Digital Signal Transmission

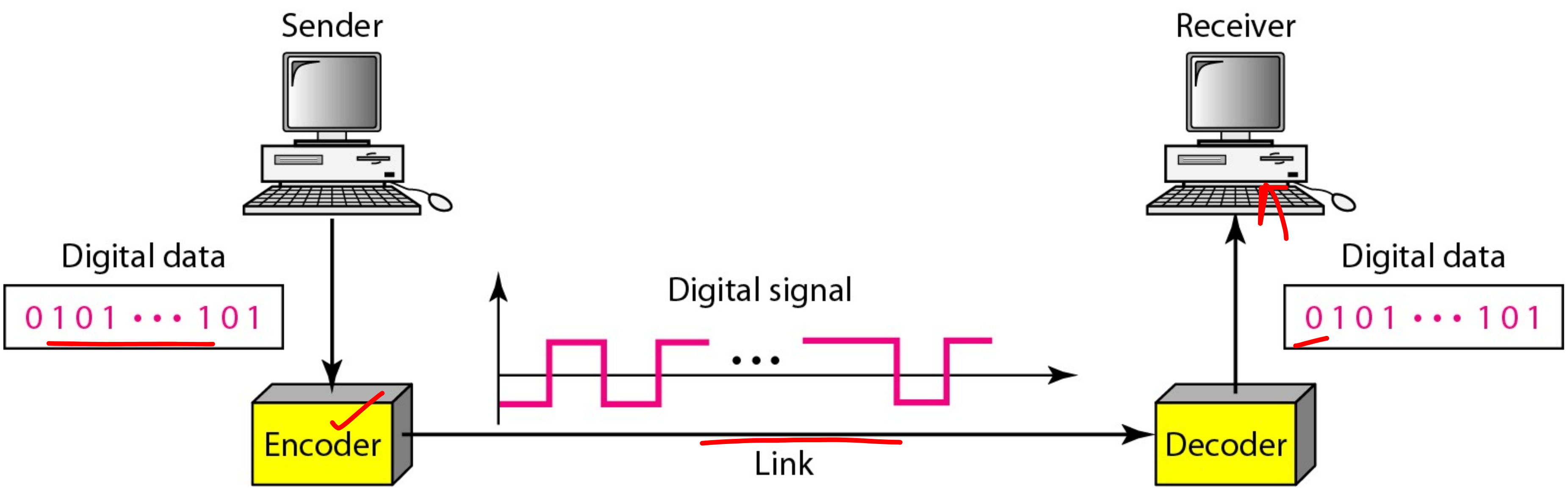
Digital-to-Digital Conversion

- *We will see how to represent digital data by using digital signals.*
- *The conversion involves three techniques: line coding, block coding, and scrambling.*
  - *Line coding is always needed;*
  - *block coding and scrambling may or may not be needed.*

Line Coding

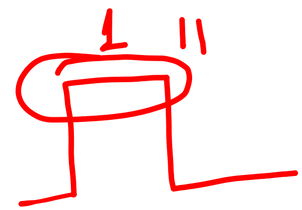
- Converting a string of 1's and 0's (digital data) into a sequence of signals that denote the 1's and 0's.
- For example a high voltage level (+V) could represent a "1" and a low voltage level (0 or -V) could represent a "0".

Line coding and decoding



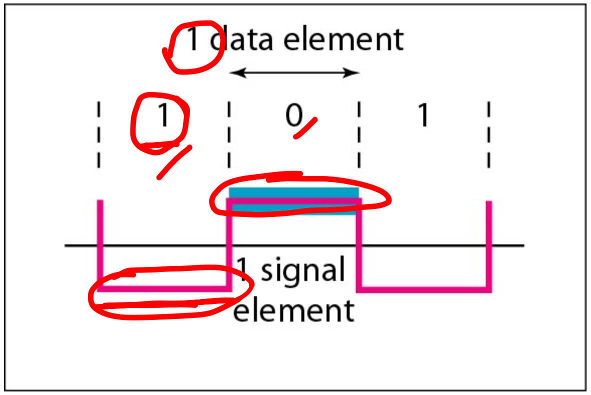
Mapping Data symbols onto Signal levels

- A data symbol (or element) can consist of a number of data bits:
  - 1, 0 or
  - 11, 10, 01, ……

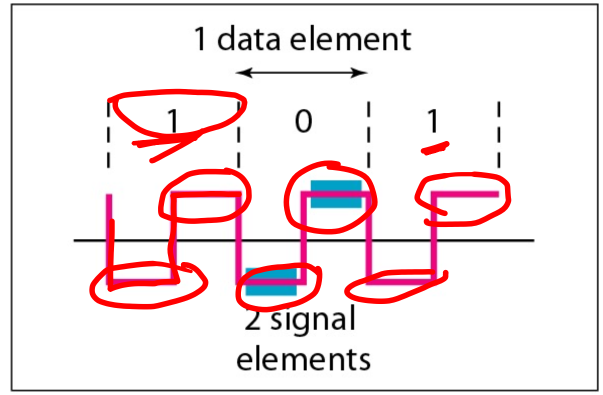


- A data symbol can be coded into a single signal element or multiple signal elements
  - 1 → +V, 0 → -V
  - 1 → +V and -V, 0 → -V and +V
- The ratio ‘r’ is the number of data elements carried by a signal element.

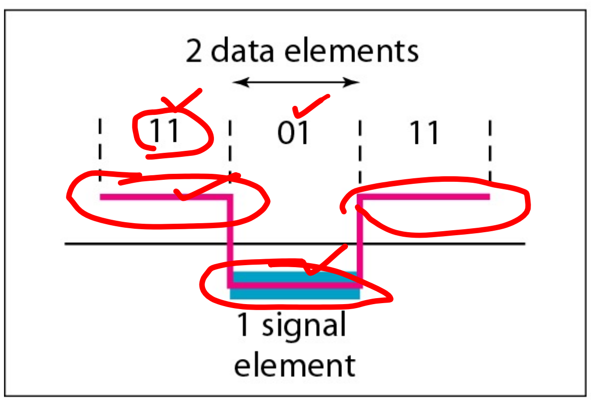
Signal element versus data element



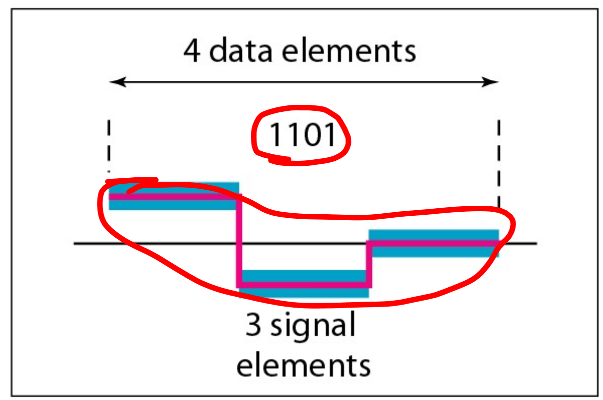
a. One data element per one signal element (r = 1)



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



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



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

Data rate and signal rate

- The data rate defines the *number of bits sent per second (bps)*.
  - It is often referred to the bit rate.
- The signal rate is the *number of signal elements sent in a second*
  - Measured in bauds.
  - It is also referred to as the modulation rate.
- Goal is to *increase* the data rate whilst reducing the baud rate.

Data rate and Baud rate

- The baud or signal rate can be expressed as:

$$S = c \times N \times 1/r \quad \textit{bauds}$$

where

- N is data rate
- c is the case factor (worst, best & avg.)
- r is the ratio between data element & signal element

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

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

Note

Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

Bandwidth and baud rate

- Baud rate determines the bandwidth
  - More changes in signal mean more frequencies into the signal
  - Again bandwidth reflects the range of frequencies
  - So bandwidth is proportional to the signal rate (baud rate)
  - The minimum bandwidth can be written as

$$B_{min} = c \times N \times (1/r)$$

$$\rightarrow N_{max} = (1/c) \times B \times r$$

Example

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

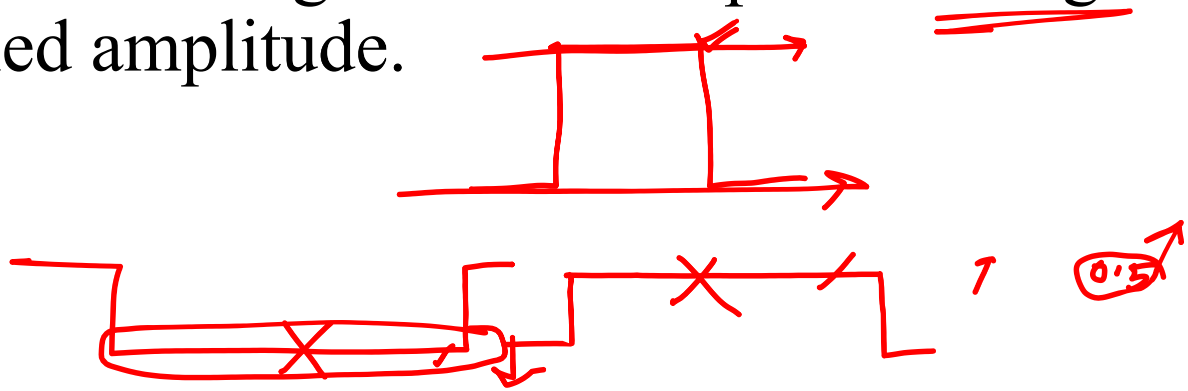
Solution

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

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

Baseline wandering

- A receiver evaluates the average power of the received signal (called the baseline)
  - uses this as reference to determine the value of the incoming data elements.
- If the incoming signal does not vary over a long period of time, the baseline will drift and thus cause errors in detection of incoming data elements.
- A good line encoding scheme will prevent long runs of fixed amplitude.



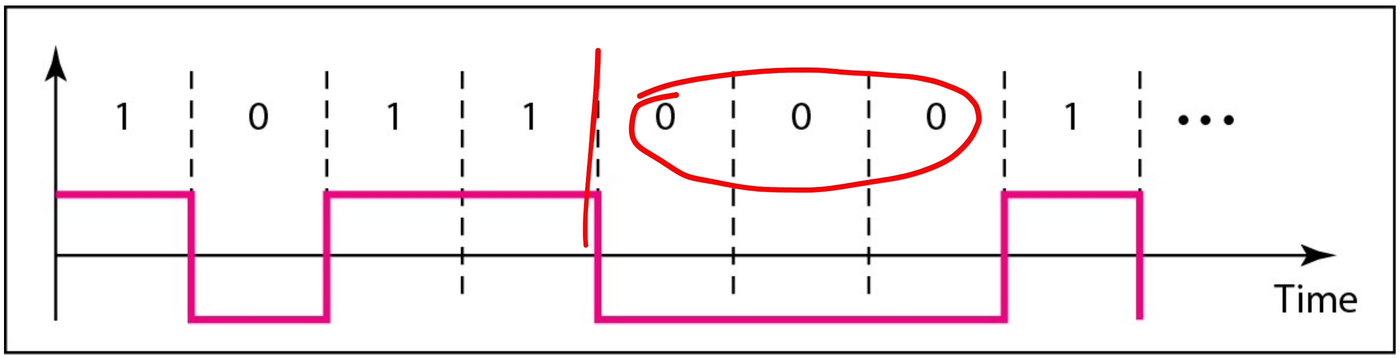
Line encoding Common Characteristics

- DC components :
  - when the voltage level remains constant for long periods of time, there is an increase in the low frequencies of the signal.
  - Most channels are bandpass and may not support the low frequencies.
- This will require the removal of the DC component of a transmitted signal.

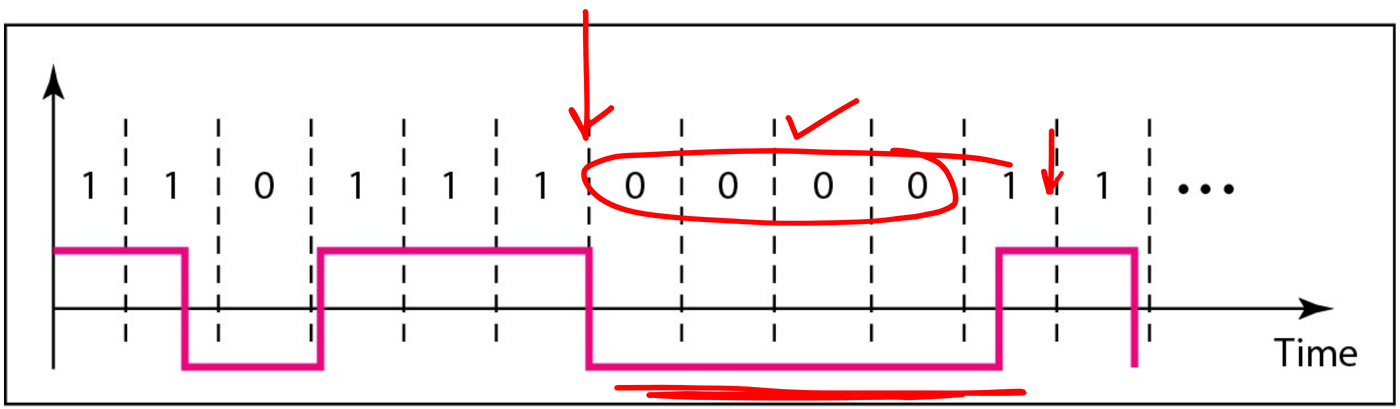
Line encoding Common Characteristics

- Self synchronization :
  - The clocks at the sender and the receiver must have the same bit interval.
- If the receiver clock is faster or slower it will misinterpret the incoming bit stream.

Effect of lack of synchronization



a. Sent



b. Received

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

At 1 kbps, the receiver receives 1001 bps instead of 1000 bps.

1000 bits sent	1001 bits received	1 extra bps

At 1 Mbps, the receiver receives 1,001,000 bps instead of 1,000,000 bps.

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

Line encoding Common Characteristics

- **Error detection:**
  - Errors occur during transmission due to line impairments.
- Some codes are constructed such that when an error occurs it can be detected.
  - Example:
    - A particular signal transition is not part of the code.
    - When it occurs, the receiver will know that a symbol error has occurred.

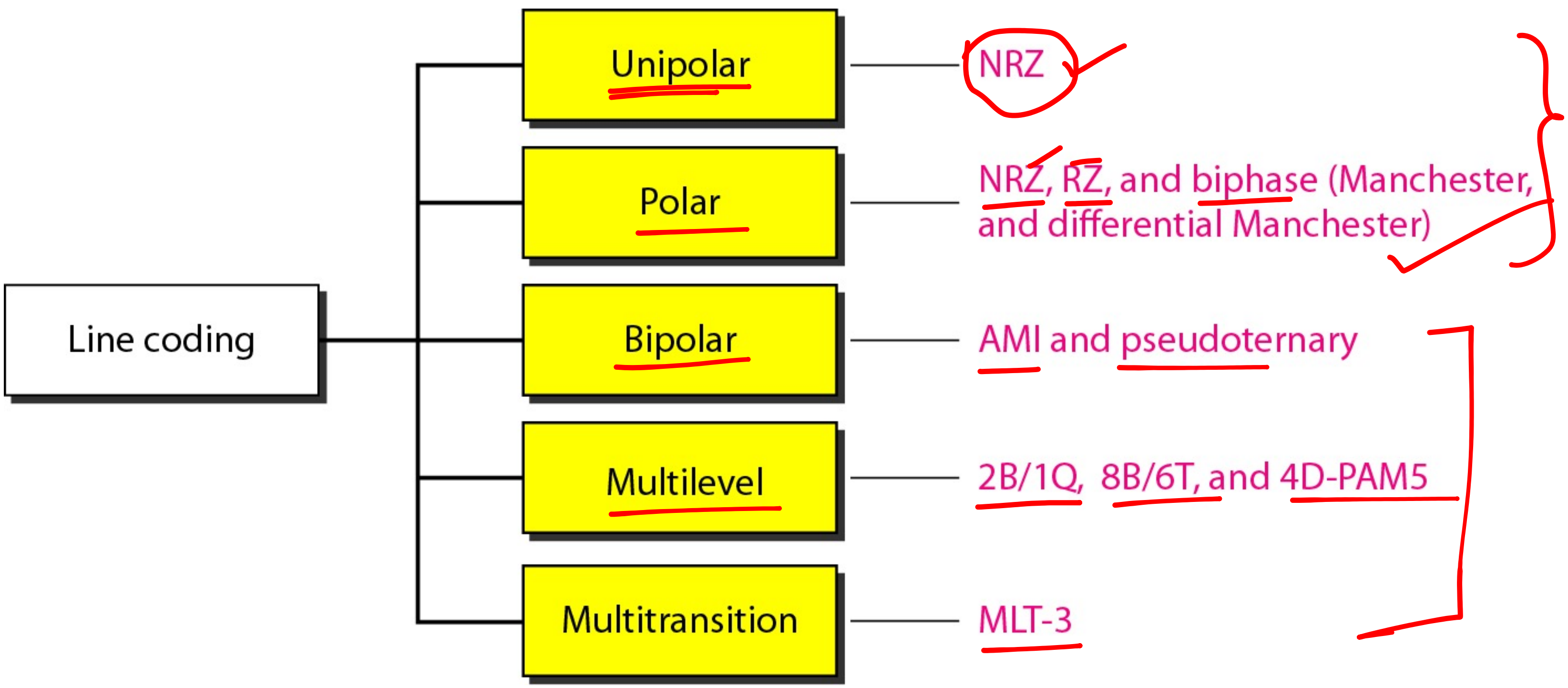
Line encoding Common Characteristics

- Noise and interference:
  - There are line encoding techniques that make the transmitted signal "immune" to noise and interference.
- This means that the signal cannot be corrupted, it is stronger than error detection.

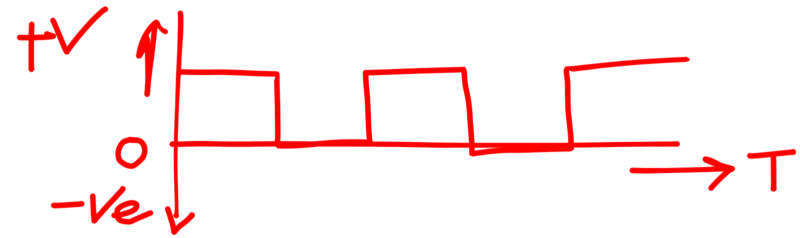
Line encoding Common Characteristics

- **Complexity:**
  - The more robust and resilient the code, the more complex it is to implement
  - The price is often paid in baud rate or required bandwidth.

Line coding schemes

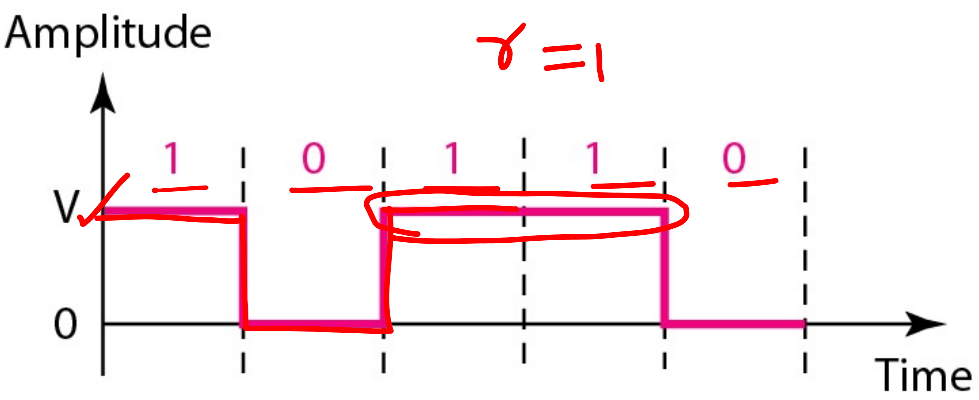


Unipolar



- All signal levels are on one side of the time axis
  - Either above or below
- **Non-Return-to-Zero (NRZ)**:
  - This 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

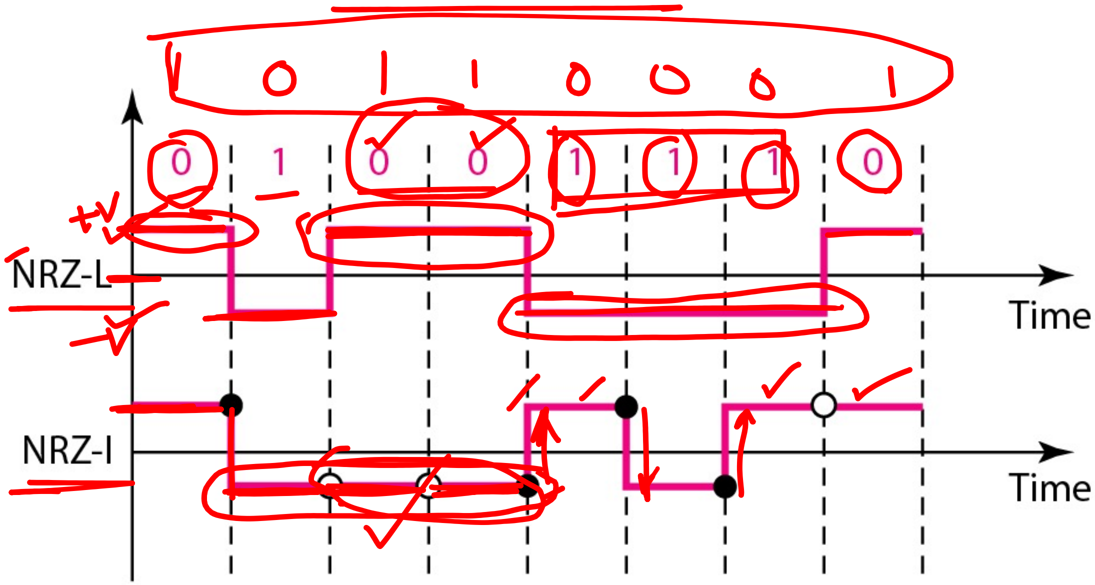
Polar - NRZ

- The voltages are on both sides of the time axis.
- Polar NRZ scheme can be implemented with two voltages.
  - Ex: +V for 1 and -V for 0.

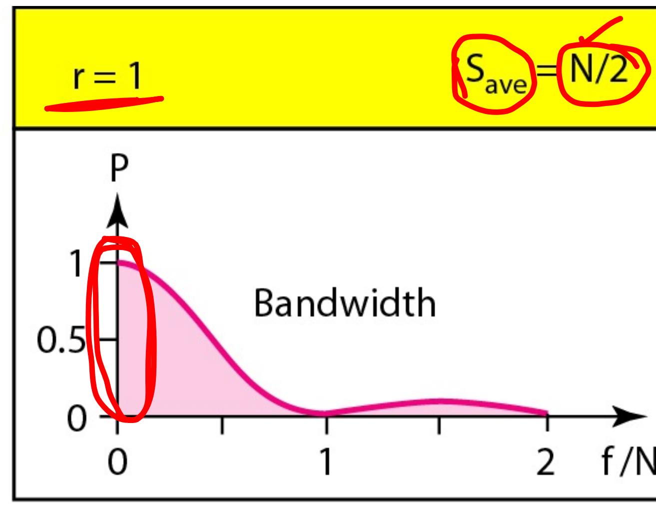
- There are two versions:
  - NZR - Level (NRZ-L):
    - Positive voltage for one symbol and negative for the other
  - NRZ - Inversion (NRZ-I) :
    - The change or lack of change in polarity determines the value of a symbol.
    - E.g. a "1" symbol inverts the polarity a "0" does not.

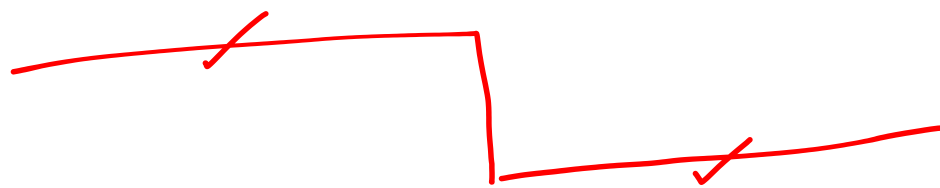
Polar NRZ-L and NRZ-I schemes

$S = c \times N \times \frac{1}{r}$



○ No inversion: Next bit is 0 ● Inversion: Next bit is 1





Polar NRZ-L and NRZ-I schemes

- Problems:
  - Baseline wandering: exist for both NRZ-L and NRZ-I
    - In NRZ-I this problem occurs only for long sequence of 0s
    - If somehow we can manage this long sequence of 0s, we can avoid baseline wandering
  - Synchronization Problem: exist for both NRZ-L and NRZ-I
  - Sudden change in polarity:
    - Sudden change in polarity results in 0s interpreted as 1s and vice versa in NRZ-L
    - NRZ-I does not have this problem

Note

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

Note

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

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 $S = c \times N \times \cancel{R}\ \frac{1}{r}$

$$= 1/2 \times N \times 1$$

$$= 500 \text{ kbaud.}$$

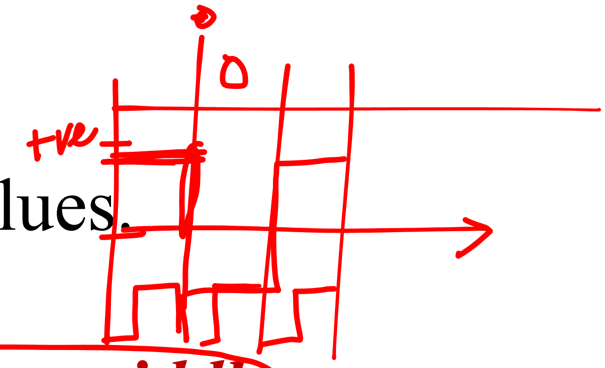
The minimum bandwidth for this average baud rate is $B_{min} = S = 500$ ***kHz.***

Note: c = 1/2 for the avg. case as worst case is 1 and best case is 0

Polar NRZ-L and NRZ-I schemes

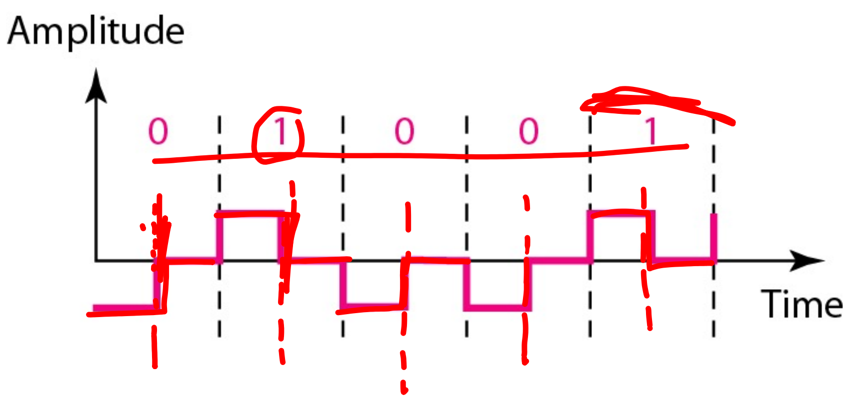
- Main problem in NRZ is synchronization in clocks for both sender and receiver
- The receiver does not know when one bit is has ended and the next bit is starting
- One solution is ***return-to-zero (RZ)*** scheme

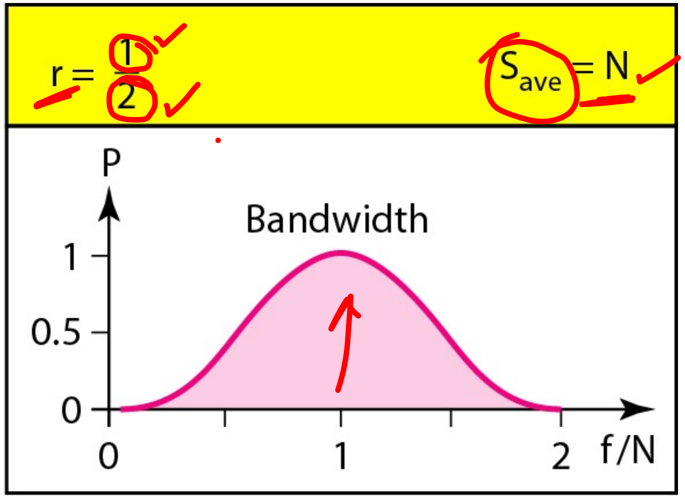
Polar - RZ



- This scheme uses three voltage values.
  - ***+ve, 0, -ve.***
- Each symbol has a ***transition in the middle***.
  - Either from high to zero or from low to zero.
  - Signal changes not between bits but during the bit

Polar RZ scheme





Polar - RZ

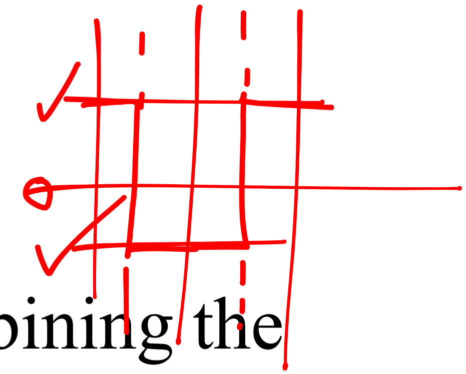
Advantages:

- ***No DC components or baseline wandering.***
- Self synchronization:
  - Transition indicates symbol value.

Disadvantage:

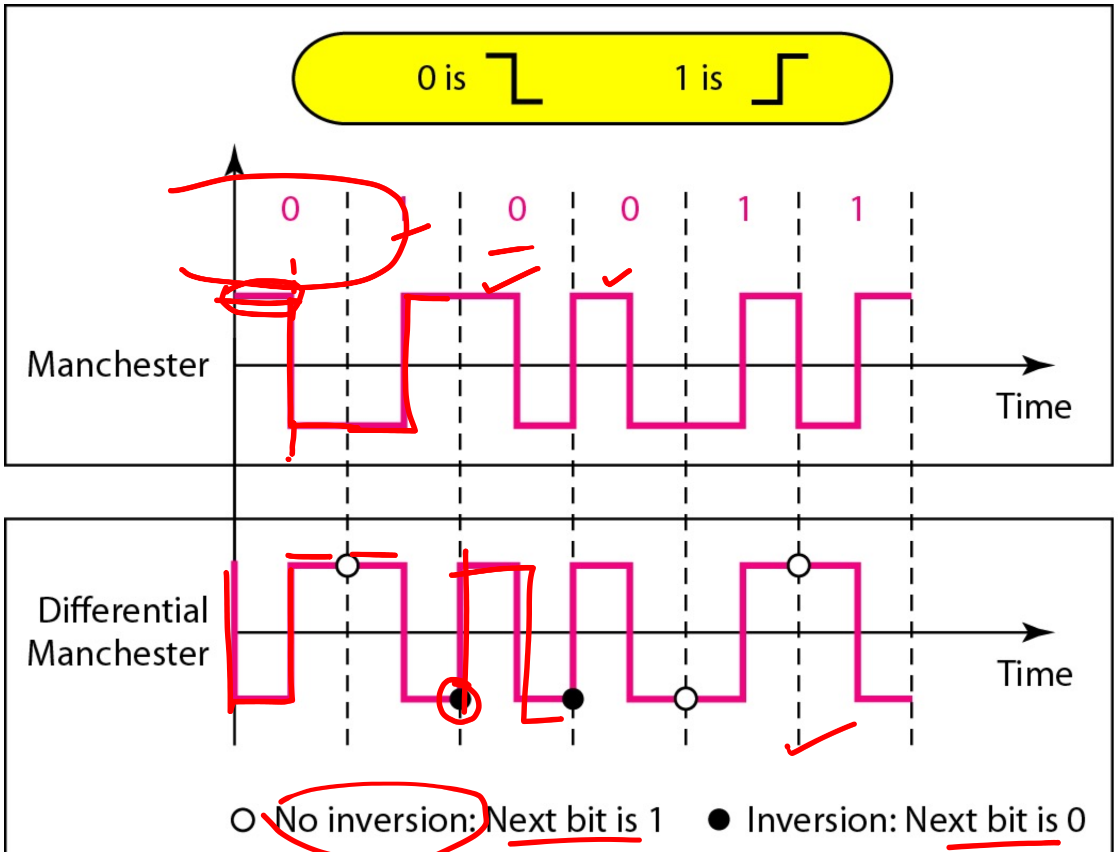
- Requires two signal changes to encode a bit
- Therefore requires a wider bandwidth
- Sudden change in polarity still exists
- Complexity
- It has ***no error detection capability***

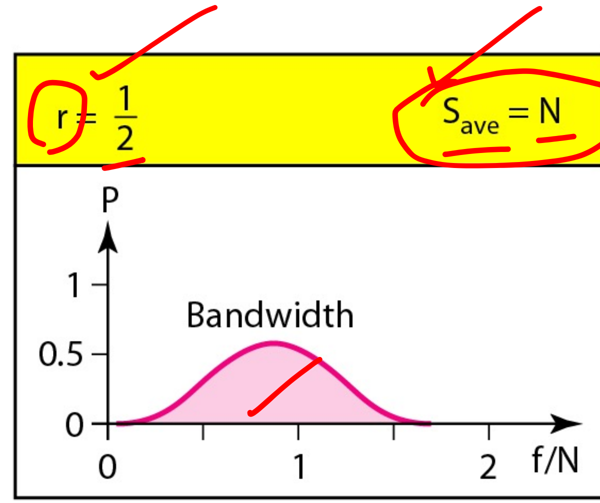
Polar - Biphase: Manchester and Differential Manchester



- **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:** 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





Note

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

Note

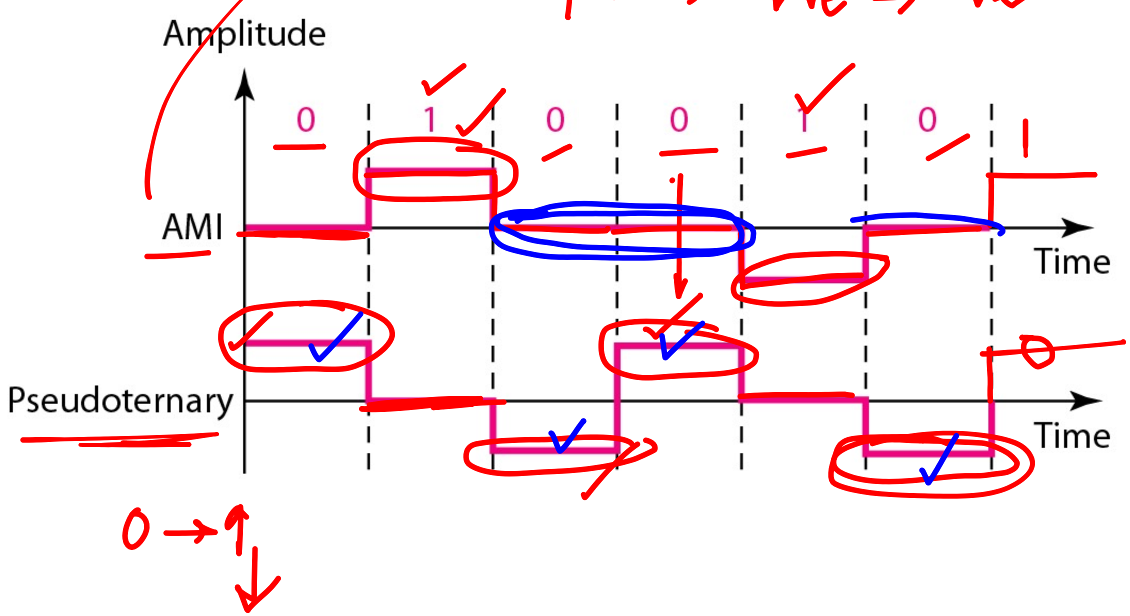
The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.

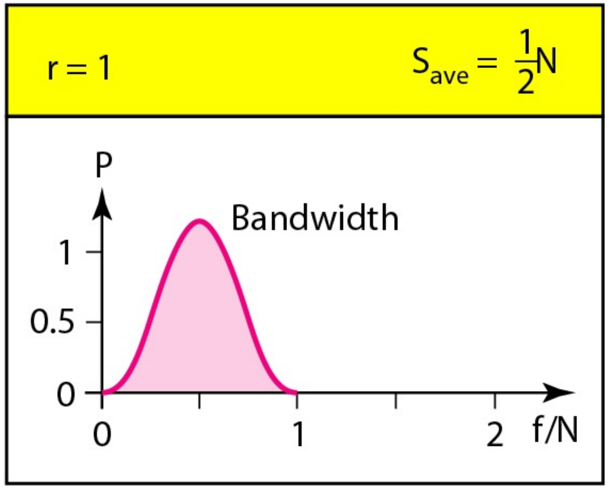
Bipolar - AMI and Pseudoternary

- Code uses 3 voltage levels:
  - +, 0, -, to represent the symbols
- 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*

0 → voltage 0
1 → +ve → -ve





Bipolar Coding Schemes

- It is a better alternative to NRZ.
- Has *no DC* component or *baseline wandering.*
- AMI is commonly used for long distance communication
  - Has *no self synchronization* because long runs of “0”s results in no signal transitions.
- *No error detection*.

Multilevel Schemes

$2^1 = 2$

- In these schemes we increase the number of data bits per baud 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.
- *m* data elements can produce a combination of 2^m data patterns.
- If we have L signal levels, we can produce L^n combinations of signal pattern.

Multilevel Schemes

- Now we have 2^m data patterns and L^n signal patterns.
  - If $2^m > L^n$ then we cannot represent the all the data patterns
    - we don't have enough signals.
  - If $2^m = L^n$ then we have an *exact mapping* of one pattern on one signal.
  - If $2^m < L^n$ then we have *more signal patterns* than data patterns.
    - we can choose the signals that are more distinct to represent the data pattern
    - therefore have better noise immunity and error detection as some signals are not valid.

Representing Multilevel Codes

- We use the notation ***mBnL***
  - *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.
- A letter is often used in place of L
  - L = B (for binary), L = T (for 3 ternary), L = Q (for 4 quaternary).

Note

In *m*B*n*L schemes, a pattern of *m* data elements is encoded as a pattern of *n* signal elements in which $2^m \leq L^n$.

Multilevel: 2B1Q scheme

14

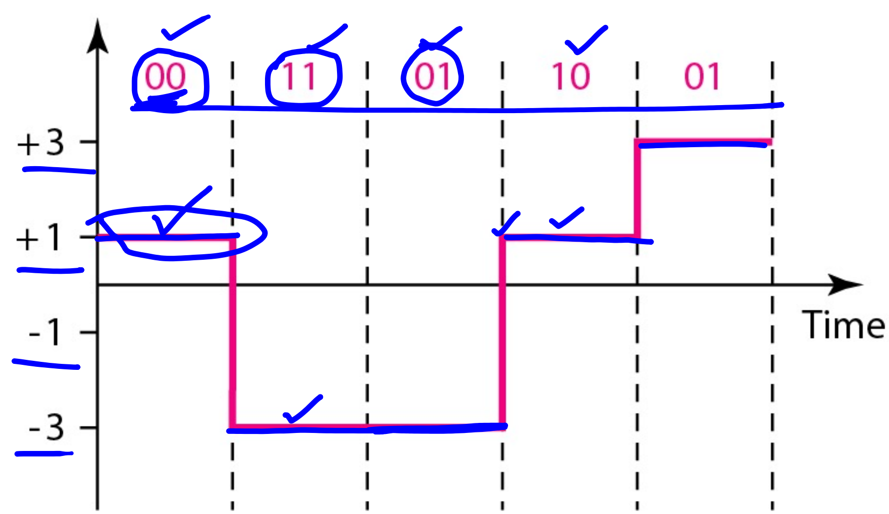
n=1 L=4 $4^1=4$

+1 +3 -1 -3

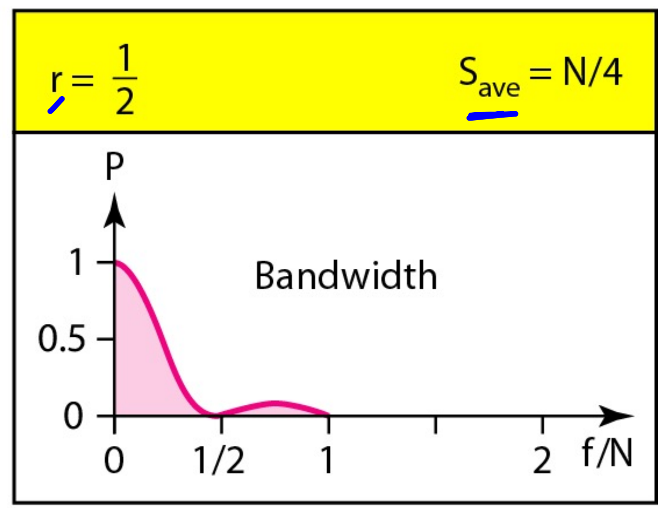
2^2

	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

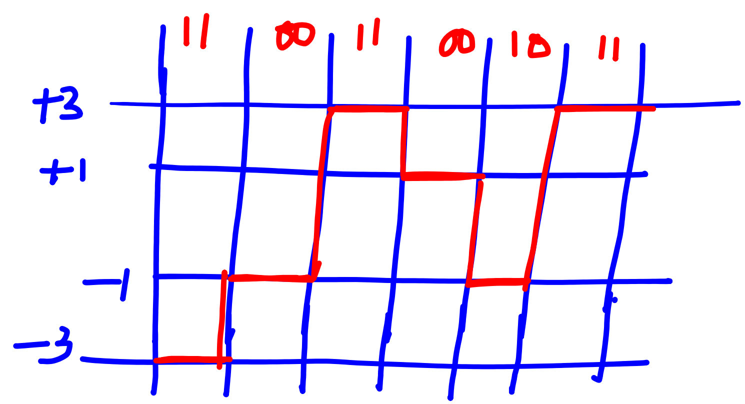


1100 11 00 1011

Redundancy

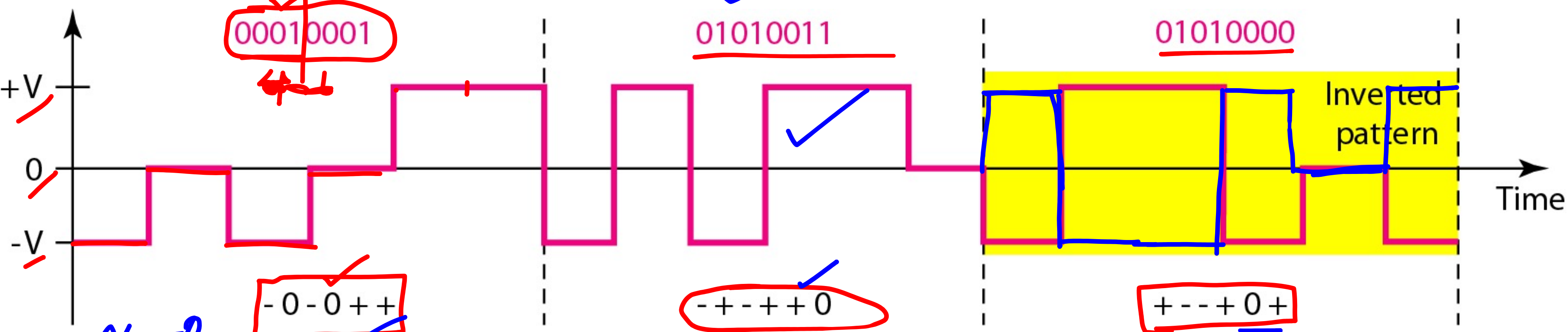
- In the 2B1Q scheme we have ***no redundancy*** and we see that a *DC component is present.*

- *2B1Q* scheme is used in DSL (Digital subscriber line) technology to provide a high speed connection to the Internet by using subscriber telephone line.



$2^8 = 256$ 3 $3^6 = 729$

Multilevel: 8B6T scheme



- This code is used with 100BASE-4T cable.
- This scheme encodes a pattern of 8 bits as a pattern of 6 signal elements using 3 levels of signals.
  - $2^8 = 256$ data patterns, $3^6 = 729$ different signal patterns
  - So 729-256 = 473 number of redundant signal elements.

Multilevel: 8B6T scheme

- Each signal pattern has a weight of 0 or +1 DC values.
  - So no pattern with the weight -1
- To make whole stream DC balanced, the sender keeps track of the weight.
- If two groups of weight 1 are encountered one after another
  - First one is send as it is
  - Second one is sent totally inverted to give weight -1.
- Average signal rate is
  - $S_{avg} = ½ \times N \times 6/8 = 3N/8$
  - In practice minimum bandwidth is close to 6N/8

Multilevel using multiple channels

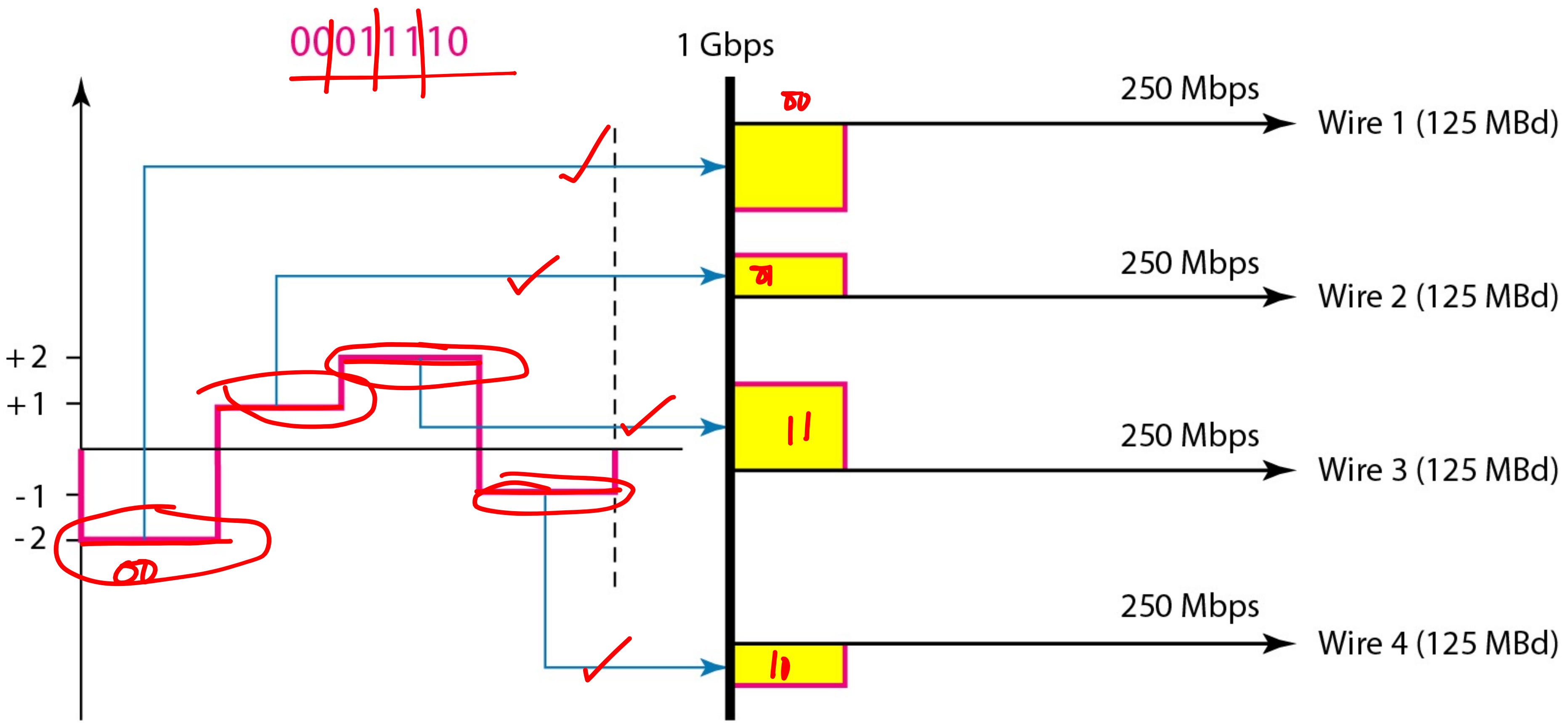
- In some cases, we split the signal transmission up and distribute it over several links.
- The separate segments are transmitted simultaneously.
  - This reduces the signaling rate per link → lower bandwidth.
- This requires all bits for a code to be stored.
- xD: means that we use 'x' links
- YYYz: We use 'z' levels of modulation where YYY represents the type of modulation (e.g. pulse ampl. mod. PAM).
- Codes are represented as: xD-YYYz

4D-PAMS

$S = c \times N \times \frac{1}{r}$

- 4D : data is sent over four wires at the same time
  - So signal rate can be reduced to N/8
- It uses 5 voltage levels: -2, -1, 0, 1, 2

Multilevel: 4D-PAM5 scheme



Advantages

- Signal rate reduced to N/8
- It can be used in Gigabit LANs to transmit data simultaneously over four wires
- Redundant data is used for error detection
- It offers self synchronization
- No DC components in the encoding

Multitransition Coding

- Because of synchronization requirements we force transitions.
- This can result in very high bandwidth requirements

  → more transitions than are bits (e.g. mid bit transition with inversion).
- Codes can be created that are differential at the bit level forcing transitions at bit boundaries.
  - This results in a bandwidth requirement that is equivalent to the bit rate.

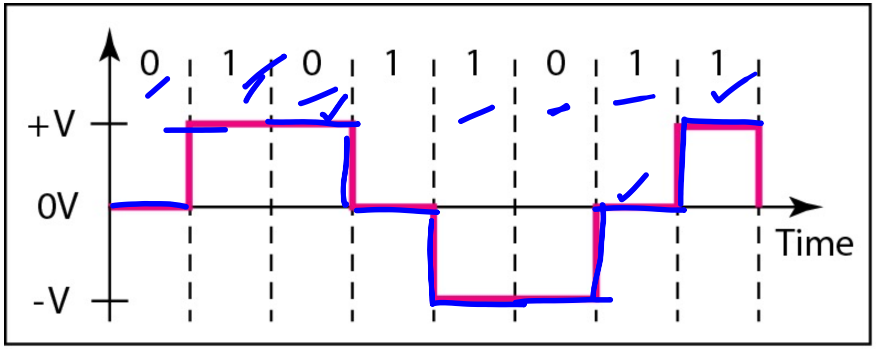
Multitransition Coding

- NRZ-I and differential Manchester are differential encoding
  - Use two transition rules to encode binary data (no inversion and inversion)
- If we have a signal with more than two levels, we can design a differential coding scheme with more than two transition rules
  - ***Multiline transmission, three levels*** (MLT 3) is such coding scheme

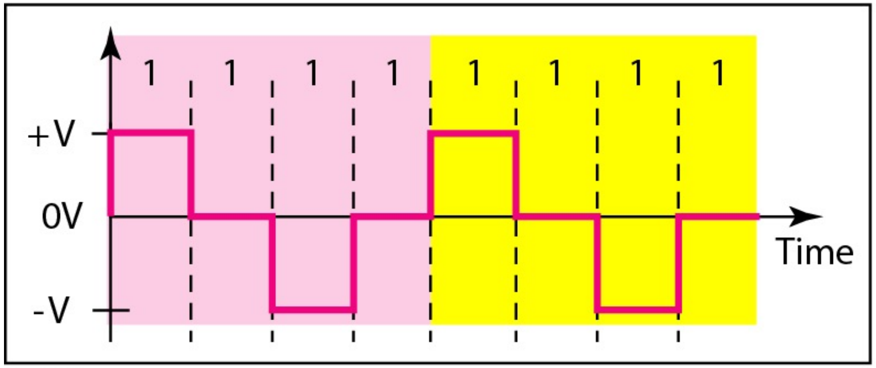
MLT 3

- Uses three levels
  - -V, 0, and +V
- Uses three transition rules to move between the **levels** voltage levels
  - If the next bit is 0, there is no transition
  - If the next bit is 1 and the current level is not 0, the next level is 0
  - If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level

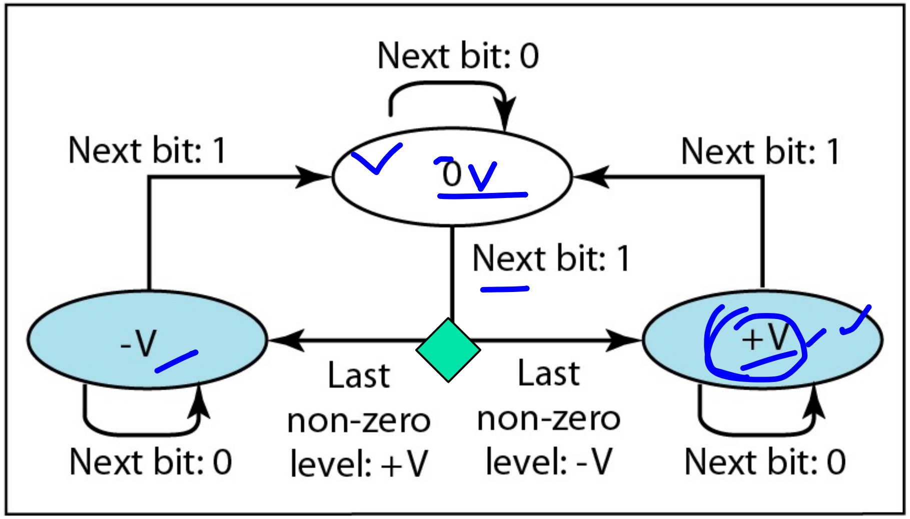
Multitransition: MLT-3 scheme



a. Typical case



b. Worse case



c. Transition states

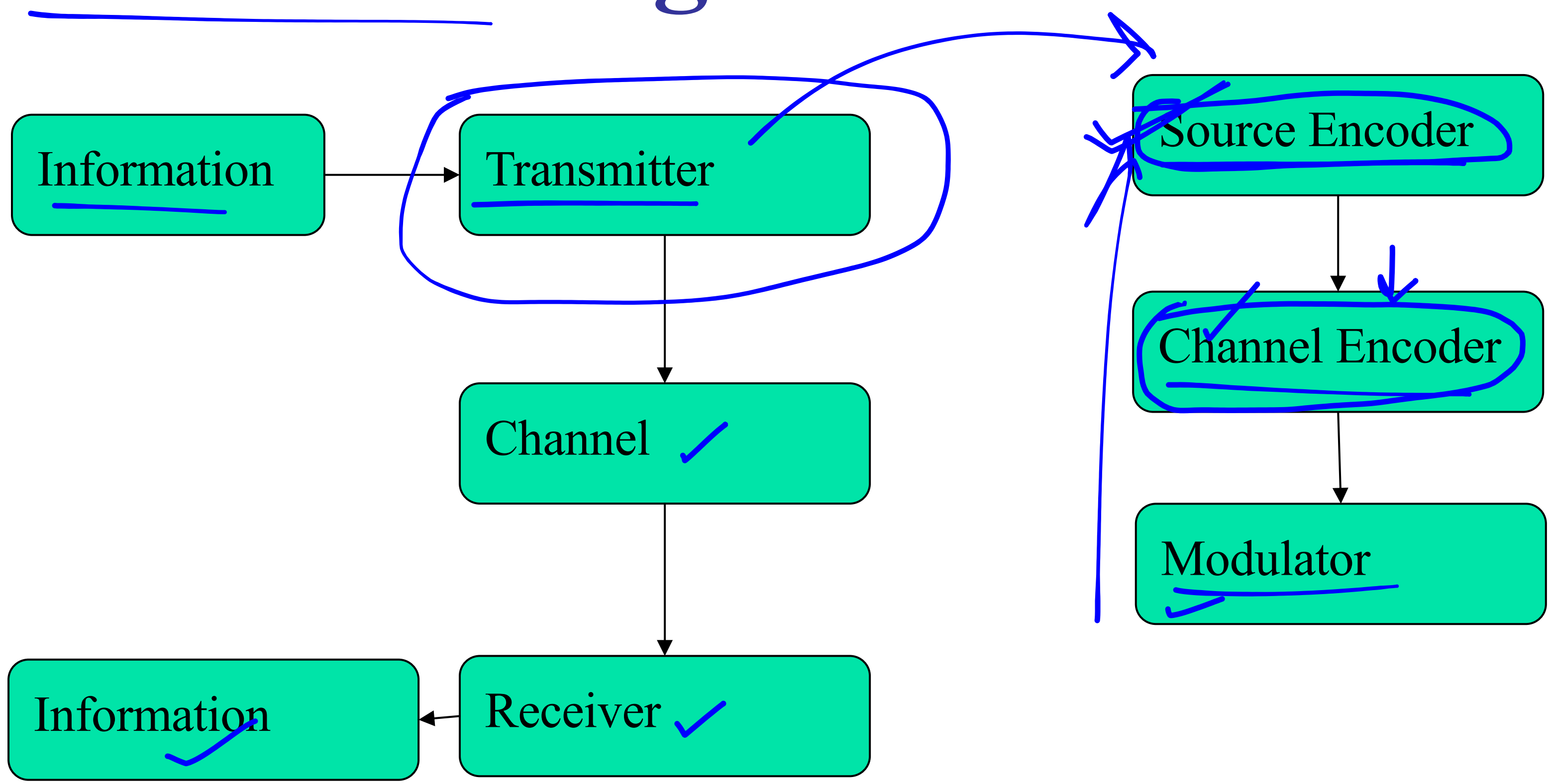
MLT-3

- Signal rate is same as NRZ-I
- But because of the resulting bit pattern, we have a periodic signal for worst case bit pattern: 1111
  - The signal pattern +V0-V0 repeated every 4 bits
  - So a nonperiodic signal has changed to a periodic signal with the period equal to 4 times the bit duration
- This can be approximated as an analog signal a frequency 1/4 the bit rate!
  - Signal rate is ¼ of the bit rate
- This makes MLT3 a suitable choice when we need to send 100Mbps on a copper wire that can not support more than 32 MHz

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



Sender

DW → data word

3

4

0001

Alg

CW → code word

DW

Checking

mBnL → 2B1Q

Block Coding

- For a code to be capable of ***error detection***, we **need to *add redundancy***, i.e., extra bits to the data bits.
- Synchronization also requires redundancy
  - transitions are important in the signal flow and must occur frequently.
- Block coding is done in three steps:
  - ***division, substitution and combination.***

3 4

- It is distinguished from multilevel coding by use of the slash
  - ***xB/yB.***

000 → 0001

1101

- The resulting bit stream *prevents certain bit combinations that when used with line encoding would result in DC components or poor sync. quality.*

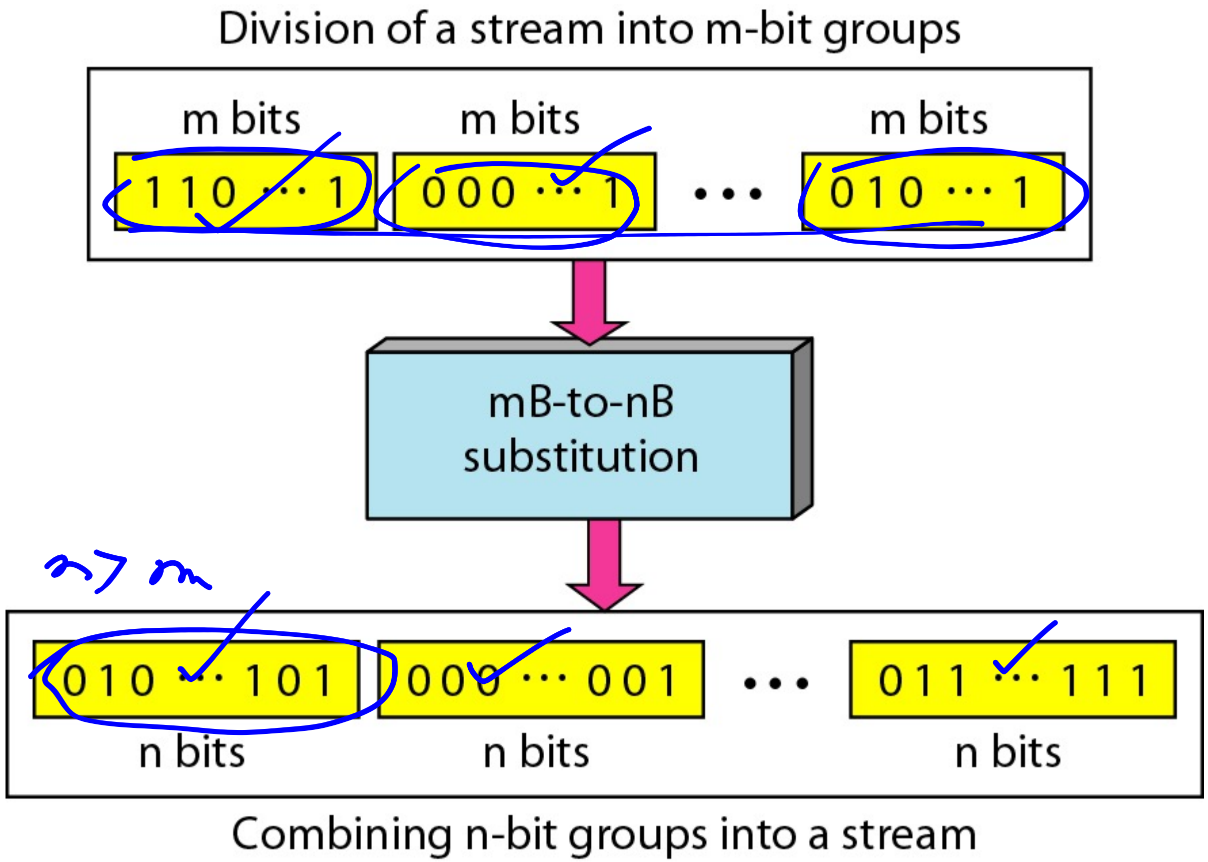
Note

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

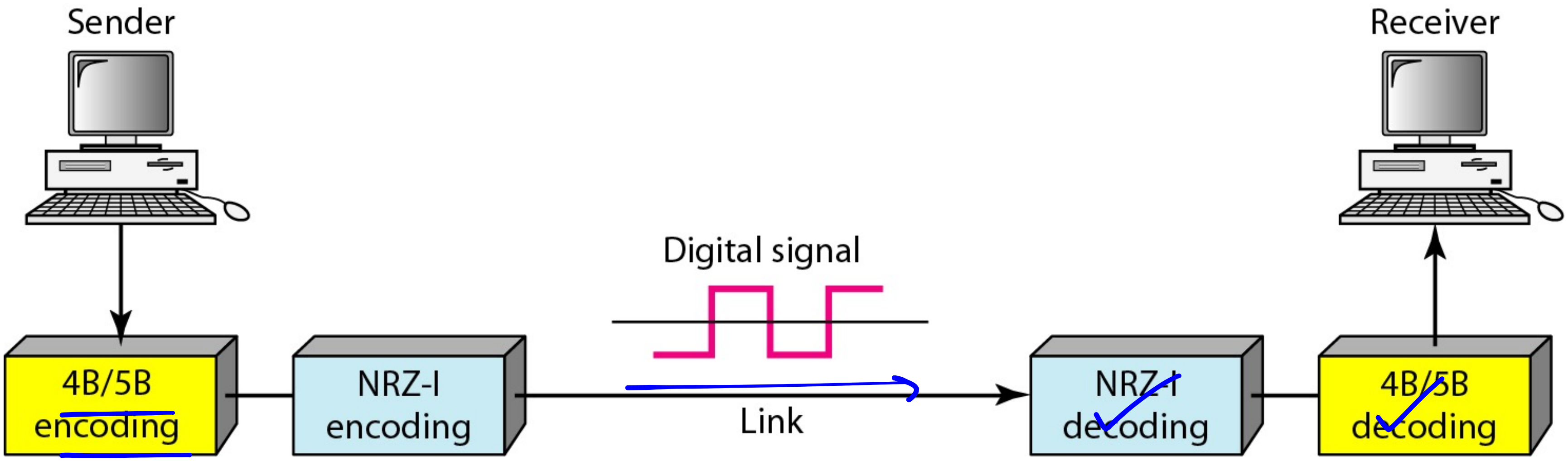
DW

Codeword.

Block coding concept



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

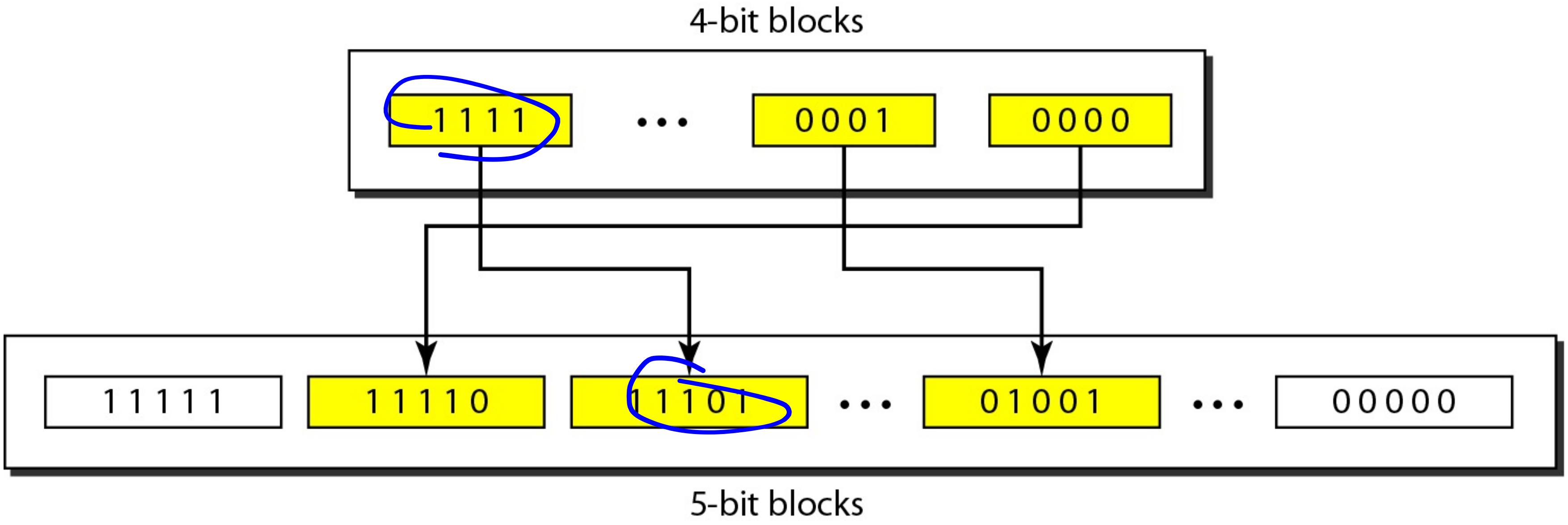


4B/5B mapping codes

$2^4 = 16$ $2^5 = 32$

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



Redundancy

- A 4 bit data word can have 2^4 combinations.
- A 5 bit word can have 2^5=32 combinations.
- We therefore have 32 - 26 = 16 extra words.
- Some of the extra words are used for control/signaling purposes.
- 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

4B/5B Encoding

- Solves the synchronization problem of NRZ -I (*no more than three consecutive 0s*)
- Increases the signal rate
  - 4 bits → 5 signals
  - Redundant bits add 20% overhead
  - Better than the Biphase scheme which has a twice the bandwidth of NRZ-I
- DC component

Example

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

Solution

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

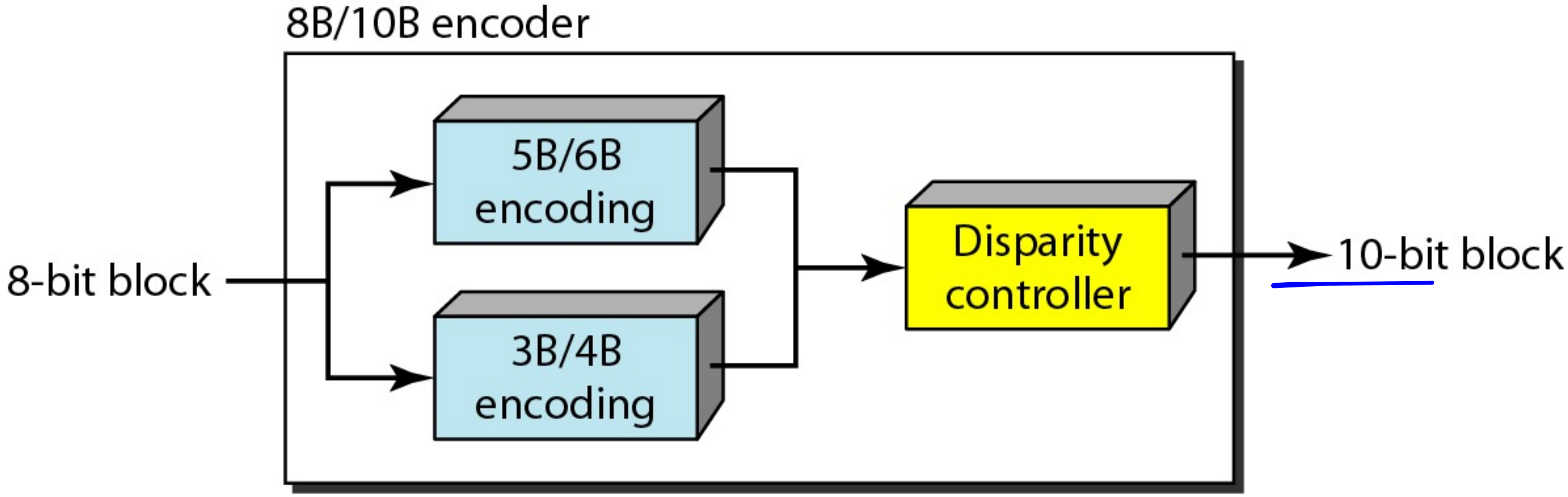
8B/10B Block Encoding

$2^8 = 256$

$2^{10} = 1024$

- Provides better error detection capability than 4B/5B
- A combination of 5B/6B and 3B/4B encoding
- The first 5 bits of. 8 bit block is fed into the 5B/6B encoder
- The last 3 bits is fed into a 3B/4B encoder
- Simplify the mapping $2^{10} \rightarrow 2^6 + 2^4 = 64+16$

8B/10B block encoding



8B/10B encoding

- **Disparity controller :**
  - Prevent a long run of consecutive 0s or 1s, by tracking the number of 0s and 1s.
  - If more 0s in the previous block and more 0s in the current block,
    - Each bit in the current block is complemented (0 → 1 and 1 → 0)

- The coding has $2^{10} - 2^{8} = 768$ redundant block codes for disparity checking and error detection.
- Better built-in error-checking capability and better synchronization than 4B/5B .

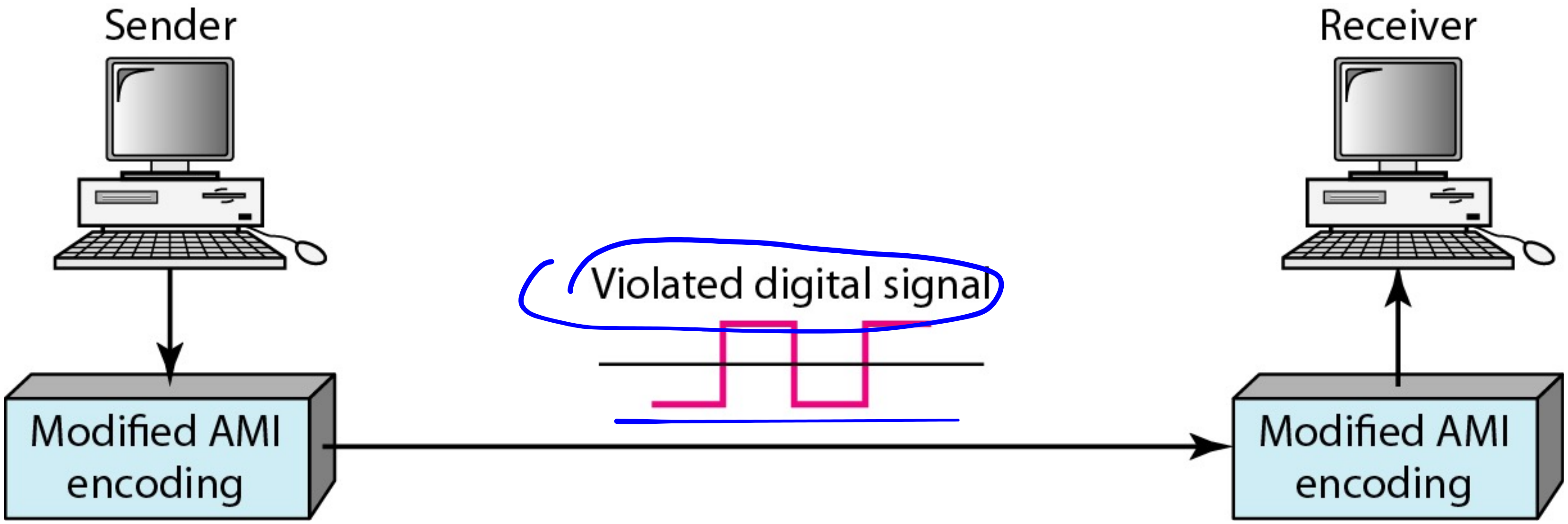
More bits - better error detection

- The 8B/10B 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.

Scrambling

- **Biphase** schemes that are suitable for dedicated links between stations in a LAN (self-sync and no DC),
  - but not suitable for long-distance communication due to *wide bandwidth requirement.*
- **Block coding (4B/5B) + NRZ-I** line coding is not suitable for long distance encoding, due to DC component problem.
- **Bipolar AMI** encoding has a narrow bandwidth, and does not create a *DC component problem*
  - However, a long sequence of "0"s *destroys synchronization*.
- If we can find a way to avoid a long sequence of 0s in the original bit stream, we can use bipolar AMI for long distances.
- One solution is called ***scrambling.***

AMI used with scrambling



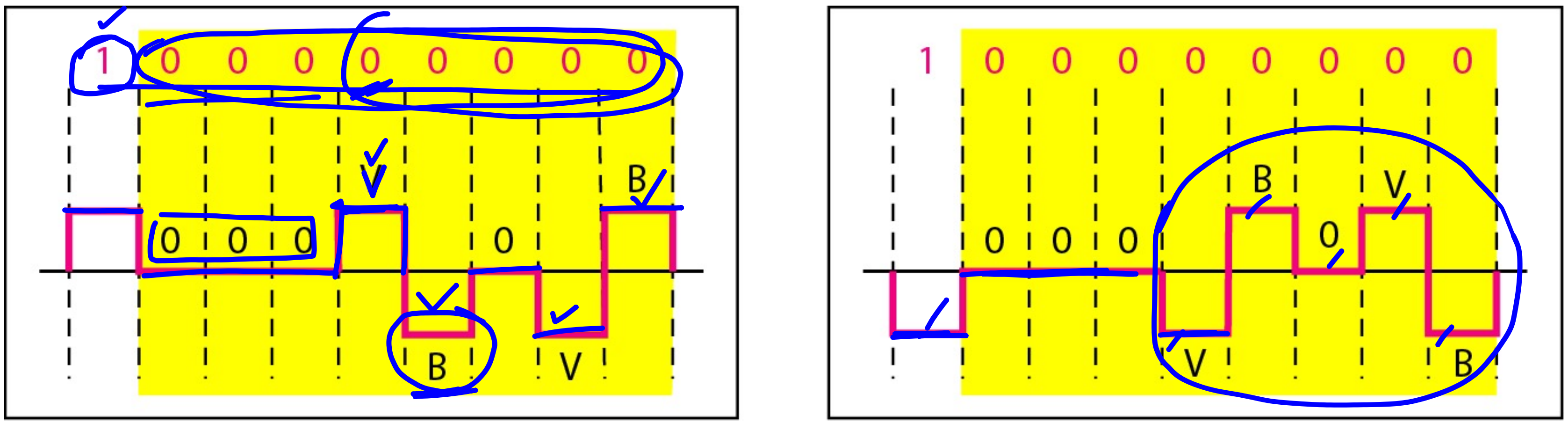
B8ZS

- Bipolar with 8-zero substitution (B8ZS)
  - Commonly used in North America
  - Eight consecutive zero-level voltages are replaced by the sequence 000VB0VB.
  - "V" in the sequence denotes violation; that is a nonzero voltage that breaks the original AMI rule of encoding.
  - "B" in the sequence denotes bipolar, which means a nonzero level voltage according to the AMI rule

Two cases of B8ZS scrambling technique

AMI Rule:

- 0: Zero voltage level
- 1: Alternating positive and negative voltages



a. Previous level is positive.   b. Previous level is negative.

- Scrambling in this case does not change the signal rate. (# of signal is the same.)
- DC balance is maintained (two positives and two negatives in 000VB0VB)

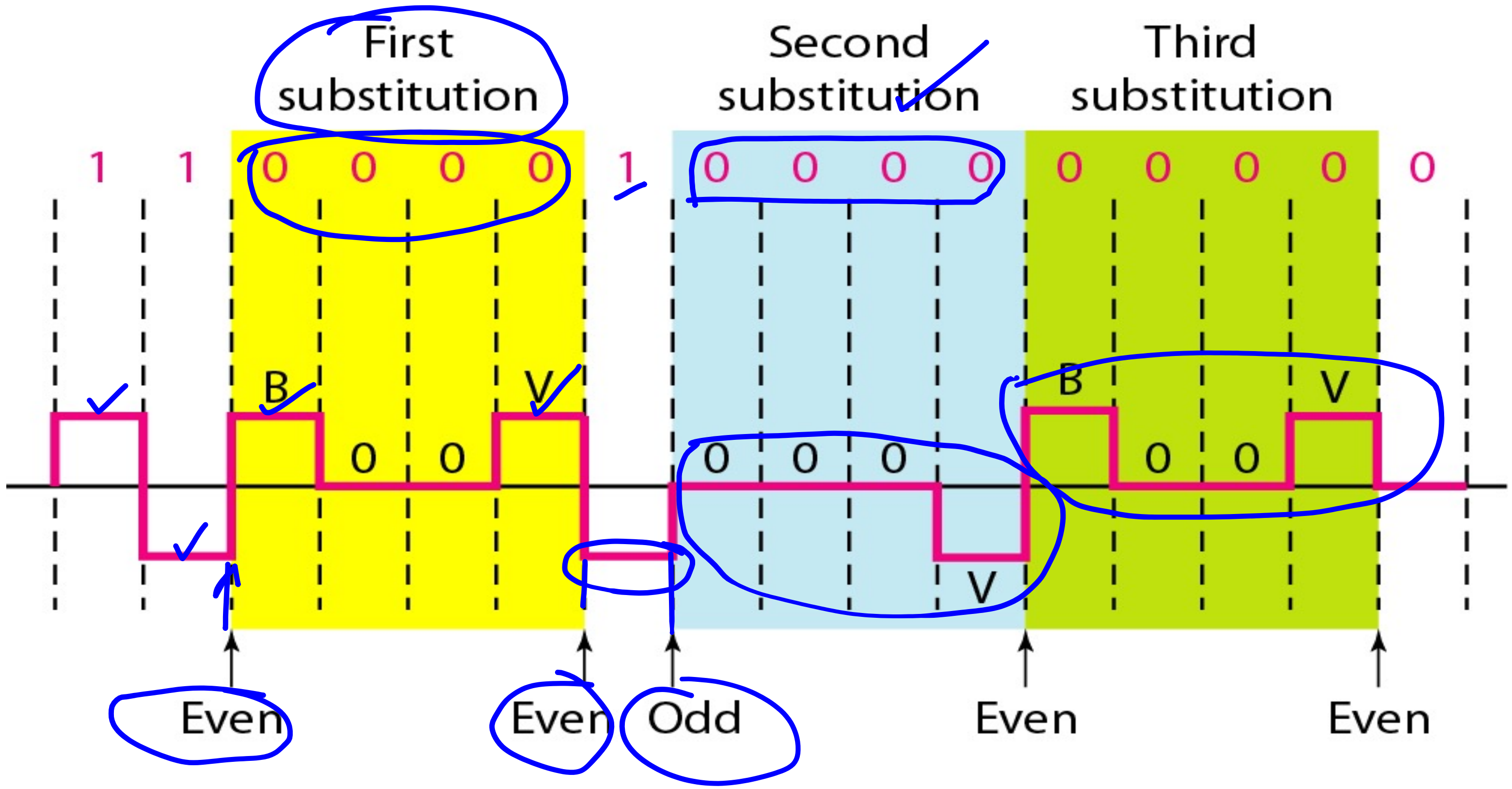
Note

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

High Density Bipolar 3 Zeros (HDB3)

- Used outside of North America
- Four consecutive zero-level voltages are replaced with 000V or B00V.
  - If # of nonzero pulses after the last substitution is odd, the substitution pattern is 000V.
  - If # of nonzero pulses after the last substitution is even, the substitution pattern will be B00V.
- By doing so, the total number of nonzero pulses is always even.

Different situations in HDB3 scrambling technique



Note

- **HDB3 substitutes four consecutive zeros with 000V or B00V depending on the number of nonzero pulses after the last substitution.**
  - **If # of non zero pulses is even the substitution is B00V to make total # of non zero pulse even.**
  - **If # of non zero pulses is odd the substitution is 000V to make total # of non zero pulses even.**

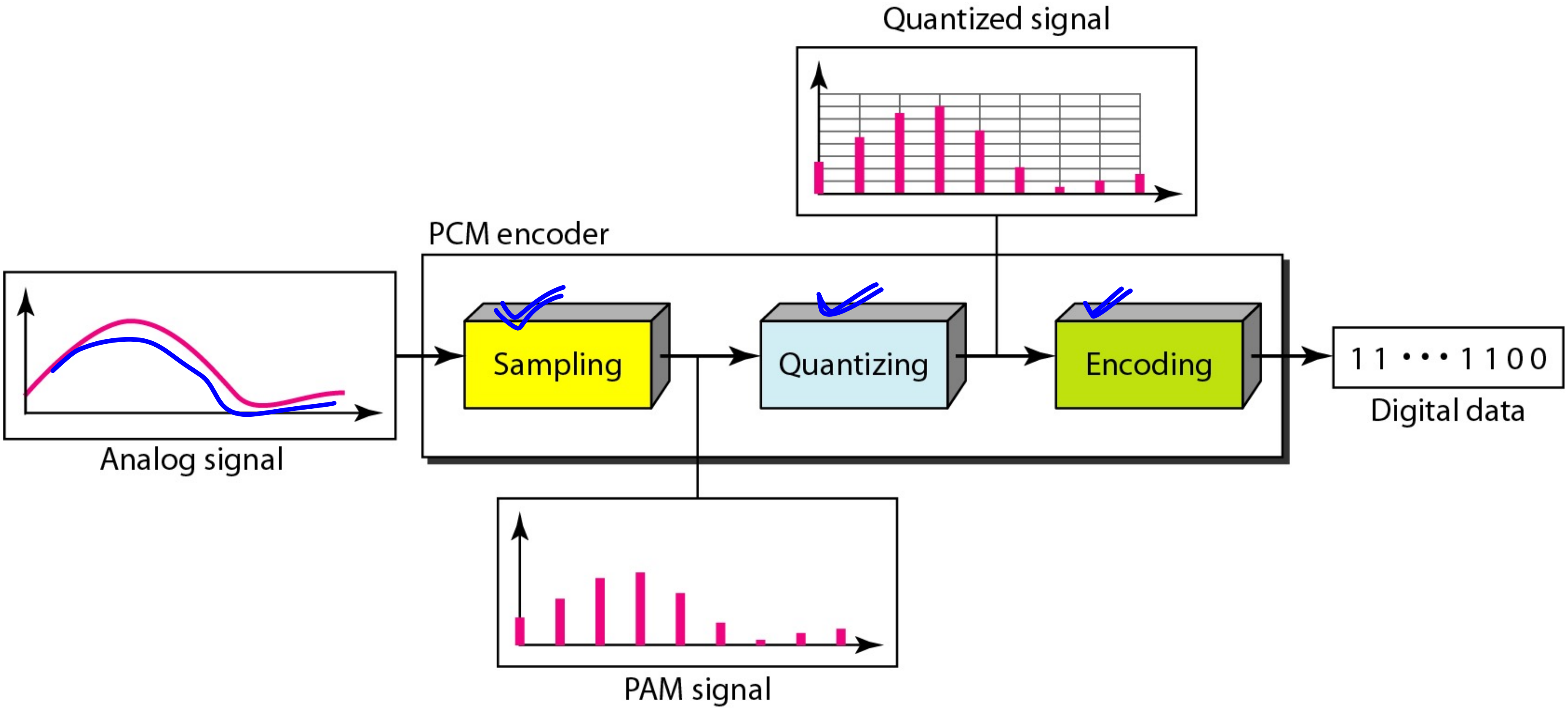
Analog-To-Digital Conversion

- ***A digital signal is superior to an analog signal because***
  - ***It is more robust to noise and can easily be recovered, corrected and amplified.***
- ***Two techniques for conversion:***
  - **Pulse Code Modulation (PCM)**
  - **Delta Modulation (DM)**

PCM

- PCM consists of three steps to digitize an analog signal:
  1. Sampling
  2. Quantization
  3. Binary encoding
- Before we sample, we have to filter the signal to limit the maximum frequency of the signal as it affects the sampling rate.
- Filtering should ensure that we do not distort the signal, i.e. remove high frequency components that affect the signal shape.

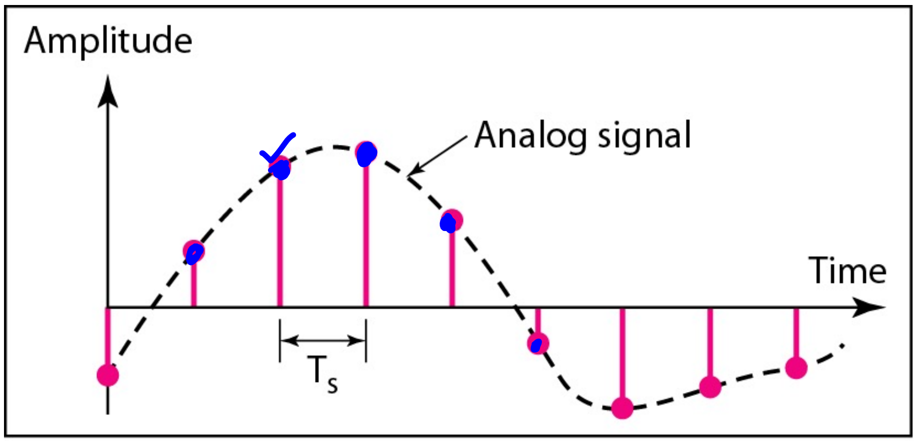
Components of PCM encoder



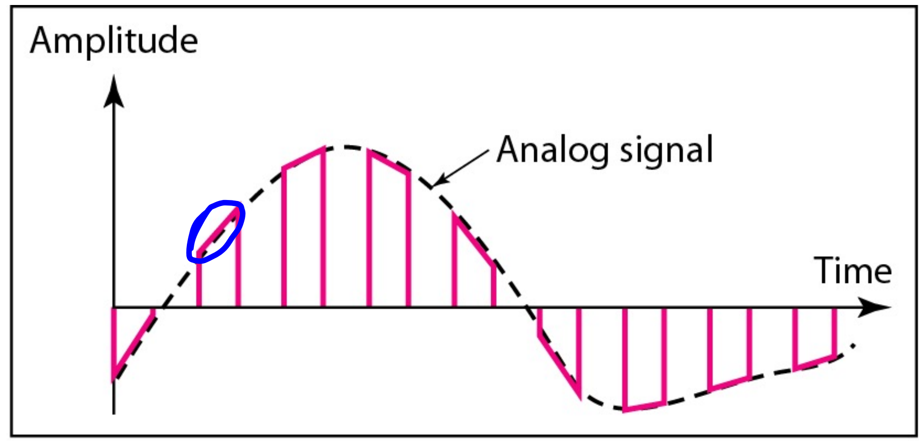
Sampling

- Analog signal is sampled every T_S secs.
- T_s is referred to as the sampling interval.
- $f_s = 1/T_s$ is called the sampling rate or sampling frequency.
- There are 3 sampling methods:
  - Ideal - an impulse at each sampling instant
  - Natural - a pulse of short width with varying amplitude
  - Flattop - sample and hold, like natural but with single amplitude value
- The process is referred to as pulse amplitude modulation PAM and the outcome is a signal with analog (non integer) values

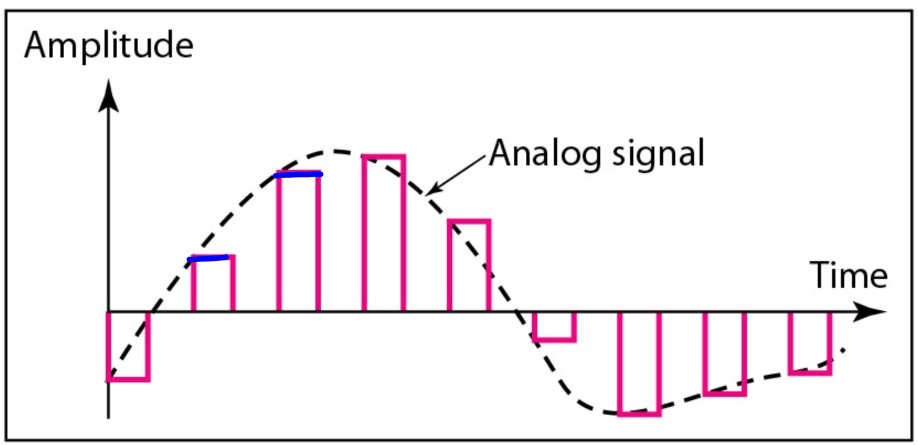
Three different sampling methods for PCM



a. Ideal sampling



b. Natural sampling

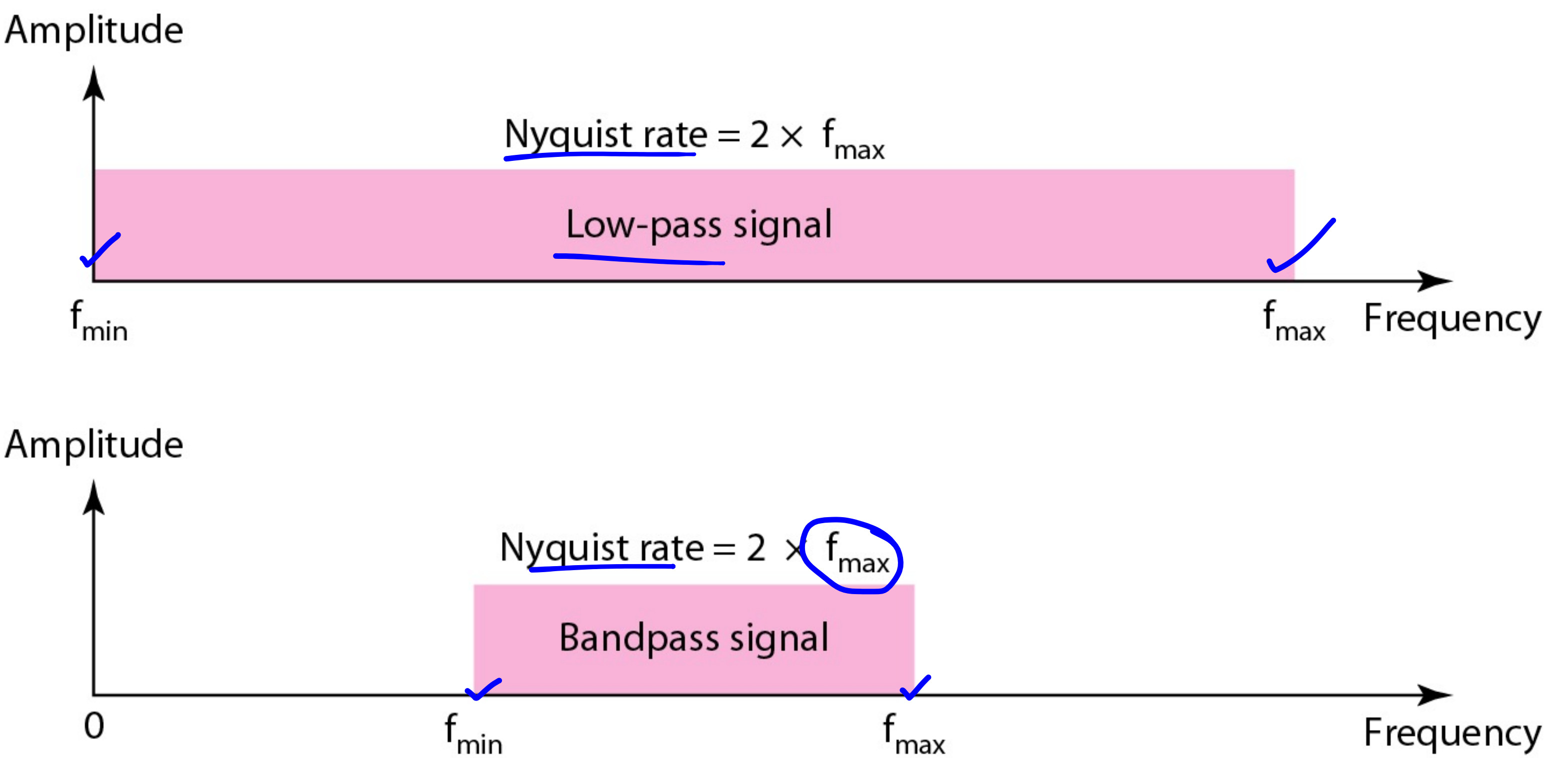


c. Flat-top sampling

Note

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

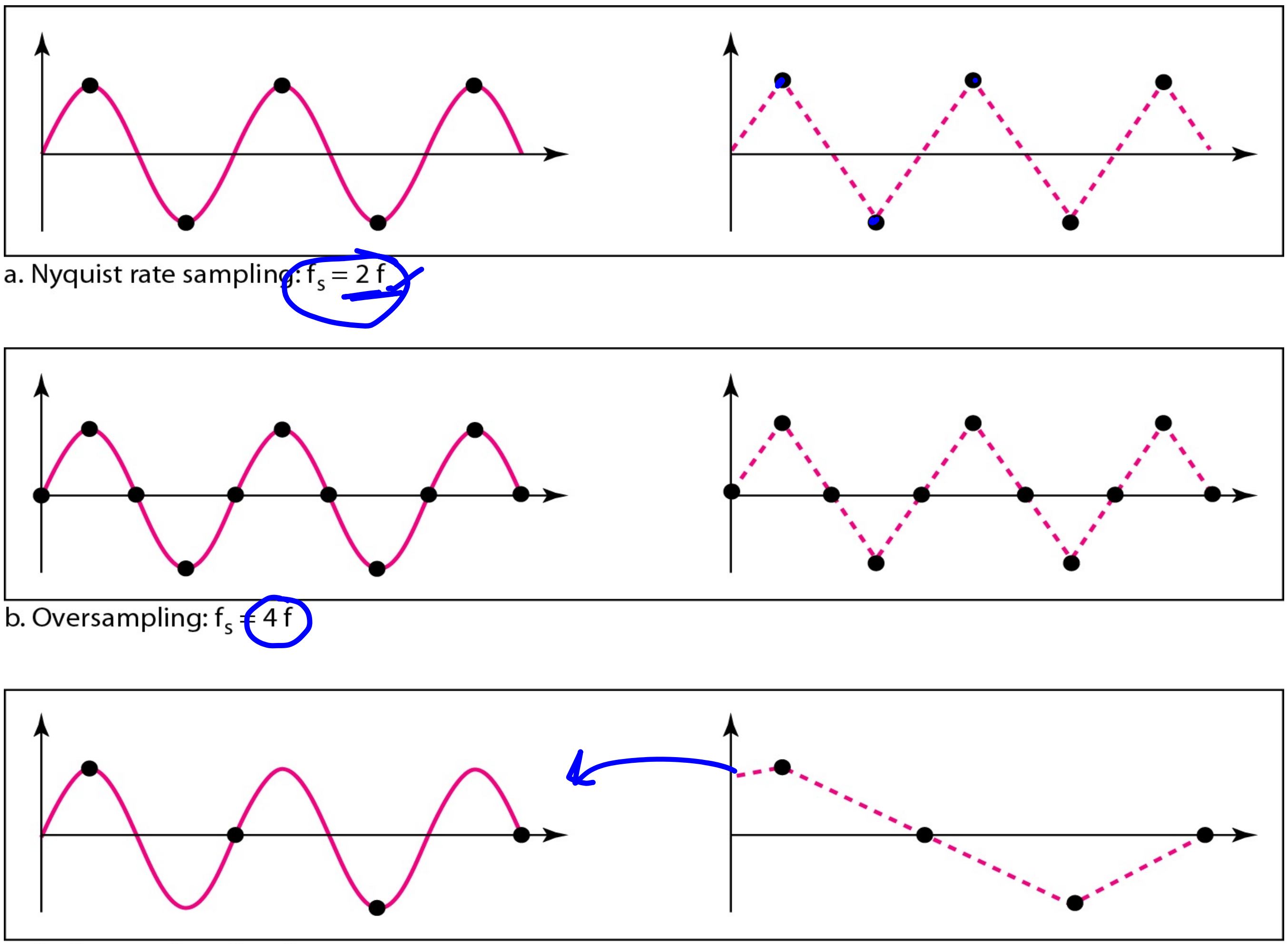
Nyquist sampling rate for low-pass and bandpass signals



Example

- ***For an intuitive example of the Nyquist theorem, let us sample a simple sine wave at three sampling rates: f_s = 4f (2 times the Nyquist rate), f_s = 2f (Nyquist rate), and f_s = f (one-half the Nyquist rate).***
  - ***It can be seen that sampling at the Nyquist rate can create a good approximation of the original sine wave (part a).***
  - ***Oversampling in part b can also create the same approximation, but it is redundant and unnecessary.***
  - ***Sampling below the Nyquist rate (part c) does not produce a signal that looks like the original sine wave.***

Recovery of a sampled sine wave for different sampling rates



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

b. Oversampling: $f_s = 4f$

c. Undersampling: $f_s = f$

Example

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

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.

Quantization

- Sampling results in a series of pulses of varying amplitude values ranging between two limits: a min and a max.
- The amplitude values are infinite between the two limits.
- We need to map the *infinite* amplitude values onto a finite set of known values.
- This is achieved by dividing the distance between min and max into L zones, each of height Δ.

$$\Delta = (\max - \min)/L$$

Quantization Levels

- The midpoint of each zone is assigned a value from 0 to L-1 (resulting in L values)
- Each sample falling in a zone is then approximated to the value of the midpoint.

Quantization Zones

- Assume we have a voltage signal with amplitutes V_{min}=-20V and V_{max}=+20V.
- We want to use L=8 quantization levels.
- Zone width Δ = (20 - -20)/8 = 5
- The 8 zones are: -20 to -15, -15 to -10, -10 to -5, -5 to 0, 0 to +5, +5 to +10, +10 to +15, +15 to +20
- The midpoints are: -17.5, -12.5, -7.5, -2.5, 2.5, 7.5, 12.5, 17.5

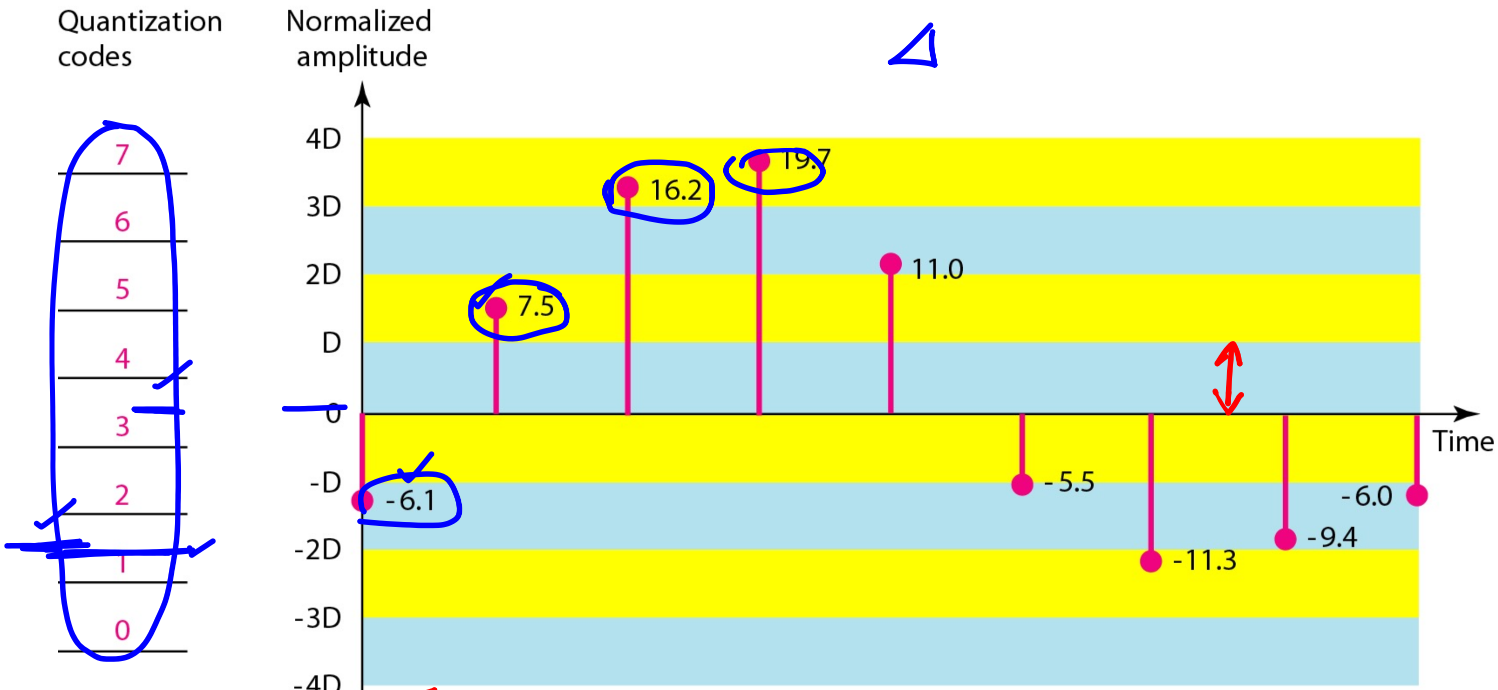
Assigning Codes to Zones

- Each zone is then assigned a binary code.
- The number of bits required to encode the zones, or the number of bits per sample as it is commonly referred to, is obtained as follows:

$$\boxed{n_b = \log_2 L}$$

- Given our example, $n_b = 3$
- The 8 zone (or level) codes are therefore: 000, 001, 010, 011, 100, 101, 110, and 111
- Assigning codes to zones:
  - 000 will refer to zone -20 to -15
  - 001 to zone -15 to -10, etc.

Quantization and encoding of a sampled signal



Normalized PAM values	-1.22	1.50	3.24	3.94	2.20	-1.10	-2.26	-1.88	-1.20
Normalized quantized values	-1.50	1.50	3.50	3.50	2.50	-1.50	-2.50	-1.50	-1.50
Normalized error	-0.38	0	+0.26	-0.44	+0.30	-0.40	-0.24	+0.38	-0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

Quantization Error

- When a signal is quantized, we introduce an error
  - the coded signal is an approximation of the actual amplitude value.
- The difference between actual and coded value (midpoint) is referred to as the quantization error.
- The more zones, the smaller Δ which results in smaller errors.
- **BUT,** the more zones the more bits required to encode the samples → higher bit rate

Quantization Error and SN_QR

- Signals with lower amplitude values will suffer more from quantization error as the error range: $\Delta/2$, is fixed for all signal levels.
- Non linear quantization is used to alleviate this problem. Goal is to keep SN_QR fixed for all sample values.
- Two approaches:
  - The quantization levels follow a logarithmic curve. Smaller Δ's at lower amplitudes and larger Δ's at higher amplitudes.
  - **Companding and expanding:** The sample values are compressed at the sender into logarithmic zones, and then expanded at the receiver. The zones are fixed in height.

Bit rate and bandwidth requirements of PCM

- The bit rate of a PCM signal can be calculated form the number of bits per sample x the sampling rate

$$\textit{Bit rate} = n_b \times f_s$$

- The bandwidth required to transmit this signal depends on the type of line encoding used.
- A digitized signal will always need more bandwidth than the original analog signal.
- Price we pay for robustness and other features of digital transmission.

Example

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

Solution

- The human voice normally contains frequencies from 0 to 4000 Hz.

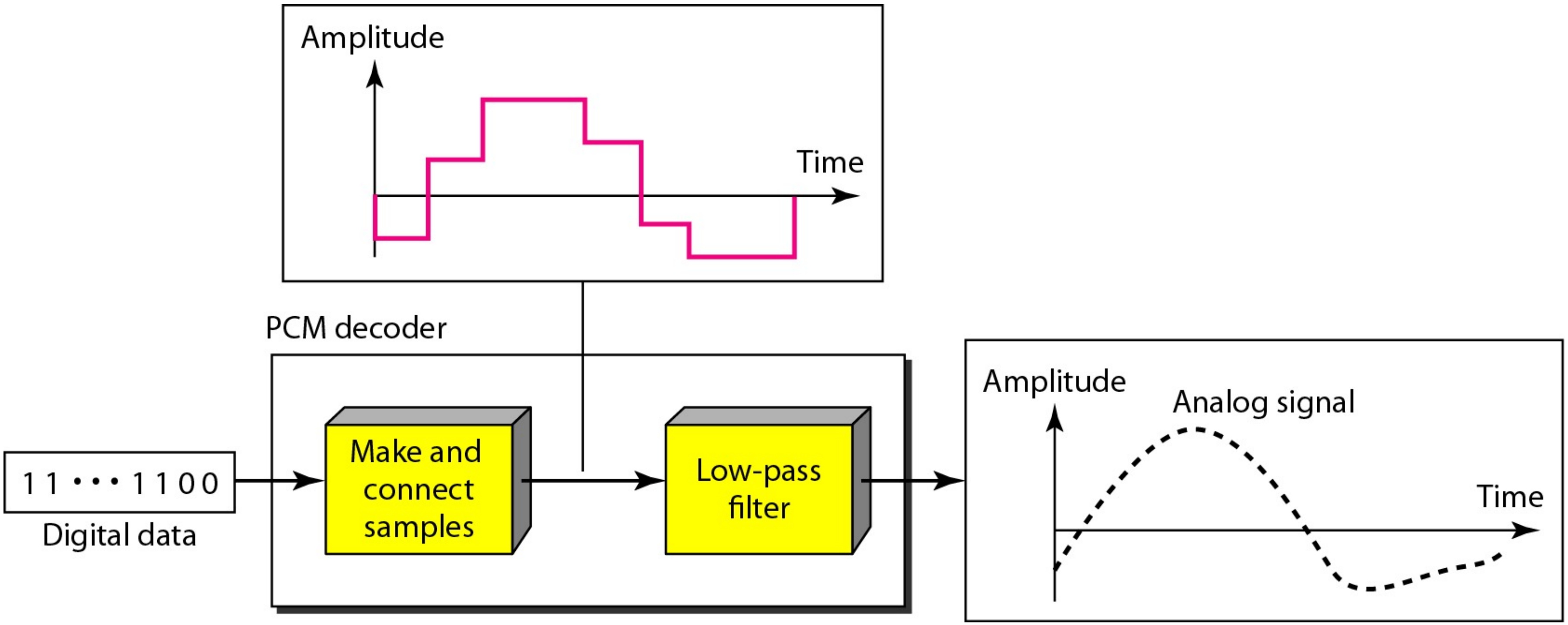
- So the sampling rate and bit rate are calculated as follows:

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

PCM Decoder

- To recover an analog signal from a digitized signal we follow the following steps:
  - We use a hold circuit that holds the amplitude value of a pulse till the next pulse arrives.
  - We pass this signal through a low pass filter with a cutoff frequency that is equal to the highest frequency in the pre-sampled signal.
- The higher the value of L, the less distorted a signal is recovered.

Components of a PCM decoder



Example

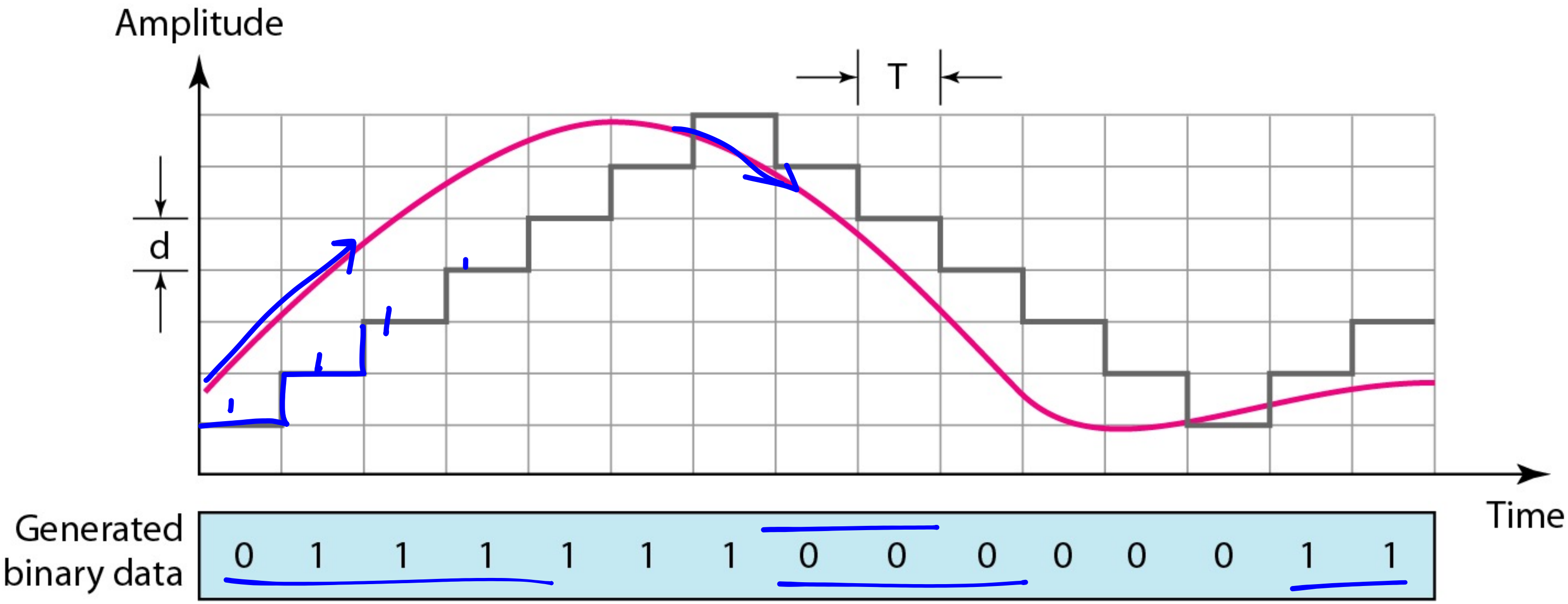
We have a low-pass analog signal of 4 kHz. If we send the analog signal, we need a channel with a minimum bandwidth of 4 kHz.

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

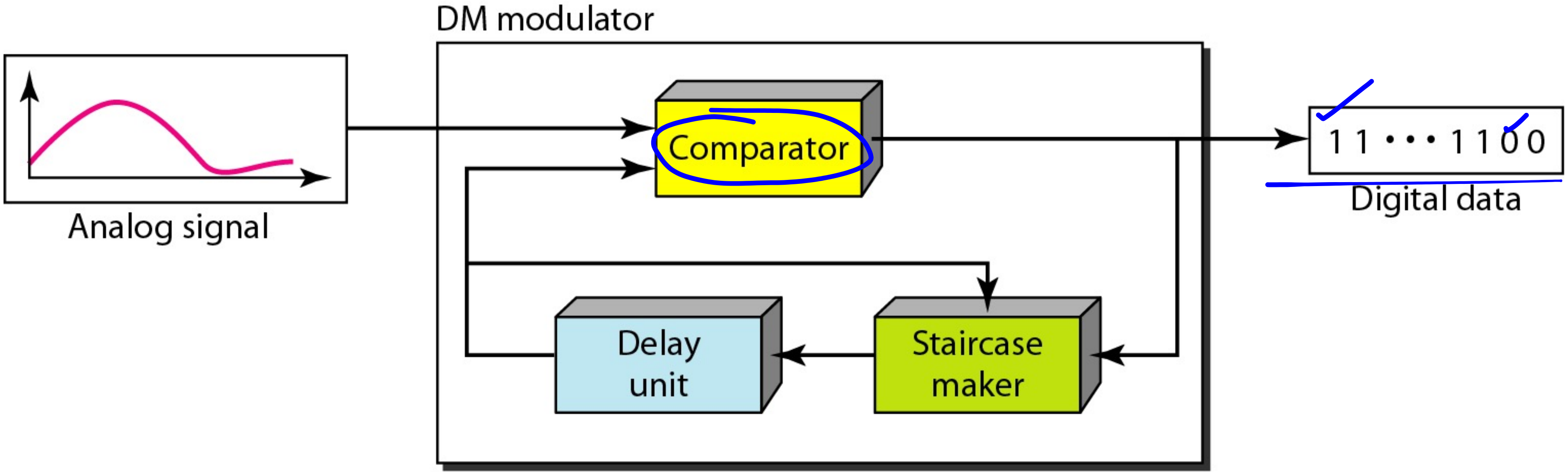
Delta Modulation

- This scheme sends only the difference between pulses
  - if the pulse at time t_{n+1} is higher in amplitude value than the pulse at time t_n,
    - then a single bit, say a "1", is used to indicate the positive value.
  - If the pulse is lower in value, resulting in a negative value, a "0" is used.
- This scheme works well for small changes in signal values between samples.
- If changes in amplitude are large, this will result in large errors.

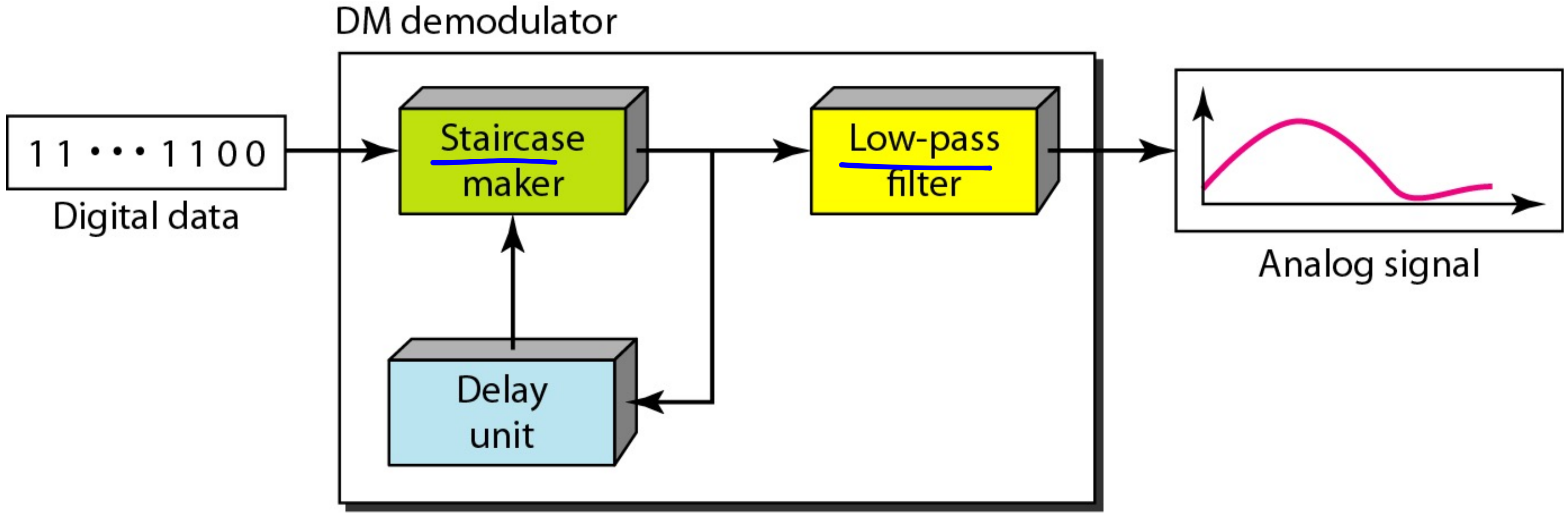
The process of delta modulation



Delta modulation components



Delta demodulation components

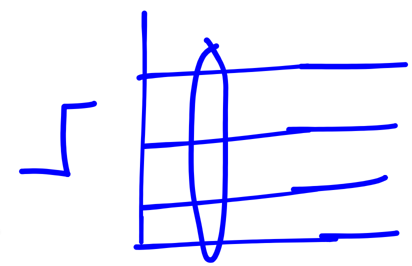


Delta PCM (DPCM)

- Instead of using one bit to indicate positive and negative differences, we can use more bits
  - quantization of the difference.
- Each bit code is used to represent the value of the difference.
- The more bits the more levels → the higher the accuracy.

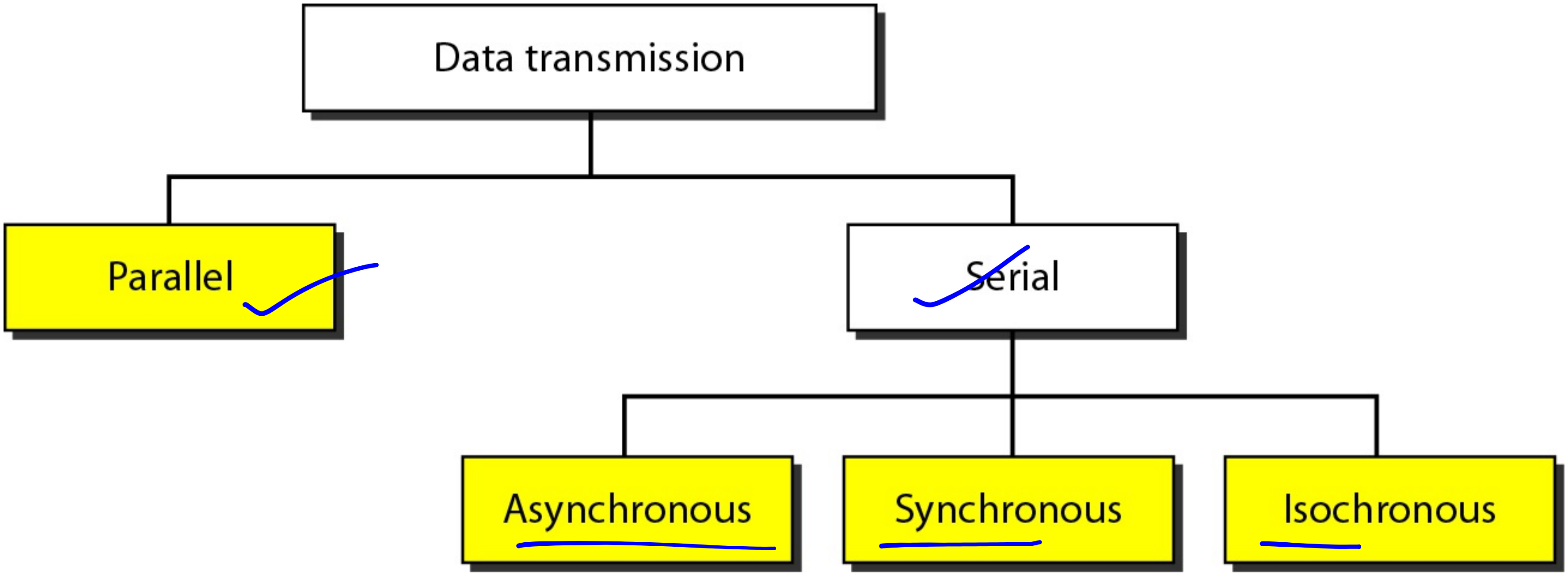
Transmission Modes

- The transmission of binary data across a link can be accomplished in either ***parallel or serial mode.***
- In parallel mode, multiple bits are sent with each clock tick.
- In serial mode, 1 bit is sent with each clock tick.

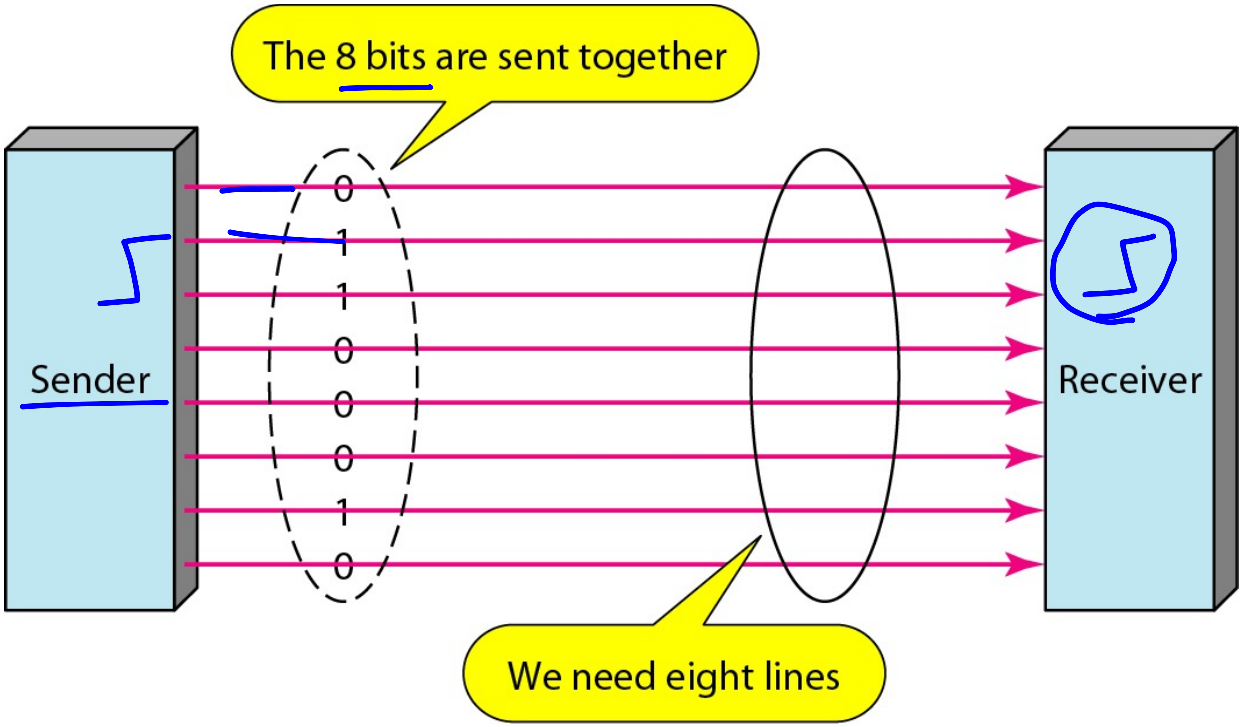


- While there is only one way to send parallel data
- There are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.

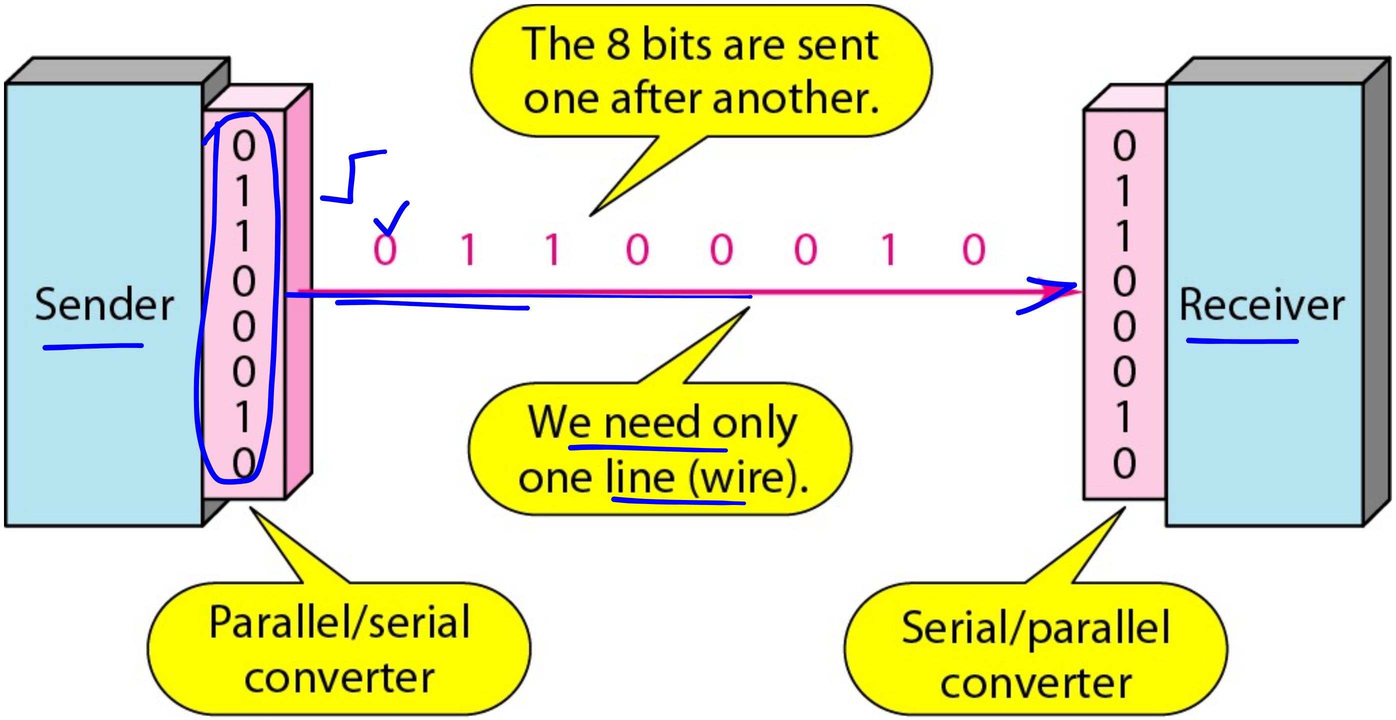
Data transmission and modes



Parallel transmission



Serial transmission



Note

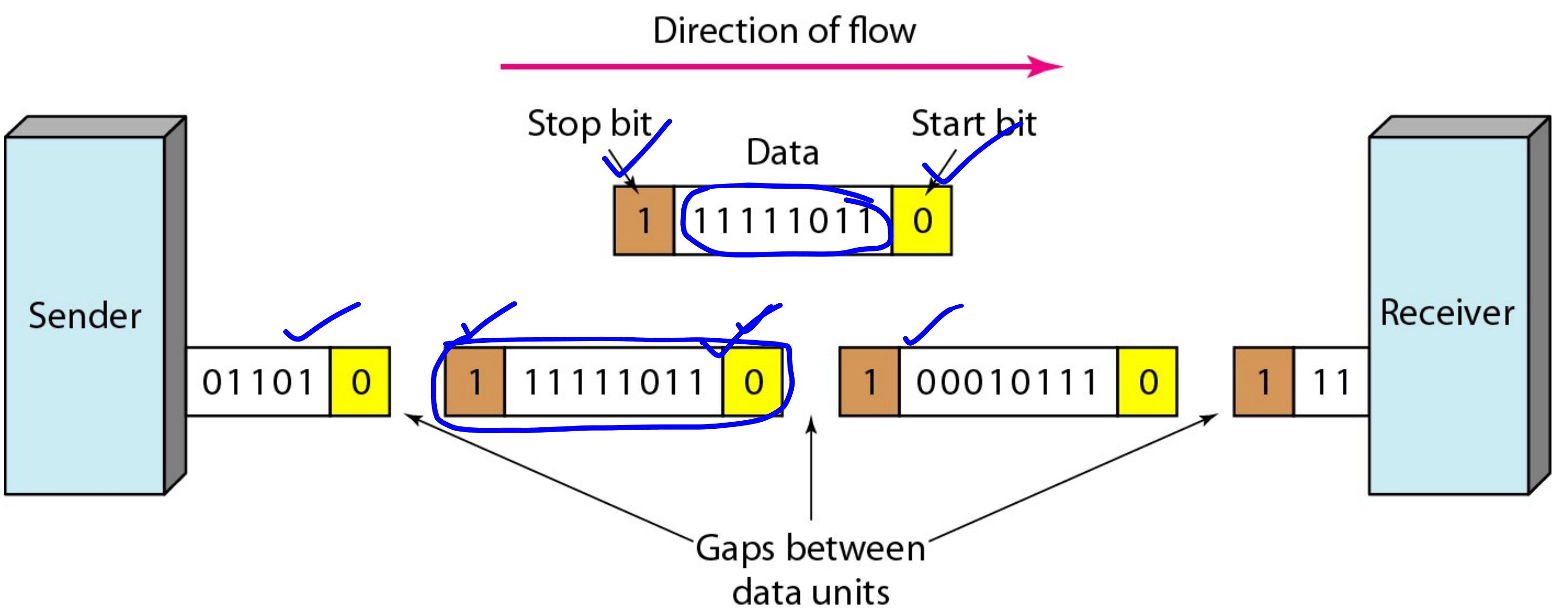
- In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte.

- There may be a gap between each byte.

Note

Asynchronous here means "asynchronous at the byte level," but the bits are still synchronized; their durations are the same.

Asynchronous transmission



Note

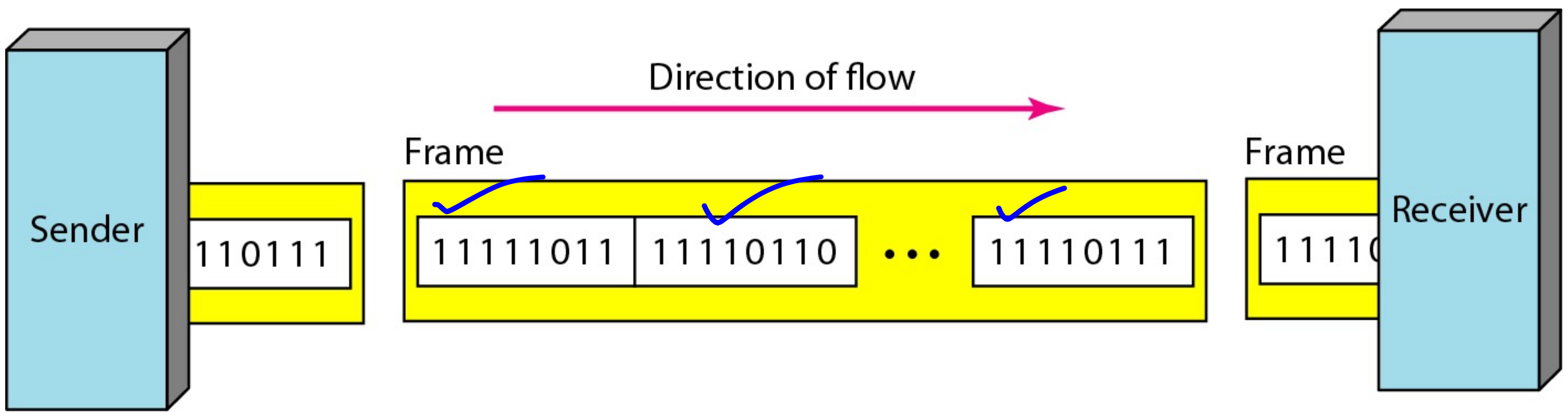
- In synchronous transmission, we send bits one after another without start or stop bits or gaps.

- It is the responsibility of the receiver to group the bits.

- The bits are usually sent as bytes and many bytes are grouped in a frame.

- A frame is identified with a start and an end byte.

Synchronous transmission



Isochronous

- In isochronous transmission we cannot have uneven gaps between frames.
- Transmission of bits is fixed with equal gaps.