

Parte Teórica

Nome:

No. Mecanográfico:

Cotações:

1 – 1 2a – 1 2b – 1 2c – 0,5 3a – 1,5 3b – 0,5 4a – 0,5 4b – 1,5 4c – 2 5a – 0,5 5b – 2

Nota: As respostas às questões 1 e 4b devem ser dadas neste enunciado

1. Preencha a tabela com a representação em 8 bits dos valores indicados nos diferentes formatos de representação. Se o valor não for representável num dado formato coloque uma cruz na respetiva célula.

Decimal	Sinal e módulo	1's complement	2's complement	Excesso-127
+35	00100011	00100011	00100011	10100010
-35	10100011	11011100	11011101	01011100
+255	X	X	X	X
-130	X	X	X	X

2. Um programa em *assembly* utiliza os registos \$s0 e \$s1 para armazenar os valores das variáveis a e b e tem valores intermédios de cálculos nos registos \$t0 e \$t1, quando invoca uma subrotina. Esta subrotina não invoca qualquer outra e utiliza os registos \$t0, \$t1, \$t2, \$s0, \$s1 e \$s2. Represente o estado do *stack*:

a		b		c	
Antes de jal		Durante exec. Subrot.		Antes de jr \$ra	
0x00000000					
\$sp					
0xFFFFFFFF					

Parte Teórica

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

A = 0 01111001 011100000011000000000001

B = 1 10001000 100100000000000000000000

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

```

                                1011100000011000000000001
                                1100100000000000000000000
      1011100000011000000000001
    1011100000011000000000001
  1011100000011000000000001
100011111101001011000000110010000000000000000000000

```

Normalizar resultado (shift right uma posição e incrementar expoente):

$\text{Exp}(A*B) = \text{Exp}(A) + \text{Exp}(B) - 127 + 1 = 01111001 + 10001000 - 01111111 + 1 = 10000011$

$\text{Significand}(A*B) = 10001111110100101100000011001$

Guard bit = 1, Round Bit = 1, Sticky bit = 1

$\text{Significand}(A*B) = 100011111101001011000001$

A*B = 1 10000011 00011111101001011000001

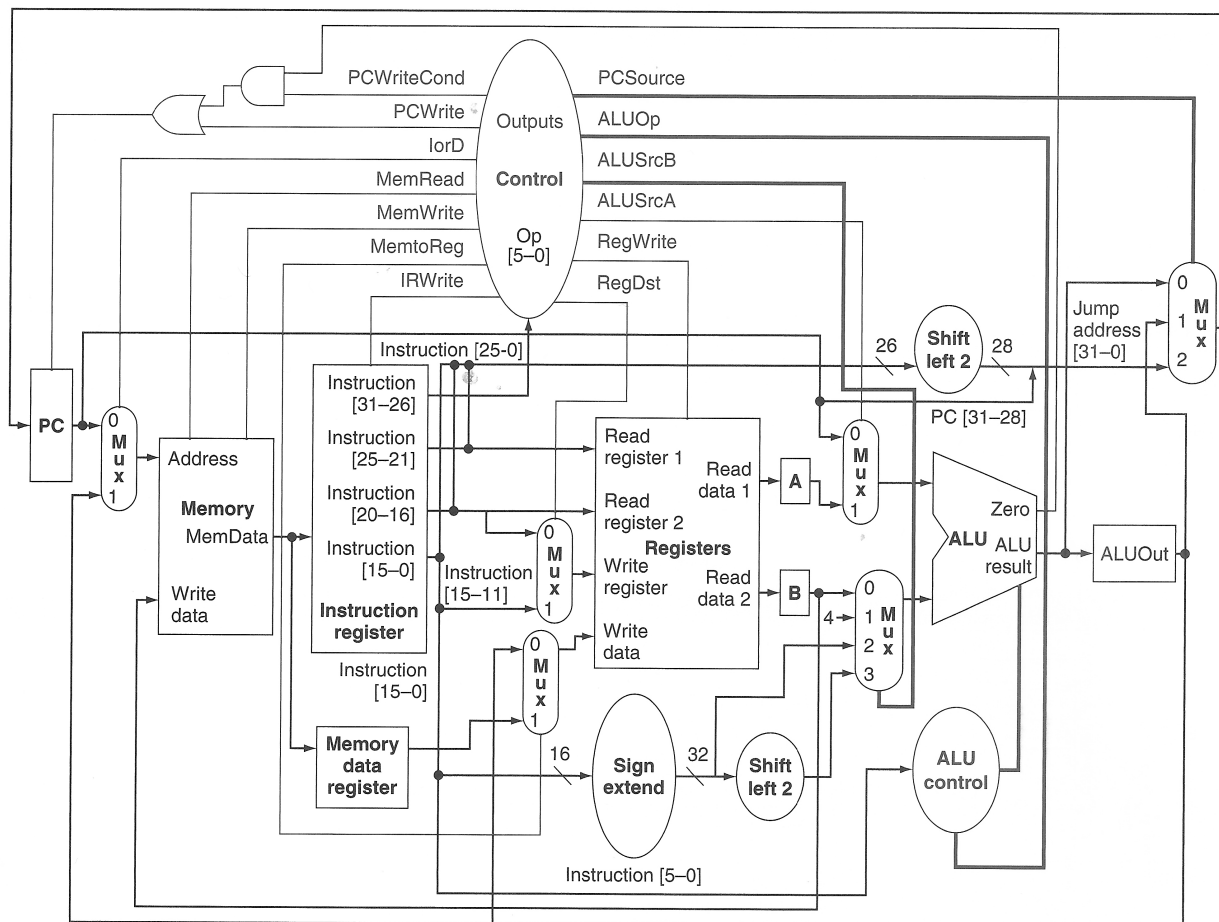
- c. Qual o valor de A*B em decimal, sob a forma $X*2^Y$

$Y = +4$

$X = -(1 + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-11} + 2^{-14} + 2^{-16} + 2^{-17} + 2^{-23}) = -(1 + 2^{-3} - 2^{-9} + 2^{-11} + 2^{-14} + 2^{-16} + 2^{-17} + 2^{-23})$

4. A figura representa uma implementação do datapath do MIPS, incluindo a indicação da unidade de controle:

Parte Teórica



- a. Que tipo de datapath está representado na figura? **Multi-cycle datapath**
- b. Preencha a tabela abaixo com o valor dos sinais de controle na execução do 3º ciclo da instrução **beq \$t0, \$t1, target1**.

PCWriteCond	1
PCWrite	0
IorD	X
MemRead	0
MemWrite	0
MemtoReg	X
IRWrite	0

PCSource	01
ALUOp	01
ALUSrcB	00
ALUSrcA	1
RegWrite	0
RegDst	X

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

Parte Teórica

10	100010	subtract
10	100100	AND
10	100101	OR
10	101010	Set on Less Than

c. O código da instrução **beq \$t0, \$t1, target1** é:

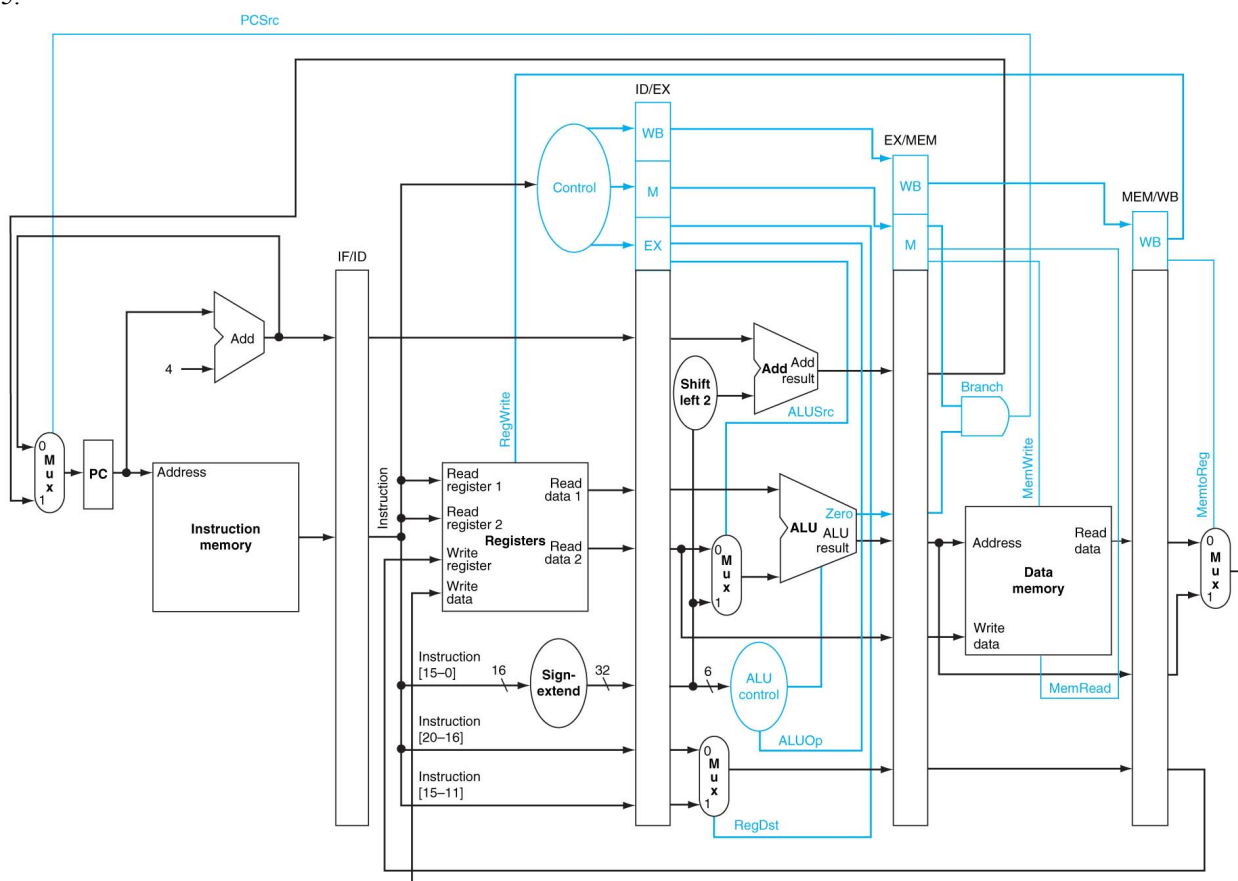
000100	01000	01001	1111111111111001
--------	-------	-------	------------------

O endereço da instrução é 0x00400140. Qual o endereço da instrução que é executada a seguir quando (\$t0) = (\$t1)?

$$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 0100_2 = 0xFFFFFE4$$

$$0x00400144 + 0xFFFFFE4 = \mathbf{0x00400128}$$

5.



a. Que tipo de datapath está representado na figura? **Pipeline de 5 andares**

b. Qual o conteúdo do registo EX/MEM quando as instruções no pipeline são:

```

sub      $2, $4, $3          # WB stage
beq      $12, $8, Lab1
add      $14, $4, $6
sw       $15, 100($7)
Lab1:    or      $13, $6, $2

```

Parte Teórica

MEM stage: beq \$12, \$8, Lab1

EX/MEM_WB: RegWrite = 0; MemtoReg = X

EX/MEM_M: Branch = 1; MemRead = 0; MemWrite = 0

EX/MEM_AddResult: endereço de or \$13,\$6,\$2

EX/MEM_Zero: 1 se (\$12) = (\$8) senão 0

EX/MEM_ALU Result: (\$12) – (\$8)

EX/MEM_Read Data 2: (\$8)

EX/MEM_Instruction[]: X