

## Exercícios sobre aritmética

### I – Aritmética inteira

1. Represente em 2's complement com 16 bits os operandos e execute as operações a seguir indicadas:
  - (a)  $3 + 12$
  - (b)  $13 - 2$
  - (c)  $5 - 6$
  - (d)  $-7 - (-7)$
  
2. Considere a adição e a subtração de números de 4-bit em 2's complement. Como os operandos podem ser negativos ou positivos e o operador pode ser subtração ou adição, há 8 combinações possíveis dos inputs. Por exemplo, um número positivo pode ser somado a um número negativo, ou um número negativo pode ser subtraído de outro número negativo, etc. Para cada uma das combinações possíveis, descreva como pode ser calculado o overflow a partir do sinal dos operandos e do carry out do bit mais significativo. Preencha a tabela seguinte:

Sign (Input 1)	Sign (Input 2)	Operation	Sign (Output)	Overflow? (Y/N)
+	+	+	+	
+	+	+	-	
+	+	-	+	
+	+	-	-	
+	-	+	+	
+	-	+	-	
+	-	-	+	
+	-	-	-	
-	+	+	+	
-	+	+	-	
-	+	-	+	
-	+	-	-	
-	-	+	+	
-	-	+	-	
-	-	-	+	
-	-	-	-	

3. Dispõe-se de um multiplicador série, análogo ao apresentado nas aulas, para inteiros *unsigned* representados em 4-bits. Pretende-se utilizá-lo para multiplicar 1011 por 1010.
  - a) Preencha a tabela indicando o valor dos registos a cada passo da execução do algoritmo de multiplicação e fazendo a respetiva descrição (shift left, shift right, add / no add).
  - b) Que alterações teria de introduzir no desenho do multiplicador para multiplicar valores em complemento para 2 utilizando o algoritmo de Booth? Assuma o mesmo conjunto de bits para o multiplicando e o multiplicador, mas representando agora valores em complemento para 2, e preencha de novo a tabela com a indicação dos vários passos do algoritmo de Booth.
  - c) Que modificações teria de introduzir no algoritmo de Booth e no esquema do multiplicador para poder utilizá-lo para efetuar multiplicações tanto *signed* como *unsigned*? Preencha de novo a tabela com a descrição dos passos da execução da multiplicação no seu multiplicador de Booth dos mesmos operandos *unsigned*.

Produto	Multiplicando	Multiplicador	Descrição	Step
0000 0000	0000 1011	1010	Valores Iniciais	Step 0
				Step 1
				Step 2
				Step 3
				Step 4
				Step 5
				Step 6
				Step 7
				Step 8
				Step 9
				Step 10
				Step 11
				Step 12
				Step 13
				Step 14
				Step 15

4. Pretende-se efetuar a divisão de inteiros de 4 bits usando um divisor idêntico ao apresentado nas aulas e no livro. Preencha a tabela indicando o valor dos registos a cada passo da execução do algoritmo de divisão e fazendo a respetiva descrição (shift left, shift right, sub). O valor do Divisor é 4 (0100, com 0000 bits à direita para o right shift), o Dividendo é 6 (inicialmente colocado no registo *Remainder*).

Quociente	Divisor	Remainder	Descrição	Step
0000	0100 0000	0000 0110	Initial Values	Step 0
				Step 1
				Step 2
				Step 3
				Step 4
				Step 5
				Step 6
				Step 7
				Step 8
				Step 9
				Step 10
				Step 11
				Step 12
				Step 13
				Step 14
				Step 15

## II – Vírgula Flutuante (representação standard IEEE)

1. Qual o valor em decimal do número representado em precisão simples como

1 01111101 0010000000000000000000

2. Neste problema usa-se um formato de representação em 8-bits IEEE 754-like, normalizado com 1 *sign bit*, 4 *exponent bits*, e 3 *mantissa bits*. O expoente é codificado em *excesso-7*. A ordem dos campos no número é (sign, exponent, mantissa). É usado *unbiased rounding to the nearest even* como especificado no standard IEEE.

a) Codifique os seguintes números neste formato:

(a)  $0.0011011_2$

(b)  $16.0_{10}$

b) Some  $1.011 + 0.0011011$  indicando o valor dos bits de guarda, de arredondamento e do *sticky bit*

c) Qual o valor em decimal de 1 1010 101

4. Represente no formato IEEE precisão simples o valor  $-11/16$  (-0.6875)

5. Qual é o menor número positivo ( $\neq 0$ ) normalizado representável no formato IEEE precisão simples? E o menor não normalizado (*denormalized*)?