ECE2305 Lecture Slides

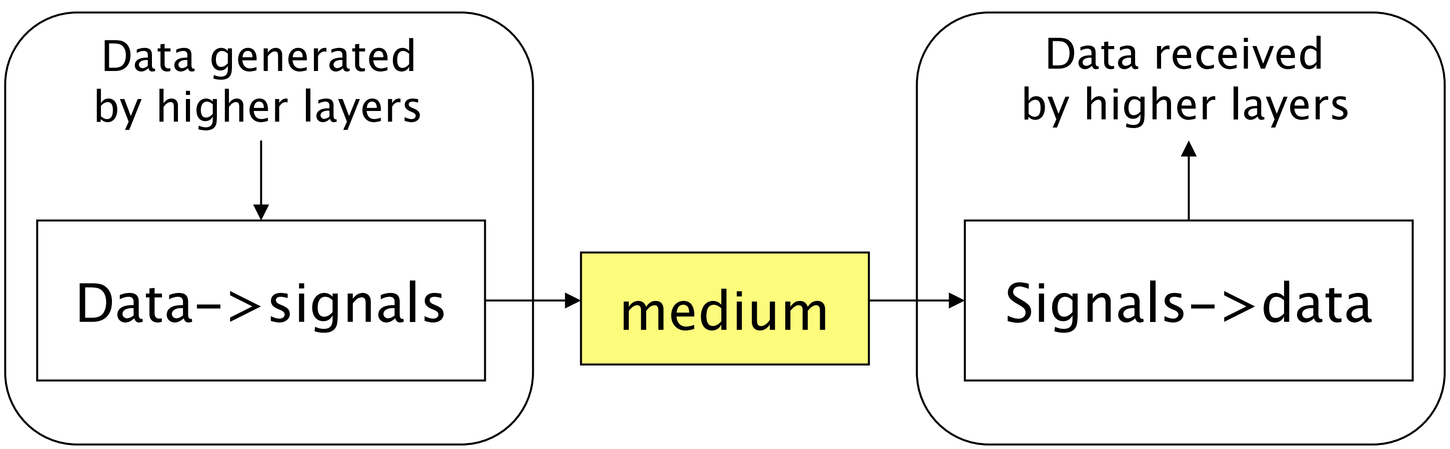
William Stallings: Data and Computer Communications
Section 5.1 – "Digital Data, Digital Signals"

D. Richard Brown III
Worcester Polytechnic Institute
Electrical and Computer Engineering Department

Adapted from Prentice Hall instructor resources

Basics of Signal Encoding

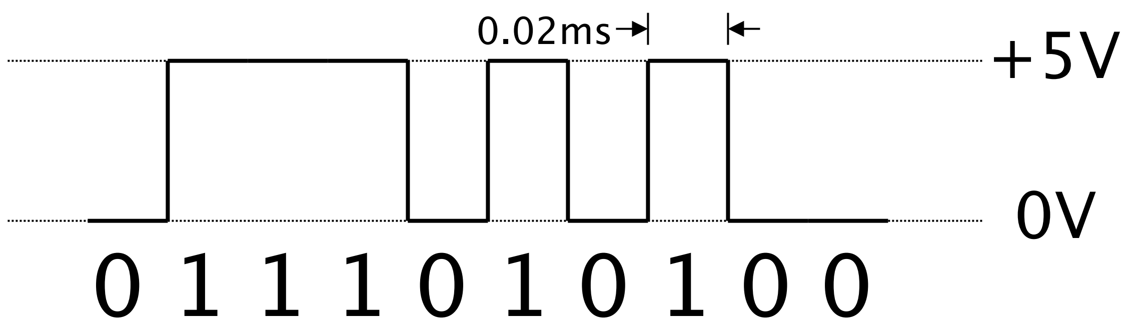
- Important function of the physical layer: Convert data (e.g. bits) to signals (e.g. voltages).
- The signal must be designed to efficiently propagate through the medium.
- The signal must also be designed so that the receiver can correctly interpret it.



"Digital" Signaling

- The waveform sent through the medium has discrete levels
  - Perfectly square pulses are impossible to generate
  - Attenuation, distortion, and/or noise may cause the received signal to look somewhat different
- Characteristics:
  - Polar vs. unipolar
  - Data rate (bits per second)
  - Modulation rate (signal transitions per second)

Example:



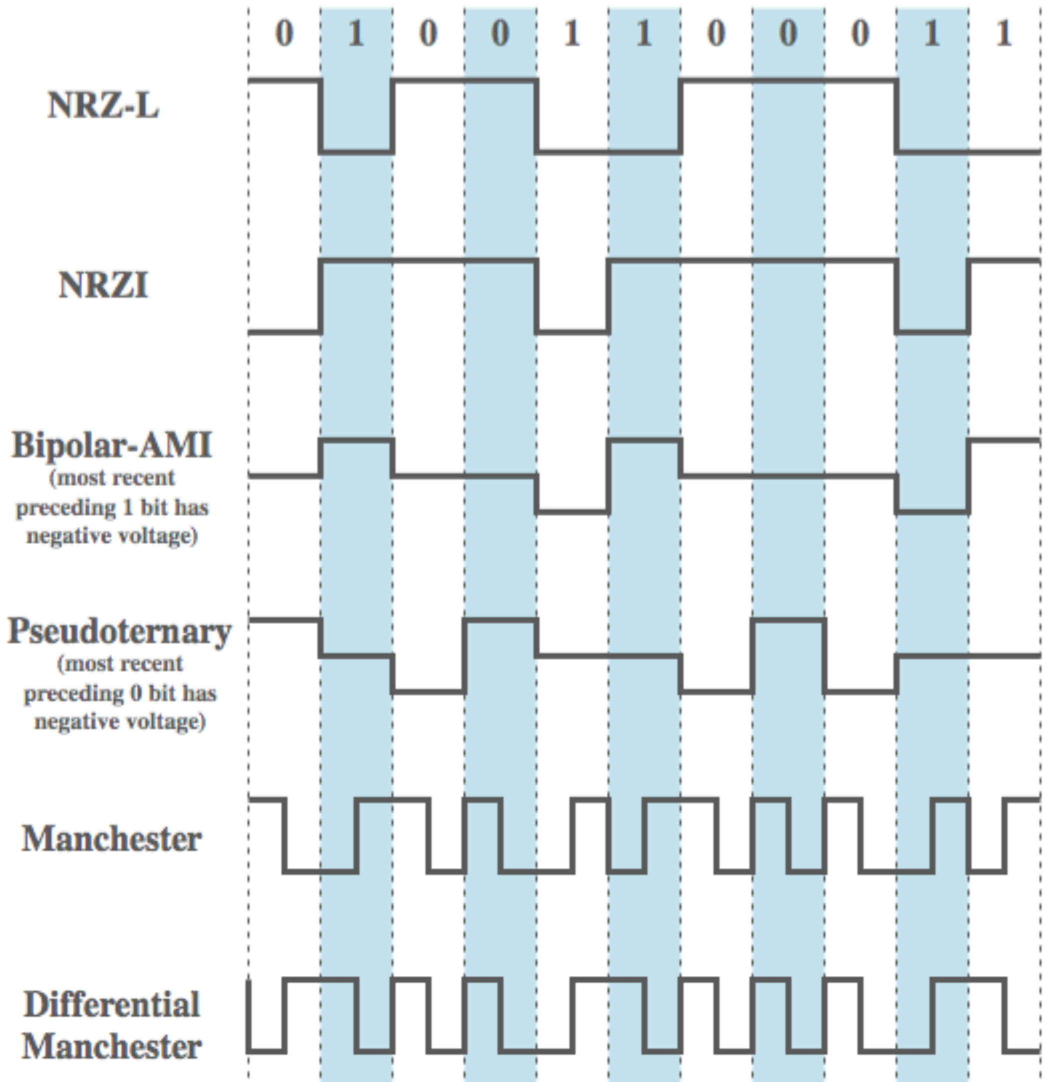
Where is "digital" signaling used?

- Often used for communication over dedicated wired media
  - Ethernet
  - RS-232
  - Etc.
- Not used for:
  - Wireless communication
  - Optical communication
  - Cable modems
  - Digital subscriber loops (DSL)

Characteristics of Digital Signal Encoding Schemes

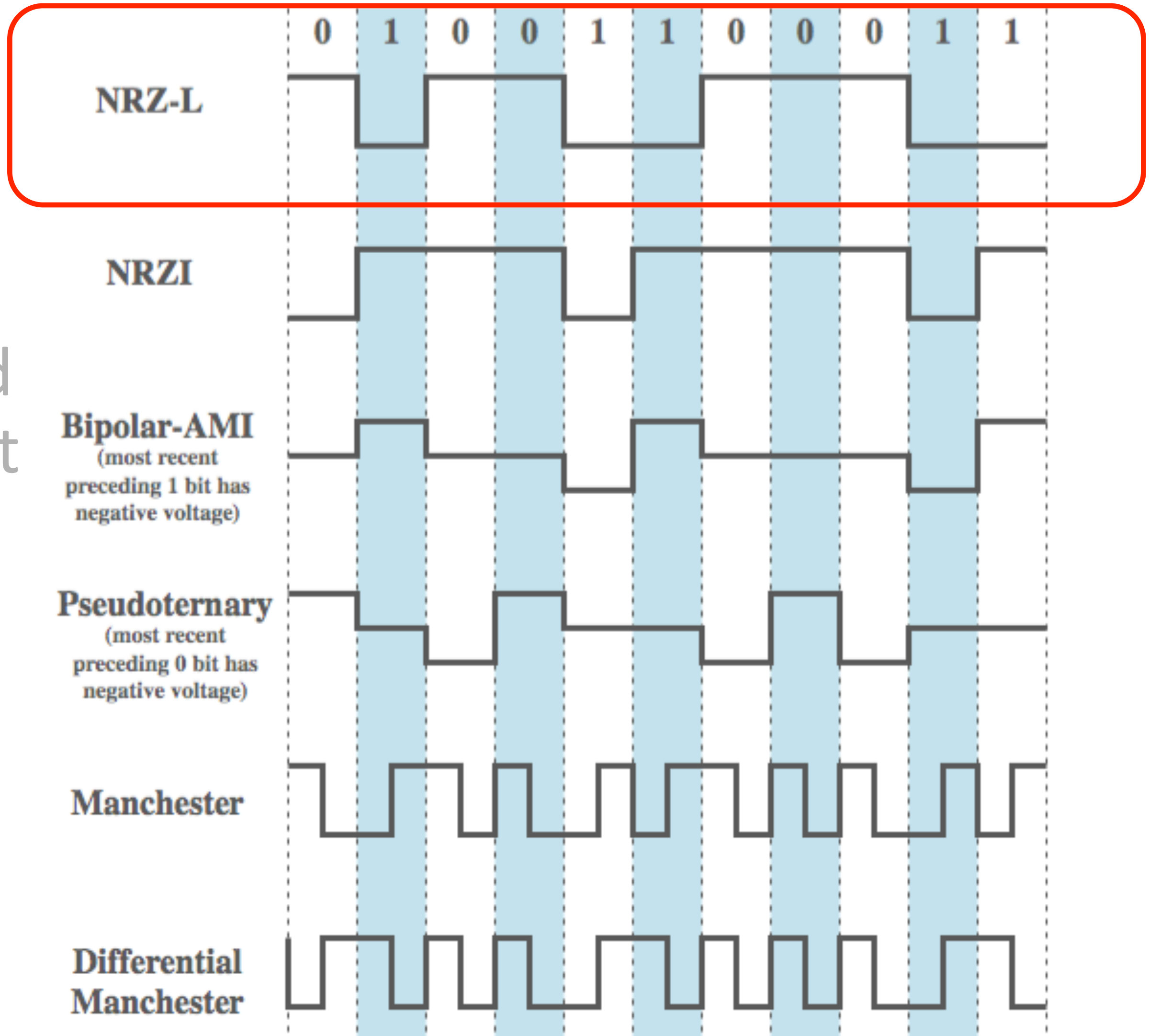
- Signal spectrum
  - Less high frequency content means we can use cheaper cables or go longer distances without repeaters.
- Clocking
  - The receiver needs to know where the start and end of each bit occurs.
  - Some signaling techniques make it easy on the receiver to determine the timing of the bits.
- Error detection
  - Features built into the signaling scheme to detect errors.
- Noise immunity
- Cost and complexity

Some Common Digital Encoding Schemes



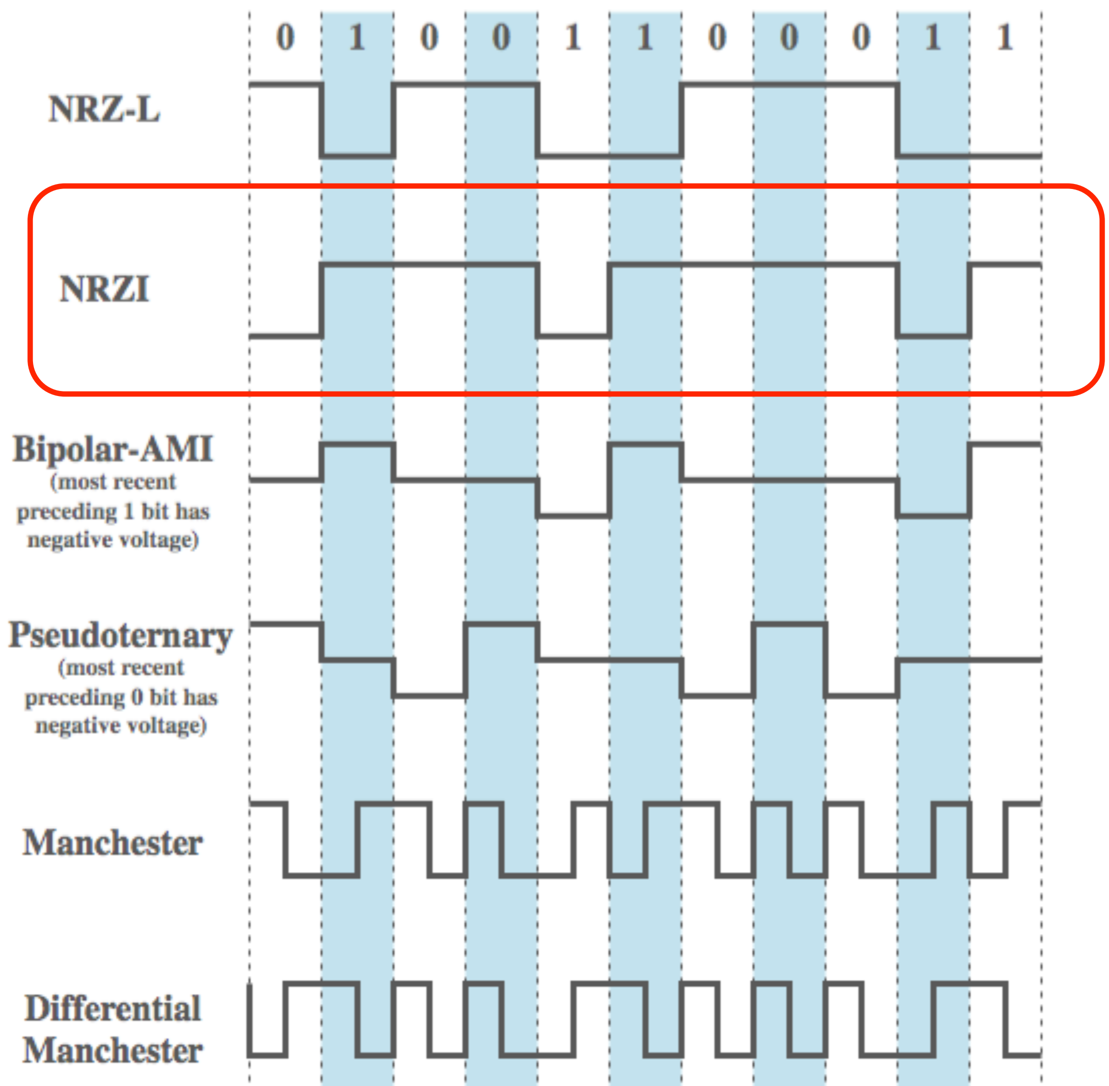
Nonreturn to Zero-Level (NRZ-L)

- Two different voltages:
  - Logical 0 -> V1
  - Logical 1 -> V2
- Signal voltage held constant during bit interval
  - Unipolar: either V1 or V2 is equal to zero. The other voltage is usually positive, e.g. +5V.
  - Bipolar: V1 = -V2



Nonreturn to Zero Inverted

- Two voltages: V1 and V2 (can unipolar or bipolar)
  - Logical 1 -> transition from V1 to V2 or V2 to V1
  - Logical 0 -> no transition
- Signal voltage held constant during bit interval
- This is an example of "differential encoding"
  - data mapped to changes in signal level rather than actual levels
  - detection of a transition is often more reliable that detection of a level



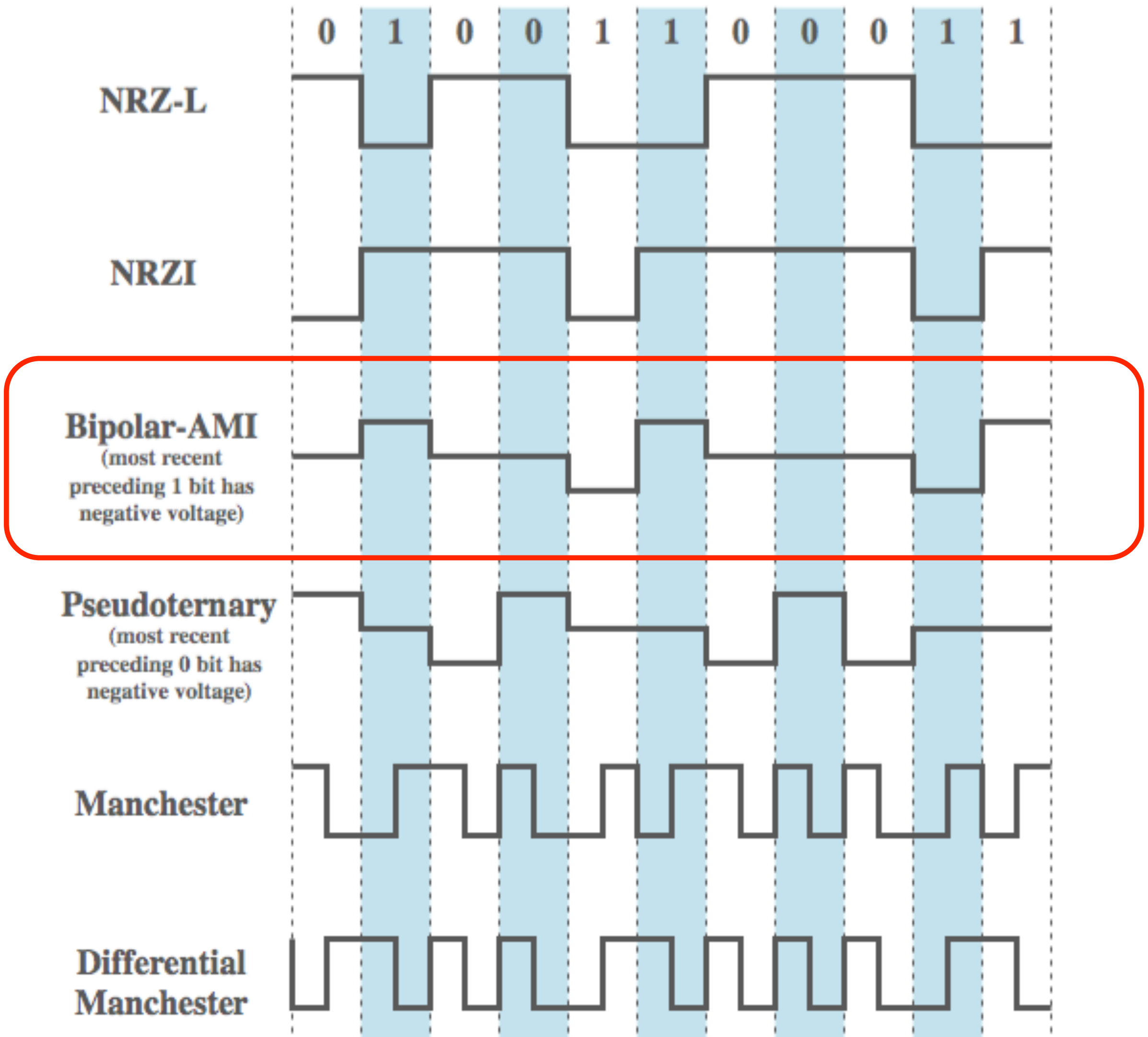
NRZ Pros & Cons

- Pros
  - Easy to engineer
  - Pretty good spectrum containment
- Cons
  - Potential DC (zero-frequency) component
  - Potential loss of synchronization if long strings of zeros or ones sent

Multilevel Binary: Bipolar-AMI

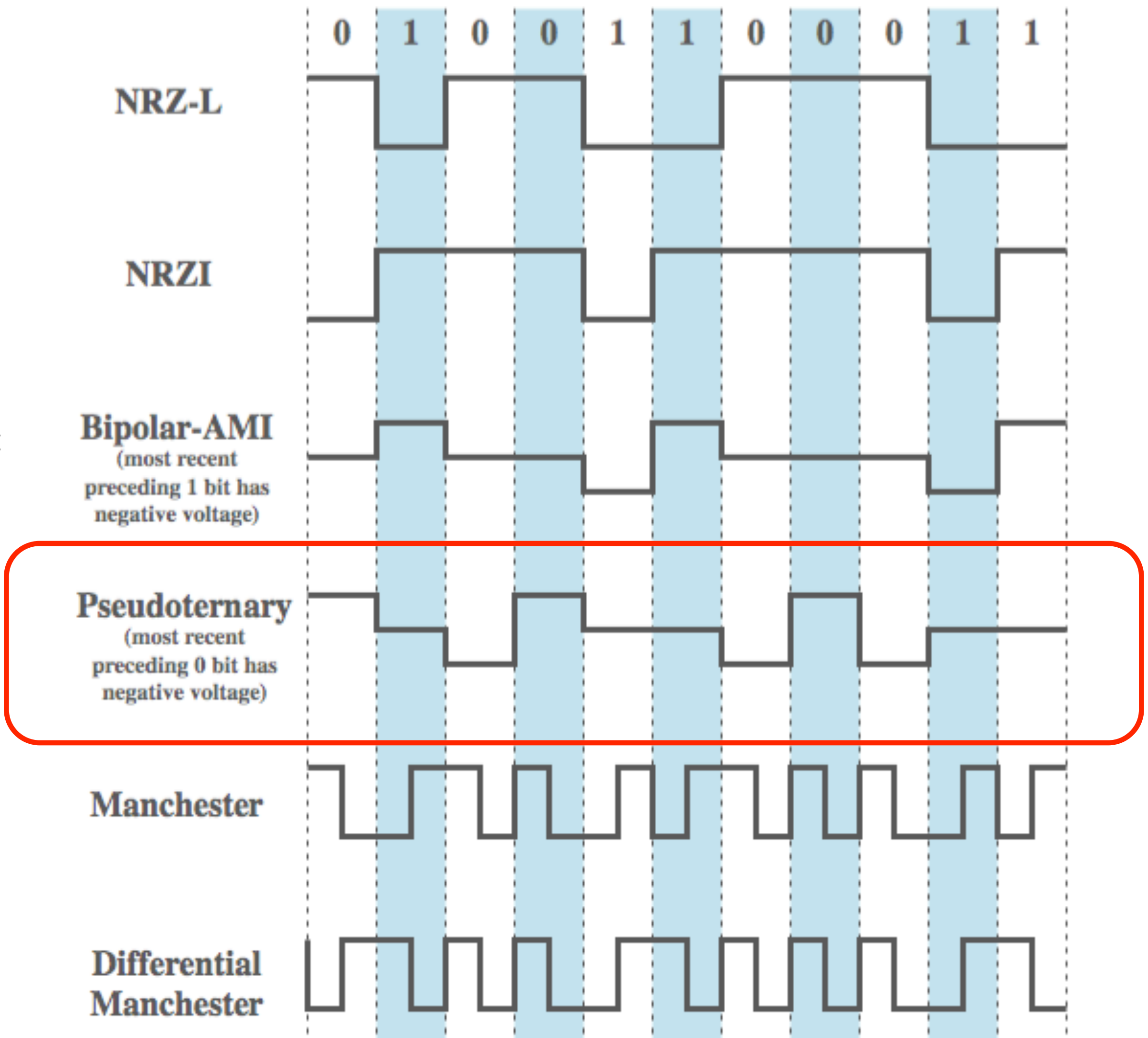
(AMI = "alternate mark inversion")

- Three voltage levels: +V, 0, -V
  - Logical 0 -> output zero voltage
  - Logical 1 -> pulse at voltage +V or -V
    - Pulse transmitted with opposite polarity of last pulse
  - Signal voltage held constant during bit interval
- Properties:
  - No loss of sync if a long string of ones
  - Long runs of zeros still a problem
  - No DC (zero-frequency) component
  - Better spectral properties than NRZ-L & NRZI
  - Some built-in error detection
    - e.g. two consecutive positive pulses: illegal!



Multilevel Binary: Pseudoternary

- Same idea as Bipolar-AMI
- Three voltage levels: +V, 0, -V
  - Logical 1 -> output zero voltage
  - Logical 0 -> pulse at voltage +V or -V
    - Pulse transmitted with opposite polarity of last pulse
  - Signal voltage held constant during bit interval
- Same properties of Bipolar-AMI
  - No advantage or disadvantages
  - Each used in different applications

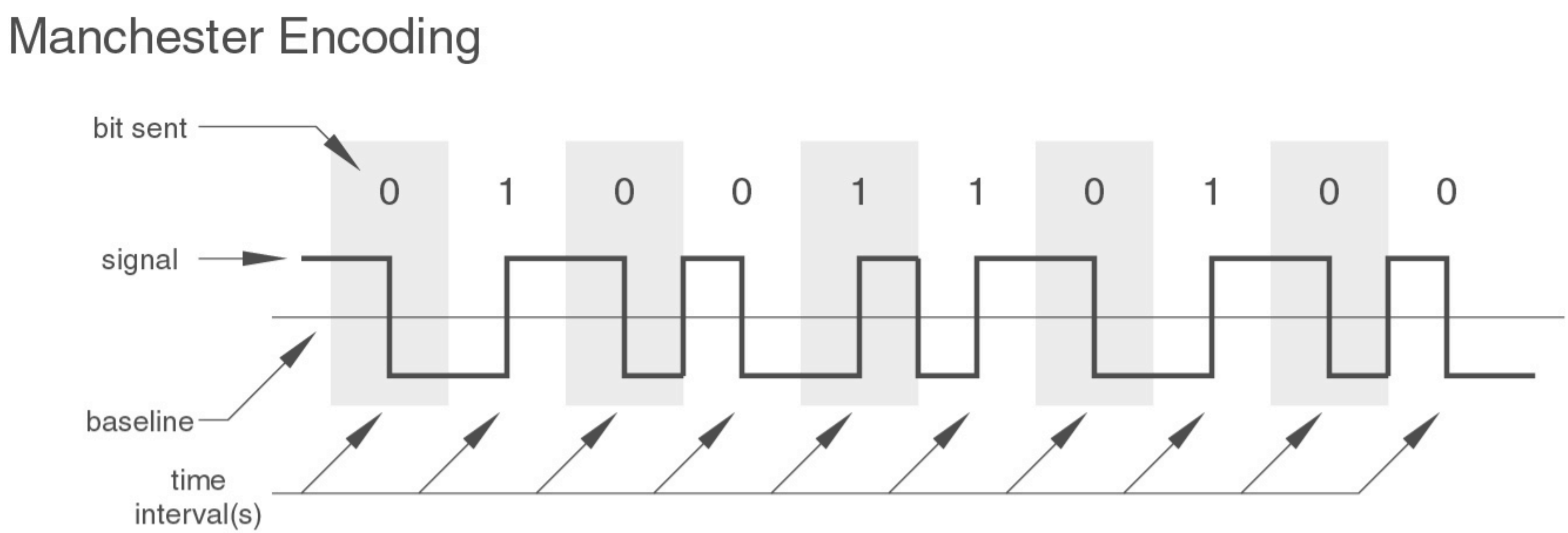


Multilevel Binary Issues

- Loss of synchronization with long runs of 0's or 1's
  - Workaround: insertion of bits or scrambling
- Not as efficient as NRZ
  - Each signal element only represents one bit
    - receiver distinguishes between three levels: +V, -V, 0
  - A 3 level system could actually represent $\log_2 3 = 1.58$ bits in each bit period
  - Requires approx. 3dB more signal power than NRZ for same of bit error rate (BER)

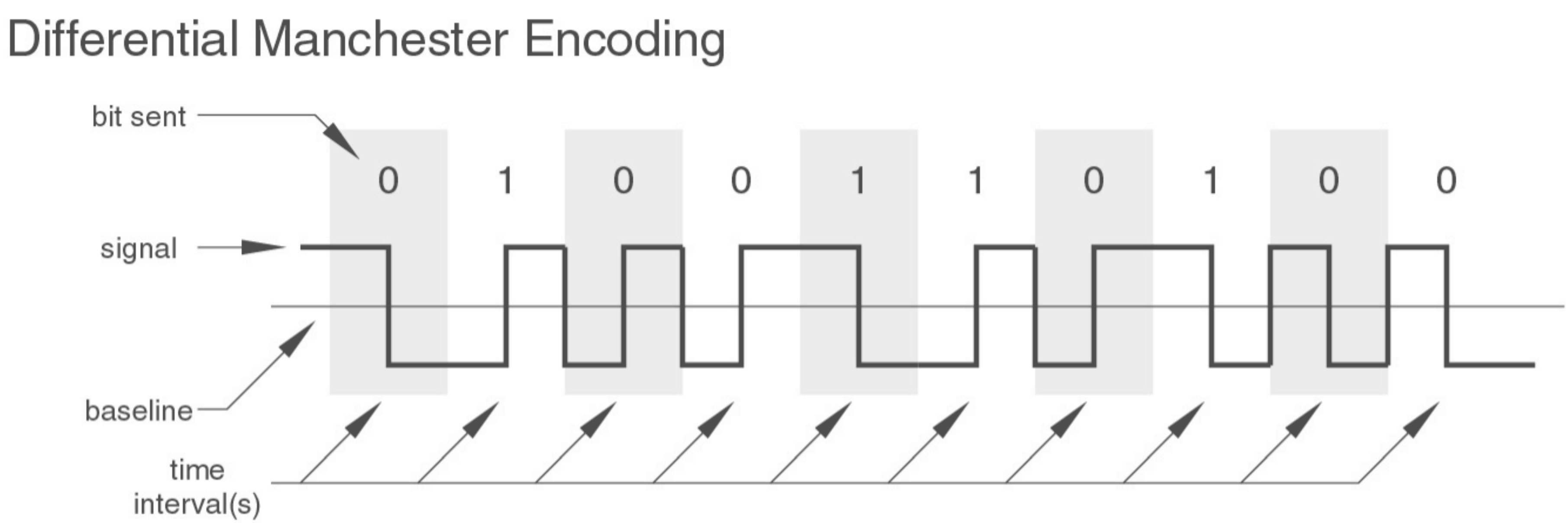
BiPhase Encoding Method 1: Manchester Encoding

- Main idea: signal transition in middle of each bit period
- Why do this? Transition serves as clock and data
- Logical 1 -> transition from low to high
- Logical 0 -> transition from high to low
- Used by IEEE 802.3 (Ethernet LAN)



BiPhase Encoding Method 2: Differential Manchester Encoding

- Like regular Manchester: transition in each bit period
- Differentially encoded
  - Logical 0 -> transition at start of bit period
  - Logical 1 -> no transition at start of bit period
- used by IEEE 802.5 (token ring LAN)



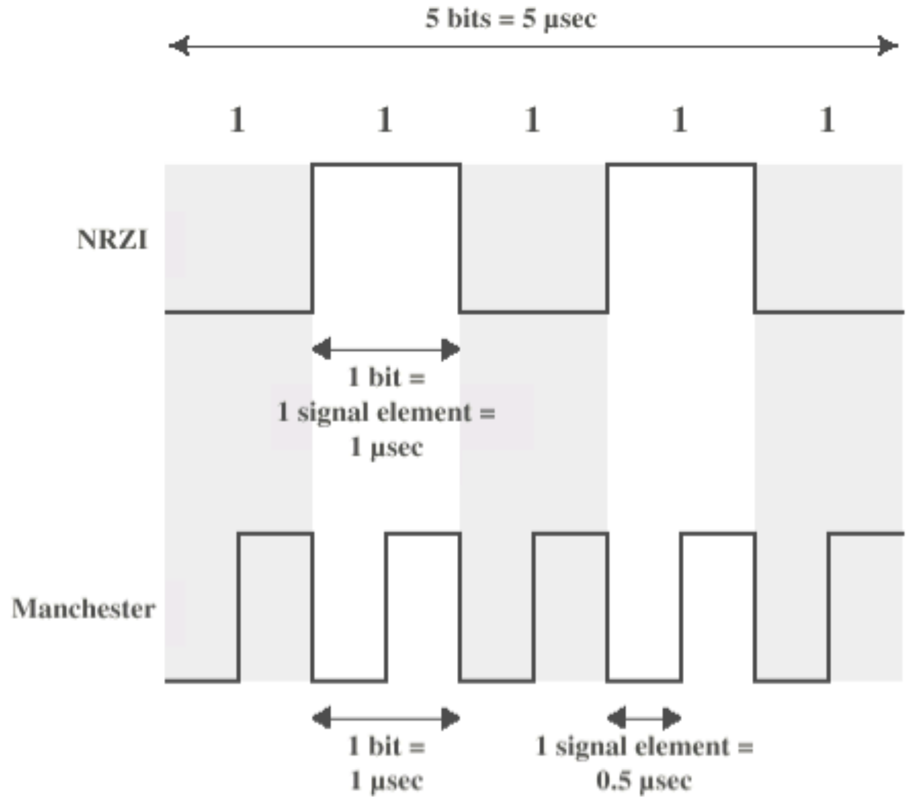
Biphase Pros and Cons

- Pros
  - Self clocking: every bit period guaranteed to have a mid-bit transition
  - No DC (zero-frequency) component
  - Some built-in error detection capabilities
- Cons
  - Poor spectral containment (requires more bandwidth)
    - At least one transition per bit period (and possibly two)
    - Maximum modulation rate is twice NRZ

Modulation Rate

$$D = \frac{R}{L} = \frac{R}{\log_2 M}$$

- D = modulation rate (baud)
- R = data rate (bits/sec)
- M = alphabet size (number of different signal elements)
- L = number of bits per signal element



	Minimum	101010. . .	Maximum
NRZ-L	0 (all 0s or 1s)	1.0	1.0
NRZI	0 (all 0s)	0.5	1.0 (all 1s)
Bipolar-AMI	0 (all 0s)	1.0	1.0
Pseudoternary	0 (all 1s)	1.0	1.0
Manchester	1.0 (1010 . . .)	1.0	2.0 (all 0s or 1s)
Differential Manchester	1.0 (all 1s)	1.5	2.0 (all 0s)

Scrambling: A workaround for the problems with multilevel modulation

- Use <u>scrambling</u> to replace sequences that result in long periods of constant voltage
- The replacement sequences must
  - produce enough transitions to maintain sync
  - be recognized by receiver & replaced with the original (intended) sequence
  - be same length as the original sequence
- Design goals
  - have no dc (zero-frequency) component
  - have no long duration of constant voltage
  - have no reduction in data rate
  - provide some error detection capability

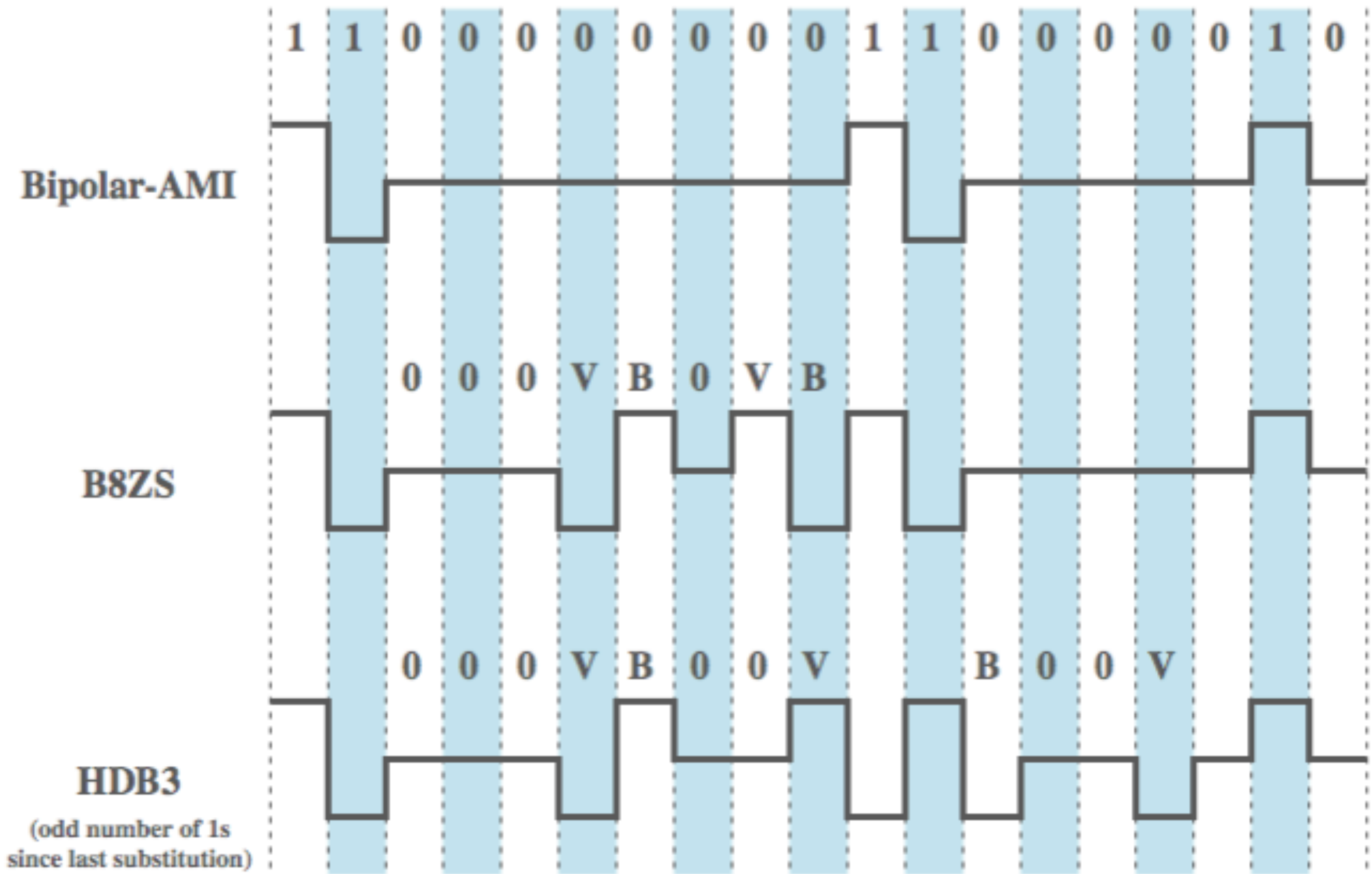
Bipolar with 8-zeros substitution (B8ZS)

- Based on bipolar-AMI
  - Recall that a long string of zeros causes a long period with no signal transitions, which could lead to loss of synchronization
- Scrambling specifics:
  - Data is buffered to detect strings of eight consecutive zeros (prior to transmission)
  - Rather than sending 0V for eight signal periods we send:
    - 000+-0-+ if the last voltage pulse preceding the 8 consecutive zeros was positive
    - 000-+0+- if the last voltage pulse preceding the 8 consecutive zeros was negative
  - Note that these cause illegal patterns in AMI. The receiver detects this and interprets these patterns as eight consecutive zeros.

High-Density Bipolar-3 Zeros (HDB3)

- Also based on bipolar-AMI
- Scrambling specifics:
  - Data is buffered to detect strings of four consecutive zeros prior to transmission
  - Substitution based on polarity of preceding pulse (P) and number of ones transmitted since last substitution (N)
    - 000- if P=- and N=odd number
    - 000+ if P=+ and N=odd number
    - +00+ if P=- and N=even number
    - -00- if P=+ and N=even number
  - As before, these signals are illegal for bipolar-AMI. The receiver knows to interpret these patterns as four consecutive zeros.

B8ZS and HDB3



B = Valid bipolar signal
V = Bipolar violation

Spectrum Comparison

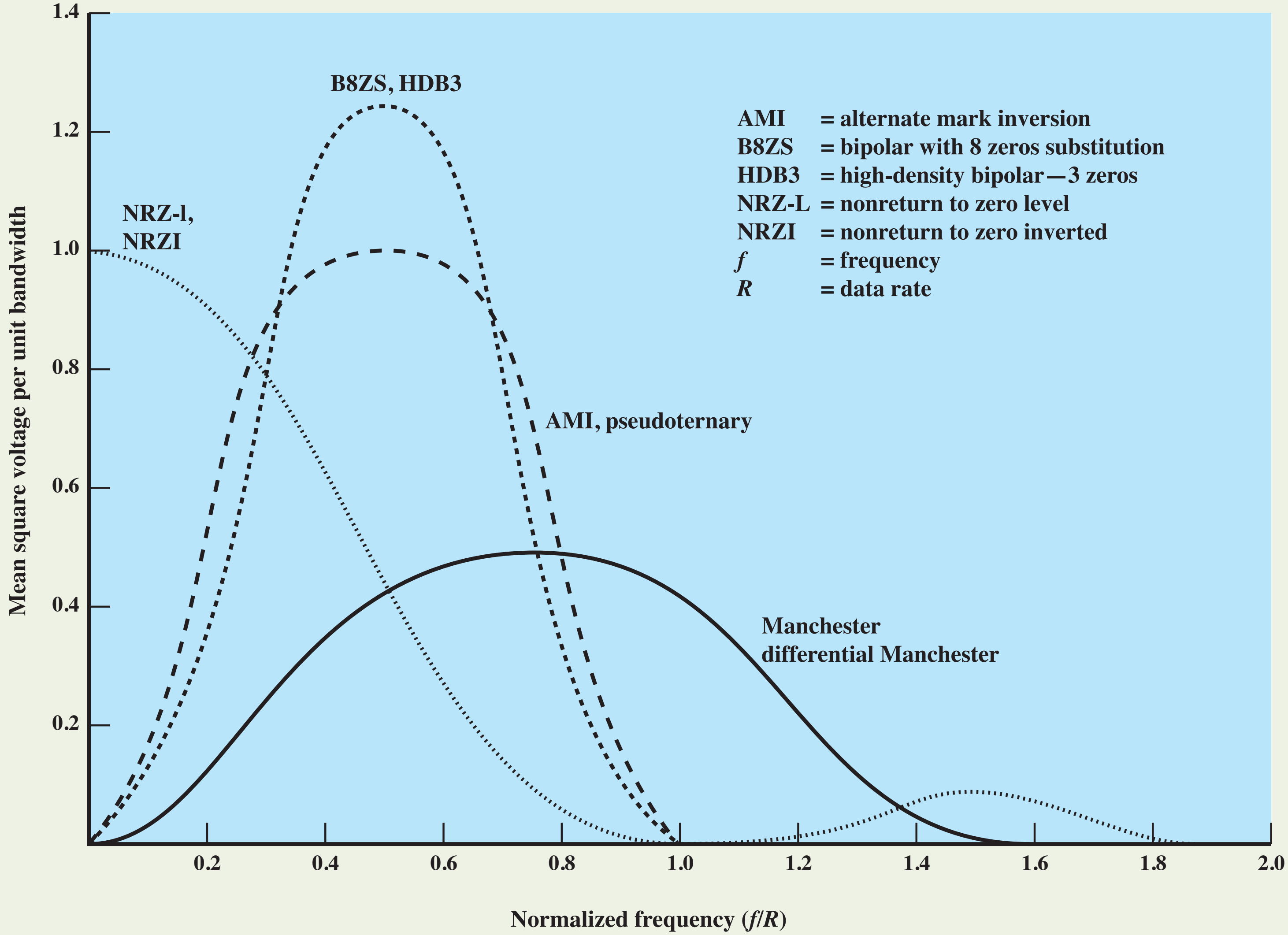


Figure 5.3 Spectral Density of Various Signal Encoding Schemes

Ethernet Standards and Signaling

Standard	Signaling
10BASE-T (copper)	Manchester coded
100BASE-TX (copper)	MLT-3 (three-level signaling similar to bipolar AMI)
100BASE-FX (fiber)	NRZI
1000BASE-T (copper)	5-level signaling (PAM-5)
1000BASE-SX (fiber)	NRZ