Unit 2

ANALOG-TO-DIGITAL CONVERSION

A digital signal is superior to an analog signal because it is more robust to noise and can easily be recovered, corrected and amplified. For this reason, the tendency today is to change an analog signal to digital data. In this section we describe two techniques, pulse code modulation and delta modulation.

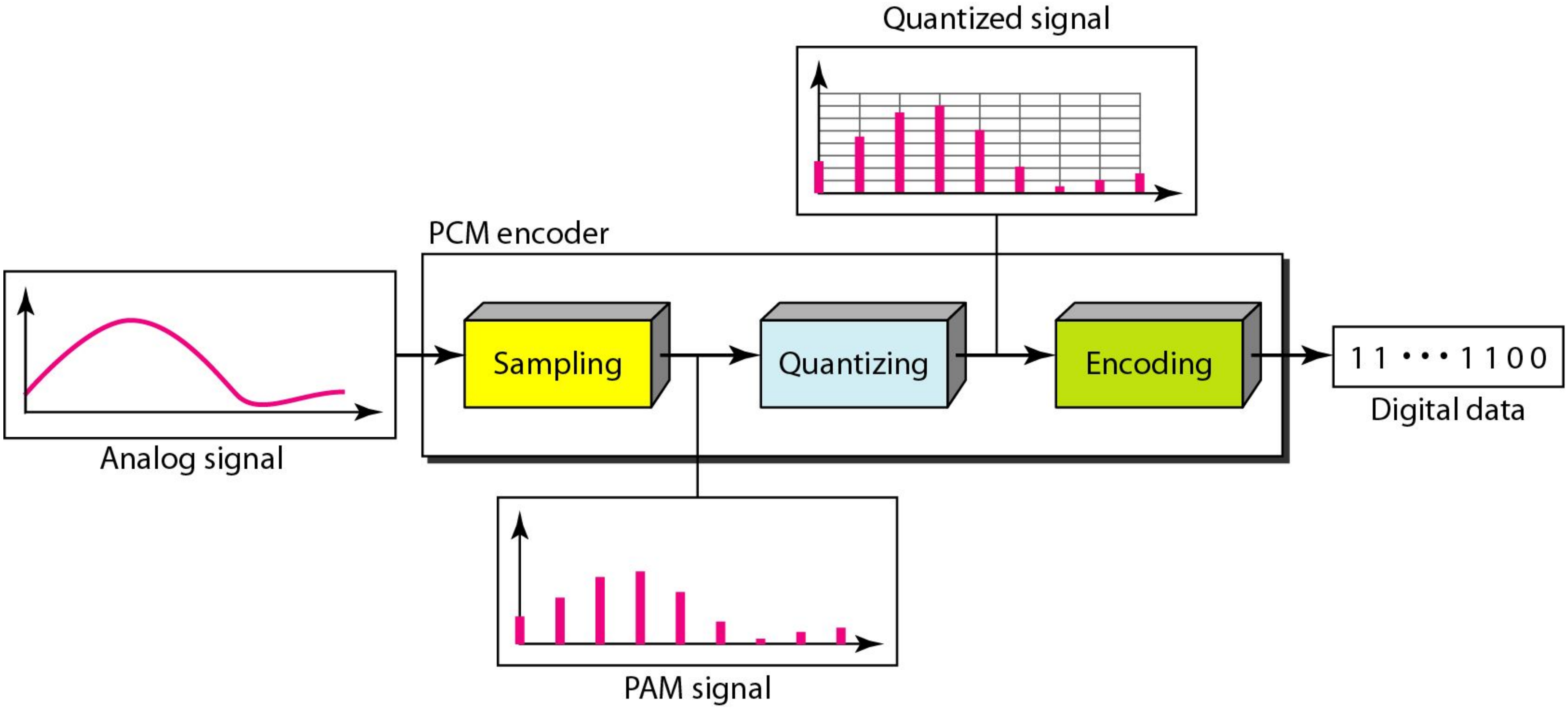
Topics discussed in this section:

- **Pulse Code Modulation (PCM)**
- **Delta Modulation (DM)**

PCM

- PCM consists of three steps to digitize an analog signal:
  1. Sampling
  2. Quantization
  3. Binary encoding
- 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, ie remove high frequency components that affect the signal shape.

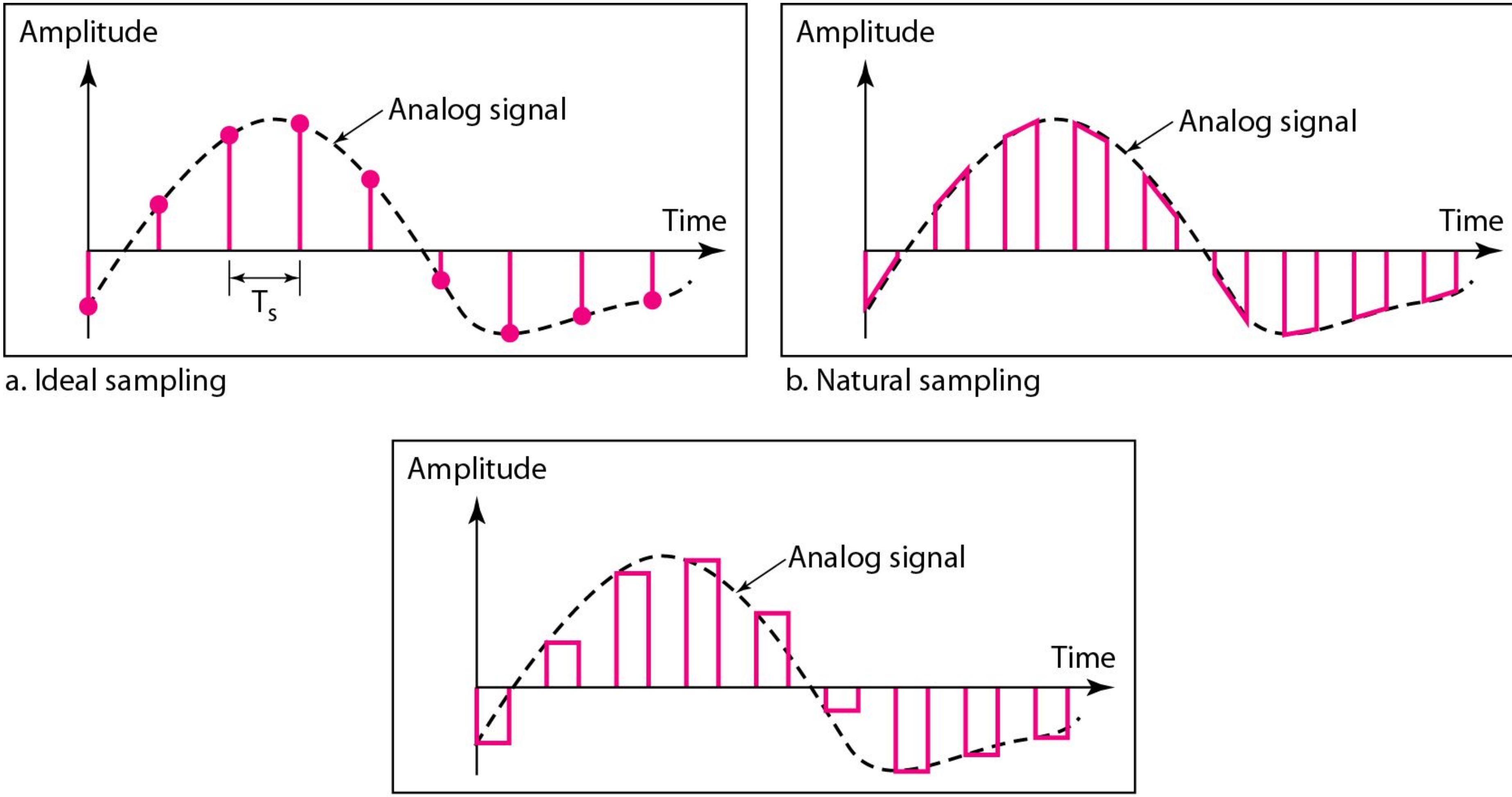
Figure 4.21 *Components of PCM encoder*



Sampling

- Analog signal is sampled every T_S secs.
- T_s is referred to as the sampling interval.
- $f_s = 1/T_s$ is called the sampling rate or sampling frequency.
- 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
- The process is referred to as pulse amplitude modulation PAM and the outcome is a signal with analog (non integer) values

Figure 4.22 *Three different sampling methods for PCM*



a. Ideal sampling

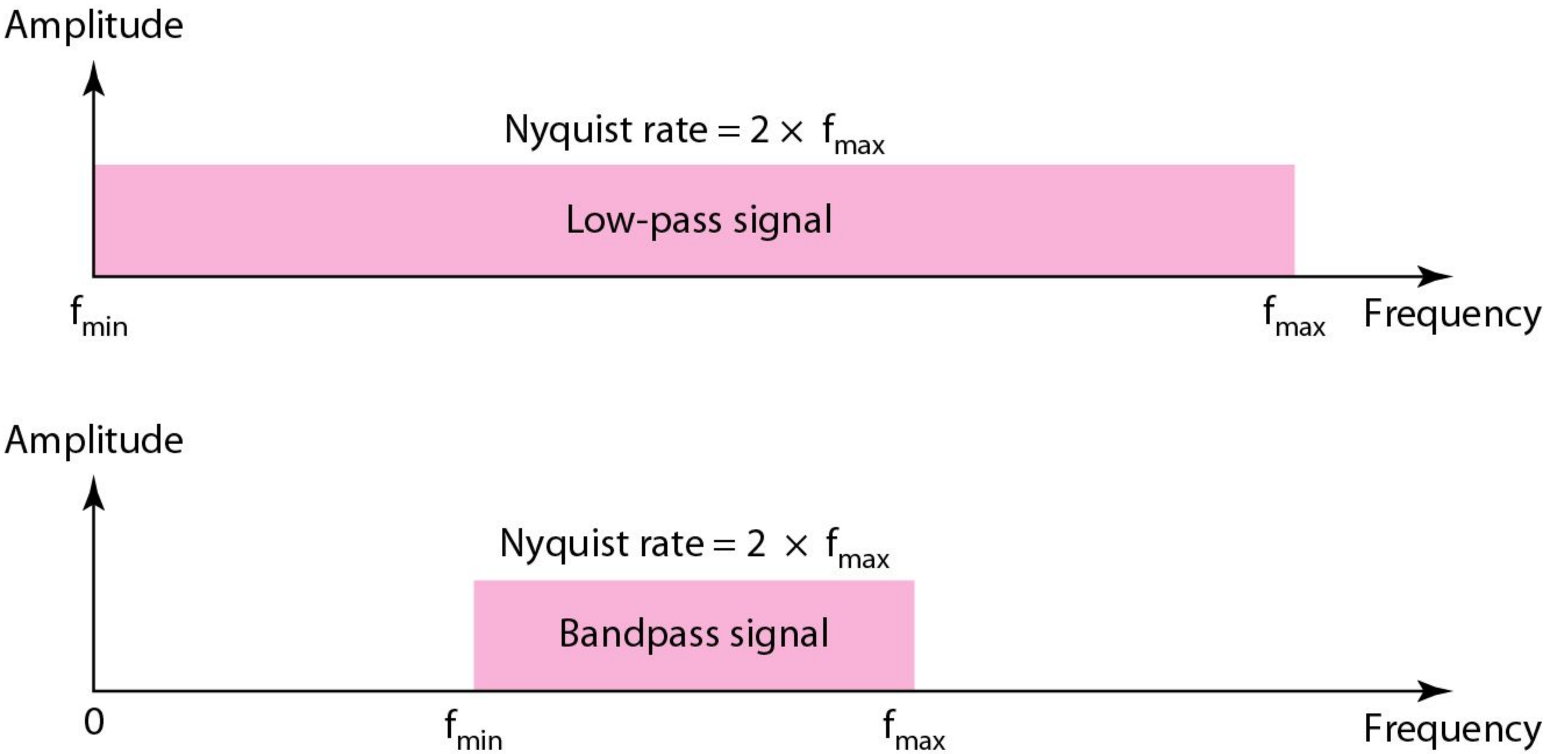
b. Natural sampling

c. Flat-top sampling

Note

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

Figure 4.23 *Nyquist sampling rate for low-pass and bandpass signals*

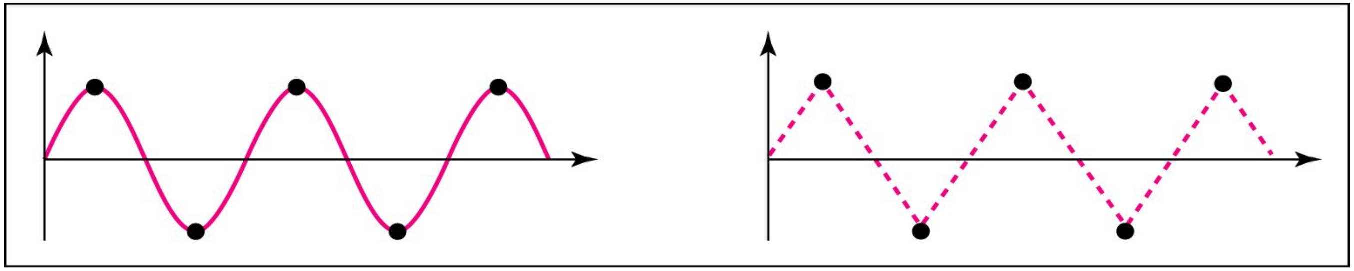


Example 4.6

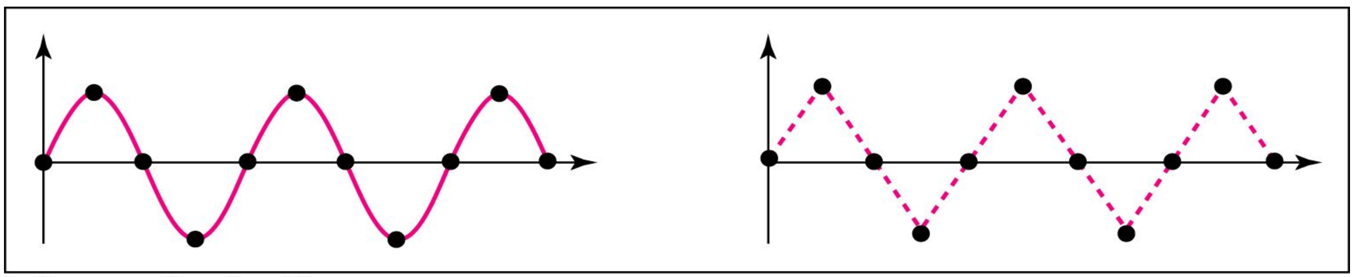
For an intuitive example of the Nyquist theorem, let us sample a simple sine wave at three sampling rates: $f_s = 4f$ (2 times the Nyquist rate), $f_s = 2f$ (Nyquist rate), and $f_s = f$ (one-half the Nyquist rate). Figure 4.24 shows the sampling and the subsequent recovery of the signal.

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.

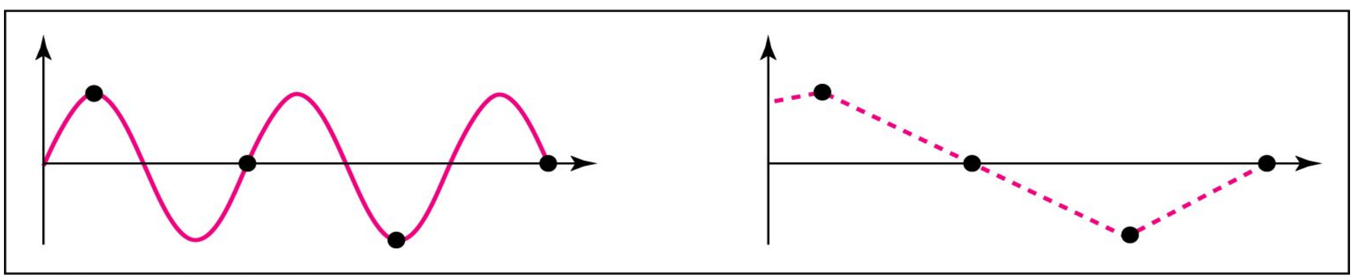
Figure 4.24 *Recovery of a sampled sine wave for different sampling rates*



a. Nyquist rate sampling: $f_s = 2f$



b. Oversampling: $f_s = 4f$

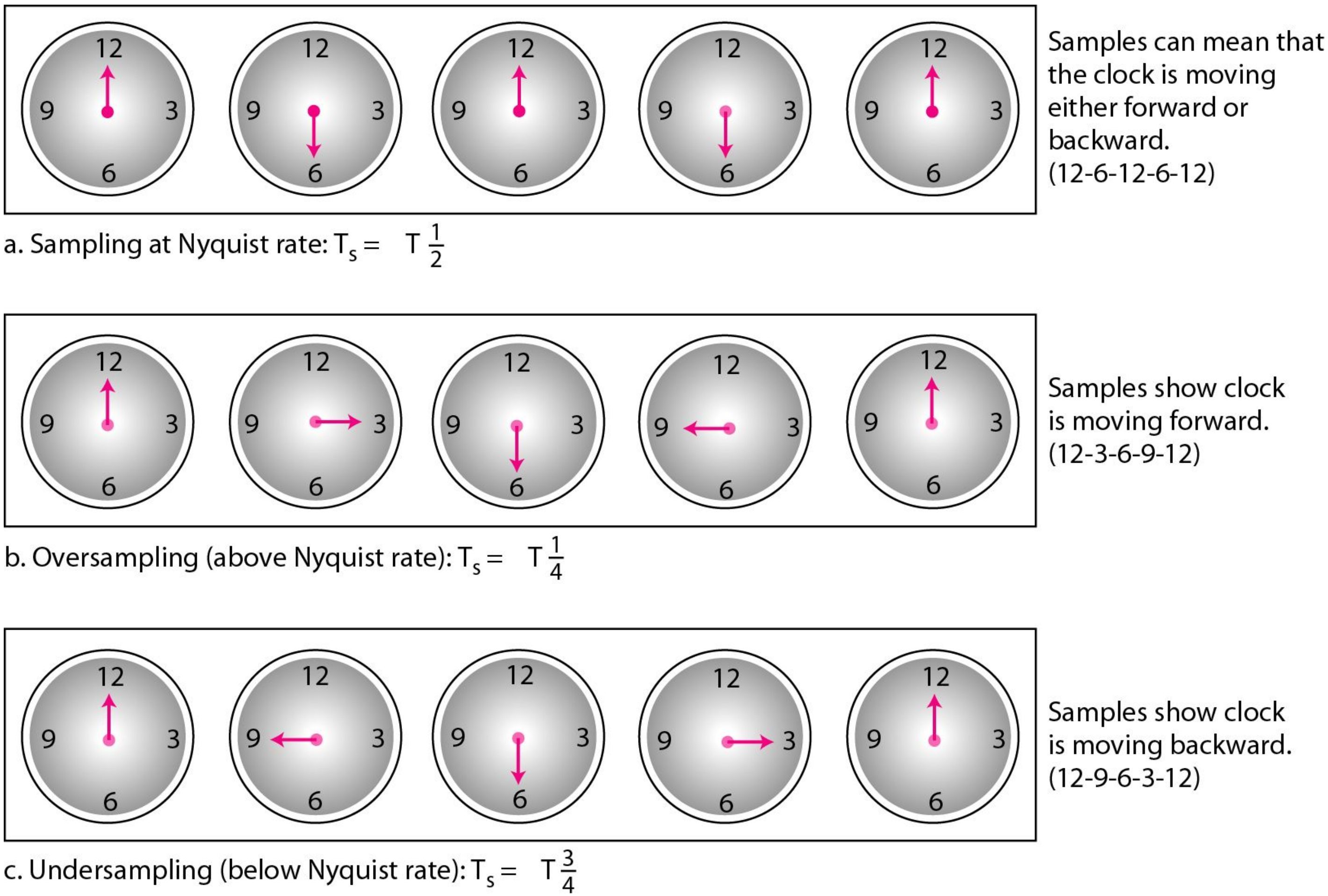


c. Undersampling: $f_s = f$

Example 4.7

Consider the revolution of a hand of a clock. The second hand of a clock has a period of 60 s. According to the Nyquist theorem, we need to sample the hand every 30 s ($T_s = T$ or $f_s = 2f$). In Figure 4.25a, the sample points, in order, are 12, 6, 12, 6, 12, and 6. The receiver of the samples cannot tell if the clock is moving forward or backward. In part b, we sample at double the Nyquist rate (every 15 s). The sample points are 12, 3, 6, 9, and 12. The clock is moving forward. In part c, we sample below the Nyquist rate ($T_s = T$ or $f_s = f$). The sample points are 12, 9, 6, 3, and 12. Although the clock is moving forward, the receiver thinks that the clock is moving backward.

Figure 4.25 *Sampling of a clock with only one hand*



Example 4.8

An example related to Example 4.7 is the seemingly backward rotation of the wheels of a forward-moving car in a movie. This can be explained by under-sampling. A movie is filmed at 24 frames per second. If a wheel is rotating more than 12 times per second, the under-sampling creates the impression of a backward rotation.

Example 4.9

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

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

- 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.

Quantization Zones

- Assume we have a voltage signal with amplitutes V_{min}=-20V and V_{max}=+20V.
- We want to use L=8 quantization levels.
- Zone width $\Delta = (20 - -20)/8 = 5$
- 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

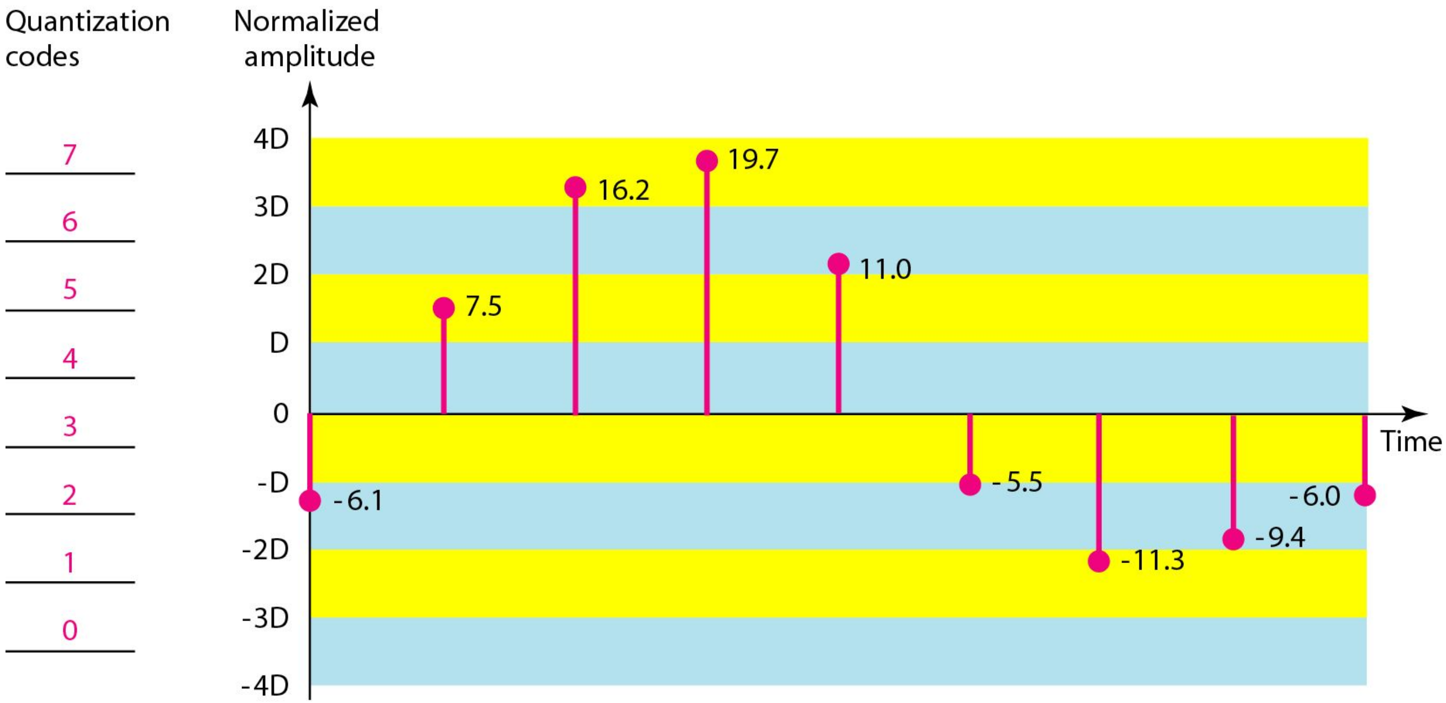
Assigning Codes to Zones

- 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.

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

- 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 form 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. Refer to previous section for discussion and formulas.
- A digitized signal will always need more bandwidth than the original analog signal. Price we pay for robustness and other features of digital transmission.

Example 4.14

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

Solution

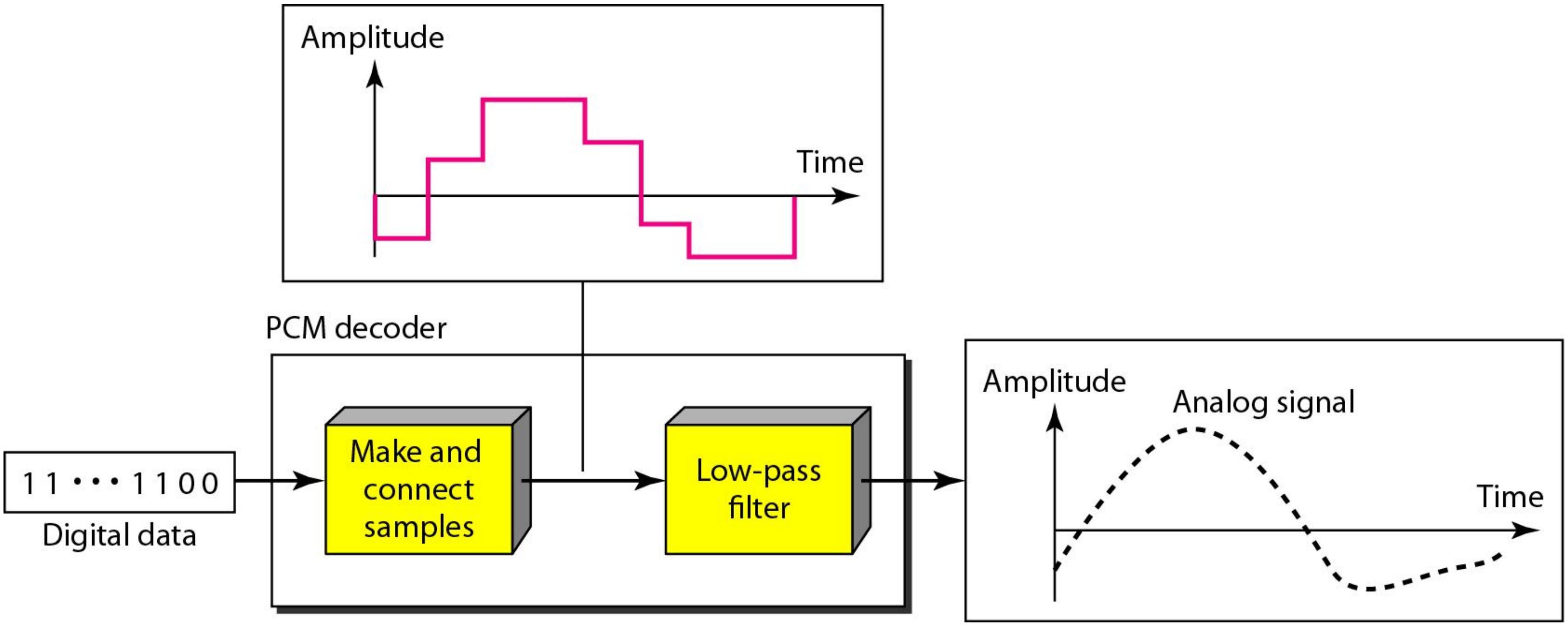
The human voice normally contains frequencies from 0 to 4000 Hz. So the sampling rate and bit rate are calculated as follows:

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

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.

Figure 4.27 *Components of a PCM decoder*



Example 4.15

We have a low-pass analog signal of 4 kHz. If we send the analog signal, we need a channel with a minimum bandwidth of 4 kHz. If we digitize the signal and send 8 bits per sample, we need a channel with a minimum bandwidth of 8 × 4 kHz = 32 kHz.

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.

Figure 4.28 *The process of delta modulation*

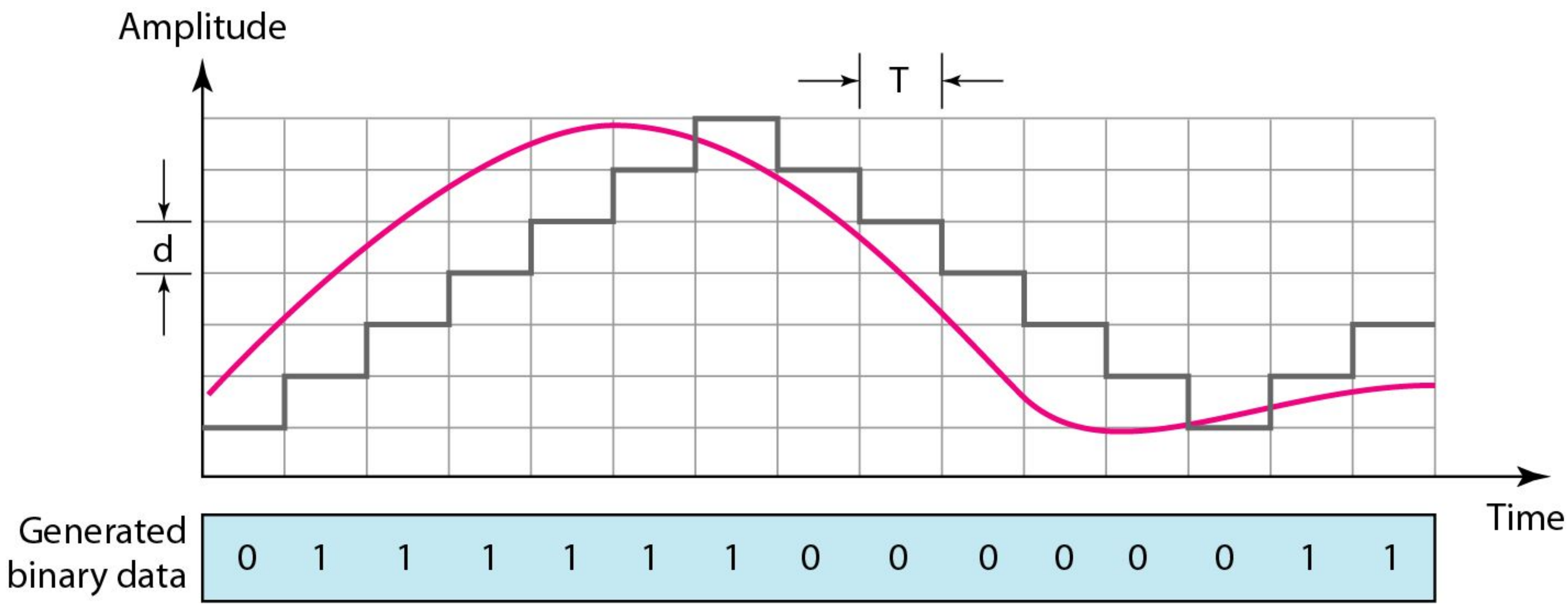


Figure 4.29 *Delta modulation components*

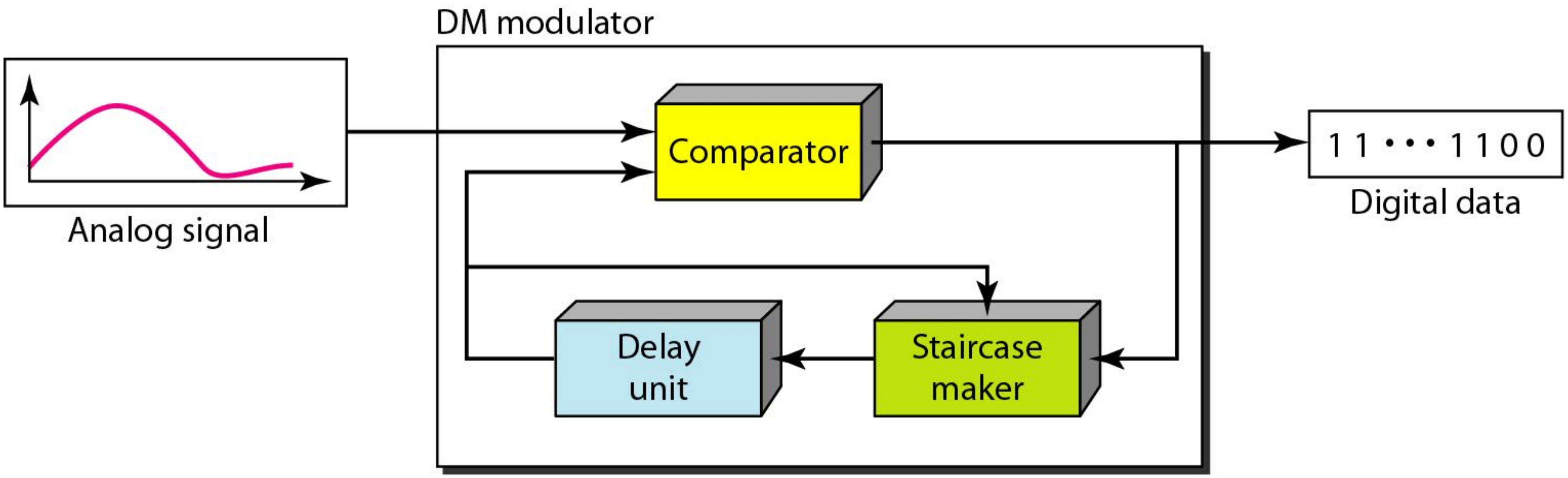
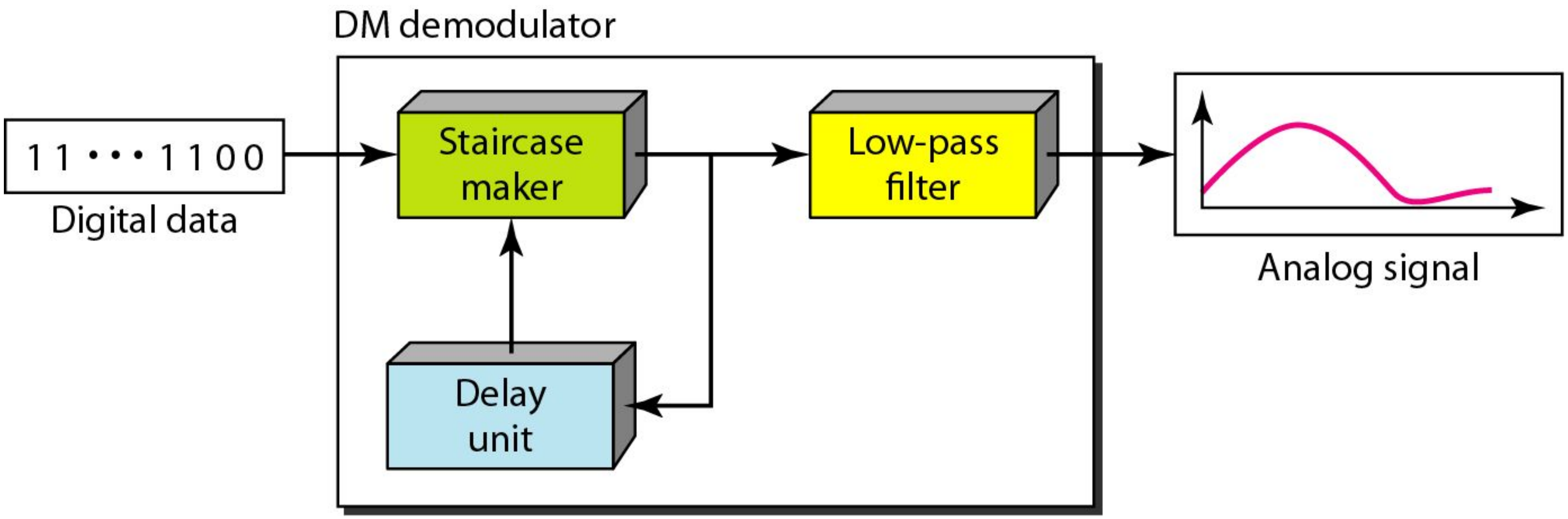


Figure 4.30 *Delta demodulation components*



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.

Note

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. The bits are usually sent as bytes and many bytes are grouped in a frame. A frame is identified with a start and an end byte.

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.

<u>Topics discussed in this section:</u>

Line Coding
Line Coding Schemes
Block Coding
Scrambling

Figure 4.1 *Line coding and decoding*

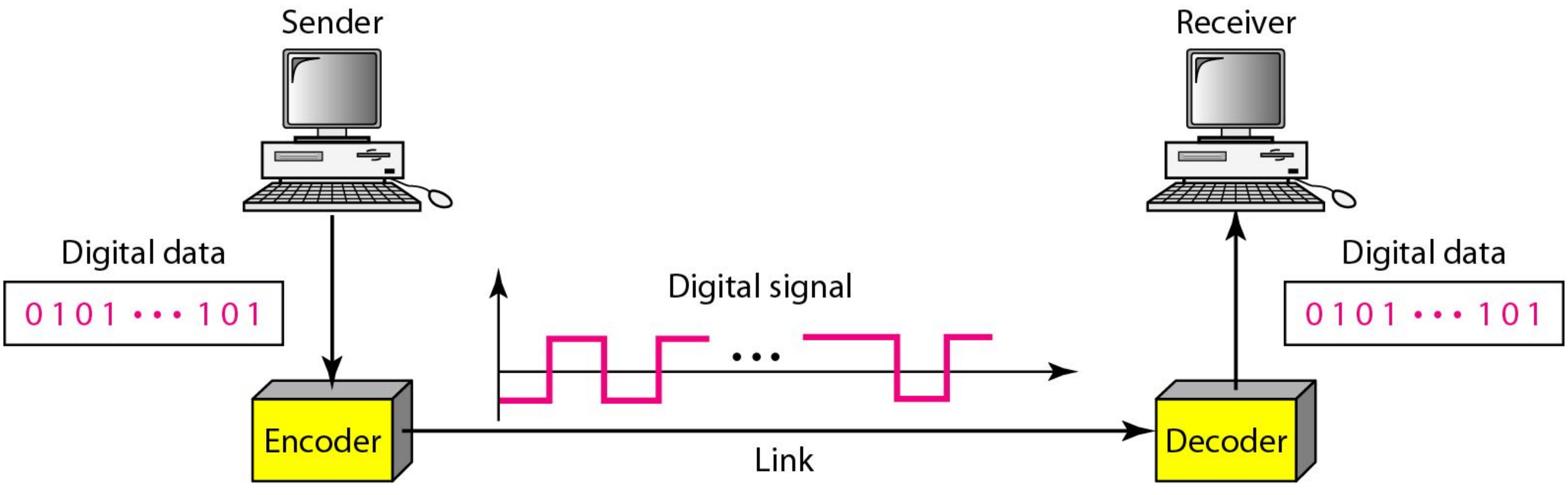
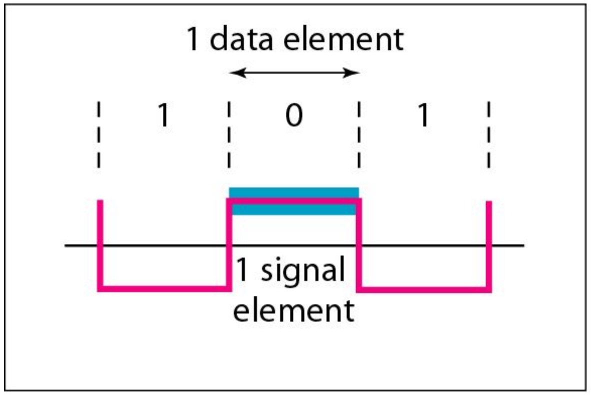
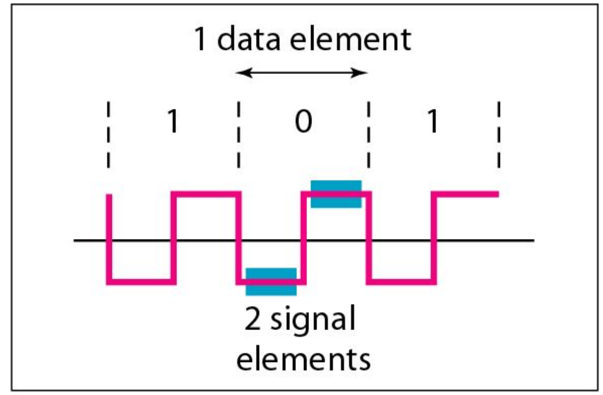


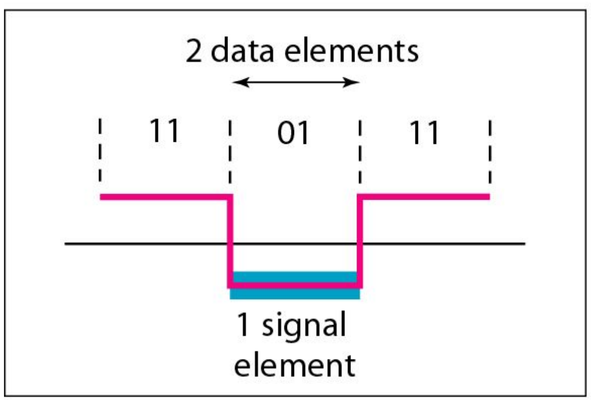
Figure 4.2 *Signal element versus data element*



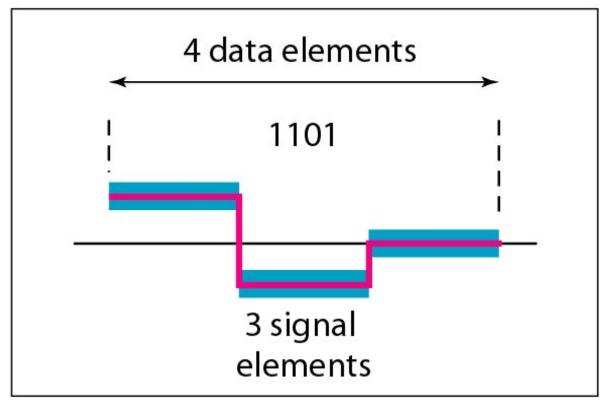
a. One data element per one signal element (r = 1)



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 $\left(r = \frac{4}{3}\right)$

Example 4.1

A signal is carrying data in which one data element is encoded as one signal element ($r = 1$). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

Solution

We assume that the average value of c is 1/2 . The baud rate is then

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100{,}000 \times \frac{1}{1} = 50{,}000 = 50 \text{ kbaud}$$

Example 4.2

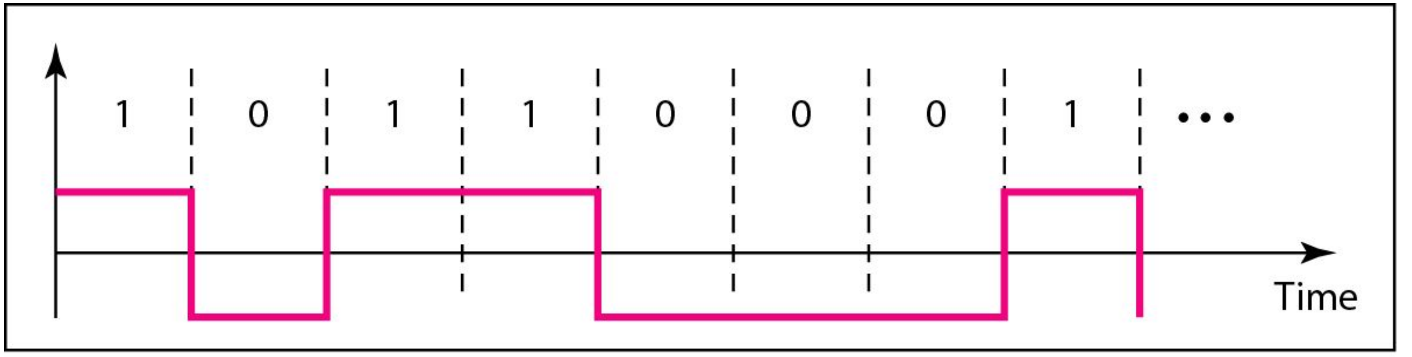
The maximum data rate of a channel is $N_{max} = 2 \times B \times \log_2 L$ (defined by the Nyquist formula). Does this agree with the previous formula for N_{max}?

Solution

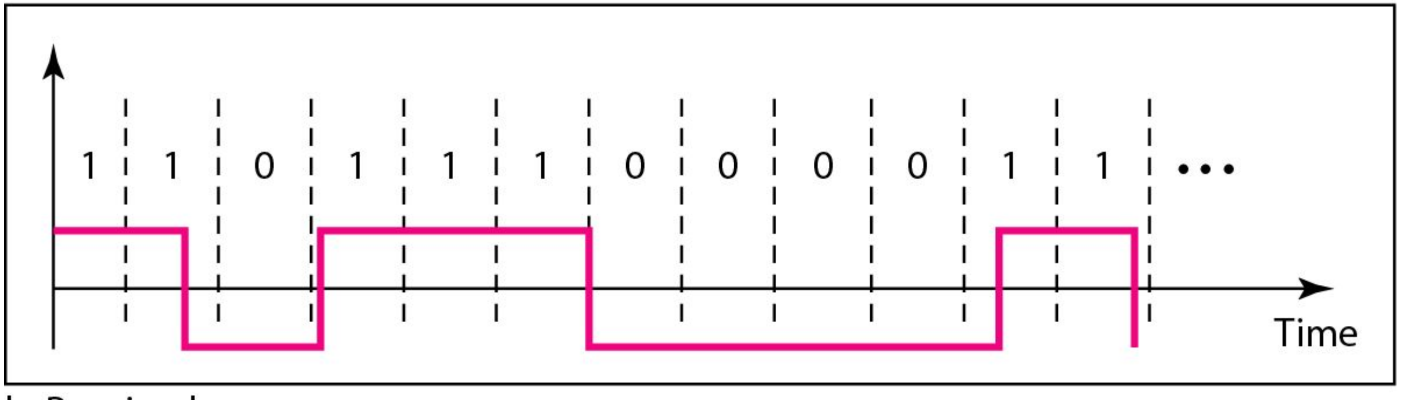
A signal with L levels actually can carry $\log_2 L$ bits per level. If each level corresponds to one signal element and we assume the average case ($c = 1/2$), then we have

$$\boxed{N_{\max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L}$$

Figure 4.3 ***Effect of lack of synchronization***



a. Sent



b. Received

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

Figure 4.4 *Line coding schemes*

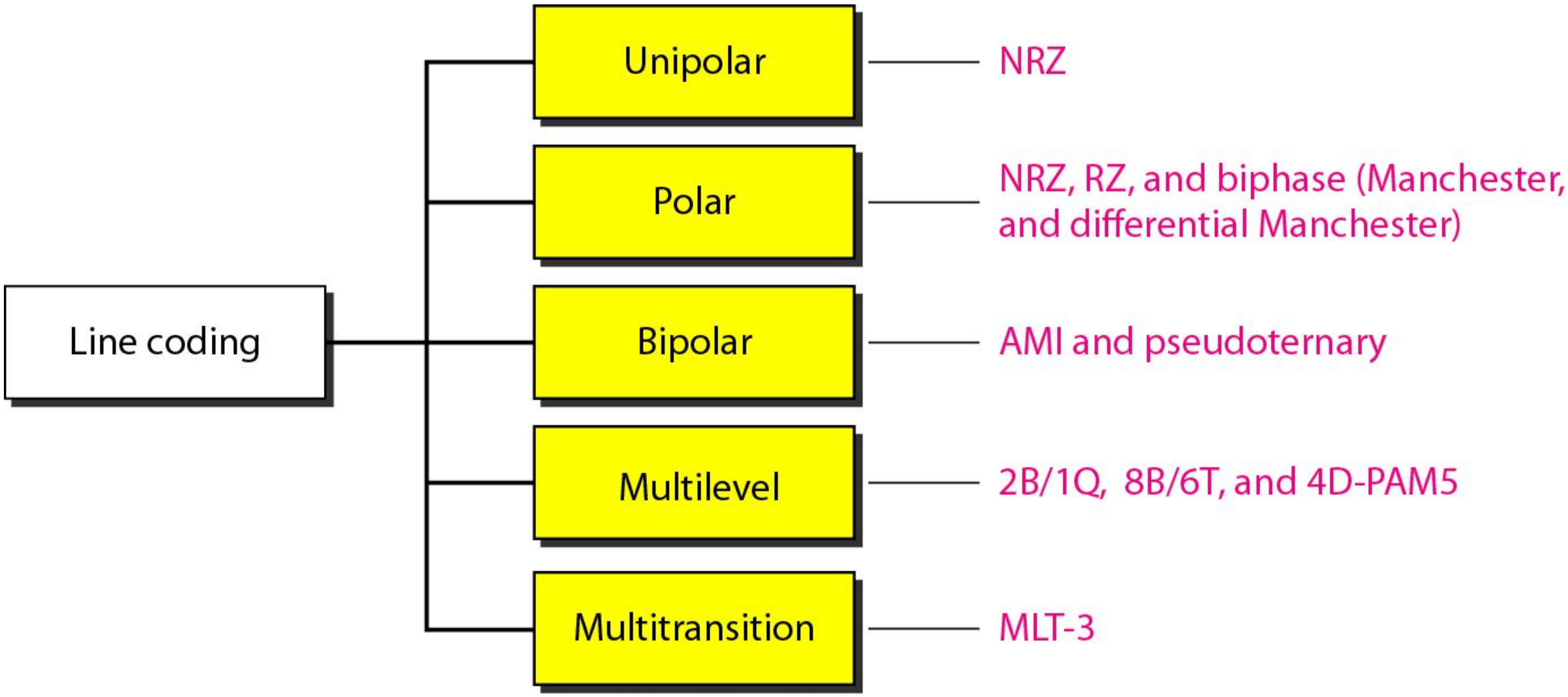
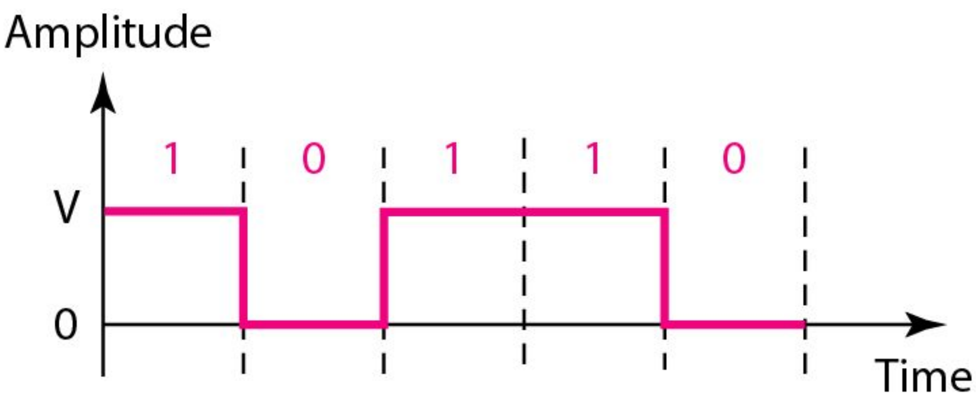


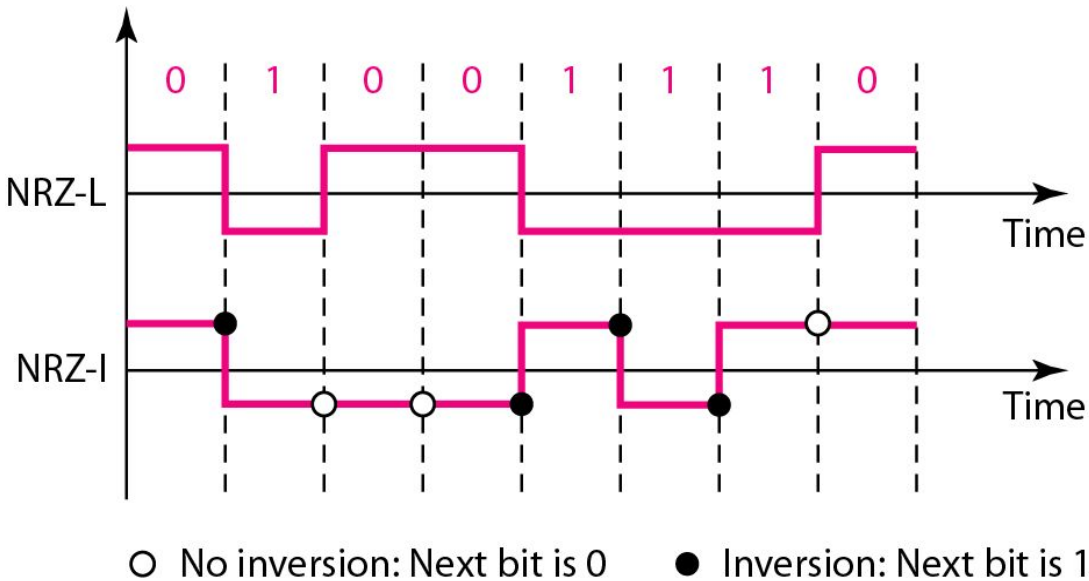
Figure 4.5 *Unipolar NRZ scheme*

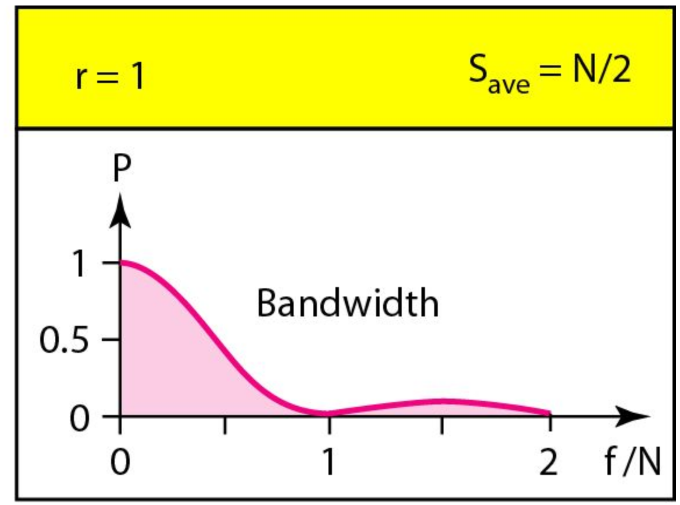


$$\frac{1}{2}V^2 + \frac{1}{2}(0)^2 = \frac{1}{2}V^2$$

Normalized power

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





Note

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.

Note

NRZ-L and NRZ-I both have an average signal rate of N/2 Bd.

Note

NRZ-L and NRZ-I both have a DC component problem.

Example 4.4

A system is using NRZ-I to transfer 10-Mbps data. What are the average signal rate and minimum bandwidth?

Solution

The average signal rate is $S = N/2 = 500$ kbaud. The minimum bandwidth for this average baud rate is $B_{min} = S = 500$ kHz.

Figure 4.7 *Polar RZ scheme*

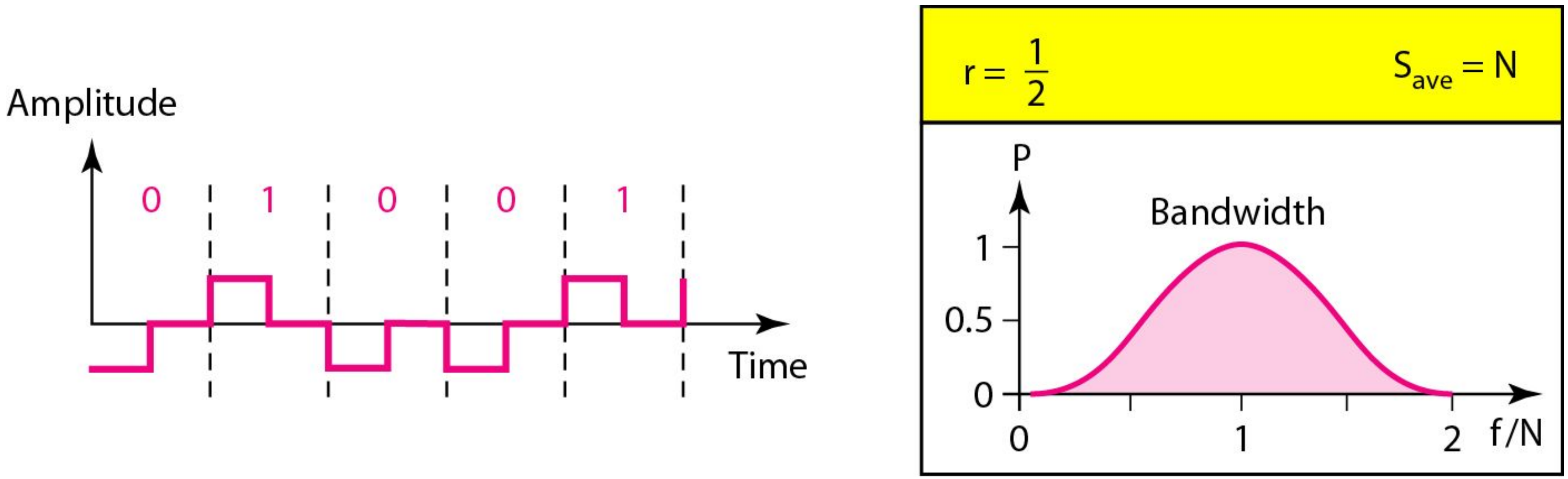
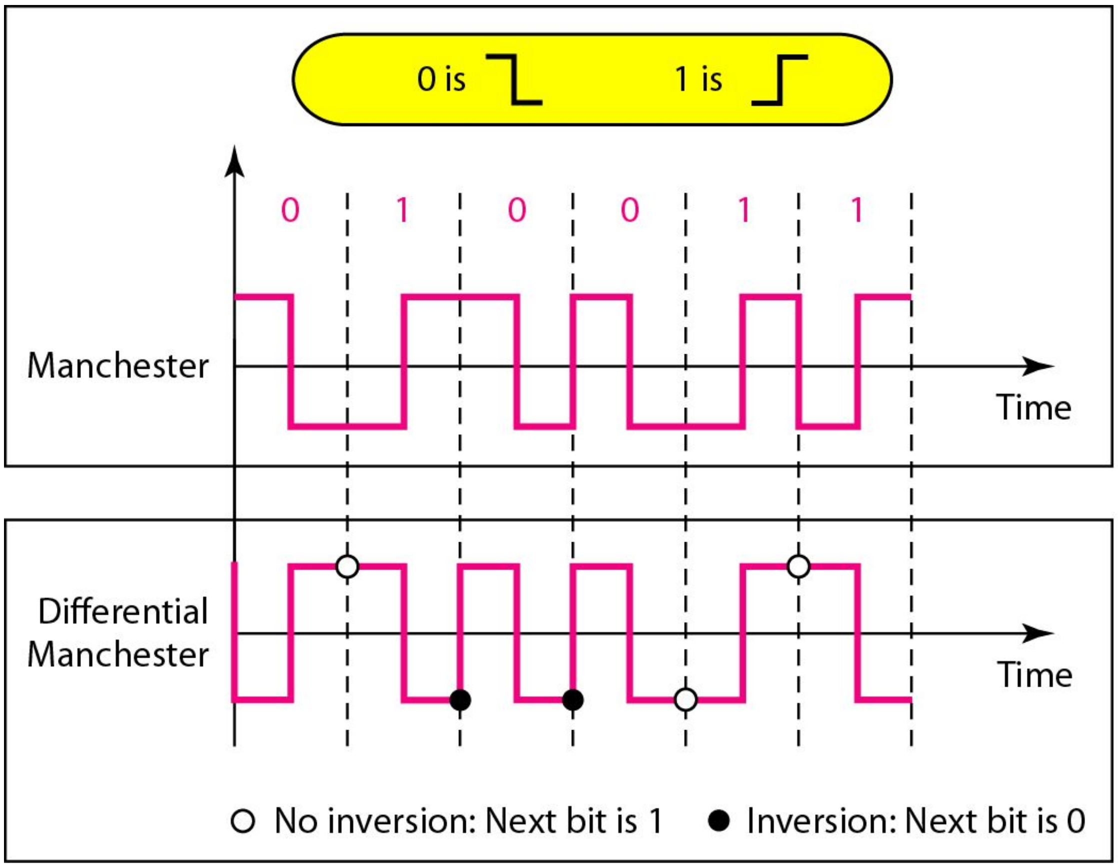
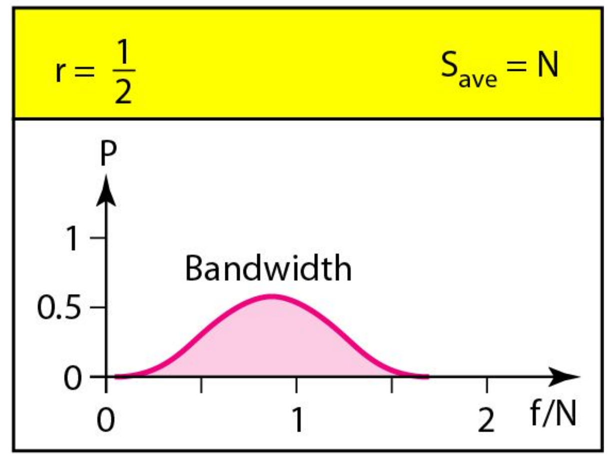


Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*





Note

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

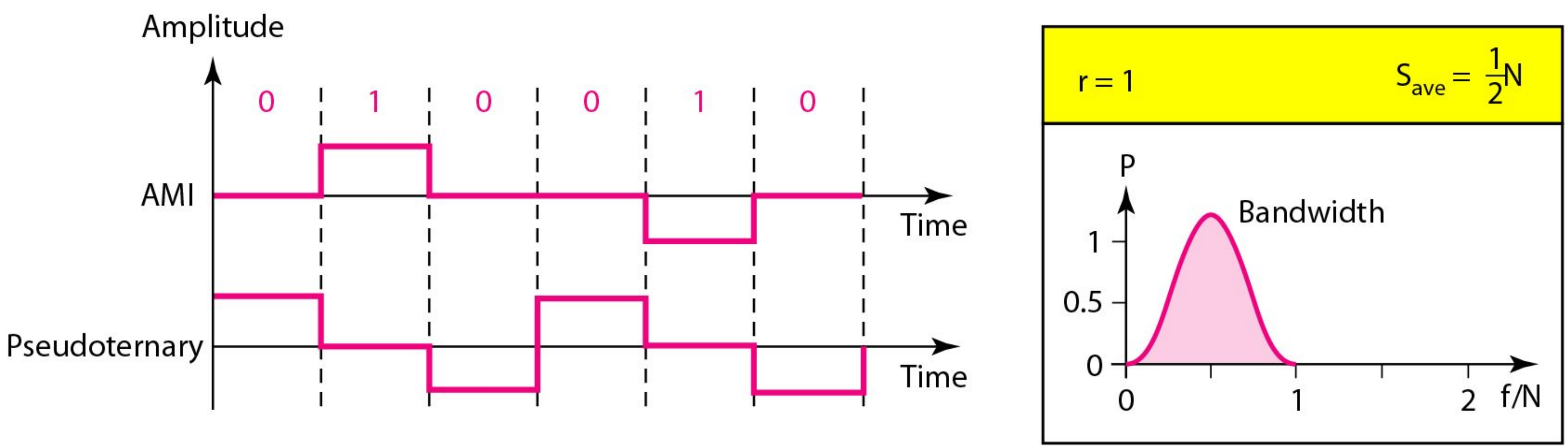
Note

The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ.

Note

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

Figure 4.9 *Bipolar schemes: AMI and pseudoternary*



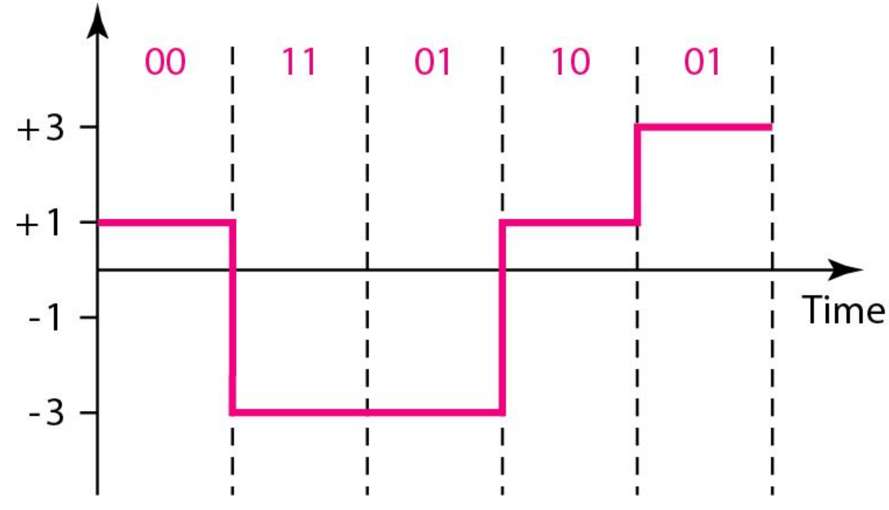
Note

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*

	Previous level: positive	Previous level: negative
Next bits	Next level	Next level
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

Transition table



Assuming positive original level

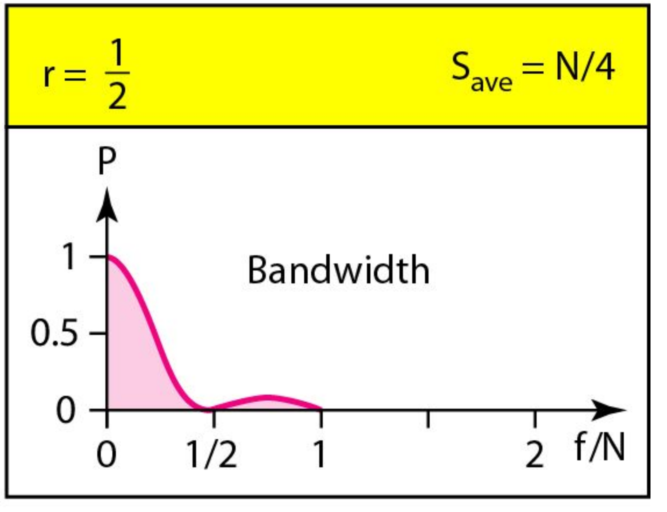


Figure 4.11 *Multilevel: 8B6T scheme*

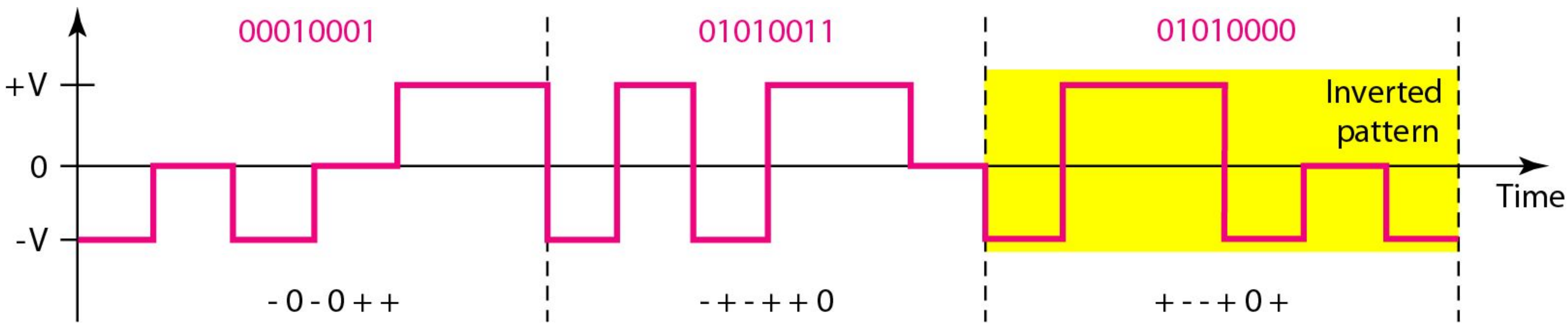


Figure 4.12 ***Multilevel: 4D-PAM5 scheme***

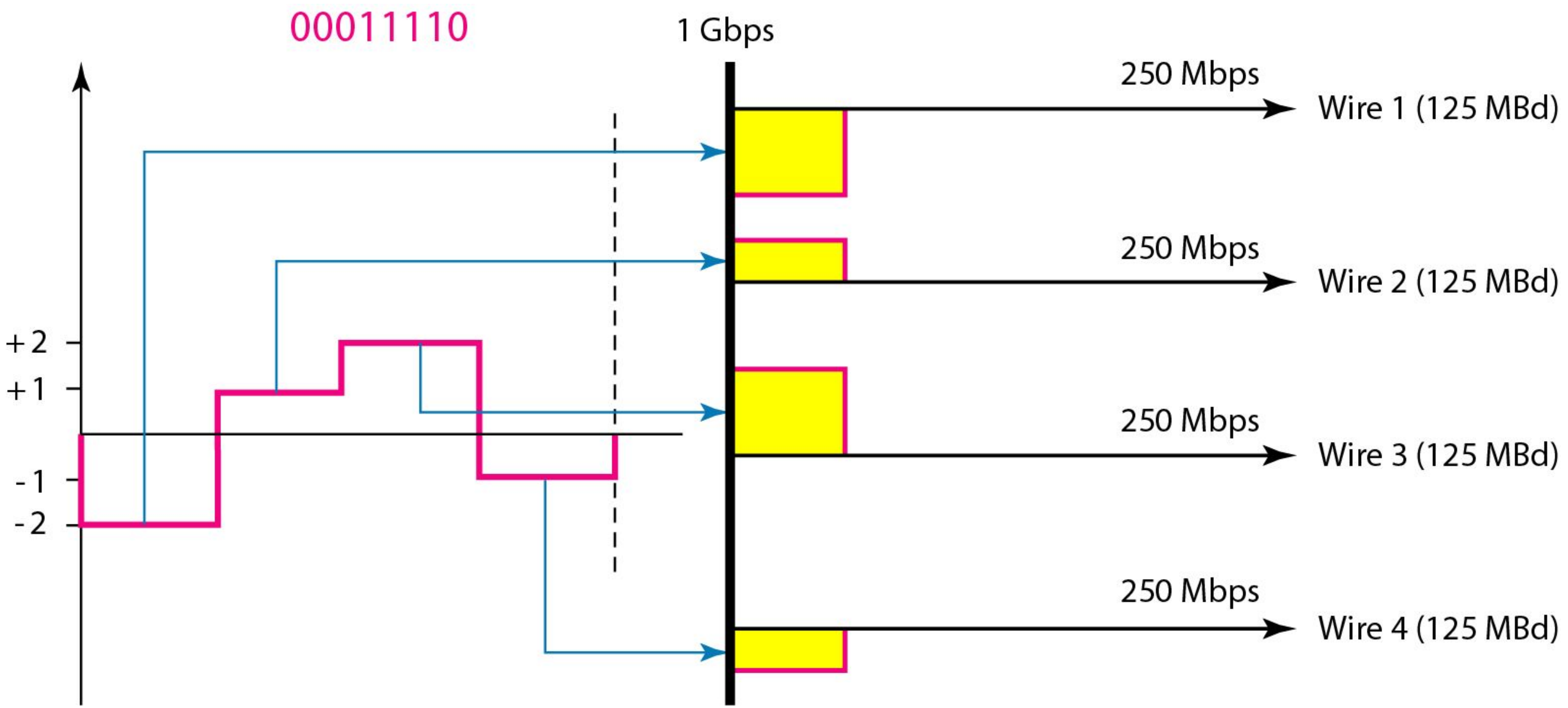
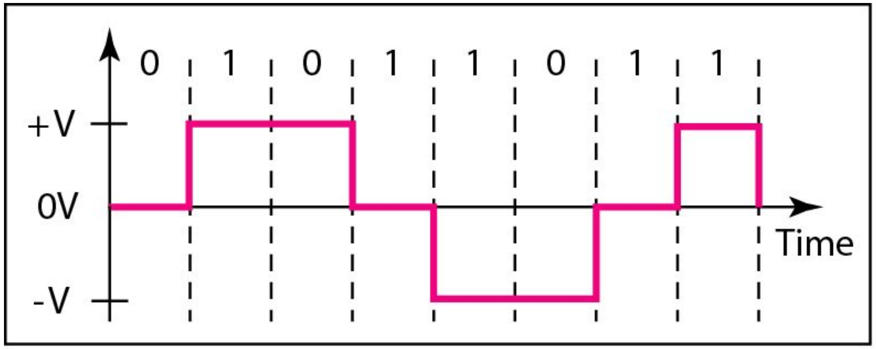
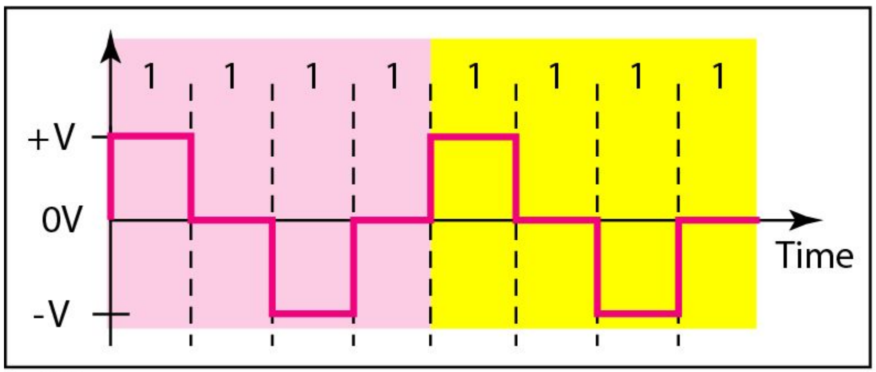


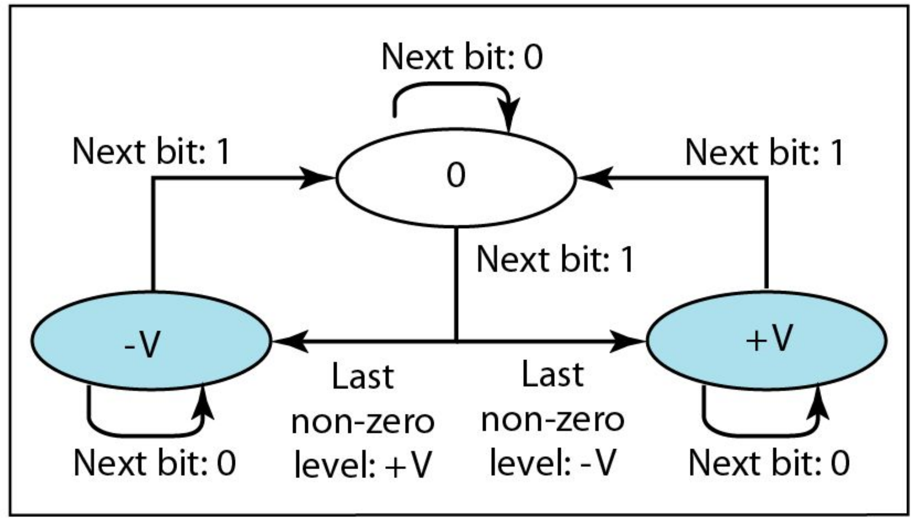
Figure 4.13 *Multitransition: MLT-3 scheme*



a. Typical case



b. Worse case



c. Transition states

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
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

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*

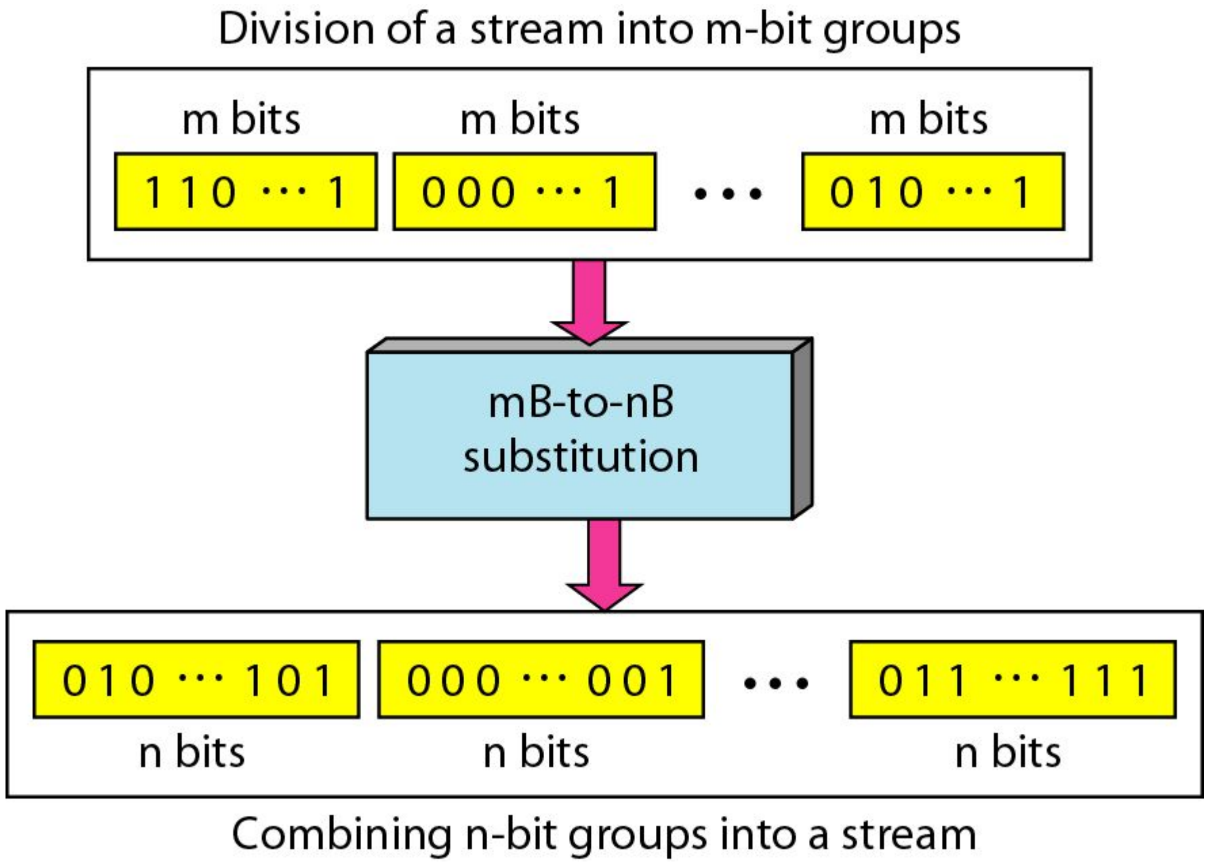


Figure 4.15 *Using block coding 4B/5B with NRZ-I line coding scheme*

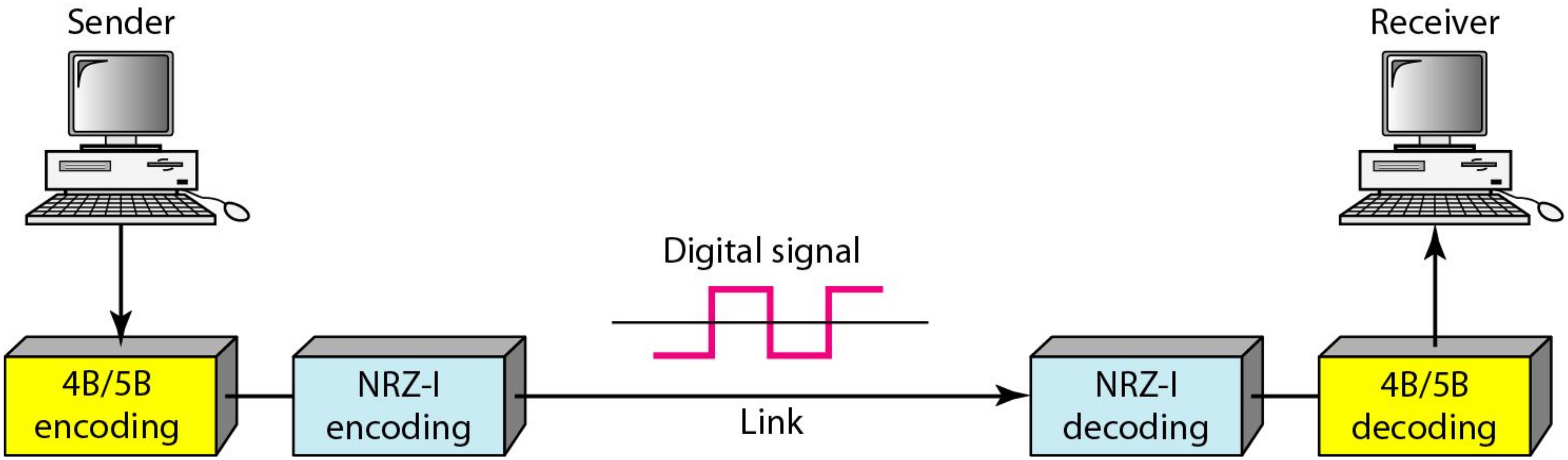
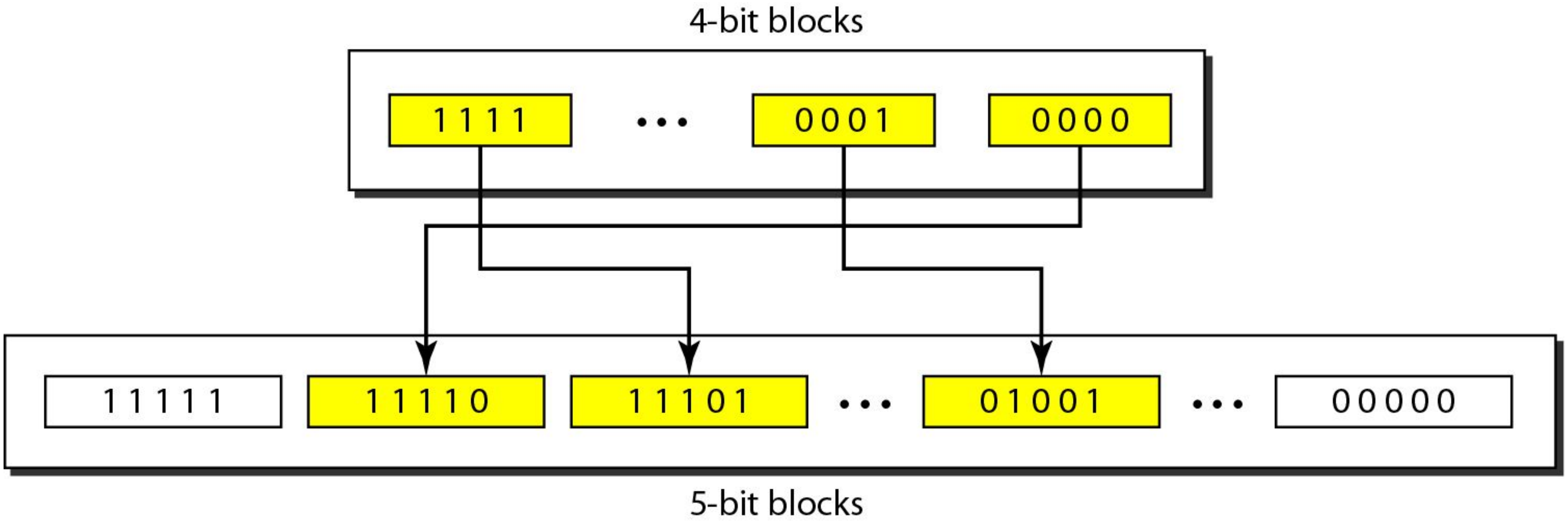


Table 4.2 *4B/5B mapping codes*

Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
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		

Figure 4.16 *Substitution in 4B/5B block coding*



Example 4.5

We need to send data at a 1-Mbps rate. What is the minimum required bandwidth, using a combination of 4B/5B and NRZ-I or Manchester coding?

Solution

First 4B/5B block coding increases the bit rate to 1.25 Mbps. The minimum bandwidth using NRZ-I is N/2 or 625 kHz. The Manchester scheme needs a minimum bandwidth of 1 MHz. The first choice needs a lower bandwidth, but has a DC component problem; the second choice needs a higher bandwidth, but does not have a DC component problem.

Figure 4.17 *8B/10B block encoding*

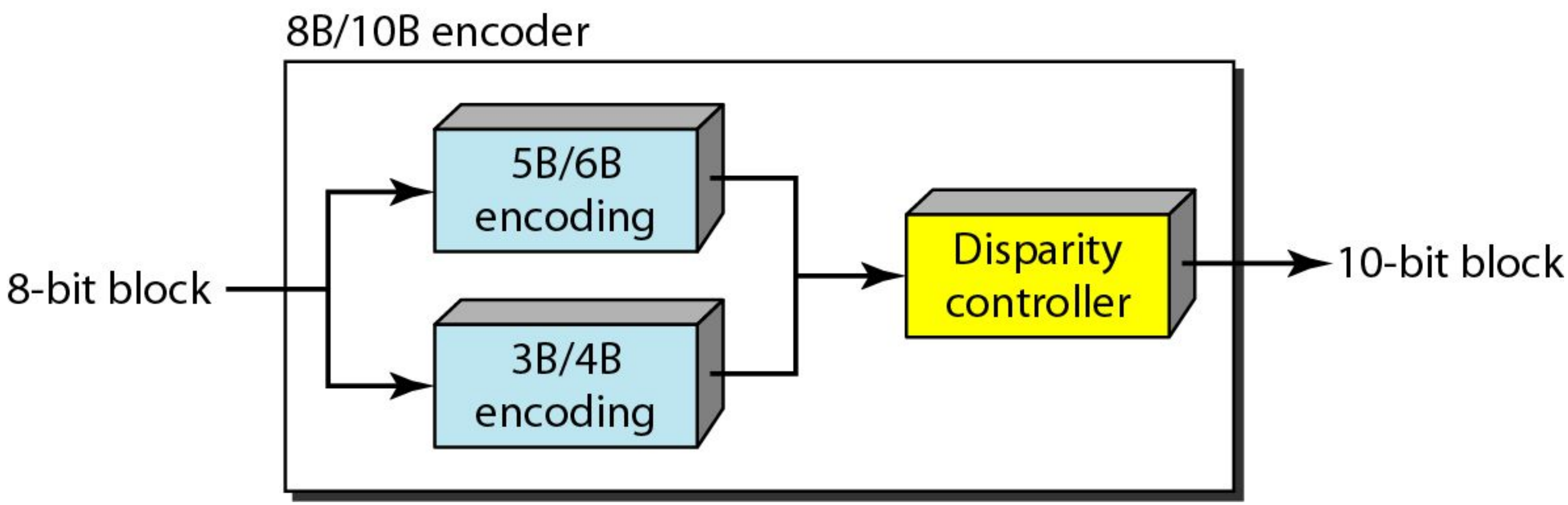


Figure 4.18 *AMI used with scrambling*

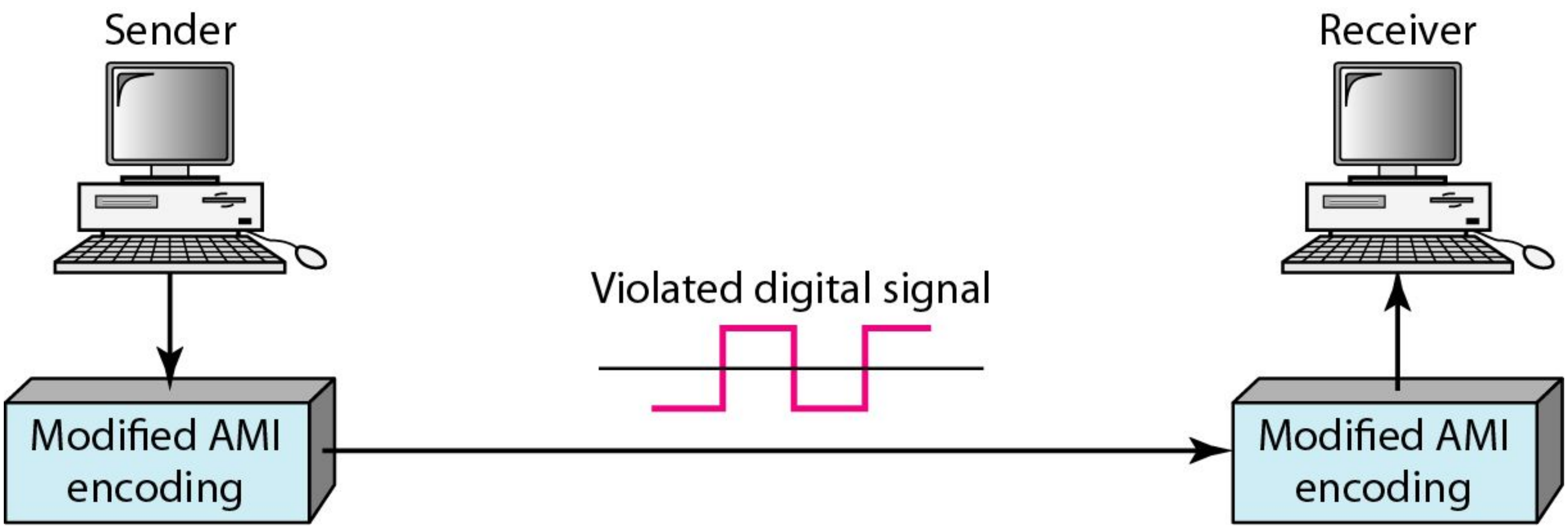
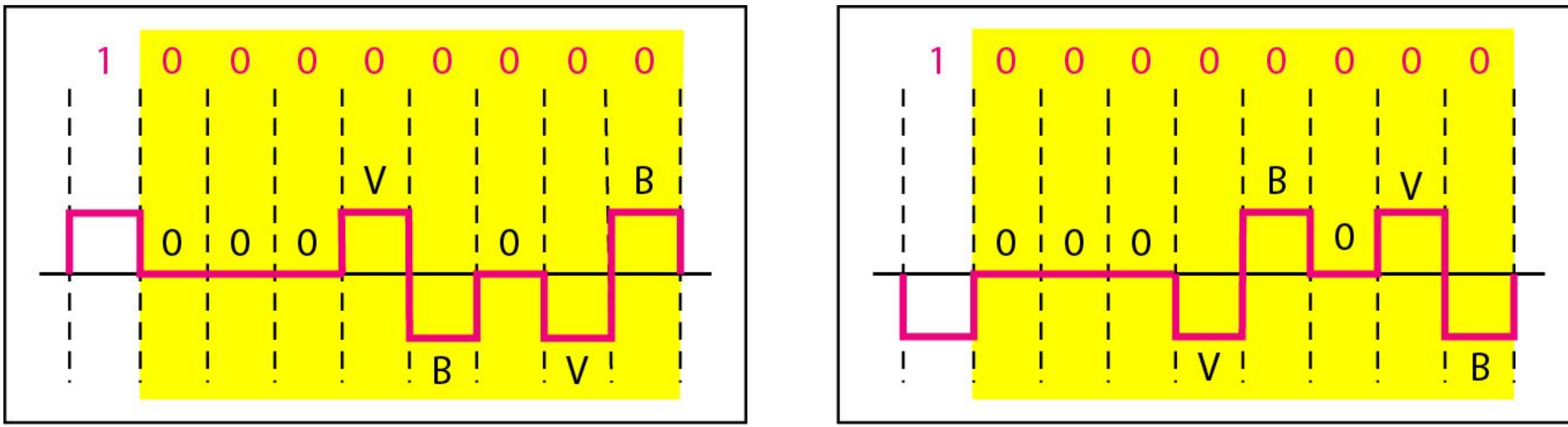


Figure 4.19 *Two cases of B8ZS scrambling technique*



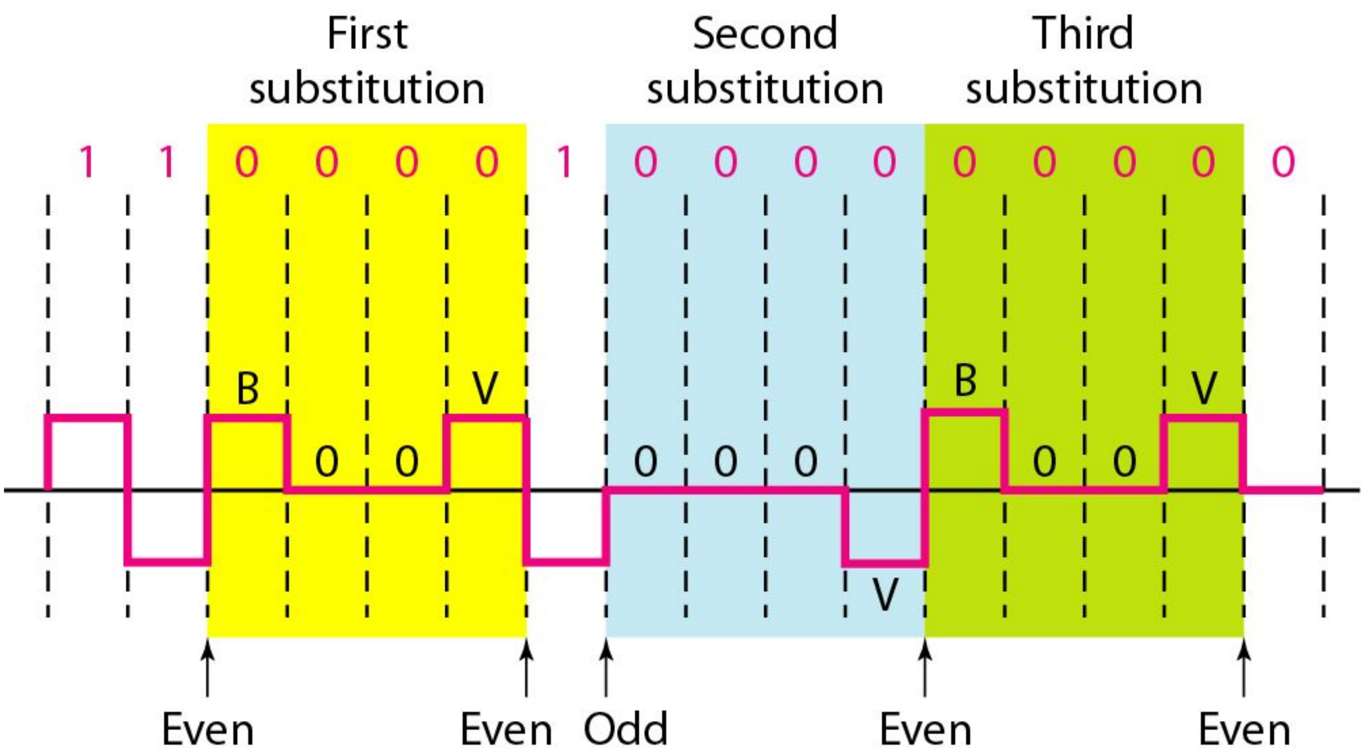
a. Previous level is positive.

b. Previous level is negative.

Note

B8ZS substitutes eight consecutive zeros with 000VB0VB.

Figure 4.20 *Different situations in HDB3 scrambling technique*



Note

HDB3 substitutes four consecutive zeros with 000V or B00V depending on the number of nonzero pulses after the last substitution.

5-1 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.

~~*Topics discussed in this section:*~~

Aspects of Digital-to-Analog Conversion

Amplitude Shift Keying

Frequency Shift Keying

Phase Shift Keying

Quadrature Amplitude Modulation

Figure 5.1 *Digital-to-analog conversion*

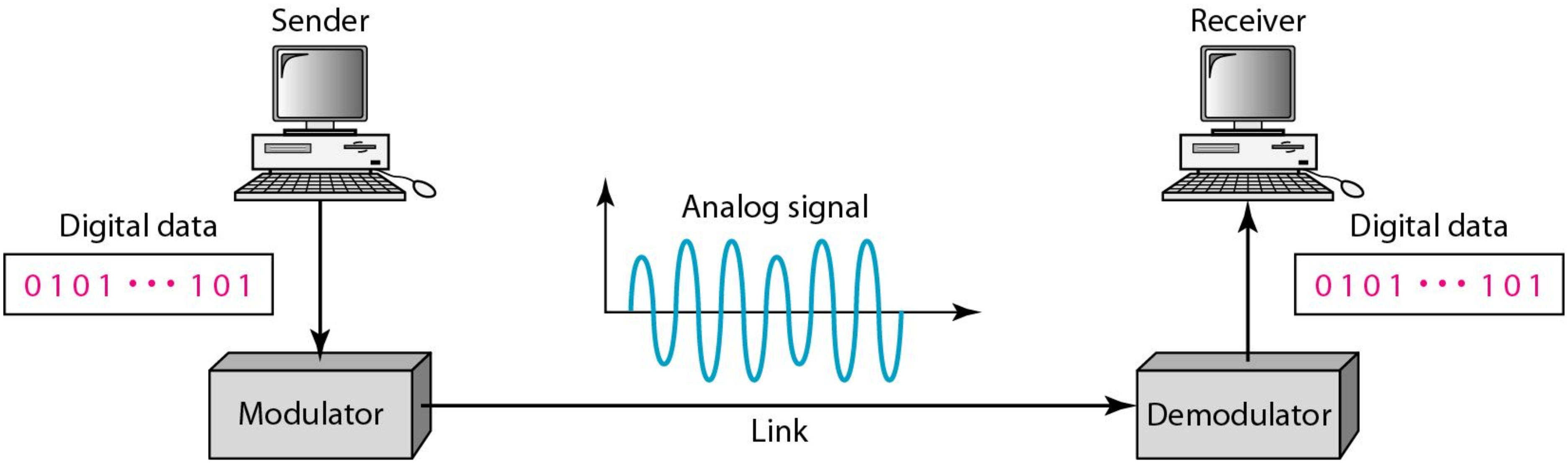
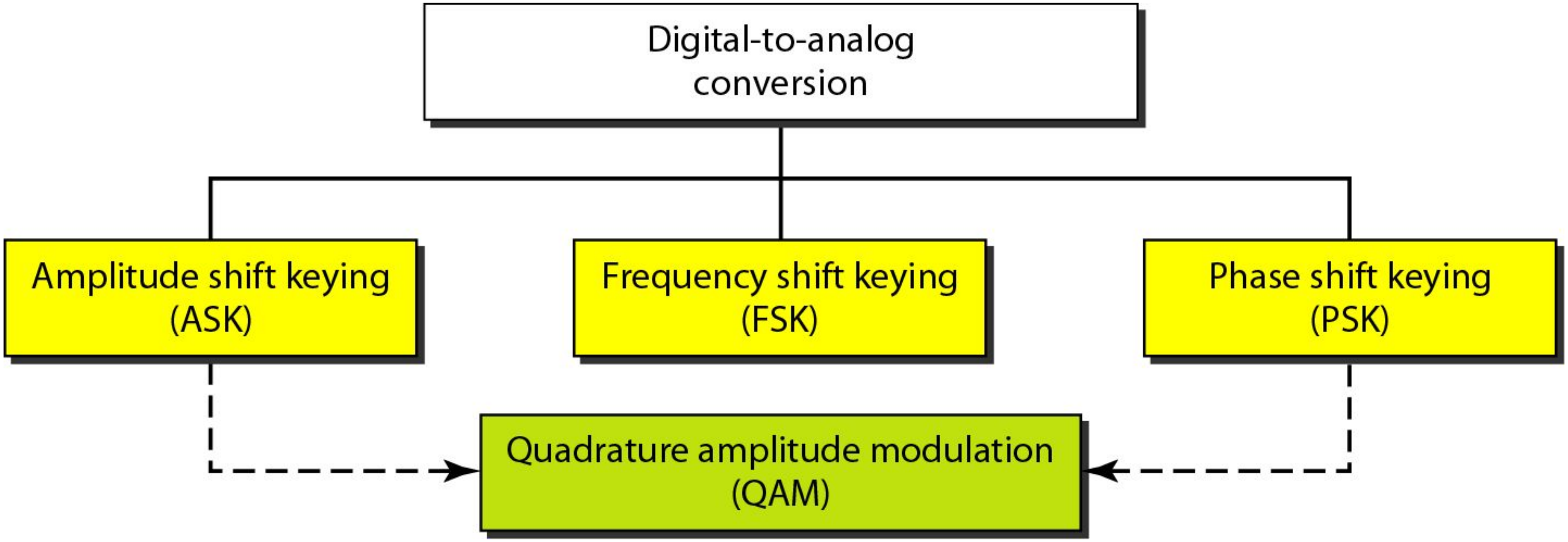


Figure 5.2 *Types of digital-to-analog conversion*



Note

Bit rate is the number of bits per second. Baud rate is the number of signal elements per second.

In the analog transmission of digital data, the baud rate is less than or equal to the bit rate.

Example 5.1

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.

Solution

In this case, r = 4, S = 1000, and N is unknown. We can find the value of N from

$$\boxed{S = N \times \frac{1}{r} \qquad \text{or} \qquad N = S \times r = 1000 \times 4 = 4000 \text{ bps}}$$

Example 5.2

An analog signal has a bit rate of 8000 bps and a baud rate of 1000 baud. How many data elements are carried by each signal element? How many signal elements do we need?

Solution

In this example, S = 1000, N = 8000, and r and L are unknown. We find first the value of r and then the value of L.

$$S = N \times \frac{1}{r} \quad \Rightarrow \quad r = \frac{N}{S} = \frac{8000}{1000} = 8 \text{ bits/baud}$$

$$r = \log_2 L \quad \Rightarrow \quad L = 2^r = 2^8 = 256$$

Figure 5.3 *Binary amplitude shift keying*

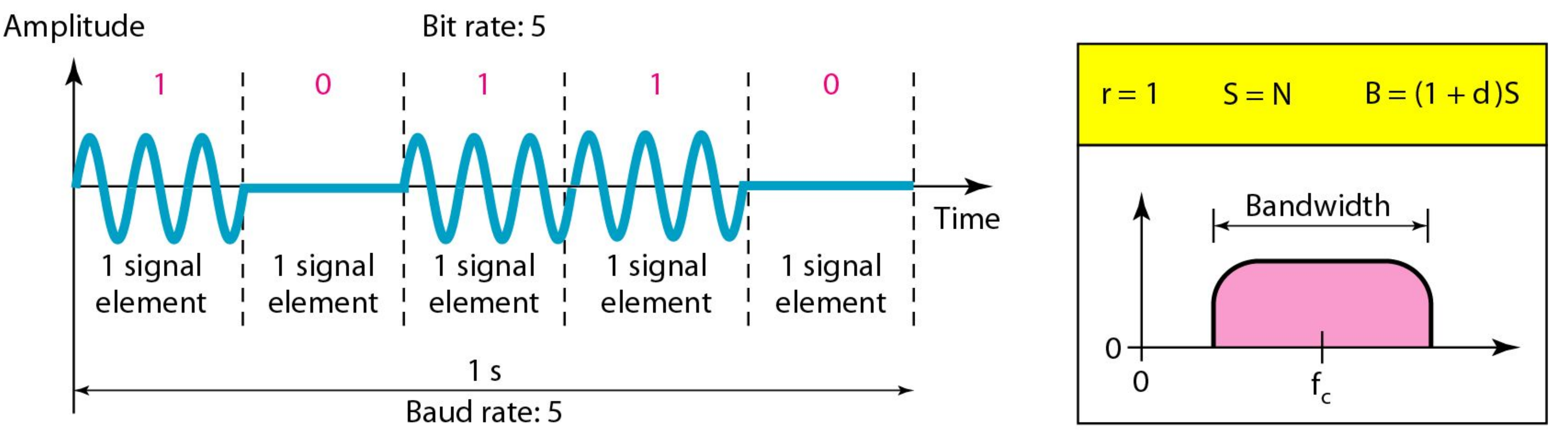
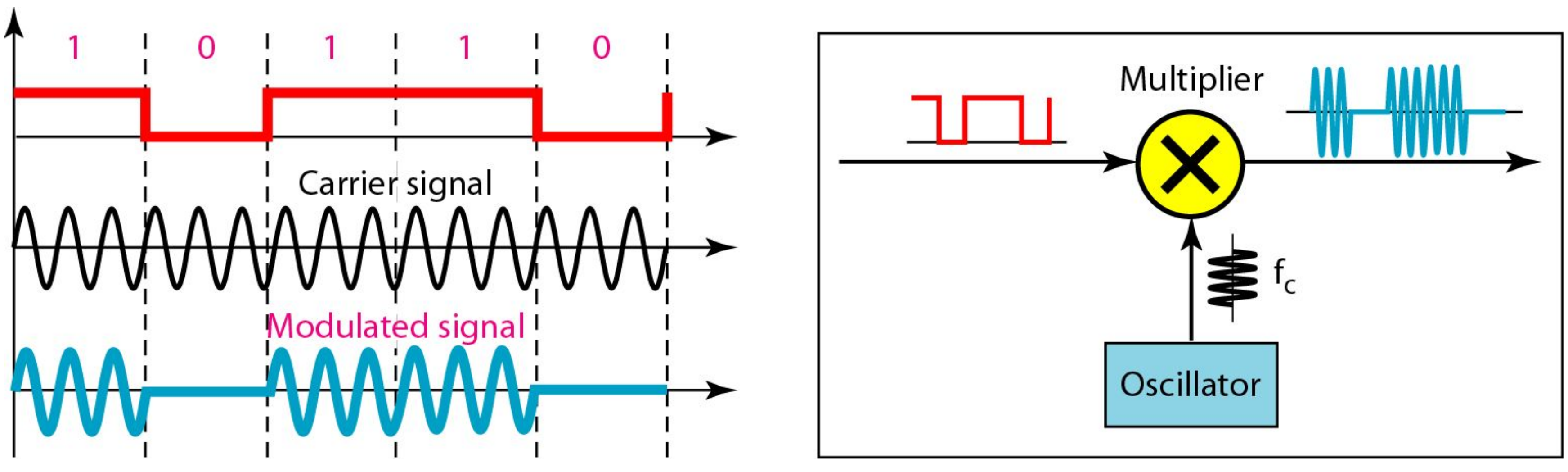


Figure 5.4 *Implementation of binary ASK*



We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What are the carrier frequency and the bit rate if we modulated our data by using ASK with d = 1?

Solution

The middle of the bandwidth is located at 250 kHz. This means that our carrier frequency can be at fc = 250 kHz. We can use the formula for bandwidth to find the bit rate (with d = 1 and r = 1).

$$\boxed{B = (1 + d) \times S = 2 \times N \times \frac{1}{r} = 2 \times N = 100 \text{ kHz} \quad \Rightarrow \quad N = 50 \text{ kbps}}$$

Example 5.4

In data communications, we normally use full-duplex links with communication in both directions. We need to divide the bandwidth into two with two carrier frequencies, as shown in Figure 5.5. The figure shows the positions of two carrier frequencies and the bandwidths. The available bandwidth for each direction is now 50 kHz, which leaves us with a data rate of 25 kbps in each direction.

Figure 5.5 *Bandwidth of full-duplex ASK used in Example 5.4*

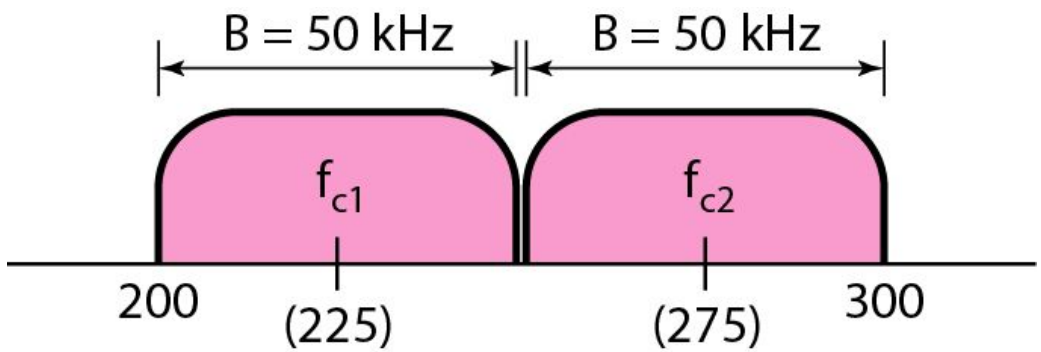
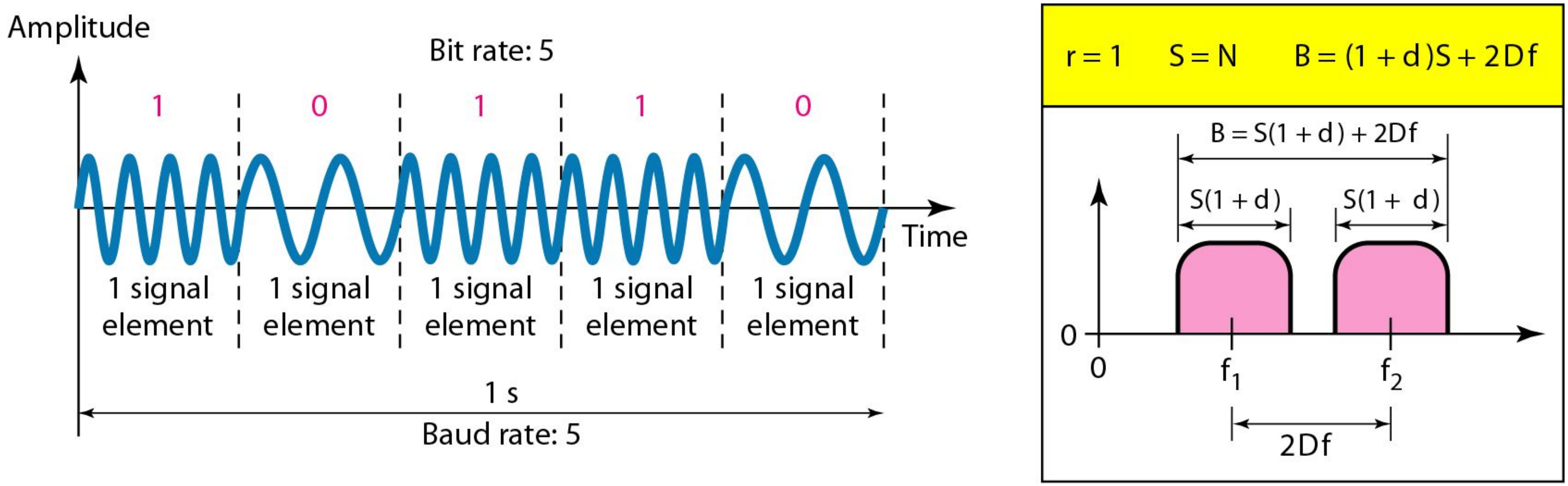


Figure 5.6 *Binary frequency shift keying*



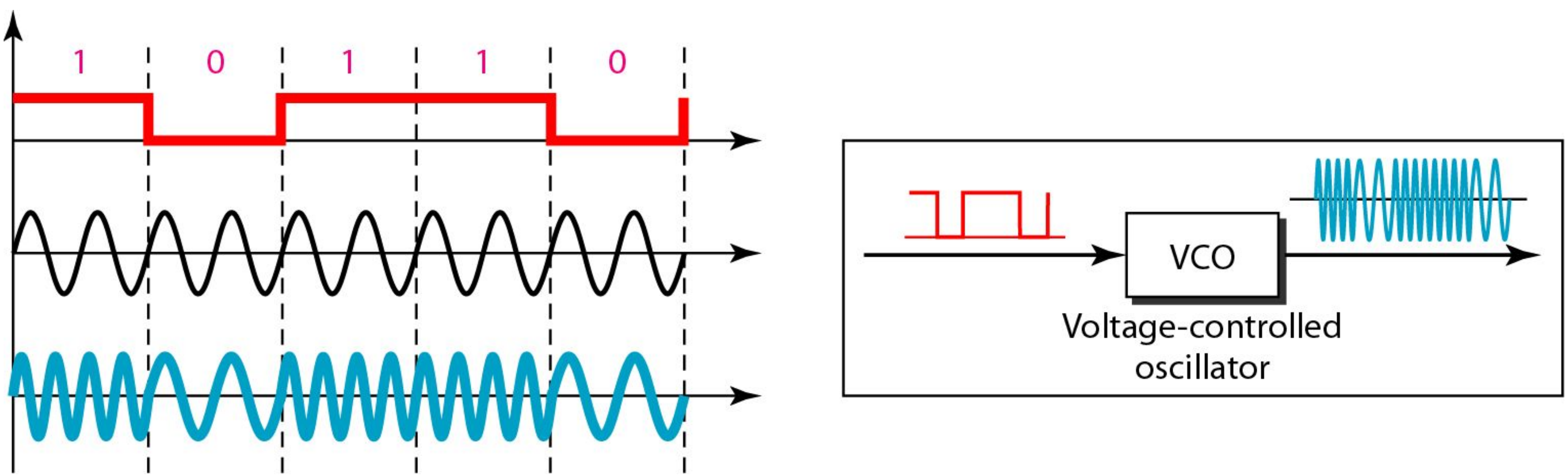
We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with d = 1?

Solution

This problem is similar to Example 5.3, but we are modulating by using FSK. The midpoint of the band is at 250 kHz. We choose 2Δf to be 50 kHz; this means

$$B = (1 + d) \times S + 2\Delta f = 100 \longrightarrow 2S = 50 \text{ kHz} \quad S = 25 \text{ kbaud} \quad N = 25 \text{ kbps}$$

Figure 5.7 *Bandwidth of MFSK used in Example 5.6*



We need to send data 3 bits at a time at a bit rate of 3 Mbps. The carrier frequency is 10 MHz. Calculate the number of levels (different frequencies), the baud rate, and the bandwidth.

Solution

We can have L = 2^3 = 8. The baud rate is S = 3 MHz/3 = 1000 Mbaud. This means that the carrier frequencies must be 1 MHz apart (2Δf = 1 MHz). The bandwidth is B = 8 × 1000 = 8000. Figure 5.8 shows the allocation of frequencies and bandwidth.

Figure 5.8 *Bandwidth of MFSK used in Example 5.6*

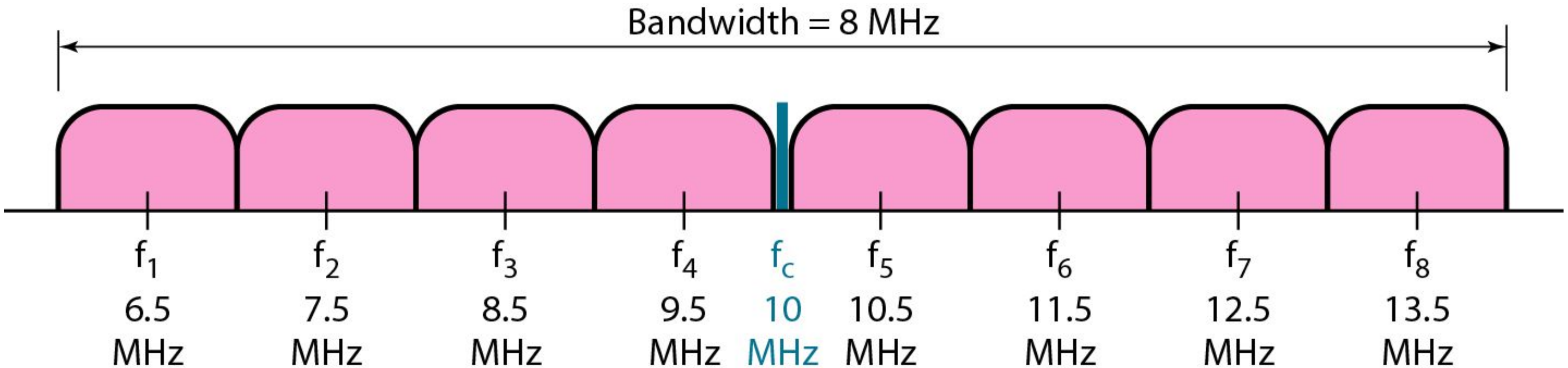
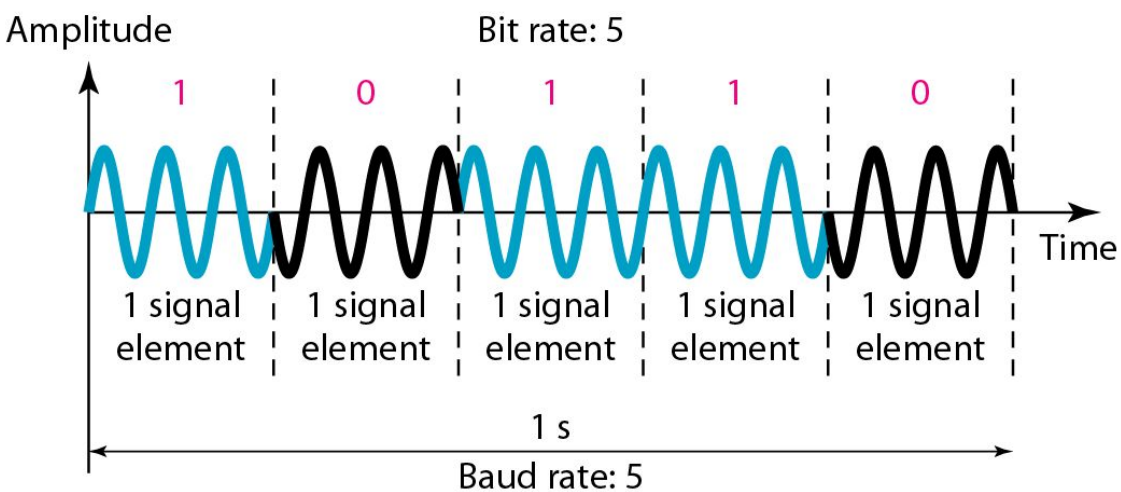


Figure 5.9 *Binary phase shift keying*



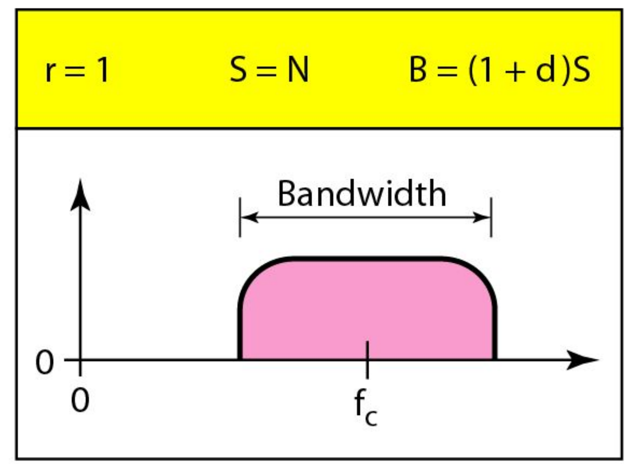


Figure 5.10 *Implementation of BASK*

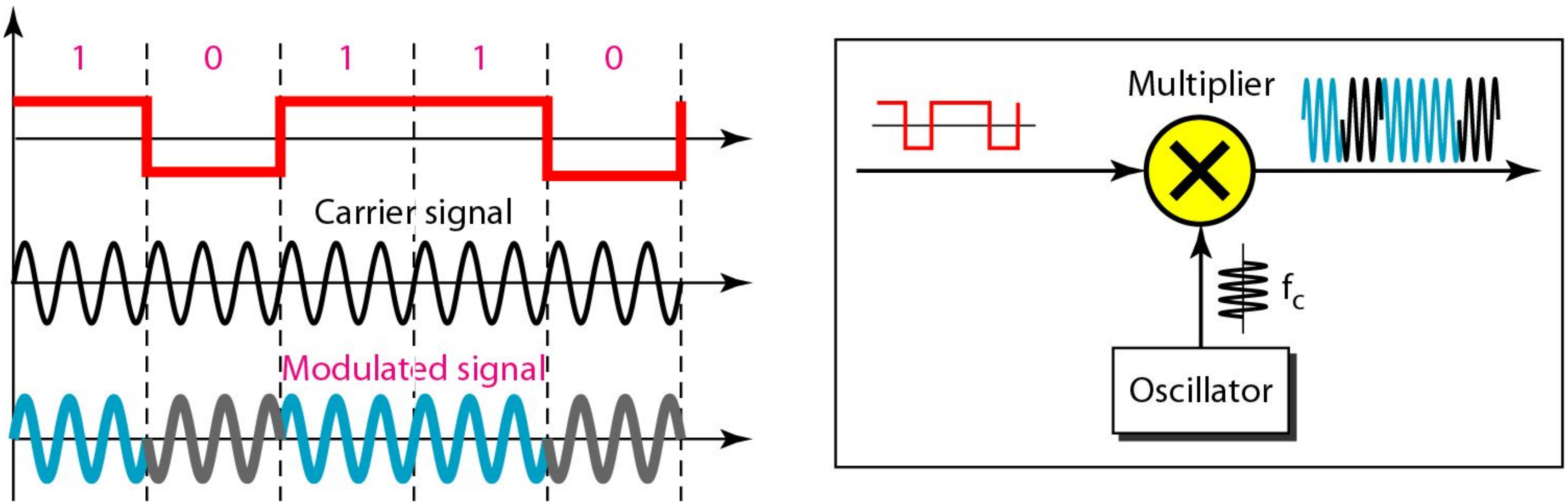
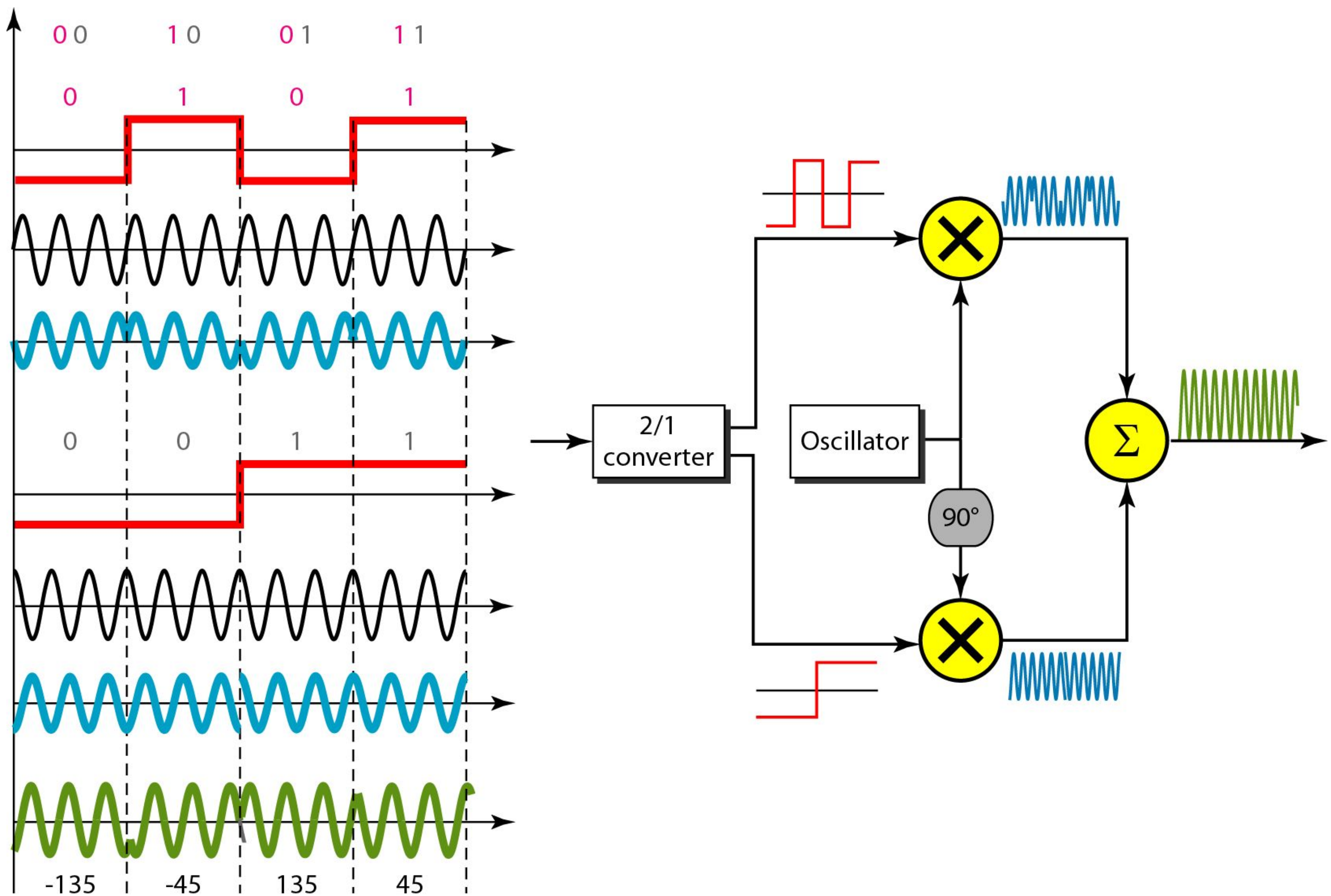


Figure 5.11 *QPSK and its implementation*



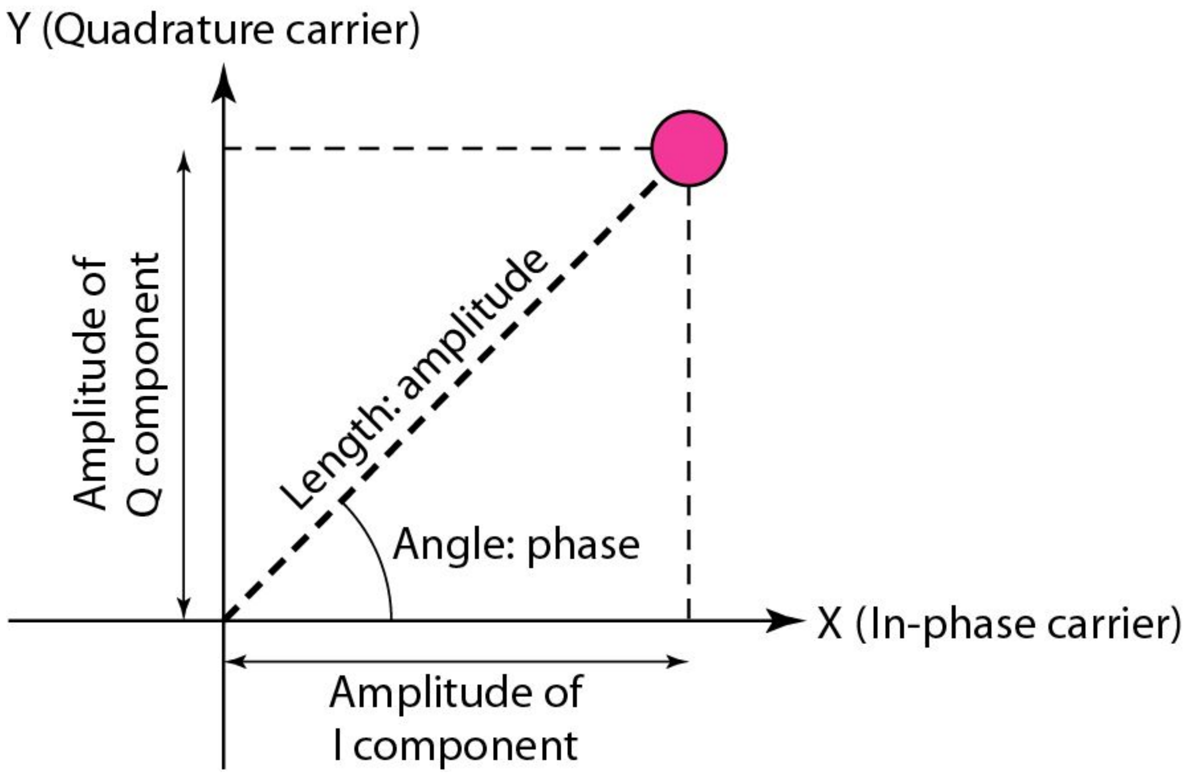
Example 5.7

Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of $d = 0$.

Solution

For QPSK, 2 bits is carried by one signal element. This means that $r = 2$. So the signal rate (baud rate) is $S = N \times (1/r) = 6$ Mbaud. With a value of $d = 0$, we have $B = S = 6$ MHz.

Figure 5.12 *Concept of a constellation diagram*

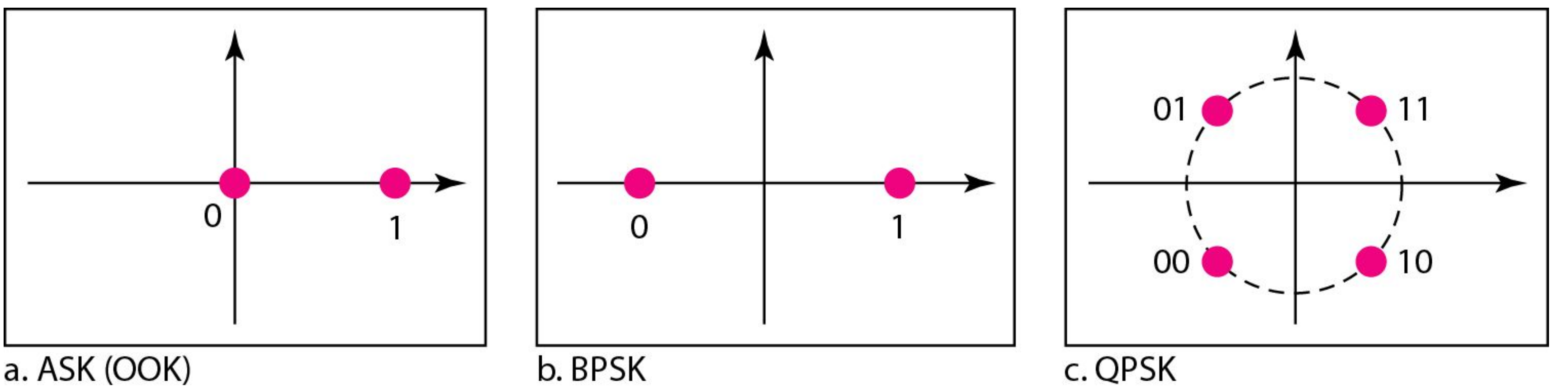


Show the constellation diagrams for an ASK (OOK), BPSK, and QPSK signals.

Solution

Figure 5.13 shows the three constellation diagrams.

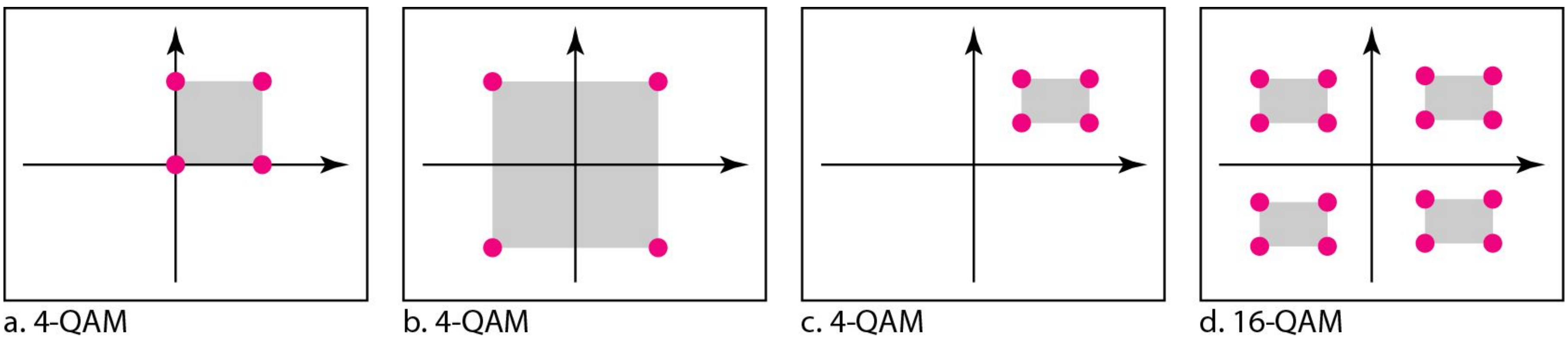
Figure 5.13 *Three constellation diagrams*



Note

Quadrature amplitude modulation is a combination of ASK and PSK.

Figure 5.14 *Constellation diagrams for some QAMs*



5-2 ANALOG AND DIGITAL

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.

~~*Topics discussed in this section:*~~

Amplitude Modulation

Frequency Modulation

Phase Modulation

Figure 5.15 *Types of analog-to-analog modulation*

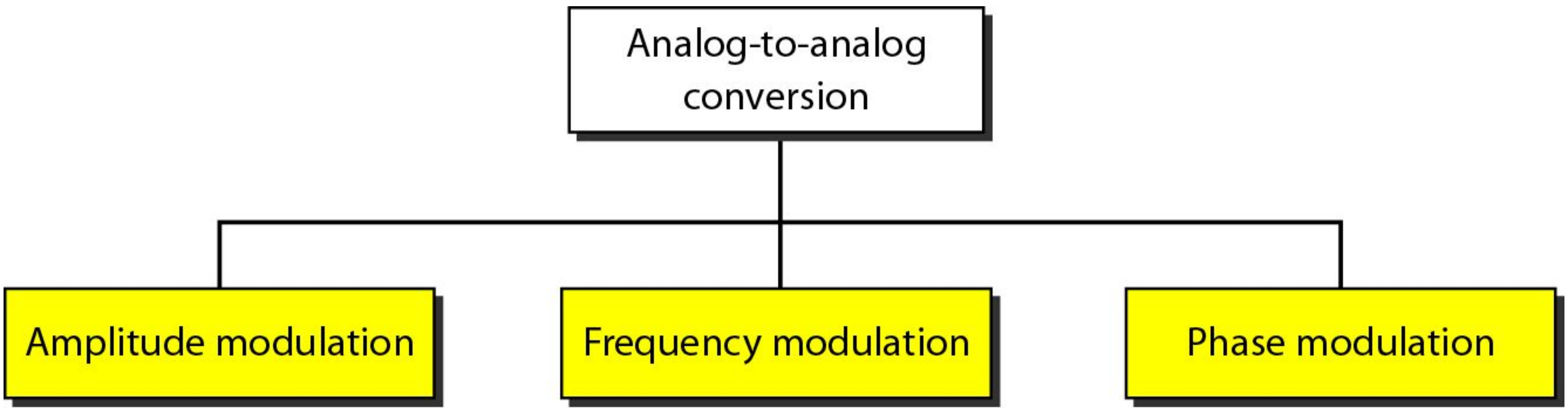
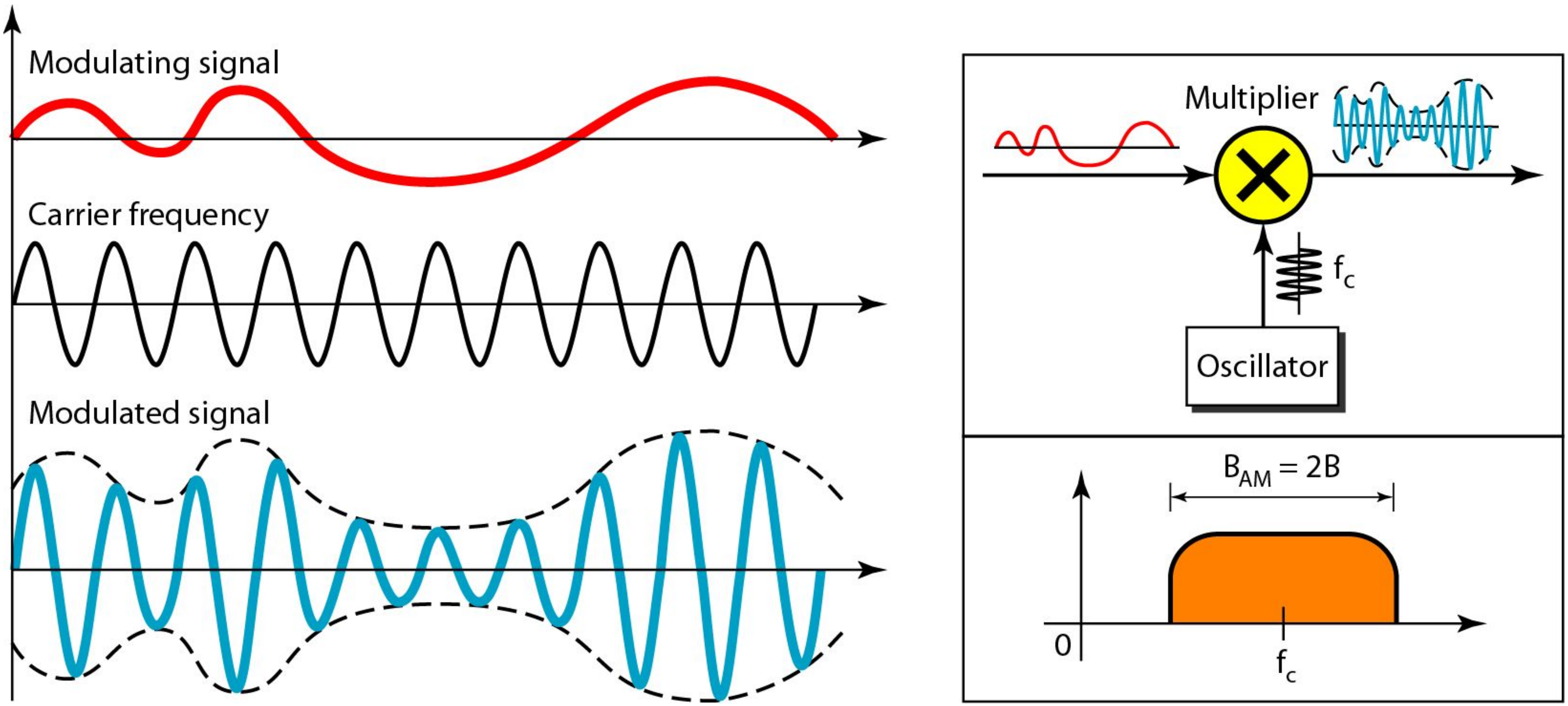


Figure 5.16 *Amplitude modulation*



Note

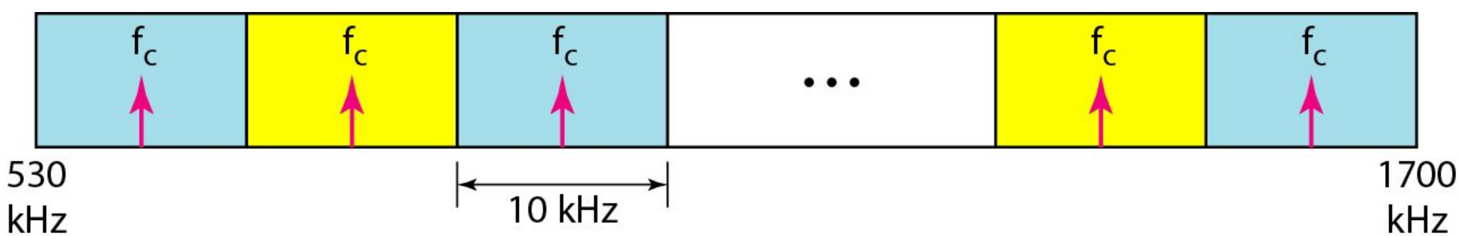
The total bandwidth required for AM

can be determined

from the bandwidth of the audio

signal: BAM = 2B.

Figure 5.17 *AM band allocation*



Note

The total bandwidth required for FM can be determined from the bandwidth of the audio signal: $B_{FM} = 2(1 + \beta)B$.

Figure 5.18 *Frequency modulation*

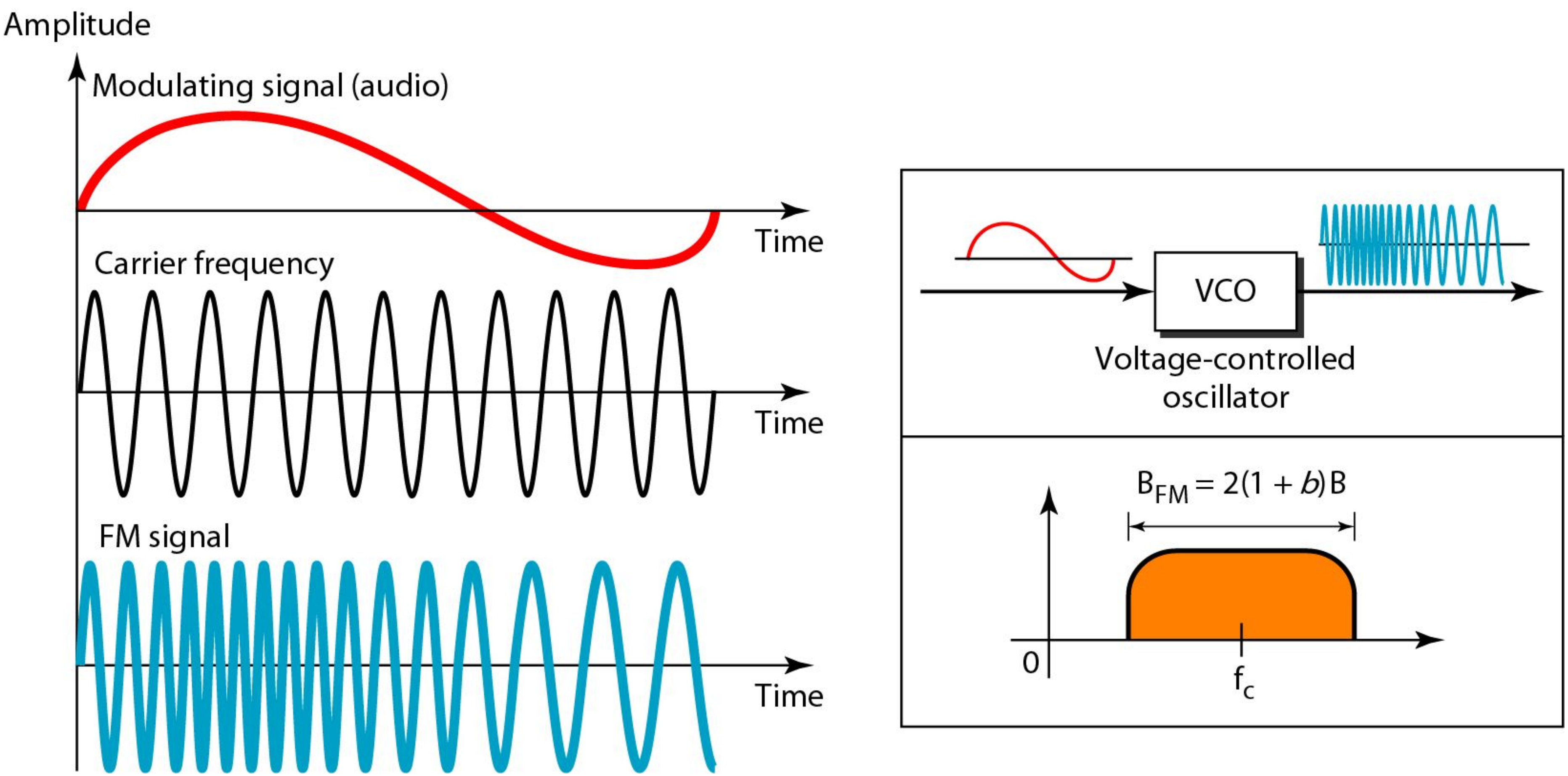


Figure 5.19 *FM band allocation*

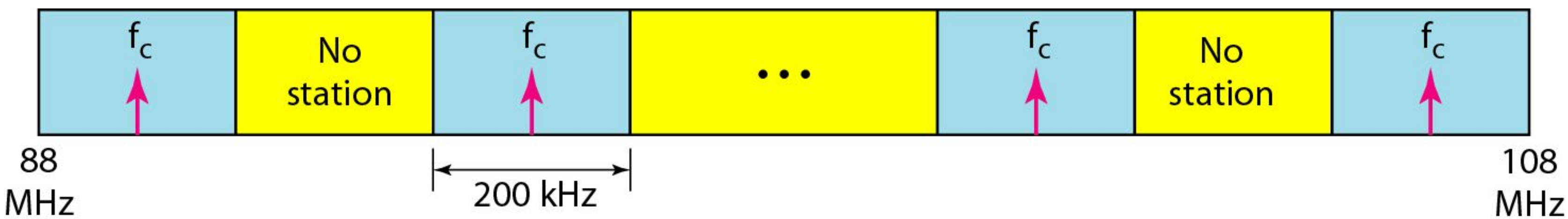
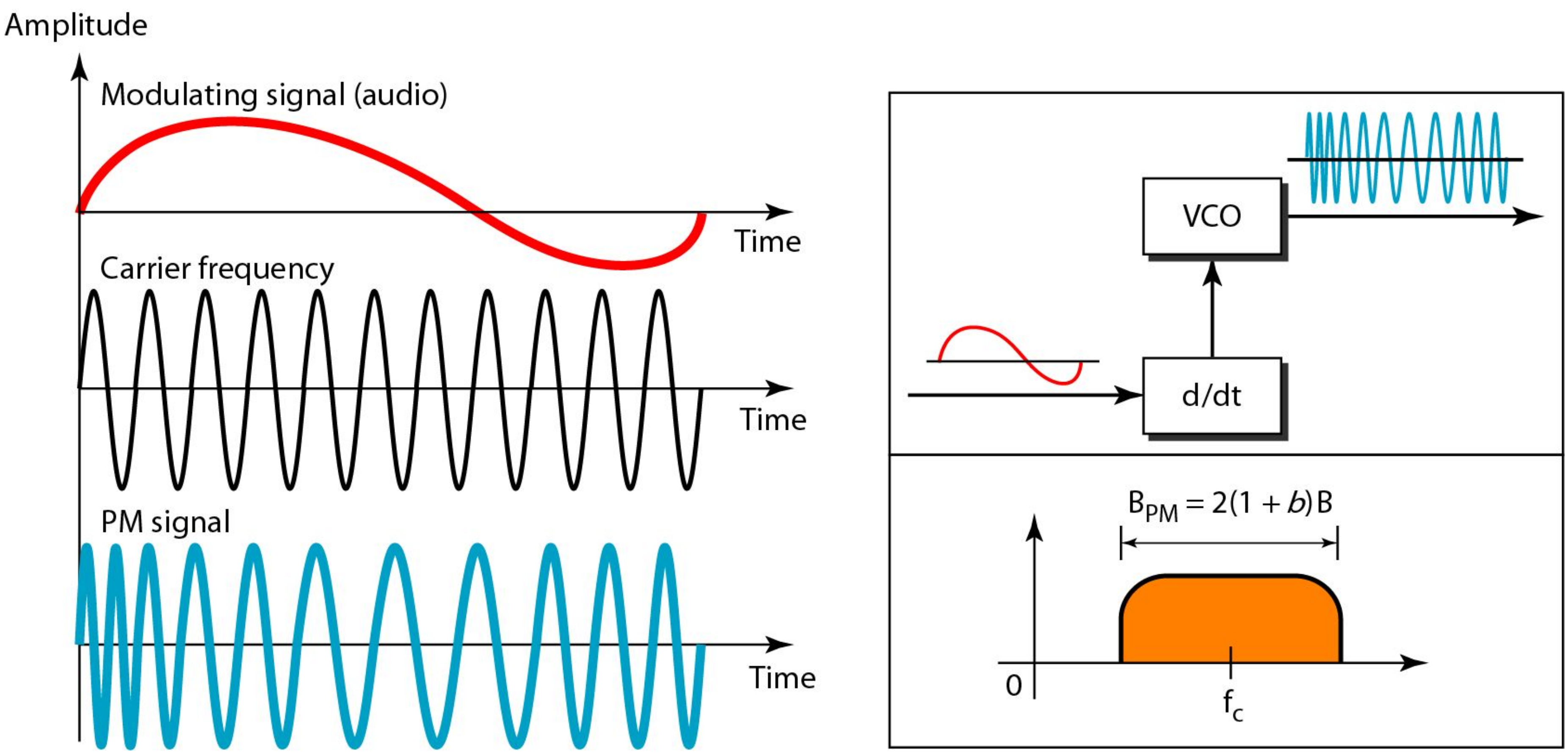


Figure 5.20 *Phase modulation*



Note

The total bandwidth required for PM can be determined from the bandwidth and maximum amplitude of the modulating signal:

$$B_{PM} = 2(1 + \beta)B.$$

Note

Bandwidth utilization is the wise use of
available bandwidth to achieve
specific goals.

Efficiency can be achieved by multiplexing; privacy and anti-jamming can be achieved by spreading.

6-1 MULTIPLEXING

Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared. Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. As data and telecommunications use increases, so does traffic.

~~*Topics discussed in this section:*~~

Frequency-Division Multiplexing

Wavelength-Division Multiplexing

Synchronous Time-Division Multiplexing

Statistical Time-Division Multiplexing

Figure 6.1 *Dividing a link into channels*

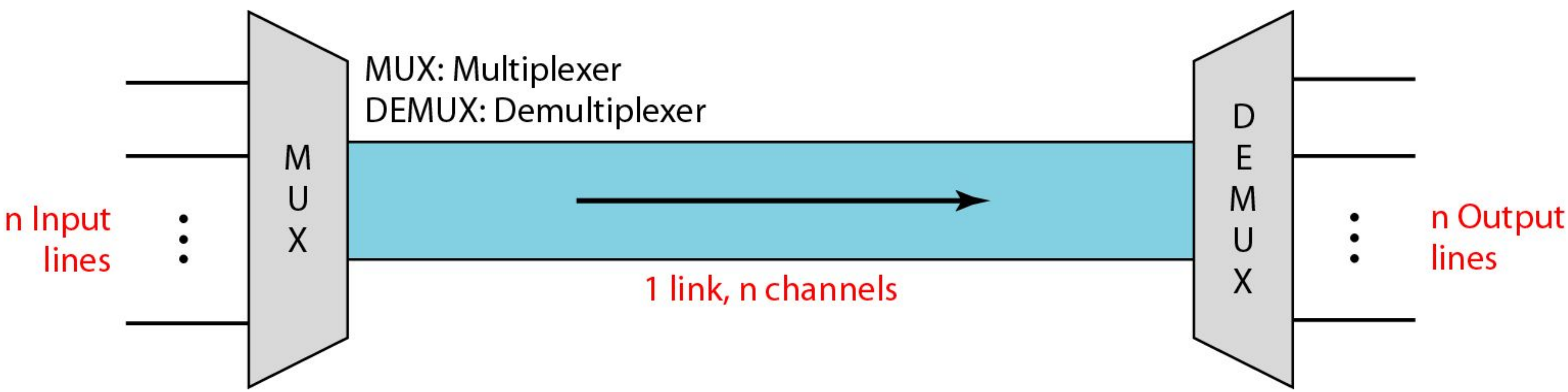


Figure 6.2 *Categories of multiplexing*

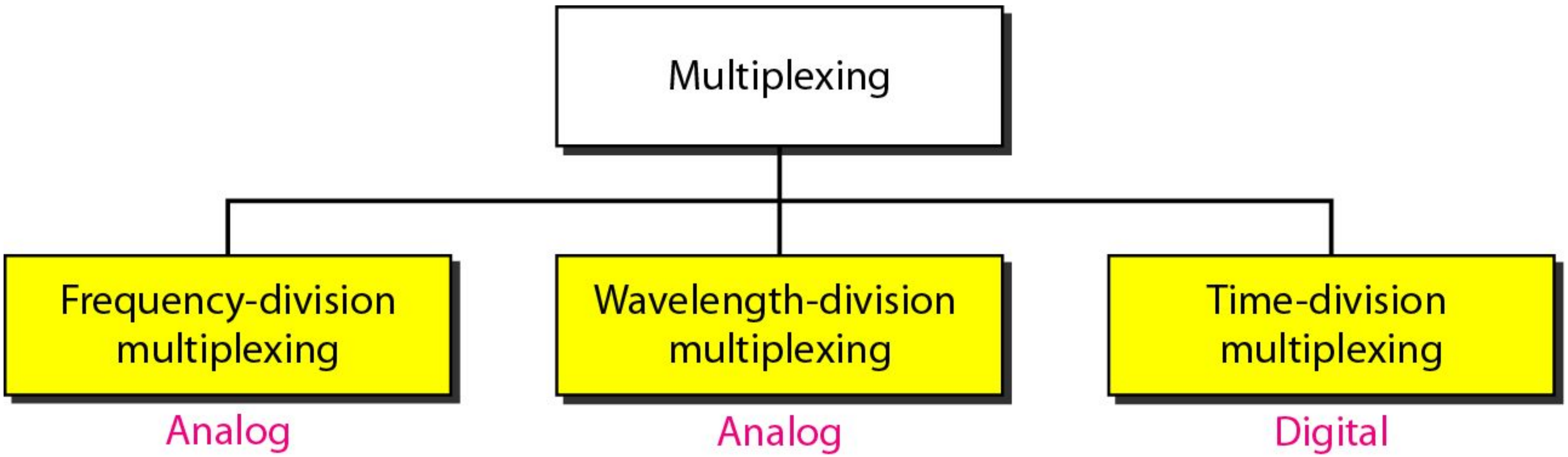


Figure 6.3 *Frequency-division multiplexing*

Note

FDM is an analog multiplexing technique that combines analog signals.

Figure 6.4 *FDM process*

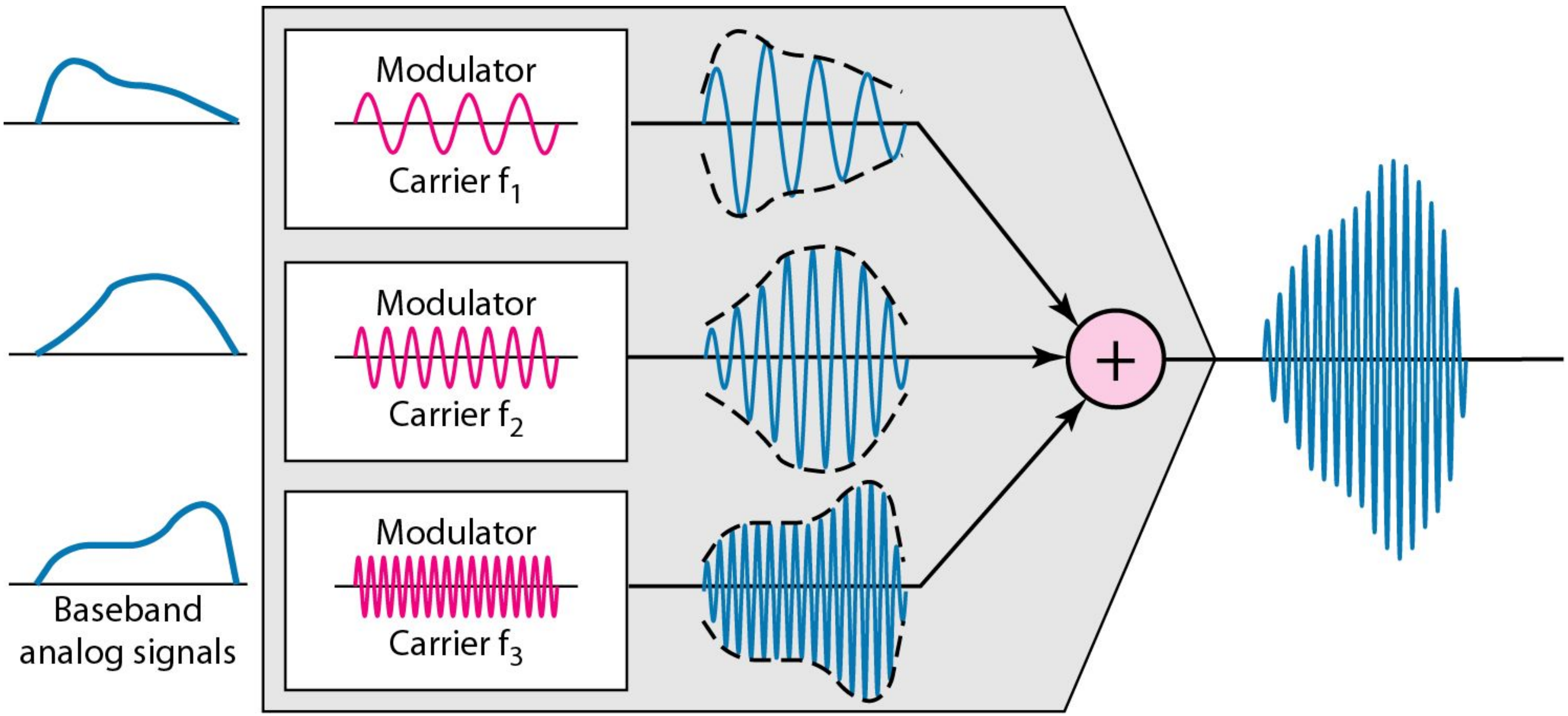
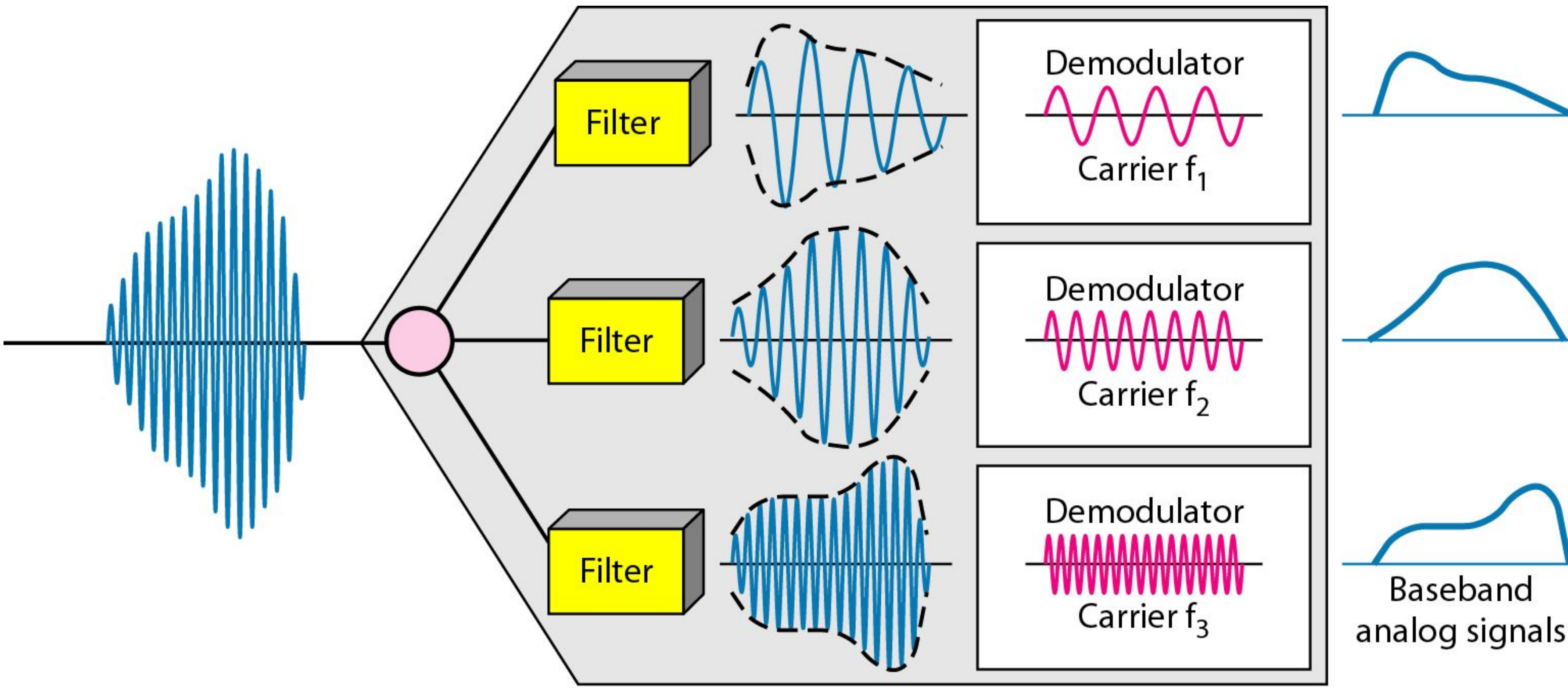


Figure 6.5 *FDM demultiplexing example*



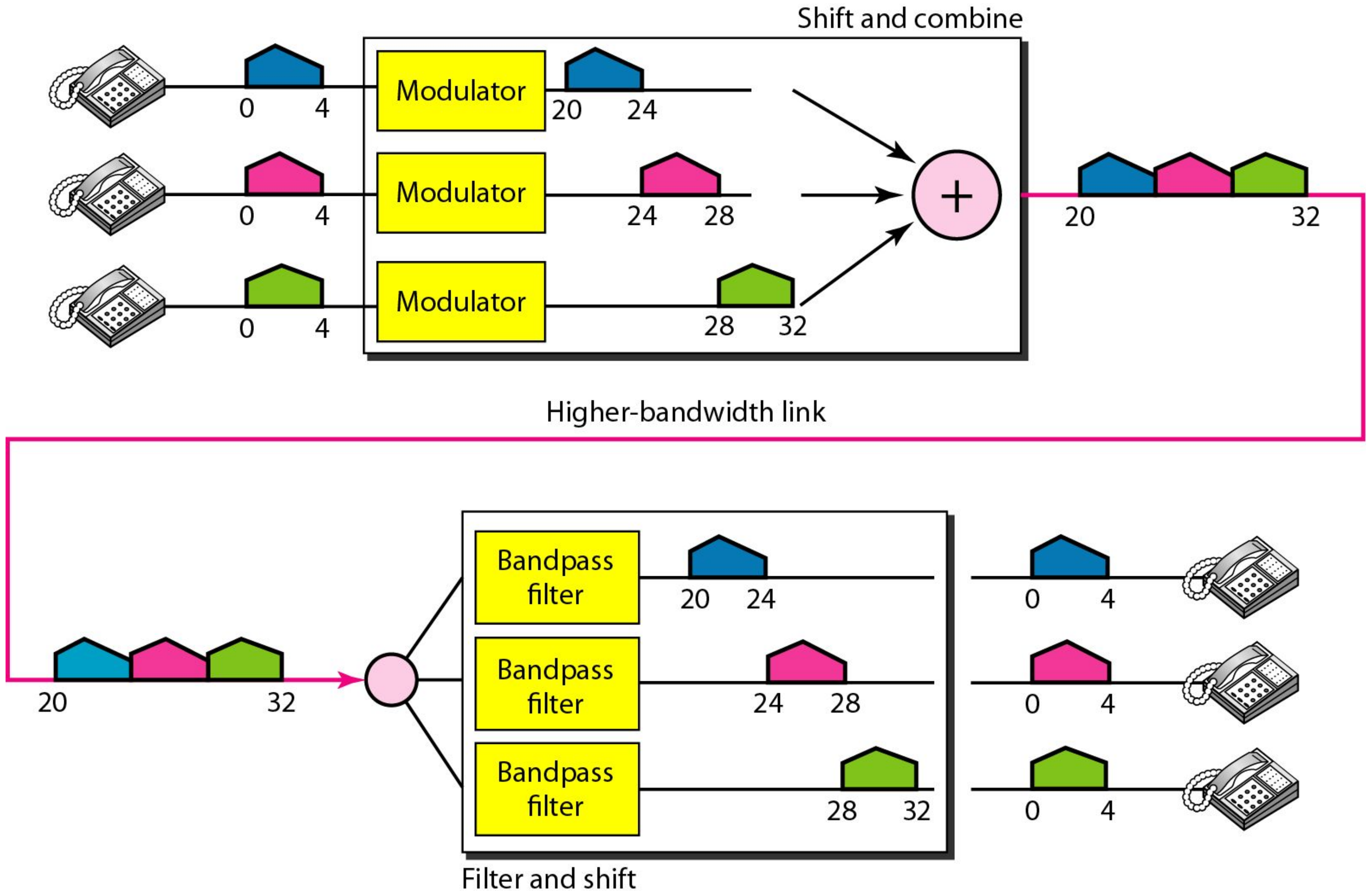
Example 6.1

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure 6.6. We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third one. Then we combine them as shown in Figure 6.6.

Figure 6.6 *Example 6.1*



Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

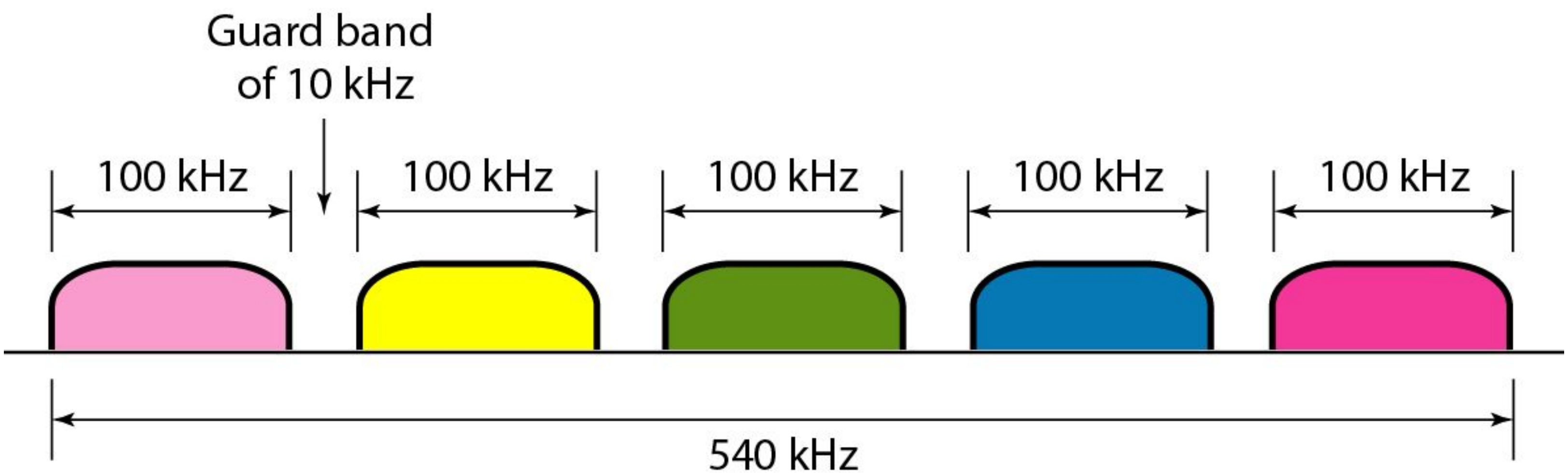
Solution

For five channels, we need at least four guard bands. This means that the required bandwidth is at least

$$5 \times 100 + 4 \times 10 = 540\ kHz,$$

as shown in Figure 6.7.

Figure 6.7 *Example 6.2*



Example 6.3

Four data channels (digital), each transmitting at 1 Mbps, use a satellite channel of 1 MHz. Design an appropriate configuration, using FDM.

Solution

The satellite channel is analog. We divide it into four channels, each channel having a 250-kHz bandwidth. Each digital channel of 1 Mbps is modulated such that each 4 bits is modulated to 1 Hz. One solution is 16-QAM modulation. Figure 6.8 shows one possible configuration.

Figure 6.8 *Example 6.3*

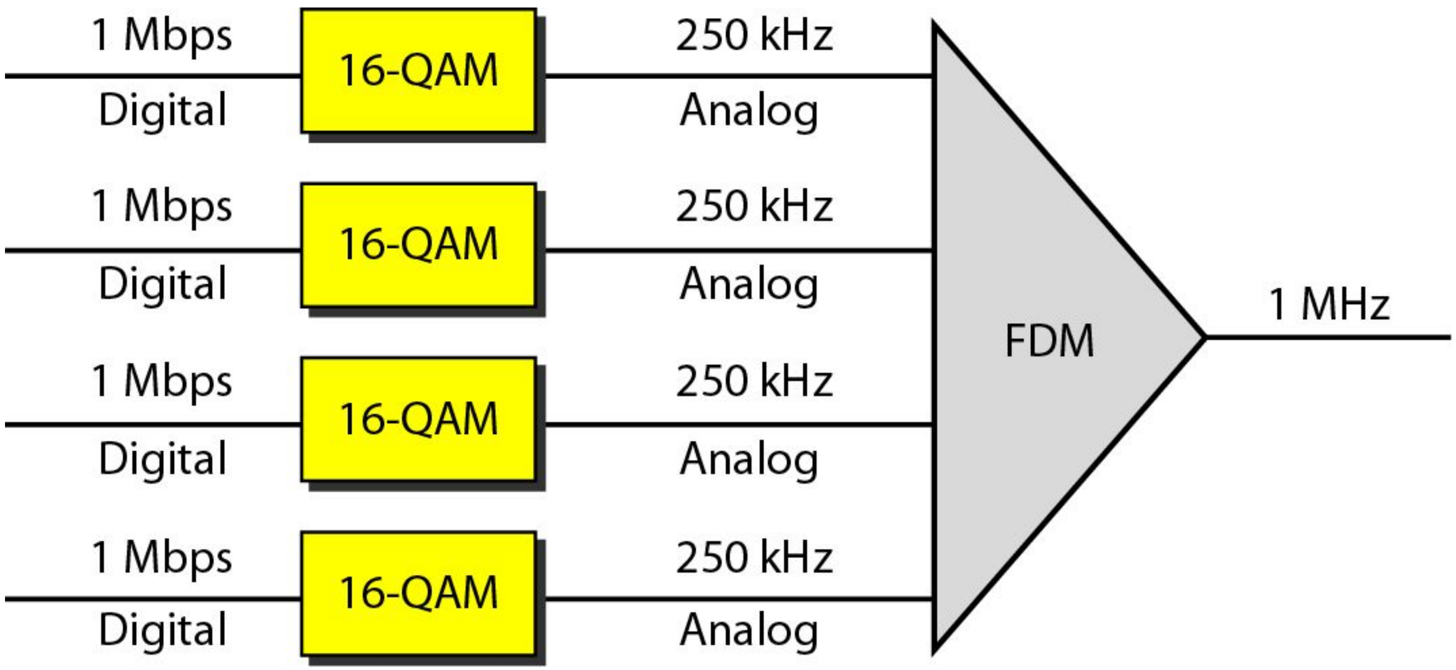
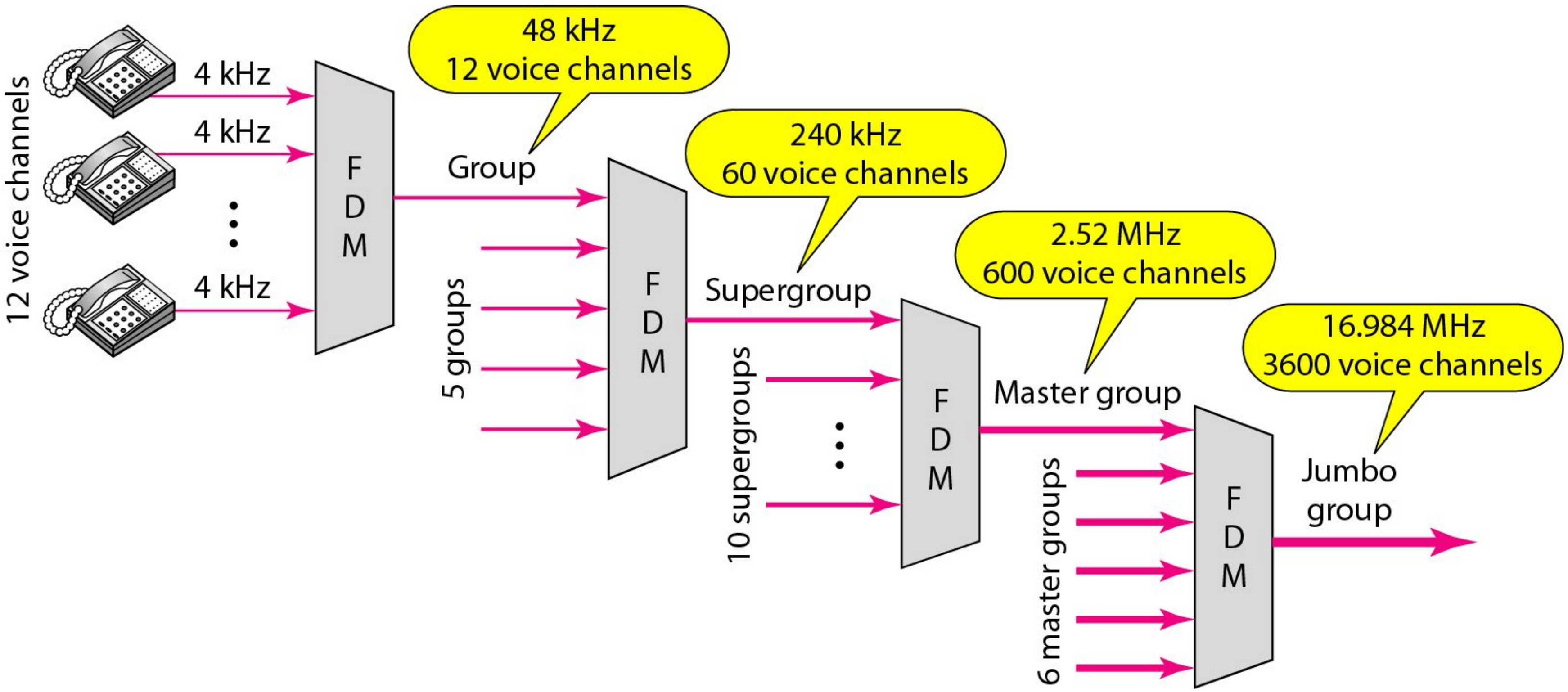


Figure 6.9 *Analog hierarchy*



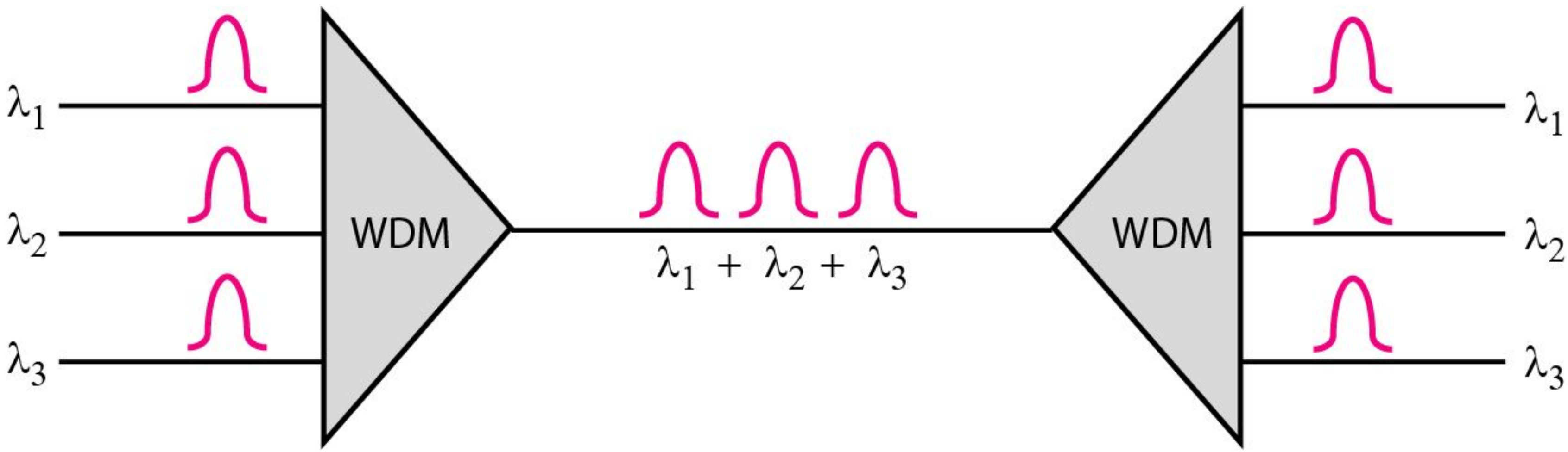
Example 6.4

The Advanced Mobile Phone System (AMPS) uses two bands. The first band of 824 to 849 MHz is used for sending, and 869 to 894 MHz is used for receiving. Each user has a bandwidth of 30 kHz in each direction. How many people can use their cellular phones simultaneously?

Solution

Each band is 25 MHz. If we divide 25 MHz by 30 kHz, we get 833.33. In reality, the band is divided into 832 channels. Of these, 42 channels are used for control, which means only 790 channels are available for cellular phone users.

Figure 6.10 *Wavelength-division multiplexing*



Note

WDM is an analog multiplexing technique to combine optical signals.

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

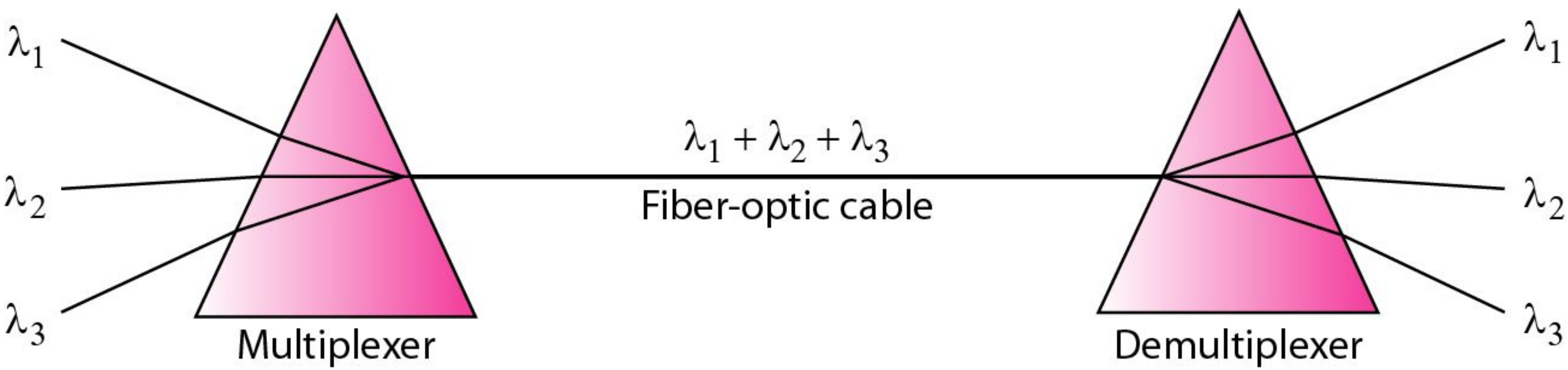
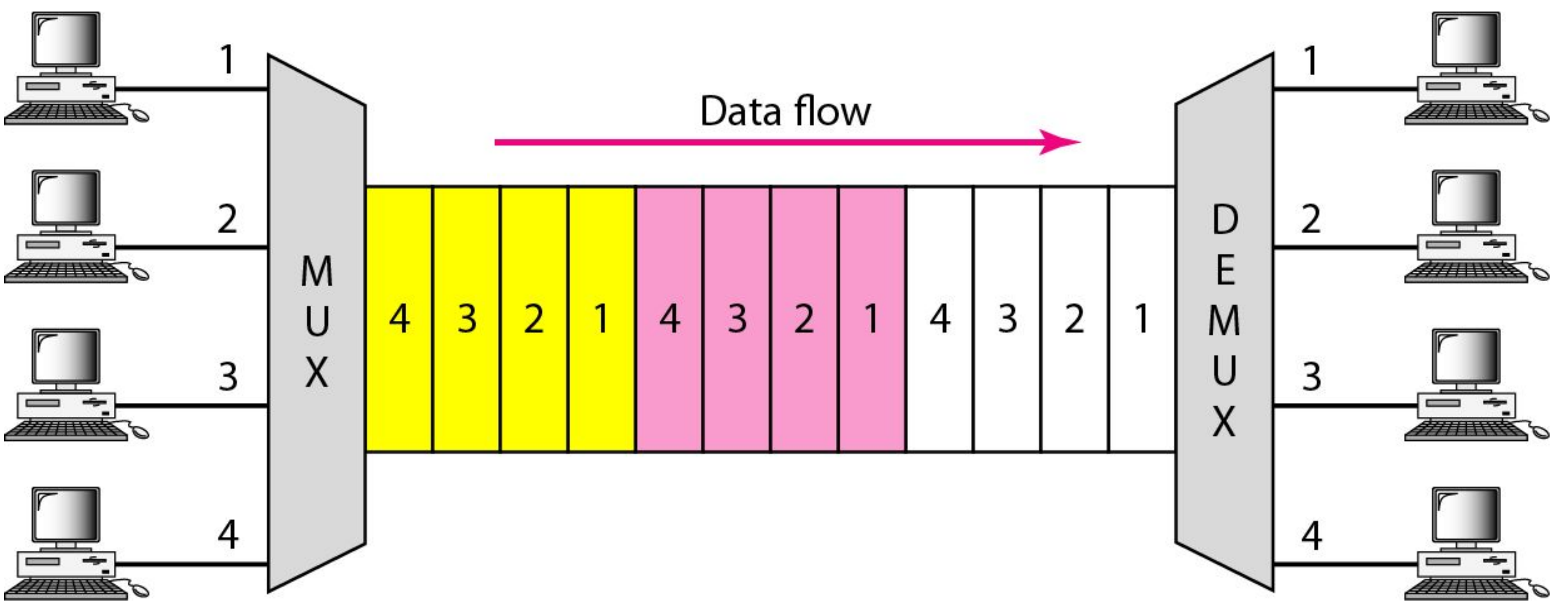


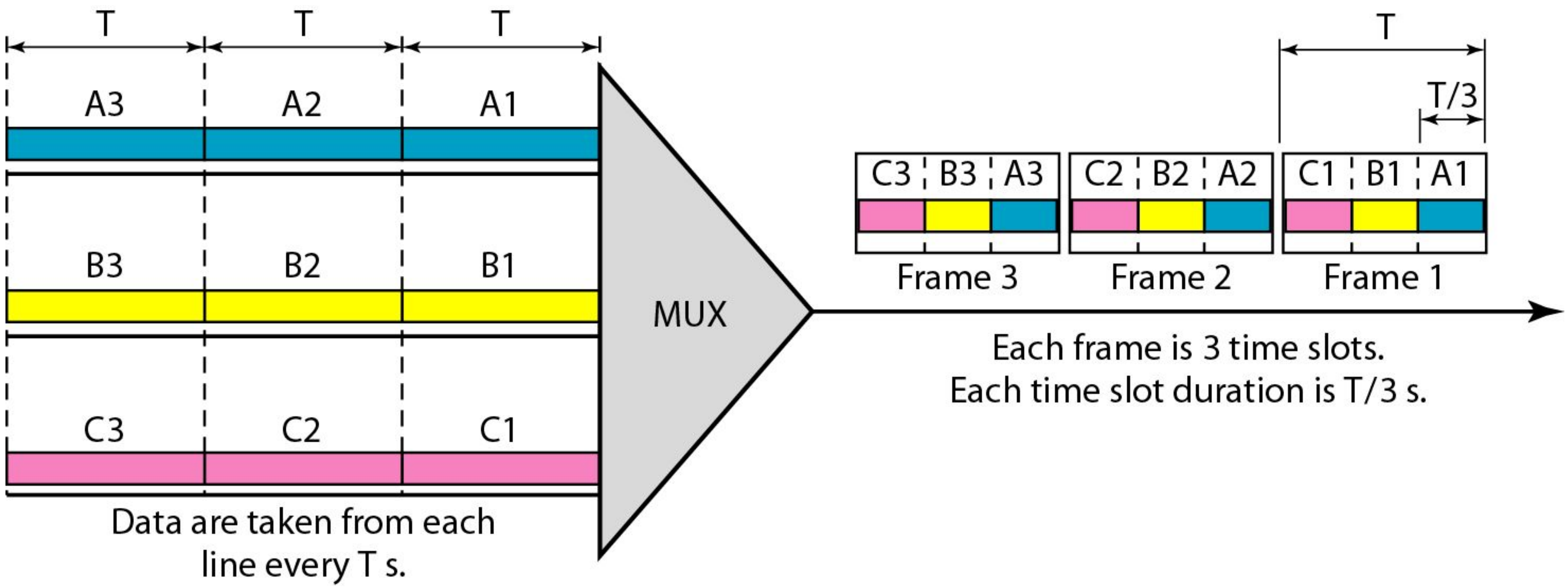
Figure 6.12 *TDM*



Note

TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.

Figure 6.13 *Synchronous time-division multiplexing*



Note

In synchronous TDM, the data rate

of the link is *n* times faster, and the unit duration is *n* times shorter.

Example 6.5

In Figure 6.13, the data rate for each input connection is 3 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of (a) each input slot, (b) each output slot, and (c) each frame?

Solution

We can answer the questions as follows:

a. *The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 s or 1 ms. The duration of the input time slot is 1 ms (same as bit duration).*

b. The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.

c. Each frame carries three output time slots. So the duration of a frame is 3 × 1/3 ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.

Figure 6.14 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (a) the input bit duration, (b) the output bit duration, (c) the output bit rate, and (d) the output frame rate.

Solution

We can answer the questions as follows:

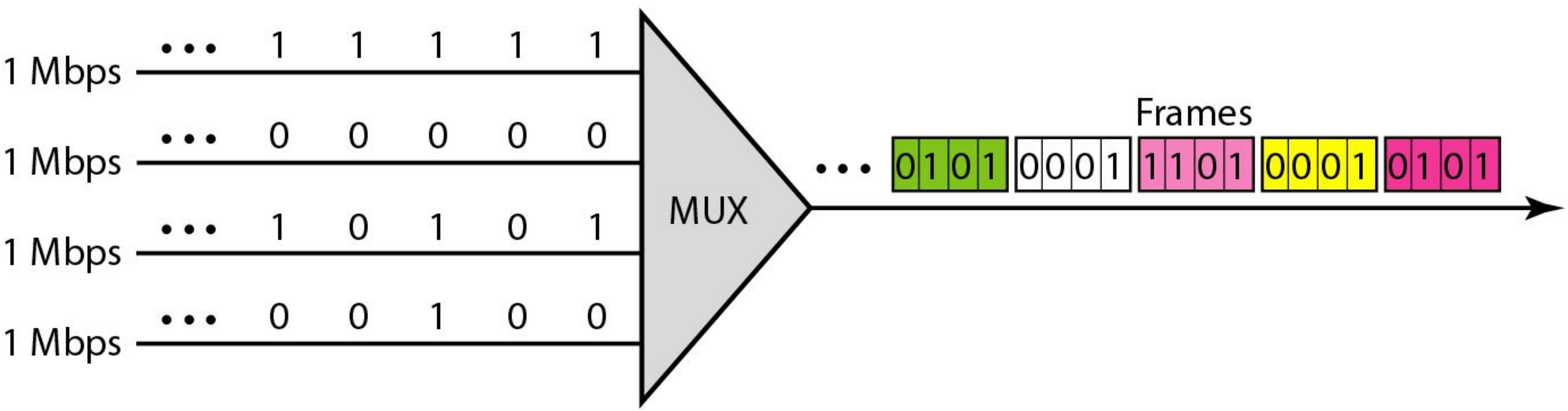
a. The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 μs.

b. The output bit duration is one-fourth of the input bit duration, or ¼ μs.

c. The output bit rate is the inverse of the output bit duration or 1/(4μs) or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate = 4 × 1 Mbps = 4 Mbps.

d. The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second. Because we are sending 4 bits in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.

Figure 6.14 *Example 6.6*



Example 6.7

Four 1-kbps connections are multiplexed together. A unit is 1 bit. Find (a) the duration of 1 bit before multiplexing, (b) the transmission rate of the link, (c) the duration of a time slot, and (d) the duration of a frame.

Solution

We can answer the questions as follows:

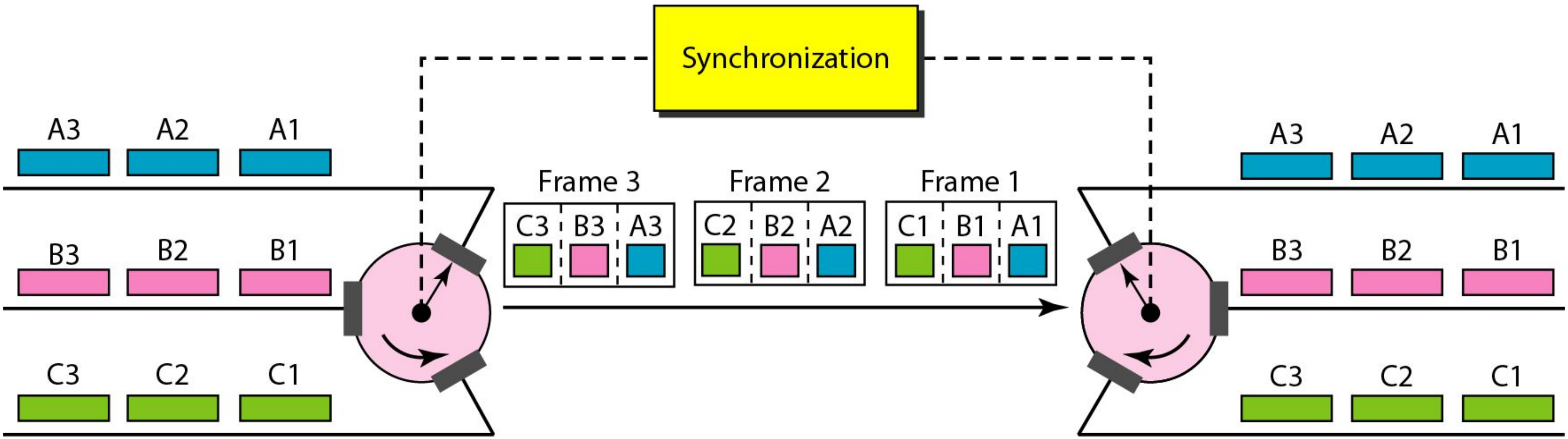
a. The duration of 1 bit before multiplexing is 1 / 1 kbps, or 0.001 s (1 ms).

b. The rate of the link is 4 times the rate of a connection, or 4 kbps.

c. The duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 1/4 ms or 250 μs. Note that we can also calculate this from the data rate of the link, 4 kbps. The bit duration is the inverse of the data rate, or 1/4 kbps or 250 μs.

d. The duration of a frame is always the same as the duration of a unit before multiplexing, or 1 ms. We can also calculate this in another way. Each frame in this case has four time slots. So the duration of a frame is 4 times 250 μs, or 1 ms.

Figure 6.15 *Interleaving*



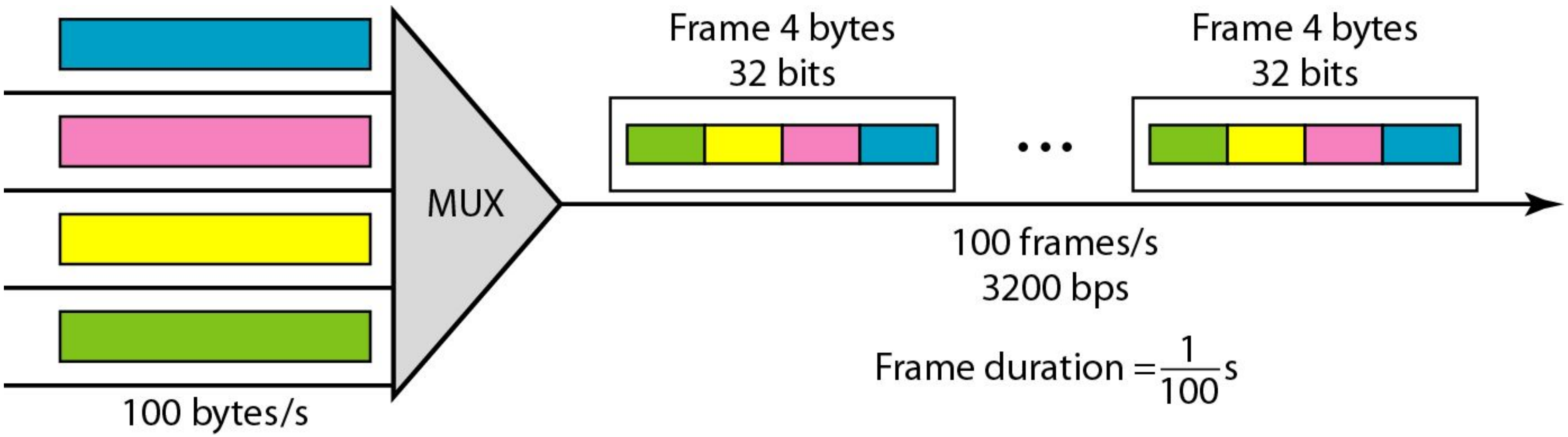
Example 6.8

Four channels are multiplexed using TDM. If each channel sends 100 bytes /s and we multiplex 1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Solution

The multiplexer is shown in Figure 6.16. Each frame carries 1 byte from each channel; the size of each frame, therefore, is 4 bytes, or 32 bits. Because each channel is sending 100 bytes/s and a frame carries 1 byte from each channel, the frame rate must be 100 frames per second. The bit rate is 100 × 32, or 3200 bps.

Figure 6.16 *Example 6.8*



A multiplexer combines four 100-kbps channels using a time slot of 2 bits. Show the output with four arbitrary inputs. What is the frame rate? What is the frame duration? What is the bit rate? What is the bit duration?

Solution

Figure 6.17 shows the output for four arbitrary inputs. The link carries 50,000 frames per second. The frame duration is therefore 1/50,000 s or 20 μs. The frame rate is 50,000 frames per second, and each frame carries 8 bits; the bit rate is 50,000 × 8 = 400,000 bits or 400 kbps. The bit duration is 1/400,000 s, or 2.5 μs.

Figure 6.17 *Example 6.9*

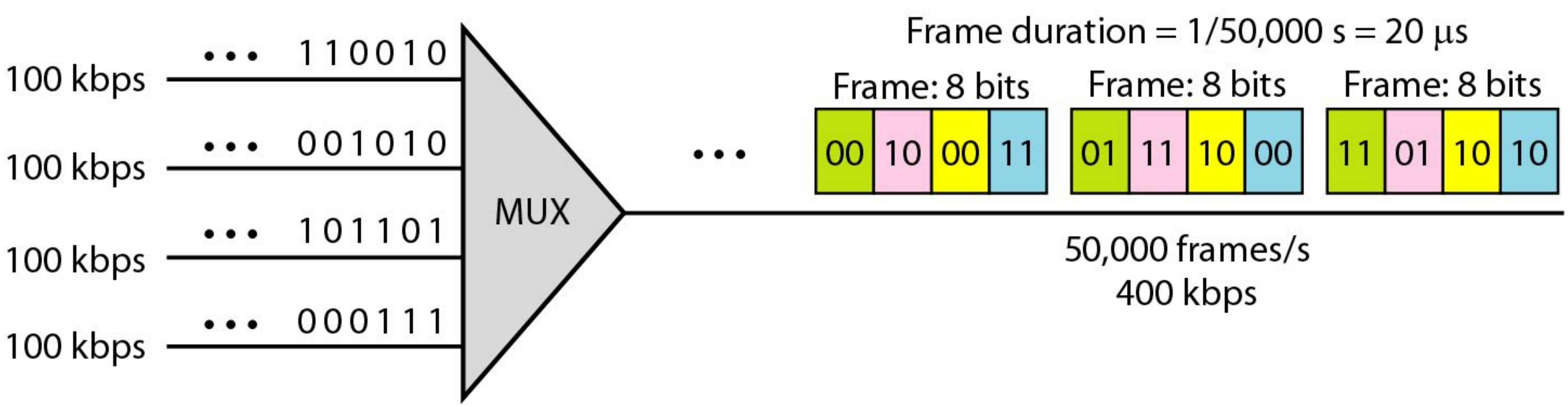


Figure 6.18 *Empty slots*

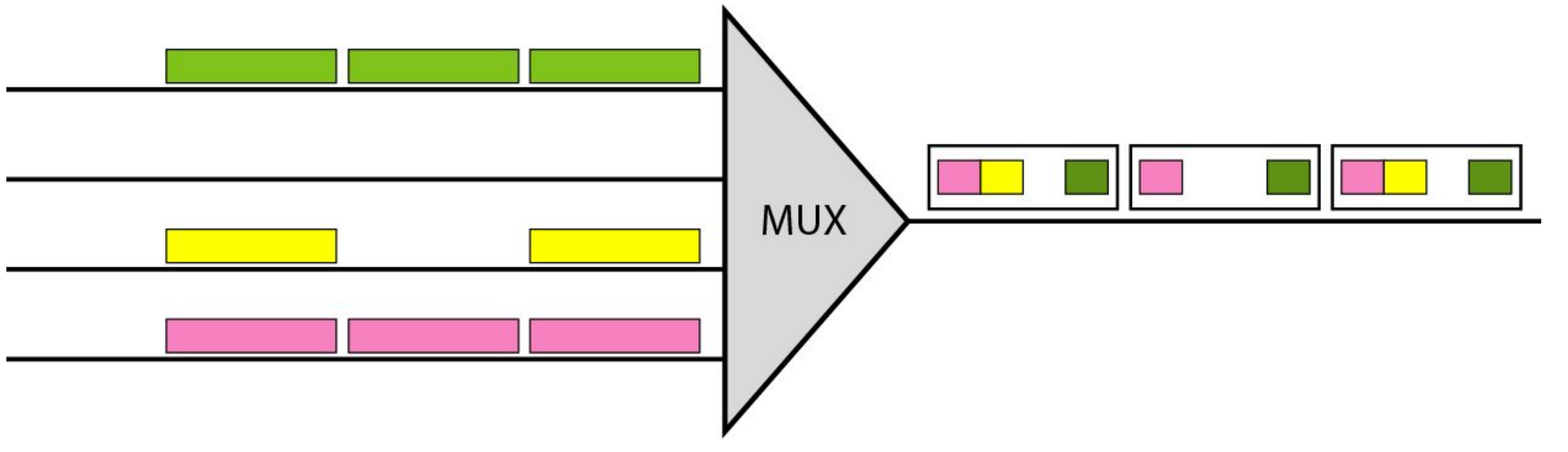


Figure 6.19 *Multilevel multiplexing*

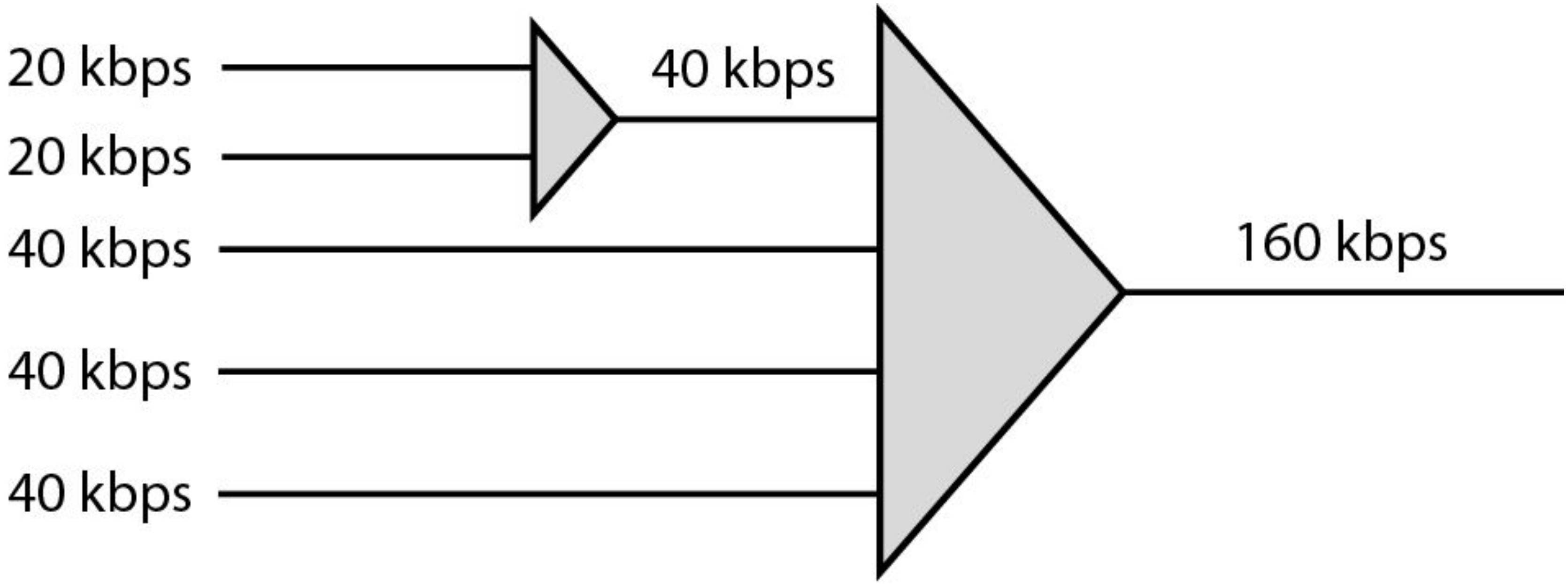


Figure 6.20 *Multiple-slot multiplexing*

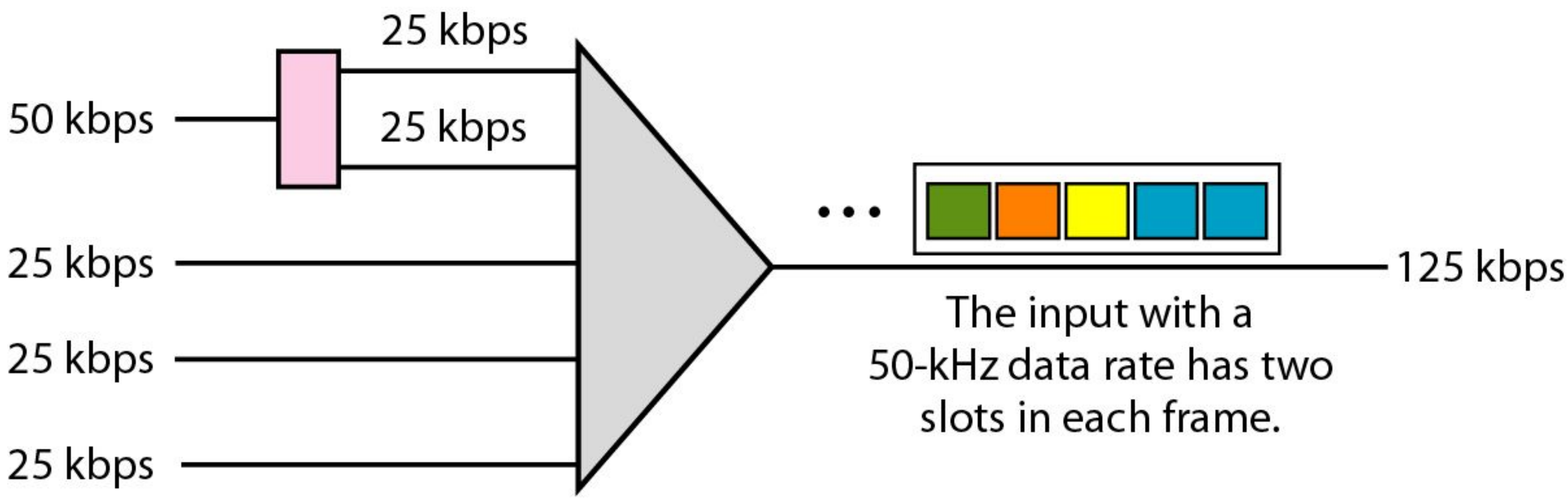


Figure 6.21 *Pulse stuffing*

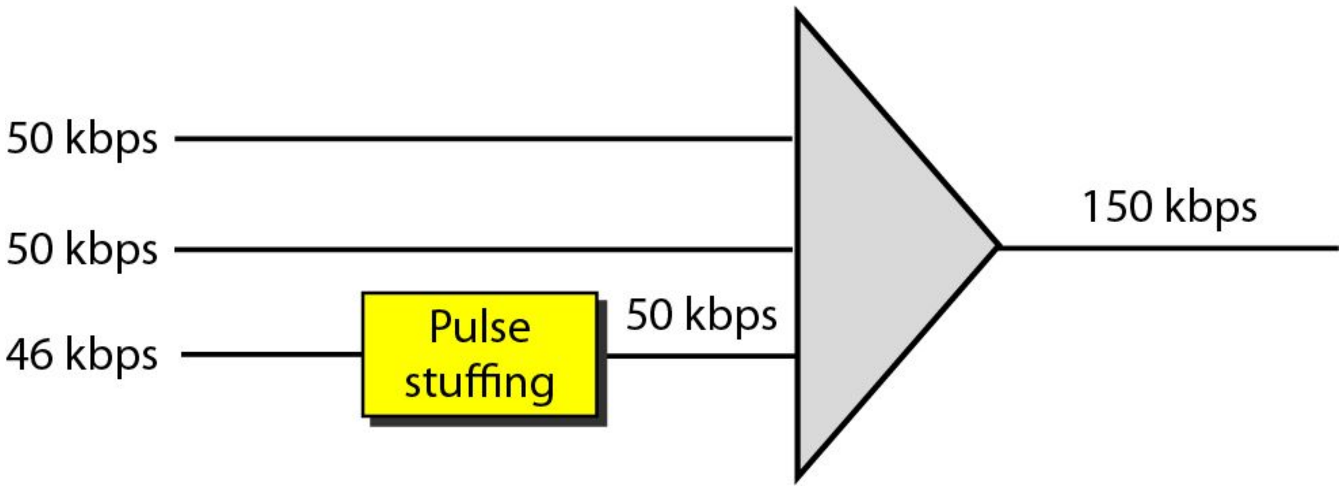
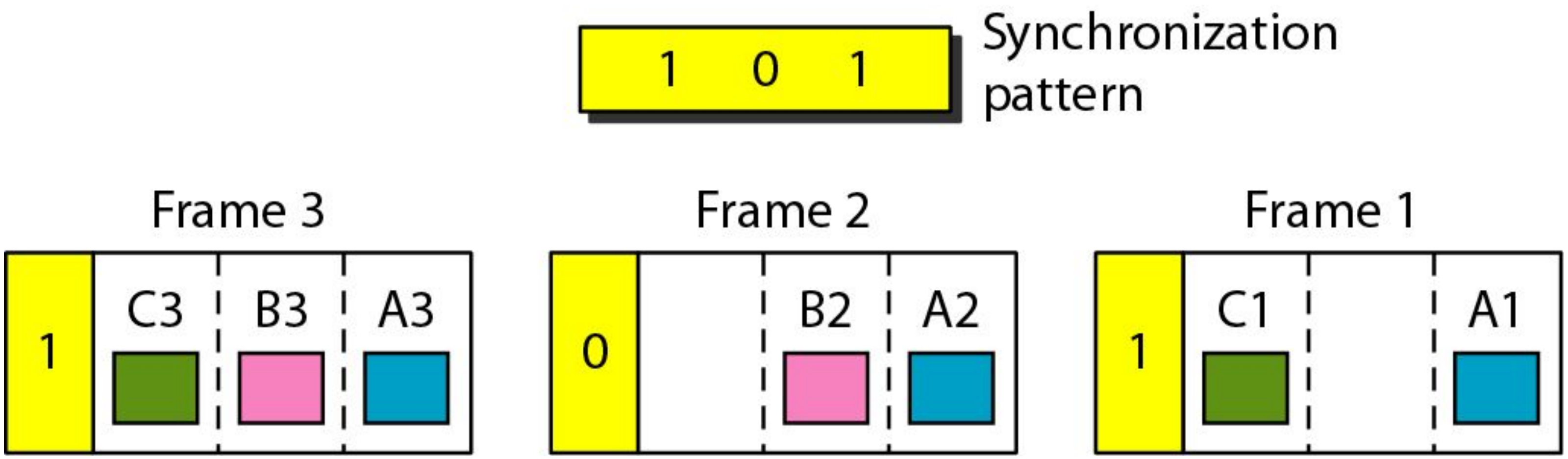


Figure 6.22 *Framing bits*



Example 6.10

We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (a) the data rate of each source, (b) the duration of each character in each source, (c) the frame rate, (d) the duration of each frame, (e) the number of bits in each frame, and (f) the data rate of the link.

Solution

We can answer the questions as follows:

a. The data rate of each source is $250 \times 8 = 2000$ *bps = 2 kbps.*

b. *Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.*

c. *Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.*

d. *The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.*

e. *Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is*

$4 \times 8 + 1 = 33$ *bits.*

Example 6.11

Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How this can be achieved? What is the frame rate? What is the frame duration? What is the bit rate of the link?

Solution

We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The bit rate is 100,000 frames/s × 3 bits per frame, or 300 kbps.

Figure 6.23 *Digital hierarchy*

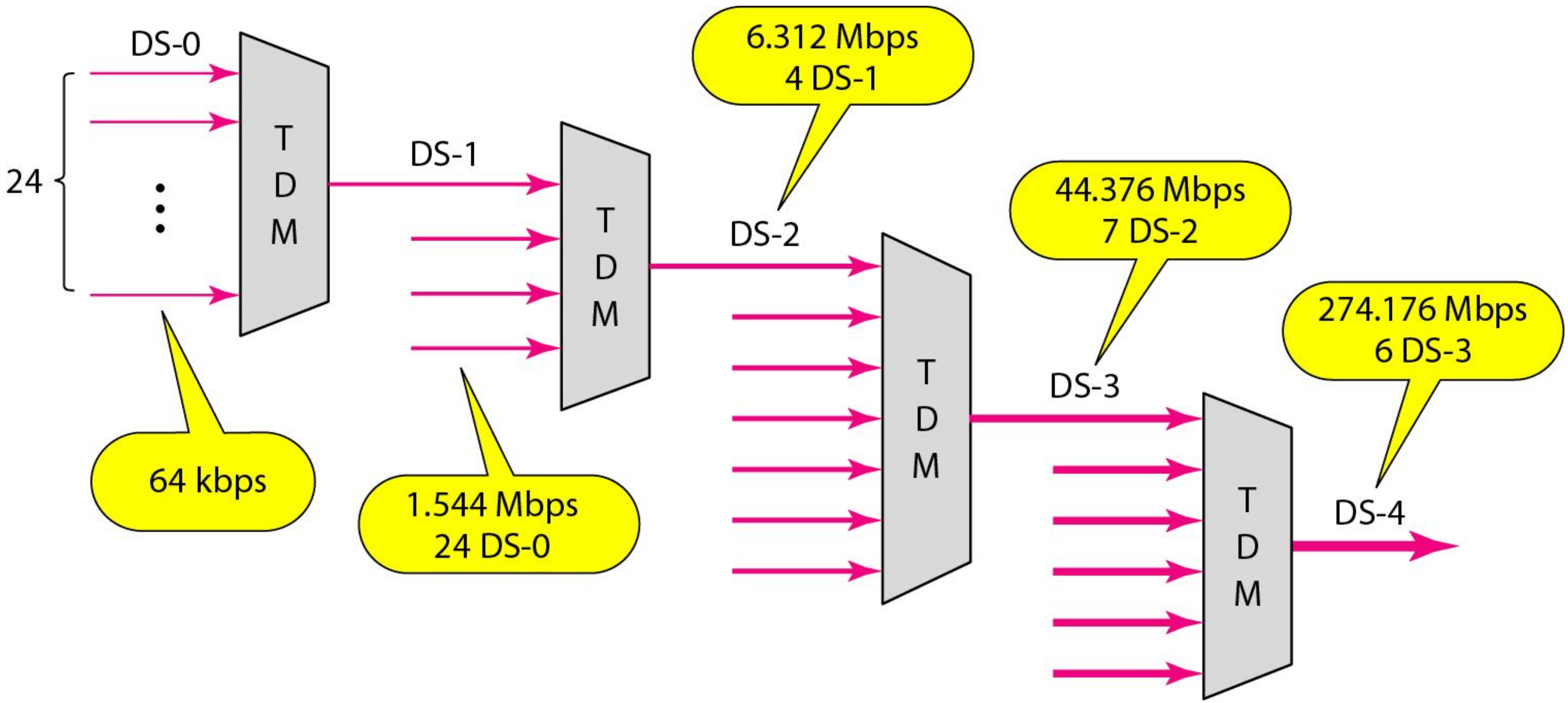


Table 6.1 *DS and T line rates*

Service	*Line*	*Rate (Mbps)*	*Voice Channels*
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

Figure 6.24 *T-1 line for multiplexing telephone lines*

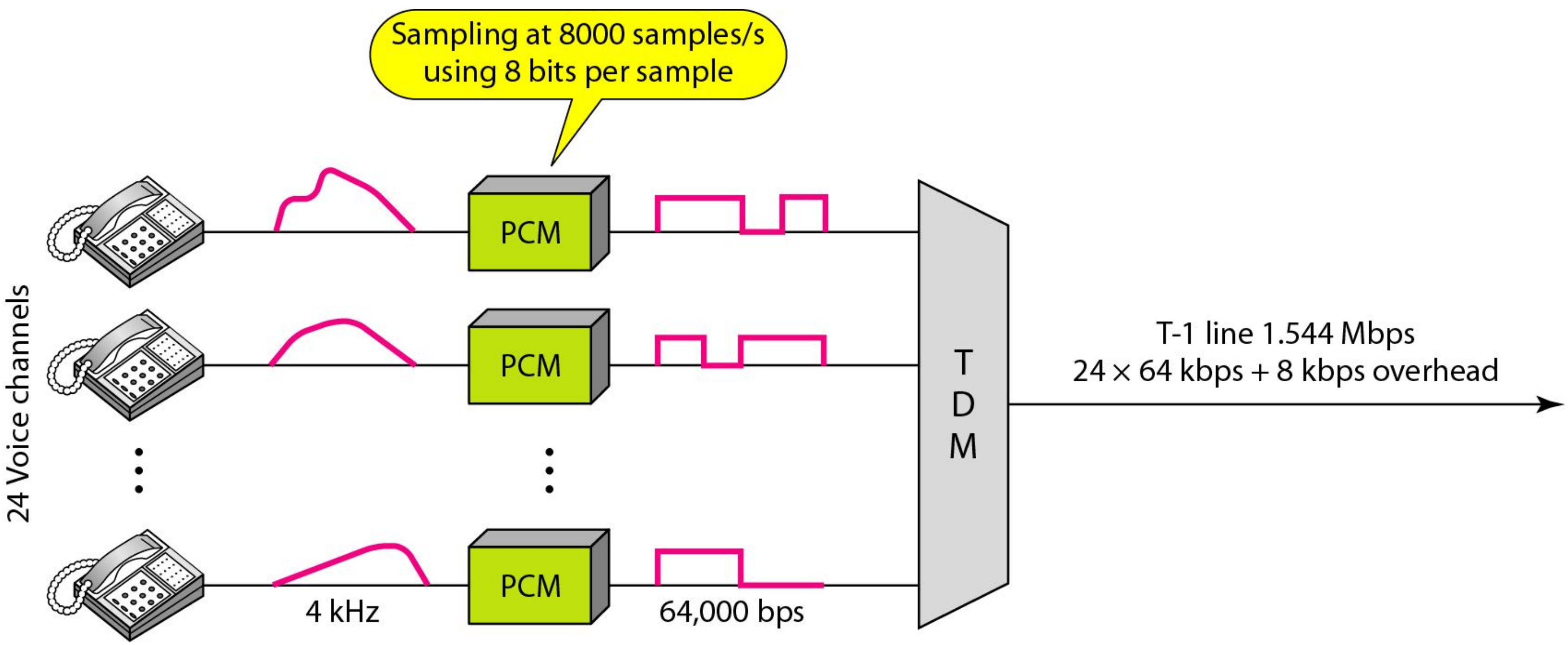


Figure 6.25 *T-1 frame structure*

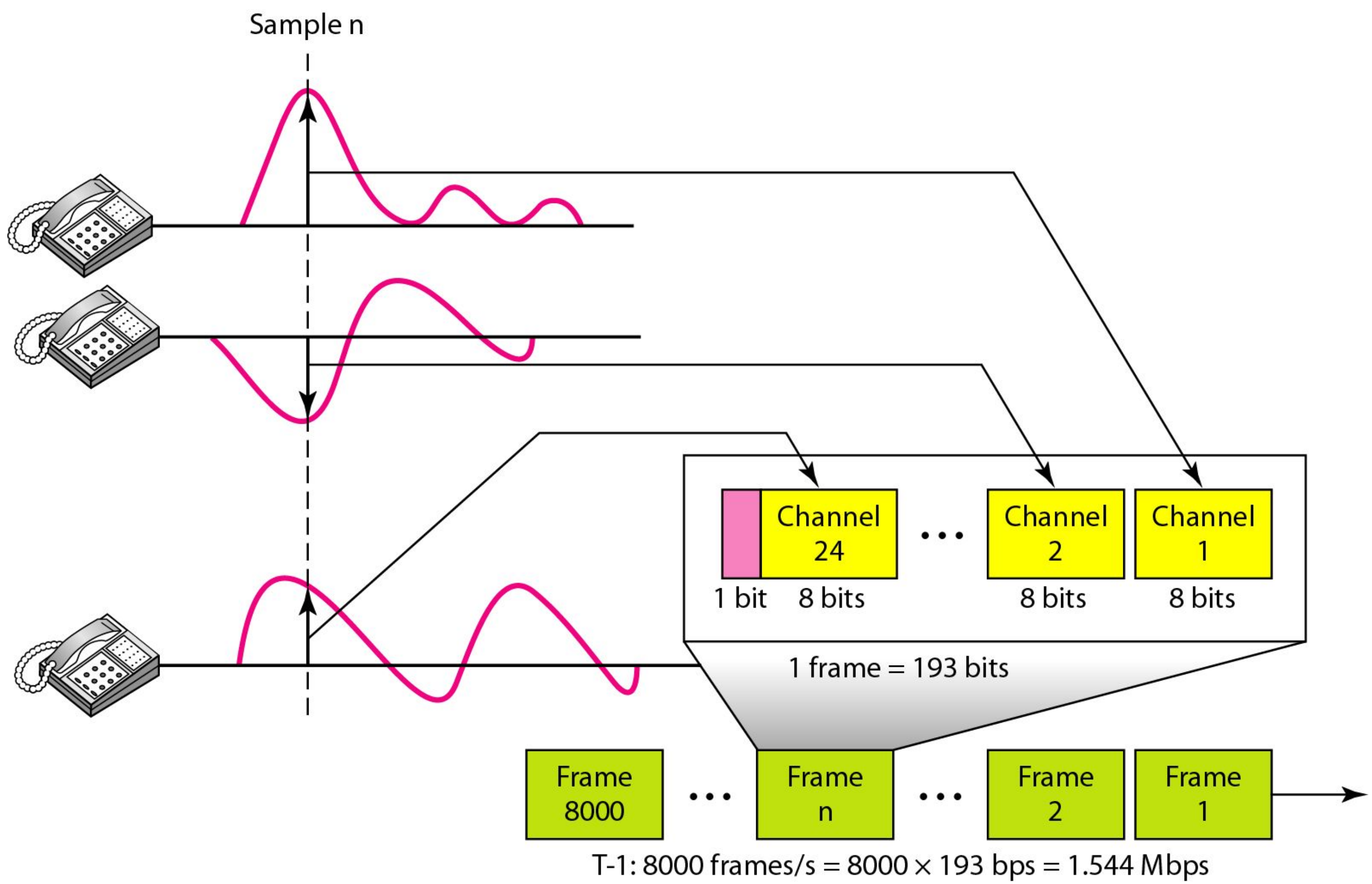
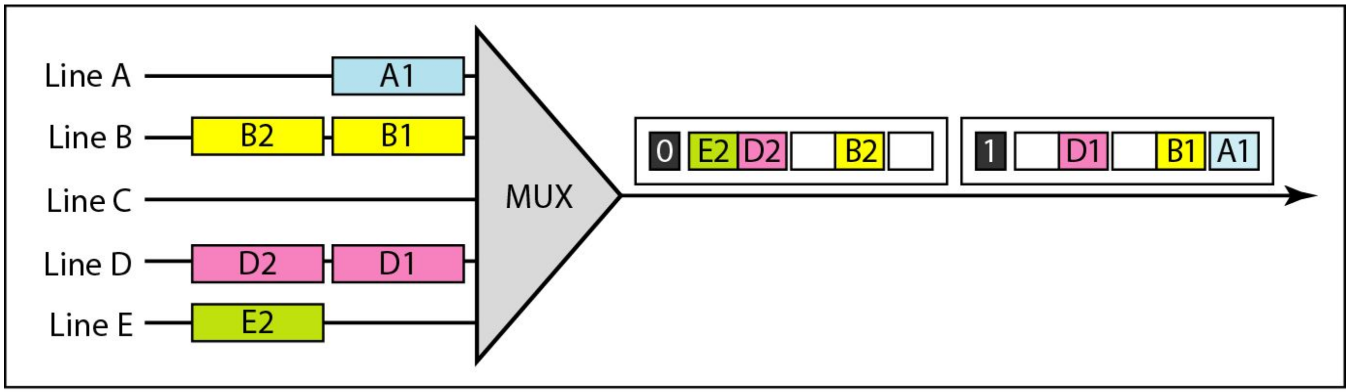


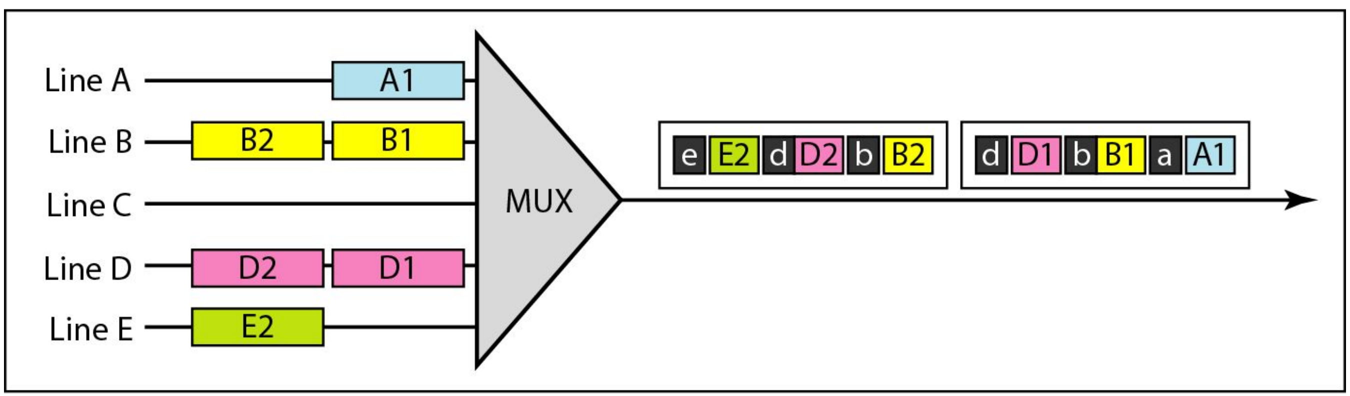
Table 6.2 *E line rates*

Line	*Rate (Mbps)*	*Voice Channels*
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1920

Figure 6.26 *TDM slot comparison*



a. Synchronous TDM



b. Statistical TDM

6-1 SPREAD SPECTRUM

In spread spectrum (SS), we combine signals from different sources to fit into a larger bandwidth, but our goals are to prevent eavesdropping and jamming. To achieve these goals, spread spectrum techniques add redundancy.

~~*Topics discussed in this section:*~~

Frequency Hopping Spread Spectrum (FHSS)

Direct Sequence Spread Spectrum Synchronous (DSSS)

Figure 6.27 *Spread spectrum*

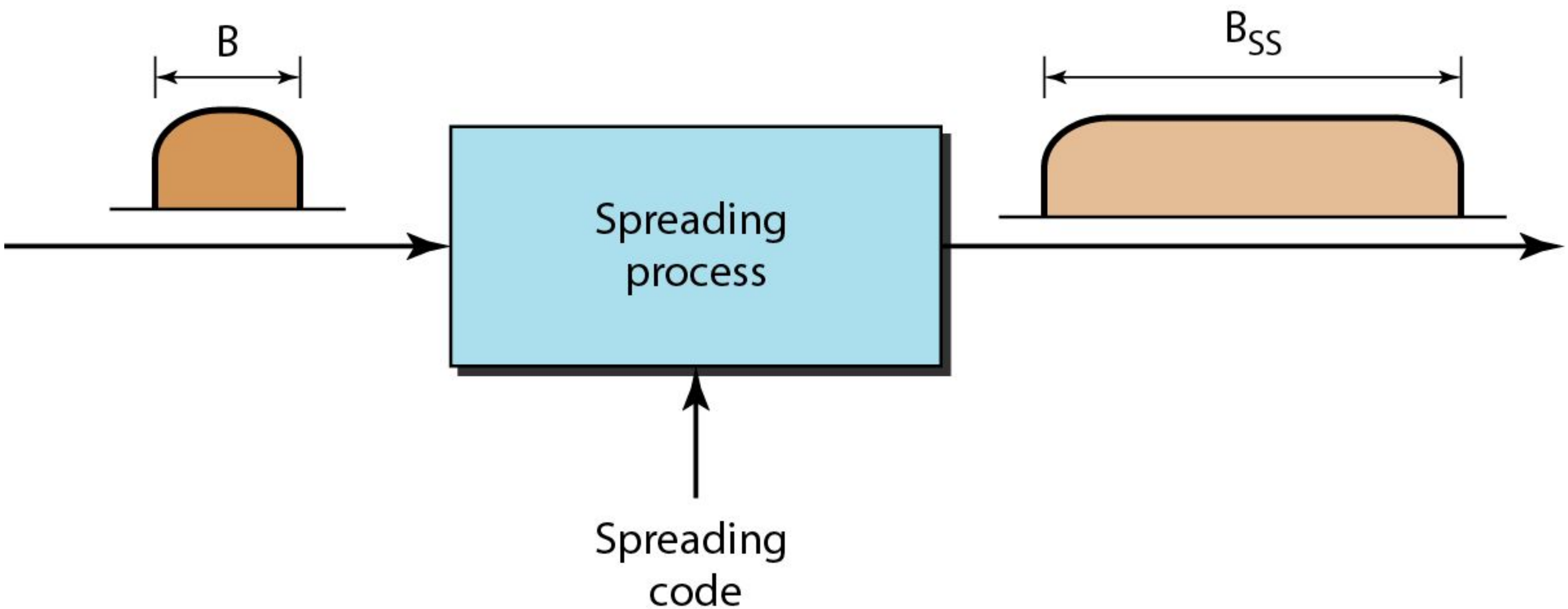


Figure 6.28 *Frequency hopping spread spectrum (FHSS)*

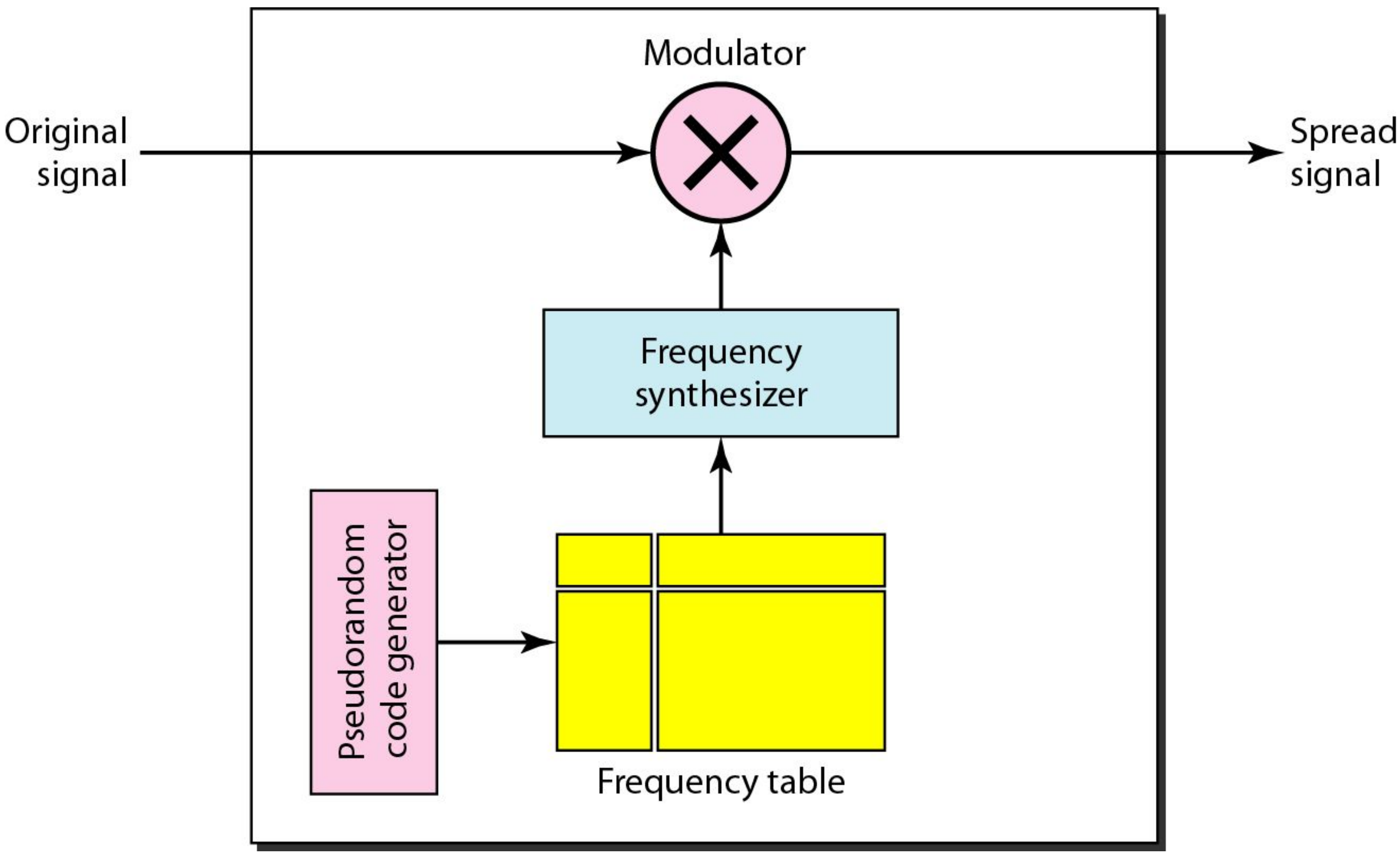


Figure 6.29 *Frequency selection in FHSS*

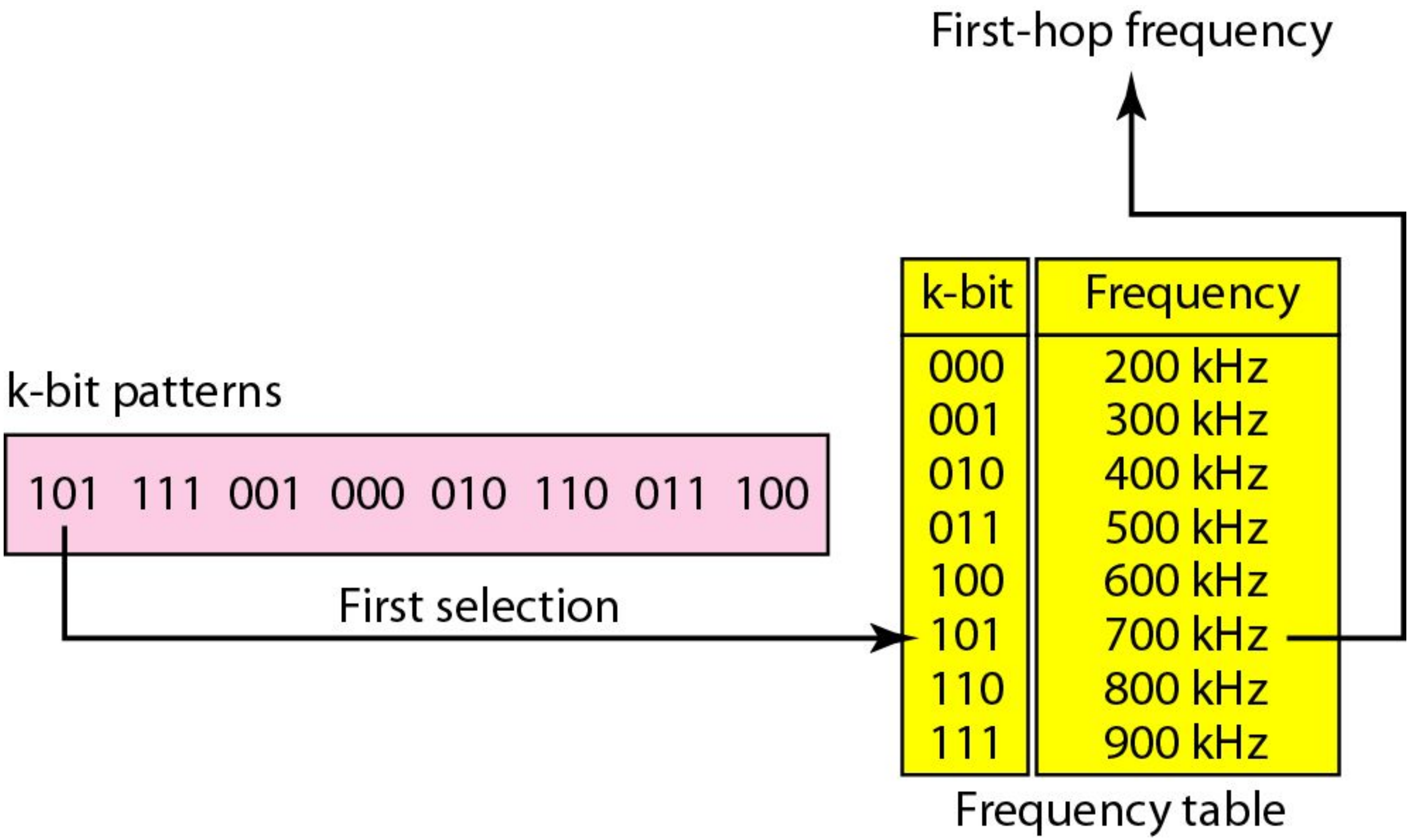


Figure 6.30 *FHSS cycles*

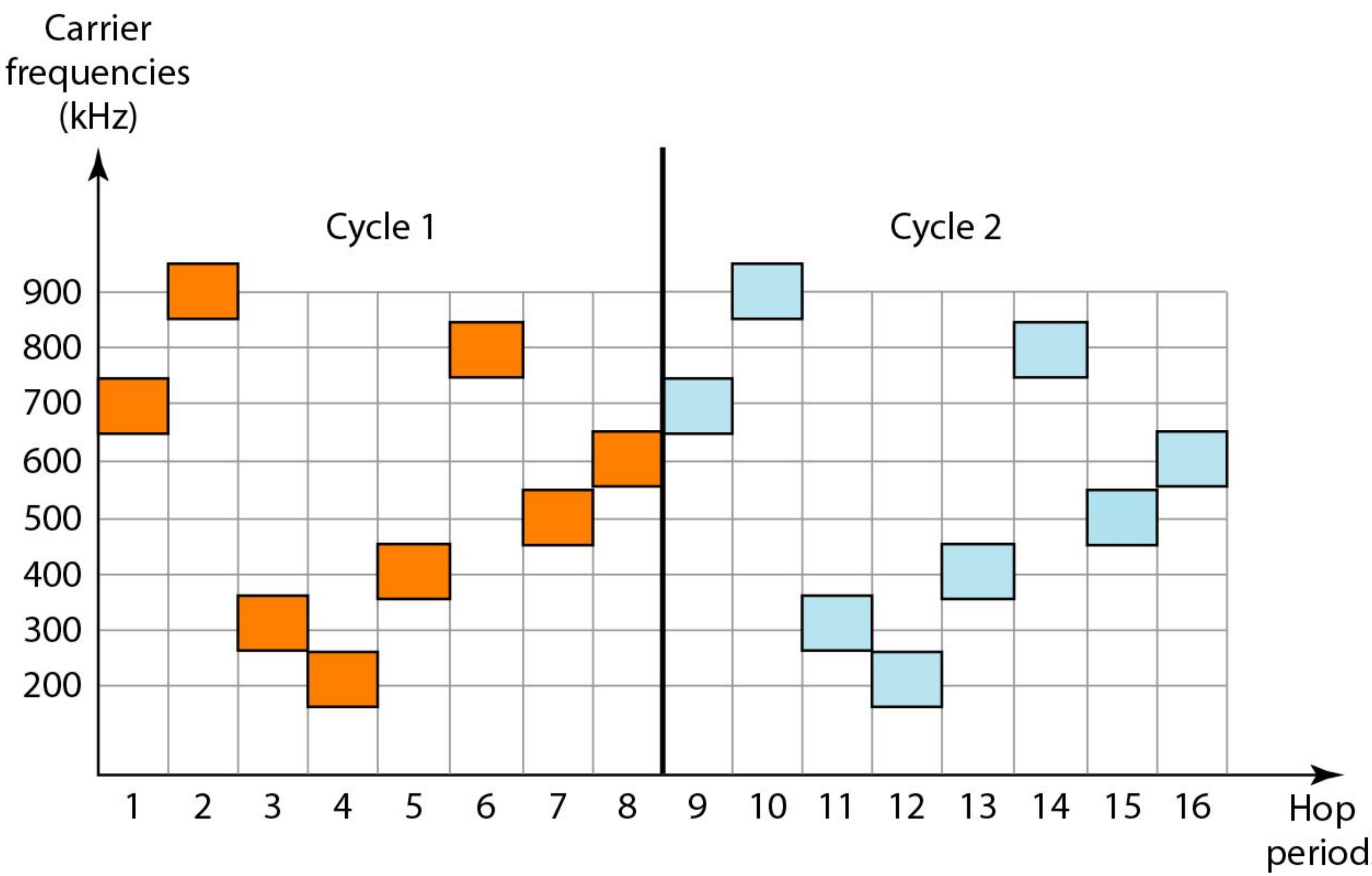


Figure 6.31 *Bandwidth sharing*

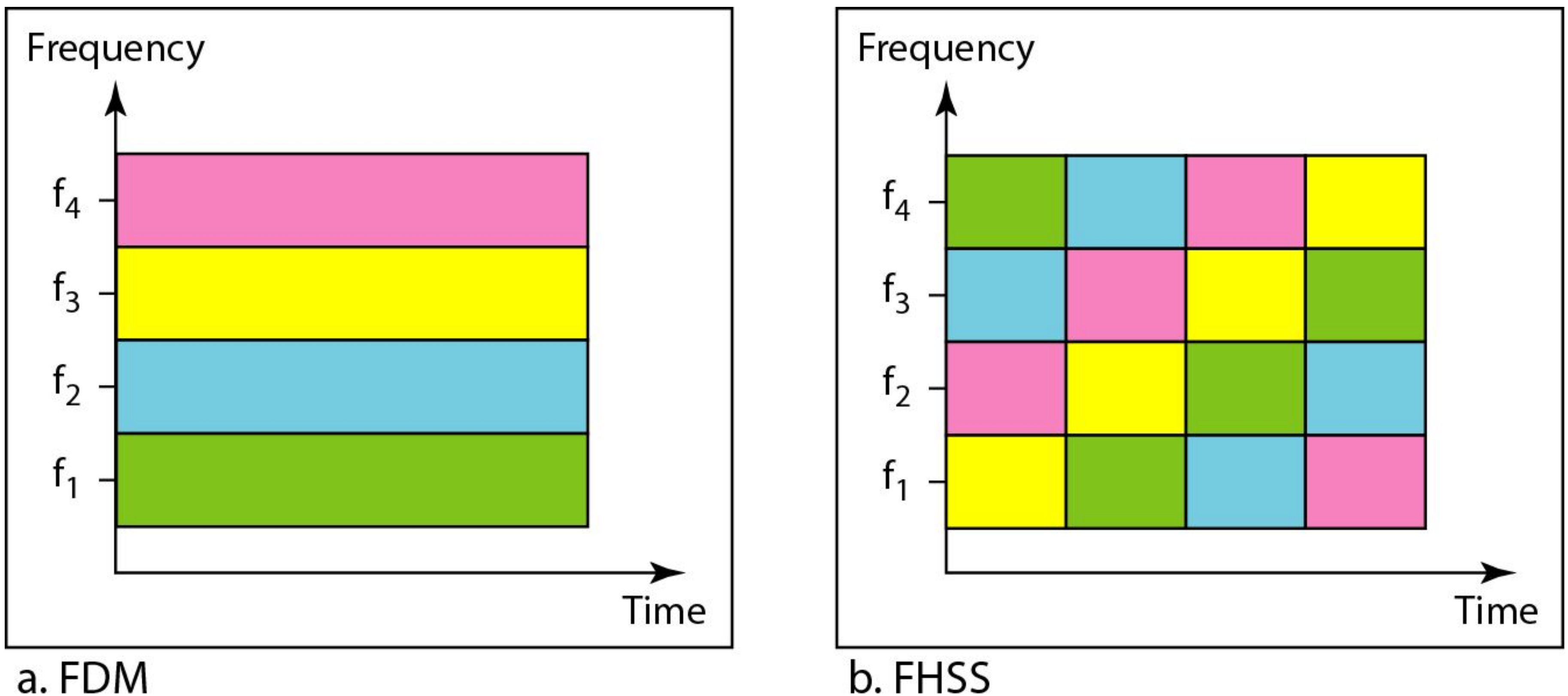


Figure 6.32 *DSSS*

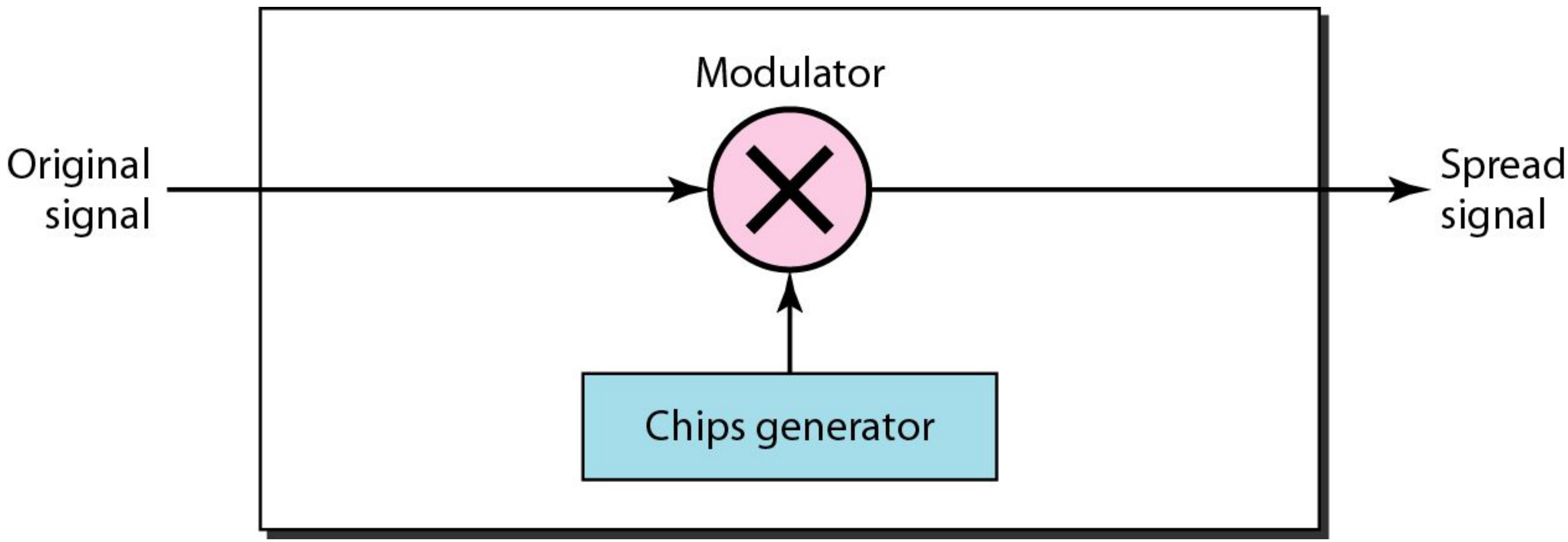
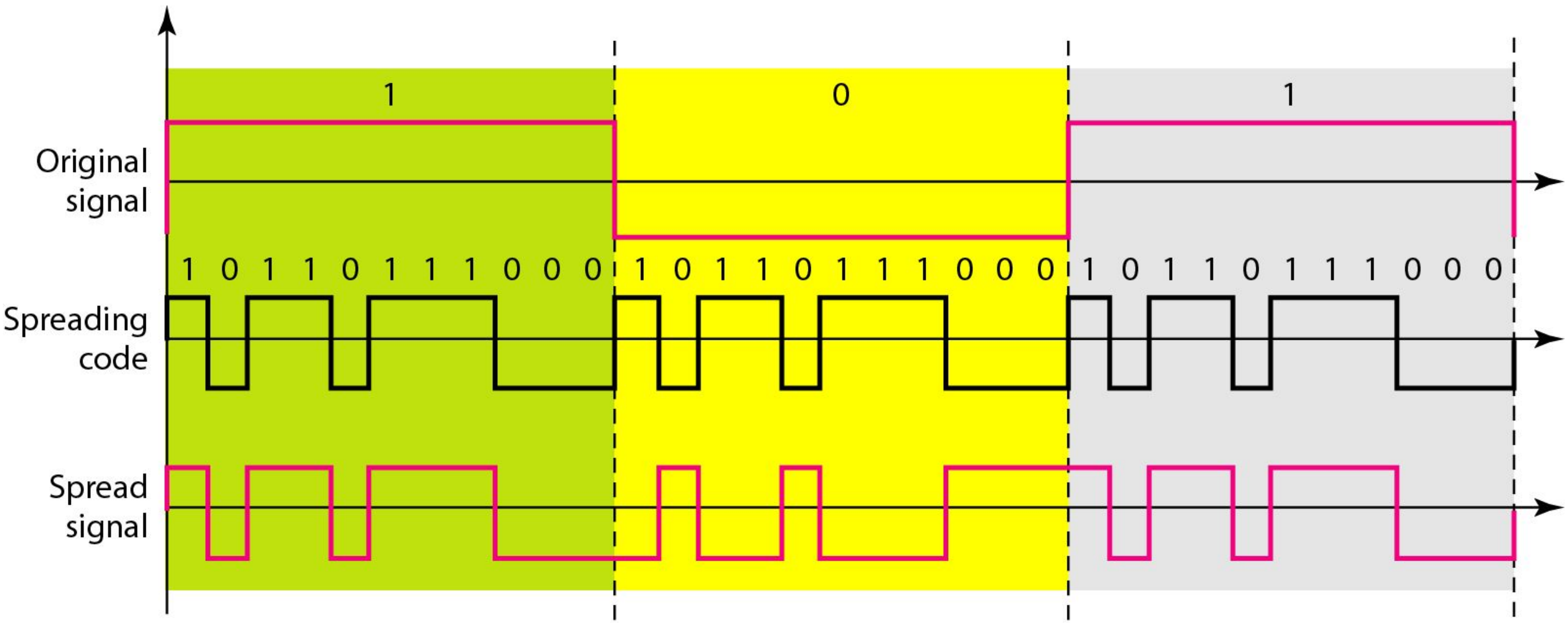


Figure 6.33 *DSSS example*



Data compression implies sending or storing a smaller number of bits. Although many methods are used for this purpose, in general these methods can be divided into two broad categories: lossless and lossy methods.

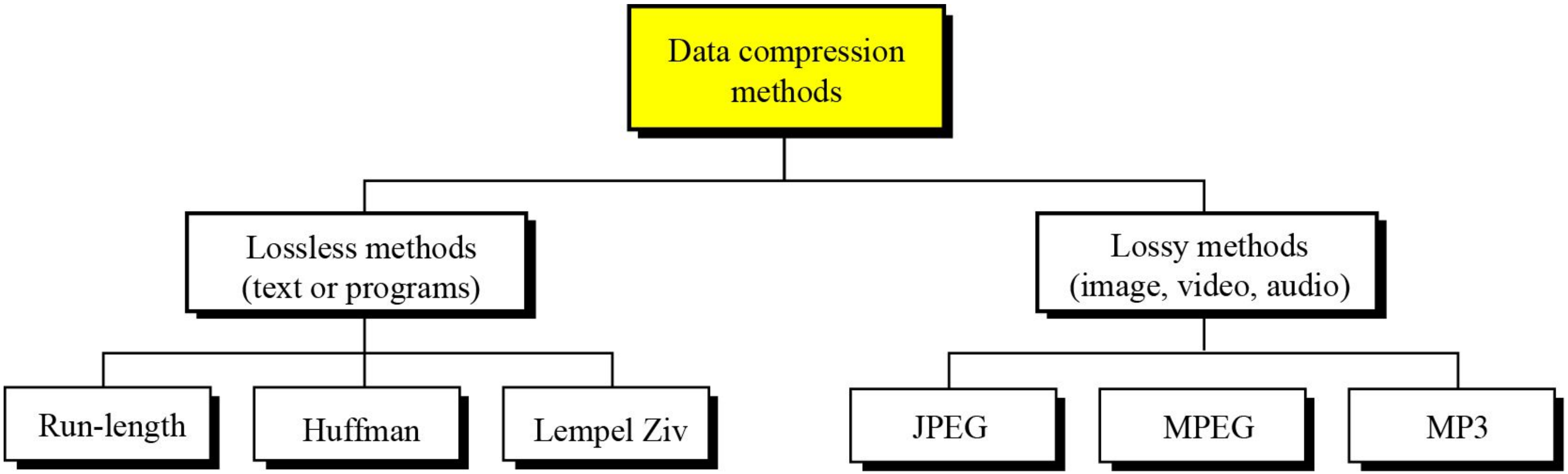


Figure 15.1 Data compression methods

15-1 LOSSLESS COMPRESSION

In lossless data compression, the integrity of the data is preserved. The original data and the data after compression and decompression are exactly the same because, in these methods, the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process. Redundant data is removed in compression and added during decompression. Lossless compression methods are normally used when we cannot afford to lose any data.

Run-length encoding is probably the simplest method of compression. It can be used to compress data made of any combination of symbols. It does not need to know the frequency of occurrence of symbols and can be very efficient if data is represented as 0s and 1s.

The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.

The method can be even more efficient if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other.

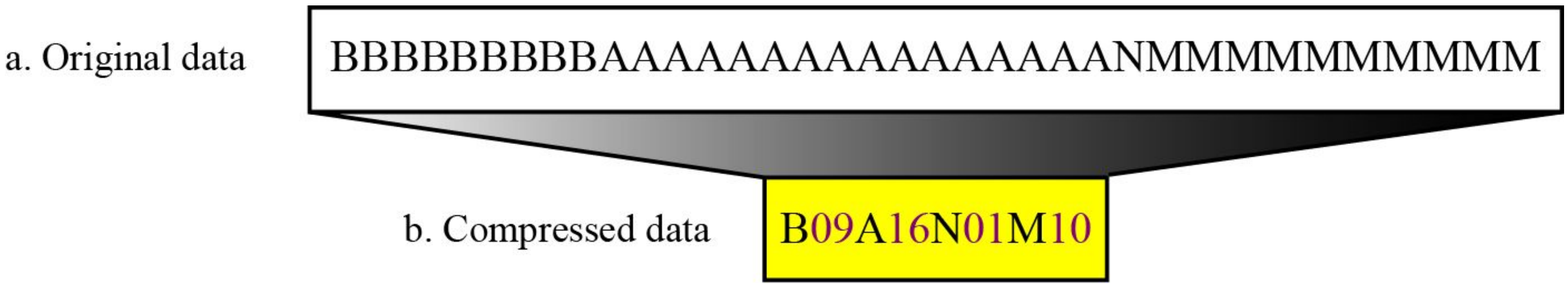


Figure 15.2 Run-length encoding example

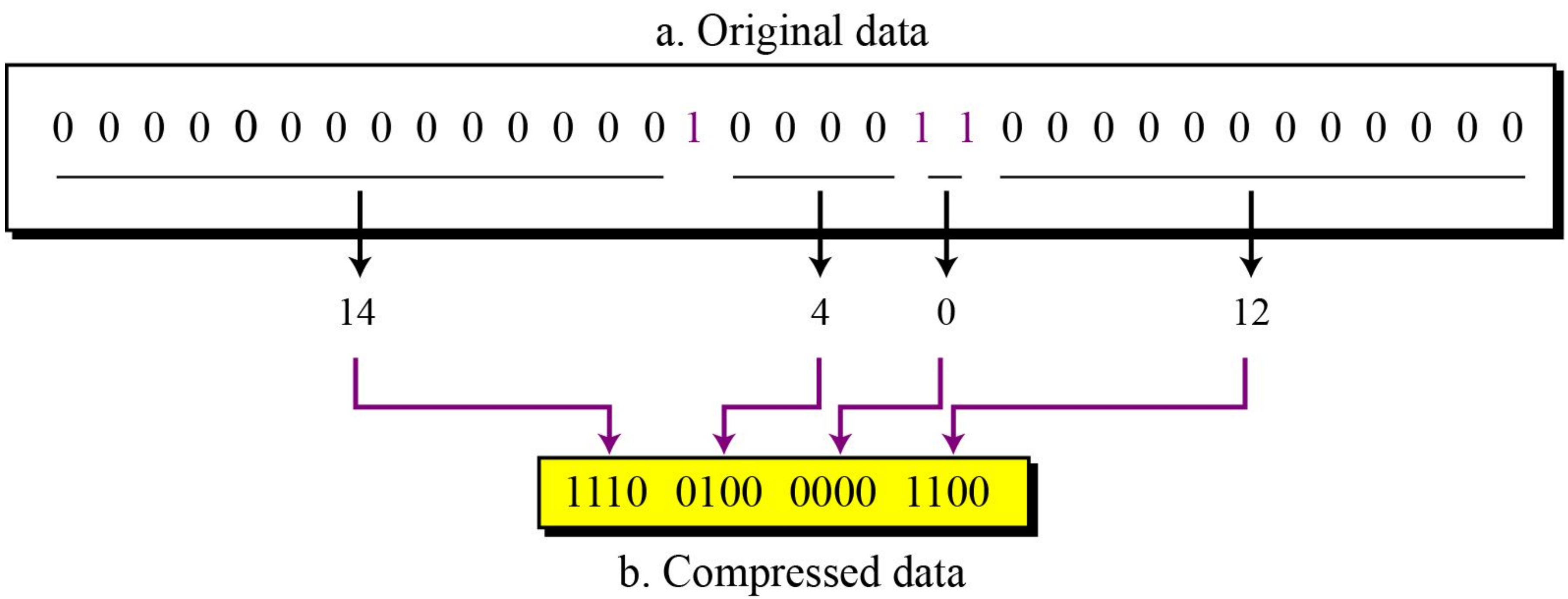


Figure 15.3 Run-length encoding for two symbols

Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently. For example, imagine we have a text file that uses only five characters (A, B, C, D, E). Before we can assign bit patterns to each character, we assign each character a weight based on its frequency of use. In this example, assume that the frequency of the characters is as shown in Table 15.1.

Table 15.1 Frequency of characters

Character	A	B	C	D	E
Frequency	17	12	12	27	32

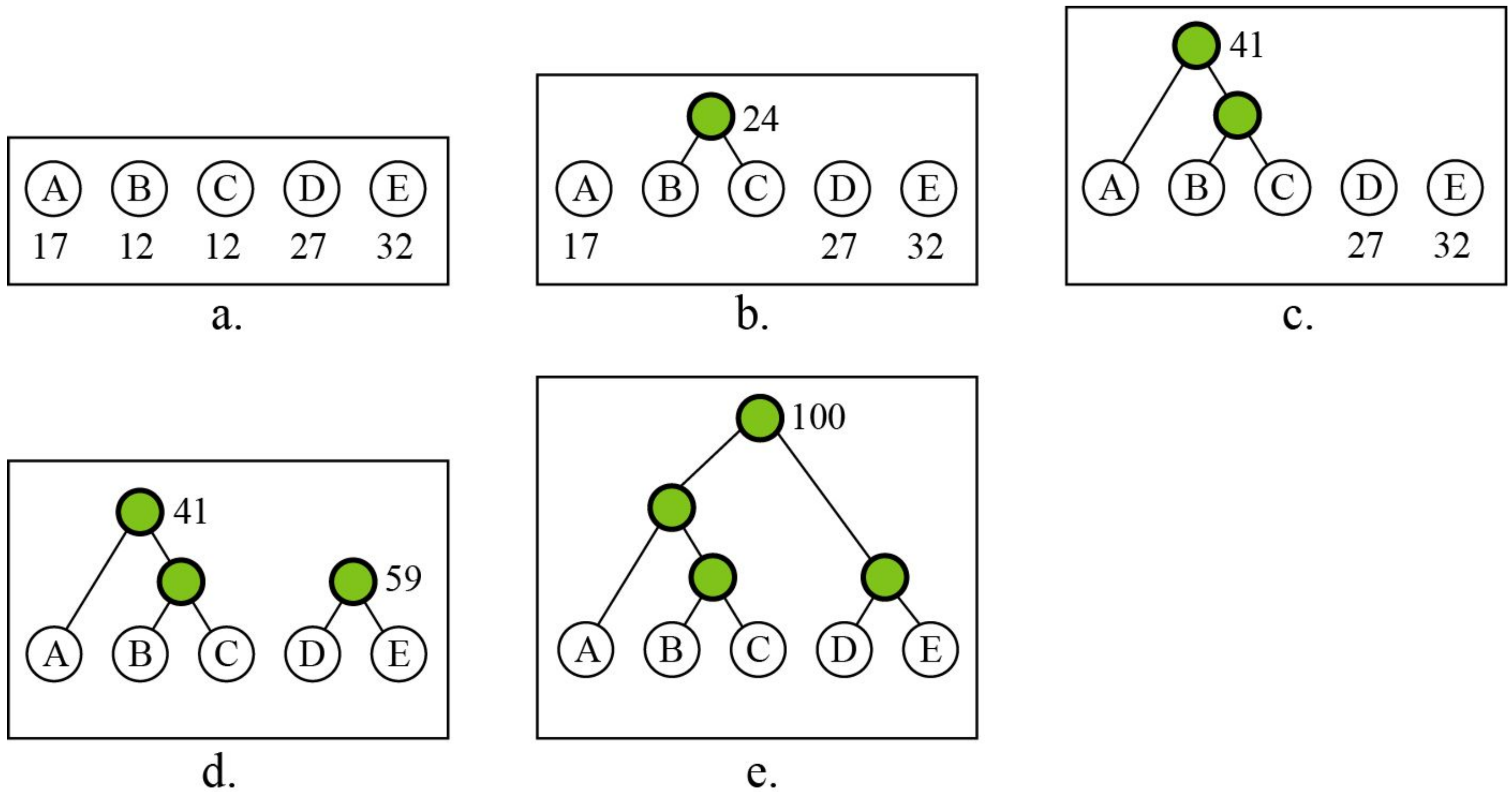
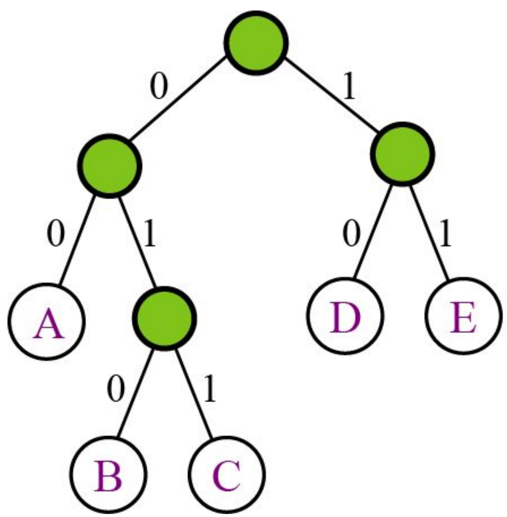


Figure 15.4 Huffman coding

A character's code is found by starting at the root and following the branches that lead to that character. The code itself is the bit value of each branch on the path, taken in sequence.



A: 00	D: 10
B: 010	E: 11
C: 011	

Code

Figure 15.5 Final tree and code

Encoding

Let us see how to encode text using the code for our five characters. Figure 15.6 shows the original and the encoded text.

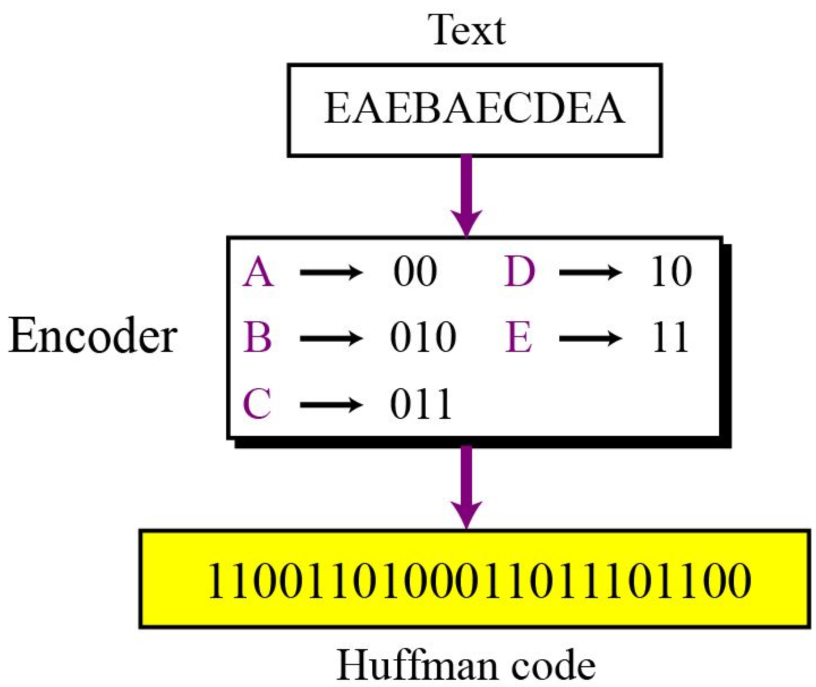


Figure 15.6 Huffman encoding

Decoding

The recipient has a very easy job in decoding the data it receives. Figure 15.7 shows how decoding takes place.

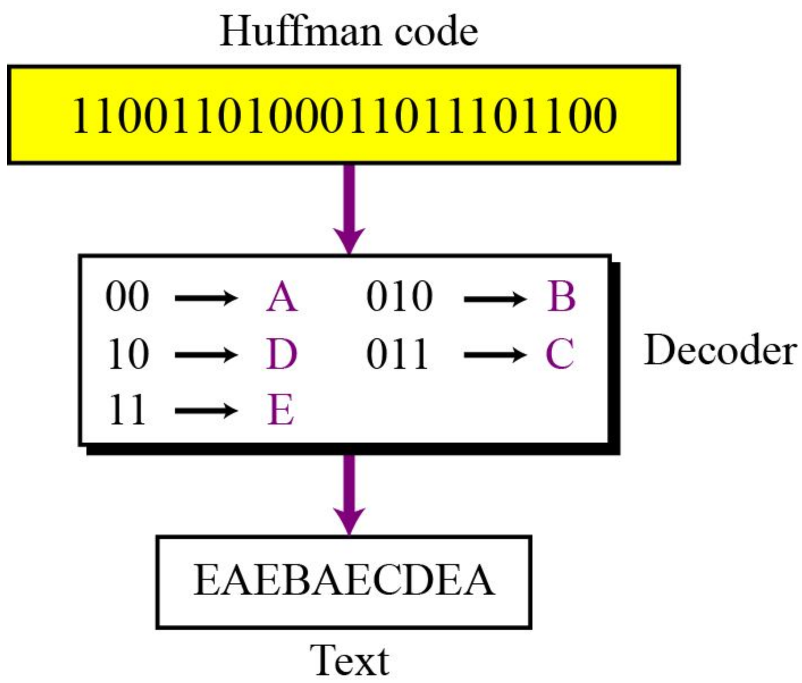


Figure 15.7 Huffman decoding

Lempel Ziv (LZ) encoding is an example of a category of algorithms called *dictionary-based* encoding. The idea is to create a dictionary (a table) of strings used during the communication session. If both the sender and the receiver have a copy of the dictionary, then previously-encountered strings can be substituted by their index in the dictionary to reduce the amount of information transmitted.

Compression

In this phase there are two concurrent events: building an indexed dictionary and compressing a string of symbols. The algorithm extracts the smallest substring that cannot be found in the dictionary from the remaining uncompressed string. It then stores a copy of this substring in the dictionary as a new entry and assigns it an index value. Compression occurs when the substring, except for the last character, is replaced with the index found in the dictionary. The process then inserts the index and the last character of the substring into the compressed string.

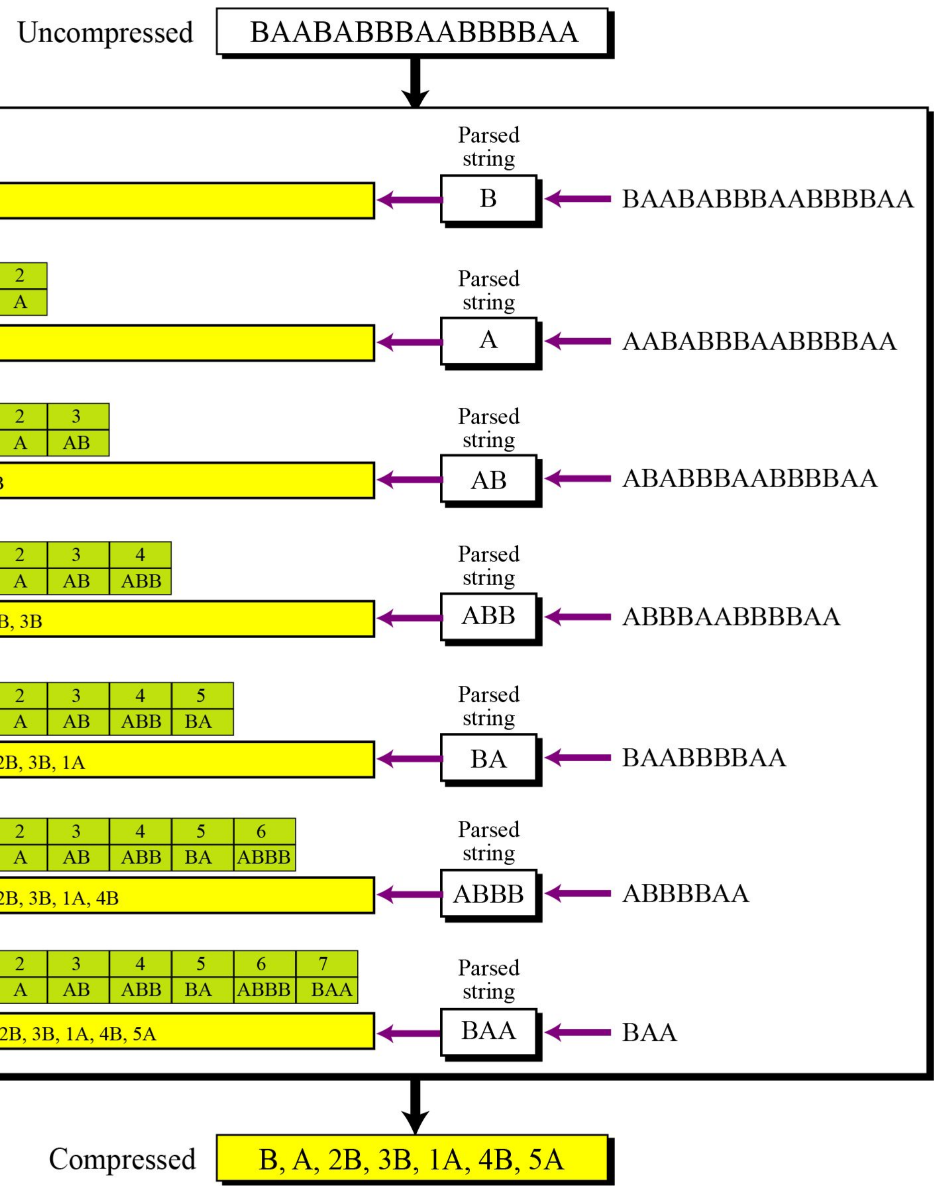


Figure 15.8 An example of Lempel Ziv encoding

Decompression

Decompression is the inverse of the compression process. The process extracts the substrings from the compressed string and tries to replace the indexes with the corresponding
entry in the dictionary, which is empty at first and built up gradually. The idea is that when an index is received, there is already an entry in the dictionary corresponding to that index.

Figure 15.9 An example of Lempel Ziv decoding