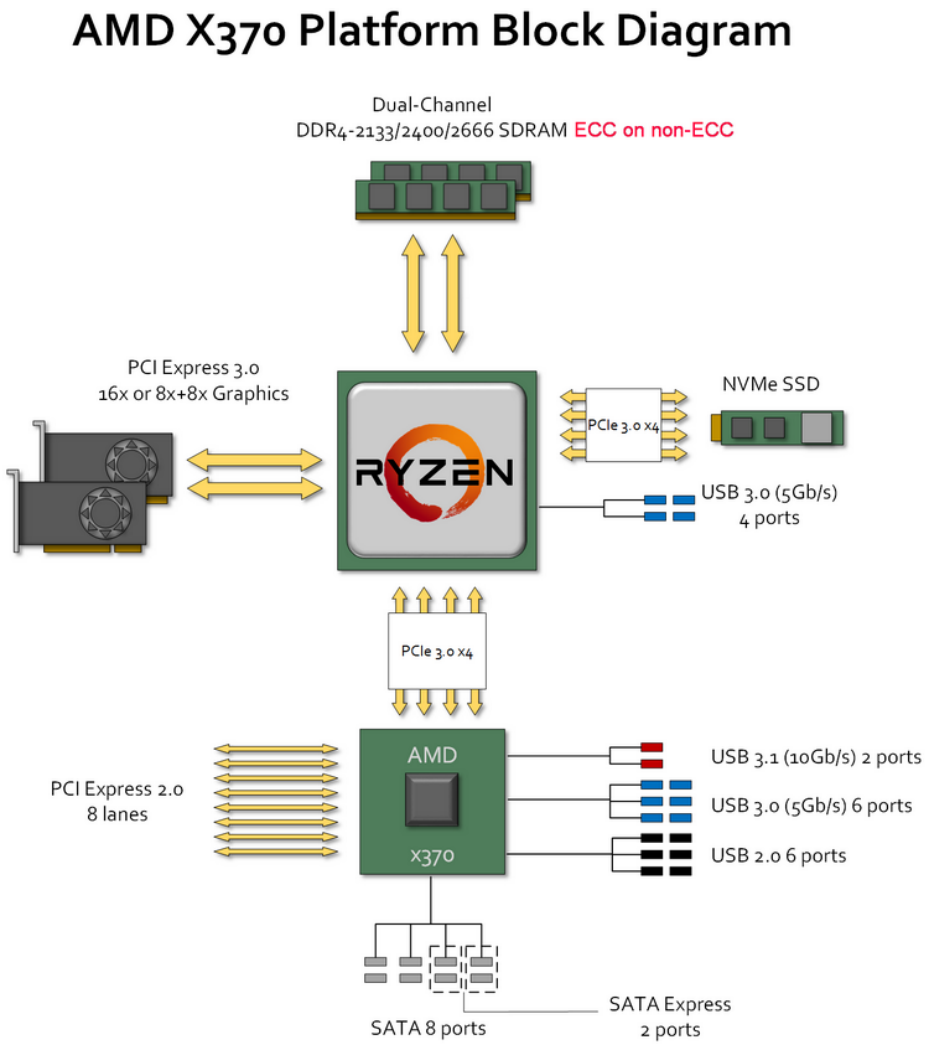
Modelling of High Speed Serial Systems, with an Emphasis on PCIe 6.0

Joseph Shaker

December 11, 2022

Background - Motherboard Block Diagram

- PCIe is a core protocol used in computer motherboards



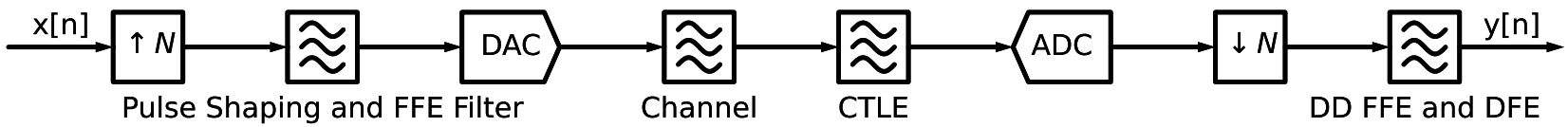
Background - History of PCIe Rates

- PCIe has been pushing the envelope of speed for years
- Today we will focus on PCIe 5.0 and 6.0, which both run at 32GSym/s, but 6.0 achieves double data the rate relative to 5.0 by using PAM-4
- Note that the lane throughput here ignores protocol overhead, and is just the raw rate
- PCIe can use up to 16 lanes to improve throughput

Version	Introduced	Modulation	Symbol Rate	Lane Throughput
1.0	2003	NRZ	2.5GSym/s	2.5Gb/s
2.0	2007	NRZ	5.0GSym/s	5.0Gb/s
3.0	2010	NRZ	8.0GSym/s	8.0Gb/s
4.0	2017	NRZ	16.0GSym/s	16.0Gb/s
5.0	2019	NRZ	32.0GSym/s	32.0Gb/s
6.0	2021	PAM-4	32.0GSym/s	64.0Gb/s

Problem Statement and System Block Diagram

- The transmitter converts some sequence of symbols to a sequence of voltages to transmit
- These are degraded through a channel transmission line, but have corrective equalization applied to them at the receiver
- The receiver can then make decisions on these equalized voltages to recover the symbols

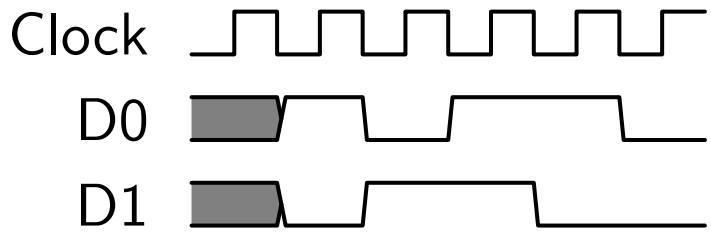


Serial vs Parallel Buses

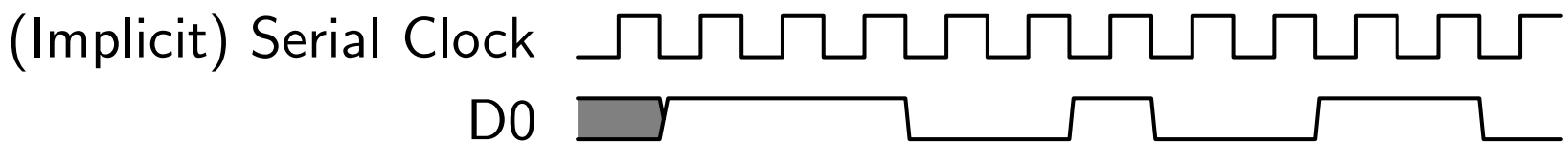
Property	Parallel Bus	Serial Bus
Clock Recovery	An explicit clock is transmitted along with the data lines	Clock times are inferred from DSP done on the data lines
Data Transition Density	No Requirement	Required by the receiver to maintain clock lock to the transmitter
Rate Limited by	Clock and Data Skew ($\approx$ 100MHz)	Physics of the conductors and sophistication of equalization ($\approx$ 32GHz)

Timing Diagrams for Serial and Parallel Buses

- In parallel buses an explicit clock is sent with the data



- In serial buses (such as PCIe) an implicit clock that the transmitter launches data on still exists, but it isn't explicitly sent so the receiver must recover it
- When there are multiple lanes, clock recovery is done independently by the receiver on each one



Serial Comms Fundamentals - Shannon-Hartley Theorem

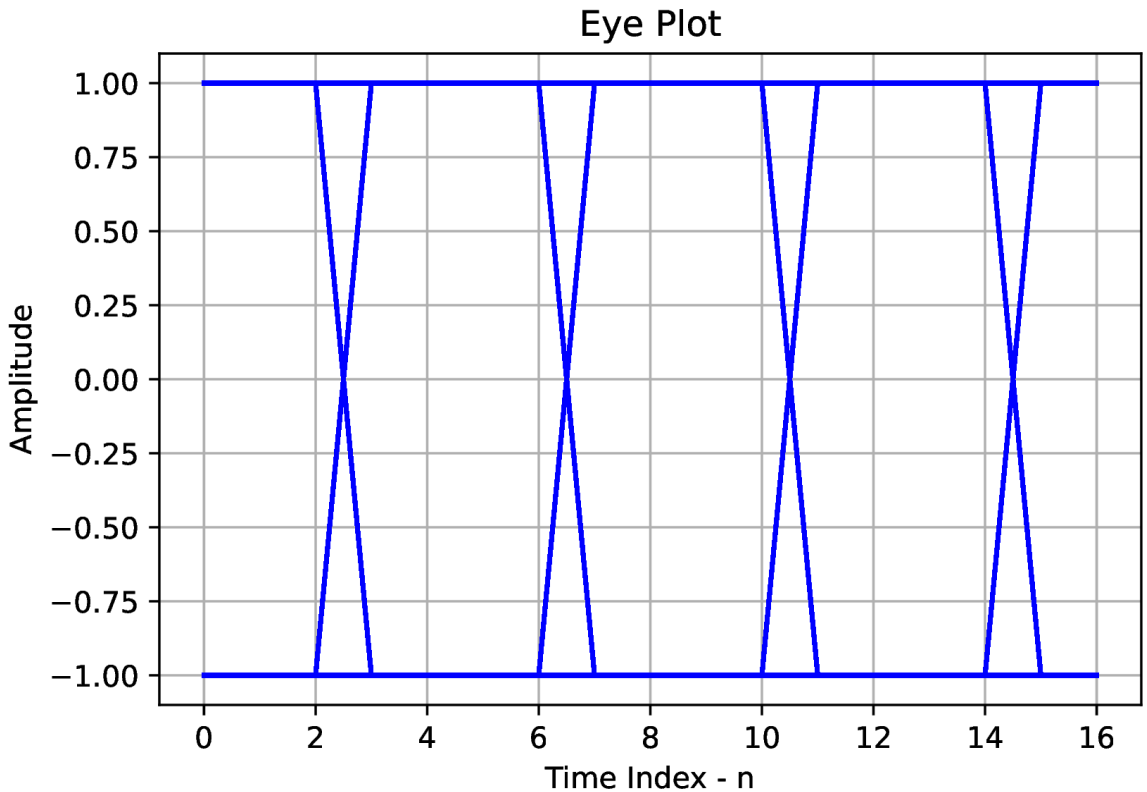
- Channel Capacity, C, is a function of Bandwidth, B, and Signal to Noise Ratio S/N

$$C = B \log_2 \left(1 + \frac{S}{N}\right)$$

- If $B > \log_2 S/N$, send NRZ waveforms at a high rate; noise limited
- If $\log_2 S/N > B$, send PAM-4, PAM-8, etc waveforms at a slower rate; bandwidth limited

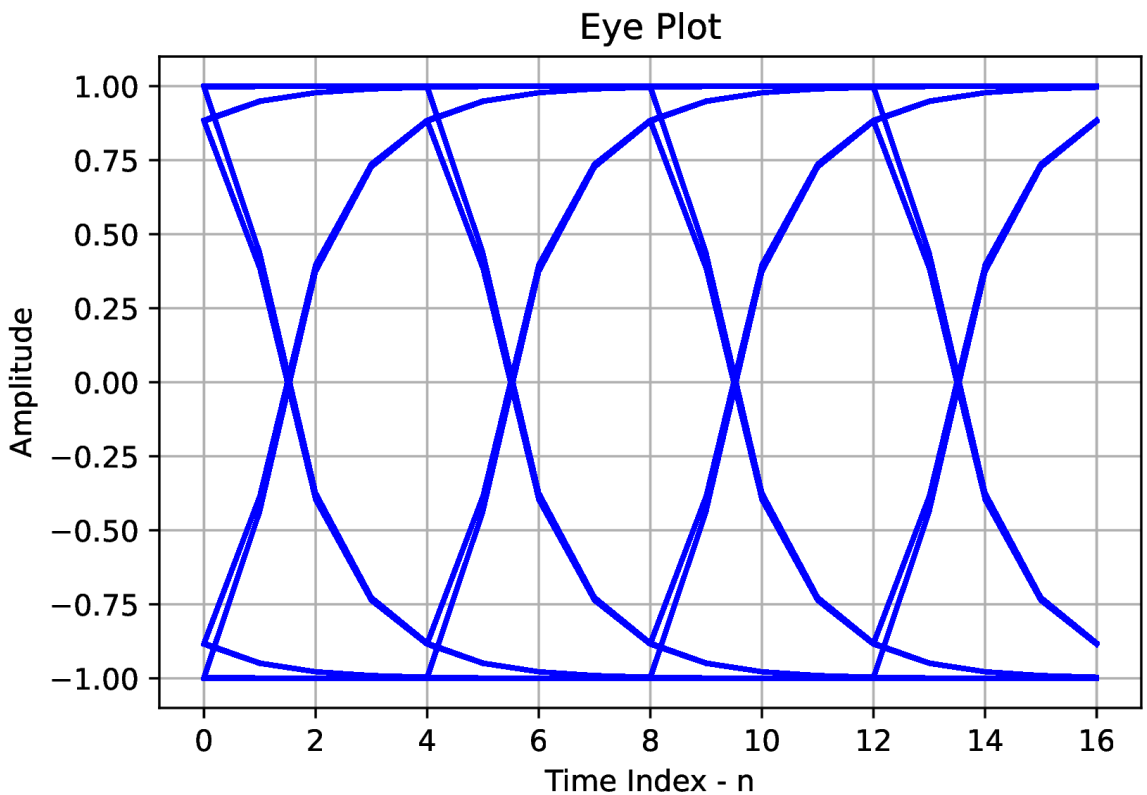
Serial Comms Fundamentals - Eye Diagrams

- Eye diagrams are generated by superimposing shifted segments of the signal on top of itself to study its trajectories.
- Below is an *ideal* eye - the transitions are short and the eye is completely open.



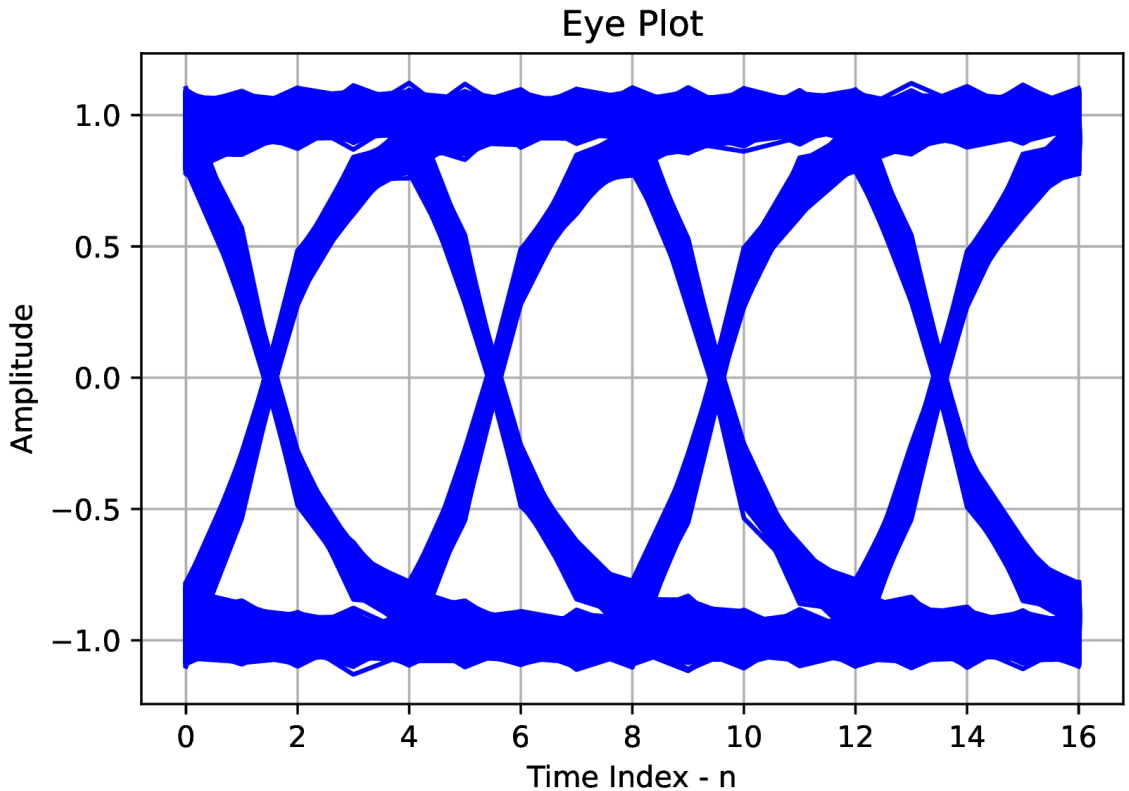
Serial Comms Fundamentals - Eye Diagrams

- This eye is more closed because the channel has a low pass filtering effect on the signal
- This causes the ideal sampling window to be narrower and closes the eye
- Of course, a lower cutoff frequency would impair the eye more



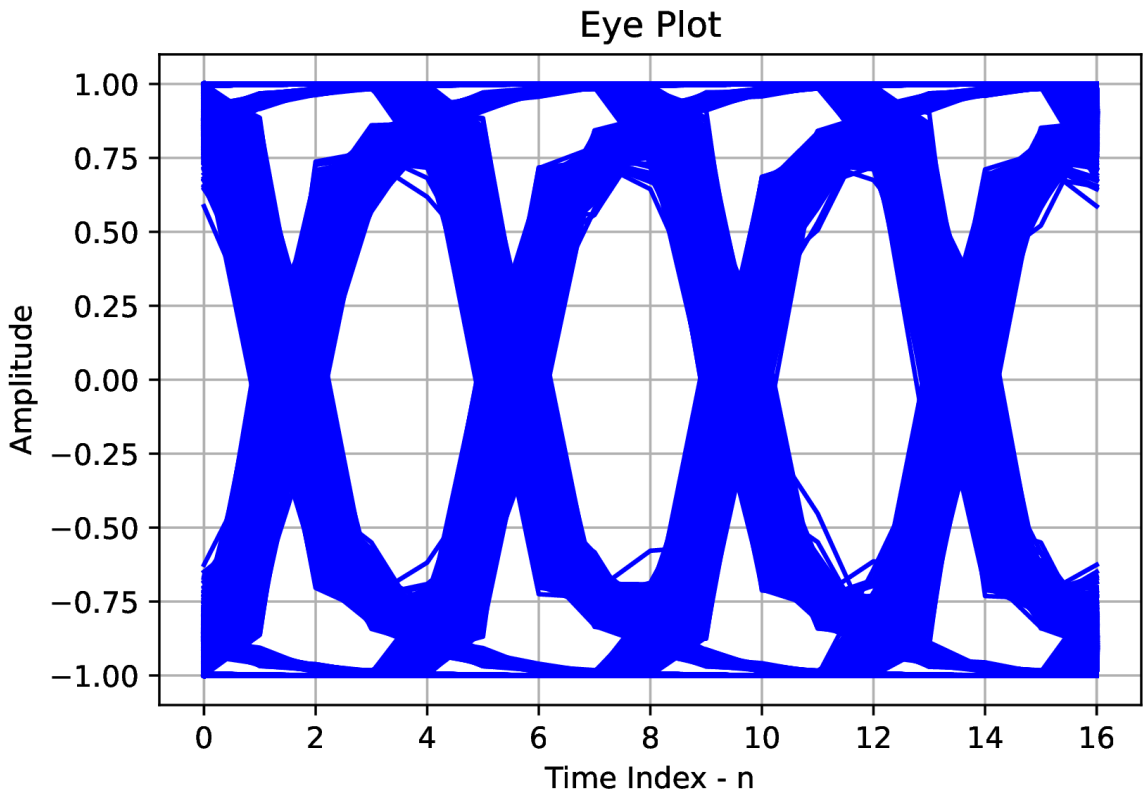
Serial Comms Fundamentals - Eye Diagrams

- This eye shows the effect of additive noise
- It closes the eye too



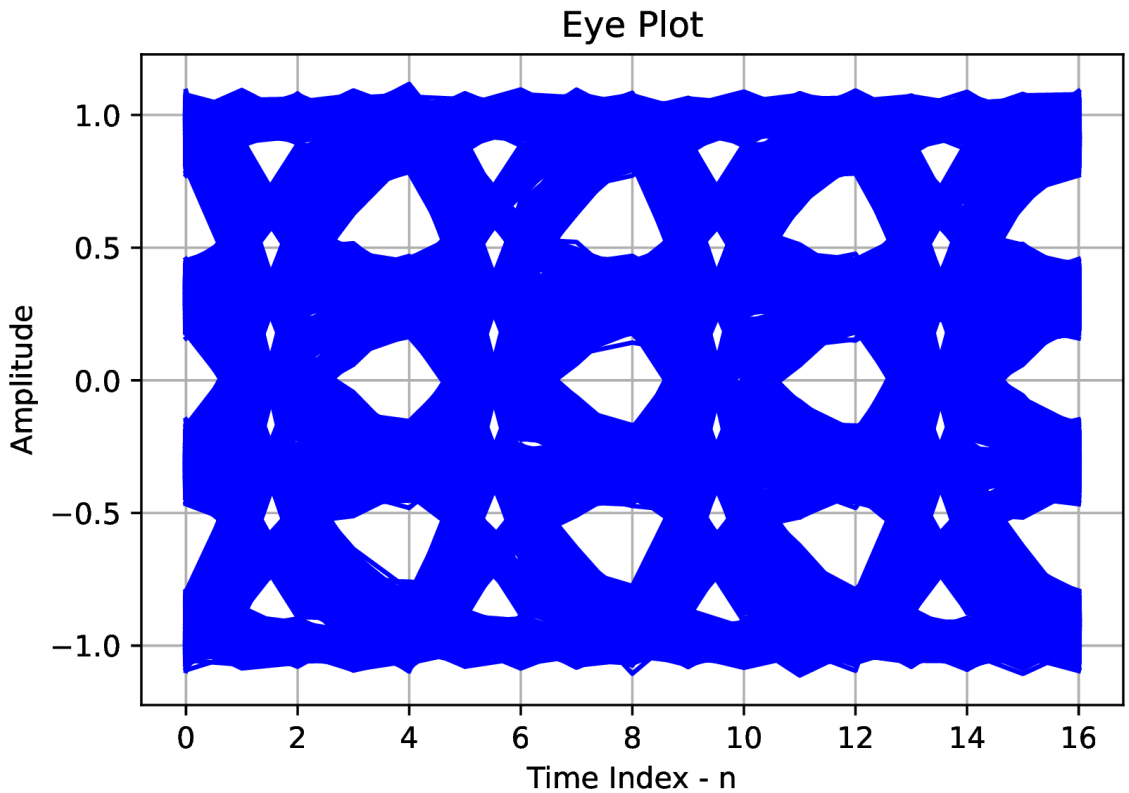
Serial Comms Fundamentals - Eye Diagrams

- This eye shows the effect of timing jitter on the waveform edges
- It narrows the sampling window



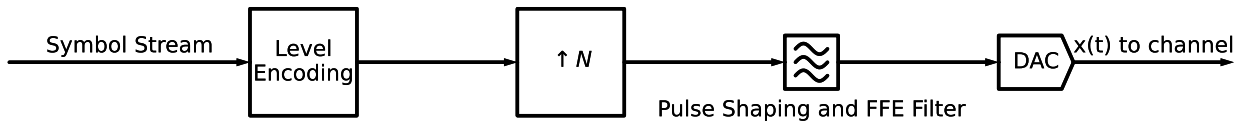
Serial Comms Fundamentals - Eye Diagrams

- PAM-4 requires 4 levels and is more sensitive to channel impairments
- Here we see an array of degradations on a PAM-4 signal

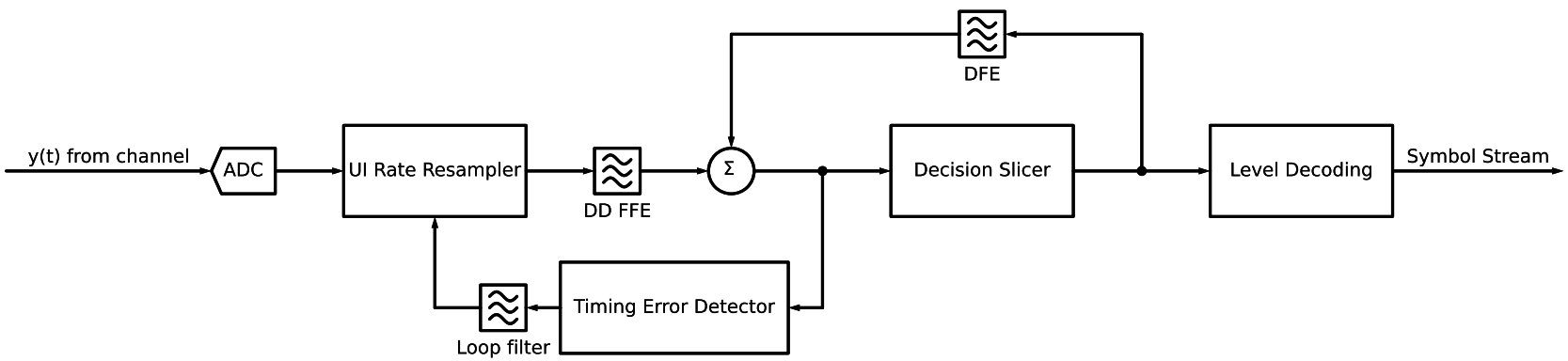


Serial Comms Fundamentals - Tx and Rx Structures

- Below is a block diagram of the transmitter, it is relatively simple

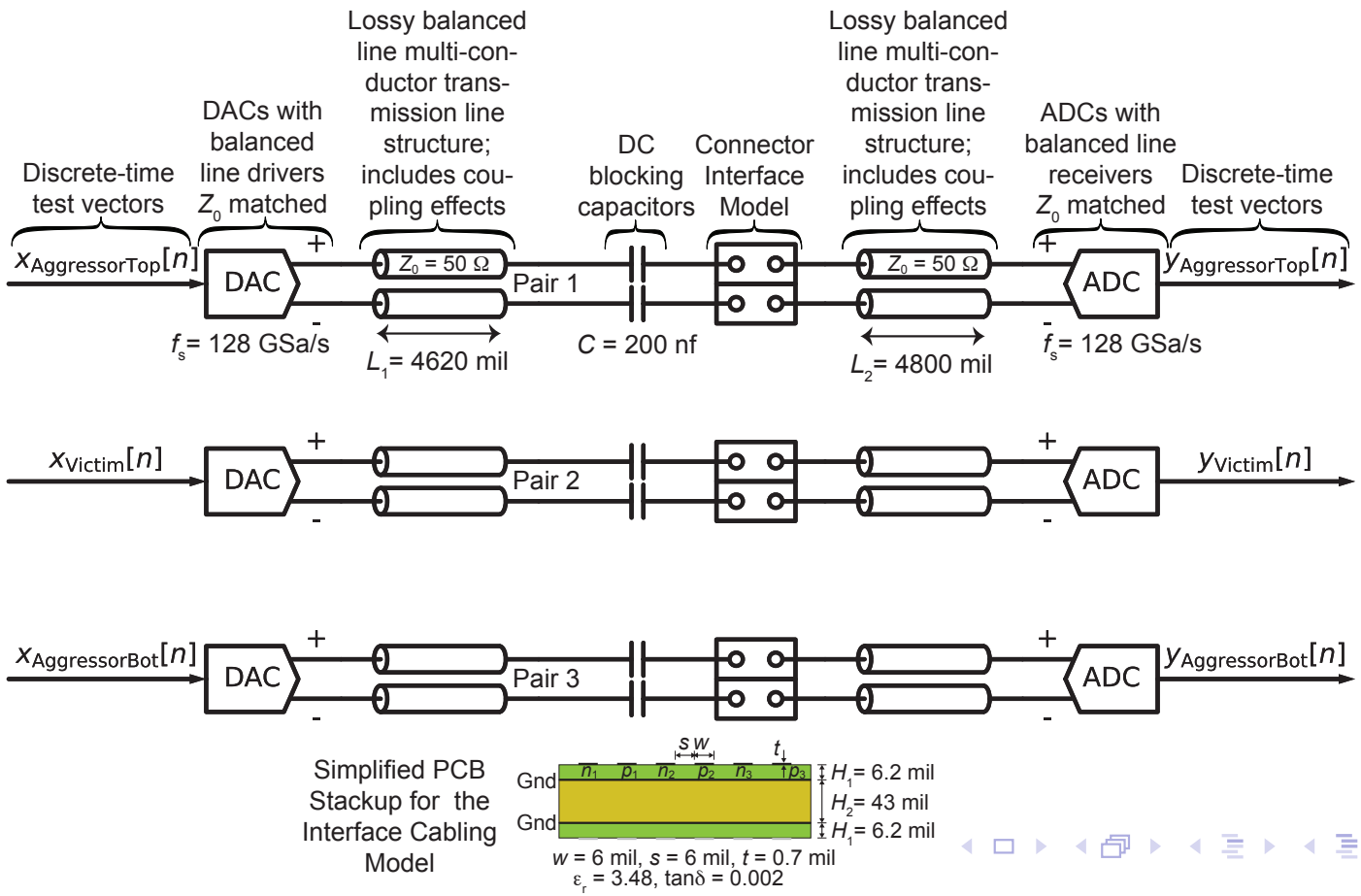


- Below is a block diagram of the complete receiver
- We see in particular subsystems for adaptive equalization and closed loop clock recovery



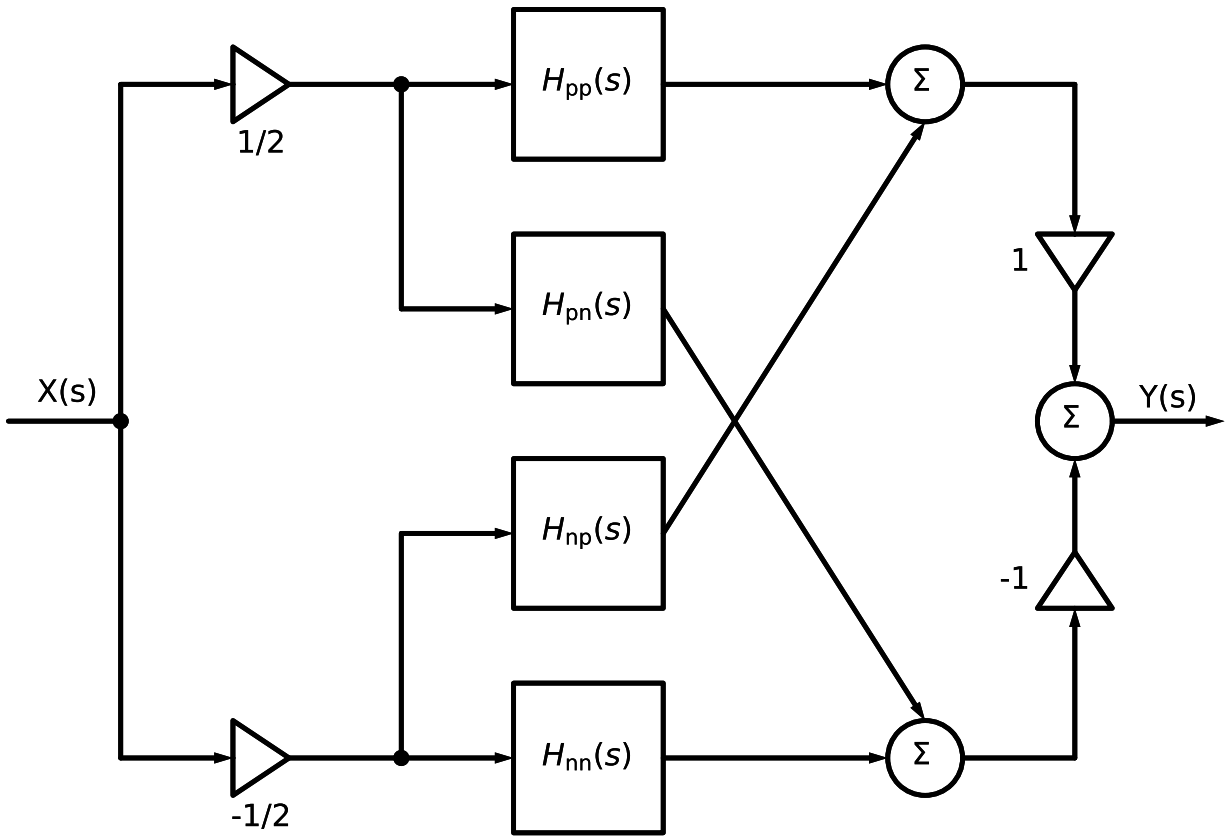
Channel Modeling

- Low loss (shown) & medium loss (with longer L_1) differential coupled transmission lines
- ADC and DAC are modeled as introducing AWGN
- The transmission lines are modeled in Keysight ADS using s-parameters



Channel Modeling - differential pairs

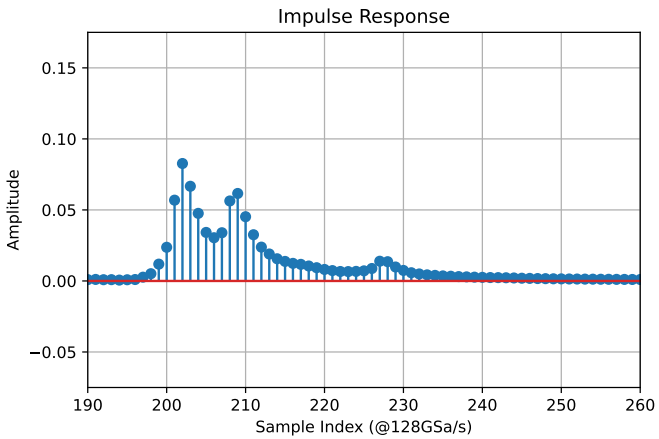
- This shows show to convert the model to single-ended system model

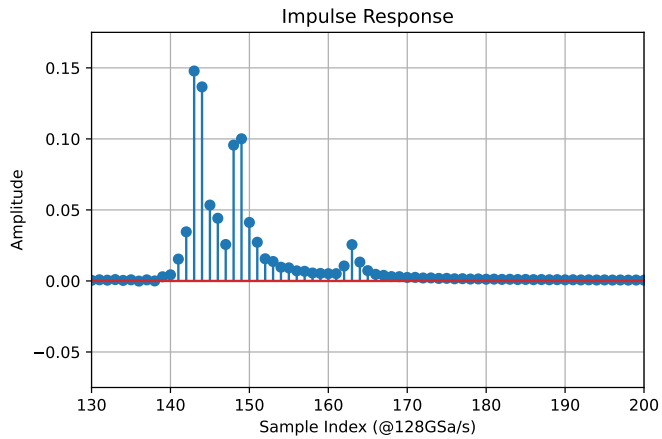


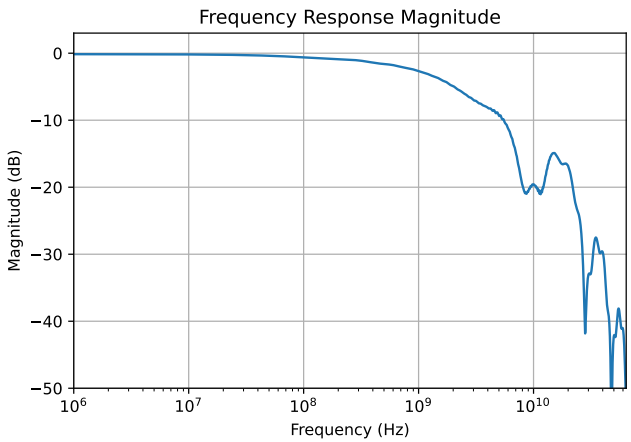
$$\frac{Y(s)}{X(s)} = \frac{1}{2}\left(H_{pp}(s) + H_{nn}(s) - H_{pn}(s) - H_{np}(s)\right)$$

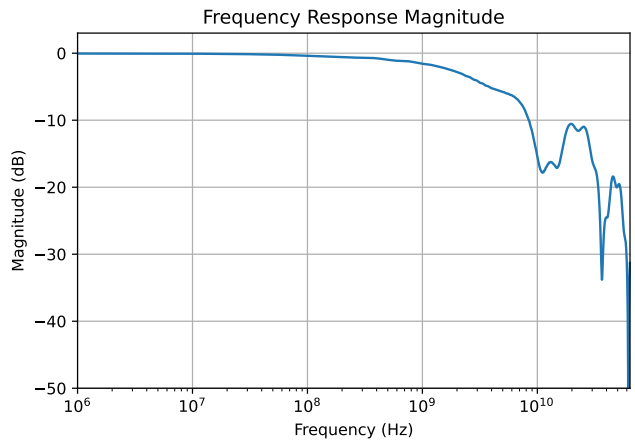
Channel Modeling - Characterization

- On the left is a *medium loss* channel modeled with a longer transmission line
- On the right is a *low loss* channel modeled with a shorter transmission line
- Note that at 32GSym/s, both's impulse responses span many UI



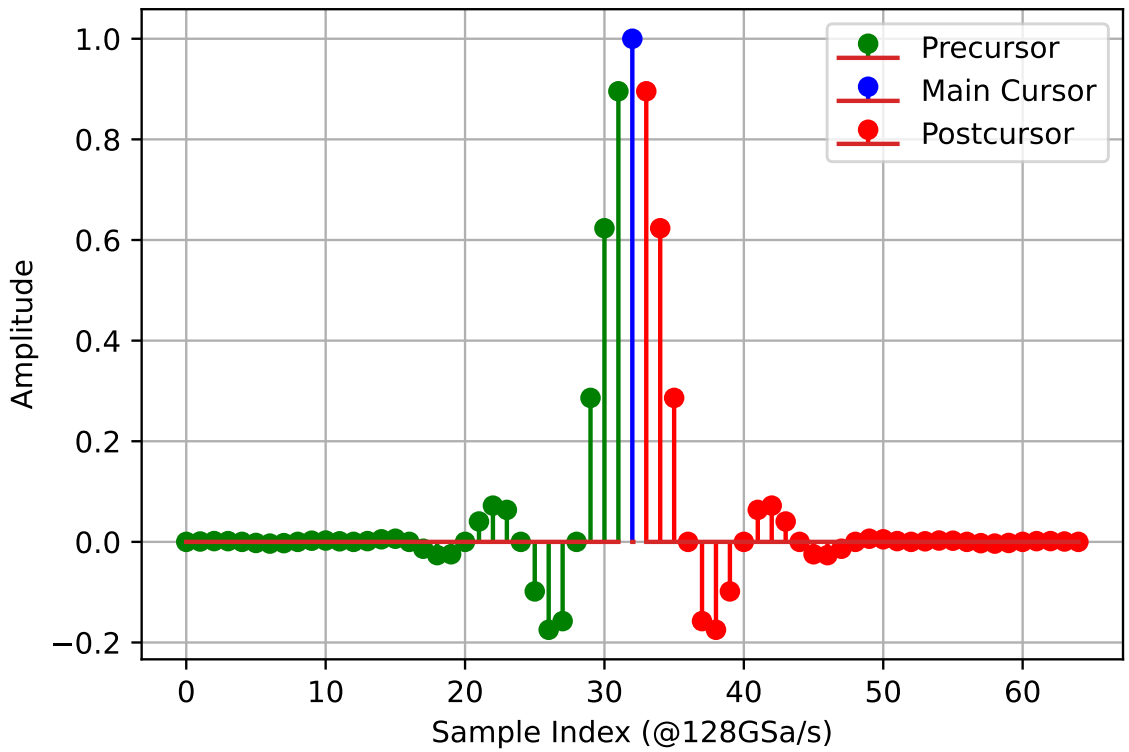






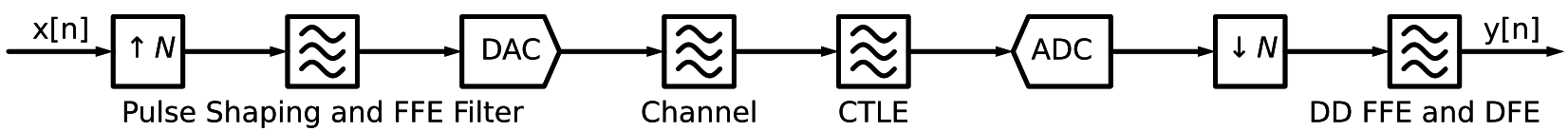
Equalization and Intersymbol Interference (ISI)

- ISI is the effect of a current symbol on past or future symbols, and can be seen in its pulse response.
- The goal of equalization is to cancel ISI at all UI period multiples, ie below see there is no ISI at multiples of 4 samples from the main cursor in this idealized response



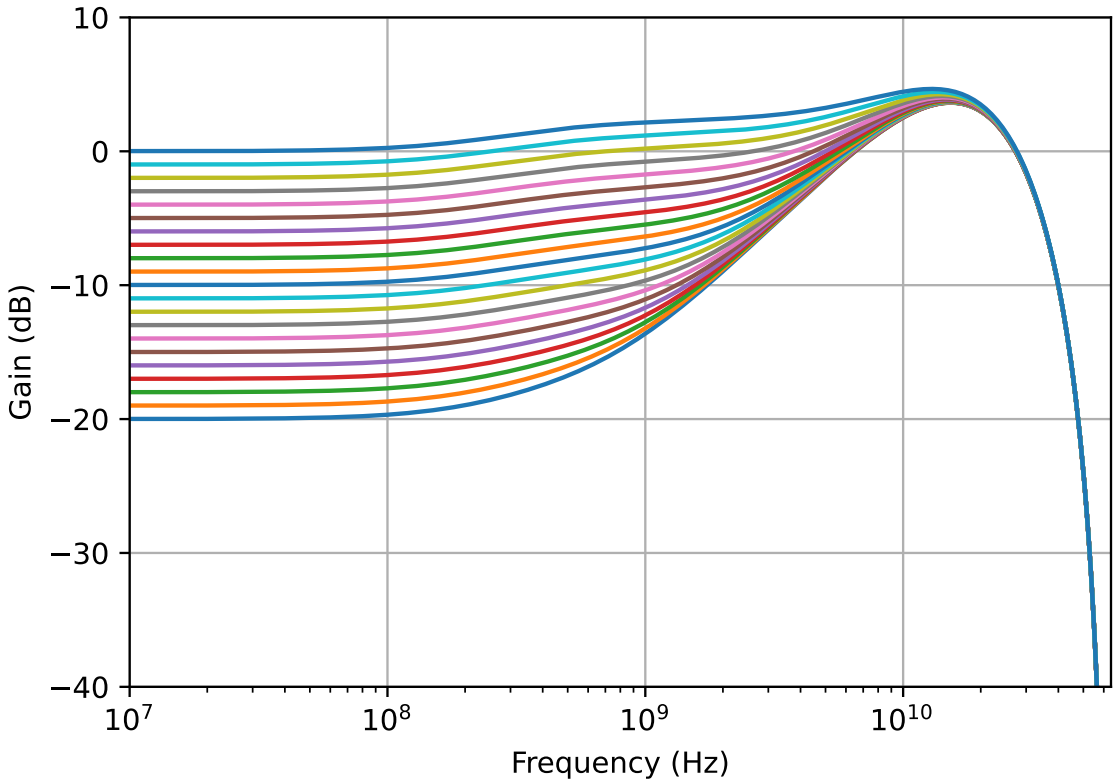
Equalization

- We use a cascaded Equalization Strategy as perscribed by the standard
- Pre distortion in the Tx Side in DSP using Feed Forward Equalization (FFE)
- Coarse correction with a Continuous Time Linear Equalizer (CTLE) with analog components
- Adaptive Equalization using Decision Directed (DD) FFE and Decision Feedback Equalization (DFE) in the Rx Side in DSP



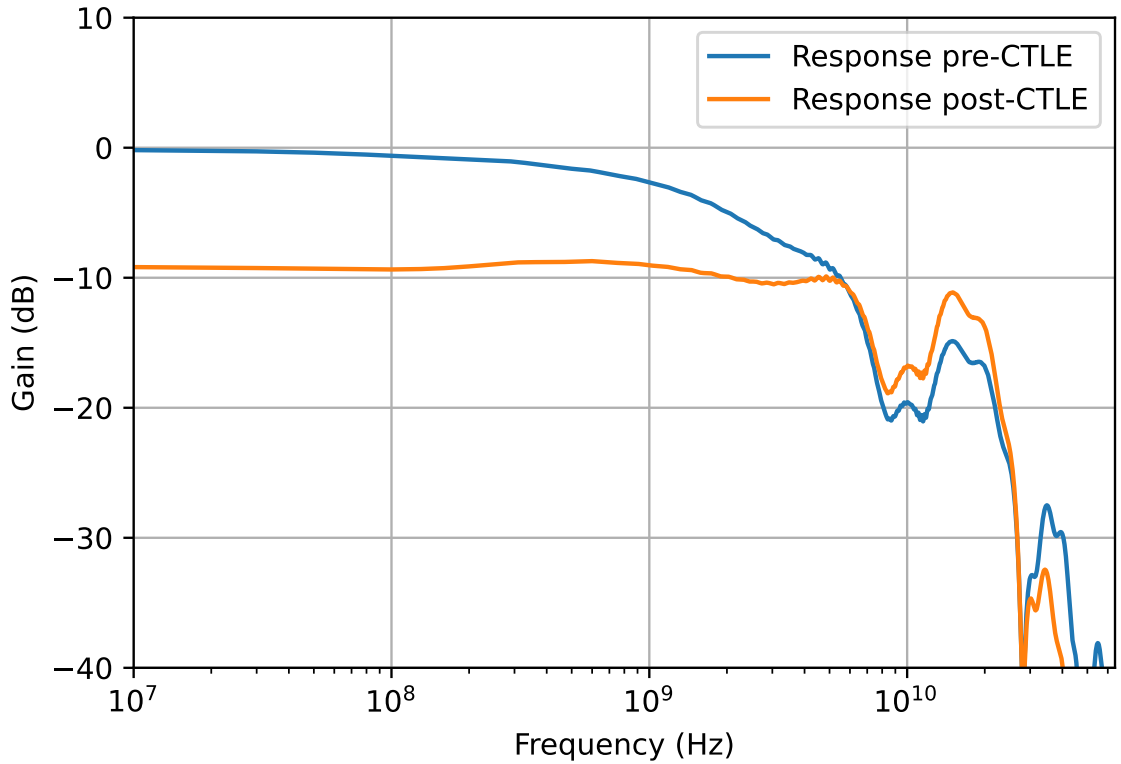
Equalization - CTLE

- In practice, the CTLE is adjustable
- Below are example CTLE filters with tuning parameter A_{DC} sweeped



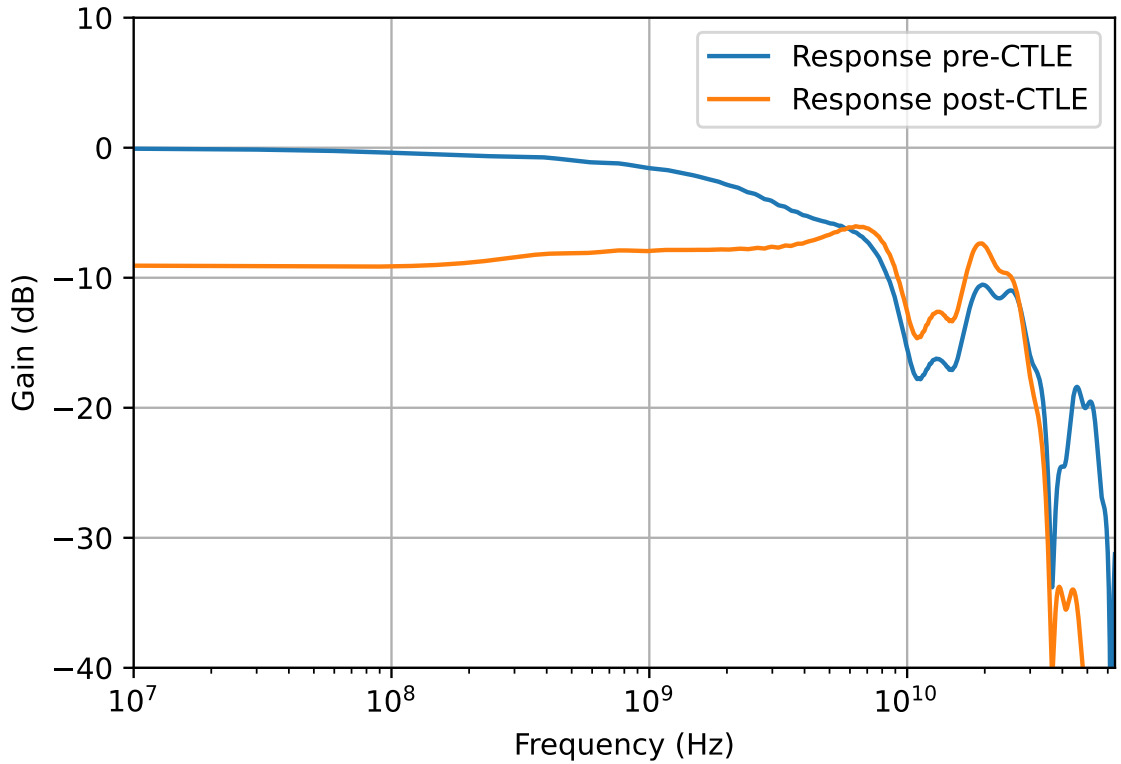
Equalization - CTLE

- Applying the CTLE with $A_{DC} = 9$dB to the medium loss channel yields:



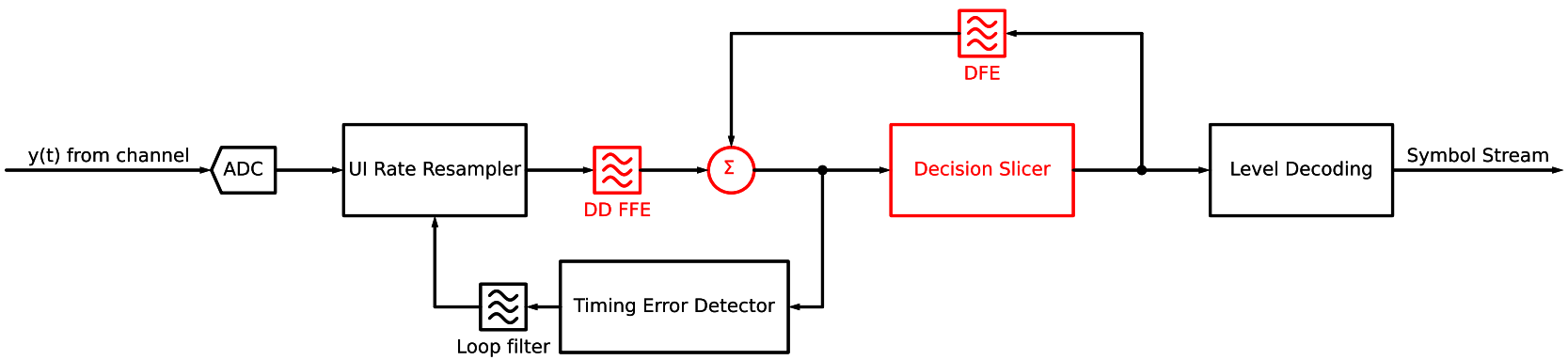
Equalization - CTLE

- Applying the CTLE with $A_{DC} = 9$dB to the low loss channel yields:



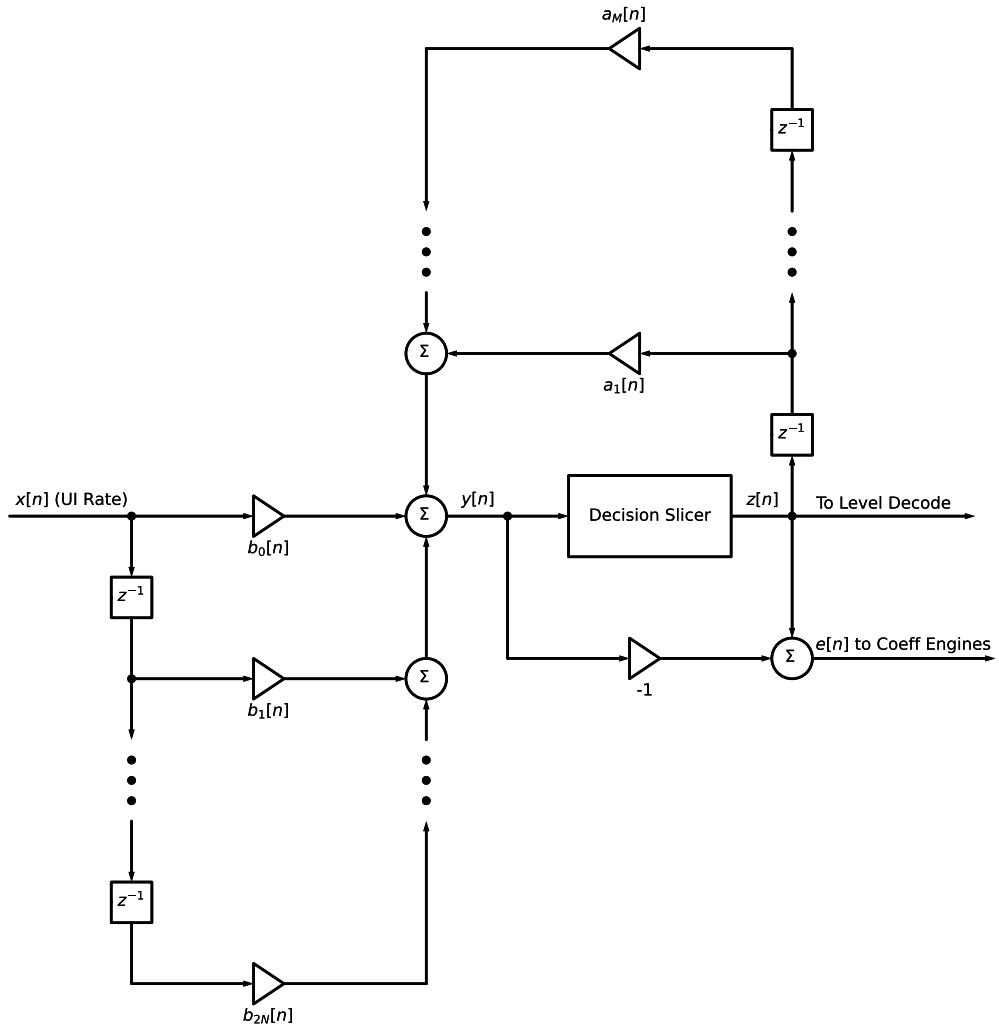
Equalization - LMS DD FFE and DFE

- Below is the receiver block diagram with the EQ subsystem highlighted
- Equalization is done at the UI rate to save power and area, even if slightly better results could be gotten by running it at the ADC sample rate



Equalization - LMS DD FFE and DFE

- At the receiver, use LMS to adapt $2N + 1$ feedforward taps $b_i[n]$, M feedback taps $a_i[n]$ to minimize ISI



Equalization - Rx Subsystem

$$e^2[n] = (z[n] - y[n])^2$$

$$e^2[n] = \left(z[n] - \left(\sum_{i=0}^{2N} x[n-i] b_i[n] + \sum_{i=1}^{M} z[n-i] a_i[n] \right) \right)^2$$

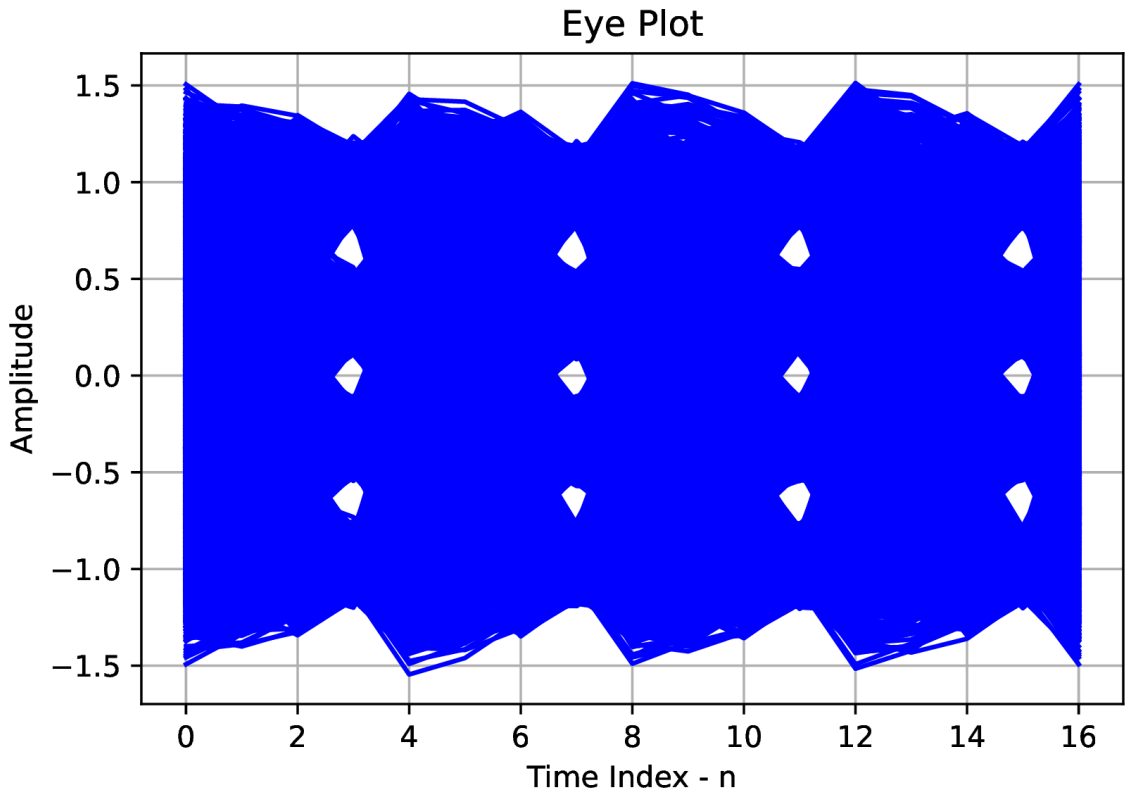
$$\frac{\partial \left(e^2[n] \right)}{\partial b_i[n]} = -2e[n]x[n-i], i \in [0, 2N+1)$$

$$\frac{\partial \left(e^2[n] \right)}{\partial a_i[n]} = -2e[n]z[n-i], i \in [1, M]$$

$$\begin{bmatrix} b_0[n] \\ \vdots \\ b_{2N}[n] \\ a_1[n] \\ \vdots \\ a_M[n] \end{bmatrix} = \begin{bmatrix} b_0[n-1] \\ \vdots \\ b_{2N}[n-1] \\ a_1[n-1] \\ \vdots \\ a_M[n-1] \end{bmatrix} + 2\mu e[n-1] \begin{bmatrix} x[n-1] \\ \vdots \\ x[n-2N-1] \\ z[n-2] \\ \vdots \\ z[n-M-1] \end{bmatrix}$$

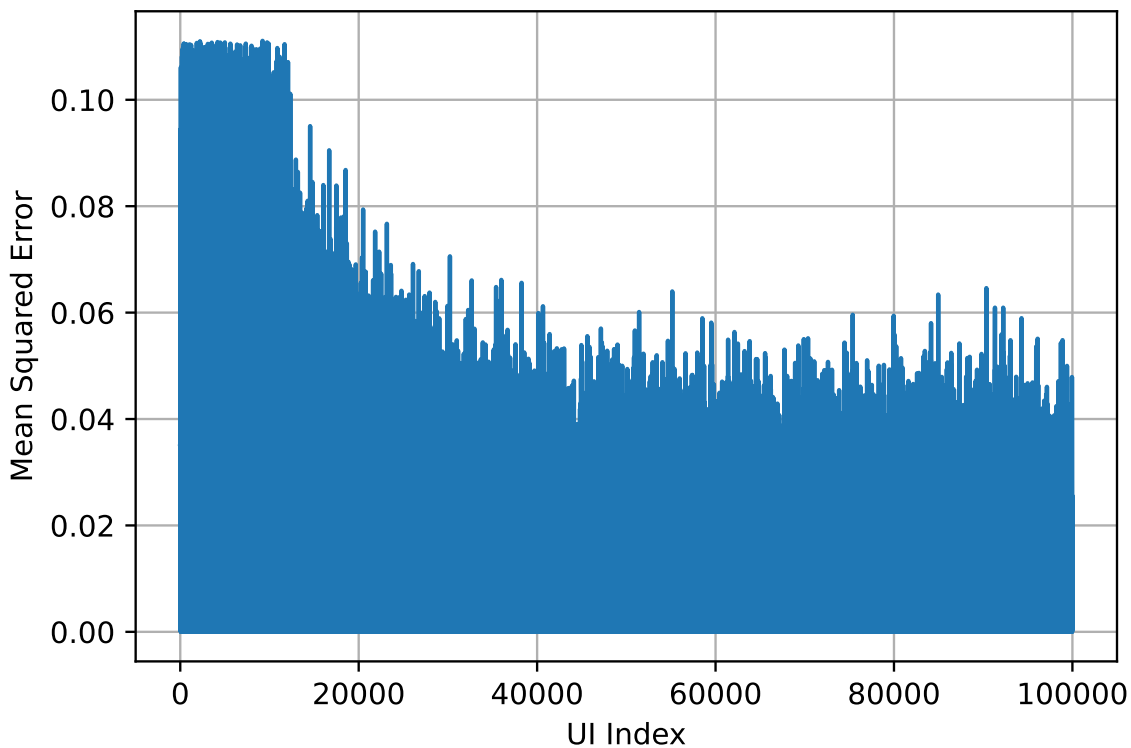
Equalization - Eye Plot

- On the low loss channel, with $N = 0$, $M = 3$, the eye is somewhat opened



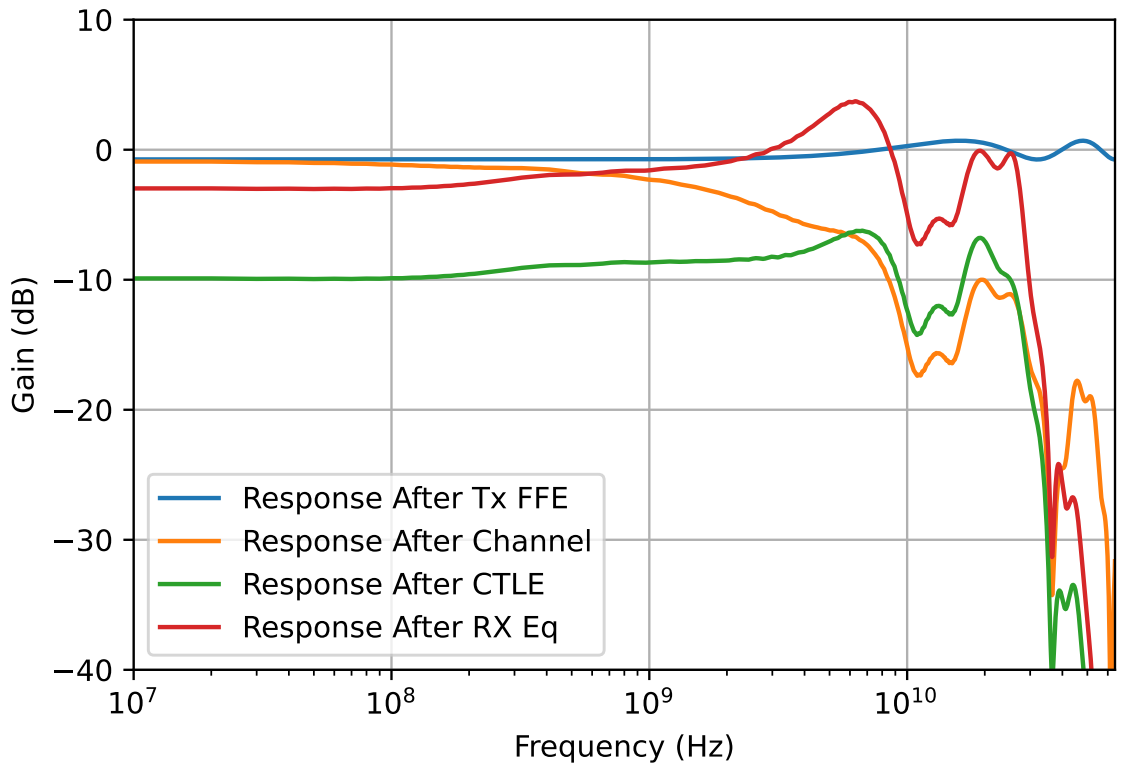
Equalization - MSE Learning Curve

- LMS slowly improves the BER, causing it to make right steps more often
- This is shown by its flatline at first, and then relatively rapid improvement



Equalization - Frequency Response

- The RX EQ is shown to want to emphasize 8GHz, 16GHz - submultiples of the symbol rate



Equalization - Eye Opening Summary (Low Loss)

- Below is the eye opening vs equalizer complexity for the low loss channel
- Note that more degrees of freedom help open the eye more
- Note that PAM-4 is more sensitive to correct EQ than NRZ is

Signal Type	N	M	Eye Opening
NRZ	0	3	70%
PAM-4	0	3	30%
NRZ	0	5	82%
PAM-4	0	5	63%
NRZ	5	3	76%
PAM-4	5	3	42%
NRZ	10	5	85%
PAM-4	10	5	66%

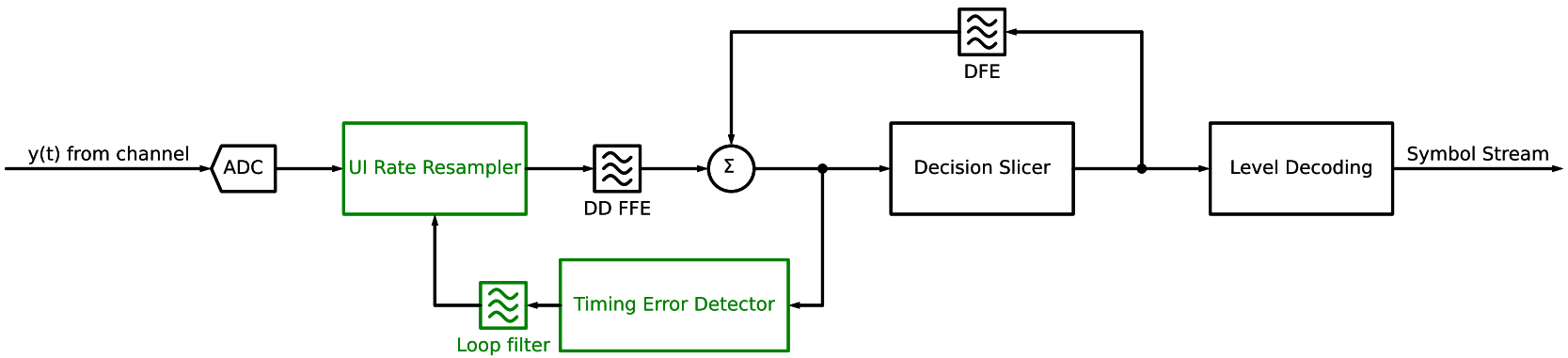
Equalization - Eye Opening Summary (Med Loss)

- Below is the eye opening vs equalizer complexity for the medium loss channel
- This channel is clearly harsher than the low loss channel
- Note that more degrees of freedom help open the eye more
- Note that PAM-4 is more sensitive to correct EQ than NRZ is
- Here, LMS failed to train on the PAM-4 eye so we bootstrapped by training on NRZ first

Signal Type	N	M	Eye Opening
NRZ	0	3	50%
PAM-4	0	3	0%
NRZ	0	5	54%
PAM-4	0	5	0%
NRZ	5	3	56%
PAM-4	5	3	$\approx$ 3%
NRZ	10	5	58%
PAM-4	10	5	38%

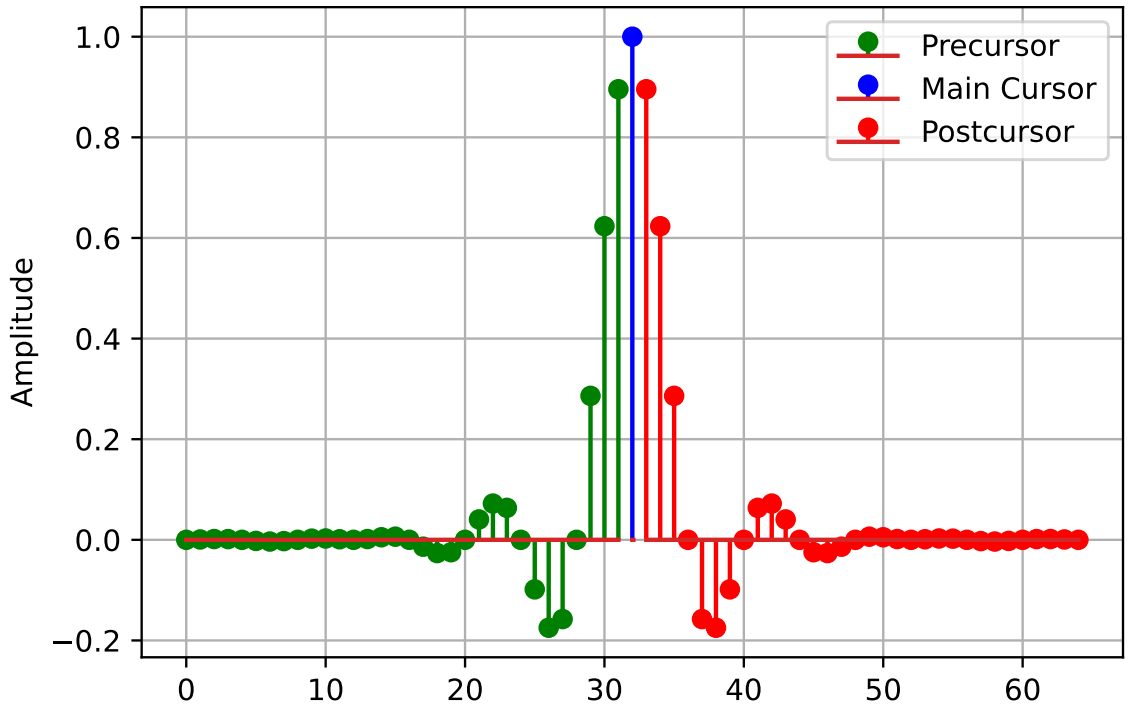
Clock Recovery - Rx Subsystem

- Below is the receiver block diagram with the modules used for CDR highlighted
- A TED is used to generate an error signal that drives a loop filter that drives a resampler
- The resampler generates samples at the provided (perhaps non integer) times
- These form a closed-loop control system



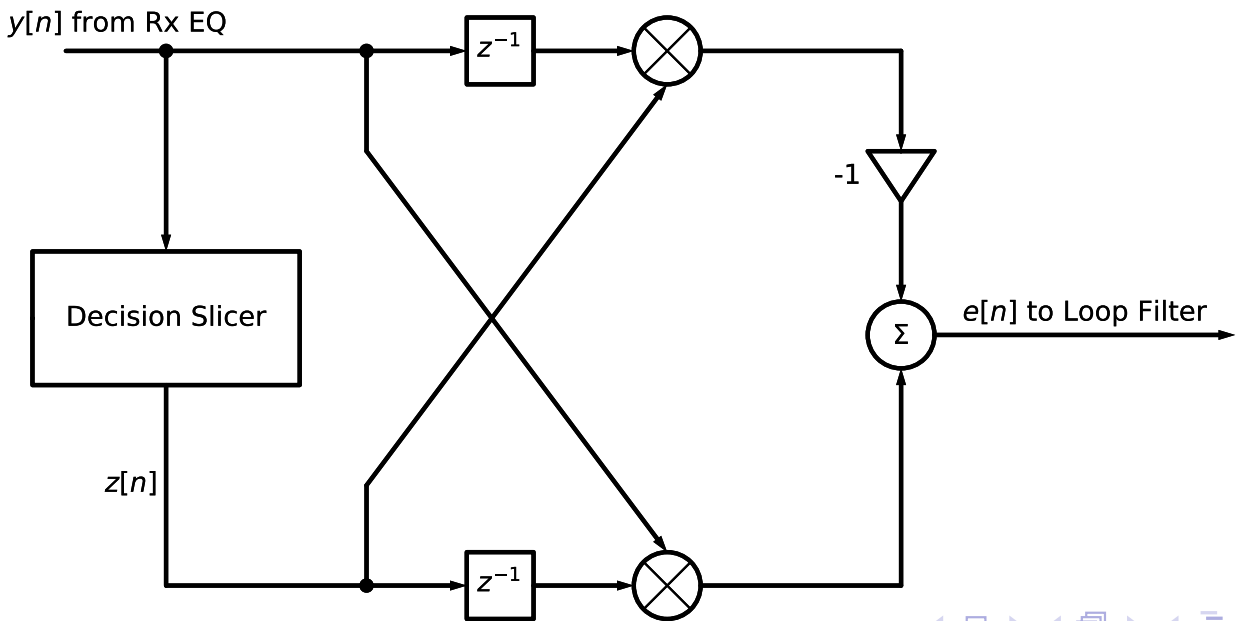
Clock Recovery - Concepts

- Receiver needs to determine from the data it receives if its sampling at the maximum eye opening, too early, or too late.
- Recall the raised cosine response; note that its derivate is 0 at the main cursor, but non zero if we sample too early or too late.
- This leads to the Early-Late TED, which uses 2 samples/symbol



Clock Recovery - MM TED

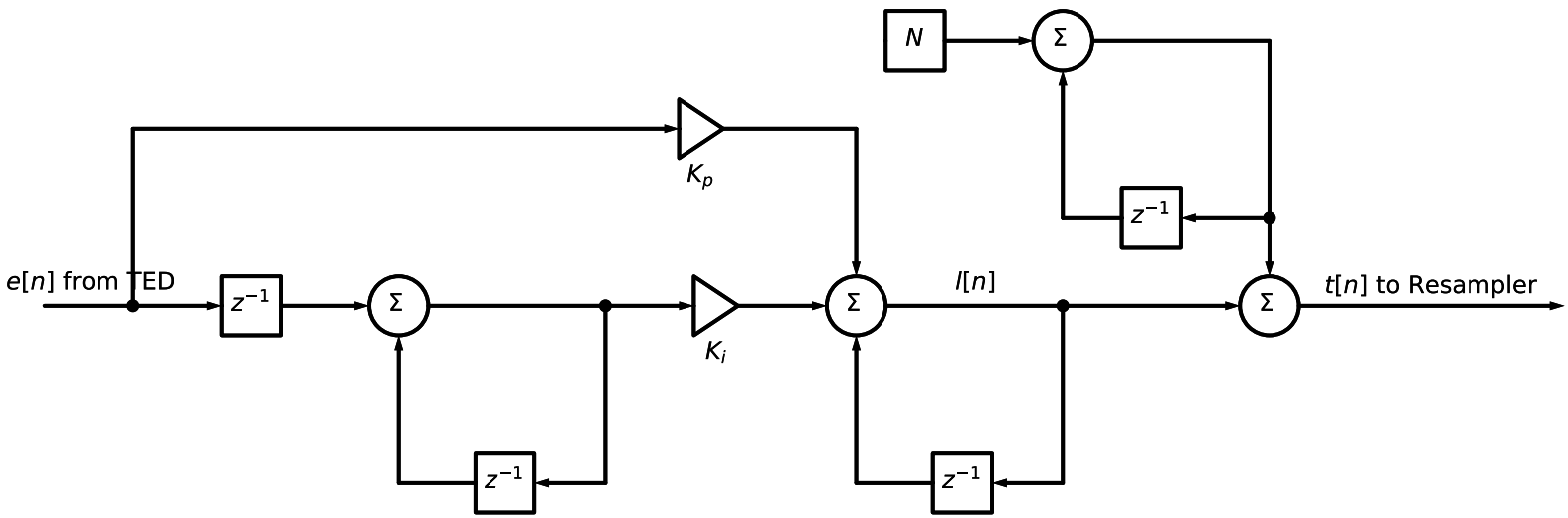
- We use a timing error detector to determine if we sampled too early or too late and use it to feed the timing control of a resampler
- Mueller and Müller's algorithm is a bit more complicated to understand, but needs only 1Sa/symbol
- The idea is that there'll be 0 ISI at symbol rate if we are sampling at the correct time, otehrewise, there'll be some correlation between past and present symbols



Clock Recovery - Loop Filter

- A loop filter with two integrators lets us track both phase and frequency errors.
- The increment by N sets the quiescent rate

$$\omega_n = B_n \frac{8\zeta}{1+4\zeta^2}; K_p = \frac{1}{K_0} 2\zeta\omega_n T_{\text{loop}}; K_i = \frac{1}{K_0} (\omega_n T_{\text{loop}})^2$$



Clock Recovery - Lagrange Interpolator

- From points (t_i, y_i), can estimate function as a polynomial at time t with $L(t)$

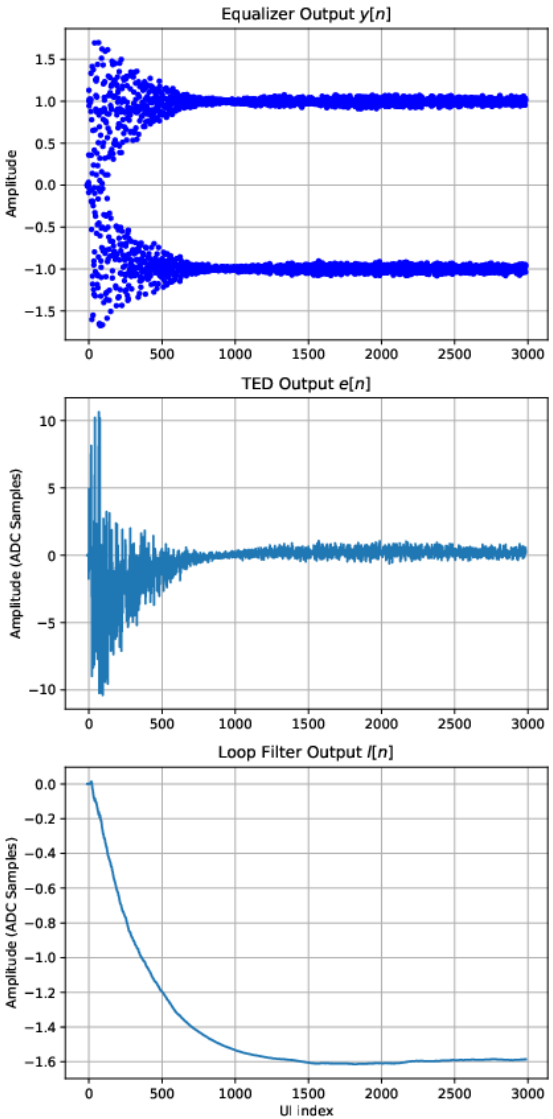
$$L(t) = \sum_{j=0}^{k-1} y_j \left(\prod_{\substack{0 \le m \le k \\ m \neq j}} \frac{t - t_m}{t_j - t_m} \right)$$

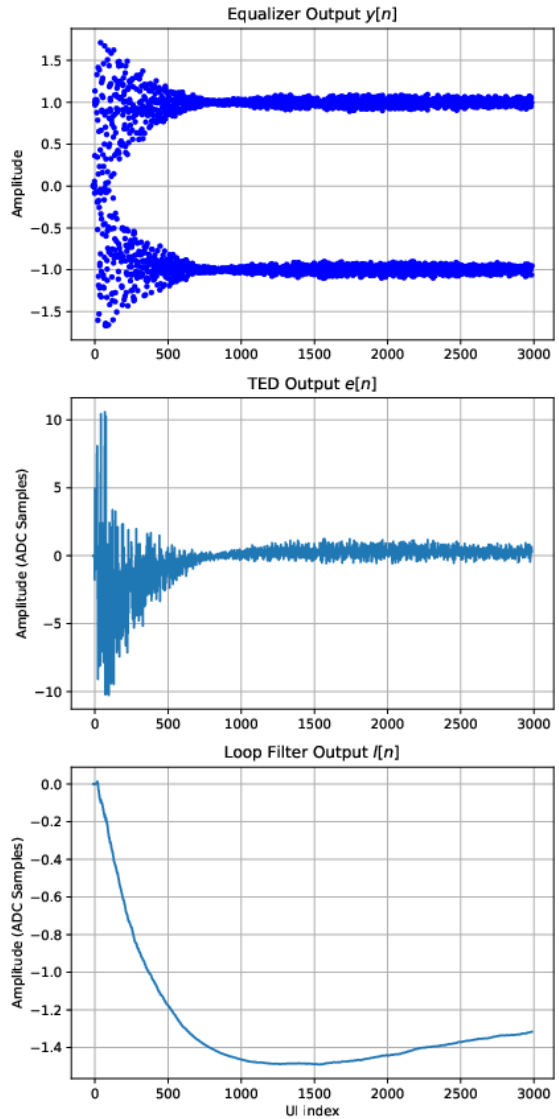
$$L_{k=1}(t) = y_0$$

$$L_{k=2}(t) = y_0 \left(\frac{t - t_1}{t_0 - t_1} \right) + y_1 \left(\frac{t - t_0}{t_1 - t_0} \right)$$

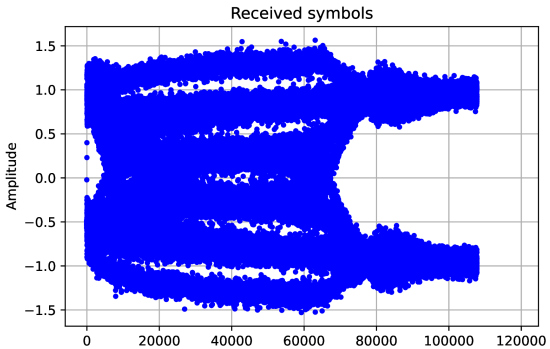
$$L_{k=3}(t) = y_0 \left(\frac{t - t_1}{t_0 - t_1} \right) \left(\frac{t - t_2}{t_0 - t_2} \right) + y_1 \left(\frac{t - t_0}{t_1 - t_0} \right) \left(\frac{t - t_2}{t_1 - t_2} \right) + y_2 \left(\frac{t - t_0}{t_2 - t_0} \right) \left(\frac{t - t_1}{t_2 - t_1} \right)$$

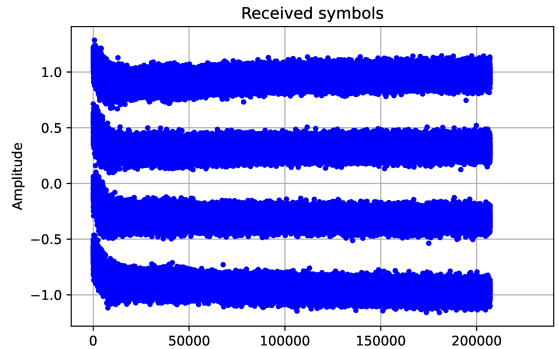
Clock Recovery - Simulation (using raised cosine)

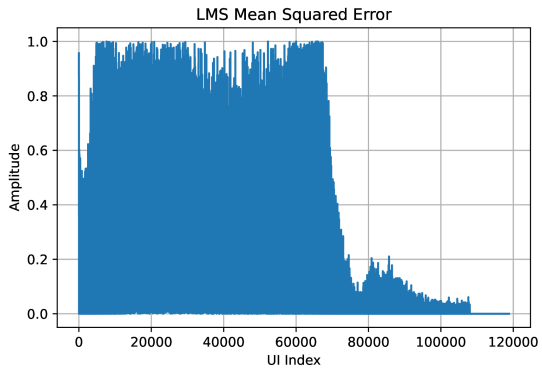


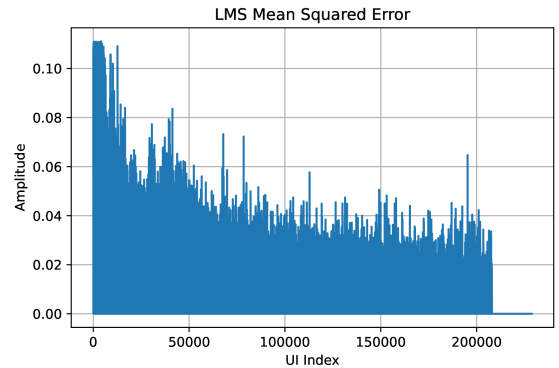


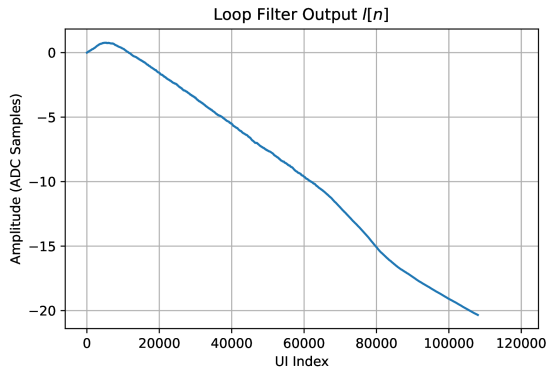
Clock Recovery - Simulation (using the channel model)

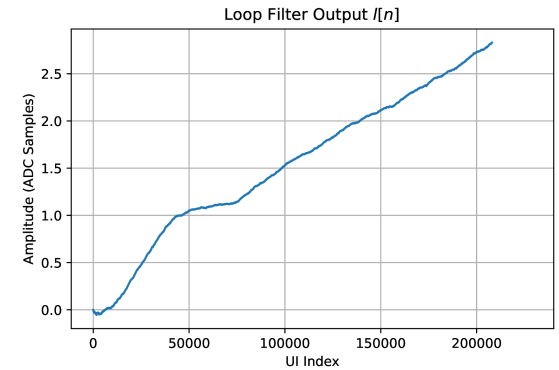












Conclusion

- We've shown that we can jointly do timing recovery and equalization
- Samples we are taking post-equalization are sufficiently seperated so that we can recover 2 bits/symbol and ensure a low BER
- We used a realistic channel and realistic assumptions as much as possible throughout this research

Suggested Additional Research

- I'd like to have spent some time exploring ASIC implementations of these algorithms
- I'd like to have spent more time studying the stability of doing joint channel estimation and channel estimation (and how does using a training pattern help)
- I'd like to have spent more time studying how the TED's performance is degraded by non-ideal pulse shapes

Questions?