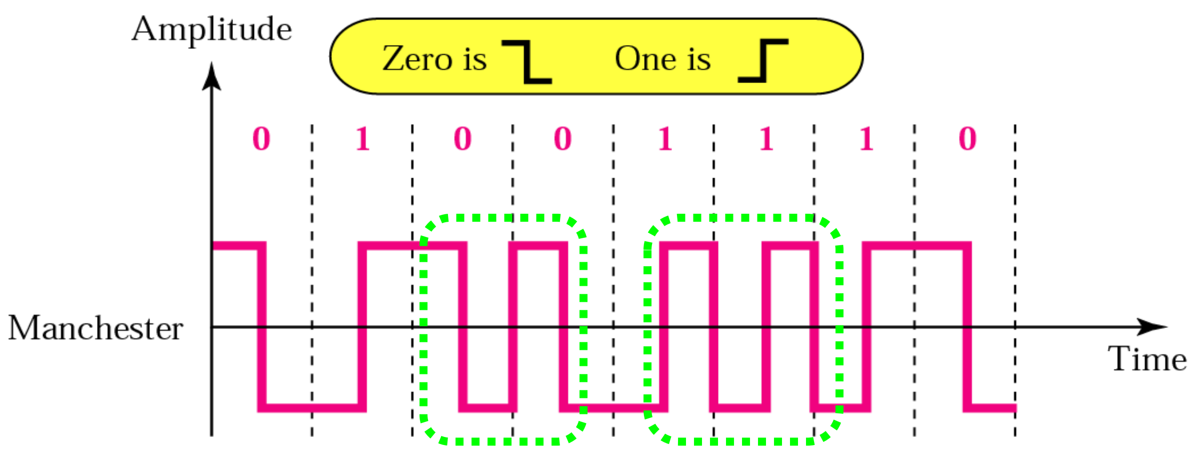
Line Coding: Polar (cont.)

(3) Manchester – **inversion at the middle of each bit interval is used for both synchronization and bit representation**

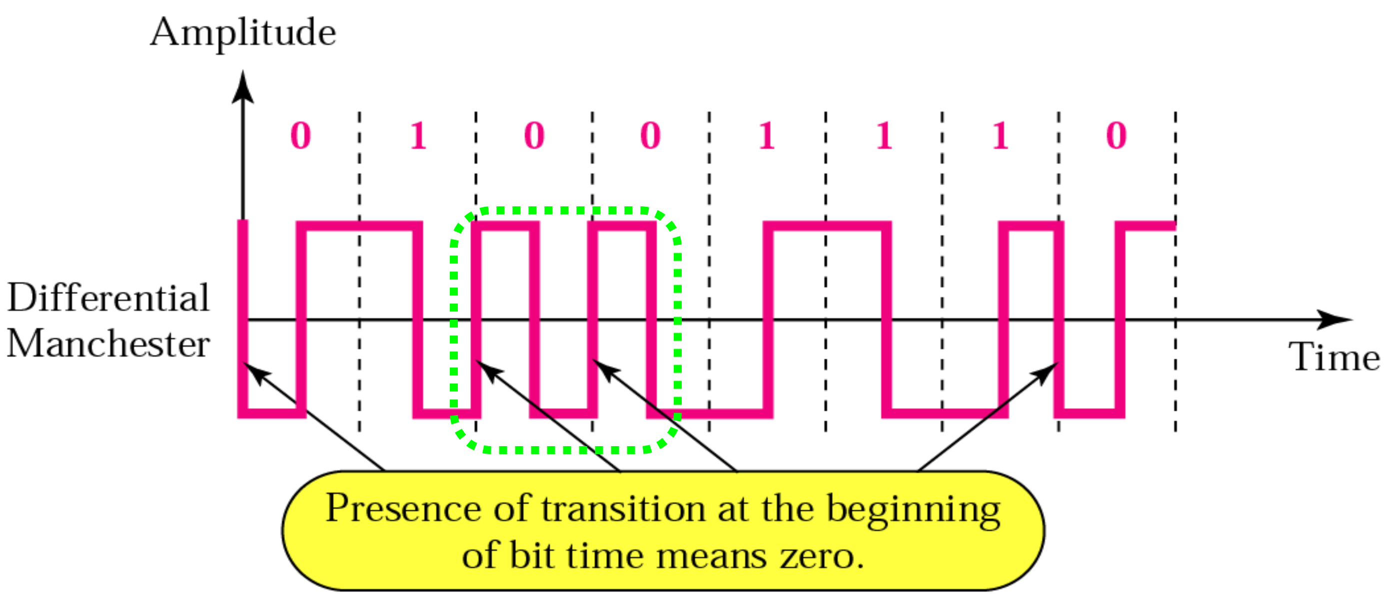
- **0 = pos-to-neg transition, 1 = neg-to-pos transition**
- **there is always transition at the middle of the bit, and maybe one transition at the end of each bit**
- **perfect synchronization** ☺
- **fine for alternating sequences of bits (10101), but wastes bandwidth for long runs of 1-s or 0-s** ☹
- **used by IEEE 802.3 (Ethernet)**



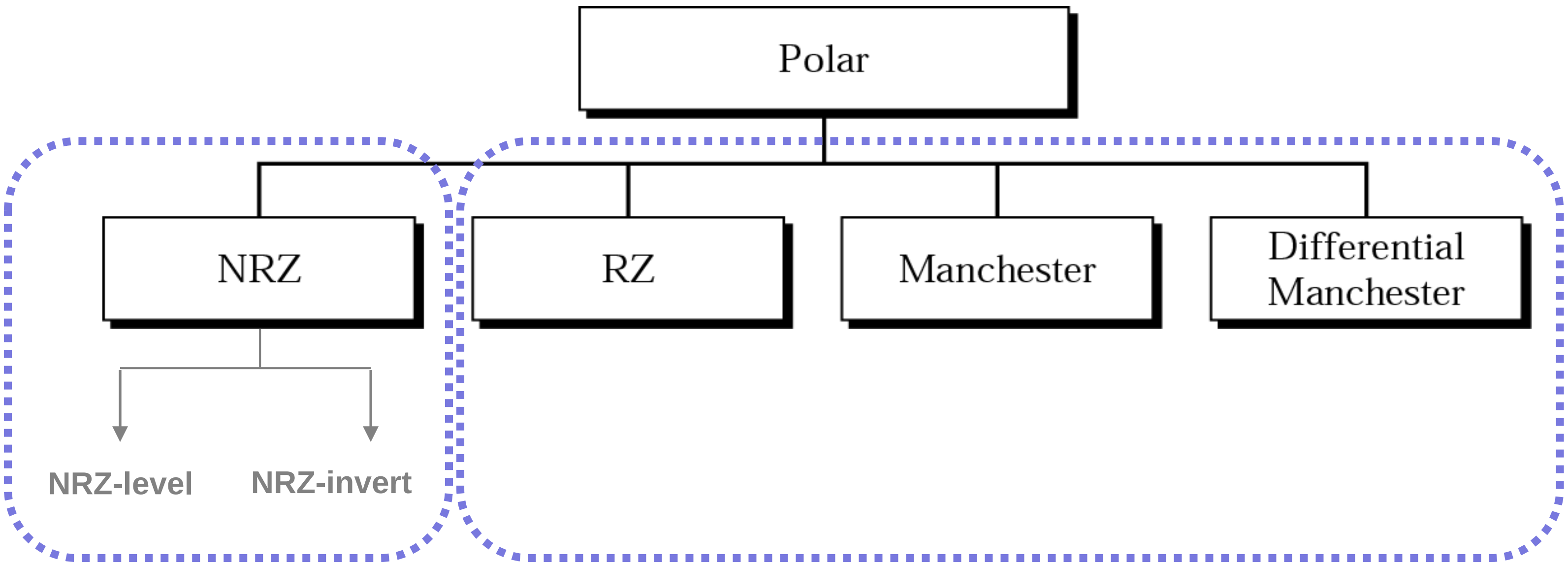
Line Coding: Polar (cont.)

(4) Differential Manchester – **inversion in the middle of bit interval is used for synchronization – presence or absence of additional transition at the beginning of next bit interval identifies the bit**

- **0 = transition, 1 = no transition**
- **perfect synchronization** ☺
- **fine for long runs of 1s, but wastes bandwidth for long runs of 0-s** ☹
- **used by IEEE 802.5 (Token Ring)**



Line Coding: Polar (cont.)



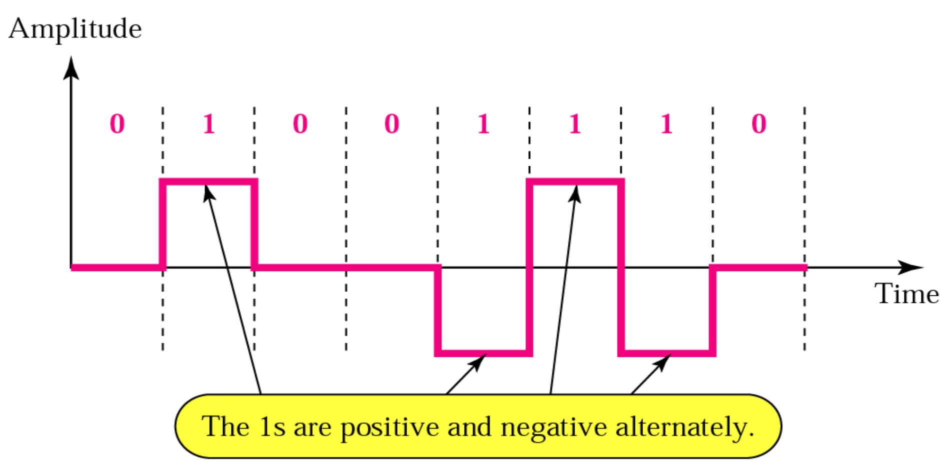
Good for bandwidth, bad for synchronization.

Good for synchronization, bad for bandwidth.

Line Coding: Bipolar

Bipolar Line Coding – aka Alternate Mark Inversion (AMI) - uses two non-zero and zero voltage level for representation of two data levels

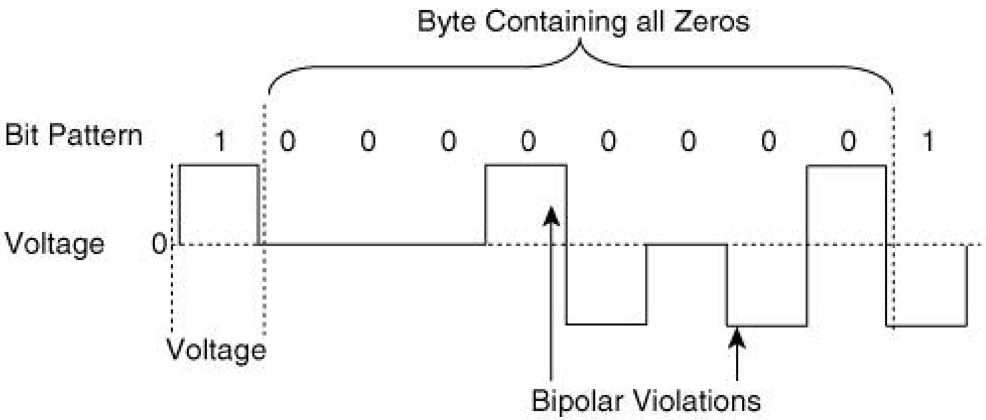
- 0 = zero level; 1 = alternating pos and neg level
- e.g. if 1st 'bit 1' is represented by positive amplitude, the 2nd will be represented by negative amplitude, the 3rd by positive, etc.
- less bandwidth required than with Manchester coding (for any sequence of bits) ☺
- loss of synchronization is possible for long runs of 0-s ☹



Line Coding: Bipolar (cont.)

Scrambling – **solution to 'long sequence of 0s' problem**

- **sequences that would result in a constant voltage level are replaced by sequences that provide sufficient number of transitions to ensure synchronization**
- **B8ZS – version of AMI: 8 consecutive 0s (when it occurs) is replaced by 000+-0-+ [used in T1 lines]**
- **perfect synchronization while the overall number of bits remains the same ☺**



B8ZS signifies a byte containing all 0s by creating two bipolar violations (i.e., two consecutive voltages of the same polarity) in the 4th and 7th bit positions.

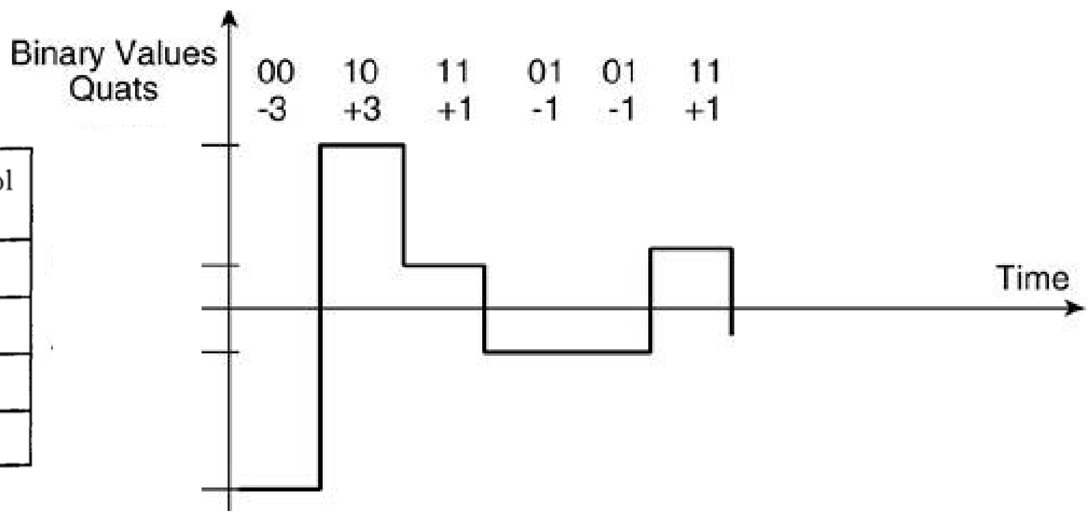
Line Coding: Multilevel

2B1Q (2 Binary 1 Quaternary) Coding – **data patterns of size 2 bits are encoded as one signal element belonging to a four-level signal**

- **data is sent 2X faster than with single-level**
- **receiver has to discern 4 different thresholds; no self-synchronizations for long double bits**
- **used in DSL lines**

First Bit (Sign)	Second Bit (Magnitudes)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1
0	1	-1
0	0	-3

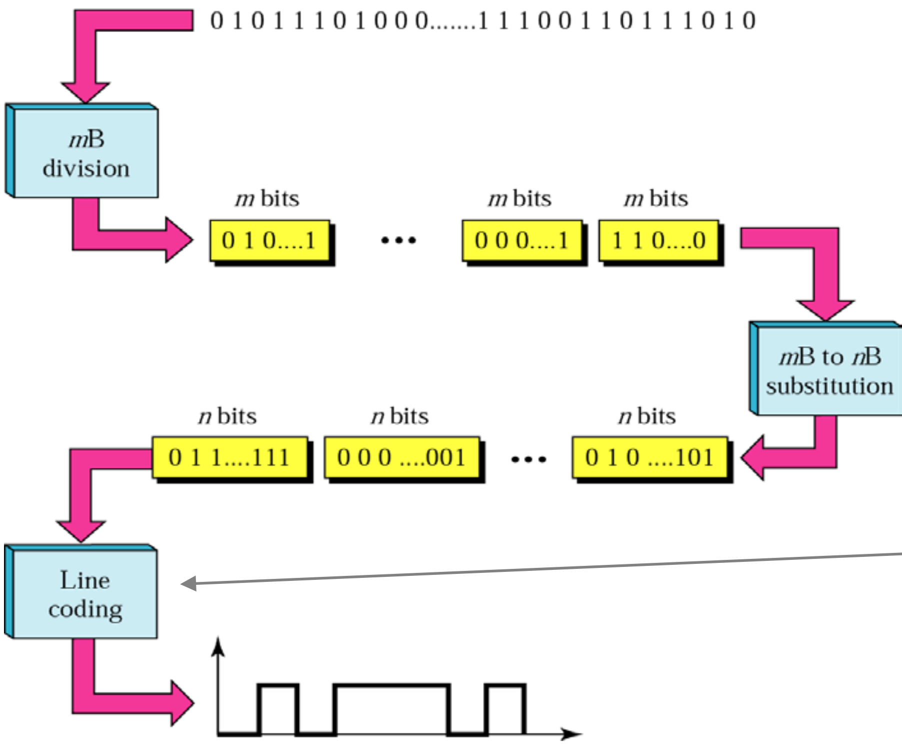
TABLE 40 - 2B1Q Symbol Coding



Block Coding

Block Coding – **unlike line codes which operate on a stream of information bits, block codes operate on block of information bits**

- **number of 'redundant bit(s)' are added to each block of information bits to ensure synchronization and error detection!**



Example:
substitute m information bits with n (new) well-chosen bits that do not have more than three consecutive 0s and 1s
(easy to synchronize with any of the line coding schemes)

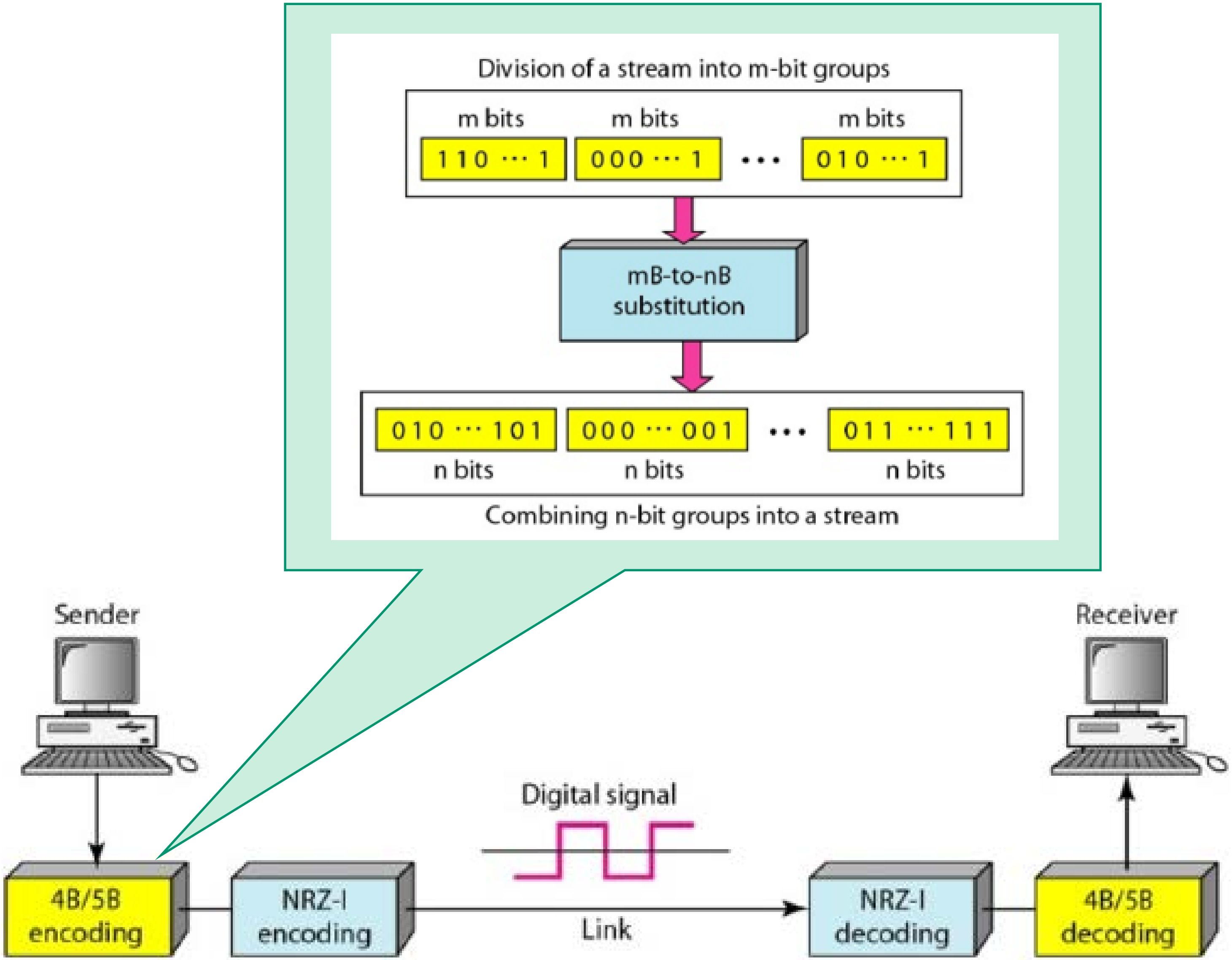
If one or more bits is changed resulting in one of the unused 5-bit blocks ⇒ receiver can easily detect the error.

Drawback 1: **longer encoding / decoding time**

Drawback 2: **lower effective bit-rate**

Block Coding (cont.)

Example [4B/5B block code]



Block Coding (cont.)

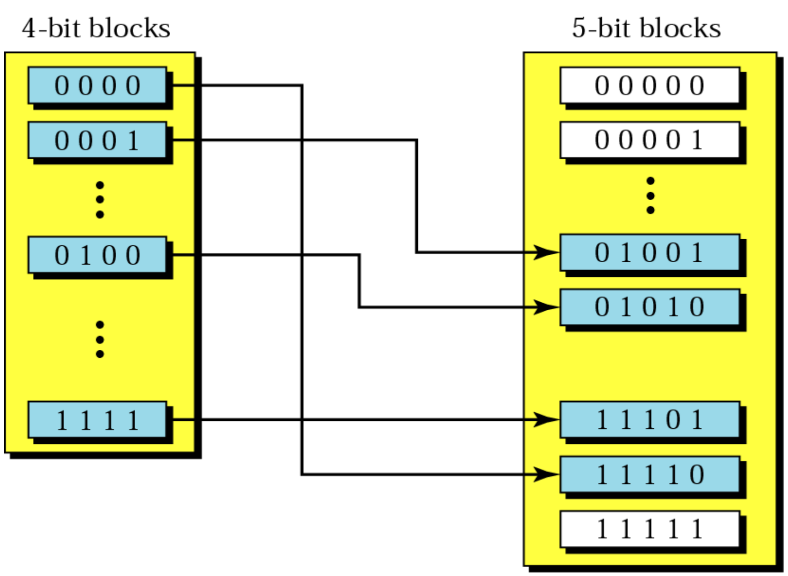
Example [4B/5B block code]

Every 4 bits of data is encoded into a 5-bit code.

The 5-bit codes are normally line coded using NRZ-invert (longer sequences of 1 are tolerated)!!!

The 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.

Therefore, when these 5-bit codes are sent in sequence, no more than three consecutive 0s are encountered.



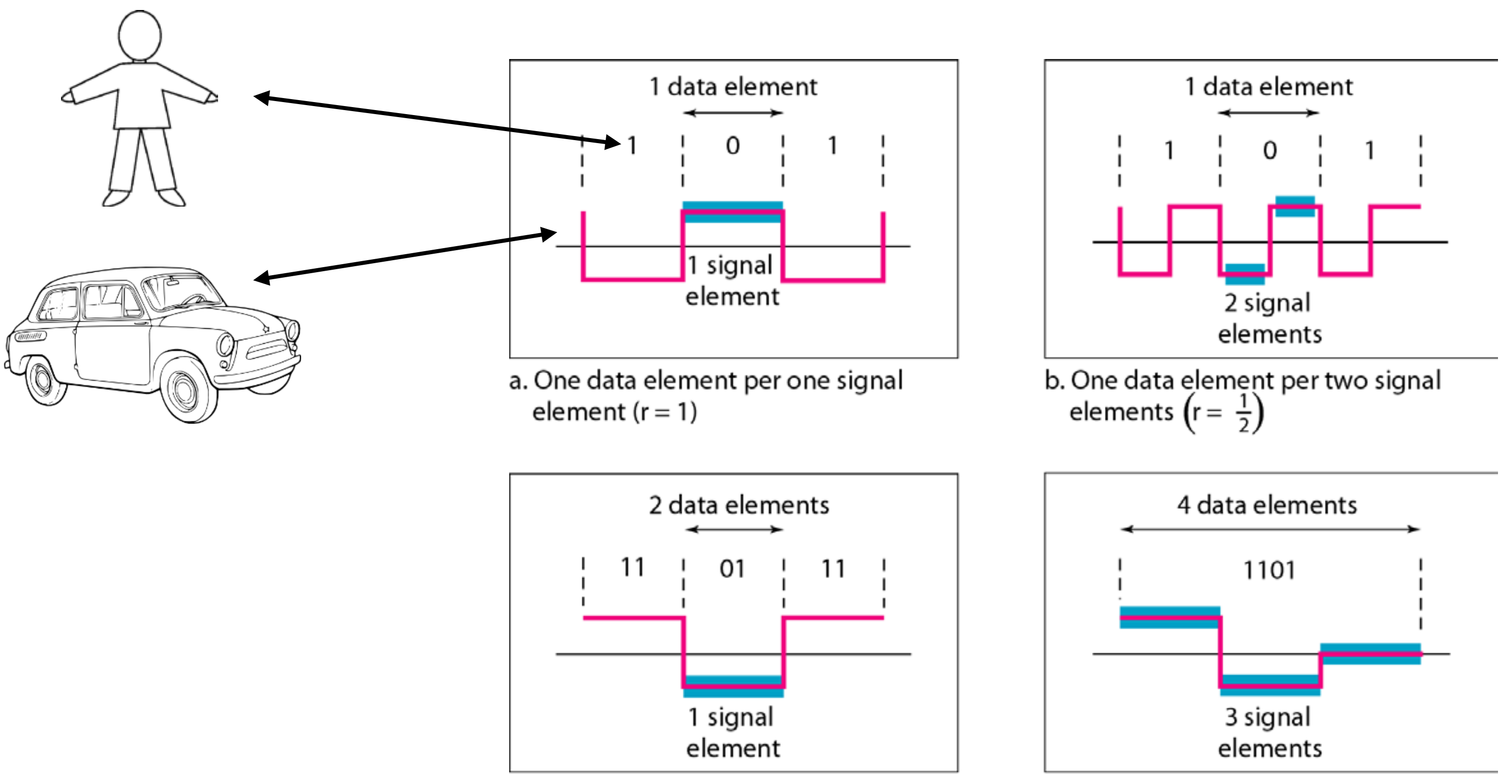
Data	Code	Data	Code
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

4B/5B coding is used in the optical fiber transmission system (FDDI).

Data Rate vs. Baud Rate

Data Rate – **number of data elements (bits) sent in 1 sec – unit: bps**

Signal Rate – **number of signal elements (pulses) sent in 1 sec – unit: baud**



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)

One goal of data communications is to increase data rate (speed of transmission) **while decreasing signal rate** (bandwidth requirements).

Data Rate vs. Baud Rate (cont.)

r = data rate / signal rate – **ratio between data rate and signal rate**

$$r \ [\text{bits/pulse}] = \frac{N \ [\text{bits/second} = \text{bps}]}{S \ [\text{pulse/second} = \text{baud}]}$$

depends on line coding scheme used

larger values preferred!

Signal rate observed in case of a particular data-bit stream: – **depends on N [bps], 1/r [pulse/bit], and the actual data pattern**

- **signal rate for a pattern of all 1-s or all 0-s may be different from that for a patter of alternating 1-s and 0-s**

$$S = c \cdot N \cdot \frac{1}{r} \ [\text{pulses/sec}]$$

case factor (c)

supported [bps] (N)

depends on deployed line coding (1/r)

Data Rate vs. Baud Rate (cont.)

Example **[S in NRZ-L]**

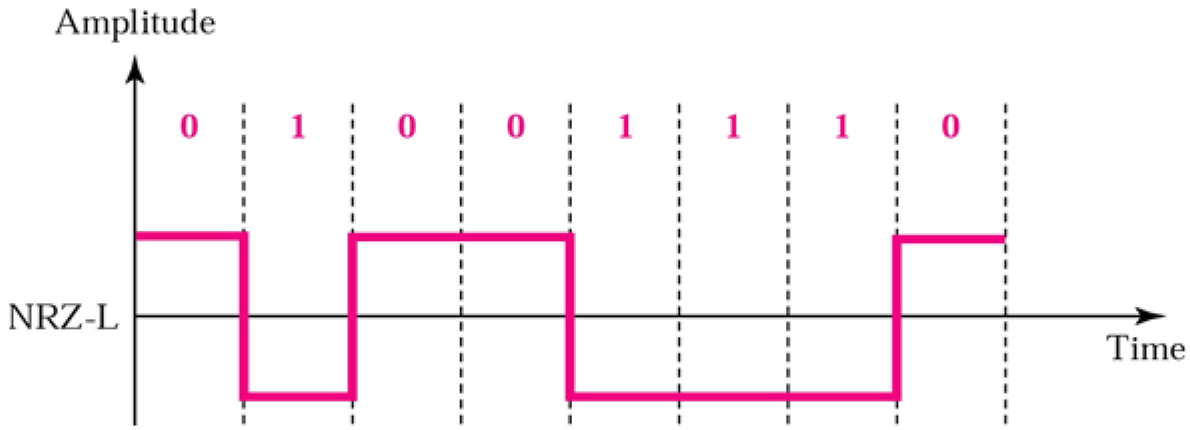
On average, for NRZ-L, r = 1 [bit/pulse]. Assume N = 1 kbps = 1000 bps.

By observing a NRZ-L signal 'on the wire', this does not mean that we will se see S = N * (1/r) = 1000 pulses per second. Instead:

$$S = c \cdot 1000 \text{ [pulses/sec]}$$

where, c $\in$ [0,1].

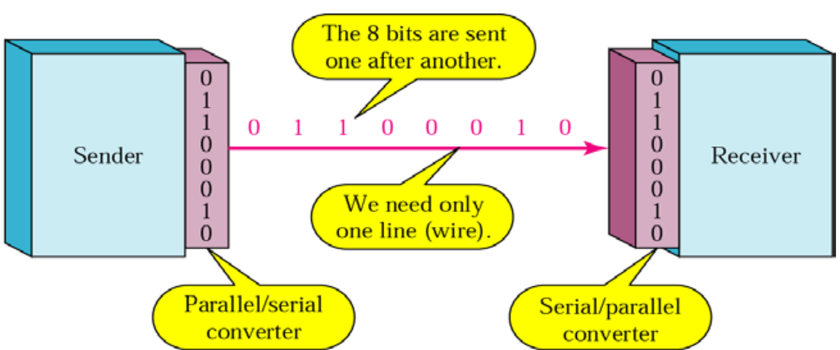
c = 0, for following sequences of bits: 0 0 0 0 0, or 1 1 1 1 1
c = 1, for following sequences of bits: 0 1 0 1 0 1

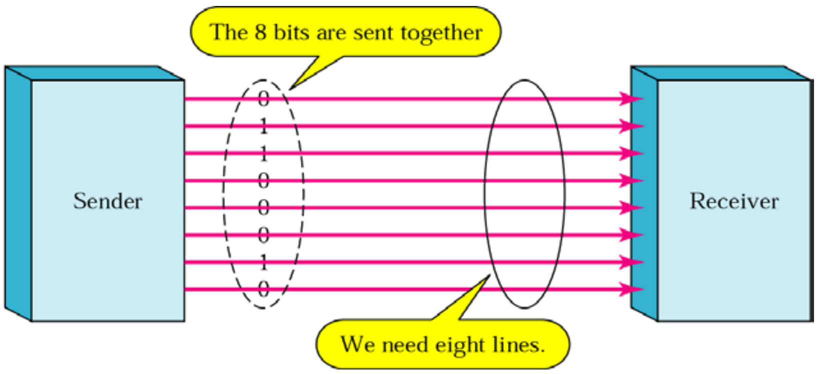


Digital Transmission Modes

How do we send bits / pulses over wire?

- **Serial Mode: 1 bit is sent with each clock tick**
  - **one communication channel / wire is needed**
- **Parallel Mode: multiple bits are sent with each clock tick**
  - **multiple channels / wires, bundled in one cable, are required**
  - **advantage: n-times faster than serial mode**
  - **disadvantages:**
    1) cost = 8x wires,
    2) more interference
    3) synchronization problem among different wires over long distance
  - **only used over short distances**





Digital Transmission Modes (cont.)

Serial Cable

- # of connector pins: 9 or 25
- typical uses: routers, firewalls, load balancers, dial-up modems

Parallel Cable

- # of connector pins: 25 or 36
- typical uses: printers, scanners

Exercise

1. Pulse rate is always _______________ the bit rate.
   (a) greater than
   (b) less than
   (c) greater than or equal to
   (d) less than or equal to

2. Which encoding type always has a nonzero average amplitude?
   (a) unipolar
   (b) polar
   (c) bipolar
   (d) all the above

3. Which of the following encoding methods does not provide for synchronization.
   (a) NRZ-L
   (b) RZ
   (c) NRZ-I
   (d) Manchester

4. Block coding can help in _______________ at the receiver.
   (a) synchronization
   (b) error detection
   (c) attenuation
   (d) (a) and (b)