Simulador Camada Física

Astro Nautas:
Gabriel Cruz Vaz Santos
Nícolas Paulin Benatto
Pedro Venzi Lima Monteiro de Oliveira

Introdução

O objetivo do trabalho é simular o funcionamento da camada física por meio da implementação dos protocolos Binário, Manchester e Bipolar.

O simulador deve, então, atravessar todas as camadas de aplicação e físicas, transmitindo uma mensagem de um ponto a outro codificada utilizando cada um dos protocolos já apresentados.

Implementação

A implementação do simulador se dá com base em funções que representam as camadas de transmissão e recepção.

- **AplicacaoTransmissora**

A função AplicacaoTransmissora recebe do usuário uma mensagem e o tipo de decodificação que o usuário deseja utilizar. Essa mensagem é enviada para a camada de aplicação transmissora.

- **CamadaDeAplicacaoTransmissora**

A função CamadaDeAplicacaoTransmissora transforma a nossa mensagem em um vetor quadro de bits que será transmitido por meio de nossa camada física transmissora.

- **CamadaFisicaTransmissora**

A função CamadaFisicaTransmissora é responsável por definir qual protocolo será utilizado e, quando temos nosso sinal a ser enviado, transmitir essa mensagem pelo meio de comunicação.

- **CamadaFisicaTransmissoraCodificacaoBinaria**

A função CamadaFisicaTransmissoraCodificacaoBinaria codifica nosso código de acordo com o protocolo Binário. Como temos vários tipos de protocolos binários, escolhemos o protocolo NRZ para a implementação de nossa codificação.

O protocolo NRZ funciona da seguinte maneira: os bits ‘0’ são codificados como um sinal nulo e os bits ‘1’ são codificados como um sinal positivo. No código, escolhemos ‘0’ para representar o sinal nulo e ‘1’ para representar o sinal positivo.

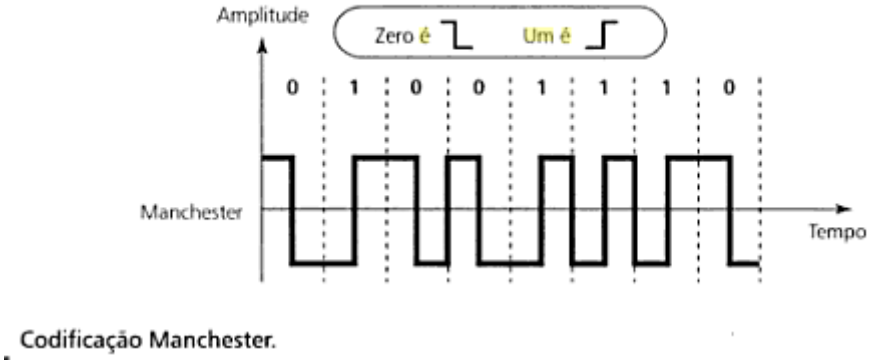
Exemplo de sinal codificado com protocolo NRZ

O sinal é armazenado como um fluxo, que retornará à camada física transmissora para ser enviada ao nosso meio de comunicação.

- **CamadaFisicaTransmissoraCodificacaoManchester**

A função CamadaFisicaTransmissoraCodificacaoManchester codifica nosso código de acordo com o protocolo Manchester.

O protocolo Manchester utiliza um clock junto com os bits que estão sendo transmitido para obtermos nosso sinal elétrico. Ele funciona com uma operação XOR em todos os bits, que são acompanhados de uma subida de clock, que faz com que o nosso sinal elétrico tenha dois valores a cada bit transmitido, uma vez que o clock é sempre “01” e, para um bit ‘0’ a saída é um sinal nulo + positivo e para um bit ‘1’ a saída é um sinal positivo + nulo.



Exemplo de sinal codificado com protocolo Manchester

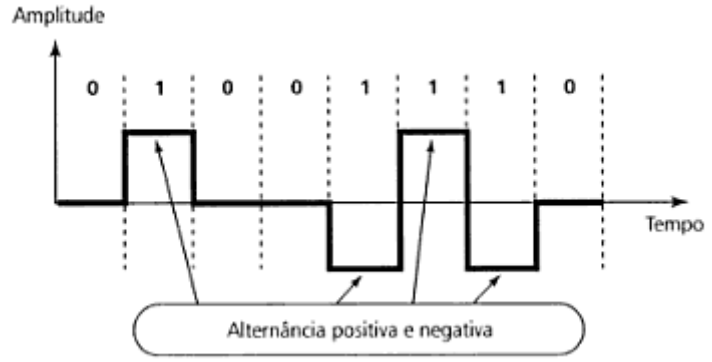
O sinal é armazenado como um fluxo, que retornará à camada física transmissora para ser enviada ao nosso meio de comunicação.

- **CamadaFisicaTransmissoraCodificacaoBipolar**

A função CamadaFisicaTransmissoraCodificacaoBipolar codifica nosso código de acordo com o protocolo Bipolar.

O protocolo Bipolar funciona da seguinte maneira: os bits ‘0’ são codificados como um sinal nulo e os bits ‘1’ são codificados como sinais alternados. Um

exemplo disso seria o sinal “01100001”, em que o primeiro ‘1’ será um sinal positivo, o segundo ‘1’ será um sinal negativo e o terceiro ‘1’ será novamente um sinal positivo. Escolhemos, na nossa codificação, ‘0’ para sinal nulo, ‘1’ para sinal positivo e ‘-1’ para sinal negativo.



Exemplo de sinal codificado com protocolo Bipolar.

O sinal é armazenado como um fluxo, que retornará à camada física transmissora para ser enviada ao nosso meio de comunicação.

- **MeioDeComunicacao**

A função MeioDeComunicacao recebe nosso fluxo elétrico e transmite para outros fluxos do meio até chegar à camada física receptora para ser decodificado.

- **CamadaFisicaReceptora**

A função CamadaFisicaReceptora é responsável por receber qual protocolo foi utilizado e decodificá-lo e, quando temos nosso sinal recebido, enviar essa mensagem para a camada de aplicação receptora.

- **CamadaFisicaReceptoraDecodificacaoBinaria**

A função CamadaFisicaReceptoraDecodificacaoBinaria decodifica nosso sinal de acordo com o protocolo Binário escolhido, que, no caso, foi o NRZ.

Para isso, recebemos nosso sinal e transformamos o sinal nulo em ‘0’s e o sinal positivo em ‘1’s.

Depois disso, nosso quadro será enviado para a camada de aplicação receptora.

- **CamadaFisicaReceptoraDecodificacaoManchester**

A função CamadaFisicaReceptoraDecodificacaoManchester decodifica nosso sinal de acordo com o protocolo Manchester.

Para isso, recebemos nosso sinal e transformamos cada par de sinais nulo + positivo em ‘0’s e cada par de sinais positivo + nulo em ‘1’s.

Depois disso, nosso quadro será enviado para a camada de aplicação receptora.

- **CamadaFisicaReceptoraDecodificacaoBipolar**

A função CamadaFisicaReceptoraDecodificacaoBipolar decodifica nosso sinal de acordo com o protocolo Bipolar.

Para isso, recebemos nosso sinal e transformamos o sinal nulo em ‘0’s, o sinal positivo em ‘1’s e o sinal negativo também em ‘1’s.

Depois disso, nosso quadro será enviado para a camada de aplicação receptora.

- **CamadaAplicacaoReceptora**

A função CamadaAplicacaoReceptora recebe nosso quadro e transforma o conjunto de bits na mensagem a ser escrita na aplicação receptora.

- **AplicacaoReceptora**

A função AplicacaoReceptora recebe nossa mensagem e mostra ela na tela, sendo que o esperado é que seja a mesma mensagem enviada ao simulador na aplicação transmissora.

Membros

A maior parte do desenvolvimento do presente simulador foi realizado por meio de reuniões remotas entre dois ou três membros do grupo por vez, tanto para as etapas de projeção e desenvolvimento quanto para as de depuração do código.

A codificação em grupo foi realizada seguindo método análogo ao *pair programming* encontrado em metodologias ágeis, onde há definição de papéis de *driver* (encarregado da aplicação das decisões de design em formato de código) e *navigator* (encarregado de revisar o código a medida em que é desenvolvido e constantemente avaliar o design presente visando melhorias futuras). Os papéis, periodicamente, são rotacionados para promover um entendimento unificado acerca do código e a diversificação das ideias durante o processo de codificação. No caso do nosso grupo, adaptamos tais conceitos de organização para um grupo de 3 integrantes.

Dessa forma, todos os membros estiveram envolvidos com as etapas do processo de codificação, participando nas decisões de design e implementação dos processos e rotinas presentes no simulador.

- Codificação (projeção, desenvolvimento, depuração, execução)
    - Gabriel, Nicolas e Pedro
- Comentários
    - Gabriel, Nicolas e Pedro
- Relatório
    - Nicolas e Pedro

Conclusão

Quanto à codificação e execução, o simulador produz resultados dentro do esperado: o usuário digita uma mensagem, a qual é ‘transmitida’ com sucesso da ponta A para a ponta B. Porém, a transmissão correta da mensagem é apenas parte do objetivo do simulador apresentado.

Crucialmente, o simulador logra êxito em descrever, passo a passo, a comunicação entre as camadas de aplicação e física e seus processos de codificação e transferência de informação. Somado à explicitação de cada passo de forma gráfica no terminal em tempo de execução, aos comentários esclarecedores ao longo do código e ao relatório, o simulador cumpre sua função didática - seja a nível de camada (*e.g* a interação entre camadas distintas), seja a nível de bit (*e.g* formatação ASCII, codificação de bit para sinal elétrico).

As dificuldades sofridas pelo grupo Astro Nautas provém da falta de familiaridade dos integrantes com a linguagem C/C++, acentuando a curva de aprendizado referente à realização do trabalho e aumentando consideravelmente o tempo de desenvolvimento e depuração do simulador.