

Organização e Arquitetura de Computadores I

Caminho de Dados
Exercício Multicilco

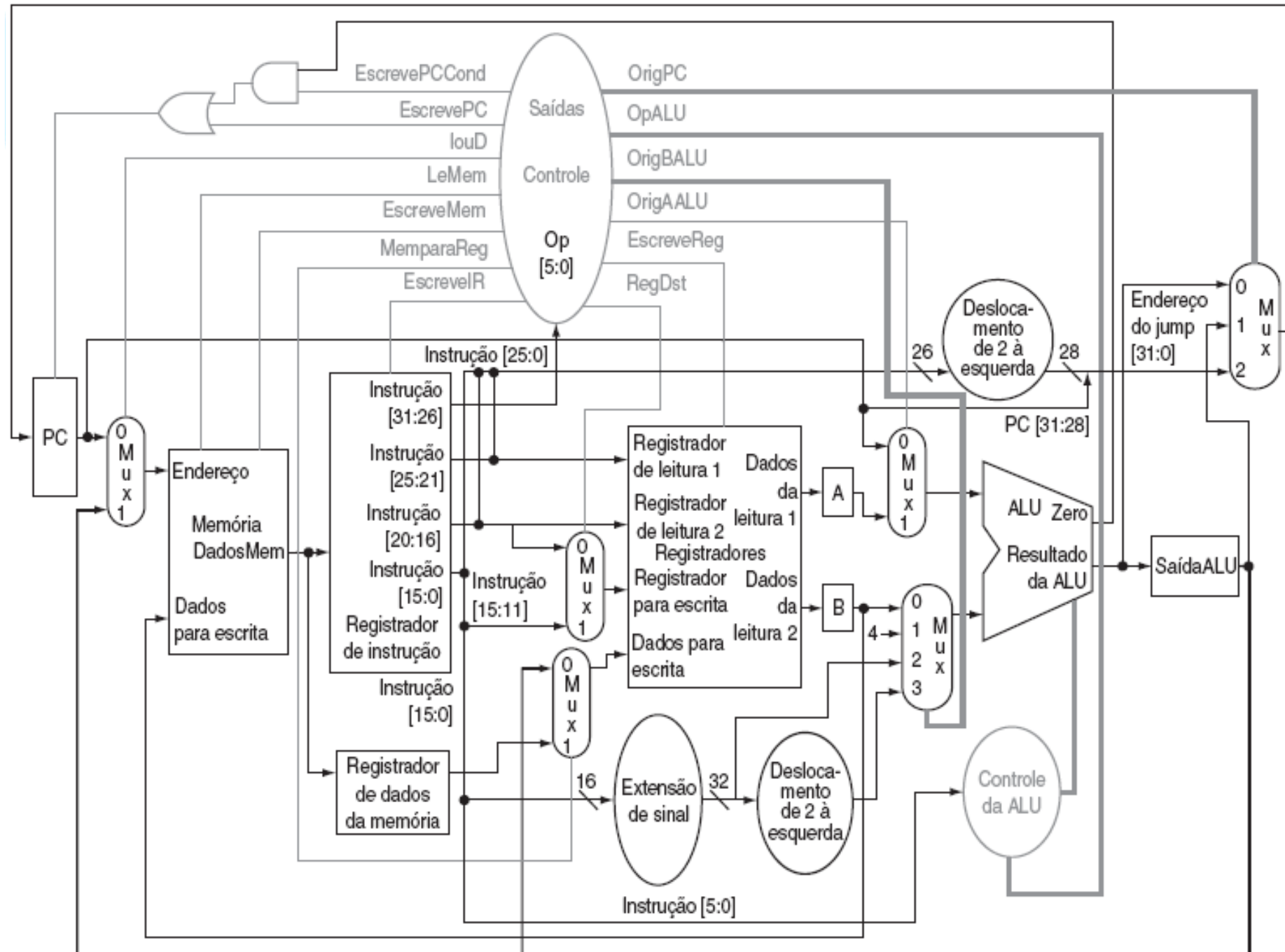
Exercício Multiciclo

● Montar uma tabela contendo os passos de execução das instruções abaixo e com os respectivos bits de controle para cada ciclo:

- add
- beq
- lw
- sw
- j
- addi

Uma Implementação Multiciclo

- Registradores temporários adicionados:
 - O **registrador de instrução (IR)** e o **registrador de dados da memória (MDR)** são incluídos para salvar a saída da memória para uma leitura de instrução e uma leitura de dados, respectivamente;
 - Os registradores **A** e **B** são usados para conter os valores dos registradores operandos lidos do banco de registradores;
 - O registrador **saídaALU** contém a saída da ULA.
- Todos os registradores exceto o IR contêm dados apenas entre um par de ciclos de *clock* adjacentes e, portanto, não precisarão de um sinal de controle de escrita.



Organização e Arquitetura de Computadores I

Uma Implementação Multiciclo

Nome do sinal	Valor	Efeito
OpALU	00	A ULA efetua uma adição
	01	A ULA efetua uma subtração
	10	O campo funct da instrução determina a operação
OrigBALU	00	A segunda entrada da ULA vem do registrador B
	01	A segunda entrada da ULA é a constante 4
	10	A segunda entrada da ULA são os 16 bits menos significativos com sinal estendido do IR
	11	A segunda entrada da ULA são os 16 bits menos significativos com sinal estendido do IR deslocados de 2 bits para a esquerda
OrigPC	00	A saída da ULA (PC +4) é enviada ao PC para escrita
	01	O conteúdo da SaídaALU (endereço de destino do desvio) é enviado ao PC para escrita
	10	O endereço de destino do jump (IR[25:0]) deslocado de 2 bits para a esquerda e concatenado com PC + 4[31:28] é enviado ao PC para escrita

Organização e Arquitetura de Computadores I

Etapas de Execução

1. Etapa de busca da instrução

- Buscar a instrução na memória e calcular o endereço da próxima instrução sequencial.
- Descrição na RTL (*Register-Transfer Language*)

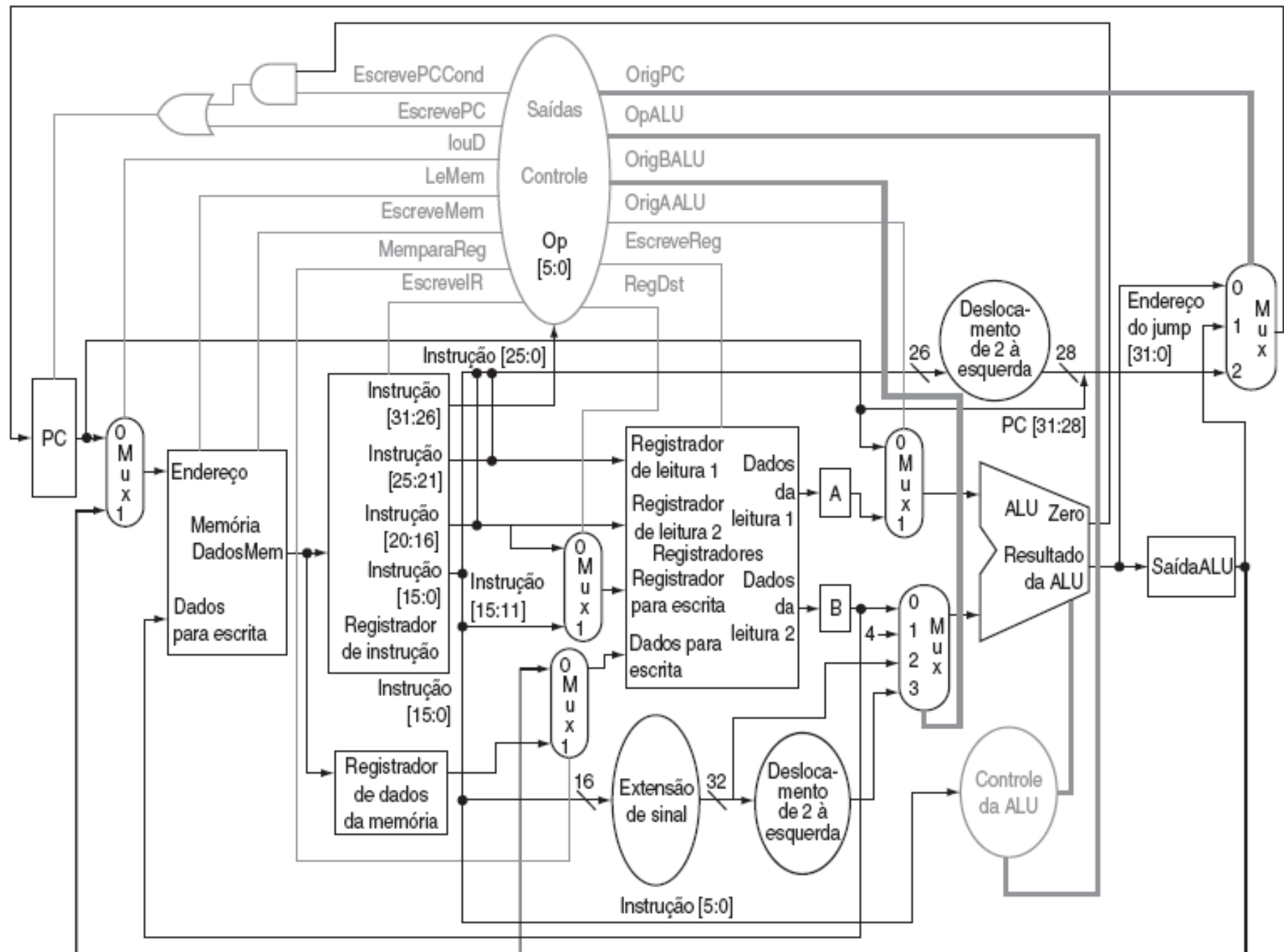
$IR \leq Memória[PC];$

$PC \leq PC + 4;$

- Operação:

- Enviar o PC para a memória como o endereço, realizar uma leitura e escrever a instrução no Registro de Instrução (IR), onde ele será armazenado.

RegDst	Escreve Reg	OrigAAL U	LeMem	Escreve Mem	MemparaReg	loutD	Escreve R	Escreve PC	Escreve PCCond	OpALU	OrigBAL U	OrigPC
		0	1			0	1	1		00	01	00



Organização e Arquitetura de Computadores I

Etapas de Execução

1. Etapa de busca da instrução

- O incremento do PC e o acesso à memória de instruções podem ocorrer em paralelo.
- O novo valor do PC não é visível até o próximo ciclo de *clock*.

RegDst	Escreve Reg	OrigAAL U	LeMem	Escreve Mem	MemparaReg	loutD	Escreve R	Escreve PC	Escreve PCCond	OpALU	OrigBAL U	OrigPC
		0	1			0	1	1		00	01	00

Organização e Arquitetura de Computadores I

Etapas de Execução

2. Etapa decodificação da instrução e busca dos registradores

- Nesta etapa ainda não foi identificada qual é a instrução e, portanto, só são realizadas ações aplicáveis a todas as instruções ou que não sejam nocivas, caso a instrução não seja o que se espera:
 - Acessar o banco de registradores, ler os registradores rs e rt, e os armazenar nos registradores A e B;
 - Calcular o endereço de destino do desvio com a ULA, o resultado é salvo na SaídaALU (útil se a instrução for um desvio condicional);

$A \leq \text{Reg}[\text{IR}[25:21]];$

$B \leq \text{Reg}[\text{IR}[20:16]];$

$\text{SaídaALU} \leq \text{PC} + (\text{estende-sinal } (\text{IR}[15:0] \ll 2);$

RegDst	Escreve Reg	OrigAALU	LeMem	Escreve Mem	MemparaReg	loutD	Escreve R	Escreve PC	Escreve PCCond	OpALU	OrigBALU	OrigPC
		0								00	11	

Organização e Arquitetura de Computadores I

Etapas de Execução

lw \$t2, 0(\$t3)

lw \$t3, 4(\$t3)

beq \$t2, \$t3, Label #considere not

add \$t5, \$t2, \$t3

sw \$t5, 8(\$t3)

Label: ...