UEZO
CENTRO UNIVERSITÁRIO
ESTADUAL DA ZONA OESTE

Arquitetura de Computadores
Sistema de Numeração

Apresentado por Prof. Fred Sauer

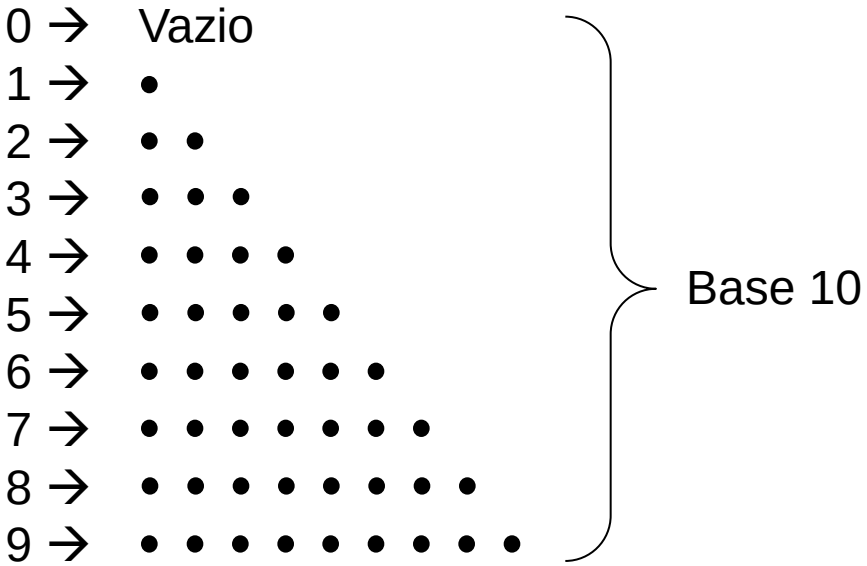
Mat. Elaborado por Prof. Ricardo Quintão

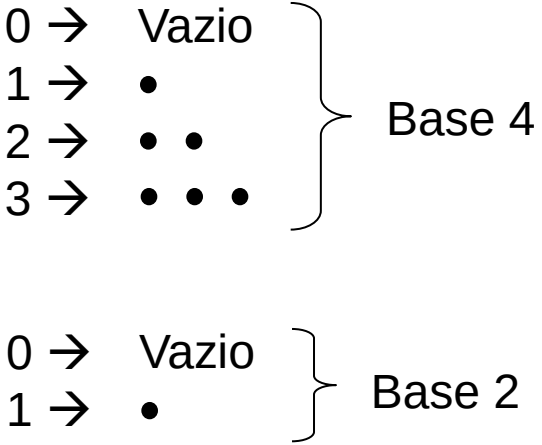
UEZO
CENTRO UNIVERSITÁRIO
ESTADUAL DA ZONA OESTE

Bases Numéricas

- A base de representação numérica de um número está relacionada com a quantidade de símbolos disponíveis para representar este número.
- Se tivermos 10 símbolos para as representações numéricas, este número estará sendo representado na base 10; se tivermos 8 símbolos, ele estará na base 8, e assim sucessivamente.

Bases Numéricas





UEZO
CENTRO UNIVERSITÁRIO
ESTADUAL DA ZONA OESTE

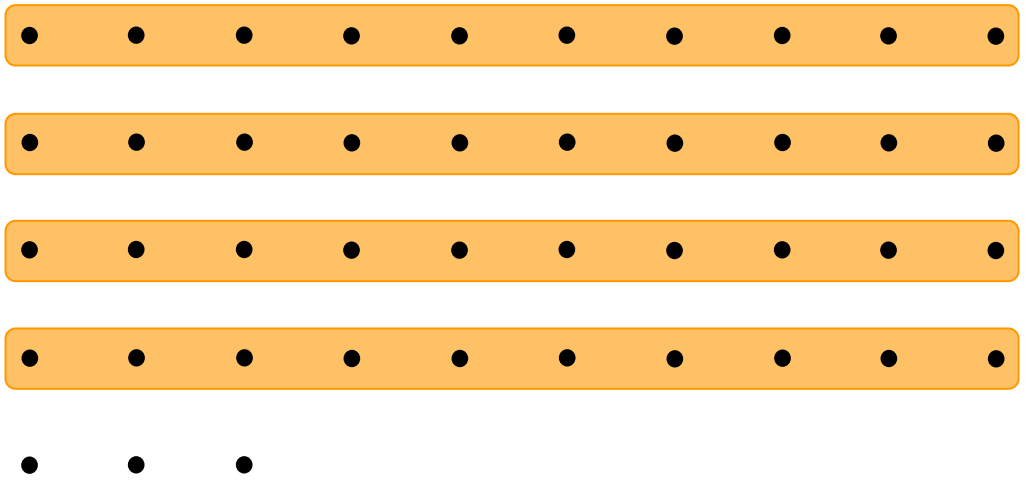
Bases Numéricas

- Como representar uma quantidade infinita de valores usando uma quantidade finita de símbolos para a representação numérica?

- A maneira criada para representar qualquer valor com um número limitado de símbolos foi o da agrupagem.
- Se montarmos grupos em que a quantidade de elementos é a mesma quantidade de símbolos, isto é, a base do número, a quantidade de objetos que sobrar será inferior à base, sendo então possível a sua representação.

Bases Numéricas

- Vamos entender graficamente como é feita esta representação.
- Vamos fazer um exemplo na base 10, que é a nossa base diária.

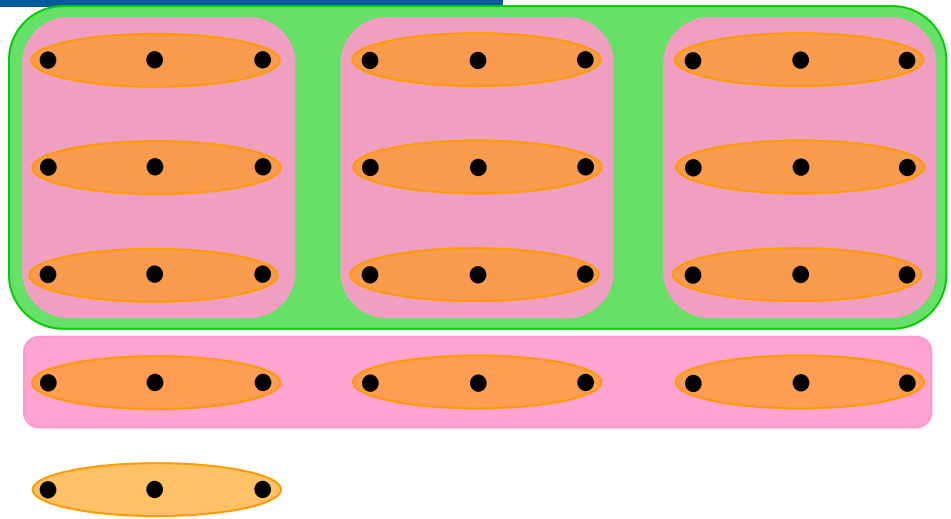


$(4\ \ 3)_{10}$

Bases Numéricas

- Agora vamos fazer um exemplo na base 3 com as mesmas 43 bolinhas.

Bases Numéricas



$(1\ 1\ 2\ 1)_3$

43	3
13	14
1	2

3	
4	
1	

3
1

$$(1\ 1\ 2\ 1)_3 = 1 \times 3 \times 3 \times 3 + 1 \times 3 \times 3 + 2 \times 3 + 1$$

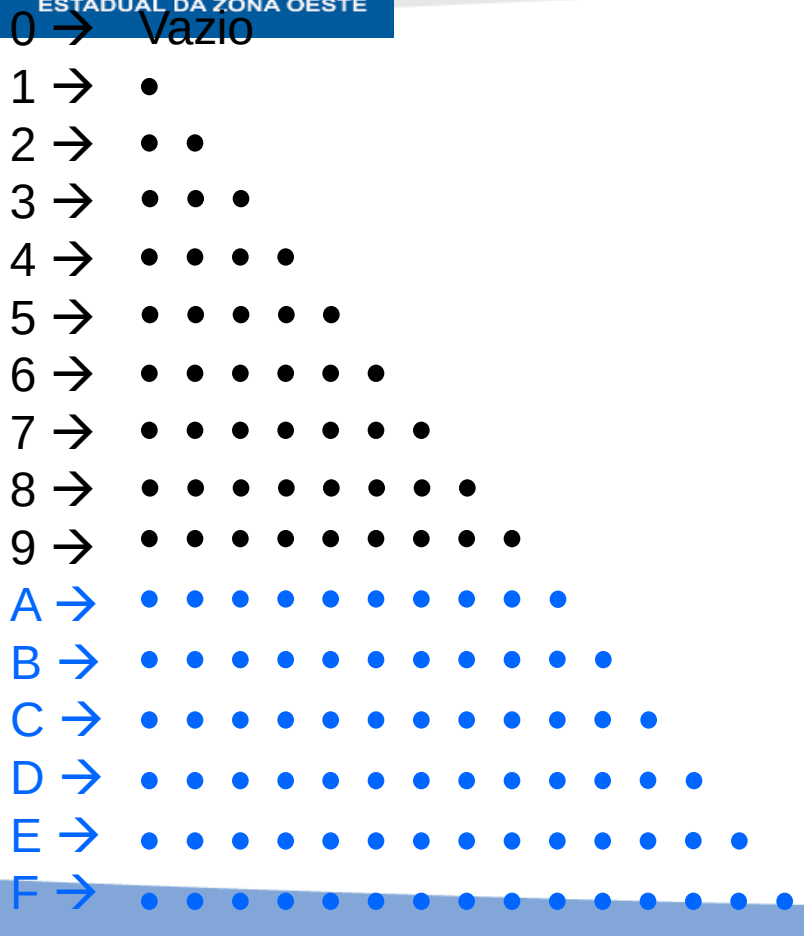
$$= 1 \times 3^3 + 1 \times 3^2 + 2 \times 3^1 + 1 \times 3^0$$

$$= 27 + 9 + 6 + 1 = (43)_{10}$$

Bases Numéricas

- Como será a representação no caso de bases superiores a base 10?
- Neste caso, precisaremos de mais símbolos para representar estes novos valores. Para facilitar, utilizou-se o alfabeto.

Bases Numéricas Superiores a Base 10

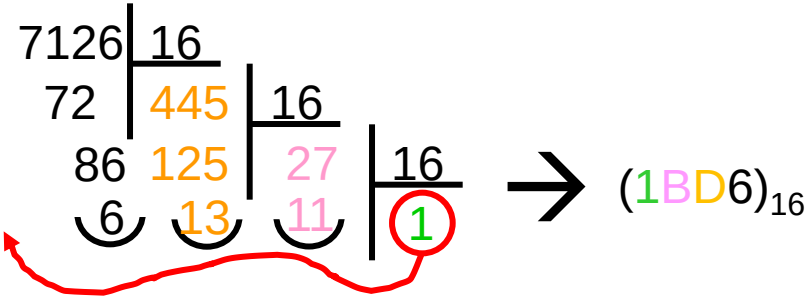


Base 16

Bases Numéricas

- Exemplo de conversão de bases superiores a base 10.

$(7126)_{10} \rightarrow (\quad)_{16}$



$\rightarrow (1BD6)_{16}$

Bases Numéricas

- Exemplo de conversão de bases superiores a base 10.

$(1BD6)_{16} \rightarrow (\quad)_{10}$

$$(1\,|\,B\,|\,D\,|\,6)_{16} \quad \text{(posições: } 3\,|\,2\,|\,1\,|\,0\text{)}$$

$$= 1 \times 16^3 + 11 \times 16^2 + 13 \times 16^1 + 6 \times 16^0$$

$$= 4096 + 2816 + 208 + 6$$

$$= (7126)_{10}$$

UEZO
CENTRO UNIVERSITÁRIO
ESTADUAL DA ZONA OESTE

Bases Numéricas

Conversão de bases que são potências entre si.

- O que significa dizer que duas bases são potências entre si?
- Primeiramente, devemos representar a base maior numa potência da base menor. ($B = b^e$) onde $B > b$ & e → número inteiro maior que 1.
- Se for possível escrever as potências acima, então as bases são potências entre si.

Exemplo:

Base 4 ⇔ Base 2

$4 = 2^2$ → são potências entre si

Bases Numéricas

Conversão de bases que são potências entre si.

- O próximo passo é montar uma tabela de conversão de duas colunas onde cada coluna representará uma base.
- Esta tabela terá uma quantidade de linhas de conversão igual ao valor da maior base (B). Normalmente coloca-se uma linha adicional para o cabeçalho das colunas.
- Na coluna da base maior (B) coloca-se, em sequência e começando do zero, todos os seus símbolos.
- Na coluna da base menor (b), coloca-se o valor equivalente escrito com uma quantidade de dígitos igual ao expoente da base menor (e).
- **A quantidade de dígitos referente aos valores da menor base deve ser igual ao expoente usado nesta base (e).**
- Pronto! Já temos a tabela, agora é só usar.

Bases Numéricas

Conversão de bases que são potências entre si.

- Exemplo:

Montar a tabela que converta da base 2 para a base 4.

Resolução:

Base 4 ⇔ base 2 (4 = 2^2 → são potências entre si)

Total de linhas da tabela: 4

Total de dígitos da base 2: 2

Bases Numéricas

Conversão de bases que são potências entre si.

Base 4	Base 2
0	00
1	01
2	10
3	11

Exemplo:

a) $(323121302)_4 \rightarrow (\quad)_2$

$(3\ 2\ 3\ 1\ 2\ 1\ 3\ 0\ 2)_4 = (1110110110011100 10)_2$

11 10 11 01 10 01 11 00 10

b) $(11110101111010101 0110)_2 \rightarrow (\quad)_4$

$(01111010111101010110)_2 = (13223311112)_4$

1 3 2 2 3 3 1 1 1 1 2

Bases Numéricas

Conversão de bases que são potências entre si.

- Exemplo:

Montar a tabela que converta da base 2 para a base 8.

Resolução:

Base 8 ⇔ base 2 (8 = 2^3 → são potências entre si)

Total de linhas da tabela: 8

Total de dígitos da base 2: 3

Bases Numéricas

Conversão de bases que são potências entre si.

Base 8	Base 2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Exemplo:

a) $(637254)_8 \rightarrow (\quad)_2$

$(6 \mid 3 \mid 7 \mid 2 \mid 5 \mid 4)_8 = (110011111010101100)_2$

110 | 011 | 111 | 010 | 101 | 100

b) $(11110101111010101010110)_2 \rightarrow (\quad)_8$

$(111 \mid 101 \mid 011 \mid 110 \mid 101 \mid 010 \mid 110)_2 = (7536526)_8$

7 | 5 | 3 | 6 | 5 | 2 | 6

Bases Numéricas

Conversão de bases que são potências entre si.

- Exemplo:

Montar a tabela que converta da base 2 para a base 16.

Resolução:

Base 16 ⇔ base 2 (16 = 2^4 → são potências entre si)

Total de linhas da tabela: 16

Total de dígitos da base 2: 4

Bases Numéricas

Conversão de bases que são potências entre si.

Base 16	Base 2
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Exemplo:

a) $(AD3B9)_{16} \rightarrow (\quad)_2$

$(A \mid D \mid 3 \mid B \mid 9)_{16} = (10101101001110111001)_2$

1010 1101 0011 1011 1001

b) $(11110101111010101010110)_2 \rightarrow (\quad)_{16}$

$(0001 \mid 1110 \mid 1011 \mid 1101 \mid 0101 \mid 0110)_2 = (1EBD56)_{16}$

1 | E | B | D | 5 | 6

Bases Numéricas

Representação de números com sinal.

- Os computadores digitais utilizam principalmente três métodos para representar números com sinal:
  - Sinal e Magnitude
  - Complemento a um
  - Complemento a dois

Para os três casos é importante definir a quantidade de bits usada na representação do número, porque o bit mais significativo (bit da esquerda) representará o sinal.

0 → Positivo
1 → Negativo

Bases Numéricas

Representação de números com sinal.

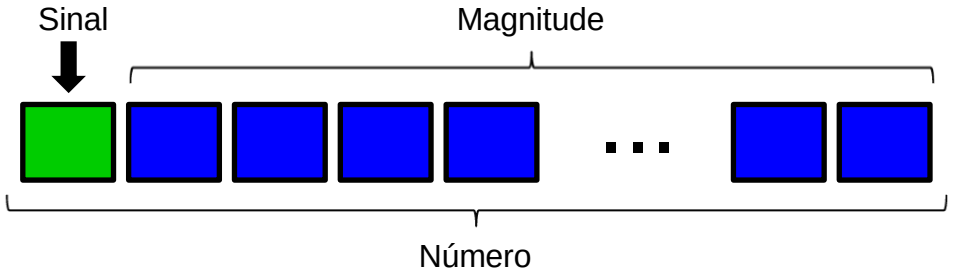
- Como o zero à esquerda do número não possui significado numérico, os números positivos ficam idênticos nos três casos anteriores. Sendo assim, usaremos os números positivos como ponto comum na conversão de bases.
- Na nossa representação em decimal utilizamos o formato Sinal e Magnitude. Se precisarmos representar um número negativo em binário, partiremos da sua representação positiva e depois trocaremos o sinal.
- Para cada formato, existe uma maneira diferente de trocar o sinal. Dependendo do formato que queremos representar, usaremos o método adequado a este formato para trocar o sinal do número.

Bases Numéricas

Representação de números com sinal.

Sinal e Magnitude

Esta representação agrupa o bit de sinal com um valor (magnitude) que representa o valor absoluto do número como ilustrado abaixo:



Para trocar o sinal de um número no formato Sinal e Magnitude basta mudar o valor do bit de sinal.

Bases Numéricas

Representação de números com sinal.

Sinal e Magnitude

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-45)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 45, para binário.

$(45)_{10} = (101101)_2$

Segundo passo: Completar (à esquerda) com o valor zero os bits que faltam para chegar a **n.** Depois ajuste o bit mais à esquerda para representar o sinal desejado. Neste caso o bit de sinal vale 1 pois o valor é negativo.

$(-45)_{10} = (10101101)_2 = (AD)_{16}$

Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Sinal e Magnitude

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-7)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 7, para binário.

$(7)_{10} = (111)_2$

Segundo passo: Completar (à esquerda) com o valor zero os bits que faltam para chegar a **n.** Depois ajuste o bit mais à esquerda para representar o sinal desejado. Neste caso o bit de sinal vale 1 pois o valor é negativo.

$(-7)_{10} = (10000111)_2 = (87)_{16}$

→ Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Sinal e Magnitude

Um ponto negativo em relação a esta representação é a existência de duas formas de representar o valor zero. Veja abaixo:

Para um número com 8 bits, temos:

$(0)_{10} = (00000000)_2$

$(-0)_{10} = (10000000)_2$

Como $(0)_{10} = (-0)_{10}$
Então, $(00000000)_2 = (10000000)_2$

Isto é, **<u>duas</u>** representações do zero.

Faixa de Valores

Para um número formado por **n** bits (incluindo o bit de sinal), é possível representar valores que vão deste $-2^{n-1} + 1$ até $2^{n-1} - 1$.

Exemplo:

Para um número representado por 8 bits, isto é n = 8, teremos como faixa de valores:

$-2^{8-1} + 1$ até $2^{8-1} - 1$ → $-128 + 1$ até $128 - 1$

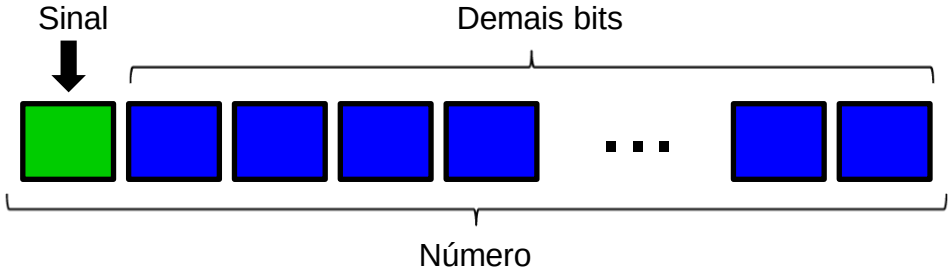
Faixa ➔ **-127 até 127**

Bases Numéricas

Representação de números com sinal.

Complemento a 1

Nesta representação, buscamos o valor que falta para chegar ao valor máximo do número, isto é, todos os bits valendo 1. No final do procedimento, o bit mais à esquerda continuará representando o sinal, porém os demais bits não representam a magnitude.



Para trocar o sinal de um número no formato de Complemento a 1 basta inverter o valor de todos os seus bits, isto é, o bit zero passa a valer um e o bit um passa a valer zero.

Bases Numéricas

Representação de números com sinal.

Complemento a 1

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-45)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 45, para binário e completar os bits.

$(45)_{10} = (101101)_2 = (00101101)_2$

Segundo passo: Inverter todos os bits do número. Repare que o bit mais à esquerda continua representando o sinal.

$(-45)_{10} = (11010010)_2 = (D2)_{16}$

Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Complemento a 1

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-7)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 7, para binário e completar os bits.

$(7)_{10} = (111)_2 = (00000111)_2$

Segundo passo: Inverter todos os bits do número. Repare que o bit mais à esquerda continua representando o sinal.

$(-7)_{10} = (11111000)_2 = (F8)_{16}$

→ Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Complemento a 1

Um ponto negativo em relação a esta representação é a existência de duas formas de representar o valor zero. Veja abaixo:

Para um número com 8 bits, temos:

$(0)_{10} = (00000000)_2$

$(-0)_{10} = (11111111)_2$

Como $(0)_{10} = (-0)_{10}$
Então, $(00000000)_2 = (11111111)_2$

Isto é, **<u>duas</u>** representações do zero.

Faixa de Valores

Para um número formado por **n** bits (incluindo o bit de sinal), é possível representar valores que vão deste $-2^{n-1} + 1$ até $2^{n-1} - 1$.

Exemplo:

Para um número representado por 8 bits, isto é n = 8, teremos como faixa de valores:

$-2^{8-1} + 1$ até $2^{8-1} - 1$ → $-128 + 1$ até $128 - 1$

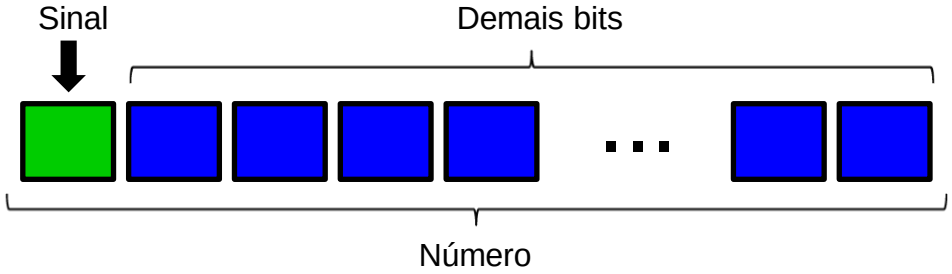
Faixa ➔ **-127 até 127**

Bases Numéricas

Representação de números com sinal.

Complemento a 2

Nesta representação, buscamos o valor que falta para chegar 2^n, onde n representa o total de bits do número. No final do procedimento, o bit mais à esquerda continuará representando o sinal, porém os demais bits não representam a magnitude.



Para trocar o sinal de um número no formato de Complemento a 2 basta inverter o valor de todos os seus bits e depois somar 1.

Bases Numéricas

Representação de números com sinal.

Complemento a 2

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-45)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 45, para binário e completar os bits.

$(45)_{10} = (101101)_2 = (00101101)_2$

Segundo passo: Inverter todos os bits do número e depois somar 1. Repare que após o procedimento o bit mais à esquerda continua representando o sinal.

$(-45)_{10} = (11010010)_2 + 1 = (11010011)_2 = (D3)_{16}$

Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Complemento a 2

Converta os valores abaixo para a sua representação em binário e em seguida apresente-os em hexadecimal (base 16). O valor de **n** representa a quantidade de bits utilizada na representação binária.

a) $(-7)_{10}$, n = 8 bits.

Primeiro passo: Converter o valor sem sinal, 7, para binário e completar os bits.

$(7)_{10} = (111)_2 = (00000111)_2$

Segundo passo: Inverter todos os bits do número e depois somar 1. Repare que após o procedimento o bit mais à esquerda continua representando o sinal.

$(-7)_{10} = (11111000)_2 + 1 = (11111001)_2 = (F9)_{16}$

Bit de Sinal

Bases Numéricas

Representação de números com sinal.

Complemento a 2

Diferente dos casos anteriores, esta representação possui apenas uma forma de representar o valor zero. Veja abaixo:

Para um número com 8 bits, temos:

$(0)_{10} = (00000000)_2$

$(-0)_{10} = (11111111)_2 + 1 = (100000000)_2$

Como o número possui apenas 8 bits, o nono bit representado em azul deve ser descartado.

$(-0)_{10} = (00000000)_2$

Isto é, **uma** única representação do zero.

Faixa de Valores

Para um número formado por **n** bits (incluindo o bit de sinal), é possível representar valores que vão deste -2^{n-1} até $2^{n-1} - 1$.

Exemplo:

Para um número representado por 8 bits, isto é n = 8, teremos como faixa de valores:

-2^{8-1} até $2^{8-1} - 1$ → -128 até 128 – 1

Faixa ➔ **-128 até 127**

Bases Numéricas

Aritmética com Complemento de 2.

Como que a CPU faz para saber se o número utilizado é com ou sem sinal? Afinal de contas, isto é importante na hora de usar o número? Vamos supor os valores em binário abaixo representados por 8 bits.

$(10000100)_2$ e $(00001110)_2$

Como seria feita a soma destes números?

Método tradicional

```
    11
  10000100
+ 00001110
  10010010
```

Considerando sem signal

```
    11
  10000100 → (+132)10
+ 00001110 →  (+14)10
  10010010 → (+146)10
```

Considerando com signal

```
    11
  10000100 → (-124)10
+ 00001110 →  (+14)10
  10010010 → (-110)10
```

Olha só que interessante! A mesma conta serve tanto para números sem sinal, como para números com sinal. O que vai mudar é a interpretação do resultado feito pelo seu programa.