Data Communication Project Report  
2nd Year Computer Engineering

Project Title: [ Polar NRZ-Space]

# 

# Team Members

|  |  |  |
| --- | --- | --- |
| **SN** | **Student Name** | **Section Number** |
| **1** | **Mohamed Sameh Mohamed** | **3** |
| **2** | **Omar Ashraf Helmy** | **2** |
| **3** | **Abdelmoez Ashraf Abdallah** | **2** |
| **4** | **Abdelrahman Elbadwy fawzy** | **2** |
| **5** | **Abdelrahman Ashraf Mahmoud** | **2** |

# 

# Overview

NRZ (S) Figure(1)modulation is an enhanced version of the Non-Return to Zero (NRZ) signaling scheme, designed to address the limitations of conventional NRZ modulation in digital communication systems. This report focuses on the practical implementation of NRZ (S) modulation, including the encoding and decoding processes using both Proteus simulation and MATLAB software.

The encoding process involves generating the NRZ (S) signal using Proteus simulation and MATLAB software. Proteus offers a comprehensive platform for simulating and visualizing the encoded signal, enabling an evaluation of its quality and performance, and MATLAB. MATLAB provides powerful tools for signal processing and analysis, facilitating the decoding of the encoded signal. The report outlines the step-by-step procedure for encoding the signal in Proteus and MATLAB.

To further assess the decoding process, the report demonstrates how to recover the original binary data from the NRZ (S) signal using the same two techniques used in encoding, Different decoding techniques are explored.

Overall, this report aims to provide a comprehensive understanding of the encoding and decoding processes involved in NRZ (S) modulation, in figure 1. By utilizing Proteus simulation and MATLAB decoding, researchers can effectively evaluate the signal quality and recover the original binary data. The insights presented herein contribute to the practical utilization of NRZ (S) modulation techniques, enabling improved data transmission rates and system performance in modern digital communication systems.

Polar NRZ-SPACE (Non-Return-to-Zero, SPACE) is a line coding scheme used in digital communication systems. It has its advantages and disadvantages, which are outlined below:

**Advantages of Polar NRZ-SPACE:**

1. Improved DC Balance: Polar NRZ-SPACE ensures a better DC balance compared to other line coding schemes like Polar NRZ (Non-Return-to-Zero). It helps in maintaining a more balanced distribution of positive and negative signal levels, reducing the accumulation of DC offset.

2. Simplicity: Polar NRZ-SPACE is relatively simple to implement compared to more complex line coding schemes. It requires only two signal levels (positive and negative), making it easier to encode and decode.

**Disadvantages of Polar NRZ-SPACE:**

1. Lack of Self-clocking: Polar NRZ-SPACE does not provide inherent self-clocking properties. It means that the receiver needs to have an external clock reference or use additional mechanisms to recover the clock signal from the encoded data. This can increase complexity and the possibility of synchronization errors.

2. Vulnerability to Long Runs of 1s: Polar NRZ-SPACE can experience problems when there are long runs of 1s in the data. Long sequences of the same bit value can cause difficulties in clock recovery and synchronization, leading to potential errors in data transmission.

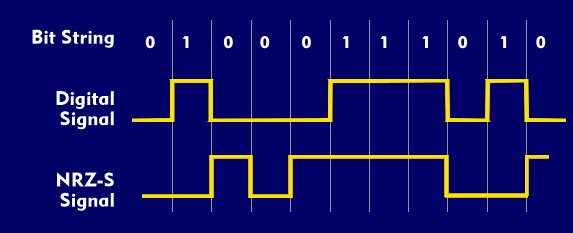


Figure NRZ(S) decoding

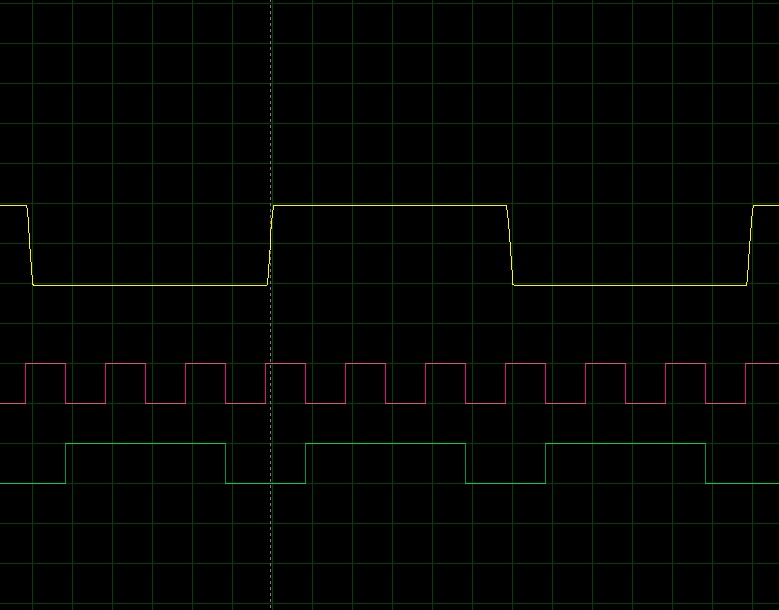
# Proteus Simulation

## Encoding

A diagram of a circuit

Description automatically generated with low confidence

Figure 2 NRZ(S) encoding cicuit

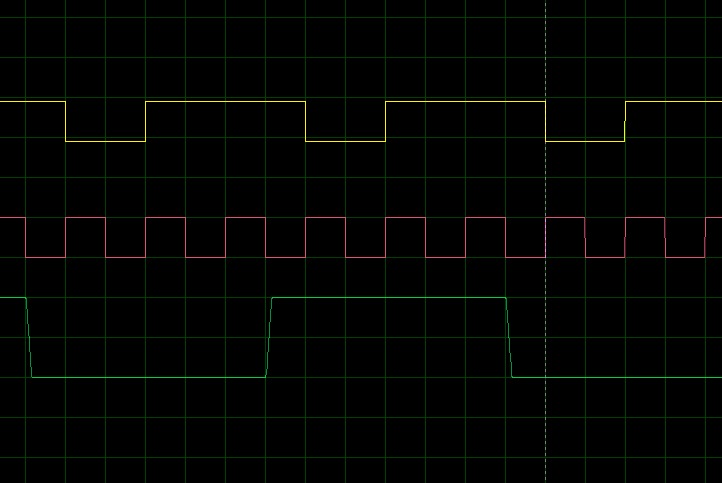


## Decoding

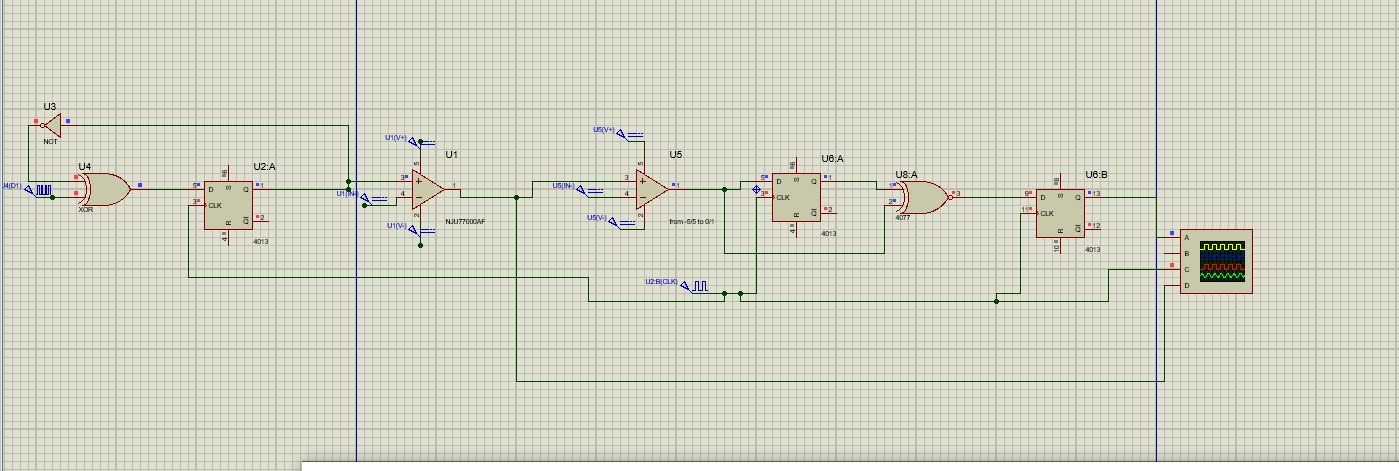
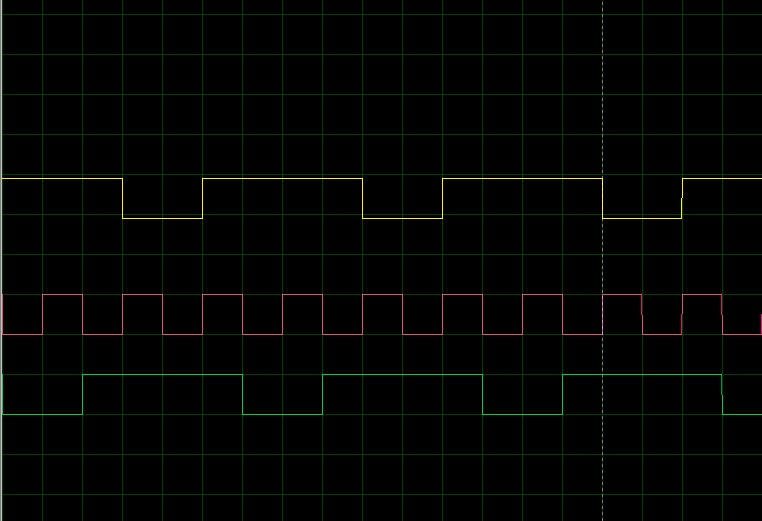
A diagram of a decoding circuit

Description automatically generated

Figure 3 NRZ(S) decoding circuit



## Encoding + Decoding



# MATLAB

* 1. Encoding

1. %--------------------------Get NRZ-S--------------------------
2. %Clear Command Window
3. clc
4. % Binary data sequence
5. input = [1 0 0 1 1 1 0 1];
6. % NRZ-I encoding
7. nrz\_i\_encoded = 2 \* input - 1; % Convert binary 0 to -1, binary 1 to +1
8. % NRZ-I signal
9. nrz\_i\_signal = [];
10. % Set initial level
11. level = 1; % Assume starting with a high level
12. for i = 1:length(nrz\_i\_encoded)
13. if nrz\_i\_encoded(i) == 1 % If binary 1
14. nrz\_i\_signal = [nrz\_i\_signal , level\*ones(1, 100)]; % Maintain the current level for 10 samples
15. else % If binary 0
16. level = -level; % Invert the level
17. nrz\_i\_signal = [nrz\_i\_signal , level\*ones(1, 100)]; % Set the new level for 10 samples
18. end
19. end
20. % Create time vector for the signal
21. t = 0:0.01:length(nrz\_i\_signal)\*0.01-0.01;
22. % Plot the NRZ-I signal
23. figure;
24. plot(t, nrz\_i\_signal, 'LineWidth', 2);
25. xlabel('Time');
26. ylabel('Amplitude');
27. title('NRZ-I Line Coding');
28. axis([0 , length(nrz\_i\_signal)\*0.01-0.01 , -1.5 , 1.5]);
29. grid on;

This MATLAB code generates an NRZ-S (Non-Return to Zero - Space) encoded signal based on a given binary data sequence. The code follows the NRZ-I (Non-Return to Zero - Inverted) encoding scheme as a basis for generating the NRZ-S signal.

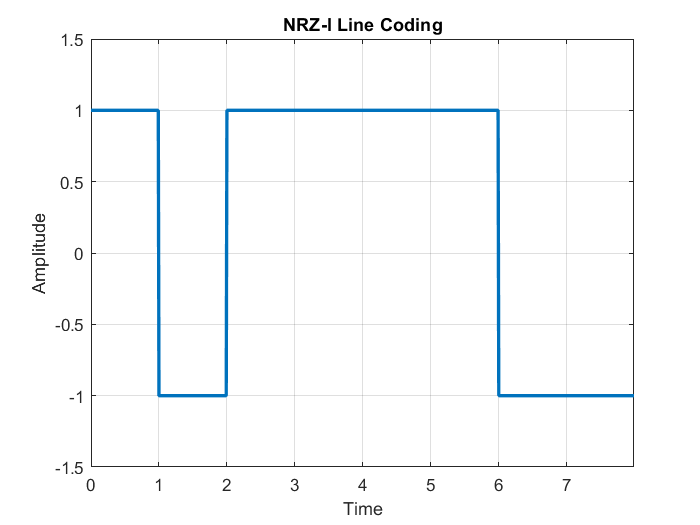
The binary data sequence is represented by the "input" variable, which is a vector of ones and zeros. The code then performs NRZ-I encoding by converting the binary 0s to -1s and binary 1s to +1s, using the equation nrz\_i\_encoded = 2 \* input - 1.

The NRZ-I encoded signal is stored in the "nrz\_i\_signal" variable. It constructs the signal by iterating through the encoded sequence and determining the level of the signal based on the current encoded value. If the value is 1, the level is maintained for a duration of 100 samples. If the value is 0, the level is inverted, and the new level is maintained for 100 samples. This process is repeated for the entire encoded sequence.

A time vector "t" is created to represent the time axis for the signal. It is based on the sampling rate of 0.01 and the length of the generated NRZ-I signal.

The code then plots the NRZ-I signal using the generated time vector and the NRZ-I signal values. The resulting plot shows the amplitude of the NRZ-I signal over time.

Overall, this code provides a visual representation of the NRZ-I signal, which serves as a basis for generating the NRZ-S signal.



* 1. Decoding

1. %--------------------------Get the orignal code back--------------------------
2. data = [1 1 -1 1 1 1 1 -1 -1]; % Input vector NRZ-S
3. output = [1]; % Initialize the output vector
4. % Compare current element with previous element
5. for i = 2:length(data)
6. if data(i) == data(i-1)

output = [output , 1\*ones(1,100)]; % Set output to 1 if equal

1. else

output = [output , 0\*ones(1,100)];

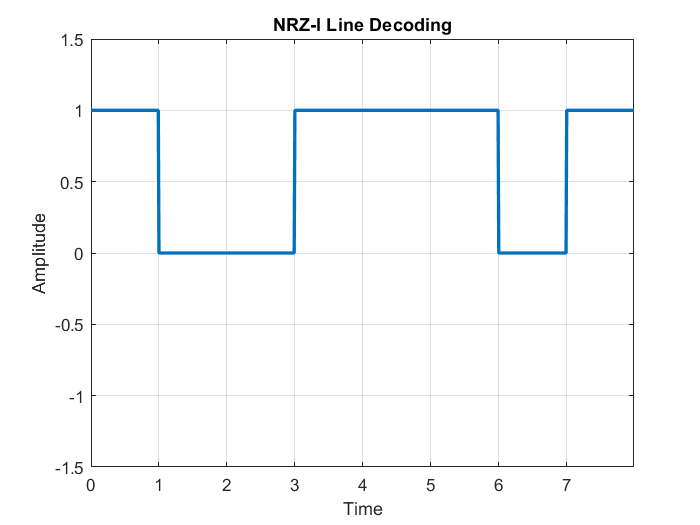
1. end
2. end
3. % Display the output vector
4. disp('The orignal code:');
5. disp(output);
6. % Create time vector for the signal
7. t = 0:0.01:length(output)\*0.01-0.01;
8. % Plot the NRZ-I signal
9. figure;
10. plot(t, output, 'LineWidth', 2);
11. xlabel('Time');
12. ylabel('Amplitude');
13. title('NRZ-I Line Decoding');
14. axis([0 , length(output)\*0.01-0.01 , -1.5 , 1.5]);
15. grid on;

This MATLAB code decodes an NRZ-S (Non-Return to Zero - Space) encoded signal and recovers the original binary data. The code takes an input vector, "data," which represents the encoded NRZ-S signal.

The decoding process starts by initializing an output vector, "output," with the first element set to 1. The code then compares each element of the input vector with its previous element. If the current element is equal to the previous element, it implies that the original binary data was a 1. In this case, the code appends a sequence of 1s, representing the original data, to the output vector. Otherwise, if the elements are different, the code appends a sequence of 0s to the output vector.

The output vector represents the recovered original binary data, and it is displayed using the "disp" function.

To visualize the decoding process, the code creates a time vector, "t," based on the sampling rate of 0.01 and the length of the output vector. It then plots the decoded NRZ-I signal using the time vector as the x-axis and the output vector as the y-axis. The resulting plot shows the amplitude of the decoded NRZ-I signal over time.

In summary, this code demonstrates the process of decoding an NRZ-S encoded signal and recovering the original binary data. It provides a visual representation of the decoded signal, aiding in the understanding of the decoding process.