

Parte Teórica

Cotações:

1 – 0,5 2 – 0,5 3a – 1 3b – 1 4a – 1,5 4b – 0,5 4c – 1 4d – 0,5 5a – 0,5 5b – 2 6a – 2 6b – 2

1. Uma *pseudo instrução* é:

b. **Uma mnemónica que é traduzida pelo *assembler* em instruções nativas**

2. A representação em complemento para 2 permite representar em n bits a gama dos números inteiros:

d. $-2^{(n-1)}$ a $+2^{(n-1)} - 1$

3. Em *assembly* os registos \$8 a \$15, \$24 e \$25 são designados \$t0 a \$t9 e os registos \$16-\$23 \$s0 a \$s7.

a. Essas designações correspondem a uma diferença, a nível da arquitetura, das funcionalidades desses subconjuntos dos registos, ou a uma convenção usada pelos compiladores e que deve ser seguida pelos programas *assembly*?

Correspondem a uma convenção usada pelos compiladores e que deve ser seguida pelos programas *assembly*

b. Quando um programa invoca um procedimento quais as regras seguidas na utilização do conjunto dos registos \$s e na do conjunto dos registos \$t?

registos \$s – o procedimento invocado (“*callee*”) tem de salvar em memória (no stack) o conteúdo dos registos \$s que pretenda alterar durante a sua execução, restaurando o seu conteúdo antes de retornar ao programa que o invocou.

registos \$t – o programa que invoca (“*caller*”) tem de salvar em memória (no stack) o conteúdo dos registos \$t de que necessita após a execução do procedimento.

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

A = 0 10001100 110000000000000000000000

B = 0 01111111 010000000011000000000001

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

Exp(A) – Exp(B) = 1101 (13_{10})

11100000000000000000000000000000

- 101000000011000000000001

11011111111101011111111111

Significand(A-B) = 1011111111101011111111

(A-B) = 0 10001100 1011111111101011111111

b. Ao efetuar a subtração quais os valores do *guard bit*, *round bit* e *sticky bit*?

***guard bit* = 0 *round bit* = 1 *sticky bit* = 1**

c. Se convertesse o resultado para precisão dupla qual a representação do campo do expoente?

10001100 – 01111111 = 00001101

00000001101 + 01111111111 = **10000001100**

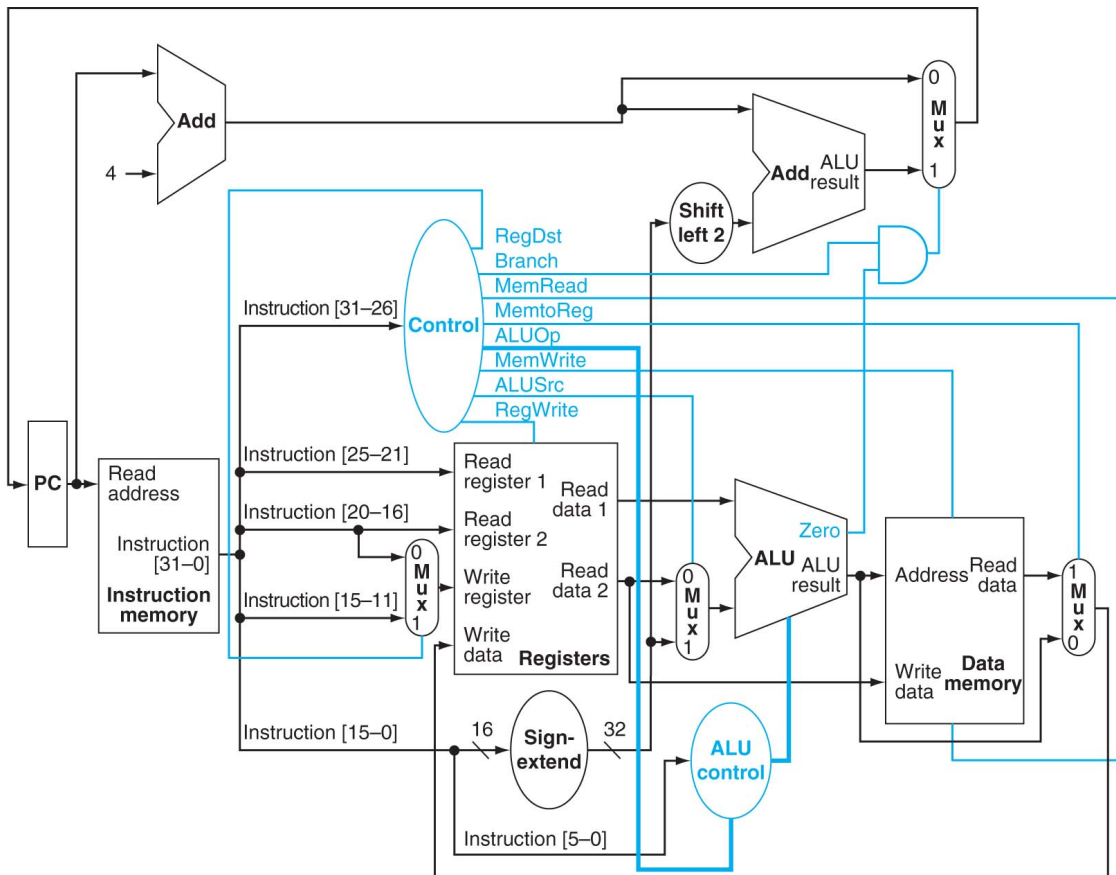
d. Qual o valor de A - B em decimal, sob a forma $X \cdot 2^Y$

$(1 + 2^{-1} + \sum_{i=-3}^{-12} 2^{-i} + 2^{-14} + \sum_{i=-16}^{-23} 2^{-i}) \cdot 2^{13} = (1 + 2^{-1} + 2^{-2} - 2^{-12} + 2^{-14} + 2^{-15} - 2^{-23}) \cdot 2^{13}$

Parte Teórica

$$= (2 - 2^{-2} - 2^{-12} + 2^{-14} + 2^{-15} - 2^{-23}) * 2^{13}$$

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



- a. Que tipo de datapath está representado na figura? **Single-cycle datapath**
 b. Preencha a tabela abaixo com o valor dos sinais de controle na execução da instrução `lw $14, 100($2)`

RegDst	0
Branch	0
MemRead	1
MemtoReg	1

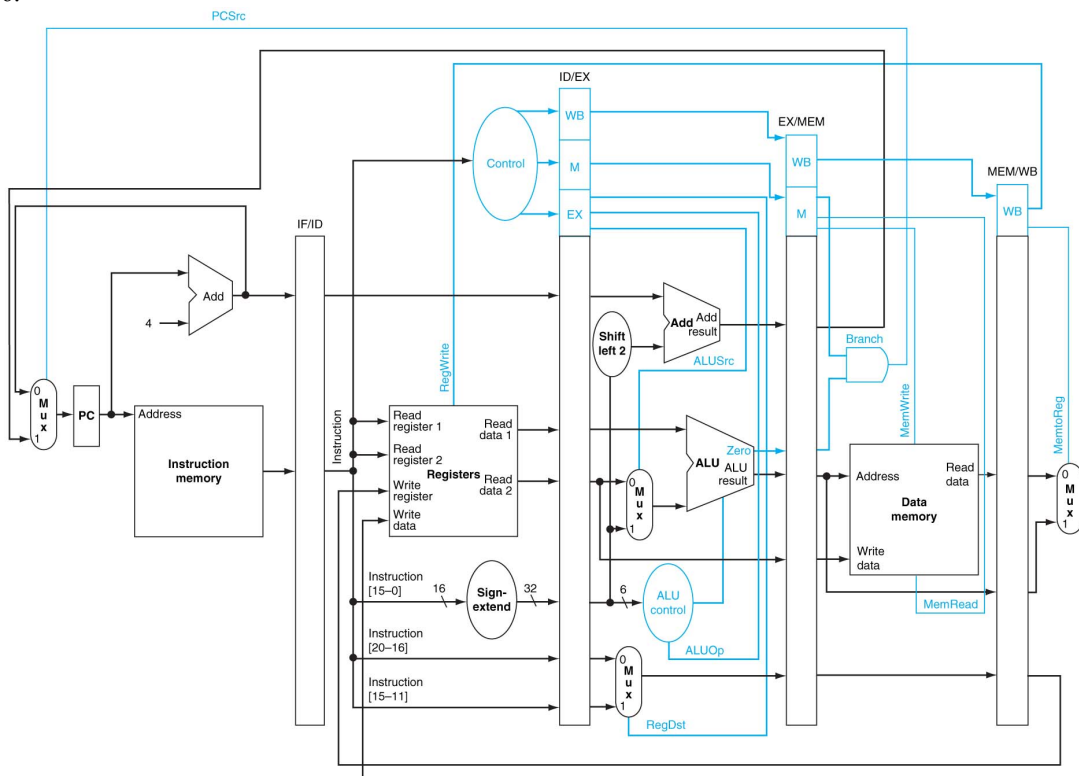
MemWrite	0
ALUOp	00
ALUSrc	1
RegWrite	1

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

Parte Teórica

6.



- a. Indique os diversos campos, e o respetivo número de bits, do registo ID/EX

ID/EX_PC: 32 bits

ID/EX_Read data 1: 32 bits

ID/EX_Read data 2: 32 bits

ID/EX_ImmediateExtended: 32 bits (inclui Instruction[5-0])

ID/EX_Instruction[20-16]: 5 bits

ID/EX_Instruction[15-11]: 5 bits

ID/EX_WB: 2 bits (RegWrite, MemtoReg)

ID/EX_M: 3 bits (Branch, MemRead, MemWrite)

ID/EX_EX: 4 bits (ALUOp, ALUSrc, RegDst)

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

```
sub    $2, $4, $3      # WB stage
and    $12, $8, $5
add    $14, $4, $6
sw     $15, 100($7)
or     $13, $6, $2
```

ID/EX_PC = endereço de sw

ID/EX_Read data 1 = conteúdo de \$r4

ID/EX_Read data 2 = conteúdo de \$r6

ID/EX_ImmediateExtended = add_Instruction[15-0] sign-extended
= 0000000000000000 00110 00000 10000

ID/EX_Instruction[20-16] = 6₁₀

ID/EX_Instruction[15-11] = 14₁₀

ID/EX_WB: 1, 1

ID/EX_M: 0, 0, 0

ID/EX_EX: 10, 0, 1