

ORGANIZAÇÃO DE COMPUTADORES

**TRABALHO 7**

Cássio Araujo

Leonardo Henrique Steil

Santa Maria, 2016

**1.** Escreva os seguintes números em binário (se possível), com 8 bits, usando a representação sem sinal, com sinal na representação sinal e magnitude e complemento de 2:

(a) +10

(b) -10

(c) +99

(d) -99

(e) +128

(f) -128

Representação sem sinal:

1. + 10 = 0000 1010
2. - 10 = Não possível representar sem sinal
3. + 99 = 0110 0011
4. - 99 = Não é possível representar sem sinal
5. + 128 = 1000 0000
6. - 128 = Não é possível representar sem sinal

Representação sinal e magnitude:

1. + 10 = 0000 1010
2. - 10 = 1000 1010
3. + 99 = 0110 0011
4. - 99 = 1110 0011
5. +128= Não é possível representar em sinal e magnitude
6. -128 = Não é possível representar em sinal e magnitude

Representação complemento de 2

1. + 10 = 0000 1010
2. - 10 = 1111 0110
3. + 99 = 0110 0011
4. - 99 = 1001 1101
5. +128= Não é possível representar em complemento de 2
6. -128 = 1000 0000

**2**. Resolva as seguintes operações (considere os números sem sinal).

(a) 01010101 + 00001110 = 0110 0011

(b) 11010001 + 11101000 = 0001 1011 1001 (se for possível aumentar os bits), caso contrário teremos overflow

(c) 01010101 – 00001110 = 0100 0111

(d) 11010001 – 11101000

a)

0 1 0 1 0 1 0 1 (85)

+0 0 0 0 1 1 1 0 (14)

0 1 1 0 0 0 1 1 (99)

b) 1 1

1 1 0 1 0 0 0 1 (209)

+ 1 1 1 0 1 0 0 0 (232)

11 0 1 1 1 0 0 1 (441)

**OBS: Analisando, verificamos que o resultado somente é possível se considerarmos a extensão de 8 para 12 bits, caso contrário teríamos como resultado (185), se fosse desprezado o 1 “extra”, o que não seria a resposta correta.**

c) 1 1 1

0 1 0 1 0 1 0 1 (85)

- 0 0 0 0 1 1 1 0 (14)

0 1 0 0 0 1 1 1 (71)

**OBS: Quando não pode subtrair e tem 0 o bit da esquerda empresta 1 para ser feita a subtração.**

d)

1 1 0 1 0 0 0 1 (209)

- 1 1 1 0 1 0 0 0 (232)

1 1 0 1 0 0 1

**OBS: Nesse caso, verificamos que quando chega na subtração final, a mesma não pode ser efetuada pois temos 0 -1 e o 0 não tem de onde pegar emprestado. Logo não é possível fazer a subtração. Além disso, não podemos representar números negativos, nesse caso seria (-23) na representação sem sinal.**

**3**. Resolva as seguintes operações (considere os números em complemento de 2).

(a) 01010101 + 00001110

(b) 11010001 + 11101000

(c) 01010101 – 00001110

(d) 11010001 – 11101000

a) 1 1 1

0 1 0 1 0 1 0 1 (85)

+ 0 0 0 0 1 1 1 0 (14)

0 1 1 0 0 0 1 1 (99)

b) 1 1

1 1 0 1 0 0 0 1 (-47)

+ 1 1 1 0 1 0 0 0 (-24)

1 1 0 1 1 1 0 0 1 (-71)

**OBS: O bit mais significativo do resultado, no caso o 1**, **é desprezado.**

c) 1 1 1 1

0 1 0 1 0 1 0 1 (85) 0 1 0 1 0 1 0 1 (85)

- 0 0 0 0 1 1 1 0 (14) -> 1 1 1 1 0 0 0 1 + 1 -> + 1 1 1 1 0 0 1 0 (-14)

1 0 1 0 0 0 1 1 1 (71)

**OBS: O bit mais significativo do resultado, no caso o 1, é desprezado. Além disso, foi feito o complemento do 14 para -14 e efetuada a soma dos valores.**

d) 1

1 1 0 1 0 0 0 1 (-47) 1 1 0 1 0 0 0 1 (-47)

- 1 1 1 0 1 0 0 0 (-24) -> 0 0 0 1 0 1 1 1 + 1 -> + 0 0 0 1 1 0 0 0 (24)

1 1 1 0 1 0 0 1 (-23)

**OBS: Foi efetuado o complemento de -24 para mais 24, dessa forma foi efetuado a soma dos valores encontrando o resultado -23.**

**4**. Qual o maior e o menor valor que pode ser armazenado em um tipo int, em um programa escrito na linguagem C? Justifique a sua resposta.

Se considerarmos números com sinal, o número maior possível seria = 2147483647. E o menor seria - = −2147483648. Se considerarmos números sem sinal, temos como menor valor o 0 e como maior valor o número = 4294967295.

**5.** Escreva em linguagem de montagem para o MIPS, um programa para realizar a soma de dois números de 64 bits. Teste o seu programa com a seguinte soma: 0x1234567812345678 + 0x8012345612345678. Comente os resultados se consideramos os números (a) sem sinal e (b) em complemento de 2.

Verificamos a alteração apenas na parte mais “alta” do resultado, tendo em vista que o número hexadecimal 0x80123456 é negativo se considerarmos complemento de 2 pois o 8 em binário é 1000. Fazendo com que o resultado final fique negativo.

Resultados encontrados:

1. Sem sinal = 2454096590610839792
2. Complemento de dois = -1840870706610839792

**OBS: Código em anexo soma 64.asm**