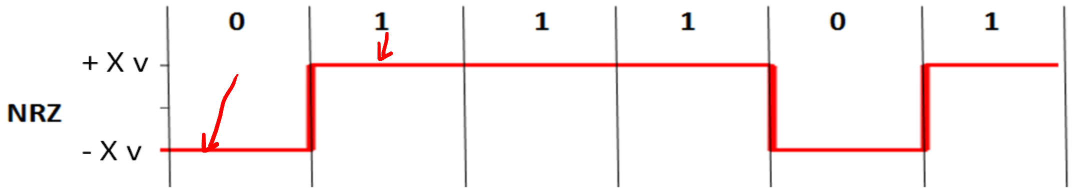
Line Codes

Kameswari Chebrolu

Quick Recap

- Data: 101111011
- Line coding converts bits to voltage or power levels
- Non Return to Zero (NRZ): Represent 0 and 1 by two different levels



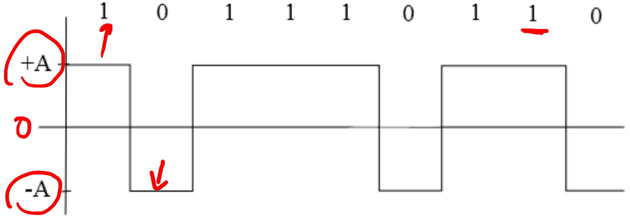
Decoding

Attenuation, noise

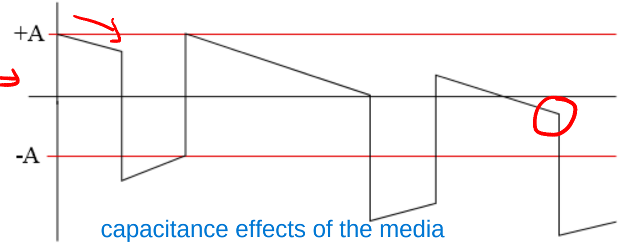
- How does a receiver decode the data i.e determine bits from waveform?
- Compare with a threshold   RC

  like above 0: 1 below 0: 0

  – Receiver maintains average of the signal, uses average to distinguish between low and high signals
- Clock to determine bit durations
  – Receiver's clock need to be perfectly synchronized with the sender, otherwise it results in errors
  – Clock should preferably be derived from the received signal itself
    - Transitions in received signal help recover the clock

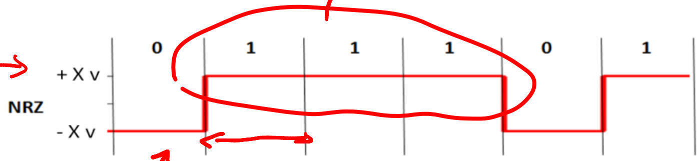


Rx



capacitance effects of the media

6 1's

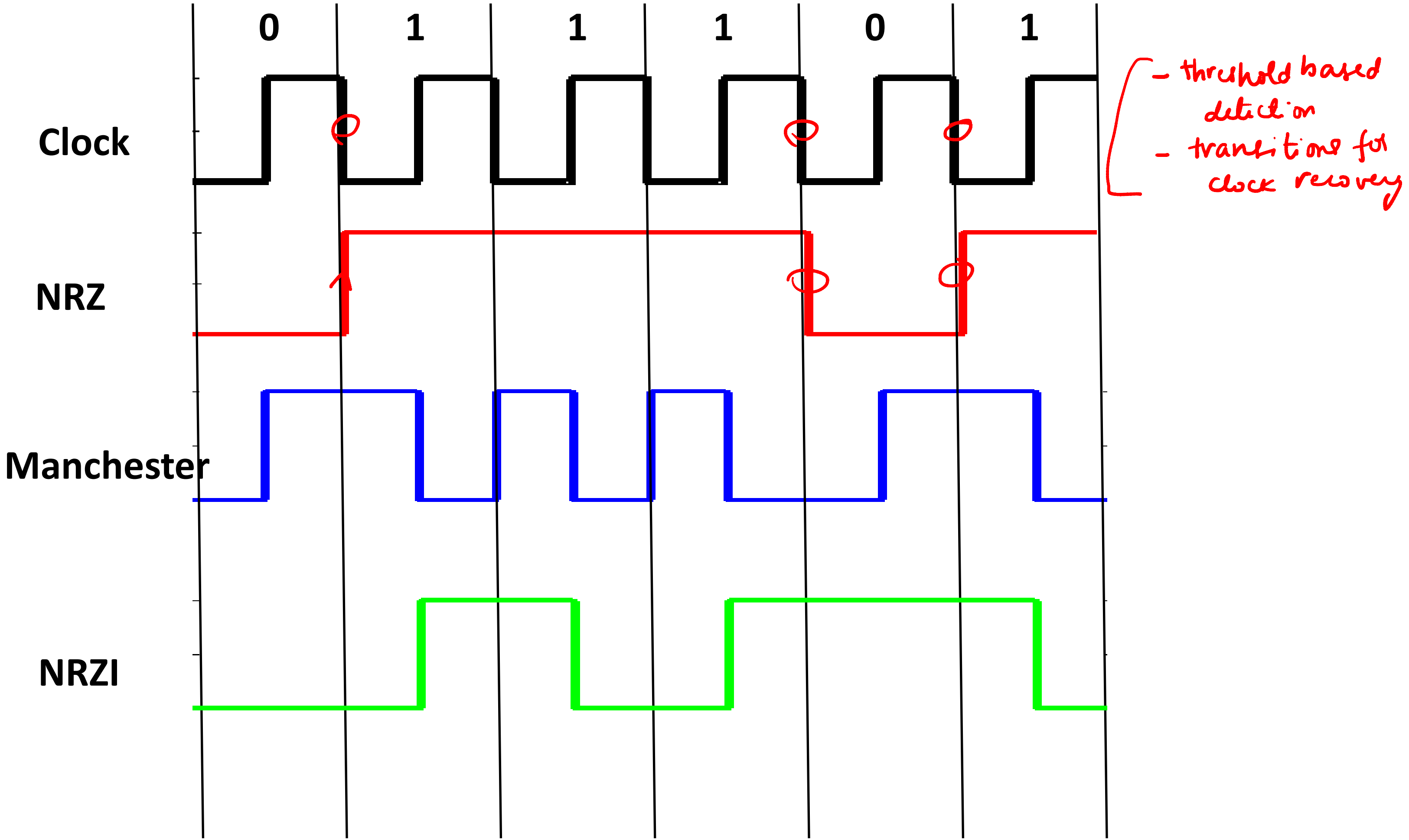


3 1's and not 5 1's or 8 1's

signal

clock → cost

we can send clock via another wire, but its costly


0
1
1
1
0
1
Clock
NRZ
Manchester
NRZI
- threshold based detection
- transitions for clock recovery

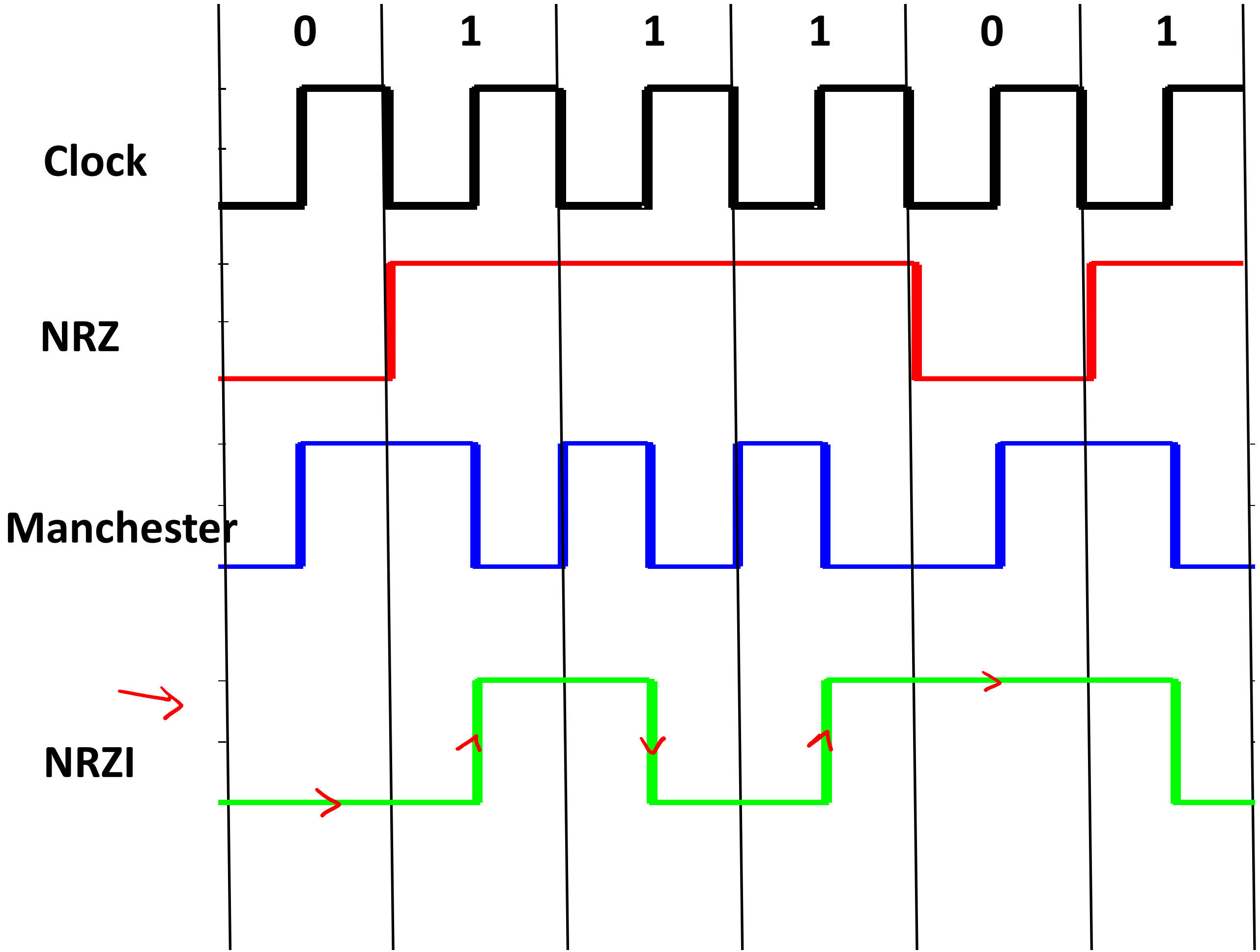
Problems with NRZ

- Consecutive 1s or 0s
  - Changes the average leading to errors (*baseline wander)*
  - Lesser number of transitions leads to clock drift between sender and receiver
- Goal of Line Encoding: Provide enough number of transitions in the signal (over a specified interval)

NRZ-Inverted (NRZI)

noise does not have that much of an effect

- Form of differential encoding
  - To encode a 1, make a transition
  - To encode a 0, stay at the current signal
- Used in USB

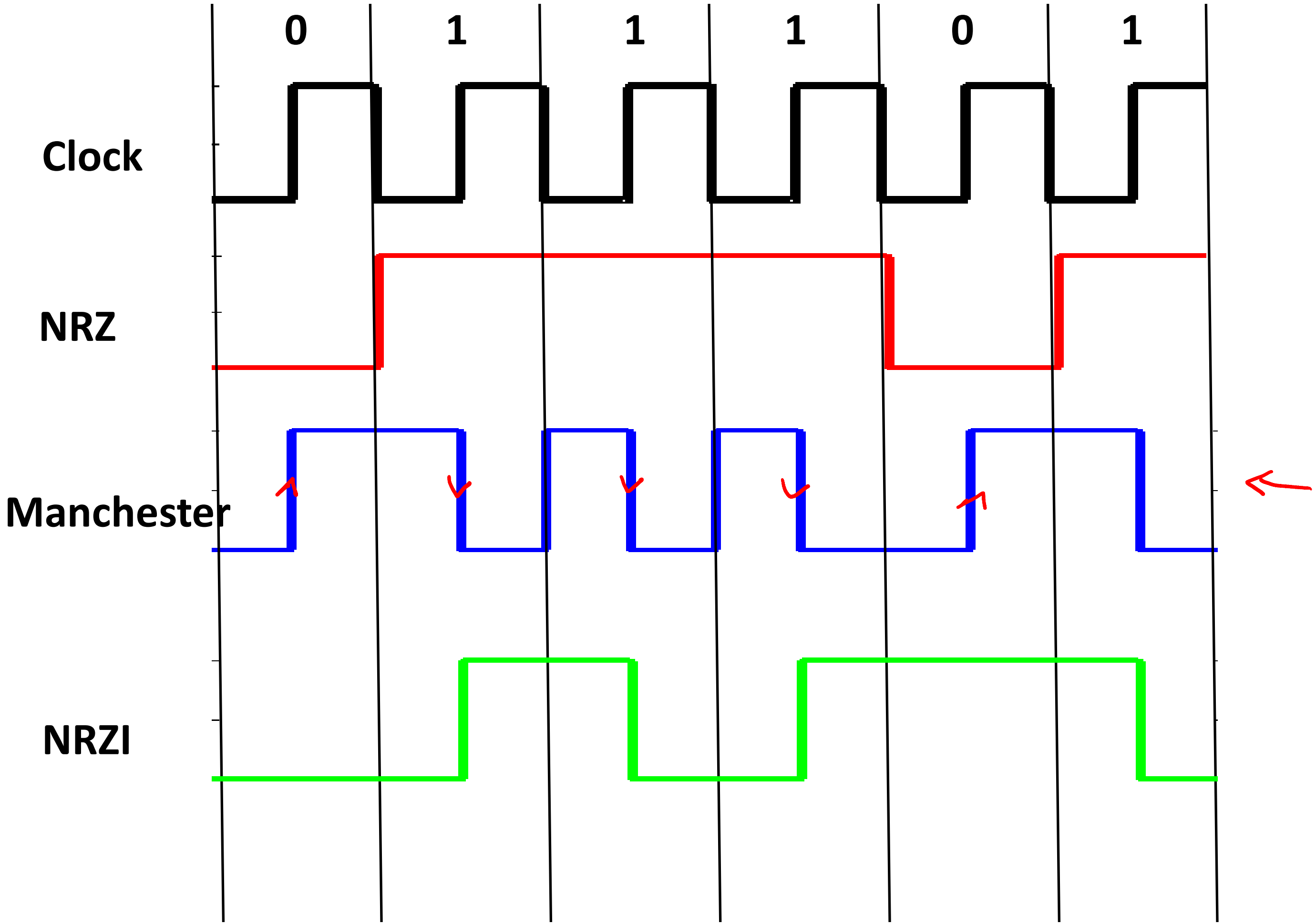

0
1
1
1
0
1
Clock
NRZ
Manchester
NRZI

NRZ-Inverted (NRZI)

- Advantages:
  - Works well in presence of noise (detecting transitions easier than comparison with threshold)
  - Eliminated baseline wander
  - Accidental inversion of leads from device to twisted pair has no effect
- Solves problem of consecutive 1's but not 0's
  - Clock recovery is difficult in presence of consecutive 0’s
  - Signal can have a dc component

Manchester Encoding

- Transmits XOR of the NRZ encoded data and the clock
  - 0 is encoded as low-to-high transition
  - 1 as high-to-low transition
- Used in Ethernet (10Mbps)


0
1
1
1
0
1
Clock
NRZ
Manchester
NRZI

Manchester Encoding

used in ethernet where we have high B.W, but still in 100Mbps ethernet we dont use this.

- Advantages:
  - Eliminates both baseline wander
  - Easy synchronization (self-clocking)
  - No DC component
- Disadvantage: Only 50% efficient
  - Maximum encoding rate is twice that of NRZ (more number of transitions) → Require more bandwidth
  - One could send twice as many bits in the same time period with NRZ, NRZI

4B/5B Encoding

there will be transition after 3 zeros.

- Used in Ethernet (100Mbps), FDDI
- Every 4 bit of actual data is encoded into a 5 bit code
- The 5 bit code words have
  - No more than one leading 0
  - No more than two trailing 0s

if 2 words together, max 3 zeros

- Solves consecutive 0s problem
- The 5 bit codes are sent using NRZI (solves consecutive 1's problem)
- Achieves 80% efficiency

bcz we introduce 1 dummy bit... so only 4/5 re useful

0	0	0000	11110
1	1	0001	01001
2	2	0010	10100
3	3	0011	10101
4	4	0100	01010
5	5	0101	01011
6	6	0110	01110
7	7	0111	01111
8	8	1000	10010
9	9	1001	10011
10	A	1010	10110
11	B	1011	10111
12	C	1100	11010
13	D	1101	11011
14	E	1110	11100
15	F	1111	11101

Summary: Encoding

- Encoding transforms string of bits to voltage levels
- Goal of many encoding techniques
  - Provide enough transitions for clock recovery
  - Achieve above while minimizing bandwidth
- Looked at a variety of line codes: NRZ, NRZI, Manchester, 4B/5B
  - Each has certain advantages and disadvantages