Chapter 4

Digital Transmission

4-1 DIGITAL-TO-DIGITAL CONVERSION

In this section, we see how we can 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.

Topics discussed in this section:

- **Line Coding**
- **Line Coding Schemes**
- **Block Coding**
- **Scrambling**

Line Coding

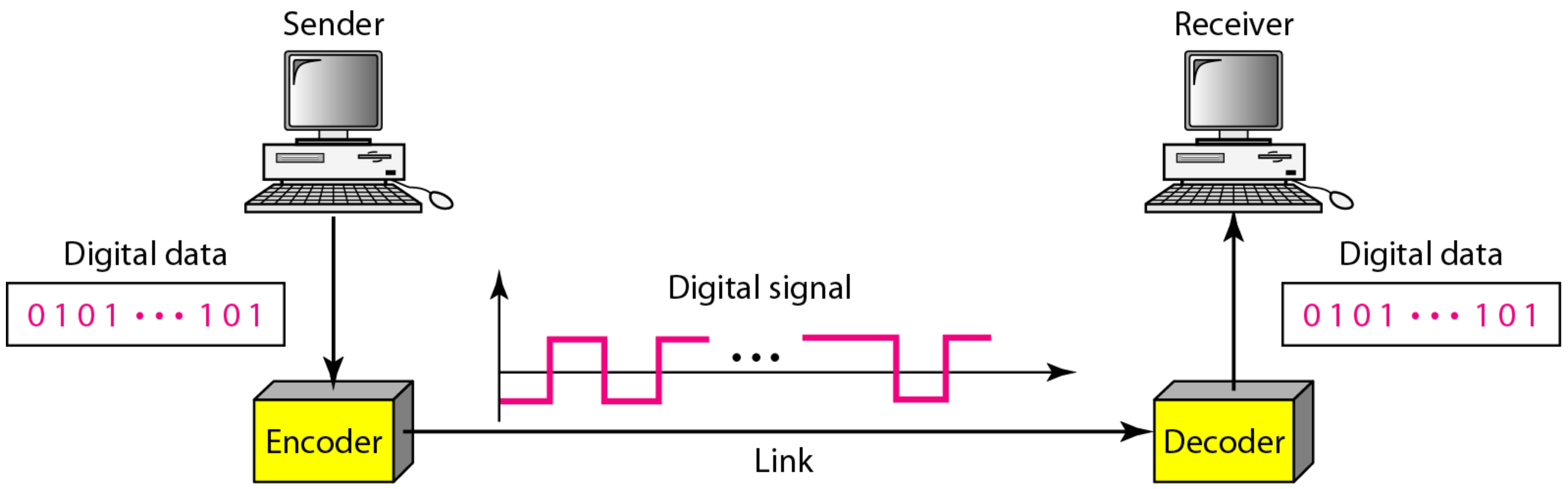
- ***Line coding is the process of converting digital data to digital signals.***

- ***We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits.***

- ***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.***

Figure 4.1 *Line coding and decoding*



Characteristics of Line coding

Some characteristics of line coding are

- ***Signal level versus data level***
- ***Pulse rate vs bit rate***
- ***Dc components and***
- ***Self-synchronization***

Data Element

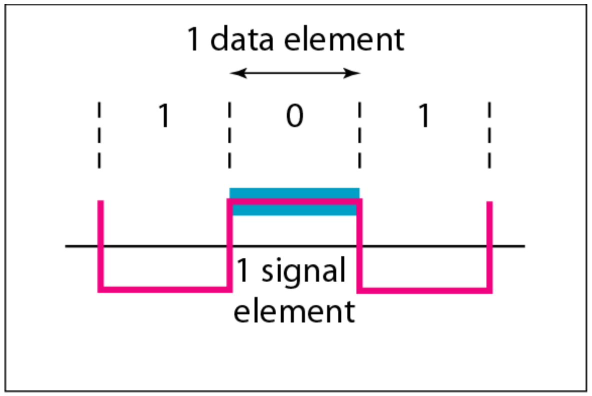
- *A data element is the smallest entity that can represent a piece of information: this is the bit.*

- *In digital data communications, a signal element carries data elements.*

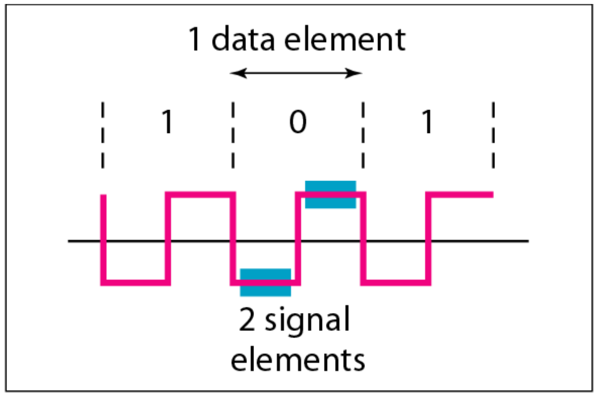
Signal Element

- *A signal element is the shortest unit (timewise) of a digital signal.*
- *In other words, 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.*

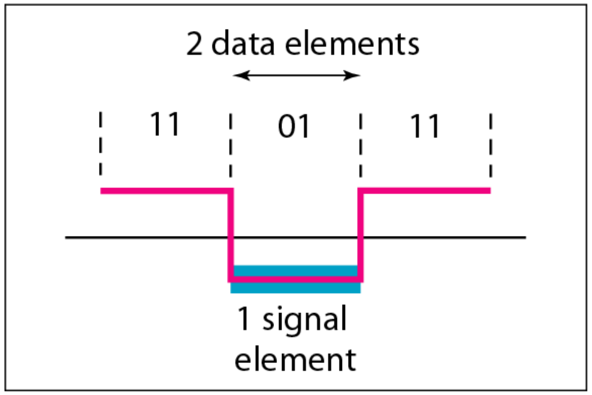
Figure 4.2 *Signal element versus data element*



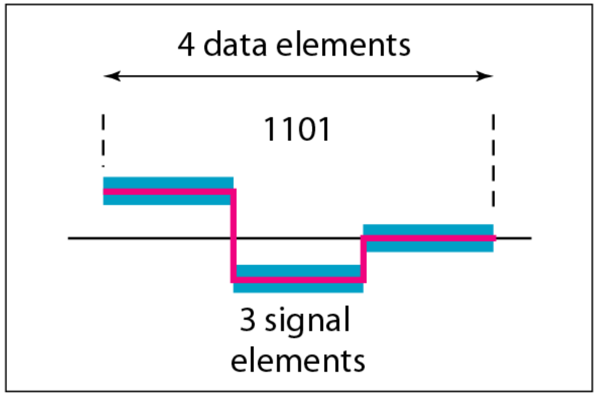
a. One data element per one signal element (r = 1)



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 Versus Signal Rate

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

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

•The data rate is sometimes called the bit rate;

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

DC Components

•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, present problems for a system that cannot pass low frequencies.

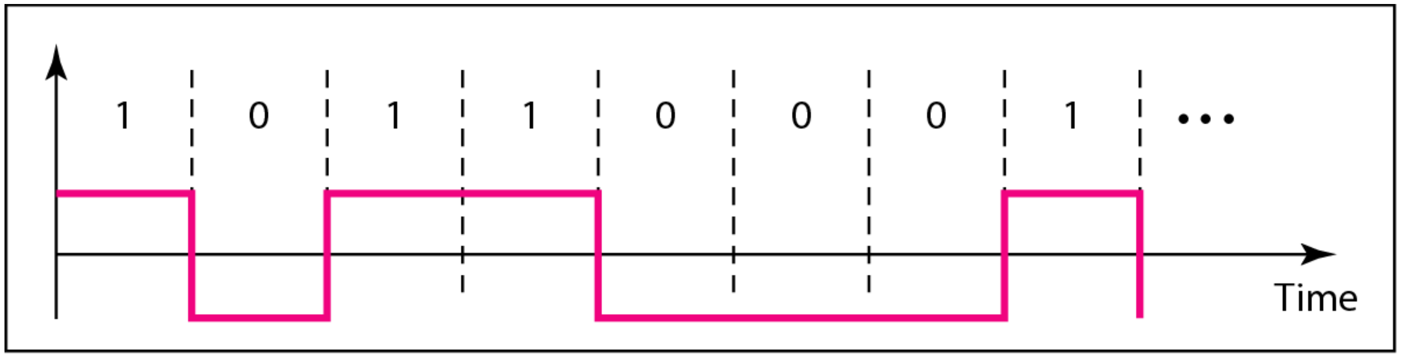
•For example, a telephone line cannot pass frequencies below 200 Hz.

Self-synchronization

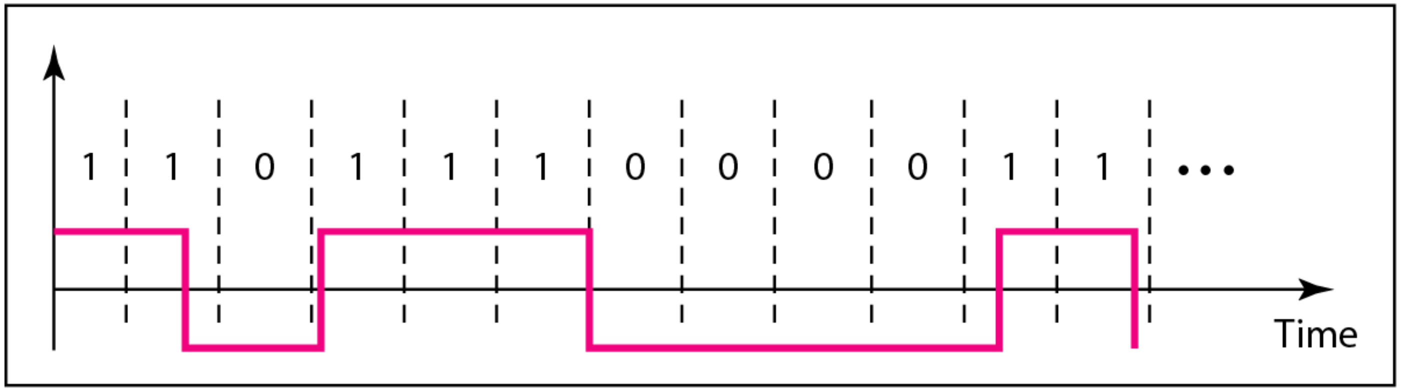
•To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals.

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

Figure 4.3 *Effect of lack of synchronization*

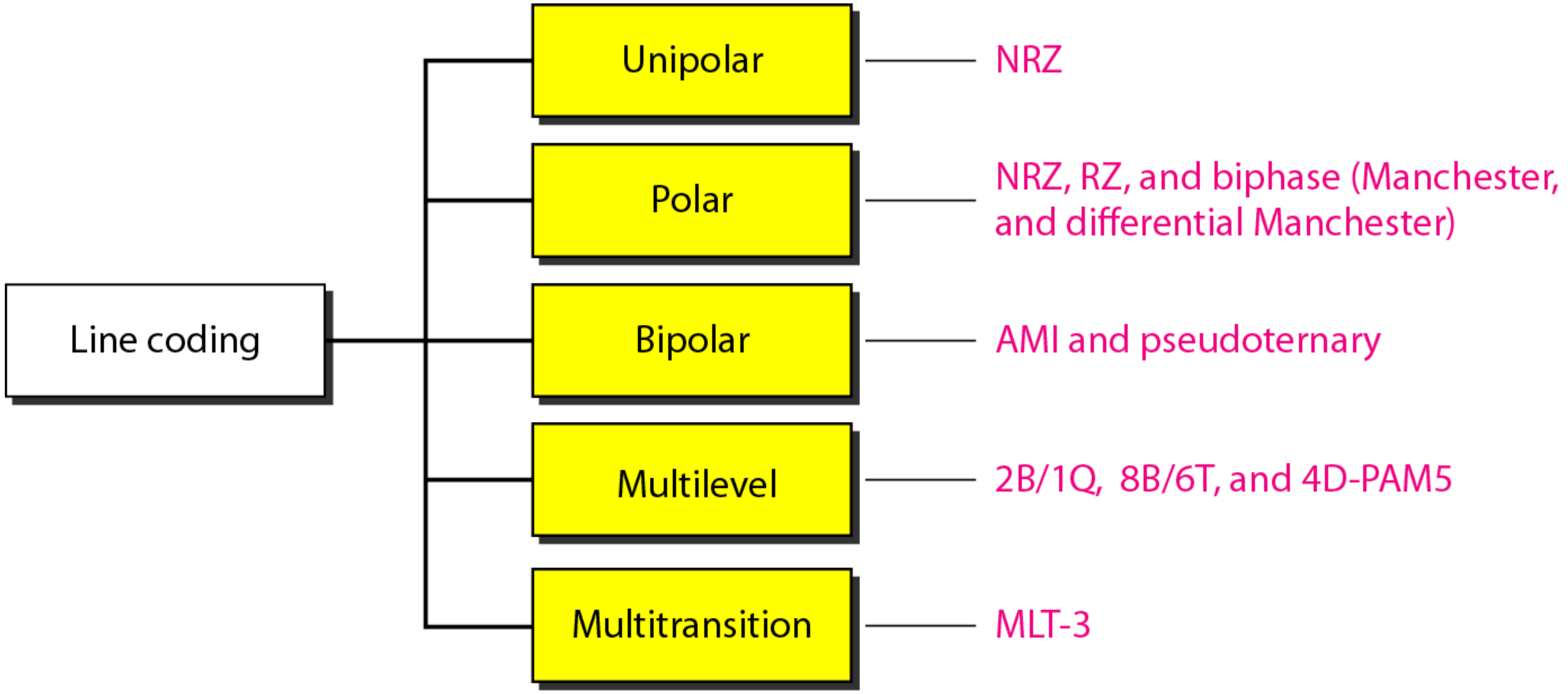


a. Sent



b. Received

Figure 4.4 *Line coding schemes*


Line coding
Unipolar
Polar
Bipolar
Multilevel
Multitransition
NRZ
NRZ, RZ, and biphase (Manchester, and differential Manchester)
AMI and pseudoternary
2B/1Q, 8B/6T, and 4D-PAM5
MLT-3

Unipolar Scheme

•In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.

NRZ (Non-Return-to-Zero)

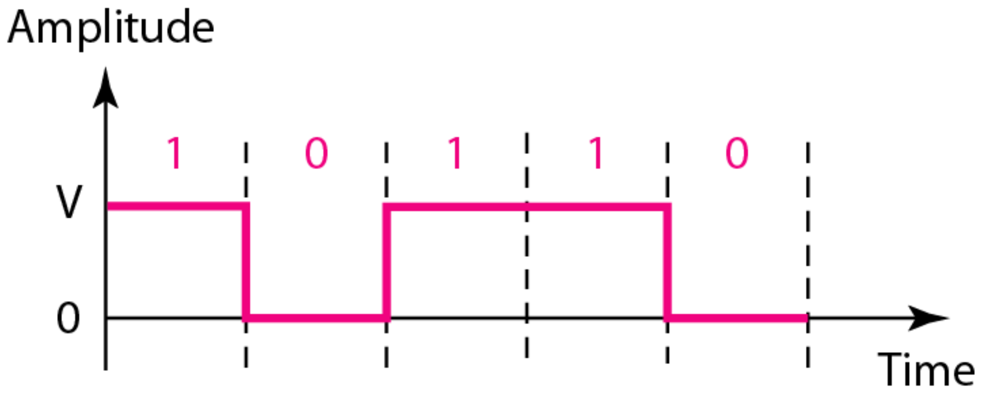
•Traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0.

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

•It has no synchronization or any error detection.

• It is simple but costly in power consumption.

Figure 4.5 *Unipolar NRZ scheme*



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

Normalized power

Polar Schemes

In polar schemes, the voltages are on the both sides of the time axis.

For example, the voltage level for 0 can be positive and the voltage level for I can be negative.

Non-Return-to-Zero (NRZ)

In polar NRZ encoding, we use two levels of voltage amplitude.

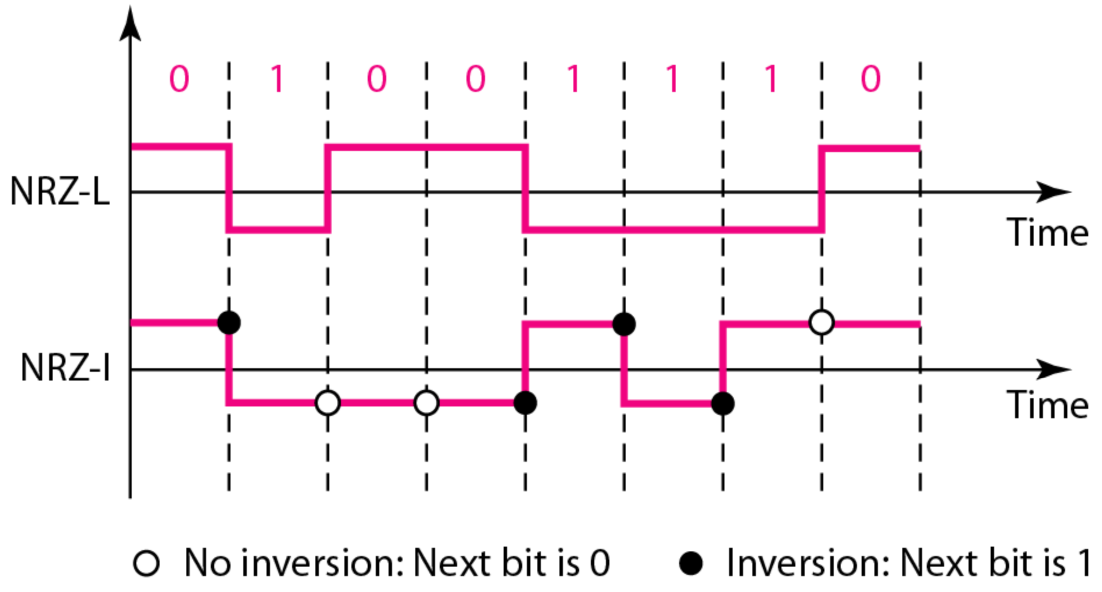
We can have two versions of polar NRZ: NRZ-Land NRZ-I

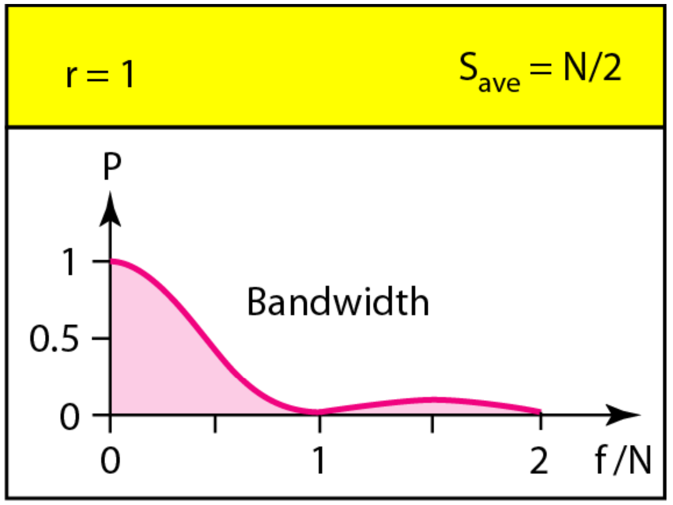
•In the first variation, NRZ-L (NRZ-Level), the level of the voltage determines the value of the bit.

•In the second variation, NRZ-I (NRZ-Invert), the change or lack of change in the level of the voltage determines the value of the bit.

•If there is no change, the bit is 0; if there is a change, the bit is 1.

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*





Note

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.

Note

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

RZ

It uses three values

Positive

Negative &

Zero

- In RZ the signal changes during each bit.
- A 1 bit is actually represented by positive-to-zero and
- A 0 bit is actually represented by negative-to-zero

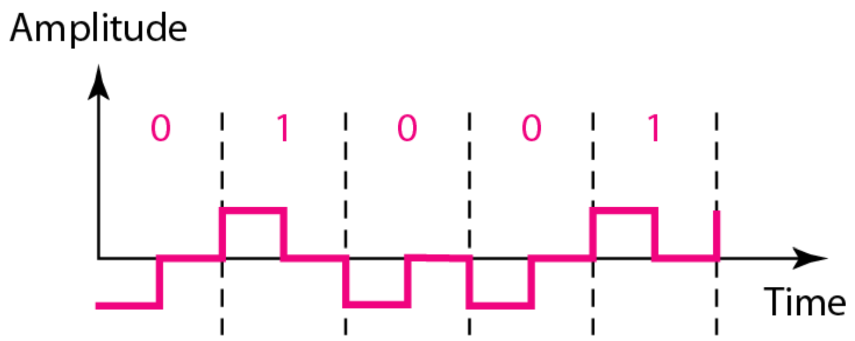
Demerits

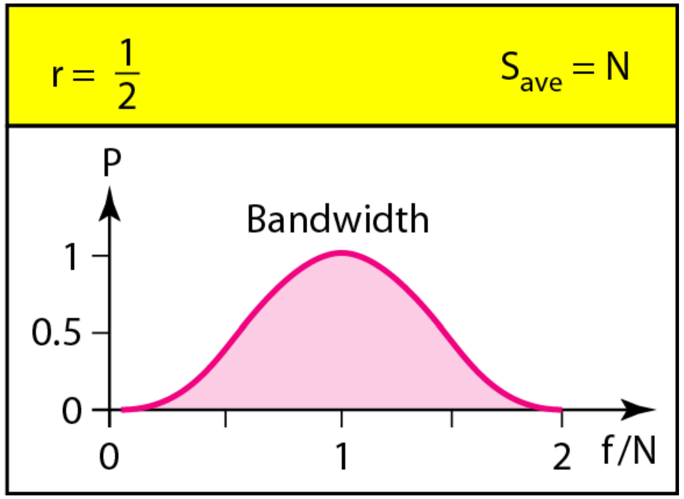
- It requires two signal changes to encode one bit.
- It occupies more bandwidth.

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.

Figure 4.7 *Polar RZ scheme*

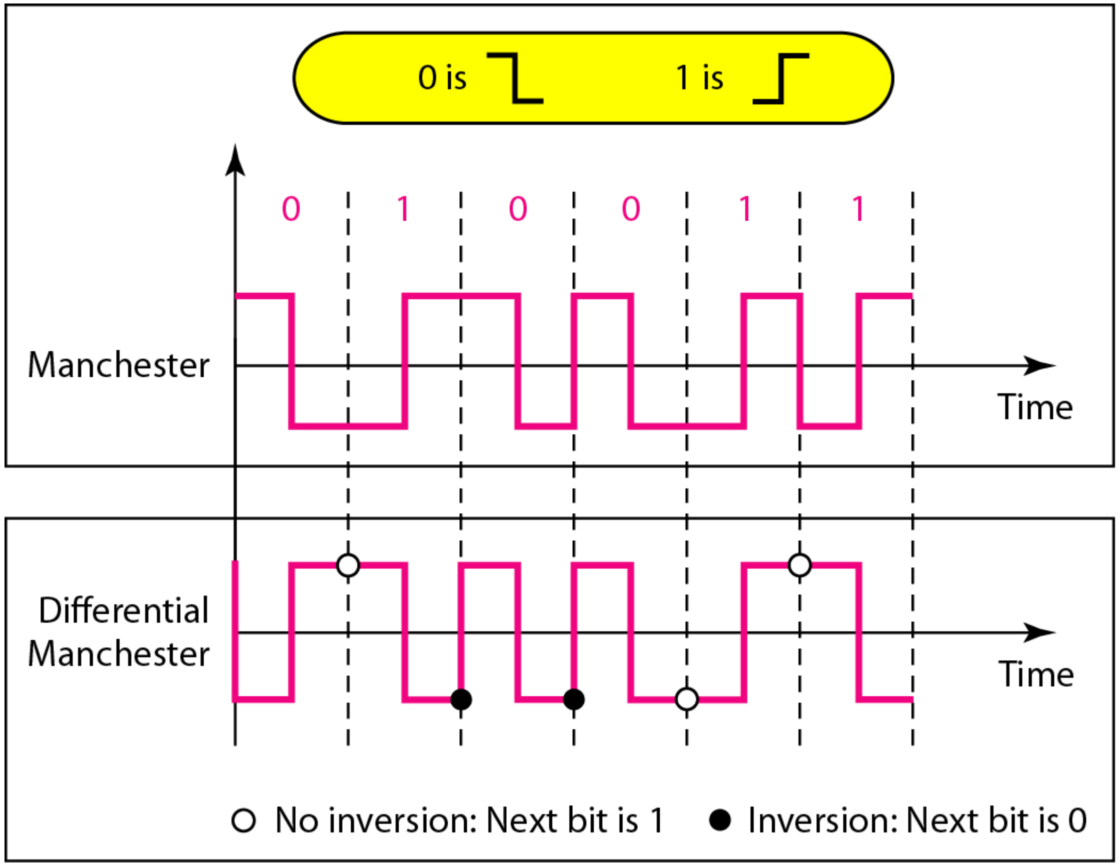


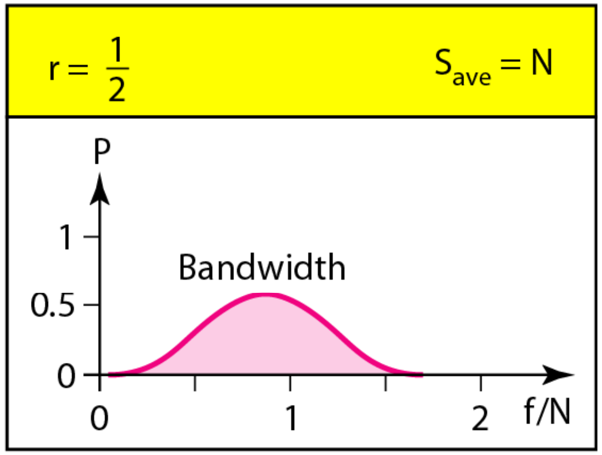


Polar - Biphase: Manchester and Differential Manchester

- 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.

Figure 4.8 *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.

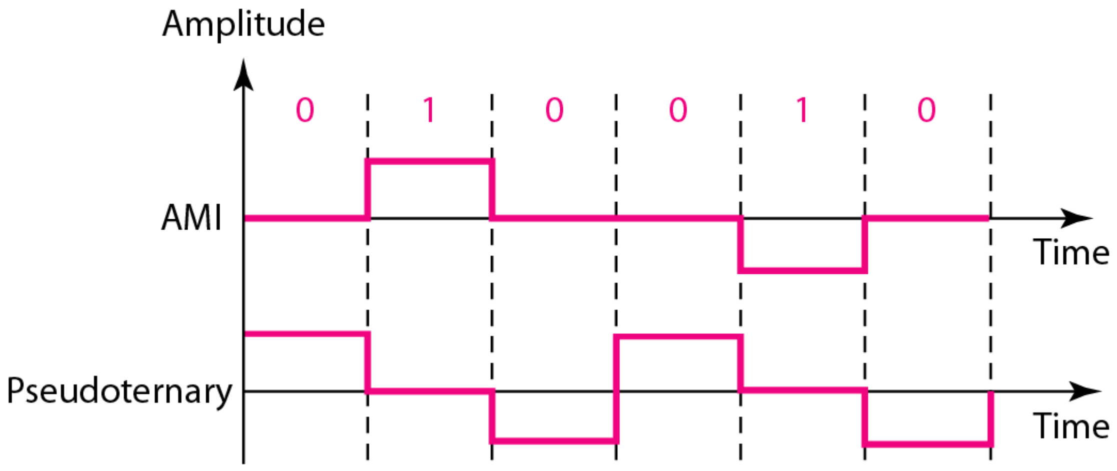
Note

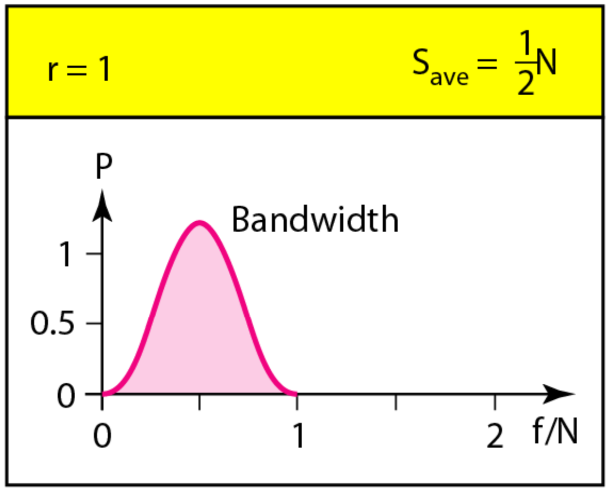
The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ. The is no DC component and no baseline wandering. None of these codes has error detection.

Bipolar - AMI and Pseudoternary

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

Figure 4.9 *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.

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 “2^m” symbols.
- If we have L signal levels, we can use “n” signal elements to create L^n signal elements.

Code C/Cs

- Now we have 2^m symbols and L^n signals.
- If $2^m > L^n$ then we cannot represent the data elements, we don’t have enough signals.
- If $2^m = L^n$ then we have an exact mapping of one symbol on one signal.
- If $2^m < L^n$ 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.

Note

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

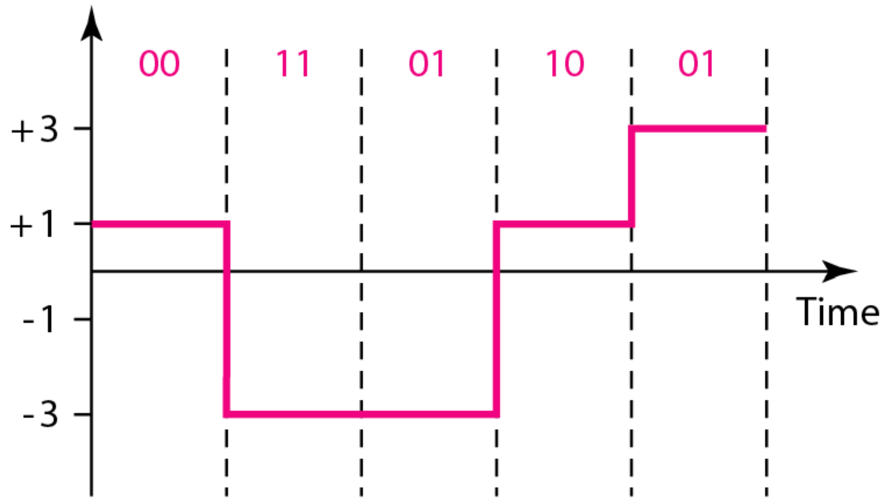
Representing Multilevel Codes

- 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.

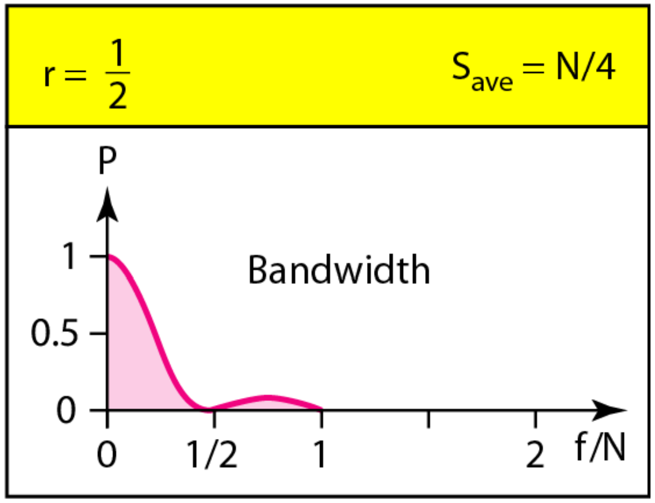
Figure 4.10 *Multilevel: 2B1Q scheme*

	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



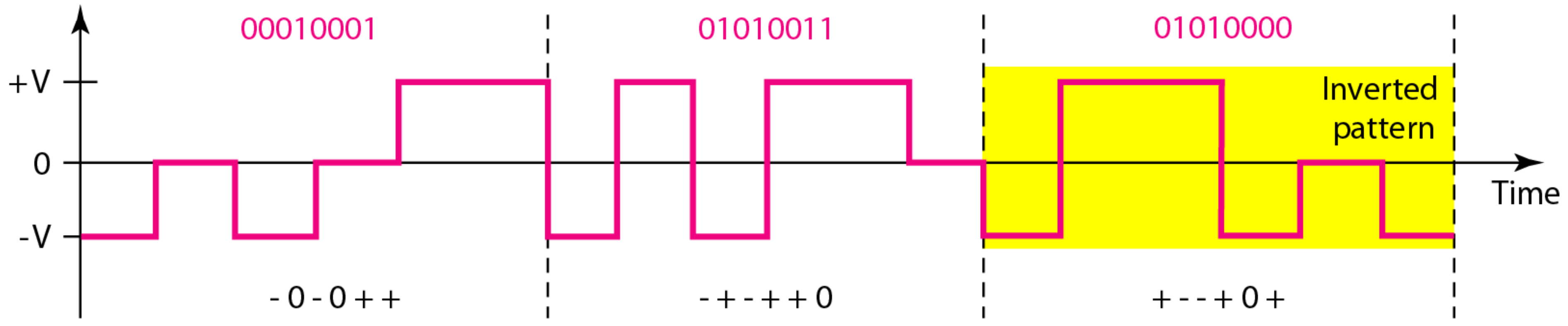
Assuming positive original level



Redundancy

- In the 2B1Q scheme we have no redundancy and we see that a DC component is present.
- If we use a code with redundancy we can decide to use only “0” or “+” weighted codes (more +’s than -’s in the signal element) and invert any code that would create a DC component. E.g. ‘+00++-’ -> ‘-00--+’
- Receiver will know when it receives a “-” weighted code that it should invert it as it doesn’t represent any valid symbol.

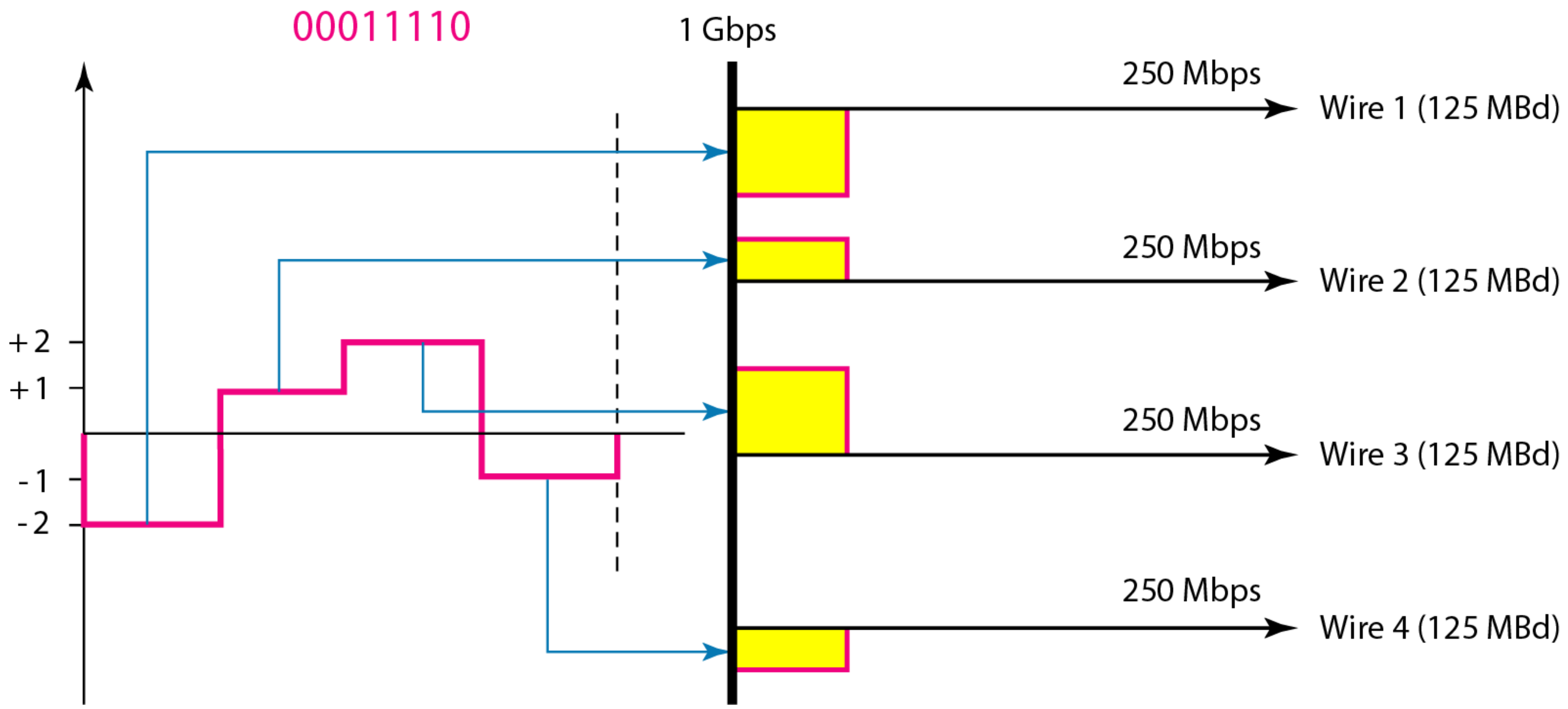
Figure 4.11 *Multilevel: 8B6T scheme*



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 signalling 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

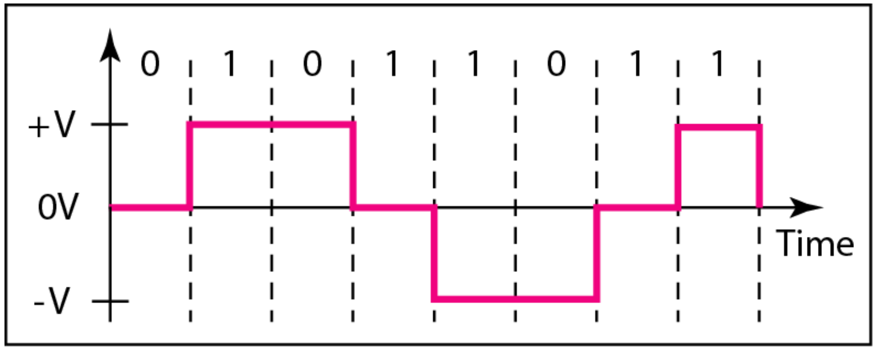
Figure 4.12 *Multilevel: 4D-PAM5 scheme*



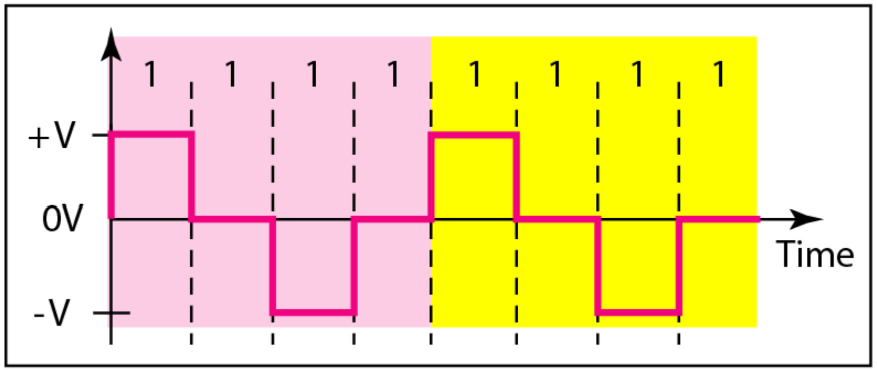
Multitransition Coding

- 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.
- In some instances, the bandwidth requirement may even be lower, due to repetitive patterns resulting in a periodic signal.

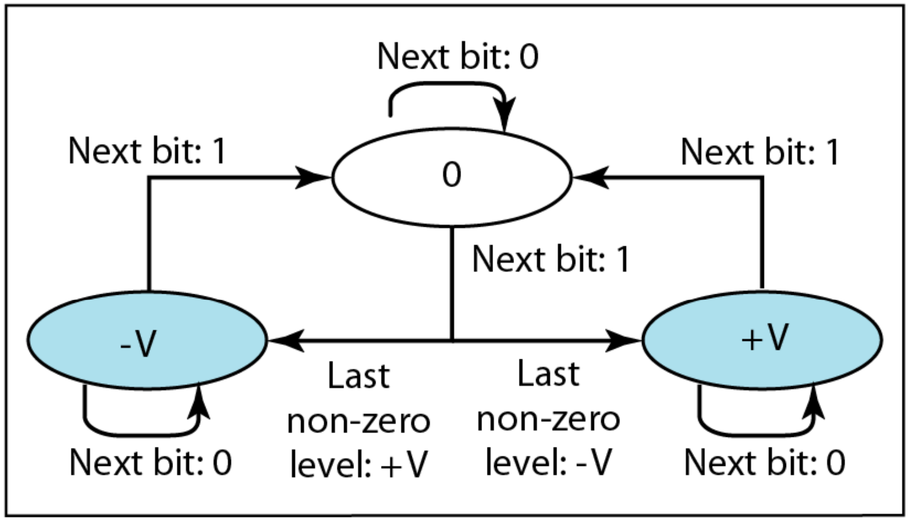
Figure 4.13 *Multitransition: MLT-3 scheme*



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
- This can be approximated as an analog signal a frequency 1/4 the bit rate!

Table 4.1 ***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

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

Figure 4.14 *Block coding concept*

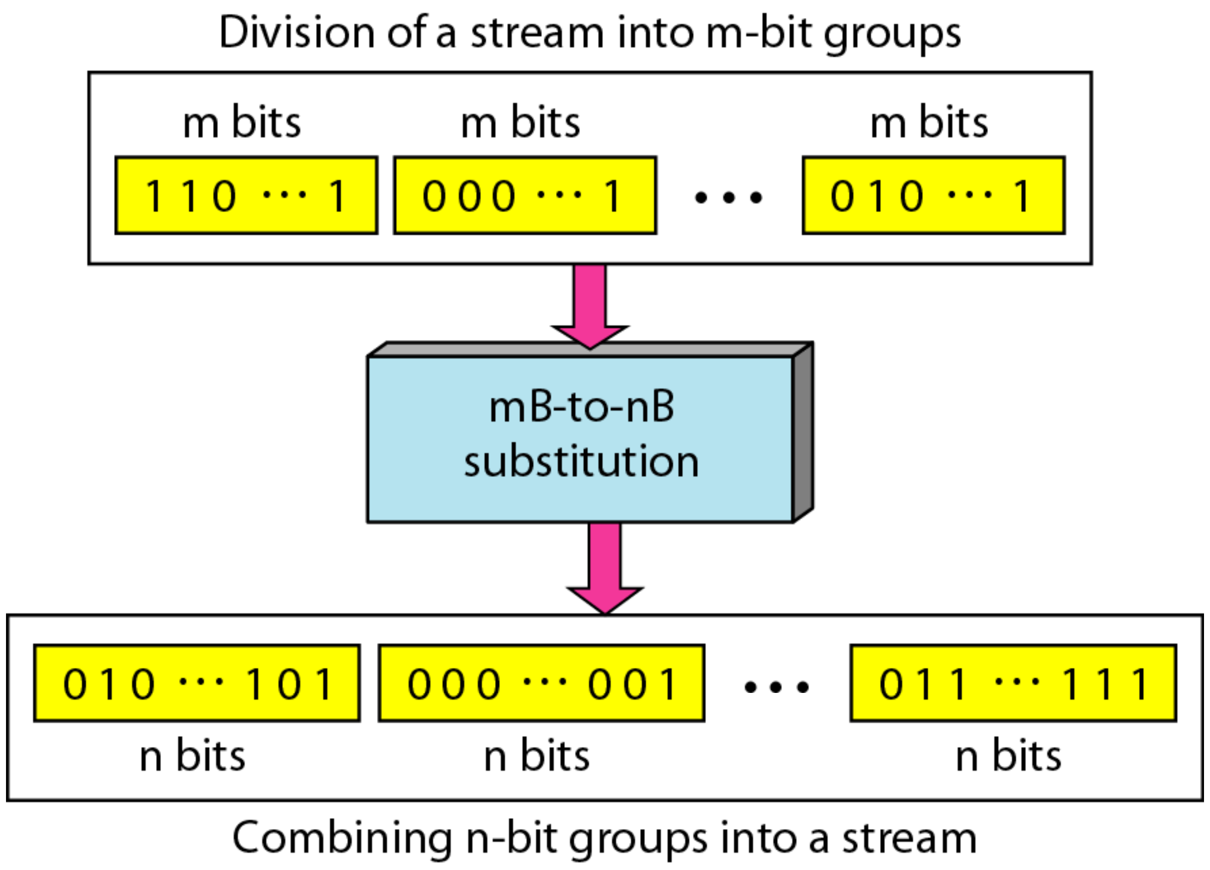


Figure 4.15 *Using block coding 4B/5B with NRZ-I line coding scheme*

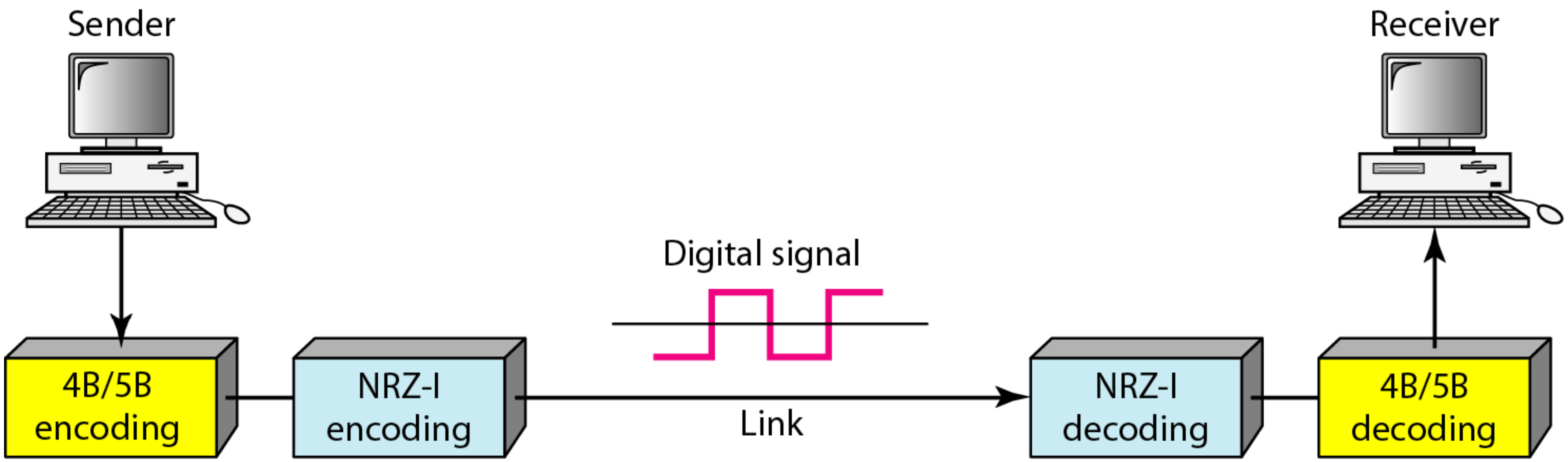
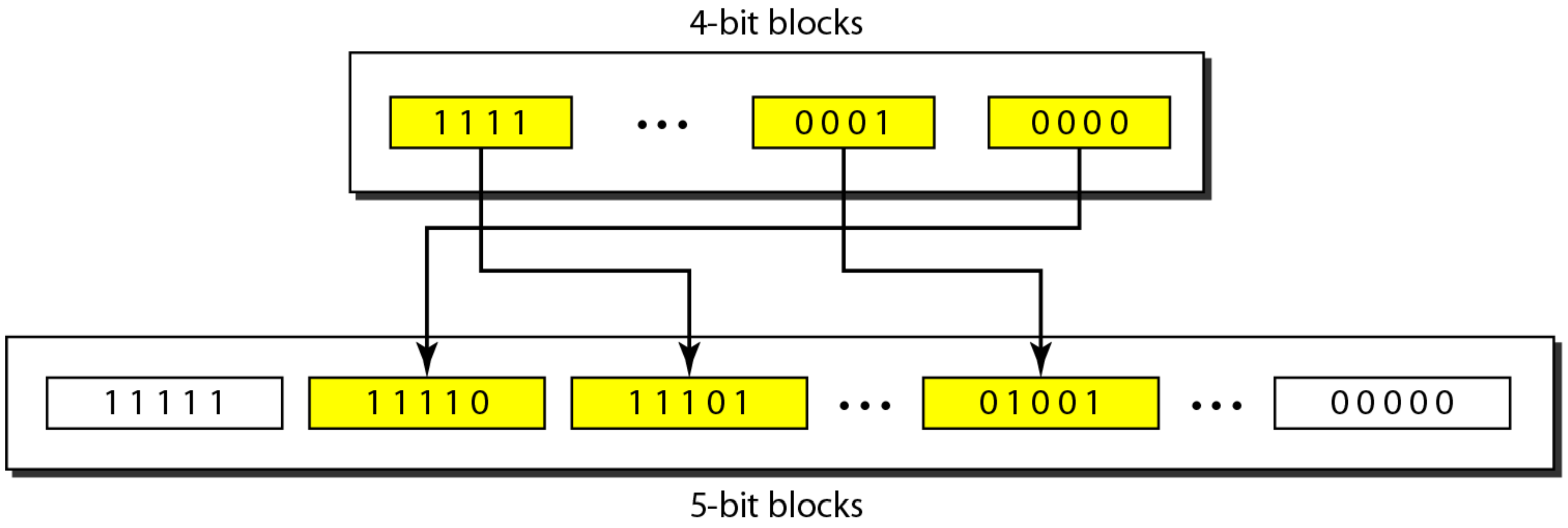


Table 4.2 *4B/5B mapping codes*

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		

Figure 4.16 *Substitution in 4B/5B block coding*



Redundancy

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

Example 4.5

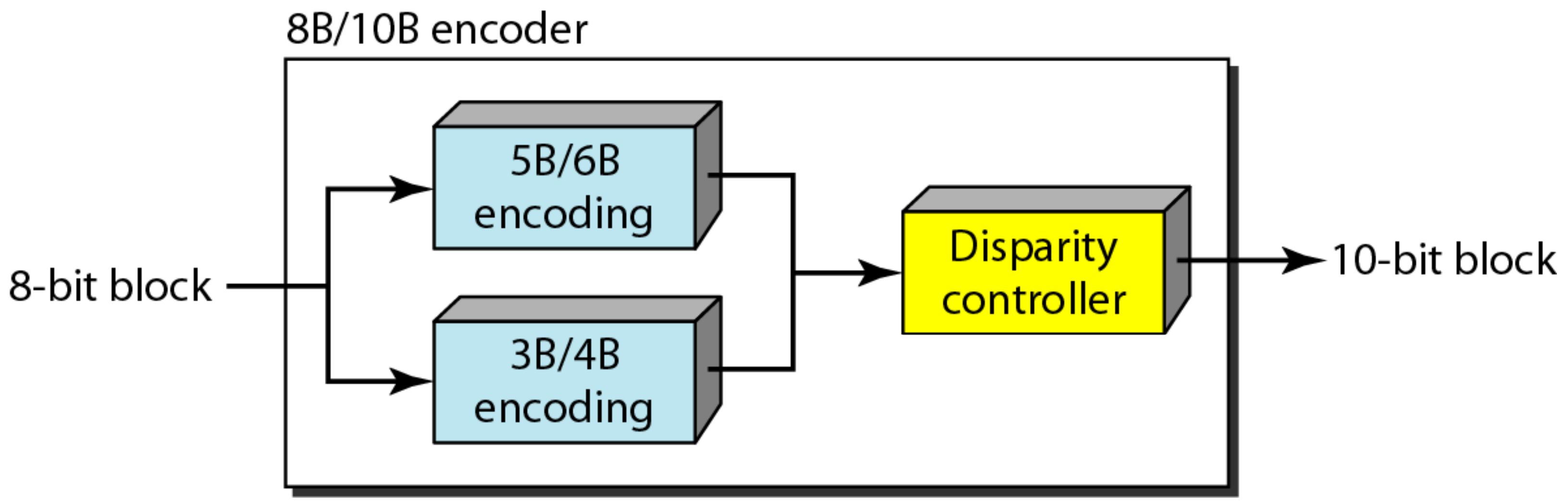
We need to send data at a 1-Mbps rate. What is the minimum required bandwidth, using a combination of 4B/5B and NRZ-I or Manchester coding?

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

Figure 4.17 *8B/10B block encoding*



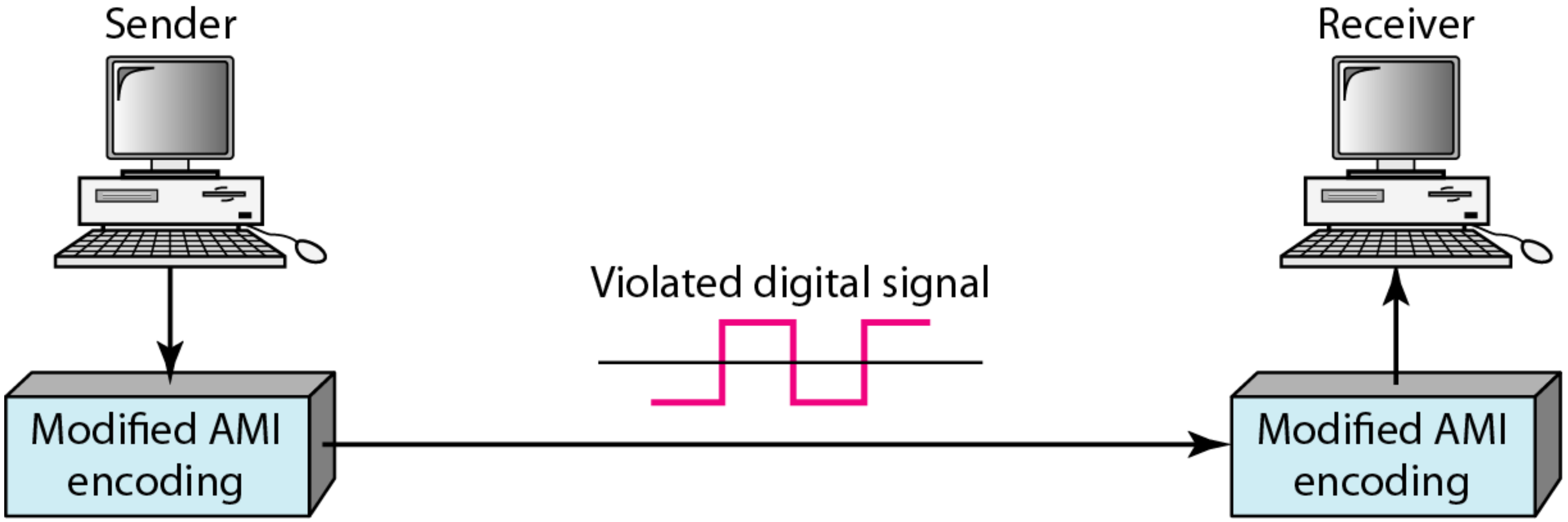
More bits - better error detection

- 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.

Scrambling

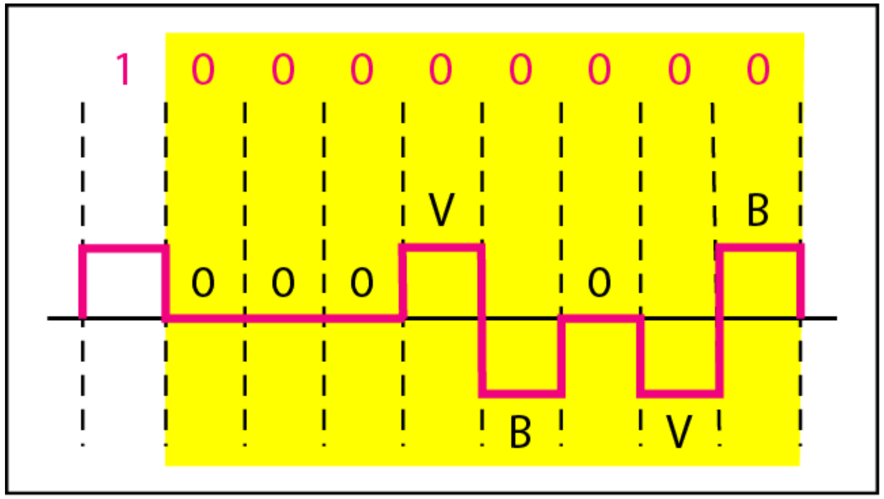
- The best code is one that does not increase the bandwidth for synchronization and has no DC components.
- Scrambling is a technique used to create a sequence of bits that has the required c/c’s for transmission - self clocking, no low frequencies, no wide bandwidth.
- It is implemented at the same time as encoding, the bit stream is created on the fly.
- It replaces ‘unfriendly’ runs of bits with a violation code that is easy to recognize and removes the unfriendly c/c.

Figure 4.18 *AMI used with scrambling*

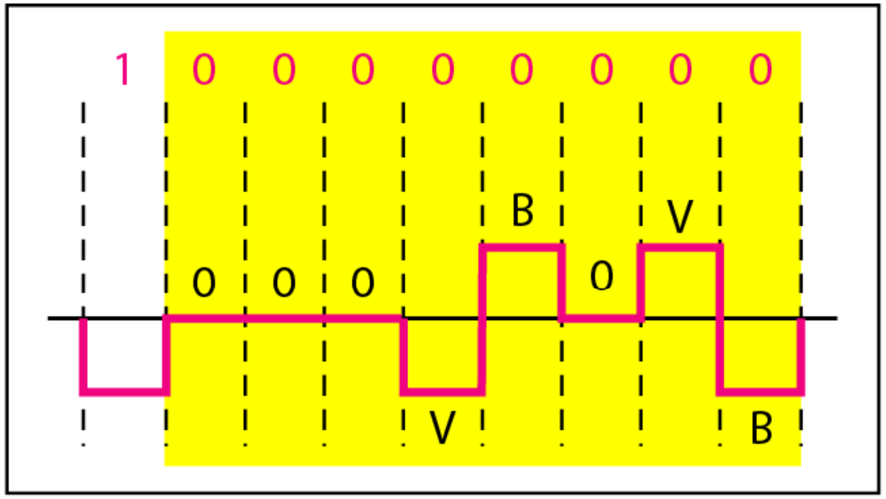


For example: B8ZS substitutes eight consecutive zeros with 000VB0VB.
The V stands for violation, it violates the line encoding rule
B stands for bipolar, it implements the bipolar line encoding rule

Figure 4.19 *Two cases of B8ZS scrambling technique*



a. Previous level is positive.



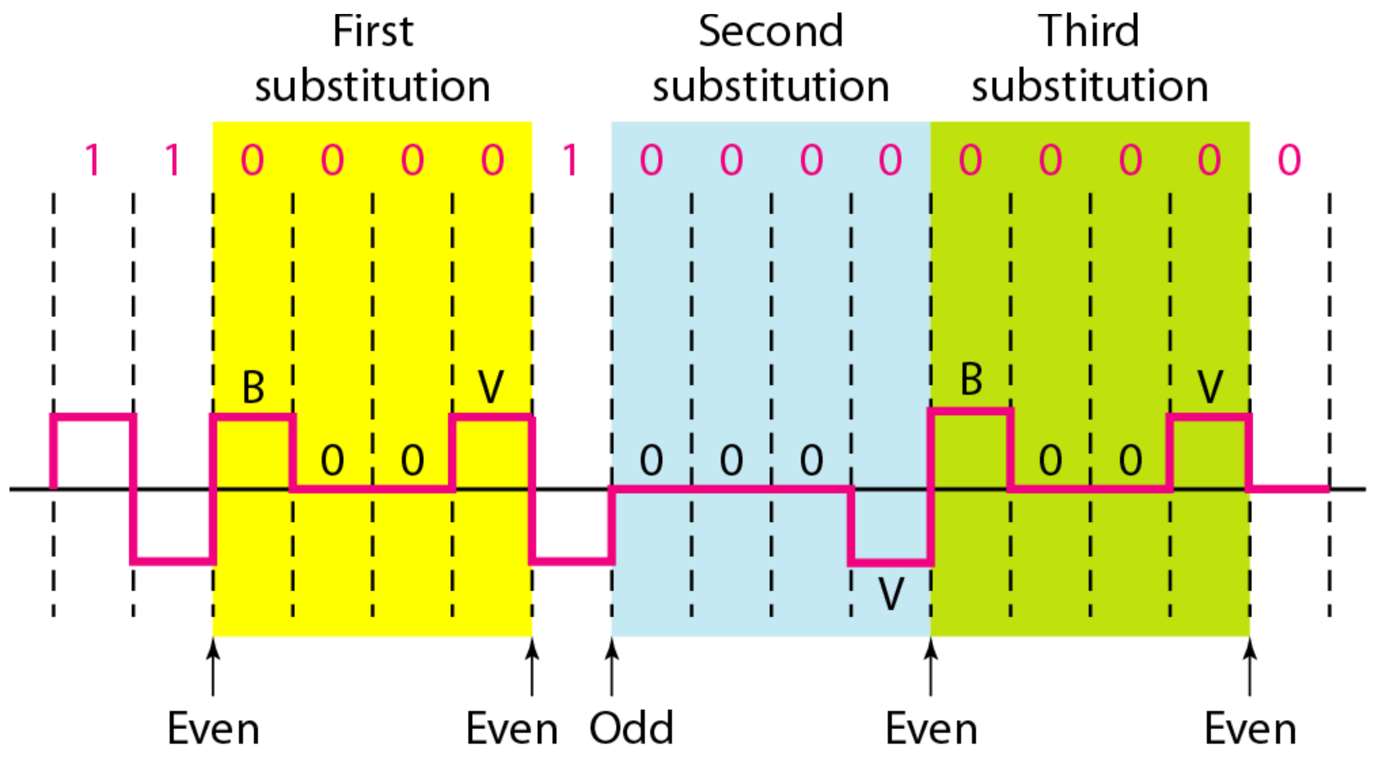
b. Previous level is negative.

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.

Figure 4.20 *Different situations in HDB3 scrambling technique*


First substitution
Second substitution
Third substitution
1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
B
0
0
V
0
0
0
V
B
0
0
V
Even
Even
Odd
Even
Even