Module 2

Digital Transmission

- A computer network is designed to send information from one point to another.
- This information needs to be converted to either a digital signal or an analog signal for transmission.
- In this chapter, we show the **schemes and techniques** that we use to transmit data digitally.
- First, we discuss **digital-to-digital conversion** techniques, methods which convert digital data to digital signals.
- Second, we discuss **analog-to-digital conversion** techniques, methods which change an analog signal to a digital signal.
- Thirdly we discuss **transmission modes**.
- Finally **digital-to-analog conversion** techniques

4-1 DIGITAL-TO-DIGITAL CONVERSION

In this section, we see how we can represent digital data by using digital signals. The conversion involves three techniques: line coding, block coding, and scrambling. Line coding is always needed; block coding and scrambling may or may not be needed.

Topics discussed in this section:

- **Line Coding**
- **Line Coding Schemes**
- **Block Coding**
- **Scrambling**

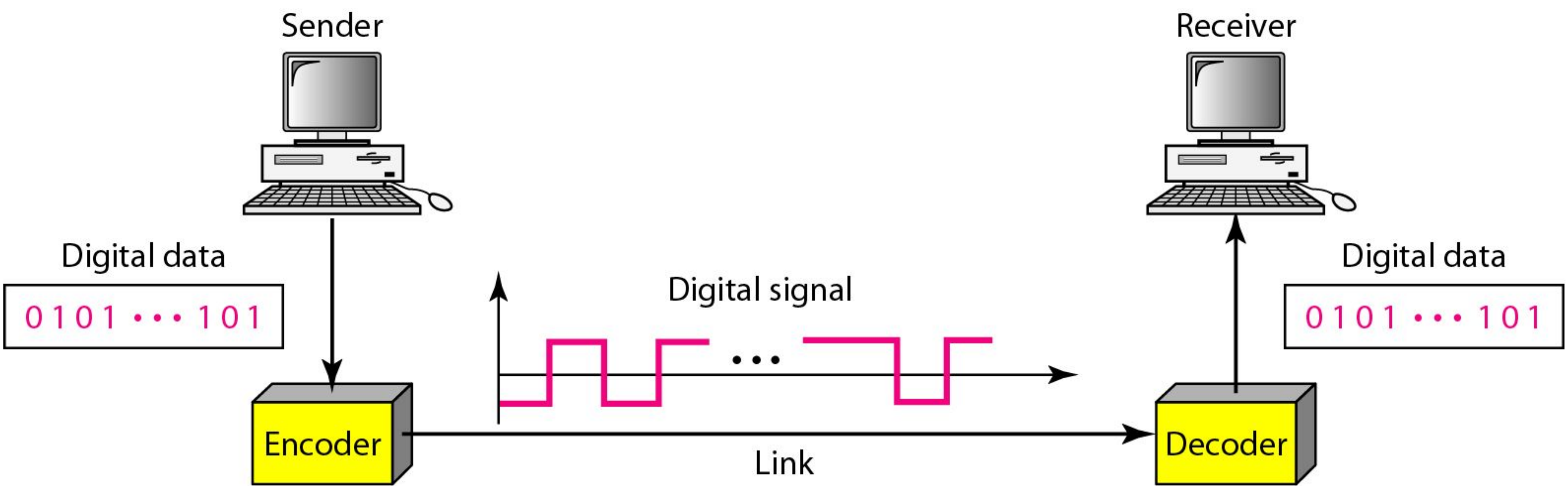
Line Coding

- Converting a string of 1’s and 0’s (digital data) into a sequence of signals that denote the 1’s and 0’s.
- For example a high voltage level (+V) could represent a “1” and a low voltage level (0 or -V) could represent a “0”.

Line Coding

- **Line coding** is the process of converting **digital data to digital signals**. We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits.
- Line coding converts a sequence of bits to a digital signal.
- At the sender, digital data are encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal.

Figure 4.1 *Line coding and decoding*



Mapping Data symbols onto Signal levels

- A data symbol (or element) can consist of a number of data bits:
  - 1 , 0 or
  - 11, 10, 01, ……
- A data symbol can be coded into a single signal element or multiple signal elements
  - 1 -> +V, 0 -> -V
  - 1 -> +V and -V, 0 -> -V and +V
- The ratio ‘r’ is the number of data elements carried by a signal element.

Characteristics

Before discussing different line coding schemes, we address their common characteristics.

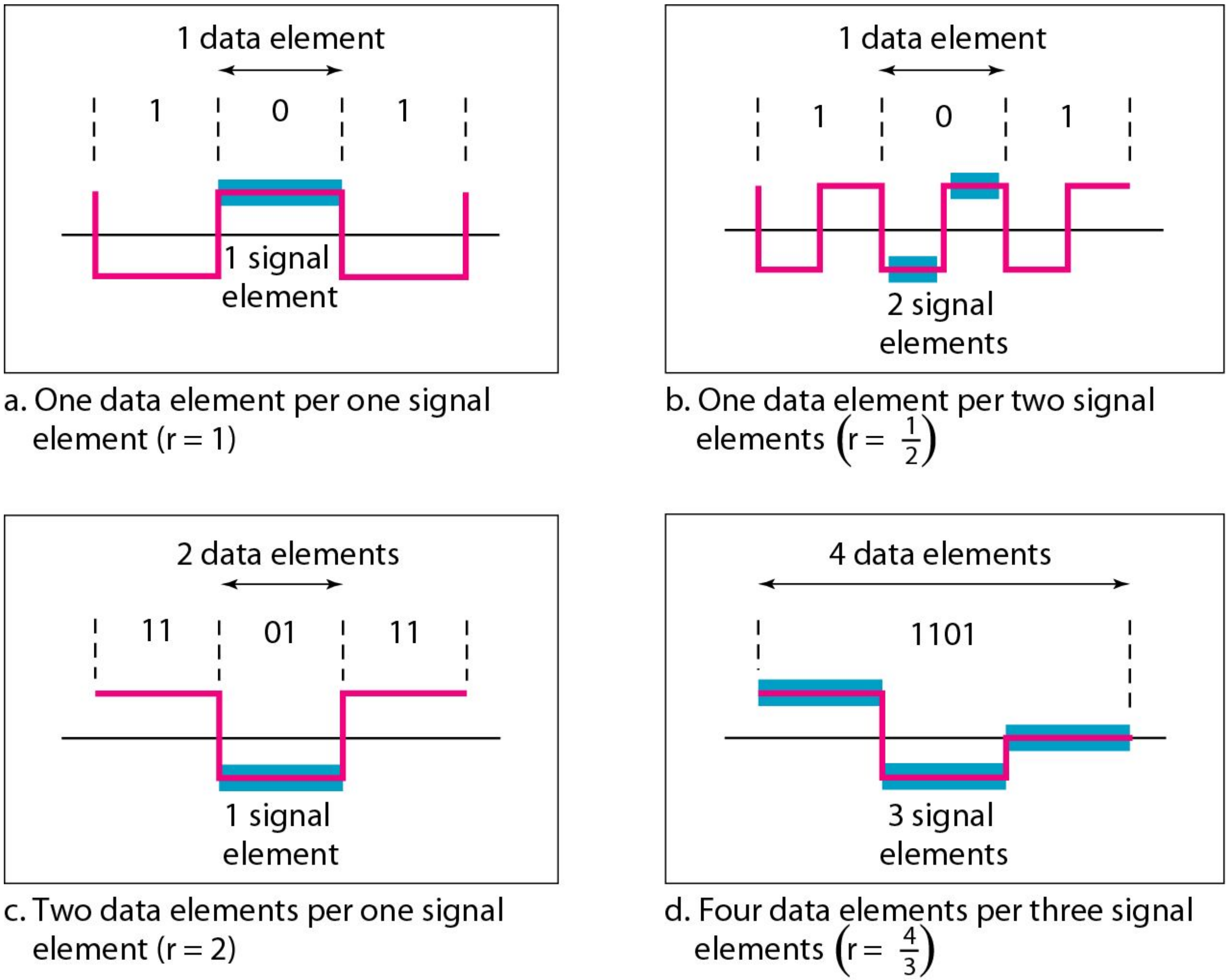
- **Signal Element Versus Data Element**

- Let us distinguish between a data element and a signal element.
- In data communications, our goal is to send data elements. A **data element** is the smallest entity that can represent a piece of information: **this is the bit.**
- In digital data communications, a **signal element** carries data elements.
- In other words, **data elements are what we need to send**; **signal elements are what we can send.**
- Data elements are being **carried**; signal elements are **the carriers**.

Relationship between data rate and signal rate

- The data rate defines the **number of bits sent per sec** - **bps**. It is often referred to the **bit rate.**
- The signal rate is the number of **signal elements sent in a second** and is measured in **bauds**. It is also referred to as the **modulation rate**.
- Goal is to increase the data rate whilst reducing the baud rate.

Figure 4.2 *Signal element versus data element*



a. One data element per one signal element ($r = 1$)

b. One data element per two signal elements $\left(r = \frac{1}{2}\right)$

c. Two data elements per one signal element ($r = 2$)

d. Four data elements per three signal elements $\left(r = \frac{4}{3}\right)$

1. In part a of the figure, one data element is carried by one signal element *(r = 1)*.
2. In part b of the figure, we need two signal elements (two transitions) to carry each data element *(r =1/2)*
3. In part c of the figure, a signal element carries two data elements *(r = 2)*.
4. Finally, in part d, a group of 4 bits is being carried by a group of three signal elements *(r = 4/3)*.
5. For every line coding scheme we discuss, we will give the value of *r*.

Data rate and Baud rate

- The baud or signal rate can be expressed as:

$$S = c \times N \times 1/r \text{ bauds}$$

Where N is data rate

c is the case factor (worst, best & avg.)

r is the ratio between data element & signal element

Example 4.1

A signal is carrying data in which one data element is encoded as one signal element (r = 1). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

Solution

We assume that the average value of c is 1/2 . The baud rate is then

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100{,}000 \times \frac{1}{1} = 50{,}000 = 50 \text{ kbaud}$$

Note

Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

Example 4.2

The maximum data rate of a channel (see Chapter 3) is $N_{max} = 2 \times B \times \log_2 L$ (defined by the Nyquist formula). Does this agree with the previous formula for N_{max}?

Solution

A signal with L levels actually can carry $\log_2 L$ bits per level. If each level corresponds to one signal element and we assume the average case ($c = 1/2$), then we have

$$\boxed{N_{\max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L}$$

Considerations for choosing a good signal element referred to as line encoding

- Baseline wandering - a receiver will evaluate the average power of the received signal (called the baseline) and use that to determine the value of the incoming data elements. If the incoming signal does not vary over a long period of time, the baseline will drift and thus cause errors in detection of incoming data elements.
- A good line encoding scheme will prevent long runs of fixed amplitude.

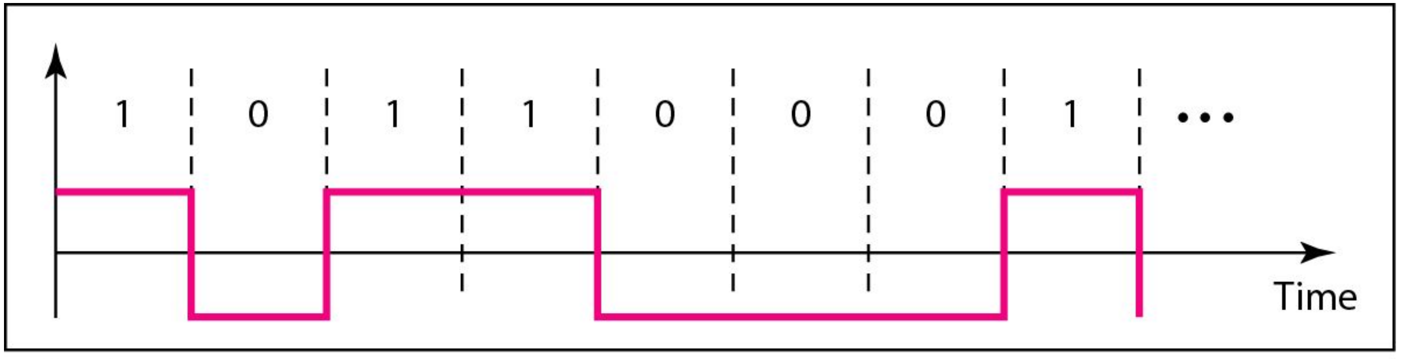
Line encoding

- DC components - when the voltage level remains constant for long periods of time, there is an increase in the low frequencies of the signal. Most channels are bandpass and may not support the low frequencies.
- This will require the removal of the dc component of a transmitted signal.

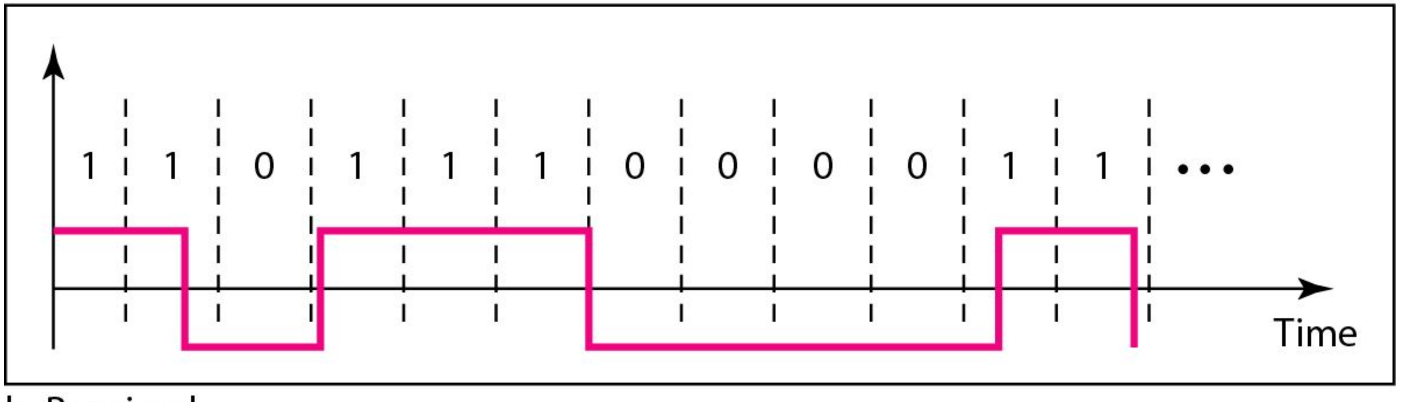
Line encoding

- Self synchronization - the clocks at the sender and the receiver must have the same bit interval.
- If the receiver clock is faster or slower it will misinterpret the incoming bit stream.

Figure 4.3 *Effect of lack of synchronization*



a. Sent



b. Received

Example 4.3

In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 kbps? How many if the data rate is 1 Mbps?

Solution

At 1 kbps, the receiver receives 1001 bps instead of 1000 bps.

1000 bits sent	1001 bits received	1 extra bps

At 1 Mbps, the receiver receives 1,001,000 bps instead of 1,000,000 bps.

1,000,000 bits sent	1,001,000 bits received	1000 extra bps

Line encoding

- **Error detection** - errors occur during transmission due to line impairments.
- Some codes are constructed such that when an error occurs it can be detected.
- For example: a particular signal transition is not part of the code. When it occurs, the receiver will know that a symbol error has occurred.

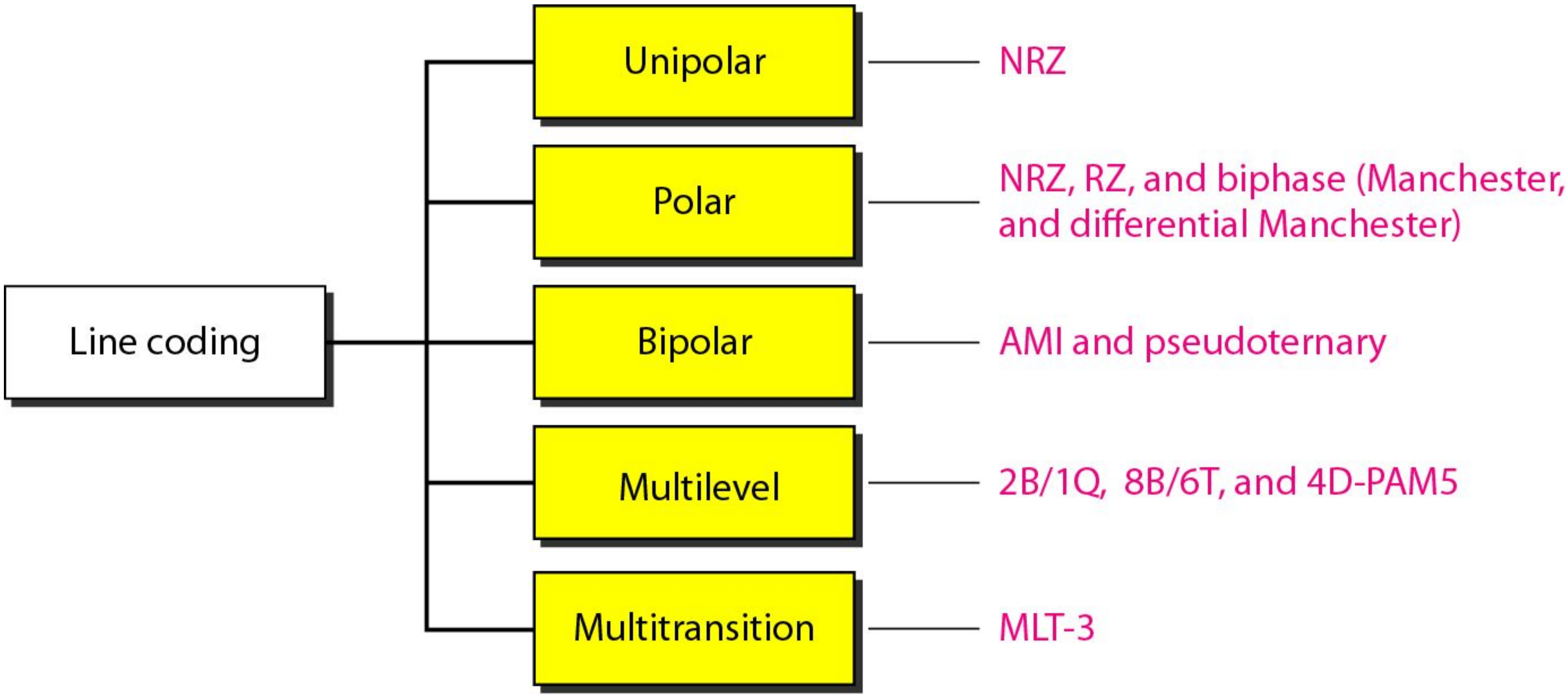
Line encoding

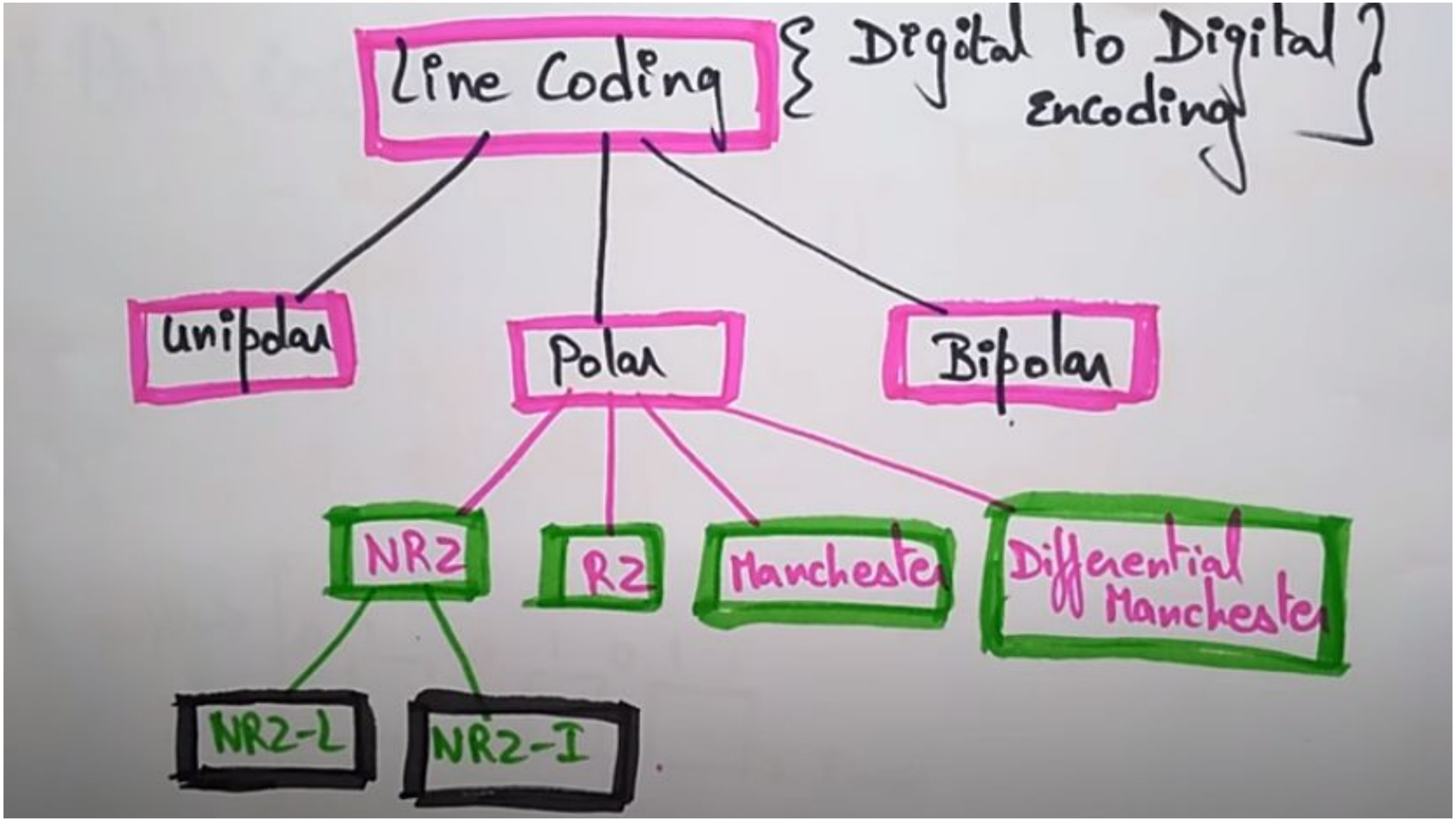
- Noise and interference - there are line encoding techniques that make the transmitted signal "immune" to noise and interference.
- This means that the signal cannot be corrupted, it is stronger than error detection.

Line encoding

- Complexity - the more robust and resilient the code, the more complex it is to implement and the price is often paid in baud rate or required bandwidth.

Figure 4.4 *Line coding schemes*

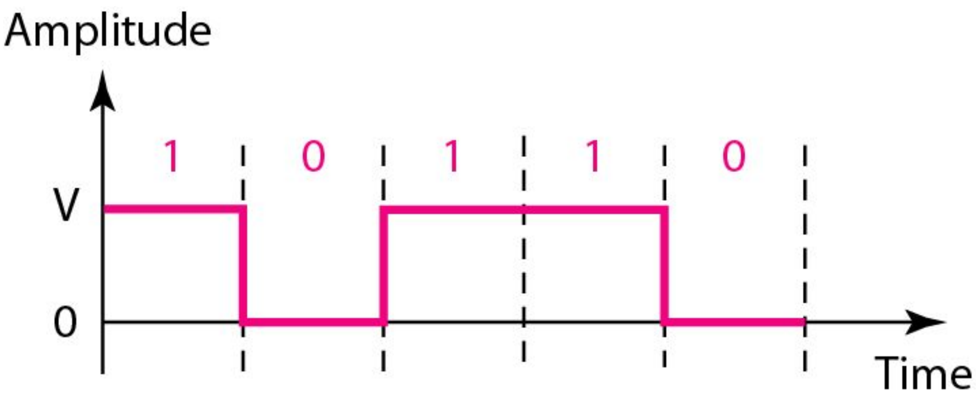



Line Coding
{ Digital to Digital Encoding }
unipolar
Polar
Bipolar
NRZ
RZ
Manchester
Differential Manchester
NRZ-L
NRZ-I

1. Unipolar

- **All signal levels are on one side of the time axis - either above or below**
- NRZ - Non Return to Zero scheme is an example of this code. The signal level does not return to zero during a symbol transmission.
- Scheme is prone to baseline wandering and DC components. It has no synchronization or any error detection. It is simple but costly in power consumption.

Unipolar NRZ Scheme



$$\frac{1}{2}V^2 + \frac{1}{2}(0)^2 = \frac{1}{2}V^2$$

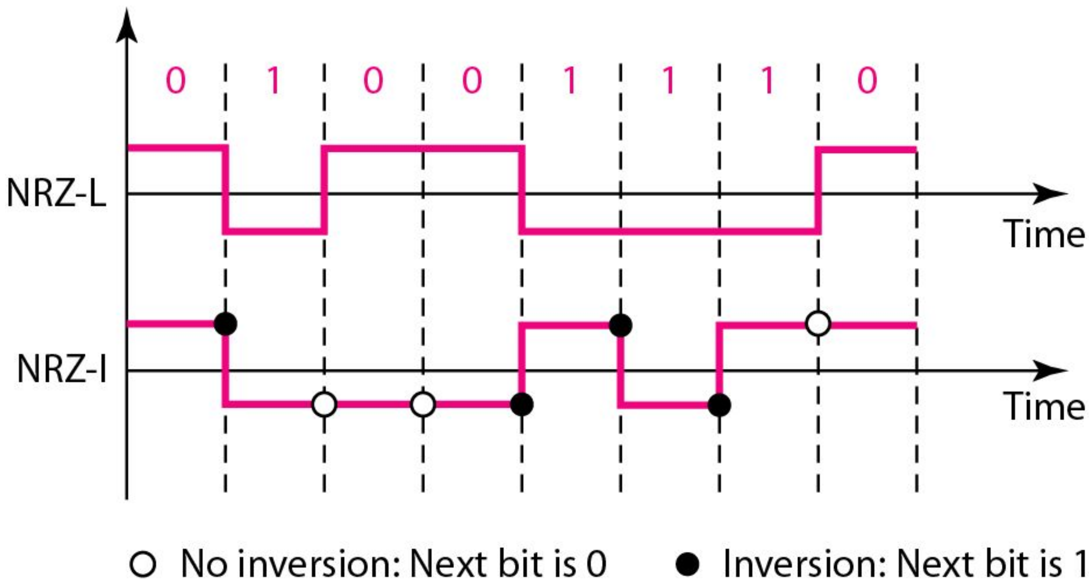
Normalized power

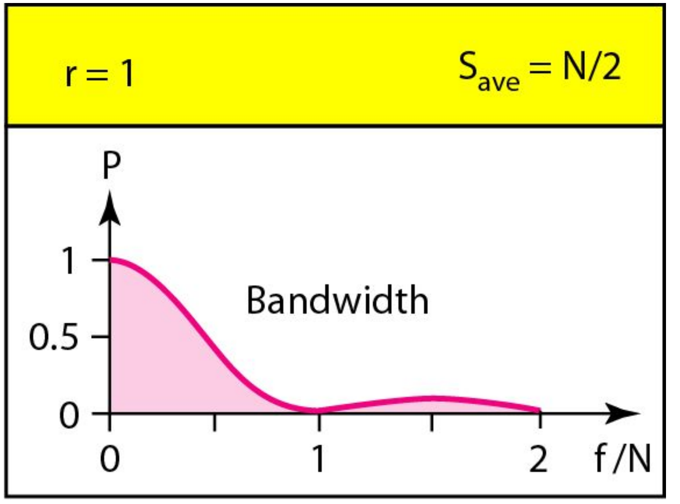
- In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.
- **NRZ** **(Non-Return-to-Zero):** unipolar scheme was designed as a non-return-to zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. **It is called NRZ because the signal does not return to zero at the middle of the bit**.

2. Polar - NRZ

- The voltages are on both sides of the time axis.
- Polar NRZ scheme can be implemented with two voltages.
- There are two versions:
  - NZR - Level (NRZ-L) - positive voltage for one symbol and negative for the other
  - **The voltage level for 0 can be positive** and the **voltage level for 1 can be negative**.
  - NRZ - Inversion (NRZ-I) - the change or lack of change in polarity determines the value of a symbol.
  - E.g. a “1” symbol inverts the polarity a “0” does not.

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*





Both schemes have an average signal rate of *N/2 Bd.*

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*

- If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed.
- The receiver might have difficulty discriminating the bit value.
- The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. long sequence of 0s can cause a problem in both schemes
- Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. a change in the polarity of the wire results in all 0s interpreted as 1s and all

Note

**In NRZ-L the level of the voltage determines the value of the bit.
In NRZ-I the inversion or the lack of inversion determines the value of the bit.**

Note

NRZ-L and NRZ-I both have an average signal rate of N/2 Bd.

Note

NRZ-L and NRZ-I both have a DC component problem and baseline wandering, it is worse for NRZ-L. Both have no self synchronization &no error detection. Both are relatively simple to implement.

Example 4.4

A system is using NRZ-I to transfer 1-Mbps data. What are the average signal rate and minimum bandwidth?

Solution

The average signal rate is S= c x N x R = 1/2 x N x 1 = 500 kbaud. The minimum bandwidth for this average baud rate is B_{min} ***= S = 500 kHz.***

Note c = 1/2 for the avg. case as worst case is 1 and best case is 0

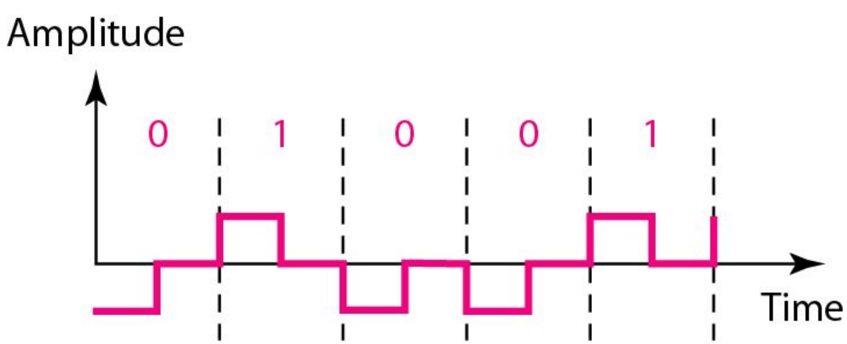
3. Polar - RZ

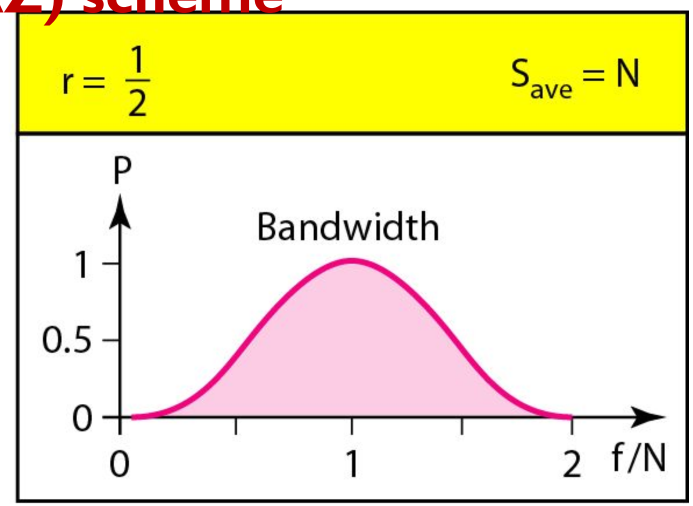
- The Return to Zero (RZ) scheme uses three voltage values. +, 0, -.
- Each symbol has a transition in the middle. Either from high to zero or from low to zero.
- In RZ, the signal changes not between bits but during the bit.
- The main disadvantage of RZ encoding is that it requires <u>two signal changes to encode a bit</u> and therefore **occupies greater bandwidth.**
- No DC components or baseline wandering.
- Self synchronization - transition indicates symbol value.
- More complex as it uses three voltage level. It has no error detection capability.

Figure 4.7 *Polar RZ scheme*

The main problem with 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**



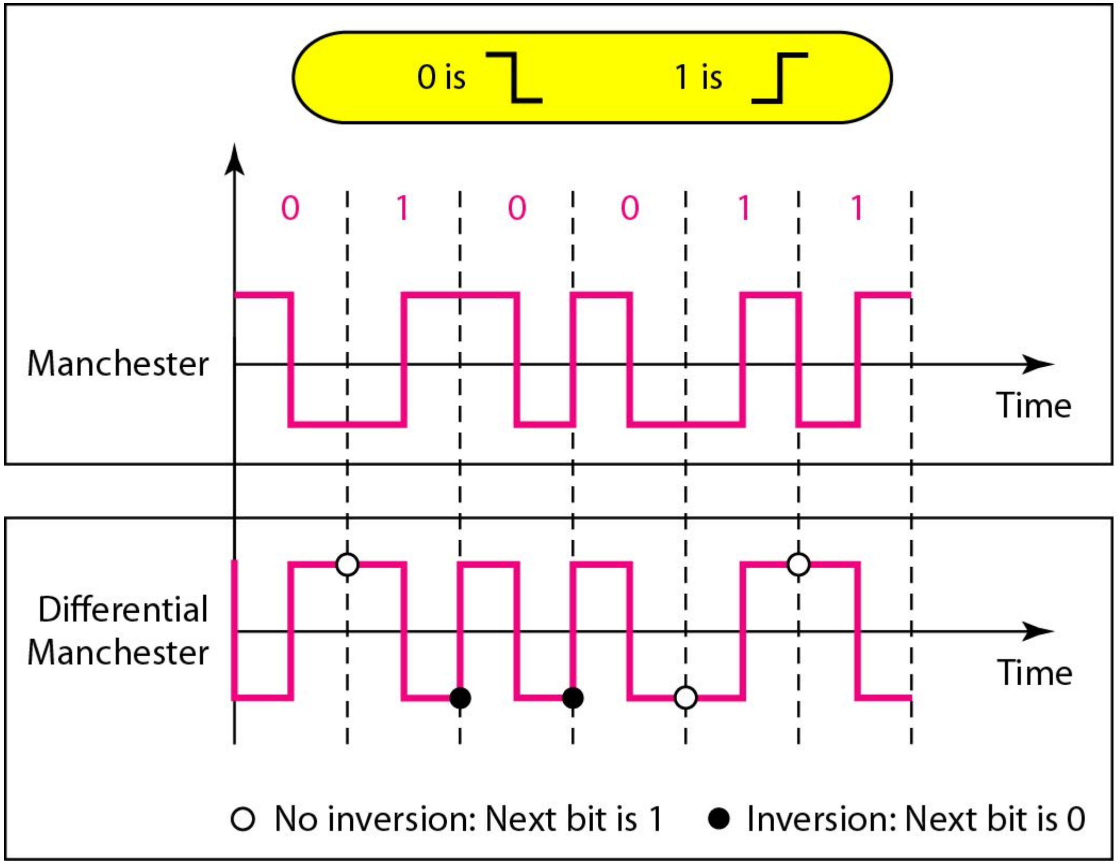


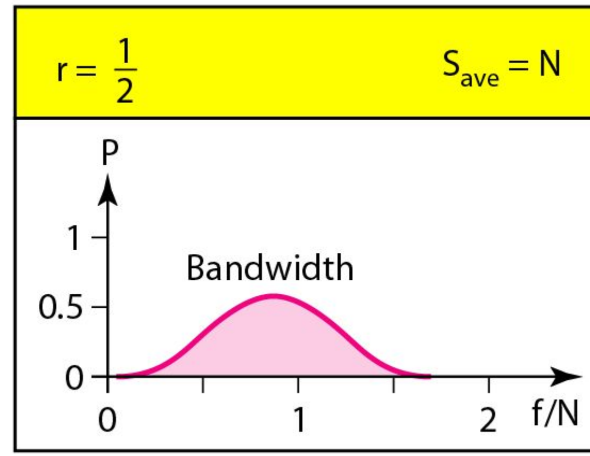
5. Polar - Biphase: Manchester and Differential Manchester

- Manchester coding consists of combining the NRZ-L and RZ schemes.
  - Every symbol has a level transition in the middle: from high to low or low to high. Uses only two voltage levels.
- Differential Manchester coding consists of combining the NRZ-I and RZ schemes.
  - Every symbol has a level transition in the middle. But the level at the beginning of the symbol is determined by the symbol value. One symbol causes a level change the other does not.

- 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.
- **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; if the next bit is 1, there is none.

Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*





Note

❖ The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I.

In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

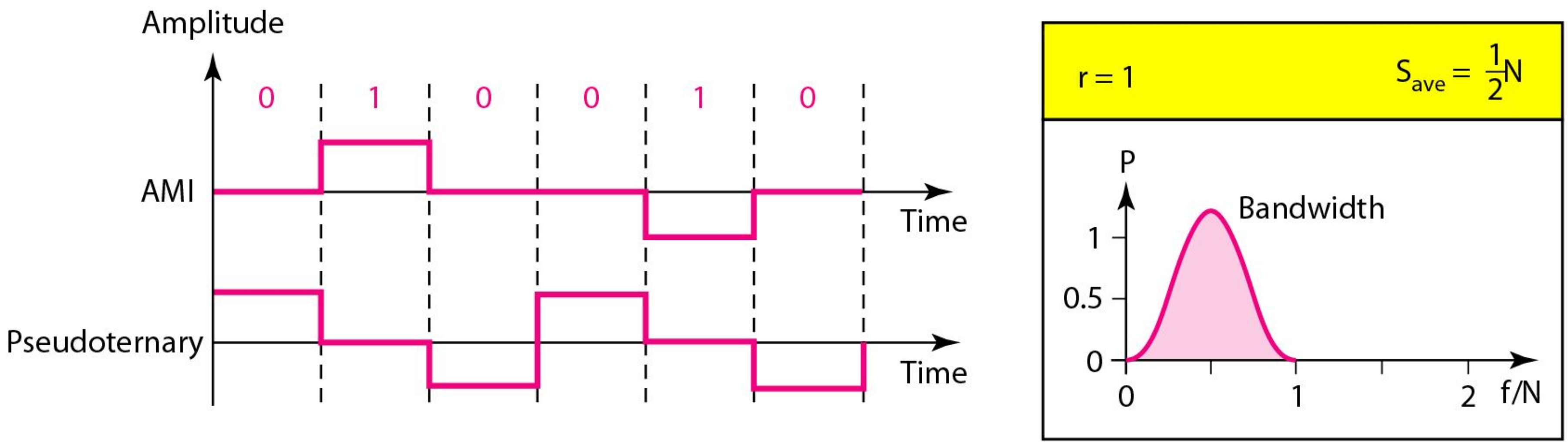
Note

The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.

4. Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at "0" and the other alternates between + & -.
- Bipolar **Alternate Mark Inversion (AMI)** - the "0" symbol is represented by zero voltage and the "1" symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Figure 4.9 *Bipolar schemes: AMI and pseudoternary*



- In bipolar encoding (sometimes called *multilevel binary),* there are three voltage levels*:* **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.
- Figure above shows two variations of bipolar encoding: **AMI** and **pseudoternary**.
- A common bipolar encoding scheme is called bipolar **alternate mark inversion** (AMI).
- In the term ***alternate mark inversion***, the word mark comes from telegraphy and 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 pseudoternary in which the **1 bit is encoded as a zero voltage** and **the 0 bit is encoded as alternating positive and negative voltages**.

Bipolar

- It is a better alternative to NRZ.
- Has no DC component or baseline wandering.
- Has no self synchronization because long runs of "0"s results in no signal transitions.
- No error detection.

4-2 ANALOG-TO-DIGITAL CONVERSION

A digital signal is superior to an analog signal because it is more robust to noise and can easily be recovered, corrected and amplified. For this reason, the tendency today is to change an analog signal to digital data. In this section we describe two techniques, pulse code modulation and delta modulation.

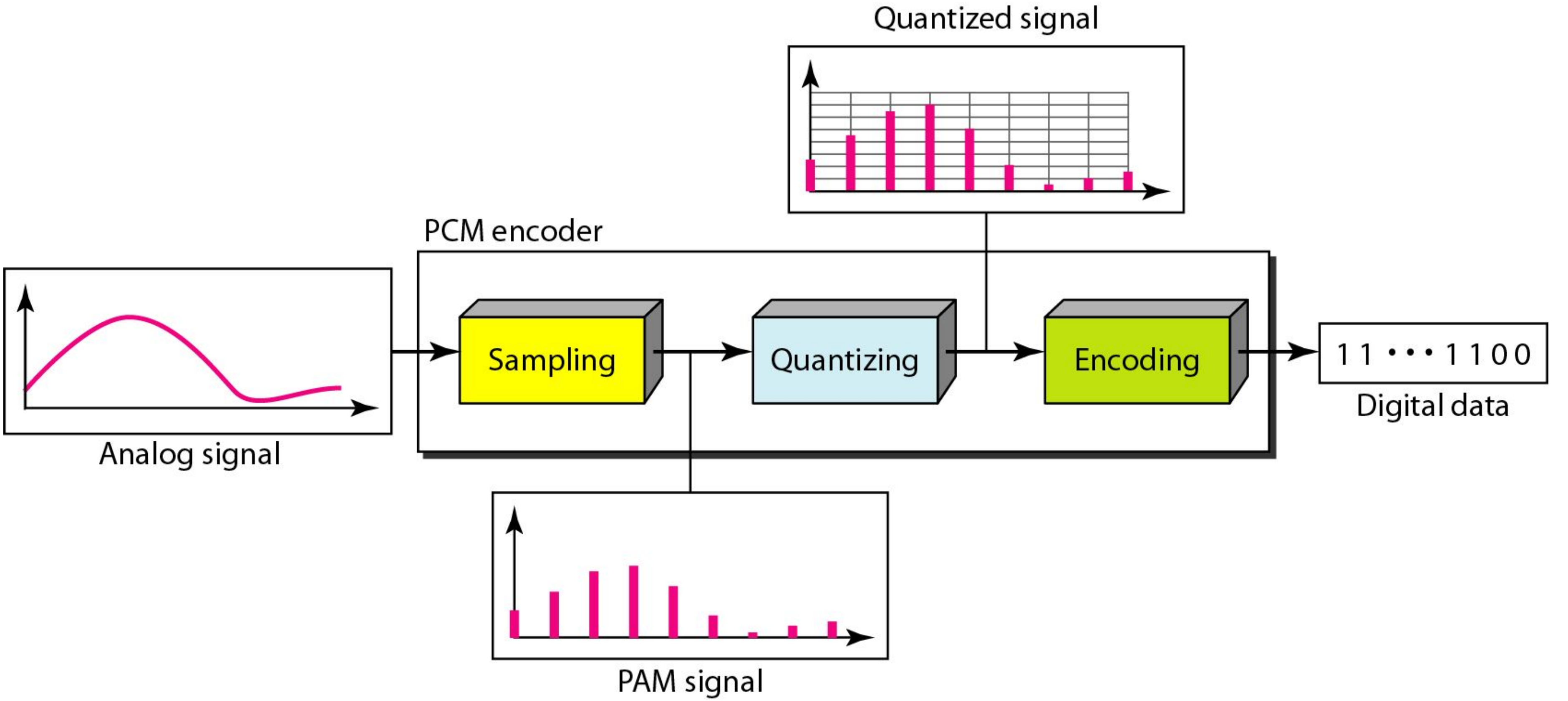
Topics discussed in this section:

- **Pulse Code Modulation (PCM)**
- **Delta Modulation (DM)**

PCM

- PCM consists of three steps to digitize an analog signal:
  1. Sampling
  2. Quantization
  3. Binary encoding
- Before we sample, we have to filter the signal to limit the maximum frequency of the signal as it **affects the sampling rate**.
- Filtering should ensure that we do not distort the signal, ie remove high frequency components that affect the signal shape.

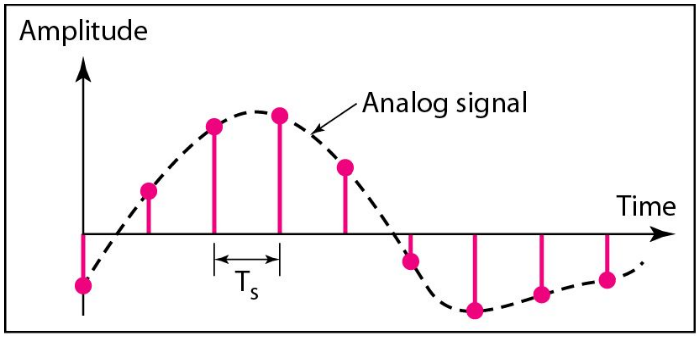
Figure 4.21 *Components of PCM encoder*



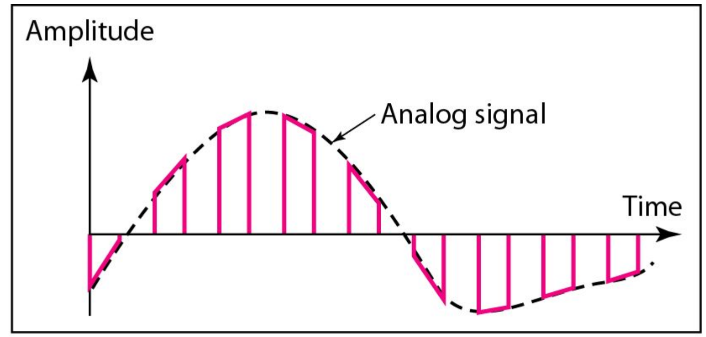
Sampling

- Analog signal is sampled every T_S secs.
- T_s is referred to as the sampling interval.
- $\mathbf{f_s = 1/T_s}$ is called the sampling rate or sampling frequency.
- There are 3 sampling methods:
  - Ideal - an impulse at each sampling instant
  - Natural - a pulse of short width with varying amplitude
  - Flattop - sample and hold, like natural but with single amplitude value
- The process is referred to as Pulse Amplitude Modulation **PAM** and the outcome is a signal with analog (non integer) values

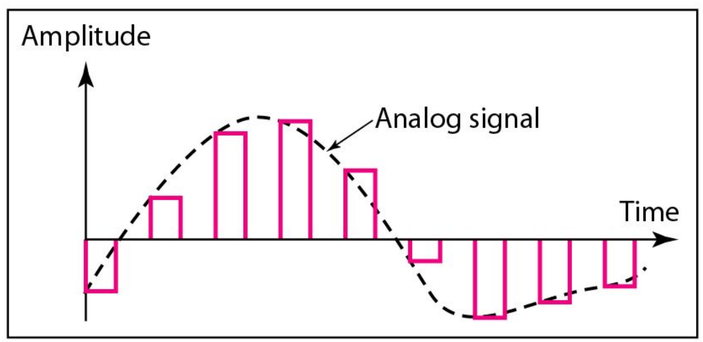
Figure 4.22 *Three different sampling methods for PCM*



a. Ideal sampling



b. Natural sampling



c. Flat-top sampling

- The analog signal is sampled every T_s, where T_s is the sample interval or period. The inverse of the sampling interval is called the ***sampling rate*** or ***sampling frequency*** and denoted by f_s where $f_s = 1/T_s$

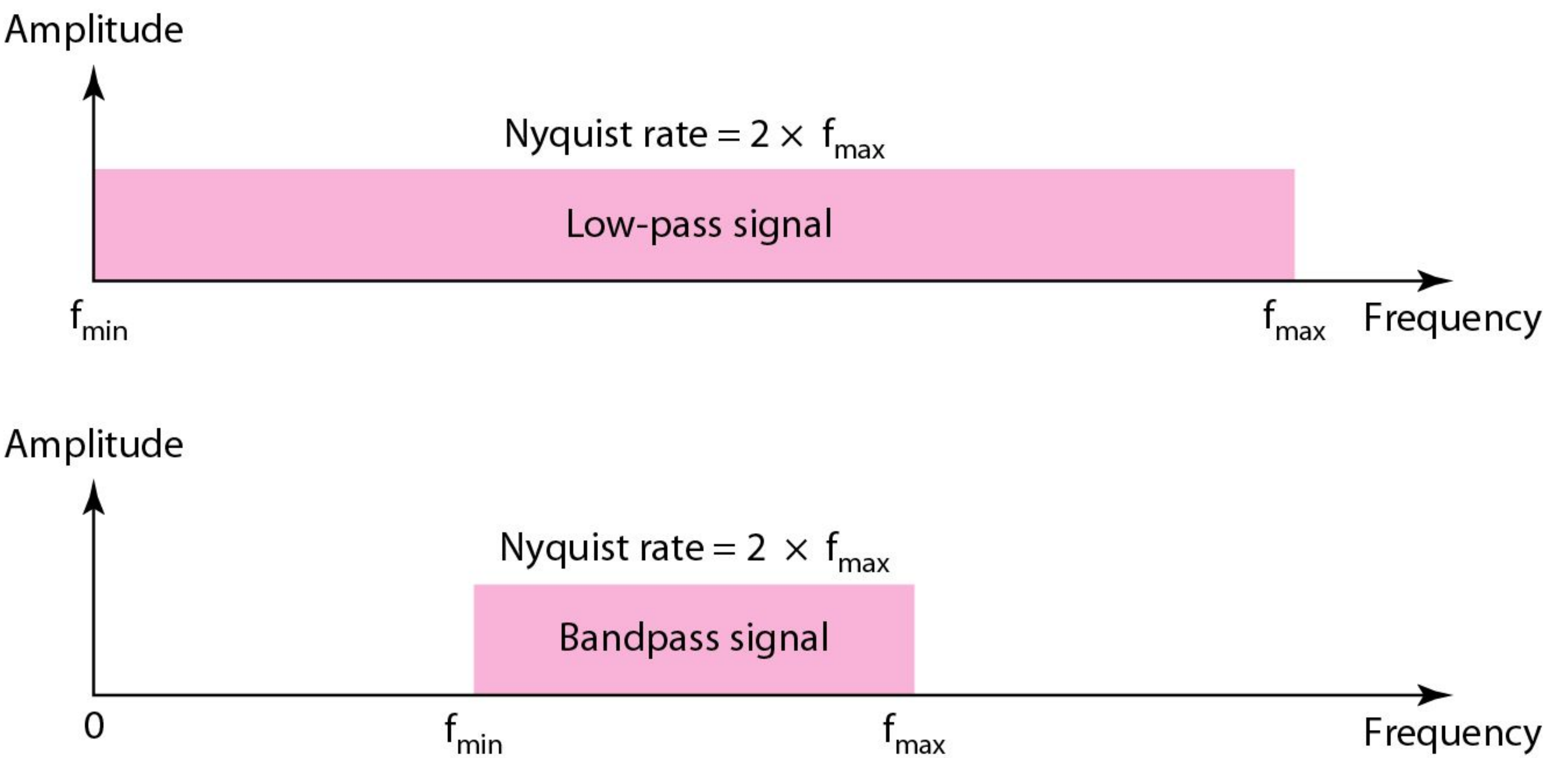
Figure 4.22 *Three different sampling methods for PCM*

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

Note

According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.

Figure 4.23 *Nyquist sampling rate for low-pass and bandpass signals*

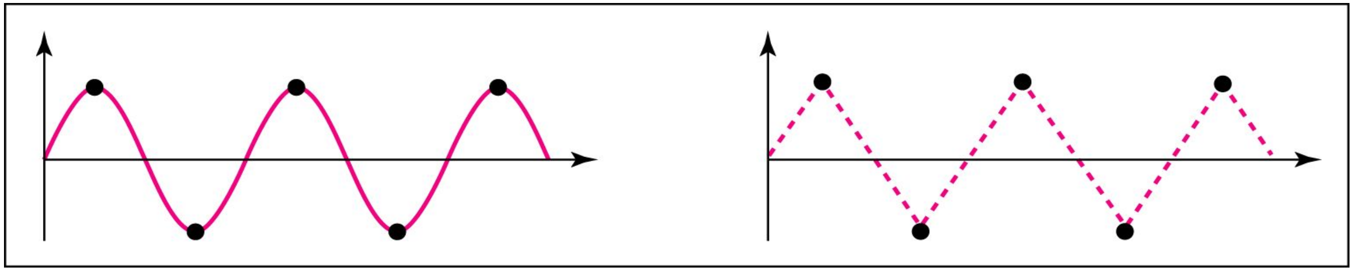


Example 4.6

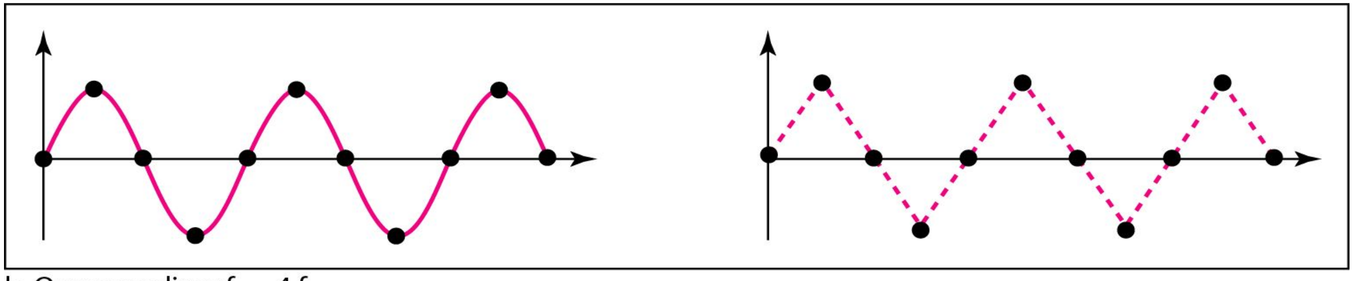
For an intuitive example of the Nyquist theorem, let us sample a simple sine wave at three sampling rates: $f_s = 4f$ (2 times the Nyquist rate), $f_s = 2f$ (Nyquist rate), and $f_s = f$ (one-half the Nyquist rate). Figure 4.24 shows the sampling and the subsequent recovery of the signal.

It can be seen that sampling at the Nyquist rate can create a good approximation of the original sine wave (part a). Oversampling in part b can also create the same approximation, but it is redundant and unnecessary. Sampling below the Nyquist rate (part c) does not produce a signal that looks like the original sine wave.

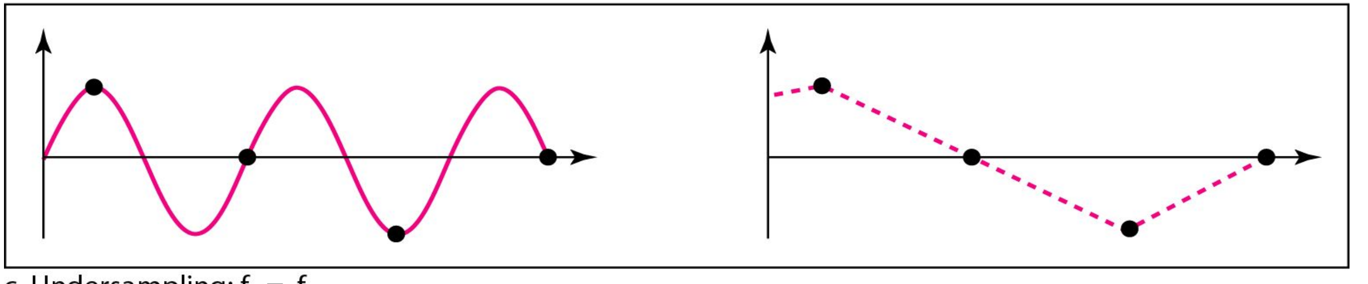
Figure 4.24 *Recovery of a sampled sine wave for different sampling rates*



a. Nyquist rate sampling: $f_s = 2f$



b. Oversampling: $f_s = 4f$



c. Undersampling: $f_s = f$

Example 4.9

Telephone companies digitize voice by assuming a maximum frequency of 4000 Hz. The sampling rate therefore is 8000 samples per second.

Example 4.10

A complex low-pass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

The bandwidth of a low-pass signal is between 0 and f, where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

Example 4.11

A complex bandpass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

We cannot find the minimum sampling rate in this case because we do not know where the bandwidth starts or ends. We do not know the maximum frequency in the signal.

Quantization

- **Sampling results** in a series of pulses of varying amplitude values ranging between two limits: **a min and a max.**
- The amplitude values are infinite between the two limits.
- We need to map the *infinite* amplitude values onto a finite set of known values.
- This is achieved by dividing the distance between min and max into L zones, each of height Δ.

$$\Delta = (\text{max} - \text{min})/L$$

PCM

o/p ⇒ Sampling → Quantization → Encoded

(0-L-1)
L=10

Δ<0

Eg: $V_{min} = -127$
$V_{max} = +127$

L = zones
↳ levels
L = 2, 4, 8, 16

$\Delta = 25$ $\Delta = (V_{max} - V_{min})/L$
$= (127+127)/10 = 25.4 \approx 25$

① Normalized PAM value (Actual value)

$\frac{24}{25} = 0.96$

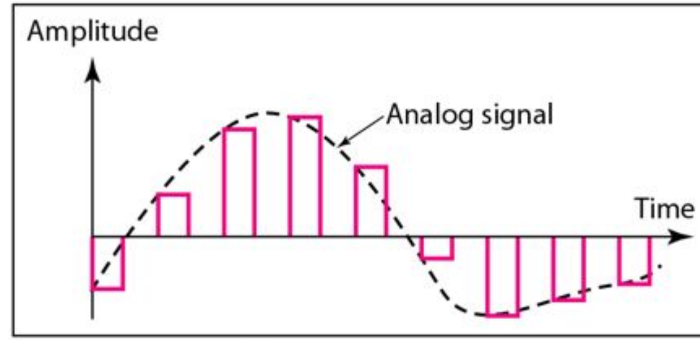
② Normalized Quantized value $0.96 \approx 1$

③ Normalized error +0.04 (difference)

④ Quantization code 6 0101 (Level value from 0 to 9)

⑤ Encoded word = 0110

$n_b = \log_2 L$
$= \log_2 10 \approx 4$



c. Flat-top sampling

Quantization Levels

- The midpoint of each zone is assigned a value from 0 to L-1 (resulting in L values)
- Each sample falling in a zone is then approximated to the value of the midpoint.
- The quantization code is assigned for each sample based on the quantization levels

Quantization Zones

- Assume we have a voltage signal with amplitutes V_{min}=-20V and V_{max}=+20V.
- We want to use L=8 quantization levels.
- Zone width Δ = (20 - -20)/8 = 5
- The 8 zones are: -20 to -15, -15 to -10, -10 to -5, -5 to 0, 0 to +5, +5 to +10, +10 to +15, +15 to +20
- The midpoints are: -17.5, -12.5, -7.5, -2.5, 2.5, 7.5, 12.5, 17.5

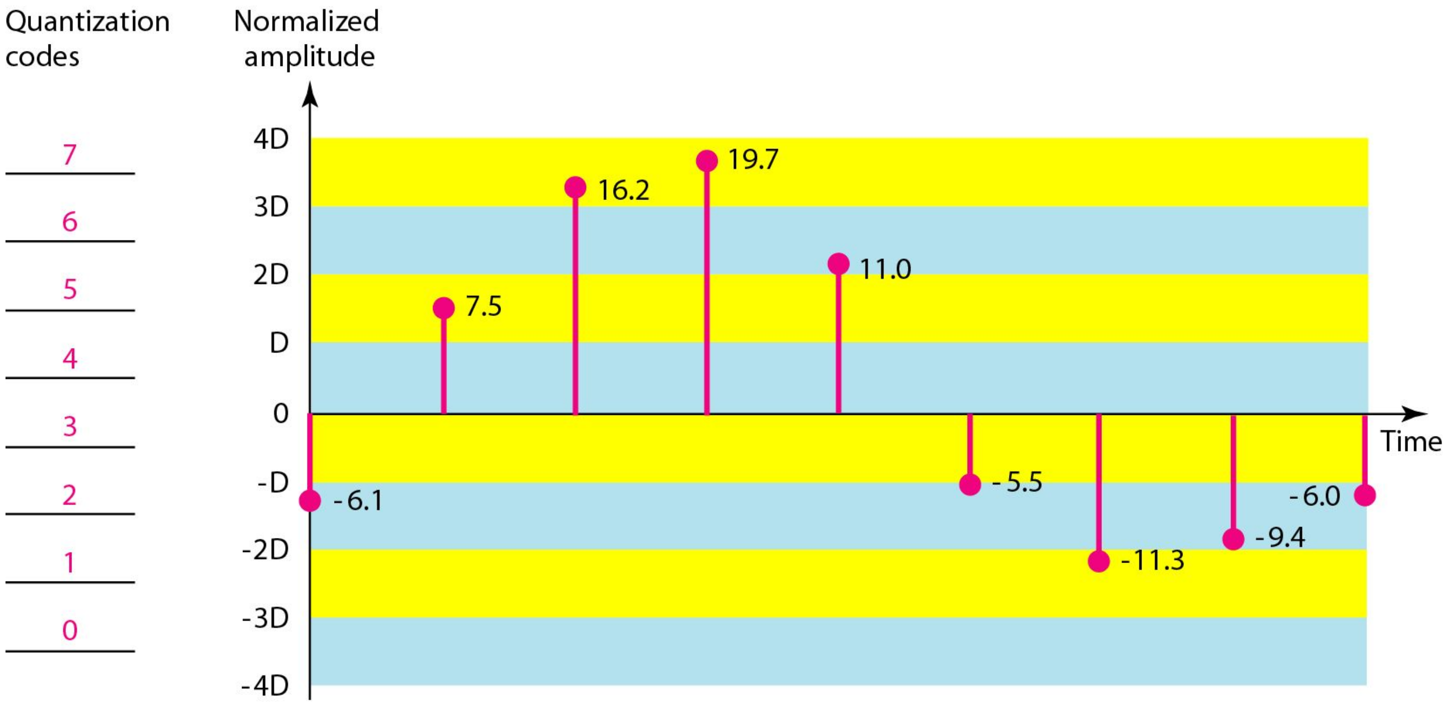
Assigning Codes to Zones

- Each zone is then assigned a binary code.
- The number of bits required to encode the zones, or the number of bits per sample as it is commonly referred to, is obtained as follows:

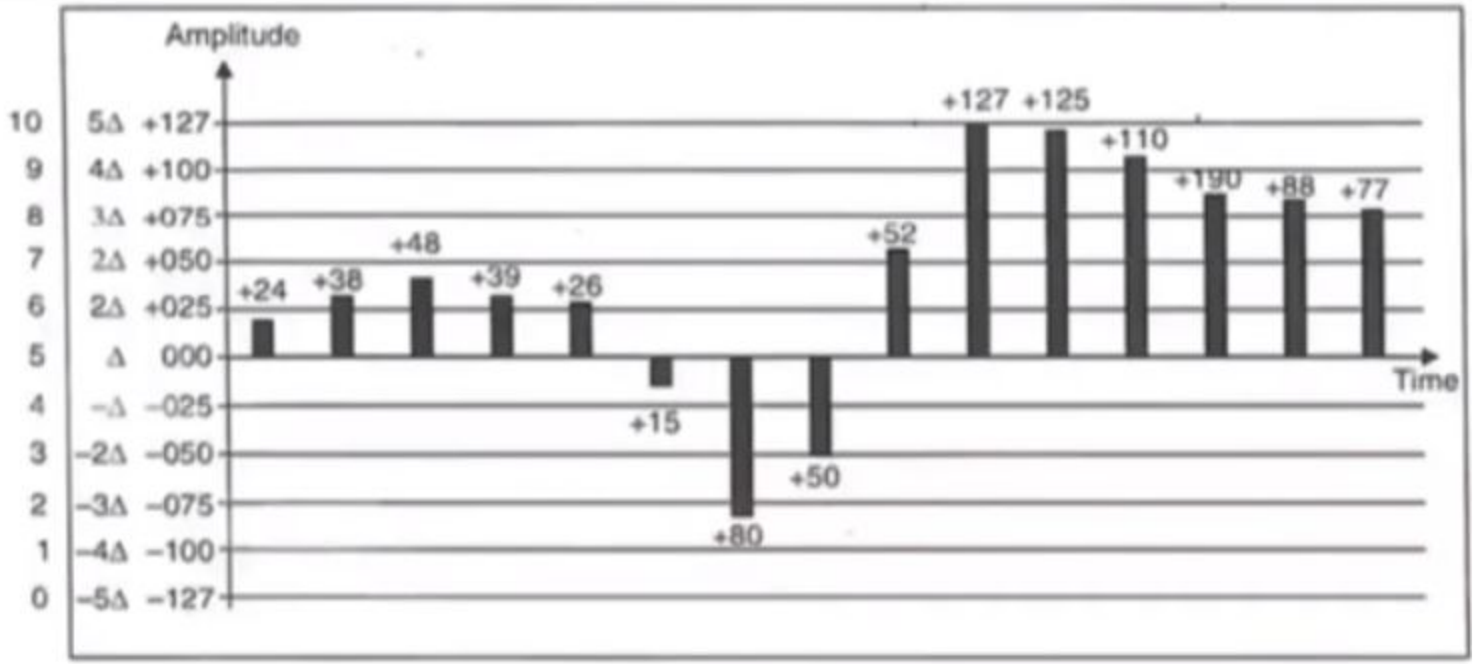
$$n_b = \log_2 L$$

- Given our example, $n_b = 3$
- The 8 zone (or level) codes are therefore: 000, 001, 010, 011, 100, 101, 110, and 111
- Assigning codes to zones:
  - 000 will refer to zone -20 to -15
  - 001 to zone -15 to -10, etc.

Figure 4.26 *Quantization and encoding of a sampled signal*



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



Row 1	Normalized PAM values (actual values)	0.96	1.52	1.92	1.56	1.04	−0.6	−3.2	−2	2.08	5.08	5	4.4	3.6	3.52	3.08
Row 2	Normalized quantized values	1	1.5	2	1.5	1	−0.5	−3.5	−2	2	5	5	4.5	3.5	3.5	3
Row 3	Normalized error	+0.04	−0.02	+0.08	−0.06	−0.04	+0.1	−0.3	0	−0.08	−0.08	0	+0.1	−0.1	−0.02	−0.08
Row 4	Quantization code	5	6	6	6	5	4	1	3	7	10	10	9	8	8	8
Row 5	Encoded word	0101	0110	0110	0110	0101	0100	0001	0011	0111	1010	1010	1001	1000	1000	1000

PCM

o/p ⇒ Sampling → Quantization → Encoded

(0-L-1)

L=10

$\Delta < 0$

Eg: $V_{min} = -127$

$V_{max} = +127$

L = zones

↳ levels

L = 2, 4, 8, 10

$\Delta = 25$ $\quad \Delta = (V_{max} - V_{min})/L$

$= (127 + 127)/10 = 25.4 \approx 25$

① Normalized PAM value (Actual value)

$\frac{24}{25} = 0.96$

② Normalized Quantized value $\quad 0.96 \approx 1$

③ Normalized error $\quad +0.04$ (difference)

④ Quantization code $\quad$ 6 (level value from 0 to 9)

0101

⑤ Encoded word = 0110

$n_b = \log_2 L$

$= \log_2 10 \approx 4$

Quantization Error

- When a signal is quantized, we introduce an error - the coded signal is an approximation of the actual amplitude value.
- The difference between actual and coded value (midpoint) is referred to as the quantization error.
- The more zones, the smaller Δ which results in smaller errors.
- BUT, the more zones the more bits required to encode the samples -> higher bit rate

Quantization Error and SN_QR

- Signals with lower amplitude values will suffer more from quantization error as the error range: **$\Delta/2$, is fixed for all signal levels**.
- Non linear quantization is used to alleviate this problem. Goal is to keep SN_QR fixed for all sample values.
- Two approaches:
  - The quantization levels follow a logarithmic curve. Smaller Δ's at lower amplitudes and larger Δ's at higher amplitudes.
  - Companding: The sample values are compressed at the sender into logarithmic zones, and then expanded at the receiver. **The zones are fixed in height**.

Bit rate and bandwidth requirements of PCM

- The bit rate of a PCM signal can be calculated form the **number of bits per sample x the sampling rate**

$$\textbf{Bit rate} = n_b \times f_s$$

- The bandwidth required to transmit this signal depends on the **type of line encoding used.** Refer to previous section for discussion and formulas.
- A digitized signal will always need more bandwidth than the original analog signal. Price we pay for robustness and other features of digital transmission.

Example 4.14

We want to digitize the human voice. What is the bit rate, assuming 8 bits per sample?

Solution

The human voice normally contains frequencies from 0 to 4000 Hz. So the sampling rate and bit rate are calculated as follows:

$$\text{Sampling rate} = 4000 \times 2 = 8000 \text{ samples/s}$$
$$\text{Bit rate} = 8000 \times 8 = 64{,}000 \text{ bps} = 64 \text{ kbps}$$

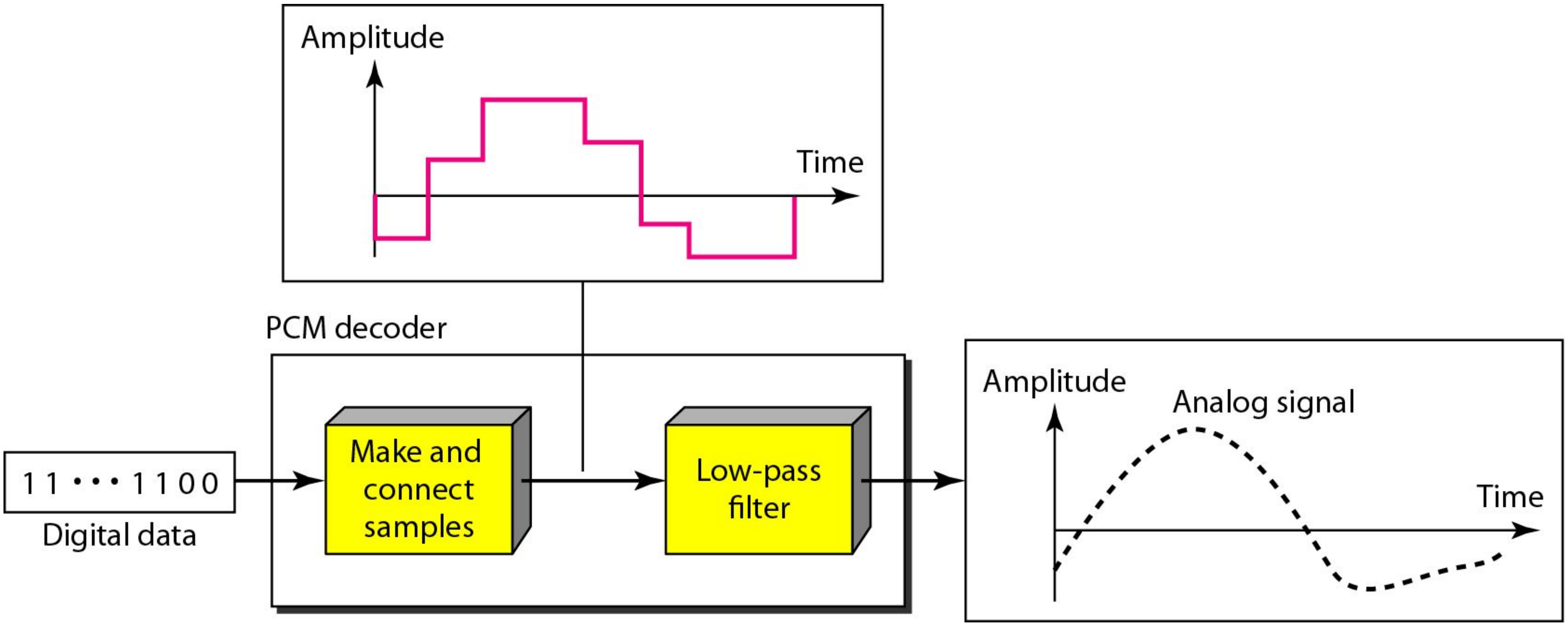
PCM Decoder

- To recover an analog signal from a digitized signal we follow the following steps:
  - We use a hold circuit that holds the amplitude value of a pulse till the next pulse arrives.
  - We pass this signal through a low pass filter with a cutoff frequency that is equal to the highest frequency in the pre-sampled signal.
- The higher the value of L, the less distorted a signal is recovered.

PCM Decoder

- The recovery of the original signal requires the **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 4.27 *Components of a PCM decoder*



Example 4.15

We have a low-pass analog signal of 4 kHz. If we send the analog signal, we need a channel with a minimum bandwidth of 4 kHz. If we digitize the signal and send 8 bits per sample, we need a channel with a minimum bandwidth of 8×4 kHz = 32 kHz.

Delta Modulation

- This scheme sends only the **difference between pulses**, if the pulse at time t_{n+1} is higher in amplitude value than the pulse at time t_n, then a single bit, say a "1", is used to indicate the positive value.
- If the pulse is lower in value, resulting in a negative value, a "0" is used.
- This scheme works well for small changes in signal values between samples.
- **If changes in amplitude are large, this will result in large errors.**

Figure 4.28 *The process of delta modulation*

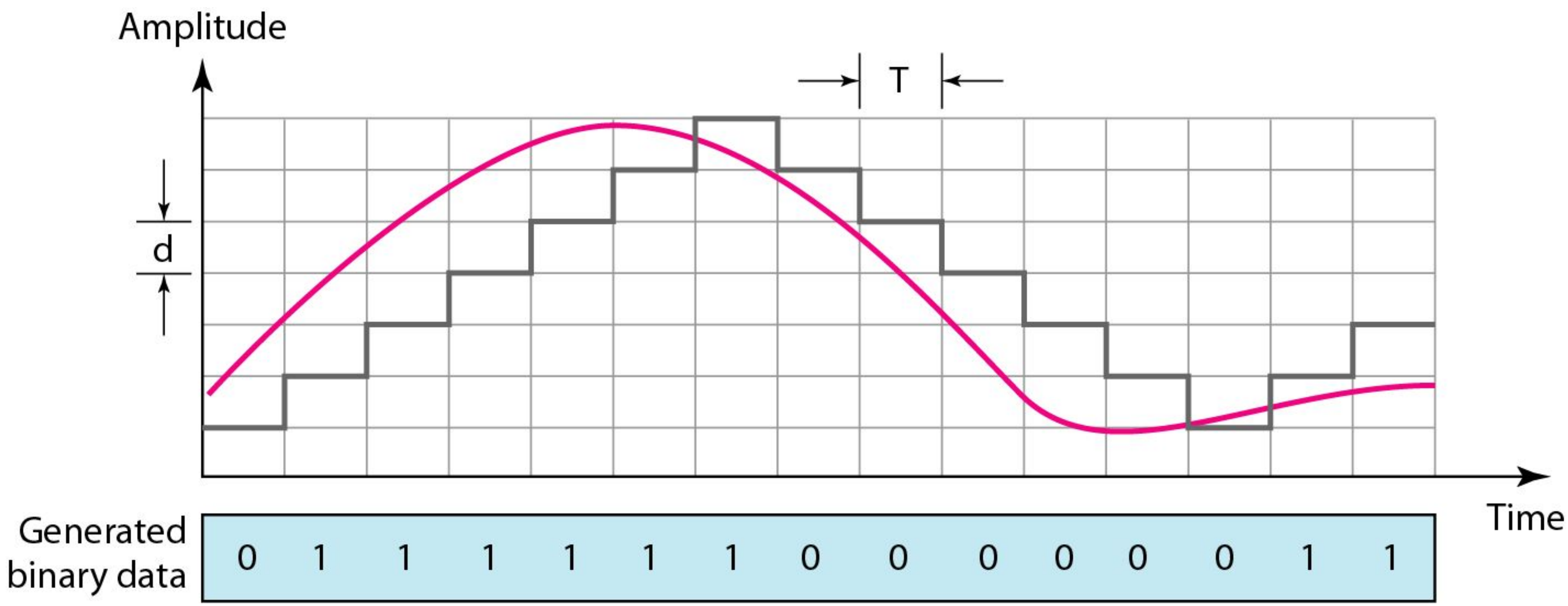


Figure 4.29 *Delta modulation components*

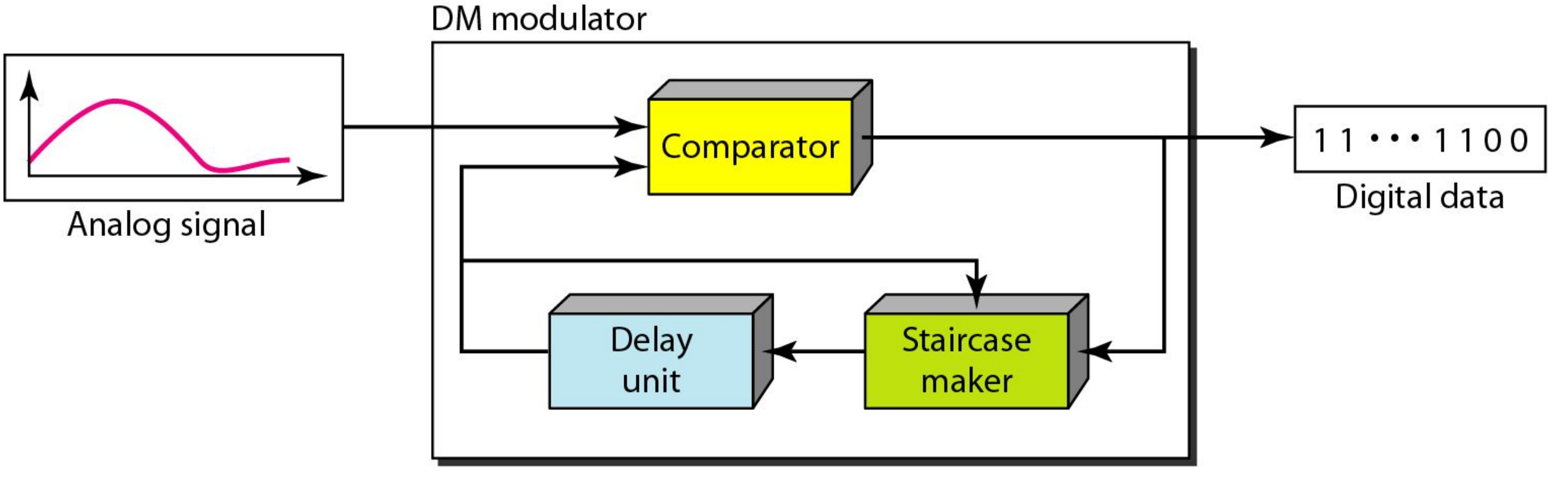
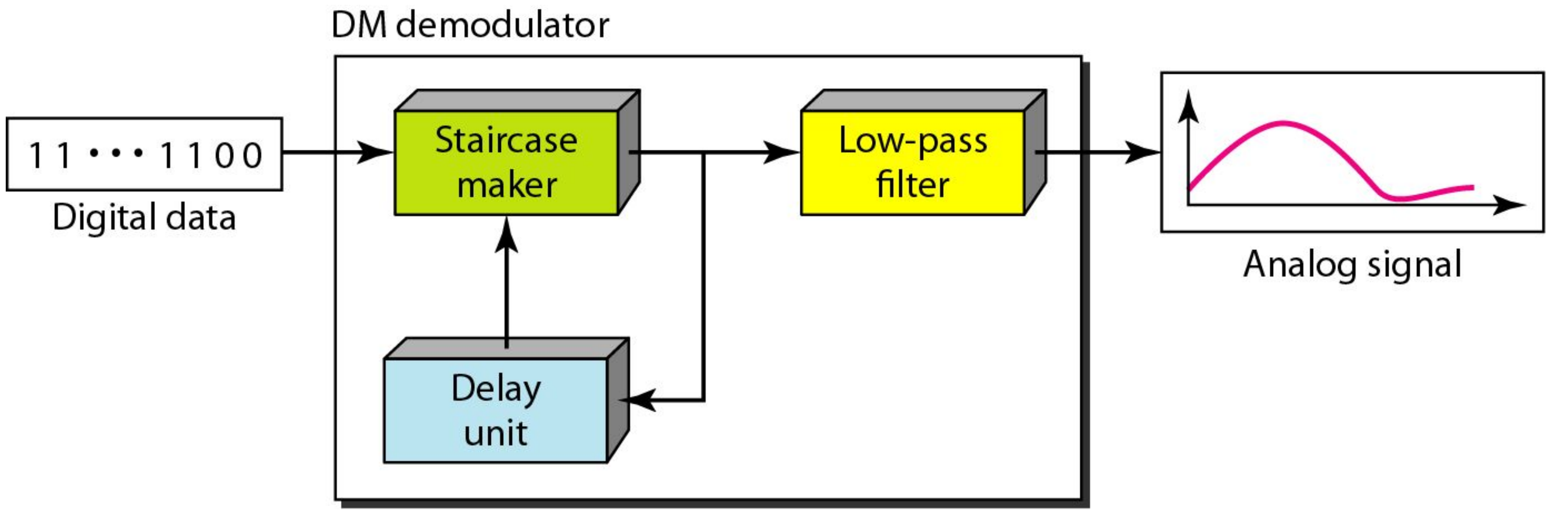


Figure 4.30 *Delta demodulation components*



Delta PCM (DPCM)

- Instead of using one bit to indicate positive and negative differences, we can use more bits -> quantization of the difference.
- Each bit code is used to represent the value of the difference.
- The more bits the more levels -> the higher the accuracy.

4-3 TRANSMISSION MODES

The 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. In serial mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.

Topics discussed in this section:

- **Parallel Transmission**
- **Serial Transmission**

Figure 4.31 *Data transmission and modes*

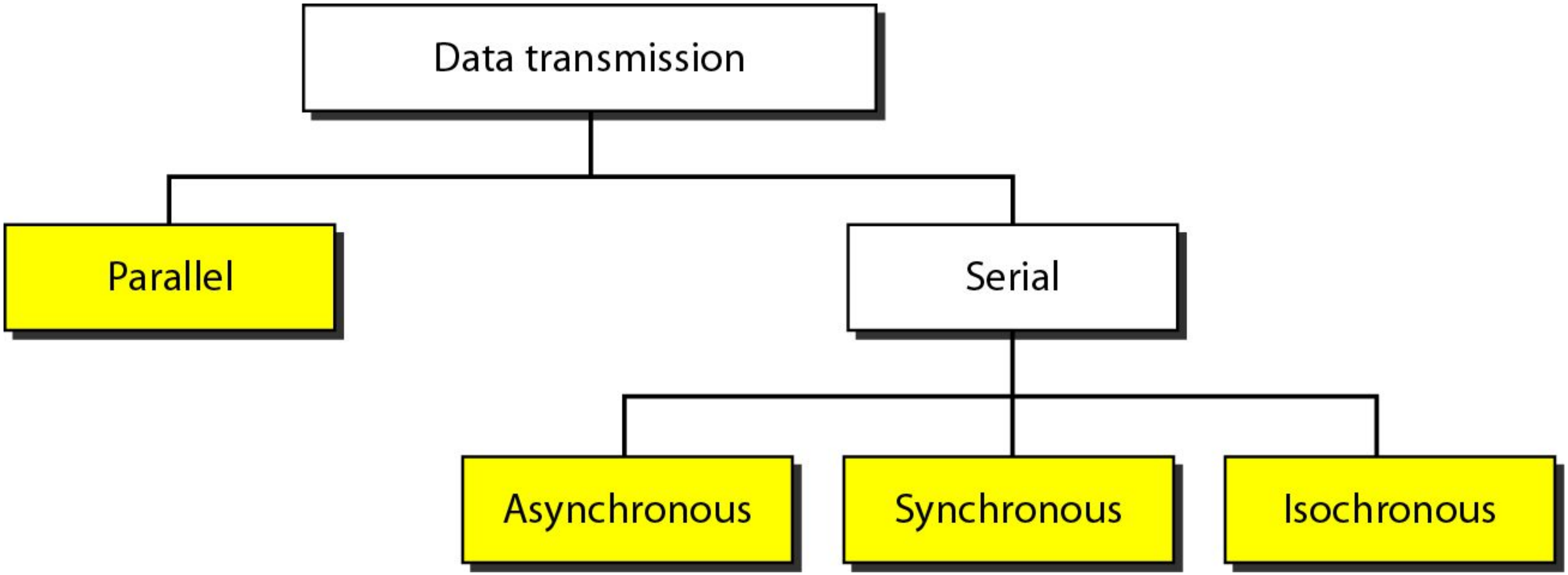
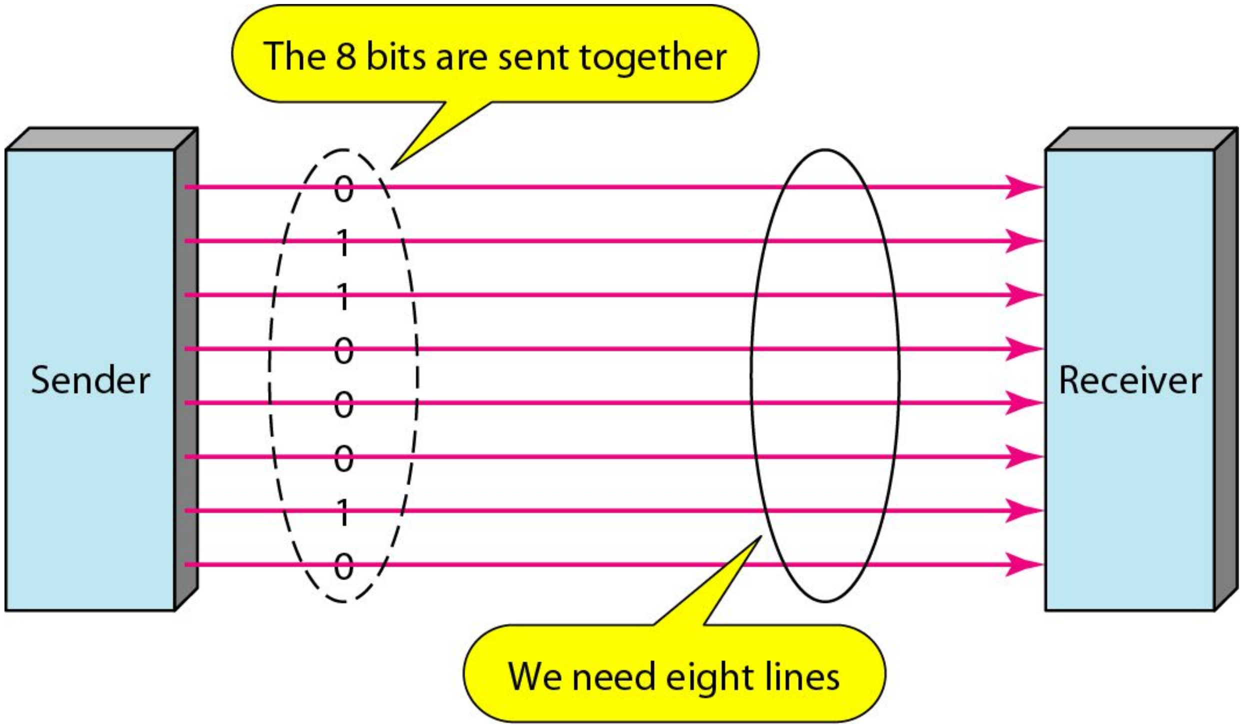


Figure 4.32 *Parallel transmission*

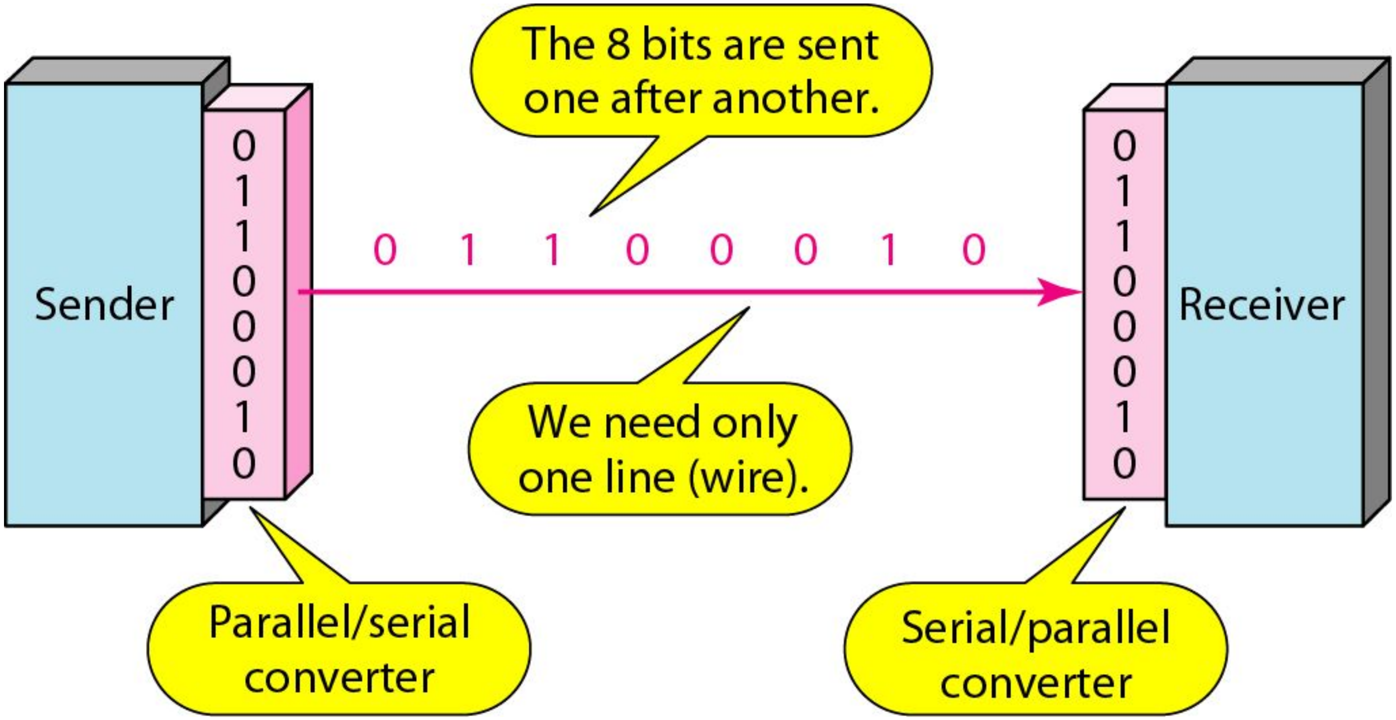


- Binary data, consisting of 1s and 0s, may be organized into **groups of *n bits*** *each.*
- The mechanism for parallel transmission is a conceptually simple one: **Use *n wires* to send *n bits at one time*.** That way each bit has its own wire, and all n bits *of one* group can be transmitted with each clock tick from one device to another .

- The advantage of parallel transmission is **speed**.
- But there is a significant disadvantage: **cost**. Parallel transmission requires

- ***n-communication* lines** (wires) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

Figure 4.33 *Serial transmission*



- In **serial transmission** one bit follows another, so we need only one communication channel rather than '*n*' to transmit data between two communicating devices.
- The advantage of serial over parallel transmission is that with only one communication channel, serial transmission **reduces the cost** of transmission over parallel.

- 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).
- Serial transmission occurs in one of three ways: **asynchronous**, **synchronous**, and **isochronous**.

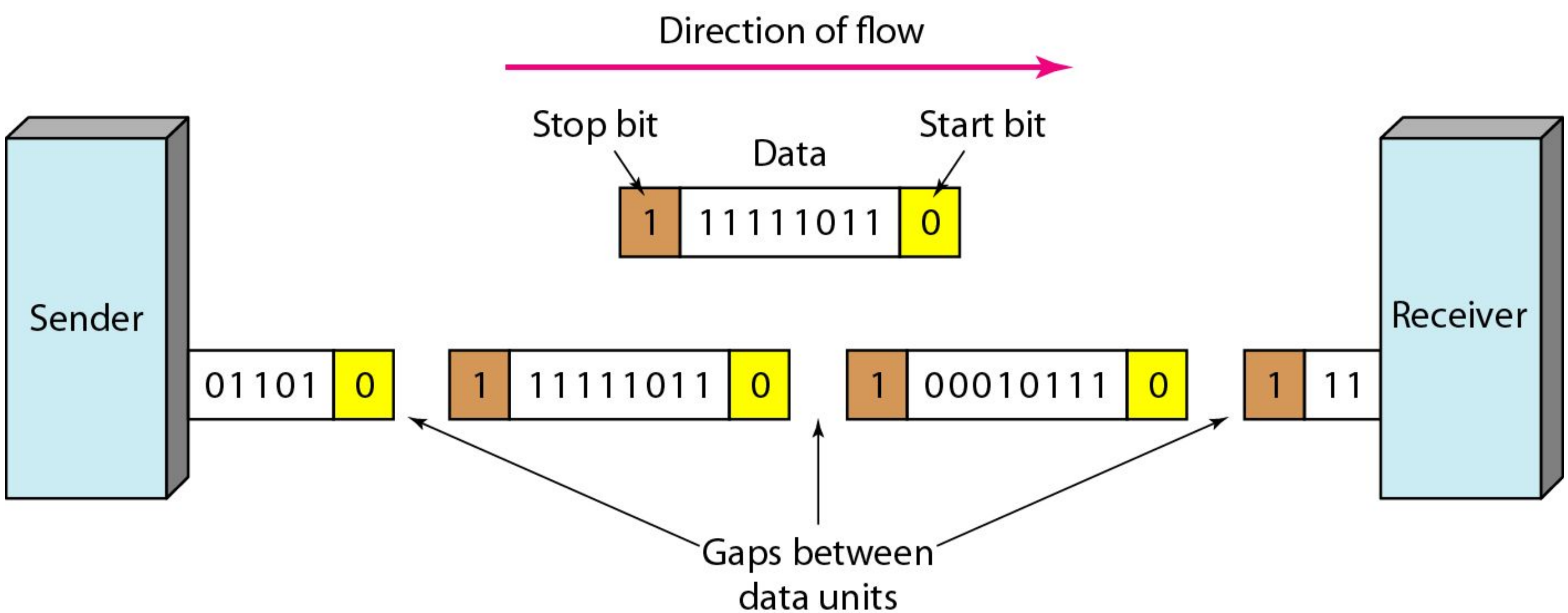
Note

In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte. There may be a gap between each byte.

Note

Asynchronous here means "asynchronous at the byte level," but the bits are still synchronized; their durations are the same.

Figure 4.34 *Asynchronous transmission*

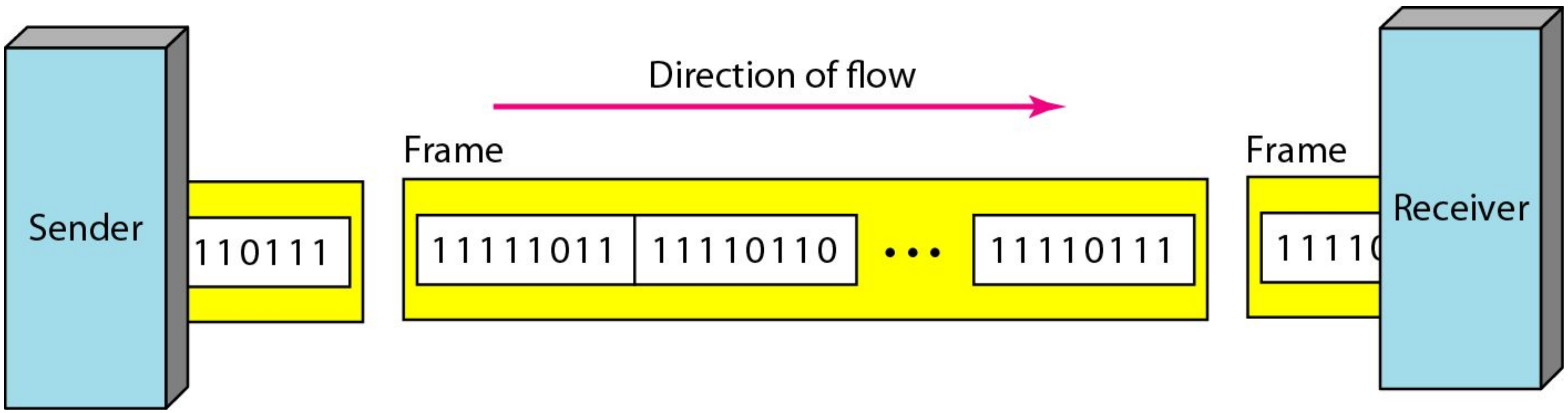


- In **asynchronous transmission**, information is received and translated by agreed upon patterns. As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent.
- In **asynchronous transmission**, we send **1 start bit** (0) at the beginning and **1 or more stop bits** (1s) at the end of each byte. There may be a **gap** between each byte.

Note

In synchronous transmission, 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 bits are usually sent as bytes and many bytes are grouped in a frame. A frame is identified with a start and an end byte.

Figure 4.35 *Synchronous transmission*



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

Isochronous

- In isochronous transmission we cannot have uneven gaps between frames.
- Transmission of bits is fixed with equal gaps.
- In **real-time** audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails, the entire stream of bits must be **synchronized**. The isochronous transmission guarantees that the **data arrive at a fixed rate**.

5-1 DIGITAL-TO-ANALOG CONVERSION

Digital-to-analog conversion is the process of changing one of the characteristics of an analog signal based on the information in digital data.

Topics discussed in this section:

- **Aspects of Digital-to-Analog Conversion**
- **Amplitude Shift Keying**
- **Frequency Shift Keying**
- **Phase Shift Keying**
- **Quadrature Amplitude Modulation**

Digital to Analog Conversion

- Digital data needs to be carried on an analog signal.
- A **carrier** signal (frequency f_c) performs the function of transporting the digital data in an analog waveform.
- The analog carrier signal is manipulated to uniquely identify the digital data being carried.

Figure 5.1 *Digital-to-analog conversion*

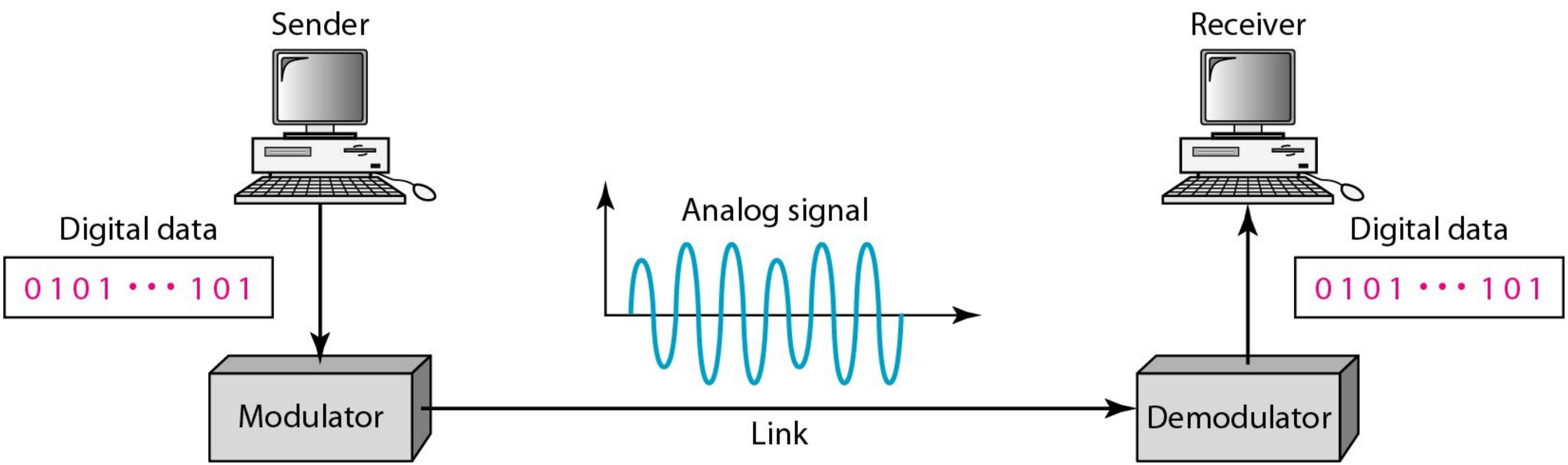
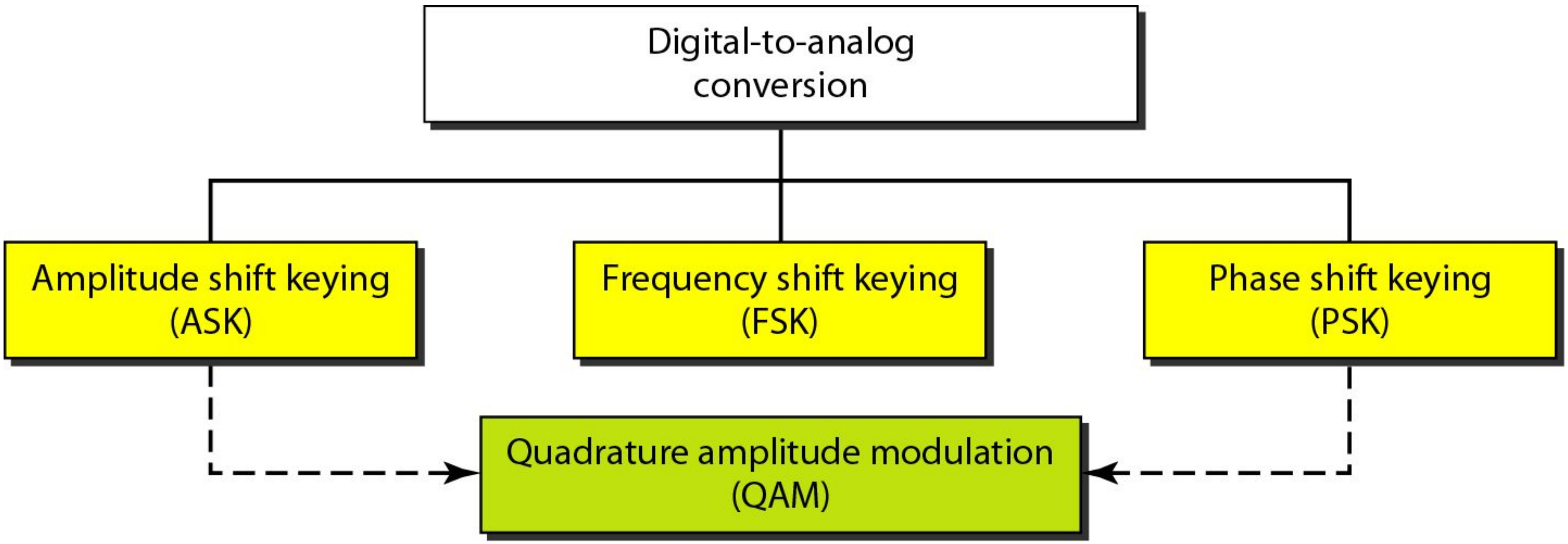


Figure 5.2 *Types of digital-to-analog conversion*



Note

Bit rate, N, is the number of bits per second (bps). Baud rate is the number of signal elements per second (bauds). In the analog transmission of digital data, the signal or baud rate is less than or equal to the bit rate.

S=Nx1/r bauds

Where r is the number of data bits per signal element.

Example 5.1

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.

Solution

In this case, r = 4, S = 1000, and N is unknown. We can find the value of N from

$$S = N \times \frac{1}{r} \qquad \text{or} \qquad N = S \times r = 1000 \times 4 = 4000 \text{ bps}$$

Example 5.2

An analog signal has a bit rate of 8000 bps and a baud rate of 1000 baud. How many data elements are carried by each signal element? How many signal elements do we need?

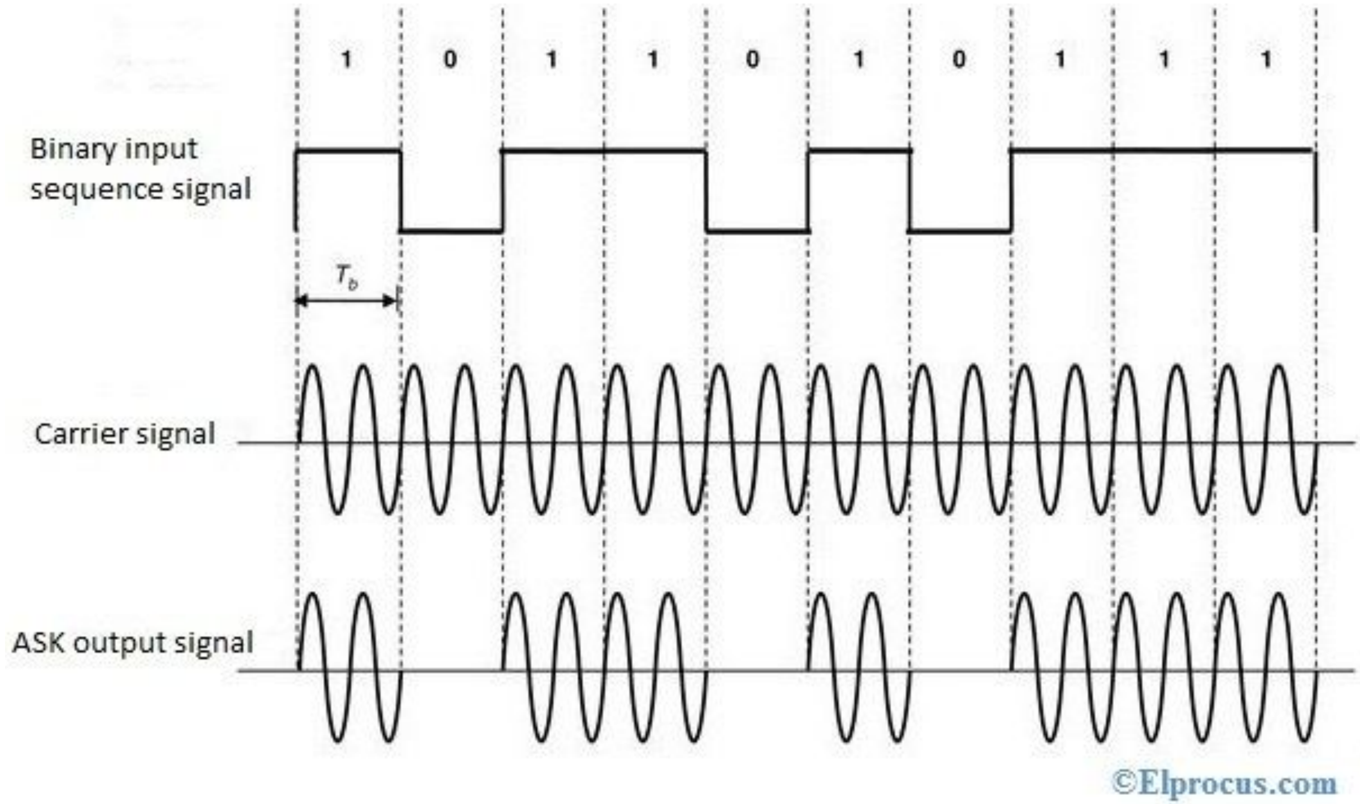
Solution

In this example, S = 1000, N = 8000, and r and L are unknown. We find first the value of r and then the value of L.

$$S = N \times \frac{1}{r} \rightarrow r = \frac{N}{S} = \frac{8000}{1000} = 8 \text{ bits/baud}$$

$$r = \log_2 L \rightarrow L = 2^r = 2^8 = 256$$

1. Amplitude Shift Keying (ASK)



Amplitude Shift Keying (ASK)

- ASK is implemented by changing the amplitude of a carrier signal to reflect amplitude levels in the digital signal.
- For example: a digital "1" could not affect the signal, whereas a digital "0" would, by making it zero.
- The line encoding will determine the values of the analog waveform to reflect the digital data being carried.

Bandwidth of ASK

- The bandwidth B of ASK is proportional to the signal rate S.

$$\mathbf{B = (1+d)S}$$

- "d" is due to modulation and filtering, lies between 0 and 1.

Figure 5.3 *Binary amplitude shift keying*

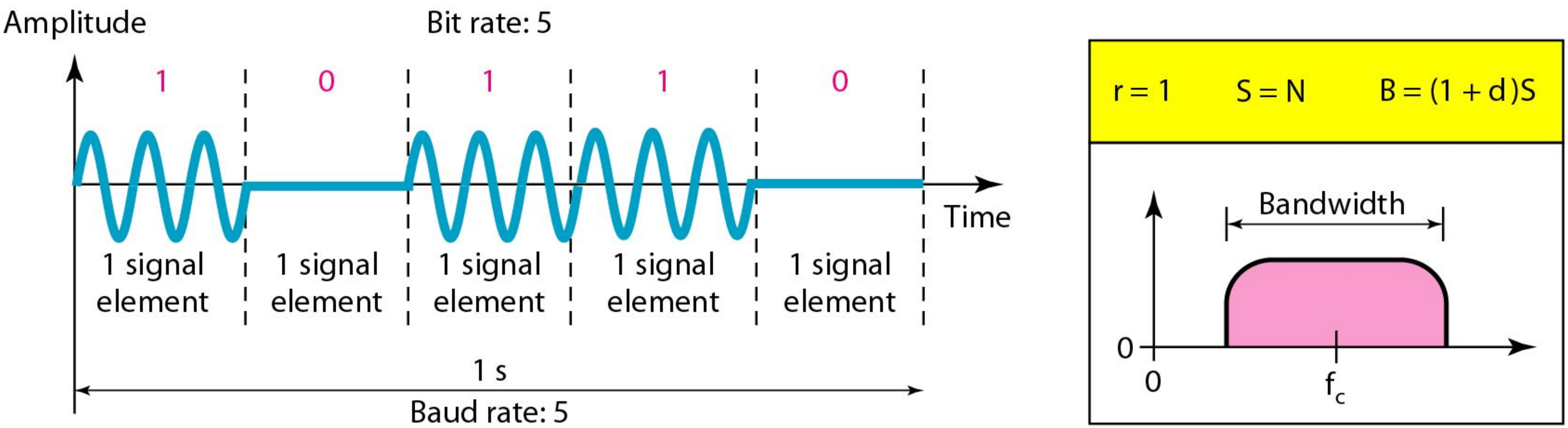
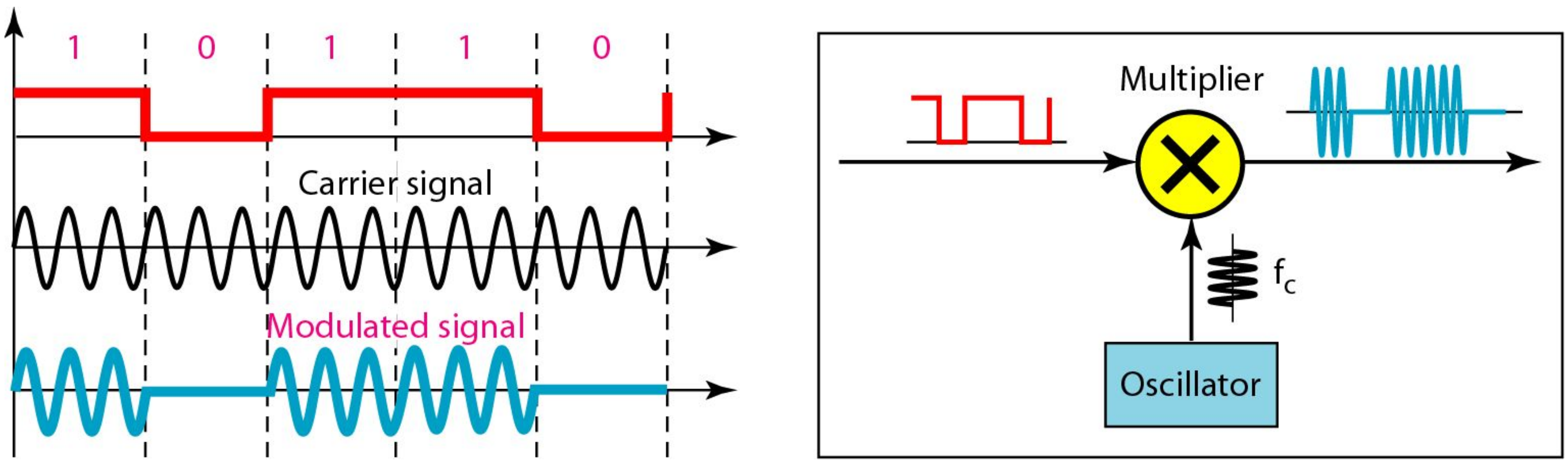


Figure 5.4 *Implementation of binary ASK*



Example 5.3

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What are the carrier frequency and the bit rate if we modulated our data by using ASK with d = 1?

Solution

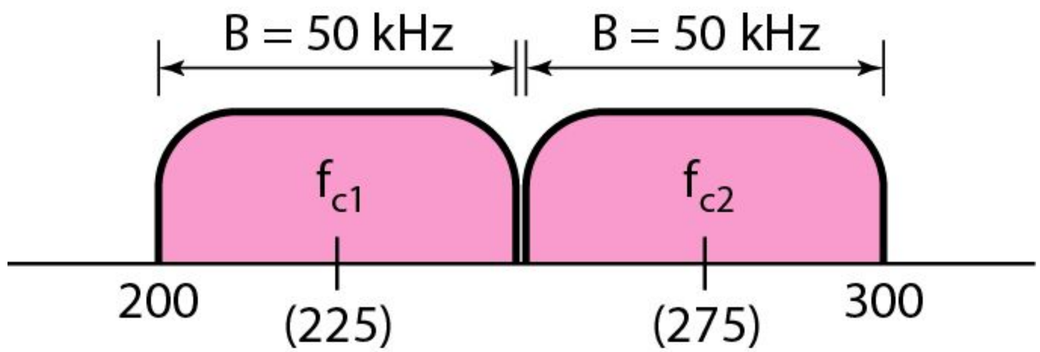
The middle of the bandwidth is located at 250 kHz. This means that our carrier frequency can be at f_c = 250 kHz. We can use the formula for bandwidth to find the bit rate (with d = 1 and r = 1).

$$B = (1 + d) \times S = 2 \times N \times \frac{1}{r} = 2 \times N = 100 \text{ kHz} \quad \Rightarrow \quad N = 50 \text{ kbps}$$

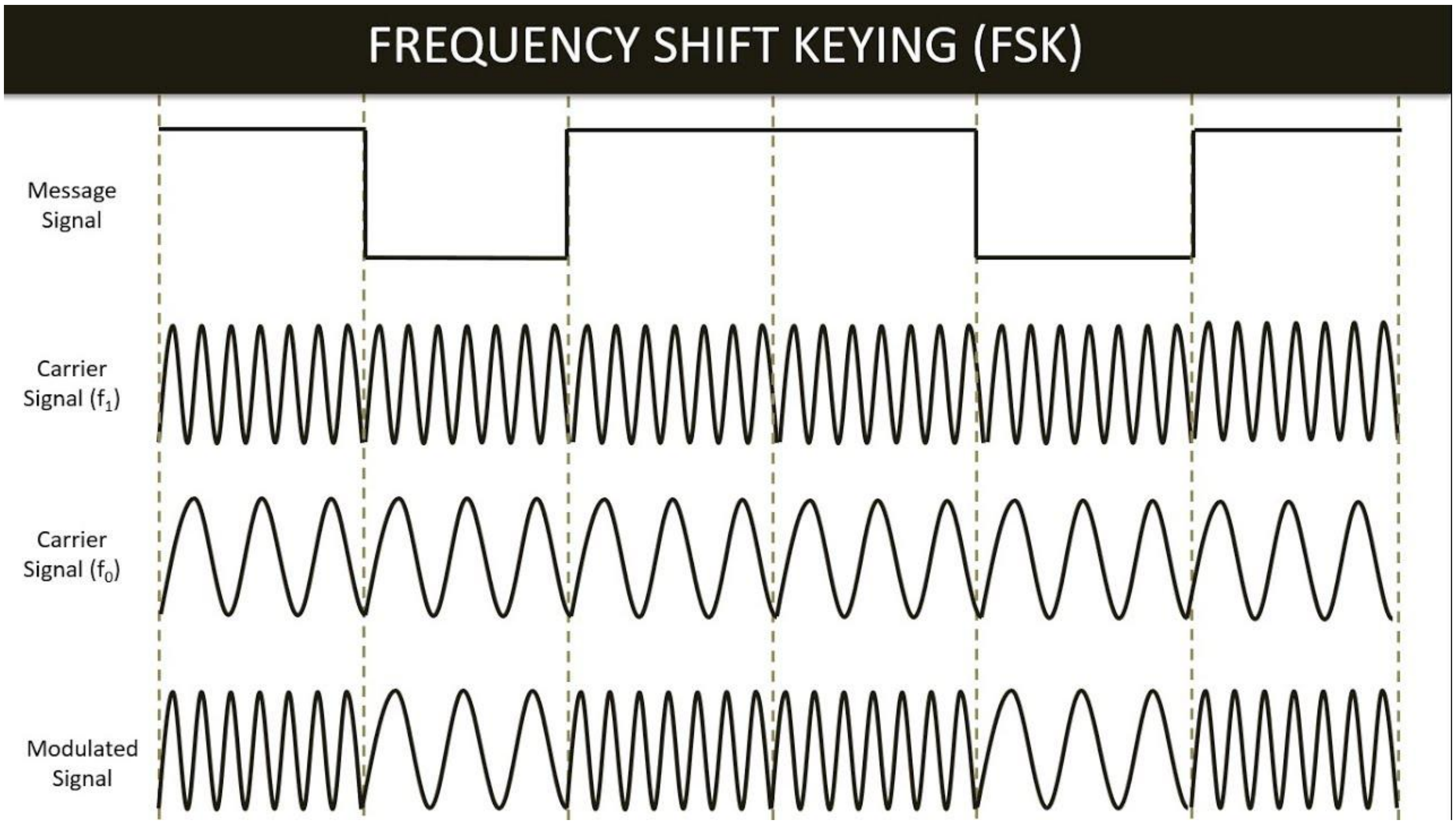
Example 5.4

In data communications, we normally use full-duplex links with communication in both directions. We need to divide the bandwidth into two with two carrier frequencies, as shown in Figure 5.5. The figure shows the positions of two carrier frequencies and the bandwidths. The available bandwidth for each direction is now 50 kHz, which leaves us with a data rate of 25 kbps in each direction.

Figure 5.5 ***Bandwidth of full-duplex ASK used in Example 5.4***



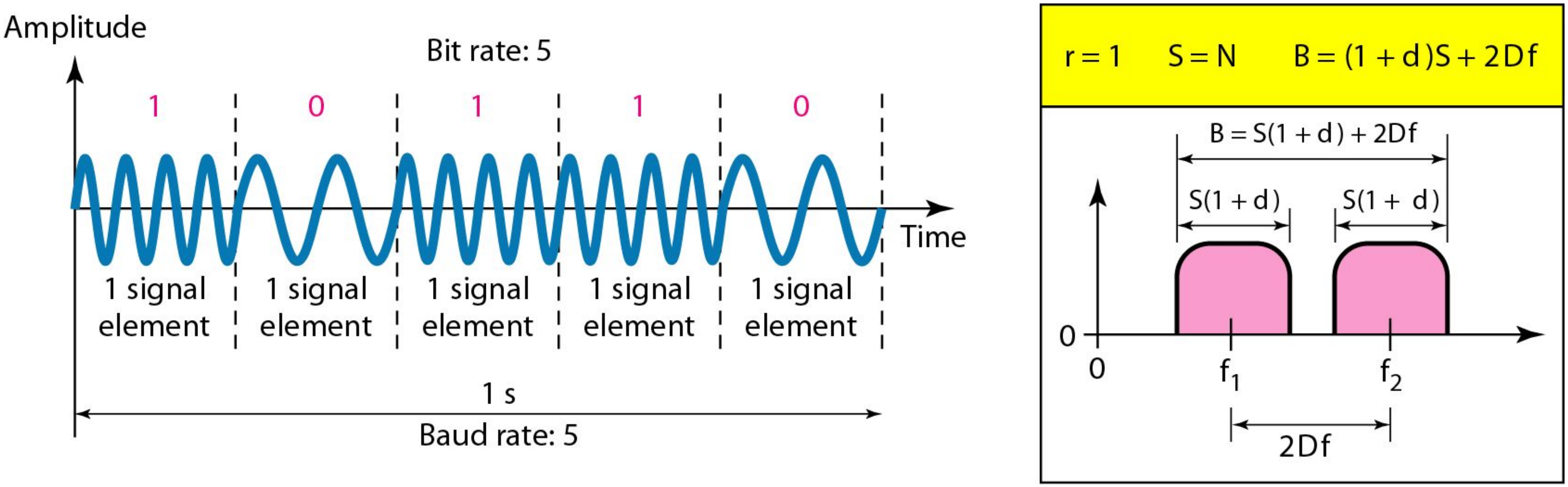
2. Frequency Shift Keying(FSK)



Frequency Shift Keying(FSK)

- The digital data stream changes the frequency of the carrier signal, f_c.
- For example, a "1" could be represented by $f_1=f_c+\Delta f$, and a "0" could be represented by $f_2=f_c-\Delta f$.

Figure 5.6 *Binary frequency shift keying*



Bandwidth of FSK

- If the difference between the two frequencies (f_1 and f_2) is $2\Delta f$, then the required BW B will be:

$$\mathbf{B = (1+d) \times S + 2\Delta f}$$

Example 5.5

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with d = 1?

Solution

This problem is similar to Example 5.3, but we are modulating by using FSK. The midpoint of the band is at 250 kHz. We choose 2Δf to be 50 kHz; this means

$$B = (1 + d) \times S + 2\Delta f = 100 \quad \longrightarrow \quad 2S = 50 \text{ kHz} \quad S = 25 \text{ kbaud} \quad N = 25 \text{ kbps}$$

Coherent and Non Coherent

- In a non-coherent FSK scheme, when we change from one frequency to the other, we do not adhere to the current phase of the signal.
- In coherent FSK, the switch from one frequency signal to the other only occurs at the same phase in the signal.

Multi level FSK

- Similarly to ASK, FSK can use **multiple bits per signal element.**
- That means we need to provision for multiple frequencies, each one to represent a group of data bits.
- The bandwidth for FSK can be higher

$$B = (1+d)\times S + (L-1)/2\Delta f = L\times S$$

Figure 5.7 *Bandwidth of MFSK used in Example 5.6*

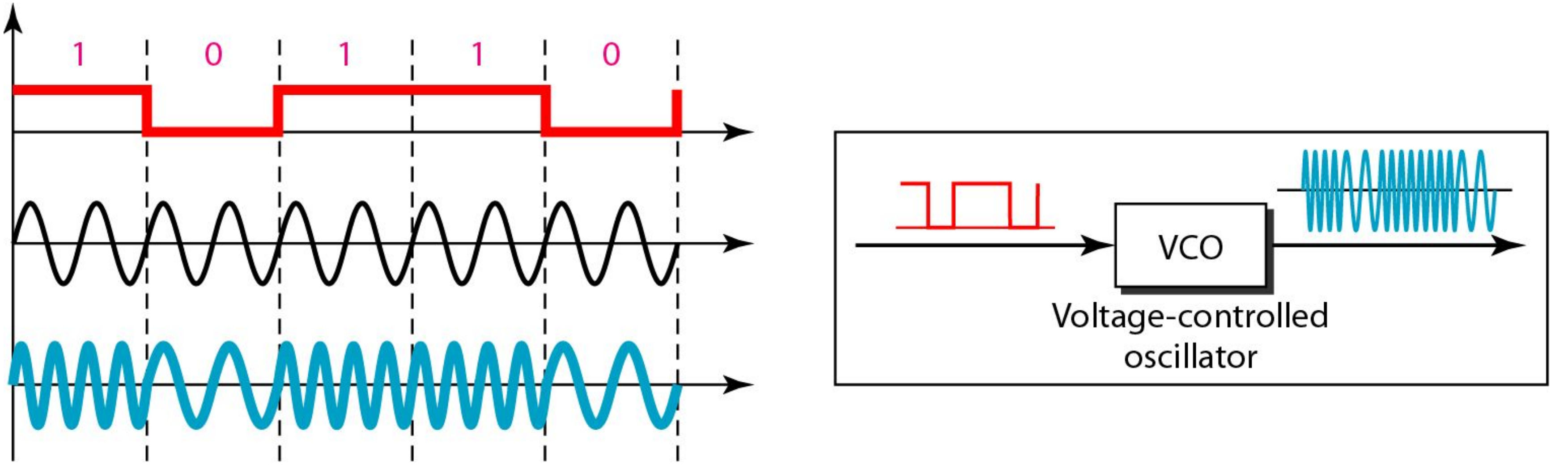


Figure 5.7 *Bandwidth of MFSK used in Example 5.6*

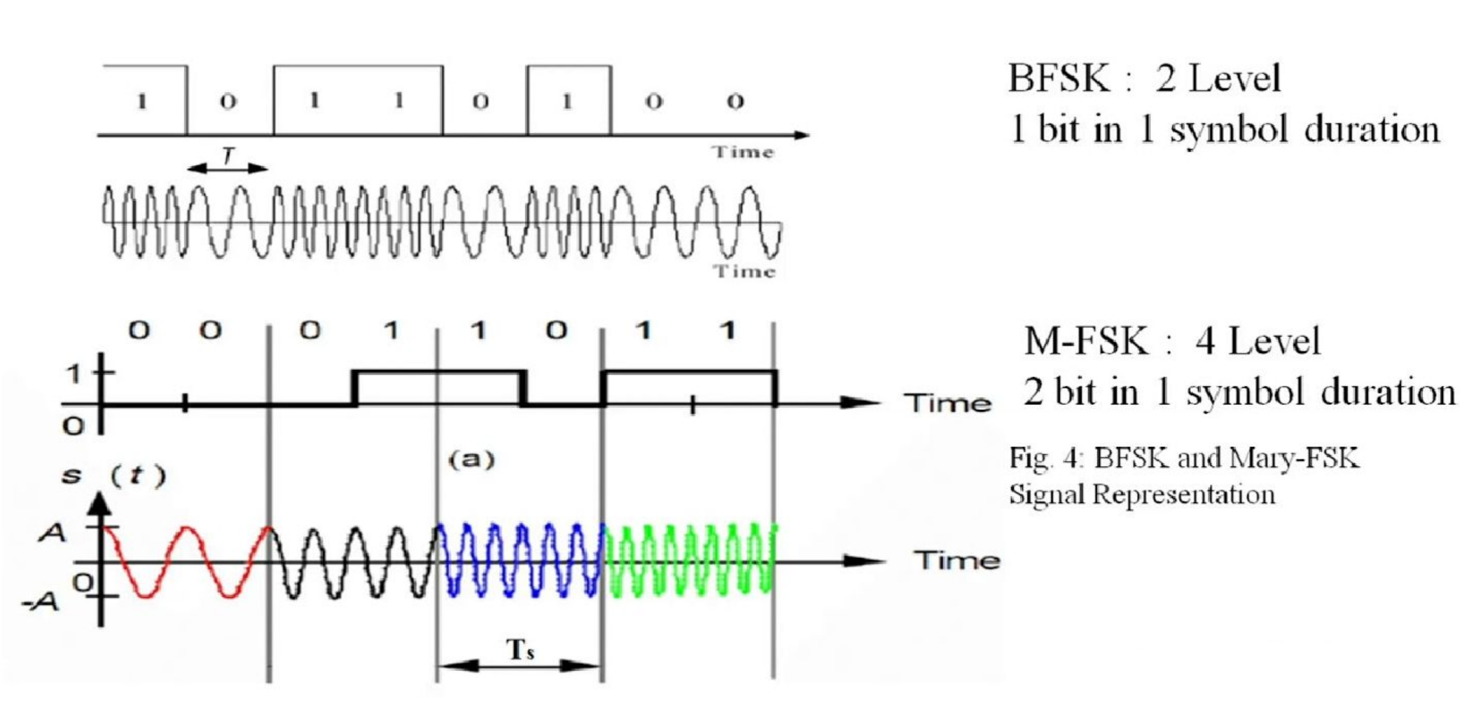


Fig. 4: BFSK and Mary-FSK Signal Representation

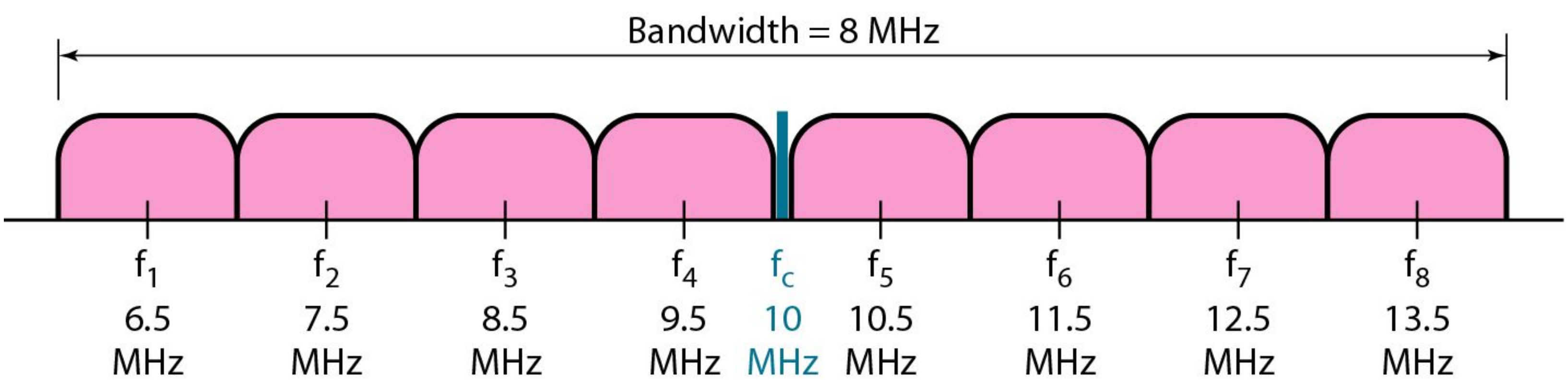
Example 5.6

We need to send data 3 bits at a time at a bit rate of 3 Mbps. The carrier frequency is 10 MHz. Calculate the number of levels (different frequencies), the baud rate, and the bandwidth.

Solution

We can have $L = 2^3 = 8$. The baud rate is $S = 3$ Mbps/3 = 1 Mbaud. This means that the carrier frequencies must be 1 MHz apart ($2\Delta f = 1$ MHz). The bandwidth is $B = 8 \times 1M = 8M$. Figure 5.8 shows the allocation of frequencies and bandwidth.

Figure 5.8 *Bandwidth of MFSK used in Example 5.6*



3. Phase Shift Keyeing

- We vary the phase shift of the carrier signal to represent digital data.
- Both peak amplitude and frequency remain constant as the phase changes.
- PSK is more common than ASK or FSK
- The simplest PSK is binary PSK

3. Phase Shift Keyeing

- We vary the phase shift of the carrier signal to represent digital data.
- The bandwidth requirement, B is:

$$\mathbf{B = (1+d)xS}$$

- PSK is much more robust than ASK as it is not that vulnerable to noise, which changes amplitude of the signal. Noise can change the amplitude easier than it can change the phase

Figure 5.9 *Binary phase shift keying*

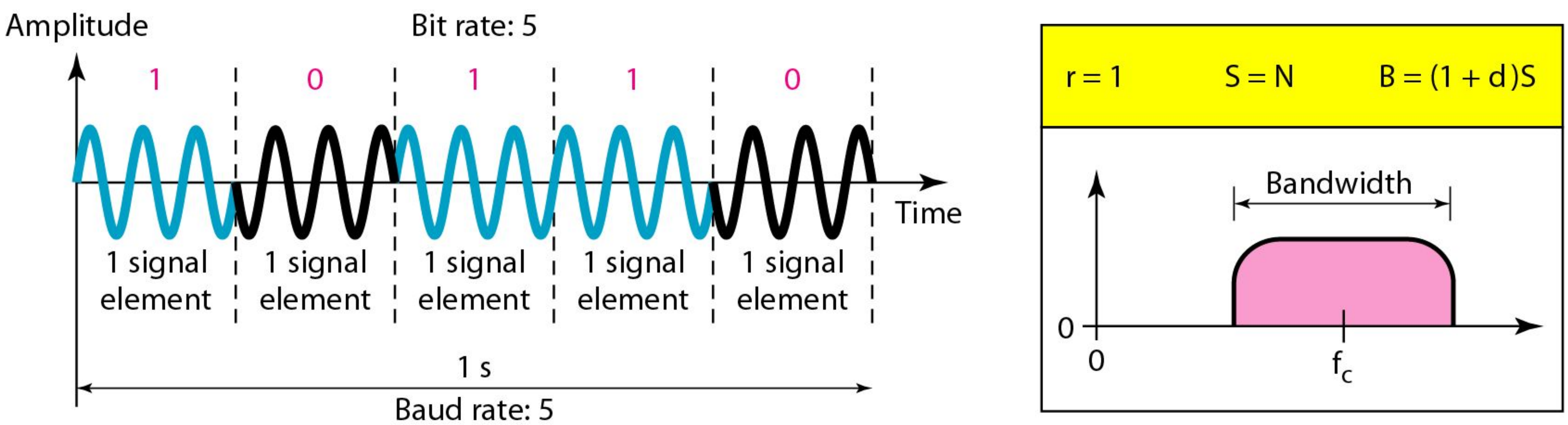
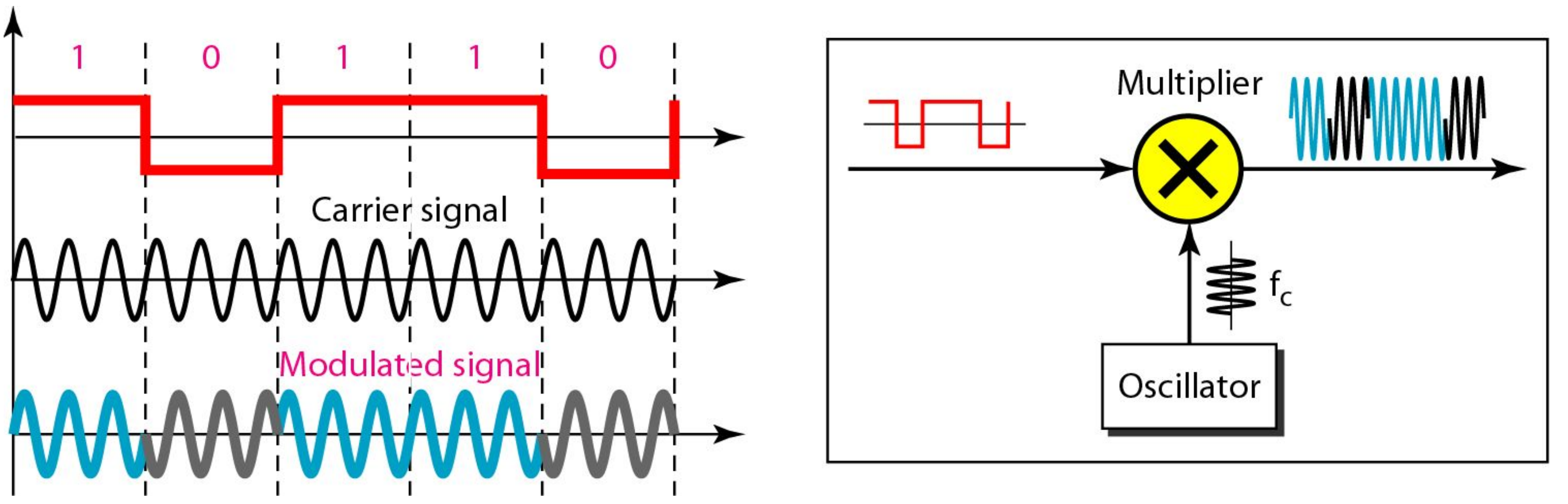


Figure 5.10 *Implementation of BASK*



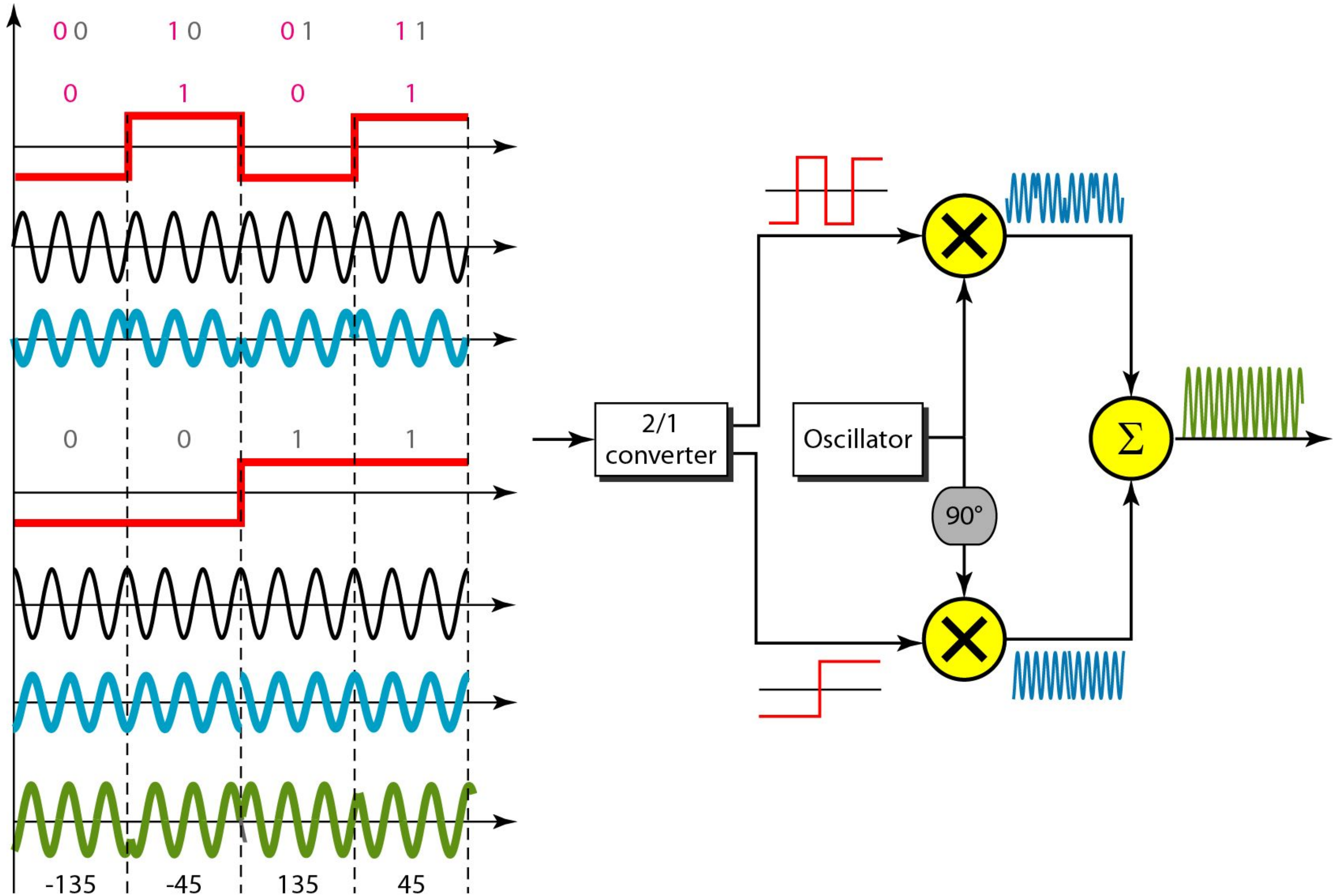
Quadrature PSK

- To increase the bit rate, we can code 2 or more bits onto one signal element.
- In QPSK, we parallelize the bit stream so that every two incoming bits are split up and PSK a carrier frequency. One carrier frequency is phase shifted 90° from the other - in quadrature.
- The two PSKed signals are then added to produce one of 4 signal elements. L = 4 here.

Quadrature PSK

- *QPSK because it uses two separate BPSK modulations; one* is in-phase, the other quadrature (out-of-phase).
    1. The incoming bits are first passed through a serial-to-parallel conversion that sends one bit to one modulator and the next bit to the other modulator.
    2. The two composite signals created by each multiplier are sine waves with the same frequency, but different phases.
    3. We can send 2 bits per signal element (r = 2).

Figure 5.11 *QPSK and its implementation*



Example 5.7

Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of $d = 0$.

Solution

For QPSK, 2 bits is carried by one signal element. This means that $r = 2$. So the signal rate (baud rate) is $S = N \times (1/r) = 6$ Mbaud. With a value of $d = 0$, we have $B = S = 6$ MHz.

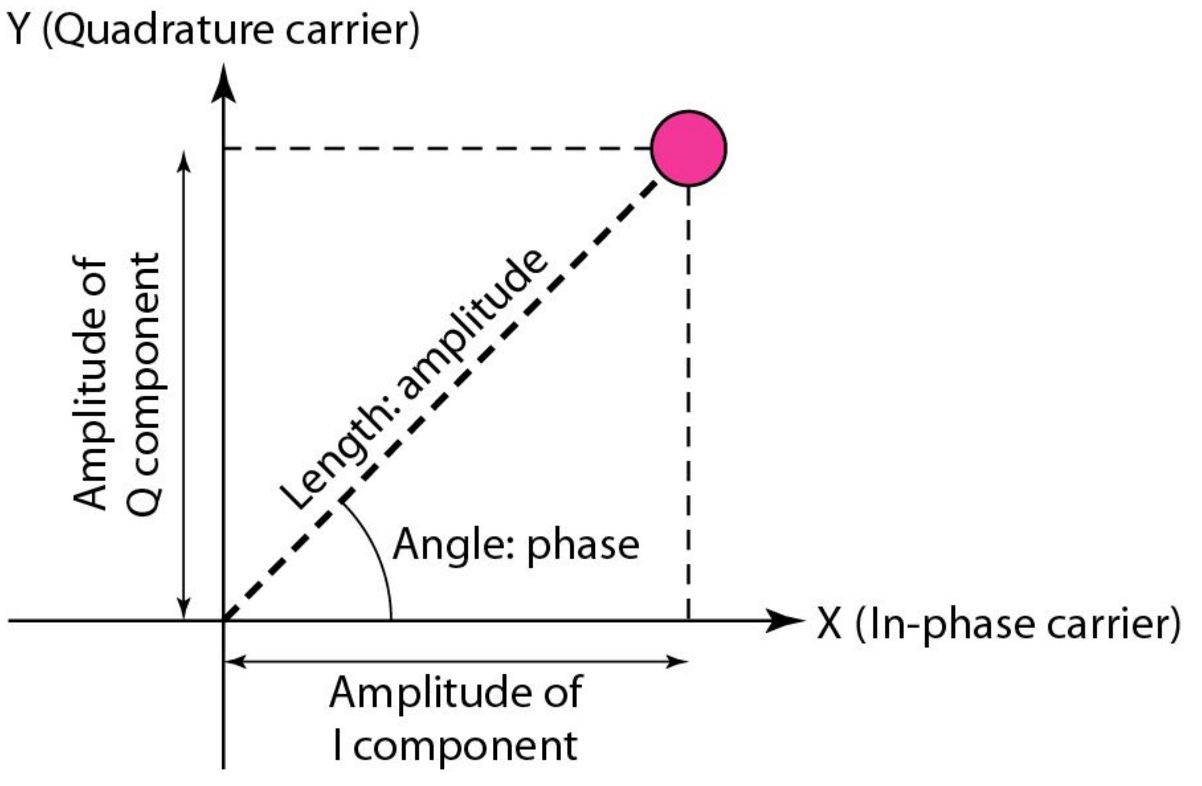
Constellation Diagrams

- A **constellation diagram** helps us to define the amplitude and phase of a signal when we are using two carriers, one in quadrature of the other.
- The X-axis represents the in-phase carrier and the Y-axis represents quadrature carrier.

Constellation Diagrams

- Diagram is useful when we are dealing with multilevel ASK, PSK, or QAM
- In a constellation diagram, a signal element type is represented as a dot

Figure 5.12 *Concept of a constellation diagram*



Constellation Diagrams

- The projection of the point on the *X axis* defines the peak amplitude of the in-phase component
- The projection of the point on the *Y axis defines the peak amplitude of the quadrature component.*

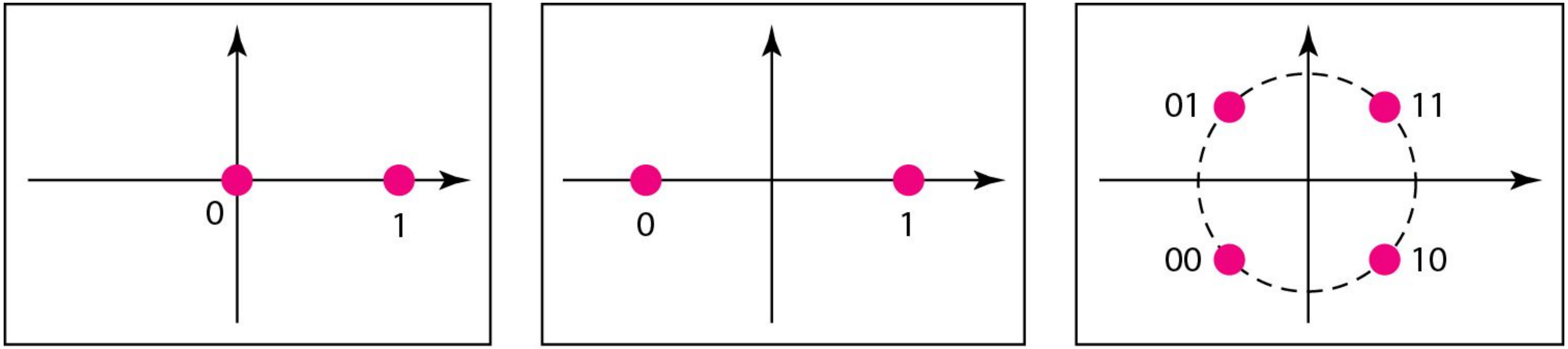
Example 5.8

Show the constellation diagrams for an ASK (OOK), BPSK, and QPSK signals.

Solution

Figure 5.13 shows the three constellation diagrams.

Figure 5.13 *Three constellation diagrams*



a. ASK (OOK) b. BPSK c. QPSK

1. **Figure 5.13** ***Three constellation diagrams***

1. For ASK, we are using only an in-phase carrier. Therefore, the two points should be on the *X axis,* Binary 0 has an amplitude of 0 V; binary 1 has an amplitude of 1 V (for example) The points are located at the origin and at 1 unit.
2. BPSK also uses only an in-phase carrier. However, we use a polar NRZ signal for modulation It creates two types of signal elements, one with amplitude 1 and the other with amplitude -1. BPSK creates two different signal elements, one with amplitude 1 V and in phase and the other with amplitude 1 V and 180° out of phase.
3. QPSK uses two carriers, one in-phase and the other quadrature. The point representing 11 is made of two combined signal elements, both with an amplitude of 1 V. One element is represented by an in-phase carrier, the other

Note

Quadrature amplitude modulation is a combination of ASK and PSK.

Figure 5.14 *Constellation diagrams for some QAMs*

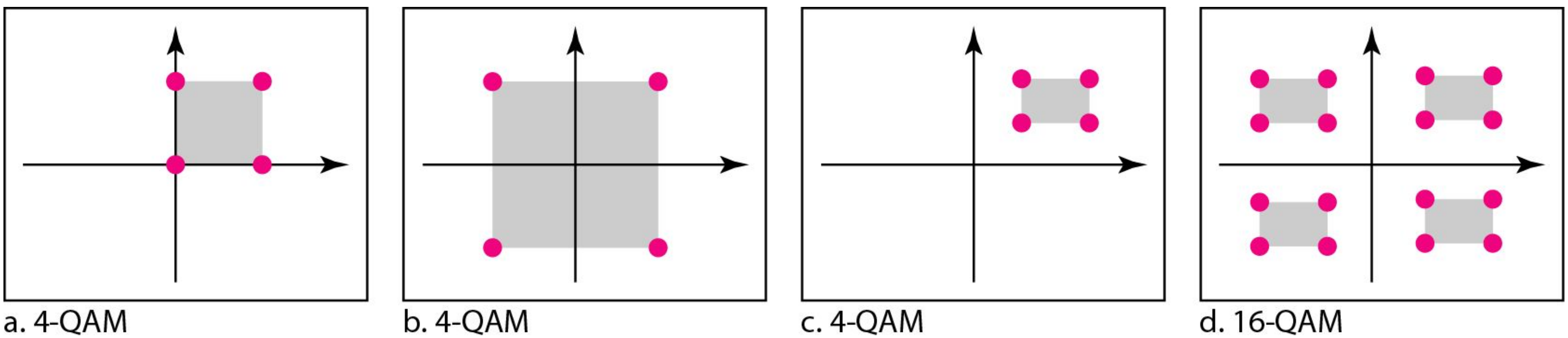


Figure 5.14 *Constellation diagrams for some QAMs*

The possible variations of QAM are numerous

1. Simplest 4-QAM scheme (four different signal element types) using a unipolar NRZ signal to modulate each carrier.
2. 4-QAM using polar NRZ-exactly the same as QPSK
3. QAM-4 in which we used a signal with two positive levels to modulate each of the two carriers
4. 16-QAM constellation of a signal with eight levels, four positive and four negative.