



Introdução aos Códigos Binários

Augusto Silva, Iouliia Sklyarova

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro



Conceito de Código Binário

- **Código** - conjunto de sequências de n bits, em que cada sequência representa um determinado valor, evento, acção, mensagem, etc
- **Palavra do código** - uma dada sequência de n bit
- n - comprimento do código
- m - número de valores a codificar

$$n \geq \lceil \log_2(m) \rceil$$



Exemplo

- Várias estratégias possíveis desde

$$n \geq \lceil \log_2(7) \rceil = 3$$



andar	codificação	codificação	codificação
cave	000	000	000001
r/c	001	001	000010
1º andar	010	011	000100
2º andar	011	010	001000
3º andar	100	110	010000
4º andar	101	111	100000



Códigos Binary Coded Decimal (BCD)

- Representação alternativa de quantidades
- Cada algarismo decimal está associado a uma palavra de código
- Usam-se as primeiras 10 palavras da representação em código binário natural das quantidades 0,1,2,...,9
- Comprimento da palavra L é o menor inteiro que satisfaz

$$L \geq \log_2 N$$

$$N = 10$$

$$L = 4 \text{ bits}$$



BCD 8421

• Propriedades

- Código regular: comprimento de palavra é fixo = 4

- Código ponderado: cada uma das palavras resulta duma elaboração analítica da forma

$$x = \sum_{i=0}^3 a_i w_i \quad w_i = 2^i, \quad a_i \in \{0,1\}$$

- Código descontínuo: palavras consecutivas diferem em mais de um dígito ("bit")
- Não autocomplementar: existem palavras para as quais é violada a condição de autocomplementaridade i.e

$$x = (a_3 a_2 a_1 a_0), \quad (9 - x) \neq (\bar{a}_3 \bar{a}_2 \bar{a}_1 \bar{a}_0)$$

- Código não cíclico: 1ª e última palavra não são adjacentes

	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



BCD 8421

• Propriedades

- Não distingue palavras abaixo e acima de 5 com o "msb"

- Distância entre 2 palavras de código ou distância de Hamming

- nº de posições que diferem entre duas palavras adjacentes. Neste caso $d_{min} = 1$

- Redundância R:

- Sendo N o número de palavras do código e L o comprimento de palavra

$$R = \frac{L - \log_2 N}{L}$$

$$R_{BCD8421} = \frac{4 - \log_2 10}{4} = 0.17$$

	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



Outros Códigos BCD

- Exercícios:
- Gere códigos BCD ponderados correspondentes aos seguintes conjuntos de pesos
 - 5 2 1 1
 - 6 4 2 -3
 - 5 0 4 3 2 1 0



Códigos BCD autocomplementares

- Num código decimal binário autocomplementar ponderado o somatório dos seus pesos deverá ser 9
- Dem:
 - Sejam N_1 e N_2 duas palavras complementares dum código decimal binário ponderado de comprimento n tal que

$$N_1 = \sum_{i=0}^{n-1} a_i w_i \quad a_i \in \{0,1\}$$

$$N_2 = \sum_{i=0}^{n-1} \bar{a}_i w_i$$

- Se o código BCD é autocomplementar então

$$9 - N_1 = N_2 \therefore N_1 + N_2 = 9 \therefore \sum_{i=0}^{n-1} (a_i + \bar{a}_i) w_i \therefore \sum_{i=0}^{n-1} w_i = 9$$



Exemplos

	5	2	1	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1

	4	2	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	1	0
5	1	0	0	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

	6	4	2	-3
0	0	0	0	0
1	0	1	0	1
2	0	0	1	0
3	0	1	1	1
4	0	1	0	0
5	1	0	1	1
6	1	0	0	0
7	1	1	0	1
8	1	0	1	0
9	1	1	1	1

	8	7	-4	-2
0	0	0	0	0
1	0	1	1	1
2	1	0	1	1
3	0	1	1	0
4	1	0	1	0
5	0	1	0	1
6	1	0	0	0
7	0	1	0	0
8	1	0	0	0
9	1	1	1	1

- Nota:
 - A partir por ex., do esquema de pesos 5211, é possível elaborar um código ponderado cujo somatório dos pesos é 9 mas que não seja autocomplementar



Outros códigos autocomplementares

algarismo decimal	2421 (AIKEN)	Excess-3 (XS3)	1-out-of-10
0	0000	0011	0000000001
1	0001	0100	0000000010
2	0010	0101	0000000100
3	0011	0110	0000001000
4	0100	0111	0000010000
5	1011	1000	0000100000
6	1100	1001	0001000000
7	1101	1010	0010000000
8	1110	1011	0100000000
9	1111	1100	1000000000



Códigos Gray

- Regular
- Contínuo
- Reflectido
- Cíclico
- Não ponderado
- Tem relação com código binário natural

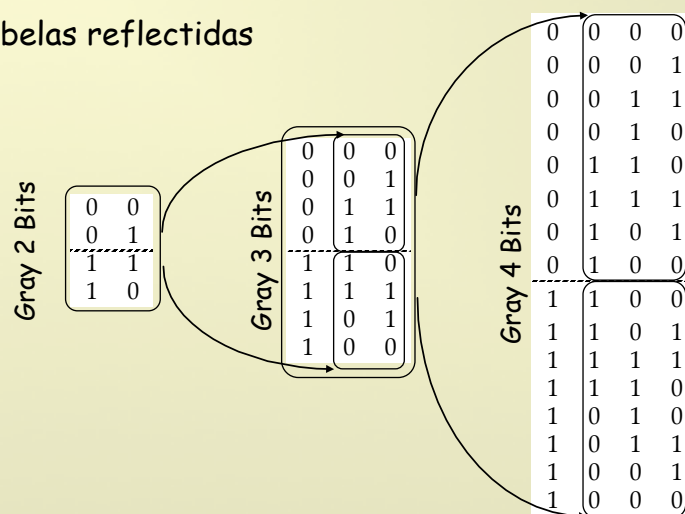
Código de Gray 4 Bits

0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0



Códigos Gray

- Tabelas reflectidas





Relações Código Binário - Código Gray

- **Binary2Gray**
 - Adicionar à palavra do código binário um bit à esquerda e atribuir-lhe o valor '0'.
 - Numerar todos os bits do código binário da direita para a esquerda.
 - Atribuir valor '1' ao bit i do código de Gray se os bits i e $i+1$ da palavra binária são diferentes.
 - Atribuir valor '0' ao bit i do código de Gray se os bits i e $i+1$ da palavra binária são iguais
- **Exercício**
 - Converta para Gray 1111, 1000



Relações Código Binário - Código Gray

- **Gray2Binary**
 - Numerar todos os bits do código de Gray da esquerda para a direita.
 - Atribuir o valor do bit 1 do código de Gray ao bit 1 do código binário.
 - Bit i ($i=2,3,...,n$) do código binário é igual à soma exclusiva (XOR) do bit $i-1$ do código binário e do bit i do código de Gray.
- **Exercício**
 - Converta para Binário 1000, 1100



Códigos Alfanuméricos

- *ASCII - American Standard Code for Information Interchange*

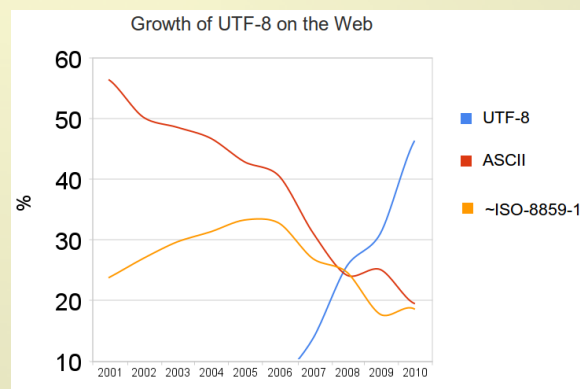
Table 2-11 American Standard Code for Information Interchange (ASCII),
Standard No. X3.4-1968 of the American National Standards Institute.

		$b_2b_3b_4$ (column)							
$b_3b_2b_1b_0$	Row (hex)	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000	0	NUL	DLE	SP	0	@	P	'	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(8	H	X	h	x
1001	9	HT	EM)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[k	{
1100	C	FF	FS	,	<	L	\	l	
1101	D	CR	GS	-	=	M]	m	}
1110	E	SO	RS	.	>	N	~	n	~
1111	F	SI	US	/	?	O	_	o	DEL



Códigos Alfanuméricos

- Outros códigos
 - UTF8, ISO-8859-1





Erros em Sistemas Digitais

- Um **erro** em sistemas digitais é a corrupção de dados do valor correto para um valor diferente. Erros podem ocorrer tanto em sistemas de **transmissão** de informação digital (ruído) como em sistemas de **armazenamento** (memória, discos rígidos).
- **Erro singular** - só um bit de dados é corrompido
- **Erros múltiplos** - 2 ou mais bits de dados são corrompidos
- Erros múltiplos são normalmente menos prováveis que erros singulares



Detecção de Erros

- Um código permite **deteção de erros** se a corrupção de uma palavra resulta numa nova palavra que não faz parte do código.
- Um sistema que permite deteção de erros só gera, transmite e guarda palavras de código válidas.
 - Se uma sequência de bits é uma palavra válida de código, é assumido que está correta.
 - Se uma sequência de bits é uma palavra não válida de código, é assumido que está errada.



Códigos de Paridade

- É possível detetar todos os **erros singulares** se a distância mínima entre todas os possíveis pares de palavras de código ≥ 2
- Para construir um código de 2^n palavras que deteta erros singulares são precisos pelo menos $n+1$ bits
- Exercício
 - Modifique o código BCD8421 de modo que se tenha paridade ímpar



Detecção e correcção de erros

- Distância de Hamming

1	0	1	0	1	0	1	0
1	1	0	1	1	1	1	0

↑ ↑ ↑ ↑
- Conceito de Distância de Hamming (DH) é fundamental
 - M: distância de Hamming mínima (DH_{\min})
 - D: nº de erros detectáveis
 - C: nº de erros corrigíveis

Distância de Hamming = 4

$$M = 2C + D + 1$$



Exemplos

$D = 1, C = 0, M = 2$

Bit de paridade

	P	8	4	2	1
0	0	0	0	0	0
1	1	0	0	0	1
2	1	0	0	1	0
3	0	0	0	1	1
4	1	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	1	0	1	1	1
8	1	1	0	0	0
9	0	1	0	0	1

Códigos m de n

	2 em 5				
	0	1	2	4	7
0	0	0	0	1	1
1	1	1	0	0	0
2	1	0	1	0	0
3	0	1	1	0	0
4	1	0	0	1	0
5	0	1	0	1	0
6	0	0	1	1	0
7	1	0	0	0	1
8	0	1	0	0	1
9	0	0	1	0	1

1 em 2 + 1 em 5

	Biquinário						
	5	0	4	3	2	1	0
0	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0
2	0	1	0	0	1	0	0
3	0	1	0	1	0	0	0
4	0	1	1	0	0	0	0
5	1	0	0	0	0	0	1
6	1	0	0	0	0	1	0
7	1	0	0	0	1	0	0
8	1	0	0	1	0	0	0
9	1	0	1	0	0	0	0

"Quase" ponderado

Ponderado



Códigos de Hamming

• Detecção e correcção de erros

- m bits de informação são entrelaçados com p bits de paridade colocados em posições chave na palavra de código composta
- Sendo m o nº de bits de informação então p deverá ser o menor inteiro tal que

$$2^p \geq m + p + 1$$

- Os bits de paridade são colocados nas posições 1, 2, 4, 8, 16,..., da palavra codificada. Os bits de informação são colocados nas posições "vagas" 3, 5, 6, 7, 9, 10,...
- O valor de cada bit de paridade depende especificamente de alguns bits de informação. Um bit de paridade verifica as posições, incluindo a própria, contendo um "1" no mesmo local da sua representação



Exemplo: Correção de 1 erro

- 4 bits de informação
 - $p \geq \log_2(m + p + 1)$
 - $p = 3$

Tabela de dependências

P1	P2	M3	P4	M5	M6	M7
			X	X	X	X
	X	X			X	X
X		X		X		X

Codificação dos bits de paridade

$$P_1 = M_3 \oplus M_5 \oplus M_7$$

$$P_2 = M_3 \oplus M_6 \oplus M_7$$

$$P_4 = M_5 \oplus M_6 \oplus M_7$$

Na decodificação calculam-se os síndromas

$$S_0 = P_1 \oplus M_3 \oplus M_5 \oplus M_7$$

$$S_1 = P_2 \oplus M_3 \oplus M_6 \oplus M_7$$

$$S_2 = P_4 \oplus M_5 \oplus M_6 \oplus M_7$$

Exemplo

palavra emitida 1 1 1 0 0 0 0

palavra recebida 1 1 0 0 0 0 0

$$S_0 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$S_1 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$S_2 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$$

erro na posição 3