Chapter 4

Digital Transmission

Dr. Mznah Al-Rodhaan

4-1 DIGITAL-TO-DIGITAL CONVERSION

In this section, we see how we can represent digital data by using digital signals. The conversion involves some techniques:

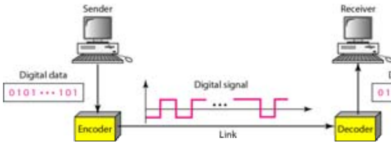
- ***Line coding,***
- ***Block coding,***
- ***Scrambling.***

Line coding is always needed; block coding may or may not be needed.

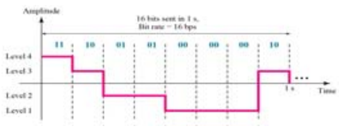
Topics discussed in this section:

Line Coding (digital baseband modulation)
Line Coding Schemes
Block Coding
Scrambling

Figure 4.1 *Line coding and decoding*



- Signal level is the number of values allowed in the signal.
- Data level is the number of values used to represent the data



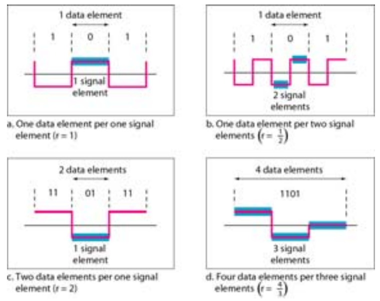
Line coding and decoding

Characteristics of Line Coding that we are going to study:

- ***Signal element versus data element***
- ***Data rate versus signal rate***
- ***DC component***
- ***Self synchronization***

Figure 4.2 *Signal element versus data element*

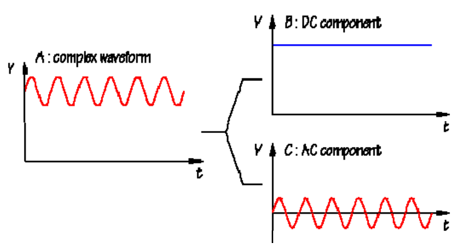
Data element = bit *Signal element = one level*



Data rate versus Signal rate

Data rate = bit rate (bps) ↑ to increase speed of transmission.
Signal rate = Pulse rate = baud rate = modulation rate (baud) ↓ to decrease the required bandwidth.
Bit rate = pulse rate * log_2 L
A pulse duration of 1ms. Calculate the pulse rate and data rate if the signal has 2 data levels or 4 data levels?
Pulse rate = 1/(1*10^{-3}) = 1000 pulse/s.
Bite rate = 1000* log_2 2 = 1000 bps.
Bite rate = 1000* log_2 4 = 2000 bps.

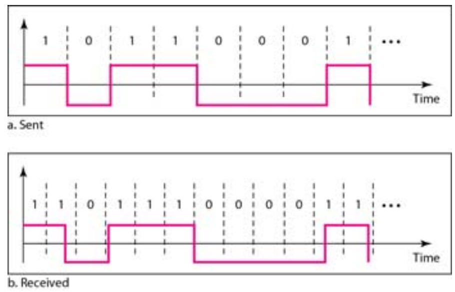
Direct Current (DC) Component



http://www.doctronics.co.uk/images/sig_17.gif

Self Synchronization

Figure 4.3 *Effect of lack of synchronization*



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 coding schemes

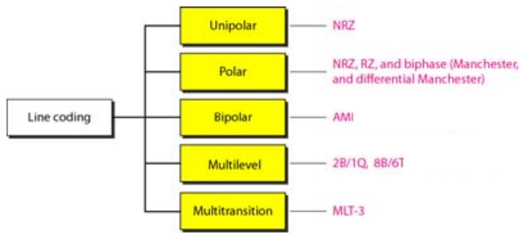
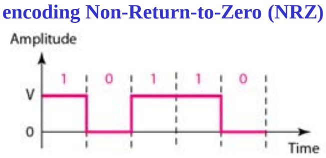


Figure 4.5 ***Unipolar NRZ scheme***

Three subcategories by voltage use:

- Unipolar (Zero and Positive)
- Polar (Negative and Positive)
- Bipolar (Negative, Zero, and Positive)

Unipolar encoding Non-Return-to-Zero (NRZ)



Unipolar NRZ is not used for data comm. Due to cost.

Polar

Polar encoding uses two voltage levels a positive and a negative voltage to represent bits

→ It solves the DC component problem(if balanced)

Absolute:

Each signal corresponds to a predetermined data

Differential:

Information is encoded by difference between current and previous signal element

Polar

Polar encoding has the following Categories:

- Non return to Zero (NRZ)
  - NRZ-L (L=Level)
  - NRZ-I (I=Inverted)
- Return to Zero (RZ)
- Manchester
- Differential Manchester

Polar

NRZ-L scheme

The voltage level is constant during a bit interval, i.e., no returns to zero

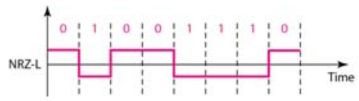
NRZ-L (L=Level)

0 = Positive voltage

1 = Negative voltage

level of voltage determines value of the bit

NRZ-L is absolute NRZ



Polar

NRZ-I scheme

NRZ-I (I=Inverted).

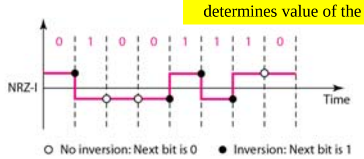
A bit is represented by the transition of the voltage level, not the voltage level itself.

0 = No inversion at beginning of bit interval

1 = Inversion at beginning of bit interval

inversion or lack of inversion determines value of the bit

It is differential NRZ



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.

NRZ-L and NRZ-I both have a DC component and synchronization problems.

Return to Zero (RZ)

Voltage level change for every bit value

→ three levels: +,-, 0

0 = Transition from negative to zero

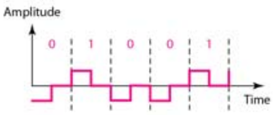
1 = Transition from positive to zero

It solves the synchronization problem

It handles both strings of 1s and 0s

Two signal changes for each bit

- More transitions
- Occupies more bandwidth



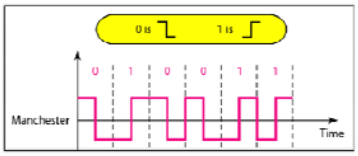
Manchester and differential Manchester

Polar biphase: Manchester scheme

Signal changes in the middle of the bit interval.

Signal change → bit representation and synchronization

0 is represented by a high voltage for the first half of the bit, followed by a low voltage (or a negative voltage) in the last half. A 1 is the exact opposite



Manchester and differential Manchester

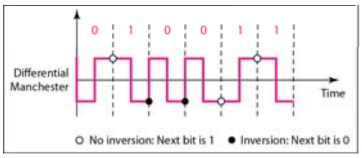
Polar biphase: Differential Manchester scheme

The inversion at the middle is used for synchronization

0 = Transition at the beginning of bit period (2 signals changes)

1 = No transition at the beginning of bit period

Disadvantages:
- Complex
- Higher frequency components (as RZ)



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

Bipolar Encoding

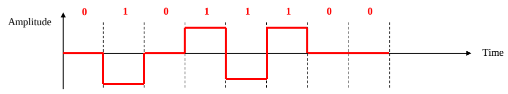
Three voltage levels are used, Like in RZ,
Zero voltage level is used for binary 0
Categories:
- Alternate Mark Inversion (AMI).
- *Pseudoternary.*

We will study AMI.

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

Figure 4.9 ***Bipolar schemes: AMI***

Alternate Mark Inversion (AMI)



Advantage
- No DC component
- Synchronized only for 1s
- Error detection

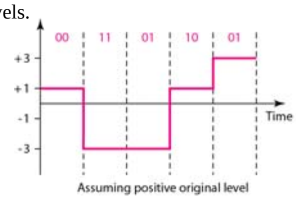
To solve the synchronization problem for 0s, we use bipolar n-zero substitution (scrambling techniques).

Multilevel schemes

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

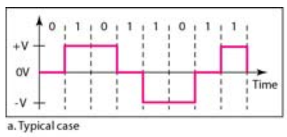
Figure 4.10 ***Multilevel: 2B1Q scheme***

Two-binary, one-quaternary (2B1Q).
2B1Q uses four signal levels.
It is used in DSL.



Multiline Transmission

MLT-3 cycles through 3 voltage levels -v, 0 and +v. It moves to the next state to transmit a 1 bit, and stays in the same state to transmit a 0 bit. Similar to simple NRZ-I encoding



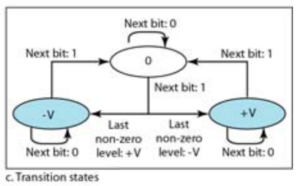


Table 4.1 *Summary of line coding schemes*

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 for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

Where B is the bandwidth and N is the data rate (bps)

Block Coding

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

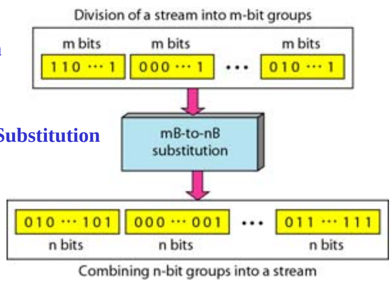
- Synchronization
- error detection capability

Block coding concept

Step 1: Division

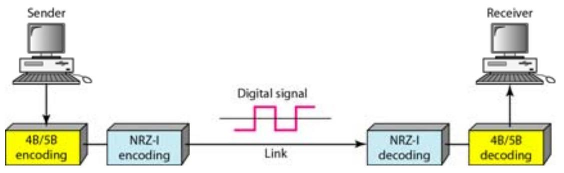
Step 2: Substitution

Step 3: Combination



Step 4: Line Coding

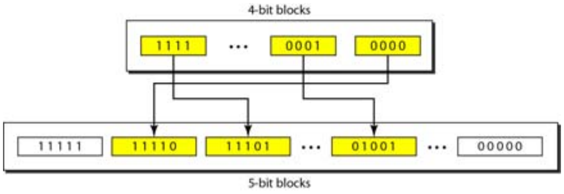
Using block coding 4B/5B with NRZ-I line coding scheme



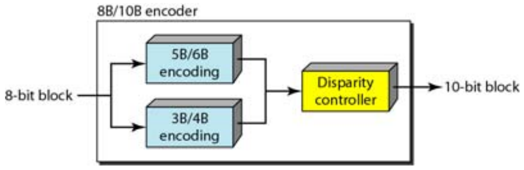
4B/5B mapping codes

Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

Substitution in 4B/5B block coding



8B/10B block encoding



Disadvantage:

Need more bandwidth.

Solution:

8B/6T

8B/6T block encoding

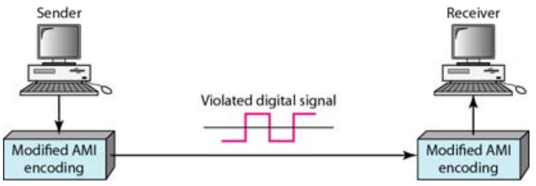
- 8 Binary to 6 Ternary
- Similar to 4B/5B in that the binary input is encoded using a table of values
- Differs from 4B/5B in that the data is not converted first into binary codes but directly into voltages

Scrambling

- The best code is the one that doesn't increase the bandwidth for synchronization and has no DC components.
- Scrambling is a technique used to create a sequence of bits that has no low frequencies, no wide bandwidth.
- It is implemented at the same time as encoding, the bit stream is created on the fly.
- It replaces sequence bits that prevent self synchronization with a violation code that is easy to recognize and removes the unfriendly bits.

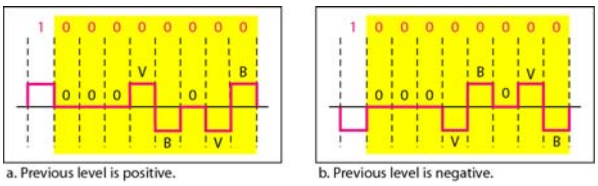
Scrambling

Figure 4.18 *AMI used with scrambling*



Bipolar 8-Zero Substitution (B8ZS)

Figure 4.19 *Two cases of B8ZS scrambling technique*



B8ZS substitutes eight consecutive zeros with 000VB0VB.

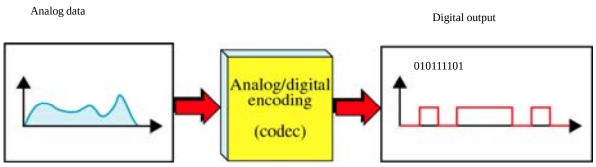
4.2 ANALOG-TO-DIGITAL Conversion

A digital signal is superior to an analog signal. The tendency today is to change an analog signal to digital data. In this section includes two techniques by pulse code modulation and Delta Modulation.
We will study only pulse code modulation.

Topics discussed in this section:

Pulse Code Modulation (PCM)
Pulse Amplitude Modulation (PAM)

Analog to Digital Encoding



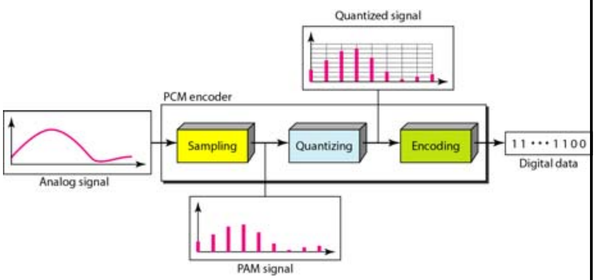
Pulse Code Modulation (PCM)

The practical implementation of PCM uses three processes:

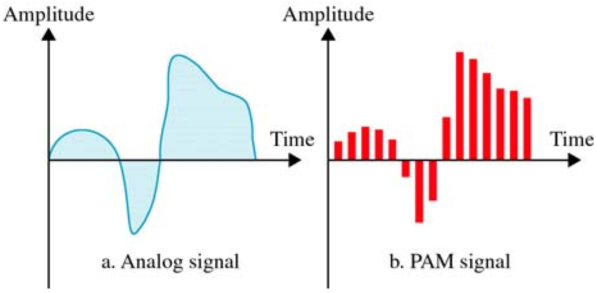
- Sampling
- Quantizing
- Binary Encoding
- Line or block coding

Pulse Code Modulation (PCM)

Figure 4.21 ***Components of PCM encoder***



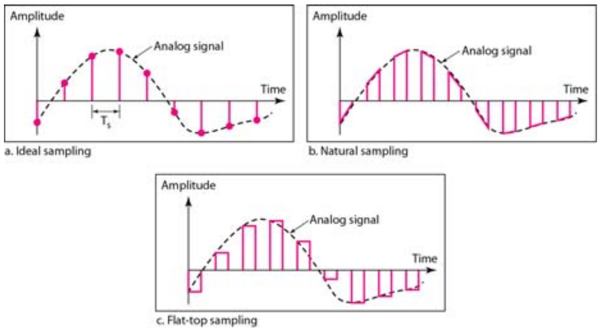
Pulse Amplitude Modulation (PAM)



Pulse Amplitude Modulation (PAM)

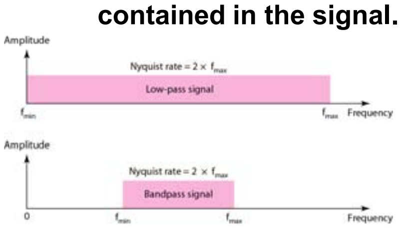
Pulse Amplitude Modulation (PAM) is not used by itself in data communication but it is the first step in Pulse Code Modulation (PCM)
PAM is a form of signal modulation where the message information is encoded in the amplitude of a series of signal pulses.
The result is an analog signal with nonintegral values.

Figure 4.22 *Three different sampling methods for PCM*



Sampling Rate

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



Nyquist sampling rate for low-pass and bandpass signals

Example 4

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

Example 5

A signal is sampled. Each sample requires at least 12 levels of precision(+0 to +5 and -0 to -5). How many bits should be sent for each sample?

Solution

4 bits is needed. 1bit for the sign and 3 bits for the value:
3 bits value = 8 levels.

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

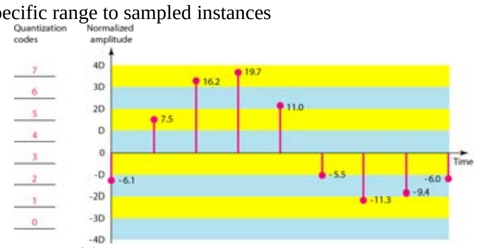
A complex bandpass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

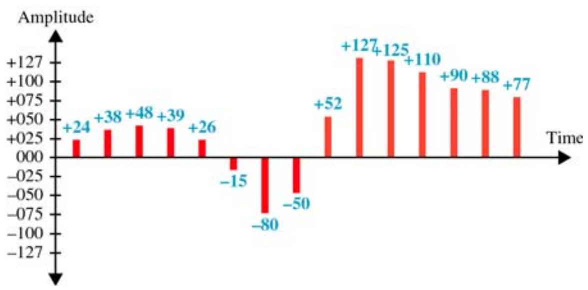
We cannot find the minimum sampling rate in this case because we do not know where the bandwidth starts or ends. We do not know the maximum frequency in the signal.

Quantization and encoding of a sampled signal

The result of PAM is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal(with real values).
These values cannot be used in the encoding process.
Quantization: is a method of assigning integer values in a specific range to sampled instances



Quantized PAM Signal

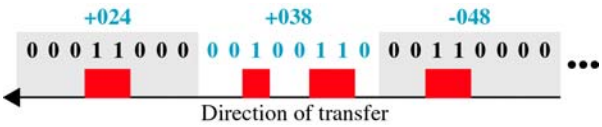


Quantizing and encoding of a sampled signal Using Sign and Magnitude

+024	00011000	–015	10001111	+125	01111101
+038	00100110	–080	11010000	+110	01101110
+048	00110000	–050	10110010	+090	01011010
+039	00100111	+052	00110110	+088	01011000
+026	00011010	+127	01111111	+077	01001101

Sign bit
+ is 0 – is 1

PCM



Bit Rate

Bit rate = Sample rate * number of bits per sample.

Example 6

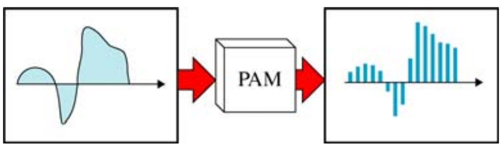
We want to digitize the human voice. What is the bit rate, assuming 8 bits per sample?

Solution

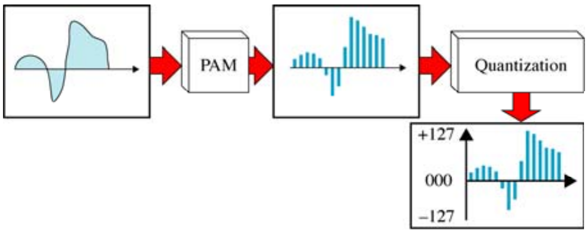
The human voice normally contains frequencies from 0 to 4 kHz. So the sampling rate and bit rate are calculated as follows:

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

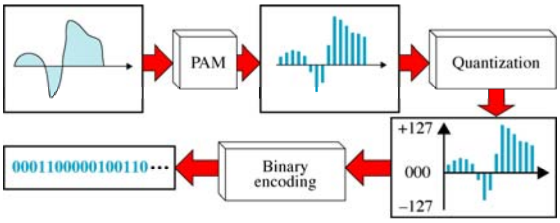
From Analog to PCM



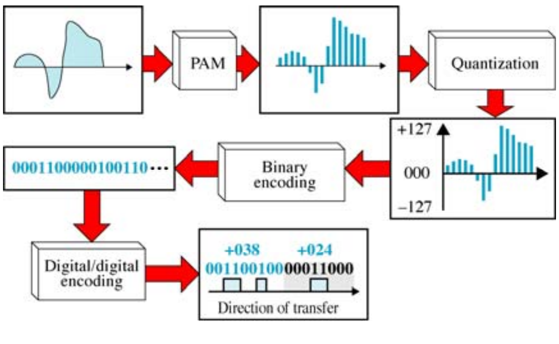
From Analog to PCM



From Analog to PCM



From Analog to PCM



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 and synchronous.

Topics discussed in this section:

Parallel Transmission
Serial Transmission

Figure 4.31 *Data transmission and modes*

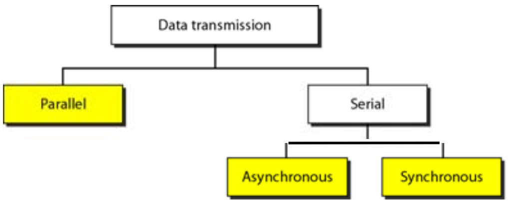


Figure 4.32 *Parallel transmission*

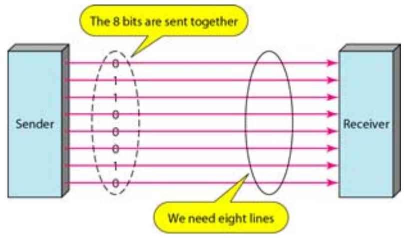
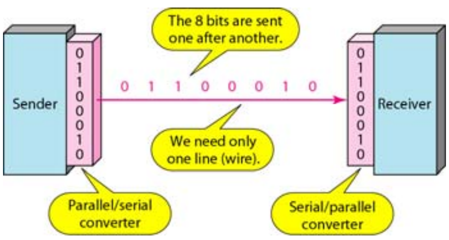


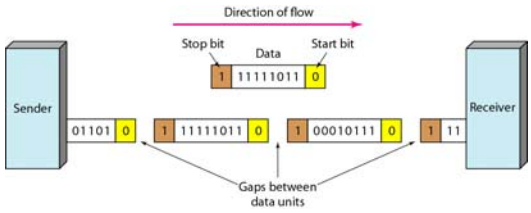
Figure 4.33 *Serial transmission*



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 each byte.

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

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 4.28 *Synchronous transmission*