

“Os humanos são pensadores lentos, desleixados e brilhantes.  
Computadores são rápidos, cuidadosos e estúpidos” (John Pfeiffer).

# Bytes e bases Octal/Hexadecimal

Paulo Ricardo Lisboa de Almeida

# Binário

A máquina opera na base 2

## **Base binária**

Temos somente 2 algarismos possíveis: 0 e 1

Os “dígitos” binários são chamados de bits

**bit:** binary digit

# Byte

A um **conjunto de 8 bits**, damos o nome de **byte**

Na maioria (não todas) das CPUs, a menor unidade em que podemos operar é um byte

Exemplo: no seu computador x86-64, a memória é endereçada a byte

Cada endereço físico da memória suporta 1 byte

# Byte

A um **conjunto de 8 bits**, damos o nome de **byte**

Na maioria (não todas) das CPUs, a menor unidade em que podemos operar é um byte

Exemplo: no seu computador x86-64, a memória é endereçada a byte

Cada endereço físico da memória suporta 1 byte

# Nibble

Um **nibble** é o equivalente a **meio byte**

4 bits

É especialmente útil quando estamos lidando com valores em **hexadecimal**



# Exemplo

$$101_{10} = 0110\ 0101_2$$

Nibble Nibble

Byte

# Prefixos

Da mesma forma que no SI, podemos adicionar prefixos

kilo, mega, giga, ...

Exemplo, um 1kB é um kilobyte, e corresponde a 1.000 Bytes

# Prefixos

Da mesma forma que no SI, podemos adicionar prefixos

kilo, mega, giga, ...

Exemplo, um 1kB é um kilobyte, e corresponde a 1.000 Bytes

Mas durante o curso, você vai aprender que se lidarmos com potências de 2, as coisas são mais simples

Então idealmente 1 kB deve ter  $2^{10} = 1024$  Bytes

E 1 MB deve ter  $2^{10} = 1024$  Kilobytes =  $2^{20}$  Bytes = 1048576 Bytes

...



# Confusão

Agora quando nos referenciamos a 1 KB, estamos falando de 1.000 Bytes, ou 1.024 Bytes?

Dica: na computação, **geralmente nos referenciar a potência de 2**

Para evitar confusão, a International Electrotechnical Commission (IEC) criou a seguinte nomenclatura

Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	$10^3$	kibibyte	KiB	$2^{10}$	2%
megabyte	MB	$10^6$	mebibyte	MiB	$2^{20}$	5%
gigabyte	GB	$10^9$	gibibyte	GiB	$2^{30}$	7%
terabyte	TB	$10^{12}$	tebibyte	TiB	$2^{40}$	10%
petabyte	PB	$10^{15}$	pebibyte	PiB	$2^{50}$	13%
exabyte	EB	$10^{18}$	exbibyte	EiB	$2^{60}$	15%
zettabyte	ZB	$10^{21}$	zebibyte	ZiB	$2^{70}$	18%
yottabyte	YB	$10^{24}$	yobibyte	YiB	$2^{80}$	21%

Patterson, D.; Hennessy, 2017.

# Mais confusão

Um kB (com B maiúsculo) é um kilobyte, enquanto um kb ou kbit (com b minúsculo) é um kilobit

# Octal e Hexadecimal

Na computação é comum o uso das bases Octal (base 8) e Hexadecimal (base 16)

**São potências de 2**

Facilita a conversão para binário

No sistema octal, os numerais válidos são ...?

# Octal e Hexadecimal

Na computação é comum o uso das bases Octal (base 8) e Hexadecimal (base 16)

**São potências de 2**

Facilita a conversão para binário

Na base octal, os numerais válidos são 0, 1, 2, 3, ...,7.

Na base hexadecimal, os numerais válidos são ...?

# Octal e Hexadecimal

Na computação é comum o uso das bases Octal (base 8) e Hexadecimal (base 16)

**São potências de 2**

Facilita a conversão para binário

Na base octal, os numerais válidos são 0, 1, 2, 3, ...,7.

Na base hexadecimal, os numerais válidos são 0,1,2,...,9,A,B,C,D,E,F.

# Conversão

A conversão octal/hexadecimal para binário, e vice-versa, é direta

# Conversão

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Para converter de binário para octal

Seperare os algarismos em grupos de 3 do bit menos significativo para o mais significativo

Um conjunto de **3 bits é um octeto**

Utilizar a tabela para obter o valor de cada octeto, e esse será o valor em octal



# Conversão

Exemplo: Converter  $110010011101_2$  para octal

Desconsidere esse primeiro zero da tabela quando converter octal.

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Exemplo: Converter  $110010011101_2$  para octal

$$110\ 010\ 011\ 101_2 = 6235_8$$

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Para converter entre octal para binário basta fazer o inverso

Convertemos cada algarismo octal para seus 3 dígitos binários equivalentes

# Conversão

Converter  $1366_8$  para binário

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Converter  $1366_8$  para binário

$$1366_8 = 001\ 011\ 110\ 110_2$$

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Para a conversão entre hexadecimal e binário, vice-versa, usamos o mesmo procedimento utilizado para octal

Agora separe em **grupos de 4 bits**

**Hextetos** ou Nibbles

# Conversão

Converter  $110010011101_2$  para hexadecimal

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

# Conversão

Converter  $110010011101_2$  para hexadecimal

$$1100\ 1001\ 1101_2 = C9D_{16}$$

Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	-	8
9	1001	-	9
10	1010	-	10
11	1011	-	11
12	1100	-	12
13	1101	-	13
14	1110	-	14
15	1111	-	15

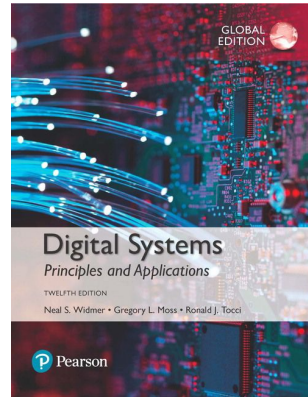


# Exercícios

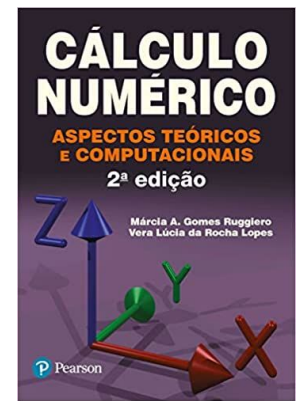
- A. Converta os seguintes valores:
- a.  $1111111111_2$  para octal e hexadecimal.
  - b.  $111011_2$  para octal, hexadecimal e decimal
  - c. 628A16 para binário
  - d.  $765_8$  para binário e hexadecimal

# Referências

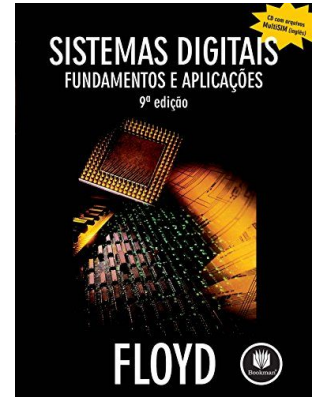
Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Marcia A. G. Ruggiero, Vera L. R. Lopes. Cálculo numérico aspectos teóricos e computacionais. 1996.



Thomas Floyd. Widmer. Sistemas Digitais: Fundamentos e Aplicações. 2009.



# Licença

Este obra está licenciado com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

