

Trabalho 5 comunicação de dados

Nicolas Beraldo

Outubro 2018

Este trabalho irá explicar e demonstrar os passos necessários para realizar um Hamming code.

Introdução

O hamming code se baseia em aumentar o tamanho da mensagem a ser enviada, assim os BITS adicionais serão usados para verificar se há erro na mensagem enviada. Os BITS adicionais tem o nome de BIT de paridade.

1 Hamming code

Para se usar a metodologia de processamento de erro do Hamming code é necessário que o emissor prepare a mensagem para o envio. O emissor faz uma análise de quantos bit há na mensagem a ser enviada e com essa informação faz uma conta simples para descobrir o tamanho da mensagem. A quantidade de BITS de paridade é “R” na seguinte equação:

$$2^R \geq \text{Tamanho da mensagem} + R + 1$$

Assim o tamanho da mensagem é:

$$\text{Tamanho para transmissão} = R + \text{Tamanho da mensagem}$$

Os BITS de paridade são todos aqueles que tem a sua posição sendo uma potencia de 2, logo são todos que tem como BIT ativo menos significativo sendo o BIT mais significativo. Para descobrir se o BIT de paridade é 0 ou 1 é necessário fazer o teste lógico “XOR” entre os BITS específicos, a escolha o BIT é feito se baseando no BIT ativo da posição em binário do bit de paridade. Então o BIT de paridade da posição “4” tem a sua posição sendo “100”, assim todos os BITS que tiverem o terceiro BIT ativo será incluso no “XOR” do BIT de paridade. Essa lógica é aplicada para todos os BITS de paridade assim criando uma mensagem que é capaz de identificar se há erro na mensagem.

Para se obter a mensagem de erro e saber se há erro ou não se realiza os “XOR” para os mesmo BITS se usando a mesma base para os BITS de paridade. Caso ao realizar os “XOR” e o resultado der 1 indica que ha um erro em algum BIT, após realizar o teste em todos os BITS de paridade se realiza uma soma com as posições em binário para descobrir em que posição se encontra o erro. Assim sendo um processo simples para preparar a mensagem e corrigi-la.

2 Mensagem de 7 dados

Nessa mensagem de 7 dados é necessário 4 BITS de paridade para transmitir a mensagem e indicar se ha erro ou não. Após definir o tamanho da mensagem a ser transmitida os bits da mensagem são inseridos nas posições corretas usando a ideia de que serão adicionados nas posições que não forem da base 2, deixando as posições dos BITS de paridade disponíveis. Agora é analisado quando se escolhe uma posição que seja da base 2 e realizamos os “XOR” necessários se baseando na no BITS da posição.

A relação dos BITS de paridade e quais BITS influenciam o seu valor está na tabela abaixo:

BIT de paridade	BITS usados no "XOR"
1	1,3,5,7,9,11
2	2,3,6,7,10,11
4	4,5,6,7
8	8,9,10,11

Se há um numero ímpar de "1"s nos BITS usados no "XOR" o BIT de paridade é "1" e caso seja um numero par de "1"s o BIT de paridade é "0", é por isso que recebe o nome "BIT de paridade". Para recuperar a mensagem após a correção ou não de um erro basta selecionar os BITS que não seja da base 2.

3 Mensagem de texto

O processo nesse caso é idêntico, apenas é adicionado um LOOP em todo o processo e o mesmo ocorre para todos os caracteres da mensagem, sendo um envio de caracteres em série.