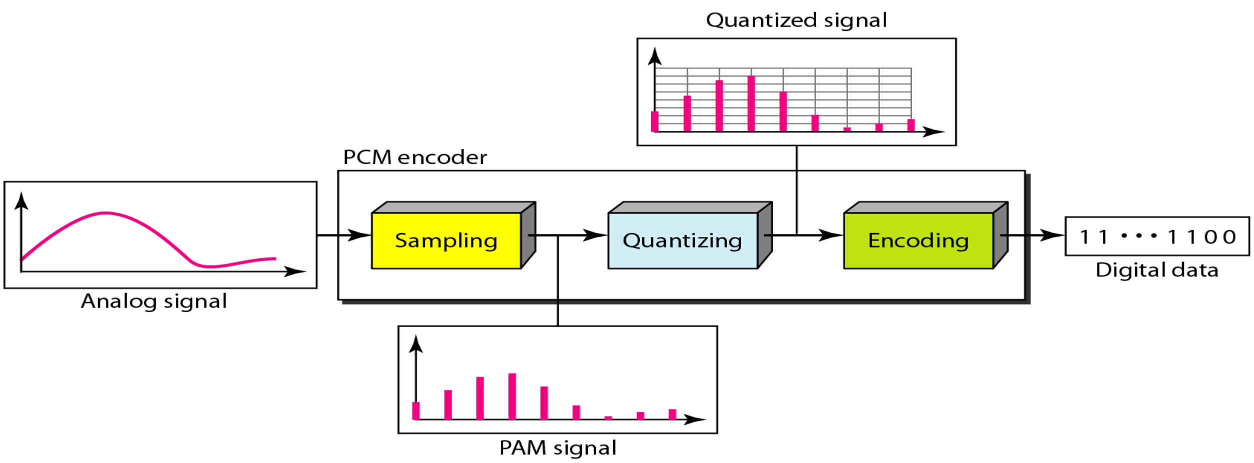
ANALOG-TO-DIGITAL CONVERSION

- ❑ A digital signal is superior to an analog signal.
- ❑ The tendency today is to change an analog signal to digital data.

Pulse Code Modulation (PCM)

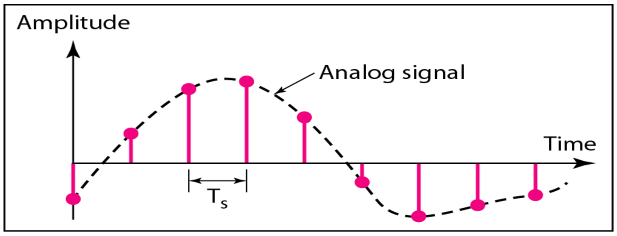
The most common technique to change an analog signal to digital data **digitization**) is called **pulse code modulation (PCM).** A PCM encoder has three processes, as shown in Figure.



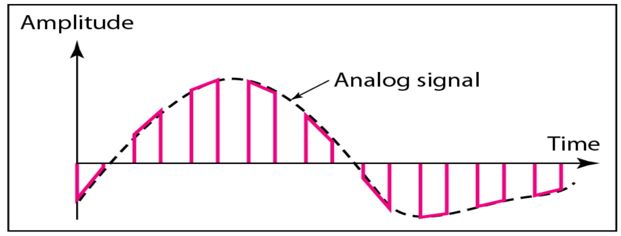
1. The analog signal is **sampled**.
2. The sampled signal is **quantized**.
3. The quantized values are **encoded** as streams of bits.

Sampling

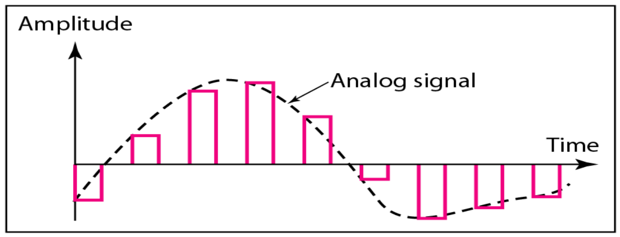
The first step in PCM is **sampling.** The analog signal is sampled every *Ts* s, where *Ts* is the sample interval or period. The inverse of the sampling interval is called the ***sampling rate*** or ***sampling frequency*** and denoted by *fs*, where *fs* = 1/*Ts*. There are three sampling methods—ideal, natural, and flat-top—as shown in Figure. 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,*** however, creates flat-top samples by using a circuit. The sampling process is sometimes referred to as **pulse amplitude modulation (PAM).** We need to remember, however, that the result is still an analog signal with nonintegral values.



a. Ideal sampling



b. Natural sampling

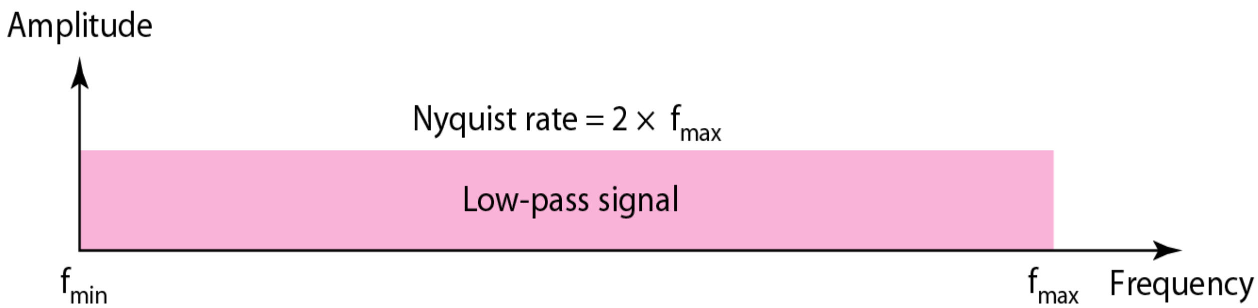


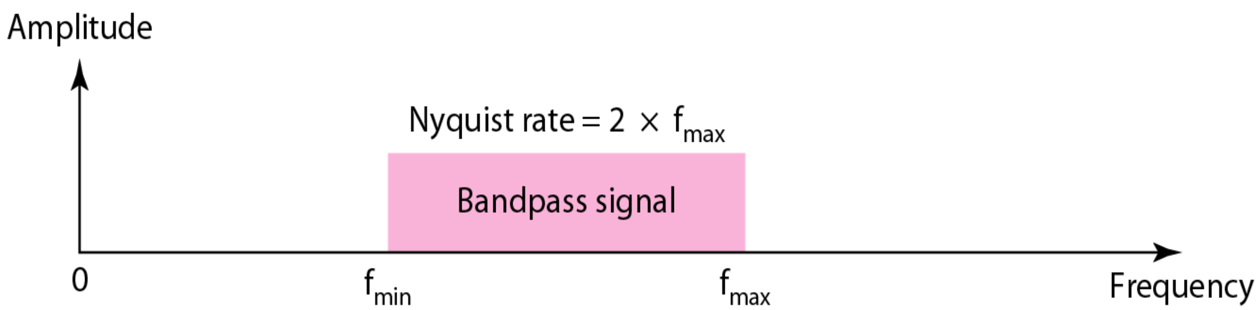
c. Flat-top sampling

Sampling Rate

According to the **Nyquist theorem,** to reproduce the original analog signal, one necessary condition is that the *sampling rate* be at least twice the highest frequency in the original signal.

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





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).
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.

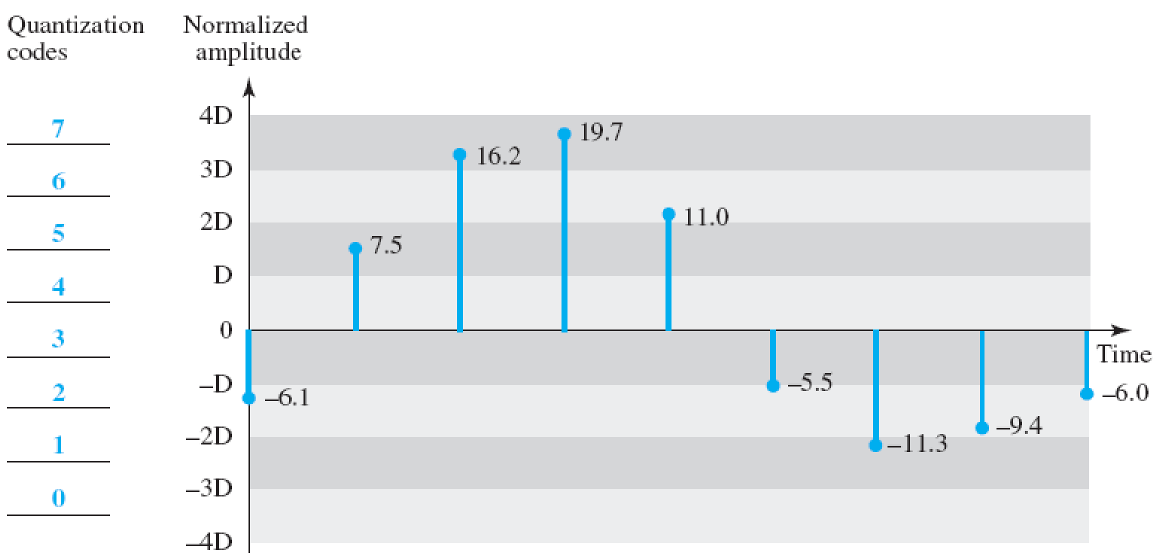
As a simple example, assume that we have a sampled signal and the sample amplitudes are between −20 and +20 V. We decide to have eight levels (*L* = 8). This means that Δ = 5 V.

In figure, We have shown only nine samples using ideal sampling (for simplicity). The value at the top of each sample in the graph shows the actual amplitude. In the chart, the first row is the normalized value for each sample (actual amplitude/Δ). The quantization process selects the quantization value from the middle of each zone. This means that the normalized quantized values (second row) are different from the normalized amplitudes. The difference is called the *normalized error* (third row). The fourth row is the quantization code for each sample based on the quantization levels at the left of the graph. The encoded words (fifth row) are the final products of the conversion.

Quantization Levels

In the previous example, we showed eight quantization levels. The choice of *L*, the number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal. If the amplitude of a signal fluctuates between two values only, we need only two levels; if the signal, like voice, has many amplitude values, we need more quantization levels. In audio digitizing, *L* is normally chosen to be 256; in video it is normally thousands. Choosing lower values of *L* increases the quantization error if there is a lot of fluctuation in the signal.

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

Quantization Error

One important issue is the error created in the quantization process. (Later, we will see how this affects high-speed modems.) Quantization is an approximation process. The input values to the quantizer are the real values; the output values are the approximated values. The output values are chosen to be the middle value in the zone. If the input value is also at the middle of the zone, there is no quantization error; otherwise, there is an error. In the previous example, the normalized amplitude of the third sample is 3.24, but the normalized quantized value is 3.50. This means that there is an error of +0.26. The value of the error for any sample is less than Δ/2. In other words, we have $-\Delta/2 \leq error \leq \Delta/2$.

The quantization error changes the signal-to-noise ratio of the signal, which in turn reduces the upper limit capacity according to Shannon. It can be proven that the contribution of the **quantization error** to the SNRdB of the signal depends on the number of quantization levels *L*, or the bits per sample *nb*, as shown in the following formula:

$$\text{SNR}_{\text{dB}} = 6.02 n_b + 1.76 \quad \text{dB}$$

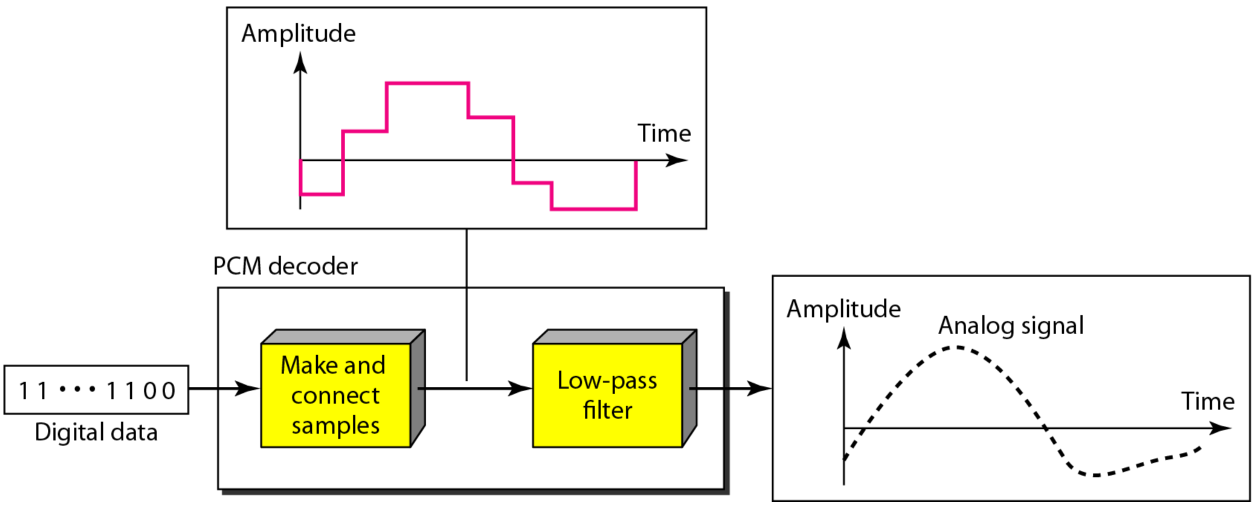
Encoding

The last step in PCM is encoding. After each sample is quantized and the number of bits per sample is decided, each sample can be changed to an *nb*-bit code word. In Figure 4.26 the encoded words are shown in the last row. A quantization code of 2 is encoded as 010; 5 is encoded as 101; and so on. Note that the number of bits for each sample is determined from the number of quantization levels. If the number of quantization levels is *L*, the number of bits is *nb* = log2 *L*. In our example *L* is 8 and *nb* is therefore 3. The bit rate can be found from the formula

$$\text{Bit rate} = \text{sampling rate} \times \text{number of bits per sample} = f_s \times n_b$$

Original Signal Recovery

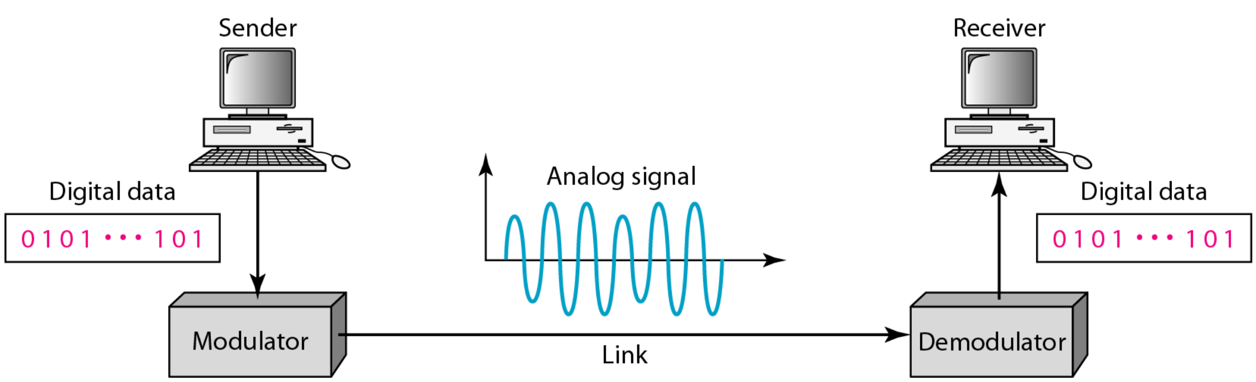
The recovery of the original signal requires the PCM decoder. The decoder first uses circuitry to convert the code words into a pulse that holds the amplitude until the next pulse. After the staircase signal is completed, it is passed through a low-pass filter to smooth the staircase signal into an analog signal. The filter has the same cutoff frequency as the original signal at the sender. If the signal has been sampled at (or greater than) the Nyquist sampling rate and if there are enough quantization levels, the original signal will be recreated. Note that the maximum and minimum values of the original signal can be achieved by using amplification.



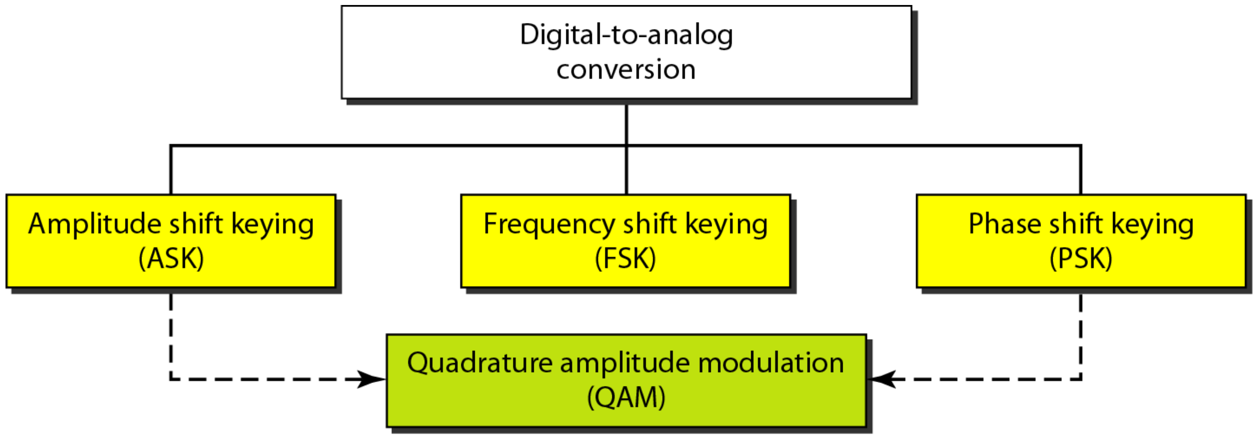
DIGITAL-TO-ANALOG CONVERSION

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

- 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.



Types of digital-to-analog conversion



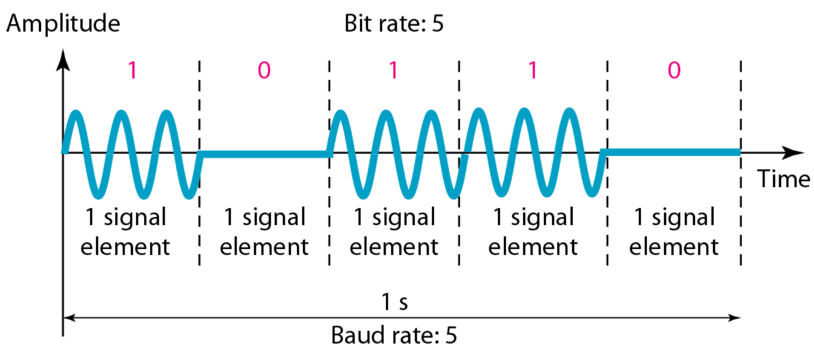
Amplitude Shift Keying (ASK)

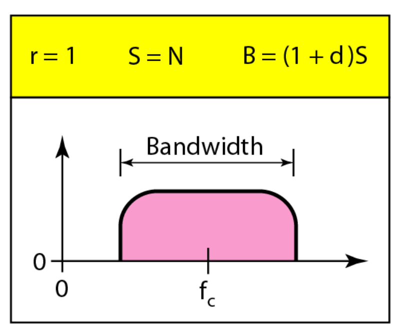
In amplitude shift keying, the amplitude of the carrier signal is varied to create signal
elements. Both frequency and phase remain constant while the amplitude changes.

Binary ASK (BASK)

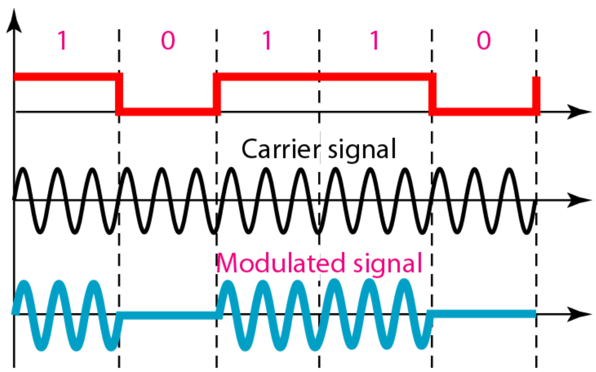
Although we can have several levels (kinds) of signal elements, each with a different amplitude, ASK is normally implemented using only two levels. This is referred to as binary amplitude shift keying or on-off keying (OOK). The peak amplitude of one signal level is 0; the other is the same as the amplitude of the carrier frequency. Figure 5.3 gives a conceptual view of binary ASK.

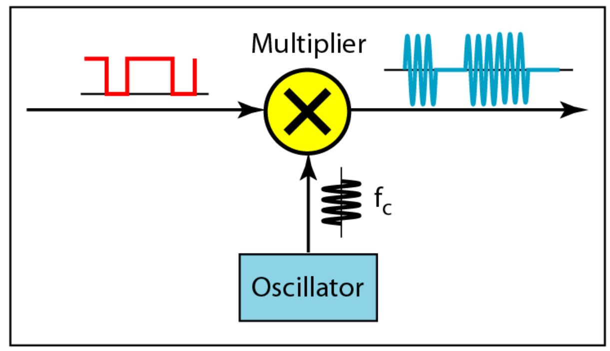
- ASK is implemented by changing the amplitude of a carrier signal to reflect amplitude levels in the digital signal.
- For example: a digital “1” could not affect the signal, whereas a digital “0” would, by making it zero.
- The line encoding will determine the values of the analog waveform to reflect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





Implementation





If digital data are presented as a unipolar NRZ digital signal with a high voltage of 1 V and a low voltage of 0 V, the implementation can achieved by multiplying the NRZ digital signal by the carrier signal coming from an oscillator. When the amplitude of the NRZ signal is 1, the amplitude of the carrier frequency is held; when the amplitude of the NRZ signal is 0, the amplitude of the carrier frequency is zero.

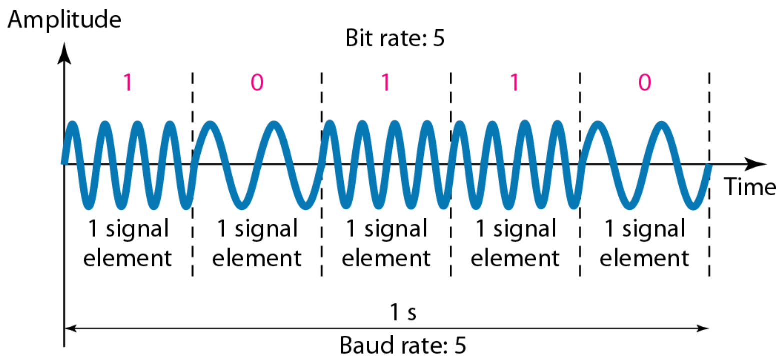
Frequency Shift Keying

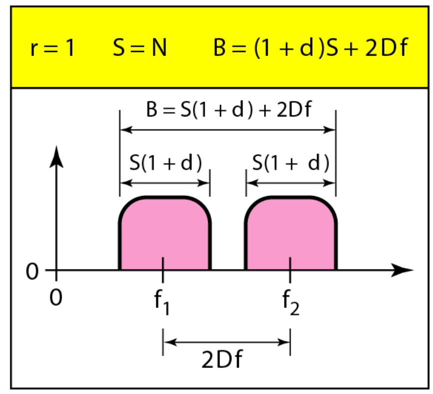
In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements.

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

Binary FSK (BFSK)

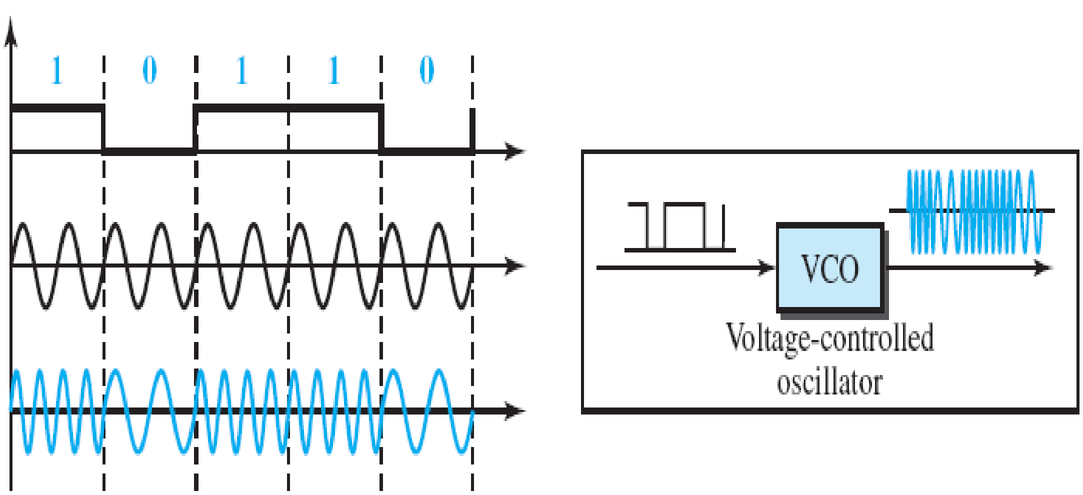
One way to think about binary FSK (or BFSK) is to consider two carrier frequencies. In Figure, we have selected two carrier frequencies, *f*1 and *f*2. We use the first carrier if the data element is 0; we use the second if the data element is 1. However, note that this is an unrealistic example used only for demonstration purposes. Normally the carrier frequencies are very high, and the difference between them is very small.





Implementation of BFSK

Figure 5.7 *Implementation of BFSK*



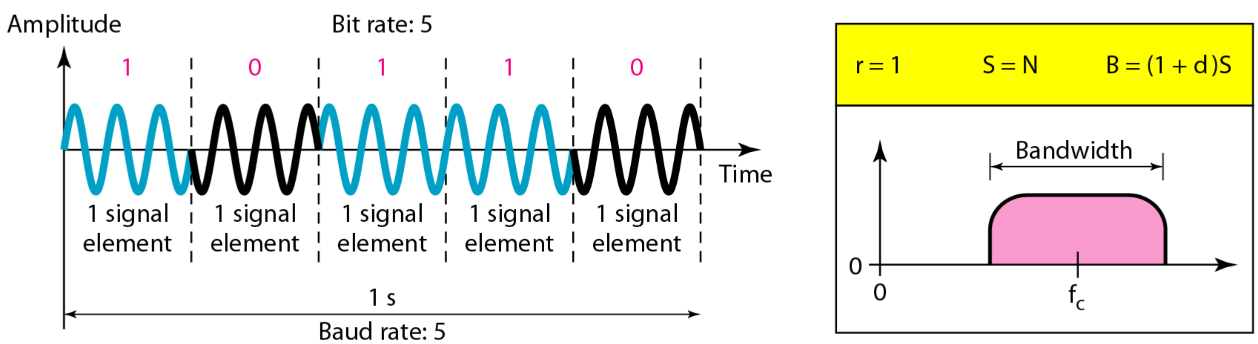
Coherent BFSK can be implemented by using one *voltage-controlled oscillator* (VCO) that changes its frequency according to the input voltage. Figure 5.7 shows the simplified idea behind the second implementation. The input to the oscillator is the unipolar NRZ signal. When the amplitude of NRZis zero, the oscillator keeps its regular frequency; when the amplitude is positive, the frequency is increased.

Phase Shift Keying

In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements. Both peak amplitude and frequency remain constant as the phase changes.

Binary PSK (BPSK)

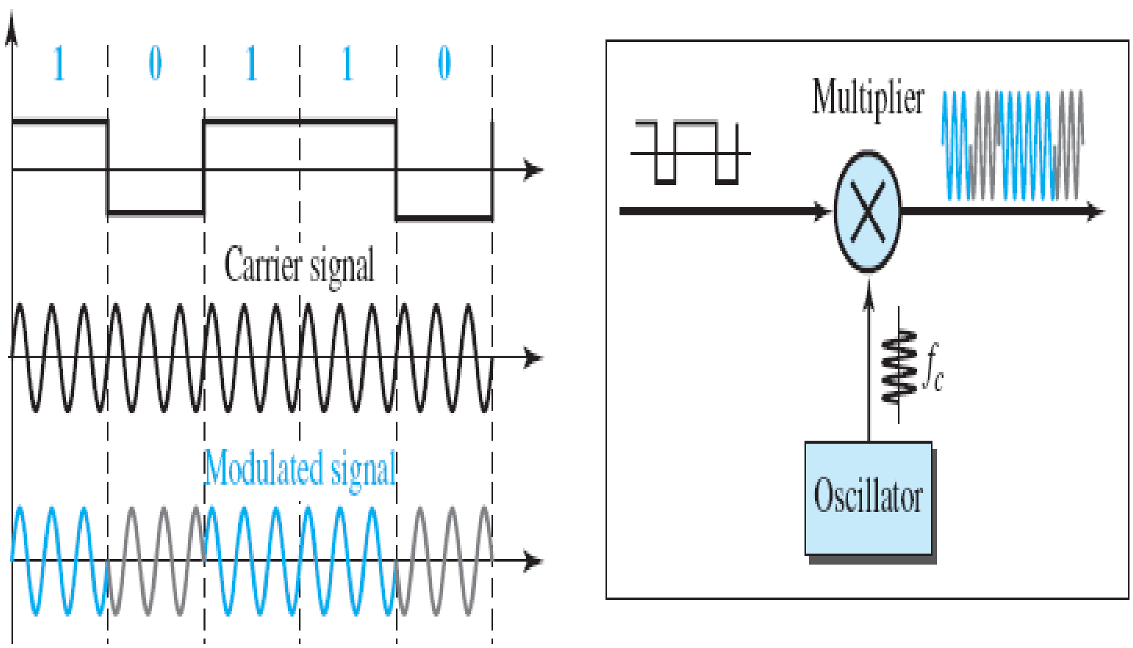
The simplest PSK is binary PSK, in which we have only two signal elements, one with a phase of 0°, and the other with a phase of 180°. Figure 5.9 gives a conceptual view of PSK. Binary PSK is as simple as binary ASK with one big advantage—it is less susceptible to noise. In ASK, the criterion for bit detection is the amplitude of the signal; in PSK, it is the phase. Noise can change the amplitude easier than it can change the phase. In other words, PSK is less susceptible to noise than ASK. PSK is superior to FSK because we do not need two carrier signals. However, PSK needs more sophisticated hardware to be able to distinguish between phases.



Implementation of BPSK

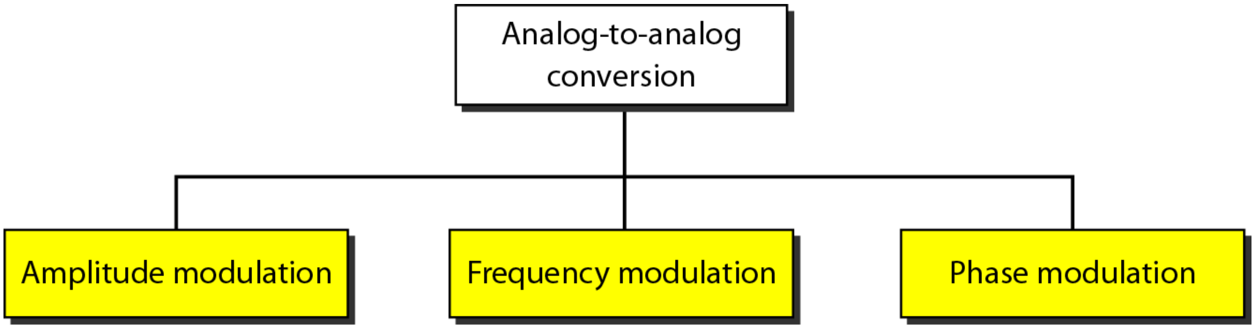
The implementation of BPSK is as simple as that for ASK. The reason is that the signal
element with phase 180° can be seen as the complement of the signal element with phase 0°. The polar NRZ signal is multiplied by the carrier frequency; the 1 bit (positive voltage) is represented by a phase starting at 0°; the 0 bit (negative voltage) is represented by a phase starting at 180°.

Figure 5.10 *Implementation of BASK*



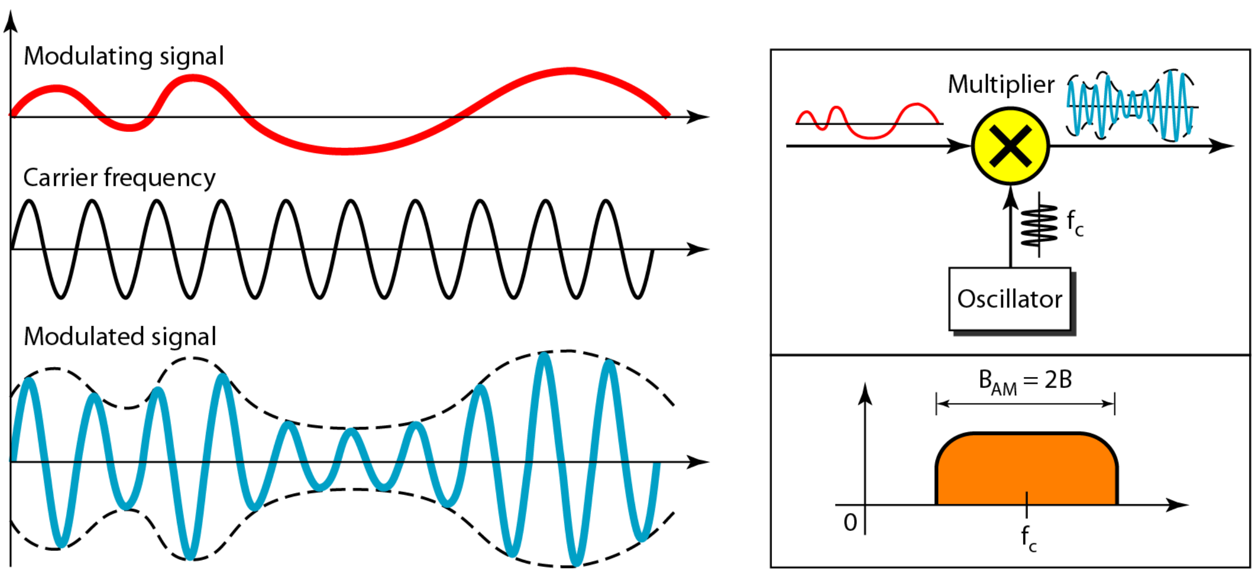
ANALOG AND DIGITAL CONVERSION

- Analog-to-analog conversion is the representation of analog information by an analog signal.
- One may ask why we need to modulate an analog signal; it is already analog.
- Modulation is needed if the medium is bandpass in nature or if only a bandpass channel is available to us.



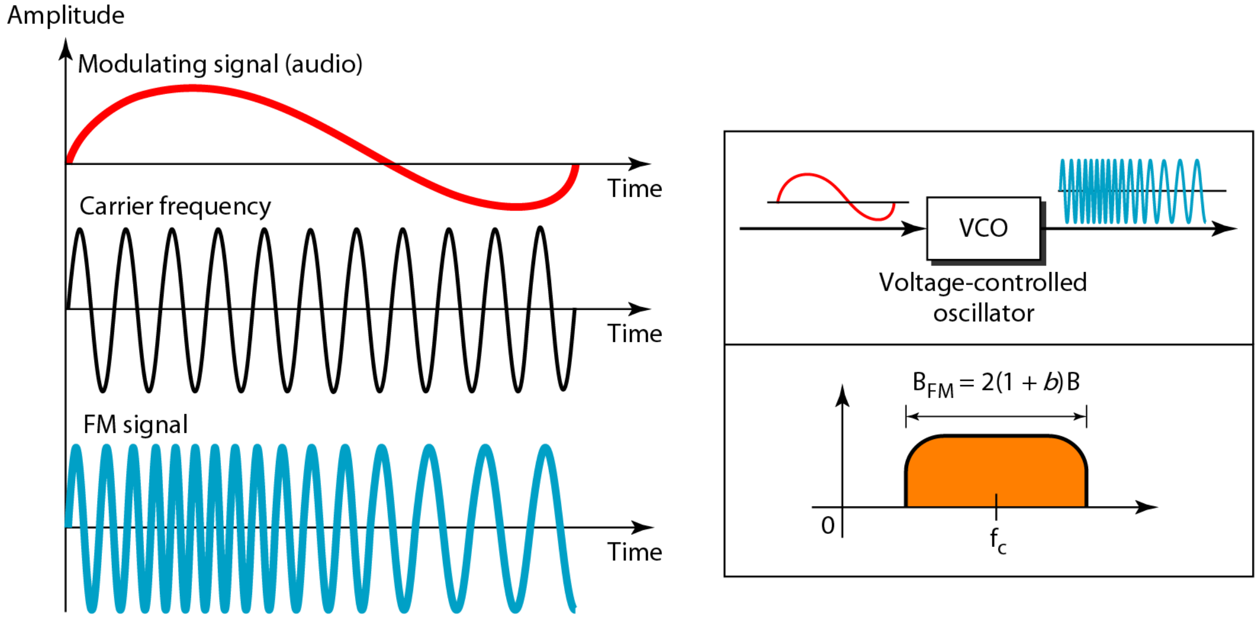
1. Amplitude modulation

In AM transmission, the carrier signal is modulated so that its amplitude varies with the
changing amplitudes of the modulating signal. The frequency and phase of the carrier remain the same; only the amplitude changes to follow variations in the information. The modulating signal is the envelope of the carrier. As Figure shows, AM is normally implemented by using a simple multiplier because the amplitude of the carrier signal needs to be changed according to the amplitude of the modulating signal.



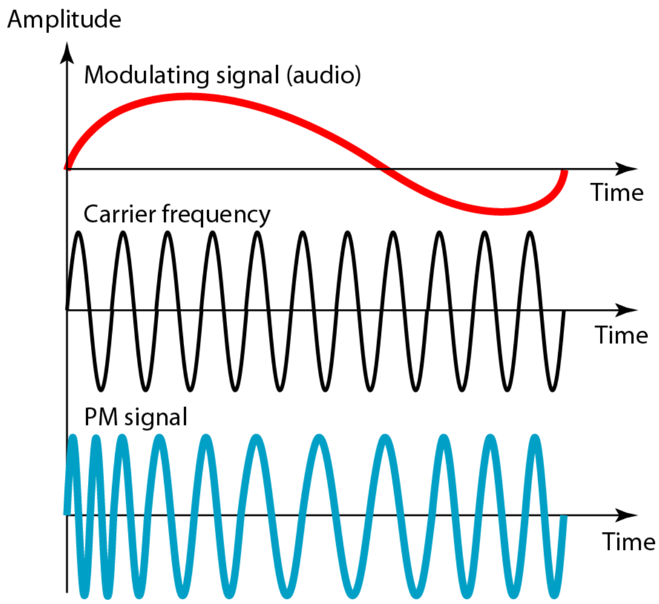
2. Frequency modulation

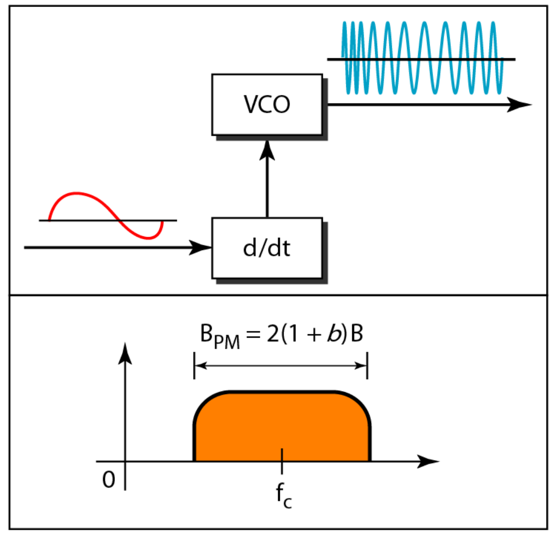
In FM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and phase of the carrier signal remain constant, but as the amplitude of the information signal changes, the frequency of the carrier changes correspondingly. Figur shows the relationships of the modulating signal, the carrier signal, and the resultant FM signal. As Figure shows, FM is normally implemented by using a voltage-controlled oscillator as with FSK. The frequency of the oscillator changes according to the input voltage which is the amplitude of the modulating signal.



3. Phase modulation

In PM transmission, the phase of the carrier signal is modulated to follow the changing
voltage level (amplitude) of the modulating signal. The peak amplitude and frequency of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly. It can be proved mathematically (see Appendix E) that PM is the same as FM with one difference. In FM, the instantaneous change in the carrier frequency is proportional to the amplitude of the modulating signal; in PM the instantaneous change in the carrier frequency is proportional to the derivative of the amplitude of the modulating signal. Figure shows the relationships of the modulating signal, the carrier signal, and the resultant PM signal.

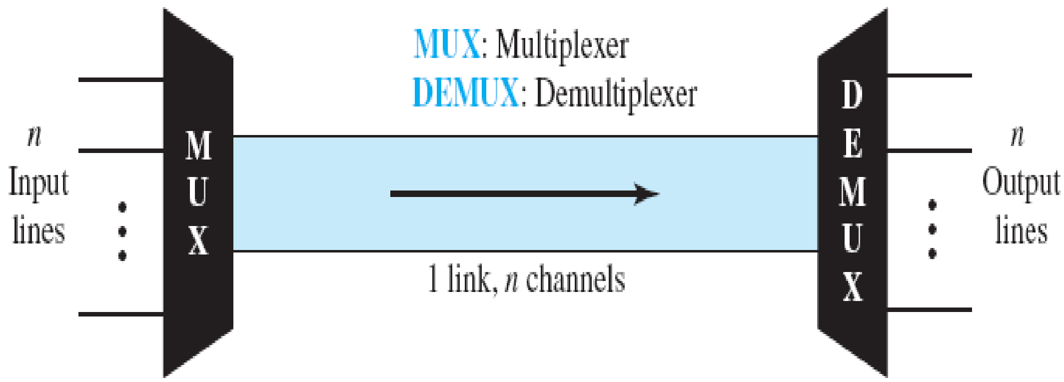




Multiplexing

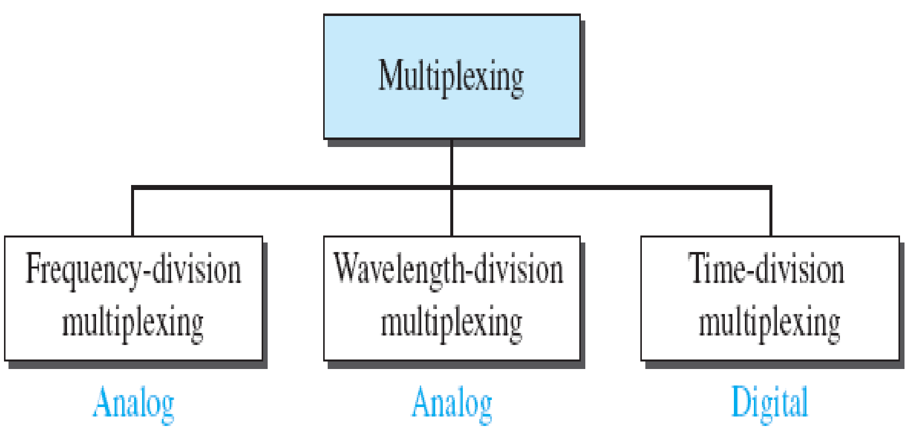
In a multiplexed system, *n* lines share the bandwidth of one link. Figure shows the basic format of a multiplexed system. The lines on the left direct their transmission streams to a **multiplexer (MUX),** which combines them into a single stream (many-toone). At the receiving end, that stream is fed into a **demultiplexer (DEMUX),** which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines. In the figure, the word **link** refers to the physical path. The word **channel** refers to the portion of a link that carries a transmission between a given pair of lines. One link can have many (*n*) channels.

Figure 6.1 *Dividing a link into channels*



TYPES

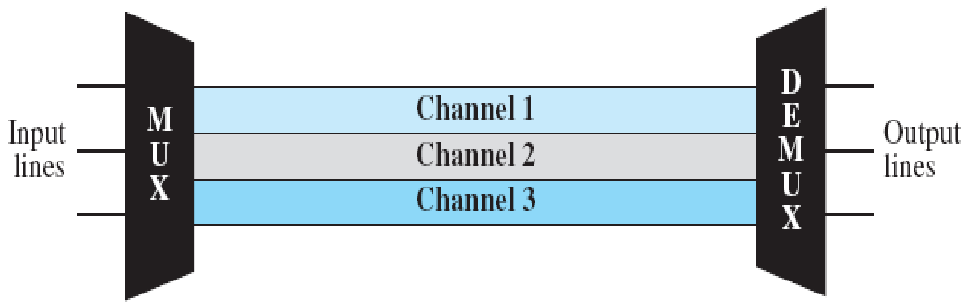
Figure 6.2 *Categories of multiplexing*



1. Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted. In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link. Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal. These bandwidth ranges are the channels through which the various signals travel. Channels can be separated by strips of unused bandwidth—**guard bands**—to prevent signals from overlapping. In addition, carrier frequencies must not interfere with the original data frequencies. Figure gives a conceptual view of FDM. In this illustration, the transmission path is divided into three parts, each representing a channel that carries one transmission.

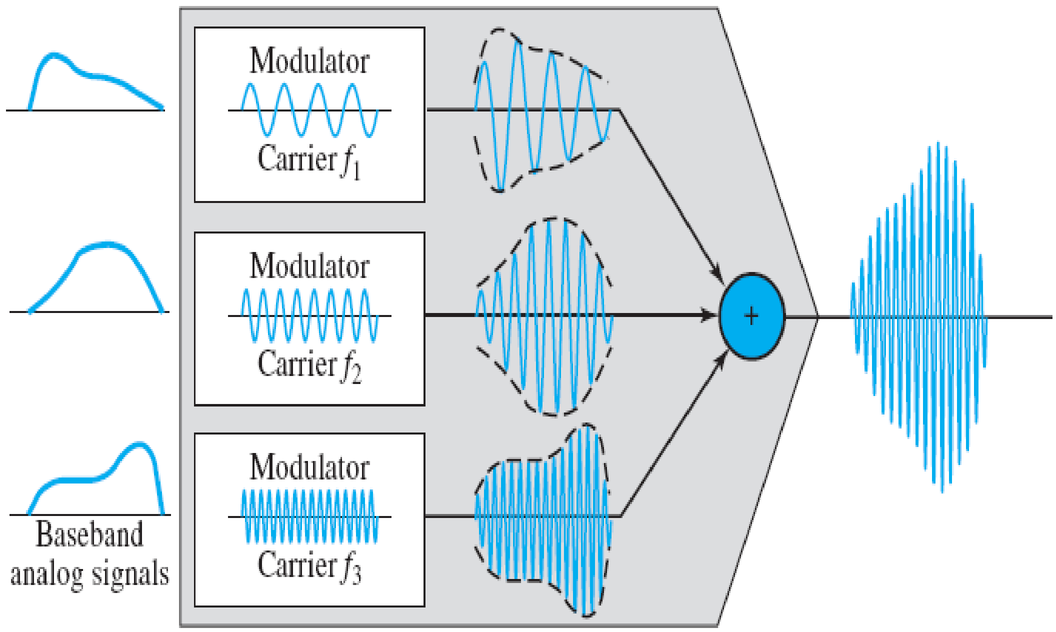
Figure 6.3 *Frequency-division multiplexing*



➢ Multiplexing Process

Figure is a conceptual illustration of the multiplexing process. Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulate different carrier frequencies ($f1$, $f2$, and $f3$). The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.

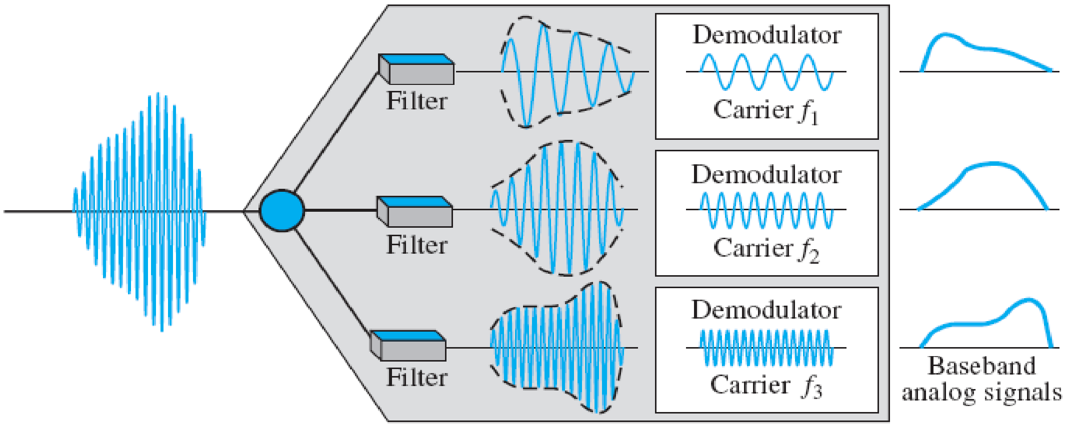
Figure 6.4 *FDM process*



➢ Demultiplexing Process

The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines. Figure is a conceptual illustration of demultiplexing process.

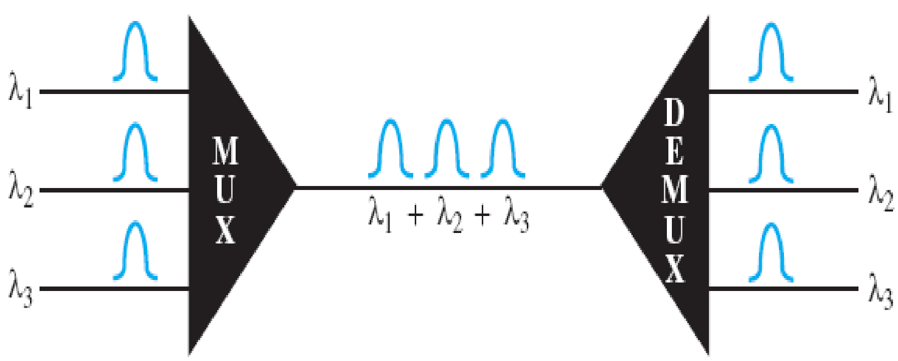
Figure 6.5 *FDM demultiplexing example*



2. Wavelength-Division Multiplexing

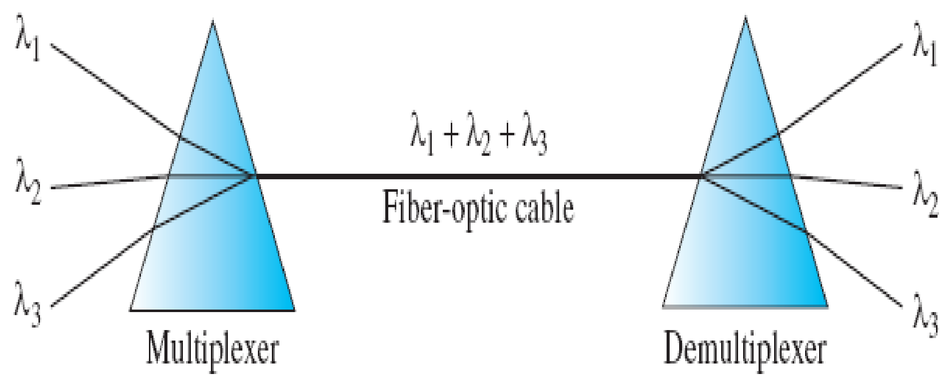
Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable. Multiplexing allows us to combine several lines into one. WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve optical signals transmitted through fiber-optic channels. The idea is the same: We are combining different signals of different frequencies. The difference is that the frequencies are very high. Figure gives a conceptual view of a WDM multiplexer and demultiplexer. Very narrow bands of light from different sources are combined to make a wider band of light. At the receiver, the signals are separated by the demultiplexer.

Figure 6.10 *Wavelength-division multiplexing*



Although WDM technology is very complex, the basic idea is very simple. We want to combine multiple light sources into one single light at the multiplexer and do the reverse at the demultiplexer. The combining and splitting of light sources are easily handled by a prism. Recall from basic physics that a prism bends a beam of light based on the angle of incidence and the frequency. Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of frequencies, into one output beam of a wider band of frequencies. A demultiplexer can also be made to reverse the process. Figure shows the concept.

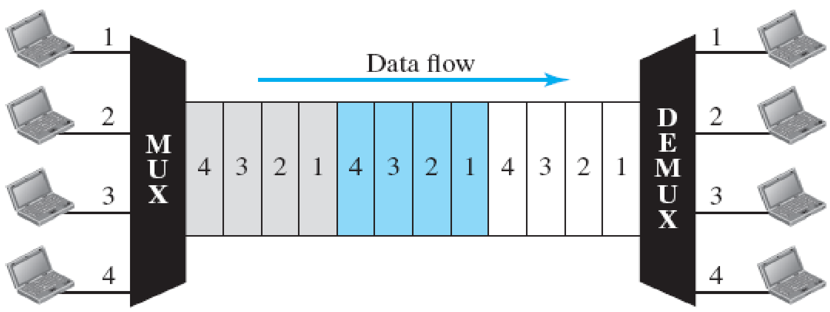
Figure 6.11 *Prisms in wavelength-division multiplexing and demultiplexing*



3. Time-Division Multiplexing

Time-division multiplexing (TDM) is a digital process that allows several connections
to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link. Figure gives a conceptual view of TDM. Note that the same link is used as in FDM; here, however, the link is shown sectioned by time rather than by frequency. In the figure, portions of signals 1, 2, 3, and 4 occupy the link sequentially.

Figure 6.12 *TDM*



We can divide TDM into two different schemes: **synchronous TDM** and **statistical TDM.**

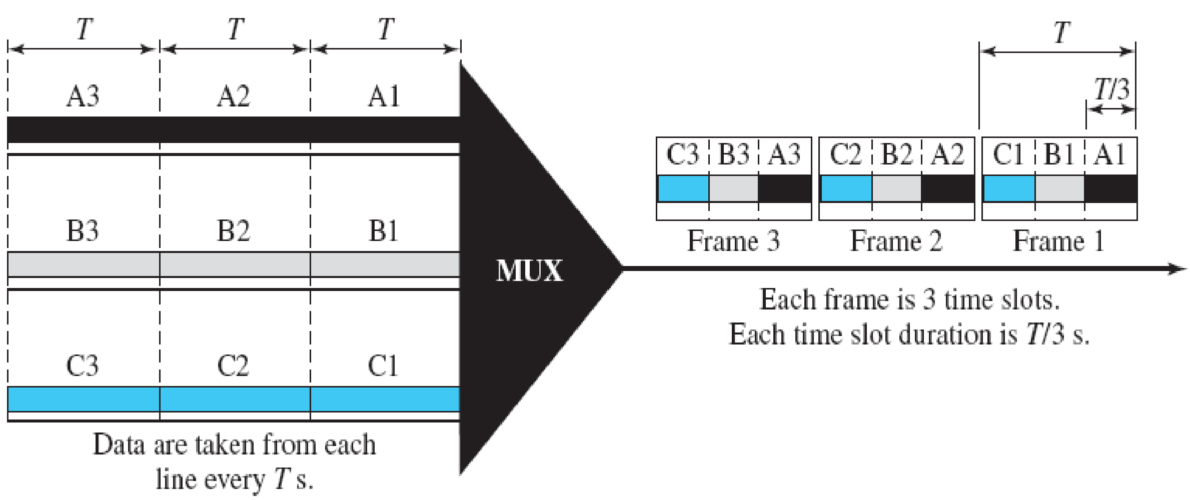
A) Synchronous TDM

In synchronous TDM, each input connection has an allotment in the output even if it is
not sending data.

Time Slots and Frames

In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot. A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is *n* times shorter than the duration of an input time slot. If an input time slot is *T* s, the output time slot is *T*/*n* s, where *n* is the number of connections. In other words, a unit in the output connection has a shorter duration; it travels faster. Figure shows an example of synchronous TDM where *n* is 3.

Figure 6.13 *Synchronous time-division multiplexing*



B) Statistical Time-Division Multiplexing

In synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame. In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in roundrobin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.

Figure shows a synchronous and a statistical TDM example. In the former, some slots are empty because the corresponding line does not have data to send. In the latter, however, no slot is left empty as long as there are data to be sent by any input line.

Figure 6.26 *TDM slot comparison*