



DATA COMMUNICATION

CSE 225/233

WEEK-4, LESSON-2

DIGITAL TRANSMISSION

Analog to Digital Conversion

As we have seen, line coding can be used to convert binary data to a digital signal. Sometimes however, our data is analog, such as audio. If we want to store the audio by recording it in a computer so that we can send it digitally, we need to change it through a process called sampling.

Topics discussed in this section:

Pulse Code Modulation (PCM)

Delta Modulation (DM)

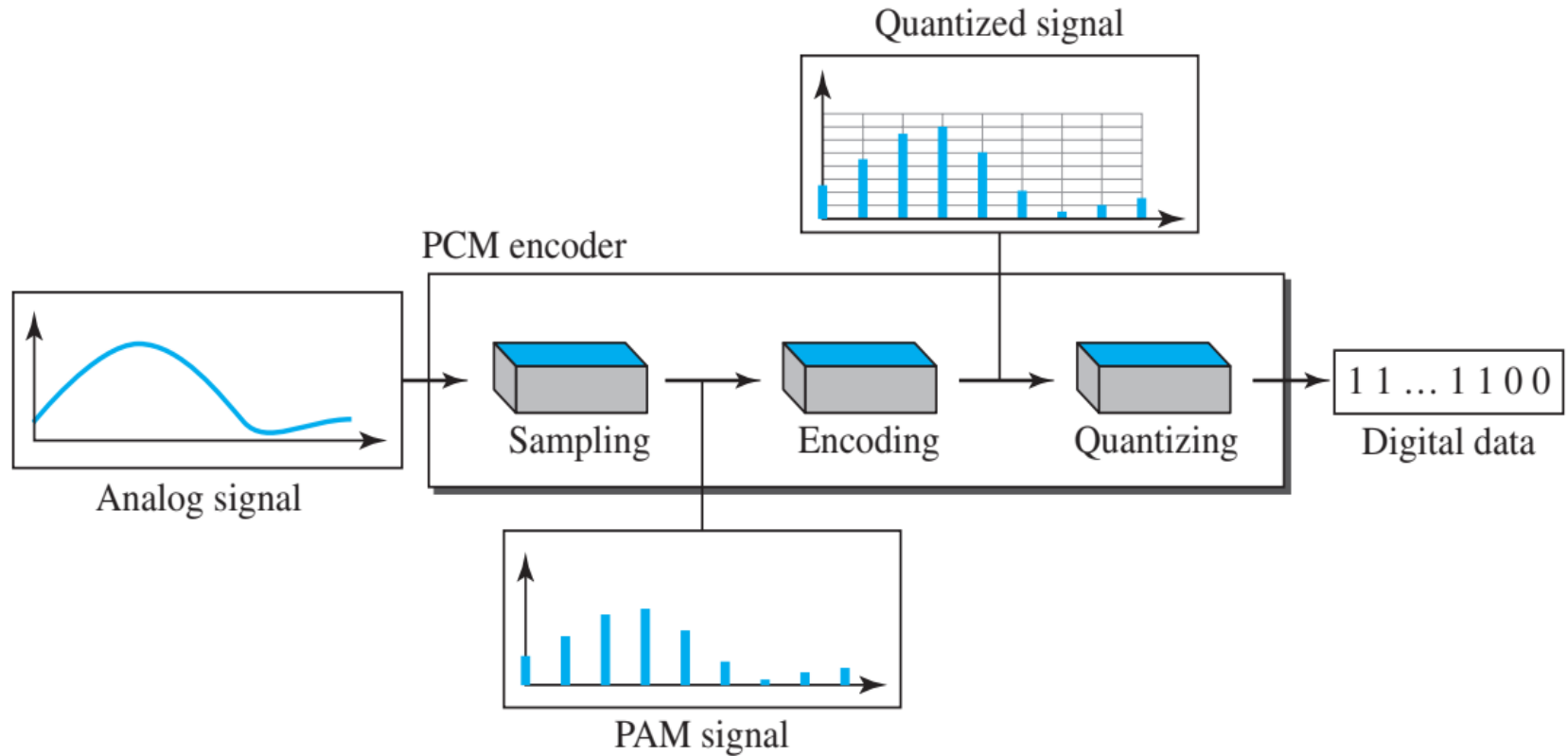
Pulse Code Modulation (PCM)

The most common technique to change an analog signal to digital data (**digitization**) is called **pulse code modulation (PCM)**. A PCM encoder has three processes

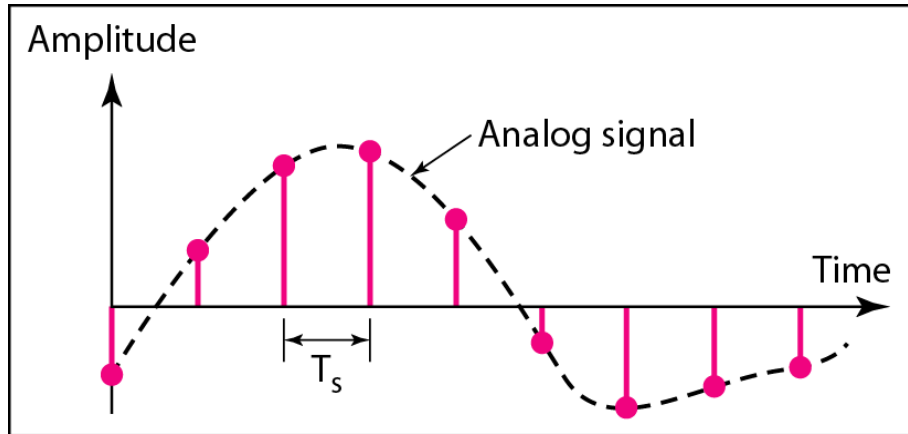
1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

Components of PCM encoder is discussed in the next slide.

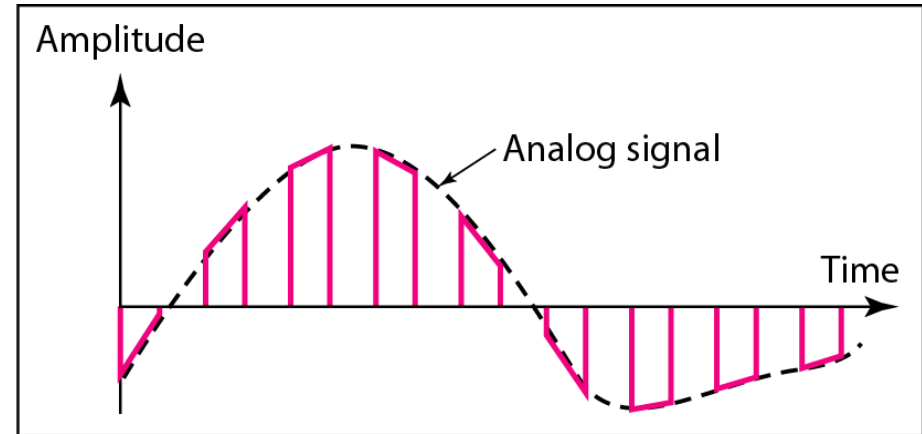
Components of PCM Encoder



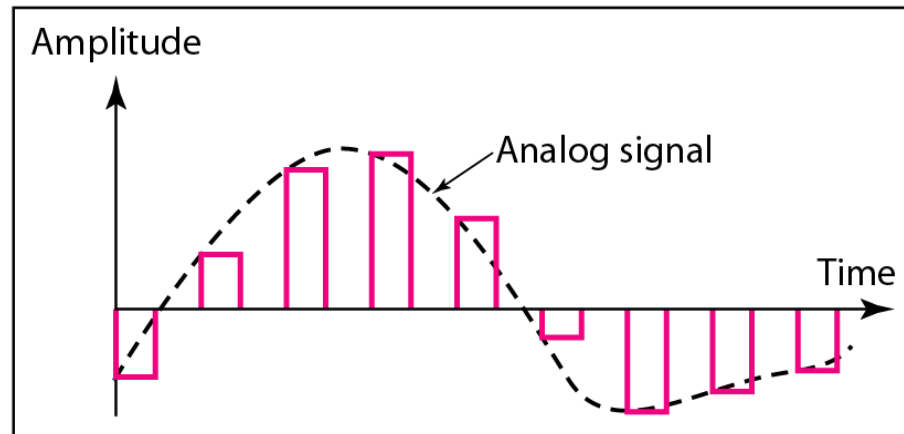
Sampling Techniques



a. Ideal sampling

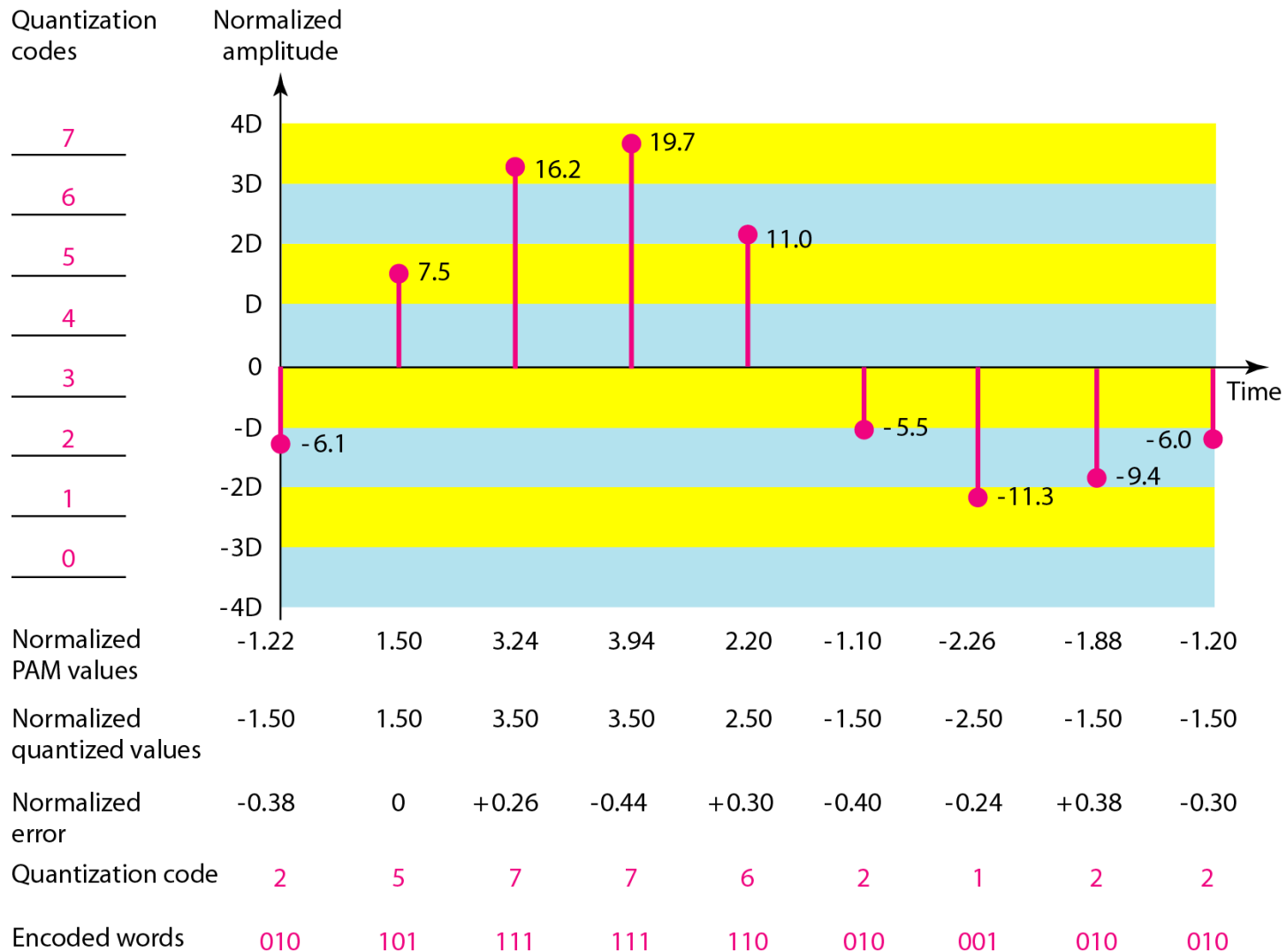


b. Natural sampling

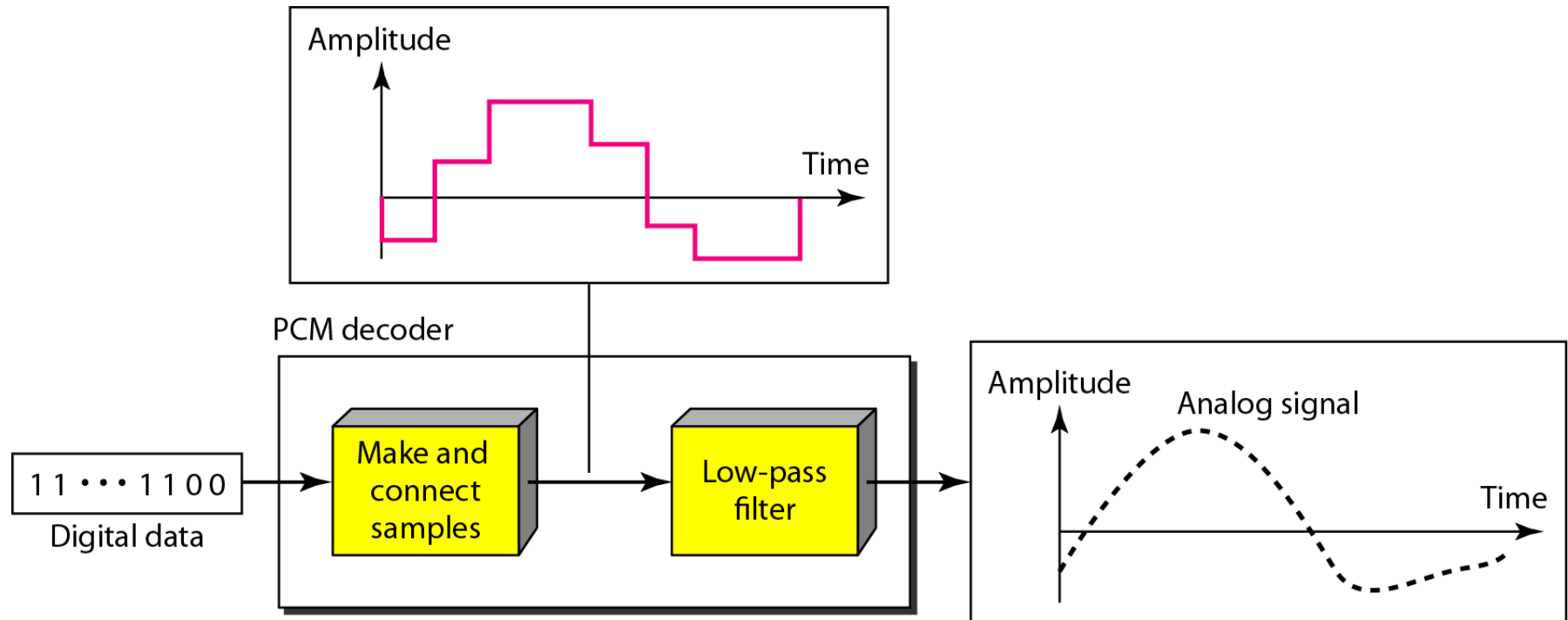


c. Flat-top sampling

Quantization and encoding of a sampled signal

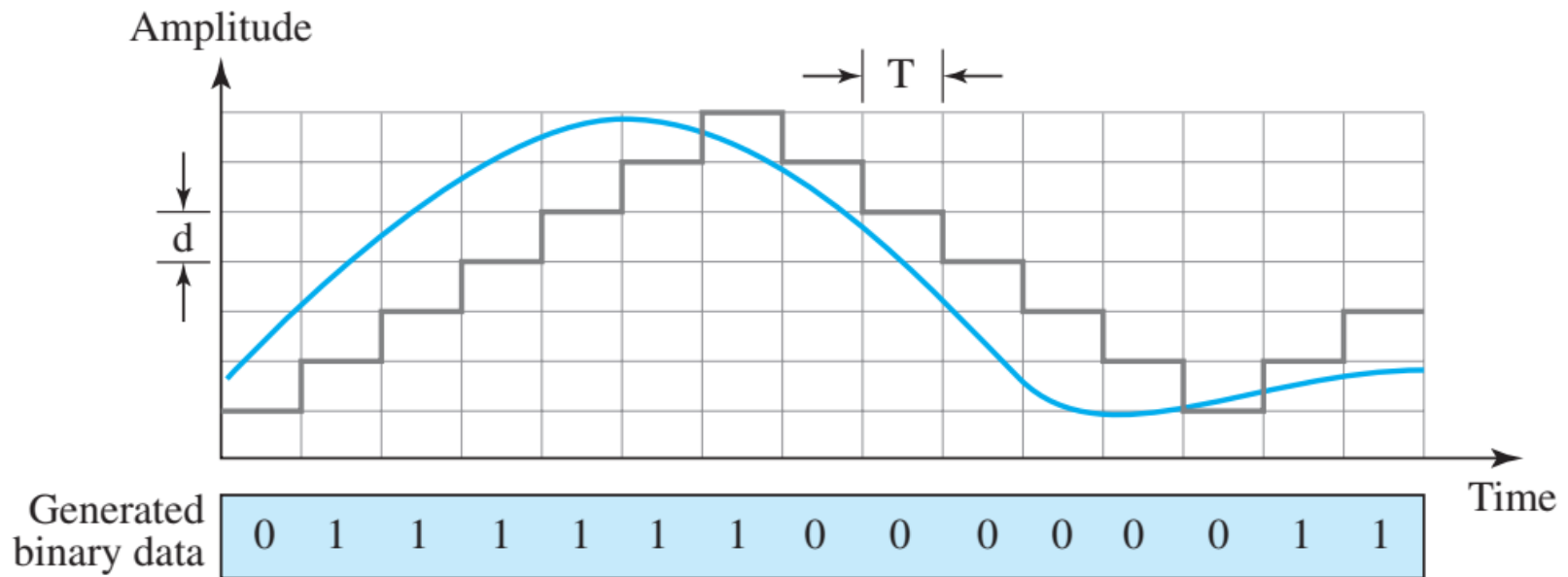


Components of PCM Decoder

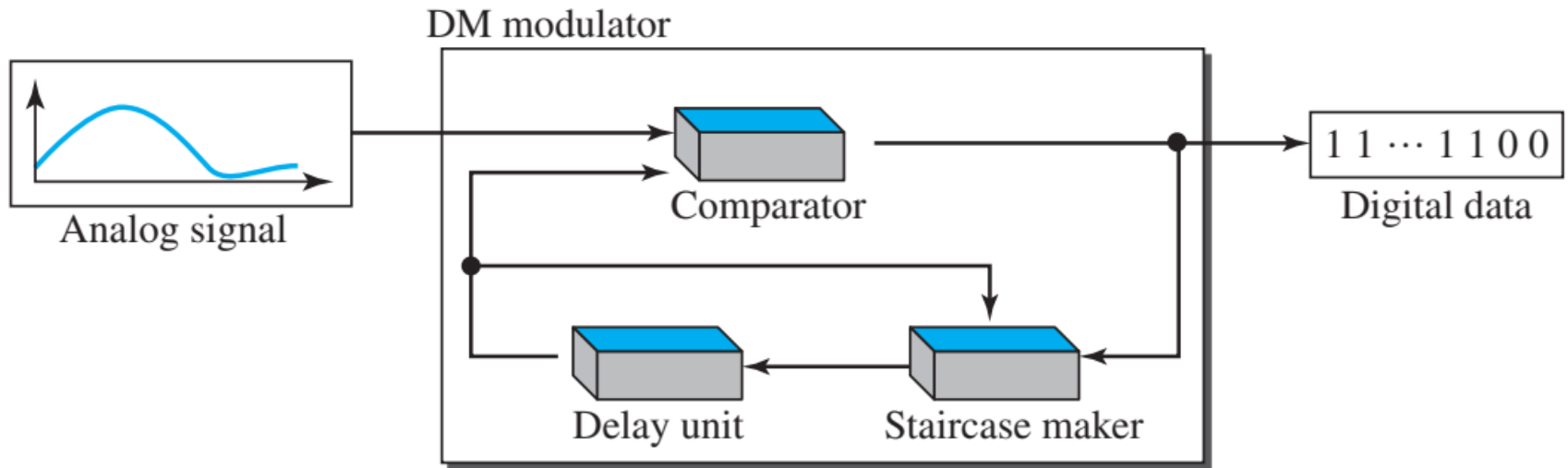


Delta Modulation (DM)

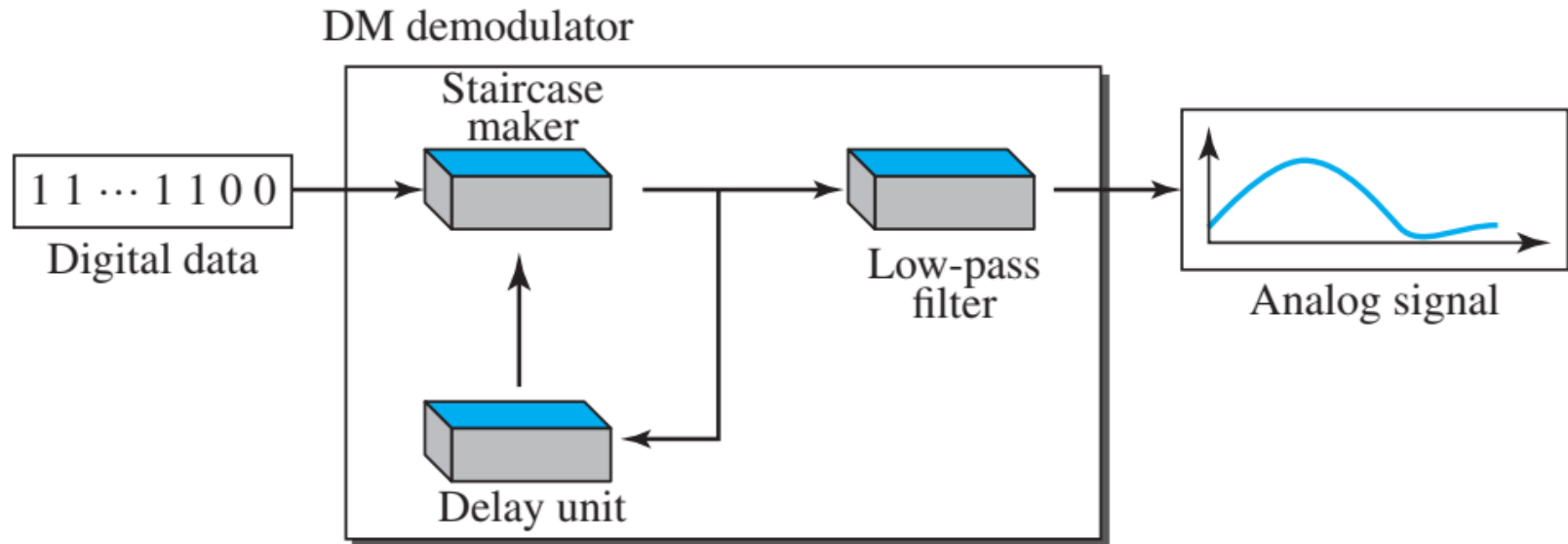
PCM is a very complex technique. Other techniques have been developed to reduce the complexity of PCM. The simplest is **delta modulation**. PCM finds the value of the signal amplitude for each sample; DM finds the change from the previous sample. Following figure shows the process. Note that there are no code words here; bits are sent one after another.



Components of Delta Modulator



Components of Delta Demodulator



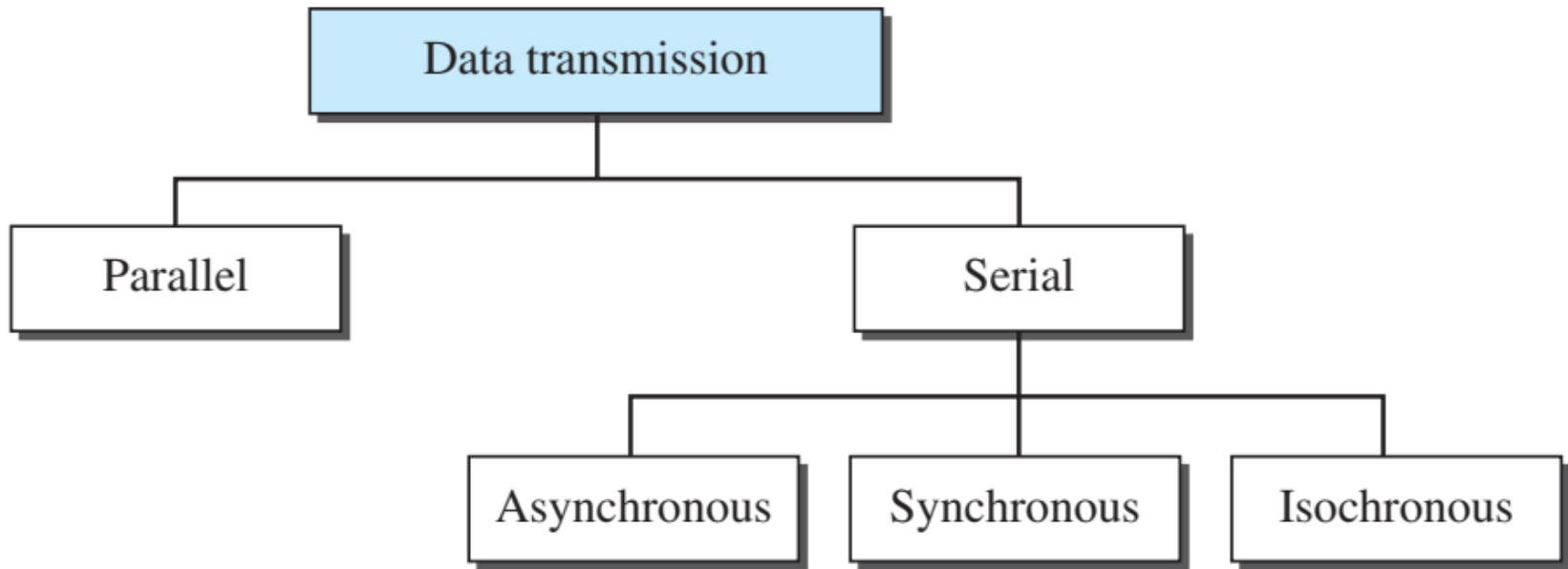
Transmission Modes

Of primary concern when we are considering the transmission of data from one device to another is the wiring, and of primary concern when we are considering the wiring is the data stream. Do we send 1 bit at a time; or do we group bits into larger groups and, if so, how? 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.

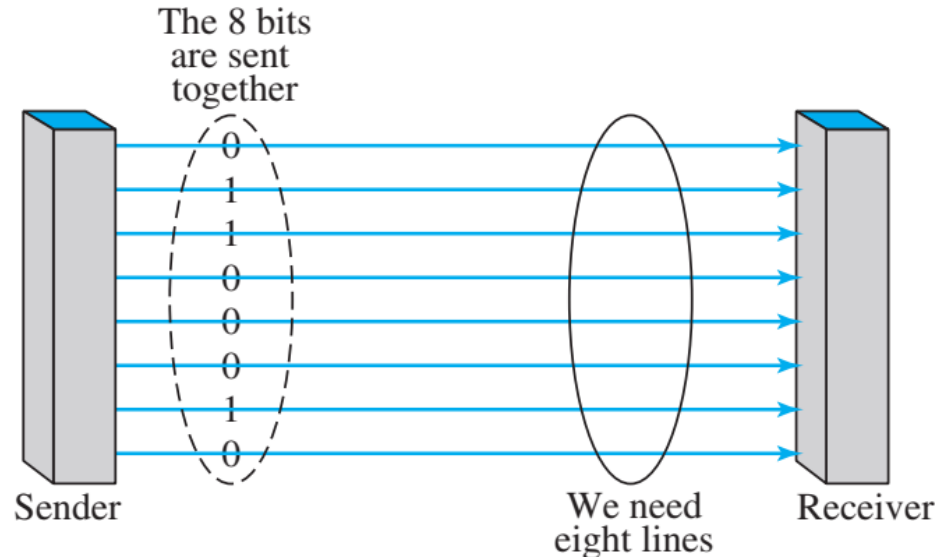
In *serial mode*, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.

Data transmission modes



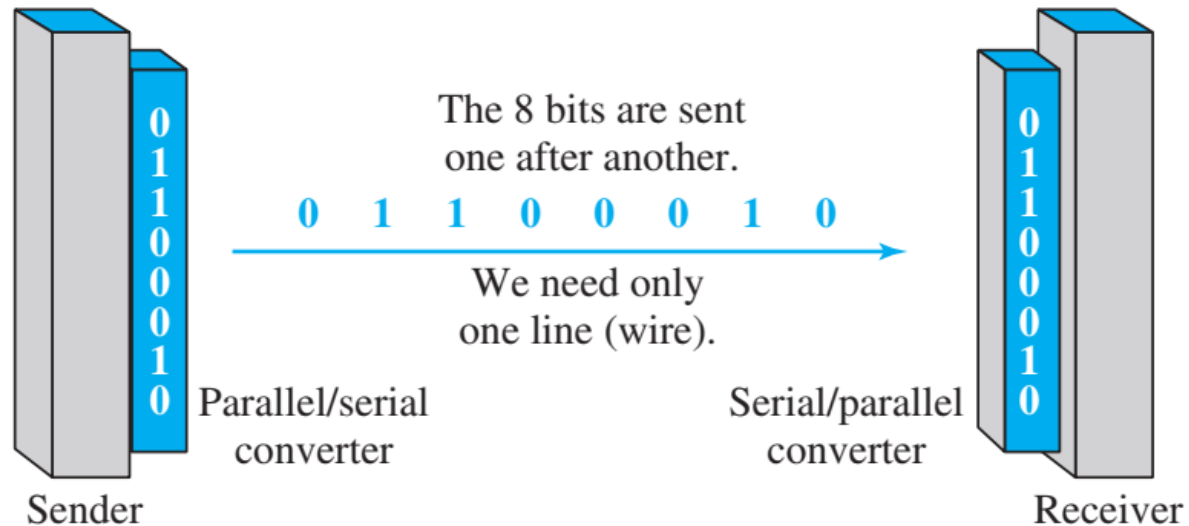
Parallel Transmission

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 much as we conceive of and use spoken language in the form of words rather than letters. By grouping, we can send data n bits at a time instead of 1. This is called ***parallel transmission***.



Serial Transmission

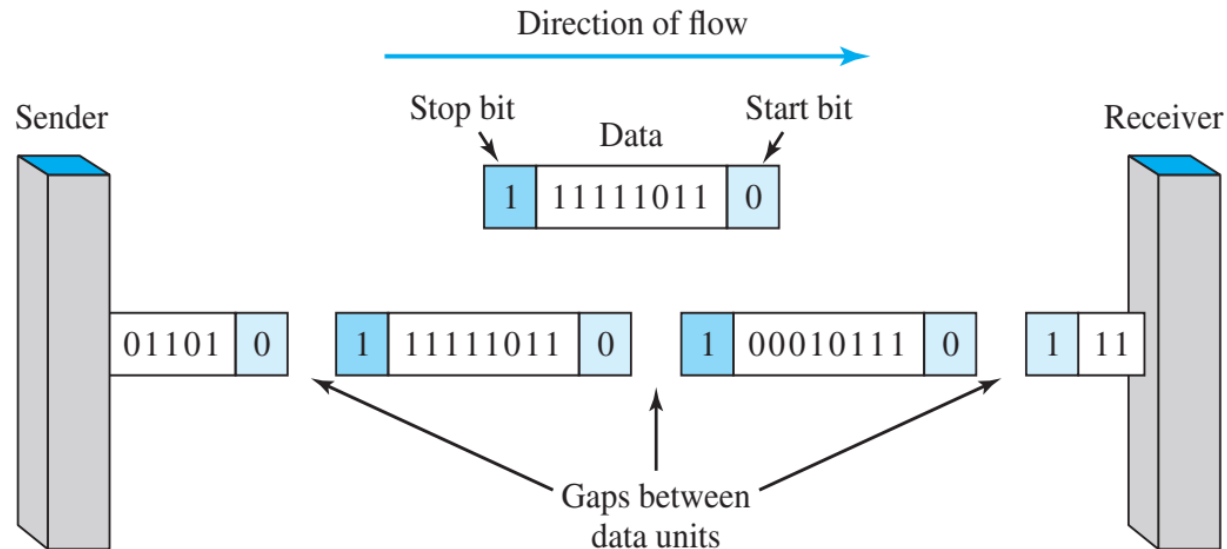
In **serial transmission** one bit follows another, so we need only one communication channel rather than n to transmit data between two communicating devices. The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission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



Asynchronous Serial Transmission

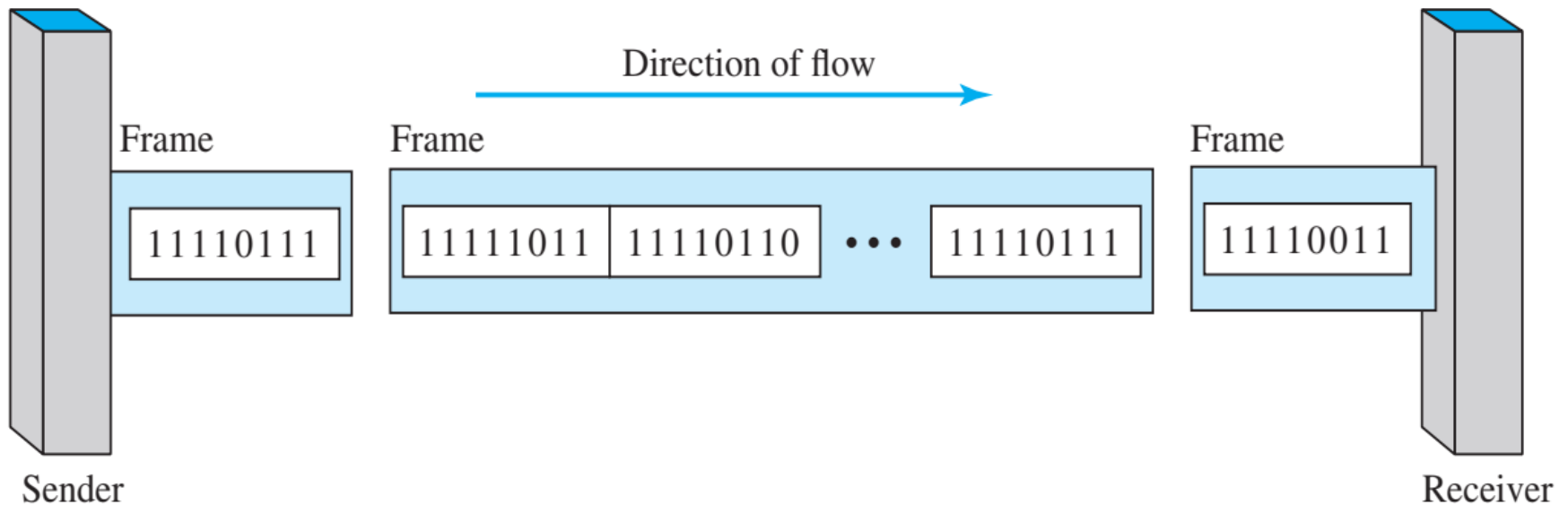
Asynchronous here means “asynchronous at the byte level,” but the bits are still synchronized; their durations are the same.

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.



Synchronous Serial 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.



Isochronous Serial Transmission

In real-time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails. 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.

