

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:

```
loop:  movl 0(%ebx), %edx
       mull $2, %edx
       addl %edx, %eax
       addl $4, %ebx
       subl $2, %ecx
       jnz loop
```

Considere que o registo `%ecx` tem inicialmente o valor 10. O programa é executado numa máquina com frequência do relógio igual a 3 GHz, $CPI_{CPU} = 1$, a *miss rate* de instruções é 1% e a de dados é 6%. Sabendo que *miss penalty* é de 150 ns, qual o tempo de execução deste programa?

- | | |
|--|--|
| <input type="checkbox"/> $T_{exec} = 150$ ns | <input type="checkbox"/> $T_{exec} = 100$ ns |
| <input type="checkbox"/> $T_{exec} = 33$ ns | <input type="checkbox"/> $T_{exec} = 13$ 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)?

☐

$t = 15$

☐

$t = 17$

☐

$t = 10$

☐

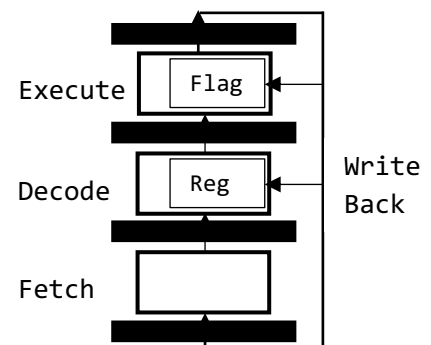
$t = 12$

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:

☐ 7 ciclos☐ 11 ciclos☐ 9 ciclos☐ 12 ciclos

7. [2.0 valores] A tabela abaixo apresenta na coluna da esquerda uma sequência de endereços ($m=6$) 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=4,E=2,B=4,m=6)	tag	cold miss/hit/colisão
1			
17			
3			
6			
32			

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.

Indique, justificando, quantos ciclos serão agora necessários para executar este programa.