

## Aula 4 - Aritmética Binária

### Aritmética em sinal e magnitude

Deve ser executado um algoritmo que observe o sinal dos números e decida a operação, além do sinal do resultado.

#### Algoritmo para soma:

- Se ambos os números têm o mesmo sinal, somam-se as magnitudes. O resultado tem o mesmo sinal das parcelas.

A soma de dois números binários utiliza as mesmas regras da soma no sistema decimal, mas existem somente dois símbolos: 0 e 1. Tem-se que:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$  e vai um

A explicação para a soma de dois dígitos 1 dar 0 e vai um é que  $1 + 1 = 10 = 2_{10}$ .

Da mesma forma, se tivermos  $1 + 1 + 1 = 1$  e vai um ( $11 = 3_{10}$ )

- Se os números têm sinais diferentes:
  - a) Identifica-se a maior das magnitudes e registra-se o seu sinal
  - b) Subtrai-se a magnitude menor da maior (sem o bit de sinal)
  - c) O sinal do resultado é igual ao sinal da maior magnitude.

A subtração em base 2 é feita de maneira semelhante à realizada no sistema decimal (minuendo – subtraendo = diferença).

Mas, como se dispõe de somente dois algarismos, 0 e 1, quando se tem 0 menos 1 necessita-se de “empréstimo” de um valor igual à base (neste caso, 2), obtido do primeiro algarismo diferente de zero, existente à esquerda. Quando se está operando na base decimal, o empréstimo é de valor 10.

#### Overflow

Ocorre sempre que, numa operação aritmética, ocorre um “vai-um” para o bit de sinal.

#### ***Exemplos:***

1) Somar os números (+13) e (+12)

		11	
+13	0	01101	
+12	0	01100	
		<u>011001</u>	
		011001	

**Mesmo sinal: 0**

2) Somar (-17) e (-9)

		1	
-17	1	10001	
-9	1	01001	
		<u>101001</u>	
		111010	

**Mesmo sinal: 1**

3) Somar (-21) e (+10)

$$\begin{array}{r} \phantom{00}00 \\ -21 \quad 1\cancel{1}0101 \\ +10 \quad 001010 \\ \hline 101011 \end{array}$$

**Sinais diferentes: maior magnitude = 21**

**Sinal: 1**

4) Somar (+17) e (+15)

$$\begin{array}{r} \phantom{0}1111 \\ +17 \quad 010001 \\ +15 \quad 001111 \\ \hline 000000 \end{array}$$

**Mesmo sinal: 0**

**Estouro de bits: vai-um para o bit de sinal**

#### Algoritmo para subtração

- Troca-se o sinal do subtraendo
- Procede-se como na soma

#### Exemplos:

1) Efetuar a subtração  $101101 - 1000111$ :

$$\begin{array}{r} \phantom{0}0\phantom{0} \\ -10\cancel{0}\cancel{0}101 \\ \phantom{-}1000111 \\ \hline 000110 \end{array}$$

2) Efetuar a subtração  $(-18) - (+12)$

$$(-18) - (+12) = (-18) + (-12)$$

Mesmo sinal: 1

$$\begin{array}{r} \phantom{0}010010 \\ -18 \quad 1010010 \\ -12 \quad 1001100 \\ \hline 1011110 = -30 \end{array}$$

3) Efetuar a subtração  $(-27) - (-14)$

$$(-27) - (-14) = (-27) + (+14)$$

Sinais diferentes

Maior magnitude: 27

Sinal: 1

$$\begin{array}{r} \phantom{0}0\phantom{0} \\ -27 \quad 10\cancel{1}1011 \\ +14 \quad 0001110 \\ \hline 1001101 = -13 \end{array}$$

## Aritmética em complemento de 2

A subtração é realizada através da soma com o complemento de 2 do subtraendo (não necessita de circuitos para subtração).

### Algoritmo da soma em complemento de 2

- 1) Somar os 2 números, bit a bit, inclusive o bit de sinal.
- 2) Desprezar o último “vai-um” (para fora do número) se houver.
- 3) Se, simultaneamente, ocorrer “vai-um” para o bit de sinal e para fora do número, ou se ambos não existirem, o resultado está correto.
- 4) Se ocorrer apenas um dos dois “vai-um”, o resultado está incorreto (ocorreu *overflow*).

### Para a subtração

O subtraendo deve ser colocado em complemento de 2 antes de iniciar o algoritmo da soma.

Exemplo: Realizar as seguintes operações (considerar números binários com 6 bits) e analisar se o valor está correto ou não.

a)  $(+12) + (+7)$

```
  11
001100
000111
010011 = +19 (valor correto, sem ambos os vai-um)
```

b)  $(+23) + (+20)$

```
  1 1
010111
010100
101011 (resultado incorreto, vai-um somente para o bit de sinal)
Isto já era esperado, pois se pode ter valores somente até  $(25 - 1) = +31$  e dois valores positivos não podem dar um valor negativo.
```

c)  $(+15) + (-13)$

```
13 = 001101
Complemento de 2 de 13 =  $110010 + 1 = 110011$ 
111111
001111
110011
000010 = +2 (valor correto, ocorreram ambos os “vai-um”)
```

d)  $(+20) - (+17) = (+20) + (-17)$

```
17 = 010001
Complemento de 2 de 17 =  $101110 + 1 = 101111$ 
1111
010100
101111
000011 = +3 (valor correto, ocorreram ambos os “vai-um”)
```

e)  $(+17) - (+20) = (+17) + (-20)$

```
20 = 010100
Complemento de 2 de 20 =  $101011 + 1 = 101100$ 
010001
101100
111101 (valor correto, sem ambos os “vai-um”)
O resultado é negativo. A magnitude será:  $00010 + 1 = 00011 = -3$ 
```

## Exercícios

- 1) Transformar na notação binária e realizar as operações solicitadas. Utilize a representação em sinal magnitude, com 4 bits, e indique quando houver *overflow* (estouro de bits).
  - a)  $3 + 2$
  - b)  $5 + 4$
  - c)  $4 - 2$
  - d)  $2 - 4$
- 2) Utilizando a representação em sinal magnitude, com 7 bits, realize as operações. Indique se houve *overflow* ou não e explique.
  - a)  $(-18) - (+12)$
  - b)  $(-27) - (-14)$
- 3) Transformar em binário e realizar as operações solicitadas (para números negativos, utilize a notação em complemento de 2, com 4 bits). Indique quando houver *overflow* (estouro de bits).
  - a)  $3 + 2$
  - b)  $4 - 2$
  - c)  $2 - 4$
  - d)  $5 + 4$
- 4) Transformar em binário, com 5 bits, e realizar as operações solicitadas (utilize a notação em sinal magnitude e complemento de 2). Verifique se houve *overflow*.
  - a)  $+10 + 8$
  - b)  $10 - 8$
- 5) Realize as operações a seguir, em binário, com 6 bits, utilizando as notações em sinal magnitude e complemento de 2. Analise se o resultado está correto ou não.
  - a)  $(+8) - (+4)$
  - b)  $(+4) - (+8)$
- 6) Utilizando a representação em sinal/magnitude e complemento de 2, com 8 bits, realize as operações abaixo, indicando se houve estouro de bits **ou não** e explicando porquê.
  - a)  $+122 + 93$
  - b)  $-43 + 56$
- 7) Utilizando a representação em sinal/magnitude, com 6 bits, realize as operações abaixo, indicando se houve estouro de bits **ou não** e explicando porquê. (1,0 ponto)
  - a)  $19 + 16$
  - b)  $20 - 12$
  - c)  $+25 - 10$
  - d)  $14 + 12$
  - e)  $20 - 15$

### Respostas dos exercícios

- 1) a) 0101
- b) 1001

- c) 0010
- d) 1010

- 2) a) 1011110
- b) 1001101

- 3) a) 0101
- b) 0010
- c) 1110
- d) 1001

- 4) a) 10010 e 10010
- b) 00010 e 00010

- 5) a) 000100 e 000100
- b) 100100 e 111100

- 6) a) 01010111 – ocorreu overflow, vai um para o bit de sinal
- b) 00001101 – não ocorreu overflow

- 7) a) 000011 – não ocorreu overflow, não houve vai-um para o bit de sinal
- b) 001000 – não ocorreu overflow
- c) 001111 – não ocorreu overflow
- d) 011010 - não ocorreu overflow, não houve vai-um para o bit de sinal
- e) 000101 - não ocorreu overflow