



Universidade Federal do ABC

# BC-0504

# Natureza da Informação

## Aula 3

**Transmissão serial e paralela da  
informação e medida da informação  
transmitida**

Equipe de professores de Natureza da Informação

# Parte 0



# Sistemas de numeração

- Decimal
- Binário
- Hexadecimal
- Octal

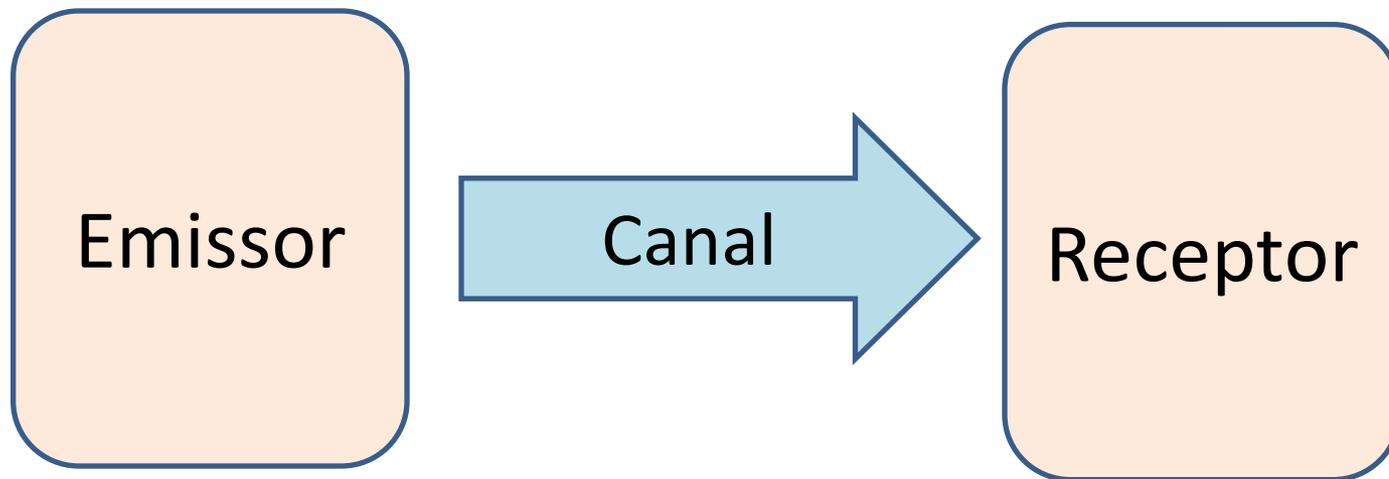
# Parte 1



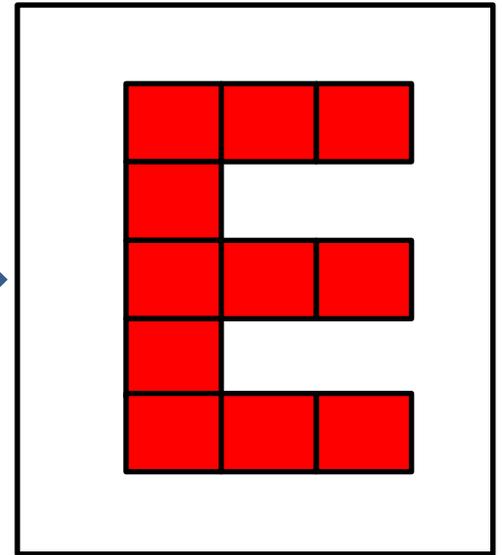
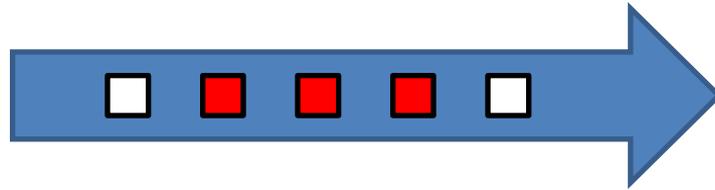
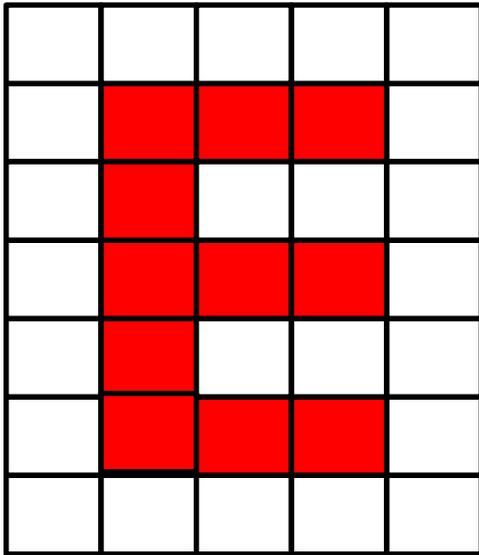
# Os pilares da teoria da informação

- Os estudos de criptografia desenvolvidos na 2ª Guerra mundial
- Os estudos de “termodinâmica”
- **As tecnologias de transmissão de informação**

# Elementos de um sistema de transmissão de informação



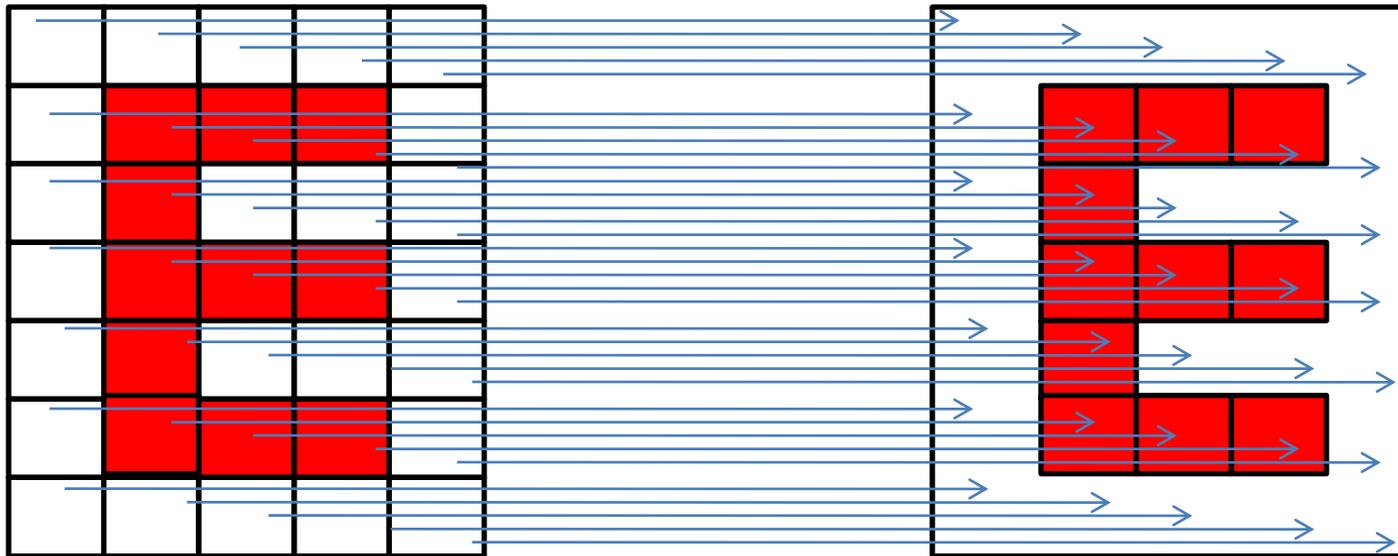
# Transmissão serial



# Exemplos de informação transmitida de modo serial

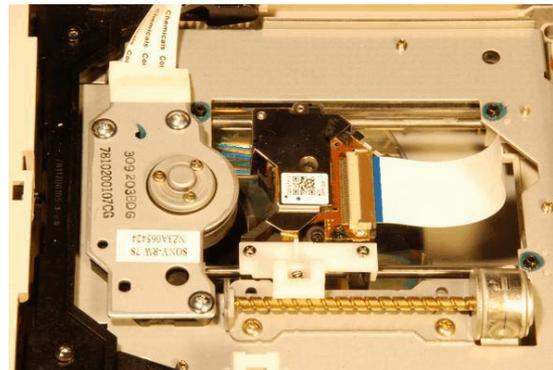
- Informação escrita
- Informação falada
- Código Morse
- Internet

# Transmissão paralela



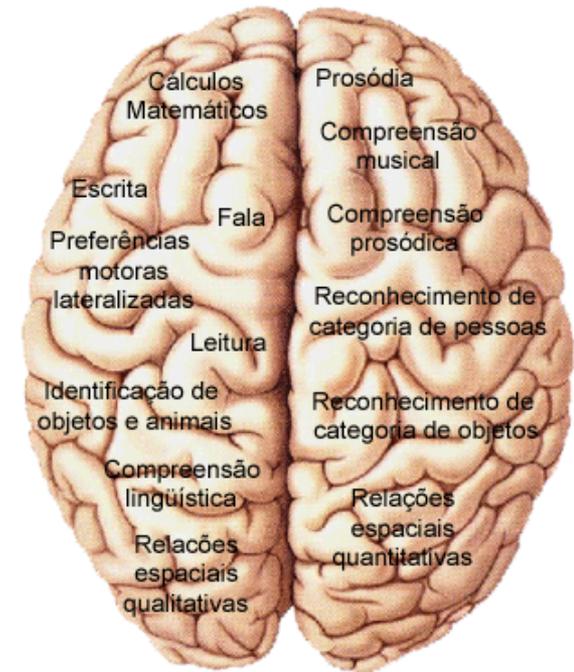
# Exemplos de informação transmitida em paralelo

- Sinais que viajam pelos barramentos do computador
- Sinais de trânsito
- Expressões faciais
- Marcas comerciais
- Símbolos
- Ideogramas



# No cérebro, cada hemisfério processa um tipo diferente de informação

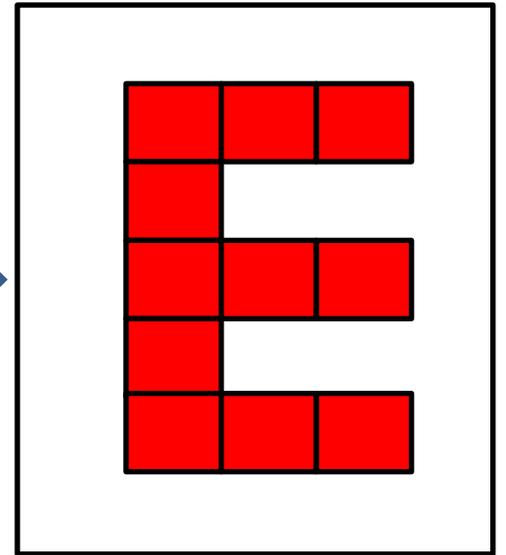
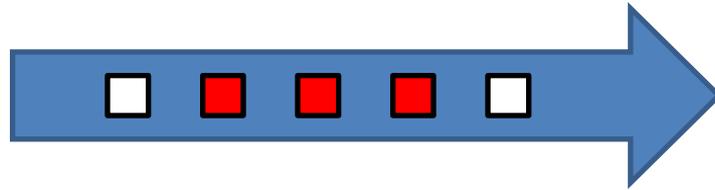
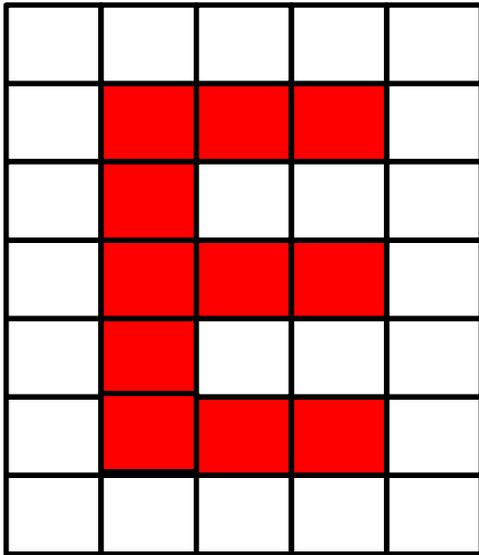
- A informação serial é processada no hemisfério esquerdo
- A informação paralela é processada no hemisfério direito.



# Parte 2



# Transmissão serial

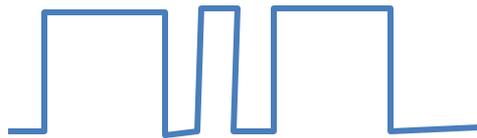


# Telegrafia: Código Morse



- Criado por samuel F.B. Morse em 1837
- As letras são representadas por uma sequência de pontos e linhas.
- Pontos e linhas são níveis altos de voltagem

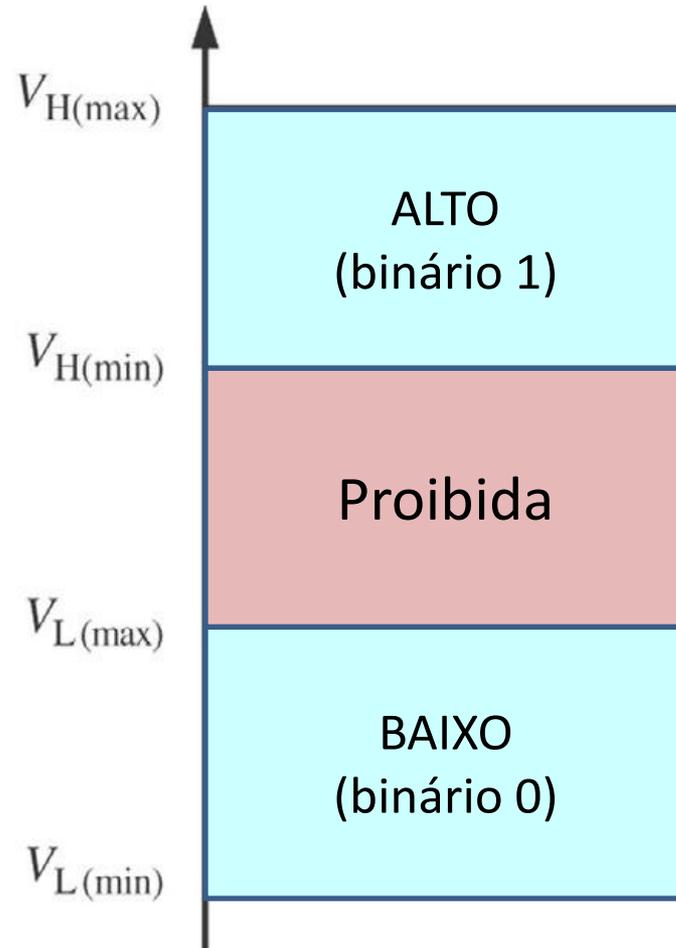
K



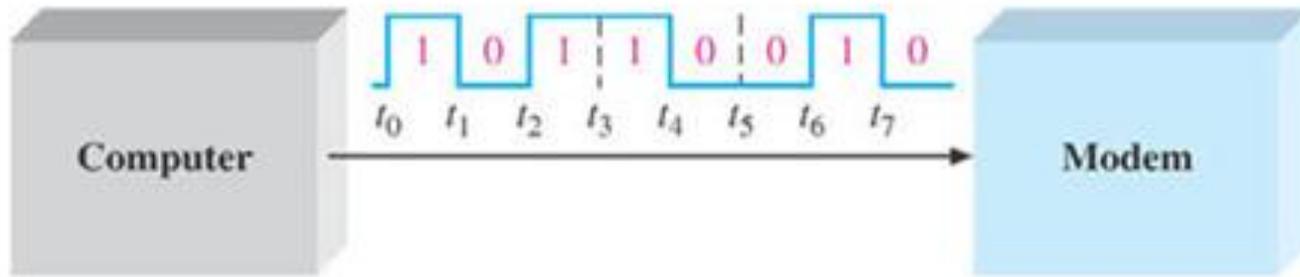
A	● —	V	● ● ● —
B	— ● ● ●	W	● — —
C	— ● — ●	X	— ● ● —
D	— ● ●	Y	— ● — —
E	●	Z	— — ● ●
F	● ● — ●	.	● — ● — ● —
G	— — ●	,	— — ● ● — —
H	● ● ● ●	?	● ● — — ● ●
I	● ●	/	— ● ● — ●
J	● — — —	@	● — — ● — ●
K	— ● —	1	● — — — —
L	● — — ● ●	2	● ● — — —
M	— —	3	● ● ● — —
N	— ●	4	● ● ● ● —
O	— — —	5	● ● ● ● ●
P	● — — ● ●	6	— ● ● ● ●
Q	— — — ● —	7	— — — ● ●
R	● — ●	8	— — — ● ●
S	● ● ●	9	— — — — ●
T	—	0	— — — — —
U	● ● —		

# Codificação binária nos computadores

Níveis lógicos  
de tensão  
para um  
circuito digital

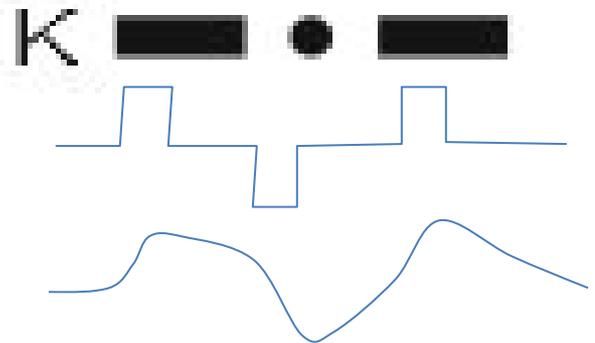
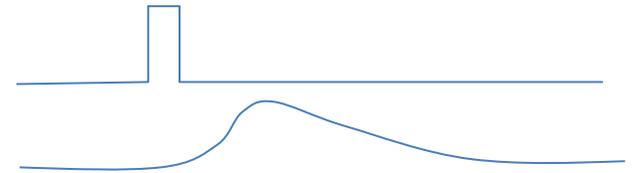


# Transferência de dados serial



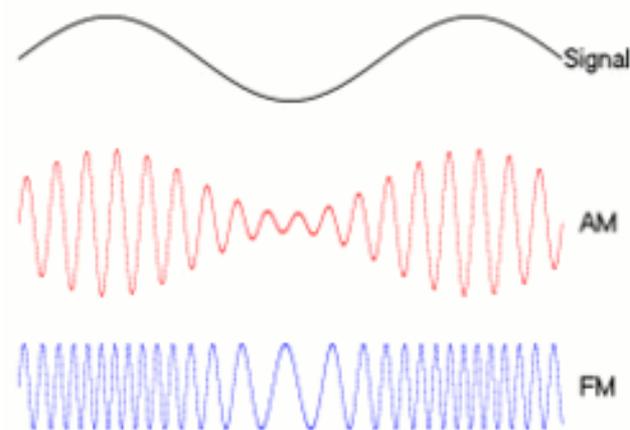
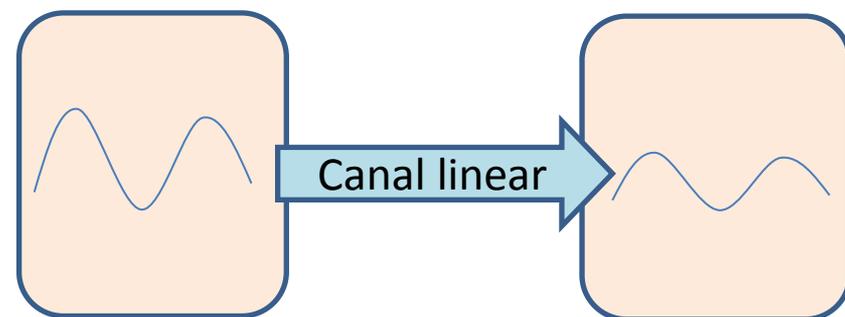
# Problemas da Códificação Morse e da codificação binária

- Ao longo do caminho a sinal vai se amortecendo e deformando.
- O que limita muito a velocidade de transmissão.
- Este problema foi resolvido em parte pela telegrafia de corrente dupla onde o traço era uma corrente elétrica numa direção, e o ponto corrente numa outra direção.
- Mesmo assim, o problema da atenuação continuava.



# Transmissão de informação e ondas senoidais

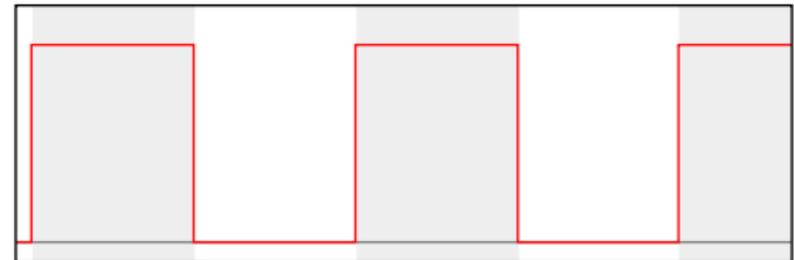
- Contudo, se a sinal for uma onda senoidal, esta conserva sua frequência com independência da distância (se o canal for linear como por exemplo um cabo elétrico ).



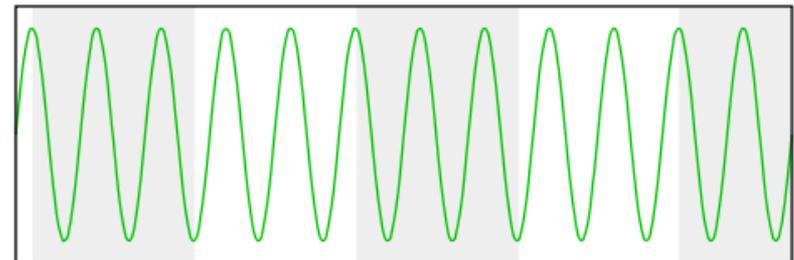
- Esta propriedade permite a transmissão via frequência modulada e a transmissão FSK

# Transmissão FSK

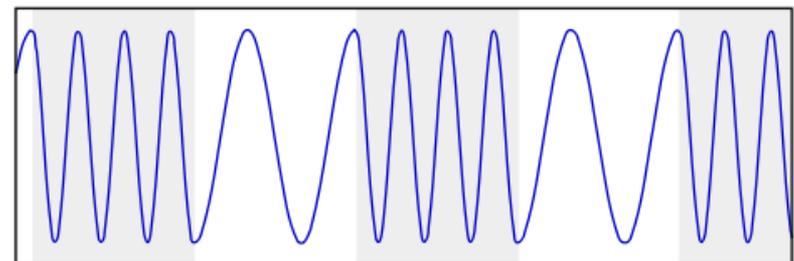
- Frequency-shift keying ou Modulação por chaveamento de frequência.
- Dígitos binários podem ser transmitidos por uma onda senoidal cuja frequência é diferente para o zero ou para o um.



Data



Carrier



Modulated Signal

# Parte 3



# Formas de corrigir os erros na transmissão serial de informação

- A transmissão de informação no computador utiliza a redundância (repetir várias vezes a mesma mensagem)
- Utilização de bits de paridade (par e ímpar)
- Códigos de correção de erros (código de Hamming).

# Causas dos erros

## ➤ Porque ocorrem os erros?

➤ Como vimos, temos diversos canais de comunicação que são **fisicamente diferentes** e conseqüentemente **diferentes tipos de erro** podem ocorrer:

- Um link de comunicação pode romper,
- A sinal se atenua ao longo de grandes distâncias.
- Uma linha telefônica pode ter muito ruído,
- Uma posição de memória (uma célula) pode falhar,
- Etc;

➤ No nosso estudo iremos considerar que independentemente da causa física, um **erro** é uma **mudança de um bit de um estado para outro estado** ( 0 para 1 ou 1 para 0)

# Erros e Códigos de Detecção

- Existem 2 formas de tratarmos um erro:
  - 1 – detectar e notificar o erro e o receptor tratará o erro,
  - 2 – recuperar o erro;
- Em ambos os casos informações adicionais ( bits adicionais ) são acrescentadas para informar o receptor como obter a informação original.
- Existem formas de informar se 2 conjuntos de bits são iguais ou não.

# Erros e Códigos de Detecção

- Uma definição muito usual é a Distancia de Hamming, que aponta quantos bits 1 conjunto de bits é diferente do outro.
- Ex: a seqüência 0110 e a seqüência 1110 estão separados pela distancia 1 de Hamming.
- Conseqüentemente, 2 conjuntos que estão separados pela distância de Hamming igual a zero são iguais;
- Para essa medida fazer sentido os conjuntos comparados têm que ter o mesmo número de bits;

# Erros e Códigos de Detecção

## ➤ Bit de Paridade

➤ Qualquer conjunto de bits possui um número par ou ímpar de 1's

➤ Paridade **par**: quando o número de 1's de uma seqüência de bits é par,

➤ Paridade **ímpar**: quando o número de 1's de uma seqüência de bits é ímpar;

➤ Para a detecção de erros é adicionado um bit a uma determinada seqüência de bits, o bit de paridade,

➤ Um sistema de comunicação só pode operar com um tipo de paridade,

# Erros e Códigos de Detecção

Exemplo:

PARIDADE PAR		PARIDADE IMPAR	
P	BCD	P	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

# Erros e Códigos de Detecção

- O bit de paridade pode ser adicionado no início ou no fim da seqüência de bits e isso depende somente do projeto do sistema;
- Um bit de paridade só consegue detectar um número ímpar de erros, isto é ,1 erro,3 erros numa seqüência mas não consegue detectar 2 erros;
  - Imagine um sistema com paridade par e transmitindo a seqüência: 0101
  - Com o bit de paridade ficaria 00101
  - Supondo que ocorra um erro no 3 bit o receptor receberá: 00001 e apontará um erro pois perceberá um número ímpar de 1's numa paridade par.
  - Agora, se ocorrerem 2 erros e o receptor receber: 01010 não perceberá o erro pois o número de 1's será par;

# Erros e Códigos de Detecção

Exercícios:

1. Determine o bit de paridade par aos seguintes grupos de códigos:

1010

111000

101101

1000111001001

101101011111

2. Um sistema de paridade impar recebe os seguintes grupos de códigos:

10110,11010,110011,11011110100 e

1100010101010. Estão todos corretos?

# Erros e Códigos de Detecção

- Vimos que com o código de detecção de paridade é possível detectar o erro mas não corrigi-lo,
- O código de Hamming permite que, além da detecção de um erro, seja possível especificar qual foi o bit no qual o erro ocorreu e corrigir o erro:

# Erros e Códigos de Detecção

1. Se o número de bits de dados projetados for  $d$  então o número de paridade  $p$  é dado por:  $2^p \geq d+p+1$  (1). Ex:

Se tivermos 4 bits podemos determinar  $p$  por tentativa e erro:

se  $p = 2$  temos  $2^2 = 4$  e  $d+p+1 = 7$  e portanto a equação 1 não se satisfaz,

se  $p = 3$  temos  $2^3 = 8$  e satisfaz a equação 1 portanto são necessários 3 bits de paridade para proporcionar a correção de um erro em uma seqüência de 4 bits;

# Erros e Códigos de Detecção

## 2. Inserção dos bits de paridade no código:

- No nosso exemplo temos 4 bits de informação e 3 de paridade,
- O bit mais a esquerda é o bit1, o próximo é o bit 2,...:  
*bit1, bit2, bit3, bit4, bit5, bit6, bit7*
- Os bits de paridade estão colocados nas posições correspondentes as potencias 1,2,4,8,... (neste exemplo, nas posições 1, 2 e 4)

$P_1, P_2, D_1, P_3, D_2, D_3, D_4$

# Erros e Códigos de Detecção

- Para determinar o valor dos bits de paridade, é necessário determinar a quais bits o bit de paridade está associado. Para isso montamos a seguinte tabela:

<b>Designação dos Bits</b>	<b>P<sub>1</sub></b>	<b>P<sub>2</sub></b>	<b>D<sub>1</sub></b>	<b>P<sub>3</sub></b>	<b>D<sub>2</sub></b>	<b>D<sub>3</sub></b>	<b>D<sub>4</sub></b>
<b>Posição do bits</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>Numero da posição em binário</b>	<b>001</b>	<b>010</b>	<b>011</b>	<b>100</b>	<b>101</b>	<b>110</b>	<b>111</b>
<b>Bits de dados dn</b>							
<b>Bits de Paridade Pn</b>							

Tabela de posicionamento de bits para um código de correção de erro de 7 bits

# Erros e Códigos de Detecção

- O número da posição do bit de paridade P2 tem 1 bit no meio. Devemos verificar todas as posições que têm 1 no meio (inclusive P2). Portanto, P2 verifica os bits das posições 2,3,6 e 7. O bit de paridade identifica a quantidade de 1's e, dependendo se foi definido paridade par ou ímpar ele é preenchido.

# Erros e Códigos de Detecção

➤ Exemplo: Determinar o código de Hamming para o número 1001 usando paridade par:

➤ Passo 1: determinar  $p$  – número de bits de paridade necessários:

$$2^p = 2^3 = 8$$

$$d+p+1 = 4+3+1 = 8$$

} **Atende a equação (1)**  
**portanto 3 bits de paridade são suficientes**

➤ Passo 2: Construir a tabela com os bits de paridade posicionados:

Designação dos Bits Posição do bits Numero da posição em binário	$P_1$ 1 001	$P_2$ 2 010	$D_1$ 3 011	$P_3$ 4 100	$D_2$ 5 101	$D_3$ 6 110	$D_4$ 7 111
Bits de dados $d_n$							
Bits de Paridade $P_n$							

# Erros e Códigos de Detecção

- Passo 3: Determine os bits de paridade:
  - bit  $P_1$  verifica os bits das posições 1,3,5 e 7 e tem que ser 0 para que o numero de 1's seja par nesse grupo,
  - bit  $P_2$  verifica os bits das posições 2,3,6 e 7 e tem que ser 0 para que o numero de 1's seja par nesse grupo,
  - bit  $P_3$  verifica os bits das posições 4,5,6 e 7 e tem que ser 1 para que o numero de 1's seja par nesse grupo,
- Passo 4: inserir os bits de paridade na tabela e obtenha o código que será enviado: 0011001

Designação dos Bits Posição do bits Numero da posição em binário	$P_1$ 1	$P_2$ 2	$D_1$ 3	$P_3$ 4	$D_2$ 5	$D_3$ 6	$D_4$ 7
	001	010	011	100	101	110	111
Bits de dados $d_n$			1		0	0	1
Bits de Paridade $P_n$	0	0		1			

# Detecção e correção de erro com o Código de Hamming

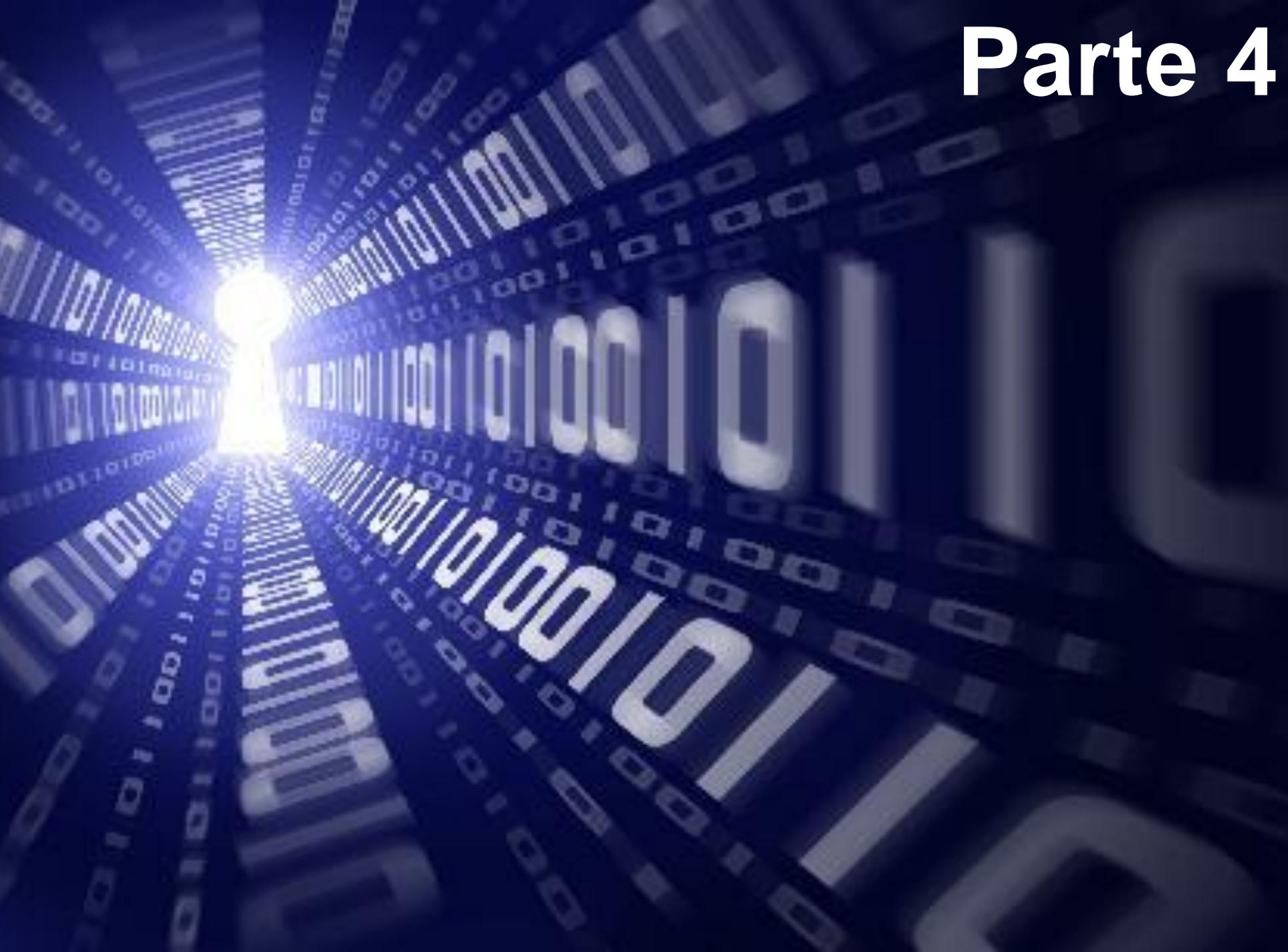
- Para a detecção do código de Hamming temos o seguinte procedimento:
  - Devem ser geradas tantas verificações quantos forem os bits de paridade. Se forem 3 bits de paridades, serão geradas 3 verificações e assim sucessivamente,
  - Passo 1: inicie começando por  $P_1$  e verifique o grupo quanto a paridade correta. Se for correta coloque 0 e se for incorreta coloque 1,
  - Passo 2: Repita o passo 1 para grupo de paridade,
  - Passo 3: O número obtido pelo resultado de todas as verificações determina a posição do bit errado. Esse é o *código de posição de erro* e se todas as verificações forem corretas não há erro. A primeira verificação gera o bit **menos** significativo.



# Códigos de erros

- Como vimos, a detecção de erros implica em um aumento de número de bits a serem transmitidos,
- Implica em um aumento de processamento na comunicação,
- Em algumas situações não é viável a verificação de erros em todas as transmissões. Exemplo:
  - As mensagens transmitidas na Internet pelo protocolo IP têm no cabeçalho da sua mensagem um campo para verificação de erro, o campo *checksum*. As estações e os roteadores da rede só verificam erro nos cabeçalhos dessas mensagens e deixam para as aplicações verificarem os erros, quando isso for necessário. Ex: transmissão de um programa (FTP).
  - E se os erros ocorrem no próprio campo de *checksum*?

# Parte 4

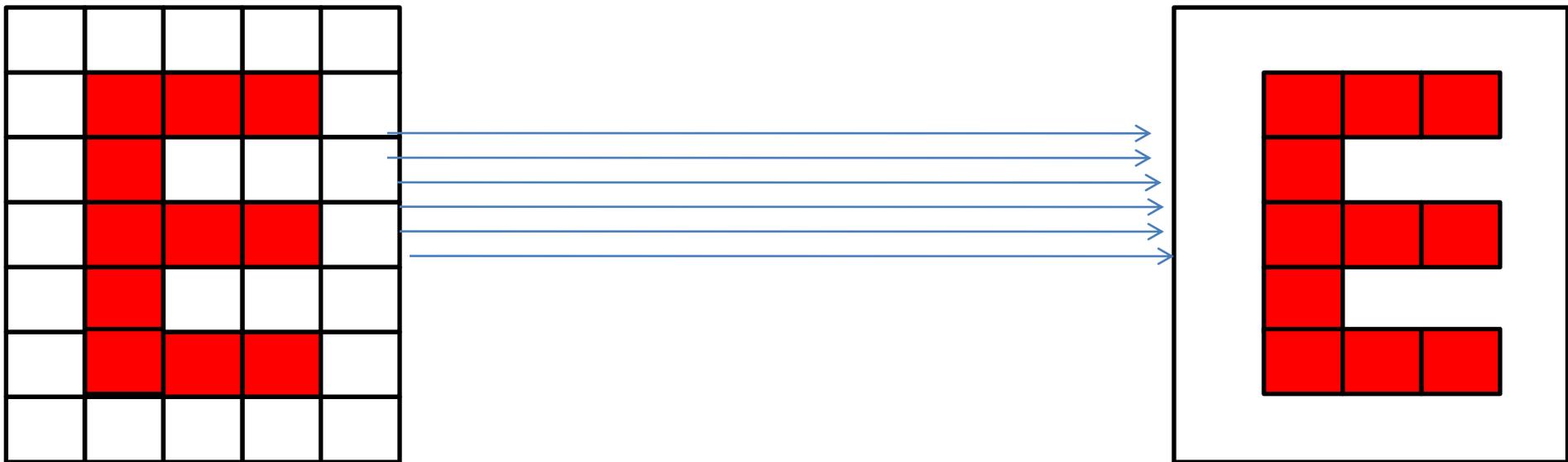


Brincadeira do baralho

# Brincadeira do baralho

- Ouros: Paula
  - Paus: Maria
  - Copas: Renata
  - Espadas: Shirley
- 1: Aguilar
  - 2: Antunes
  - 3: Camargo
  - 4: Dos santos
  - 5: Fernánderes
  - 6: Oliveira
  - 7: Ferreiro
  - 8: Souza
  - 9: Tanaka
  - 10: Taniguchi

# Transmissão paralela via séries de Fourier e componentes principais



Permite redução do número de canais por onde a informação é transmitida

# Transferência de dados paralela

Nos computadores um bit é transmitido por cada canal

