# CHAPTER 4

# *Digital Transmission*

## *Solutions to Review Questions and Exercises*

### Review Questions

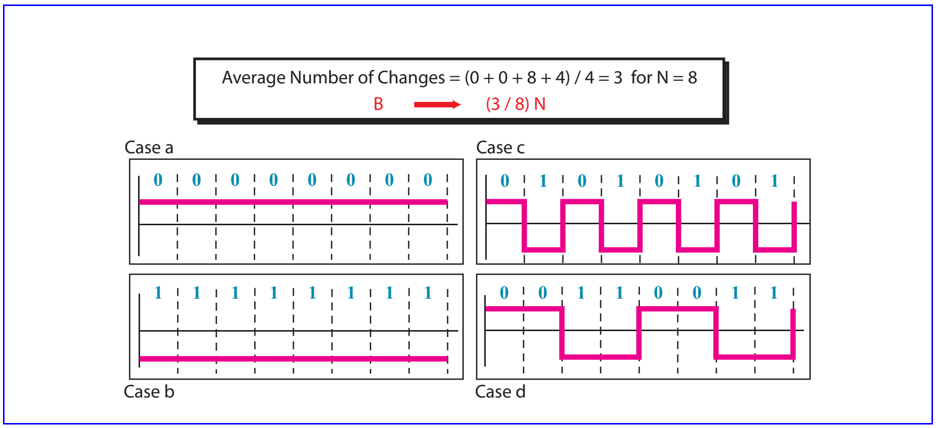
1. The three different techniques described in this chapter are ***line coding***, ***block coding***, and ***scrambling***.
2. A ***data element*** is the smallest entity that can represent a piece of information (a bit). A ***signal element*** is the shortest unit of a digital signal. Data elements are what we need to send; signal elements are what we can send. Data elements are being carried; signal elements are the carriers.
3. The ***data rate*** defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps). The ***signal rate*** is the number of signal elements sent in 1s. The unit is the baud.
4. In decoding a digital signal, the incoming signal power is evaluated against the ***baseline*** (a running average of the received signal power). A long string of 0s or 1s can cause ***baseline wandering*** (a drift in the baseline) and make it difficult for the receiver to decode correctly.
5. When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies, called ***DC components***, that present problems for a system that cannot pass low frequencies.
6. A ***self-synchronizing*** digital signal includes timing information in the data being transmitted. This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.
7. In this chapter, we introduced ***unipolar***, ***polar***, ***bipolar***, ***multilevel***, and ***multitransition*** coding.
8. ***Block coding*** provides redundancy to ensure synchronization and to provide inherent error detecting. In general, block coding changes a block of *m* bits into a block of *n* bits, where *n* is larger than *m*.
9. ***Scrambling,*** as discussed in this chapter, is a technique that substitutes long zero-level pulses with a combination of other levels without increasing the number of bits.

10. Both ***PCM*** and ***DM*** use sampling to convert an analog signal to a digital signal. PCM finds the value of the signal amplitude for each sample; DM finds the change between two consecutive samples.
11. In ***parallel transmission*** we send data *several* bits at a time. In ***serial transmission*** we send data *one* bit at a time.
12. We mentioned ***synchronous***, ***asynchronous***, and ***isochronous***. In both synchronous and asynchronous transmissions, a bit stream is divided into independent frames. In synchronous transmission, the bytes inside each frame are synchronized; in asynchronous transmission, the bytes inside each frame are also independent. In isochronous transmission, there is no independency at all. All bits in the whole stream must be synchronized.

## Exercises

13. We use the formula **s = c × N × (1/r)** for each case. We let c = 1/2.
    a. r = 1 → s = (1/2) × (1 Mbps) × 1/**1** = **500 kbaud**
    b. r = 1/2 → s = (1/2) × (1 Mbps) × 1/(**1/2**) = **1 Mbaud**
    c. r = 2 → s = (1/2) × (1 Mbps) × 1/**2** = **250 Kbaud**
    d. r = 4/3 → s = (1/2) × (1 Mbps) × 1/(**4/3**) = **375 Kbaud**
14. The number of bits is calculated as (0.2 /100) × (1 Mbps) = **2000 bits**
15. See Figure 4.1. Bandwidth is proportional to **(3/8)N** which is within the range in Table 4.1 (B = 0 to N) for the NRZ-L scheme.

**Figure 4.1** *Solution to Exercise 15*




16. See Figure 4.2. Bandwidth is proportional to **(4.25/8)N** which is within the range in Table 4.1 (B = 0 to N) for the NRZ-I scheme.
17. See Figure 4.3. Bandwidth is proportional to **(12.5 / 8) N** which is within the range in Table 4.1 (B = N to B = 2N) for the Manchester scheme.
18. See Figure 4.4. B is proportional to **(12/8) N** which is within the range in Table 4.1 (B = N to 2N) for the differential Manchester scheme.

**Figure 4.2** *Solution to Exercise 16*

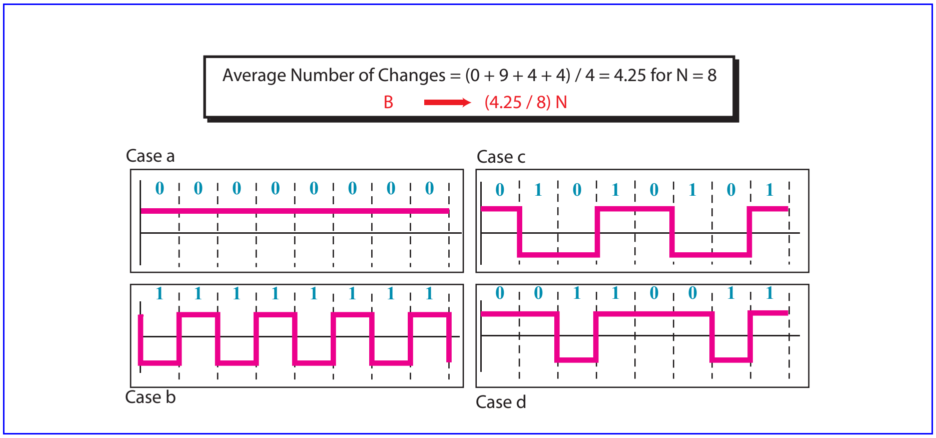



**Figure 4.3** *Solution to Exercise 17*

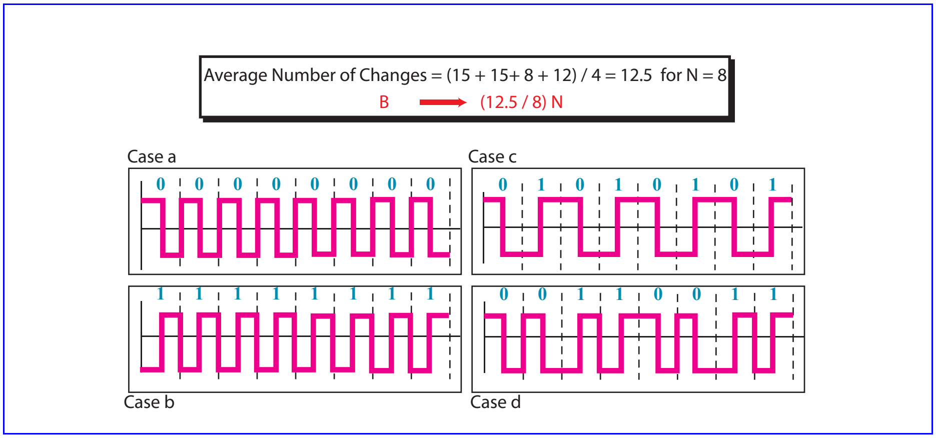
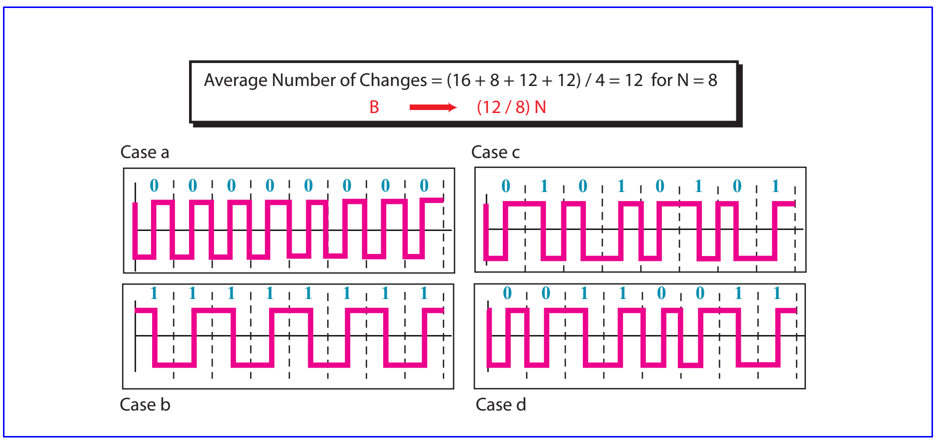
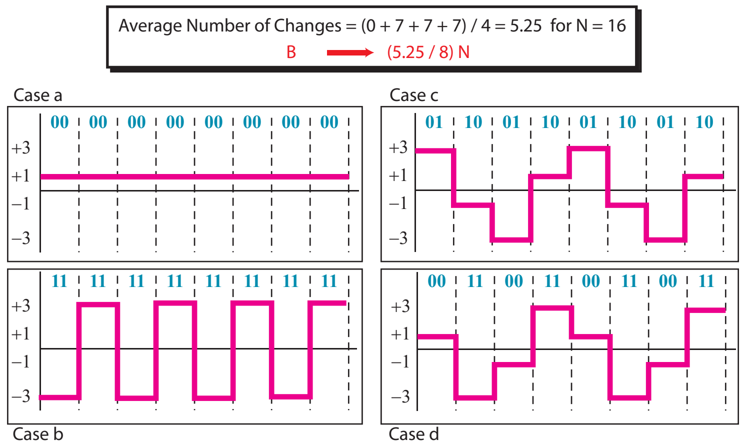



**Figure 4.4** *Solution to Exercise 18*



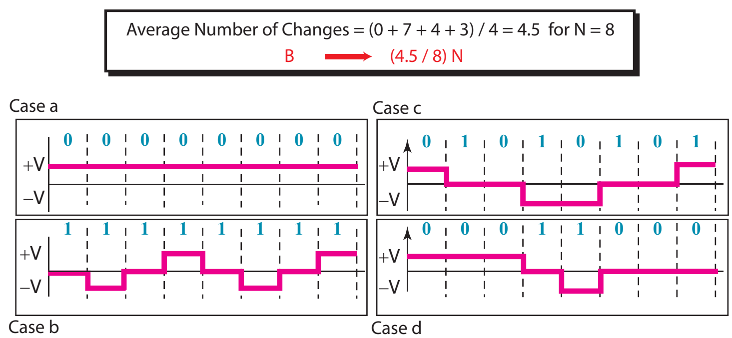
19. See Figure 4.5. B is proportional to **(5.25 / 16) N** which is inside range in Table 4.1 (B = 0 to N/2) for 2B/1Q.

**Figure 4.5** *Solution to Exercise 19*




20. See Figure 4.6. B is proportional to **(5.25/8)** × **N** which is inside the range in Table 4.1 (B = 0 to N/2) for MLT-3.

**Figure 4.6** *Solution to Exercise 20*




21. The data stream can be found as
    a. NRZ-I: **10011001**.
    b. Differential Manchester: **11000100**.
    c. AMI: **01110001**.
22. The data rate is 100 Kbps. For each case, we first need to calculate the value f / N. We then use Figure 4.6 in the text to find P (energy per Hz). All calculations are approximations.

a. f /N = 0/100 = 0 → **P = 1.0**

b. f /N = 50/100 = 1/2 → **P = 0.5**

c. f /N = 100/100 = 1 → **P = 0.0**

d. f /N = 150/100 = 1.5 → **P = 0.2**

23. The data rate is 100 Kbps. For each case, we first need to calculate the value f/N. We then use Figure 4.8 in the text to find P (energy per Hz). All calculations are approximations.

a. f /N = 0/100 = 0 → **P = 0.0**

b. f /N = 50/100 = 1/2 → **P = 0.3**

c. f /N = 100/100 = 1 → **P = 0.4**

d. f /N = 150/100 = 1.5 → **P = 0.0**

24.

a. The output stream is **01010 11110 11110 11110 11110 01001**.

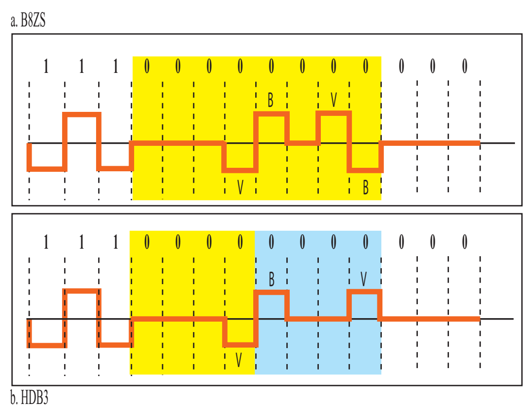
b. The maximum length of consecutive 0s in the input stream is **21**.

c. The maximum length of consecutive 0s in the output stream is **2**.

25. In 5B/6B, we have $2^5 = 32$ data sequences and $2^6 = 64$ code sequences. The number of unused code sequences is 64 – 32 = **32**. In 3B/4B, we have $2^3 = 8$ data sequences and $2^4 = 16$ code sequences. The number of unused code sequences is 16 – 8 = **8**.

26. See Figure 4.7. Since we specified that the last non-zero signal is positive, the first bit in our sequence is positive.

**Figure 4.7** *Solution to Exercise 26*




27.

a. In a low-pass signal, the minimum frequency 0. Therefore, we have

$f_{max} = 0 + 200 = 200$ KHz. → $f_s = 2 \times 200{,}000 =$ **400,000 samples/s**

b. In a bandpass signal, the maximum frequency is equal to the minimum frequency plus the bandwidth. Therefore, we have

$$f_{max} = 100 + 200 = 300 \text{ KHz.} \rightarrow f_s = 2 \times 300{,}000 = \mathbf{600{,}000 \text{ samples /s}}$$

28.

a. In a lowpass signal, the minimum frequency is 0. Therefore, we can say

$$f_{max} = 0 + 200 = 200 \text{ KHz} \rightarrow f_s = 2 \times 200{,}000 = \mathbf{400{,}000 \text{ samples/s}}$$

The number of bits per sample and the bit rate are

$$n_b = \log_2 1024 = 10 \text{ bits/sample} \qquad N = 400 \text{ KHz} \times 10 = \mathbf{4 \text{ Mbps}}$$

b. The value of $n_b = 10$. We can easily calculate the value of $SNR_{dB}$

$$SNR_{dB} = 6.02 \times n_b + 1.76 = \mathbf{61.96}$$

c. The value of $n_b = 10$. The minimum bandwidth can be calculated as

$$B_{PCM} = n_b \times B_{analog} = 10 \times 200 \text{ KHz} = \mathbf{2 \text{ MHz}}$$

29. The maximum data rate can be calculated as

$$N_{max} = 2 \times B \times n_b = 2 \times 200 \text{ KHz} \times \log_2 4 = \mathbf{800 \text{ kbps}}$$

30. We can first calculate the sampling rate (fs) and the number of bits per sample (nb)

$$f_{max} = 0 + 4 = 4 \text{ KHz} \rightarrow f_s = 2 \times 4 = \mathbf{8000 \text{ sample/s}}$$

We then calculate the number of bits per sample.

$$\rightarrow n_b = 30000 / 8000 = 3.75$$

We need to use the next integer $n_b = 4$. The value of $SNR_{dB}$ is

$$SNR_{dB} = 6.02 \times n_b + 1.72 = \mathbf{25.8}$$

31. We can calculate the data rate for each scheme:

**a.** NRZ $\rightarrow$ $N = 2 \times B = 2 \times 1 \text{ MHz} = \mathbf{2 \text{ Mbps}}$
**b.** Manchester $\rightarrow$ $N = 1 \times B = 1 \times 1 \text{ MHz} = \mathbf{1 \text{ Mbps}}$
**c.** MLT-3 $\rightarrow$ $N = 3 \times B = 3 \times 1 \text{ MHz} = \mathbf{3 \text{ Mbps}}$
**d.** 2B1Q $\rightarrow$ $N = 4 \times B = 4 \times 1 \text{ MHz} = \mathbf{4 \text{ Mbps}}$

32.

a. For synchronous transmission, we have $1000 \times 8 =$ **8000** bits.

b. For asynchronous transmission, we have $1000 \times 10 =$ **10000** bits. Note that we assume only one stop bit and one start bit. Some systems send more start bits.

c. For case a, the redundancy is 0%. For case b, we send 2000 extra for 8000 required bits. The redundancy is **25%**.