

CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Arquitetura de Computadores I – Turmas 4 e 5 (EARTE) – 2020/2

Prof. Rodolfo da Silva Villaca – rodolfo.villaca@ufes.br

Segunda Prova – 22 de abril de 2021

NOME:	
MATRÍCULA:	

Importante: Para esta prova considere que o seu número de matrícula na UFES pode ser representado pelo formato 20*****YX, sendo Y e X inteiros decimais no intervalo [0..9].

1ª Questão – A Figura 1 representa o projeto da CPU MIPS monociclo conforme apresentado no Capítulo 4.4 do livro texto¹.

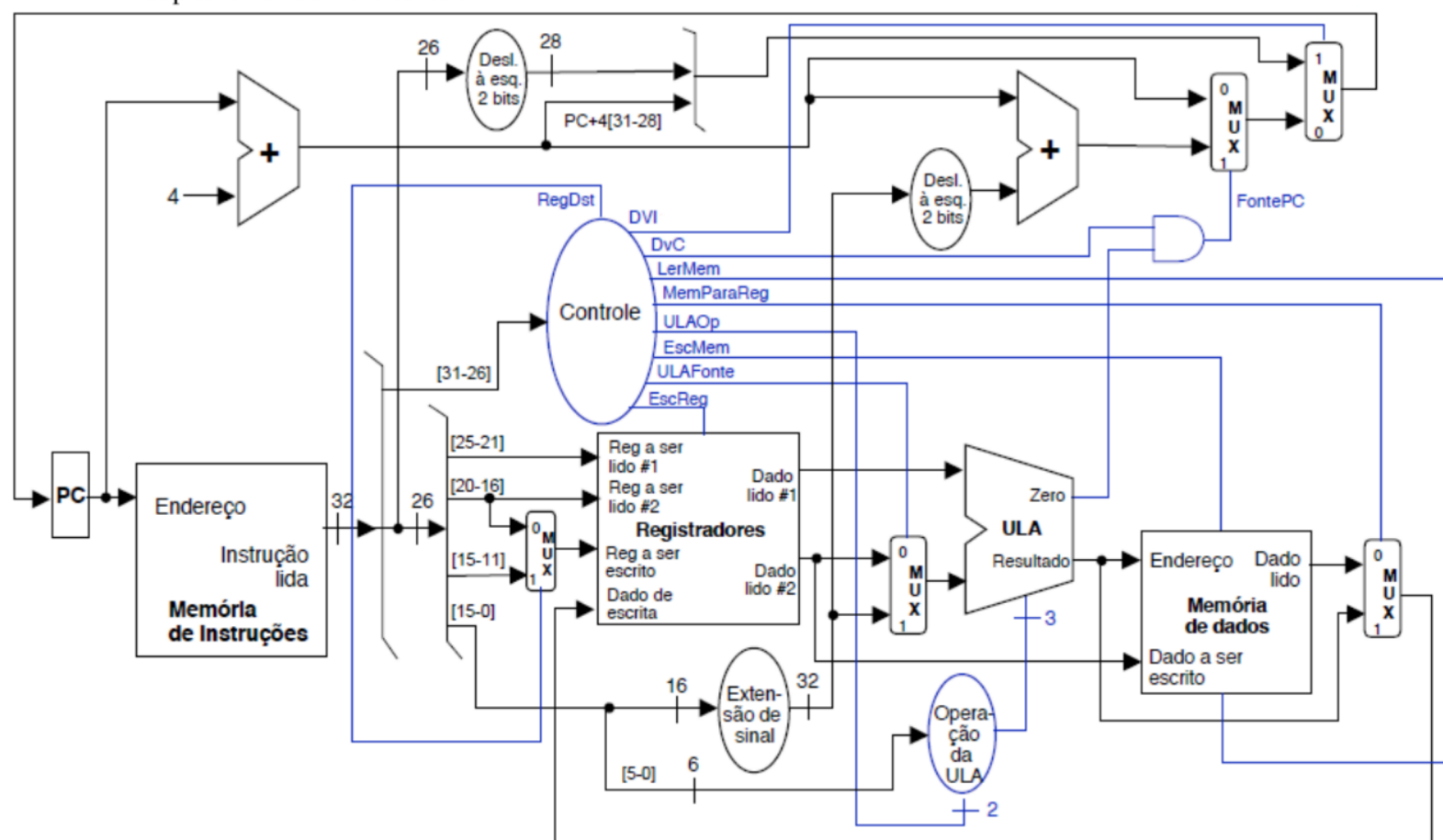


Figura 1 – Projeto do Caminho de Dados e Controle da CPU MIPS monociclo

A instrução *jal* é uma instrução de salto incondicional, similar ao *j*, porém com a característica de armazenar o endereço da próxima instrução (PC+4) no registrador *\$ra*. A instrução *jal* é bastante útil na chamada de funções e procedimentos que tenham retorno ao final da execução.

1 HENNESSY, J. L.; PATTERSON, D. A. Computer Organization and Desing: The hardware/software Interface. 5th Edition, 2014. ISBN: 978-0-12-407726-3

CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Em resumo, a operação da instrução *jal target* possui o seguinte efeito:

$\$ra = PC + 4$ //armazena o endereço da próxima instrução no registrador $\$ra$
 $PC = target$ //carrega *target** no registrador PC

* *target* está no mesmo formado usado pela instrução *j*, ver Figura 2 (Apêndice A.10 do livro texto¹).

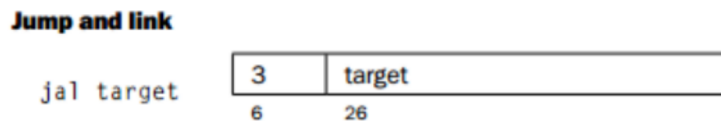


Figura 2 – Formato da Instrução *jal*, conforme linguagem de montagem MIPS

- a) (1,5) Identifique as modificações necessárias no caminho de dados (*muxes* e novos caminhos) do projeto apresentado na Figura 1 para adicionar suporte à instrução *jal*. Justifique. Se necessário use esquemas (desenhos) localizados em sua resposta.
- b) (1,0) Identifique as modificações de sinais de controle necessárias no projeto apresentado na Figura 1 para adicionar suporte à instrução *jal*. Justifique. Se necessário use esquemas (desenhos) localizados em sua resposta.

2ª Questão – Considere o programa *p2q2.asm* a seguir e substitua “X+Y” pelo valor da soma X+Y conforme seu número de matrícula na UFES. Se em seu caso específico X e Y forem iguais a 0, escolha aleatoriamente um valor para X+Y no intervalo [1..18].

p2q2.asm:

```
.data
numbers:    .word  1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35
count:      .word  "X+Y"
res:        .word  0

.text
la $a0, numbers // Linha 0
lw $a1, count   // Linha 1
li $t2, 0       // Linha 2
sum:
beq $t2, $a1, finish // Linha 3
addi $a0, $a0, 4   // Linha 4
addi $t2, $t2, 1   // Linha 5
lw $t0, 0($a0)     // Linha 6
add $t1, $t1, $t0  // Linha 7
b sum             // Linha 8
finish:
sw $t1, res       // Linha 9
```

CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

- a) (1,5) Apresente o estado das linhas de dados (caminhos dos *muxes*) e sinais de controle para a execução da instrução na Linha X do programa *p2q2.asm*. Justifique sua resposta.
- b) (1,0) Preencha a tabela a seguir conforme o valor de $X+Y$ no seu caso específico. Justifique o valor de *res* obtido no seu caso específico (o que faz o programa?).

Valor de <i>res</i>	# Instruções Executadas do Tipo R	# Instruções Executadas do Tipo I	# Instruções Executadas do Tipo J	# Total de Instruções Executadas

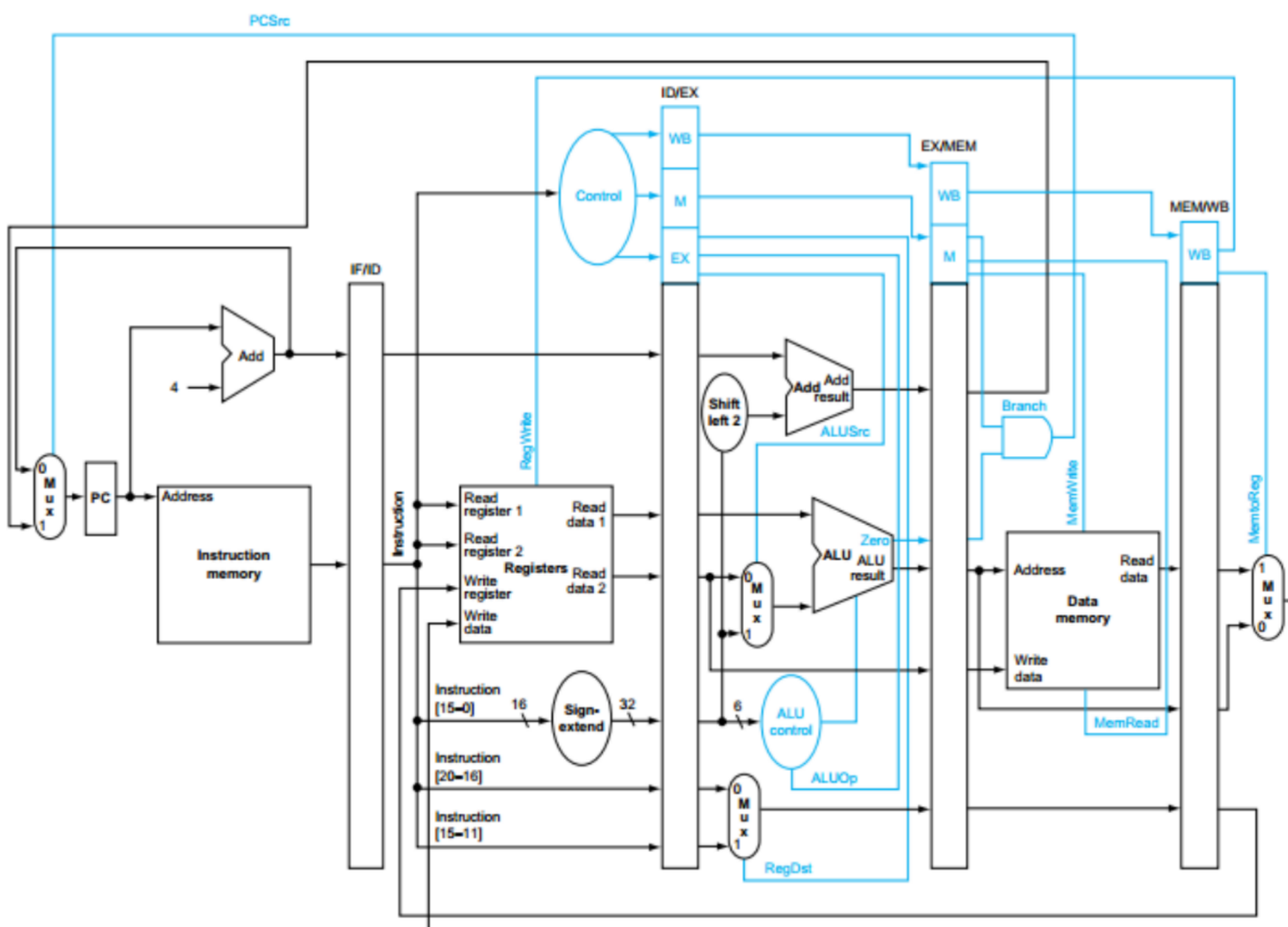


Figura 3 – Versão simplificada da CPU MIPS com *pipeline* de 5 estágios