

ECE 546

Lecture - 25

Equalization

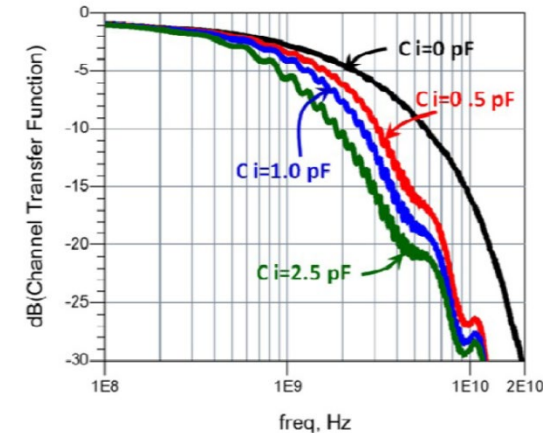
Spring 2026

Jose E. Schutt-Aine
Electrical & Computer Engineering
University of Illinois
jesa@illinois.edu

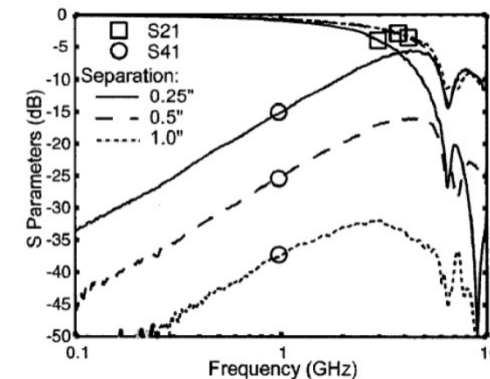
Signal Integrity Impairments In High-Speed Buses

- SI issues limit system performance to well below channel Shannon capacity
- Inter-Symbol Interference (ISI) is an issue for long backplane buses
- For short, low-cost parallel links, dominant noise source is crosstalk
 - Far-end crosstalk (FEXT) induces timing jitter (CIJ), impacts timing budget
- Other SI impairments:
 - Simultaneous-switching (SSO) noise
 - Thermal noise
 - Jitter from PLL/DLL

Insertion loss of a single DDR channel

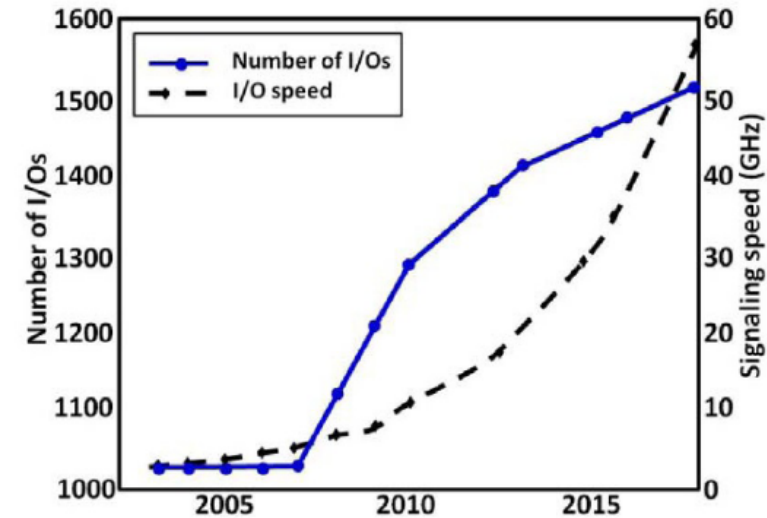


FEXT increases with routing density

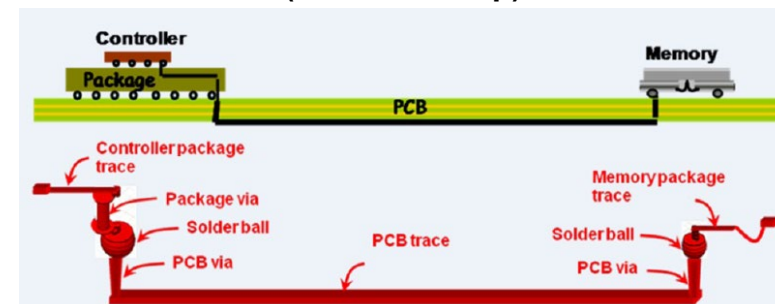


Channel Impairments

- Modern computer systems require Tb/s aggregate off-chip signaling throughput
 - Interconnect resources are limited
 - Parallel buses with fast edge rates must be used
 - Stringent power and BER requirements to be met
 - High-performance signaling requires high-cost channels
 - Difficult to design and costly to manufacture
 - One of main limiting factors: crosstalk-induced jitter

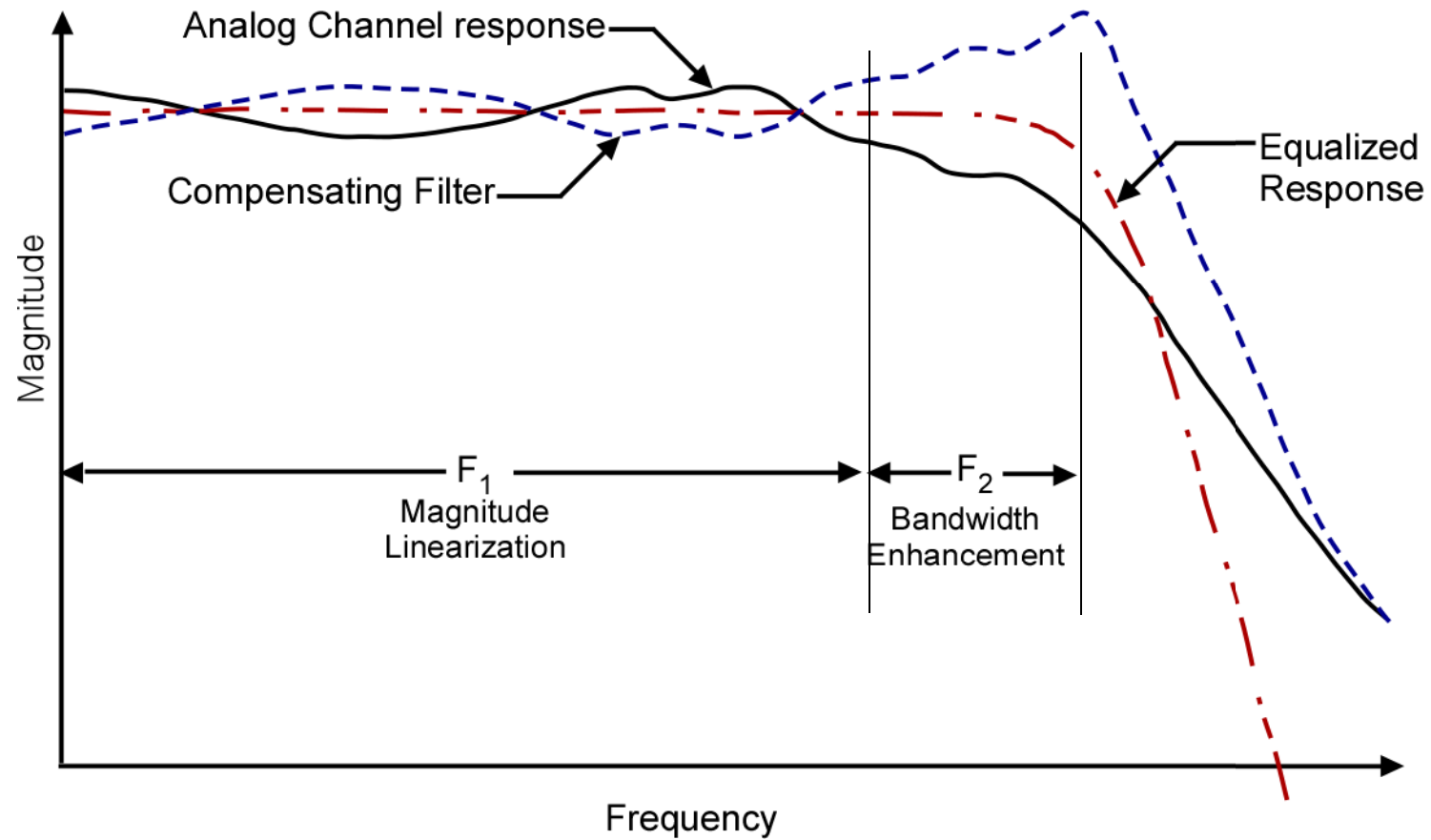


Available number and required speed of I/Os (ITRS roadmap)

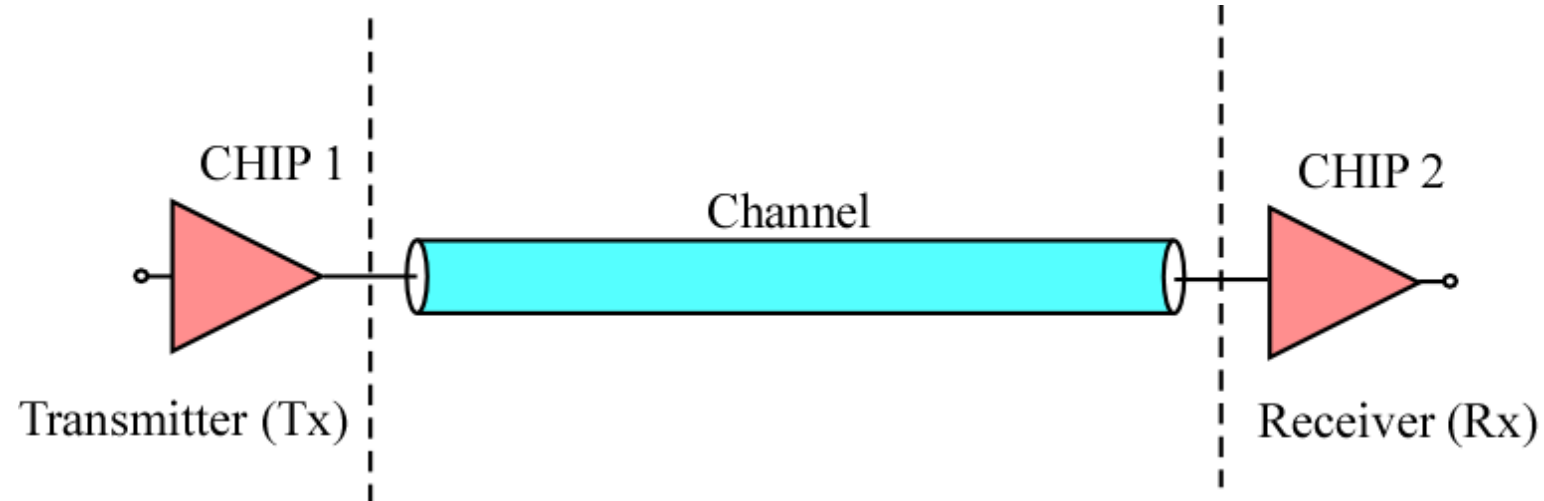


A typical controller-memory interface

Channel Equalization

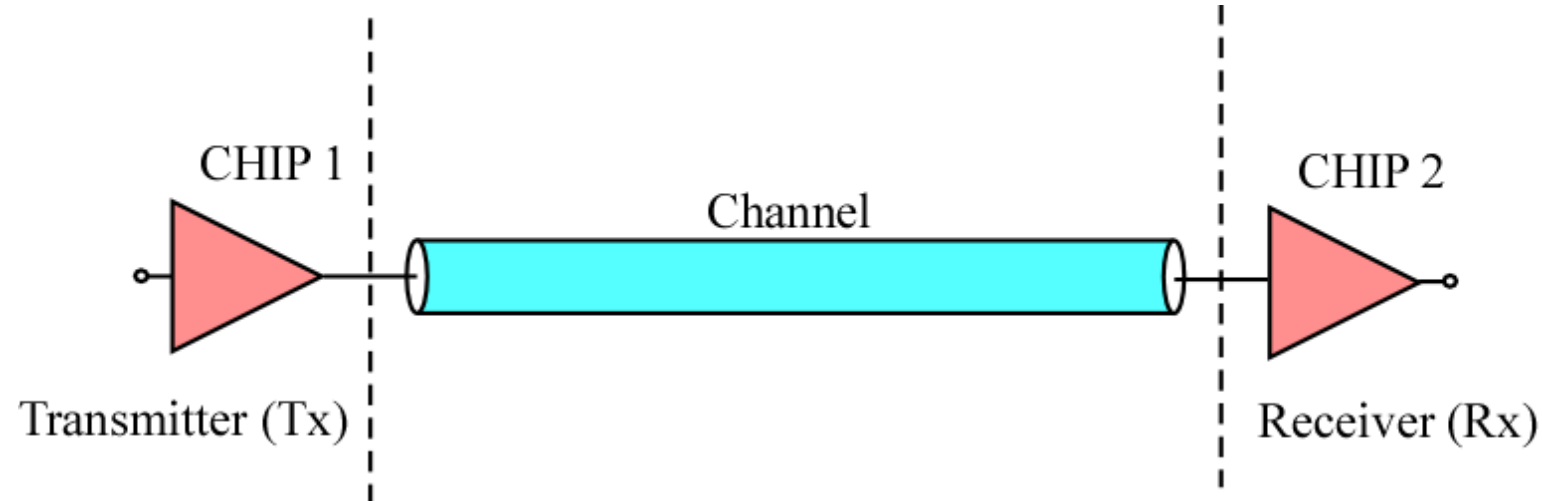


Equalization



Off-chip bandwidth scales at a much lower rate than on-chip bandwidth. Primary objective is to have low bit error rate (BER). Typical BER is 10^{-12} .

Equalization



Frequency shaping filters that flatten the channel response up to a certain frequency. Objective is to improve BER and increase eye opening.

Pre-Emphasis and Equalization

- Pre-emphasis boosts the high-frequency contents of the signal at the transmitter before the signal is sent through the channel.
- A two-tap finite impulse response (FIR) filter is an example of pre-emphasis implementation.
- Pre-emphasis has high power requirements, aggravates crosstalk and increase EMI.
- Pre-emphasis cannot improve SNR
- Data converters are required to implement pre-emphasis

Receiver Equalization

- The loss in the channel is suppressed by boosting the high-frequency content of the signal.
- Often results in larger noise margins.
- Receivers can be implemented in discrete-time or continuous time.
- Implementations include digital FIR equalizer, analog FIR equalizer, continuous time equalizer.

Tap in Equalization

Tap: **adjustable weight** used to shape signal and compensate for impairments

Consists of three blocks:

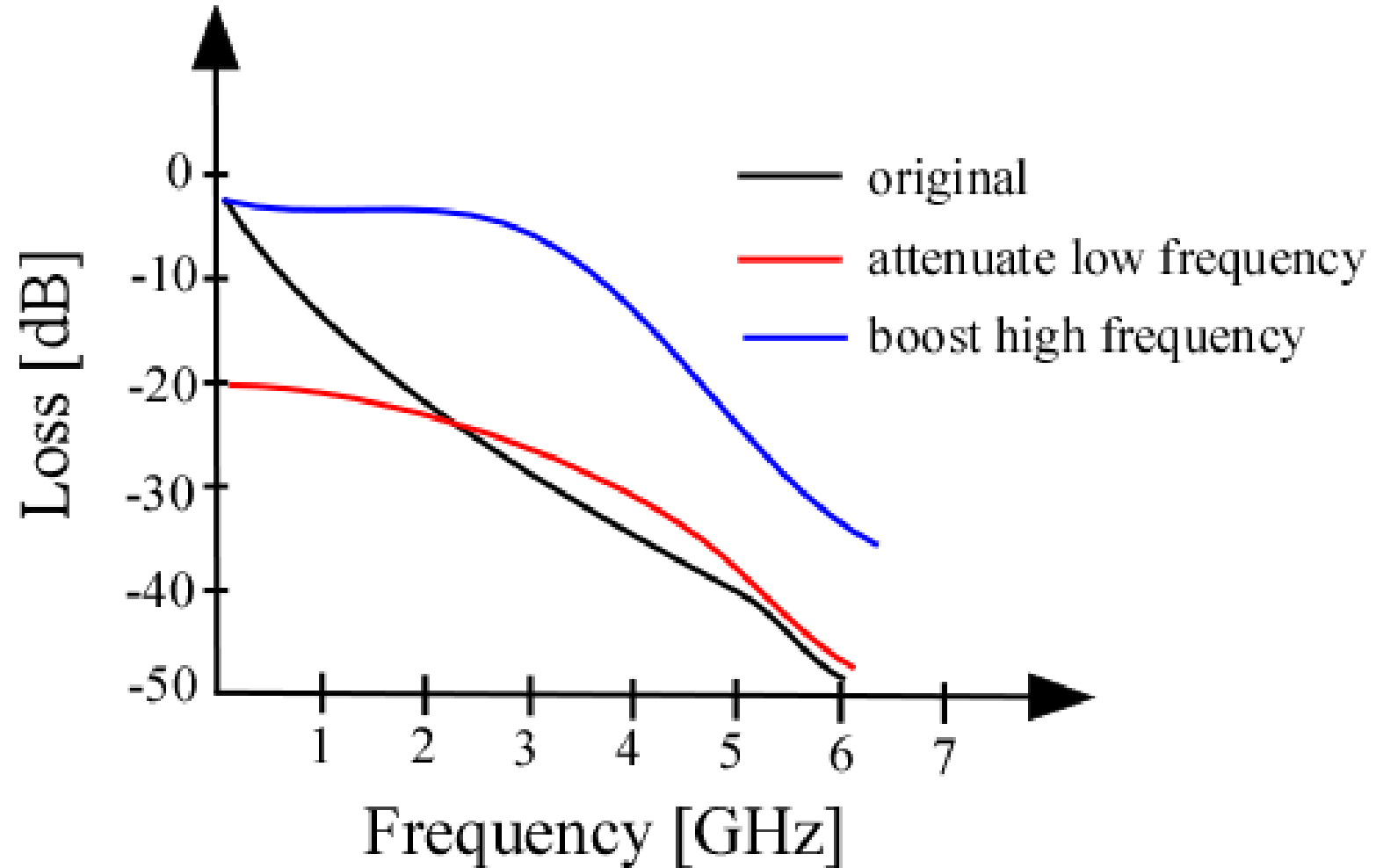
1. **Delay element** – stores previous samples
2. **Multiplier** – scales sample by tap coefficient
3. **Adder** – sums all taps

Tap Implementation

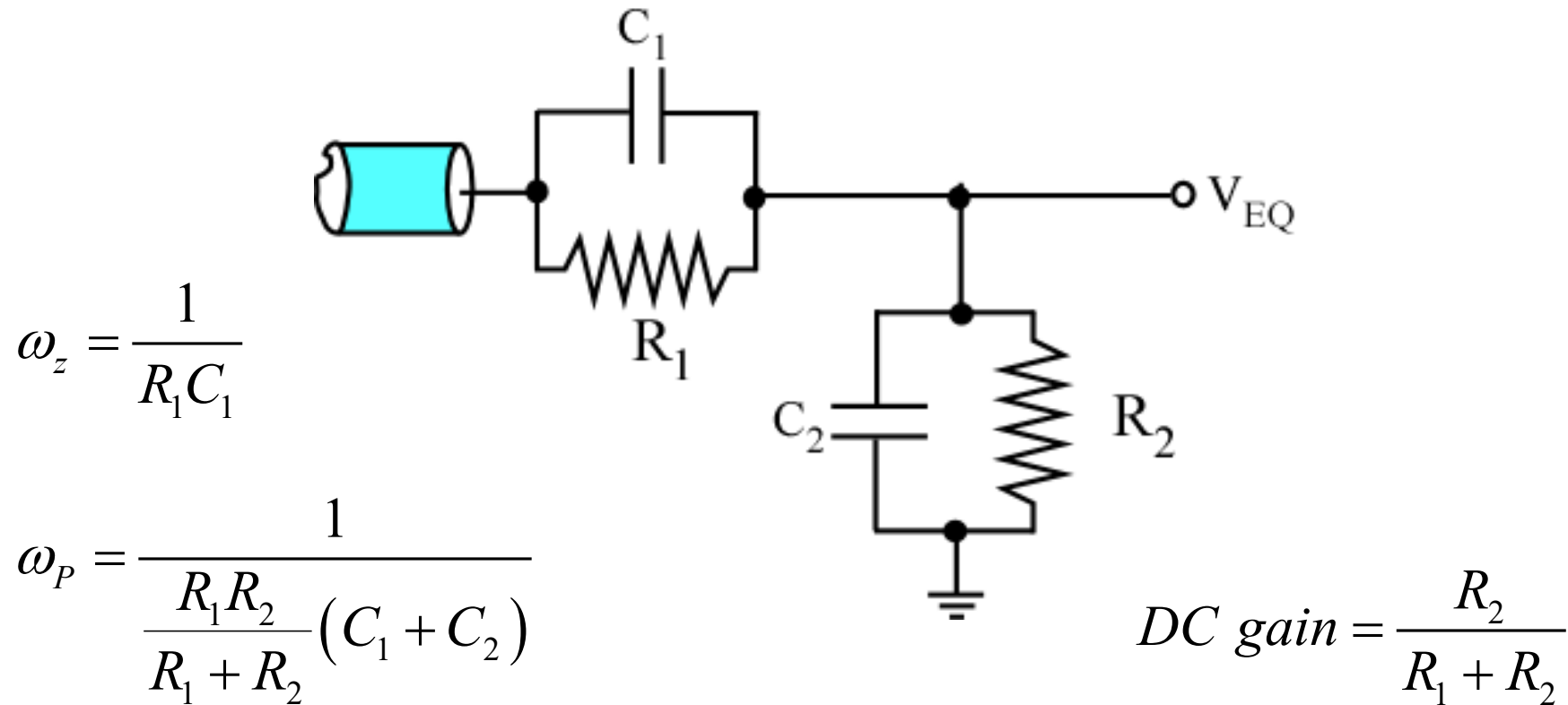
Equalizer Type	Delay Implemented By	Weight Implemented By	Sum Implemented By
DSP FIR	Registers	Multiplier	Adders
TX FFE	Symbol timing latches	Driver current slices	Wire/current sum
RX DFE	Bit memory	DAC/current cell	Analog subtract node
Analog FIR	Delay line	Gain stage	Summing amp

Source: ChatGpt

Equalization Techniques



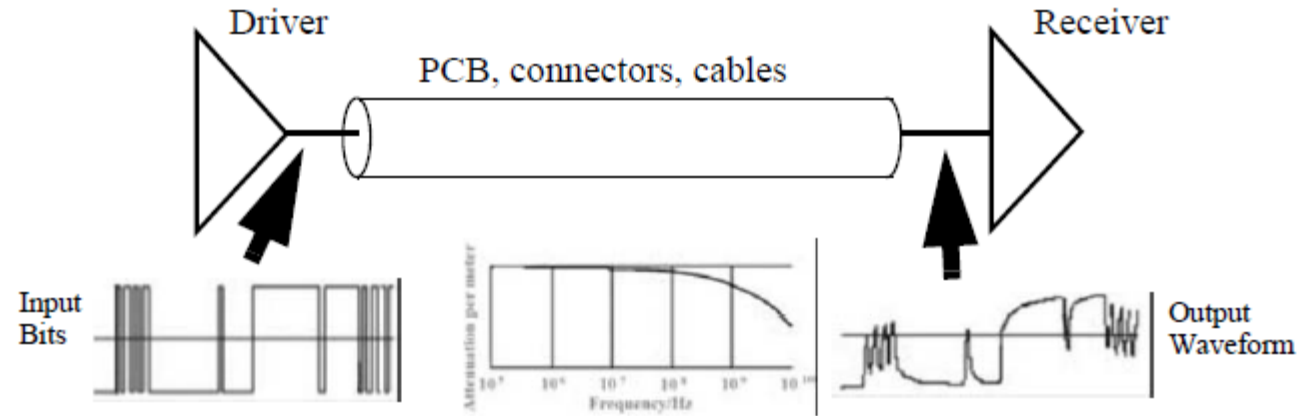
Continuous Time Passive Equalizer



$$H(s) = \frac{R_2}{R_1 + R_2} \frac{1 + R_1 C_1 s}{1 + \frac{R_1 R_2}{R_1 + R_2} (C_1 + C_2) s}$$

Channel-Equalization

Typical Channel Response w/o Equalization:

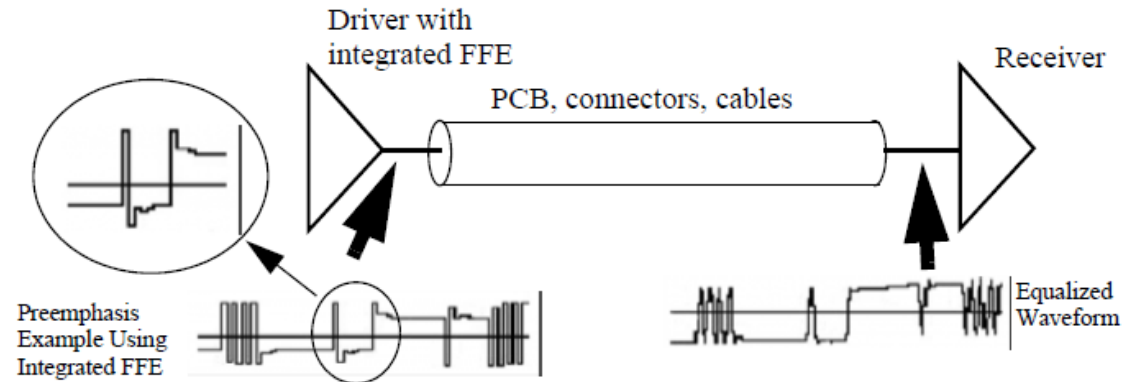


- Equalization at TX and RX needed to counter the effects of channel, properly decode signals.
- TX: FFE (Feed-Forward Equalizer)
- RX: DFE (Decision-Feedback Equalizer)

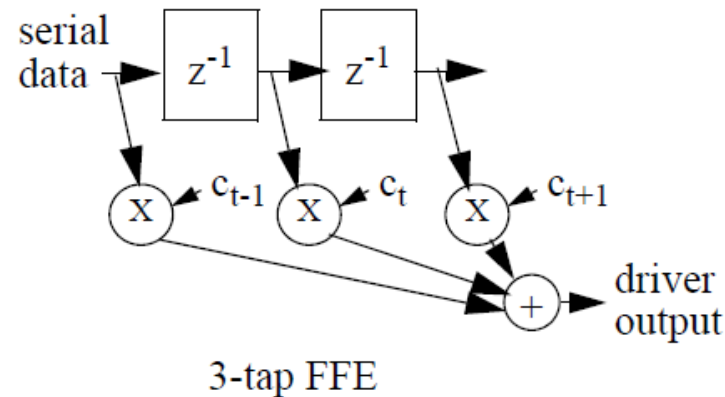
D. R. Stauffer et al., "High Speed Serdes Devices and Applications", Springer 2008

FFE Circuit Architecture

Typical Channel Response at Receiver with FFE at TX:



Sample 3-tap FFE Architecture:

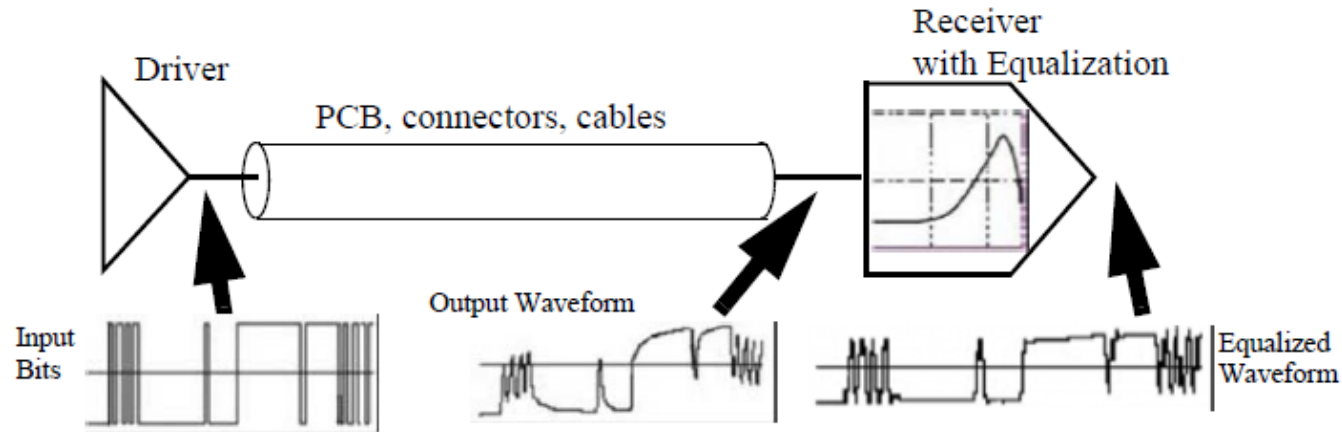


D. R. Stauffer et al., "High Speed Serdes Devices and Applications", Springer 2008

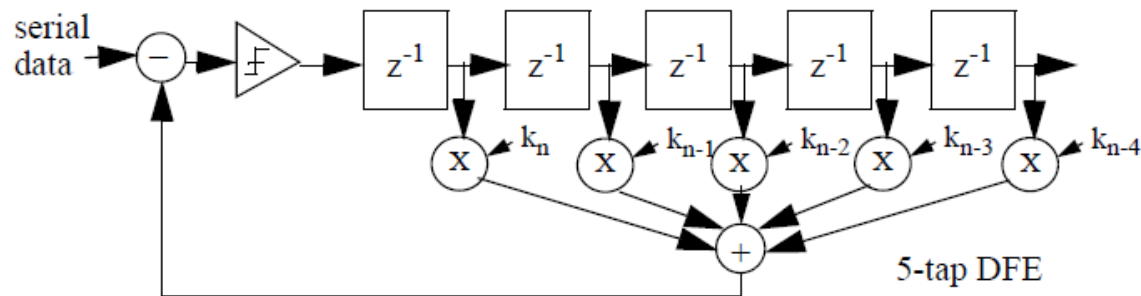
- FFE taps selected to generate a filter with the inverse transfer-function as that of channel.
- Trade-off b/w signal amplitude at receiver and jitter.

DFE Circuit Architecture

Typical Channel Response at Receiver with DFE at RX:



Sample 5-tap DFE Architecture:



D. R. Stauffer et al., "High Speed Serdes Devices and Applications", Springer 2008

- DFE is needed in links with a high-baud rate to min. signal ampl. at high freq. caused by channel jitter.
- Filter weights selected dynamically in a feedback loop to max. eye opening.

FFE vs. DFE

- **FFE**

- Can mitigate the pre-cursor channel response in low-BW channels.
- Can compensate ISI arising from transient TL loss over wide time-spans.

- **DFE**

- Cannot equalize ISI arising from pre-cursor channel response.
- Can only compensate ISI from a fixed time-span.

FFE + DFE

- Guarantees max. performance from the SerDes.
- Advantage:
 - DFE permits use of low-frequency de-emphasis at TX resulting in a larger received signal envelope, smaller signal/crosstalk ratio.
 - System capable of employing continuous adaptive equalization of its feedback taps to optimize performance.

Equalization Techniques

- CTLE (Continuous-Time Linear Equalizer) Basics
- FFE (Feed-Forward Equalizer) Basics
- DFE (Decision Feedback Equalizer) Basics
- More Complex Equalization

Continuous Time Linear Equalization

- Goal: To counteract the effects of the channel's transfer function (s-domain)
- Accomplished via amplification
 - More amplification at operating frequency
 - Less amplification at \ll operating frequency (DC Gain)
 - Reduce higher frequency noise

Drawbacks of CTLE Design

- Drawbacks of RX CT Equalization:
 - Amplifying signal also amplifies noise + crosstalk (SNR stays same)
 - Trade-off: High Gain + Output Swing vs. Small Size + Low Power Consumption
- When designing CTLE, need to iterate in order to optimize on all of these ends
- Still need to utilize filtering for noise and crosstalk

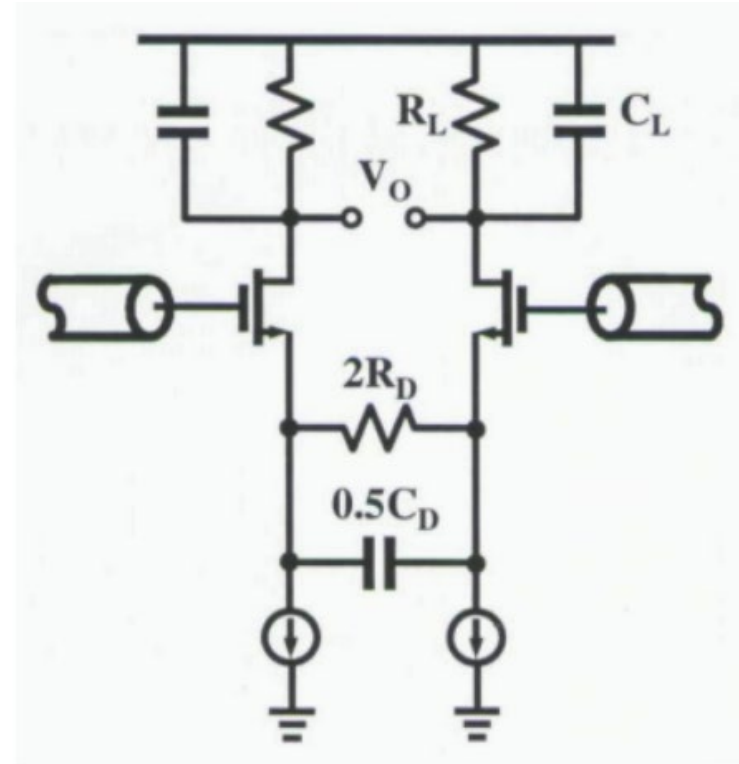
Continuous Time Linear Equalizer (CTLE)

– Pros

- Single block → lower power consumption and smaller sizing
- Easy to cancel precursor and more ISI

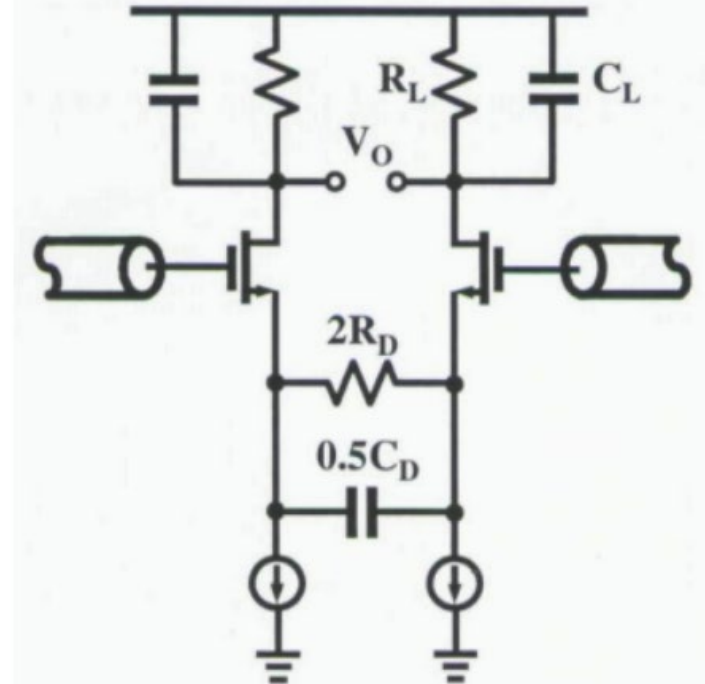
– Cons

- Noise+Crosstalk amplified as well
- Hard to tune



Continuous Time Linear Equalizer (CTLE)

- Active equalizer topology shown to right
- Differential amplifier with degeneration
 - Introduces an extra pole and zero
 - Total: One zero, two poles
- Transfer Function = Peaking Amplifier



Equations for CTLE (Derived from Circuit)

$$H(s) = \frac{g_m}{C_L} \frac{s + \frac{1}{R_D C_D}}{\left(s + \frac{g_m R_D + 1}{R_D C_D}\right) \left(s + \frac{1}{R_L C_L}\right)}$$

$$\omega_z = \frac{1}{R_D C_D}$$

$$\omega_{p1} = \frac{g_m R_D + 1}{R_D C_D}$$

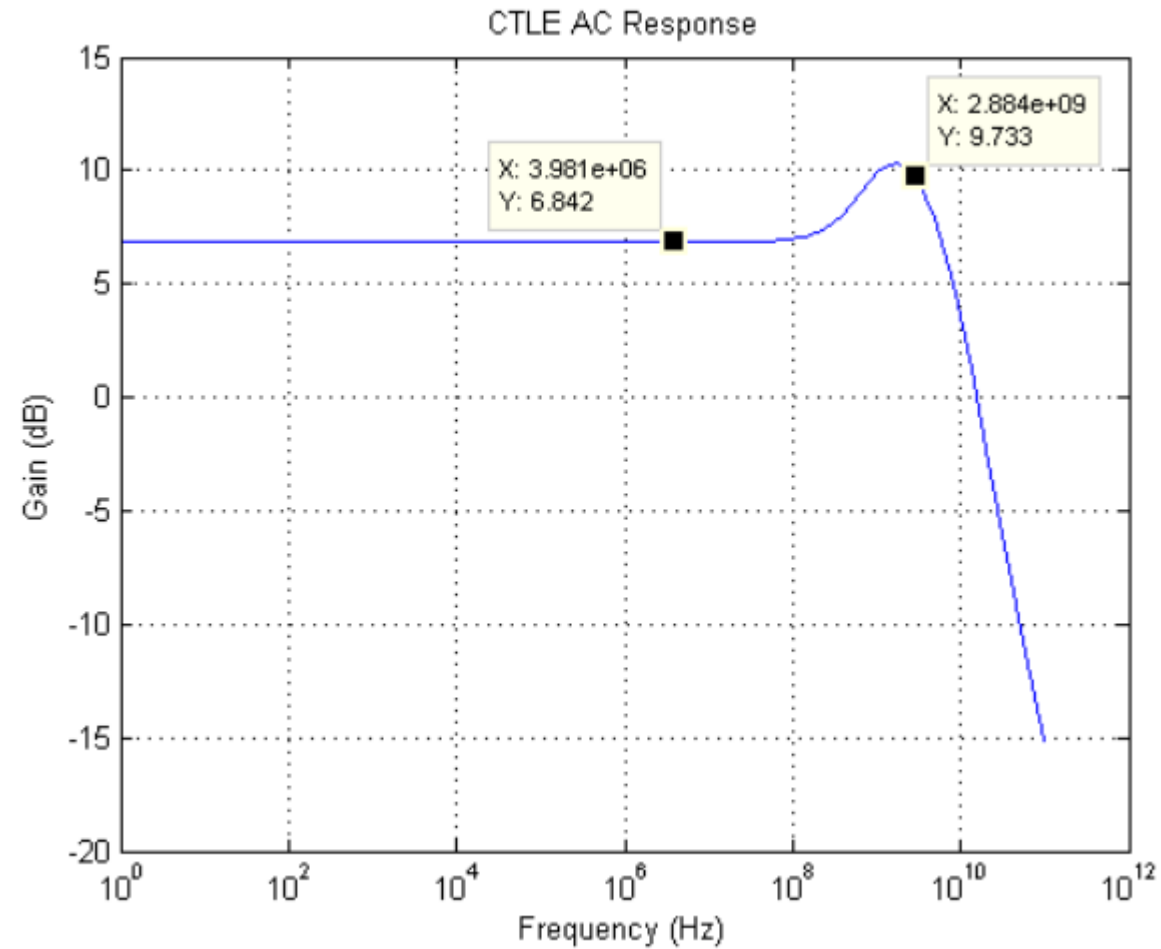
$$\omega_{p2} = \frac{1}{R_L C_L}$$

$$DC \text{ Gain} = \frac{g_m R_L}{g_m R_D + 1}$$

CTLE Design Process

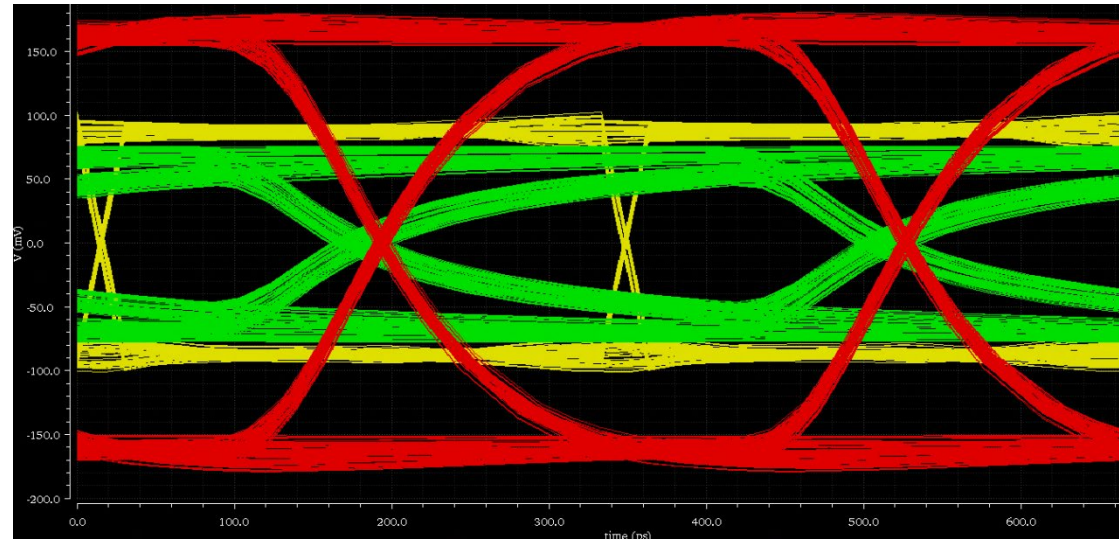
- 1) Choose DC Gain and Peaking Gain (use insertion loss curve)
- 2) Decide optimal poles and zero frequency placements
- 3) Determine load capacitance from next stage (CDR input)
- 4) Determine equalizer output swing
- 5) Calculate component parameters to meet above specs
- 6) Test and optimize as necessary (iterative process)

CTLE Transfer Function (Bode Plot)

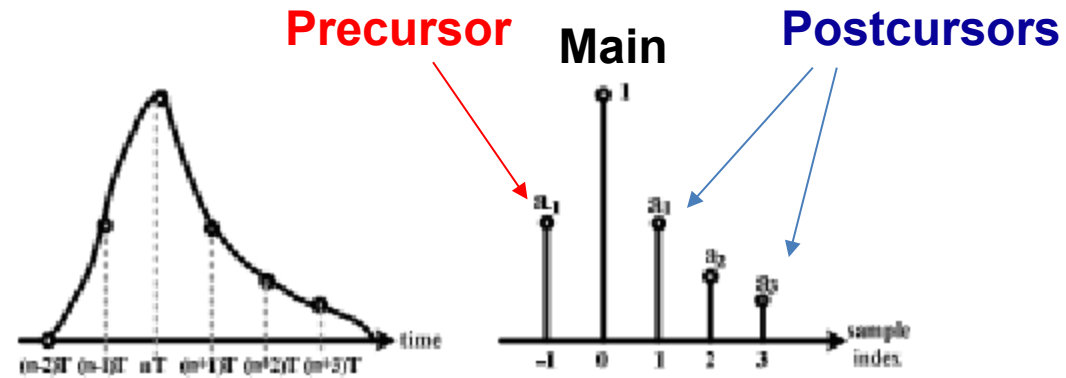
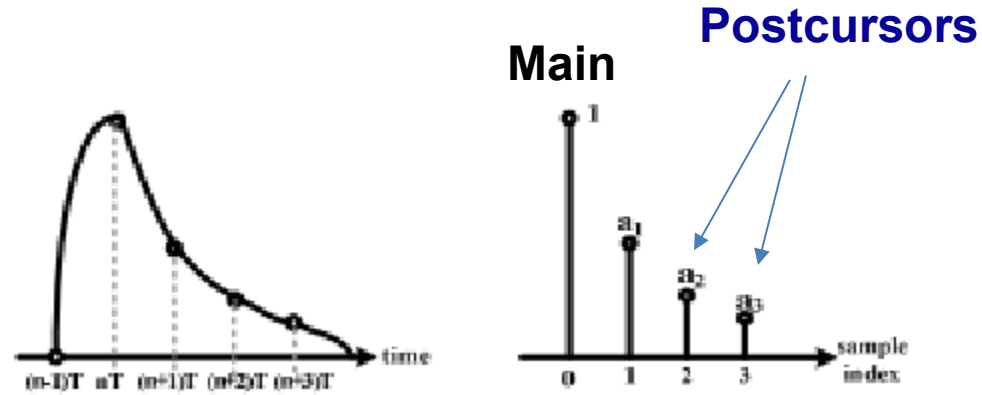


Effects of CTLE (Eye Diagram)

- Eyes
 - Yellow = TX end
 - Green = Post-Channel
 - Red = Post-EQ

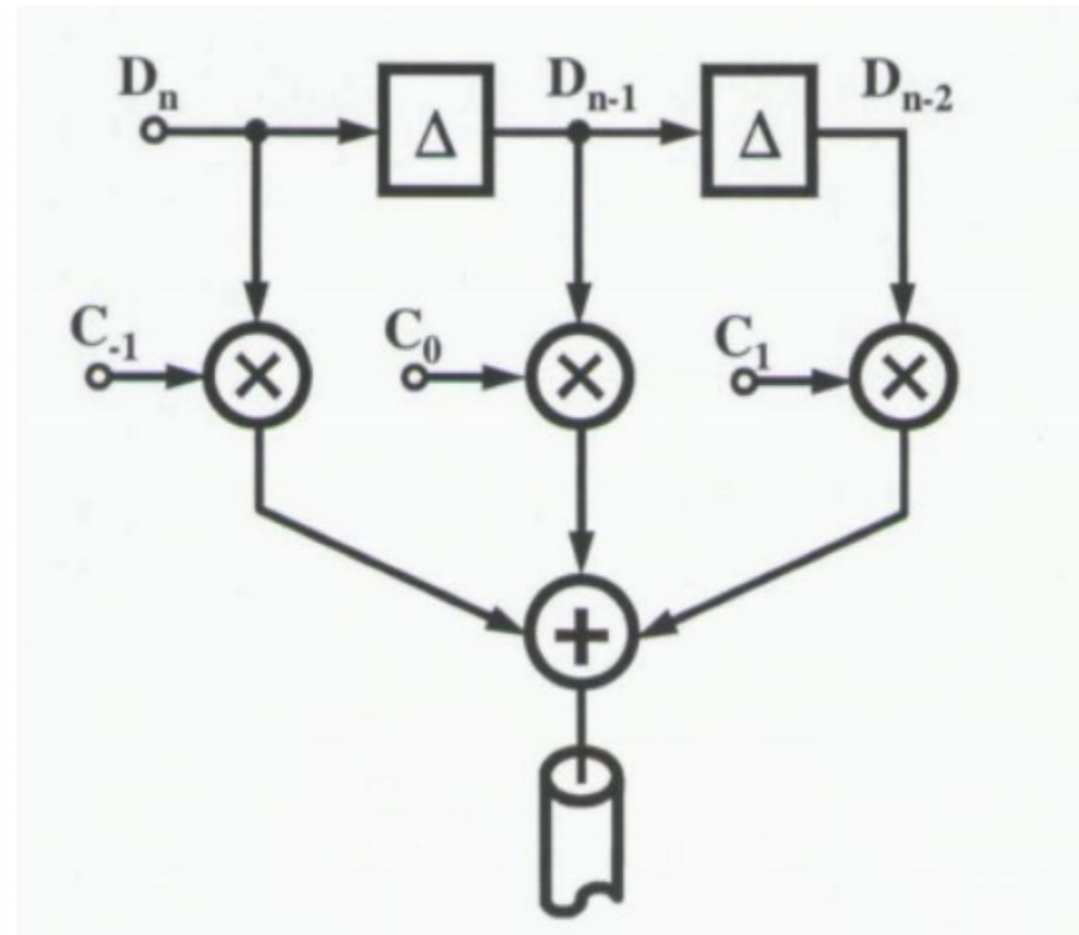


Precursors and Postcursors



Understanding FFE

- Pros
 - Simple to implement
 - Doesn't amplify noise
 - Easily cancels precursors
- Cons
 - Signal Attenuated due to peak-power limitation (output swing limit)
 - Hard to tune taps



FFE Coefficient Calculation

- Need to calculate FFE coefficients such that convolution with channel results in solely the main cursor
 - A = channel coefficients
 - b = FFE coefficients
 - c = equalized response

$$\mathbf{A} \times \mathbf{b} = \mathbf{c}$$

$$\begin{bmatrix} a_0 & a_{-1} & 0 & 0 & 0 \\ a_1 & a_0 & a_{-1} & 0 & 0 \\ a_2 & a_1 & a_0 & a_{-1} & 0 \\ a_3 & a_2 & a_1 & a_0 & a_{-1} \\ 0 & a_3 & a_2 & a_1 & a_0 \end{bmatrix} \times \begin{bmatrix} b_{-1} \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

FFE Coefficient Calculation (Only precursor)

- When solely eliminating precursor, matrix becomes:
 - Only b_{-1} and b_0 matter to eliminate precursor
- Appending an extra zero at beginning in order to properly account for full sampled response
- A-matrix goes down to n amount of postcursors
 - Can match number with number of FFE coefficients
 - However, more postcursors \rightarrow more ISI eliminated

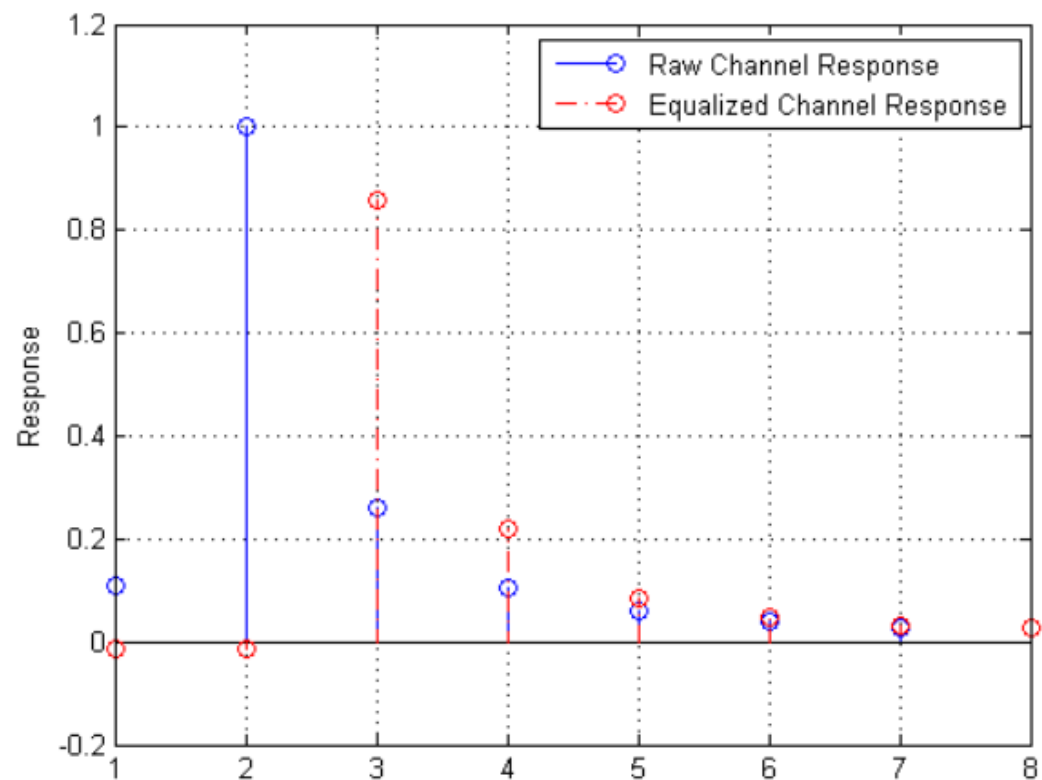
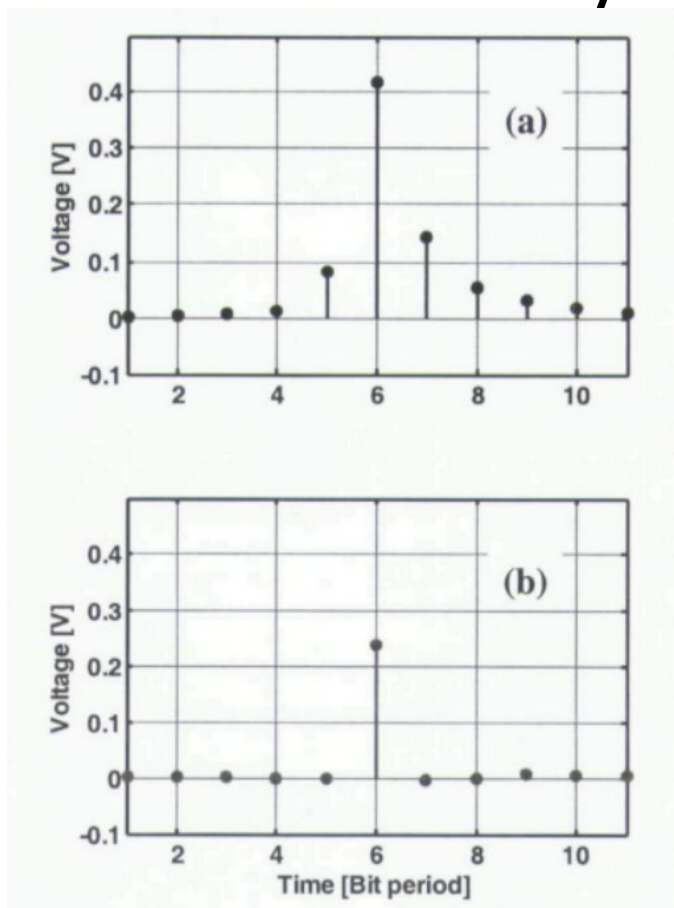
$$\mathbf{A} = \begin{bmatrix} a_{-1} & 0 \\ a_0 & a_{-1} \\ a_1 & a_0 \\ a_2 & a_1 \\ a_3 & a_2 \\ a_4 & a_3 \\ \vdots & \vdots \\ 0 & a_n \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Effects of FFE

Full FFE

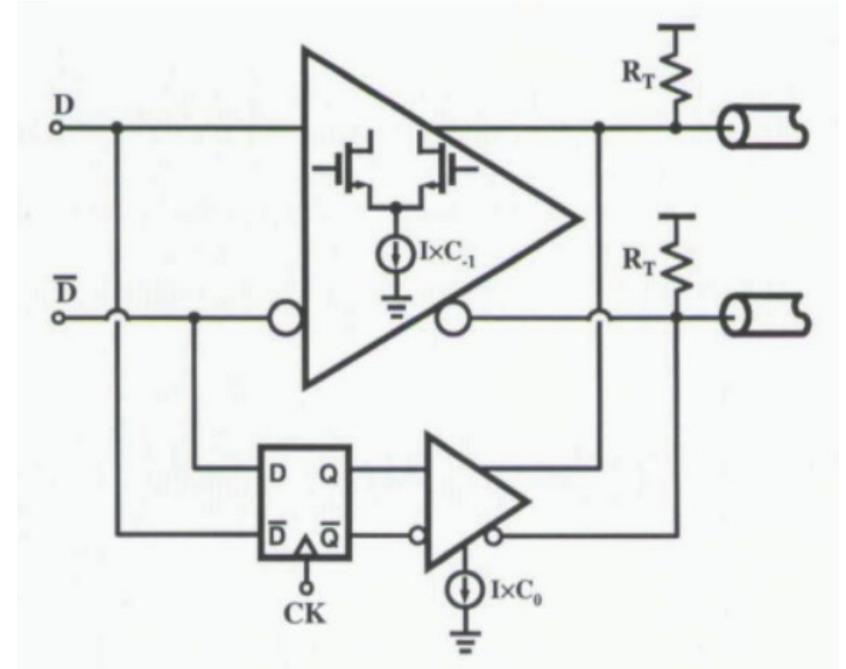
Precursor Only



Actual FFE Design: Normalize Coefficients

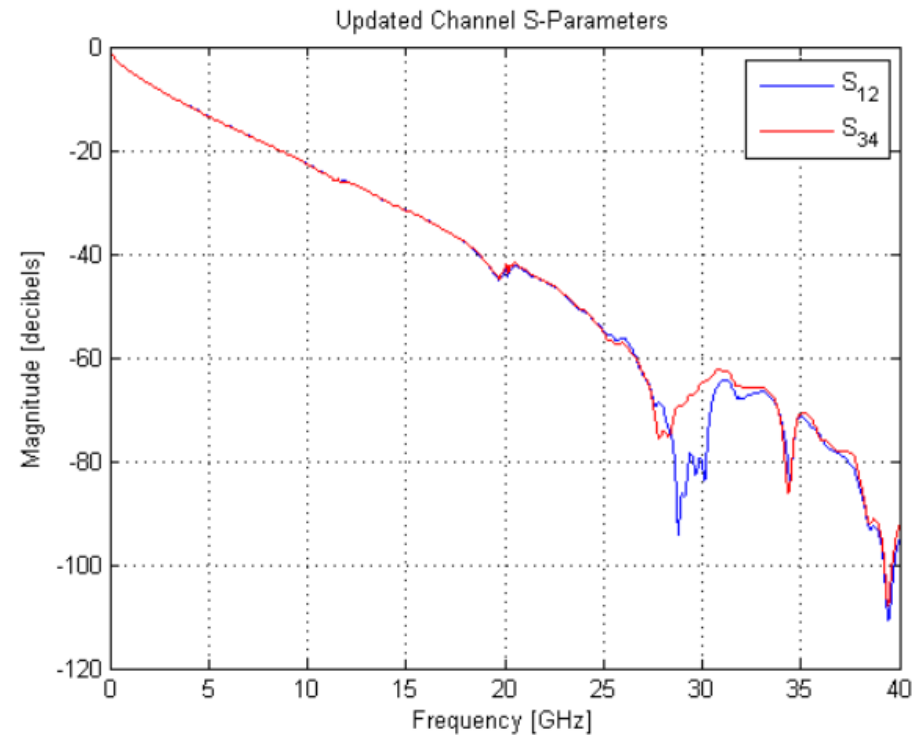
- Why?
 - Output swing is limited by headroom of design
 - Extra taps \rightarrow reduction of cursor's tap weight
- In order to account for limitations, currents must add up to equal output termination current, meaning that:

$$I \times \sum |b_i| = I \quad \Rightarrow \quad \sum |b_i| = 1$$



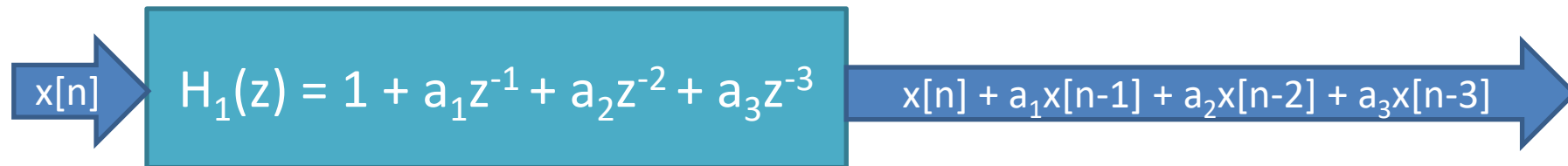
Understanding the DFE

- Continuous-Time Transfer Function of Channel (s-domain):
 - Low Pass Filter

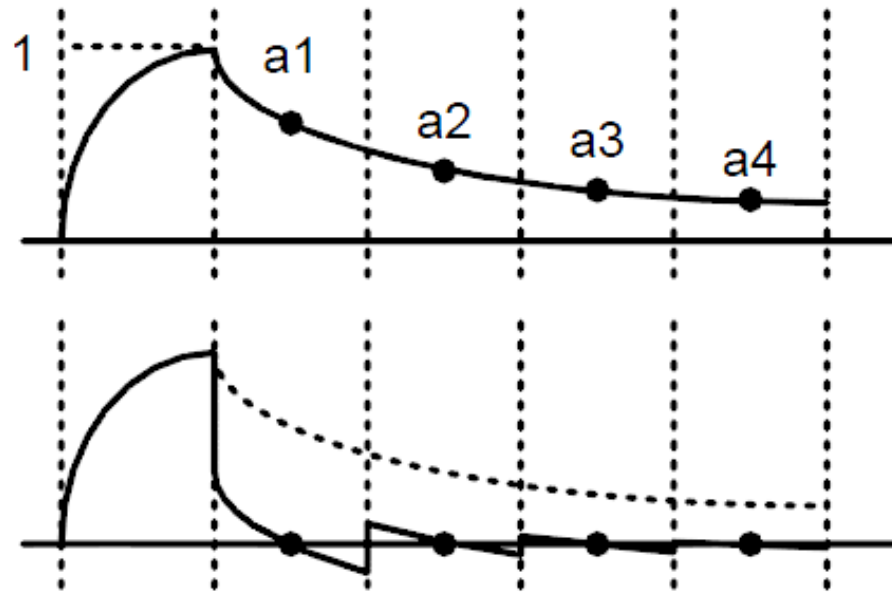
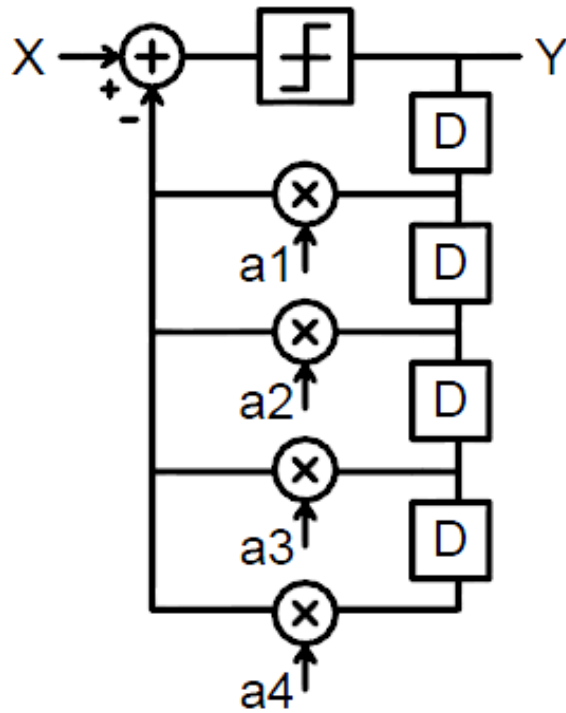


Understanding the DFE

- Discrete-Time Transfer Function of the Channel (z-domain):

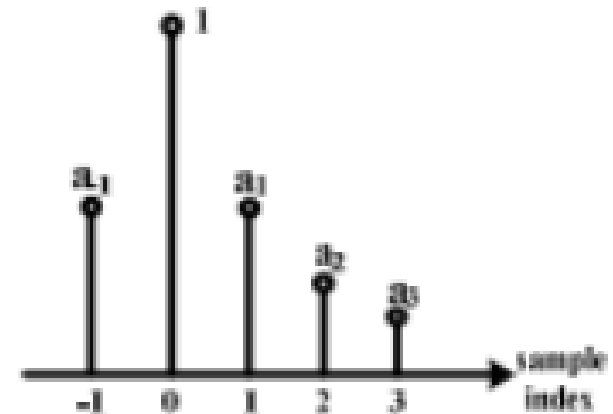
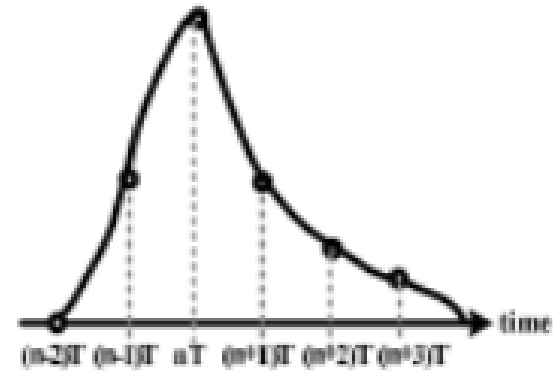


Simple DFE System

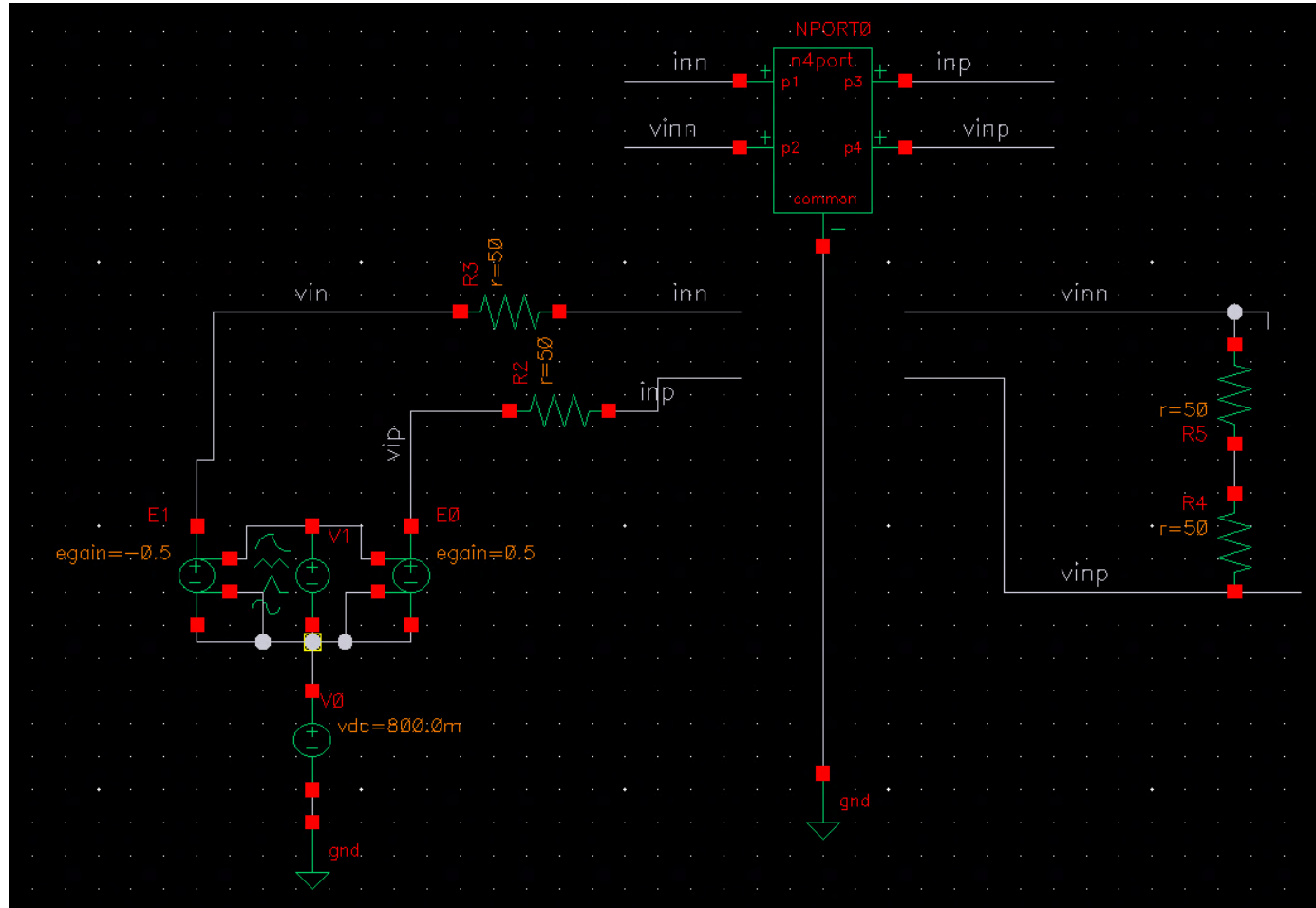


Understanding the DFE

- Pulse Response of Channel:
 - Top = Continuous Time Plot
 - Bottom = Sampled Plot

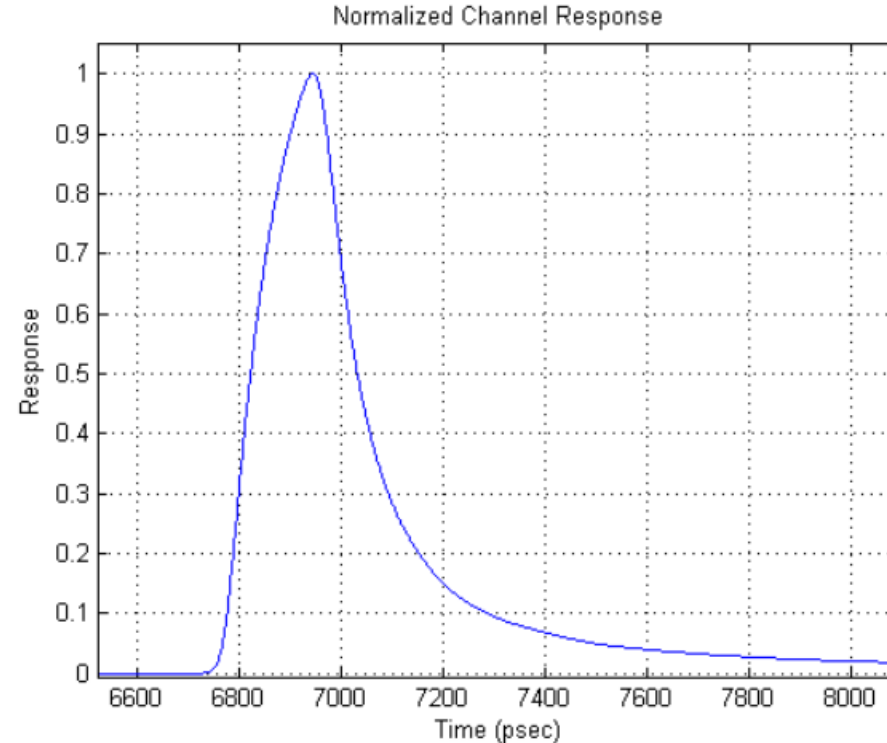


Pulse Response (Testbench)



Normalized Pulse Response

- Next, normalize the pulse response:
 - Set time of peak = $n * T$
 - Post cursors =
Response($T * (n+1)$),
Response($T * (n+2)$),
Response($T * (n+3)$), ...

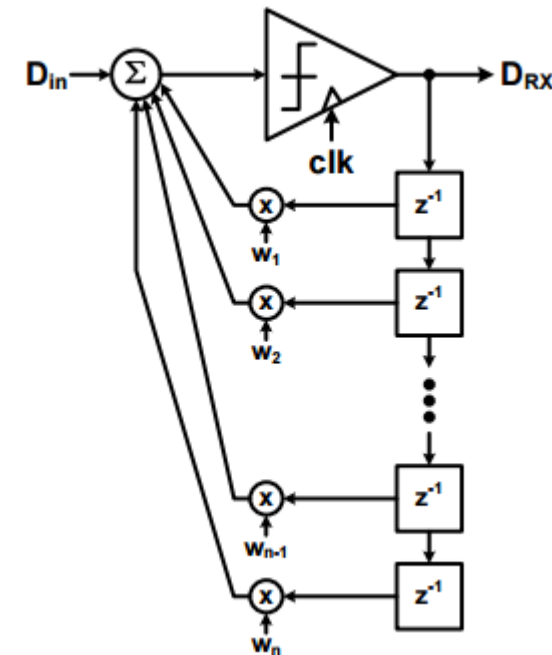


Post Cursor Calculations

- Calculated Postcursors:
 - Postcursor $a_1 = 0.2605$*
 - Postcursor $a_2 = 0.104$*
 - Postcursor $a_3 = 0.0588$*
 - Postcursor $a_4 = 0.0387$*
 - Postcursor $a_5 = 0.0284$*

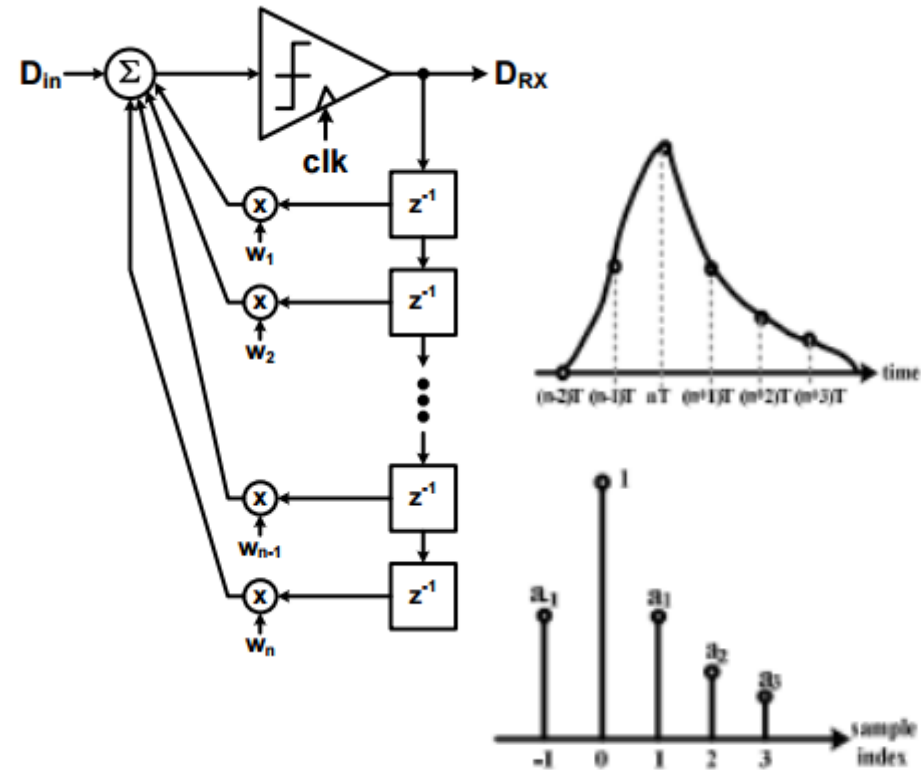
Understanding the DFE

- Objective: Negate the effects of the post-cursors ($a_1, a_2, a_3\dots$) through feedback FIR filter and accurate sampling (decision circuit)
 - Pros:
 - No amplification of noise+crosstalk
 - Can make feedback filter adaptive
 - Cons:
 - Can only account for post-cursors (no pre-cursors)
 - Critical feedback timing path



DFE Tap Coefficients

- If channel causes postcursors a_1, a_2, a_3 , etc., ...
- DFE tap coefficients must negate postcursors
- Thus, DFE tap coefficients = negative postcursors



Implementation of DFE

```
//Verilog-AMS HDL for "ece546", "dfe_sampler" "verilogams"

`include "constants.vams"
`include "disciplines.vams"

module dfe_sampler (in, inbar, out, outbar, Dout, clk, rst);
input in, inbar, clk, rst;
output reg Dout;
output out, outbar;
electrical in, inbar, out, outbar;
logic clk, rst;

parameter real tap1 = 0;
parameter real tap2 = 0;
parameter real tap3 = 0;
parameter real tap4 = 0;
parameter real tap5 = 0;

reg[4:0] data_history;

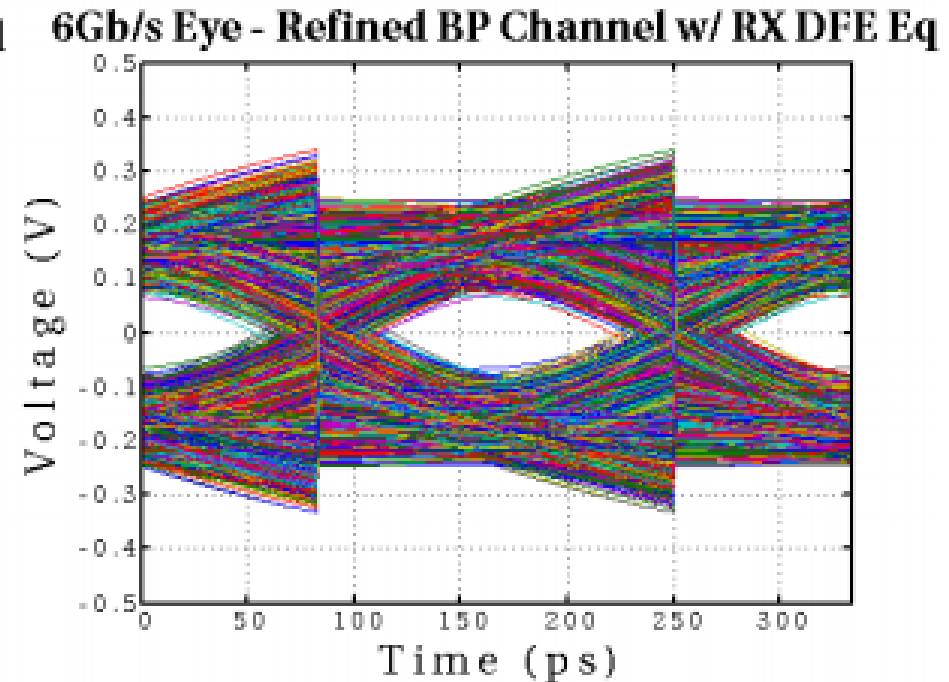
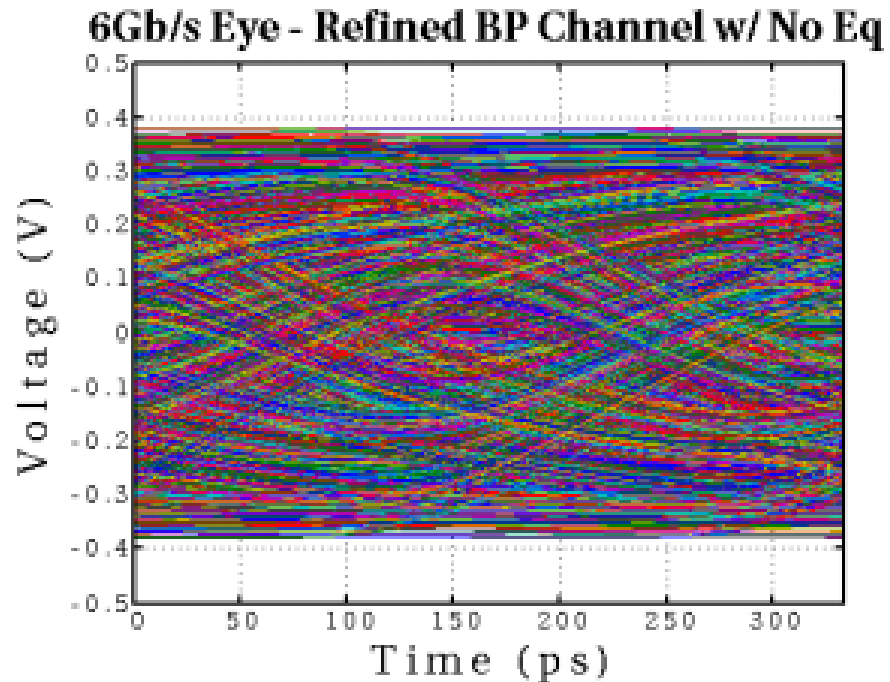
analog begin
    V(out) <+ slew(V(in)+tap1*(2*data_history[0]-1)+tap2*(2*data_history[1]-1)+tap3*(2*data_history[2]-1)+tap4*
(2*data_history[3]-1)+tap5*(2*data_history[4]-1),1e11, 1e11);
    V(outbar) <+ slew(V(inbar)-tap1*(2*data_history[0]-1)-tap2*(2*data_history[1]-1)-tap3*(2*data_history[2]-1)-tap4*
(2*data_history[3]-1)-tap5*(2*data_history[4]-1),1e11, 1e11);
end

always@(posedge(clk), rst) begin

    if(rst) begin
        data_history <= 5'b00000;
        Dout <= 1'b0;
    end
    else begin
        if(V(out) - V(outbar) > 0.2)
            Dout <= 1'b1;
        else if (V(out) - V(outbar) < -0.2)
            Dout <= 1'b0;
        data_history[4:0] = {data_history[3:0],Dout};
    end
end

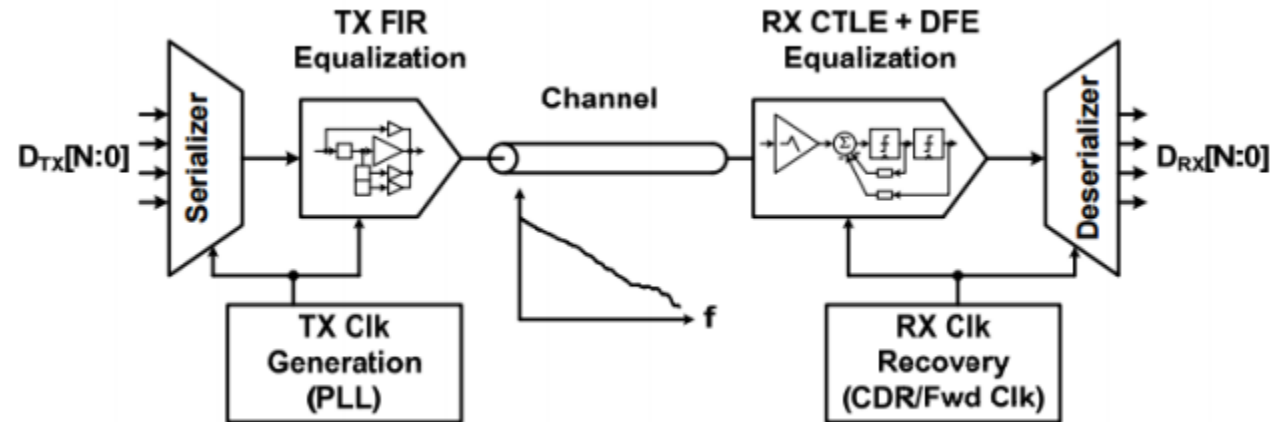
endmodule
```

Effects of DFE (Eye diagram)



MORE COMPLEX EQUALIZATION (SETUP)

- Full equalization setup with FFE + CTLE + DFE (in SERDES)



COMPLEX EQUALIZATION DESIGN PROCESS

- 1) Design CTLE to account for as much loss @ operating frequency
- 2) Design RX Driver Amp to account for remaining loss (~5-10 dB)
- 3) Analyze pulse response of channel+CTLE+RX Driver to calculate FFE coefficients (solely precursor) and test FFE behaviorally
- 4) Analyze pulse response again (no precursor this time) to determine postcursors for DFE coefficients and test DFE behaviorally