

## Parte Teórica

Nome:

Nº Mecanografico:

**Cotações:** 1 a 6 – 0,5 cada; 7 – 1,5; 8a – 1,5 8b – 0,5 8c – 0,5; 9a – 0,5 9b – 1 9c – 1 9d – 1,5 9e – 1

**Note bem:** As respostas às perguntas **1 a 6 e 9** são dadas no enunciado

1. Qual a forma mais realista de comparar o desempenho de processadores:
  - d. A execução de programas característicos de diferentes domínios de aplicação
2. Considere as implementações multi-cycle e pipelined da arquitetura MIPS descritas no curso e suponha que ambas operam com a mesma frequência de relógio. Qual das seguintes afirmações é verdadeira:
  - d. O processador *pipelined* é mais rápido que o processador *multi-cycle*, mas a diferença de desempenho entre os 2 processadores varia consoante os programas executados
3. Na soma de inteiros positivos de n-bits, o que é que indica que o resultado não é representável em n-bits:
  - a. O *carry-out* da posição mais significativa
4. No MIPS qual o formato das instruções de *branch*:
  - a. Tipo I
5. No MIPS os valores em vírgula flutuante precisão dupla:
  - c. Apenas podem estar contidos nos registos de indice par do coprocessador de vírgula flutuante
6. O formato IEEE precisão dupla permite:
  - b. Maior precisão e uma maior gama de representação do que o formato precisão simples
7. Considere duas implementações diferentes, M1 e M2, de uma mesma arquitetura. Existem quatro classes de instruções (A, B, C e D). M1 é uma implementação com um relógio de frequência 100 MHz, e M2 tem um relógio de 150 MHz. A tabela indica o CPI para cada classe de instruções para M1 e M2:

Instruction Class	Frequency	CPI for M1	CPI for M2
A	40%	1	2
B	25%	2	2
C	25%	3	4
D	10%	4	4

Qual a máquina mais rápida? Qual o *speedup* que consegue relativamente à mais lenta?

$$CPI_{M1} = 0,4 \cdot 1 + 0,25 \cdot 2 + 0,25 \cdot 3 + 0,1 \cdot 4 = 2,05$$

$$CPI_{M2} = 0,4 \cdot 2 + 0,25 \cdot 2 + 0,25 \cdot 4 + 0,1 \cdot 4 = 2,7$$

$$t_{M1} = 2,05 \cdot (10^{-8}) \cdot \text{No. de instruções}$$

$$t_{M2} = 2,7 \cdot (10^{-8}/1,5) \cdot \text{No. de instruções} = 1,8 \cdot (10^{-8}) \cdot \text{No. de instruções}$$

$$speedup = t_{M1} / t_{M2} \approx 1,14 \quad \longrightarrow \quad M2 \text{ é 14\% mais rápida do que M1}$$

**Parte Teórica**

8. A e B são dois números representados no formato IEEE de vírgula flutuante, precisão simples:

A = 0 10001100 110000000000000000000000

$B = 0\ 01111111\ 01000000001100000000001$

- a. Qual a representação de  $A * B$  no mesmo formato?

$$\text{Sinal}(A^*B) = \text{Sinal}(A) \text{ XOR } \text{Sinal}(B) = 0$$

Exponentes representados em excesso-127:

$$\text{Expoente (A)} = 10001100 - 01111111 = 13$$

$$\text{Expoente (B)} = 01111111 - 01111111 = 0$$

$$\text{Expoente (A)} + \text{Expoente (B)} + 127 = 10001100$$

$$\text{Mantissa}(A^*B) = \text{Mantissa}(A) * \text{Mantissa}(B) =$$

$$= 11100000000000000000000000000 * 10100000001100000000001$$

Resultado  $\geq 2$  – necessário normalizar: 1.0001100000101010000000011 \* 2

Somar 1 ao expoente: Expoente (A\*B) = 10001101

Arredondar mantissa: 1.0001100000101010000001

$$A * B = 0\ 10001101\ 00011000001010100000001$$

- b. Ao efetuar a multiplicação quais os valores do *guard bit* e do *round bit*?

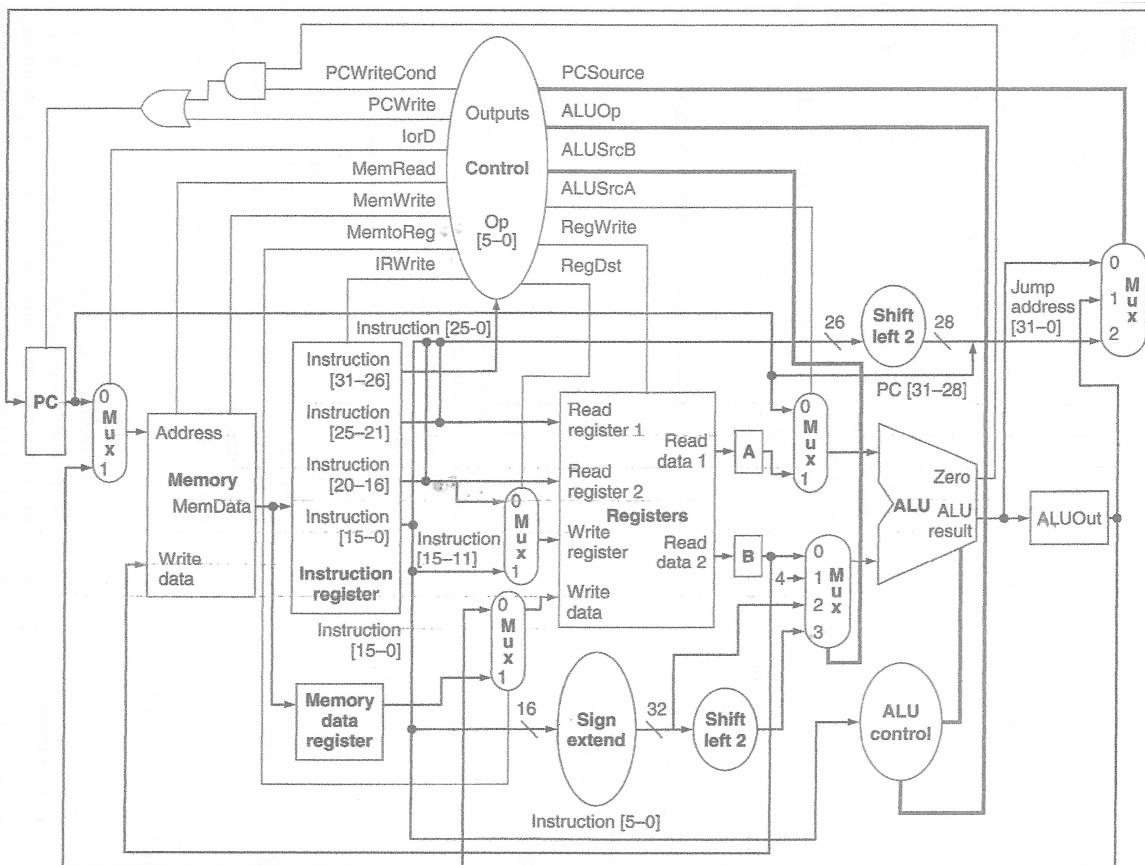
*guard bit = 1*      *round bit = 1*

- c. Qual o valor de A \* B em decimal, sob a forma X\*2<sup>Y</sup>

$$A * B = (1 + 2^{-4} + 2^{-5} + 2^{-11} + 2^{-13} + 2^{-15} + 2^{-23}) * 2^{14}$$

9.

## Parte Teórica



- Que tipo de datapath está representado na figura? **Multicycle**
- Explique a função dos blocos **Instruction Register** e **Memory Data Register**

Nesta implementação do processador a memória é única, contendo instruções e dados, podendo ser acedida mais do que uma vez durante a execução de uma instrução (caso das instruções de *load* e *store*). O processador tem de ter registos que preservem os dados transferidos da memória e permitam identificar o seu tipo (instrução ou dados). Assim o *Instruction Register* guarda o código da instrução que está a ser executada durante os respetivos ciclos de execução e o *Memory Data Register* guarda o dado lido da memória no 4º ciclo da execução das instruções de *load* para que o valor lido seja armazenado no registo especificado no código de instrução no ciclo seguinte.

- Explique a função dos blocos **A**, **B** e **ALUOut**

Os registos temporários **A** e **B** têm como função guardar os valores dos registos lidos no 2º ciclo de execução para que possam ser utilizados em ciclos subsequentes.

O registo temporário **ALUOut** guarda o resultado da operação efetuada durante o 3º ciclo de execução para ser utilizado no ciclo seguinte.

- Atendendo aos valores dos sinais de control indicados na tabela, qual a instrução que está a ser executada?  
**Branch on Equal**
- Em que ciclo de execução se encontra a instrução? Qual o valor dos sinais de control não indicados na tabela?

## Parte Teórica

**No 3º ciclo, o último de execução da instrução de *branch***

PCWriteCond	1	PCSource	01
PCWrite	0	ALUOp	01
IorD	X	ALUSrcB	00
MemRead	0	ALUSrcA	1
MemWrite	0	RegWrite	0
MemtoReg	X	RegDst	X
IRWrite	0		

Nota: A tabela seguinte expressa a lógica de controle da ALU:

ALUOp	Funct field	Desired ALU action
00	XXXXXX	add
01	XXXXXX	subtract
10	100000	add
10	100010	subtract
10	100100	AND
10	100101	OR
10	101010	Set on Less Than