

cin.ufpe.br



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CPU: Estrutura e Funcionalidade

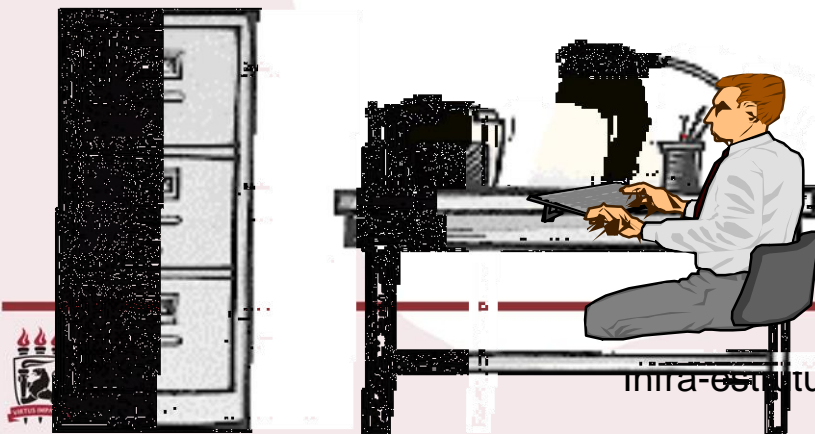
Roteiro da Aula



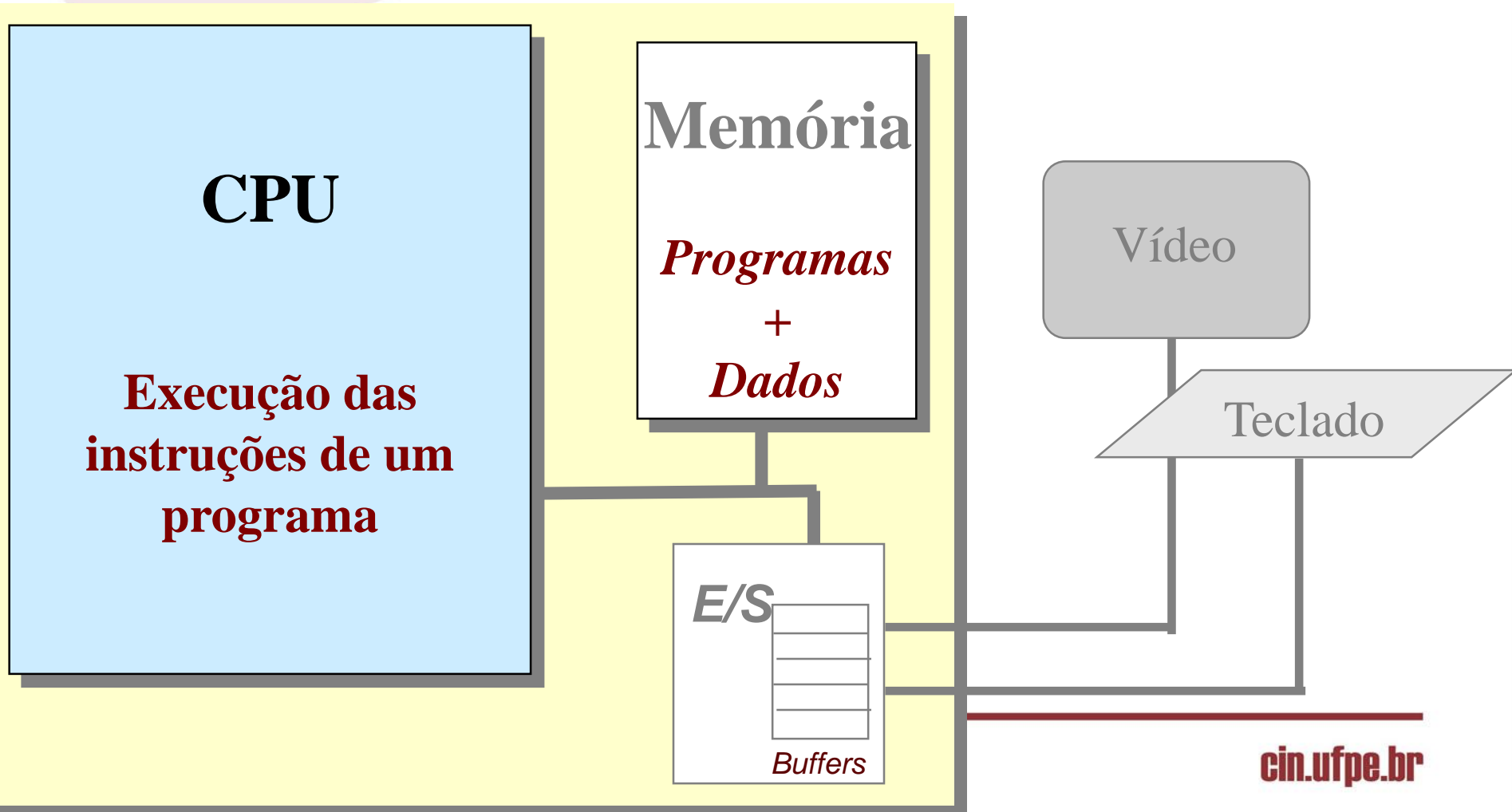
- Ciclo de Instrução
- Projeto de uma CPU simples: conceitos
- Componentes básicos
 - Leitura da instrução
 - Operação entre registradores
 - Acesso à memória
- Implementação Mono-ciclo
 - Leitura da Instrução
 - Operações Aritméticas
 - Leitura + Operação entre registradores
 - Acesso à Memória
 - Desvio Condicional
- Visualizando a execução da instrução
- Análise de Desempenho

Como um documento é processado?

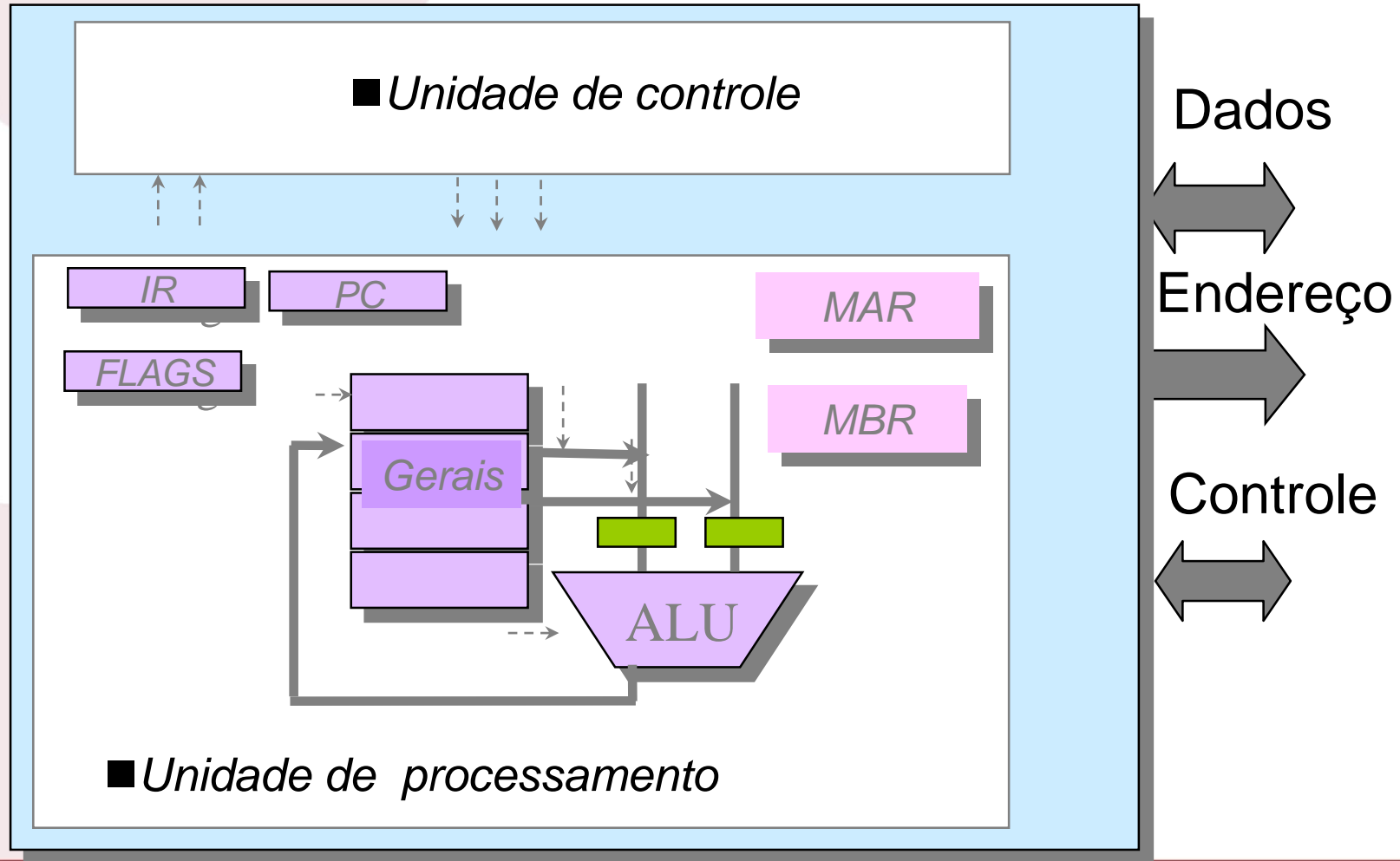
- 1- Busca documento
- 2 - Identifica tipo de transação
- 3 - Verifica se saldo é positivo
- 4 - Efetiva transação e atualiza saldo



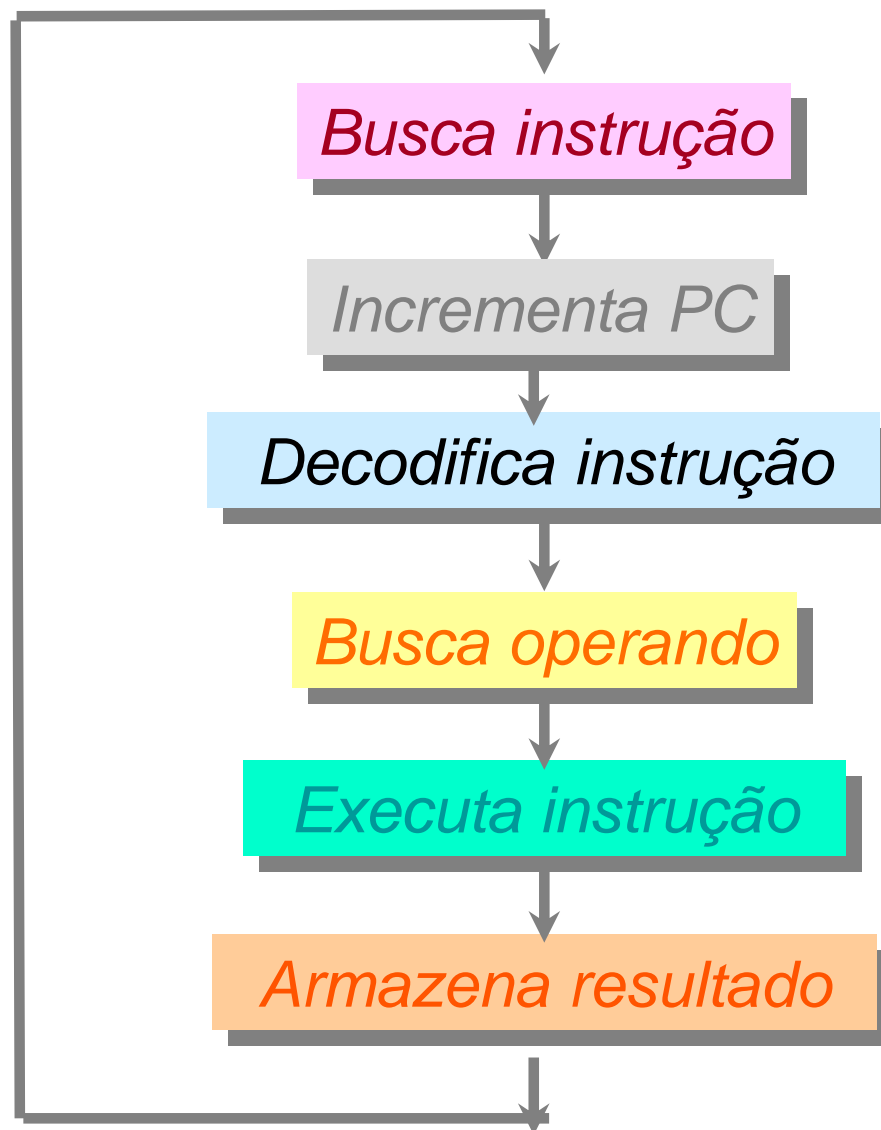
Componentes de um computador



Unidade Central de Processamento



Ciclo de Instrução



Projeto de uma Arquitetura



- Conjunto de registradores
- Tipos de Dados
- Formato e Repertório de instruções



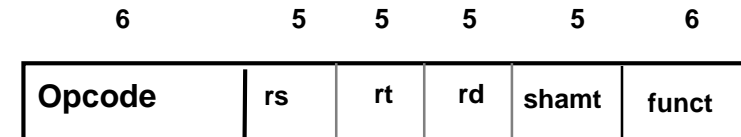
Instruções

- Tipos de instruções

- Processamento:

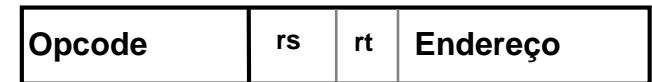
- aritméticas e lógicas

F1



- Armazenamento

F2



6 5 5 16

- E/S

- Controle:

- teste e desvio

MIPS

Instrução	Descrição
nop	No operation
lw reg, desl(reg_base)	reg. = mem (reg_base+desl)
sw reg, desl(reg_base)	Mem(reg_base+desl) = reg
lui reg, constante	reg(31..16) = constante
add regi, regj, regk	Regi. <- Regj. + Regk
addi regi, regj, cte	Regi = regj + cte
sub regi, regj, regk	Regi. <- Regj. – Regk
and regi, regj, regk	Regi. <- Regj. and Regk
andi regi, regj, cte	Regi = regj and cte
shrl regd, regs, n	Desloca regs para direita n vezes (Lógico) e armazena valor deslocado em regd .
shfra regd, regs, n	Desloca regs para dir. n vezes (aritmético), armazena valor deslocado em regd.
shfl regd, regs, n	Desloca regs para esquerda n vezes, armazena valor deslocado em regd.
rotr regd, regs, n	Rotaciona regs para direita n vezes, armazena valor deslocado em regd.
rotl regd, regs, n	Rotaciona regs para esquerda n vezes, armazena valor deslocado em regd.
beq regi, regj, end	Desvia para end. se regi = regj
bne regi, regj, end	Desvia para end se regi <> regj
slt regi, regj, regk	Regi = 1 se regj < regk senão regi=0
slti regi, regj, cte	Regi = 1 se regj < cte senão regi=0
j end	Desvio para end
jr regi	PC = (regi)
jal end	R31 = PC; PC = end
break	Para a execução do programa

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	$rd \leftarrow rs + rt$
SUB rd, rs, rt	$rd \leftarrow rs - rt$
AND rd, rs, rt	$rd \leftarrow rs \text{ and } rt$
BEQ rs, rt, end	Desvio se $rs = rt$

Aritm

Opcode	rs	rt	rd	shamt	funct
--------	----	----	----	-------	-------

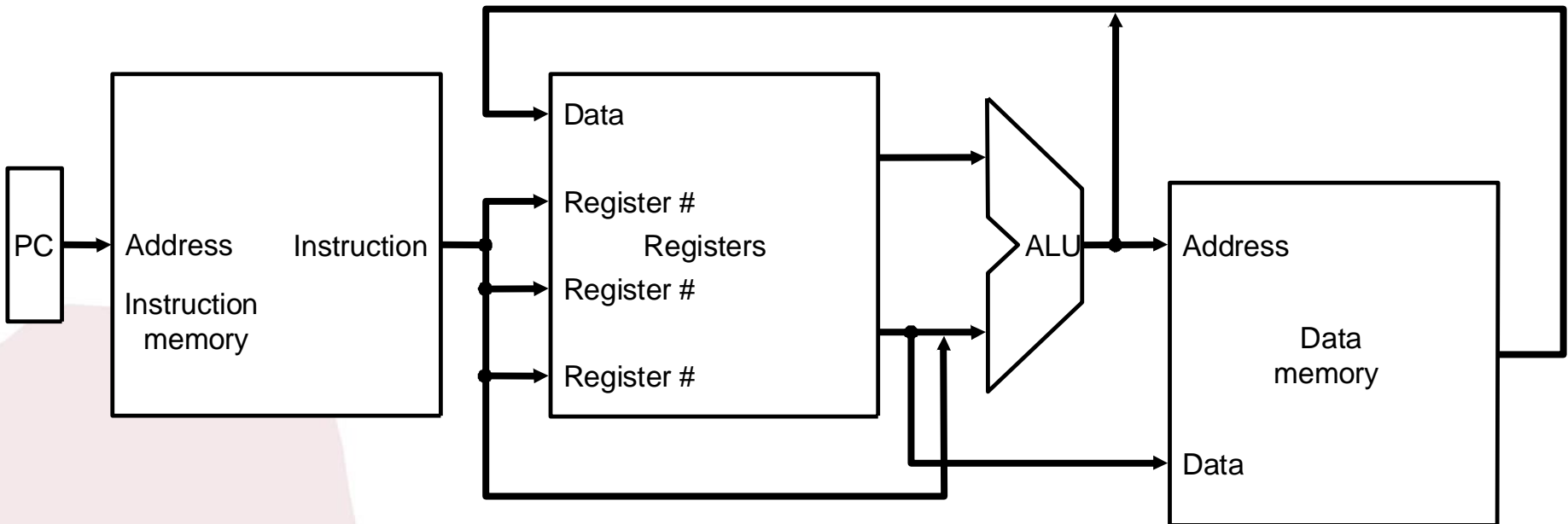
lw/sw

Opcode	rs	rt	Endereço
--------	----	----	----------

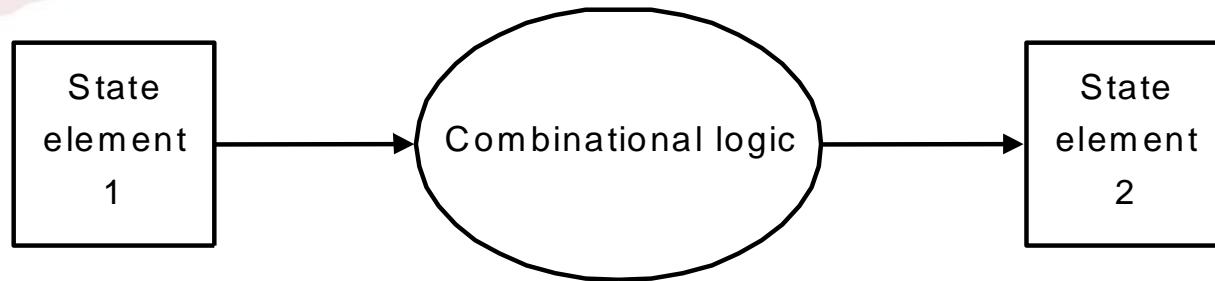
beq

Opcode	rs	rt	Endereço
--------	----	----	----------

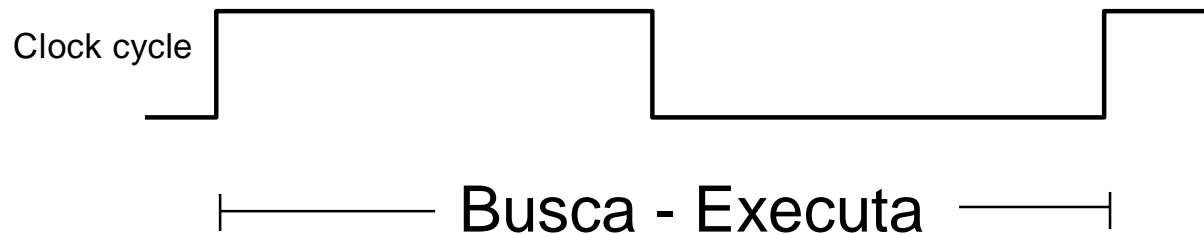
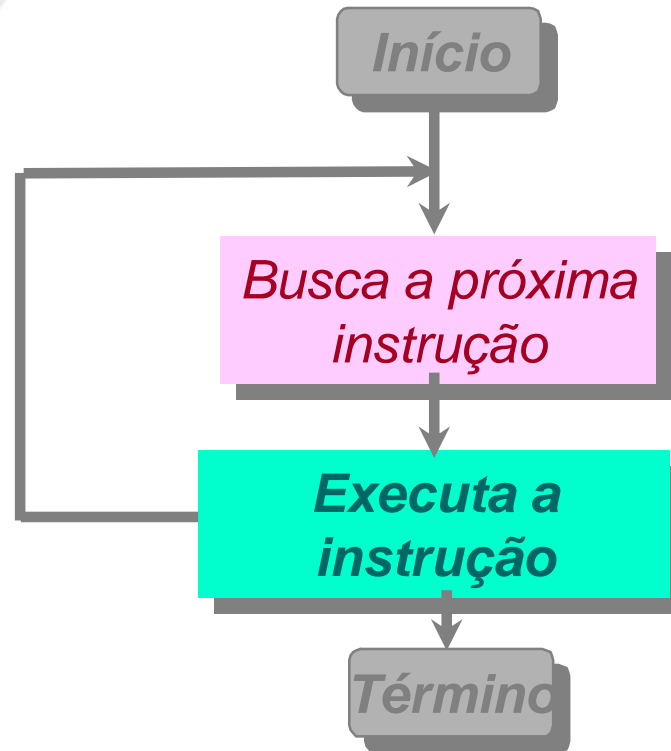
MIPS - Visão Abstrata



Relógio - Clock

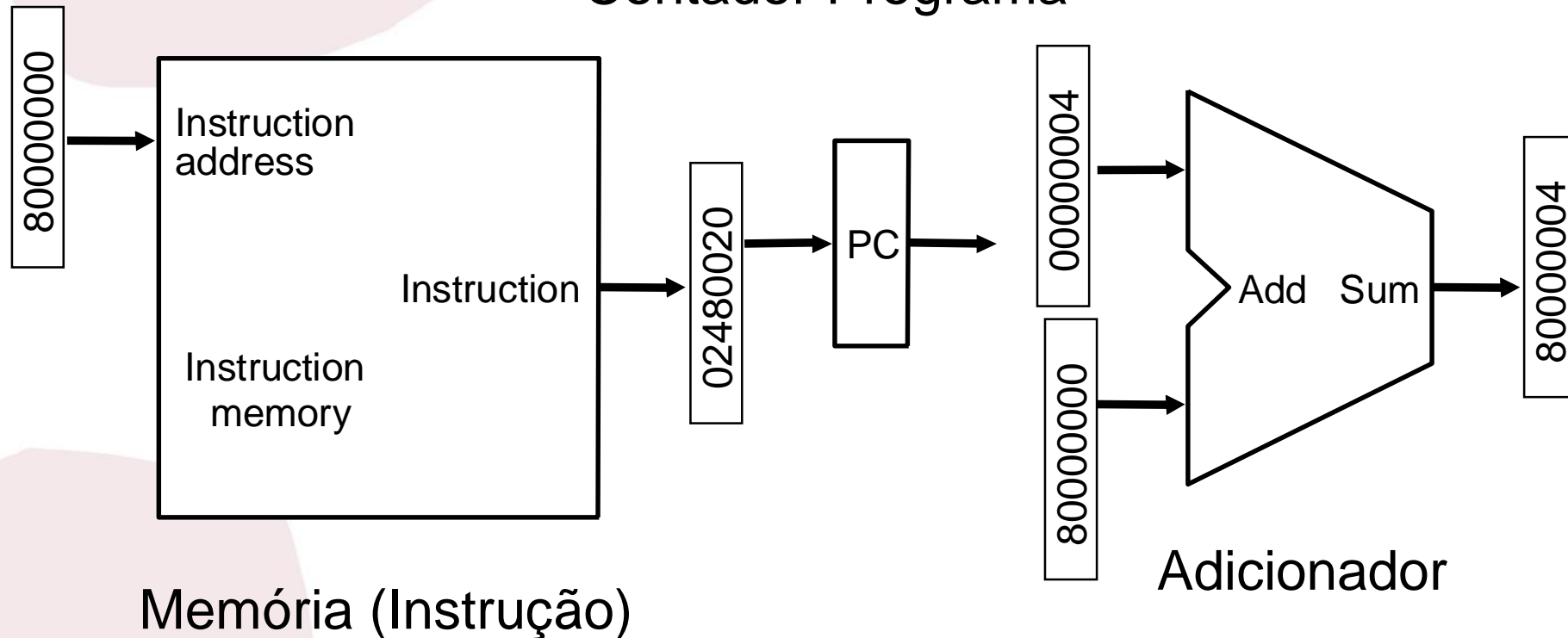


Mono-ciclo

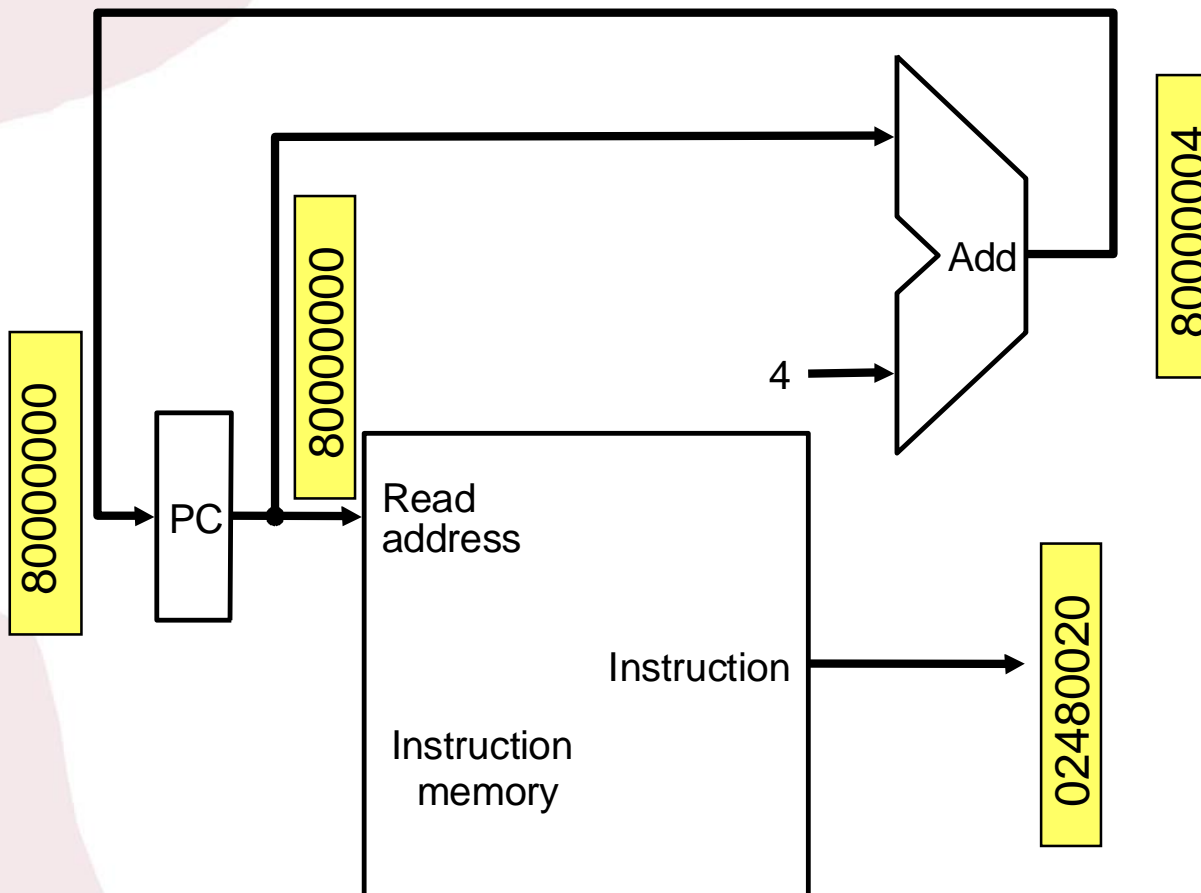


Componentes Básicos: Busca de Instrução

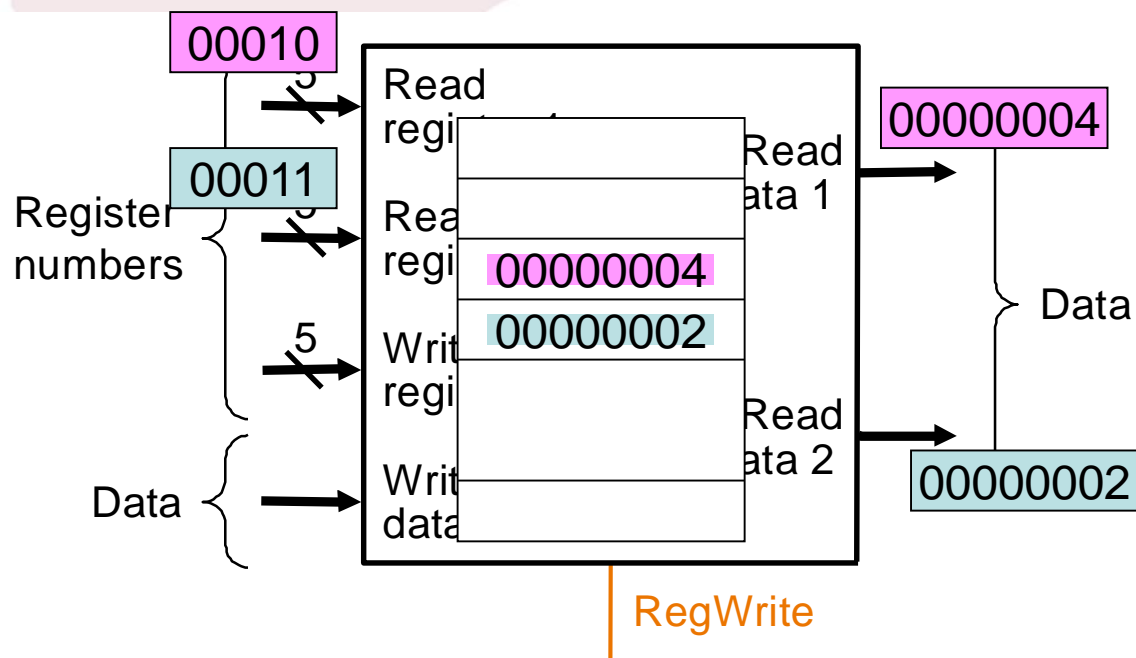
Contador Programa



Busca de Instrução

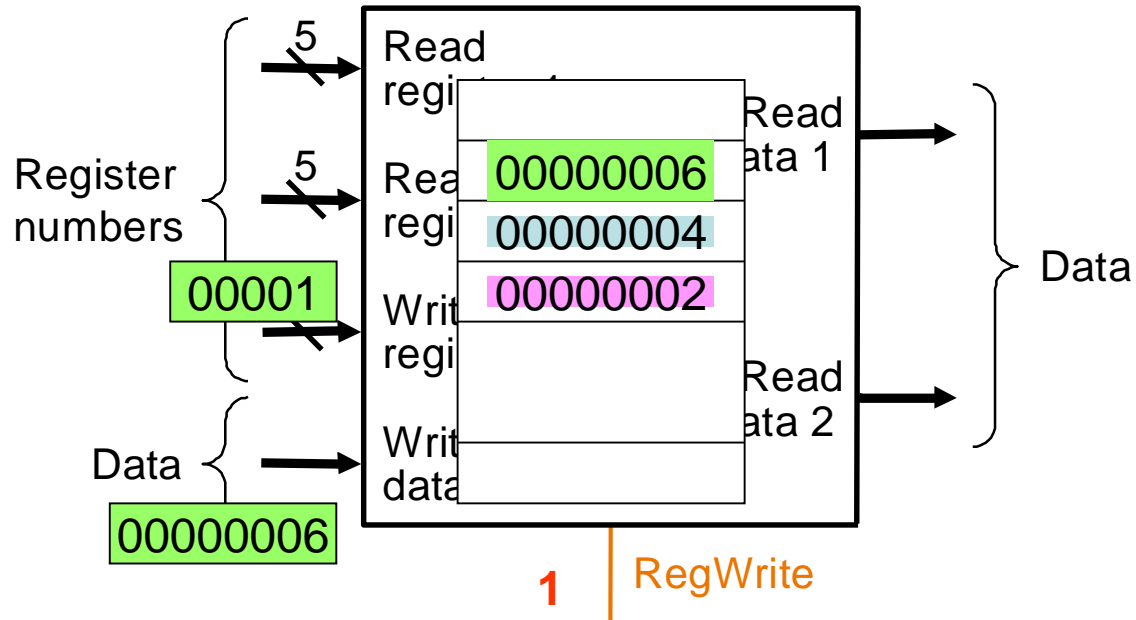


Componentes Básicos: Operações Aritméticas



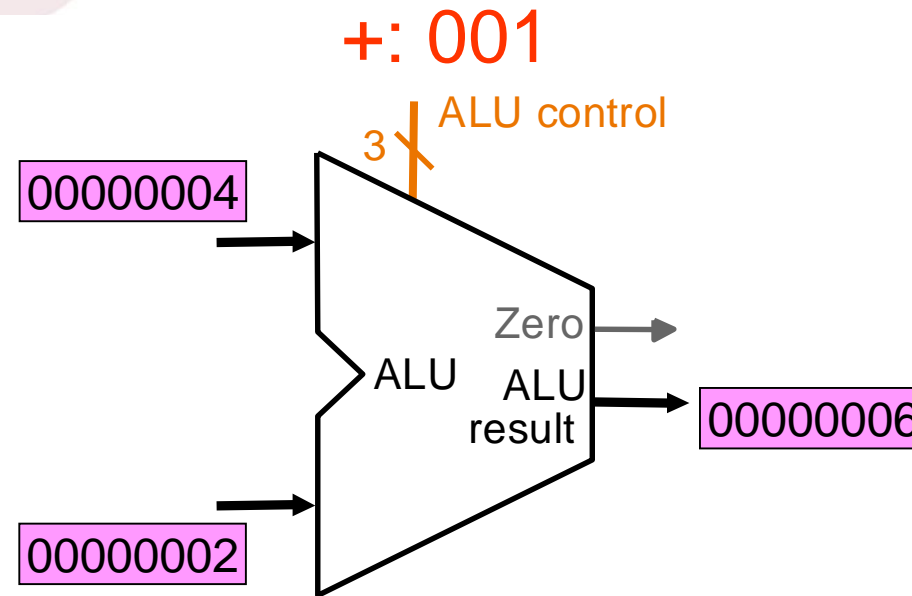
Banco de registradores
Leitura

Componentes Básicos: Operações Aritméticas



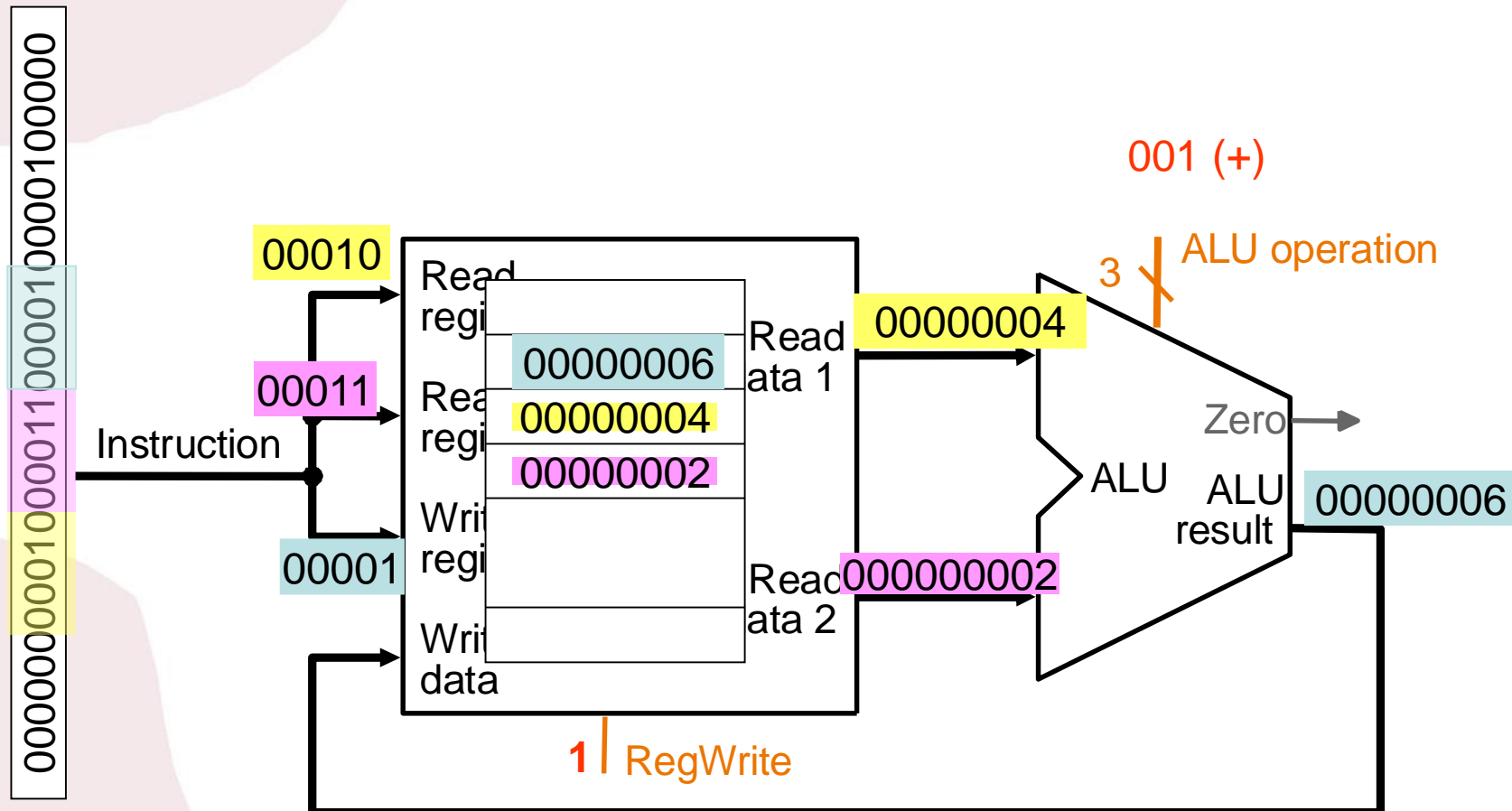
Banco de registradores
Escrita

Componentes Básicos: Operações Aritméticas

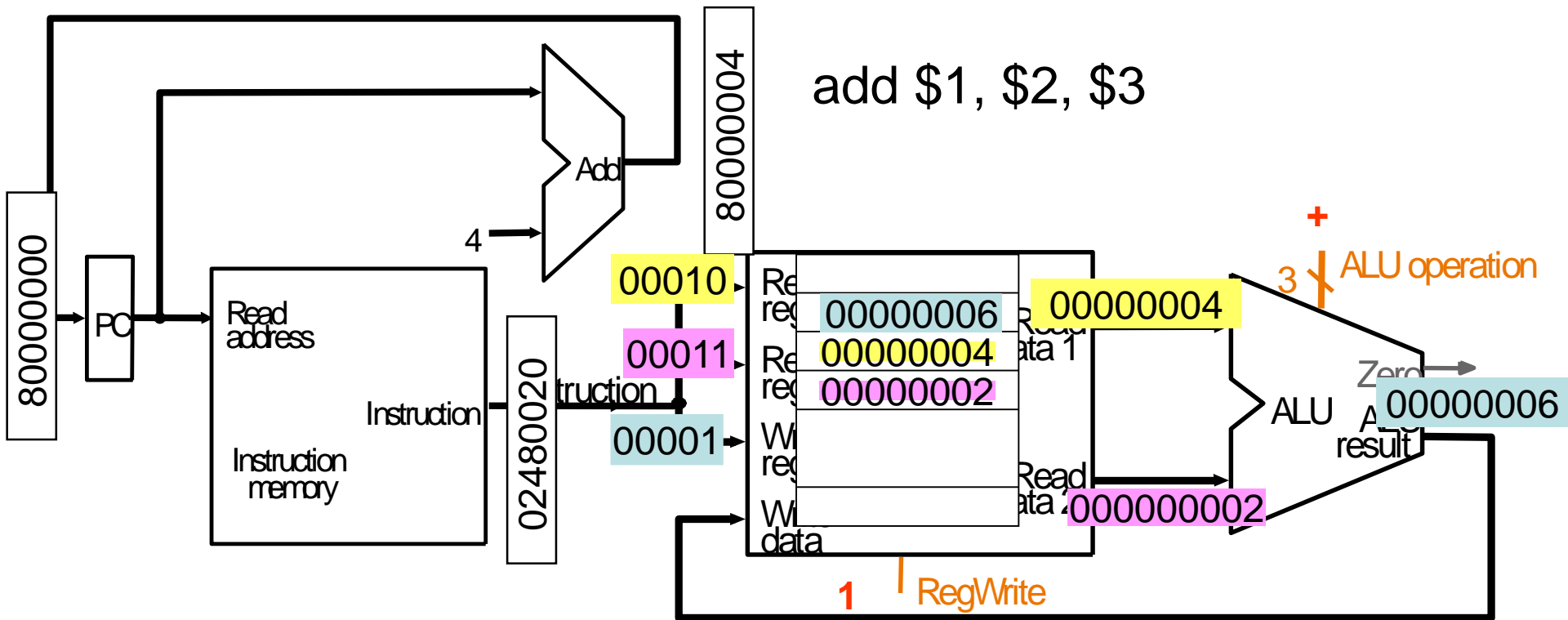


ALU

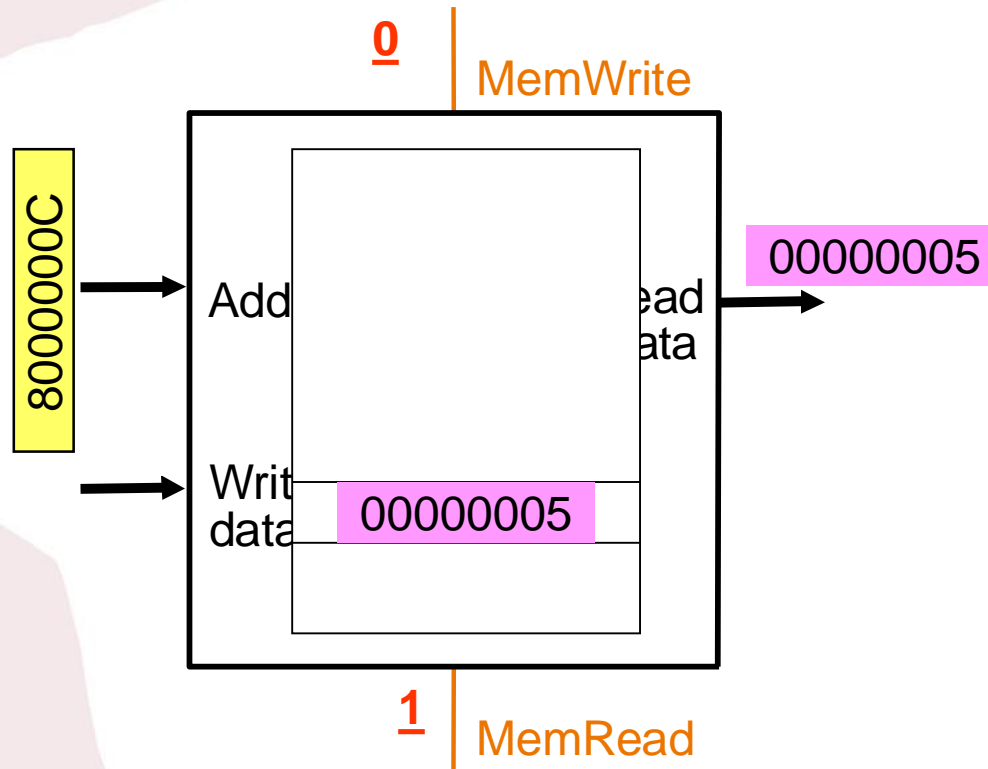
Instruções Aritméticas/Lógicas



Busca e Execução de Instruções Aritméticas/Lógicas

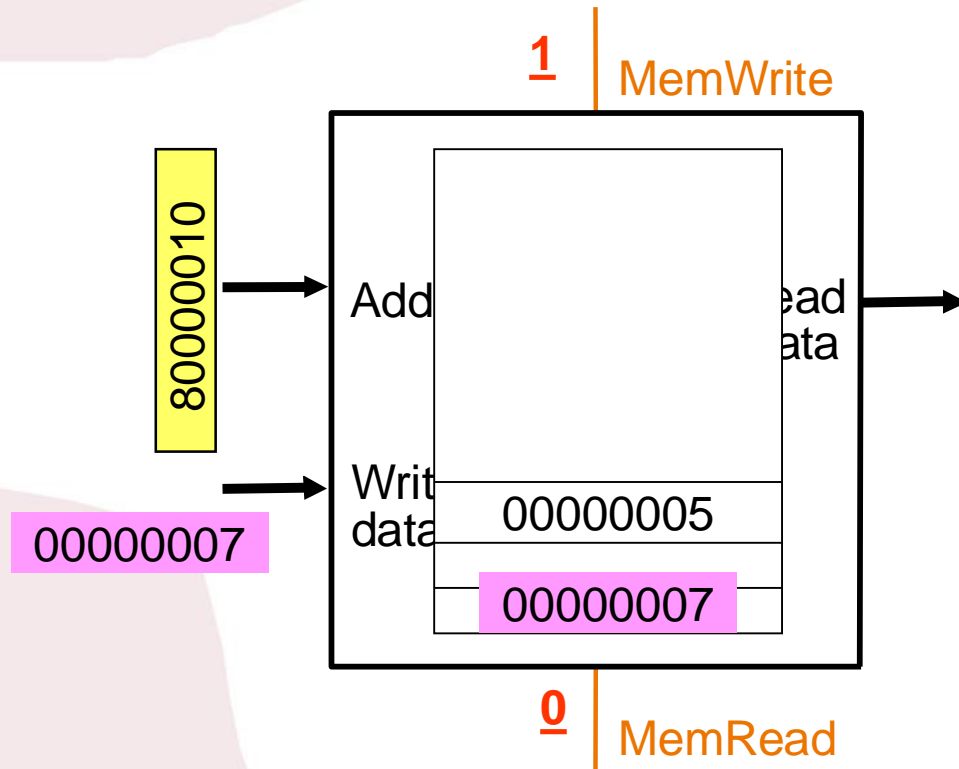


Componentes Básicos: Acesso à memória



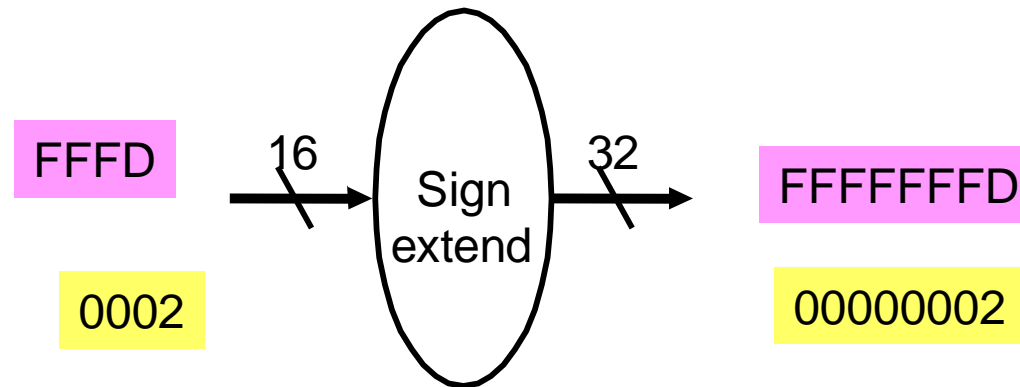
Memória de Dados - Leitura

Componentes Básicos: Acesso à memória



Memória de Dados - Escrita

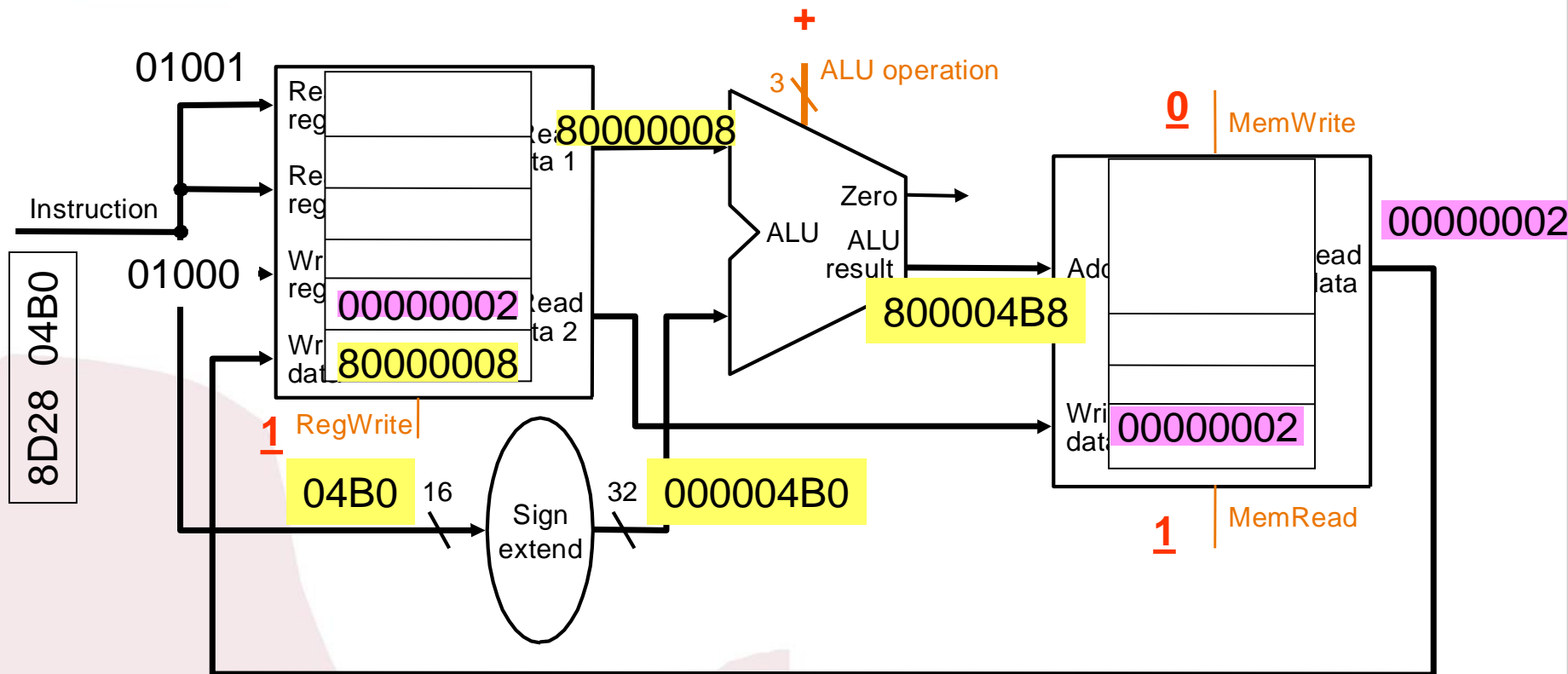
Componentes Básicos: Acesso à memória



Extensão de Sinal

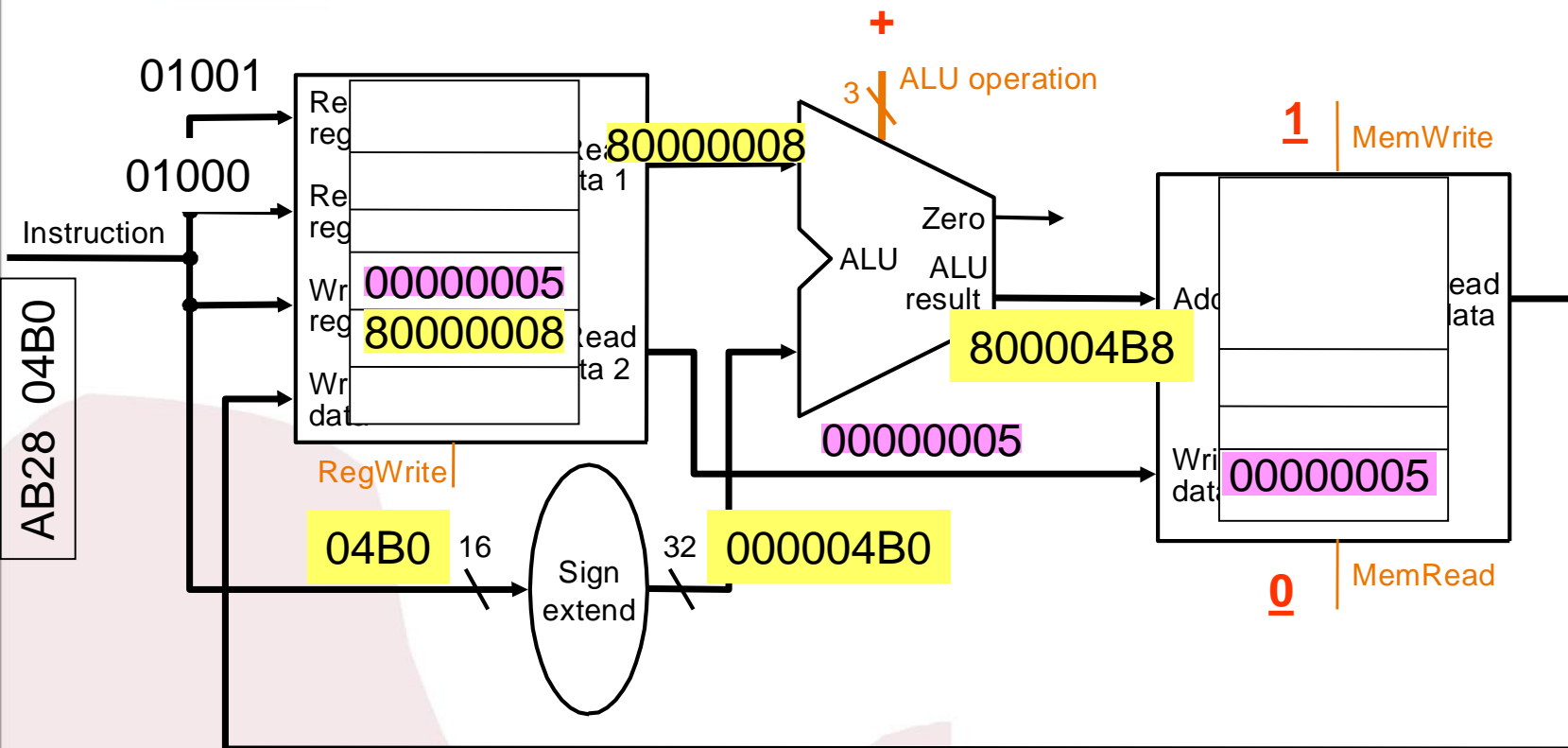
Instruções Load/Store

lw \$8, desl(\$9)



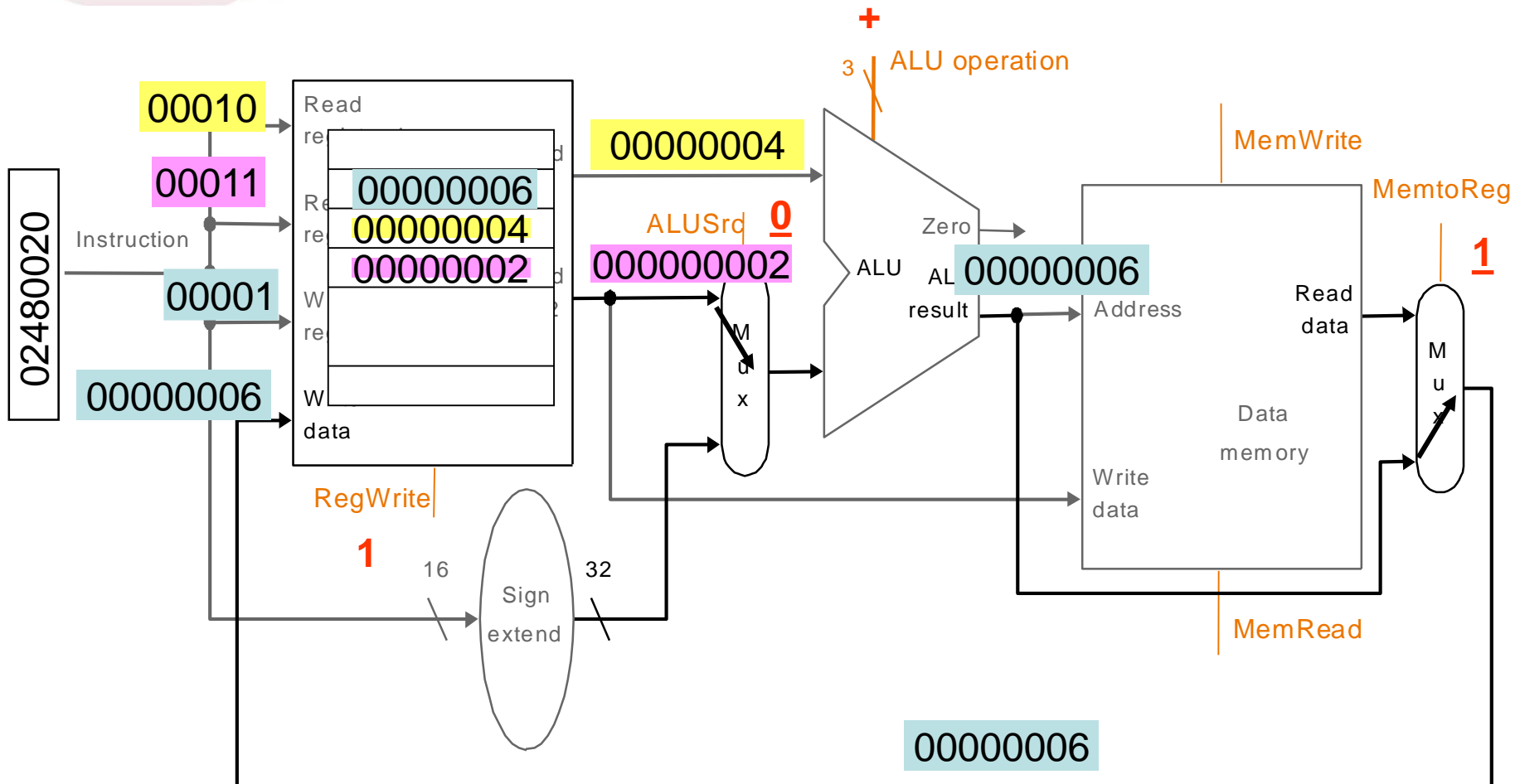
Instruções Load/Store

sw \$8, desl(\$9)



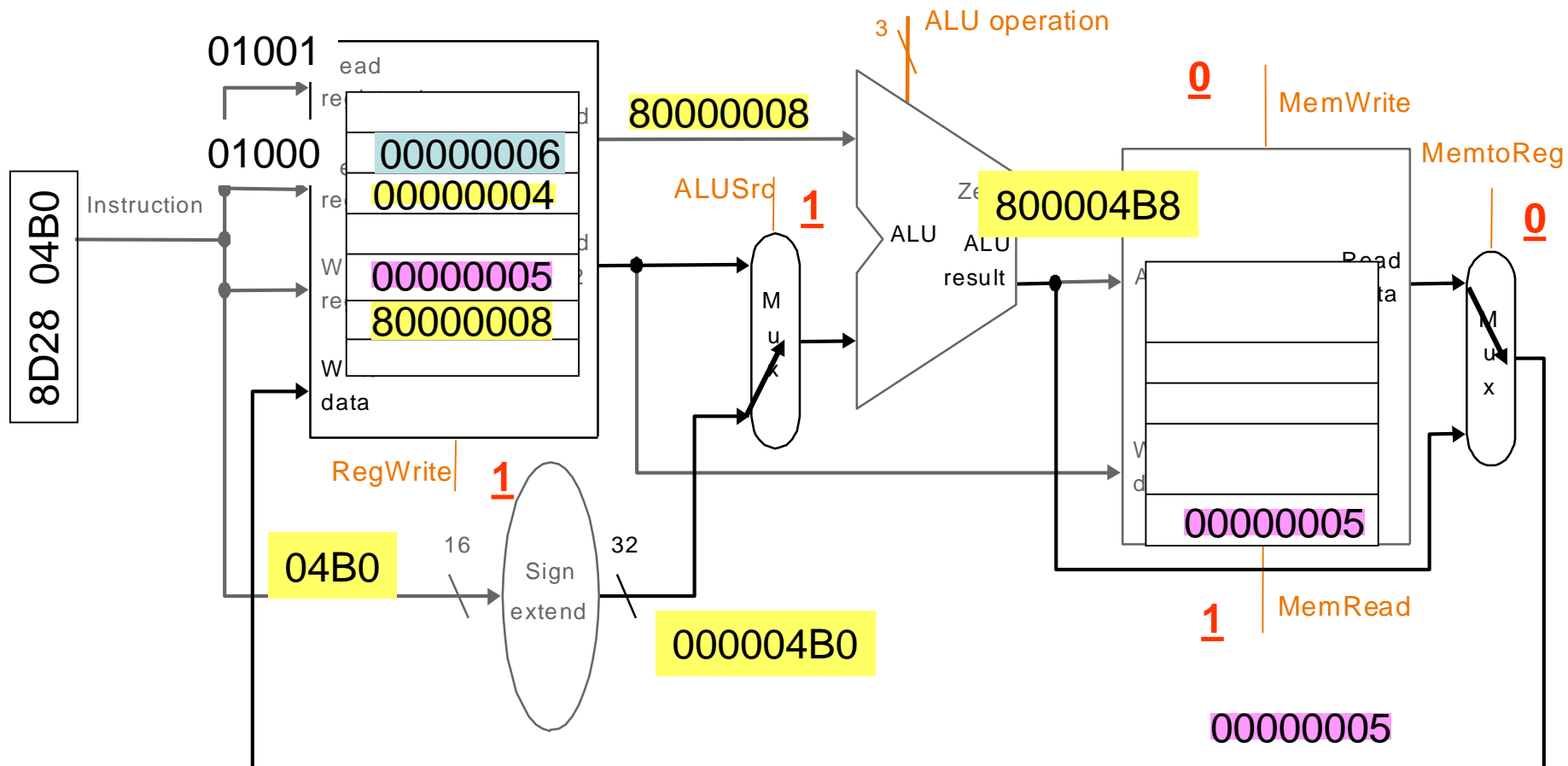
Instruções Aritméticas e de Load/Store

add \$1, \$2, \$3

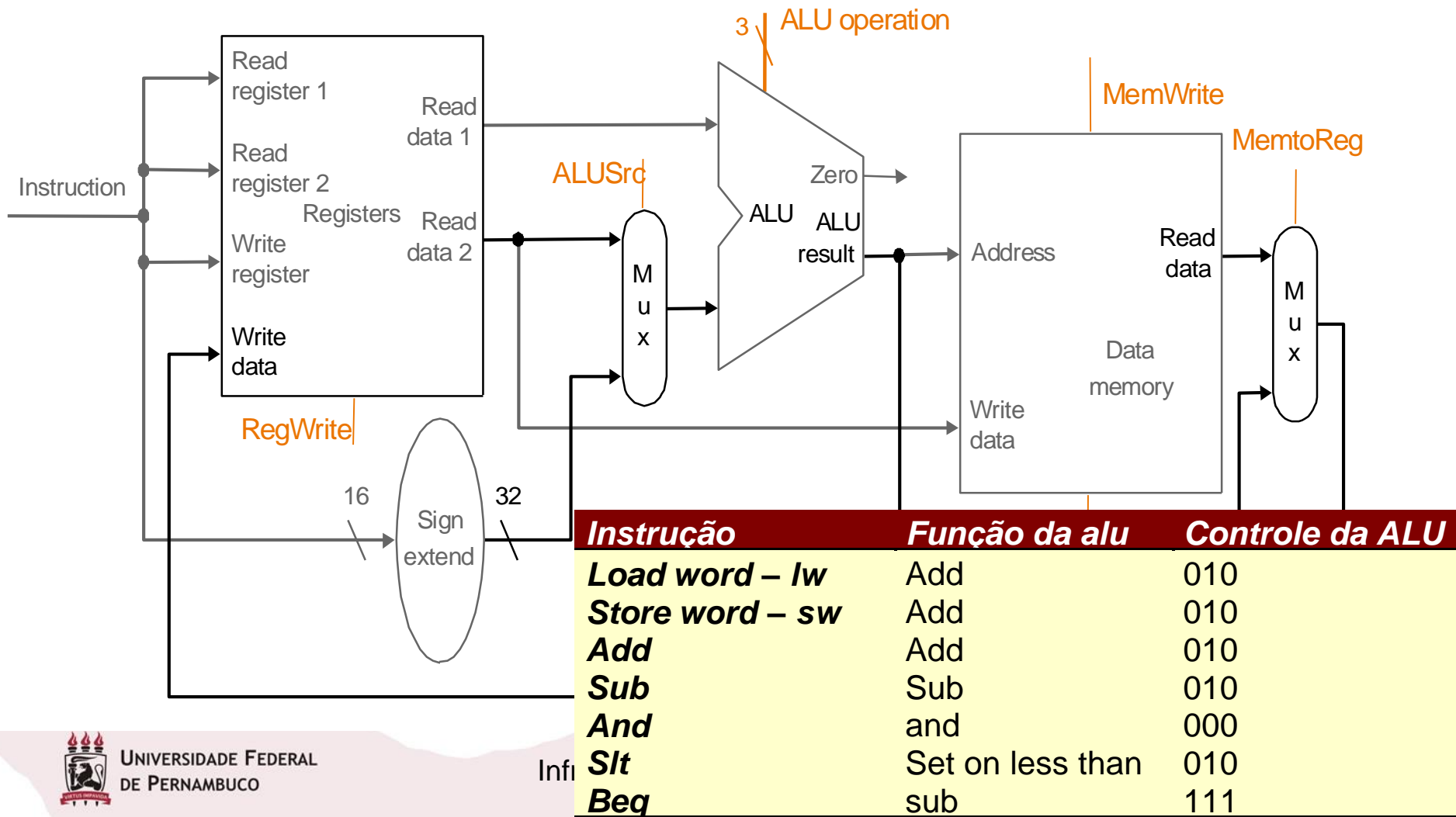


Instruções Aritméticas e de Load/Store

lw \$8, desl(\$9)

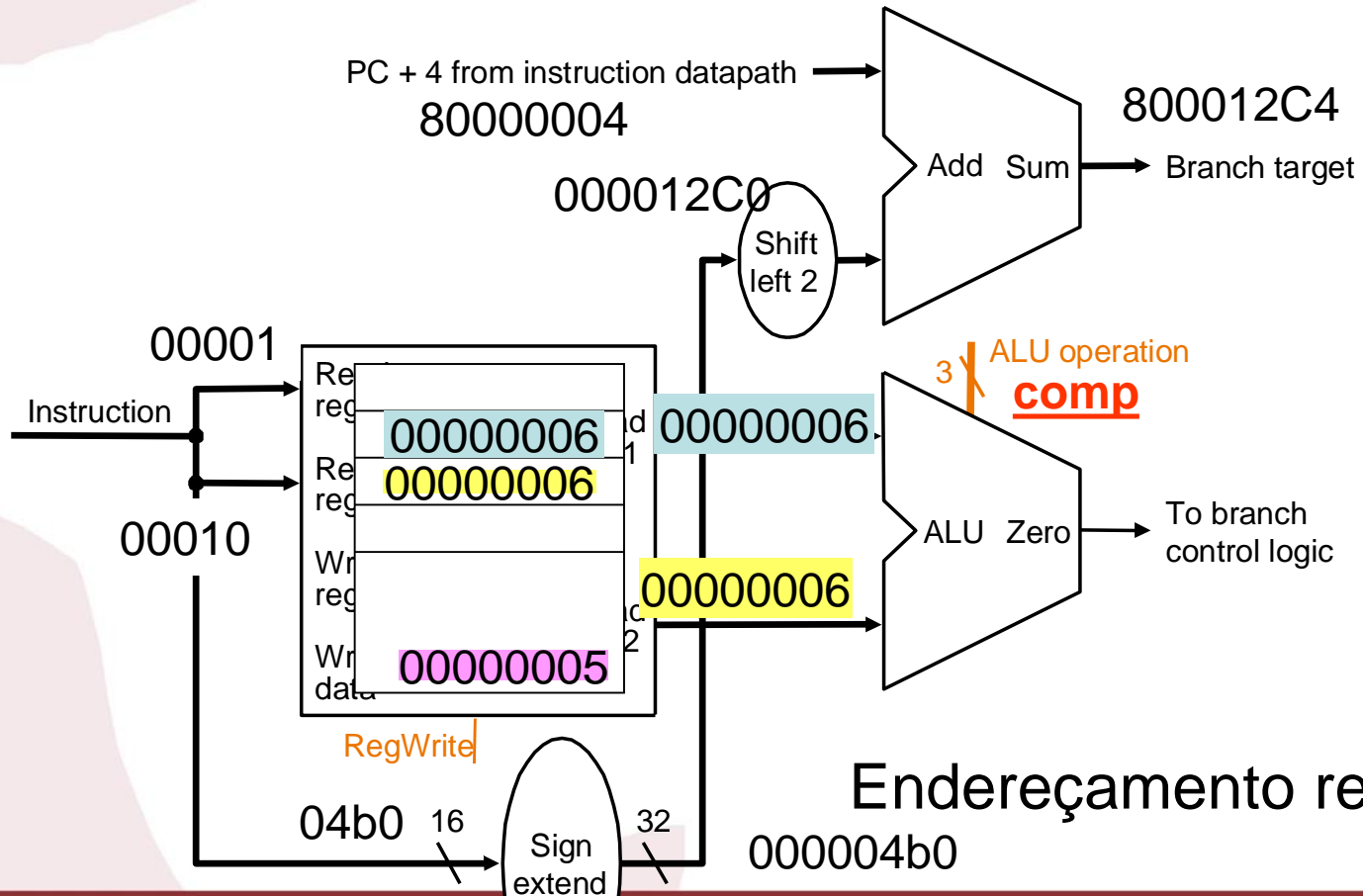


Instruções Aritméticas e de Load/Store

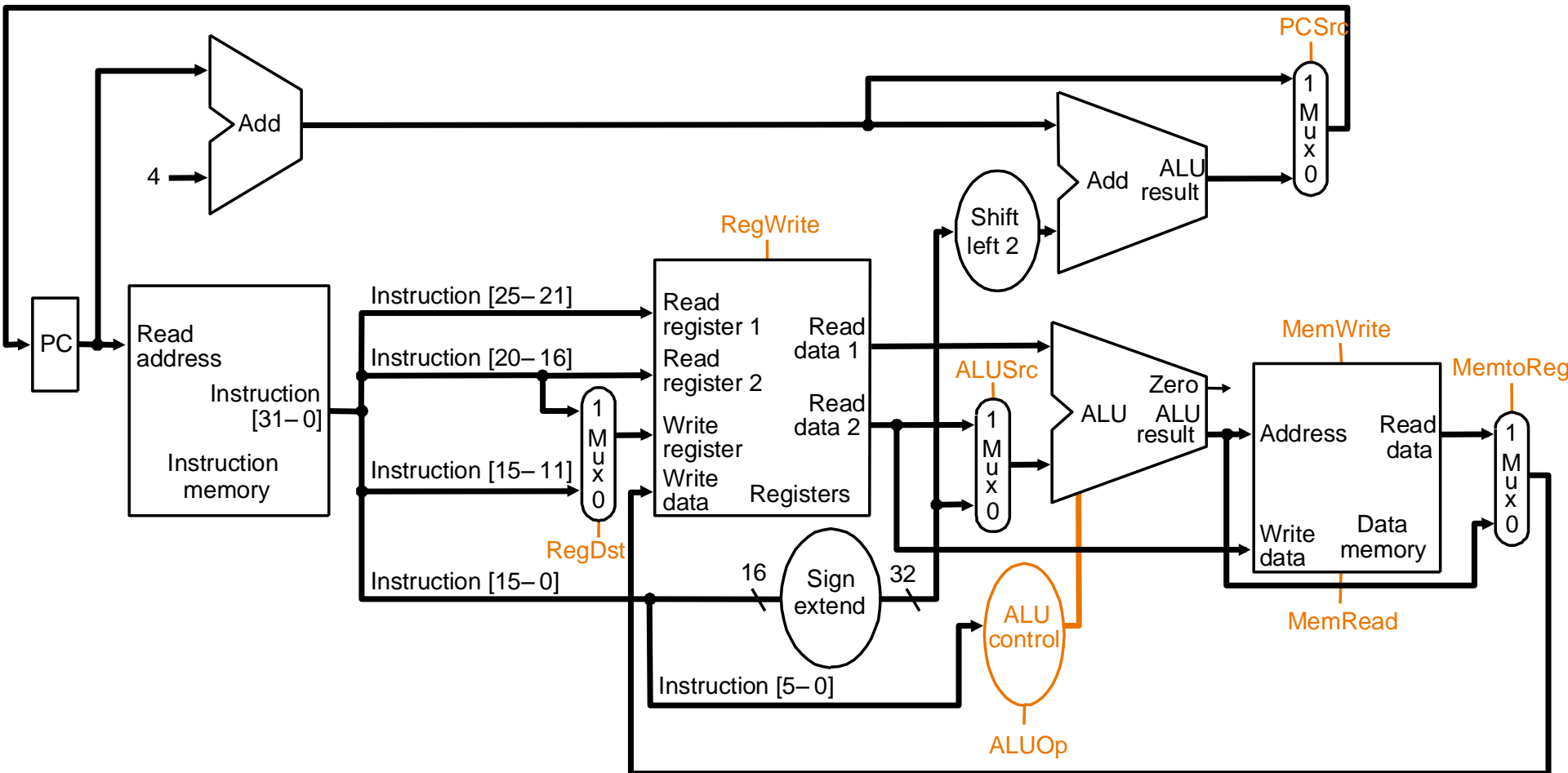


Instrução de Branch on equal

Beq \$1,\$2, end.



Unidade de Processamento



Unidade

Aritm

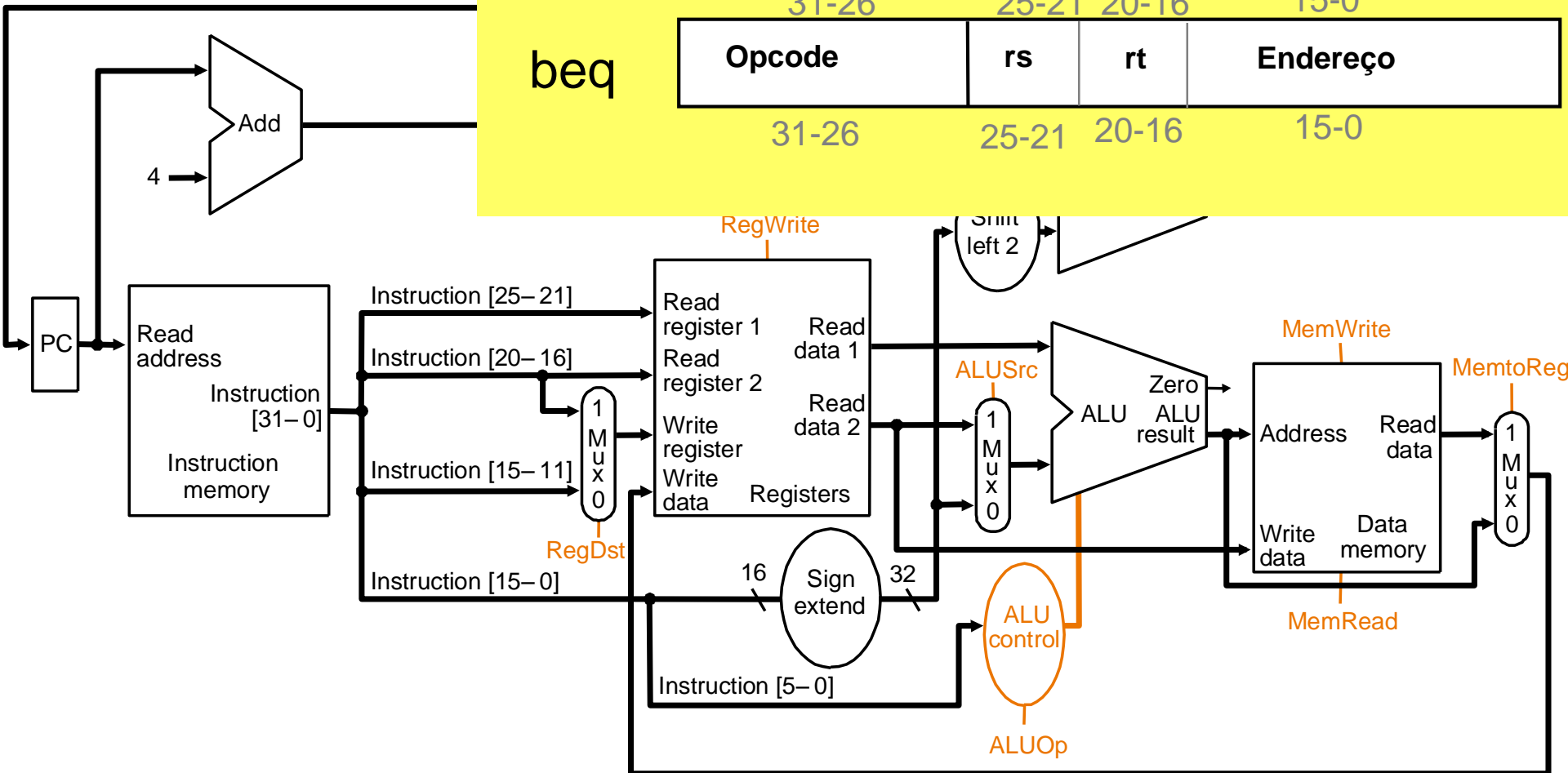
Opcode	rs	rt	rd	shamt	funct
31-26	25-21	20-16	15-11	10-6	5-0

lw/sw

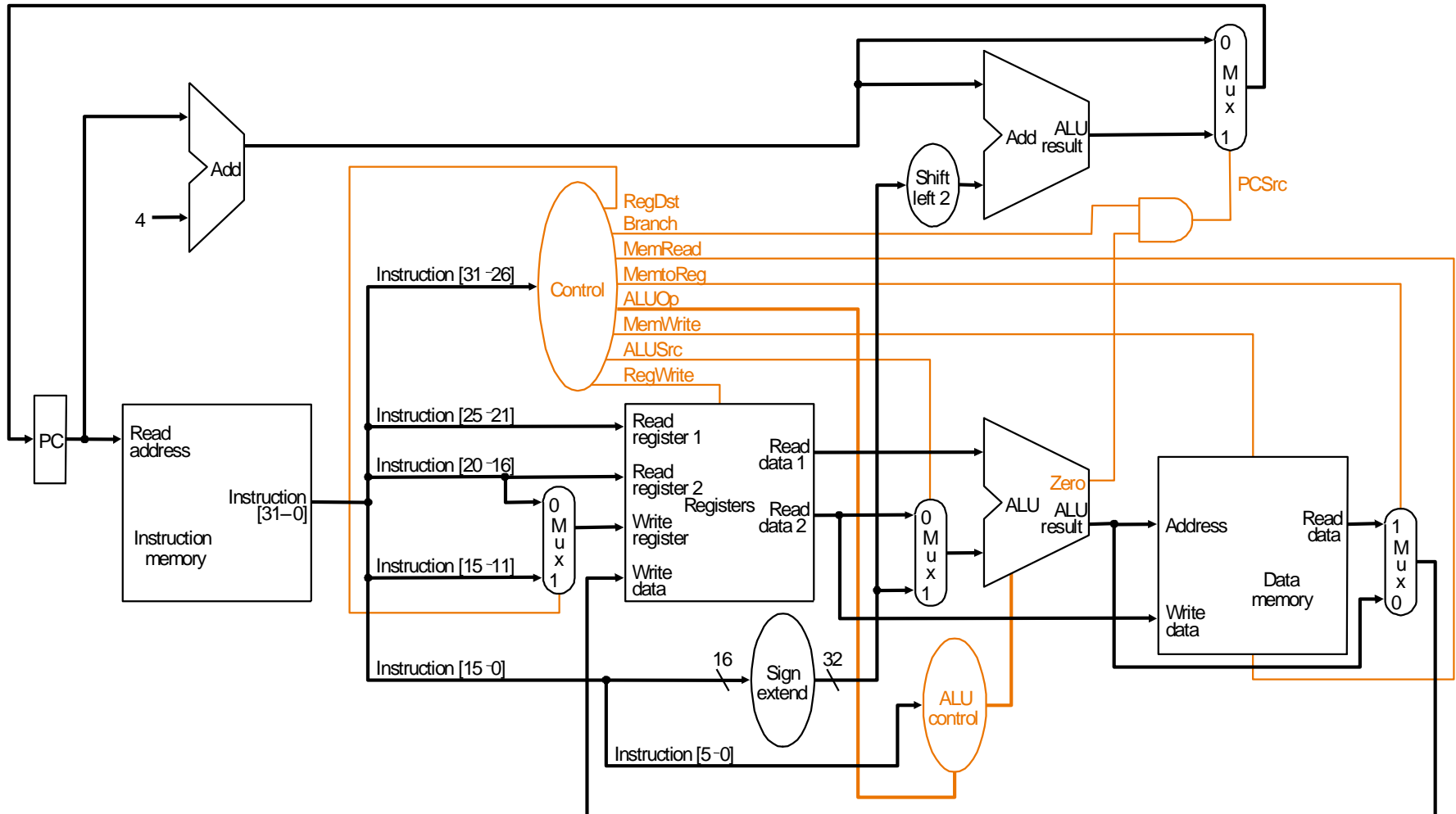
Opcode	rs	rt	Endereço
31-26	25-21	20-16	15-0

beq

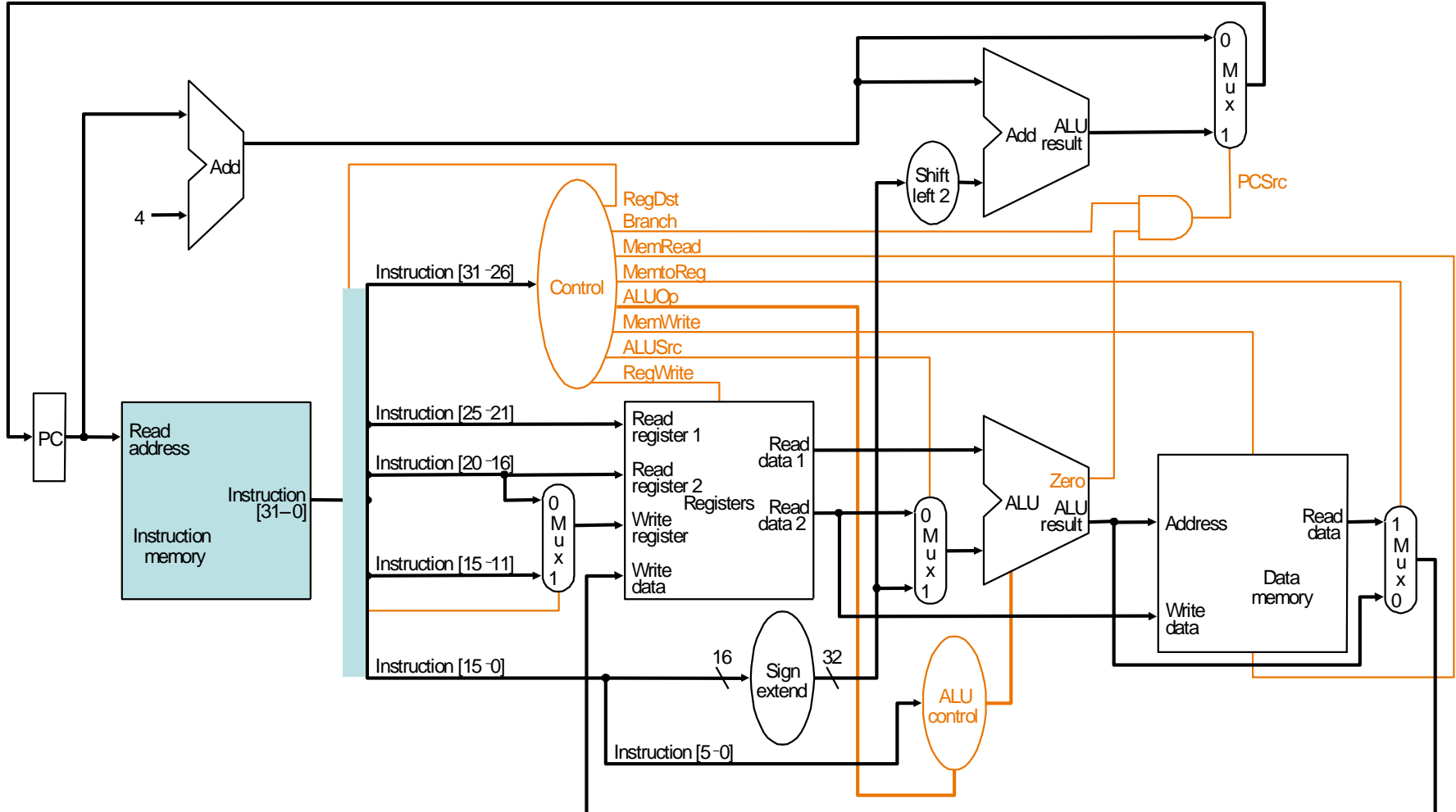
Opcode	rs	rt	Endereço
31-26	25-21	20-16	15-0



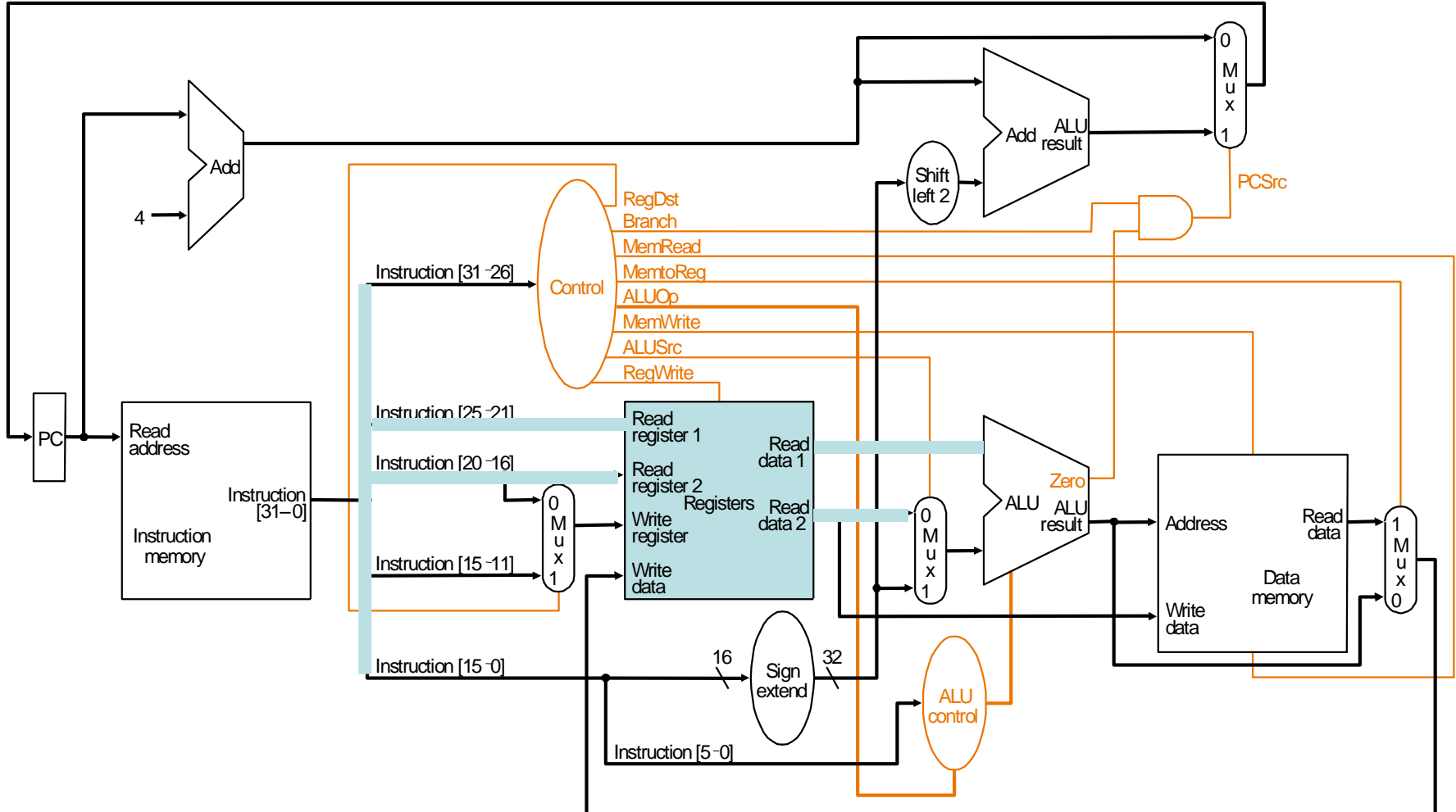
Unidade de Controle



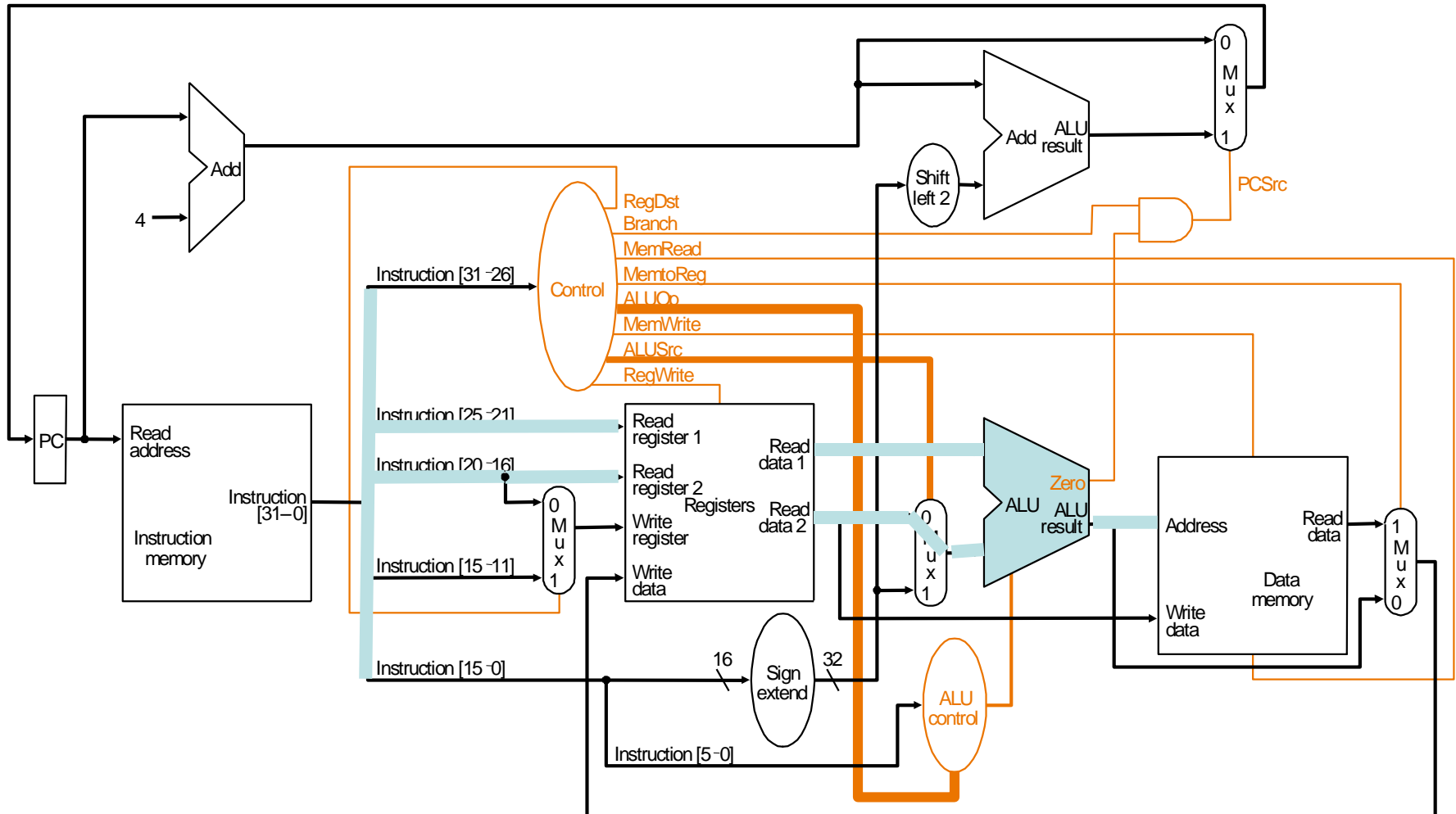
Busca de instrução



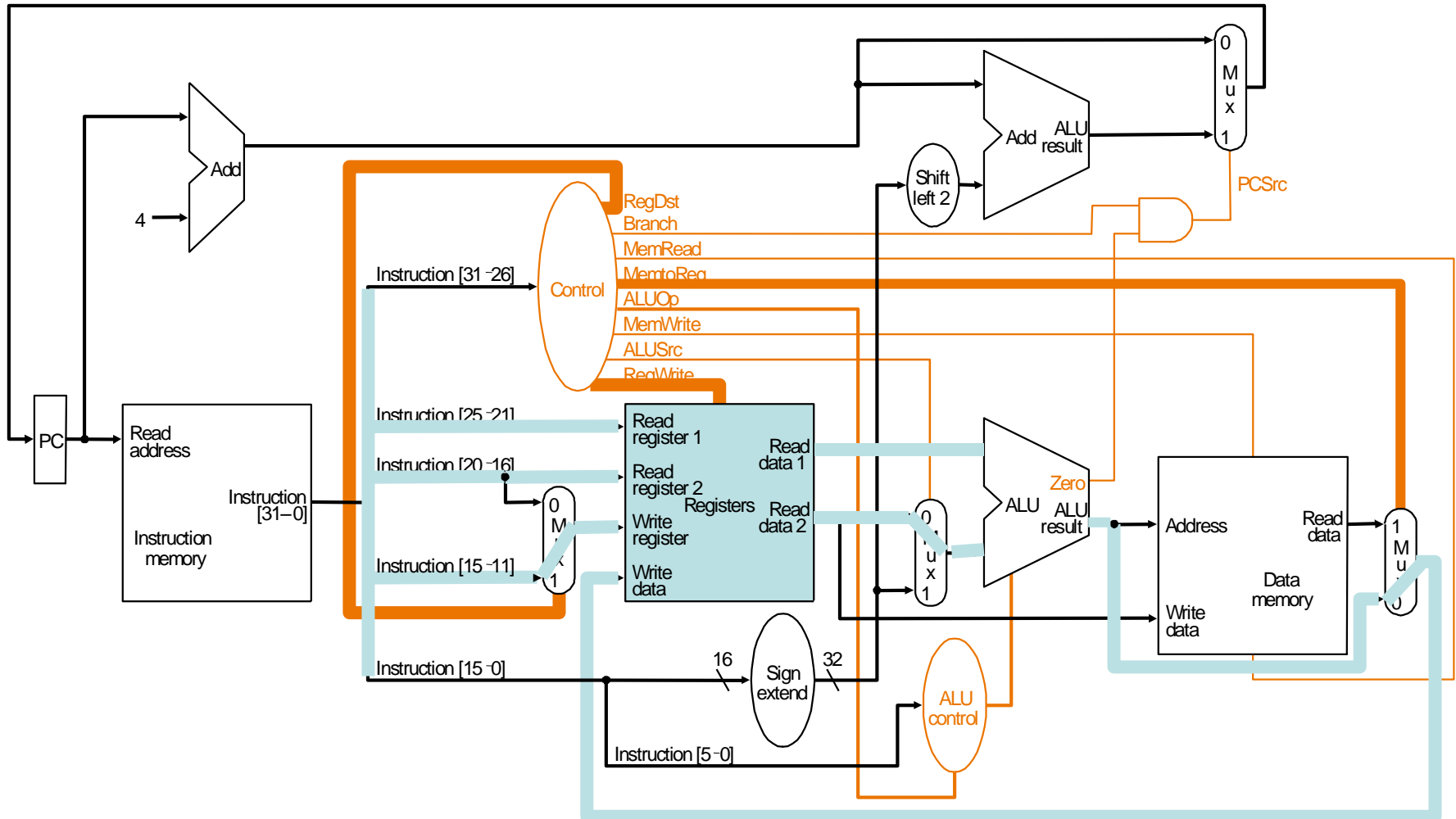
Leitura de registradores



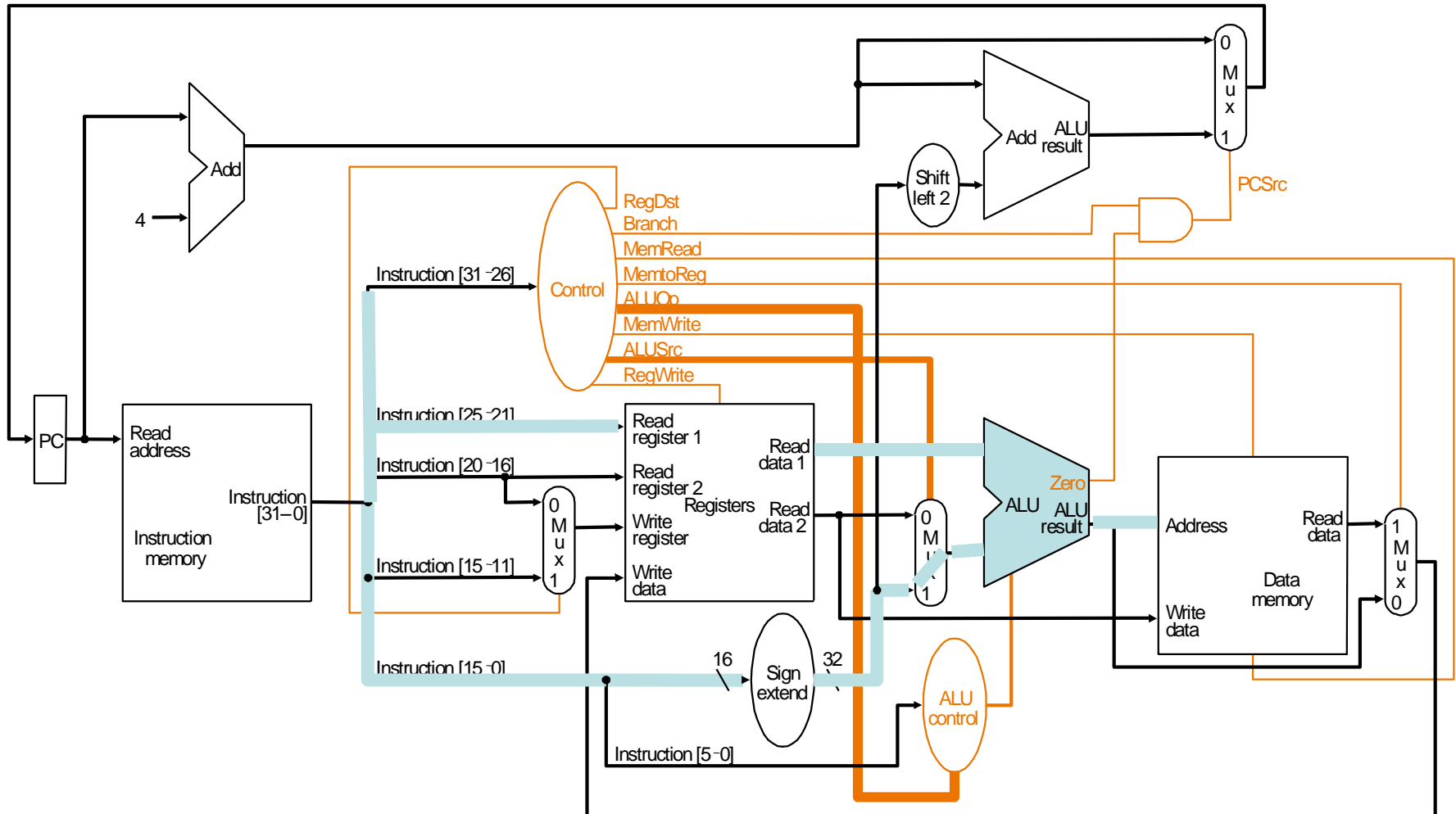
Instr. Aritmética: Operação com ALU



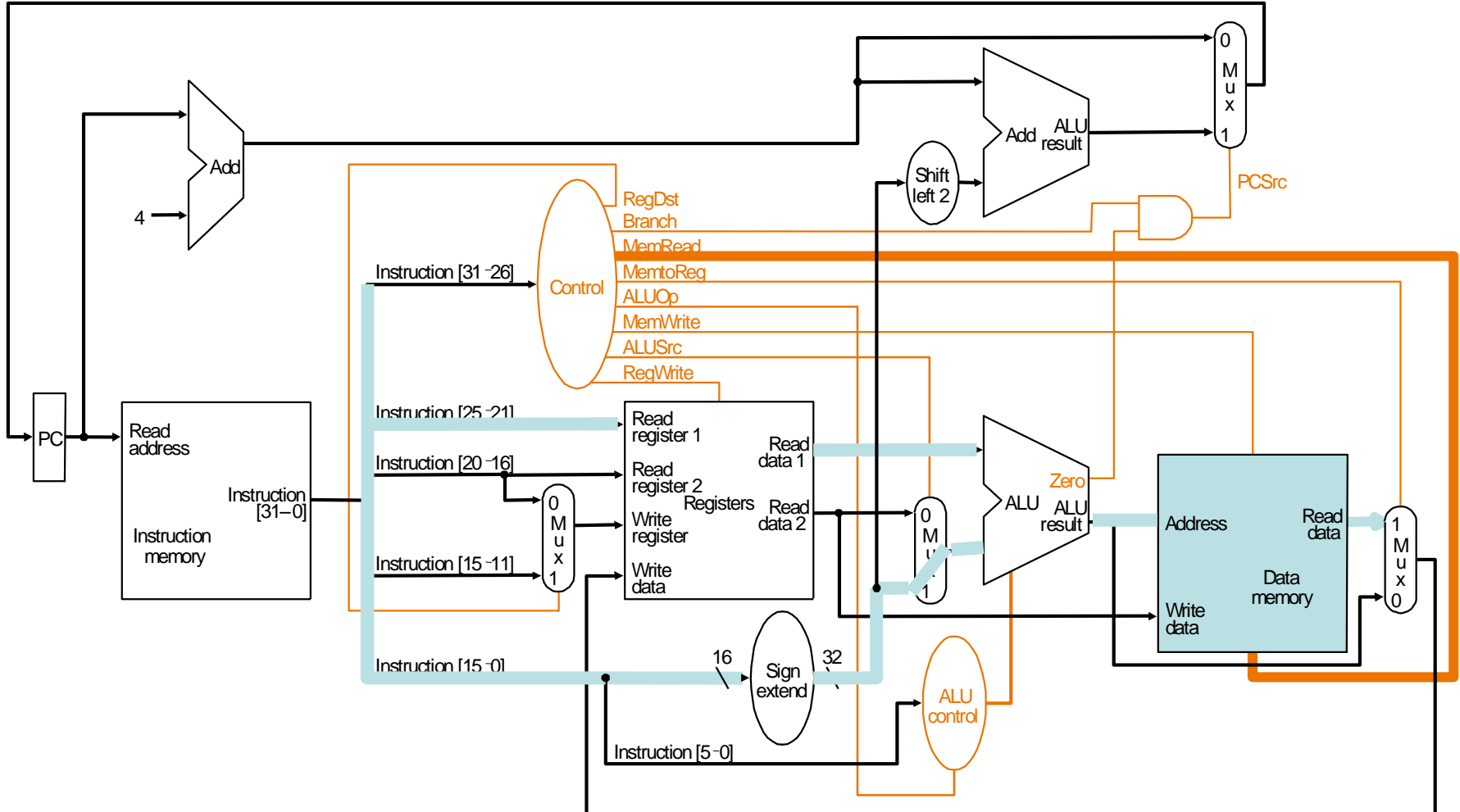
Instr. Aritmética: Escrita Reg.



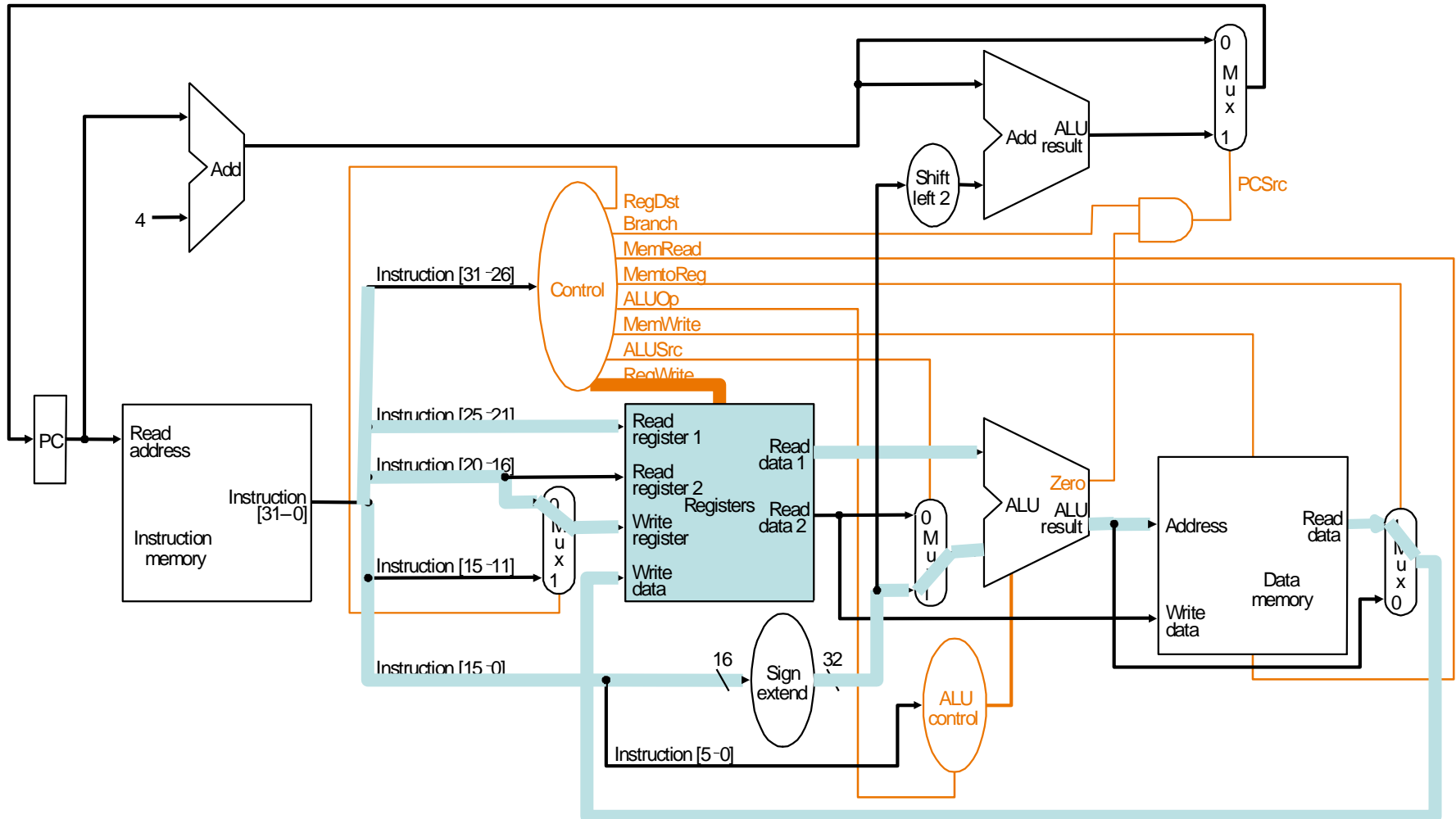
Lw/Sw: Operação com ALU (cálculo do endereço)



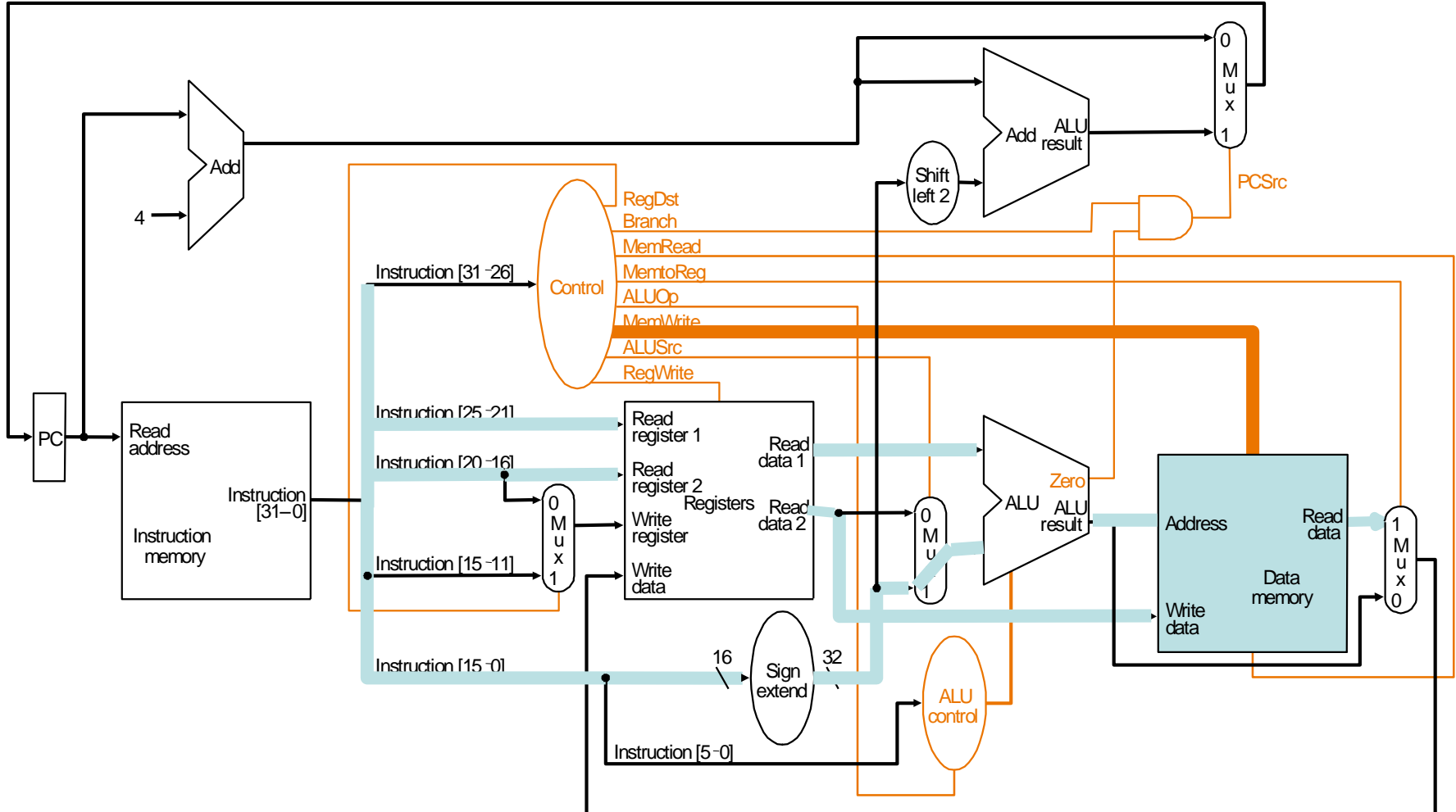
Centro de Informática
U · F · P · E



Lw: Escrita em Registrador

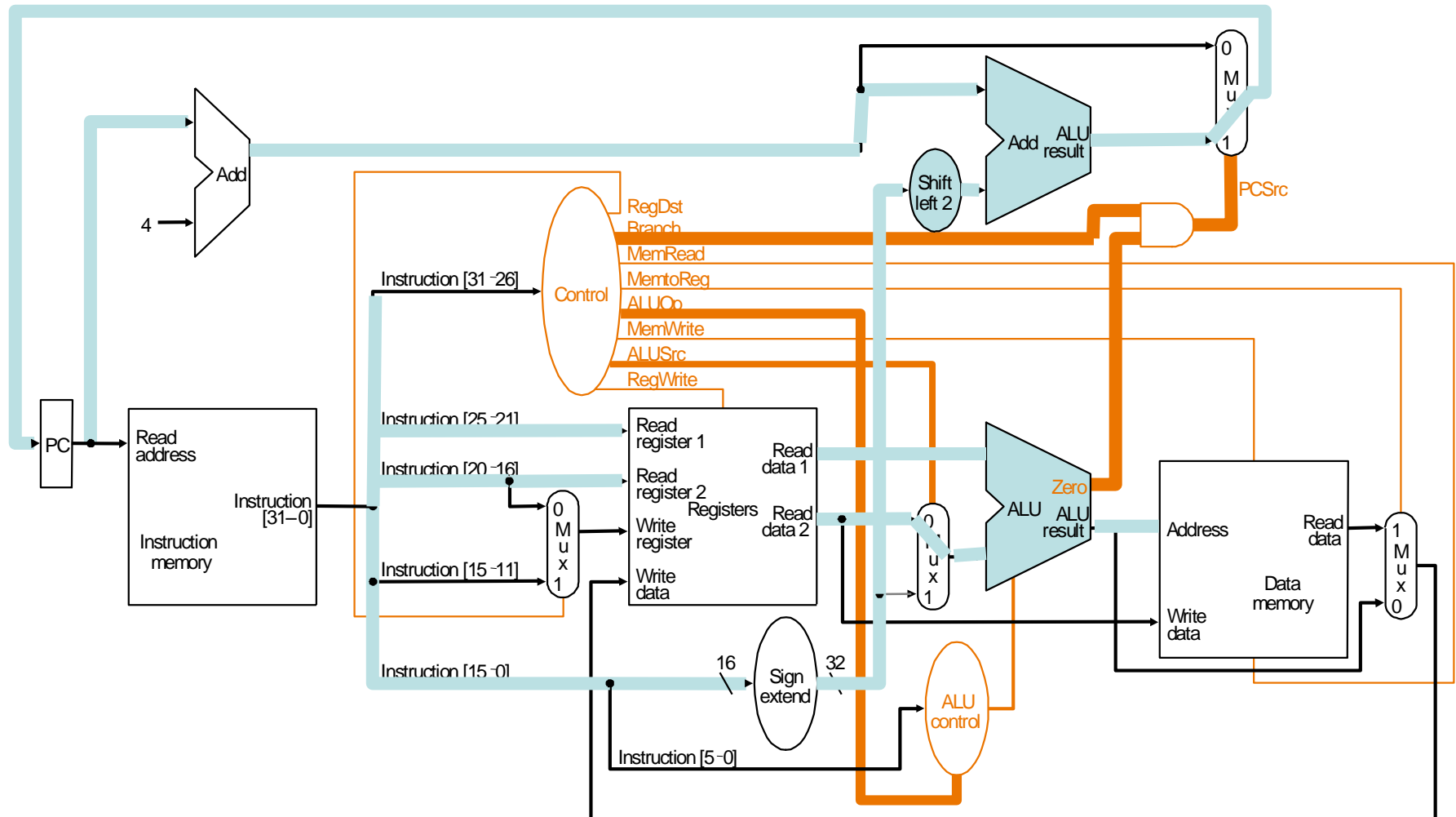


Sw: Escrita de Memória



Beq: Operação com ALU (comparação) e cálculo de endereço de desvio

U · F · P · E



Análise de Desempenho



- Mono-ciclo:
 - Período do relógio definido em função da duração da instrução mais lenta
 - 8ns (5ns)
 - Implementação pouco eficiente
 - $\text{CPU}_{\text{time}} = \text{nr. Instruções} \times \text{período_clock}$
- Como melhorar o desempenho na execução de várias instruções?
 - Multi-ciclo: cada estágio é executado em um ciclo do relógio.
 - Começar uma instrução ANTES da última instrução iniciada terminar.