

# Arquitectura de Sistemas e Computadores II

## 3ª Frequência e Exame

Departamento de Informática  
Universidade de Évora

8 de Janeiro de 2019

- Os símbolos à esquerda de cada pergunta identificam a prova ou provas a que ela pertence:
  - ♣ assinala as perguntas do exame;
  - ◇ assinala as perguntas da frequência.
- O exame está cotado para 20 valores. A frequência está cotada para 15 valores.
- Indique todos os cálculos efectuados

### Perguntas rápidas

- ♣ 1. [0,5 valores] Sabendo que a execução de um programa no processador  $X$  necessita do dobro dos ciclos de relógio da execução do mesmo programa no processador  $Y$ , pode concluir que o desempenho de  $X$  para esse programa é inferior ao de  $Y$ ?
- ♣ 2. [0,5 valores] Quais são as técnicas usadas, no *pipeline* MIPS de cinco andares, para garantir que uma instrução não usa o conteúdo desactualizado de um registo?
- ♣ ◇ 3. [0,5 valores] Se a dimensão dos blocos de cache for aumentada, o número de *compulsory misses* deverá aumentar, manter-se ou diminuir?
- ◇ 4. [0,5 valores] Em que estratégia para lidar com as operações de escrita numa cache é necessário associar um *dirty bit* aos blocos na cache?
- ♣ ◇ 5. [0,5 valores] Uma página virtual de um programa tem de residir na mesma página física durante toda a execução do programa?
- ◇ 6. [0,5 valores] A comunicação entre processos em sistemas de multiprocessamento de memória partilhada é explícita ou pode ser implícita?

### Desempenho

- ♣ 7. No processador  $A$ , cujo relógio funciona a uma frequência de 2 GHz, o CPI do programa  $P$  é 3,6.
  - (a) [2 valores] Se a distribuição das instruções do programa for a apresentada abaixo, qual o CPI das instruções de salto?

Classe	Aritméticas	Acesso à memória	Saltos
%	40	40	20
CPI	2	5	?

- (b) [2,5 valores] Seja  $B$  um processador que implementa a mesma arquitectura que  $A$ . No processador  $B$ , cujo relógio funciona a uma frequência de 4 GHz, o CPI de  $P$  é 4,8.

Calcule o *speedup* que se obtém quando  $P$  é executado em  $B$  em relação à sua execução em  $A$ .

(Note que os CPI da alínea anterior não se aplicam ao processador  $B$ .)

(CONTINUA...)

## Implementação MIPS monociclo

- ♣ 8. [4 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução *auil* (*add upper immediate*), que é uma instrução tipo-I com três argumentos:

auil rt, rs, immediate		<table><tr><td>auil</td><td>rs</td><td>rt</td><td>immediate</td></tr></table>	auil	rs	rt	immediate
	auil	rs	rt	immediate		
bits	6	5	5	16		

Esta instrução coloca no registo *rt* o valor obtido somando *immediate*  $\ll$  16 ao valor contido no registo *rs*.

- Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?
- Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- Quais os valores que os vários sinais de controlo deverão ter durante a execução desta instrução?  
(Não precisa de indicar o valor de *ALUOp*, basta dizer qual será a operação executada pela ALU durante a execução desta instrução.)
- Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

## Pipeline MIPS de 5 andares

- ♣ 9. [2 valores] Simule a execução do código à direita num processador com *forwarding*, com decisão dos saltos condicionais no andar ID, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, assumindo que o salto condicional não é efectuado. Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.
- |    |             |  |
|----|-------------|--|
| 1. | <i>add</i>  | <i>\$t0</i> , <i>\$a1</i> , <i>\$a1</i>  |
| 2. | <i>add</i>  | <i>\$t0</i> , <i>\$t0</i> , <i>\$t0</i>  |
| 3. | <i>addu</i> | <i>\$t1</i> , <i>\$a0</i> , <i>\$t0</i>  |
| 4. | <i>lw</i>   | <i>\$t2</i> , 0( <i>\$t1</i> )           |
| 5. | <i>lw</i>   | <i>\$t2</i> , 4( <i>\$t2</i> )           |
| 6. | <i>sw</i>   | <i>\$t2</i> , 4( <i>\$a2</i> )           |
| 7. | <i>beq</i>  | <i>\$t2</i> , <i>\$a3</i> , <i>salta</i> |
| 8. | <i>or</i>   | <i>\$v0</i> , <i>\$0</i> , <i>\$t2</i>   |

## Cache

10. Considere um sistema com uma cache com dois níveis em que, para um programa, a cache de primeiro nível apresenta uma *miss rate* de 4% e a de segundo nível apresenta uma *miss rate* de 50%.

- ♣ ◇ (a) [2,5 valores] Se o *hit time* da cache de primeiro nível for 3 ciclos, o *hit time* da cache de segundo nível for 20 ciclos, e o tempo de acesso à memória física for 120 ciclos, qual o tempo médio de acesso à memória (em ciclos) na presença e na ausência da cache de segundo nível?
- ◇ (b) [2 valores] Mantendo-se a *miss rate* da cache de primeiro nível, qual deveria ser a *miss rate* da cache de segundo nível para se obter uma *miss rate* global de 1,5%?

## Memória virtual

- ♣ ◇ 11. [3 valores] Considere um sistema em que a dimensão das páginas de memória é de 4 KB, os endereços virtuais têm 40 bits, os endereços físicos têm 32 bits, o TLB é *direct mapped* e tem espaço para 16 blocos com uma tradução cada, e em que é acedido o endereço virtual 00 1000 2308<sub>16</sub>.

Apresente o conteúdo da posição do TLB em que a tradução relativa àquele acesso é inserida e indique o seu índice, assim como o conteúdo da posição correspondente da tabela de páginas. Assuma que a página virtual é colocada na página física 734.

- ◇ 12. [1,5 valores] Aponte duas vantagens do uso de memória virtual.

## Multiprocessamento

13. O código abaixo vai ser executado nos dois processadores MIPS de um multiprocessador de memória partilhada. No início da execução destas instruções, o endereço guardado no registo *\$a0* é 1000 0000<sub>16</sub> nos dois processadores e a palavra nesse endereço tem o valor 200.

1.	<i>lw</i>	<i>\$t1</i> , 0( <i>\$a0</i> )
2.	<i>addi</i>	<i>\$t1</i> , <i>\$t1</i> , 10
3.	<i>sw</i>	<i>\$t1</i> , 0( <i>\$a0</i> )

- ♣ ◇ (a) [2 valores] Mostre que, depois da execução destas instruções nos dois processadores, o valor no endereço 1000 0000<sub>16</sub> pode não ser 220.
- ◇ (b) [2 valores] Altere o código apresentado de modo a que quando for executado em paralelo nos dois processadores, o valor naquela posição de memória seja sempre 20 mais o valor original.

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

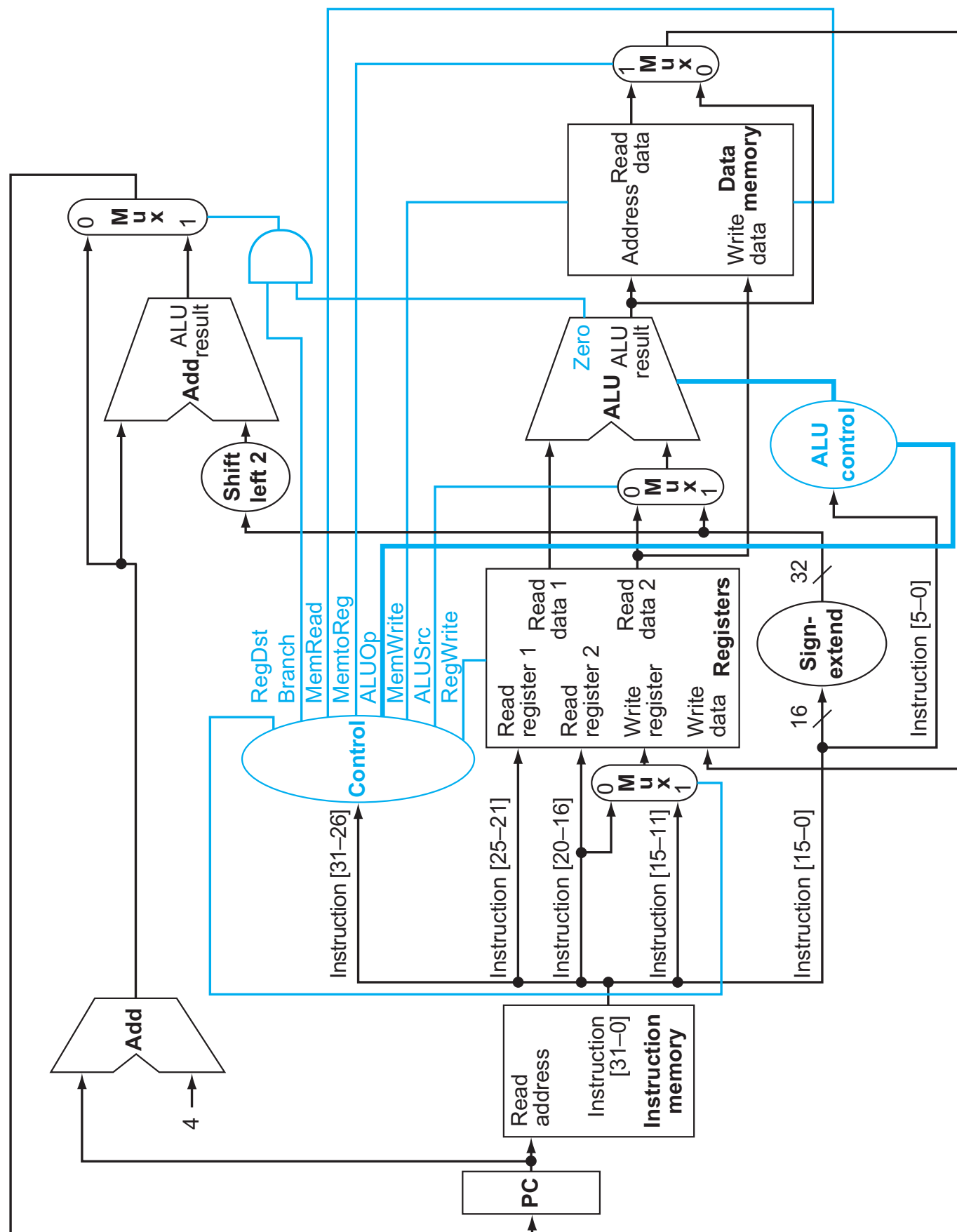


Figura 1: Diagrama de blocos da implementação MIPS monociclo

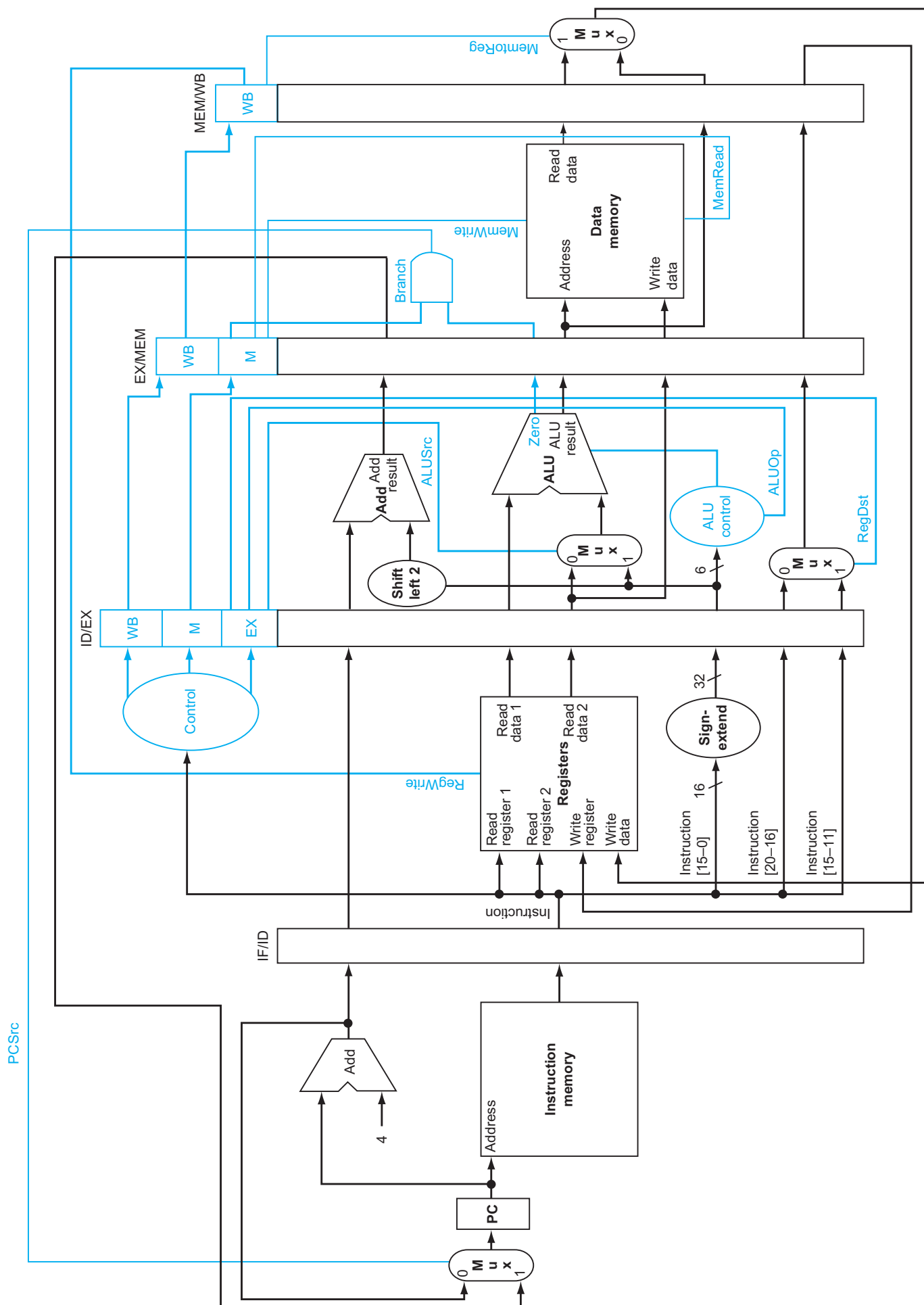


Figura 2: Diagrama de blocos do *pipeline* MIPS