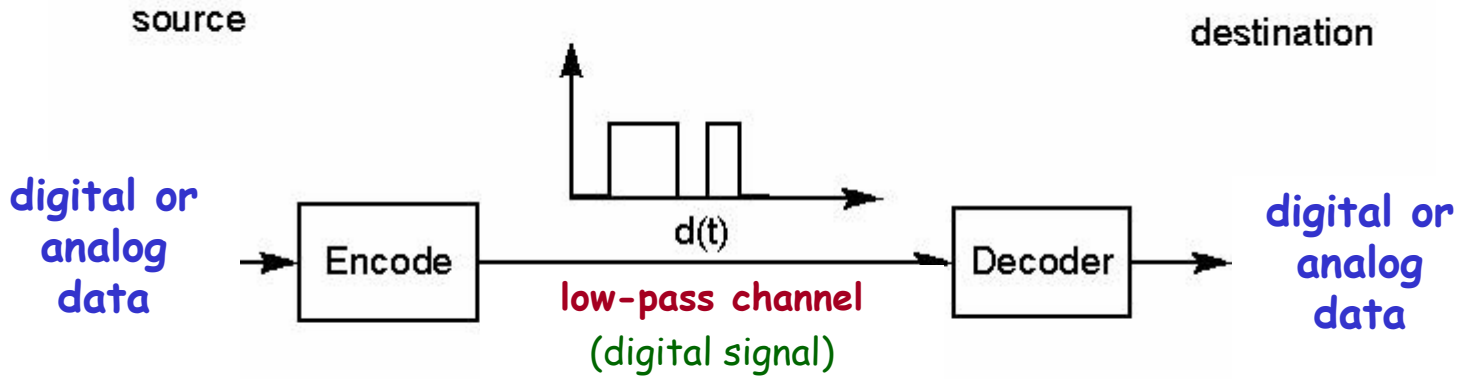
Digital Transmission of Digital Data:

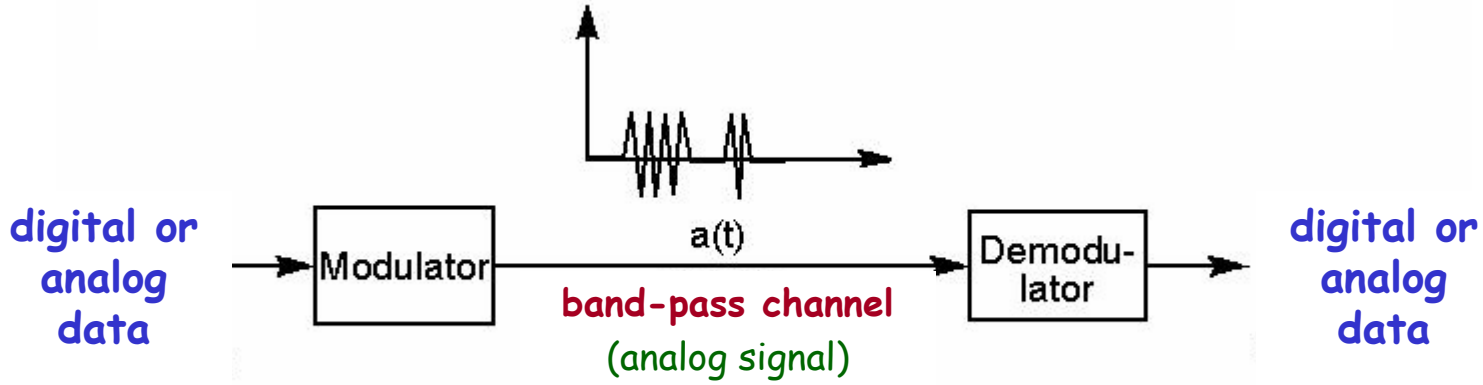
Line and Block Coding, Digital Transmission Modes

Required reading:
Forouzan 4.1
Garcia 3.6

CSE 3213, Fall 2015
Instructor: N. Vlajic

Analog vs. Digital Transmission (cont.)

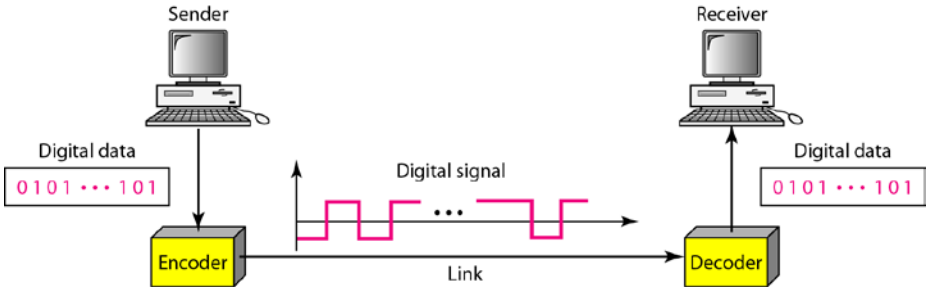




Line Coding: Design Consideration

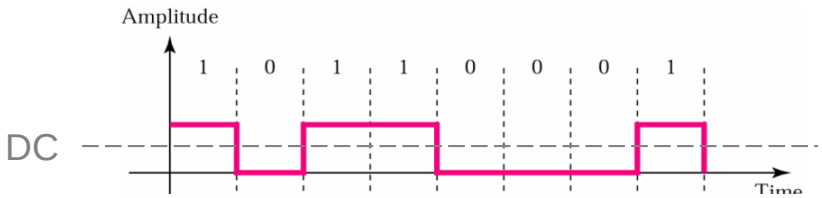
Line Coding – process of converting binary data (sequence of bits) to a digital signal

- digital signal depends 'linearly' on information bits, i.e. bits are transmitted 'one-by-one' – different from block coding

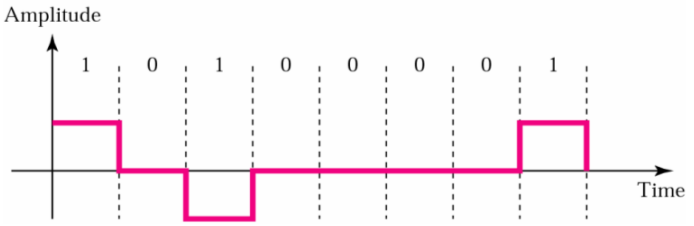


Data Level vs. Signal Level

- **data levels** – number of values / levels used to represent data (typically only two: 0 & 1)
- **signal levels** – number of values / levels allowed in a particular signal



Two signal levels, two data levels.



Three signal levels, two data levels.

Line Coding: Design Consideration (cont.)

DC Component in Line Coding

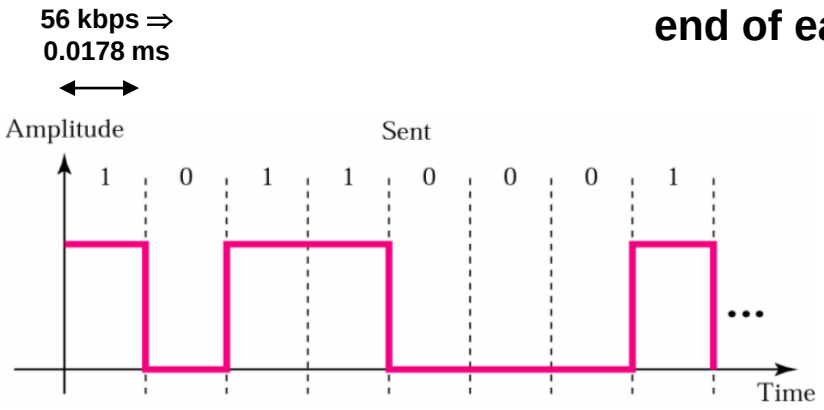
– some line coding schemes have a residual (DC) component, which is generally undesirable

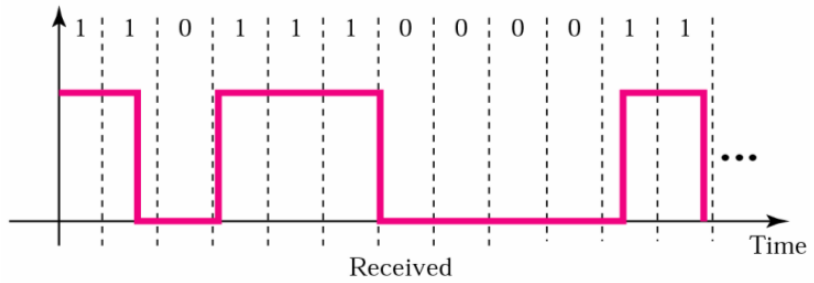
- transformers do not allow passage of DC component (used when switching from one medium to another)
- DC component ⇒ extra energy – useless!

Self-Synchronization (Clocking)

– to correctly interpret signal received from sender receiver's bit interval must correspond exactly to sender's bit intervals

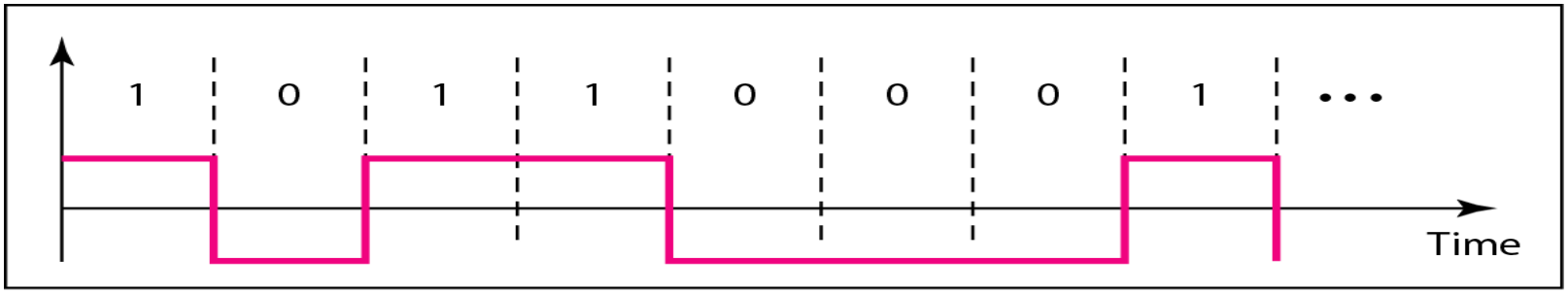
- if receiver clock is faster/slower, bit intervals are not matched ⇒ receiver might misinterpret signal
- self-synchronizing digital signals include timing information in itself, to indicate the beginning and end of each pulse (see pp. 8-10)



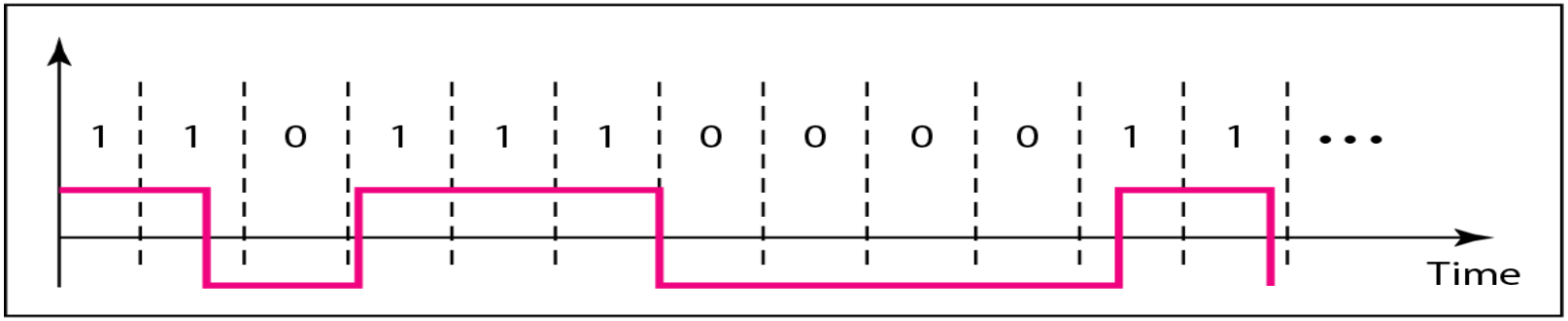


Line Coding: Design Consideration (cont.)

Example: effect of lack of synchronization



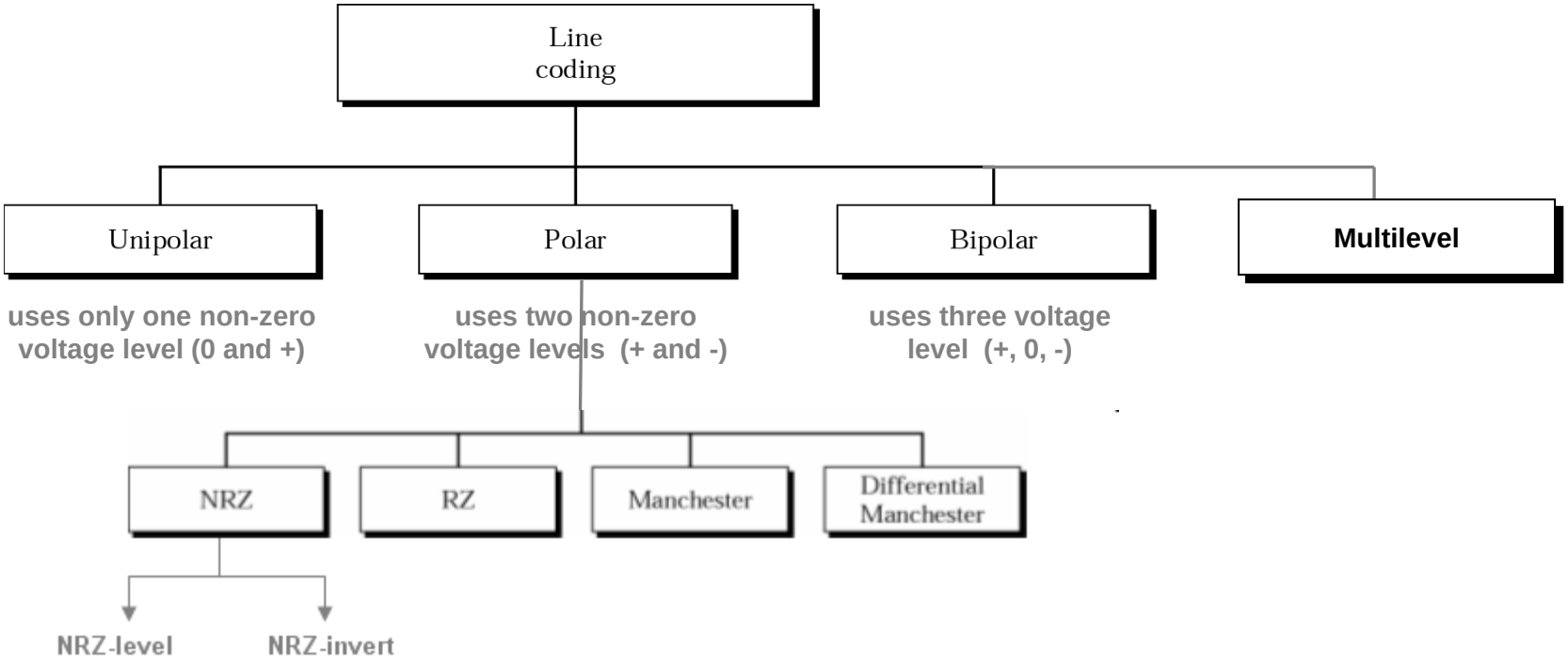
a. Sent



b. Received

Line Coding: Design Consideration (cont.)

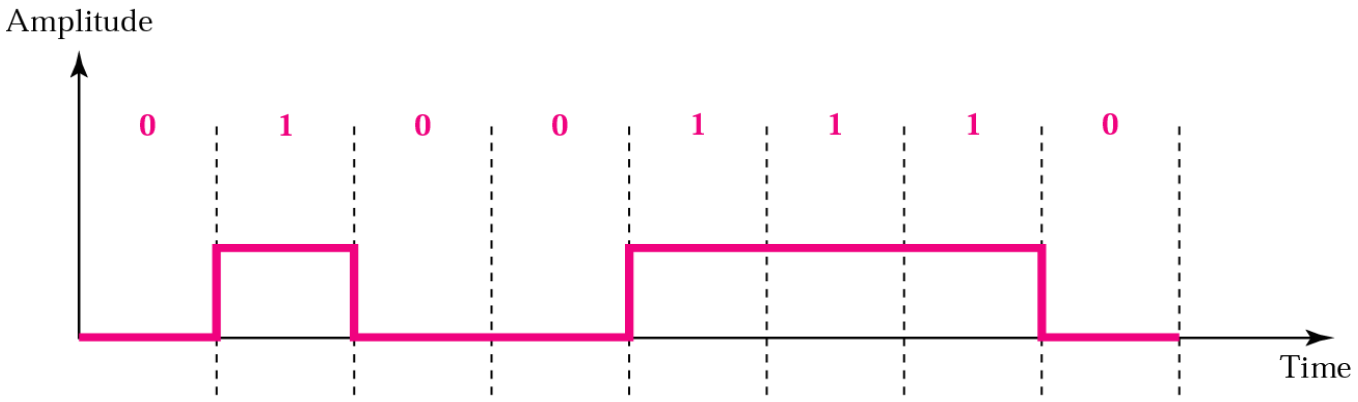
Line Coding Schemes – **can be divided into three broad categories**



Line Coding: Unipolar

Unipolar Line Coding – **uses only <u>one non-zero</u> and one zero voltage level**

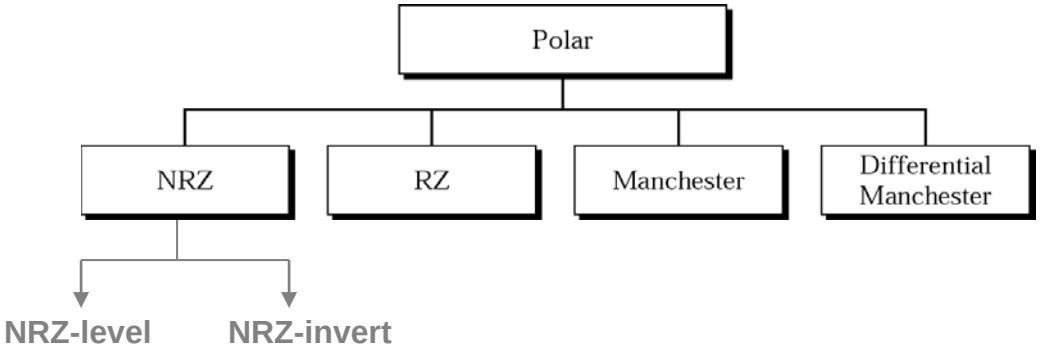
- **(e.g.) 0 = zero level, 1 = non-zero level**
- **simple to implement, but obsolete due to two main problems:**
  - **DC component present ☹**
  - **lack of synchronization for long series of 1-s or 0-s ☹**



Line Coding: Polar

Polar Line Coding – uses two non-zero voltage level for representation of two data levels – one positive and one negative

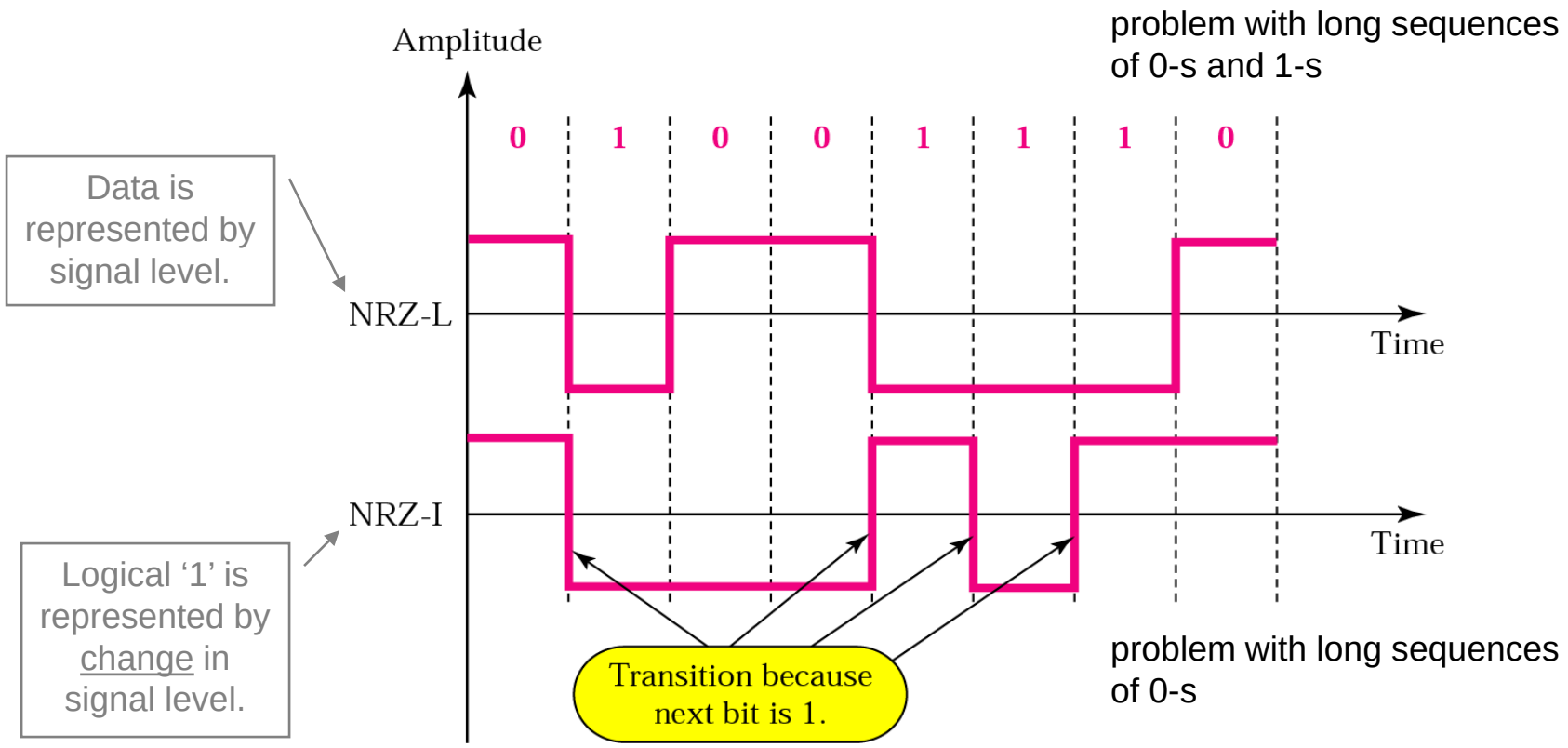
- "DC-problem" alleviated, mostly ☺
- 4 main types of polar coding:



(1) Nonreturn to Zero (NRZ)

- NRZ-level: signal level represents particular bit, (e.g.) 0 = positive volt. , 1 = negative volt.
  - lack of synchronization for long series of 1-s & 0-s ☹
- NRZ-invert: inversion of voltage level represents bit 1, no voltage change represents bit 0
  - 1s in data streams enable synchronization
  - long sequence of 0-s still a problem ☹ →

Line Coding: Polar (cont.)

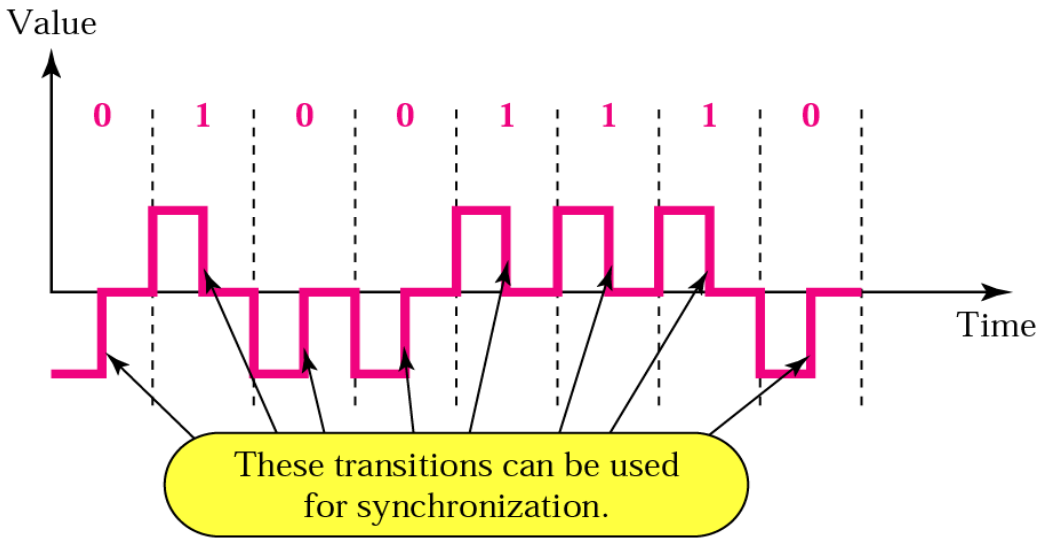


NRZ-I is better than NRZ-L, but it still does not provide complete synchronization.
To ensure complete synchronization, there must be a signal change for each bit.

Line Coding: Polar (cont.)

(2) Return to Zero (RZ) – **(e.g.) 0 = negative volt., 1 = positive volt., AND signal must return to zero halfway through each bit interval**

- **perfect synchronization** ☺
- **drawback – 2 signal changes to encode each bit ⇒ pulse rate is x2 rate of NRZ coding, i.e. more bandwidth is required, regardless of bit sequence** ☹
- **more complex to implement – 3 sig. lev. required**



Non-zero level ⇒ beginning of a new bit.