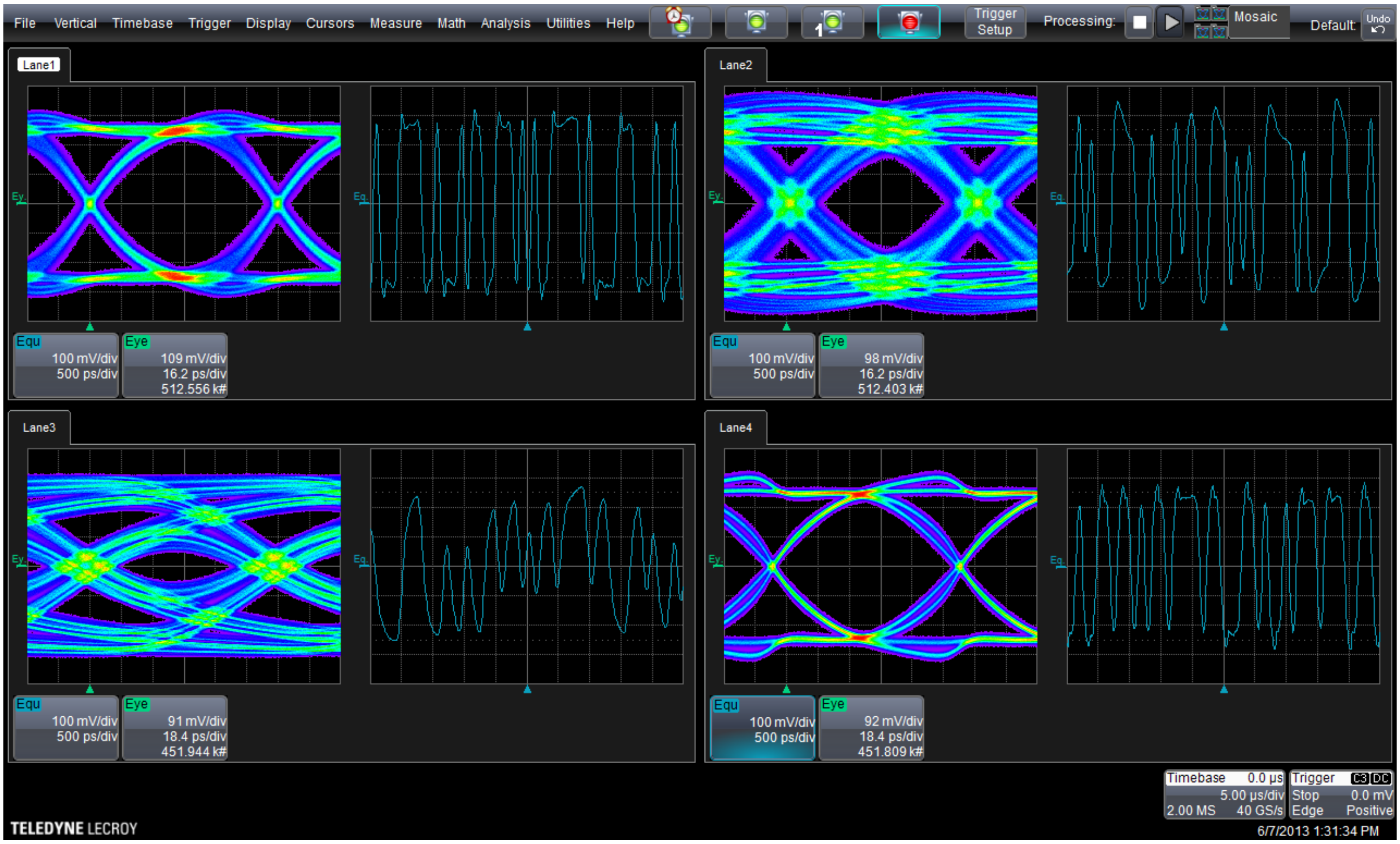
TELEDYNE LECROY
Everywhereyoulook™



Operator's Manual

Eye Doctor II Software

Eye Doctor II Software Operator's Manual

922961 Rev A
May 2013

TABLE OF CONTENTS

Eye Doctor II Overview

Eye Doctor II is a complete set of tools that allows the full-range of de-embedding, emulation and equalization on full record lengths (up to 1024 Mpts) with little impact on waveform processing time and with results integrated into the oscilloscope interface for further analysis using LeCroy's complete, deep analysis toolbox. Eye Doctor II uses industry-standard S-Parameter measurements and Touchstone 1.1 files created with Teledyne LeCroy SPARQ series network analyzers, VNAs, TDRs, simulators, LeCroy WaveExpert (to name a few), and are the universal language of signal integrity engineers. All basic capability is easily accessed in a streamlined, simple user interface.

Eye Doctor II is fully integrated into the **SDAIII-CompleteLinQ** product, which is available for WavePro 7 Zi/Zi-A, WaveMaster 8 Zi/Zi-A, LabMaster 9 Zi-A and LabMaster 10 Zi series oscilloscopes. Users of SDAIII-CompleteLinQ will want to access Eye Doctor II capabilities from the SDA Framework dialog, where they are part of the overall serial data analysis processing chain.

Note: Virtual Probing, which was part of the original Eye Doctor package and is also included in Eye Doctor II, is an advanced capability that can still be configured from the WebEditor view. Most users will prefer the newer VirtualProbe software package instead. This package provides a user-interface that is far simpler than the virtual probe capability included with Eye Doctor II.

Key Features

- **Transmitter Emphasis Emulation** - Transmitter Emphasis Emulation allows for adding or removing emphasis from signals. Using the Auto Add or Auto Remove function, enter the amount of emphasis to add or remove. Custom mode allows for specifying the specific tap values for the linear tapped delay line filter creating the emphasis. Both Pre-emphasis and De-emphasis are supported in Eye Doctor II.
- **Fixture De-embedding** - Fixture De-embedding allows for moving the reference for measurements to a point before the fixture (such as, directly to the transmitter output) by removing the effects of the fixture. A Touchstone Format S-Parameter file describing the fixture is required to de-embed.
- **Channel Emulation / De-embedding** - Channel Emulation / De-Embedding can be used to see the effect a channel is having on measurements. This works in both directions. You can emulate channel effects to see what the signal would have looks like if the measurement is taken after the serial data channel. Conversely, you can de-embed the effects of a physical channel to see what the signal looks like if the measurements are taken before the serial data channel. It is very similar to Fixture De-embedding. For both emulation and de-embedding of serial data channels, a Touchstone Format S-Parameter file describing the channel is required.
- **Receiver Equalization Emulation** - Receiver Equalization provides the ability to emulate the effects a Continuous Time Linear Equalizer, Feed Forward Equalizer, and/or Decision Feedback Equalizer has on a serial data signal. This is critical step for some serial data signals that could have a completely closed eye at the receiver input.

Accessing Eye Doctor II

From Within SDAIII-CompleteLinQ

Users of SDAIII-CompleteLinQ will find Eye Doctor II capabilities on the SDA Framework dialog:

From Analysis Menu

If you are not running an SDA oscilloscope packaged with the SDAIII option, access the Eye Doctor II dialog by choosing **Analysis > Eye Doctor II** from the menu bar.

Eye Doctor II Dialog

The Eye Doctor II dialog shows the flow of the data from the transmitter to the receiver. Each block in the flow diagram is a button that opens its corresponding dialog (e.g., touching the Signal Input block displays the Signal Input dialog).

Configure each function you wish to apply to the circuit model by selecting each block in turn, working from left to right. Only Signal Input is required.

Signal Input - set up your signal input sources and nominal bit rate.

Emphasis - add or remove pre-emphasis or de-emphasis from your serial data signal.

Emulate / De-embed

- Emulate Channel / De-embed Fixture
- De-embed Channel / De-embed Fixture
- De-embed Fixture Only

Equalizer - emulate:

- Continuous Time Linear Equalization (CTLE)
- Feed Forward Equalization (FFE)
- Decision Feedback Equalization (DFE)

Show - show/hide the output waveform results from that block on the display grid.

The buttons at the left side of the dialog enable additional functionality:

Save/Recall - lets you save your current Eye Doctor II setup or recall a previous Eye Doctor II Setup.

Turn Off Views - automatically clears all four of the **Show** checkboxes underneath each block in the flow.

Using Eye Doctor II with SDAII and SDAIII-CompleteLinQ

Besides being a very powerful yet easy to use analysis tool for fixture de-embedding and equalizer emulation, Eye Doctor II also has several output waveforms that can be processed by other applications for additional measurements:

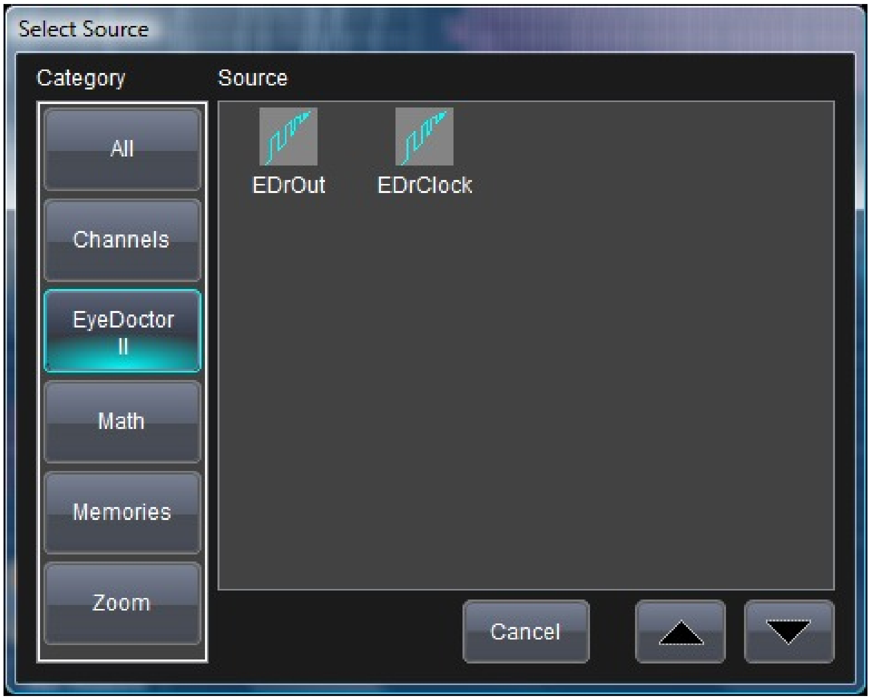
Note: The names below appear when using the path Analysis > Eye Doctor II to access the Eye Doctor functions. When accessing Eye Doctor II via the SDAIII-CompleteLinQ dialog, most of these selections appear without the "EDr" prefix. (e.g., Emp is equivalent to EDrEmp.)

- **EDrInP** - The positive, signal-ended input signal specified by Input1 in Eye Doctor II's Signal Input.
- **EDrInN** - The negative, signal-ended input signal specified by Input2 in Eye Doctor II's Signal Input.
- **EDrIn** - Either the signal-ended input signal specified by Input1 in 1 Input Mode or the differential subtraction of Input1-Input2 in Input1-Input2 mode.
- **EDrEmpP** - The positive, signal-ended signal output by the Emphasis block.
- **EDrEmpN** - The negative, signal-ended signal output by the Emphasis block.
- **EDrEmp** - The differential signal output by the Emphasis block.
- **EDrChP** - The positive, signal-ended signal output by the Emulate / De-embed block.
- **EDrChN** - The negative, signal-ended signal output by the Emulate / De-embed block.
- **EDrCh** - The differential signal output by the Emulate / De-embed block.
- **EDrCTLE** - The differential signal output from the CTLE block in the Equalizer.
- **EDrFFE** - The differential signal output from the FFE block in the Equalizer.
- **EDrOut** - The differential signal output from the Equalizer block. This includes all processing performed by Eye Doctor II.
- **EDrClock** - The recovered clock signal created by the Clock Recovery block in the Equalizer.
- **EDrResp** - The last S-Parameter file viewed using the View Response feature of the Emulate / De-embed block.

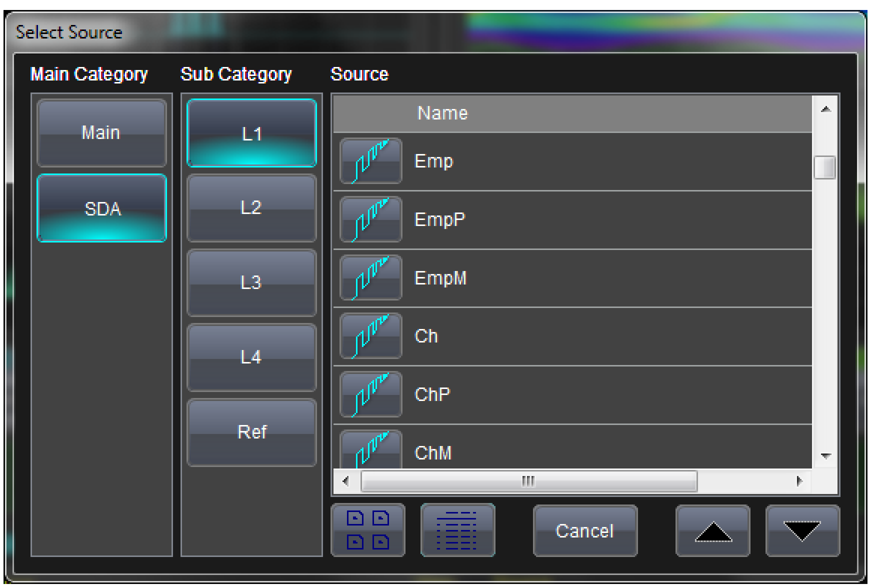
Note: All the above are available as inputs to any oscilloscope Math function or Measurement. However, only **EDrOut** and **EDrClock** are available as inputs to SDAII.

Selecting Eye Doctor II Signals

When accessing Eye Doctor II via the Analysis > Eye Doctor II path, signals appear in the Eye Doctor II catagory of the source selector dialog:



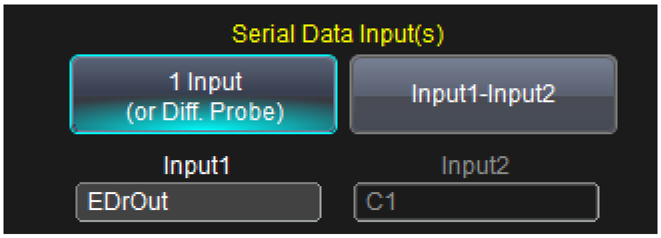
When using Eye Doctor from within SDAIII-CompleteLinQ, signals appear in the SDA category:



Input to SDAII

When using an Eye Doctor II output signal as the source input for SDAII processes, choose **Input1 EDrOut**. This is not necessary when using SDAIII-CompleteLinQ, since the Eye Doctor capabilities are integrated into the SDA signal processing flow:

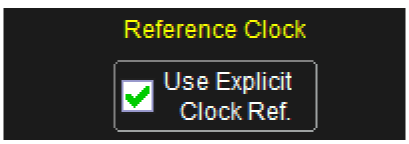
1. In the **Signal Input** dialog of SDA II, set the mode to **1 Input**.
2. Set **Input1** to be **EDrOut**.



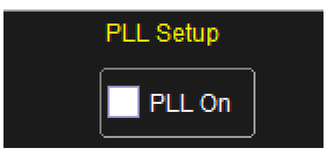
Note: This takes the output from Eye Doctor II and uses it as the input to SDA II.

When using the Eye Doctor II clock recovery capability for SDA II processes, the steps below are also required.

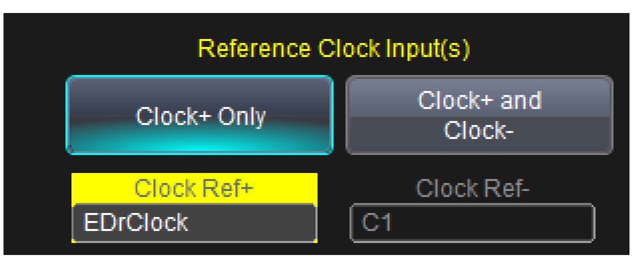
3. On the **Clock Recovery** dialog of SDA II, make sure the **Use Explicit Clock Reference** checkbox is checked.



4. Also, on the **Clock Recovery** dialog of SDA II, turn off the PLL by clearing the PLL On checkgox. (A PLL has already been used to recover the clock in the Eye Doctor II Clock Recovery section.)



5. On the **Reference Clock Input** dialog of SDA II, set the mode to **Clock+ Only**.
6. Set the **Clock Ref +** value to **EDrClock**.



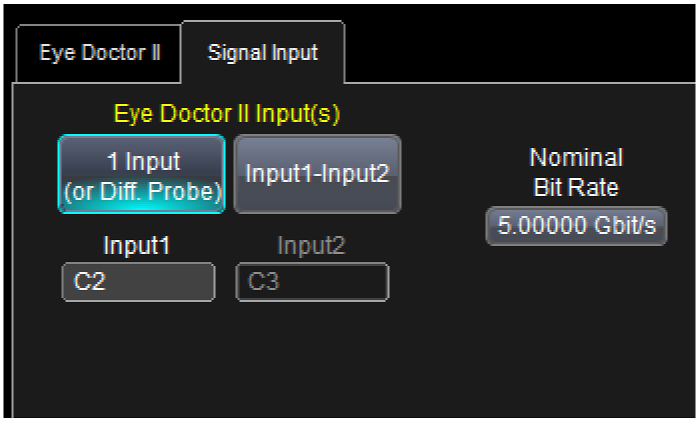
At this point, the Eye Doctor II clock and data output waveforms are set as the inputs for SDAII. Now eye and jitter analysis can be performed on the signals previously processed by Eye Doctor II.

Signal Input

Signal Input

Touch the **Signal Input** button on the **Eye Doctor II** dialog to access the **Signal Input** dialog.

Note: SDAIII-CompleteLinQ users should always access Eye Doctor II capabilities from the SDAIII-CompleteLinQ Framework dialog.



Signal Input Setup

1. Choose **Analysis > Eye Doctor II** from the menu bar.
2. On the **Eye Doctor II** dialog, touch the **Signal Input** button.
3. In the **Eye Doctor II Input(s)** section, if you are using a differential probe, touch the **1 Input (or Diff. Probe)** button. Touch the **Input1** button and select an input source from the **Select Source** popup window.

   OR

   If you are using two single-ended probes to calculate the differential signal, touch the **Input1-Input2** button. **Input2** is subtracted from **Input1**. Touch the **Input1** and **Input2** buttons and select a source for each from the **Select Source** pop-up window.

Nominal Bit Rate

The nominal bit rate must be set in order for the Emphasis and Equalizer blocks in Eye Doctor II to function properly.

Note: While the bit rate does not have to be exact, it should be within ±3% for these blocks to function correctly.

For more information, see Eye Doctor II Overview.

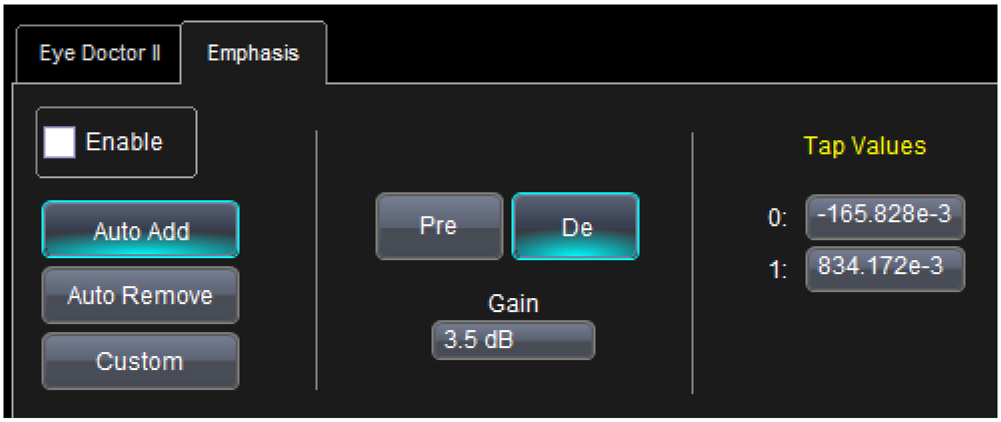
Emphasis

Emphasis Overview

Emphasis is added to a signal as a way to pre-compensate for expected channel losses. Since high frequency components are attenuated more by serial data fixtures and channels (making the high frequency components higher amplitude than the low frequency components at the far end of the serial data channel) they can be the same amplitude.

Emphasis Dialog

Touch the **Emphasis** button on the **Eye Doctor II** dialog to access the **Emphasis** dialog.



User interface for adding or removing pre-emphasis or de-emphasis

The Emphasis dialog allows you to:

- **Add Pre-emphasis** - This implements a 2-tap FFE filter that boosts the amplitude of transitional bits in your input signal by the amount **Gain** specified. See Auto Add Emphasis for more information.
- **Add De-emphasis** - This implements a 2-tap FFE filter that attenuates the amplitude of transitional bits in your input signal by the amount **Gain** specified. See Auto Add Emphasis for more information.
- **Remove Pre-emphasis** - This implements an 8-tap FFE filter that attenuates the amplitude of transitional bits in your input signal by the **Gain** amount specified. See Auto Remove Emphasis for more information.
- **Remove De-emphasis -** - This implements an 8-tap FFE filter that boosts the amplitude of transitional bits in your input signal by the **Gain** amount specified. See Auto Remove Emphasis for more information.
- Add **Custom** emphasis - Specify up to 8 tap values for a custom filter response. See Custom Emphasis for more information.

Checking or unchecking the **Enable** checkbox allows you to quickly enable or disable the **Emphasis** block. Make sure to have the Nominal Bitrate set to the correct value in the **Signal** tab (SDAIII users) or in the **EyeDoctor II** tab when configuring via **Analysis --> EyeDoctor II** (SDAII users).

Emphasis Implementation

Emphasis is implemented as a tapped delay line filter, with 2 taps for auto-add and 8-taps for auto-remove. For more information on how the filter is computed.

EyeDoctor II's Emphasis feature incorporates algorithms to automatically determine tap values for use in a tapped delay line filter. The filter is equivalent to Feed Forward Equalization (FFE), (without user-selectable precursor taps), and is mathematically described as follows:

$$y_{out}(t) = \sum_{k=0}^{k=N} C_k \cdot y_{in}(t - kT)$$

Where:

- y_{in}**(t)** is the initial waveform
- y_{out}***(t)*** is the transformed waveform
- ***N*** is the number of taps, which is either 2 or 8
- ***k*** is the tap index, ranging from 0 to N
- C_k are the coefficients, or tap values
- ***T*** is the time delay for each of the delay elements (nominally 1/bitrate)

Take care to have the nominal bitrate set correctly. The bitrate is configure in the .

Auto Add Emphasis

Auto Add Emphasis automatically determines the appropriate tap values for your input signal when the amount of emphasis to add is specified using the **Gain** box. It implements a 2-tap FFE filter that boosts the amplitude of transitional bits in your input signal by the amount **Gain** specified.

Either pre-emphasis or de-emphasis can be automatically added. Use this feature when wanting to add pre-emphasis or de-emphasis to an signal that is not emphasized in order to emulate the behavior of a transmitter that nominally includes emphasis.

The image below shows SDAIII-CompleteLinQ performing simultaneous multi-lane analysis to illustrate the effect of adding pre-emphasis and de-emphasis. Three lanes are being analyzed, with each lane containing a simulated 5Gbps NRZ pattern configured on F1 using the JitSim function.

Examples of Auto Add Pre-emphasis (center) and De-emphasis (right) along with no emphasis (left).

The analysis for each lane shows three plots Eye diagrams(top grids), and two waveforms. The waveforms are as follows:

- The blue waveform is a zoom of the **SDAInput** trace, which is the output of the Signal step in the SDA processing chain. (In this case, it is a copy of F1, but if we were using two inputs, such as C1 and C2 to receive a differential signal, SDAInput would be C2-C1.)
- The orange waveform is a zoom of the **Emp** trace, which shows the output of the Emphasis block.

Lane1 (left) is configured with emphasis turned off, and is shown for reference. (The **Emp** trace is equal to **SDAInput**).

Lane2 (center) is configured to add pre-emphasis. The **Emp** trace shows a gain in the amplitude for transitional bits (the first bit after a rising or falling edge). The eye diagram shows multiple levels as a result of the added pre-emphasis, with the emphasized bits causing an increase in the overall peak-peak voltage.

Lane3 (right) is configured to add de-emphasis. The **Emp** trace shows a attenuation for all non-transitional bits (bits that are not the first bit after a rising or falling edge). The eye diagram shows multiple levels as a result of the added de-emphasis, with the emphasis appearing within the un-emphasized eye such that the overall peak-peak voltage is not increased.

Auto Remove Emphasis

Auto Remove Emphasis automatically determines the appropriate tap values for your input signal when the amount of emphasis to remove is specified in the **Gain** box.

Either pre-emphasis or de-emphasis can be automatically removed. Use this feature when seeking to make jitter and noise measurements on emphasized signals that have not been "naturally" de-emphasized by passing through a channel.

- **Remove Pre-emphasis** - This implements an 8-tap FFE filter that attenuates the amplitude of transitional bits in your input signal input signal by the **Gain** amount specified.
- **Remove De-emphasis -** - This implements an 8-tap FFE filter that boosts the amplitude of transitional bits in your input signal by the **Gain** amount specified.

The image below uses SDAIII-CompleteLinQ ability to perform multi-lane analysis simultaneously to illustrate the effect of removing pre-emphasis and de-emphasis. It shows three lanes being analyzed, with each lane containing a simulated 5Gbps NRZ pattern configured on F1 using the JitSim function.

Auto-remove of pre-emphasis (center) and de-emphasis (right) along with no emphasis (left).

The analysis for each lane shows three plots Eye diagrams(top grids), and two waveforms. The waveforms are as follows:

- The blue waveform is a zoom of the **SDAInput** trace, which is the output of the Signal step in the SDA processing chain. (In this case, it is a copy of F1, but if we were using two inputs, such as C1 and C2 to receive a differential signal, SDAInput would be C2-C1.)
- The orange waveform is a zoom of the **Emp** trace, which shows the output of the Emphasis block.

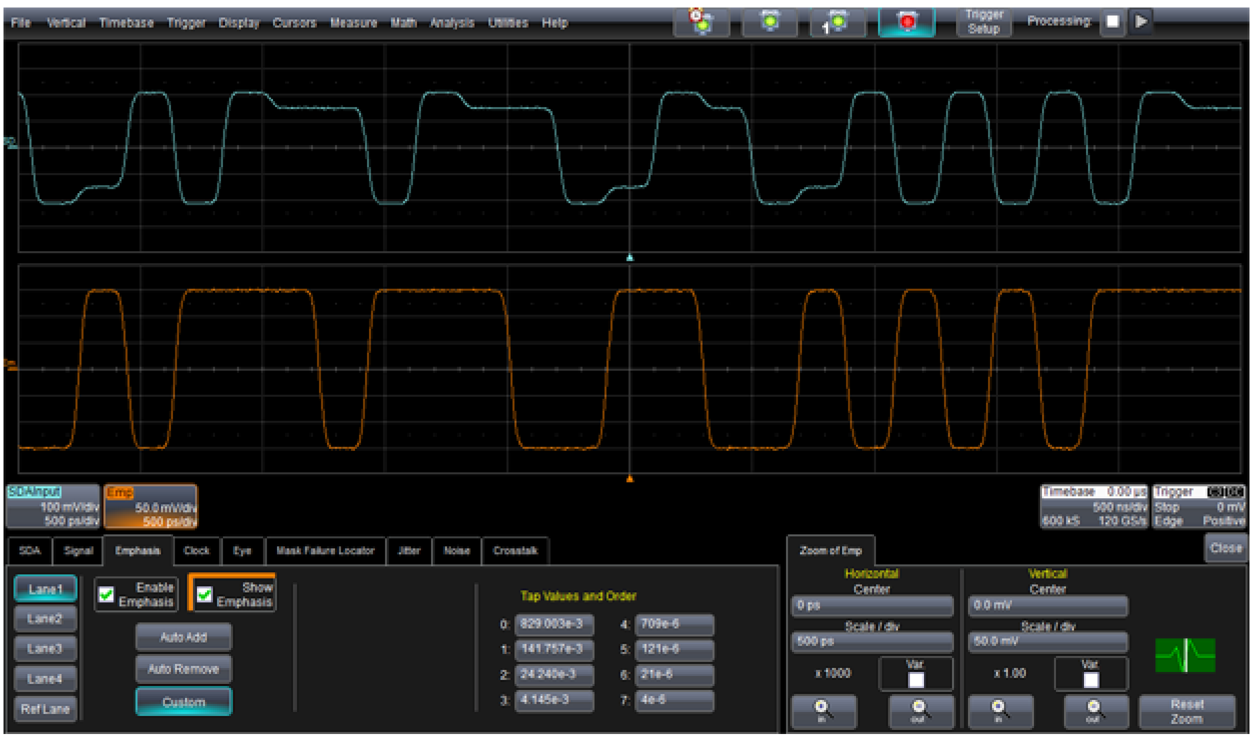
Lane1 (left) is configured with emphasis removal turned off, and is shown for reference. (The **Emp** trace is equal to **SDAInput**). Note that the **Emp** trace shows emphasis: bits after a rising transition are higher, and after a falling transition are lower than successive bits. The eye diagram also shows the effects of added emphasis.

Lane2 (center) is configured to remove pre-emphasis. The **Emp** trace shows that the emphasized levels of the transitional bits (the first bit after a rising or falling edge) have been attenuated to meet the voltage levels of the non-transitional bits.

Lane2 (center) is configured to remove de-emphasis. The **Emp** trace shows that the de-emphasized levels of the non-transitional bits (bits that are not the first bit after a rising or falling edge) have been amplified to meet the voltage levels of the transitional bits.

Custom Emphasis

Custom emphasis allows you to enter up to 8 tap values for a custom FFEtapped delay line filter.

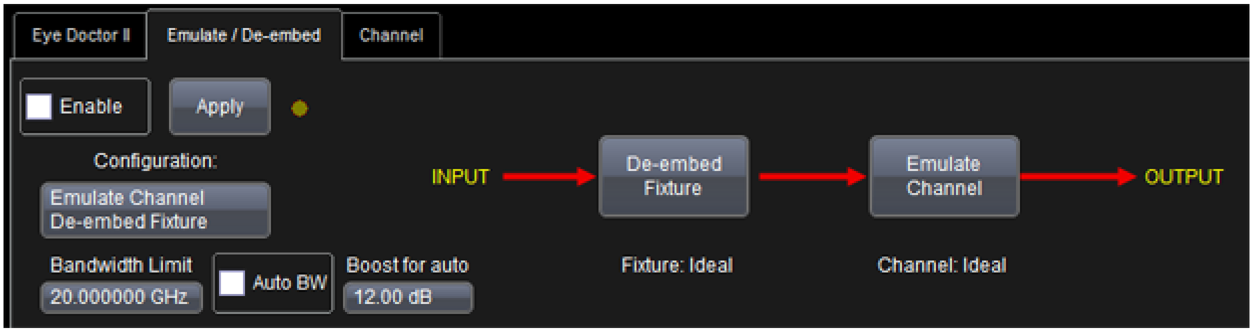


Custom emphasis setup

Emulate / De-embed

Emulate / De-embed Overview

Touch the **Emulate / De-embed** button on the **Eye Doctor II** dialog to access the **Emulate / De-embed** dialog.

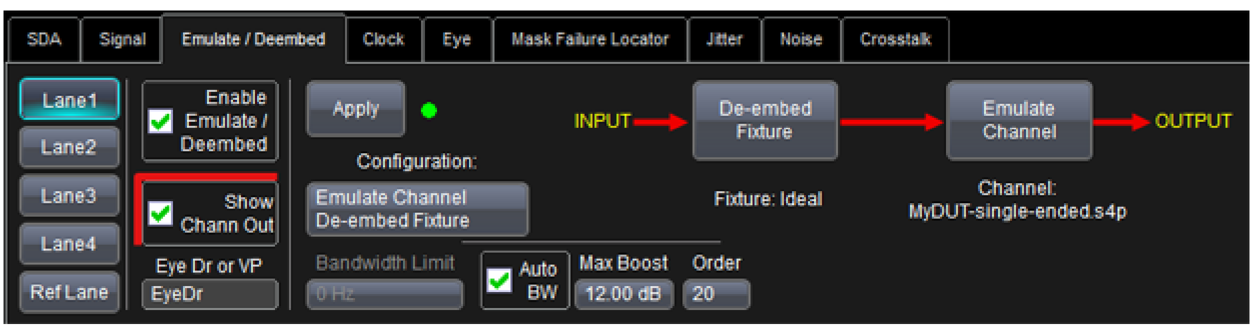


The **Emulate / De-embed** block allows you to De-embed Fixtures, De-embed Serial Data Channels, and Emulate Serial Data Channels.

- **Fixture De-embedding** - Use Fixture De-embedding to move the reference for your measurement to a point before the fixture (such as directly to the transmitter output) by removing the effects of the fixture. A Touchstone Format S-Parameter file describing your fixture is required to de-embed.
- **Channel Emulation / De-embedding** - Use Channel Emulation / De-Embedding to move the reference for your measurement to a different point in your system. You can either move the reference to a point after a serial data channel by emulating the channel effects, or to a point before a serial data channel by de-embedding the channel (similar to Fixture De-embedding). For both emulation and de-embedding of serial data channels, a Touchstone Format S-Parameter file describing your channel is required.

Emulate / De-embed Within SDAIII-CompleteLinQ

When accessed through SDAIII-CompleteLinQ, the Emulate / De-embed dialog will look like this:



If you also have the Virtual Probe package, the **EyeDr or VP** selector appears at the bottom of the dialog. This allows you to choose whether to use the EyeDr. user interface or the Virtual Probe user interface to configure de-embedding and emulation.

You can configure emulation and de-embedding on up-to-four lanes independently when using SDAIII-CompleteLinQ.

Emulate / De-embed Setup

1. Choose a Configuration.
2. Touch the **De-Embed Fixture**, **Emulate Channel** and/or **De-embed Channel** buttons to specify the S-Parameters for the fixtures and channels. See the Fixture and Channel Dialog for more information.
3. Specify a Bandwidth Limit or check the **Auto BW** checkbox and specify a **Boost for Auto** amount.
4. Touch **Apply**.

Note:Any time you make a change in the Emulate / De-embed dialog, click Apply again to update the associated processing. If Apply is not clicked, the processing defined from the last time the Apply button was clicked remains unchanged and in use.

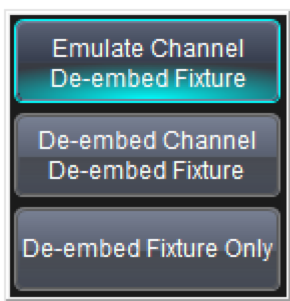
PLEASE NOTE THE FOLLOWING:

- Bandwidth Limit is only required if the chosen configuration requires de-embedding. For example, if you choose the Emulate Channel / De-embed Fixture configuration, but leave the de-embed fixture block as "ideal," the bandwidth limit has no effect on the signal.
- The indicator light next to the Apply button turns green if everything is properly setup. If the indicator does not turn green, something is wrong with the S-Parameter file or port assignment. Use the View Response functionality to help debug your issue.
- For more information on the De-embed Fixture, Emulate Channel or De-embed Channel, refer to Fixture and Channel Dialog.

For more information, see Eye Doctor II Overview.

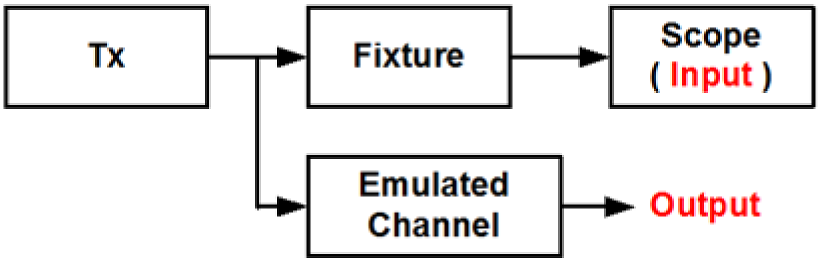
Configuration

The **Configuration** selection box in the **Emulate / De-embed** dialog allows you to choose from the following three configurations:



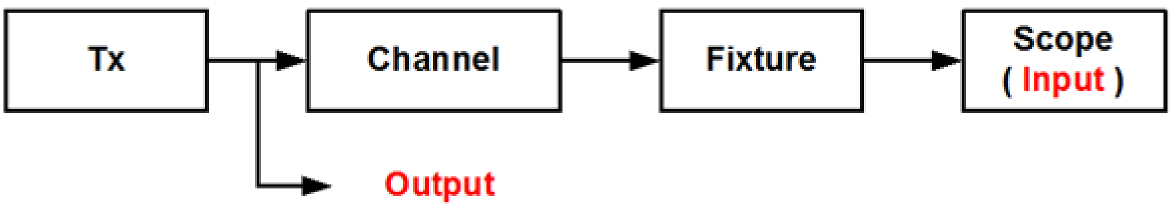
Emulate Channel / De-embed Fixture

The **Emulate Channel / De-embed Fixture** configuration allows you to both de-embed a fixture and emulate a serial data channel at the same time. This is used when you are using a fixture to measure at the transmitter side of the serial data channel. It first moves the measurement reference to the output of the transmitter, and then moves the reference to the far side of a serial data channel. This allows for emulating the effects of the serial data channel on your signal with the fixture effects removed.



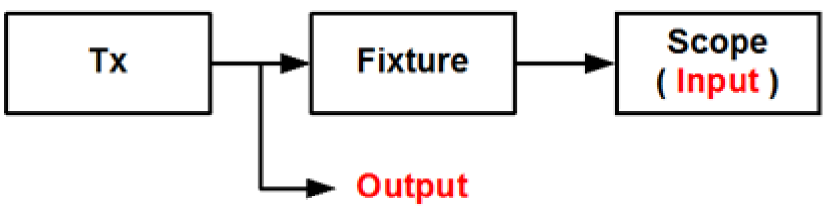
De-embed Channel / De-embed Fixture

The De-embed Channel / De-embed Fixture configuration allows to you both de-embed a fixture and de-embed a serial data channel at the same time. This is used when using a fixture to measure at the receiver side of the serial data channel. It moves the measurement reference back to the output of the transmitter. You then see what the serial data signal looked like at the transmitters output with the effects of the channel and fixture removed.



De-embed Fixture Only

The De-embed Fixture Only configuration allows you to remove the fixture effects from you measurements. You can then see what the serial data signal looked like at the transmitters output with the effects of the fixture removed.



Bandwidth Limit

When De-embedding fixture or channels a bandwidth limit is required and must be set.

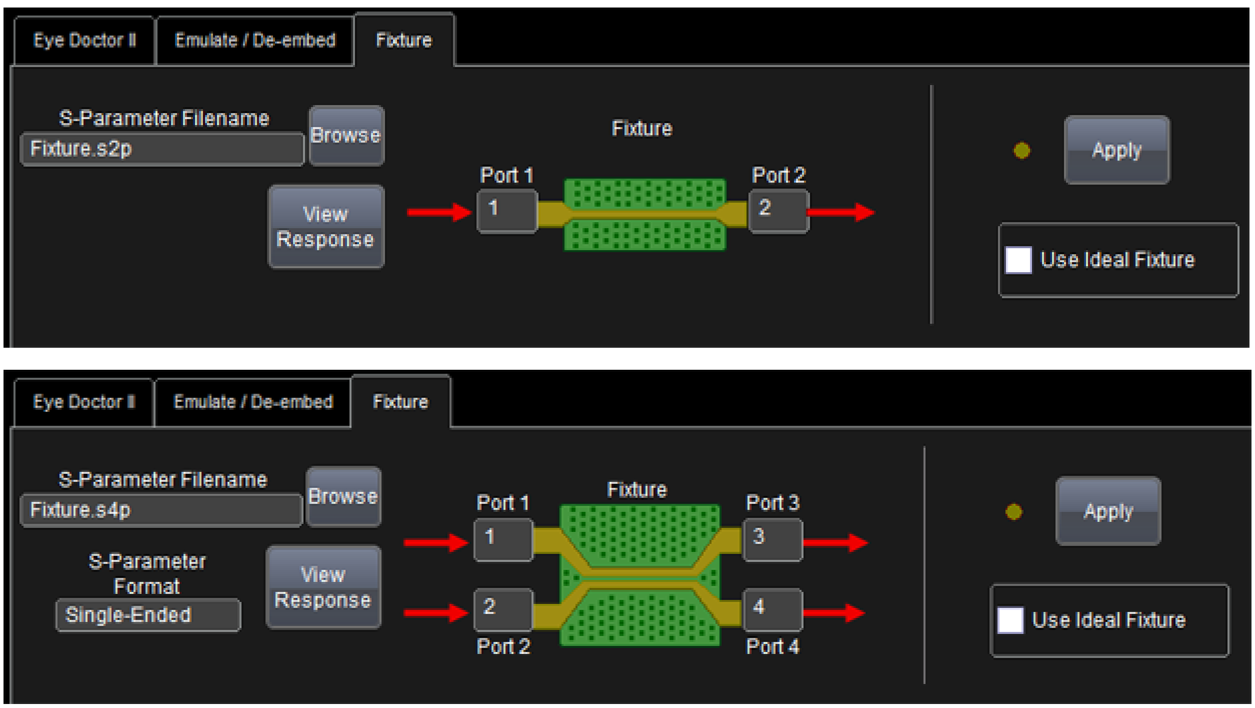
Bandwidth Limit imposes a reasonably sharp low pass filter in addition to the S-Parameter system response. This is useful when de-embedding a lossy channel, to limit the amount of boost applied. When a channel is de-embedded, high frequency response must be boosted to compensate for the high frequency attenuation in the channel. However, if the signal has been attenuated into the noise floor, boosting the signal on the oscilloscope makes it impossible to distinguish between the signal and the noise. The system boosts the noise along with the signal. The Bandwidth Limit setting can limit the overall response to the lower frequencies where signal components are detectable above the noise. If this value is set to zero (the default) then no bandwidth limit is applied.

Auto BW and **Boost for Auto** have the same effect as Bandwidth Limit but instead of setting the bandwidth, you set the maximum boost allowed. Eye Doctor II looks at the S- parameter responses and sets up the low pass filter (see Bandwidth Limit above) at the frequency where one of the outputs has more boost than the specified Boost for Auto. Boost for Auto can be set between 3 dB and 20 dB. The bandwidth in use is reported back to the user.

Note: When emulating a passive fixture or channel, the S-Parameter system does not have any boost, and the bandwidth in use is the highest bandwidth possible.

Fixture and Channel Dialogs

Use the **De-embed Fixture**, **De-embed Channel** or **Emulate Channel** buttons on the Emulate / De-embed dialog to access the Fixture and Channel dialogs. These dialogs have identical functionality and differ only in name. You will see the single-ended or differential version of the dialog depending on the Input selection made on the Signal Input dialog.



Single-ended and differential input versions of the Fixture dialog.
The Channel dialog differs only in name.

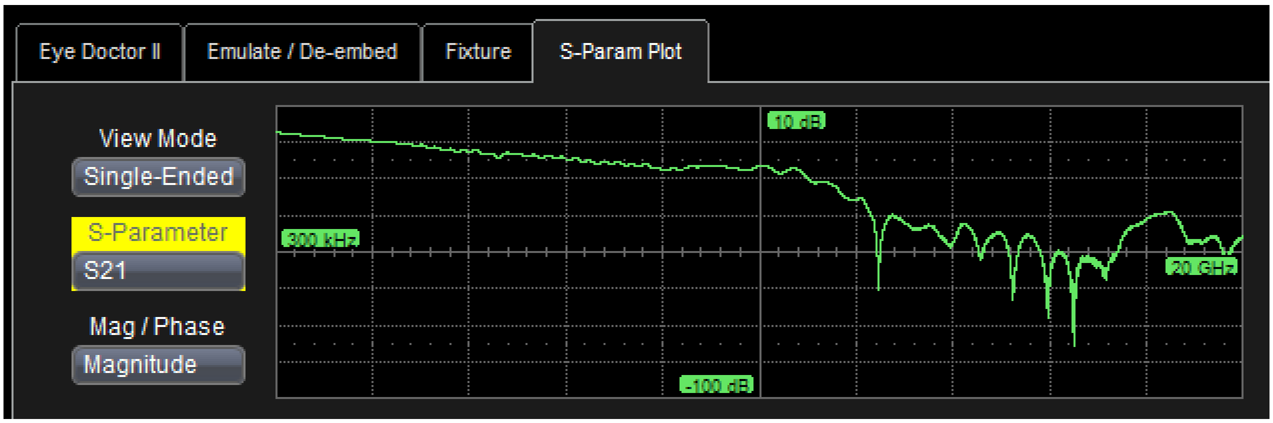
1. Check the **Use Ideal Fixture** or **Use Ideal Channel** checkbox to use a perfect fixture or channel with no loss.
2. Touch **Browse** to select the Touchstone format S-Parameter file representing the fixture or channel.
3. If using Differential inputs, choose the **S-Parameter Format**, either **Single-Ended** or **Mixed Mode.**
4. Assign to each **Port** a data column from the selected S-Parameter file. (This is useful for re-mapping from the same S-Parameter file.)
5. Check the status of the file mapping. The indicator next to the Apply button should be green.

6. View Response to plot the characteristics of the fixture or channel as it will appear after import.
7. Touch **Apply** to build the configuration.

Note: S-Parameter files should cover the frequency range up to (at least) 1/2 the oscilloscope's sample rate. For example, if the S-Parameter file only covers up to 10 GHz, then the oscilloscope sample rate should be 20 GS/s. When the fixture or channel is being emulated, there is little effect. However, if the fixture or channel is being de-embedded, then the S-Parameter matrix must be inverted. If it does not have data up to half the oscilloscope's sample rate it can cause problems. For this reason, this requirement is particularly important when de-embedding.

View Response

The **View Response** button on the Fixture and Channel Menus allows you to view a magnitude or phase plot for a selected S-Parameter following Touchstone file import and association with a channel or fixture.



1. Choose which **S-Parameter** (row from the original file) to view.
2. Choose to view the **Magnitude** or **Phase** of that S-Parameter.

PLEASE NOTE THE FOLLOWING:

- For 4 port S-Parameter files, you can view either the Single-Ended or Mixed Mode S-Parameters using the **View Mode** control.
- You can use a Math trace to Zoom the S-Parameter plot last viewed. Do this by selecting **EDrResp** as the source for the zoom function. This enables you to view a S-Parameter plot on the full grid and use cursors on the trace.

Touchstone File Format

Touchstone is a file format used for specifying S-parameters. It is a standard that is commonly adopted by vector network analyzers (VNAs), time-domain network analyzers (TDNA) and EDA tools such as microwave simulators.

Touchstone files must meet the following specification:

- The extension must be in the form **.s[N]p** where [N] corresponds to the number of ports in the device. For example, a two port S-parameter file has an extension ".s2p".
- Lines cannot be longer than 2000 characters.

- Lines beginning with **!** are comment lines. Comments must be at the front of the file. No comments are allowed once the frequencies and S-parameter values start.
- There should be at least one line that begins with '#'. This line is a line that contains tokens that help interpret the S-parameters. Valid tokens are:
  - **Hz** – frequencies are in Hz.
  - **MHz** – frequencies are in MHz.
  - **GHz** – frequencies are in GHz.
  - **MA** – S-parameters are in magnitude/angle form where the magnitudes are true magnitudes and the angle is in degrees.
  - **RI** – S-parameters are in real/imaginary form.
  - **DB** – S-parameters are in magnitude/angle form where the magnitude is in decibels and the angle is in degrees.
- If the above tokens are not found, the frequencies are assumed to be in MHz, and the S-parameters are assumed to be in magnitude/angle form.
- The S-parameters for each frequency are assumed to be listed as:

  [frequency] (S11) (S12) …(S1[N])(S21) (S22) … (S2[N]) … (S[N]1) (S[N]2) … (S[N][N])

  where N is the number of ports dictated by the file extension, except for two-port S-parameters which are assumed to be listed as [frequency] (S11) (S21) (S12) (S22) according to the standard.
- All S-parameters are assumed to be 50 Ohm. In S-parameter files, on the line beginning with '#', usually there is a token pair: R [Z0] where [Z0] is the characteristic impedance. We assume Z0 is 50 and ignore these tokens.
- All frequencies are assumed to go from the first frequency listed to the last frequency listed with constant frequency spacing.

Notes About S-parameter File Interpretation And Usage

S-parameters provided to the virtual probe will tend to have various frequency scales, differing in both resolution (frequency spacing) and range (start frequency and end frequency). The virtual probe component will resample all S-parameter sets onto a new frequency scale dictated by the system sample rate (half the sample rate will be the last frequency) and the time length (see Time Length) – where the time length dictates the new frequency resolution.

In performing this resampling, it is important to understand that the resolution may be higher or lower than the original resolution and that the frequency range may require extrapolation of points. When the frequency range is higher than that provided in the S-parameter data (i.e. the S-parameters are not provided to half the system sample rate), then the S-parameter data will be set to zero at these extra required data points. Furthermore, most S-parameter data does not go to DC, because VNAs don't measure data to DC, and therefore the DC response is also extrapolated by setting the low frequency points equal to the first frequency provided in the S-parameter data.

Therefore, it is helpful to follow some simple rules to maximize the effectiveness of using a touchstone file:

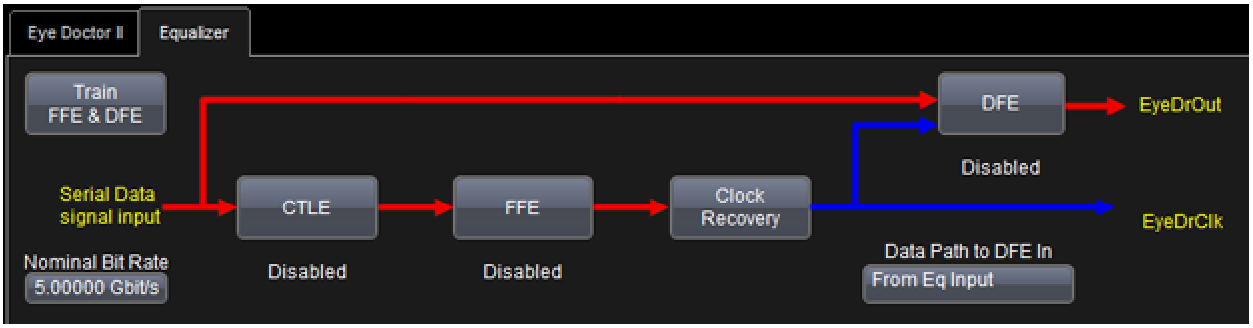
- Take S-parameter data as low in frequency as the VNA allows to minimize DC extrapolation errors.
- Take S-parameter data up to one half of the system sample rate. In other words, if you are going to supply waveforms to the virtual probe at 40 GS/s, it is good to provide data up to 20 GHz. Otherwise, if the frequency content of the signal provided to the system is much lower, then it is unnecessary to sample at a high sample rate anyway.
- Take S-parameter data with sufficient frequency resolution. The resolution that needs to be provided depends on the electrical lengths of the circuit elements involved. In other words, if you have a cable that is 5 ns long, and there are reflections going on in the system, then taking data at maybe one tenth the minimum frequency resolution of 1/5 ns = 200 MHz (like 20 MHz) is a good idea. Said differently, in a more practical manner, if there are lots of bumps and wiggles in the S-parameters that are narrow in frequency, then you will need to measure S-parameter data with an appropriate frequency resolution to capture these bumps and wiggles. Usually the narrowness of the bumps and wiggles becomes smaller when the electrical lengths of circuit elements becomes longer.

Equalizer

Equalizer Overview

EyeDoctor II includes Continuous Time Linear Equalizer (CTLE), Feed Forward Equalizer (FFE) and Decision Feedback Equalizer (DFE) capabilities, in addition to the Emphasis feature (which is also referred to as transmitter equalization.) This section covers CTLE, FFE and DFE equalization; Emphasis is covered in its own section.

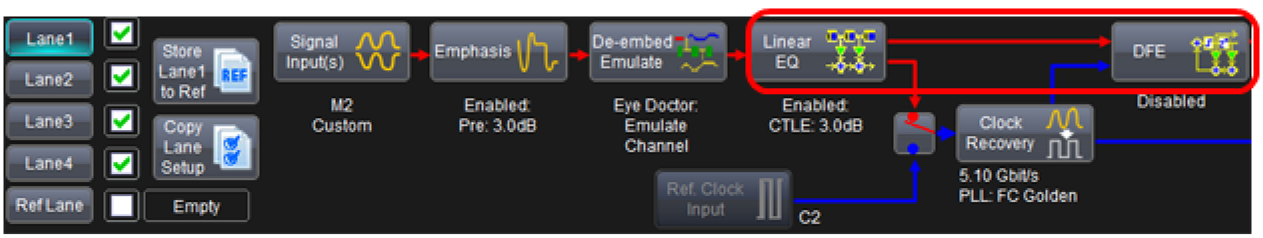
Choose **Analysis > Eye Doctor II** from the instrument menu bar, then touch the **Equalizer** button on the Eye Doctor II dialog to access the Equalizer dialog.



Equalizer user-interface when accessed via Analysis > Eye Doctor II

Using Equalizers Within SDAIII-CompleteLinQ

When configuring EyeDoctorII equalizers via the SDAIII-CompleteLinQ dialog, equalization is performed as part of the analysis flow for serial data analysis. This is the preferred approach, since all configuration can be done via the SDAIII-CompleteLinQ dialog.



The equalization features of are accessed from two separate buttons in the SDAIII-CompleteLinQ user interface:

- **Linear EQ** button: accesses Continuous Time Linear Equilization (CTLE) and Feed-Foreward Equalization (FFE) features
- **DFE** button: accesses the Decision-Feedback Equilization (DFE) dialog.

Equalization can be applied to up-to-four lanes independently when using SDAIII-CompleteLinQ .

Equalizer Settings

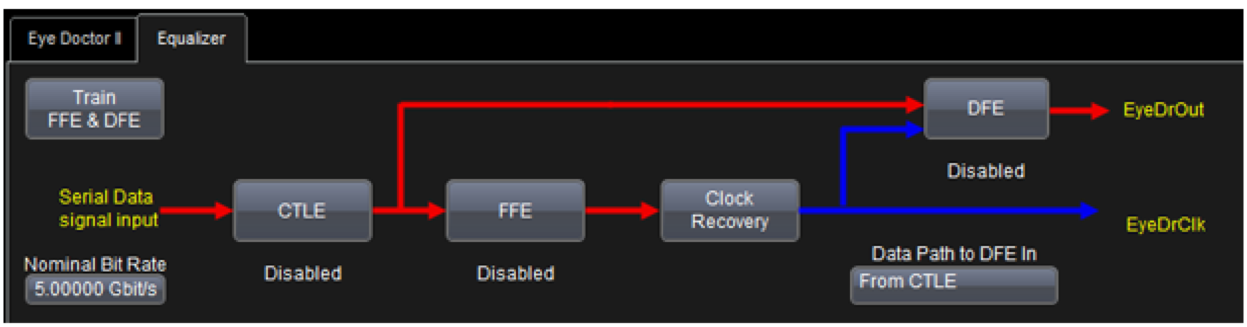
The Equalizer feature allows you to see the waveform after equalization. Since Decision Feedback Equalization requires a clock in order to work, Clock Recovery is available from the Eye Doctor II Equalizer dialog.

The Eye Doctor II Equalizer dialog contains several user interface elements in addition to buttons for configuring CTLE, FFE, Clock Recovery and DFE. (When using SDAIII-CompleteLinQ, these items appear in different places, as described.)

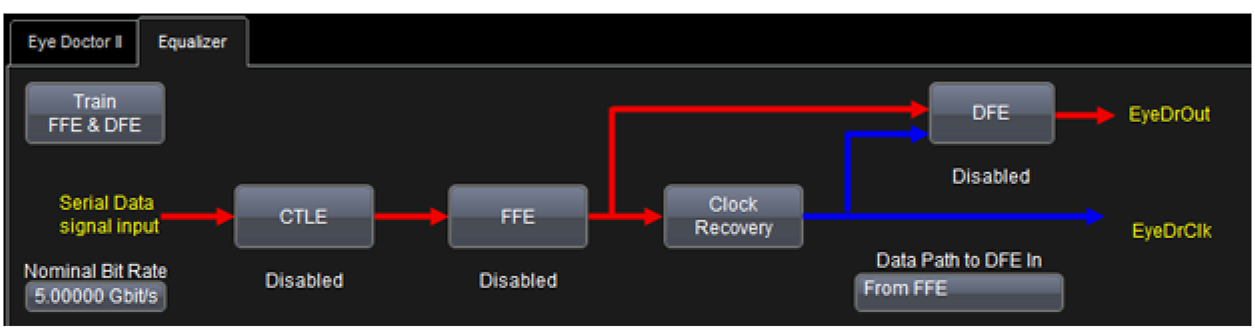
The **Train FFE & DFE** button automatically trains the FFE and the DFE (if enabled) and finds the bit rate for clock recovery. The automatic training of the FFE and/or DFE can be disabled with the respective dialog boxes. When using EyeDoctor II equalization capabilities from within SDAIII-CompleteLinQ, use the **Train** buttons that are contained within the **FFE** and **DFE** configurations screens.

The **Nominal Bit Rate** shown in this dialog sets the bitrate used for Emphasis, FFE and DFE. When using SDA II or SDAIII-CompleteLinQ, this value is inherited from the Nominal Rate configured in the Signal dialog.

The **Data Path to DFE In** selection box allows the user to decide if the data path to the DFE should come directly from the serial data signal input, or should first pass through the CTLE, or should first pass through the CTLE and FFE (as shown in the following images). This provides the ability to perform CTLE and/or FFE equalization on the signal prior to the DFE equalization. Additionally, this allows the CTLE and/or FFE equalization to be performed on the signal used for clock recovery and only DFE used on the actual data signal. (When using SDAIII-CompleteLinQ, this selector is configured in the Linear EQ dialog.)



CTLE equalizer applied to data only, CTLE and FFE applied to clock recovery.



CTLE and FFE equalizers applied to both data and clock recovery.

There are two output waveforms from the Equalizer section of Eye Doctor II. These are **EyeDrOut** and **EyeDrClk**. The signals represent the equalized data signal and the recovered clock signal. These outputs can be used as inputs to SDA II to perform eye diagram and jitter analysis. These are not used when using equalization within SDAIII-CompleteLinQ, since equalization is simply a step in the overall serial data analysis flow when using SDAIII-CompleteLinQ.

Note to SDAII Users: We strongly recommend that you use **EyeDrClk** as the explicit clock reference waveform in SDAII in order to properly analyze an unequalized signal in SDA II. SDA II cannot recover a clock from a severely degraded signal; Eye Doctor II must recover the clock after equalizing the signal, and the SDA II must use EyeDrClk to show the nearly closed eye.

For detailed information on training the equalizers in Eye Doctor II, see How to Train Eye Doctor II Equalizers.

Continuous Time Linear Equalization (CTLE)

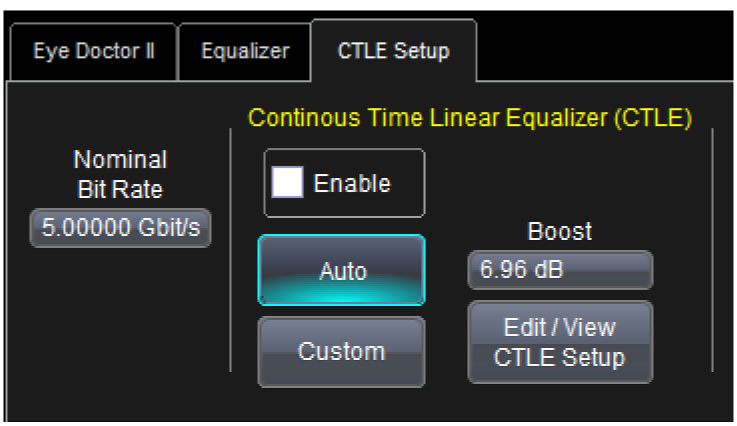
A first order Continuous Time Linear Equalizer (CTLE) is implemented in your Eye Doctor II software. The reference equalization algorithm for PCI-Express 3.0 and for SuperSpeed USB is a first order CTLE. The default settings for DC Gain and Boost for our implementation are the settings for SuperSpeed USB.

The CTLE is defined by DC Gain and the placement of 2 Poles and 1 Zero.

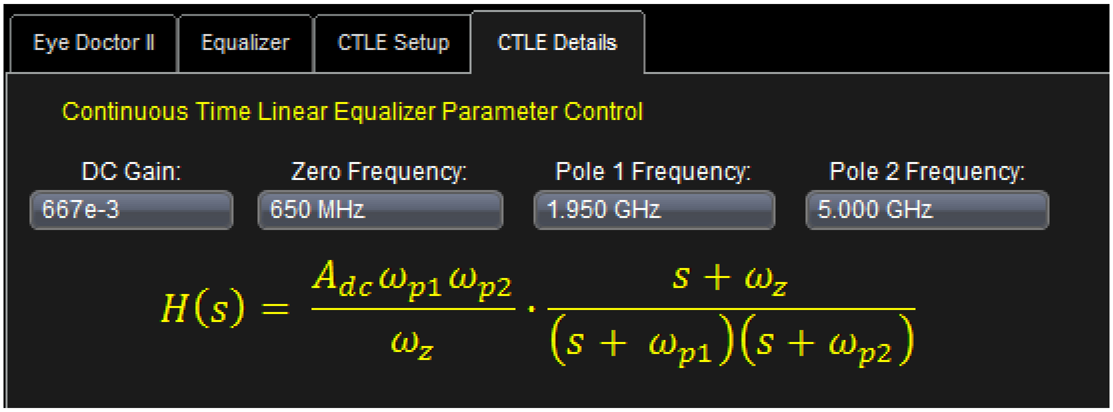
To use the CTLE:

1. Check the **Enable** checkbox in the **Continuous Time Linear Equalizer (CTLE)** section.
2. Ensure the **Nominal Bit Rate** is set correctly.
3. Specify the amount of **Boost** to use for equalization.

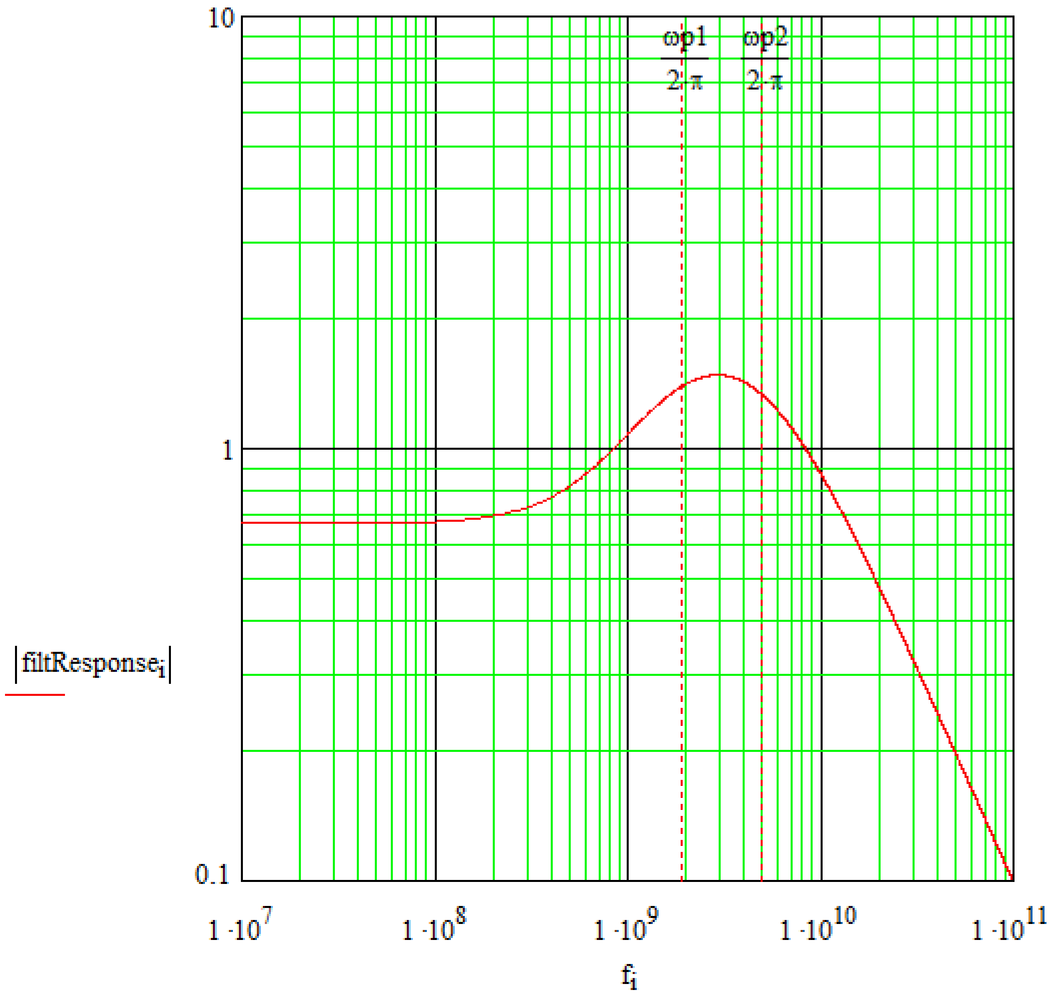
If you set the **Boost** on the **Equalizer** dialog, the DC Gain is unaffected and the higher frequency pole is placed at the bit rate and the lower frequency pole at a fraction of the bit rate as used in SuperSpeed USB. In order to set the requested boost, the Zero is also moved.



If you press **Custom** and then press the **Edit / View CTLE Setup...** you can set the DC Gain and the positions of the 2 Poles and 1 Zero (as follows).



This graph shows the response of the CTLE response for SuperSpeed USB. Eye Doctor II uses these setting as the default for the CTLE.



Feed Forward Equalization (FFE)

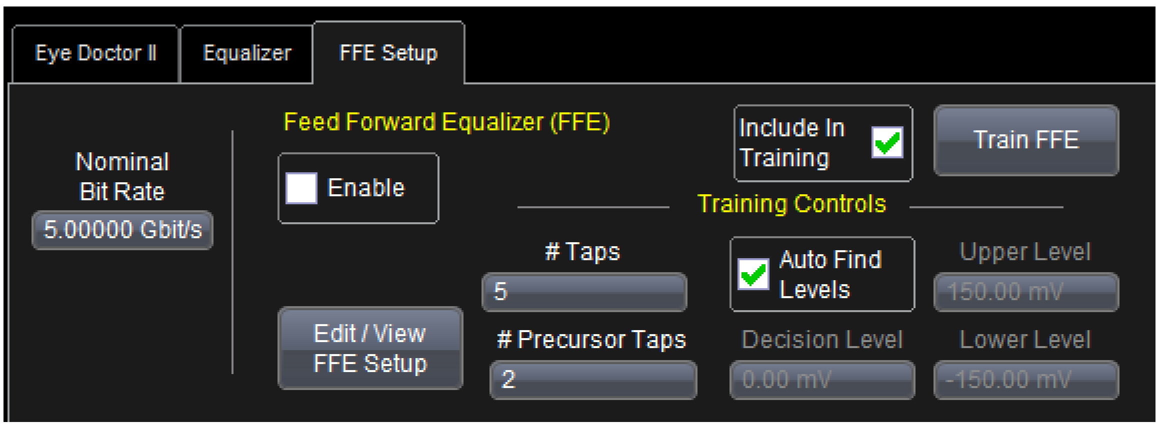
A Feed Forward Equalizer (FFE) is implemented as a tapped delay line filter, with 1 tap per unit interval. An FFE can compensate for Inter-Symbol Interference (ISI) due to preceding and following bits. For details on the algorithm for the FFE filter, see FFE Filter Algorithm.

When using SDAIII-CompleteLinQ, select the **Linear Eq** button on the SDAIII Framework dialog to configure FFE. Otherwise, select the **Equalizers** button on the Eye Doctor II dialog to configure FFE.

To configure FFE:

1. Check the **Enable** checkbox in the Feed Forward Equalizer (FFE) section.
2. Specify the **# Taps**.
3. Specify how many of those taps should be precursor taps in the **# Precursor Taps** dialog.

   Note: The number of precursor taps is often around half of the total number of taps.
4. Either check **Auto Find Levels** or manually enter the **Upper Level**, **Lower Level** and **Decision Level**.
5. Press the **Train FFE** button.



The **Include In Training** checkbox allows you to decide whether or not the Feed Forward Equalizer should be trained when the **Train FFE & DFE** button is pressed on the **Equalizer** dialog. The ability to exclude the FFE from training is particularly useful when specific tap values from the Feed Forward Equalizer have been entered by the user, which is discussed in the next paragraph.

While the best way to use the FFE is to train it, you can also enter the exact tap weights (if known) on the **FFE Details** dialog (as follows). The **Clear Taps** button sets all of the taps to 0 except the first tap which is set to 1. This is the pass-through state for the FFE.

FFE Filter Algorithm

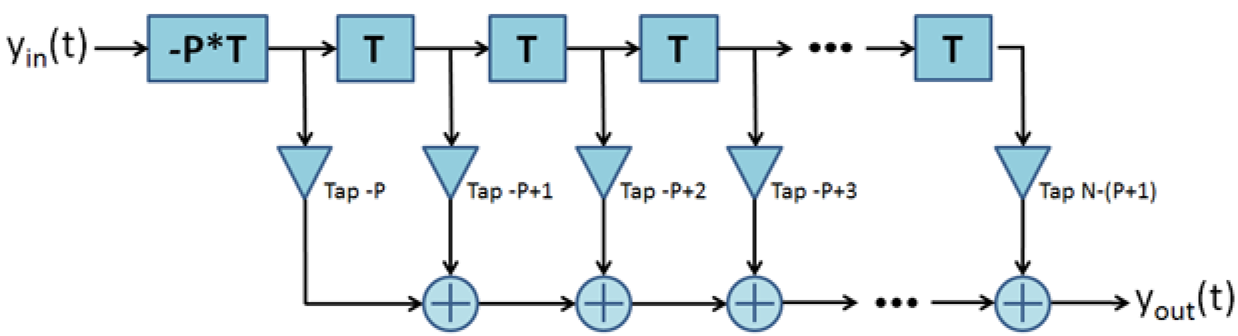
Feed-forward equalization is implemented as a tapped delay-line filter, with the 1 tap per unit interval. The filter is mathematically described as follows:

$$y_{out}(t) = \sum_{k=-P}^{k=N-(P+1)} C_k \cdot y_{in}(t - kT)$$

Where:

- ***y_{in}(t)*** is the initial waveform
- ***y_{out}(t)*** is the transformed waveform
- ***N*** is the number of taps
- ***P*** is the number of precursor taps
- ***k*** is the tap index, ranging from 0 to N
- ***C_k*** are the coefficients, or tap values
- ***T*** is the time delay for each of the delay elements (nominally 1/bitrate)

Depending on whether you have chosen to include precursor taps, the FFE filter can introduce a delay in your output waveform. The training algorithm seeks to find a solution for the tap values that retains the edge position, but note that when manually creating tap values, the output waveform can be advanced by adding precursor taps.



FFE block diagram.

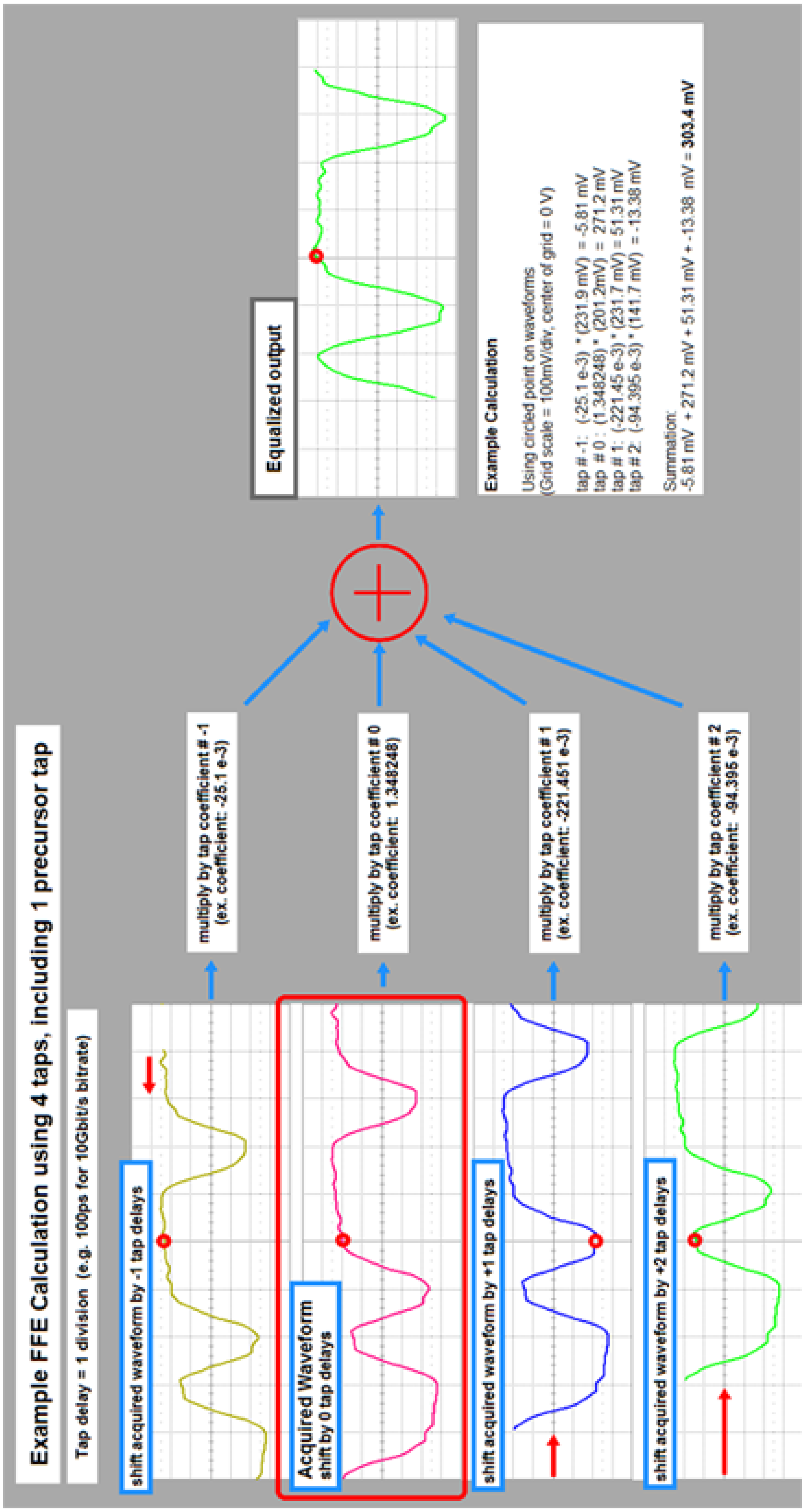
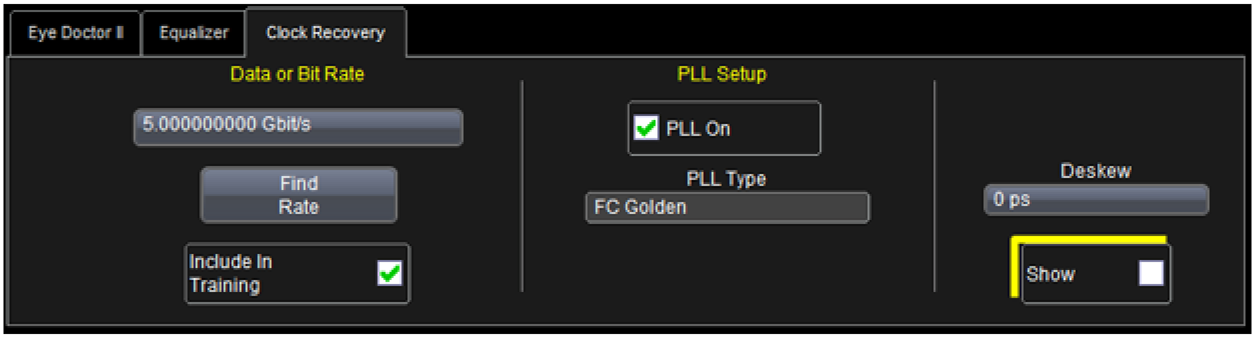


Illustration of an FFE filter with 4 taps, including 1 precursor tap

Clock Recovery

A recovered clock signal is required for Decision Feedback Equalization. For this reason, a clock recovery dialog is provided inside the equalizer section of Eye Doctor II.



Clock Recovery dialog for DFE Equalization when working in Eye Dr II or SDAII.

When entering the **Clock Recovery** dialog, the **Data or Bit Rate** is set to the nominal bit rate set in either the **Signal Input** dialog or **Equalizer** dialog. DFE requires a more precise bit rate than the **Emphasis** or **FFE** sections. Using the **Find Rate button** Eye Doctor II calculates the bit rate by analyzing the input waveform.

The **PLL Setup...** section of the **Clock Recovery** dialog contains the controls for setting the type and bandwidth of the digital PLL used to produce the clock (EyeDrClk) for DFE. The PLL has a low pass filter response; it tracks variations in the data rate that occur below a certain frequency, determined by the PLL bandwidth setting.

Example: A PLL bandwidth of 5 MHz allows the recovered clock to track frequency variations below approximately half this frequency, thereby removing their effect from jitter and eye pattern measurements. The software PLL used to produce the clock (EyeDrClk) for DFE allows you to choose among several types of PLL.

The selected PLL is applied to the data stream under test when the **PLL On** control is checked. The PLL recovers a reference clock from the data stream which is used by DFE and available as **EyeDrClk**.

The **Deskew** control can be adjusted manually or optimized when the DFE is trained. This time plus 1/2 of a unit interval is the point at which the decision is made for the decision feedback equalizer. It is the same control as the **Deskew** control in the **DFE** dialog.

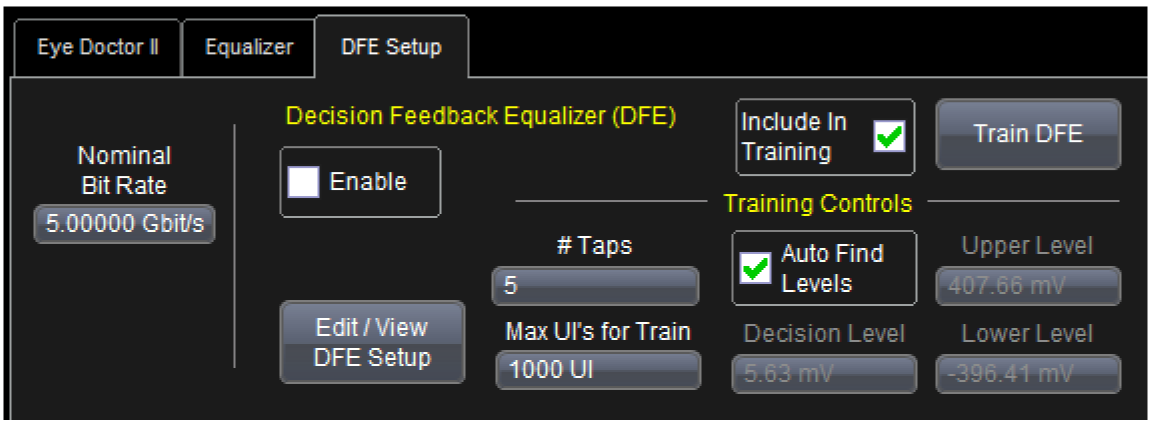
The **Show** check box allows the use view the **EyeDrClk** waveform on the grid display.

Decision Feedback Equalization (DFE)

To use the DFE:

1. Check the **Enable** checkbox in the **Decision Forward Equalizer (DFE)** section.
2. Specify the **# Taps**. The **# Taps** is set in order to span the # of UI the signal takes when settling to a final value after a transition.
3. Specify the maximum number of unit intervals to use for training in **Max UI's for Train**. We suggest leaving this value at 1000. Higher values make the training slower, and much lower values may reduce the training accuracy.
4. Either check **Auto Find Levels** or manually enter the **Upper Level**, **Lower Level** and **Decision Level**.
5. Press the **Train DFE** button.

   Note: The **Include In Training** checkbox allows the user to select whether or not the DFE is trained when the **Train FFE & DFE** button is pressed on the **Equalizer** dialog.



While the best way to use the DFE is to train it, you can also enter the exact taps weights (if known) on the **DFE Details** (as follows). The **Clear Taps** button sets all of the taps to 0. This is the pass-through state for the DFE.

The **Deskew** control can be adjusted manually by the user or optimized when the DFE is trained. This time plus 1/2 of a unit interval is the point at which the decision is made for the decision feedback equalizer. This is the same control as the **Deskew** control in the **Clock Recovery** dialog.

DFEs are prone to burst errors (meaning, once errors are created, they run for a long time). The DFE's propensity to create burst errors is based on the effectiveness of the DFE to aid in the accurate decoding of bits, which depends on its ability to accurately decode bits - a case of circular logic. Depending on the strength of DFE applied, a single bit error may lead to a long run of errors.

Erasure DFE is used for this problem and improves the situation. Erasure DFE effectively sets a band around the threshold value. When the signal falls inside the band at the time of slicing, it indicates an uncertainty in the bit decoding. The receiver, although obliged to decode the bit, can then decide not to apply this bit to the decision feedback, since the decoding was not certain enough. In operation, if erasure DFE is utilized and the signal is within the voltage delta about the threshold specified, a voltage value that is the average of the ideal +1 and -1 values is applied to the DFE delay taps, thus causing the bit to have no effect on decision feedback.

The **Erasure DFE** checkbox enables or disables erasure DFE. This feature excludes certain bits within the specified Erasure Delta from the decision feedback.

The **Erasure Delta** control allows you to set the indecision band around the DFE threshold.

Training Eye Doctor II Equalizers

Eye Doctor II provides receiver equalization with its Equalizer components. Equalizer capability is made up of software implementations of Continuous Time Linear Equalization (CTLE), Feed Forward Equalization (FFE), and Decision Feedback Equalization (DFE). Each can be enabled or disabled. The data signal can pass only through the DFE (which may be off, in which case the signal is unchanged); or through the CTLE and then the DFE; or through the CTLE and the FFE and the DFE.

DFE requires a clock, which Eye Doctor II recovers from the data after CTLE and/or FFE. As the following cases illustrate, if DFE is enabled you may have to enable one or both of CTLE and FFE to make sure Clock Recovery can accurately recover a clock.

Both the FFE and the DFE can automatically **Train** or find their own tap values. The **Train FFE & DFE** button on the main Equalizer dialog first trains the FFE, finds the bit rate (shown on the Clock Recovery dialog), and then trains the DFE including clock phase adjustment.

Three basic cases are covered in this section.

1. Case 1 - The first (and simplest) has the signal is slightly degraded, so it forms an open eye without equalization.
2. Case 2 - The second case has the eye is closed, so some equalization is required before Clock Recovery.
3. Case 3 - The third case has the eye severely degraded (beyond just being closed). In this severe case, DFE automatic training may not work unless a bit of help from the CTLE is applied and the input from the DFE is taken from CTLE output.

Generally, the following pictures show a zoom of the input to Eye Doctor II (EDrIn), in the upper left; a zoom of the output of Eye Doctor II's equalizer (EDrEqu), in the lower left; and an eye diagram formed by SDA II using **EDrOut** as the signal source, and **EDrClock** as the explicit clock source.

This setup has Eye Doctor II doing the clock recovery, it has the PLL setup; SDA II does not have a PLL enabled.

Eye Doctor II also needs to know the signal's bit rate to a reasonably high accuracy. For this reason there is a **Find Frequency** button on its **Clock Recovery** dialog; if the signal is appropriately equalized for Clock Recovery, it does work; and SDA II needs to know the signal's bit rate (at least nominally) to form an appropriate Eye diagram. SDA II has a **Find Frequency** button on the **Clock Recovery** tab of its dialog, if the data signal is appropriately equalized, it does work.

PLEASE NOTE THE FOLLOWING:

Each of the training cases uses Eye Doctor II in simple ways.

- **Auto Find Levels** was left checked for the DFE.
- We only used tap values for DFE that were the result of our Training algorithm. We could have pressed **Edit/View DFE Setup**, and entered our own tap values, if desired (and knew the values).
- We used the **CTLE** in **Auto** mode. In **Auto** mode, the only parameter shown on the CTLE dialog is **Boost**.

  The oscilloscope automatically places the zero and two poles (the second being the parasitic pole, which we place at the bit rate, as previously explained).

  If we pressed **Edit/View CTLE Setup...** while in **Auto** mode, the pole and zero frequencies are shown with no change. In fact, all that could be changed is the DC gain, which is just a scale factor.

  We could have set CTLE mode to **Custom**, and set the pole and zero frequencies manually. This is a great way to experiment with the CTLE.
- Under **Emphasis**, we used **Auto Add**. **Auto Add** uses two taps. For 6dB, the two taps are approximately -0.25 and 0.75; so that the first UI in a state is 1 and all subsequent bits in the same state are 0.5.

  We could have selected Custom, and entered up to eight tap values. PCI Express Gen 3 will use 3 tap TX equalization, which can be easily set up in Custom.

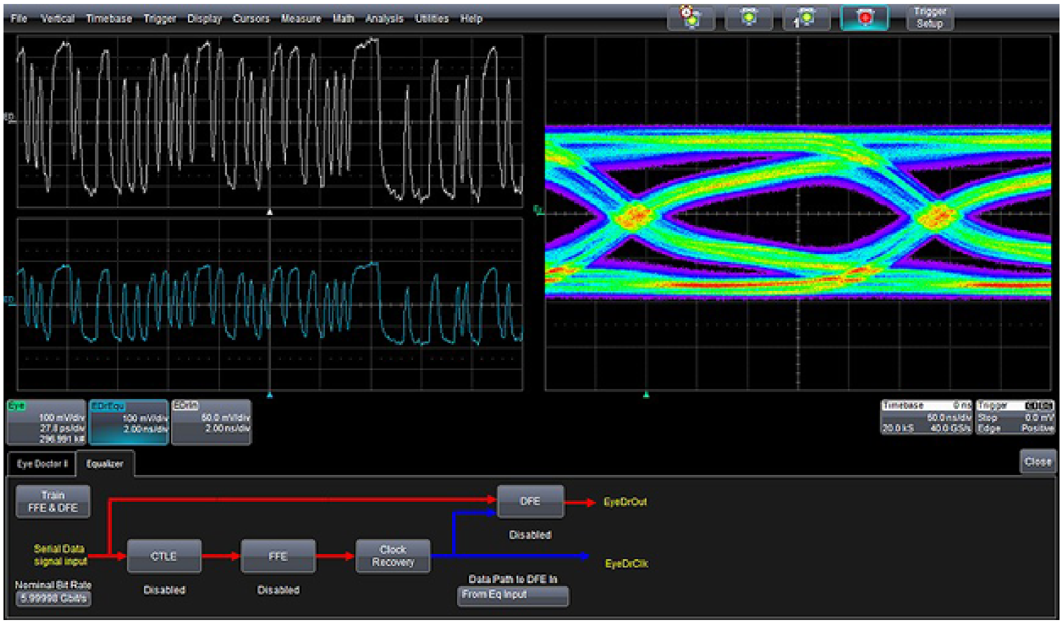
Case 1

SLIGHTLY DEGRADED SIGNAL (6GB/S THROUGH 10” OF FR4)

With no equalization applied, this signal has an open eye; this shows that Eye Doctor II's clock recovery worked on this signal without any equalization. We touched **Find Frequency** and otherwise accepted the default state (PLL on, etc) on Eye Doctor II's clock recovery dialog.

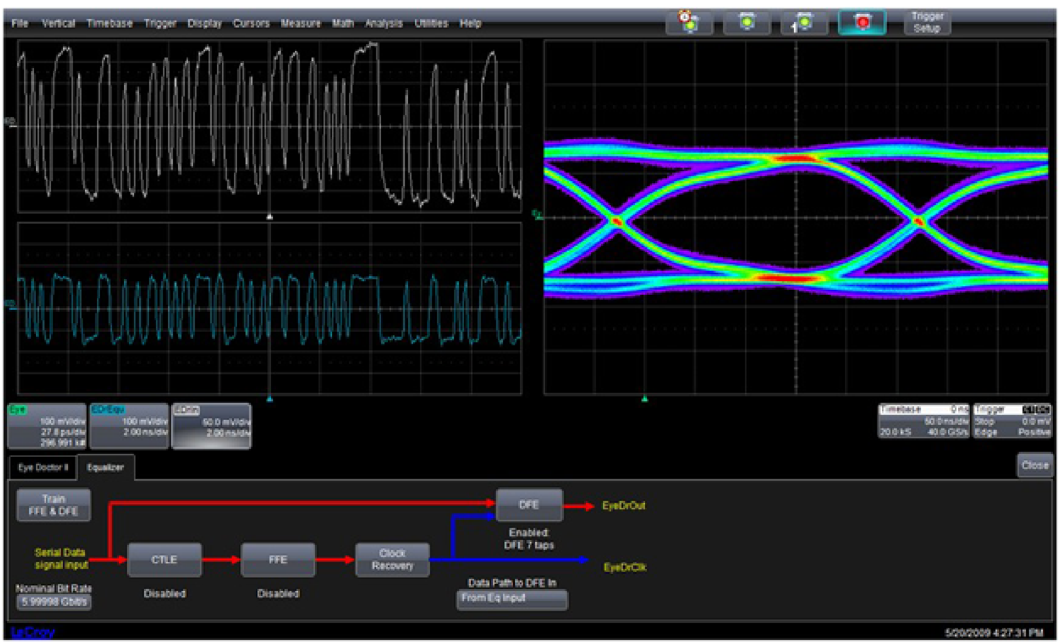
The following screen-shot shows both EDrIn and EDrEqu traces with the same signal, because no equalization is applied. The output of EDrEqu is reframed, meaning it is displayed on a different scale.

Since Eye Doctor II is not actually doing anything, we could have fed this signal directly into SDA II and had SDA II recover clock from the data, to make the Eye show as in the following screen-shot.



Case 1, without equalization.

Equalize the eye by enabling DFE and training it:



Case 1, with DFE equalization (at 100mV/div vertically).

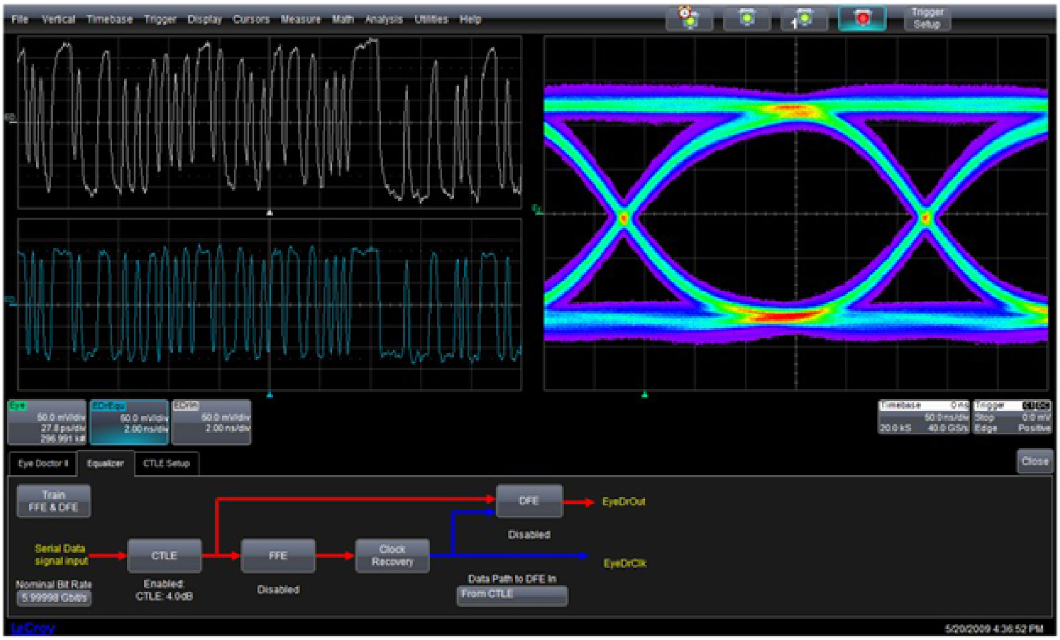
DFE has the property of not actually boosting high frequency content and instead shifting its threshold (to display an eye we actually shift each UI of the waveform relative to a fixed threshold), so DFE does not emphasize high frequency noise.

The previous screen-shot shows the most open position of the eye is located at the horizontal center of the screen, that is a result of DFE training. The clock phase is adjusted in this example by -11ps to achieve the equalization.

If this is not what you want to see, use SDA II's Deskew control on its Ref. Clock Input dialog, to move the eye horizontally. Do not change Eye Doctor II's Deskew (also shown on the DFE setup details page as Clock Adjust) because the trained DFE taps are optimal at that Deskew setting.

This signal can be equalized with only the CTLE or the FFE. The following example shows the DFE disabled, the CTLE is enabled, and the Data Path set to the CTLE Output so its effect can be observed.

Equalization by the CTLE:



Case 1, with CTLE equalization (at 50mV/div vertically).

The CTLE auto setup applies a DC gain of .667 (the default), and boosts high frequencies with a peak at about half the bit rate, and then rolls off due to a second pole at the bit rate. This suppresses boosting high frequency noise where there is little energy from the data signal. The CTLE does not have automatic Training, it is very easy to adjust the Boost in auto mode until the eye looks good.

There are two other examples not pictured here; however, the following descriptions clearly explain the logic:

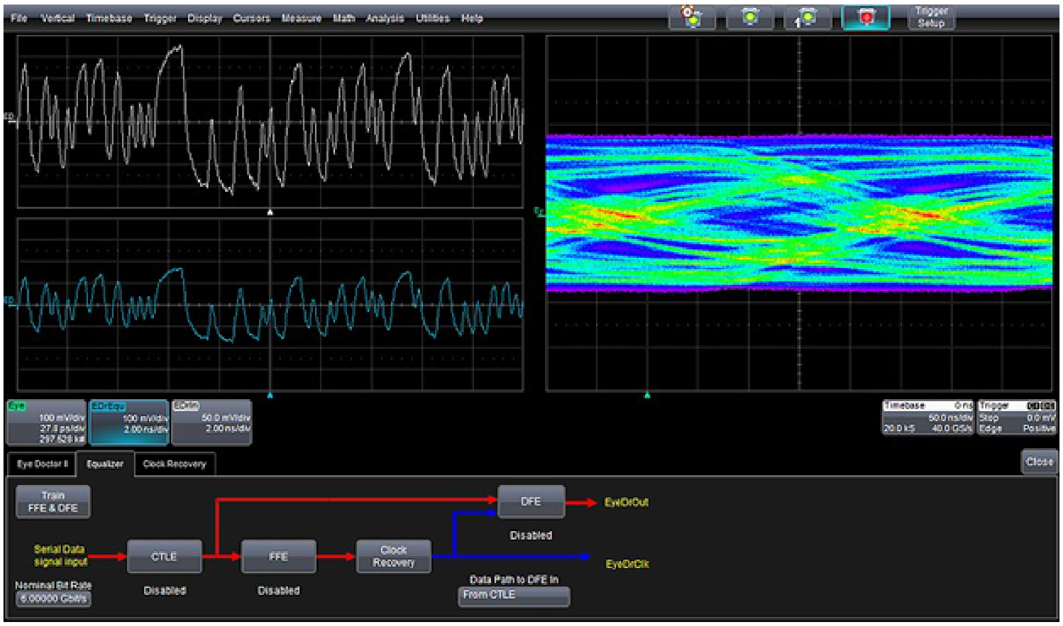
- Set Data Path to **DFE In** to **From FFE**, enabling only the FFE, and Training the FFE to equalize this signal using only the FFE.
- Set Data Path to **DFE In** to **From CTLE** (or **From FFE**), enabling only the CTLE and the DFE, setting the CTLE to 2dB boost (its minimum) and Training the DFE so it supplies the rest of the equalization.

Case 2

A Closed Eye (6Gb/s Through 30” Of FR4)

With no equalization applied, this signal clearly shows a closed eye. However, we cannot yet say the signal truly does have a closed eye because we cannot correctly recover the clock from the signal. Therefore, we must feed a reasonable signal into Clock Recovery.

The following screen-shot shows both EDrIn and EDrEqu traces having the same signal, because no equalization was applied. The output of EDrEqu is reframed, meaning it is displayed on a different scale.

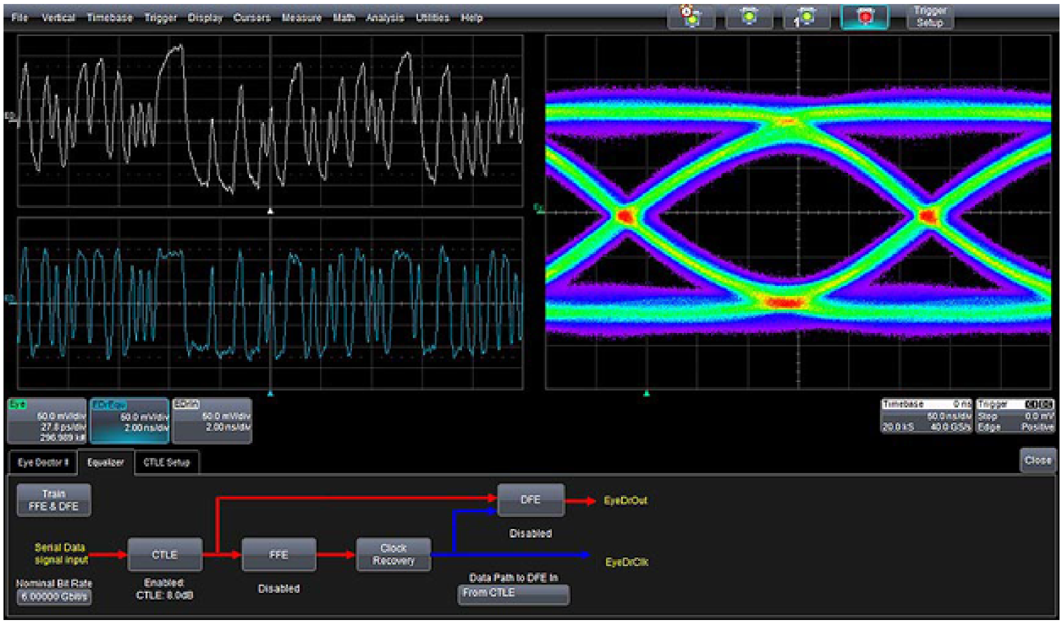


Case 2, without equalization

If we go to Eye Doctor II's Clock Recovery dialog and touch **Find Frequency**, a clearly incorrect result (in this case, 9.04878Gb/s instead of nominally 6Gb/s) is produced. If we **Show** the recovered clock (**Show** on the Clock Recovery dialog) we can see it is missing some edges. The issue is, of course, that we need to feed a reasonable signal (one forming an open eye) into the Clock Recovery. As follows, we use the CTLE to equalize the signal, change the data path so CTLE is only used on the path to Clock Recovery, and then only use DFE to equalize the signal for EyeDrOut.

Note: We have entered the nominal bit rate, 6Gb/s, manually.

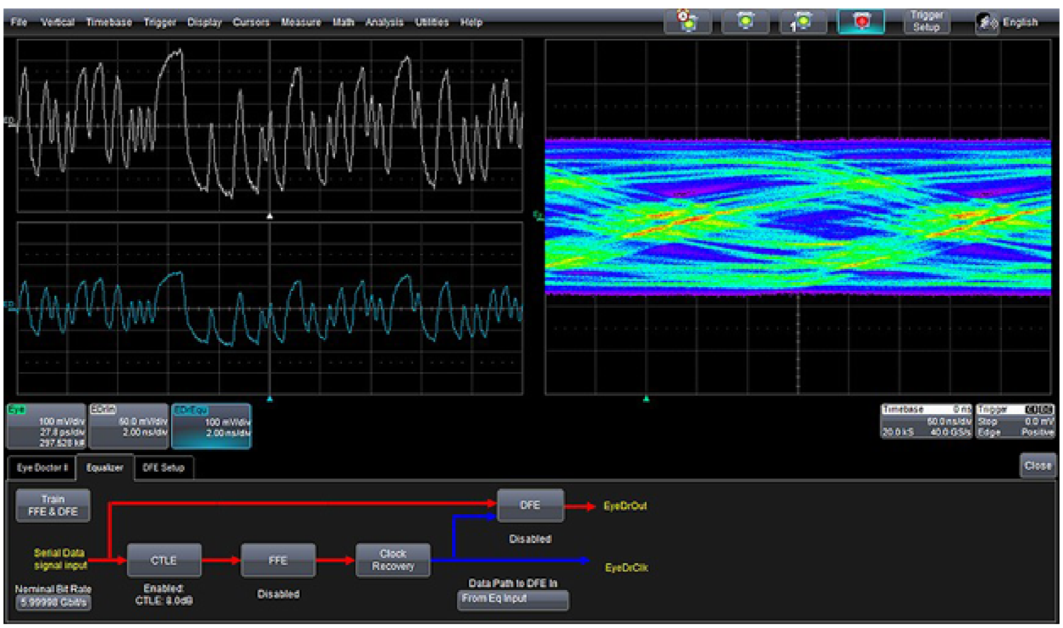
First, we can set up the CTLE by adjusting the Boost until the signal appears equalized. It just takes a few seconds to determine that 8dB looks good:



Case 2, with CTLE equalization which will be used for clock recovery

Now touch **Clock Recovery** and on the dialog shown, touch **Find Frequency**.

Next, change **Data Path to DFE In** to **From Input**, so the CTLE is applied only on the path to Clock Recovery. The Eye diagram once again looks closed, and since the Clock Recovery is now operating correctly, we can be sure it is closed. Here is the correctly clocked closed Eye:



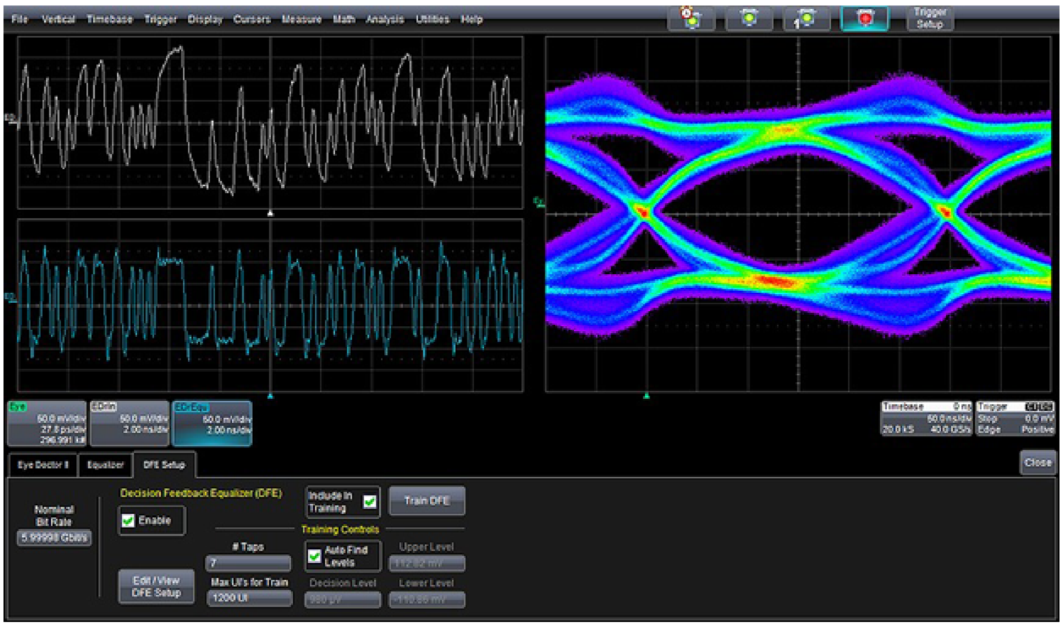
Case 2, correctly clocked closed eye. No equalization applied on the data path.

Finally, touch DFE showing the DFE dialog to **Enable**, and then **Train** the DFE.

Note: DFE taps is set to 7, the default is 3. The result looks significantly better when using 7 taps, because on long runs of the same state, the input waveform continues to drift to larger magnitudes for more than 3 UI.

In the following screen-shot we show Eye Doctor II's DFE Setup dialog. On the Equalizer dialog, the only difference is the caption under the **DFE** button showing **Enabled: DFE 7 taps**. So, in the following screen-shot, equalization is only being done by DFE.

The following screen-shot shows Eye Doctor II applying a reasonable transition time to the shift at the UI boundaries. The shift reflects the DFE threshold shift, by moving each UI of the waveform relative to the fixed center level. The transition time is automatically determined as a fraction of a UI, adjusted to be reasonable at the greater of the oscilloscope's maximum bandwidth or 16GHz.



Case 2, with DFE equalization.

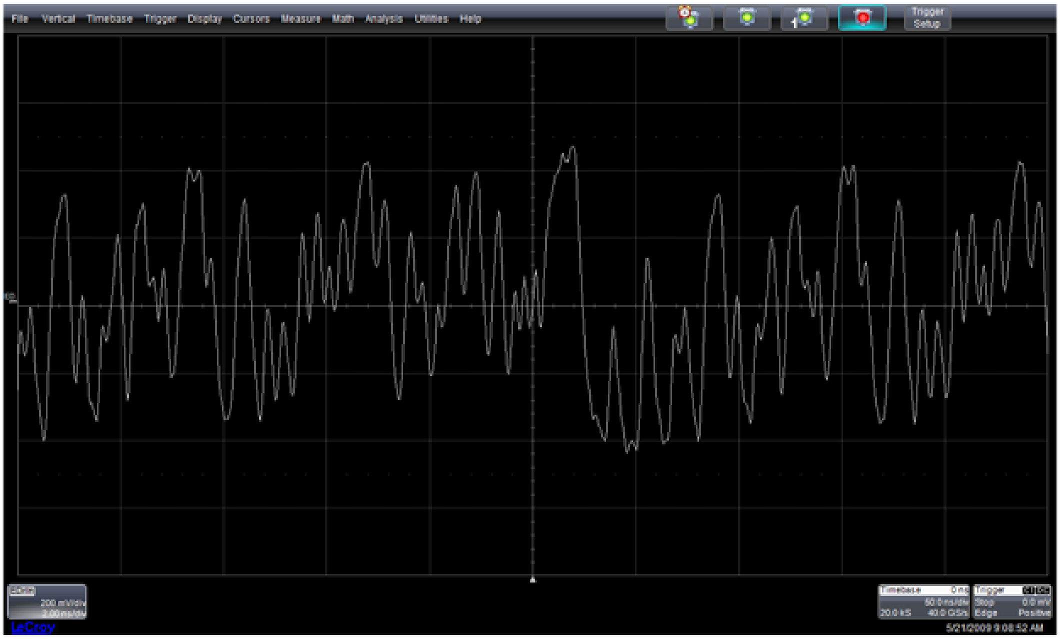
In this case, if the FFE is used instead of CTLE on the clock recovery path, the result is the same.

See for yourself and set **Data Path** to **From FFE** to see the result of training the FFE, and then change the **Data Path** back to **From Input** so only the DFE does the equalization for the data out. The result should look exactly like the previous screen-shot.

Case 3

A SEVERELY DEGRADED SIGNAL (10GB/S THROUGH 30" OF FR4)

First, we'll look at the input waveform. It's clear that this waveform is much worse than just a closed eye; there are wiggles in it that don't even get close to the center line. Look, for example, at about 3.5 divisions from the left.



Case 3 input signal.

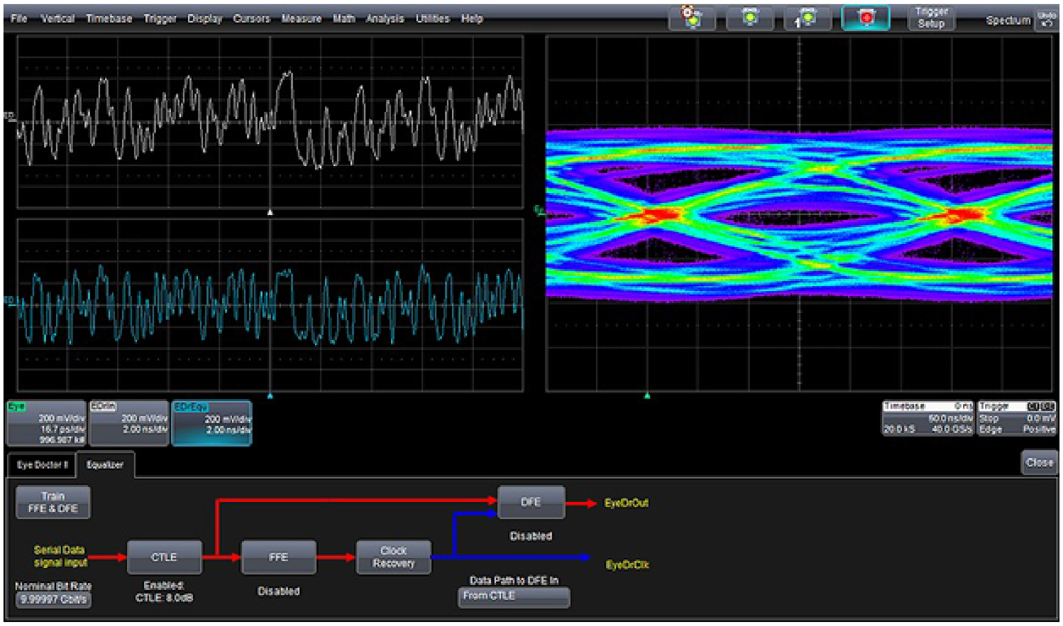
First, we'll try what we did for Case 2. Along the way we run into one additional problem and resolve it.

As it says in case 2: *setup the CTLE to equalize the signal, change the data path so that CTLE is only used on the path to Clock Recovery, and then only use DFE to equalize the signal for EyeDrOut.*

Note: We have entered the nominal bit rate, in this case 10Gb/s, manually.

It appears that 7dB of boost is enough, the first clue we have is that **Find Frequency** on the **Clock Recovery** dialog produces a reasonable bit rate for 7dB and up to beyond 10dB of boost. At much lower boost or much higher boost (say, 15dB) it does not. The following screen-shot shows the SDA II Eye turned on again and, after pressing **Find Rate** on the **SDA II Clock Recovery** dialog (so SDA II also knows about the new signal rate) we see the eye produced by 8dB of boost.

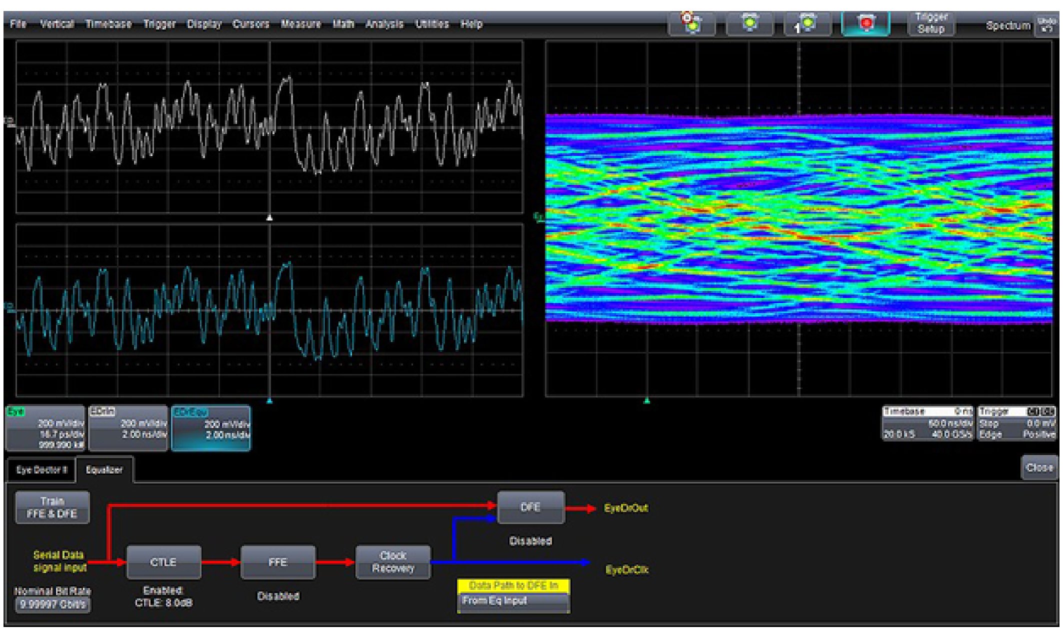
It's not ideal, but the eye is open.



Case 3, with CTLE equalization.

An interesting point about this waveform is we would expect higher CTLE boost to improve the eye, but it doesn't do everything we hoped. That probably means there is some degradation besides the linear slope of magnitude reduction with increasing frequency.

Next, following the procedure from case 2, we should *change the data path so CTLE is only used on the path to Clock Recovery, and then only use DFE to equalize the signal for EyeDrOut.* Once again, we now have the correctly clocked closed eye. However, unlike in case 2, the following screen-shot shows the outline of the eye as not being discernable. The difference is clear when comparing the closed eye picture in case 2 to the following screen-shot. This signal is much more degraded.



Case 3, correctly clocked closed eye. No equalization applied on the data path.

Finally the last step in the case 2 procedure is to **Train** the DFE processor. When we do this it clearly does not succeed - the result, shown as EDrEqu and in the Eye diagram, does not change.

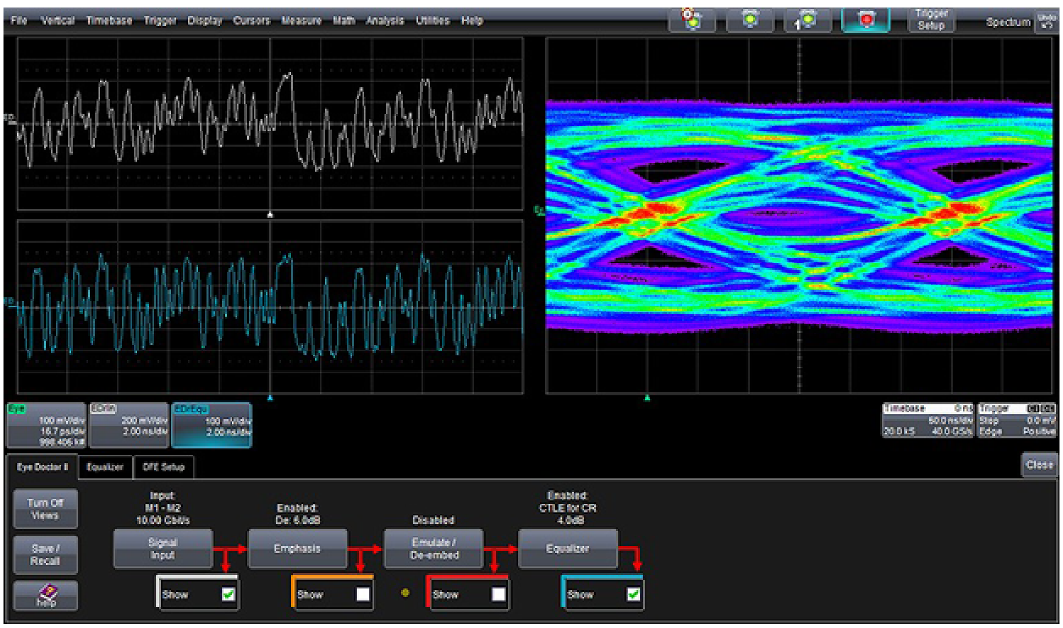
If we go to the DFE menu, and touch **Edit/View DFE Setup**, we see all the taps are 0.0 (the default), which does nothing.

The training fails because too many bits are decoded incorrectly in the first pass and the beginning of training always starts with all taps set to 0.

The problem is the received signal is so degraded that receiver equalization is having trouble dealing with it. The usual way to address this problem is to split responsibility for equalization between the transmitter and the receiver.

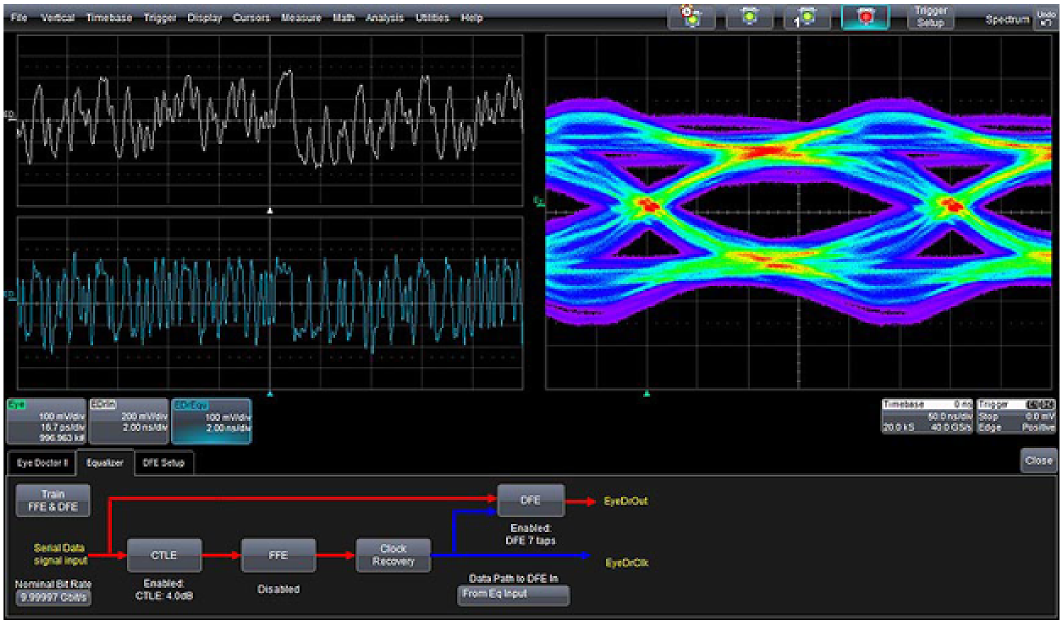
For this procedure, we add 6dB of de-emphasis to this waveform, which simulates TX equalization. We also reduce the CTLE boost on the Clock Recovery path to 4dB, to compensate (we don't want to over-equalize on the clock recovery path).

With DFE turned off, the eye diagram shows only the effect of 6dB of de-emphasis:



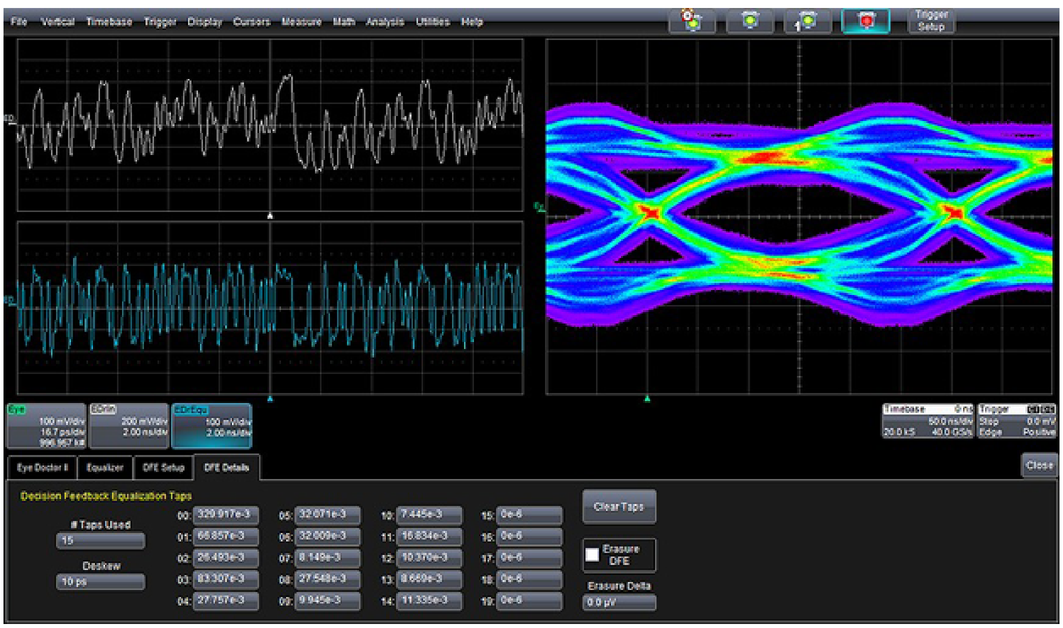
Case 3, with added 6dB de-emphasis (simulating TX equalization).

While much improved by 6dB of de-emphasis, the eye is just barely open. We can be sure the DFE can Train on this. Now, we enable DFE and Train it again, producing the following result:



Case 3, with 6dB de-emphasis and 7 tap DFE.

Let's see if more DFE taps help. Note the somewhat tighter distributions at the crossings.



Case 3, with 6dB de-emphasis and 15 tap DFE.

The previous screen-shot shows the DFE Details dialog (accessed by touching **Edit/View DFE Setup**). The tap values are shown and can be changed. In addition, the number of taps in use are shown along with **Clock Adjust** (also shown on **Eye Doctor II's Clock Recovery** dialog as **Deskew**). This screen-shot shows the Training decided that 10ps of deskew was needed. Since the Eye diagram horizontal scale is 16.7ps/div we clearly see 10ps if it was not deskewed. The need for this deskew is almost certainly due to the group delay of the CTLE, only present on the clock recovery path, so its group delay caused a skew between the data and the recovered clock.

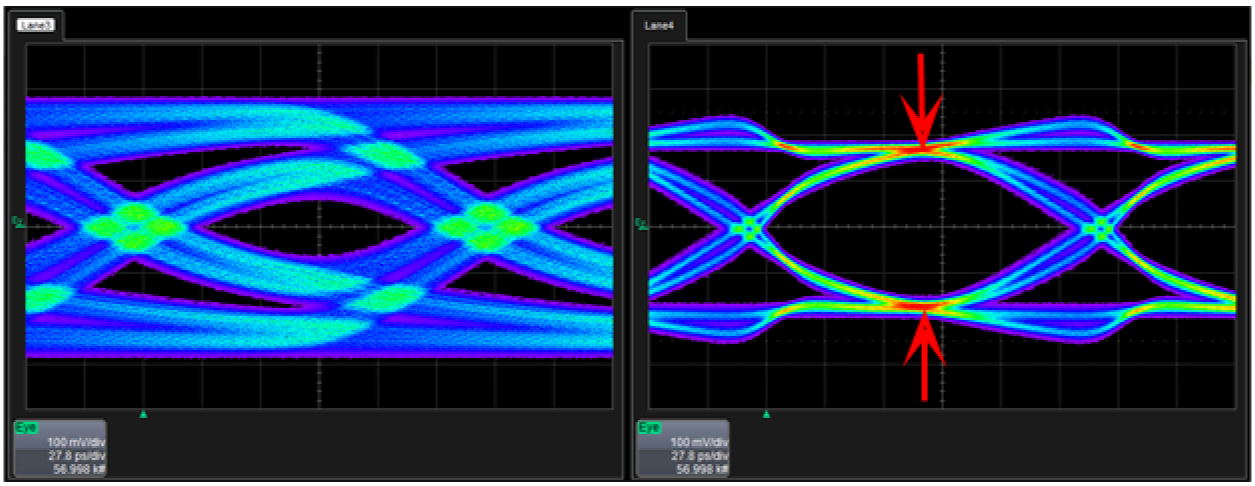
As noted previously, **this value should not be changed**.

The DFE taps are optimized together with this value; changing it changes the clock skew into DFE and degrades the result. Instead, to shift the eye diagram horizontally, use the **Deskew** control on the **SDA II Clock Recovery** dialog.

Conclusion for case 3: beyond a certain level of degradation, it becomes very difficult to compensate with only receive equalization. Eye Doctor II can compensate by using both a CTLE and DFE, or both a CTLE and FFE, but real silicon implementations are unlikely to do either. In such cases it is necessary to use TX equalization (meaning de-emphasis or pre-emphasis). Eye Doctor II can simulate TX equalization (by adding it to a waveform that doesn't have it) using controls under the Emphasis tab.

FFE and DFE Training Algorithm

EyeDoctor II includes a training algorithm that determines the optimal tap values to use in the FFE and DFE filters. The method used is the Levenberg–Marquardt algorithm. The variable that is minimized is the range of the high and low voltages near the center of the eye, as shown in the figure below. Minimizing this quantity maximizes the eye opening.



You can configure certain arguments that are part of the training algorithm. Both FFE and DFE include the following controls, which can be set independently:

Number of Taps: The training algorithm will create a filter with the number of taps selected.

Auto find levels: Checking this will have the algorithm find the upper and lower levels of the equalized eye as well as the decision level to use. When selected, the voltage levels of the minimized locations of high and low states will tend to match the outmost voltage levels of the unequalized eye.

Upper Level: Voltage to place the upper level of the equalized eye.

Lower Level: Voltage to place the lower level of the equalized eye.

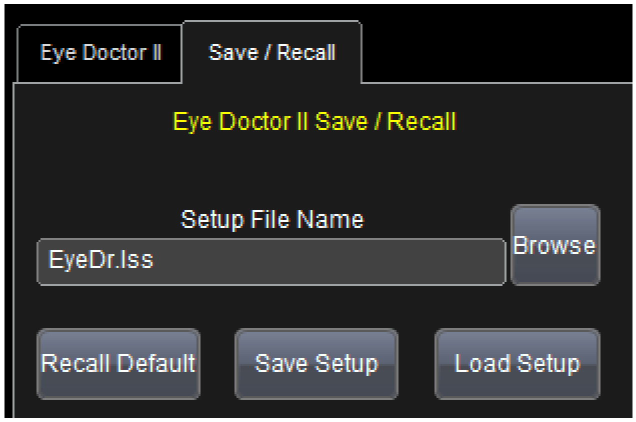
Precursor taps(FFE only): The training algorithm create a filter that includes the specified number of precursor taps.

Max UIs for train (DFE only): Maximum number of unit intervals to use for training. We suggest leaving this value at 1000. Higher values make the training slower, and much lower values may reduce the training accuracy.

Save / Recall

Open the **Save/Recall** tab of the **Eye Doctor II** dialog to save your current Eye Doctor II setup or recall a previously saved Eye Doctor II setup.

This capability is available when using EyeDoctor II via the Analysis menu only.



Saving Current Eye Doctor II Setup

1. Touch **Setup File Name** to specify the desired name for your saved setup file.

   OR

   Click **Browse** button to select an existing file to overwrite or to specify a new location for the setup file.

2. Touch the **Save Setup...** button.

Recalling Saved Eye Doctor II Setup

1. Touch the **Browse** button to select the setup file to recall.
2. Touch the **Load Setup...** button.

TELEDYNE LECROY
Everywhereyoulook™

700 Chestnut Ridge Road
Chestnut Ridge, NY 10977
USA

teledynelecroy.com