

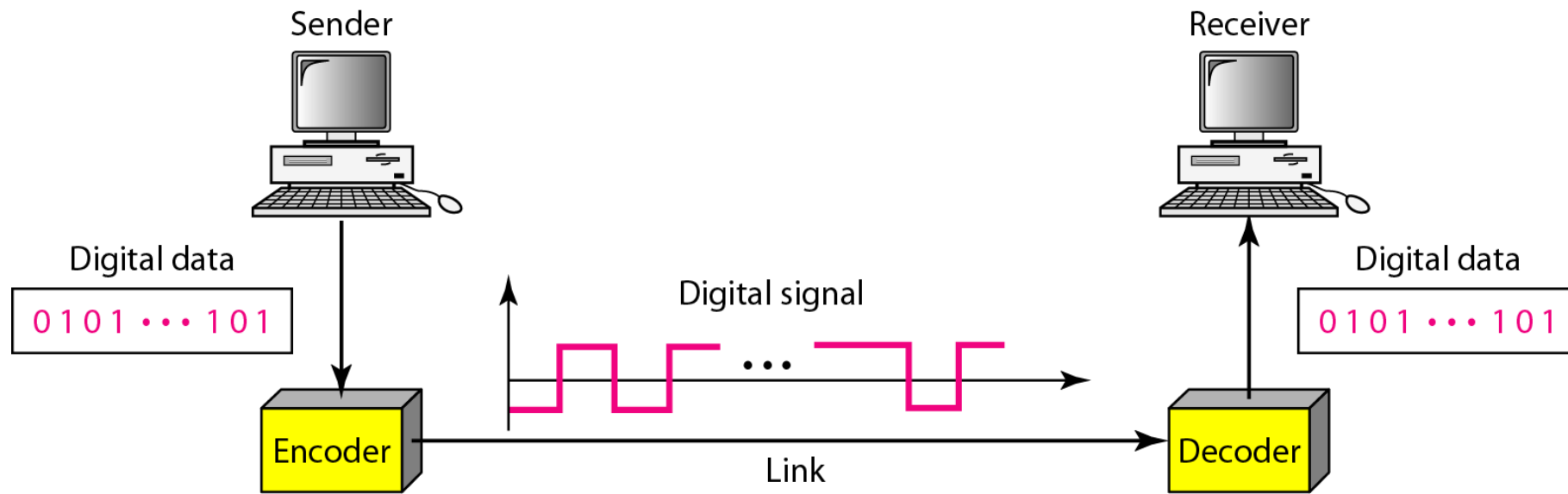
Chapter-4

Digital Transmission

Digital Transmission

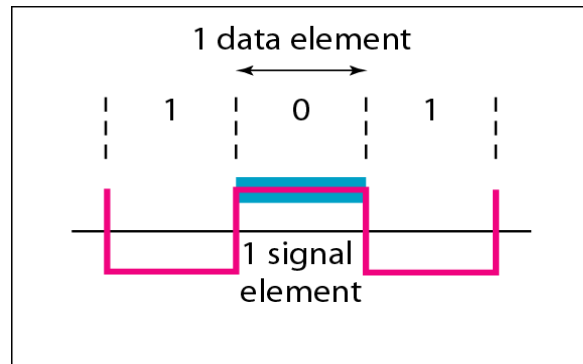
Digital transmission mainly two types:

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

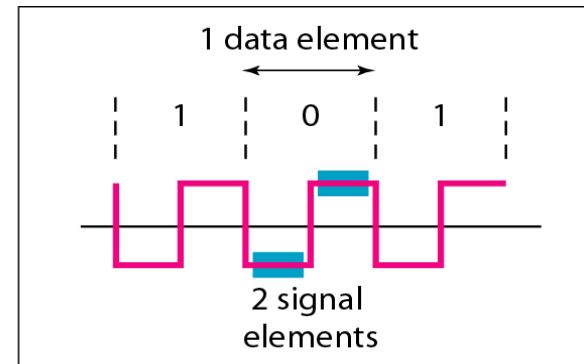


Data Element vs Signal Element

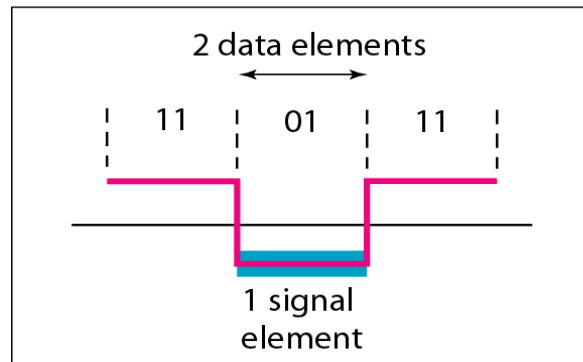
- ✓ Data elements are what we need to send; signal elements are what we can send. Data elements are being carried; signal elements are the carriers.
- ✓ Ratio, $r = \text{Data element} / \text{Signal element}$



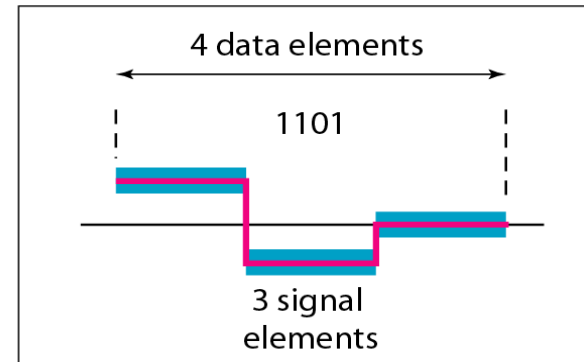
a. One data element per one signal element ($r = 1$)



b. One data element per two signal elements ($r = \frac{1}{2}$)



c. Two data elements per one signal element ($r = 2$)



d. Four data elements per three signal elements ($r = \frac{4}{3}$)

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:**

$$S = c * N * 1/r \text{ baud}$$

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

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

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.

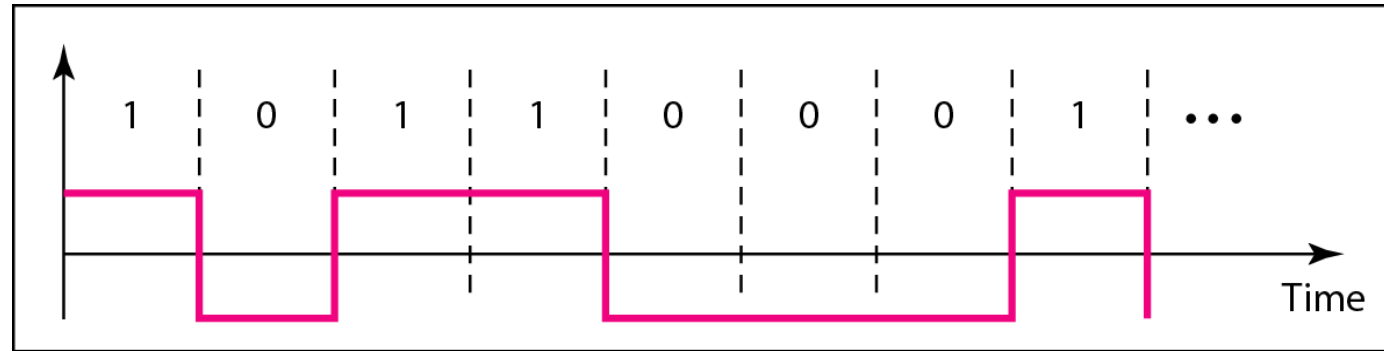
Basic Terminologies for Line Coding Scheme

❑ Self Synchronization

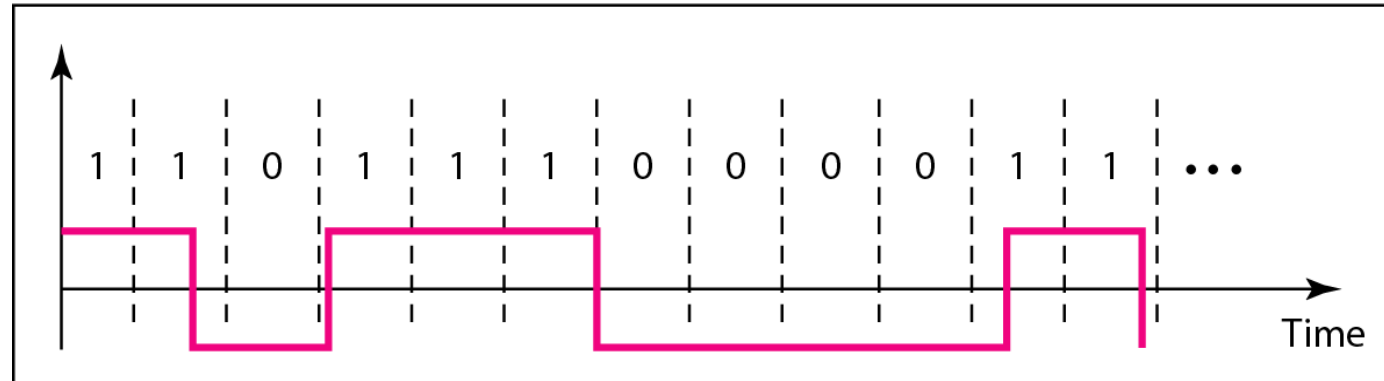
- ✓ To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals.
- ✓ If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals.
- ✓ A self-synchronizing 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 then extra bit will be added to the receiver. If receiver is slower than the sender then 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

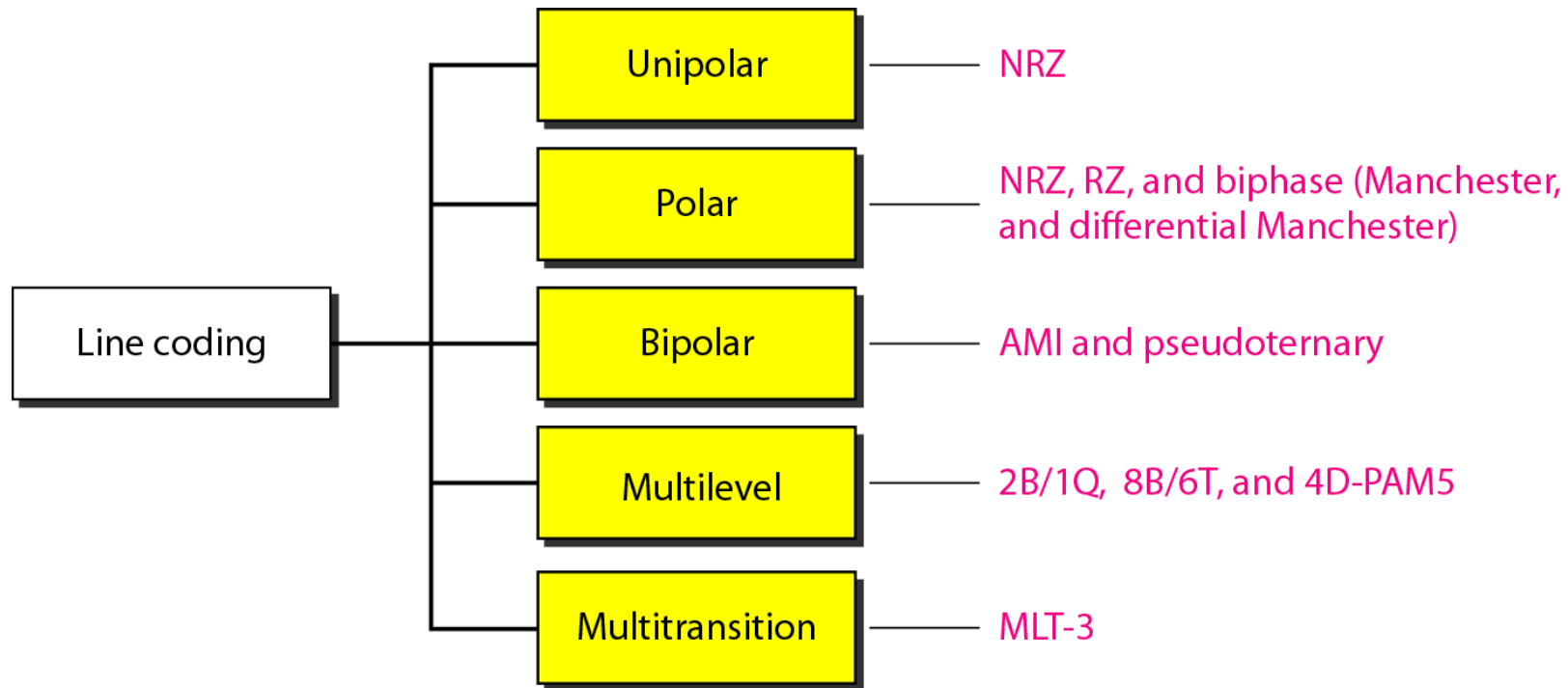
- ✓ Digital to digital conversion is used to represent **digital data in digital signal**.
- ✓ The conversion involves three techniques: **line coding, block coding, and scrambling**.

□ 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

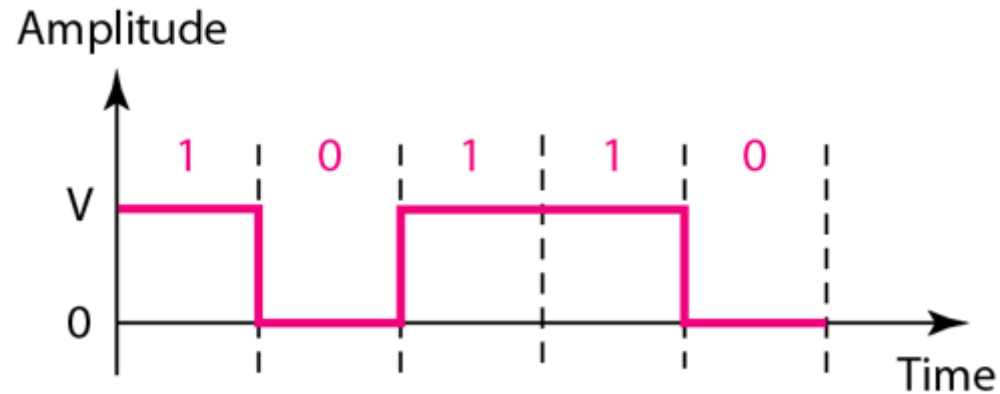
Line coding is divided into five categories:-



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.

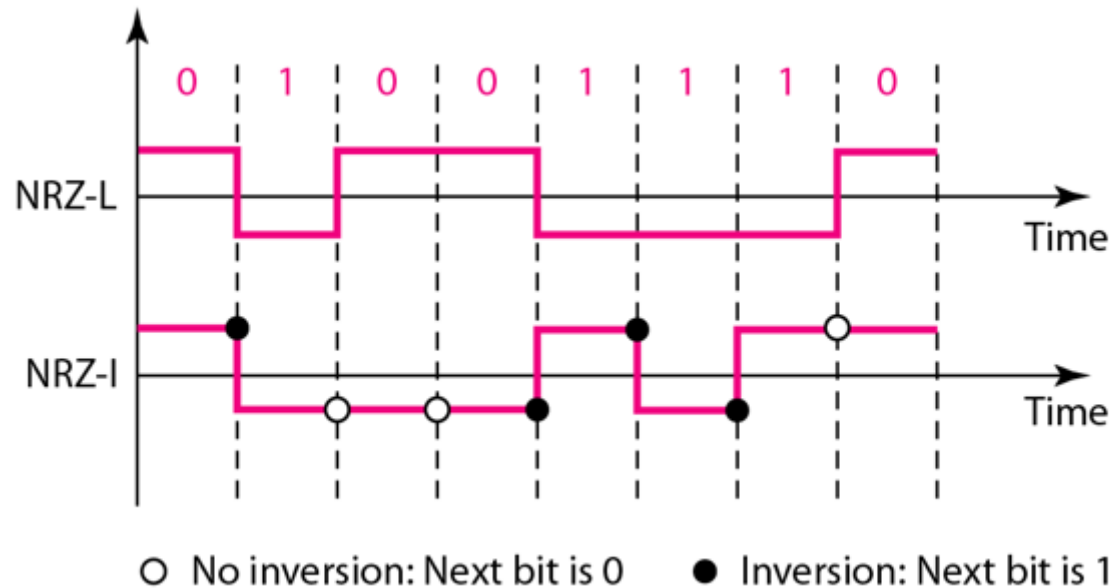
Line Coding

❑ 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:** $S_{average} = c * N * \frac{1}{r}$

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

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

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

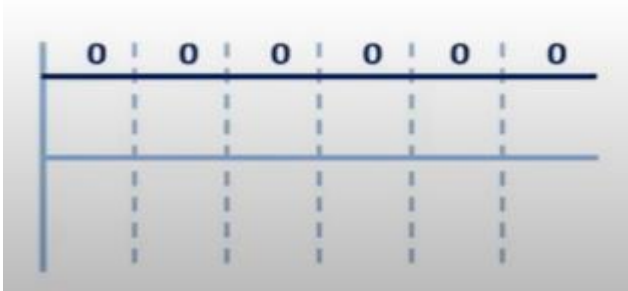
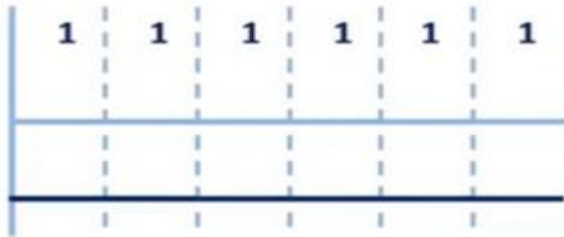
✓ **Solution:** The average signal rate is $S = c \times N \times R = 1/2 \times N \times 1 = 500 \text{ kbaud}$. The minimum bandwidth for this average baud rate is $B(\text{min}) = S = 500 \text{ kHz}$.

Line Coding

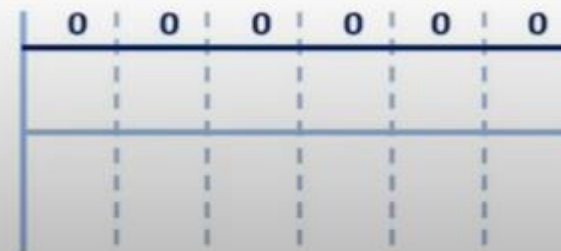
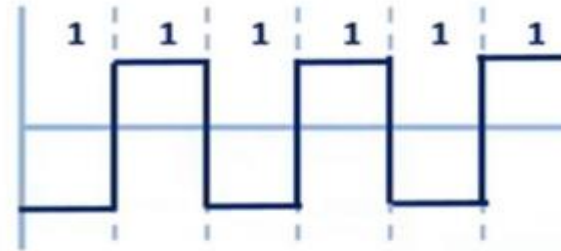
❑ Disadvantages of Polar (NRZ-L & NRZ-I):

- ✓ Baseline wandering problems for long 0's and long 1's (NRZ-L).
- ✓ Baseline wandering problems for long sequence of 0's (NRZ-I).

NRZ - L



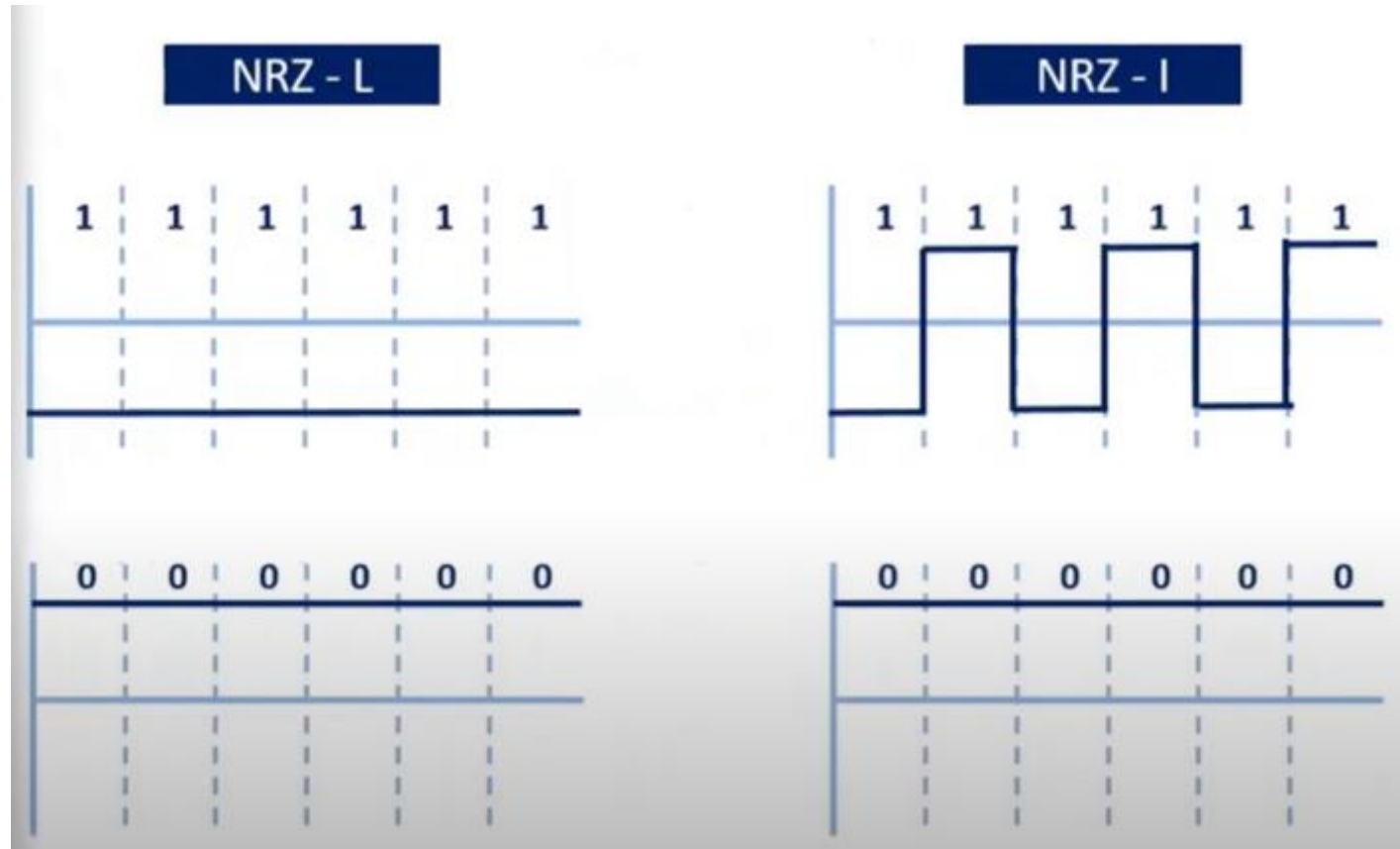
NRZ - I



Line Coding

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

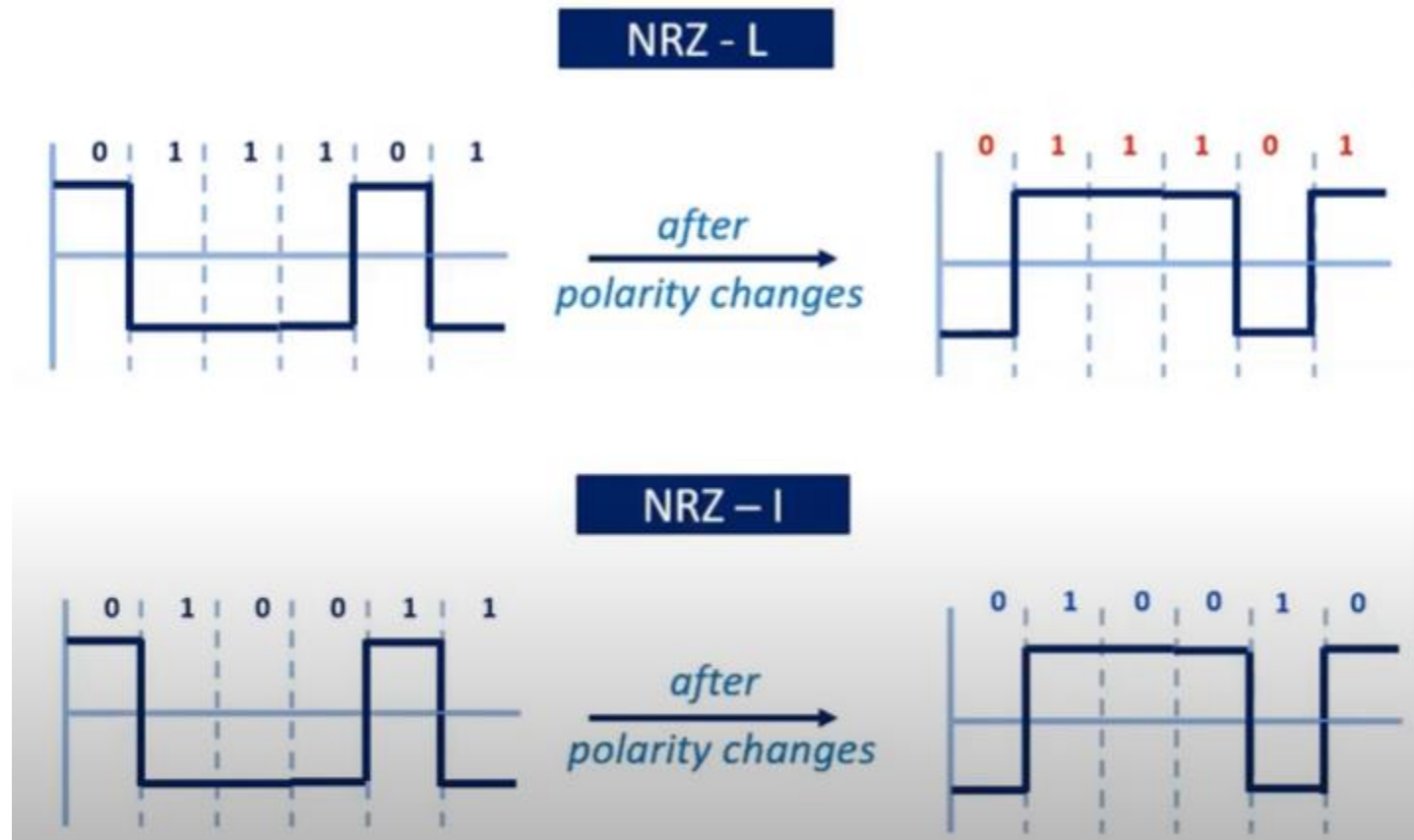
- ✓ Synchronization problem also exists in both NRZ-L and NRZ-I and like before the problem is more severe in NRZ-L.



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)

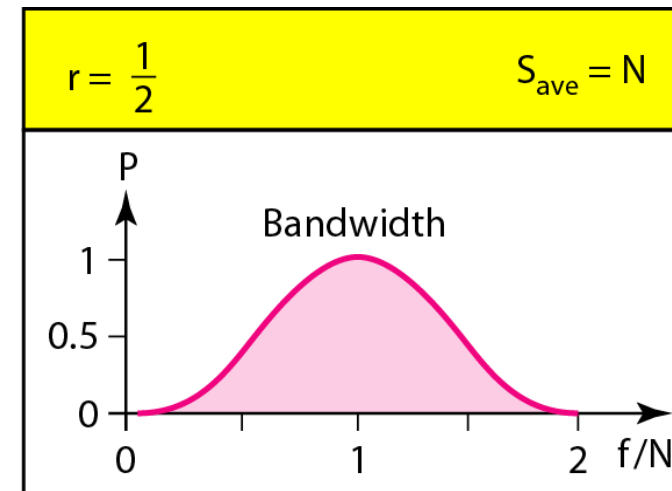
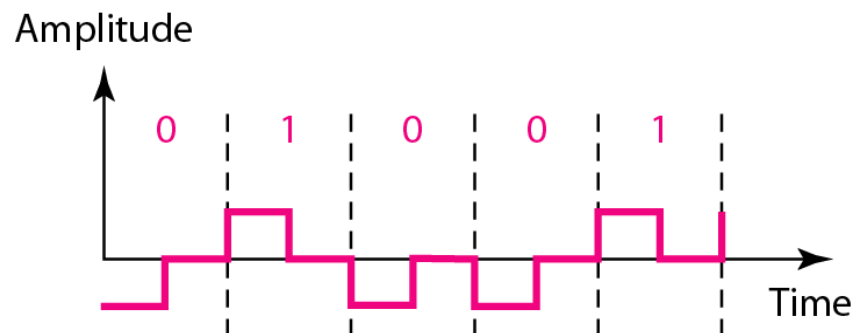
- ✓ It uses three voltage levels $+V$, 0 and $-V$.
- ✓ Negative voltage represents '0' and positive voltage represents '1'.
- ✓ Always goes to zero at the middle of the bit and stays there until the start of the next bit.

❑ Advantages

- ✓ There is no DC component problem.
- ✓ Synchronization is possible.

❑ Disadvantages

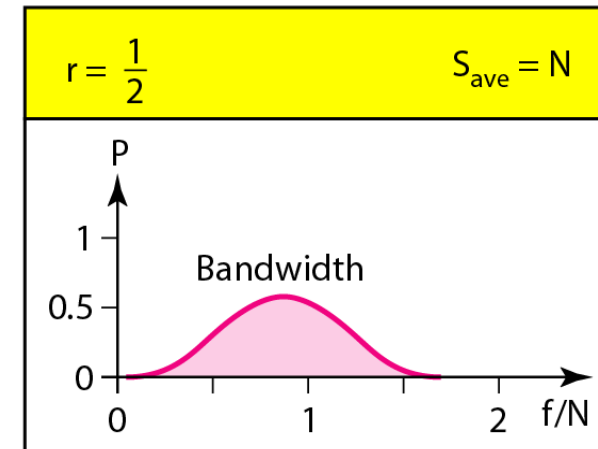
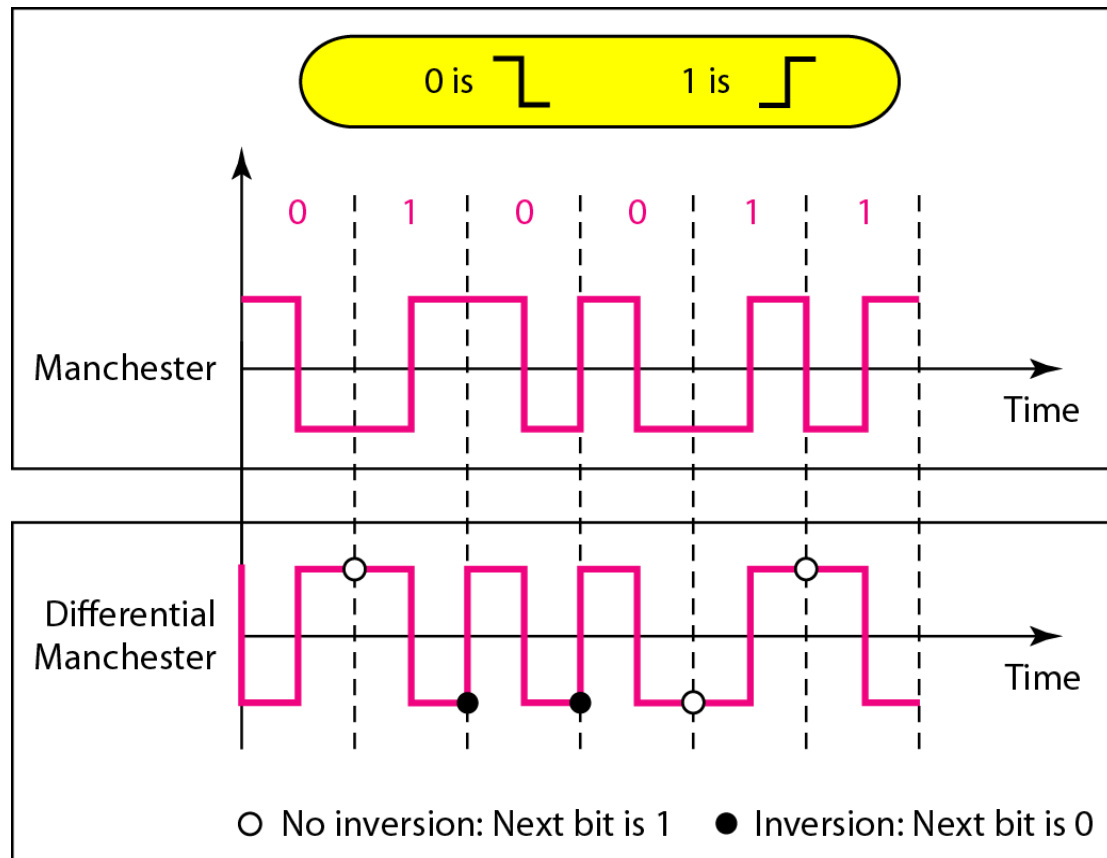
- ✓ It uses three level of voltage and complex.
- ✓ Signal rate is high.



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

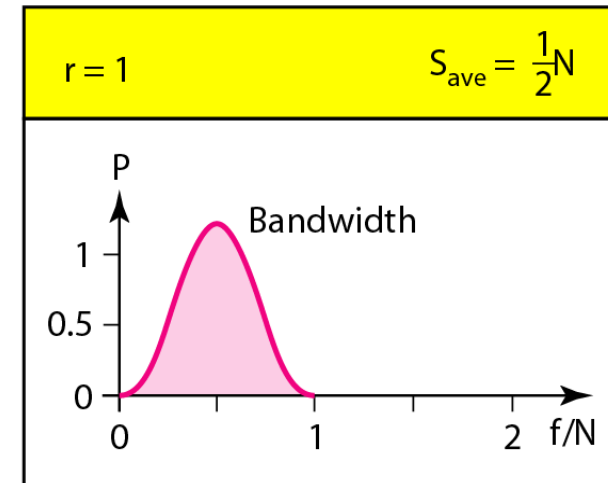
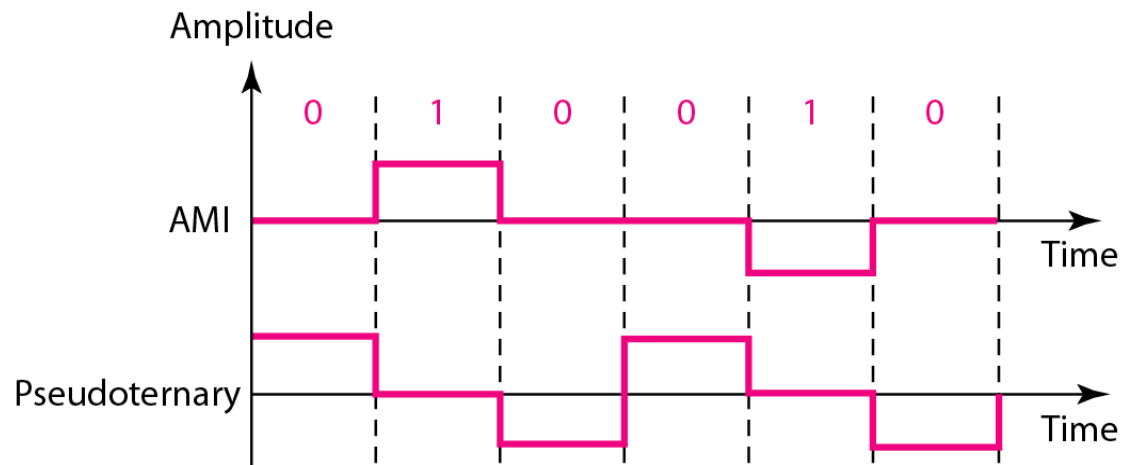
❑ In bipolar encoding (*multilevel binary*), there are three voltage levels: positive, negative, and zero.

❑ 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:

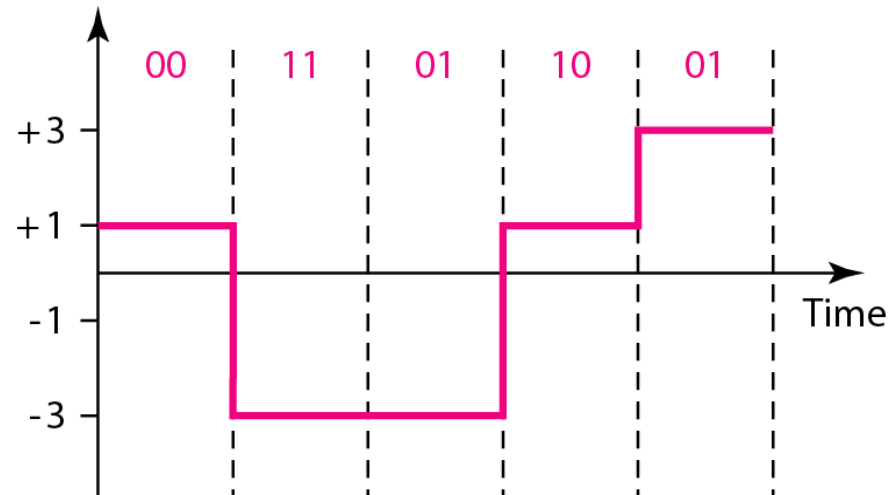
✓ 2B1Q is used in DSL (Digital Subscriber Line) technology to provide a high-speed connection to the Internet by using subscriber telephone lines.

Line Coding

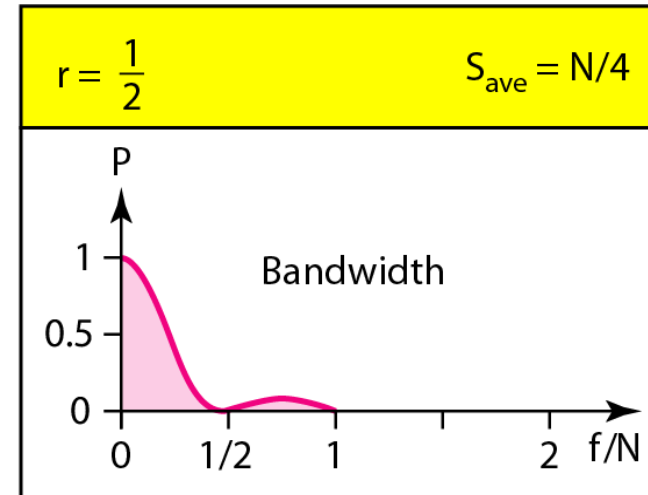
□2B1Q

	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



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 $2^8 = 256$ different data patterns and $3^6 = 478$ 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

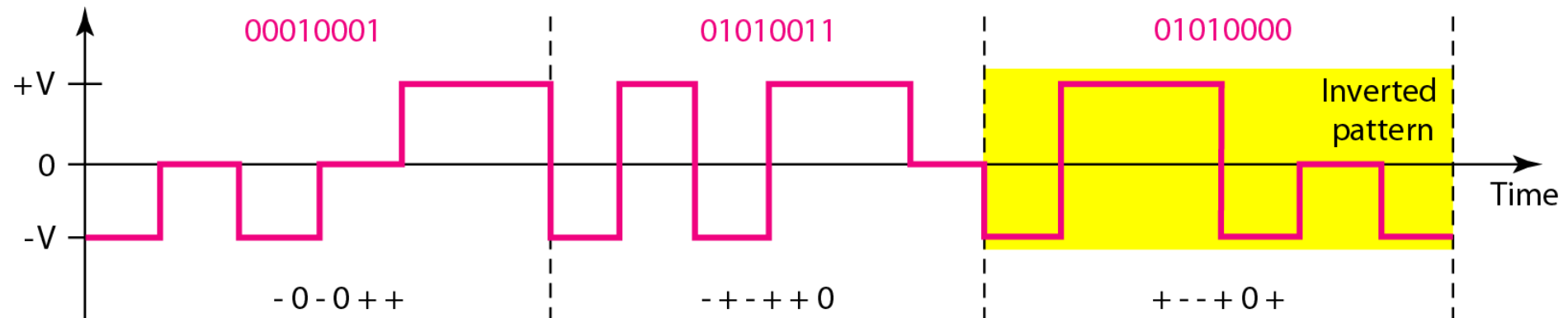
□8B6T

- ✓ 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

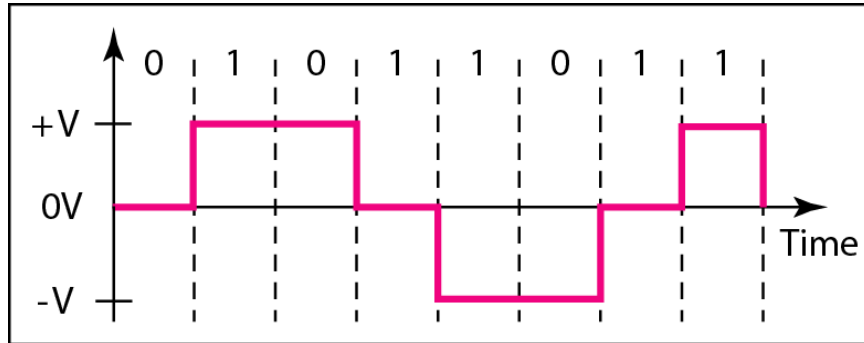
❑ Multiline Transmission (MLT-3)

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

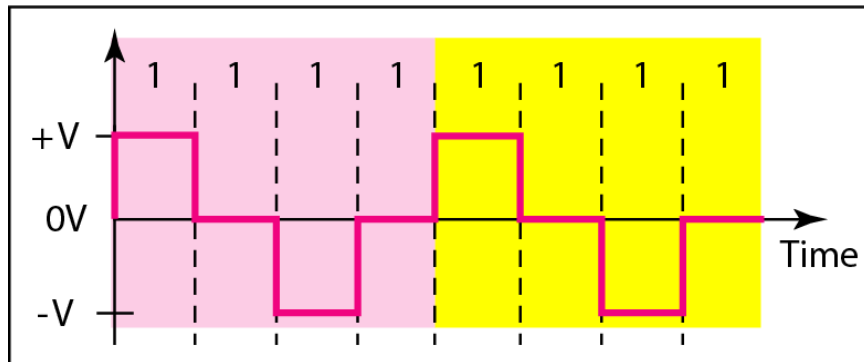
1. If the next bit is 0, there is no transition.
2. If the next bit is 1 and the current level is not 0, the next level is 0.
3. If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level.

Line Coding

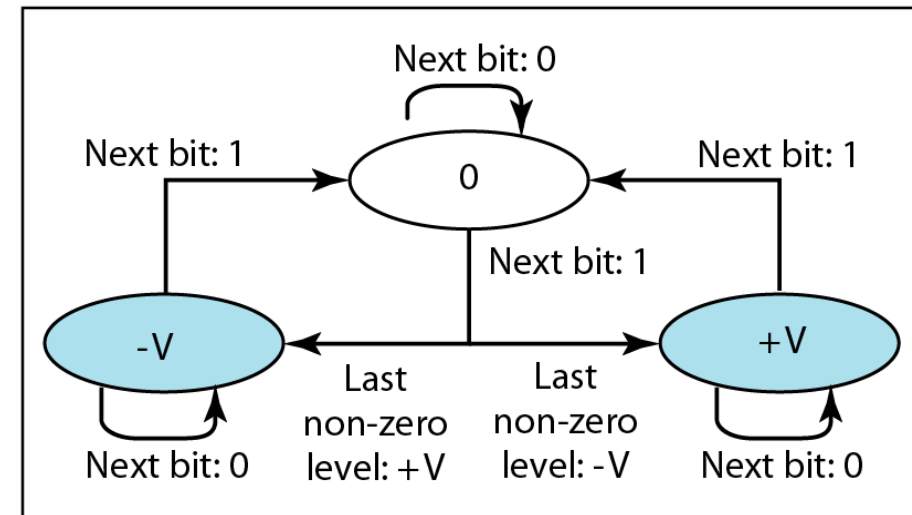
□ Multiline Transmission (MLT-3)



a. Typical case



b. Worse case

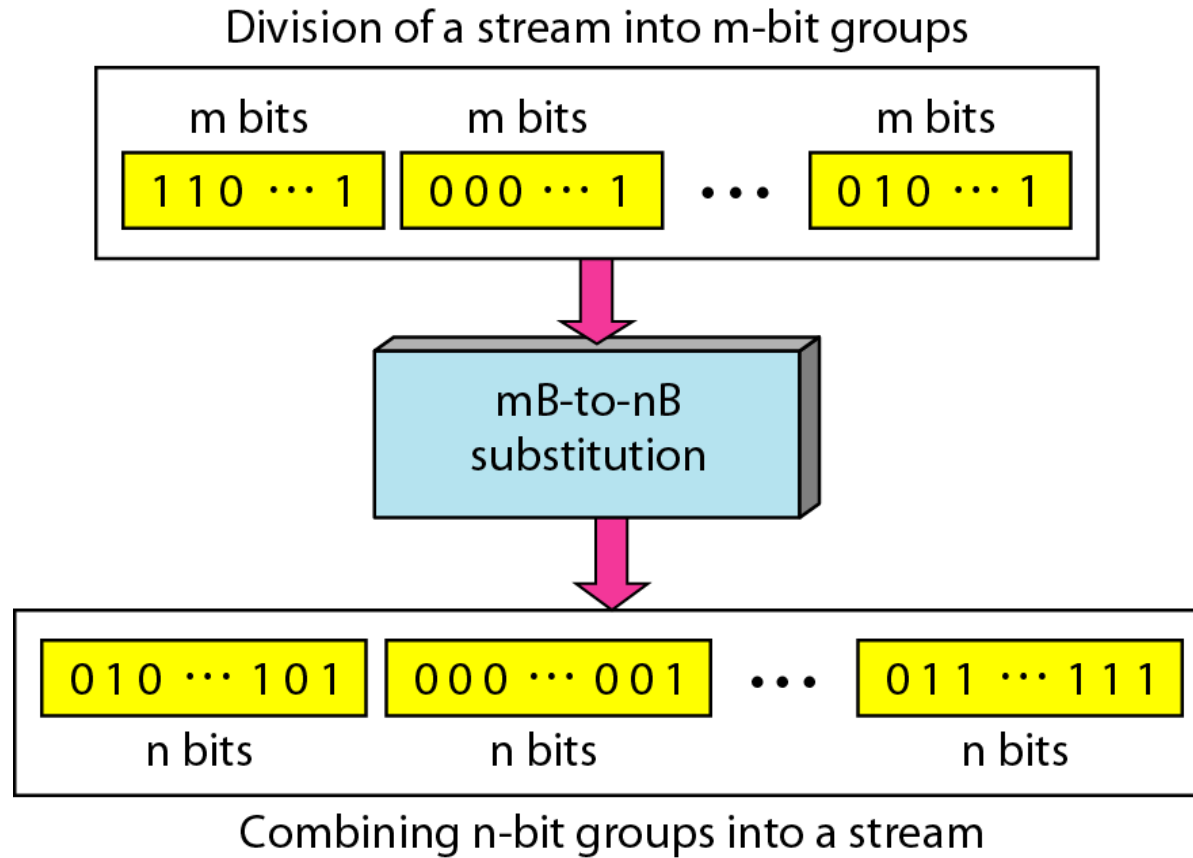


c. Transition states

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



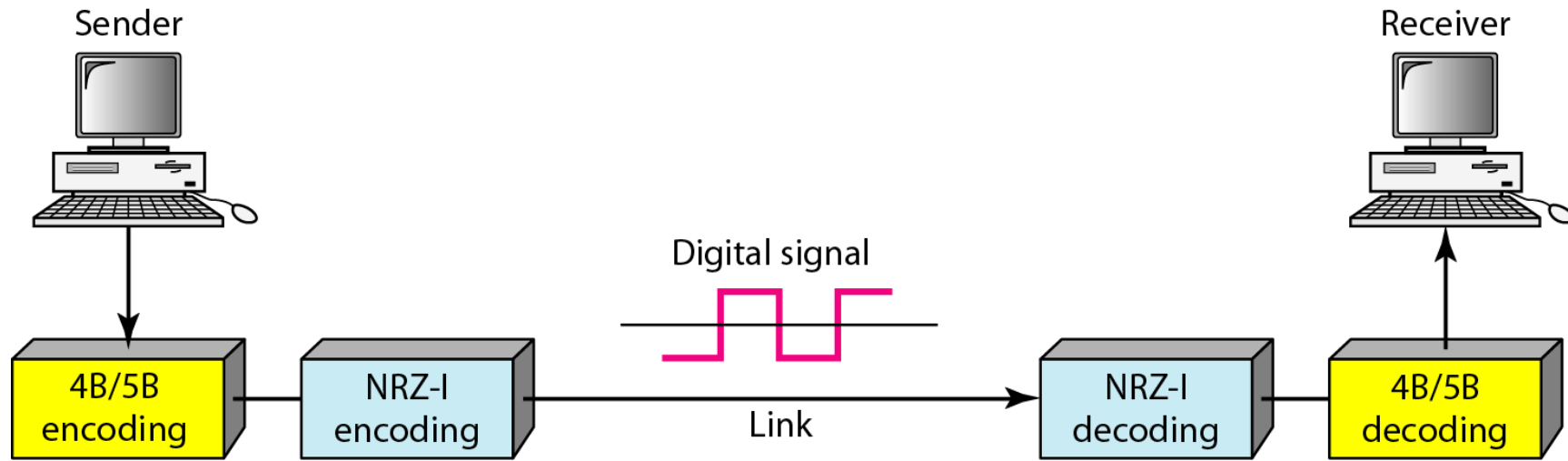
Block Coding

□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

<i>Data Sequence</i>	<i>Encoded Sequence</i>	<i>Control Sequence</i>	<i>Encoded Sequence</i>
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

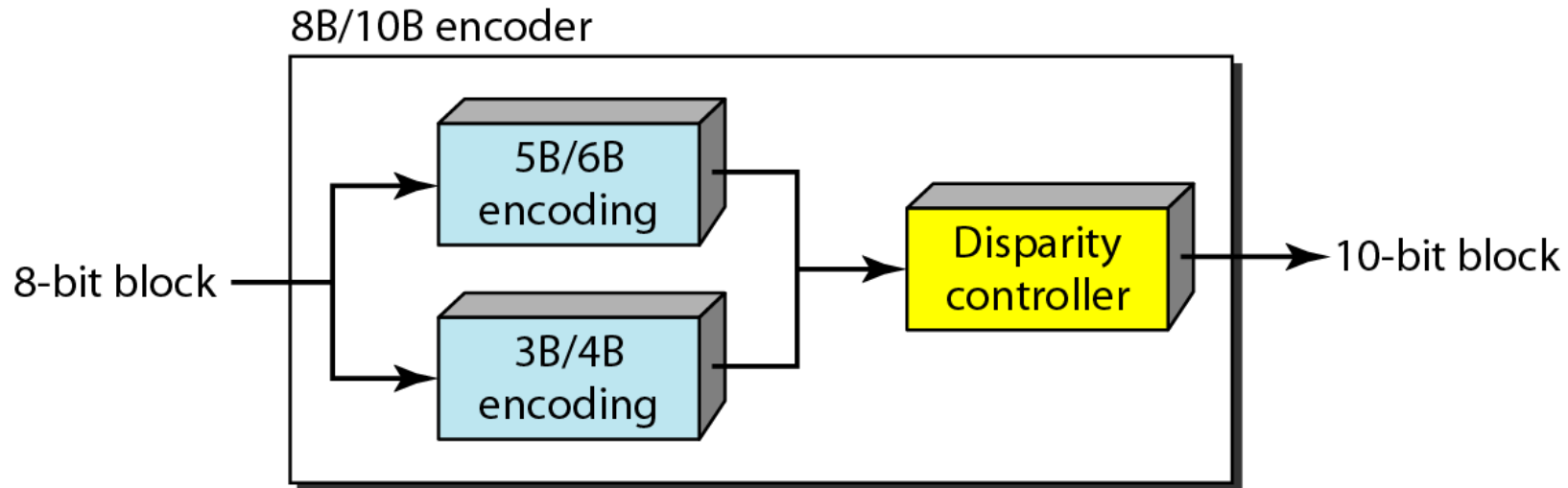
❑ Disadvantages

- ✓ 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 10-bit code.
- ✓ It provides greater error detection capability than 4B/5B. The 8B/10B 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

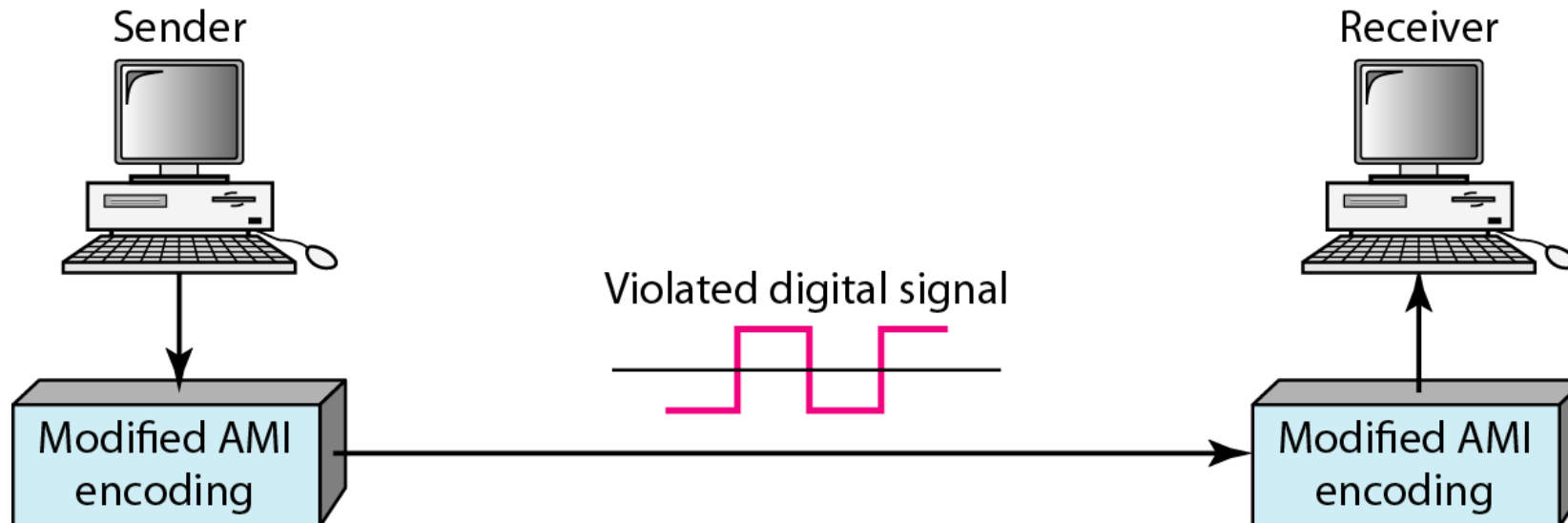
□8B/10B

- ✓ 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 $2^{10} - 2^8 = 768$ 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

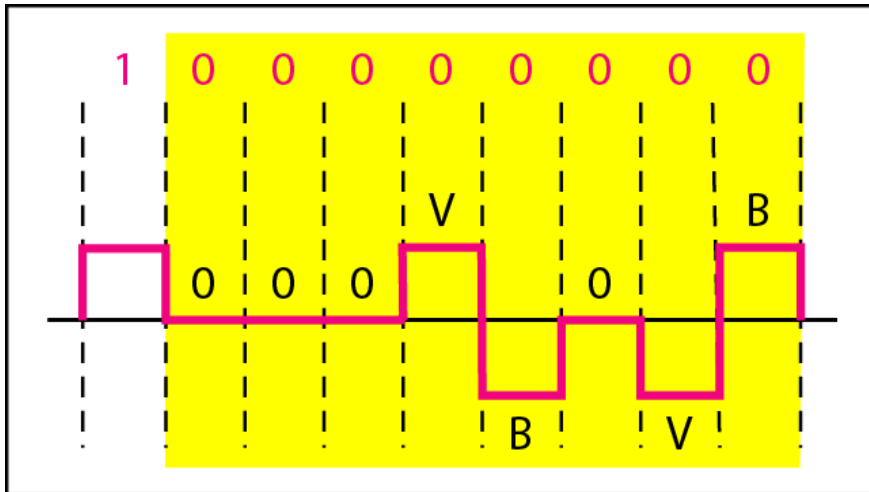
- ✓ Provide synchronization without increasing the number of bits.
- ✓ Part of AMI rule is modified to include scrambling.
- ✓ It is done at same time as line coding.
- ✓ Two common scrambling techniques are B8ZS and HDB3.



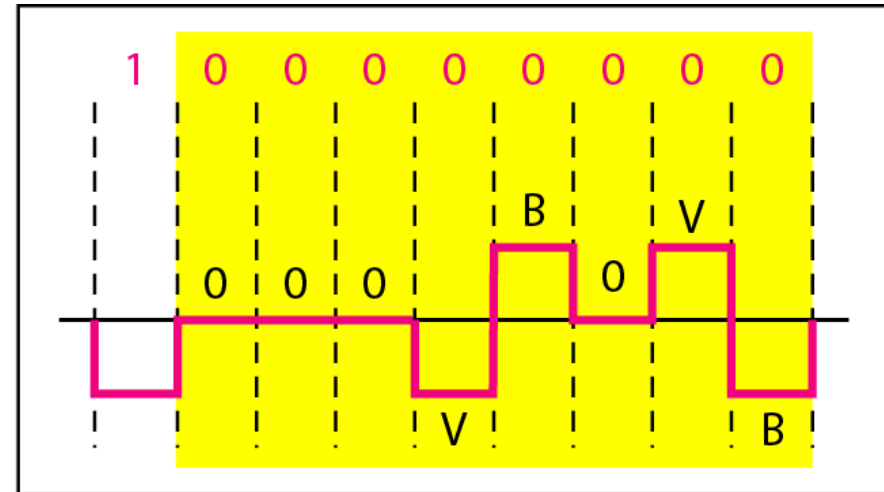
Block Coding

❑ 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.

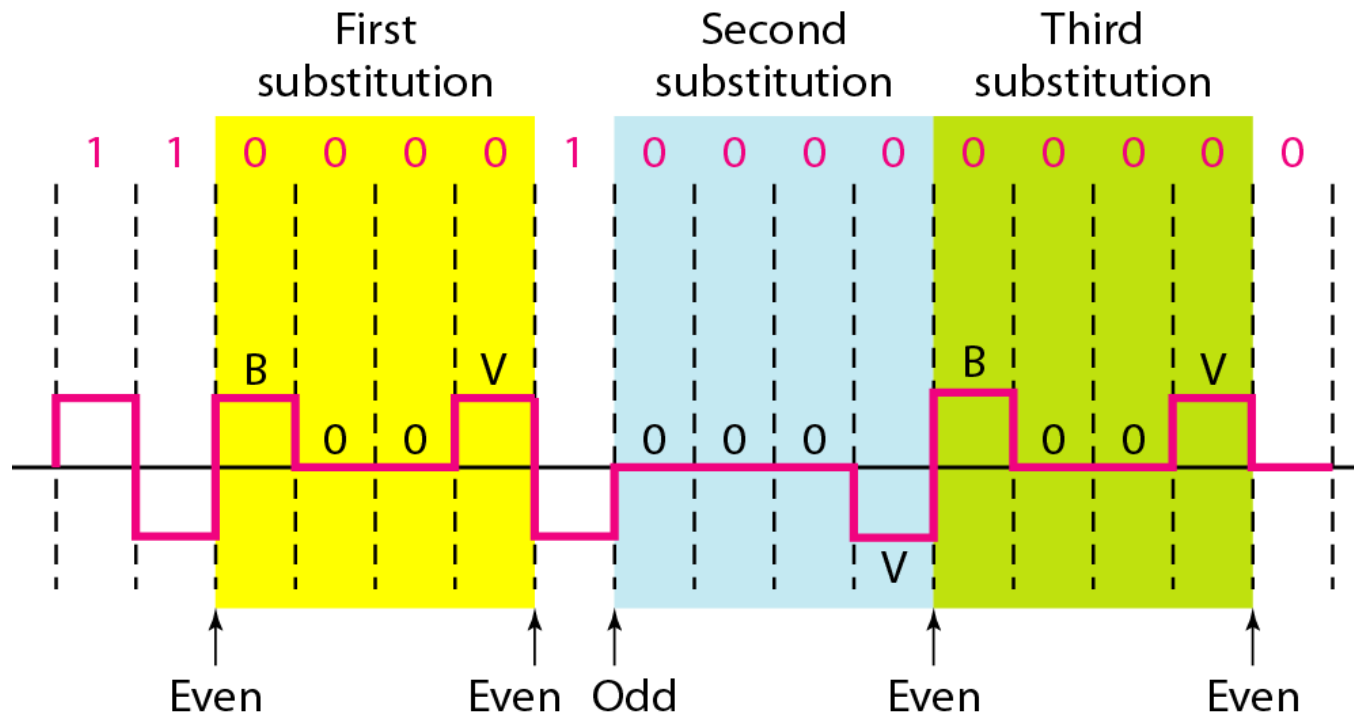
Block Coding

❑ Scrambling(HDB3)

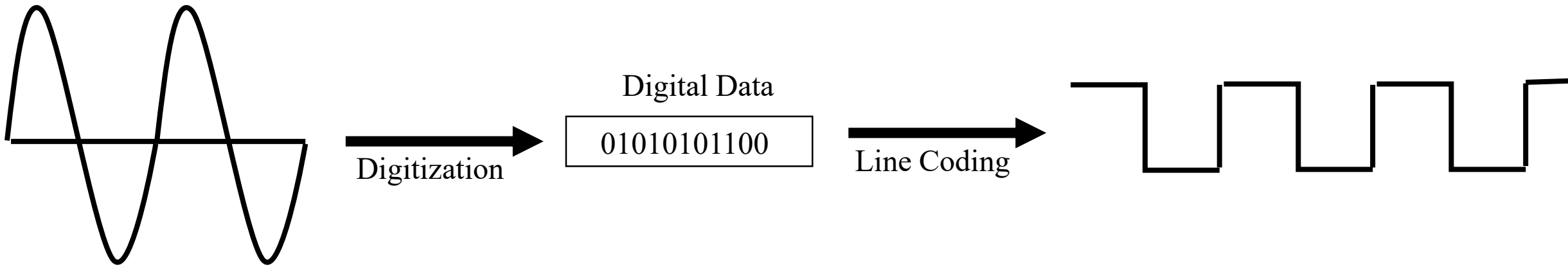
✓ Four consecutive zero-level voltages are replaced with a sequence of 000V or B00V.

✓ **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.

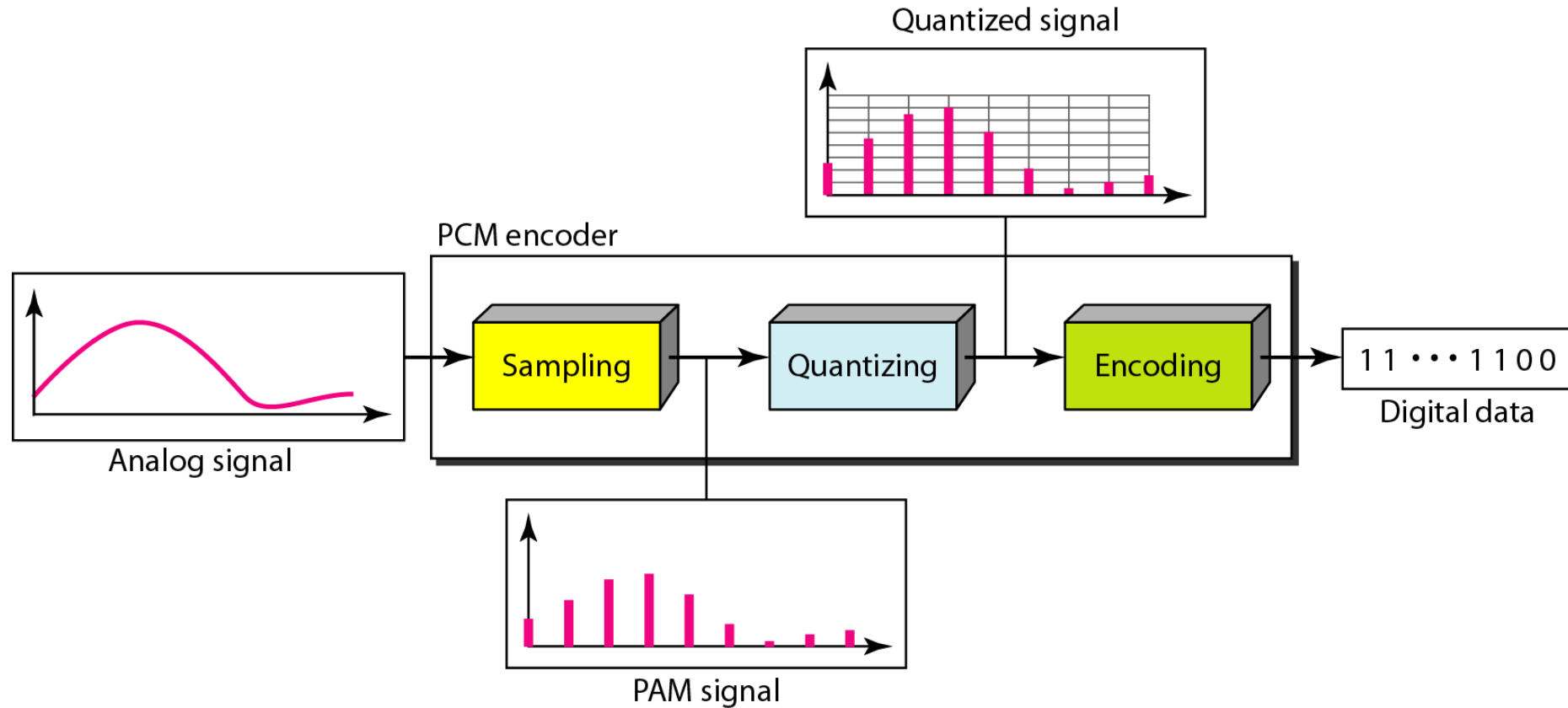
Sampling

Quantizing

Encoding

Analog to Digital Signal

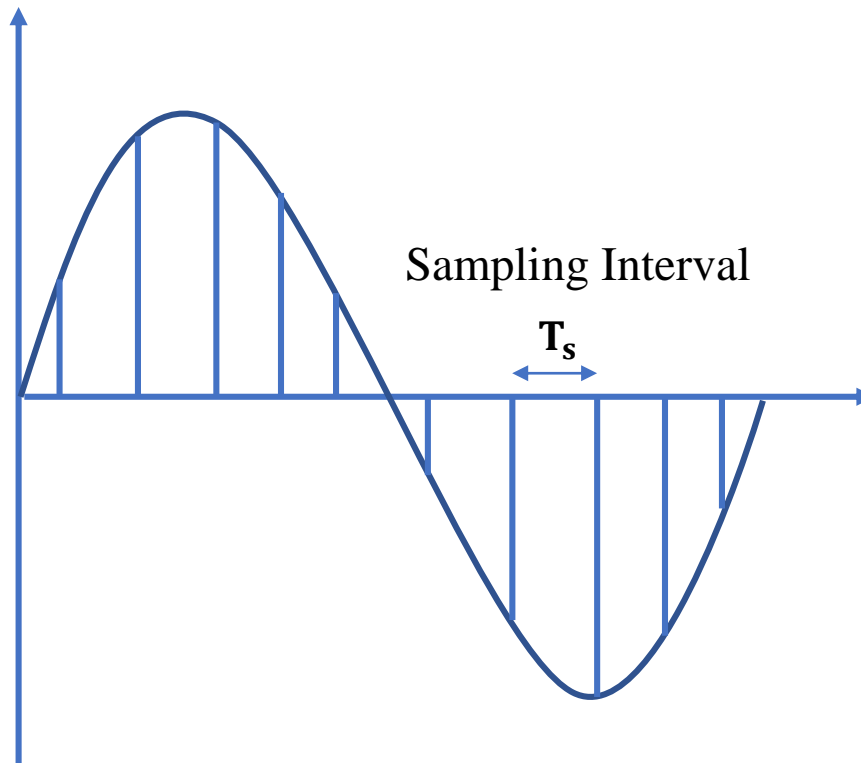
❑ Pulse Code Modulation (PCM)



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

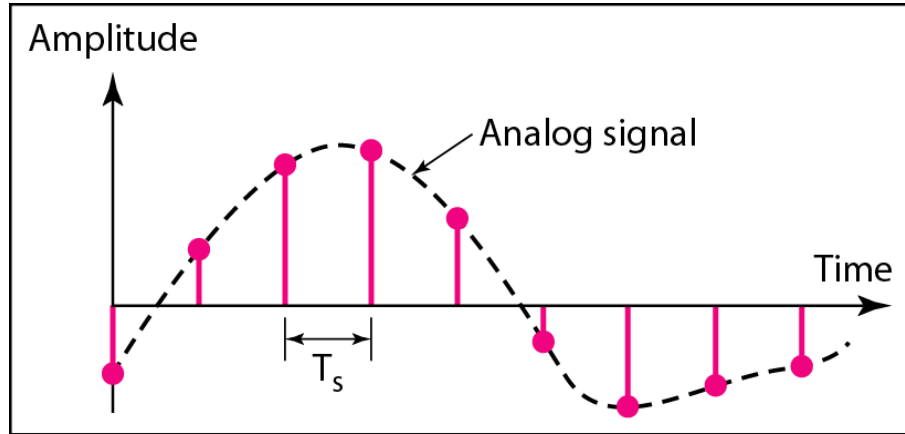
☐ Types of Sampling

✓ Natural Sampling

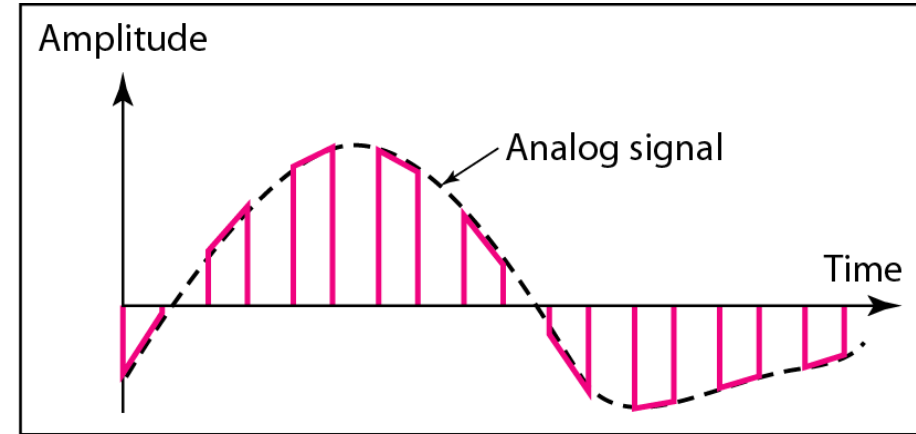
✓ Ideal Sampling

✓ Flat-Top Sampling

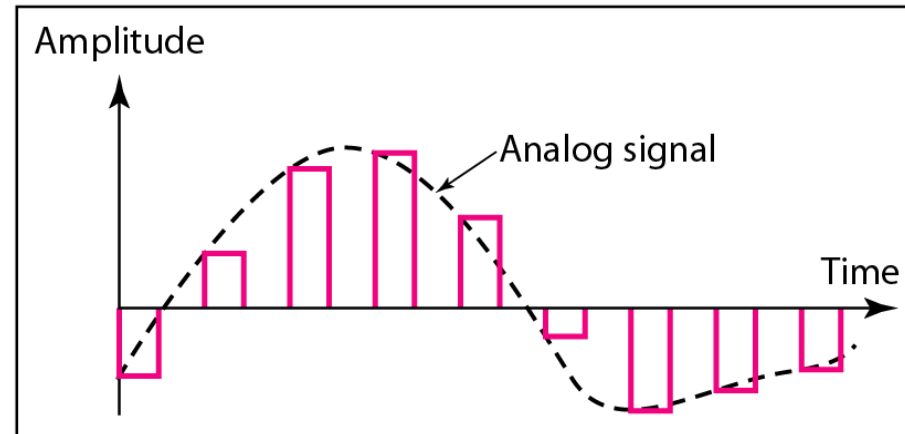
Analog to Digital Signal



a. Ideal sampling



b. Natural sampling

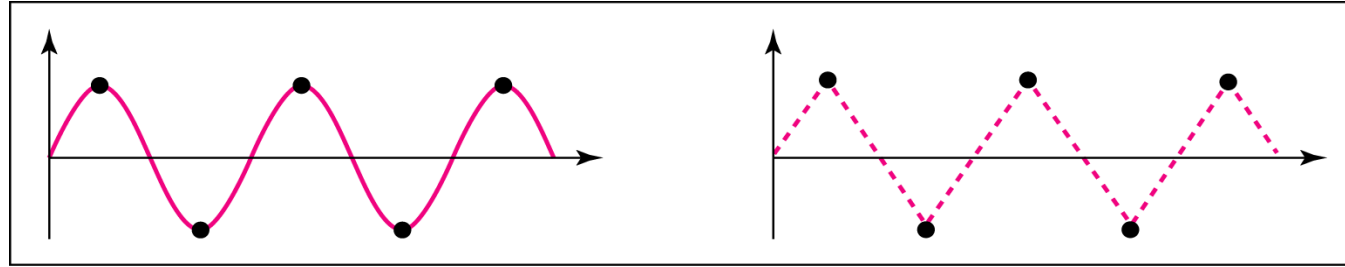


c. Flat-top sampling

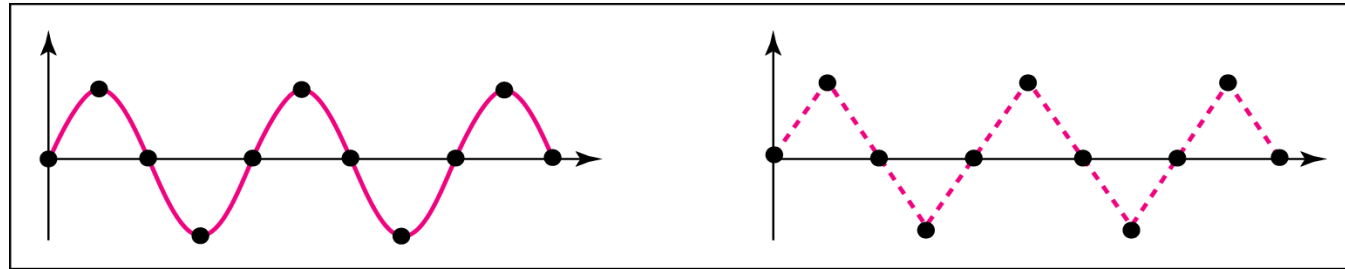
Analog to Digital Signal

□ Nyquist Sampling Theorem

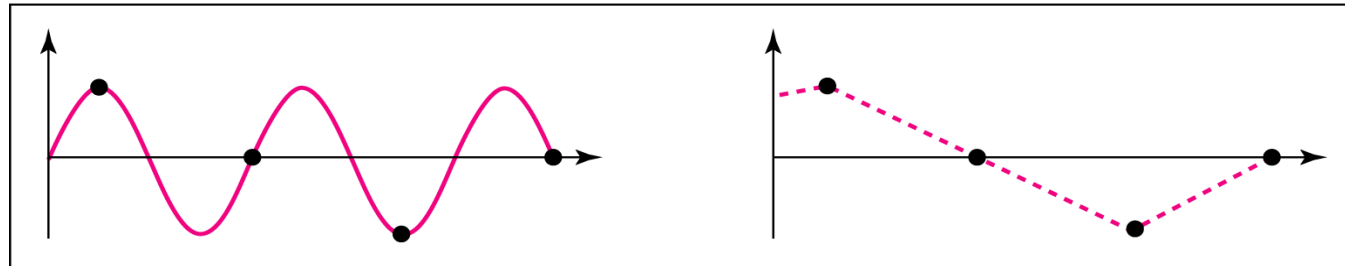
✓ The sampling rate must be at least 2 times the highest frequency contained in the signal.



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



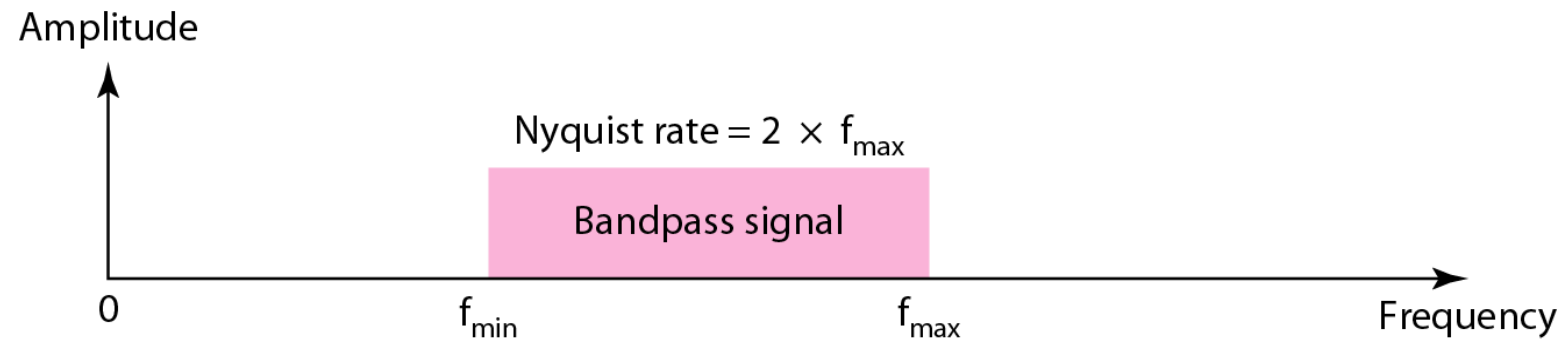
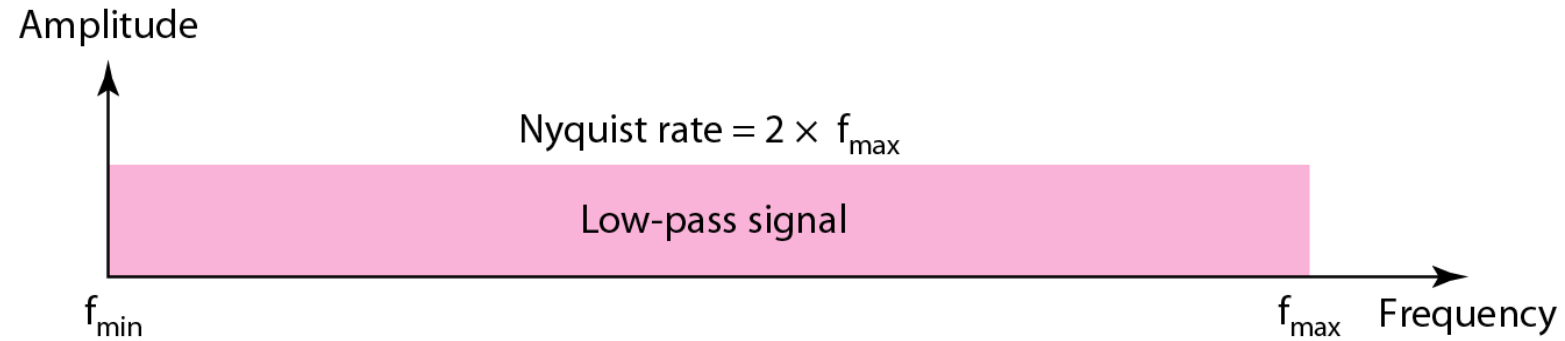
b. Oversampling: $f_s = 4f$



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?

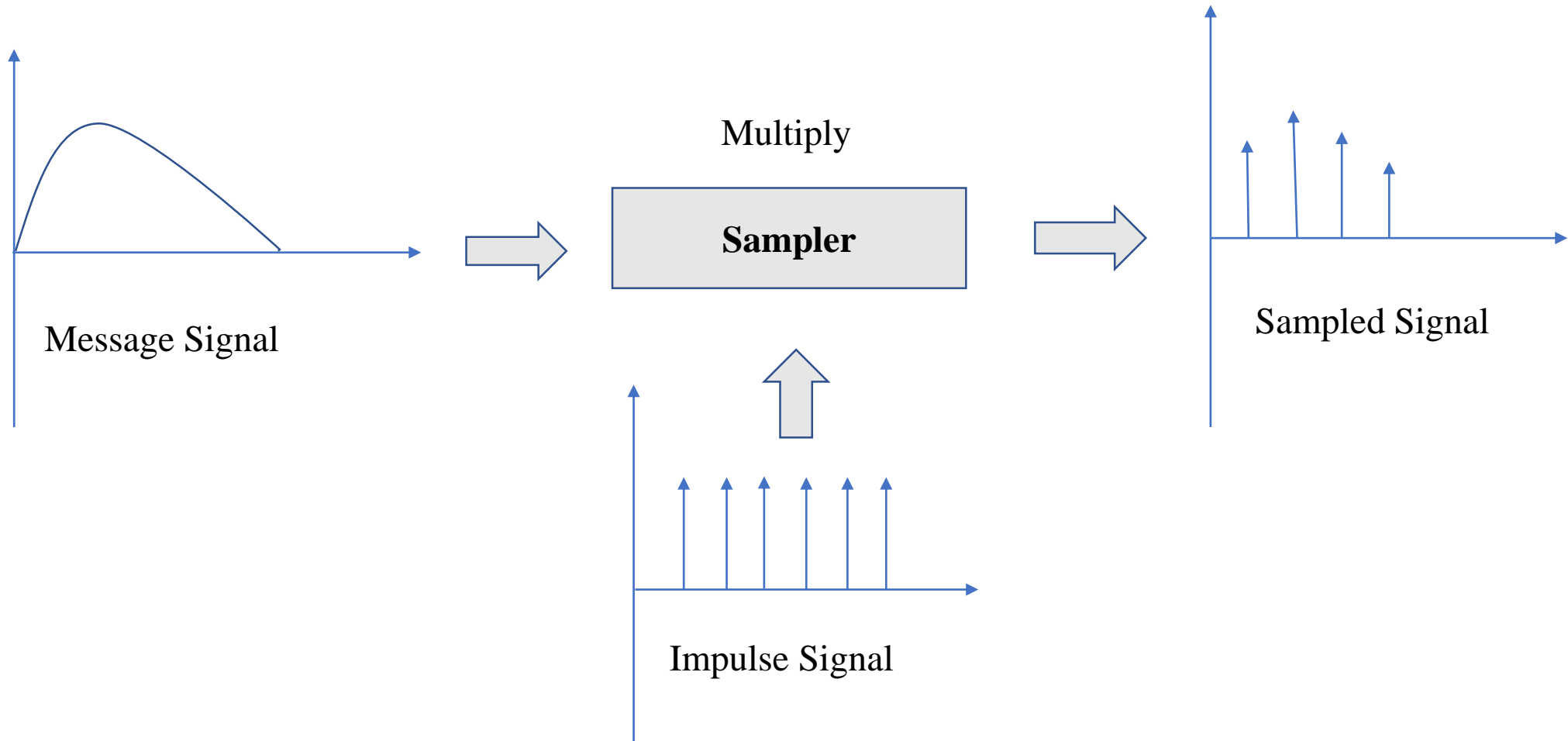
Solution: The bandwidth of a low-pass signal is between 0 and f , where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

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.

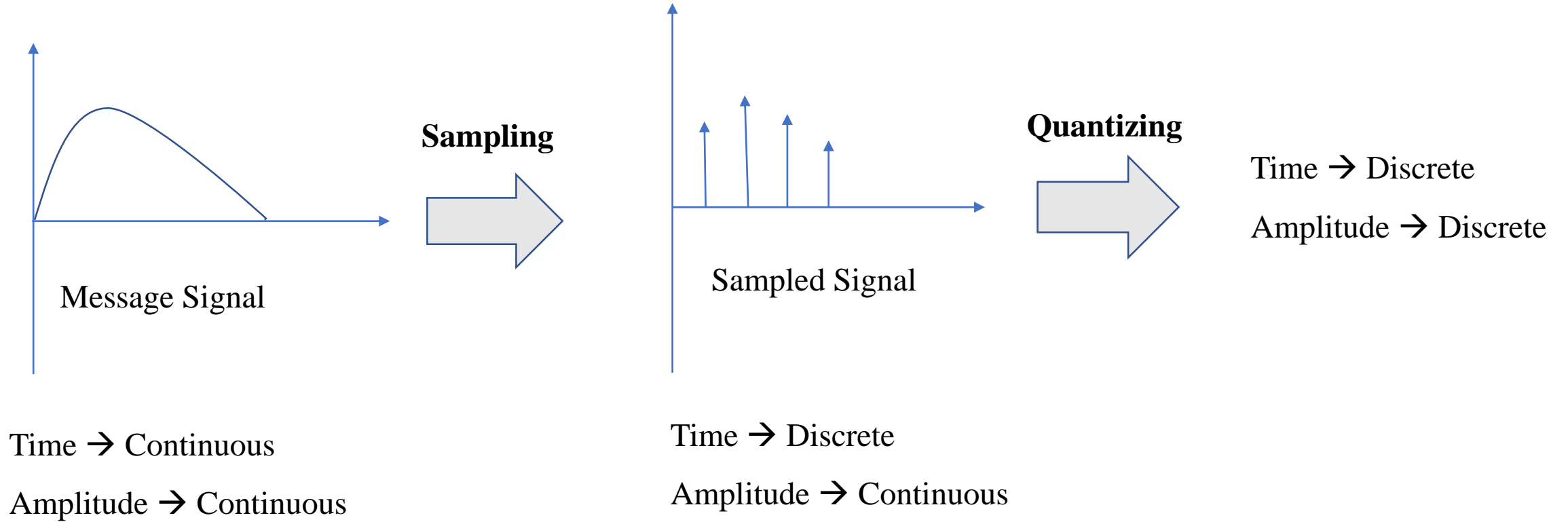
Analog to Digital Signal

❑ Sampling



Analog to Digital Signal

□ Quantization



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.

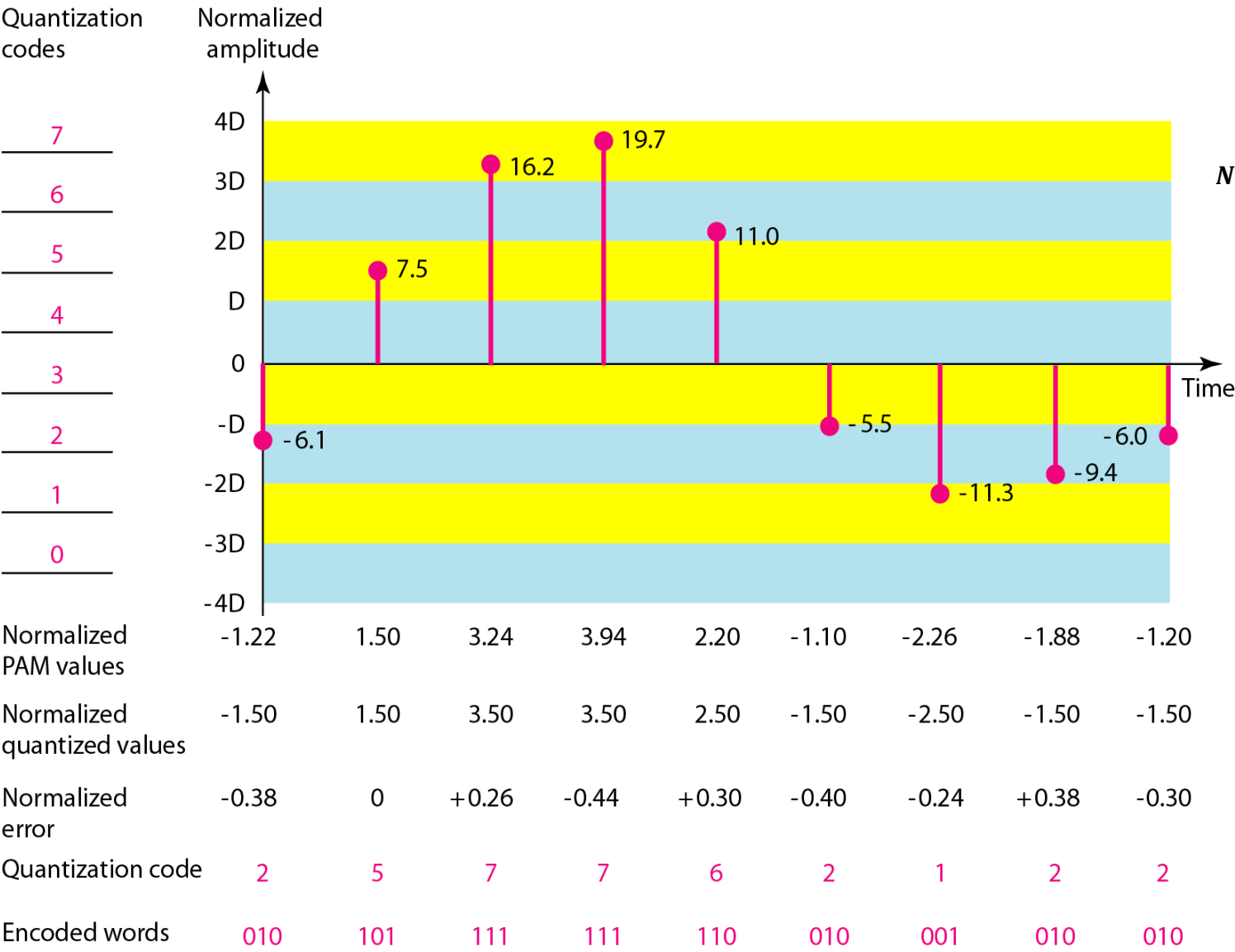
Analog to Digital Signal

□ 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



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

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

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

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 = $1/2 * n_b * f_s$

$$= 1/2 * n_b * 2 * f_{max}$$

$$= n_b * f_{max}$$

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

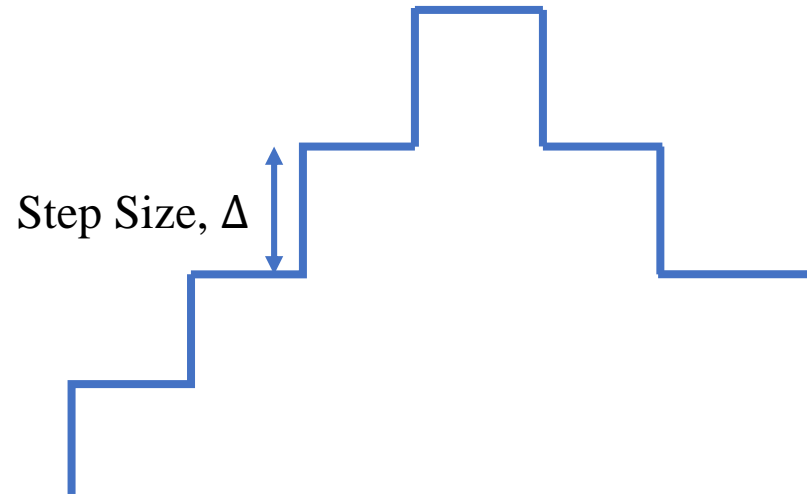
Here, $f_{min}=0$, so bandwidth = f_{max}

Minimum Bandwidth = $n_b * B_{analog}$

Analog to Digital Signal

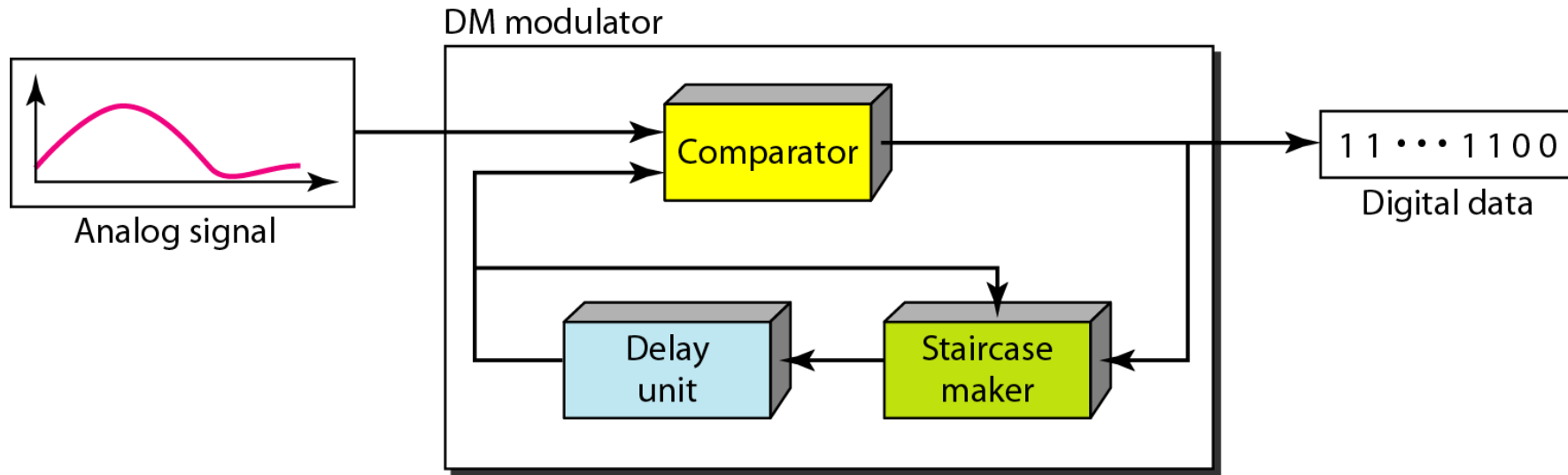
☐ Delta Modulation (DM)

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



Analog to Digital Signal

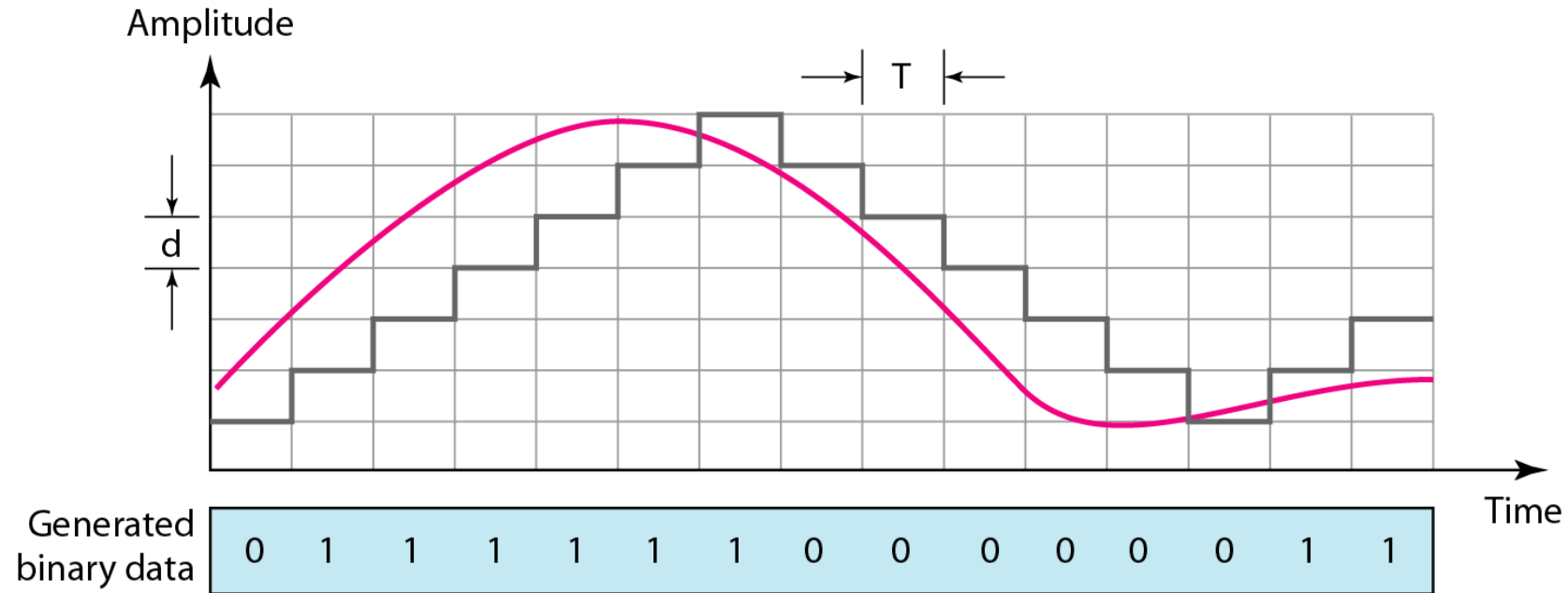
- ✓ At each sampling interval, compares the value of the analog signal with the last value of the gradually made staircase signal.
- ✓ If analog signal $>$ staircase signal $\rightarrow +\Delta (+1)$
- ✓ If analog signal $<$ staircase signal $\rightarrow -\Delta (-0)$



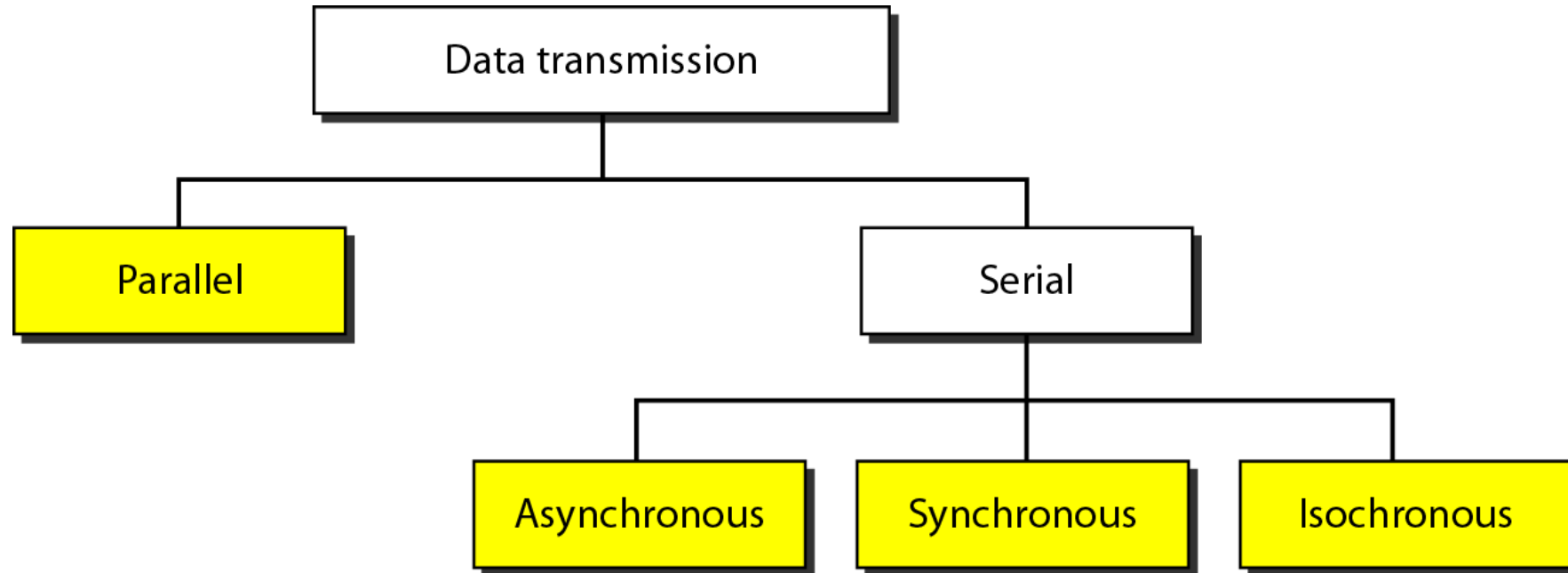
Analog to Digital Signal

✓ If analog signal $>$ staircase signal $\rightarrow +\Delta (+1)$

✓ If analog signal $<$ staircase signal $\rightarrow -\Delta (-0)$



Transmission Modes (Digital Transmission)

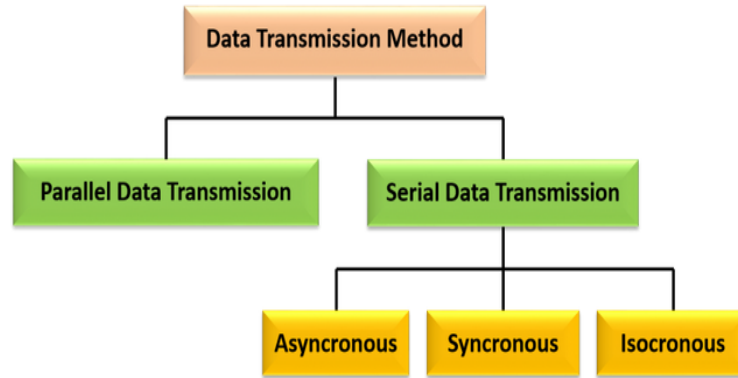


ডেটা ট্রান্সমিশন মেথড কী? (What is Data Transmission Method?)

ডেটা ট্রান্সমিশন বলতে ডেটা পরিবহন বা ডেটার স্থানান্তরকে বুঝায়। ডেটা কমিউনিকেশন সিস্টেমে বিভিন্ন ডিভাইসের সাথে লিঙ্ক স্থাপনের জন্য ব্যবহৃত পদ্ধতি এবং ডেটা ট্রান্সমিশনের জন্য সময়ের সাথে বিট সিনক্রোনাইজেশনকে ডেটা ট্রান্সমিশন মেথড বলা হয়।

অন্যভাবে বলা যায়, ডেটা ট্রান্সমিশন হওয়ার জন্য প্রেরক ও প্রাপকের মধ্যে একটি সুনির্দিষ্ট পদ্ধতি থাকতে হয়, এই পদ্ধতিকে ডেটা ট্রান্সমিশন মেথড বা পদ্ধতি বলে।

ডেটা ট্রান্সমিশন পদ্ধতি বা মেথড কত প্রকার?



সিরিয়াল ও প্যারালাল ট্রান্সমিশনের মধ্যে পার্থক্য

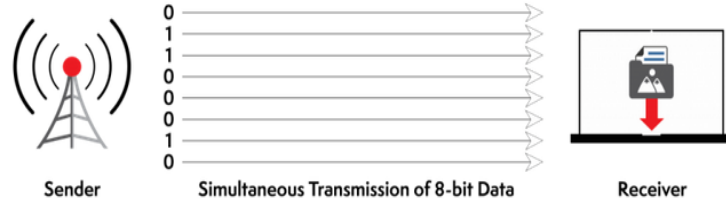
সিরিয়াল ট্রান্সমিশন	প্যারালাল ট্রান্সমিশন
ডেটা প্রবাহিত হয় দুই দিকে, এক বিটের পর অপর বিট	ডেটা প্রবাহিত হয় একাধিক দিকে, একসাথে ৮ টি বিট
খরচ কম	খরচ বেশি
প্রতি ক্লক পালসে ১ টি বিট ট্রান্সফার হয়	প্রতি ক্লক পালসে ৮ টি বিট ট্রান্সফার হয়
ডেটা ট্রান্সমিশন স্পিড কম	ডেটা ট্রান্সমিশন স্পিড বেশি
দীর্ঘ দূরত্বে যোগাযোগের জন্য ব্যবহৃত হয়	কম দূরত্বে যোগাযোগের জন্য ব্যবহৃত হয়
কম্পিউটার থেকে কম্পিউটার	কম্পিউটার থেকে প্রিন্টার

তারের সংযোগ সংখ্যার ভিত্তিতে ডেটা ট্রান্সমিশন পদ্ধতি দুই প্রকারের। যেমন:

1. সমান্তরাল ডেটা ট্রান্সমিশন (Parallel Data Transmission)
2. অনুক্রম ডেটা ট্রান্সমিশন (Serial Data Transmission)

সমান্তরাল ট্রান্সমিশন কী? (What is Parallel Data Transmission?)

সমান্তরাল ডেটা ট্রান্সমিশনে একাধিক চ্যানেলের মাধ্যমে একই সাথে একাধিক ডেটা বিট প্রেরণ করা হয়। প্রতিটি চ্যানেল একই সময়ে একটি বিট বহন করে। অর্থাৎ প্রেরক ও প্রাপকের মধ্যে সমান্তরালে ডেটা স্থানান্তর করলে তাকে সমান্তরাল ডেটা ট্রান্সমিশন বলে।



সমান্তরাল ট্রান্সমিশন মেথড ব্যবহৃত হয় যখন:

- বিপুল পরিমাণে ডেটা প্রেরণ করার প্রয়োজন হয়;
- সময়-সংবেদনশীল ডেটা প্রেরণ করার ক্ষেত্রে;
- এবং ডেটা দ্রুত প্রেরণ করা প্রয়োজন হলে;

উদাহরণস্বরূপ- ভিডিও স্ট্রিমিংয়ে ডেটা প্রেরণে সমান্তরাল ডেটা ট্রান্সমিশন মেথড ব্যবহৃত হয়। কারণ ভিডিও স্ট্রিমিংয়ের জন্য প্রচুর পরিমাণে ডেটা ট্রান্সমিশনের প্রয়োজন হয়। এক্ষেত্রে ডেটাগুলো সময় সংবেদনশীল, কারণ ধীর গতির ডেটা স্ট্রিমগুলোর ফলাফল দর্শকের খারাপ অভিজ্ঞতা হবে। এছাড়া প্যারালাল প্রিন্টার পোর্ট ও ক্যাবল ব্যবহার করে কম্পিউটারের সাথে প্রিন্টারের সংযোগ ইত্যাদি।

সুবিধাঃ এটি প্রোগ্রাম করা সহজ; এবং ডেটা দ্রুত প্রেরণ করা যায়।

অসুবিধাঃ এই পদ্ধতি ব্যয় সাপেক্ষ; কারণ n বিট ডেটা ট্রান্সমিশন করার জন্য n টি চ্যানেল প্রয়োজন।

সিরিয়াল ট্রান্সমিশন কী? (What is Serial Data Transmission?)

সিরিয়াল ডেটা ট্রান্সমিশনে, ডেটা বিটগুলো একের পর এক একক চ্যানেলে প্রেরণ করা হয়। প্রতিটি বিটের একটি ক্লক পালস রয়েছে। একই সময়ে আটটি বিট এর সাথে একটি স্টার্ট এবং স্টপ বিট (যথাক্রমে 0 এবং 1) প্রেরণ করা হয়। এটি একটি নির্ভরযোগ্য ডেটা ট্রান্সমিশন পদ্ধতি; কারণ পূর্ববর্তী ডেটা বিট ইতিমধ্যে পৌঁছে থাকলে কেবল অপর একটি ডেটা বিট প্রেরণ করা হয়। একের পর এক নির্দিষ্ট ক্রমে ডেটা বিট প্রেরণ করার ফলে এটি ডেটা ইন্টিগ্রিটি(integrity) নিশ্চিত করে। এইভাবে, ডেটা বিট একে অপরের সাথে সিলেক্ট প্রাপ্ত হয়।



সিরিয়াল ট্রান্সমিশন ব্যবহৃত হয় যখন:

- সাধারণত দীর্ঘ-দূরত্বে ডেটা স্থানান্তরের প্রয়োজন হয়।
- ডেটার পরিমাণ তুলনামূলকভাবে কম হয়।

উদাহরণস্বরূপ- মডেম, মাউস, ইত্যাদি যন্ত্রে সিরিয়াল ডেটা ট্রান্সমিশন পদ্ধতিতে ডেটা আদান-প্রদান হয়ে থাকে।

সুবিধাঃ ব্যয় কম, কারণ ডেটা চলাচলের জন্য একটি লাইন বা চ্যানেলের প্রয়োজন হয়।

অসুবিধাঃ ধীর গতি সম্পন্ন।

সিরিয়াল ট্রান্সমিশন এর প্রকারভেদ

বিট সিনক্রোনাইজেশন ও সময় নিয়ন্ত্রণের উপর ভিত্তি করে সিরিয়াল ডেটা ট্রান্সমিশন মেথড তিন প্রকার।

১। অ্যাসিনক্রোনাস ট্রান্সমিশন (Asynchronous Transmission)

২। সিনক্রোনাস ট্রান্সমিশন (Synchronous Transmission)

৩। আইসোক্রোনাস ট্রান্সমিশন (Isochronous Transmission)

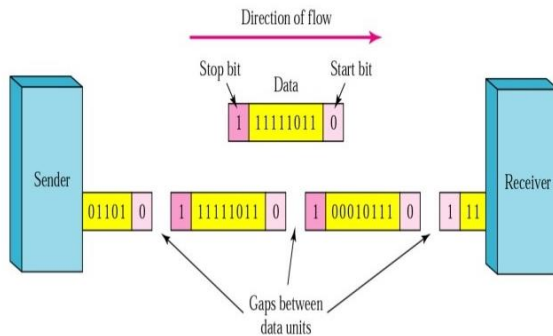
সিরিয়াল ডেটা ট্রান্সমিশন পদ্ধতিতে ডেটা ট্রান্সমিশনের সময় বিভিন্ন বিটের মধ্যে সমন্বয়ের প্রয়োজন হয়। এই সমন্বয় না থাকলে প্রাপক যন্ত্র ডেটার শুরু ও শেষ বুঝতে পারে না। ফলে ডেটা পুনরুদ্ধার করা সম্ভব হয় না। তাই এই ট্রান্সমিশন পদ্ধতিগুলো বিট সিনক্রোনাইজেশন ব্যবহার করে। ডেটা ট্রান্সমিশনের শুরু এবং শেষ চিহ্নিত করতে বিট সিনক্রোনাইজেশন প্রয়োজন হয়। ডেটা ট্রান্সমিশনের সময় কখন ডেটা শুরু এবং শেষ হয় তা সনাক্ত করতে বিট সিনক্রোনাইজেশন প্রাপক কম্পিউটারকে সাহায্য করে। অতএব, বিট সিনক্রোনাইজেশন সময় নিয়ন্ত্রণ করে। অর্থাৎ সিরিয়াল ডেটা ট্রান্সমিশন পদ্ধতিতে ডেটা ট্রান্সমিশনের সময় বিভিন্ন বিটের মধ্যে সমন্বয়ের জন্য ব্যবহৃত পদ্ধতিকে বলা হয় বিট সিনক্রোনাইজেশন।

অ্যাসিনক্রোনাস ট্রান্সমিশন কী? (What is Asynchronous Transmission?)

অ্যাসিনক্রোনাস ট্রান্সমিশন(Asynchronous Transmission) ডেটা প্রেরক হতে প্রাপকে অসম সময়ের ব্যবধানে কারেক্টার বাই কারেক্টার পাঠানো হয়।

যখনই ডেটা প্রচার করা হয় তখন এটি একবারে পাঠানো হয়। সুতরাং ডেটা সংরক্ষণের জন্য প্রেরকের কোনও মেমোরির প্রয়োজন হয় না। সাধারণত ডেটা বাইট এর সাথে একটি স্টার্ট বিট এবং একটি স্টপ বিট যুক্ত করা হয় এবং মোট ১০ বিট ডেটা প্রেরণ করা হয়। কখনও কখনও ত্রুটি(Error) পরীক্ষা করার জন্য স্টপ বিটের আগে একটি প্যারিটি বিট যুক্ত করা হয়।

এ ধরনের ট্রান্সমিশনে প্রেরক যে কোনো সময় ডেটা প্রেরণ করতে পারে এবং প্রাপক তা গ্রহণ করতে পারে। প্রেরক এবং প্রাপককে সিনক্রোনাইজ করার জন্য এবং ডেটা সঠিকভাবে প্রাপক প্রাপ্তে পৌঁছেছে তা নিশ্চিত করার জন্য ডেটা বাইটগুলোর মধ্যে স্টপ বিট এবং স্টার্ট বিট ব্যবহার করা হয়। ডেটা বিট প্রেরণ এবং গ্রহণের মধ্যবর্তী সময় সমান নয়, তাই ডেটা ট্রান্সমিশনে সময় দেওয়ার জন্য ফাঁকা(gaps) ব্যবহৃত হয়।



অ্যাসিনক্রোনাস ট্রান্সমিশনের সুবিধা

১। যেকোনো সময় প্রেরক ডেটা পাঠাতে পারে এবং প্রাপক তা গ্রহণ করতে পারে।

২। প্রেরক স্টেশনে প্রাইমারি স্টোরের প্রয়োজন হয় না।

৩। তুলনামূলকভাবে খরচ কম।

৪। কম ডেটা ট্রান্সমিশনের ক্ষেত্রে বেশি উপযোগী।

অ্যাসিনক্রোনাস ট্রান্সমিশনের অসুবিধা

১। দক্ষতা ও গতি কম।

২। যখন ডেটা চলাচল বন্ধ থাকে তখন মাধ্যমটি অব্যবহৃত অবস্থায় পড়ে থাকে।

অ্যাসিনক্রোনাস ট্রান্সমিশনের ব্যবহার

১। কম্পিউটার হতে প্রিন্টারে

২। কার্ড রিডার হতে কম্পিউটারে

৩। কম্পিউটার হতে কার্ড রিডারে

৪। কীবোর্ড হতে কম্পিউটারে

অ্যাসিনক্রোনাস ট্রান্সমিশনের দক্ষতা

$$\text{দক্ষতা} = \frac{\text{প্রকৃত ডেটা বিট}}{\text{মোট ডেটা বিট}} \times ১০০\%$$

০.৮৬৮৮৮

অ্যাসিনক্রোনাস ট্রান্সমিশনের দক্ষতা

দক্ষতা = $\frac{\text{প্রকৃত ডেটা বিট}}{\text{মোট ডেটা বিট}} \times 100\%$

এখানে,

প্রকৃত ডেটা বিট হলো যে পরিমাণ ডেটা বিট প্রেরণ করা হবে

মোট ডেটা বিট হলো প্রকৃত ডেটা বিট ও ওভারহেড ডেটা বিটের সমষ্টি

ওভারহেড ডেটা বিট হলো স্টার্ট বিট (১ বিট), স্টপ বিট(১ বিট) ও প্যারিটি বিট(১ বিট)

উদাহরণ: অ্যাসিনক্রোনাস ট্রান্সমিশন পদ্ধতি ব্যবহার করে 20KB ডেটা ট্রান্সমিশনের দক্ষতা নির্ণয়।

সমাধান:

প্রকৃত ডেটা বিট = 20KB = 20 × 8 Kb = 160 Kb = 1000 × 160 b = 160000 bit

প্রতিটি 8 bit গ্রুপের জন্য ওভারহেড ডেটা বিট প্রয়োজন = 3 bit

সুতরাং 160000 bit ডেটা প্রেরণের জন্য মোট ওভারহেড ডেটা বিট প্রয়োজন = (3/8) × 160000 = 60000 bit

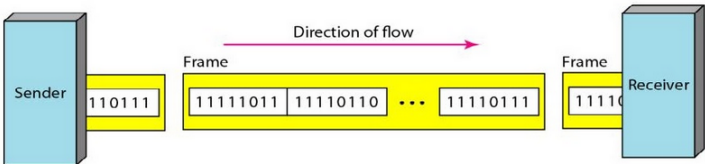
মোট ডেটা বিট = প্রকৃত ডেটা বিট + ওভারহেড ডেটা বিট = 160000 bit + 60000 bit = 220000 bit

সুতরাং, দক্ষতা = (160000/220000) × 100% = 73%

সিনক্রোনাস ট্রান্সমিশন কী? (What is Synchronous Transmission?)

সিনক্রোনাস ট্রান্সমিশনে (Synchronous Transmission) ডেটা প্রেরক হতে প্রাপকে সমান সময়ের ব্যবধানে ব্লক বাই ব্লক বা ফ্রেম আকারে পাঠানো হয়।

প্রথমে প্রেরক স্টেশনে ডেটাগুলোকে প্রাথমিক স্টোরেজে সংরক্ষণ করে নেওয়া হয়। তারপর ডেটার ক্যারেঙ্করগুলোকে ব্লক বা প্যাকেট আকারে ভাগ করে নিয়মিত বিরতিতে প্রতিবারে একটি করে ব্লক বা প্যাকেট ট্রান্সমিট করা হয়। ব্লক ডেটার শুরুতে এবং শেষে যথাক্রমে হেডার(১৬ বিট) এবং ট্রেইলার(১৬ বিট) ইনফরমেশন থাকে। সাধারণত ৪০ হতে ১৩২ টি ক্যারেঙ্কর নিয়ে এক একটি ব্লক তৈরি হয়। তবে ব্লকে ক্যারেঙ্কর 12৪, 256, 512, 1024 সংখ্যক হতে পারে। দুটি ব্লকের মাঝখানে সময় বিরতি সমান হয়ে থাকে।



সিনক্রোনাস ট্রান্সমিশনের সুবিধা

১। সিনক্রোনাস ট্রান্সমিশনের দক্ষতা অ্যাসিনক্রোনাসের তুলনায় বেশি।

২। ট্রান্সমিশন গতি অনেক বেশি।

৩। প্রতি ক্যারেঙ্করের শুরুতে Start bit এবং শেষে Stop bit এর প্রয়োজন হয় না।

৪। অধিক ডেটা স্থানান্তর করার ক্ষেত্রে, এই পদ্ধতিটি উপযুক্ত।

সিনক্রোনাস ট্রান্সমিশনের সুবিধা

১। সিনক্রোনাস ট্রান্সমিশনের দক্ষতা অ্যাসিনক্রোনাসের তুলনায় বেশি।

২। ট্রান্সমিশন গতি অনেক বেশি।

৩। প্রতি ক্যারেঙ্করের শুরুতে Start bit এবং শেষে Stop bit এর প্রয়োজন হয় না।

৪। অধিক ডেটা স্থানান্তর করার ক্ষেত্রে, এই পদ্ধতিটি উপযুক্ত।

সিনক্রোনাস ট্রান্সমিশনের অসুবিধা

১। প্রেরক স্টেশনে একটি প্রাইমারি স্টোরেজ ডিভাইসের প্রয়োজন হয়।

২। তুলনামূলকভাবে ব্যয়বহল।

৩। প্রেরক এবং প্রাপকের মধ্যে সিনক্রোনাইজেশন প্রয়োজন হয়।

সিনক্রোনাস ট্রান্সমিশনের ব্যবহার

১। কম্পিউটার থেকে কম্পিউটারে ডেটা কমিউনিকেশনে ।

২। এক স্থান থেকে দূরবর্তী কোন স্থানে ডেটা স্থানান্তরে ।

সিনক্রোনাস ট্রান্সমিশনের দক্ষতা

দক্ষতা = $\frac{\text{প্রকৃত ডেটা বিট}}{\text{মোট ডেটা বিট}} \times 100\%$

সিনক্রোনাস ট্রান্সমিশনের দক্ষতা

দক্ষতা = $\frac{\text{প্রকৃত ডেটা বিট}}{\text{মোট ডেটা বিট}} \times ১০০\%$

এখানে,
প্রকৃত ডেটা বিট হলো যে পরিমাণ ডেটা বিট প্রেরণ করা হবে
মোট ডেটা বিট হলো প্রকৃত ডেটা বিট ও ওভারহেড ডেটা বিটের সমষ্টি
ওভারহেড ডেটা বিট হলো হেডার ইনফরমেশন(১৬ বিট) ও ট্রেইলার ইনফরমেশন(১৬ বিট)

উদাহরণ: সিনক্রোনাস ট্রান্সমিশন পদ্ধতি ব্যবহার করে 20KB ডেটা ট্রান্সমিশনের দক্ষতা নির্ণয়।

সমাধান:
প্রকৃত ডেটা বিট = 20KB = 20 × ৪ Kb = 160 Kb = 1000 × 160 b = 160000 bit
হারা ফ্রাক, প্রতিটি ব্লক ৮০ টি ক্যারেক্টার আছে, অর্থাৎ ব্লক সাইজ = 80×৪ bit = 640 bit
প্রতিটি 640 bit গ্রুপের ব্লকের জন্য মোট ওভারহেড ডেটা বিট প্রয়োজন = 32 bit
সুতরাং 160000 bit ডেটা প্রেরণের জন্য মোট ওভারহেড ডেটা বিট প্রয়োজন = (32/640)× 160000 = 8000 bit
মোট ডেটা বিট = প্রকৃত ডেটা বিট + ওভারহেড ডেটা বিট = 160000 bit + 8000 bit = 168000 bit
সুতরাং, দক্ষতা = (160000/168000) × 100% = 95%

আইসোক্রোনাস ট্রান্সমিশন কী (What is Isochronous Transmission?)

এটি অ্যাসিনক্রোনাস ও সিনক্রোনাস ডেটা ট্রান্সমিশন এর সমন্বিত রূপ, যাকে আবার সিনক্রোনাস ট্রান্সমিশনের উন্নত ভার্সনও বলা হয়ে থাকে। এ পদ্ধতিতে প্রেরক হতে প্রাপকে অ্যাসিনক্রোনাস পদ্ধতির স্টার্ট ও স্টপ বিটের মাঝে ব্লক আকারে ডেটা স্থানান্তরিত হয়। তবে দুটি ব্লকের মধ্যে সময়ের পার্থক্য 0 (শূন্য) একক করার চেষ্টা করা হয়। যেহেতু পুরোটা সিনক্রোনাস নয় তাই স্টোরেজ ডিভাইসে ডেটা সংরক্ষণ ন করেই যখন প্রয়োজন তখন সেই ডেটা ট্রান্সমিট করা যায়।

আইসোক্রোনাস ট্রান্সমিশনের সুবিধা

- ১। ট্রান্সমিশন স্পিড অনেক বেশি।
- ২। প্রেরক স্টেশনে প্রাইমারি স্টোরেজ ডিভাইসের প্রয়োজন হয় না।

আইসোক্রোনাস ট্রান্সমিশনের অসুবিধা

- ১। ডেটা ব্লক যথাযথভাবে প্রাপক পেয়েছে কিনা তা চেক করা যায় না এবং ভুল সংশোধন করার ব্যবস্থা নেই।
- ২। তুলনামূলকভাবে ব্যয়বহল।

আইসোক্রোনাস ট্রান্সমিশনের ব্যবহার

- ১। সাধারণত রিয়েল টাইম অ্যাপ্লিকেশনের ডেটা ট্রান্সফারে এ পদ্ধতি ব্যবহার করা হয়।
- ২। বিভিন্ন মাল্টিমিডিয়া কমিউনিকেশন যেমন- অডিও বা ভিডিও কল এর জন্য এই পদ্ধতিতে ডেটা ট্রান্সমিশন হয়ে থাকে।

অ্যাসিনক্রোনাস এবং সিনক্রোনাস ট্রান্সমিশনের মধ্যে পার্থক্য

সিনক্রোনাস ট্রান্সমিশন	অ্যাসিনক্রোনাস ট্রান্সমিশন
ব্লক বাই ব্লক ডেটা প্রেরণ করে	ক্যারেন্টার বাই ক্যারেন্টার ডেটা প্রেরণ করে
ডেটা ট্রান্সমিশন গতি বেশি	ডেটা ট্রান্সমিশন গতি কম
ব্যয়বহল মেথড	কম ব্যয়বহল মেথড
ব্লকসমূহের মধ্যে সময়ের ব্যবধান সমান থাকে	ক্যারেন্টার সমূহের মধ্যে সময়ের ব্যবধান অসমান থাকে
স্টোরেজ ডিভাইস এর প্রয়োজন হয়	স্টোরেজ ডিভাইস এর প্রয়োজন হয় না
Chat Rooms, Telephonic Conversations, Video Conferencing ইত্যাদি ক্ষেত্রে ব্যবহৃত হয়।	Email, Forums, Letters ইত্যাদি ক্ষেত্রে ব্যবহৃত হয়।

Transmission Modes (Digital Transmission)

❑ Parallel Transmission

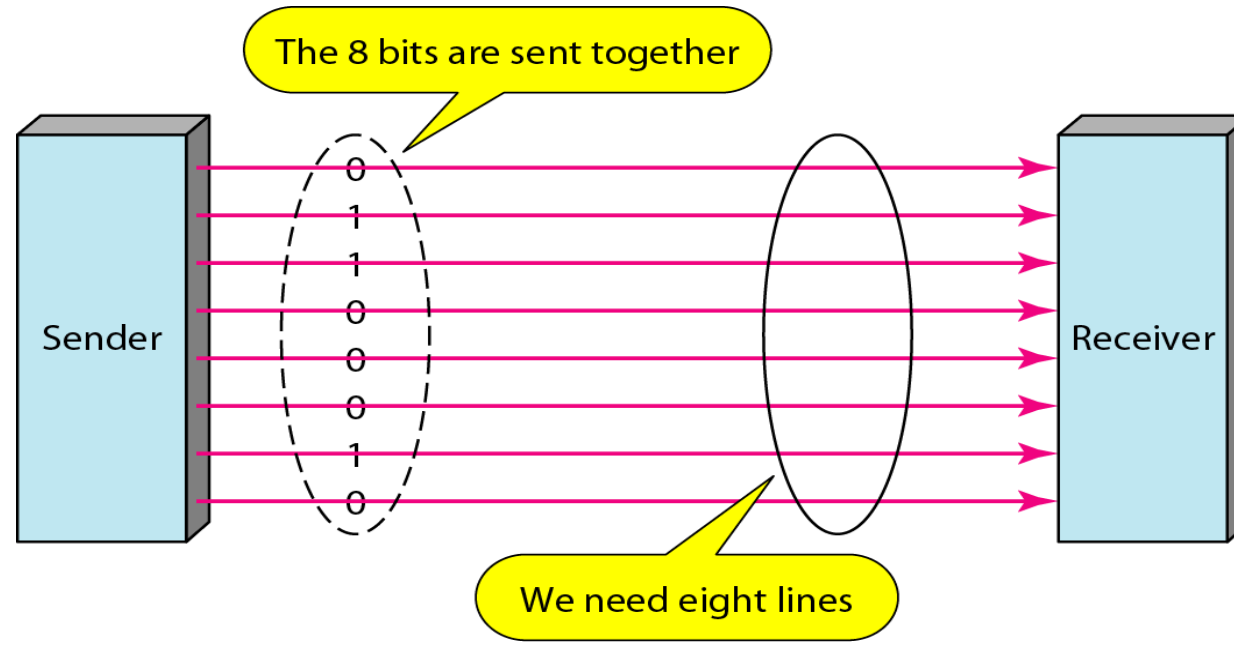
- ✓ Use n wires to send n bits at one time.

❑ Advantages

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

❑ Disadvantages

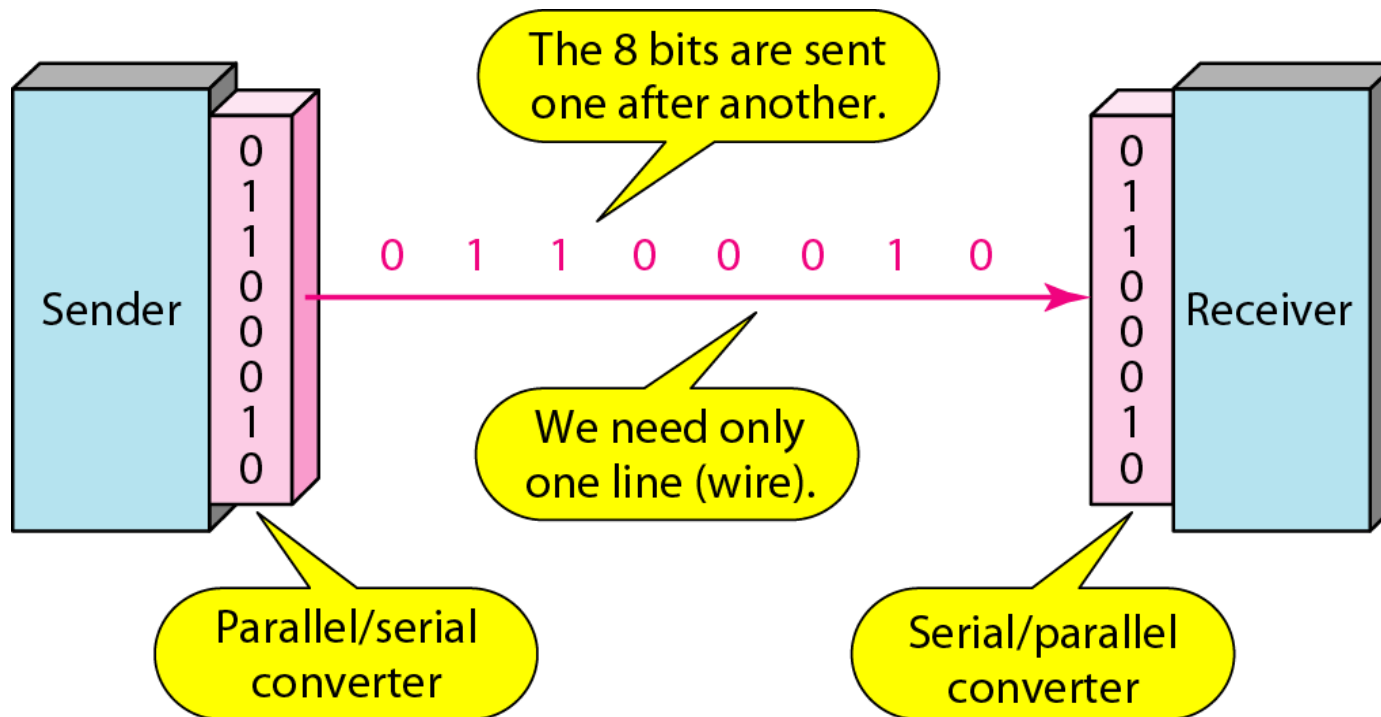
- ✓ Parallel transmission requires n communication lines. Because this is expensive, parallel transmission is usually limited to short distances.



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 .

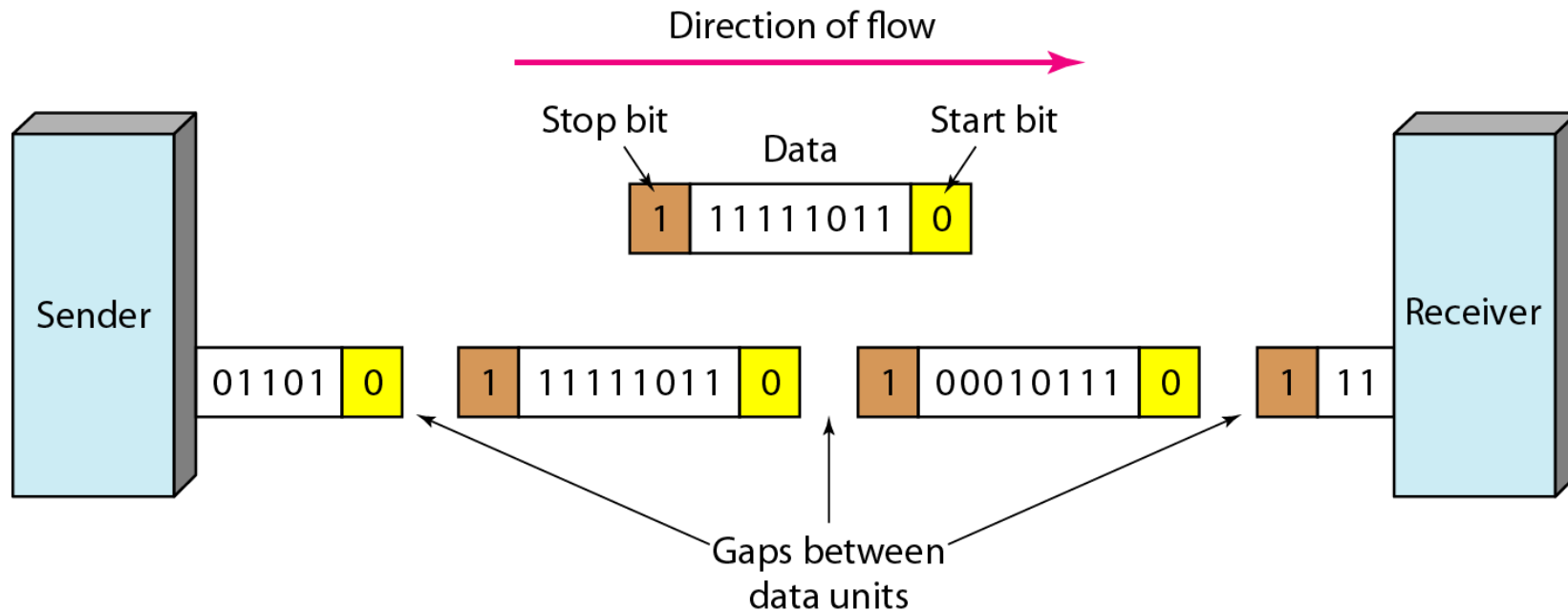
❑ Disadvantages of Serial Transmission

- ✓ 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 (1s) 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.

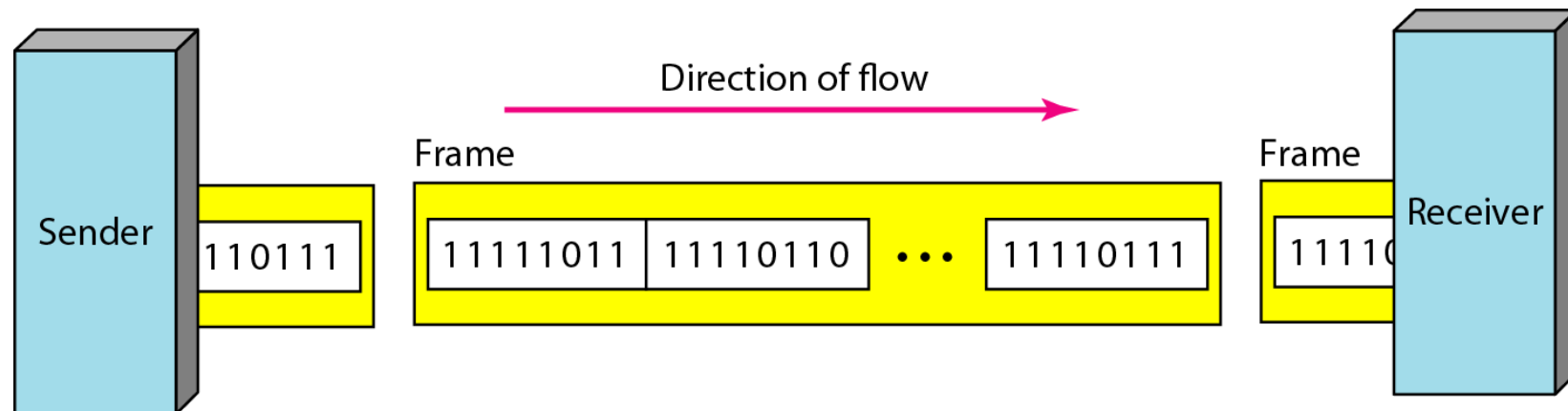
❑ Uses

- ✓ 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.