**ENCODING SCHEMES**

**Experiment No: 4 DATE: 04/2/2024**

**Aim:** To study different Encoding schemes.

1. Non-Return-to-Zero (NRZ) Encoding.
2. Manchester Encoding.
3. Differential Manchester Encoding.

**Theory:**

Encoding refers to the process of converting digital data into a format suitable for transmission over a communication medium, such as copper wires or fiber optic cables. The primary goal of encoding is to ensure reliable and accurate data transmission while considering factors such as signal integrity, synchronization, and noise resilience.

Different types of Encoding Schemes

1. Non-Return-to-Zero (NRZ) Encoding:

* In NRZ encoding, each bit is represented by a specific voltage level over the duration of the bit period.
* A high voltage level (e.g., +5V) may represent a 1, while a low voltage level (e.g., 0V) represents a 0.
* NRZ encoding does not utilize transitions within the bit period, which can simplify hardware implementation.
* However, NRZ encoding is susceptible to long runs of consecutive 0s or 1s, which can cause synchronization and clock recovery issues.
* It's worth noting that NRZ can be further categorized into NRZ-L (Non-Return-to-Zero-Level) and NRZ-I (Non-Return-to-Zero-Inverted).

1. Manchester Encoding:

* In Manchester encoding, each bit of the digital signal is represented by a transition in the middle of a bit period.
* A high-to-low transition (also called a falling edge) represents a 1, while a low-to-high transition (rising edge) represents a 0.
* This encoding scheme ensures both clock recovery and data recovery, making it self-clocking.Manchester encoding effectively doubles the data rate compared to non-return-to-zero (NRZ) encoding, as each bit now occupies half the time.
* However, Manchester encoding is less efficient in terms of bandwidth utilization compared to NRZ.

1. Differential Manchester Encoding:

* Differential Manchester encoding is a variation of Manchester encoding where the presence or absence of a transition at the middle of the bit period indicates the bit value, not the direction of the transition.
* A transition from the previous bit state to the same state represents a 1, while a transition from the previous bit state to the opposite state represents a 0.
* This encoding scheme ensures synchronization and eliminates the need for a separate clock signal.
* Differential Manchester encoding is more robust against noise and provides better error detection compared to regular Manchester encoding.
* Like Manchester encoding, it effectively doubles the data rate compared to NRZ.

**Code:**

#include <stdio.h>

#include <inttypes.h>

int main()

{

uint8\_t binaryString[100];

printf("Enter length of sequence: ");

int length;

scanf("%d", &length);

printf("\n");

printf("Enter binary sequence: ");

for (int i = 0; i < length; i++)

{

scanf("%hhu", &binaryString[i]);

}

printf("\n");

printf("NRZ\n");

printf(" 5v\_");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

printf("¦\n |");

for (int i = 0; i < length; i++)

{

if (binaryString[i] == 0)

printf("\_\_\_\_|");

else

printf("\u203E\u203E\u203E\u203E|");

}

printf("\n 0v\u203E");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

printf("¦\n");

printf("\n\n");

printf("Manchester\n");

printf(" 5v\_");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

printf("¦\n |");

for (int i = 0; i < length; i++)

{

if (binaryString[i] == 0)

printf("\_\_|\u203E\u203E");

else

printf("\u203E\u203E|\_\_");

}

printf("\n 0v\u203E");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

printf("¦\n ");

printf("\n\n");

// Differential Manchester Encoding

printf("Differential Manchester\n");

printf(" 5v\_");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

printf("¦\n |");

// Initial state

int lastBit = 1; // Start with a high level

for (int i = 0; i < length; i++)

{

if (binaryString[i] == 0)

{

// For '0', invert at the midpoint

printf("%s", lastBit ? "\_\_|\u203E\u203E" : "\u203E\u203E|\_\_");

}

else

{

// For '1', toggle at the midpoint

lastBit = !lastBit;

printf("%s", lastBit ? "\_\_|\u203E\u203E" : "\u203E\u203E|\_\_");

}

}

printf("\n 0v\u203E");

for (int i = 0; i < length; i++)

{

printf("¦ ");

}

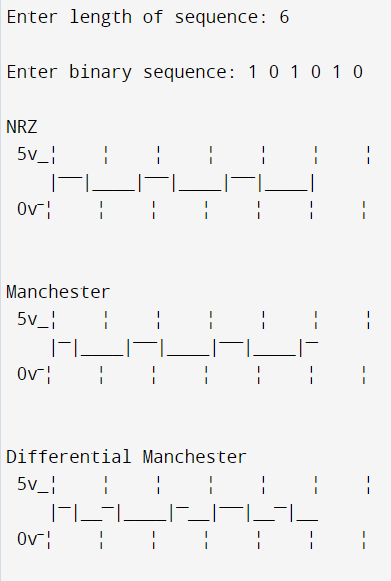
printf("¦\n ");

printf("\n\n");

return 0;

}

Output:



**Conclusion**

NRZ, Manchester and Diffrential Manchester Encoding schemes were studied and implemented successfully.