

PARTE I (sem consulta)

1. **NOTA:** Use no máximo 40 palavras para responder a cada uma das 6 alíneas seguintes:

- a) Indique o nome dos cinco elementos básicos de um computador
- b) Explique de forma breve o conceito de *stored-program computer*
- c) Descreva de forma sucinta a instrução do MIPS “slt”
- d) Diga o que entende por estratégia *caller saved*
- e) As arquitecturas baseadas no MIPS são do tipo *load/store*. Justifique esta afirmação.
- f) Na arquitectura *pipelined* do MIPS, em que situação podem ocorrer *hazards* de controlo?

2. Considere o seguinte conteúdo de dois registos de uma arquitectura de quatro bits.

RegA = 0111

RegB = 1011

Admita que estes conteúdos representam valores inteiros codificados em complemento para dois. Determine o resultado da multiplicação $\text{RegA} * \text{RegB}$, usando o algoritmo de Booth, admitindo que RegA representa o multiplicando e RegB o multiplicador. Admita ainda que o registo que armazena o resultado é um registo de oito bits. Indique os vários passos para obter o resultado e converta o mesmo para base dez em sinal e módulo.

3. Considere que o conteúdo dos registos \$f20 e \$f22 é respectivamente:

\$f20 = 0 10000010 0101000000000000000000

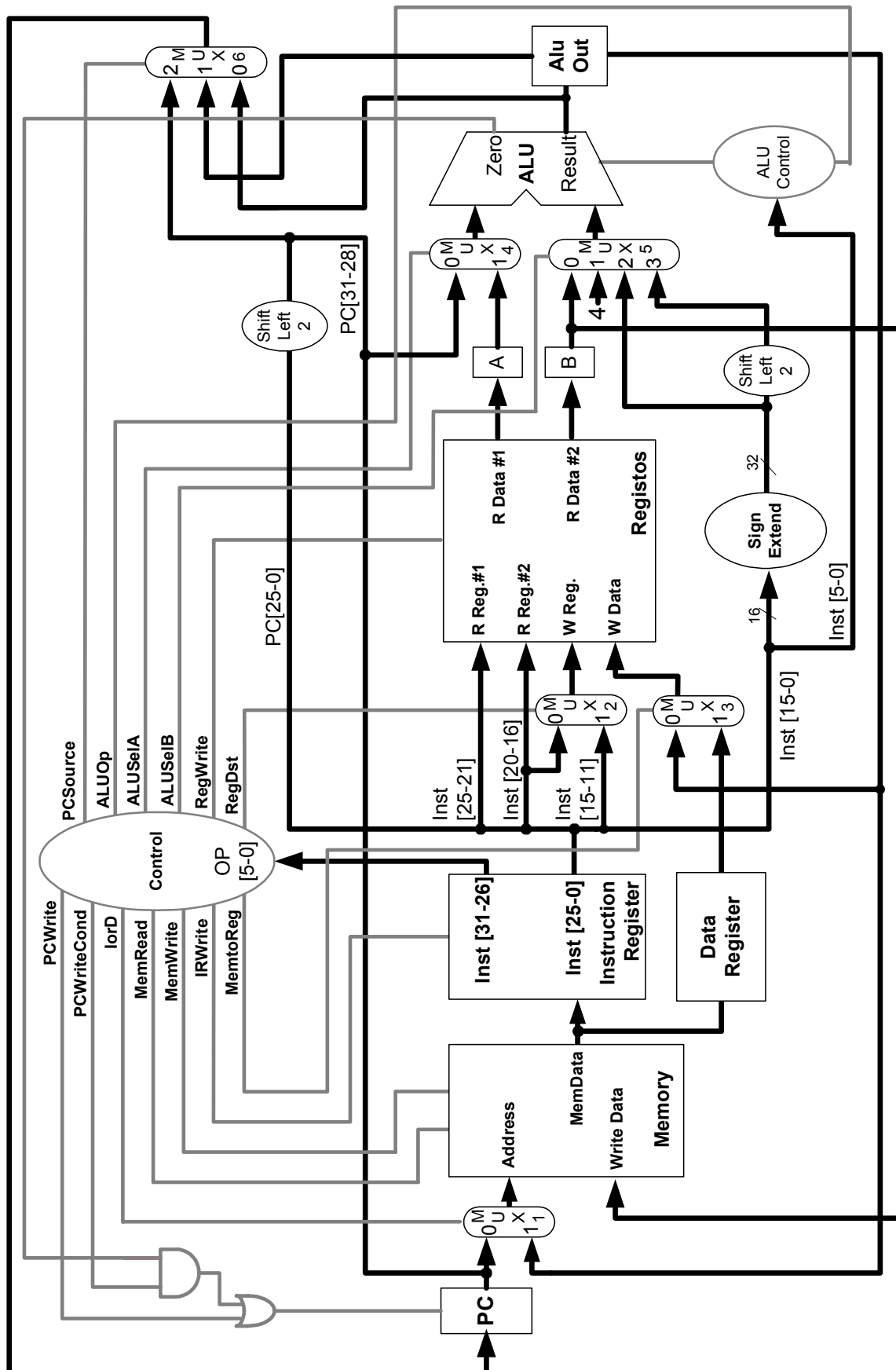
\$f22 = 0 01111110 1100000000000000000000

- a) Obtenha a representação em decimal, vírgula fixa, das quantidades armazenadas naqueles registos no pressuposto de que estão codificadas segunda a norma IEEE 754.
- b) Determine o resultado da instrução **div.s \$f0, \$f20, \$f22**, indicando os vários passos necessários à sua obtenção e fornecendo o valor binário presente no registo **\$f0** após a execução da instrução.

4. Observe com atenção a **figura 2** fornecida em anexo. Considere o conteúdo das posições de memória nos intervalos de endereço **[0x00400058, 0x00400068]** e **[0x10010020, 0x10010028]**, e o conteúdo dos registos do CPU no preciso momento em que se concluiu a execução da instrução armazenada no endereço **0x00400054**. Considere ainda o *datapath* e a unidade de controle de uma versão simplificada do MIPS cujo diagrama é fornecido na próxima página, no pressuposto de que corresponde a uma implementação de execução multi-ciclo sem *pipelining*:

- a) Escreva, em Assembly do MIPS, o trecho de código armazenado no primeiro intervalo de endereços (5 instruções).
- b) Considere a instrução presente no endereço 0x00400064. Preencha a tabela fornecida em anexo com o nome de cada uma das fases de execução da instrução e com o valor que tomam, em cada uma dessas fases, os sinais do *datapath* e os vários sinais de controle ali indicados. Admita que o valor lógico “1” corresponde ao estado activo dos sinais, correspondendo o valor lógico “0” ao seu estado não activo. **NOTA:** Não se esqueça de preencher o cabeçalho (nome, curso e N.M).
- c) Determine, justificando, o valor armazenado no segundo intervalo de memória, após a execução do trecho de código indicado, e no momento em que o conteúdo do **PC** é **0x0040006C**.
- d) Identifique qual das instruções do trecho de código indicado não é suportada pelo *datapath* fornecido. Sugira as alterações necessárias para que o pudesse ser e determine o número mínimo de ciclos de relógio necessários à sua execução.

Cotações: 1a) a 1f) – 0,5; 2 – 2,0; 3a) – 1,0; 3b) – 1,5; 4a) – 1,0; 4b) – 2,0; 4c) – 0,5; 4d) – 1,0

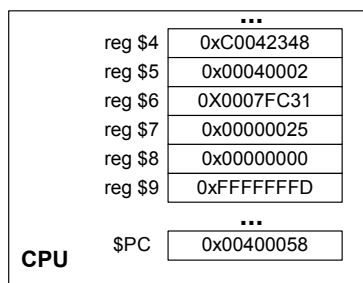


Nome: _____

Curso: _____ N° Mecanográfico: _____

OpCode Funct Operação

0	0x20	add
0	0x22	sub
0x02		j
0x03		jal
0x04		beq
0x05		bne
0x08		addi
0x0F		lui
0x23		lw
0x2b		sw



0x00400058
0x0040005C
0x00400060
0x00400064
0x00400068

0x10010020
0x10010024
0x10010028

Memória

001111 00000 00001 00010000000000001
100011 00001 01000 0000000000101000
000000 01001 00111 00110 00000 100000
101011 00001 00110 0000000000100100
000100 00111 01000 111111111111010
...
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0011 0111 0100 1000 0010 0010
0000 0000 0000 0000 0000 1010 1111 0010
...

Figura 2 (Problema 4)

Fase 1	Fase 2	Fase 3	Fase 4	Fase 5
--------	--------	--------	--------	--------

Nome da fase					
--------------	--	--	--	--	--

Datapath					
A					
B					
Data Register					
ALU Result					
ALU Out					
ALU Zero					

Controlo					
PCWrite					
PCSource					
PCWriteCond					
IorD					
MemRead					
MemWrite					
IRWrite					
ALUOp					
ALUSelA					
ALUSelB					
RegWrite					
MemtoReg					
RegDst					