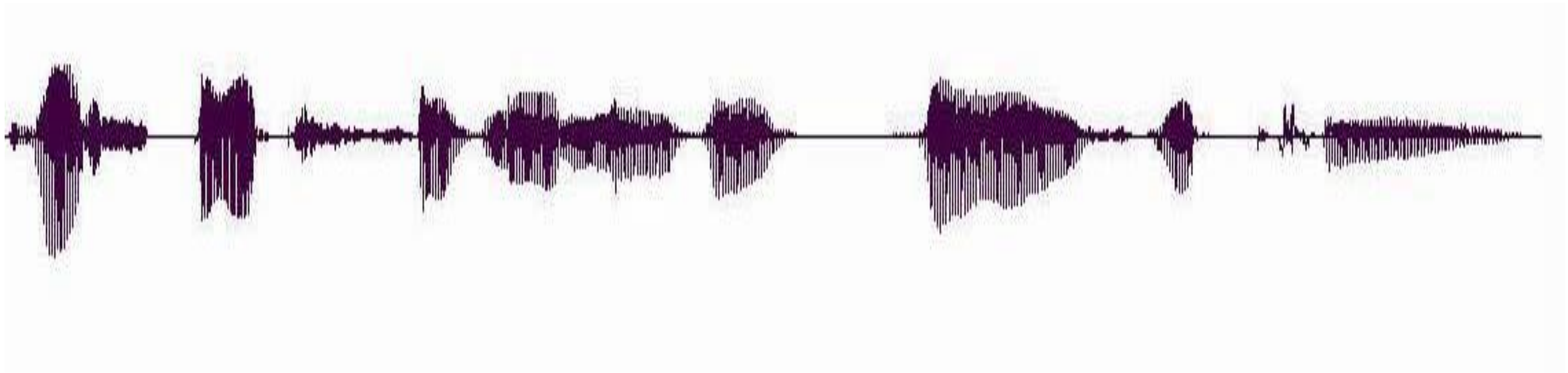
ANALOG TO DIGITAL CONVERSION

- Sometimes, we have an analog signal such as one created by a microphone (analog voice) or camera (analog videos), which are treated as analog data.
- To transmit this analog data over digital signals we need an analog to digital conversion.
- Analog data is wave form continuous stream of data whereas digital data is discrete.
- For conversion two techniques are used, **pulse code modulation and delta modulation**.

Stream Information

A real-time voice signal must be digitized & transmitted as it is produced

Analog signal level varies continuously in time



Pulse Code Modulation (PCM)

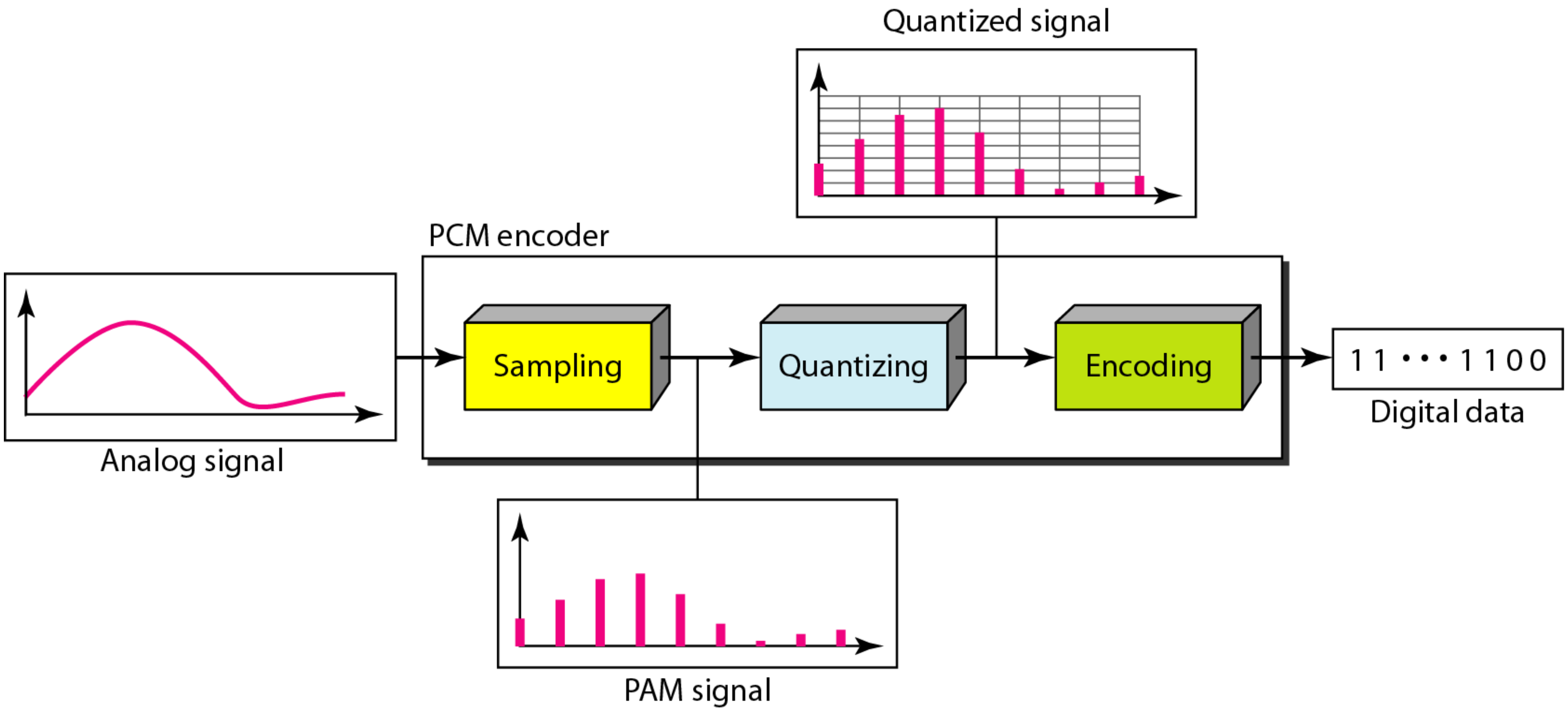
PCM consists of three steps to digitize an analog signal:

- Sampling- obtain samples of *x(t)* at uniformly spaced time intervals
- Quantization - map each sample into an approximation value of finite precision
- Binary encoding/Compression

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, i.e remove high frequency components that affect the signal shape.

Components of PCM encoder



Sampling

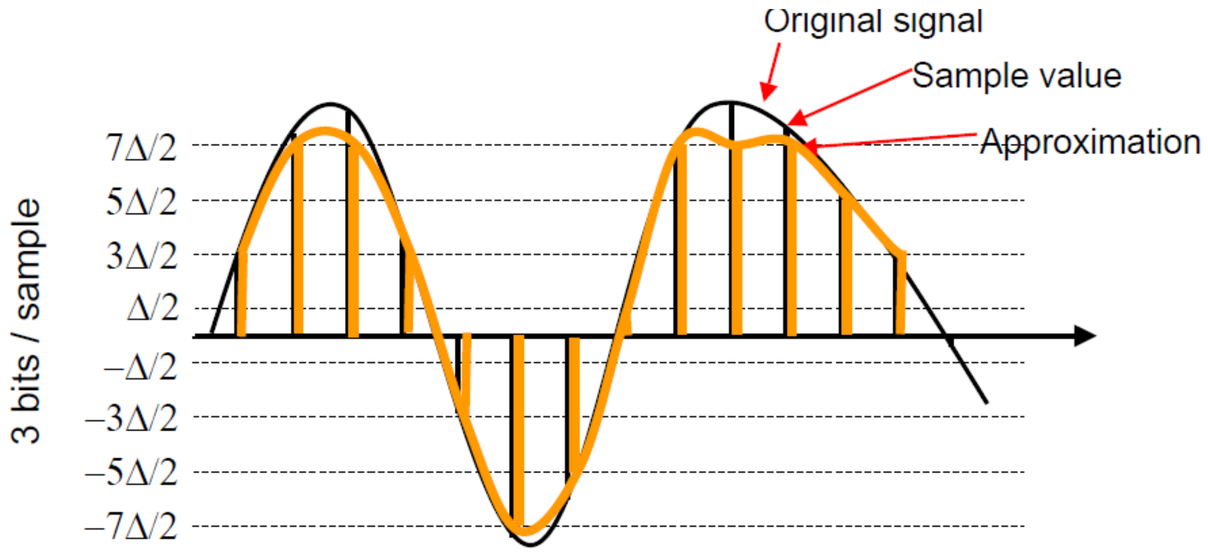
- The analog signal is sampled every *Ts* s, where *Ts* is the sample interval or period. The inverse of the sampling interval is called the sampling rate or sampling frequency.

$$f_s = 1/T_s$$

- The process is referred to as Pulse Amplitude Modulation (PAM) and the outcome is a signal with analog (non integer) values

Digitization of Analog Signal

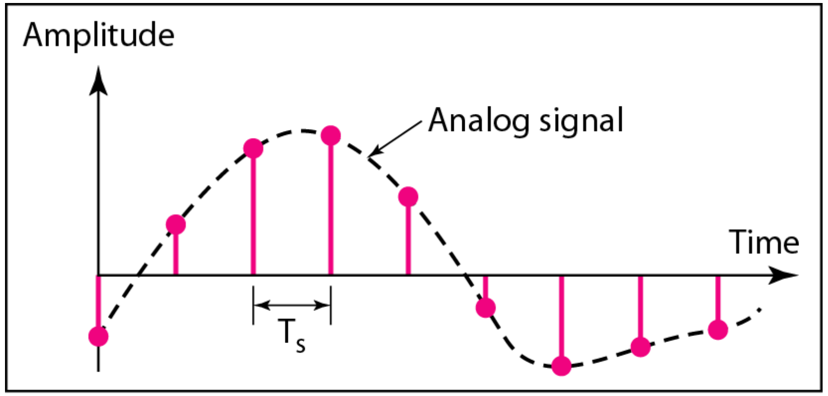
- Sample analog signal in time and amplitude
- Find closest approximation



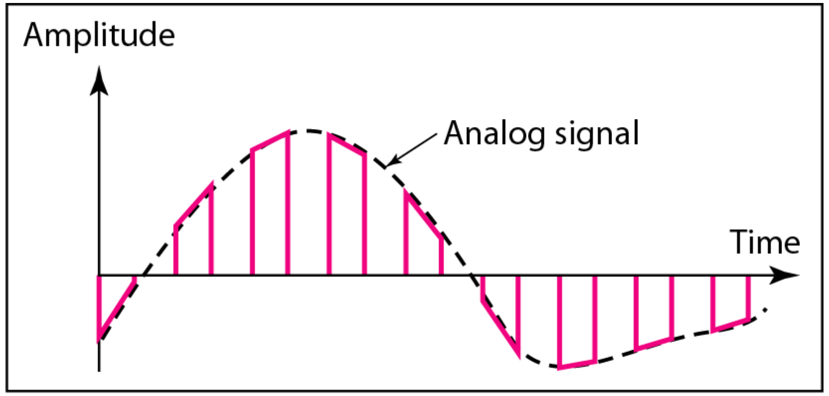
R_s = Bit rate = # bits/sample x # samples/second

- 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
- 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 the 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, called **sample and hold,**
  - creates **flat-top** samples by using a circuit.

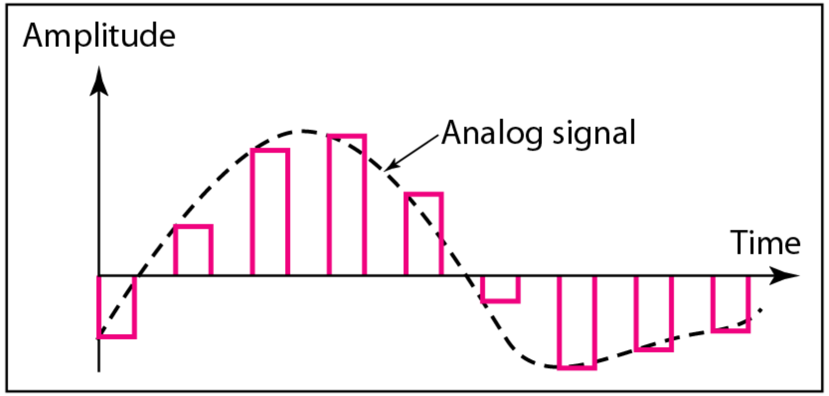
Three different sampling methods for PCM



a. Ideal sampling



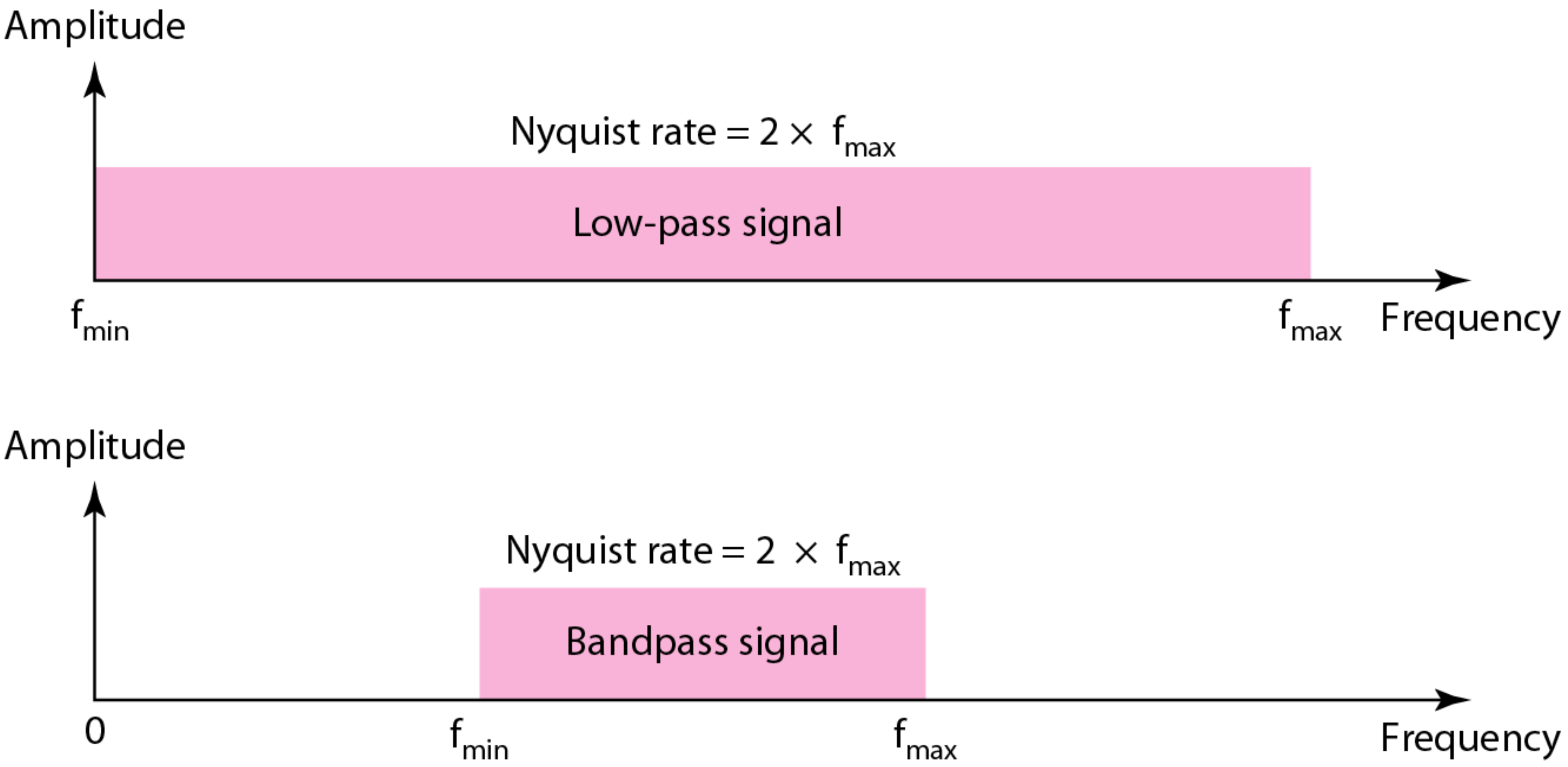
b. Natural sampling



c. Flat-top sampling

- 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



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.

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

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.

- Assume we have a voltage signal with amplitudes 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.

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 from the number of bits per sample x the sampling rate

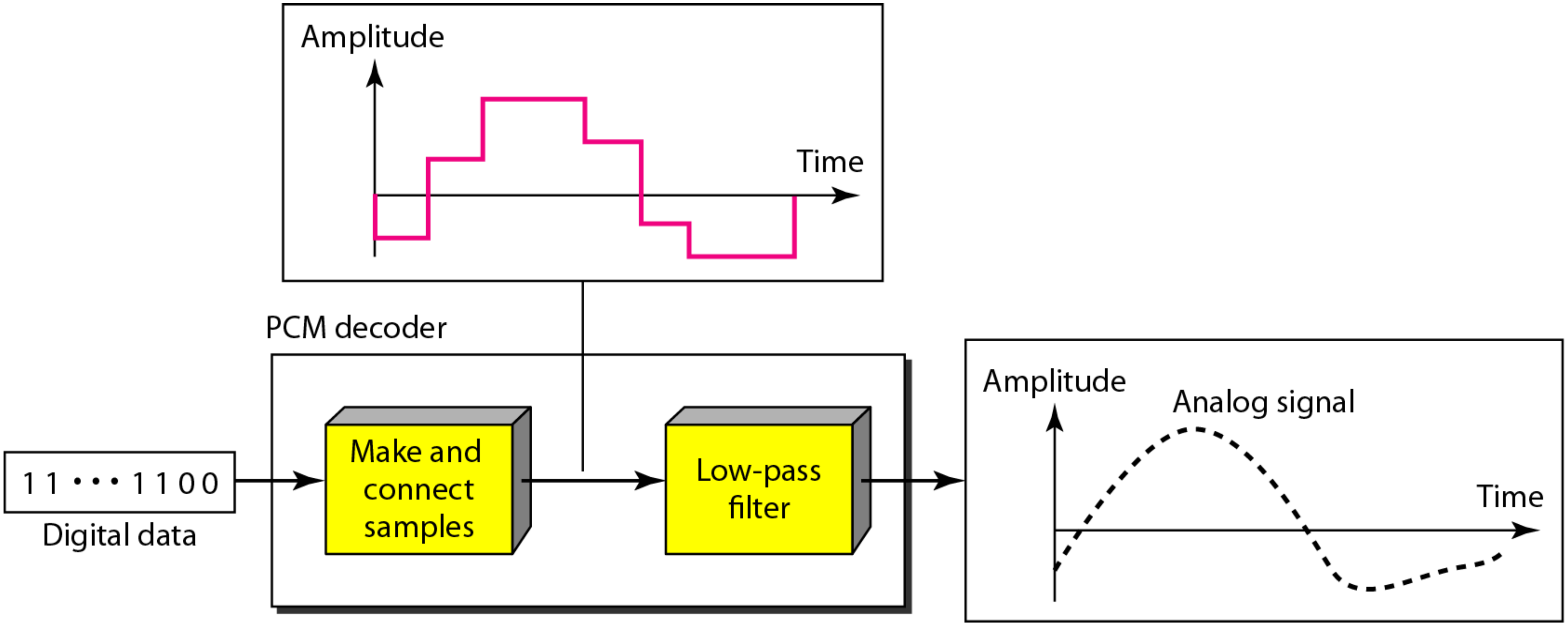
$$\text{Bit rate} = n_b \times f_s$$

- The bandwidth required to transmit this signal depends on the type of line encoding used.
- A digitized signal will always need more bandwidth than the original analog signal. Price that is to be paid for robustness and other features of digital transmission.

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.

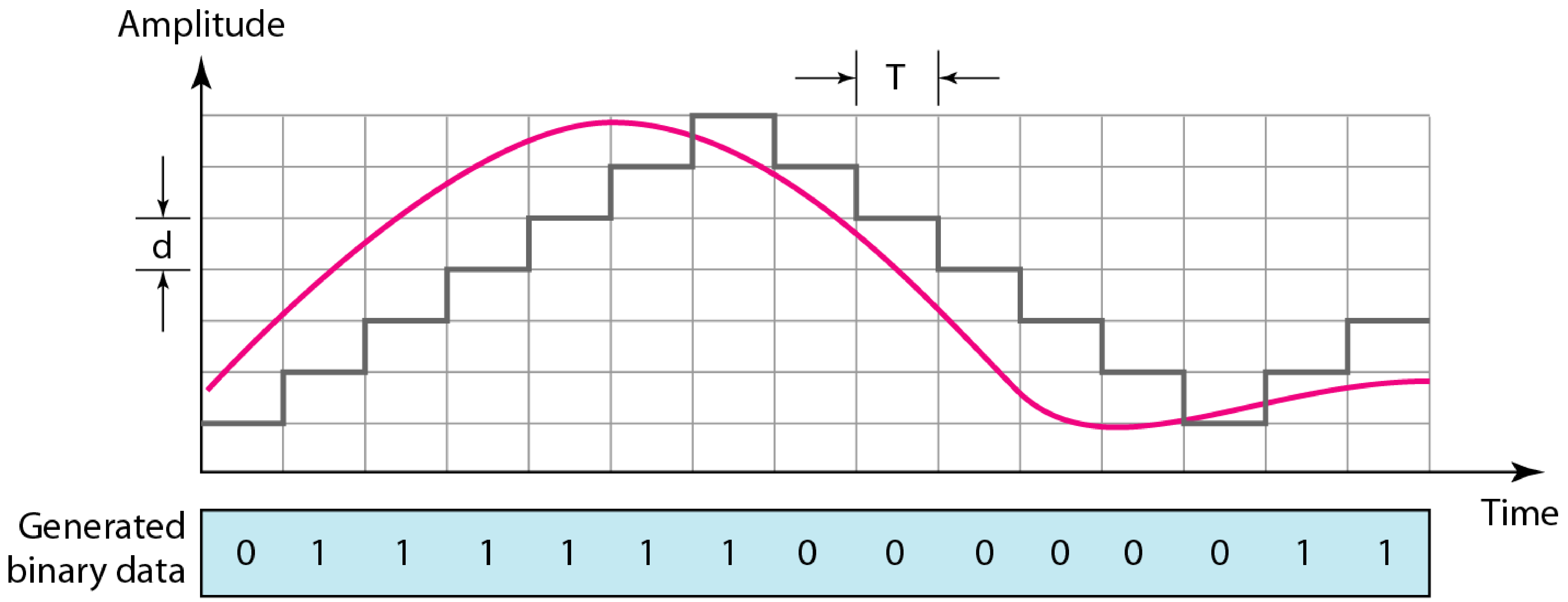
Components of a PCM decoder



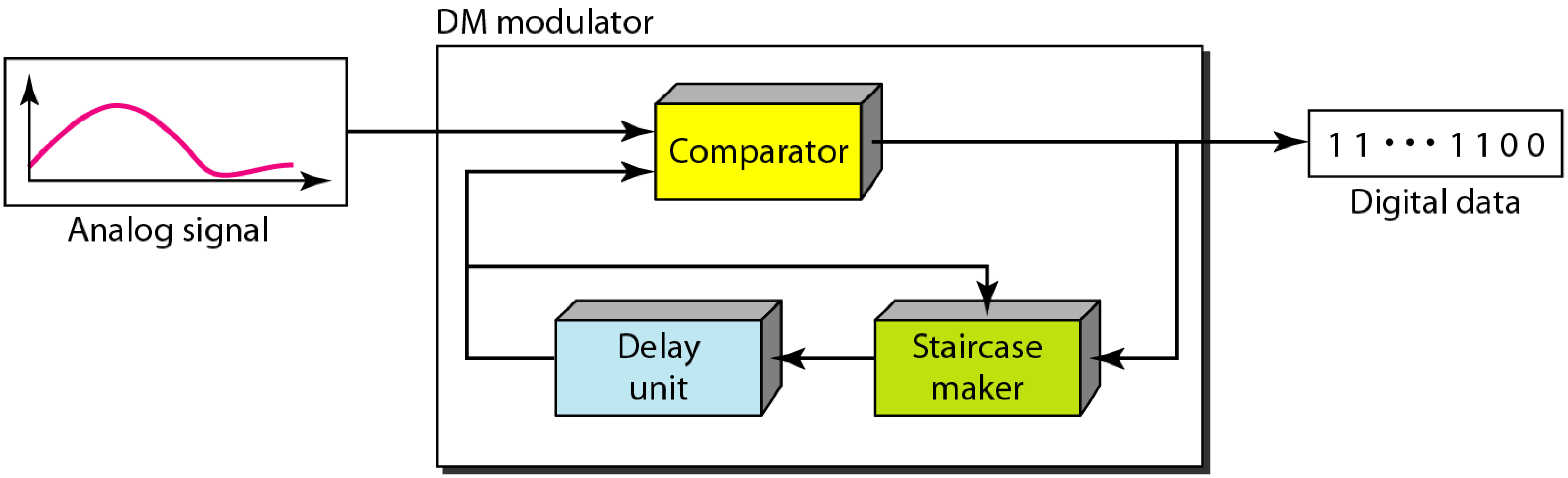
Delta Modulation

- 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.

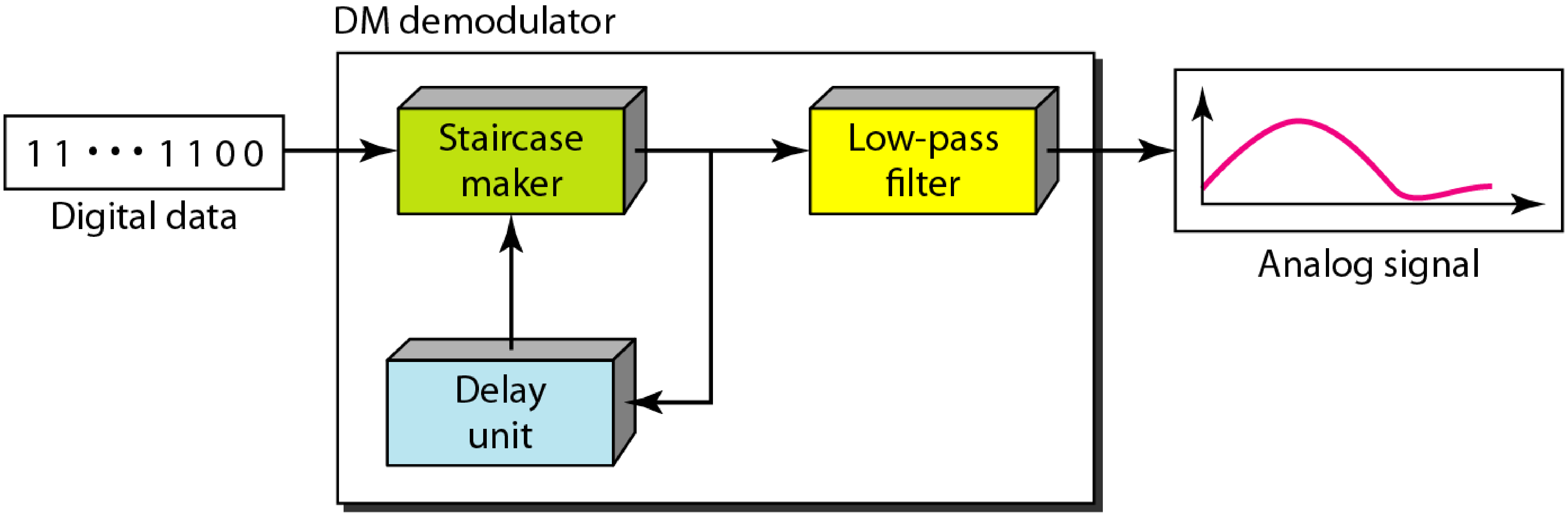
The process of delta modulation



Delta modulation components



Delta demodulation components



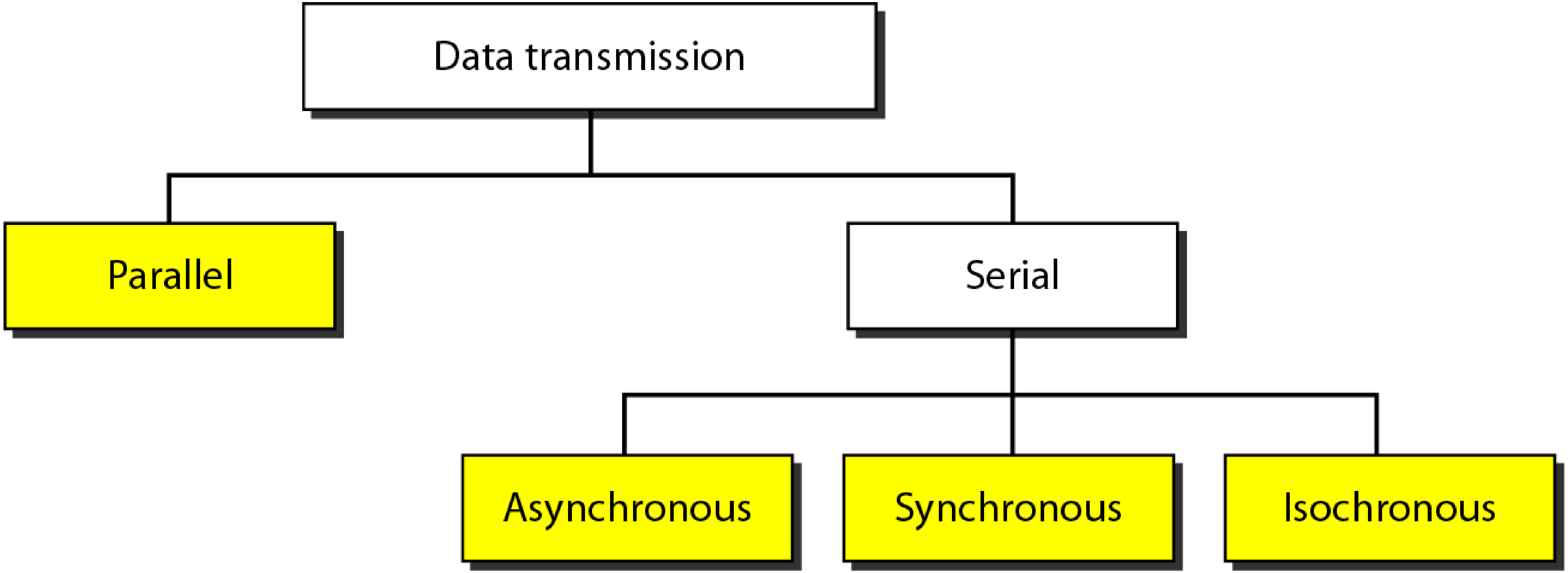
Delta PCM (DPCM)

- 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.

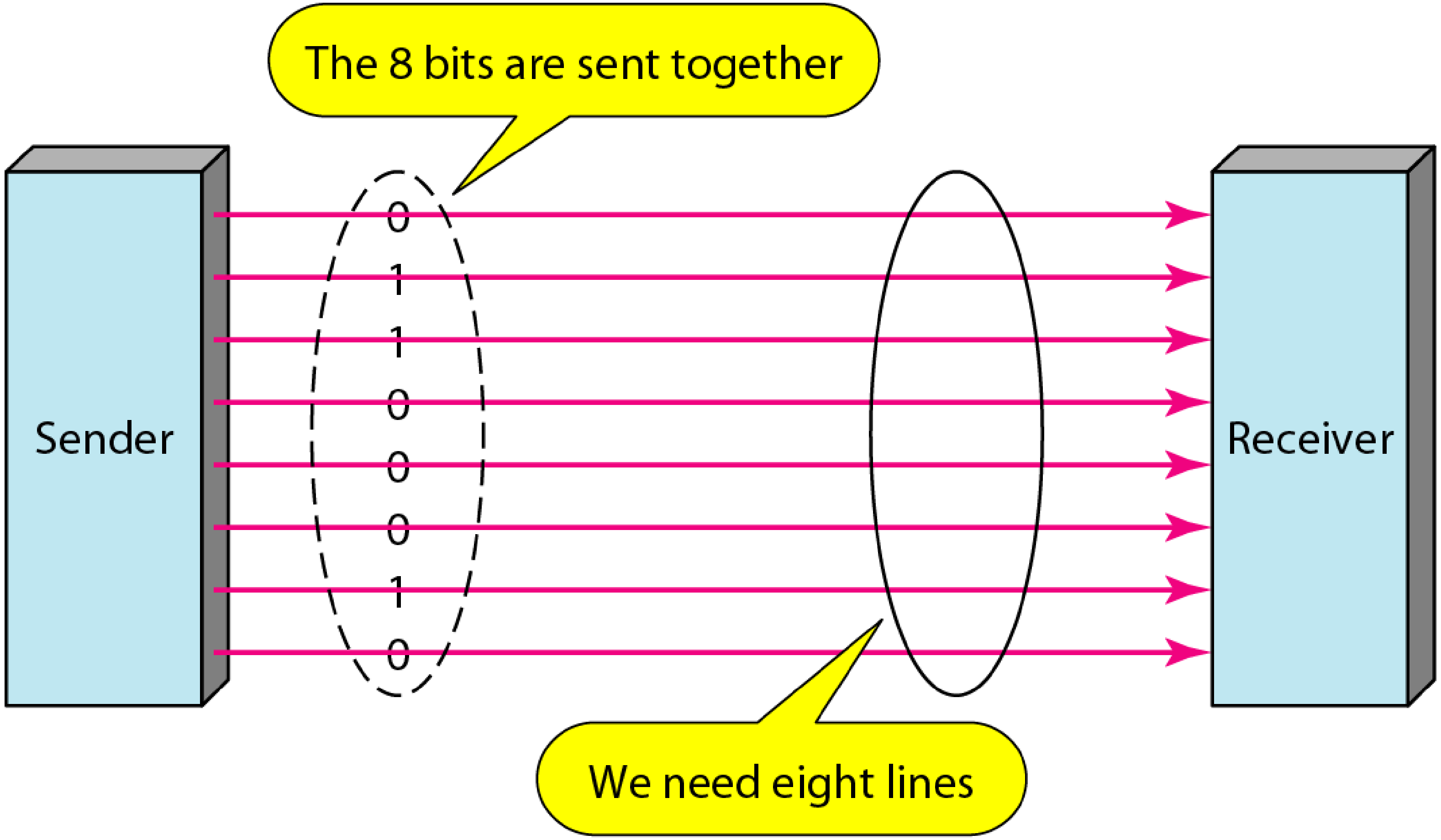
TRANSMISSION MODES

- 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
  - Isochronous.

Data transmission and modes



Parallel transmission

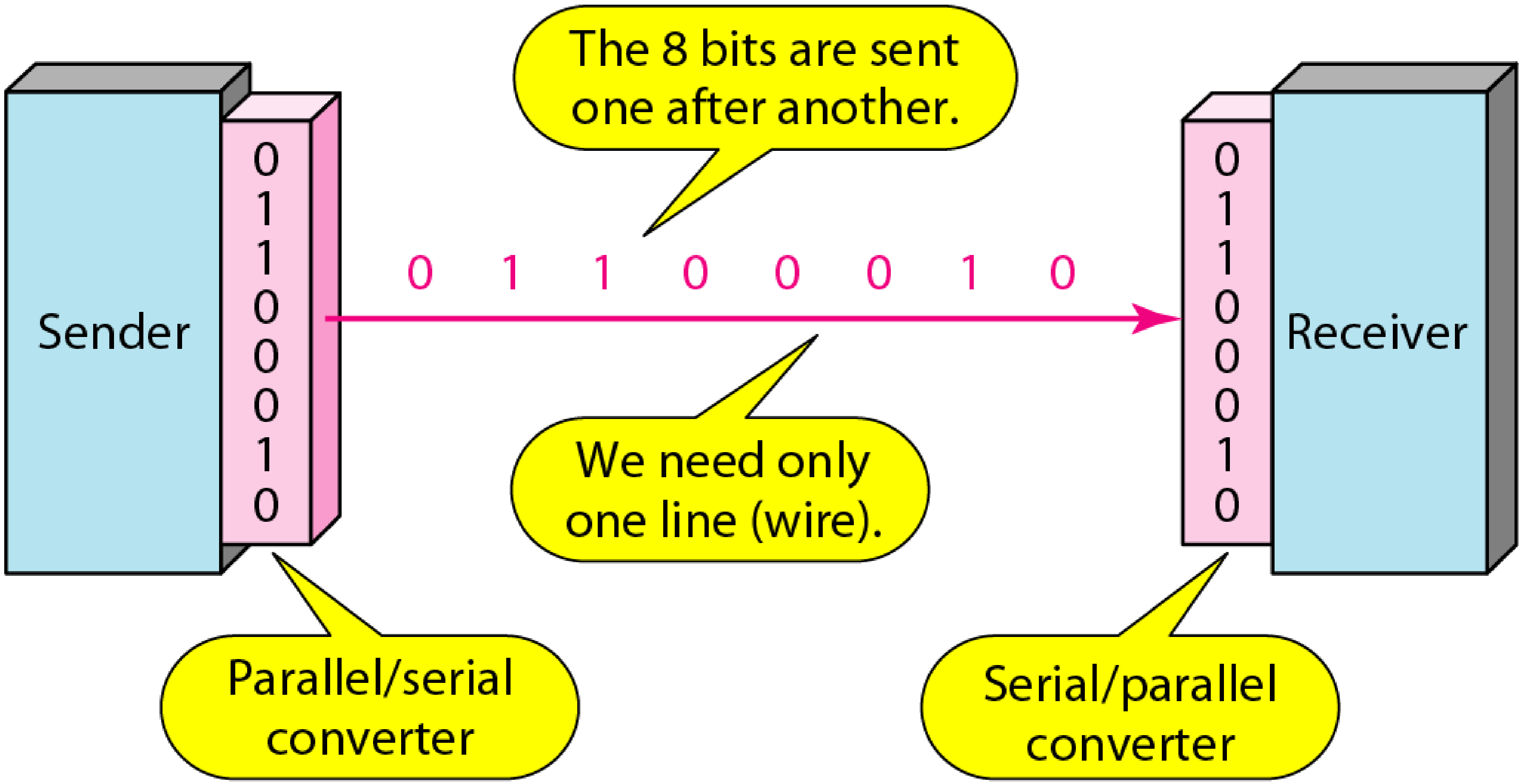


PARALLEL TRANSMISSION

- Binary data, consisting of 1s and 0s, may be organized into groups of *n* bits each.
- Computers produce and consume data in groups of bits. By grouping, we can send data *n* bits at a time instead of 1. This is called parallel transmission.
- 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.**
  - That is parallel transmission can increase the transfer speed by a factor of *n* over serial transmission.
- But there is a significant **disadvantage is cost:**
  - Parallel transmission requires *n* communication lines (wires in the example) just to transmit the data stream. This is expensive, so parallel transmission is usually limited to short distances.

Serial transmission



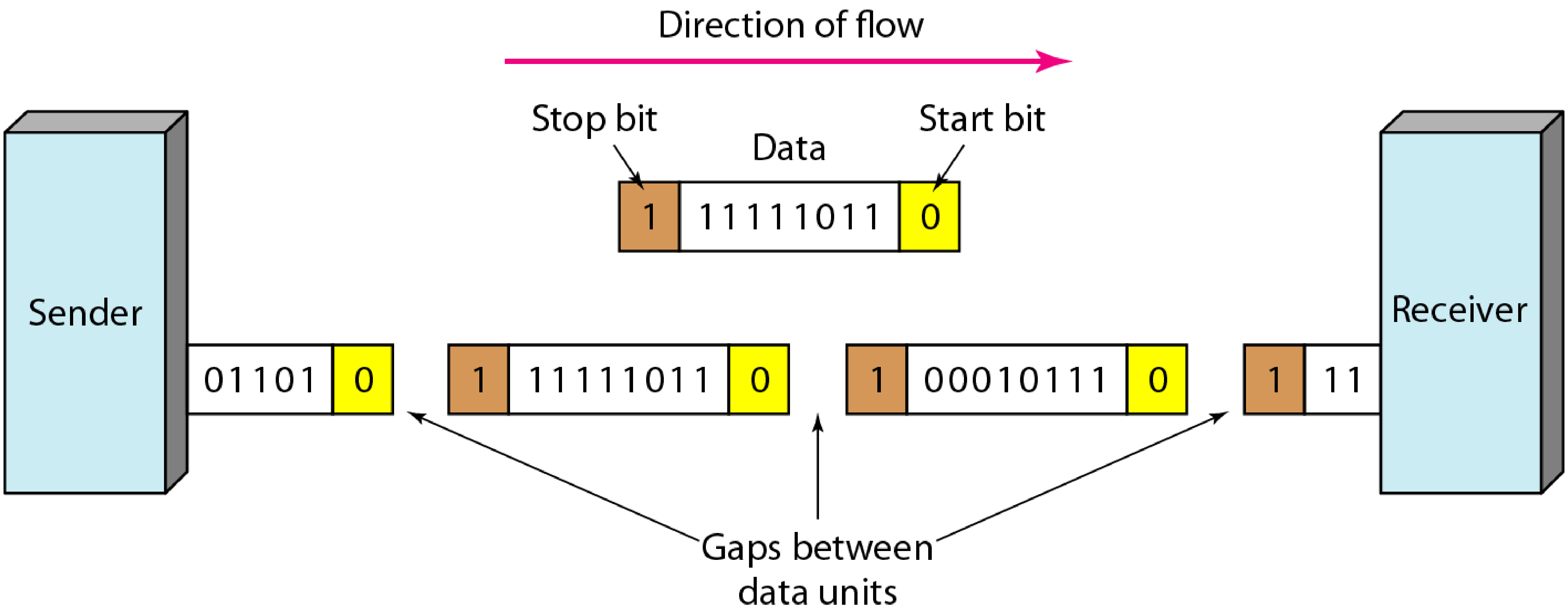
SERIAL TRANSMISSION

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

Asynchronous transmission

- Asynchronous transmission is so named because the timing of a signal is unimportant. Instead, information is received and translated by agreed upon patterns.
- Patterns are based on grouping the bit stream into bytes. Each group, usually 8 bits, is sent along the link as a unit.
- The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer. Without synchronization, the receiver cannot use timing to predict when the next group will arrive.

- To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte. This bit, usually a 0, is called the **start bit**.
- To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte. These bits, usually 1s, are called **stop bits**.
- There may be a gap between each byte.
- The start and stop bits and the gap alert the receiver to the beginning and **end of each byte** allow it to synchronize with the data stream.
- This mechanism is called ***asynchronous*** because, at the byte level, the sender and receiver do not have to be synchronized.
- But within each byte, the receiver must still be synchronized with the incoming bit stream i.e., their durations are the same.

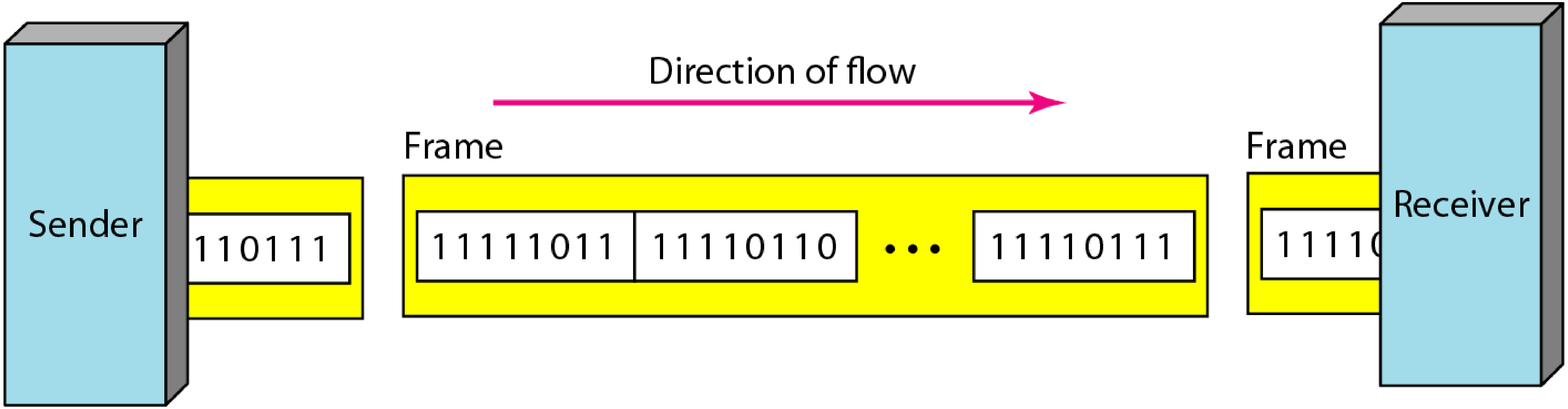

Direction of flow
Stop bit
Data
Start bit
1
11111011
0
Sender
01101
0
1
11111011
0
1
00010111
0
1
11
Receiver
Gaps between
data units

Synchronous Transmission

- In synchronous transmission, the bit stream is combined into longer "frames," which may contain multiple bytes. A frame is identified with a start and an end byte.
- Each byte 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 **other words**, data are transmitted as an unbroken string of 1s and 0s, and the receiver separates that string into the bytes, or characters, it needs to reconstruct the information.

- Byte synchronization is accomplished in the data link layer.
- Although there is no gap between characters in synchronous serial transmission, there may be uneven gaps between frames.
- The **advantage** of synchronous transmission is **speed**.
- With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link.
- Synchronous transmission is faster than asynchronous transmission. For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another.

Synchronous transmission



Isochronous Transmission

- Synchronous transmission fails in real-time audio and video, in which uneven delays between frames are not acceptable.
- For example, TV images are broadcast at the rate of 30 images per second; 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 i.e., Transmission of bits is fixed with equal gaps.