American International University- Bangladesh (AIUB)
Faculty of Engineering
Data Communications Lab
Open Ended Lab (OEL)

Course Name:	**Data Communication**		
Course Code:	EEE 3205	**Section:**	G
Semester:	Fall 2023-24	**Group No:**	

Assignment Name:	**Open Ended Lab (OEL)**		
Assessed CO2: **Assessed CO5**	Convert this bit stream to digital signal using different Line Coding Method		
Assessed POI:	**P.d.1.C5**		
Student Name:	NOKIBUL ARFIN SIAM	**Student ID:**	21-44793-1
Student Name:	MD. RAFIUR RAHMAN TURJO	**Student ID:**	21-45055-2
Student Name:	MD MEHEDI HASAN	**Student ID:**	21-45180-2
Student Name:	MUHAMMAD NAZMUS SAADAT	**Student ID:**	21-45157-2
Student Name:	AAMIR MOHAMMED ZOBAYER	**Student ID:**	20-44193-2
Student Name:	FOWSIA JAHAD JASSY	**Student ID:**	20-44148-2

Mark distribution (to be filled by Faculty):

Objectives	Proficient [10-8]	Good [7-4]	Needs Improvement [3-1]	Secured Marks
Depth of knowledge displayed through appropriate research (P1)	Student was able to apply in-depth engineering knowledge achieved by appropriate research about digital/analog communication to design the communication model correctly and **fulfilled all design criteria**.	Design process is not completely supported by in-depth engineering knowledge achieved by appropriate research about digital/analog communication, **some but not all of the design criteria are fulfilled**.	Design process contains mistakes and does not display enough in-depth engineering knowledge achieved by appropriate research about digital/analog communication. **Most of the design criteria are not fulfilled**.	
Depth of analysis (P3)	Student defended the diversified approach taken to solve the problem with **well-justified in-depth analysis that demonstrated abstract thinking**.	Student's attempts to analyze the diversified approach taken to solve the problem **is not enough in-depth, some of design choices do not demonstrate adequate abstract thinking** and are not properly justified.	Student **did not attempt any in-depth analysis** of the designed system and **displayed no abstract thinking**.	
Level of integration of multiple sections of design for solution of high-level problem (P7)	Student correctly identified all problems and successfully integrated the interdependent parts into a high-level design using a block diagram. Block diagram was at best match with the given problem.	Student was able to identify some of the problems correctly and integrated the interdependent parts into a high-level design using a block diagram. Some parts of the block diagram were not a good match for the given problem.	Student was able to identify only one/two of the problems correctly and could not properly integrate the interdependent parts into a high-level design using a block diagram. Only one/two blocks were correct and/or block diagram was incomplete.	
Comments:			**Total Marks (Out of 30):**	

Purpose:

The purpose of this experiment is to explore and implement four different digital data transmission techniques, Polar NRZ-I (Invert), Differential Manchester, Pseudoternary, and MLT-3. Each technique will be simulated in MATLAB, and their performance will be analyzed.

Procedure:

From student ID: **21-45055-2**(AB-CDEFG-H) we take B, E, and G value and make a make a12 bits binary bit stream. We can generate digital signal using this bit stream.

Polar NRZ-I (Invert):

In Polar NRZ-I, the voltage level or polarity of the signal is changed when the data bit is a '1', and there is no change when the data bit is '0'.

The "I" in NRZ-I stands for "inverted," which means that the signal is inverted or flipped when a logical '1' is encountered. The basic idea is to use the inversion of the signal to represent binary information.

For Polar NRZ-I:

- Logical '0': No change in the signal level.
- Logical '1': Invert the signal level.

This inversion of the signal helps in mitigating the problem of long sequences of consecutive zeros or ones, which can be a concern in regular NRZ encoding. In Polar NRZ-I, consecutive '1's will result in signal transitions, helping in clock recovery and synchronization in the receiver.

Differential Manchester:

- Logical '0' is represented by a transition from the current voltage level to the opposite level in the middle of the bit period.
- Logical '1' is represented by no transition in the middle of the bit period.
- At the start of each bit period, there is always a transition, either from a high to a low level or from a low to a high level.
- If the next bit is '1', there is no additional transition in the middle of the bit period.
- If the next bit is '0', there is an additional transition in the middle of the bit period.
-

Pseudoternary:

Pseudoternary is a type of bipolar encoding method used in digital communications to transmit binary data over a communication channel. It is a non-return-to-zero (NRZ) encoding, and it's characterized by the representation of binary '0' as a signal transition and binary '1' as the absence of a signal transition.

In pseudoternary encoding:

- Logical '0': Represented by a signal transition (from positive to negative or from negative to positive).
- Logical '1': Represented by the absence of a signal transition.

MLT-3:

MLT-3, or Multi-Level Transmit 3, is a line coding scheme used in digital communication systems to encode binary data into a waveform for transmission.

- MLT-3 uses three voltage levels: +V, 0V, and -V.
- Logical '0': No change in voltage level.
- Logical '1': Transition to the next voltage level in the sequence.
- MLT-3 utilizes a three-level pattern, ensuring that the signal does not remain at the same level for consecutive '0's.

Calculation & Result:

ID: 21-45055-2

B = 1

F = 5 [we take F cause E = 0]

G = 5

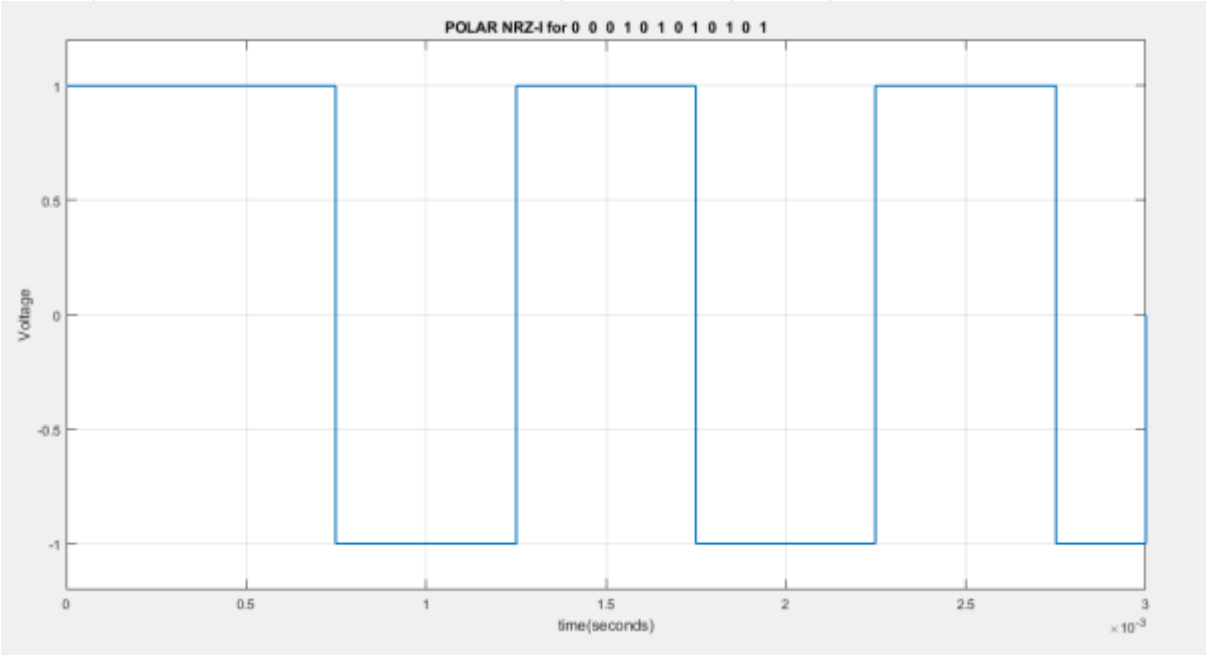
12-bit binary stream,

155 = 0 0 0 1 0 1 0 1 0 1 0 1.

1. Polar NRZ-I (invert) assuming bit rate is 4 kbps.

```
clc
clear all
close all
bit_stream = [0 0 0 1 0 1 0 1 0 1 0 1]; % ID = 21-45055-2
no_bits = length(bit_stream);
bit_rate = 4000; % 4 kbps
pulse_per_bit = 1; % for polar nrz
pulse_duration = 1 / (pulse_per_bit * bit_rate);
no_pulses = no_bits * pulse_per_bit;
samples_per_pulse = 500;
fs = samples_per_pulse / pulse_duration;
t = 0:1/fs:(no_pulses) * pulse_duration;
no_samples = length(t); % total number of samples
dig_sig = zeros(1, no_samples);
voltage_level = 1; % Initial voltage level for NRZ-I
for i = 1:no_bits
    if bit_stream(i) == 0
        dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
voltage_level * ones(1, samples_per_pulse);
    else
        voltage_level = -voltage_level; % Toggle the voltage level for each '1'
        dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
voltage_level * ones(1, samples_per_pulse);
    end
end
plot(t, dig_sig, 'linewidth', 1.5)
grid on
hold on
% Overlay bitstream
bit_time = pulse_duration / pulse_per_bit;
```

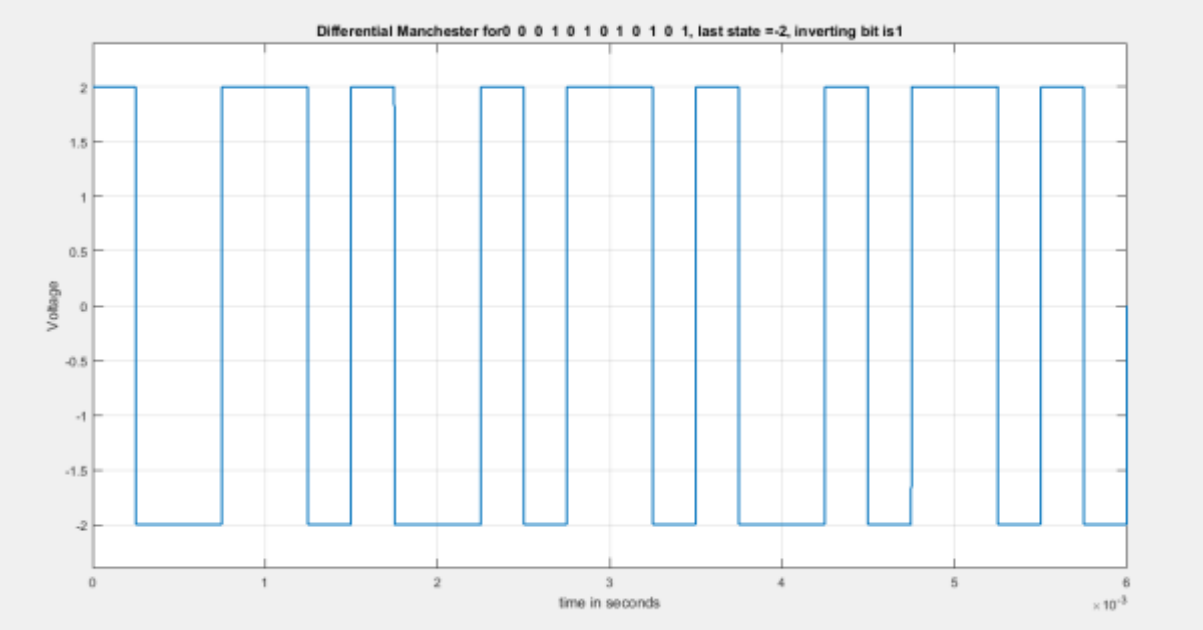
```
bit_stream_offset = 0.5 * bit_time;
for i = 1:no_bits
    text(i * bit_time - bit_stream_offset, 1.5, num2str(bit_stream(i)), 'FontSize',
10);
end
xlabel('time(seconds)')
ylabel('Voltage')
ylim([-1.2 * max(dig_sig) 1.2 * max(dig_sig)])
title(['POLAR NRZ-I for ', num2str(bit_stream), ''])
```



2. Differential Manchester assuming bit rate is 2 kbps.

```
clear all
close all
bit_stream = [0 0 0 1 0 1 0 1 0 1 0 1]; % ID = 21-45055-2
no_bits = length(bit_stream);
bit_rate = 2000; % 2 kbps
pulse_per_bit = 2; % for differential manchester
pulse_duration = 1/((pulse_per_bit)*(bit_rate));
no_pulses = no_bits*pulse_per_bit;
samples_per_pulse = 500;
fs = (samples_per_pulse)/(pulse_duration); %sampling frequency
% including pulse duration in sampling frequency
% ensures having enough samples in each pulse
t = 0:1/fs:(no_pulses)*(pulse_duration); % sampling interval
% total duration = (no_pulse)*(pulse_duration)
no_samples = length(t); % total number of samples
dig_sig = zeros(1,no_samples);
max_voltage = +2;
min_voltage = -2;
inv_bit = 1; % inverting bit
last_state = max_voltage;
inv_last_state = min_voltage; % inverse of last state
for i = 1:no_bits
```

```
j = (i-1)*2;
if bit_stream(i) == inv_bit
dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
inv_last_state*ones(1,samples_per_pulse);
dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
last_state*ones(1,samples_per_pulse);
else
dig_sig((j*(samples_per_pulse)+1):(j+1)*(samples_per_pulse)) =
last_state*ones(1,samples_per_pulse);
dig_sig(((j+1)*(samples_per_pulse)+1):(j+2)*(samples_per_pulse)) =
inv_last_state*ones(1,samples_per_pulse);
temp_cons = last_state; % temporary constant
last_state = inv_last_state;
inv_last_state = temp_cons;
end
end
figure
plot(t,dig_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('Voltage')
ylim([(min_voltage - (max_voltage)*0.2)
(max_voltage+max_voltage*0.2)])
title(['Differential Manchester for',num2str(bit_stream),', last state
=',num2str(last_state),', inverting bit is',num2str(inv_bit),''])
```

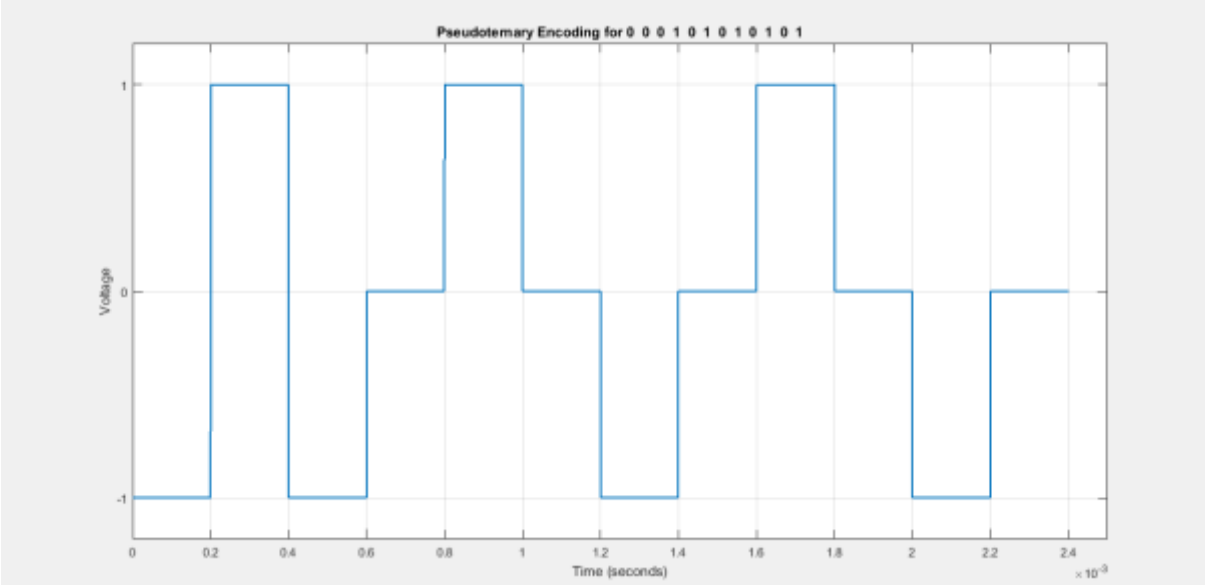


3. Pesudoternary assuming bit rate is 5 kbps.

```
clc
clear all
close all
bit_stream = [0 0 0 1 0 1 0 1 0 1 0 1]; % ID= 21-45055-2
no_bits = length(bit_stream);
bit_rate = 5000; % 5 kbps
pulse_per_bit = 1;
```

```
pulse_duration = 1 / (pulse_per_bit * bit_rate);
no_pulses = no_bits * pulse_per_bit;
samples_per_pulse = 500;
fs = samples_per_pulse / pulse_duration;
t = 0:1/fs:(no_pulses) * pulse_duration;
no_samples = length(t); % total number of samples
dig_sig = zeros(1, no_samples);
last_level = 1; % Assuming the last level for '0' was positive
invert_state = 0; % Initialize inversion state
for i = 1:no_bits
    if bit_stream(i) == 0
        last_level = -last_level; % Invert state for '0' bit
        dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
last_level * ones(1, samples_per_pulse);
        invert_state = last_level; % Record the inversion state
    else
        dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) = 0; %
Set amplitude to 0 for '1'
    end
end
plot(t, dig_sig, 'linewidth', 1.5)
grid on
xlabel('Time (seconds)')
ylabel('Voltage')
ylim([-1.2 1.2])
title(['Pseudoternary Encoding for ', num2str(bit_stream), ''])

% X and Y axis texts
set(gca, 'XTick', 0:pulse_duration:(no_pulses * pulse_duration));
set(gca, 'YTick', [-1 0 1]);
```



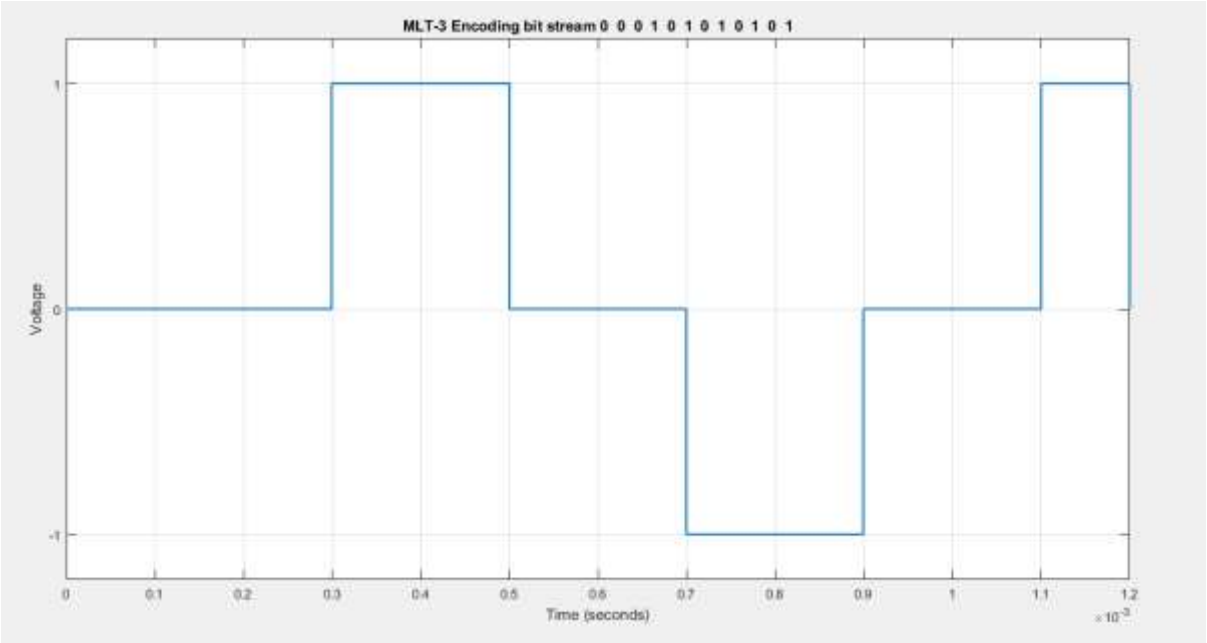
4. MLT-3 assuming bit rate is 10 kbps.

```
clc
clear all
close all
bit_stream = [0 0 0 1 0 1 0 1 0 1 0 1]; % ID= 21-45055-2
no_bits = length(bit_stream);
bit_rate = 10000; % 10 kbps
pulse_per_bit = 1; % for unipolar nrz
pulse_duration = 1 / (pulse_per_bit * bit_rate);
no_pulses = no_bits * pulse_per_bit;
samples_per_pulse = 500;
fs = samples_per_pulse / pulse_duration;
t = 0:1/fs:(no_pulses) * pulse_duration;
no_samples = length(t); % total number of samples
dig_sig = zeros(1, no_samples);
last_non_zero_level = -1; % Last non-zero level is Negative
last_level = 0; % Last level was zero voltage
for i = 1:no_bits
    if bit_stream(i) == 0
        dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
last_level * ones(1, samples_per_pulse); % No change for '0' bit
    else
        if last_level ~= 0 % If the previous level is not zero voltage
            last_level=0;
            dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
last_level * ones(1, samples_per_pulse); % Set amplitude to 0 for '1'
        else
            if last_level == 0 % If the last non-zero level was not zero voltage
                last_non_zero_level = -last_non_zero_level; % Invert the last state
for '1' bit
                dig_sig(((i - 1) * samples_per_pulse + 1):i * (samples_per_pulse)) =
last_non_zero_level * ones(1, samples_per_pulse);
                last_level = last_non_zero_level;
            else

            end
        end
    end
end

plot(t, dig_sig, 'linewidth', 1.5)
grid on
xlabel('Time (seconds)')
ylabel('Voltage')
ylim([-1.2 1.2])
title(['MLT-3 Encoding bit stream ', num2str(bit_stream), ''])

% X and Y axis texts
set(gca, 'XTick', 0:pulse_duration:(no_pulses * pulse_duration));
set(gca, 'YTick', [-1 0 1]);
```



Discussion and Conclusions:

In this experiment, each way of encoding information serves specific purposes in digital communication. Polar NRZ-I and Differential Manchester help with keeping the timing in sync, Pseudoternary is good for balancing voltage levels, and MLT-3 is efficient for using bandwidth wisely. The choice of encoding method depends on what a communication system needs, like how much bandwidth it uses, how well it handles errors, and if it follows certain standards.

Polar NRZ-I and Differential Manchester are good at making sure the timing stays right, while Pseudoternary, even though it's older, helps balance voltage levels. MLT-3, with its three voltage levels, is a modern option that balances efficiency and signal strength, especially in fast data transmission over twisted cables. Knowing about these encoding methods helps make digital communication systems work better in different situations.

Reference:

1. https://www.technologyuk.net/telecommunications/telecom-principles/line-coding-techniques.shtml#ID05

2. Forouzan AB. Data communications & networking. 5th ed., Tata McGraw-Hill Education.

3. https://www.tutorialspoint.com/digital_communication/digital_communication_line_codes.htm