TÉCNICO LISBOA

Sistemas Digitais (SD)

Sistemas de Numeração e Códigos

Aula Anterior

- **Na aula anterior:**
  - Motivação:
    - O que é um Sistema Digital?
    - Onde estão os Circuitos Digitais?
    - Perspectiva histórica:
      - Dos primórdios da história até aos computadores de hoje
    - De que é feito um computador?
  - Sistemas Digitais:
    - Programa da cadeira
    - Organização
    - Corpo docente
    - Planeamento
    - Método de Avaliação
    - Aulas Teóricas, Problemas e de Laboratório
    - Bibliografia

Planeamento

SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
17/Fev a 21/Fev	Introdução	Sistemas de Numeração	
24/Fev a 28/Fev	CARNAVAL	Álgebra de Boole	P0
02/Mar a 06/Mar	Elementos de Tecnologia	Funções Lógicas	VHDL
9/Mar a 13/Mar	Minimização de Funções	Minimização de Funções	L0
16/Mar a 20/Mar	Def. Circuito Combinatório; Análise Temporal	Circuitos Combinatórios	P1
23/Mar a 27/Mar	Circuitos Combinatórios	Circuitos Combinatórios	**L1**
30/Mar a 03/Abr	Circuitos Sequenciais: Latches	Circuitos Sequenciais: Flip-Flops	P2
06/Abr a 10/Abr	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA
13/Abr a 17/Abr	Caracterização Temporal	Registos	L2
20/Abr a 24/Abr	Contadores	Circuitos Sequenciais Síncronos	P3
27/Abr a 01/Mai	Síntese de Circuitos Sequenciais Síncronos	Síntese de Circuitos Sequenciais Síncronos	L3
04/Mai a 08/Mai	Exercícios **Teste 1**	Memórias	P4
11/Mai a 15/Mai	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Microprograma	L4
18/Mai a 22/Mai	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	P5
25/Mai a 29/Mai	P6	P6	L5

Sumário

- **Tema da aula de hoje:**
  - Sistemas de numeração
    - Base 10
    - Base 2
    - Base 8 e 16
  - Operações aritméticas básicas
  - Mudança de sistema de numeração
  - Códigos

- **Bibliografia:**
  - **M. Mano, C. Kime**: Capítulo 1
  - **G. Arroz, J. Monteiro, A. Oliveira**: Capítulo 1

Definição de um Sistema de Numeração Posicional

- **Um sistema de numeração é composto por:**
  - **Base** - *b*

    e.g. Base = 16
  - **Alfabeto Ordenado** - conjunto de *b* símbolos distintos (dígitos)

    e.g. [0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]
  - **Número** - representado por uma sequência de dígitos

    e.g. $N(b) <> \ldots d_2\ d_1\ d_0,\ d_{-1}\ d_{-2} \ldots$
  - **Valor do Dígito** - função do símbolo e da posição na sequência (peso).

    e.g. $v_2 = d_2\ b^2$

<u>Exemplos</u>:

S.N. :	Decimal	Binário	Octal	Hexadecimal
	28886_{10}	10101110_2	5270_8	$A32C_{16}$

Sistemas de numeração

- Exemplos com várias bases

Base 10	0	1	2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	17	18	19

Base 4	0	1	2	3	10	11	12	13	20	21
	22	23	30	31	32	33	100	101	102	103

Base 3	0	1	2	10	11	12	20	21	22	100
	101	102	110	111	112	120	121	122	200	201

Base 2	0	1	10	11	100	101	110	111	1000	1001
	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011

Base 16	0	1	2	3	4	5	6	7	8	9
	A	B	C	D	E	F	10	11	12	13

Determinação do Equivalente Decimal

- **Equivalente Decimal** - Representação no sistema decimal de um número na base b.

$$N_{(10)} = \sum_{-\infty}^{+\infty} d_i b^i = \ldots + d_2 b^2 + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + \ldots$$

Exemplos:

- **S.N.: Binário** **Decimal**

$10101110_2 \longrightarrow (2^7 + 0 + 2^5 + 0 + 2^3 + 2^2 + 2^1 + 0)_{10} \longrightarrow 174_{10}$

- **S.N.: Hexadecimal** **Decimal**

$A32C_{16} \longrightarrow (10x16^3 + 3x16^2 + 2x16^1 + 12x16^0)_{10} \longrightarrow 41772_{10}$

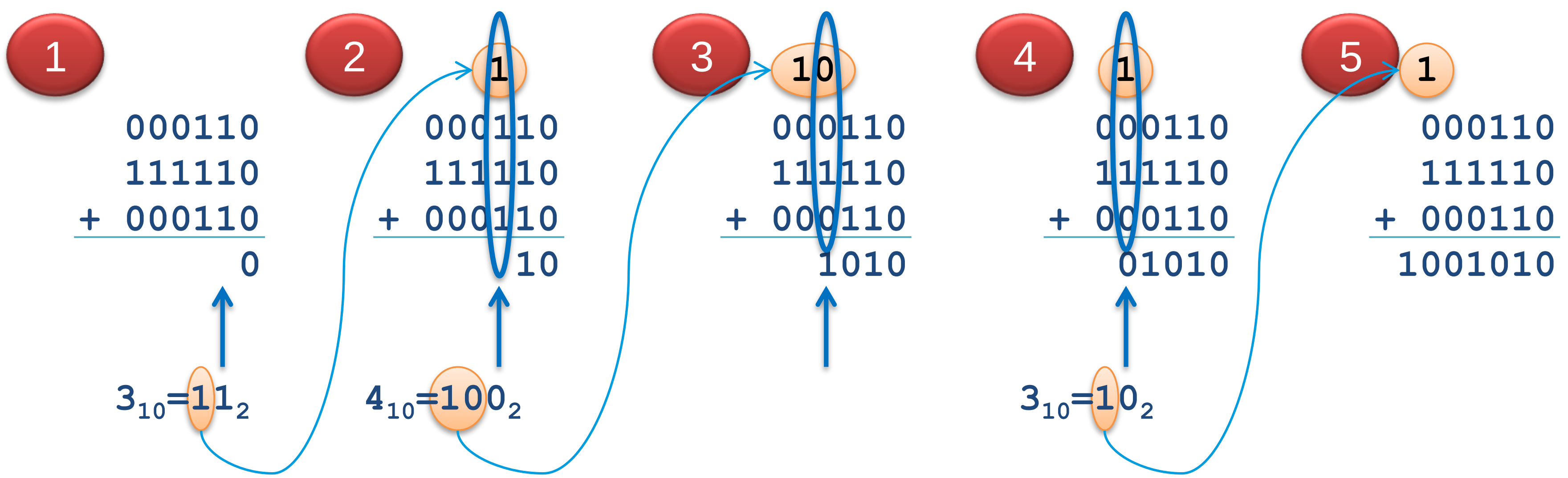
TÉCNICO LISBOA

Operações Aritméticas Básicas

- Algoritmos em tudo semelhantes ao do sistema decimal, excepto na base utilizada.

Exemplo:

$$110_2 + 111110_2 + 110_2$$



Operações Aritméticas Básicas

- Algoritmos em tudo semelhantes ao do sistema decimal, excepto na base utilizada.

Exemplos:

S.N. : Binário

```
  0110
+ 1101
 10011
```

```
    10110
x    1101
    10110
   00000_
  10110__
+ 10110___
100011110
```

S.N. : Hexadecimal

```
  5AF1
+ B32D
 10E1E
```

```
  A24
x  13
 1E6C
+ A24_
 C0AC
```

Mudança de Sistema de Numeração

CONVERSÃO DE BASES ($b_1 \neq 10$ para $b_2 = 10$)

- A conversão de um número numa base diferente de 10 para a base decimal reduz-se a representar esse número como um polinómio e de seguida determinar o equivalente decimal:

$$N_{(10)} = \sum_{-\infty}^{+\infty} d_i b^i = \ldots + d_2 b^2 + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + \ldots$$

Exemplos:

$10101110_2 \longrightarrow (2^7 + 0 + 2^5 + 0 + 2^3 + 2^2 + 2^1 + 0)_{10} \longrightarrow 174_{10}$

$A32C_{16} \longrightarrow (10x16^3 + 3x16^2 + 2x16^1 + 12x16^0)_{10} \longrightarrow 41772_{10}$

Mudança de Sistema de Numeração

- **CONVERSÃO DE BASES (b_1=10 para $b_2 \neq$ 10)**
  - A conversão de um número na base 10 para uma base diferente realiza-se em duas fases:
    1. A parte inteira é convertida segundo o método das divisões sucessivas.
    2. A parte fraccionária é convertida segundo o método das multiplicações sucessivas.

Mudança de Sistema de Numeração

■ CONVERSÃO DE BASES (b_1=10 para $b_2 \neq$ 10)

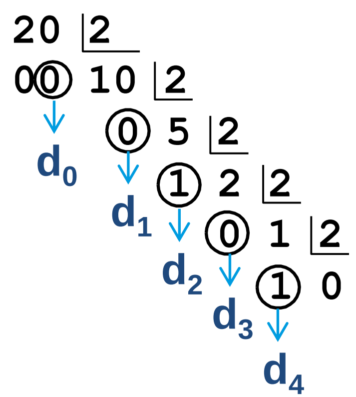
<u>Exemplo (parte inteira)</u>:

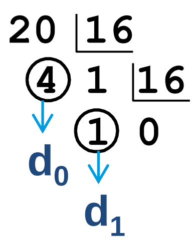
S.N. :	Decimal		Binário		Hexadecimal
	$20{,}35_{(10)}$	→	$10100{,}\ldots_{(2)}$	→	$14{,}\ldots_{(16)}$

O número a converter e os quocientes sucessivos são divididos pela base.

A sequência de restos constitui o resultado da conversão.

1º resto = dígito menos significativo





Mudança de Sistema de Numeração

CONVERSÃO DE BASES (b_1=10 para $b_2 \neq$ 10)

Exemplo (parte fraccionária):

S.N. : Decimal		Binário		Hexadecimal
$20{,}35_{(10)}$	→	$10100{,}\ldots_{(2)}$	→	$14{,}\ldots_{(16)}$

Binário:

```
  0,35
   x 2
  ----
  0,70
  0,70
   x 2
  ----
  1,40
  0,40
   x 2
  ----
  0,80
  0,80
   x 2
  ----
  1,60
```

$10100{,}0101\ldots_{(2)}$

Hexadecimal:

```
  0,35
  x 16
  ----
  5,60
  0,60
  x 16
  ----
  9,60
  0,60
  x 16
  ----
  9,60
  ...
```

$14{,}599\ldots_{(16)}$

Mudança de Sistema de Numeração

CONVERSÃO DE BASES ($b_t = 2^t$ para $b = 2$)

- Atendendo às propriedades das potências, facilmente se infere que:

1. Na conversão da base 2^t para a base 2, transforma-se cada dígito da base 2^t em t bits da base 2.
2. Na conversão da base 2 para a base 2^t, transforma-se cada t bits da base 2 num dígito da base 2^t.

Exemplos:

Binário: $\overbrace{0001}^{4}\ \overbrace{0100}^{4},\overbrace{0101}^{4}{}_{(2)}$

$\updownarrow$ $\updownarrow$ $\updownarrow$

Hexadecimal: $1\quad 4\quad ,\quad 5\ {}_{(16)}$

Binário: $\overbrace{000}^{3}\ \overbrace{010}^{3}\ \overbrace{100}^{3},\overbrace{010}^{3}\ \overbrace{100}^{3}\ {}_{(2)}$

$\updownarrow$ $\updownarrow$ $\updownarrow$ $\updownarrow$ $\updownarrow$

Octal: $0\quad 2\quad 4\quad ,\quad 2\quad 4\ {}_{(8)}$

Códigos Binários

Entende-se por **código binário** uma correspondência entre palavras escritas num qualquer sistema de numeração e palavras constituídas por caracteres binários.

Exemplo: $12_{(10)}$ <> $1100_{(2)}$

CÓDIGO BINÁRIO NATURAL (CBN)

- Código ponderado, gerado pelo sistema de numeração de base 2, em que os pesos das colunas são sucessivamente 2^{n-1}, 2^{n-2}, …, 2^1, 2^0.

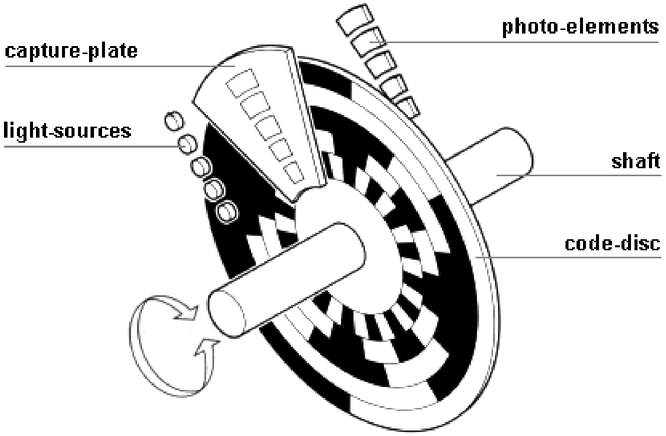
CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY

- Código <u>não ponderado</u>, obtido do CBN por troca de símbolos do alfabeto binário;
- Apresenta como característica fundamental o facto de dois símbolos que representam números consecutivos terem apenas um bit diferente.

$$
\begin{array}{ccccc}
 & & \text{CBN} & & \text{CBR} \\
\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} & \Leftrightarrow & \begin{bmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{bmatrix} & & \begin{bmatrix} 000 \\ 001 \\ 011 \\ 010 \\ 110 \\ 111 \\ 101 \\ 100 \end{bmatrix}
\end{array}
$$

Códigos Binários

- **CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY**
  - Motivação:



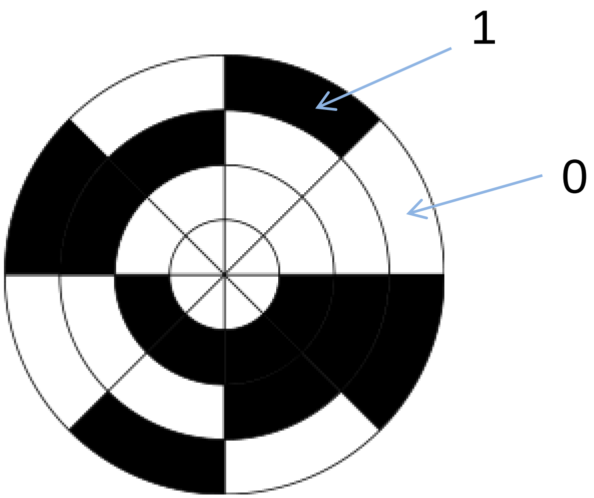
    - Muitos dispositivos indicam a sua posição através da abertura e fecho de interruptores.
    - Se a posição desses interruptores for codificada em código binário natural, as seguintes duas posições serão adjacentes: 011 → 100
    - Na prática, é muito difícil garantir que os interruptores comutem exactamente ao mesmo tempo. Neste exemplo, em particular, os 3 interruptores trocam de estado. Como consequência, durante o intervalo de tempo em que eles estão a trocar de estado poderão surgir estados transitórios. Exemplo:

      011 — 001 — 101 — 100.
    - Quando os interruptores aparentam estar na posição 001, o observador não sabe se este é o estado definitivo (001) ou apenas uma transição entre outros dois estados, dando assim origem a leituras incorrectas.
  - Solução:
    - **Código de Gray**: 2 símbolos que representam números consecutivos diferem apenas 1 bit.

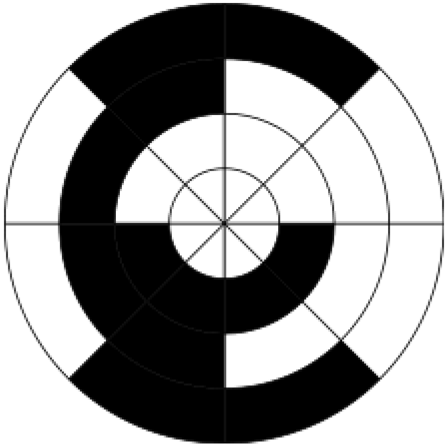
Códigos Binários

- **CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY**

  Aplicação: encoder de posição



Código Binário Natural (CBN)



Código Binário Reflectido (CBR)

Os códigos de duas posições adjacentes diferem apenas num bit

Códigos Binários

CÓDIGO BINÁRIO REFLECTIDO (CBR) ou CÓDIGO DE GRAY

Construção:

Código de Gray

a	b	c	d	e	decimal
0	0	00	00	000	0
1	1	01	01	001	1
	1	11	11	011	2
	0	10	10	010	3
			10	110	4
			11	111	5
			01	101	6
			00	100	7

a- Código gray 1 bit
b- Reflexão do código
c- Adicionar 0's e 1's = código gray 2 bits
d- Reflexão do código anterior
e- Adicionar os 0's e 1's código gray 3 bits

Nota: as colunas b e d não fazem parte do código de gray

Códigos Decimais-Binários

Entende-se por **código decimal-binário** um código que estabelece a correspondência directa entre caracteres da palavra constituída por símbolos da base 10 e a sua codificação binária.

- **CÓDIGO BCD (“Binary-Coded Decimal”)**
  - O código BCD corresponde ao CBN com N=4.

    Exemplo: $12_{(10)}$ <> $0001\ \ 0010_{\ (BCD)}$

Nota: Nas operações aritméticas deve ser introduzido um factor de correcção, $\mathbf{6_{(10)}}$ **<>** $\mathbf{0110_{(BCD)}}$, sempre que o resultado seja superior ou igual a 10.

BCD

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0000 \\ 0001 \\ 0010 \\ 0011 \\ 0100 \\ 0101 \\ 0110 \\ 0111 \\ 1000 \\ 1001 \end{bmatrix}$$

Códigos Decimais-Binários

Exemplo: operação de soma

599_{10}
$+984_{10}$

BCD →

	0101	1001	1001	
	+ 1001	1000	0100	
	1110	10001	1101	***Dígitos d_0,d_1,d_2 inválidos***
	(14_{10})	(17_{10})	(13_{10})	
	1110	10001	1101	Correção ao dígito d_0
			+0110	
	1110	10001	10011	***Dígitos d_1,d_2 inválidos***
	(14_{10})	(18_{10})	(3_{10})	
		+1		
	+1	+0110		Correção ao dígito d_1
	1111	1000	0011	***Dígito d_2 inválido***
	(15_{10})	(8_{10})	(3_{10})	
	1111			
	+0110			Correção ao dígito d_1
0001	0101	1000	0011	***Todos os dígitos válidos***
(1_{10})	(5_{10})	(8_{10})	(3_{10})	

A **operação de soma** em BCD é semelhante à soma em binário. A diferença consiste no seguinte:

- Sempre que o resultado da soma originar um dígito não válido em decimal (valor >9), deve-se somar 6 ao resultado.

Códigos Alfanuméricos

- **CÓDIGO ASCII (American Standard Code for Information InterChange):**
  - Exemplo de código alfanumérico que permite codificar informação numérica, alfabética e também caracteres de controlo.

$B_3\,B_2\,B_1\,B_0$

$B_6\,B_5\,B_4$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Representação numérica: 46_{16}

Exemplo: "Margarida" equivale à sequência de números

$4D_{16}$, 61_{16} , 72_{16} , 67_{16} , 61_{16} , 72_{16} , 69_{16} , 64_{16} , 61_{16}

Representação Digital da Informação

- É habitual organizar os bits em unidades de maior capacidade
  - Exemplos:
    - tabela ASCII com 127 símbolos → 7 bits
    - Tabela ISO-8859-1 com 256 símbolos (inclui acentos) → 8 bits
    - Representação RGB da cor de um pixel → 24 bits

- Em geral, a informação é processada, transferida e armazenada em unidades de 8-bits: ***byte*** (ou octeto)

- Em algumas aplicações (ex: codificação BCD), é usual utilizar unidades de 4-bits: ***nibble***

Como é natural, ***2 nibbles = 1 byte***

Representação Digital da Informação

Representação Digital da Informação

- Quando se consideram grandes quantidades de informação, é usual utilizar múltiplos da unidade:

Múltiplo	Potência	Relação com o múltiplo inferior	Representação na base 10	Denominação
1	2^0		1	
1K	2^{10}	$= 2^{10}$	1024	Quilo
1M	2^{20}	$= 2^{10}$ K	1 048 576	Mega
1G	2^{30}	$= 2^{10}$ M	1 073 741 824	Giga
1T	2^{40}	$= 2^{10}$ G	1 099 511 627 776	Tera

<u>Exemplo</u>: um ficheiro ocupa 2,37MB

2,37MBytes = 2,37 x 2^{20} = 2,37 x 1024 x 1024 = **2 485 125 bytes**

Conceito de palavra (*word*)

- Unidade mínima processada ou armazenada num dado sistema.
  - Exemplos:

Intel 4004	4 bits	Intel 486	32 bits
Intel 8080	8 bits	Intel Pentium	32 bits
Motorola 6800	8 bits	ARM Cortex A-9	32 bits
Intel 8086	16 bits	Intel Core 2 i7	64 bits
Motorola 68000	16 bits	Cell (STI)	128 bits

- Ao contrário do conceito de ***byte*** e ***nibble***, o conceito de palavra não está ligado a uma dimensão fixa. O número de bits de uma palavra depende do contexto que se está a considerar.

PRÓXIMA AULA

Próxima Aula

- **Tema da Próxima Aula:**
  - Álgebra de Boole
    - Operações básicas
    - Propriedades
    - Portas Lógicas
  - Leis de Morgan
    - Simplificação algébrica

Agradecimentos

Algumas páginas desta apresentação resultam da compilação de várias contribuições produzidas por:

- Nuno Roma
- Guilherme Arroz
- Horácio Neto
- Nuno Horta
- Pedro Tomás