

Códigos BCD

Dígito decimal	Cód. NBCD (8421)	Cód.Aiken (2421)	Cód.Stibitz (8421 – 3)	Cód.7421 (7421)	Cód. 642-1 (642-1)
0	0000	0000	0011	0000	0000
1	0001	0001	0100	0001	0011
2	0010	0010	0101	0010	0010
3	0011	0011	0110	0011	0101
4	0100	0100	0111	0100	0100
5	0101	1011	1000	0101	0111
6	0110	1100	1001	0110	1000
7	0111	1101	1010	0111	1011
8	1000	1110	1011	1001	1010
9	1001	1111	1100	1010	1101

Soma em BCD

Na soma de dois dígitos BCD (entre 0 e 9) em binário, o dígito resultante pode estar em um de três casos:

1. Dígito legal (entre 0 e 9), sem “vai-um”. Resultado está correto e não existe “vai-um” para o dígito seguinte
2. Dígito ilegal sem “vai-um”. Resultado está entre 10 e 15 (em binário); para obter o dígito correto, subtrair 10 do dígito (ou somar seis, o que é equivalente), e gerar um “vai-um” para o dígito decimal seguinte
3. Dígito legal com “vai-um”. Quando o resultado cai entre 16 e 19; para obter o dígito correto subtrair 10 do dígito (ou somar seis). O “vai-um” gerado está correto

Soma em BCD

Sejam A = 0832 e B = 0983

	1			
A =	0000	1000	0011	0010
B =	0000	1001	1000	0011
	<hr/>			
	0001	0001	1011	0101
	caso	caso	caso	caso
	1	3	2	1

Tratando-se cada caso, tem-se:

	1		
0001	0001	1011	0101
	0110	0110	
	<hr/>		
0001	1000	0001	0101

Soma em BCD

A=0372 e B=0633

A = 0000 0011 0111 0010

B = 0000 0110 0011 0011

0000 1001 1010 0101

caso

2

Com a correção do caso 2 tem-se:

1

0000 1001 1010 0101

0110

0000 1010 0000 0101

caso

2

E a correção deste caso fornece o resultado final (1005)

Algoritmo de Hellerman

- soma 6 em todos os dígitos de um dos operandos antes da soma das duas parcelas
- só existem dois casos a serem tratados, distinguidos pelo “vai-um”:
 1. O resultado não deu “vai-um” e então caiu entre 6 e 15. Deve-se subtrair 6 para obter o dígito correto.
 2. O resultado produziu um “vai-um”. Então este “vai-um” já foi propagado e o dígito está correto entre 0 e 9.

Exemplo: seja A=0372 e B=0633

A =	0000	0011	0111	0010
soma de 6	0110	0110	0110	0110
<hr/>				
	0110	1001	1101	1000

Algoritmo de Hellerman

Segunda etapa: soma de (A+6) com B:

	1	1		
A+6=	0110	1001	1101	1000
B=	0000	0110	0011	0011
	<hr/>			
	0111	0000	0000	1011
	caso	caso	caso	caso
	1	2	2	1

Terceira etapa: somar 10 e ignorar “vai-um”:

0111	0000	0000	1011
1010			1010
<hr/>			
0001	0000	0000	0101

Códigos BCD

Dígito decimal	NBCD (8421)	Excesso-de-3 (8421 – 3)
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Código Stibitz (8421-3)

Vantagens:

- nenhum código utiliza a combinação 0000
- é auto-complementado para 9 (para se obter o complemento de 9 basta inverter todos os bits)

Aritmética:

- resultado é em excesso de 6
- somente dois casos a tratar:
 - soma dos dois dígitos é nove ou menos - basta subtrair 3
 - soma dos dois dígitos é 10 ou mais - basta somar 3; vai-um correto

Assim, a regra para a soma em excesso de três é simples: somam-se os dígitos usando aritmética binária; se um 'vai-um' é gerado, somar 3 (0011) ao dígito decimal; senão, subtrair 3 (0011) ao dígito decimal (ou somar 1101 e desprezar o 'vai-um')

Códigos de 5 bits ponderados

Dígito decimal	Cód. 74210
0	11000
1	00011
2	00101
3	00110
4	01001
5	01010
6	01100
7	10001
8	10010
9	10100

- somente 2 bits em 1 em cada código
- permite detecção de erros simples
- exceção: código de zero não usa os pesos, mas mantém dois 1s

Códigos de 7 bits ponderados

Dígito decimal	50 43210
0	01 00001
1	01 00010
2	01 00100
3	01 01000
4	01 10000
5	10 00001
6	10 00010
7	10 00100
8	10 01000
9	10 10000

- somente 2 bits em 1 em cada código: 1 à esquerda e 1 à direita
- esquerda = 01 → valor de 0 a 4; esquerda = 10 → valor de 5 a 9
- incrementar em 1 = deslocar para a esquerda a parte da direita

Códigos Gray (cíclicos)

- somente 1 bit varia de um código para o seguinte
- usados para indicar a variação de grandezas analógicas
- não existem combinações ilegais

Dígito decimal	Cód. Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Códigos Gray (cíclicos)

- “walking code”
- para somar 1:
 - deslocar para esquerda
 - complementar msbit
 - e usar como novo lsbit

Dígito	Código
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

Códigos de detecção de erros

- exigem redundância:
 - bits adicionais ou
 - combinações “válidas” e “inválidas”
- exemplos:
 - códigos “m de n” (já vistos)
 - paridade
 - código de Hamming

Códigos de paridade

Código	Paridade par	Soma dos '1'	Paridade ímpar	Soma dos '1'
000	0	0	1	1
001	1	2	0	1
010	1	2	0	1
011	0	2	1	3
100	1	2	0	1
101	0	2	1	3
110	0	2	1	3
111	1	4	0	3

Códigos de Hamming

- “distância de Hamming”: número de bits que é preciso alterar para passar de um código válido para outro:
 - entre 0100 e 0010, $d = 2$
 - entre 0000 e 1111, $d = 4$
- distância 1: não é possível detectar/corrigir erros
- distância 2: é possível apenas detectar erro em 1 bit
- distância 3: possível detectar e corrigir erro em 1 bit; possível detectar erros em 2 bits sem corrigir
- distância $2n+1$: detecta $2n$, corrige apenas n

Códigos de Hamming - exemplo

Posição	1	2	3	4	5	6	7
Código	A	B	8	C	4	2	1
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2	0	1	0	1	0	1	0
3	1	0	0	0	0	1	1
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1
10	1	0	1	1	0	1	0
11	0	1	1	0	0	1	1
12	0	1	1	1	1	0	0
13	1	0	1	0	1	0	1
14	0	0	1	0	1	1	0
15	1	1	1	1	1	1	1

Usa paridade par:

A

paridade de 1,3,5,7

B

paridade de 2,3,6,7

C

paridade de 4,5,6,7

→ 128

combinações

→ só 16 válidas

Códigos de Hamming - correção

- calcula-se o bit de paridade par de cada grupo e forma-se um número binário de 3 bits: *cba*

posição:	1	2	3	4	5	6	7
pesos:	A	B	8	C	4	2	1
valor:	1	0	1	0	0	1	0
c = 1				0	0	1	0
b = 0		0	1			1	0
a = 0	1		1		0		0
cba = 100₂ (ou 4₁₀)							

- se *cba* = 0, não há erro
- se *cba* ≠ 0, então há erro no bit apontado pelo valor decimal representado por *cba*

Códigos de Hamming - exemplo

Posição	1	2	3	4	5	6	7
Código	A	B	8	C	4	2	1
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2	0	1	0	1	0	1	0
3	1	0	0	0	0	1	1
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1
10	1	0	1	1	0	1	0
11	0	1	1	0	0	1	1
12	0	1	1	1	1	0	0
13	1	0	1	0	1	0	1
14	0	0	1	0	1	1	0
15	1	1	1	1	1	1	1

Usa paridade par:

A

paridade de 1,3,5,7

B

paridade de 2,3,6,7

C

paridade de 4,5,6,7

Exemplos:

0111010

1011001

1110110

Código ASCII padrão USA (7 bits)

Bits	Bits superiores (mais significativos)							
inferiores	000	001	010	011	100	101	110	111
0000	null	dle		0	@	P	`	p
0001	soh	dc1	!	1	A	Q	a	q
0010	stx	dc2	"	2	B	R	b	r
0011	etx	dc3	#	3	C	S	c	s
0100	eot	dc4	\$	4	D	T	d	t
0101	enq	nak	%	5	E	U	e	u
0110	ack	syn	&	6	F	V	f	v
0111	bell	etb	'	7	G	W	g	w
1000	bsp	can	(8	H	X	h	x
1001	ht	em)	9	I	Y	i	y
1010	lf	sub	*	:	J	Z	j	z
1011	vt	esc	+	;	K	[k	{
1100	ff	fs	,	<	L	\	l	
1101	cr	gs	-	=	M]	m	}
1110	so	rs	.	>	N	^	n	~
1111	si	us	/	?	O	_	o	del