

PCS 3115 (PCS2215)

Sistemas Digitais I

Módulo - Códigos

Prof. Dr. Edison Spina

sobre o material do Prof. Dr. Marcos A. Simplicio Jr.

versão: 4.1 (Mai/2018)

Conteúdo

- Códigos Binários para Decimais
 - BCD ou 8421
 - 2421
 - Excesso 3
 - 2 entre 5
 - 1 entre n
 - 7 segmentos
 - Gray
- Detecção e correção de erros
 - Hamming
- Caracteres
 - ASCII
 - Unicode
- Transmissão Serial
 - Alguns exemplos informativos

CÓDIGOS NUMÉRICOS

Códigos Numéricos

- Conjunto de cadeias com n bits: cadeias diferentes representam coisas diferentes;
 - Para n bits existem 2^n códigos válidos de comprimento fixo
- Exemplo: códigos numéricos
 - Computadores: operam com bits
 - Humanos: preferem operar com decimais
 - → Códigos permitem codificar 10 símbolos decimais (0-9) em termos de bits

Códigos Numéricos

- **Pergunta 1:** quantos bits são necessários para representar 10 dígitos?
 - Resposta: 4 bits → 16 possibilidades (6 códigos não são usados)
- **Pergunta 2:** Qual o código numérico mais imediato e fácil de ser compreendido ? Como você o chamaria ?
 - Resposta: código em binário que representa os números decimais, conforme visto na aula anterior, i.e., número representa valor do bit. Nome dado: BCD (*binary-coded decimal*) ou “8421”.

BCD – Binary Coded Decimal

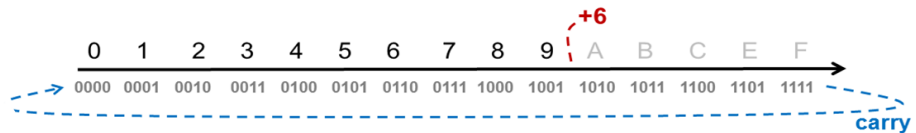
- Cadeias de 4 bits;
 - Código 8421: indica peso de cada um dos bits na cadeia
- Variantes
 - *Packed BCD*: 2 dígitos BCD em 1 byte → 00 a 99;
 - BCD com sinal: um dígito extra representando “+” ou “-”

Cuidado: 1 byte = 8 bits sempre?

Dígito decimal	BCD (8421)	Dígito decimal	BCD (8421)
0	0000	8	1000
1	0001	9	1001
2	0010	não usado	1010
3	0011		1011
4	0100		1100
5	0101		1101
6	0110		1110
7	0111		1111

BCD – Binary Coded Decimal

- **Adição:** similar à de números binários de 4 bits
 - Mas com correção se resultado ultrapassar 1001: somar 6



- Exemplos:

5	0101	8	1000	4	0100
+ 9	+ 1001	+ 8	+ 1000	+ 5	+ 0101
14	1110	16	1 0000	9	1001
	+ 0110		+ 0110		
10+4	1 0100	10+6	1 0110		

correção (indicated by a red dashed arrow pointing from the text to the correction steps in the table)

Código 2421

- Nome indica peso de cada um dos bits na cadeia
 - Ex.: $5 = 1011$, pois $2*1+4*0+2*1+1+1 = 5$
- **Propriedade: auto-complementar**
 - Ao inverter os bits do código de um determinado dígito, obtém-se o código do **complemento de 9** daquele dígito
 - Ex.: $\text{Compl}_9(8) = 1 \rightarrow$ código de 8 é 1110 (o inverso de 0001)

Dígito decimal	2421	Dígito decimal	2421
0	0000	9	1111
1	0001	8	1110
2	0010	7	1101
3	0011	6	1100
4	0100	5	1011
Não usado	0101 (5)	Não usado	1010 (4)
(mas poderia ser)	0110 (6)	(mas poderia ser)	1001 (3)
	0111 (7)		1000 (2)

Código Excesso de 3

- Equivalente a BCD + 3
- **Propriedade: auto-complementar**
 - Ex.: $\text{Compl}_9(8) = 1 \rightarrow$ código de 8 é 1110 (o inverso de 0001)
- Vantagem sobre 2421: aritmética similar a BCD
 - **Lição de casa:** pesquisar/deduzir como fazer

Dígito decimal	Excesso-3	Dígito decimal	Excesso-3
Não usado	0000	5	1000
	0001	6	1001
	0010	7	1010
0	0011	8	1011
1	0100	9	1100
2	0101	Não usado	1101
3	0110		1110
4	0111		1111

Código 2 entre 5

- Código de 5 bits:
 - Usam-se todos os códigos em que (número de bits 1) = 2
- **Propriedade: correção de erros**
 - Em uma transmissão digital, sempre se esperam dois bits '1': se houver um erro na transmissão e um bit for invertido, erro é facilmente detectado.

Dígito decimal	2 entre 5	Dígito decimal	2 entre 5
0	00011	6	10001
1	00101	7	10010
2	00110	8	10100
3	01001	9	11000
4	01010	Não usados: todos os códigos que não tenham dois bits 1	
5	01100		

Código 1 entre n

- Código de n bits:
 - A posição do bit indica o valor representado
- **Propriedade:** correção de erros; facilita seleção
 - Grande possibilidade de detectar inversão de bits
 - Apenas 10 códigos válidos dentre 1024 possíveis
 - Sinal de habilitação de circuito de seleção pode ser ligado diretamente a código

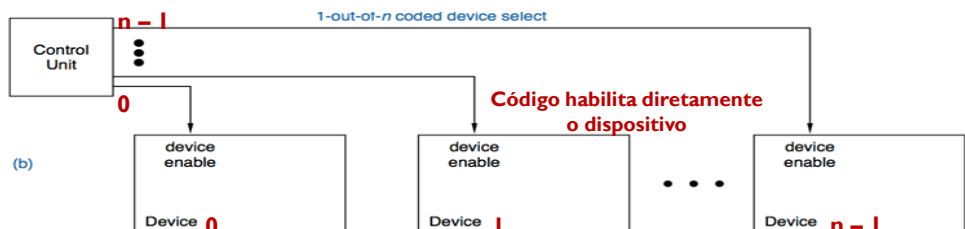
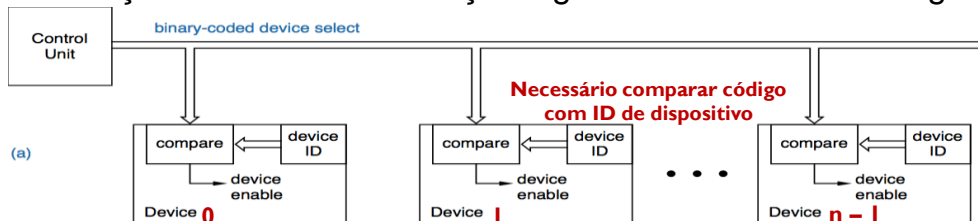
Dígito decimal	1 entre 10	Dígito decimal	1 entre 10
0	000000001	5	0000100000
1	000000010	6	0001000000
2	0000000100	7	0010000000
3	0000001000	8	0100000000
4	0000010000	9	1000000000

Não usados: todos os códigos que não tenham somente um bit 1

Código 1 entre n

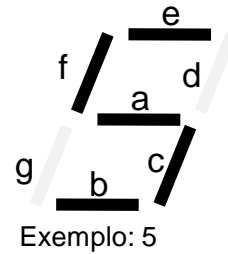
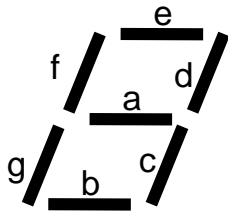
Decodificação automática

- Habilitação de circuito de seleção: ligável diretamente a código



Código 7 Segmentos

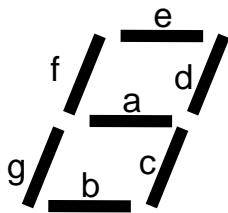
- **Problema:** criar código para iluminar LEDs correspondentes em display de 7 segmentos
 - Objetivo: evitar necessidade de decodificadores



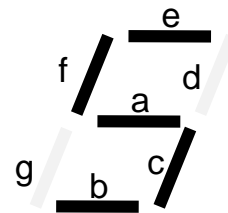
- Quantos bits? Qual o código para cada valor decimal?

Código 7 Segmentos

- Utilizado para iluminar LEDs correspondentes em display de 7 segmentos



Exemplo: 5
→ 1110110
abcdefg

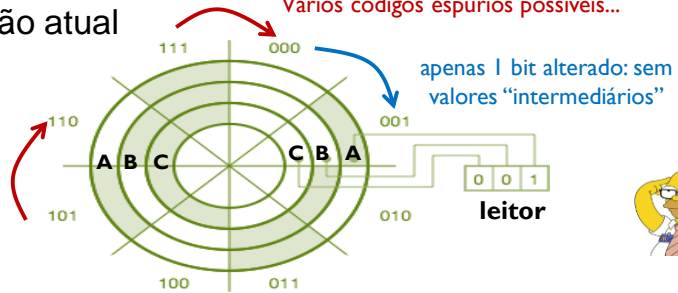


Dígito decimal	7 Segmentos	Dígito decimal	7 Segmentos
0	0111111	5	1110110
1	0011000	6	1110011
2	1101101	7	0011100
3	1111100	8	1111111
4	1011010	9	1011110

Código de Gray

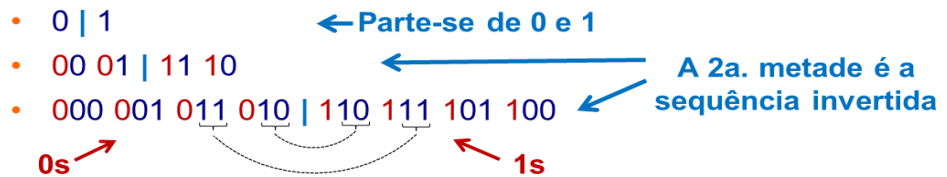
- Cenário: aplicações eletromecânicas (ex.: copiadora, freio automotivo, etc.)
 - Valor digital no leitor indica posição mecânica
- **Problema:**
 - **Valor “na transição”** pode ser interpretado (incorretamente) como a posição atual
 - *Vários códigos espúrios possíveis...*

Se B muda mais rápido que A: transição gera código espúrio 111
 Caso inverso: gera código espúrio 100



Código de Gray

- Propriedades:
 - Apenas **um bit alterado** entre um código e seu sucessor
- Construção:



Dígito decimal	BCD	Gray	Dígito decimal	BCD	Gray
0	000	000	4	100	110
1	001	001	5	101	111
2	010	011	6	110	101
3	011	010	7	111	100

Código de Gray

- Propriedades:
 - Apenas **um bit alterado** entre um código e seu sucessor
- Construção alternativa (baseada no BCD):
 - Enumere os bits da direita para a esquerda
 - Inspeccione código BCD: se, em BCD, bit $i = \text{bit}(i+1)$, então bit i em Gray é **0**; senão, é **1**. Manter bit mais significativo

(alguns exemplos abaixo)

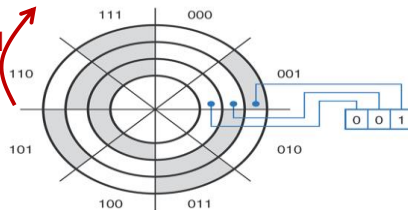
Dígito decimal	BCD	Gray	Dígito decimal	BCD	Gray
0	000	000	4	00	110
1	001	001	5	01	111
2	010	011	6	110	101
3	01	010	7	11	100

Código de Gray

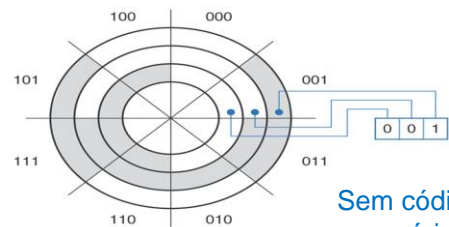
- Não aparecem valores espúrios na transição entre posições do disco



códigos espúrios: 100 ou 111



Codificação BCD

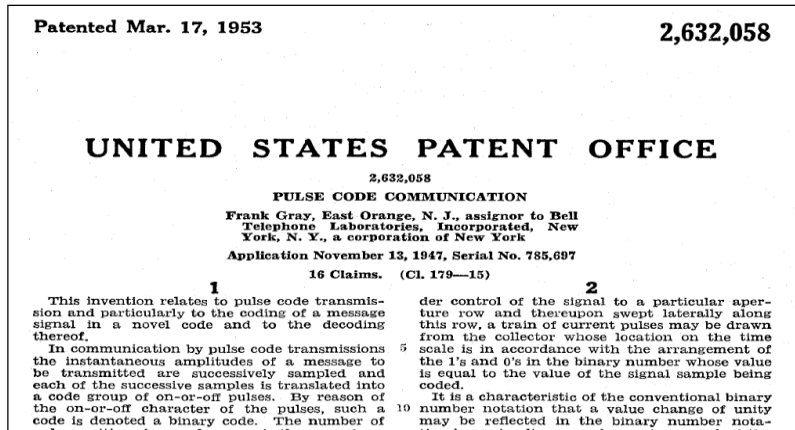


Codificação Gray

Sem códigos espúrios

Código de Gray

- Patente US 2,632,058 – “Pulse Code Communication” (1953).
Inventor: Frank Gray, Bell Labs



Exercícios

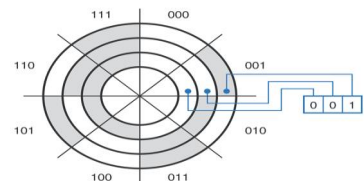
1) Converta os números decimais para BCD e 2421:

a) 742

b) 268

2) Some os números 742 e 268 em BCD

3) Quantas “fronteiras ruins” existem no codificação BCD de 3 bits?)



4) Responda a questão 3 para um disco de n bits, em função de n

Exercícios: Respostas

1) Converta os números decimais para BCD e para 2421:

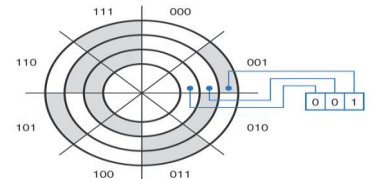
a) **742** → BCD: 0111 0100 0010 2421: 1101 0100 0010

b) **268** → BCD: 0010 0110 1000 2421: 0010 1100 1110

2) Some os números 742 e 268 em BCD

$$\begin{array}{r}
 742 \\
 + 268 \\
 \hline
 1010
 \end{array}
 \qquad
 \begin{array}{r}
 \overset{1}{0}\overset{1}{1}\overset{1}{1} \quad \overset{1}{0}100 \quad 0010 \\
 + 0010 \quad \quad \quad 0110 \quad \quad 1000 \\
 \hline
 100\overset{1}{1} \quad \quad \quad \overset{1}{1}\overset{1}{0}\overset{1}{1} \quad \quad \quad \overset{1}{1}\overset{1}{0}\overset{1}{1} > 1001 \\
 + 0000 \quad \quad \quad + 0110 \quad \quad \quad + 0110 \\
 \hline
 \overset{1}{1}\overset{1}{0}\overset{1}{1} \quad \quad \quad 0001 \quad \quad \quad 0000 \\
 + 0110 \\
 \hline
 1 \quad 0000 \\
 \hline
 1 \quad 0 \quad \quad \quad 1 \quad \quad \quad 0
 \end{array}$$

Exercícios: Respostas



3) Quantas “fronteiras ruins” existem no disco de codificação BCD de 3 bits?)

001 → 010

101 → 110

011 → 100

111 → 000

4) Responda a questão 3 para um disco de n bits, em função de n

→ Quando bit menos significativo (LSB) muda de 0 para 1, não há problema. O problema é quando ele muda de 1 para 0, gerando carries, o que altera pelo menos um bit mais alto. Logo, metade das fronteiras (2^{n-1}) geram problemas

CÓDIGOS PARA CARACTERES

Código ASCII

- Informação processada por computador: bits
- Então como representar texto...?
 - Usa-se um código: tabela que especifica representação binária para um determinado conjunto de símbolos.
 - Comumente: ASCII (*American Standard Code for Information Interchange*),
 - Código alfanumérico: letras do alfabeto, números, símbolos, sinais e alguns caracteres não-imprimíveis de controle
 - 7 bits: 128 caracteres diferentes (letras com acentos não incluídas)
 - 8 bits: 256 caracteres diferentes (acentuadas, gregas, etc)
 - Pronúncia: “ASKI”, não “ASK2”

“echo ^G”

Código ASCII

$b_3b_2b_1b_0$	Row (hex)	$b_6b_5b_4$ (column)							
		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

Código ASCII

Control codes

NUL	Null	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronize
BEL	Bell	ETB	End transmitted block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete or rubout

Unicode

- ASCII não suporta caracteres com acentos...
 - Apenas 7 bits: desenvolvido para alfabeto inglês
- Unicode: suporte a caracteres em múltiplos idiomas
 - 32 bits: ~4 bilhões de símbolos possíveis
 - Construído para ser compatível com ASCII: códigos de 00 a 7F representam os mesmos símbolos em Unicode e ASCII
 - Padrão ISO/IEC 10646: define 3 métodos de codificação UTF-8, UTF-16 e UTF-32
 - UTF = Unicode Transformation Format
 - Maiores informações: www.unicode.org

Lição de Casa

- Leitura Obrigatória:
 - Capítulo 2 do Livro Texto.
- Exercícios Obrigatórios:
 - Capítulo 2 do Livro Texto;
 - Lista de Exercícios do Módulo 4.

Livro Texto

- Wakerly, J.F.; *Digital Design – Principles & Practices*; Fourth Edition, ISBN: 0-13-186389-4, Pearson & Prentice-Hall, Upper Saddle, River, New Jersey, 07458, 2006.

Bibliografia Adicional

- Giozza, William Ferreira; et all; *Redes Locais de Computadores: Tecnologia e Aplicações – Seção 3.2.3 Codificação em Banda Básica – Códigos*; Editora McGraw-Hill, 1.986;
- Hayes, J.P.; *Computer Architecture and Organization*; McGraw-Hill, 1988;