BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 3

DCA2104

BASICS OF DATA COMMUNICATION

Unit 4

Digital Transmission

Table of Contents

SL No		Topic	Fig No / Table / Graph	SAQ / Activity	Page No
1	Introduction		-	-	3
	1.1	Objectives	-	-	
2	Digital to digital conversion		1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	1	4 - 21
	2.1	Line Encoding	-	-	
	2.2	Types of Line Coding	-	-	
	2.3	Unipolar Scheme	-	-	
	2.4	Polar Schemes	-	-	
	2.5	Bi-phase: Manchester and differential Manchester	-	-	
	2.6	Bipolar schemes	-	-	
	2.7	Scrambling	-	-	
3	Analog to Digital Conversion		16, 17, 18, 19, 20, 21, 22	2	22 - 28
	3.1	Pulse code modulation (PCM)	-	-	
	3.2	Delta modulation (DM)	-	-	
4	Transmission Modes		23, 24, 25, 26, 27	3	29 - 33
5	Summary		-	-	34
6	Terminal Questions		-	-	
7	Answers		-	-	35 - 36

1. INTRODUCTION

This unit introduces the concept of digital transmission. By using a computer network, we can transmit information from one point to another. This information should be converted to either a digital signal or analog signal for transmission.

In this unit, we will be discussing different types of signal/ data conversion. First, we discuss digital to digital conversion. In the next section we will discuss line encoding. Then we will discuss different types of line encoding. We will also discuss analog to digital conversion. In the last section, we will discuss various transmission modes.

1.1 Objectives

After studying this unit, you should be able to:

- *Describe digital to digital conversion*
- *Explain line encoding*
- *Describe types of line coding*
- *Explain analog to digital conversion*
- *Describe transmission modes*

2. DIGITAL TO DIGITAL CONVERSION

We studied data and signals in unit 3. We have discussed that data can be either analog or digital and signals that represent data can also be digital or analog. In this section, we see how to represent digital data by using digital signals. This conversion involves three techniques. They are line coding, block coding and scrambling. In this, line coding is always needed; block coding and scrambling may or may not be needed.

2.1 Line Coding

Line coding is the process of converting digital data to digital signals. We know that data in any form such as text, images, audio, video are stored in computer memory as sequences of bits. Line coding converts a sequence of bits to a digital signal. Digital data are encoded into a digital signal at sender side and at the receiver, the digital data is recreated by decoding the digital signal. Figure 4.1 shows the process of line coding.

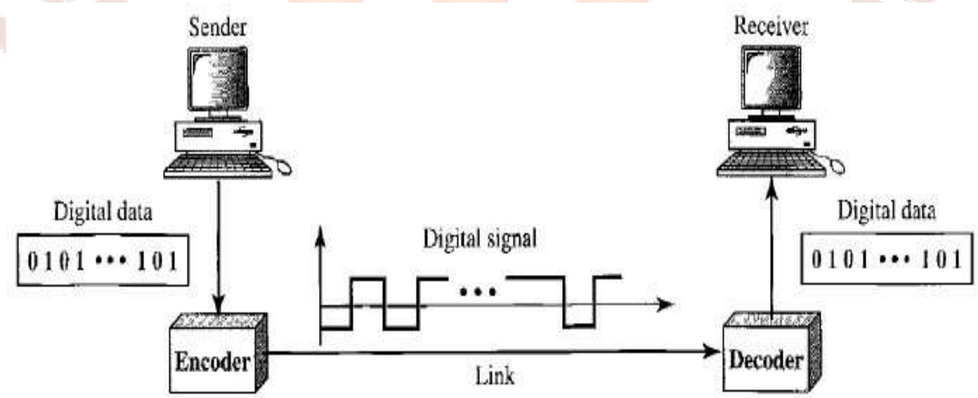


Figure 4.1: Line coding and decoding

Now we will discuss different characteristics of line coding.

Signal Element versus Data Element:

We can say, a data element is the smallest entity that can represent a piece of information. This is known as the bit. Signal element is the shortest unit of a digital signal. The signal element carries data elements in digital communication. That is, signal elements are the carriers and data elements are being carried. The number of data element carried by each signal element is represented by a ratio ***r***. Figure 4.2 shows several situations with different values of ***r***.

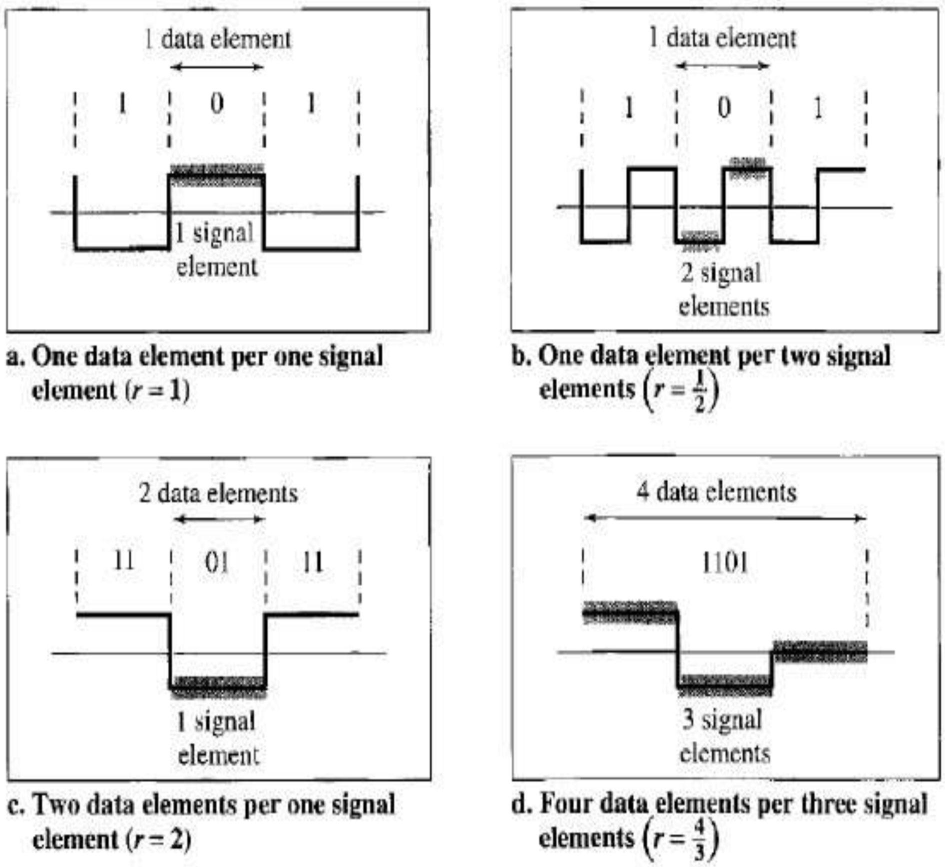


Figure 4.2: Signal element versus data element

Suppose each data element is a person who needs to be carried from one place to another, then signal element is a vehicle that can carry people. In figure 4.2 (a), one data element is carried by one signal element (r = 1). In

4.2 (b), two signal elements are used to carry each data element (r = ½). In

4.2 (c), a signal element carries two data elements (r = 2). In 4.2 (d), a group of 4 bits is being carried by a group of three signal elements (r = 4/3).

Data rate versus Signal rate

The data rate defines the number of data elements sent in 1 second. The unit used to measure is *bits* per second (bps). The signal rate is the number of signal number of elements sent in one second. The unit is called as *baud*. The data rate is also called the bit rate; the signal rate is also called the pulse rate, the modulation rate or the baud rate. In data communication, the goal is to increase data rate while decreasing the signal rate. Increasing data rate increases the speed of transmission, decreasing signal rate decreases the bandwidth requirement. The relationship between signal rate and data rate depends on the value and data pattern. In order to derive a formula for relationship, we have to define three cases, they are: the worst, best and average. Worst case is when we need maximum signal rate, the best case is when we need the minimum. In data communication, we usually prefer the average case. We can formulate the relationship between data rate and signal rate as

$$S = c \times N \times \frac{1}{r} \qquad \textbf{baud}$$

Where N is the data rate (bps); c is the case factor which varies for each case; S is the number of signal elements and r is the previously defined ratio.

Bandwidth

In unit 3, we discussed that a digital signal that carries information is nonperiodic. Theoretically, bandwidth of a nonperiodic signal is continuous with an infinite range. But in real life, most digital signals have a bandwidth with finite values. That means, although the actual bandwidth of a digital signal is infinite but the effective bandwidth is finite. Bandwidth (range of frequencies) is proportional to the signal rate (baud rate). The minimum bandwidth can be given as

$$B_{\min} = c \times N \times \frac{1}{r}$$

If the bandwidth of the channel is given, we can solve the maximum data rate using the equation,

$$N_{max} = \frac{1}{c} \times B \times r$$

Baseline Wandering

The receiver calculates a running average of the received signal power while decoding a digital signal. This average is called the baseline. The incoming signal power is evaluated against this baseline to determine the value of the data element. A long string of 0s or 1s can cause a float in the baseline (known as baseline wandering) and make it difficult for the receiver to decode correctly. A good line coding scheme is required to prevent baseline wandering.

DC Components

Signals creates very low frequencies when the voltage level in a digital signal is constant for a while. These frequencies around zero are called DC (*direct-current*) *components*, which creates problems for a system that cannot pass low frequencies or system that uses electrical coupling. For instance, a telephone line cannot pass frequencies below 200 Hz, for which we will need a scheme with no DC component.

Self-synchronization

To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals. If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals. Figure 4.3 shows the example of lack of synchronization. Here, the sender sends 10110001 while receiver receives 110111000011.

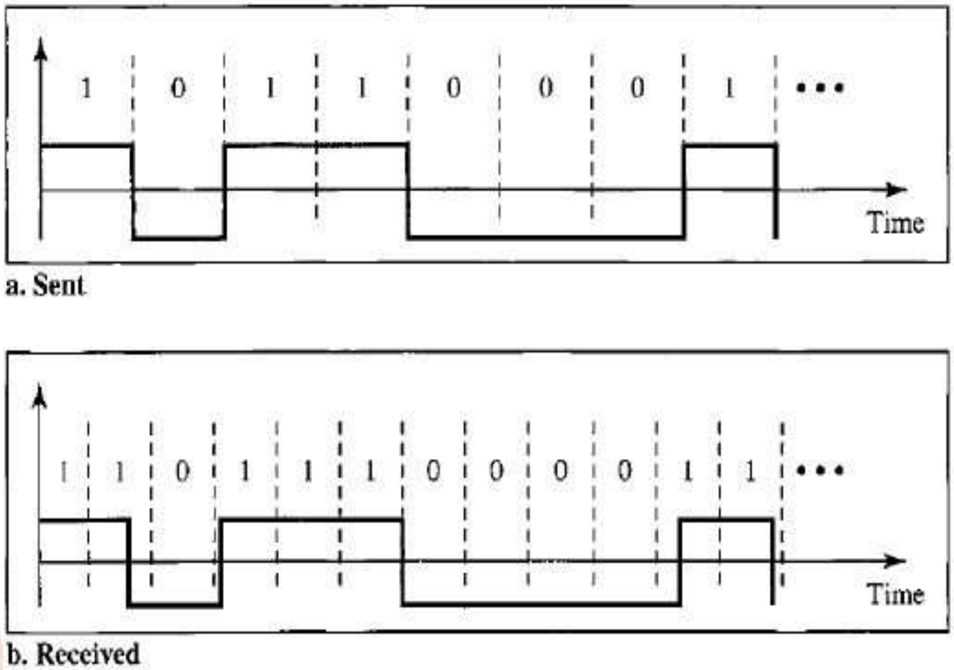


Figure 4.3: Effect of lack of synchronization

A self-synchronizing digital signal includes timing information in the data being transmitted. This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle or end of the pulse. If receiver's clock is out of synchronization, these points can reset the clock.

Built-in Error Detection

To find out the errors that occurred during transmission, it is desirable to have a built-in error-detecting capability in the generated code.

Immunity to Noise and Interference

Another desirable code characteristic is a code that is immune to noise and other interferences.

Complexity

A complex scheme is more costly to implement than a simple one. For instance, a scheme that uses four signal levels is more difficult to interpret than one that use only two levels.

2.2 Types Of Line Coding

Line coding schemes can be divided into five broad categories, they are: unipolar, polar, bipolar, multilevel, multitransition. Figure 4.4 shows line coding schemes.

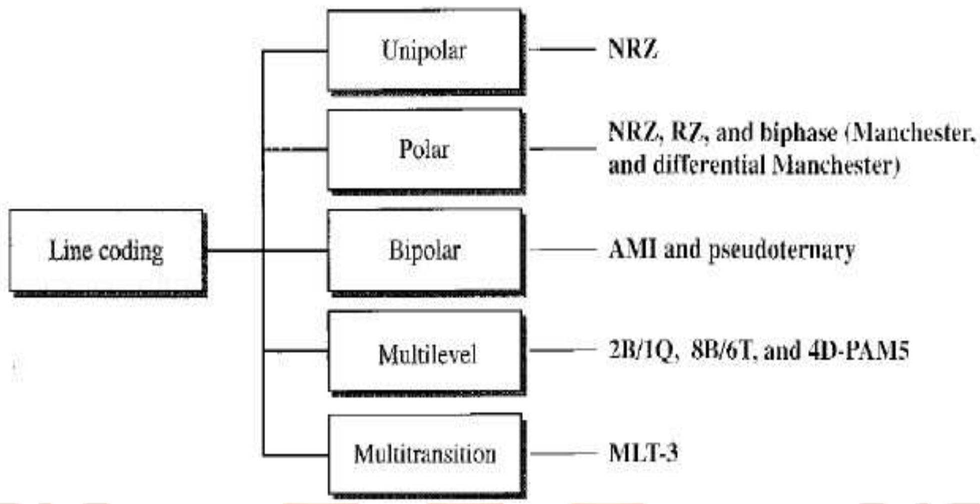


Figure 4.4: Line coding schemes

2.3 Unipolar Scheme

In this scheme, all the signal levels are on one side of the time axis that is either above or below the axis.

Non-Return-to-Zero (NRZ): A unipolar scheme was designed as a non- return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. It is called NRZ because the signal does not return to zero at the middle of the bit. Figure 4.5 shows a unipolar NRZ scheme.

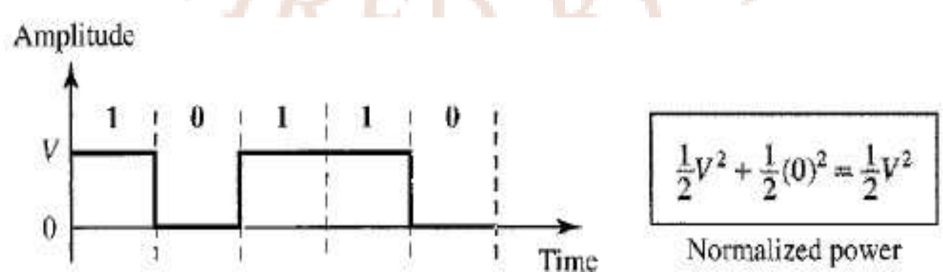


Figure 4.5: Unipolar NRZ scheme

Unipolar scheme is very costly; hence, this scheme is normally not used in data communications today.

2.4 Polar Schemes

In polar schemes, the voltages are on the both sides of the time axis. For example, the voltage level for 0 can be positive and the voltage level for 1 can be negative.

Non-Return-to-Zero (NRZ)

In polar NRZ encoding, we use two levels of voltage amplitude. There are two versions of polar NRZ, they are NRZ-Level (NRZ-L) and NRZ-Invert (NRZ-I). Figure 4.6 shows polar NRZ-L and NRZ-I schemes.

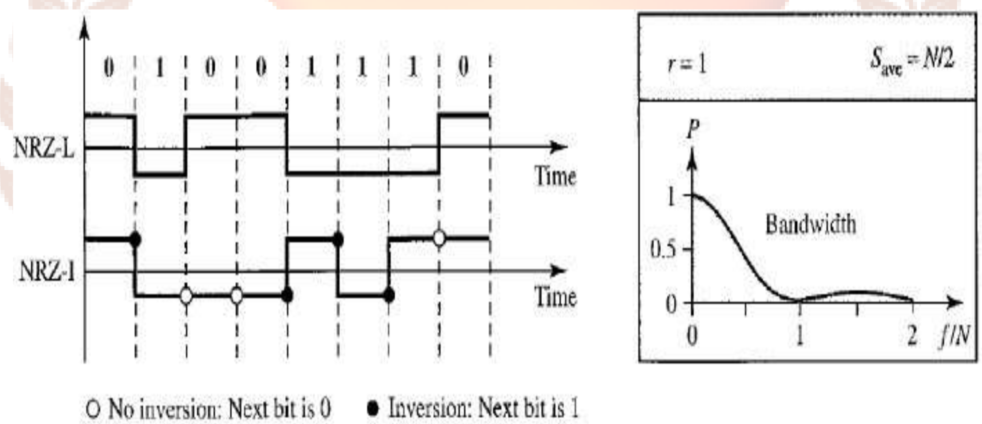


Figure 4.6: polar NRZ-L and NRZ-I schemes

The figure shows the value of *r*, the average baud rate and the bandwidth. In *NRZ-Level (NRZ-L),* the level of the voltage determines the value of the bit. In *NRZ-Invert (NRZ-I),* the change or lack of change in the level of the voltage determines the value of the bit. If there is no change, the bit is 0 and if there is a change, the bit is 1.

Let us now compare NRZ-L and NRZ-I, although baseline wandering is a problem for both variations, it is twice as severe in NRZ-L. If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes reoriented. The receiver might have difficulty recognizing the bit value. In NRZ-I, this issue occurs only for a long sequence of 0s. So, if we eliminate long sequence of 0s, we can avoid baseline wandering.

The synchronization problem (that is, sender and receiver clocks are not synchronized) also exists in both schemes. A long sequence of 0s can cause problem in both schemes, a long sequence of 1s affects only NRZ-L. The problem is more serious in NRZ-L than in NRZ-I.

Another issue with NRZ-L is when there is a quick change of polarity in the system. For instance, if twisted pair cable is the medium, a change in the polarity of the wire results in all 0s interpreted as 1s and all 1s interpreted as 0s. NRZ-I doesn't have this problem. Both schemes have an average signal rate of N/2 Bd.

Figure 4.6 also shows the normalized bandwidth for both variations. The vertical axis shows the power density (the power for each 1Hz of bandwidth) and the horizontal axis shows frequency. From the figure, the bandwidth reveals a very serious problem for both NRZ-L and NRZ-I. The value of the power density is very high around frequencies close to zero. This means that there are DC components that carry a high level of energy. Most of the energy is concentrated in frequencies between 0 and N/2. This means that although the average of the signal rate is N/2, the energy is not distributed equally between the two halves.

Return to Zero (RZ)

Drawback of NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the return-to-zero (RZ) scheme. This scheme uses three values, they are positive, negative and zero. In RZ, the signal changes not between bits but during the bit.

Figure 4.7 shows polar RZ scheme. In that the signal goes to 0 in the middle of each bit and remains there until the beginning of the next bit.

The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth. Another drawback is that, a sudden change of polarity resulting in all 0s interpreted as 1s and all 1s interpreted as 0s. But in case of polar RZ scheme, there is no DC component problem.

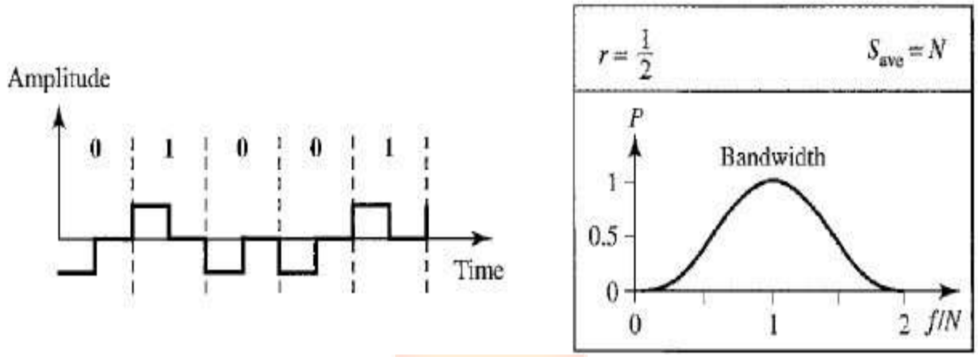


Figure 4.7: Polar RZ Scheme

Another problem is the complexity. RZ uses three levels of voltage, which is more complex to create and recognize. Due to all these disadvantages, this scheme is not used today. Instead, it has been replaced by the better- performing Manchester and differential Manchester schemes.

2.5 Biphase: Manchester And Differential Manchester

The idea of RZ and the idea of NRZ-L are combined into the Manchester scheme. In *Manchester encoding*, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization. *Differential Manchester*, on the other hand, combines the ideas of RZ and NRZ-I. There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition and if the next bit is 1, there is none.

The Manchester scheme overcomes several problems associated with NRZ-L and differential Manchester overcomes several problems associated with NRZ-I. First, there is no baseline wandering. There is no DC component because each bit has a positive and negative voltage contribution. The only drawback is the signal rate. The signal rate for Manchester and differential Manchester is double that for NRZ. The reason is that there is always one transition at the middle of the bit and may be one transition at the end of each bit. Figure 4.8 shows both Manchester and differential Manchester encoding. Manchester and differential Manchester schemes are also called Biphase schemes.

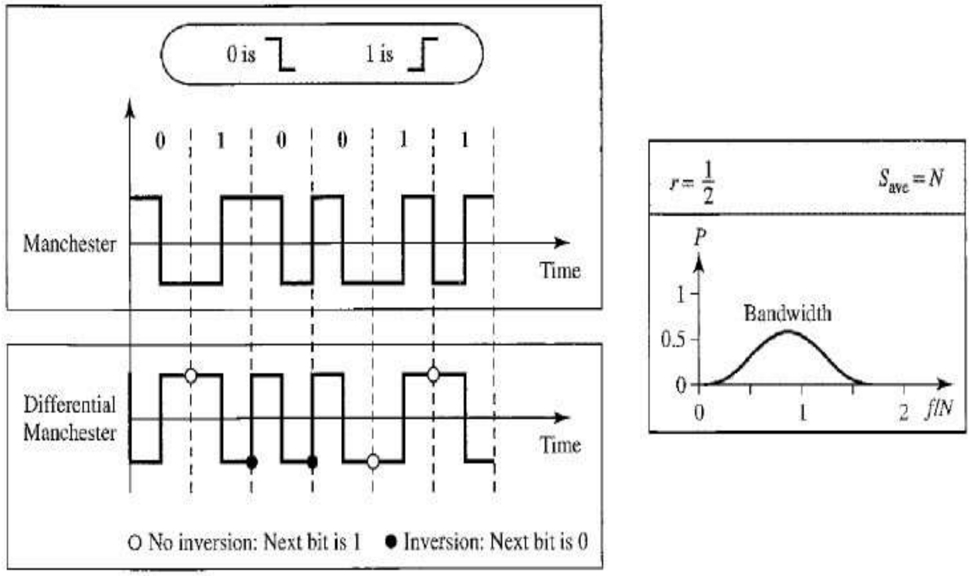


Figure 4.8: Polar Biphase: Manchester and differential Manchester schemes

2.6 Bipolar Schemes

In bipolar encoding (also called multilevel binary), there are three voltage levels. They are positive, negative and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

AMI and Pseudo ternary

There are two variations of bipolar encoding. They are alternate mark inversion (AMI) and Pseudo ternary. A common bipolar encoding scheme is called bipolar alternate mark inversion (AMI). In this, the term mark comes from telegraphy and it means 1. So, AMI means alternate 1 inversion. A neutral zero voltage represents binary 0, binary 1s are represented by alternating positive and negative voltages. A variation of AMI encoding is called *pseudo ternary* in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages. Figure 4.9 shows AMI and pseudo ternary bipolar schemes.

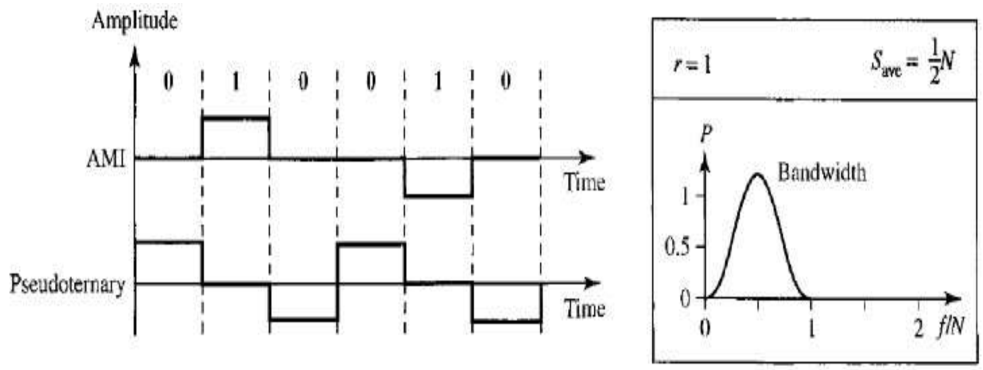


Figure 4.9: Bipolar schemes: AMI and pseudo ternary

The bipolar scheme was developed as an alternative to NRZ. The bipolar scheme has the same signal rate as NRZ, but there is no DC component. The NRZ scheme has most of its energy concentrated near zero frequency, which make it unsuitable for transmission over channels with poor performance around this frequency. The concentration of the energy in bipolar encoding is around frequency N/2. Figure 4.9 shows the typical energy concentration for a bipolar scheme.

We do not have a DC component in bipolar encoding because if we have a long sequence of 1s, the voltage level alternates between positive and negative, it is not constant. For a long sequence of 0s, the voltage remains constant, but its amplitude is zero which is same as having no DC component. That is, a sequence that creates a constant zero voltage does not have a DC component. AMI is commonly used for long-distance communication, but it has a synchronization problem when a long sequence of 0s is present in the data.

1) Multilevel Schemes

The desire to increase the data speed or decrease the required bandwidth has resulted in the creation of many schemes. The goal is to increase the number of bits per baud by encoding a pattern of *m* data elements into a pattern of *n* signal elements. We only have two types of data elements (0s and 1s), which means that a group of m data elements can produce a combination of 2^m data patterns. We can have different types of signal elements by allowing different signal levels. If we have L different levels, then we can produce L^n combinations of signal patterns. If $2^m = L^n$, then each data pattern is encoded into one signal pattern. If $2^m < L^n$, data patterns occupy only a subset of signal patterns.

The subset can be carefully designed to prevent baseline wandering, to provide synchronization, and to detect errors that occurred during data transmission. Data encoding is not possible if $2^m > L^n$ because some of the data patterns cannot be encoded.

Code designers have classified these types of coding as mBnL, where m is the length of the binary pattern, B means binary data, n is the length of the signal pattern, and L is the number of levels in the signaling. A letter is used in place of L. that is B (binary) for L=2, T (ternary) for L=3 and Q (quaternary) for L=4. In *mBnL*, first two letters define the data pattern, second two define the signal pattern.

2B1Q

The first *mBnL* scheme, ***two binary, one quaternary (2B1Q)*** uses data patterns of size 2 and encodes the 2-bit patterns as one signal element belonging to a four-level signal.in this type of encoding m = 2, n = 1 and L= 4 (quaternary). A 2B1Q signal is shown in figure 4.10.

	Previous level: positive	Previous level: negative
Next bits	Next level	Next level
00	+1	−1
01	+3	−3
10	−1	+1
11	−3	+3

Transition table

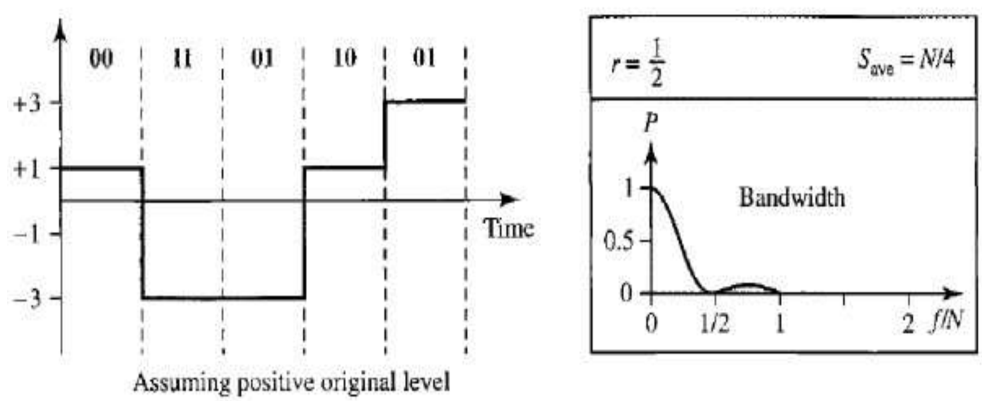


Figure 4.10: Multilevel 2B1Q scheme

The average signal rate of 2B1Q is S=N/4. That is, we can transfer data 2 times faster than by using NRZ-L. But, 2B1Q uses four different signal levels, which means the receiver has to distinguish four different thresholds. There are no redundant signal patterns in this scheme because $2^2 = 4^1$. This scheme is used in DSL (digital subscriber line) technology to provide a high- speed connection to the internet by using subscriber telephone lines.

8B6T

Another interesting mBnL scheme is ***eight binary, six ternary (8B6T)***. This code is used with 100BASE-4T cable. The idea behind this scheme is to encode a pattern of 8 bits as a pattern of 6 signal elements, where the signal has three levels (ternary). In this scheme, we have $2^8 = 256$ different data patterns and $3^6 = 478$ different signal patterns. There are 478 - 256 = 222, redundant signal elements which provide synchronization and error detection. Part of the redundancy is also used to provide DC balance. Each signal pattern has a weight of 0 or +1 DC values. This means that there is no pattern with the weight -1. To make the whole stream DC balanced, the sender keeps track of the weight. If two groups of weight 1 are encountered one after another, the first one is sent as it is and the next one is totally inverted to give a weight of -1. Figure 4.11 shows 8B6T scheme. Here, three data patterns encoded as three signal patterns. The three possible signal levels are represented as -, 0 and +.

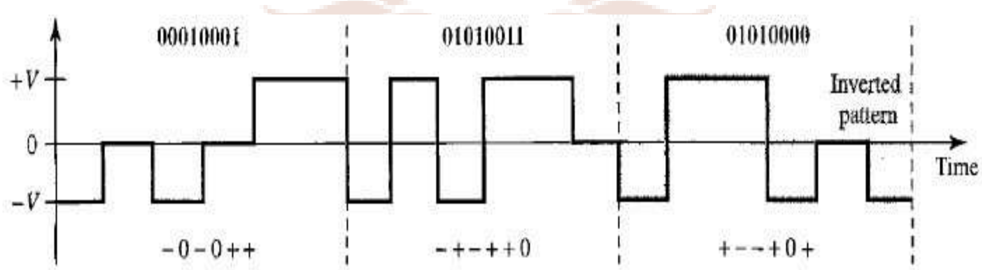


Figure 4.11: Multilevel 8B6T scheme

The first 8-bit pattern 00010001 is encoded as the signal pattern -0-0++ with weight 0; the second 8-bit pattern 01010011 is encoded as - + - ++0 with weight +1. The third bit pattern should be encoded as + - - +0+ with weight +1. To create DC balance, the sender inverts the actual signal. The receiver can easily recognize that this is an inverted pattern because the weight is - 1. The pattern is inverted before decoding. The average signal rate of the scheme is theoretically S_{ave} = ½ $_x$ N $_x$ 6/8. In practice, the minimum bandwidth is very close to 6N/8.

4D-PAM5

Last signaling scheme in this category is called ***four-dimensional five- level pulse amplitude modulation (4D-PAM5)***. The 4D means that data is sent over four wires at the same time. It uses five voltage levels such as -2, -1, 0, 1 and 2. Level 0 is used only for forward error detection. If we assume that the code is just one-dimensional, the four levels creates something similar to 8B4Q i.e., an 8-bit word is translated to a signal element of four different levels. The worst signal rate for this imaginary one- dimensional version is N_x 4/8 or N/2.

This technique is designed to send data over four channels (four wires). This means the signal rate can be reduced to N/8. All 8 bits can be fed into a wire simultaneously and sent by using one signal element. The point here is that the four signal elements comprising one signal group are sent simultaneously in a four-dimensional setting. Figure 4.12 shows the imaginary one-dimensional and actual four-dimensional implementation.

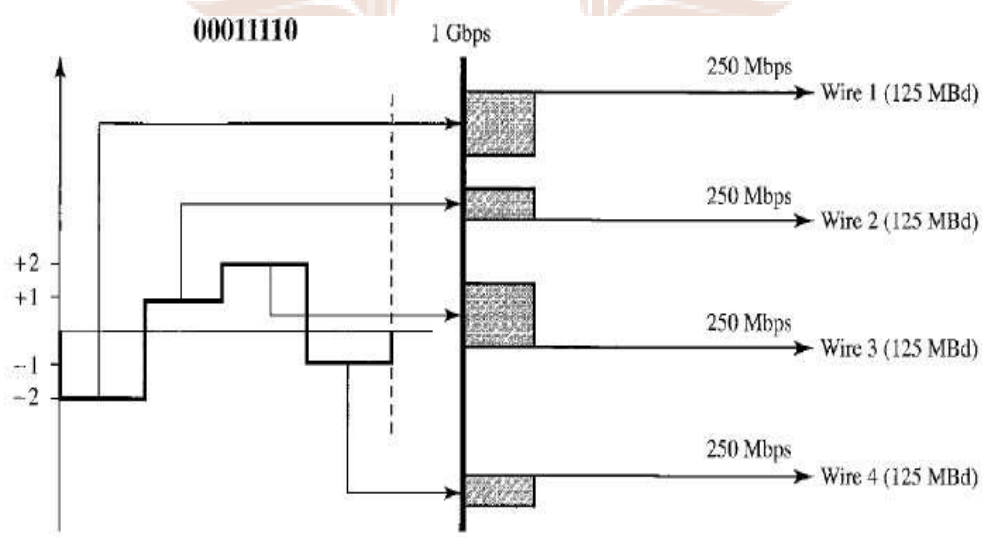


Figure 4.12: Multilevel 4D-PAM5 scheme

Gigabit LANs use this technique to send 1Gbpa data over four copper cables that can handle 125 M baud. This scheme has a lot of redundancy in the signal pattern because 28 data patterns are matched to 44 = 256 signal patterns. The extra signal patterns can be used for other purposes such as error detection.

2) Multiline transmission: MLT-3

NRZ-I and differential Manchester are classified as differential encoding but use two transition rules to encode binary data. If we have a signal with more than two levels, we can design a differential encoding scheme with more than two transition rules. MLT-3 is one of them. The Multiline Transmission, Level-3 (MLT-3) scheme uses three levels (+V, 0, and –V) and three transition rules to move between the levels.

a) If the next bit is 0, there is no transition
b) If the next bit is 1 and the current level is not 0, the next level is 0.
c) If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level.

Figure 4.13 shows the behavior of MLT-3. The three voltage levels (-V, 0, and +V) are shown by three states (ovals). The transition from one state to another is shown by the connecting lines. Figure 4.13 also shows two examples of an MLT-3 signal.

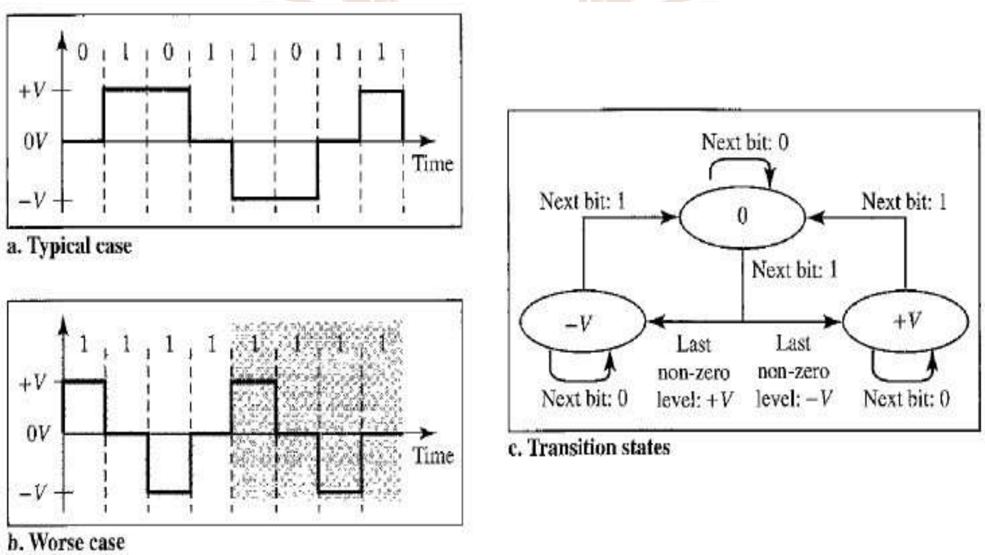


Figure 4.13: Multitransition: MLT-3 scheme

In MLT-3 scheme, signal rate is same as that for NRZ-I, but with greater complexity (three levels and complex transition rules). It turns out that the shape of the signal in this scheme helps to reduce the required bandwidth. When considering the worst-case scenario, a sequence of 1s, the signal element pattern +V0-V0 is repeated every 4 bits.

A nonperiodic signal has changed to a periodic signal with the period equal to 4 times the bit duration. This worst-case situation can be simulated as an analog signal with a frequency one fourth of the bit rate. In other words, the signal rate for MLT-3 is one fourth the bit rate. So MLT-3 is a suitable choice when we need to send 100 Mbps on a copper wire that cannot support more than 32 MHz (frequencies above this level create electromagnetic emissions). Table 4.1 shows the summary of different line coding schemes.

Table 4.1: Summary of line coding schemes

Category	Scheme	Bandwidth (average)	Characteristics
Unipolar	NRZ	B = N/2	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	B = N/2	No self-synchronization if long 0s or 1s, DC
	NRZ-I	B = N/2	No self-synchronization long for 0s, DC
	Biphase	B = N	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	B = N/2	No self-synchronization long for 0s, DC
Multilevel	2B1Q	B = N/4	No self-synchronization long for same double bits
	8B6T	B = 3N/4	Self-synchronization, no DC
	4D-PAM5	B = N/8	Self-synchronization, no DC
Multiline	MLT -3	B = N/3	No self-synchronization long for 0s

2.7 Scrambling

Biphase schemes that are suitable for dedicated links between stations in a LAN are not suitable for long-distance communication because of their wide bandwidth requirement. The combination of block coding and NRZ line coding is not suitable for long-distance encoding either, because of the DC component. Bipolar AMI encoding, on the other hand has a narrow bandwidth and does not create a DC component. However, a long sequence of 0s upsets the synchronization. If we can find a way to avoid a long sequence of 0s in the original stream, we can use bipolar AMI for long distances. We need a technique that does not increase the number of bits and does provide synchronization. We need a solution that substitutes long zero-level pulses with a combination of other levels to provide synchronization. One solution for this is called scrambling. We modify a part of AMI rule to include scrambling. Figure 4.14 shows AMI used with scrambling.

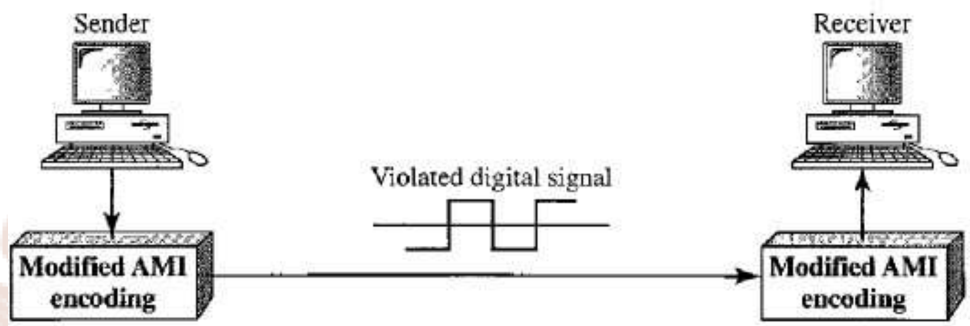


Figure 4.14: AMI used with scrambling

Scrambling is done at the same time as encoding. The system needs to insert the required pulses based on the defined scrambling rules. Two common scrambling techniques are B8ZS and HDB3.

B8ZS

Bipolar with 8-zero substitution (B8ZS) is commonly used in North America. In this technique, eight consecutive zero level voltages are replaced by the sequence.

HDB3

High-density bipolar 3-zero (HDB3) is commonly used scrambling technique, outside of North America. This technique is more conservative than B8ZS, four consecutive zero-level voltages are replaced with a sequence of 000V or B00V. The reason for two different substitutions is to maintain the even number of nonzero pulses after each substitution. The two rules can be stated as follows:

1. If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be 000V, which makes the total number of nonzero pulses even.
2. If the number of nonzero pulses after the last substitution is even, the substitution pattern will be B00V, which makes the total number of nonzero pulses even.

Self-Assessment Questions - 1

1. The process of converting digital data to digital signals is known as________.
2. The smallest entity that represent a piece of information is known as_________.
3. The unit of data rate is __________ .
4. The running average of the received signal power is known as_____________.
5. Which of the following scheme, the voltages are on the both sides of the time axis?
   a) unipolar b) polar c) multilevel d) multitransition
6. Two variations of bipolar encoding are __________ and _______________.

3. ANALOG-TO-DIGITAL CONVERSION

Data can be transmitted in the form of analog signal as well as digital signal. In section 4.2, we discussed digital data to digital signal conversion. Also, we have analog signals such as the signals created by a microphone or camera. Digital signal is superior to analog signal. So nowadays, we convert analog signal to digital data. After the digital data is created (the technique is called digitization), we can use any of the two techniques to convert digital data to digital signal. The two techniques are: pulse code modulation and delta modulation.

3.1 Pulse Code Modulation (PCM)

Pulse code modulation (PCM) is the most common technique to change an analog signal to digital data (digitization).

A PCM encoder has three processes:

1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

Figure 4.15 illustrates these three processes.

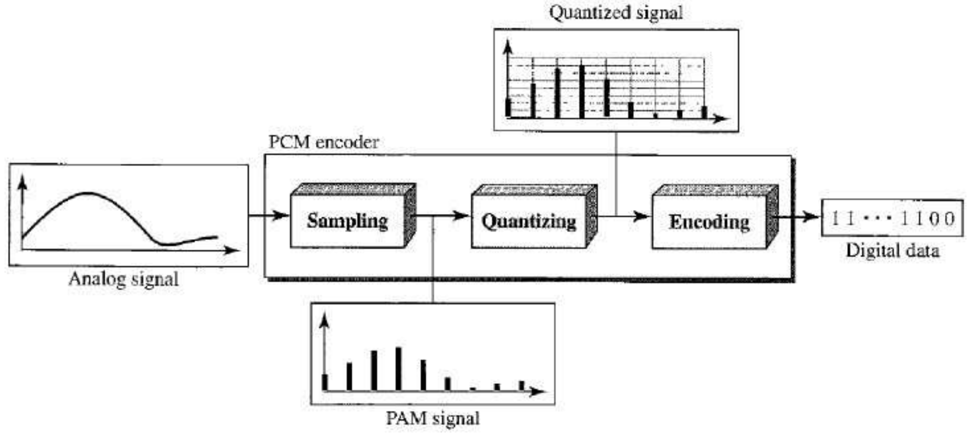


Figure 4.15: Components of PCM encoder

Sampling

The first step in PCM is sampling. The analog signal is sampled every T_S second, where T_S is the sample interval or period. The inverse of the sampling interval is called sampling rate or sampling frequency and denoted by f_S, where $f_S = 1/T_S$. There are three sampling methods: *ideal, natural* and *flat-top*. This is shown in figure 4.16.

In ideal sampling, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented. In natural sampling, a high-speed switch is turned on for only small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal. The most common sampling method known as sample and hold creates flat-top samples by using a circuit.

The sampling process is also known as pulse amplitude modulation (PAM). Another important consideration is sampling rate or frequency. According to Nyquist theorem, to reproduce the original analog signal, the sampling rate must be at least 2 times the highest frequency contained in the signal.

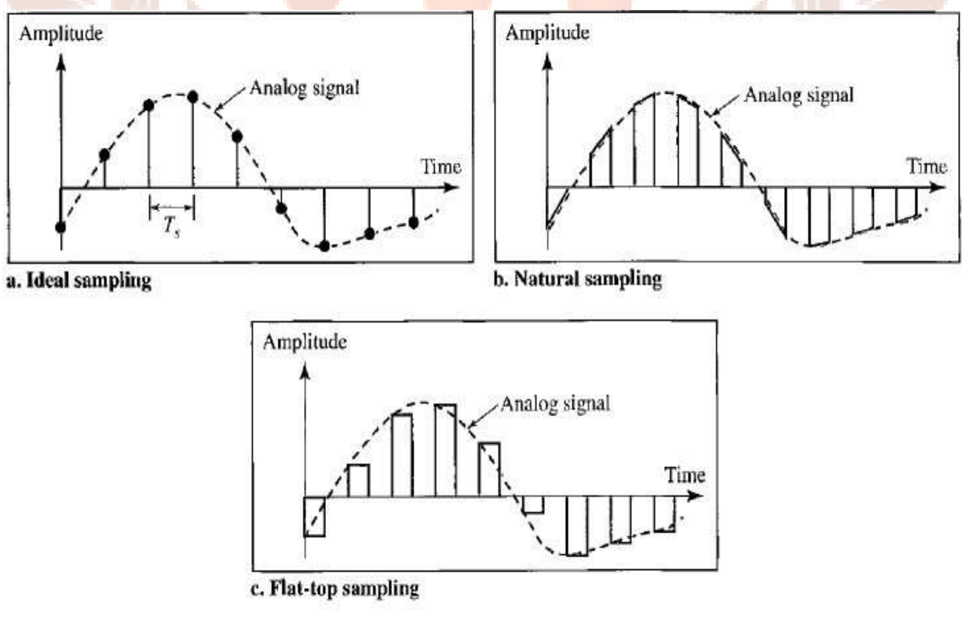


Figure 4.16: Three different sampling methods for PCM

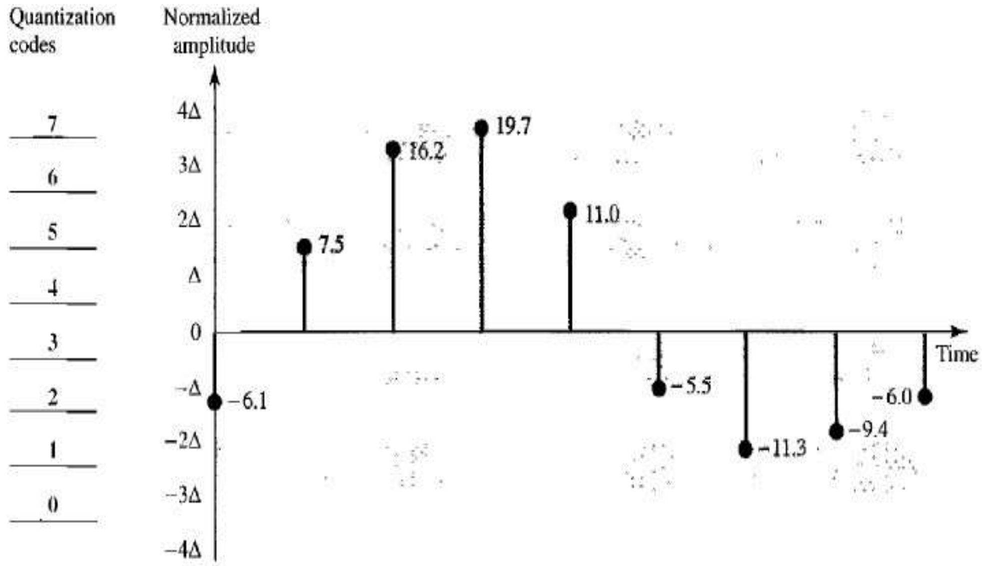
Quantization

The result of sampling is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal. The set of amplitudes can be infinite with nonintegral values between the two limits. These values cannot be used in the encoding process. The following are the steps in quantization:

1. We assume that the original analog signal has instantaneous amplitudes between V_{min} and V_{max}.
2. We divide the range into L zones, each of height Δ (delta)
   $\Delta = (V_{max} - V_{min}) / L$
3. We assign quantized values of 0 to L-1 to the midpoint of each zone.
4. We approximate the value of the sample amplitude to the quantized values.

Consider a simple example, assume that we have a sampled signal and the sample amplitudes are between -20 and +20 V. We decide to have eight levels (L=8). This means that Δ = 5V. Figure 4.17 shows this example.

In the figure, nine samples using ideal sampling is shown. The value at the top of each sample in the graph shows the actual amplitude. In the chart, first row is the normalized value for each sample (actual amplitude/Δ). The quantization process selects the quantization value from the middle of each zone. This means that the normalized quantized values (second row) are different from the normalized amplitudes. The difference is called normalized error (third row). The fourth row is the quantization code for each sample based on the quantization levels at the left of the graph. The encoded words (fifth row) are the final products of the conversion.



Normalized PAM values	−1.22	1.50	3.24	3.94	2.20	−1.10	−2.26	−1.88	−1.20
Normalized quantized values	−1.50	1.50	3.50	3.50	2.50	−1.50	−2.50	−1.50	−1.50
Normalized error	−0.38	0	+0.26	−0.44	+0.30	−0.40	−0.24	+0.38	−0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

Figure 4.17: Quantization and encoding of a sampled signal

Encoding

Encoding is the last step in PCM. After each sample is quantized and number of bits per sample is decided, each sample can be changed to n_b (bit per sample is denoted as n_b) bit code word. In figure 4.17 the encoded words are shown in the last row. A quantization code of 2 is encoded as 010; 5 is encoded as 101 and so on. The number of bits for each sample is determined by the number of quantization levels. If the number of quantization levels is L, the number of bits is $n_b = \log_2 L$. in our example, L is 8 and nb is therefore 3. Bit rate can be found using the formula

$$\textit{Bit rate} = \textit{sampling rate} \times \textit{number of bits per sample} = f_s \times n_b$$

Original Signal Recovery

The recovery of original signal requires PCM decoder. The decoder first uses circuitry to convert the code words into a pulse that holds the amplitude until the next pulse. After the staircase signal is completed, it is passed through a low-pass filter to smooth the staircase signal into an analog signal. Figure 4.18 shows the process.

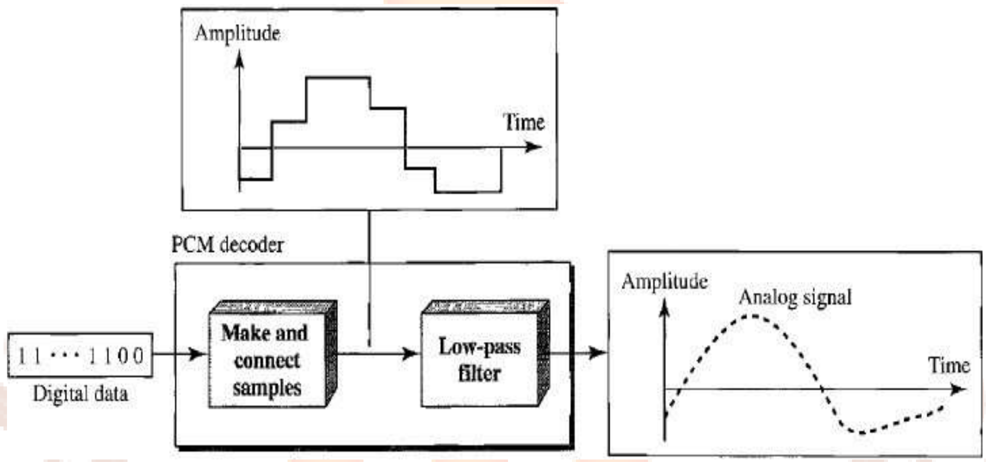


Figure 4.18: Components of a PCM decoder

The filter has the same cut off frequency as the original signal at the sender. If the signal has been sampled at the Nyquist sampling rate and if there are enough quantization levels, the original signal will be recreated. The maximum and minimum values of the original signal can be achieved by using amplification.

3.2 Delta Modulation (DM)

Delta modulation is the simpler one when compared to PCM. PCM finds the value of the signal amplitude for each sample, DM finds the change from the previous sample. Quantization error (error created in the quantization process) of DM is much less than that for PCM. There are no codewords here, bits are sent one after another. Figure 4.19 shows the process.

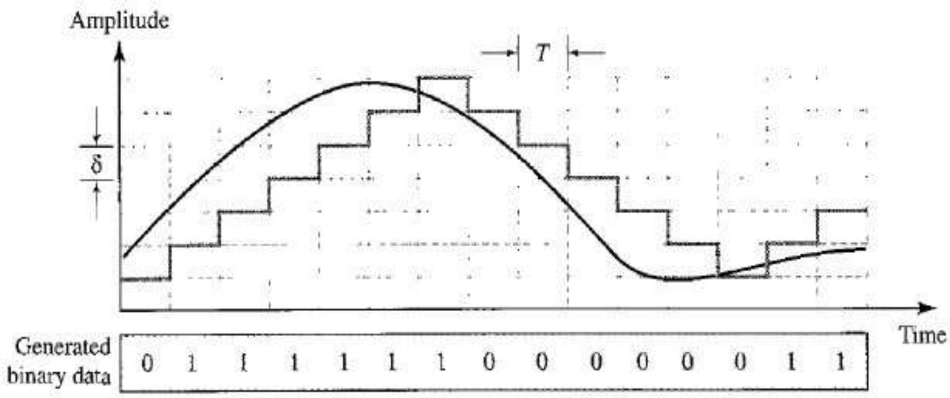


Figure 4.19: The process of delta modulation

Modulator

The modulator is used at the sender site to create a stream of bits from an analog signal. The process records the small positive or negative changes called delta δ. If delta is positive, the process records a 1; if it is negative, the process records a 0. The process needs a base against which the analog signal is compared. The modulator builds a second signal that is known as staircase signal as it resembles a staircase. We can find the variation by comparing the input signal with staircase signal. Figure 4.20 illustrates modulation process.

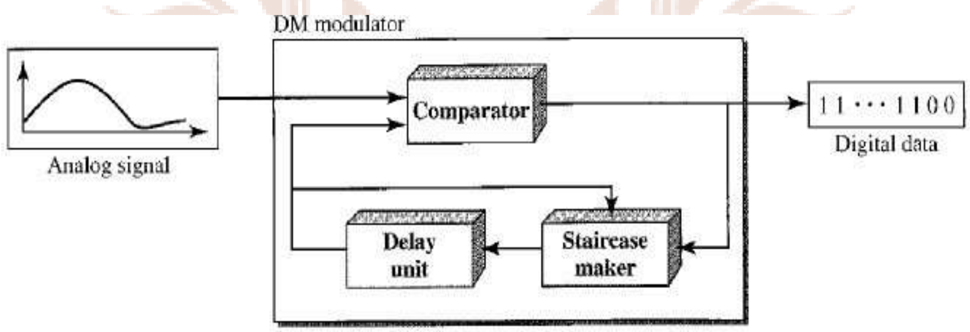


Figure 4.20: Delta modulation components

The modulator at each sampling interval compares the value of the analog signal with the last value of the staircase signal. If the amplitude of the analog signal is larger, the next bit in the digital data is 1 and otherwise it is 0. The output of the comparator also makes the staircase itself. If the next bit is 1, the staircase maker moves the last point of staircase signal δ up. If the next bit is 0, it moves it down. We need delay unit to hold staircase function for a period between two comparisons.

Demodulator

Demodulator takes the digital data and using the staircase maker and delay unit, it creates the analog signal. This analog signal then passes through a low pass filter for smoothing. Figure 4.21 illustrates demodulation process.

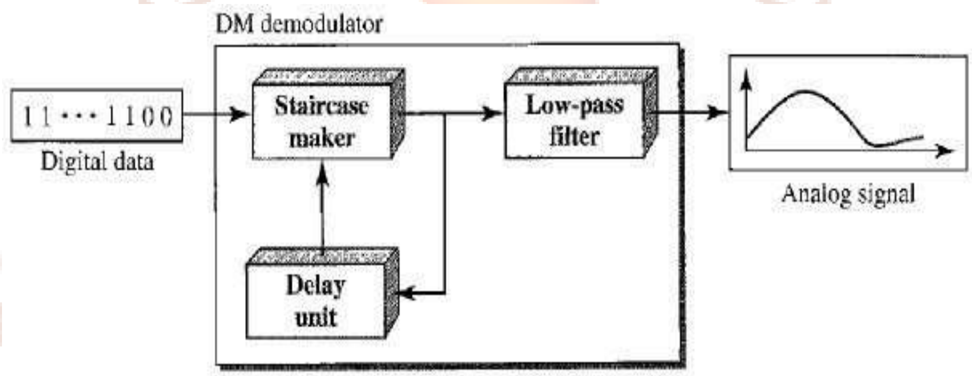


Figure 4.21: Delta demodulation components

Adaptive DM

We can achieve a better performance if the value of δ is not fixed. In adaptive delta modulation, the value of δ changes according to the amplitude of the analog signal.

Self-Assessment Questions - 2

7. Two techniques for converting digital data to digital signal are______ and_______.
8. The first step in PCM is __________.
9. The three sampling methods are__________ ,___________ and__________ .

4. TRANSMISSION MODES

Transmission of binary data across a link can be accomplished in either parallel or serial mode. In parallel mode, multiple bits are sent with each clock tick and in serial mode, 1 bit is sent with each clock tick. There is only one way to send parallel data but there are three subclasses of serial transmission, they are asynchronous, synchronous and isochronous. Figure

4.22 illustrates data transmission and modes.

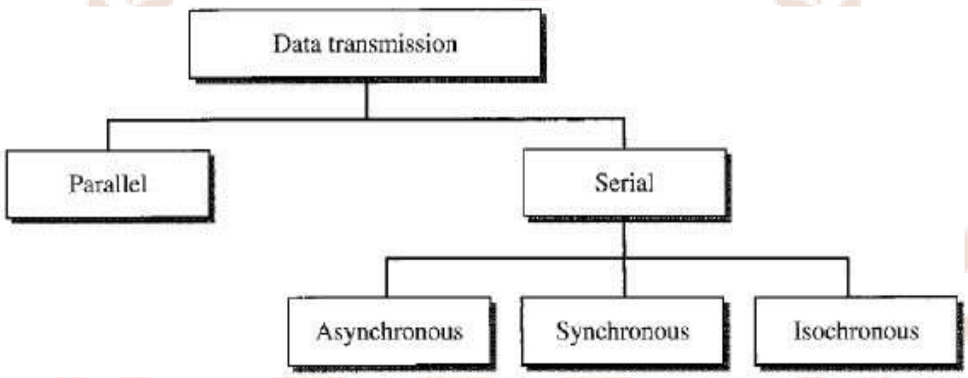


Figure 4.22: Data transmission and modes

Parallel Transmission

Binary data may be organized into group of n bits each. By grouping, we can send data n bits at a time instead of 1. This is called parallel transmission. In case of parallel transmission, we use n wires to send n bits at one time. That is, each bit has its own wire and all n bits of one group can be transmitted with each clock tick from one device to another. Figure 4.23 shows parallel transmission when n=8. Eight wires are bundled in a cable with a connector at each end.

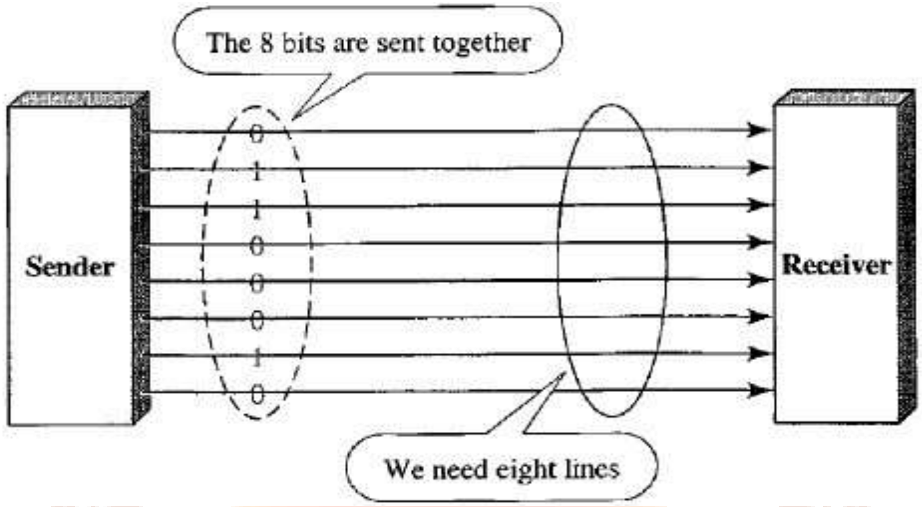


Figure 4.23: Parallel transmission

The advantage of parallel transmission is speed. Parallel transmission can increase the transfer speed by a factor of n over serial transmission. Disadvantage is cost, because parallel transmission requires n communication lines to transmit data stream. Since this is expensive, parallel transmission is limited to short distances.

Serial Transmission

In serial transmission, one bit follows another. Instead of n, we need only one communication channel to transmit data between two communicating devices. Figure 4.24 shows serial transmission.

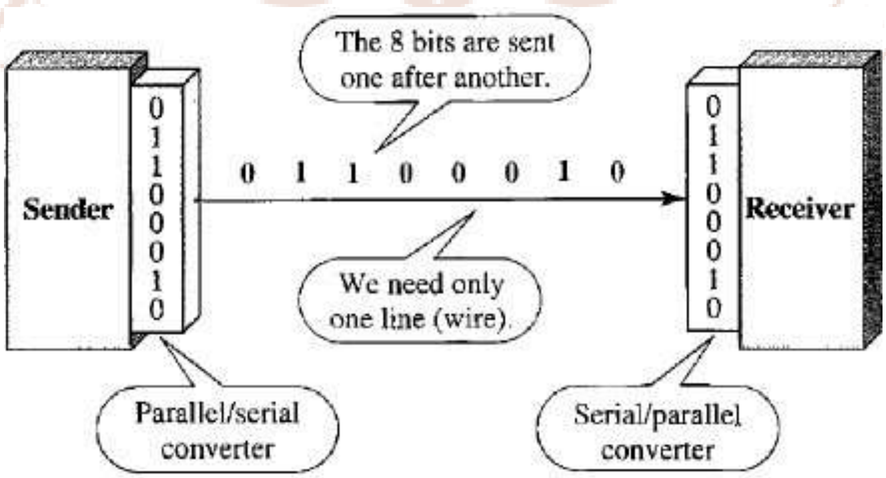


Figure 4.24: Serial transmission

The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by a factor of n. since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (Parallel to serial) and between the line and the receiver (serial to parallel).

a) Asynchronous Transmission

Asynchronous transmission is so called because the timing of a signal is unimportant. In asynchronous transmission, information is received and translated by agreed upon patterns. As long as those patterns are followed, the receiving device can retrieve information without regard to the rhythm in which it is sent. Patterns are based on grouping the bit stream into bytes. Each group with 8 bits is sent along the link as a unit. Sending system handles each group independently, relaying it to the link whenever ready without regard to a timer.

In asynchronous transmission, without synchronization, the receiver cannot use timing to predict when the next group will arrive. An extra bit is used to alert the receiver about the arrival of a new group. We send one start bit (usually 0s) at the beginning of each byte and one or more stop bits (usually 1s) at the end of each byte. There may be a gap between each byte. This gap can be represented either by an idle channel or by a stream of additional stop bits. Here, asynchronous means, asynchronous at the byte level, but the bits are still synchronized and their durations are the same. Figure 4.25 shows asynchronous transmission.

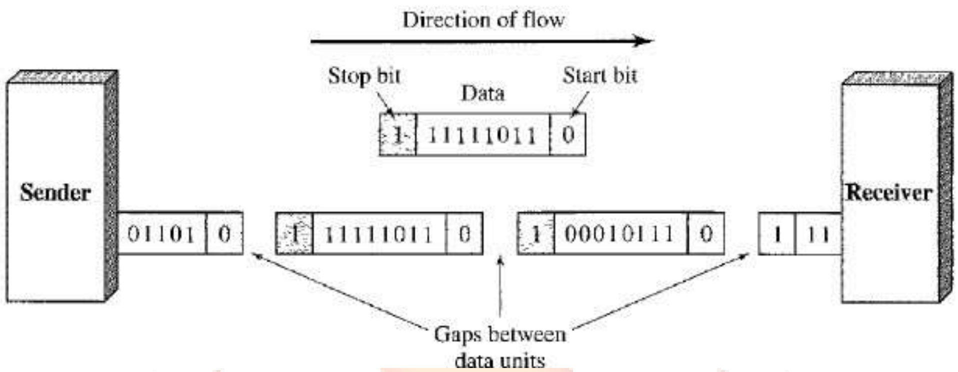


Figure 4.25: Asynchronous transmission

The addition of start and stop bits and the insertion of gaps into the bit stream make asynchronous transmission slower than forms of transmission that can operate without the addition of control information. But it is cheap and effective and these two advantages make it an attractive choice for situations such as low-speed communication. For instance, the connection of a keyboard to a computer is a natural application for asynchronous transmission.

b) Synchronous transmission

In synchronous transmission, the bit stream is combined into longer frames which may contain multiple bytes. Each byte is introduced onto the transmission link without a gap between it and the next one. Receiver is responsible to separate the bit stream into bytes for decoding purposes. That means, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

The sender puts its data onto the link as one long string. If sender wishes to send data as separate bursts, the gap between bursts must be filled with a special sequence of 0s and 1s that means idle. The receiver counts the bits as they arrive and groups them in 8-bit units. Figure 4.26 illustrates synchronous transmission.

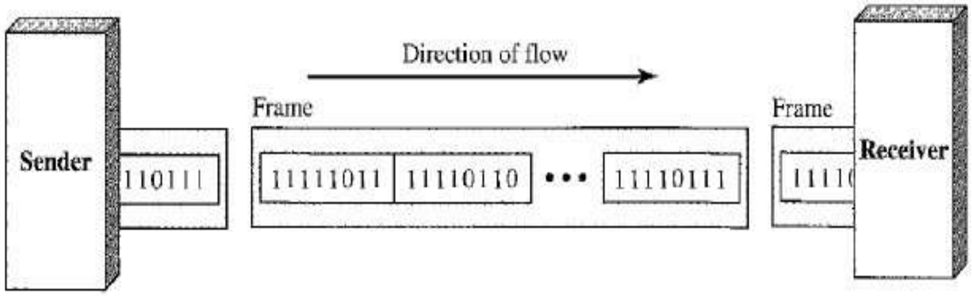


Figure 4.26: Synchronous transmission

The advantage of synchronous transmission is speed. It is faster than asynchronous transmission due to the absence of extra bits. So, this is more useful for high speed applications such as the transmission of data from one computer to another.

c) Isochronous transmission

In real time audio and video, uneven delays between frames are not acceptable, in that case synchronous transmission fails. For instance, TV images are broadcast at the rate of 30 images per second, then they must be viewed at the same rate. If each image is sent by using one or more frames, there should be no delays between frames. For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized. The isochronous transmission guarantees that the data arrive at a fixed rate.

Self-Assessment Questions - 3

10. Transmission of binary data across a link can be accomplished in either ___________ or ___________ mode.
11. In ___________ transmission, we need only one communication channel to transmit data between two communicating devices.
12. In _________ transmission, we can send n bits at a time.

5. SUMMARY

Let us recapitulate the important concepts discussed in this unit:

- Line coding is the process of converting digital data to digital signals.
- A data element is the smallest entity that can represent a piece of information. This is known as the bit.
- Signal element is the shortest unit of a digital signal.
- The signal element carries data elements in digital communication.
- The data rate defines the number of data elements sent in 1 second. The unit is *bits* per second (bps). The signal rate is the number of signal elements sent in one second. The unit is *baud*.
- Line coding schemes can be divided into five broad categories, they are: unipolar, polar, bipolar, multilevel, multitransition.
- The two techniques to convert digital data to digital signal are pulse code modulation and delta modulation.
- Transmission of binary data across a link can be accomplished in either parallel or serial mode.
- In parallel mode, multiple bits are sent with each clock tick and in serial mode, 1 bit is sent with each clock tick.

6. TERMINAL QUESTIONS

1. Explain different line coding schemes.
2. Differentiate between pulse code modulation and delta modulation.
3. Differentiate between parallel and serial transmission.
4. Write short note on asynchronous, synchronous and isochronous transmission.

7. ANSWERS

Self-Assessment Questions

1. Line coding
2. Data element / bit
3. Bits per second
4. Baseline
5. (b) polar
6. Alternate mark inversion (AMI), Pseudoternary
7. Pulse code modulation (PCM), Delta modulation (DM)
8. Sampling
9. Ideal, natural and flat-top
10. Parallel, serial
11. Serial
12. parallel

Terminal Questions

1. Line coding schemes can be divided into five broad categories, they are: unipolar, polar, bipolar, multilevel, multitransition. (Refer section 2.2 for detail).
2. After the digital data are created (the technique is called digitization), we can use any of the two techniques to convert digital data to digital signal. The two techniques are: pulse code modulation and delta modulation. (Refer section 3 for detail).
3. Binary data may be organized into group of n bits each. By grouping, we can send data n bits at a time instead of 1. This is called parallel transmission. In serial transmission, one bit follows another. Instead of n, we need only one communication channel to transmit data between two communicating devices. (Refer section 4 for detail).
4. There are three subclasses of serial transmission, they are asynchronous, synchronous and isochronous. (Refer section 4 for detail).

References

1. Behrouz A. Forouzan, Sophia Chung Fegan, "Data Communications and Networking", Fourth edition.
2. William Stallings, "Data and Computer Communications", Sixth edition, Pearson Education, Delhi, 2002.
3. Taub and Schilling, "Principles of Communication Systems", Tata Mc Graw Hill, Delhi, 2002.
4. S.Tanenbaum, "Computer Networks", Pearson Education, Fourth Edition.
5. N.Olifer, V.Olifer, "Computer Networks: Principles, technologies and Protocols for Network design", Wiley India Edition, First edition.
6. Simon Poulton (2003), packet switching and X.25 Networking, Pitman publishing.
7. Walrand, P.Varaiya, "high performance communication networks", Morgan kaufmann.