



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



# Sistemas Digitais

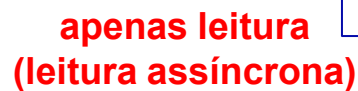
INE 5406

## Aula 9-T

**3. O Processador MIPS monociclo: execução das instruções e análise de *timing***  
**(tempos de estabilização dos sinais, atrasos dos caminhos, caminhos críticos e frequência máxima do relógio).**

**Profs. José Luís Güntzel, Rafael Luiz Cancian e Cristina Meinhardt**  
**{j.guntzel, rafael.cancian, cristina.meinhardt}@ufsc.br**

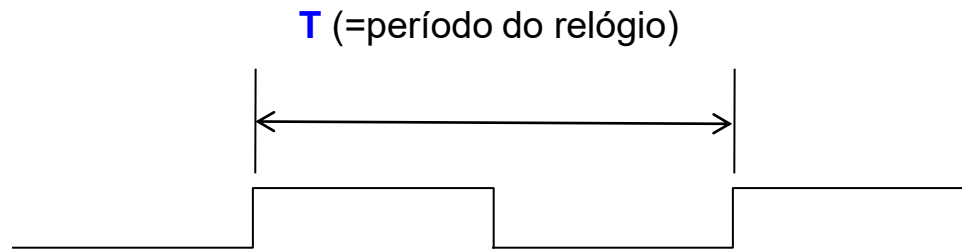
## Elementos de Memória com Capacidade de Escrita



# O Processador MIPS Monociclo

## Estimativa do Período do Relógio

Uma instrução por ciclo de relógio...

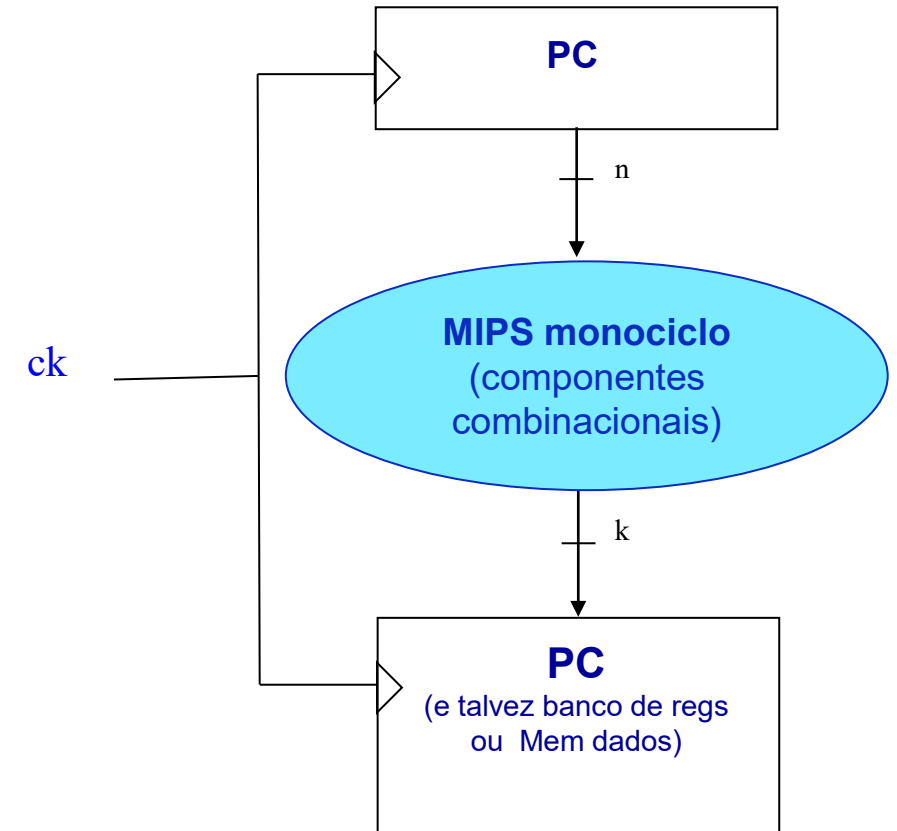


Para que o MIPS monociclo tenha tempo suficiente para realizar qualquer instrução, é necessário que:

$$T \geq D$$

Onde  $D$  = atraso crítico, da instrução mais lenta.

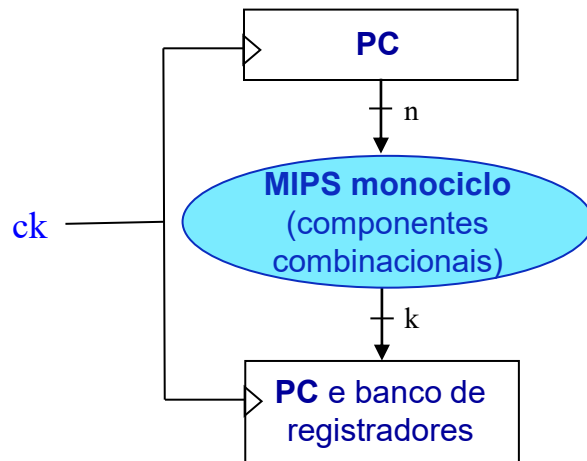
Será preciso realizar a análise de *timing* cada uma das instruções



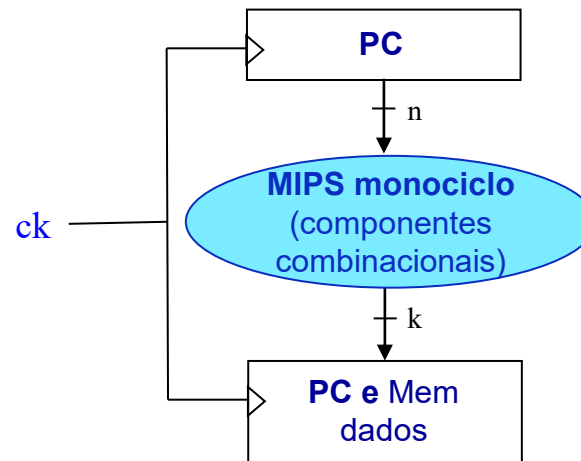
# O Processador MIPS Monociclo

## Período do Relógio x Escritas em Elementos de Memória

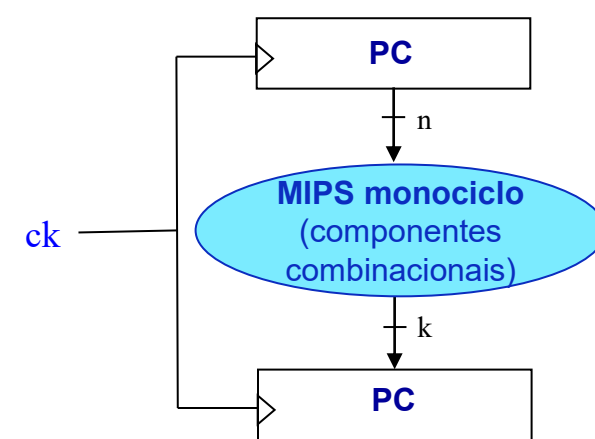
Instruções tipo R e lw



Instrução sw



Instruções beq e j



- Toda instrução inicia com a escrita do PC
- Toda a instrução termina com a escrita do PC
  - Instruções tipo R e lw também escrevem em um registrador
  - Instruções sw também escrevem na memória de dados

# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

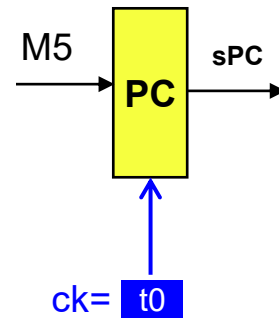
Nomenclatura para as características temporais dos componentes do MIPS monociclo:

Componente	Característica	Símbolo
Memória de instruções	tempo para leitura	$td_{I\_MEM}$
Memória de dados	tempo para leitura	$td_{I\_MEM}$
Memória de dados	tempo para escrita (setup)	$td_{EMEM}$
Banco de Registradores	tempo para leitura	$td_{I\_REG}$
Banco de Registradores	tempo para escrita (setup)	$td_{FREG}$
ULA	atraso para qualquer operação	$td_{ULA}$
Somador (qualquer)	atraso	$td_{soma}$
Mux 2:1 (qualquer)	atraso	$td_{mux}$
PC	tempo de setup	$tsu$
PC	tempo de carga	$tco$
PC	tempo de hold	$th$
Deslocador, extensão de sinal, porta AND	atraso	$td_{fios}$
Controle	atraso	$td_{controle}$
Controle da ULA	atraso	$td_{ctrlUL A}$

# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

### O Caso do PC



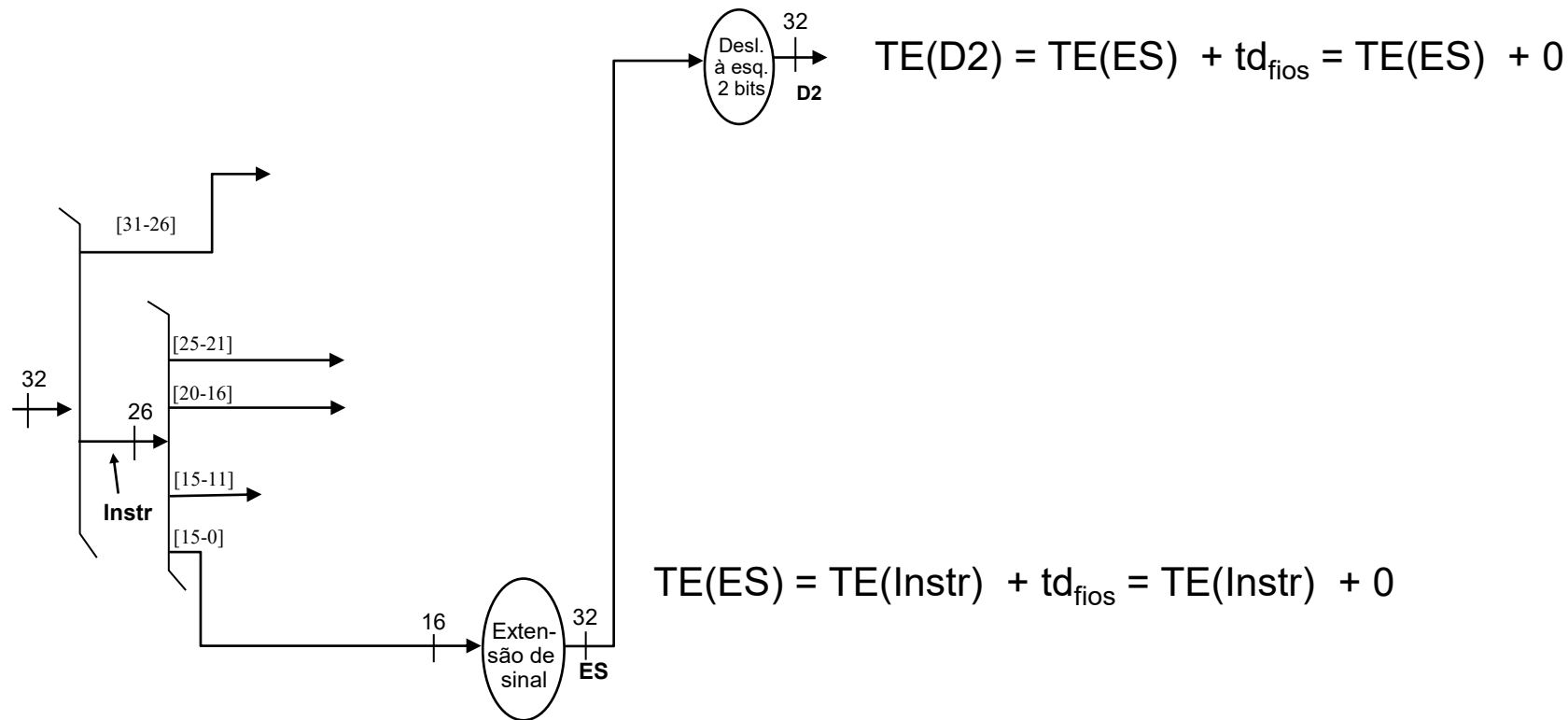
$$TE(sPC) = tco(PC) *$$

\* Toma-se a borda de relógio  $t_0$  como referência de tempo; logo,  $TE(sPC) = tco$  após  $t_0$ .

# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

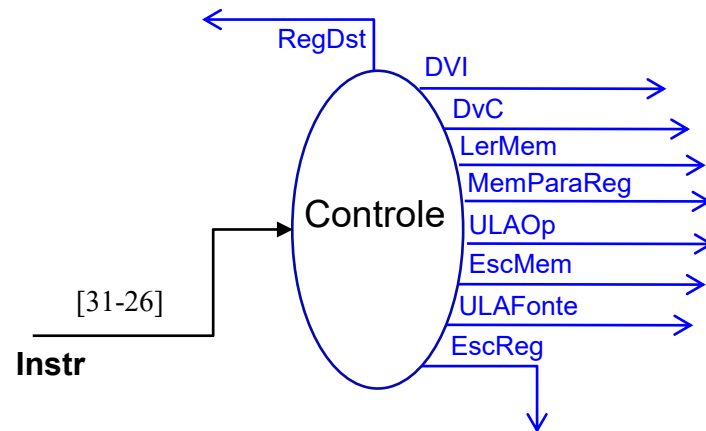
Para Elementos Combinacionais (exceto muxes):  
o caso da Extensão de sinal e do Deslocador



# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

Para Elementos Combinacionais (exceto muxes):  
o caso do Controle do MIPS monociclo



$$TE(\text{qq sinal de controle}) = TE(\text{Instr}) + td_{\text{controle}}$$

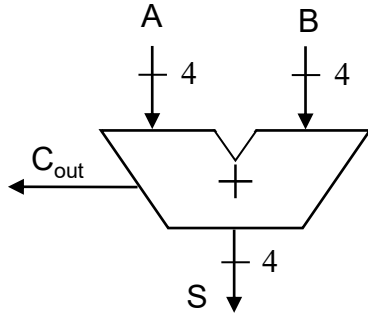
Obs: esta equação só vale para os sinais de controle que só dependem do opcode. Isto excluir **cULA** e **FontePC**.



# O Processador MIPS Monociclo

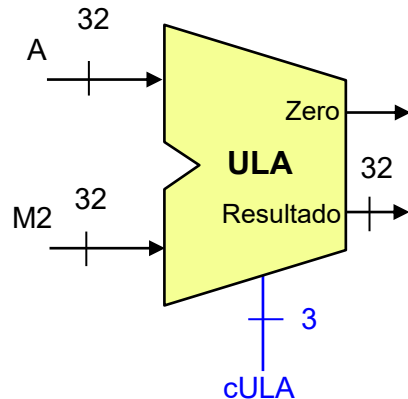
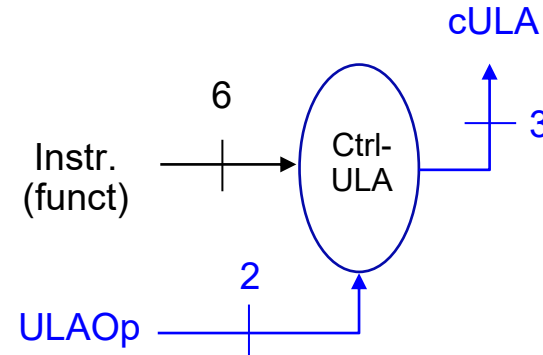
## Tempo de Estabilização (TE) no Nível RT

Para Elementos Combinacionais (exceto muxes)



$$TE(S) = \max \{ TE(A), TE(B) \} + td_{\text{soma}}$$
$$TE(C_{\text{out}}) = \max \{ TE(A), TE(B) \} + td_{\text{soma}}$$

$$TE(cULA) = \max \{ TE(\text{Instr}), TE(ULAOp) \} + td_{\text{ctrl-ULA}}$$



$$TE(\text{Resultado}) = \max \{ TE(A), TE(M2), TE(cULA) \} + td_{\text{ULA}}$$
$$TE(\text{Zero}) = \max \{ TE(A), TE(M2), TE(cULA) \} + td_{\text{ULA}}$$

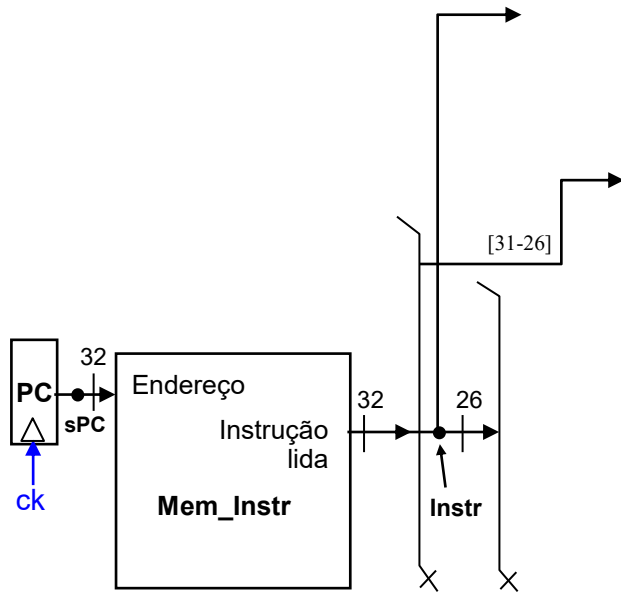


# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

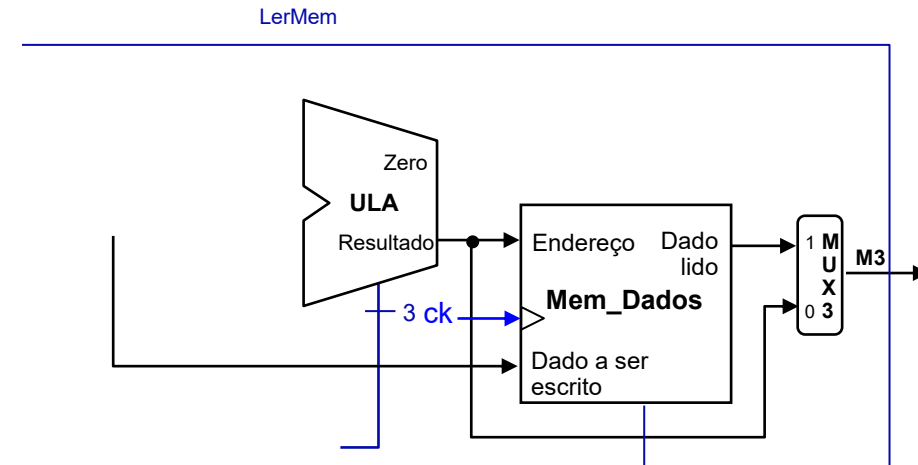
**Para Elementos Combinacionais (exceto muxes):**

Leituras das Memórias são assíncronas (e portanto, ocorrem de maneira similar à operação de um bloco combinacional)



**Nas leituras de Mem\_Instr:**

$$TE(Instr) = TE(sPC) + td_{LMEM}$$



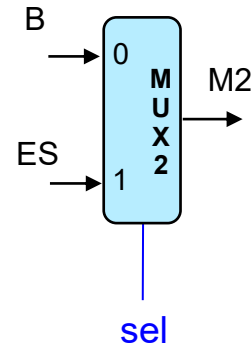
**Nas leituras de Mem\_Dados:**

$$TE(Dado lido) = \max\{ TE(Resultado), TE(LerMem) \} + td_{LMEM}$$

# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

### O Caso dos Muxes



$$TE(M2) = \max\{ TE\_S, TE(sel) \} + td_{mux}$$

Onde:

$TE\_S = TE(B)$  se  $sel = 0$

e

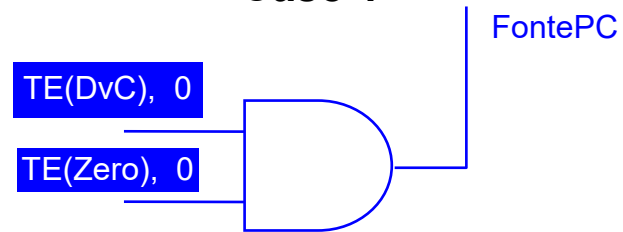
$TE\_S = TE(ES)$  se  $sel = 1$

# O Processador MIPS Monociclo

## Tempo de Estabilização (TE) no Nível RT

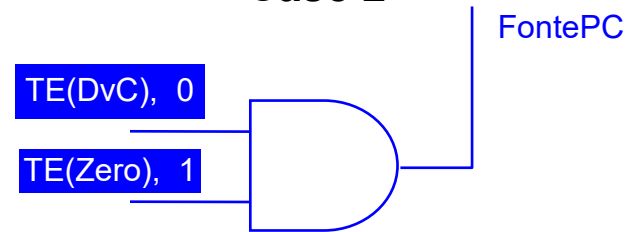
### O Caso do sinal FontePC

Caso 1



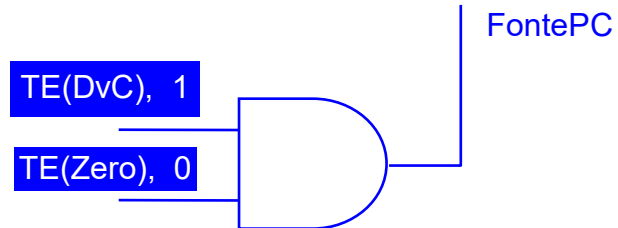
$$TE(\text{FontePC}) = \min\{TE(\text{DvC}), TE(\text{Zero})\}$$

Caso 2



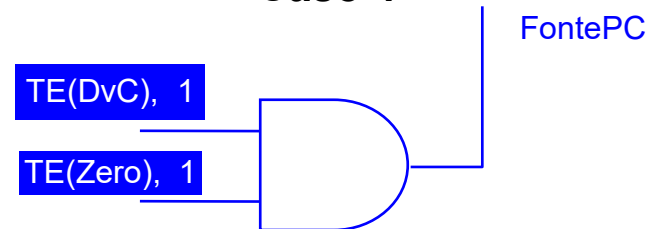
$$TE(\text{FontePC}) = TE(\text{DvC})$$

Caso 3



$$TE(\text{FontePC}) = TE(\text{Zero})$$

Caso 4



$$TE(\text{FontePC}) = \max\{TE(\text{DvC}), TE(\text{Zero})\}$$

### Legenda

TE(Zero), 0

↑      ↑

TE, valor lógico  
do sinal

# O Processador MIPS Monociclo

## Análise de *Timing* por meio dos TEs

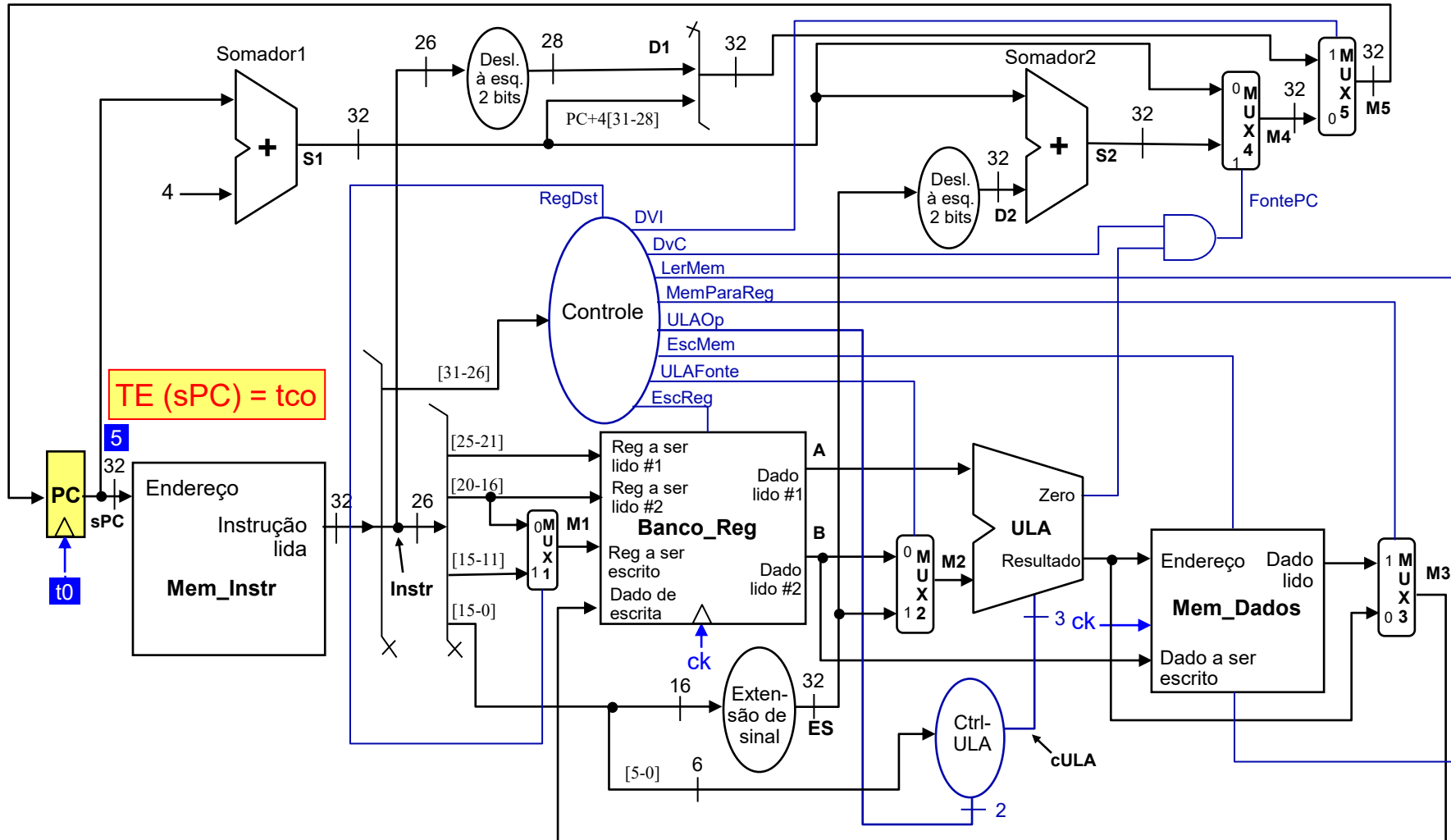
Para um exemplo concreto, assumamos os seguintes valores para as características temporais dos componentes do MIPS monociclo:

Componente	Característica	Símbolo	Valor [ps]
Memória de instruções	tempo para leitura	$td_{I\_MEM}$	300
Memória de dados	tempo para leitura	$td_{I\_MEM}$	300
Memória de dados	tempo para escrita (setup)	$td_{EMEM}$	300
Banco de Registradores	tempo para leitura	$td_{I\_REG}$	70
Banco de Registradores	tempo para escrita (setup)	$td_{EREG}$	70
ULA	atraso para qualquer operação	$td_{ULA}$	60
Somador (qualquer)	atraso	$td_{soma}$	20
Mux 2:1 (qualquer)	atraso	$td_{mux}$	2
PC	tempo de setup	$tsu$	5
PC	tempo de carga	$tco$	5
PC	tempo de hold	$th$	desprezível (=0)
Deslocador, extensão de sinal, porta AND	atraso	$td_{fios}$	desprezível (=0)
Controle	atraso	$td_{controle}$	desprezível (=0)*
Controle da ULA	atraso	$td_{ctrlULA}$	desprezível (=0)*

\* isto significa que o atraso para gerar qualquer sinal de controle é zero

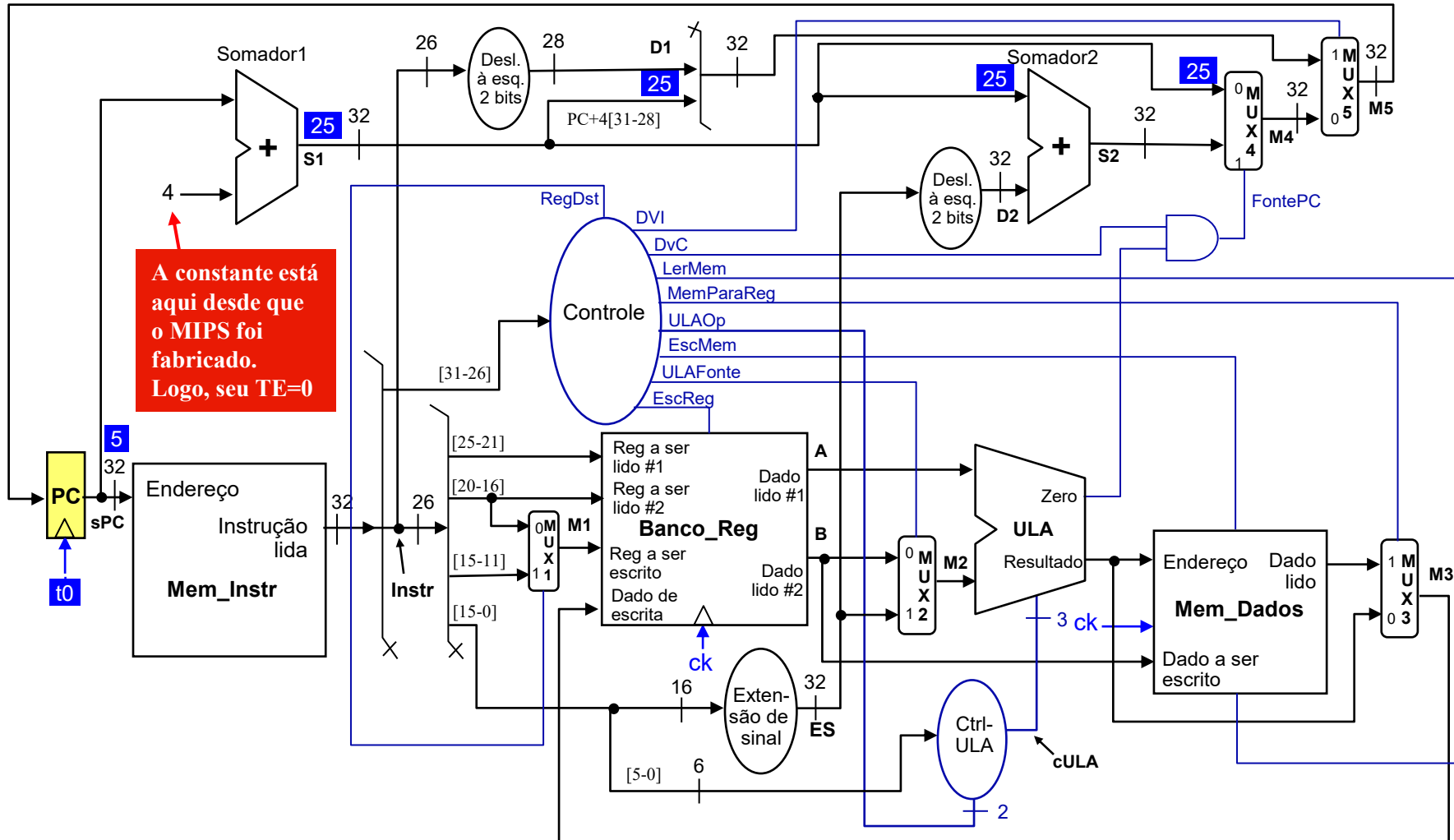
# O Processador MIPS Monociclo

## Início da Execução de Qualquer Instrução



# O Processador MIPS Monociclo

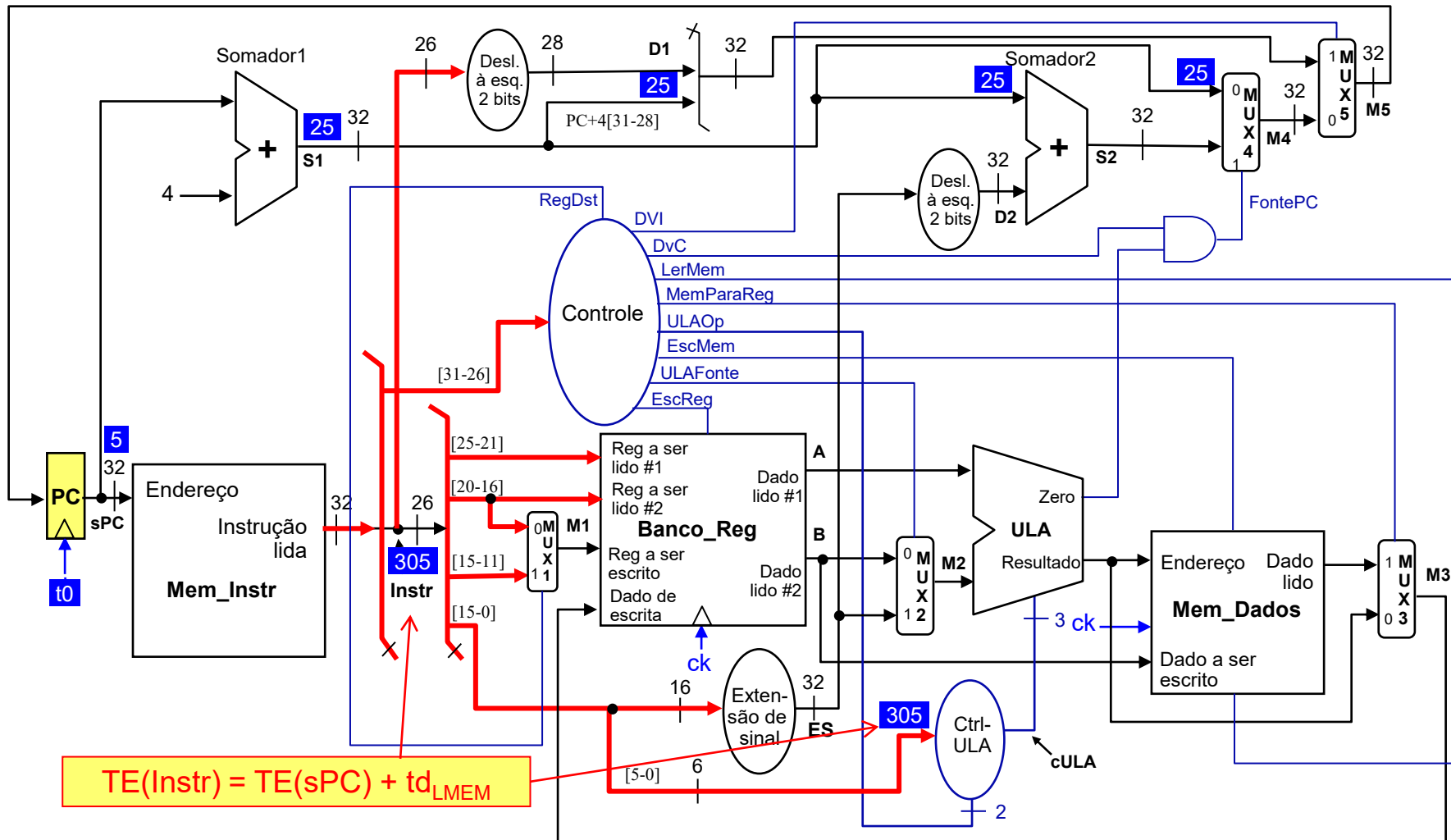
## Início da Execução de Qualquer Instrução





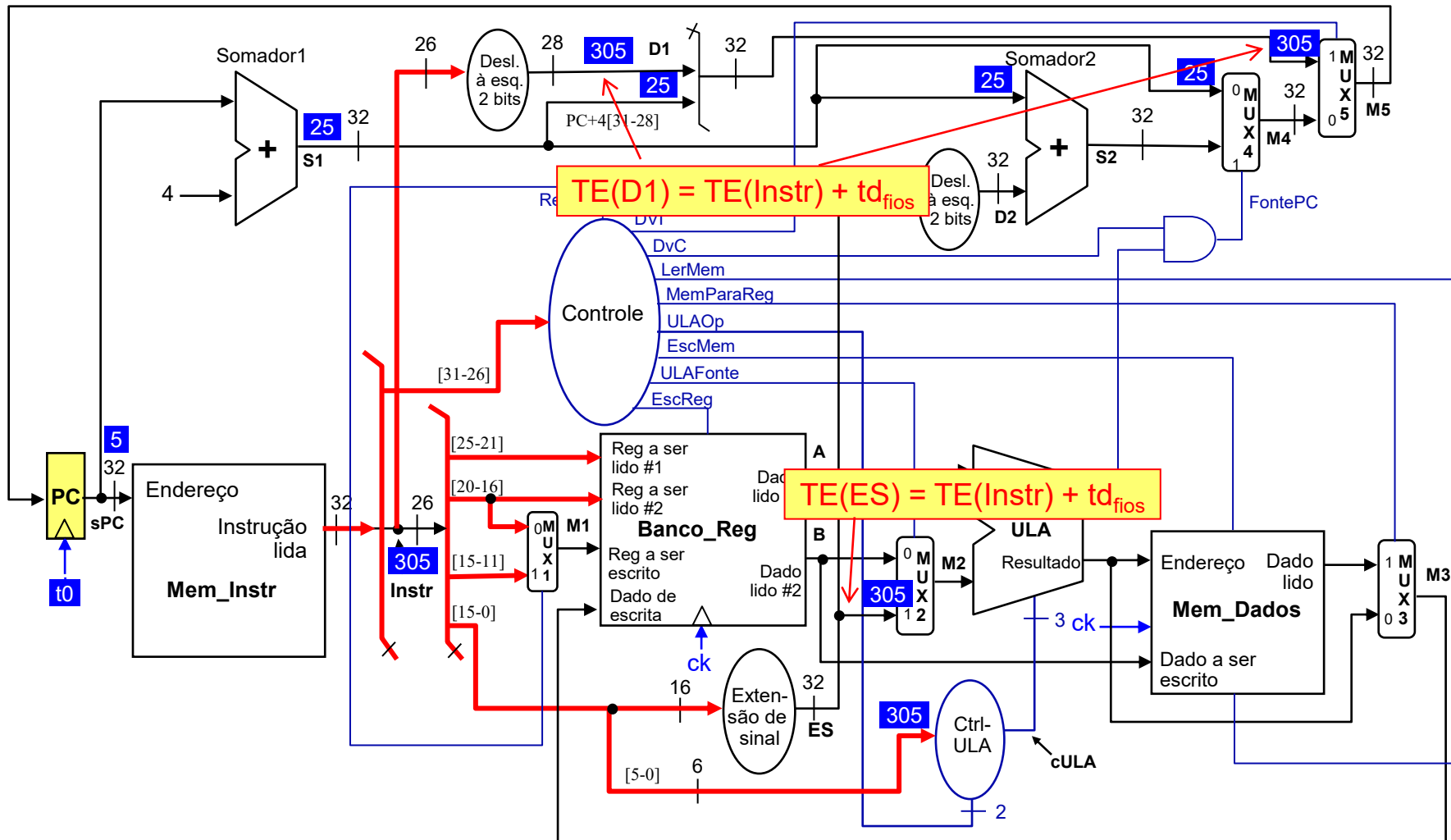
# O Processador MIPS Monociclo

## Instrução Lida!!



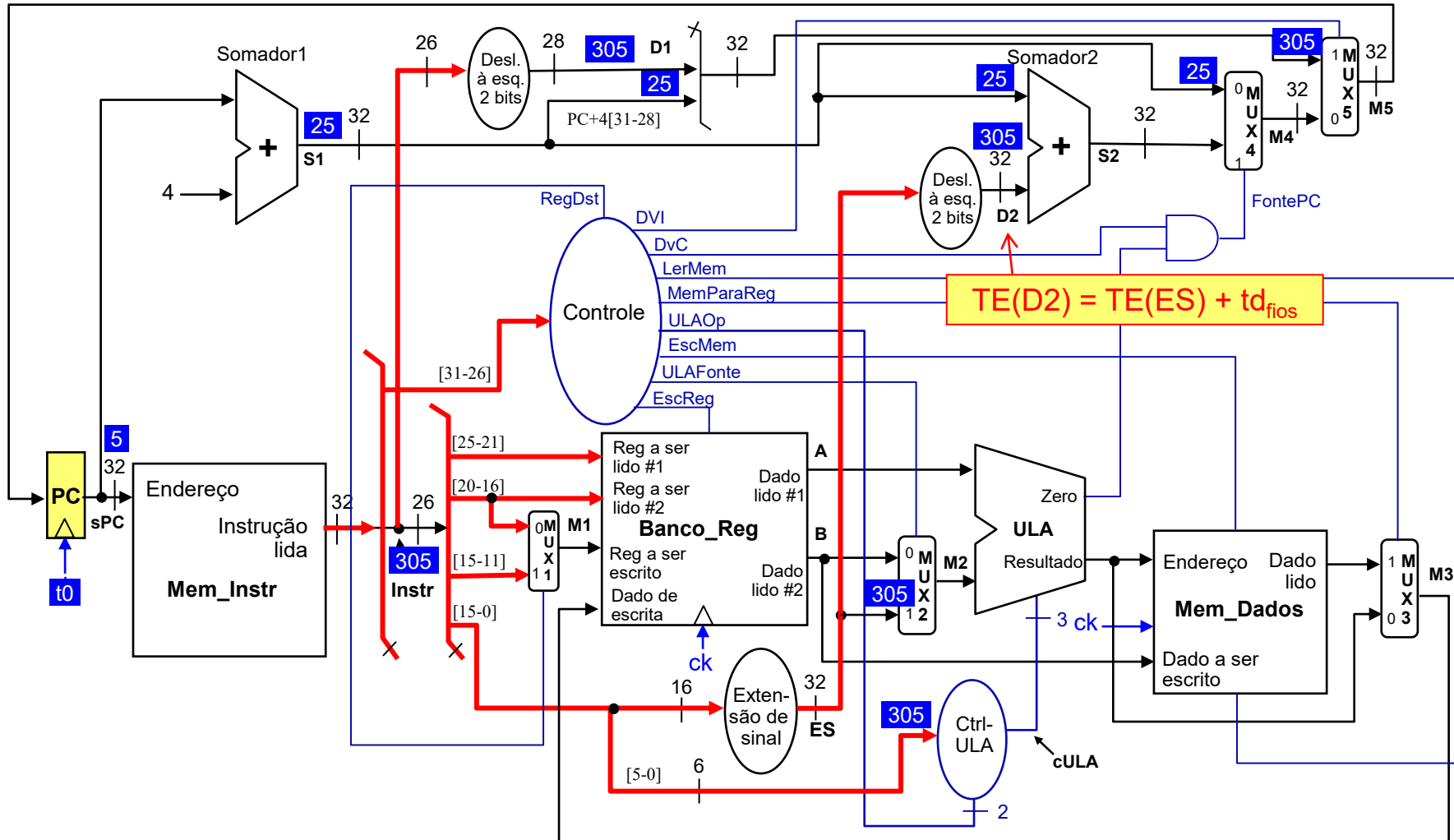
# O Processador MIPS Monociclo

Calculam-se todos  $TE(??) = TE(Instr) + td_{fios}$



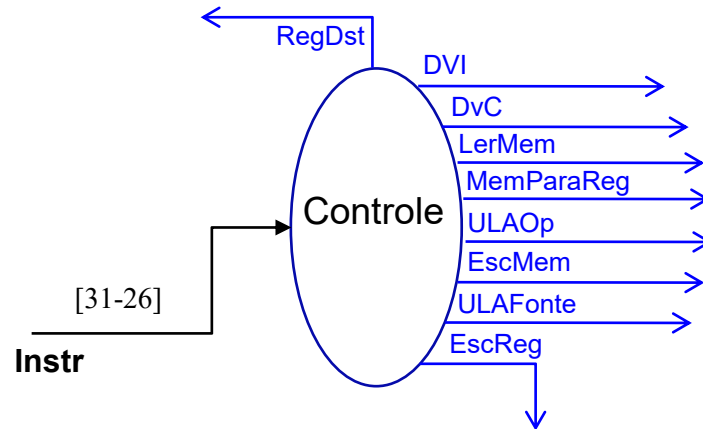
# O Processador MIPS Monociclo

Calculam-se todos  $TE(??) = TE(Instr) + td_{fios}$



# O Processador MIPS Monociclo

## A Seguir, calculam-se os TEs dos Sinais de Controle



$$TE(\text{qq sinal de controle}) = TE(\text{Instr}) + td_{\text{controle}}$$

No exemplo em questão, temos:

Componente	Característica	Símbolo	Valor [ps]
...	...	...	...
Controle	atraso	$td_{\text{controle}}$	desprezível (=0)
Controle da ULA	atraso	$td_{\text{ctrlULA}}$	desprezível (=0)

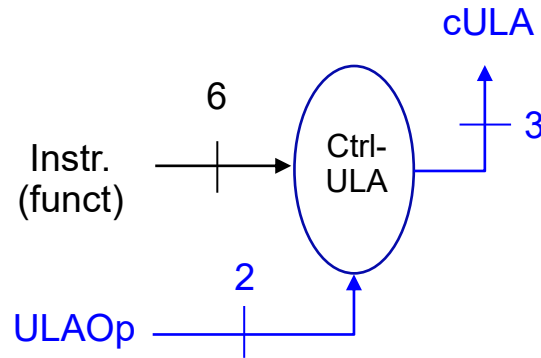
Então:

$$TE(\text{qq sinal de controle}) = TE(\text{Instr}) + td_{\text{controle}} = 305 + 0 = \mathbf{305 \text{ ps}}$$

**Cuidado!** Se  $td_{\text{controle}} \neq 0 \rightarrow TE(\text{qq sinal de controle}) \neq TE(\text{Instr})$

# O Processador MIPS Monociclo

## E o TE de cULA



$$TE(cULA) = \max \{ TE(Instr), TE(ULAOp) \} + td_{ctrl-ULA}$$

No exemplo em questão, temos:

Componente	Característica	Símbolo	Valor [ps]
...	...	...	...
Controle	atraso	$td_{controle}$	desprezível (=0)
Controle da ULA	atraso	$td_{ctrlULA}$	desprezível (=0)

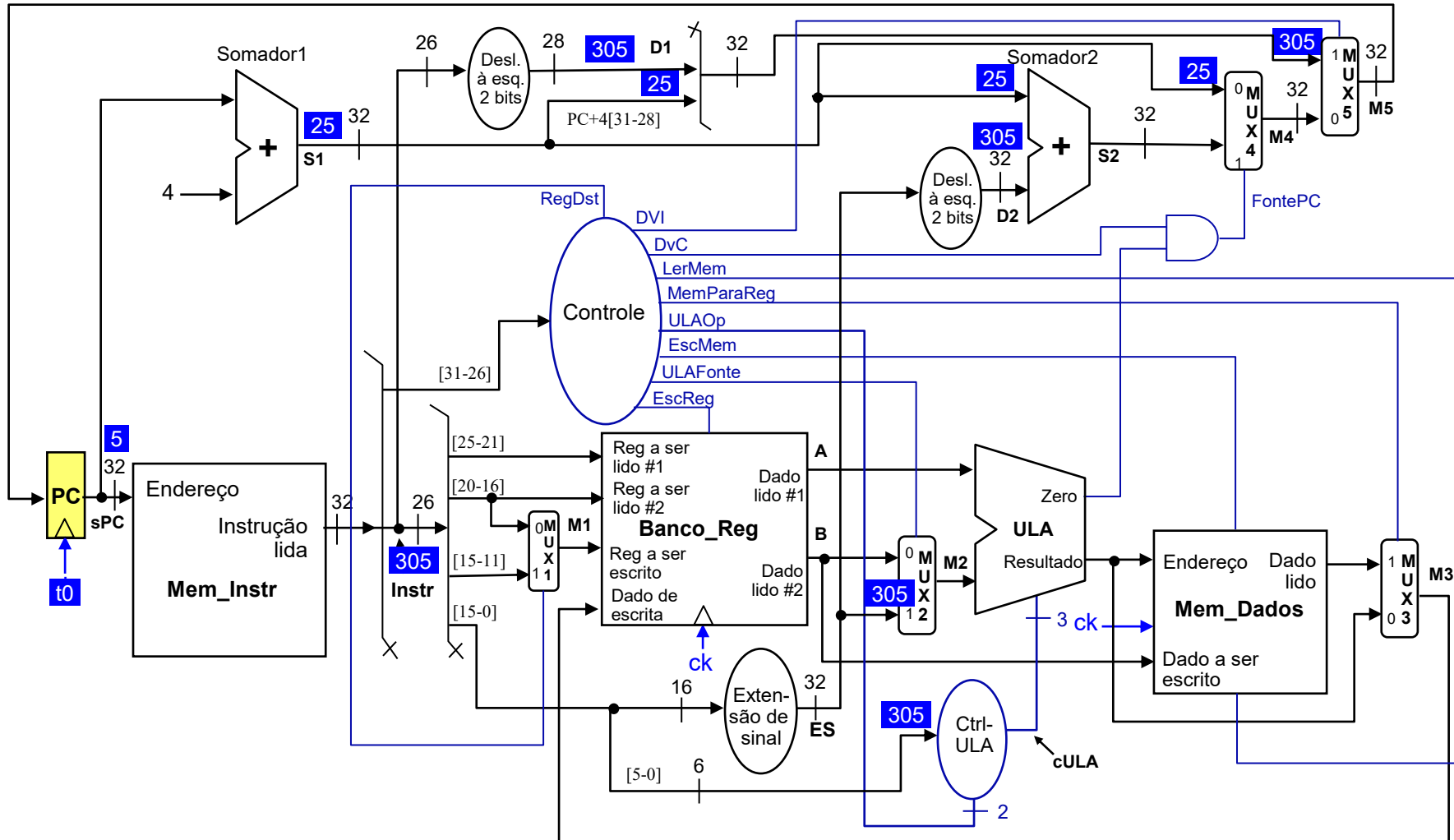
Então:

$$TE(cULA) = \max \{ TE(Instr), TE(ULAOp) \} + td_{ctrl-ULA} = \max \{ 305, 305 \} + 0 = 305 \text{ ps}$$

**Cuidado!** Se  $td_{ctrlULA} \neq 0 \rightarrow TE(cULA) \neq TE(ULAOp)$

# O Processador MIPS Monociclo

Consideraremos cada Tipo de Instrução em Separado



# O Processador MIPS Monociclo

---

## Exercício:

Elabore em grupo de até 6 participantes um vídeo explicando a análise de timing passo a passo para as partes seguintes das instruções do tipo R, Load, Store, Desvio Condicional e Desvio Incondicional.

O vídeo deverá ser didático e mostrar passo-a-passo os tempos de estabilização dos sinais, como apresentado para o bloco inicial do processamento de instruções nestes slides ( 18 a 28).

A tabela de valores continuará sendo a apresentada no slide 17.

No Fórum específico deste exercício, vocês devem:

- Postar o link para o vídeo produzido, identificando o nome de todos os participantes na postagem
- Comentar as postagens dos colegas

# O Processador MIPS Monociclo

## 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)

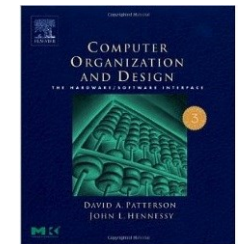
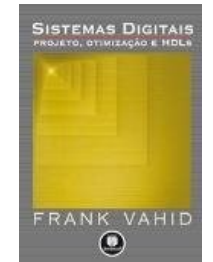
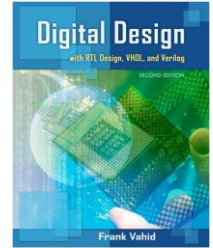
Seção 4.3.

## Aritmética com Inteiros e construção da ULA:

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

Se usar a 2<sup>a</sup> Edição: Seções 4.1 a 4.5.

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





# O Processador MIPS Monociclo

---

## Controle da ULA do MIPS

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

Se usar a 2<sup>a</sup> Edição: Seção 5.1 a 5.2, 5.3 e C.2 (Apêndice C)

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

