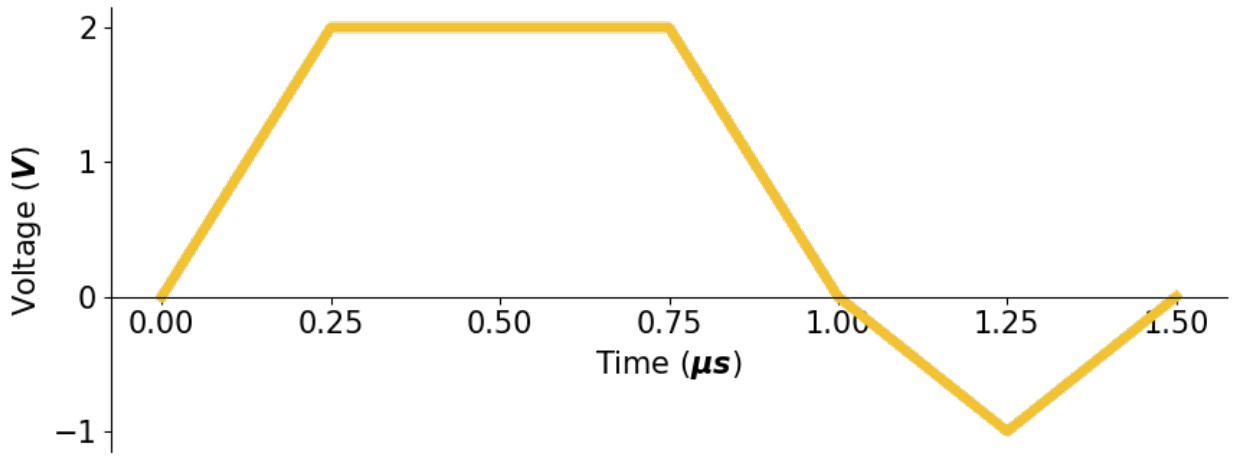
Homework 1: Intersymbol Interference, Clock Recovery

Due on Tuesday, October 8th, 2024 at 2:00 pm (Week 2)

Question 1: Intersymbol Interference (20 points)

This is a small variation on your worksheet question on Intersymbol Interference. The figure below shows the response of a wire to an input **bit 1** of width **1 µs**. The response to both **slow and fast bits are the same**, which is an approximation to the $sinc$ function we studied in class. Unlike the worksheet, this approximation is a trapezoid and not a triangle.



The response:

- rises to **2 V** at **0.25 µs**,
- stays **constant** until **0.75 µs**,
- falls to **0 V** at **1 µs**,
- falls to **-1 V** at **1.25 µs**,
- rises to **0 V** at **1.5 µs**,
- and finally remains at **0 V**.

Assume that:

- a **1** is encoded as a **2 V** signal,
- a **0** is encoded as **0 V**, and
- the output to a **0 V** input of **any bit width** is also **0 V** for all time.

If at any sampling instant, the receiver measures a voltage of **1 V** or more, it assumes it received a **1**. Otherwise, it assumes it has received a **0**. The sender will now send a bit pattern of **101**.

Questions

1. **Slow Bits** (6 points)
   The sender sends bits at the slow rate of once every **1 µs**: 3 bits at times **0**, **1**, and **2 µs** respectively.
   The sampling instants the receiver uses are **0.5**, **1.5**, and **2.5 µs** respectively for the 3 bits.
   At any sampling instant, the receiver measures the output voltage as the sum of the voltage values of the output of the 3 bits.
   Write down the measured outputs. What bits do the receiver output?

2. **Fast Bits** (6 points)
   Follow the same instructions as in part 1, but now the sender sends its 3 bits at times **0**, **0.5**, and **1 µs** respectively where the bit width is now **0.5 µs** as opposed to **1 µs**.
   The sampling instants the receiver uses are **0.5**, **1**, and **1.5 µs** respectively for the 3 bits.
   Write down the measured outputs. What bits do the receiver output?

3. **Supersonic Bits** (6 points)
   Follow the same instructions as in part 1, but now the sender sends its 3 bits at times **0**, **0.25**, and **0.5 µs** respectively where the bit width is now **0.25 µs** as opposed to **1 µs**.
   The sampling instants the receiver uses are **0.5**, **0.75**, and **1 µs** respectively for the 3 bits.
   Write down the measured outputs. What bits do the receiver output?

4. **Sensitivity to Shape** (2 points)
   Based on your knowledge of the $sinc$ function, the Worksheet 2 problem, and this problem, what do these results suggest about the “shape” of the output wave required to transmit at the Nyquist limit of **2B**.

Question 2: Clock Recovery (25 points)

The first problem taught you output distortions can cause intersymbol interference if you send too fast. However, we assumed in Question 1 that the sender and receiver clocks were perfectly synchronized. In reality, they are not and we need clock synchronization.

In Question 2, you will simulate the effect of clock recovery on some bits sent using **4-5 encoding** using the clock synchronization algorithm below. Assume a perfect channel, no distortion.

Assume the preamble has been received, and the receiver is basically in sync—except for possible clock drift. Thus, the receiver is sampling according to its current clock (see figure below) and should be expecting transitions only at what it thinks are bit boundaries (see dotted lines in figure).

However, because of clock drift the actual transitions may be a little off (see the solid line in the figure below). Remember that in 4-5 coding, you are guaranteed to get at least **one transition in every 5 consecutive bits;** however, you may get **up to 5 transitions**. Pseudocode for the receiver clock recovery algorithm is as shown on the next page.

Please run the code shown assuming a nominal bit time of 1 **µs** (e.g., **T = 1 µs**) and a sender who is sending **7% slower than the receiver**. (The sender sends his first bit from **0** to **1.07**, the second bit from **1.07** to **2.14**, the third from **2.14** to **3.21**, etc.) Without doing any clock recovery or lag adjustment, the receiver would sample at what it thinks is the middle of a bit; e.g. at **0.5**, **1.5 µs**, and so on.

We would like to see what happens on the ten-bit sequence **0101110010** *with and without clock recovery*.

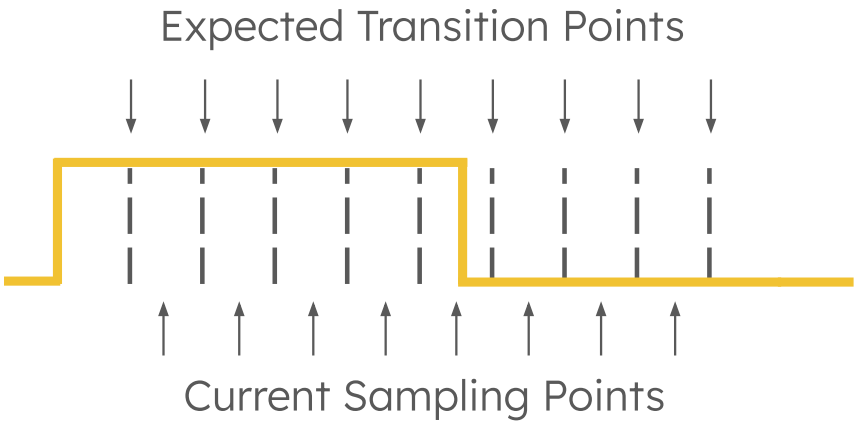
Assume a **0** is encoded as **0 V** and a **1** as **1 V**.

Clock Recovery Code

```
T: real constant; // nominal time to send a bit, input to program
P: real; // predicted next time at which a transition may occur
A: real; // actual real time at which a transition occurs
lag: real; // difference between predicted and expected

// After preamble is detected
Initialize
Clock = 0
lag = 0
P = 0
StartTimer(T/2)
Wait(TimerExpiry)
```

```
Do until end of frame
      Output (Sample Signal) // Output sampled value when timer expires
      P = P + T + lag
      StartTimer (T + lag)
      Wait(TimerExpiry)
      In parallel with Wait, look for Transition if any
      If Transition is detected at actual time A
            lag = A - P // difference between real and predicted
End
```



*not the same signal; just for demonstration

Questions

1. Using a drawing tool, draw a waveform of the 10 bits sent by the sender. Explicitly label the start and end times of each bit, as well as their ideal sampling points. (2 points)
2. If the receiver does not run the clock recovery code, how far off is the sampling by the **10th bit**? (2 points)
3. Using the pseudocode above for clock recovery and assuming there is no noise, compute the sampling instants and values of lag for each bit. (17 points)
4. Now suppose there is a sharp noise spike of **1 V** at time **0.3 µs**. How would it affect your sampling times? Provide an intuitive explanation. No need to recompute the sampling times. (2 points)
5. Finally, suppose there is a sharp noise spike of **1 V** at time **8.2 µs**. How would it affect your sampling times? Provide an intuitive explanation. No need to recompute the sampling times. (2 points)