



Universidade Federal de Pelotas

Instituto de Física e Matemática

Departamento de Informática

Bacharelado em Ciência da Computação

Arquitetura e Organização de Computadores II

Aula 3

2. MIPS monociclo: controle da ULA, execução das instruções, bloco de controle principal, desempenho de máquinas monociclo.

Prof. José Luís Güntzel

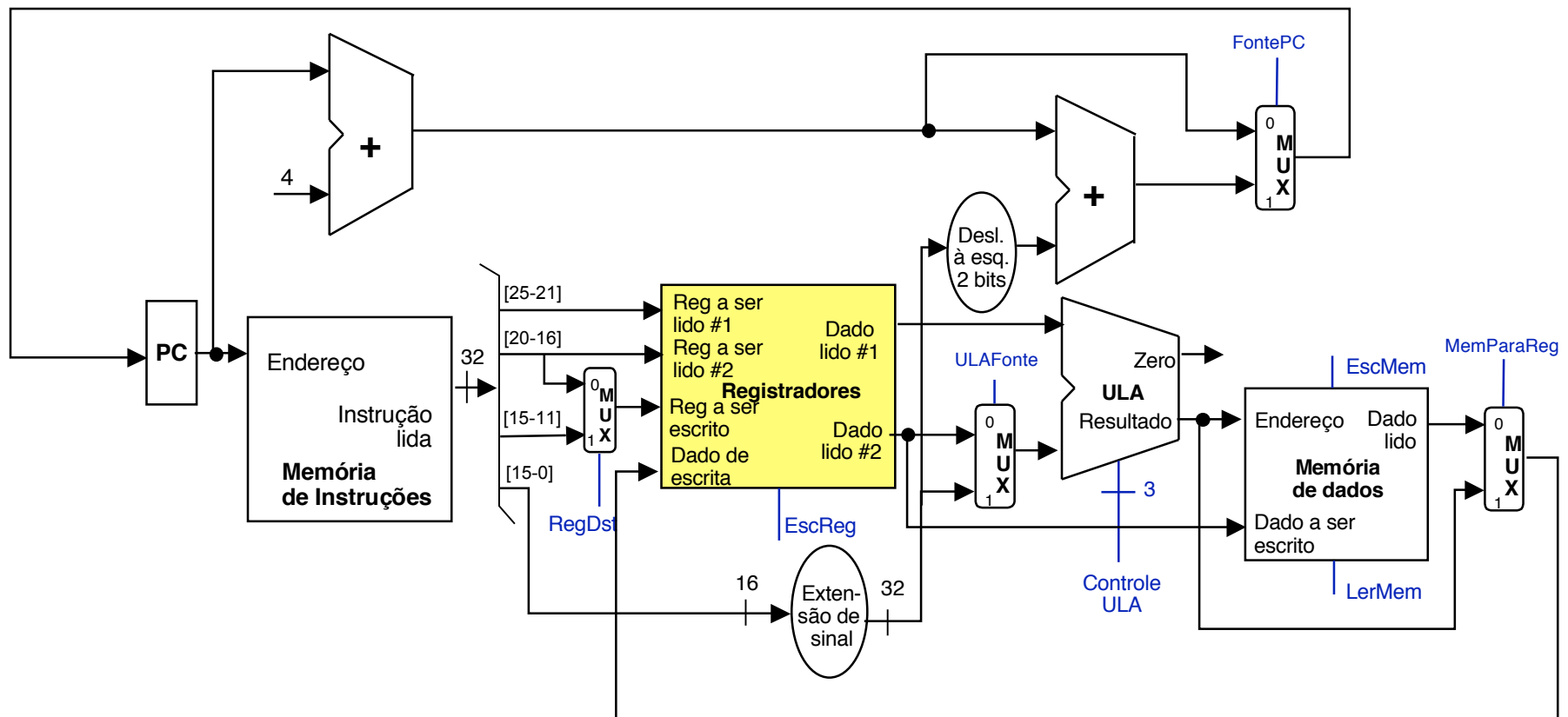
guntzel@ufpel.edu.br

www.ufpel.edu.br/~guntzel/AOC2/AOC2.html

2. Organizações do MIPS: monociclo

► Bloco Operativo Monociclo

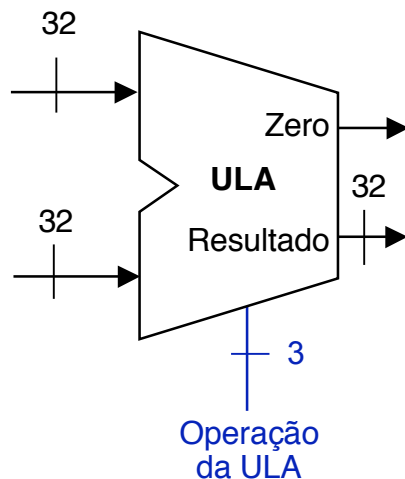
Capaz de Realizar instruções tipo R, lw, sw e beq



2. Organizações do MIPS: monociclo

► O Controle da ULA

- A ULA tem 3 entradas de controle (8 combinações possíveis)
- Somente 5 combinações são usadas

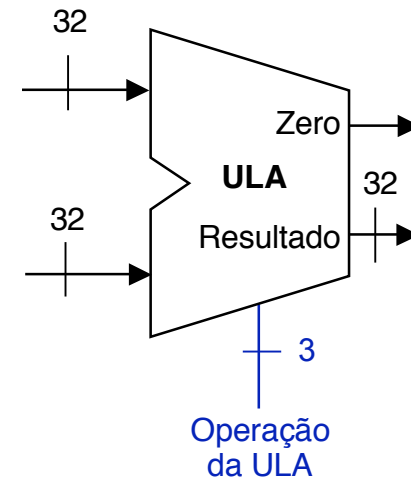


Entrada de controle da ULA	Função
000	AND
001	OR
010	Adição
110	Subtração
111	Set on less than

2. Organizações do MIPS: monociclo

► O Controle da ULA

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



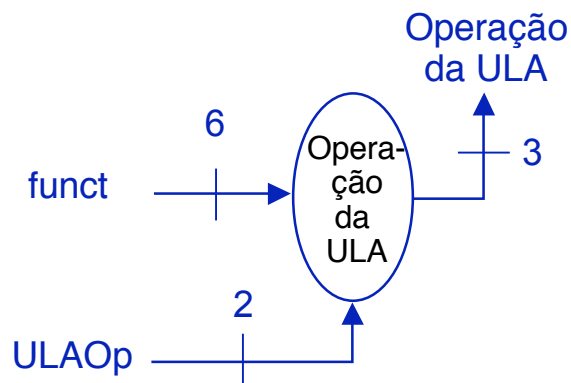
Uso da ULA

- **lw/sw: adição** (para calcular o endereço de acesso à memória)
- **beq: subtração** (para testar se os registradores Rs e Rt são iguais)
- **Instruções tipo R: depende do campo “funct”, podendo ser AND, OR, adição, subtração ou set on less than**

2. Organizações do 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)
- 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

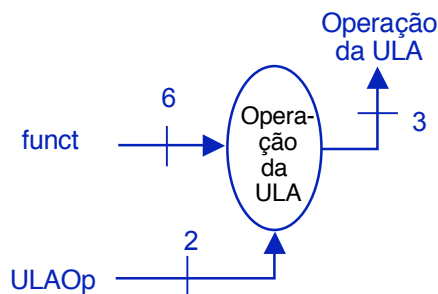


ULAOp	Classe de instrução	ULA deve realizar
00	lw/sw	adição
01	beq	subtração
10	tipo R	depende do campo “funct”
11	-	-

2. Organizações do MIPS: monociclo

► O Controle da ULA

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



Instrução	ULAOp	Campo “funct”	Operação da ULA	Operação da ULA
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

2. Organizações do 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	Operação da ULA
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	Operação da ULA
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

2. Organizações do MIPS: monociclo

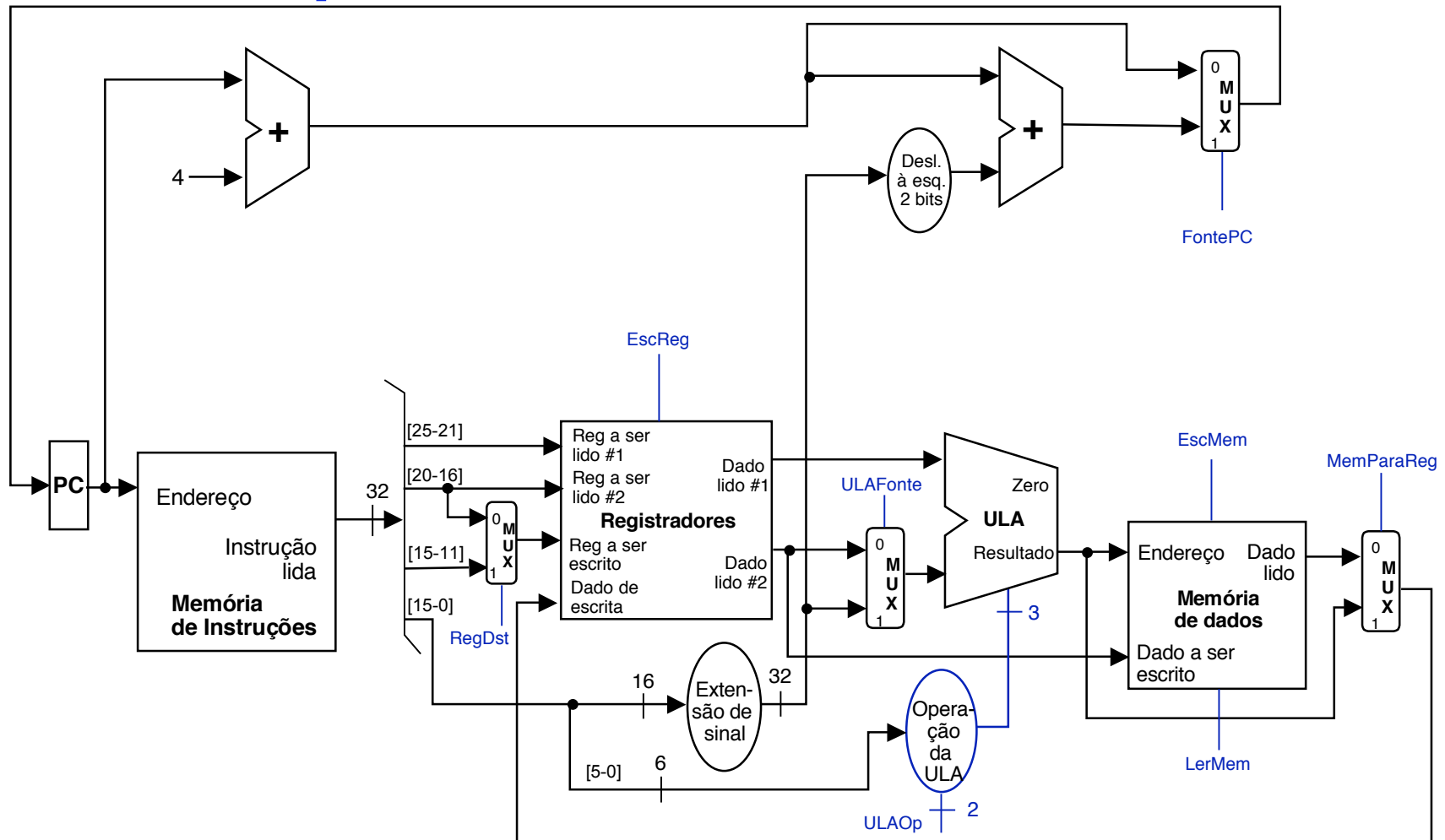
► O Controle da ULA

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

ULAOp1	ULAOp2	F5	F4	F3	F2	F1	F0	Operação da ULA
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

2. Organizações do MIPS: monociclo

► Bloco Operativo + Controle da ULA

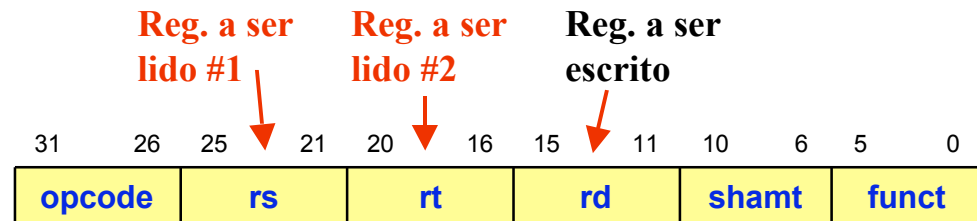


2. Organizações do MIPS: monociclo

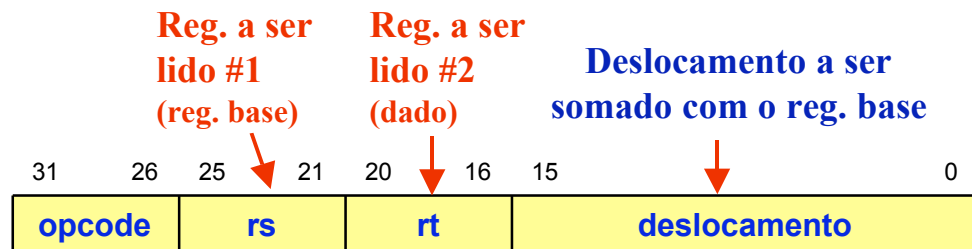
► O Bloco de Controle Principal

Relembrando os formatos das 3 principais classes de instrução

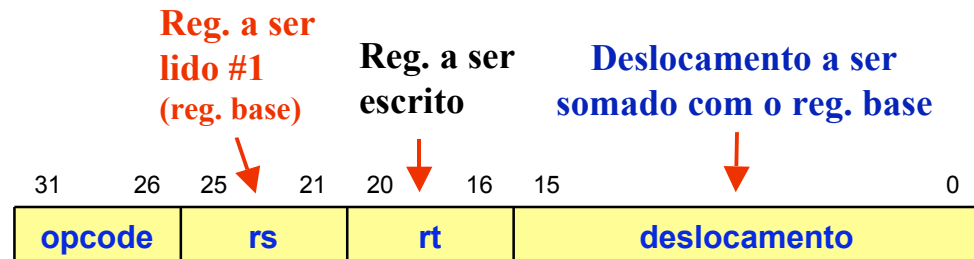
Instrução tipo R:



Instrução store word :



Instrução load word:

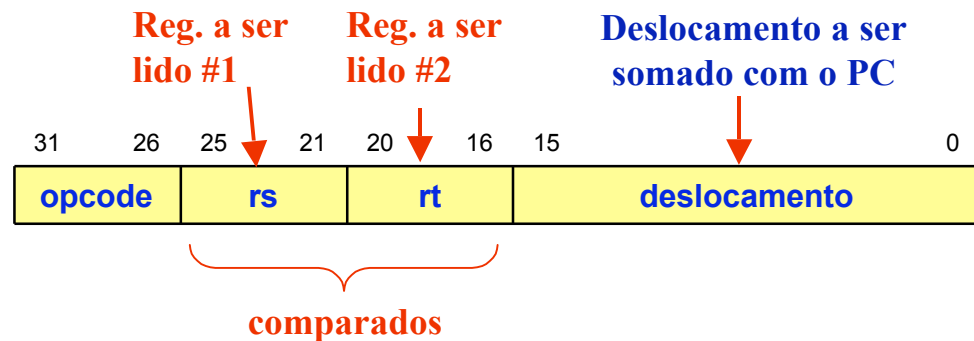


2. Organizações do MIPS: monociclo

► O Bloco de Controle Principal

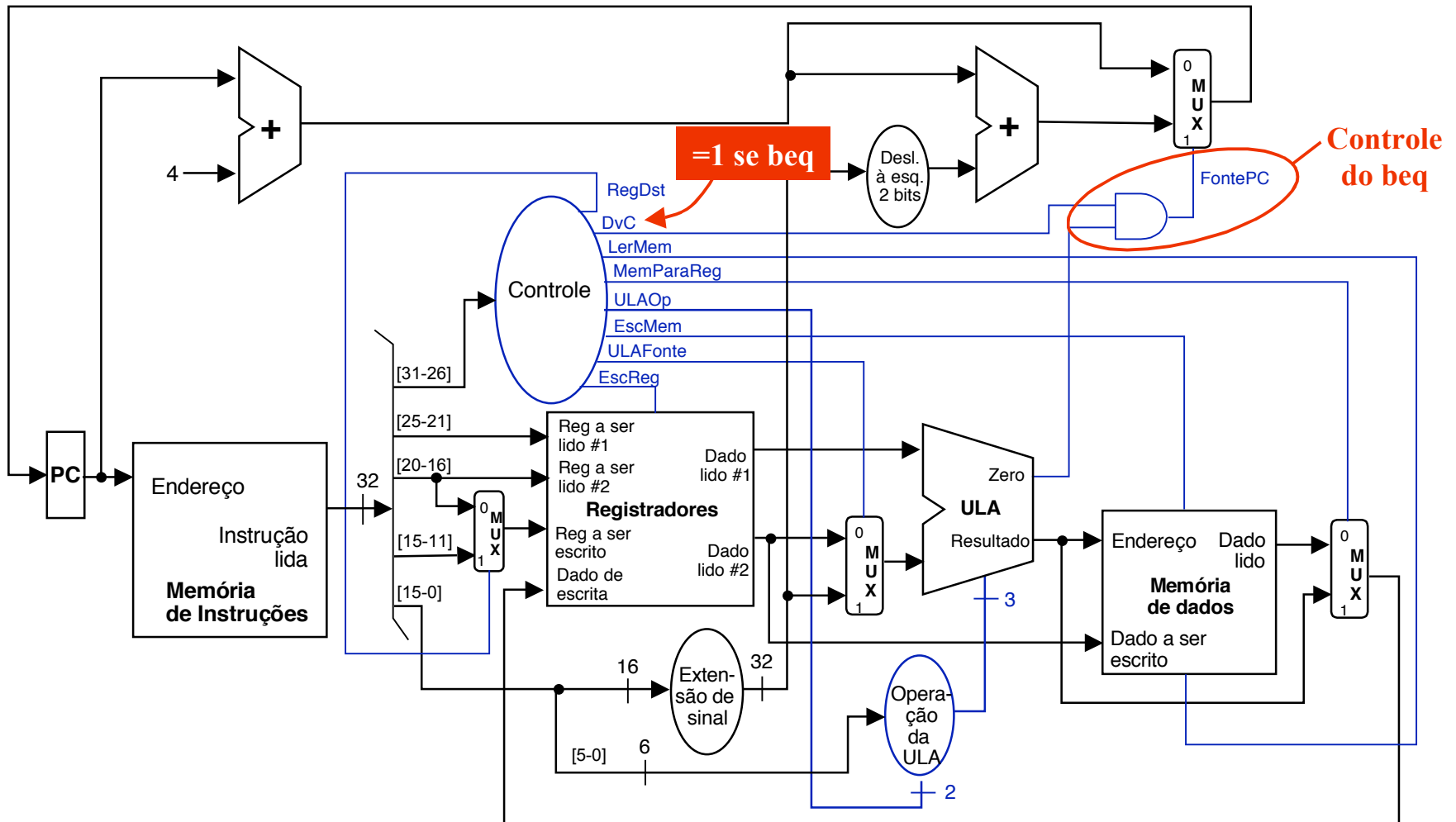
Relembrando os formatos das 3 principais classes de instrução

Instrução branch on equal:



2. Organizações do MIPS: monociclo

► Os Sinais de Controle



2. Organizações do MIPS: monociclo

► Execução de uma Instrução Tipo R

Seja uma instrução tipo R, como por exemplo `add $t1, $t2, $t3`:

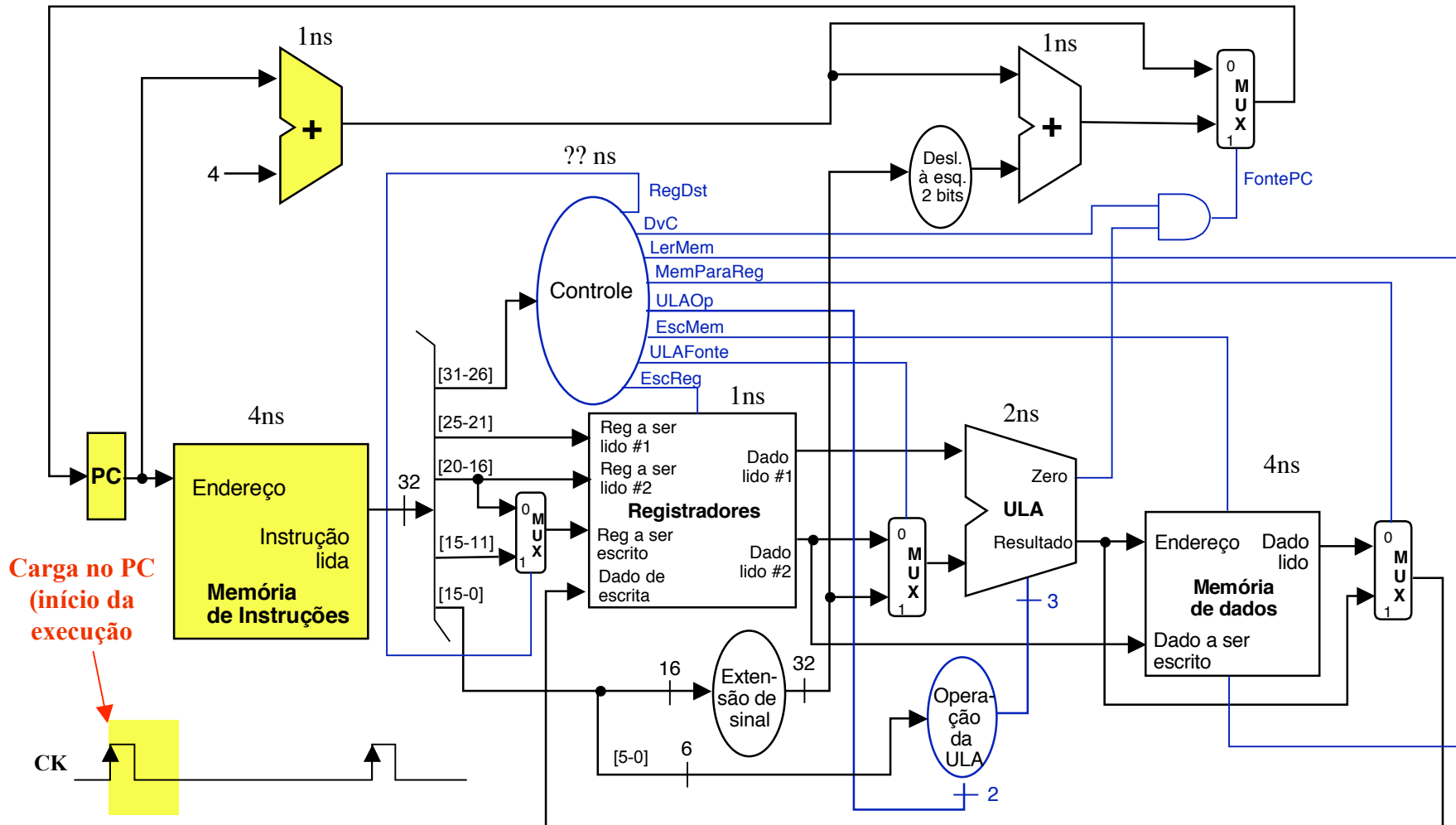
Podemos imaginar que esta instrução é executada em 4 etapas:

1. Busca da instrução (na memória de instruções) e incremento do PC
2. Leitura de dois registradores (no caso, `$t2` e `$t3`, ou `Rs` e `Rt`) e geração dos sinais de controle para o resto do bloco operativo (decodificação da instrução)
3. Operação na ULA
4. Escrita (do resultado da operação realizada na ULA) no registrador destino (`$t1` ou `Rd`)

Como estes passos ocorrem dentro do mesmo ciclo de relógio (regime monociclo), a ordem real irá depender do atraso de cada componente.

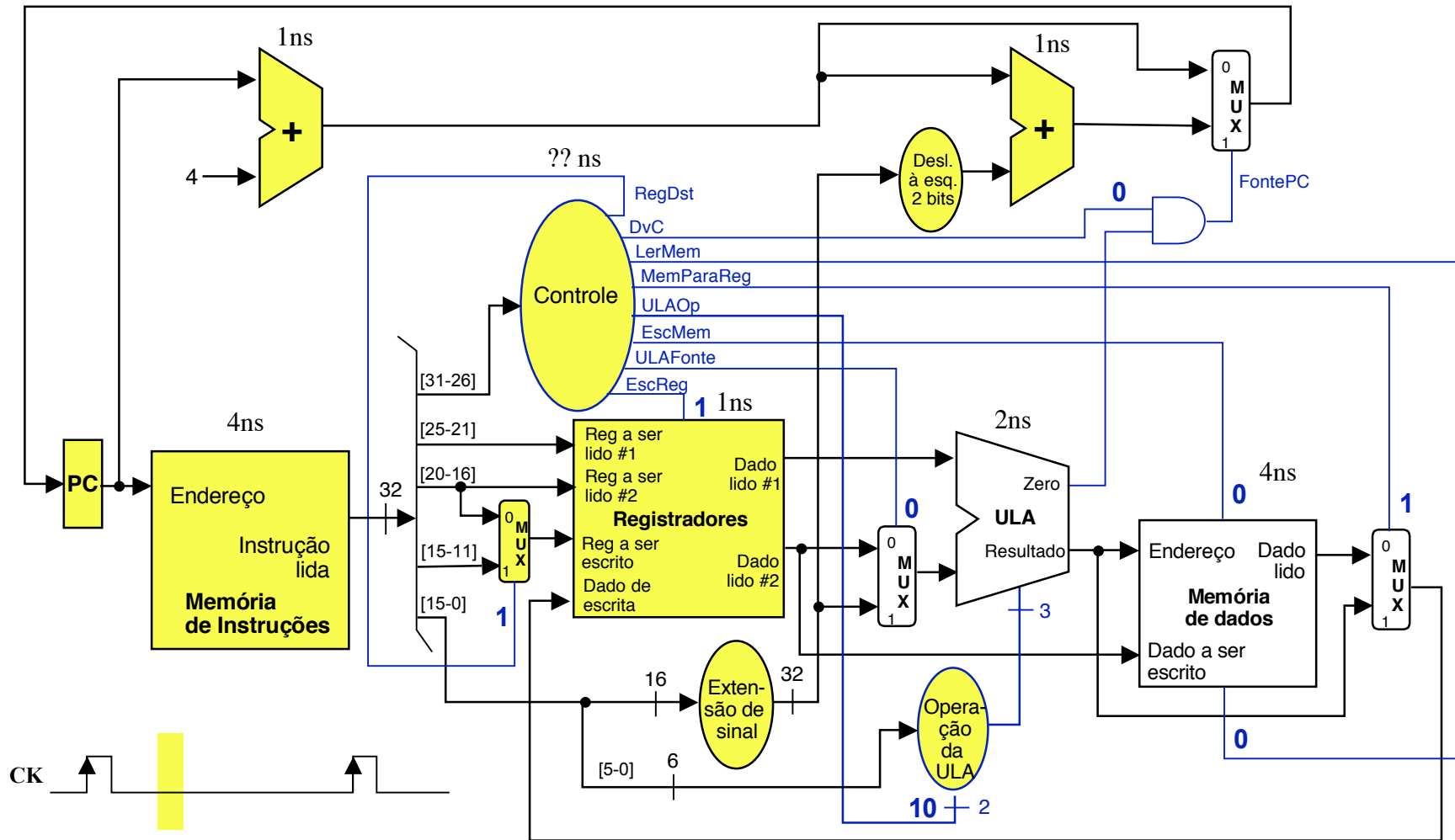
2. Organizações do MIPS: monociclo

► Instrução Tipo R: busca da instrução e cálculo de PC+4



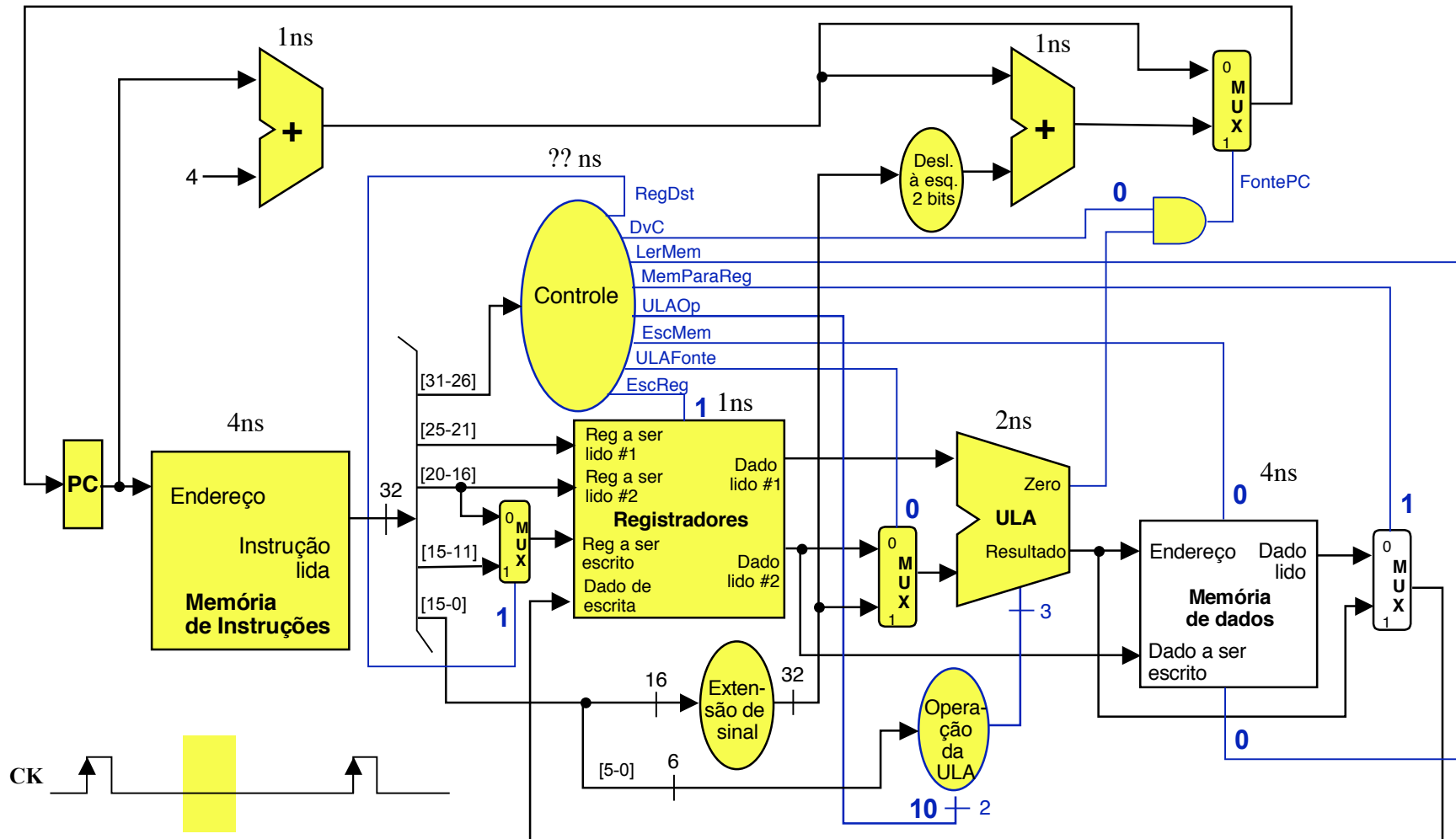
2. Organizações do MIPS: monociclo

► Instrução Tipo R: leit. de Rs e Rt e geração sinais de controle



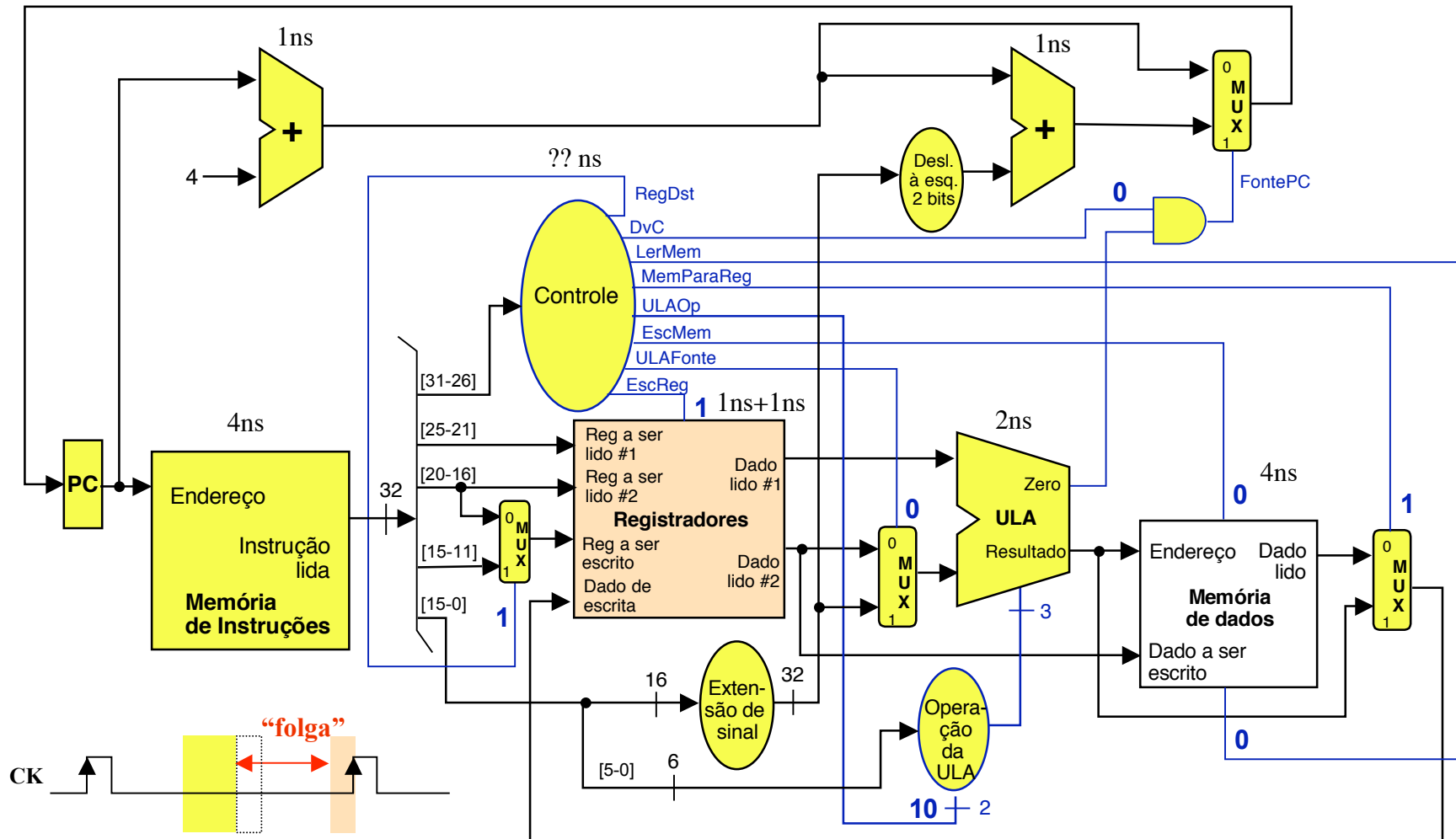
2. Organizações do MIPS: monociclo

► Instrução Tipo R: operação na ULA (depende de “funct”)



2. Organizações do MIPS: monociclo

► Instrução Tipo R: escrita no registrador-destino

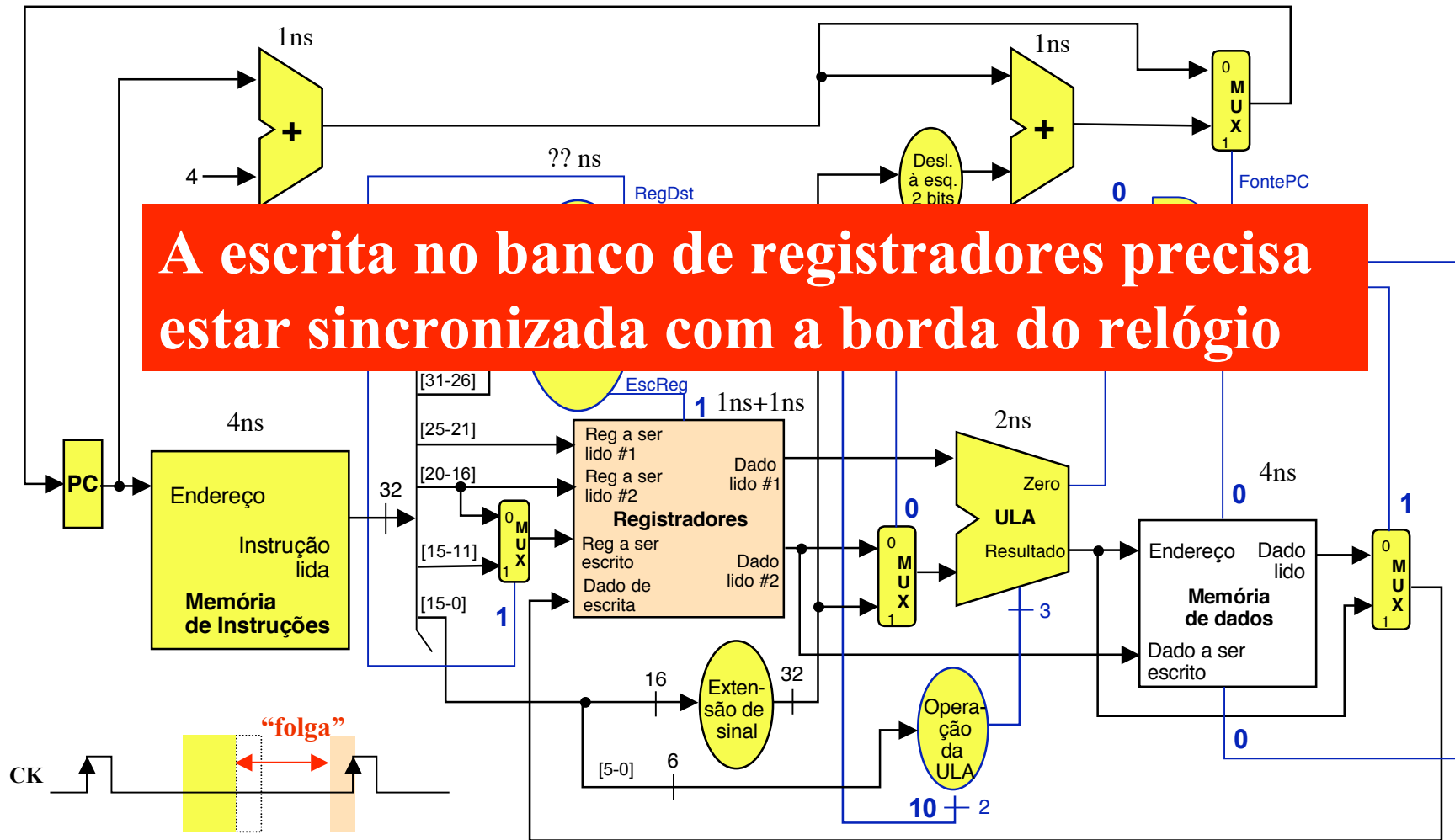


slide 3.17

Prof. José Luís Güntzel

2. Organizações do MIPS: monociclo

► Instrução Tipo R: escrita no registrador-destino



2. Organizações do MIPS: monociclo

► Execução de uma Instrução lw

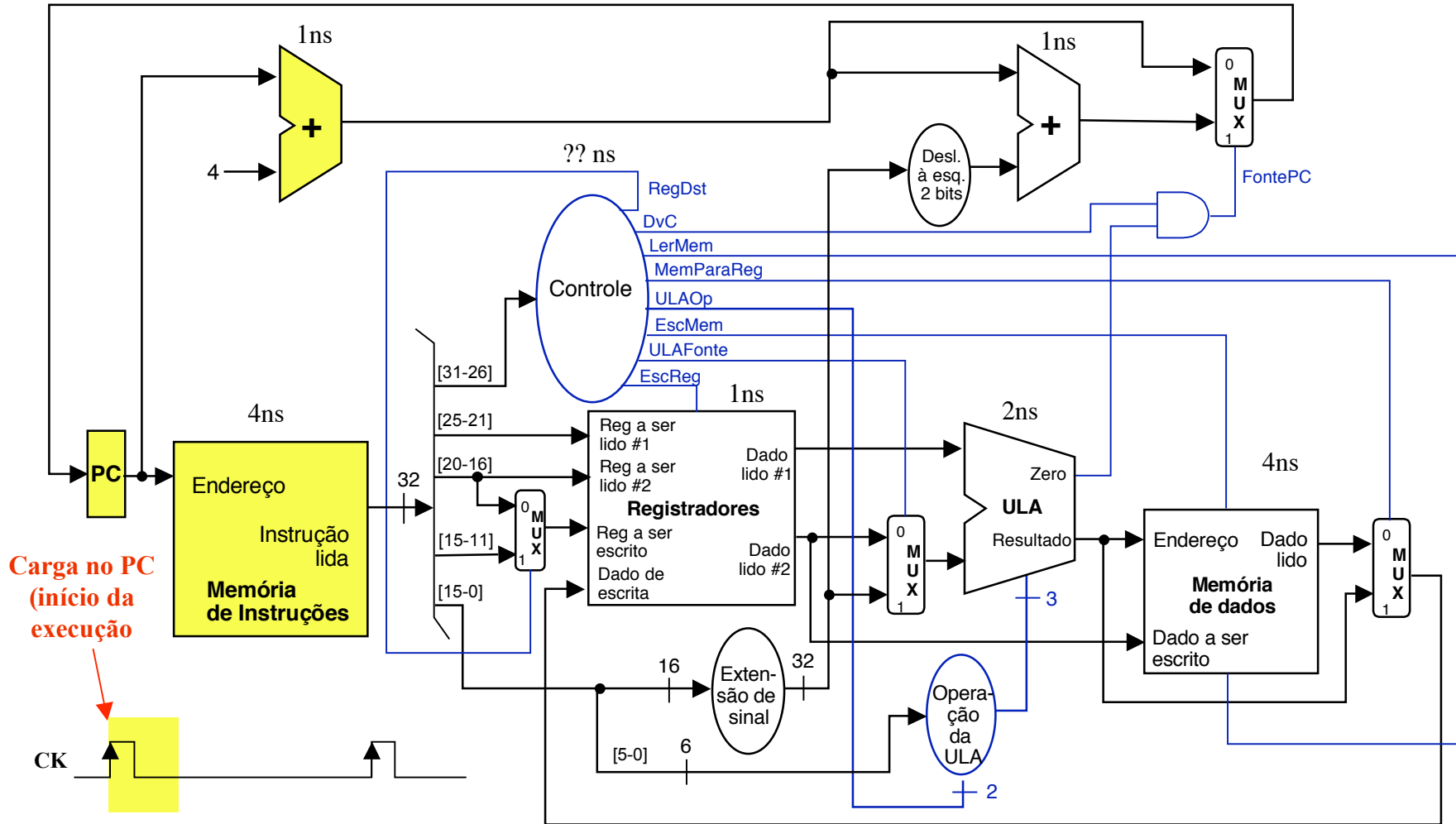
Seja a instrução **load word**, lw \$t1, deslocamento(\$t2):

Podemos imaginar que esta instrução é executada em 5 etapas:

1. Busca da instrução (na memória de instruções) e incremento do PC
2. Leitura de dois registradores (no caso, \$t1 e \$t2, ou Rs e Rt) e geração dos sinais de controle para o resto do bloco operativo (decodificação da instrução). **Apenas o registrador \$t2 (Rs) interessa, pois é o registrador-base. Rt será desprezado...**
3. Cálculo do endereço usando a ULA (adição)
4. Acesso à memória de dados para uma leitura (endereço = resultado da ULA)
5. Escrita (do valor lido da memória de dados) no registrador destino (\$t1, que neste caso corresponde ao campo Rt)

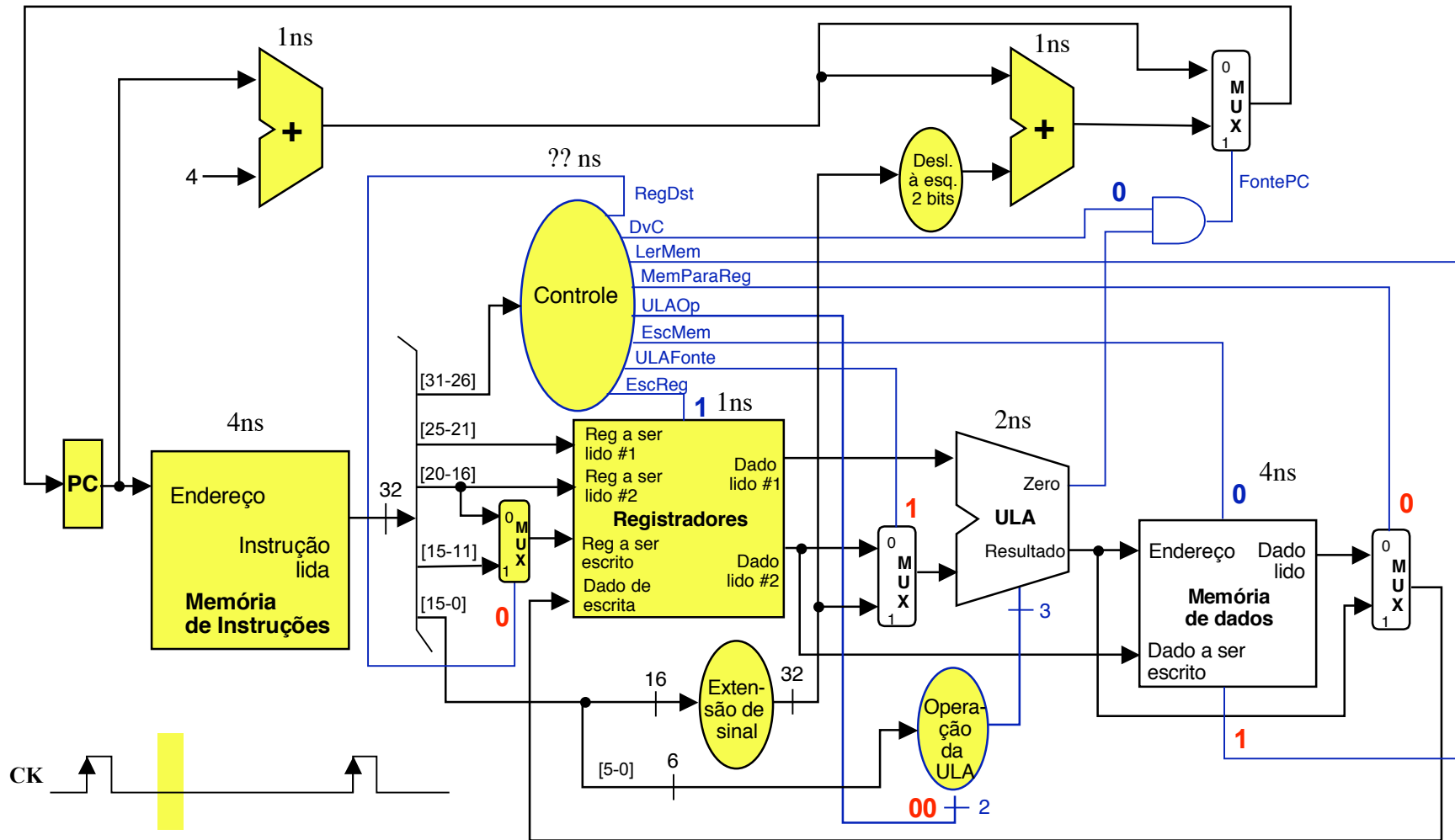
2. Organizações do MIPS: monociclo

► **Instrução lw:** busca da instrução e cálculo de PC+4



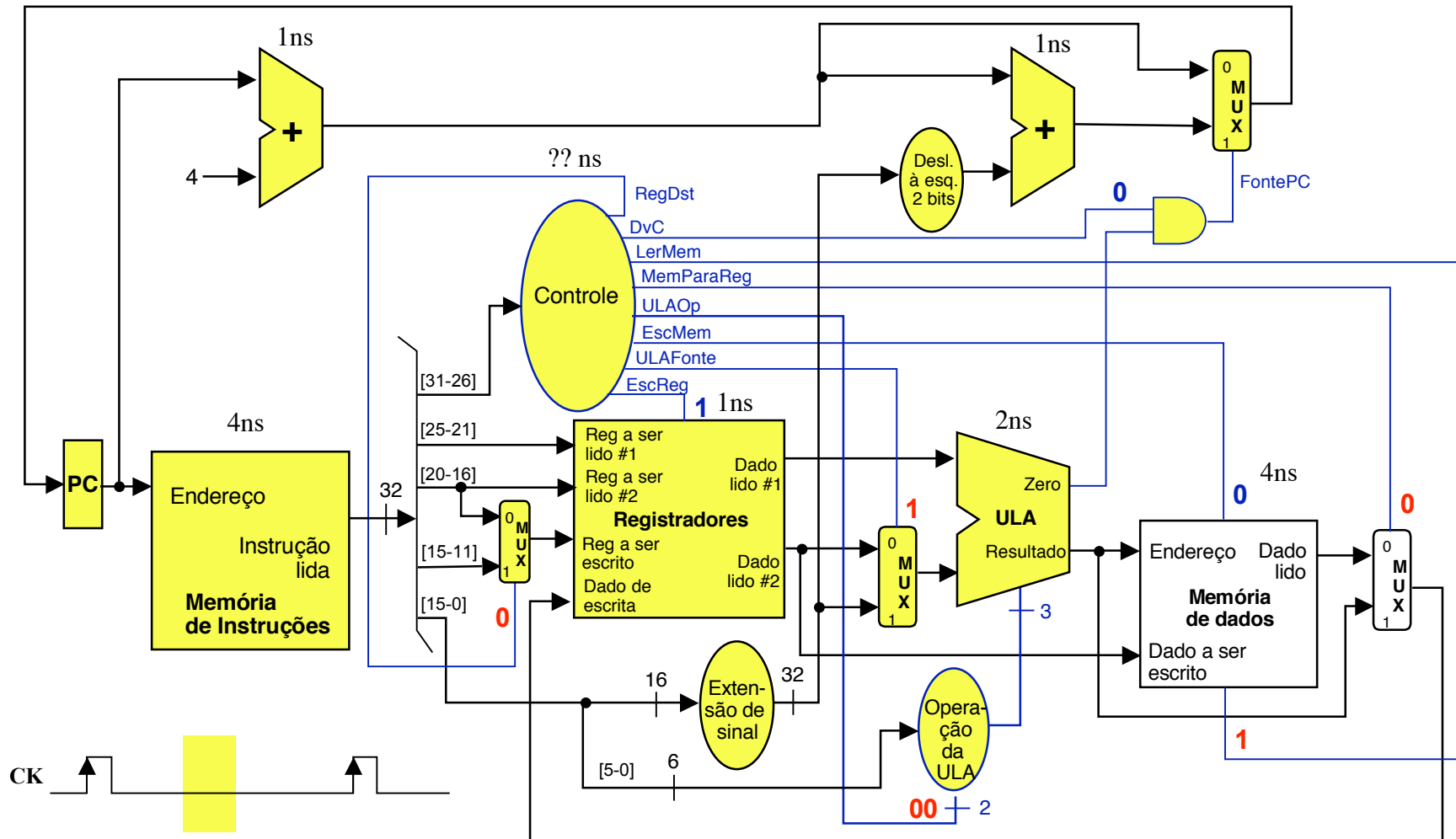
2. Organizações do MIPS: monociclo

► Instrução lw: leitura de Rs (e Rt) e geração sinais de controle



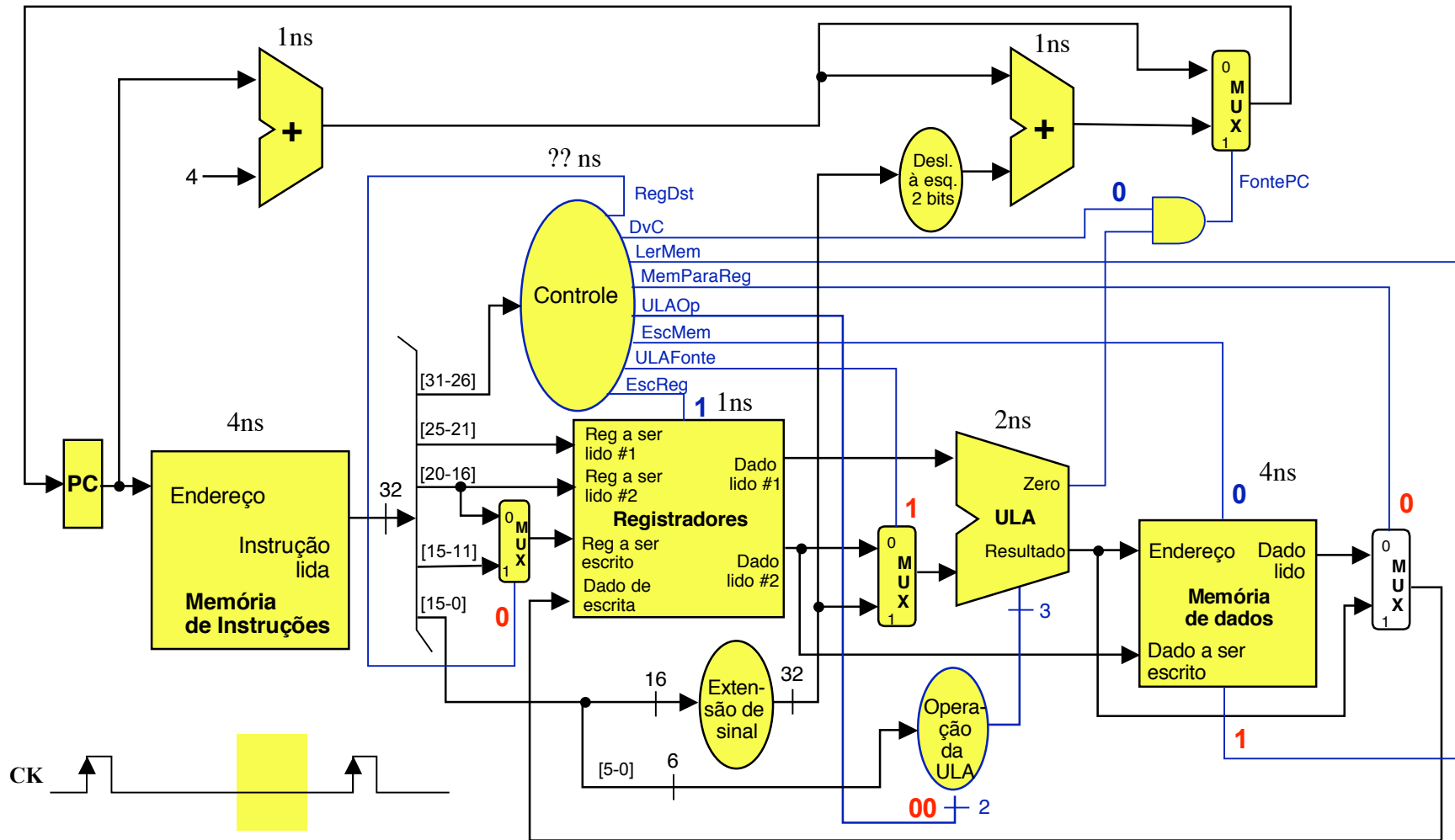
2. Organizações do MIPS: monociclo

► Instrução lw: cálculo do endereço usando a ULA (adição)



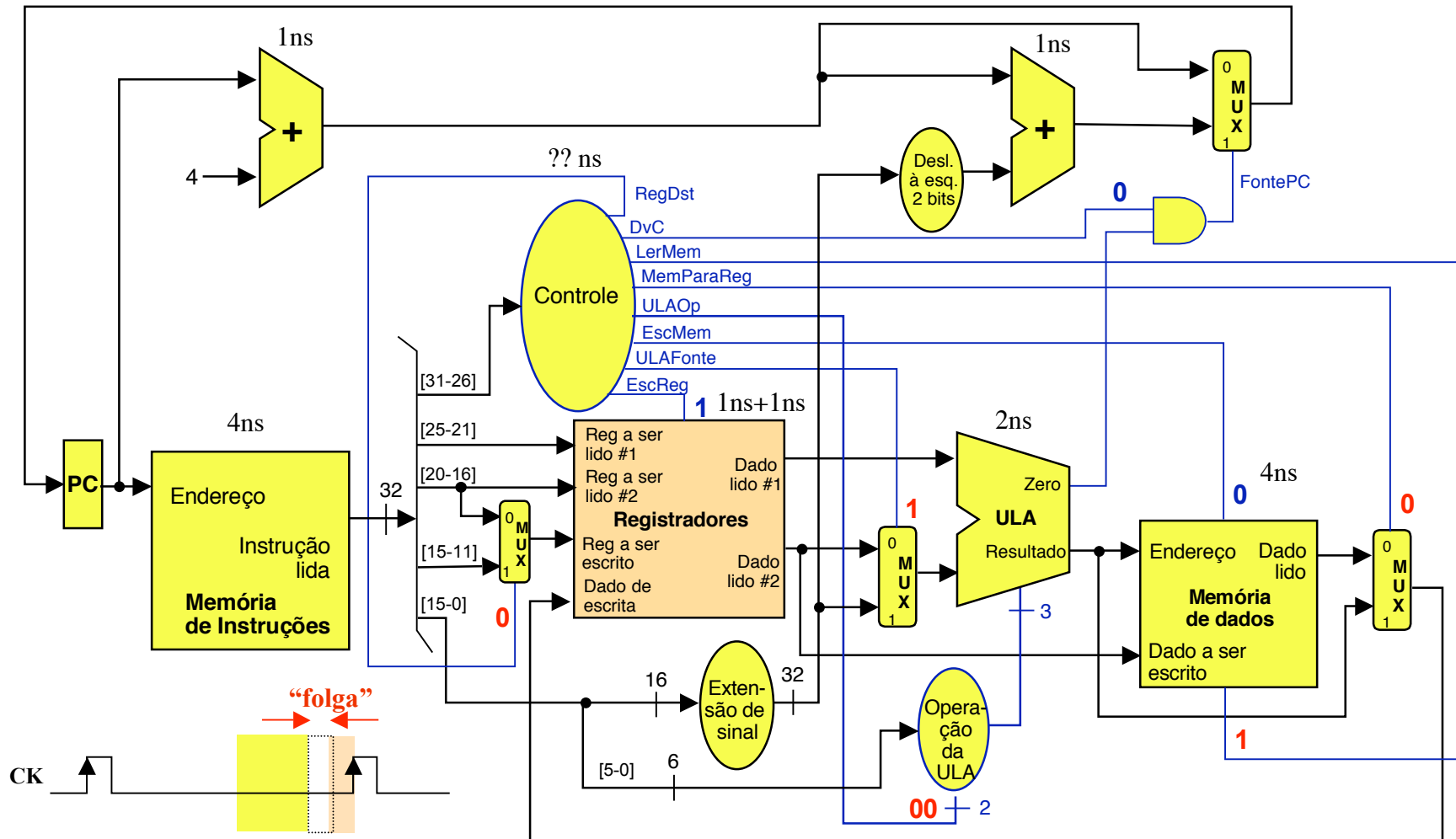
2. Organizações do MIPS: monociclo

► Instrução lw: acesso à memória de dados para uma leitura



2. Organizações do MIPS: monociclo

► Instrução lw: escrita no registrador-destino



2. Organizações do MIPS: monociclo

► Execução de uma Instrução beq

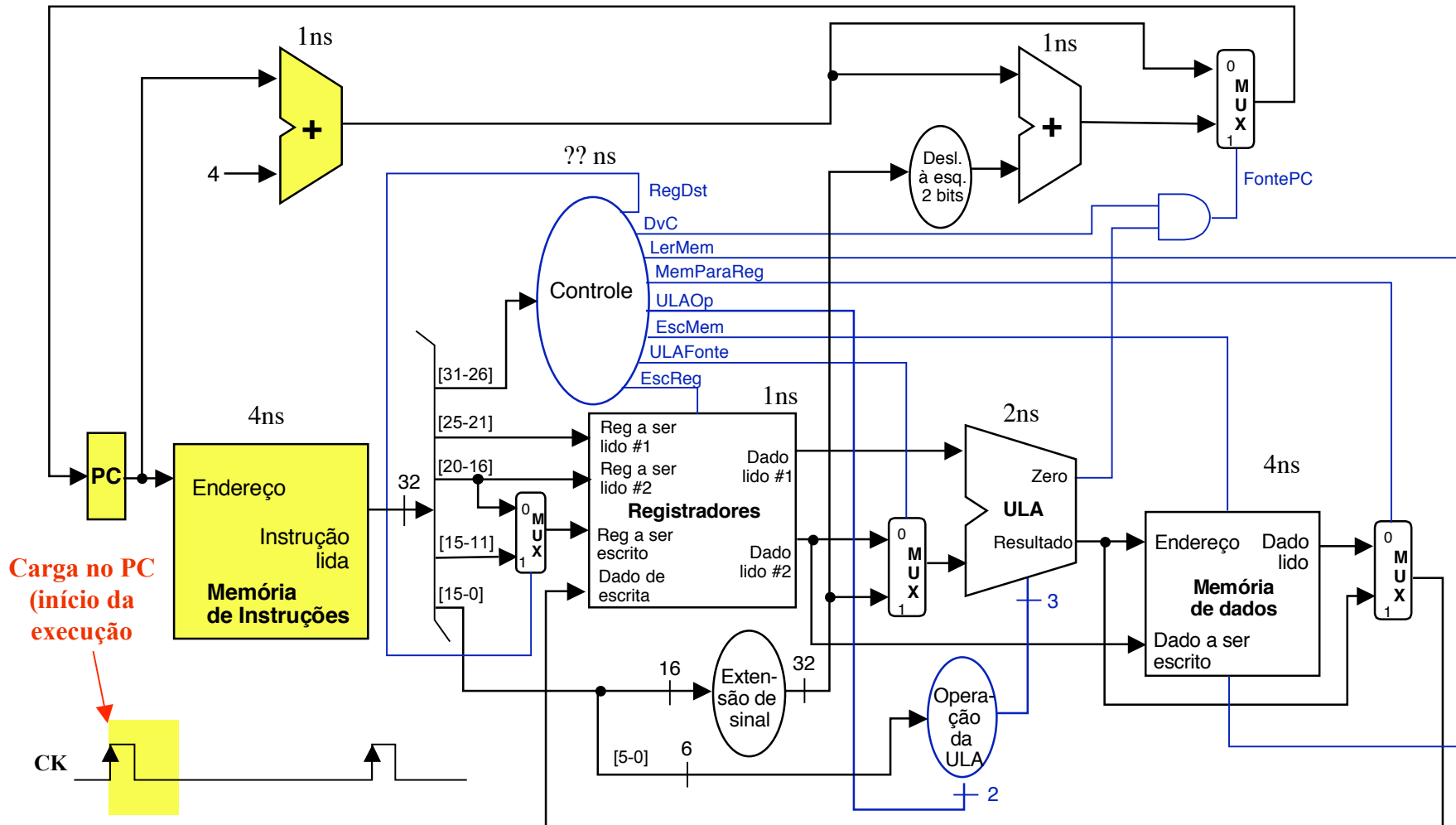
Seja a instrução de desvio condicional, beq \$t1, \$t2, deslocamento:

Podemos imaginar que esta instrução é executada em 3 etapas:

1. Busca da instrução (na memória de instruções) e incremento do PC
2. Leitura de dois registradores (no caso, \$t1 e \$t2, ou Rs e Rt) e geração dos sinais de controle para o resto do bloco operativo (decodificação da instrução)
3. Cálculo da subtração entre os registradores \$t1 e \$t2 (Rs e Rt) usando a ULA (adição). **O valor na saída “zero” da ULA é usado para decidir se o PC será atualizado com PC+4 ou com o endereço-alvo do desvio condicional.**

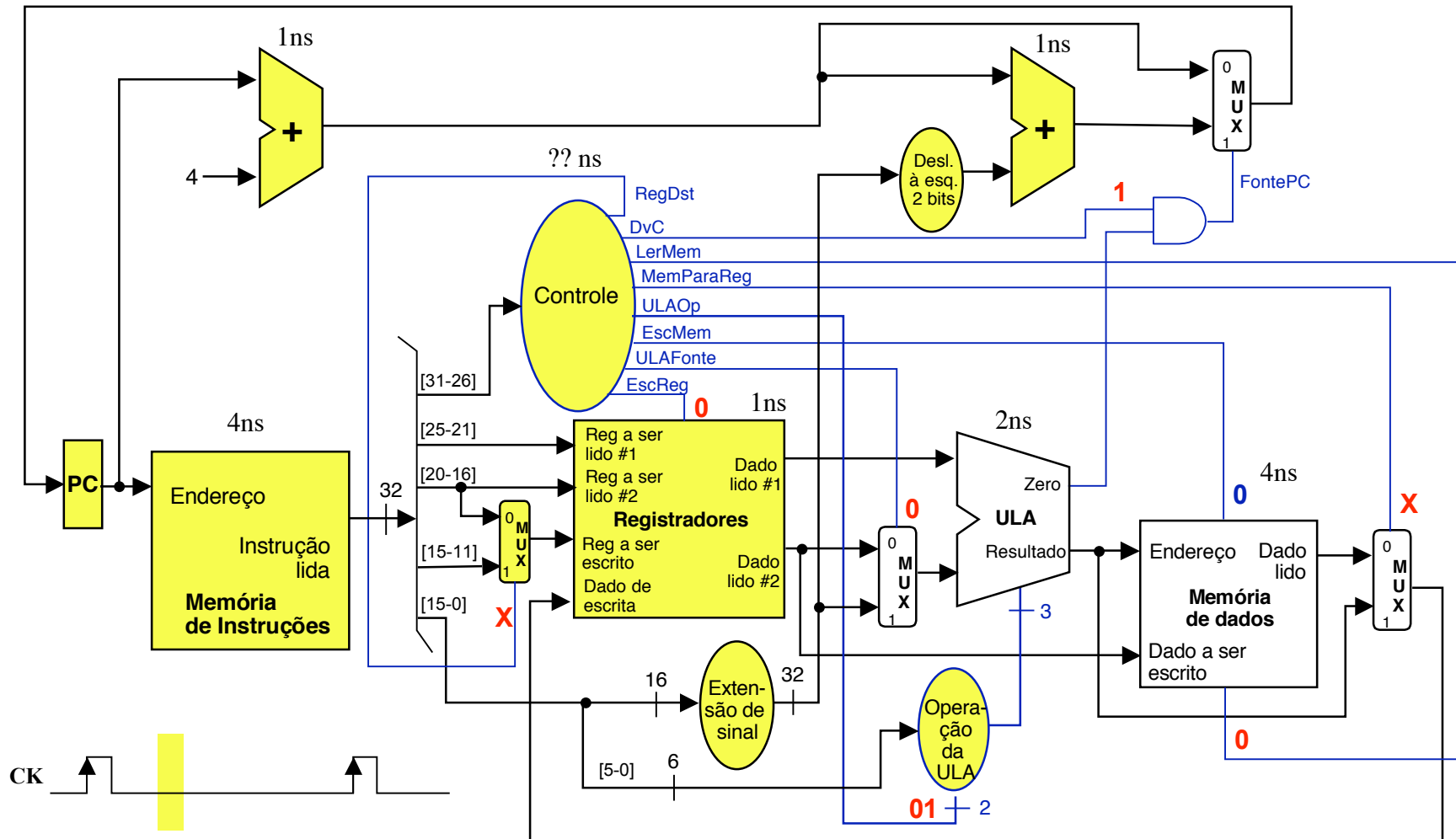
2. Organizações do MIPS: monociclo

► Instrução beq: busca da instrução e cálculo de PC+4



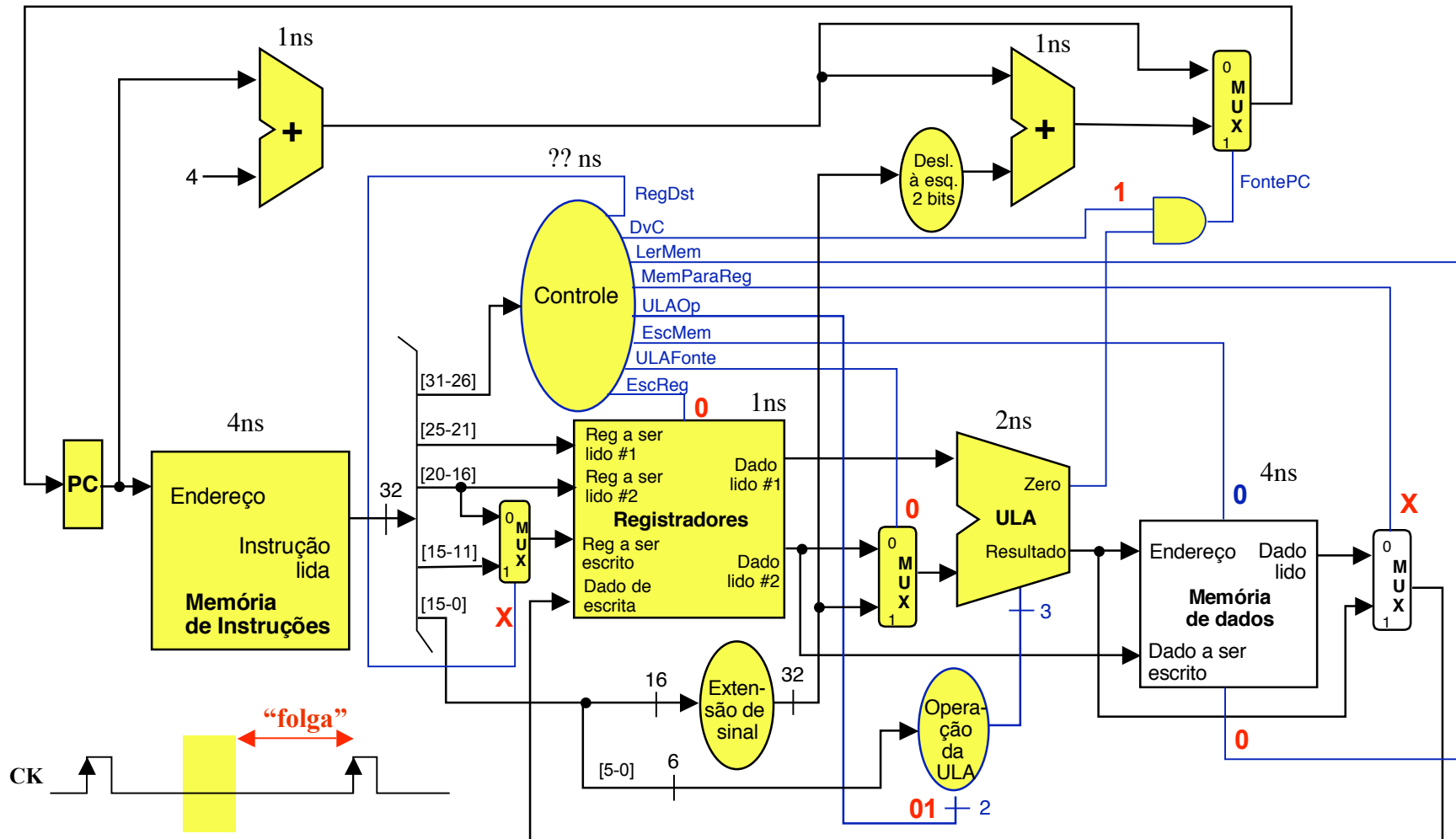
2. Organizações do MIPS: monociclo

► Instrução beq: leit. de Rs e Rt e geração sinais de controle



2. Organizações do MIPS: monociclo

► Instrução beq: operação na ULA (subtração)



2. Organizações do MIPS: monociclo

► Projeto do Bloco de Controle Principal

Valores dos Sinais de Controle para as Instruções R, lw, sw e beq

instrução	RegDst	ULAFonte	MemParaReg	EscReg	LerMem	EscMem	DvC	ULAOp1	ULAOp0
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

Valores do campo “opcode” para as Instruções R, lw, sw e beq

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

2. Organizações do MIPS: monociclo

► Projeto do Bloco de Controle Principal

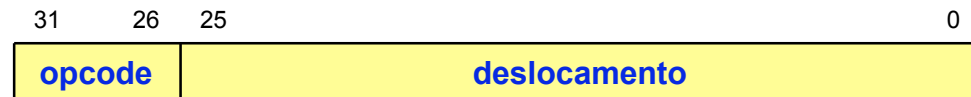
Tabela-Verdade para os Sinais de Controle

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

2. Organizações do MIPS: monociclo

► Inclusão do Desvio Incondicional (jump)

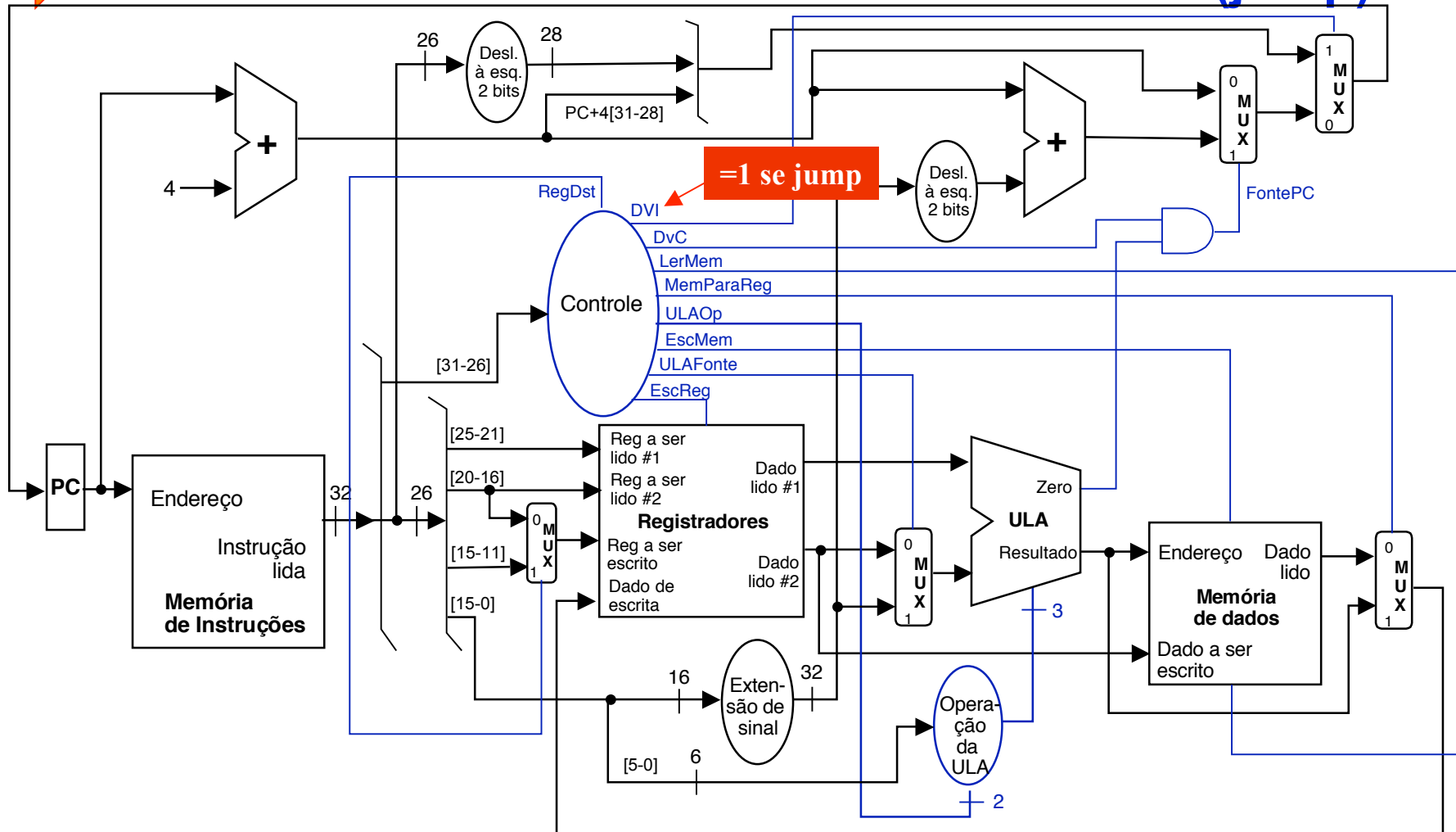
Instrução jump:



- A instrução **jump** se parece com a instrução **beq**
- Porém, calcula o endereço-alvo concatenando:
 - Os 4 bits de mais significativos de **PC+4** com
 - Os 26 bits do campo “deslocamento” da instrução de **jump** (campo imediato) e com
 - os bits “00” (menos significativos)

2. Organizações do MIPS: monociclo

► Inclusão do Desvio Incondicional (jump)



2. Organizações do MIPS: monociclo

► Desempenho de Máquinas Monociclo

Unidades funcionais utilizadas por cada instrução

instrução	Etapa 1	Etapa 2	Etapa3	Etapa 4	Etapa 5
Tipo R	Busca da instrução	Lê registrador(es)	ULA	Escreve registrador	
lw	Busca da instrução	Lê registrador(es)	ULA	Lê memória	Escreve registrador
sw	Busca da instrução	Lê registrador(es)	ULA	Escreve na memória	
beq	Busca da instrução	Lê registrador(es)	ULA		
jump	Busca da instrução				

2. Organizações do MIPS: monociclo

► Desempenho de Máquinas Monociclo

Tempo de execução de cada instrução (com valores hipotéticos de atraso para cada etapa)

instrução	Acesso à memória de instruções	Leitura de registradores	Operação na ULA	Acesso à memória de dados	Escrita no registrador	Total
Tipo R	4 ns	1 ns	2 ns	---	1 ns	8 ns
lw	4 ns	1 ns	2 ns	4 ns	1 ns	12 ns
sw	4 ns	1 ns	2 ns	4 ns	---	11 ns
beq	4 ns	1 ns	2 ns	---	---	7 ns
jump	4 ns	---	---	---	---	4 ns