Codificação de linha.

Teleinformática e Redes 1

Integrantes

David Herbert de Souza Brito - 200057405

João Marcos Melo Monteiro - 130143031

Luigi Paschoal Westphal de Oliveira - 190062894

DEPARTAMENTO: Departamento de Ciência da Computação – CiC/UnB
DISCIPLINA: Teleinformática e Redes I
CÓDIGO: 204315
PROFESSOR: Marcelo Antônio Marotta

Introdução

A fonte do sinal de comunicação possui uma sequência de bits (mensagem digital) a ser transmitida; O processo de codificação de linha consiste em modificar o sinal digital binário em uma representação elétrica, adequando-o para a transmissão.

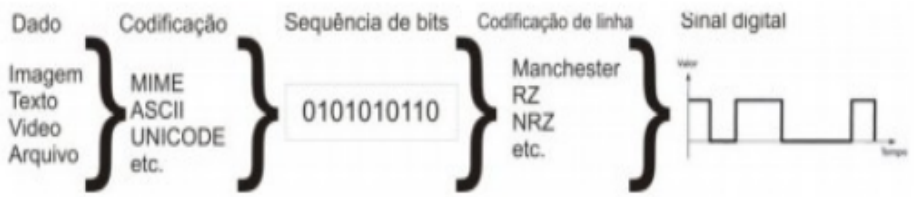

Figura 1. Codificação de linha.

Camada Física - TR1

Trabalho 2 - Teleinformática e Redes 1

Unb - 2022/2

Especificação do trabalho

Simular o funcionamento da camada física por meio da implementação dos seguintes protocolos:

- Binário
- Manchester
- Bipolar

Implementação

Codificação de Linha

A fonte do sinal de comunicação possui uma sequência de bits (mensagem digital) a ser transmitida; O processo de codificação de linha consiste em modificar o sinal digital binário em uma representação elétrica, adequando-o para a transmissão. Existem inúmeras formas de codificação de linha, estudaremos as mais utilizadas:

- Sinal Binário NRZ e RZ;
- Código Bipolar ou AMI ;
- Código Manchester;

Propriedades requeridas nos códigos de Linha Banda de Transmissão:

- Ocupação da menor banda de transmissão possível;

Eficiência Energética:

- A potência de transmissão deve ser a menor possível;

Detecção e correção de erros:

- Deve ser possível a detecção de erros na transmissão e a correção dos mesmos.

Densidade espectral de potência:

- É desejável densidade espectral de potência nula em w=0 (valor DC nulo), pois acoplamentos AC (capacitivo) e transformadores (indutivo) são utilizados em alguns repetidores.

Tipos de códigos de linha binários.

SINAL BINÁRIO NÃO RETORNA A ZERO (NRZ):

• O sinal NRZ (Não Retorna a Zero) consiste em manter o nível do sinal em alto quando o bit é “1” e em nível baixo quando o bit é “0”.

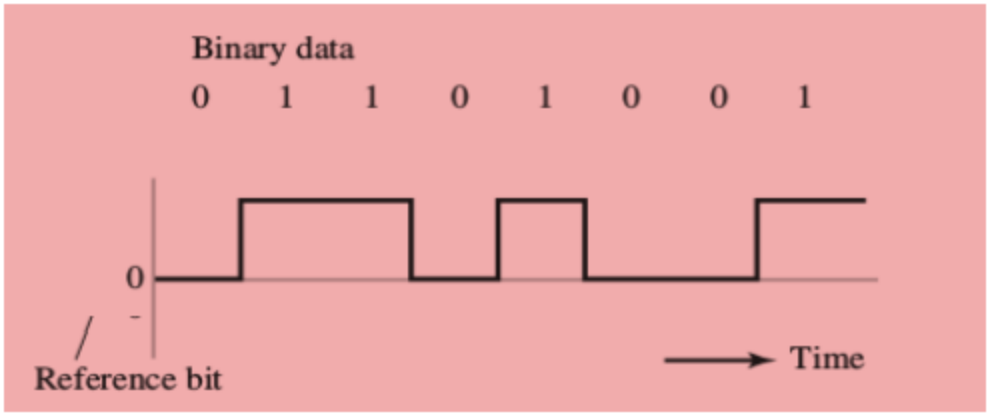

Figura 2. Sinal NRZ.

Tipos de códigos de linha binários.

SINAL BINÁRIO RETORNA A ZERO (RZ):

• O sinal RZ (Retorna a Zero) consiste em uma pequena modificação onde o sinal retorna ao nível baixo no meio do bit “1”.

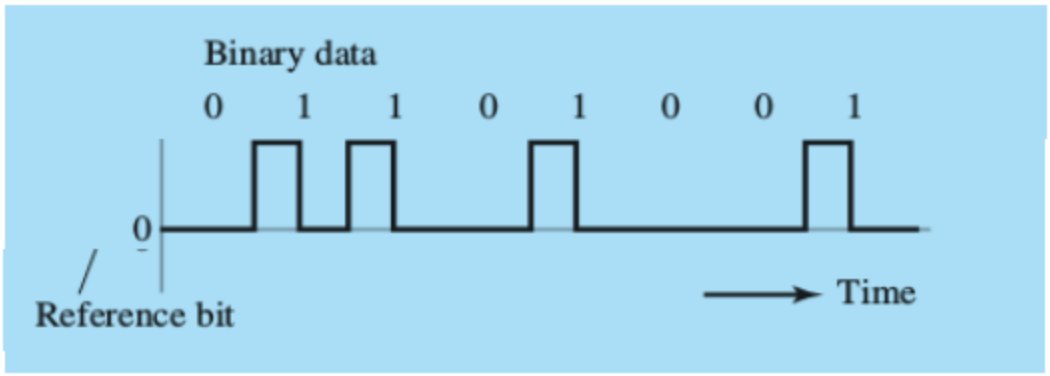

Figura 3. Sinal RZ.

O código AMI (Alternate Mark Inversion), ou código Bipolar Simples, ou simplesmente sinal bipolar. O código AMI (Alternate Mark Inversion), ou código Bipolar Simples, ou simplesmente sinal binário ou AMI (Alternate Mark Inversion),utiliza três níveis de sinal (+,0,-) para codificar a informação binária.

BIPOLAR OU AMI:

• O sinal AMI consiste em gerar pulsos positivos e negativos alternadamente cada vez que a informação é um pulso "1". Os espaços "0" são caracterizados pela ausência de pulso.

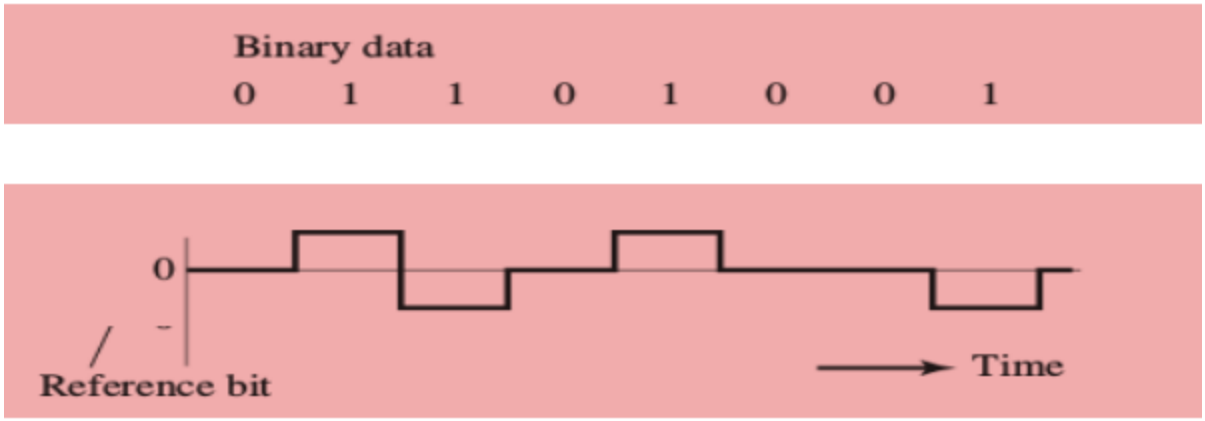

Figura 4. Sinal bipolar.

CÓDIGO MANCHESTER ou Bifase:

CÓDIGO MANCHESTER ou Bifase:

• No código Manchester o bit "1" é representado por um pulso positivo seguido de um pulso negativo, ambos de mesma amplitude e largura. • Para o bit "0", a polaridade dos pulsos é invertida.

Codificação de linha sinal de Manchester. O sinal faz parte da subclasse de sinais com codificação de fase, onde os bits, em vez de serem representados pelo nível dos pulsos, são pelas fases dos pulsos (transições).

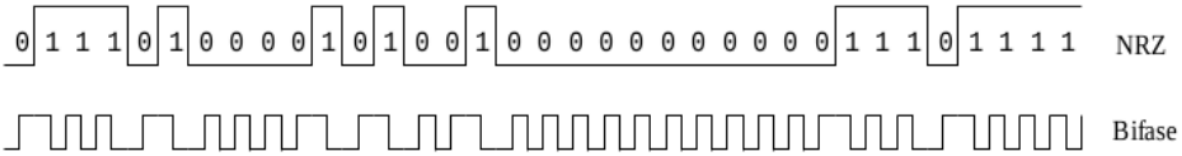

Figura 5: Comparação do clock do sinal binário NRZ (Não Retorna a Zero) e Bifase ou Manchester.

Nome do simulador:

O projeto foi criado no VScode e compilado no wsl.

Para compilar, basta rodar:

→make

Para executar:

→make run

Para limpar o diretório:

→make clean

Compilador:

→CPP = g++

Nome do Projeto:

→Sim = simulador

Flags G++:

→FLAGS= -c -Wall -pedantic

all: $(SIM)

Nome do simulador:

O projeto foi criado no VScode e compilado no wsl.

Para compilar, basta rodar:

→make

Para executar:

→make run

Para limpar o diretório:

→make clean

```
#COMPILER
CPP=g++

# Nome do projeto
SIM=simulador

#FLAGS G++
FLAGS= -c -Wall -pedantic

all: $(SIM)

$(SIM): simulador.o
 camadafisicaprotocolo.o
	$(CPP) -o $@ $^

simulador.o: simulador.cpp
camadafisicaprotocolo.hpp
	$(CPP) -o $@ $< $(FLAGS)
```

```
camadafisicaprotocolo.o: camadafisicaprotocolo.cpp
camadafisicaprotocolo.hpp
	$(CPP) -o $@ $< $(FLAGS)

run:
	./$(SIM)

clean:
	rm -rf *.o $(SIM)
```

Atividades dos Membros

Descrição das atividades desenvolvidas por cada membro do grupo:

David Herbert

Desenvolveu o código .h e ajudou na implementação do relatório.

João Marcos

Desenvolveu códigos .cpp, .h e ajudou na implementação do relatório.

Luigi Paschoal

Desenvolveu o código .cpp e ajudou na implementação do relatório.

Conclusão

Os dados e os sinais que representam os dados podem ser digitais ou analógicos. Codificação de linha é o processo de conversão de dados digitais em sinais digitais. Com esta técnica, convertemos uma sequência de bits em um sinal digital. No lado do envio, os dados digitais são codificados em um sinal digital e, no lado da recepção, os dados digitais são recriados pela decodificação do sinal digital. Podemos dividir os esquemas de codificação de linha em cinco categorias: Unipolar (por exemplo, esquema NRZ). Polar (ex. NRZ-L, NRZ-I, RZ e Bifásico - Manchester e Manchester Diferencial). Bipolar (como IAM e Pseudo-Triplo). Multinível múltiplas transformações.

Referências bibliográficas

https://edisciplinas.usp.br/pluginfile.php/5228477/mod_resource/content/1/2020_Aula_05_Camada_Fisica_Parte_3.pdf

https://acervolima.com/diferenca-entre-esquemas-de-codificacao-de-linha-unipolar-polar-e-bipolar/