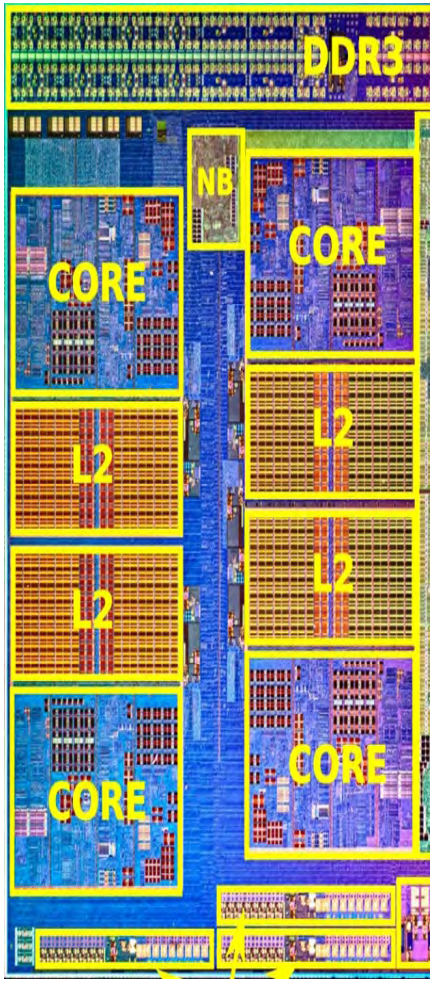
0273359 – Arquitetura e Organização de Computadores 1



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

Datapath e Controle Multiciclo para MIPS

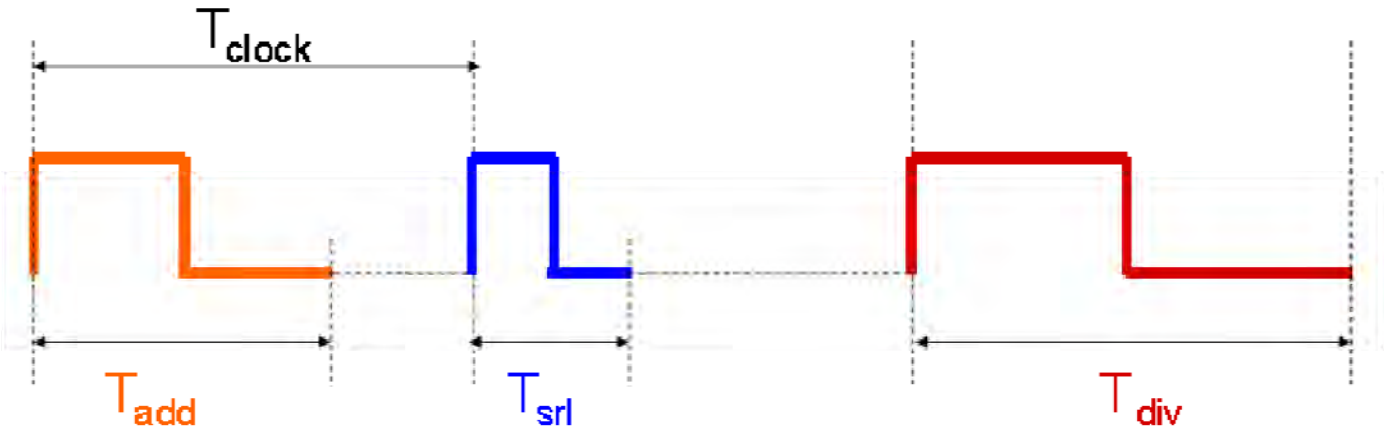
Luciano de Oliveira Neris
luciano@dc.ufscar.br

Adaptado de slides do prof. Marcio Merino Fernandes

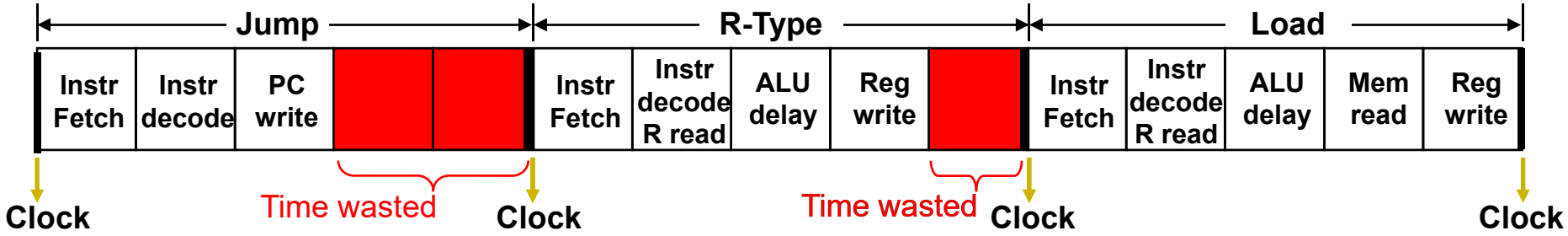
Departamento de Computação
Universidade Federal de São Carlos

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



Fonte: Prof. Savio chau: CSM151B

Datapath Monociclo

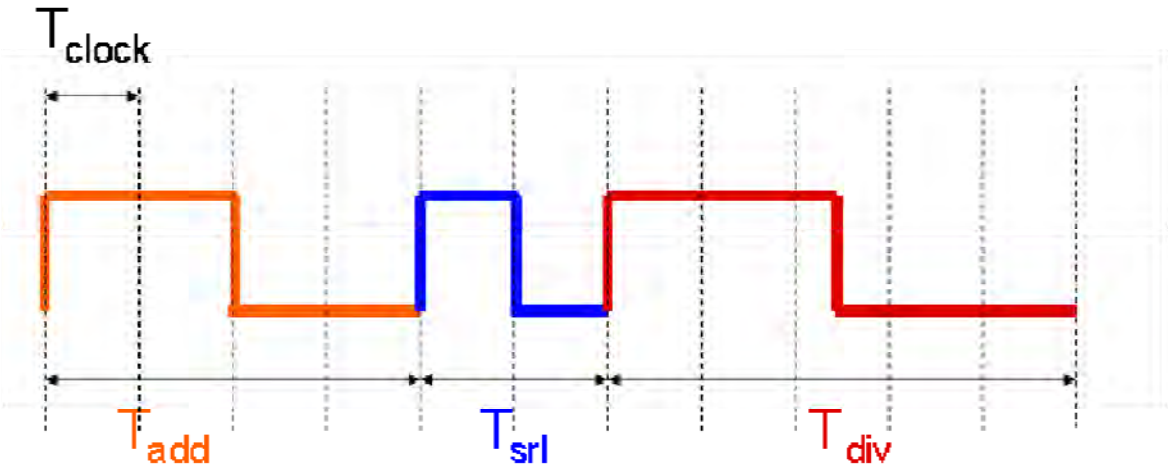
- Viola o princípio de "tornar o caso comum mais rápido"
- Datapaths monociclo não são mais utilizados em processadores modernos
- É mais eficiente executar cada instrução em um número variável de ciclos mais rápidos, utilizando apenas o necessário
- Este é o principio básico do datapath multiciclo

5

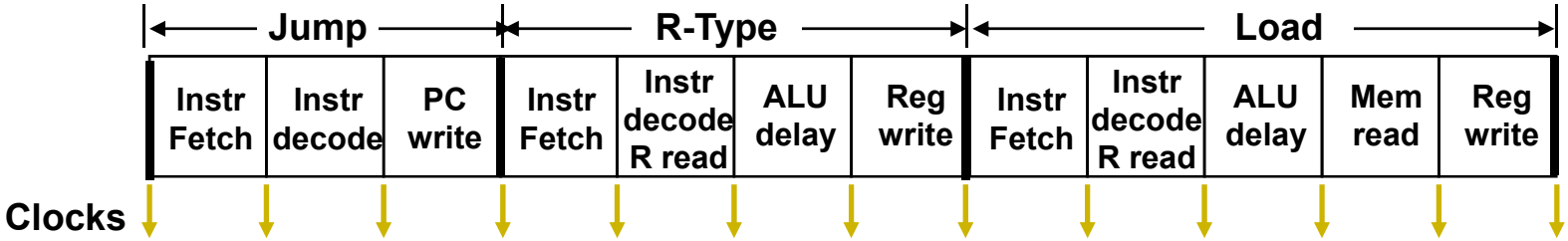
Multiciclo

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



Fonte: Prof. Savio chau: CSM151B

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 os uso de somadores extras)
- Dados a serem usados na mesma instrução em 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)

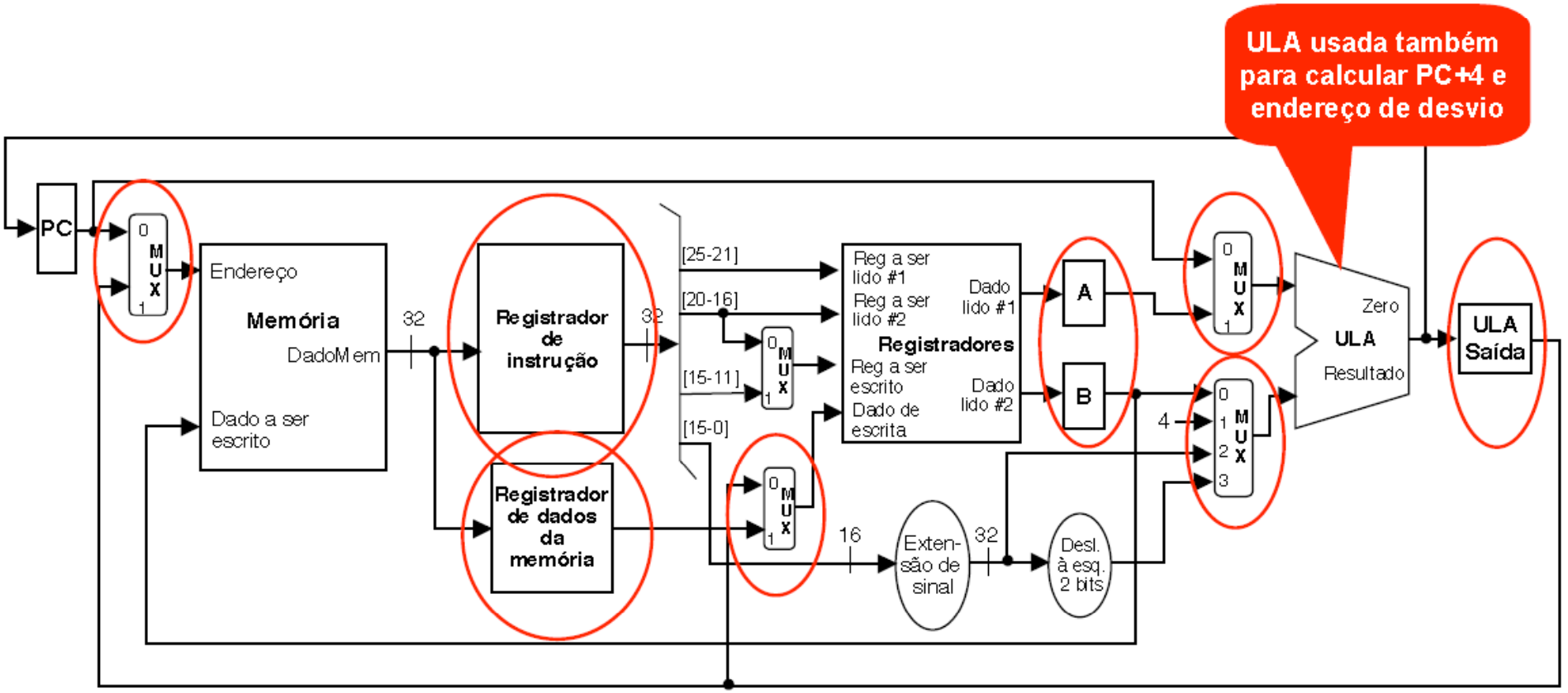
Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

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

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

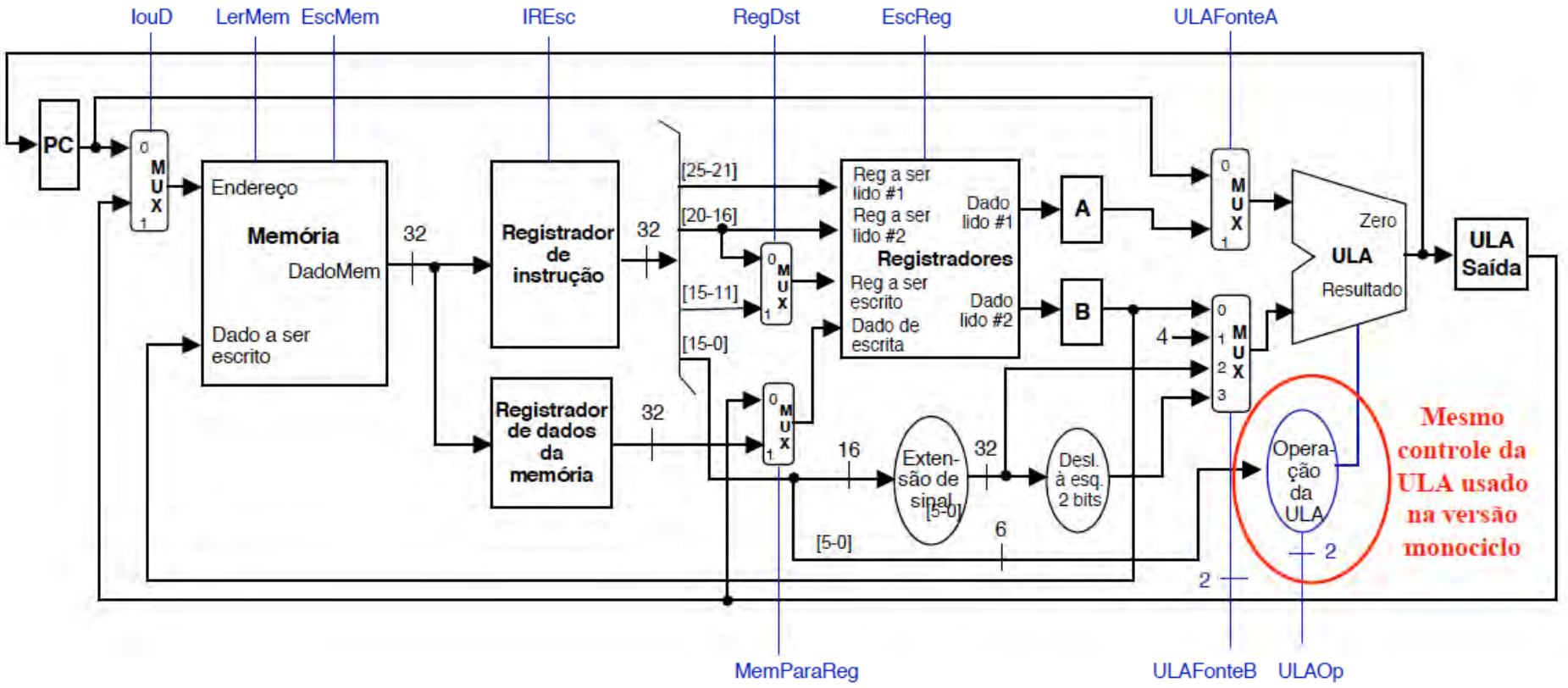
Datapath Multiciclo



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Sinais de Controle



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

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)

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

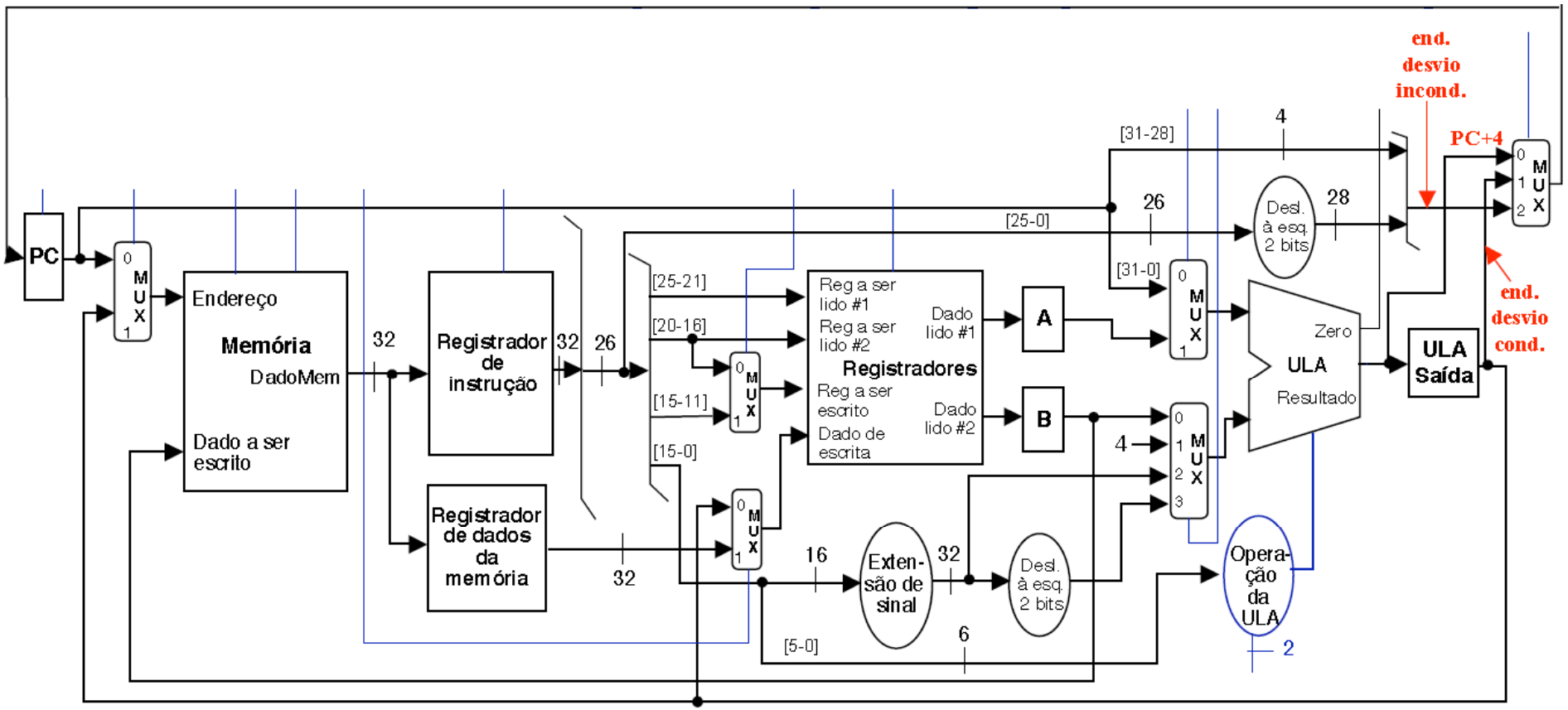
Datapath Multiciclo

- Três origens para o valor de PC:
  - Saída da ALU (PC + 4) (Entrada 0): este valor sempre será armazenado no PC
  - Registrador ULASaída, 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)

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

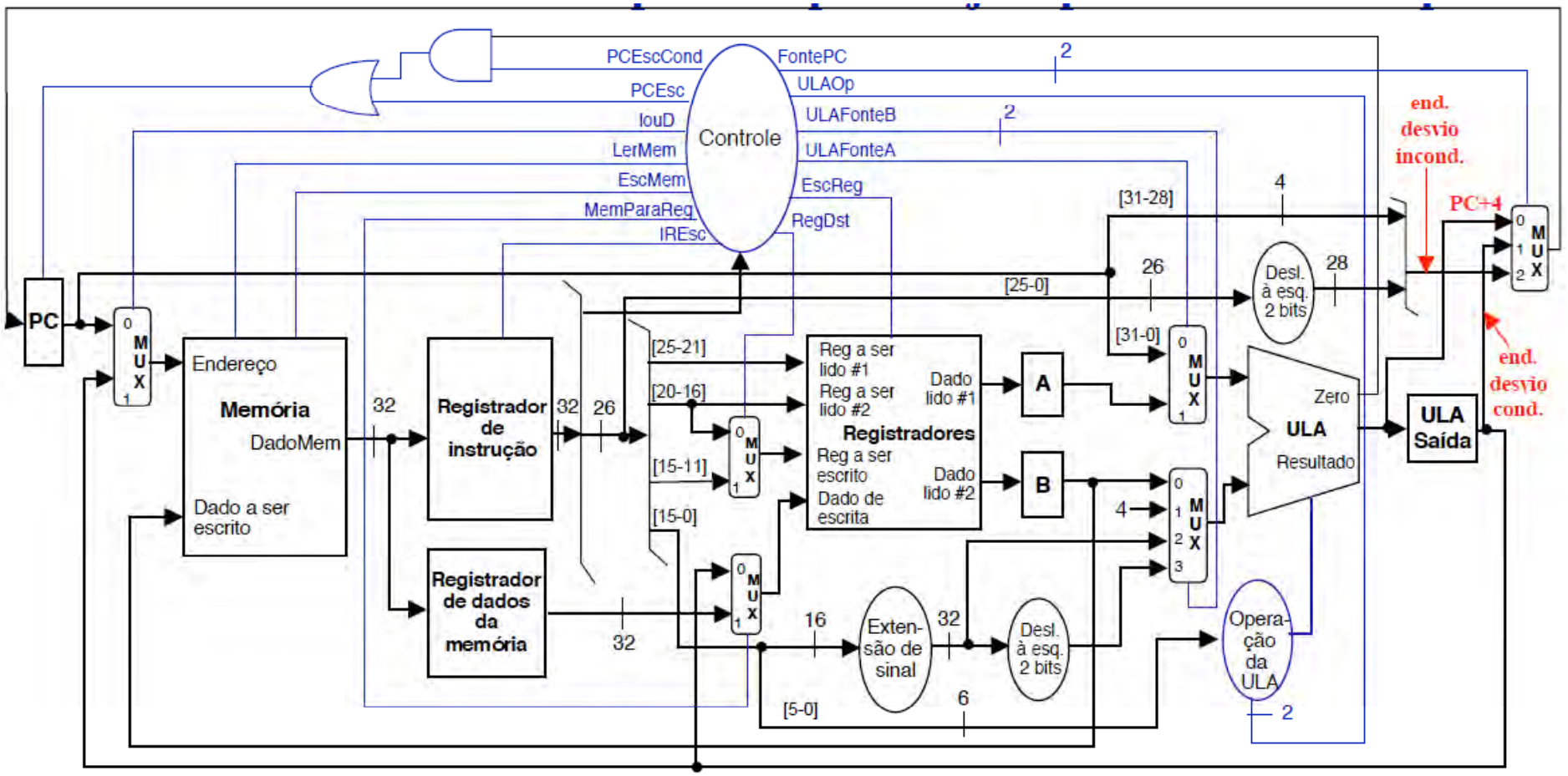
- Três origens para o valor de PC:



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Sinais de Controle



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

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

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções

1. Busca da Instrução (e incremento do PC)

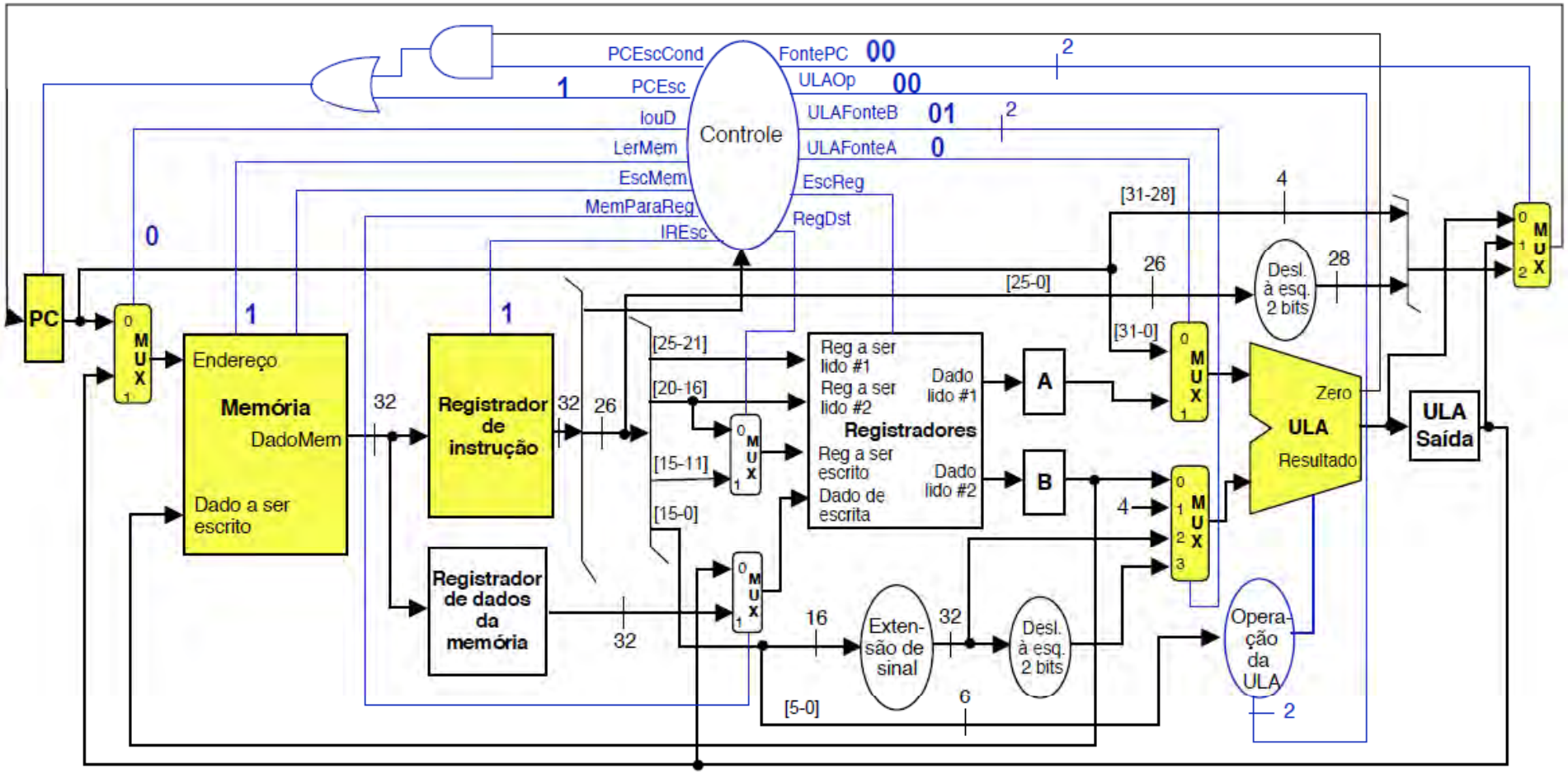
```
RI = Mem[PC];
PC = PC + 4;
```

Estas operações ocorrem em paralelo

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções: busca



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções

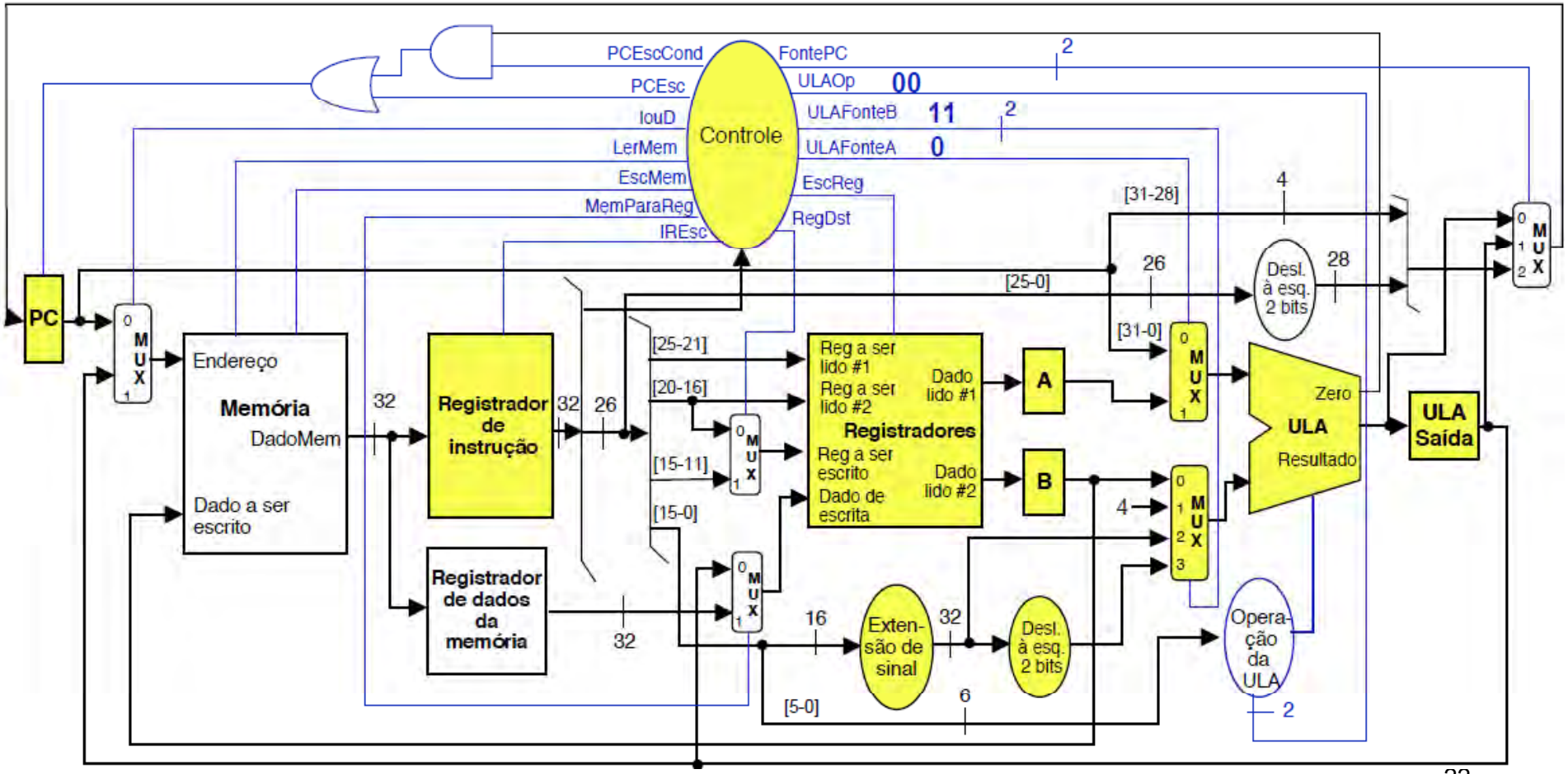
  2. Decodificação (Geração dos Sinais de Controle) e Leitura de Rs e Rt

  A = Reg[RI[25-21]];
  B = Reg[RI[20-16]];
  ULASaída = PC + (extensão de sinal (RI[15-0]) << 2)

  Estas operações ocorrem em paralelo

Datapath Multiciclo

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



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções

3. Execução da Instrução:
   - Referência à memória (lw e sw)

     ULASaída = A + extensão de sinal(RI[15-0]);

   - Tipo R

     ULASaída = A op B;

   - Desvio Condicional

     Se (A == B) então PC = ULASaída;

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

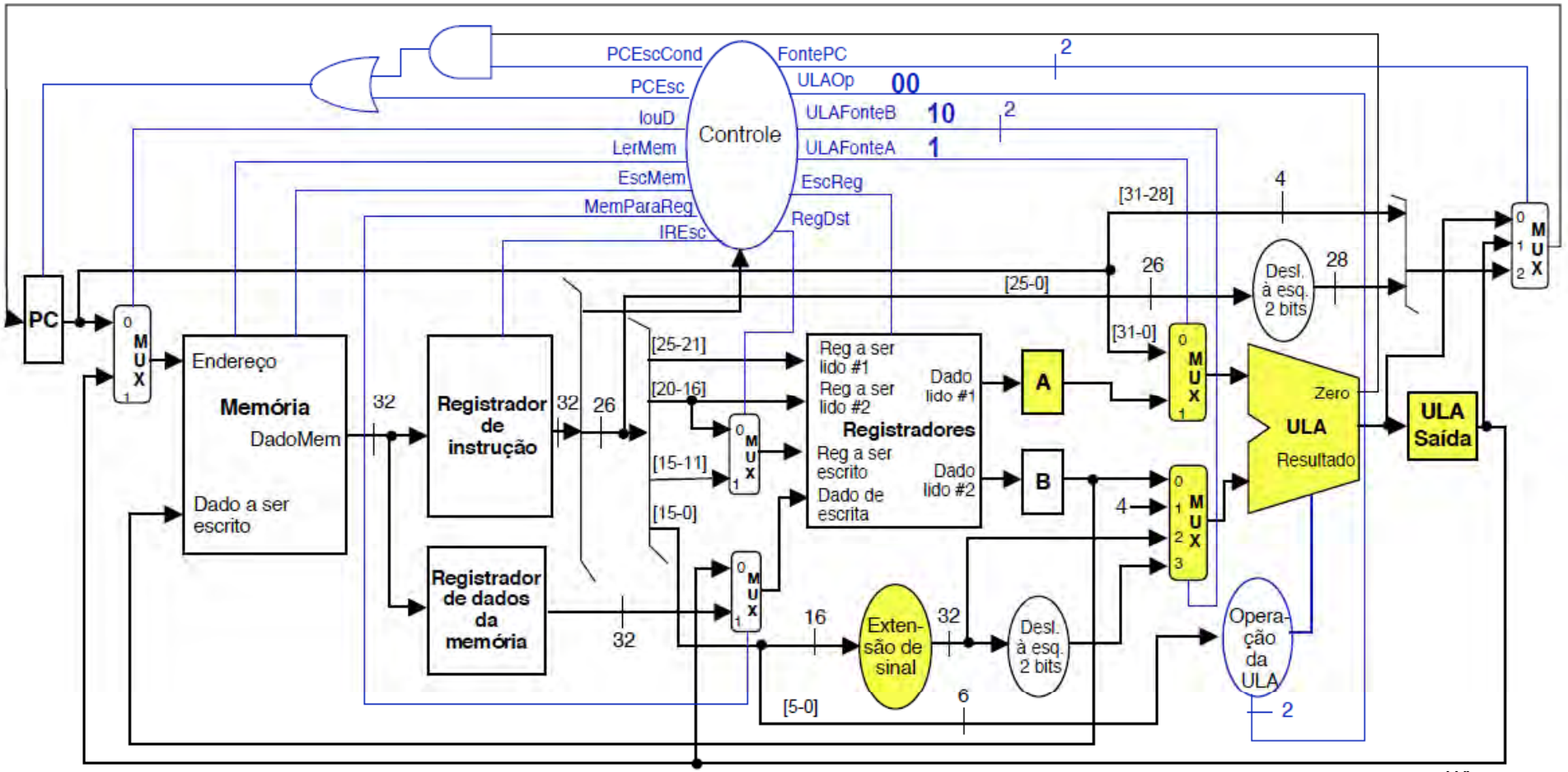
   - Desvio Incondicional

     PC = PC[31-28] || (RI[25-0] << 2);

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

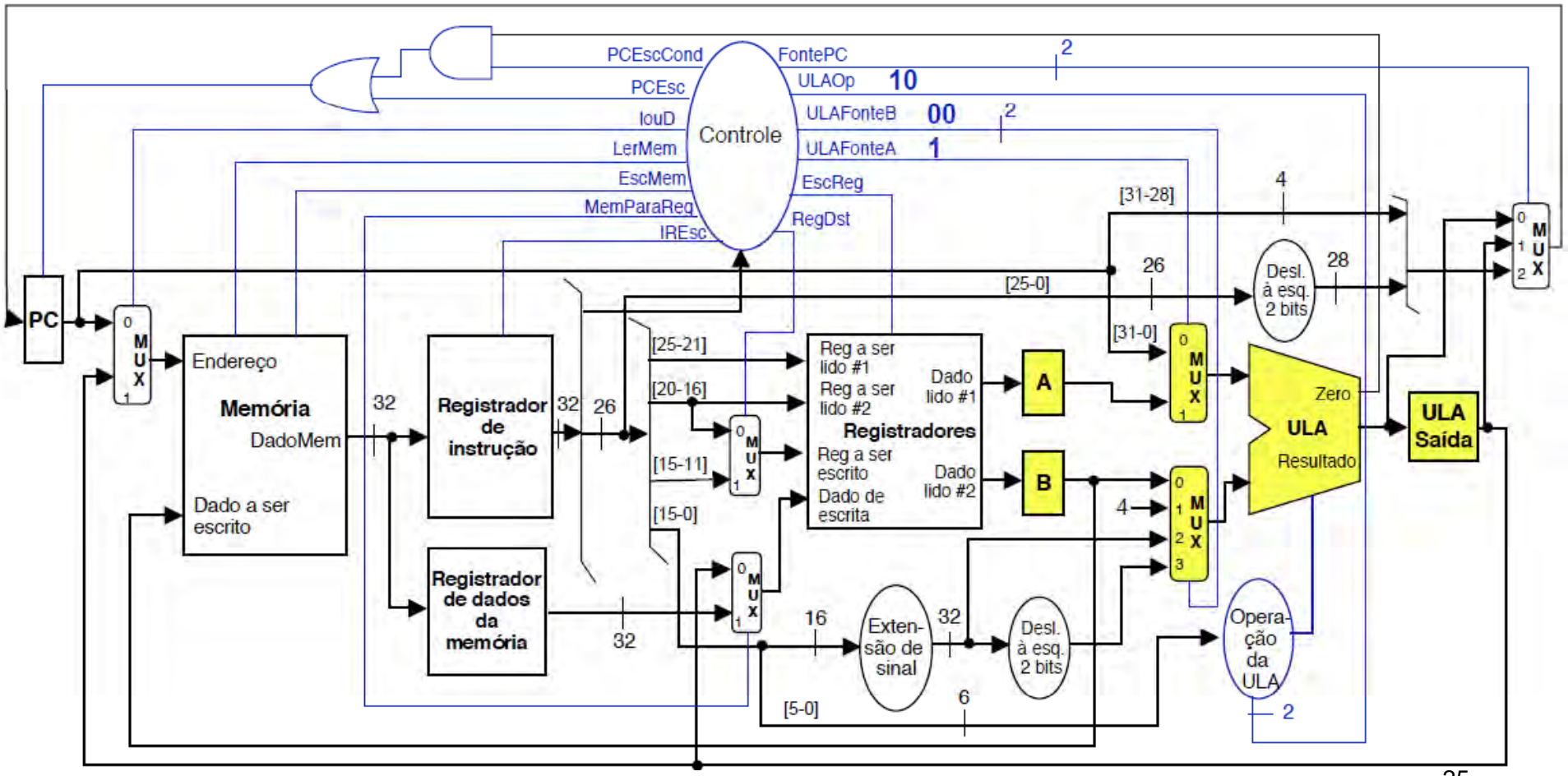
- Execução de Instruções: execução lw/sw



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

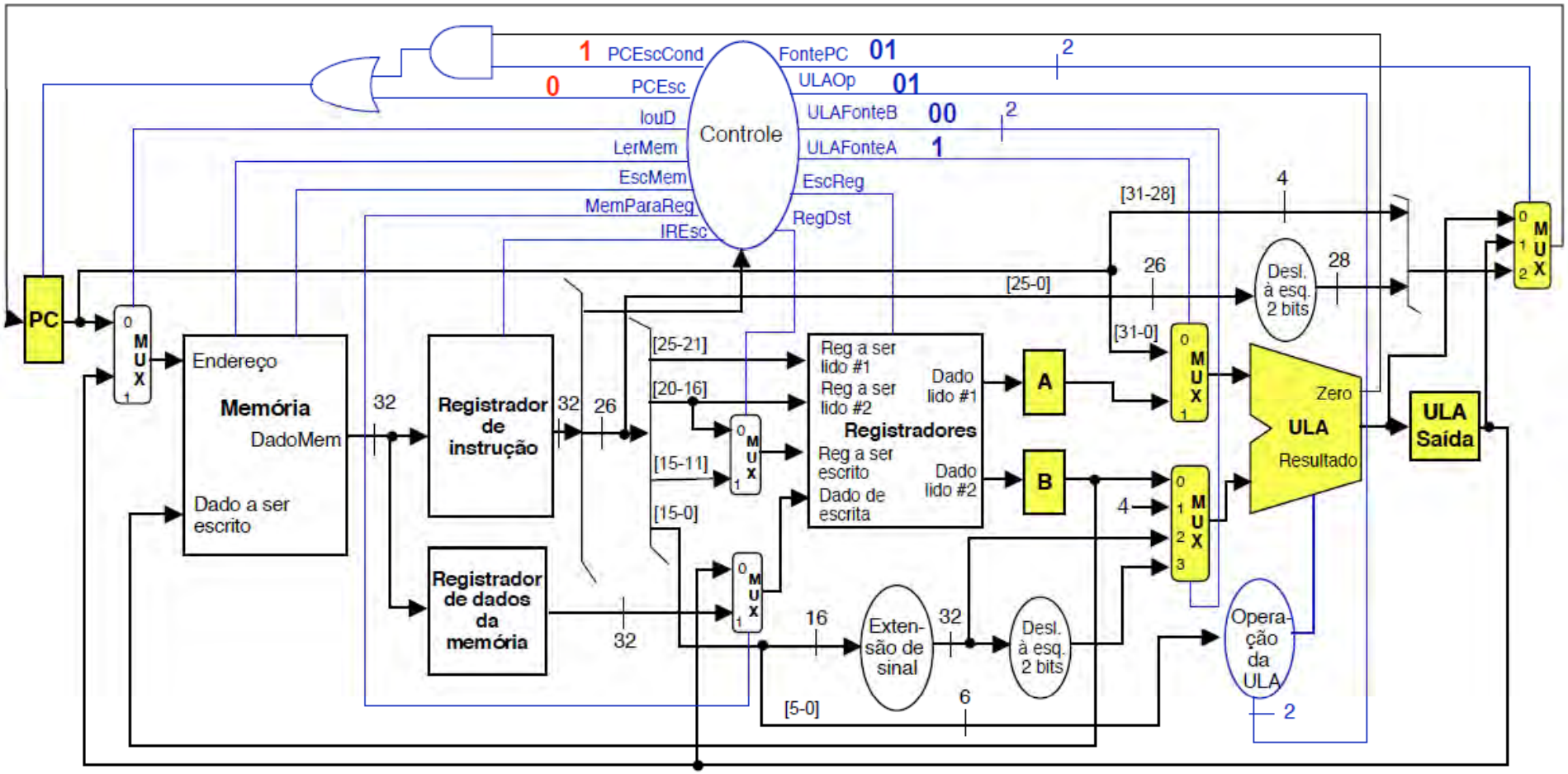
- Execução de Instruções: execução tipo R



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

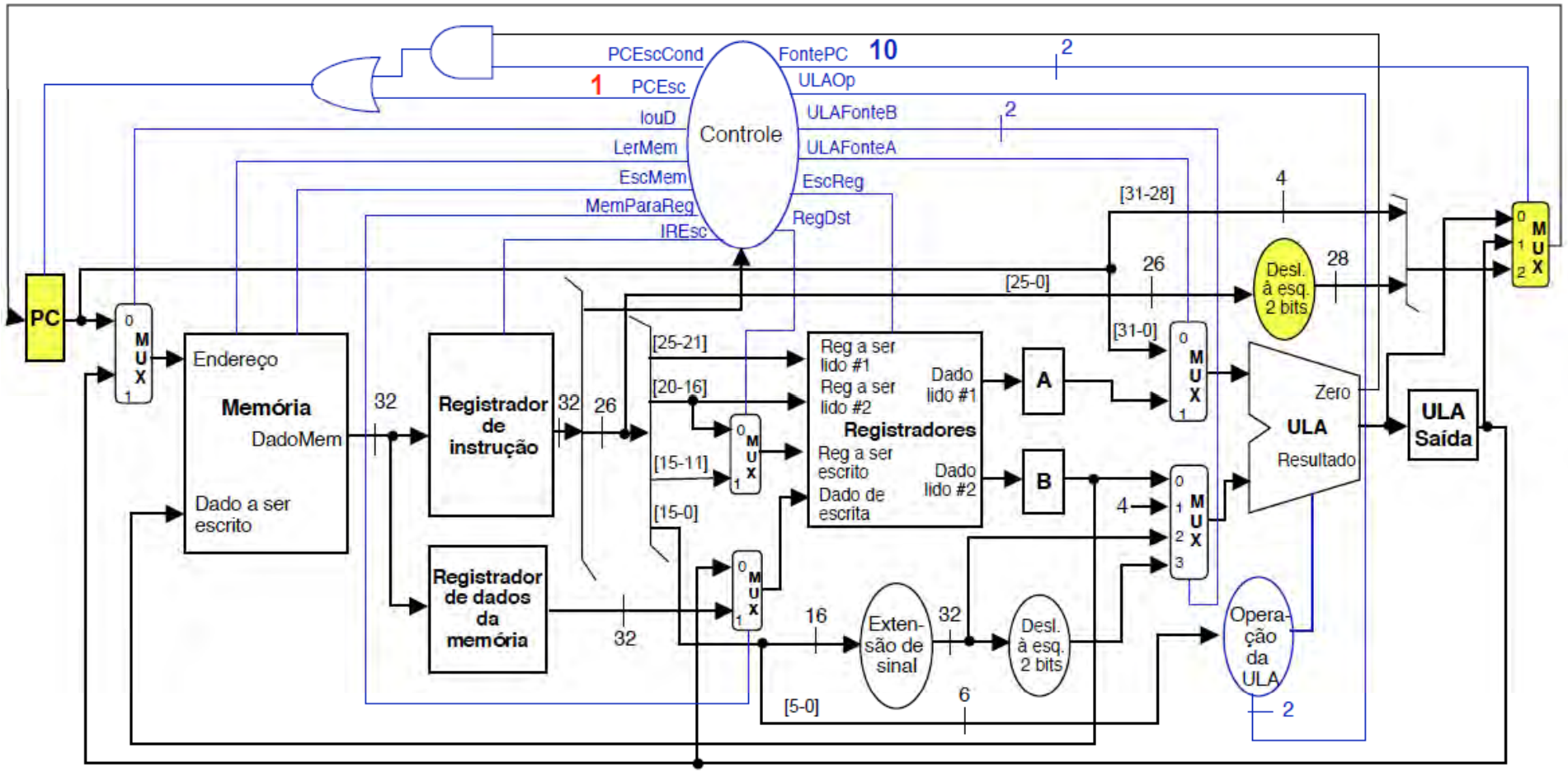
- Execução de Instruções: execução beq



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções: execução jump



Datapath Multiciclo

- Execução de Instruções

4. Final da Execução de sw e Tipo R:
   - lw:

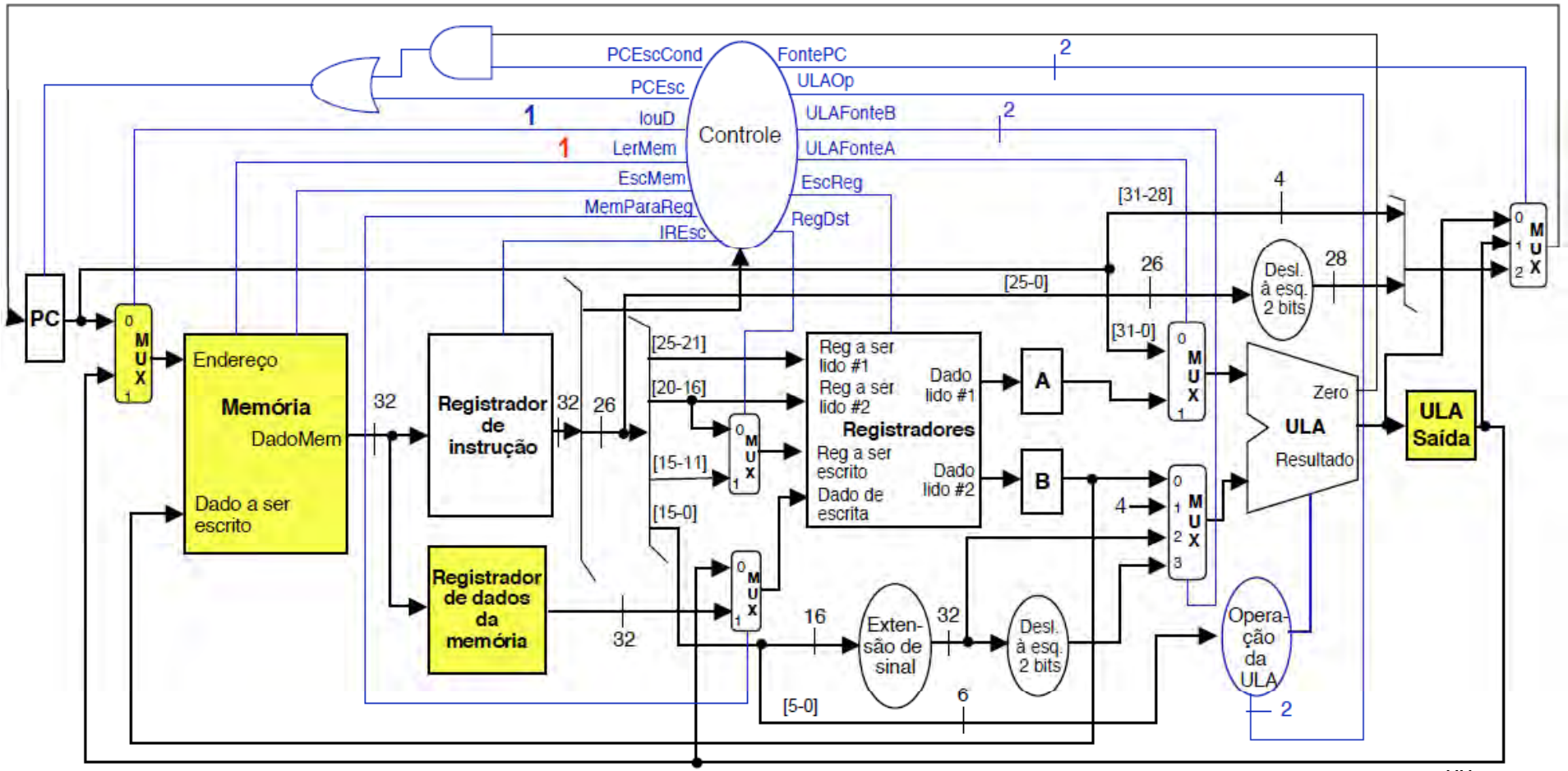
     RDM = Mem[ULASaída];
   - sw:

     Mem[ULASaída] = B;
   - Tipo R:

     Reg[RI[15-11]] = ULASaída;

Datapath Multiciclo

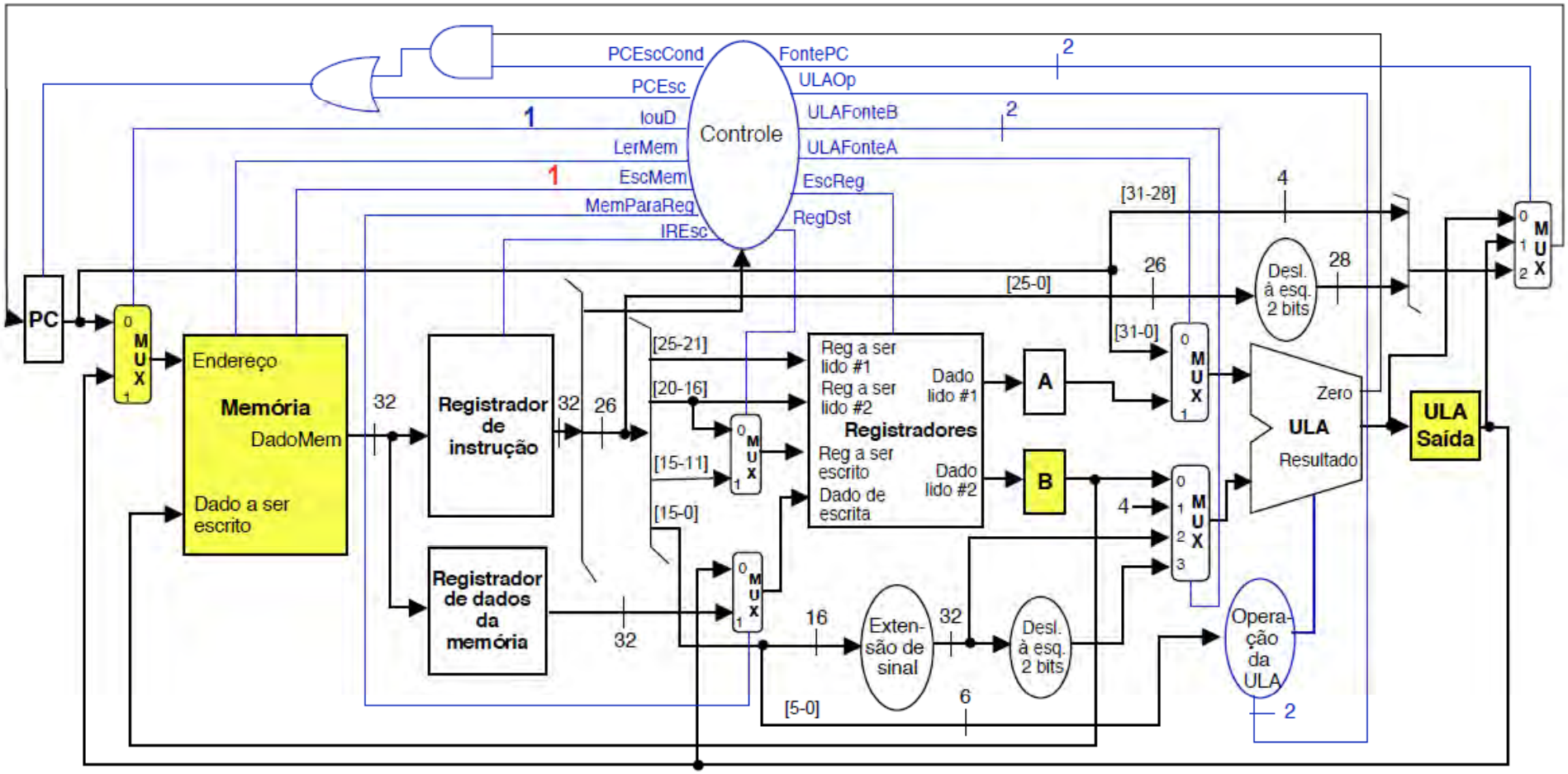
- Execução de Instruções: lw lê dado da memória



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

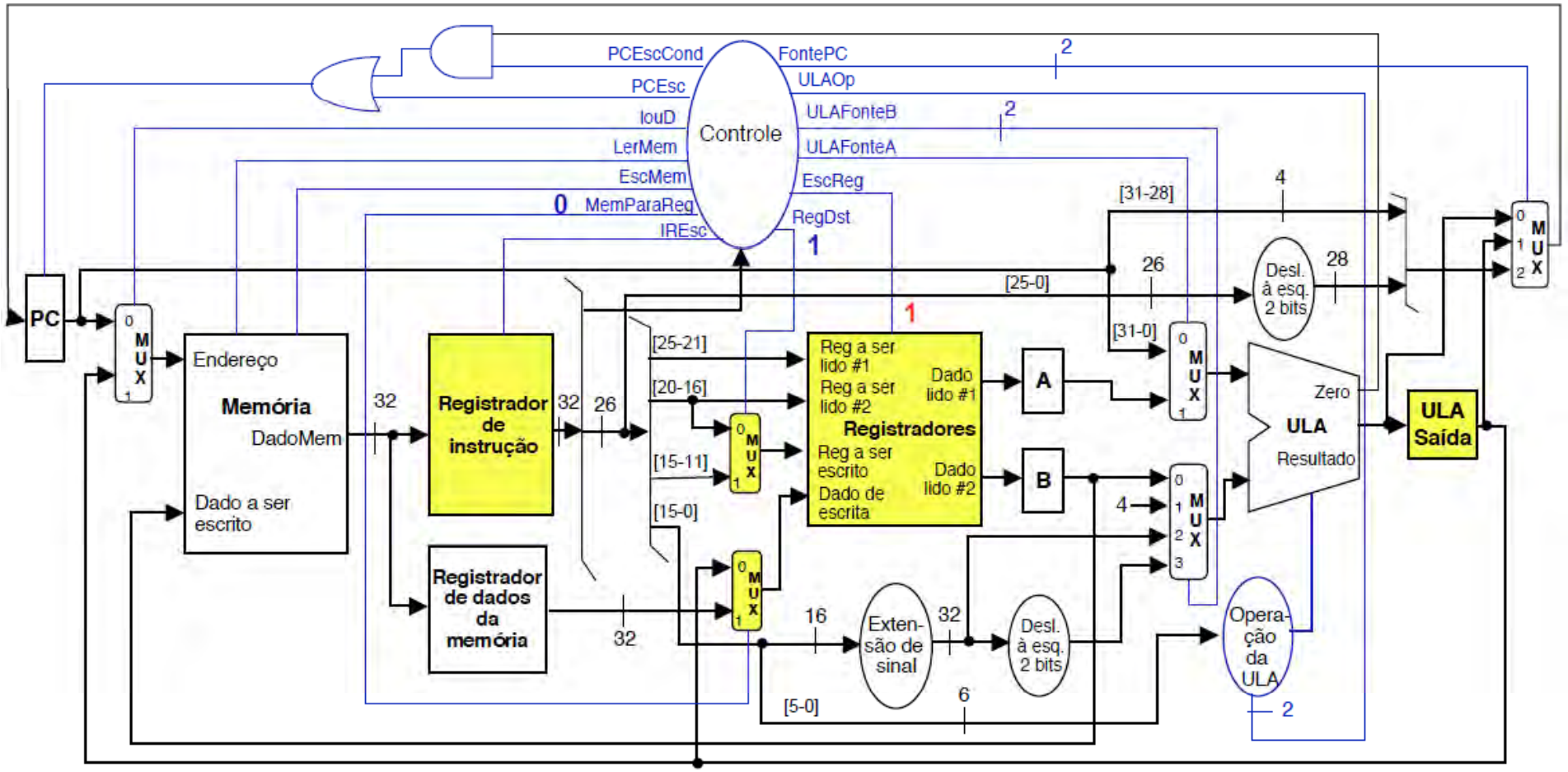
- Execução de Instruções: conclui sw



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções: conclui tipo R



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções

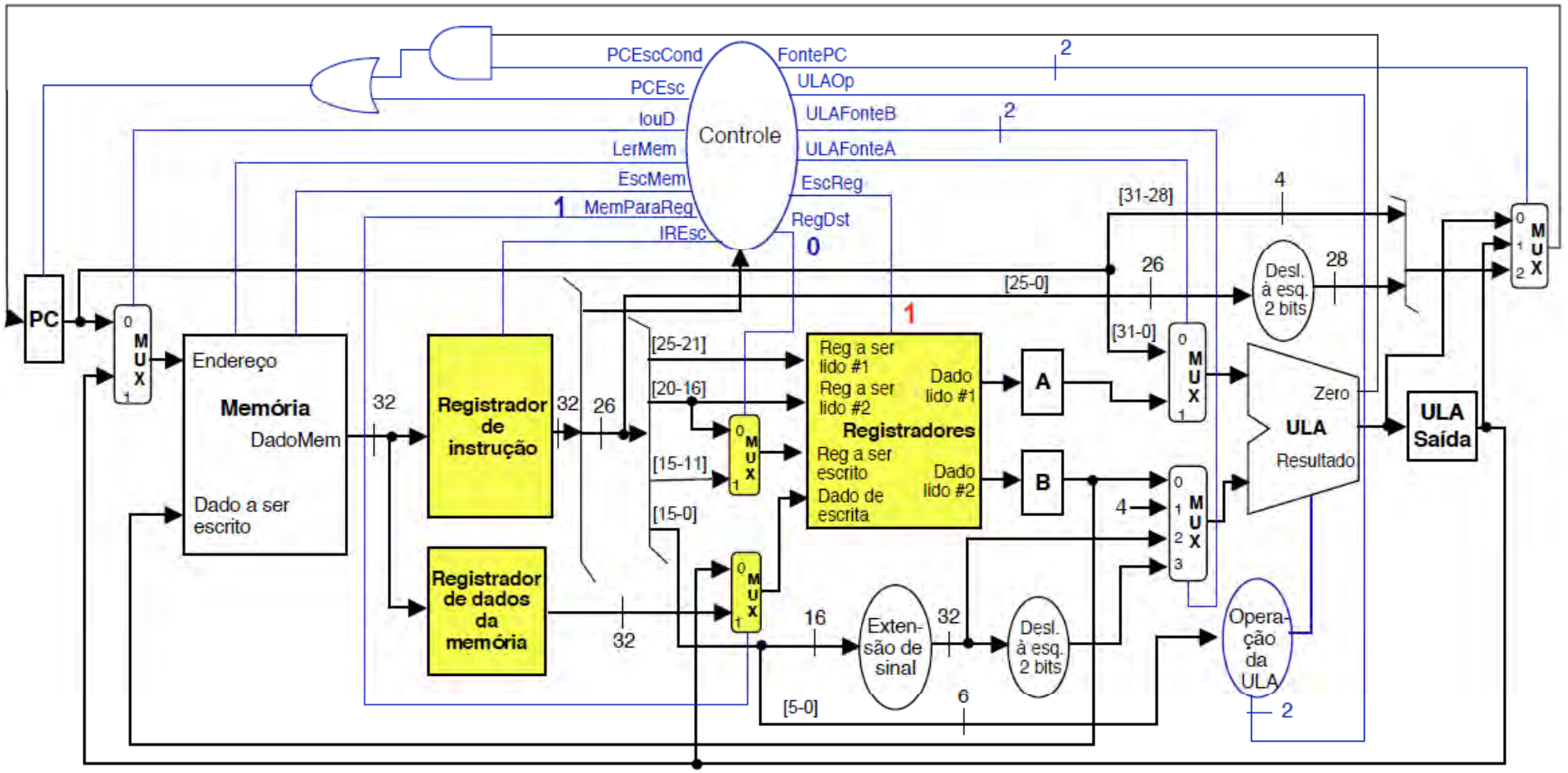
  5. Final da Execução de lw:
     - lw:

       Reg[RI[20-16]] = RDM;

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Execução de Instruções: conclui lw



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

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]) <<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] || (RI[25-0] <<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

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

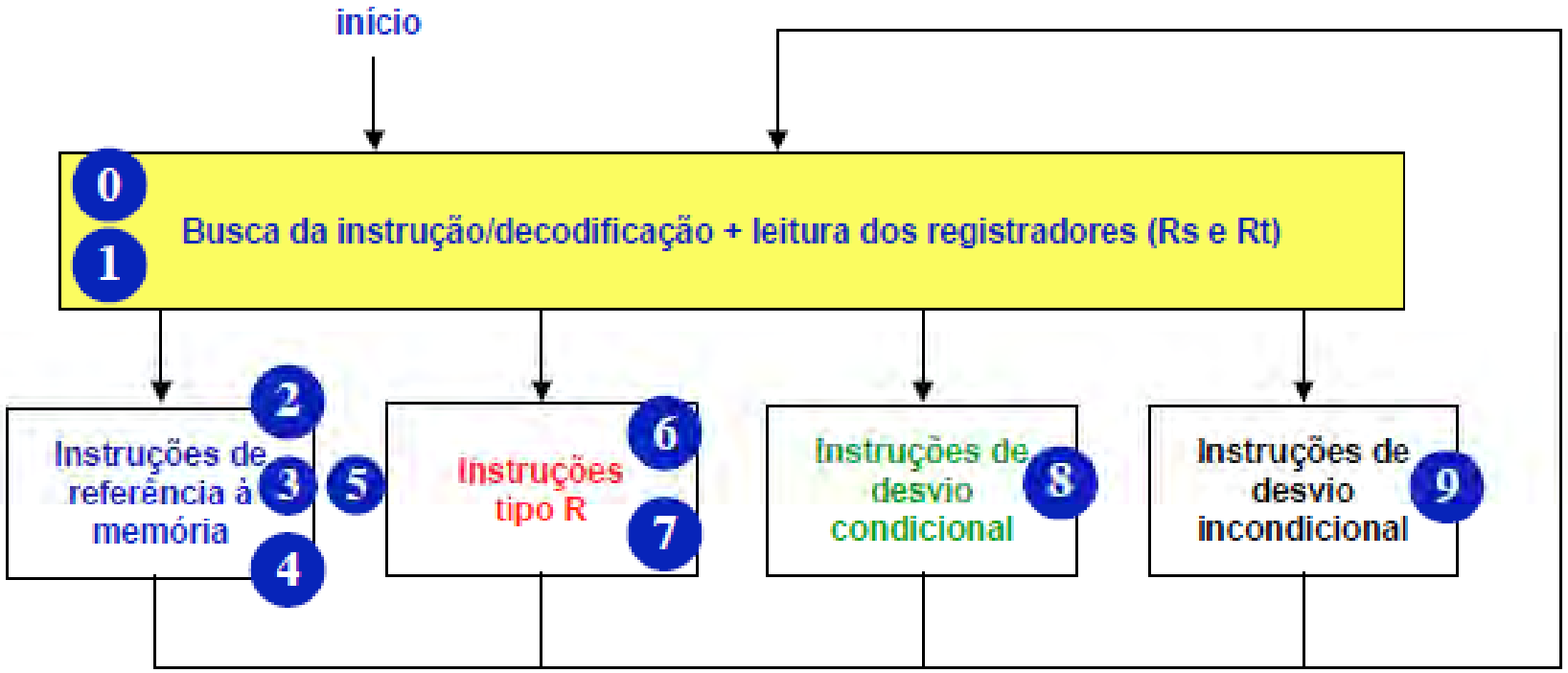
- Passos Necessários para Cada Instrução

Nome do passo	Instrução tipo R	Instrução lw	Instrução sw	Instrução beq	Instrução j
Busca da instrução	0 RI = Mem[PC] PC = PC + 4				
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)				
Execução, cálculo do endereço de acesso à memória, término de uma instrução branch/jump	6 ULAOp = A op	2 ULASaída = A + extensão de (RI[15-0])		8 Se (A == B) PC = ULASaída	9 PC = PC[ ] || (RI[25-0])
Término de uma instrução store word ou de tipo R	7 Reg [RI[15-11] ULASaída	3 RDM = Mem [ULASaída]	5 Mem [ULASa = B		
Término de uma instrução load word		4 Reg[RI[20- = RDM			
Número de passos	4	5	4	3	3

Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

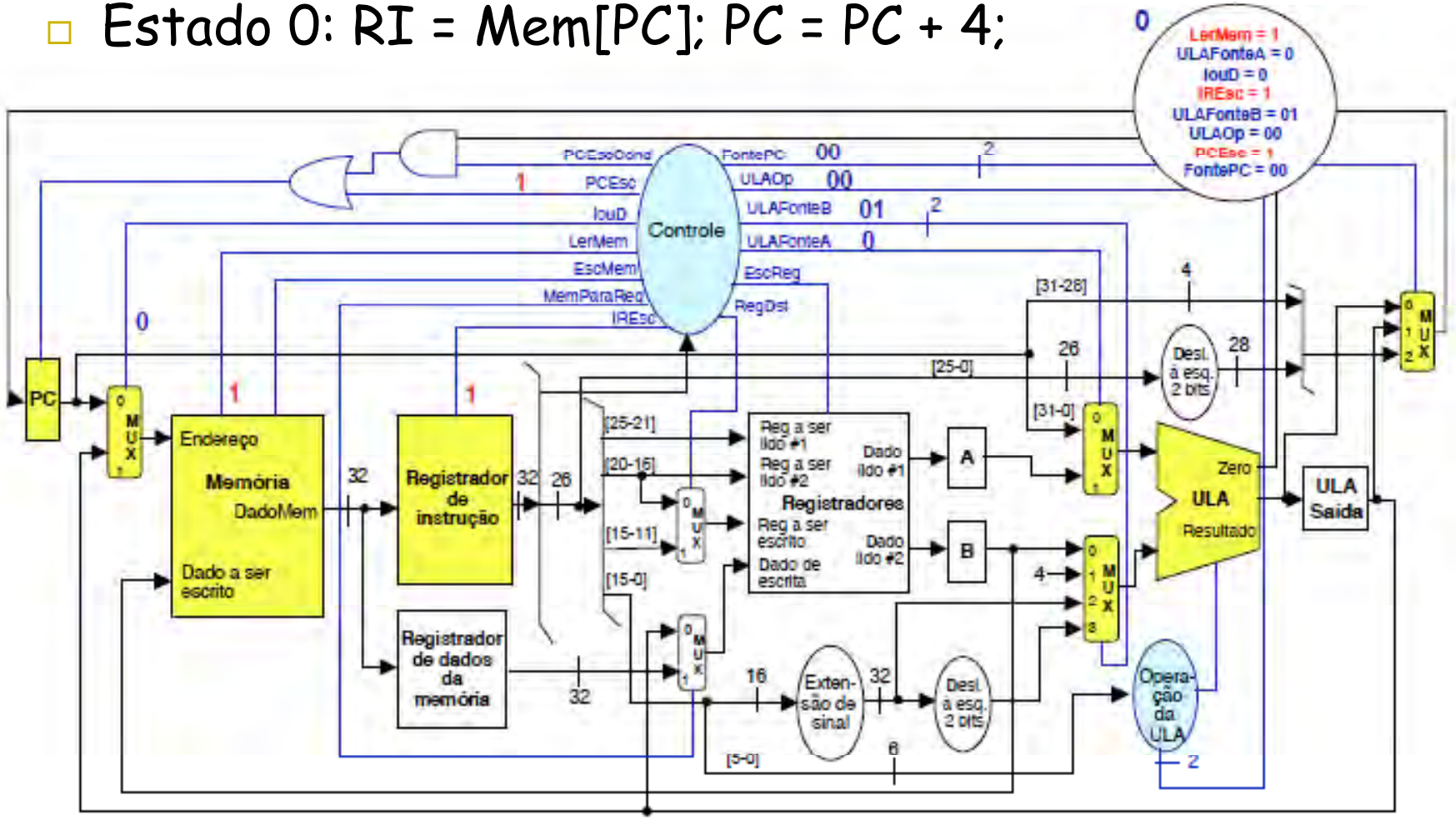
- Máquina de Estados



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

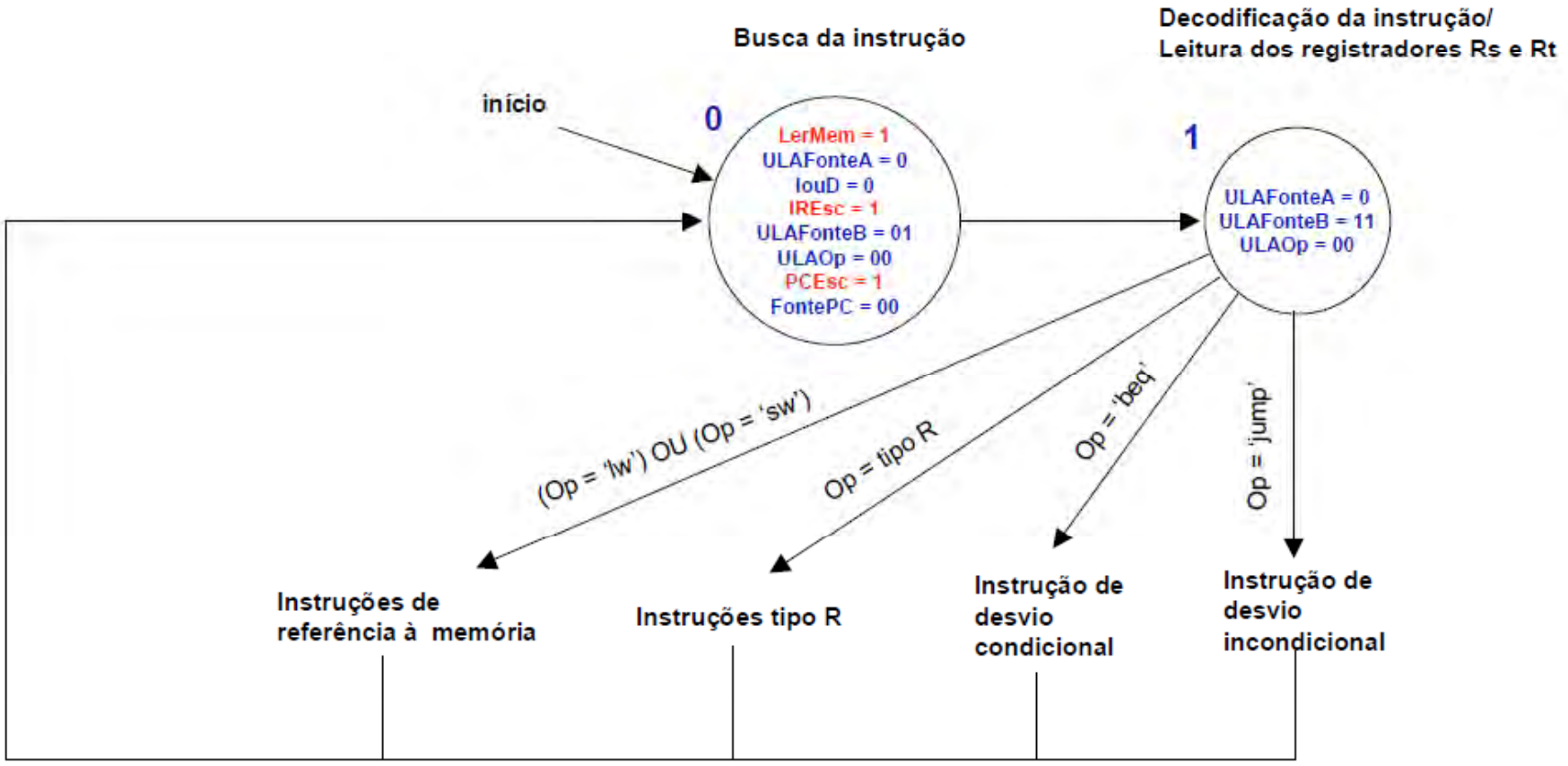
- Estado 0: RI = Mem[PC]; PC = PC + 4;



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

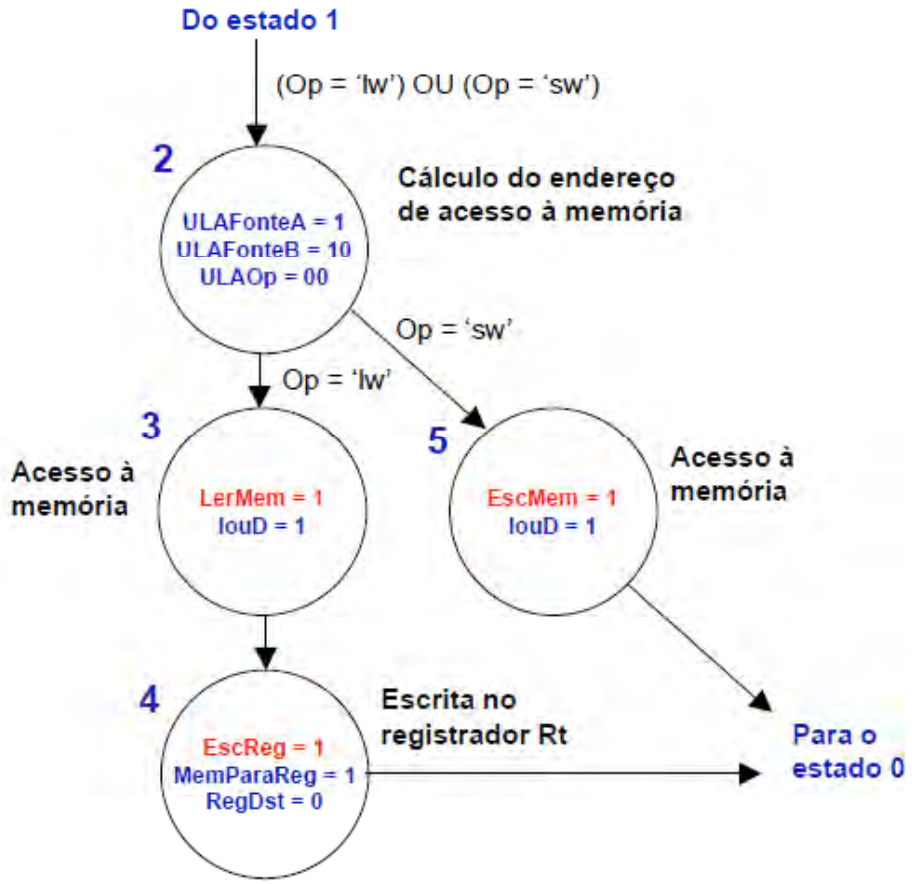
- Máquina de Estados: busca e decodificação



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

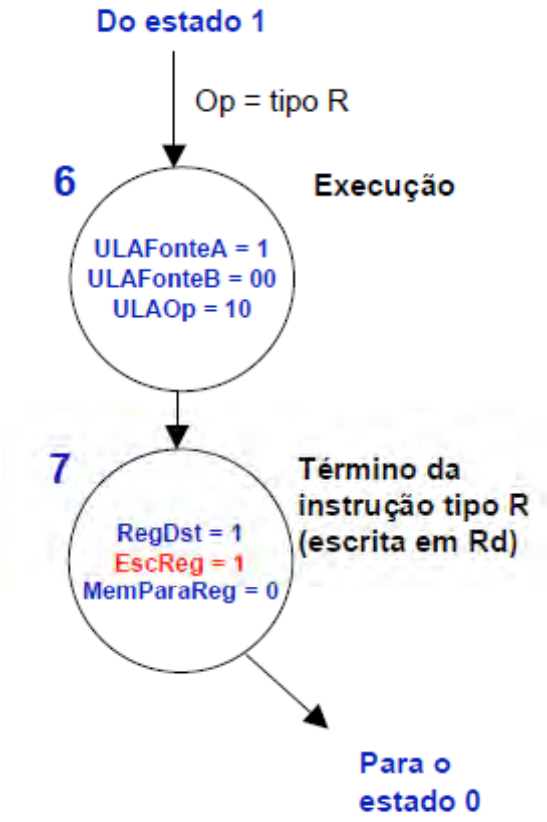
- Máquina de Estados: execução de instruções lw/sw



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

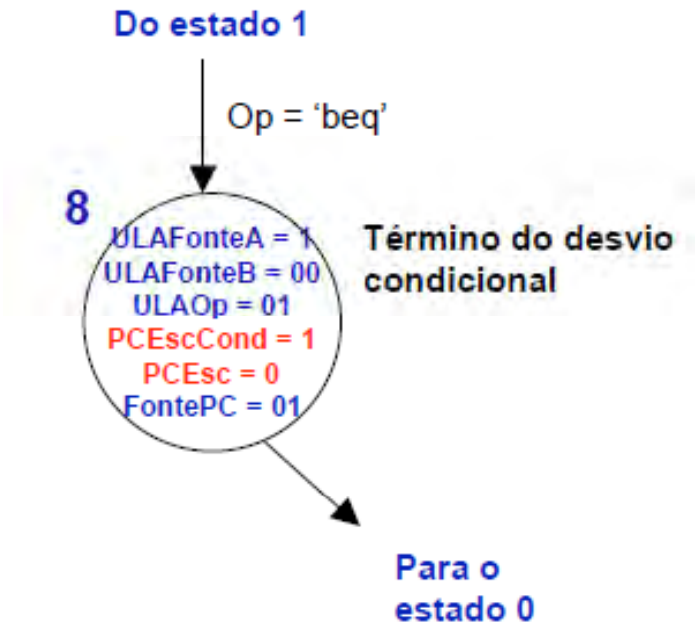
- Máquina de Estados: execução de instrução tipo R



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

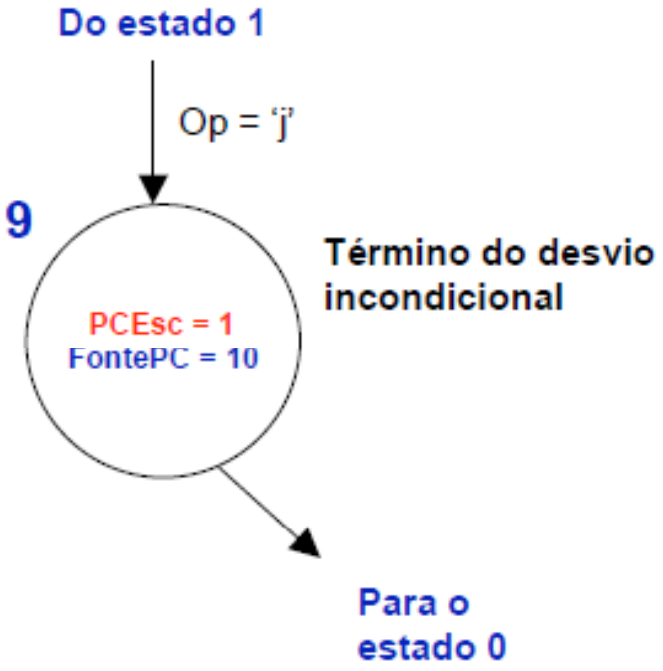
- Máquina de Estados: execução de instrução beq



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

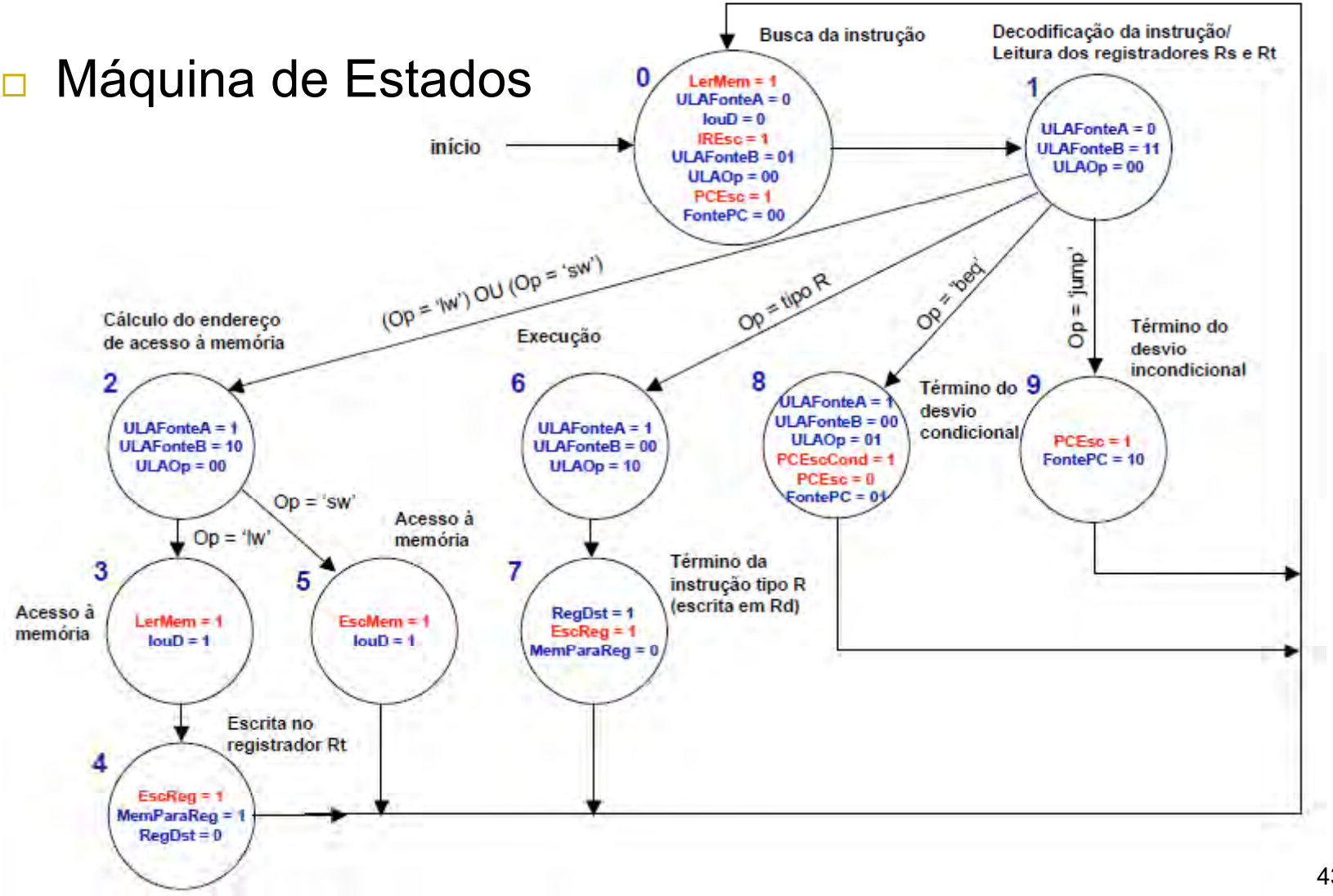
- Máquina de Estados: execução de instrução jump



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

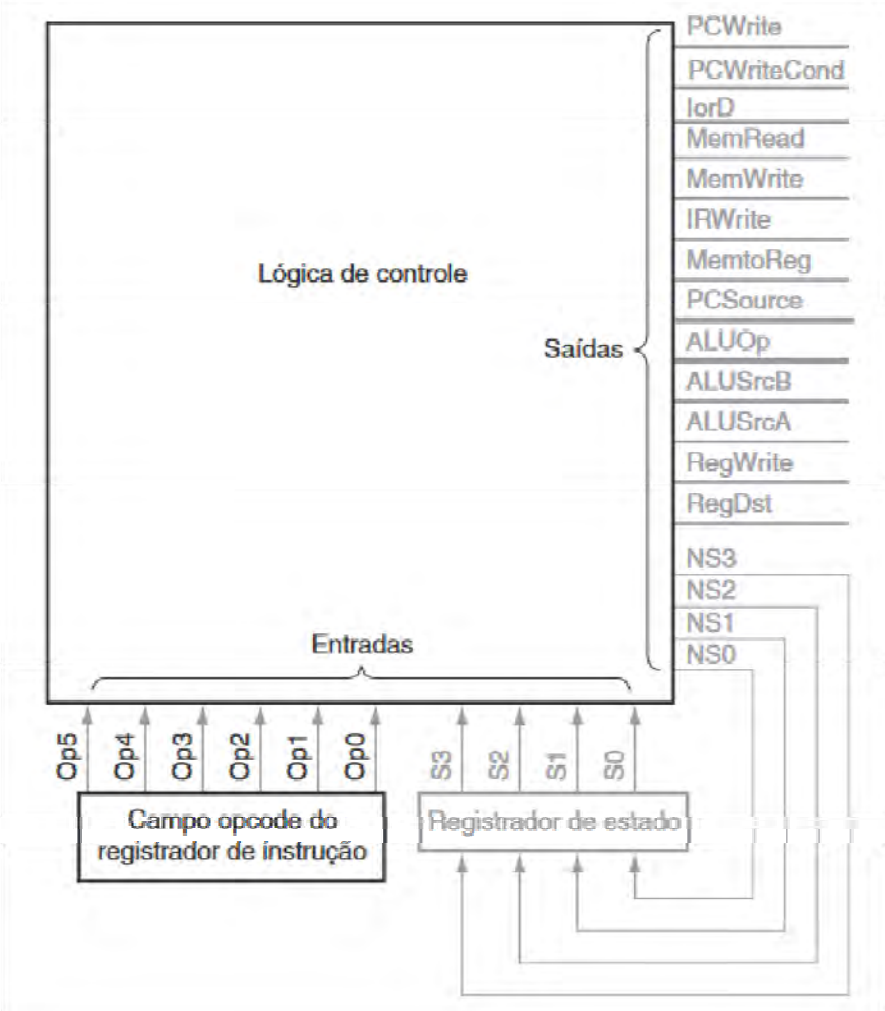
- Máquina de Estados



Fonte: Prof. José Luís Güntzel - Arquitetura e Organização de Computadores II

Datapath Multiciclo

- Máquina de Estados Moore



Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual	Opcode	Próximo Estado
0	-	1
1	'lw' ou 'sw'	2
1	tipo R	6
1	'beq'	8
1	'j'	9
2	'lw'	3
2	'sw'	5
3	-	4
4	-	0
5	-	0
6	-	7
7	-	0
8	-	0
9	-	0

Datapath Multiciclo

- Tabela de Atribuição de Estados

Estado	Flip-Flop			
	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual	Opcode	Próximo Estado
0000	-	0001
0001	LW	0010
0001	SW	0010
0001	TIPO r	0110
0001	BEQ	1000
0001	J	1001
0010	LW	0011
0010	SW	0101
0011	-	0100
0100	-	0000
0101	-	0000
0110	-	0111
0111	-	0000
1000	-	0000
1001	-	0000

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual				Opcode (Entrada)						Próximo Estado			
S3	S2	S1	S0	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0
0	0	0	0							0	0	0	1
0	0	0	1	1	0	0	0	1	1	0	0	1	0
0	0	0	1	1	0	1	0	1	1	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	1	0	1	0	0	1
0	0	1	0	1	0	0	0	1	1	0	0	1	1
0	0	1	0	1	0	1	0	1	1	0	1	0	1
0	0	1	1							0	1	0	0
0	1	0	0							0	0	0	0
0	1	0	1							0	0	0	0
0	1	1	0							0	1	1	1
0	1	1	1							0	0	0	0
1	0	0	0							0	0	0	0
1	0	0	1							0	0	0	0

Datapath Multiciclo

- Tabela de Transição de Estados

Estado Atual				Opcode (Entrada)						Próximo Estado			
S3	S2	S1	S0	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0
0	0	0	0							0	0	0	1
0	0	0	1	1	0	0	0	1	1	0	0	1	0
0	0	0	1	1	0	1	0	1	1	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	1	0	1	0	0	1
0	0	1	0	1	0	0	0	1	1	0	0	1	1
0	0	1	0	1	0	1	0	1	1	0	1	0	1
0	0	1	1							0	1	0	0
0	1	0	0							0	0	0	0
0	1	0	1							0	0	0	0
0	1	1	0							0	1	1	1
0	1	1	1							0	0	0	0
1	0	0	0							0	0	0	0
1	0	0	1							0	0	0	0

NS0 = S3'·S2'·S1'·S0' +
S3'·S2'·S1'·S0·OP5'·OP4'·OP3'·OP2'·OP1·OP0' +
S3'·S2'·S1·S0'·OP5·OP4'·OP3'·OP2'·OP1·OP0 +
S3'·S2'·S1·S0'·OP5·OP4'·OP3·OP2'·OP1·OP0 +
S3'·S2·S1·S0'

Datapath Multiciclo

- Sinais de controle (saída)

Saída	Estados atuais
EscrevePC	state0 + state9
EscrevePCCond	state8
IouD	state3 + state5
LeMem	state0 + state3
EscreveMem	state5
EscreveIR	state0
MemparaReg	state4
OrigPC1	state9
OrigPC0	state8
OpALU1	state6
OpALU0	state8
OrigBALU1	state1 + state2
OrigBALU0	state0 + state1
OrigAALU	state2 + state6 + state8
EscreveReg	state4 + state7
RegDst	state7

Datapath Multiciclo

- Sinais de controle (saída)

Estado Atual				Opcode (Entrada)						Próximo Estado																				
S3	S2	S1	S0	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0	EscrevePC	EscrevePCCon	IouD	LeMem	EscreveMem	EscreveIR	MemparaReg	OrigPC1	OrigPC0	OpALU1	OpALU0	OrigBALU1	OrigBALU0	OrigAALU	EscreveReg	RegDst
0	0	0	0							0	0	0	1	1			1		1							1				
0	0	0	1	1	0	0	0	1	1	0	0	1	0												1	1			
0	0	0	1	1	0	1	0	1	1	0	0	1	0												1	1			
0	0	0	1	0	0	0	0	0	0	0	1	1	0												1	1			
0	0	0	1	0	0	0	1	0	0	1	0	0	0												1	1			
0	0	0	1	0	0	0	0	1	0	1	0	0	1												1	1			
0	0	1	0	1	0	0	0	1	1	0	0	1	1												1		1		
0	0	1	0	1	0	1	0	1	1	0	1	0	1												1		1		
0	0	1	1							0	1	0	0			1	1												
0	1	0	0							0	0	0	0							1								1	
0	1	0	1							0	0	0	0			1		1											
0	1	1	0							0	1	1	1										1				1		
0	1	1	1							0	0	0	0															1	1
1	0	0	0							0	0	0	0		1							1		1			1		
1	0	0	1							0	0	0	0	1							1								

Datapath Multiciclo

- Sinais de controle (saída)

Estado Atual				Opcode (Entrada)						Próximo Estado				EscrevePC	EscrevePCCon	IouD	LeMem	EscreveMem	EscreveIR	MemparaReg	OrigPC1	OrigPC0	OpALU1	OpALU0	OrigBALU1	OrigBALU0	OrigAALU	EscreveReg	RegDst
S3	S2	S1	S0	OP5	OP4	OP3	OP2	OP1	OP0	NS3	NS2	NS1	NS0																
0	0	0	0							0	0	0	1	1			1		1							1			
0	0	0	1	1	0	0	0	1	1	0	0	1	0												1	1			
0	0	0	1	1	0	1	0	1	1	0	0	1	0												1	1			
0	0	0	1	0	0	0	0	0	0	0	1	1	0												1	1			
0	0	0	1	0	0	0	1	0	0	1	0	0	0												1	1			
0	0	0	1	0	0	0	0	1	0	1	0	0	1												1	1			
0	0	1	0	1	0	0	0	1	1	0	0	1	1												1		1		
0	0	1	0	1	0	1	0	1	1	0	1	0	1												1		1		
0	0	1	1							0	1	0	0			1	1												
0	1	0	0							0	0	0	0							1								1	
0	1	0	1							0	0	0	0			1		1											
0	1	1	0							0	1	1	1										1				1		
0	1	1	1							0	0	0	0															1	1
1	0	0	0							0	0	0	0		1							1		1			1		
1	0	0	1							0	0	0	0	1							1								

EscrevePC = S3'·S2'·S1'·S0' +
S3·S2'·S1'·S0

Datapath Multiciclo

- Acelera a execução de algumas instruções
  - Ex: jump não utiliza todos os estágios do datapath
  - Ex: Type-R não acessa a memória de dados

- Não acelara a execução das instruções mais complexas (ex: lw)

Datapath Multiciclo

- É possível melhorar ainda mais o desempenho?
- Sim:
  1. Processo de fabricação
  2. Exploração de Paralelismo
  3. Microarquitetura Avançada
     - Pipelining