Signal Encoding Techniques

Dr. Sumit Srivastava

Dept. of CSE, BIT Mesra Ranchi

Email:- sumit.srivs88@gmail.com

References

- Tanenbaum and David J Wetherall, Computer Networks, 5th Edition, Pearson Edu, 2010
- Computer Networks: A Top-Down Approach, Behrouz A. Forouzan, FirouzMosharraf, McGraw Hill Education
- Larry L. Peterson and Bruce S. Davie, “Computer Networks – A Systems Approach” (5th ed), Morgan Kaufmann/ Elsevier, 2011
- Data Communications and Networking, Behrouz A. Forouzan, TMH
- Data and Computer Communications, William Stallings, PHI
- Computer Networks, Andrew S. Tanenbaum, PHI Version 2 CSE IIT, Kharagpur
- Internetworking with TCP/IP; Principles, Protocols, and Architecture, Douglas E. Commer, 3rd Edition, Prentice Hall of India

Introduction

- **Data**
  - digital
  - analog
- **Signal**
  - digital
  - analog
- **Data Encoding (converted into)**
  - ➢ Converting **source data** → into communication **signal**
  - digital to → digital
  - digital to → analog
  - analog to → digital
  - analog to → analog

Introduction

- **Digital data, digital signals:** simplest form of digital encoding of digital data
  - the equipment is less complex and less expensive than digital-to-analog

    Using Line coding Technique.
- **Digital data, analog signal:** A modem converts digital data to an analog signal so that it can be transmitted over an analog medium.
- **Analog data, digital signals:** Analog data, such as voice and video, are often digitized to be able to use digital transmission facilities.
- **Analog data, analog signals:** Analog data are modulated by a carrier frequency to produce an analog signal in a different frequency band, which can be utilized on an analog transmission system
  - Eg:Voice transmission

Q)What type of signal we should use?

It depends on the situation and available bandwidth

Data can be encoded in either form of signals

Analog Signals

Analog Data

Voice (sound waves) ⟷ **Telephone** ⟷ Analog Signal

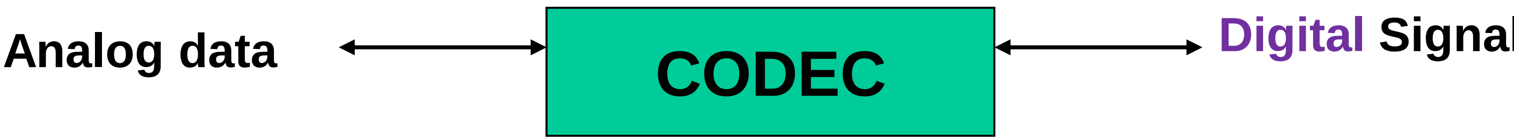
Digital Data

Binary Voltage pulses ⟷ **Modem** ⟷ Analog Signal (Carrier frequency)

Data can be encoded in either form of signals

Digital Signals

Analog Data



Digital Data

Signal Encoding Techniques

- **Digital signaling: For digital signaling, a data source $g(t)$, which may be either digital or analog, is encoded into a digital signal $x(t)$.**

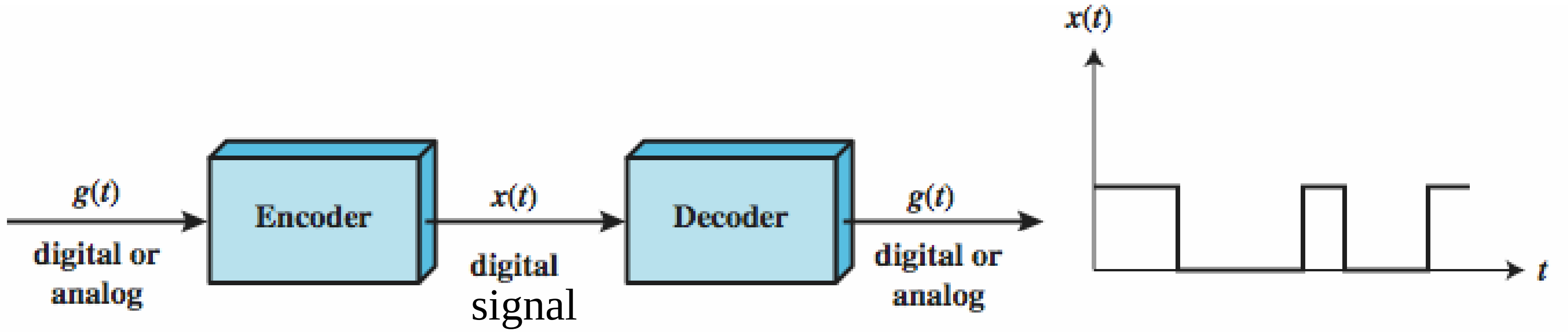
- **Analog signaling: The basis for analog signaling is a continuous constant-frequency f_c signal known as the Carrier signal. E.g. AM or FM**

- **Baseband signal: The input signal may be analog or digital and is called the modulating signal or baseband signal.**
  - Baseband signals are the fundamental group of frequencies in an analog or digital waveform that may be transmitted along a channel. Examples of a digital baseband signal may be Ethernet signals operating over a LAN

Encoding is required because

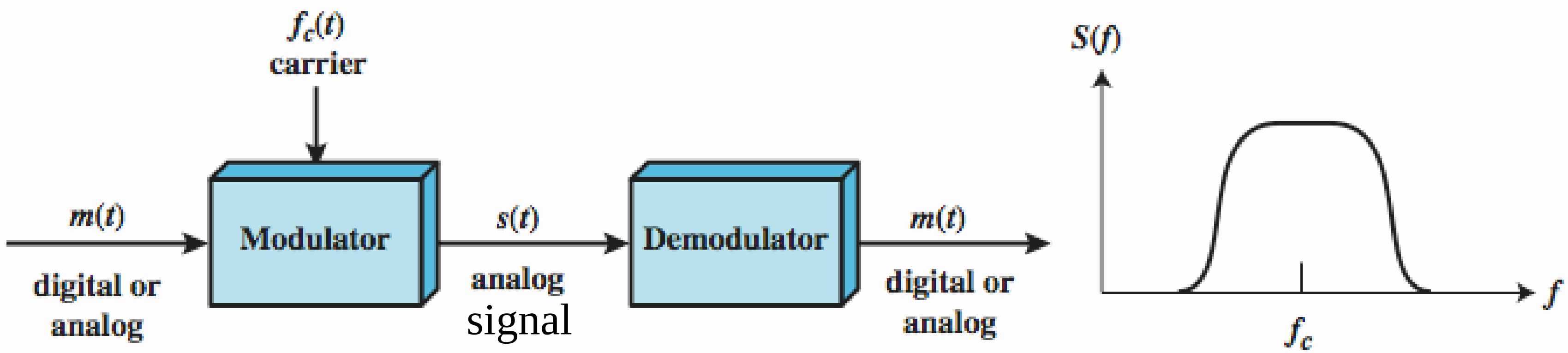
- **Signals are carried by the physical medium**
- **The performance of medium will vary depending on the kind of signal, with varying characteristics in terms of**
  - Attenuation
  - Error rate
  - Distance
  - Noise influence
  - etc….

Encoding



(a) Encoding onto a digital signal

Modulation



(b) Modulation onto an analog signal

Modulation

- Modulation is the process of encoding source data onto a carrier signal with frequency f_c.
  - —The frequency of the carrier signal is chosen to be compatible with the transmission medium being used.
  - —Modulation techniques involve operation on one or more of the three parameters: ***amplitude, frequency, and phase***
- According to the input source signal m(t) (either analog or digital), which is called baseband signal (or modulating signal) , the carrier signal $f_c(t)$ will be modulated into modulated signal s(t).

Signal Encoding Techniques

- **The electromagnetic signal is generated at the physical layer**
- **The electromagnetic signal could be analog or digital signals that must carry the data(message).**

- **The data must be encoded into signals:**
  - Data Encoding is done to produce a *analog* or *digital signal*
  - Modulation of a analog/digital data is used to produce an *analog signal*

- **This analog signal can then be transmitted over a network/communication link.**

Which one of the 4 combinations to choose?

- **Digital data/Digital signal**
  - Equipment for encoding is less complex
  - Digital transmission has less errors
- **Analog data/Digital signal**
  - Digital transmission can be done on the existing analog medium – good return on investment (ROI)
  - Digital transmission has less errors
- **Digital data/Analog signal**
  - Some high data rate mediums are analog (e.g. optical fiber)
  - most of the unguided media are analog (e.g. Wireless)
- **Analog data/Analog signal**
  - Can be transmitted easily and cheaply; different position on the spectrum can be shared on the same media (e.g. Frequency-division multiplexing)

Data Encoding

Data Encoding

- **digital data to digital signal**
- **analog data to analog signal**
- **digital data to analog signal**
- **analog data to digital signal**

Properties that need to be considered while encoding

- **At least 5 properties need to be considered when encoding any form of data to any signals**
  1. Signal spectrum requirement
  2. Signal synchronization capability
  3. Signal error-detecting capability
  4. Signal interference and noise immunity
  5. Cost and complexity of the encoding/decoding equipment.

Encoding

- **Unfortunately we will not be able to achieve all the first four properties at the cheapest cost by a single encoding technique.**
- **Hence cost is an last 5th important factor that needs to be considered as well**
- **Each encoding technique will satisfy (either fully or partially) only a sub-set of the properties.**

Some Popular Encoding Schemes of Digital data to Digital signals

Terminologies

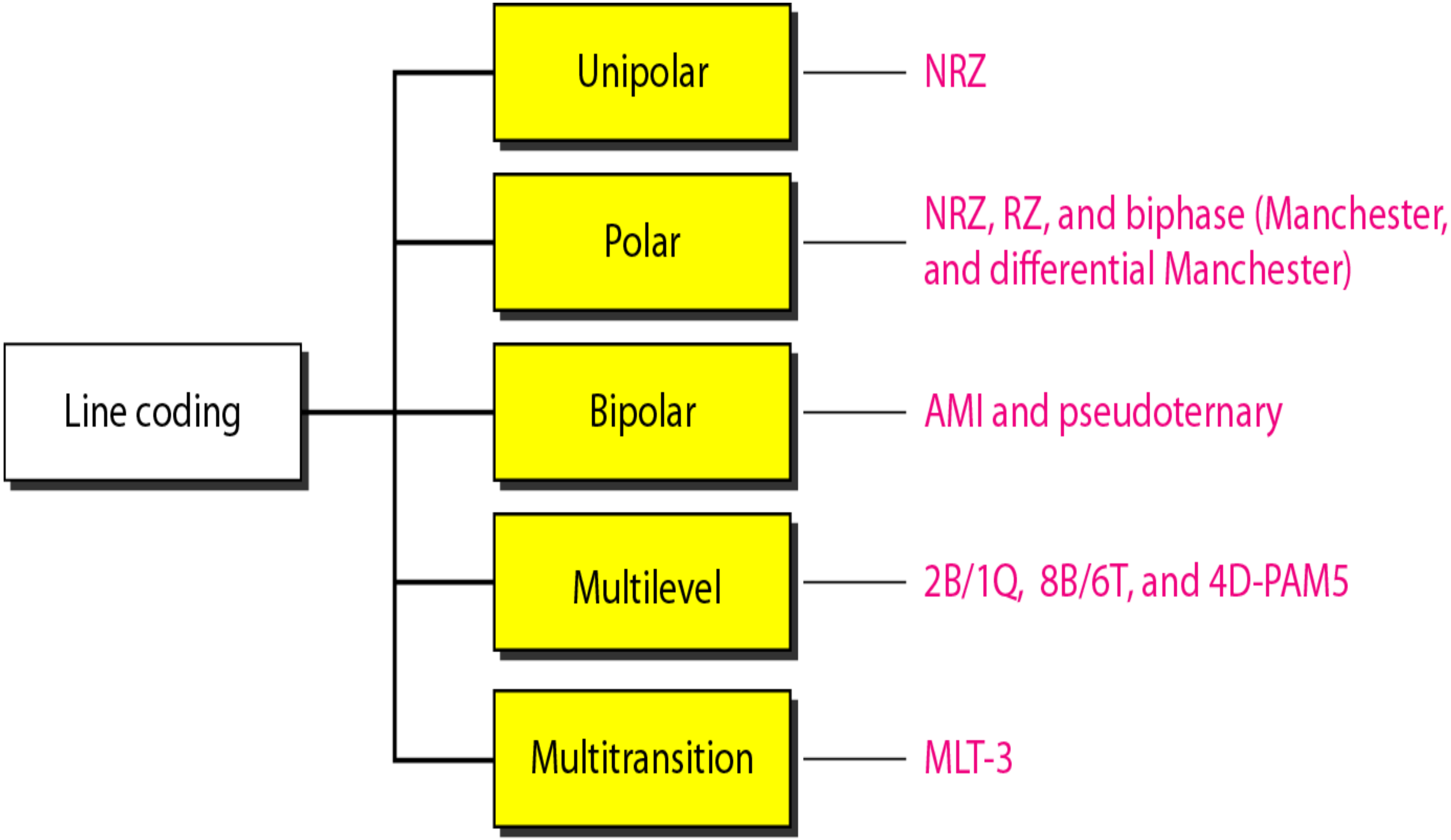
- Unipolar
  - If all signal elements have the same algebraic sign (all positive or all negative), then the signal is unipolar.
- Polar
  - One logic state represented by positive voltage, the other by negative voltage
- Data rate
  - Rate of data transmission measured in **bps**: **bits per second**
- Duration or length of a bit
  - Time taken for transmitter to emit the bit
- Modulation rate
  - Rate at which the signal level changes
  - Measured in **baud: signal elements per second**
- Mark and Space
  - Mark: Binary 1
  - Space: Binary 0

Interpreting Digital Signals

- **Receiver needs to know**
  - timing of bits - when they start and end
  - signal levels
- **Four factors affecting signal interpretation**
  - signal to noise ratio
  - data rate
  - Bandwidth
    - An increase in data rate increases bit error rate (BER).
    - An increase in SNR decreases bit error rate.
    - An increase in bandwidth allows an increase in data rate.
- Another factor that can improve performance:

  Encoding scheme: the mapping from data bits to signal elements

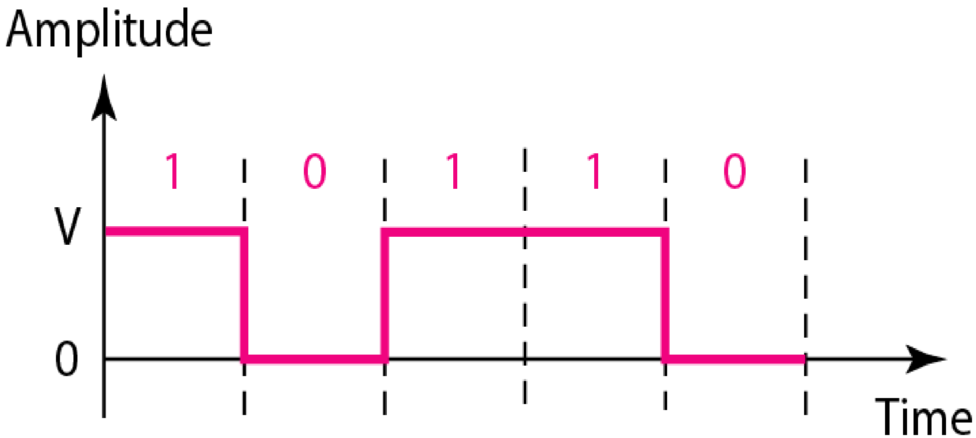
Line Coding Schemes



Unipolar

- All signal levels are on one side of the time axis - either above or below
- NRZ - Non Return to Zero scheme is an example of this code. The signal level does not return to zero during a symbol transmission.
- Scheme is prone to baseline wandering and DC components. It has no synchronization or any error detection. It is simple but costly in power consumption.

Unipolar NRZ scheme



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

Normalized power

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below. In ***Non-Return-to-Zero,*** the signal does not return to zero at the middle of the bit, *where* positive voltage defines bit 1 and the zero voltage defines bit 0. Costly. the normalized power (the power needed to send 1 bit per unit line resistance) is double that for polar NRZ.

<u>Disadvantage:</u> DC Component and Synchronization.

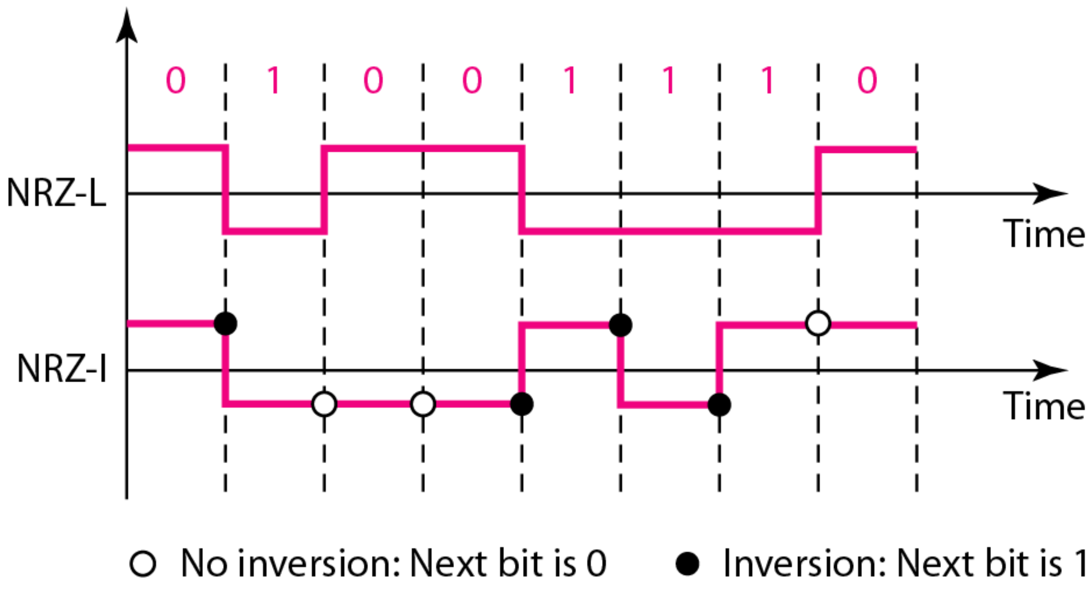
Polar - NRZ

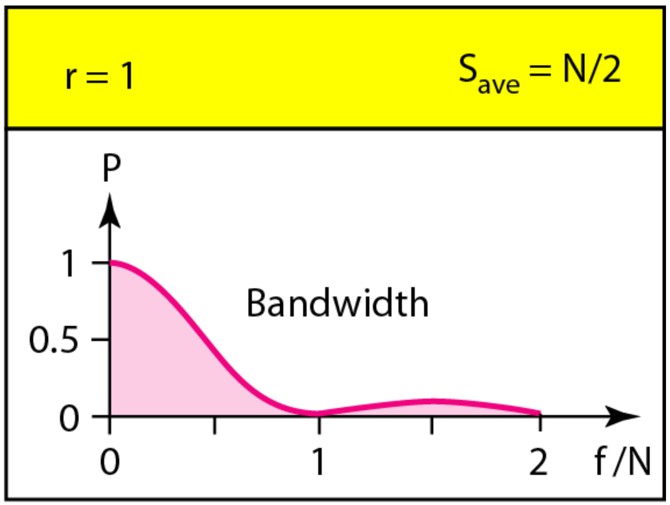
- The voltages are on both sides of the time axis.

- Polar NRZ scheme can be implemented with two voltages. E.g. +V for 1 and -V for 0.

- There are two versions:
  - NZR - Level (NRZ-L) - positive voltage for one symbol and negative for the other
  - NRZ - Inversion (NRZ-I) - the change or lack of change in polarity determines the value of a symbol. E.g. a "1" symbol inverts the polarity a "0" does not.

Polar NRZ-L and NRZ-I schemes





Polar NRZ-L and NRZ-I schemes

- Non-Return-to-Zero (NRZ) with L (Level) and I (Invert).
- In NRZ-L the level of the voltage determines the value of the bit.
- In NRZ-I the inversion or the lack of inversion determines the value of the bit. If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed.
- In NRZ-I this problem occurs only for a long sequence of 0s. The synchronization problem.
- Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. NRZ-L and NRZ-I both have an average signal rate of N/2 Bd. NRZ-L and NRZ-I both have a DC component problem.

Polar - NRZ

- In NRZ-L the level of the voltage determines the value of the bit. In NRZ-I the inversion or the lack of inversion determines the value of the bit.

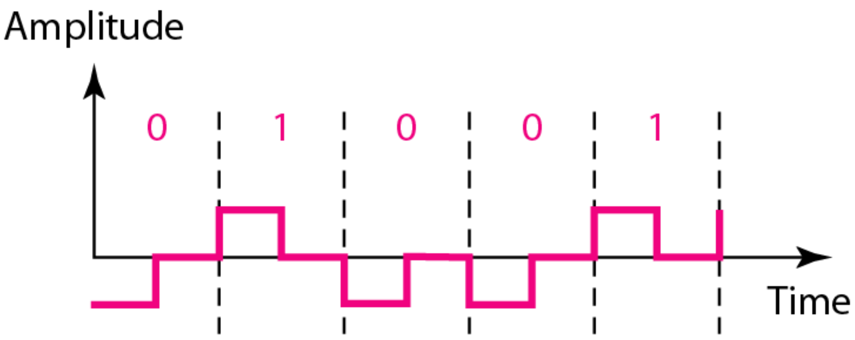
- NRZ-L and NRZ-I both have an average signal rate of N/2 Bd.

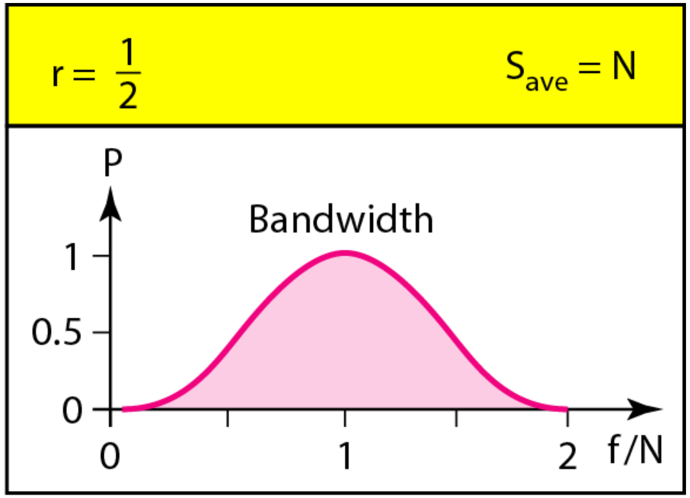
- NRZ-L and NRZ-I both have a DC component problem and baseline wandering, it is worse for NRZ-L. Both have no self synchronization &no error detection. Both are relatively simple to implement.

Polar - RZ

- The Return to Zero (RZ) scheme uses three voltage values. +, 0, -.
- Each symbol has a transition in the middle. Either from high to zero or from low to zero.
- This scheme has more signal transitions (two per symbol) and therefore requires a wider bandwidth.
- No DC components or baseline wandering.
- Self synchronization - transition indicates symbol value.
- More complex as it uses three voltage level. It has no error detection capability.

Polar - RZ

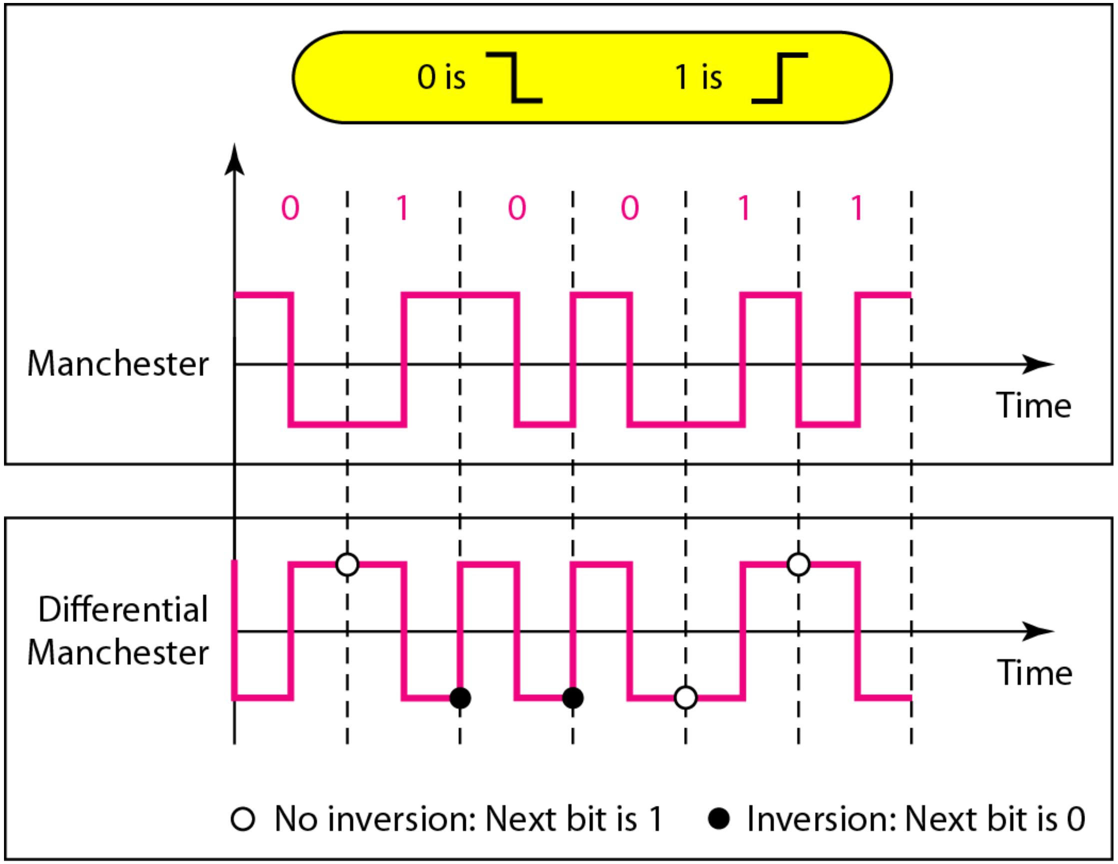


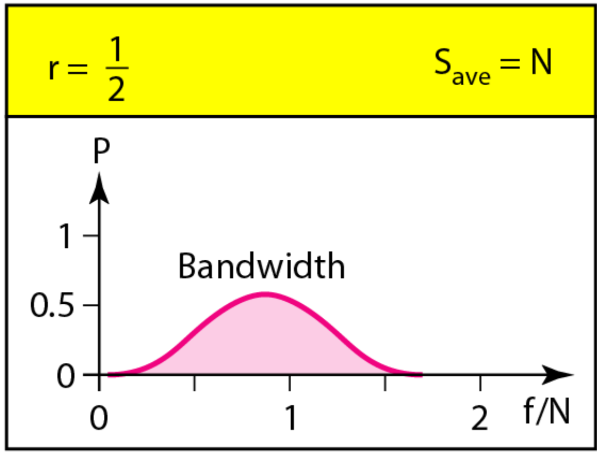


Polar - Biphase: Manchester and Differential Manchester

- Manchester coding consists of combining the NRZ-L and RZ schemes.
  - Every symbol has a level transition in the middle: from high to low or low to high. Uses only two voltage levels.
- Differential Manchester coding consists of combining the NRZ-I and RZ schemes.
  - Every symbol has a level transition in the middle. But the level at the beginning of the symbol is determined by the symbol value. One symbol causes a level change the other does not.

Polar biphase: Manchester and Differential Manchester schemes





Polar biphase: Manchester and Differential Manchester schemes

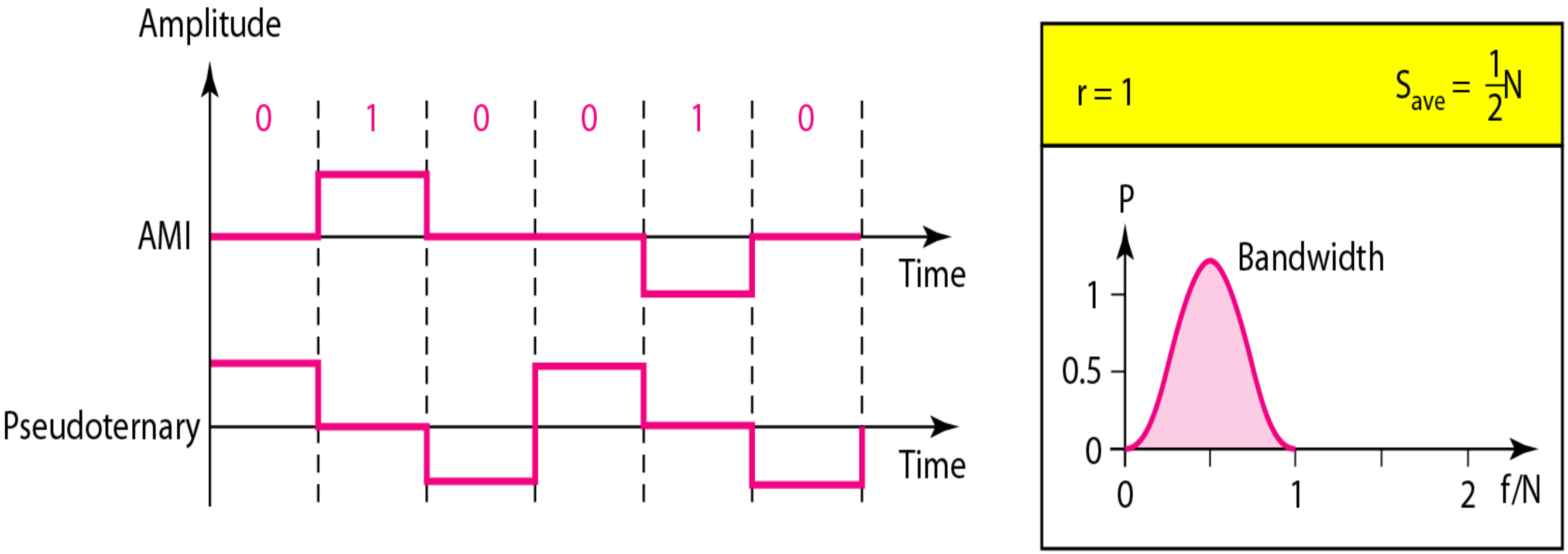
- **In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.**

- **The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.**

Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at “0” and the other alternates between + & -.
- Bipolar Alternate Mark Inversion (AMI) - the “0” symbol is represented by zero voltage and the “1” symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Bipolar schemes: AMI and Pseudoternary



Bipolar C/Cs

- It is a better alternative to NRZ.
- Has no DC component or baseline wandering.
- Has no self synchronization because long runs of "0"s results in no signal transitions.
- No error detection.

Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at “0” and the other alternates between + & -.
- Bipolar Alternate Mark Inversion (AMI) - the “0” symbol is represented by zero voltage and the “1” symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Multilevel Schemes

- In these schemes we increase the number of data bits per symbol thereby increasing the bit rate.
- Since we are dealing with binary data we only have 2 types of data element a 1 or a 0.
- We can combine the 2 data elements into a pattern of “m” elements to create “2m” symbols.
- If we have L signal levels, we can use “n” signal elements to create Ln signal elements.

Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels: - +, 0, -, to represent the symbols (note not transitions to zero as in RZ).
- Voltage level for one symbol is at “0” and the other alternates between + & -.
- Bipolar Alternate Mark Inversion (AMI) - the “0” symbol is represented by zero voltage and the “1” symbol alternates between +V and -V.
- Pseudoternary is the reverse of AMI.

Code C/Cs

- Now we have 2m symbols and Ln signals.
- If 2m > Ln then we cannot represent the data elements, we don’t have enough signals.
- If 2m = Ln then we have an exact mapping of one symbol on one signal.
- If 2m < Ln then we have more signals than symbols and we can choose the signals that are more distinct to represent the symbols and therefore have better noise immunity and error detection as some signals are not valid

- In mBnL schemes, a pattern of m data elements is encoded as a pattern of n signal elements in which $2m \leq Ln$.

Representing Multilevel Codes

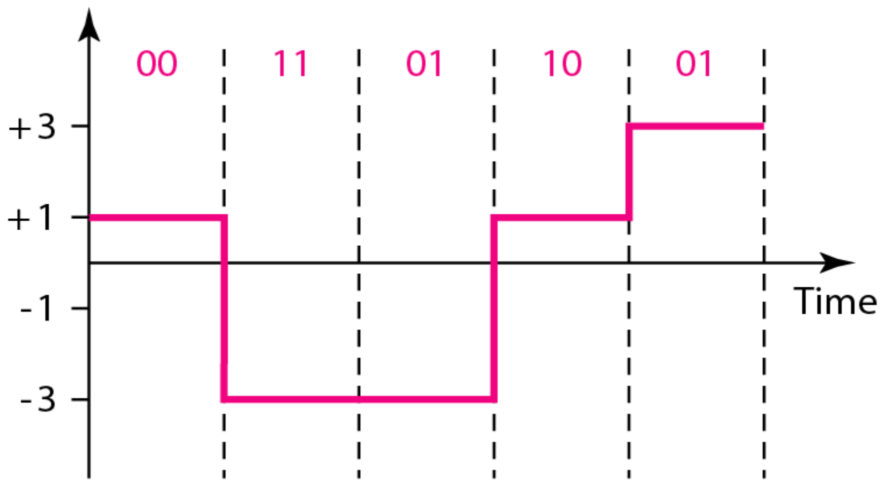
- We use the notation mBnL, where m is the length of the binary pattern, B represents binary data, n represents the length of the signal pattern and L the number of levels.

- L = B binary, L = T for 3 ternary, L = Q for 4 quaternary.

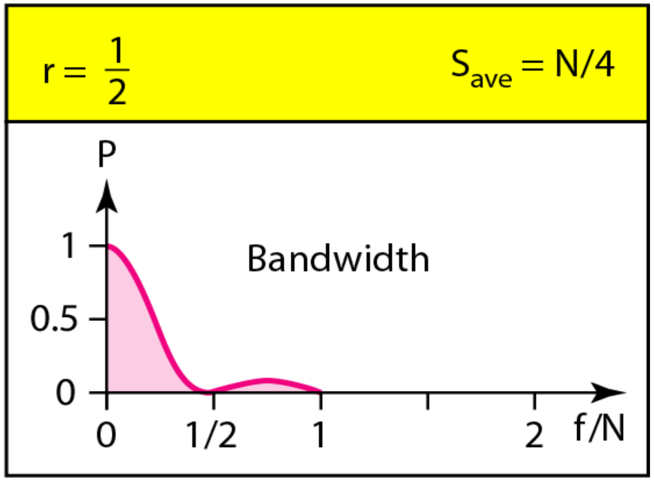
Multilevel: 2B1Q scheme

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

Transition table

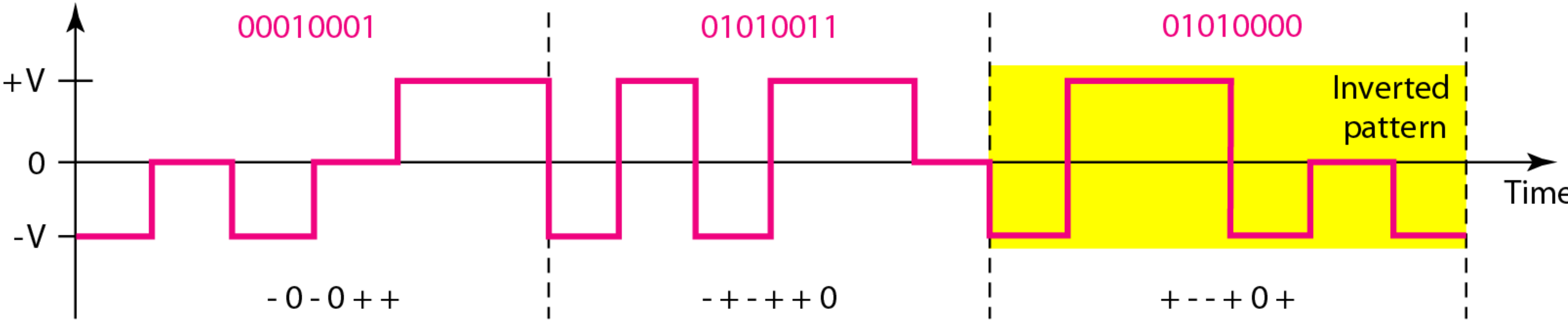


Assuming positive original level



Multilevel: 8B6T scheme

- The **eight binary, six ternary (8B6T)** is used with 100BASE-4T cable.
- Signal has three levels (ternary) $2^8 = 256$ different data patterns and $3^6 = 729$ different signal patterns.
- There are 729 - 256 = 473 redundant signal elements that provide synchronization, error detection and provide DC balance.
- The first 8-bit pattern 00010001 is encoded as the signal pattern - 0 - 0 + + with weight 0; the second 8-bit pattern 01010011 is encoded as - + - + + 0 with weight +1. The third 8-bit pattern 01010000 should be encoded as + - - + 0 + with weight +1.
- The receiver can easily recognize that this is an inverted pattern because the weight is -1.



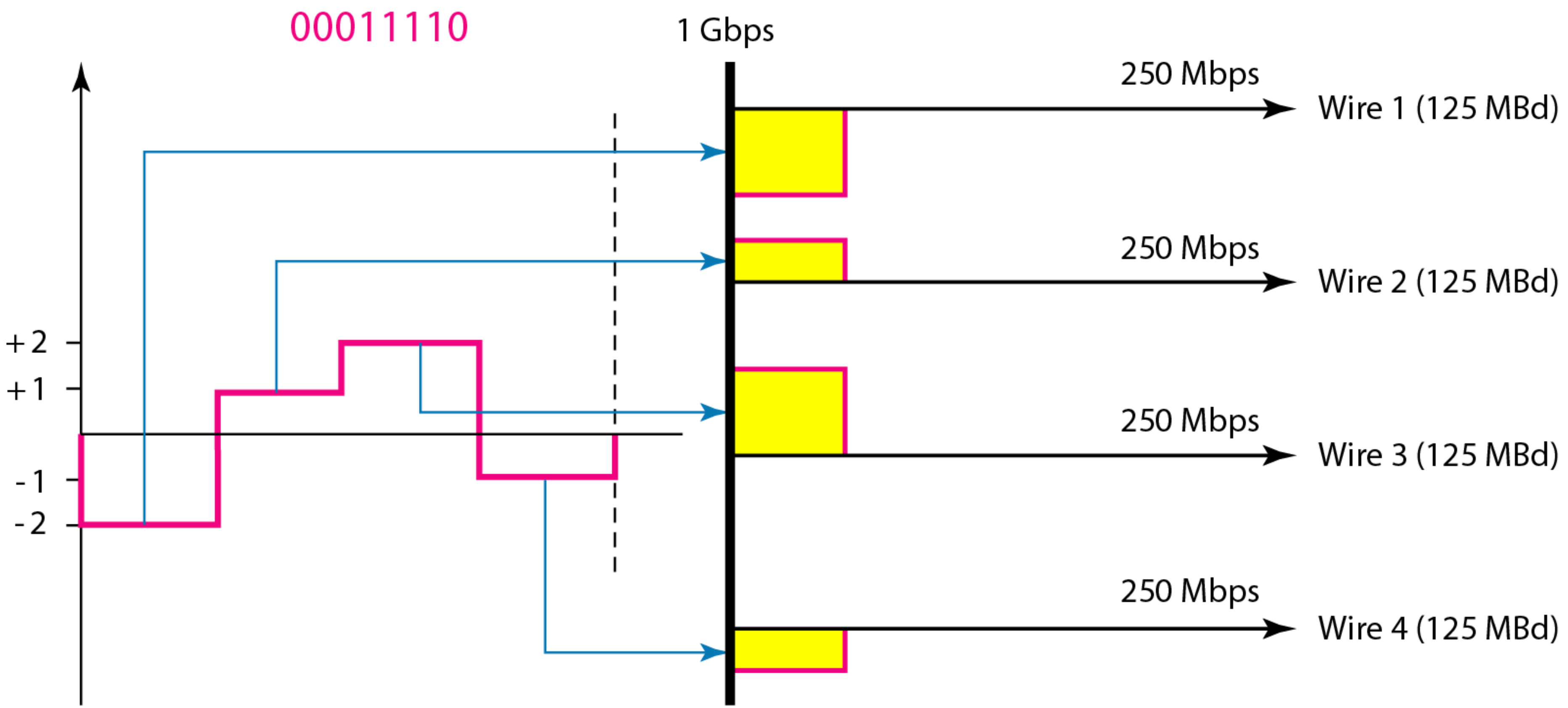
4D-PAM5

- Four-dimensional five level pulse amplitude modulation (4D-PAM5)
- The 4D means that data is sent over four wires at the same time. It uses five voltage levels, such as -2, -1, 0, 1, and 2.
- However, one level, level 0, is used only for forward error detection.
- Gigabit LANs use this technique to send 1-Gbps data over four copper cables that can handle 125 Mbaud.
- The extra signal patterns can be used for other purposes such as error detection.

4D-PAM5

- Four-dimensional five level pulse amplitude modulation (4D-PAM5)
- The 4D means that data is sent over four wires at the same time. It uses five voltage levels, such as -2, -1, 0, 1, and 2.
- However, one level, level 0, is used only for forward error detection.
- Gigabit LANs use this technique to send 1-Gbps data over four copper cables that can handle 125 Mbaud.
- The extra signal patterns can be used for other purposes such as error detection.

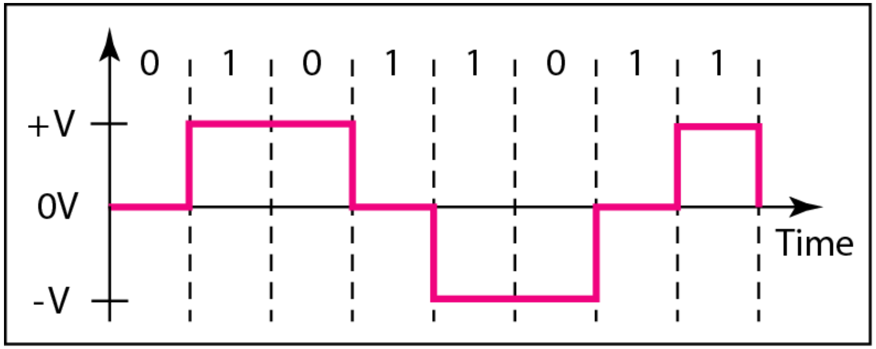
Multilevel: 4D-PAM5 scheme



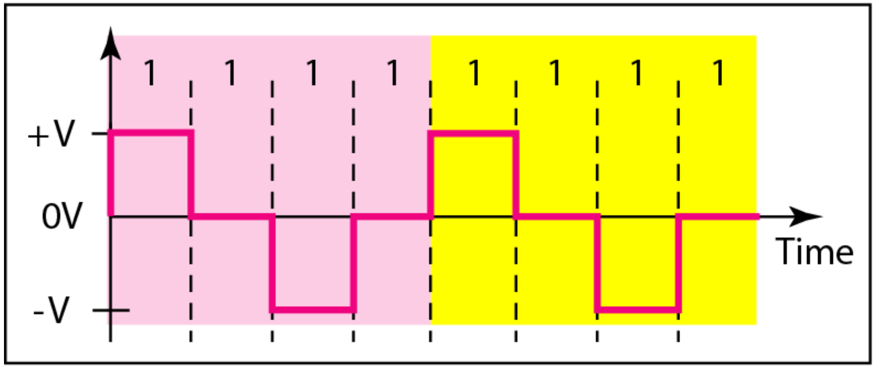
Multitransition: MLT-3

- 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.
- The three voltage levels (-*V*, 0, and +*V*) are shown by three states (ovals).
- It turns out that the shape of the signal in this scheme helps to reduce the required bandwidth.
- MLT-3 a suitable choice when we need to send 100 Mbps on a copper wire that cannot support more than 32 MHz.
- 1 = level change. 0 = no change.

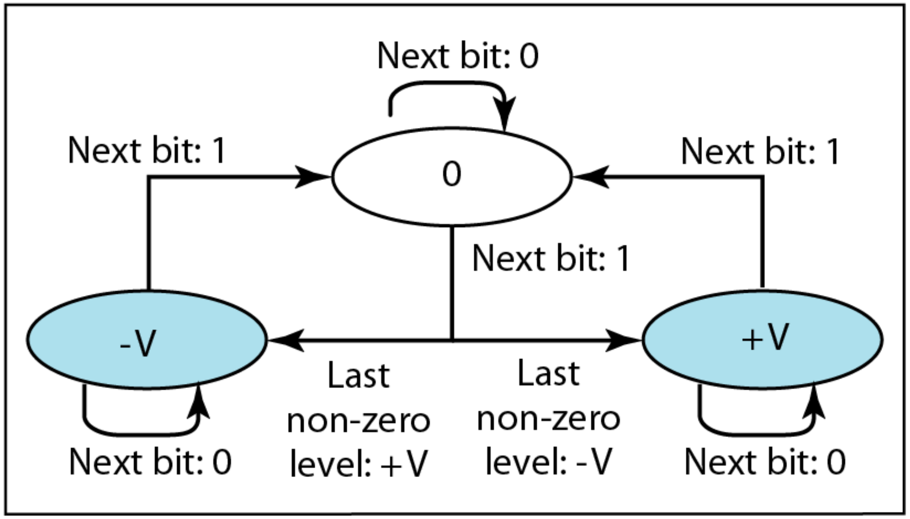
Multitransition: MLT-3 scheme



a. Typical case



b. Worse case



c. Transition states

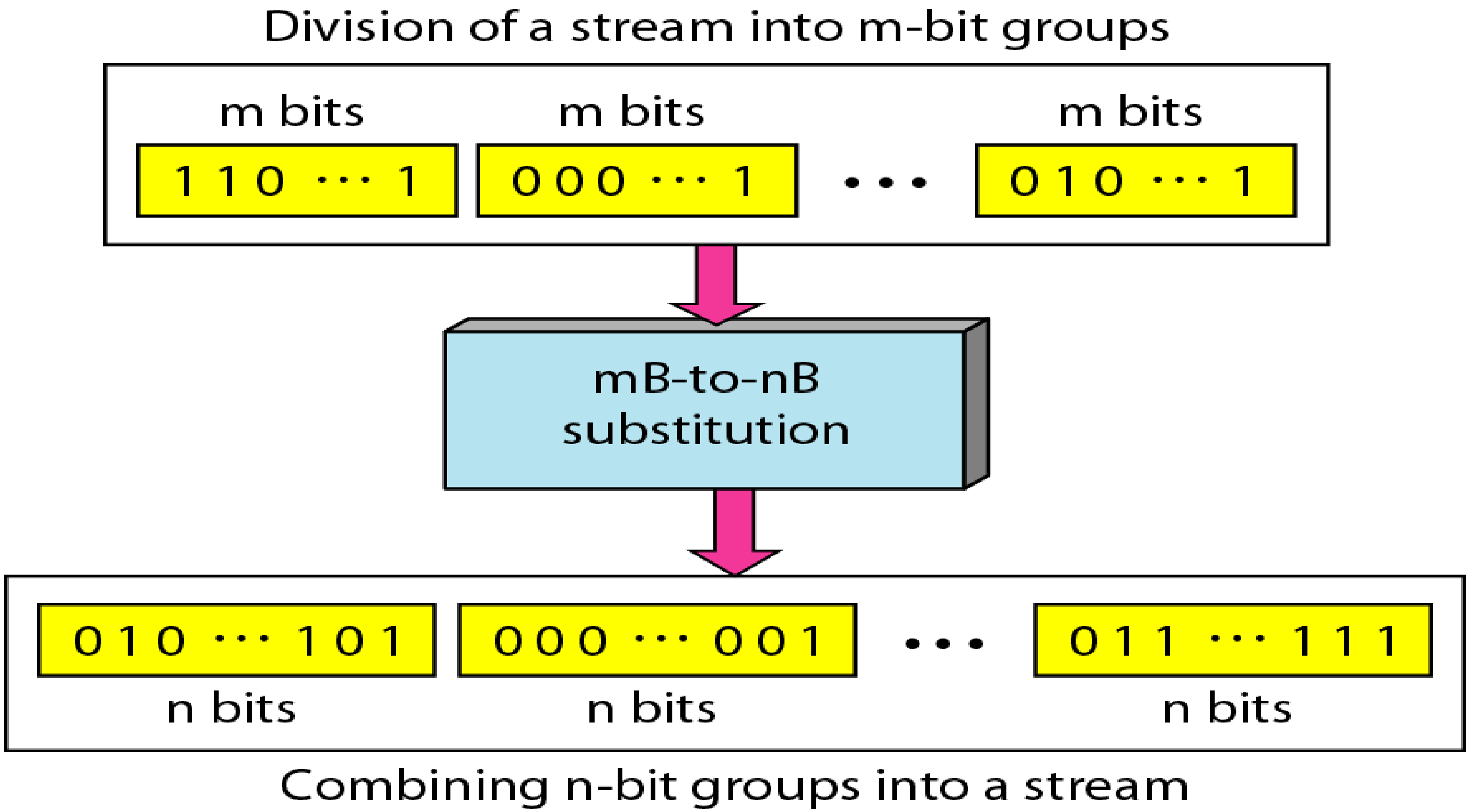
Summary of line coding schemes

Category	*Scheme*	*Bandwidth (average)*	*Characteristics*
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

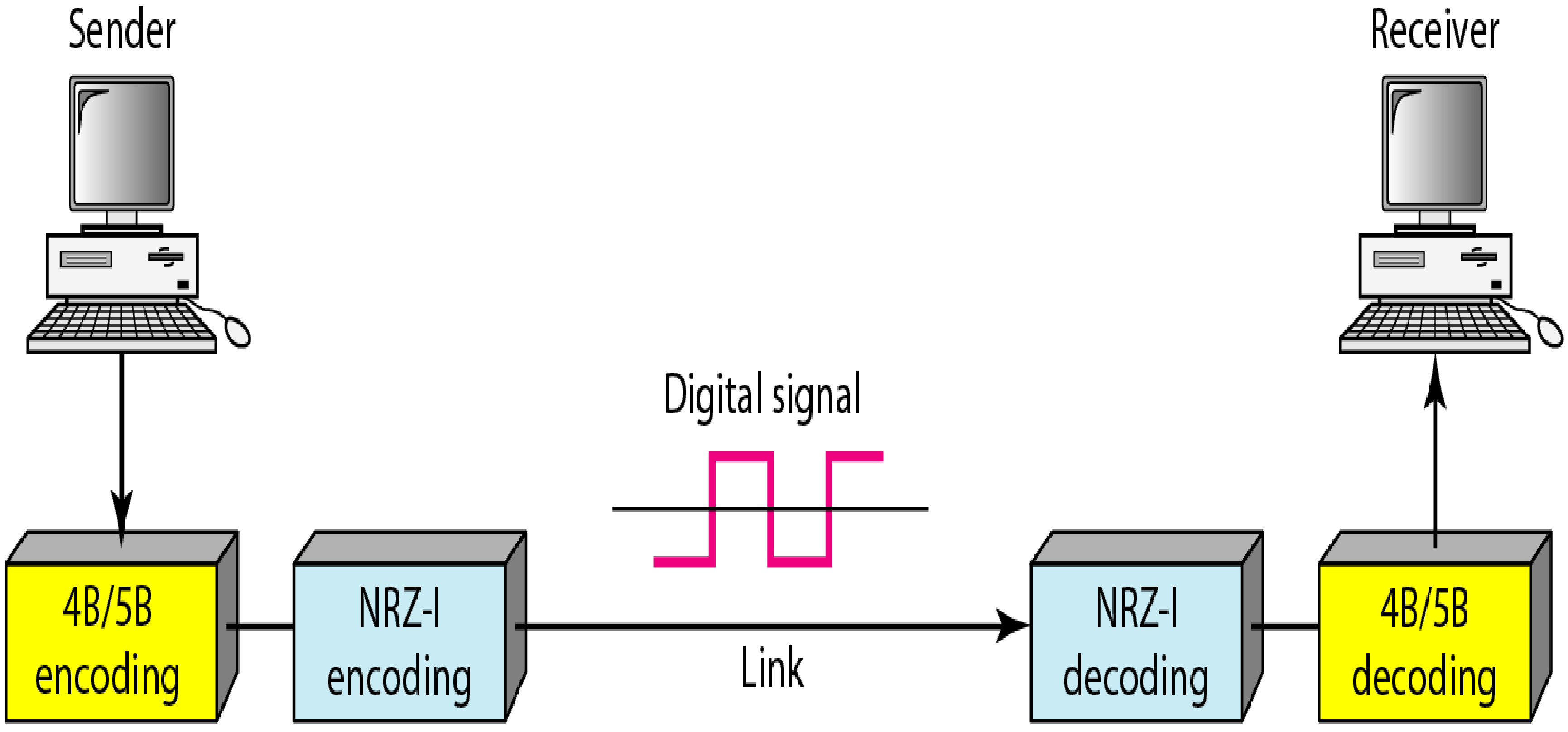
Block Coding

Block coding is normally referred to as *m*B/*n*B coding;
it replaces each *m*-bit group with an *n*-bit group.

Block coding concept



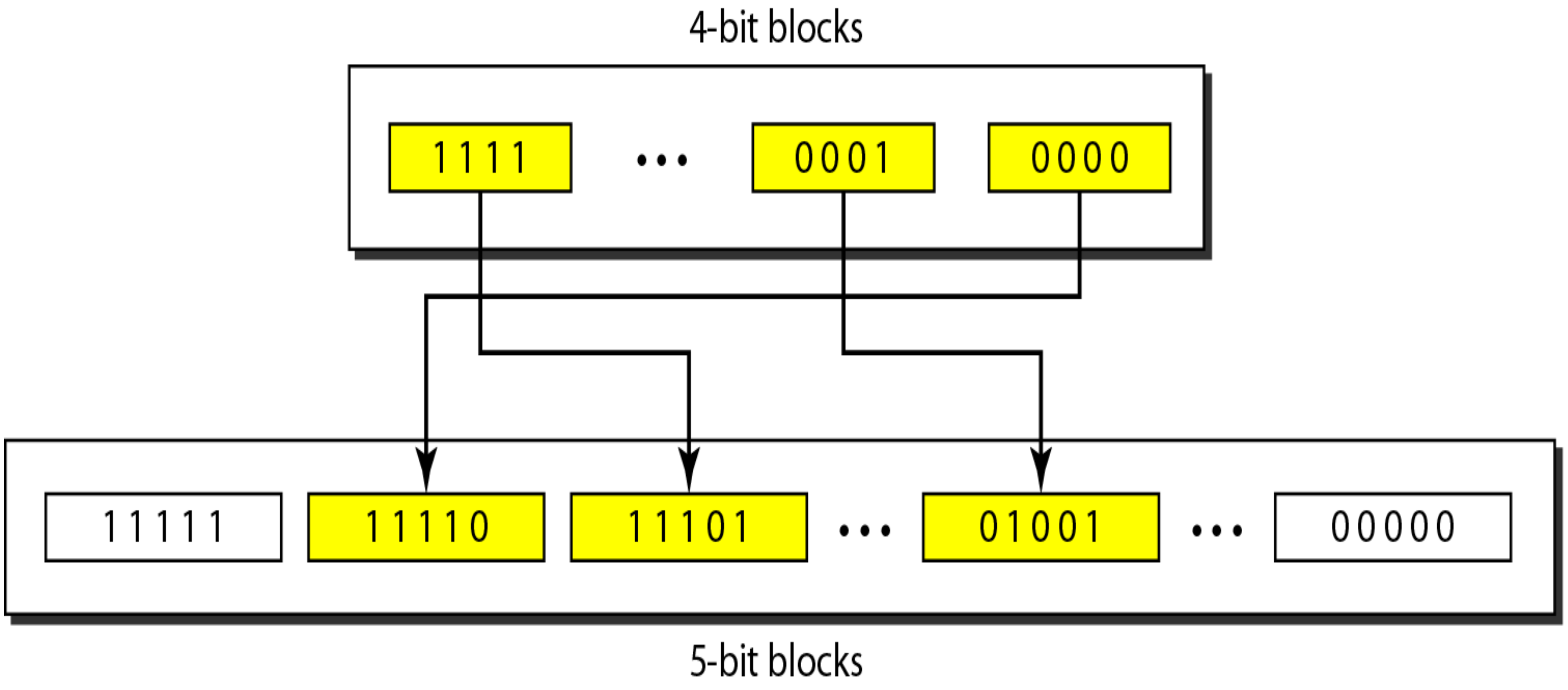
Using block coding 4B/5B with NRZ-I line coding scheme



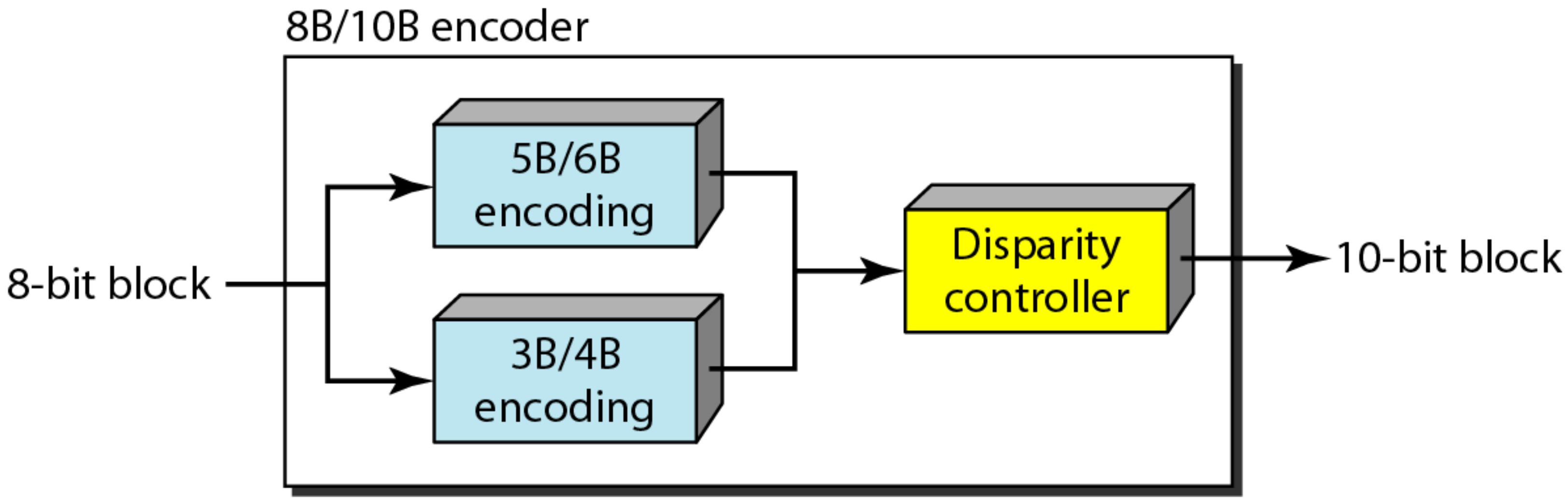
4B/5B mapping codes

Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

Substitution in 4B/5B block coding



8B/10B block encoding



More bits - better error detection

- The 8B10B block code adds more redundant bits and can thereby choose code words that would prevent a long run of a voltage level that would cause DC components.

Redundancy

- A 4 bit data word can have 24 combinations.
- A 5 bit word can have 25=32 combinations.
- We therefore have 32 - 26 = 16 extra words.
- Some of the extra words are used for control/signalling purposes.

Scrambling

- We are looking for a technique that does not increase the number of bits and does provide synchronization.
- We are looking for a solution that substitutes long zero-level pulses with a combination of other levels to provide synchronization.
- One solution is called **scrambling.**
- It is done at the same time when encoding.
- Two common scrambling techniques are B8ZS and HDB3.

Scrambling

- Use scrambling to replace sequences that would produce constant voltage
- Main idea:
  - Sequences that would result in a constant voltage are replaced by filling sequences that will provide sufficient transitions for the receiver's clock to maintain synchronization.
  - Filling sequences must be recognized by receiver and replaced with original data sequence.
  - Filling sequence is the same length as original sequence.
- Design goals:
  - No dc component
  - No long sequences of zero-level line signals
  - No reduction in data rate
  - Error detection capability

Scrambling

- Two common scrambling techniques are B8ZS and HDB3.
- **Bipolar with 8-zero substitution (B8ZS)**: In this technique, eight consecutive zero-level voltages are replaced by the sequence **000VB0VB.**
- **High-density bipolar 3-zero (HDB3)** : Two rules

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.

B8ZS

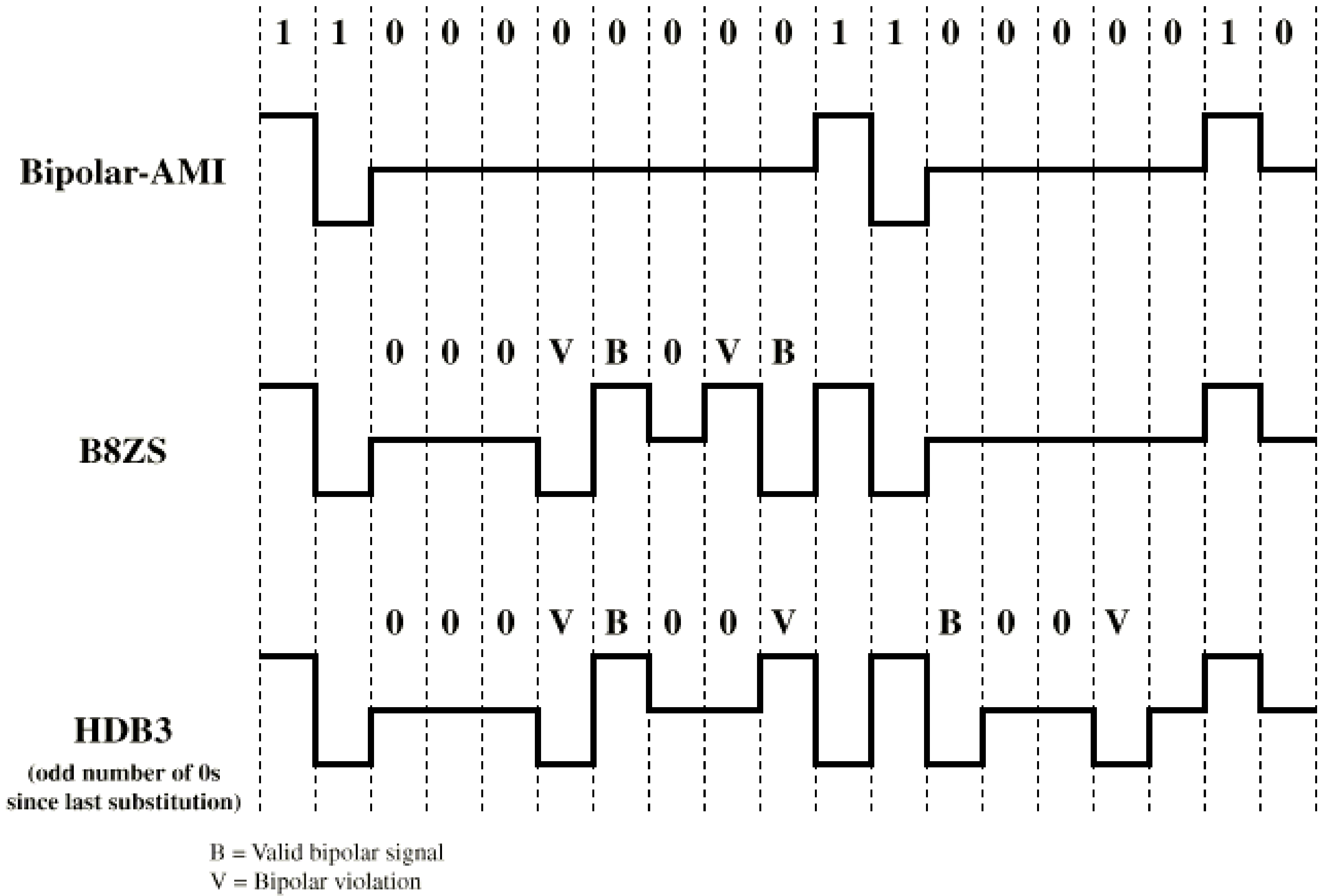
- Bipolar With 8-Zeros Substitution
- Based on bipolar-AMI, whose drawback is a long string of zeros may result in loss of synchronization.
- If octet of all zeros occurs and the last voltage pulse preceding this octet was positive, encode as 000+-0-+
- If octet of all zeros occurs and the last voltage pulse preceding this octet was negative, encode as 000-+0+-
- Causes two violations of AMI code
- Unlikely to occur as a result of noise
- Receiver recognizes the pattern and interprets the octet as consisting of all zeros.

HDB3

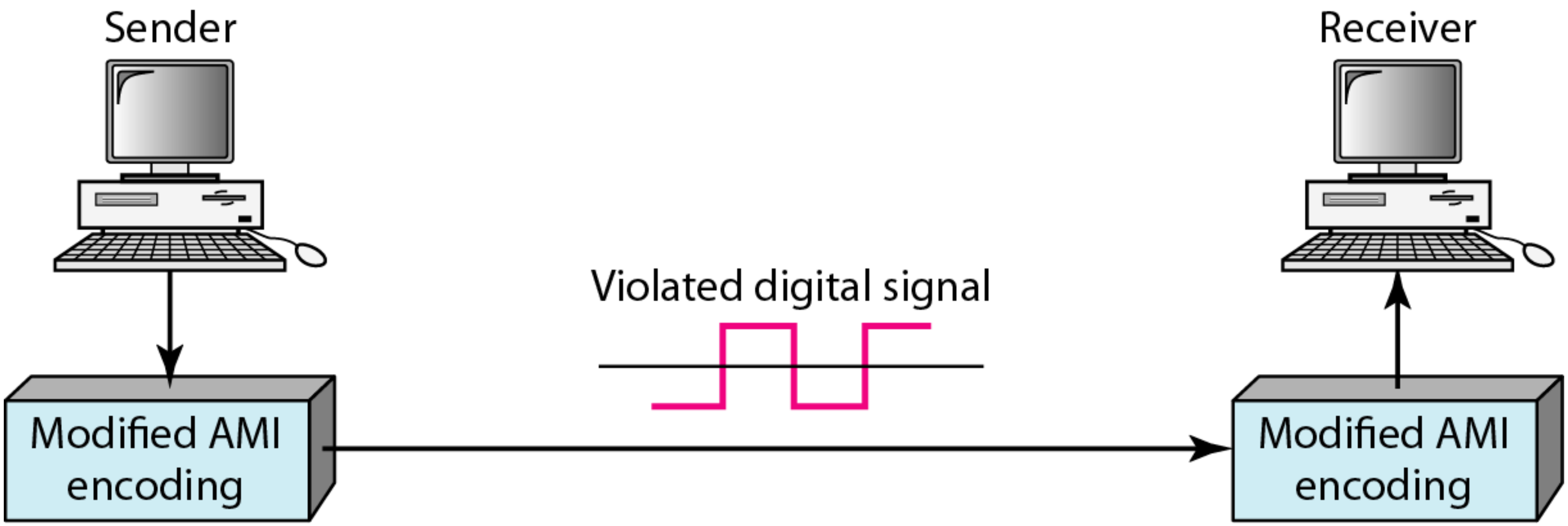
- High-Density Bipolar-3 Zeros
- Based on bipolar-AMI
- String of four zeros is replaced with sequences containing one or two pulses.

	Number of Bipolar Pulses since last substitution	
Polarity of Preceding Pulse	Odd	Even
-	000-	+00+
+	000+	-00-

B8ZS and HDB3



AMI used with scrambling



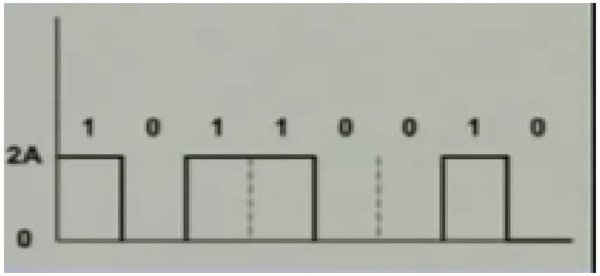
- **For example: B8ZS substitutes eight consecutive zeros with 000VB0VB.**
- **The V stands for violation, it violates the line encoding rule**
- **B stands for bipolar, it implements the bipolar line encoding rule**

Comparison of Encoding Schemes

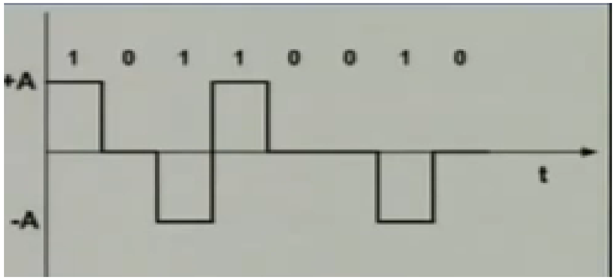
Ways of evaluating or comparing the various encoding techniques:

1.Signal Spectrum - Lack of high frequencies reduces required bandwidth, lack of dc component provide isolation.

2.Number of signal levels: two levels (for binary) , or multilevel



Two data levels, two signal levels



Two data levels , three signal levels

3.Clocking - need for synchronizing transmitter and receiver either with an external clock or with a sync mechanism based on signal

4.Error detection - useful if can be built in to signal encoding

5.Signal interference and noise immunity - some codes are better than others

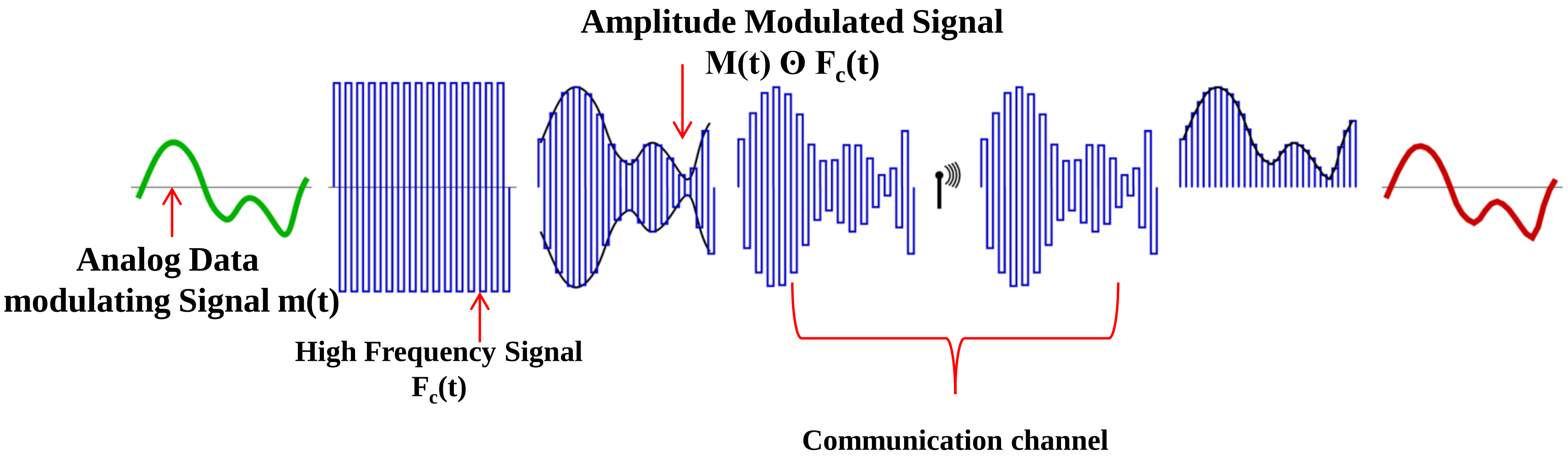
6.Cost and complexity - Higher signal rate (& thus data rate) lead to higher costs,

Data Encoding

- ✓ digital data to digital signal
- **analog data to analog signal**
- digital data to analog signal
- analog data to digital signal

Modulation

- is a process of encoding an analog (or digital) data on to an analog carrier signal whose frequency, say is f_c Where f_c is a high frequency
- Input – analog (digital) data – called baseband or modulating signal m(t)
- Output - analog signal - carried modulated signal s(t)



Modulation

Modulation:

The process by which some characteristics of the carrier, ie(amplitude/frequency/phase) is varied in accordance with the instantaneous value of the modulating signal

- **main use is public telephone system**
  - has freq range of 300Hz to 3400Hz
  - use modem (modulator-demodulator)
- **encoding techniques**
  - Amplitude shift keying (ASK)
  - Frequency shift keying (FSK)
  - Phase shift keying (PSK)

Modulation Techniques

- **is to vary one of the following three aspects of the carrier signal**
- **Amplitude of the carrier (Amplitude modulation (AM))**
- **Frequency of the carrier (Frequency modulation (FM))**
- **Phase of the carrier (Phase modulation(PM))**
- **Example – AM and FM radio stations for the first two respectively**

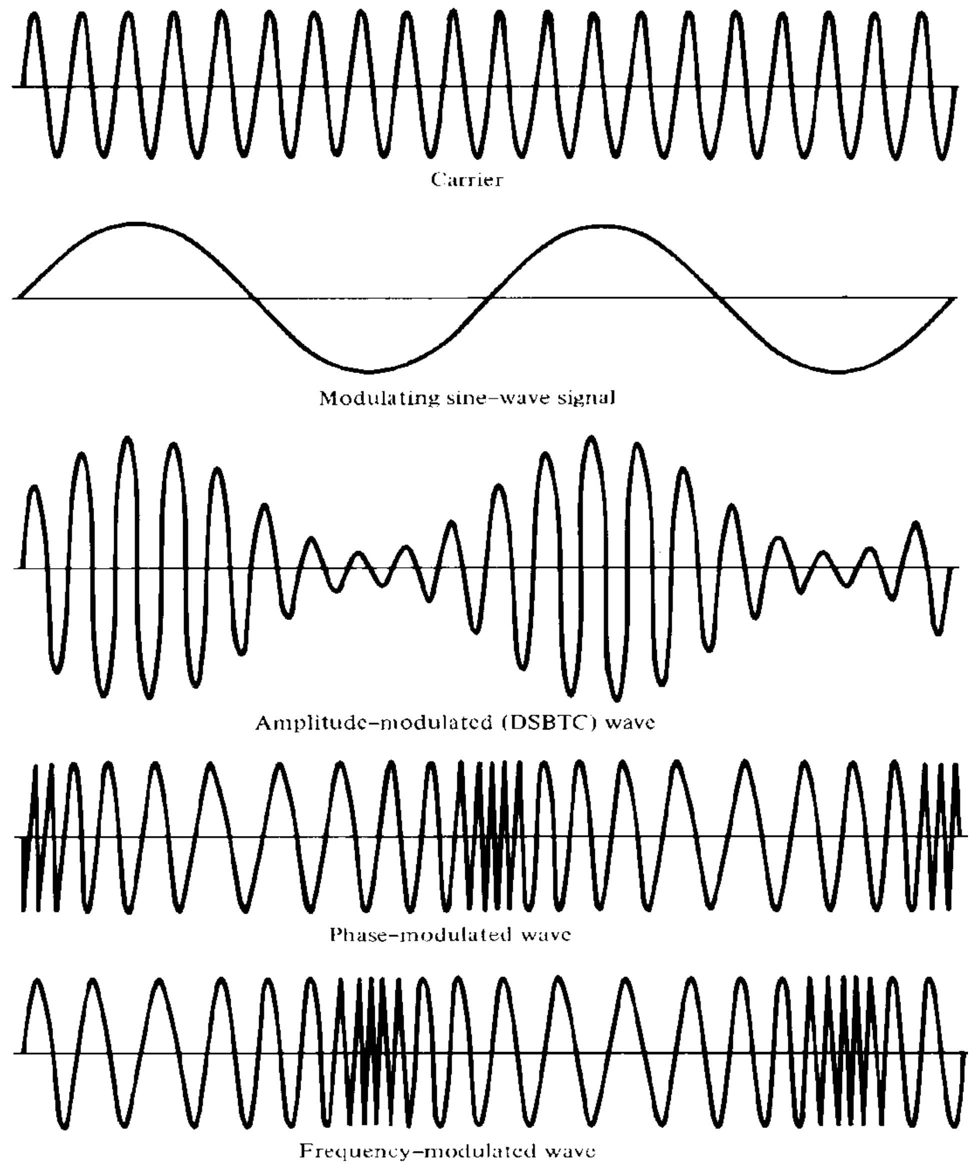
Modulation Techniques

Analog Modulation Techniques

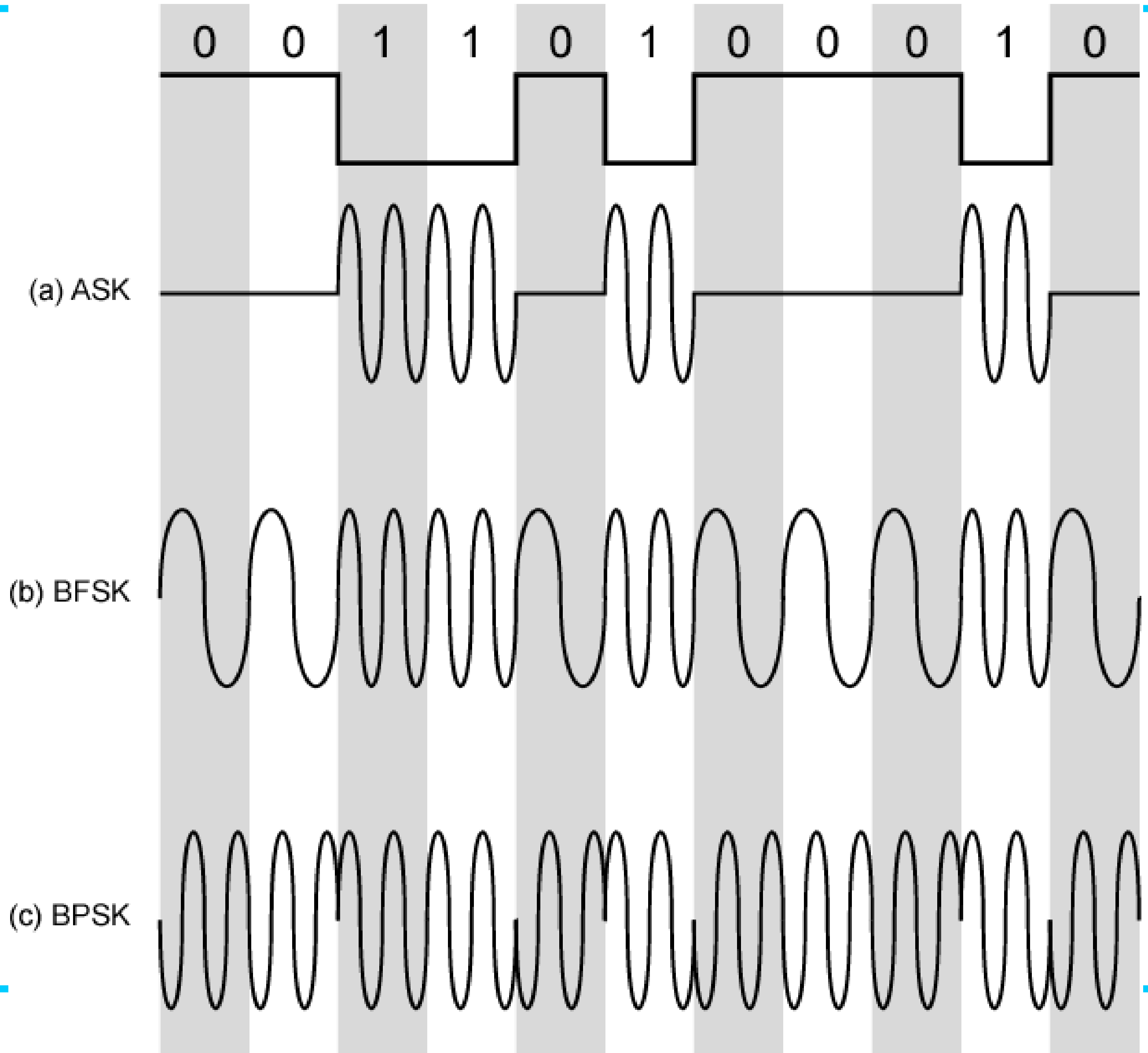
Amplitude Modulation

Phase Modulation

Frequency Modulation



Modulation Techniques



Data Encoding

- ✓digital to digital
- ✓analog to analog
- **digital to analog**
- analog to digital

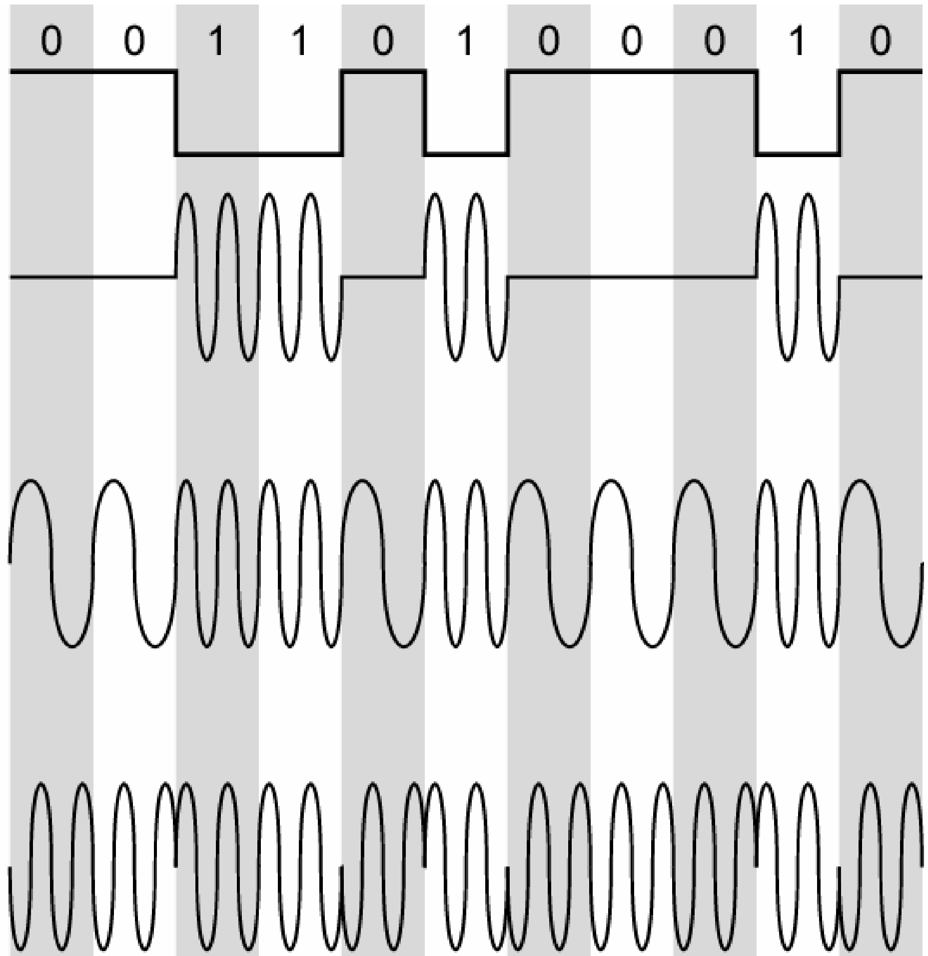
Digital Data, Analog Signal – Modulation Techniques

Digital Data

Amplitude Shift keying

Frequency Shift keying

Phase Shift keying



Amplitude Shift Keying

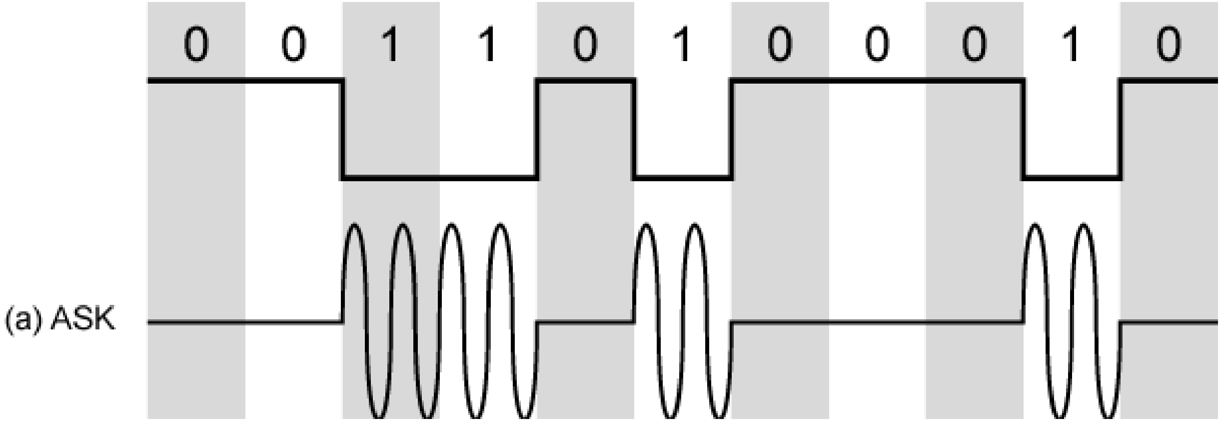
- **encode 0/1 by different carrier amplitudes**
  - usually have one amplitude zero
- **susceptible to sudden gain changes**
- **inefficient**
- **used for**
  - up to 1200bps on voice grade lines
  - very high speeds over optical fiber

$s(t) = A * \sin(2\pi f t + \theta)$

A = Amplitude

f = carrier frequency

θ = Phase Angle



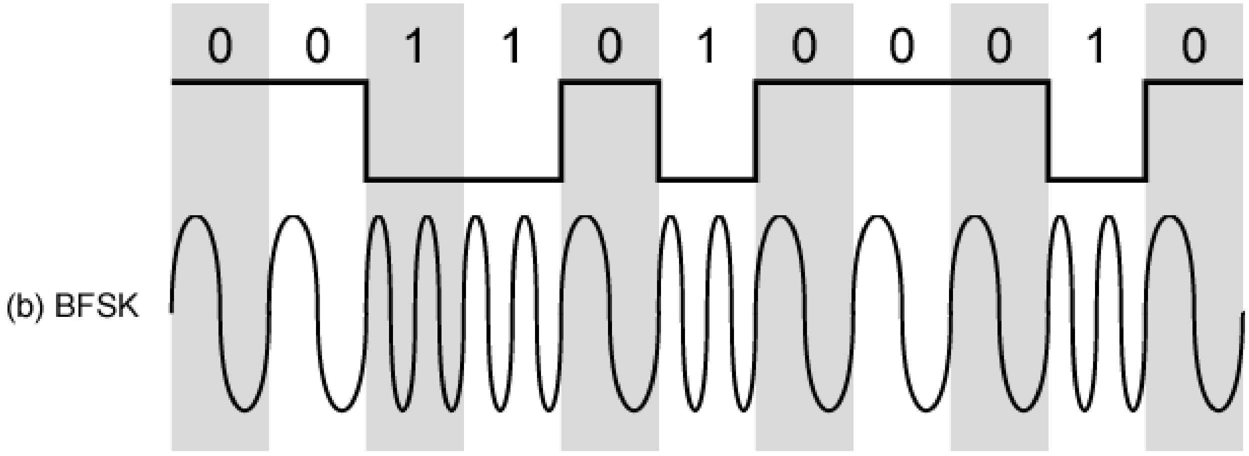
"0" $= 0 * \sin(2\pi f t + 0)$

"1" $= A * \sin(2\pi f t + 0)$

Binary Frequency Shift Keying

- **most common is binary FSK (BFSK)**
- **two binary values represented by two different frequencies (near carrier)**
- **less susceptible to error than ASK**
- **used for**
  - up to 1200bps on voice grade lines
  - high frequency radio
  - even higher frequency on LANs using co-axial cable

$s(t) = A * \mathrm{Sin}\,(2\pi f\, t + \theta)$
A = Amplitude
f = carrier frequency
θ = Phase Angle



“0” $= A * \mathrm{Sin}\,(2\pi f_1\, t + 0)$

“1” $= A * \mathrm{Sin}\,(2\pi f_2\, t + 0)$

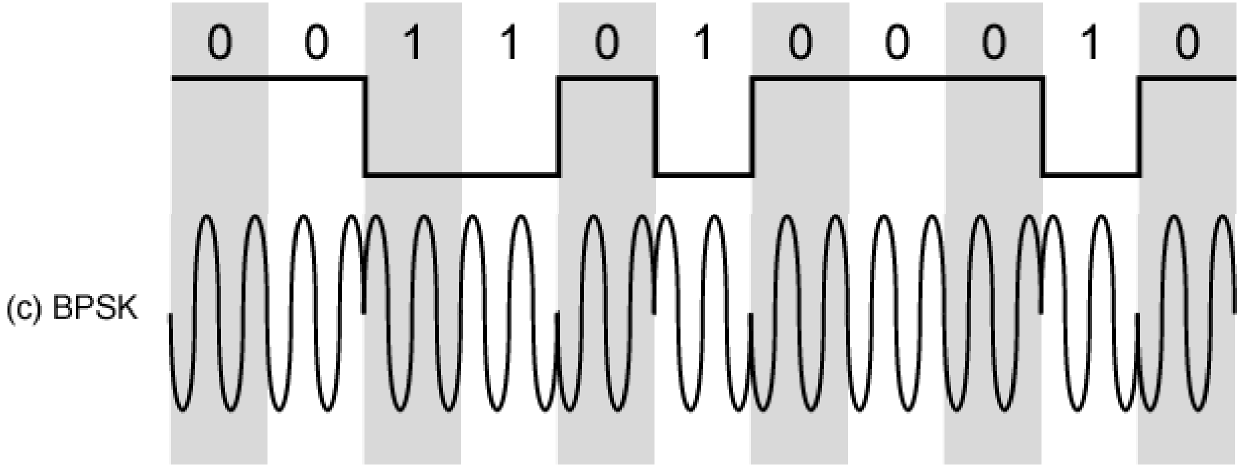
Multiple FSK

- **Each signalling element can also represent more than one bit**
  - then more than two frequencies are required (e.g:2 bits – 4 frequencies for 00, 01, 10 and 11)
- **More bandwidth efficient**
- **However more prone to error**

Phase Shift Keying

- **phase of carrier signal is shifted to represent data**
- **binary PSK**
  - two phases represent two binary digits
- **differential PSK**
  - phase shifted relative to previous transmission rather than some reference signal

$s(t) = A * \text{Sin}\,(2\pi f\, t + \theta)$
A = Amplitude
f = carrier frequency
θ = Phase Angle

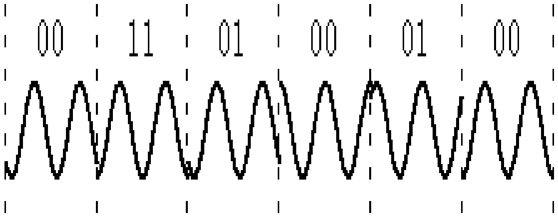


"0" $= A * \text{Sin}\,(2\pi f_1\, t + 0)$

"1" $= A * \text{Sin}\,(2\pi f_1\, t + 180)$

Quadrature PSK

- **get more efficient use if each signal element represents more than one bit**
  - e.g.. shifts of $\pi/2$ (90^o)
  - each element represents two bits
  - split input data stream in two & modulate onto carrier & phase shifted carrier
- **can use 8 phase angles & more than one amplitude**
  - 9600bps modem uses 12 angles, four of which have two amplitudes



$$\text{“11”} = A * \mathrm{Cos}\,(2\pi f_c\, t + 45)$$
$$\text{“10”} = A * \mathrm{Cos}\,(2\pi f_c\, t + 135)$$
$$\text{“00”} = A * \mathrm{Cos}\,(2\pi f_c\, t + 225)$$
$$\text{“01”} = A * \mathrm{Cos}\,(2\pi f_c\, t + 315)$$

Quadrature Amplitude Modulation

- **QAM used in asymmetric digital subscriber line (ADSL) and some wireless communication**
- **combination of ASK and PSK**
- **logical extension of QPSK**
- **send two different signals simultaneously on same carrier frequency**
  - uses two copies of carrier f_c, one shifted by a phase angle of 90°
  - each carrier is ASK modulated
  - two independent signals over same medium
  - demodulate and combine for original binary output

QAM Modulator

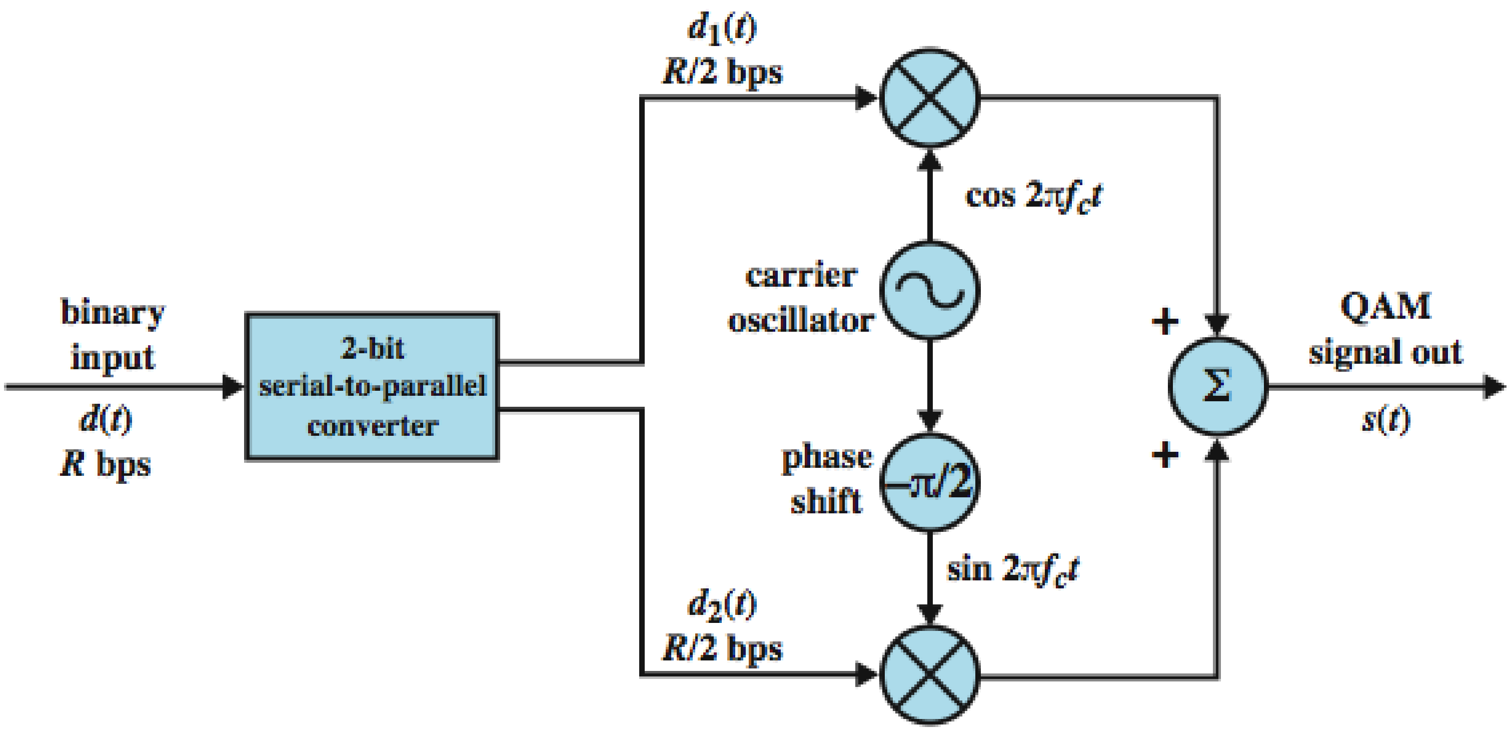
send two different signals simultaneously on same carrier frequency

uses two copies of carrier f_c, one shifted by a phase angle of 90°

each carrier is ASK modulated

two independent signals over same medium

demodulate and combine for original binary output



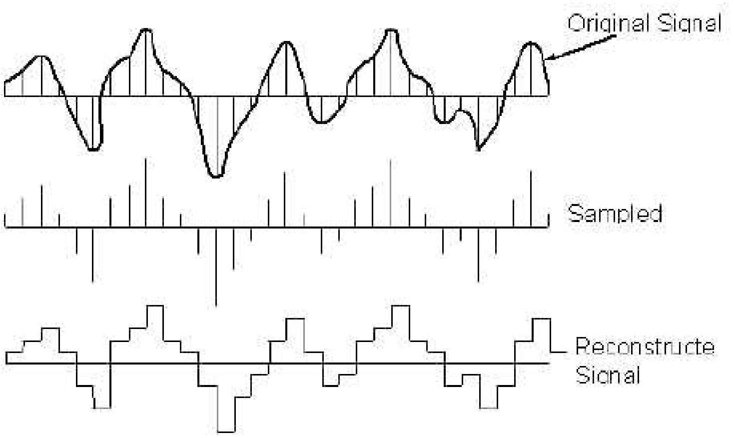
Data Encoding

- ✓digital data to digital signal
- ✓analog data to analog signal
- ✓digital data to analog signal
- analog data to digital signal

Analog Data, Digital Signal

- **Need to convert**
  - **analog data into → digital Data**
    - **called digitization**
  - **then digital data to → digital signal**

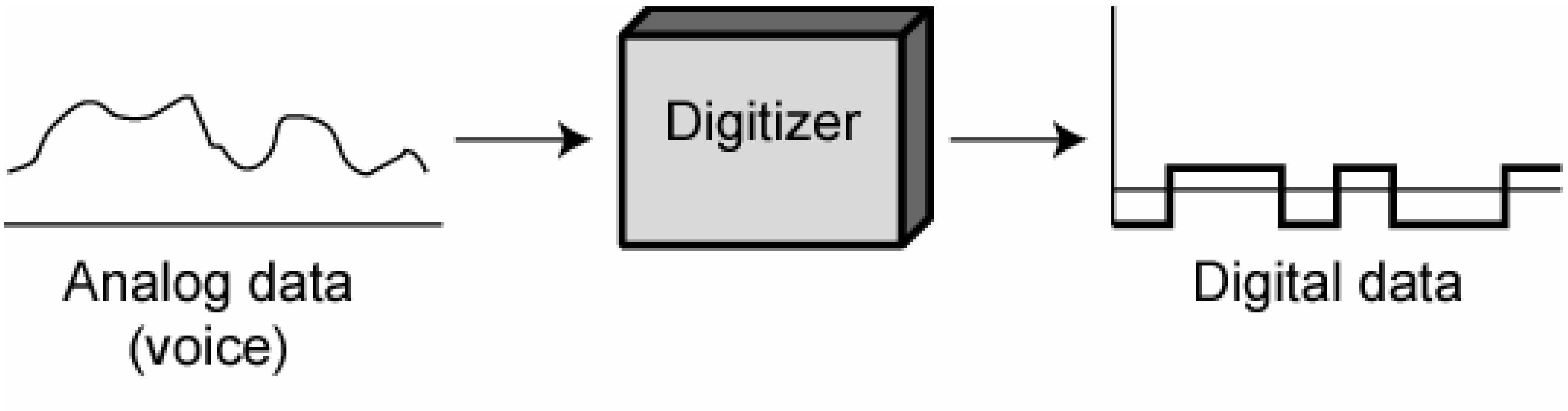
- **Digitization Principle: Take the amplitude of the signal at different intervals and convert the signal value into digital data.**

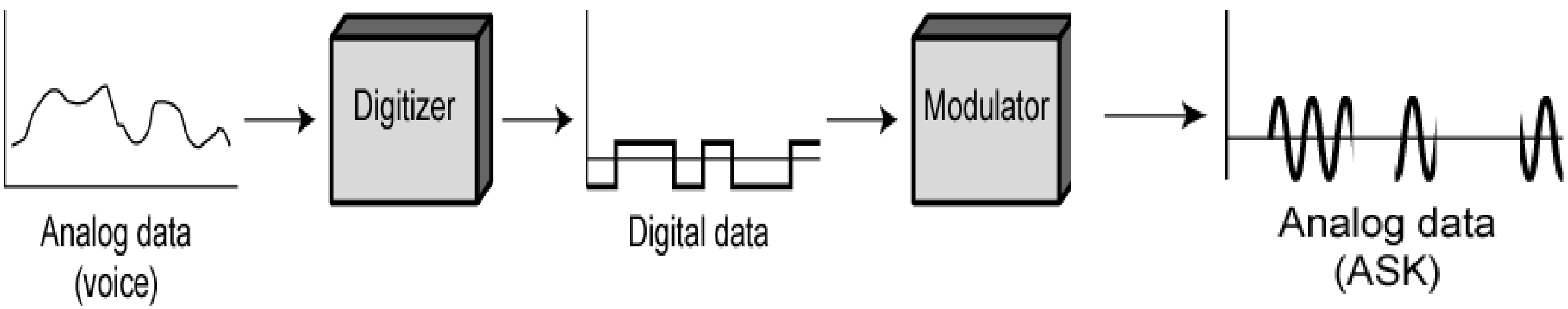


Analog Data, Digital Signal

- **digitization is conversion of analog data into digital data which can then:**
  - be transmitted using NRZ-L
  - be transmitted using code other than NRZ-L
  - be converted to analog signal using one of the modulation techniques
- **Codec (coder-decoder)** : device used for converting analog data to digital form . two principal techniques used in codecs **:**
  - pulse code modulation
  - delta modulation

Digitizing Analog Data

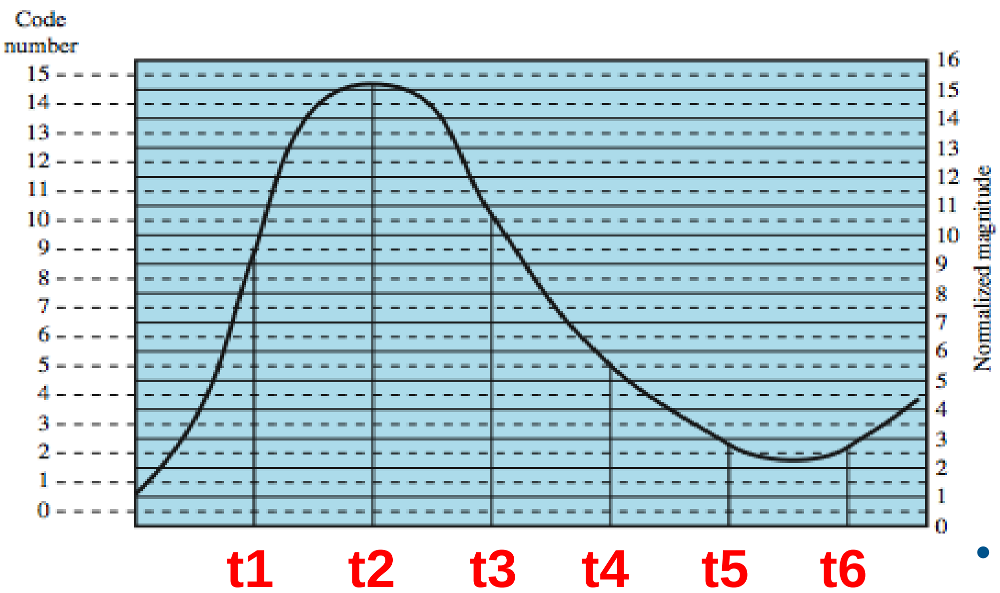




Analog Data, Digital Signal

- **analog to digital conversion is done using a codec (coder-decoder)**
- **Conversion is done at least in two different ways**
  - pulse code modulation (PCM)
  - delta modulation (DM)

An example



- **Let the analog data be as shown in the left picture**

  Find the amplitude of the signal at regular intervals (say at t1, t2, t3, t4, t5, t6 in the figure)

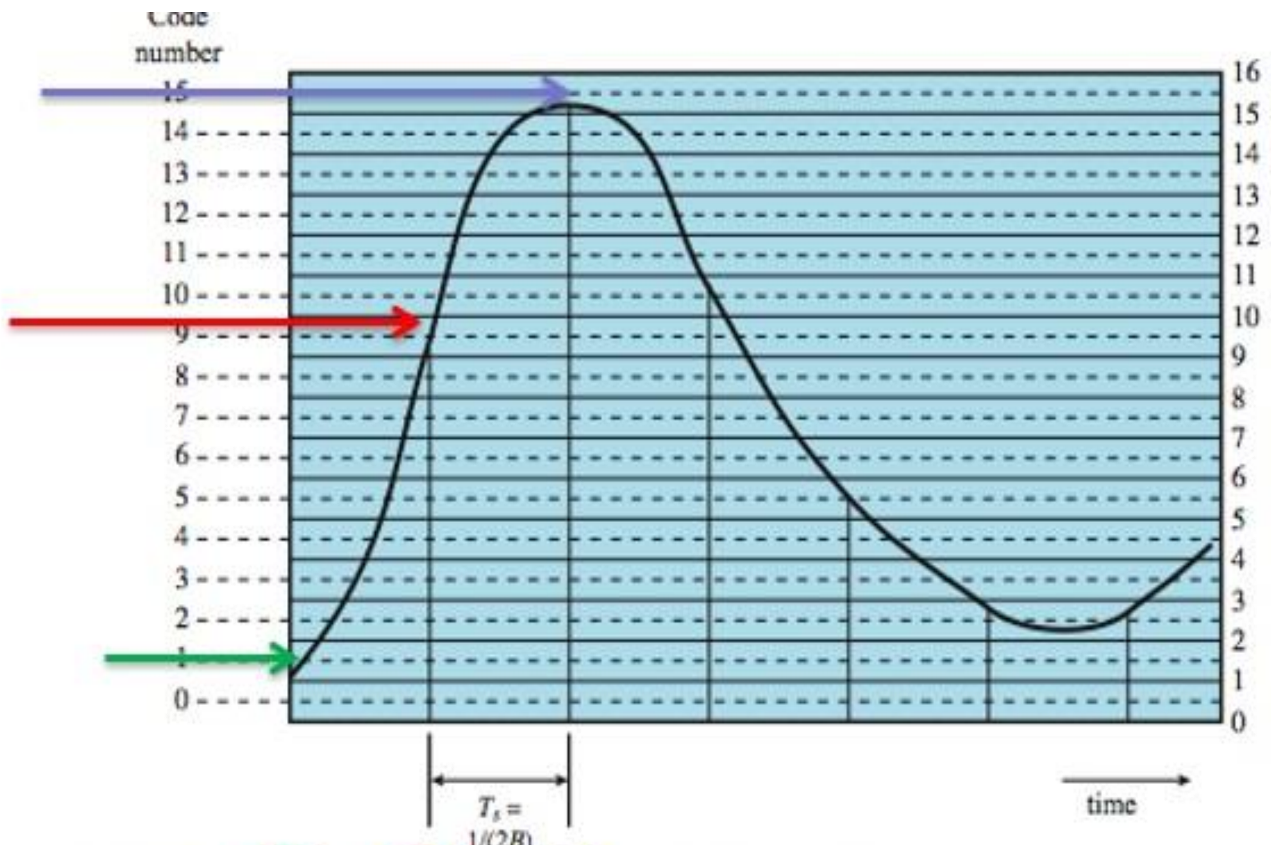
  Express each amplitude in a binary form.

- **e.g. a signal voltage level, say at t2 is represented as binary code**

  code # [15] in base10= $[1111]_2$

  which is binary - digital data

An Example (cont'd)



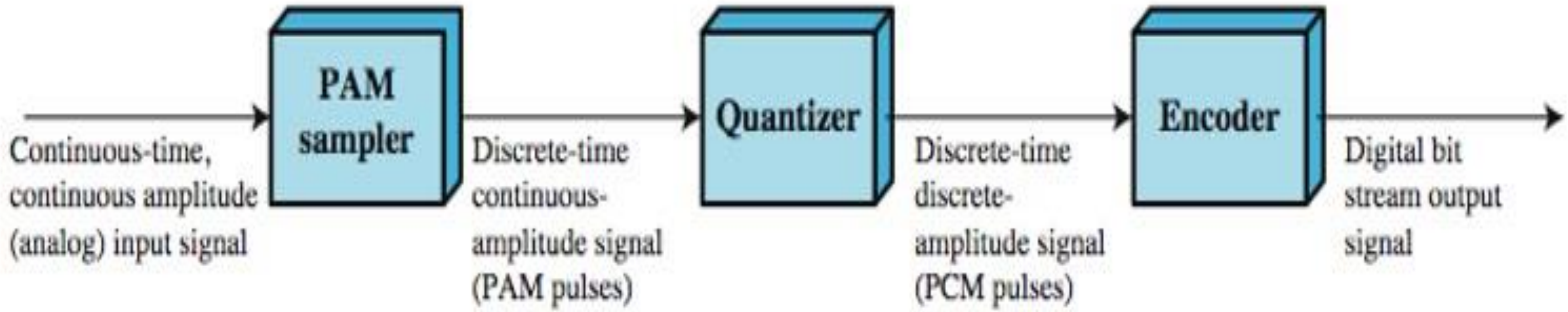
PAM value	1.1	9.2	15.2	10.8	5.6	2.8	2.7
quantized code number	1	9	15	10	5	2	2
PCM code	0001	1001	1111	1010	0101	0010	0010

Voltage levels (M)	Binary 4-bit code
15	1111
14	1110
13	1101
12	1100
11	1011
10	1010
9	1001
8	1000
7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000

How often you need to sample the analog data?

- **If you want to decode the original signal from its digital data, then the sampling theorem specifies the minimum rate at which you need to sample the analog signal**
  - "If a signal f*(t)* is sampled at regular intervals of time at a rate higher than twice the highest signal frequency, then the samples contain all the information of the original signal" (Stallings DCC8e)
  - e.g.. 4000Hz voice data, requires 8000 sample per sec
  - Note that these are *analog* samples, referred to as **pulse amplitude modulation (PAM)** samples
  - The PAM samples must be assigned to a binary code to complete the digitization of the original analog signal

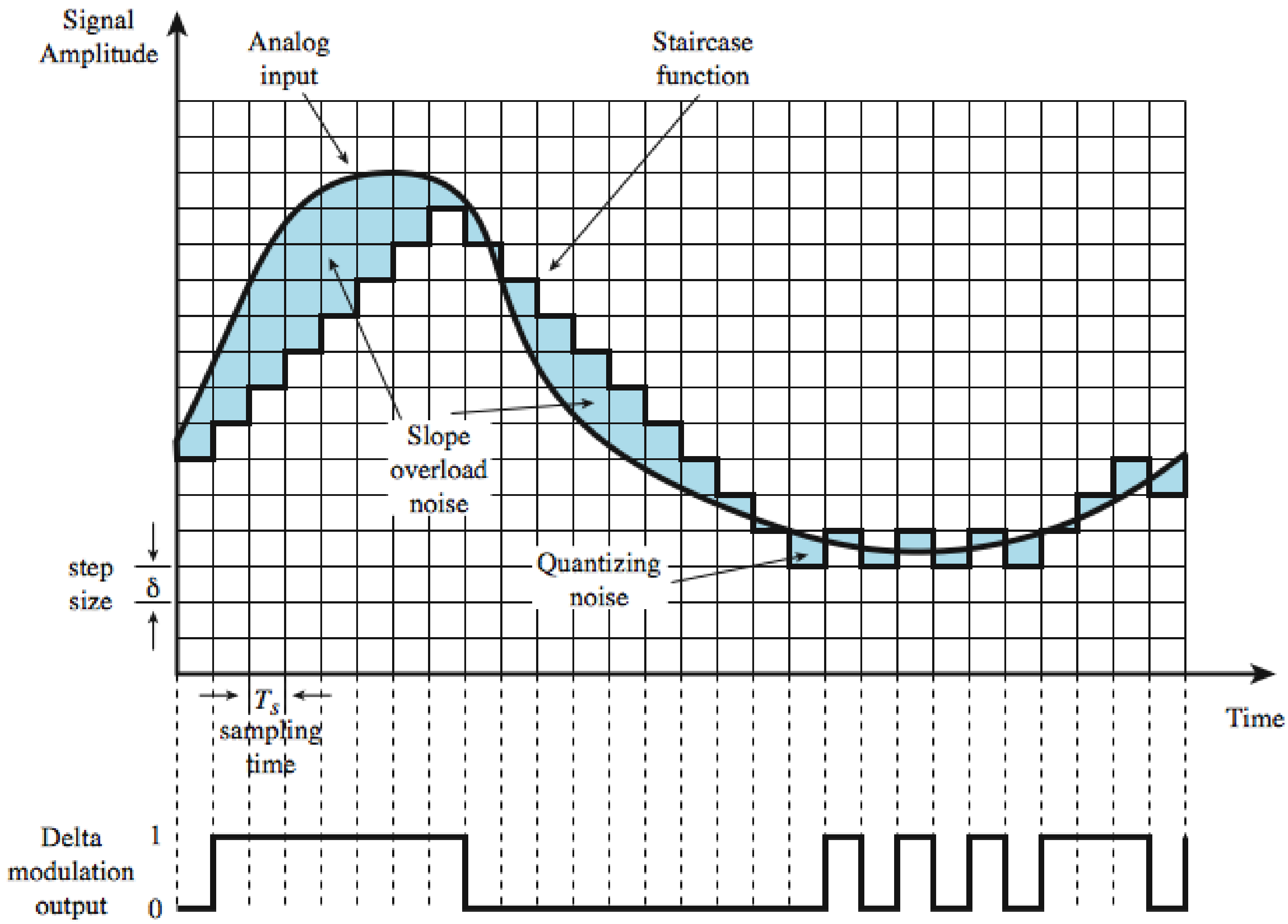
PCM Block Diagram



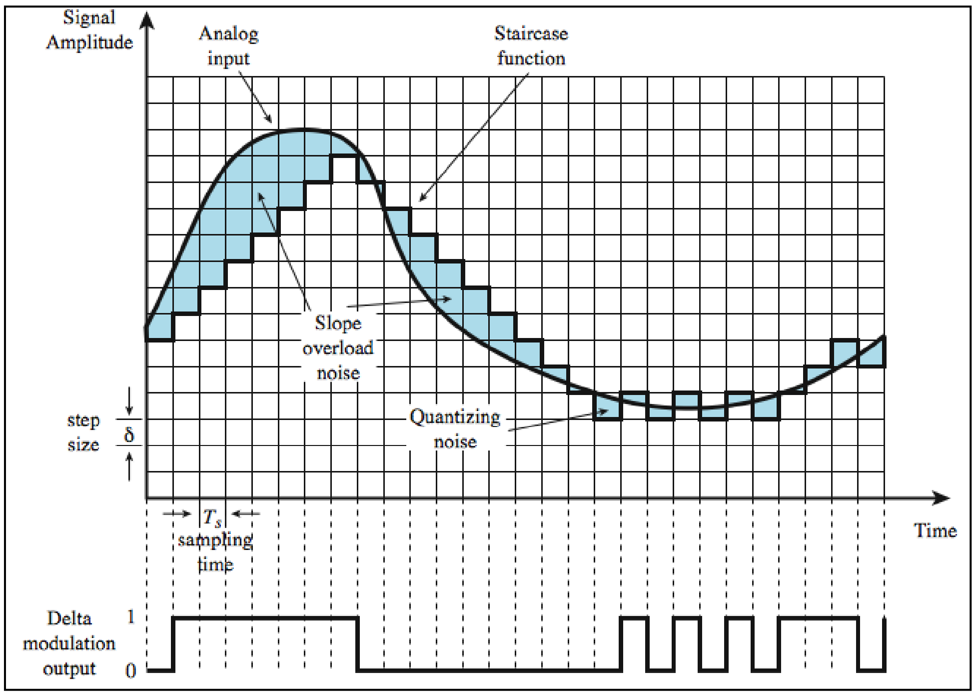
Delta Modulation

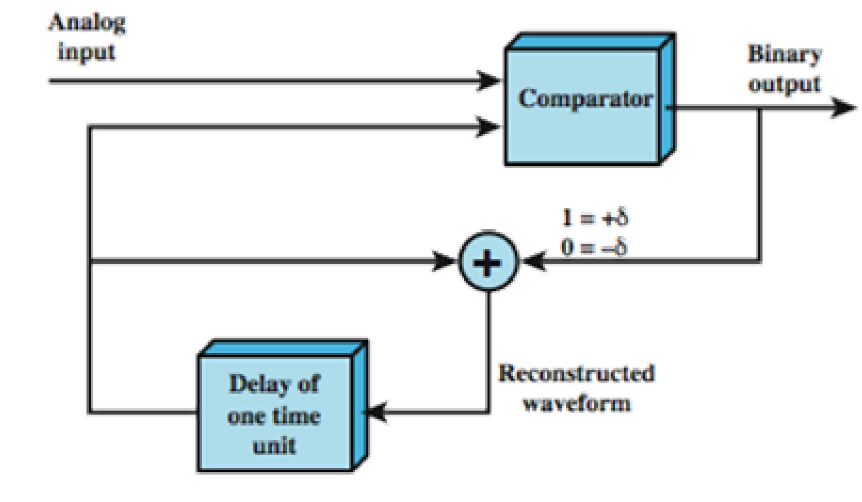
- **analog input is approximated by a staircase function**
  - can move up or down one level (δ) at each sample interval
- **has binary behavior**
  - since function only moves up or down at each sample interval
  - hence can encode each sample as single bit
  - 1 for up or 0 for down

Delta Modulation Example

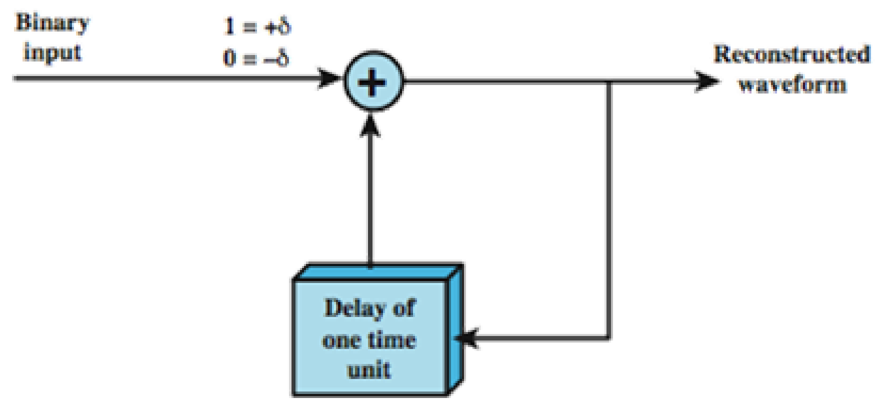


Delta Modulation Operation





(a) Transmission



(b) Reception

PCM versus Delta Modulation

- **DM has simplicity compared to PCM**
- **DM is easier to implement compared to PCM**
- **DM has worse SNR compared to PCM**
- **PCM has better SNR compared to DM**
- **PCM has issue of bandwidth being used**
  - e.g. for good voice reproduction with PCM
    - > want 128 levels (7 bit) & **voice bandwidth of 4khz**
    - > need 8000 x 7 = 56kbps
- **data compression can improve on BW issues..**
- **PCM is the choice and still growing demand for digital signals**
  - Due to use of repeaters, TDM(no intermodulation noise), and efficient switching
- **PCM is preferred to DM for analog signals**

Summary

- **We looked at signal encoding techniques**
  - digital data to digital signal
  - analog data to analog signal
  - digital data to analog signal
  - analog data to digital signal

Thank You