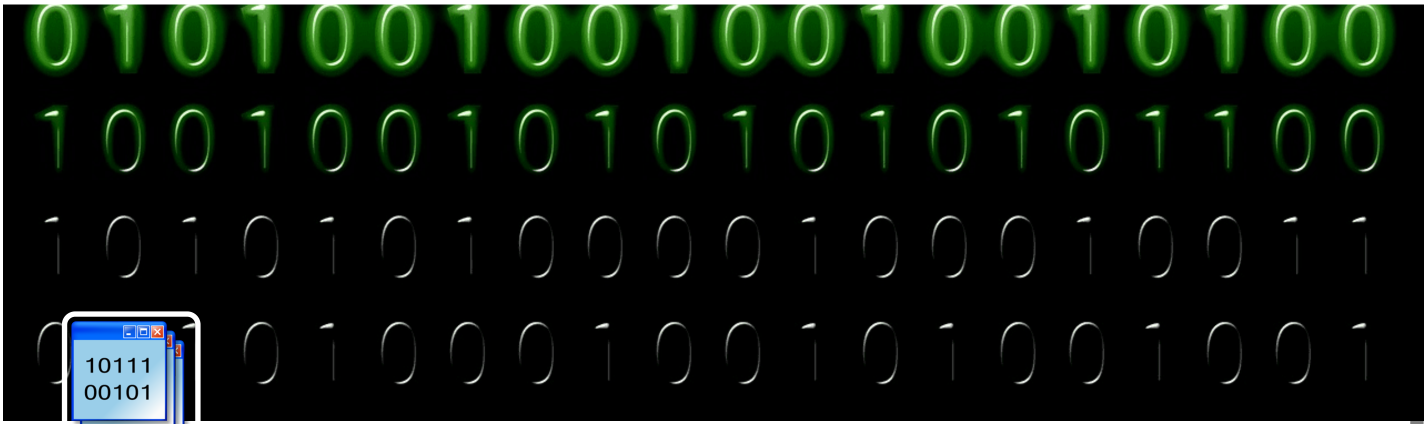


Sistemas de Numeração e Códigos

Apontamentos sobre sinais analógicos, sinais digitais, binário, base 4, 8 e 16

Page

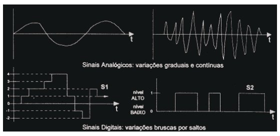
Sinais Analógicos e Sinais Digitais

Sinais Analógicos

- Os sinais podem variar continuamente

Sinais Digitais

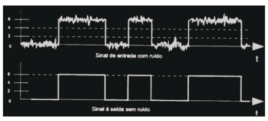
- Apenas podem assumir uma gama de valores discretos (só tomam determinados valores de tensão)
- Circuitos eletrónicos que funcionam apenas com 2 valores de amplitude são chamados de Digitais Binários



Utilizam 2 estados distintos → Papel importante do sistema binário (Sistema Octal e Sistema Hexadecimal têm um papel importante)

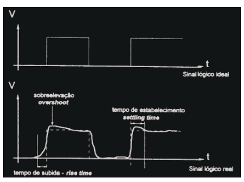
Porquê utilizar sinais binários?

- Simplicidade tecnológica e tolerância dos componentes utilizados
- Grande imunidade ao ruído



- Simplicidade de projeto
- Capacidade de integração
- Velocidade
- Parâmetros imutáveis com o tempo
- Economia

Sinais ideais/reais



Sistemas de Numeração

245 → Duzentos e Quarenta e Cinco

Qual o algoritmo de cálculo de um número?

$$2\times100 + 4\times10 + 5\times1$$
ou seja
$$2\times10^2 + 4\times10^1 + 5\times10^0$$

- **Base 10 (Sistema Decimal). PORQUÊ?**

  Símbolos Válidos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

  Total de símbolos = 10

Sistema Binário (Base 2)

- **Quais os dígitos válidos?**

  0 e 1 (Total 2 logo base 2)

- **Qual o equivalente decimal de 10110 no sistema binário**

  $$1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$$
  $$= 16 + 0 + 4 + 2 + 0 = 22_{(10)}$$

  - 10110 no sistema binário:
    - **LSB** - *Least Significant Bit*
    - **MSB** - *Most Significant Bit*

Conversão para decimal:

- **Número inteiro**

  $$\overset{4\,3\,2\,1\,0}{10110}_{(2)} = ?_{(10)}$$
  $$= 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$$
  $$= 16 + 0 + 4 + 2 + 0 = 22_{(10)}$$

- **Decimal**

  $$\overset{1\,0\,-1\,-2\,-3}{11{,}011}_{(2)}$$
  $$= 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$$
  $$= 2 + 1 + 0 + 0{,}25 + 0{,}125 = 3{,}375_{(10)}$$

- **Octal**
  - Símbolos Válidos: 0, 1, 2, 3, 4, 5, 6, 7
  - Total de símbolos: 8

  - Exemplo - Converta para o sistema decimal

    $$563_{(8)} = 5\times8^2 + 6\times8^1 + 3\times8^0 = 371_{(10)}$$
    $$26{,}6_{(8)} = 2\times8^1 + 6\times8^0 + 6\times8^{-1} = 22{,}75_{(10)}$$
    $$39{,}03_{(8)}$$

- **Hexadecimal**
  - Símbolos Válidos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - Total de Símbolos: 16

  - Exemplo - Converta para o sistema decimal:

    $$26_{(16)} = 2\times16^1 + 6\times16^0 = 38_{(10)}$$
    $$1A{,}C_{(16)} = 1\times16^1 + 10\times16^0 + 12\times16^{-1} = 26{,}75_{(10)}$$

Conversão

- Bases 4, 8 e 16 são potências de 2

- **Base 4 → 4 = 2^2**
  - 1 digito na base 4 representa 2 dígitos na base 2

- **Octal → 8 = 2^3**
  - 1 digito na base 8 representa 3 dígitos na base 2

- **Hexadecimal → 16 = 2^4**
  - 1 digito na base 16 representa 4 digitos na base 2

> ℹ️ Conversão direta através de uma tabela de conversão

Tabela de Conversão

Decimal	Binário	Octal	Hexadecimal	4-bits
0	0	0	0	0000
1	01	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	1010
11	1011	13	B	1011
12	1100	14	C	1100
13	1101	15	D	1101
14	1110	16	E	1110
15	1111	17	F	1111

Notas sobre logaritmos para trabalhar os números com parte fracionária

$$202{,}1_{(3)} = ?\ 2\times3^2 + 0\times3^1 + 2\times3^0 + 1\times3^{-1}$$
$$= 20{,}333333(3)_{(10)}$$
Qual o número de casas decimais?

De modo a não reduzir a capacidade de numeração devemos ter:

$$b'^{n'} \geq b^n$$

Onde:

b' = base de destino

n' = número de digitos na base de destino (incógnita)

b = base de origem

n = número de digitos na base de origem

$$a^x = b \Leftrightarrow x = log_a b \Leftrightarrow x = \frac{\ln b}{\ln a}$$

Por exemplo:

$$2^x = 8 \Leftrightarrow x = log_2 8 \Leftrightarrow x = \frac{\ln 8}{\ln 2} = 3$$

Então:

$$n' \geq n \times \frac{\log b}{\log b'}$$

Logo:

$$-\ 202{,}1_{(3)} = 2\times3^2 + 0\times3^1 + 2\times3^0 + 1\times3^{-1}$$
$$= 20{,}333333(3)_{(10)}$$

$b'^{n'} \geq b^n$ *(parte fraccionária)*

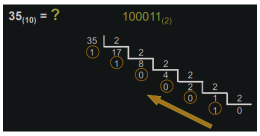
b' = base de destino
n' = número de dígitos na base de destino
b = base de origem
n = número de dígitos na base de origem

LOGO… $n' \geq n \times \frac{\log b}{\log b'} \Leftrightarrow n' \geq 1 \times \frac{\log 3}{\log 10} \Leftrightarrow n' \geq 1 \times 0{,}477$

e $202{,}1_{(3)} = 20{,}3_{(10)}$ (pq 2º digito < 5)

Decimal para binário

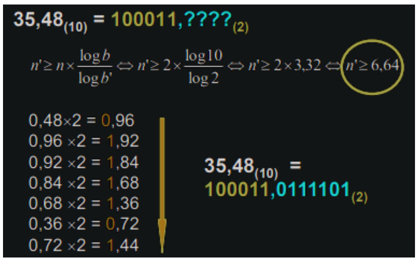
- O número 35 no sistema decimal = ?



> Processo idêntico para outros sistemas de numeração

Decimal para binário com parte fracionária

- O número 35,48 no sistema decimal = ?



Representação de números negativos

- Sinal + módulo

$= -101101_{(2)}$

- Complemento para 1

$= 1010010_{(2)}$

- Complemento para 2

$= 1010011_{(2)}$

Soma e Subtração binária

- Soma

$$\begin{array}{r}0\\+0\\\hline 0\end{array}\qquad\begin{array}{r}1\\+0\\\hline 1\end{array}\qquad\begin{array}{r}0\\+1\\\hline 1\end{array}\qquad\begin{array}{r}1\\+1\\\hline 10\end{array}\qquad\begin{array}{rcccccccc} & & 0 & 1 & 1 & 0 & 1 & 0 & 1\\ + & & 1 & 1 & 0 & 1 & 1 & 1 & 0\\\hline & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1\end{array}$$

- Subtração

$$\begin{array}{r}0\\-0\\\hline 0\end{array}\qquad\begin{array}{r}1\\-0\\\hline 1\end{array}\qquad\begin{array}{r}0\\-1\\\hline 11\end{array}\qquad\begin{array}{r}1\\-1\\\hline 0\end{array}\qquad\begin{array}{rcccccccc} & & 0 & 1 & 1 & 0 & 1 & 0 & 1\\ - & & 1 & 1 & 0 & 1 & 1 & 1 & 0\\\hline & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1\end{array}$$

Códigos

Ponderados

- Existem pesos associados

- **BCD - *Binary Coded Decimal***
  - Representa os dígitos "0" a "9" através de representações binárias com 4 bits
  - Não são utilizadas as representações binárias > "9" [1010, 1111]
  - Conversões de BCD são efetuadas diretamente 4 bits por cada digito decimal

Decimal	BCD (8421)	BCD (2421)	BCD (442-1)	BCD Excesso 3
0	0000	0000	0000	0011
1	0001	0001	0011	0100
2	0010	0010/1000	0010	0101
3	0011	1001/0011	1001/0101	0110
4	0100	1010/0100	1000/0100	0111
5	0101	1011/0101	1011/0111	1000
6	0110	1100/0110	0110/1010	1001
7	0111	1101/0111	1101	1010
8	1000	1110	1100	1011
9	1001	1111	1111	1100

- Exemplos:
    - 1101 = 1x8 + 4x1 + 1x1 = 13
    - 1001 = 9

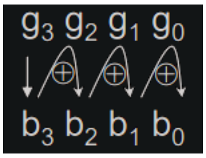
Não Ponderados

- **EBCDIL**
- **ASCII - American Standard Code for Information Intercharge**
    - Código que transforma os caracteres em números decimais e vice-versa

- **Gray** (código espelhado)

BCD	GRAY
00 - 0000	00 - 0000
01 - 0001	01 - 0001
02 - 0010	02 - 0011
03 - 0011	03 - 0010
04 - 0100	04 - 0110
05 - 0101	05 - 0111
06 - 0110	06 - 0101
07 - 0111	07 - 0100
08 - 1000	08 - 1100
09 - 1001	09 - 1101
10 - 1010	10 - 1111
11 - 1011	11 - 1110
12 - 1100	12 - 1010
13 - 1101	13 - 1011
14 - 1110	14 - 1001
15 - 1111	15 - 1000

Conversão de gray para binário



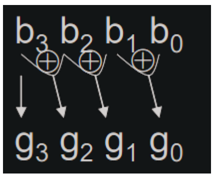
⊕ - ou exclusivo - XOR

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

- Exemplos:
  - 1101 → ?
    - g3 - 1 ; g2 - 1 ; g1 - 0 ; g0 - 1
    - b3 - 1
    - b2 - b3 ⊕ g2 → 1 ⊕ 1 = 0
    - b1 - b2 ⊕ g1 → 0 ⊕ 0 = 0
    - b0 - b1 ⊕ g0 → 0 ⊕ 1 = 1
  - 1101 → 1001

  - 1110 → ?
    - g3 - 1 ; g2 - 1 ; g1 - 1 ; g0 - 0
    - b3 - 1
    - b2 - b3 ⊕ g_2 → 1 ⊕ 1 = 0
    - b1 - b2 ⊕ g1 → 0 ⊕ 1 = 1
    - b0 - b1 ⊕ g0 → 1 ⊕ 0 = 1
  - 1110 → 1011

Conversão de binário para gray



- Exemplos:
  - 1010 → ?
    - b3 - 1 ; b2 - 0 ; b1 - 1 ; b0 - 0
    - g3 - 1
    - g2 - b3 ⊕ b2 → 1 ⊕ 0 = 1
    - g1 - b2 ⊕ b1 → 0 ⊕ 1 = 1
    - g0 - b1 ⊕ b0 → 1 ⊕ 0 = 1
  - 1010 → 1111

  - 0011 → ?
    - g3 - 0 ; g2 - 0 ; g1 - 0 ; g0 - 1

- $g3$ - 0
- $g2$ - $b3 \oplus b2 \rightarrow 0 \oplus 0 = 0$
- $g1$ - $b2 \oplus b1 \rightarrow 0 \oplus 1 = 1$
- $g0$ - $b1 \oplus b0 \rightarrow 1 \oplus 1 = 0$

- 0011 → 0010

Álgebra das Variáveis Lógicas