CSE 320
Data Communications

Digital Transmission

Chapter 4

Recap Previous Lecture

- Digital data
  - Digital Signal
  - Analog Signal
- Analog data
  - Digital Signal
  - Analog Signal

Encoding Techniques

- Digital Data, Digital Signal
  - Less expensive and less complex than digital to analog modulation.
- Digital Data, Analog Signals
  - Some transmission media, such as optical fiber and unguided media, will only propagate analog signals.

Digital Data → Digital Signal

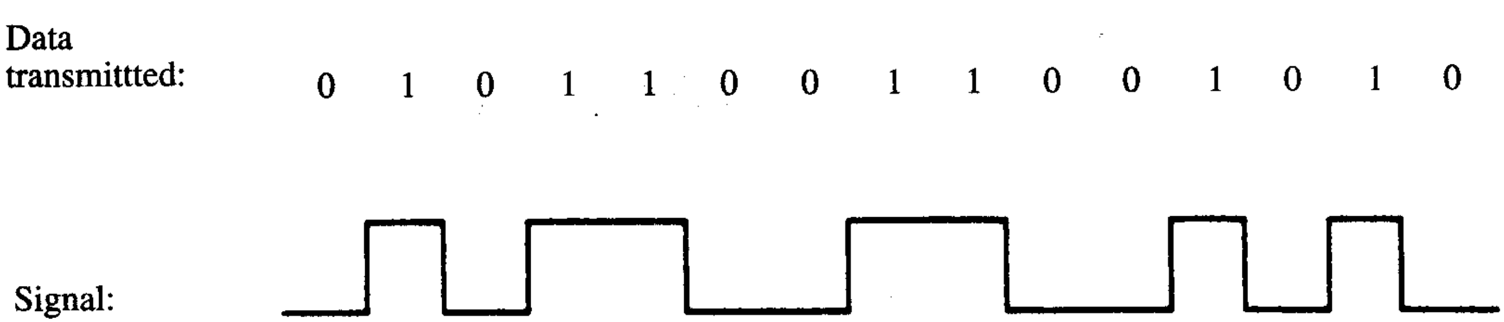
Signal Encoding Techniques:

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

Line coding is always needed; block coding and scrambling may or may not be needed.

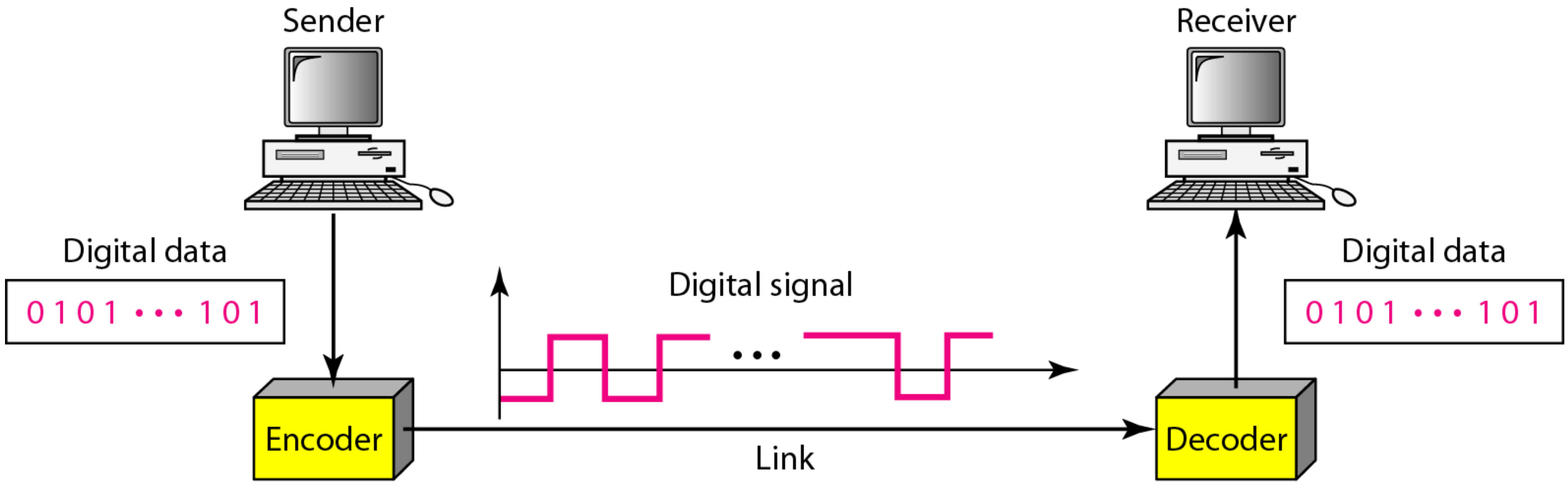
Digital Data, Digital Signal

- Digital signal
  - Discrete, discontinuous voltage pulses
  - Each pulse is a signal element
  - Binary data encoded into signal elements

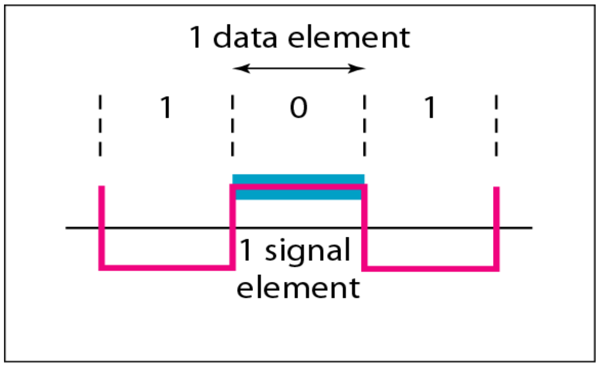


Line Coding

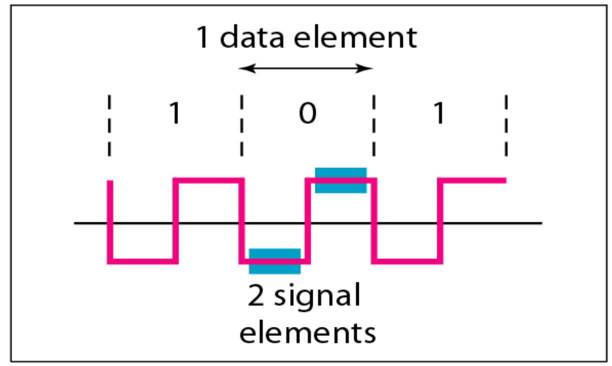
Line coding is the process of converting binary data, a sequence of bits, to a digital signal.



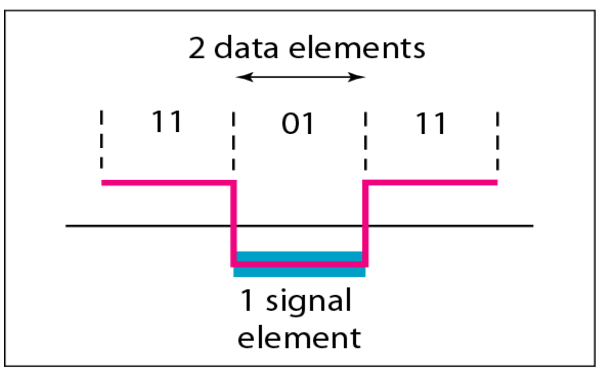
Signal Vs Data Element



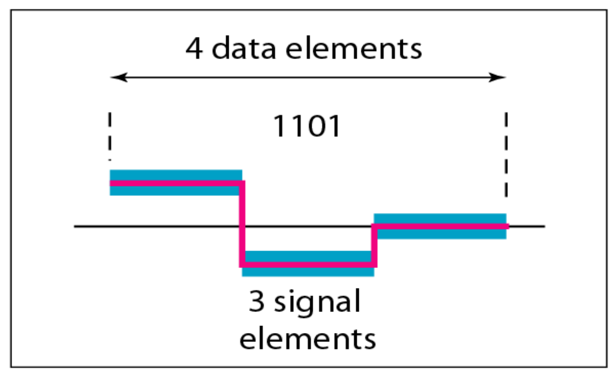
a. One data element per one signal element ($r = 1$)



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

Pulse /Modulation /Signal/ Baud Rate versus Bit Rate

- The pulse rate defines the number of pulses/signals sent in one second. Also known as Baud Rate.
- The bit rate defines the number of bits per second.

$$BitRate = PulseRate \times \log_2 L$$

$$L = Number\ of\ data\ levels$$

Digital Data → Digital Signal

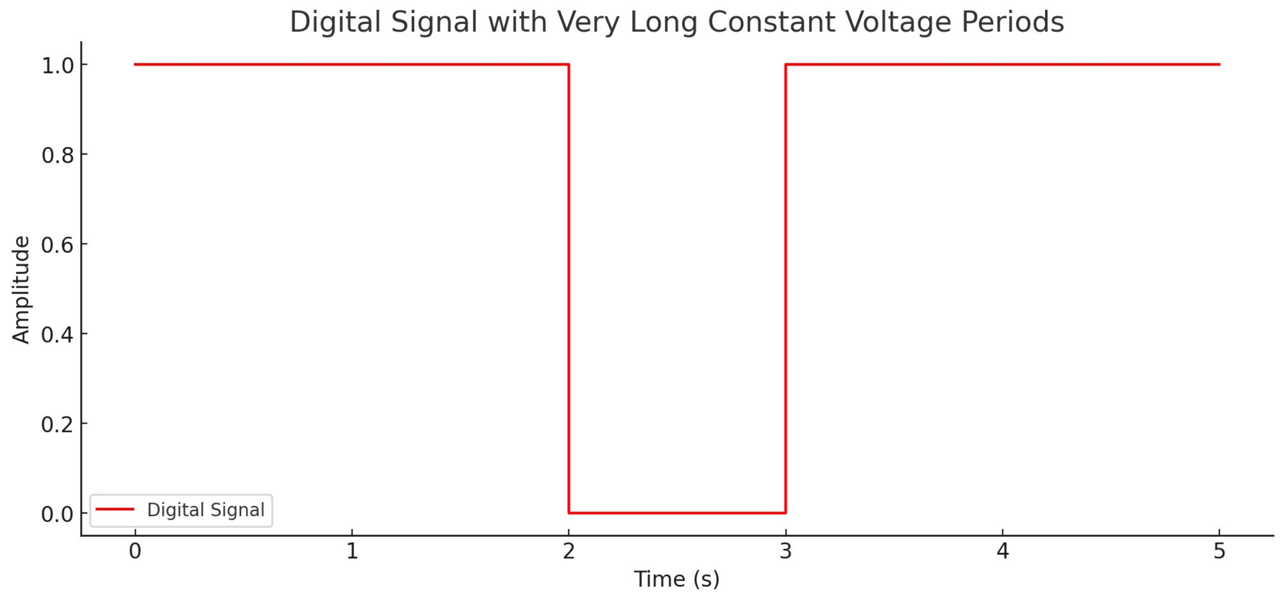
- Receiver needs to know
  - Timing of bits
  - Signal levels
- Factors affecting successful interpretation of signals
  - Baseline Wandering
  - DC Components
  - Self-synchronization
  - Built in Error Detection
  - Immunity to Noise and Interference
  - Complexity

BaseLine Wandering

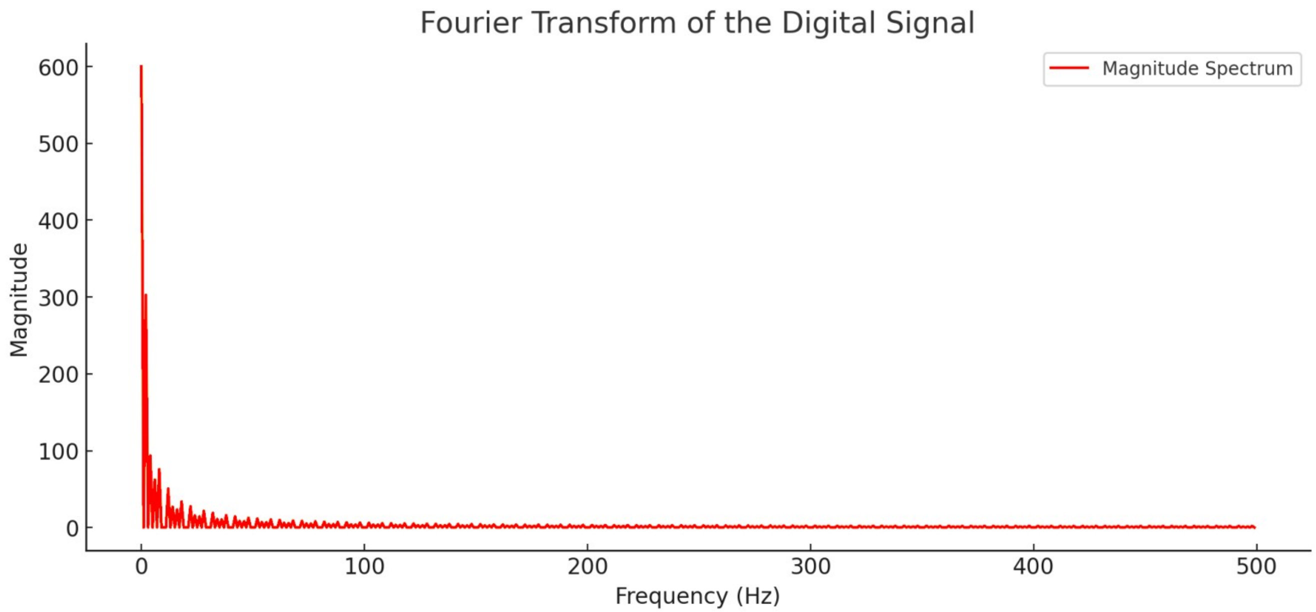
- Receiver calculates the running average of received signal power.
- Average = baseline
- A log strings of 0’s and 1’s can cause a drift in the baseline making it difficult for the receiver to decode properly.

DC Components

- When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies (results of Fourier analysis).
- If the signal is to pass through a system (such as a transformer) that does not allow the passage of a dc component, the signal is distorted and may create errors in the output.
- This component is extra energy residing on the line and is useless.



Digital Signal with Very Long Constant Voltage Periods
Amplitude
1.0
0.8
0.6
0.4
0.2
0.0
Digital Signal
0
1
2
3
4
5
Time (s)

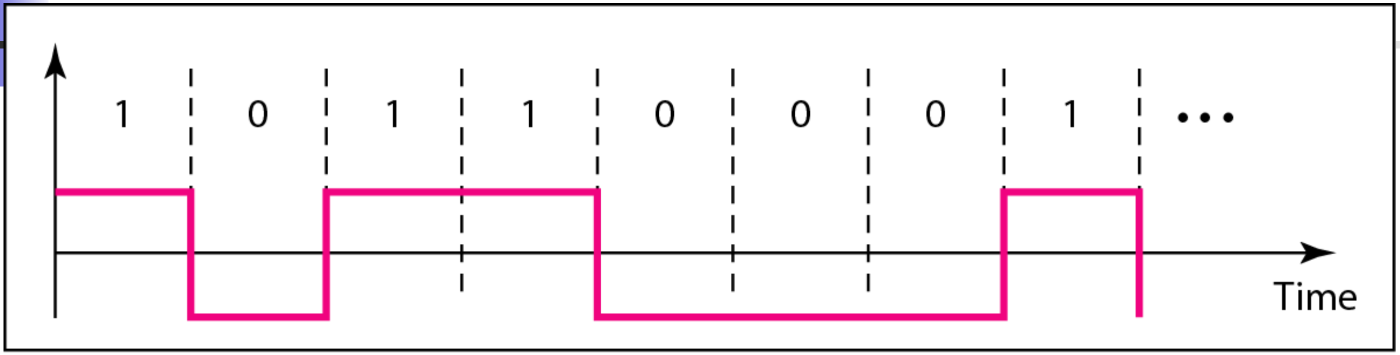


Fourier Transform of the Digital Signal
Magnitude Spectrum
Magnitude
600
500
400
300
200
100
0
0
100
200
300
400
500
Frequency (Hz)

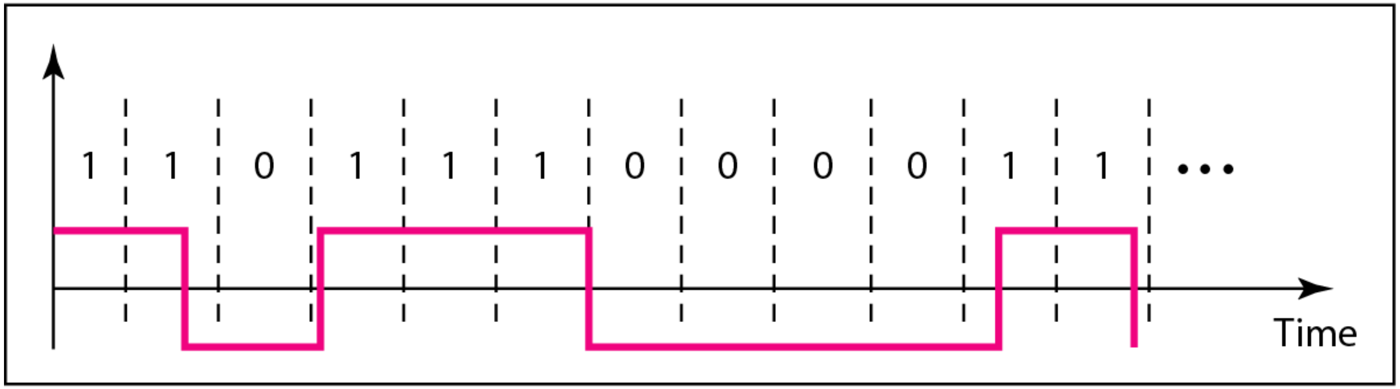
Self-Synchronization

- The receiver's bit intervals must correspond exactly to the senders 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

Self-Synchronization

- A self-synchronizing digital signal includes timing information in the data being transmitted.
- This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.
- If the receiver’s clock is out of synchronization, these points can reset the clock.

Different Line Encoding Schemes

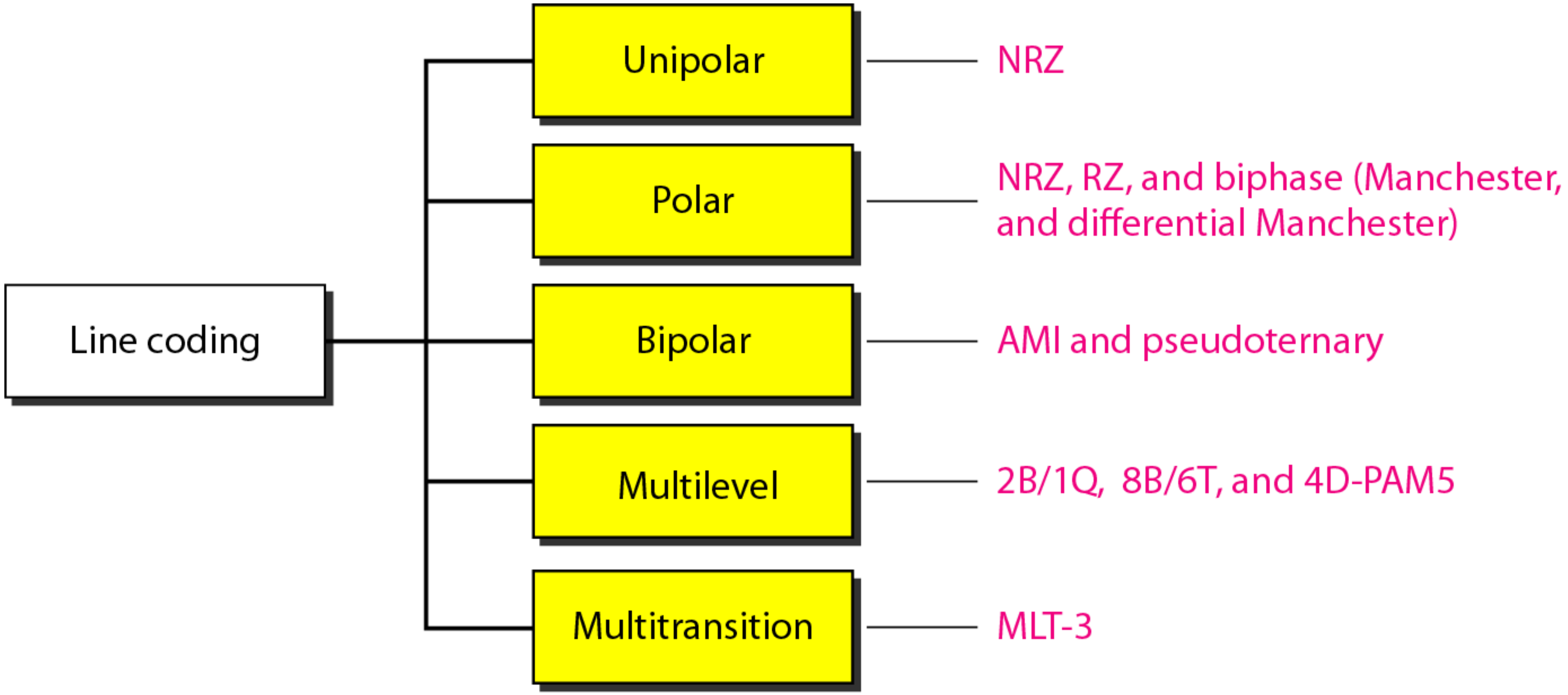
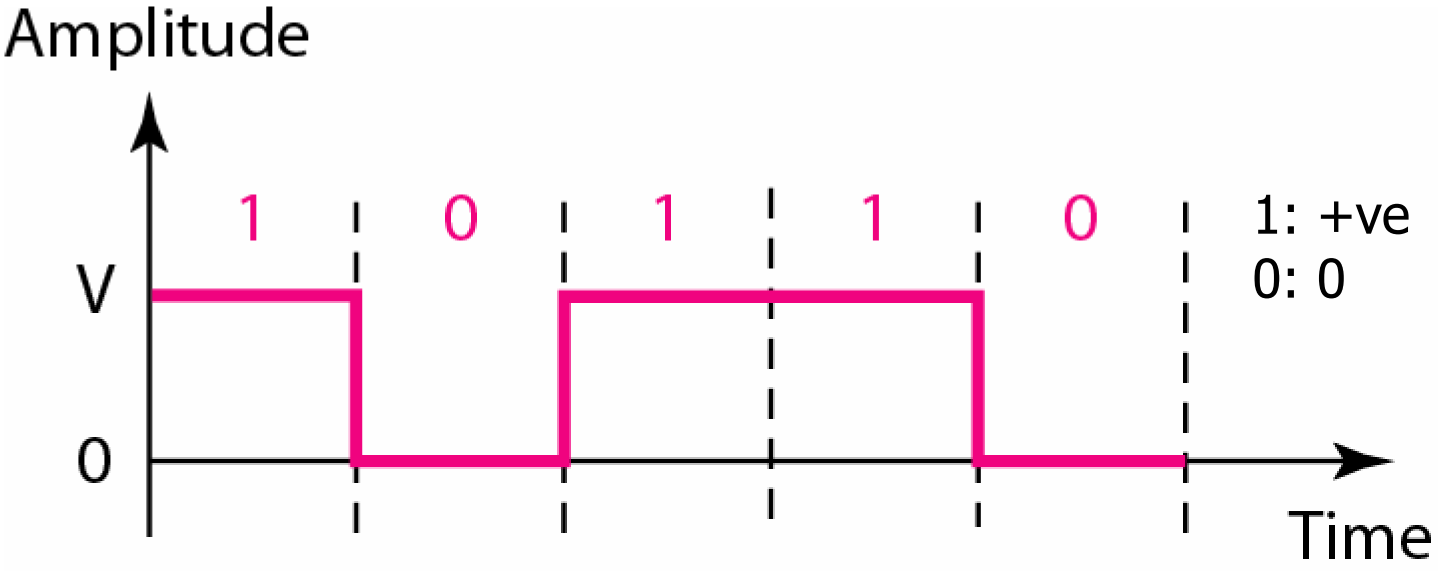


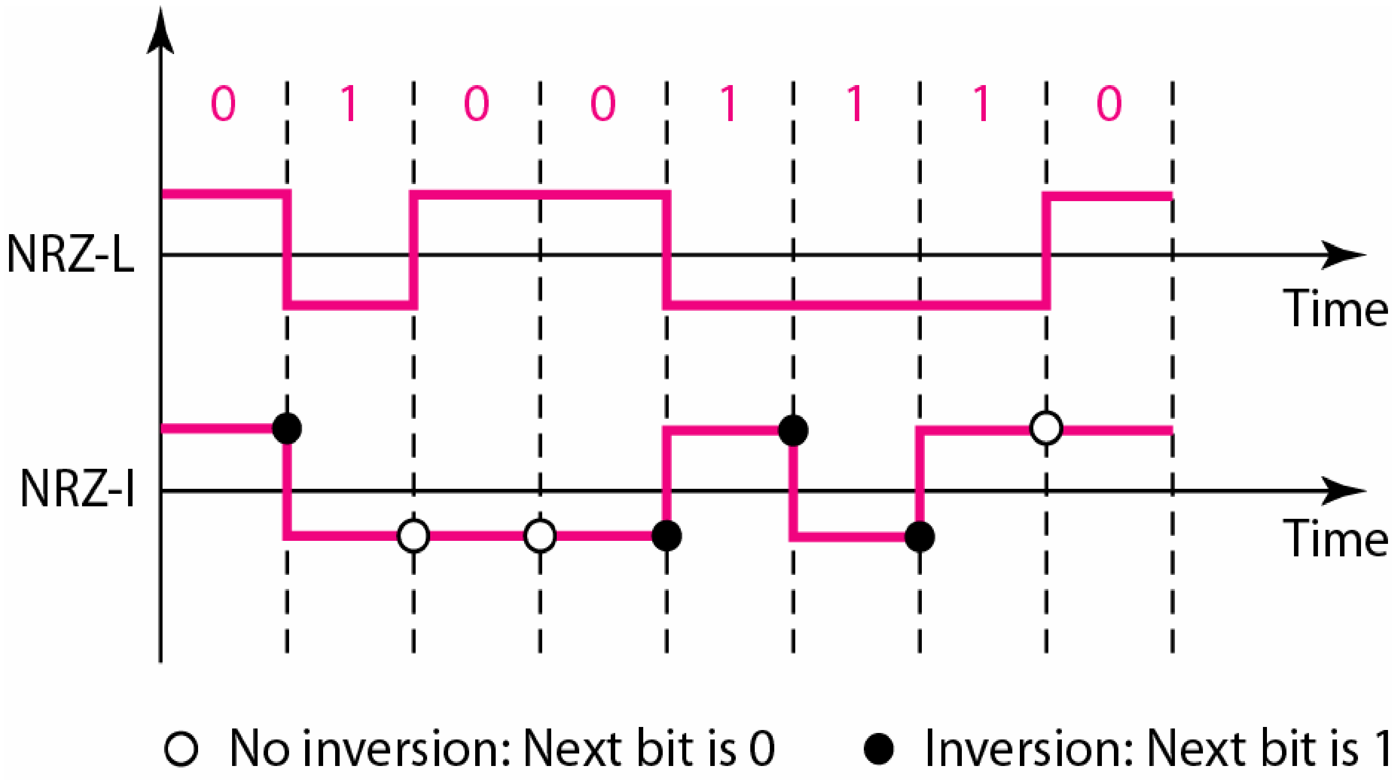
Table 4.1 *Summary of line coding schemes*

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

Unipolar



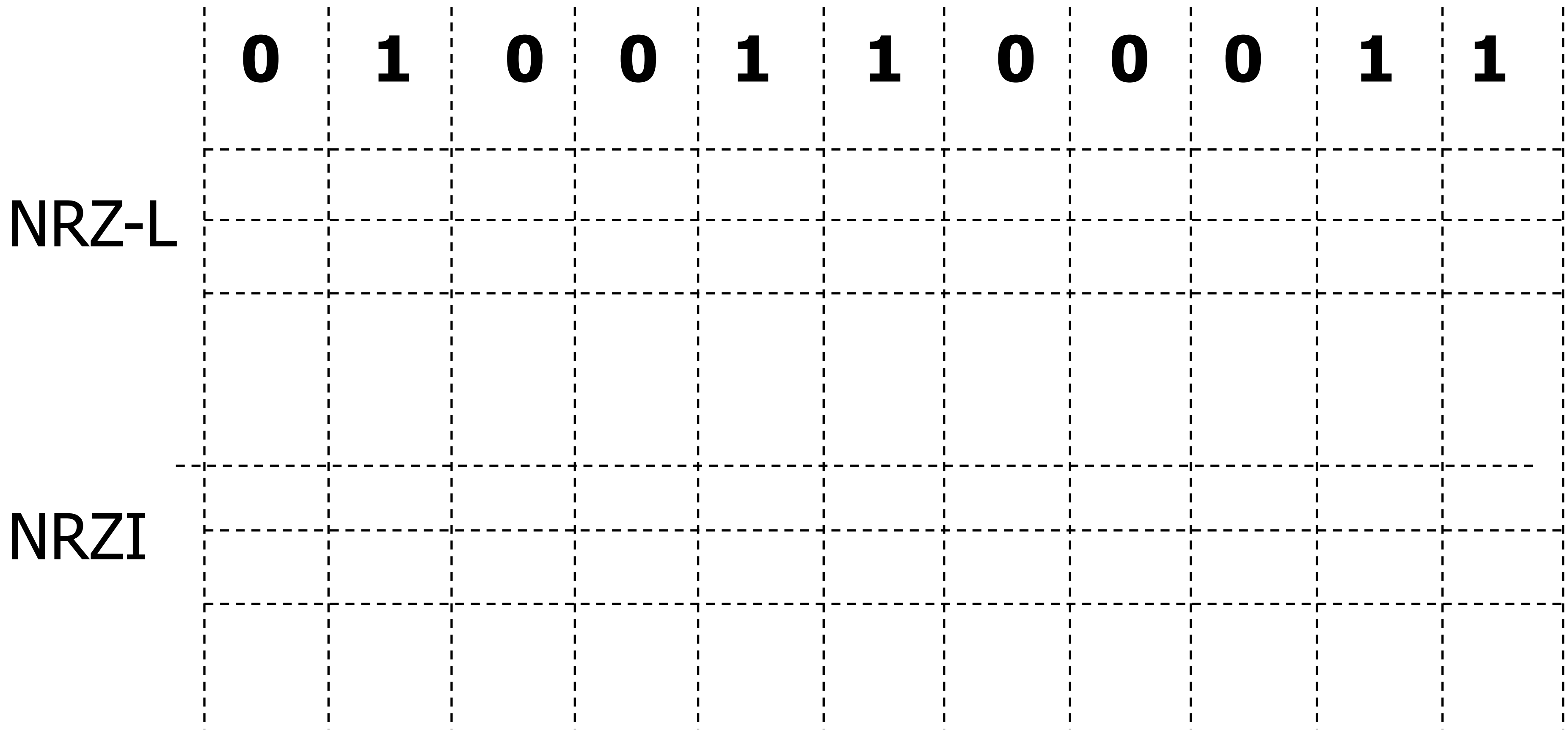
NRZ-L & NRZ-I (Bipolar)



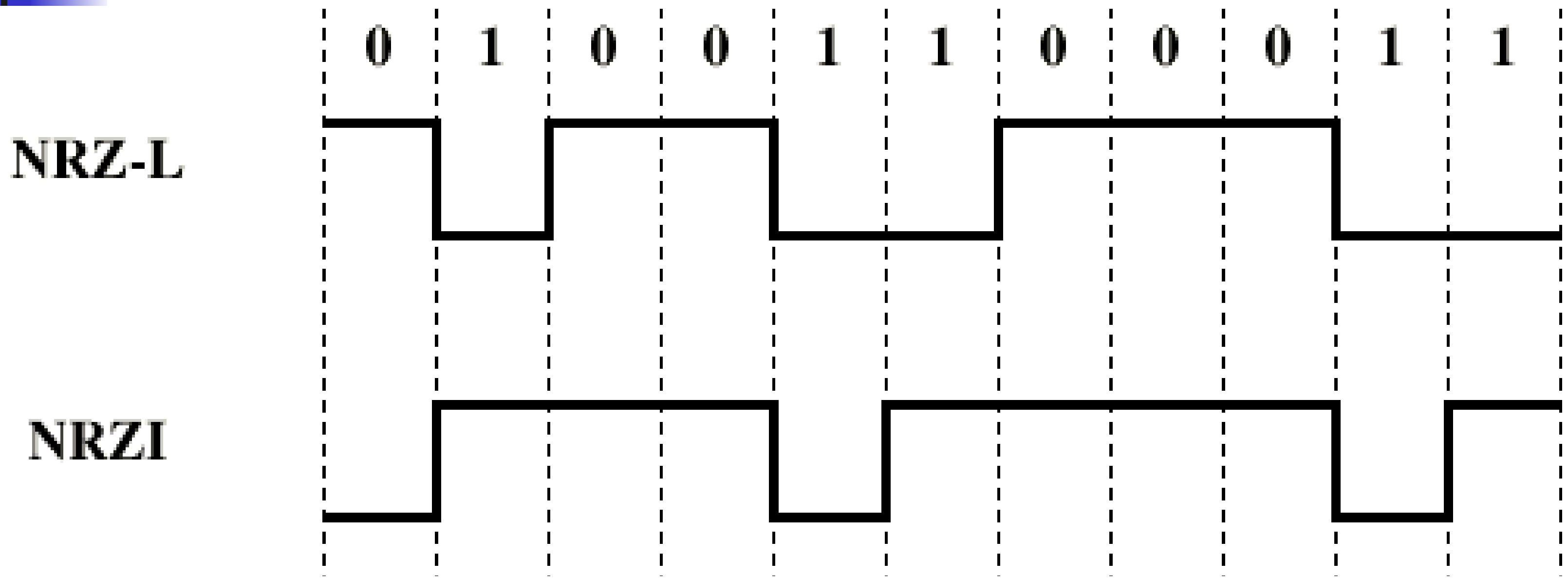
0: +ve
1: -ve

start with +ve
0: no change
1: invert

Nonreturn to Zero (NRZ)



NRZ



NRZ – Pros and Cons

- Pros
  - Easy to engineer
  - Make good use of bandwidth (r=1)
- Cons
  - DC component
  - Lack of synchronization capability
- Used for magnetic recording
- Not often used for signal transmission

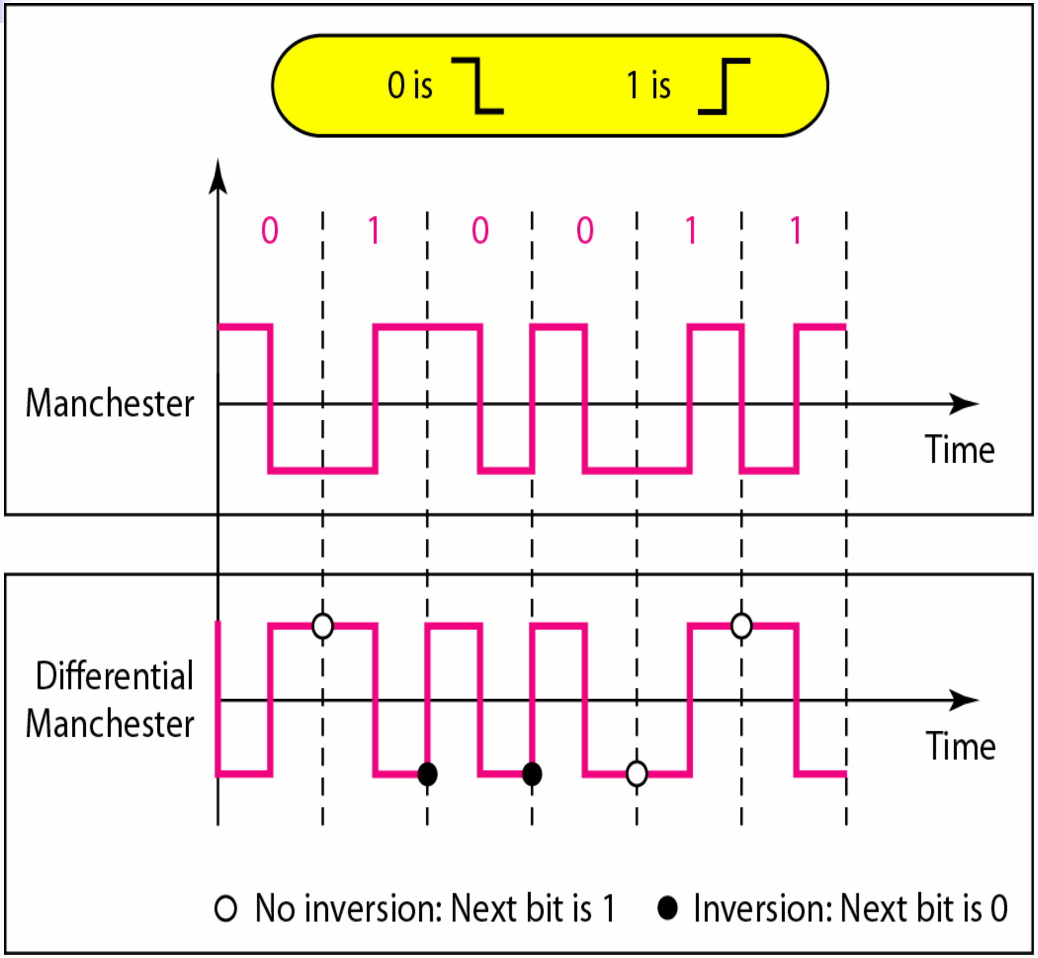
NRZ-L vs NRZ-I

- NRZ-L
  - Baseline wandering, DC components
- NRZ-I
  - No baseline wandering and DC components for consecutive 1s
  - Still exist for consecutive 0s

Differential Encoding

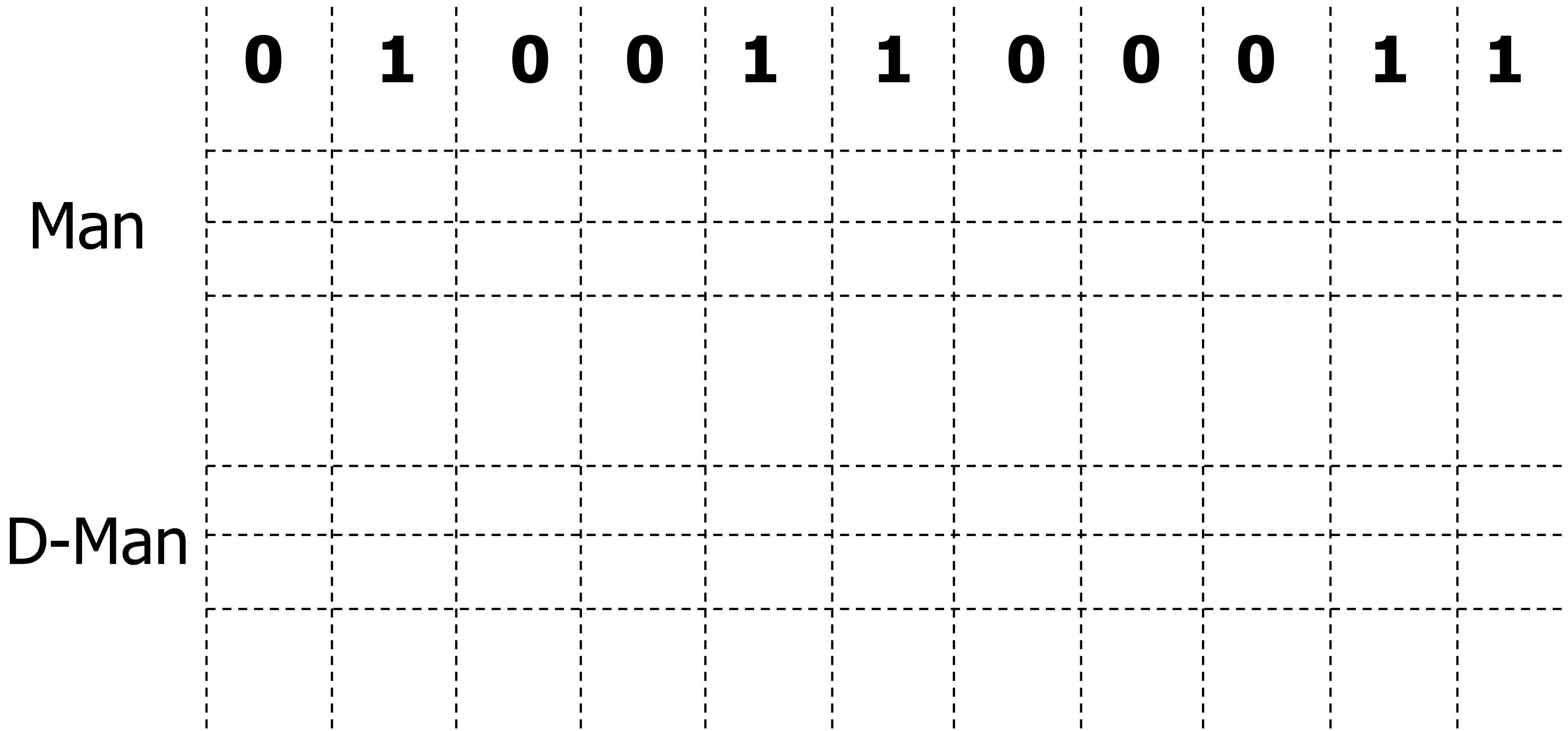
- In complex transmission layouts, it is easy to lose sense of polarity
- Therefore
  - Data represented by changes (i.e., transitions) rather than levels
  - More reliable detection of transition rather than level

Manchester Encoding & Differential Manchester Encoding



start with +ve
0: transition to opposite level
1: no transition

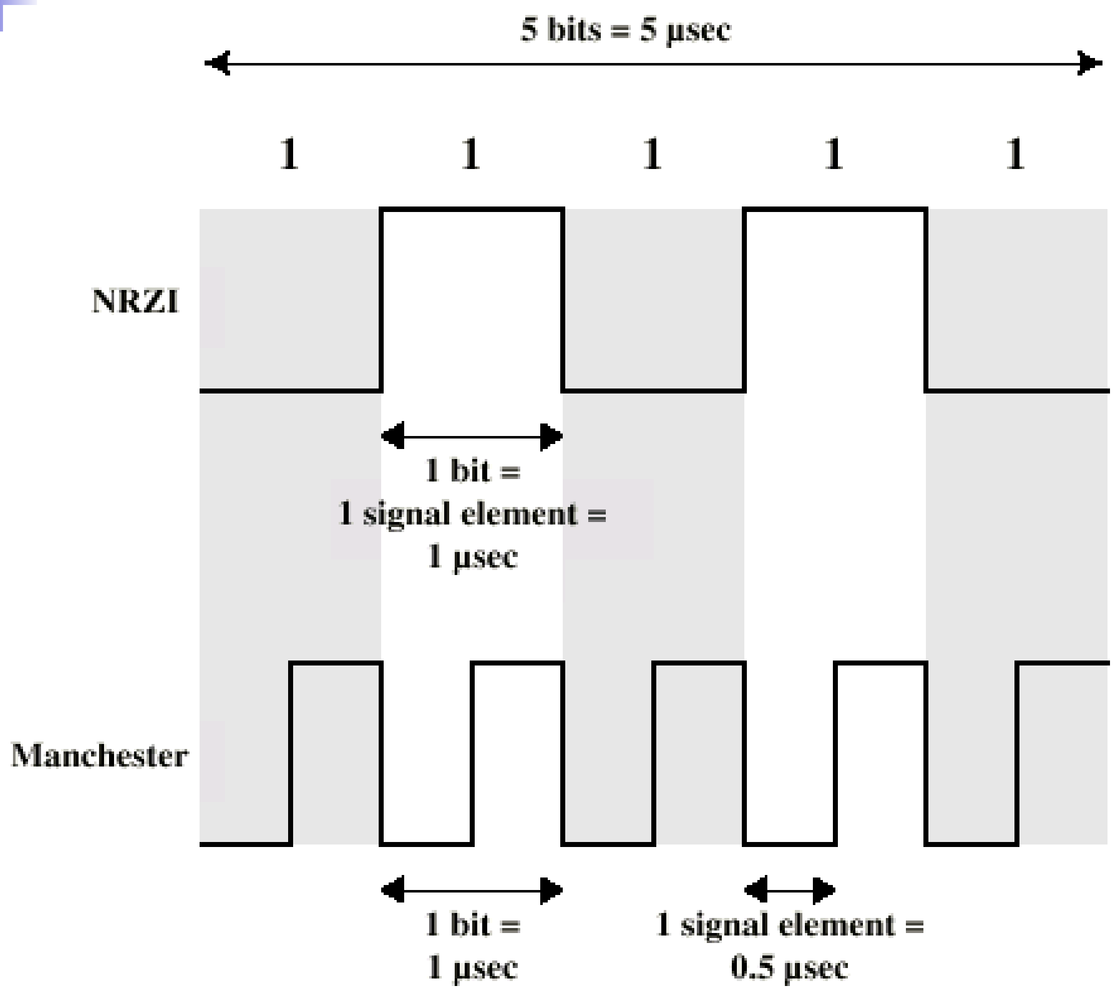
Biphase (Manchester and D-Manchester)



Biphase -- Pros and Cons

- Pros
  - Synchronization on mid bit transition (self clocking)
  - No dc component or baseline wandering
  - Error detection
    - Absence of expected transition
- Cons
  - At least one transition per bit time and possibly two
  - Maximum modulation rate is twice NRZ
  - Requires more bandwidth (r=0.5)

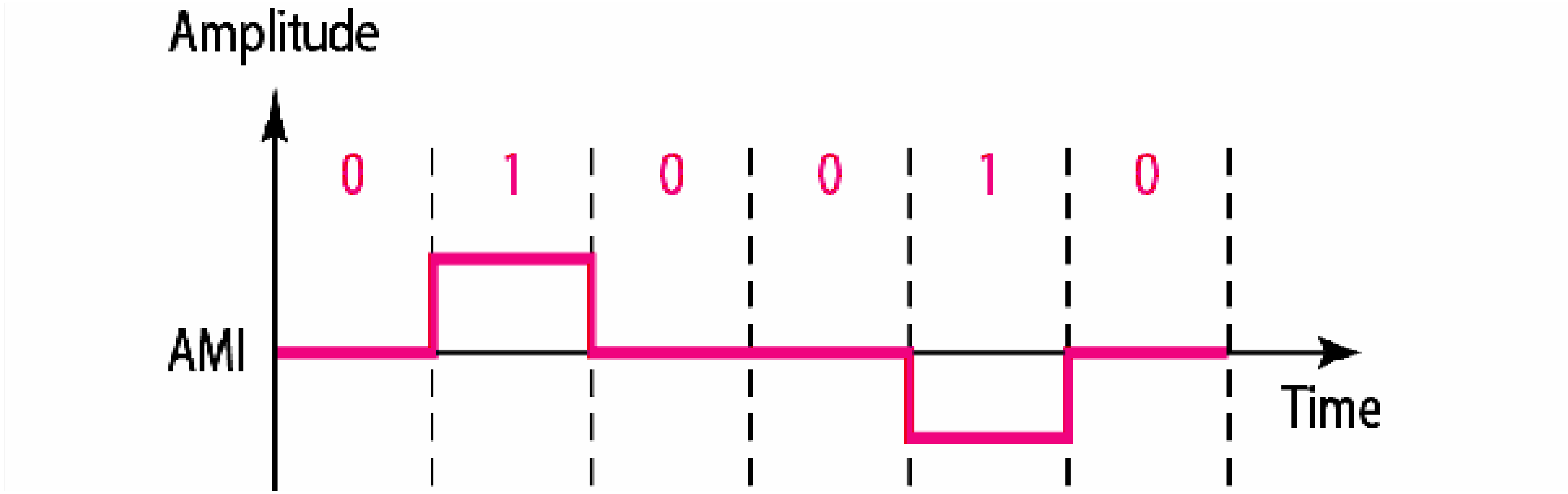
Modulation Rate



Multilevel Binary

- Use more than two levels
- Bipolar-AMI
  - zero represented by no line signal
  - one represented by positive or negative pulse
  - No baseline wandering for consecutive 1s (zeros still a problem)
  - Good use of bandwidth

Bipolar-AMI



- 0: 0
- 1: opposite of previous non-zero level
- First 1 is +ve

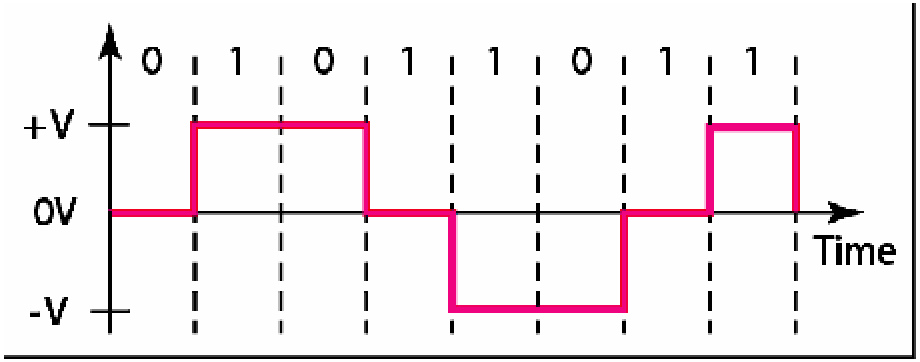
Bipolar AMI

- Baseline wandering for consecutive 0s but not for 1s
- No DC components issue
- Lack of synchronization for a long sequence of 0s
- Good use of bandwidth (r=1)

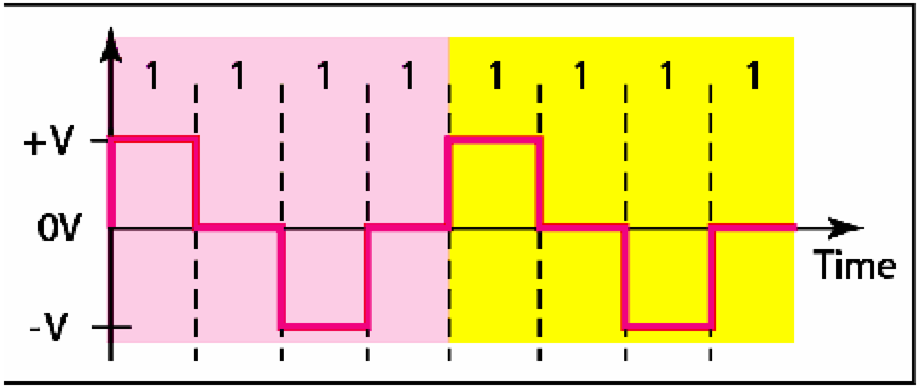
Multilevel Transition, three level- MLT-3

- There is no transition for a 0 bit.
- The signal transitions from one level to the next at the beginning of a 1 bit
- Transition occurs using three levels of signals (+1, 0, -1)
- Start with 0
- First 1 is positive, next 1 is zero, next 1 is negative, next 1 is zero, next 1 is again positive, and so on
- Staircase pattern

MLT-3



a. Typical case

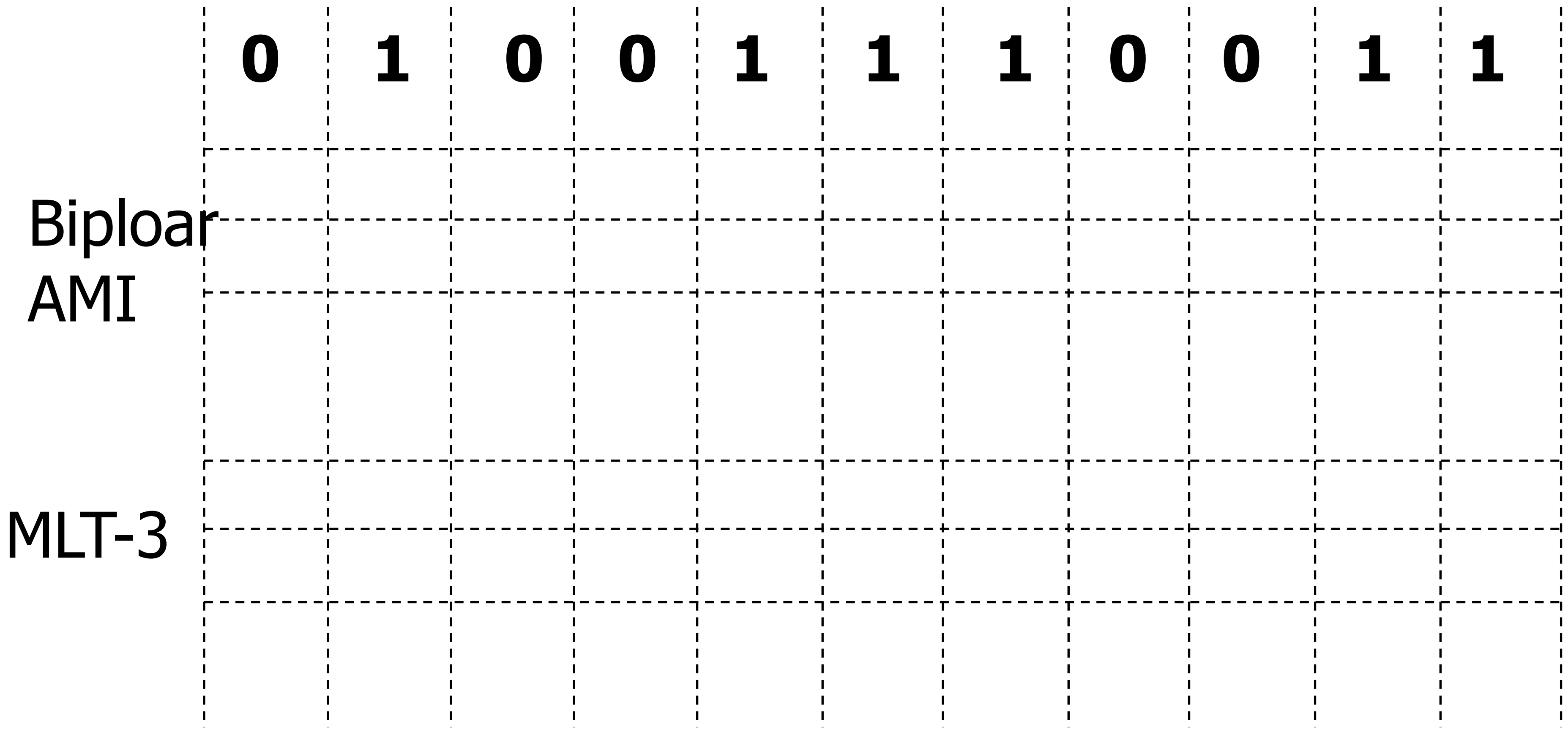


b. Worse case

MLT-3

- Baseline wandering for consecutive 0s only, not for 1s
- DC component is an issue
- Lack of synchronization
- Good use of bandwidth (r=1)
    - This is similar to AMI but actually more efficient in practical applications

Biploar AMI and MLT-3 Example



Block Coding

Block Coding/Scrambling

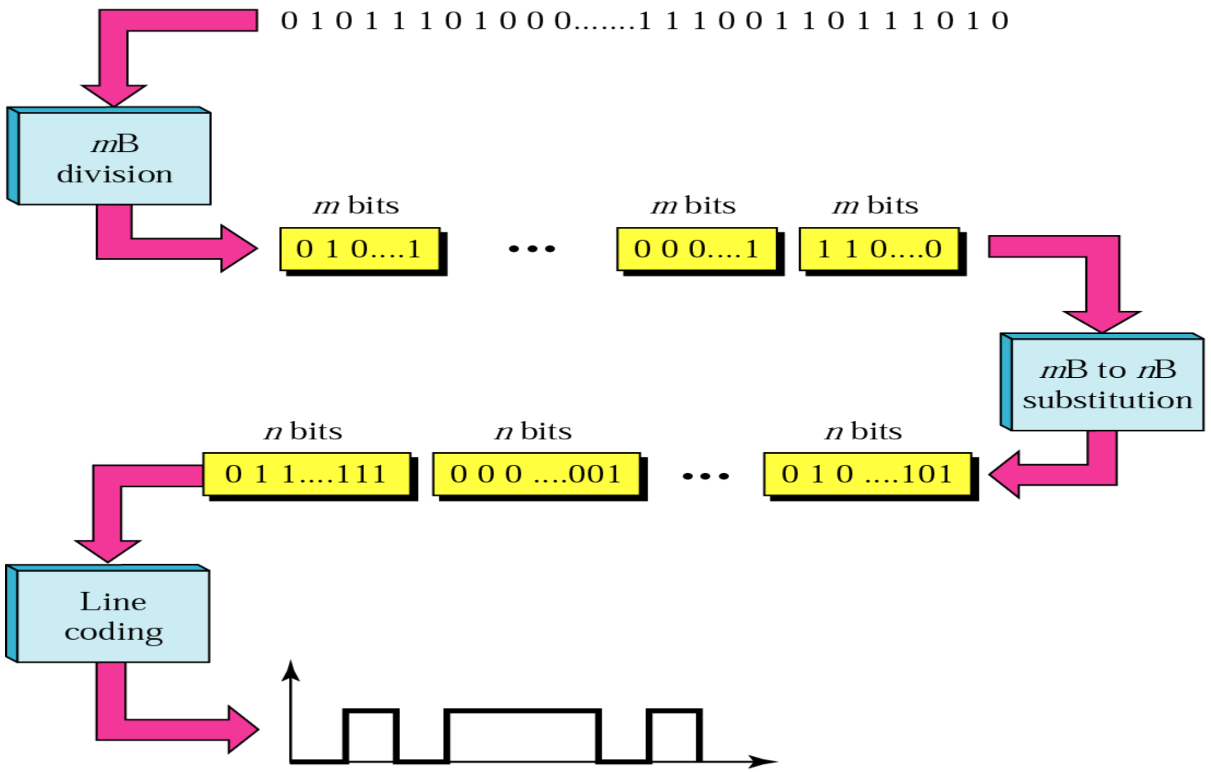
- NRZ, Bipolar AMI, MLT-3 all has a common problem.
- Long sequence of 0 can make the receiver lose synchronization
- Solutions:
  - Change the bit stream before encoding with NRZ-I so that there is no long streams of 0s.

Solutions:

- Block Coding
- Scrambling
- Block Coding
  - Changes a block of m bits to a block of n bits.
  - Referred to as mB/nB encoding.

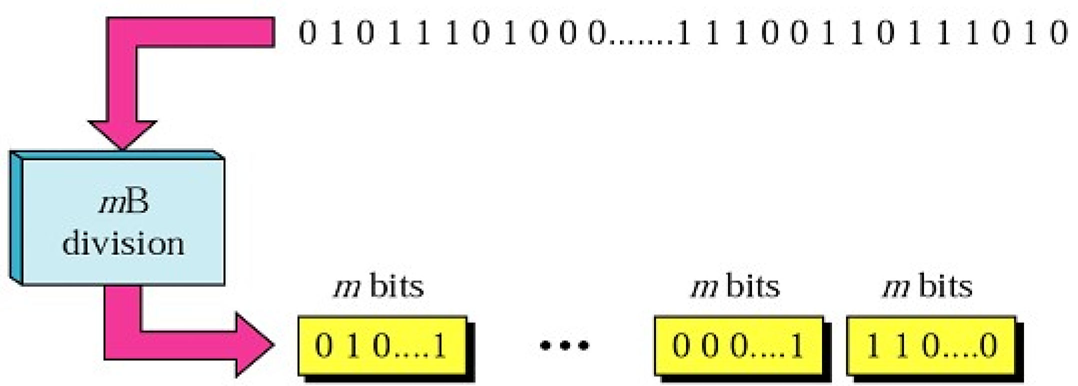
Block Coding

- Three Steps Process:
  - Division
  - Substitution
  - Line Coding /Combination



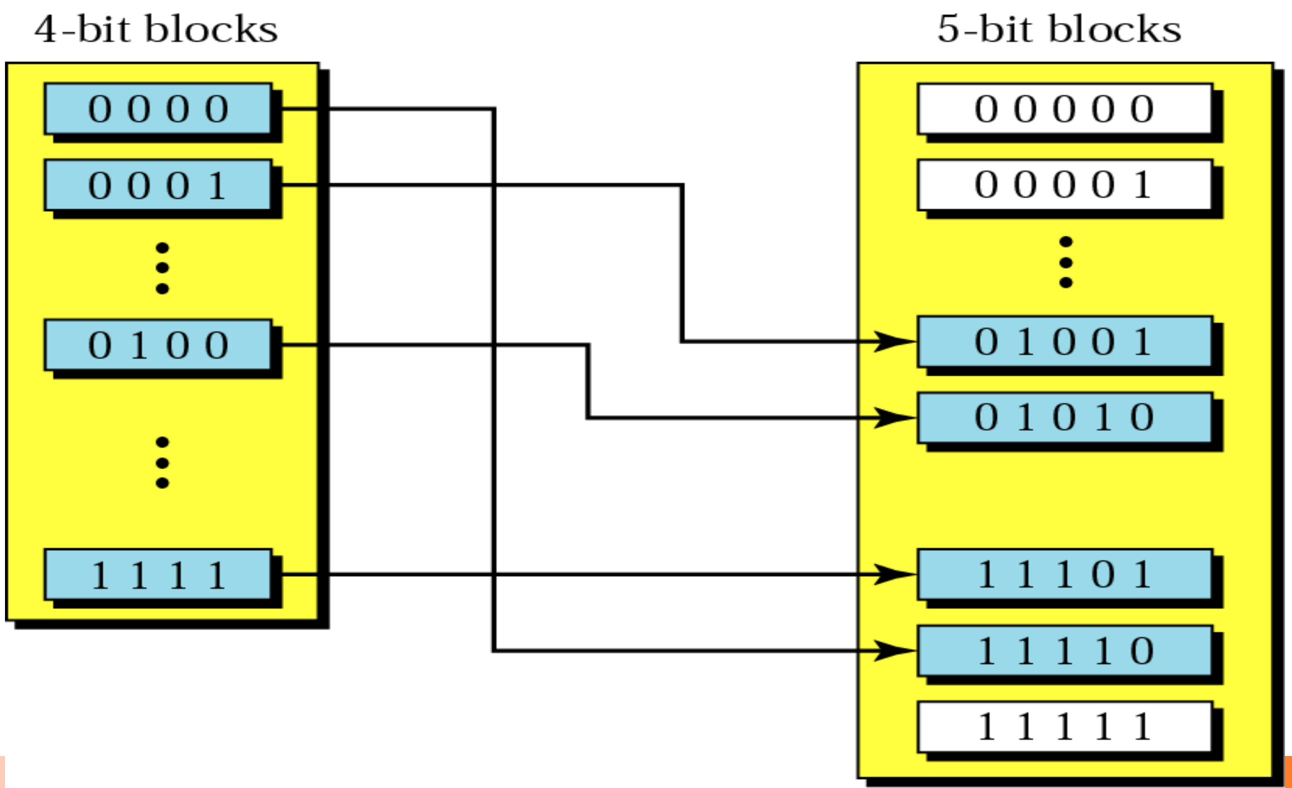
Step 1-Division

- The sequence of bits in data in divided into m Bits.
- For example in 4B/5B encoding, the original bit sequence is divided into 4-bit codes/sequence.



Step 2-Substitution

- Each m bits sequence is substituted for a n bit code.



4B/5B Block Coding

- 4-bit code ->16 different combinations
- 5-bit code -> 32 possible combinations.
- So not all of 5-bit codes are required.
- Selection of the 5-bit code is such that each code contains no more than
  - "one leading 0 and no more than two trailing 0s." (3 consecutive 0s)

Table : 4B/5B encoding

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		

Step 3: Line Coding

- After substitution, a line coding scheme, exp NRZ-I is chosen to create a signal.
- A very simple line coding scheme is chosen, because the block coding procedure provides
  - two desirable features (??) of complex line coding schemes.

Block Coding-Pros/Cons

- Solves the synchronization problem but not the DC component problem.
- If DC is unacceptable, use bipolar or biphase encoding.
- Increases the baud rate by 20%, still better than Manchester schemes.

Scrambling

- Use scrambling to replace sequences that would produce constant voltage
- Filling sequence
  - Must be recognized by receiver and replace with original
  - Same length as original
- Design Goals
  - No dc component
  - No long sequences of zero level line signal
  - No reduction in data rate
  - Error detection capability

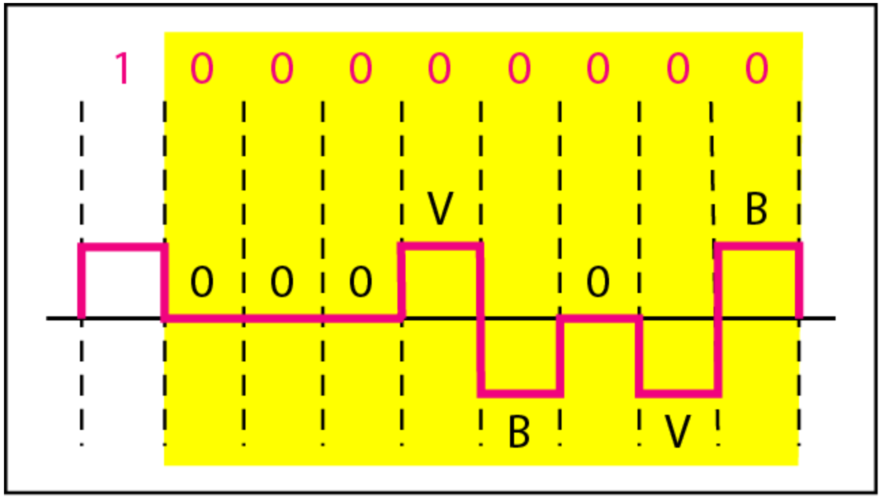
Types of Scrambling:

- B8ZS
  - Bipolar With 8 Zeros Substitution
  - Commonly used US.
- HDB3
  - High Density Bipolar 3 Zeros
  - Based on Bipolar AMI
  - Commonly used Europe and Japan.

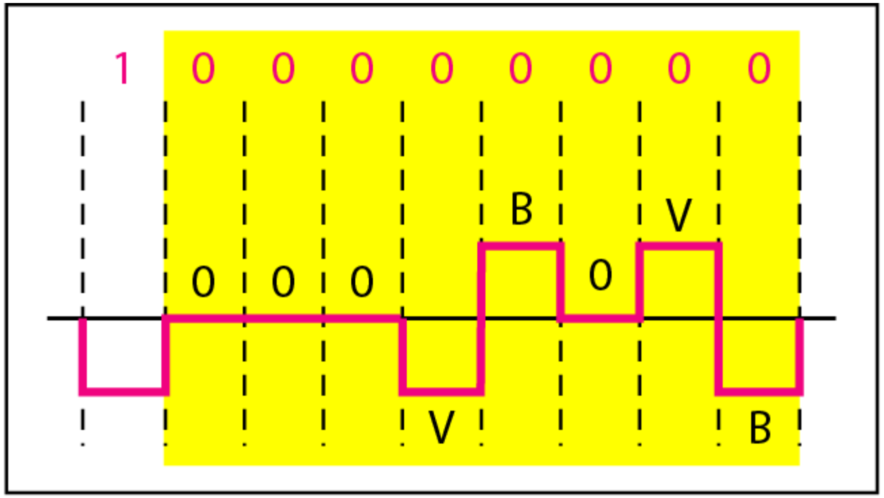
B8ZS

- Based on bipolar-AMI
  - If octet of all zeros and last voltage pulse preceding was positive encode as 000+-0-+
  - If octet of all zeros and last voltage pulse preceding was negative encode as 000-+0+-
- Causes two violations of AMI code
  - Unlikely to occur as a result of noise

B8ZS



a. Previous level is positive.



b. Previous level is negative.

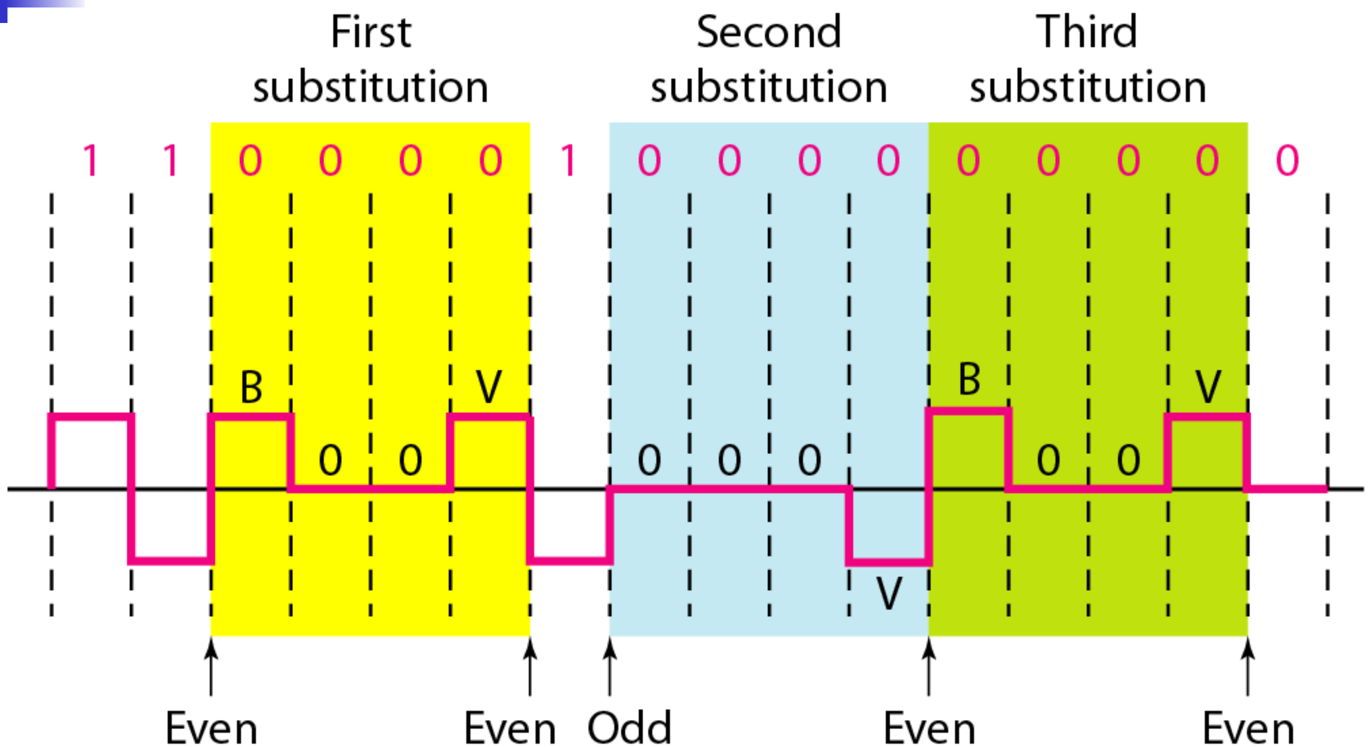
HDB3

- High Density Bipolar 3 Zeros
- Based on Bipolar AMI, Commonly used Europe and Japan.
- Based on bipolar-AMI
- String of four zeros replaced with one or two pulses

HDB3 Subtitution Table

Polarity of Preceding Pulse	Number of Bipolar Pulses (ones) since Last Substitution	
	Odd	Even
–	000–	+00+
+	000+	–00–

HDB3



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