

Fonte: <http://www.techspot.com/article/904-history-of-the-personal-computer-part-5/>

Datapath e Controle Multiciclo para MIPS

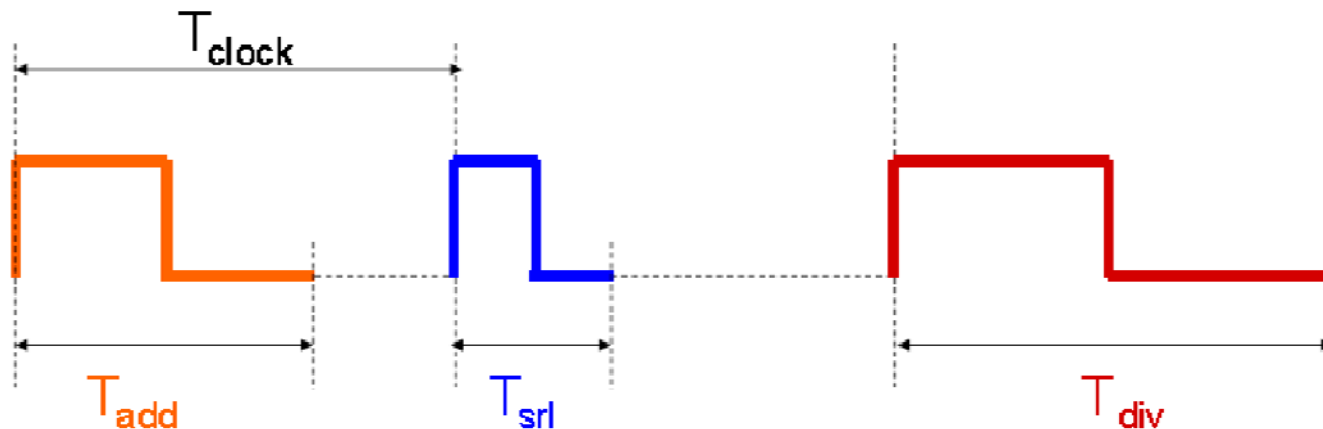
Luciano de Oliveira Neris

luciano@dc.ufscar.br

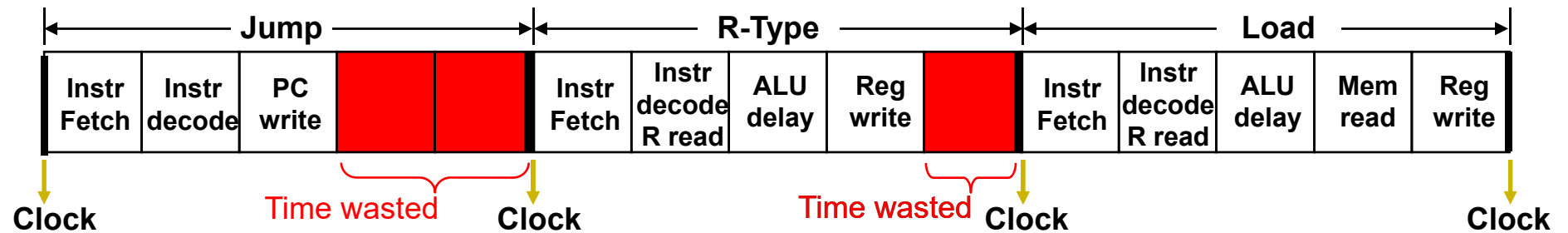
Adaptado de slides do prof. Marcio Merino Fernandes

Datapath Monociclo

- Cada instrução é executada em um 1 ciclo de clock
- Ciclo de clock deve ser longo o suficiente para executar a instrução mais longa
- Desvantagem: velocidade global limitada à velocidade da instrução mais lenta



Datapath Monociclo



Datapath Monociclo

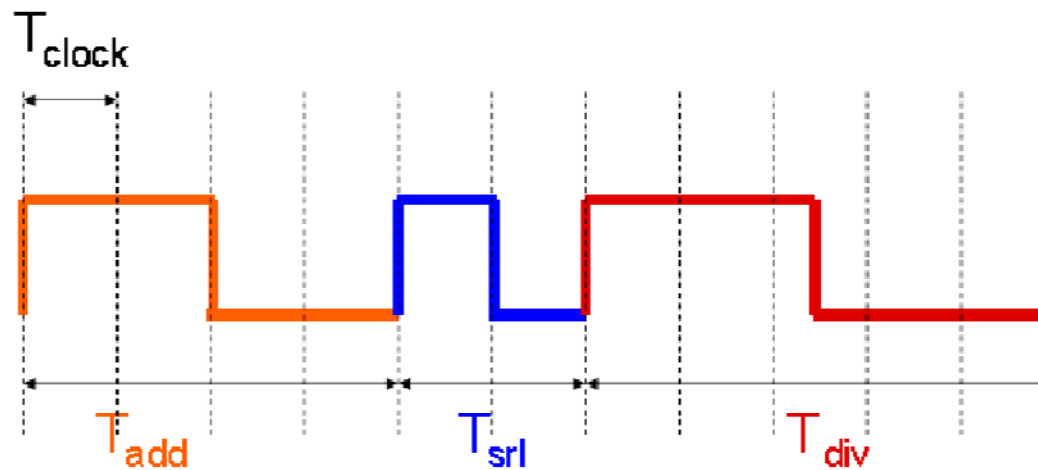
- Viola o princípio de “tornar o caso comum mais rápido”
- Datapaths monociclo não são mais utilizados em processadores modernos
- É mais eficiente executar cada instrução em um **número variável de ciclos mais rápidos**, utilizando apenas o necessário
- Este é o principio básico do **datapath multiciclo**

5

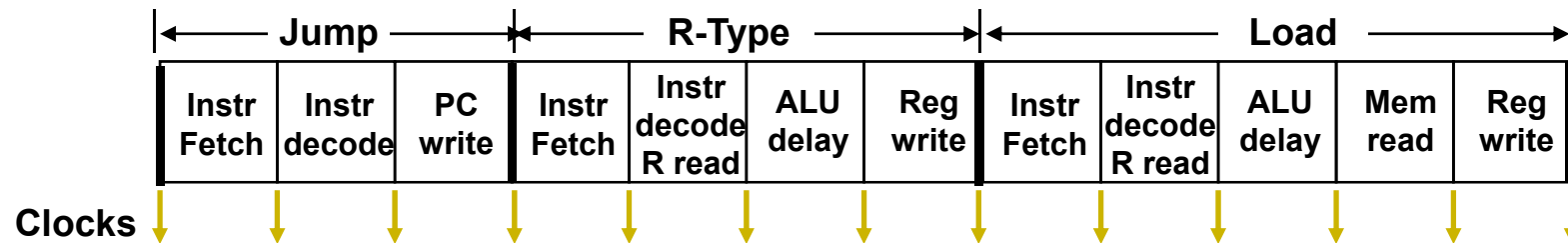
Multiciclo

Datapath Multi-ciclo

- Quebra o ciclo de execução em vários passos
- Executa cada passo em um ciclo de clock
- Cada instrução usa apenas o número de ciclos que ela necessita



Datapath Multi-ciclo



Datapath Multi-ciclo

Vantagens:

- Redução no tempo médio de execução de cada instrução
- Uma mesma unidade funcional pode ser utilizada em ciclos distintos de uma mesma instrução (ou seja, utilizada mais de uma vez).
 - ▣ * *Utilizar multiplexadores para determinar a origem dos dados*
- Pergunta: Como subdividir o datapath / instruções?

Datapath Multi-ciclo

- Esquema geral: partindo do datapath monociclo, **acrescentar registradores temporários** para armazenar valores entre as diversas unidades funcionais utilizadas por uma instrução.
 - ▣ Esses registradores são "invisíveis" ao programador
 - ▣ Sua função é evitar perda de sincronização nas transições de clock
- Dessa forma, subdivide-se o ciclo longo por uma **sequencia de ciclos mais curtos**

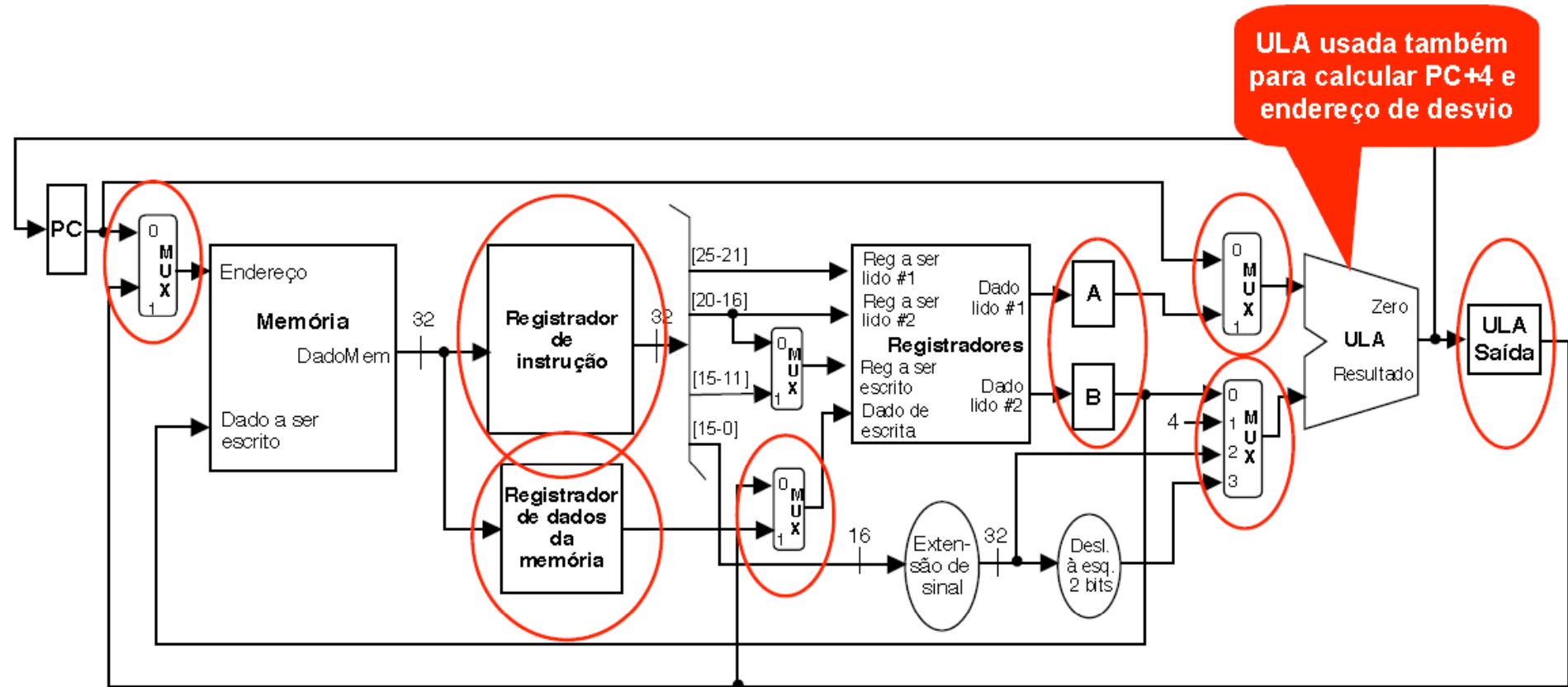
Datapath Multiciclo

- Uma **única** memória para instruções e dados
- Apenas uma ULA (dispensa o uso de somadores extras)
- Dados a serem usados na mesma instrução um ciclo de relógio posterior ficam armazenados nos registradores não-visíveis ao programador
- Dados a serem usados em outras instruções devem ser armazenados em elementos de memória visíveis ao programador (banco de registradores, PC ou memória)

Datapath Multiciclo

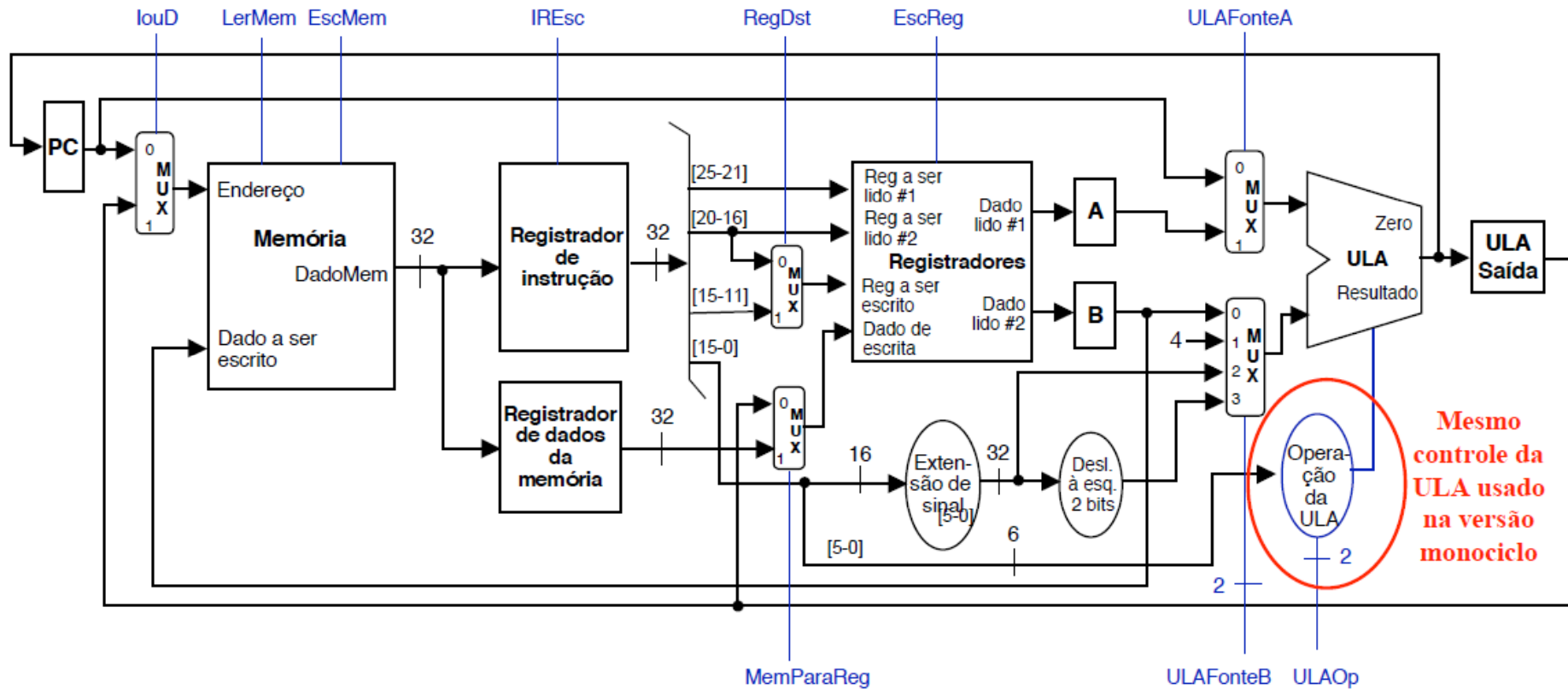
- **IR** (registrador de instrução) - usado para guardar a saída da memória para uma leitura de instrução
- **MDR** (registrador de dados da memória) - usado para guardar a saída da memória para uma leitura de dados
- **A e B** - usados para conter os valores dos registradores operandos lidos do banco de registradores
- **ALUOut** - contém a saída da ALU

Datapath Multiciclo



Datapath Multiciclo

□ Sinais de Controle



Datapath Multiciclo

- **Memória única:**

- ▣ Requer um **MUX** para selecionar se o endereço de acesso à memória vem de **PC** (instrução) ou de **SaídaALU** (dados)

- **ALU única:**

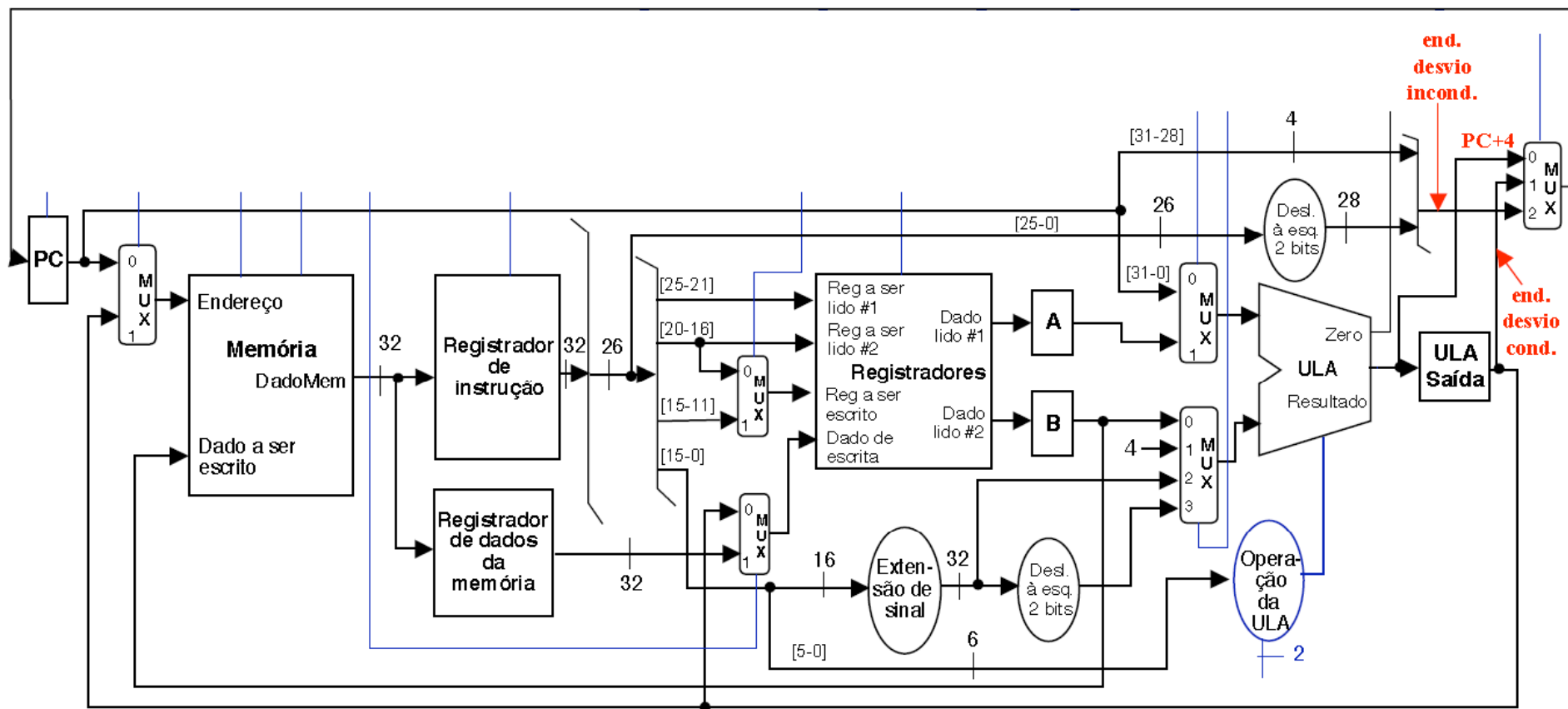
- ▣ Um **MUX** adicional é incluído na **primeira entrada** para escolher entre o registrador **A** ou o **PC**
- ▣ O **MUX** da **segunda entrada** da ALU é **expandido** para quatro entradas, a fim de poder **selecionar** a **constante 4** (incremento do PC) e o **campo offset** estendido e deslocado (desvios)

Datapath Multiciclo

- Três origens para o valor de PC:
 - ▣ Saída da ALU ($PC + 4$) (Entrada 0): este valor sempre será armazenado no PC
 - ▣ Registrador ULA Saída, onde é armazenado o endereço de desvio, após ele ser calculado (Entrada 1): este registrador armazena o endereço-alvo do desvio condicional, após este ter sido calculado pela ULA (beq)
 - ▣ 26 bits menos significativos do IR deslocados de 2 à esquerda e concatenados com os 4 bits mais significativos de $PC+4$, no caso de jumps (Entrada 2)

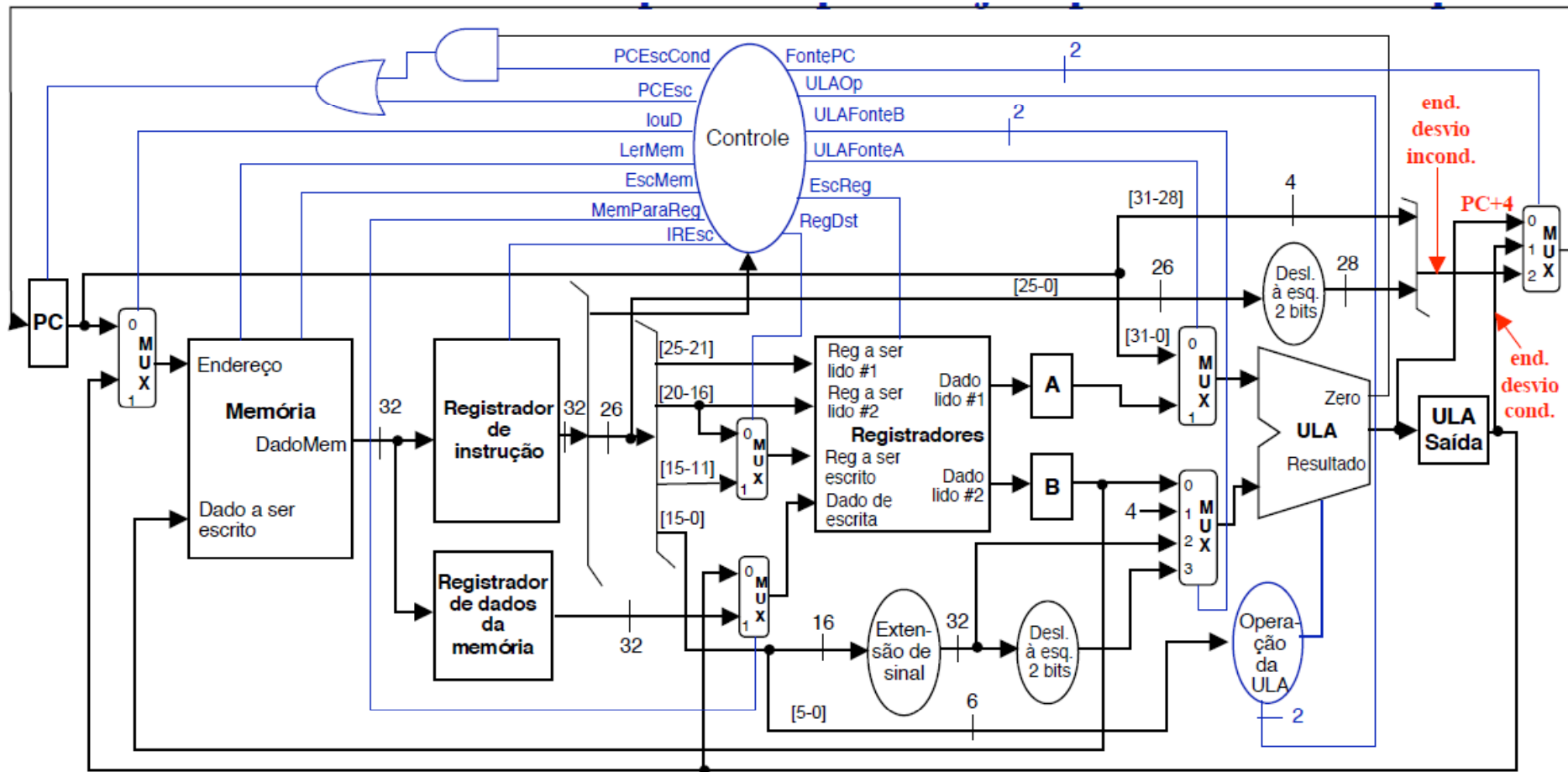
Datapath Multiciclo

- Três origens para o valor de PC:



Datapath Multiciclo

□ Sinais de Controle



Datapath Multiciclo

- Passos (etapas) das Instruções
 - ▣ 1. Busca da instrução
 - ▣ 2. Decodificação da instrução
 - Leitura dos registradores – mesmo que não sejam utilizados
 - Cálculo do endereço do branch – mesmo que instrução não seja branch
 - ▣ 3. Execução da operação
 - Instruções tipo R
 - Cálculo do endereço efetivo do operando – instruções load e store
 - Determinar se branch deve ser executado – instrução branch
 - ▣ 4. Acesso à memória
 - Instruções load e store
 - Escrita de registrador – instruções tipo R
 - ▣ 5. Escrita de registrador
 - Instrução load

Datapath Multiciclo

□ Execução de Instruções

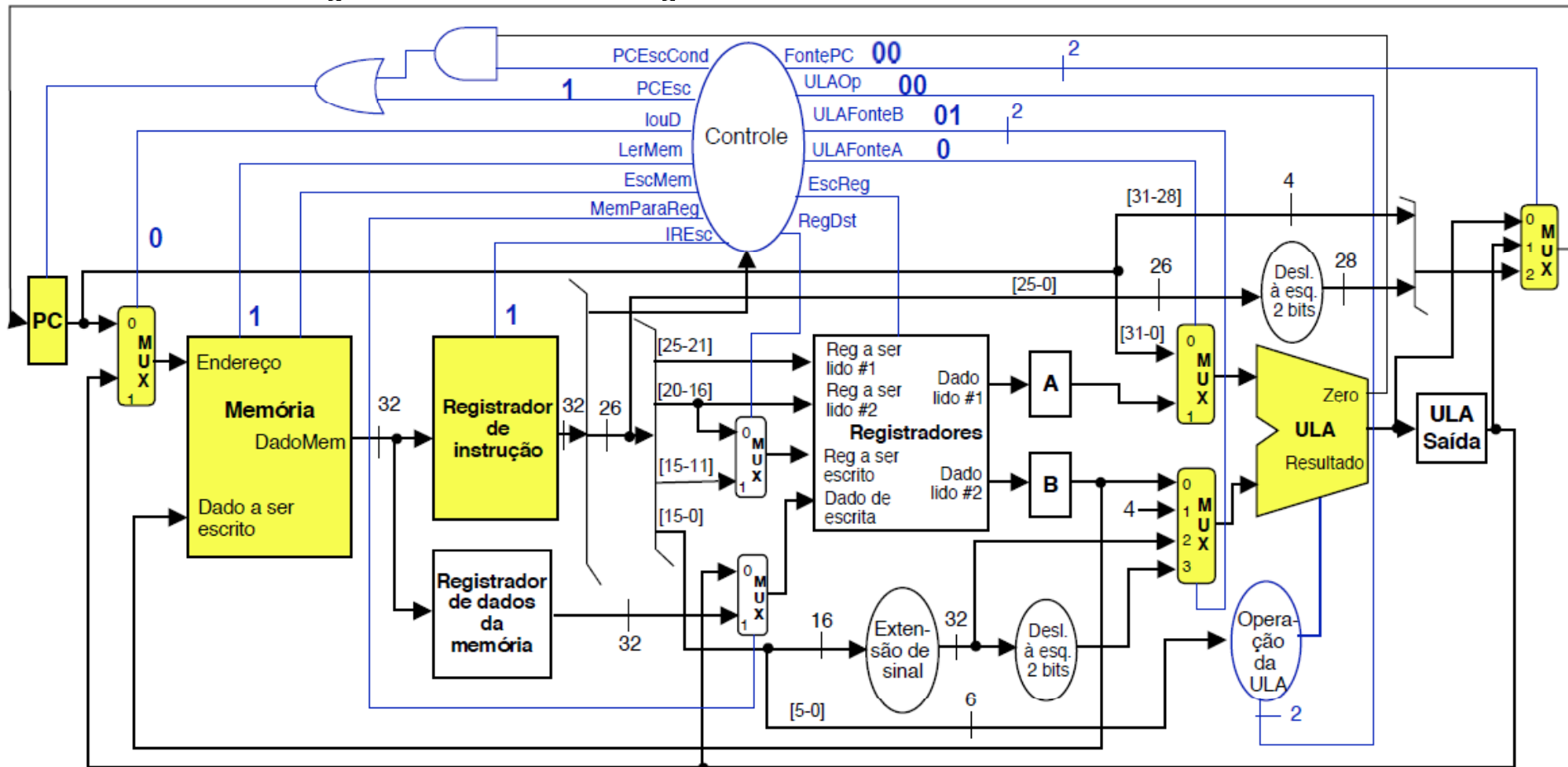
1. Busca da Instrução (e incremento do PC)

$$RI = \text{Mem}[PC];$$
$$PC = PC + 4;$$

Estas operações ocorrem em paralelo

Datapath Multiciclo

□ Execução de Instruções: busca



Datapath Multiciclo

□ Execução de Instruções

2. Decodificação (Geração dos Sinais de Controle) e
Leitura de Rs e Rt

$A = \text{Reg}[\text{RI}[25-21]];$

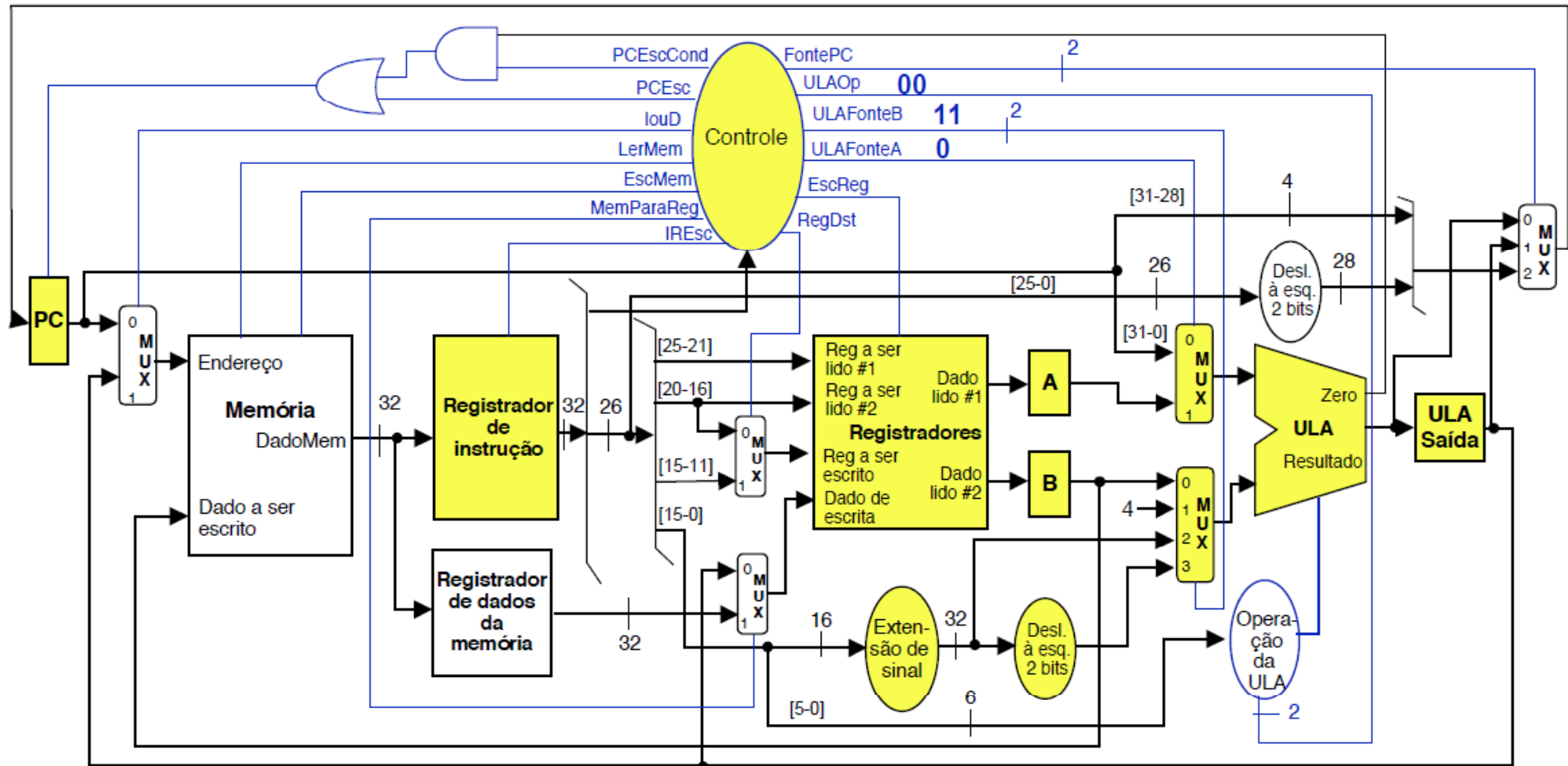
$B = \text{Reg}[\text{RI}[20-16]];$

$\text{ULASaída} = \text{PC} + (\text{extensão de sinal}(\text{RI}[15-0]) \ll 2)$

Estas operações ocorrem em paralelo

Datapath Multiciclo

□ Execução de Instruções: leit. Rs e Rt e decodificação



Datapath Multiciclo

□ Execução de Instruções

3. Execução da Instrução:

- Referência à memória (lw e sw)

$ULASaída = A + \text{extensão de sinal}(RI[15-0]);$

- Tipo R

$ULASaída = A \text{ op } B;$

- Desvio Condicional

Se $(A == B)$ então $PC = ULASaída;$

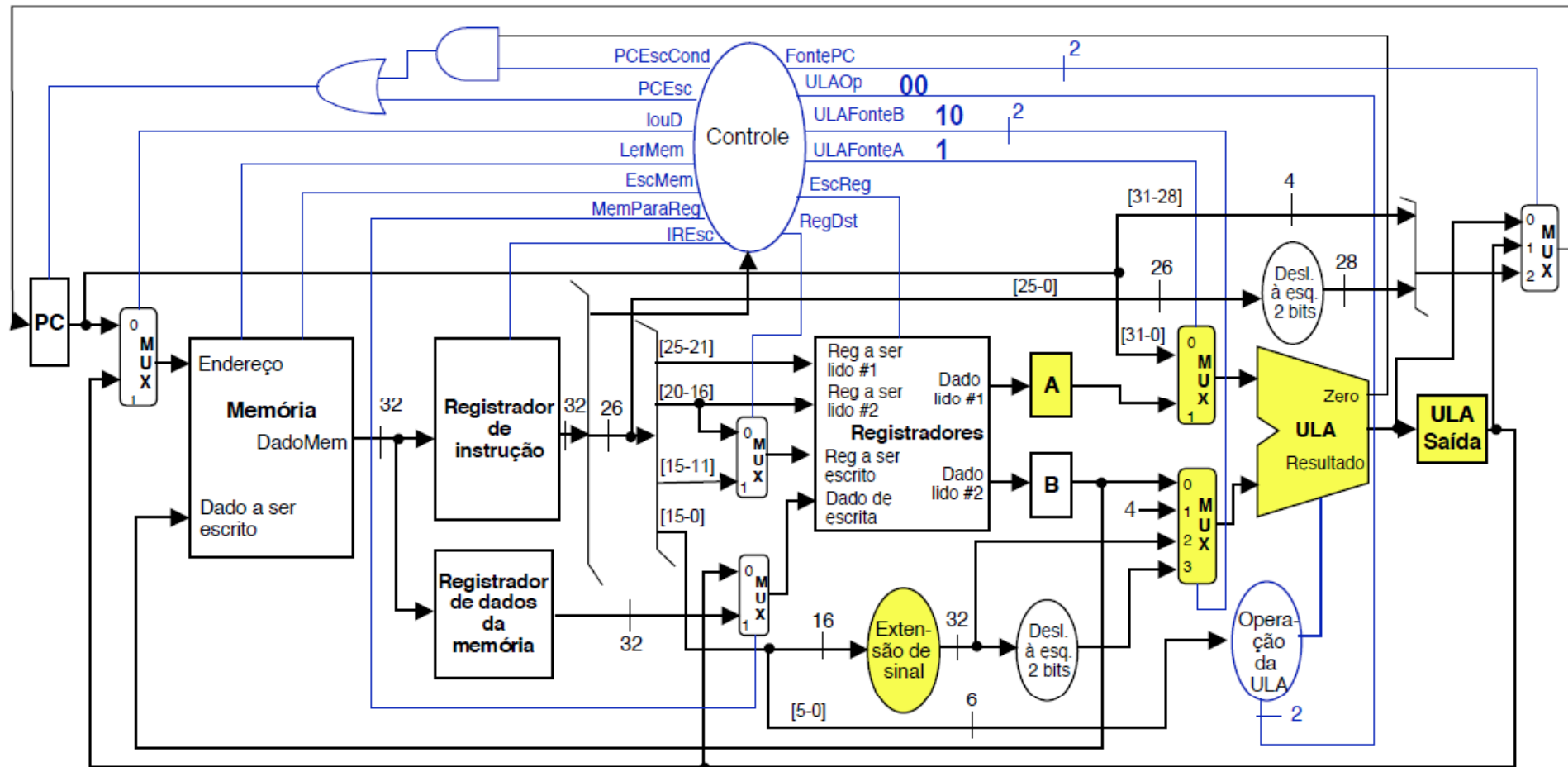
(usa a ULA para fazer $A-B$, porém sem escrever em $ULASaída$)

- Desvio Incondicional

$PC = PC[31-28] || (RI[25-0] \ll 2);$

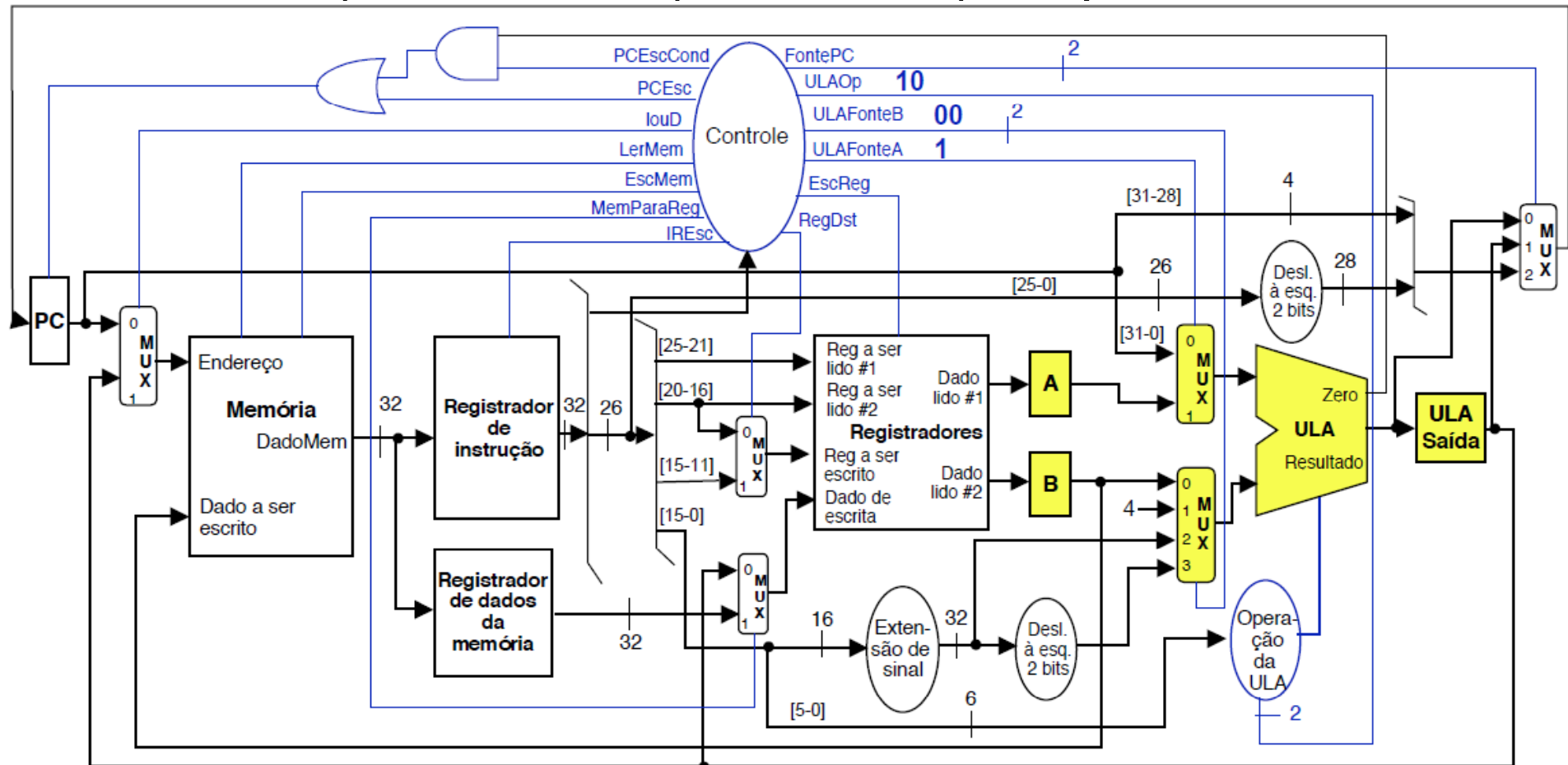
Datapath Multiciclo

□ Execução de Instruções: execução lw/sw



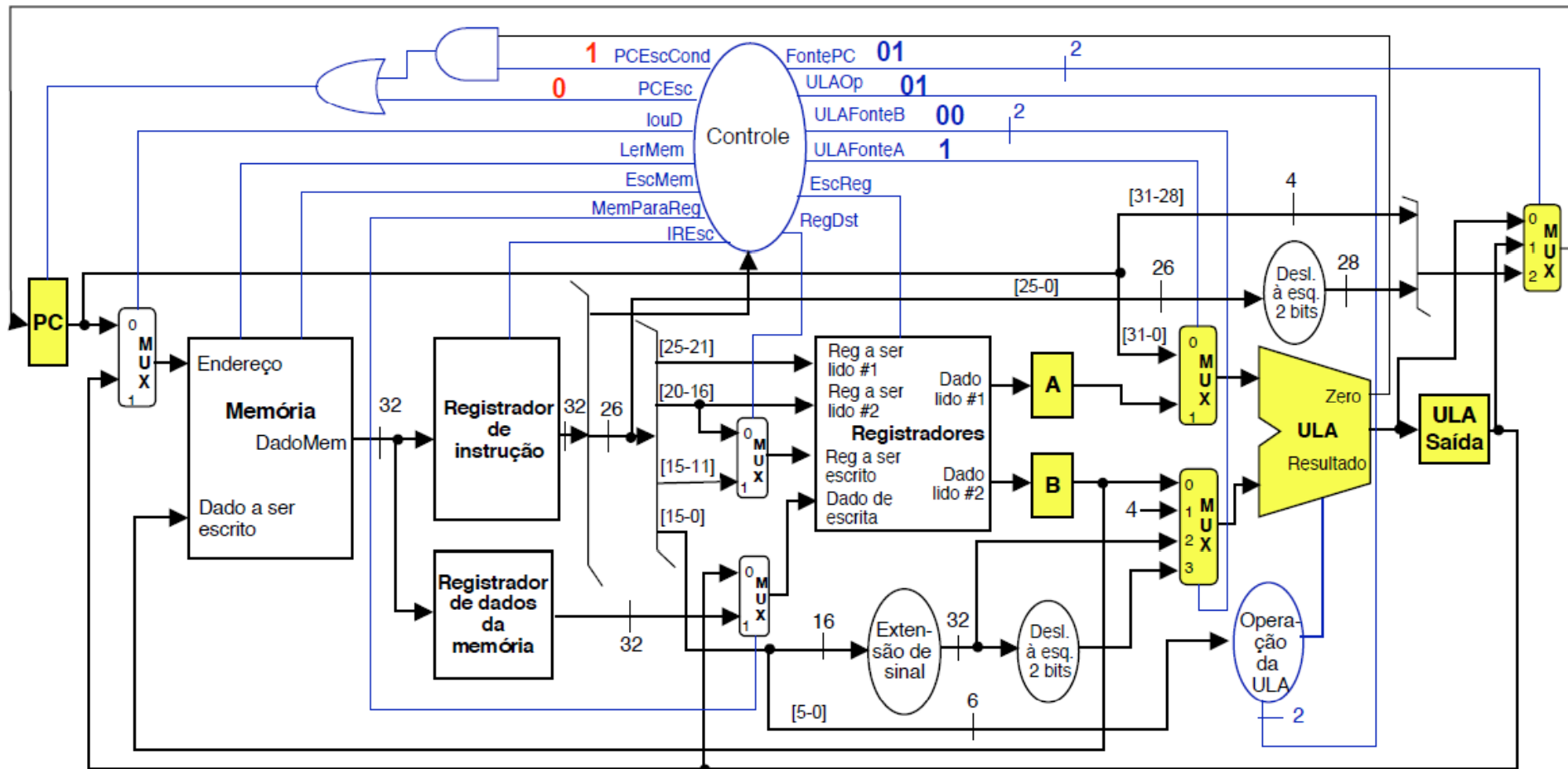
Datapath Multiciclo

□ Execução de Instruções: execução tipo R



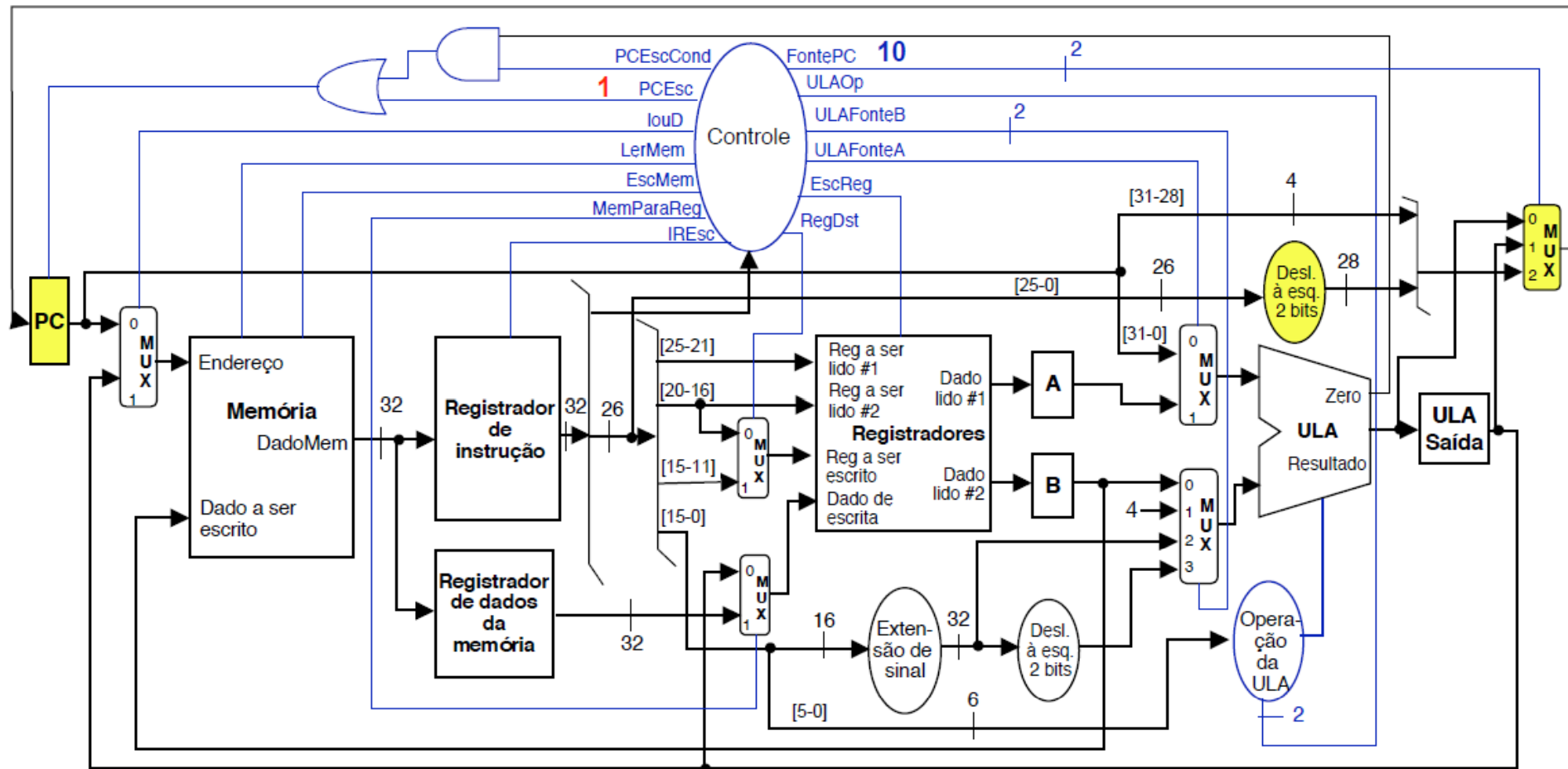
Datapath Multiciclo

□ Execução de Instruções: execução beq



Datapath Multiciclo

□ Execução de Instruções: execução jump



Datapath Multiciclo

□ Execução de Instruções

4. Final da Execução de sw e Tipo R:

- lw:

RDM = Mem[ULASaída];

- sw:

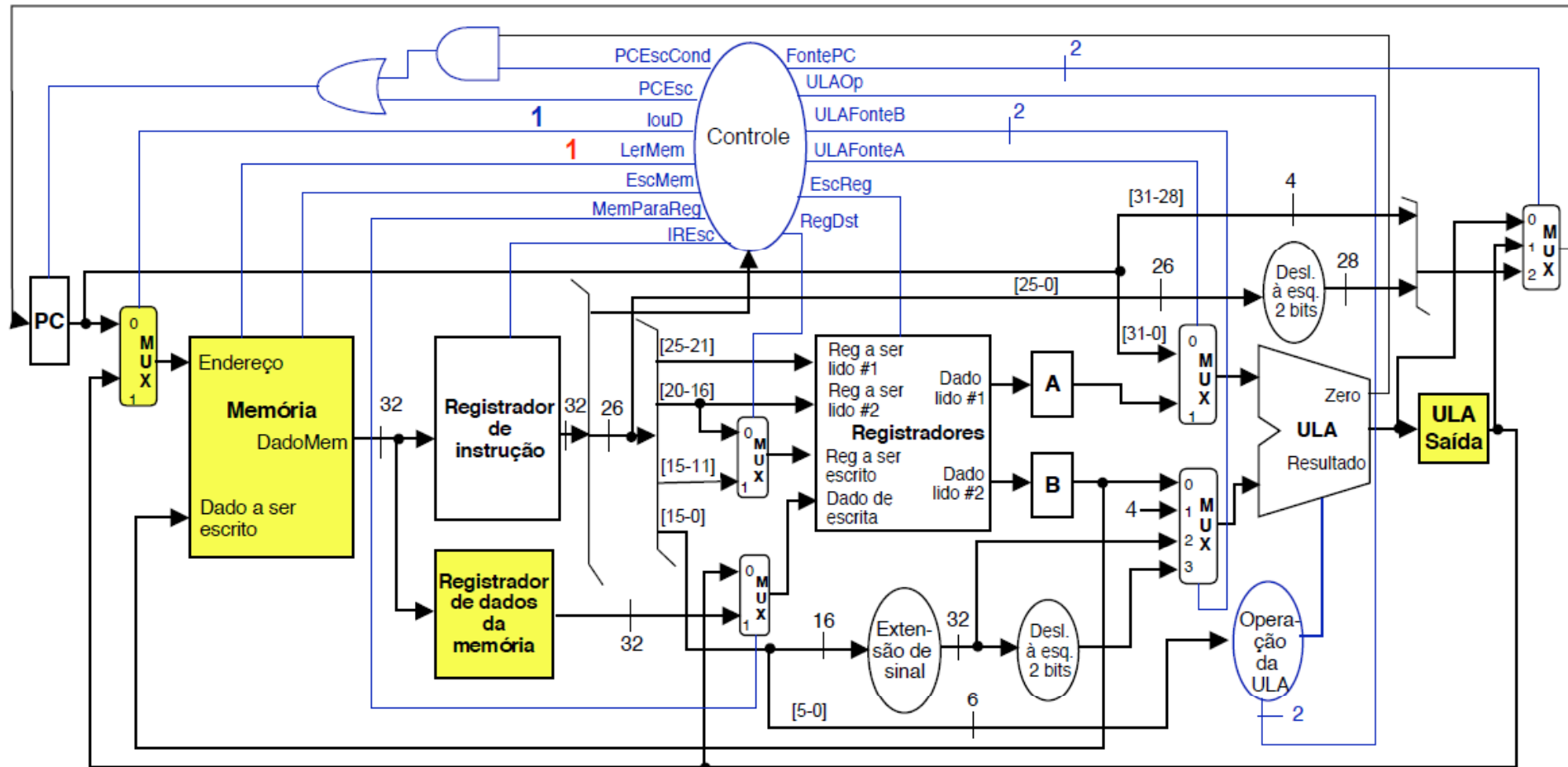
Mem[ULASaída] = B;

- Tipo R:

Reg[RI[15-11]] = ULASaída;

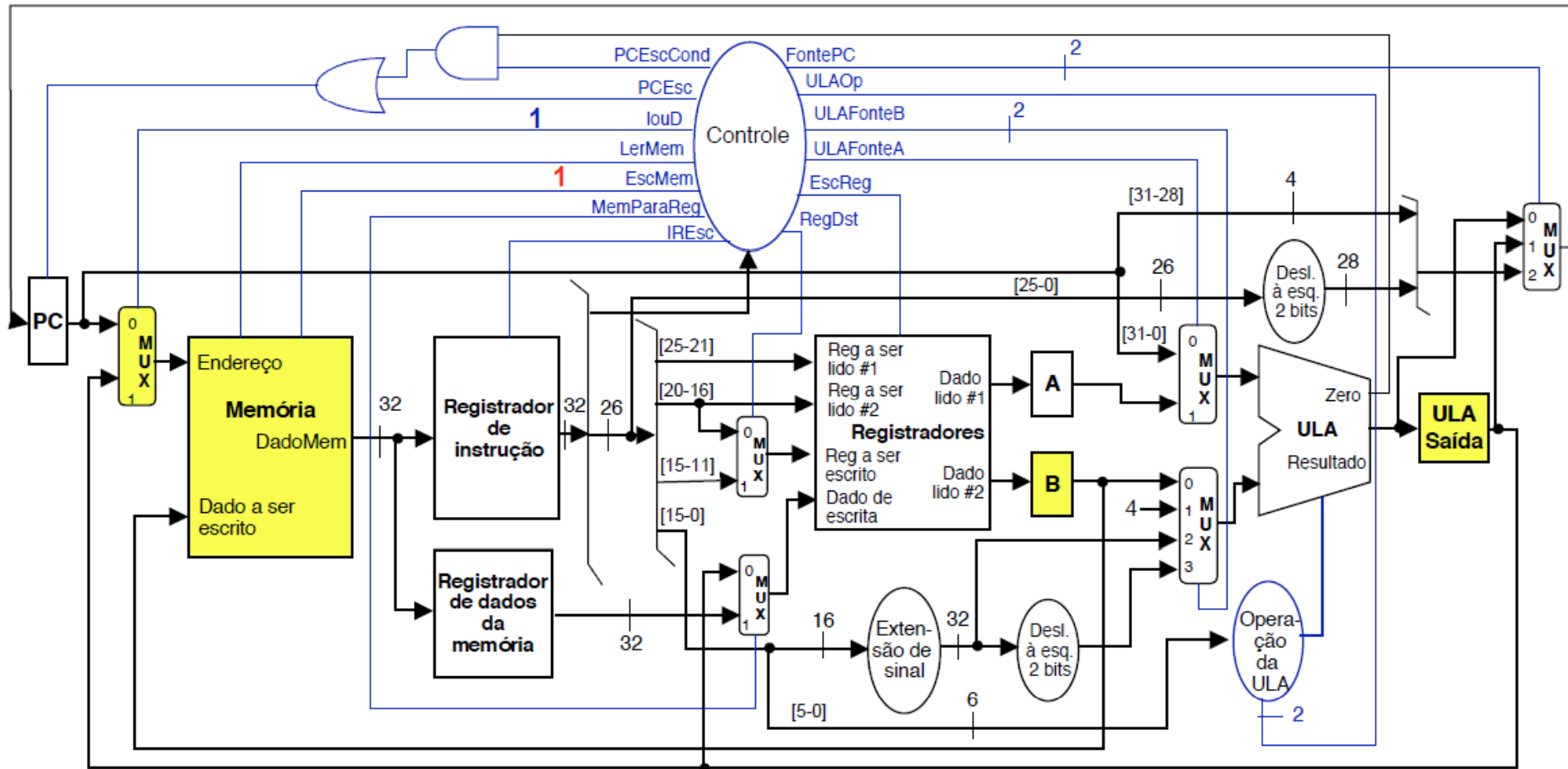
Datapath Multiciclo

□ Execução de Instruções: lw lê dado da memória



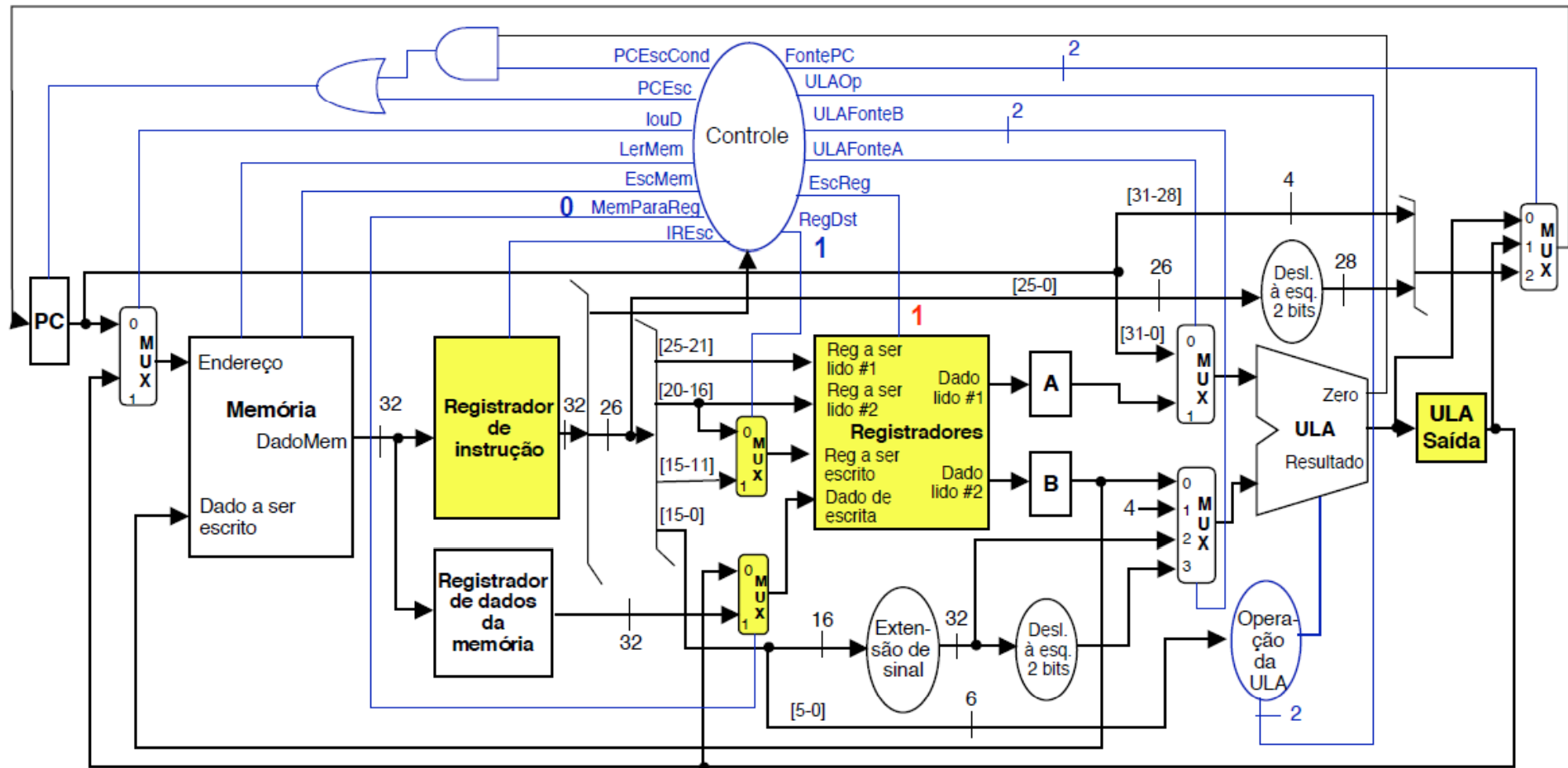
Datapath Multiciclo

□ Execução de Instruções: conclui sw



Datapath Multiciclo

□ Execução de Instruções: conclui tipo R



Datapath Multiciclo

□ Execução de Instruções

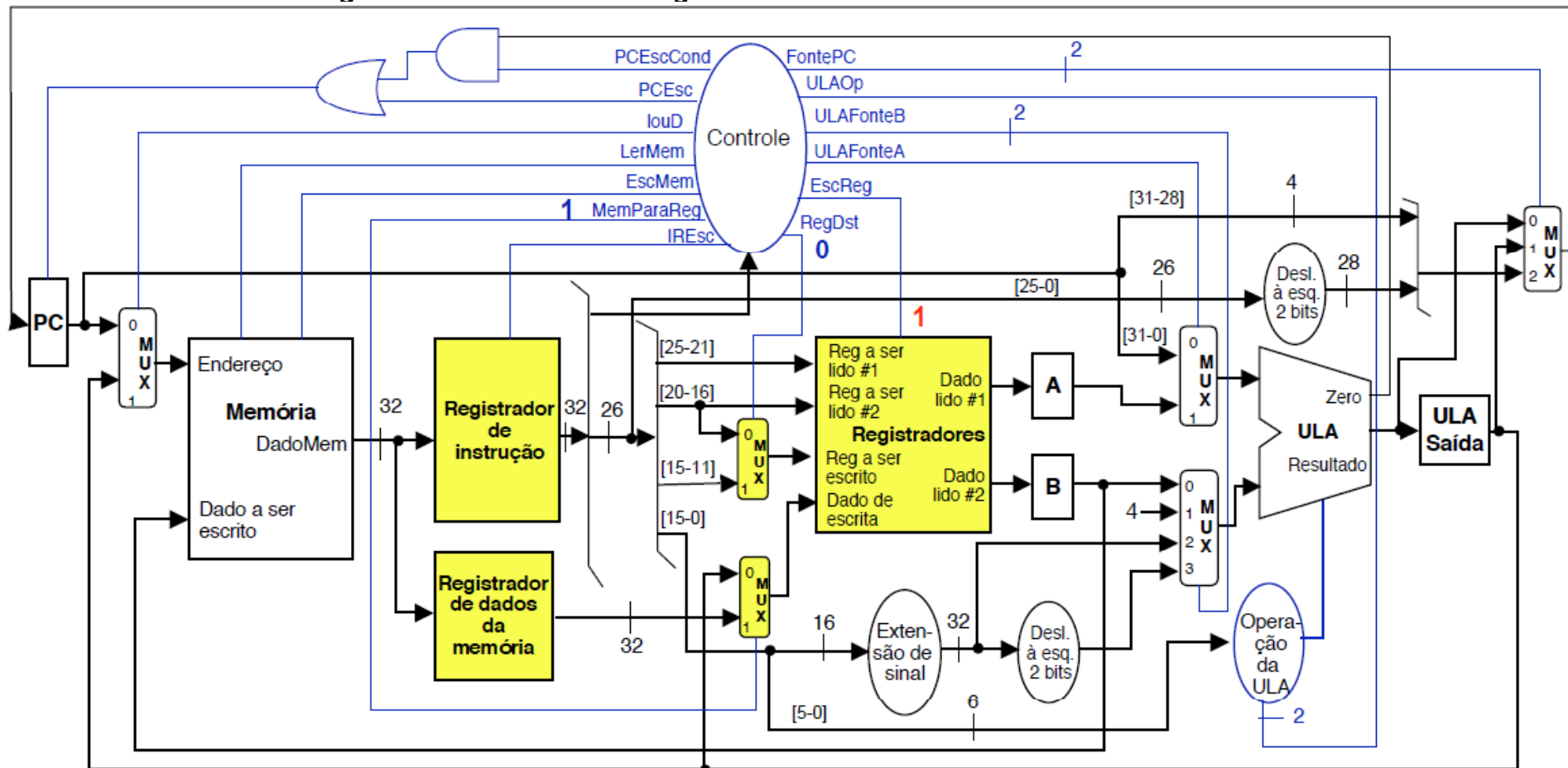
5. Final da Execução de lw:

■ lw:

$\text{Reg[RI[20-16]]} = \text{RDM};$

Datapath Multiciclo

□ Execução de Instruções: conclui lw



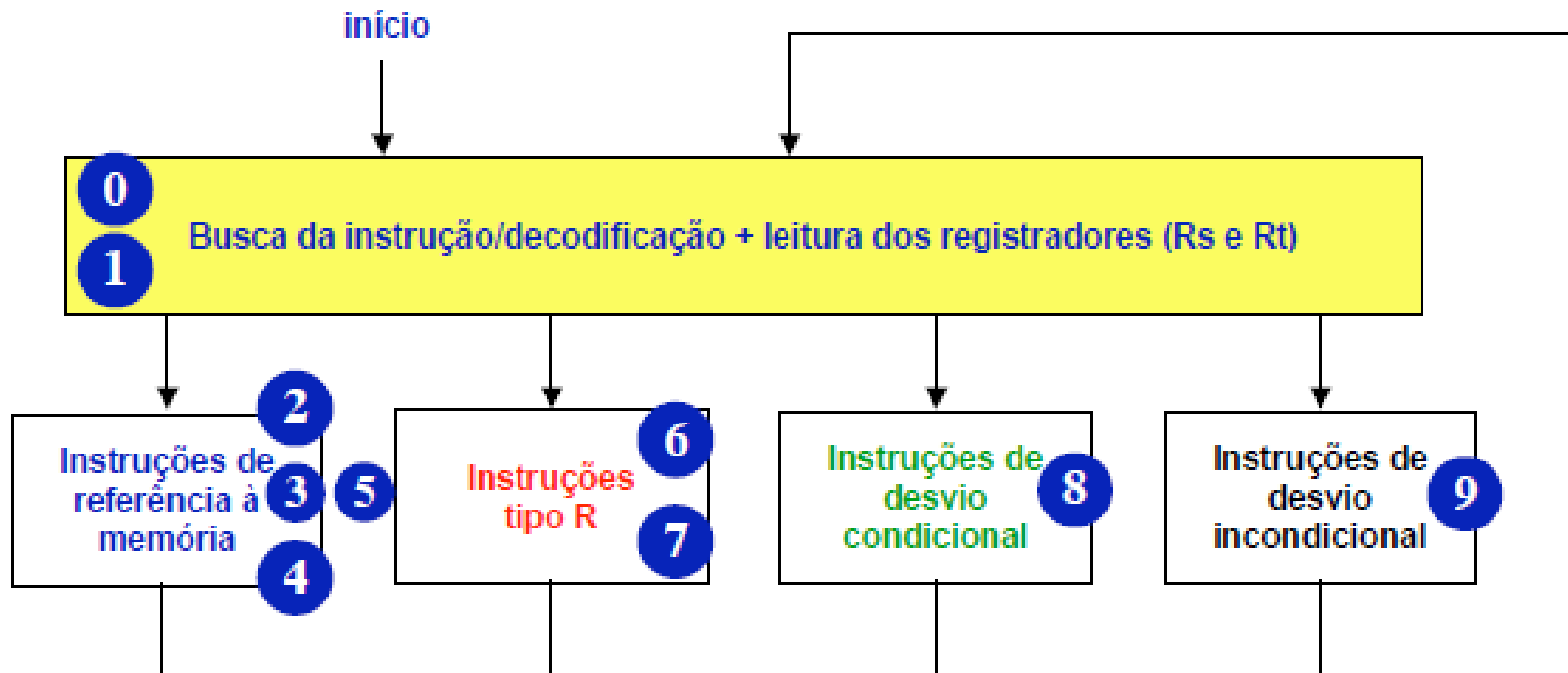
Datapath Multiciclo

□ Passos Necessários para Cada Instrução

Nome do passo	Instrução tipo R	Instrução lw	Instrução sw	Instrução de desvio condicional	Instrução de desvio incondicional
Busca da instrução	$RI = Mem[PC]$ $PC = PC + 4$				
Decodificação da instrução & leitura dos registradores Rs e Rt & cálculo do endereço de desvio (cond.)	$A = Reg [RI[25-21]]$ $B = Reg [RI[20-16]]$ $ULASaída = PC + (extensão\ de\ sinal(RI[15-0]) \ll 2)$				
Execução, cálculo do endereço de acesso à memória, término de uma instrução branch/jump	$ULAOp = A\ op\ B$	$ULASaída = A + extensão\ de\ sinal(RI[15-0])$		Se $(A == B)$ então $PC = ULASaída$	$PC = PC[31-28] \parallel (RI[25-0] \ll 2)$
Término de uma instrução store word ou de tipo R	$Reg [RI[15-11]] = ULASaída$	$RDM = Mem [ULASaída]$	$Mem [ULASaída] = B$		
Término de uma instrução load word		$Reg[RI[20-16]] = RDM$			
Número de passos	4	5	4	3	3

Datapath Multiciclo

□ Máquina de Estados



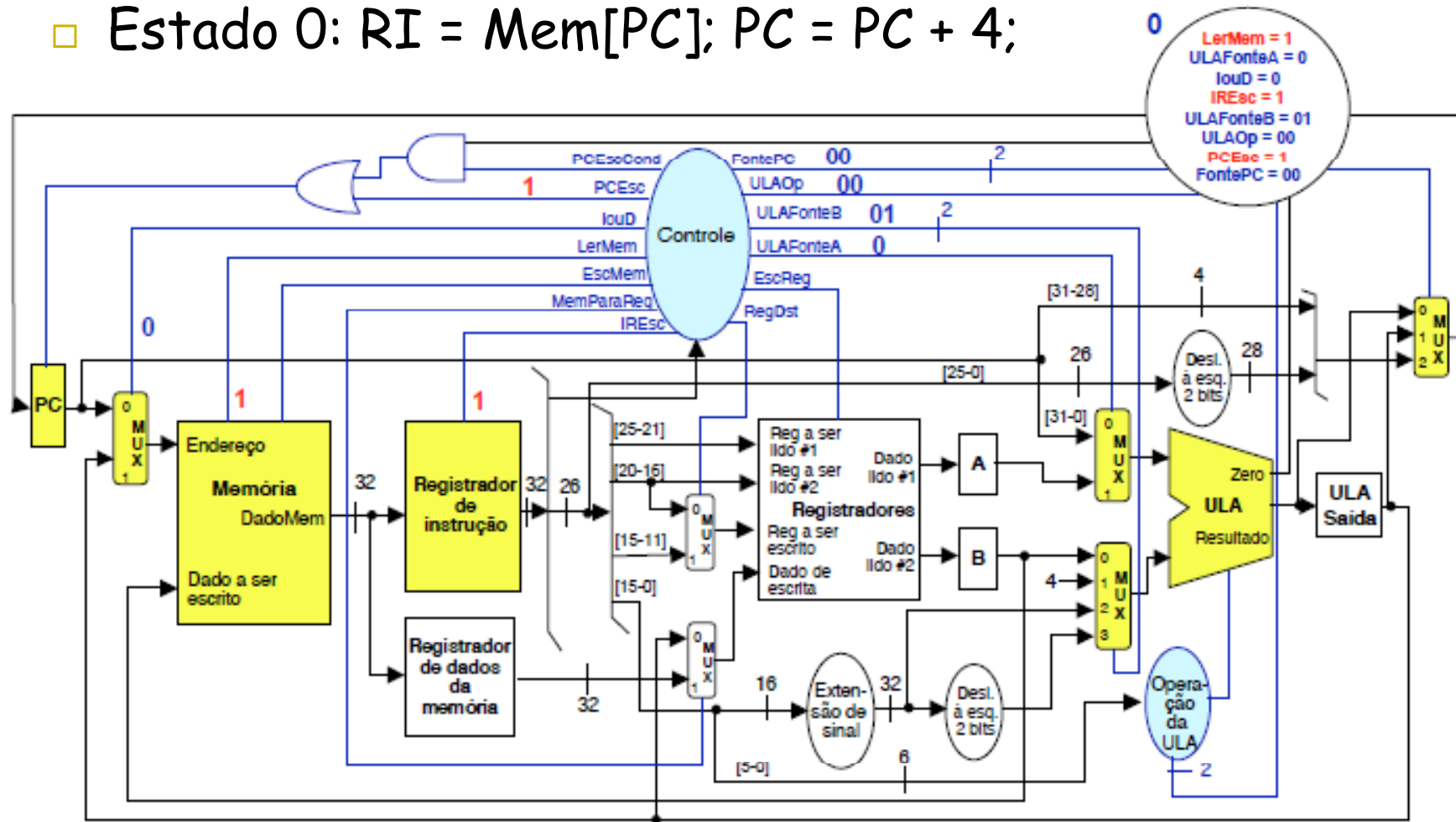
Datapath Multiciclo

□ Passos Necessários para Cada Instrução

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

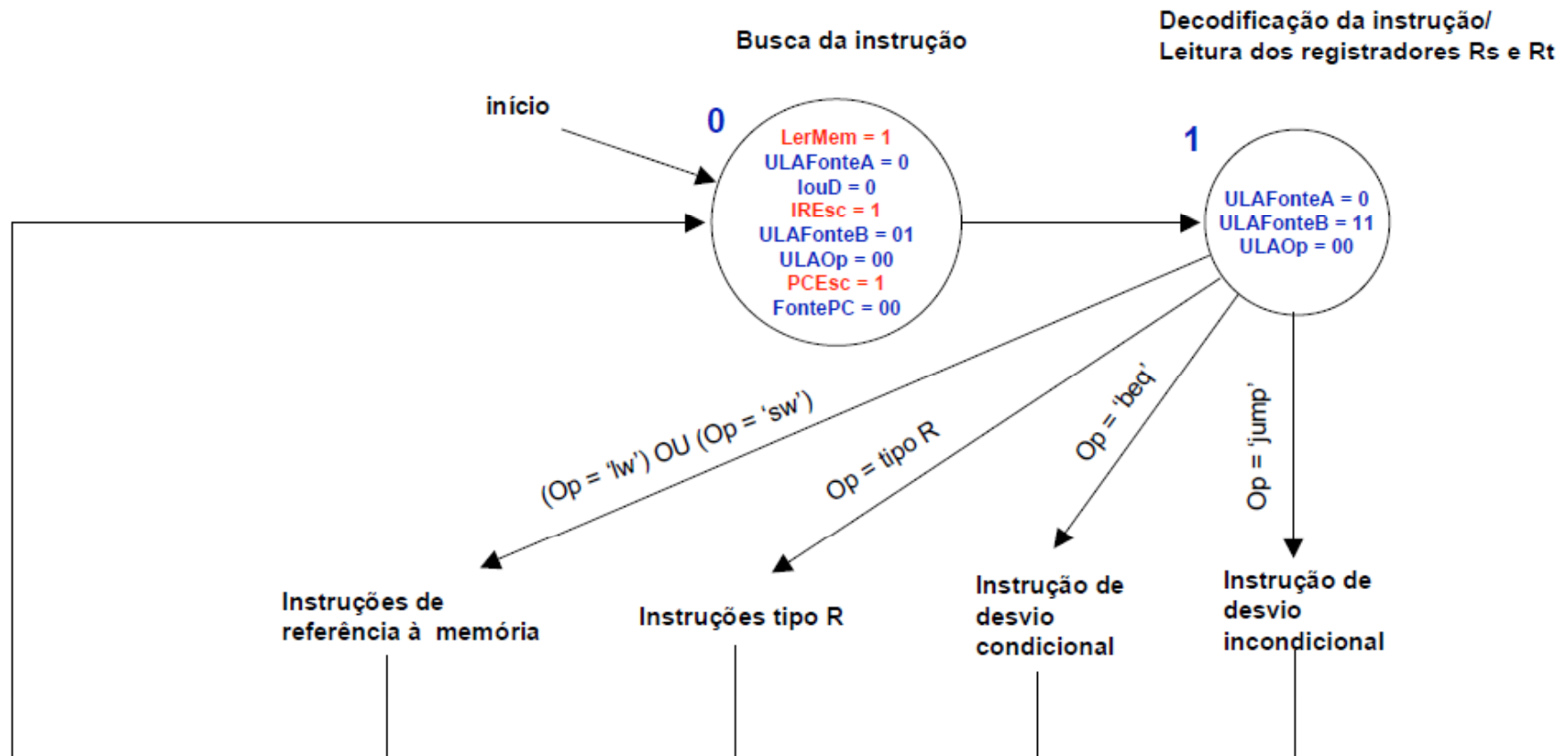
Datapath Multiciclo

- Estado 0: $RI = Mem[PC]; PC = PC + 4;$



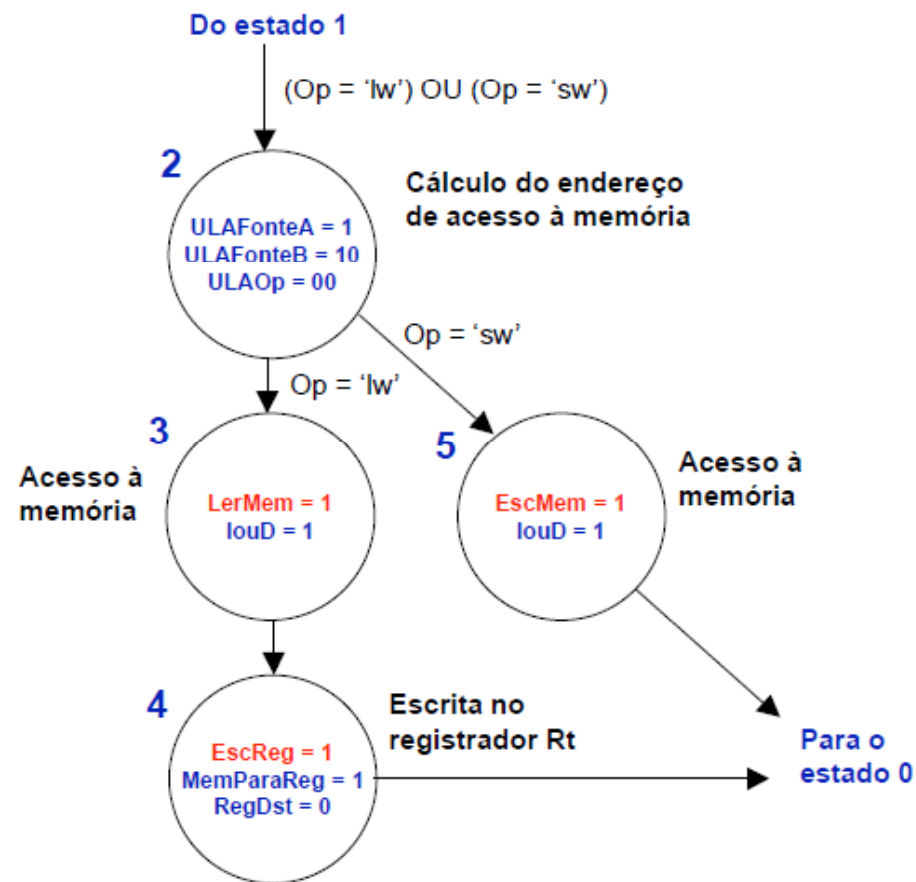
Datapath Multiciclo

□ Máquina de Estados: busca e decodificação



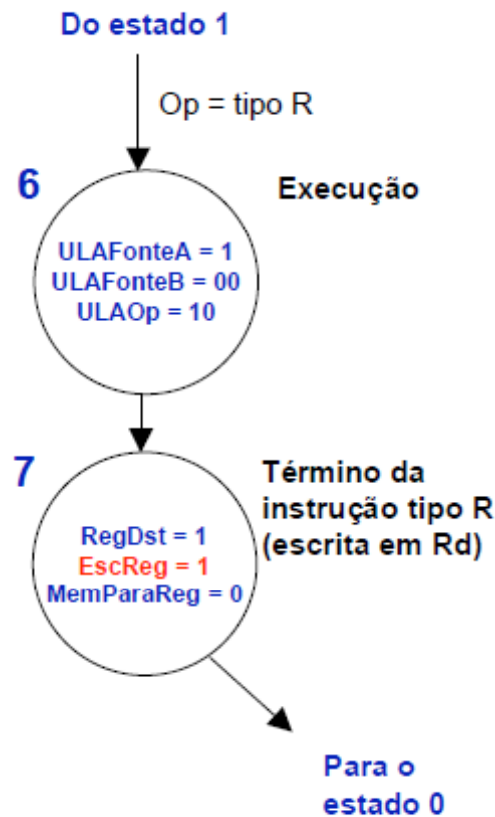
Datapath Multiciclo

- Máquina de Estados: execução de instruções lw/sw



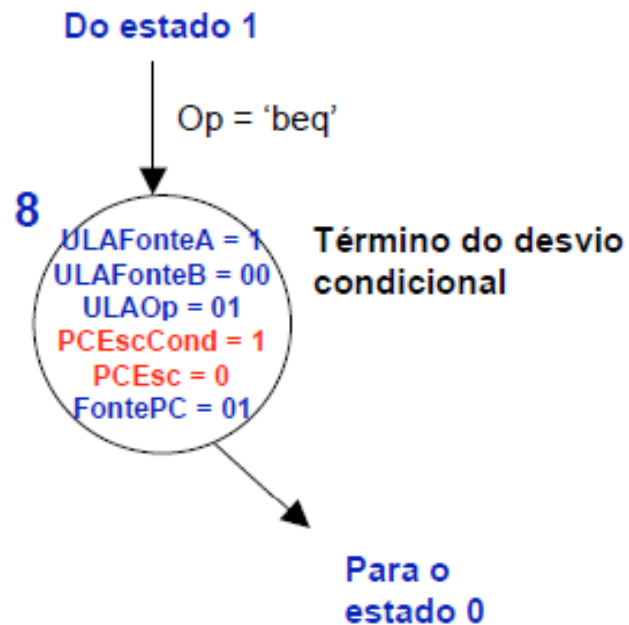
Datapath Multiciclo

- Máquina de Estados: execução de instrução tipo R



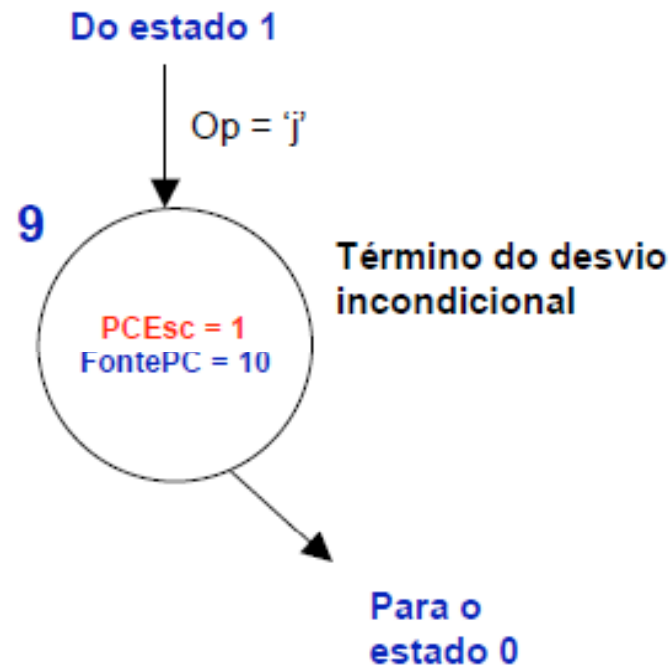
Datapath Multiciclo

- Máquina de Estados: execução de instrução beq



Datapath Multiciclo

- Máquina de Estados: execução de instrução jump



Datapath Multiciclo

□ Máquina de Estados

