-ciclo

Centro de Informática U·F·P·E

Sinais de Controle



Implementação Multi-ciclo



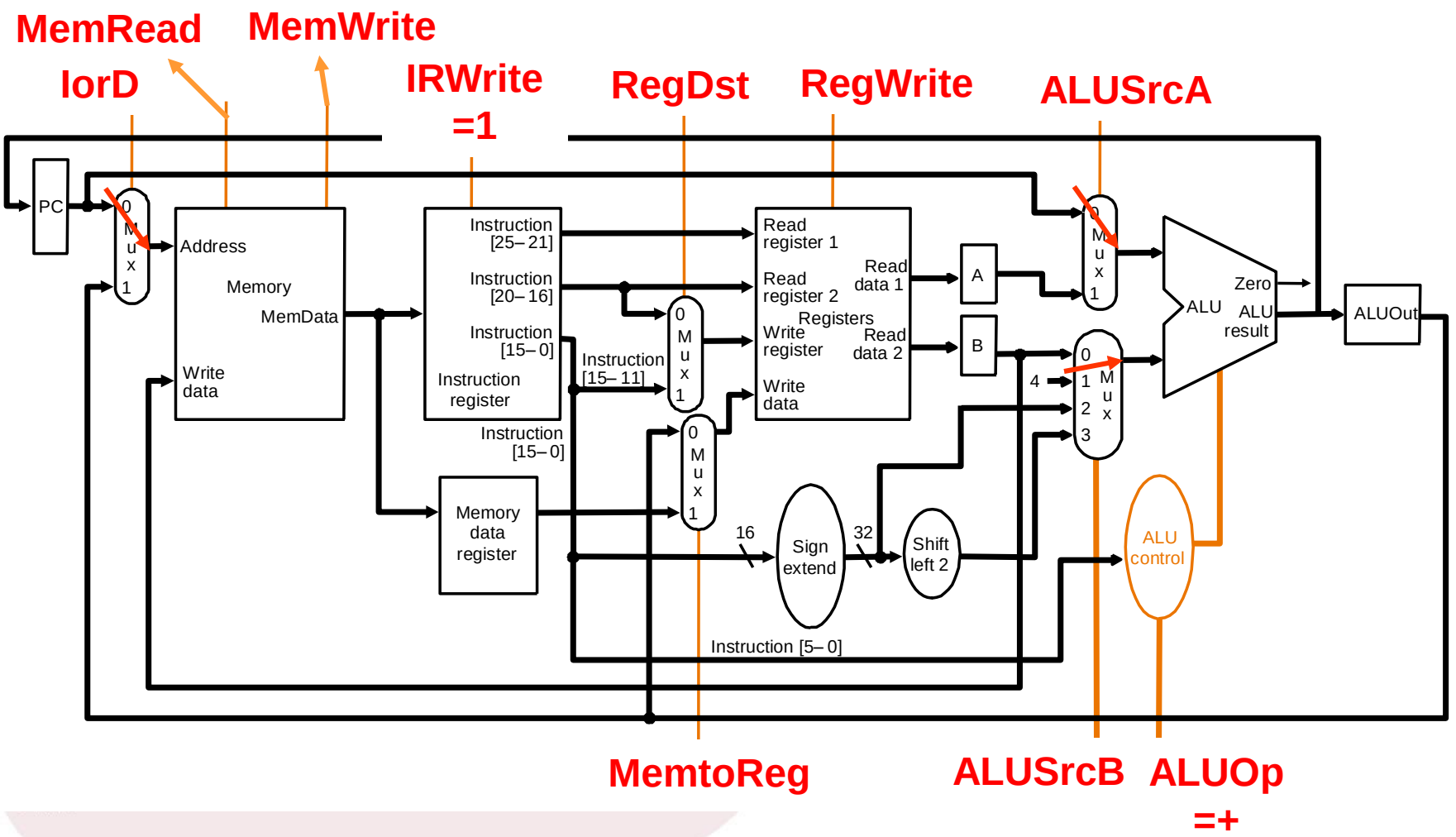
Implementando em ciclos

- Busca de Instrução
  - IR= Memória(PC)
  - PC = PC + 4

Implem

Busca de Instrução

- *IR= Memória(PC), PC = PC + 4*

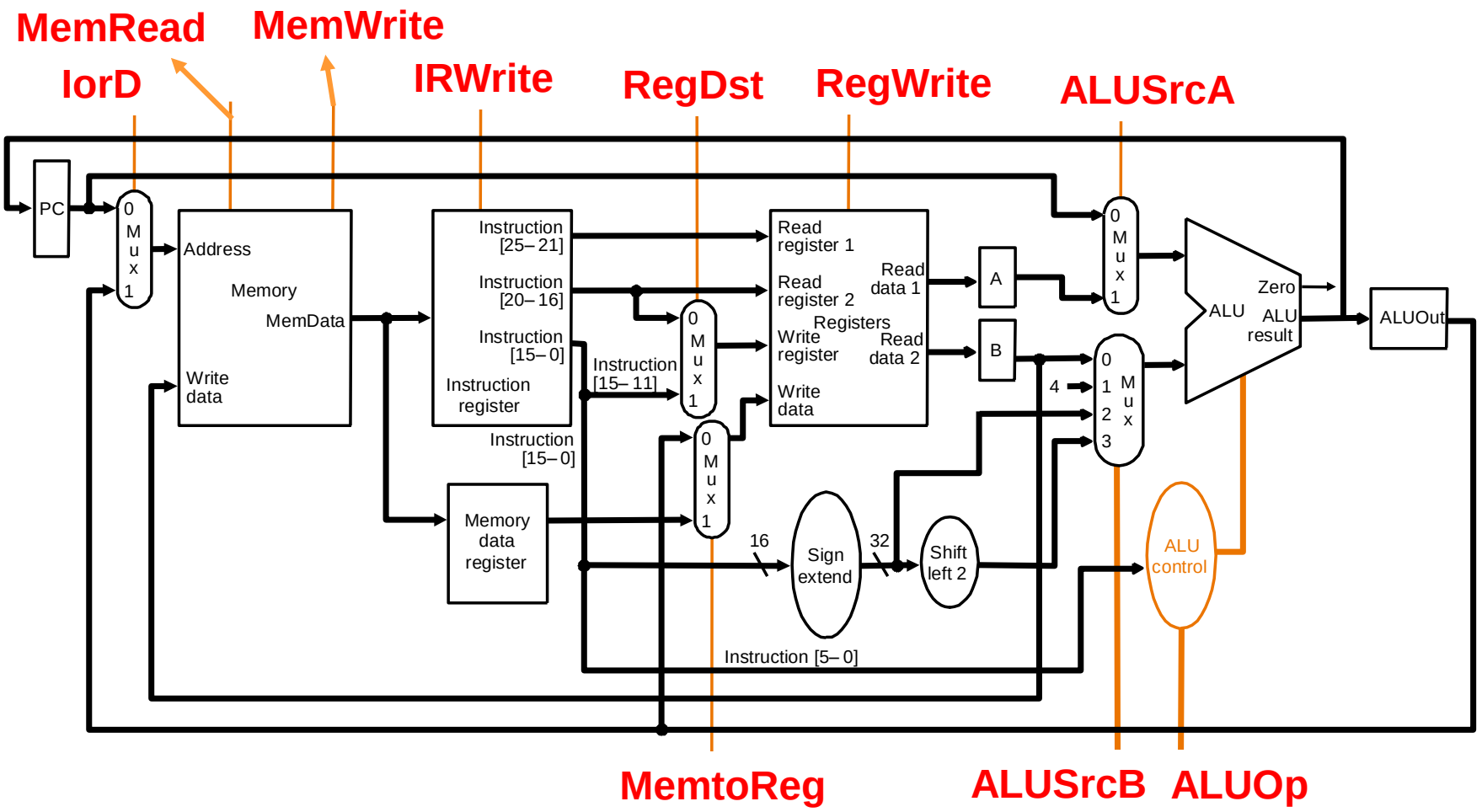


Implementando em ciclos

- Busca de Instrução
  - IR= Memória(PC)
  - PC = PC + 4
- Decodificação e leitura registradores
  - A= Reg(IR(25-21))
  - B= Reg(IR(20-16))
  - ALUout = PC + (ext-sinal(IR(15-0)<<2))

Implem

- *Decodificação e leitura registradores*
  - *A= Reg(IR(25-21)), B= Reg(IR(20-16))*



Implementação Multi-ciclo

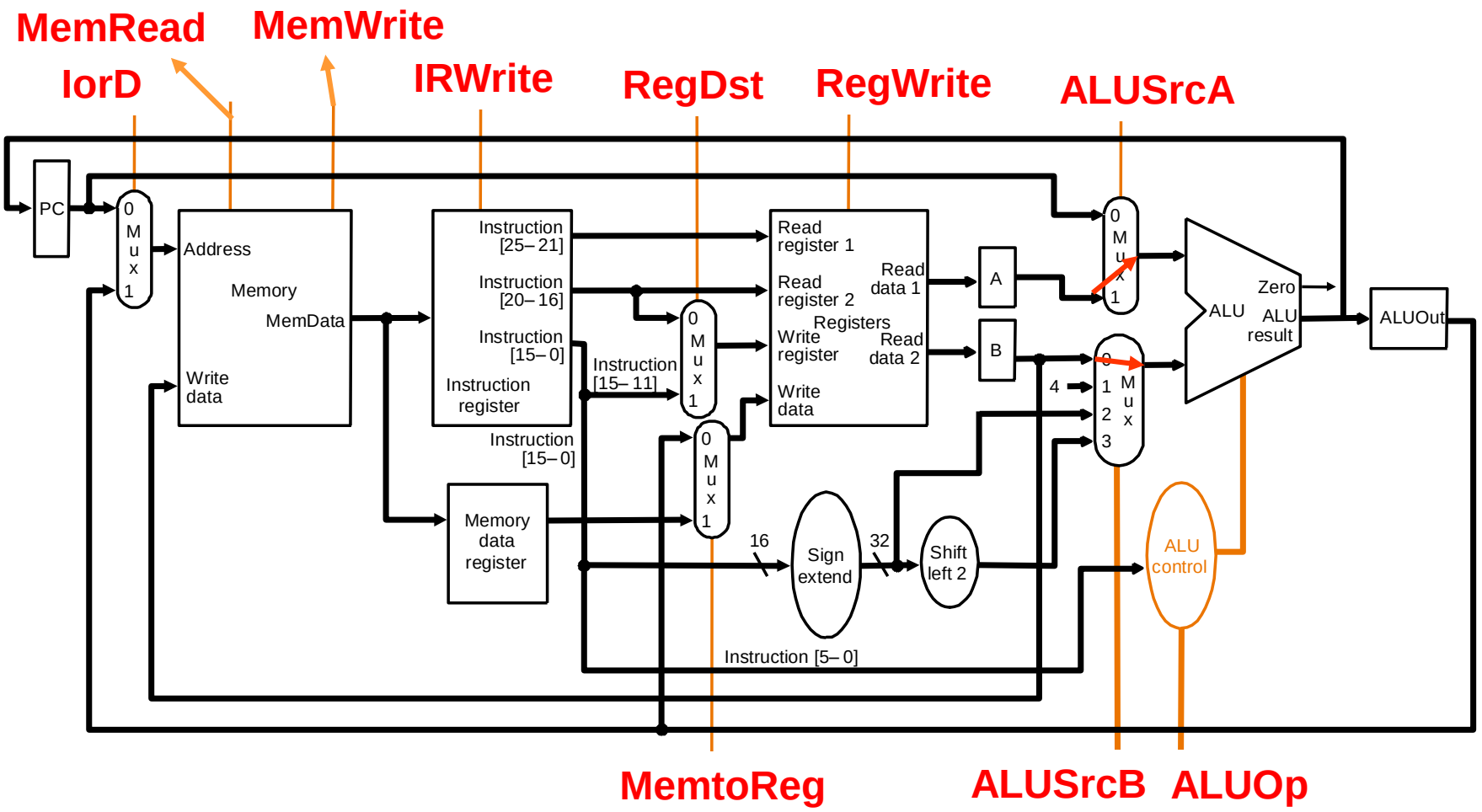
- Load/Store
  - ALUout = A + ext-sinal(IR(15-0))
  - Load: MDR = Memória(ALUout)
  - Reg(IR(20-16)) = MDR
  - Store: Memória(ALUout) = B

ação Multi-ciclo

Load/Store

–ALUout = A +

ext-sinal(IR(15-0))

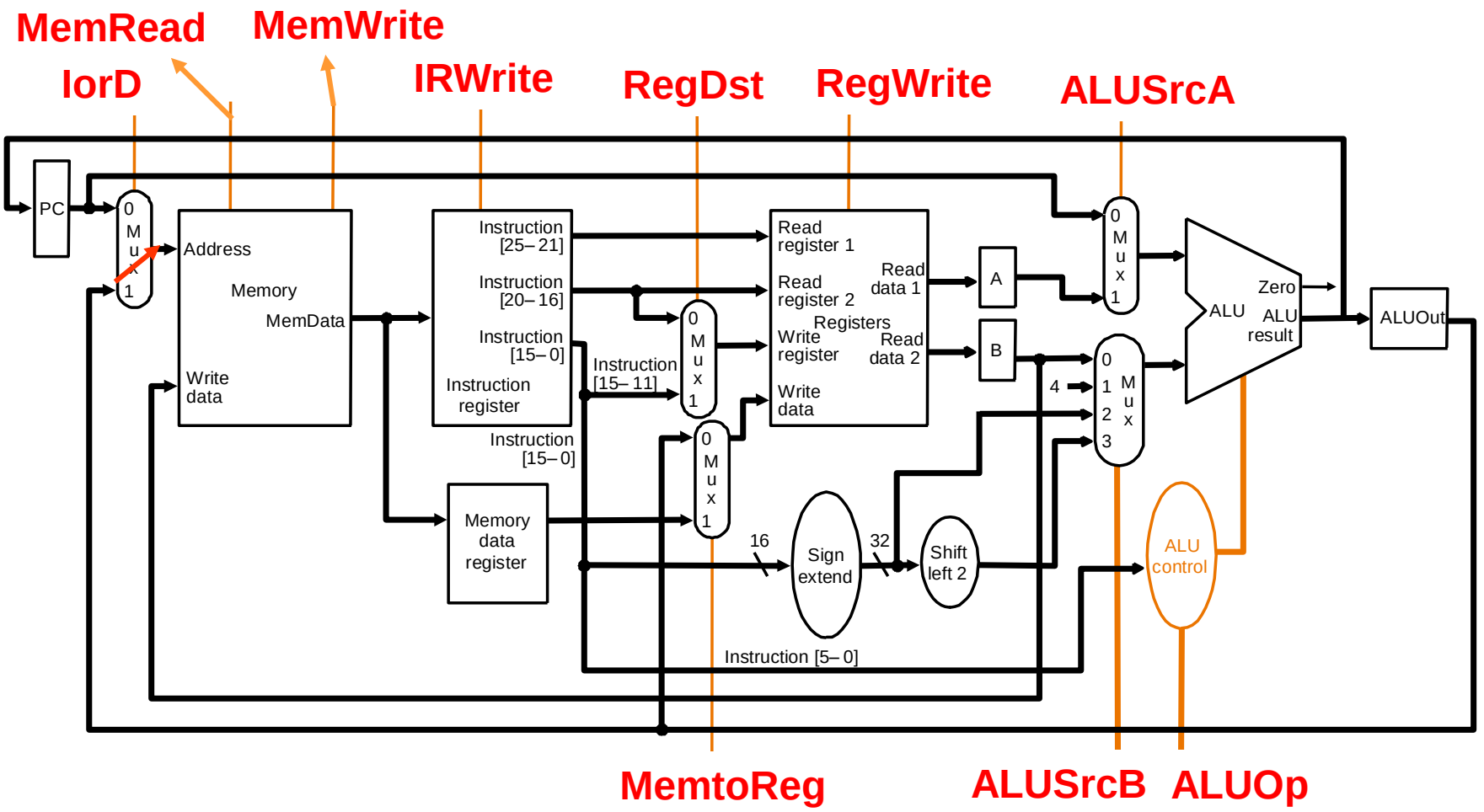


Implementação Multi-ciclo

- Load/Store
  - ALUout = A + ext-sinal(IR(15-0))
  - Load: MDR = Memória(ALUout)
  - Reg(IR(20-16)) = MDR
  - Store: Memória(ALUout) = B

Implementaç

- *Load/Store*
  - *–Load: MDR = Memória(ALUout)*
  - *–Store: Memória(ALUout) = B*

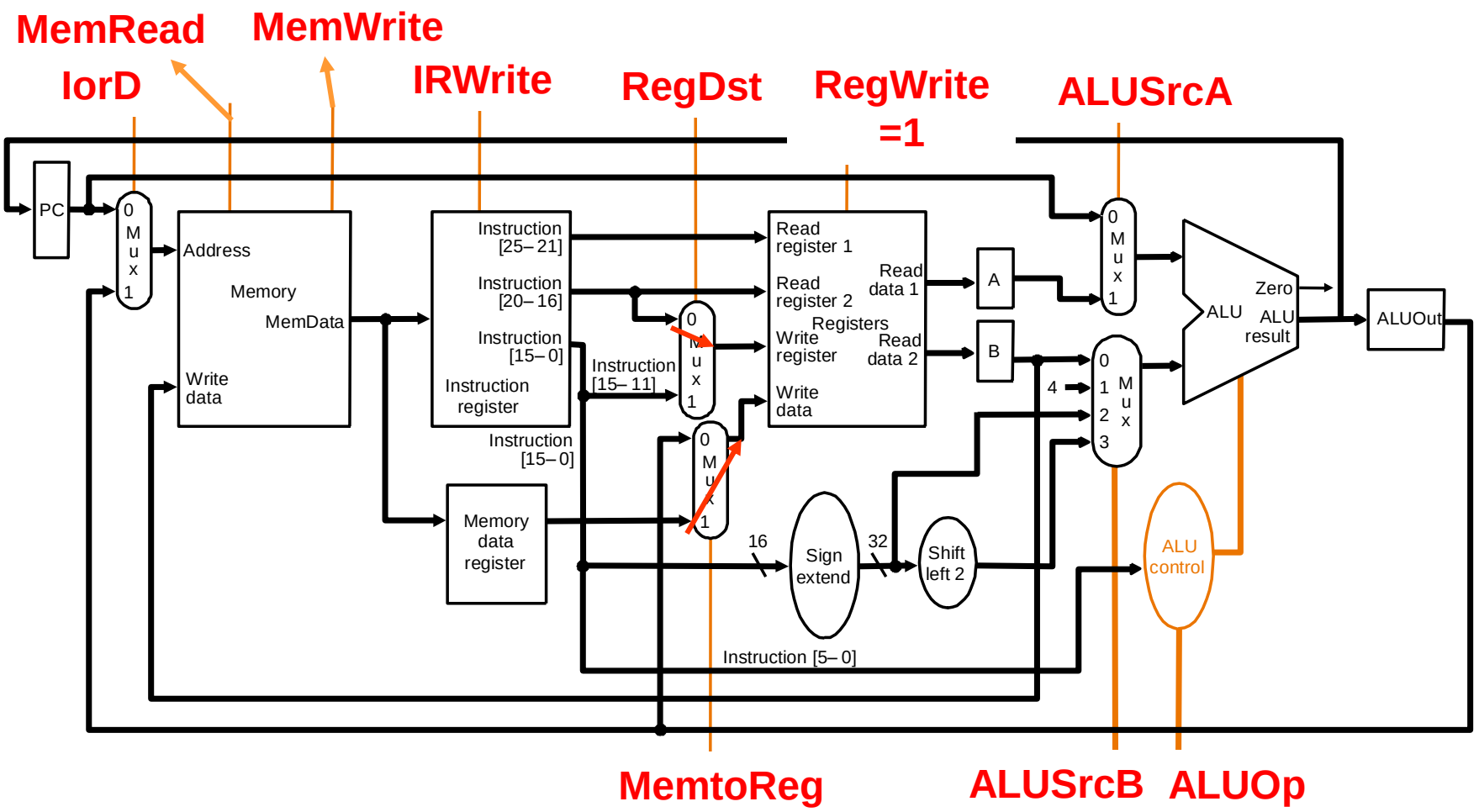


Implementação Multi-ciclo

- Load/Store
  - ALUout = A + ext-sinal(IR(15-0))
  - Load: MDR = Memória(ALUout)
  - Reg(IR(20-16)) = MDR
  - Store: Memória(ALUout) = B

Implementação Multi-ciclo

- *Load*
  - *–Reg(IR(20-16)) = MDR*

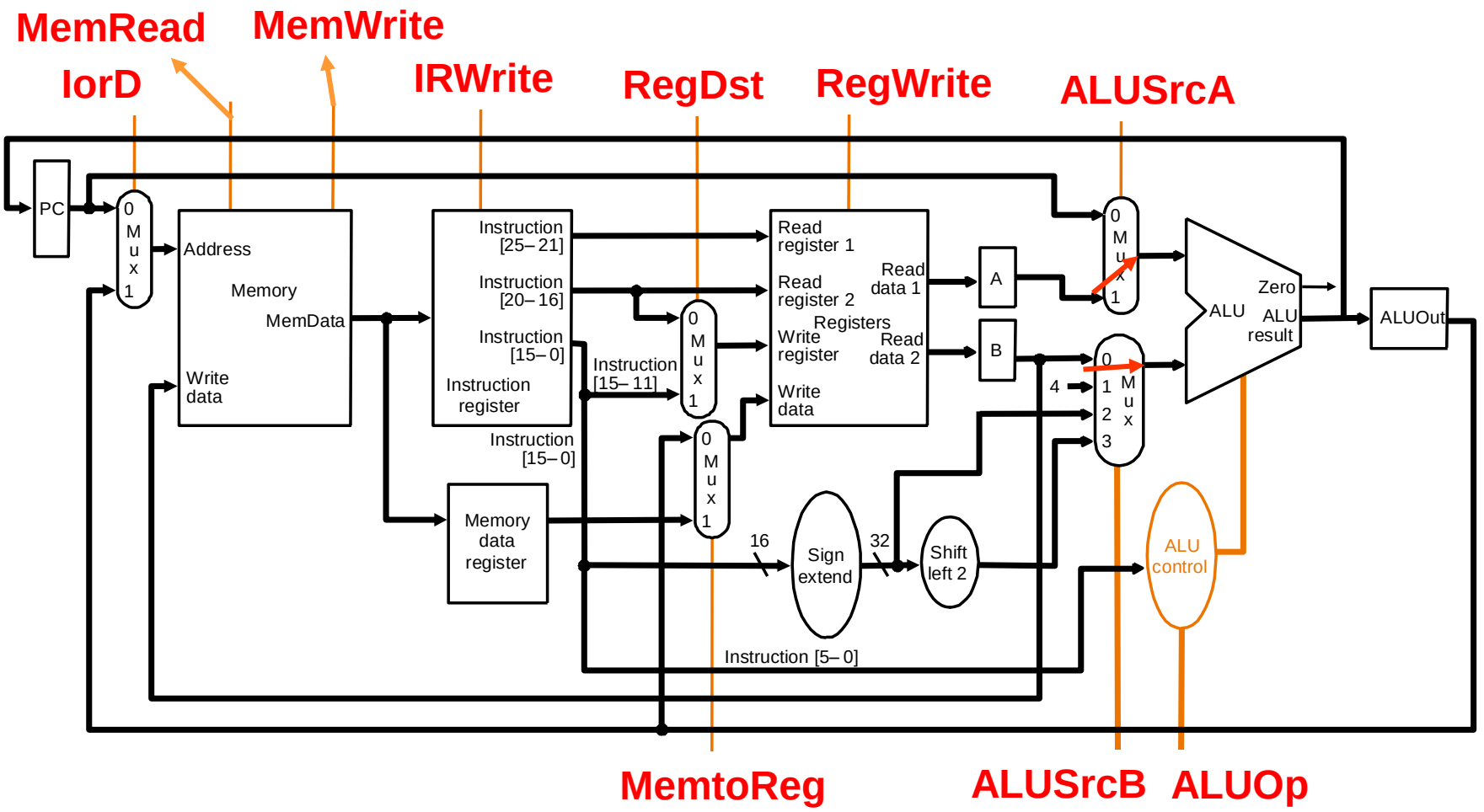


Implementação Multi-ciclo

- Aritmética
  - ALUout = A op B
  - Reg(IR(15-11)) = ALUout

ação Multi-ciclo

- *Aritmética*
  - *–ALUout = A op B*

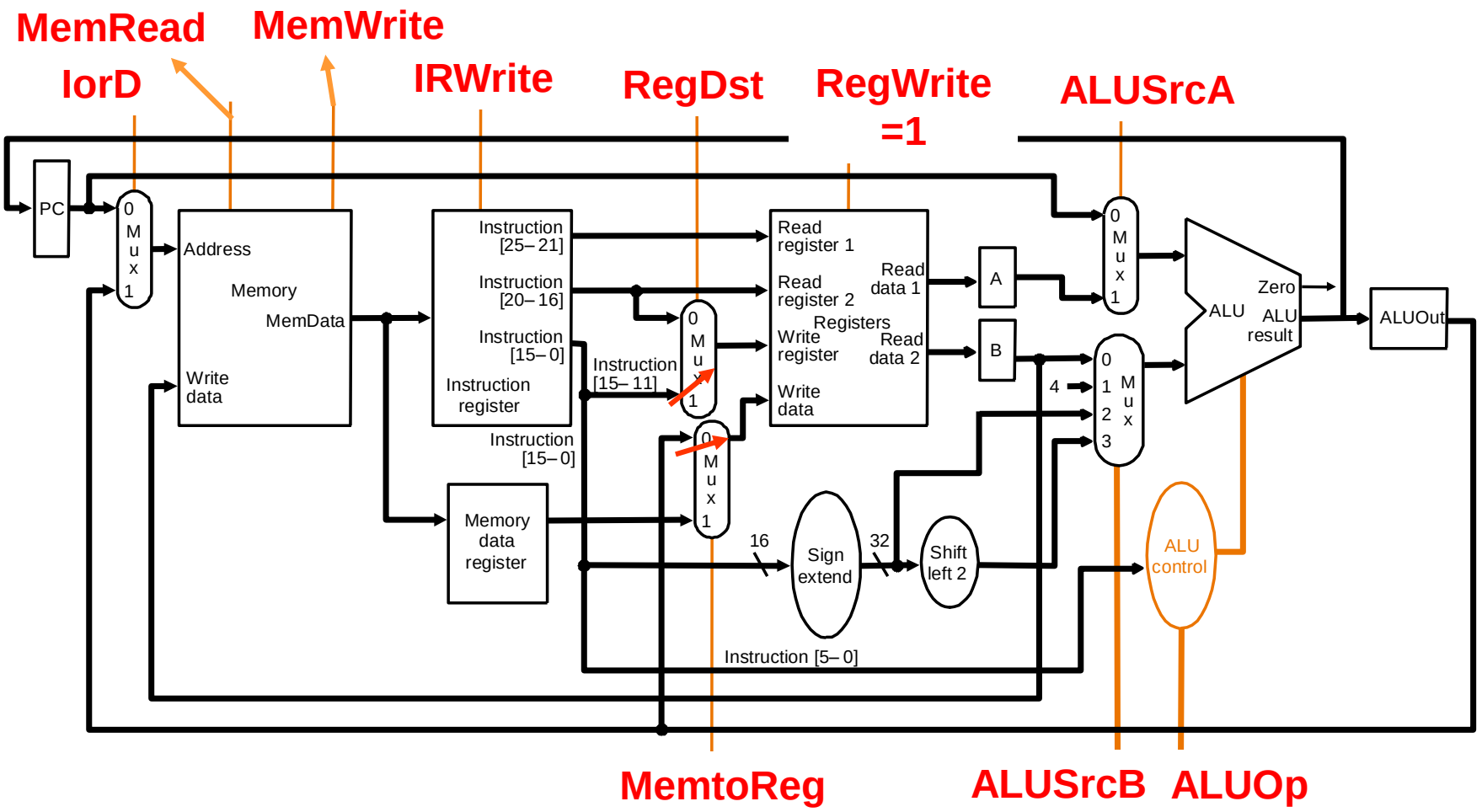


Implementação Multi-ciclo

- Aritmética
  - ALUout = A op B
  - Reg(IR(15-11)) = ALUout

Implementaç

- *Aritmética*
  - *–Reg(IR(15-11)) = ALUout*

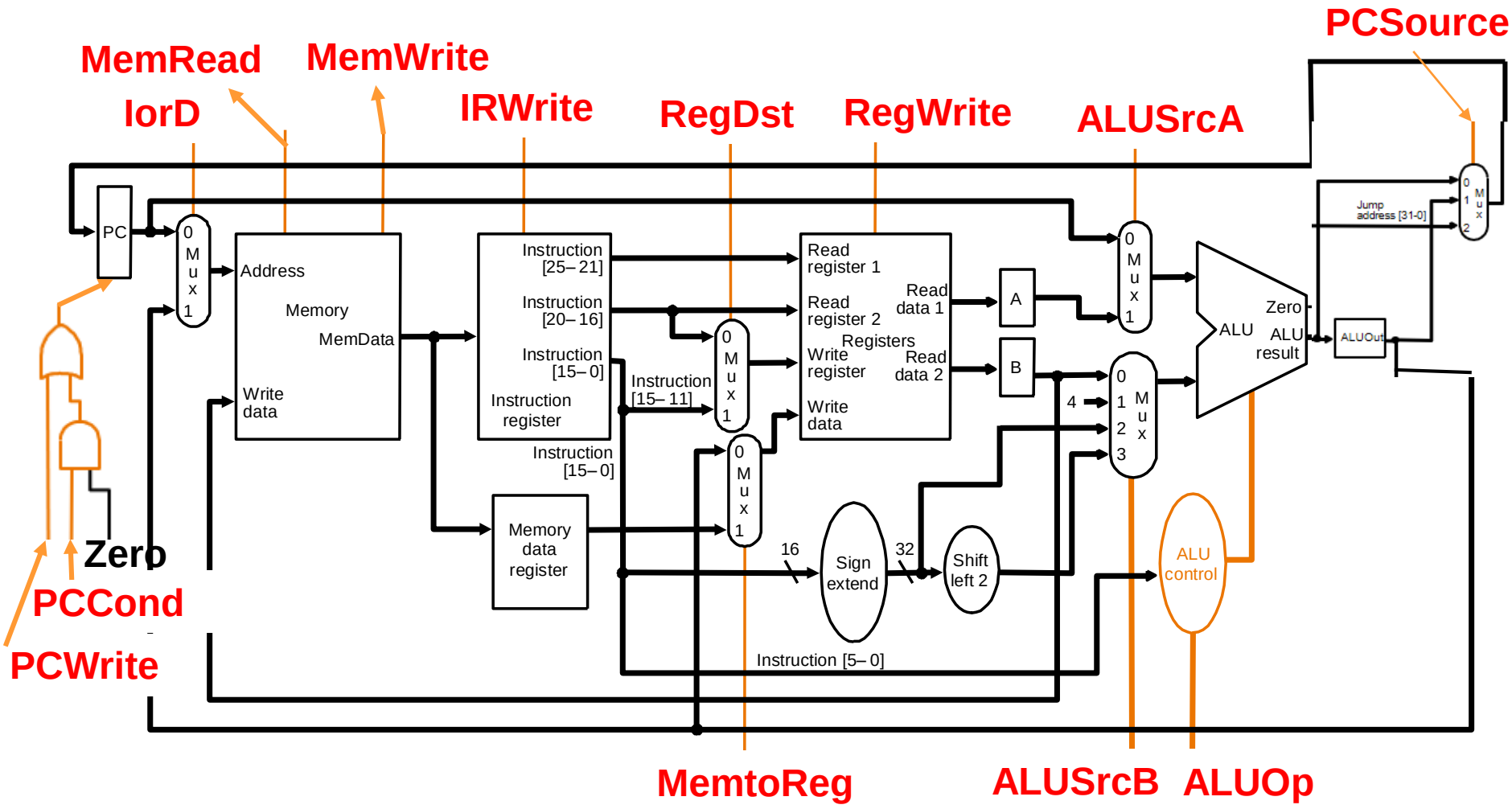


Implementação Multi-ciclo

- Branch
  - if (A ==B) PC = ALUout
- Jump end

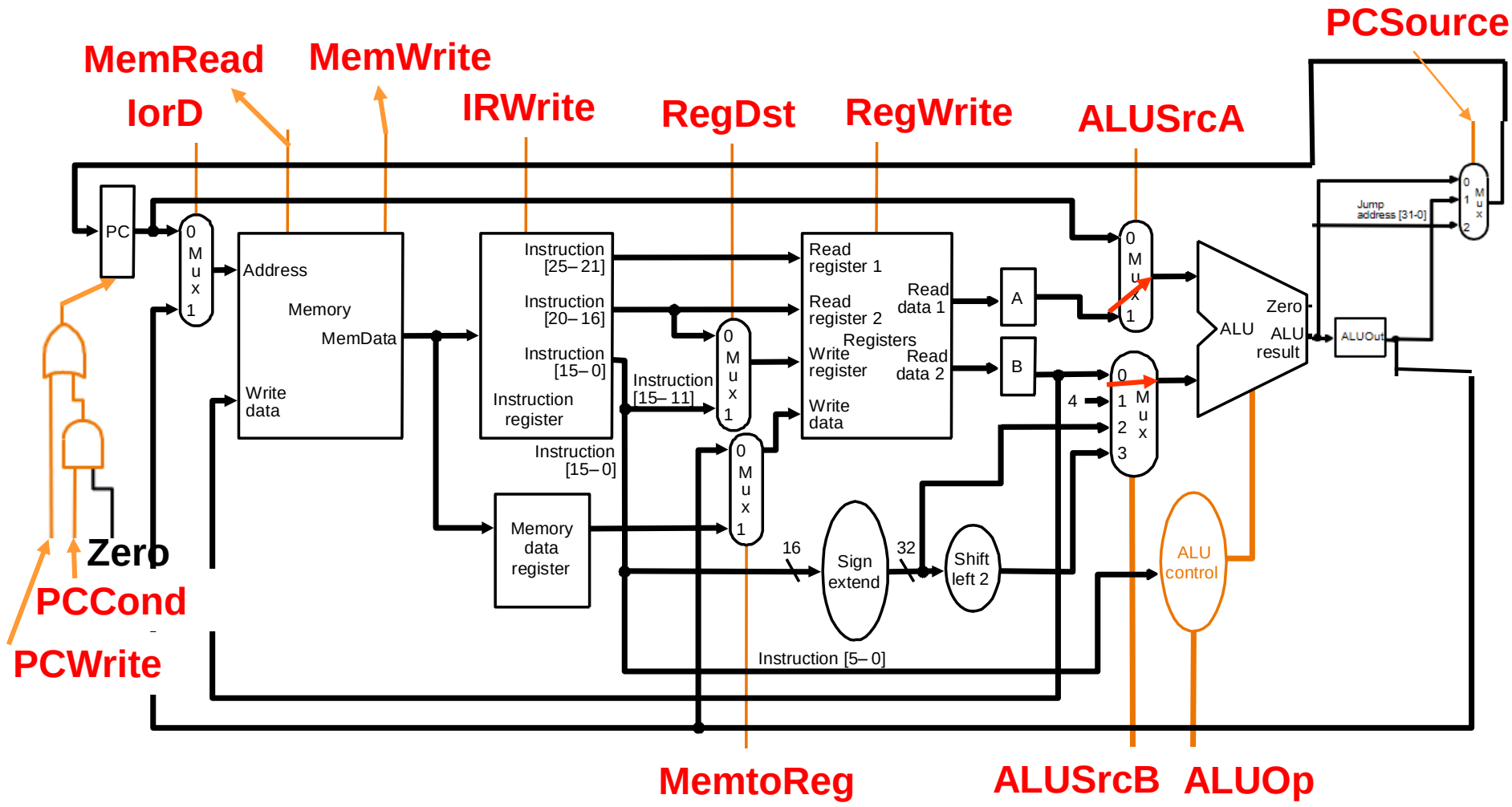
ação Multi-ciclo

- *Branch*
  - *if (A ==B) PC = ALUout*
  - *Jump end*

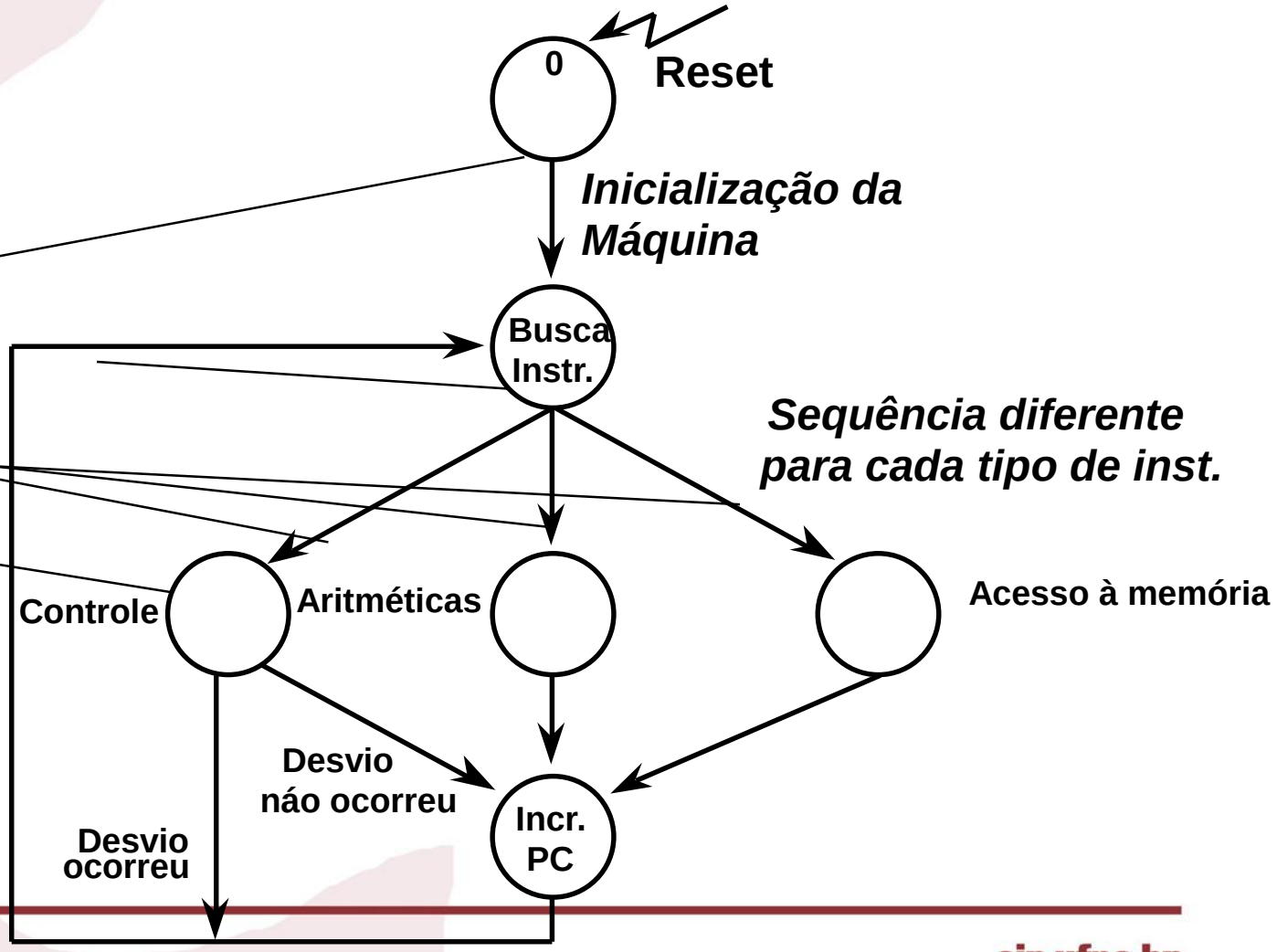


ação Multi-ciclo

- *Branch*
  - *if (A ==B) PC = ALUout*
  - *Jump end*



Unid. Controle - Diagr. Estados

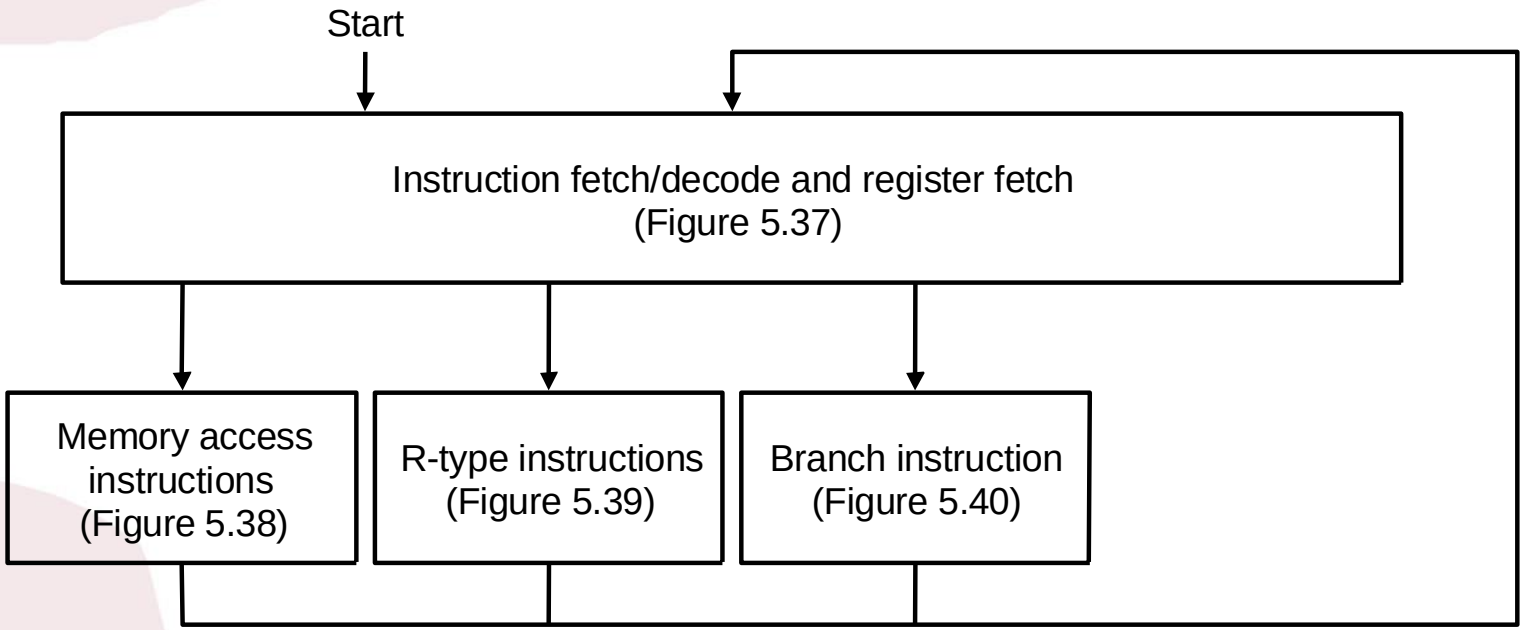


- Reset
- Busca
- Decodif.
- Exec.

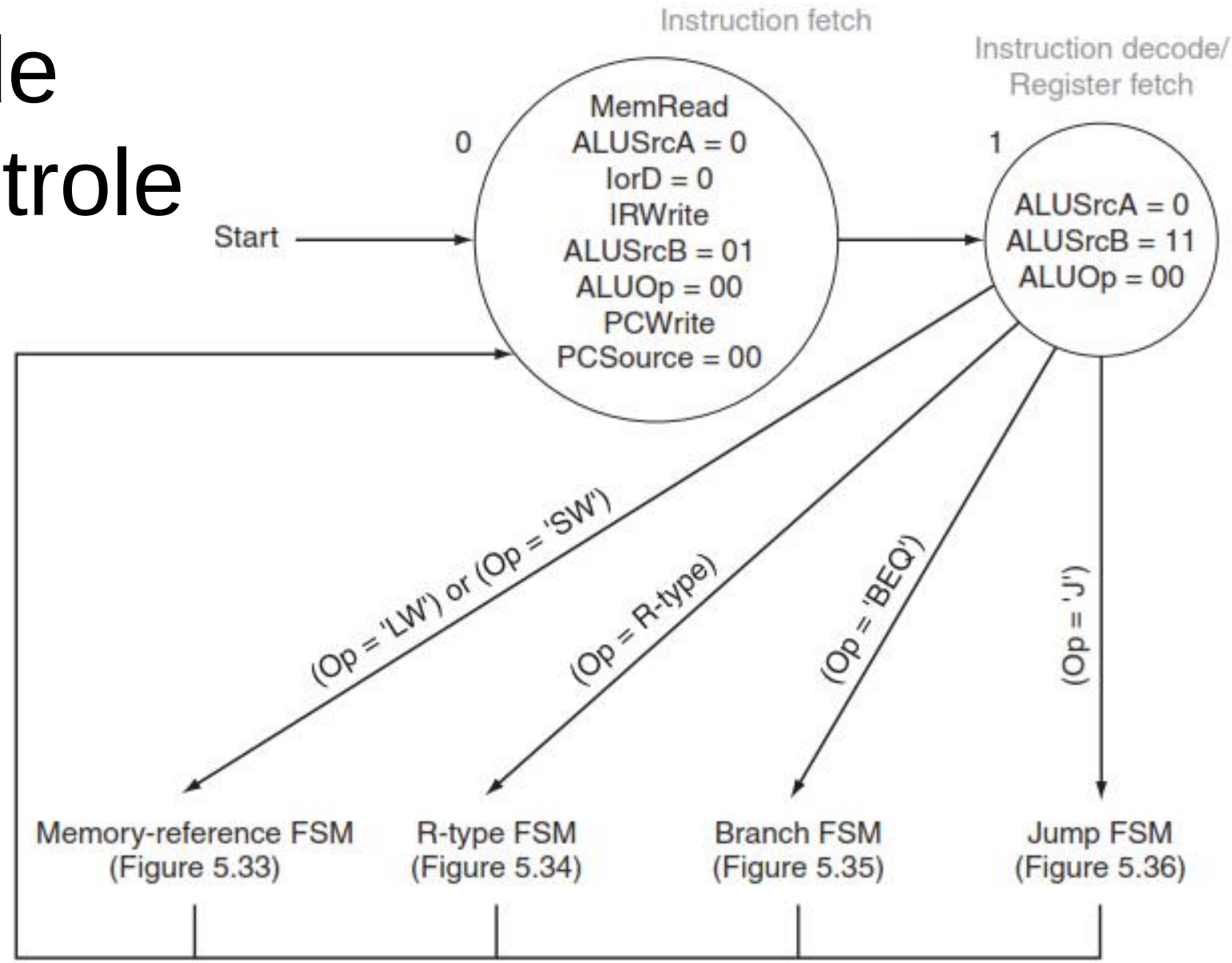
Três classes de instruções:

- Aritméticas
- Movimentação
- Controle

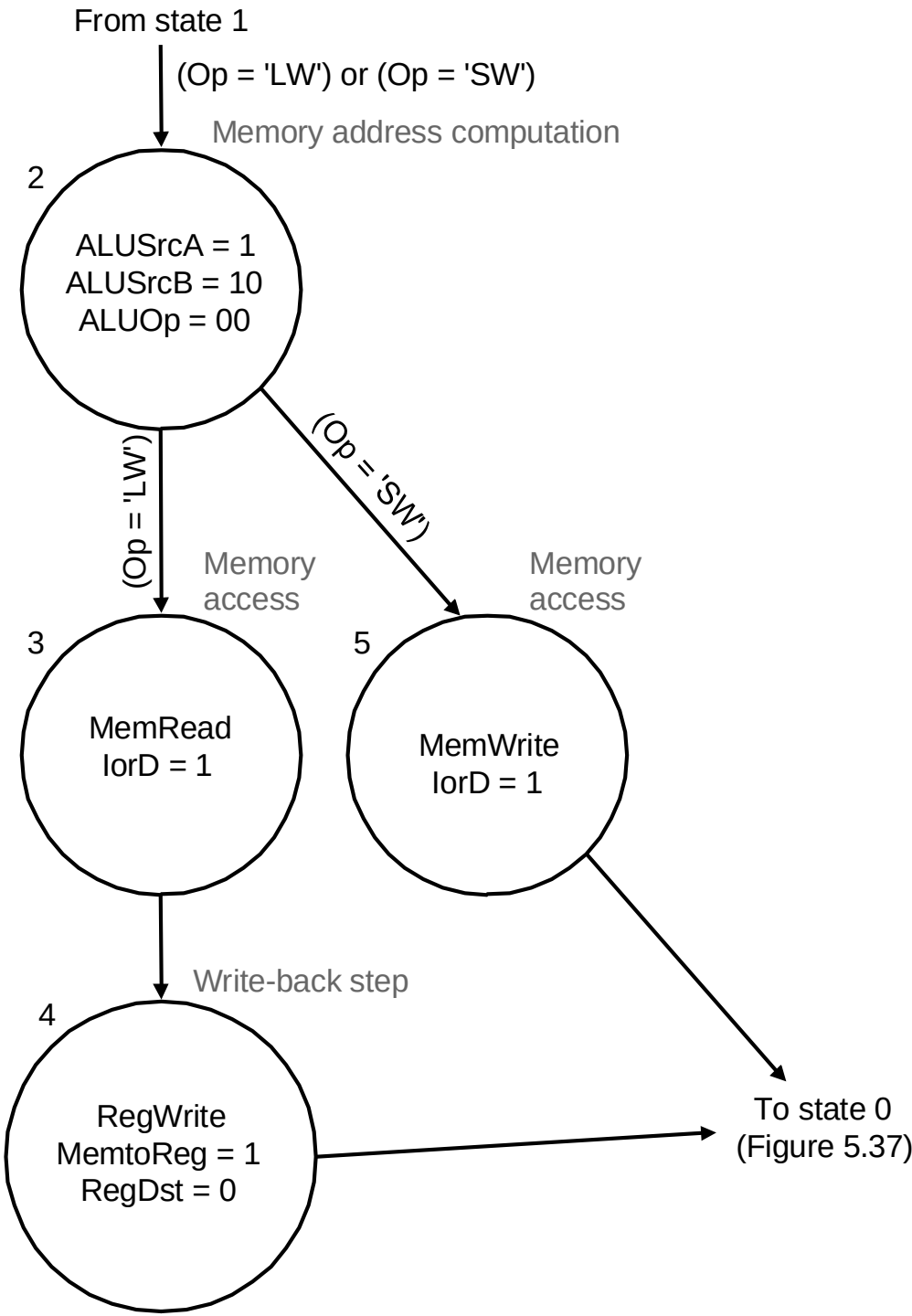
Unidade de Controle



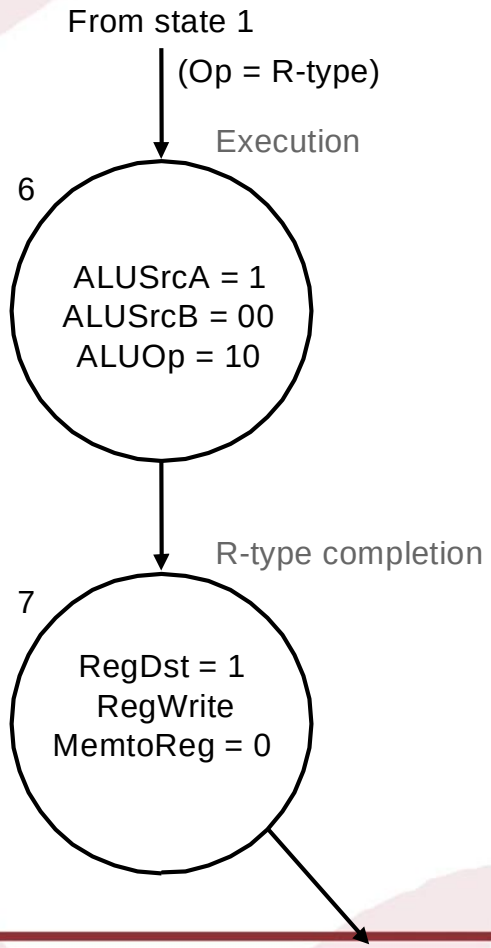
Unidade de Controle



Unidade de Controle

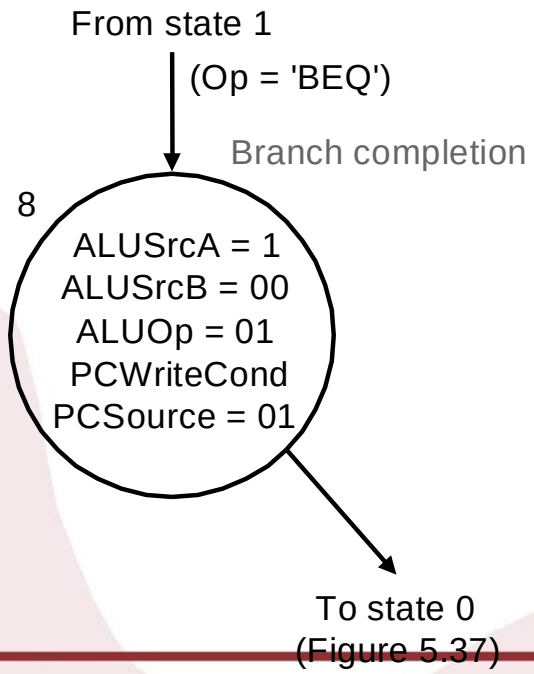


Aritméticas

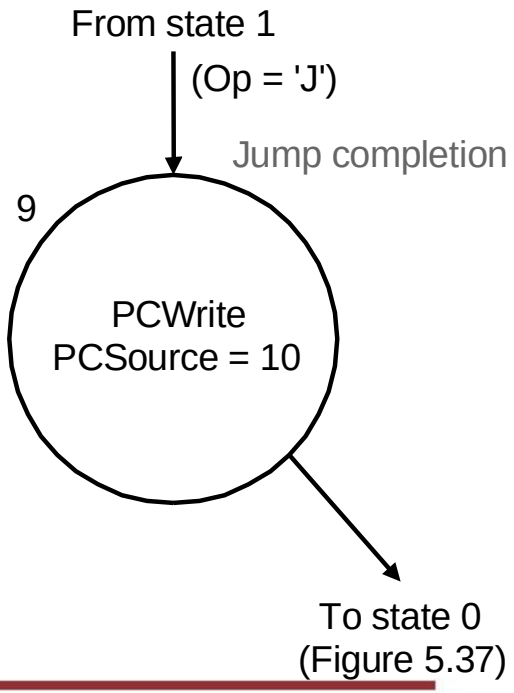


Desvio

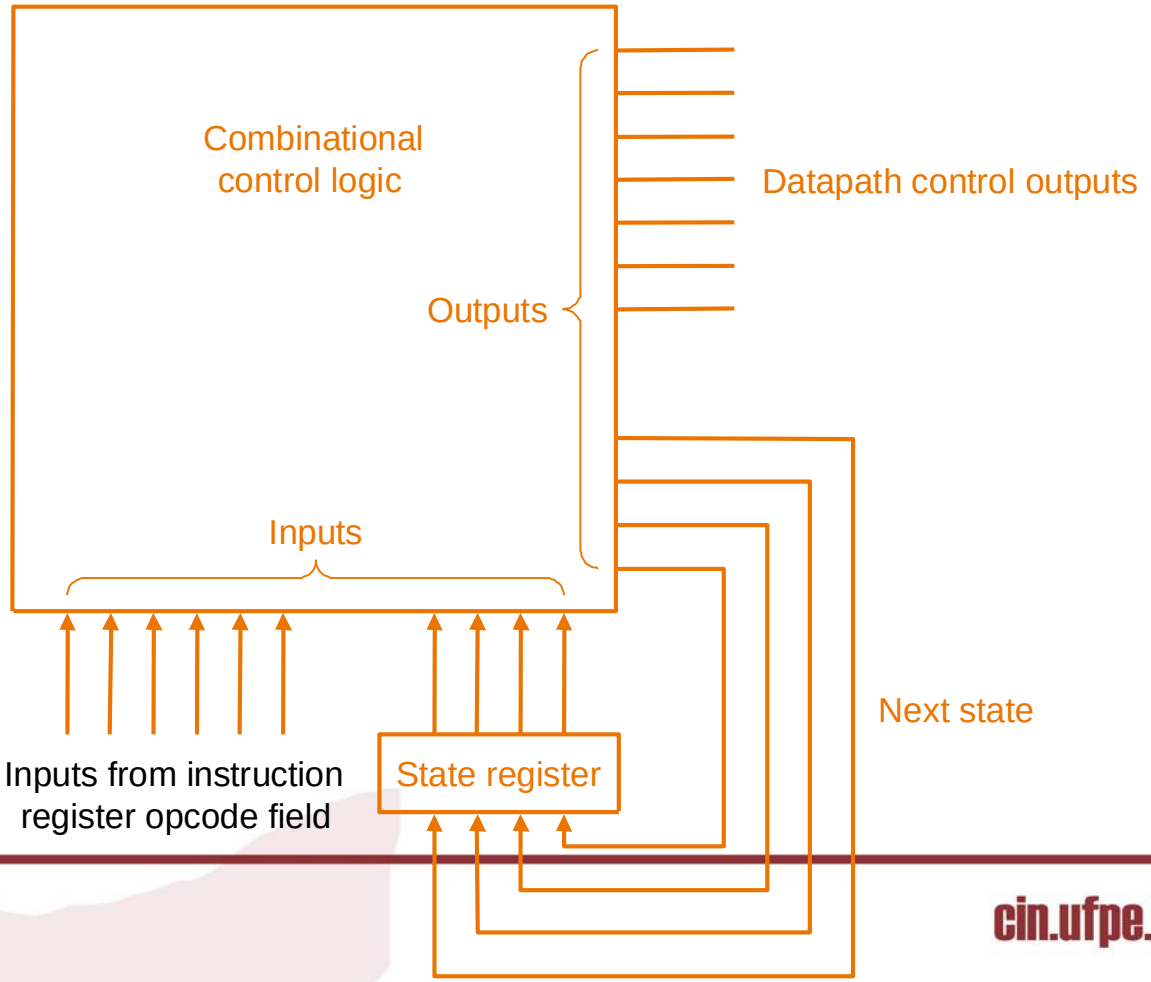
- BEQ



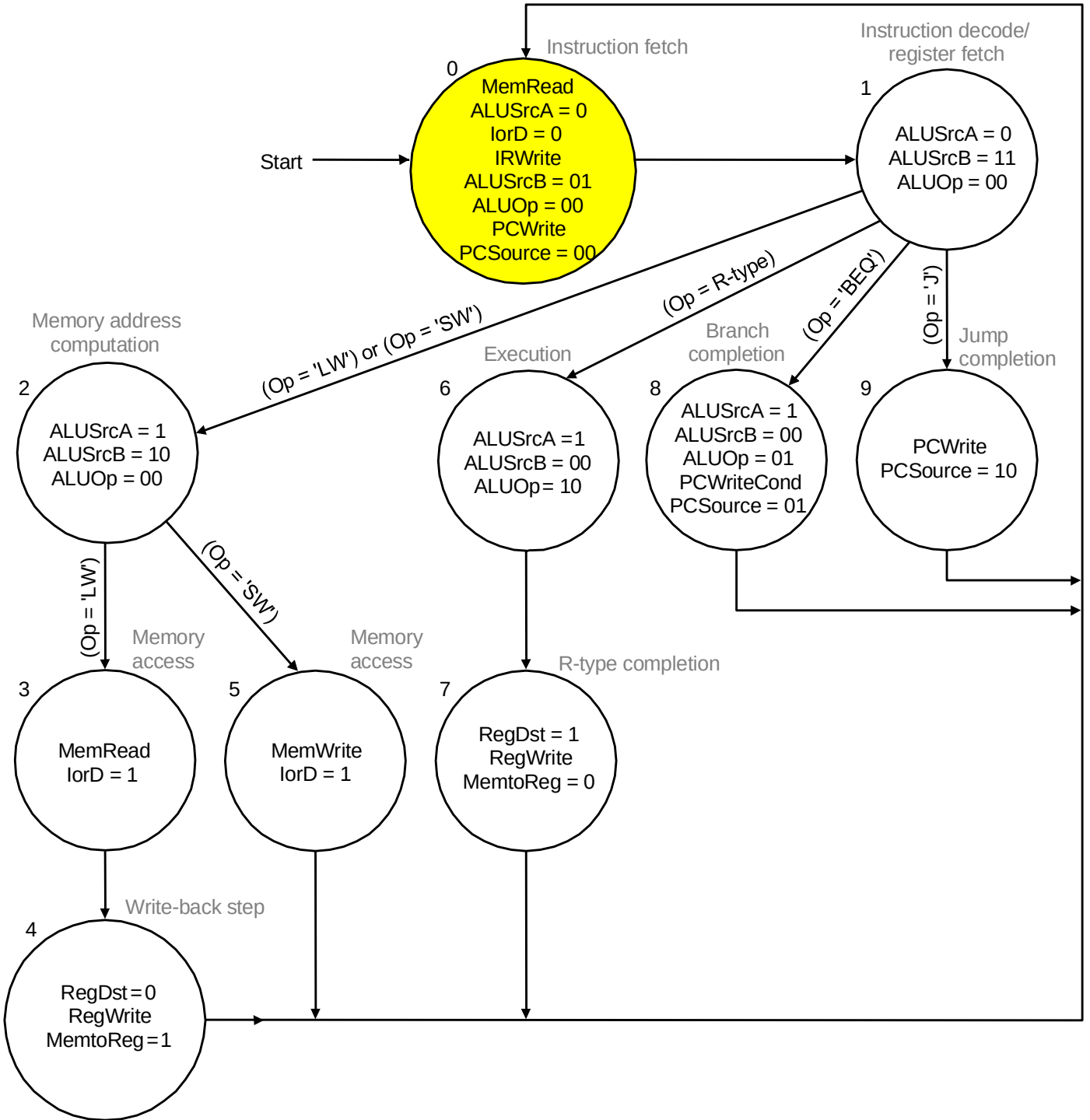
- Jump



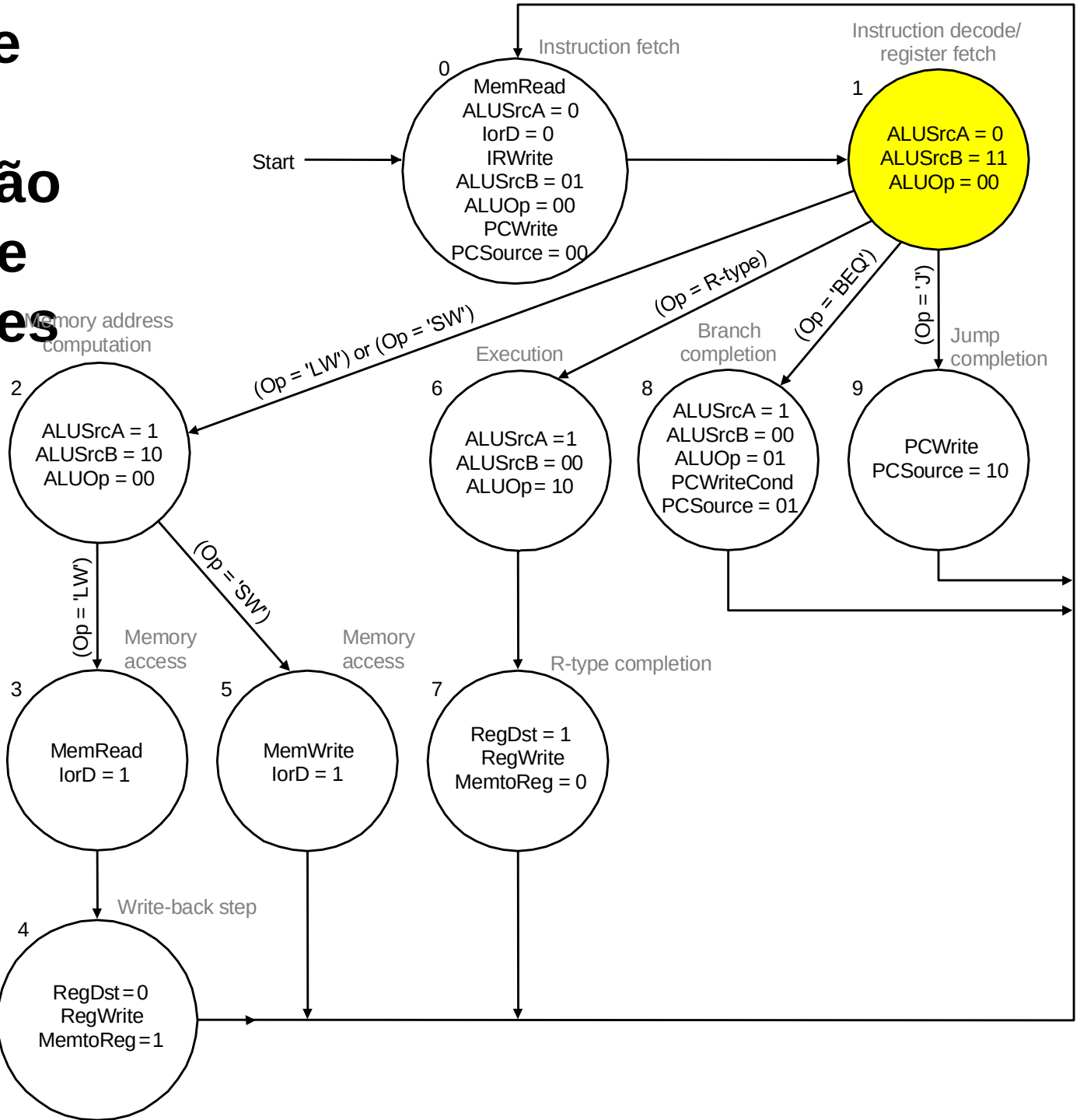
Unidade de Controle - FSM



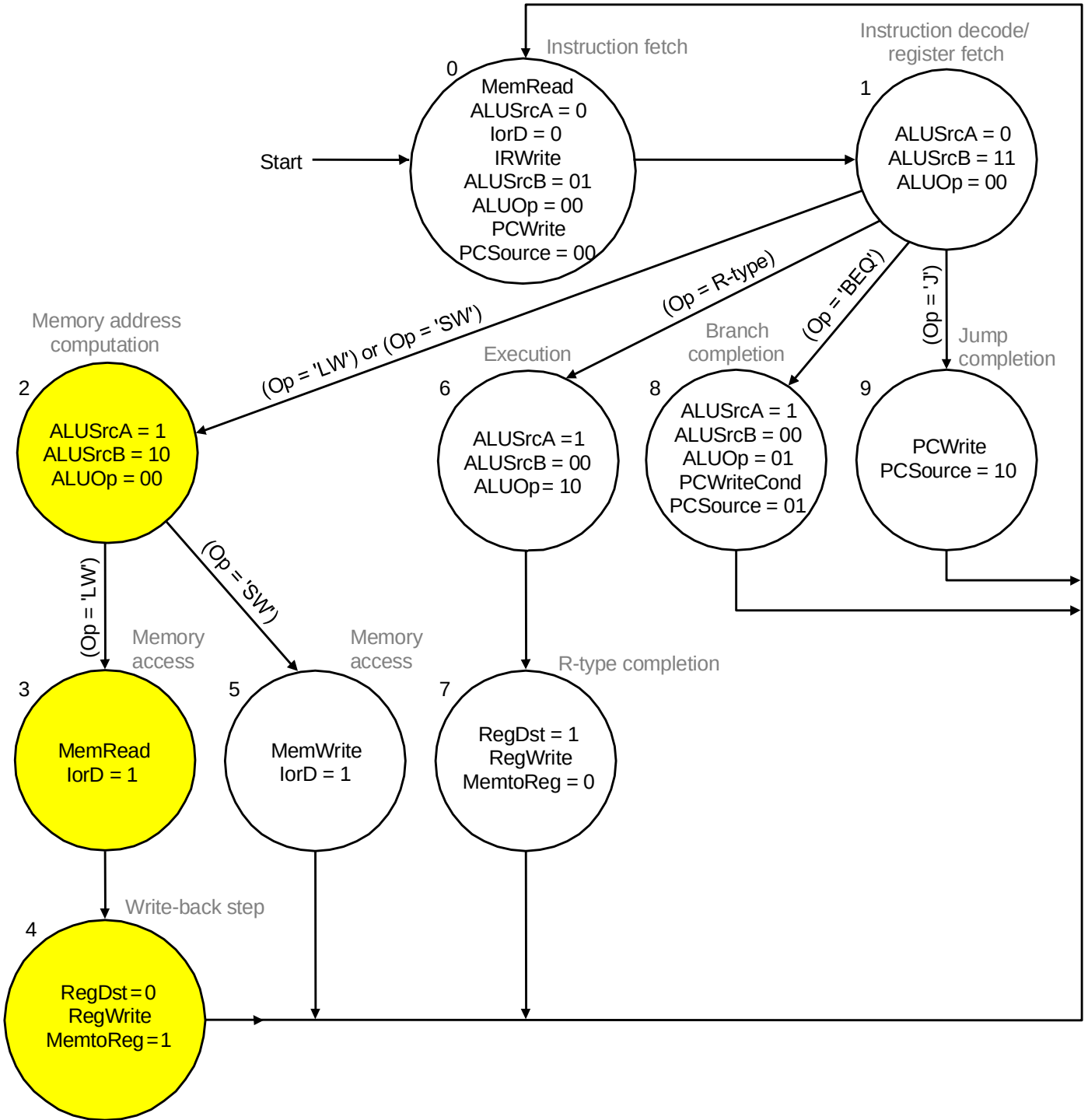
Unidade de Controle: Busca de Instrução



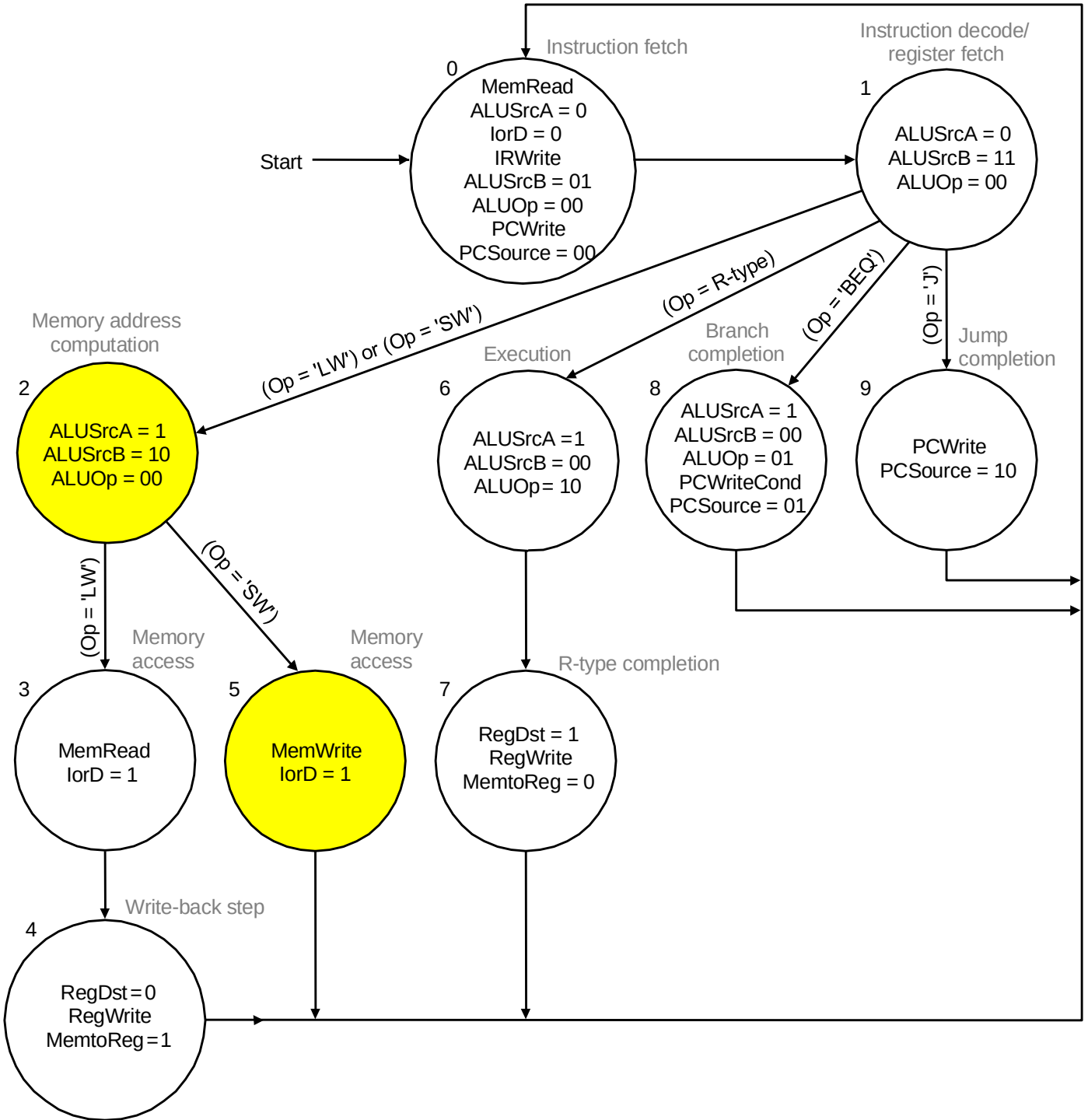
Unidade de Controle: Decodificação e Leitura de Registradores



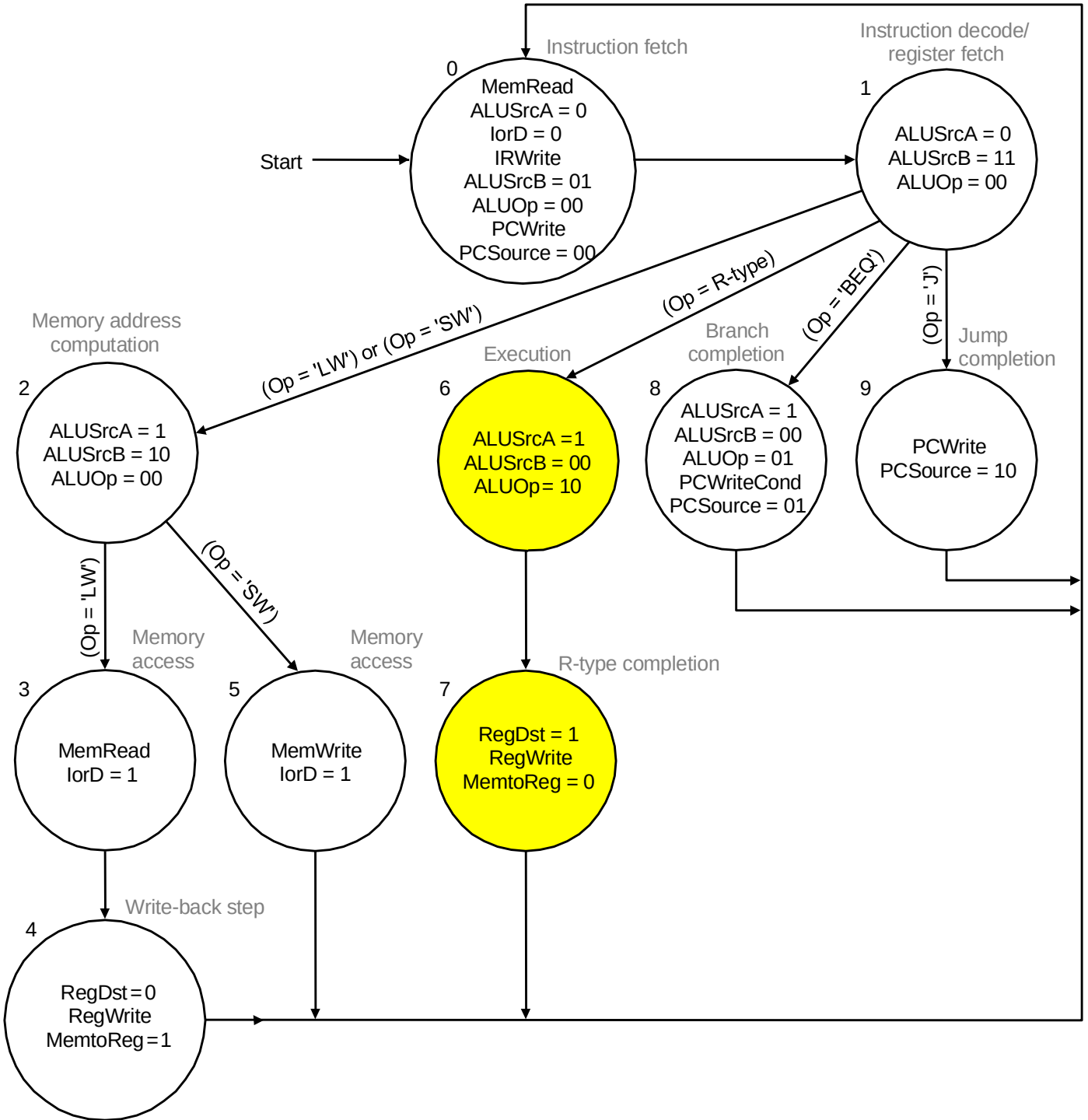
Unidade de Controle: Execução do LW



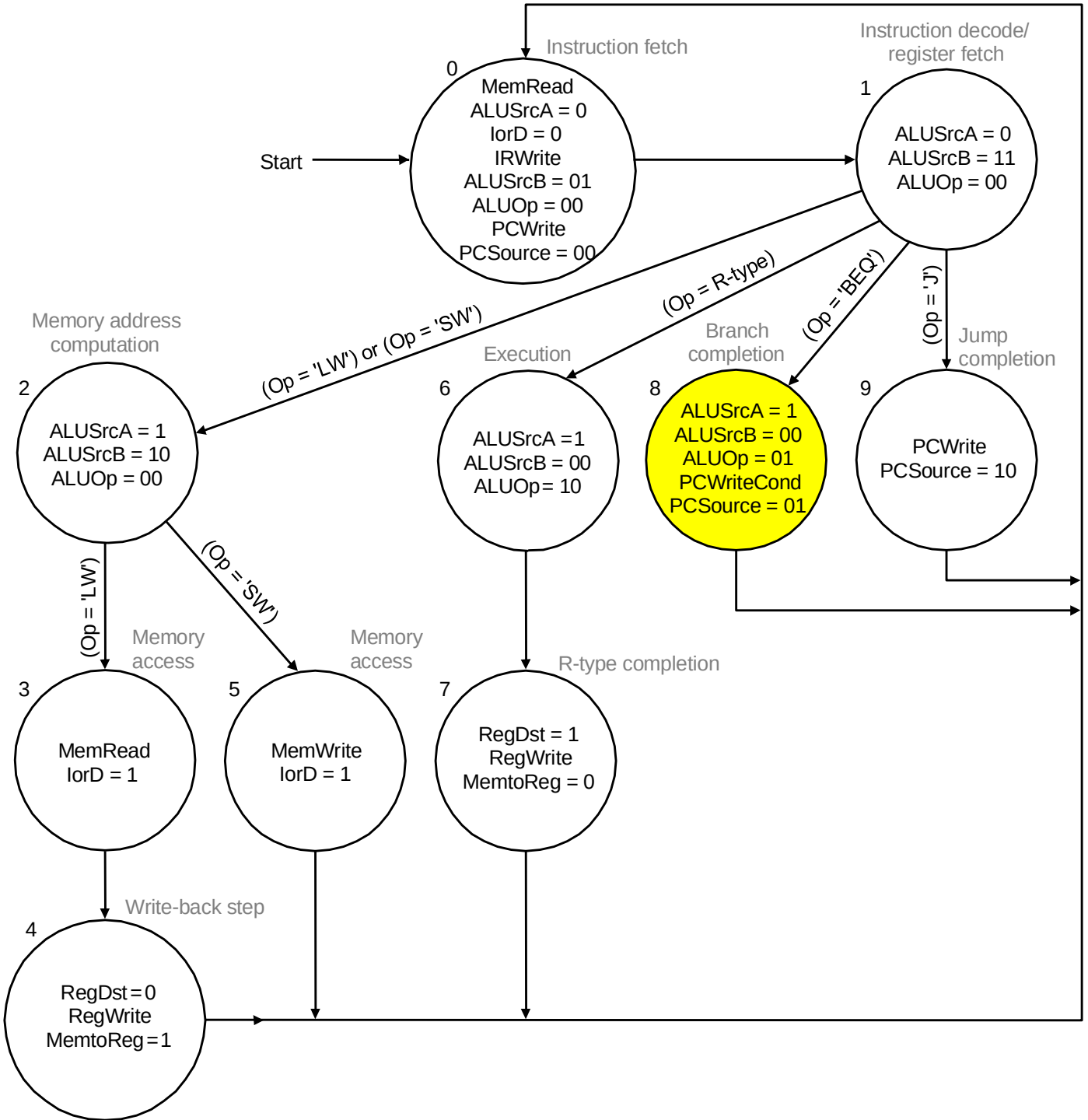
Unidade de Controle: Execução do SW



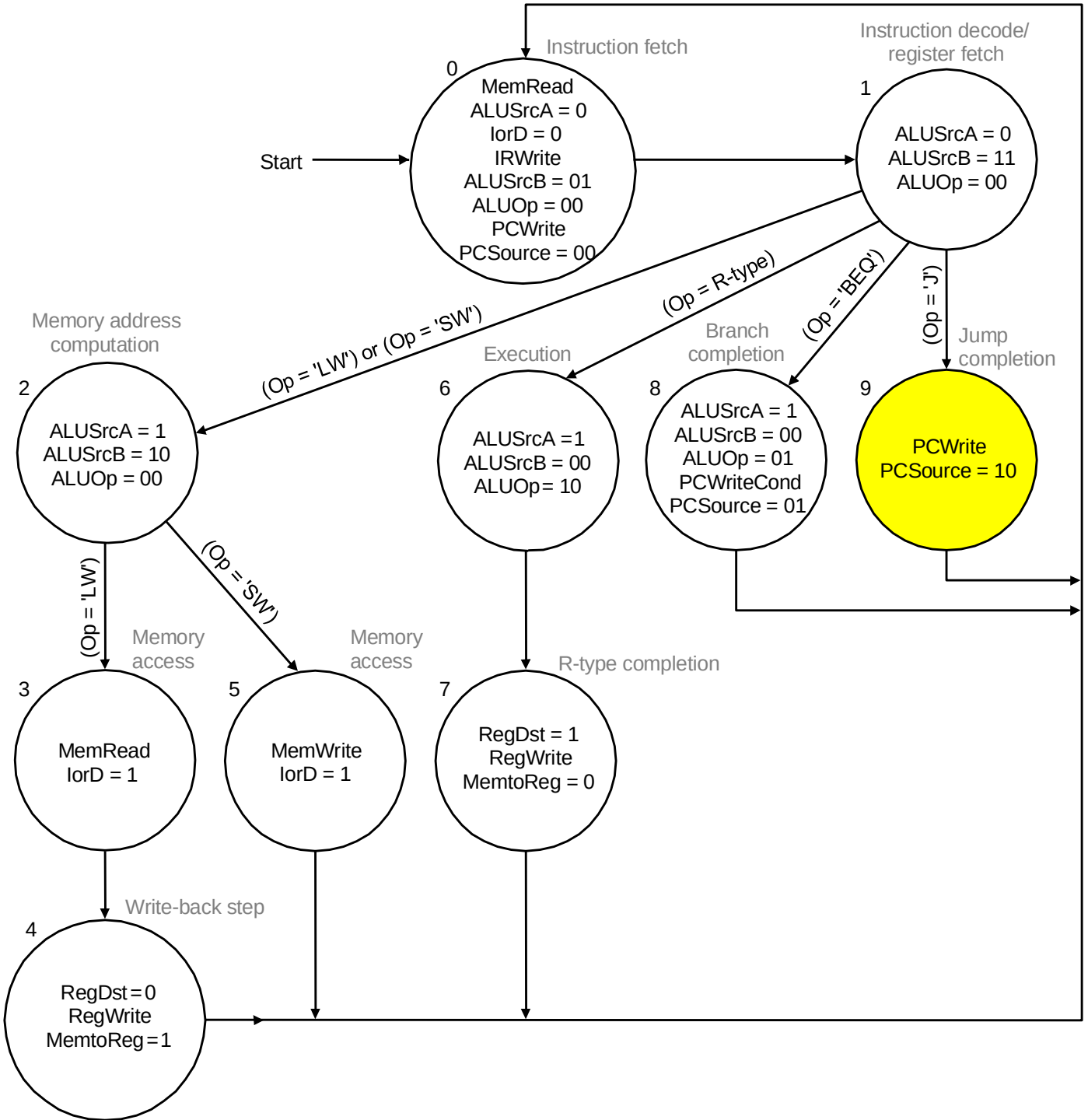
Unidade de Controle: Execução de ADD



Unidade de Controle: Execução do BEQ



Unidade de Controle: Execução do Jump



Centro de Informática U·F·P·E

CPU: Exceções

Exceções

- Sequência de execução é alterada devido a eventos não esperados:
  - internos:
    - opcode inexistente
    - overflow
    - divisão por zero
  - externos:
    - dispositivo de entrada/saída

Exceções

- Execução do programa é interrompida e uma rotina de tratamento é executada
  - Valor do PC deve ser guardado
  - O endereço da rotina de tratamento deve ser carregado em PC

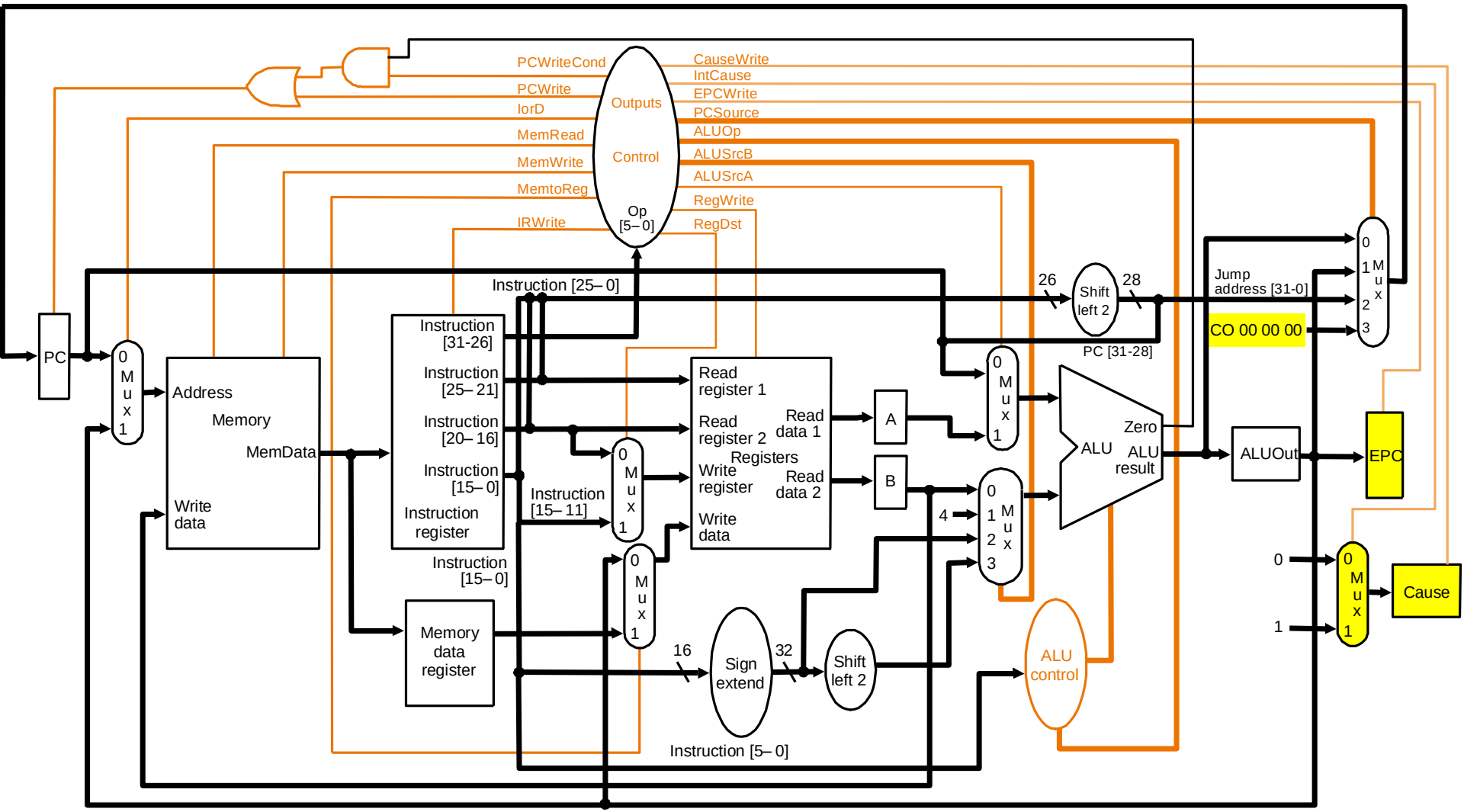
Exceções

- Onde guardar o endereço do PC
  - Registrador
    - MIPS: EPC
  - Pilha
- Onde o endereço da sua subrotina deve ser guardado
  - Valor fixo
    - precisa-se saber a causa da exceção
  - Vetor de endereços na memória

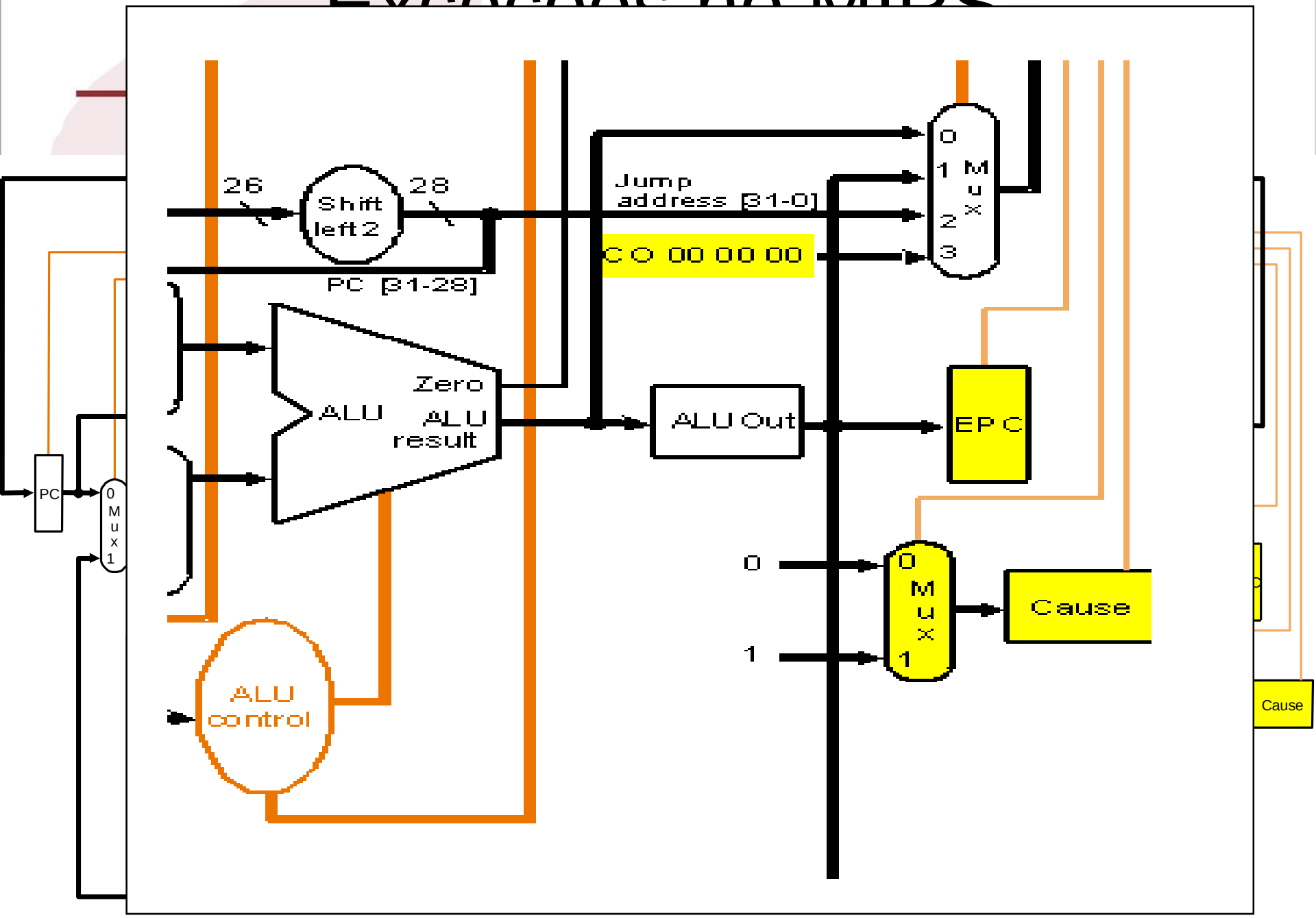
Exceções - MIPS

- Tipos de Exceções:
  - Instrução Indefinida
    - endereço: C0 00 00 00
  - Overflow aritmético
    - endereço: C0 00 00 20
- Registrador EPC
  - guarda endereço da instrução afetada
- Registrador de Causa
  - identifica o tipo de evento que causou a exceção

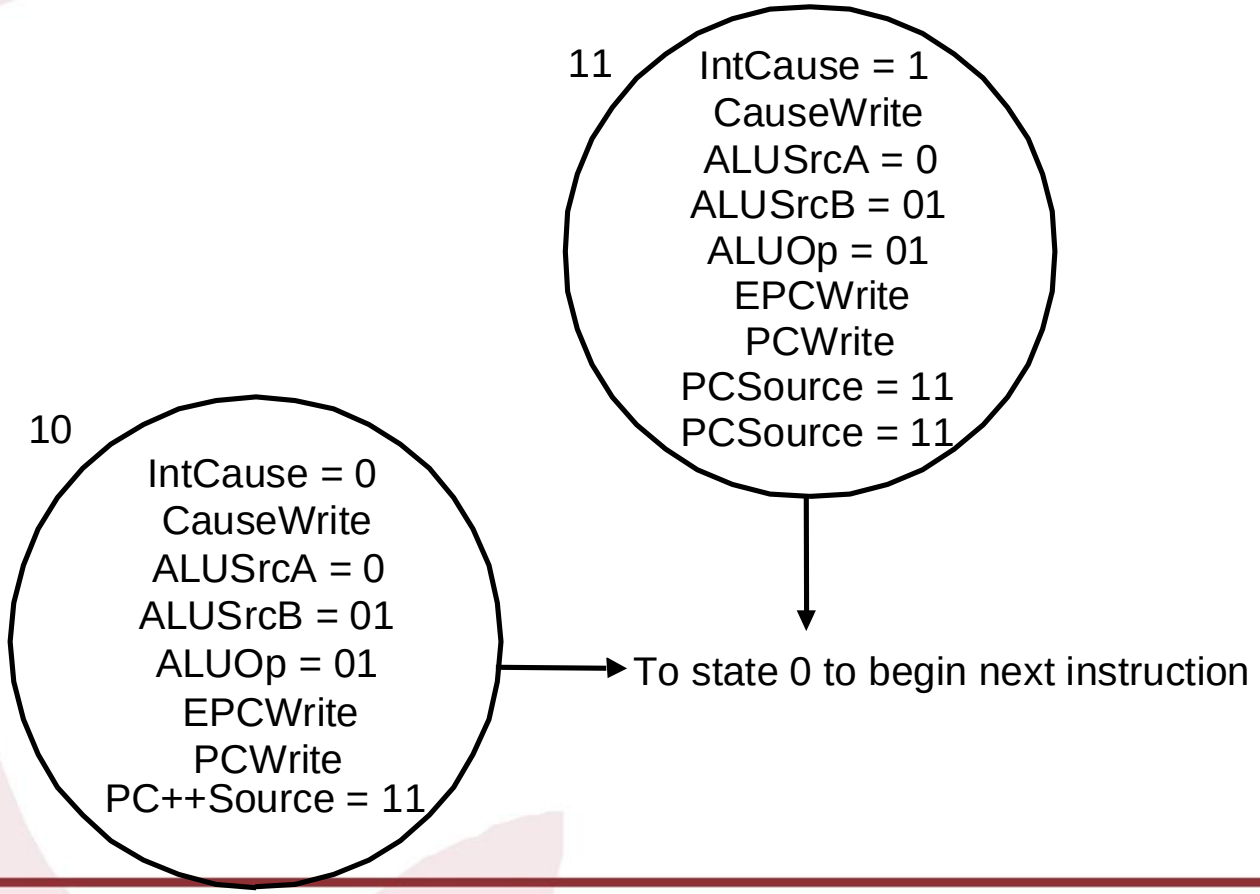
Exceções no MIPS



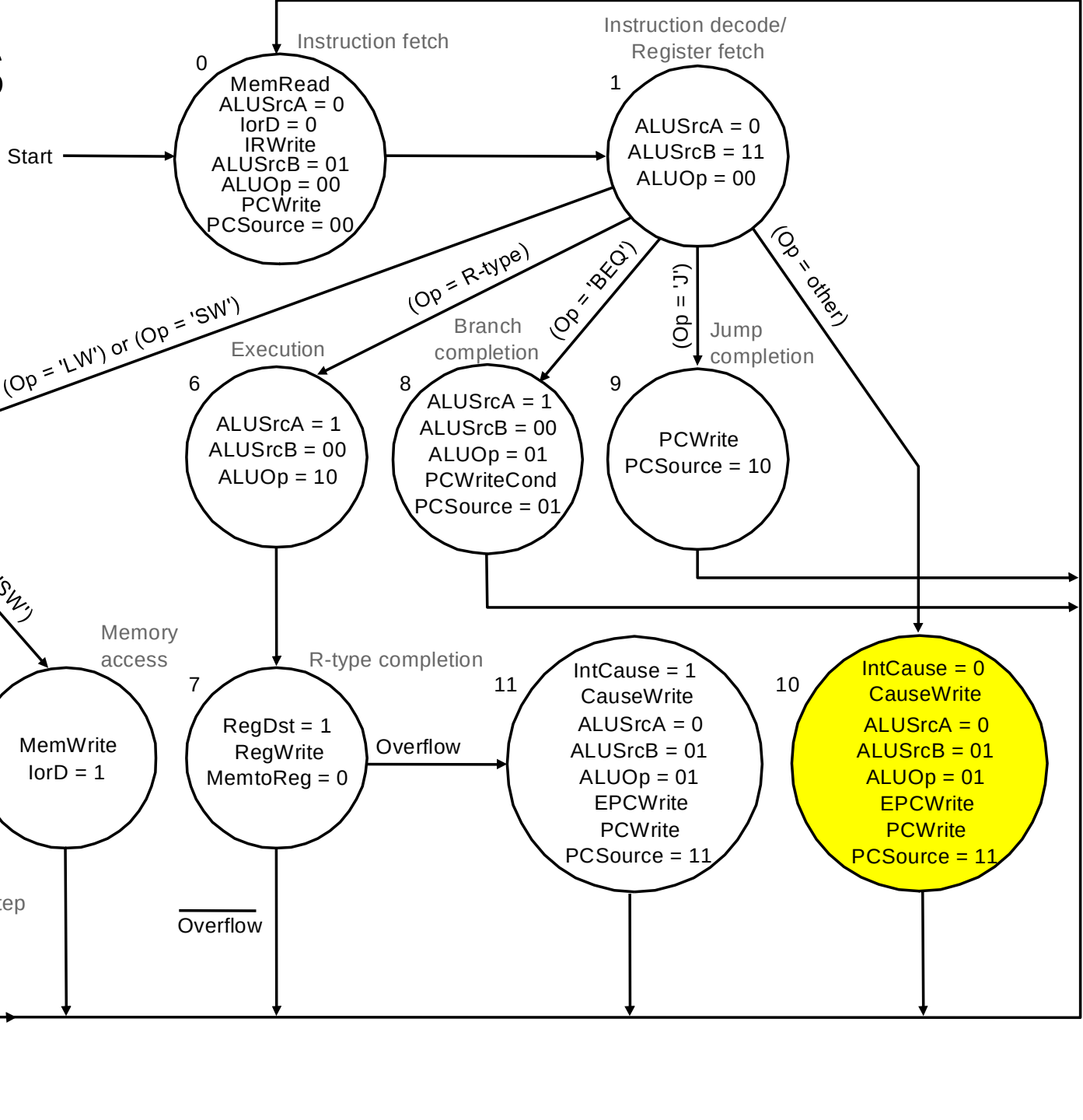
Excepções no MIPS



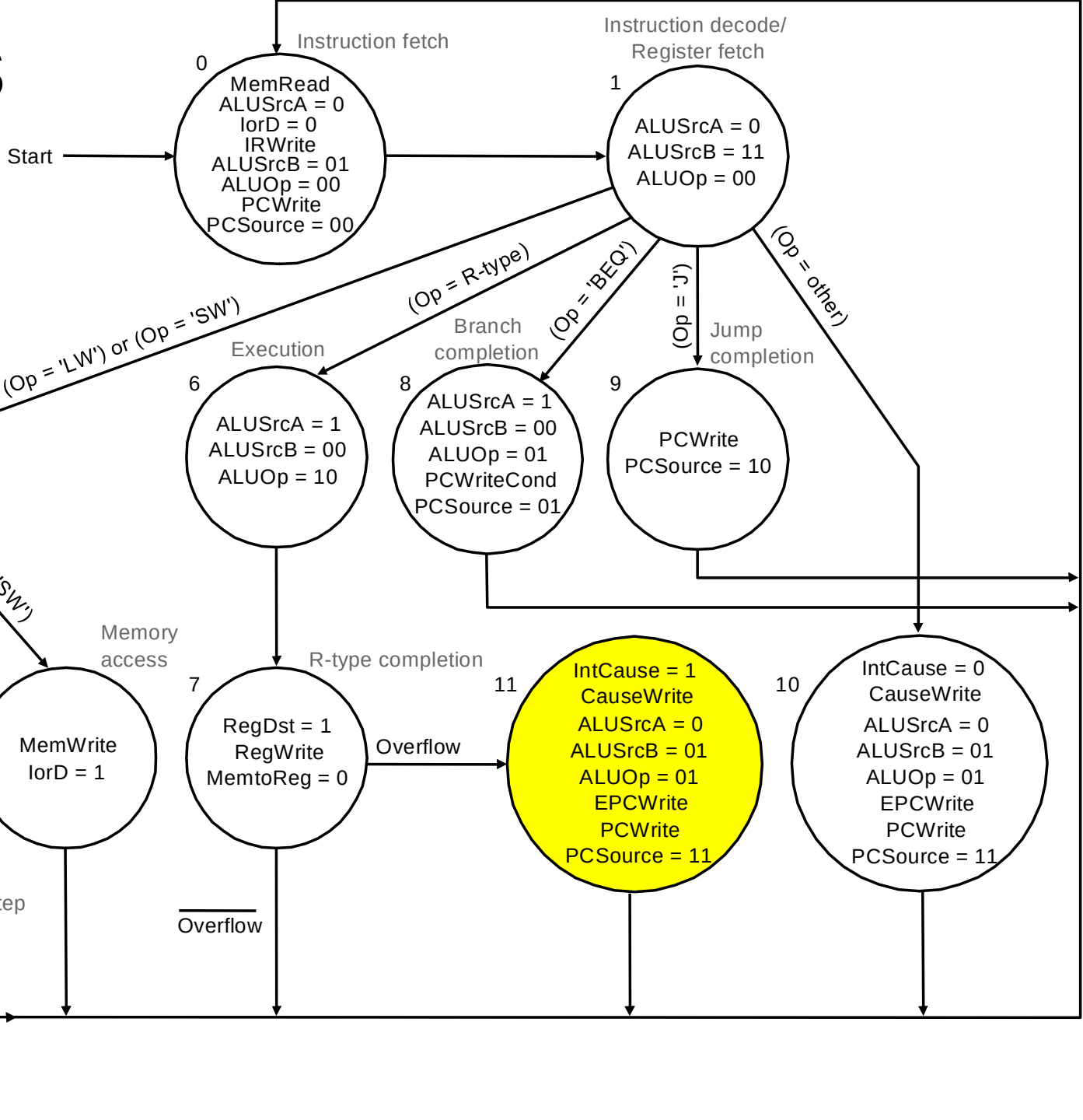
Exceções no MIPS



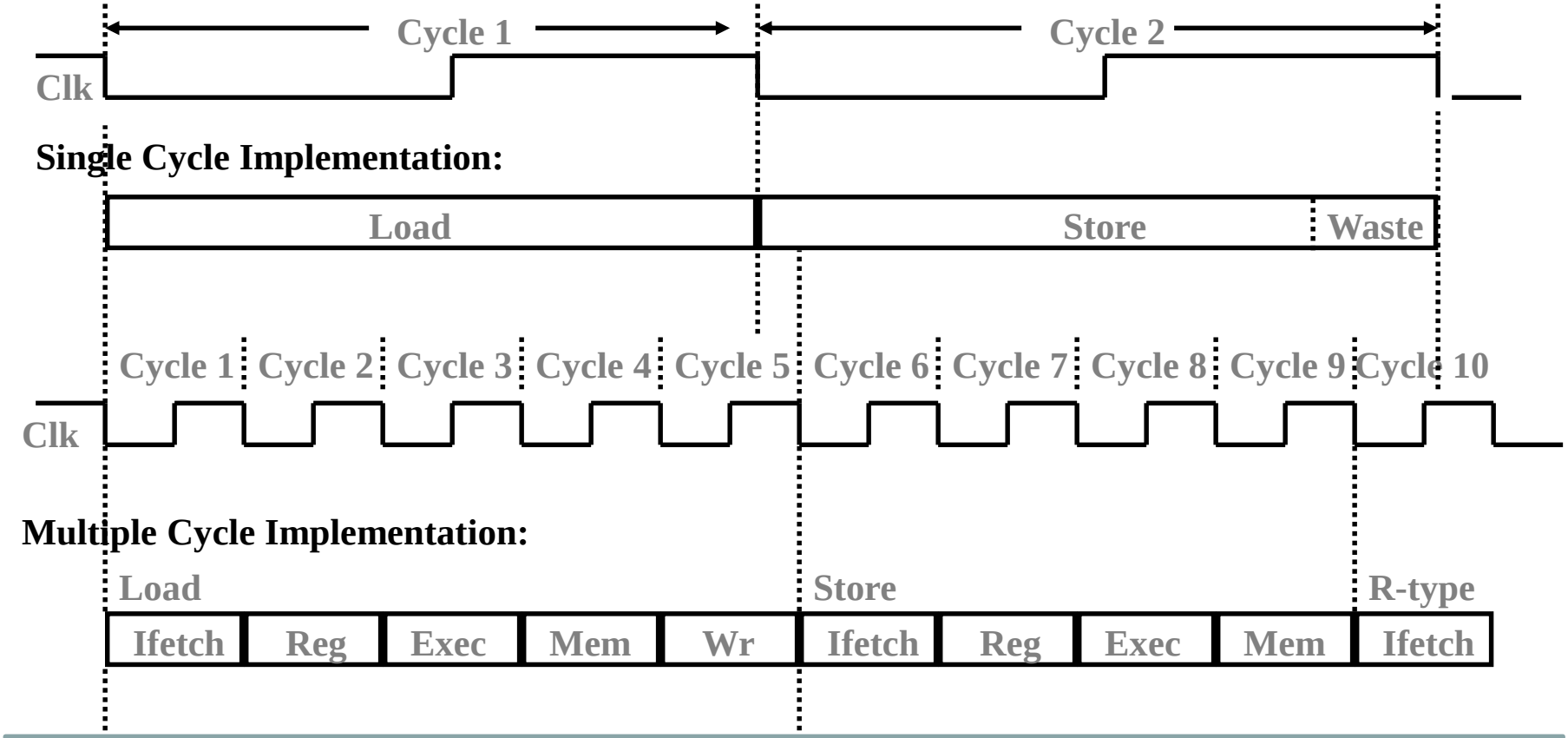
Exceções no MIPS



Exceções no MIPS



Mono, multi-ciclo



Projeto

- Projetar CPU que implementa repertório de uma CPU similar ao MIPS
- Dados módulos em Verilog para cada componente da Unidade de processamento (ALU, Banco de Registradores, PC, Memória, etc...)
- Projetar unidade de processamento pela interligação dos módulos
- Projetar unidade de controle