Analog and Digital Communication

Part - 5

Data Encoding and Communication Technique

We know that data (information) may be of analog or digital type and the signal that is transmitted through the channel, whether guided or unguided, are also of Analog and Digital type. Now data needs to be encoded into signal, before they are actually transmitted through the channel. Based on this data and signal types, there are four encoding techniques.

1. Analog Data to Analog Signal
2. Analog Data to Digital Signal
3. Digital Data to Analog Signal
4. Digital Data to Digital Signal

To send a signal over a physical medium, we need to encode or transform the signal in some way so that the transmission medium can transmit it. The sender and receiver must agree on what kind of transmission has been done so that the signal may be received properly and the information may be recovered without error.

The information content in a signal is based upon having some changes in it, i.e. having some variation in the signal. Thus the signal needs to be modified in some way to carry the information we want to convey. This is called modulation. Thus the act of changing or encoding the information in the signal is known as **modulation**.

3. Digital to Analog Modulation

This type of encoding technique is useful when we transmit digital data from computers over a telephone line. This is different from transmitting voice over telephone, which was analog to analog modulation.

This method appears to be simple because we only have to give two distinct values (0 and 1) to the analog signal, but more than one successive 0s or 1s may lead to confusion, as it may be treated as a single 0 or 1.

In case of analog to analog encoding, the carrier signal that we want to transmit is modulated by the analog information, where there is potentially infinite number of values that could have been transmitted, but in this case, the carrier signal is modulated by the digital information, to produce the composite signal that is to be transmitted. We have only two values (0 or 1) that modulate the signal.

Like in analog to analog encoding, there are three properties of the carrier wave that we can alter to convey information. These are the amplitude, the frequency and the phase. In addition, we can also use a combination of amplitude and phase change to encode information more efficiently.

Modulation is also known as Shift Keying in this type of encoding technique. Thus we have four different type of digital to analog modulation.

1. Amplitude Shift Keying (ASK) – where the amplitude of the carrier wave is modified.
2. Frequency Shift Keying (FSK) – Where the frequency of the carrier wave is modified.
3. Phase Shift Keying (PSK) – Where the phase of the carrier wave is modified.
4. Quadrature Amplitude Modulation (QAM) – The phase as well as Amplitude of the carrier wave is modified.

Amplitude Shift Keying (ASK)

In this kind of encoding, a high voltage is used to represent 1 and a negative high voltage is used to represent 0, although the reverse convention may also be used. Sometimes to reduce the energy needed for transmission, we can choose to represent a 0 or a 1 by a zero voltage and the other bit by a high voltage. This scheme is called **on-off-keying (OOK)**.

Like Amplitude Modulation, ASK is also vulnerable to noise. As amplitude changes easily as a result of the noise in the line and if the noise is great enough, it may result an error at the receiving end.

It can be shown that the bandwidth required for ASK encoding is equal to the baud rate of the signal. So if we want to transfer data at 256 kbps, we need a bandwidth of at least 256 KHz.

Frequency Shift Keying (FSK)

Just as in the case of analog to analog modulation, we can encode a digital signal by changing the frequency of the analog carrier. This is Frequency Shift Keying (FSK). When a bit is encoded, the frequency of the carrier wave changes and it remains constant for a particular duration. Just as in FM, noise in the line has little effect on the frequency and so this method is less susceptible to noise.

FSK is not much used in practice because of the need for higher bandwidth and the comparative complex requirement for changing the frequency.

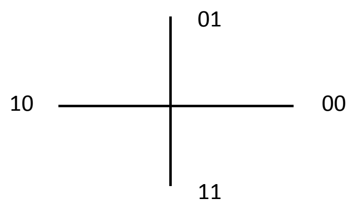
Phase Shift Keying (PSK)

We can encode by varying the phase of the carrier signal and this is called Phase Shift Keying (PSK). To send a 1, we could use a phase of 0 while we could change it to 180 degrees to represent a 0. Such an arrangement is not affected by the noise in the line.

Instead of having only 2 phases, we could have four phases 0°, 90°, 180° and 270°. Each of these phase shifts could represent 2 bits at one go, say the combination of 00, 01, 10 and 11 respectively. Such a scheme is called 4-PSK. The concept can be extended to higher levels and we could have 8-PSK to send a group of 3 bits in one go.

Di-bit	Phase
00	0°
01	90°
10	180°
11	270°

Di-bit (2 bits)



Constellation Diagram

4 PSK

Clearly, this scheme is more efficient than ASK because, we can now achieve a higher bit rate from the same bandwidth.

Quadrature Amplitude Modulation (QAM)

To further improve the efficiency of transmission, we can combine ASK and PSK together. This technique is called Quadrature Amplitude Modulation (QAM). It can be shown that QAM required the same bandwidth as ASK or PSK. So we are able to achieve a much higher bit rate using the same baud rate. That is why QAM is the method of encoding used currently in data communication applications.

4. Digital to Digital Encoding

The last combination of encoding possible is Digital to Digital Encoding. A simple example is when sending data from a computer to a printer or transmission is made between computer to computer over Local Area Networks such as Ethernet.

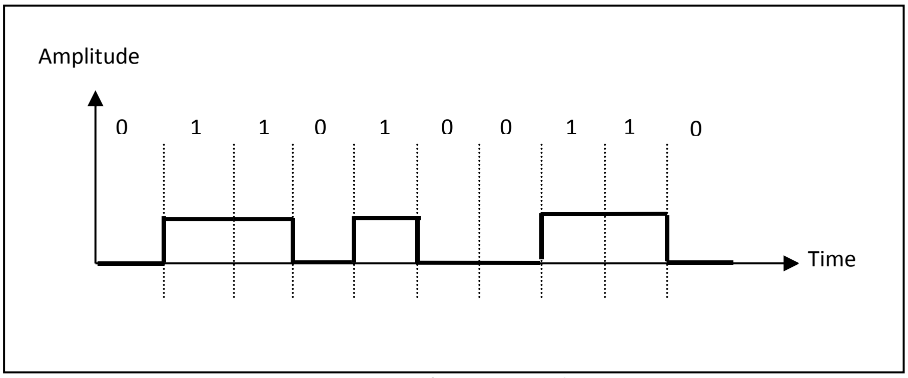
This kind of encoding is of three types.

1. Unipolar Encoding
2. Polar Encoding and
3. Bi-Polar Encoding

Unipolar Encoding

This is a simple mechanism, but is rarely used in practice because of the inherent problems of the method. As the name implies, there is only one polarity used in the transmission of the pulses. A positive pulse may be taken as a 1 and a zero voltage can be taken as 0, or the other way round. Unipolar encoding uses less energy and also has low cost.

The two problems are those of synchronization and the direct current (DC) component in the signal.



Unipolar Encoding

Polar Encoding

Unlike Unipolar schemes, the polar methods use both a positive as well as a negative voltage level to represent the bits. So a positive voltage may represent a 1 and a negative voltage may represent a 0, or the other way round. Because both positive and negative voltages are present, the average voltage is much lower than in the unipolar case, mitigating the problem of the DC component in the signal. The zero voltage will indicate no transmission in the channel.

There are three popular polar encoding schemes:

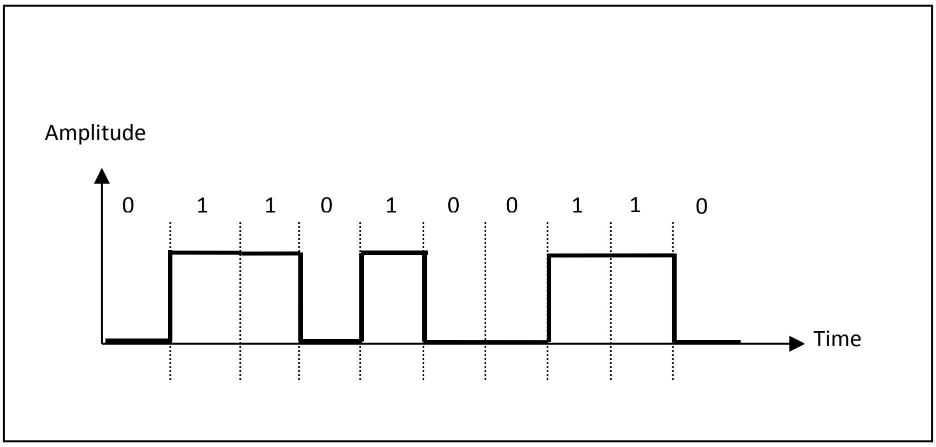
1. Non – Return to Zero (NRZ) Encoding – Which has two more variants:
    - NRZ-L (NRZ – Level) encoding
    - NRZ-I (NRZ – Inversion) encoding
2. Return to Zero (RZ) Encoding
3. Biphase Encoding – Which has two more variants
    - Manchester Encoding
    - Differential Manchester Encoding

Non-Return to Zero (NRZ-L Encoding and NRZ-I Encoding)

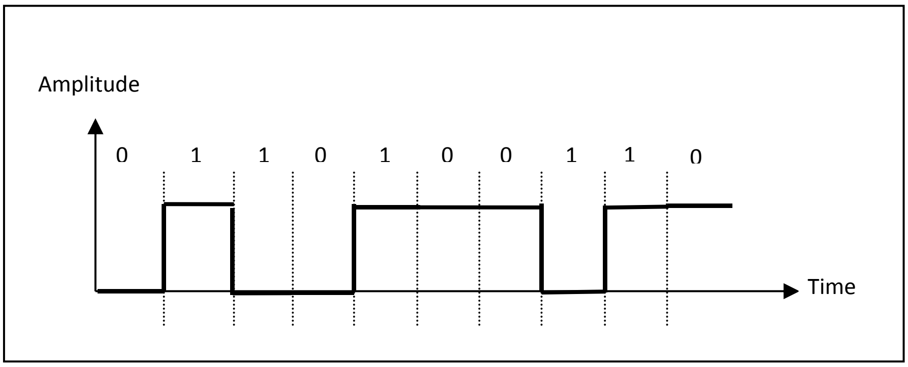
In NRZ-L (for Level) encoding, it is the levels themselves that indicate the bits. For Example, a positive voltage may be used for 1 and a negative voltage may be used for a 0, or the other way round.

On the other hand, in NRZ-I (Inversion) encoding, an inversion of the current level indicates a 1, while a continuation of the level indicates a 0.

Regarding synchronization, NRZ-L is not well placed to handle it if, there is a string of 0's and 1's in the transmission. But NRZ-I is better off here as a 1 is signaled by the inversion of the voltage. So every time a 1 occurs, the voltage is inverted and we know the exact start of the bit, allowing the clock to synchronize with the transmitter. However, this advantage does not extend to the case where there is a string of 0s.



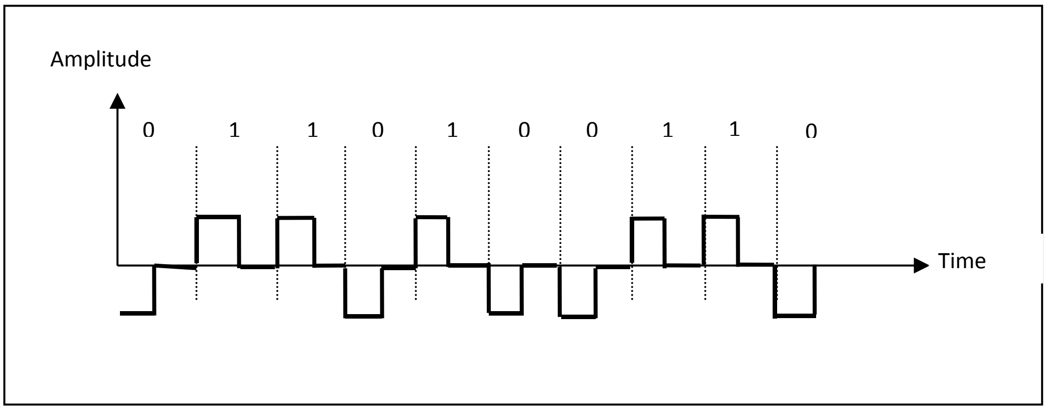
NRZ-L Encoding – The two levels indicates the bits



NRZ-I Encoding- The inversion of the current level shows 1

Return to Zero (RZ) Encoding

The Return to Zero encoding method allows for synchronization after each bit. This is achieved by going to the zero level midway through each bit. So a 1 bit could be represented by the <u>positive to zero transition</u> and a 0 bit by a <u>negative to zero transition</u>. At each transition to zero, the clocks of the transmitter and receiver can be synchronized. The price one has to pay for this is the higher band-width requirement.



RZ Encoding

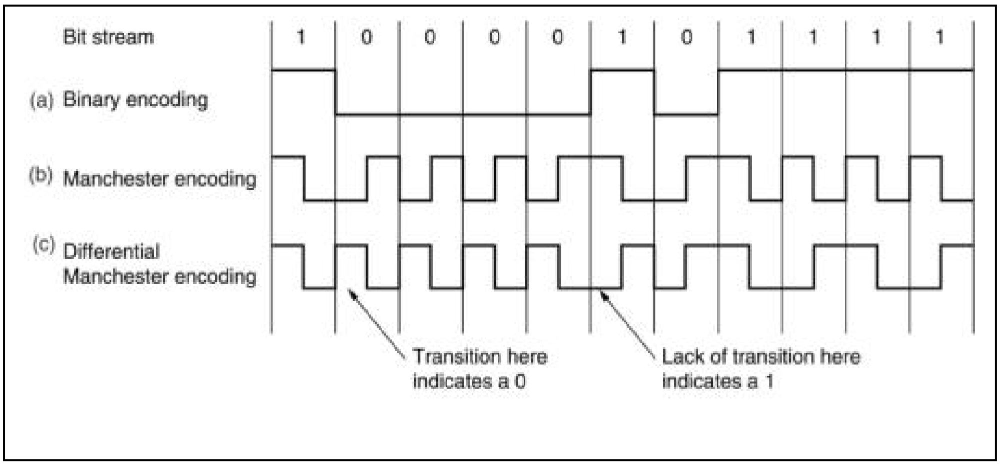
Biphase Encoding (Manchester Encoding and Differential Manchester Encoding)

To overcome the limitations of NRZ Encoding (presence of the dc component and the lack of synchronization capability) and RZ Encoding (Higher band-width requirement), Biphase coding techniques can be adopted. There are two different variations of this technique, namely Manchester Encoding and Differential Manchester Encoding technique.

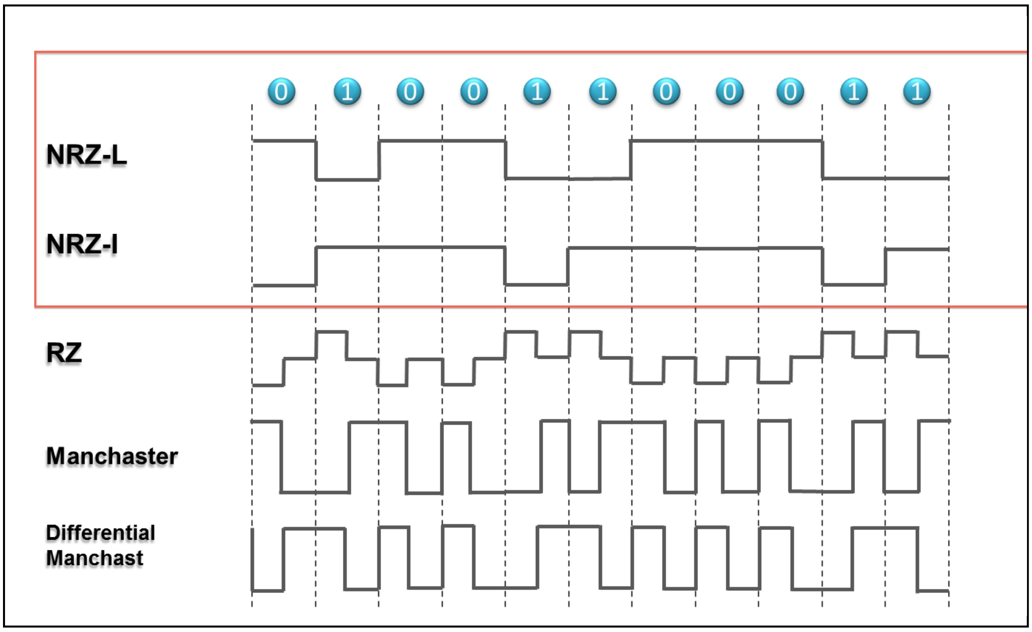
In the Manchester Encoding there is a transition at the middle of each bit. A binary 1 corresponding to high-to-low transition and a binary 0 corresponds to a low-to-high transition in the middle or the other way round.

In Differential Manchester Encoding, the encoding of a 0 is represented by the presence of a transition both at the beginning and at the middle and 1 is represented by a transition only in the middle of the bit period.

Manchester codes are now very popular and have been specified for the IEEE 802.3 standard for baseband coaxial cables and twisted pair CSMA/CD bus LANs.



Manchester encoding and Differential Manchester encoding techniques



A Comparison between the different polar encoding techniques

Multiplexing

If we want to share data between a single source and a single destination, things would be comparative easy. But in the real world, things are not so straightforward. We are having a large number of nodes, each of which may be transmitting or receiving from or possibly to different nodes at the same point of time.

This problem is solved by performing **multiplexing**, which is done by sharing the available frequency band or by dividing up the time between the different transmissions. The former is called **Frequency Division Multiplexing (FDM)** while the other scheme is **Time Division Multiplexing (TDM)**.

A Multiplexer is a device that can accept n different inputs and send out 1 single output. This output can be transmitted over a link or medium to its destination, where the original inputs have to be recovered. This is done by a Demultiplexer. We need to realize that for this kind of scheme to work, the creation of the composite signal by the multiplexer needs to be such that the original component signals can be separated at the receiving end.

Frequency Division Multiplexing (FDM)

Suppose that the human voice has to be transmitted over a telephone. This has frequencies that are mostly within the range of 300 Hz to 3400 Hz. We can modulate this on a carrier channel, such as one at 300 KHz. Another transmission may be made in a different carrier channel of frequency 304 KHz, yet another may use 308 KHz. Thus we are dividing up the channel from 300 KHz up to 312 KHz into different frequencies for sending data. (Note that each band of 4 KHz can carry 4000 different frequency components, which is more than the human voice frequency range). This is Frequency Division Multiplexing (FDM) because all the different transmissions are happening at the same time and it is only the frequencies that are divided up.

The composite signal to be transmitted over the medium of our choice is obtained by summing up the different signals to be multiplexed. At the destination, the original components are to be separated by demultiplexing. In practice, a scheme like this could result in **interference or cross talk** between adjacent channels, because the band-pass filters that are used to contain the data (300 Hz to 3400 Hz) are not sharp. To minimize this, there are **guard bands**, or unused portion of the spectrum between each two channels.

Another possible cause of interference could arise because of the fact that the equipments, such as amplifiers used to increase the strength of the signal, may not behave linearly over the entire set of frequencies that we want to transmit. Then the output may contain frequencies that are the sum or difference of the frequencies used by the input. This produces what is called **Inter-modulation Noise**.

It may be noted that in this case the actual modulation technique is not of consequence. So one can use Amplitude Modulation or Frequency Modulation to generate the composite signal.

FDM has the disadvantage of not being entirely efficient if the transmissions that are multiplexed together have periods of silence or no data. Since a frequency band is dedicated to each data source, any such periods are simply not utilized.

Time Division Multiplexing (TDM)

Another multiplexing scheme is to use the entire bandwidth for each channel, which is divided into different time slots. So for a voice transmission, transmitting four different signals, a cycle of 100ms could be sufficient, where each transmission would be allotted a slice of 25ms. On the receiving end, the transmission has to be reconstructed by dividing up the cycle into the different slices, taking into account the transmission delays.

This synchronization is essential, for if the transmission delay is, say 40ms, then the first transmission would start reaching 40ms later, and would extend to 65ms. If we interpret it to be from 0 to 25ms, there would be complete loss of the original transmission.

There are two types of Time Division Multiplexing; Synchronous TDM and Statistical TDM.

Synchronous Time Division Multiplexing

In Synchronous TDM, the time slots are reserved for each transmission, irrespective of whether it has any data to transmit or not. Therefore this method can be inefficient because many time slots may have only silence. But this method is easier to implement because the act of multiplexing and demultiplexing is easier.

Here we usually transmit digital signals. We can also support data sources with differing data rates. This could be done by assigning fewer slots to slower sources and more slots to the faster ones. Here the composite cycle of time slots is called a **Frame**. At the receiving end, the frame is decomposed into the constituent data streams by sending the data in each time slot to the appropriate buffer. The sequence of time slots allotted to a single data source makes up a transmission channel.