

Arquitectura de Computadores II

3ª Frequência e Exame

Departamento de Informática
Universidade de Évora

5 de Janeiro de 2022

- 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, que serão convertidos para 20 valores.
- Indique todos os cálculos efectuados e todas as fórmulas usadas

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] Indique uma técnica usada, no *pipeline* RISC-V de cinco andares, para garantir que uma instrução não usa o conteúdo desactualizado de um registo.
- ◇ 3. [0,5 valores] Se se mantiver a dimensão da cache e se aumentar o número de blocos por conjunto, o número de *misses* devidos a conflitos deverá aumentar, manter-se ou diminuir?
- ♣ ◇ 4. [0,5 valores] Na estratégia *write-through*, perante uma operação de escrita sobre um bloco que está na cache, onde é efectuada a alteração: só na cache, só na memória, ou na cache e na memória?
- ♣ ◇ 5. [0,5 valores] Quando é necessário traduzir um endereço virtual para endereço físico, o que é consultado em primeiro lugar: a tabela de páginas, o TLB ou a cache de instruções?
- ◇ 6. [0,5 valores] A comunicação entre processos, em sistemas de multiprocessamento de memória distribuída, é explícita ou pode ser implícita?

Desempenho

- ♣ 7. [2,5 valores] Calcule o tempo que demora a execução de um programa, se na sua execução forem executadas 100 milhões de instruções, com a distribuição e o CPI abaixo, num processador cujo relógio tem uma frequência de 2 GHz.

Classe	A	B	C	D
%	25	30	30	15
CPI	2	5	3	1

- ♣ 8. [2 valores] Um determinado conjunto de instruções é responsável por 40% do tempo de execução de um programa. Se essas instruções forem substituídas por outras, que demorem a executar um sexto do tempo que as instruções originais demoravam, qual será o *speedup* obtido?

(CONTINUA...)

Implementação RISC-V monociclo

- ♣ 9. [4 valores] Pretende-se que a implementação RISC-V monociclo da Figura 1 suporte a execução da instrução `ori` (*OR immediate*), que é uma instrução tipo-I com três argumentos:

	31					0						
ori rd, rs1, immediate	<table><tr><td><i>immediate</i></td><td><i>rs1</i></td><td>6</td><td><i>rd</i></td><td>19</td><td></td></tr></table>						<i>immediate</i>	<i>rs1</i>	6	<i>rd</i>	19	
<i>immediate</i>	<i>rs1</i>	6	<i>rd</i>	19								
bits	12	5	3	5	7							

Esta instrução coloca no registo `rd` o resultado de fazer o OU-lógico bit-a-bit entre o valor contido no registo `rs1` e o valor de `immediate` estendido com sinal. Depois da execução da instrução, é executada a instrução que se lhe segue no programa.

- (a) Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução? (Identifique os *multiplexers* através dos seus sinais de controlo, e.g., `mux(MemtoReg)`.)
- (b) Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar? Se não for necessário fazer qualquer alteração à implementação, explique brevemente qual será funcionamento do processador na execução da instrução. Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.
- (c) Quais os valores que os vários sinais de controlo deverão ter e qual a operação realizada pela ALU durante a execução desta instrução? (Não é necessário apresentar o valor de `ALUOp`.)

Pipeline RISC-V de 5 andares

Pode usar como referência o *pipeline* da Figura 2. Tenha, no entanto, em atenção a caracterização do funcionamento do *pipeline* feita na pergunta.

- ♣ 10. [2,5 valores] Simule a execução do código à direita no *pipeline* de 5 andares, com *forwarding*, com decisão dos saltos condicionais no andar ID, e com previsão perfeita do resultado das instruções de salto condicional, 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. | <code>addi t1, a0, 20</code> |
| 2. | <code>sub t2, t1, t0</code> |
| 3. | <code>sw t2, 0(t1)</code> |
| 4. | <code>lw t1, 28(a0)</code> |
| 5. | <code>andi t1, t1, -16</code> |
| 6. | <code>beq t1, zero, skip</code> |
| 7. | <code>lw t2, 0(t1)</code> |
| 8. | <code>sw t2, 4(t1)</code> |

Cache

11. Considere um sistema com uma cache com dois níveis em que, para um programa, a cache de primeiro nível apresenta uma *miss rate* de 5% e a de segundo nível apresenta uma *miss rate* de 25%.

- ♣ ◇ (a) [2,5 valores] Se o *hit time* da cache de primeiro nível for de 4 ciclos, o *hit time* da cache de segundo nível for de 30 ciclos, e o tempo de acesso à memória física for 300 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%?

Memória virtual

12. Considere um sistema com uma tabela de páginas cujo conteúdo é parcialmente apresentado abaixo e com um TLB 2-way set associative, com 2 conjuntos e com substituição por LRU.

- ♣ ◇ (a) [2,5 valores] Estando o TLB vazio, são feitos acessos às seguintes páginas virtuais, para leitura, pela ordem indicada:

2 4 5 2 6 4

- Para cada acesso, calcule e apresente o índice da posição do TLB em que a tradução para página física fica e o respectivo *tag*, indique se há um *hit* ou um *miss* no TLB, e, quando aplicável, qual a página virtual cuja tradução é substituída.
- Apresente o estado final do TLB, tão completo quanto possível.

- ◇ (b) [2 valores] Sabendo que uma página tem 64 KB, calcule o endereço físico correspondente ao endereço virtual $503E7_{16}$.

Tabela de páginas

Índice	Dirty	Pág. física
0	0	26
1	0	41
2	0	32
3	0	84
4	1	63
5	0	10
6	1	77
7	0	3
8	0	91
...		

Multiprocessamento

13. O código à direita vai ser executado em dois processadores de um sistema de memória partilhada. No início da execução destas instruções, o endereço guardado no registo `a0` é $1000\ 1000_{16}$ nos dois processadores e a palavra nesse endereço tem o valor 500.

```
1. lw  t0, 0(a0)
2. add t0, t0, t0
3. sw  t0, 0(a0)
```

- ♣ ◇ (a) [2 valores] Mostre que, depois da execução destas instruções nos dois processadores, o valor no endereço $1000\ 1000_{16}$ pode não ser 2000.
- ◇ (b) [2 valores] Altere o código apresentado de modo a garantir que, depois de ser executado nos dois processadores, o valor naquela posição de memória será sempre o quádruplo do valor inicial.

Número: _____

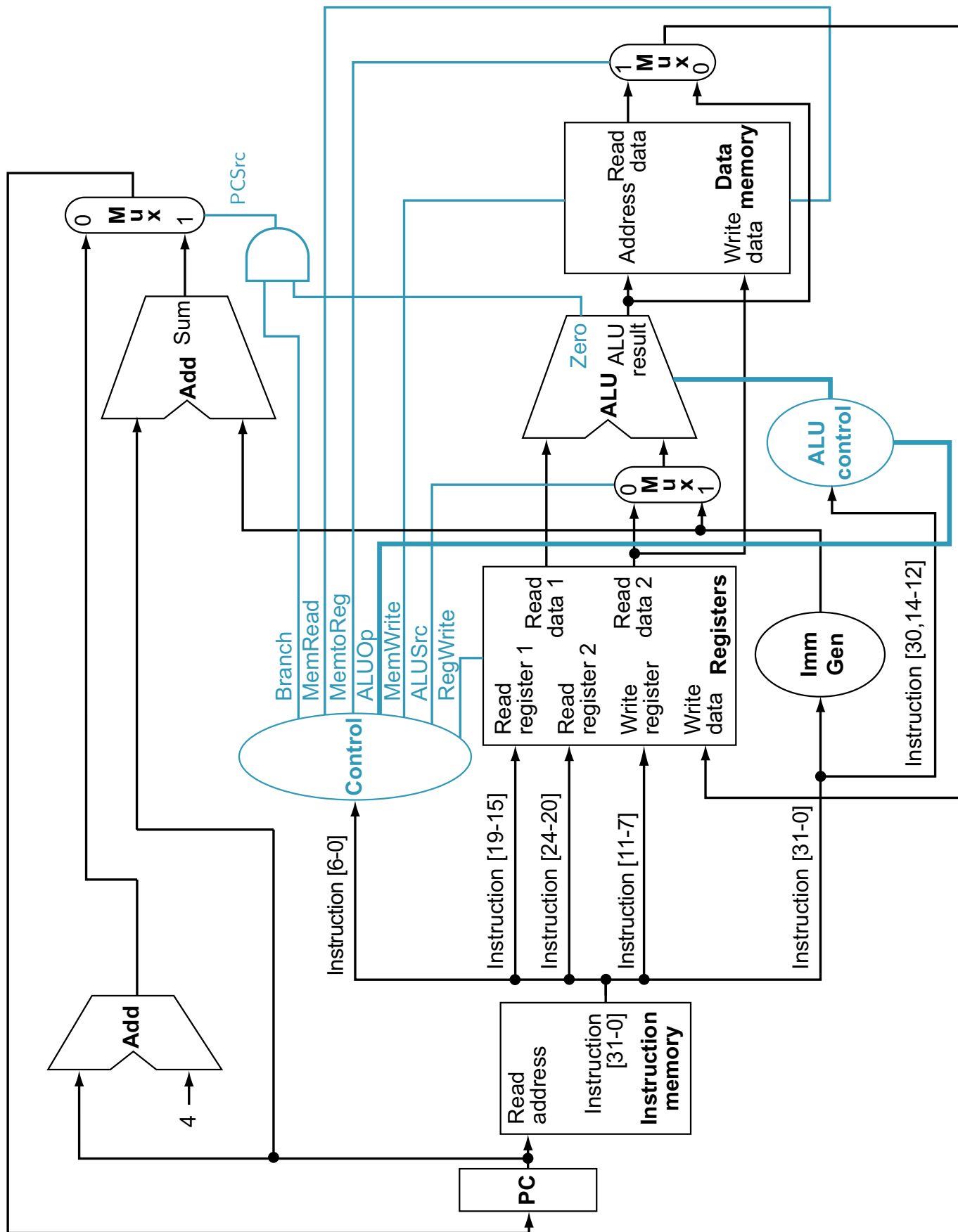


Figura 1: Diagrama de blocos da implementação RISC-V monociclo

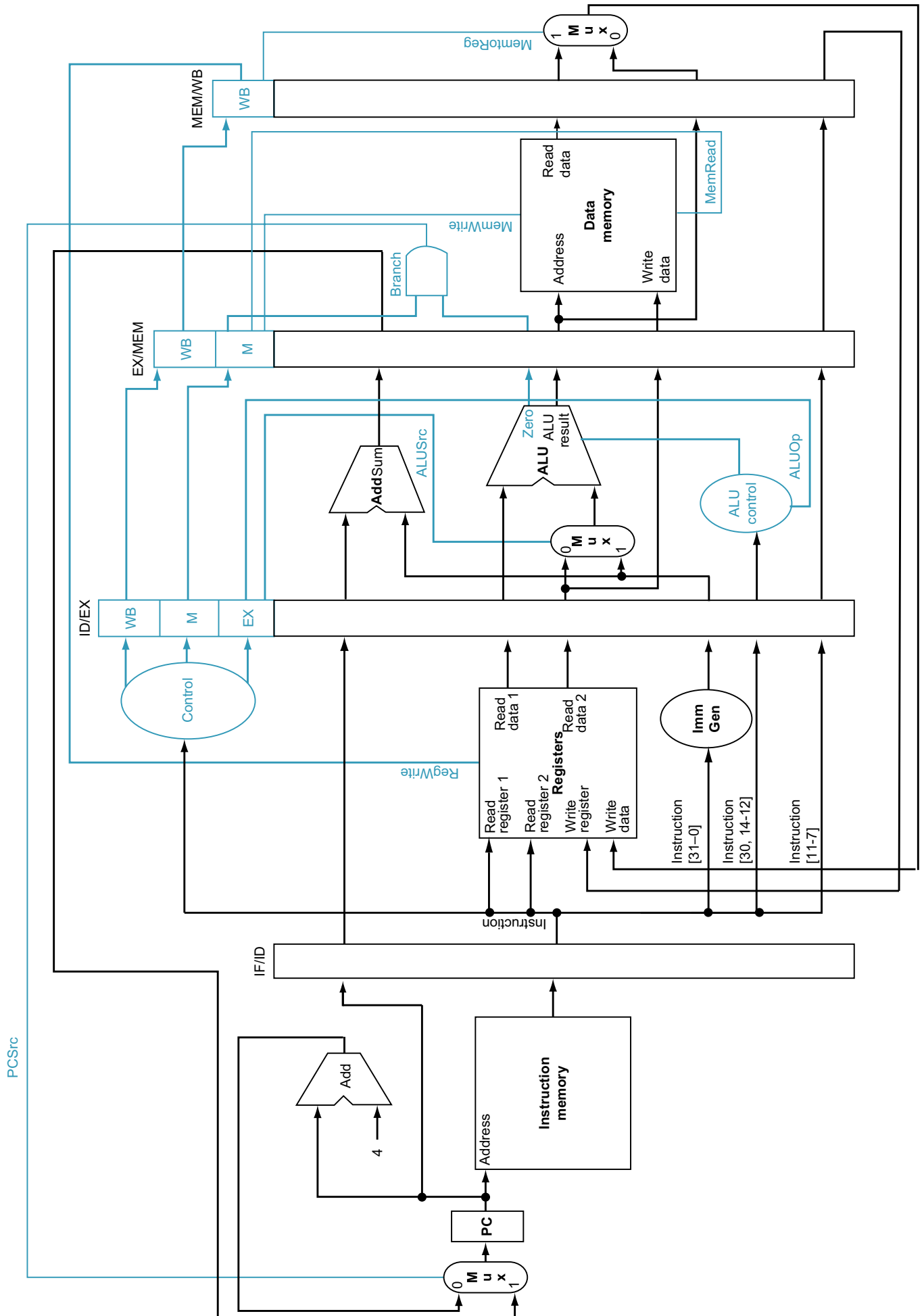


Figura 2: Diagrama de blocos do *pipeline* RISC-V