



Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Ciências da Computação & Engenharia Eletrônica



Sistemas Digitais

INE 5406

Aula 8-T : Parte 1

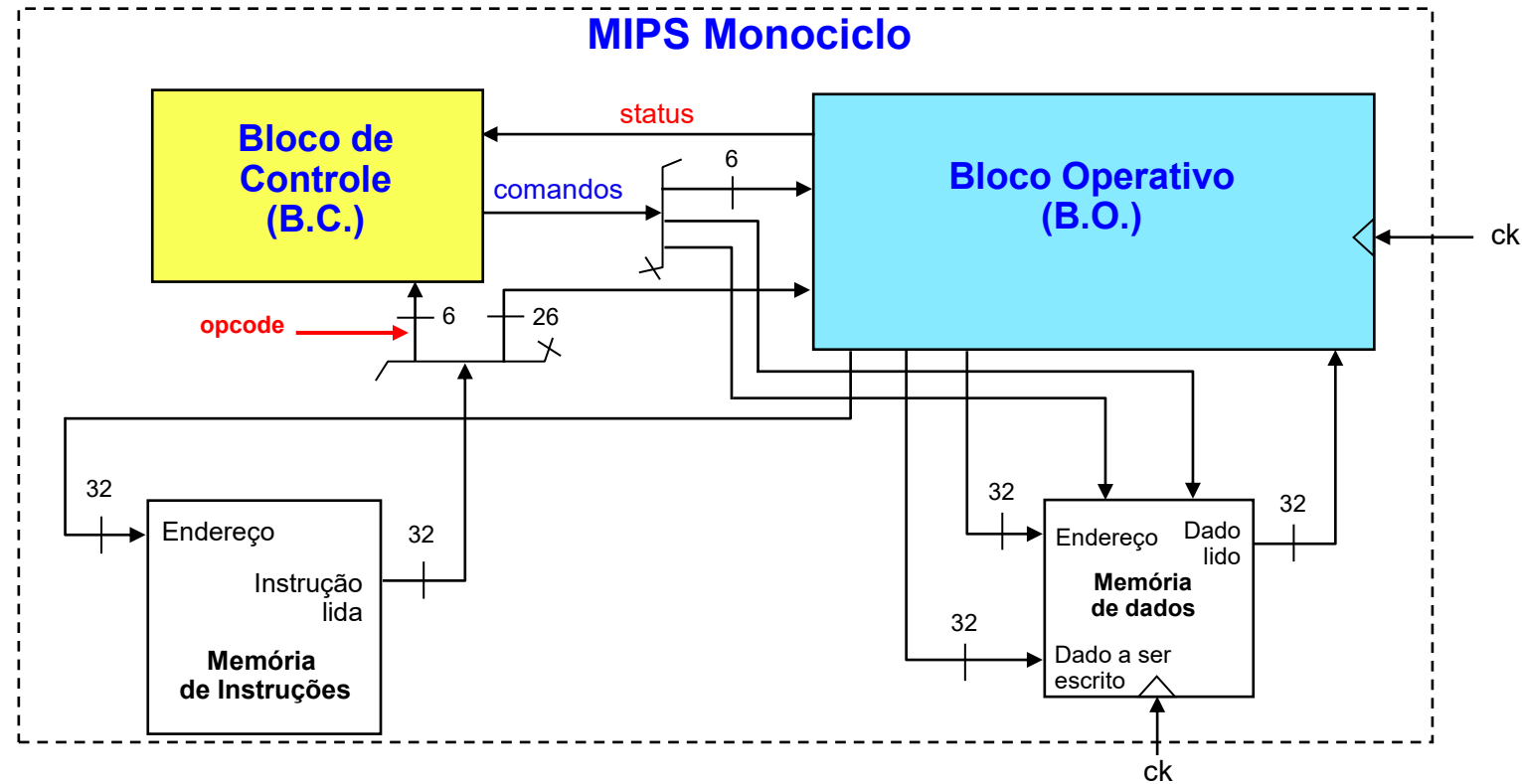
3. O Processador MIPS monociclo: construção do bloco operativo (incluindo a ULA e seu controle).

Profs. José Luís Güntzel e Cristina Meinhardt
{j.guntzel, cristina.meinhardt}@ufsc.br

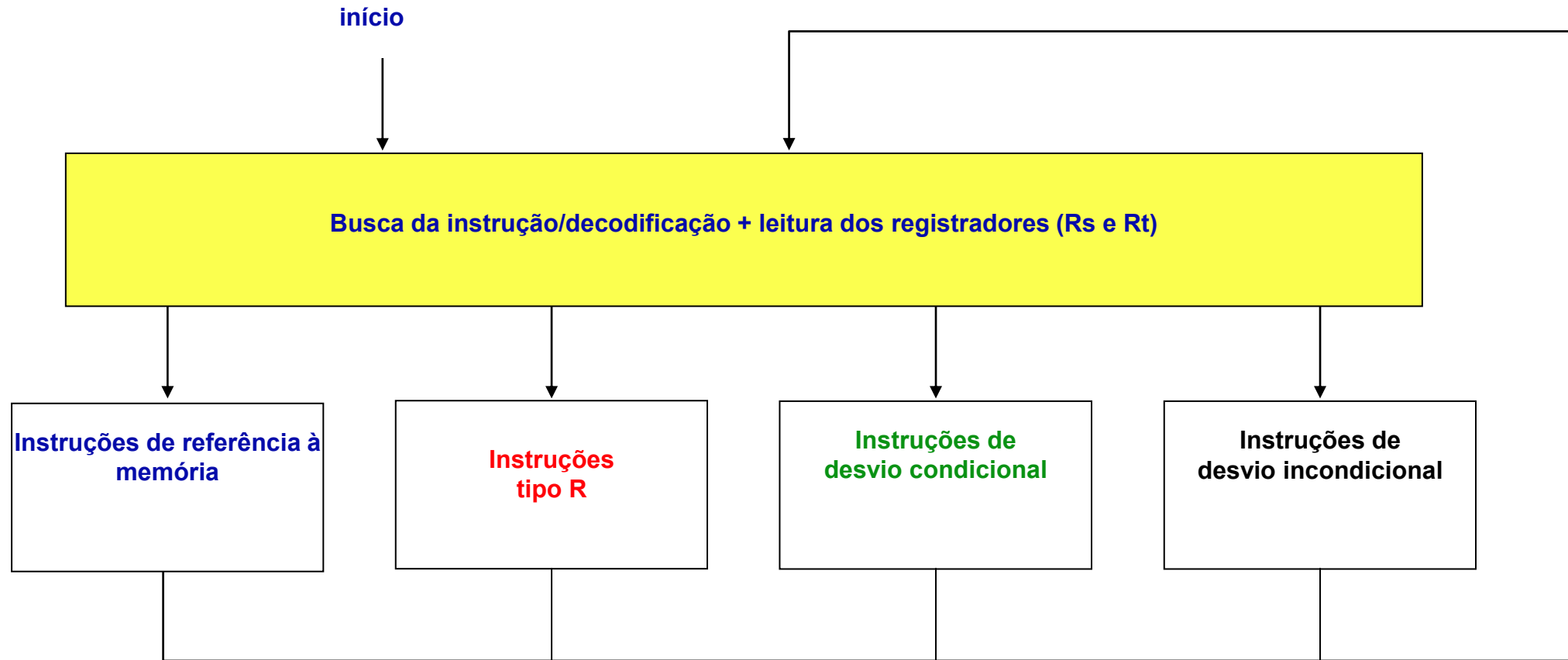
O Processador MIPS Monociclo

Diagrama de Blocos do Sistema

Assumindo que as Memórias estejam embarcadas no chip do MIPS



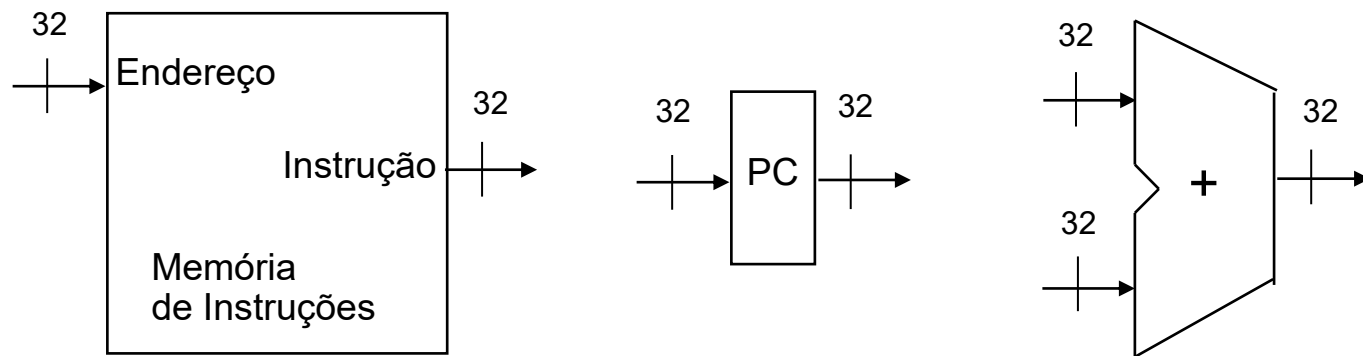
O Processador MIPS Monociclo



O Processador MIPS Monociclo

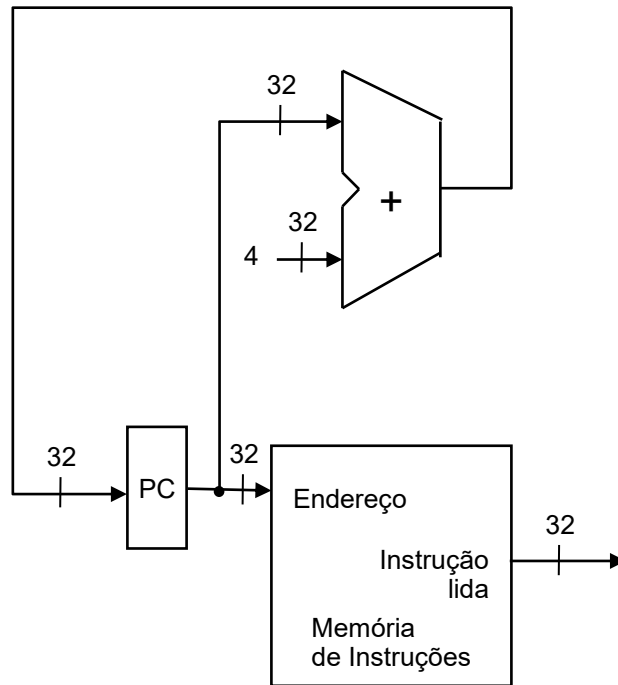
Componentes do Bloco Operativo

- a memória onde estão armazenadas as instruções
- o contador de programa (PC) para armazenar o endereço da instrução
- um somador para calcular o endereço da próxima instrução (alternativamente, o PC poderia ser um registrador-incrementador)



O Processador MIPS Monociclo

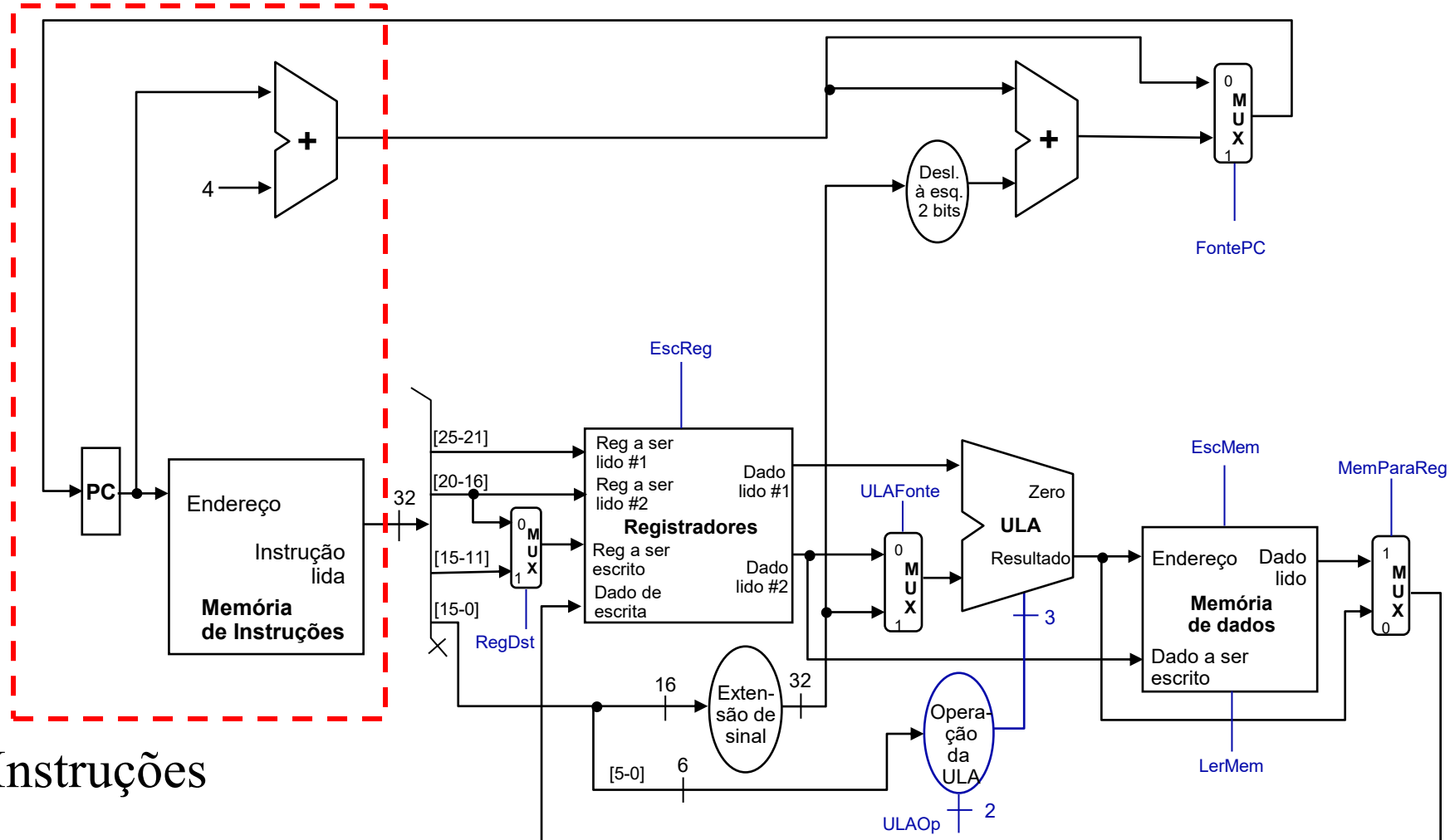
Componentes do Bloco Operativo



- PC (contador de programa): contém o endereço da instrução em execução
- O endereço da próxima instrução é obtido pela soma de 4 posições ao contador de programa
- A instrução lida é usada por outras porções do bloco operativo

O Processador MIPS Monociclo

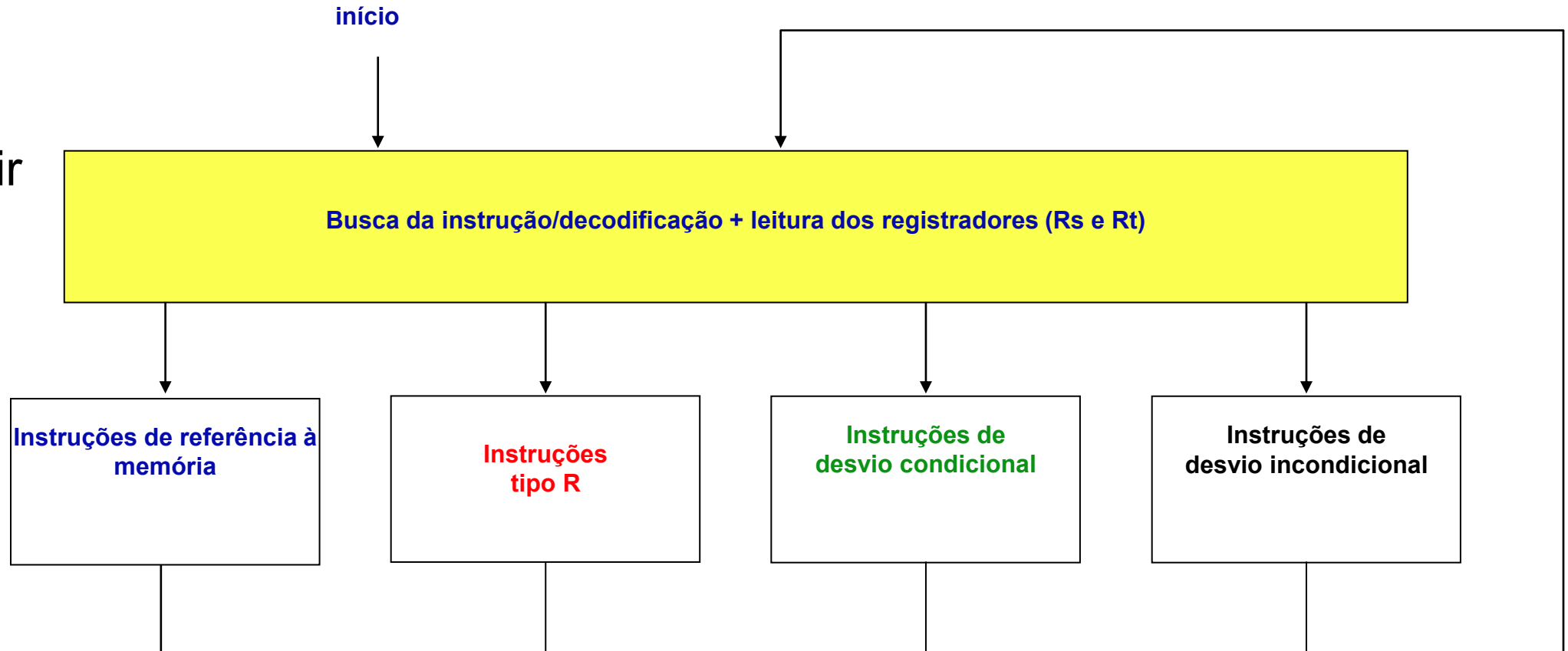
Bloco Operativo



Busca de Instruções

O Processador MIPS Monociclo

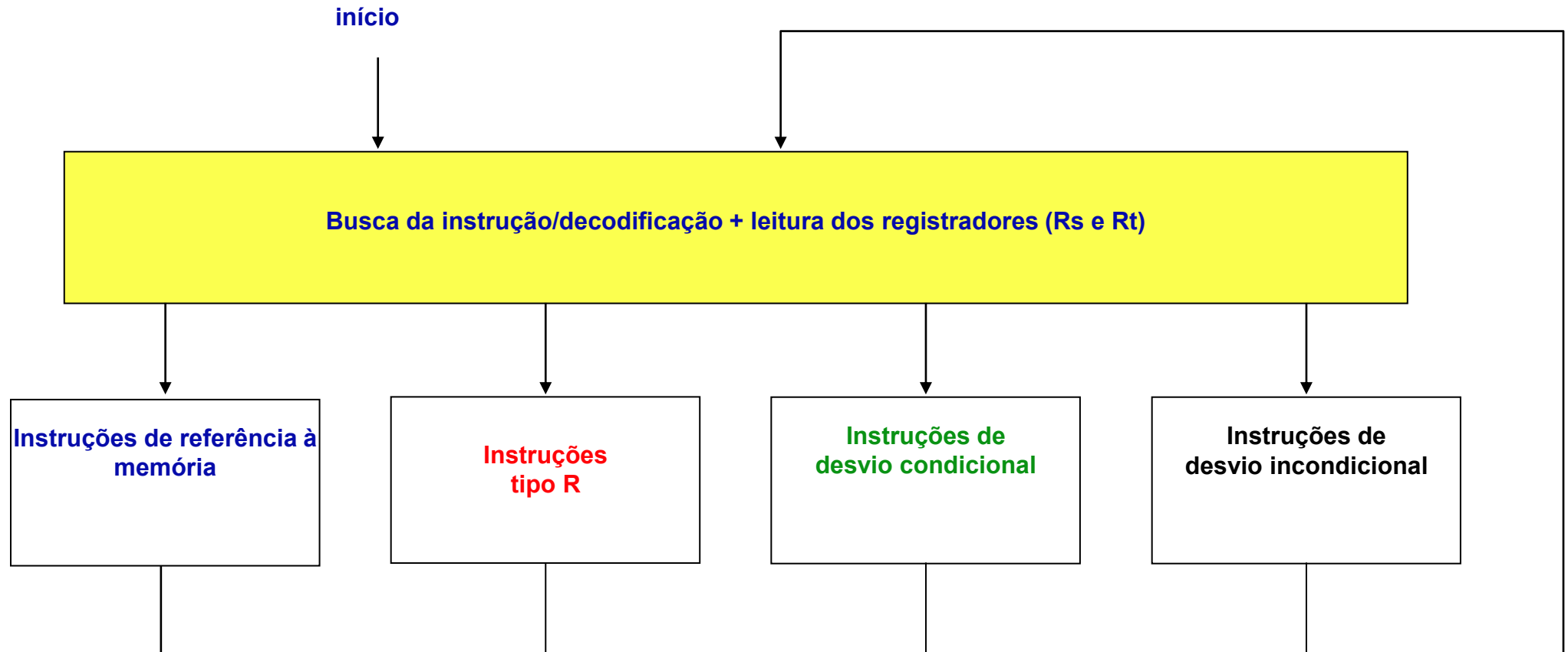
Como definir
que tipo de
Instrução?



O Processador MIPS Monociclo

Como
definir que
tipo de
Instrução?

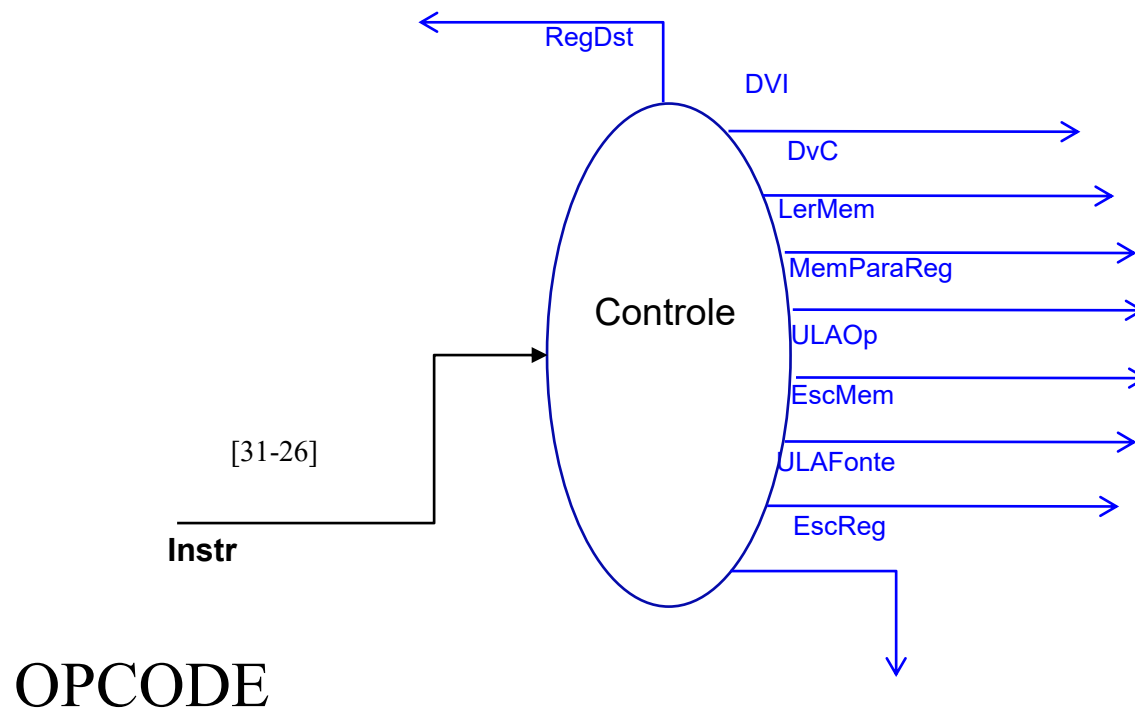
OPCODE!



O Processador MIPS Monociclo

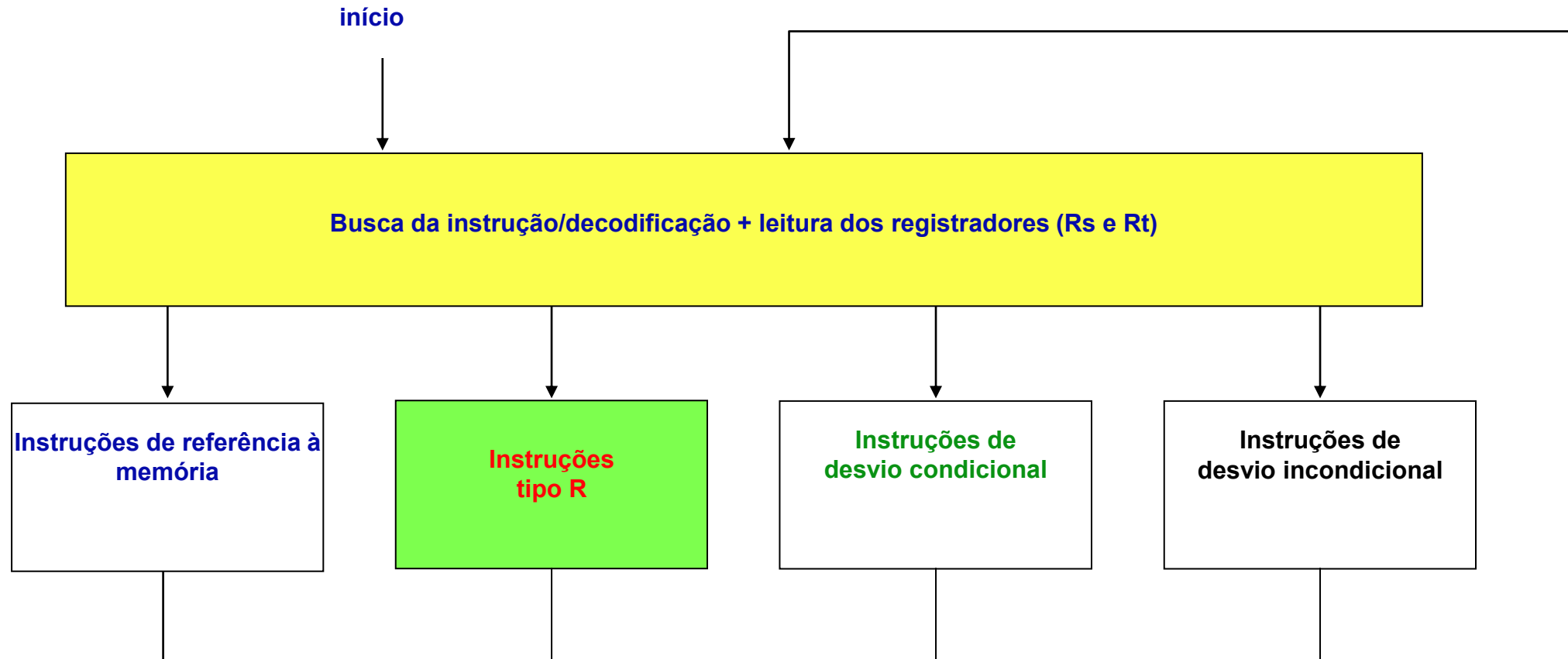
Bloco de Controle

Decodificação



Sinal de Controle	Descrição	Função
DVI	Desvio Incondicional	Controle de MUX
DvC	Desvio Condicional	Entrada de AND
LerMem	Leitura na Memória	Controle Memória
MemParaReg	Memória para Registrador	Controle de MUX
ULAOp	Operação da ULA	Controle da ULA
EscMem	Escrita na Memória	Controle Memória
ULAFonte	Fonte dos operandos da ULA	Controle de MUX
EscReg	Escrita no Banco de Registradores	Controle Banco de Registradores
RegDst	Registrador Destino	Controle de MUX

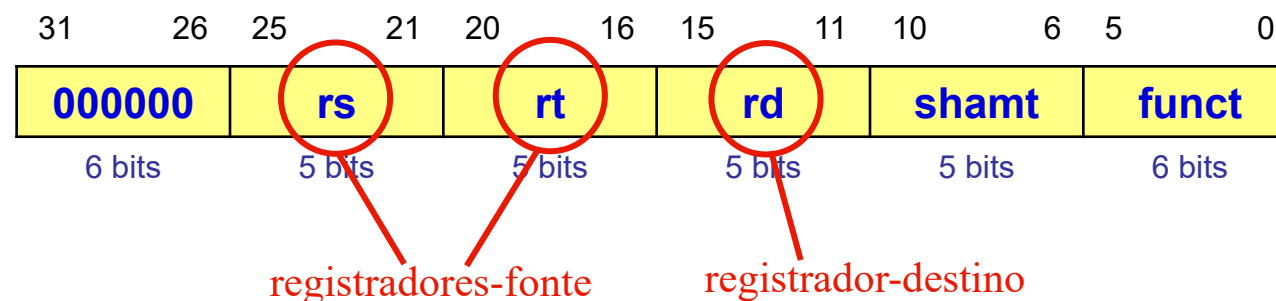
O Processador MIPS Monociclo



O Processador MIPS Monociclo

Instruções formato R: add, sub, or, and

- opcode = 0
- “funct” define a operação a ser feita pela ALU
- “shamt” (shift amount) é usado em instruções de deslocamento

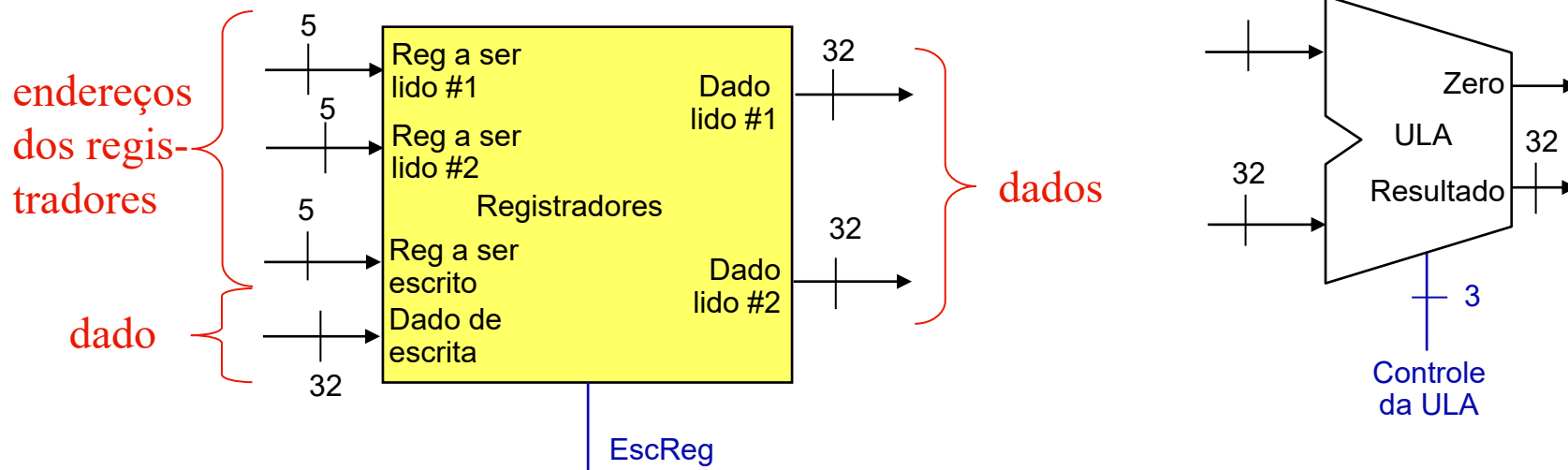


Simbólico (exemplo): `add $s1, $s2, $s3` ($\$s1 \leftarrow \$s2 + \$s3$)

O Processador MIPS Monociclo

Componentes do Bloco Operativo

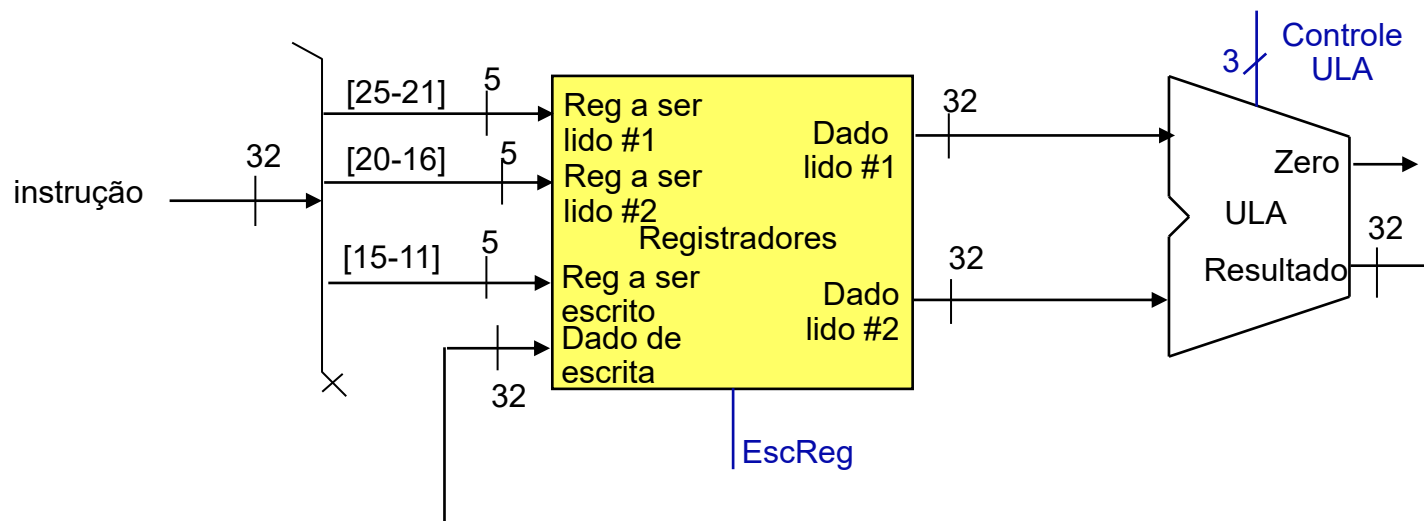
- Um banco de registradores para armazenar os operandos e o resultado das operações
- Uma Unidade Lógica/Aritmética (ULA) que será utilizada para realizar as operações



O Processador MIPS Monociclo

Componentes do Bloco Operativo

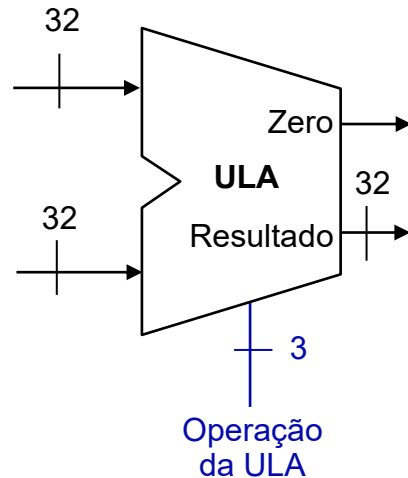
- A instrução (fornecida pelo hardware de busca de instruções) contém o endereço de três registradores
- Dois destes registradores são lidos e passados para a ULA realizar a operação
- O resultado é armazenado em um terceiro registrador
- O controle da ULA determina a operação que será realizada (a partir do campo FNCT da instrução)



O Processador MIPS Monociclo

A ULA do MIPS

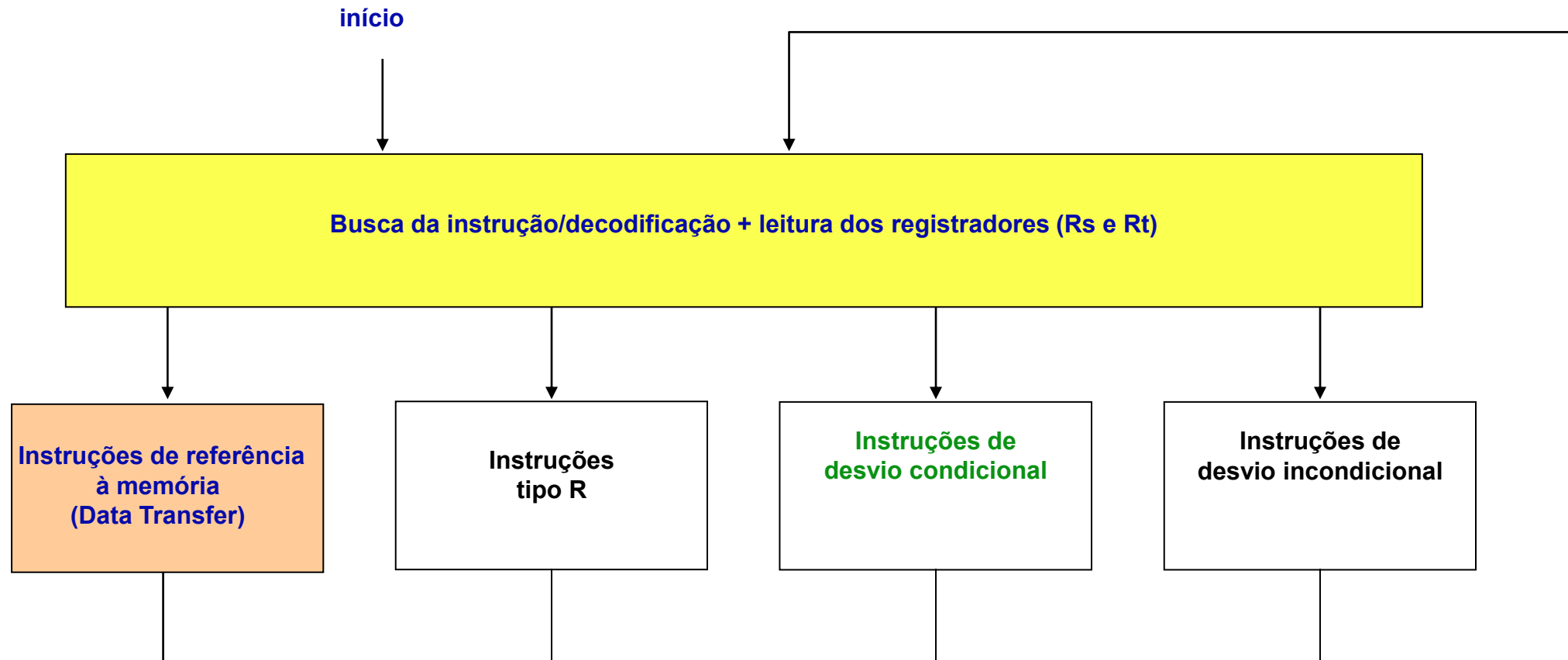
- Operandos com 32 bits
- 1 saída para sinalizar se o resultado = 0 (sinal “Zero”)
- Possui 3 entradas de controle (8 combinações possíveis)
- Somente 5 combinações são usadas



Entrada de controle da ULA	Função
000	AND
001	OR
010	Adição
110	Subtração
111	Set on less than

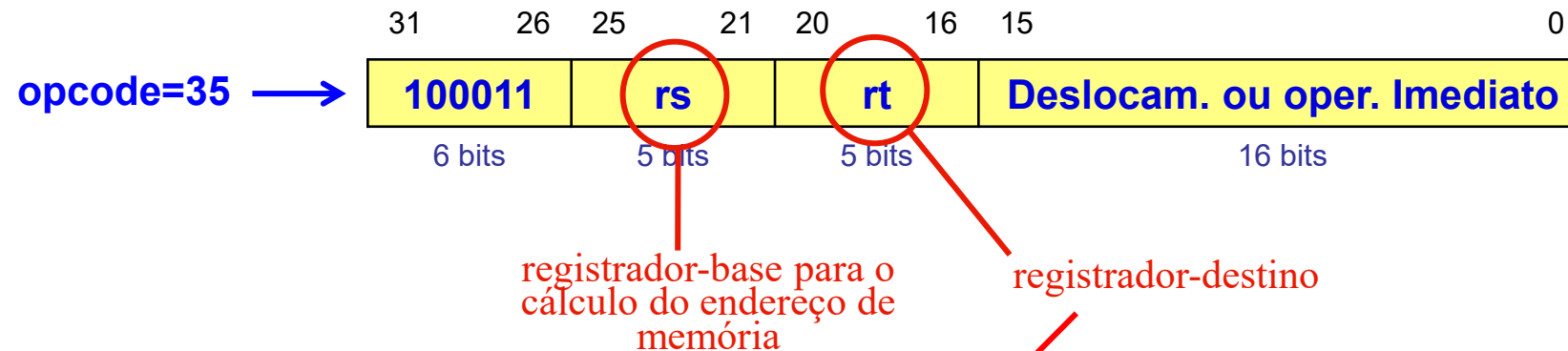
Será estuda com mais profundidade ainda nesta aula

O Processador MIPS Monociclo



O Processador MIPS Monociclo

Instruções formato I: load word (lw)



Simbólico

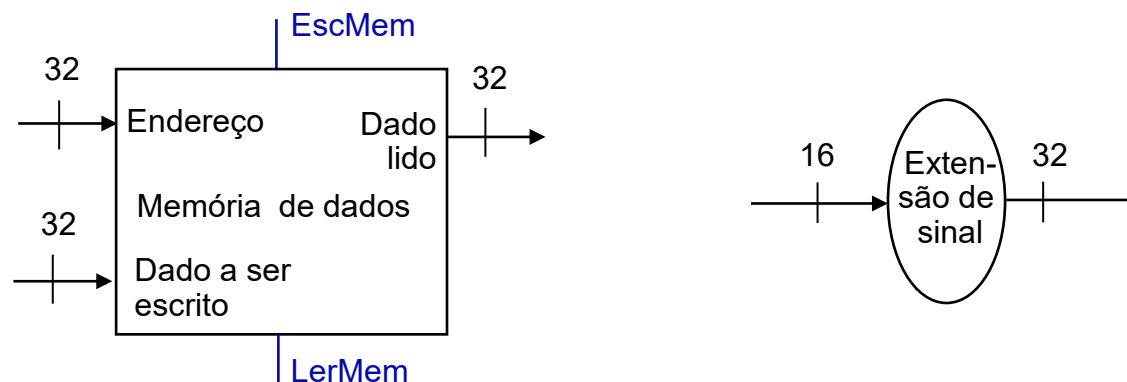
lw \$s1, deslocam(\$s2) ($\$s1 \leftarrow \text{Mem}[\$s2 + \text{deslocam}]$)

O Processador MIPS Monociclo

Componentes do Bloco Operativo

Elementos Necessários para Executar lw (e sw)

- Uma memória de dados
- Um módulo de extensão de sinal
- Um banco de registradores (já mostrado)
- Uma ULA (já mostrada)

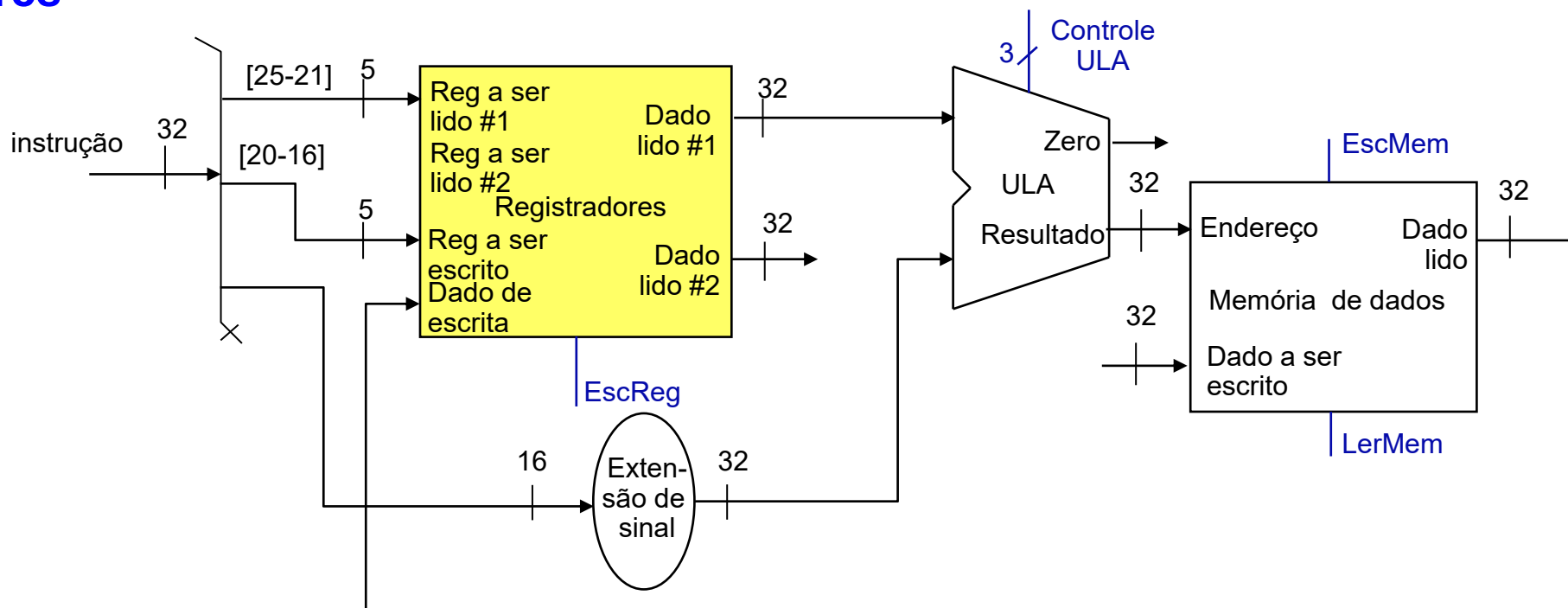


O Processador MIPS Monociclo

Componentes do Bloco Operativo

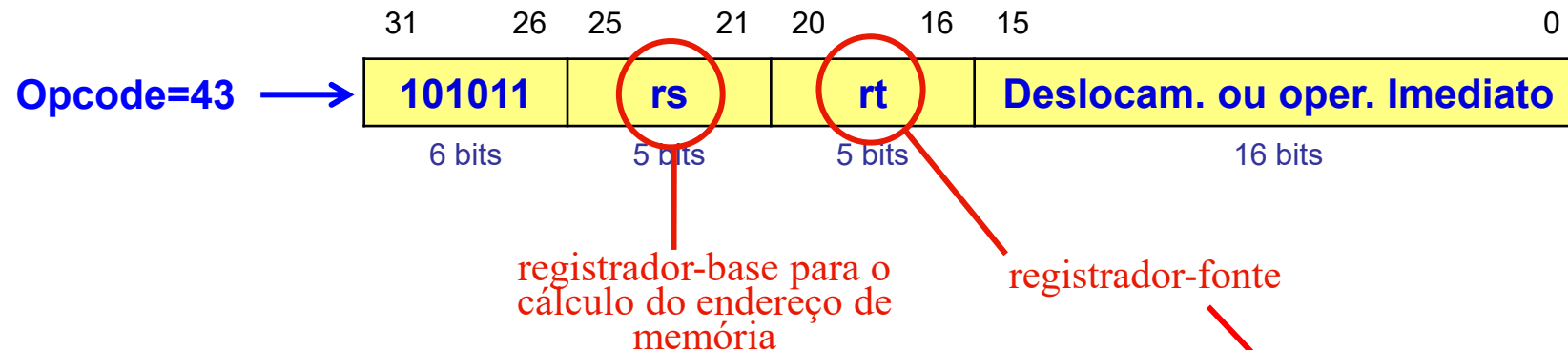
Combinando os Elementos para executar lw

- O processo de leitura é semelhante ao de escrita
- A diferença básica é a existência de um caminho para escrever o valor lido no banco de registradores



O Processador MIPS Monociclo

Instruções formato I: store word (sw)



Simbólico

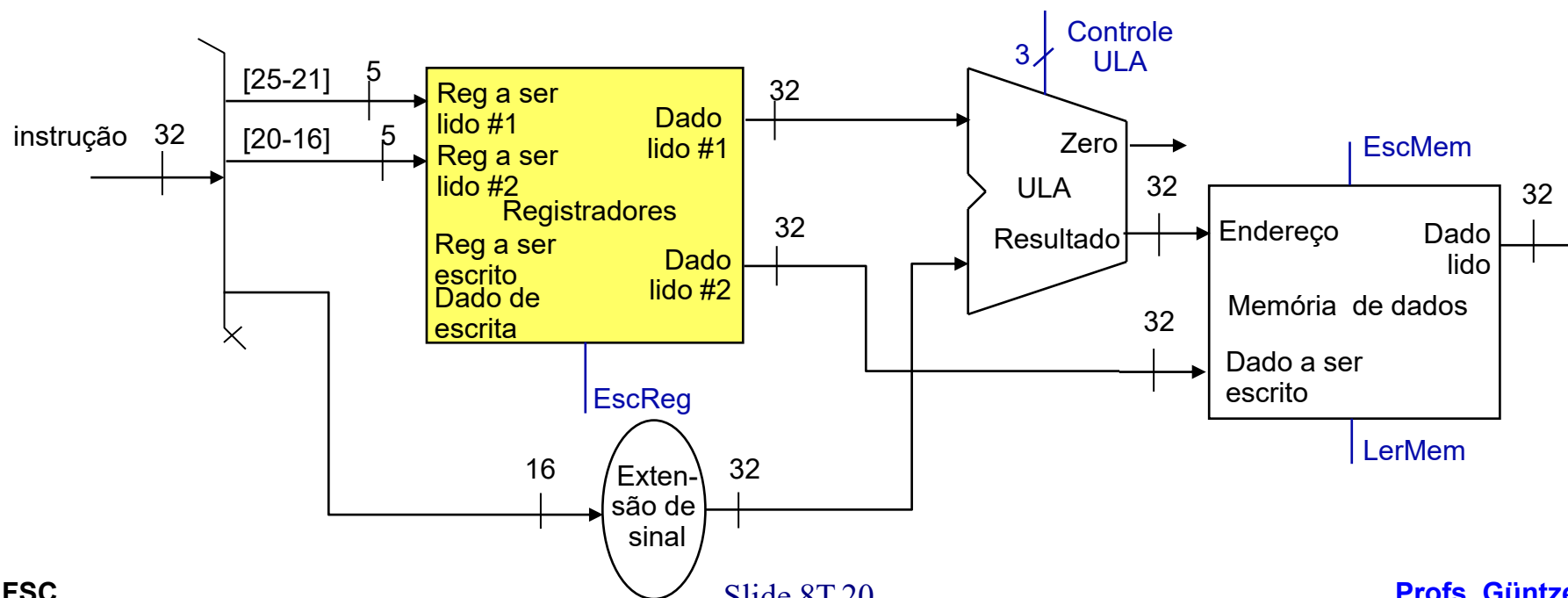
sw \$s1, deslocam(\$s2) (Mem[\$s2 + deslocam] ← \$s1)

O Processador MIPS Monociclo

Componentes do Bloco Operativo

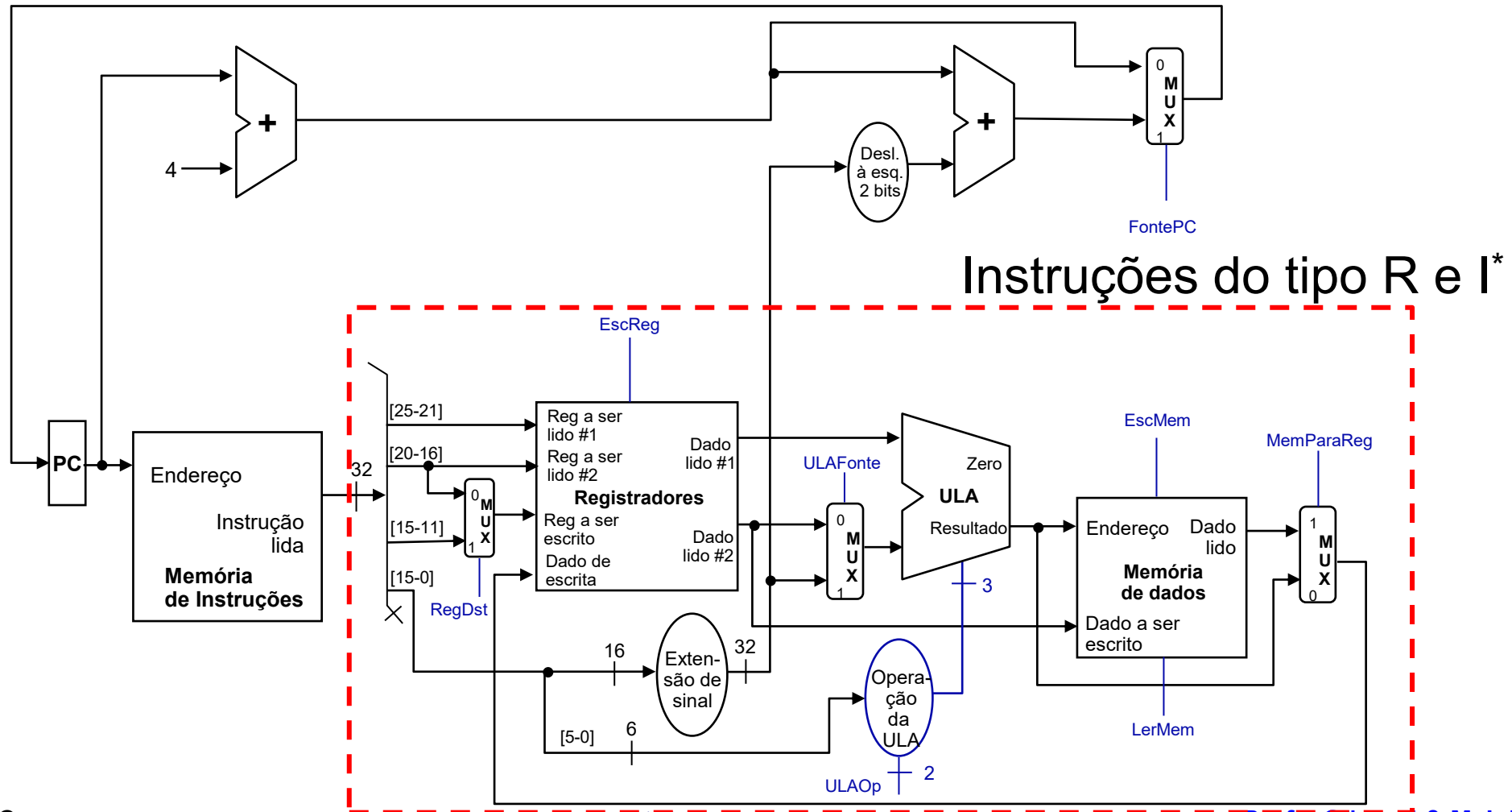
Combinando os Elementos para executar sw

- O endereço de escrita é obtido pela soma de um registrador de base (registrador 1) com um deslocamento de 16 bits estendido para 32 bits
- O registrador 2 é escrito na memória

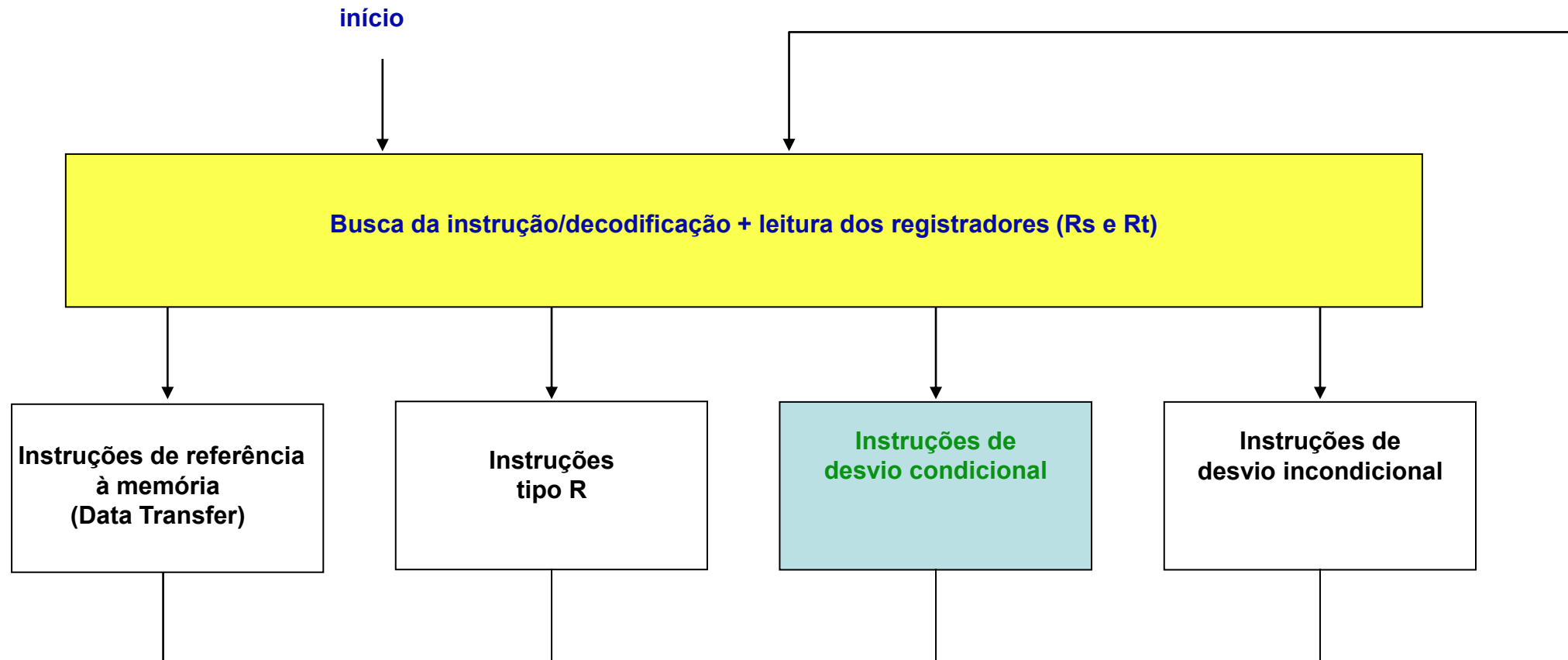


O Processador MIPS Monociclo

Bloco Operativo



O Processador MIPS Monociclo

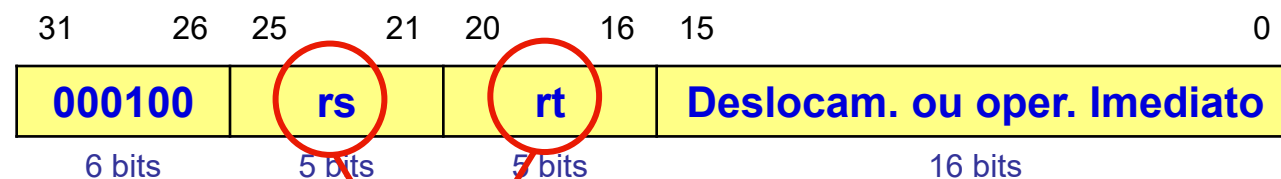


O Processador MIPS Monociclo

Instrução formato I: Desvio Condicional

beq: branch on equal

- Opcode = 4
- Campo deslocamento usado para calcular o endereço-alvo
- Se o conteúdo do registrador cujo endereço está no campo rs for igual ao conteúdo do registrador cujo endereço está em rt, então salta para a posição endereço+PC+4



registradores a serem
comparados

Simbólico

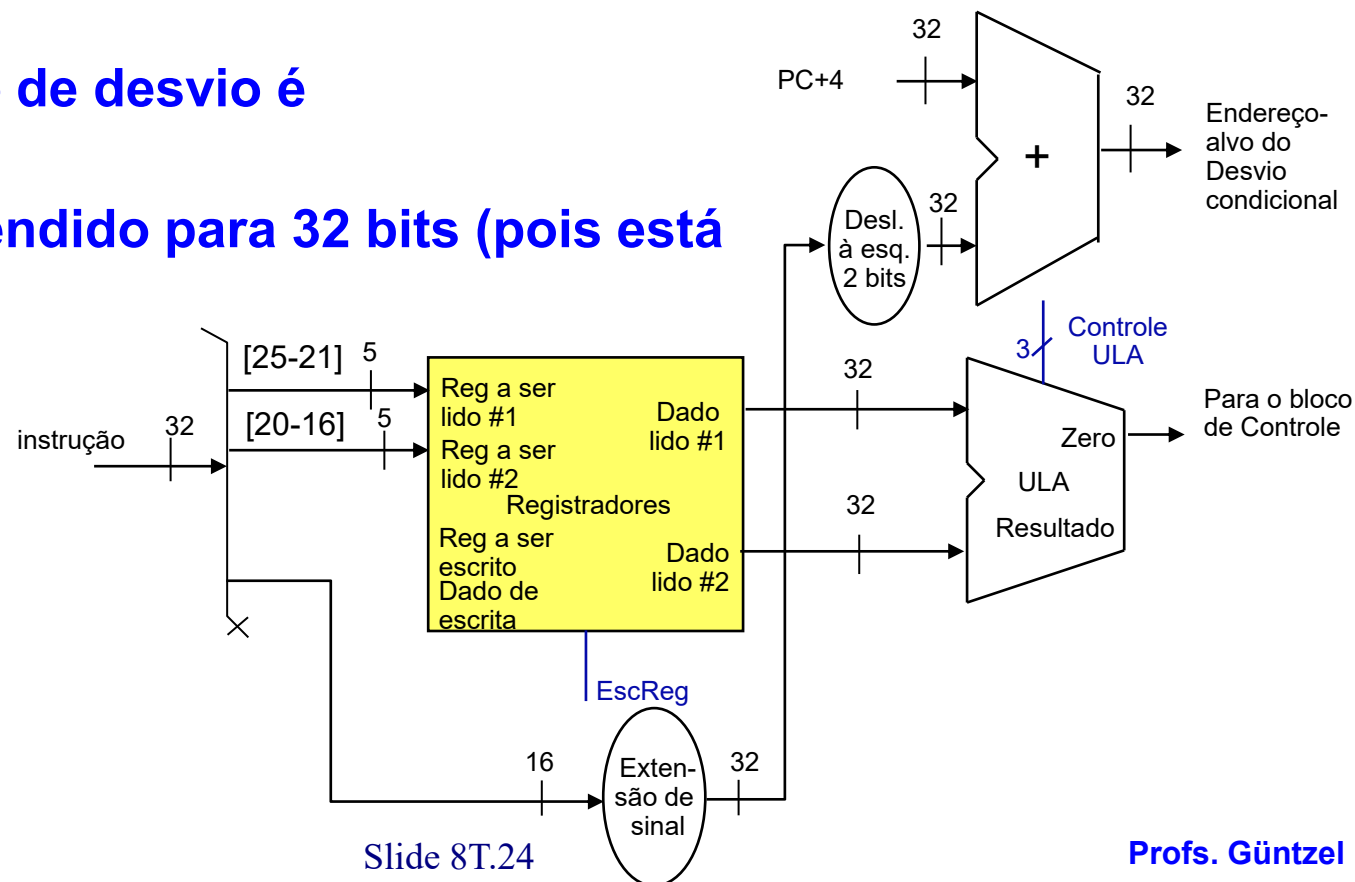
beq \$s1, \$s2, deslocam (if ($\$s1 == \$s2$) then $PC \leftarrow PC + 4 + \text{deslocam}$)

O Processador MIPS Monociclo

Componentes do Bloco Operativo

Elementos Necessários para Implementar um Branch on Equal

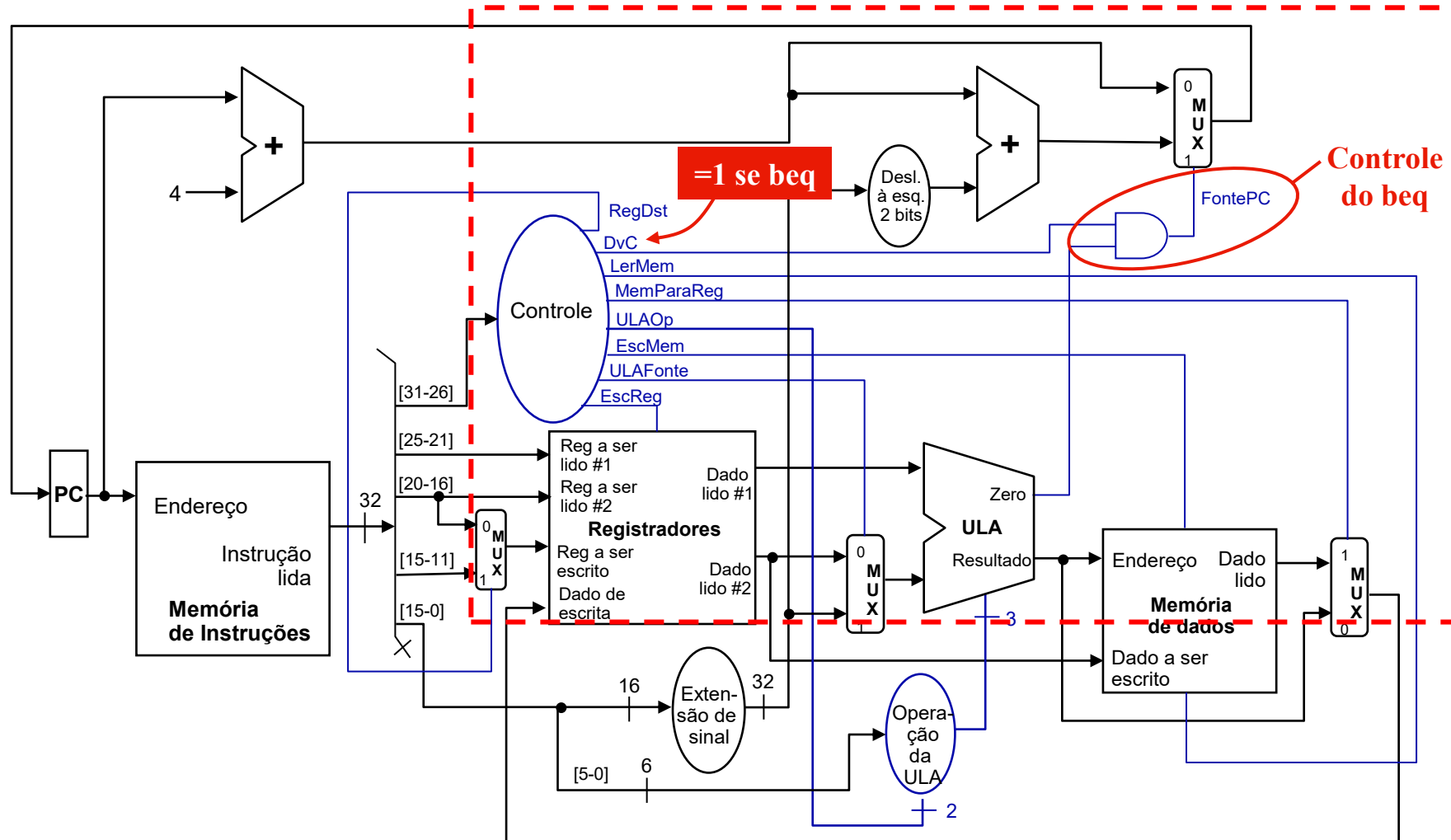
- Comparar dois registradores usando a ULA para fazer uma subtração
- Se ocorrer desvio, o endereço de desvio é $PC+4+\text{deslocamento}$
- O deslocamento deve ser estendido para 32 bits (pois está em 16 bits)



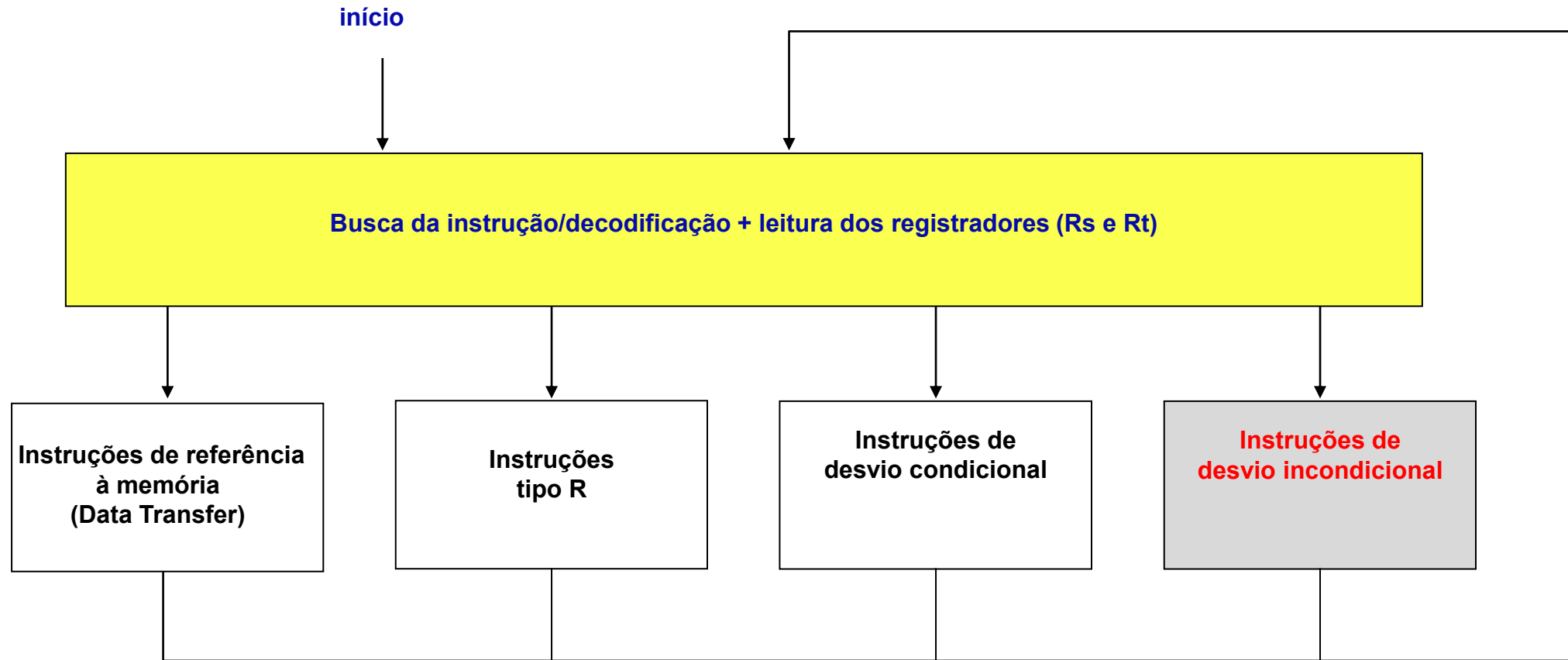
O Processador MIPS Monociclo

Bloco Operativo + Bloco de Controle

Instruções I (BEQ)



O Processador MIPS Monociclo

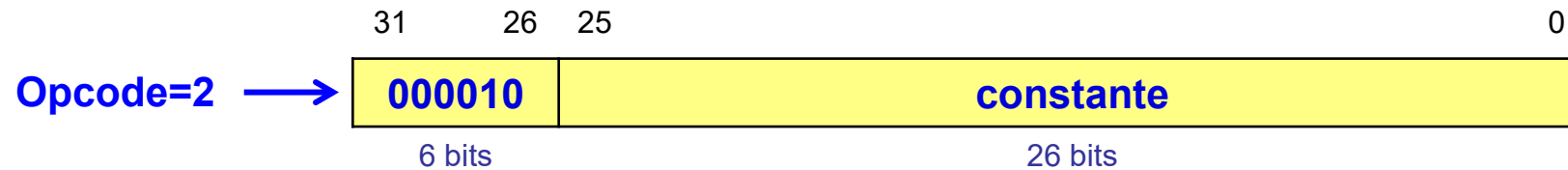


O Processador MIPS Monociclo

Instrução formato J: Desvio Incondicional

j: jump

- Opcode = 2
- Constante de 26 bits usada para calcular o endereço-alvo de desvio



Simbólico

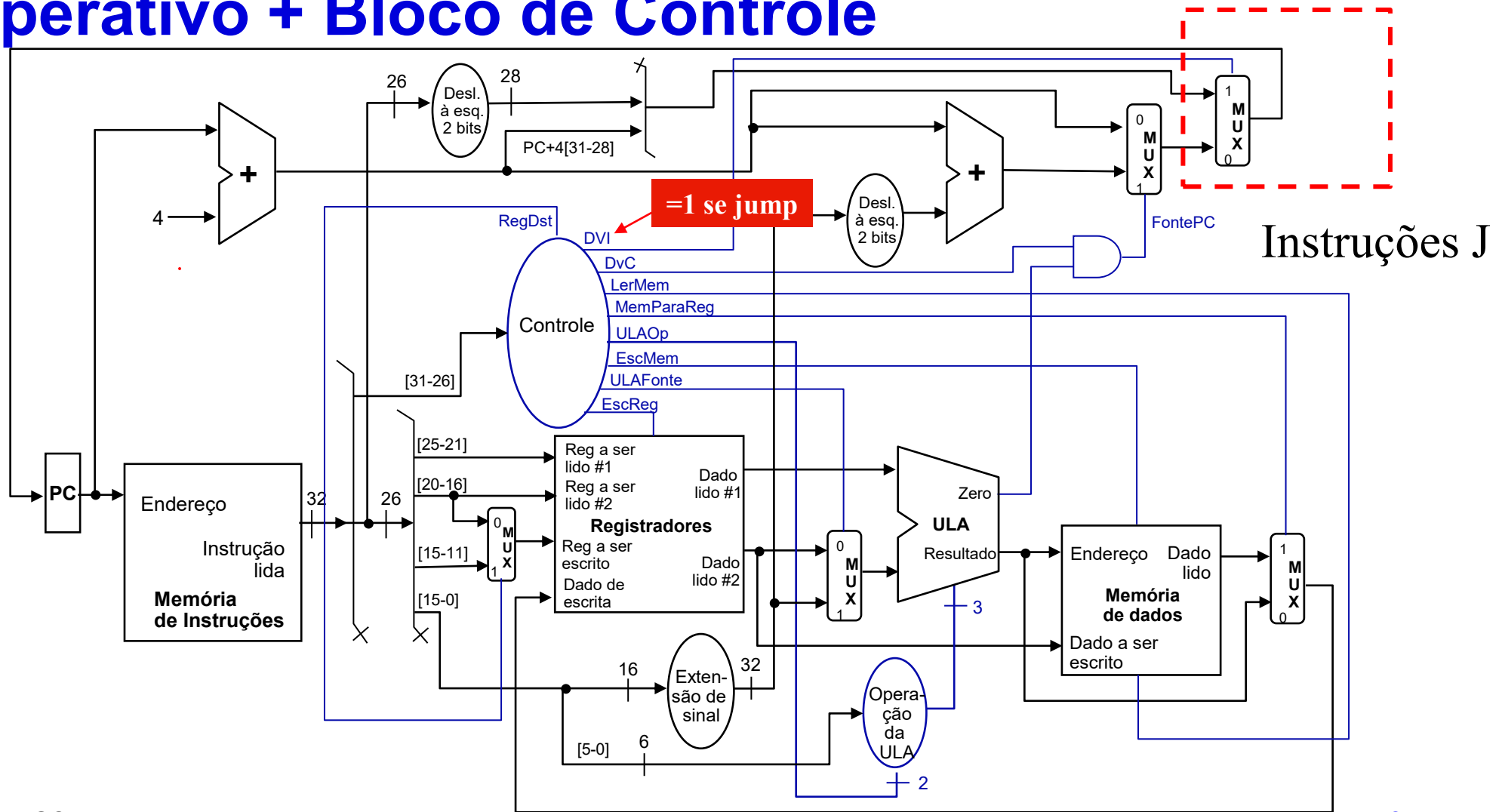
j endereço ($PC \leftarrow \text{endereço}$), onde $\text{endereço} = (PC+4)[31-28] \parallel \text{constante} \ll 2$

Concatenação de bits



O Processador MIPS Monociclo

Bloco Operativo + Bloco de Controle



O Processador MIPS Monociclo

Atividade de Fixação:

Reforce o conteúdo visto nesta aula refazendo os passos do projeto do bloco operativo, desenhando passo-a-passo em folhas diferentes cada etapa, e, em uma nova folha, desenhe a interligação de cada parte, formando o desenho final do bloco operativo e bloco de controle.

O Processador MIPS Monociclo

Como o MIPS usa a ULA?

Instrução	Formato	Linguagem de Montagem	Significado
Adição	R	add \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 + \$s3$
Subtração	R	sub \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 - \$s3$
AND bit a bit	R	and \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 \text{ and } \$s3$
OR bit a bit	R	or \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 \text{ or } \$s3$
Load word	I	lw \$s1, desl(\$s2)	$\$s1 \leftarrow \text{Mem}[\$s2 + \text{desl}]$
Store word	I	sw \$s1, desl(\$s2)	$\text{Mem}[\$s2 + \text{desl}] \leftarrow \$s1$
Salto condicional	I	beq \$s1, \$s2, desl	if ($\$s1 == \$s2$) then $\text{PC} \leftarrow \text{PC} + 4 + (\text{desl} \ll 2)$
Salto incondicional	J	j L	$\text{PC} \leftarrow L$ onde $L = ((\text{PC} + 4)[31-28]) \parallel (\text{constante} \ll 2)$

O Processador MIPS Monociclo

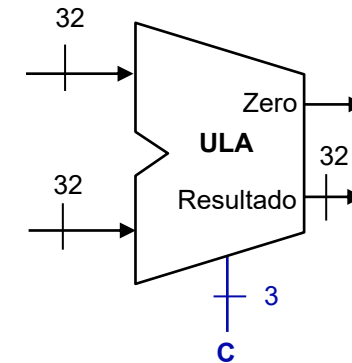
Como o MIPS usa a ULA?

Instrução	Formato	Linguagem de Montagem	Significado	Uso da ULA
Adição	R	add \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 + \$s3$	Definido pelo campo "funct"
Subtração	R	sub \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 - \$s3$	
AND bit a bit	R	and \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 \text{ and } \$s3$	
OR bit a bit	R	or \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 \text{ or } \$s3$	
Load word	I	lw \$s1, des1(\$s2)	$\$s1 \leftarrow \text{Mem}[\$s2 + \text{des1}]$	adição
Store word	I	sw \$s1, des1(\$s2)	$\text{Mem}[\$s2 + \text{des1}] \leftarrow \$s1$	adição
Salto condicional	I	beq \$s1, \$s2, des1	if ($\$s1 == \$s2$) then $\text{PC} \leftarrow \text{PC} + 4 + (\text{des1} \ll 2)$	Subtração
Salto incondicional	J	j L	$\text{PC} \leftarrow L$ onde $L = ((\text{PC} + 4)[31-28]) \mid\mid (\text{constante} \ll 2)$	---

O Processador MIPS Monociclo

Como Funciona a ULA do MIPS?

Entrada de controle da ULA c2 c1 c0	Operação
000	AND
001	OR
010	Adição
110	Subtração
111	Set on less than



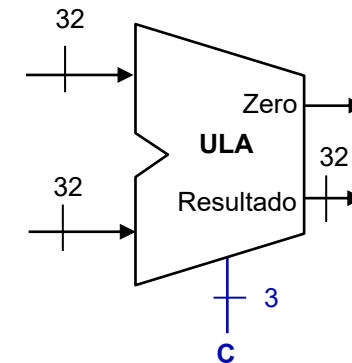
O Processador MIPS Monociclo

Será Preciso Mapear as Operações

Instrução	Significado	Uso da ULA
Adição	$\$s1 \leftarrow \$s2 + \$s3$	Definido pelo campo "funct"
Subtração	$\$s1 \leftarrow \$s2 - \$s3$	
AND bit a bit	$\$s1 \leftarrow \$s2 \text{ and } \$s3$	
OR bit a bit	$\$s1 \leftarrow \$s2 \text{ or } \$s3$	
Load word	$\$s1 \leftarrow \text{Mem}[\$s2 + \text{desl}]$	adição
Store word	$\text{Mem}[\$s2 + \text{desl}] \leftarrow \$s1$	adição
Salto condicional	if ($\$s1 == \$s2$) then $\text{PC} \leftarrow \text{PC} + 4 + (\text{desl} \ll 2)$	Subtração
Salto incondicional	$\text{PC} \leftarrow \text{L onde}$ $\text{L} = ((\text{PC} + 4)[31-28]) \mid (\text{constante} \ll 2)$	---



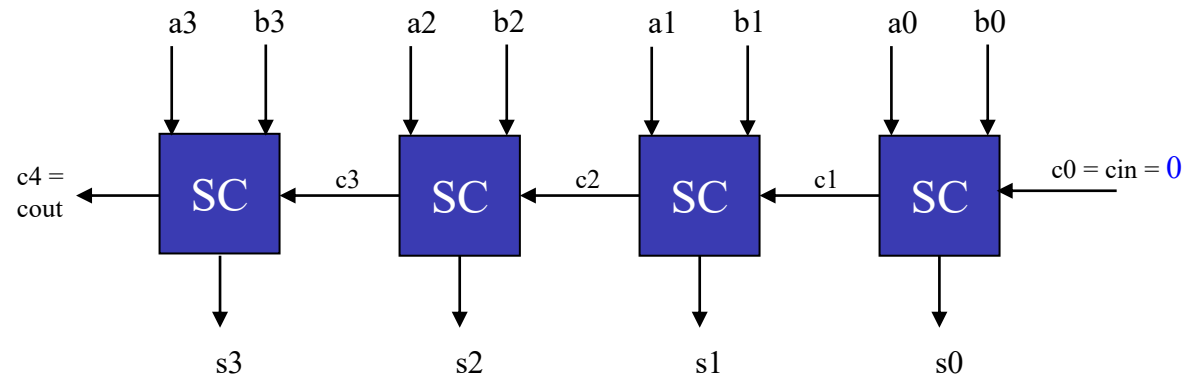
Entrada de controle da ULA c2 c1 c0	Operação
000	AND
001	OR
010	Adição
110	Subtração
111	Set on less than



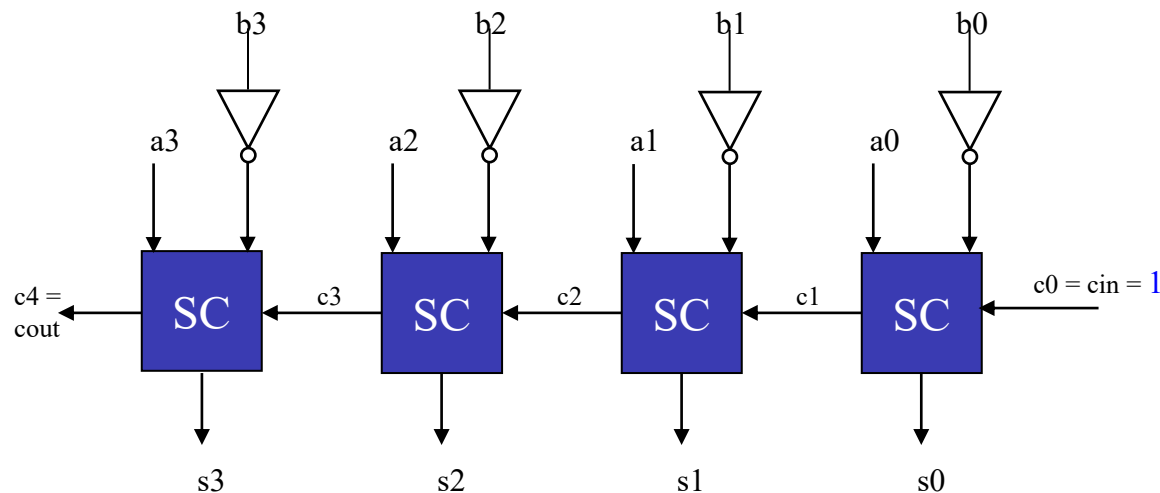
Construindo uma ULA

Somador/Subtrator Paralelo (de 4 bits)

Somador



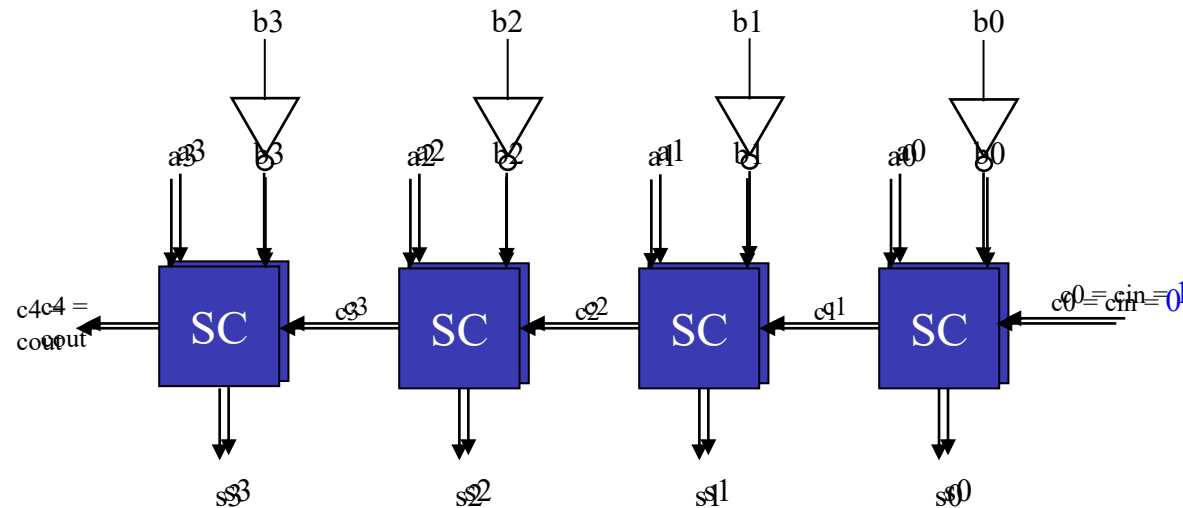
Subtrator



Construindo uma ULA

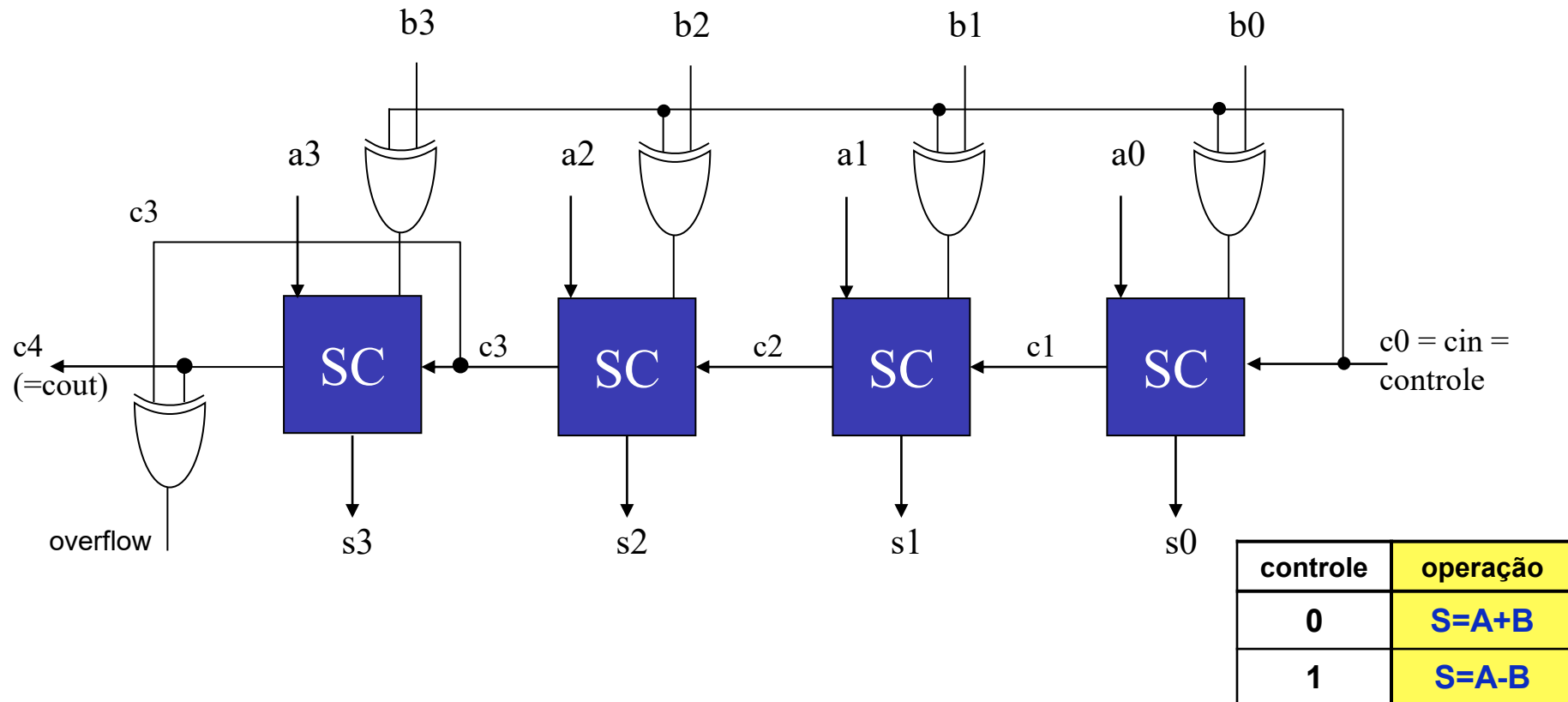
Somador/Subtrator Paralelo (de 4 bits)

Como uni-los em um único circuito, configurável?



Construindo uma ULA

Somador/Subtrator Paralelo (de 4 bits)



Construindo uma ULA

Somador/Subtrator Paralelo (de 4 bits)

Símbolo no Nível RT

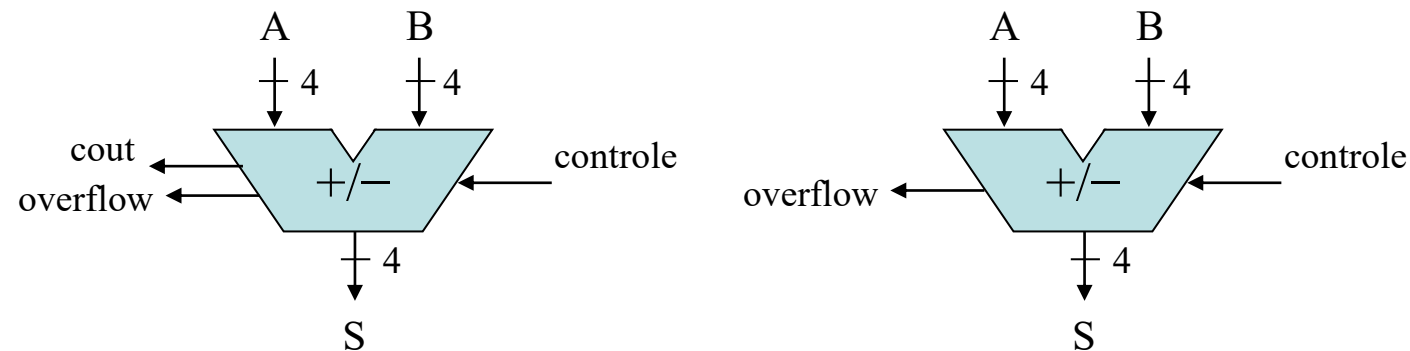


Tabela de Operação

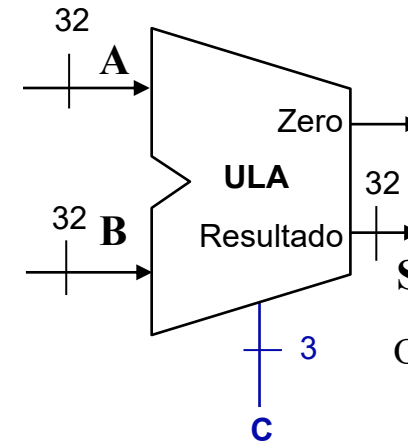
controle	operação
0	$S=A+B$
1	$S=A-B$

O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	Operação	
000	AND	$S = A \text{ AND } B$
001	OR	$S = A \text{ OR } B$
010	Adição	$S = A + B$
110	Subtração	$S = A - B$
111	Set on less than	$S = 1$ se $A < B$ senão, $S = 0$

Símbolo no nível RT



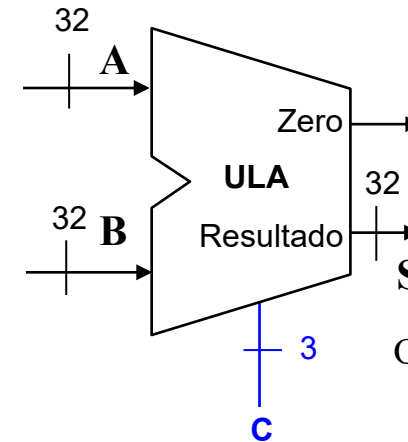
Obs: o sinal de overflow está omitido

O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	Operação	
000	AND	$S = A \text{ AND } B$
001	OR	$S = A \text{ OR } B$
010	Adição	$S = A + B$
110	Subtração	$S = A - B$
111	Set on less than	$S = 1 \text{ se } A < B$ senão, $S = 0$

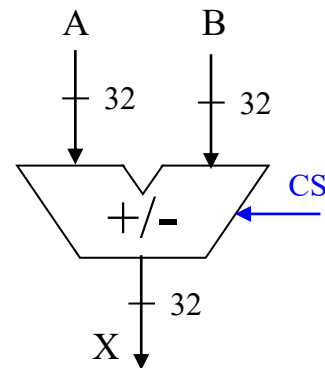
Símbolo no nível RT



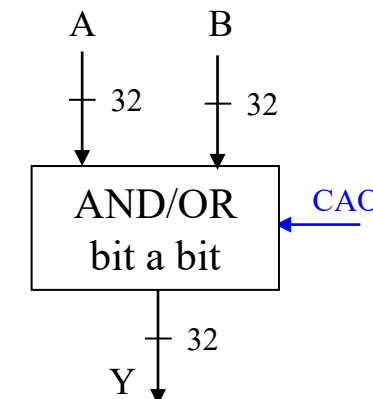
Obs: o sinal de overflow está omitido

Usaremos os seguintes componentes RT:

C S	operação
0	$X = A + B$
1	$X = A - B$



CAO	operação
0	$Y = A \text{ AND } B$
1	$Y = A \text{ OR } B$

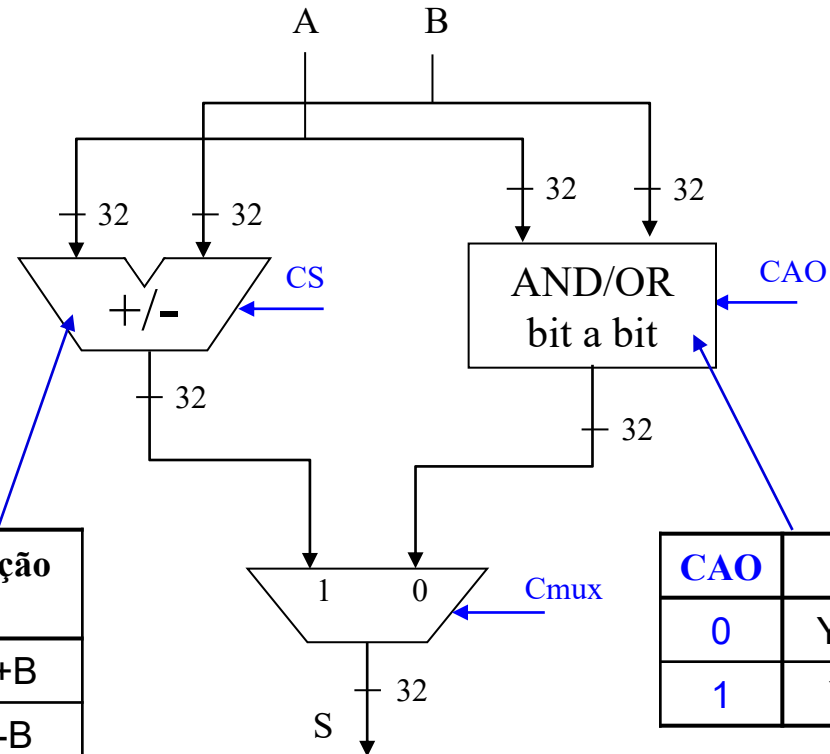


O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$

C S	operação
0	$X = A + B$
1	$X = A - B$

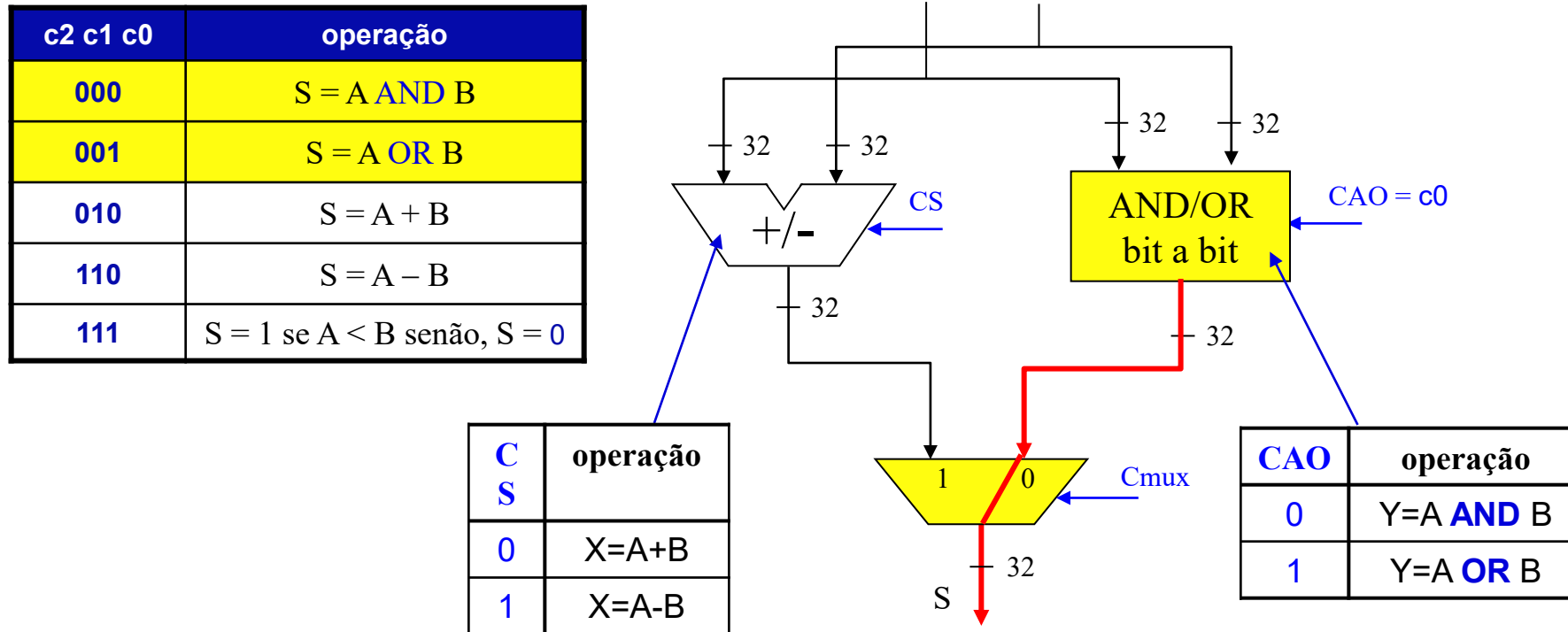


CAO	operação
0	$Y = A \text{ AND } B$
1	$Y = A \text{ OR } B$

Associar os sinais {c2, c1, c0} às entradas de controle {CS, CAO, Cmux} para que este circuito realize as primeiras 4 linhas da tabela de operações da ULA do MIPS

O Processador MIPS Monociclo

A ULA do MIPS

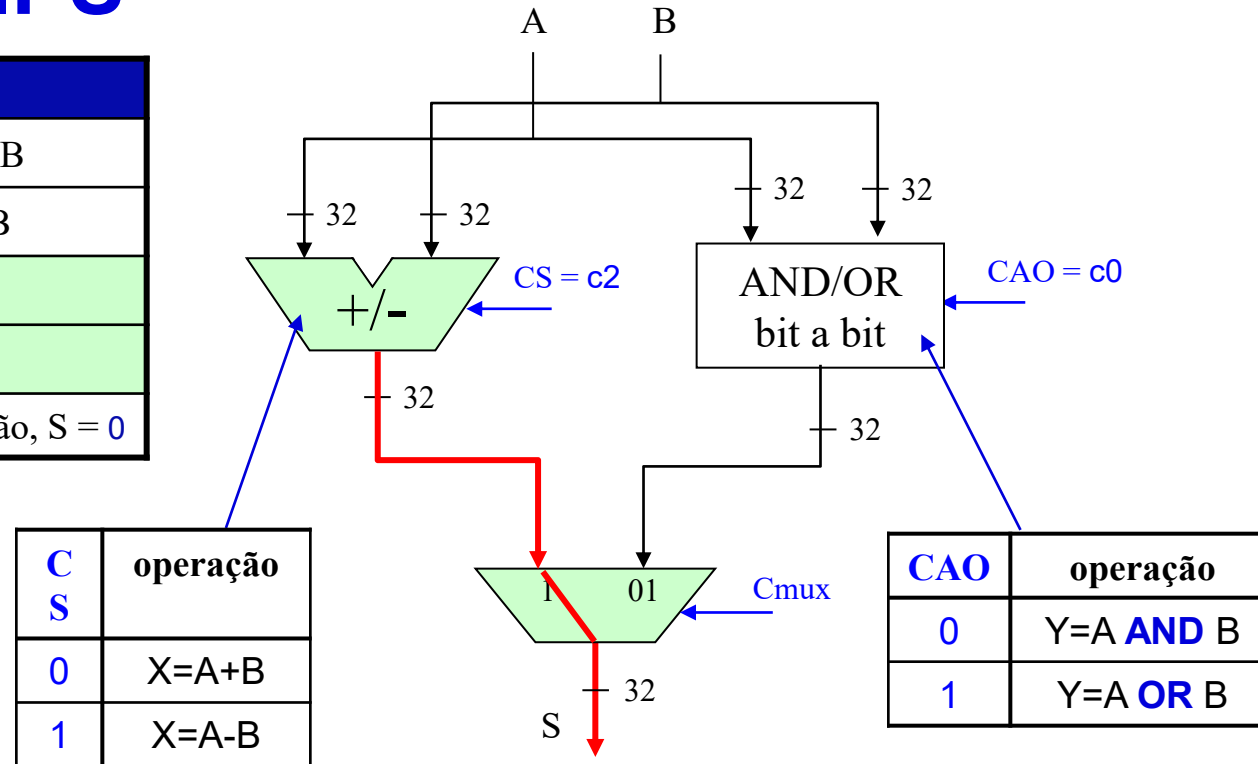


Associar os sinais {**c2**, **c1**, **c0**} às entradas de controle {**CS**, **CAO**, **Cmux**} para que este circuito realize as primeiras 4 linhas da tabela de operações da ULA do MIPS

O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$



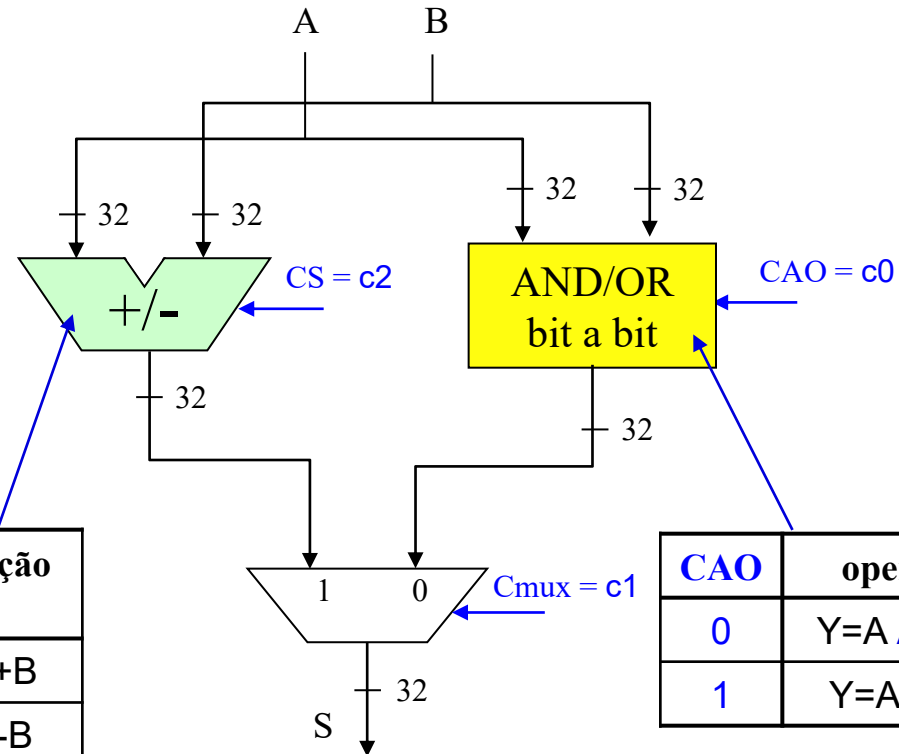
Associar os sinais {c2, c1, c0} às entradas de controle {CS, CAO, Cmux} para que este circuito realize as primeiras 4 linhas da tabela de operações da ULA do MIPS

O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$

C	operação
S	
0	$X = A + B$
1	$X = A - B$



CAO	operação
0	$Y = A \text{ AND } B$
1	$Y = A \text{ OR } B$

Resumindo:

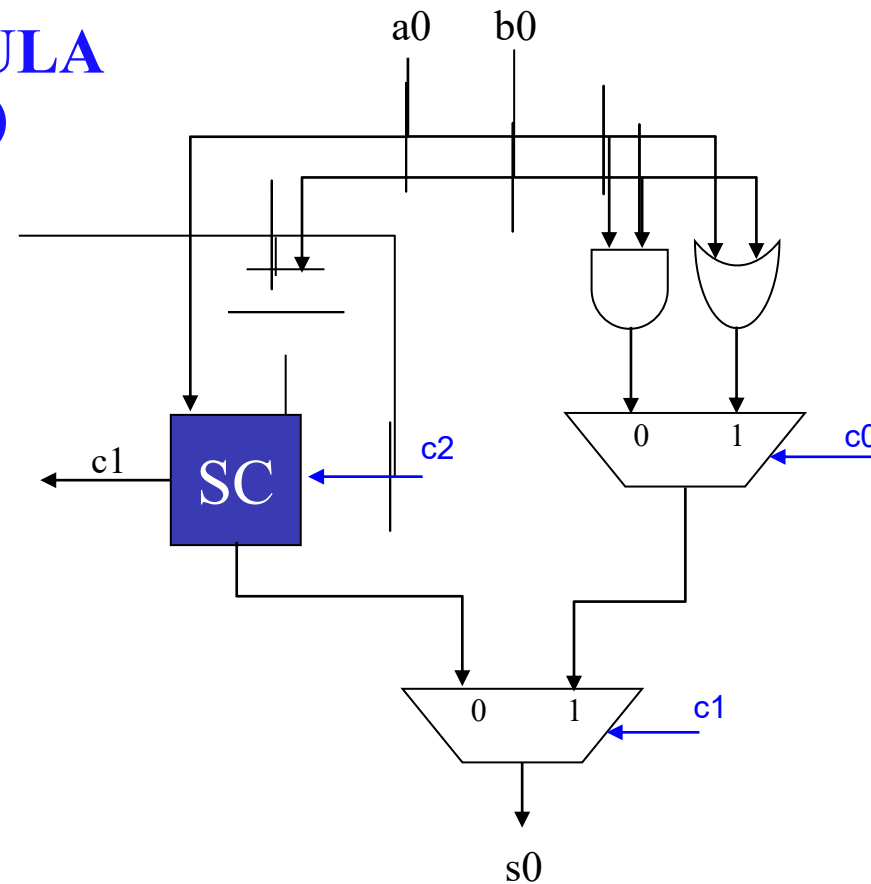
$CS = c2$; $CAO = c0$; $Cmux = c1$

O Processador MIPS Monociclo

A ULA do MIPS

Visão parcial de um bit desta ULA
(os demais bits serão similares)

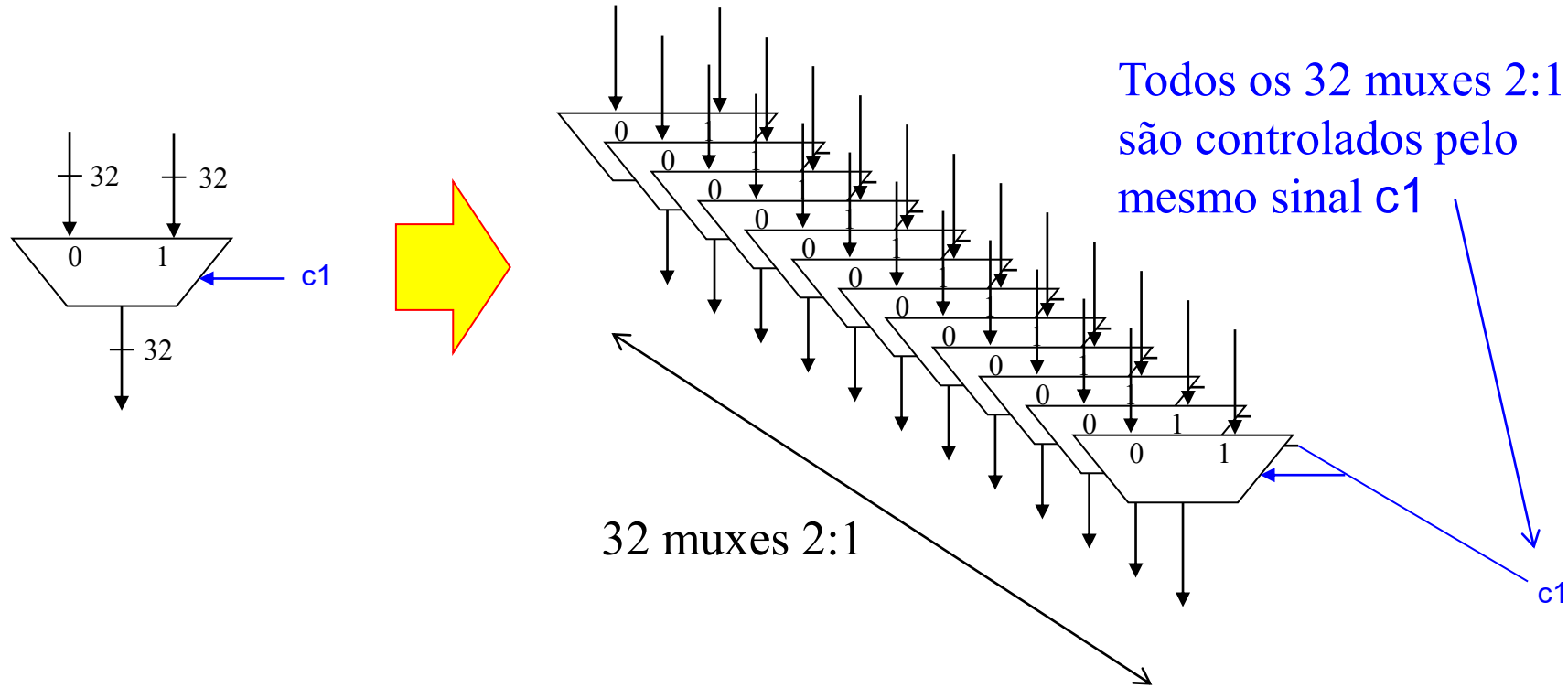
c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$



O Processador MIPS Monociclo

A ULA do MIPS

Multiplexador no Nível RT...

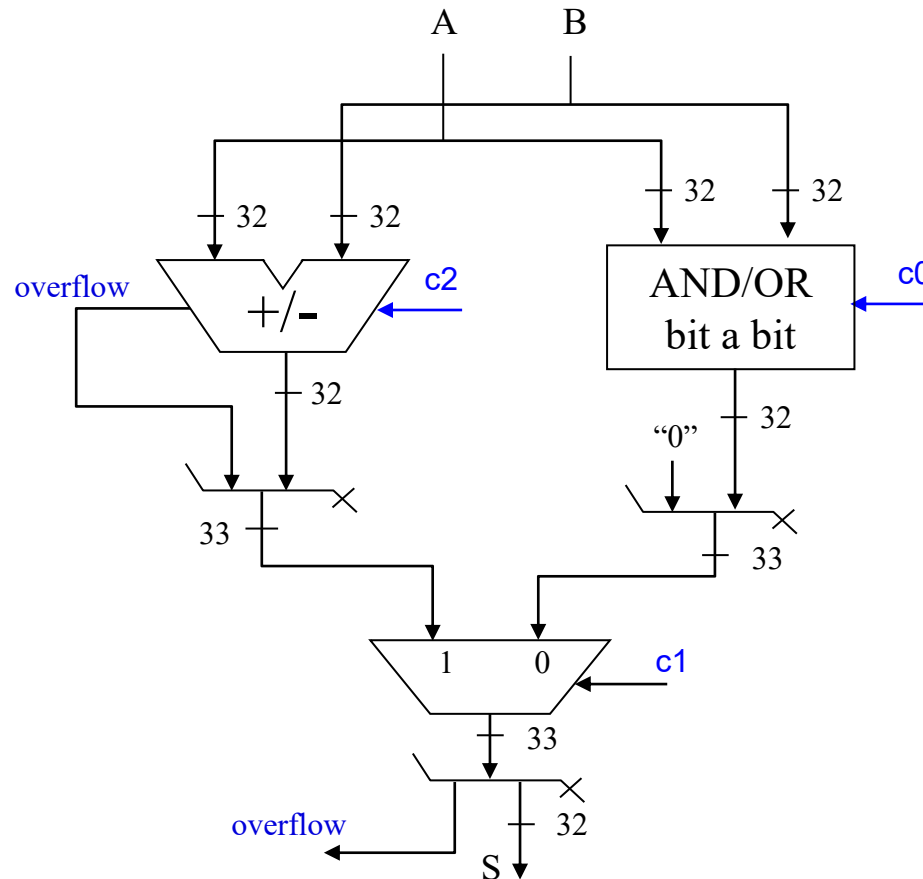


O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$

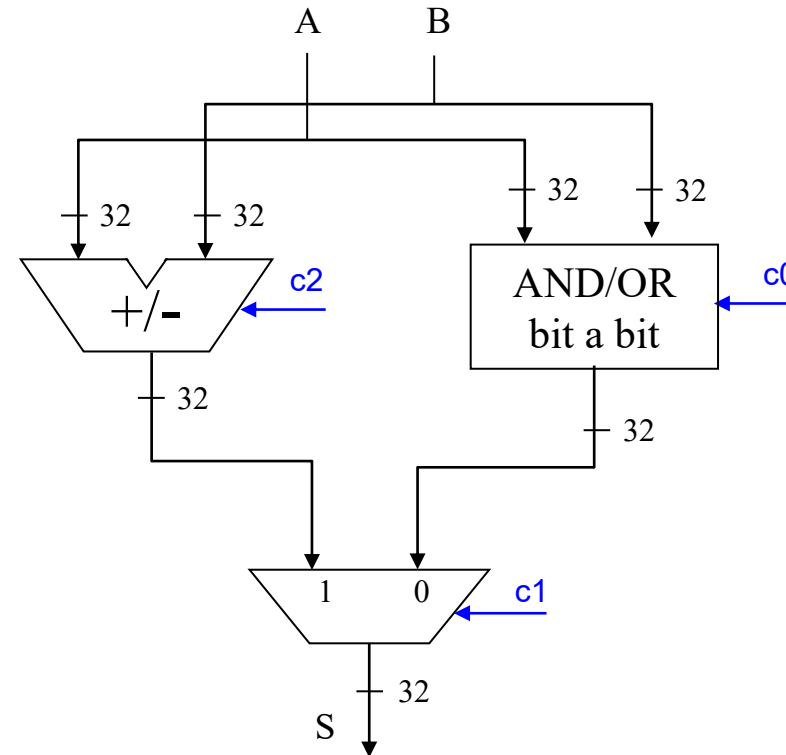
Possível solução para o *overflow*



O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	operação
000	$S = A \text{ AND } B$
001	$S = A \text{ OR } B$
010	$S = A + B$
110	$S = A - B$
111	$S = 1 \text{ se } A < B \text{ senão, } S = 0$



Tarefa de casa:

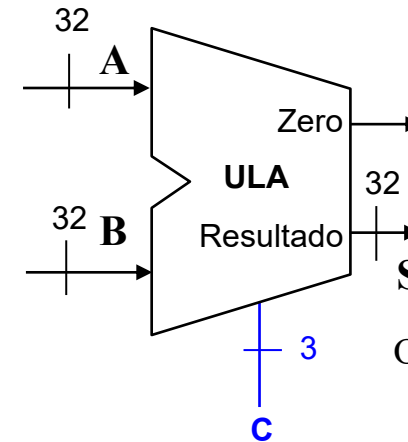
Acrescentar os componentes para realizar a última linha da tabela de operações. (Ver exercício 1 da Lista 3.)

O Processador MIPS Monociclo

A ULA do MIPS

c2 c1 c0	Operação	
000	AND	$S = A \text{ AND } B$
001	OR	$S = A \text{ OR } B$
010	Adição	$S = A + B$
110	Subtração	$S = A - B$
111	Set on less than	$S = 1$ se $A < B$ senão, $S = 0$

Símbolo no nível RT

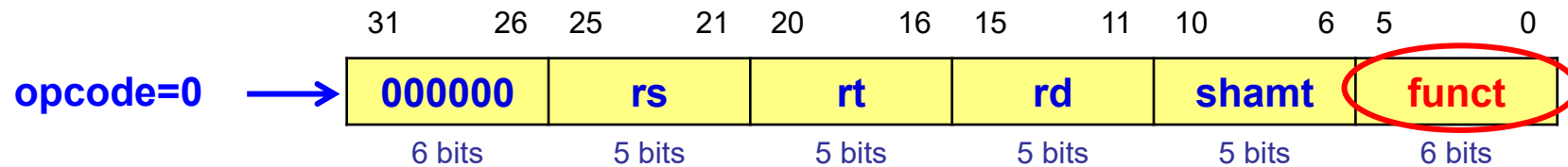


Obs: o sinal de overflow está omitido

O Processador MIPS Monociclo

Uso da ULA do MIPS

- Instruções tipo R: depende do campo “**funct**”, podendo ser AND, OR, adição, subtração ou set on less than

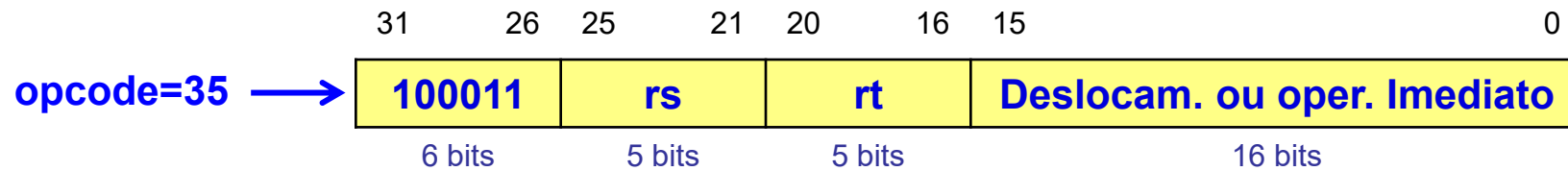


O Processador MIPS Monociclo

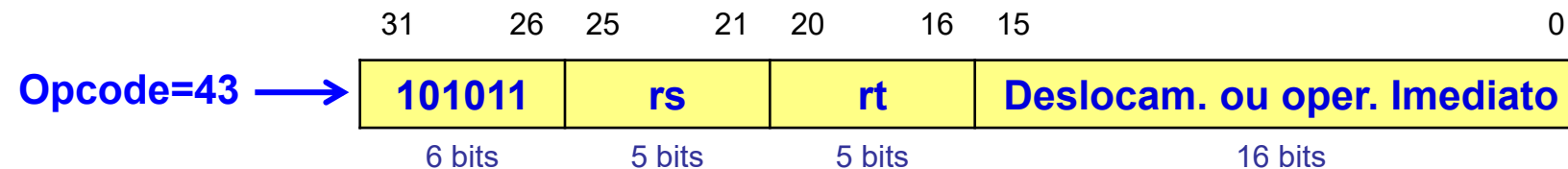
Uso da ULA do MIPS

- **lw/sw: adição (para calcular o endereço de acesso à memória)**

lw \$s1, deslocam(\$s2) ($\$s1 \leftarrow \text{Mem}[\$s2 + \text{deslocam}]$)



sw \$s1, deslocam(\$s2) ($\text{Mem}[\$s2 + \text{deslocam}] \leftarrow \$s1$)

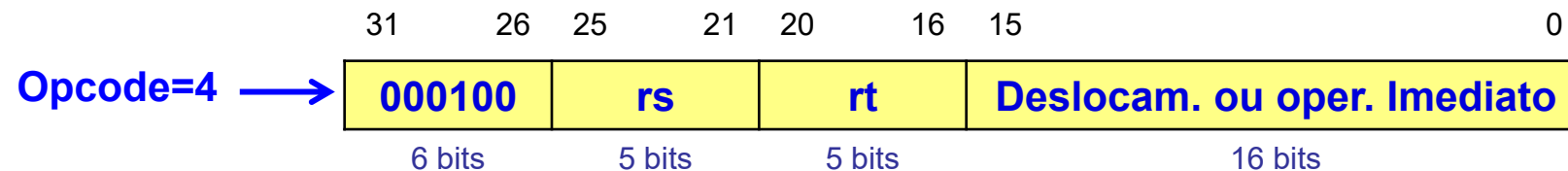


O Processador MIPS Monociclo

Uso da ULA do MIPS

- **beq: subtração** (para testar se os registradores Rs e Rt são iguais)

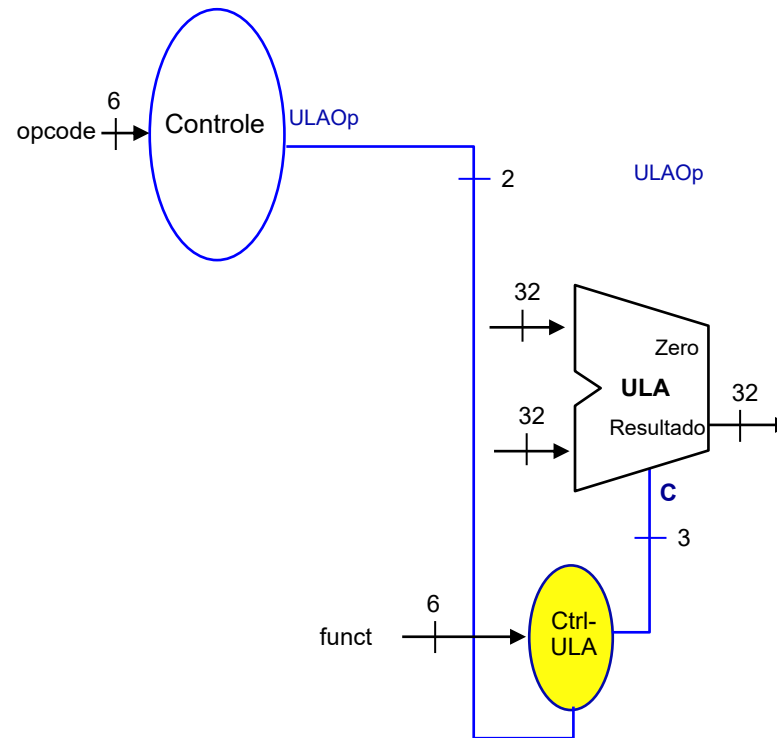
beq \$s1, \$s2, deslocam (if (\$s1==\$s2) then $PC \leftarrow PC + 4 + \text{deslocam}$)



O Processador MIPS Monociclo

O Controle da ULA

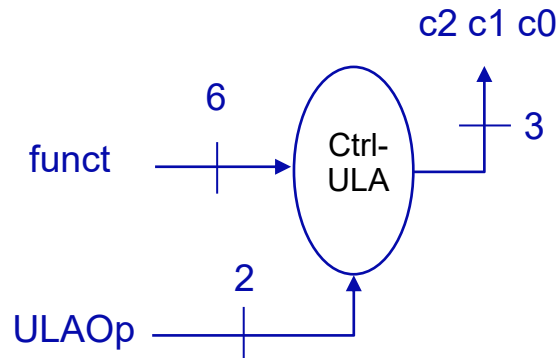
- A ULA terá um controle próprio, separado do bloco de controle principal (e portanto, menos complexo e mais rápido)



O Processador MIPS Monociclo

O Controle da ULA

- Este controle tem como entradas:
 - Os **6 bits** do campo “funct”
 - **2 bits** vindos do bloco de controle global (“ULAOp”), que serve para identificar a qual classe a instrução a ser executada pertence

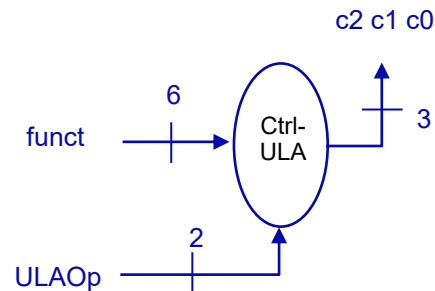


ULAOp	Classe de instrução	ULA deve realizar	c2 c1 c0
00	lw/sw	adição	010
01	beq	subtração	110
10	tipo R	depende do campo “funct”	000 ou 001 ou 010 ou 110 ou 111
11	-	-	-

O Processador MIPS Monociclo

O Controle da ULA

Correspondência entre “funct” & “ULAOp” com “controle da ULA”



Instrução	ULAOp	Campo “funct”	Operação da ULA	c2 c1 c0
lw	00	XXXXXX	adição	010
sw	00	XXXXXX	adição	010
beq	01	XXXXXX	subtração	110
add	10	100000	adição	010
sub	10	100010	subtração	110
and	10	100100	and	000
or	10	100101	or	001
slt	10	101010	set on less than	111

Conclusões:

- Apenas algumas das 64 combinações possíveis a partir dos 6 bits do campo “funct” são de interesse
- O campo “funct” somente interessa quando ULAOp = 10

O Processador MIPS Monociclo

O Controle da ULA

Tabela-Verdade para a Geração dos 3 Bits de Controle da ULA

Instrução	ULAOp	Campo "funct"	Operação da ULA	c2 c1 c0
lw	00	XXXXXX	adição	010
sw	00	XXXXXX	adição	010
beq	01	XXXXXX	subtração	110
add	10	100000	adição	010
sub	10	100010	subtração	110
and	10	100100	and	000
or	10	100101	or	001
slt	10	101010	set on less than	111

ULAOp1	ULAOp2	F5	F4	F3	F2	F1	F0	c2 c1 c0
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

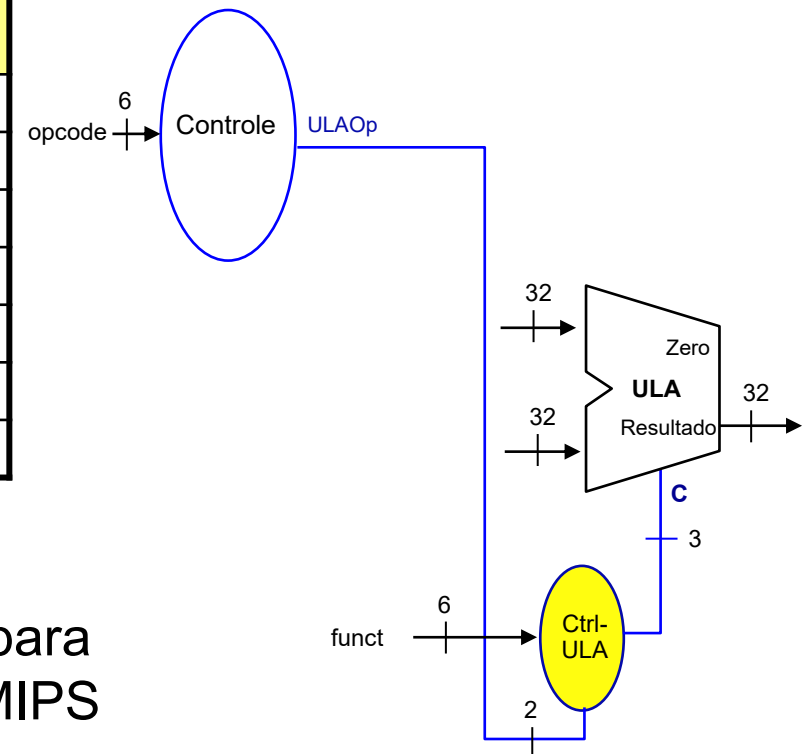
O Processador MIPS Monociclo

O Controle da ULA

Tabela-Verdade para a Geração dos 3 Bits de Controle da ULA

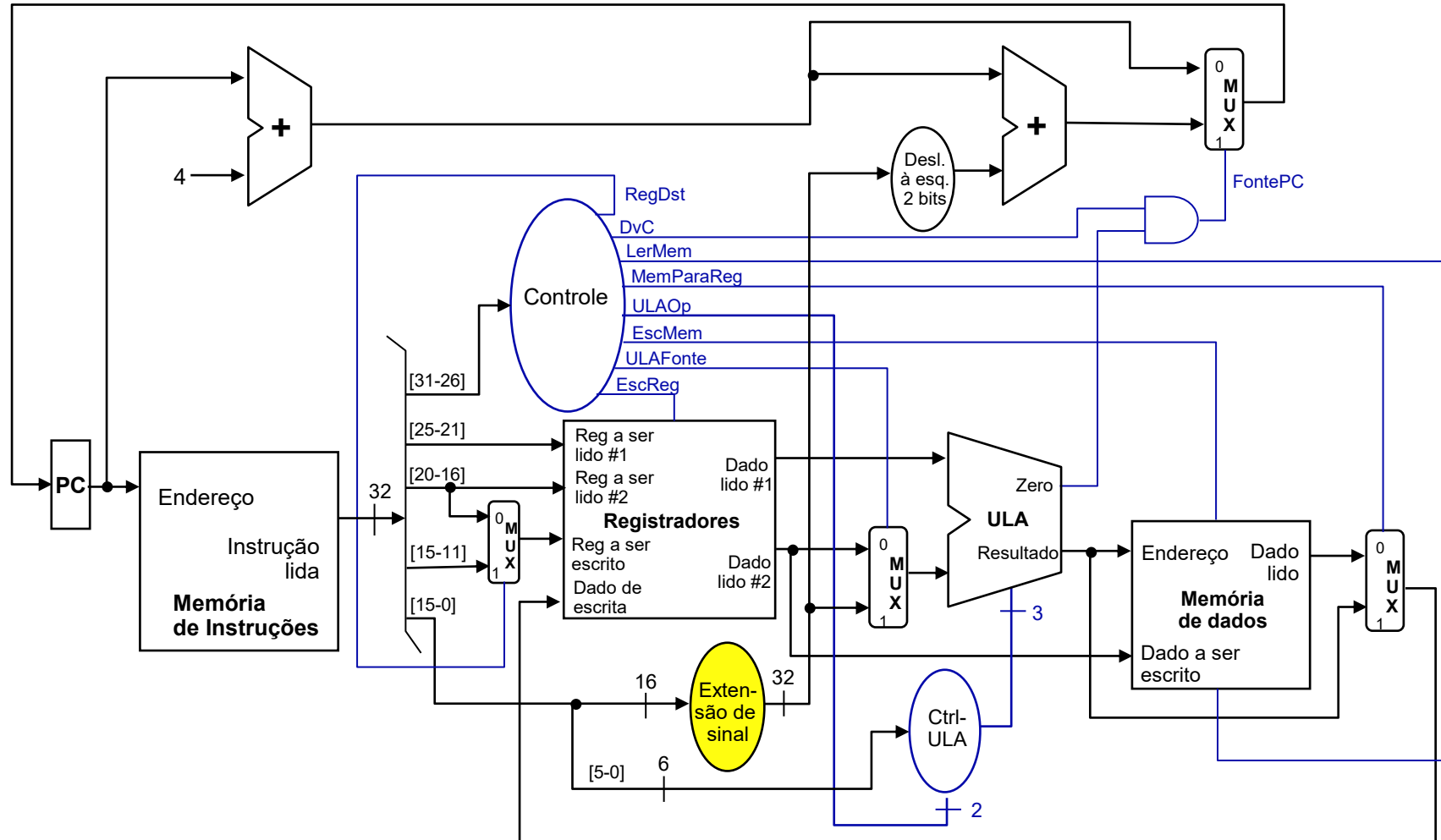
UOp1	UOp2	F5	F4	F3	F2	F1	F0	c2 c1 c0
0	0	X	X	X	X	X	X	010
X	1	X	X	X	X	X	X	110
1	X	X	X	0	0	0	0	010
1	X	X	X	0	0	1	0	110
1	X	X	X	0	1	0	0	000
1	X	X	X	0	1	0	1	001
1	X	X	X	1	0	1	0	111

As equações para c2, c1 e c0 são usadas para construir o circuito de controle da ULA do MIPS



O Processador MIPS Monociclo

Extensão de Sinal



O Processador MIPS Monociclo

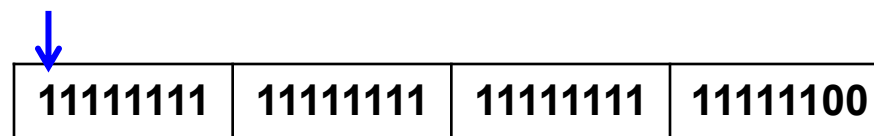
Extensão de Sinal

Como transformar um número de 16 bits (com sinal) em um número de 32 bits (também com sinal)?

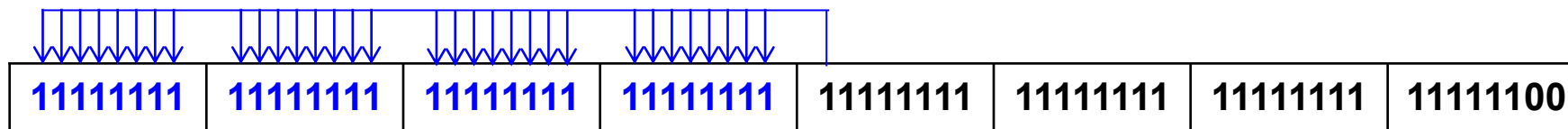
- Resposta: copiando-se seu bit de sinal, de modo a preencher os bits mais significativos que faltam.

Exemplo:

bit de sinal



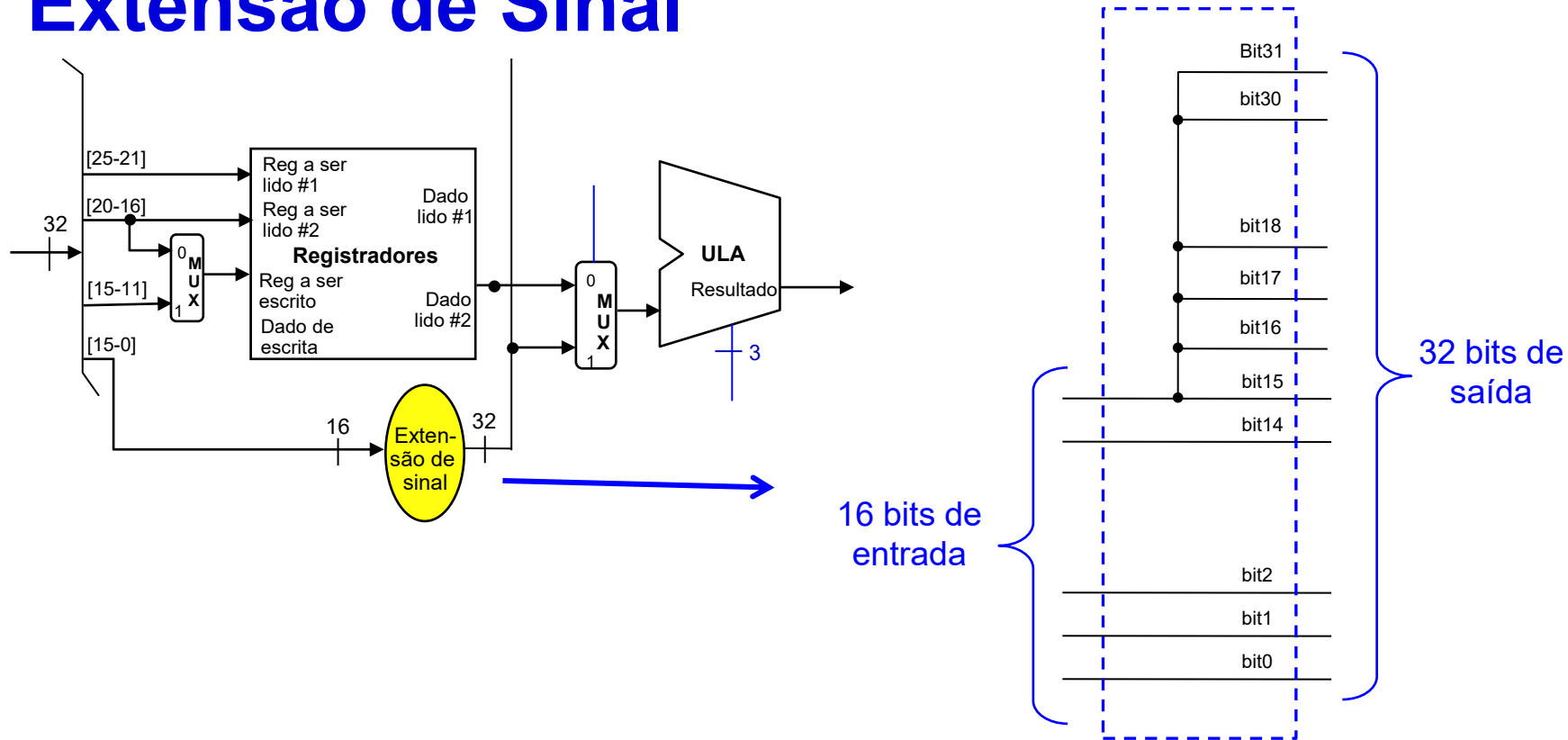
Número original, com 16 bits: (Que número é esse, em decimal?)



Número estendido para 32 bits: (Que número é esse, em decimal?)

O Processador MIPS Monociclo

Extensão de Sinal



Número de transistores = zero
Custo: apenas fios

O Processador MIPS Monociclo

Leituras da Semana - 1

Somadores

VAHID, Frank. Digital Design with RTL Design, VHDL and Verilog. 2nd Edition. Wiley, 2010.

ou

VAHID, Frank. Sistemas Digitais: projeto, otimização e HDLs. Porto Alegre: Bookman, 2008. (ISBN 978-85-7780-190-9)

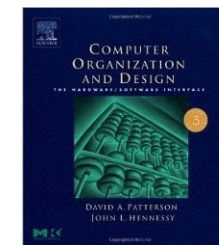
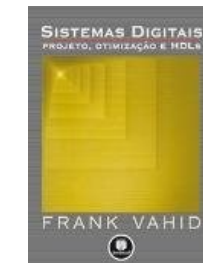
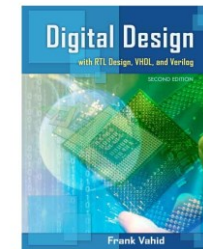
Seção 4.3.

Aritmética com Inteiros e construção da ULA:

PATTERSON, David A.; HENNESSY, John L. "Computer Organization and Design: the hardware/software Interface", 3rd edition, Morgan Kaufmann Publishers, San Francisco, California, USA, 2007.

Se usar a 2ª Edição: Seções 4.1 a 4.5.

Se usar a 3ª Edição: Seções 3.1 a 33 e Apêndice B (no CD)



O Processador MIPS Monociclo

Leituras da Semana - 2

Controle da ULA do MIPS

PATTERSON, David A.; HENNESSY, John L. “Computer Organization and Design: the hardware/software Interface”, 3rd edition, Morgan Kaufmann Publishers, San Francisco, California, USA, 2007.

Se usar a 2^a Edição: Seção 5.1 a 5.2, 5.3 e C.2 (Apêndice C)

Se usar a 3^a Edição: Seções 5.1 a 5.3, 5.4 e C.2 (Apêndice C, no CD)

