C-208 – Arquitetura de Computadores

Capítulo 1 – Introdução à Arquitetura de Computadores

(parte 3)

Prof. Yvo Marcelo Chiaradia Masselli

Arquitetura MIPS

Componentes

Caminho de Dados e Controle

Arquitetura MIPS – Componentes do Hardware

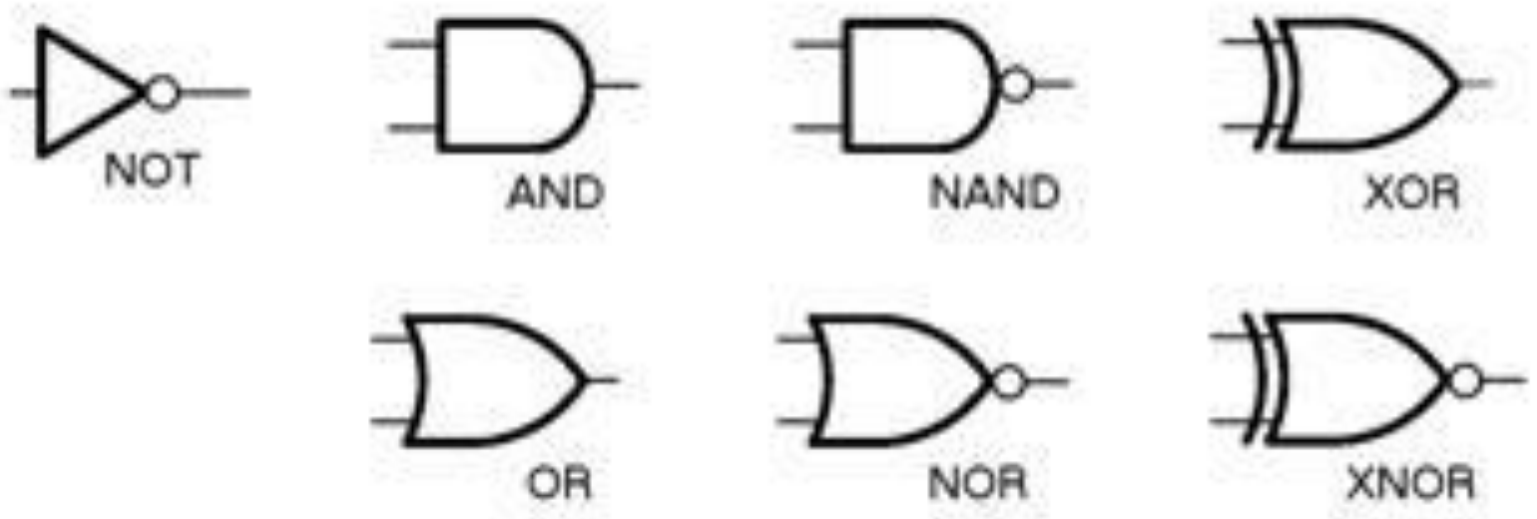
O Hardware do processador MIPS é composto de:

- Banco de Registradores
- Memória de Dados
- Memória de Programa
- Unidade de Controle
- ALU (Unidade Lógica e Aritmética)
- Circuitos auxiliares (portas lógicas, multiplexadores, somadores, etc)

Arquitetura MIPS – Componentes do Hardware

Circuitos auxiliares:

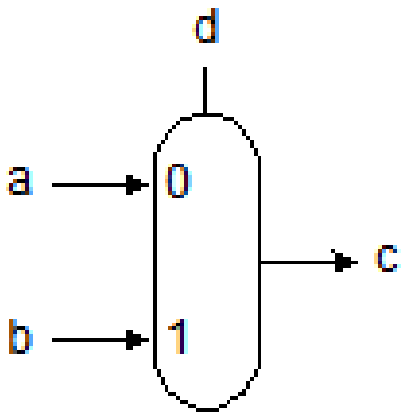
Portas lógicas



Arquitetura MIPS – Componentes do Hardware

Circuitos auxiliares:

Multiplexadores:



Mux de 2 canais

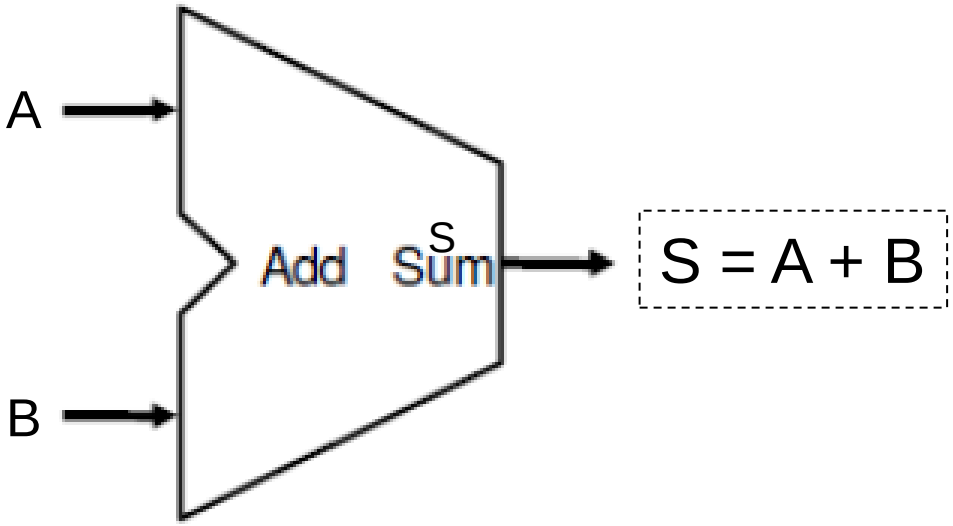
d	c
0	a
1	b

```
if  (d==0) c = a;
else c = b;
```

Arquitetura MIPS – Componentes do Hardware

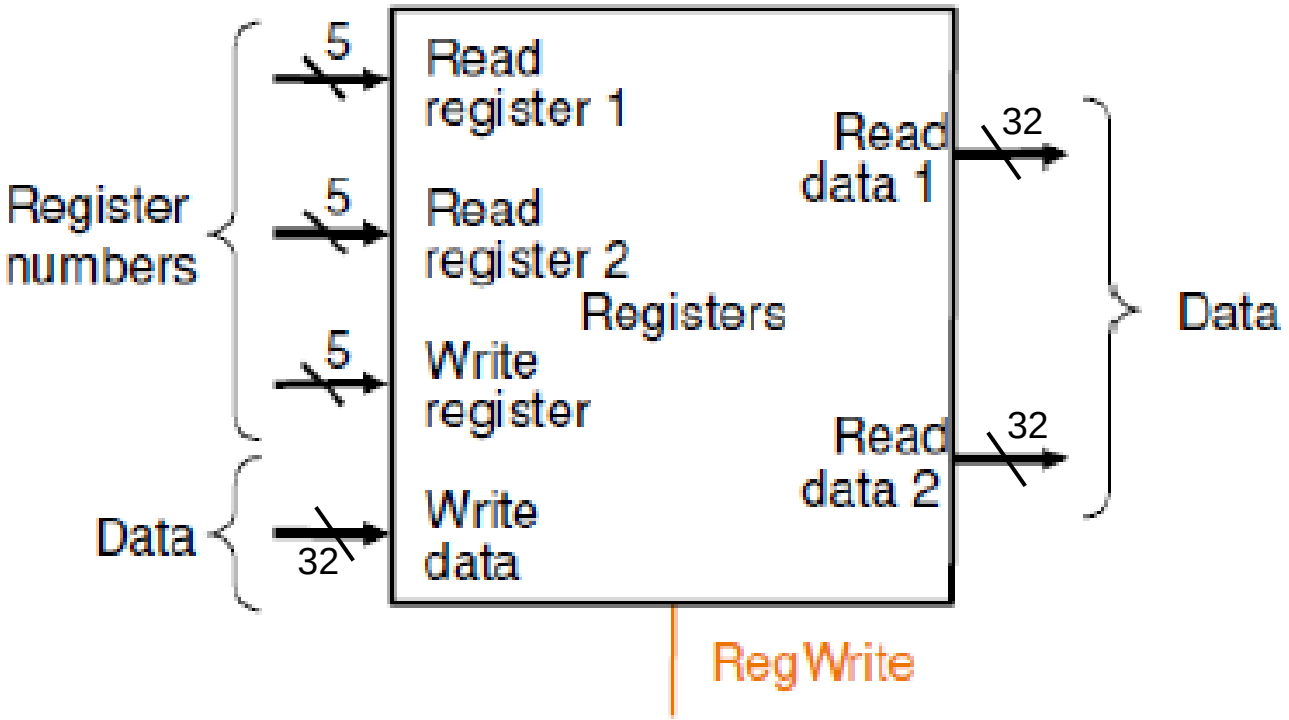
Circuitos auxiliares:

Somadores:



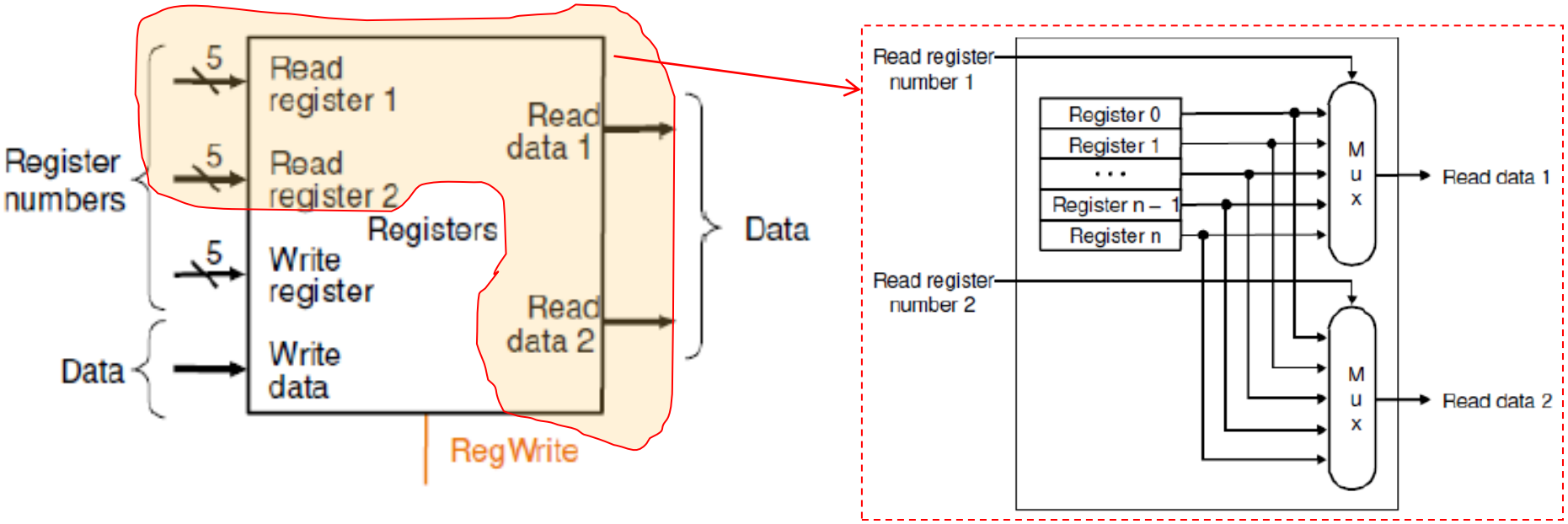
Arquitetura MIPS – Componentes do Hardware

Banco de Registradores:



Arquitetura MIPS – Componentes do Hardware

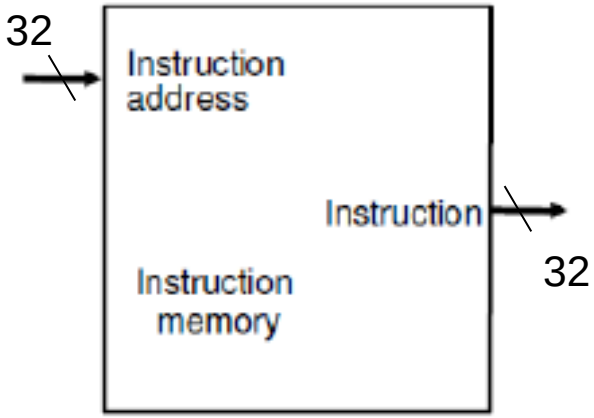
Banco de Registradores:



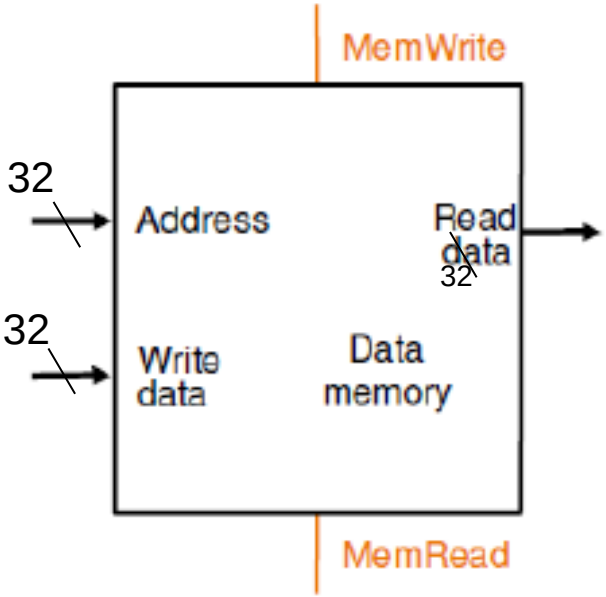
Arquitetura MIPS – Componentes do Hardware

Memórias:

Programa (instruções)

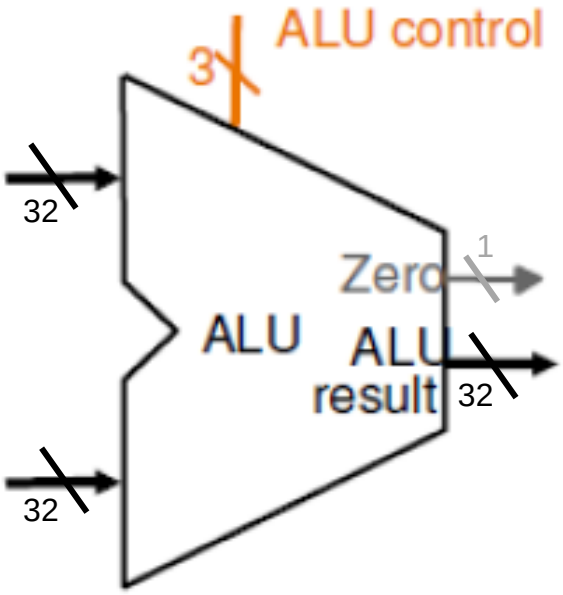


Dados



Arquitetura MIPS – Componentes do Hardware

Unidade Lógica e Aritmética (ULA):



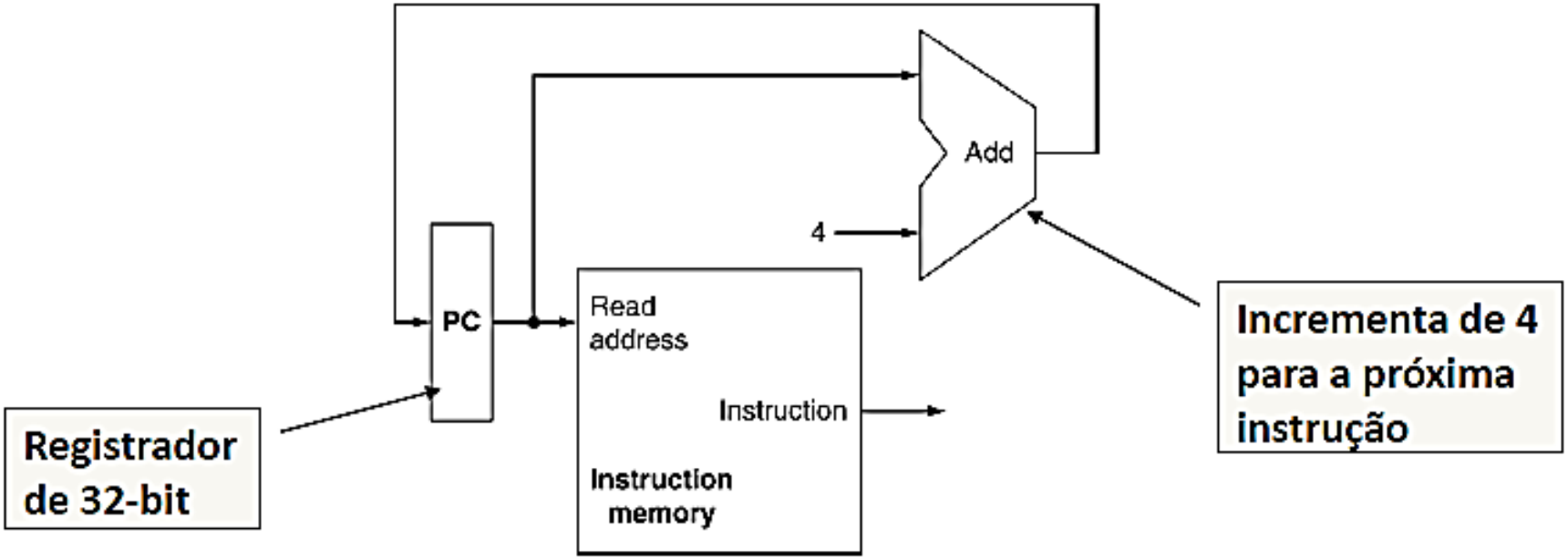
Arquitetura MIPS – Caminho de Dados e de Controle

Passos de execução das instruções:

- Buscar uma instrução na memória
- Interpretar a instrução
- Atualizar o registrador PC
- Trazer (se for o caso) os operandos para a CPU
- Executar a operação
- Armazenar (se for o caso) os dados de saída
- Repetir o processo com uma nova instrução

Arquitetura MIPS – Caminho de Dados e de Controle

Busca da Instrução:



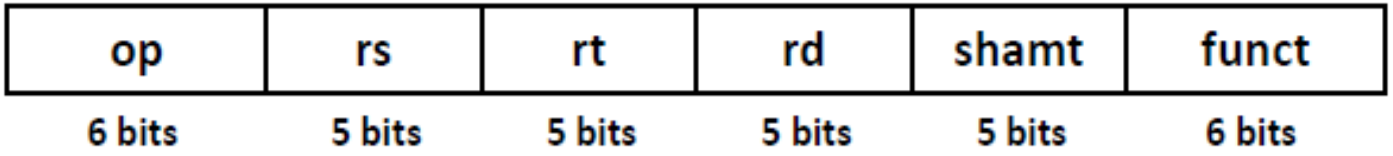
PC → é atualizado ao final de cada ciclo de clock para PC ← PC + 4

O valor 4 numa das entradas do somador indica que a próxima instrução está armazenada 4 bytes (32 bits) depois do valor corrente do PC.

Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo R

- Leitura de dois registradores
- Realização da operação aritmética/lógica
- Escrita do resultado no registrador
- Uso dos componentes:
  - Banco de registradores
  - ALU

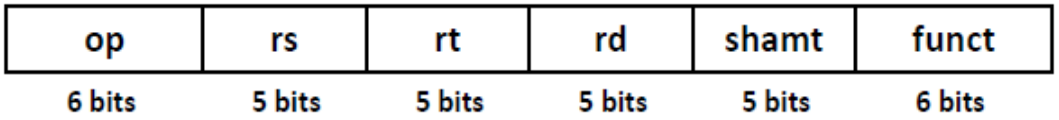


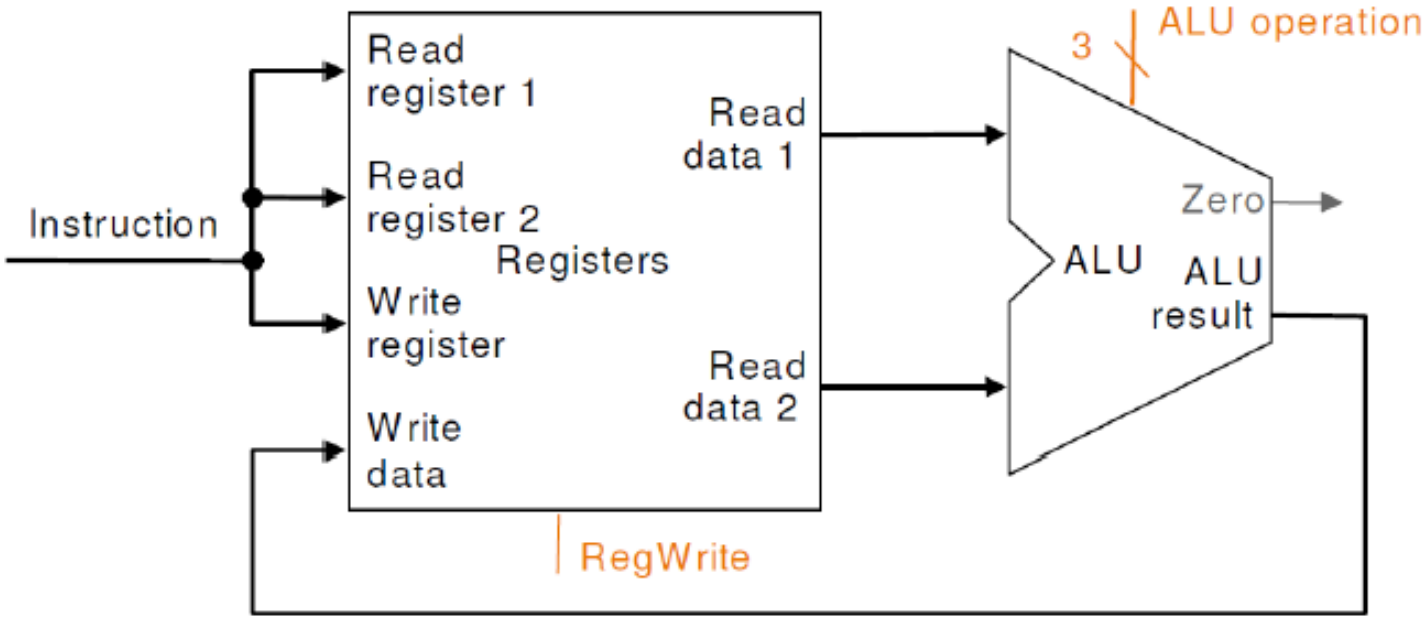
Exemplos: add $t0, $t1, $t2
sub $s0, $s1, $s2

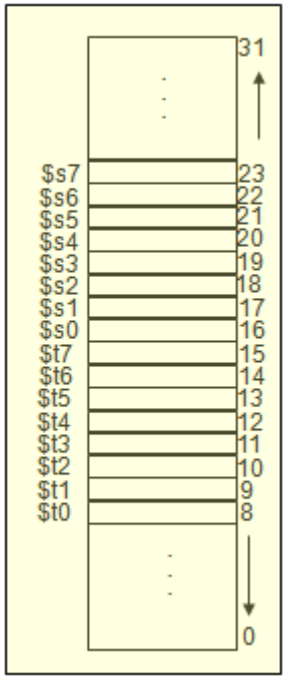
Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo R

- Exemplo: **add $t0, $t1, $t2**



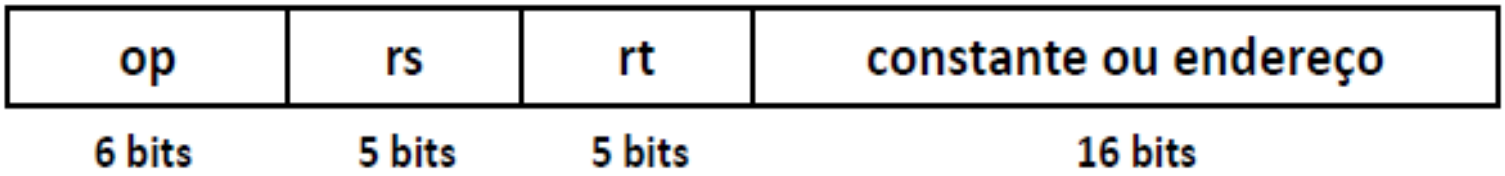




Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo I (Load/Store)

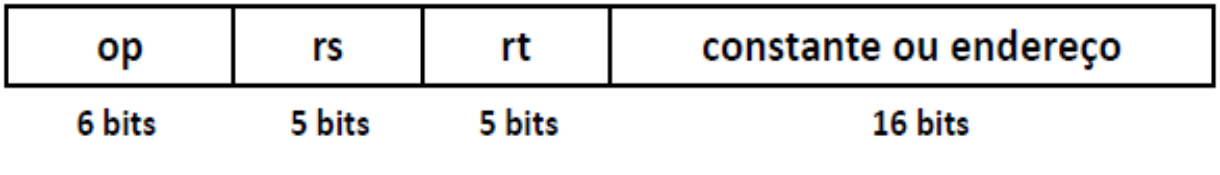
- Instruções com valor imediato de 16 bits
- Leitura dos operandos a partir dos registradores
- Calcula o endereço usando um deslocamento de 16 bits
  - –Uso da ALU, com o componente de extensão de sinal
- Load: lê a memória e atualiza o registrador
- Store: escreve o valor do registrador na memória

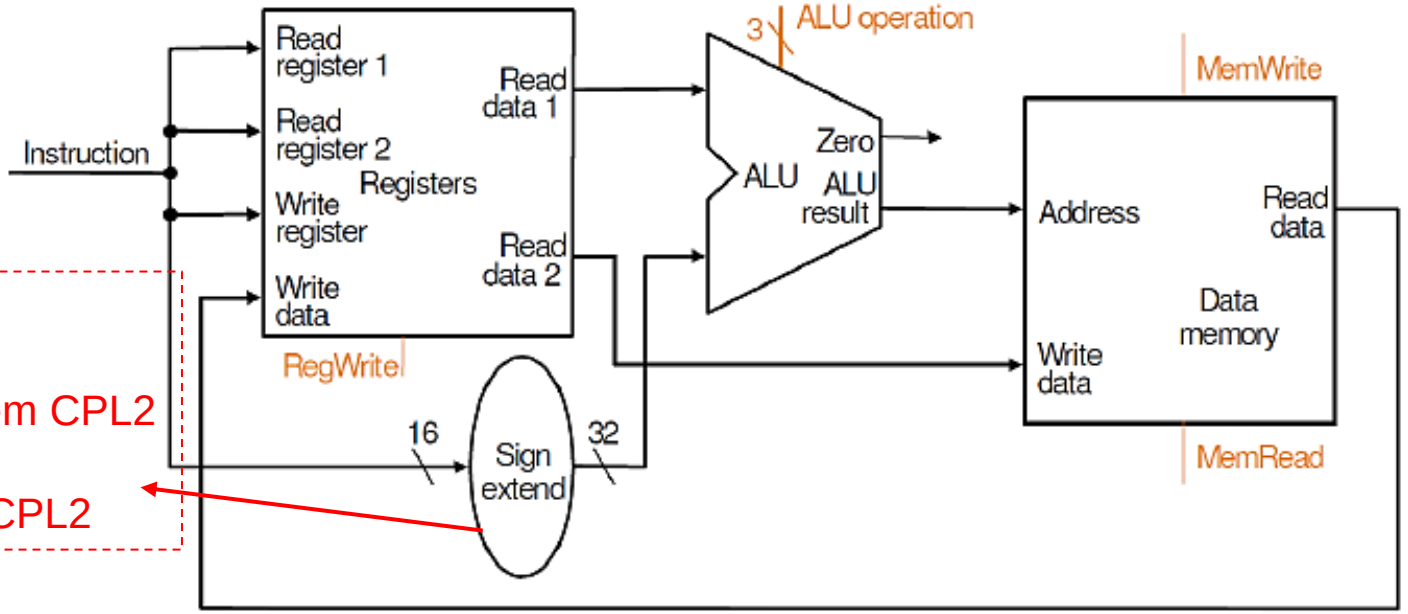


Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo I (Load/Store)

- Exemplo: **lw $t0, 4($t5)**





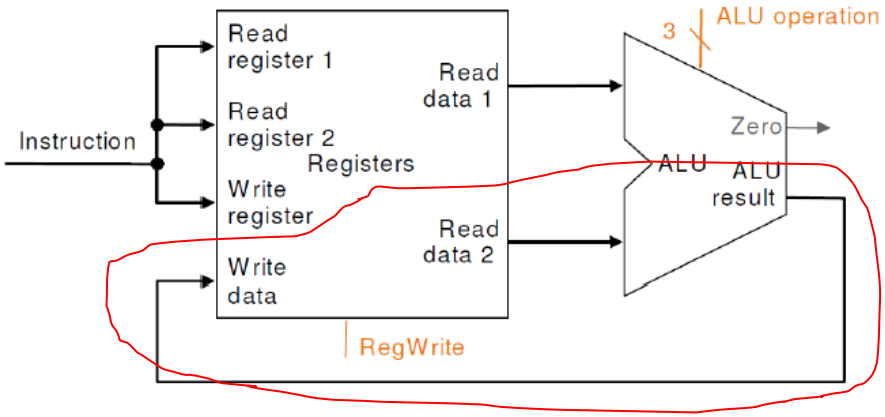
Sign extend:

Valor imediato: 16 bits em CPL2

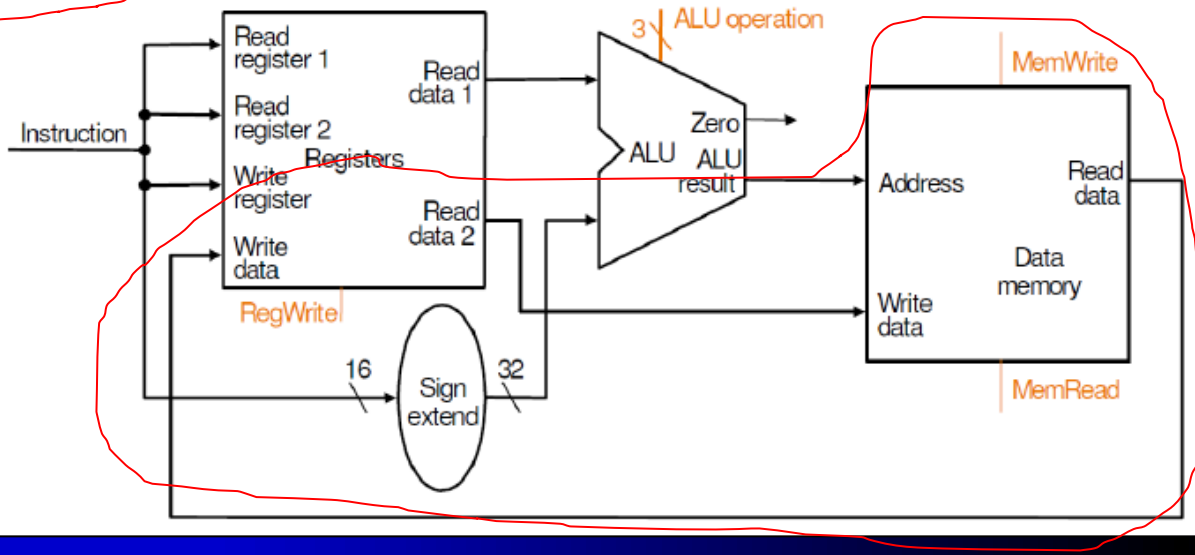
Transforma em 32 bits CPL2

Arquitetura MIPS – Caminho de Dados e de Controle

Instruções: Tipo R e I → conflito no Hardware !!!

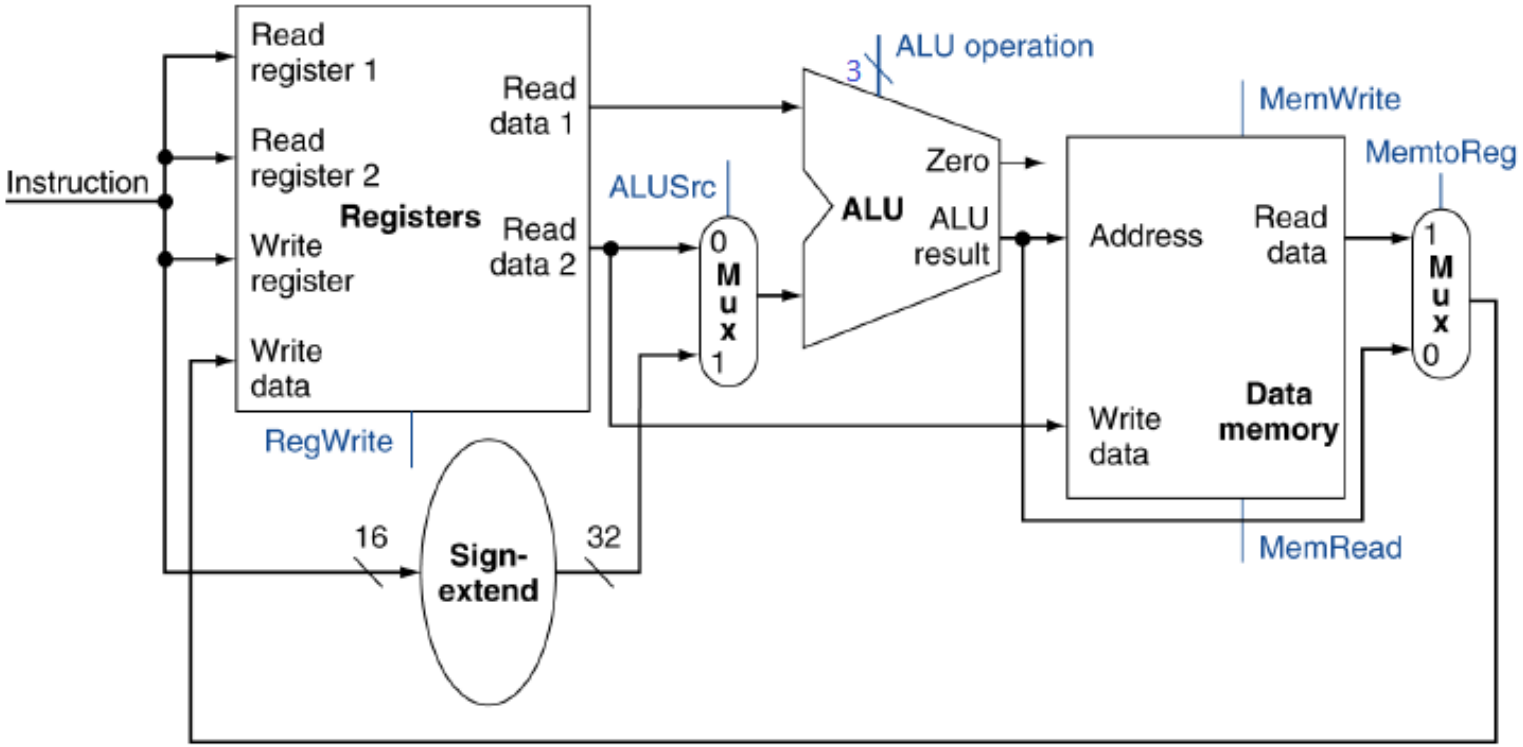


COMO RESOLVER?



Arquitetura MIPS – Caminho de Dados e de Controle

Instruções: Tipo R e I



Tipo-R: ALUSrc = 0 e MemtoReg = 0

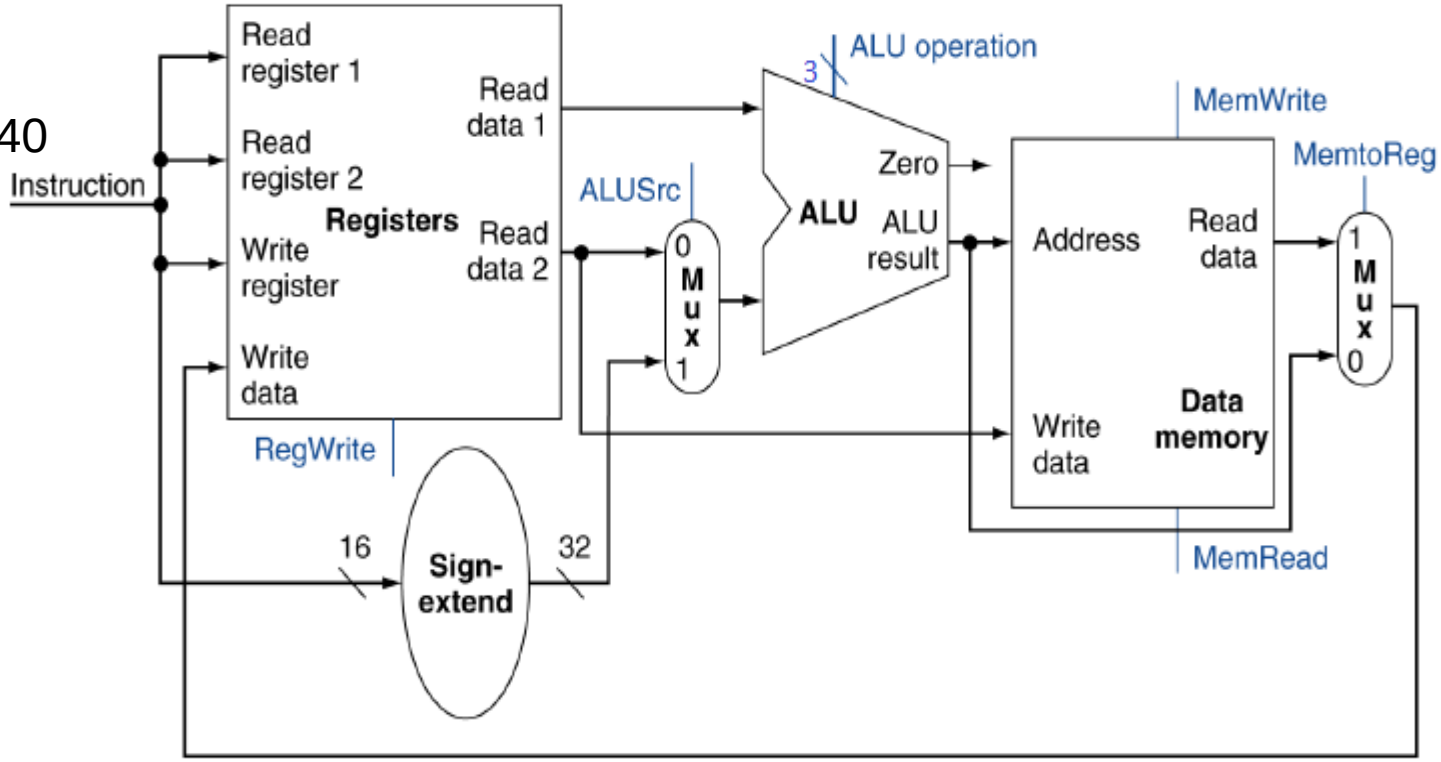
Tipo-I : (lw e sw) ALUSrc = 1 e MemtoReg = 1

Arquitetura MIPS – Caminho de Dados e de Controle

Exercícios:

Mostre o caminho de dados para as instruções a seguir, indicando os valores de todos os bits necessários para a execução das instruções.

a) add $t5, $t4, $t3
b) addi $t0, $s0, 0x40
c) sw $t0, 12($t1)
d) lw $t5, 16($t0)

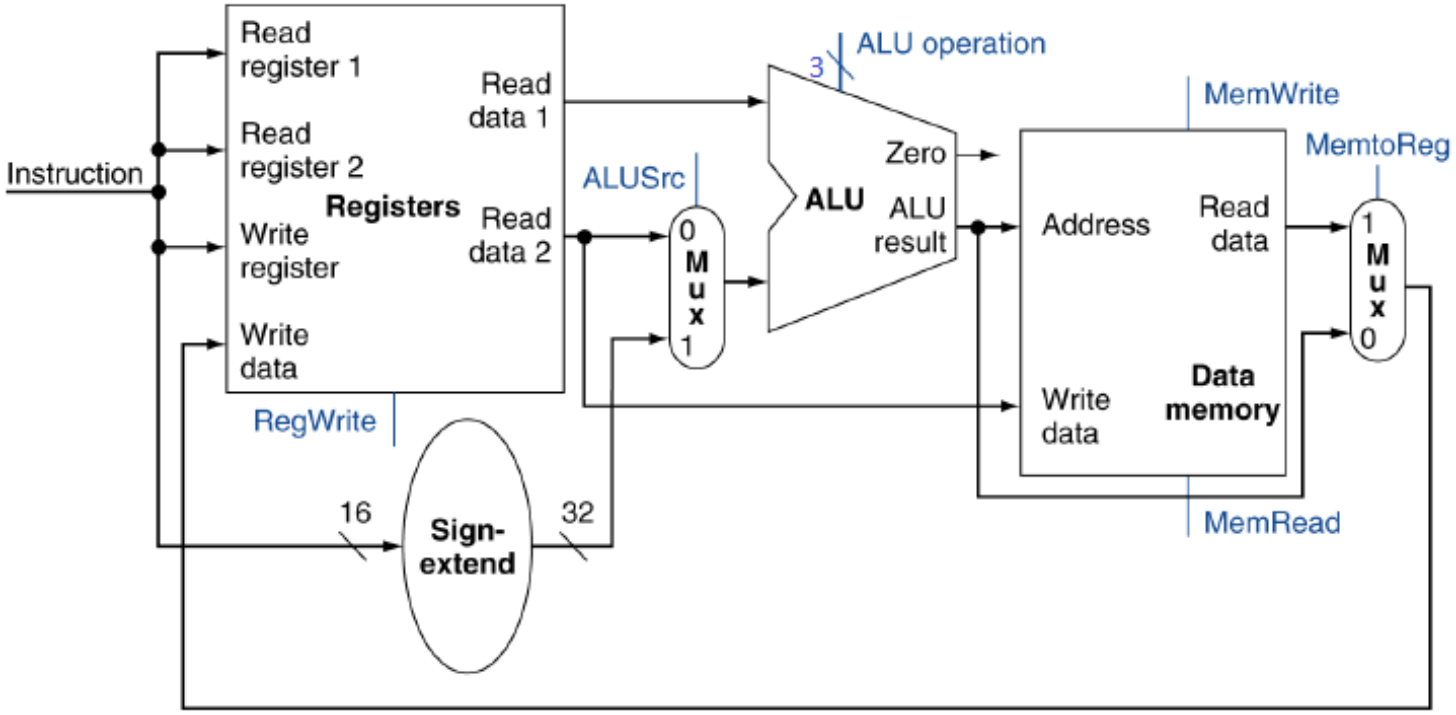


Arquitetura MIPS – Caminho de Dados e de Controle

Exercícios:

Complete a tabela:

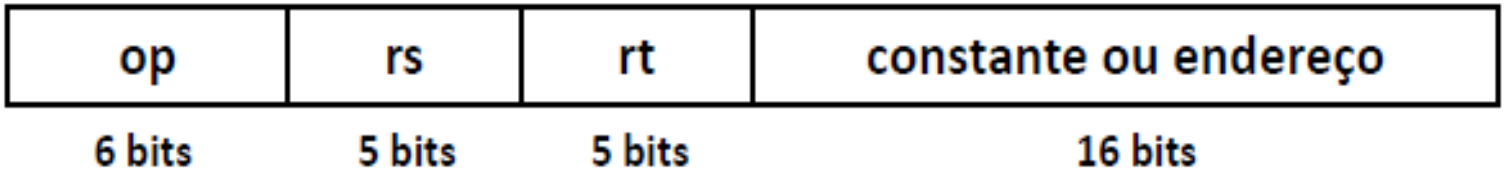
Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write
R-format						
`lw`						
`sw`						
`beq`						



Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo I (Desvio condicional (Branch): beq)

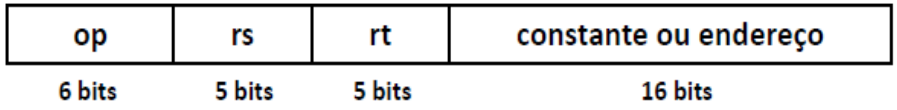
- Leitura dos operandos a partir dos registradores
- Comparação dos dois através da ALU
  - Subtração dos dois e caso sejam iguais, a saída Zero é setada
- Cálculo do endereço destino
  - Se houver desvio: PC + 4 + deslocamento

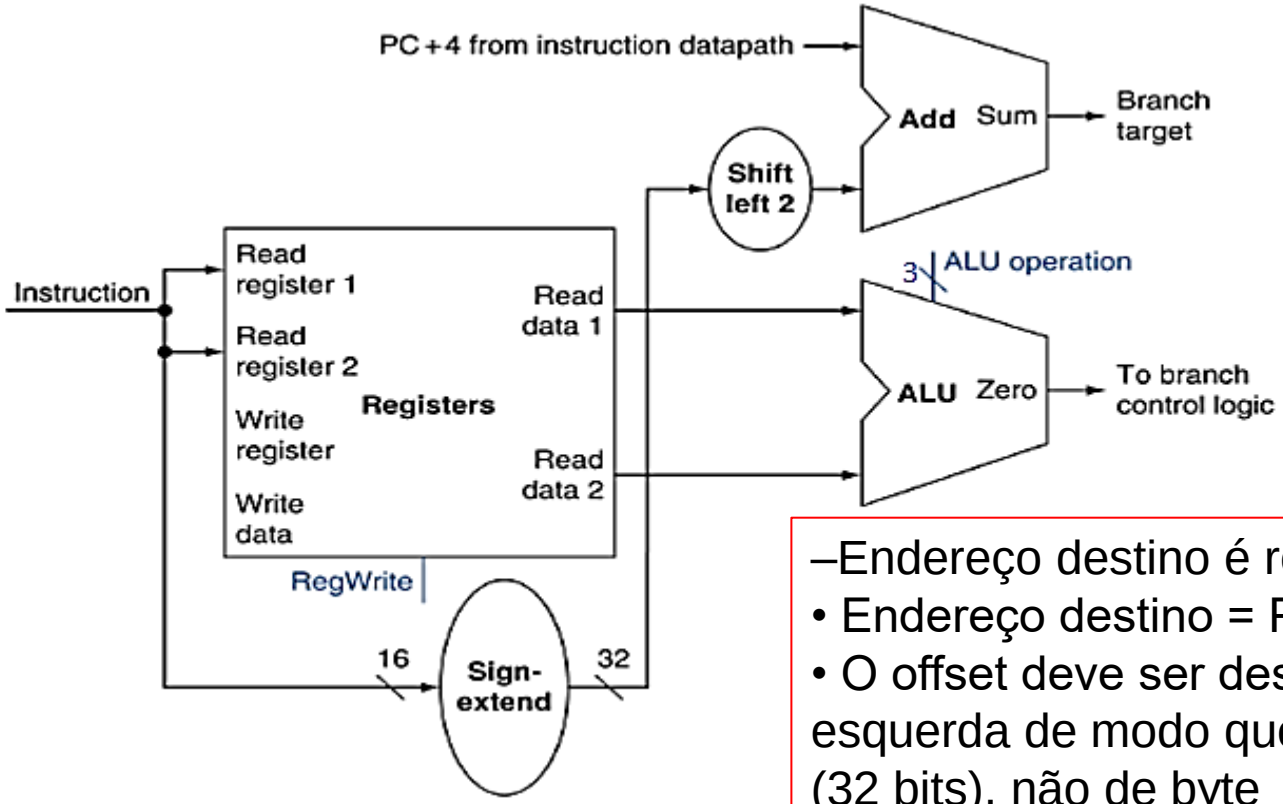


Arquitetura MIPS – Caminho de Dados e de Controle

Execução das instruções: Tipo I (Branch)

- Exemplo: **beq $t0, $t1, LABEL**

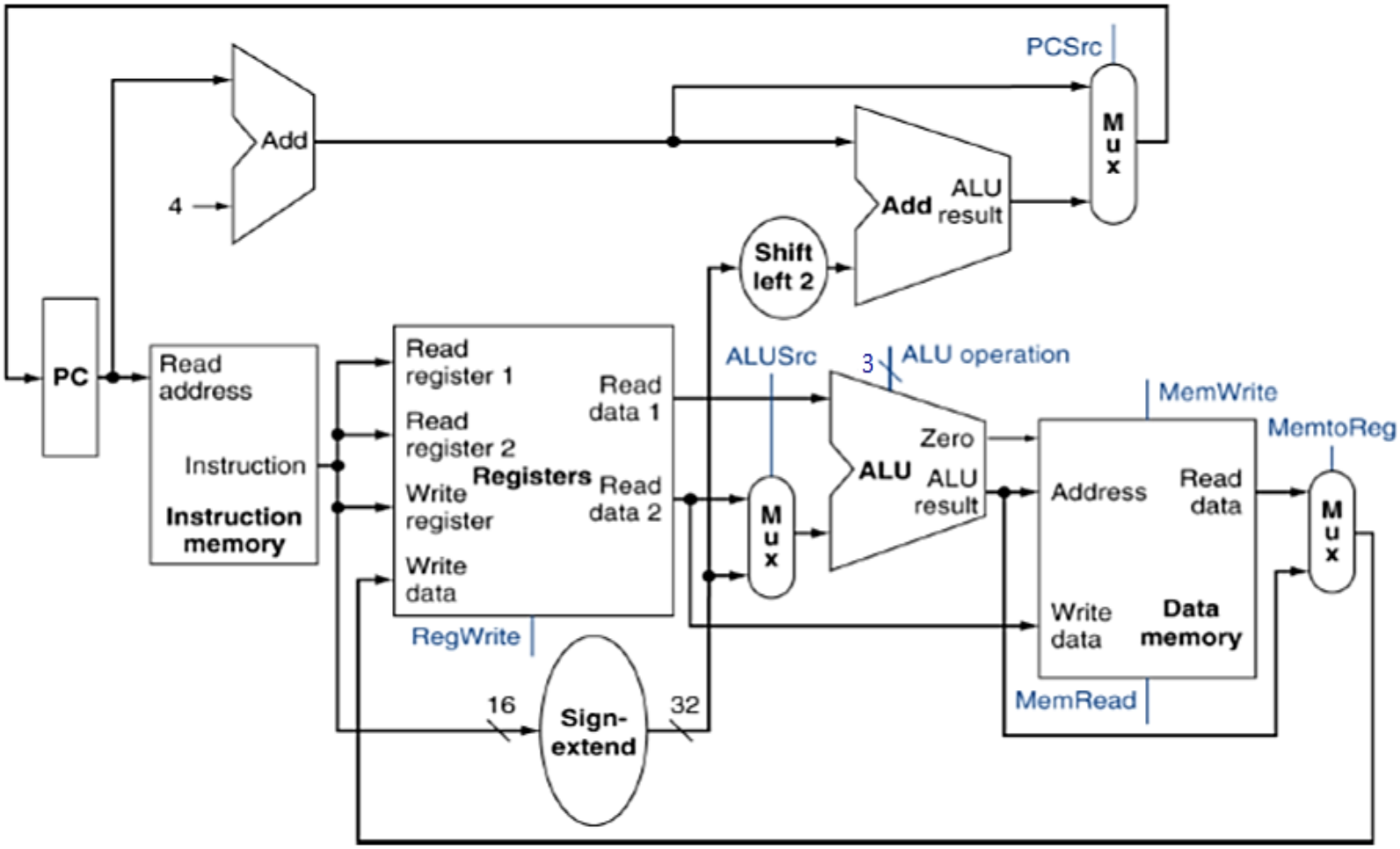




–Endereço destino é relativo a próxima instrução
• Endereço destino = PC + 4 + offset
• O offset deve ser deslocado de 2 bits para a esquerda de modo que seja um offset de palavra (32 bits), não de byte

Arquitetura MIPS – Caminho de Dados e de Controle

O caminho ... quase completo.

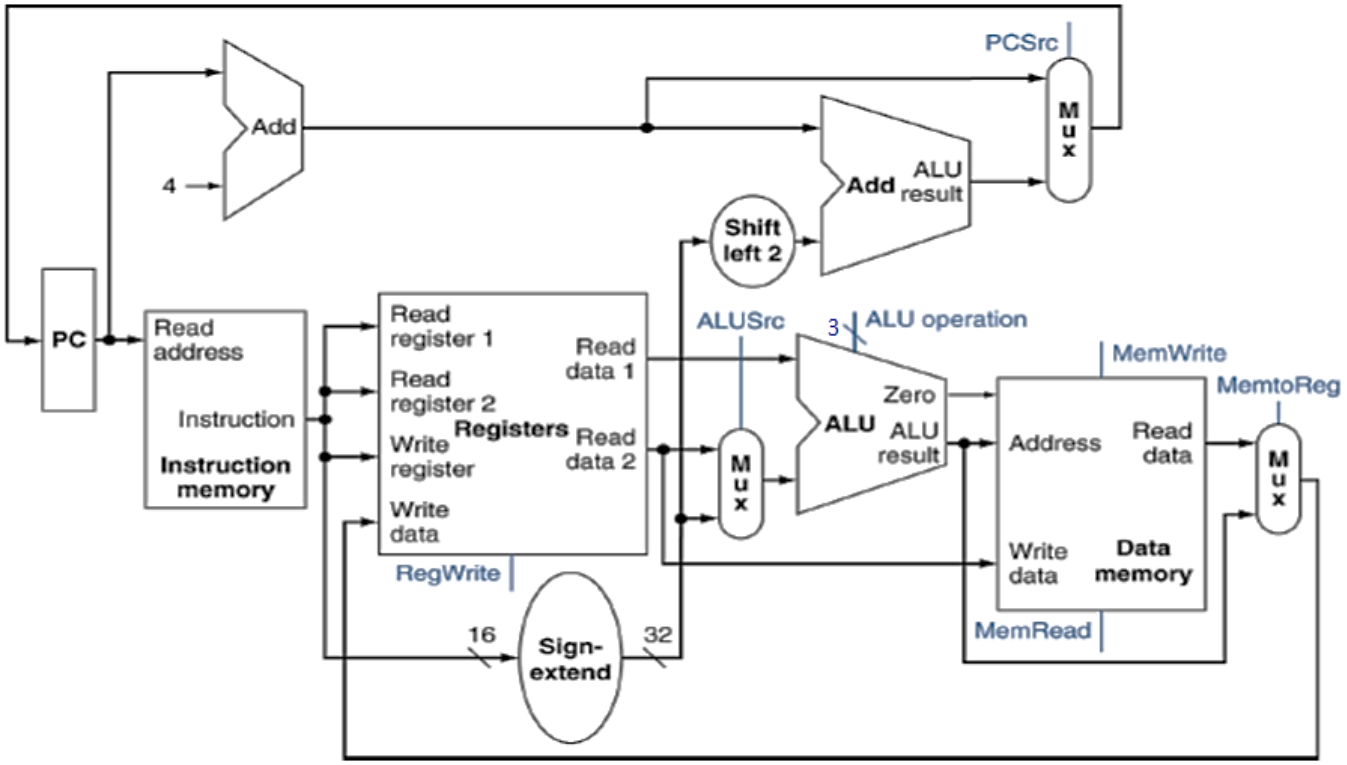


Arquitetura MIPS – Caminho de Dados e de Controle

Exercícios:

Mostre o caminho de dados para as instruções a seguir, indicando os valor de todos os bits necessários para a execução das instruções.

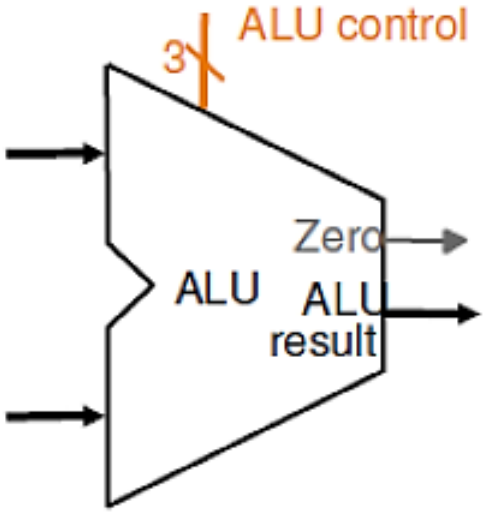
a) add $S2, $S3, $S4
b) addi $t3, $t0, 0x55
c) sw $t3, 8($s1)
d) beq $t0, $t1, 0x24



Arquitetura MIPS – Caminho de Dados e de Controle

Controle da ALU

A ALU possui três entradas e são usadas somente cinco das oito possíveis combinações de acordo com o campo de “Função” em algumas instruções (6 bits).



Entrada da ALU	Função
000	AND
001	OR
010	soma
110	subtração
111	set less than

	31-26	25-21	20-16	15-11	10-6	5-0
Tipo R	0	rs	rt	rd	shmat	func

Arquitetura MIPS – Caminho de Dados e de Controle

Controle da ALU

A saída da unidade de controle para a ALU é um sinal de 2 *bits (ALUop)* que controla diretamente a ALU, gerando uma das cinco combinações necessárias.

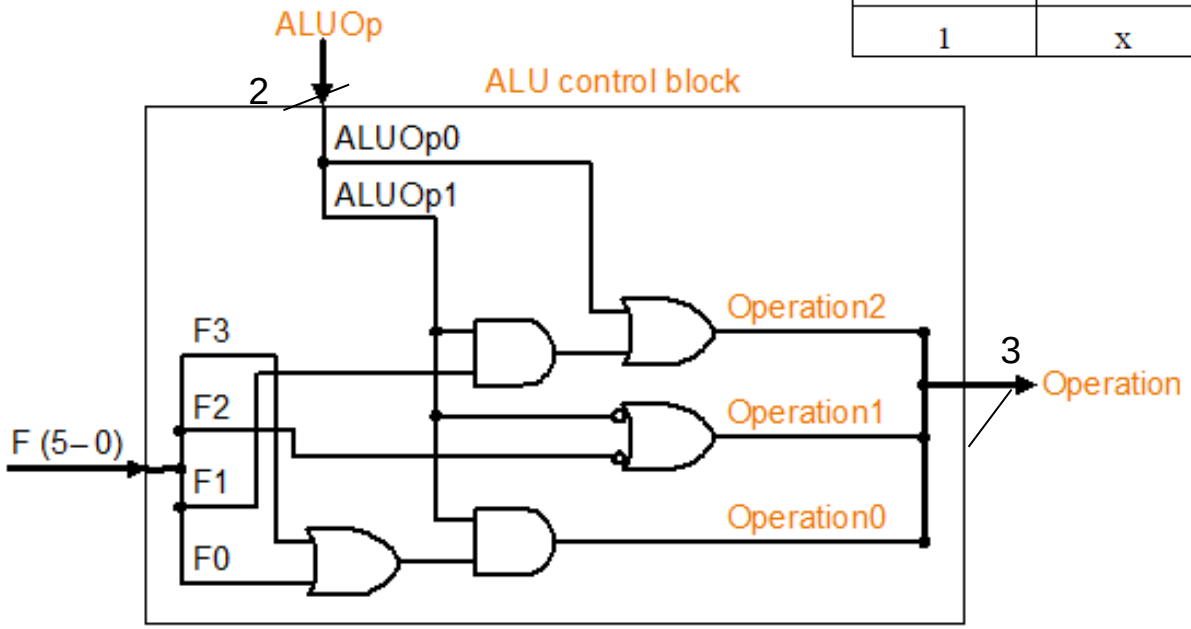
ALUOP		Func						Operação da ALU	
Bit 1	Bit 0	F5	F4	F3	F2	F1	F0		
0	0	x	x	x	x	x	x	010	**Soma**
x	1	x	x	x	x	x	x	110	**Sub**
1	x	x	x	0	0	0	0	010	**Soma**
1	x	x	x	0	0	1	0	110	**Sub**
1	x	x	x	0	1	0	0	000	**And**
1	x	x	x	0	1	0	1	001	**Or**
1	x	x	x	1	0	1	0	111	**Slt**

O campo ALUOp indica se a operação é de soma (00), para *lw e sw, subtração (01), para beq, ou se deve ser determinada pelo campo func (10), para* instruções lógicas e aritméticas.

Arquitetura MIPS – Caminho de Dados e de Controle

Controle da ALU

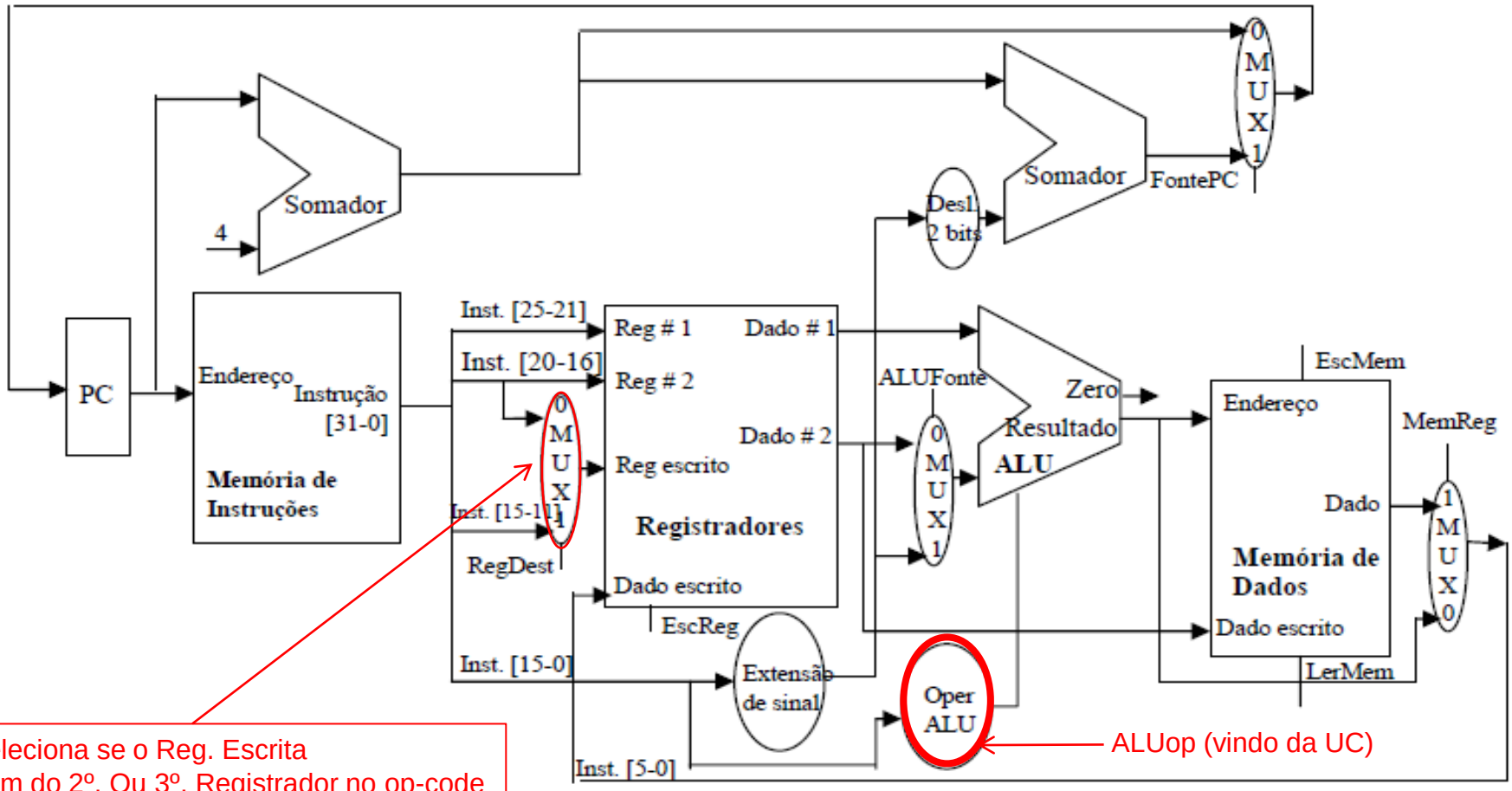
ALUOP		Func						Operação da ALU
Bit 1	Bit 0	F5	F4	F3	F2	F1	F0	
0	0	x	x	x	x	x	x	010
x	1	x	x	x	x	x	x	110
1	x	x	x	0	0	0	0	010
1	x	x	x	0	0	1	0	110
1	x	x	x	0	1	0	0	000
1	x	x	x	0	1	0	1	001
1	x	x	x	1	0	1	0	111



Arquitetura MIPS – Caminho de Dados e de Controle

Controle da ALU

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits



Arquitetura MIPS – Caminho de Dados e de Controle

Unidade de Controle

A UC deve gerar todos os sinais de controle para o circuito.

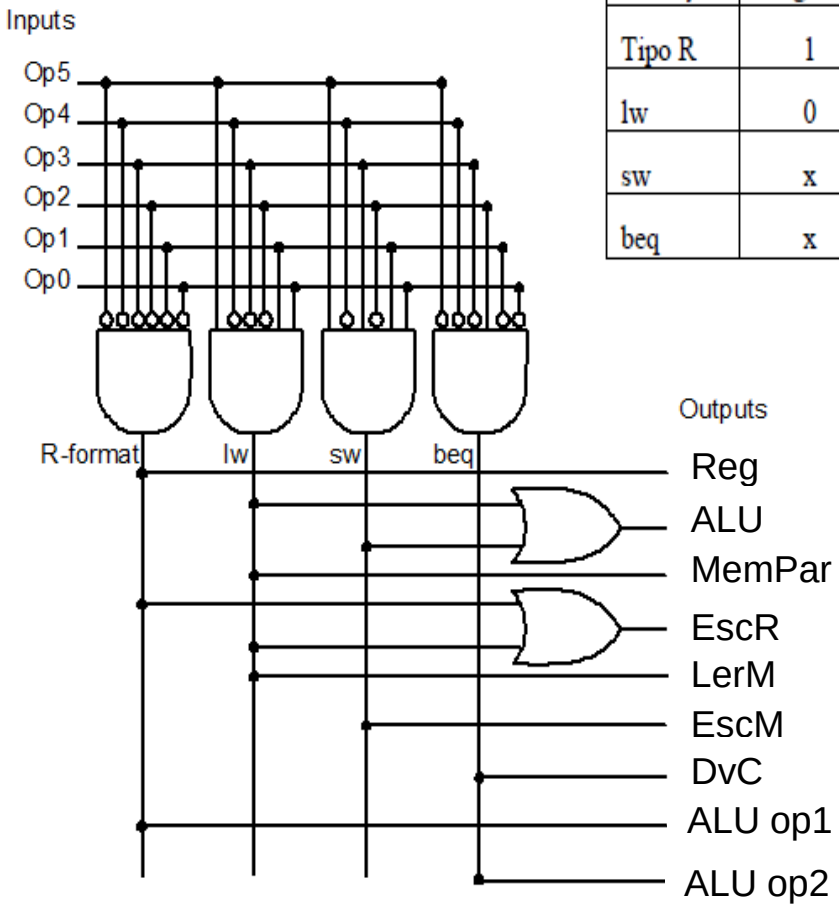
instrução	Opcode em decimal	Op5	Op4	Op3	Op2	Op1	Op0
Tipo R	0	0	0	0	0	0	0
lw	35	1	0	0	0	1	1
sw	43	1	0	1	0	1	1
beq	4	0	0	0	1	0	0

entradas						saídas								
Op5	Op4	Op3	Op2	Op1	Op0	RegDst	ULAFonte	MemParaReg	EscReg	LerMem	EscMem	DvC	ULAOp1	ULAOp0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
1	0	0	0	1	1	0	1	1	1	1	0	0	0	0
1	0	1	0	1	1	X	1	X	0	0	1	0	0	0
0	0	0	1	0	0	X	0	X	0	0	0	1	0	1

Arquitetura MIPS – Caminho de Dados e de Controle

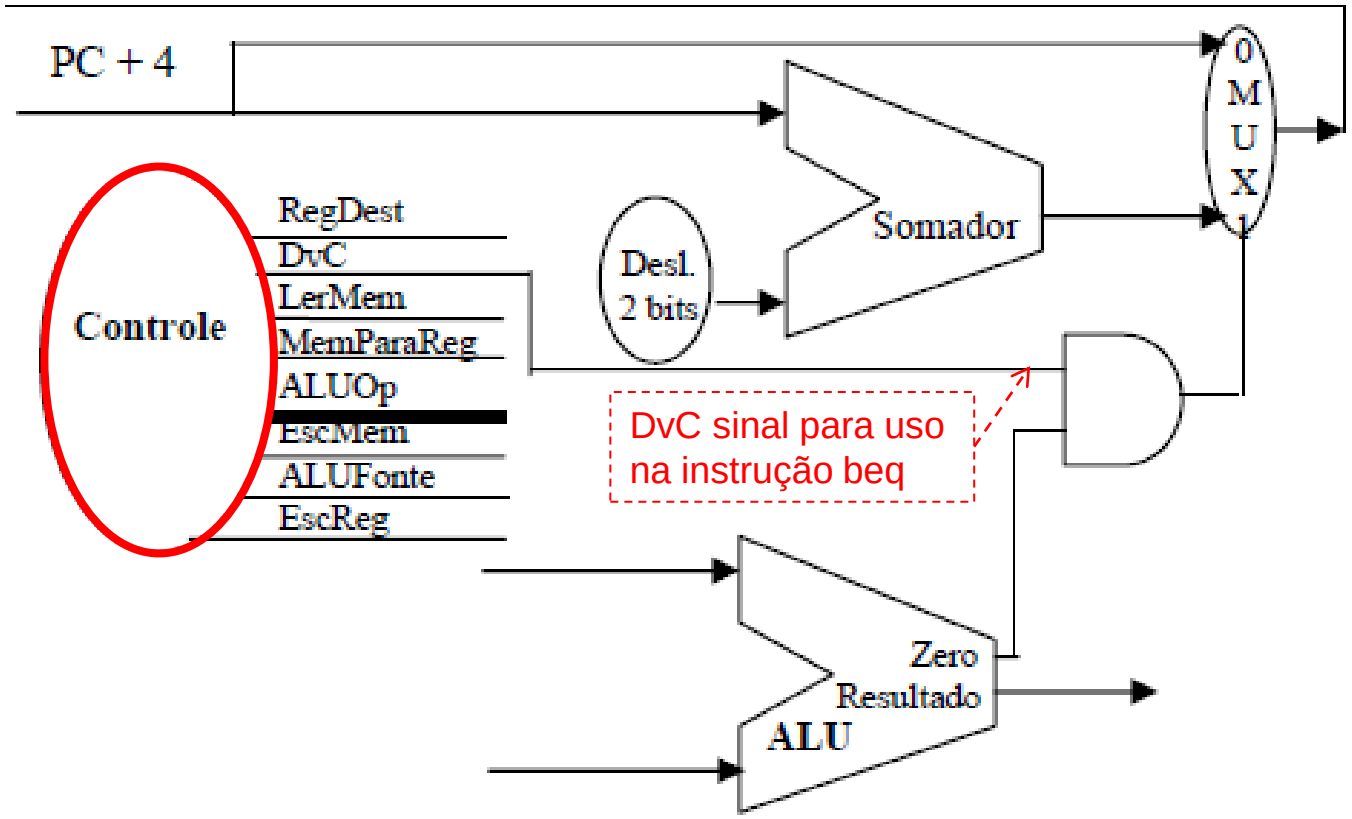
Unidade de Controle

Instrução	RegDest	ALUFonte	MemReg	EscReg	LerMem	Escmem	DvC	ALUOp	
Tipo R	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	x	1	x	0	0	1	0	0	0
beq	x	0	x	0	0	0	1	0	1



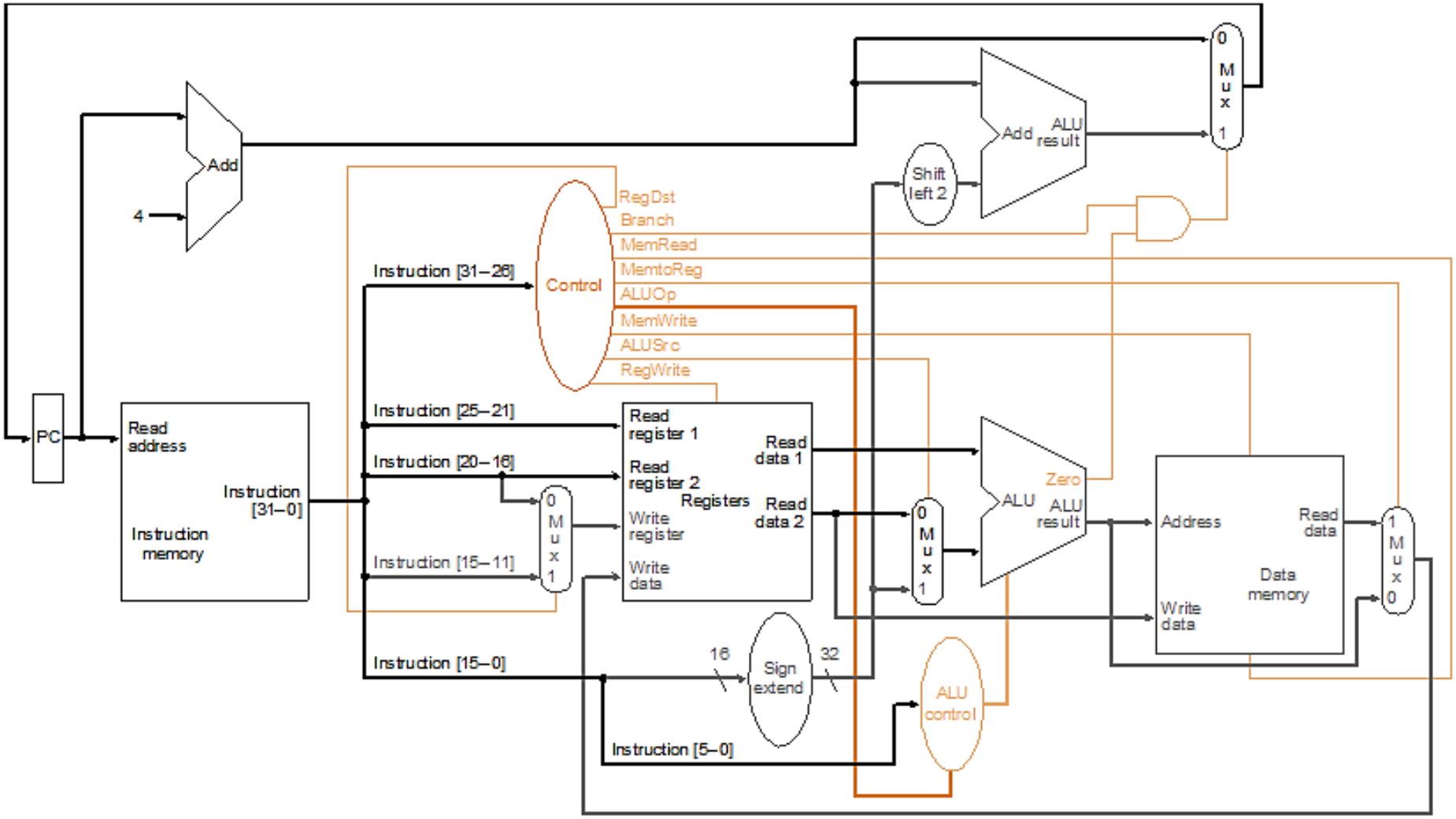
Arquitetura MIPS – Caminho de Dados e de Controle

Unidade de Controle



Arquitetura MIPS – Caminho de Dados e de Controle

Quase completo .... (ainda não executa a instrução de jump)



Arquitetura MIPS – Caminho de Dados e de Controle

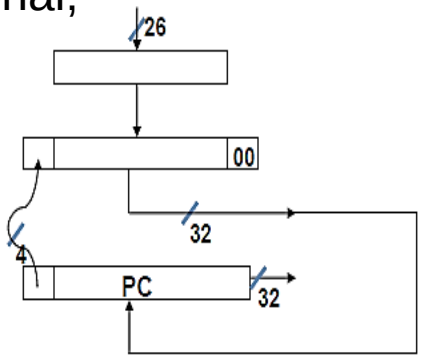
A instrução de jump

31-26	25-0
2	endereço

• Esta instrução é similar a de desvio condicional. Porém, produz o endereço alvo de desvio de forma diferente:
  • O campo de endereço (26 bits) é transformado para 28 bits (shift)
  • Os 2 *bits menos significativos são sempre iguais a zero (00).*
  • *Os* 26 *bits são fornecidos pela própria instrução.*
  • *Os 4 bits mais significativos (formando 32 bits) são* obtidos do PC corrente (PC + 4).

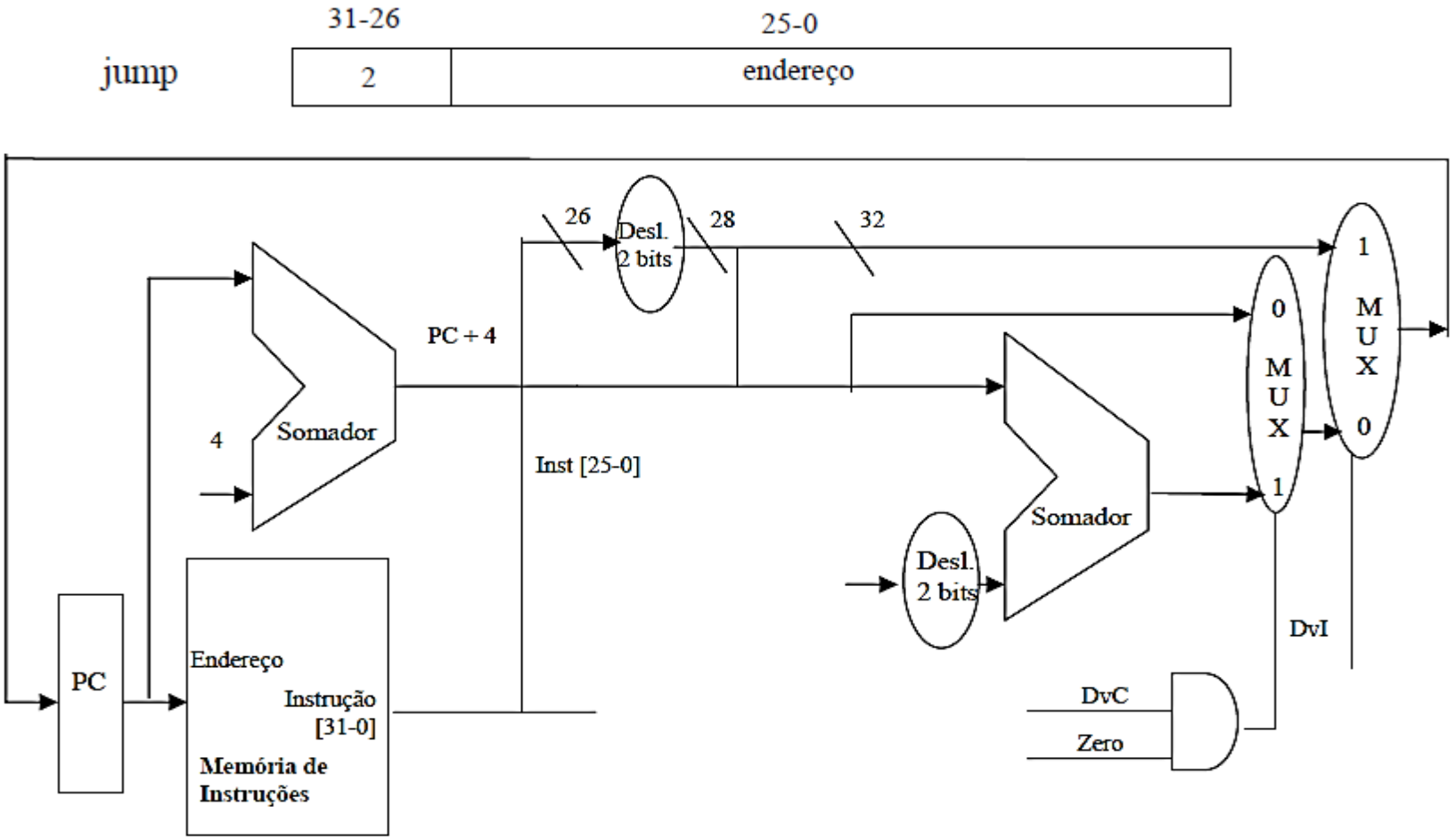
• Assim, o endereço alvo de desvio é resultado da combinação de:
  • 4 *bits mais significativos do valor PC + 4, que são os bits de 31 a 28;*
  • *26 bits* do campo imediato da instrução de desvio incondicional;
  • e 2 *bits menos significativos em 0.*

•*O sinal da unidade de* controle **DvI ativa o multiplexador que carrega o PC.**



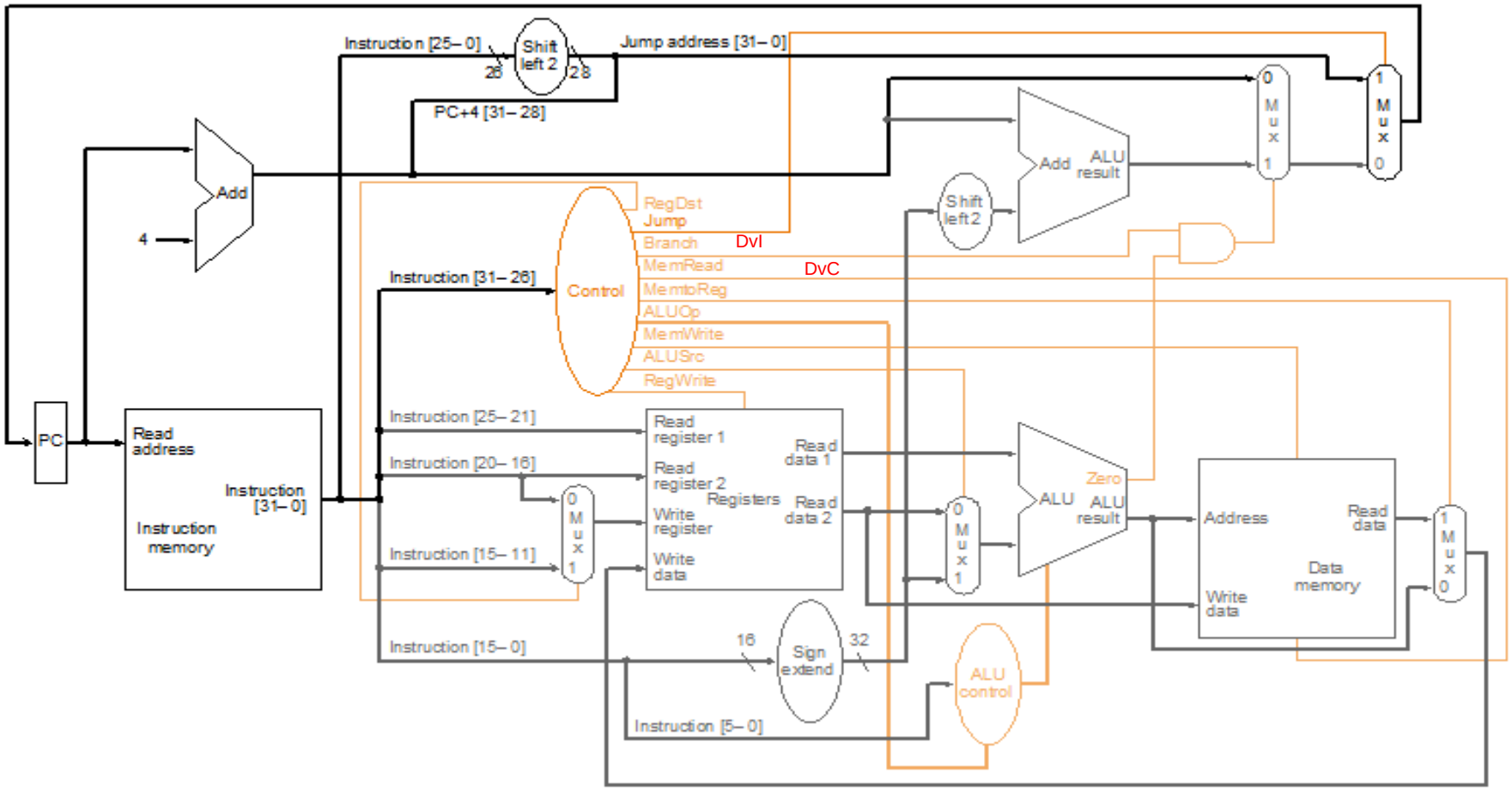
Arquitetura MIPS – Caminho de Dados e de Controle

A instrução de jump



Arquitetura MIPS – Caminho de Dados e de Controle

Caminho completo. Executa instruções Tipo-R, Tipo-I e Tipo J



Arquitetura MIPS – Caminho de Dados e de Controle

Exercícios: Mostre o caminho de dados e de controle (com o valor de todos os sinais), para as instruções a seguir:

a) add $t1, $t0, $t2

b) and $s0, $s1, $s0

c) addi $t3, $t0, 5

d) addi $t0, $zero, -1

e) lw $t0, 24($s0)

f) sw $t1, 4($s1)

g) beq $s0,$s1,100

h) j 350