Line, Block Coding and Scrambling

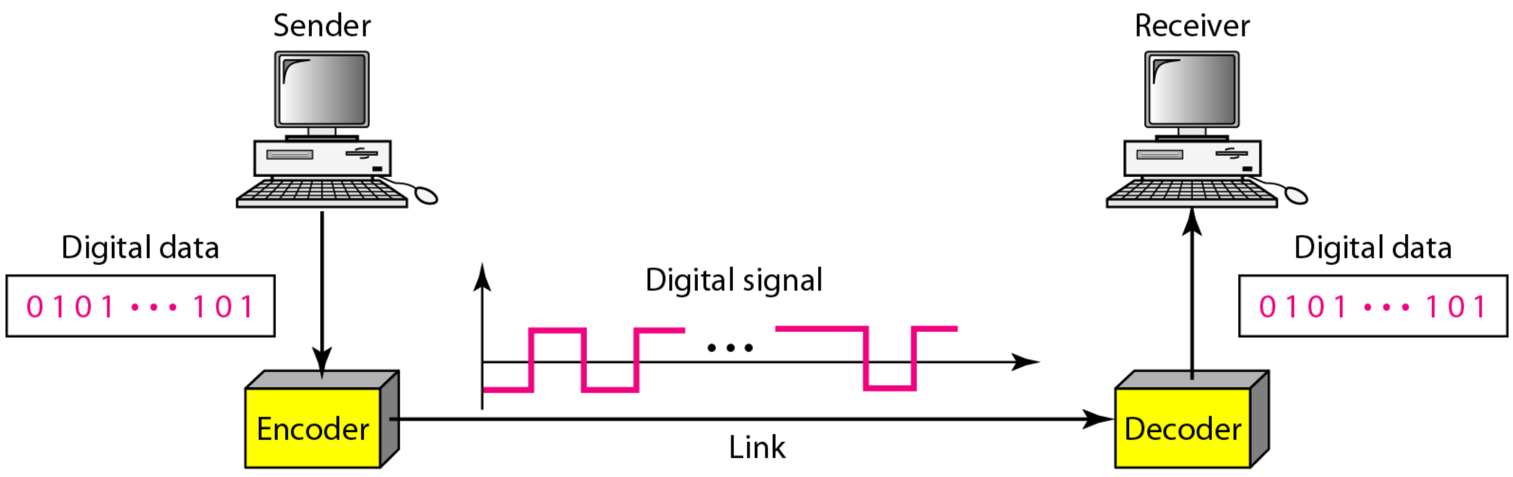
Dr. Md. Fazlul Kader
Professor
Dept. of EEE, University of Chittagong

Digital-to-Digital Conversion

- Representing 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

- Characteristics:
  - Signal Element vs Data Element
  - Bit Rate vs Baud Rate
  - Baseline Wandering
  - DC Components
  - Self Synchronization
  - Bandwidth
  - Built-in Error Detection
  - Immunity to Noise and Interference
  - Complexity

Cont.

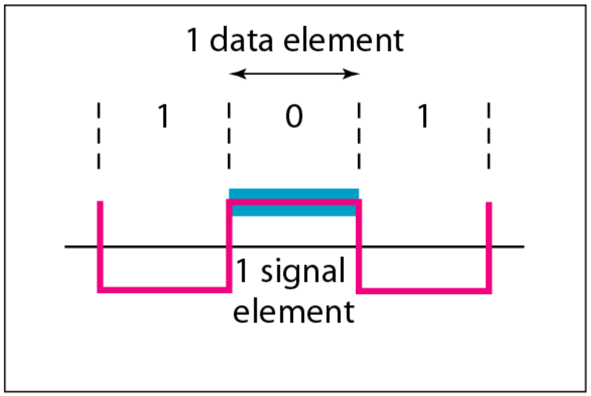
- **Signal Element vs Data Element**
  - **Data Element**
    - A data element is the smallest entity that can represent a piece of information: this is the bit.
  - **Signal Elements**
    - A signal element carries data elements. A signal element is the shortest unit (timewise) of a digital signal.
  - **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.**

Cont.

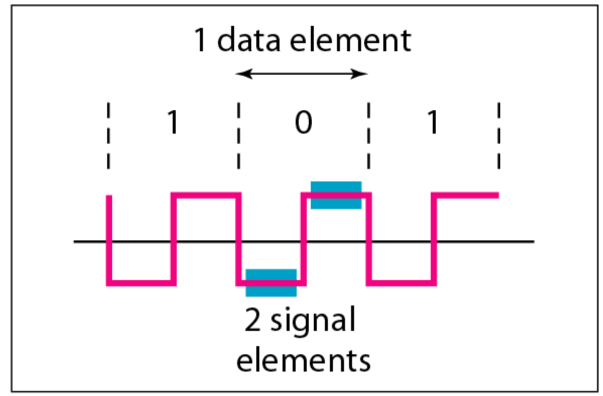
- **Mapping Data symbols onto Signal levels**
  - A data symbol (or element) can consist of a number of data bits:
    - 1 , 0 or
    - 11, 10, 01, ……
  - A data symbol can be coded into a single signal element or multiple signal elements
    - 1 -> +V, 0 -> -V
    - 1 -> +V and -V, 0 -> -V and +V
  - **The ratio ‘r’** is the number of data elements carried by a signal element.

Cont.

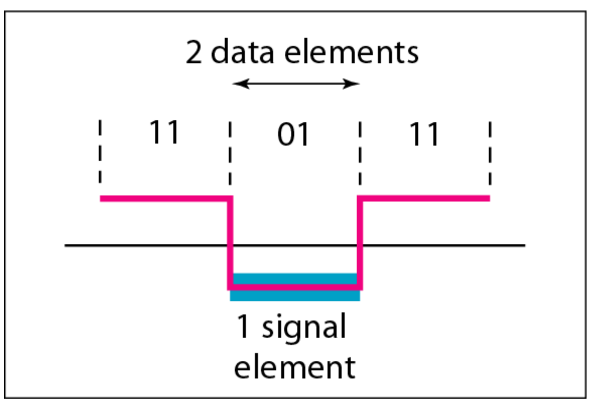
- **Signal Element vs Data Element**



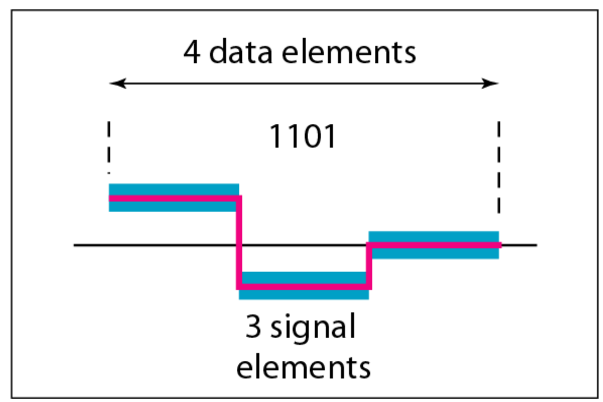
a. One data element per one signal element (r = 1)



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

Cont.

- **Bit Rate vs Baud/Signal Rate/Pulse Rate**
  - The data rate defines the number of bits sent per sec - **bps.**
    - It is often referred to the bit rate.
  - The signal rate is the number of signal elements sent in a second and is measured in **bauds**.
    - It is also referred to as the modulation rate.
  - **Pulse Rate:** Number of pulses per second
    - A pulse is the minimum amount of time required to transmit a symbol
    - Bit rate =Pulse rate * Log_2 (L),
      - L→number of data levels of the signals
  - **Goal:** To increase the data rate whilst reducing the baud rate.

Cont.

- **Example-1:**
  - A signal has two data levels with a pulse duration of 1 ms. We calculate the pulse rate and bit rate as follows:
  - Pulse Rate = $1/\ 10^{-3}$= 1000 pulses/s
  - Bit Rate = Pulse Rate x $\log_2 L$ = 1000 x $\log_2 2$ = 1000 bps
- **Example-2:**
- A signal has four data levels with a pulse duration of 1 ms. We calculate the pulse rate and bit rate as follows:
  - Pulse Rate = = 1000 pulses/s
  - Bit Rate = Pulse Rate x $\log_2 L$ = 1000 x $\log_2 4$ = 2000 bps

Cont.

- **Data Rate and Baud Rate**
  - The baud or signal rate can be expressed as:

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

    - where N is data rate
    - c is the case factor (worst, best & avg.)
    - r is the ratio between data element & signal element
    - **Best case:** minimum signal rate
    - **Worst case:** maximum signal rate

Cont.

- **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?
  - Soln: ***We assume that the average value of c is 1/2 . The baud rate is then***

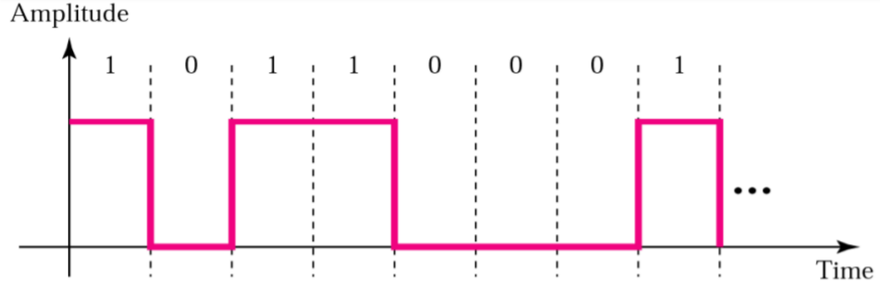
$$\boxed{S = c \times N \times \frac{1}{r} = \frac{1}{2} \times \mathbf{100{,}000} \times \frac{1}{1} = \mathbf{50{,}000} = \mathbf{50\ kbaud}}$$

Cont.

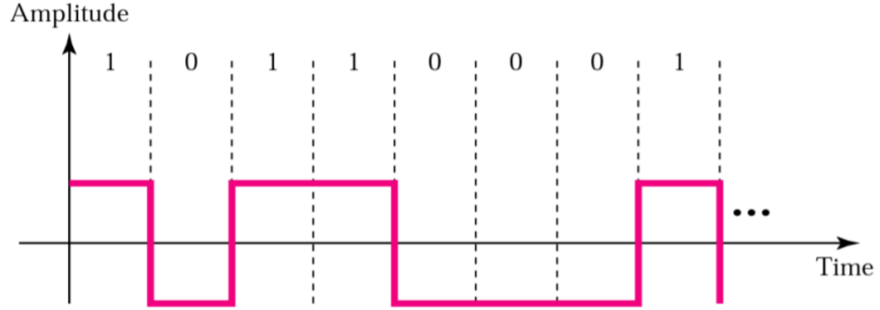
- **Baseline Wandering**
  - A receiver will evaluate the average power of the received signal (**called the baseline**) and use that to determine the value of the incoming data elements.
  - If the incoming signal does not vary over a long period of time, the baseline will drift and thus cause errors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

Cont.

- **DC Components**
  - Some line coding schemes have a residual (DC) component, which is generally undesirable
    - Transformers do not allow passage of DC component
    - DC component ⇒ extra energy – useless!



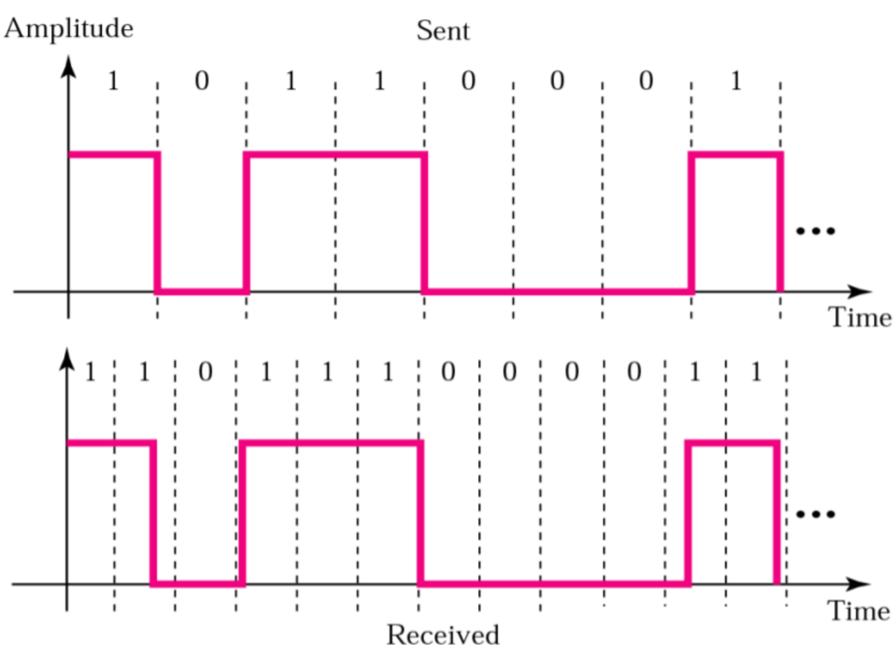
a. A signal with dc component



b. A signal without dc component

Cont.

- **Self-synchronization (Clocking)**
  - To correctly interpret signal received from sender, receiver's bit interval must exactly correspond to sender's bit intervals
    - If receiver clock is faster/slower, bit intervals not matched ⇒ receiver misinterprets signal
    - **Self-synchronizing digital signals** include timing information in itself, to indicate the beginning & end of each pulse



Cont.

- **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?

At 1 Kbps:

1000 bits sent ➔ 1001 bits received ➔ 1 extra bps

At 1 Mbps:

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

Cont.

- **Bandwidth**
  - Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.
  - **The baud rate**, not the bit rate, determines the **required bandwidth** for a digital signal.
  - The minimum bandwidth can be given as

$$B_{min} = c \times N \times \frac{1}{r}$$

  - The maximum data rate if the bandwidth of the channel is given as

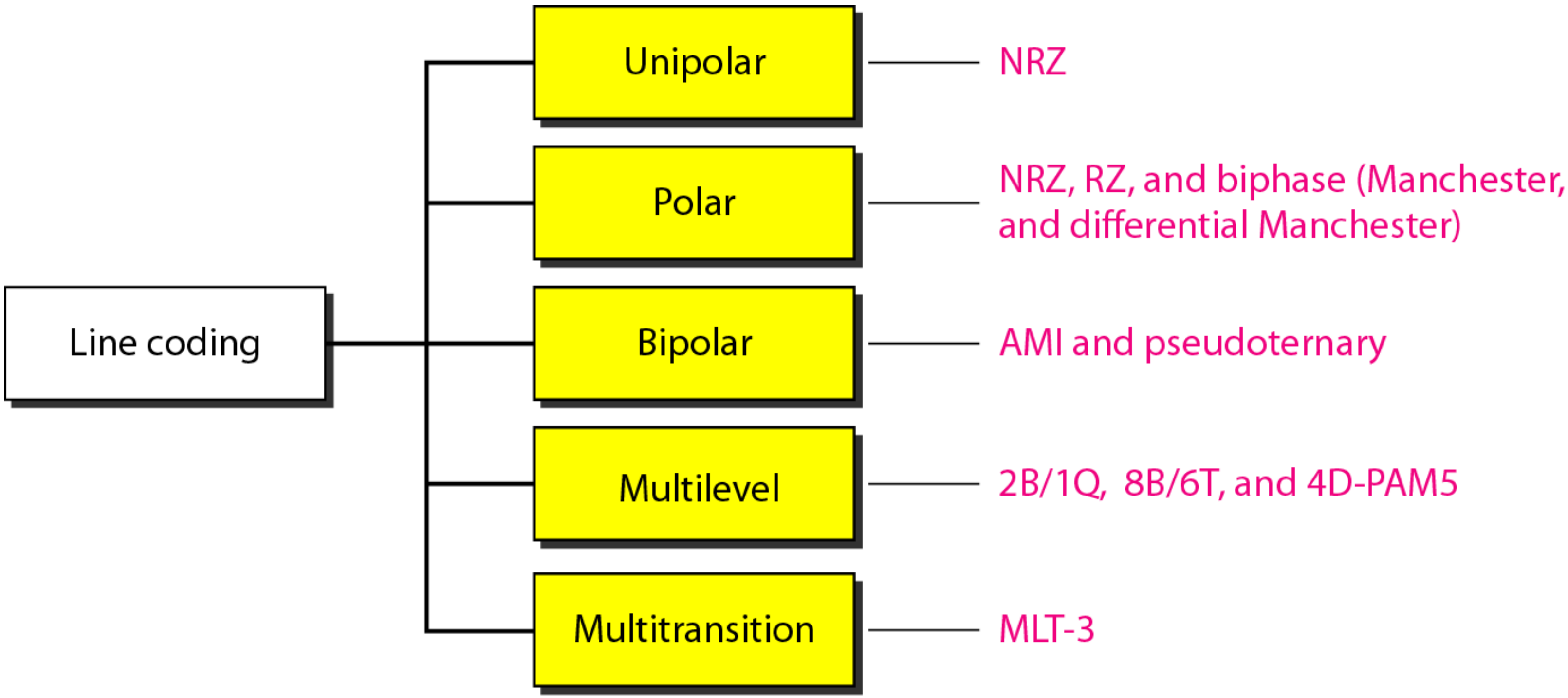
$$N_{max} = \frac{1}{c} \times B \times r$$

Cont.

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

Line Coding Schemes

- **Classification:**
  - Roughly divided into five broad categories



Cont.

- **For Unipolar and Polar:**

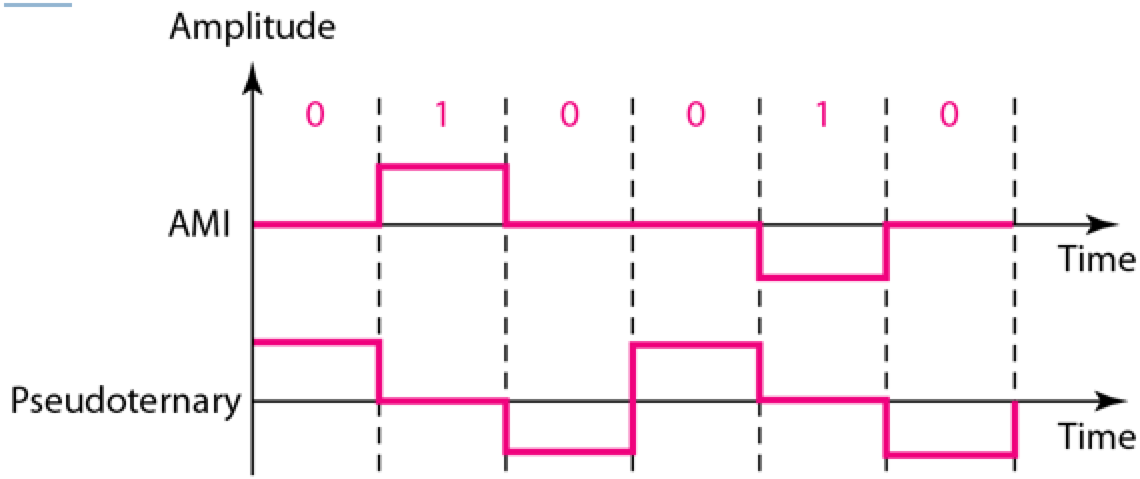
  Check: 1.Lec_DigitalTrans_Part2_2

Cont.

- **Bipolar**
  - Classification:
    - Alternate Mark Inversion (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.

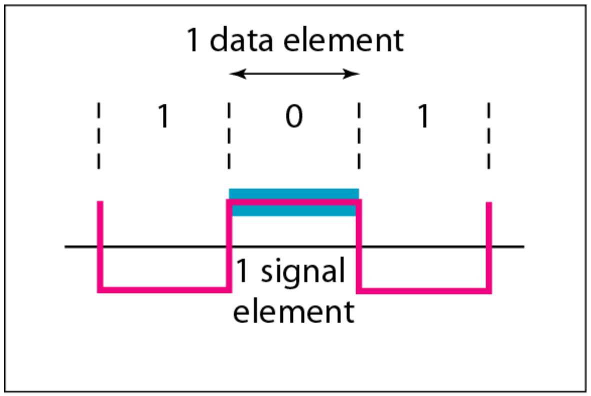
Cont.

- Bipolar schemes

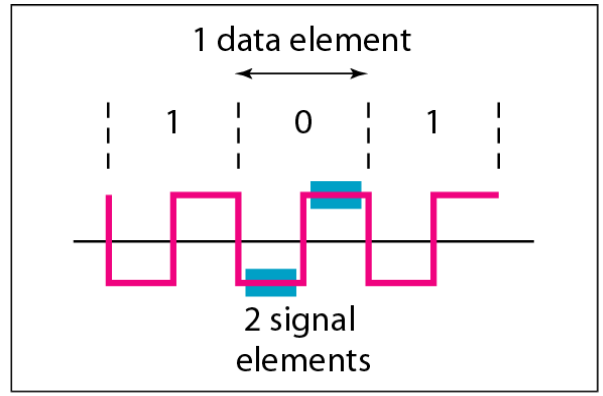


Cont.

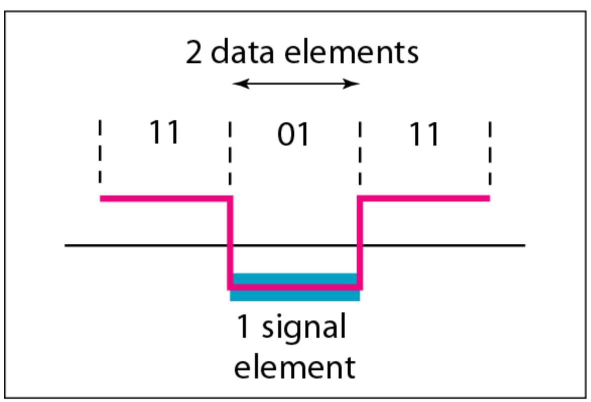
- **Signal Element vs Data Element**



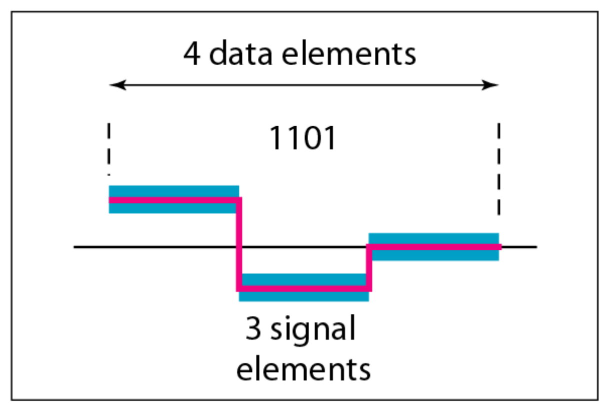
a. One data element per one signal element (r = 1)



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

Multilevel Schemes

- In these schemes
  - Increase the number of data bits per symbol
  - Increasing the bit rate.
- Binary data→2 types of data element: 1 or 0.
- ***m*** data elements create "2^m" symbols/data pattern.
- Different signal levels→ different types of signal elements
- ***L*** signal levels→L^n combinations of signal patterns
  - ***n*** → length of a signal pattern
- If $2^m = L^n$ →an exact mapping of one symbol to one signal pattern
- If $2^m > L^n$ →data encoding is not possible
  - Not enough signal patterns
- If $2^m < L^n$ →data patterns occupy only a subset of signal patterns
  - The subset can be carefully designed
    - to prevent baseline wandering,
    - to provide synchronization, and
    - to detect errors that occurred during data transmission.

Cont.

- Classified as ***mBnL***,
  - where *m* is the length of the binary pattern,
  - *B* means binary data,
  - *n* is the length of the signal pattern,
  - *L* is the number of levels in the signaling.
  - A letter is often used in place of L:
    - *B* (binary) for *L* =2, *T* (ternary) for *L* =3, and Q (quaternary) for *L* =4.
- Note that the **first two letters define the data pattern**, and **the second two define the signal pattern**.

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

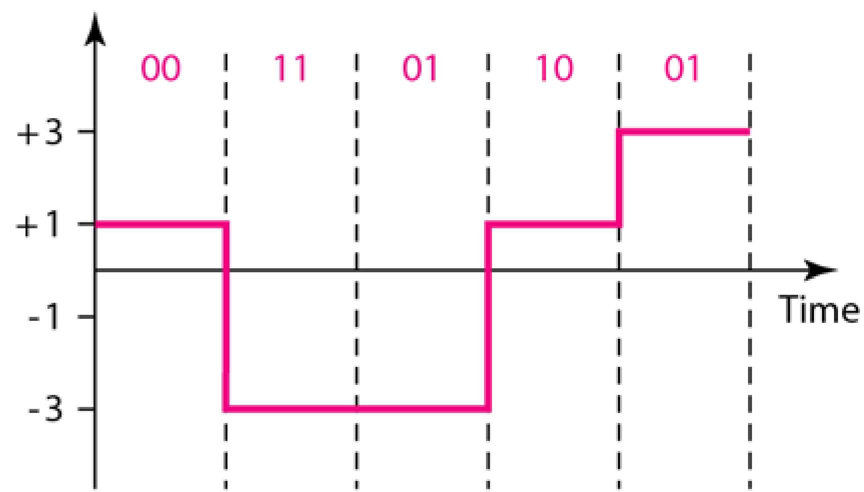
Cont.

Multilevel: 2B1Q scheme

- Two binary, one quaternary (2B1Q),
- Uses data patterns of size 2 and
- Encodes the 2-bit patterns as one signal element
- A four-level signal.
- In this type of encoding $m = 2$, $n = 1$, and $L = 4$ (quaternary).
- No redundant signal patterns
- N=data rate
- Average signal rate S_{avg}=N/4
- Send data 2 times faster than by using NRZ-L.
- Used in DSL technology to provide a high-speed connection to the Internet
- Receiver complexity
- Reduced bandwidth

	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

Cont.

- **In the 2B1Q scheme**
  - no redundancy and
  - DC component is present.
- **Redundancy**
- A code with redundancy
  - Decide to use only “0” or “+” weighted codes (more +’s than -’s in the signal element) and
    - Used to provide DC balance.
  - Each **signal pattern** has a weight of 0 or +1 DC values.
  - No pattern with the weight -1.
  - To make the whole stream DC-balanced, the sender keeps track of the weight.
    - If two groups of weight 1 are encountered one after another
      - the first one is sent as is,
      - while the next one is totally inverted to give a weight of -1
  - Invert any code that would create a DC component. E.g. ‘+00++-’ >> ‘-00--+’

Cont.

- Receiver will know when it receives a “-” weighted code that it should invert it as it doesn’t represent any valid symbol.
- **How to measure weight:**
  - **For example:**
    - **00010001** is encoded as the signal pattern **-0-0++** with **weight 0**; (+ ve = - ve) signals)
    - The second 8-bit pattern **010 10011** is encoded as **- + - + + 0** with weight +1. (+ ve > - ve) signals)
    - The third bit pattern **01010000** should be encoded as **+ - - + 0 +** with **weight +1**.
      - **Should be inverted**

Cont.

- **Multilevel: *8B6T* scheme**

➢ 8 binary, 6 ternary (8B6T),

- Used with 100BASE-4T cable
- **Idea:**
  - To encode a pattern of 8 bits as a pattern of 6 signal elements,
  - The signal has three levels (ternary).
- 2^8 =256 different data patterns and 3^6 =478 different signal patterns
- 478 - 256 =222 redundant signal elements that provide synchronization and error detection.
- S_{avg}=1/2*N*6/8=3N/8

What is 100Base-T4?

- 100BASE-T4 is the early implementation of Fast Ethernet over twisted pair cables,
- Carrying data traffic at 100 Mbps (Mega bits per second) in local area networks (LAN).
- It was launched as the IEEE 802.3u standard in 1995.
- Here, 100 is the maximum throughput, i.e. 100 Mbps, BASE denoted use of baseband transmission, and
- T4 denotes use of four twisted pair cables in Fast Ethernet.

Cont. (Appendix D)

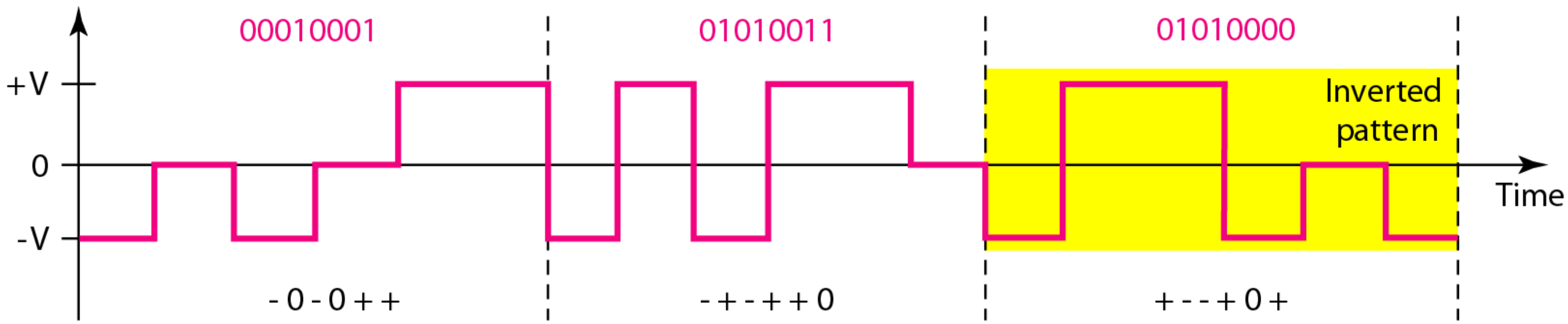
- **8B/6TCode Mapping Table:**

Table D.1 *8B/6T code*

Data	*Code*	*Data*	*Code*	*Data*	*Code*	*Data*	*Code*
00	-+00-+	20	-++-00	40	-00+0+	60	0++0-0
01	0-+-+0	21	+00+--	41	0-00++	61	+0+-00
02	0-+0-+	22	-+0-++	42	0-0+0+	62	+0+0-0
03	0-++0-	23	+-0-++	43	0-0++0	63	+0+00-
04	-+0+0-	24	+-0+00	44	-00++0	64	0++00-
05	+0--+0	25	-+0+00	45	00-0++	65	++0-00
06	+0-0-+	26	+00-00	46	00-+0+	66	++00-0
07	+0-+0-	27	-+++--	47	00-++0	67	++000-
08	-+00+-	28	0++-0-	48	00+000	68	0++-+-
09	0-++-0	29	+0+0--	49	++-000	69	+0++--
0A	0-+0+-	2A	+0+-0-	4A	+-+000	6A	+0+-+-
0B	0-+-0+	2B	+0+--0	4B	-++000	6B	+0+--+
0C	-+0-0+	2C	0++--0	4C	0+-000	6C	0++--+
0D	+0-+-0	2D	++00--	4D	+0-000	6D	++0+--
0E	+0-0+-	2E	++0-0-	4E	0-+000	6E	++0-+-
0F	+0--0+	2F	++0--0	4F	-0+000	6F	++0--+

- **The 8-bit data are shown in hexadecimal format.**
- The 6T code is shown as **+** (positive signal), **-** (negative signal), and 0 (lack of signal) notation.

Cont.

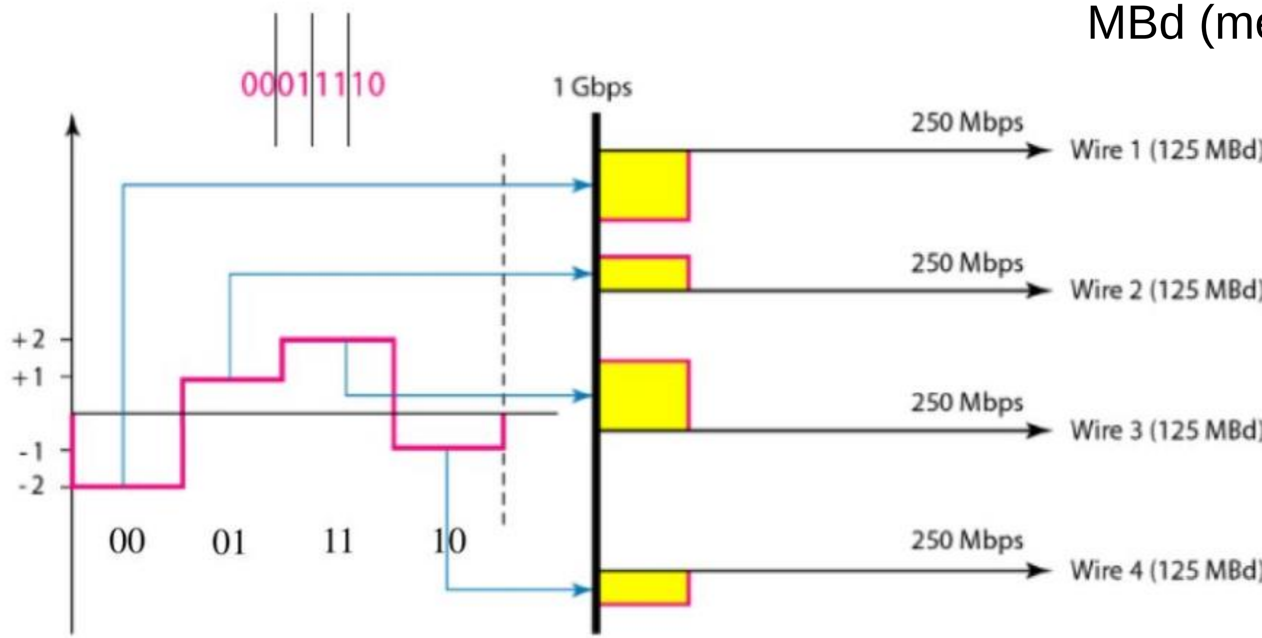


- \+ (positive signal),
- \- (negative signal), and
- \- 0 (lack of signal)

Cont.

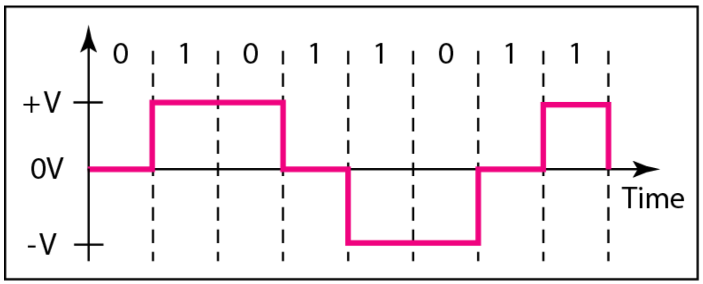
- **Multilevel: 4D-PAM5 scheme**
  - Four dimensional five-level pulse amplitude modulation (4D-PAM5).
  - The 4D means that data is sent over four wires at the same time.
  - It uses five voltage levels, such as -2, -1, 0, 1, and 2.
  - One level, level 0, is used only for forward error detection

MBd (megabaud)

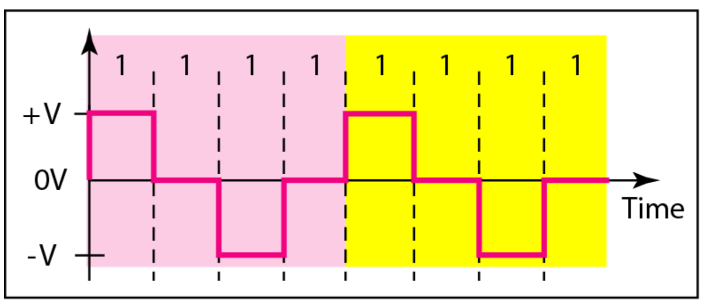


Cont.

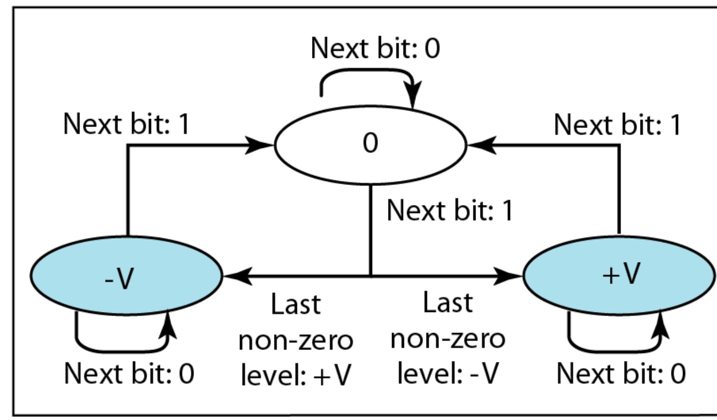
- **Multiline transition: MLT-3 scheme**
  - Uses three levels (+*v*, 0, and - *V*) and three transition rules to move between the levels.



a. Typical case



b. Worse case



c. Transition states

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 cutTent level is 0, the next level is the opposite of the last nonzero level.

Summary of line coding schemes

Category	*Scheme*	*Bandwidth (average)*	*Characteristics*
Unipolar		$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Polar	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

Redundancy for Error Detection

VERTICAL REDUNDANCY CHECK

- Example :

  1110110 1101111 1110010

  - After adding the parity bit

  11101101 11011110 11100100

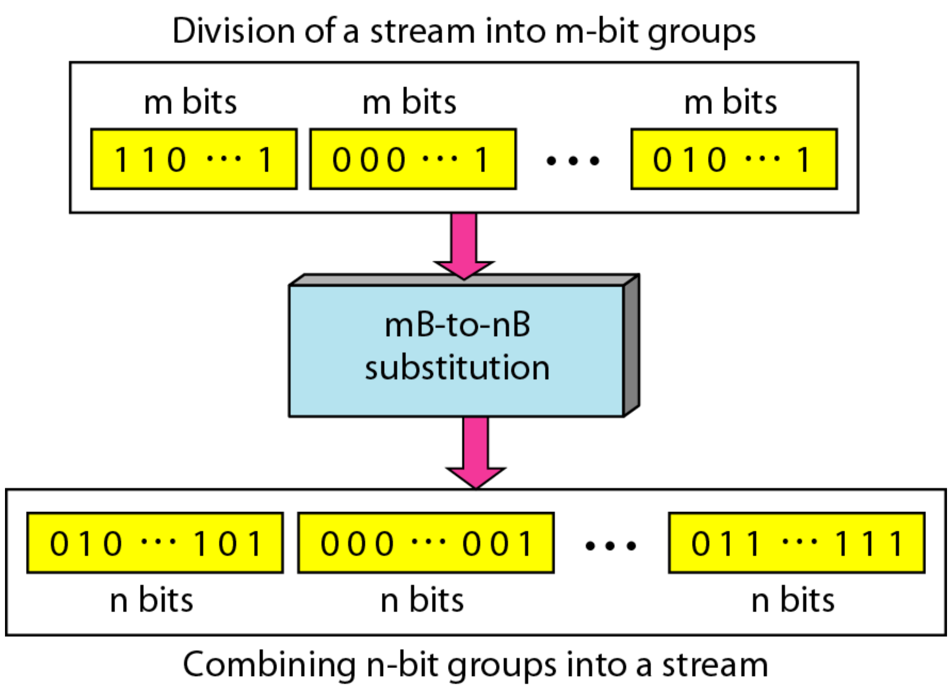
Block Coding

- Redundancy is required
  - To ensure synchronization and
  - To provide some kind of inherent error detecting
- Block coding can give
  - this redundancy and
  - improve the performance of line coding
- In general, block coding **changes a block of *m* bits** into **a block of *n* bits**, where *n* is larger than *m.*
- Block coding is referred to as an ***mB/nB* encoding** technique.

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

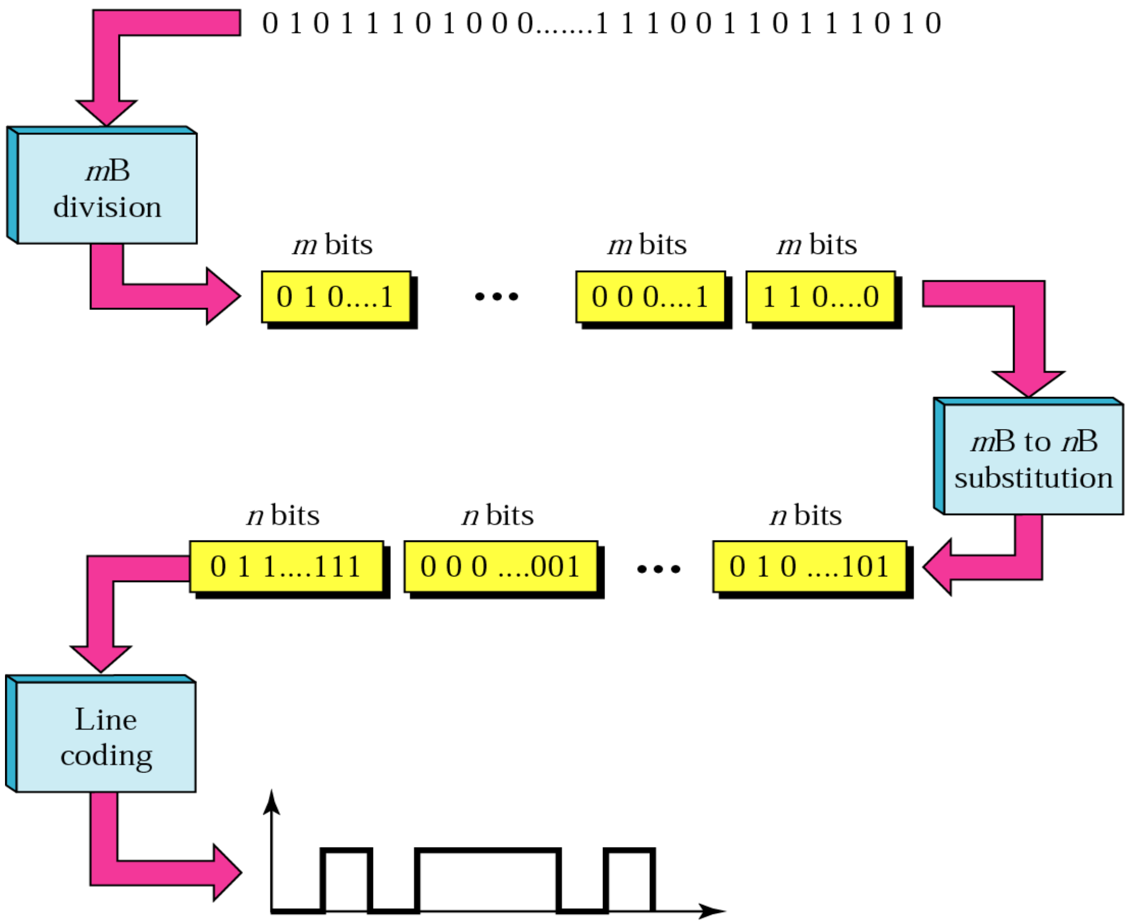
Cont.

- **Block coding is done in three steps:**
  - division,
  - substitution and
  - combination.
- It is distinguished from multilevel coding by use of the slash - xB/yB.



Cont.

- Block coding



Cont.

- **4B/5B Block Coding Scheme**
  - The four binary/five binary (4B/5B) coding scheme

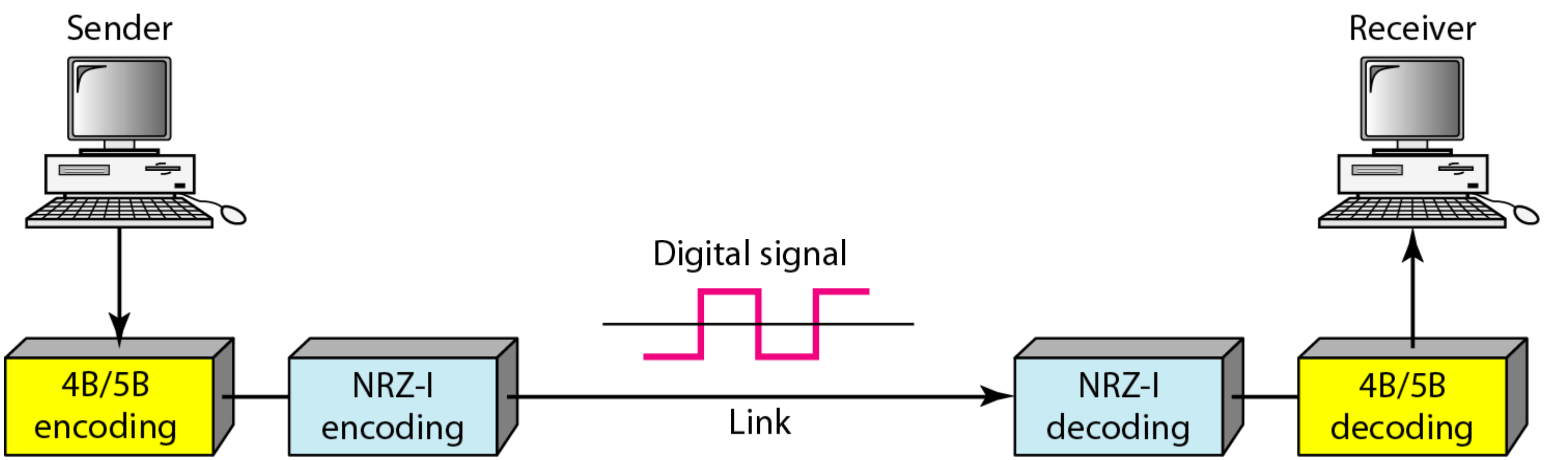


Figure: Using block coding 4B/5B with NRZ-I line coding scheme

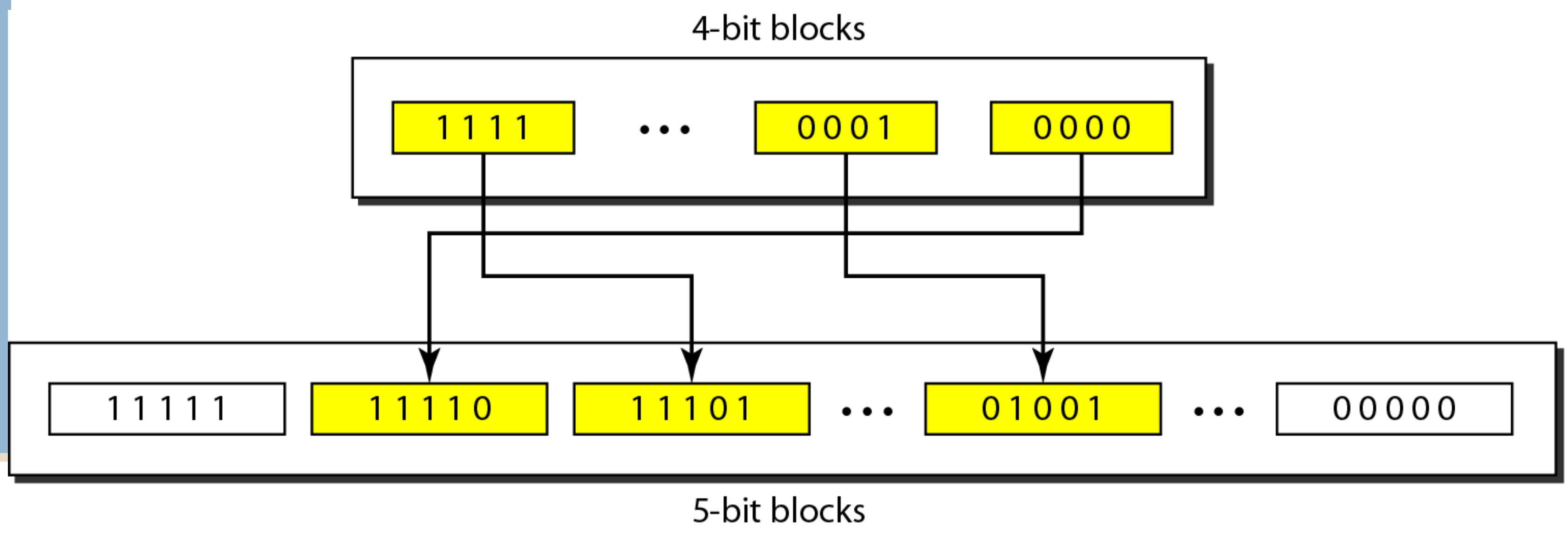
Cont.

Table: ***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		

Cont.

Figure: Substitution in 4B/5B block coding



Cont.

- **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/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.

Cont.

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.

Cont.

- A group of 8 bits of data is now substituted by a l0-bit code.
- The most five significant bits of a 10-bit block is fed into the 5B/6B encoder;
- The least 3 significant bits is fed into a 3B/4B encoder. The split is done to simplify the mapping table.
- To prevent a long run of consecutive 0s or Is, the code uses a **disparity controller**
  - keeps track of excess Os over 1s (or 1s over Os)

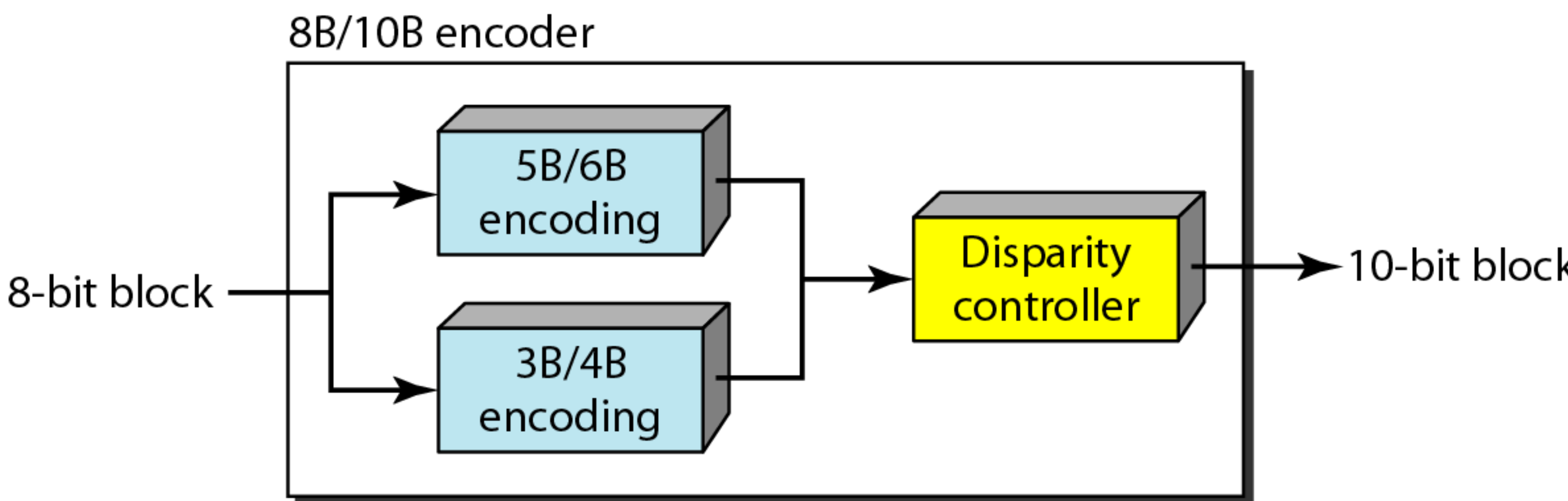


Figure: ***8B/10B block encoding***

Cont.

- **More bits - better error detection**
  - The 8B10B block code adds more redundant bits
  - Choose code words that would prevent a long run of a voltage level that would cause DC components.
  - Greater error detection capability than 4B/5B
  - Better synchronization

Applications

Table: ***Summary of Standard Ethernet implementations***

Characteristics	*lOBase5*	*lOBase2*	*lOBase-T*	*IOBase-F*
Media	Thick coaxial cable	Thin coaxial cable	2UTP	2 Fiber
Maximum length	500m	185 m	100m	2000m
Line encoding	Manchester	Manchester	Manchester	Manchester

Table: ***Summary of Fast Ethernet implementations***

Characteristics	*lOOBase-TX*	*lOOBase-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100m	100m	100m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

Applications

Table: ***Summary of Gigabit Ethernet implementations***

Characteristics	*lOOOBase-SX*	*lOOOBase-LX*	*lOOOBase-CX*	*lOOOBase-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550m	5000m	25m	100m
Block encoding	8B/lOB	*8B/lOB*	8B/lOB	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

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 characteristics for transmission –
  - self clocking,
  - 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
- Two common scrambling techniques are
  - B8ZS and
  - HDB3.

Cont.

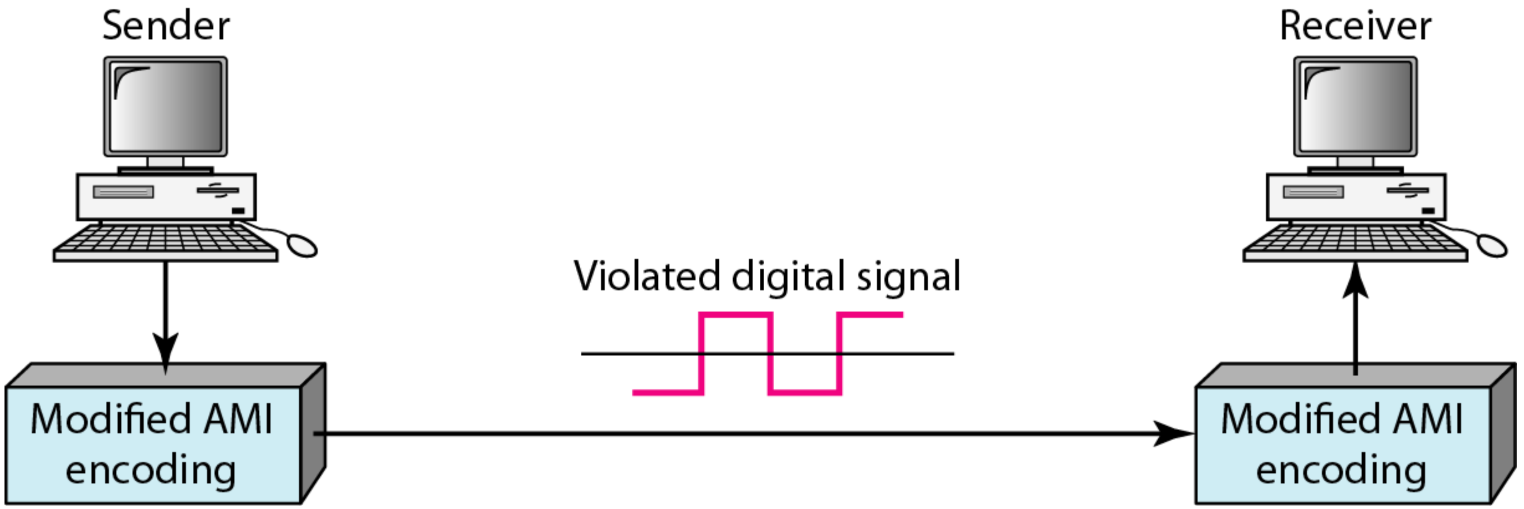


Figure: ***AMI used with scrambling***

Cont.

- **Bipolar with 8 zero substitution (B8ZS):**
  - Commonly used in North America
  - **8 consecutive zero-level** voltages are replaced by the sequence **000VB0VB**
    - The **V** stands for violation, it violates the line encoding rule
      - This is a nonzero voltage that breaks an AMI rule of encoding (opposite polarity from the previous).
    - **B** stands for bipolar, a nonzero level voltage that implements the bipolar line encoding rule

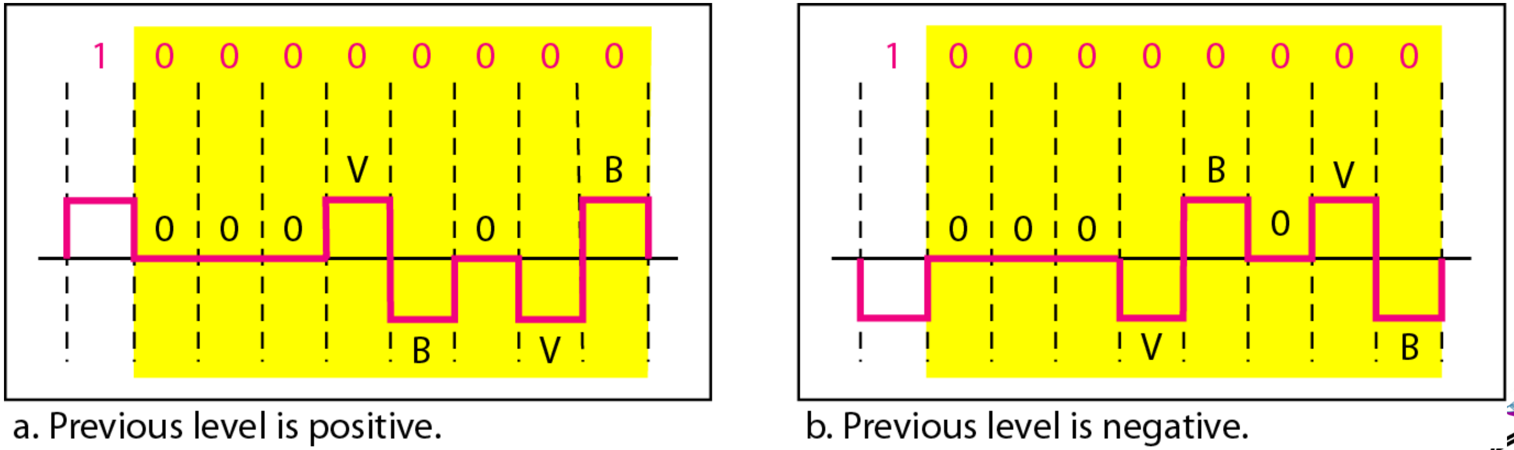
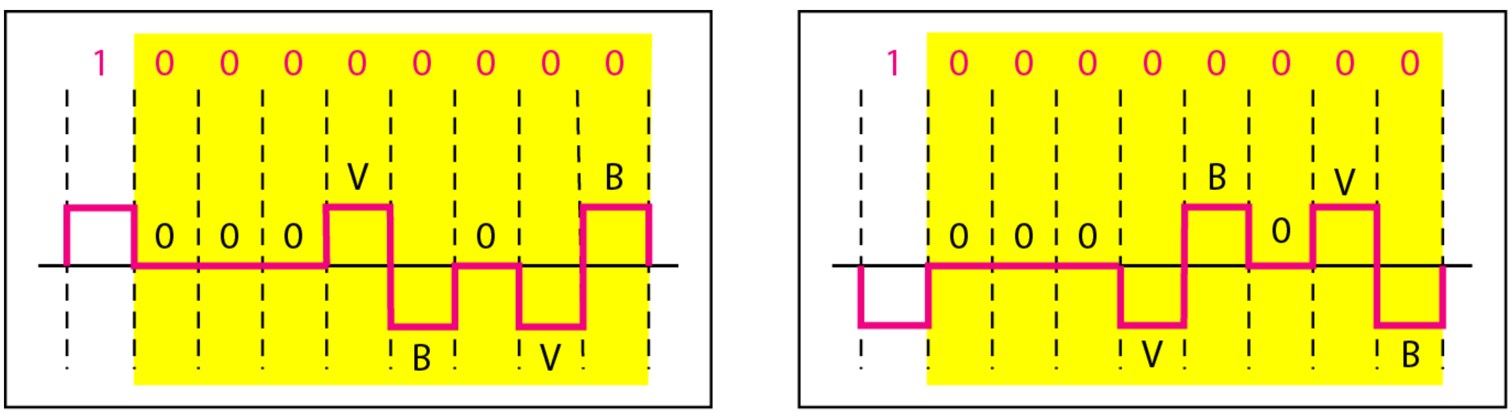


Figure: *Two cases of B8ZS scrambling technique*

Cont.

- **Bipolar with 8 zero substitution (B8ZS):**
  - The scrambling in this case does not change the bit rate.
  - The DC balance is maintained.
  - The substitution may change the polarity of a 1 because, after the substitution, AMI needs to follow its rules.
  - The letter V (violation) or B (bipolar) here is relative.
    - The V means the same polarity as the polarity of the previous nonzero pulse;
    - B means the polarity opposite to the polarity of the previous nonzero pulse.



a. Previous level is positive.

b. Previous level is negative.

Figure: *Two cases of B8ZS scrambling technique*

Cont.

- **High-density bipolar 3-zero (HDB3)**
  - Commonly used outside of North America.

HDB3 substitutes four consecutive zeros with OOOV or BOOV depending on the number of nonzero pulses after the last substitution.

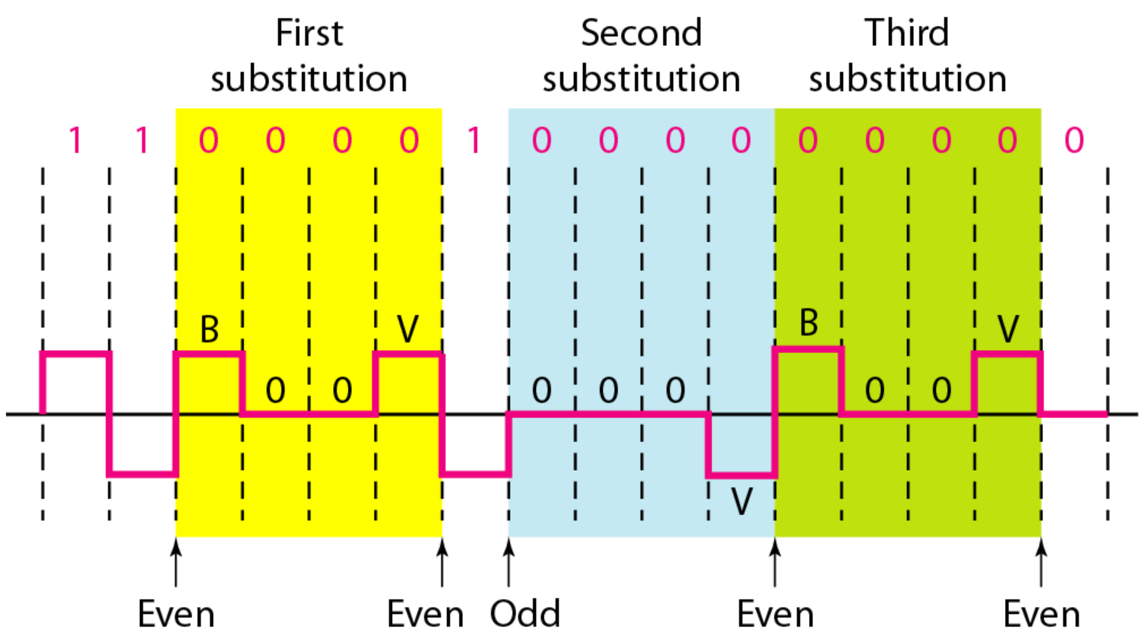


Figure: *Different situations in HDB3 scrambling technique*

Cont.

- **High-density bipolar 3-zero (HDB3)**
  - The two rules can be stated as follows:

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

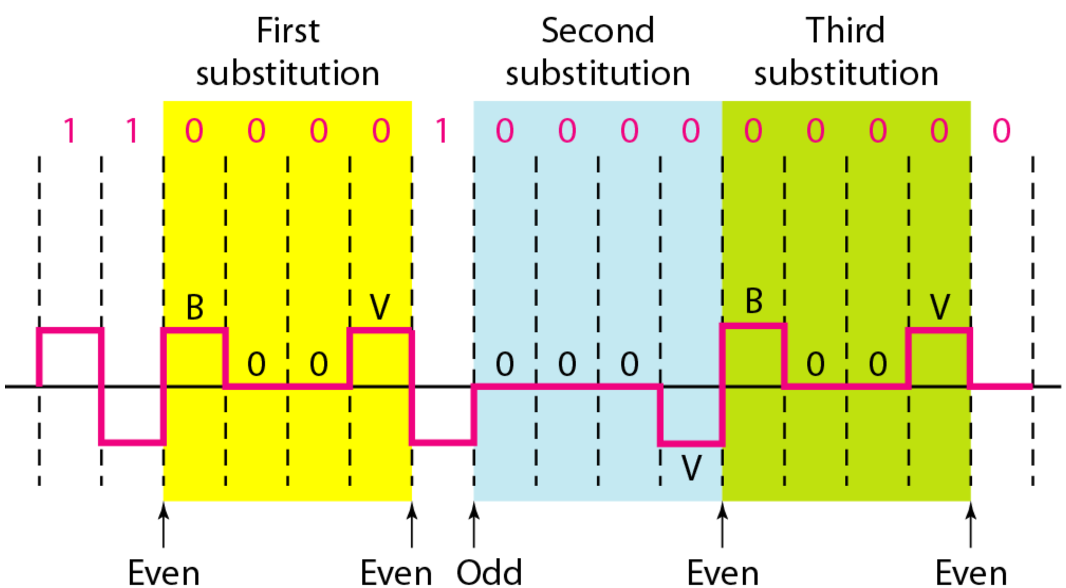


Figure: *Different situations in HDB3 scrambling technique*

Reference Books

- Data Communications and Networking: Behrouz A. Forouzan