

Arquitectura de Sistemas e Computadores II

2ª Frequência

Departamento de Informática
Universidade de Évora

28 de Novembro de 2018

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] No *pipeline* MIPS, a leitura dos registos a usar por uma instrução acontece antes, em simultâneo, ou depois da descodificação da instrução?
2. [1 valor] Quando, no *pipeline* MIPS, uma instrução é atrasada um ciclo devido a um conflito de dados que a envolve, é a entrada da instrução no *pipeline* que é atrasada, ou a instrução fica um ciclo adicional nalgum andar do *pipeline*?
3. [1 valor] Se a instrução que está no andar EX do *pipeline* MIPS gera uma excepção, o que acontece à instrução está no andar IF quando a excepção é gerada?
4. [1 valor] Numa cache com *no write-allocate*, quando há um *miss* numa operação de escrita, o conteúdo da cache pode ser alterado devido a essa operação ou fica na mesma?

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, excepto o do registo \$v0.

```
1.          ori    $t1, $0, 0
2.          or     $t0, $t1, $a2
3.  ciclo:  beq    $a1, $0, fim
4.          lw     $t2, 0($a2)
5.          bne    $t2, $a0, copia
6.          addi   $t1, $t1, 1
7.          beq    $0, $0, avanca
8.  copia:  addiu   $t0, $t0, 4
9.          sw     $t2, -4($t0)
10. avanca: addiu   $a2, $a2, 4
11.          addi   $a1, $a1, -1
12.          beq    $0, $0, ciclo
13. fim:    or     $v0, $t1, $0
14.          jr     $ra
```

- (a) [2,5 valores] Identifique todas as dependências (de dados) existentes no código apresentado.

(CONTINUA...)

- (b) [3 valores] Simule a execução do código apresentado 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*, com o salto correspondente à instrução 5 a ser efectuado e o correspondente à instrução 3 a só ser efectuado na segunda vez que a instrução for executada.

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.

- (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*.

ILP

6. [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.

Caches

7. Considere uma cache 3-way *set associative*, com dois conjuntos, blocos de 2 palavras, palavras de 32 bits, e aplicando a estratégia LRU para a escolha do bloco a substituir. O conteúdo dessa cache é parcialmente apresentado na figura seguinte (onde $M[p]$ representa o valor da palavra de memória p):

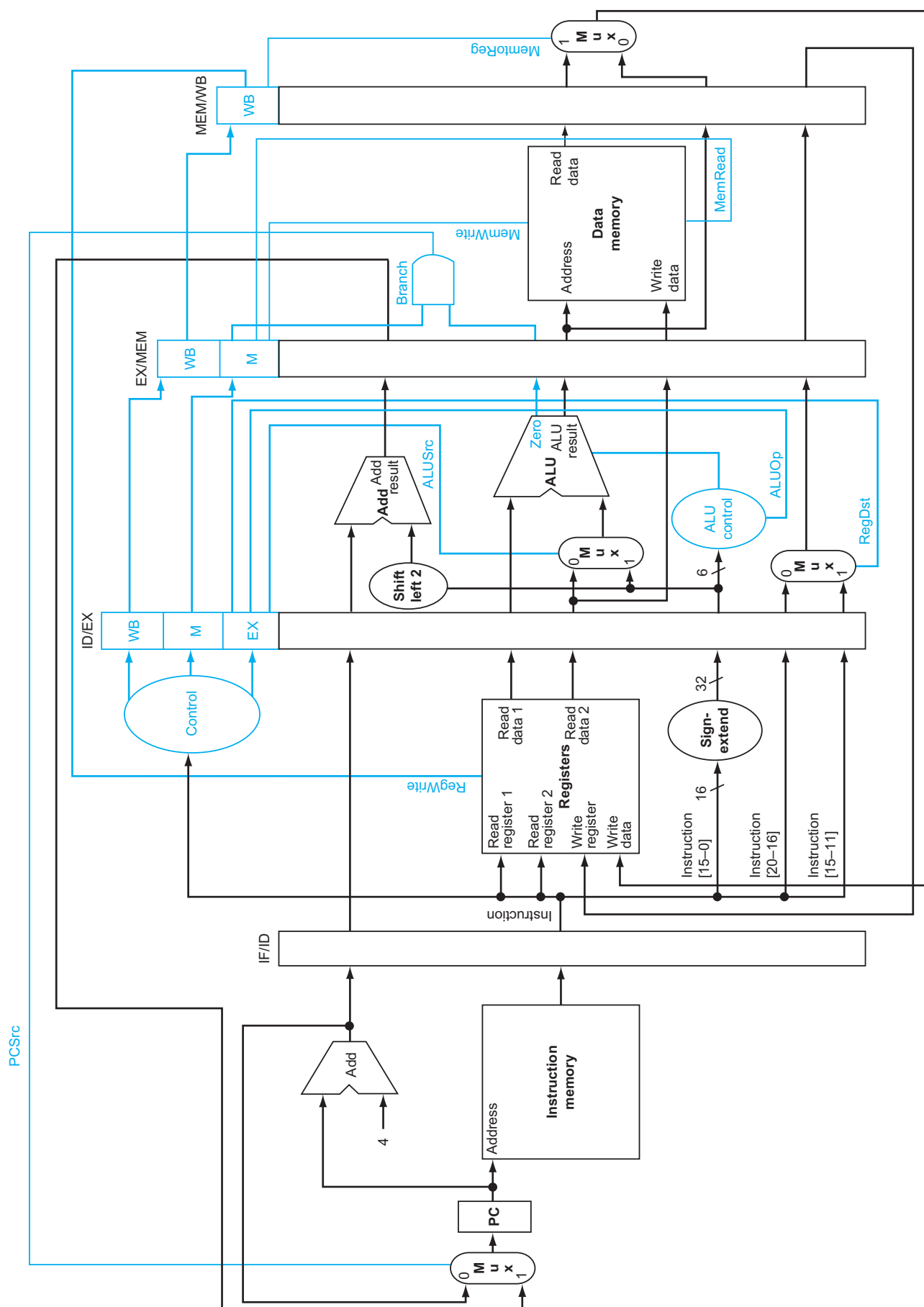
Índice	Valid	Tag	Palavras	Valid	Tag	Palavras	Valid	Tag	Palavras
0	1	6	M[24]	0				6	
1	1	4		1		M[15]	0		

- (a) [2,5 valores] Complete o que puder da figura.
- (b) [3 valores] Simule o funcionamento da cache, a partir do estado apresentado acima, para a sequência de acessos aos endereços seguintes:

103 94 62 40 74

Assuma que o último bloco da cache acedido foi o que se encontra na posição central, no conjunto com índice 1. 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.

Nome: _____ Número: _____

Figura 1: Diagrama de blocos do *pipeline* MIPS