

Arquitectura de Computadores II

2ª Frequência

Departamento de Informática
Universidade de Évora

16 de Novembro de 2022

Indique todos os cálculos efectuados e todas as fórmulas usadas

Perguntas rápidas

1. [1 valor] Idealmente, qual seria o *speedup* obtido com a execução de um programa num *pipeline* de sete andares em relação à sua execução numa implementação monociclo?
2. [1 valor] Qual é o mecanismo usado para transferir o controlo do processador para o sistema operativo?
3. [1 valor] A execução de um programa num processador com execução fora de ordem pode levar a um resultado diferente do obtido na execução do mesmo programa num processador em que as instruções são executadas estritamente por ordem?
4. [1 valor] A utilização de blocos com mais do que uma palavra, numa cache, destina-se a tirar partido de que tipo de localidade de acessos?

Pipeline RISC-V de 5 andares

Use como referência o *pipeline* da Figura 1. Tenha, no entanto, em atenção, as caracterizações do funcionamento do *pipeline* feitas nas várias perguntas.

5. [1,5 valores] Se as latências dos caminhos críticos de cada um dos 5 andares do processador *pipelined* forem de 350 ps, 500 ps, 290 ps, 380 ps e 190 ps, qual a frequência máxima a que o relógio deste processador pode funcionar?

6. Considere o código RISC-V seguinte:

```
1.  ciclo: lw    x5, 0(x10)
2.          lw    x6, 0(x11)
3.          add   x6, x5, x6
4.          sw    x6, 0(x10)
5.          sw    x5, 0(x11)
6.          addi  x10, x10, -4
7.          addi  x11, x11, -4
8.          bne   x10, x0, ciclo
9.          or     x10, x11, x0
```

- (a) [3 valores] Identifique todas as dependências (de dados) existentes no código, indicando os registos envolvidos, e diga quais as que constituem conflitos.
- (b) [4 valores] Simule a execução do código apresentado, no *pipeline* RISC-V com *forwarding*, com decisão dos saltos condicionais (**beq** e **bne**) no andar MEM e com previsão perfeita do resultado das instruções de salto condicional, assumindo que o salto da linha 8 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.

(CONTINUA...)

7. [2,5 valores] Durante a execução de um programa, no *pipeline* RISC-V de 5 andares, é executado 1 milhão de instruções, das quais 10% são instruções de salto condicional. Se, durante a execução do programa, o processador prevê correctamente o efeito de 80% das instruções de salto condicional, e cada previsão incorrecta custar 4 ciclos de relógio, qual seria o *speedup* obtido se o processador conseguisse prever perfeitamente o efeito das instruções de salto condicional?

ILP

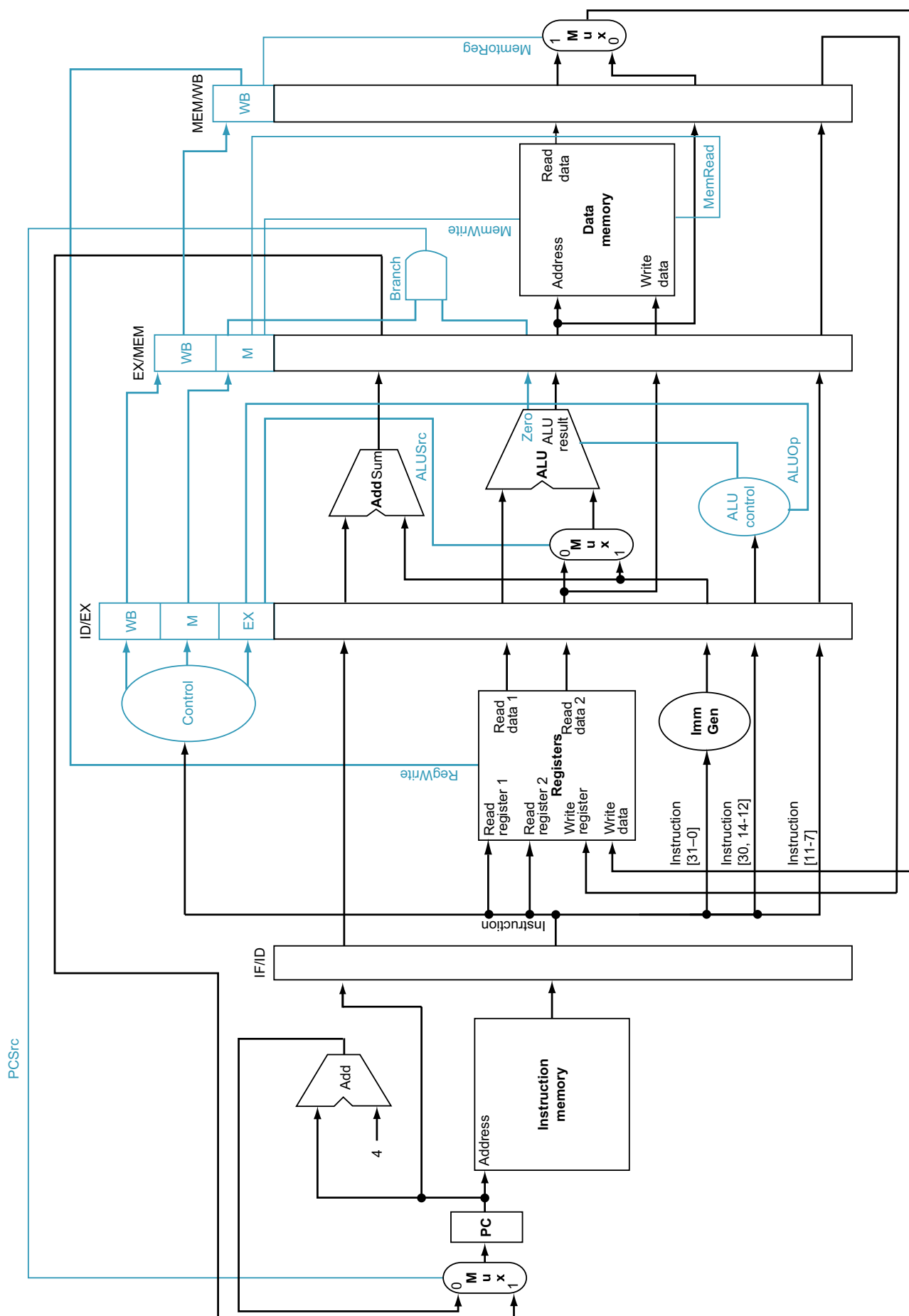
8. [3,5 valores] Organize o código da pergunta 6, introduzindo as alterações que considerar convenientes, para ser executado no *pipeline* RISC-V com *double issue* (com *forwarding*, decisão dos saltos condicionais no andar ID e previsão perfeita), em que cada *issue packet* pode conter uma instrução 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. (Se precisar de algum registo adicional, pode usar os registos **x28** a **x31**.)

Qual o IPC correspondente ao código que escreveu, para cada iteração do ciclo?

9. [2 valores] Considere o código RISC-V abaixo. Para cada uma das instruções, a partir da linha 2, diga, justificando, em que circunstâncias essa instrução poderia ser executada antes da instrução da linha 1, num processador com execução fora de ordem.

```
1.          add  x28, x8, x5
2.          sw   x28, 4(x2)
3.          addi x8, x8, 10
4.          beq  x8, x11, label
           ...
5.  label: or   x6, x11, x12
```

Nome: _____ Número: _____

Figura 1: Diagrama de blocos do *pipeline* RISC-V