





Número:

Nome:

4. [2.5 valores] Considere o seguinte segmento de código assembly MIPS, em que o registo \$s0 armazena o valor da variável x, e o registo \$s1 da variável y.

Loop:	slt \$t0, \$s1, \$s0	#
	beq \$t0, \$zero, Fim	#
	addi \$s0, \$s0, -1	#
	j Loop	#
Fim:		

- a) No comentário no final de cada linha, indique qual o pseudo-código correspondente a essa instrução assembly. Exemplo do tipo de pseudo-código que deve escrever: **if (x!=\$t0) goto Loop**
- b) Escreva o código C correspondente a este assembly usando o menor número instruções em C possível.


5. [2.5 valores] Complete a implementação em MIPS das seguintes funções: a função somaVector que recebe como argumentos um apontador para o início do array e a respectiva dimensão e soma todos os elementos do array, e a função mediaVector com os mesmo argumentos, e que calcula a média dos elementos (invocando para tal a primeira função). Deve usar exactamente os espaços reservados para cada instrução e a instrução deve equivaler ao respectivo comentário.

somaVector:

```
    add $t0, $zero, $zero    # $t0 acumula a soma dos elementos do array
loop: beq $a1, $zero, exit   # se o número de elementos a percorrer é zero sai
    _____             # obtem próximo elemento inteiro da memória
    _____             # soma o valor do elemento ao acumulador
    _____             # passa para o próximo inteiro no vector
    addi $a1, $a1, -1        # decrementa o número de elementos a percorrer
    j loop
exit: _____            # retorna o valor contido em $t0
    jr $ra
```

mediaVector:

```
_____ # armazena dados que serão necessários mais tarde
_____
_____
jal somaVector
_____ # repõe dados armazenados
_____
_____ # calcula o valor de retorno
_____
jr $ra
```

6. [6 valores] Nesta pergunta pretende-se adicionar ao conjunto de instruções do MIPS uma instrução chamada *addmi* que efectua a adição entre o conteúdo de um registo e um valor imediato, armazenando o resultado no endereço de memória contido no registo *rt* (ao contrário da instrução existente de soma de um registo com um imediato que armazena o resultado num registo).

A especificação em RTL da instrução é a seguinte:

$$\$pc \leftarrow \$pc + 4$$

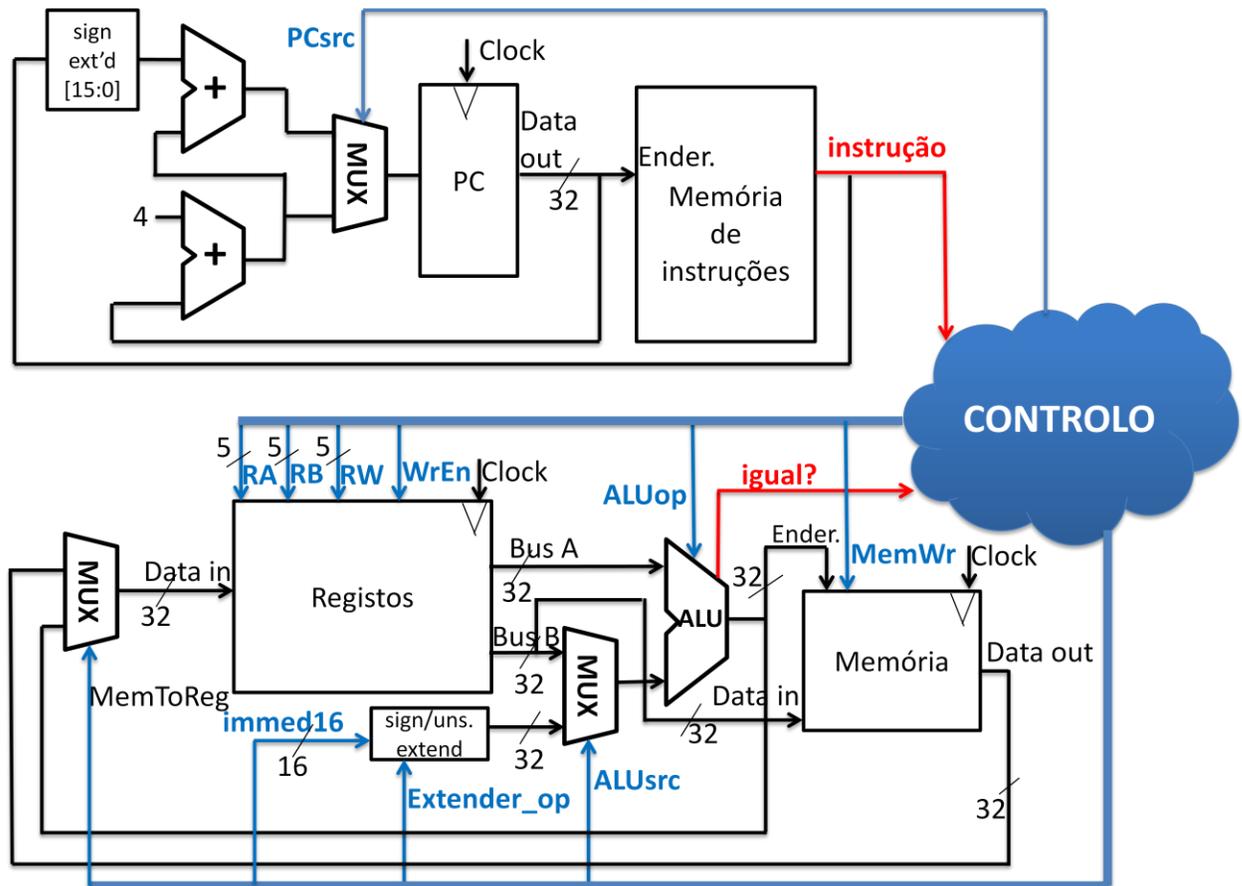
$$MEM[rt] \leftarrow R[rs] + ext\_sinal(imed)$$

- a) Qual o formato (R, I, ou J) da nova instrução em linguagem máquina MIPS? Justifique brevemente.

--	--

- b) Para suportar esta instrução, quais as alterações mínimas que são necessárias efectuar ao hardware do CPU de ciclo único sem pipeline que estudámos na cadeira? Deverá em particular indicar quais os componentes novos que têm de ser adicionados ou modificados. Pode adicionar multiplexers, ALUs, somadores, ou fios. Adicionalmente, pode adicionar novas saídas e entradas aos componentes existentes. A sua resposta deve ter estar dividida em três partes:

- b.1) Efectue as alterações directamente no circuito seguinte:



Número:

Nome:

b.2) Liste os componentes que foram adicionados e modificados, e para cada componente indique claramente a que entradas ou saídas de outros componentes estão ligados:


b.3) Liste as novas linhas de controlo que foram adicionadas:


c) Qual os valores para todas as linhas de controlo do CPU quando esta instrução estiver a ser executada? Deve indicar os valores da respectiva linha (0 ou 1) ou X no caso de ser indiferente (don't care). Deve também adicionar à tabela abaixo as novas linhas de controlo que teve de adicionar ao CPU.

PCSrc	ALUctrl	WrEn	Extender_op	ALUsrc
0=PC+4+(sign_ext (immed)<<2); 1=PC+4	00=ADD; 01=SUB; 10=OR; 11=COMPARE	0 = no write; 1 = write	0 = zero extend, 1 = sign extend	0=RegB; 1=imediato

MemWr	MemToReg				
0 = no write; 1 = write	0 = ALU; 1 = Memory				

d) Suponha que após adicionar esta nova instrução e produzir o respectivo CPU, e ao medir o desempenho do seu programa ao executar no novo CPU, reparou que este se executa mais rapidamente que anteriormente. Indique uma possível explicação para este facto.


e) No entanto, o fabricante teve de diminuir a frequência do CPU para o novo modelo. Indique uma possível explicação para este facto.


f) A resposta da ainea anterior permaneceria válida se o CPU usasse pipelining? Explique justificadamente porquê.


7. [2 valores] Admita uma arquitectura de um computador com palavras e endereços de 32 bits.

a. Qual é o tamanho máximo para o espaço de endereçamento de um programa, ou seja, quanta memória pode um programa endereçar?


b. Dado o seguinte endereço: 0000 0110 1111 0000 1111 1100 1101 1100

Indique as componentes em que é subdividido - tag, índice (quando necessário) e deslocamento - para cada uma das seguintes arquiteturas de cache:

i. completamente associativa de 1 MByte com linhas de 64 Bytes (isto é, a dimensão do bloco é de 64 Bytes).


ii. associativa com 8 vias, com capacidade de 4 MBytes, e com linhas de 32 Bytes (isto é, a dimensão do bloco é de 32 Bytes).


Número:

Nome:

FINAL DO EXAME – Espaço adicional para resolução.

(Cópia do circuito de CPU de ciclo único para o caso de terem de repetir a resolução.)

