Data Communications and Networking Fourth Edition

Forouzan

Chapter 4

Digital Transmission

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

Figure 4.1 *Line coding and decoding*

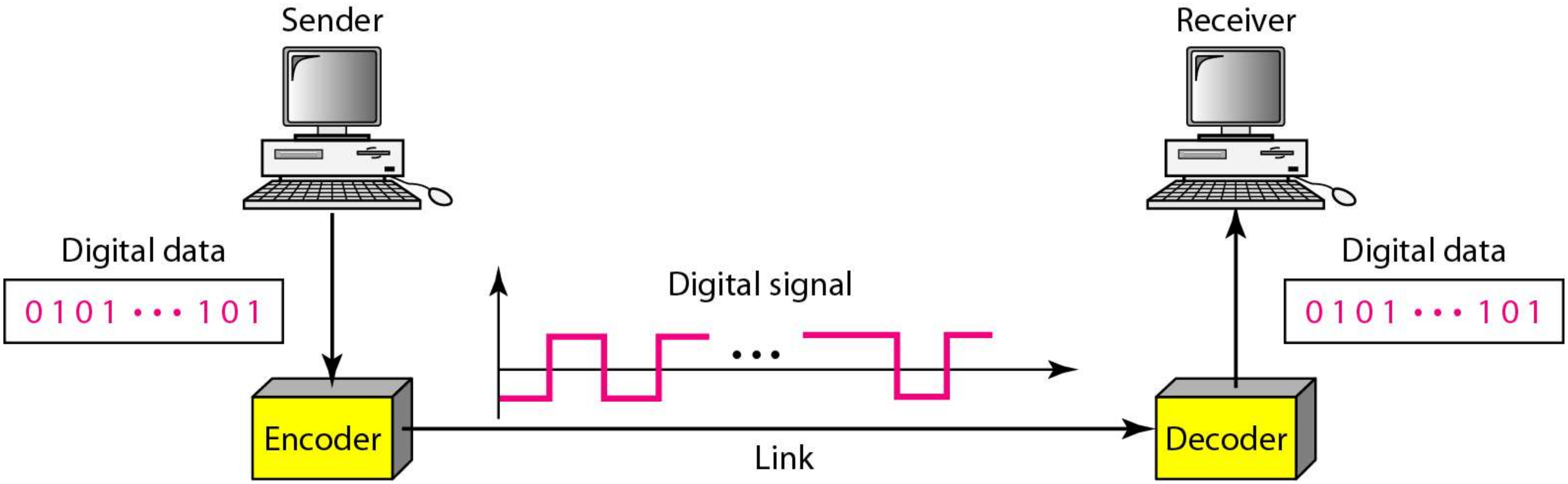
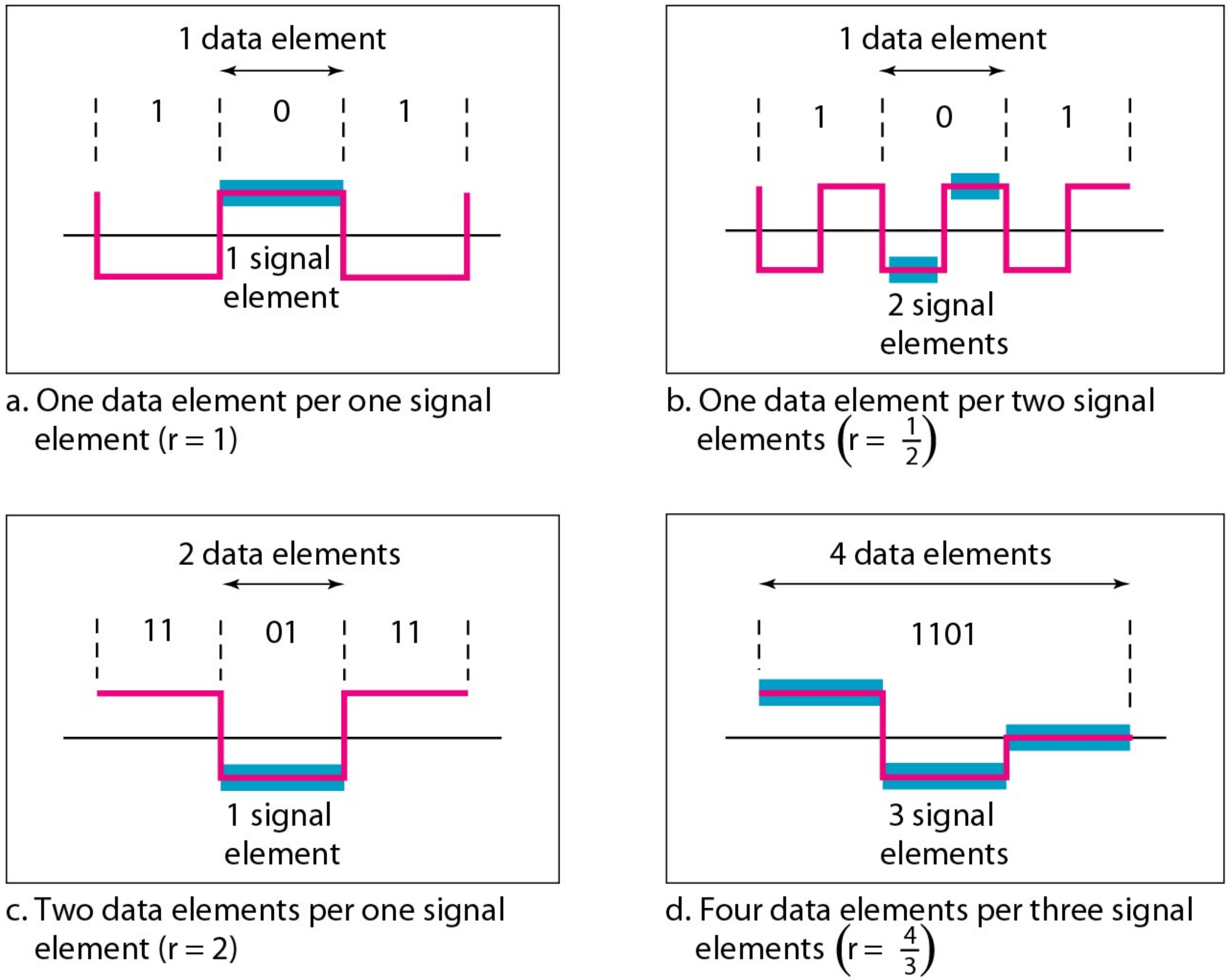


Figure 4.2 *Signal element versus data element (ratio)*

guarantee synchronization

- 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.
- r=data element/ signal 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 $\left(r = \frac{4}{3}\right)$

- The data rate defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps). The signal rate is the number of signal elements sent in 1s. The unit is the baud. There are several common terminologies used in the literature.

  The data rate is sometimes called the bit rate; the signal rate is sometimes called the pulse rate, the modulation rate, or the baud rate.

One goal in data communications is to increase the data rate while decreasing the signal rate

- Example
- Suppose each data element is a person who needs to be carried from one place to another. We can think of a signal element as a vehicle that can carry people. When r = 1, it means each person is driving a vehicle. When r > 1, it means more than one person is travelling in a vehicle (a carpool, for example). We can also have the case where one person is driving a car and a trailer (r = 1/2).

Figure 4.4 *Line coding schemes*

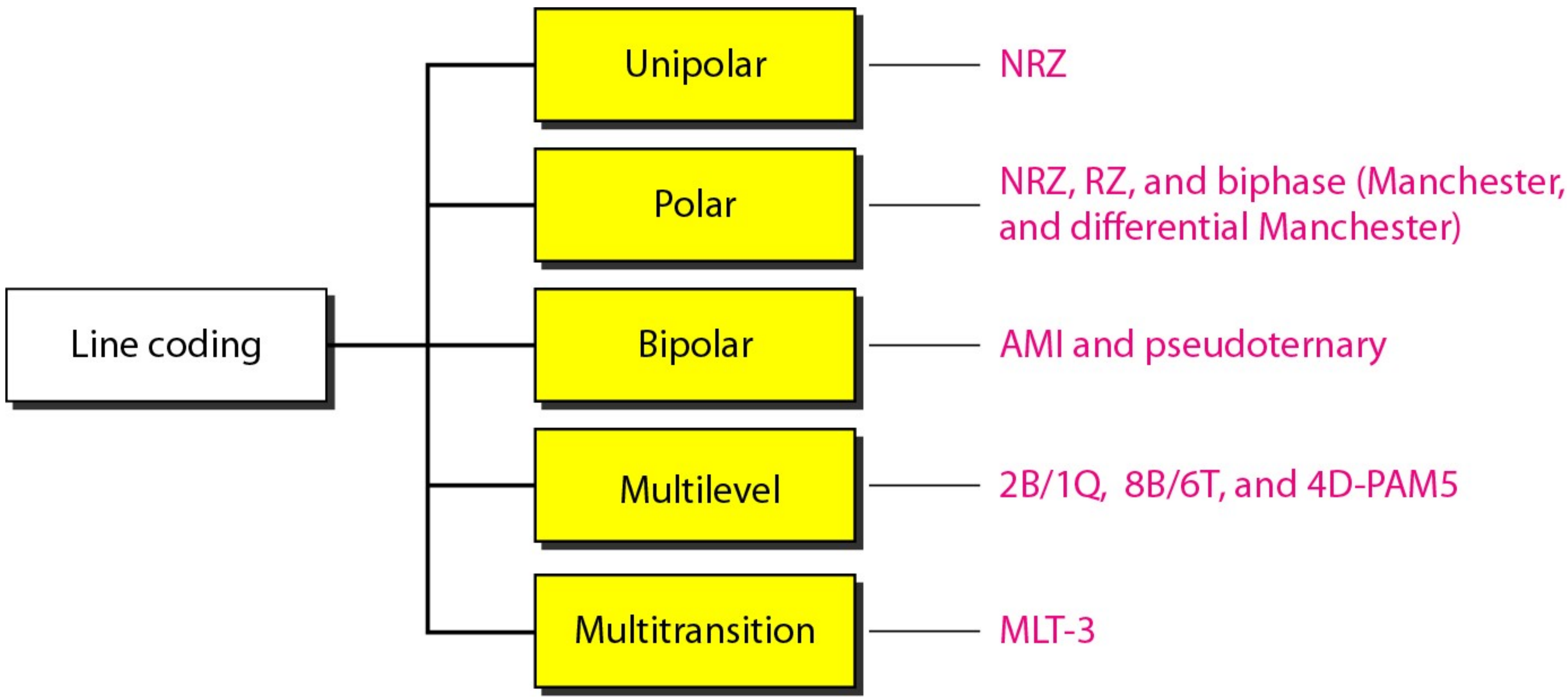


Figure 4.5 *Unipolar NRZ scheme (*(Non-Return-to-Zero)*)*

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.

NRZ because the signal does not return to zero in the middle of the bit

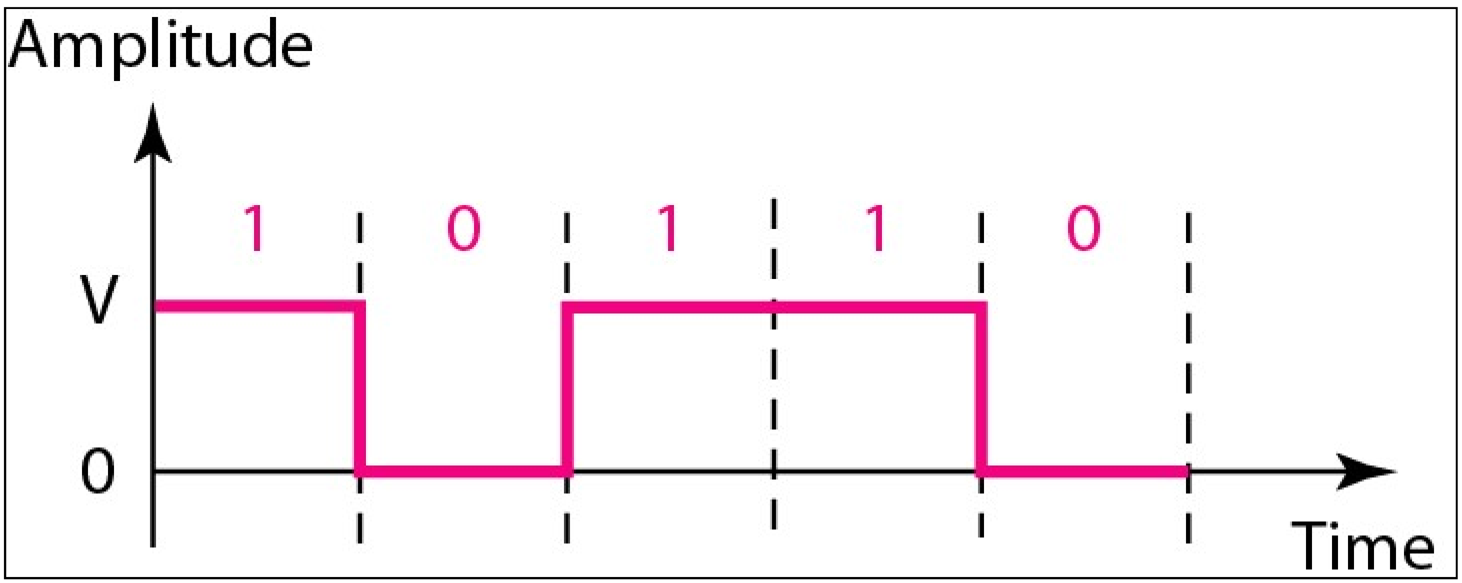


Figure 4.6 *Polar NRZ-L and NRZ-I schemes*

Polar: the voltages are on both sides of the time axis.

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

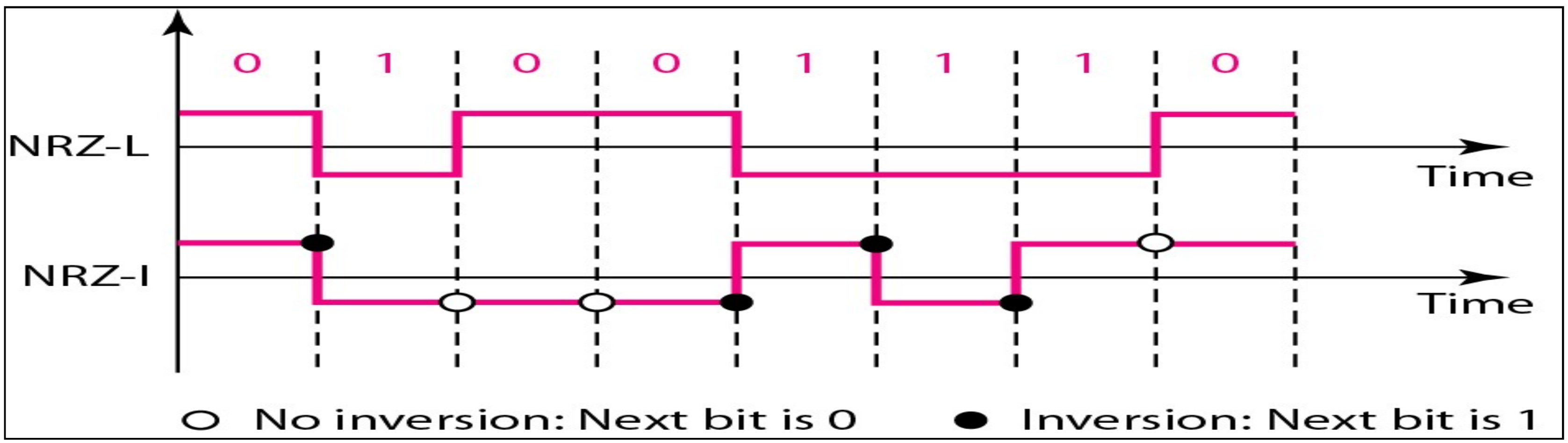


Figure 4.7 ***Polar RZ scheme,*** Return-to-Zero (RZ)

RZ, which uses three values: positive, negative, and zero.

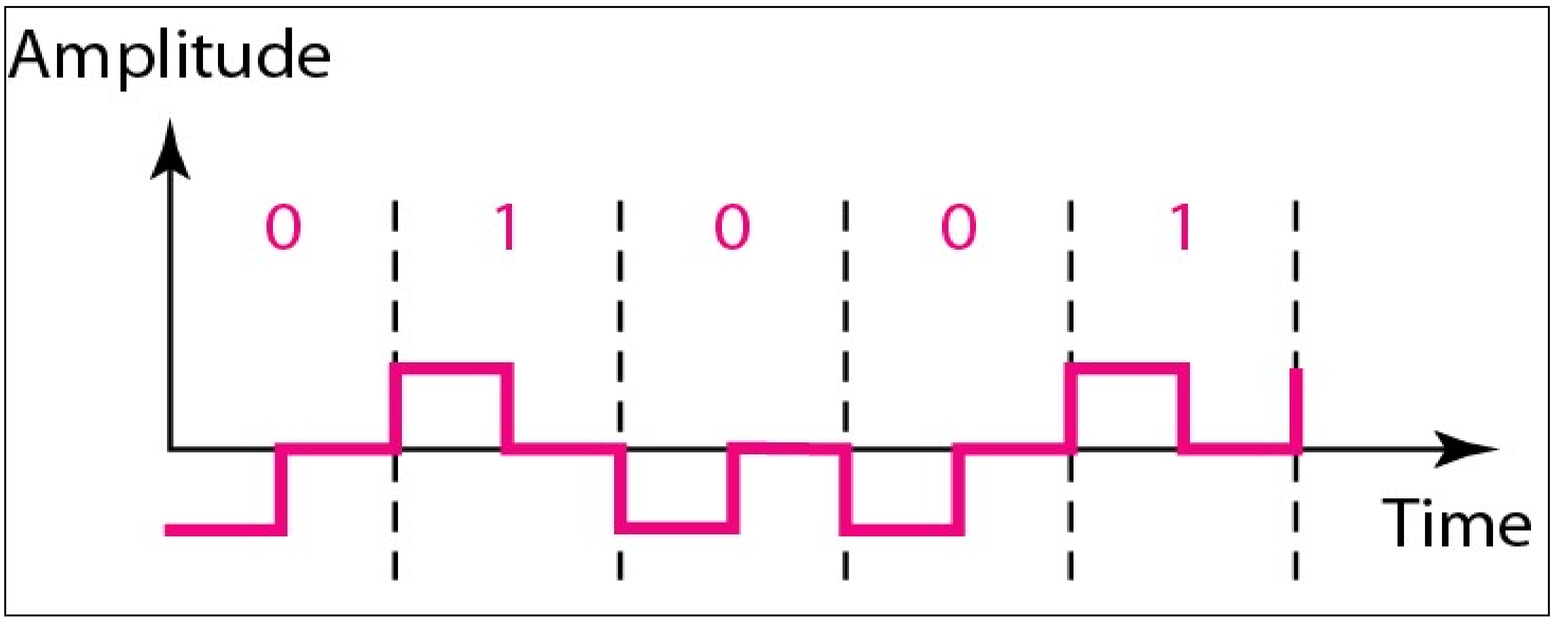
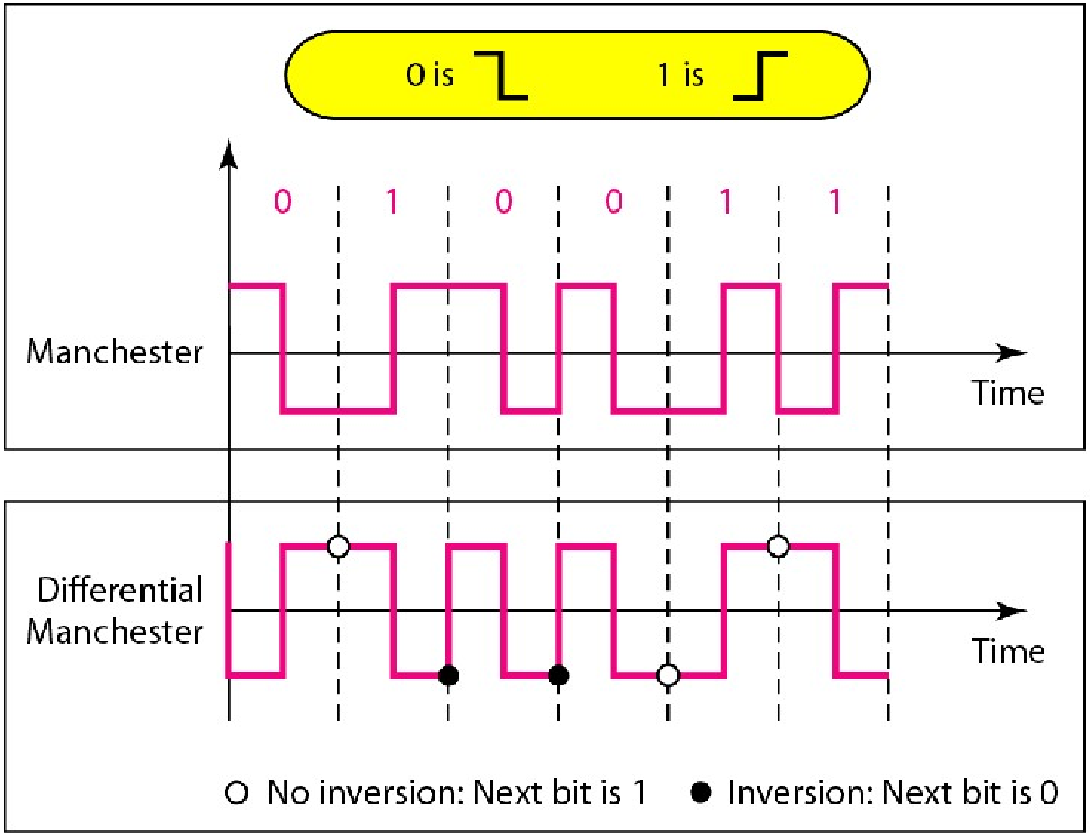


Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*

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



Manchester scheme combines RZ and NRZ-L .
Differential Manchester combines the ideas of RZ and NRZ-I.

Note that Manchester and differential Manchester schemes are also called biphase schemes.

Figure 4.9 ***Bipolar schemes: AMI and pseudoternary (***multilevel binary) alternate mark inversion (AMI)

In bipolar encoding, we use three levels: positive, zero, and negative.

A neutral zero voltage represents binary 0, 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.

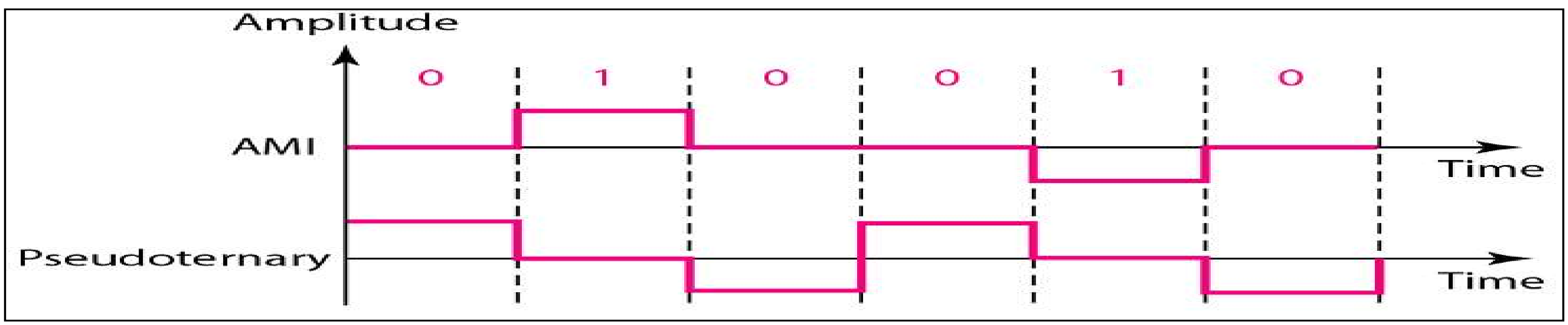


Figure 4.10 ***Multilevel: 2B1Q scheme,*** two binary, one quaternary (2B1Q),

In *m*B*n*L schemes, a pattern of *m* data elements is encoded as a pattern of *n* signal elements in which $2^m \leq L^n$.

uses data patterns of size **2** and encodes the 2-bit patterns as **one signal** element belonging to a **four-level signal**

Figure 4.10 *Multilevel: 2B1Q scheme*

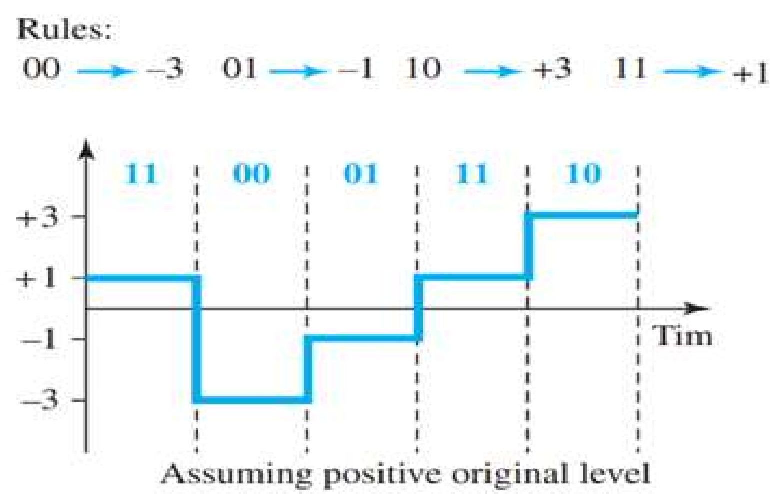
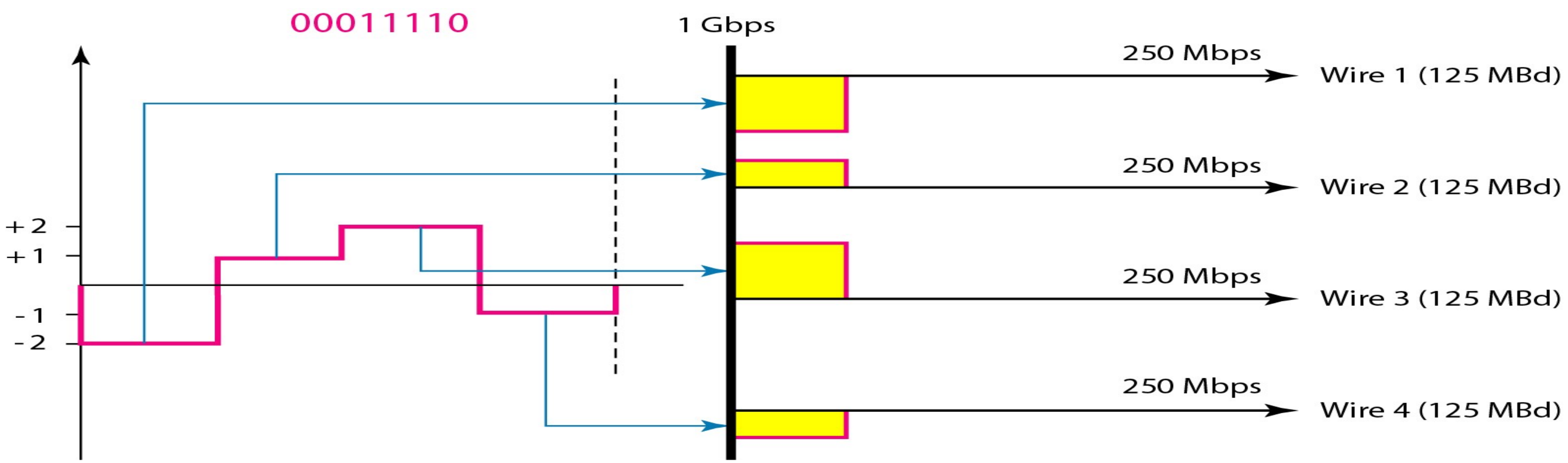


Figure 4.12 ***Multilevel: 4D-PAM5 scheme,*** four-dimensional five-level pulse amplitude modulation (4D-PAM5).

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.
All 8 bits can be fed into a wire simultaneously and sent by using one signal element



4-1 DIGITAL-TO-DIGITAL CONVERSION

Topics discussed in this section:

Line Coding
Line Coding Schemes
Block Coding

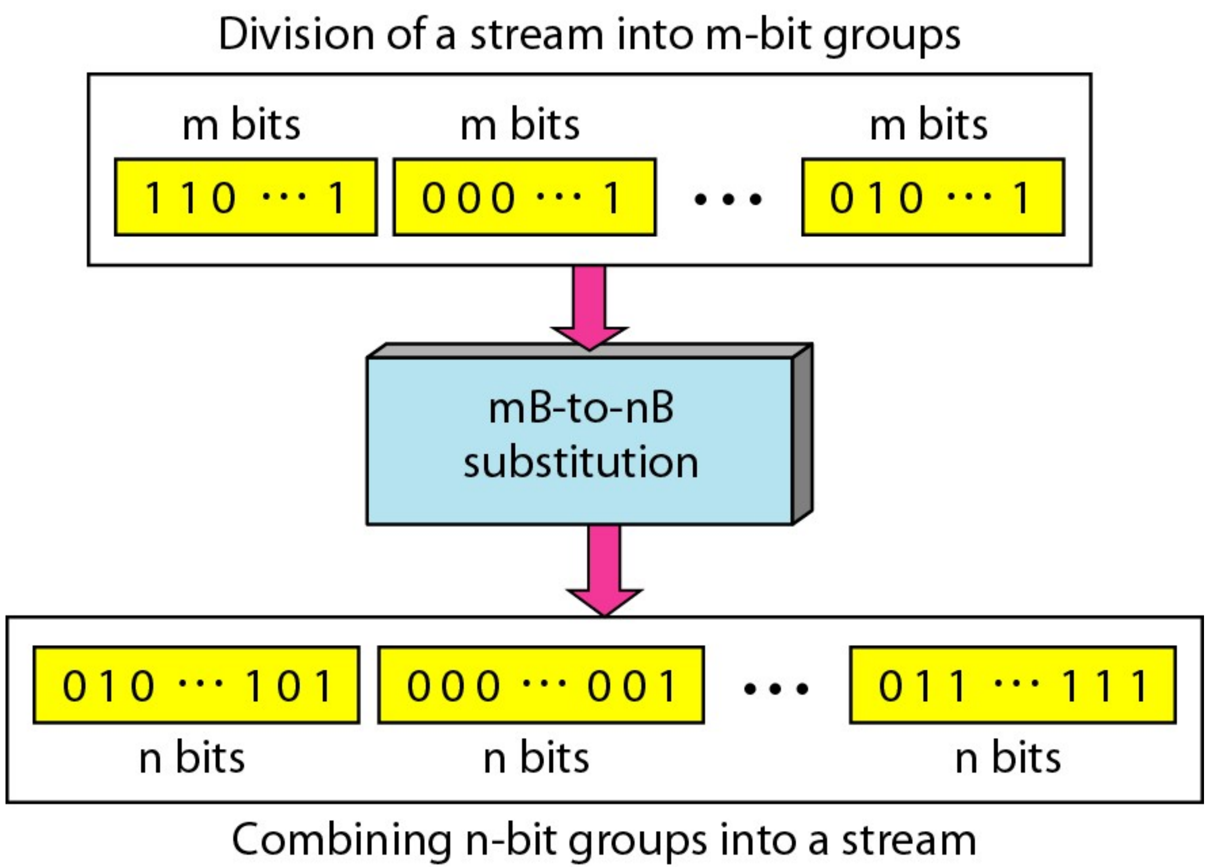
In general, block coding changes a block of m bits into a block of n bits,

Block coding normally involves three steps: division, substitution, and combination.

Note

**Block coding is normally referred to as *m*B/*n*B coding;
it replaces each *m*-bit group with an *n*-bit group.**

Figure 4.14 *Block coding concept*



In the **division** step, a sequence of bits is divided into groups of m bits. For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups. The heart of block coding is the **substitution** step. In this step, we substitute an m-bit group with an n-bit group. For example, in 4B/5B encoding we substitute a 4-bit group with a 5-bit group. Finally, the n-bit groups are **combined** to form a stream. The new stream has more bits than the original bits.

Figure 4.15 *Using block coding 4B/5B with NRZ-I line coding scheme*

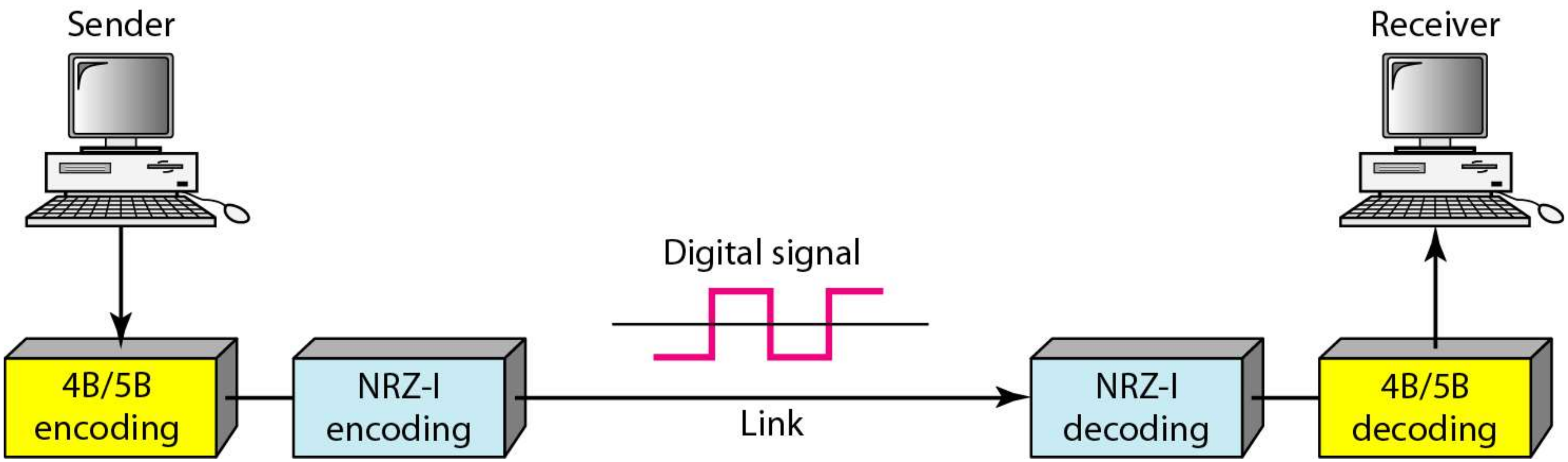


Table: ***4B/5B mapping codes***

4-bit Data Symbol	5-bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

4-2 ANALOG-TO-DIGITAL CONVERSION

Digitization

4-2 ANALOG-TO-DIGITAL CONVERSION

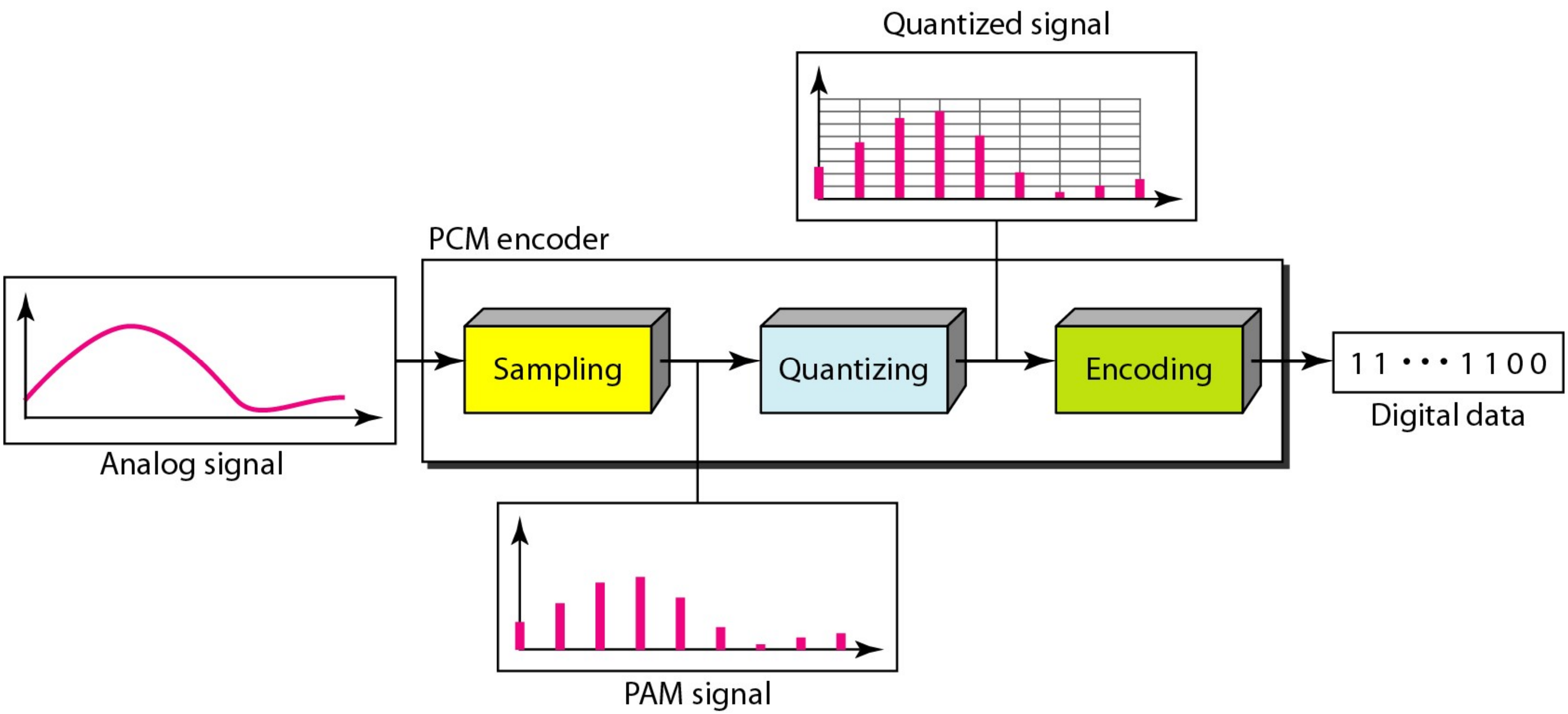
Digitization

We have seen in Chapter 3 that a digital signal is superior to an analog signal. The tendency today is to change an analog signal to digital data. In this section we describe one technique, pulse code modulation.

Topics discussed in this section:

Pulse Code Modulation (PCM)

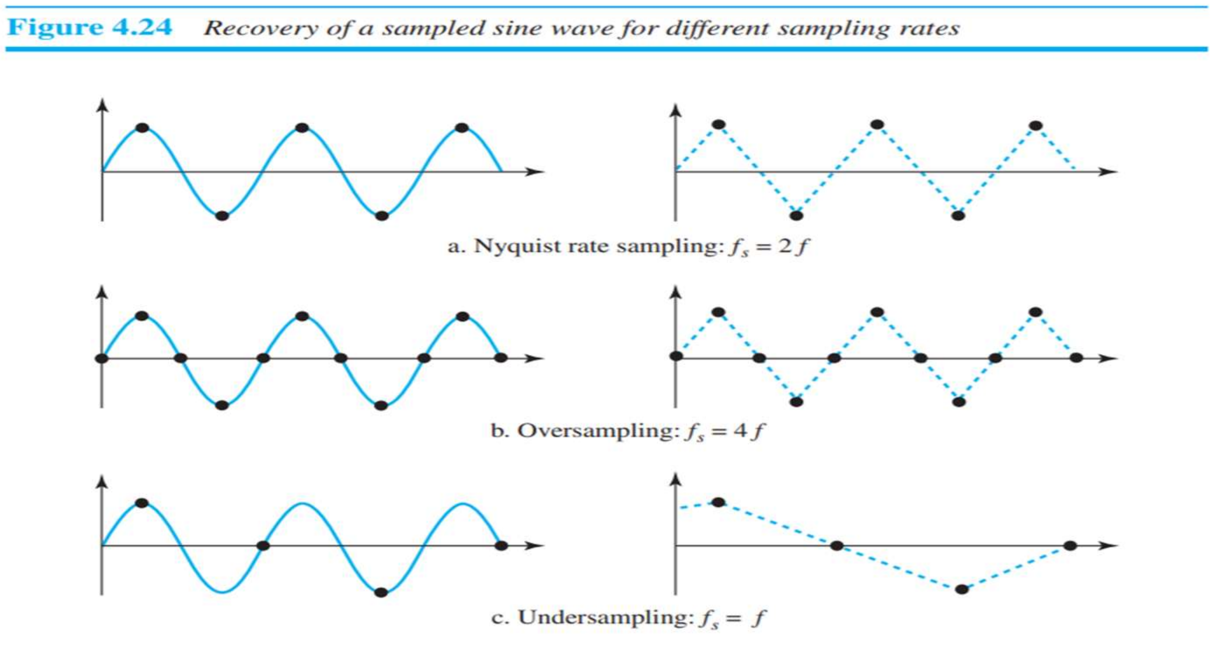
Figure 4.21 *Components of PCM encoder*



Note

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

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



let us sample a simple sine wave at three sampling rates:
$f_s = 4f$ (2 * Nyquist rate),
$f_s = 2f$ (Nyquist rate), and
$f_s = f$ (1/2 * Nyquist rate).
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.

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:

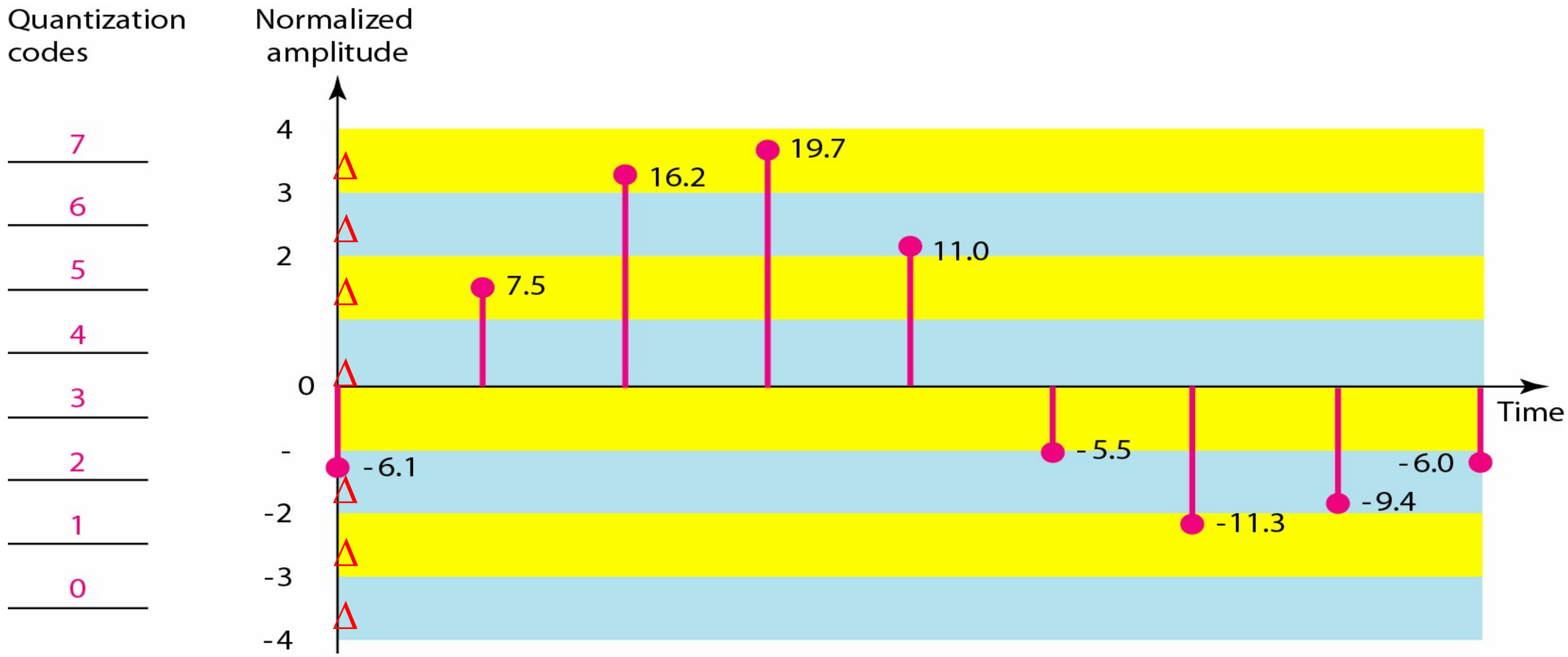
1. We assume that the original analog signal has instantaneous amplitudes between V_{min} and V_{max}.
2. We divide the range into L zones, each of height Δ (delta).

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

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.

As 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 $\Delta = 5$ V. Figure 4.26 shows this example.

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	0.26	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

Quantization error and noise

The quantization error changes the signal-to-noise ratio SNR of the signal, which in turn reduces the upper limit capacity according to Shannon.

It can be prove**n that the contribution of the quantization error to the SNR_{dB} of the signal depends on the number of quantization levels** L, **or the bits per sample** n_b, as shown in the following formula:

$$SNR_{dB} = 6.02\, n_b + 1.76\ \text{dB}$$

What is the SNR_{dB} in the example of Figure 4.26?

Solution

We can use the formula to find the quantization. We have eight levels and 3 bits per sample, so

$$SNR_{dB} = 6.02(3) + 1.76 = 19.82\ \text{dB}$$

Increasing the number of levels increases the SNR.

Example 4.13

A telephone subscriber line must have an SNR_{dB} above 40. What is the minimum number of bits per sample?

Solution

We can calculate the number of bits as

$$SNR_{dB} = 6.02n_b + 1.76 = 40 \rightarrow n = 6.35$$

Telephone companies usually assign 7 or 8 bits per sample.

Encoding

Bit rate = sampling rate * number of bits per sample = $f_s * n_b$

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:

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

4-3 TRANSMISSION MODES

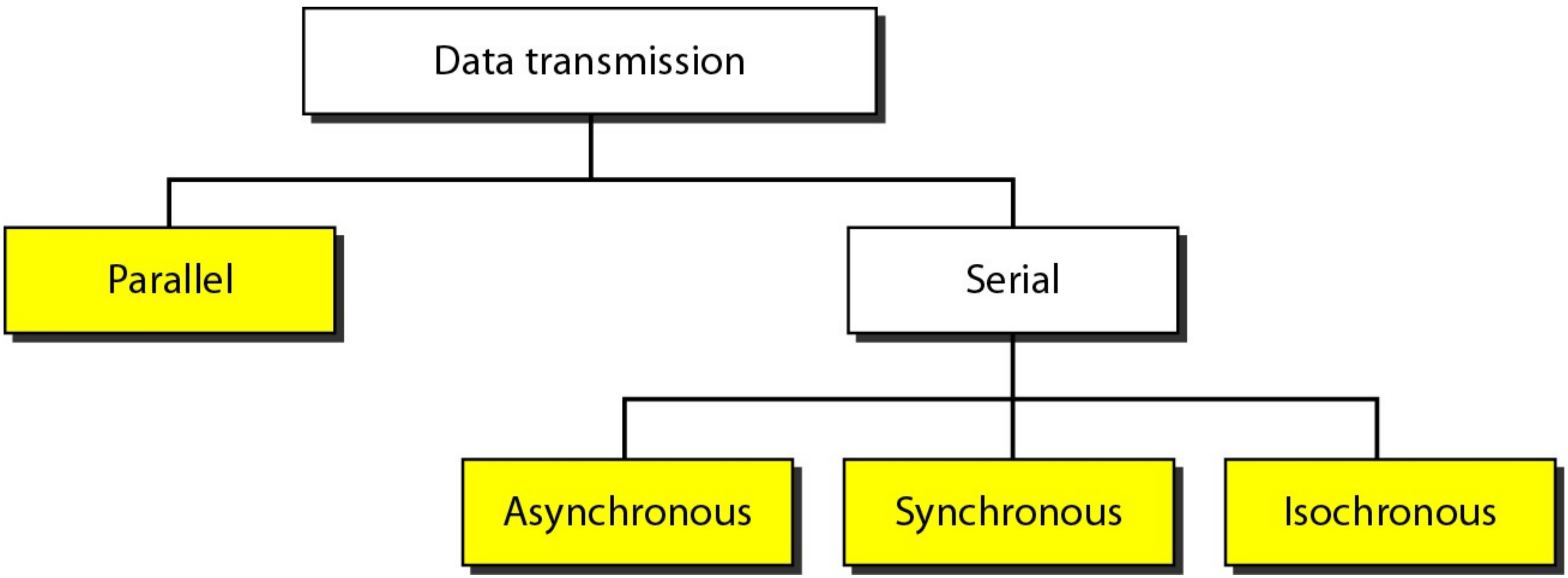
The transmission of binary data across a link can be accomplished in either parallel or serial mode.

- ***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*



No delays, The **isochronous transmission** guarantees that the data arrive at a fixed rate.

Figure 4.32 *Parallel transmission*

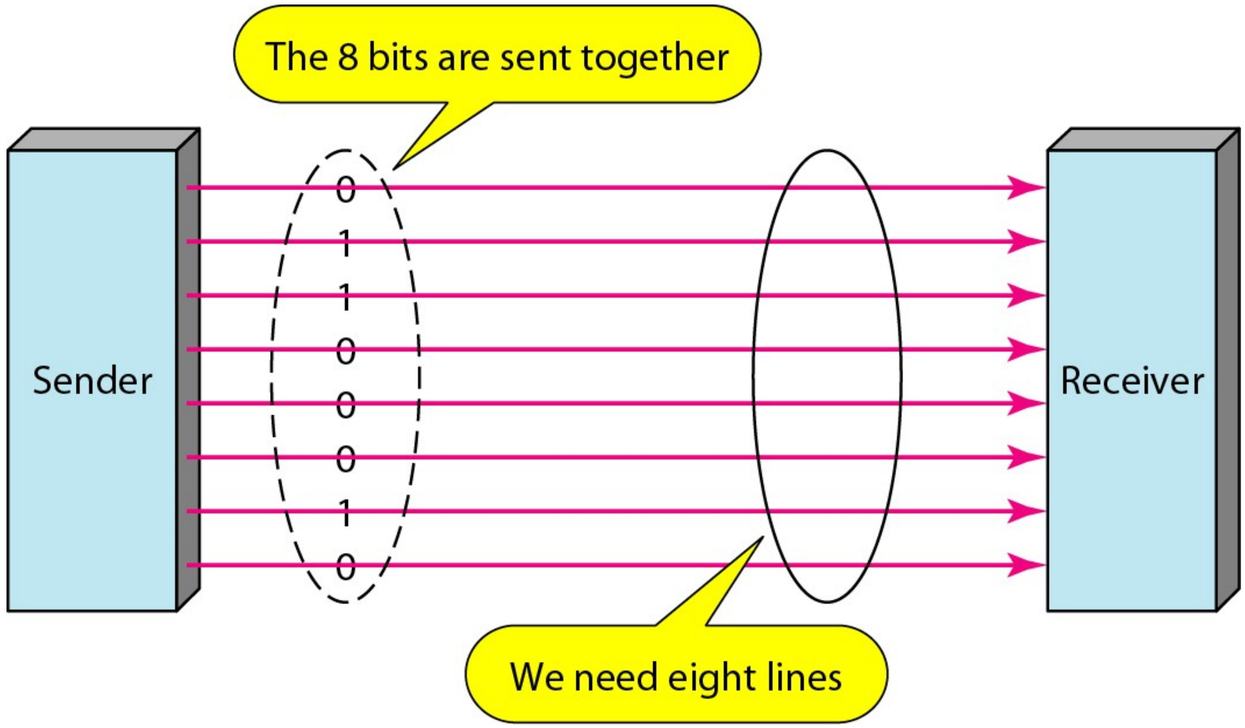
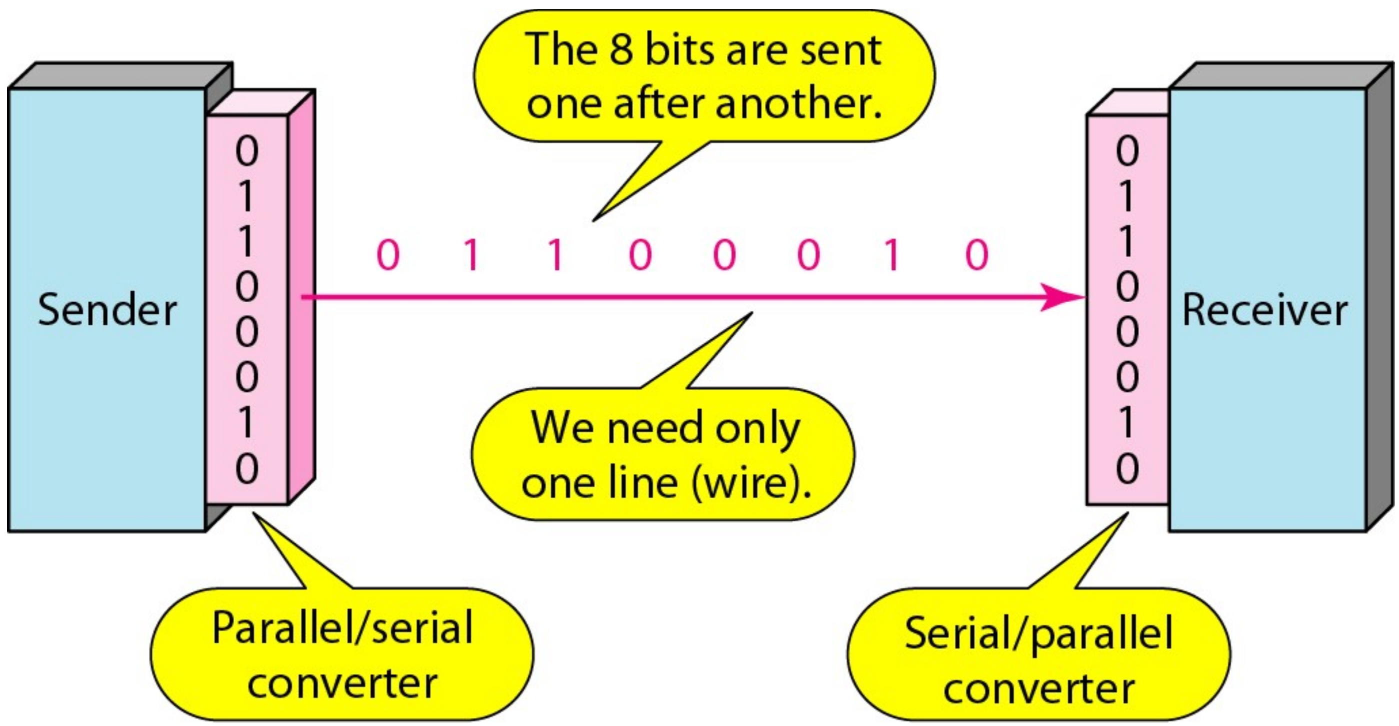


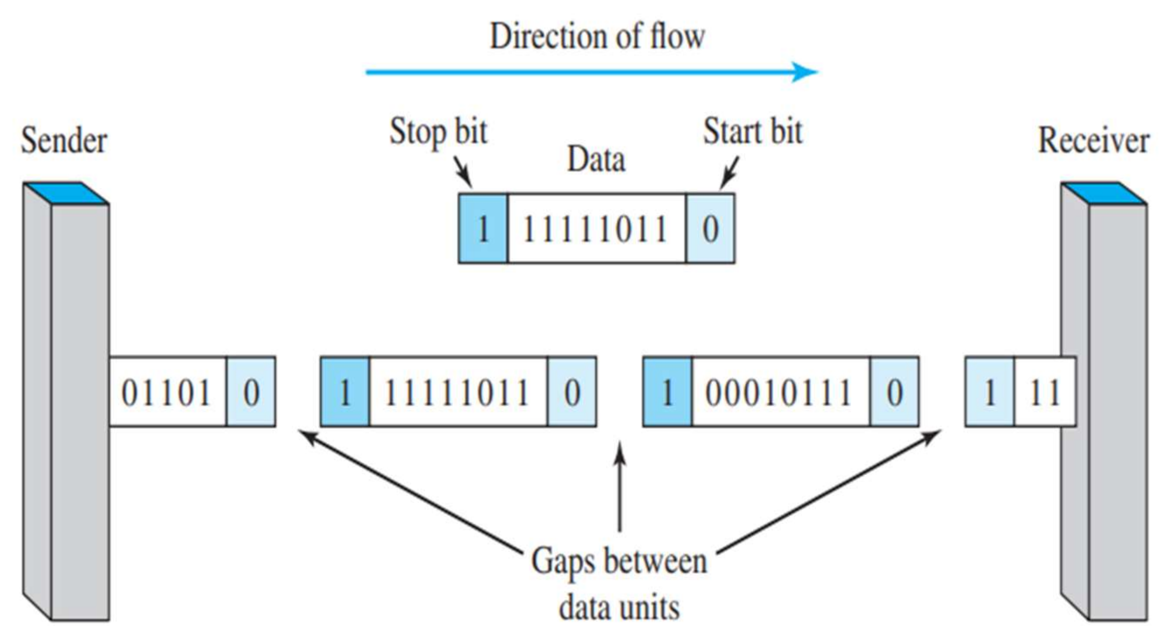
Figure 4.33 *Serial transmission*



Asynchronous Transmission

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 bytes.

Figure 4.34 *Asynchronous transmission*



Synchronous Transmission

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.

Figure 4.35 *Synchronous transmission*