Homework 1

Tejas Kamtam

305749402

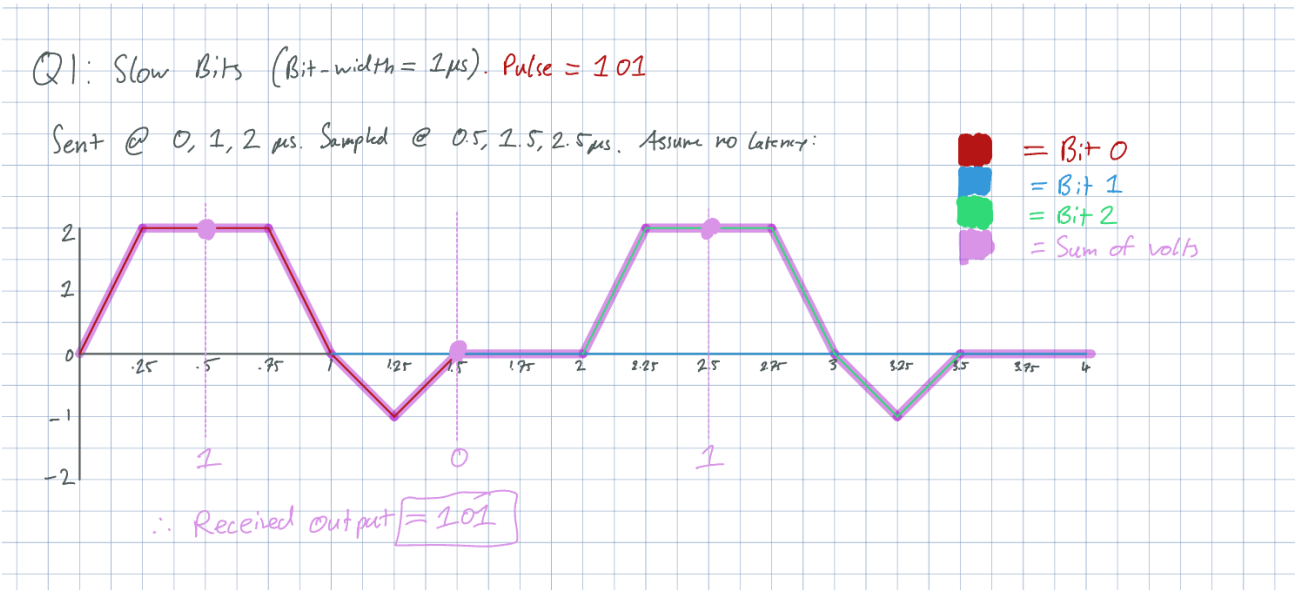
CS 118, Section 1C

Question 1 - Intersymbol Interference (ISI)

Assume no latency. Assume synchronized clocks.

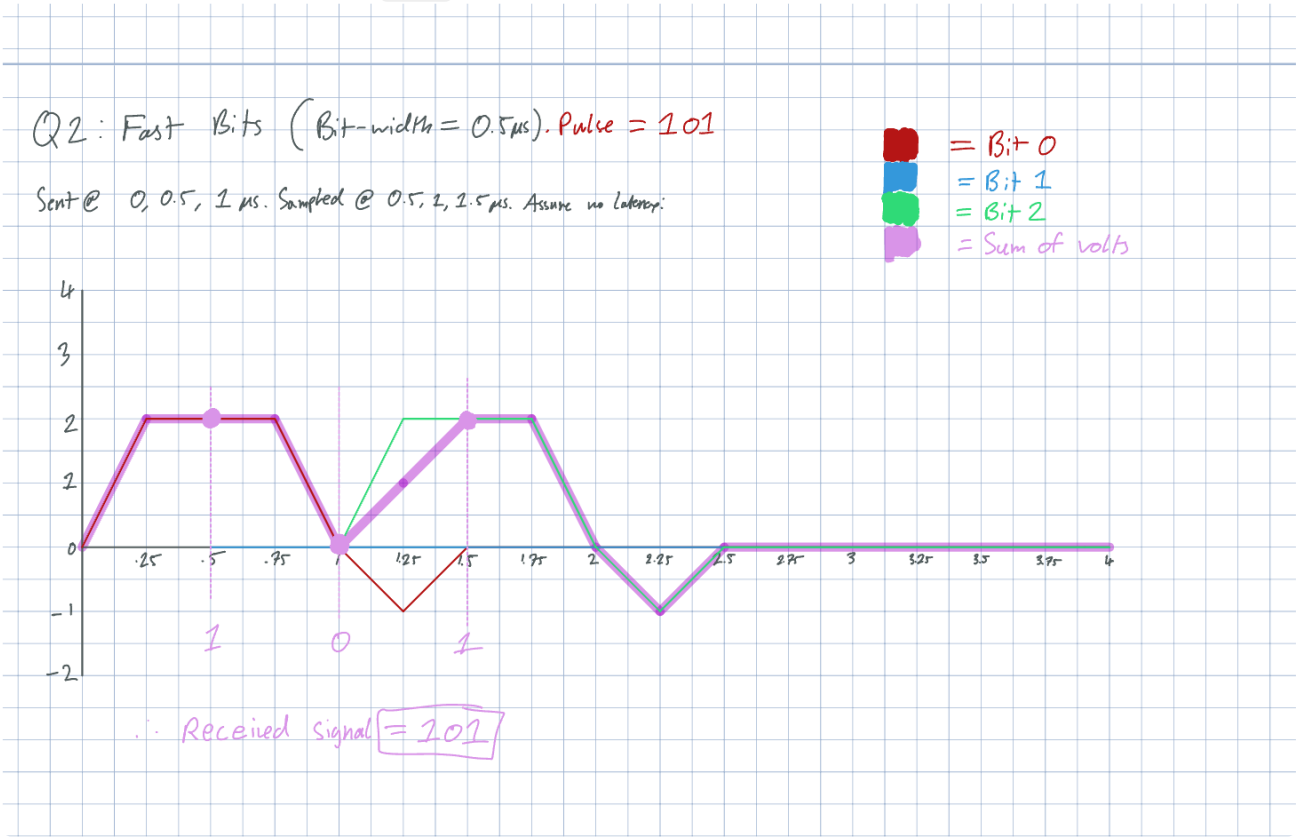
Q1.1 - Slow Bits

Received signal: `101`



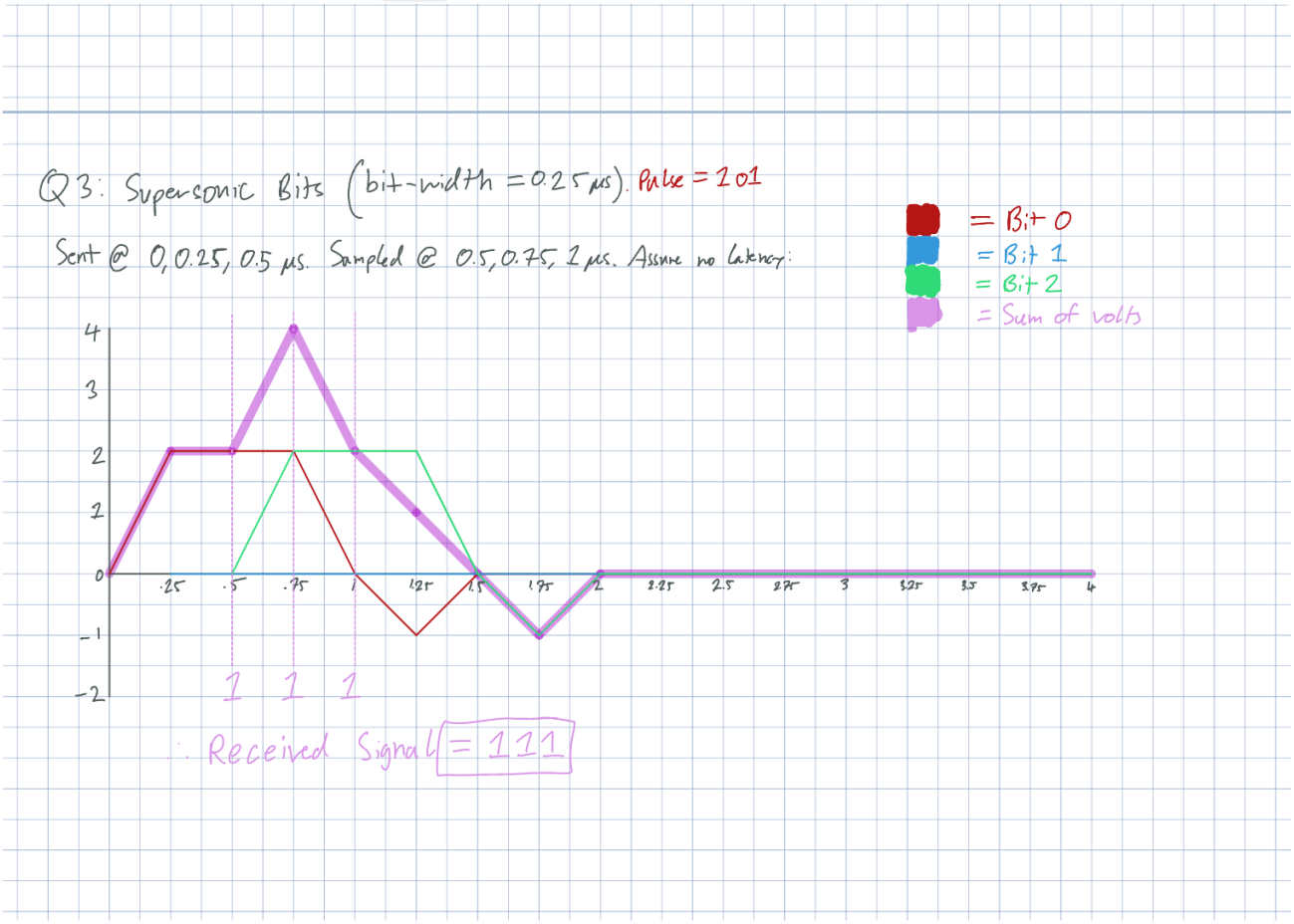
Q1.2 - Fast Bits

Received signal: `101`



Q1.3 - Supersonic Bits

Received Signal: `111`

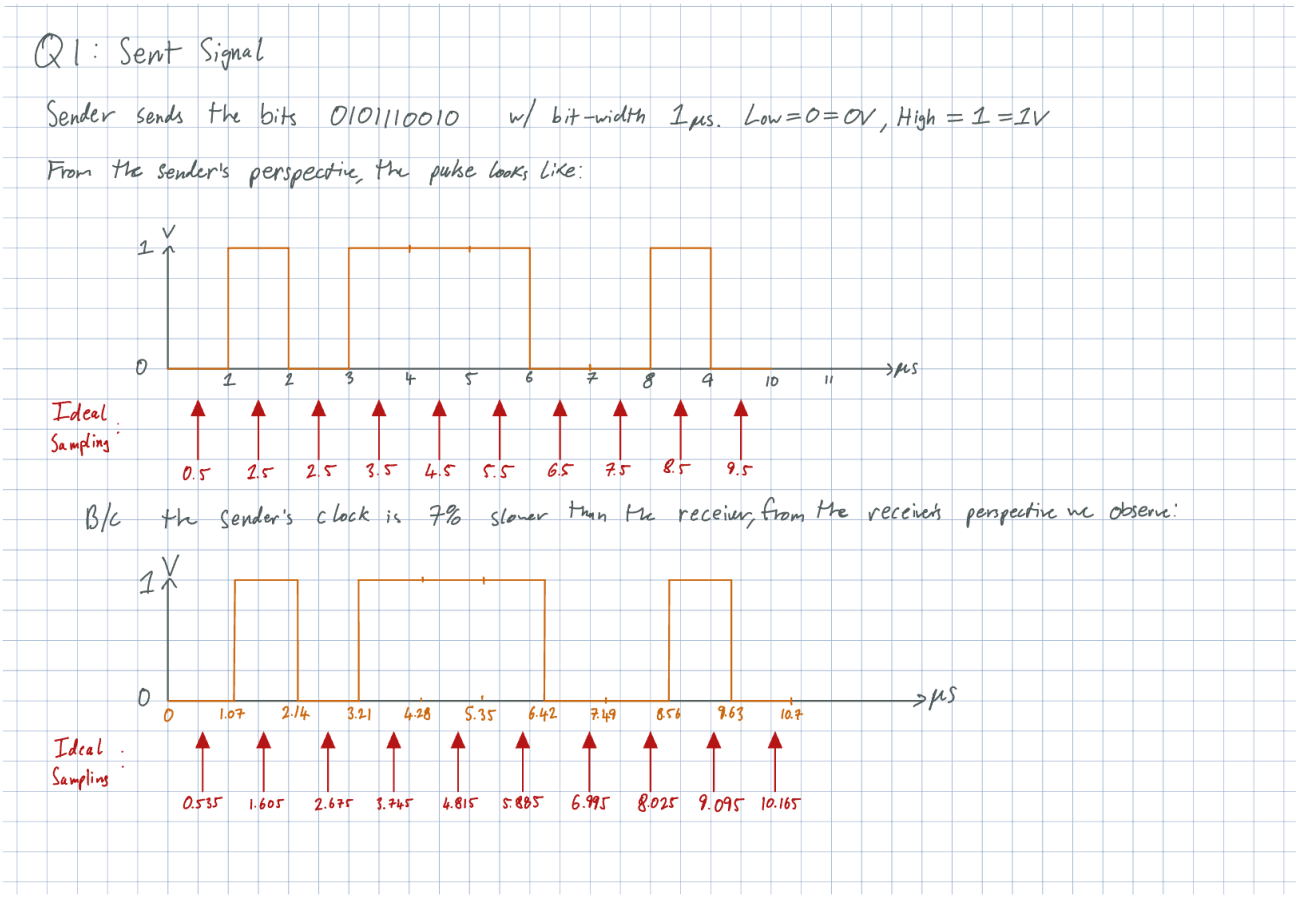


Q1.4 - Sensitivity to Shape

Based on the output signals received from the same pulse across Worksheet 2 and this assignment, we can see that the shape of the receiver response matters little in comparison to the "speed" of the bits (the bit-width). From the previous examples, we can see that the bottleneck to transmit at the Nyquist limit of $2B$ is dependent on the bit-width being large enough to avoid ISI.

Question 2 - Clock Recovery

Q2.1 - Waveform Diagram



Q2.2 - Sampling Shift

From the visual in Q2.1, we can see that, by the 10th bit, the ideal sampling time has shifted by $0.665\,\mu s$.

Q2.3 - Clock Recovery

Given signal: `0101110010`, run the clock recovery code:

Bit Pos.	Sampling Instant (μs)	Value	P	Timer	A	Lag
0	0.5	0	1	1	1.07	0.07
1	1.5	1	2.07	1.07	2.14	0.07
2	2.57	0	3.14	1.07	3.21	0.07
3	3.64	1	4.21	1.07	4.28	0.07
4	4.71	1	5.28	1.07	5.35	0.07
5	5.78	1	6.35	1.07	6.42	0.07
6	6.85	0	7.42	1.07	7.49	0.07

Bit Pos.	Sampling Instant (μs)	Value	P	Timer	A	Lag
7	7.92	0	8.49	1.07	8.56	0.07
8	8.99	1	9.56	1.07	9.63	0.07
9	10.06	0	10.63	1.07	10.7	0.07

Q2.4 - Noise Spike @ 0.3μs

A noise spike of 1V at time 0.3μs would be inconsequential. This is because our first sample is taken at time 0.5μs, at which point, we haven't begun checking for transitions, so our lag would still be the same as it is in Q2.3.

Q2.5 - Noise Spike @ 8.2μs

Unlike Q2.4, a noise spike of 1V at time 8.2μs would cause a change in sampling as this noise spike would introduce a new perceived transition/edge which would preemptively update the lag for that bit. This wouldn't cause the next timing to change, but the timing after to occur sooner. This is because the timer had already been started for the next sample before the transition was detected, so the new/updated lag would be used in the timer for the following instance.