

PCS 3115 (PCS2215)

Sistemas Digitais I

Módulo – Códigos

Para Detecção e Correção de Erros

Prof. Dr. Edison Spina

sobre o material do Prof. Dr. Marcos A. Simplicio Jr.

versão: 4.1 (Fev/2018)

Spina

Detecção de erros

- **Erro:** alteração de um ou mais bits de um dado
 - Ex.: Interferência eletromagnética durante comunicação
 - Ex.: Eletricidade estática alterando dados armazenados
- **Pergunta:** como detectar esses erros?
 - Resposta: detecção de erros possível se nem todas as palavras do código são válidas!
 - Usar mais bits do que número mínimo necessário para representar todos os dados possíveis (**redundância**)
 - Ex.: código 1-de-10 tem 10 palavras válidas dentre 1024
- **Problema:** projete um código otimizado para:
 - Representar 4 símbolos diferentes (ex.: A, T, C, G), e
 - Detectar qualquer erro simples (i.e., em apenas 1 bit)

Spina

Cubos

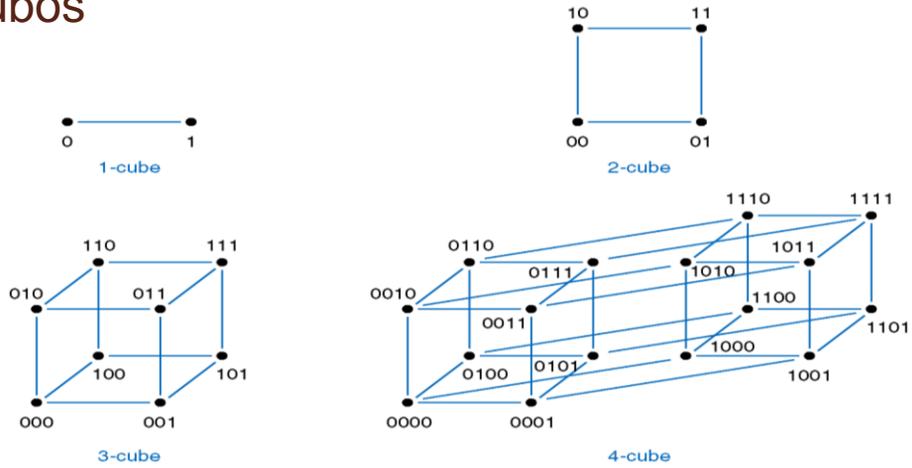
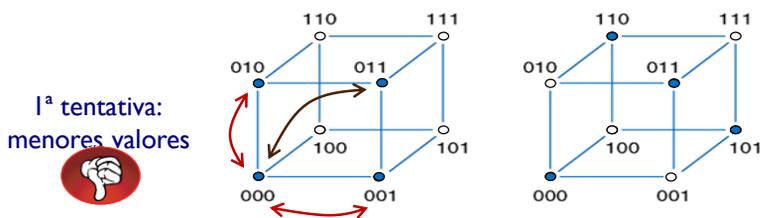


Figure 2-8
n-cubes for n = 1, 2, 3, and 4.

From *Digital Design: Principles and Practices*, Fourth Edition, John F. Wakerly. ISBN 0-13-186389-4. ©2006, Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

Detecção de erros

- **Problema:** projete um código otimizado para:
 - Representar 4 símbolos diferentes
 - Pelo menos 2 bits
 - Detectar qualquer erro simples (i.e., em apenas 1 bit) → quantos bits adicionais...?
 - Aumentando um bit ... (dobra a quantidade de representações possíveis!)
 - Palavras possíveis (3 bits): 000, 001, 010, 011, 100, 101, 110, 111
 - Quais palavras definir como válidas/inválidas...?

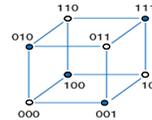
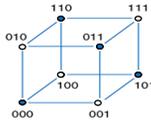


2ª tentativa:
valores distantes por
pelo menos 2 vértices

Detecção de erros: um pouco de teoria

- Código com **distância** mínima m : quaisquer pares de palavras do código diferem em pelo menos m bits
 - Conseguem detectar erros em até $d = m - 1$ bits!
- Para detectar todos os **erros de 1 bit**: $m = 2$
 - Para n bits de informação, é necessário um código de $n+1$ bits: bit adicional denominado **“bit de paridade”**
 - **Paridade par**: bit adicional é 1 se isso faz com que palavra tenha **número par de bits 1**; caso contrário, esse bit é 0
 - **Paridade ímpar**: bit adicional é 1 se isso se isso faz com que palavra tenha **número ímpar de bits 1**; caso contrário, esse bit é 0
 - Também permite detectar erros em **número ímpar de bits**

Paridade par

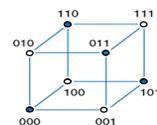


Paridade ímpar

Correção de erros

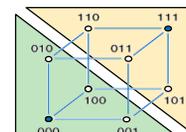
- Como corrigir (não apenas detectar!) erros?
 - Ideia: corrigir uma palavra inválida para a palavra válida mais próxima dela!
- É possível corrigir erros de 1 bit com um código de distância 2?
 - Não: palavras inválidas são equidistantes das palavras válidas

$000 \xleftarrow{?} 010 \xrightarrow{?} 110$



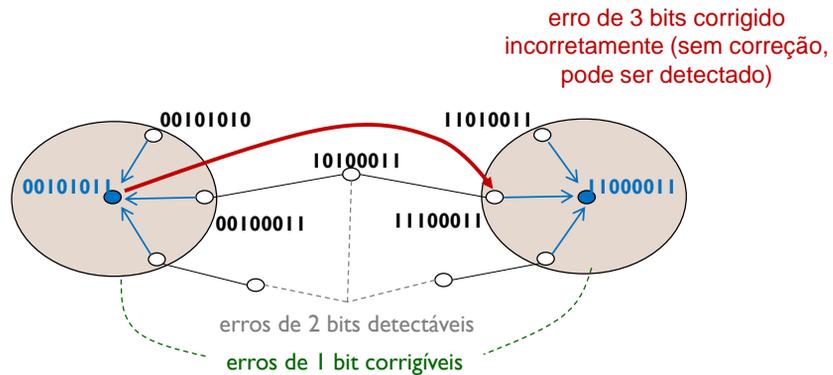
- E se a distância for 3?

- Sim:
 - 010, 100, 001 \rightarrow 000
 - 110, 011, 101 \rightarrow 111



Correção de erros: um pouco de teoria

- Código com distância mínima $m = 2c+d+1$: corrige até c erros e é capaz de detectar $c+d$ erros
 - Ex.: $m = 4$, configurado para corrigir erros ($c = 1, d = 1$) ou apenas para detectar erros ($c = 0, d = 3$)



Lição de Casa

- Leitura Obrigatória:
 - Capítulo 2 do Livro Texto.
- Exercícios Obrigatórios:
 - Capítulo 2 do Livro Texto;
 - Lista de Exercícios do Módulo 4.

Livro Texto

- Wakerly, J.F.; *Digital Design – Principles & Practices*; Fourth Edition, ISBN: 0-13-186389-4, Pearson & Prentice-Hall, Upper Saddle, River, New Jersey, 07458, 2006.

Bibliografia Adicional

- Giozza, William Ferreira; et all; *Redes Locais de Computadores: Tecnologia e Aplicações – Seção 3.2.3 Codificação em Banda Básica – Códigos*; Editora McGraw-Hill, 1.986;
- Hayes, J.P.; *Computer Architecture and Organization*; McGraw-Hill, 1988;