# CID

# Circuitos Digitais

## Aula 02 – Bases Numéricas

**Binário**
**Decimal**
**Octal**
**Hexadecimal**

# Revisão da Aula 1...

Números escritos em bases diferentes...

**Decimal (base 10): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
**Binário (base 2): 0, 1**
**Octal (base 8): 0, 1, 2, 3, 4, 5, 6, 7, 8**
**Hexadecimal (base 16): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**

**Base B genérica: ( 0 ) a ( B – 1 )**

# Revisão da Aula 1...

**Conversão Binário para Octal** - Agrupam-se de 3 em 3 algarismos a partir da direita e complementando-se com zeros caso sejam necessários. Faz-se com esses agrupamentos a conversão direta.

EX: $01011111_{(2)} = [001][011][111] = 137_{(8)}$

**Conversão Binário para Decimal** - Multiplicam-se o algarismos do número binário por 2 elevado a potência da posição menos 1.

EX: $1101101_{(2)} = 1.2^6 + 1.2^5 + 0.2^4 + 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = 109_{(10)}$

**Conversão Binário para Hexadecimal** - Agrupa-se de 4 em 4 algarismos a partir da direita e complementando-se com zeros caso sejam necessários. Faz-se com esses agrupamentos a conversão direta.

EX: $101011111_{(2)} = [0001][0101][1111] = 15F_{(16)}$

# Revisão da Aula 1...

**Conversão Octal e Hexadecimal para Binário** - Fazem-se as conversões diretas, algarismo por algarismo para um número binário com 3 bits (octal) e com 4 bits (hexadecimal).

EX: $426_{(8)} = [100][010][110] = 100010110_{(2)}$

EX: $39AB_{(8)} = [0011][1001][1010][1011] = 0011100110101011_{(2)}$

**Conversão Decimal para Binário** – Podemos fazer essa operação pelo método da soma dos pesos ou pelas divisões sucessivas.

$\mathbf{45_{(10)} = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0}$

$\mathbf{45_{(10)} = 64 + 32 + 16 + 08 + 04 + 02 + 01}$

$\mathbf{45_{(10)} = \ 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1}$

$\mathbf{45_{(10)} = 101101_{(2)}}$

$$
\begin{array}{r|l}
25 & 2 \\
\hline
\mathbf{1} \quad 12 & 2 \\
\hline
\mathbf{0} \quad 6 & 2 \\
\hline
\mathbf{0} \quad 3 & 2 \\
\hline
\mathbf{1} & \mathbf{1}
\end{array}
$$

$\mathbf{25_{(10)} = 11001_{(2)}}$

# Revisão da Aula 1...

**Conversão base (r) qualquer para Decimal**

$$D = \sum_{i=-n}^{p-1} d_i . r^i$$

**Conversão Decimal para base (b) qualquer**

Para a parte inteira: divisões sucessivas por (b);

Para a parte fracionária: multiplicações sucessivas por (b).

**Conversão Hexadecimal para Octal (e vice-versa)** – Converte-se para Binário e depois para Octal ou Hexadecimal.

# Aritmética Binária - Adição

As quatro regras básicas para a adição de dígitos binários (bits) são:

**0 + 0 = 0**
**0 + 1 = 1**
**1 + 0 = 1**
**1 + 1 = 10**

**1 1**
0 1 1
+ 0 0 1
-----------
**1 0 0**

**110 + 100 || 111 + 11 || 100 + 10 || 11 + 11**

**1010 1010 110 110**

# Aritmética Binária - Subtração

As quatro regras básicas para a subtração de dígitos binários (bits) são:

**0 - 0 = 0**
**10 - 1 = 1**
**1 - 0 = 1**
**1 - 1 = 0**

```
   0
   1̶ ¹0 1
-  0  1  1
-----------
   0 1 0
```

**11 - 10**

# Aritmética Binária - Multiplicação

As quatro regras básicas para a subtração de dígitos binários (bits) são:

**0 x 0 = 0**
**0 x 1 = 0**
**1 x 0 = 0**
**1 x 1 = 1**

```
      1  1
   x  1  1
-------------
      1     1  1
 +    1  1
-------------
   1  0  0  1
```

101 + 111

100011

# Aritmética Binária - Divisão

A divisão binária segue as mesmas regras da divisão decimal:

$$\begin{array}{r|l} 110 & 10 \\ \hline 10 & 11 \\ 0 & \end{array}$$

**110 / 11**

# Aritmética Octal e Hexadecimal

As operações de adição, subtração, multiplicação e divisão em sistemas octais e hexadecimais seguem os mesmos padrões das operações binárias.

# Complemento de 1 e de 2 em Números Binários

O complemento de 1 e de 2 de um número binário são importantes porque eles permitem a representação de números negativos. O método da aritmética do complemento de 2 é geralmente usado em computadores na operação com números negativos.

**Complemento de 1:** trocam-se todos os 1s por 0s e todos os 0s por 1s.

Exemplo: 1 0 1 1 0 0 1 0 – Número binário
0 1 0 0 1 1 0 1 – Complemento de 1

**Complemento de 2:** soma-se 1 ao LSB do complemento de 1.

Exemplo: 1 0 1 1 0 0 1 0 – Número binário
0 1 0 0 1 1 0 1 – Complemento de 1
\+ 1
------------------
**0 1 0 0 1 1 1 0 - Complemento de 2**

# Bit de Sinal

O bit mais a esquerda em um número binário sinalizado é o bit de sinal.

**Um bit de sinal 0 indica um número positivo e um bit de sinal 1 indica um número negativo.**

**Exemplo: 25 (8 bits) = 0 0 0 1 1 0 0 1**

**-25 (8 bits) = 1 0 0 1 1 0 0 1**

# Complemento de 1 e de 2

Os computadores usam a representação em complemento de 2 para os números inteiros negativos em todas as operações aritméticas. A razão para isso é que a subtração de um número é o mesmo que a adição com complemento de 2 do número.

**E daí???**

# Complemento de 1 e de 2

Os computadores usam a representação em complemento de 2 para os números inteiros negativos em todas as operações aritméticas. A razão para isso é que a subtração de um número é o mesmo que a adição com complemento de 2 do número.

Exemplo:

**10 – 7 =**

**10 + ( - 7 ) =**

$\mathbf{1\,0\,1\,0_{(2)} + 1\,0\,0\,1_{(2)} =}$

$\mathbf{1\,0\,0\;\,1\,1_{(2)} = 3_{(10)}}$

| Decimal | Binário s/ sinal | Binário (Compl. 2) |
|---|---|---|
| -8 | - | 1000 |
| -7 | - | 1001 |
| -6 | - | 1010 |
| -5 | - | 1011 |
| -4 | - | 1100 |
| -3 | - | 1101 |
| -2 | - | 1110 |
| -1 | - | 1111 |
| 0 | 000 | 0000 |
| 1 | 001 | 0001 |
| 2 | 010 | 0010 |
| 3 | 011 | 0011 |
| 4 | 100 | 0100 |
| 5 | 101 | 0101 |
| 6 | 110 | 0110 |
| 7 | 111 | 0111 |

# Complemento de 1 – Valores Decimais

Valores decimais de números positivos na forma de complemento de 1 são determinados somado os pesos de todos os bits 1s e ignorando os pesos relativos aos zeros. Os valores decimais de números negativos são determinados atribuindo um valor negativo ao peso do bit de sinal, somando os pesos relativos ao bits 1s e somando 1 ao resultado.

Exemplo 1:

00010111 = $(-2)^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ =

0 0 0 1 0 1 1 1 = 16 + 4 + 2 + 1 = **23**

Exemplo 2:

11101000 = $(-2)^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ =

1 1 1 0 1 0 0 0 = -128 + 64 + 32 + 8 = -24 **+1** **= -23**

# Complemento de 2 – Valores Decimais

Valores decimais de números positivos e negativos na forma de complemento de 2 são determinados somado os pesos de todos os bits 1s e ignorando as posições em que os bits são zeros. O peso do bit de sinal em números negativos é dado com um valor negativo.

Exemplo 1:

$01010110 = (-2)^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 =$

0 1 0 1 0 1 1 0 = 64 + 16 + 4 + 2 = **86**

Exemplo 2:

$10101010 = (-2)^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 =$

1 0 1 0 1 0 1 0 = -128 + 32 + 8 + 2 = **-86**

# Faixa de números que podem ser representados

Os bytes são conhecidos por agruparem o 8 bits e com isso podemos representar 256 números diferentes. Com dois bytes ou 16 bits podemos encontrar 65536 números diferentes. Com 4 bytes ou 32 bits podemos representar $4295 \times 10^9$ números diferentes.

**Para números sinalizados em complemento de 2 a faixa de valores para números de n bits é:**

$$-(2^{n-1}) \text{ a } (2^{n-1} - 1)$$

# Operações Aritméticas com números sinalizados

**Adição** – Os dois números de uma adição são a 1ª e a 2ª parcelas. O resultado é a soma. Podem ocorrer 4 situações:

1. **Os dois são positivos**
2. **O número positivo é maior**
3. **O número negativo é maior**
4. **Os dois são negativos**

# Operações Aritméticas com números sinalizados

**Adição** – Os dois são positivos:

**Exemplo:**

$$\begin{array}{r} 00000111 \\ +\ 00000100 \\ \hline 00001011 \end{array} \qquad \begin{array}{r} 7 \\ +\ 4 \\ \hline 11 \end{array}$$

# Operações Aritméticas com números sinalizados

**Adição** – Número positivo maior:

**Exemplo:**

$$\begin{array}{r} 00001111 \\ +\quad 11111010 \\ \hline 100001001 \end{array} \qquad\qquad \begin{array}{r} 15 \\ +\quad -6 \\ \hline 9 \end{array}$$

# Operações Aritméticas com números sinalizados

**Adição** – Número negativo maior:

**Exemplo:**

$$\begin{array}{r} 00010000 \\ +\ 11101000 \\ \hline 11111000 \end{array} \qquad \begin{array}{r} 16 \\ +\ -24 \\ \hline -8 \end{array}$$

# Operações Aritméticas com números sinalizados

**Adição** – Os dois são negativos:

**Exemplo:**

| | | | |
|---|---|---|---|
| | 11111011 | | -5 |
| + | 11110111 | + | -9 |
| | 111110010 | | -14 |

# Operações Aritméticas com números sinalizados

**Overflow** – Quando dois números são somados e o número de bits necessário para representar a soma excede o número de bits nos dois números, resulta em um overflow (transbordamento de capacidade) conforme indicado por um bit de sinal incorreto.

| | | |
|---|---|---|
| | 01111101 | 125 |
| + | 00111010 | + 58 |
| | 10110111 | 183 |

# CID

# Circuitos Digitais

## Aula 02 – Bases Numéricas

**Binário**
**Decimal**
**Octal**
**Hexadecimal**

**Obrigado!**