

**Uniube**

UNIUBE – CAMPUS VIA CENTRO – Uberlândia/MG

Curso de Engenharia Elétrica e Engenharia de Computação

Disciplina: Sistemas Digitais

Aula 05

Código BCD e Código de Gray

Revisão 3, de 12/03/2025

Prof. João Paulo Seno

joao.seno@uniube.br

1

**Uniube**

Código BCD

- BCD significa “Decimal Codificado em Binário” (*Binary Coded Decimal*);
- A ideia aqui é que cada dígito de um número decimal seja representado por um número binário.
- Como um dígito decimal tem 10 símbolos, de 0 a 9, são necessários 4 bits para codificar um dígito.
- Veja os dois exemplos abaixo, os números decimais 874 e 943 seriam representados da seguinte forma:

8	7	4	(decimal)
↓	↓	↓	
1000	0111	0100	(BCD)

e

9	4	3	(decimal)
↓	↓	↓	
1001	0100	0011	(BCD)

2



Código BCD

- O código BCD representa, então, cada dígito de um número decimal por um número binário de 4 bits.
- Há alguns conjuntos, que são 1010, 1011, 1100, 1101, 1110 e 1111 que não são utilizados, ou seja, são utilizadas apenas 10 das 16 combinações possíveis para os 4 bits.
- Se algum número “proibido” aparecer em alguma operação, é sinal de que provavelmente aconteceu um erro.

3



Exemplo

- Converta o número binário abaixo, representado em BCD, para seu equivalente decimal.

0110100000111001 (BCD)

- Resposta: 6839

4

Importante

- Observe que o BCD não é outro sistema de numeração, como é o binário, o octal, o decimal e o hexadecimal. Trata-se apenas de uma codificação de números decimais utilizando números binários.
- Por que é utilizado? Resposta: Pela facilidade de conversão entre decimal e BCD. É possível utilizar uma tabela e não são necessários cálculos, como no caso da conversão binário – decimal.
- Problema: desperdício de informação! Seis combinações possíveis dos 4 bits não são utilizadas.

5

Tipos de representação BCD

Dígito decimal	Cód. NECD (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

6



Vantagens da representação BCD

- É mais simples apresentar os números em displays e impressoras, pois basta fazer um mapeamento dígito a dígito;
- Alguns números que são exatos no sistema decimal (por exemplo: 0,20) tem uma representação binária infinita (0,001100110011...). Em BCD este problema não ocorre, a representação seria: 0,0010 simplesmente;
- Multiplicar (ou dividir) por 10 é muito mais simples em BCD;
- Arredondamentos também são mais fáceis em BCD.

7



Desvantagens da representação BCD

- Algumas operações são mais complexas de serem implementadas e exigem circuitos eletrônicos maiores.
- É ineficiente, pois há combinações de dígitos binários que não são utilizadas.
- Consomem cerca de 20% a mais de recurso em relação à representação binária pura..
- As implementações das operações são geralmente mais lentas.

8

**Uniube**

Algoritmo para adição de números em BCD (NBCD)

- A seguir é apresentado um exemplo de algoritmo para adição de números em BCD (NBCD).
- Na aritmética BCD os dígitos decimais são operados em binário, agrupados de quatro em quatro. Quando se soma dois dígitos BCD em binário, o dígito resultado pode estar em um de três casos:

9

**Uniube**

Algoritmo para adição de números em BCD (NBCD) (CASOS)

1. Dígito legal ou válido (entre 0 e 9), sem “vai um”. Neste caso o resultado está correto, e não existe “vai um” para o dígito seguinte.
2. Dígito ilegal ou inválido sem “vai um”. Neste caso o resultado está entre 10 e 15 (em binário); para obter-se o dígito correto deve-se subtrair 10 do dígito (ou somar seis, o que é equivalente), e gerar-se um “vai um” para o dígito decimal seguinte.
3. Dígito legal ou válido com “vai um”. Este caso ocorre quando o resultado cai entre 16 e 19. Da mesma maneira que o caso 2, para obter-se o dígito correto deve-se subtrair 10 do dígito (ou somar seis). O “vai um” que havia sido gerado está correto.

10



Exemplo

- Sejam $A = 0832$ e $B = 0983$, calcular $A + B$
- Constrói-se a representação BCD e efetua-se a soma da maneira tradicional:

$$\begin{array}{cccc}
 & 0000 & 1000 & 0011 & 0010 \\
 + & 0000 & 1001 & 1000 & 0010 \\
 \hline
 & 0001 & 0001 & 1011 & 0101 \\
 & \text{caso 1} & \text{caso 3} & \text{caso 2} & \text{caso 1}
 \end{array}$$

- É preciso corrigir o resultado, levando-se em consideração cada caso, conforme descrito. No exemplo acima, precisamos corrigir os casos 2 e 3.

11



Exemplo (continuação)

- Tomando o resultado, somando 6 onde necessário e lembrando-se do “vai um”, temos:

$$\begin{array}{cccc}
 & & & 1 & \\
 & 0001 & 0001 & 1011 & 0100 \\
 + & & & 0110 & 0110 \\
 \hline
 & 0001 & 1000 & 0001 & 0101
 \end{array}$$

- Obtemos então o resultado final, que é 1815.
- Observações:
 - 1) As correções dos casos tipo 2 podem gerar a necessidade de novas correções. É preciso verificar.
 - 2) Há diversos algoritmos para evitar ou antecipar as correções. Porém, não nos interessam porque não tem aplicação prática.

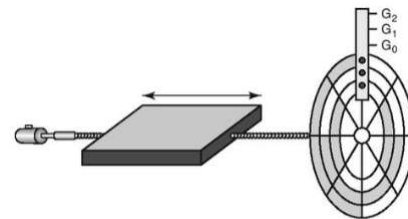
12

Código de Gray

- É utilizado para diminuir a probabilidade de que um sistema digital interprete de forma errada uma entrada que está mudando, como é o caso de “encoders”;
- A única característica particular dos códigos de Gray é que apenas um *bit* muda entre dois números sucessivos na sequência;
- Foi inventado por Frank Gray em 1953.

Equivalentes entre binários de três bits e código Gray.

B ₂	B ₁	B ₀	G ₂	G ₁	G ₀
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0



13

Exemplos de aplicação do código de Gray

- Controle de posição de eixos rotativos em máquinas;
- Minimização de circuitos;
- Correção de erros na transmissão digital;
- Sensores digitais;
- Conversores analógico-digitais (ADCs);
- Codificadores rotativos;
- Computação quântica;
- Mapas de Karnaugh;
- Altímetros;
- Displays de 7 segmentos.

14

Bit de paridade

- É um bit extra anexado a um conjunto de bits do código a ser transferido de uma localidade para outra.
- O bit de paridade pode ser 0 ou 1, dependendo do número de “1s” que exista no conjunto. No método da paridade par, se o número de bits “1s” contido no código for par, o bit de paridade será 0. Caso contrário, será 1.

1 1 0 0 0 0 1 1
↑ bit de paridade anexado¹

- A lógica poderia ser outra. Por exemplo, caso o número de bits “1s” fosse par, o bit de paridade seria 1. O importante, obviamente, é que tanto o transmissor quanto o receptor estejam utilizando a mesma regra.

15

Fim

16