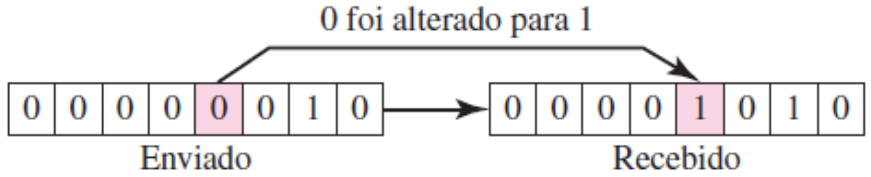
Roteiro – Unidade I

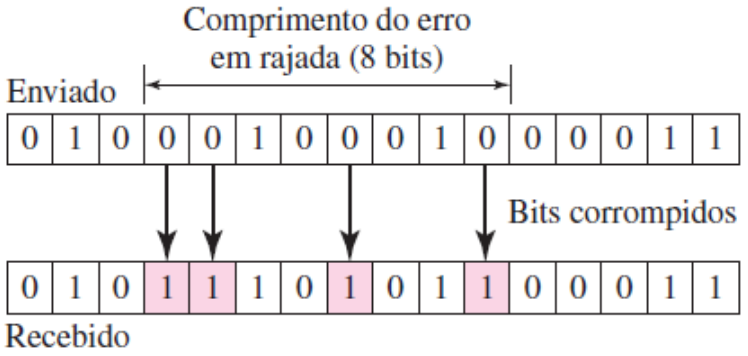
- Introdução às redes de computadores
  - Modelos de camadas OSI e TCP/IP
- Interligação de redes
  - Comutação de circuitos e de pacotes
  - Meios físicos de transmissão
  - Equipamentos e topologias de rede
- Comunicação de dados
  - Modelos de comunicação
  - Controle de acesso ao meio de comunicação
  - **Técnicas de correção e detecção de erros**

Introdução

- Apesar dos algoritmos de controle de acesso ao meio, ainda é possível que ocorram erros nas transmissões dos dados em canais compartilhados
  - Colisão
  - Interferências externas
- Os erros podem ocorrer em um único bit:



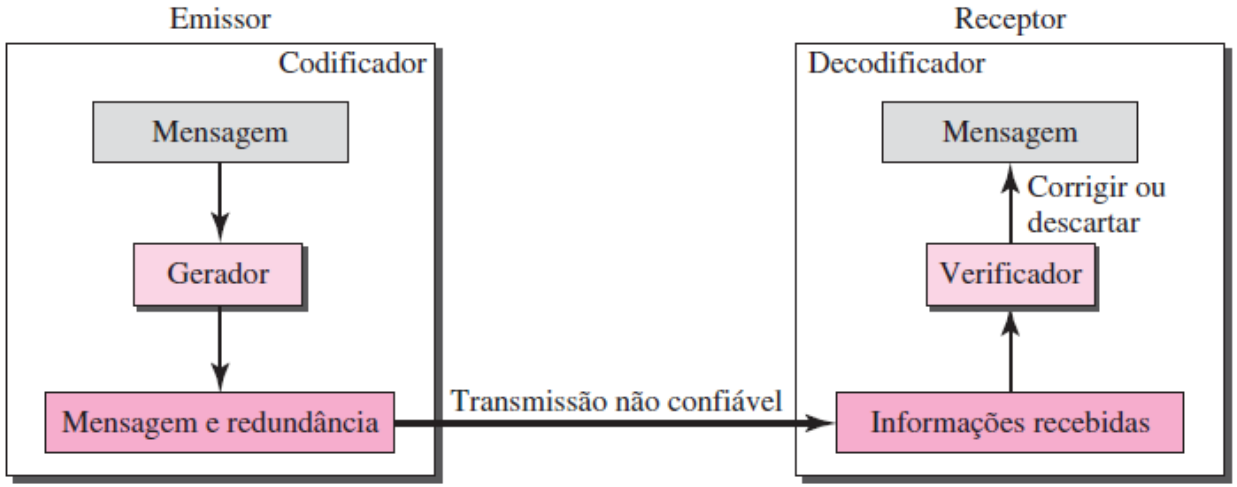
- Ou em rajada (mais de um bit):



Introdução

- Mas como os erros são detectados?
  - Cada quadro recebe no final um campo de informação: FCS - *Frame Check Sequence*
    - Transmissor calcula e inclui o FCS no quadro que será enviado:

    - Ao receber o quadro, o receptor calcula o FCS e o compara com FCS enviado pelo transmissor
    - Se iguais, transmissão com sucesso
    - Se diferentes, transmissão com erro
      - Quadro recebido é descartado
    - O cálculo do FCS se dá por meio de algoritmos de detecção de erros:
    - Bit de paridade
    - Soma de verificação (*checksum*)
    - Verificação de redundância cíclica (CRC)

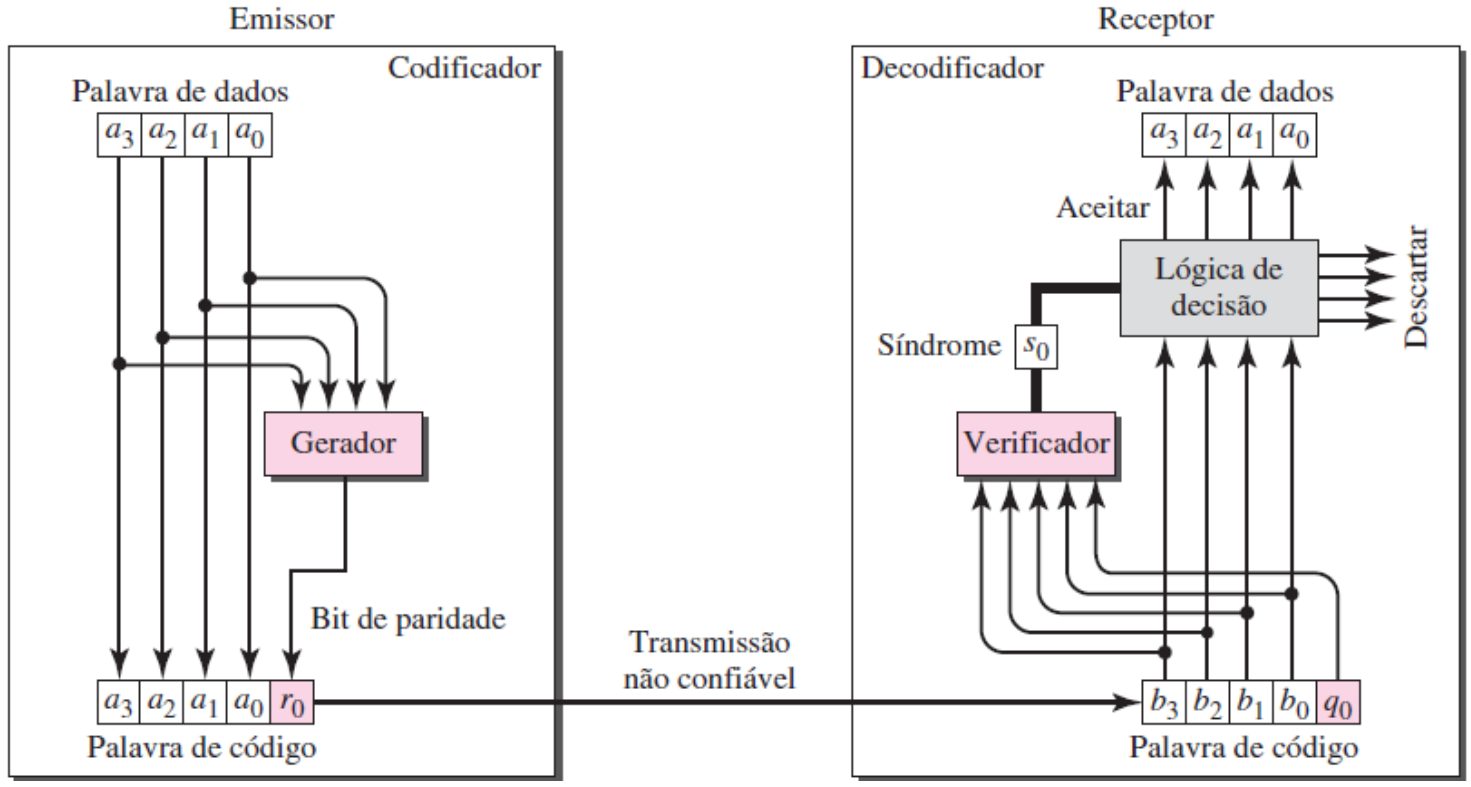


Técnicas de detecção de erros

- Bit de paridade
  - A ideia do bit de paridade é fazer com que cada fragmento do quadro tenha um número par ou ímpar de bits ‘1’ - ou seja, há dois tipos de paridade
    - Paridade ímpar:
      - Se houver um número ímpar de bits 1 no quadro, o bit de paridade é ‘0’
      - Se houver um número par de bits 1 no quadro, o bit de paridade é ‘1’
    - Paridade par:
      - Se houver um número ímpar de bits 1 no quadro, o bit de paridade é ‘1’
      - Se houver um número par de bits 1 no quadro, o bit de paridade é ‘0’
  - Exemplo: a sequência de bits 1011010 apresenta:
    - Paridade par: 10110100
    - Paridade ímpar: 10110101
  - Este método funciona bem para erros em apenas 1 bit (eficiência de 100% na detecção)
  - Entretanto, apresenta falhas para erros em rajada (eficiência de 50% na detecção)

Técnicas de detecção de erros

- Bit de paridade (funcionamento)
  - O emissor envia o quadro com o bit de paridade incluso
  - O receptor verifica se a paridade está correta, comparando com o que foi recebido
  - O resultado da verificação, denominado síndrome, é de apenas 1 bit
    - É 0 (zero) quando o número de 1s no quadro recebido for par; caso contrário, é 1 (um).



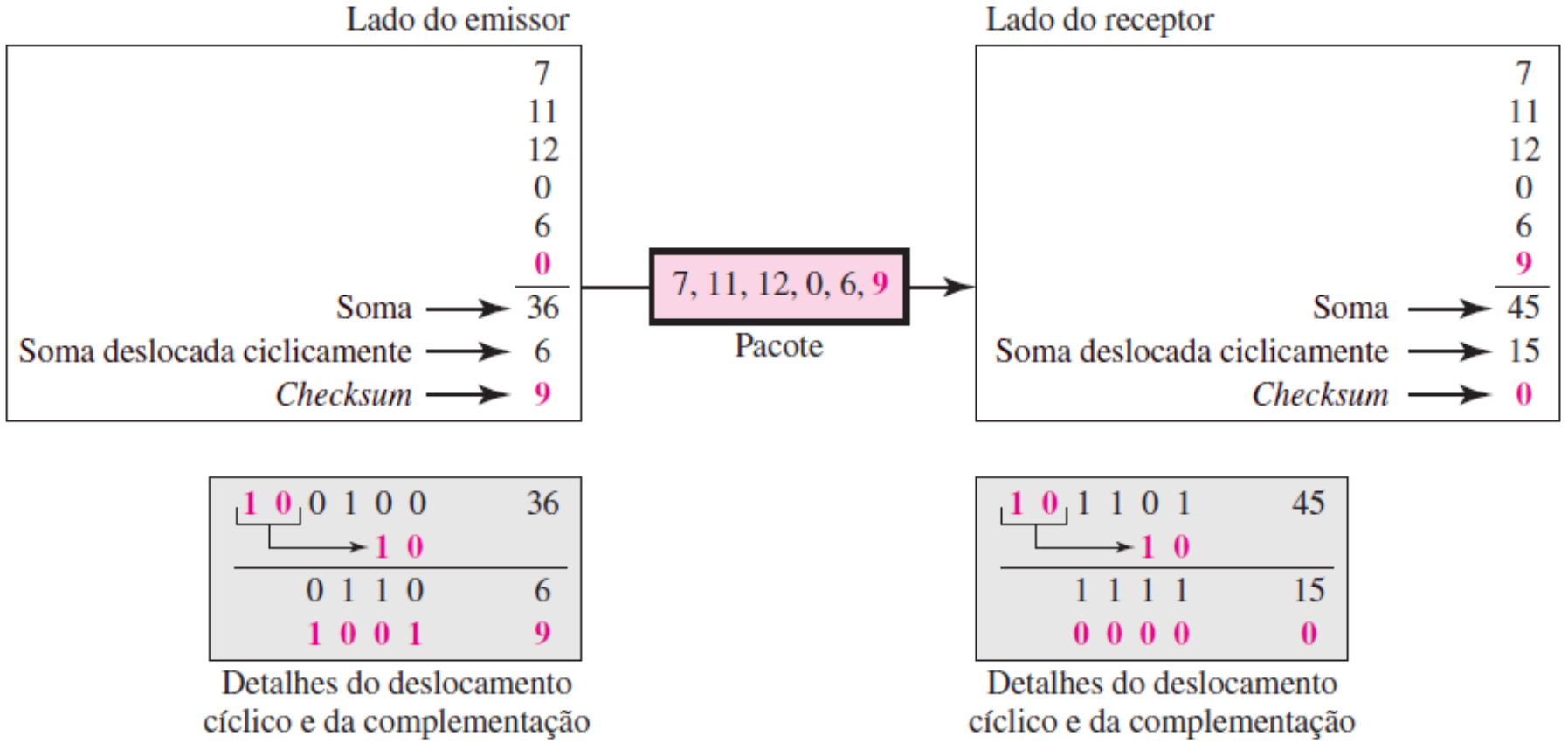
Técnicas de detecção de erros

- Soma de verificação (*checksum*)
  - Consiste em transmitir todas as mensagens juntamente com o resultado da soma dos bits delas (*checksum*)
  - Quando o receptor receber as mensagens, calcula o *checksum* para verificar se há erros
  - Funcionamento:
    - No lado do emissor:
      - As mensagens são divididas em palavras menores (de acordo com o sistema em uso)*
      - O valor do checksum é iniciado em zero
      - Todos os bits das mensagens, incluindo o *checksum*, são somados usando a aritmética complemento de um
      - A soma é computada e enviada juntamento com as mensagens
    - No lado do receptor:
      - As mensagens recebidas também são divididas em palavras menores (de acordo com o sistema em uso)*
      - Todos os bits das mensagens são somados com a mesma artimética anterior
      - A soma é computada e verificada:
        - Caso seja 0 (zero) a mensagem foi recebida sem erros, caso contrário há erros (descarta)

* Na Internet utilizam-se palavras de 16 bits, por exemplo

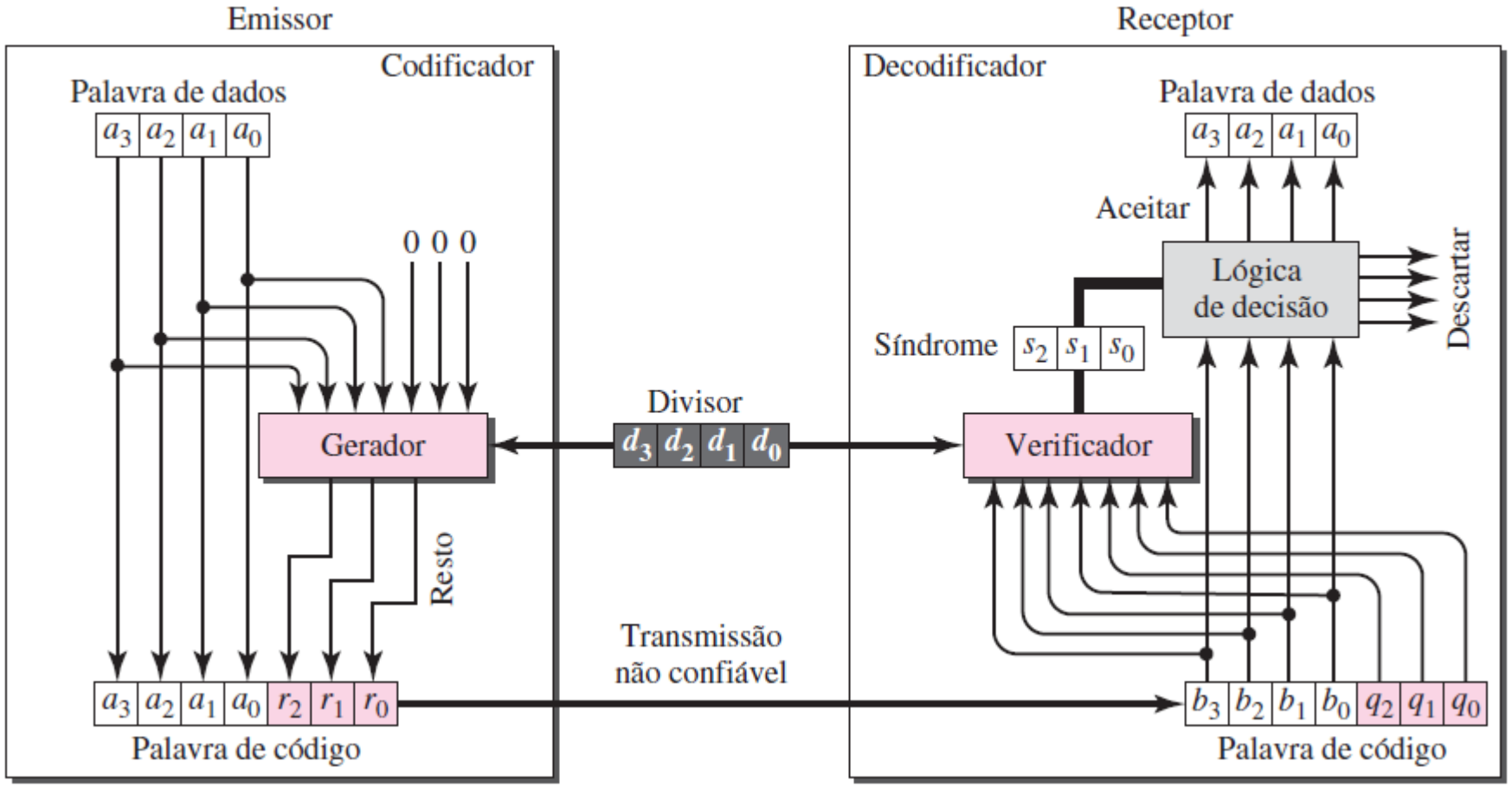
Técnicas de detecção de erros

- Soma de verificação (*checksum*)
  - Exemplo: envio de 5 mensagens de 4 bits cada



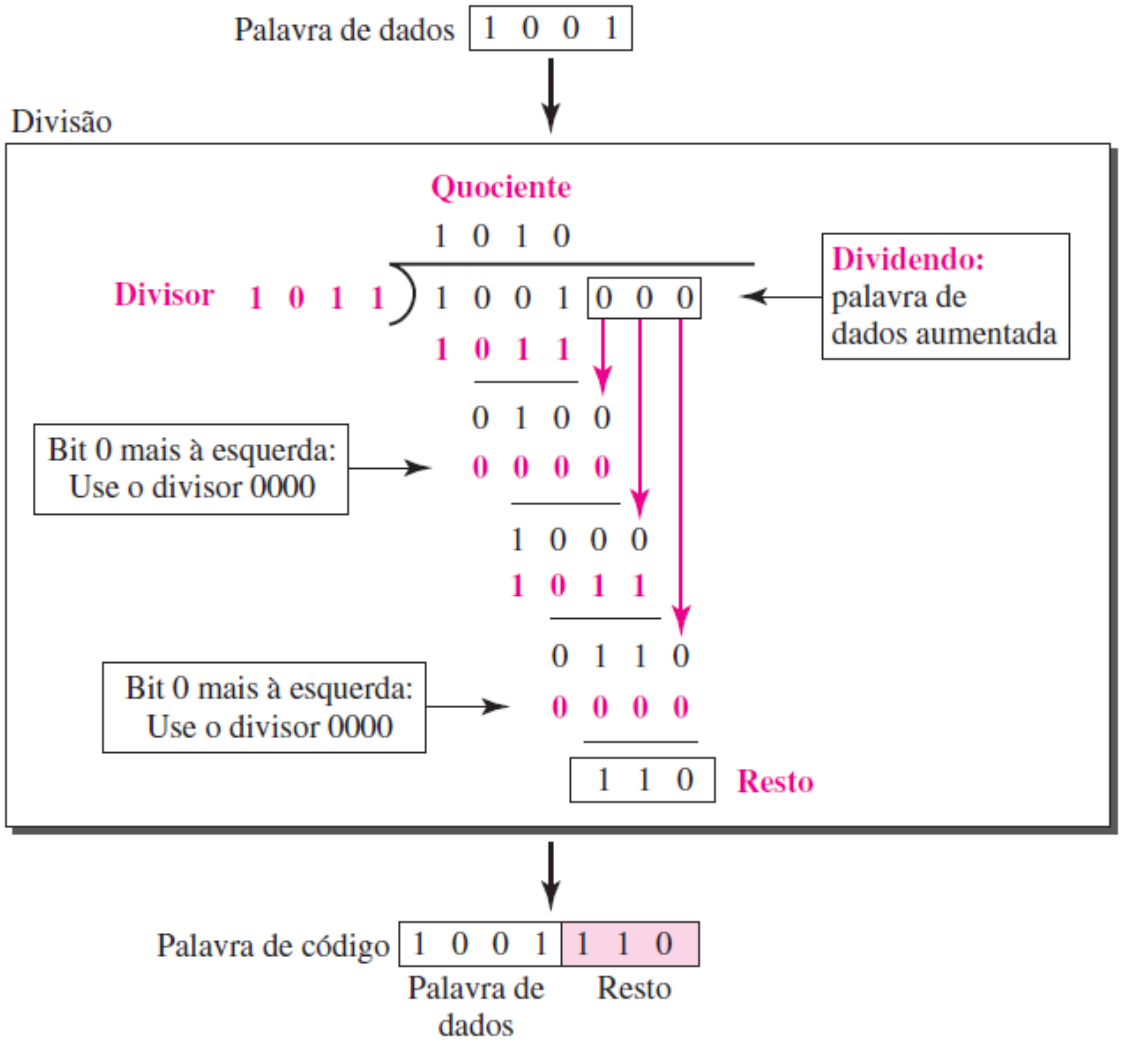
Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)



Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)
  - No codificador:
    - A palavra de dados tem k bits e a palavra de código tem n bits
    - O tamanho da palavra de dados é aumentado adicionando-se (n-k) 0s (zeros) ao lado direito da palavra original e repassado ao gerador
    - O gerador usa um divisor de tamanho (n-k)+1, predefinido e estabelecido por ambas as partes
    - O gerador divide a palavra de dados aumentada pelo divisor (divisão de módulo 2)
    - O quociente da divisão é descartado; o resto ($r_2 r_1 r_0$) é anexado à palavra de dados para criar a palavra de código

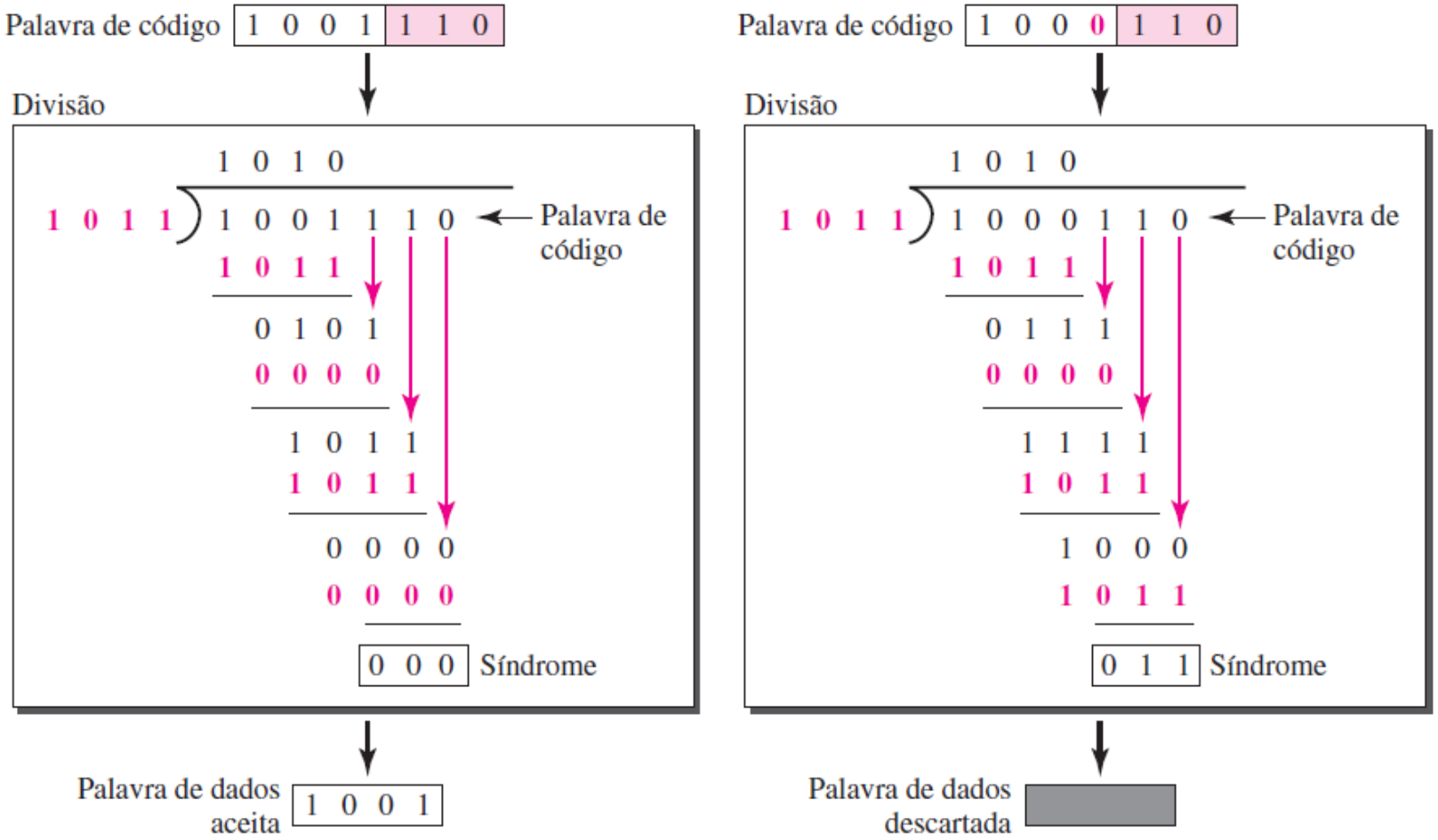


Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)
  - No decodificador:
    - Recebe a palavra de código possivelmente corrompida
    - Uma cópia de todos os n bits é alimentada no verificador, que é uma réplica do gerador
    - O resto produzido pelo verificador é uma síndrome de (n-k) bits que alimenta o analisador lógico de decisão
    - Se os bits de síndrome forem todos 0s, os k bits mais à esquerda da palavra de código são aceitos como palavras de dados (interpretado como não sendo um erro); caso contrário, os k bits são descartados (erro).

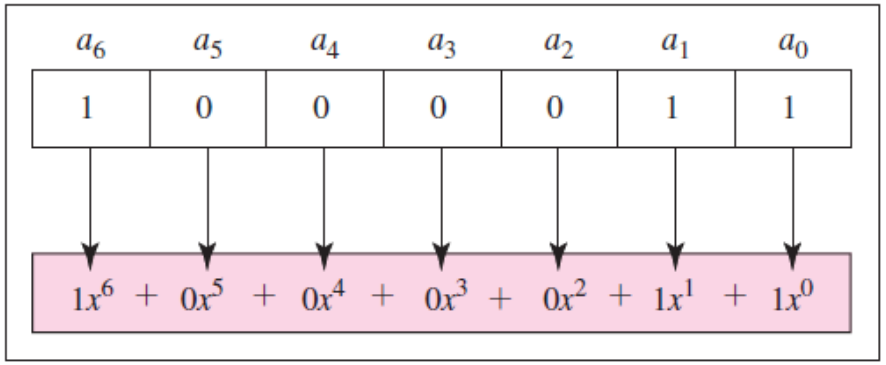
Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)
  - No decodificador:

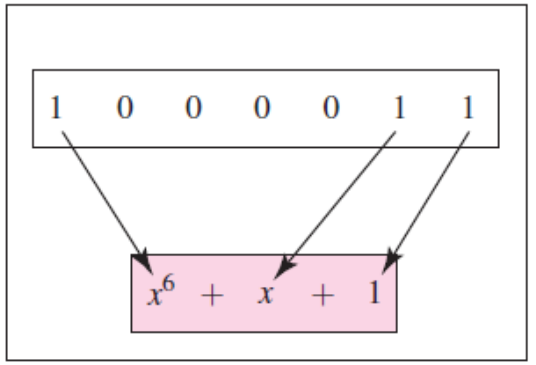


Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)
  - CRC também pode ser visto como polinômios
  - Consiste em um quadro de n bits representado por um polinômio P em x de ordem n-1.



a. Padrão binário e polinômio



b. Forma reduzida

- Exemplo: Quadro com 8 bits (11100010), polinômio de ordem 7:
  - Polinômio gerado: $1x^7 + 1x^6 + 1x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 0x^0$
  - Polinômio final: $x^7 + x^6 + x^5 + x^1$

Técnicas de detecção de erros

- Código de Redundância Cíclica (CRC)
  - Polinômios geradores mais comuns:
    - CRC-8:
      - Exemplo:

        Sequência de bits: 100000111 (9 bits)

        Polinômio: $x^8 + x^2 + x + 1$

    - CRC-12:
      - Exemplo:

        Sequência de bits: 1100000001101 (13 bits)

        Polinômio: $x^{12} + x^{11} + x^3 + x^2 + x + 1$

    - CRC-32:
      - Exemplo:

        Sequência de bits: 10000010011000001000111010110111 (33 bits)

        Polinômio: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Técnicas de correção de erros

- A correção de erros é basicamente efetuada retransmitindo os quadros que tiveram erros
- Tem como objetivo garantir transmissões confiáveis
  - Utiliza mensagens de reconhecimento (confirmação) de quadros que pode ser:
    - Positivo: o quadro chegou corretamente (ACK)
    - Negativo: o quadro chegou com erro e deve ser retransmitido (NAK)
  - Além disso, são utilizados temporizadores de retransmissão para evitar que o emissor fique esperando indefinidamente pela mensagens de reconhecimento
    - O transmissor utiliza um intervalo de tempo para que o reconhecimento do quadro chegue
  - Existem algoritmos para determinar a forma de retransmissão das mensagens:
    - *Stop-and-wait*: só envia o próximo quadro quando o anterior for confirmado
    - Janela deslizante: vários quadros podem ser enviados em sequência sem ter que aguardar pelo reconhecimento de cada um. Entretanto, se após um intervalo de tempo alguma confirmação falhar, aquele quadro não confirmado será retransmitido

Exercícios de fixação

1. Como são detectados os erros nas transmissões?
2. Quais os mecanismos para verificação de erros?
3. Como são corrigidos os erros?