Unidade de Controle

- Linhas de Controle: valores

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	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

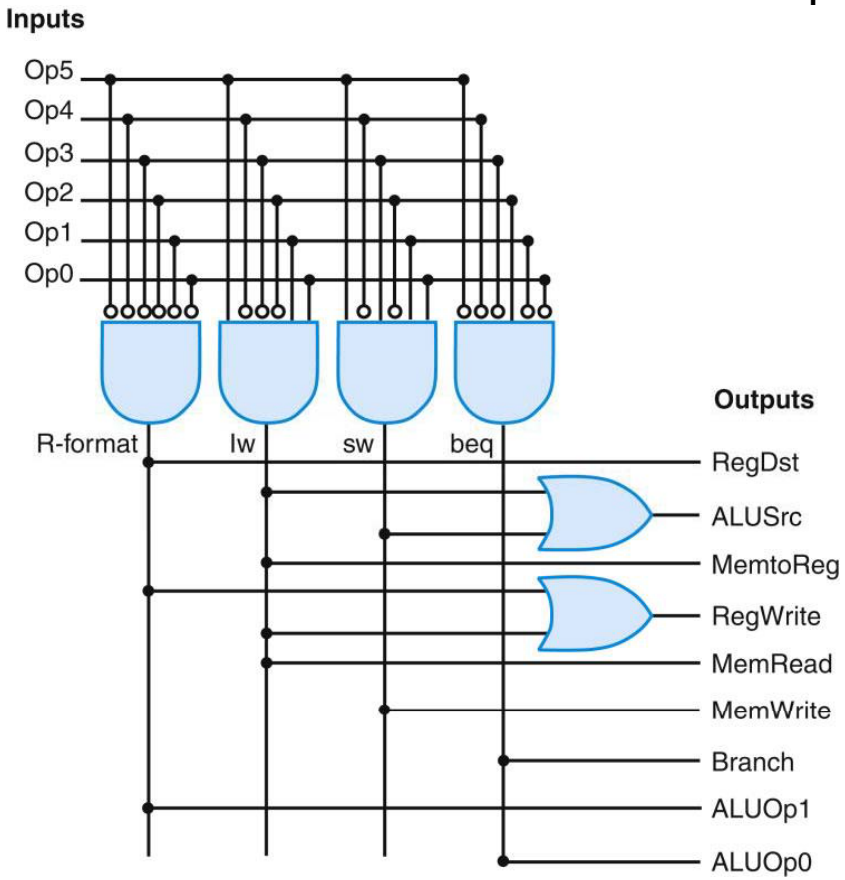
Unidade de Controle

- Linhas de Controle
  - Os níveis lógicos dos sinais de controle gerados pela unidade central dependem exclusivamente do opcode da instrução.

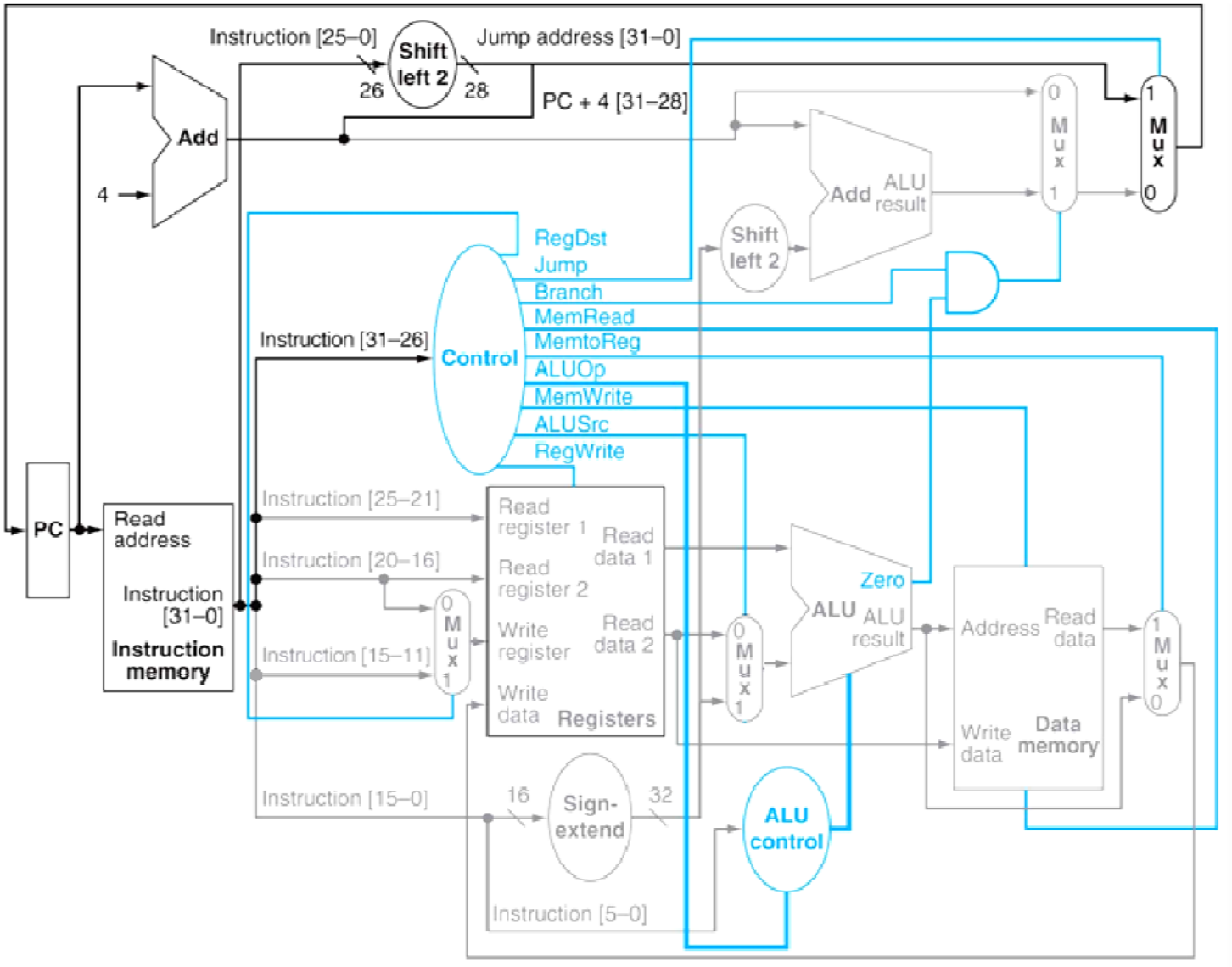
Input or output	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Unidade de Controle

- Linhas de Controle
  - A partir das informação da tabela verdade, é possível implementar diretamente a unidade central de controle usando portas lógicas:



Unidade de Controle



Execução de uma Instrução Tipo R

- Seja a instrução tipo R 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

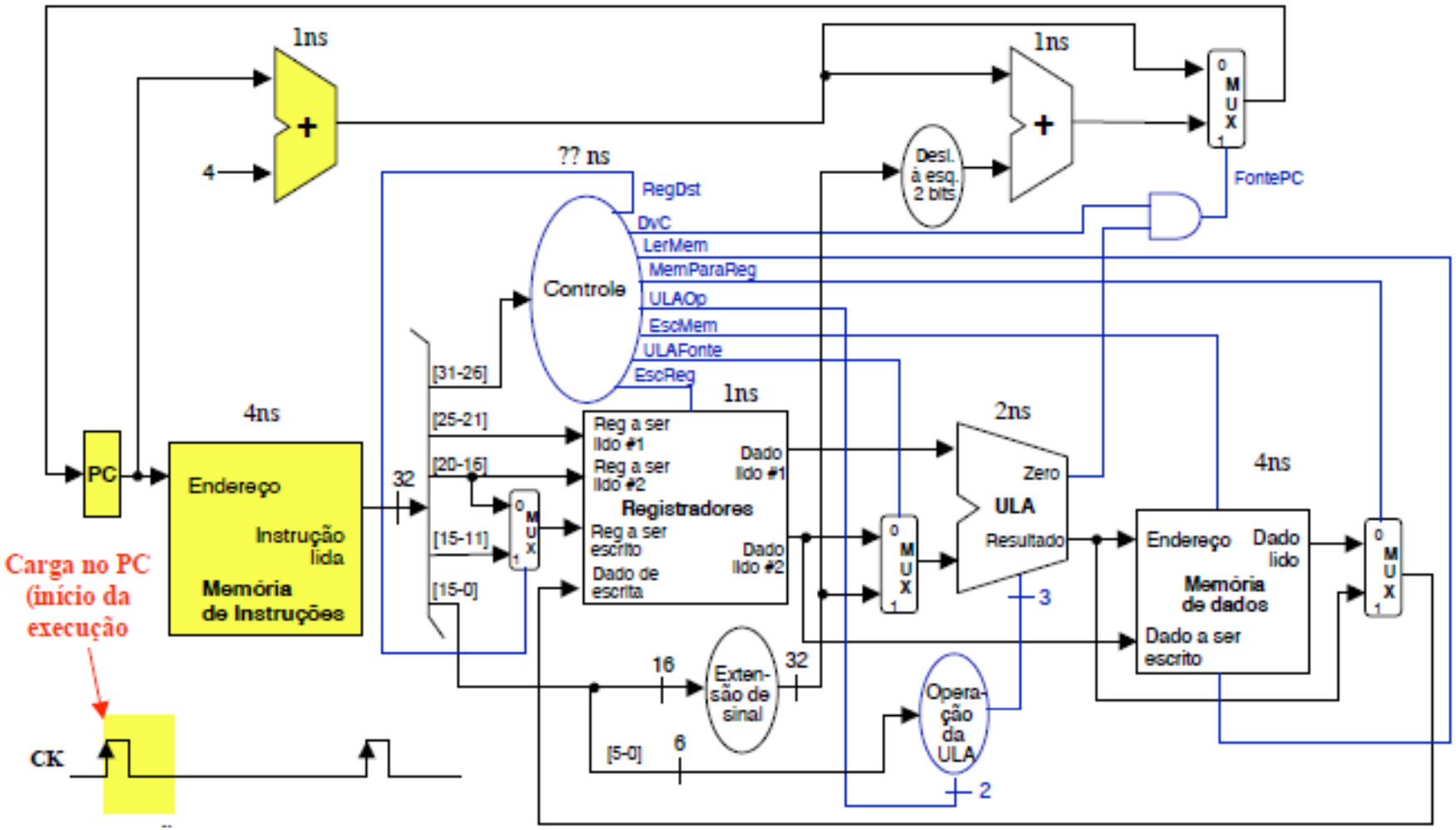
  4. Escrita (do resultado da operação realizada na ULA) no registrador destino ($t1 ou Rd)

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

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

Execução de uma Instrução Tipo R

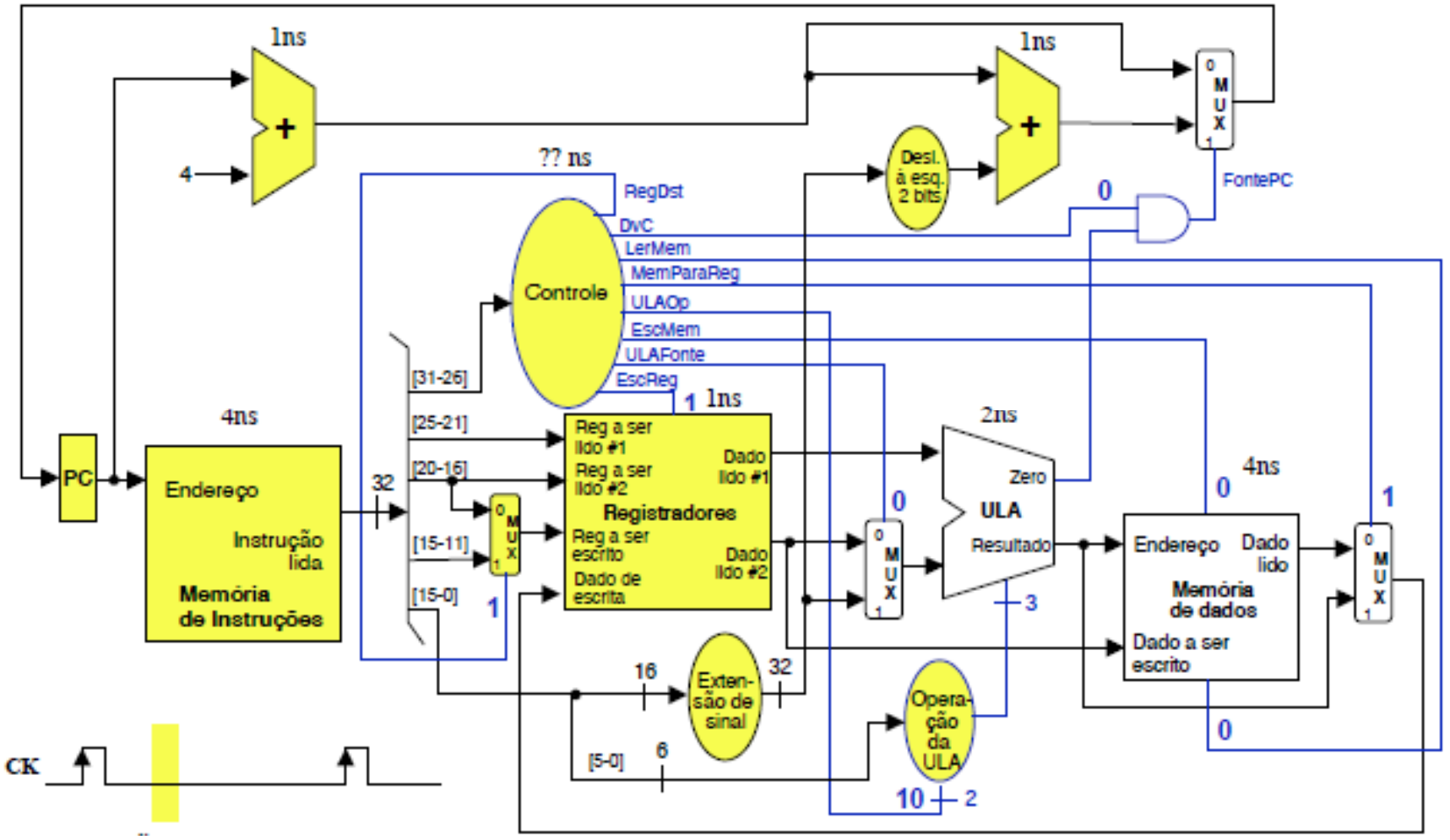
- Busca da instrução e cálculo de PC+4



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

Execução de uma Instrução Tipo R

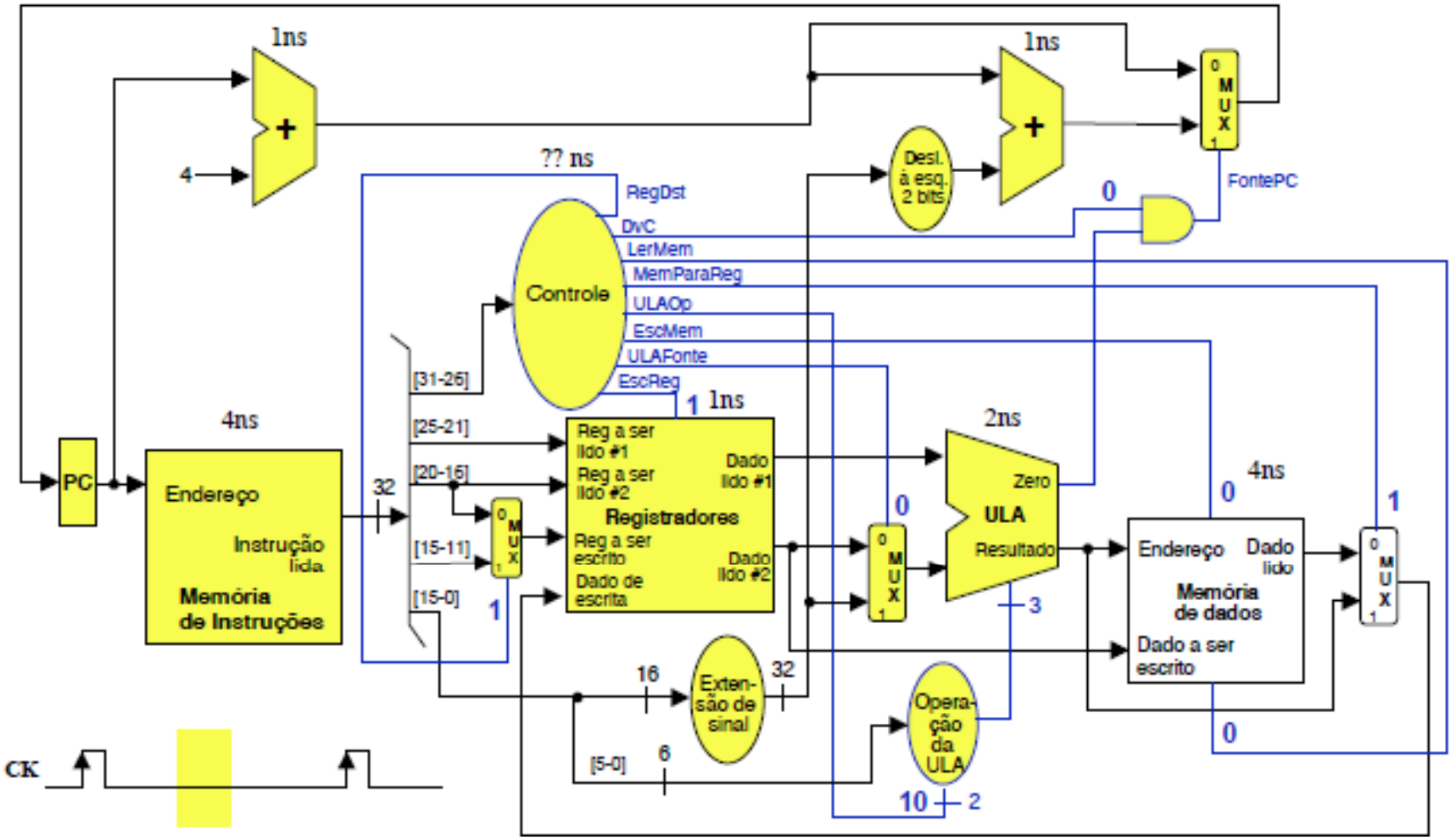
- Leitura de Rs e Rt e geração sinais de controle



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

Execução de uma Instrução Tipo R

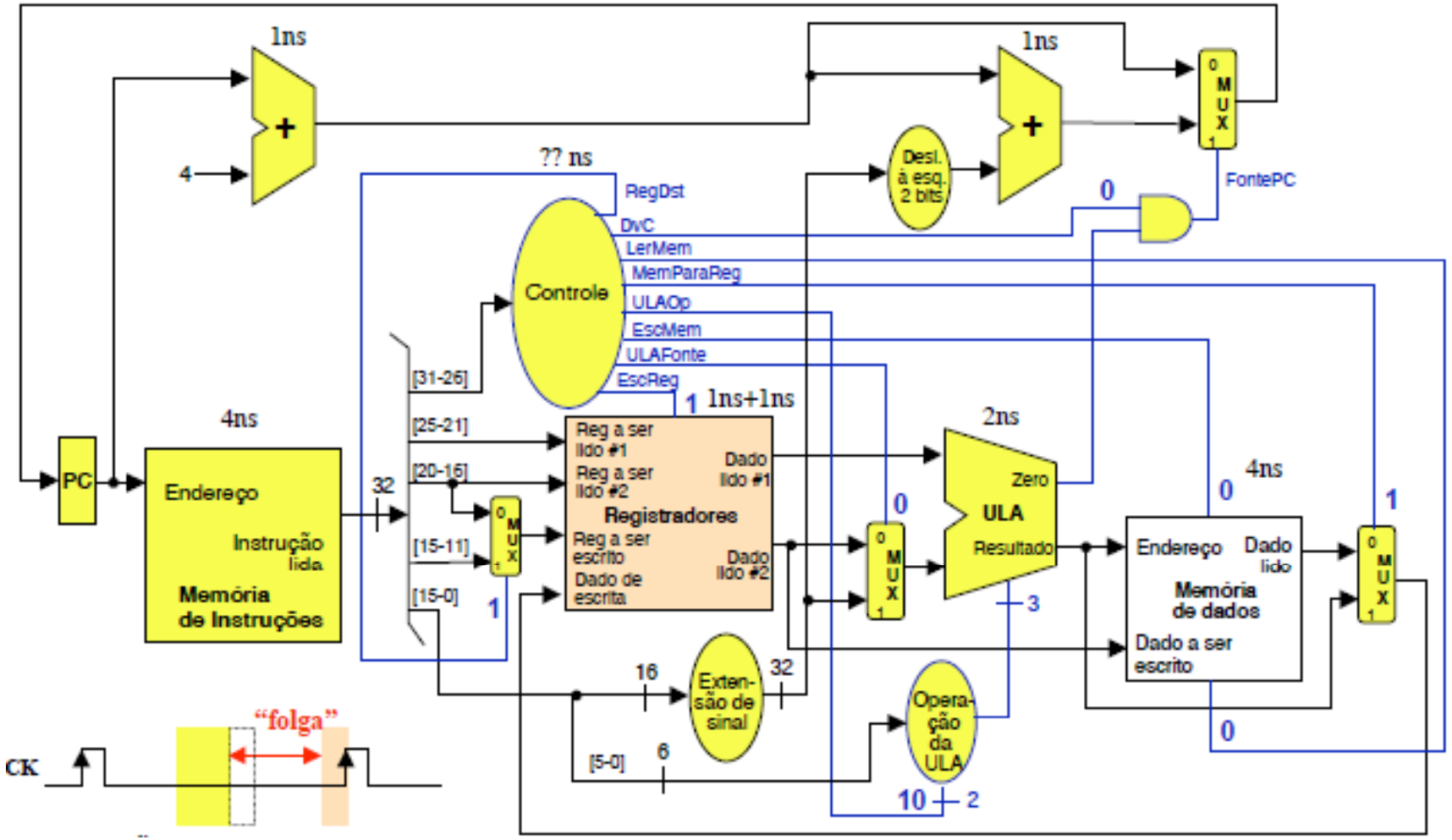
- Operação na ULA (depende de "funct")



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

Execução de uma Instrução Tipo R

- Escrita no registrador-destino



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

Execução de uma Instrução lw

- Seja a instrução tipo I 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)

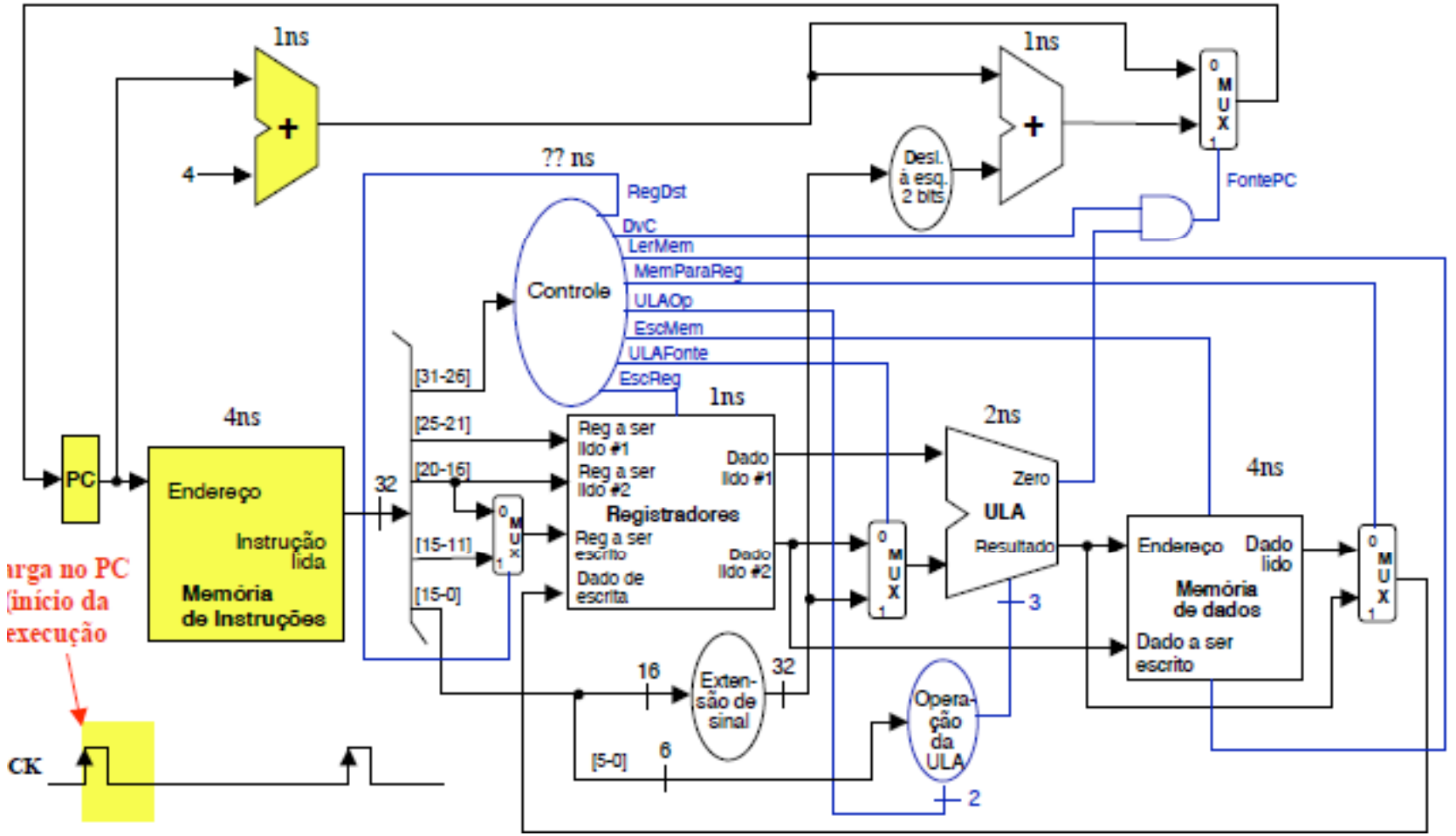
4. Acesso à memória de dados para uma leitura (endereço = resultado da ULA)

5. Escrita (do valor lido da memória de dados) no registrador destino ($t1, que neste caso corresponde ao campo Rt)

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

Execução de uma Instrução lw

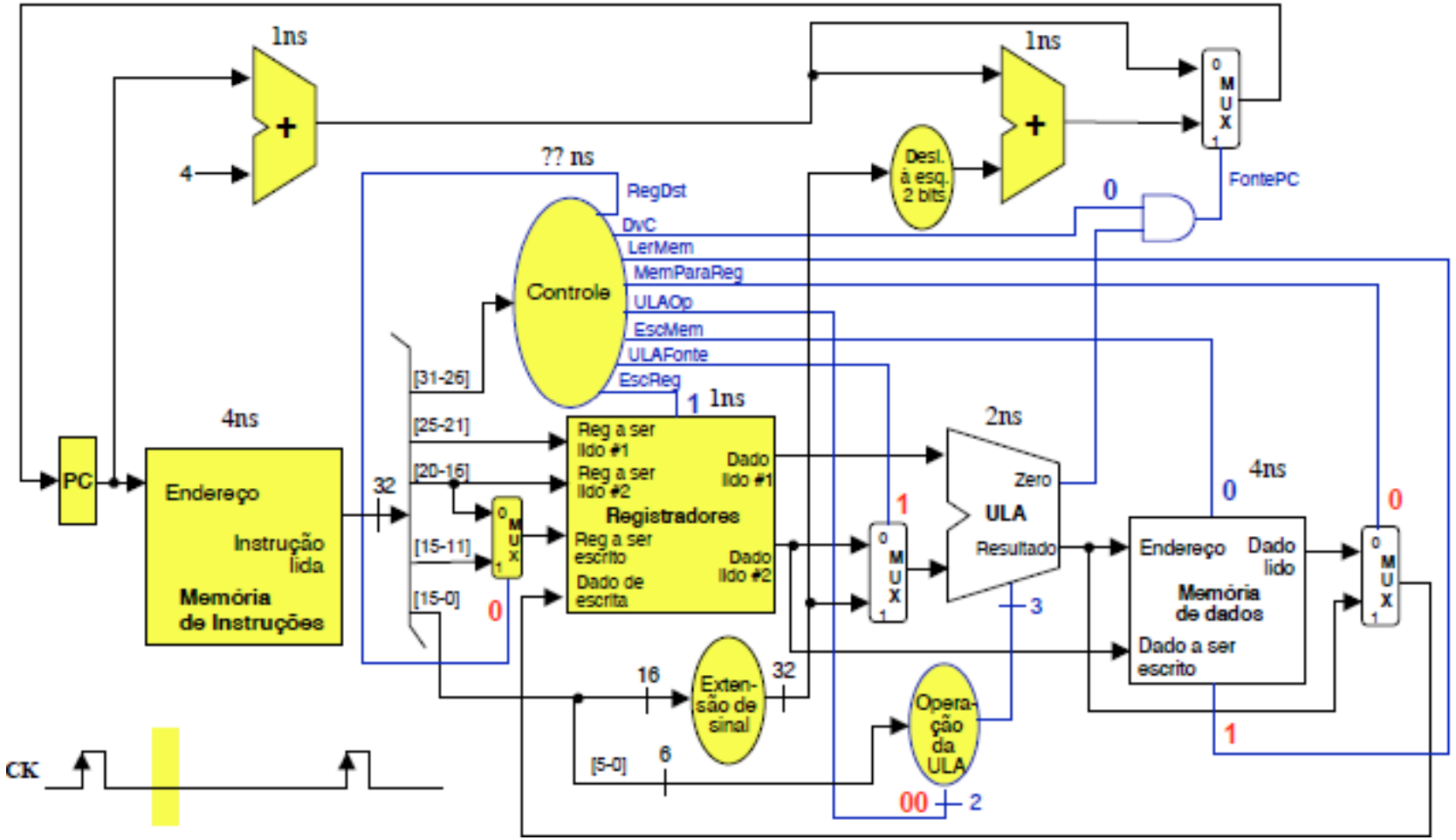
- Busca da instrução e cálculo de PC+4



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

Execução de uma Instrução lw

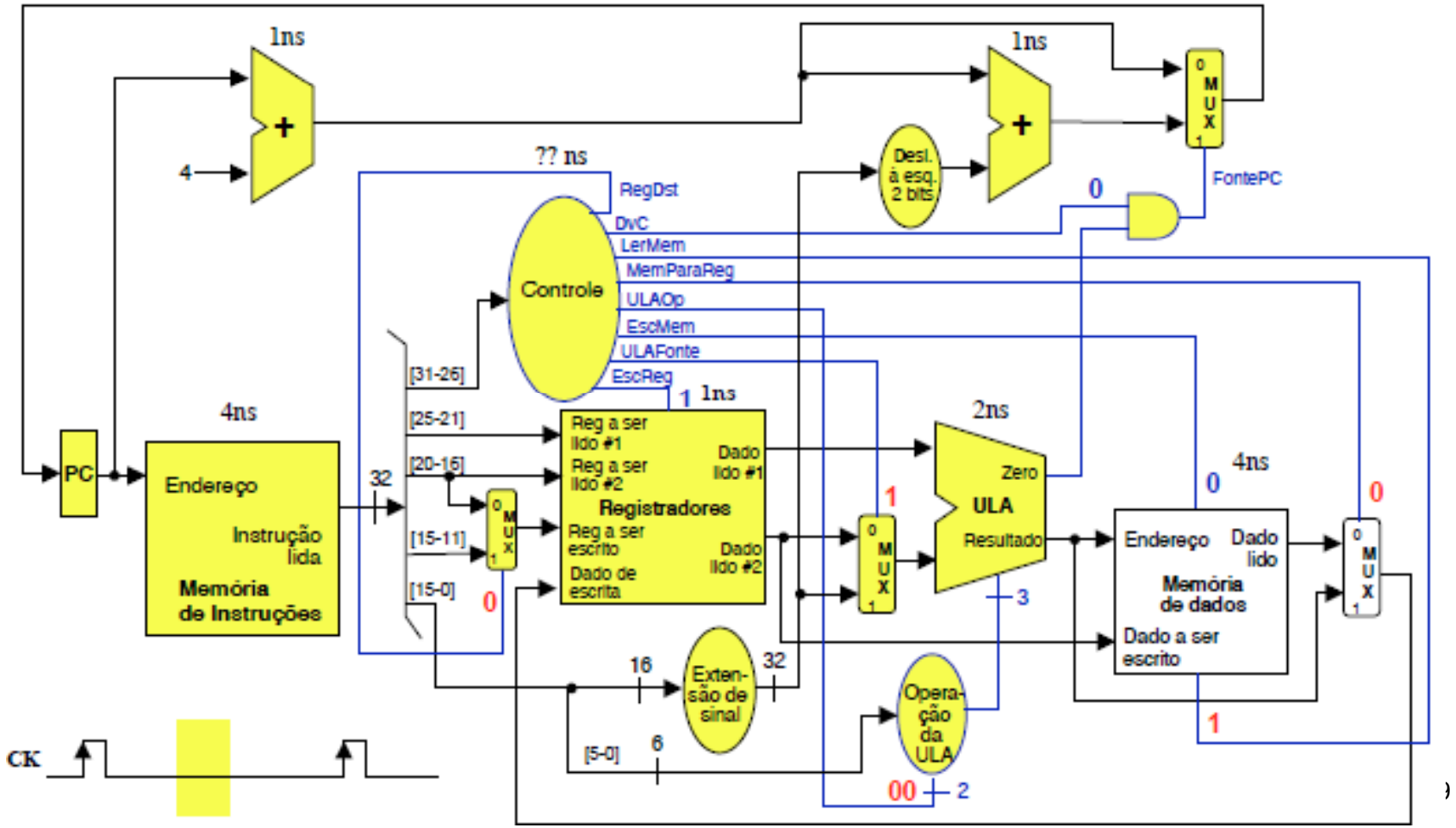
- Leitura de Rs (e Rt) e geração sinais de controle



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

Execução de uma Instrução lw

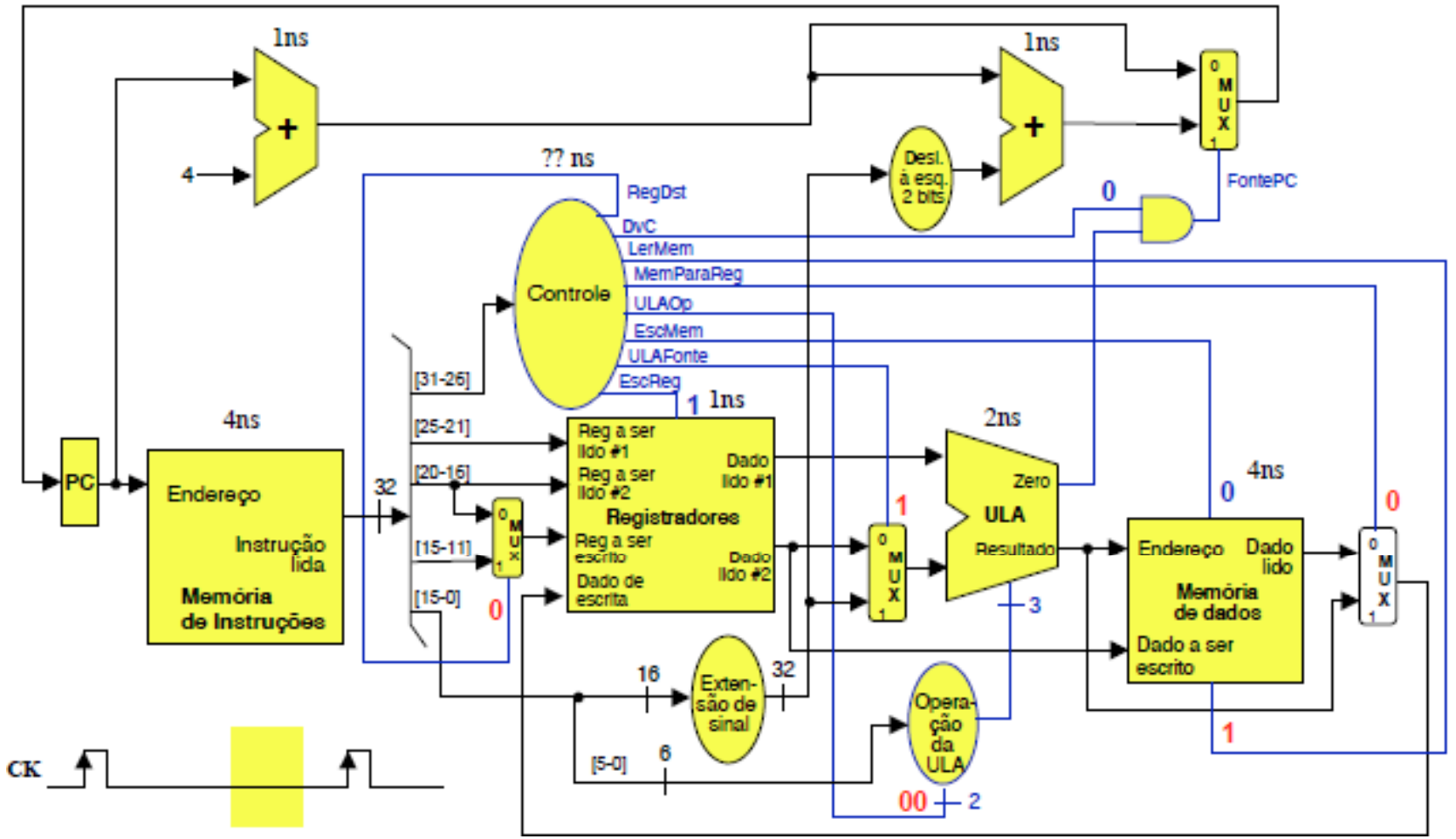
- Cálculo do endereço usando a ULA (adição)



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

Execução de uma Instrução lw

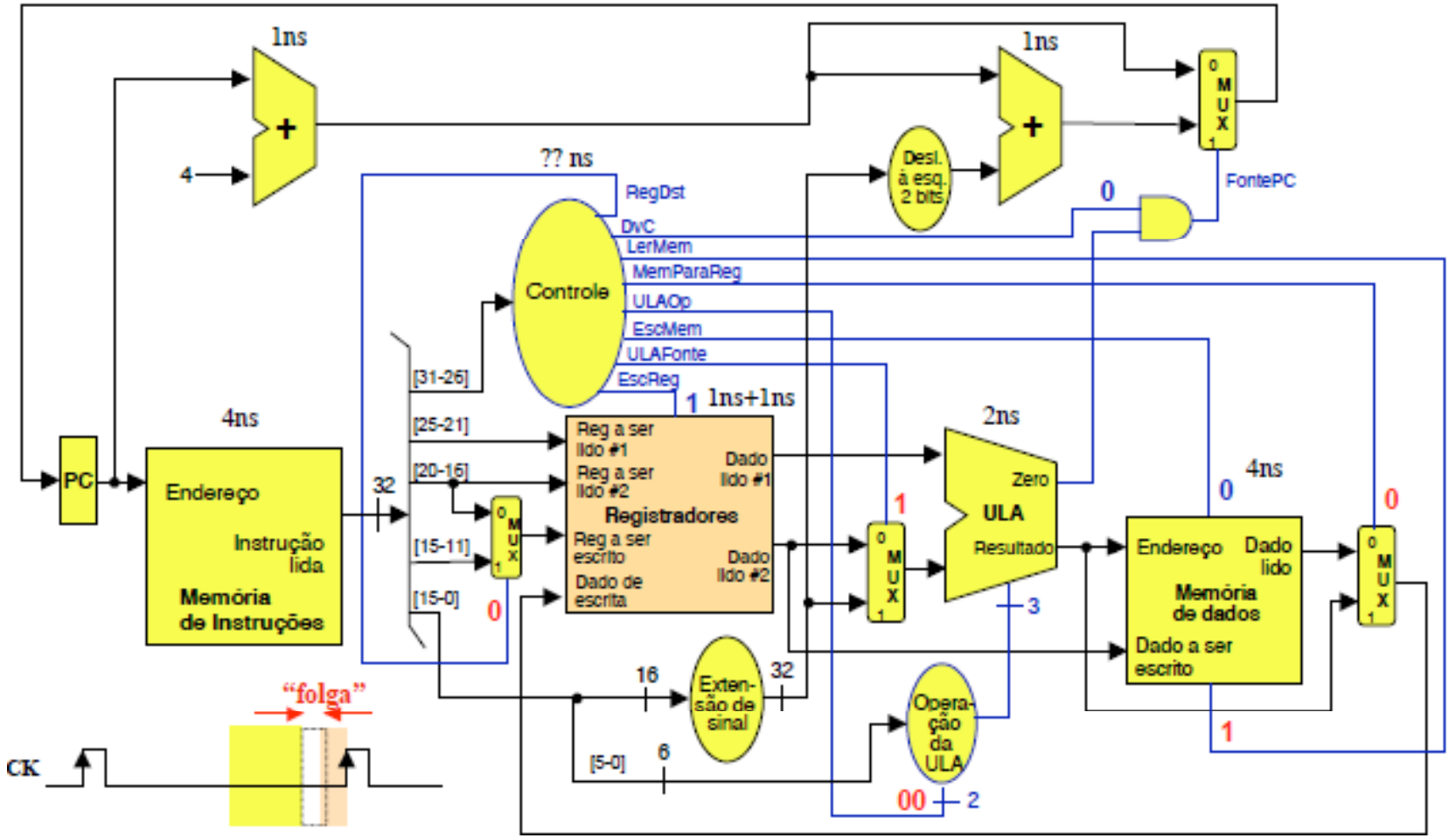
- Acesso à memória de dados para uma leitura



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

Execução de uma Instrução lw

- Escrita no registrador-destino



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

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				

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

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

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

Desempenho de Processadores

$$T_{exec} = N_{instr} \times CPI \times T_{clock}$$

- Instruções por program (Instruction Count – IC): depende do código fonte, compilador e ISA

- Ciclos por instrução (CPI): depende da ISA e microarquitetura. Na implementação monociclo Ciclo de clock tem mesma duração para todas instruções (CPI = 1)

- Tempo por ciclo: depende da microarquitetura e da tecnologia básica de fabricação do chip

Conclusões

- Vimos como é possível construir um datapath relativamente simples para executar instruções da arquitetura MIPS
- Os mesmos princípios se aplicam para se implementar outras instruções
- O datapath apresentado pode ser otimizado (para execução mais rápida) em vários aspectos
- Uma de suas limitações é o fato de ser mono-ciclo, ou seja, executar qualquer instrução em um ciclo de clock, porém esse ciclo é bastante demorado.
- Solução: implementação multi-ciclo desse datapath, mais conhecida como organização em pipelining.

Conclusões

- A duração de um ciclo de clock deve ser longa o bastante para comportar o caminho mais longo no processador.
- A penalidade por utilizar um projeto uniciclo é significativa, mas pode ser considerada aceitável para um conjunto bastante reduzido de instruções – foi explorada inicialmente em computadores com conjuntos bastante simples de instruções.
- Observação:
  - Como o ciclo de relógio é determinado pelo atraso de pior caso entre todas as instruções, é absolutamente inútil o uso de técnicas que reduzem o atraso do caso comum, mas que não trazem qualquer otimização do pior caso.