VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS & COMMUNICATIONS

Data Communication Report

Topic: Simulation of image transmission using line coding methods

1st Trinh Ngoc Huy *19207075@student.hcmus.edu.vn*
2nd Nguyen Quoc Huy *19207074@student.hcmus.edu.vn*
3rd Truong Trong Duc *19207062@student.hcmus.edu.vn*
4th Huynh Nguyen Dat *19207058@student.hcmus.edu.vn*
5th Vo Nguyen Long An *19207047@student.hcmus.edu.vn*

September 28, 2022

Contents

List of Figures

Chapter 1

Line Coding

1.1 Definition of Line Coding

The binary data transmitted can use a variety of different types of pulses. The selection of a special pair of pulses to represent symbols 1 and 0 is called line coding [1]. This is the process of converting or mapping a string of binary data into a digital signal (transmission waveform).

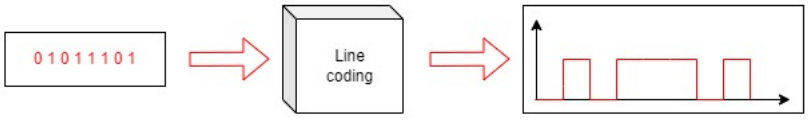


Figure 1.1: The process of converting binary numbers to digital signals.

1.1.1 Purpose

- Create the spectrum of digital signals so that it is more suitable for the transmission channel.
- Create the ability to separate the synchronized signal in the receiver.
- Increase transmission speed.
- Quality monitoring. It is capable of detecting errors and can correct errors.

1.1.2 Requirements for transmission code

- Self-synchronization.
- There is no one-dimensional component (DC=0).
- The energy at the low frequency should be small.
- There are many pulse edges to restore clock pulse in the receiver.
- The encryption signal must be able to decipher uniquely into the original signal.
- Easy to restore clock efficiency for easy synchronization.
- Narrow frequency range to save transmission bandwidth.
- The transformation has rules so that the receiver can control the bit error.

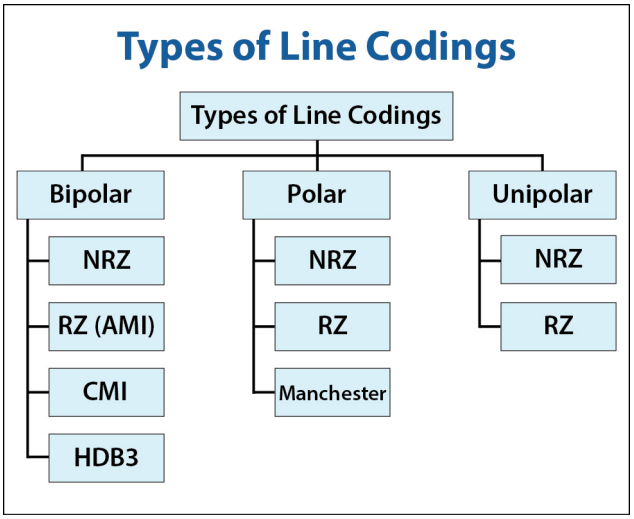


Figure 1.2: Types of line codings.

1.2 Digital - Digital convert

- **Concept:** Digital conversion (Encoding) is a method of representing digital data using digital signals.
- **For example:** When transferring data from a computer to a printer, the original and transmitted data are both digital.
- **Characteristics:** Bits '1' and '0' are converted into voltage pulse chains so that they can be transmitted through the line.
- **Block diagram**

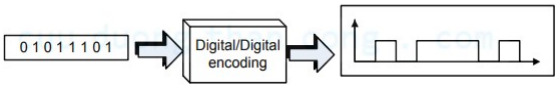


Figure 1.3: A method of representing digital data with digital signals.

- **Classification:** Unipolar, Polar, Bipolar

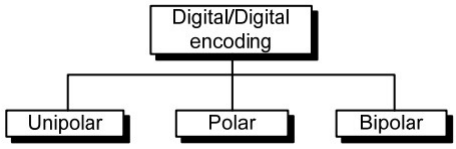


Figure 1.4: Classification Digital-Digital encoding.

1.2.1 Unipolar

- It is the simplest form of encryption (original-born first).
- One voltage level denotes bit '0' and another voltage level indicates bit '1'.

  For example: Bit '0' $\rightarrow$ 0 volts and '1 $\rightarrow$ +V volt (+5V, +9V...).; Exists in a Bit Cycle.

  For example: for a 01001110 bit sequence, represent this bit string as unipolar code.

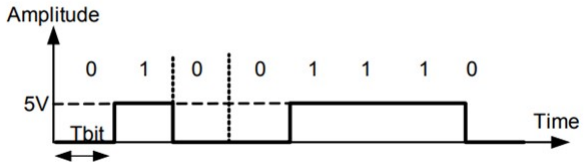


Figure 1.5: Unipolar.

- Advantages: simple and low cost.
- Disadvantage: One-way voltage (DC) exists and synchronization problem.

DC component: The average value of other unipolar code, creating the DC voltage component on the transmission line. When the signal exists dc component, it is impossible to pass through the transmission environment [2].

Synchronization: When the transmission signal has an unchanged value, the receiver cannot determine the lifetime of a bit (Bit Cycle). The solution can use an additional conductor to transmit a synchronized signal to help the receiver know about the bit cycle [2].

1.2.2 Polar

- Polar encryption uses two voltage levels: one with a positive value and one with a negative value, in order to reduce the DC component.
- Classification: NRZ, RZ and Biphase.
  - NRZ: NRZ-L (nonreturn to zero-level: COM port RS232) and NRZ-I (nonreturn to zero – invert).
  - RZ (return to zero).
  - Biphase: Manchester (used in LAN ethernet networks), Manchester differential (commonly used in Token Ring LAN).

1.2.3 Non Return To Zero (NRZ)

- Characteristics: The signal has a value of positive (+V) or negative (-V).
- Classification: NRZ – L (PORT COM RS232) and NRZ - I.

a. NRZ-L

- Characteristics: Bit '0 → +V (+3V, +5V, +15V..); Bit '1' → -V (-3V, -5V,- 15V...).

  For example: for 01001110 string, represent this bit string as NRZ-L code.

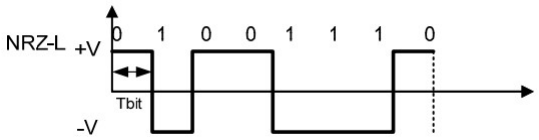


Figure 1.6: NRZ-L.

  - **Advantages:** DC components are reduced over monopolar code.
  - **Disadvantage:** Sync problem: When the transmission signal has an unchanged value, the receiver cannot determine the lifetime of a bit (bit cycle). The solution can be used to add a conductor to transmit a synchronous signal to help the receiver know about the bit cycle.

b. NRZ-I

- Characteristics:
  - Meet bit '1'→ will reverse the voltage pole earlier.
  - Meet the bit '0'→ will not reverse the voltage pole beforehand. (The first bit can be assumed to be positive or negative).

    For example: for the 01001110 sequence, represent this bit string as the code NRZ - I. Assuming the initial positive voltage.

- The advantage over NRZ - L synchronized problem was solved when encountering the 1st bit sequence in a row.

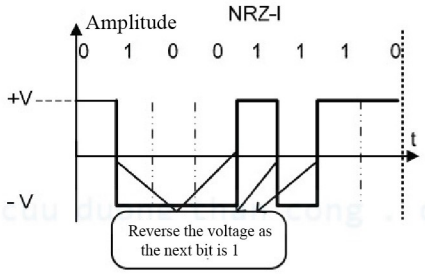


Figure 1.7: NRZ-I.

1.2.4 Return To Zero (RZ)

- Characteristics:
  - Bit '0 → The first half of the bit's cycle is voltage-V and the second half of the bit is 0V voltage.
  - Bit '1 → The first half of the bit's cycle is +V voltage and the second half of the bit's cycle is 0V voltage. For example, for 01001110 string, represent this bit string as a RZ code.

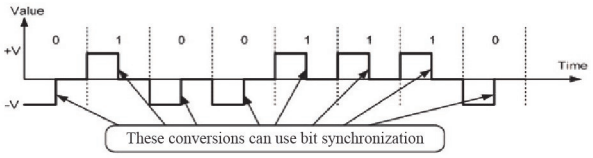


Figure 1.8: RZ.

- Advantages: Solve sync problems for consecutive '1' or '0' bit strings.
- Disadvantages: having broader bandwidth (large frequency range). There are three voltage levels.

However, we will see that this is the most effective method. (A good method of encrypting digital signals must have redundancy for sync mode).

1.2.5 Biphase

- Characteristics:
  - +V and -V voltages exist in 1 bit.
  - DC components in helium.
  - Good synchronization method.
- Classification: Manchester and Different Manchester.

1.2.6 Manchester

- Characteristic [2]:
  - The manchester code combines the clock pulse signal with the data signal. Not only does it increase the signal frequency, it also makes data transmission easier and more reliable.
  - The key feature of the Manchester code is that it encodes the clock signal by transferring the rod in the middle of the bit cycle.
  - This level transfer is used at the receiver to restore the clock.
  - Manchester codes are widely used, such as Ethernet for RFID.
  - Different Manchester: used in token-ring network.
  - Bit '0 $\rightarrow$ The first half of the bit's cycle is voltage +V and the other half cycle is voltage -V.
  - Bit '1 '$\rightarrow$ The first half of the bit's cycle is voltage-V and the other half cycle is voltage +V.

Transcoding rules:

Manchester code (based on G.E. Thomas):

- Bit 0: Move from low to high in the middle of the bit.
- Bit 1: Move from high to low in the middle of the bit.

Manchester Code (based on IEEE 802.3) :

- Bit 0: Move from high to low in the middle of the bit.
- Bit 1: Move from low to high in the middle of the bit.

The two Manchester codes above have the same properties: Since each bit is coded by 2 voltage phases, the velocity is even The mechanism of this type of code doubles compared to other types of code.

Specifically, assuming the duration of a bit is T, the modulation velocity the maximum (corresponding to the pulse sequence 1 or 0 in a row) is 2/T.

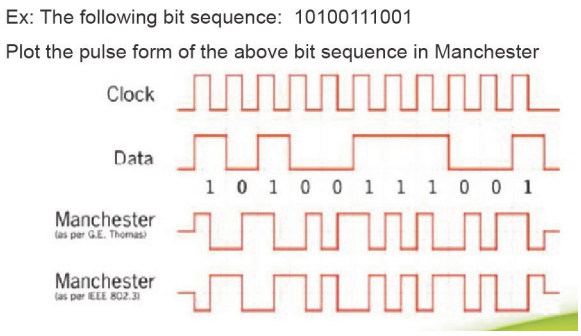


Figure 1.9: Manchester.

Advantage:

- The ability to synchronize pulses with the current clock pulse, overcoming the disadvantages of NRZI in decoding and encoding.
- Do not suffer from the phenomenon of "Signal Droop" (suitable for transmission on ac transmission lines).
- Does not contain DC components.
- It's a transparent signal.

Disadvantages:

- Large bandwidth.
- Inability to detect errors.

1.2.7 Different Manchester

- Encountering bit '0' will reverse the voltage pole earlier.
- Meeting bit '1' will keep the previous voltage pole intact.
- There is always a voltage change in the middle of the bit cycle.

  For example: for 01001110 series, represent this bit string as code Manchester and Different Manchester. Let's assume the initial positive voltage.

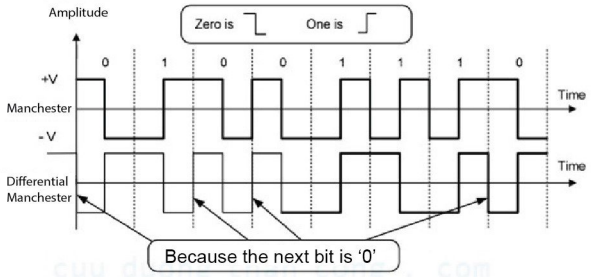


Figure 1.10: Different Manchester.

1.2.8 AMI (Alternate Mark Inversion)

- Characteristics:
  - Bit '0 → 0 Volt.
  - Bit '1 → voltage -V or + V alternately (1 bit cycle exists).
    For example: for 01001110 data series, represent this bit string as AMI code.

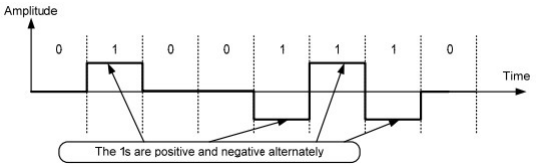


Figure 1.11: AMI (Alternate Mark Inversion).

- The variant of this method is called pseudo-ternary whereby bits 0 receive positive and negative voltage values, respectively.
- Advantages:
  - AMI destroys the DC component of the signal.
  - Synchronize for the sequence of consecutive "1" bit values.
- Disadvantages:
  - Easy to lose sync to the sequence of "0" bit values in a row.

Chapter 2

Overview of image file format

2.1 Digital image concept

Digital images are made up of hundreds of thousands to millions of very small squares – considered elements of the photo and are often known as pixels [3]. A computer or printer uses these small squares to display or print out a photo. To do that, the computer or printer divides the screen, the page into a network containing squares, then uses the values contained in the image file to determine the color, the dark brightness of each pixel in that network - the digital image is formed.

2.1.1 JPEG image format

a. Define

- JPEG stands for Joint Photographic Experts Group, an organization that creates the graphic image format. JPEG uses an algorithm that compresses information loss. JPEG images have a maximum of 16 million colors (24-bit), providing information that specifies the rate and level of compression [3].

- JPEG is one of the most effective methods of compressing images, with a rate of compressing images up to several dozen times. However, the image after unpacking will be different from the original photo. The image quality is degraded after decompression. This decline increases with the compression factor. However, this loss of information is acceptable and the removal of unnecessary information is based on studies of the human eye's eye system [3].

- Extensions of JPEG files are usually in the form of .jpeg, .jfif, .jpg, .JPG, or .JPE. the .jpg form is the most commonly used form. Currently, JPEG image compression is very popular in mobile phones as well as small storage devices .

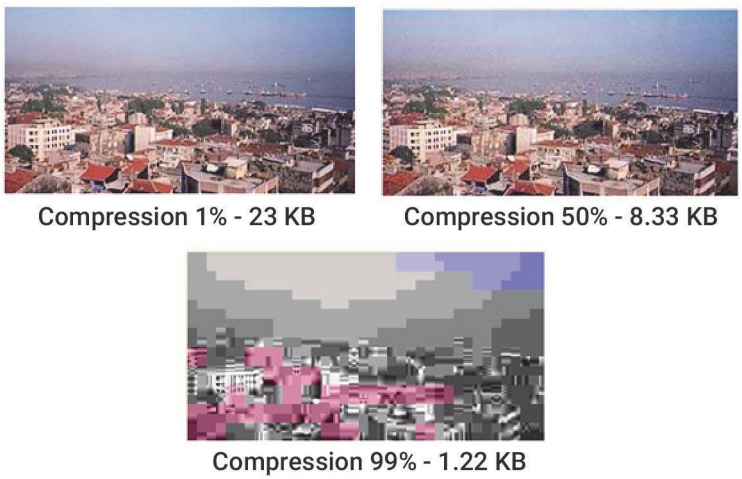


Figure 2.1: Compression capabilities of JPEG format.

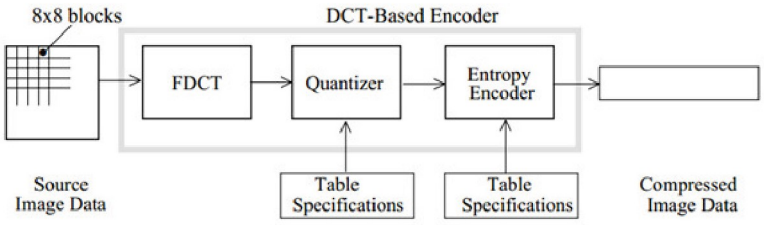
b. Encode

- The main step is to break the image down into small areas (usually 8x8 pixels) and then use discrete cosin transformation to transform these expression zones into a matrix with 64 coefficients that represent the "reality" of pixels. It is important that here the first coefficient is most likely to represent the "status quo", which decreases very quickly with other coefficients. In other words, the amount of information of 64 pixels is concentrated mainly in some matrix coefficients according to the above variation.

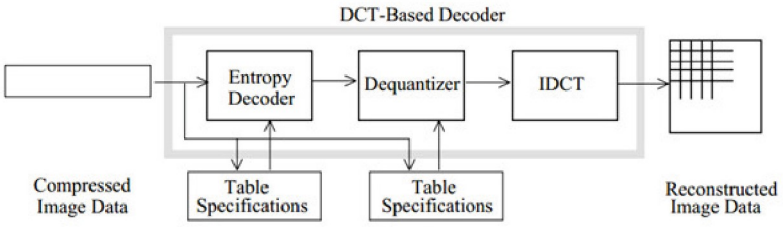
- During this period there is a loss of information, because there is no precise reverse transformation. But this amount of lost information is not significant compared to the next stage. The matrix received after the discrete cosin transformation is reduced to the difference between the coefficients. This is the time to lose a lot of information because people will throw away small changes in the coefficients. So when we open the compressed image, we get other parameters of the pixels. The above variations apply to the U and V components of the image to a

higher degree than Y. Then apply Gernot Hoffman's coding method: analyzing the sequence of numbers, the more repeating elements are encoded by the short symbol (marker). When deploying photos, one simply reworks the above steps in the opposite process along with the reverse transformations.

- The biggest disadvantage is that the image quality has been compressed (lossy), some of the lines between the color blocks will appear blurry points, and the areas will lose clarity. And like mp3, JPEG will not be able to recover the same as the original image even though the capacity is increased to the same as the actual image capacity.



a. DCT Ecoding Progress



b. DCT Reversed Transformation Progress

Figure 2.2: DCT and DCT inverse conversion process.

2.1.2 JPEG compressed image assessment

a. Advantages

- There are different rates of data loss compression.
- Can be compressed with a very high compression rate.
- Offers 24-bit color (16 million colors).

- The best format for photos requires a variety of natural colors (consecutive colors in the color palette) such as landscape shots, photos and illustrations on books, 3-dimensional photos.
- It is possible to download each part of the photo online at different times (progressive JPEGs").

b. Disadvantage

- Unable to index color in color palette.
- No support for creating transparent background images.
- Compress data loss.
- Not every computer displays well, fixed 24-bit color.

Chapter 3

Simulation and results

In this chapter, we will use MATLAB command lines to demonstrate the process of image transmission.

As it can be seen from the code in the next page, the receive structure analyze the information of the image from the transmitter to decode it properly. But in reality, the receiver cannot know what kind of image the transmitter has transmitted.

The image resolution is 316 x 347 correspond to height and width respectively, that's mean we have 316 * 347 = 109652 pixels in gray-scale, each pixel have the range from 0 to 255 in bits.

After import the image to the code the result can be seen from bellow:

Figure 3.1: Input and output image.

```
clear all;
clc;
A = imread('low_res.jpg'); % dinh dang png hoac jpg
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
    binvecc = logical(dec2bin(x, 8) - '0');
end
bit_steam=reshape(binvecc',1,[]);

writematrix(bit_steam','data.txt');
%xuat ra chuoi bit nhi phan trong file text
type data.txt;
writematrix(dec2bin(size(B,1)),'height.txt');
%xuat ra chuoi bit nhi phan chieu cao cuaanh
type width.txt;
writematrix(dec2bin(size(B,2)),'width.txt');
%xuat ra chuoi bit nhi phan chieu rong cua anh
type height.txt;

% Ma hoa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Khoi phuc

if(B(1,1)>255)
    bits=reshape(bit_steam',16,[])';
    w=uint16(bin2dec(num2str(bits)));
    y=reshape(w',size(B,2),size(B,1))';
    C = mat2gray(y);
    imwrite(C,'output.png')
    imshow(C);
else
    bits=reshape(bit_steam',8,[])';
    w=uint8(bin2dec(num2str(bits)));
    y=reshape(w',size(B,2),size(B,1))';
```

```
    C = mat2gray(y);
    imwrite(C,'output.jpg')
    imshow(C);
end
```

In an actual system, every factor of the image including resolution, pixels, and colors must be format into bits in order to convey through the channel.

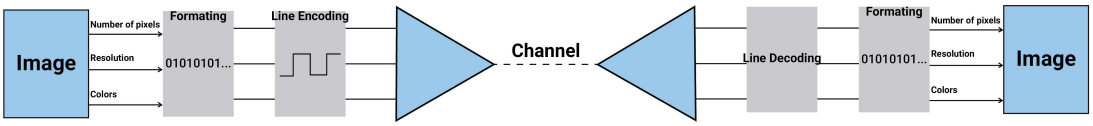


Figure 3.2: Block diagram of Image transmission using Line coding.

If we combine with Line coding, the process is still the same, but in the middle of the process, Symbols of line coding has been form.

3.1 Line coding Forms

3.1.1 NRZ-L

```
%Ma hoa
clear all;
clc;
A = imread('low_res.jpg');
B=rgb2gray(A);
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
    binvecc = logical(dec2bin(x, 8) - '0');
end
bits=reshape(binvecc',1,[]);

%bits = [1 0 1 1 1 0 0 1];
bitrate = 1;
```

```
n = 1000;
T = length(bits)/bitrate;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t));
for i=1:length(bits)
  if bits(i)==1
    x((i-1)*n+1:i*n) = 1;
  else x((i-1)*n+1:i*n) = -1;
  end
end
plot(t, x, 'Linewidth', 3);
if(B(1,1)>255)
    xlim([0 16]);
else
    xlim([0 8]);
end

%Giai ma
counter = 0;
for i = 1:length(t)
  if t(i)>counter
    counter = counter + 1;
    if x(i)>0
      result(counter) = x(i);
    else result(counter) = 0;
    end
  end
end
disp('NRZ-L Decoding:');
disp(result);

writematrix(result','data.txt');
%Xuat chuoi bit cua hinh anh
writematrix(dec2bin(size(B,1)),'height.txt');
%Xuat chuoi bit chieu cao anh
writematrix(dec2bin(size(B,2)),'width.txt');
%Xuat chuoi bit chieu rong anh
```

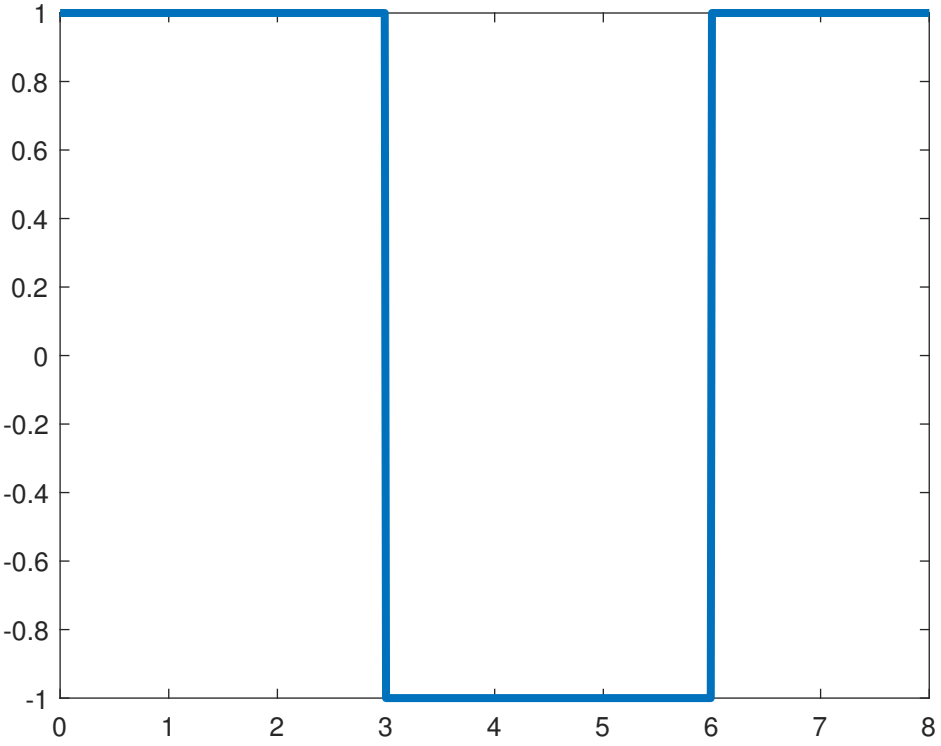


Figure 3.3: Waveform of NRZ-L.

3.1.2 NRZ-I

```
%Ma hoa
clear all;
clc;
A = imread('low_res.jpg');
B=rgb2gray(A);
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
```

```
    binvecc = logical(dec2bin(x, 8) - '0');
end
bits=reshape(binvecc',1,[]);

%bits = [1 0 1 1 1 0 0 1];
bitrate = 1;
n = 1000;
T = length(bits)/bitrate;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t));
lastbit = 1;
for i=1:length(bits)
  if bits(i)==1
    x((i-1)*n+1:i*n) = -lastbit;
    lastbit = -lastbit;
  else x((i-1)*n+1:i*n) = lastbit;
  end
end
plot(t, x, 'Linewidth', 3);
if(B(1,1)>255)
    xlim([0 16]);
else
    xlim([0 8]);
end

%Giai ma
counter = 0;
lastbit = 1;
for i = 1:length(t)
  if t(i)>counter
    counter = counter + 1;
    if x(i)~=lastbit
      result(counter) = 1;
      lastbit = -lastbit;
    else result(counter) = 0;
    end
  end
end
disp('NRZ-I Decoding:');
```

```
disp(result);

writematrix(result',data.txt');
%Xuat chuoi bit cua hinh anh
writematrix(dec2bin(size(B,1)),height.txt');
%Xuat chuoi bit chieu cao anh
writematrix(dec2bin(size(B,2)),width.txt');
%Xuat chuoi bit chieu rong anh
```

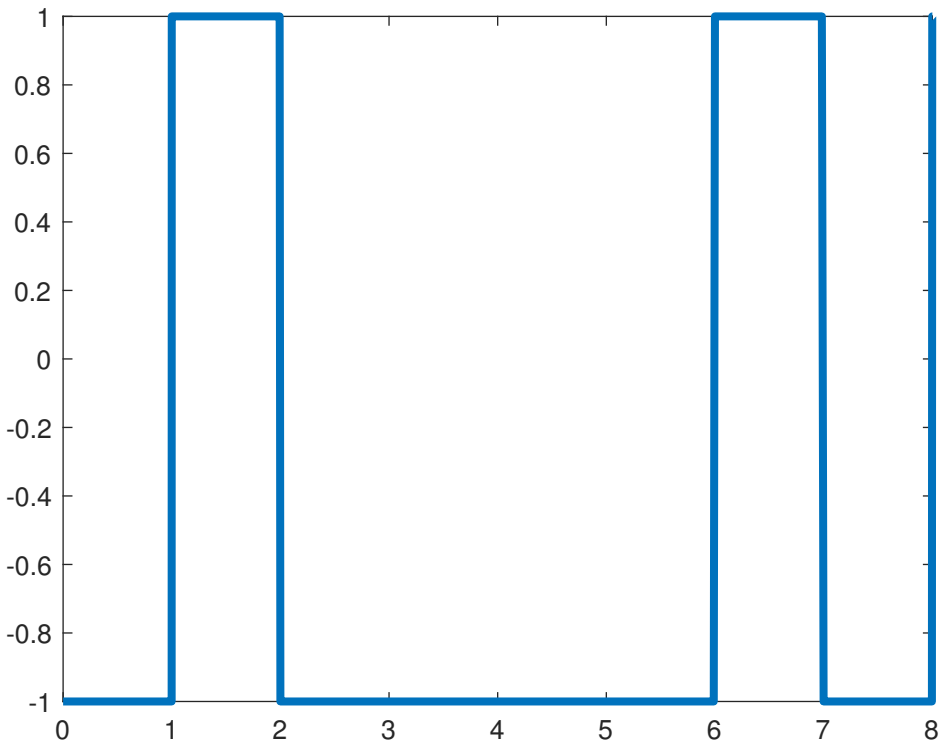


Figure 3.4: Waveform of NRZ-I.

3.1.3 Manchester

```
%Ma hoa
clear all;
clc;
A = imread('low_res.jpg');
B=rgb2gray(A);
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
    binvecc = logical(dec2bin(x, 8) - '0');
end
bits=reshape(binvecc',1,[]);

%bits = [1 0 1 1 1 0 0 1];
bitrate = 1;
n = 1000;
T = length(bits)/bitrate;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t));
for i=1:length(bits)
  if bits(i)==1
    x((i-1)*n+1:(i-1)*n+n/2) = 1;
    x((i-1)*n+n/2:i*n) = -1;
  else
    x((i-1)*n+1:(i-1)*n+n/2) = -1;
    x((i-1)*n+n/2:i*n) = 1;
  end
end
plot(t, x, 'Linewidth', 3);
if(B(1,1)>255)
    xlim([0 16]);
else
```

```
    xlim([0  8]);
end

%Giai ma
counter = 0;
for i = 1:length(t)
  if t(i)>counter
    counter = counter + 1;
    if x(i)>0
      result(counter) = x(i);
    else result(counter) = 0;
    end
  end
end
disp('Manchester Decoding:');
disp(result);

writematrix(result',data.txt');
%Xuat chuoi bit cua hinh anh
writematrix(dec2bin(size(B,1)),'height.txt');
%Xuat chuoi bit chieu cao anh
writematrix(dec2bin(size(B,2)),'width.txt');
%Xuat chuoi bit chieu rong anh
```

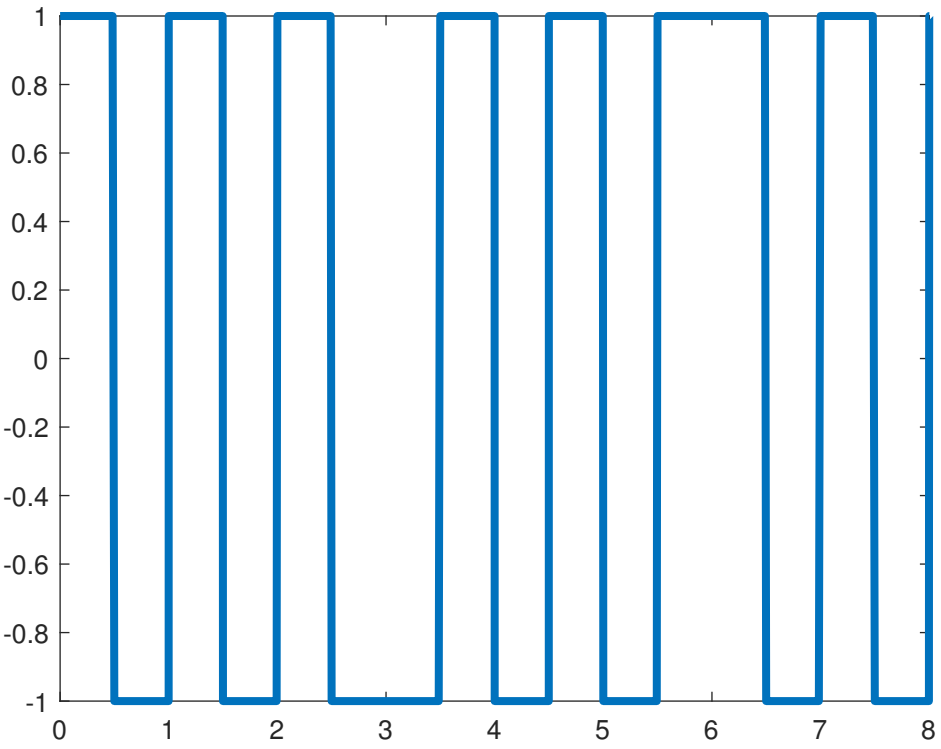


Figure 3.5: Waveform of Manchester.

3.1.4 Differential Manchester

```
%Ma hoa
clear all;
clc;
A = imread('low_res.jpg');
B=rgb2gray(A);
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
```

```
    binvecc = logical(dec2bin(x, 8) - '0');
end
bits=reshape(binvecc',1,[]);

%bits = [1 0 1 1 1 0 0 1];
bitrate = 1;
n = 1000;
T = length(bits)/bitrate;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t));
lastbit = 1;
for i=1:length(bits)
  if bits(i)==0
    x((i-1)*n+1:(i-1)*n+n/2) = -lastbit;
    x((i-1)*n+n/2:i*n) = lastbit;
  else
    x((i-1)*n+1:(i-1)*n+n/2) = lastbit;
    x((i-1)*n+n/2:i*n) = -lastbit;
    lastbit = -lastbit;
  end
end
plot(t, x, 'Linewidth', 3);
if(B(1,1)>255)
    xlim([0 16]);
else
    xlim([0 8]);
end

%Giai ma
counter = 0;
lastbit = 1;
for i = 1:length(t)
  if t(i)>counter
    counter = counter + 1;
    if x(i)==lastbit
      result(counter) = 1;
      lastbit = -lastbit;
    else result(counter) = 0;
    end
```

```
    end
end
disp('Differential Manchester Decoding:');
disp(result);

writematrix(result',data.txt');
%Xuat chuoi bit cua hinh anh
writematrix(dec2bin(size(B,1)),'height.txt');
%Xuat chuoi bit chieu cao anh
writematrix(dec2bin(size(B,2)),'width.txt');
%Xuat chuoi bit chieu rong anh
```

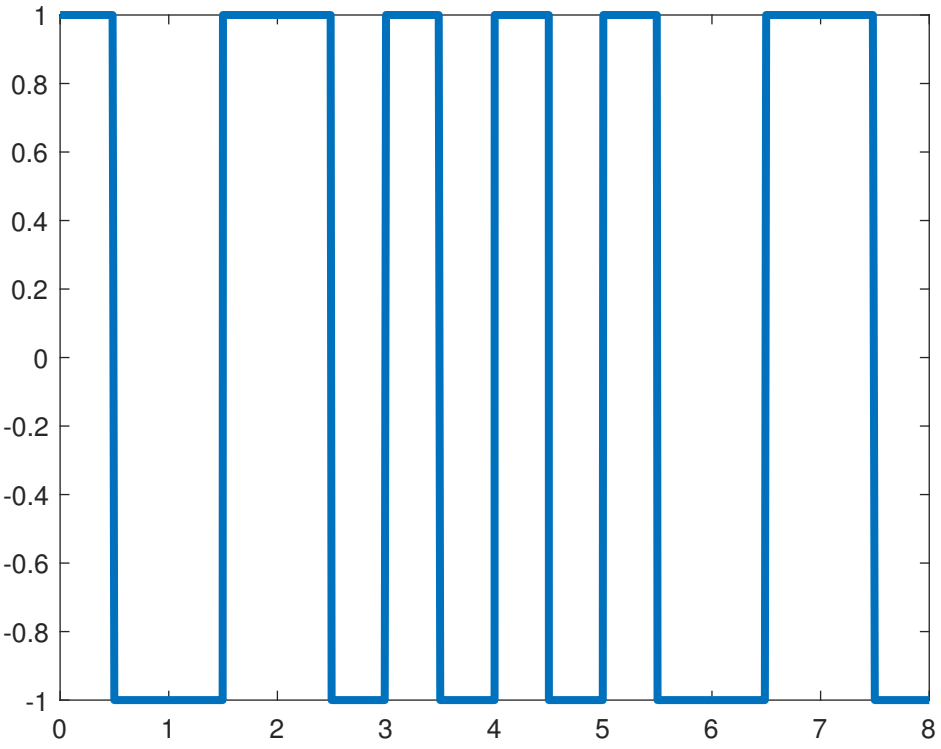


Figure 3.6: Waveform of Differential Manchester.

3.1.5 Return to Zero

```
%Ma hoa
clear all;
clc;
A = imread('low_res.jpg');
B=rgb2gray(A);
if (size(A,3)==3)
    B=rgb2gray(A);
else
    B=A;
end
x=reshape(B',[],1);
if(B(1,1)>255)
    binvecc = logical(dec2bin(x, 16) - '0');
else
    binvecc = logical(dec2bin(x, 8) - '0');
end
bits=reshape(binvecc',1,[]);

%bits = [1 0 1 1 1 0 0 1];
bitrate = 1;
n = 1000;
T = length(bits)/bitrate;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t));
for i=1:length(bits)
  if bits(i)==1
    x((i-1)*n+1:(i-1)*n+n/2) = 1;
  else x((i-1)*n+1:(i-1)*n+n/2) = -1;
  end
end
plot(t, x, 'Linewidth', 3);
if(B(1,1)>255)
    xlim([0 16]);
else
    xlim([0 8]);
end
```

```
%Giai ma
counter = 0;
for i = 1:length(t)
  if t(i)>counter
    counter = counter + 1;
    if x(i)>0
      result(counter) = x(i);
    else result(counter) = 0;
    end
  end
end
disp('RZ Decoding:');
disp(result);

writematrix(result','data.txt');
%Xuat chuoi bit cua hinh anh
writematrix(dec2bin(size(B,1)),'height.txt');
%Xuat chuoi bit chieu cao anh
writematrix(dec2bin(size(B,2)),'width.txt');
%Xuat chuoi bit chieu rong anh
```

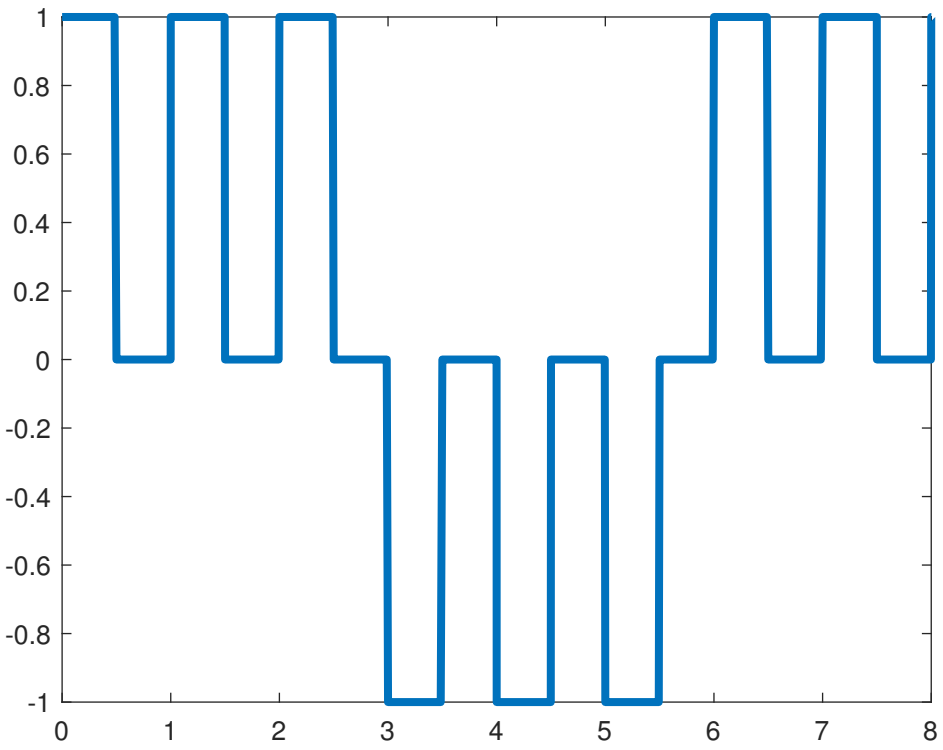


Figure 3.7: Waveform of Return to Zero.

3.2 Recover the image

This is the important base step to recover the image completely, After we have the exported bits data from the text files, we can recover it using algorithms, PNG and JPEG are different in types of format, so it need to be considered to use the right recover method, with JPEG image we'll use:

```
fid = fopen('data.txt');
z=textscan(fid,'%s');

h=fopen('height.txt');
he=textscan(h,'%s');
w=fopen('width.txt');
wi=textscan(w,'%s');

hei=char(he{1});
wid=char(wi{1});

heig=logical(hei(:)'-'0');
widt=logical(wid(:)'-'0');

height=double(bin2dec(num2str(heig)));
width =double(bin2dec(num2str(widt)));

variables = str2mat(z{1, 1});
bit_steam=logical(variables-'0')';
bits=reshape(bit_steam',8,[])';
w=uint8(bin2dec(num2str(bits)));
y=reshape(w',width,height)';
C = mat2gray(y);
imwrite(C,'output.jpg');
imshow(C);
```

The output.jpg is the result of after recovery. Although the image was still in gray-scale, but the precise of the system when sorting all the bits together to form into a black and white picture can tell how significant it is.

Figure 3.8: Output image after recovery.

References

[1] https://www.slideshare.net/tht040492/cc-loi-m-ng-truyn-v-ng-dng-neptune
Kyūyo kurejitto. Types of Line Coding and Neptune Application.

[2] https://cuuduongthancong.com/pvf/1006239/ky-thuat-truyen-so-lieu/nguyen-viet-hung/chuong-5_ma-hoa-va-dieu-che.pdf,
Nguyen Viet Hung. Chapter 5 - Coding and Decoding.

[3] https://123docz.net/document/3111767-tong-quan-ve-dinh-dang-file-anh\-va-phuong-phap-dinh-dang-file-anh.htm,
Nguyen Trung Kien, Ngo Tuan Linh, La Hung Anh, 2015. Overview of image formats and formatting methods. Posts and Telecommunications Institute of Technology, Ha Noi.