Introduction to Computer Networks

CS640

Better Encoding

https://pages.cs.wisc.edu/~mgliu/CS640/F22/

Ming Liu
mgliu@cs.wisc.edu

Today

Last lecture

- How to transmit bits reliably across the link?

Today

- How to transmit bits reliably and efficiently across the link?

Announcements

- Lab1 due next Tuesday

Non-Return to Zero (NRZ)

Issue #1: baseline wander

- Root cause: dynamic baseline adjustment
- Hard to identify low (0), high (1), and noise signals

Issue #2: clock synchronization

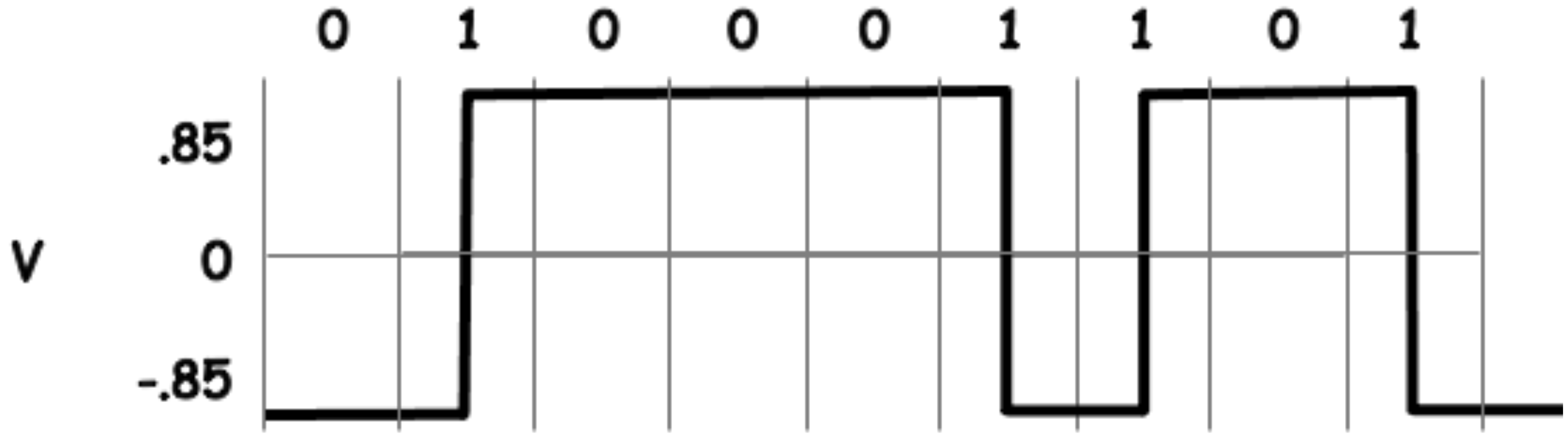
- Root cause: clock drafting cannot be avoided; no global clock domain
- Especially there are consecutive 1s or 0s

Q: Can we get rid of the baseline?

Q: Can we get rid of the baseline?

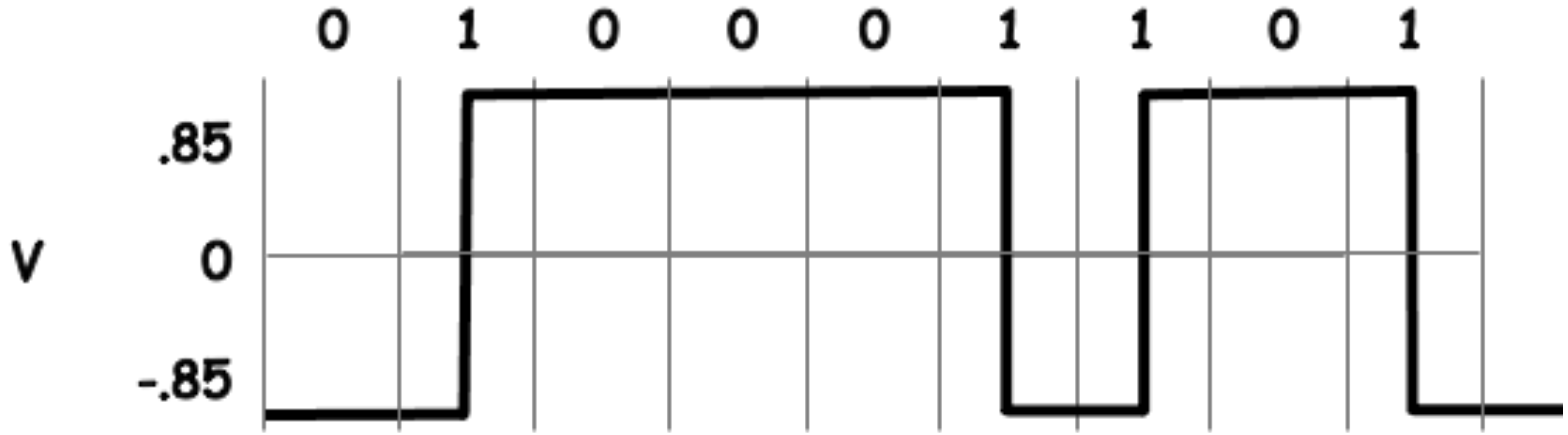
A: Yes. Let's use signal transition instead of the value of a signal to represent a bit.

Non-Return to Zero Inverted (NRZI)



1 -> make transition; 0 -> signal stays the same

Non-Return to Zero Inverted (NRZI)



1 -> make transition; 0 -> signal stays the same

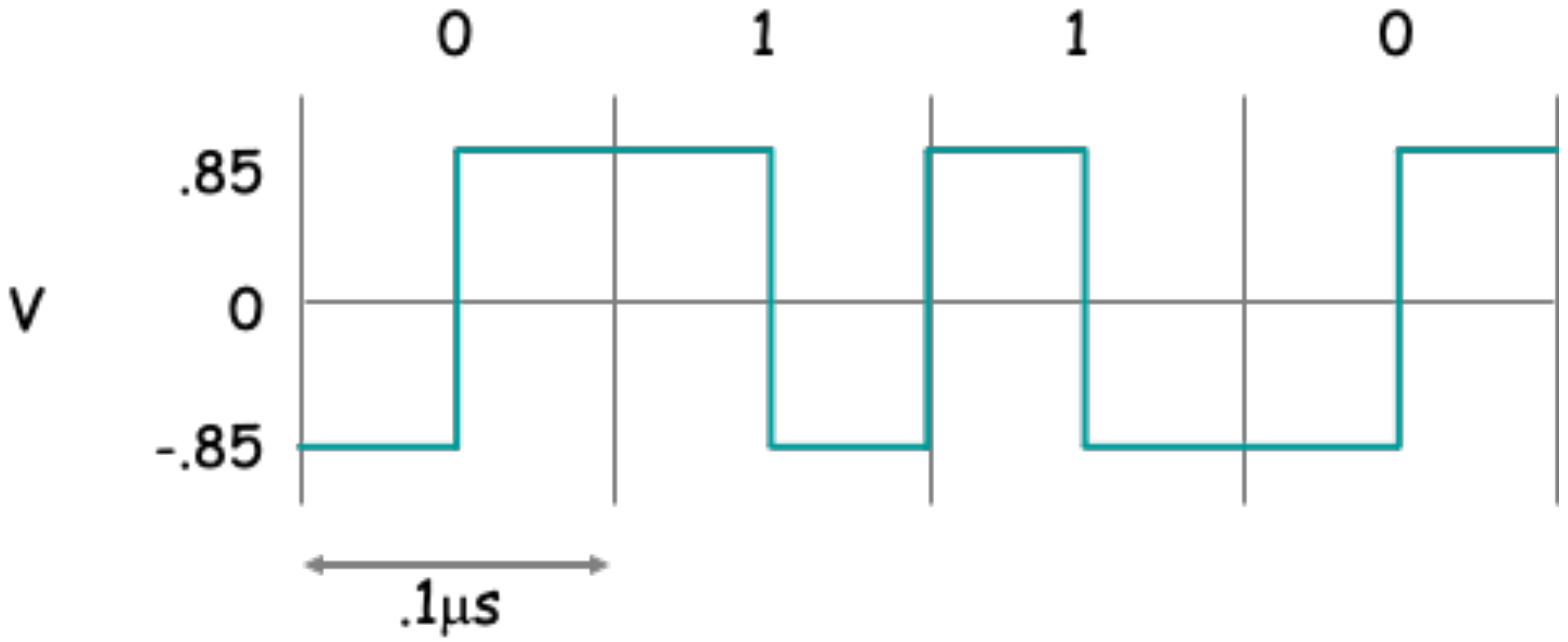
- The clock synchronization issue still exists, especially for long sequences of 0s
- Sender and receiver need more logics

Q: Can we perform frequent in-line synchronization?

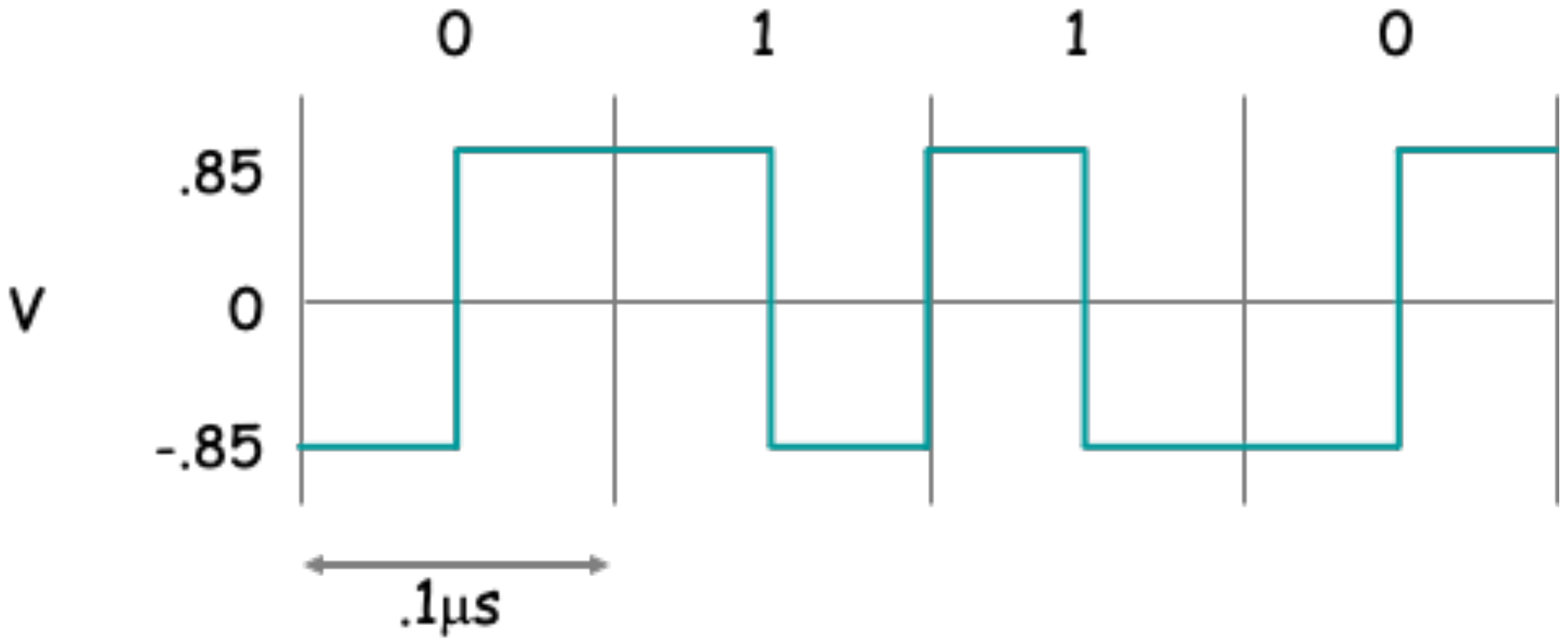
Q: Can we perform frequent in-line synchronization?

A: Yes. Let's enable signal transition for every bit.

Manchester Encoding



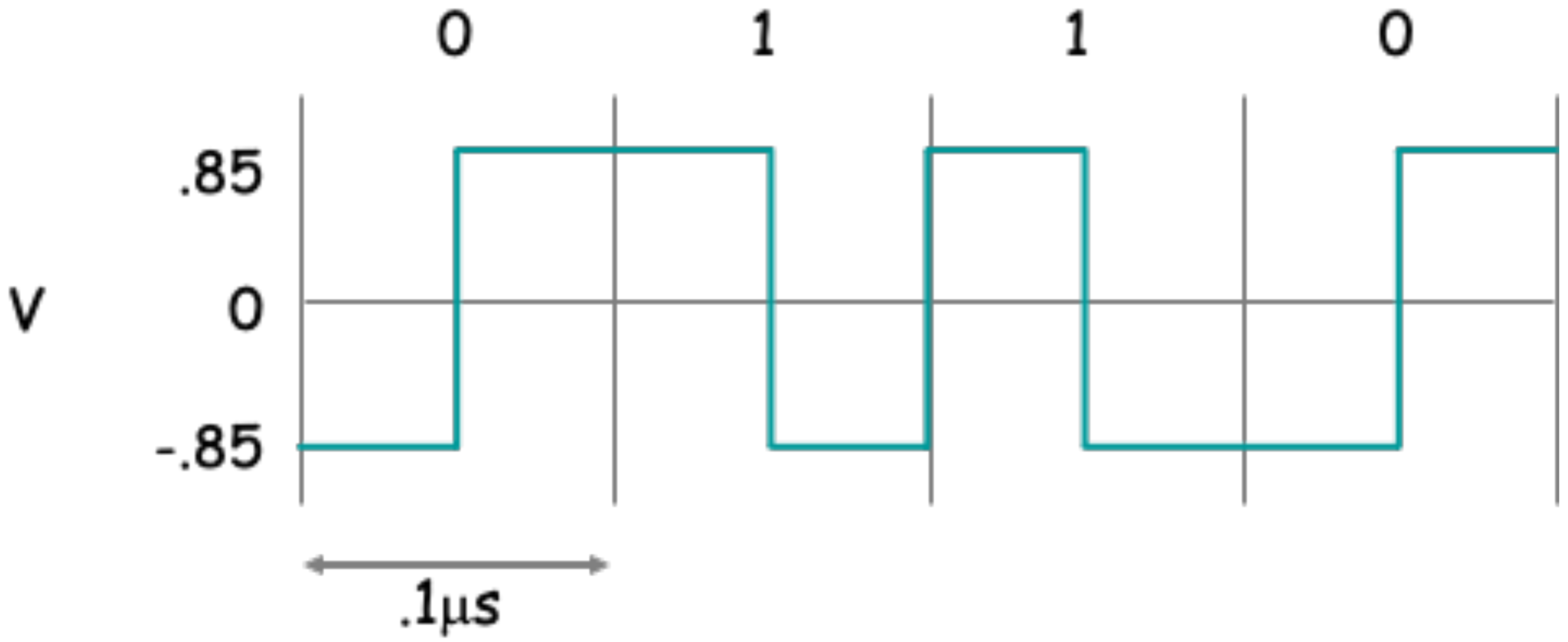
Manchester Encoding



Positive transition for 0, negative for 1

- XOR of NRZ and a local internal clock
- Clock is communicated from the sender to the receiver

Manchester Encoding



More logic is added to the sender and receiver

- Doubles the rate at which signal transitions are made
- The receiver has half the time to detect the pulse

	NRZ	NRZI	Manchester
Baseline wander			
Clock synchronization			
Complexity			
Baud rate utilization			

	NRZ	NRZI	Manchester
Baseline wander			
Baud rate utilization			

Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization	Y	Y/N	N
Complexity	Low (signal monitoring)	Medium (signal monitoring + signal transition detection)	High (signal monitoring + directional signal transition detection)
Baud rate utilization	100%	[50%, 100%]	50%

Q: Can we improve NRZI?

Q: Can we improve NRZI?

A: Yes. Let’s integrate in-line synchronization.

4B/5B Encoding

Every 4 bits of data are encoded in a 5-bit code

- Each symbol had no more than one leading zero and no more than two trailing zeros
- Use NRZI to encode the 5 code bits => 80% utilization

Data	Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111

Data	Code
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Q: On the sender side, how to encode the following bit stream under NRZ, NRZI, Manchester, and 4B/5B? Suppose we represent bits 1 and 0 using high and low signals.

0010 1111 0100

A: Specify the local clock first

Encoding Discussion

An encoding scheme depends on how many signals are used to represent "0" and "1"

- More than "Low" and "High"

Other Encodings

- 8B/10B, 64B/66B, 128/130B, …

Design/evaluate an encoding scheme:

- Signal transition or signal value, clock
- Implementation complexity, baud rate utilization

Physical Layer

Encoding is the key functionality

Physical Layer

Encoding is the key functionality

Vertical view

- A reliable bit delivery channel for a fixed length of bit stream

Physical Layer

Encoding is the key functionality

Vertical view

- A reliable bit delivery channel for a fixed length of bit stream

Horizontal view

- Sender: transfer bits to signals (modulation)
- Receiver: convert signals to bits (demodulation)

Terminology

1. Host
2. NIC
3. Multi-port I/O bridge
4. Protocol
5. RTT
6. Packet
7. Header
8. Payload
9. BDP
10. Baud rate

Principle

1. Layering

Technique

1. NRZ Encoding
2. NRZI Encoding
3. Manchester Encoding
4. 4B/5B Encoding

Summary

Today's takeaways

#1: NRZ, NRZI, Manchester, 4B/5B all have pros and cons.

#2: Evaluating an encoding scheme should consider reliability, synchronization, complexity, and utilization

Next lecture

- Framing and Error Detection