

Lógica e Sistemas Digitais

Códigos numéricos

João Pedro Patriarca (jpatri@cc.isel.ipl.pt)

Slides inspirados nos slides do prof. Mário Véstias

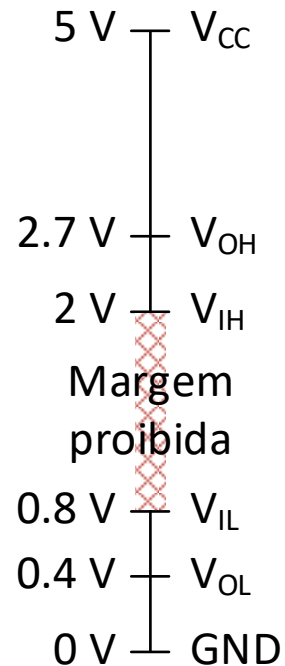


Sistema digital

- Um sistema digital trabalha apenas com dois valores binários:
 - Dígitos 0 e 1 ou nível lógico 0 e nível lógico 1
 - Falso (F) \Rightarrow 0 e Verdadeiro (V) \Rightarrow 1
 - Baixo (L – *Low*) \Rightarrow 0 e Alto (H – *High*) \Rightarrow 1
- Tradução de nível de tensão em nível lógico/binário
- Necessário um sistema de numeração e de uma álgebra que considere apenas dois símbolos: 0 e 1

Tradução de nível de tensão em nível lógico

- Um circuito digital precisa ser alimentado
 - Os níveis de tensão de referência são: 3.3V ou 5V
 - Tecnologia TTL (*Transistor-Transistor Logic*)
- Exemplo de tradução de nível de tensão 5V TTL para nível lógico:



- V_{OH} – Nível de tensão mínima que um dispositivo TTL providencia à saída (O de *output*) para apresentar o nível lógico 1
- V_{OL} – Nível de tensão máxima que um dispositivo TTL providencia à saída para apresentar o nível lógico 0
- V_{IH} – Nível de tensão mínima que um dispositivo TTL permite ter à entrada (I de *input*) para considerar o nível lógico 1
- V_{IL} – Nível de tensão máxima que um dispositivo TTL permite ter à entrada para considerar o nível lógico 0
- Leitura complementar: <https://learn.sparkfun.com/tutorials/logic-levels/all>

Sistema de numeração decimal

- Sistema decimal (base 10) utiliza 10 símbolos: 0 a 9
- Sistema posicional: cada símbolo do número possui um peso (potência de 10)

$$123_{10} = 1 * 10^2 + 2 * 10^1 + 3 * 10^0$$

- Para qualquer base, o valor correspondente na base 10 é obtido através da Expressão de significância posicional
- Para um número N na base b

$$N_{(b)} = \underbrace{d_{k-1}d_{k-2} \dots d_1d_0}_{\text{componente inteira}}, \quad \underbrace{d_{-1} \dots d_{-m}}_{\text{componente decimal}}$$

o valor decimal correspondente é dado por

$$N'_{(10)} = \sum_{i=-m}^{k-1} d_i b^i = \underbrace{d_{k-1}b^{k-1} + \dots + d_0b^0}_{\text{componente inteira}} + \underbrace{d_{-1}b^{-1} \dots d_{-m}b^{-m}}_{\text{componente decimal}}$$

Exemplos de conversões para base 10

- Base 4 (símbolos entre 0 e 3)

$$123_{(4)} = 1 * 4^2 + 2 * 4^1 + 3 * 4^0 = 16 + 8 + 3 = 27_{(10)}$$

- Base 8 (símbolos entre 0 e 7)

$$123_{(8)} = 1 * 8^2 + 2 * 8^1 + 3 * 8^0 = 64 + 16 + 3 = 83_{(10)}$$

- Base 16 (símbolos entre 0 e 9 e A e F)

$$\begin{aligned} 123AD_{(16)} &= 1 * 16^4 + 2 * 16^3 + 3 * 16^2 + 10 * 16^1 + 13 * 16^0 \\ &= 65536 + 8192 + 768 + 160 + 13 = 74669_{(10)} \end{aligned}$$

Sistema binário – base 2

- Utiliza apenas dois símbolos: 0 e 1

$$\begin{aligned}1101_{(2)} &= 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 \\ &= 8 + 4 + 1 = 13_{(10)}\end{aligned}$$

- Potências de 2 (tabela ao lado)
- Exemplos:

$$2^{12} = 2^2 * 2^{10} = 4 * 1 \text{ k} = 4 \text{ k}$$

$$2^{16} = 2^6 * 2^{10} = 64 * 1 \text{ k} = 64 \text{ k}$$

$$2^{28} = 2^8 * 2^{20} = 256 * 1 \text{ M} = 256 \text{ M}$$

n	2 ⁿ
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024 = 1 k
16	65536 = 64 k
20	1048576 = 1 M = 1 k * 1 k

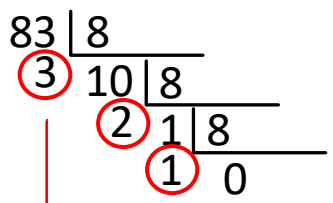
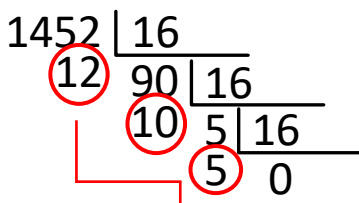
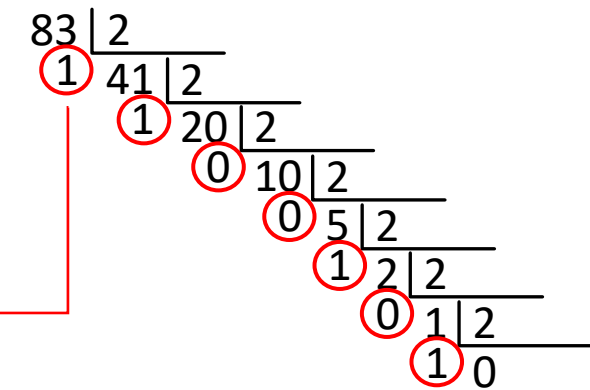
Contar em qualquer base

- Tal como na base 10, adiciona-se 1 à posição seguinte sempre que forem esgotados os símbolos

Binária	Octal	Decimal	Hexadecimal
00000	0	0	0
00001	1	1	1
00010	2	2	2
00011	3	3	3
00100	4	4	4
00101	5	5	5
00110	6	6	6
00111	7	7	7
01000	10	8	8
01001	11	9	9
01010	12	10	A
01011	13	11	B
01100	14	12	C
01101	15	13	D
01110	16	14	E
01111	17	15	F
10000	20	16	10

Conversão da base 10 para outra base

- Divide-se sucessivamente pela base pretendida até o valor do quociente dar 0. O resultado da conversão é dado pelos restos sucessivos

$83_{(10)} \rightarrow \text{base } 8$	$1452_{(10)} \rightarrow \text{base } 16$	$83_{(10)} \rightarrow \text{base } 2$
		
$83_{(10)} \rightarrow 123_{(8)}$	$1452_{(10)} \rightarrow 5AC_{(16)}$	$83_{(10)} \rightarrow 1010011_{(2)}$

Conversão entre bases não decimais

- Regra geral, utiliza-se a base decimal como base auxiliar
 1. Converte-se base origem para base 10
 2. Converte-se base 10 para base destino
- Com bases potências de 2 (2, 4, 8, 16, etc.) o processo é simplificado
- Por exemplo base 2 e 16
 - $2^4 = 16$, logo 4 dígitos da base 2 constituem 1 dígito da base 16 e vice-versa
 - As combinações de 4 dígitos na base 2 esgotam para representar os 16 dígitos da base 16

0101 1010 1100₍₂₎
 5 A C₍₁₆₎

6D₍₁₆₎
0110 1101₍₂₎

Códigos binários

- Representação de valores expressos na base 2
 - Números naturais
 - Códigos dos complementos
 - Código BCD (*Binary Coded Decimal*)
 - Representação em binário apenas dos dígitos de 0 ($0000_{(BCD)}$) a 9 ($1001_{(BCD)}$)
 - Conversão simples e direta
 - Exemplo: $2039_{(10)} = 0010\ 0000\ 0011\ 1001_{(BCD)}$
- Representação de valores não numéricos
 - Código ASCII standard: código de 7 bits usado para codificar letras, símbolos de pontuação e de controlo
 - Códigos UNICODE: UTF-8, UTF-16, UTF-32

Exercícios com conversões entre bases

1. $125_{(6)} \Rightarrow ???_{(4)}$
2. $1075_{(8)} \Rightarrow ???_{(2)}$
3. $58E_{(16)} \Rightarrow ???_{(2)}$
4. $101001011_{(2)} \Rightarrow ???_{(16)}$
5. $B2DF_{(16)} \Rightarrow ???_{(8)}$

Resultados finais dos exercícios com conversões entre bases

1. $125_{(6)} \Rightarrow 311_{(4)}$
2. $1075_{(8)} \Rightarrow 1000111101_{(2)}$
3. $58E_{(16)} \Rightarrow 10110001110_{(2)}$
4. $101001011_{(2)} \Rightarrow 14B_{(16)}$
5. $B2DF_{(16)} \Rightarrow 131337_{(8)}$