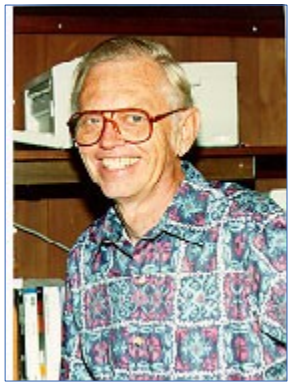
CRC – *Cyclic Redundancy Check*

TÓPICOS

- Introdução
- CRC – *Cyclic Redundancy Check*
- Conclusão

INTRODUÇÃO



- **História**
  - William Wesley Peterson e D. T. Brown;
  - W. W. Peterson e D. T. Brown, "*Cyclic Codes for Error Detection*", *Proceedings of the* IRE, vol. 49, p. 228-235, Jan 1961;
- **Objetivo**
  - Identificar a ocorrência de erros ocasionados por ruídos durante o processo de transmissão (controle de erros);
- **Ruído**
  - Fenômeno aleatório capaz de perturbar a transmissão de dados;
  - EletroMagnectic Interference (EMI);
  - RadioFrequency Interference (RFI).

CRC – *Cyclic Redundancy Check*

- Produz um valor expresso em poucos *bits* que é anexado à mensagem original:
  - CRC-64 – 64 *bits*;
  - CRC-32 – 32 *bits*;
  - CRC-16 – 16 *bits*.

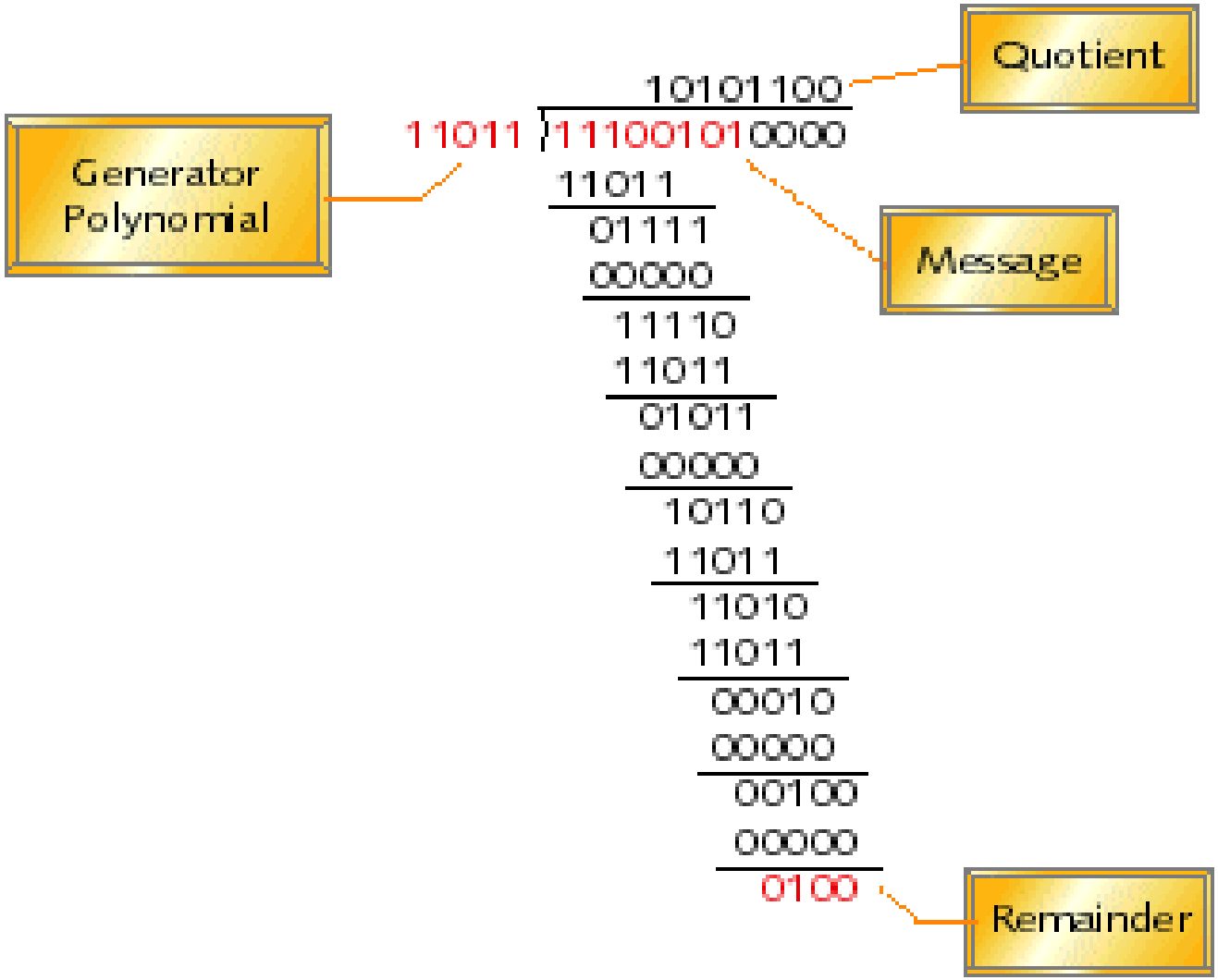
- Propriedades:
  - Todos os bits da mensagem são utilizados no cálculo do valor do CRC;
  - Mudança de um único *bit* é refletida no valor do CRC;
  - Probabilidade de valores uniforme.

CRC – *Cyclic Redundancy Check*

- **Cálculo do CRC:**
  - **Resto da divisão polinomial (divisão módulo 2) entre os dados e o polinômio gerador. Exemplo.**
    - Polinômio gerador de grau 16 ($x^{16} + x^{15} + \ldots\ldots + x^2 + x^1 + x^0$)
    - Resto de grau 15 (16 bits)

- **Divisão Módulo 2:**
  - **Acrescentar, à direita da mensagem, uma quantidade de zeros equivalente ao grau do polinômio gerador;**
  - **A partir do *bit* mais significativo da mensagem**
    - *Bit* 1
      - Acrescentar o *bit* 1 ao quociente;
      - Aplicar um XOR entre o divisor e o polinômio gerador;
    - *Bit* 0
      - Acrescentar o *bit* 0 ao quociente;
      - Aplicar um XOR entre o divisor e zeros;
  - **Efetuar uma rotação à esquerda (excluir *bit* + a esquerda).**

CRC – *Cyclic Redundancy Check*



CRC – *Cyclic Redundancy Check*

- **Verificação de Erros:**
  - **Dividir a mensagem recebida pelo polinômio gerador e analisar o resto**
    - Resto = 0 => mensagem correta;
    - Resto ≠ 0 => mensagem com erro.
  - **Separar a mensagem recebida do CRC**
    - Acrescentar N (grau do polinômio gerador) zeros à mensagem original;
    - Calcular o CRC da mensagem;
    - Comparar o CRC calculado com o CRC recebido. Se o valor for idêntico, a mensagem está correta.

CRC – *Cyclic Redundancy Check*

- **CRCs Padronizados:**

Aplicação	Polinômio
CRC-1 (Paridade)	$x + 1$
CRC-8-ATM	$x^8 + x^2 + x + 1$
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32-MPEG2 CRC-32-IEEE 802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
CRC-64-ISO	$x^{64} + x^4 + x^3 + x + 1$

Conclusão

- Excelente algoritmo para detecção de erros provocados por ruídos em canais de comunicação;
- Implementação simples e eficiente em *hardware* binário;
- Amplamente utilizado.

EXERCÍCIO

- Faça o estudo da transmissão e recepção para o caractere “a” em código ASCII numa rede local de computadores que utiliza o padrão ETHERNET.