

Arquitectura de Sistemas e Computadores II

1ª Frequência

Departamento de Informática
Universidade de Évora

7 de Novembro de 2013

Indique todos os cálculos efectuados

Desempenho

1. [3 valores] Num computador, cujo relógio funciona à frequência de 2 GHz, as instruções dividem-se em três classes, cada uma com o seu CPI:

Classe	A	B	C
CPI	1	2	3

A execução do código produzido pelo compilador X para um programa leva à execução de 400 milhões de instruções, sendo 30% delas da classe A, 40% da classe B e 30% da classe C.

Na execução do código gerado para o mesmo programa pelo compilador Y, são executadas só 300 milhões de instruções, distribuídas do seguinte modo pelas três classes: 20% da A, 30% da B e as restantes da C.

Qual o compilador que gera o código mais rápido para o programa? E qual o *speedup* obtido com o uso desse compilador em relação ao outro?

Implementação MIPS monociclo

Para este grupo, use como referência a implementação monociclo da Figura 1.

2. Pretende-se incluir a instrução **jr** (*jump register*) na implementação MIPS monociclo. A instrução **jr** é uma instrução tipo-R com um argumento

	0	rs	0	0	0	8
jr rs	bits	6	5	5	5	6

Esta instrução altera o fluxo de execução para a instrução cujo endereço está no registo **rs**.

- [2,5 valores] Quais das unidades funcionais existentes serão usadas para a execução desta instrução e que unidades funcionais é necessário acrescentar?
- [2,5 valores] Que sinais de controlo é necessário acrescentar e quais os valores que os vários sinais de controlo deverão ter para a execução desta instrução?
(Não precisa de indicar o valor de **ALUOp**, basta dizer qual será a função da ALU na execução desta instrução.)

Se considerar necessário fazer alguma alteração ao caminho de dados, apresenta-a na Figura 1.

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 2. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

3. [1,5 valores] Considere, agora, a inclusão da instrução *jr* na implementação *pipelined* do MIPS. Se o valor no registo *rs* for escrito no PC no fim do ciclo de relógio em que a instrução esteve no andar EX, quantos *delay slots* deverão ser usados para o processador não ter de introduzir qualquer atraso no *pipeline*? Porquê?

4. Pretende-se executar o código MIPS seguinte de modo a que o seu efeito seja exactamente o que teria se fosse executado na implementação monociclo do processador. No fim da execução do código, os valores nos registos usados não são importantes.

```
1.          and    $7, $0, $0
2.          beq    $5, $0, fim
3.  ciclo:  lw     $8, 0($4)
4.          add    $7, $7, $8
5.          addiu  $4, $4, 4
6.          addi   $5, $5, -1
7.          bne    $5, $0, ciclo
8.  fim:     sw     $7, 0($6)
```

- (a) [2 valores] Identifique as dependências de dados existentes no código apresentado.
- (b) [2 valores] Simule a execução do código quando só é executada uma iteração do ciclo (a execução termina tendo a instrução 7. sido executada uma única vez). Considere um processador com *forwarding*, com decisão dos saltos condicionais no andar EX, sem *delay slots* e com previsão perfeita do resultado das instruções de salto condicional. 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 fossem executadas 100 iterações do ciclo?

- (c) [2 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 durante a sua execução no *pipeline* com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

5. As latências das várias componentes do *pipeline* são as apresentadas na tabela seguinte:

PC/Registos do <i>pipeline</i>	Memória instruções	Banco registos	ALU	Memória dados	Somadores	<i>Multiplexors</i>	Controlo	Controlo da ALU
10 ps	400 ps	170 ps	330 ps	450 ps	250 ps	30 ps	40 ps	20 ps

Considere que os restantes elementos lógicos têm latência zero.

- (a) [1,5 valores] Nestas condições, qual será a frequência máxima a que processador poderá funcionar?
- (b) [1,5 valores] Quais as unidades funcionais, de controlo, etc., que se encontram no caminho crítico do andar EX? Qual a latência desse caminho?

6. [1,5 valores] Diga, justificando, se será possível um *issue packet* do *pipeline* MIPS com *double issue* estático ser constituído pelas duas instruções

```
add $1, $2, $3    e    lw $1, 0($4).
```

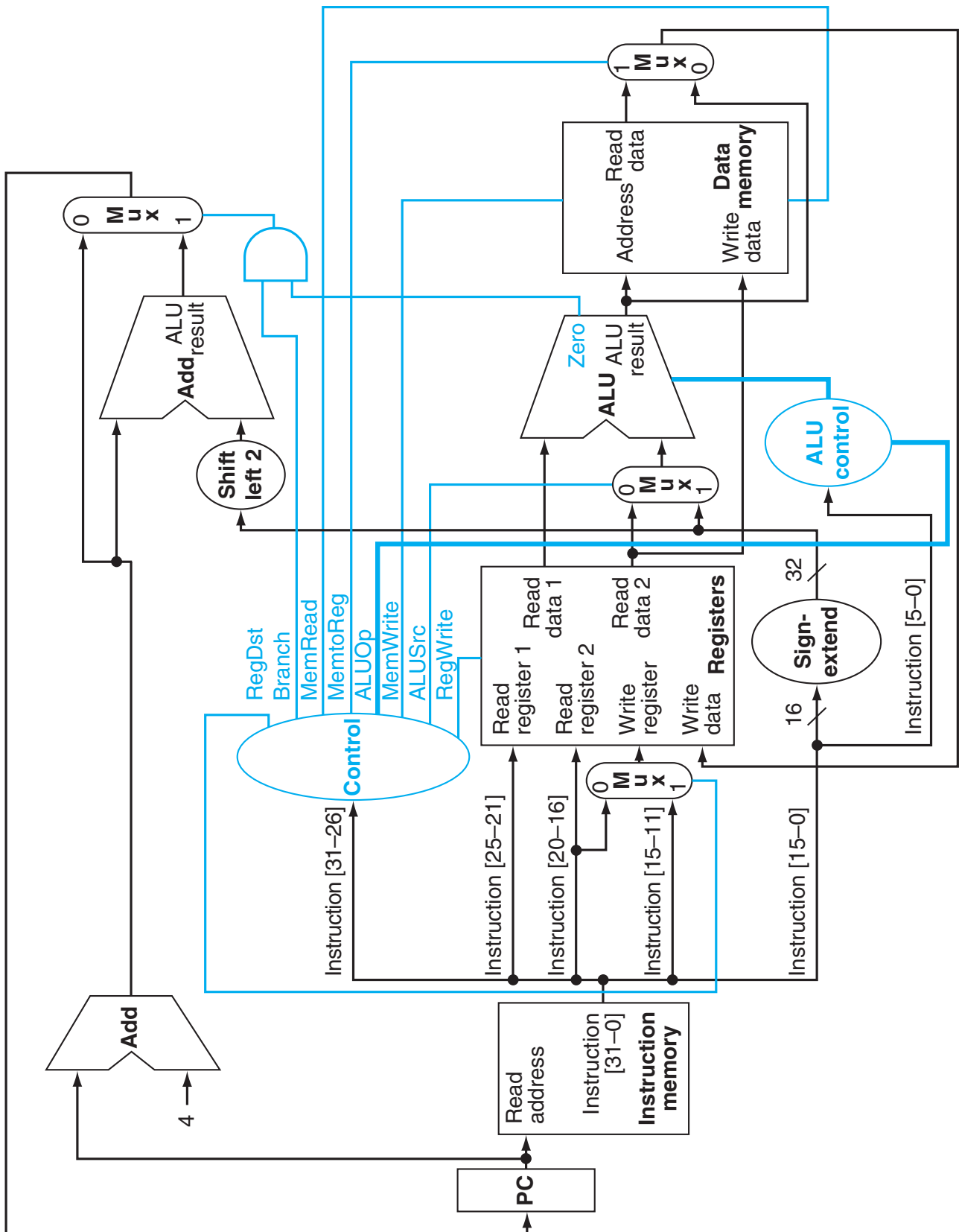


Figura 1: Diagrama de blocos da implementação MIPS monociclo

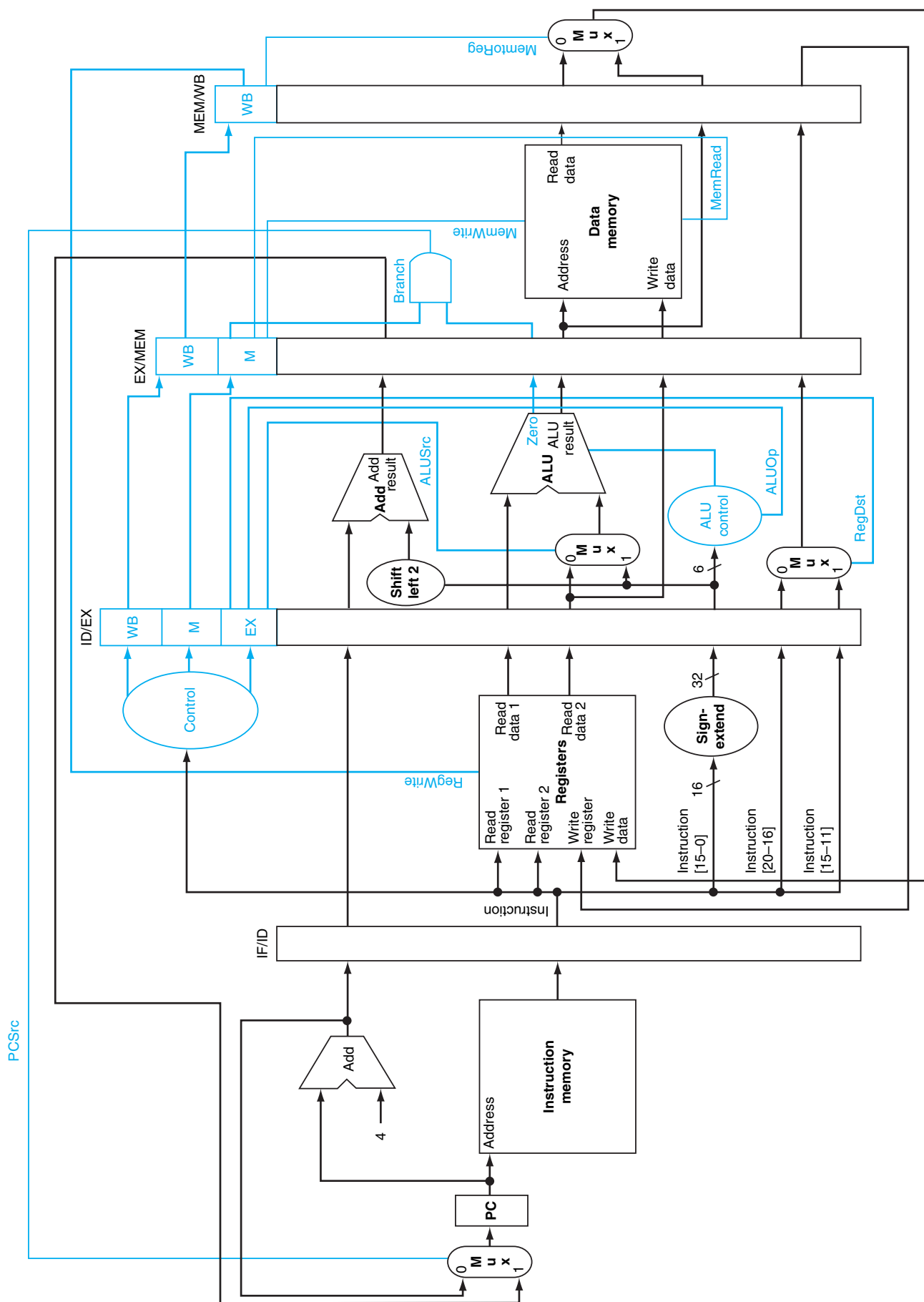


Figura 2: Diagrama de blocos do *pipeline* MIPS