Arquitectura de Sistemas e Computadores II
2ª Frequência

Departamento de Informática
Universidade de Évora

13 de Dezembro de 2016

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] Idealmente, quantas vezes mais rápida será a execução de um programa num *pipeline* de cinco andares em relação à execução numa implementação monociclo?

2. [1 valor] Quantos ciclos de relógio demora a execução da instrução `add` no *pipeline* MIPS de cinco andares?

3. [1 valor] Se a instrução que está no andar EX do *pipeline* MIPS gera uma excepção, as instruções que se encontram nos andares MEM e WB terminam a sua execução normalmente ou essa execução é interrompida?

4. [1 valor] Quando existe execução de instruções fora de ordem, o resultado do programa pode ser diferente de quando as instruções são executadas estritamente por ordem?

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 1. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

5. Considere que o significado e o efeito do código MIPS seguinte são exactamente aqueles que teria se fosse executado na implementação monociclo do processador (onde não existem *delay slots*). No fim da execução do código, os valores presentes nos registos usados não são importantes.

```
1.           beq   $a0, $0, fim
2.    ciclo: lw    $t0, 0($a1)
3.           lw    $t1, 0($a2)
4.           addu  $t0, $t0, $t1
5.           sw    $t0, 0($a3)
6.           addiu $a1, $a1, 4
7.           addiu $a2, $a2, 4
8.           addiu $a3, $a3, 4
9.           addi  $a0, $a0, -1
10.          bne   $a0, $0, ciclo
11.   fim:   jr    $ra
```

(a) [3 valores] Identifique todas as dependências (de dados) existentes no código apresentado.

(b) [4 valores] Simule a execução do código 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.

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 2 a 10) fosse executado 1000 vezes?

(CONTINUA...)

(c) [2,5 valores] Altere o código apresentado, reordenando as instruções e, se considerar útil, modificando o *offset* das instruções de acesso à memória, de modo a eliminar o maior número possível de atrasos e de ciclos desperdiçados durante a sua execução no *pipeline* com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

6. [2 valores] Durante a execução de um programa, o conteúdo do *pipeline* MIPS da Figura 1, num determinado ciclo de relógio, é o seguinte:

IF	ID	EX	MEM	WB
`or $t5, $t4, $t6`	`addu $t2, $t3, $t7`	`addi $t1, $t0, 5`	`beq $0, $0, l1`	`sw $s0, 0($s1)`

Diga qual deverá ser o valor dos sinais de controlo *em uso*, neste ciclo, em cada um dos andares, e qual a operação realizada pela ALU. (Não é necessário referir o valor de ALUOp.)

7. [2 valores] Se as latências dos andares do *pipeline* forem as apresentadas abaixo, qual o andar que determina a frequência máxima a que poderá funcionar o relógio do processador e qual é essa frequência?

Andar	IF	ID	EX	MEM	WB
Latência	360 ps	235 ps	385 ps	365 ps	255 ps

ILP

8. [2,5 valores] Organize o código original da pergunta 5, introduzindo as alterações que considerar convenientes, para ser executado no *pipeline* MIPS *double issue* (com *forwarding*, decisão dos saltos condicionais no andar ID, previsão perfeita e sem *delay slot*), em que cada *issue packet* pode conter uma instrução aritmética ou de salto, e uma instrução de acesso à memória, de modo a não haver a necessidade da introdução de atrasos durante a sua execução.

Nome: _____ Número: _____

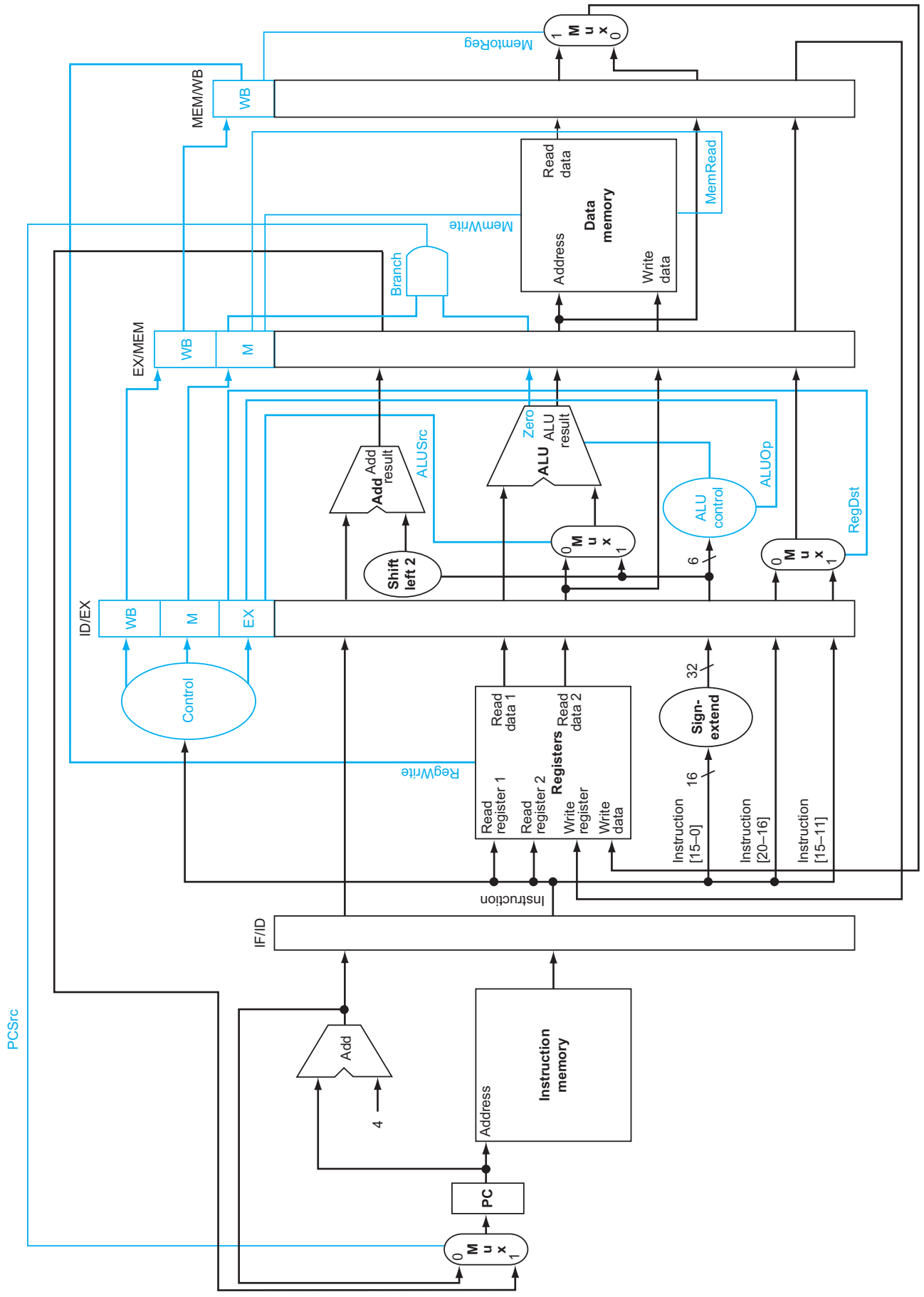

Figura 1: Diagrama de blocos do *pipeline* MIPS