

Arquitectura de Sistemas e Computadores II

3ª Frequência e Exame

Departamento de Informática
Universidade de Évora

11 de Janeiro de 2017

- 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 2ª frequência.
- Indique todos os cálculos efectuados

Perguntas rápidas

- ♣ 1. [0,5 valores] Sabendo que o CPI de um programa no processador X é inferior ao CPI do mesmo programa no processador Y , pode concluir que o desempenho de X é superior ao de Y ?
- ♣ 2. [0,5 valores] A execução *pipelined* é uma técnica que aumenta o *throughput* ou que diminui o tempo que uma instrução demora a executar?
- ♣ ◇ 3. [0,5 valores] Se uma cache *direct mapped* e uma cache *fully associative* tiverem a mesma capacidade, em qual é de esperar uma menor *miss rate* durante a execução do mesmo programa?
- ♣ ◇ 4. [0,5 valores] Na memória virtual, qual a estratégia usada para lidar com as operações de escrita, *write-through* ou *write-back*?
- ◇ 5. [0,5 valores] Qual a excepção que ocorre quando é acedida uma página virtual que não se encontra em memória física?
- ◇ 6. [0,5 valores] A comunicação entre processos em sistemas de multiprocessamento de memória partilhada é explícita ou pode ser implícita?

Desempenho

- ♣ 7. Na execução do programa P no computador A , cujo relógio funciona a uma frequência de 2 GHz, são executadas 10^9 instruções, com a seguinte distribuição:

Classe	Aritméticas	Acesso à memória	Saltos
%	40	40	20
CPI	1	4	2

- (a) [1,5 valores] Quanto tempo demora a execução de P em A ?
- (b) [1,5 valores] A que frequência deverá o funcionar o relógio de um computador B , em que o CPI do programa é 20% superior, para que o programa execute em metade do tempo?

(CONTINUA...)

Implementação MIPS monociclo

- ♣ 8. [4 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução *nori* (*nor immediate*), que é uma instrução tipo-I, com três argumentos:

<code>nori rt, rs, immediate</code>	<code>nori</code>	<code>rs</code>	<code>rt</code>	<code>immediate</code>
bits	6	5	5	16

Esta instrução calcula o OU-lógico negado (*NOR*) *bit* a *bit* entre o valor no registo *rs* e o valor *immediate* estendido com zeros, e guarda o resultado no registo *rt*.

- (a) Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?
- (b) Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- (c) 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.)
- (d) 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 seguinte 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 valor inicial no registo *\$a0* é 1. 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.          or    $v0, $0, $0
2.          beq   $a1, $0, fim
3.  ciclo:  lw    $t2, 0($a0)
4.          add   $v0, $v0, $t2
5.          addiu $a0, $a0, 4
6.          addi  $a1, $a1, -1
7.          bne   $a1, $0, ciclo
8.  fim:     jr    $ra
```

Quantos ciclos de relógio são necessários para executar o código nas condições acima?

Quantos ciclos de relógio seriam necessários para executar o código se o ciclo (instruções 3 a 7) fosse executado 100 vezes?

Cache

Considere que uma palavra tem 32 bits e que os endereços seguintes são acedidos pela ordem indicada:

48 56 52 48 120 60 24 16

- ♣ ◇ 10. [3 valores] Simule o funcionamento de uma cache *direct-mapped*, com 8 palavras e blocos de 2 palavras, para a sequência de acessos indicada. Assuma que a cache inicialmente está vazia e, para cada acesso, indique a palavra acedida, o número do bloco a que pertence a palavra, o índice da posição da cache que irá ocupar, o *tag*, se há um *hit* ou um *miss* e, quando aplicável, o número do bloco que será substituído. Apresente o conteúdo final da cache, tão completo quanto possível, e calcule a *miss rate* verificada.
- ◇ 11. [2 valores] No índice 5 de uma cache com 16 conjuntos e blocos com 4 palavras, encontra-se um bloco cujo *tag* tem o valor 20₁₆. Qual o número desse bloco e quais as palavras que pertencem ao bloco? (Dê a resposta na base que preferir.)

(CONTINUA...)

Memória virtual

- ♣ ◇ **12.** Considere um sistema MIPS em que os endereços virtuais têm 42 bits, os endereços físicos têm 38 bits e a dimensão das páginas de memória é de 64KB. Num momento da execução de um programa, a sua tabela de páginas apresenta o conteúdo (parcialmente) mostrado:

Tabela de páginas

	Dirty	Pág. física
		...
29	0	19
30	0	12
31	1	5
32	0	DISCO
33	0	23
		...

- (a) [2 valores] Assumindo que as páginas virtuais foram acedidas pela ordem 29, 30, 31 e 33, apresente o conteúdo do TLB do sistema (*direct-mapped*, com 4 blocos de uma tradução).
- (b) [2 valores] Descreva, tão detalhadamente quanto conseguir, o que acontece no sistema quando é feito um acesso ao endereço virtual $20\ 33\ F0_{16}$, até ao momento em que é identificada a página física que irá ser usada.

Multiprocessamento

- ◇ **13.** [2 valores] Qual deverá ser a duração da parte sequencial de um programa para permitir obter um *speedup* de 50 quando o programa é executado em paralelo em 99 processadores?
- ♣ ◇ **14.** [2 valores] Num multiprocessador de memória partilhada é executado um programa paralelo, sendo as instruções seguintes executadas no processador indicado:

Processador 1	Processador 2	Processador 3
$x = x * 5;$	$x = x + 4;$	$x = 17$

Liste as possíveis sequências de valores que a variável x poderá assumir durante a execução deste código se, inicialmente, o valor de x for 2.

Nome: _____ Número: _____

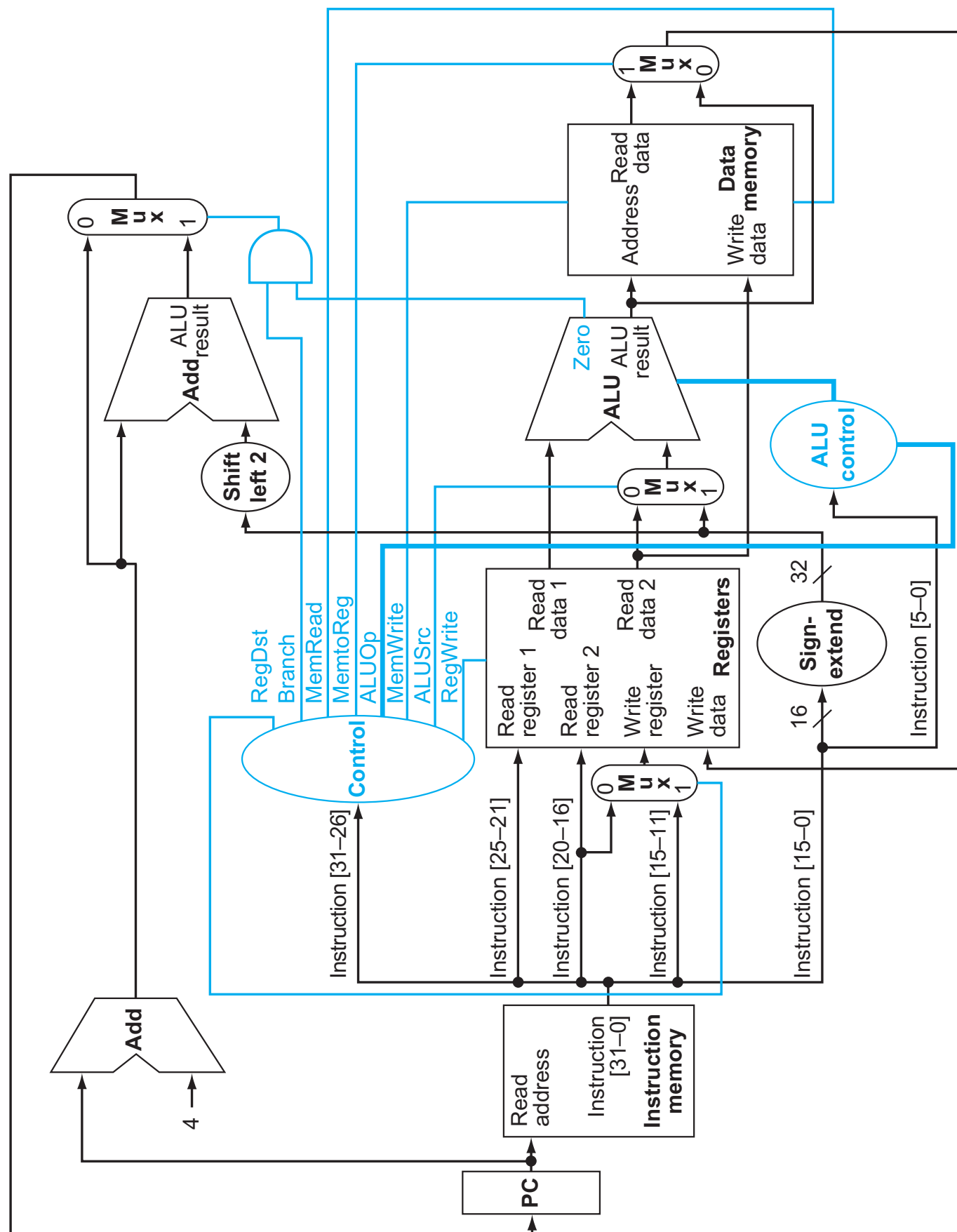


Figura 1: Diagrama de blocos da implementação MIPS monociclo

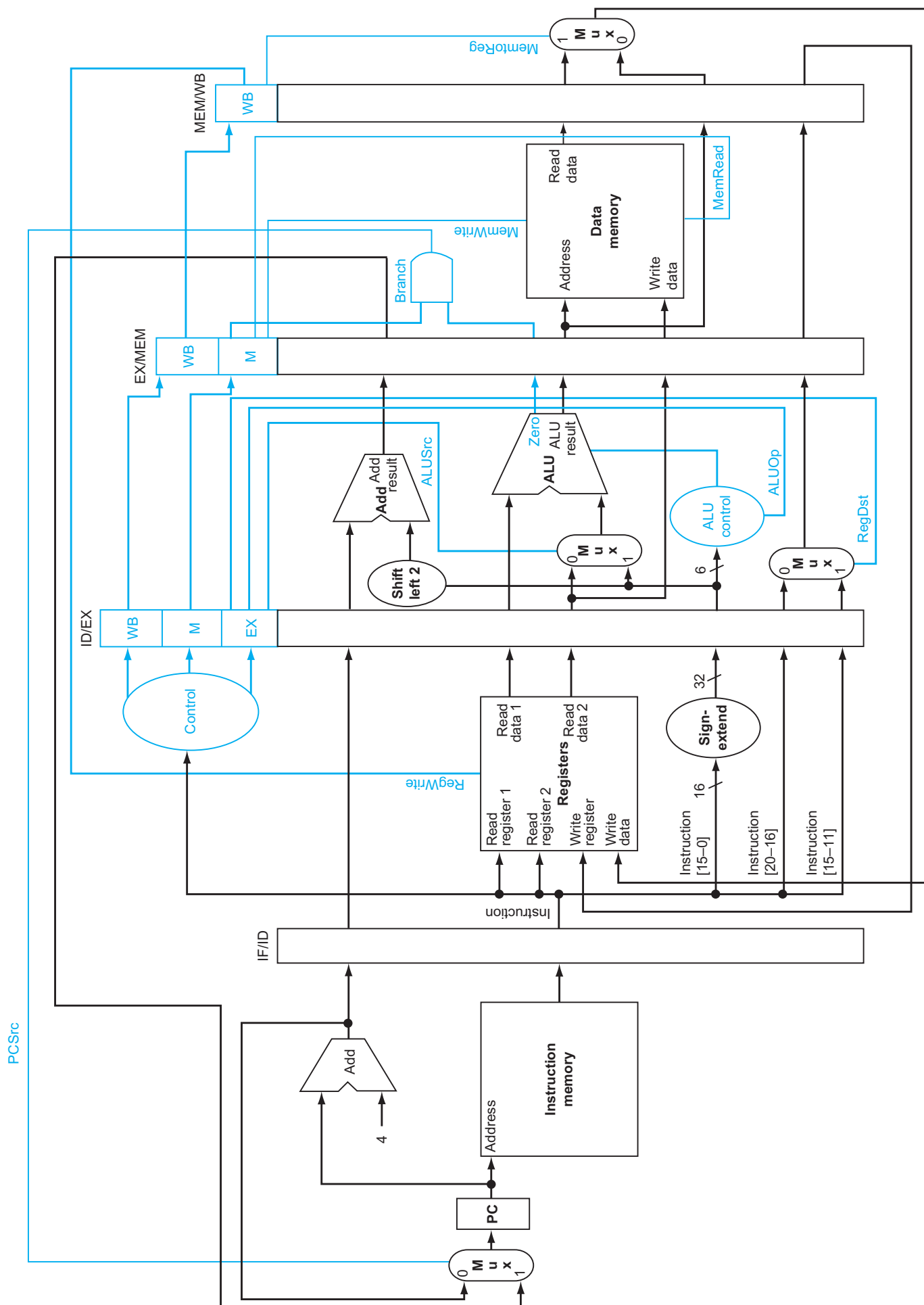


Figura 2: Diagrama de blocos do *pipeline* MIPS