

Simulador Camada Física - Linkcodesim

Teleinformática e Redes 1 - Turma A 201/2

Professor: Marcelo Antônio Marotta

Grupo: Gabriel Ritter Domingues dos Santos - 19/0067543

Marco Tulio Villela Ribeiro Faria - 18/0035495

João Tito do Nascimento Silva - 18/0123301

João Pedro de Oliveira Silva - 19/0057807

Universidade de Brasília

02 de Abril de 2022



Introdução

O presente trabalho tem como objetivo a implementação das codificações Binária, Manchester e Bipolar utilizando a linguagem de programação "C ++" com a finalidade de simular o funcionamento do enlace físico. O entendimento a respeito do funcionamento dos Bits na memória foi essencial para a realização do trabalho.

A troca de informações entre dispositivos e cabos é bastante efetiva justamente por existir a conversão dos códigos citados acima. Como exemplo, é possível citar que a repetição de bits iguais (vários “zeros” ou vários “uns”) causa problemas de tensão constante e usando codificações podemos contornar esses tipos de problema. O simulador foi desenvolvido para identificar, na prática, as diferentes codificações e seus funcionamentos.

Palavras-chave: Bits, Codificação, Binária, Manchester, Bipolar.

Implementação

No caso da Codificação Binária, a função presente no trabalho chamada “CamadaDeAplicacaoTransmissora(String mensagem)” é a responsável por codificar a mensagem, transformando cada caractere existente na mensagem em uma palavra de 8 bits, como podemos ver na imagem abaixo.

```
48 void CamadaDeAplicacaoTransmissora(string mensagem) {
49     // Transformar em bits
50     vector<int> quadro;
51
52     for (int i = 0; i < mensagem.length(); i++) {
53         auto c = mensagem[i];
54         quadro.push_back((c >> 7) & 0x1);
55         quadro.push_back((c >> 6) & 0x1);
56         quadro.push_back((c >> 5) & 0x1);
57         quadro.push_back((c >> 4) & 0x1);
58         quadro.push_back((c >> 3) & 0x1);
59         quadro.push_back((c >> 2) & 0x1);
60         quadro.push_back((c >> 1) & 0x1);
61         quadro.push_back((c >> 0) & 0x1);
62     }
63
64     // Chama a próxima camada
65     CamadaFisicaTransmissora(quadro);
66 }
```

Na codificação Bipolar são utilizados três níveis de tensão para representar os dados, sendo eles: -1, 0 e +1. Sendo que o 0 representa o “zero” binário e o -1 e +1 representam o “um” binário. Dessa forma, na implementação primeiramente é verificado se o último “um” foi -1 ou +1 e logo após ocorre a iteração nos bits e transformação dos mesmos em amostras do sinal.

```
14  vector<int> CamadaFisicaTransmissoraCodificacaoBipolar(vector<int> quadro) {
15      vector<int> result;
16
17      // vê se o ultimo 1 armazenado foi positivo ou negativo, -1 == negativo, 1 == positivo
18      int lastSignalOf1 = -1; // Para iniciar com nível alto no sinal
19
20      // Itera os bits e transforma em amostras do sinal
21      for (auto iter = quadro.begin(); iter < quadro.end(); iter++) {
22          auto bit = *iter;
23
24          if (bit == 0) {
25              result.push_back(0);
26          } else {
27              // Se o bit for 1, inverter a saída que foi dada no bit 1 anterior para a nova saída
28              result.push_back(-1 * lastSignalOf1);
29              // Salvar o que foi escrito dessa vez
30              lastSignalOf1 *= -1;
31          }
32      }
33
34      return result;
35  }
```

Já a codificação Manchester é um código de linha em que a codificação de cada bit de dados é “baixa e alta” ou “alta e baixa” por um tempo igual (é um sinal de sincronização automática sem componente DC). Dessa forma, quando o bit é 1, transforma-se em “10” na codificação e quando o bit é 0, transforma-se em “01” na codificação. É possível ver a representação dessa explicação na imagem abaixo, em que a função recebe um vetor da

mensagem já transformada em bits e itera a mesma, transformando os “zeros” em baixa→alta e os “uns” em alta→baixa.

```
37 vector<int> CamadaFisicaTransmissoraCodificacaoManchester(vector<int> quadro) {
38     vector<int> result;
39
40     for (auto iter = quadro.begin(); iter < quadro.end(); iter++) {
41         auto bit = *iter;
42
43         if (bit == 0) {
44             // 0 é representado por um "rising" edge
45             result.insert(result.begin(), 1);
46             result.insert(result.begin(), 0);
47         } else {
48             result.insert(result.begin(), 0);
49             result.insert(result.begin(), 1);
50         }
51     }
52
53     return result;
54 }
```

Membros

O trabalho foi bastante aproveitado por todos os integrantes do grupo, tendo em vista que a compreensão dos conceitos de codificação Binária, Bipolar e Manchester na prática foi o mais importante. Tito Silva foi o responsável por criar um repositório no GitHub e organizar a função de cada um no trabalho, dessa forma foi possível trabalhar com mais eficiência em equipe. Assim, os códigos foram revisados/corrigidos por todos os componentes, mas foram muito bem desenvolvidos por João Pedro, Marco Tulio e Tito Silva. Gabriel Ritter ficou com a responsabilidade de revisar/reparar os códigos e desenvolver o relatório da melhor maneira possível.

Conclusão e Resultados

Os resultados do trabalho desenvolvido foram bastante satisfatórios e os conceitos de Camadas Físicas, Codificação Binária, Codificação Manchester e Codificação Bipolar foram devidamente esclarecidos ao decorrer da realização do projeto. As aulas gravadas disponibilizadas pelo professor no Aprender também foram bastante esclarecedoras e contribuíram para o êxito na realização do presente trabalho. Como exemplificação dos resultados, será ilustrado um processo de execução e compilação do código final com o objetivo de mostrar que tudo ocorreu como o esperado.

```
Windows PowerShell
PS C:\src> g++ main.cpp -o main
PS C:\src> ./main
-----
Qual o tipo de transmissao deseja?
1 - Codificacao Binaria
2 - Codificacao Bipolar
3 - Codificacao Manchester
-----
1
Digite uma mensagem:
teste

Codificacao Binaria: 0111010001100101011100110111010001100101
Decodificacao Binaria: 0111010001100101011100110111010001100101
A mensagem recebida foi:
teste
PS C:\src> ./main
-----
Qual o tipo de transmissao deseja?
1 - Codificacao Binaria
2 - Codificacao Bipolar
3 - Codificacao Manchester
-----
2
Digite uma mensagem:
teste

Codificacao Bipolar: 01-110-10001-10010-101-1100-110-11-101000-1100-101
Decodificacao Bipolar: 0111010001100101011100110111010001100101
A mensagem recebida foi:
teste
PS C:\src> ./main
-----
Qual o tipo de transmissao deseja?
1 - Codificacao Binaria
2 - Codificacao Bipolar
3 - Codificacao Manchester
-----
3
Digite uma mensagem:
teste

Codificacao Manchester: 10011001011010010101100110101001101001011010100110011001011010010101100110101001
Decodificacao Manchester: 0111010001100101011100110111010001100101
A mensagem recebida foi:
teste
PS C:\src>
```

Referências

- <https://br.ccm.net/contents/56-a-codificacao-binaria>
- <http://comunicacaodedados2014.blogspot.com/p/codificacao.html#:~:text=Bipolar%3A%20Nesse%20esquema%2C%20utiliza-,exemplo%2C%20é%20a%20codificação%20AMI>
- <https://pt.lambdageeks.com/line-coding-manchester-encoding/>