

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

Jean Lucca Uchaki

Relatório

Porto Alegre

2021

SUMÁRIO

1	NRZ-I.....	1
2	Manchester Diferencial.....	3
3	8B6T	5
4	6B/8B	7
5	HDB3.....	8

1 NRZ-I

Encode:

Consiste em percorrer os bits e converter em sinais digitais, o sinal muda sempre que o bit 1 é encontrado na String, começando com um sinal negativo.

```
public static String encode(String bin) {
    String res = "";
    char aux = '-';
    for(int i=0;i<bin.length();i++) {
        if(!Util.isValidBin(bin.charAt(i))) { return "Erro"; }
        if( bin.charAt(i) == '1' ) {
            if( aux == '-' ) {
                aux = '+';
            } else {
                aux = '-';
            }
        }
        res+=aux;
    }
    return res;
}
```

Decode:

Converte os sinais em uma String binaria, sempre que houver uma mudança de sinal converte o sinal para 1, começando com um sinal negativo.

```
public static String decode(String signal) {
    String aux = '-' + signal;
    String res = "";
    for(int i=1;i<aux.length();i++) {
        //caso tenha mudanca de sinal condatenar 1
        if(!Util.isValidSignal(aux.charAt(i))) { return "Erro"; }
        if(aux.charAt(i) == aux.charAt(i-1)) {
            res += '0';
        } else {
            res += '1';
        }
    }
    return res;
}
```

Exemplos de Codificação/Decodificação:

```
C:\Windows\system32\cmd.exe
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador nrzi 5678
++--+-----
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador nrzi +----+---+-----
5678
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador nrzi 1234
--++-----+-----
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador nrzi ---++-----+---
1234
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador nrzi 1a34
--+-----+-----
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador nrzi ---+-----+---
1a34
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador nrzi 1aff
--+-----+-----
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador nrzi ---+-----+---
1aff
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador nrzi 0aff
---+---+---+---+
```

2 Manchester Diferencial

Encode:

Começa com um sinal negativo, caso encontre um bit zero antera o sinal e bit 1 permanece no mesmo nível anterior e depois troca.

```
public static String encode(String bin) {
    String res = "";
    String aux = "--";
    for(int i=0;i<bin.length();i++) {
        if(!Util.isValidBin(bin.charAt(i))){ return "ERRO"; }
        if( bin.charAt(i) == '0' ) {
            if( aux.charAt(1) == '-' ) {
                aux = "+-";
            } else {
                aux = "-+";
            }
        } else {
            if( aux.charAt(1) == '-' ) {
                aux = "-+";
            } else {
                aux = "+-";
            }
        }
        res+=aux;
    }
    return res;
}
```

Decode:

Percorre os bit de 2 em 2, se houver alteração no sinal concatena 1

```
public static String decode(String signal) {
    String aux = '-' + signal;
    String res = "";
    for( int i = 0; i < aux.length()-1; i+=2 ) {
        if(!Util.isValidSignal(signal.charAt(i))){ return "ERRO"; }
        if(aux.charAt(i) != aux.charAt(i+1)) {
            res += '0';
        } else {
            res += '1';
        }
    }
    return res;
}
```

Exemplos de Codificação/Decodificação:

```
C:\Windows\system32\cmd.exe

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 5678
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
0101011001111000

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 5678
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
5678

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 1234
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
1234

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 1a34
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
1a34

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 1aff
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
1aff

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador mdif 1af0
+--++++-+--++++-+--++++-+--+

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador mdif +--++++-+--++++-+--++++-+--+
1af0
```

3 8B6T

Encode:

Utiliza a tabela de conversão para a codificação (nesta implementação foi usado um Hashmap), calcula a disparidade dos sinal e inverte quando for necessário equilibrar a positividade.

```
public String encode(String hex) {
    String curr = "";
    String res = "";
    int positivity = 0;
    hex = hex.toUpperCase();
    for(int i=0;i<hex.length()-1;i+=2) {
        curr = table.get(""+hex.charAt(i) + hex.charAt(i+1));
        if( curr == null ) { return "ERRO"; }
        if( positivity > 0 && Util.countSignalDisparity( curr ) > 0) {

            res += Util.invertSignal(curr);
            positivity += Util.countSignalDisparity(
                Util.invertSignal( curr ) );

        } else if( positivity < 0
            && Util.countSignalDisparity( curr ) < 0) {

            res += Util.invertSignal( curr );
            positivity += Util.countSignalDisparity(
                Util.invertSignal( curr ) );

        } else {

            res+= curr;
            positivity += Util.countSignalDisparity( curr );

        }
    }
    return res;
}
```

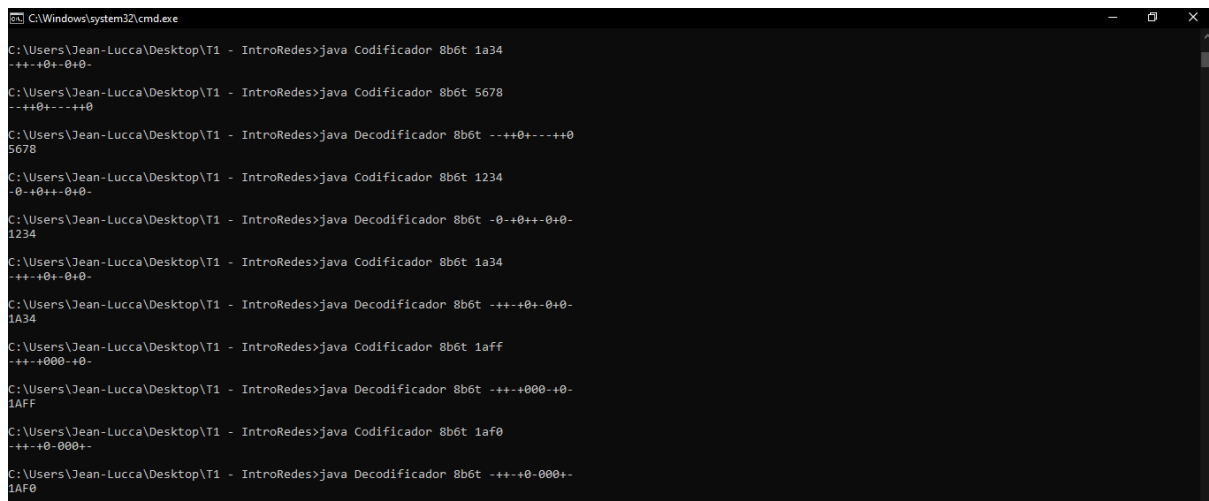
Decode:

Procura o sinal na tabela de conversão, caso não encontre inverte o sinal e procura novamente.

```
public String decode(String signal) {
    String res = "";
    for(int i = 0; i < signal.length(); i+=6 ) {
        String key = searchSignal(signal.substring(i, i+6));

        if(!key.equals("not found")) {
            res += key;
        } else {
            String aux = Util.invertSignal(signal.substring(i, i+6));
            res += searchSignal(aux);
        }
    }
    return res;
}
```

Exemplos de Codificação/Decodificação:



```
C:\Windows\system32\cmd.exe
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 1a34
-+-+0+-0+0-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 5678
-++0+---++0
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 8b6t -++0+---++0
5678
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 1234
-0-+0+-0+0-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 8b6t -0-+0+-0+0-
1234
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 1a34
-+-+0+-0+0-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 8b6t -+-+0+-0+0-
1A34
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 1aff
-++-+000-+0-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 8b6t -++-+000-+0-
1AFF
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 8b6t 1af0
-++-+0-000+-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 8b6t -++-+0-000+-
1AF0
```


4 6B/8B

Encode:

Aplica as regras de codificação, primeiro convertendo os blocos de 6 para 8 bits de acordo com a tabela de conversão, depois codifica utilizando Nrzi.

```
public String encode(String bin) {
    String res = "";
    for ( int i = 0; i < bin.length(); i+=6 ) {
        res += applyEncode(bin.substring(i, i+6));
    }
    return Nrzi.encode(res);
}
```

Decode:

Decodifica utilizando Nrzi, depois converte de os blocos 8 para 6 bits e converte de acordo com a tabela e as regras de conversão.

```
public String decode(String signal) {
    String res = "";
    signal = Nrzi.decode(signal);

    for ( int i = 0; i < signal.length(); i+=8 ) {
        res += applyDecode(signal.substring(i, i+8));
    }
    return res;
}
```

Exemplos de Codificação/Decodificação:

```
C:\Windows\system32\cmd.exe

C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 6b8b 56789A
+++++-----+++++-----+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 6b8b +---+-----++-+-----+-----+
56789a
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 6b8b 5678AA
+++++-----+++++-----+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 6b8b +---+-----++-+-----+-----+
5678aa
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 6b8b 1234AA
+++++-----+++++-----+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 6b8b +-+-----++-+-----+-----+
1234aa
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 6b8b 1234FA
+++++-----+++++-----+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 6b8b +-+-----++-+-----+-----+
1234fa
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador 6b8b 1234F0
+++++-----+++++-----+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador 6b8b +-+-----++-+-----+-----+
1234f0
```

5 HDB3

Encode:

Funciona como AMI quando não são encontrados 4 zeros consecutivos, caso contrário verifica a disparidade do sinal e codifica de acordo com a paridade e o pulso anterior.

```
public static String encode(String bin) {
    String res = ""; char aux = '-'; int zeros = 0;
    for( int i = 0; i < bin.length(); i++ ) {
        if(!Util.isValidBin(bin.charAt(i))) { return "Erro"; }
        if( i+4 <= bin.length() ) {
            zeros = Util.countZeros(bin.substring(i,i+4));
        } else { zeros = 0; }

        if( zeros == 4 ) {
            if( Util.countSignalDisparity(res)%2 == 0 ) {
                if(aux == '+') {
                    res += "-00-"; aux = '-';
                } else {
                    res += "+00+"; aux = '+';
                }
            } else {
                if(aux == '+') {
                    res += "000+"; aux = '+';
                } else {
                    res += "000-"; aux = '-';
                }
            }
        }
        i+=3;
    } else if(bin.charAt(i) == '0') {
        res+='0';
    } else {
        if( aux == '-' ) {
            aux = '+';
        } else {
            aux = '-';
        }
        res += aux;
    }
}
return res;
```

Decode (Incompleto):

Caso encontre um sinal positivo ou negativo concatena um, senão concatena 0.
Decodifica corretamente apenas quando a String decodificada não tem 4 zeros consecutivos.

Exemplo: 0001 0010 0011 0100, 0101 0110 0111 1000

```
public static String decode(String signal) {
    String res = "";
    char aux = '-';
    for( int i = 0; i < signal.length(); i++ ) {
        if(!Util.isValidSignal2(signal.charAt(i))) { return "Erro"; }
        if(signal.charAt(i) == '0') {
            res+='0';
        } else {
            if( aux == '-' ) {
                aux = '+';
                res += '1';
            } else {
                aux = '-';
                res += '1';
            }
        }
    }
    return res;
}
```

Exemplos de Codificação/Decodificação:

```
C:\Windows\system32\cmd.exe
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador hdb3 000+00-000+-0+00
1234
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador hdb3 5678
0+0-0+-00+-+-000
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador hdb3 0+0-0+-00+-+-000
5678
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador hdb3 6785
0+-00+-+-+00+-0+
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador hdb3 0+-00+-+-+00+-0+
67cd
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Codificador hdb3 12ff
000+00-0+-+-+-
C:\Users\Jean-Lucca\Desktop\T1 - IntroRedes>java Decodificador hdb3 000+00-0+-+-+-
12ff
```