



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



Sistemas Digitais

INE 5406

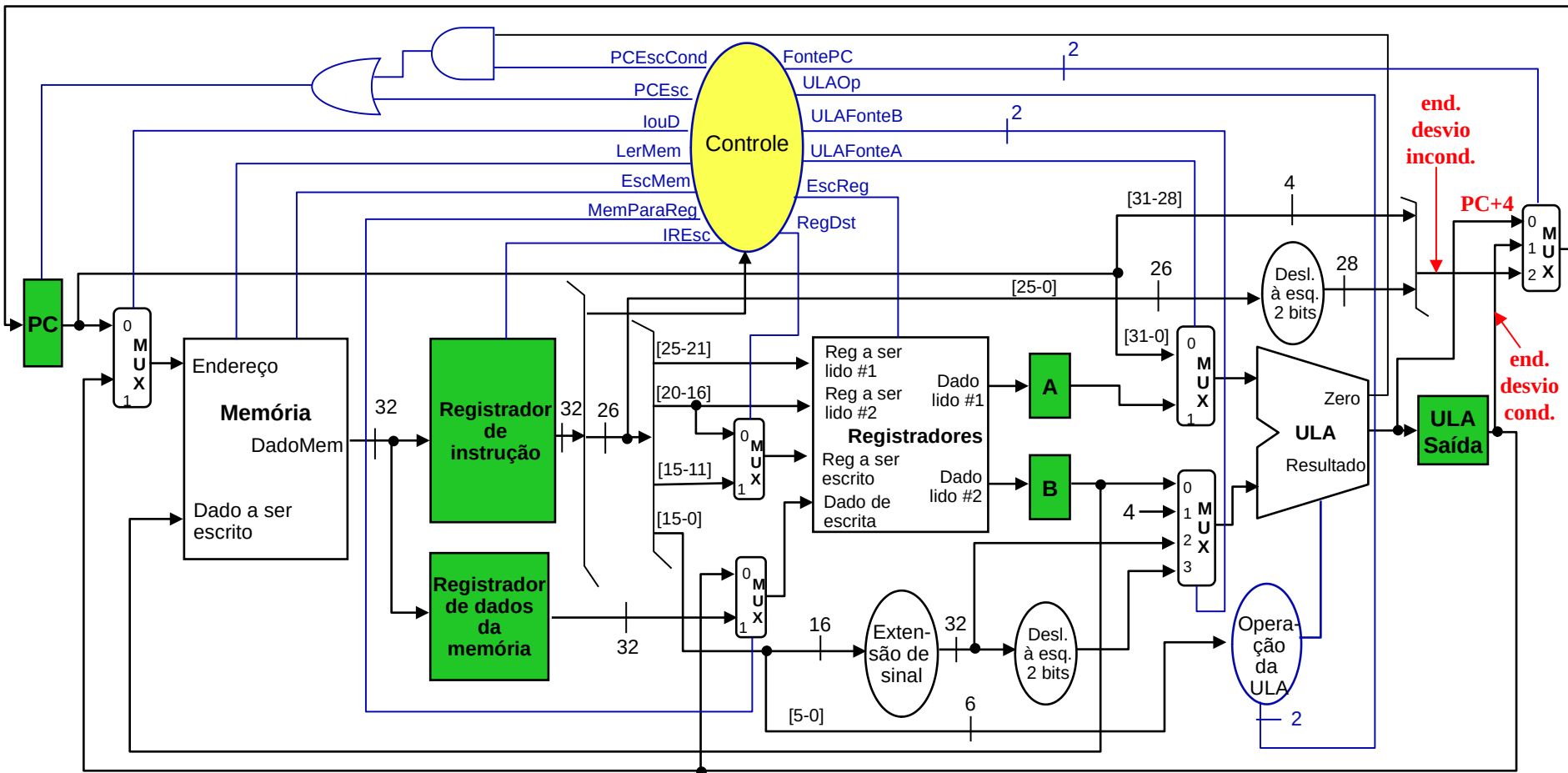
Aula 12-T

4. O Processador MIPS multiciclo: análise de *timing* (tempos de estabilização dos sinais, atrasos dos caminhos, caminhos críticos e frequência máxima do relógio).

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

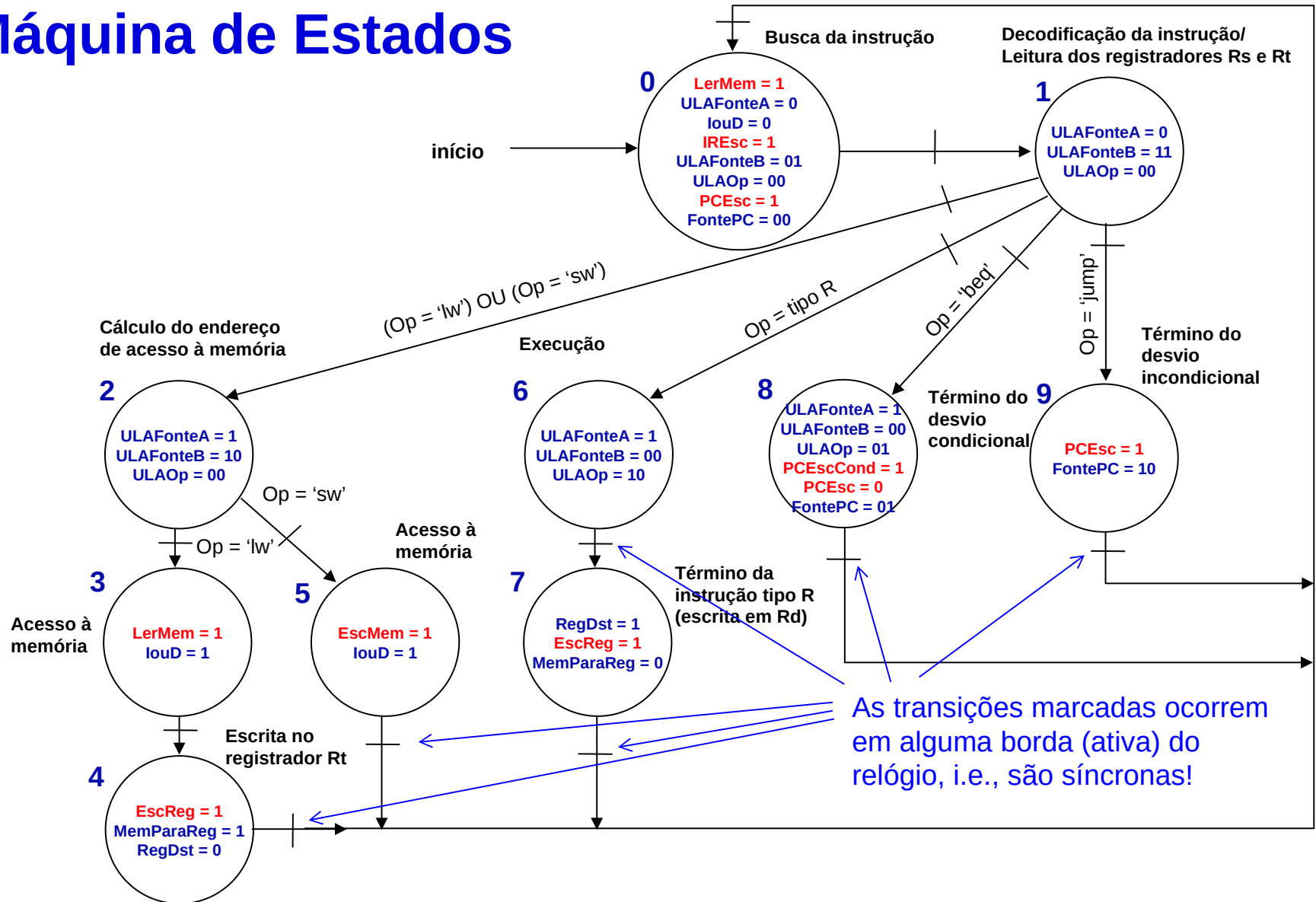
O Processador MIPS Multiciclo

Identificação dos Estágios entre Registradores



O Processador MIPS Multiciclo

Máquina de Estados



O Processador MIPS Multiciclo

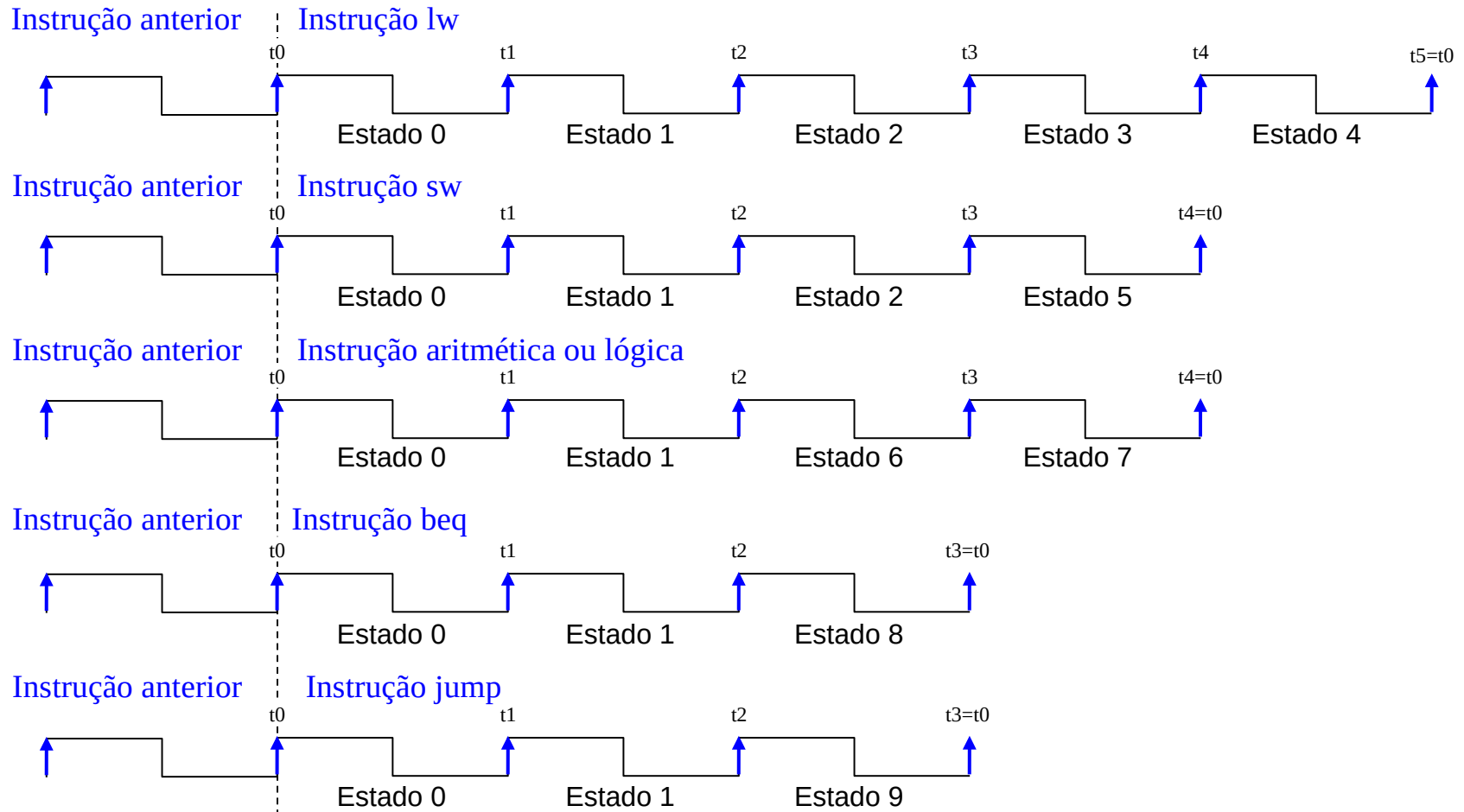
Descrição dos Estados no Nível RT

Nome do passo	Instrução tipo R	Instrução lw	Instrução sw	Instrução beq	Instrução j	
Busca da instrução	0 $RI = Mem[PC]$ $PC = PC + 4$					→ t0
Decodificação da instrução & leitura dos registradores Rs e Rt & cálculo do endereço de desvio (cond.)	1 $A = Reg [RI[25-21]]$ $B = Reg [RI[20-16]]$ $ULASaída = PC + (extensão\ de\ sinal(RI[15-0]) << 2)$					→ t1
Execução, cálculo do endereço de acesso à memória, término de uma instrução branch/jump	6 $ULASaída = A op B$	2 $ULASaída = A + extensão\ de\ sinal(RI[15-0])$	8 se $(A == B)$ então $PC = ULASaída$	9 $C = PC[31-28] (RI[25-0] << 2)$		→ t2
Término de uma instrução store word ou de tipo R	7 $Reg [RI[15-11]] = ULASaída$	3 $RDM = Mem [ULASaída]$	5 $Mem [ULASaída] = B$			→ t3
Término de uma instrução load word		4 $Reg[RI[20-16]] = RDM$				→ t4
Número de estados	4	5	4	3	3	→ t5

Nesta tabela, cada linha horizontal pode ser imaginada como uma borda (ativa) do relógio (observando, entretanto, que o período do relógio é fixo, independente do estado)

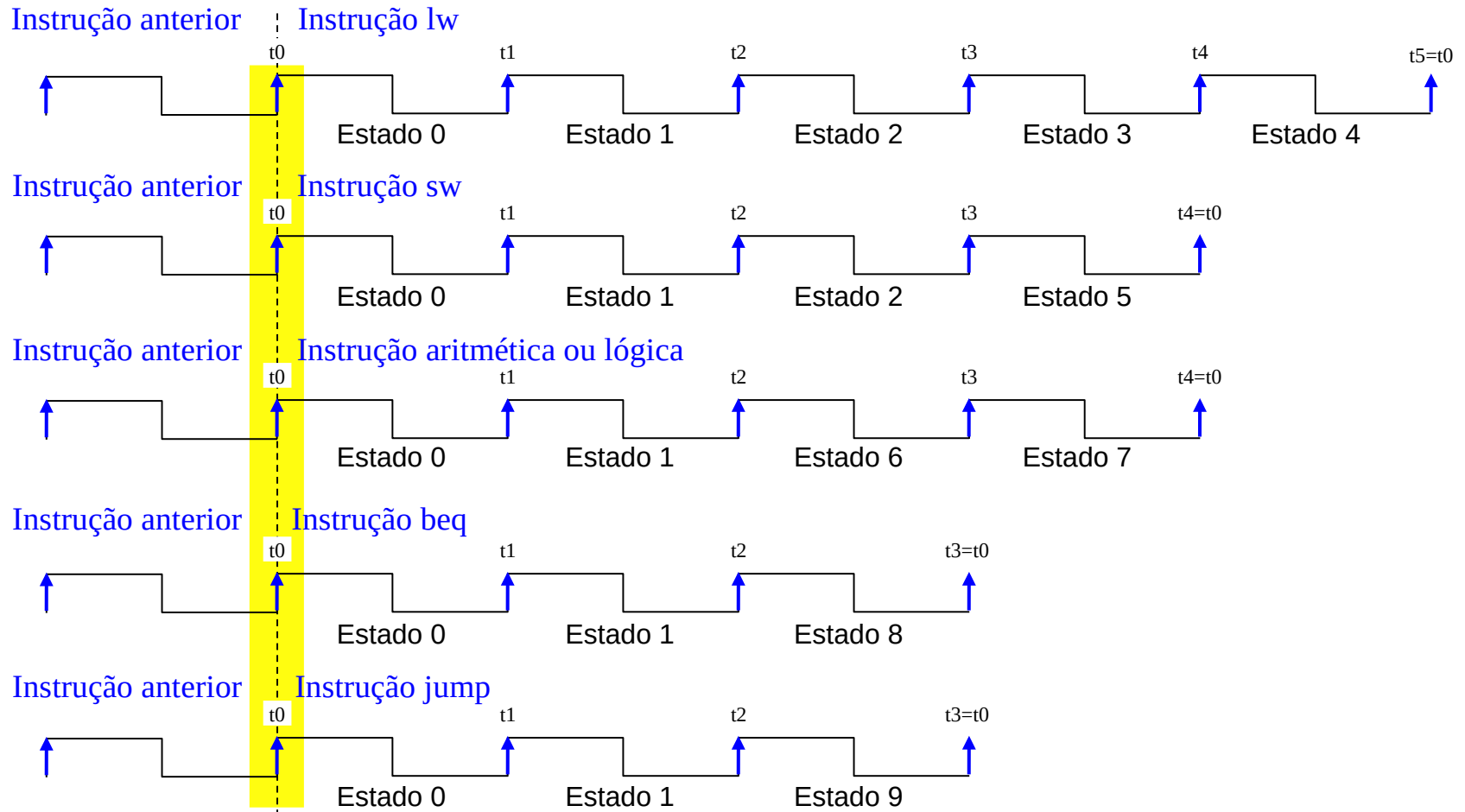
O Processador MIPS Multiciclo

Execução Multiciclo das Instruções



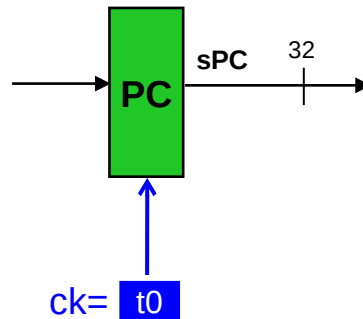
O Processador MIPS Multiciclo

Execução Multiciclo das Instruções



O Processador MIPS Multiciclo

Tempo de Estabilização (TE) do PC em t_0



$$TE(sPC) = t_{co}(PC) *$$

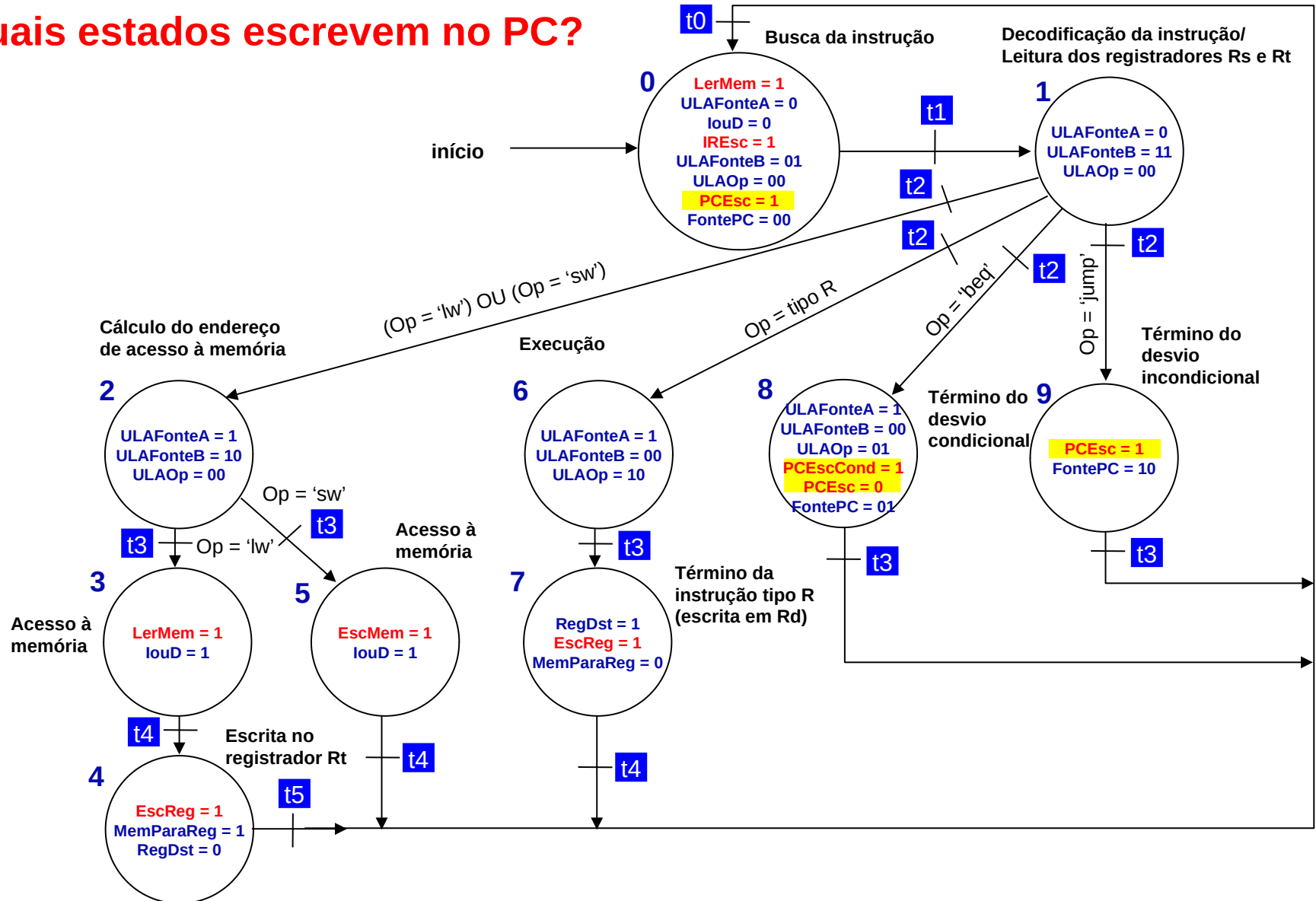
O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Nome do passo	Instrução tipo R	Instrução lw	Instrução sw	Instrução beq	Instrução j	
Busca da instrução	0 $RI = Mem[PC]$ $PC = PC + 4$					→ t0
Decodificação da instrução & leitura dos registradores Rs e Rt & cálculo do endereço de desvio (cond.)	1 $A = Reg [RI[25-21]]$ $B = Reg [RI[20-16]]$ $ULASaída = PC + (extensão\ de\ sinal(RI[15-0]) << 2)$					→ t1
Execução, cálculo do endereço de acesso à memória, término de uma instrução branch/jump	6 $JLASaída = A\ op\ B$	2 $JLASaída = A + extensão\ de\ sinal(RI[15-0])$	8 se $(A == B)$ então $PC = ULASaída$	9 $C = PC[31-28] (RI[25-0] << 2)$		→ t2
Término de uma instrução store word ou de tipo R	7 $Reg [RI[15-11]] = ULASaída$	3 $RDM = Mem [ULASaída]$	5 $Mem [ULASaída] = B$			→ t3
Término de uma instrução load word		4 $Reg[RI[20-16]] = RDM$				→ t4
Número de estados	4	5	4	3	3	→ t5

O Processador MIPS Multiciclo

Quais estados escrevem no PC?

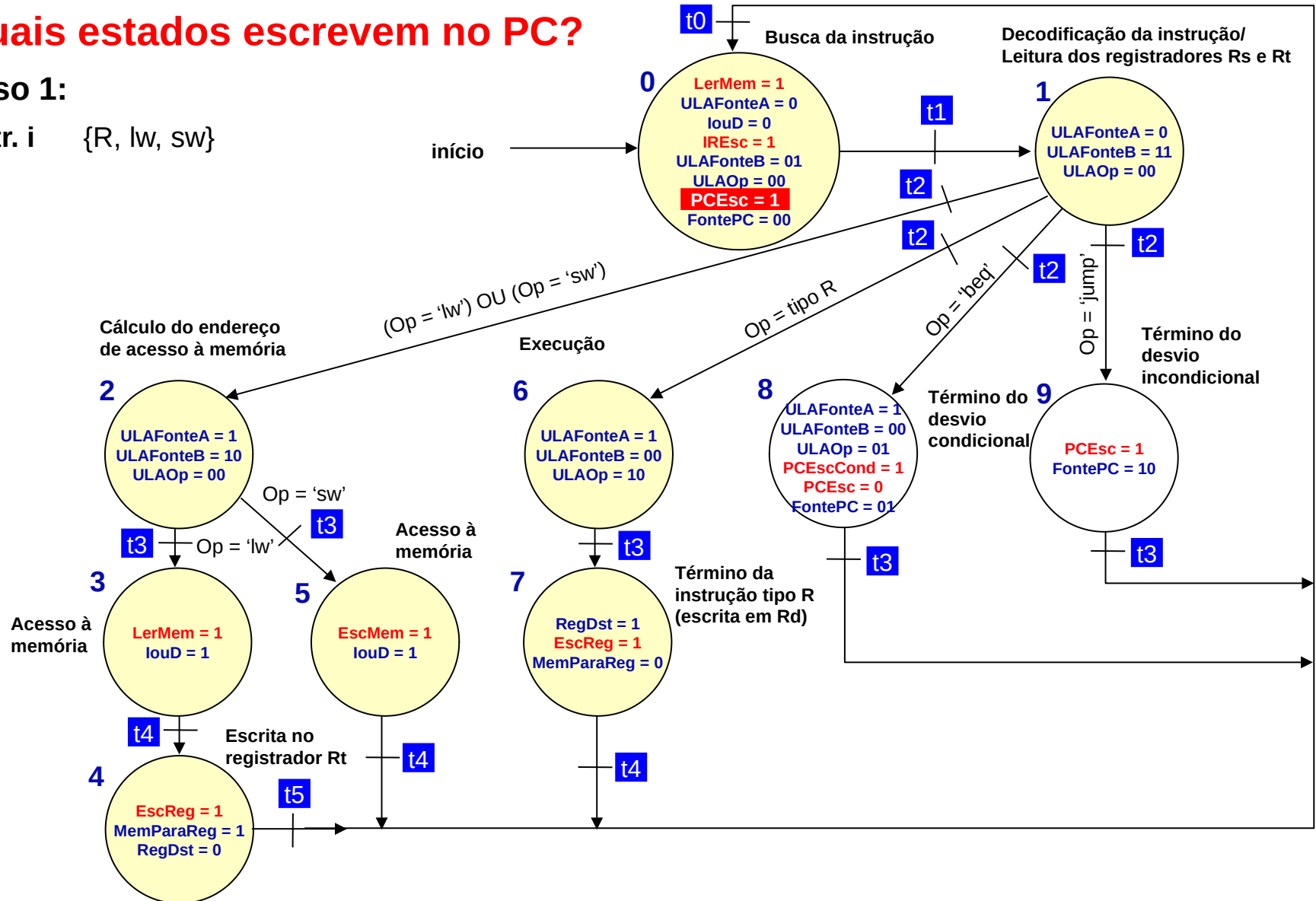


O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 1:

Instr. i {R, lw, sw}



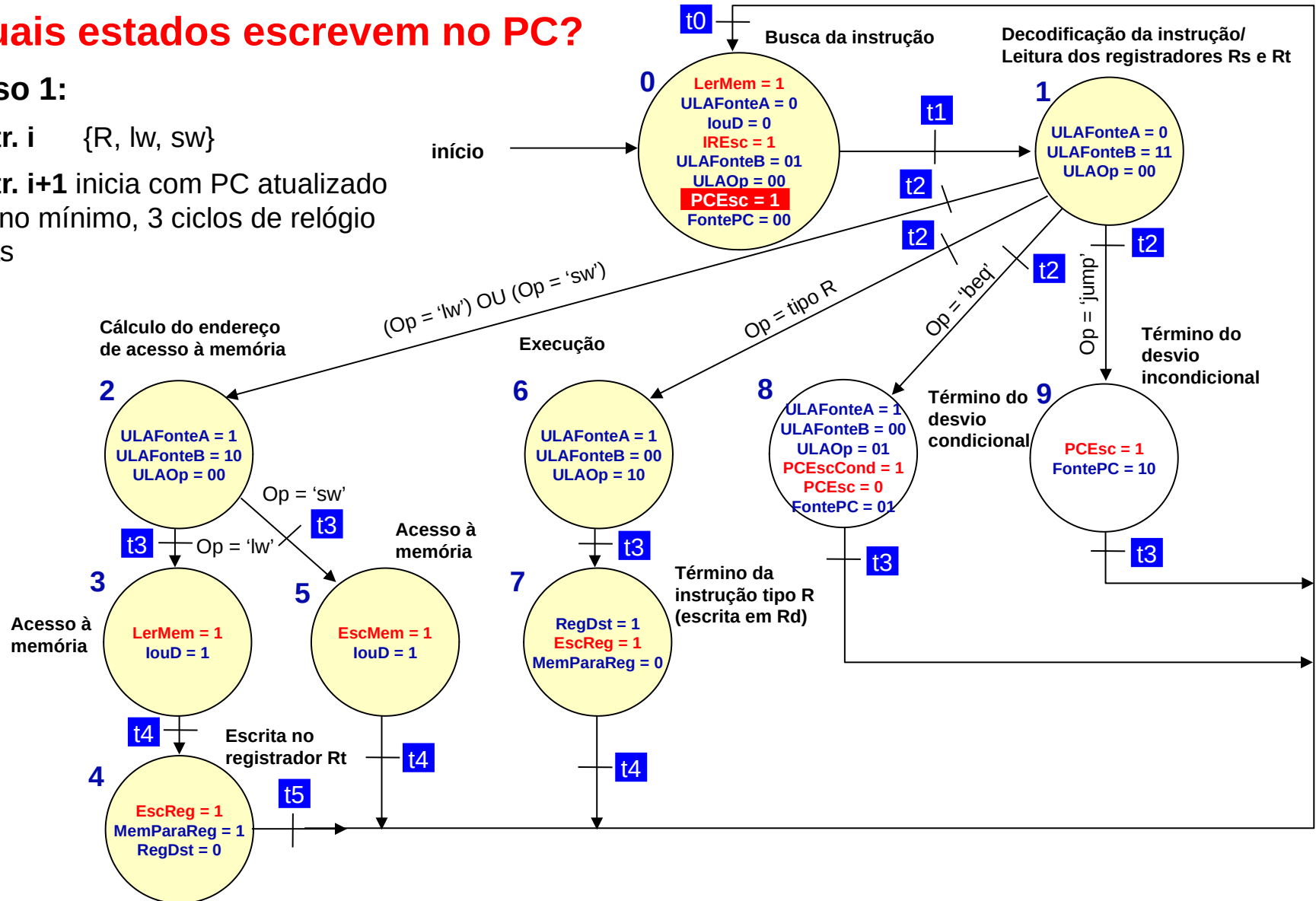
O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 1:

Instr. i {R, lw, sw}

Instr. i+1 inicia com PC atualizado
há, no mínimo, 3 ciclos de relógio
atrás

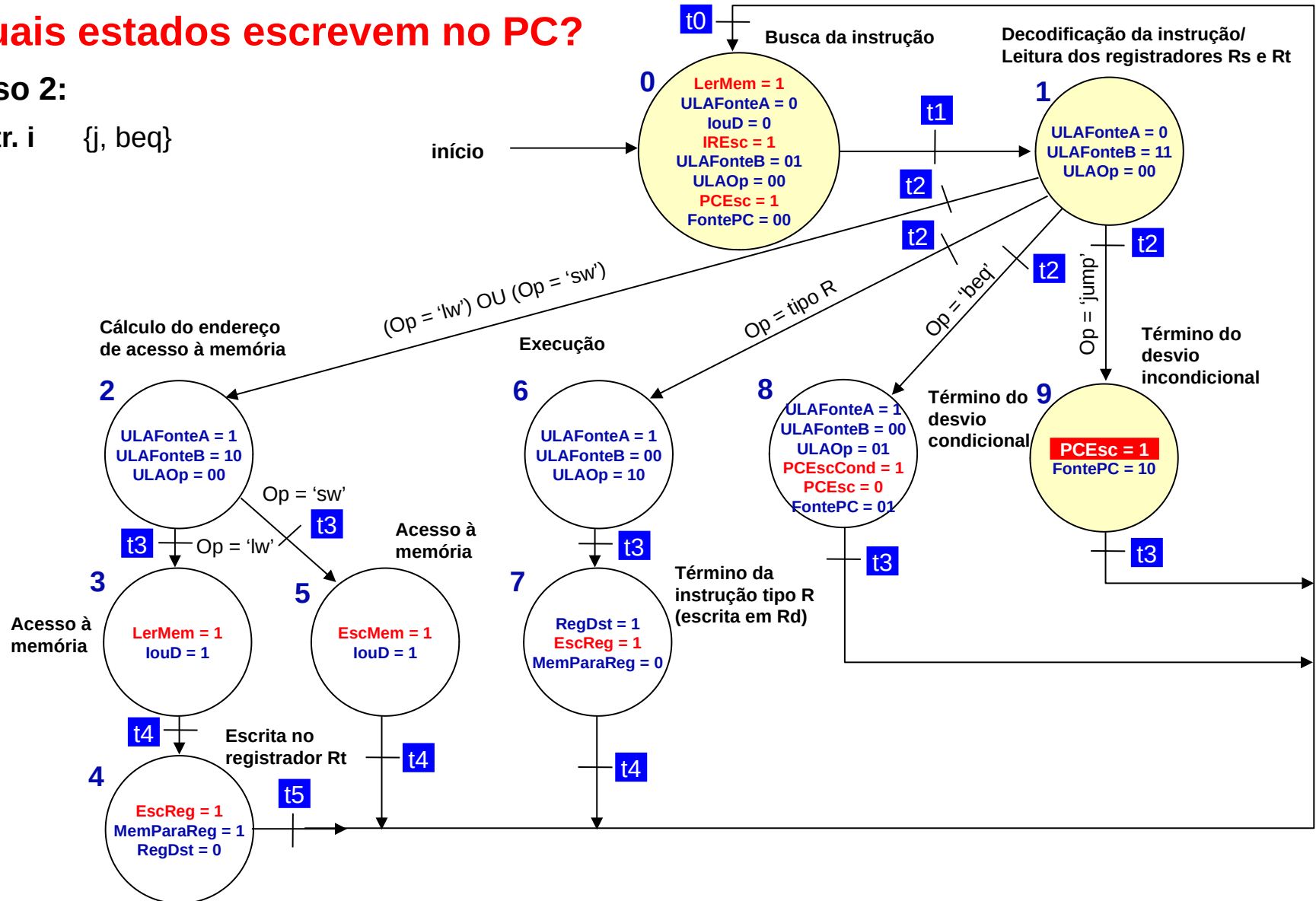


O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 2:

Instr. i {j, beq}

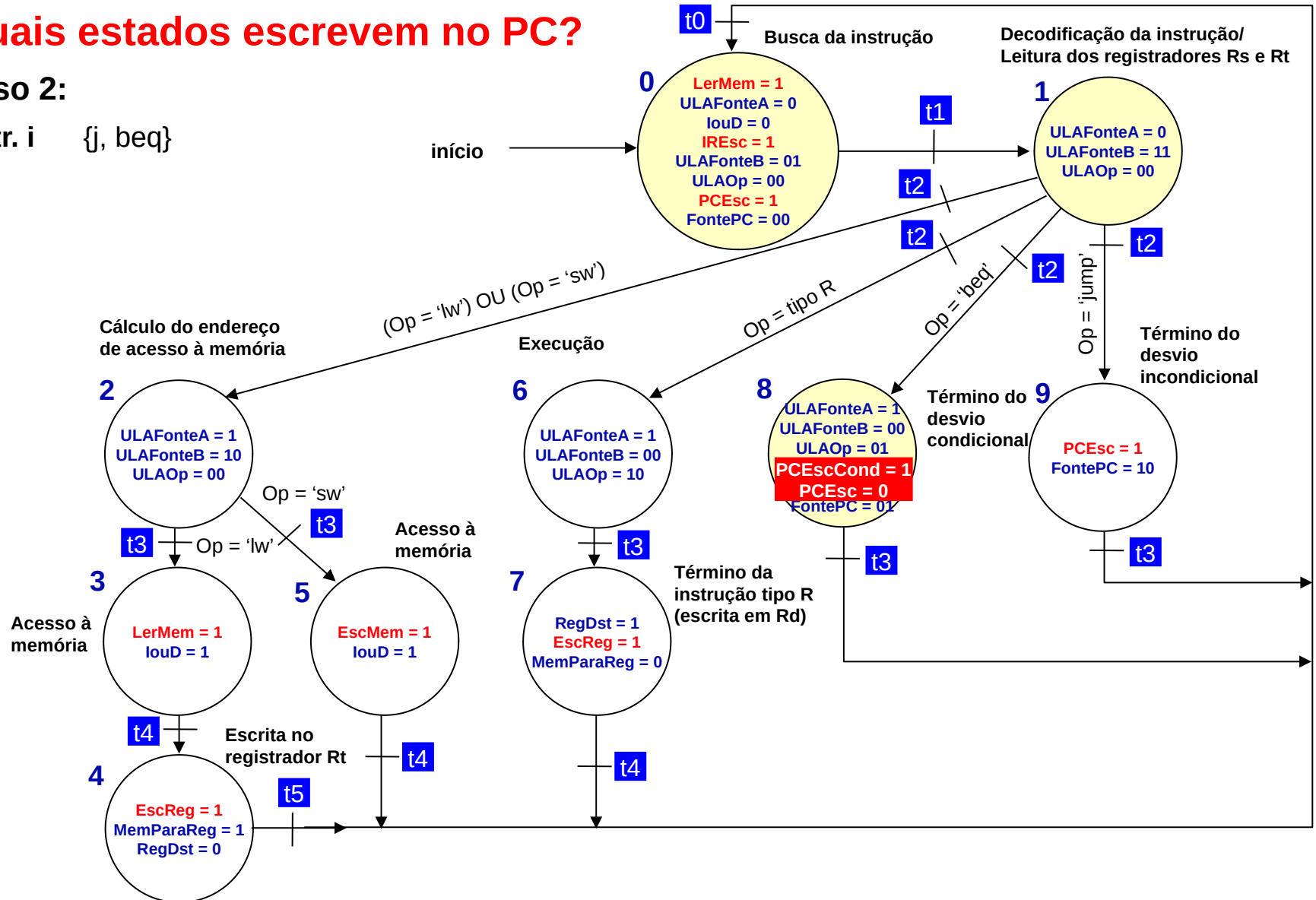


O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 2:

Instr. i {j, beq}



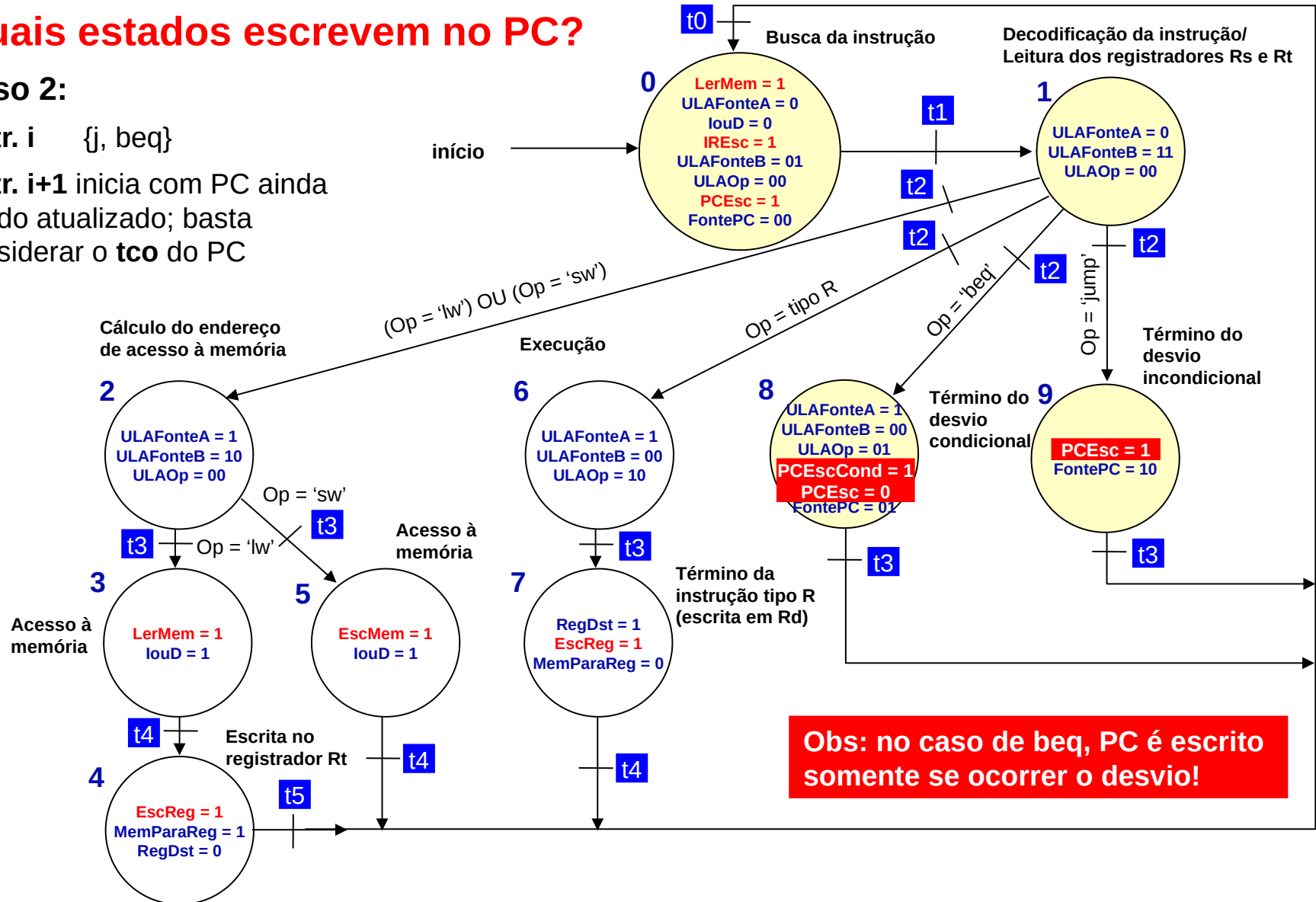
O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 2:

Instr. i {j, beq}

Instr. i+1 inicia com PC ainda sendo atualizado; basta considerar o **tco** do PC



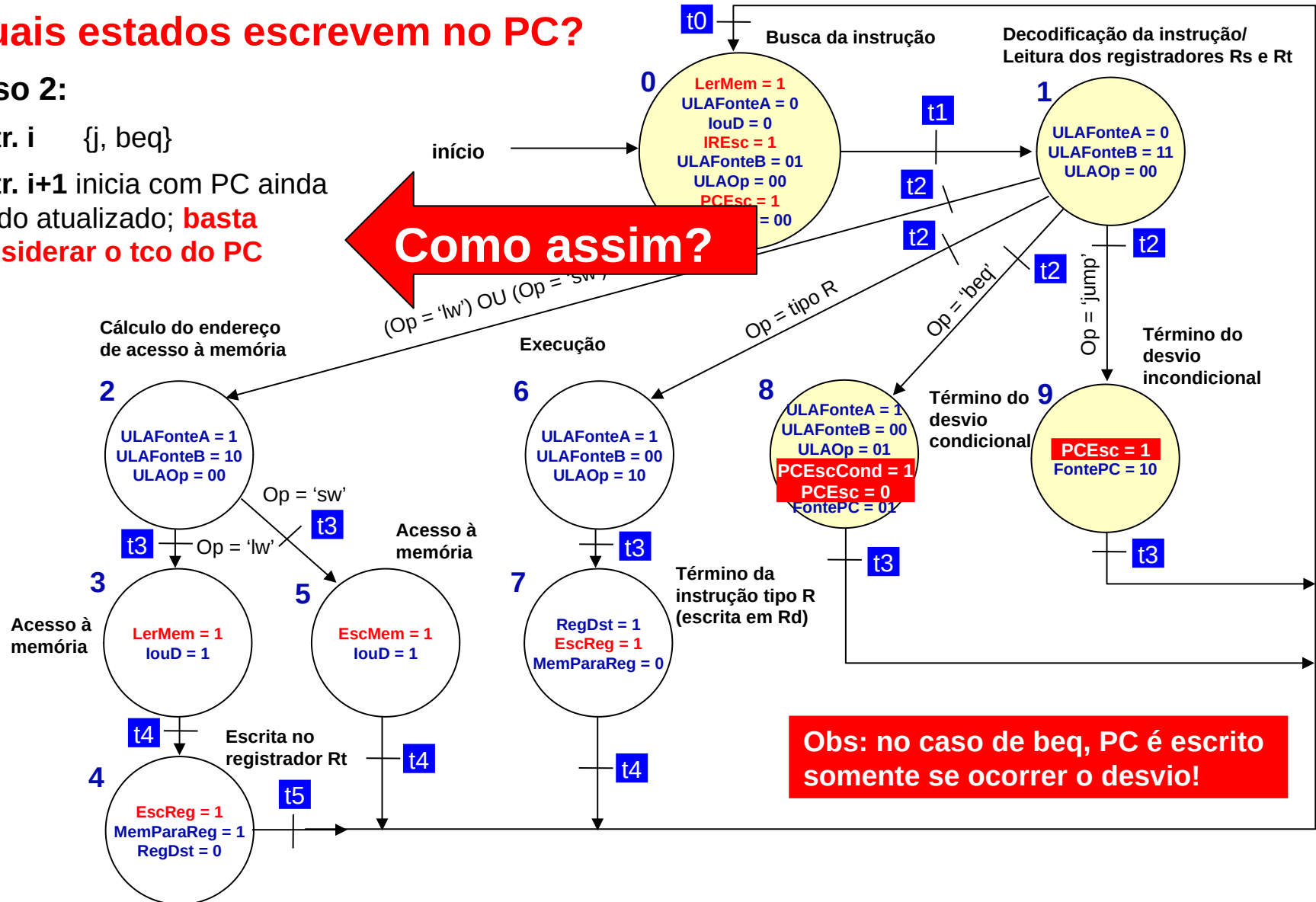
O Processador MIPS Multiciclo

Quais estados escrevem no PC?

Caso 2:

Instr. i {j, beq}

Instr. i+1 inicia com PC ainda sendo atualizado; **basta considerar o tco do PC**



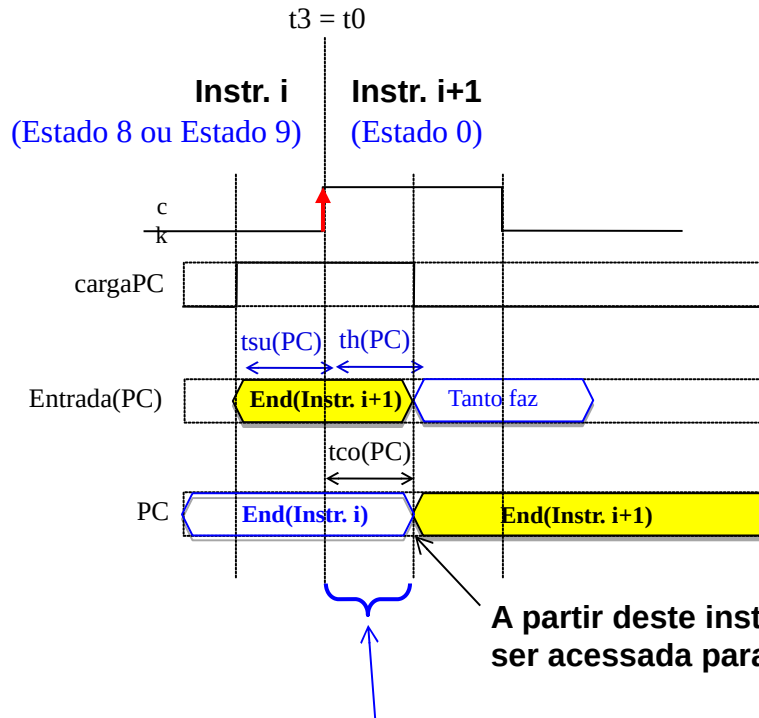
O Processador MIPS Multiciclo

Assim ó (Explicação com formas de onda para o PC...)

Caso 2:

Instr. i {beq, j}

Instr. i+1 inicia com PC ainda sendo atualizado; basta considerar o **tco** do PC



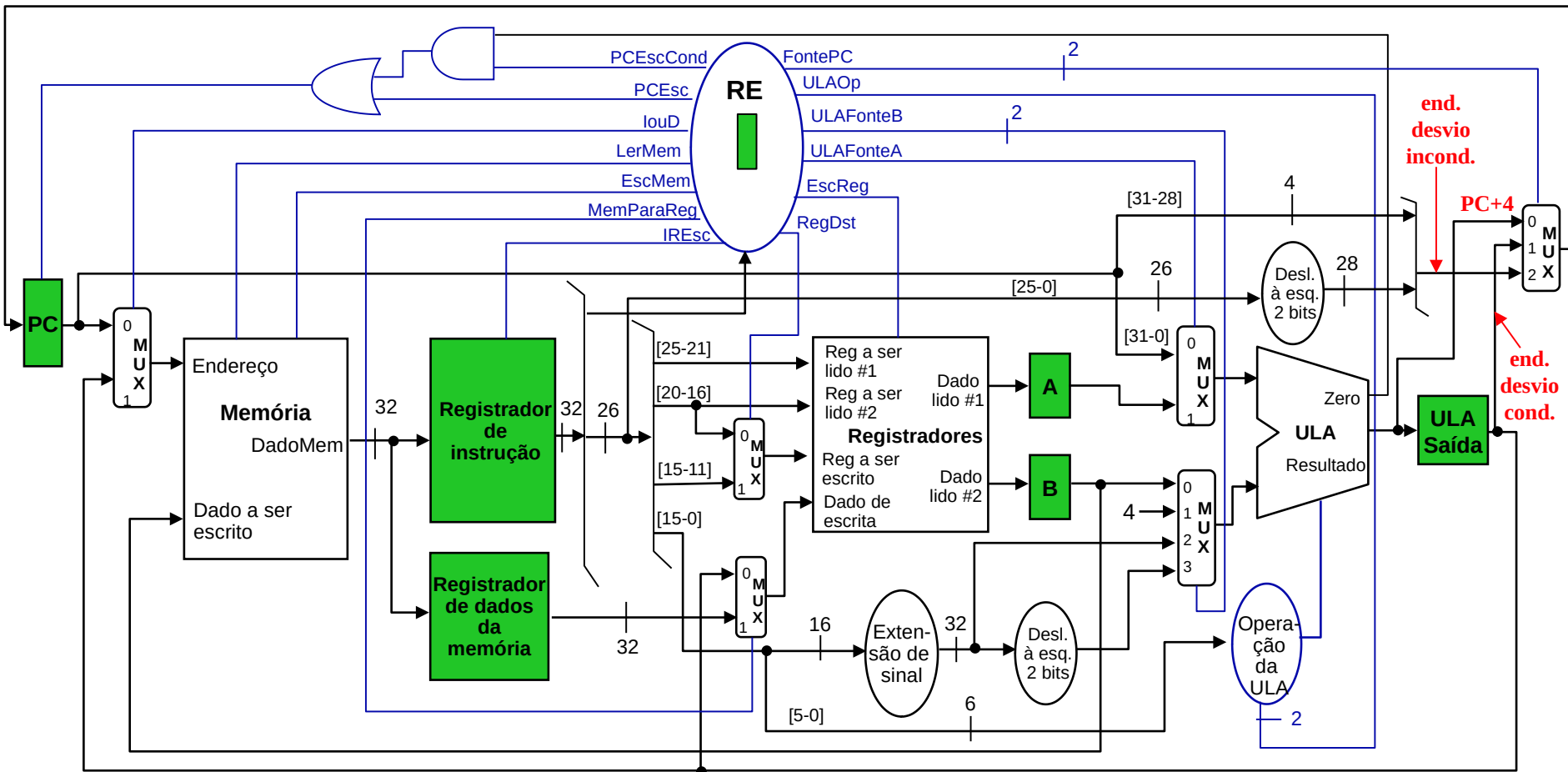
A partir deste instante a memória já pode ser acessada para buscar a Instr. i+1

Atualização do PC avança o Estado 0 da Instr. i+1 por um tempo = tco(PC)

Conclusão: considerando-se o tco(PC) na análise de *timing* do Estado 0, leva-se este efeito em conta

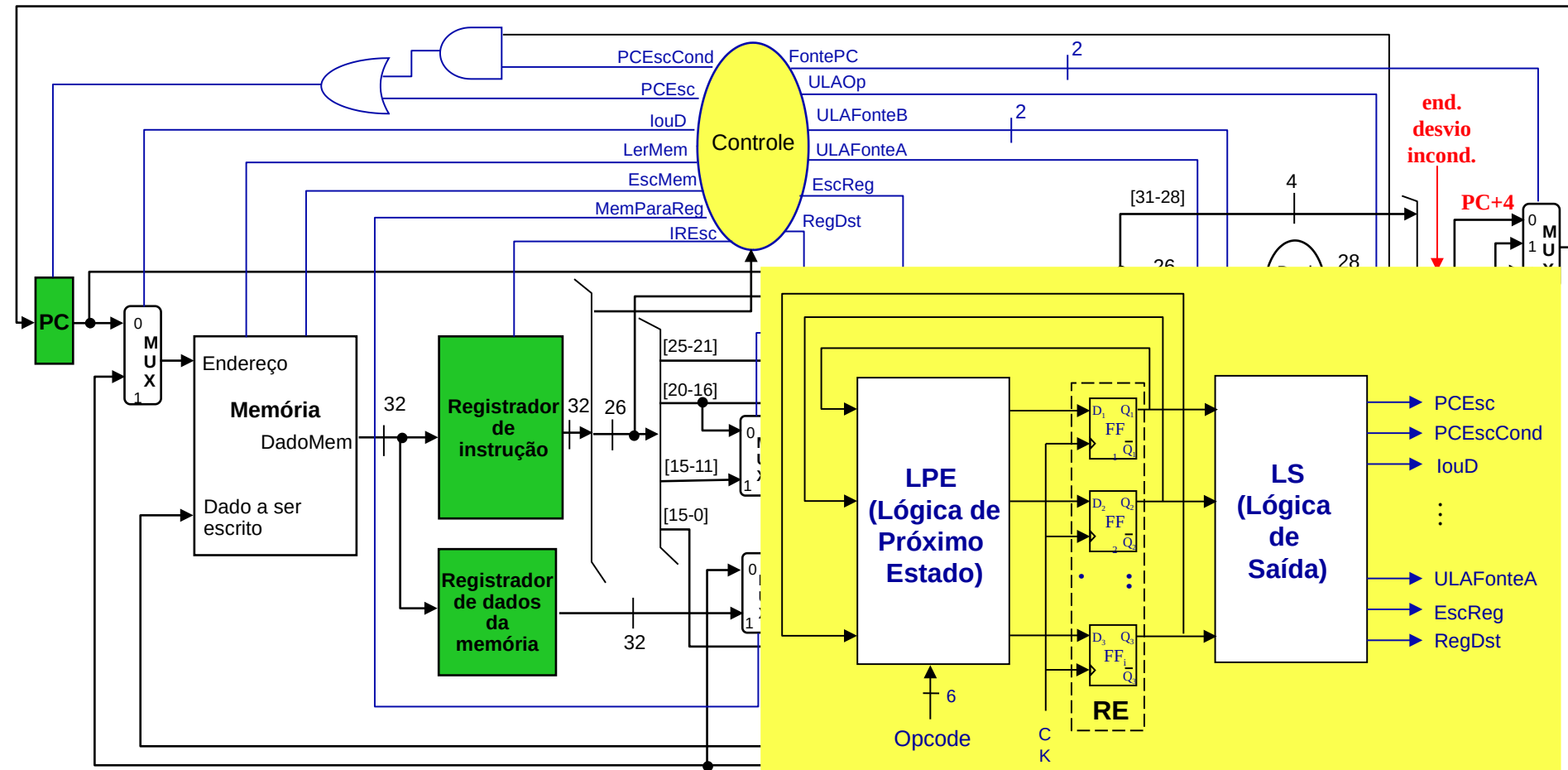
O Processador MIPS Multiciclo

Tempo de Estabilização (TE) dos Sinais de Controle



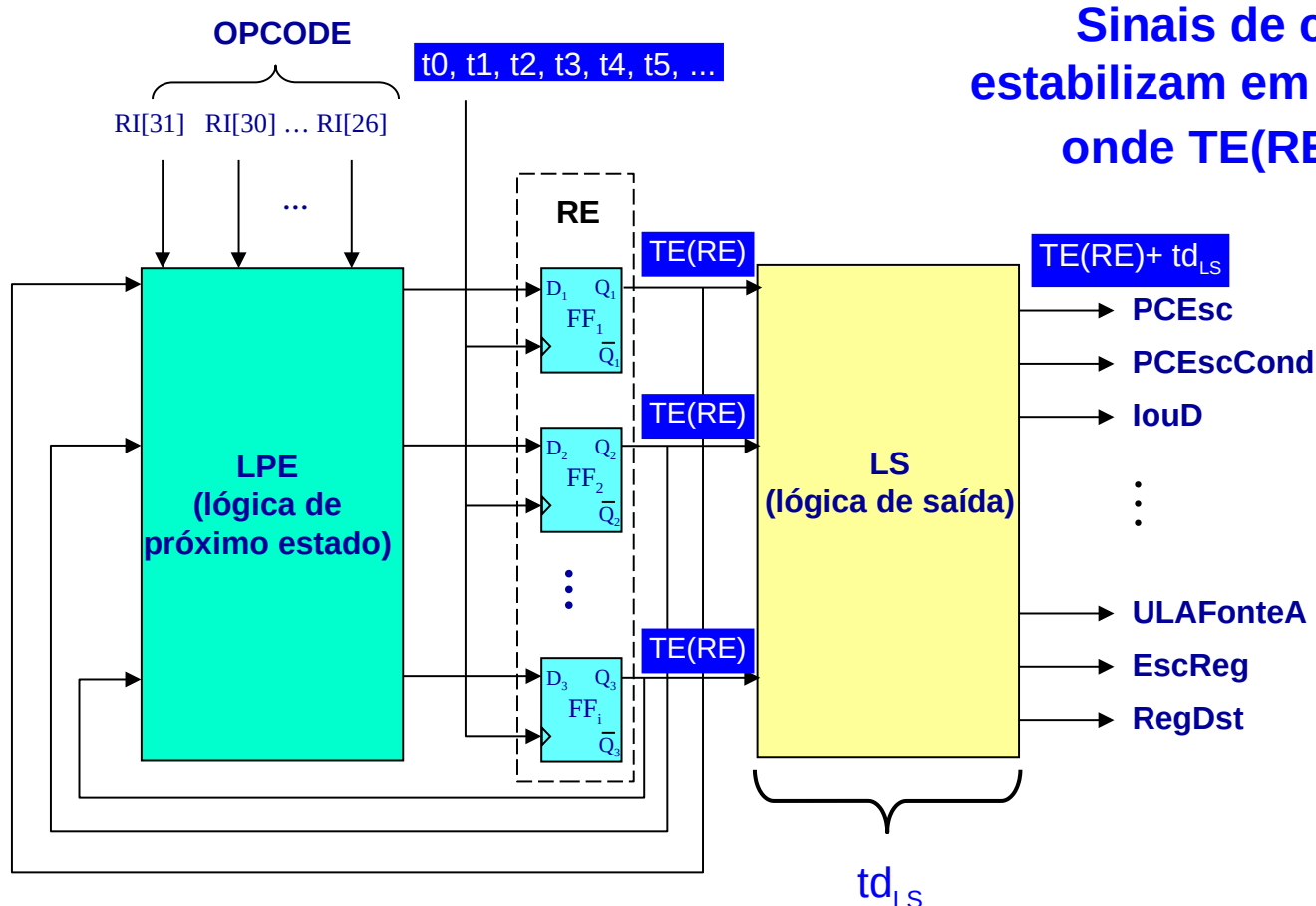
O Processador MIPS Multiciclo

Tempo de Estabilização (TE) dos Sinais de Controle



O Processador MIPS Multiciclo

Tempo de Estabilização (TE) dos Sinais de Controle (e dos Sinais de Próximo Estado...)

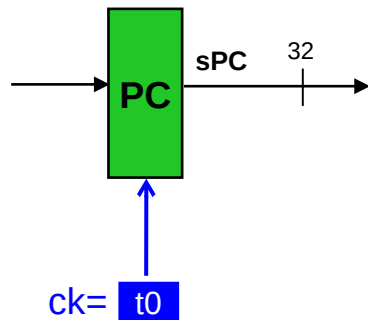


Sinais de comando
estabilizam em $TE(RE) + td_{LS}$
onde $TE(RE) = tco_{RE}$

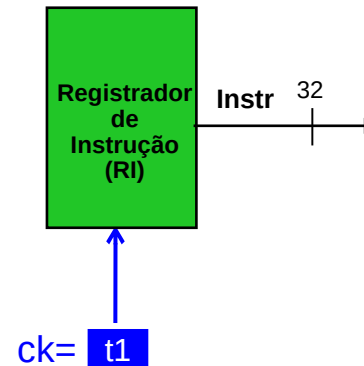
O Processador MIPS Multiciclo

Tempo de Estabilização (TE) no Nível RT

O Caso dos registradores temporários (PC, RI, RDM, A, B, ULASaída) e do registrador de estados (RE)



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



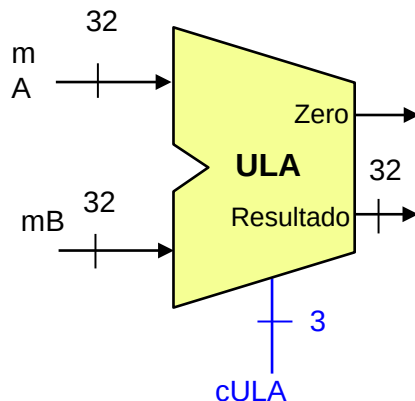
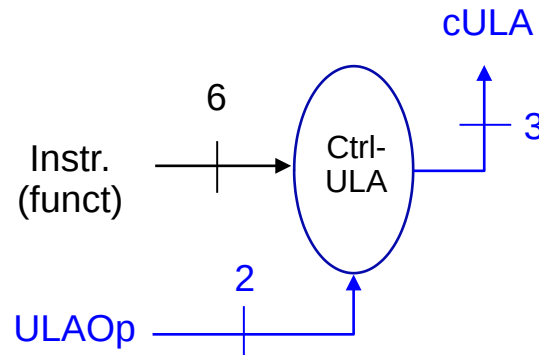
$$TE(Instr) = tco(RI) *$$

* Toma-se como referência a borda de relógio que inicia o ciclo do relógio e na qual o registrador é carregado (p. ex., t_0 , t_1 etc).

O Processador MIPS Multiciclo

Tempo de Estabilização (TE) no Nível RT Para Elementos Combinacionais (exceto muxes)

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



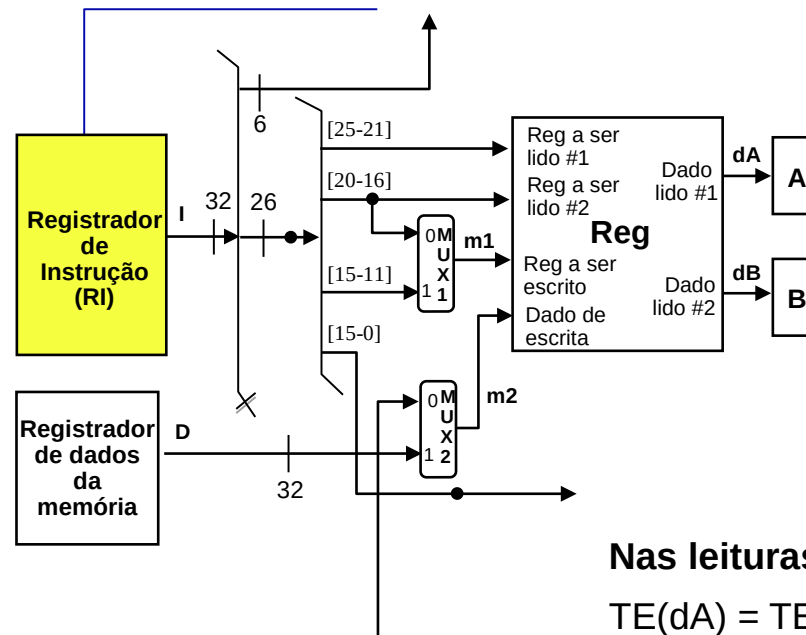
$$TE(Resultado) = \max \{ TE(mA), TE(mB), TE(cULA) \} + td_{ULA}$$

$$TE(Zero) = \max \{ TE(mA), TE(mB), TE(cULA) \} + td_{ULA}$$

O Processador MIPS Multiciclo

Tempo de Estabilização (TE) no Nível RT

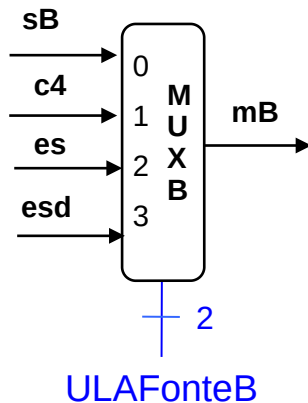
Leituras do Banco de Registradores (são assíncronas e portanto, ocorrem de maneira similar à operação de um bloco combinacional)



O Processador MIPS Multiciclo

Tempo de Estabilização (TE) no Nível RT

O Caso dos Muxes



$$TE(mB) = \max\{ TE_S, TE(\text{ULAFonteB}) \} + td_{\text{mux4-1}}$$

Onde:

$TE_S = TE(sB)$ se $sel = 00$

$TE_S = TE(c4)$ se $sel = 01$

$TE_S = TE(es)$ se $sel = 10$

$TE_S = TE(esd)$ se $sel = 11$

O Processador MIPS Multiciclo

Assumindo como Exemplo os Seguintes Atrasos

Componente	Característica	Símbolo	Valor
Memória	tempo para leitura	td_{LMEM}	300 ps
Memória	tempo para escrita (setup)	td_{EMEM}	300 ps
Banco de Registradores	tempo para leitura	td_{LREG}	70 ps
Banco de Registradores	tempo para escrita (setup)	td_{EREG}	70 ps
ULA	atraso para qualquer operação	td_{ULA}	60 ps
Qualquer mux 2:1	atraso	td_{mux}	2 ps
mux 3:1 e mux 4:1	atraso	td_{mux}	4 ps
PC, RI, RDM, A, B, ULASaída	tempo de setup	tsu	5 ps
PC, RI, RDM, A, B, ULASaída	tempo de carga	tco	5 ps
PC, RI, RDM, A, B, ULASaída	tempo de hold	th	Desprezível (0 ps)
RE (Registrador de Estados)	tempo de setup	tsu_{RE}	Desprezível (0 ps)
RE (Registrador de Estados)	tempo de carga	tco_{RE}	Desprezível (0 ps)*
RE (Registrador de Estados)	tempo de hold	th_{RE}	Desprezível (0 ps)
Deslocador, extensão de sinal, portas E, OU	atraso	-	Desprezível (0 ps)
Lógica de saída do controle	atraso	td_{LS}	Desprezível (0 ps)*
Lógica de próximo estado do controle,	atraso	td_{LPE}	Desprezível (0 ps)
Controle da ULA	atraso	td_{CULA}	Desprezível (0 ps)*

* isto significa que o atraso para gerar os sinais de controle é desprezível

O Processador MIPS Multiciclo

Exercicio 1:

- Elaborar em grupo de até 6 alunos um vídeo explicando o passo a passo do timing de um estado da FSM;
- O estado é determinado através do último dígito da matrícula. Ex: matrícula terminada em 1, o estado é 1;
- Indicar as diferenças entre o MIPS monociclo e o MIPS multiciclo quanto ao timing;
- Utilizar a tabela do slide 24 para os valores de tempo;

O Processador MIPS Multiciclo

Exercicio 2:

- Elaborar em grupo de até 6 alunos um vídeo apresentando uma ideia de como reduzir o período de relógio do MIPS multiciclo;
- Indicar quais são as vantagens e desvantagens da abordagem proposta;