Tribhuvan University

Institute of Engineering

Thapathali Campus, Thapathali

Telecommunication Lab 3

Submitted By:

Name: Raman Bhattarai

Roll No.: THA077BEI033

Submitted To:

Department of Electronics and Computer Engineering

Date: Feb 28, 2025

Write a MATLAB Function for Differential Manchester line coding scheme.

Line coding refers to the process of converting binary data (1s and 0s) into a digital signal for transmission over a communication medium. It defines how the digital information is represented in terms of voltage or current levels in the transmitted signal. The primary goals of line coding are to ensure synchronization between the sender and receiver, minimize errors, and achieve efficient bandwidth utilization. Common types of line coding schemes include Non-Return-to-Zero (NRZ)**,** Manchester**,** and Differential Manchester encoding.

Differential Manchester encoding is a self-clocking line coding technique where each bit period has a transition in the middle to maintain synchronization. The key feature is that the transition at the midpoint of each bit ensures that the receiver can synchronize to the clock signal. Additionally, if the data bit is a 0, there is an extra transition at the beginning of the bit period, while a 1 results in no additional transition at the start. This encoding scheme improves error detection and synchronization and is widely used in Ethernet (IEEE 802.3) due to its robustness against DC components and clock drift.

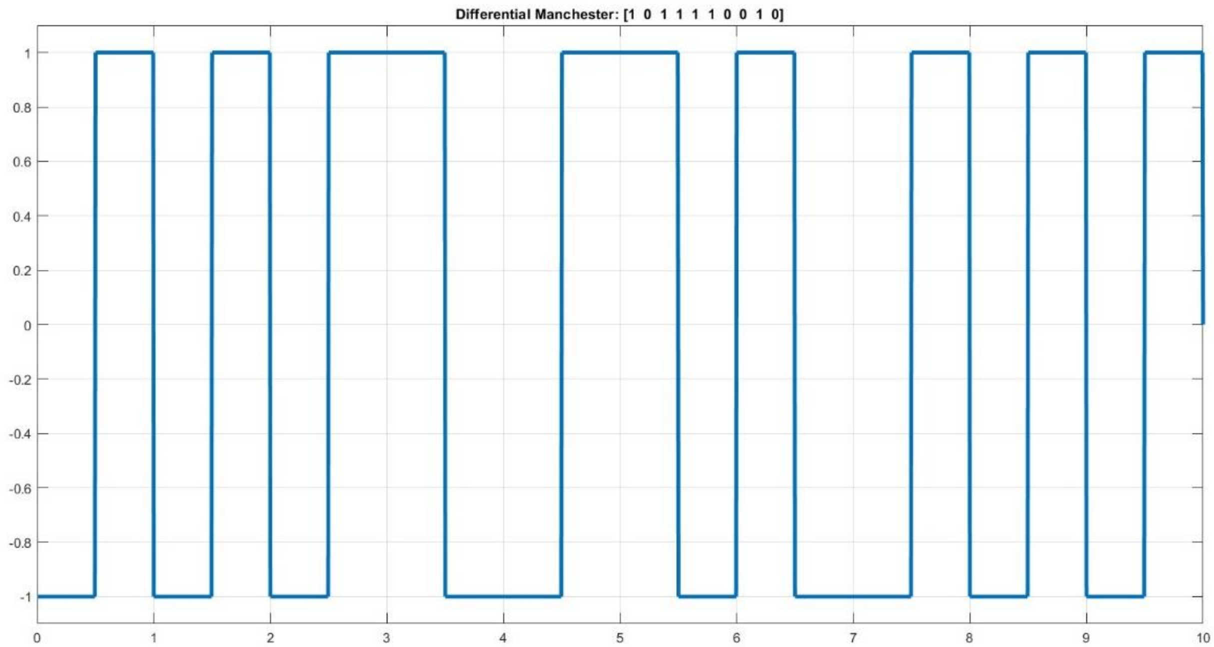
Source Code:

```
function [t, x] = diffmanch(bits, bitrate)
    T = length(bits) / bitrate; % Total time of the bit sequence
    n = 200; % Number of samples per bit
    N = n * length(bits); % Total number of samples
    dt = T / N; % Time resolution
    t = 0:dt:T; % Time vector
    x = zeros(1, length(t)); % Output signal

    % Initialize the polarity (start high or low)
    current_level = 1;

    for i = 0:length(bits) - 1
        if bits(i + 1) == 1
            % For bit "1": Transition at the start of the bit period
            x(i * n + 1:(i + 0.5) * n) = -current_level;
            x((i + 0.5) * n + 1:(i + 1) * n) = current_level;
        else

            x(i * n + 1:(i + 0.5) * n) = current_level;
            x((i + 0.5) * n + 1:(i + 1) * n) = -current_level;
        end

        current_level = -current_level;
    end
end
```

Calling function:

```
bits = [1 0 1 1 1 1 0 0 1 0];
bitrate = 1;
figure;
[t, s] = diffmanch(bits,bitrate);
plot(t,s,'LineWidth',3);
axis([0 t(end) -1.1 1.1]);
grid on;
title(['Differential Manchester: [' num2str(bits) ']'])
```

Output:



Conclusion:

The MATLAB function successfully implements Differential Manchester Encoding, a self-clocking line coding scheme widely used in data communication systems. By ensuring a mid-bit transition for every bit and an additional transition for binary 0, the method maintains synchronization and improves noise immunity. The function efficiently converts a binary data sequence into an encoded waveform and visually represents the signal. This encoding scheme is particularly useful in applications like Ethernet (IEEE 802.3), where clock synchronization and error reduction are crucial.