

Lab 1:

The purpose of this lab is to familiarize ourselves with the SerDes Designer in MATLAB. To do this, we looked at simulating an eye, increasing losses, and opening the eye by adding a feed-forward equalizer (FFE). As shown in figures 1 through 3, the 5 GHz default signal with -10dB of loss still has a small eye. When the losses are increased to -20dB (see figures 4-6), the eye is closed. By adding a FFE, I am able to open the eye such that the -20dB signal makes it to the receiver in a better state than even the -10dB signal without a FFE (see figures 7 through 9). The tap weights for my FFE were [-.07 1 -.57 0 0 -.02].

-10dB loss w/o FFE

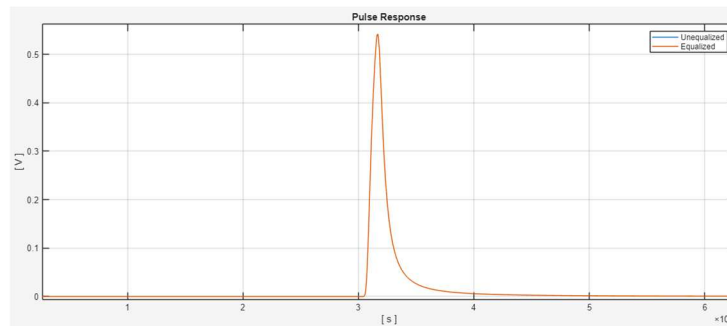


Figure 1: SBR of the default 5 GHz signal with 10dB of losses.

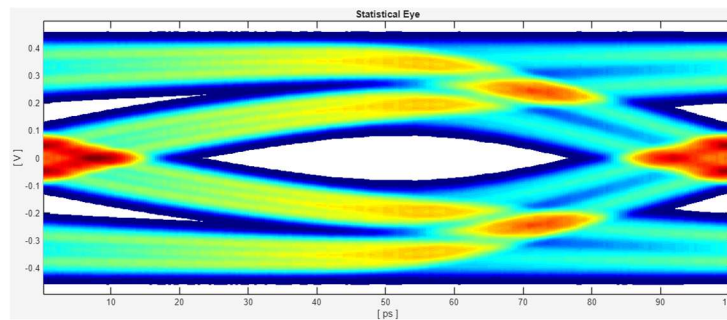


Figure 2: Eye diagram of the default 5 GHz signal with 10dB of losses. It is barely open.

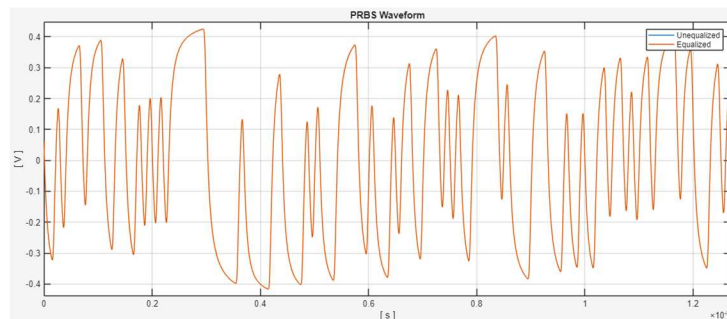


Figure 3: PRBS of the default 5 GHz signal with 10dB of losses.

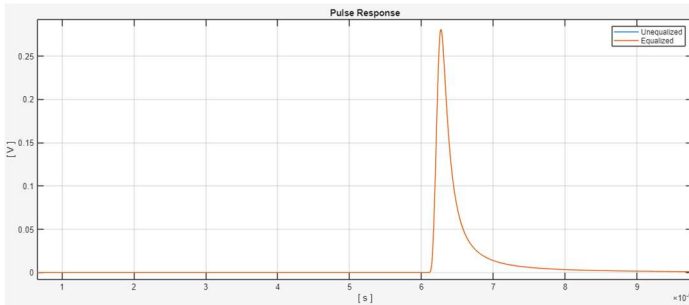


Figure 4: SBR of the default 5 GHz signal with 20dB of losses.

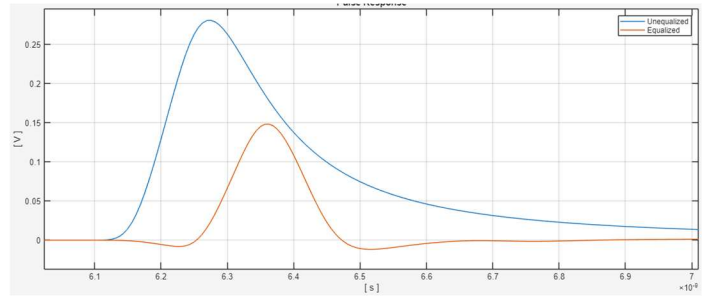


Figure 7: SBR of the default 5 GHz signal with 20dB of losses and a feed-forward equalizer.

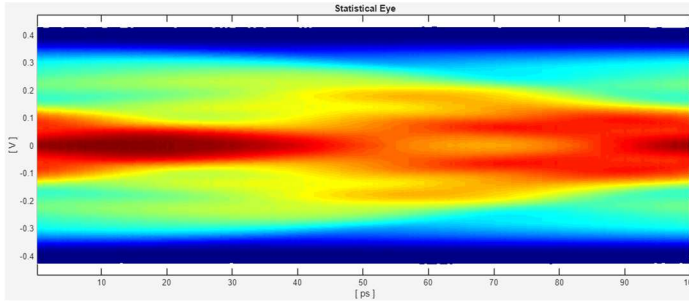


Figure 5: Eye diagram of the default 5 GHz signal with 20dB of losses. It is completely closed.

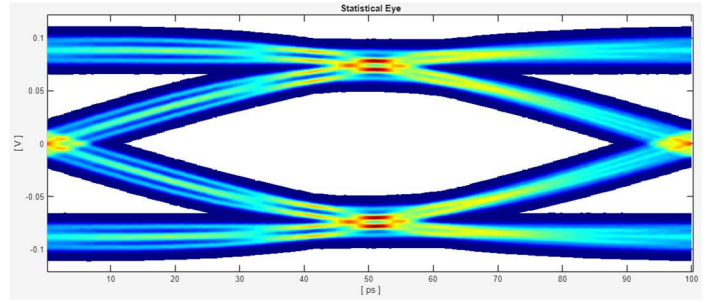


Figure 8: Eye diagram of the default 5 GHz signal with 20dB of losses and a feed-forward equalizer. It is open more than the condition with 10dB of losses and no FFE.

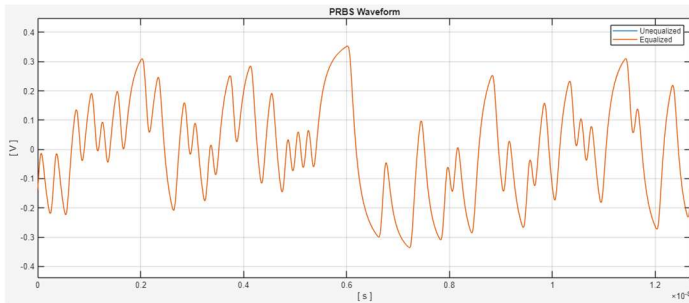


Figure 6: PRBS of the default 5 GHz signal with 20dB of losses.

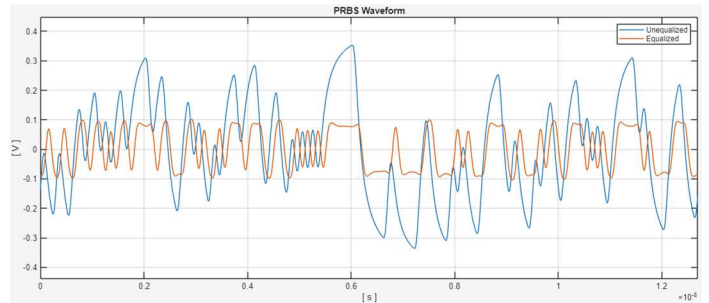


Figure 9: PRBS of the default 5 GHz signal with 20dB of losses and a feed-forward equalizer. The bits are much flatter.

So what:

It is clear that a feed-forward equalizer can have an enormous impact on opening up an eye. One of the drawbacks is that the amplitude of the eye decreases, making it more suitable for differential signals than single-ended signals. Something interesting that I found was that a bit of compensation before the bit was sent signal really tightened up the bit width and opened up the eye. I also found it interesting that adding some compensation after a few bits of 0s actually was helpful in opening the eye.

Lab 2:

The purpose of this lab was to utilize the skills built in Lab 1 and apply them to measured S-parameters to simulate a real scenario where we might need to program a FFE based on measured transmission characteristics. I simulated two sets of S-parameters — one with a stub and one without a stub — and I compared the minimum bit width for each scenario.

For the no-stub situation, the minimum bit width I could achieve with a wide open eye was 100 ps. The situation with a stub performed worse, and I was only able to maintain a wide open eye with a bit width of 150 ps.

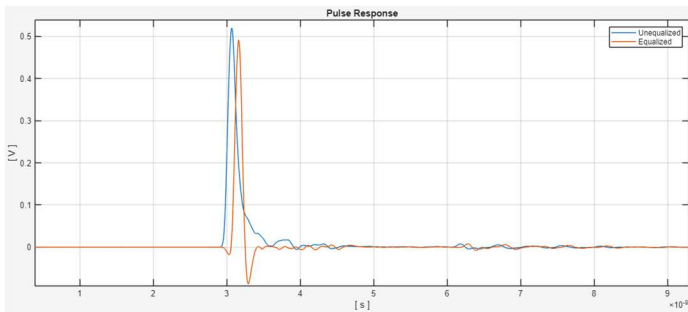


Figure 1: SBR for the no-stub case with a FFE and a symbol time of 100 ps.

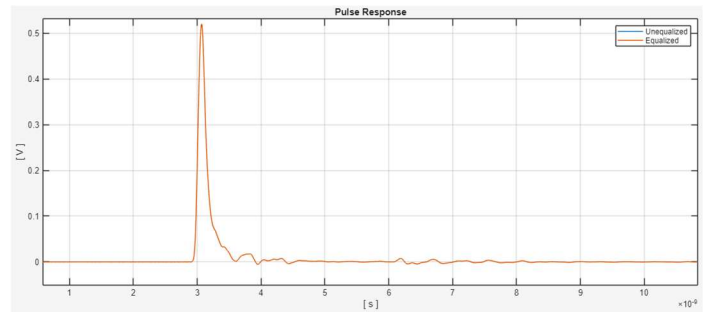


Figure 2: SBR for the no-stub case without a FFE and a symbol time of 100 ps.

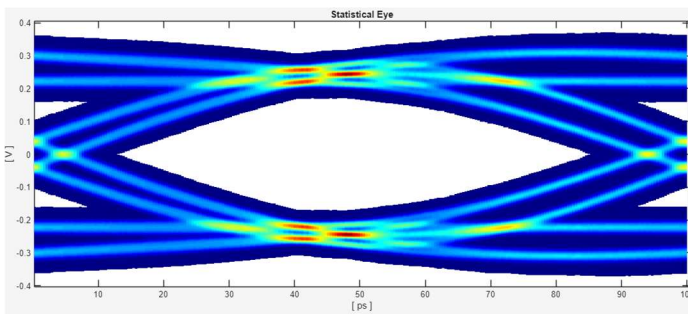


Figure 3: Eye diagram for the no-stub case with a FFE and a symbol time of 100 ps.

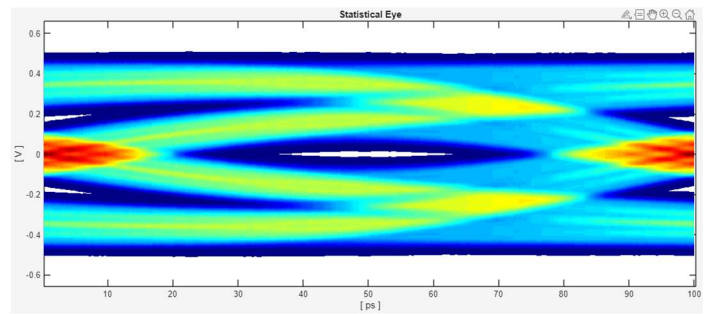


Figure 4: Eye diagram for the no-stub case without a FFE and a symbol time of 100 ps.

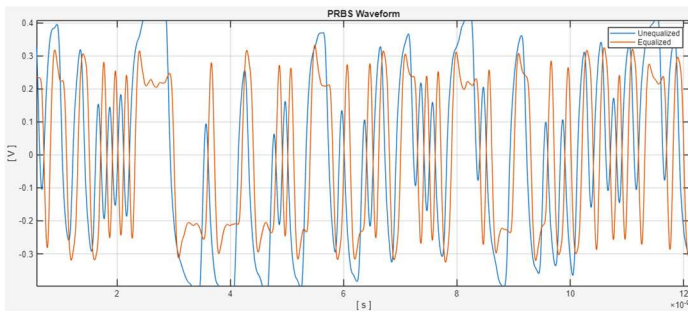


Figure 5: PRBS for the no-stub case with a FFE and a symbol length of 100 ps. The preshoot and overshoot to get a clear, open eye at the sample time is clearly visible.

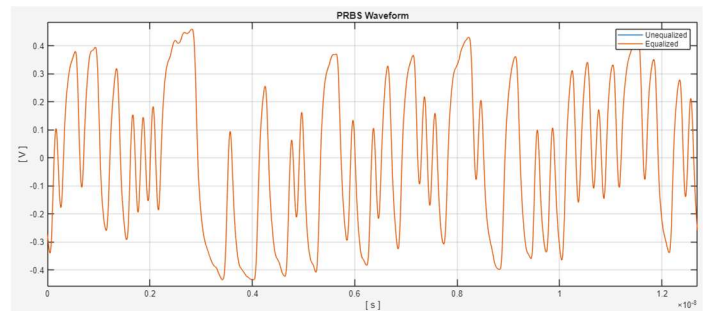


Figure 6: PRBS for the no-stub case without a FFE and a symbol length of 100 ps

From the SBR, I can tell that there are reflections since there is noise that appears far after the bit. In theory, the FFE could compensate for this, but I found that it had minimal impact so I did not compensate for those reflections for simplicity.

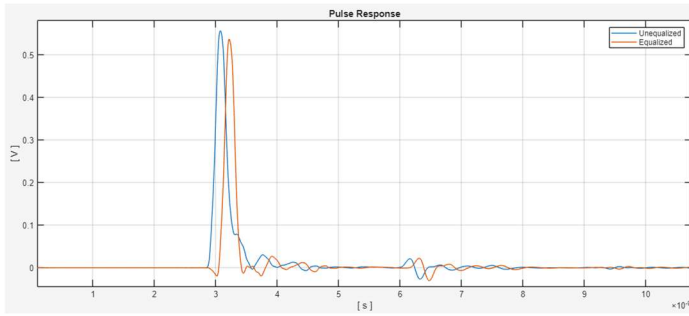


Figure 7: SBR for the stub case with a FFE and a symbol time of 150 ps.

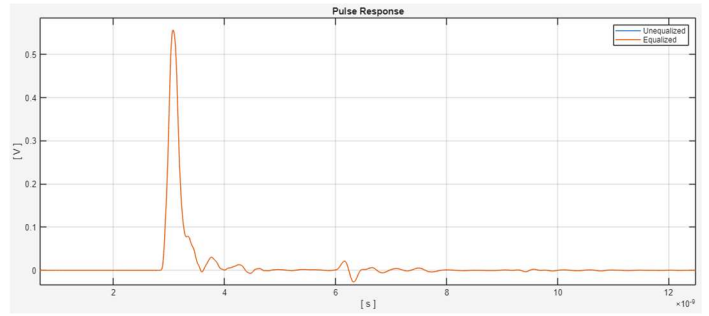


Figure 8: SBR for the stub case without a FFE and a symbol time of 150 ps.

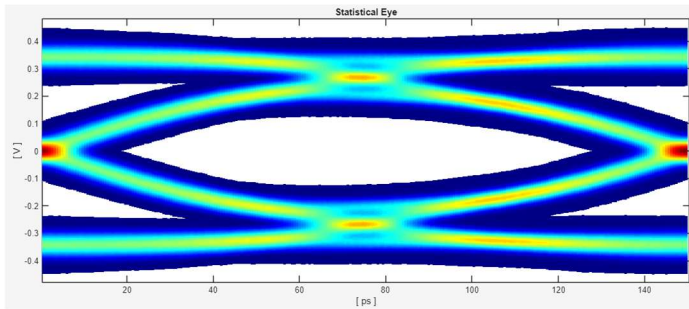


Figure 9: Eye diagram for the stub case with a FFE and a symbol time of 150 ps.

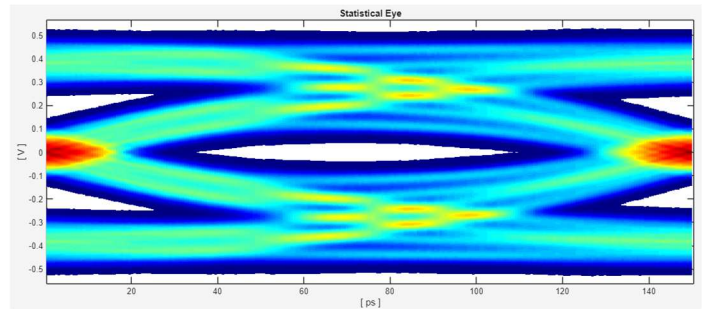


Figure 10: Eye diagram for the stub case without a FFE and a symbol time of 150 ps.

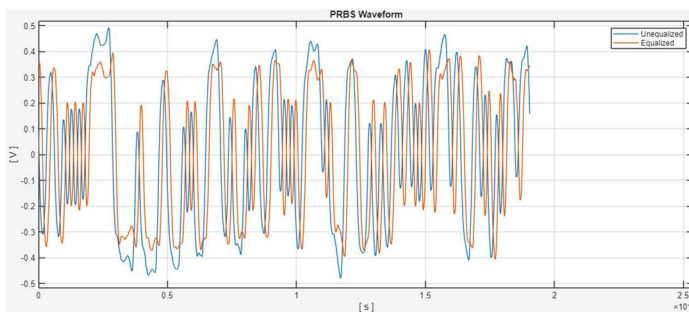


Figure 11: PRBS for the stub case with a FFE and a symbol length of 150 ps. The preshoot and overshoot to get a clear, open eye at the sample time is clearly visible.

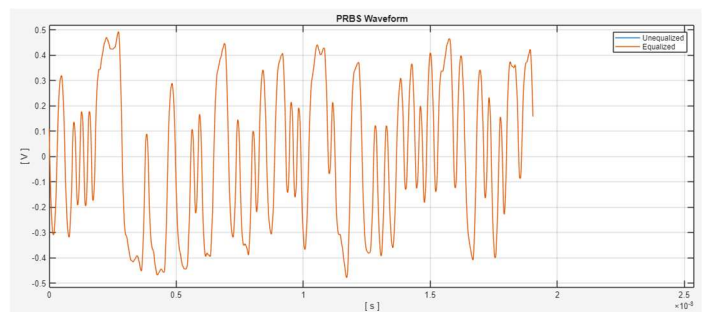


Figure 12: PRBS for the stub case without a FFE and a symbol length of 150 ps

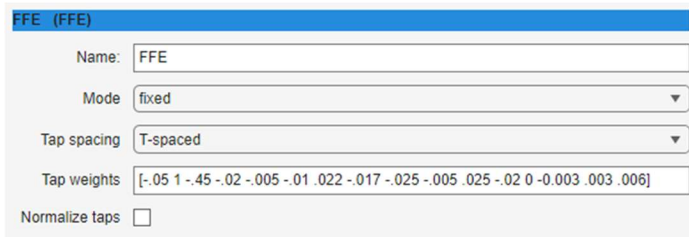


Figure 13: FFE weights for the no-stub case.

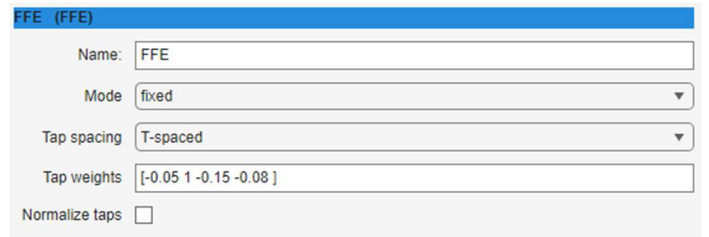


Figure 14: FFE weights for the stub case.

So what:

I was able to get a shorter bit width in the case without the stub, suggesting that stubs contribute to losses and noise. I also learned that FFEs can help with noise far after a bit is sent (like with reflections on mismatched lines), but there are diminishing returns. The most important bits to compensate are the bits directly before and after the bit of interest.