**Computer Networks Home Assignment**

Shrawani Sunil Shinde

Roll No 46

GR No 11911031

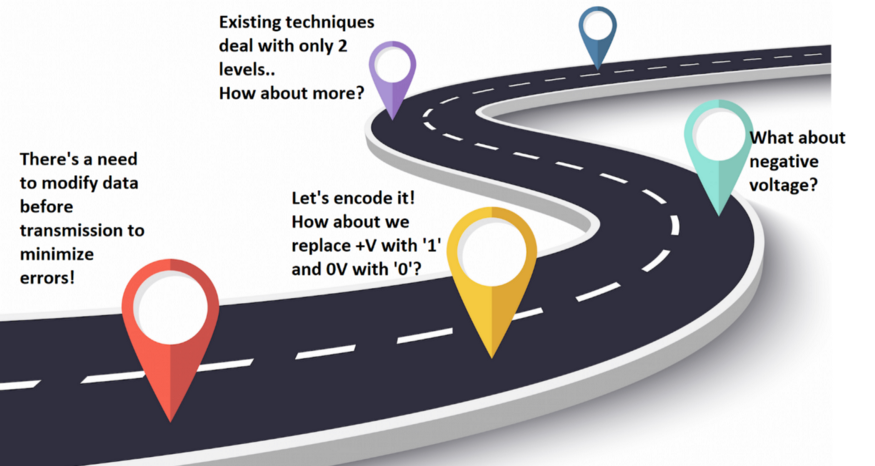
Instrumentation and Control Division C

**Journey of Line Encoding Methods**

Data is everywhere. The world is getting digitalized, we live in a generation where data is transmitted within milliseconds. Talking about Digitalization, data is stored and transmitted in digital form where every institution, organization are invested into and one small error in the transmission might prove a great loss!

Well, that’s where Line Encoding Techniques come into the picture! Get ready to brace yourselves as this Blog is going to take you on a journey of how Line Encoding Methods started and have been used in the field of Telecommunication!

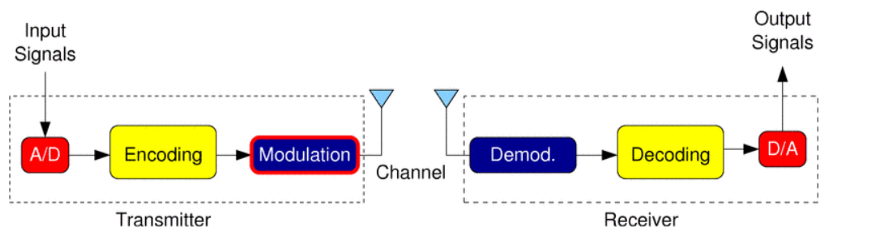
**“Everything on this earth is in a continuous state of evolving, refining, improving, adapting, enhancing, and changing. It is a continuous process of new things coming up, new techniques adopted to support its growth!”**



**What is Line Encoding?**

**In**[**telecommunication**](https://www.wikiwand.com/en/Telecommunication)**, a line code is a pattern of voltage, current, or photons used to *represent digital data***[***transmitted***](https://www.wikiwand.com/en/Transmission_(telecommunications))***down a***[***transmission line***](https://www.wikiwand.com/en/Transmission_line)***.***

*Line coding* (also called *digital baseband modulation* or *digital baseband transmission*) is a process carried out by a transmitter that converts data, in the form of binary digits, into a baseband digital signal that will represent the data on a transmission line. The transmission line in question could be a link between two devices in a computer network, or it could form part of a much larger telecommunications network. The receiver is responsible for converting the incoming line-coded signal back into binary data.



**Why Line Encoding?**

There are many reasons for using line coding. Each of the line codes you will be **examining offers one or more of the following advantages:**

· **Spectrum Shaping and Relocation**without modulation or filtering. This is important in telephone line applications, for example, where the transfer characteristic has heavy attenuation below 300 Hz.

·**Bit clock recovery**can be simplified.

· **DC component can be eliminated;** this allows AC (capacitor or transformer) coupling between stages (as in telephone lines). Can control baseline wander (baseline wander shifts the position of the signal waveform relative to the detector threshold and leads to severe erosion of noise margin).

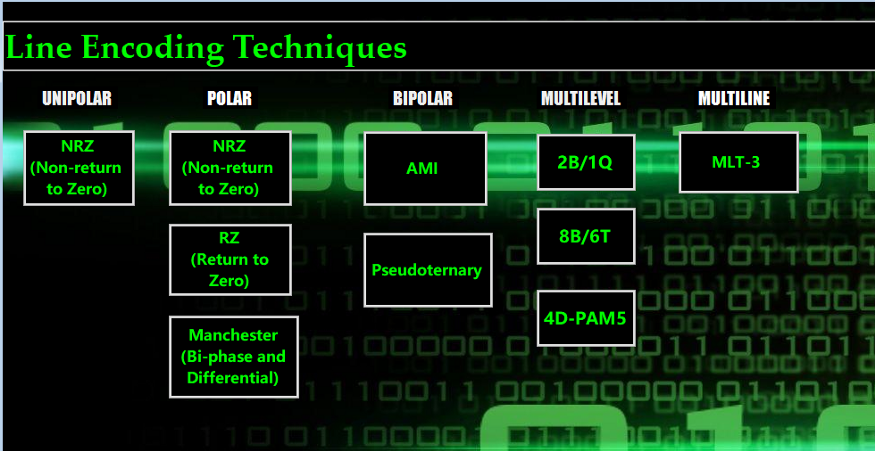
· **Error detection capabilities.**

· **Bandwidth usage**; the possibility of transmitting at a higher rate than other schemes over the same bandwidth.

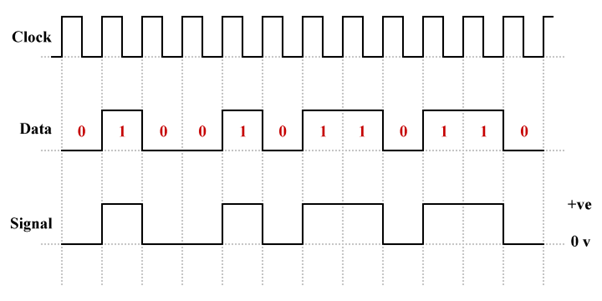
**Line Encoding techniques**

There are many different line coding techniques, ranging in complexity **from very basic unipolar schemes** in which the presence or absence of a voltage is used to represent a binary one or a binary zero, **to highly sophisticated multilevel schemes in which different signal amplitudes are used, each representing a unique grouping of binary digits.**

The **simplest line coding** schemes are generally only used for relatively **low-speed asynchronous transmission**, in which data is sent in small independent blocks, and the receiver can re-synchronize itself with the transmitter at the start of each incoming block. **For high-speed synchronous applications** in which much larger blocks of information are transmitted, **timing becomes far more critical, as do factors such as noise and the potential presence in a signal of a net DC component.**



**Let Us Start with how the First Encoding Technique (Unipolar NRZ) started!**



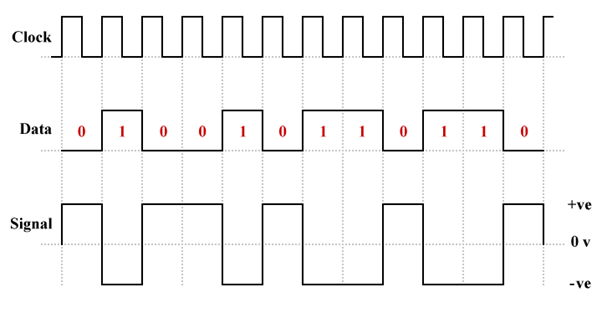
The First Encoding Technique was introduced in order to have unipolar signaling where **all non-zero signaling elements have the same polarity**— either they are all positive or they are all negative. **It was an attempt analogous to a simple on-off keying scheme in which the presence of a voltage pulse signifies a binary one and the absence of a pulse signifies a binary zero.**The term “Non-return-to-zero” signifies that the signal does not return to zero in the middle of the bit time. A positive voltage represents a binary one, and a zero voltage represents a binary zero.

**This Encoding Algorithm proved helpful as it was easy to implement and required low bandwidth whereas it also had a few drawbacks as it didn’t satisfy some of the motives such as error correction.**

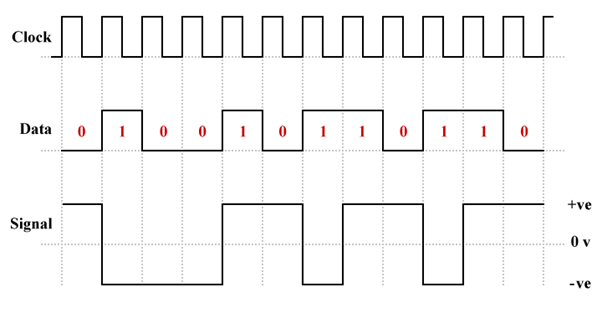
**Polar Encoding (NRZ)**

With the introduction of Unipolar Encoding, a technique was introduced for Polar Encoding, which uses both poles to represent a given data.

There are two types of NRZ Encoding Techniques. ***NRZ-I(Inversion)***and***NRZ-L(Level)***



Example of NRZ-Level Encoding



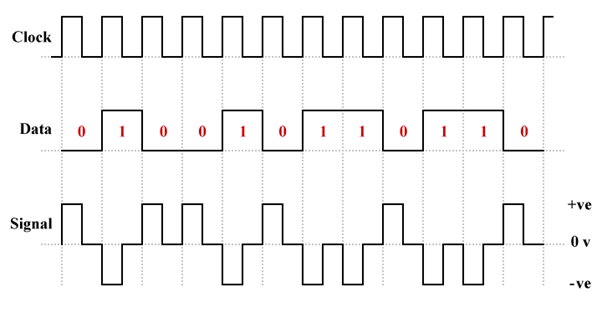
Example of NRZ-Inversion Encoding

The first one we will look at is called***NRZ-Level* (NRZ-L).**Here, the voltage level determines the value of a bit. **Typically, a *logic low* (binary zero) is represented by a positive voltage while a *logic high* (binary one) is represented by a negative voltage.**

The second polar NRZ line coding scheme we will look at is called ***NRZ-Invert* (NRZ-I).**Here, the value of a bit is determined by the presence or absence of a transition from a positive voltage to a negative voltage, or vice versa.**A transition signals that the next bit is a logic high (binary one), while no transition signals a logic low (binary zero).**

**Although the technique proved helpful in certain ways, neither polar-NRZ-L nor polar NRZ-I is DC balanced. The problem of Baseline wandering was still unsolved when long sequences of zeros occur.**

**Polar RZ( Return-to-Zero)**



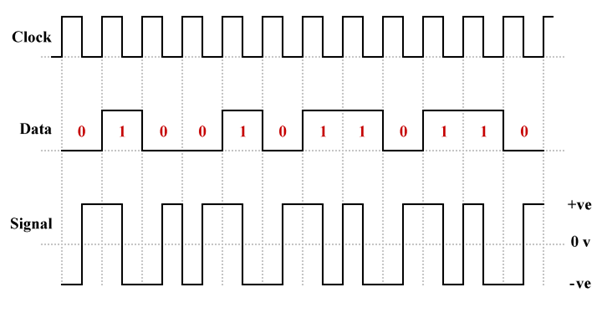
Example of Polar RZ Encoding

Some of the problems relating to polar NRZ line coding schemes are mitigated here through the use of three signaling levels. It is still the case that (typically) logic low is represented by a positive voltage and logic high is represented by a positive voltage, but in both cases, the signal level returns to zero halfway through the bit time and stays there until the next bit is transmitted (note that, somewhat confusingly, some sources refer to this line coding scheme as bipolar RZ or BPRZ).

**Polar RZ proved beneficial as it uses less power than polar NRZ-L or polar NRZ-I and the transitions in the middle of each bit time enabled the receiver to re-synchronize itself. On the other hand, the technique is not DC balanced, although the DC component will be negligible if the number of ones and zeros is approximately the same. It requires twice as much bandwidth as either polar NRZ-L or polar NRZ-I.**

**Manchester Encoding(Bi-phase and Differential)**

Manchester encoding is a widely used line coding scheme that embeds timing information in the transmitted signal. It does this by ensuring that there is a transition (high-to-low or low-to-high) in the middle of every bit time, making it easy for the receiver to retrieve a clock signal from the incoming bitstream and maintain synchronization with the transmitted signal.

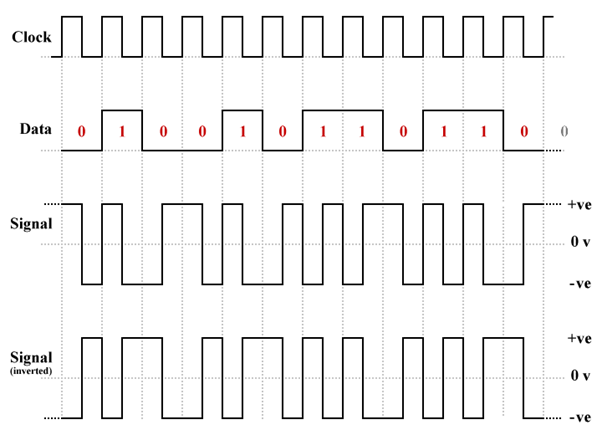


Example of Bi-Phase Manchester Encoding

**In Bi-phase encoding, logic high (binary one) is represented by a positive pulse with a period of half a bit time followed by a negative pulse of the same duration.** Similarly, A logic low (binary zero) consists of a negative pulse followed by a positive pulse, each with a period of half a bit of time.

**Manchester encoding addresses the shortcomings of polar NRZ-L and polar RZ schemes by ensuring that there is sufficient embedded timing information in the transmitted signal to enable the receiver to easily maintain synchronization (there is at least one transition per bit time), and by eliminating the possibility of a DC component developing and Baseline wandering. The main disadvantage of Manchester encoding, and the main reason it is not used for Fast Ethernet (100 MBps) and above, is because of its inefficient use of bandwidth.**

**Manchester encoding suffers from the problem that, if the signal gets *inverted* (i.e. changes polarity) whilst in transit, it is transformed from one convention to the other, leading to *polar ambiguity***(the receiver will interpret ones as zeros, and vice versa). The problem can be overcome by using a variant of Manchester encoding called *differential Manchester encoding*. **Whereas Manchester encoding could be described as combining aspects of polar RZ and NRZ-L, differential Manchester is more like a combination of RZ and NRZ-I.**

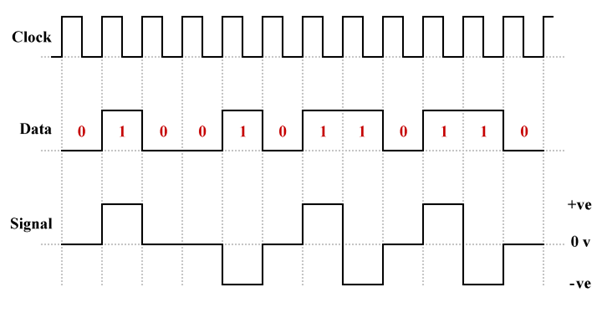


Example of Differential Manchester Encoding

**In Differential Encoding, there is still a transition in the middle of each bit time, but there is only a transition at the *start* of a bit time if the bit will be a logic high (binary one).** There is *no transition* at the start of the bit time if the bit will be a logic low (binary zero). It is the *presence or absence* of a transition at the beginning of a bit time that signifies what logic state the bit represents.**Neither the direction of the transition nor the actual voltage level on the line during the bit time have any significance in this respect.**

**Unfortunately, differential Manchester suffers from the same inefficiency as Manchester in terms of bandwidth usage, but it *has* seen commercial use (it was used, for example, in IEEE 802.5 token ring local area networks). An important feature of differential Manchester is that, like polar NRZ-I, it is a *differential* line coding scheme, relying solely on the presence or absence of transitions at the start of each bit time to signal the logic state of each bit.**

**Bipolar Encoding -AMI (Alternate Mark Inversion)**

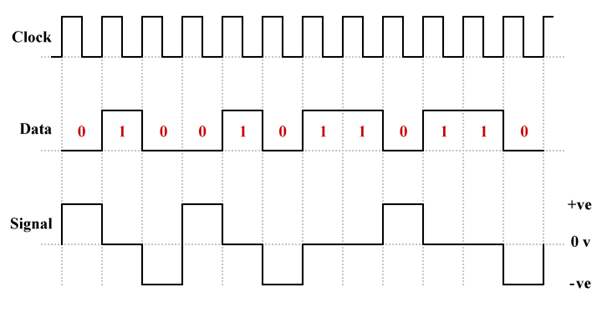


Example of AMI Encoding

**Like polar RZ, bipolar line coding schemes (sometimes called a *multi-level binary* or *duo-binary*) use three voltage levels — positive, negative, and zero.**That, however, is pretty much where the similarity ends. **Bipolar *alternate mark inversion* (AMI) uses alternate positive and negative voltages** to represent *logic high* (binary one), and a zero voltage to represent *logic low* (binary zero). Although AMI is technically an NRZ line coding scheme itself, it was developed as an alternative to other NRZ schemes **in which long runs of ones or zeros introduced a DC component into the signal.**

**The technique proved helpful in solving several issues as it uses less power than the polar NRZ line coding scheme. The signal has no DC component. It also avoided issues like Baseline wandering and polar ambiguity. On the other hand, · long sequences of zeros can cause loss of synchronization at the receiver due to the absence of voltage transitions.**

**Bipolar Encoding -Pseudoternary**

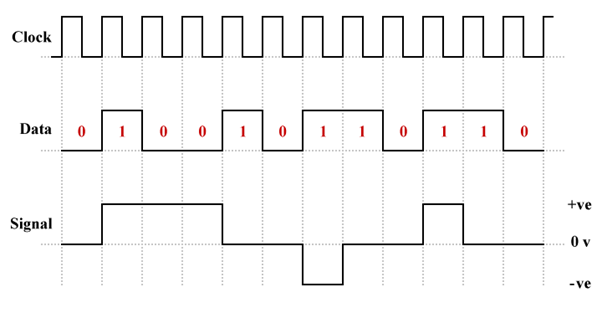


Example of Pseudoternary Encoding

The pseudoternary version of bipolar line coding is essentially **identical to AMI except that logic high is represented by a zero voltage and logic low is represented by alternate positive and negative voltages**— the exact opposite of what happens with AMI.

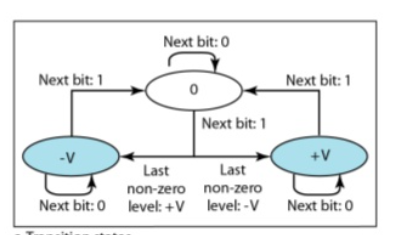
**The advantages and disadvantages of pseudoternary line coding are the same as for AMI with the single exception that, rather than long sequences of zeros, it is long sequences of ones that can cause the receiver to lose synchronization.**

**Multiline Encoding- Multi-level-Transmit(MLT-3)**



Example of MLT-3 Encoding

MLT-3 (multi-level transmit) is, like polar NRZ-I and differential Manchester, a *differential* line coding scheme. The “3” in the name reflects the fact that, whereas both of the aforementioned schemes use only two signal levels (positive and negative) to represent binary values, **MLT-3 uses *three* levels (positive, negative and zero).**



Rule for MLT-3 Encoding

Because MLT-3 requires four transitions **to complete one complete cycle, the maximum fundamental frequency that must be supported by the transmission medium is only one quarter of the baud rate,** making MLT-3 ***suitable for transmission over twisted-pair copper wire cables.*Consequently, MLT-3 has been used successfully for 100BASE-TX Ethernet, the predominant form of Fast Ethernet (100 Mbps), operating over two wire pairs in category 5 (or above) twisted pair cables.**

**As we may have surmised from the MLT-3 signaling diagrams above, MLT-3 is not a DC-balanced line coding scheme. It is also susceptible to loss of synchronization at the receiver if there are long runs of zeros. One solution to this problem is to use block coding(multi-level).**

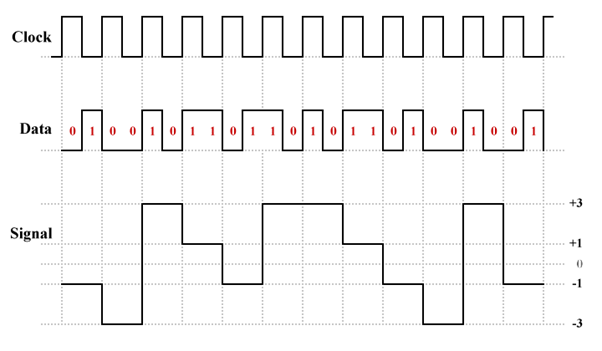
**Multilevel/Block Coding (mBnL)**

In this technique, he data to be transmitted is **broken down into fixed-length blocks.** Each block is replaced, prior to transmission, with a slightly longer block selected from an ***encoding table***.**The replacement block contains a unique bit pattern designed to ensure that the transmitted signal will contain a minimum number of transitions**, ensuring that the resulting line code is DC-balanced and that the receiver can remain synchronized.

***In mBnL schemes, a pattern of m data elements is encoded as a pattern of n signal elements in which 2m ≤ Ln.***

**Multilevel Encoding — 2B/1Q**

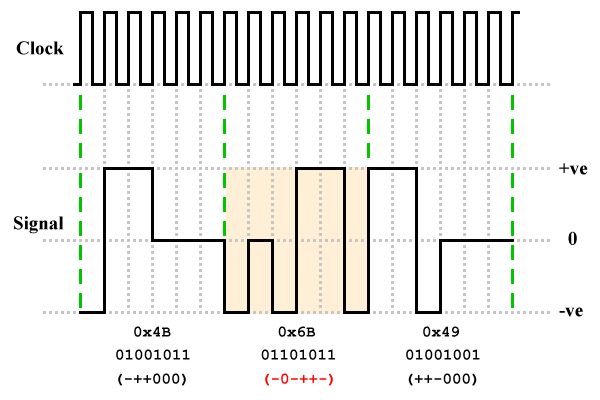
**The *2B1Q* (two binary, one quaternary)** line coding scheme is the first of three very different *multi-level* schemes we will be looking at. **The idea behind multi-level schemes generally is to encode more than one data bit per signal symbol in order to maximize bandwidth efficiency.** The *2B1Q* (two binary, one quaternary) line coding scheme is the **first of three very different *multi-level* schemes we will be looking at.** The idea behind multi-level schemes generally is to encode more than one data bit per signal symbol in order to maximize bandwidth efficiency.



Example of 2B/1Q Encoding

2B1Q is one of the line coding schemes used to **transmit data over an *Integrated Services Digital Network* (ISDN) *Basic Rate Interface* (BRI).**That’s the part of an ISDN network that connects an *exchange terminator* (ET) in a local telephone exchange to a *network terminator* (NT1 or NT2) at the customer’s premises for ISDN’s basic rate service. **This part of the network is sometimes called the *U loop*, or *U interface*.**

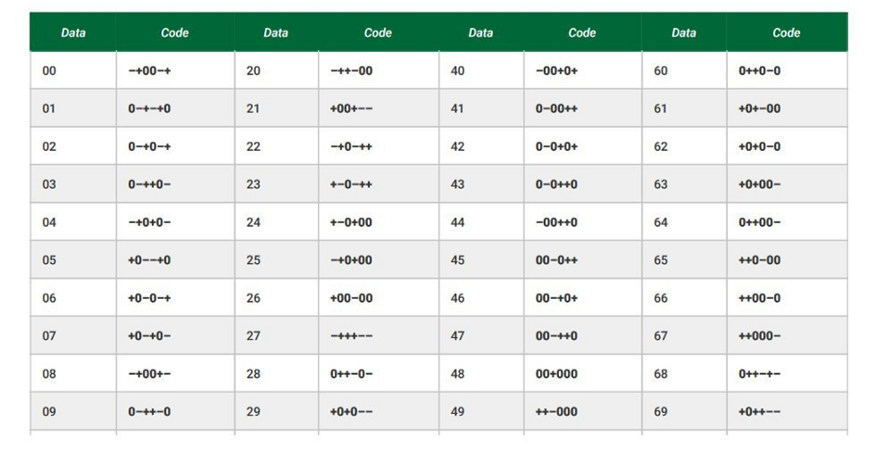
**Multilevel Encoding — 8B/6T**



Example of 8B/6T Encoding

The next multi-level line coding scheme we want to look at is ***8B6T* (eight binary, six ternary),** which can also be classed as a multi-level scheme despite the fact that it uses only three signaling levels (*positive*, *negative* and *zero*). 8B6T is a rather unusual coding scheme that **uses *three-level pulse amplitude modulation* (PAM-3) to code a block of *eight* binary digits onto a sequence of *six* signaling elements.** Each signal element takes one of three values — positive, negative, or zero.

**A sequence of eight binary digits has only 256 possible permutations (2 8 = 256), whereas there are 729 unique ways in which in which six ternary signal levels can be combined (3 6 = 729). This leaves 473 redundant signaling combinations that can be used to provide synchronisation and error detection capabilities, as well as ensuring that the signal output is balanced with respect to DC.**

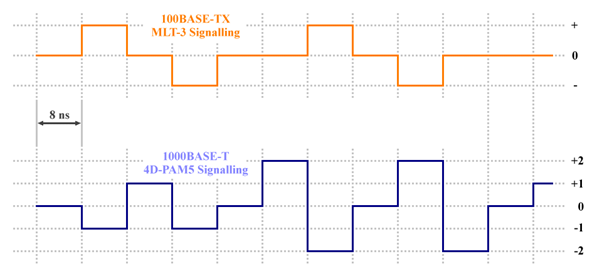


Encoding Table for 8B/6T

**Multilevel Encoding — 4D-PAM5**

The last multi-level line coding scheme we are going to look at is ***4D-PAM5* (four-dimensional five-level pulse amplitude modulation)**. As the rather long-winded name suggests, 4D-PAM5 uses ***five* signalling levels (we’ll refer to them as -2, -1, 0, 1 and 2).** The 4D part refers to the fact that data is transmitted using four wire pairs simultaneously.

1998 saw the emergence of *Gigabit Ethernet*, which has a gross bit rate of 1 Gbps (1000 Mbps) — **a tenfold increase in performance over Fast Ethernet**. As we saw earlier, the**MLT-3 line coding scheme used for 100BASE-TX is not DC-balanced**, and does not guarantee the presence of sufficient transitions to enable the receiver to maintain synchronization. The data is therefore encoded before transmission using the 4B5B block coding scheme, which maps four data bits onto a 5-bit code. **That explains** **why the actual gross data rate is only 100 Mbps, despite the fact that the 125 MHz analogue bandwidth allows a signaling rate of 125 megabaud.**



4D-PAM5 line coding has the same baud rate as MLT-3 but uses 5 signalling levels

The 4D-PAM5 line coding scheme uses five different signalling levels and transmits on all four wire pairs simultaneously, achieving a baud rate of 125 megabauds per second on each channel, and an overall baud rate of 500 megabauds per second. **Every symbol transmitted corresponds to two bits of data, so the gross bit rate is 1,000 Mbps — 8 bits during each 8- nanosecond cycle.**

**Summary**

So as we have discussed the generic as well as a few advanced encoding techniques, let’s have a look at the **summary:**

1. During digital data **Baseband Transmission, faces many challenges which include errors while transmission**and many more, and that’s where Line Encoding and Decoding comes into the picture
2. Various Encoding Techniques ranging from**Unipolar, Polar, Bi-polar to Manchester, Multilevel and Multiline Encoding Techniques**have been discussed where we can **witness the journey of how things started from a small element and have emerged into something really useful nowadays that we can send a signal at over 1000 Mbps and even more speed with more advanced techniques**
3. **Technology keeps evolving but the process of evolution, the journey is what teaches us to keep a mindset of curiosity and watch how many wonders it can create!**

***References:***

<https://www.technologyuk.net/telecommunications/telecom-principles/line-coding-techniques.shtml>

<https://www.tutorialspoint.com/digital_communication/digital_communication_line_codes.htm>l

<https://www.wikiwand.com/en/Line_code>

<https://www.fiberoptics4sale.com/blogs/archive-posts/95046214-line-coding-in-digital-communication>