



# Números Binários

## Operações Aritméticas

Prof. Patrícia G. P. Magalhães

# Soma de números Binários

- Se faz semelhante as operações com números decimais, mas lembre-se que estamos falando de binários, assim:
- Binário:  $1 + 1 = 10$
- Decimal:  $1 + 1 = 2$
- Semelhança:  $2_{10} = 10_2$

## Adição de Binários

- $1 + 0 = 1$
- $0 + 1 = 1$
- $0 + 0 = 0$
- $1 + 1 = 0$ , e vai “um” ou  $10_2$

			1		← Transporte do Bit (vai 1)
	1	0	0	1	$9_{10}$
+	0	1	0	1	$5_{10}$
<hr/>					
	1	1	1	0	$14_{10}$

# Subtração de números Binários

- A subtração é mais complexa por dispormos somente dos algarismos 0 e 1. Assim, na conta “0 – 1” necessita-se “pedir emprestado” do próximo algarismo diferente de zero à esquerda. A operação é exatamente a mesma de uma subtração de números decimais. Por Exemplo:

$$\begin{array}{r} \text{-1 ("VEM UM")} \\ 1\ 1\ 0\ 1\ 1\ 0 \\ -\quad 1\ 1\ 0\ 0 \\ \hline 1\ 0\ 1\ 0\ 1\ 0 \end{array}$$

# Overflow ou estouro do limite

- Caso durante a realização da operação estoure a quantidade máxima de bits no registrador, ocorre um overflow, quando isso ocorre o processador sinaliza informando. Vamos a um exemplo:
- Somar os números  $15_{10} + 49_{10}$  num registrador de 6 bits.
- Observe que neste caso o 64 não poderá ser guardado num registrador de 6 bits.

Sua conta dará ERRO!

<b>1 1 1 1 1 1</b>	<b>(“vai um”)</b>
$15 = (0\ 0\ 1\ 1\ 1\ 1)_2$	
<u><math>49 = (1\ 1\ 0\ 0\ 0\ 1)_2</math></u>	
$64 = (0\ 0\ 0\ 0\ 0\ 0)_2$	

# Representação de Números Negativos

- **Sinal - Magnitude**

O bit mais à esquerda de um número binário é reservado para indicar se o número é positivo ou negativo e é conhecido como bit de sinal, onde, convencionalmente, “0” indica um número positivo e “1” um número negativo. O restante indica a magnitude do número (ou seu valor absoluto).

Sinal	Magnitude
-------	-----------

- Por exemplo, para representar o número - 7 em um registrador de 6 bits:

$$- 7 = 1 0 0 1 1 1$$

- O primeiro bit foi usado para representar o sinal (negativo) e apenas 5 bits foram usados para representar o valor absoluto (7).

# Como saber quantos números cabem num registrador? Seguindo o método Sinal –Magnitude.

- Use a Fórmula:

$$-\left(2^{(n-1)} - 1\right) \leq X \leq +\left(2^{(n-1)} - 1\right), \text{ onde } n \text{ é a quantidade de bits do número.}$$

- Para um registrador de 6 bits:

$$\text{de } -(2^{6-1} - 1) \text{ a } +(2^{6-1} - 1)$$

$$\text{de } -(2^5 - 1) \text{ a } +(2^5 - 1)$$

**Faixa: de -31 a +31**

- Assim, você saberá se irá ocorrer Overflow, sem precisar fazer a conta.

# Representação de número negativo

## Complemento a um

- Neste método, para passar o número para negativo, todos os números serão invertidos, incluindo o do sinal.
- Por exemplo: + 3 para - 3

+3 = 0 0 0 0 1 1

-3 = 1 1 1 1 0 0 -> (complemento a um – todos os bits invertidos)

# Vantagens e Desvantagens

- A representação **sinal-magnitude** apresenta como vantagem: basta verificar o bit de sinal para saber se o número é positivo ou negativo.
- Como desvantagem: ocupa 1 bit do registrador para sinal e também o número zero pode ser representado de duas maneiras distintas. Não é muito utilizado.

$$+0_{10} = 0000_2$$

$$-0_{10} = 1000_2$$

- A representação **complemento a um**, possui como vantagem: há apenas um componente para soma e subtração, diminuindo custo e o tempo de execução.
- Entretanto, assim como em sinal-magnitude, há duas representações para o número zero.

# Representação de número negativo

**Complemento de dois:** São utilizados dois passos para a representação de número negativo:

- **Primeiro passo:** Pegue o número positivo e troque 0 por 1 e vice-versa, incluindo o bit do sinal (segundo o mesmo processo do complemento a um);
- **Segundo passo:** Utilizando o resultado obtido no primeiro passo soma-se 1 (em binário), desprezando-se o último transporte, se existir.
- Por exemplo: + 10 para - 10 num registrador de 8 bits

	+10 =	0	0	0	0	1	0	1	0
<b>Primeiro passo</b>	-10 =	1	1	1	1	0	1	0	1
<b>Segundo passo</b>	+								1
<b>Resultado</b>	-10 =	1	1	1	1	0	1	1	0

# Propriedades do Complemento de 2

- **Vantagens:** Negar um número duas vezes produz o número original; O número zero tem apenas uma representação, pois o último transporte é desprezado. É o método mais utilizado.

	0	=	0	0	0	0	0	0
<b>Primeiro passo</b>	-0	=	1	1	1	1	1	1
<b>Segundo passo</b>		+						1
<b>Resultado</b>	-0	=	4	0	0	0	0	0

- Para saber até qual número cabe no registrador, use a **fórmula:**

$$\left(-2^{(n-1)}\right) \leq X \leq \left(2^{(n-1)} - 1\right), \text{ onde } n \text{ é a quantidade de bits do número.}$$

# Exercícios

1. Converta os números para Binários e realize as seguintes Operações com os mesmos:

a)  $16_{10} + 15_{10}$  (Registrador de 6 bits) =

b)  $30_{10} - 21_{10}$  (Registrador de 7 bits) =

c)  $45_{10} + 13_{10}$  (Registrador de 8 bits) =

d)  $20_{10} - 06_{10}$  (Registrador de 5 bits) =

2. Passe números positivos para binários e transforme-o em negativos utilizando a representação por sinal-magnitude, utilizando um registrador de 8 bits:

a)  $+ 12_{10}$  para  $- 12_{10} =$

b)  $+ 35_{10}$  para  $- 35_{10} =$

c)  $+ 324_{10}$  para  $- 324_{10} =$

3. Passe números positivos para binários e transforme-o em negativos utilizando a representação por Complemento a um , utilizando um registrador de 8 bits:

a)  $+ 12_{10}$  para  $- 12_{10} =$

b)  $+ 35_{10}$  para  $- 35_{10} =$

c)  $+ 134_{10}$  para  $- 134_{10} =$

4. Passe números positivos para binários e transforme-o em negativos utilizando a representação por Complemento de 2 , utilizando um registrador de 8 bits :

a)  $+ 12_{10}$  para  $- 12_{10} =$

b)  $+ 35_{10}$  para  $- 35_{10} =$

c)  $+ 225_{10}$  para  $- 225_{10} =$

# Referências

- FERREIRA, F. C. **Módulo de:** Arquitetura de Computadores. Vila Velha-ES: ESAB, 2009.
- Vídeos do site: [Youtube.com](https://www.youtube.com)