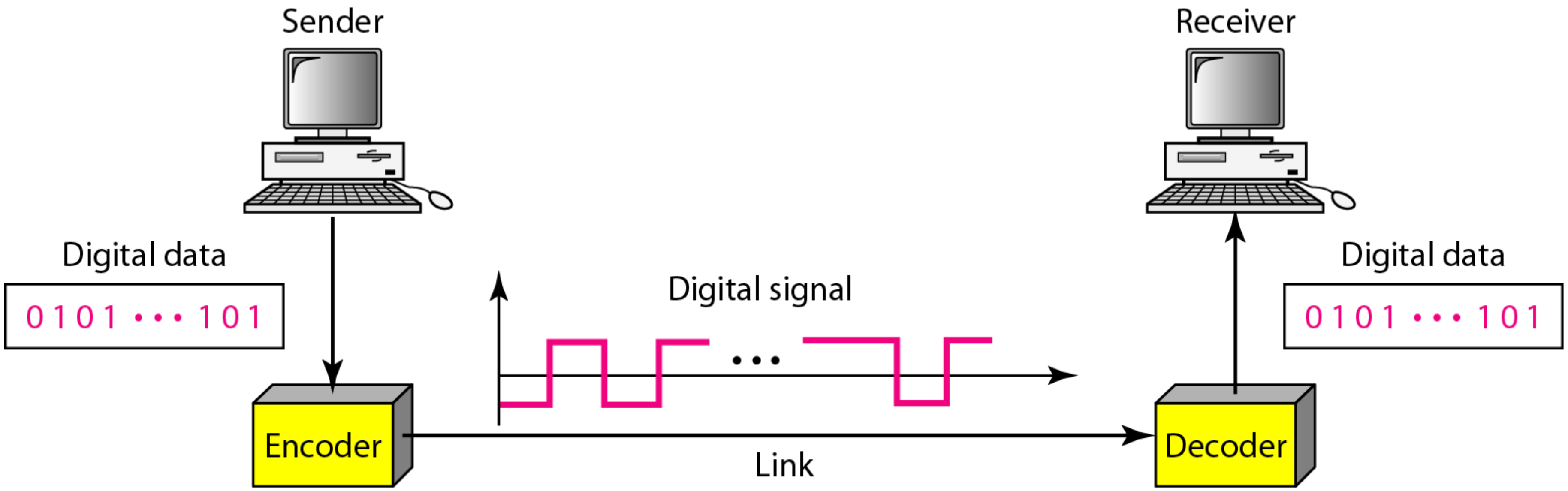
Chapter-4
Digital Transmission

Digital Transmission

Digital transmission mainly two types:

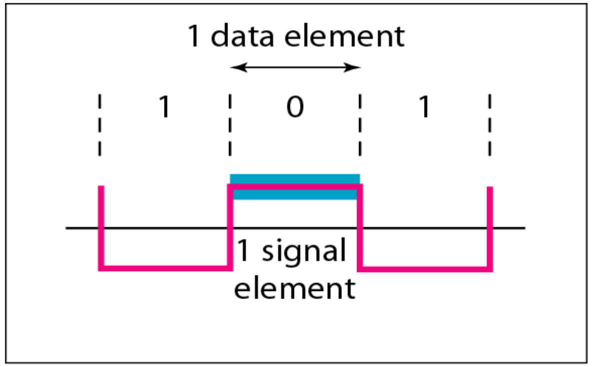
1. Digital data to Digital signal
2. Analog signal to Digital data



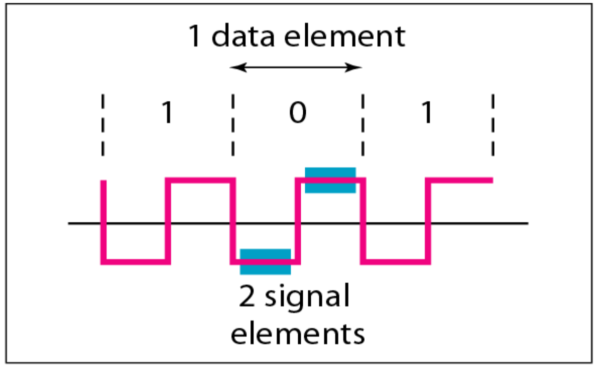
Data Element vs Signal Element

- ✓ 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.

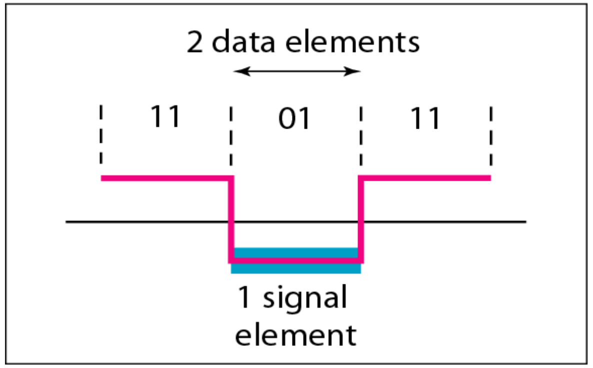
- ✓ Ratio, r = Data element/Signal element



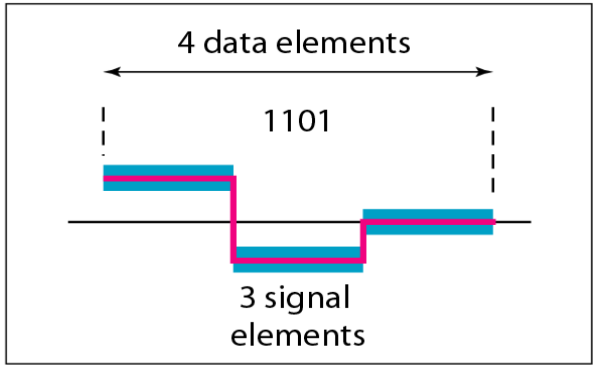
a. One data element per one signal element (r = 1)



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

Data Rate vs Signal Rate

❑**Data Rate**

✓ The data rate defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps).

✓ The data rate is sometimes called the bit rate.

❑**Signal Rate**

✓ The signal rate is the number of signal elements sent in 1s. The unit is the baud.

✓ The signal rate is sometimes called the pulse rate, the modulation rate, or the baud rate.

❑**Goal of Data Communication**

✓ Increase the data rate while decreasing the signal rate.

✓ Increasing the data rate increases the speed of transmission.

✓ Decreasing the signal rate decreases the bandwidth requirement.

Data Rate vs Signal Rate

✓**Relationship between data rate and signal rate:**

S = c * N * 1/r baud

Where *N* is the data rate (bps); c is the case factor, which varies for each case; S is the number of signal elements; and *r* is the previously defined factor. Here, c=1 worst case, c=0 best case and c=1/2 average case.

Example: 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

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

Basic Terminologies for Line Coding Scheme

❑ Baseline Wandering

- ✓In decoding a digital signal, the receiver calculates a running average of the received signal power (voltage or amplitude).
- ✓This average is called the **baseline.**
- ✓The incoming signal power is evaluated against this baseline to determine the value of the data element.
- ✓A long string of 0's or 1's can cause a drift in the baseline (baseline wandering) and make it difficult for the receiver to decode correctly.

Basic Terminologies for Line Coding Scheme

❑ **DC Components Problem**

✓ When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies.

✓ These frequencies around zero, called DC (direct-current) components.

✓ For example, a telephone line cannot pass frequencies below 200 Hz.

Basic Terminologies for Line Coding Scheme

❑ **Self Synchronization**

✓ To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals.

✓ If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals.

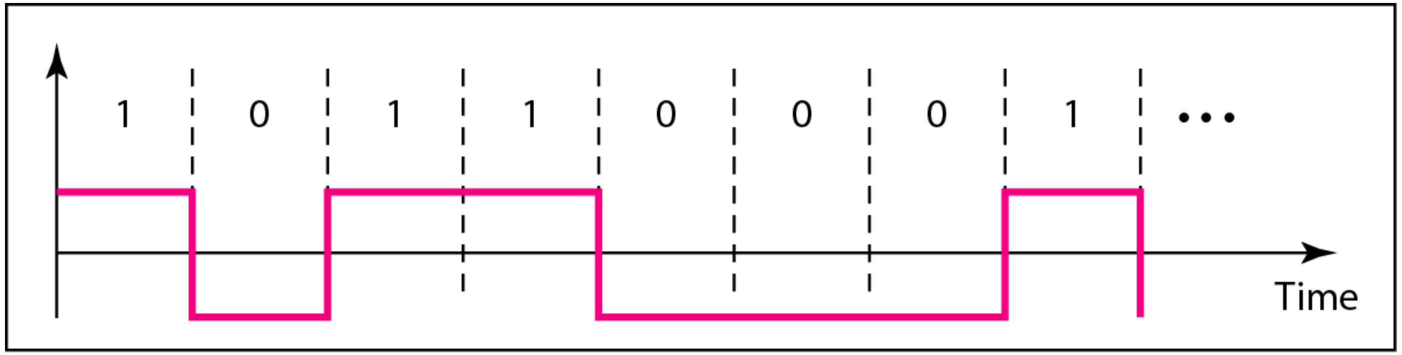
✓ A self-synchronizing digital signal includes timing information in the data being transmitted.

✓ This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.

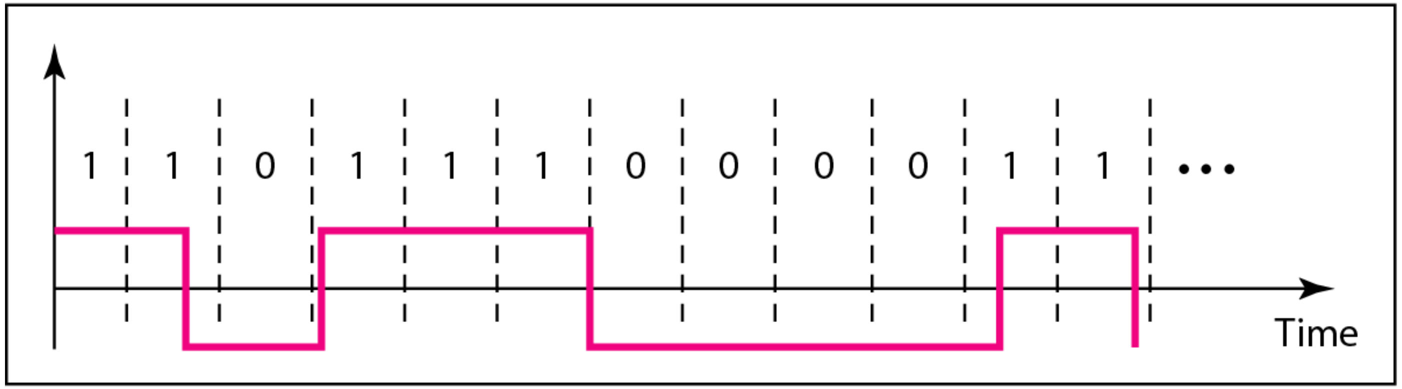
✓ **Notes: If receiver faster than sender than extra bit will be added to the receiver. If receiver is slower than the sender than the receiver might loss some bits.**

Basic Terminologies for Line Coding Scheme

- **Self Synchronization**



a. Sent



b. Received

Figure: Effect of lack of synchronization

Basic Terminologies for Line Coding Scheme

❑ **Self Synchronization**

✓ **Example:** 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:

100 …………… 0.1 extra bit

1………………. (0.1/100) extra bit

1000…………….(0.1/100) * 1000 = 1 extra bit

1000 bits sent	1001 bits received	1 extra bps

1,000,000 bits sent	1,001,000 bits received	1000 extra bps

Basic Terminologies for Line Coding Scheme

- **Built-in Error Detection:-** It is desirable to have a built-in error-detecting capability in the generated code to detect some of or all the errors that occurred during transmission.
- **Immunity to Noise and Interference:-** Another desirable code characteristic is a code that is immune to noise and other interferences.
- **Complexity:-** A complex scheme is more costly to implement than a simple one. For example, a scheme that uses four signal levels is more difficult to interpret than one that uses only two levels.

Digital to Digital Conversion

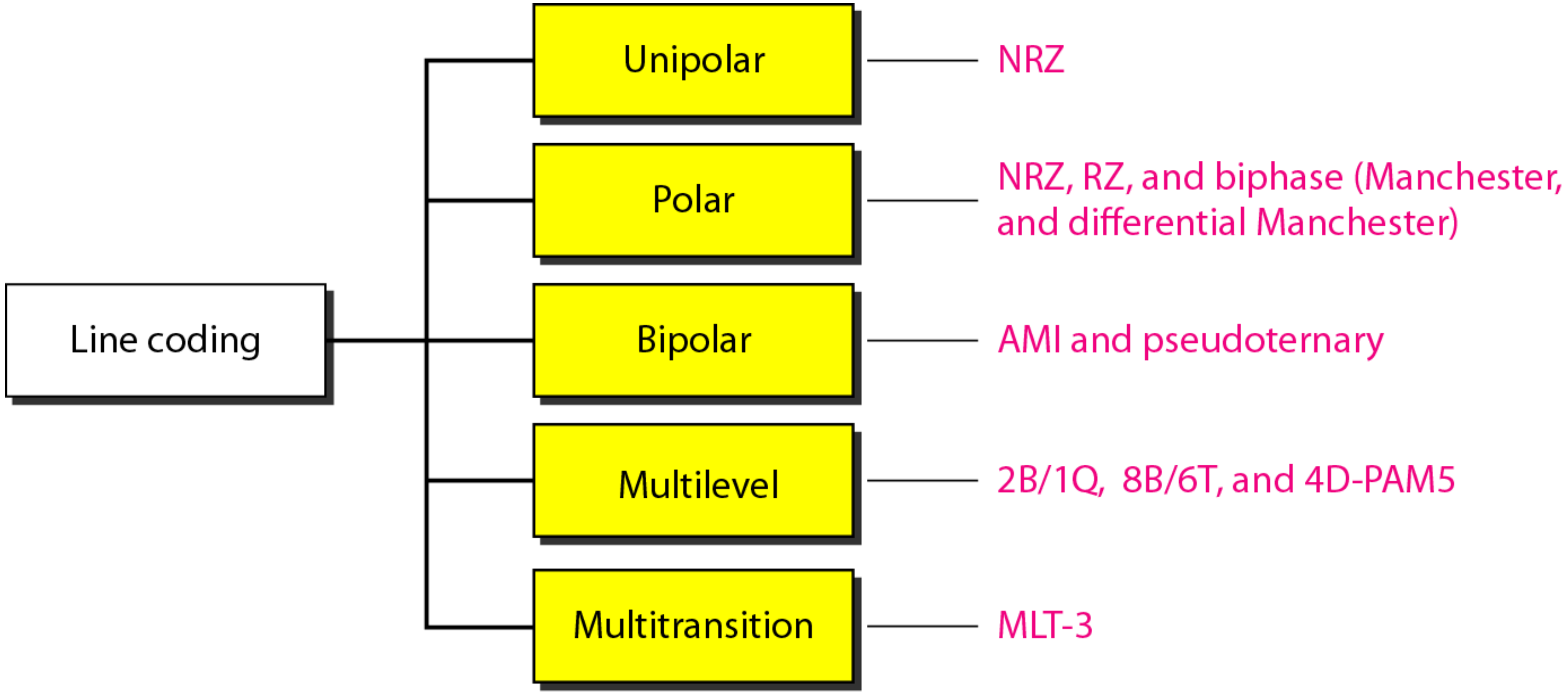
- ✓ Digital to digital conversion is used to represent digital data in digital signal.
- ✓ The conversion involves three techniques: **line coding, block coding, and scrambling.**

❑ **Line Coding**

- ✓ Line coding is the process of converting digital data to digital signals.
- ✓ Line coding converts a sequence of bits to a digital signal.
- ✓ At the sender, digital data are encoded into a digital signal.
- ✓ At the receiver, the digital data are recreated by decoding the digital signal.

Line Coding

Line coding is divided into five categories:-

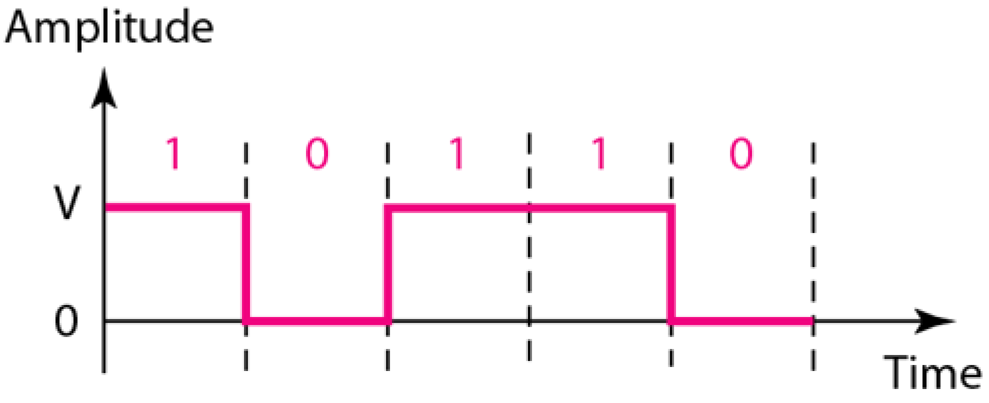


Line Coding

❑**Unipolar (Not-Return-to-Zero)**

✓Positive voltage defines bit 1 and the zero voltage defines bit 0.

✓It is called NRZ because the signal does not return to zero at the middle of the bit.



Line Coding

- ❑ **Advantages of Unipolar (Not-Return-to-Zero)**
- ✓ Very simple.
- ✓ Occupies lesser bandwidth.
- ❑ **Disadvantages of Unipolar NRZ**
- ✓ It has DC component problem for 1’s.
- ✓ It suffers from baseline wandering problem for 1’s.
- ✓ No error correction done.
- ✓ No self synchronization.
- ✓ Greater normalized power.

Line Coding

❑ **Polar (NRZ-L)**

✓ The level of voltage determines the values of bit.

✓ Bit 1→ Negative Voltage, Bit→0 Positive Voltage.

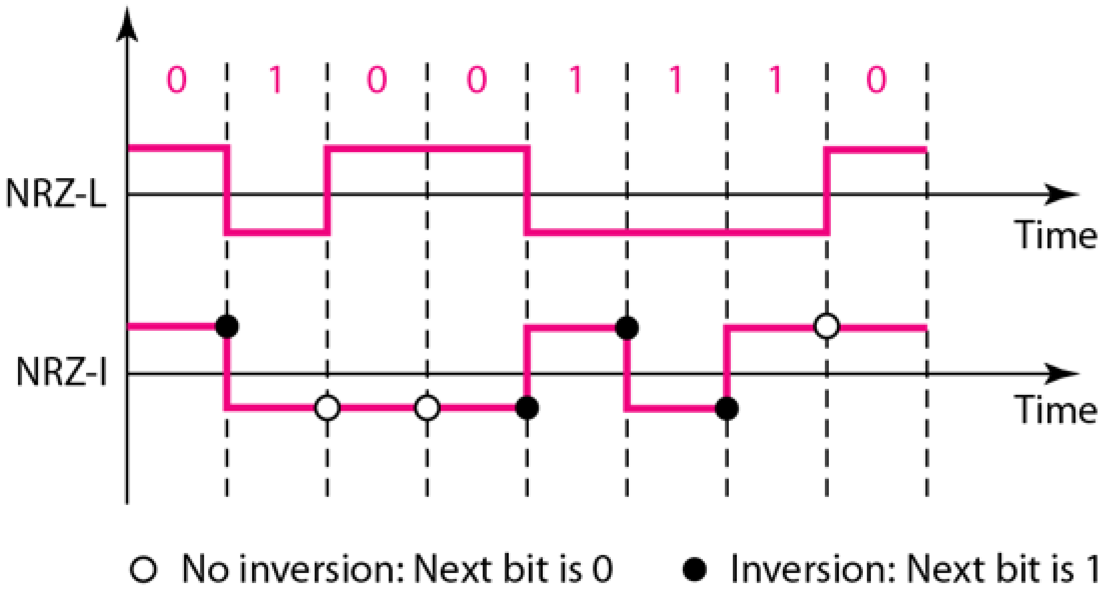
✓ The voltages are on both sides of the time axis.

❑ **Polar (NRZ-I)**

✓ Change or lack of change in the level of voltage determines the values of bit.

✓ Bit 1→ Change, Bit 0→No Change.

✓ Let us assume the last voltage level was positive.



Line Coding

- ❑ **Average Signal Rate:** $\boldsymbol{S_{average} = c * N * \frac{1}{r}}$

Here, for average signal rate, the case factor c=1/2 and N=data rate, $r = {}^{data\ element}/_{signal\ element} = \frac{1}{1} = 1$

$$S_{average} = \frac{1}{2} * N$$

- ❑ **Example:** A system is using NRZ-I to transfer 1-Mbps data. What are the average signal rate and minimum bandwidth?

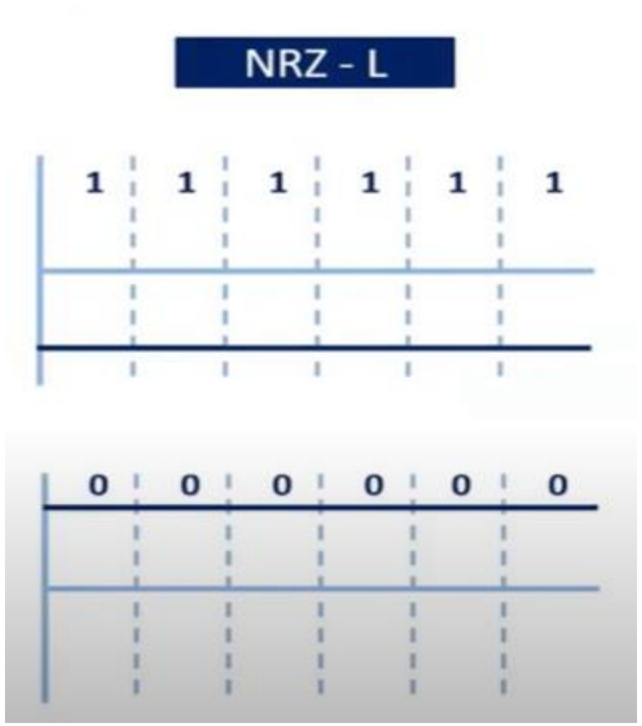
- ✓ **Solution:** The average signal rate is S= c x N x R = 1/2 x N x 1 = 500 kbaud. The minimum bandwidth for this average baud rate is B(min) = S = 500 kHz.

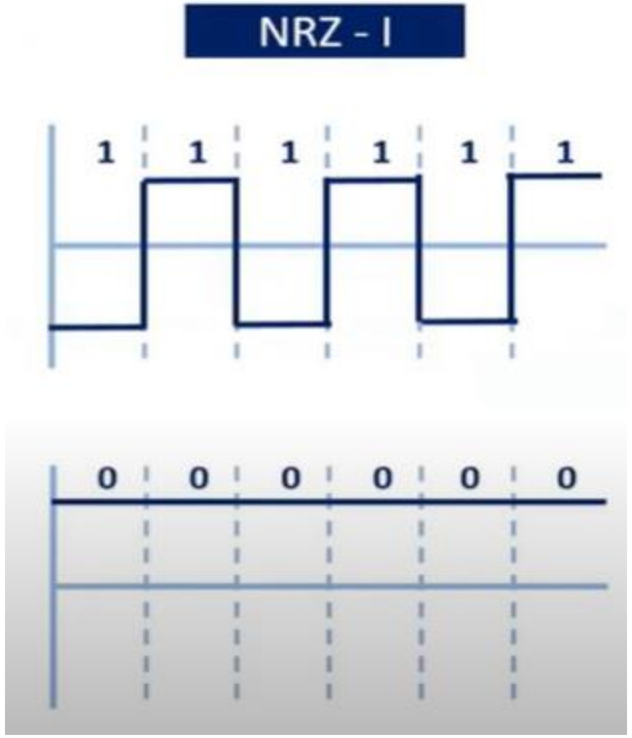
Line Coding

❑ **Disadvantages of Polar (NRZ-L & NRZ-I):**

✓ Baseline wandering problems for long 0’s and long 1’s (NRZ-L).

✓ Baseline wandering problems for long sequence of 0’s (NRZ-I).

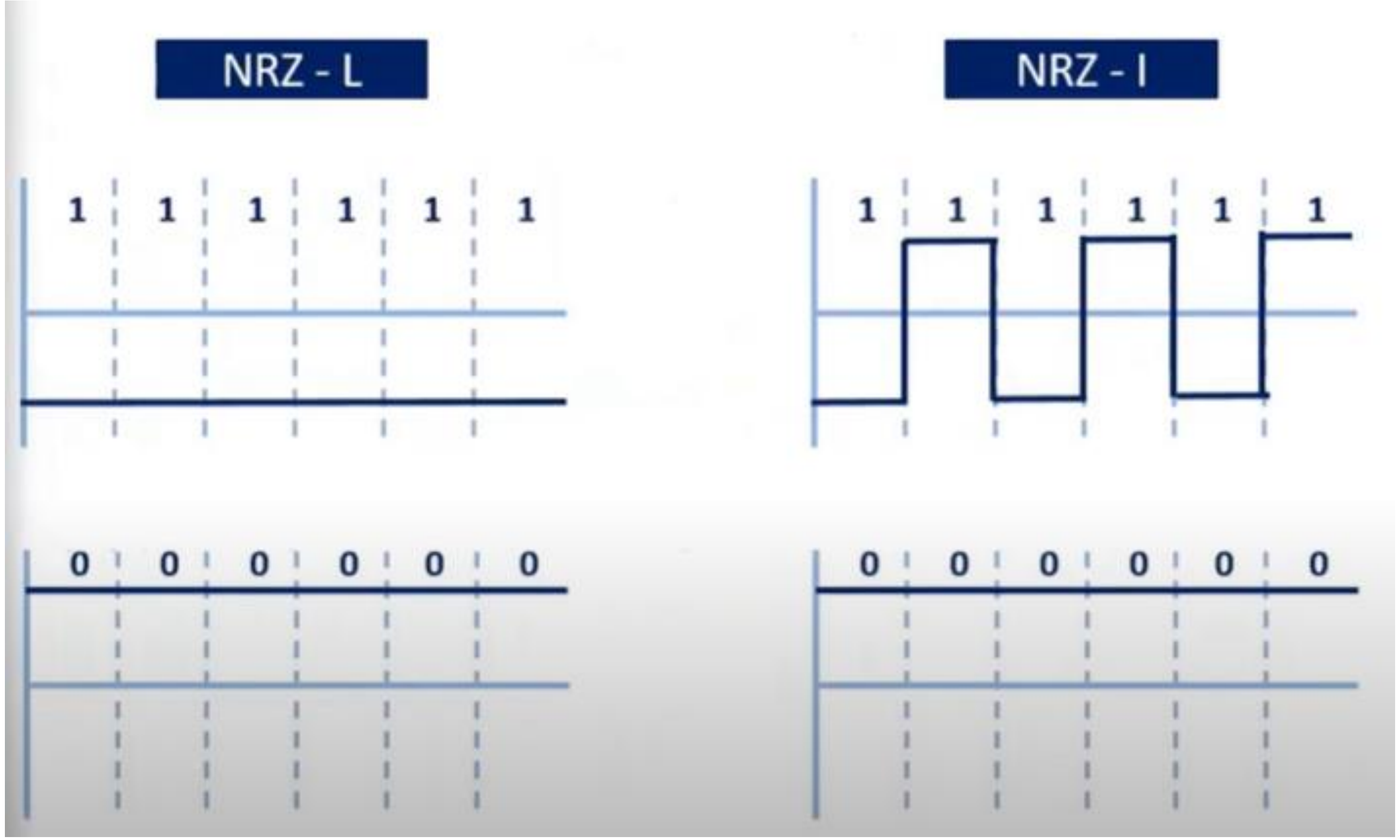




Line Coding

❑Disadvantages of Polar (NRZ-L & NRZ-I):

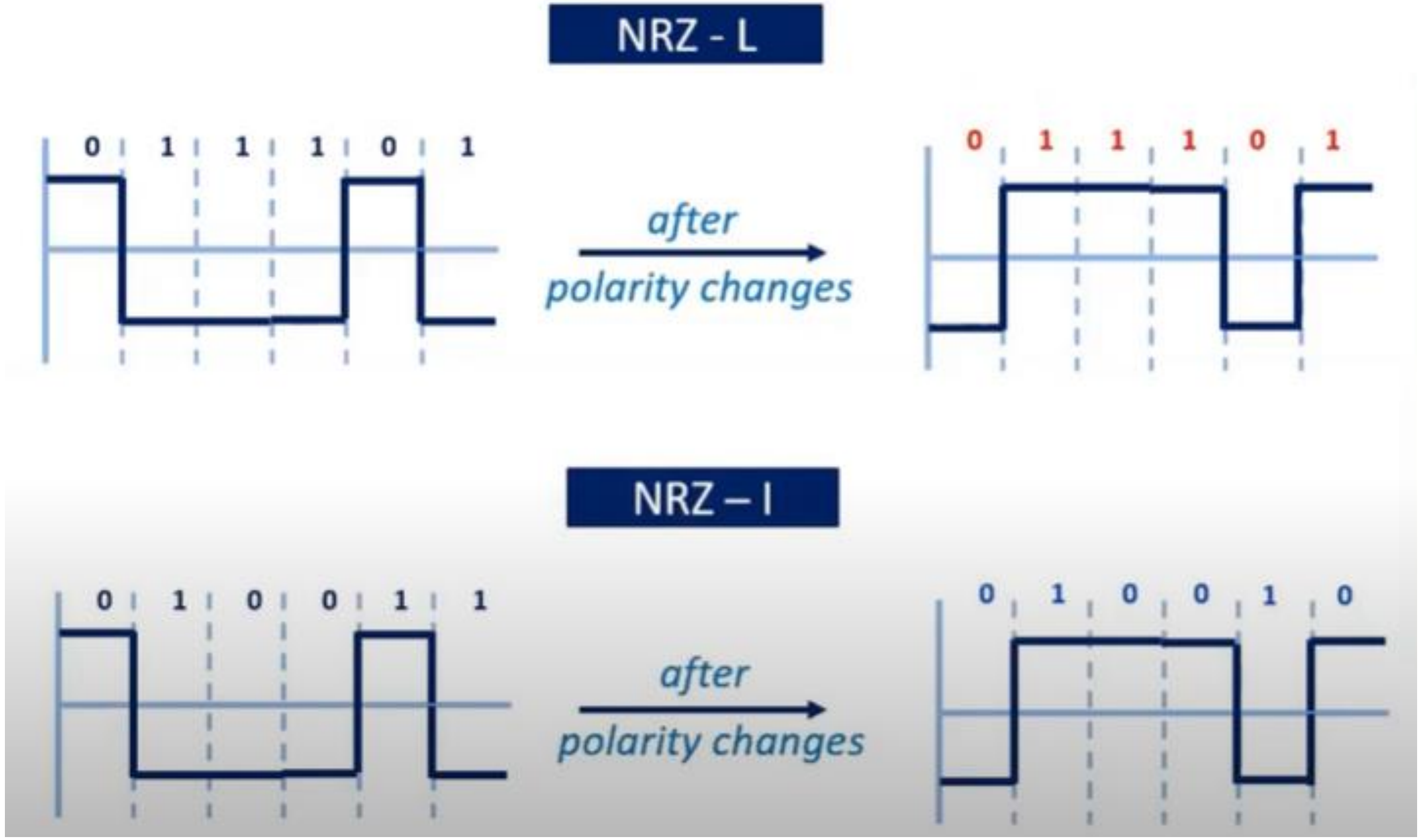
✓ Synchronization problem also exists in both NRZ-L and NRZ-I and like before the problem is more severe in NRZ-L.



Line Coding

❑Disadvantages of Polar (NRZ-L & NRZ-I):

✓ Change of polarity in the system changes the interpretation of data in NRZ-L not in NRZ-I.



Line Coding

❑ **Disadvantages of Polar (NRZ-L & NRZ-I):**

✓ DC component problems for 0’s and 1’s (NRZ-L).

✓ DC component problems for 0’s (NRZ-I).

Line Coding

Due to lack of synchronization in NRZ scheme, the receiver does not know when one bit has ended and next bit is starting.

Line Coding

Polar return to zero scheme solves the problem by marking the middle of the by a transition to zero.

Line Coding

❑ **Polar (Return-to-Zero)**

✓ It uses three voltage levels +V, 0 and –V.

✓ Negative voltage represents ‘0’ and positive voltage represents ‘1’.

✓ Always goes to zero at the middle of the bit and stays there until the start of the next bit.

❑ **Advantages**

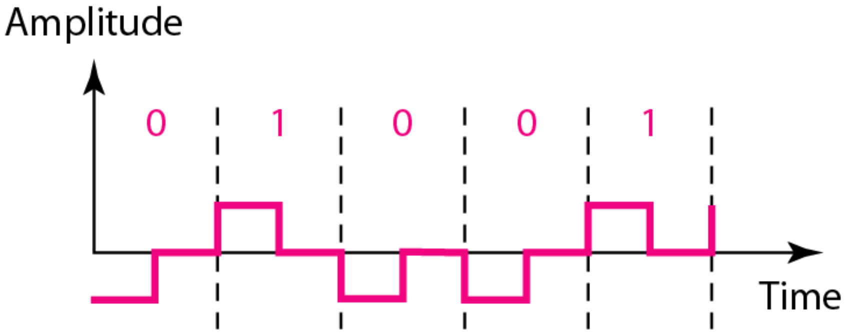
✓ There is no DC component problem.

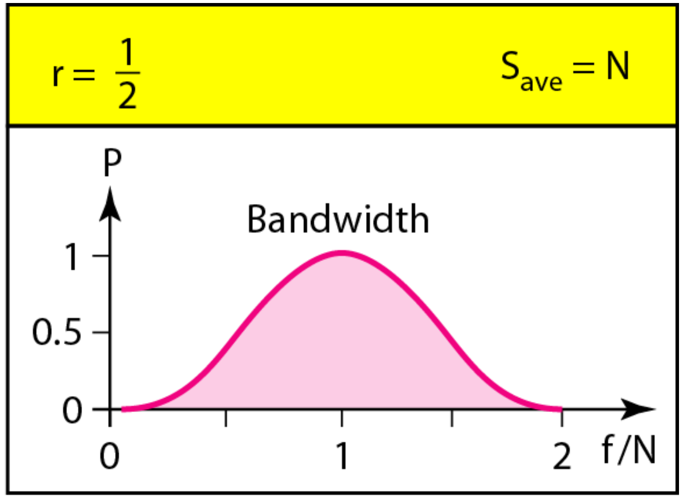
✓ Synchronization is possible.

❑ **Disadvantages**

✓ It uses three level of voltage and complex.

✓ Signal rate is high.





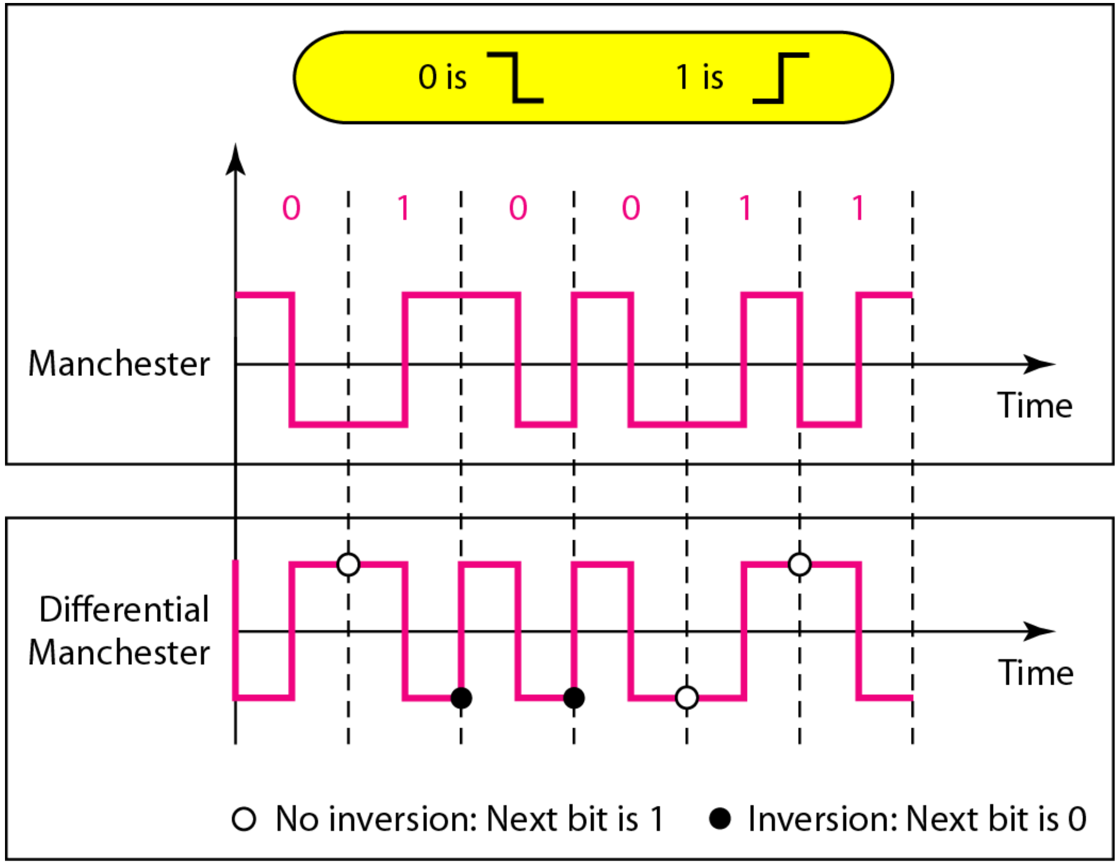
Line Coding

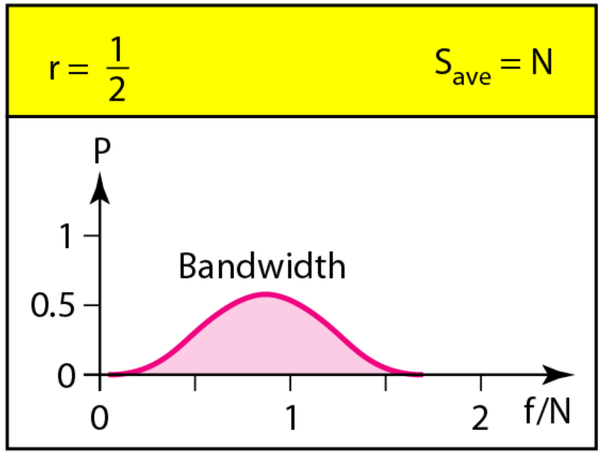
❑ **Polar Bi-phase (Manchester and Differential Manchester)**

✓ Manchester coding consists of combining the NRZ-L and RZ schemes.

✓ Differential Manchester coding consists of combining the NRZ-I and RZ schemes.

✓ Let us assume last voltage level was positive.





Line Coding

- ❑ **Advantages**

✓ There is no DC component problem.

✓ Synchronization is possible.

✓ There is no baseline wandering problem.

- ❑ **Disadvantages**

✓ Signal rate is high.

Note

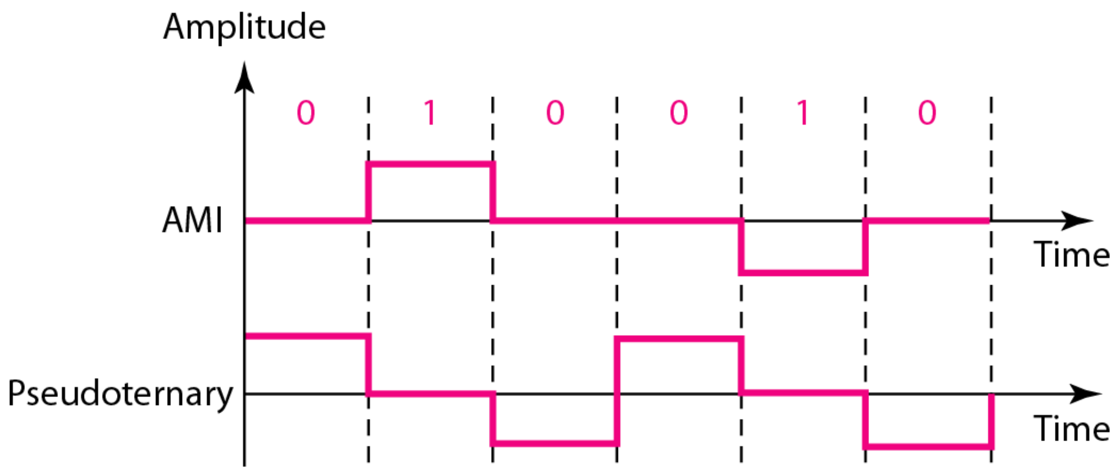
In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

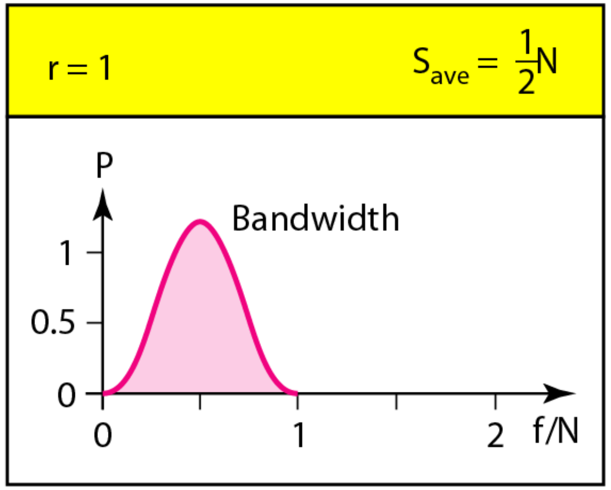
Line Coding

- **In bipolar encoding (*multilevel binary*),** there are three voltage levels: positive, negative, and zero.
- The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.
- **It has two types:**

1. Alternative Mark Inversion (AMI): For AMI, bit 1→ alternate between +V and –V and bit 0→Zero Voltage. Let's assume last non-zero pulse was negative.

2. Pseudo-ternary: For AMI, bit 0→ alternate between +V and –V and bit 1→Zero Voltage. Let's assume last zero pulse was negative.





Line Coding

- ❑ **Advantages**

- ✓ No baseline wandering

- ✓ No dc component problem

- ❑ **Disadvantages**

- ✓ Synchronization problem

- ❑ **Uses:** It is commonly used for long-distance communication.

Line Coding

❑Multilevel Schemes

- ✓ If we have m data elements then we can produce a combination of 2^m data patterns.
- ✓ If we have L different levels, then we can produce L^n combinations of signal patterns.
- ✓ If $2^m < L^n$, data patterns occupy only a subset of signal patterns. The subset can be carefully designed to prevent baseline wandering, to provide synchronization, and to detect errors that occurred during data transmission.
- ✓ Data encoding is not possible if $2^m > L^n$ because some of the data patterns cannot be encoded.
- ✓ These types of coding can be expressed as ***mBnL****,* where *m* is the length of the binary pattern, *B* means binary data, *n* is the length of the signal pattern, and *L* is the number of levels in the signaling. A letter is often used in place of L: *B* (binary) for *L* =2, *T* (ternary) for *L* =3, and Q (quaternary) for *L* =4. Note that the first two letters define the data pattern, and the second two define the signal pattern.

Line Coding

❑**Multilevel Schemes**

❑**2B1Q**

✓In this type of encoding m =2, n =1, and L =4 (quaternary).

❑**Disadvantages:**

✓No self synchronization.

✓No error detection.

✓It has dc component problem.

❑ **Uses:**

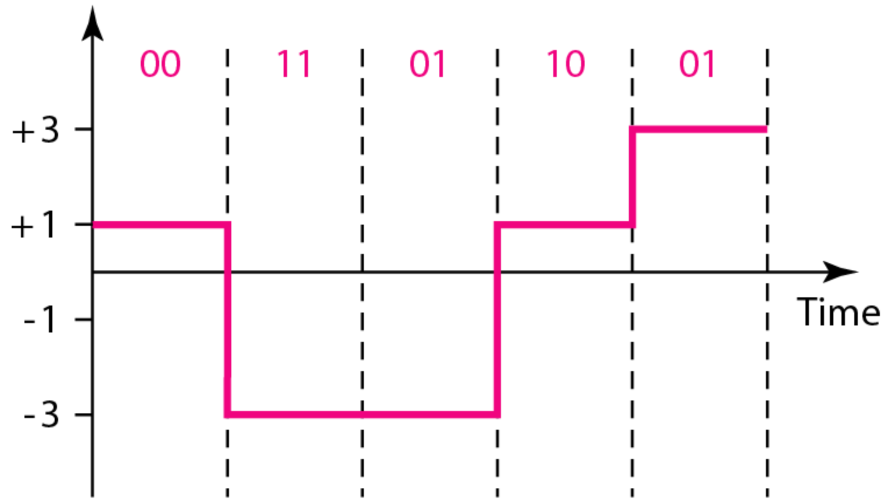
✓2BIQ is used in DSL (Digital Subscriber Line) technology to provide a high-speed connection to the Internet by using subscriber telephone lines.

Line Coding

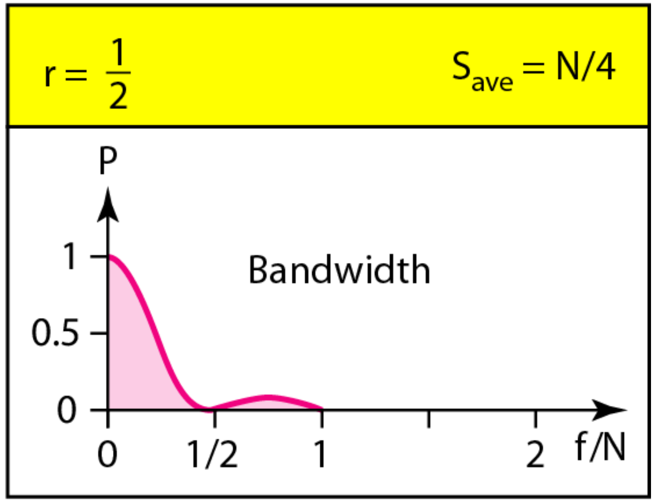
❑ **2B1Q**

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

Transition table



Assuming positive original level



Line Coding

❑**8B6T**

✓Here, 8 bits are used as a pattern of 6 signal elements, where the signal has three levels (ternary). In this type of scheme, we can have 28 =256 different data patterns and 36 =478 different signal patterns.

✓ There are 478 - 256 =222 redundant signal elements that provide synchronization and error detection.

✓Part of the redundancy is also used to provide DC balance.

✓Each signal pattern has a weight of 0 or +1 DC values.

✓This means that there is no pattern with the weight 1.

✓To make the whole stream Dc-balanced, the sender keeps track of the weight.

✓If two groups of weight 1 are encountered one after another, the first one is sent as is, while the next one is totally inverted to give a weight of -1.

Line Coding

❑8B6T

- ✓The three possible signal levels are represented as -,0, and +. The first 8-bit pattern 00010001 is encoded as the signal pattern -0-0++ with weight 0;
- ✓The second 8-bit pattern 010 10011 is encoded as - + - + + 0 with weight +1.
- ✓The third bit pattern should be encoded as + - - + 0 + with weight +1.
- ✓To create DC balance, the sender inverts the actual signal. The receiver can easily recognize that this is an inverted pattern because the weight is -1.
- ✓The pattern is inverted before decoding.
- ✓The average signal of the scheme theoretically, $S_{avg} = \frac{1}{2} * N * \frac{6}{8} = \frac{3N}{8}$
- ✓In practice, the minimum bandwidth = $\frac{3N}{8}$

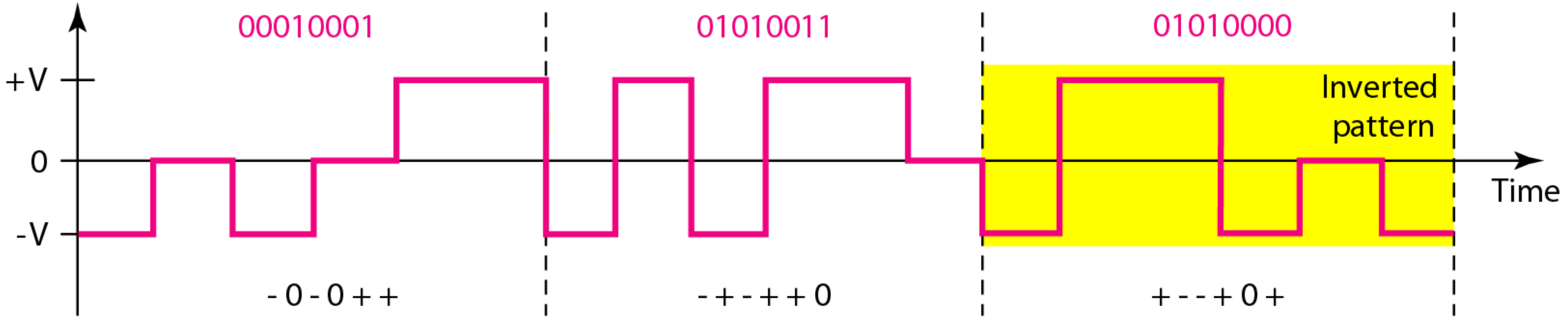
Line Coding

❑ **Advantages of 8B6T**

✓ No dc component problem.

✓ Synchronization is available.

✓ Error detection available.



Line Coding

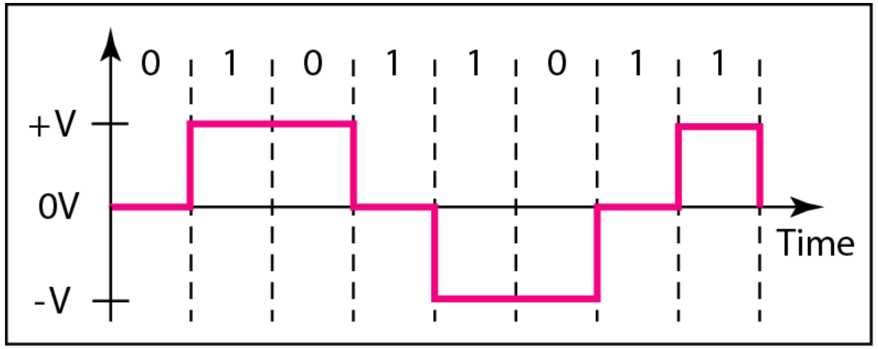
❑ **Multiline Transmission (MLT-3)**

✓ The multiline transmission, three level (MLT-3) scheme uses three levels (+*v*, 0, and - *V)* and three transition rules to move between the levels.

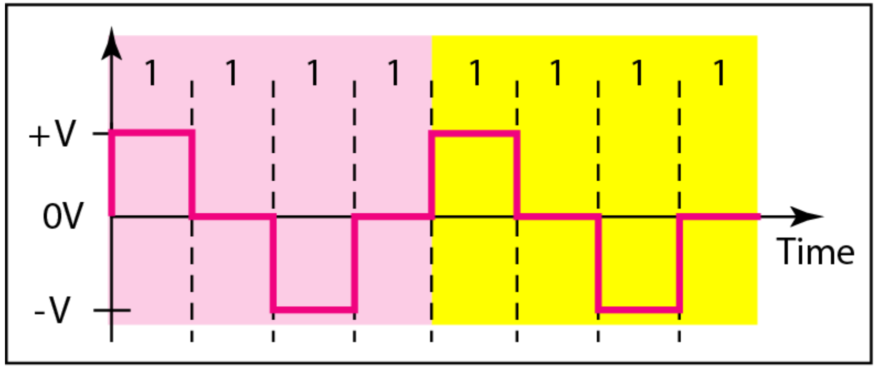
1. If the next bit is 0, there is no transition.
2. If the next bit is 1 and the current level is not 0, the next level is 0.
3. If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level.

Line Coding

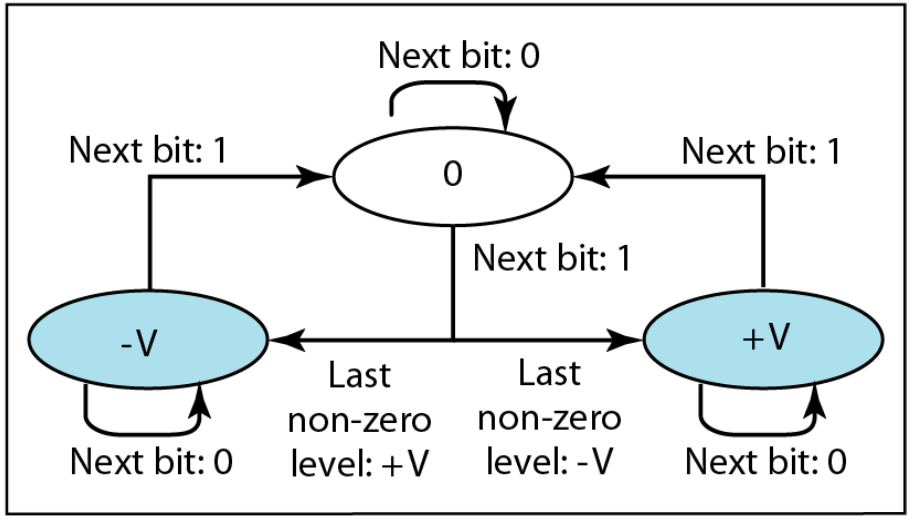
- **Multiline Transmission (MLT-3)**



a. Typical case



b. Worse case

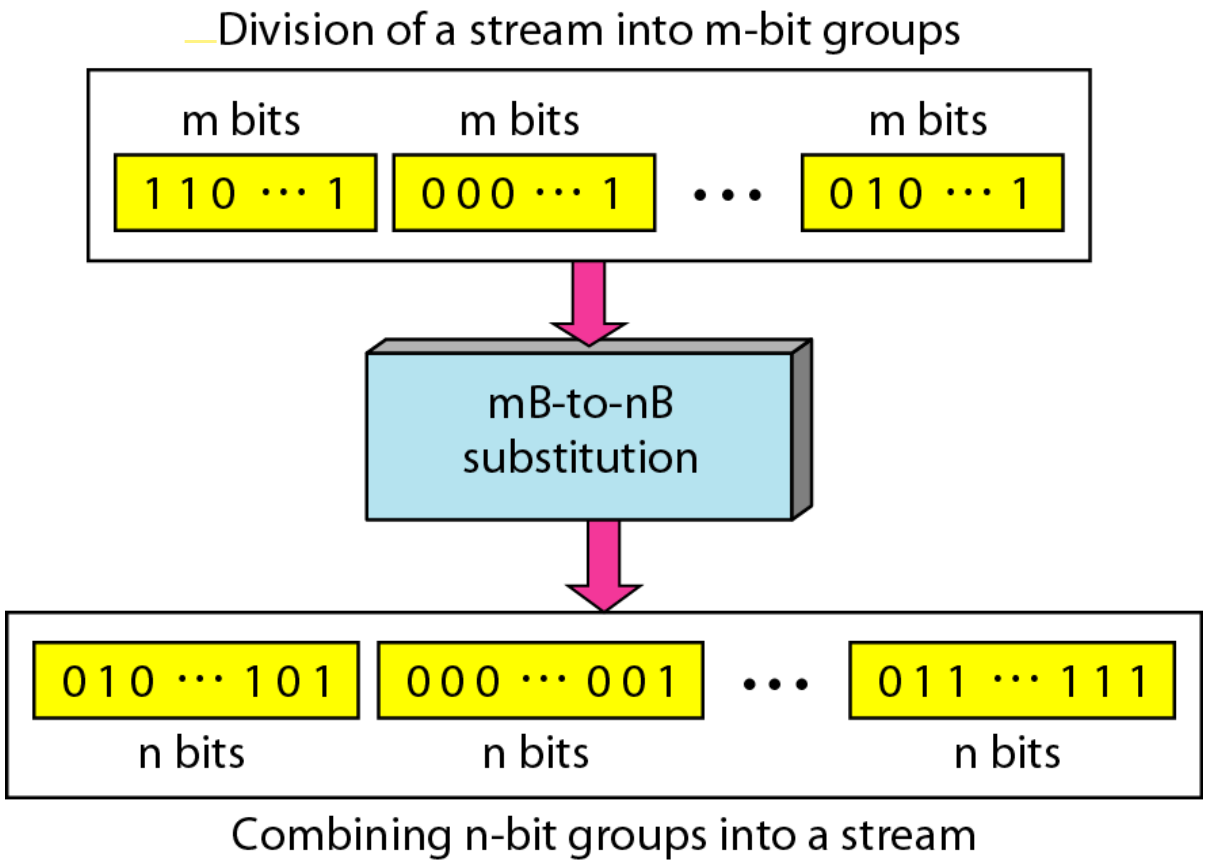


c. Transition states

Block Coding

- ✓ Block coding is applied before line coding.
- ✓ It provides redundancy and improves the performance of line coding.
- ✓ In general, block coding changes a block of *m* bits into a block of *n* bits, where *n* is larger than *m*. Block coding is referred to as an *mB/nB* encoding technique.
- ✓ It works in three steps: **1. Division 2. Substitution 3. Combination**
- ✓ In the division step, a sequence of bits is divided into groups of *m* bits.
- ✓ In the substitution step, it substitutes an m-bit group for an n-bit group.
- ✓ Finally, the n-bit groups are combined together to form a stream.

Block Coding



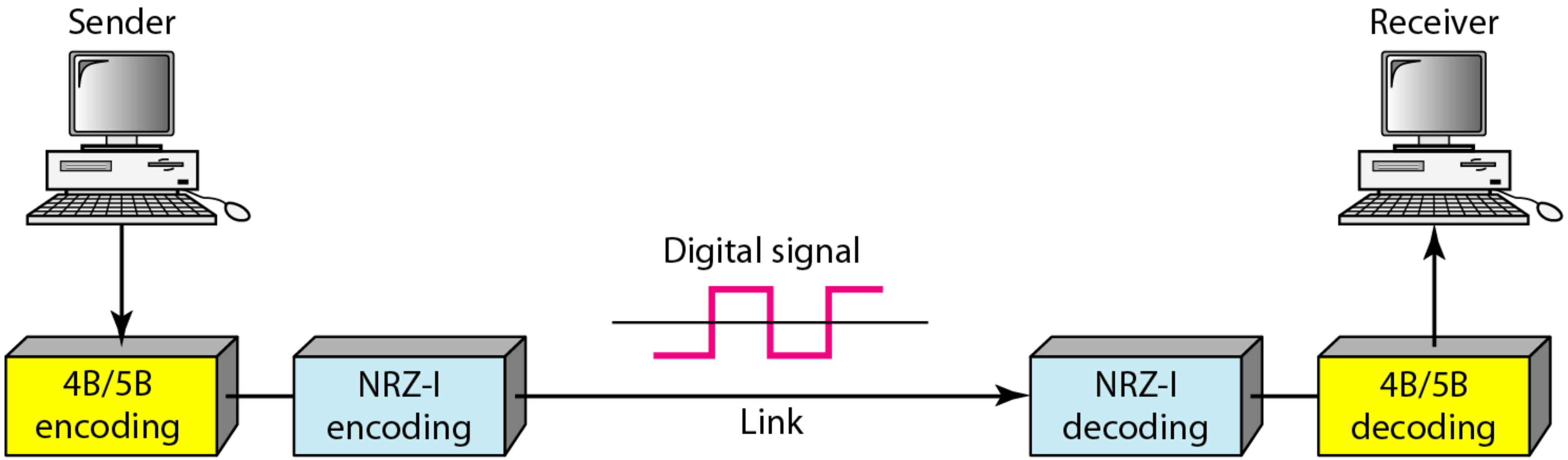
Block Coding

❑ **4B/5B**

- ✓ The four binary/five binary (4B/5B) coding scheme was designed to be used in combination with NRZ-I and solve the synchronization problem of NRZ-I.
- ✓ It replaces each block of 4 bits with a block of 5 bits has no more than one leading zero (left bit) and no more than two trailing zeros (right bits).
- ✓ A group of 4 bits can have only 16 different combinations while a group of 5 bits can have 32 different combinations.
- ✓ This means that there are 16 groups that are not used for 4B/5B encoding.
- ✓ Some of these unused groups are used for control purposes; the others are not used at all.
- ✓ The latter provide a kind of error detection. If a 5-bit group arrives that belongs to the unused portion of the table, the receiver knows that there is an error in the transmission.

Block Coding

- 4B/5B



Block Coding

❑ **4B/5B Mapping Table**

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		

Block Coding

- ❑ **4B/5B**

- ❑ **Disadvantages**

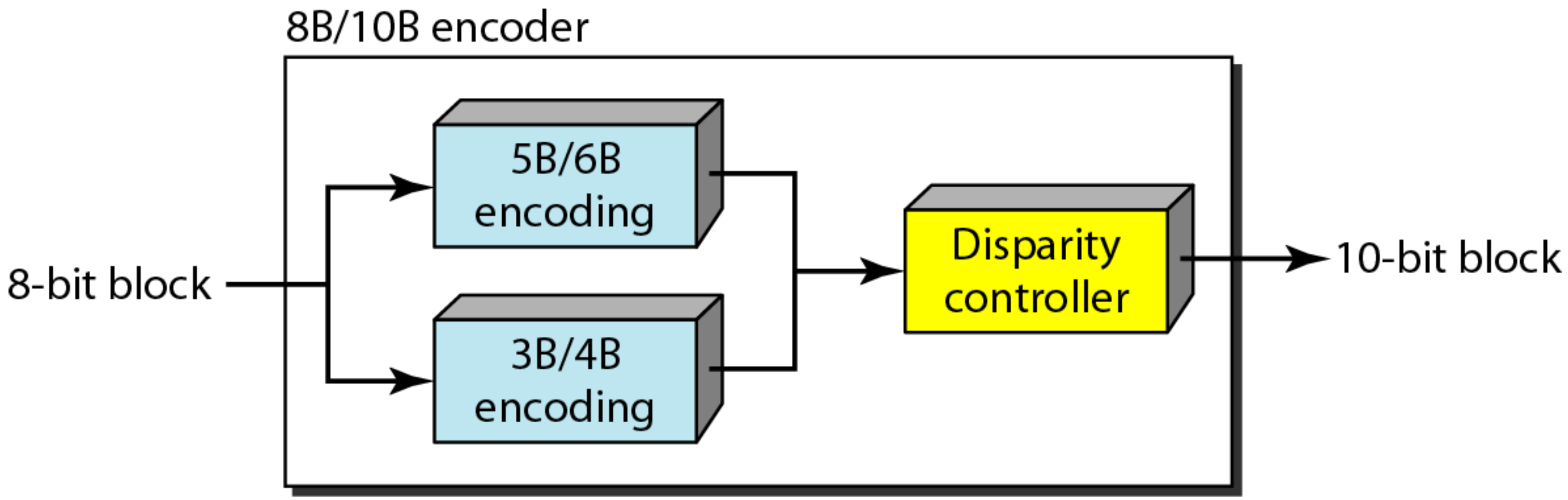
- ✓ It can’t solve the dc component problem of NRZ-I.

- ✓ It increases the signal rate of NRZ-I.

Block Coding

❑8B/10B

- ✓ A group of 8 bits of data is now substituted by a l0-bit code.
- ✓ It provides greater error detection capability than 4B/5B. The 8BIlOB block coding is actually a combination of 5B/6B and 3B/4B encoding.
- ✓ The most five significant bits of a 10-bit block is fed into the 5B/6B encoder; the least 3 significant bits is fed into a 3B/4B encoder.



Block Coding

❑**8B/10B**

✓ To prevent a long run of consecutive 0's or 1's, the code uses a disparity controller which keeps track of excess 0's over 1's (or 1's over 0's).

✓ If the bits in the current block create a disparity that contributes to the previous disparity (either direction), then each bit in the code is complemented (a 0 is changed to a 1 and a 1 is changed to a 0).

✓ The coding has 210 - 28 = 768 redundant groups that can be used for disparity checking and error detection.

✓ In general, the technique is superior to 4B/5B because of better built-in error-checking capability and better synchronization.

Block Coding

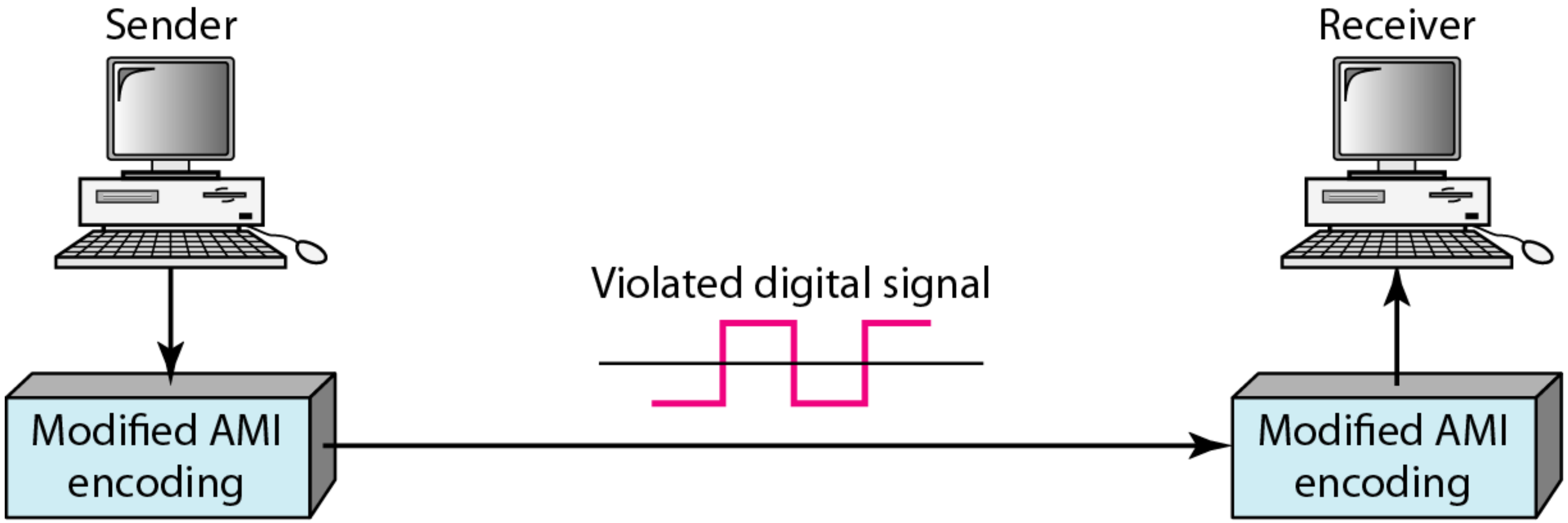
❑**Scrambling** (Modified Bipolar AMI)

✓ Provide synchronization without increasing the number of bits.

✓ Part of AMI rule is modified to include scrambling.

✓ It is done as same time as line coding.

✓ Two common scrambling techniques are B8ZS and HDB3.



Block Coding

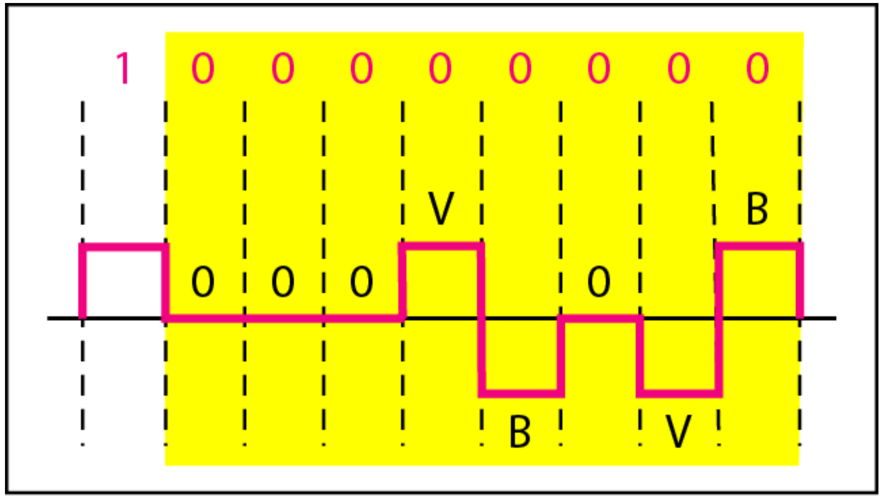
❑**Scrambling(B8ZS)**

✓ It is known as Bipolar with 8 zero substitution.

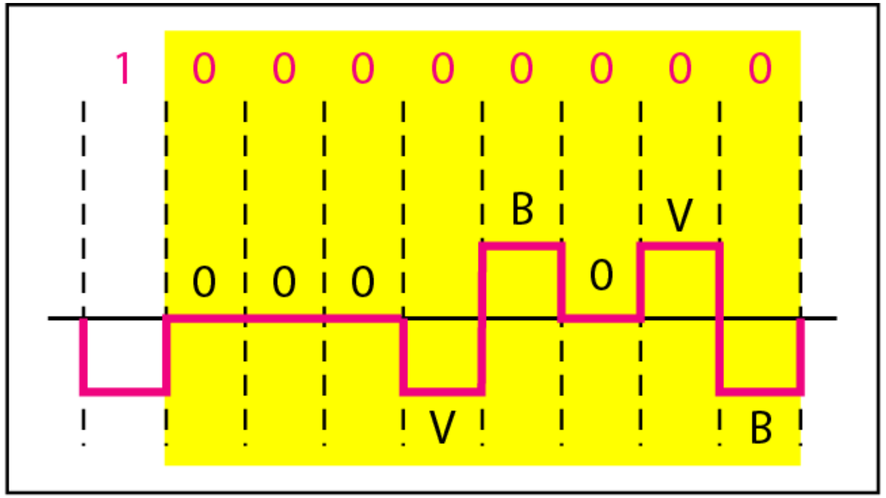
✓ 8 consecutive zeros are substituted by 000VB0VB.

✓ **V (violation)- Same as last non-zero level**

✓ **B(Bipolar)- opposite to last non-zero level**



a. Previous level is positive.



b. Previous level is negative.

Block Coding

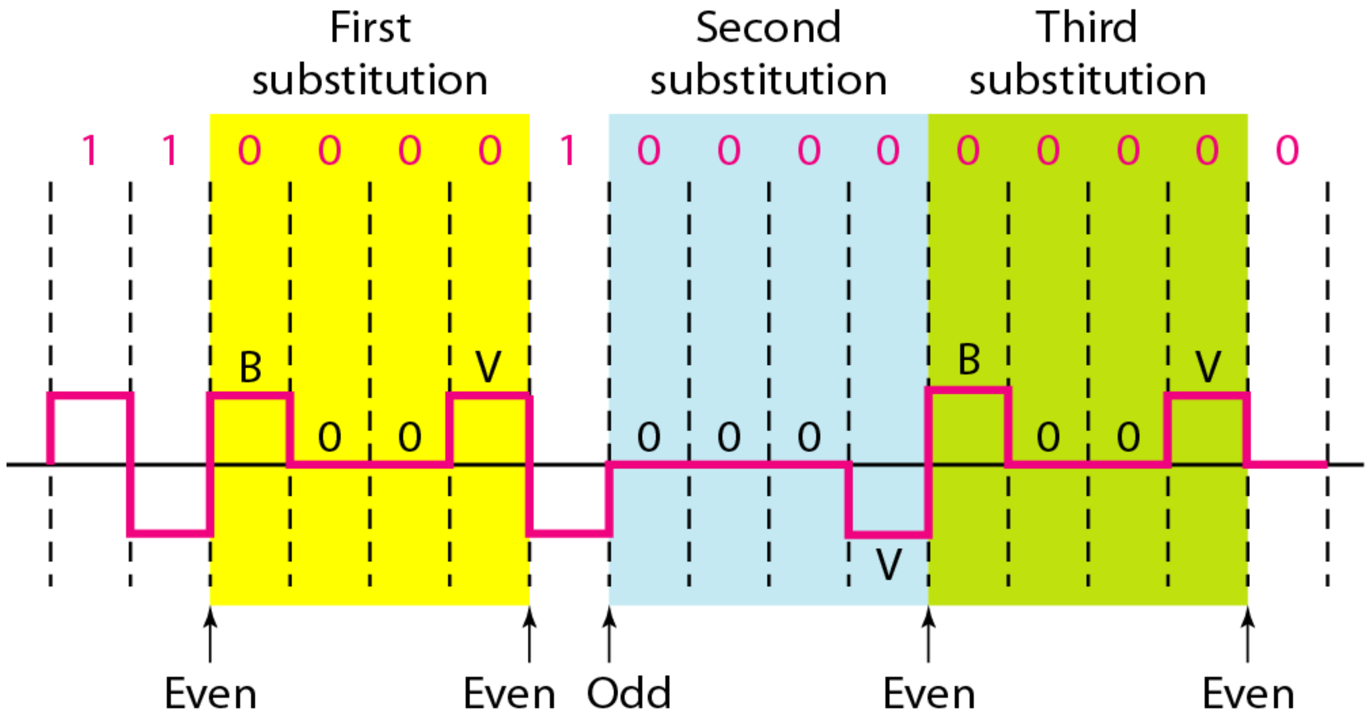
- ❑ **Scrambling(HDB3)**

✓ Four consecutive zero-level voltages are replaced with a sequence of 000V or B00V.

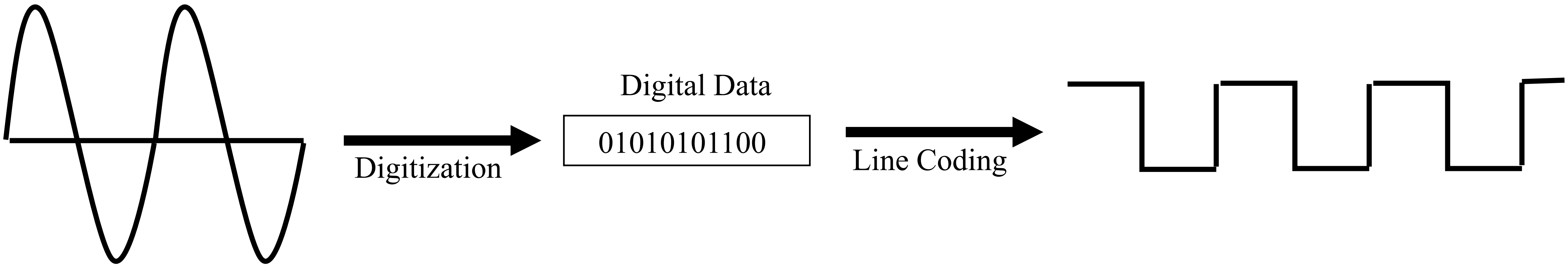
✓ **The two rules can be stated as follows:**

1. If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be 000V, which makes the total number of nonzero pulses even.

2. If the number of nonzero pulses after the last substitution is even, the substitution pattern will be B00V, which makes the total number of nonzero pulses even.



Analog to Digital Signal



Analog to Digital Signal

Pulse Code Modulation (PCM)

Delta Modulation (DM)

Analog to Digital Signal

❑ **Pulse Code Modulation (PCM)**:-

✓ It has three steps.

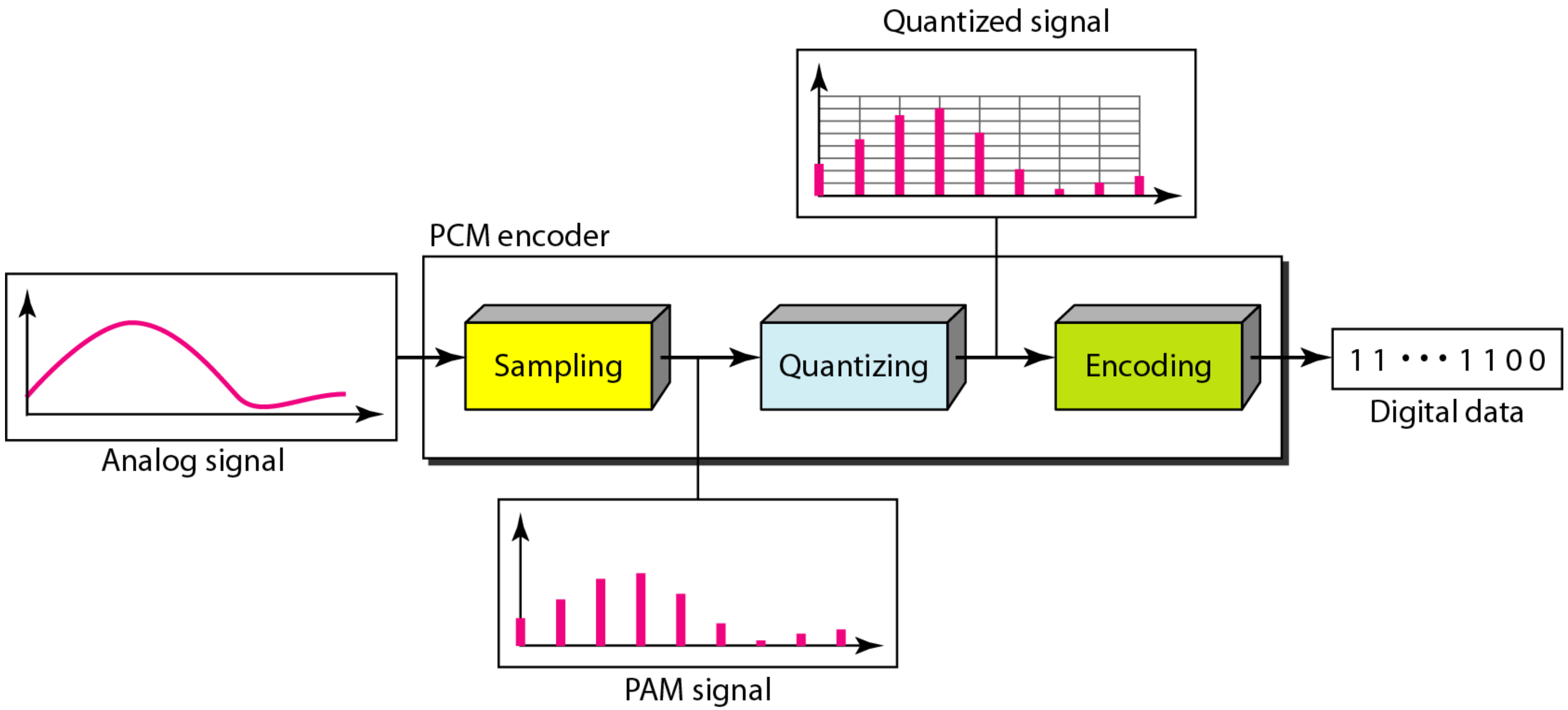
Sampling

Quantizing

Encoding

Analog to Digital Signal

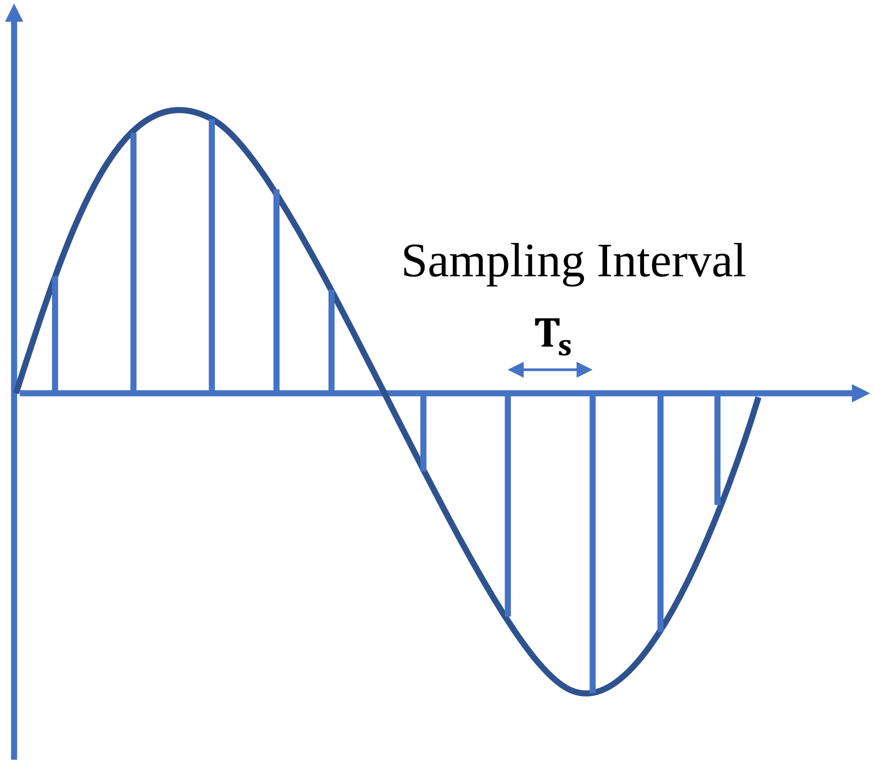
- **Pulse Code Modulation (PCM)**



Analog to Digital Signal

❑**Pulse Code Modulation (PCM)**

✓ **Sampling**



Sampling rate / frequency, $f_s = 1/T_s$

How many samples are taken in 1s.

This process is also referred as Pulse Amplitude Modulation (PAM)

Analog to Digital Signal

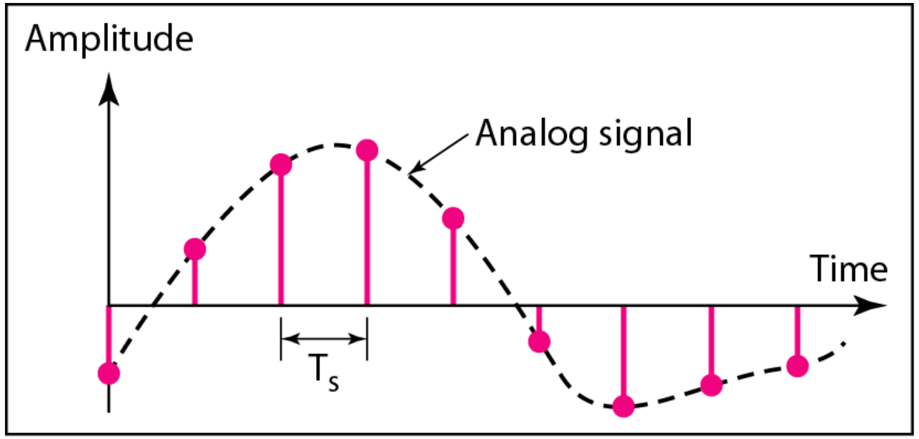
- **Types of Sampling**

✓Natural Sampling

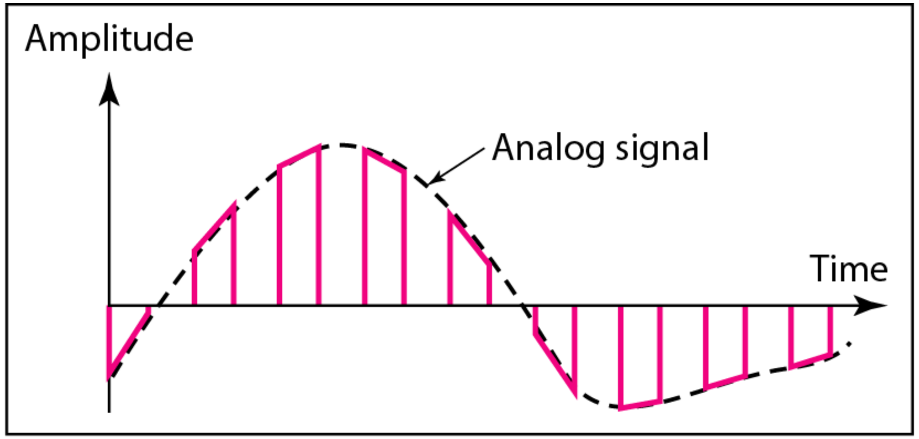
✓Ideal Sampling

✓Flat-Top Sampling

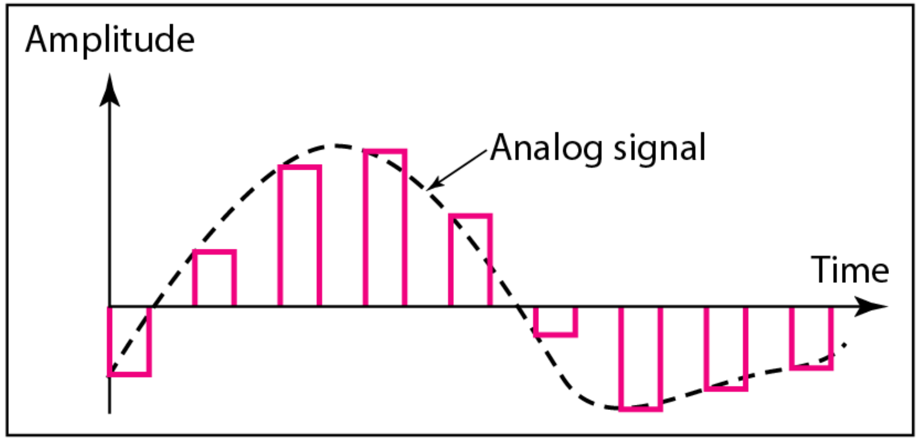
Analog to Digital Signal



a. Ideal sampling



b. Natural sampling

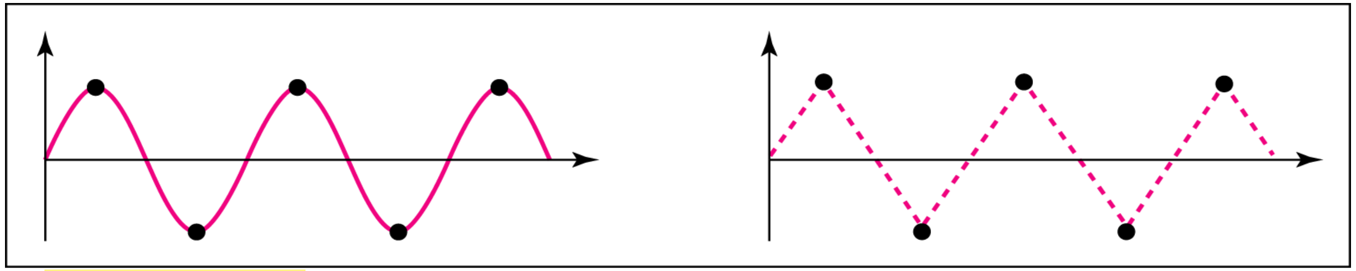


c. Flat-top sampling

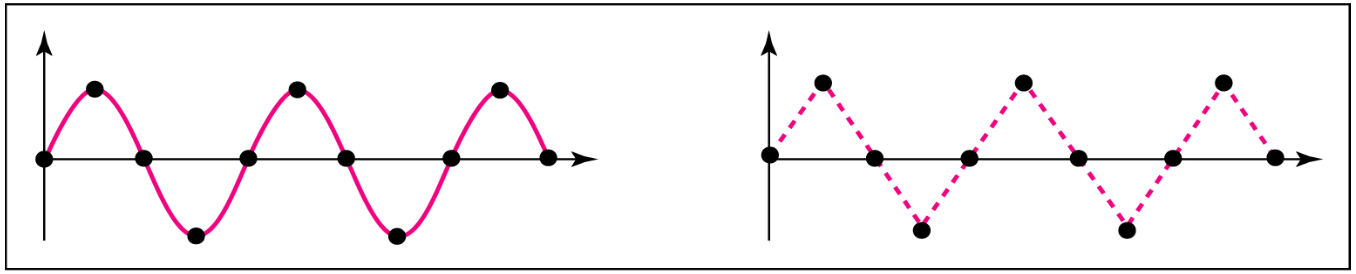
Analog to Digital Signal

❑ Nyquist Sampling Theorem

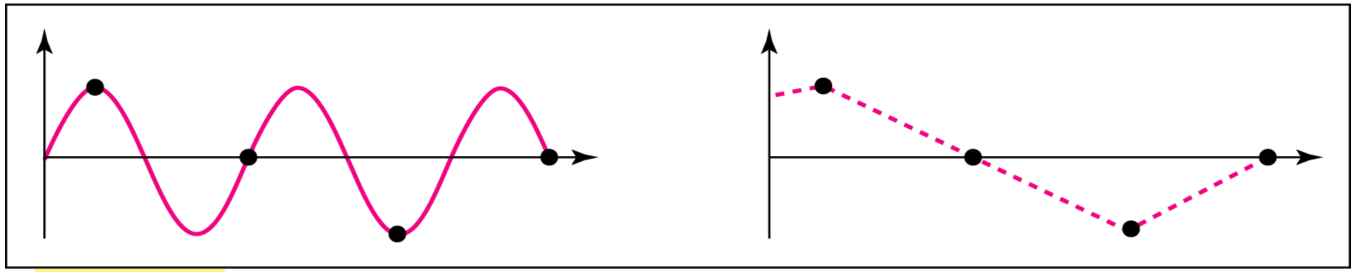
✓ The sampling rate must be at least 2 times the highest frequency contained in the signal.



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



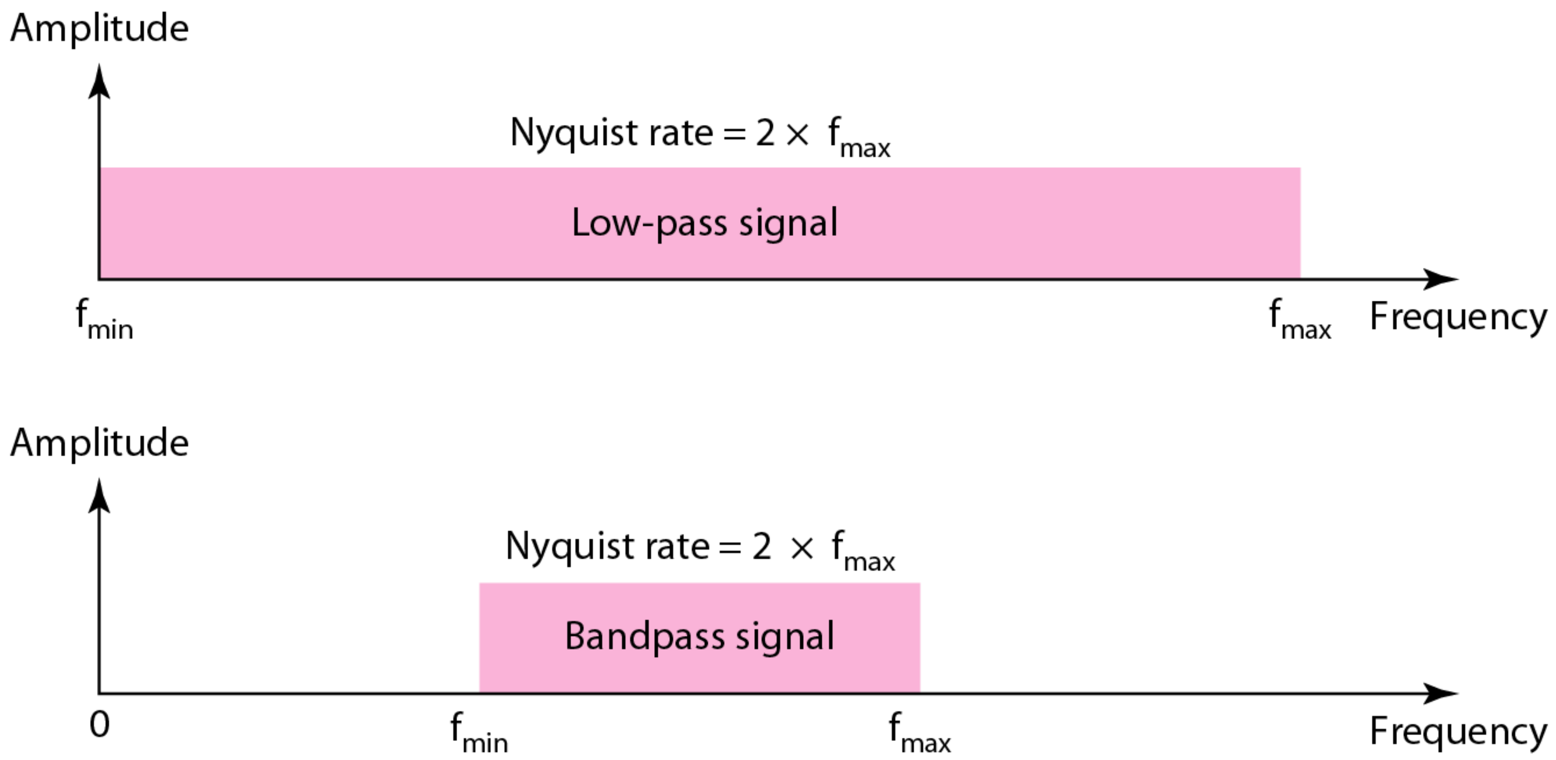
b. Oversampling: $f_s = 4f$



c. Undersampling: $f_s = f$

Analog to Digital Signal

- **Nyquist Sampling Theorem**



Analog to Digital Signal

❑ Nyquist Sampling Theorem

Example: A complex low-pass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

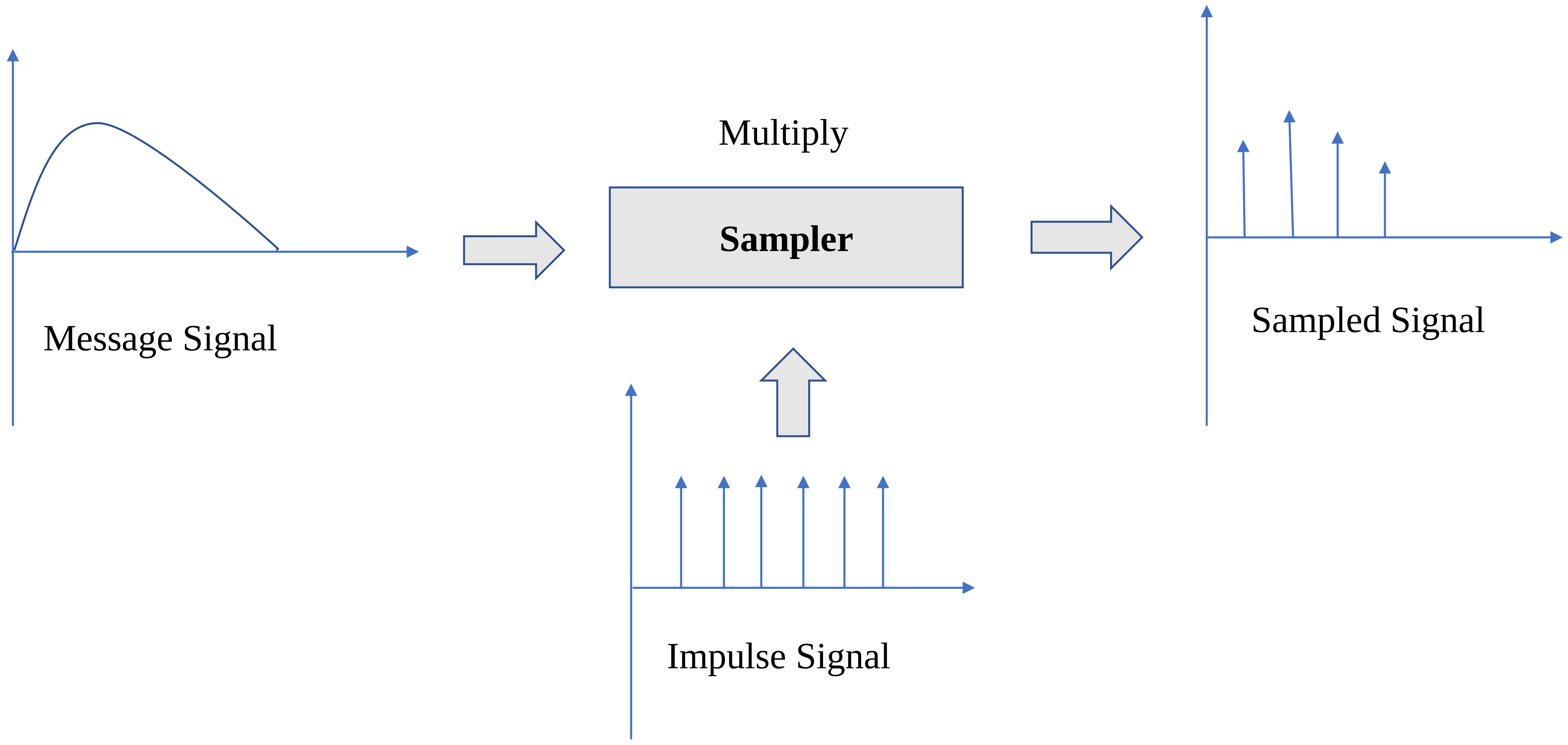
Solution: The bandwidth of a low-pass signal is between 0 and f, where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

Example: 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.

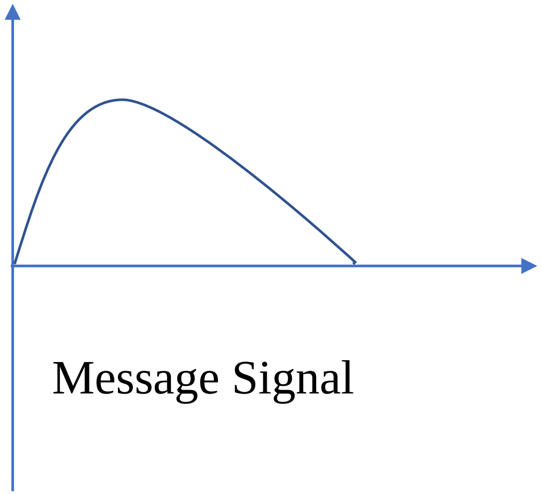
Analog to Digital Signal

- Sampling



Analog to Digital Signal

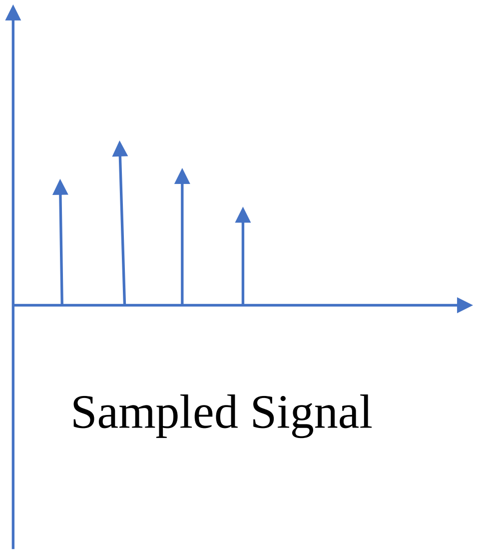
❑ Quantization



Time → Continuous

Amplitude → Continuous

Sampling



Time → Discrete

Amplitude → Continuous

Quantizing

Time → Discrete

Amplitude → Discrete

Analog to Digital Signal

❑**Quantization**

✓ Sampling results in a series of pulses of varying amplitude values ranging between two limits: **maximum amplitude** and **minimum amplitude.**

✓ The amplitude values are infinite between these two limits.

✓ Quantization process maps these infinite amplitude values onto a finite set of known values.

Analog to Digital Signal

❑ Steps of Quantization

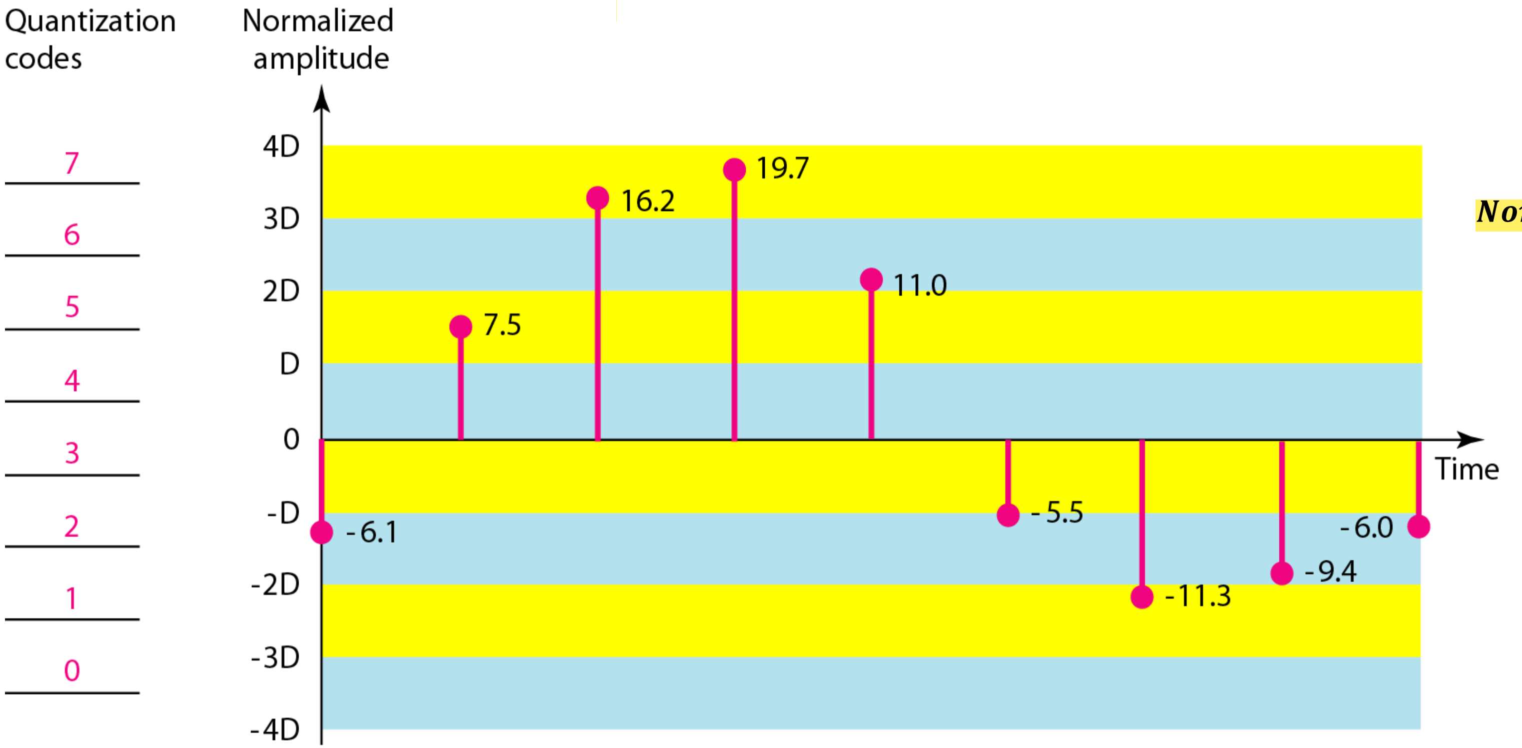
1. The original analog signal has amplitude values between V_{min} and V_{max} and the range is divided into L zones where each of height is delta, $\Delta = \frac{V_{max} - V_{min}}{L}$

2. The midpoint of each zone is assigned a value from 0 to L-1 (resulting in L values).

3. Each sample falling in a zone is then approximated to the value of the midpoint.

Analog to Digital Signal

❑Quantization



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

$$\textit{Normalized PAM value} = \frac{\textit{Actual Amplitude}}{\Delta}$$

$$\textit{Normalized Quantized value} = \frac{\textit{Quantization Value}}{\Delta}$$

$$\textit{Normalized error} = \textit{N.Quantized} - \textit{N.PAM}$$

Analog to Digital Signal

❑Quantization Error

✓ Quantization error is the difference between the analog signal and the closest available digital value at each sampling instant.

✓ Quantization error also introduces noise, called quantization noise, to the sample signal which in turn changes the signal to noise ratio.

$$SNR_{DB} = 6.02n_b + 1.76 \text{ DB}$$

Analog to Digital Signal

❑Quantization Levels

The quantization level mainly depends on-

The ranges of amplitudes of the analog signal

How accurate we need to recover the signal

Analog to Digital Signal

- **Transmission Bandwidth**

Bit rate = $n_b (no\ of\ bits\ per\ sample) * f_s\ (sampling\ rate)$

Bandwidth = c * N * 1/r

= c * n_b * f_s $*$ 1/r

If r =1 and c=1/2

Minimum Bandwidth = ½* n_b * f_s

= ½ * n_b * 2 * f_{max}

= n_b * f_{max}

If the analog signal is low pass signal then we Bandwidth = $f_{max} - f_{min}$

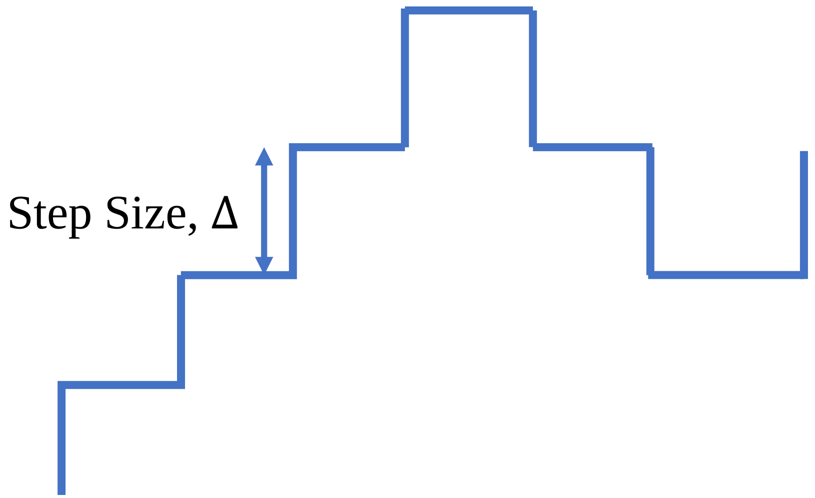
Here, f_{min}= 0, so bandwidth = f_{max}

Minimum Bandwidth = n_b * B_{analog}

Analog to Digital Signal

❑ **Delta Modulation (DM)**

✓ It builds a second signal that resembles as staircase signal.

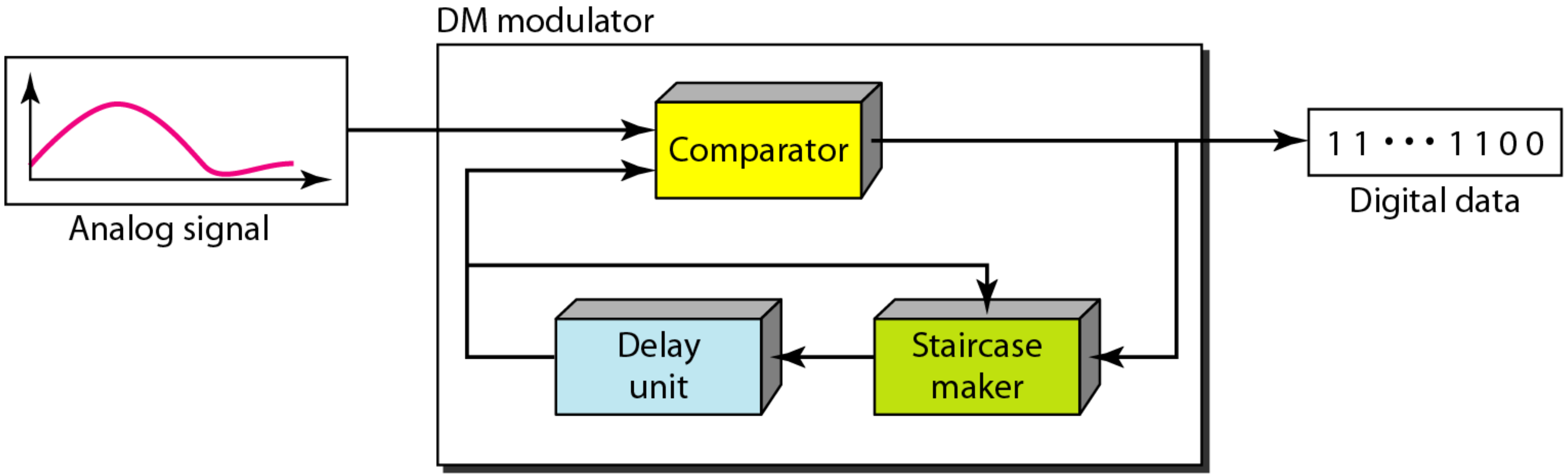


Analog to Digital Signal

✓At each sampling interval, compares the value of the analog signal with the last value of the gradually made staircase signal.

✓ If analog signal > staircase signal → + Δ (+1)

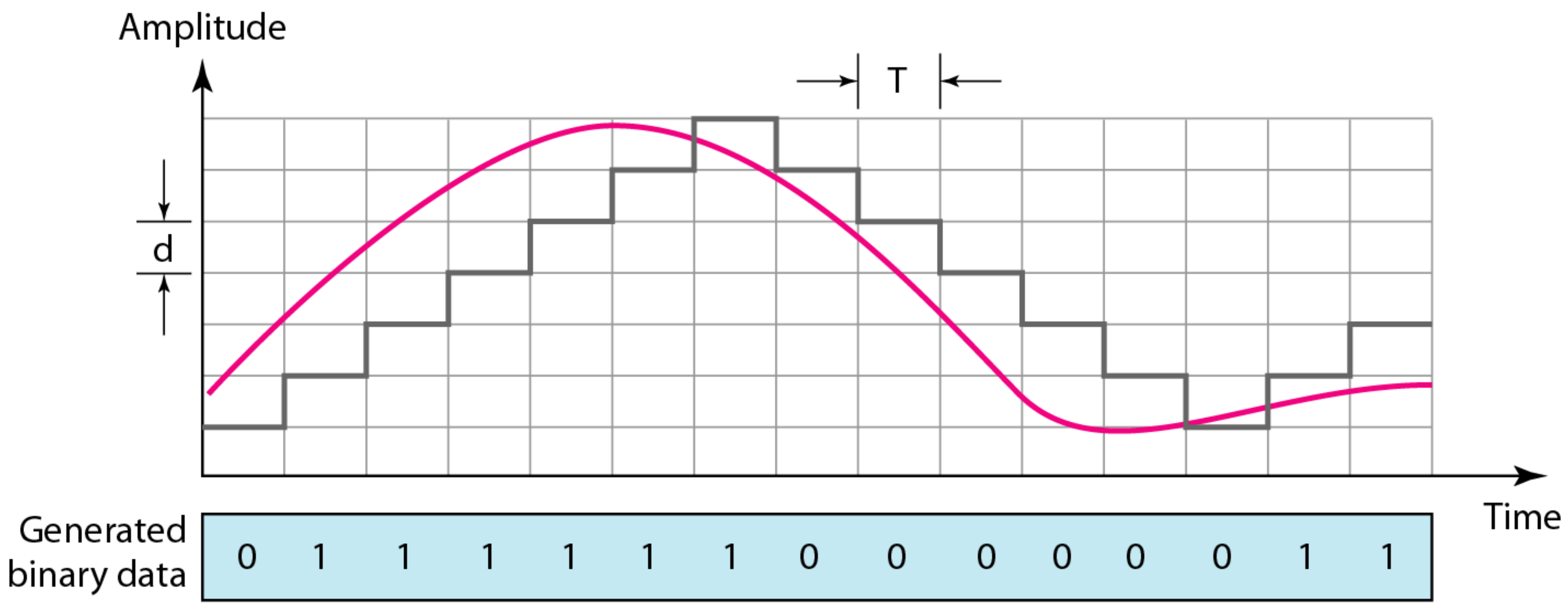
✓ If analog signal < staircase signal→ - Δ (-0)



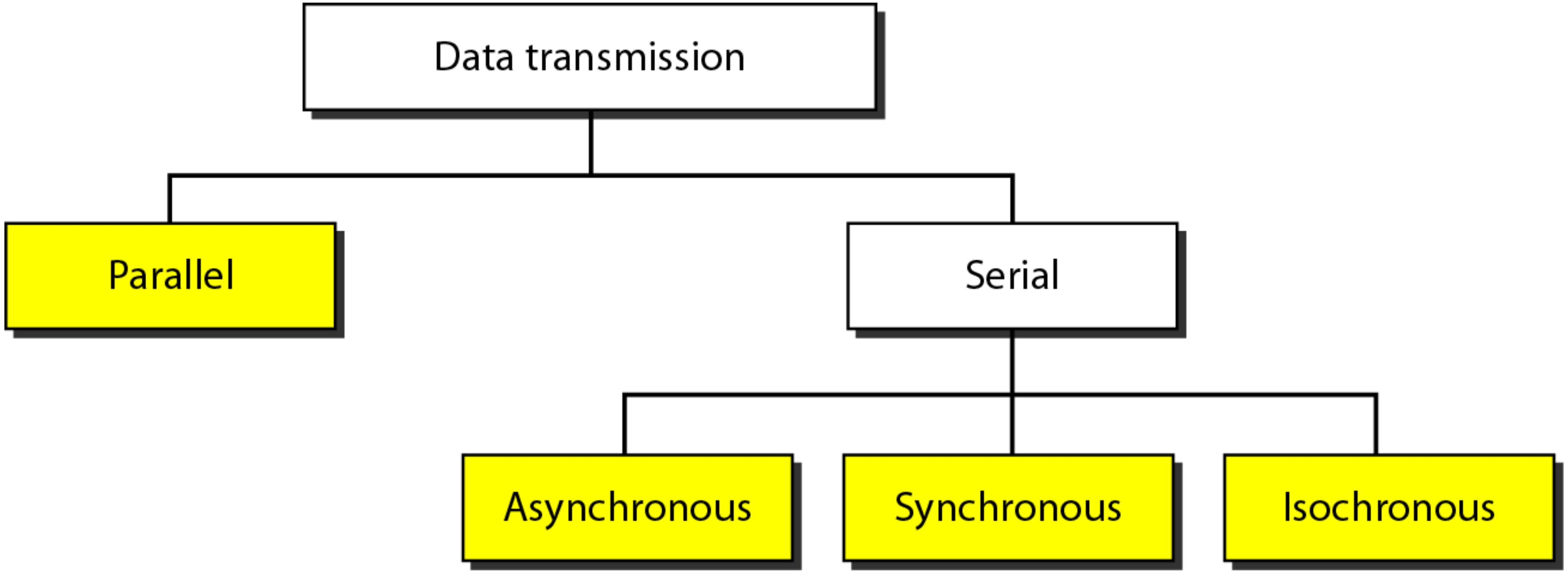
Analog to Digital Signal

✓If analog signal > staircase signal → + Δ (+1)

✓ If analog signal < staircase signal→ - Δ (-0)



Transmission Modes (Digital Transmission)



Transmission Modes (Digital Transmission)

- **Parallel Transmission**

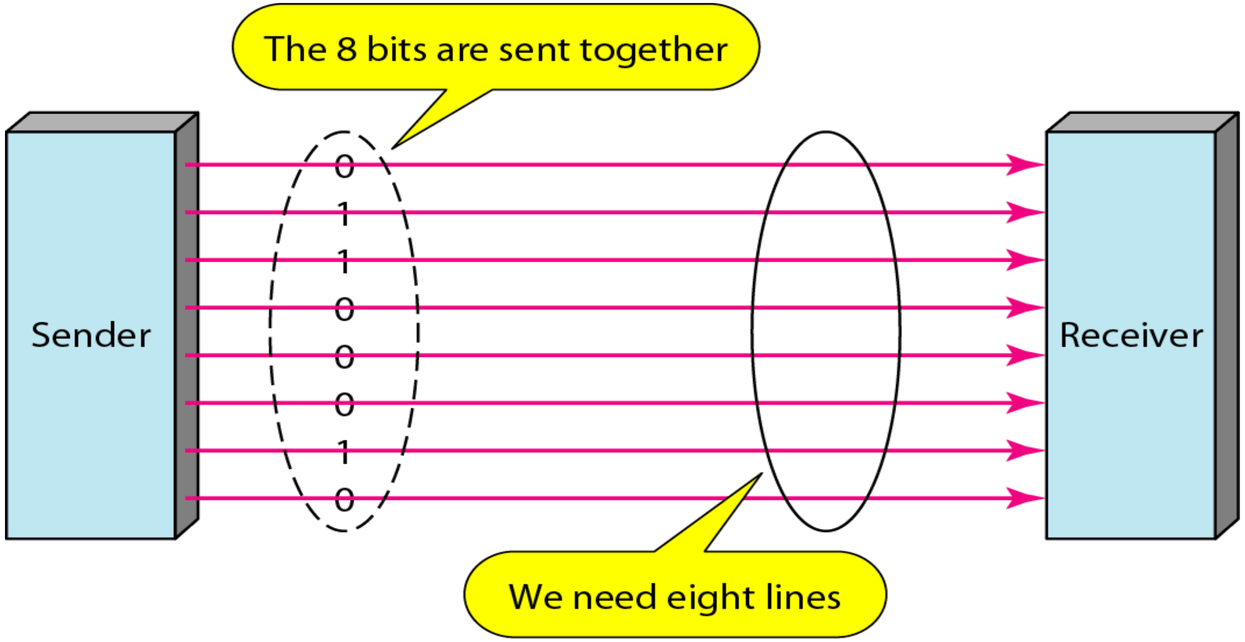
✓ Use *n* wires to send *n* bits at one time.

- **Advantages**

✓ Speed is high, all else being equal, parallel transmission can increase the transfer speed by a factor of *n* over serial transmission.

- **Disadvantages**

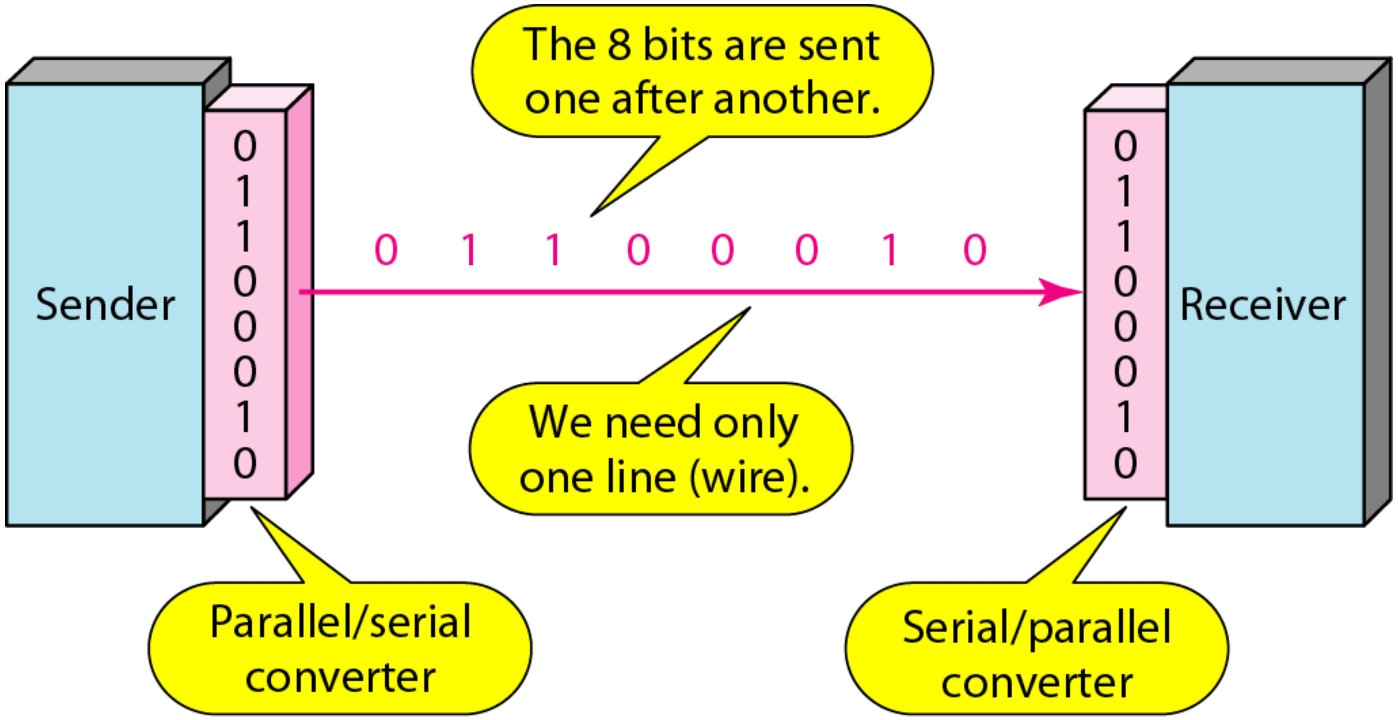
✓ Parallel transmission requires *n* communication lines. Because this is expensive, parallel transmission is usually limited to short distances.



Transmission Modes (Digital Transmission)

❑ **Serial Transmission**

✓ In serial transmission one bit follows another, so we need only one communication channel rather than *n* to transmit data between two communicating devices.



Transmission Modes (Digital Transmission)

- ❑ **Advantages of Serial Transmission**

- ✓ The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of *n*.

- ❑ **Disadvantages of Serial Transmission**

- ✓ Speed comparatively slower than parallel transmission.

Transmission Modes (Digital Transmission)

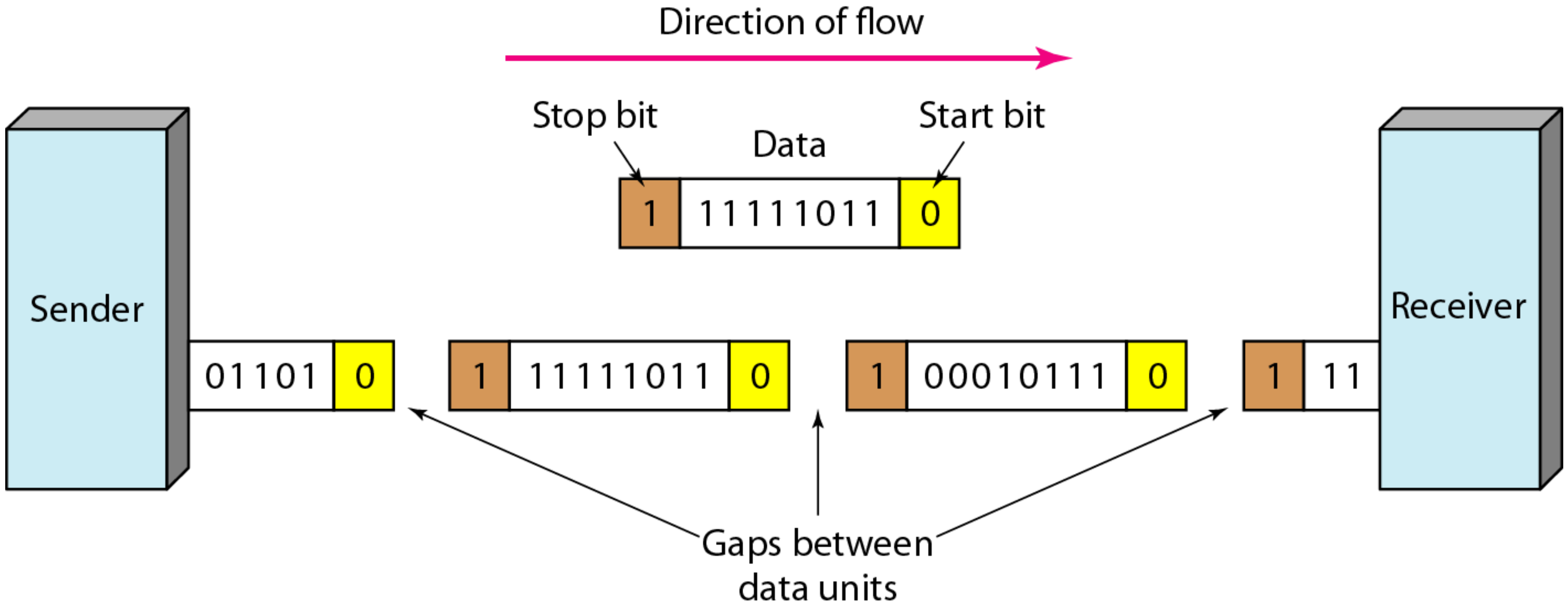
❑ **Asynchronous Transmission**

✓ The timing of a signal is unimportant.

✓ Instead, information is received and translated by agreed upon patterns. As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent.

✓ In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (Is) at the end of each byte. There may be a gap between each byte.

✓ The start and stop bits and the gap alert the receiver to the beginning and end of each byte and allow it to synchronize with the data stream.



Transmission Modes (Digital Transmission)

- ❑ **Asynchronous Transmission**
- ❑ **Advantages**
- ✓ But it is cheap and effective, two advantages that make it an attractive choice for situations such as low-speed communication.
- ❑ **Uses**
- ✓ For example, the connection of a keyboard to a computer is a natural application for asynchronous transmission.

Transmission Modes (Digital Transmission)

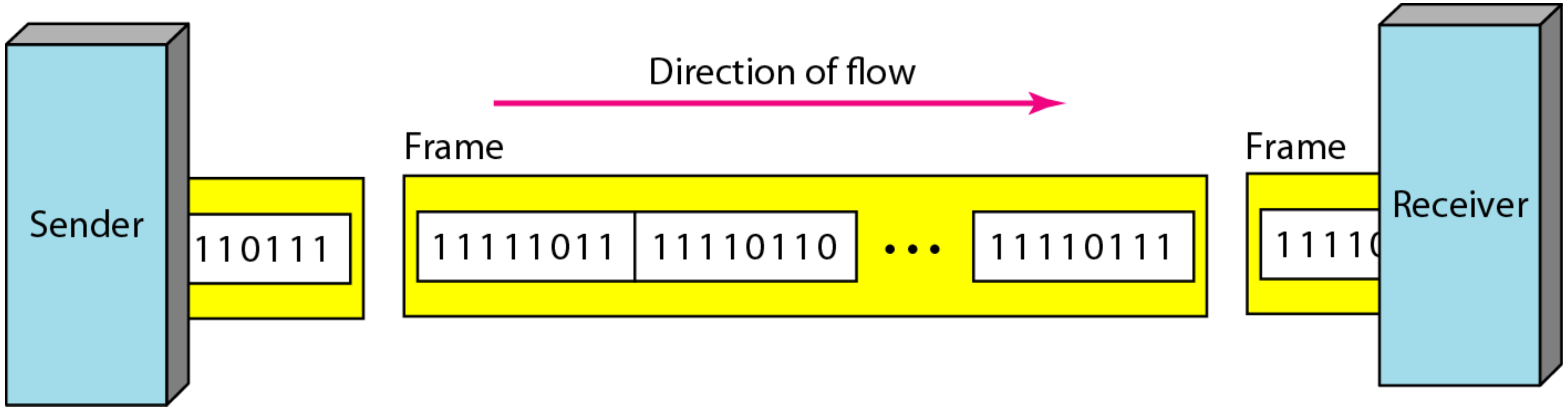
❑ **Synchronous Transmission**

✓ **In** synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

✓ Timing becomes very important, therefore, because the accuracy of the received information is completely dependent on the ability of the receiving device.

✓ The advantage of synchronous transmission is speed. With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link, synchronous transmission is faster than asynchronous transmission.

✓ For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another. Byte synchronization is accomplished in the data link layer.



Transmission Modes (Digital Transmission)

- **Isochronous Transmission**

- ✓ In real-time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails.
- ✓ For example, TV images are broadcast at the rate of 30 images per second; they must be viewed at the same rate.
- ✓ If each image is sent by using one or more frames, there should be no delays between frames.
- ✓ For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized. The isochronous transmission guarantees that the data arrive at a fixed rate.