Simulation of Physical Layer Line Coding Schemes

Ramika Chakhaiyar[1], Devansh Chawla[1]

[1]*Student, Institute of Technology, Nirma University, Ahmedabad-382481, Gujarat, India*

Abstract - *Line coding holds a great significance in the field of telecommunication and is intensively used in High Speed Serial Links (HSSL) mostly on the physical layer before the transmission of these raw data on the link. These codings govern several kinds of sophisticated coding methods for transmitting a digital signal and are exhaustively employed in baseband communication systems. In educational institutions, Line Coding is practiced on electronic trainers. This research paper represents the implementation of different Line Coding schemes like NRZ, RZ, Manchester and Differential Manchester on various softwares including MATLAB and Python. The results have been observed on the software interface in the form of various waveforms as well as compared, highlighting the advantages and disadvantages of each of the schemes. An attempt has been made for a user-friendly software oriented approach to study different kinds of Line Codes by creation of a Graphic User Interface for different line code schemes.*

Key Words: Modulation, Non-Return-to-Zero (NRZ), Return-to-Zero(RZ), Manchester, Differential Manchester, 5G, Internet of Things, Python, MATLAB, Graphic User Interface

1. INTRODUCTION

Serial data transmission (sending data bits consecutively over a single channel) has gained popularity in the current generation, replacing parallel transmission (sending multiple data bits simultaneously over multiple channels) due to the various advantages it possesses, especially in reduction of the power consumption, the decrease in pin count as well as the routing complexity.

However, there are various limitations in the transition from parallel to serial. A few examples of these are that the control symbols have to be sent on the same lane with the data in serial links, and there must be a mechanism defined in order so that the receiver can distinguish between data symbols and control symbols.[1] Also in the case of asynchronous serial links, which implies that there is no clock line sent along with the data line, enough transitions in the data must be present to be certain that the receiver could synchronize as well as recover the clock present on his side. Added to this, they must run at higher speeds, high enough to reach many gigabits per second (Gbps). This results in various impacts such as an increase in EMI (Electro-Magnetic Interferences). [2] In order to mitigate the various constraints, the line encoding presents solutions before the data is transmitted.

In every digital transmission, the data has to be reformatted before modulation. Modulation is a technique of producing a new signal by mixing it with a sinusoid of high frequency. Expressing the digital data in a specified format is known as Line Coding. It is a process of encoding every bit into some signal elements where in every signal element is a discrete and discontinuous voltage pulse. This has a huge application in modulation. It is an important part of any digital communication system. There are many line coding schemes which are introduced for signaling, like 8b10b encoding and scrambling.

In this research paper, implementation of different Line Coding schemes has been taken into account. Schemes such as NRZ, RZ, Manchester and Differential Manchester have been executed on softwares such as MATLAB and Python. The waveforms obtained have then been scrutinized, taking into attention the pros and cons of the different schemes. Also, a Graphic User Interface has been created so as to study the different kinds of Line Codes in the best possible way.

1.1 Necessity of Line Coding

There are numerous advantages for using line coding. Each of the codes which will be discussed in Section 2.1 would have at least one of the following advantages:

i. **Spectrum Shaping and Relocation without modulation or filtering:** This is of high importance in applications such as telephone lines, in which the transfer characteristic has heavy attenuation below 300 Hz. [1]
ii. **Bit Clock recovery:** Due to line coding, bit clock recovery can be easily simplified.
iii. **Elimination of DC components:** This permits AC (capacitor or transformer) coupling between stages (for instance in telephone lines) and can also control baseline wander. Baseline wander leads to shifts in the position of the signal waveform with respect to the detector threshold. It also results in severe erosion of noise margin. [2]
iv. **Error detection:** It is easy to detect an error in the case of line coding.
v. **Bandwidth usage:** Line coding provides the likelihood of transmission at a higher rate compared to other schemes over the same bandwidth. [3]

One can say that the LINE-CODE ENCODER acts as an interface between the TTL level signals of the transmitter and those of the analog channel. Similarly, the LINE-CODE

DECODER acts as a connection between the analog signals of the channel and the TTL level signals required by the digital receiver. [3]

The entire study has been organized as follows. In section I, an introduction to the topic is given. Section 1.1 takes into account the need of line coding and the various reasons why it is preferred. Section 2 consists of the literature review. Section 2.1 gives an account of the different types of line codes like NRZ, RZ, Manchester and Differential Manchester; the advantages and disadvantages of each of these. Section 2.2 tells regarding the various applications in which line coding is useful. In Section 3, the various simulations of these line codes are performed. Section 3.1 discusses the flowchart which was followed for the same. Section 3.2 and Section 3.3 examine the results of the simulation in Python and MATLAB respectively. Section 4 gives a detailed explanation of creation of a Graphic User Interface (GUI) for line codes prepared in Python. A few concluding remarks as a part of this paper are given in Section 5, followed by the references taken into account.

2. LITERATURE REVIEW

2.1 Types of Line Codes

There are various types of line codes which are present, which help reduce our errors or to have other positive effects on our signal. A few important ones are listed.

i. **Non-Return-to-Zero (NRZ):** This is a simple line coding scheme where a positive voltage is defined as bit 1 and a zero voltage is defined as bit 0. Signal does not return to zero at the middle of the bit, thus it is called Non-Return-to-Zero.[4] For example, if the data is **10110**, the signal would be as shown in Fig -1.

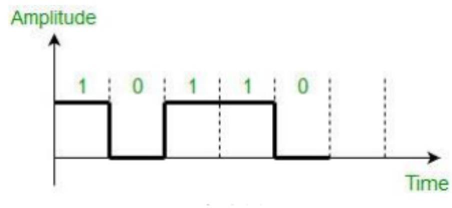

Fig -1: Non-Return-to-Zero (NRZ) Code for Data 10110

A disadvantage of this scheme is the usage of high power to send one bit per unit line resistance. Also, in the case of sets having continuous zeros or ones there will be self-synchronization as well as baseline wandering problems associated. [4]

The NRZ Codes can be further classified as Unipolar NRZ, Polar NRZ, AMI (Alternate Mark Inversion) and CDI (Conditioned Diphase Interface).

A few advantages of NRZ codes as compared to other codes are that there is more bandwidth efficiency. However, their spectral components become 0 Hz, though. This prohibits them from being used on transmission lines that are coupled with transformers or cannot hold DC for any other purpose. [5]

ii. **Return-to-Zero (RZ):** The logic used to represent data in RZ is that half of the signal for bit 1 is represented by +V and half by zero voltage, and half of the signal for bit 0 is represented by -V and half by zero voltage.[6] To understand the same, an example can be seen in Fig -2 having the input data as 01001.

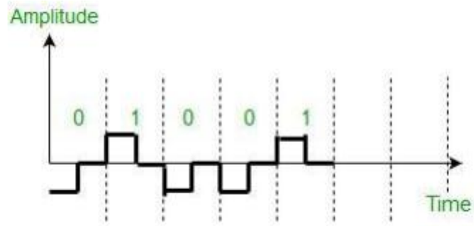

Fig -2: Return-to-Zero (RZ) Code for Data 01001

A limitation of RZ encoding is that it needs high bandwidth. Another concern is the consistency, since three voltage ranges are used. This scheme is not seen today, as a result of all these shortcomings. Instead, the better-performing Manchester and differential Manchester systems have been replaced.

iii. **Manchester:** In Manchester encoding (also known as Phase encoding, or PE), the length of the bit is broken into two parts. During the first half, the voltage stays at one level and transfers in the second half to the other level. Synchronization is given by the transition at the centre of the bit. The logic that is used to describe data using Manchester is that there is a transition from V to +V volts in the centre of the bit for bit 1 and a in the centre of the bit, there is a transition from + V to -V volts for bit 0. For instance, if the input data provided is **010011**, the signal would be in the form given in Fig -3.

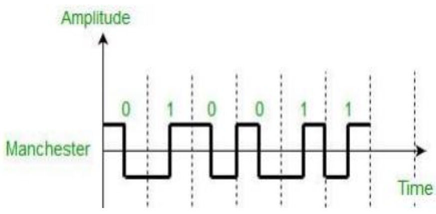

Fig -3: Manchester Code for Data 010011

It is a signal with no DC component that is self-clocking. Consequently, it is simple to galvanically separate electrical contacts using a Manchester code.

The name of the Manchester code comes from its creation at the University of Manchester, where the code was used to store data on the Manchester Mark 1 computer's magnetic drums.

Before the introduction of 6250 bpi tapes, the Manchester code was commonly used for magnetic storage on 1600 bpi electronic tapes. In early Ethernet physical layer standards, the Manchester code was used and is still used in consumer IR protocols, RFID and near-field communication. [6]

iv. **Differential Manchester:** There is often a change at the centre of the bit in Differential Manchester, but the bit values are calculated at the beginning of the bit. In the case where the next bit is 0, the transformation occurs; if the next bit is 1, the transition does not occur.

An example to illustrate the same can be given by having the input data as **010011**, signal would be as shown in Fig -4.

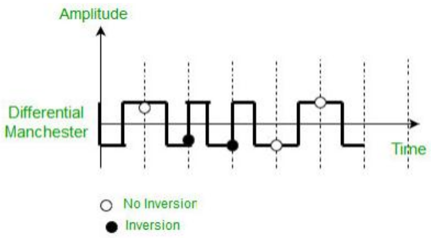

Fig -4: Differential Manchester Code for Data 010011

Similar to Manchester, as each bit has a positive and negative voltage contribution, there is no baseline wandering and no DC part in Differential Manchester. [7]

The drawback is that Manchester's and Differential Manchester's minimum bandwidth are double than that of the NRZ.

In order to compare the different schemes and observe their characteristics, Table 1 summarizes the necessary features of the codes.

Table -1: Comparison of the Line Codes [8]

Sr. No.	Code Name	Timing	Bandwidth	DC Value
1	NRZ	Poor clock recovery	Low	High DC component
2	RZ	Good clock recovery	High	High DC component
3	Manchester	Good clock recovery	High	No DC component
4.	Differential Manchester	Good clock recovery	Moderate	No component

2.2 Applications of Line Codes in various domains

Line Codes are extensively used in various fields, so as to utilize its usage. A few of the many have been listed.

i. **5G:** Polar codes are promising techniques that will be used in 5G systems because they have advantages which outperform turbo codes that were used in previous generations of cellular communications systems. These advantages include higher gain than other codes and there are no error floors like turbo and LDPC codes. Polar codes are proven to be the codes that have the lowest complexity. These codes are considered a breakthrough in coding systems because they are the first provably codes that attain Shannon limit. Moreover, polar codes have the ability to work better in control channels, hence it is dedicated to control channels in 5G-NR systems. [8]

The drawbacks of polar codes include the code length must be a power of two which causes coding rate problems. The complicated design of polar decoders make them expensive; therefore many studies are still searching how to use these codes efficiently.

ii. **Internet of Things:** There is a high importance of line encoding in various platforms including data communication which includes Internet of Things (IoT). The basic principle of line encoding is to encode raw bit streams and apply a few rules before transmitting the same to the transmission medium. Non-return zero encoding (NRZ), return-zero encoding, and Manchester encoding are some examples of encoding methods. Compared with NRZ and RZ encryption, the Manchester coding technique has benefits. Manchester coding will understand the bit interval and encoding the same bit stream for a long time would have no issues. As a clock is used in operation, the common configuration has a drawback, and the output would be divided into two parallel bit streams. In certain transformations, typical architecture even has a spike glitch. In order to solve typical design problems, the latest design of Manchester encoders using FPGA devices is preferred for implementation in the IoT environment. [9]

3. SIMULATION

3.1 Flowchart of Line Codes

In order to understand the various schemes of line coding, implementation of the same was done on various software platforms. The flowchart and methodology followed in each case of scheme is followed so as to get the required waveform. At first, input will be received by the user by entering the data. Then it will check if the entered data is high or low, according to which encoded data will give the resultant waveform.

In NRZ, if the input signal is high, then the output signal will also be high. Similarly, if the input signal is low, the output signal will also be low. [5] So, the encoded signal will be high only if the input signal is high.

In case of RZ, if the input signal is high, the output signal will be high for half of the bit duration after which it will return to zero indicating the absence of pulse for the other half of the bit duration. Similarly, if the input signal is low, the output signal will be low for half of the bit duration, after which it will return to zero indicating absence of the pulse for the other half of the bit duration.

For Manchester Line code, if the input signal is high, the output signal will be high for half of the bit time duration and then the transition will take place from high to low and the output signal will be low for the other half of the bit time duration.[9] While, if the input signal is low, the output signal will be low for half of the bit time duration and then the transition will take place from low to high and the output signal will be high for the other half of the bit time duration. This has been shown in Fig -5.

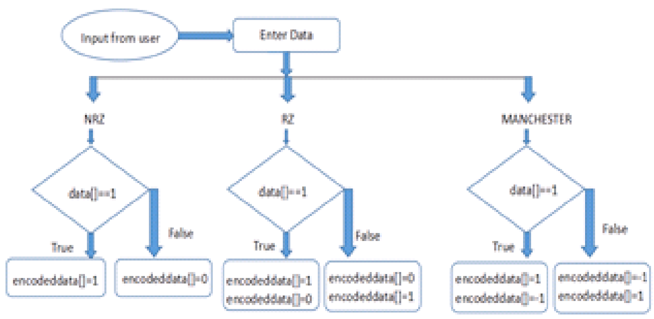

Fig -5: Flowchart of NRZ, RZ and Manchester Line Coding

In the case of Differential Manchester, there will always be transition in the middle of the bit time duration. Now, if the input signal is high, there will be no transition in the beginning, and in case of low signal, transition will take place in the beginning of the bit time duration.

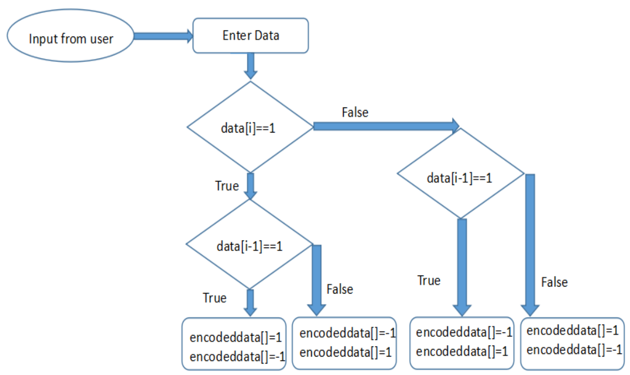

Fig -6: Flowchart of Differential Manchester Coding

For transition in the beginning of the bit time duration, comparison will be made with the previous bit, accordingly the output signal will be generated i.e. transition will be there in beginning of bit time duration only in the case of low input. The same can be seen in Fig 6.

3.2 Simulation on Python Software

In order to carry out the modeling of the proposed flowchart, "Python" software was used. Python is a well known and highly used high-level as well as general-purpose programming language. The basic design philosophy of Python focuses on readability of the code along with its usage of whitespace and object-oriented approach. [10]

The flowchart as described in Section 3.1 was written down in the form of code and the resultant waveforms of NRZ, RZ, Manchester and Differential Manchester are shown in Fig -7 to Fig -10 respectively.

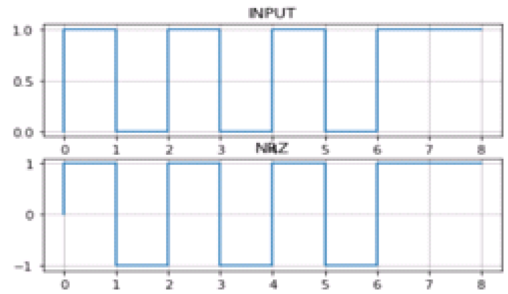

Fig -7: Waveform of Python Simulation for NRZ Scheme

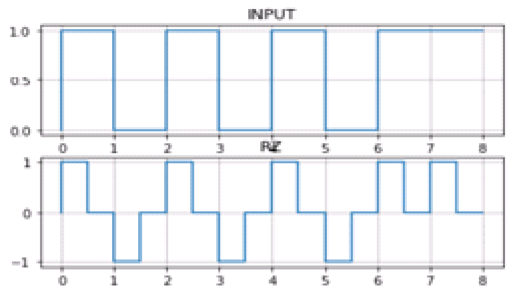

Fig -8: Waveform of Python Simulation for RZ Scheme

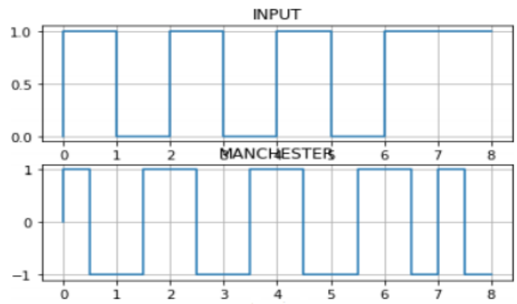

Fig -9: Waveform of Python Simulation for Manchester Scheme

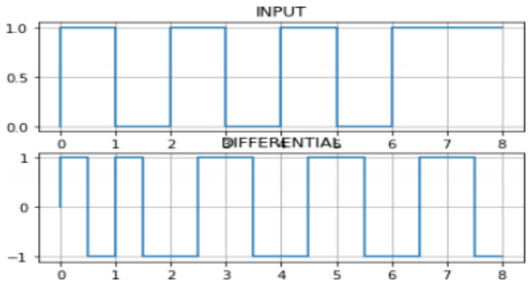

Fig -10: Waveform of Python Simulation for Differential Manchester Scheme

3.3 Simulation on MATLAB Software

As seen in section 3.2 where the simulation was carried out in Python Software, the same can also be applied to various other softwares. One of the most commonly used Software is MATLAB and hence the implementation of the schemes in the same has been carried out. MATLAB provides the best solution in order to combine a desktop environment which has been modified for iterative analysis as well as design processes along with any programming language which has been used in order to implement matrix and array mathematics. Various things are a part of it including, the Live Editor for the creation of scripts that are used to combine code, output, and formatted text. [10]

In order to implement the same, the signal which was obtained by each of the schemes can be seen in Fig -11.

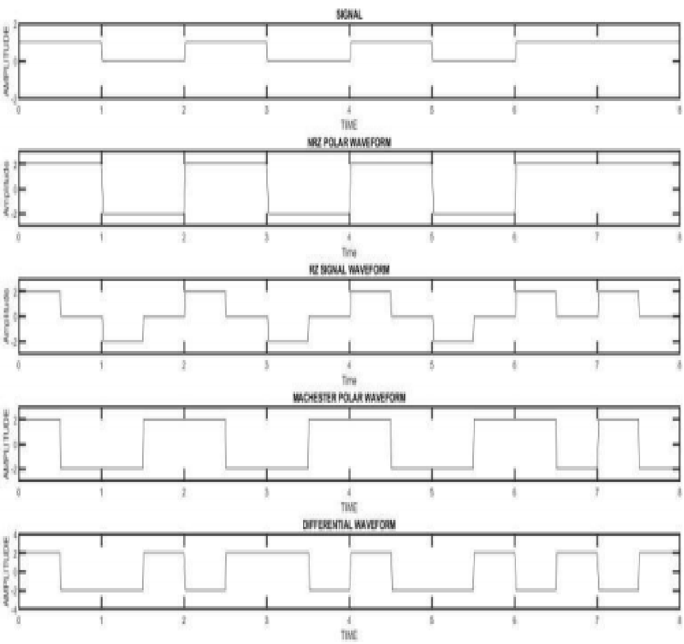

Fig -11: Waveform of MATLAB Simulation for various Schemes

4. GRAPHIC USER INTERFACE (GUI)

A graphical user interface (GUI) can be said to be a program interface due to which the program can be easily understood. There are numerous visual indicator (graphics) indicators which display the much needed information, like icons, menus etc. in order to display information and related user controls. This is in contrast to text-based interfaces, in which the data and commands are in the form of text. In order to carry out the manipulations, pointing devices, like a mouse, trackball, or a finger on a touch screen are used.

Few of the basic components on a graphical user interfaces are: [6]

i. **Pointer:** Pointer can be defined as a small symbol which appears on the screen. It is due to the pointer that one can easily navigate to select objects and commands.

ii. **Pointing Device:** A pointing device, enables one for the selection of the objects on the display screen. These can be a mouse, trackball etc.

iii. **Icons:** Icons represent the small pictures which demonstrate the commands, files, or windows. In order to execute a specific command or for opening a window, one can move the pointer to the icons and press the mouse button.

iv. **Desktop:** The area which is present on the display screen where one can easily find the icons is known as the desktop.

v. **Windows:** A person can divide the screen into different areas. These areas are referred to as windows. In each of these windows, different programs can be developed or a different file can be displayed. One can change the shape and size of the window, by moving it around the display screen.

vi. **Menus:** In most of the graphical user interfaces, one can run and execute the needed commands by selecting a choice from a menu.

To make the presentation of the Simulation easy for the common man, GUI was implemented using the Python Code. The result which was obtained by the same for the various schemes has been shown in Fig -12-15. This makes it simpler for visualization as well as comparison done on the various schemes, hence resulting in more accessibility.

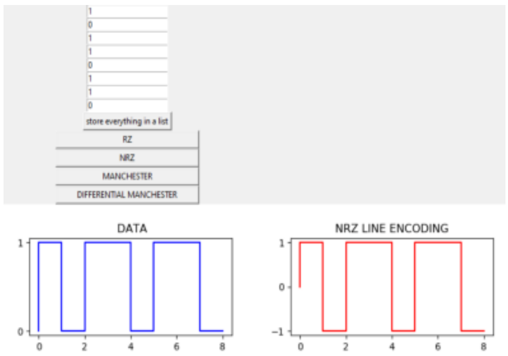

Fig -12: GUI for NRZ Coding Scheme for Data 10110110

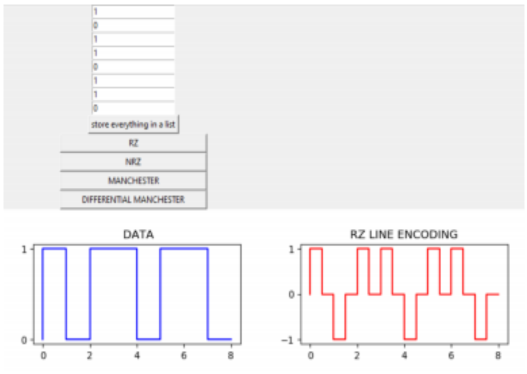

Fig -13: GUI for RZ Coding Scheme for Data 10110110

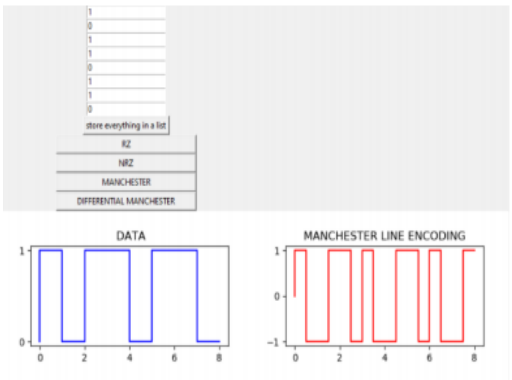

Fig -14: GUI for Manchester Scheme for Data 10110110

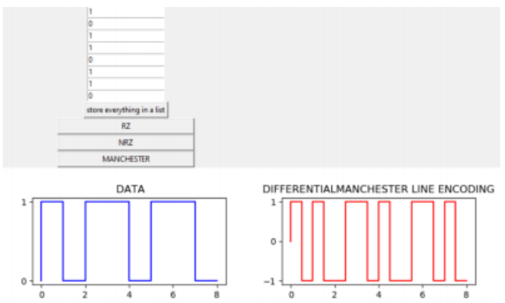

Fig -15: GUI for Differential Manchester Scheme for Data 10110110

The GUI implemented above makes it easy in order to compare the various schemes; hence making it practical as well as convenient to perform modulation. This is beneficial for digital communication systems.

5. CONCLUSION

Modulation can be said to be a technique of producing a new signal which is carried out by mixing it with a sinusoid of high frequency. The way in which the digital data is expressed in a particular format is called Line Coding. There are various Line Coding schemes that are commonly employed in baseband modulation systems. Characteristics and the usage of each of these differ. These different Line Coding schemes namely NRZ, RZ, Manchester and Differential Manchester, can be implemented on various platforms so as to see which one suits the best for a particular application. Simulation on MATLAB and Python is easy to be carried out, taking into account when the input(s) raw signal changes from 1 to 0 (and vice-versa) based on the parameters of the specific scheme. Creation of a Graphic User Interface (GUI) makes it convenient for any end user to easily understand and perform the simulation of the schemes in a systematic manner. The major applications of line codes include 5G and IOT.

REFERENCES

[1] J. Saadé, A. Goulahsen, A. Picco, J. Huloux and F. Pétrot, "A scalable low overhead line coding for asynchronous High Speed Serial transmission," 2014 IEEE 18th Workshop on Signal and Power Integrity (SPI), Ghent, 2014, pp. 1-4, doi: 10.1109/SaPIW.2014.6844527.

[2] Chen Wang and J. L. Drewniak, "Quantifying the effects on EMI and SI of source imbalances in differential signaling," 2003 IEEE Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.03CH37446), Boston, MA, USA, 2003, pp. 865-868 vol.2, doi: 10.1109/ISEMC.2003.1236722.

[3] Dae Kim and Jae-Kyoon Kim, "A Condition for Stable Minimum-Bandwidth Line Codes," in IEEE Transactions on Communications, vol. 33, no. 2, pp. 152-157, February 1985, doi: 10.1109/TCOM.1985.1096260.

[4] V. Munteanu and D. Tarniceriu, "Information analysis of the NRZ-M line codes and their generalization," International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005., Iasi, Romania, 2005, pp. 231-234 Vol. 1, doi: 10.1109/ISSCS.2005.1509896.

[5] Swati Verma and Rohit Singh, "Multilevel NRZ Coding for Transmission of Digital Signals", International Journal of Advanced Research in Science and Engineering, Vol. No.2, Issue No.9, September 2013.

[6] S. Hykin, 2001, Communication systems, 4th ed, John Wiley & Sons, Inc.

[7] Z. Song and Y. Takasaki, "Line Coding for Clock Recovery With Minimal Filtering," 2007 IEEE Pacific

Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, 2007, pp. 430-433, doi: 10.1109/PACRIM.2007.4313266.

[8] Juma Abu-Ghalune, Mahmoud Alja'fari, Parallel Data Transmission Using New Line Code Methodology, International Journal of Networks and Communications, Vol. 6 No. 5, 2016, pp. 98-101. doi: 10.5923/j.ijnc.20160605.02.

[9] Arianto, Adi Putra, S. A. Sudiro, Abdul Hakim and Fivtatianti Hendajani. “Manchester encoder component for data processing IoT environment.” (2019) 2nd International Conference on Data and Information Science doi:10.1088/1742-6596/1192/1/012043 9.

[10] S. Wilson, 1996, digital modulation and coding, prentice hall.inc.