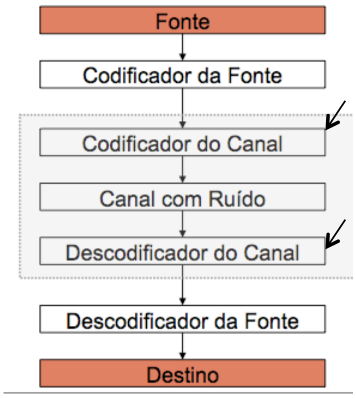
VII. CÓDIGOS PARA CONTROLO DE ERROS

OBJECTIVO:

- Construção de **códigos para controlo de erros**
- Abordar as bases matemáticas que permitem construir códigos (**codificação de canal**) para **controlar os erros** de transmissão em sistemas de telecomunicações não fiáveis ou ruidosos

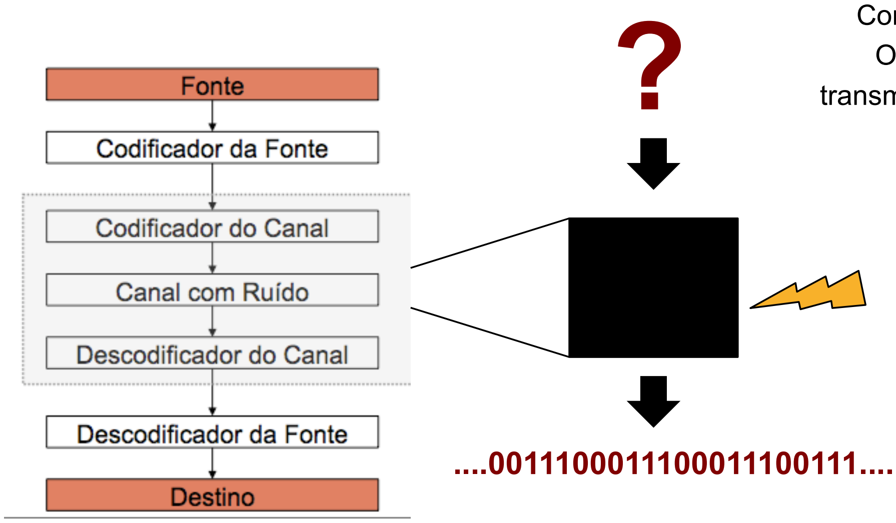


Considera-se somente o caso da **transmissão digital binária**

Técnicas utilizadas em várias tecnologias de comunicações ... (*e não só...*)

VII. CÓDIGOS PARA CONTROLO DE ERROS

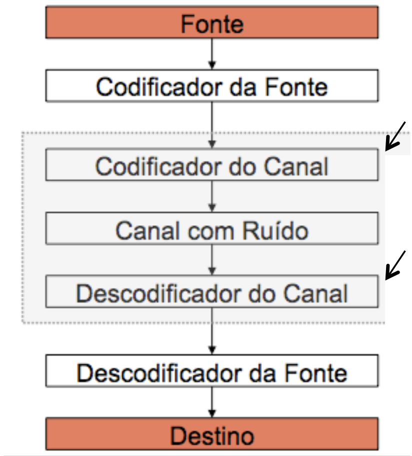
Como vimos (e.g. Cap. 3)
Os canais/sistemas de
transmissão não são "perfeitos"



Atenuação
Distorção
+
Ruído
Interferência
....
etc...

Dígitos/Bits estão corretos?

VII. CÓDIGOS PARA CONTROLO DE ERROS



Teoria da Informação

Dado um canal de comunicação e uma fonte cujo débito de informação não excede a capacidade do canal, existe um código tal que a informação pode ser transmitida através do canal com uma frequência de erros arbitrariamente pequena, apesar da presença de ruido.

VII. CÓDIGOS PARA CONTROLO DE ERROS

TIPOS DE ERROS

- Dois **tipos de ruído** que afectam as comunicações digitais:
  - **ruído branco:** erros de transmissão causados por este ruído são tais que o erro num determinado dígito não afecta os dígitos subsequentes (ocorrências de erros estatisticamente independentes, ou seja **erros aleatórios**)
  - **ruído impulsivo:** a sua presença caracteriza-se por longo intervalos de tempo em que os dígitos não são corrompidos, intercalados por molhos (***burts***) de dígitos corrompidos (ou seja, erros não são estatisticamente independentes)

VII. CÓDIGOS PARA CONTROLO DE ERROS

TIPOS DE ERROS

- Neste capítulo serão abordadas as bases para a construção de **códigos de correcção de erros aleatórios** ....

- .... embora, em termos de fundamentos, a base matemática é semelhante à usada nos códigos de correcção de erros aos "*molhos*"

VII. CÓDIGOS PARA CONTROLO DE ERROS

7 bits of data	byte with parity bit	
	even	odd
0000000	**0**0000000	**1**0000000
1010001	**1**1010001	**0**1010001
1101001	**0**1101001	**1**1101001
1111111	**1**1111111	**0**1111111

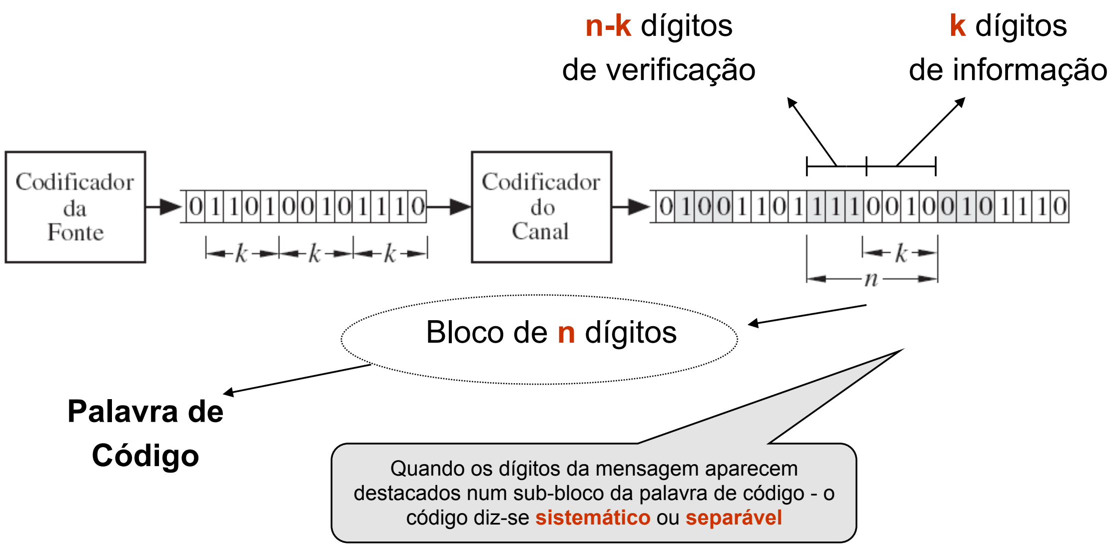
Exemplo de esquemas bastante simples - e.g. bit paridade - muito simples mas muito limitado

TIPOS DE CÓDIGOS

- Existem diferentes tipos de códigos para controlo de erros, iremos abordar:
  - **CÓDIGOS DE BLOCO:** cada conjunto de k dígitos de informação é acompanhado de n-k dígitos redundantes (dígitos de verificação de paridade) calculados a partir dos dígitos de informação, formando assim um bloco de tamanho fixo, de n dígitos, designada por palavra de código

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO (os mais usuais....)



VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

- Um bloco de dígitos de informação será um tuplo $D = (d_0\ d_1\ d_2\ \ldots\ d_{k-1})$ com $d_j \in \{0,1\}$, existem 2^k blocos de dígitos de informação ...
- ... cada um deles transformado numa palavra de código representada pelo tuplo $C = (c_0\ c_1\ c_2\ \ldots\ c_{n-1})$ com $c_j \in \{0,1\}$
- Haverá apenas 2^k palavras de código válidas distintas
- As restantes $2^n - 2^k$ palavras não fazem parte do dicionário do código; se forem recebidas é sinal da ocorrência de erro

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

- Códigos designados por códigos-(n,k) ou **C(n,k)**
- **Rendimento** de um código:

$$\rho = \frac{k}{n}$$

- Conceito **Distância de Hamming**

Definição 9.1 *Distância de Hamming entre duas palavras de um código de bloco, $d(C_i, C_j)$, é o número de posições em que as duas palavras, C_i e C_j, diferem.*

Fundamentos de Comunicação de Dados
Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

VII. CÓDIGOS PARA CONTROLO DE ERROS

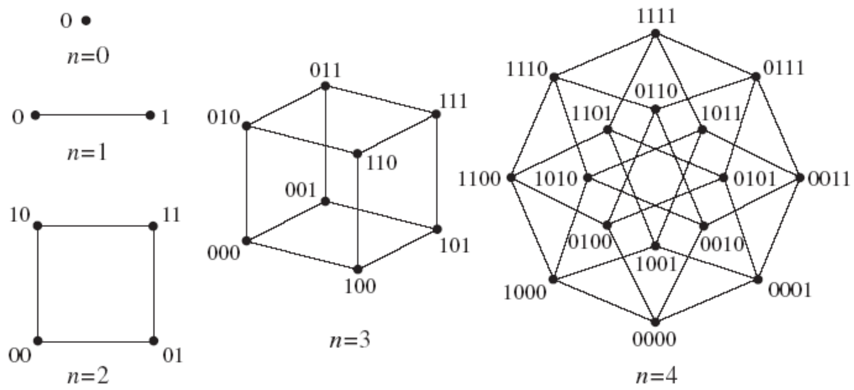
CÓDIGOS LINEARES DE BLOCO

- Conceito **Distância de Hamming**
  - Duas palavras de código idênticas estarão à distância zero...
  - Duas palavras de código distintas estarão a uma distância igual ou superior a uma unidade
  - ... O conceito de distância de *hamming* é passível de uma interpretação geométrica semelhante à distância euclideana entre dois pontos

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

- ... **interpretação geométrica** do conceito de distância de *Hamming* ... (correspondência entre 2^n palavras distintas de n dígitos vs 2^n vértices de um hipercubo num espaço de n dimensões)



VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

– Conceito de **distância mínima de um código**

Definição 9.2 *Distância mínima de um código de bloco, d_{min}, é a menor das distâncias de Hamming entre quaisquer duas palavras desse código.*

– A distância mínima de um código condiciona a sua **capacidade de control de erros** (tanto de detecção como de correcção)

– Quantos erros poderão ser detectados/corrigidos por um código com uma determinada distância mínima?

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

– Exemplo: código com distância mínima 2?
pode detectar-se um único erro ... mas não se pode corrigir o erro
– E para um Código com distância mínima igual a 3?

Seja d_{min} a distância mínima de um código,

Para detectar até e_d erros:	$d_{min} = e_d + 1$
Para corrigir até e_c erros:	$d_{min} = 2e_c + 1$

– ... um código que corrige e_c erros pode ser alternativamente usado como um código detector de $e_d = 2\ e_c$ erros

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

Algumas propriedades / teoremas associados a códigos lineares de blocos

Definição 9.3 *Peso de uma palavra C_i de um código de bloco, $p(C_i)$, é o número de dígitos 1 que a palavra C_i contém.*

Definição 9.4 *Peso mínimo de um código de bloco, $[p(C_i)]_{min}$ é o peso da palavra de menor peso desse código, exceptuando a palavra de peso zero.*

Teorema 9.1 — Distância mínima
A distância mínima de um código de bloco é igual ao seu peso mínimo.

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS LINEARES DE BLOCO

– Existem vários tipos de códigos.... exemplo:

Codigos de hamming

- C(n,k) - verificam a relação

$$\boxed{n = 2^{n-k} - 1}$$

- códigos correctores de erros simples / detectores de erros duplos

Fundamentos de Comunicação de Dados
Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS CÍCLICOS BINÁRIOS

– São uma **sub-classe dos códigos lineares de bloco** sendo **fáceis de realizar** (estrutura matemática simples)
– Nestes códigos utiliza-se uma **representação polinomial**
– operações são realizadas em **aritmética módulo 2**
– A partir de uma palavra de código é possível obter outras

Definição 9.5 *Um código linear de bloco $C(n,k)$ é cíclico se possuir a seguinte propriedade:*

> *Se o tuplo $C = (c_0, c_1, c_2, \ldots, c_{n-1})$ fôr uma palavra de código então o tuplo $C^{(1)} = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$ obtido por deslocação cíclica direita de uma posição de C também é uma palavra de código.*

VII. CÓDIGOS PARA CONTROLO DE ERROS

GERAÇÃO DE CÓDIGOS CÍCLICOS C(n,k)

- Utilização de um polinómio gerador, **g(x)**
- g(x) é usado para gerar o código (n,k) - (g(x) é de grau n-k e divide o polinomio $x^n + 1$)
- Códigos podem ser gerados de duas formas:
  - originando palavras de código em que os dígitos de informação e de verificação estão misturados (códigos **criptográficos**)
  - ou, de **forma sistemática**, em que os dígitos de verificação e de informação aparecem separados

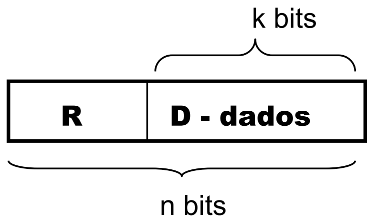
Vamos analisar em detalhe os segundos - **códigos cíclicos sistemáticos** -

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS CÍCLICOS SISTEMÁTICOS C(n,k)

- Vai-se adoptar as seguinte sintaxe para as palavras de código (ver sebenta) :

Palavra de Código (n,k) :



$$C = (\underbrace{r_0, r_1, r_2, \ldots, r_{n-k-1}}_{\substack{n-k \text{ dígitos} \\ \text{de verificação} \\ \text{de paridade}}}, \underbrace{d_0, d_1, d_2, \ldots, d_{k-1}}_{\substack{k \text{ dígitos} \\ \text{da mensagem}}})$$

a sintaxe adoptada influencia a forma como se representa (sob forma de polinómios) os dados (D(x)), os dígitos de verificação (R(x)) e a palavra de código (C(x))

de que forma são gerados os dígitos de verificação?

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS CÍCLICOS SISTEMÁTICOS C(n,k)

$$C = (\underbrace{r_0, r_1, r_2, \ldots, r_{n-k-1}}_{\substack{n-k \text{ dígitos} \\ \text{de verificação} \\ \text{de paridade}}}, \underbrace{d_0, d_1, d_2, \ldots, d_{k-1}}_{\substack{k \text{ dígitos} \\ \text{da mensagem}}})$$

$$D(x) = d_0 + d_1 x + d_2 x^2 + \ldots + d_{k-1} x^{k-1}$$

$$r(x) = r_0 + r_1 x + r_2 x^2 + \ldots + r_{n-k-1} x^{n-k-1}$$

$$\boxed{r(x) \text{ é o resto da divisão de } x^{n-k} D(x) \text{ por } g(x)}$$

em aritmética módulo 2

Fundamentos de Comunicação de Dados
Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS CÍCLICOS SISTEMÁTICOS C(n,k)

Exemplo:

Seja g(x) = 1 + x + x^3 o polinómio gerador de um cíclico (7,4). Determinar a palavra de código (sistemática) correspondente à mensagem (dados) D = (1110).

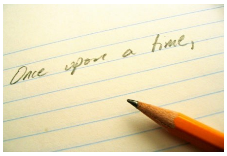
$$\boxed{r(x) \text{ é o resto da divisão de } x^{n-k} D(x) \text{ por } g(x)}$$

> D(x) = 1 + x + x^2
> x^{n-k} D(x) = x^3 D(x) = $x^3 + x^4 + x^5$
> calcular r(x) = ?
> Palavra de código?

$$C = (\underbrace{0\,1\,0}_{r(x)}\ \underbrace{1\,1\,1\,0}_{D(x)})$$

$$\begin{array}{rl|l}
x^5 + x^4 + x^3 & & x^3 + x + 1 \\
\hline
x^5 + \ x^3 + x^2 & & x^2 + x \\
\hline
0 + x^4 + 0 + x^2 & & \\
x^4 + \ x^2 + x & & \\
\hline
0 + \ 0 + x & = r(x) &
\end{array}$$

$$C = (r_0, r_1, r_2, \ldots, r_{n-k-1}, d_0, d_1, d_2, \ldots, d_{k-1})$$



?

Considere que g(x) = 1+x+x⁴ é o polinómio gerador de um código ciclico sistemático (15,11) utilizado para comunicação num canal de transmissão.

A palavra de código correspondente aos dados D = (11101000000) é C = (001111101000000) ?

$$\boxed{r(x) \text{ é o resto da divisão de } x^{n-k}D(x) \text{ por } g(x)}$$

$$C = (\underbrace{r_0, r_1, r_2, \ldots, r_{n-k-1}}_{\substack{n-k \text{ dígitos} \\ \text{de verificação} \\ \text{de paridade}}}, \underbrace{d_0, d_1, d_2, \ldots, d_{k-1}}_{\substack{k \text{ dígitos} \\ \text{da mensagem}}})$$

VII. CÓDIGOS PARA CONTROLO DE ERROS

EXEMPLO ...

Tabela 9.1: Código cíclico (7, 4) gerado por $g(x) = 1 + x + x^3$

Informação $D(x)$	Código criptográfico $C(x) = D(x) \cdot g(x)$	Código sistemático $C(x) = r(x) + x^{n-k}D(x)$	Peso $p(C_i)$
0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0
0 0 0 1	0 0 0 1 1 0 1	1 0 1 0 0 0 1	3
0 0 1 0	0 0 1 1 0 1 0	1 1 1 0 0 1 0	4
0 0 1 1	0 0 1 0 1 1 1	0 1 0 0 0 1 1	3
0 1 0 0	0 1 1 0 1 0 0	0 1 1 0 1 0 0	3
0 1 0 1	0 1 1 1 0 0 1	1 1 0 0 1 0 1	4
0 1 1 0	0 1 0 1 1 1 0	1 0 0 0 1 1 0	3
0 1 1 1	0 1 0 0 0 1 1	0 0 1 0 1 1 1	4
1 0 0 0	1 1 0 1 0 0 0	1 1 0 1 0 0 0	3
1 0 0 1	1 1 0 0 1 0 1	0 1 1 1 0 0 1	4
1 0 1 0	1 1 1 0 0 1 0	0 0 1 1 0 1 0	3
1 0 1 1	1 1 1 1 1 1 1	1 0 0 1 0 1 1	4
1 1 0 0	1 0 1 1 1 0 0	1 0 1 1 1 0 0	4
1 1 0 1	1 0 1 0 0 0 1	0 0 0 1 1 0 1	3
1 1 1 0	1 0 0 0 1 1 0	0 1 0 1 1 1 0	4
1 1 1 1	1 0 0 1 0 1 1	1 1 1 1 1 1 1	7

VII. CÓDIGOS PARA CONTROLO DE ERROS

CÓDIGOS CÍCLICOS C(n,k)

Tabela 9.1: Código cíclico (7, 4) gerado por $g(x) = 1 + x + x^3$

Informação $D(x)$	Código criptográfico $C(x) = D(x) \cdot g(x)$	Código sistemático $C(x) = r(x) + x^{n-k}D(x)$	Peso $p(C_i)$
0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0
0 0 0 1	0 0 0 1 1 0 1	1 0 1 0 0 0 1	3
0 0 1 0	0 0 1 1 0 1 0	1 1 1 0 0 1 0	4
0 0 1 1	0 0 1 0 1 1 1	0 1 0 0 0 1 1	3
0 1 0 0	0 1 1 0 1 0 0	0 1 1 0 1 0 0	3
0 1 0 1	0 1 1 1 0 0 1	1 1 0 0 1 0 1	4
0 1 1 0	0 1 0 1 1 1 0	1 0 0 0 1 1 0	3
0 1 1 1	0 1 0 0 0 1 1	0 0 1 0 1 1 1	4
1 0 0 0	1 1 0 1 0 0 0	1 1 0 1 0 0 0	3
1 0 0 1	1 1 0 0 1 0 1	0 1 1 1 0 0 1	4
1 0 1 0	1 1 1 0 0 1 0	0 0 1 1 0 1 0	3
1 0 1 1	1 1 1 1 1 1 1	1 0 0 1 0 1 1	4
1 1 0 0	1 0 1 1 1 0 0	1 0 1 1 1 0 0	4
1 1 0 1	1 0 1 0 0 0 1	0 0 0 1 1 0 1	3
1 1 1 0	1 0 0 0 1 1 0	0 1 0 1 1 1 0	4
1 1 1 1	1 0 0 1 0 1 1	1 1 1 1 1 1 1	7

Seja d_{min} a distância mínima de um código,

Para detectar até e_d erros:	$d_{min} = e_d + 1$
Para corrigir até e_c erros:	$d_{min} = 2e_c + 1$

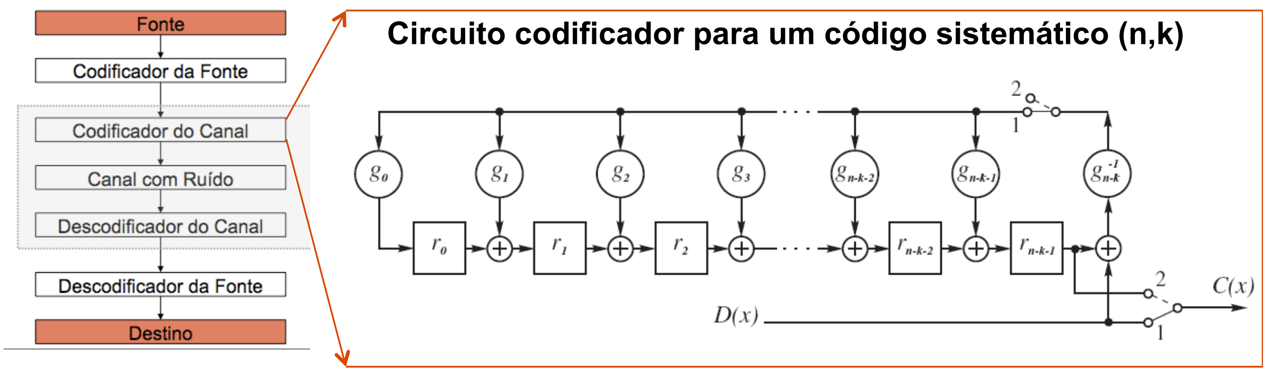
- mesmo conjunto de palavras em ambas as codificações
- possível obter palavras de código por deslocação cíclica
- nos códigos sistemáticos há uma separação visível entre os dígitos de informação e verificação

distância mínima? **?**
capacidade de correcção / detecção?

Fundamentos de Comunicação de Dados
Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

VII. CÓDIGOS PARA CONTROLO DE ERROS

GERAÇÃO DE CÓDIGOS CÍCLICOS SISTEMÁTICOS



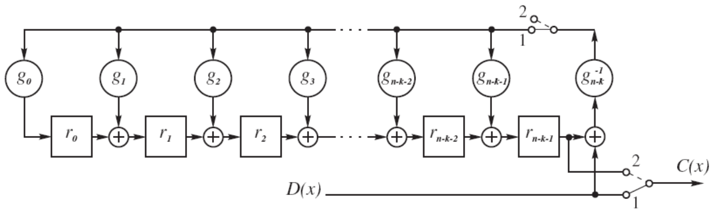
O circuito contém:

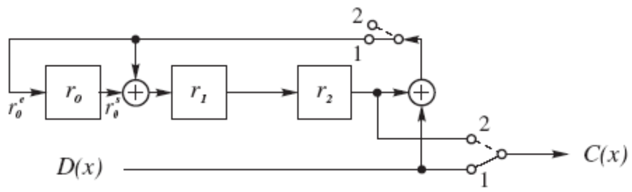
- registos para *n-k* bits (dígitos de verificação)
- conjunto de ou-exclusivos
- conjunto de ligações abertas ou fechadas conforme os coeficientes do polinómio g(x)

VII. CÓDIGOS PARA CONTROLO DE ERROS

EXEMPLO

Esquematize um circuito codificador para um código sistemático (7,4) com $g(x) = 1 + x + x^3$



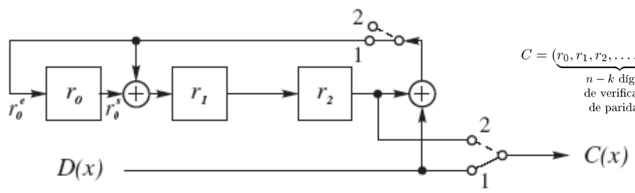


Fundamentos de Comunicação de Dados
Licenciatura em Engenharia Informática
Departamento de Informática, Universidade do Minho

VII. CÓDIGOS PARA CONTROLO DE ERROS

EXEMPLO

Verificar a operação do circuito utilizando a palavra de dados D = (0101)



$$C = (\underbrace{r_0, r_1, r_2, \ldots, r_{n-k-1}}_{\substack{n-k \text{ dígitos} \\ \text{de verificação} \\ \text{de paridade}}}, \underbrace{d_0, d_1, d_2, \ldots, d_{k-1}}_{\substack{k \text{ dígitos} \\ \text{da mensagem}}})$$

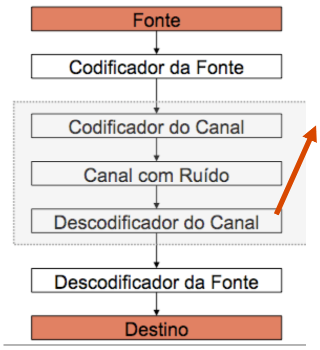
bit de entrada	entrada nos registos				saída dos registos		
$D(x)$	r_0^e	r_1^e	r_2^e		r_0^s	r_1^s	r_2^s
—	0	0	0		0	0	0
1	1	1	0	→	1	1	0
0	0	1	1	→	0	1	1
1	0	0	1	→	0	0	1
0	1	1	0	→	1	1	0

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

VII. CÓDIGOS PARA CONTROLO DE ERROS

SÍNDROMA

- As palavras de código, C(x), são transmitidas através do canal
- No caso de ocorrência de erro(s) a palavra que chega ao descodificador, R(x), poderá permitir saber qual a palavra transmitida



- O descodificador divide R(x) por g(x) obtendo um resto S(x) (designado por **síndroma** de R(x))
- Se S(x)=0 o receptor toma a palavra como válida
- Se S(x) ≠ 0 o receptor assume então que houve erro e pode (ou não, se for só detector) tentar corrigir a palavra recorrendo a circuitos específicos e à informação presente em S(x)

VII. CÓDIGOS PARA CONTROLO DE ERROS

Exemplo de Circuitos para Detecção / Correcção (breve referência)

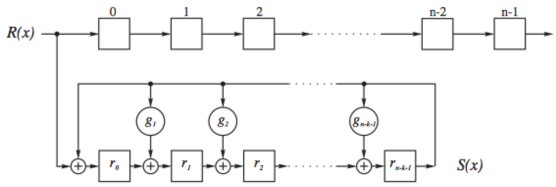


Figura 9.6: Divisão de $R(x)$ por $g(x)$ no descodificador

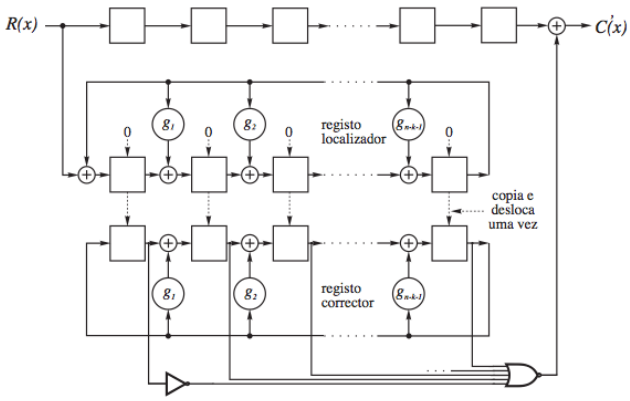


Figura 9.7: Circuito corrector de erros simples

VII. CÓDIGOS PARA CONTROLO DE ERROS

EXEMPLOS DE ALGUNS CÓDIGOS ....

- Nem todos os polinómios geradores são capazes de gerar um bom código
- Procura de códigos "*bons*" – para um dado valor de *n* e rendimento *k/n* encontrar aqueles códigos que possuem maior distância mínima, ou seja códigos com maior capacidade de detecção / correcção de erros
- Exemplos de alguns códigos conhecidos…. Diferenças?

Tipo	n	k	ρ	d_{min}	$g(x)$
códigos de Hamming	7	4	0.57	3	x^3+x+1
	15	11	0.73	3	x^4+x+1
	31	26	0.84	3	x^5+x^2+1
códigos BCH	15	7	0.46	5	$x^8+x^7+x^6+x^4+1$
	31	21	0.68	5	$x^{10}+x^9+x^8+x^6+x^5+x^3+1$
	63	45	0.71	7	$x^{18}+x^{17}+x^{16}+x^{15}+x^9+x^7+$ $+x^6+x^3+x^2+x+1$
código Golay	23	12	0.52	7	$x^{11}+x^9+x^7+x^6+x^5+x+1$

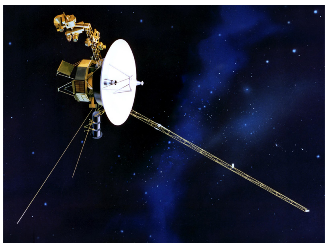
VII. CÓDIGOS PARA CONTROLO DE ERROS

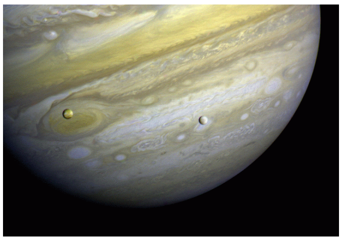
EXEMPLOS DE ALGUNS CÓDIGOS ....

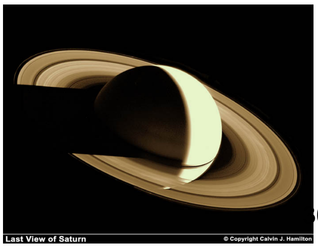
Curiosidade:

NASA - Voyager 1 e 2 1979/1980
Na transmissão de imagens a cores de Jupiter, Saturno foi usado um código *"parecido"* com este

Tipo	n	k	ρ	d_{min}	$g(x)$
códigos de Hamming	7	4	0.57	3	x^3+x+1
	15	11	0.73	3	x^4+x+1
	31	26	0.84	3	x^5+x^2+1
códigos BCH	15	7	0.46	5	$x^8+x^7+x^6+x^4+1$
	31	21	0.68	5	$x^{10}+x^9+x^8+x^6+x^5+x^3+1$
	63	45	0.71	7	$x^{18}+x^{17}+x^{16}+x^{15}+x^9+x^7+$ $+x^6+x^3+x^2+x+1$
código Golay	23	12	0.52	7	$x^{11}+x^9+x^7+x^6+x^5+x+1$





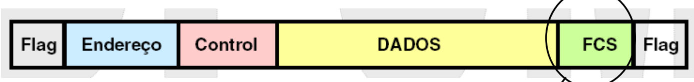


VII. CÓDIGOS PARA CONTROLO DE ERROS

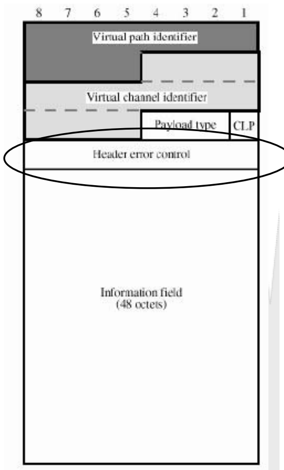
EXEMPLOS DE UTILIZAÇÃO DE MECANISMOS DE CONTROLO DE ERROS ...

não necessariamente do mesmo tipo dos códigos abordados neste capítulo

HDLC

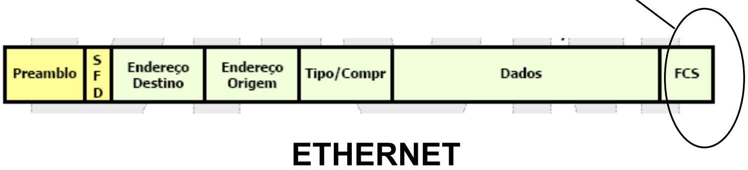


ATM



(b) Network-Network Interface

Diferentes tipos de códigos são utilizados por diferentes tecnologias; normalmente é referido o polinómio gerador utilizado



ETHERNET

VII. CÓDIGOS PARA CONTROLO DE ERROS

TÉCNICAS DE CORRECÇÃO DE ERROS

Forward Error Correction (FEC)

- Correção de erros progressiva, quando os códigos para controlo de erros são **utilizados como correctores**
  - pouco usadas em sistemas de transmissão de dados.... a não ser em condições especiais
- Usado em <u>canais simplex</u> onde não é possível a retransmissão (ou é impraticável)
- Cenários em que o <u>tempo de propagação é muito elevado</u> (e.g. comunicação com sondas espaciais, ...)
- Técnicas também usadas em gravações digitais (CD, DVD, ...), memórias *flash*, *hard drives* ....

VII. CÓDIGOS PARA CONTROLO DE ERROS

TÉCNICAS DE CORRECÇÃO DE ERROS

– ***Automatic Repeat Request*** (ARQ)

- Código **usado só como detector**
- Correcção processa-se por repetição (pedido de retransmissão das palavras)
- Necessário um canal de comunicação duplex
- Técnicas utilizadas nos sistemas/tecnologias de transmissões de dados mais comuns
- Técnicas ARQ - Tópico expandido e coberto em detalhe noutra UC (*Redes de Computadores*)

Gralha - pág. 241

$$= (1 + x) \cdot (1 + x + x) = 1 + x + x + x + x + x$$
$$= 1 + x + x^2 + x^5$$

dado que $x^3 + x^3 = (1 + 1) \cdot x^3 = 0 \cdot x^3 = 0$. *Portanto a palavra de código é* $C = (1110010)$. *Podem obter-se outras palavras do código por deslocação cíclica desta. A segunda coluna da tabela 9.1 lista o código completo assim calculado.*

b) Na forma sistemática os três primeiros dígitos são os de verificação e os últimos quatro são os da mensagem. Os dígitos de verificação são os coeficientes do polinómio $r(x)$ *que é o resto da divisão de* $x^{n-k}D(x)$ *por* $g(x)$*, isto é,*

$$\frac{x^{n-k}D(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)}$$

Considere-se uma sequência qualquer de mensagem, por exêmplo $D = (1110)$*, a que corresponde* $D(x) = \cancel{1 + x^2 + x^3}$. *Como* $n - k = 7 - 4 = 3$*, tem-se* $x^3D(x) = x^3 + x^4 + x^5$ *e executando a divisão polinomial:*

deve ler-se $D(x) = 1 + x + x^2$