Analog/Digital Conversions

- Analog data are a function of time and occupy a limited frequency spectrum; such data can be represented by an electromagnetic signal occupying the same spectrum.
- Digital data can be represented by digital signals, with a different voltage level for each of the two binary digits

But these are not the only possibilities

- **Digital data** can also be represented by **analog signals** by use of a **modem** (modulator/demodulator)
- Similarly, **analog data** can be represented by **digital signals**. The device that performs this function for voice data is a **codec** (coder-decoder)

Analog/Digital Conversions

- Data stored in the computer is in the form of 0’s and 1’s. To be carried from one place to the other, data is usually converted to digital signals.
  - This is called “**Digital-to-Digital Conversion**” or “**Encoding digital data into digital signals**”
- Sometimes we need to convert analog data to the digital signal, for example, conversion of telephone conversation to digital signal for different reasons, like to ***decrease effect of noise***
  - This is called “**Analog-to-Digital Conversion**” or “**Digitizing an Analog Signal**”

Analog/Digital Conversions

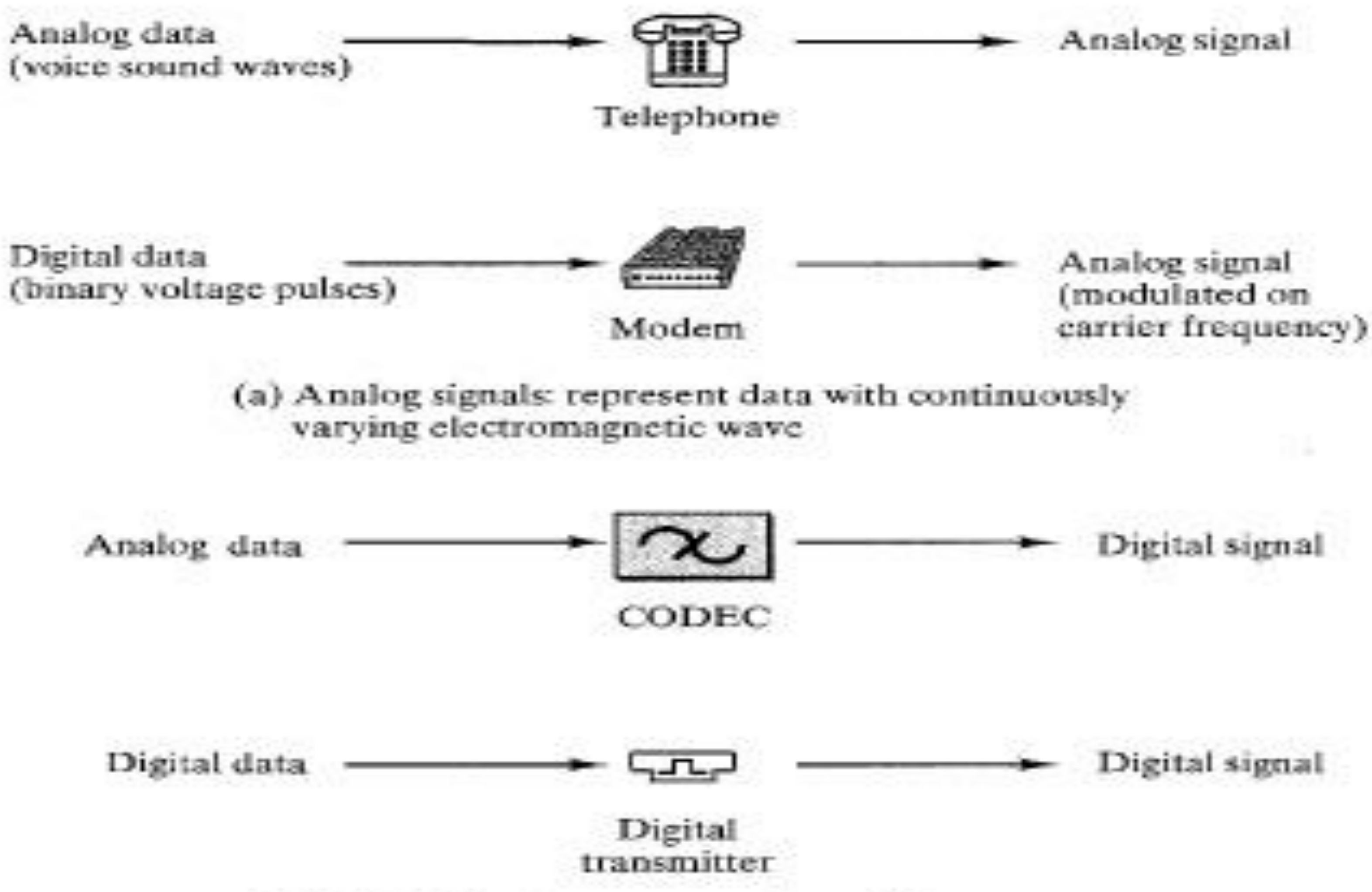
- We might want to send a digital signal coming out of computer through a medium designed for analog signals, for example, to send data from one place to the other using a telephone line.
  - This is called “**Digital-to-Analog Conversion**” or “**Modulating a digital Signal**”

Analog/Digital Conversions

- Often an analog signal is sent over long distances using analog media, for example, voice or music from a radio station which is an analog signal is transmitted through the air, however the frequency of voice or music is not, suitable for this kind of transmission.

- The signal should be carried by a higher frequency signal. This is called “**Analog-to-Analog Conversion**” or “**Modulating an analog Signal**”

Analog/Digital Conversions



(a) Analog signals: represent data with continuously varying electromagnetic wave

(b) Digital signals: represent data with sequence of voltage pulses

Types of Conversions

Digital to digital
Analog to digital
Digital to analog
Analog to analog

Digital-To-Digital Conversion

- Representation of digital data by using digital signals
- Techniques for digital-to-digital conversion
  - Line coding
  - Block coding
  - Scrambling

Digital-To-Digital Conversion

Line Coding: the process of converting digital data to digital signals

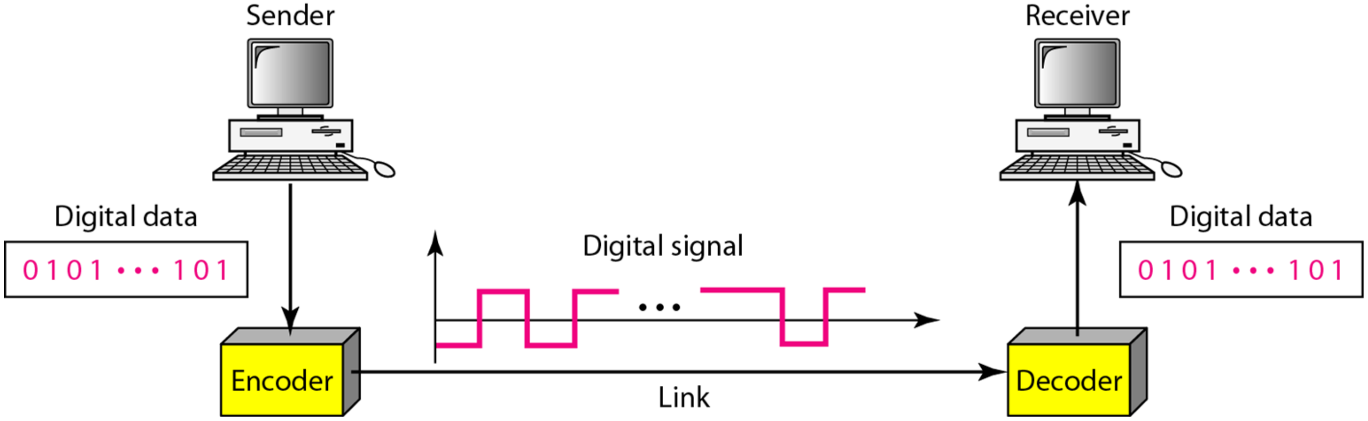
- **Assumption:** Data is stored in computer memory as sequence of bits
- Line coding converts sequence of bits to digital signals

- For example when we transmit data from computer to the printer, both original and transmitted data have to be digital
- Encoding a digital signal is where 1’s and 0’s generated by the computer are translated into voltage pulses that can be propagated over the wire

Digital-To-Digital Conversion

Line Coding

- **At sender:** encoding of digital data into digital signal
- **At receiver:** recreation of digital data by decoding the digital signal



Digital-To-Digital Conversion

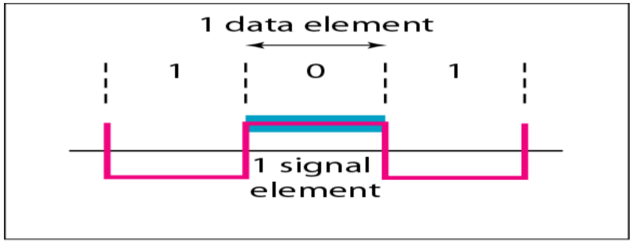
Characteristics of Line Coding Schemes

- **Signal element versus data element**
- **A data element** is the smallest entity (bit) that can represent a piece of information
  - In data communications, our goal is to send data elements which are what we need to send
  - Data elements are being carried
- **A signal element** is the shortest unit (time wise) of a digital signal
  - In digital data communications, a signal element carries data elements (carriers)
  - So signals elements are what we can send
  - **Ratio** of data elements to signals elements

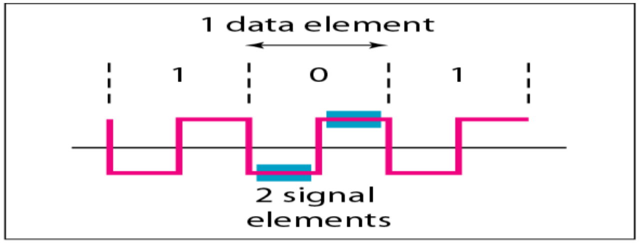
Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

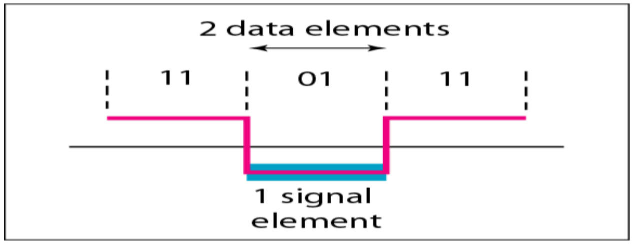
- **Signal element versus data element**
- In the simplest case, there is a one-to-one correspondence between bits and signal elements



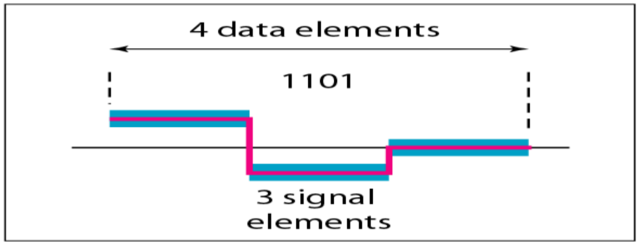
a. One data element per one signal element (r = 1)



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

Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

- **Data rate versus signal rate**
- The **data rate** (bit rate) is the number of data elements (bits) sent in **I second**
  - The unit is bits per second (bps)
- The **signal rate** (baud rate, modulation rate or pulse rate) is the number of signal elements sent in **I second**.
  - The unit is the baud
- **Data communications goal:** increase the data rate (increasing speed of transmission) and decrease the signal rate (decreasing bandwidth requirement)

Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

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 bandwidth reflects the range of frequencies we need

Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

Baseline wandering

- A long string of 0s or 1s can cause a drift in the baseline (**baseline wandering**) and make it difficult for the receiver to decode correctly
- A good line coding scheme needs to prevent baseline wandering

Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

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
- Consider a situation where the receiver has a shorter bit duration
  - Let the sender sends 10110001, while the receiver receives 110111000011

Digital-To-Digital Conversion

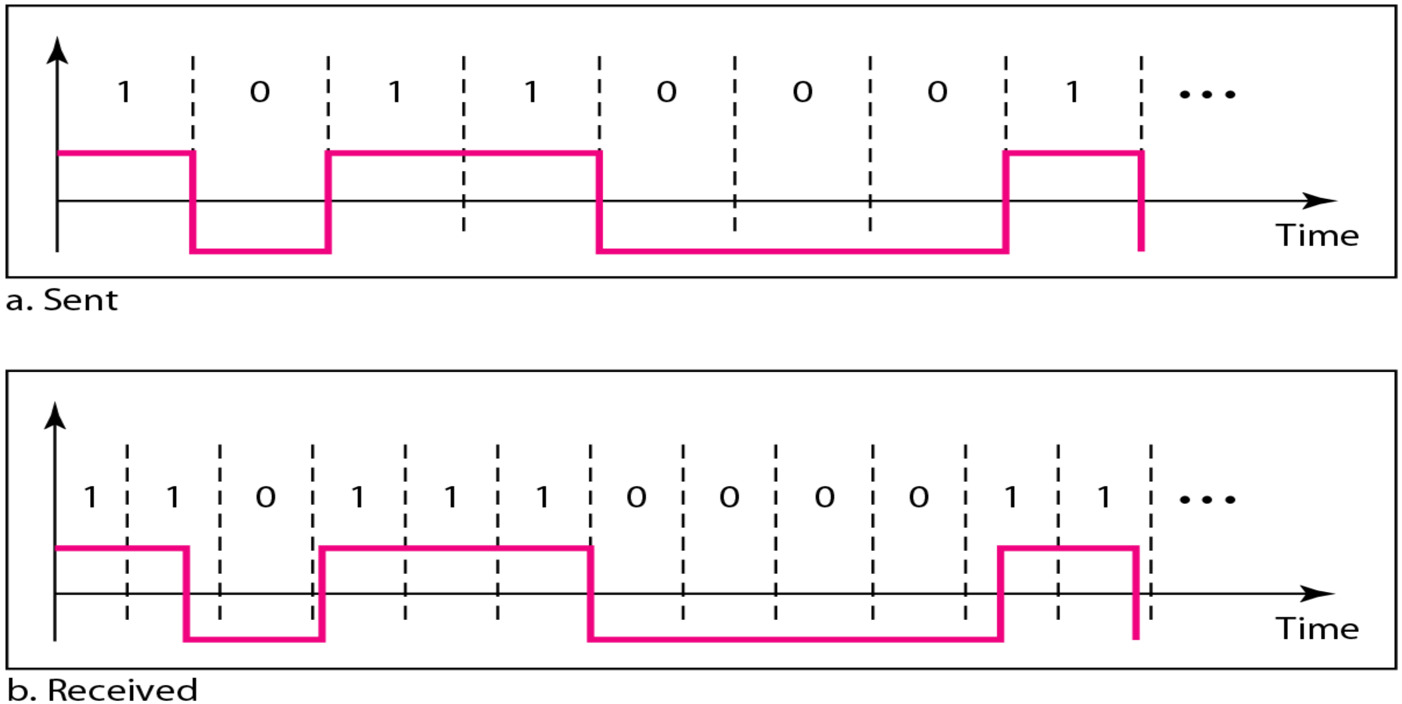
Characteristics of Line Coding Schemes

Self synchronization

- A self-synchronizing digital signal includes timing information in the data being transmitted
  - The same can be achieved by adding transitions in the signal that alert the receiver to the start, middle or end of the pulse.
  - These points can reset the clock in case the receiver's clock is out of synchronization

Effect of lack of synchronization

- Consider a situation where the receiver has a shorter bit duration
- Let the sender sends 10110001, while the receiver receives 110111000011



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

Digital-To-Digital Conversion

Characteristics of Line Coding Schemes

Built-in error detection

- Desirable to have a built-in error-detecting capability in the generated code to detect some of or all the errors that occurred during transmission

Immunity to noise and interference

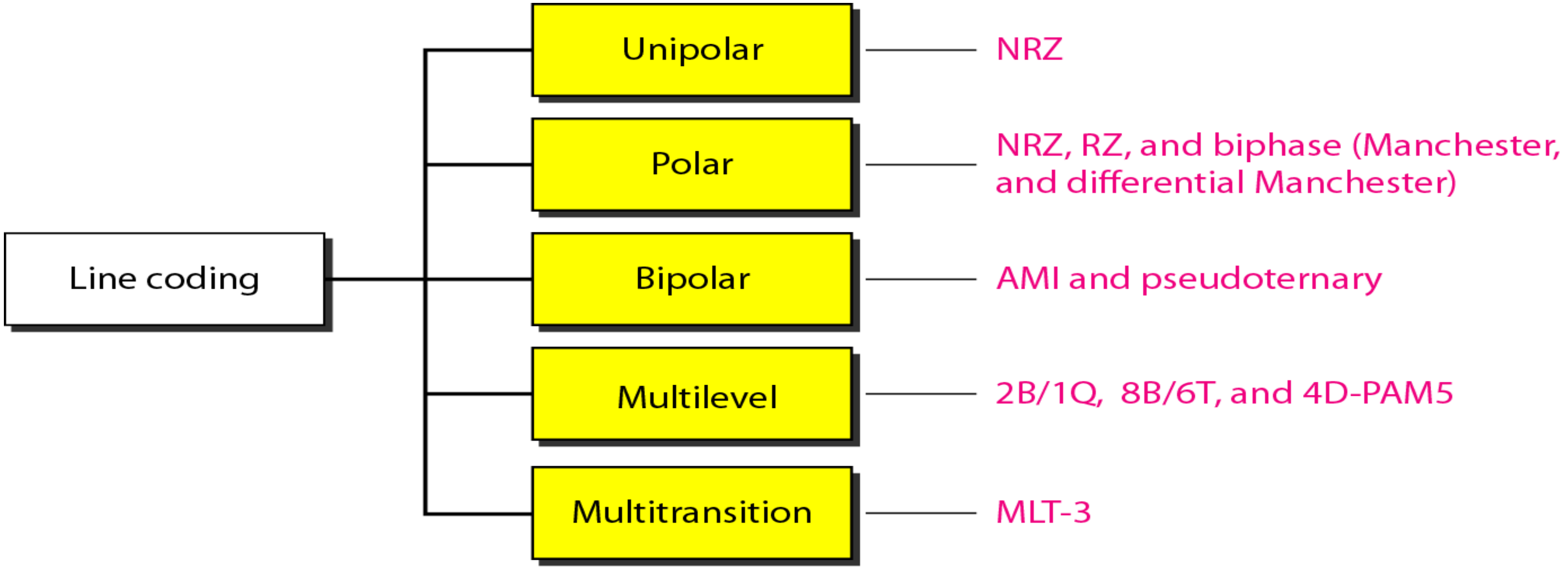
- Desirable to have a code that is immune to noise and other interferences

Complexity

- A complex scheme is more costly to implement than a simple one
- **Example**: a scheme that uses four signal levels is more difficult to interpret than one that uses only two levels

Digital-To-Digital Conversion

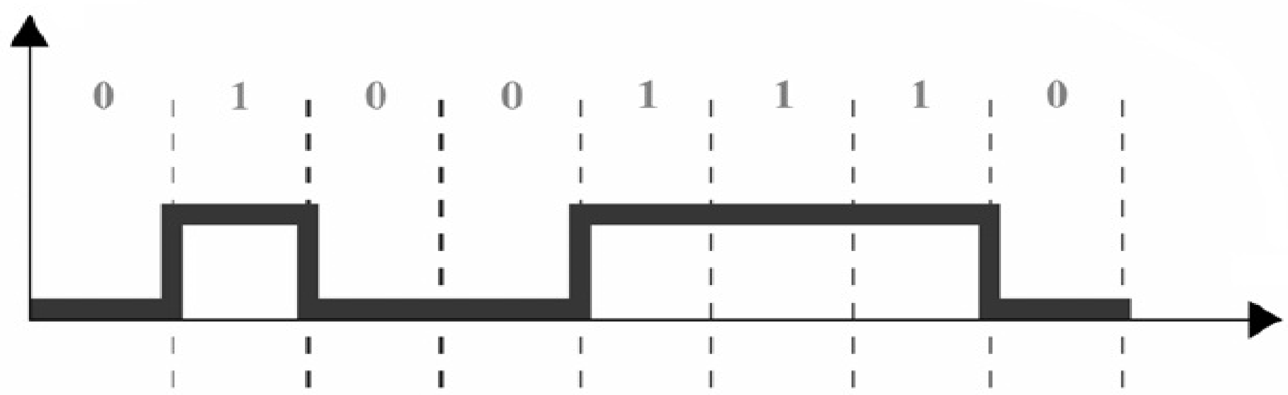
Line Coding Schemes (types of encoding)



Line Coding Schemes: Unipolar

Unipolar: non-return-to-zero (NRZ) [no return in middle**]**

- Encoding is simple , with only one technique in use
- all the signal levels are **on one side of the time axis**, either above or below
- It is called **Unipolar** because it uses only one **polarity**
- positive voltage defines bit 1 and the zero voltage defines bit 0



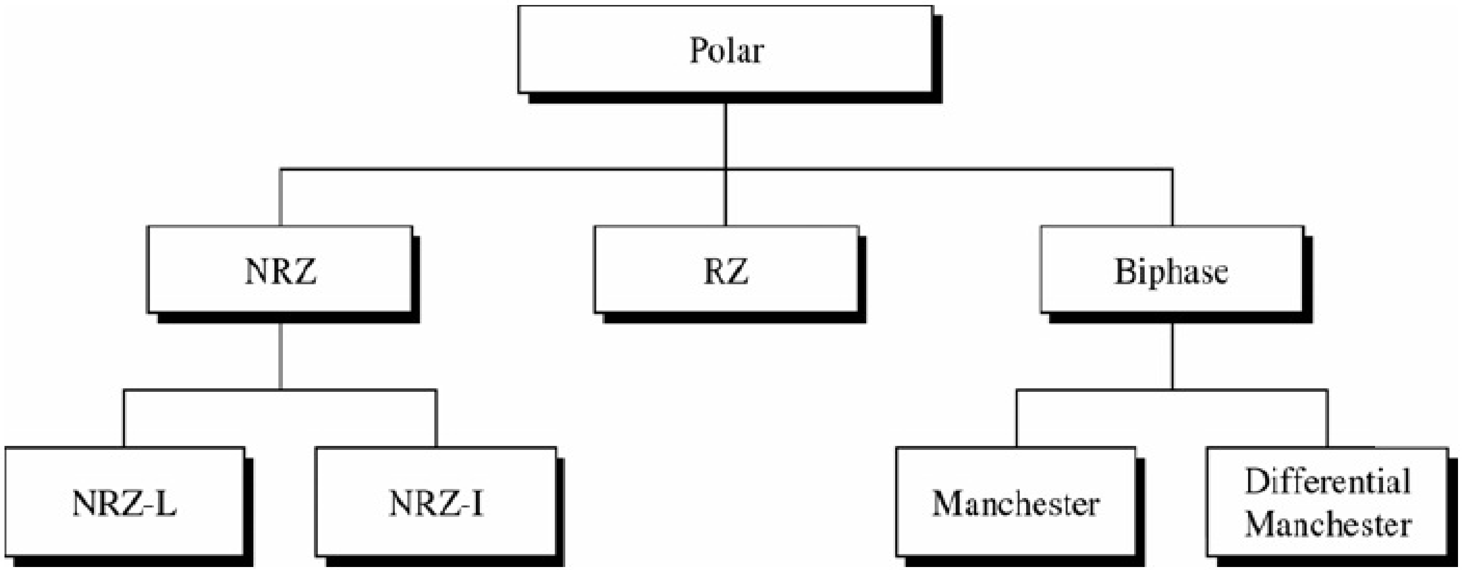
Practice Example

- Convert the following data elements (bit stream) into digital signals using Unipolar Scheme
- 11001101

Line Coding Schemes: Polar

Polar

- Polar encoding uses two voltage levels, positive and negative (on both sides of time axis)
  - The voltage level for 0 bit can be positive (+ve) and the voltage level for 1 can be negative
- 3 subcategories: NRZ, RZ, Biphase

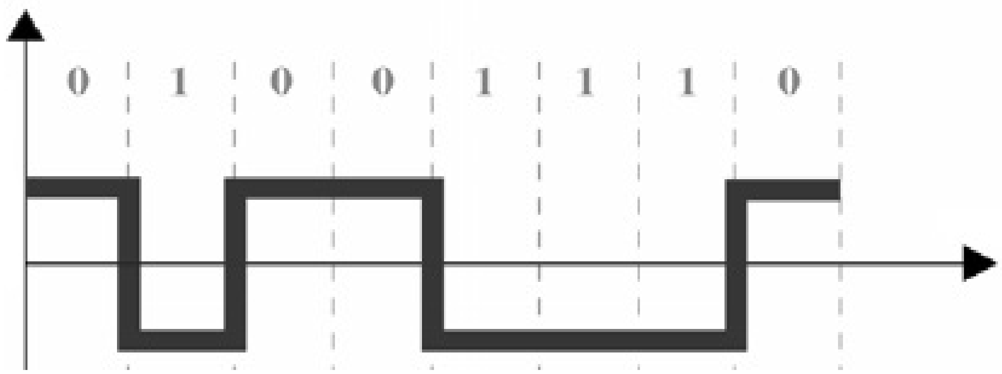


Line Coding Schemes: Polar

Polar: NRZ

Non-return-to-zero-level (NRZ-L)

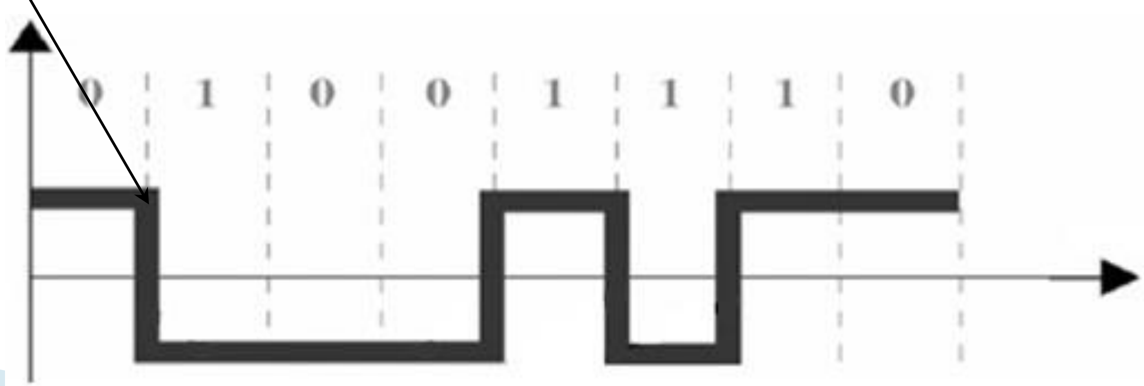
- A +ve voltage means the bit is a 0 and a –ve voltage means the bit is a 1 (vice versa may also be true)



Line Coding Schemes: Polar

Polar: NRZ : Non-return-to-zero-Invert (NRZ-I)

- 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
- A transition (low-to-high or high-to-low) at the beginning of a bit time denotes a binary 1 for that bit time; no transition indicates a binary 0



Practice Example

- Convert the following data elements (bit stream) into digital signals using NRZ-I Scheme
- 10101100

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.

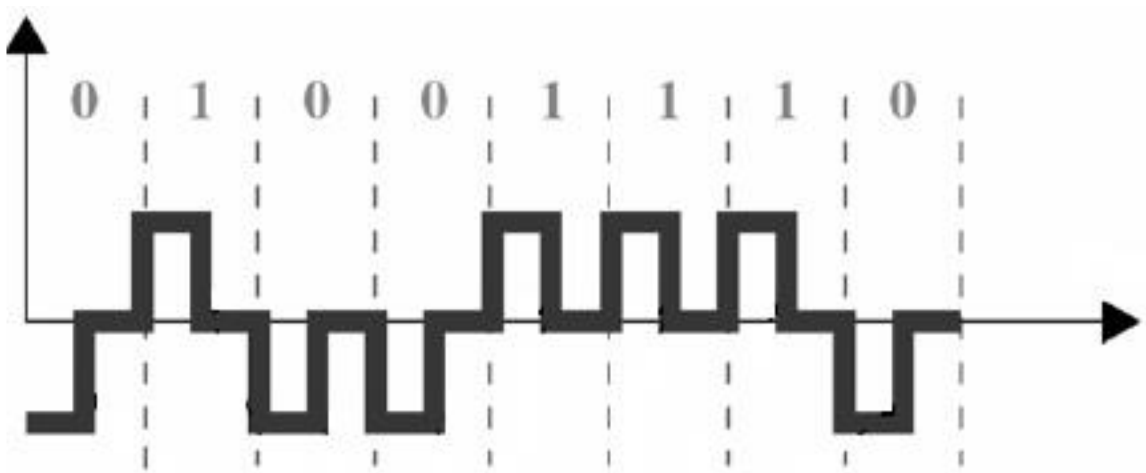
Line Coding Schemes: Polar

Polar: Return-to-zero (RZ)

- Any time, data contains long strings of 1’s or 0’s, receiver can loose its timing (synchronization problem of NRZ)
- One solution is **RZ** encoding which uses 3 values; Positive, Negative and Zero
- Signal changes not between bits but during each bit

Line Coding Schemes

- Here, +ve voltage means 1 and a –ve voltage means 0,
- but unlike NRZ-L, half way through each bit interval, the signal returns to zero
- A 1 bit is represented by positive to zero and a 0 is represented by negative to zero transition



Practice Example

- Convert the following data elements (bit stream) into digital signals using Polar RZ Scheme
- 10101100

Line Coding Schemes

Problem with RZ

- Main problem with RZ encoding is that it requires two signal changes to encode one bit
  - therefore occupies more bandwidth
  - But it is most effective solution so that receiver cannot lose timing (no DC component problem)
- Complexity: uses three levels of voltage

Line Coding Schemes: Polar

<u>Biphase</u>

- Best existing solution to the problem of synchronization
- **Signal changes at the middle of bit interval but does not stop at zero**
  - Instead it continues to the opposite pole

There are two types of biphase encoding

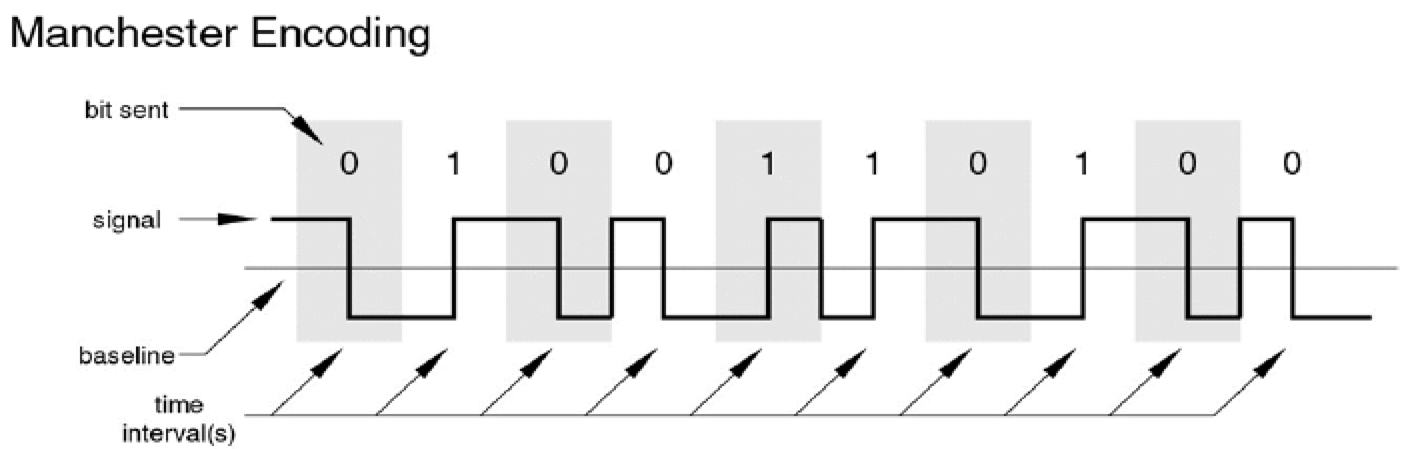
1. **Manchester**

1. **Differential Manchester**

Line Coding Schemes: Polar

Biphase: Manchester

- ➢Transition in middle of each bit period
- ➢Transition serves as clock and data
- ➢Used by IEEE 802.3 (10Mbps Ethernet)
- ➢ Negative-to-Positive Transition (low to high) = 1
- ➢ Positive-to-Negative Transition (high to low) = 0



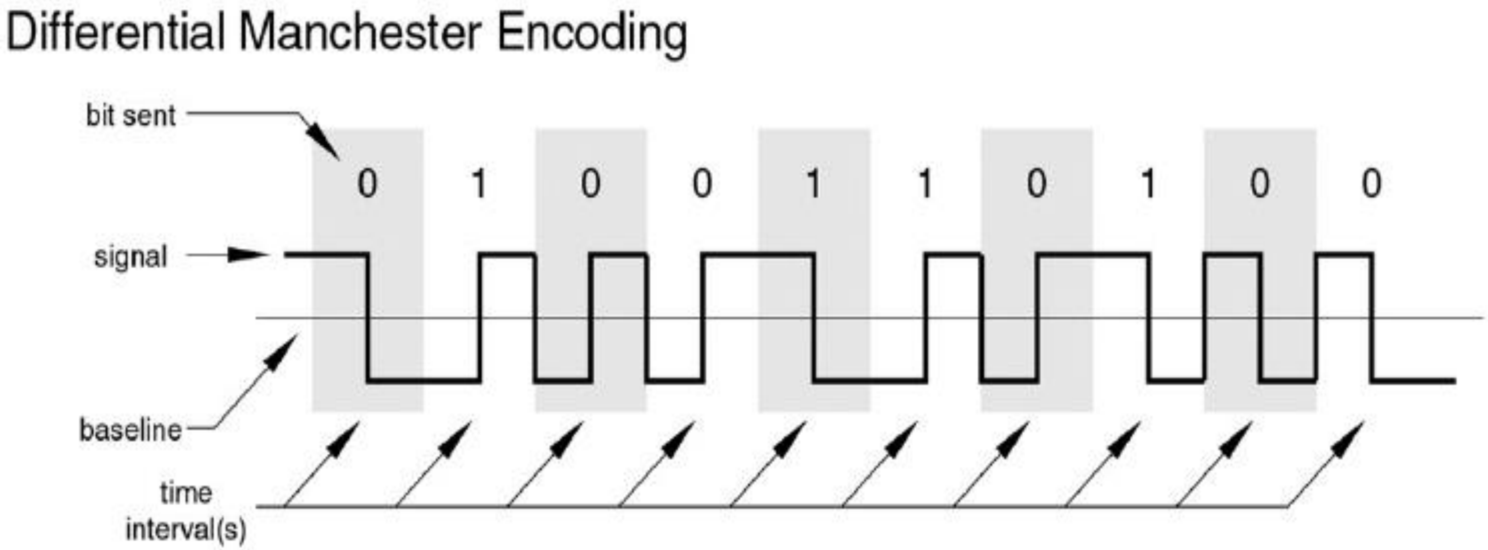
Practice Example

- Convert the following data elements (bit stream) into digital signals using Biphase Manchester encoding Scheme
- 10101100

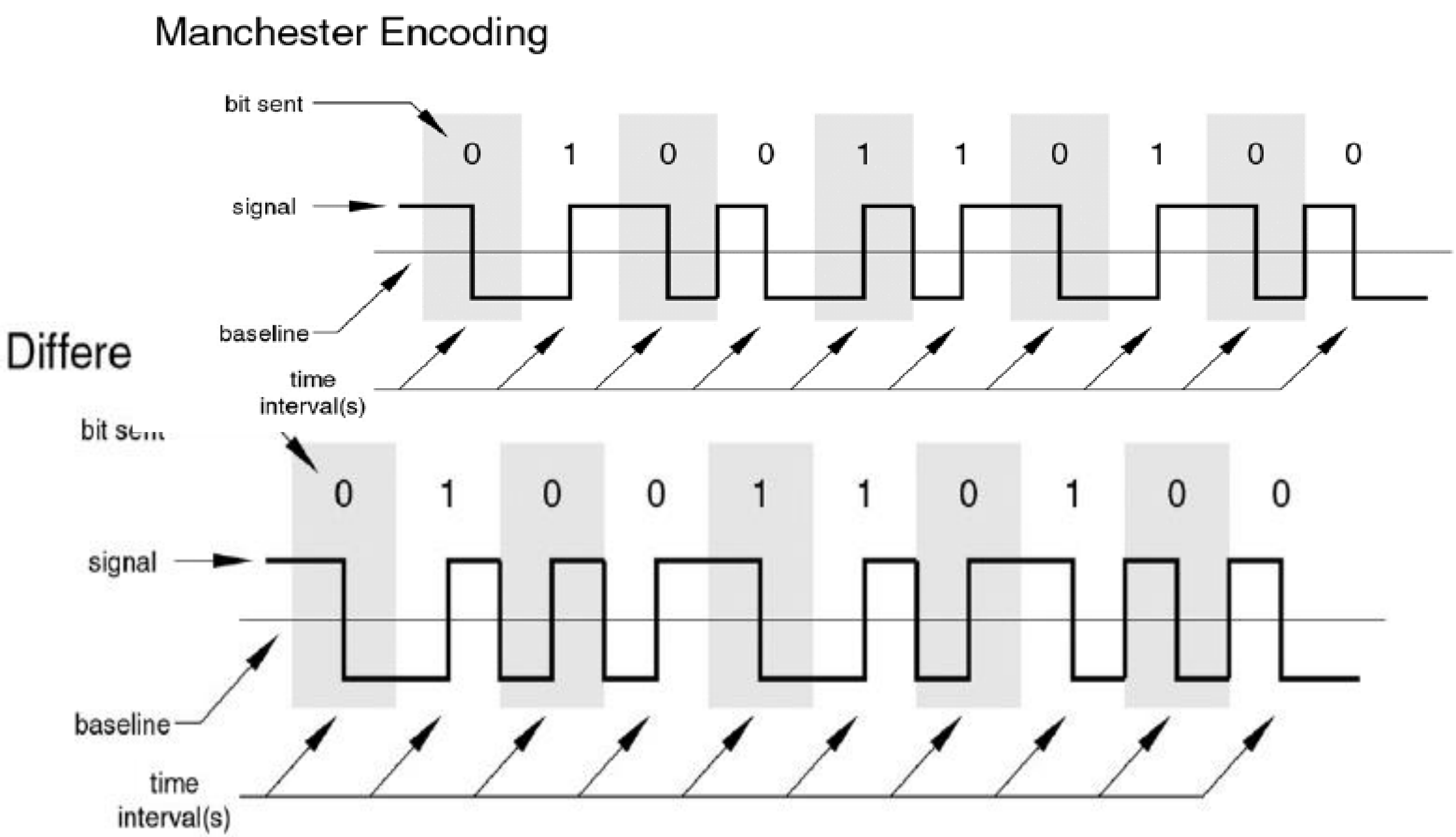
Line Coding Schemes: Polar

Biphase: Differential Manchester

- ➢ Midbit transition is clocking only (for synchronization)
- Transition at start of a bit period represents zero
- No transition at start of a bit period represents one
- Note: this is a differential encoding scheme
- Used by IEEE 802.5 (Token ring)



Line Coding Schemes: Polar



Practice Example

- Convert the following data elements (bit stream) into digital signals using Biphase Differential Manchester encoding Scheme
- 10101100

Line Coding Schemes: Bipolar

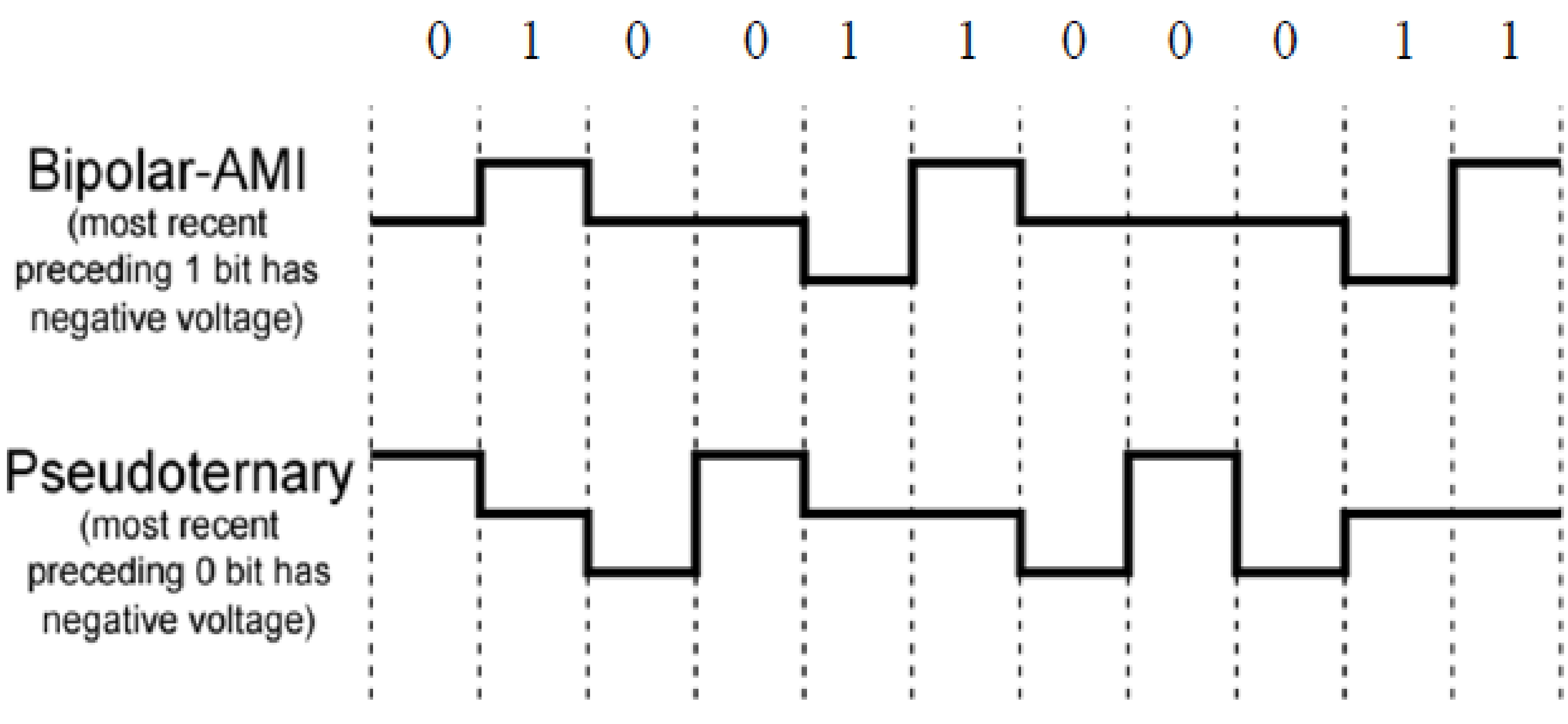
Bipolar encoding

- There are three voltage levels; +ve, -ve, zero 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 +ve & -ve.
- **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: the “1” symbol is represented by zero voltage and the “0” symbol alternates between +V and -V

Line Coding Schemes: Bipolar

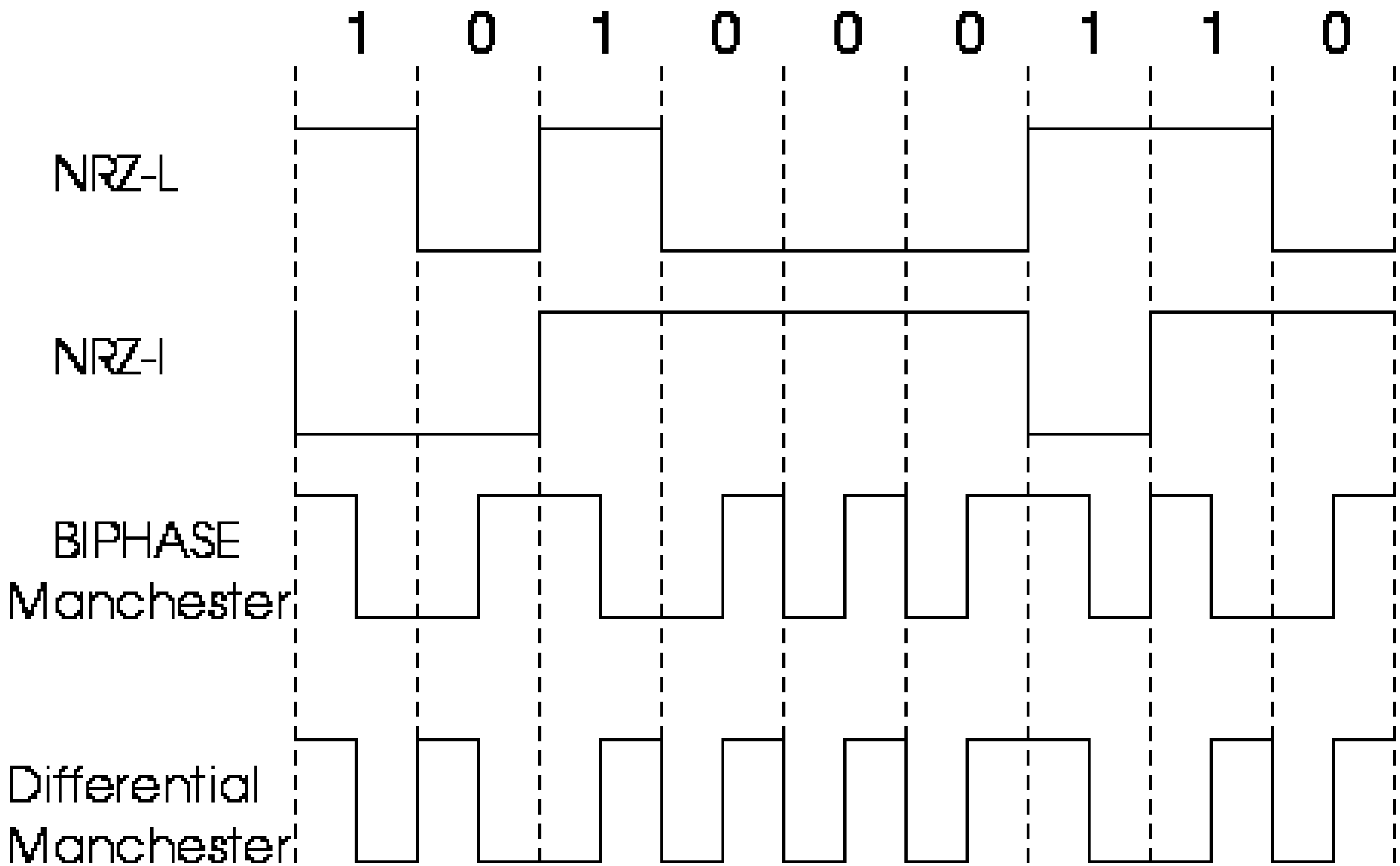
<u>Bipolar encoding</u>

Bipolar Alternate Mark Inversion (**AMI**) and **Pseudoternary**



Practice Example

- Convert the following data elements (bit stream) into digital signals using AMI and **Pseudoternary** Scheme
- 10101100


1 0 1 0 0 0 1 1 0
NRZ-L
NRZ-I
BIPHASE Manchester
Differential Manchester

Alternate Mark Inversion(AMI)

- Pros and Cons:

- There will be no loss of synchronization if a long string of 1s occurs
- Each 1 introduces a transition, and the receiver can resynchronize on that transition
- A long string of 0s would still be a problem
- No error detection