Sistema de Numeração e Códigos

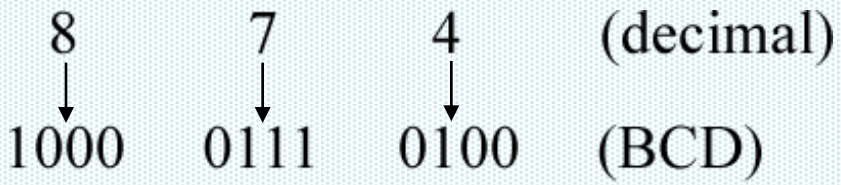
CPCX – UFMS
Prof. Renato F. dos Santos

2.4 Código BCD (Binary-coded decimal)

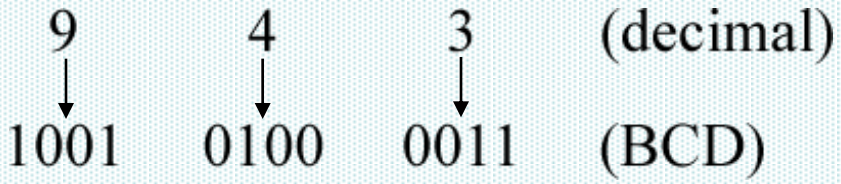
- **Quando um número decimal é representado pelo seu número binário equivalente, dizemos que é uma codificação em binário puro**
- **Todos os sistemas digitais usam alguma forma de numeração binária em suas operações internas;**
- **Devido ao mundo externo ser naturalmente decimal, conversões entre os sistemas decimal e binário são realizadas freqüentemente**
- **São usados apenas 10 dos 16 possíveis grupos de 4 bits**

Exemplo

Conversão do número decimal 874 para BCD:



Exemplificando novamente, vamos converter 943 em código BCD:



Comparação entre BCD e binário

- **BCD não é um sistema de numeração como os sistemas binário, decimal e hexadecimal**
- **Um número BCD não é o mesmo que um número binário puro**
- **O binário puro é obtido a partir do número decimal completo**
- **No código BCD, cada dígito decimal é convertido, individualmente, em binário**

Exemplo

O número 137, comparando os códigos BCD e binário puro:

$$137_{10} = 10001001_{2} \quad \text{(binário)}$$
$$137_{10} = 0001\ \ 0011\ \ 0111 \quad \text{(BCD)}$$

Comparação entre BCD e binário (Continuação)

- **A vantagem do código BCD é a relativa facilidade de conversão em decimal e vice-versa**
- **Essa característica de fácil conversão é importante do ponto de vista do hardware, porque nos sistemas digitais há circuitos lógicos que realizam as conversões mútuas entre BCD e decimal**

2.5 O código Gray

- **Desenvolveu-se com o intuito de reduzir a probabilidade de um circuito digital interpretar mal uma entrada que está mudando**
- **Representa uma seqüência de números**
- **Sua única característica distintiva é que apenas um bit muda entre dois números sucessivos na seqüência**

2.5 O código Gray (Continuação)

- **A tabela 2.2 mostra a transição entre valores binários de três bits e do código Gray**
- **Para converter binários para Gray**
  - **Comece pelo bit mais significativo e use-o como o Gray MSB com o próximo bit binário (B1)**
  - **Se forem iguais, então G1 = 0. Se forem diferentes, nesse caso G1 = 1.**
  - **G0 pode ser encontrado comparando-se B1 com B0**

2.5 O código Gray (Continuação)

- **Para converter Gray em binários**
  - **Observe que o MSD em Gray é sempre o mesmo que o MSD em binário**
  - **O próximo bit de binário é encontrado, comparando-se o bit binário da esquerda com o correspondente bit em código Gray**
  - **Bits similares produzem um 0 e bits diferentes produzem um 1**

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

Tabela 2.2 Equivalentes entre binários de três bits e código Gray

Conversão >>

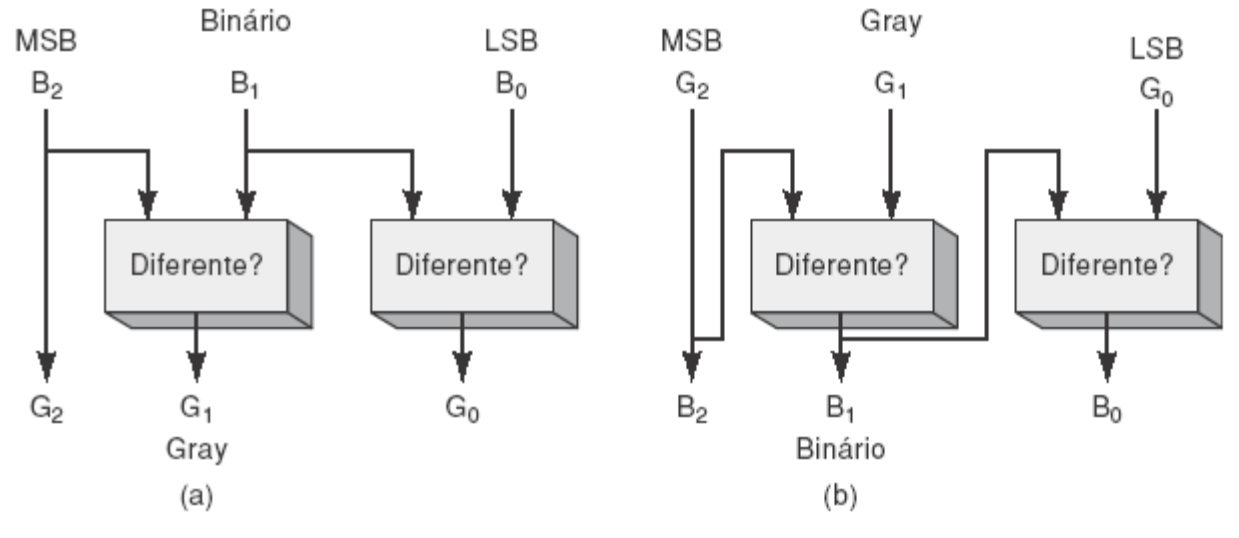
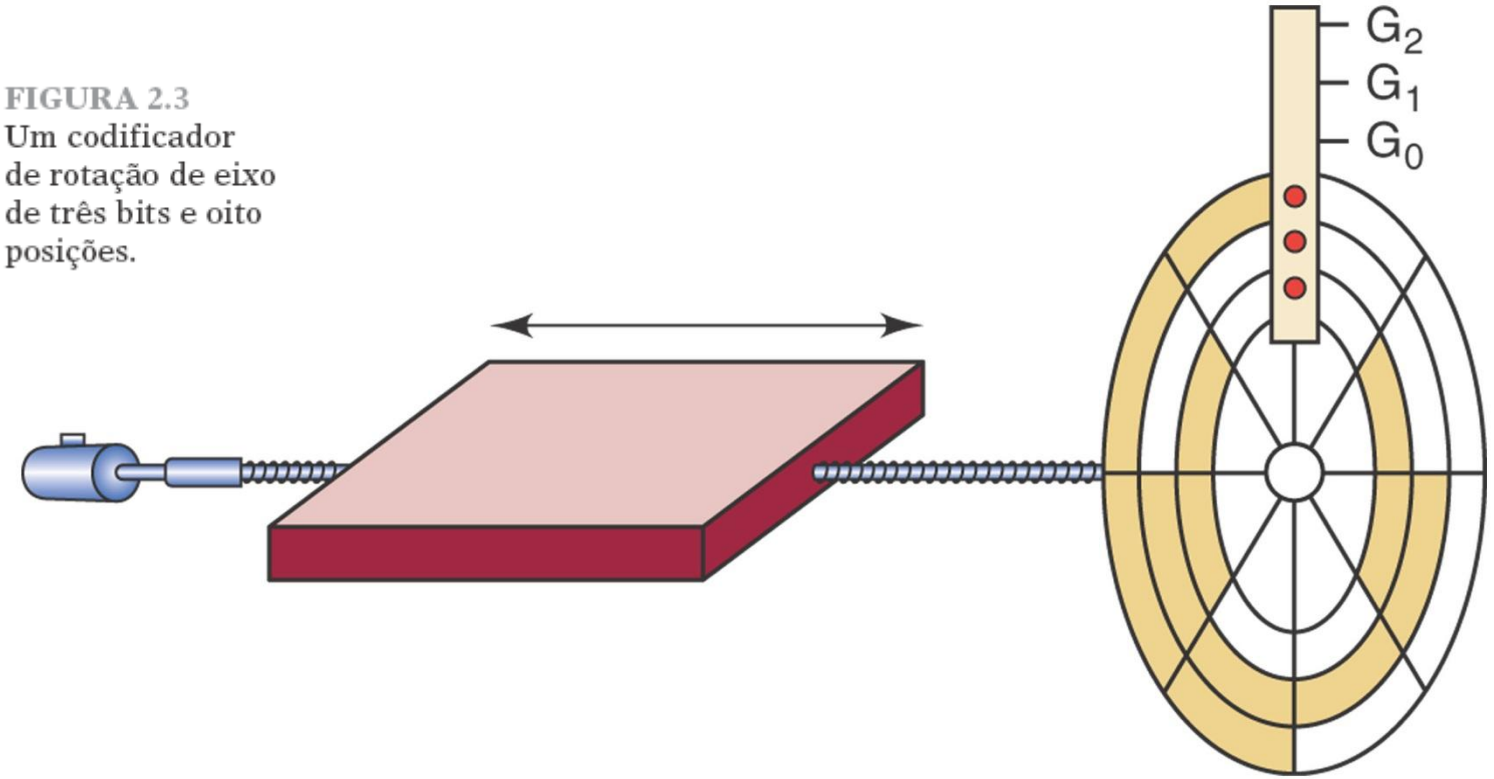


FIGURA 2.2
Convertendo
(a) binário em Gray e
(b) Gray em binário.

2.5 O código Gray (Continuação)

- **A aplicação mais comum do código Gray é nos codificadores de rotação de eixo como mostra a fig. 2.3**
- **Esses dispositivos produzem um valor binário que representa a posição de um eixo mecânico em rotação**
- **Um codificador de rotação prático usaria mais do que três bits e dividiria a rotação em mais do que oito segmentos, de modo a poder detectar incrementos de rotação muito menores**

FIGURA 2.3
Um codificador de rotação de eixo de três bits e oito posições.



2.6 Relações entre as representações numéricas

Decimal	Binário	Hexa	BCD	GRAY
0	0	0	0000	0000
1	1	1	0001	0001
2	10	2	0010	0011
3	11	3	0011	0010
4	100	4	0100	0110
5	101	5	0101	0111
6	110	6	0110	0101
7	111	7	0111	0100
8	1000	8	1000	1100
9	1001	9	1001	1101
10	1010	A	0001 0000	1111
11	1011	B	0001 0001	1110
12	1100	C	0001 0010	1010
13	1101	D	0001 0011	1011
14	1110	E	0001 0100	1001
15	1111	F	0001 0101	1000

2.7 Bytes, nibbles e palavras

- **Bytes**
  - **A maioria dos microcomputadores manipula e armazena informações e dados binários em grupos de 8 bits**
- **Nibbles**
  - **O termo surgiu nos primórdios dos sistemas digitais para descrever um grupo de 4 bits**
  - **Números muitas vezes são representados em grupos de 4 bits, assim como nas conversões de BCD e de números hexadecimais**

2.7 Bytes, nibbles e palavras (Continação)

- **Palavras (Words)**
  - **É um grupo de bits que representa uma certa unidade de informação**
  - **Seu tamanho pode ser definido como o número de bits da palavra binária sobre o qual um sistema digital opera**

Exemplo parcial da tabela ASCII

Caractere	HEX	Decimal
Space	20	32
!	21	33
@	40	64
A	41	65
a	61	97
:	3A	58
=	3D	60
?	3F	63

2.8 Códigos alfanuméricos

- **Um computador deve reconhecer dígitos que representem letras do alfabeto, sinais de pontuação e outros caracteres especiais, assim como números**
- **Tais códigos são denominados *códigos alfanuméricos***
- **Um código alfanumérico completo inclui:**
  - **26 letras minúsculas, 26 maiúsculas, 10 dígitos numéricos, 7 sinais de pontuação e em torno de 10 a 40 caracteres como +, /, #, %, * e assim por diante.**

Código ASCII

- **Código Padrão Americano para Troca de Informações (*Americam Standard Code for Information Interchange)* – ASCII**
- **Código alfanumérico mais utilizado**
- **É um código de 7 bits (128 representações)**

Bit de paridade (Continuação)

- **Paridade ímpar**
  - **Usado exatamente da mesma maneira**
  - **Exceto que o bit de paridade é determinado para que o número total de 1s seja ímpar (incluindo o bit de paridade)**
  - **Por exemplo, o conjunto de bits 1000001 (código ASCII do caractere 'A')**
    - **Esse conjunto de bits tem dois 1s**
    - **Portanto anexamos um bit de paridade ímpar igual a 1, tornando ímpar o número total de 1s.**
    - **O novo conjunto de bits passa a ser: 1 1 0 0 0 0 0 1**

Detecção de erros pelo método da paridade

- **A movimentação de dados e códigos binários de um local para outro é a operação mais freqüente realizada em sistemas digitais.**
- **Eis alguns exemplos:**
  - **A transmissão de voz digitalizada por um enlace (*link*) de microondas**
  - **O armazenamento e a recuperação de dados armazenados em dispositivos de memorização externa, como fitas e discos magnéticos**
  - **A transmissão de dados digitais de um computador para outro que esteja distante por meio da linha telefônica**

Detecção de erros pelo método da paridade (Continuação)

- **Quando uma informação é transmitida, há a possibilidade de ocorrência de erros**
- **Isso ocorre quando o receptor não recebe uma informação idêntica a que foi enviada pelo transmissor**
- **A principal causa de erro é o ruído elétrico**
  - ***Ruídos elétricos*: consistem em flutuações espúrias (aleatórias) na tensão ou corrente que estão presentes em todos os sistemas em intensidades diversas**

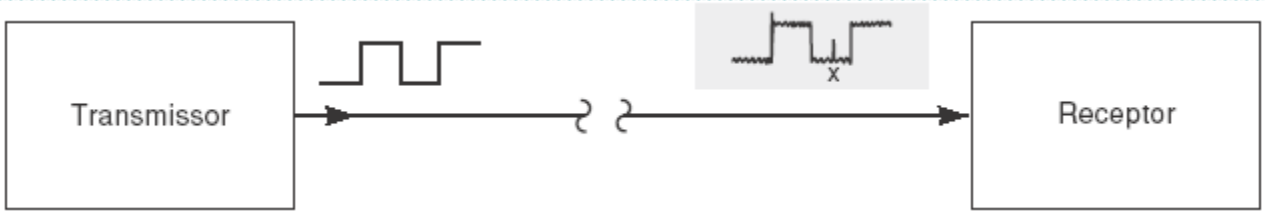


FIGURA 2.4
Exemplo de um erro causado por um ruído em uma transmissão digital.

Detecção de erros pelo método da paridade (Continuação)

- **O transmissor envia um sinal digital, no formato serial, por meio de uma linha de sinal para o receptor**

- **No momento em que chega ao receptor, apresenta um certo nível de ruído sobreposto ao sinal original**

- **O ruído pode ter amplitude grande o suficiente para alterar o nível lógico do sinal (*ponto x – fig.2.4)***

- **O receptor pode interpretar incorretamente que o bit em questão tenha nível lógico 1, o que não corresponde a informação enviada pelo transmissor**

Bit de paridade

- **Consiste em um bit extra anexado ao conjunto de bits do código a ser transferido**
- **Pode ser 0 ou 1**
- **Dois métodos são usados**
  - **Paridade *par***
  - **Paridade *ímpar***

Bit de paridade (Continuação)

- **Paridade par**
  - **O valor do bit de paridade é determinado para que o número total de 1s no conjunto de bits do código (incluindo o bit de paridade) seja um número par**
  - **Por exemplo, o conjunto de bits 1000011 (código ASCII do caractere 'C')**
    - **Esse conjunto de bits tem *três* 1s**
    - **Portanto anexamos um bit de paridade par igual a 1, tornando par o número total de 1s.**
    - **O novo conjunto de bits passa a ser:** 1 1 0 0 0 0 1 1

bit de paridade anexado

Bit de paridade (Continuação)

- **Quer a paridade utilizada seja par, ou ímpar, o bit de paridade passa a ser parte do código atual da informação**
- **É utilizado para detectar erros de um só bit que ocorram durante a transmissão**
- **Suponha que o caractere ‘A’ seja transmitido e que seja usada a paridade ímpar. O código seria:**

1 1 0 0 0 0 0 1

Bit de paridade (Continuação)

- **Quando o circuito receptor recebe esse código, ele verifica se contém um número ímpar de 1s**
- **Suponha que devido a algum ruído ou mau funcionamento do circuito receptor seja recebido o código:**

1 1 0 0 0 0 0 0

- **O receptor identificará que o código tem um número par de 1s, o que significa para o receptor que há um erro no código**
- **Não há uma maneira do receptor identificar qual bit está errado**

Bit de paridade (Continuação)

- **O método de paridade não funciona se ocorrer erro em dois bits**
- **Tem de haver concordância entre transmissor e receptor com relação ao tipo de paridade a ser usada**

Exemplo 2.15

– **Quais seriam as cadeias de caracteres de bits transmitidas por um computador para enviar a mensagem 'HELLO' usando ASCII com paridade par?**