

Nota: cada resposta errada nas questões 1 a 6 desconta 0.33 valores

1. [1,0 valores]- Considere o seguinte excerto de um programa escrito em *assembly* e a executar numa máquina com cache:

```
ciclo: movl 0(%ebx), %edx
      movl $10, 0(%ebx)
      addl $4, %ebx
      cmpl $0, %edx
      jnz ciclo
```

Considere que o registo `%ebx` aponta para o início de um array de inteiros (4 bytes) com os seguintes valores: $\{-10, 30, 1024, -33, 0\}$. Note que o ciclo termina quando o valor lido do array for 0. A frequência do relógio é de 2 GHz, o CPI_{CPU} é 2, a *miss rate* de instruções é de 3% e a de dados de 5%. Sabendo que o tempo de execução deste programa é de 150 ns, qual é a *miss penalty* (expressa em tempo)?

- | | |
|--|--|
| <input type="checkbox"/> $mp_T = 150 \text{ ns}$ | <input type="checkbox"/> $mp_T = 50 \text{ ns}$ |
| <input type="checkbox"/> $mp_T = 200 \text{ ns}$ | <input type="checkbox"/> $mp_T = 100 \text{ ns}$ |

2. [1,0 valores]- Complete a afirmação abaixo :

“A técnica de *pipelining*, relativamente a uma arquitectura sequencial de ciclo único, acelera o desempenho de um processador pois ...

- ☐ resulta numa diminuição do CPI, uma vez que mais do que uma instrução se encontra em execução em cada ciclo.”
- ☐ resulta numa diminuição do número de instruções executadas, uma vez que algumas instruções são internamente transformadas em `NOPS`”
- ☐ resulta numa diminuição do período do relógio, uma vez que este deve ser apenas tão longo quanto o estágio mais demorado do *pipeline*.”
- ☐ resulta num aumento da frequência devido a ciclos de *stalling* causados por dependências de dados e/ou controlo.”

3. [1,0 valores]- Complete a afirmação abaixo:

“O programa `for (i=0 ; i<N ; i++) a[i] = b[100*i] * 2; ...`

- ☐ permite explorar a hierarquia de memória pois exhibe localidade espacial nos acessos a `i`.”
- ☐ permite explorar a hierarquia de memória pois exhibe localidade espacial nos acessos a `a[]`.”
- ☐ permite explorar a hierarquia de memória pois exhibe localidade temporal nos acessos a `a[]`.”
- ☐ permite explorar a hierarquia de memória pois exhibe localidade espacial nos acessos a `b[]`.”

4. [1,0 valores] - Quantos *bits* tem a *tag* de uma hierarquia de memória (S=1024, E=8, B=128, m=32)?

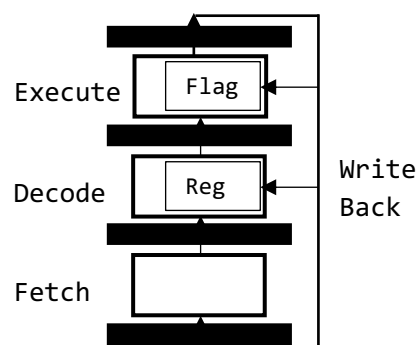
☐ $t = 15$
☐ $t = 17$
☐ $t = 10$
☐ $t = 12$

5. [1,0 valores] - Considere um processador com um bloco de lógica combinatória que pode ser dividido em 4 blocos, cada com uma duração de 214, 283, 252 e 201 picosegundos. Com uma organização em pipeline de 4 estágios este processador permite um ciclo de relógio mínimo de 333 picosegundos. A frequência máxima da organização sequencial correspondente a esta lógica combinatória é de:

☐ $f = 3.0 \text{ GHz}$
☐ $f = 1.0 \text{ GHz}$
☐ $f = 2 \text{ GHz}$
☐ $f = 1.5 \text{ GHz}$

6. [1,0 valores] - Considere o programa abaixo executado numa máquina com *pipeline* com 4 estágios idêntica à apresentada nas aulas (e representada na figura ao lado).

I1: mov \$10, %eax
I2: sub \$5, %eax
I3: jz I1
I4: add \$10, %eax



Se esta máquina resolve todas as dependências (dados e controlo) recorrendo ao *stalling*, então o programa executa em:

☐ $f = 7 \text{ ciclos}$
☐ $f = 11 \text{ ciclos}$
☐ $f = 9 \text{ ciclos}$
☐ $f = 12 \text{ ciclos}$

7. [2.0 valores] A tabela abaixo apresenta na coluna da esquerda uma sequência de endereços ($m=4$) de acesso à memória gerados por um determinado programa. As 3 colunas seguintes referem-se a um modo de mapeamento numa cache que usa o algoritmo de substituição LRU. Preencha-as indicando em que *set*/linha (dentro do *set*) mapeia cada endereço, qual a *tag* associada a essa linha depois deste acesso e indicando se se trata de um *cold miss*, colisão ou de um *hit*. Considere a cache inicialmente fria.

Addr	(S=2,E=2,B=2,m=4)	tag	cold miss/hit/colisão
1			
13			
0			
6			
8			

8. [2.0 valores] Considere de novo a máquina e o programa apresentados na questão 6. Considere agora que esta máquina suporta:
- . previsão estática de saltos, prevendo sempre os saltos condicionais como tomados;
 - . *data forwarding*, com reencaminhamento do registo WR para o estágio de Decode e da saída da ALU para o estágio de Decode - isto é, idêntico ao modelo analisado nas aulas.

Preencha a tabela abaixo, indicando em cada ciclo do relógio qual a instrução em execução em cada estágio e assinalando a injeção de eventuais NOPs com um círculo.

	1	2	3	4	5	6	7	8	9	10	11	12