Infiniium Serial Data Equalization

KEYSIGHT TECHNOLOGIES

User's Guide

Notices

Revision

Version 11.30.00203

Edition

July 2022

Available in electronic format only

Trademarks

Infineon is a registered trademark of Infineon Technologies AG. The Infiniium mark is used with the kind approval of Infineon Technologies, AG.

MIPI® service marks and logo marks are owned by MIPI Alliance, Inc. and any use of such marks by Keysight Technologies is under license. Other service marks and trade names are those of their respective owners.

PCI-SIG®, PCIe®, and the PCI Express® are US registered trademarks and/or service marks of PCI-SIG.

Warranty

Technology License

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Safety Notices

> **CAUTION**
>
> A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

> **WARNING**
>
> **A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.**

Contents

1 Equalization

NOTE   You must have the D9020ASIA Advanced Signal Integrity Bundle installed on your oscilloscope to have access to the features described below.

There are two ways to set up the Infiniium Serial Data Equalization application:

- The Equalization Wizard is the recommended setup method because it guides you through the process in a simplified, step-by-step fashion. See "Using the Equalization Wizard" on page 34.

- If you are comfortable with the Equalization dialog box controls and are familiar with equalization, you can use the manual setup method. See "Manually Setting Up Equalization" on page 51.

See Also   If you prefer to view a PDF version of this help, see *Equalization User's Guide*.

KEYSIGHT
TECHNOLOGIES

Equalization Background / Theory

To understand how equalization works, you first need to understand the problem it corrects.

The Problem

A serial signal consists of a transmitter sending a signal over a transmission channel (examples: backplane, cable) to a receiver. As the signal rate increases, the channel the signal travels through distorts the signal at the receiver. This distortion can result in a partially or completely closed eye diagram where the clock or data cannot be extracted by the receiver. To recover a clock or data from the eye diagram, the eye must be re-opened. This is where equalization can help.

As shown in the following diagram, you can see that an open, clean eye leaves the transmitter and is sent through the channel. As it passes through the channel, random noise, crosstalk, and inter-symbol interference (ISI) distort the signal. These factors can cause the eye to close. In the diagram s(t) is the transmitted signal, r(t) is the received signal, and e(t) is the equalized signal.

You can use equalization to re-open the eye by correcting for the ISI because ISI is deterministic (that is, the ISI can be determined prior to transmission).
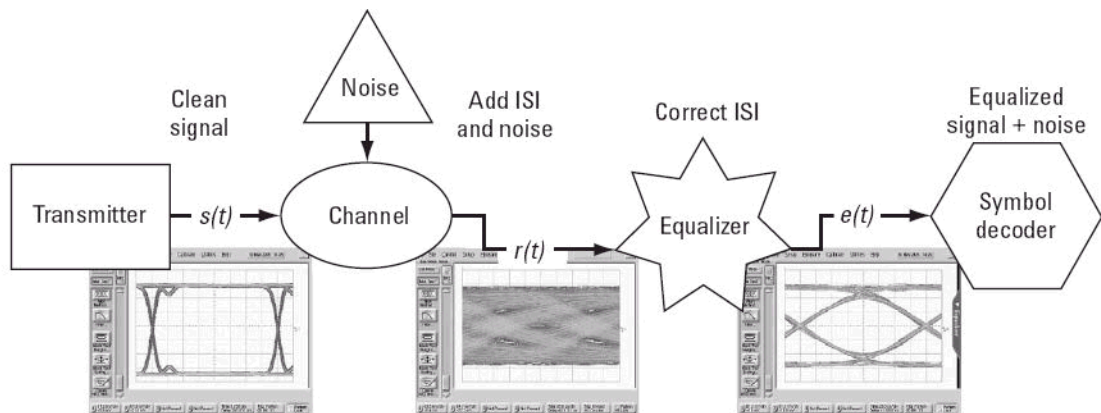
The random noise that was introduced cannot be corrected for by equalization because it is non-deterministic. Therefore, you can re-open the eye, but maybe not completely due to the inability to correct for this random noise.

To see why ISI (and the closing of the eye) causes problems when analyzing serial data, look at the following two screen shots showing a data signal before and after traversing a backplane.

Transmitted signal, *s(t)*

Received signal, *r(t)*
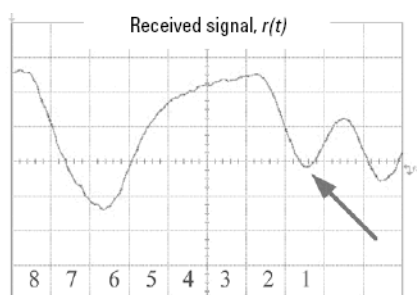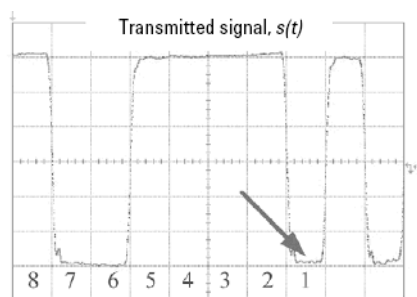
Looking at the five bits labeled **5**, **4**, **3**, **2**, and **1**, you see the logical pattern is 11110 (decision threshold of V = 0). Also, notice even though the bit indicated by the arrow (bit **1**) is a logical zero when it is transmitted, the logic level of the bit as seen by the receiver is right around the decision threshold. It may, therefore, be inaccurately counted as a logical one. This occurred because the frequency response of the backplane filtered out the high-frequency content of the waveform.

It is vital that your receiver measures the same pattern the transmitter sends. Equalization is a technique used to ensure this occurs.

Inter-Symbol Interference (ISI)

Inter-symbol interference (ISI) is caused by the non-flat frequency response of the channel the signal travels through (high frequencies exhibit more loss than low frequencies) and results in the distortion of pulse shapes in your signal.

Look at the following simplified frequency response plot.
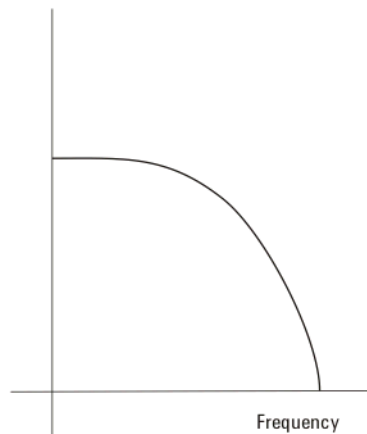
You can see that at low frequencies, the frequency response is flat (like you want), but as the frequency increases, there is a drop in the amplitude. If your signal contains frequency components in this region, it will be distorted.

For example, imagine you are trying to send a serial data sequence (11010 . . .) from a transmitter, through a channel, and to a receiver. Theoretically, you could send each one of these bits as square pulses from the transmitter. In real life, however, you are limited by bandwidth and, therefore, they are somewhat rounded.

As this data sequence is sent through the channel, the medium it is traveling through distorts the rounded pulses by smearing and elongating them (the degree of this distortion depends on the frequency components). This can cause each bit to have some of its energy overlap with neighboring bits, thus impacting the voltage level of those bits. When the signal exits the channel and enters the receiver, the receiver may measure different bit voltage levels as a result.

If you were to plot an eye of a serial signal with high enough frequency components that ISI is a significant problem, you would see that the low-frequency trajectories of the waveform would still be accurate, but the high-frequency components would be attenuated, resulting in either the complete or partial closing of the eye.

ISI is caused by a combination of:

·   The geometry of the circuit (that is, the design of the trace)

·   The medium from which it is composed (that is, the conductor or dielectric)
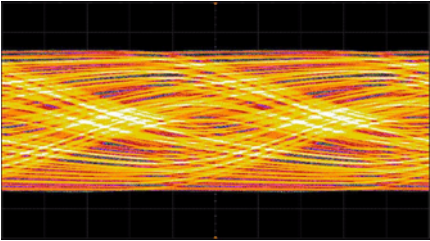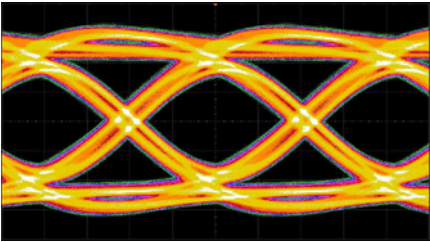
Each of these factors can be determined prior to signal transmission. Hence, ISI is deterministic, and due to this, you can correct for it with equalization techniques.

The Solution: Equalization

The main purpose of equalization is to correct for the problems caused by the transmission channel. Equalization techniques provide a way to discern the original signal (the signal coming out of the transmitter) given a distorted signal at the receiver.

In other words, equalization corrects for the high-frequency component voltage levels and, in the process, corrects the trajectories of these components in the corresponding eye diagram (that is, opens up the eye).

Unequalized 6 Gb/s Eye Diagram Notice that the eye is completely closed.	
Equalized 6 Gb/s Eye Diagram Notice that the eye is now open after equalization was applied.	

Equalization Methods

The Infiniium oscilloscope has three equalization methods available:

- "Feed-Forward Equalization (FFE)" on page 12
- "Continuous Time Linear Equalization (CTLE)" on page 18
- "Decision Feedback Equalization (DFE)" on page 19

Multiple "Lanes" of Equalization

Before the 6.40 version of Infiniium oscilloscope software, you could perform equalization on a single input source. Now, you can perform equalization on up to four sources at once. Each of the four "lanes" of equalization has its own settings, distinct from the other lanes, allowing for independent equalization on different signals, concurrent equalization on the same signal, or any combination thereof.

Equalized Clock Recovery   Equalized clock recovery now has up to four different sources, one for each lane. If a lane is equalizing in-place (see ""Equalize in place" vs. "Display as function"" on page 11), it will not be available as an equalized clock recovery source. Presently, the only way to use an in-place lane for clock recovery is to turn it on and use its source for explicit clock recovery.

Chaining Lanes   Equalization lanes can be chained from one to the next when they are displayed as a function (see ""Equalize in place" vs. "Display as function"" on page 11). That is, the equalized signal from Lane<X> can be the source for Lane<X+1>.

Copying Equalization Settings from One Lane to Another   The Equalization dialog box lets you copy one equalization lane's settings to another lane. This includes all CTLE, FFE, and DFE settings, with one special case exception: If Lane<X> is using Lane<X-1> as its source, and Lane X's settings are copied to Lane Y, Lane<Y> will attempt to use Lane<Y-1> as its source. The only case in which this does not work is if Y is 1, in which case Eq1's source is unchanged.

Legacy Setups   In legacy setup files created before version 6.40, linear equalization (CTLE/FFE) could have a different source than DFE; therefore, these settings are now recalled into separate lanes:

- Linear equalization (CTLE/FFE) settings recall into Lane 1.
- DFE settings recall into Lane 2.

"Equalize in place" vs. "Display as function"

You can now choose to either **Equalize in place** (modifying the waveform itself) or **Display as function** (creating a separate equalized waveform, which is what was done in the Infiniium oscilloscope software versions before 6.40).

For linear equalization (CTLE and FFE), "in-place" means the equalization runs completely in hardware, greatly improving speed.

For DFE, "in-place" means the pre-6.40 version of DFE is performed (that is, the DFE that modifies the display of the real-time eye only). The "as-a-function" option now lets you display a waveform with DFE applied (see "The DFE "Display as function" Waveform" on page 26).

Rules for In-Place Equalization

The following rules determine whether in-place equalization is legal, and can be used to explain interactions involving the "in-place" vs. "as-a-function" selection:

1 Of all running lanes using source <S>, only one lane can equalize <S> in-place. Any of the lanes can be selected as the one that equalizes in-place.

2 In-place CTLE and FFE can be applied to analog channels only.

3 In-place DFE can be applied to all sources with a real-time eye displayed.

Feed-Forward Equalization (FFE)

Feed-Forward Equalization (FFE) is an equalization technique so named because, unlike "Decision Feedback Equalization (DFE)" on page 19, FFE corrects the received waveform with information about the waveform itself and not information about the logical decisions made on the waveform. FFE basically acts like a FIR (finite impulse response) filter and uses the voltage levels of the received waveform associated with previous and current bits to correct the voltage level of the current bit.

One key thing to remember when working with FFE is that the equalization is performed on the actual waveform. At no point in the FFE algorithm are logical decisions made (is this bit a 1 or a 0?). Instead, FFE is concerned only with correcting voltage levels in the waveform. This is in direct contrast to Decision Feedback Equalization (DFE).

FFE is described in four different ways in this section. Each description contains the same basic content, but lets you choose the viewpoint that best suits your understanding.

- "Mathematical Description of FFE" on page 12
- "Description of FFE Acting on a Waveform" on page 13
- "FFE Decision Diagram" on page 15
- "Understanding FFE Through Frequency Response Plots" on page 15

In addition to FFE, the Infiniium oscilloscope can also perform Decision Feedback Equalization (DFE) or both DFE and FFE in combination.

See Also
- "Continuous Time Linear Equalization (CTLE)" on page 18
- "Decision Feedback Equalization (DFE)" on page 19

Mathematical Description of FFE

For the purpose of this discussion, assume the FFE algorithm we are using has three taps. (For more information on taps and how to determine the number to use, see "Taps" on page 32.) Taps are unitless correction factors applied to voltage levels to correct them. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

The mathematical description of a three-tap FFE is as follows:

$$e(t) = c_0 r(t-0T_D) + c_1 r(t-1T_D) + c_2 r(t-2T_D)$$

where:

- $e(t)$ is the corrected (or equalized) voltage waveform at time t.
- T_D is the tap delay.

- $r(t-nT_D)$ is the uncorrected input waveform n tap delays before the present time.
- c_n is the correction coefficient (tap) multiplied with the version of the uncorrected waveform that has been time-advanced by n tap delays.

So FFE obtains the corrected (or equalized) voltage level at the location of interest on the waveform (time t) by forming a sum of the taps and voltage levels of the previous two tap-delayed locations as well as the location of interest before being equalized.

Notice that if the FFE method above used a tap delay equal to one unit interval, then it would use three bits to correct one bit (it uses the bit of interest and the previous two bits). This is because we used three taps in the algorithm. If we had used five taps then the equation above would have two more terms added to the sum [$c_3r(t-3T_D)$ and $c_4r(t-4T_D)$], which would mean the FFE algorithm would have used the bit of interest and the previous four bits to correct the bit of interest's voltage level.

Description of FFE Acting on a Waveform

Another way to understand FFE is to see how it acts on a waveform. Assume the unequalized waveform measured at the receiver looks like the following:
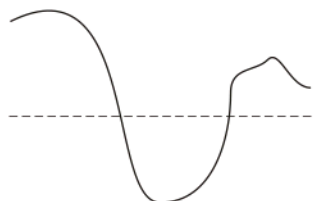
Now, apply FFE and see how it progresses along the signal to correct it. Again, assume we are using three taps in our FFE algorithm.
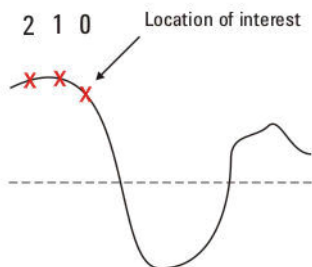
The part of the waveform labeled as the "location of interest" is the voltage we are trying to correct (or equalize). The locations on the waveform marked **1** and **2** are determined by the tap delay you specify. Location 1 is one tap delay away from the location of interest and location 2 is two tap delays away from the location of interest.

It is important to note that we are working just with voltage levels in the waveform. At no point in the process are we going to use logical decisions (1's or 0's) in the FFE algorithm.

First we need to know the tap values for the distortion present in our transmission channel. Determining these values is discussed in "Taps" on page 32. For now, assume these three tap values have been determined (call them Tap0, Tap1, and Tap2 corresponding to c_0, c_1, and c_2 in the equation above).

The oscilloscope measures the voltage of the location of interest (location **0**) and then multiples it by the Tap0 value. Then the oscilloscope measure the voltage of location **1** and multiplies it by the Tap1 value. Next, the oscilloscope measures the voltage of location **2** and multiplies it by the Tap2 value. Finally, the oscilloscope adds each of these voltage/tap products together to obtain the corrected (or equalized) voltage for location **0**. The voltage at the location **0** will then be adjusted to this corrected value.

Now the three Xs move along with the sampling rate to their next locations on the waveform.
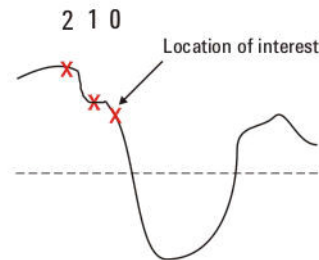
Notice the old "location of interest" (now labeled **1**) has its new equalized voltage level. You can see the tap values caused the voltage level to drop for that location.

The same process is repeated for the new location of interest. Location **0**'s voltage level is multiplied by Tap0 and then added to the product of location **1**'s voltage and Tap1. Finally, this is added to the product of location **2**'s voltage and Tap2 to give bit 0's new corrected voltage.

The three Xs will then move along with the sampling rate again to their new locations and repeat the entire process. This continues until the entire waveform has been traversed.

FFE Decision Diagram

It may also be useful to see the decision diagram for a three-tap FFE.
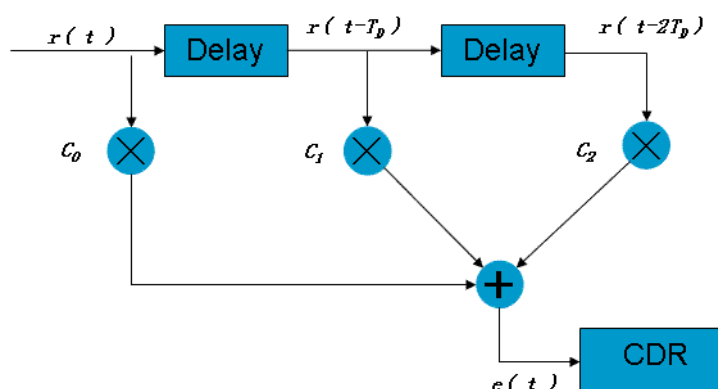
where:

- $r(t-nT_D)$ is the input waveform n tap delays before the present time.

- C_n is the nth coefficient (tap).

- T_D is the tap delay.

- e(t) is the equalized waveform at time t.
- CDR is Clock/Data Recovery.

This is just a different way of looking at the process described in the previous two sections. Again you see that each voltage level is multiplied by its corresponding tap value and then all of these products are summed together to give the new equalized voltage for the location of interest.

Understanding FFE Through Frequency Response Plots

One final way to understand FFE is in terms of the frequency response plot of your signal.
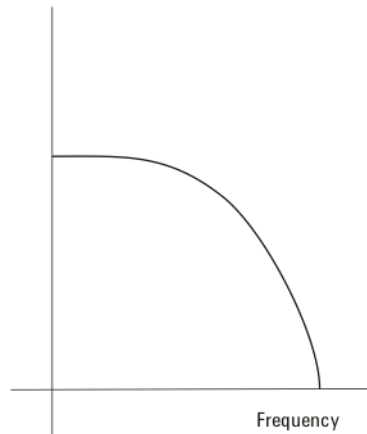
Frequency

Ideally, you would like to transmit a signal whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented. However, the channel sometimes attenuates the signal at higher frequencies (as shown in the plot above).

To correct for this attenuation, we can apply a peaking filter via equalization. With FFE, we apply a filter response as shown in the plot below (dashed line) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).

Frequency
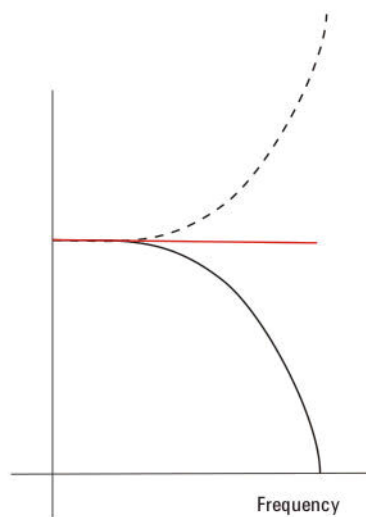
Remember, inter-symbol interference (ISI) is a result of the non-flat response of the channel. When we flatten the response, we are eliminating the ISI.

Continuous Time Linear Equalization (CTLE)

Continuous Time Linear Equalization (CTLE) is an alternate linear equalization method to Feed-Forward Equalization (FFE). It acts as a linear, low-pass filter. CTLE is defined in the frequency domain.

CTLE is standards-driven, so many of the coefficient values you enter come from the standard.

When you select the **CTLE** method in the Equalization dialog box (**Analyze > Equalization...**), the equation or graph for the filter is shown, and the remaining fields in the dialog box let you enter values for the equation's variables.

See: "Linear Eq, CTLE Tab" on page 57.

See Also   ·   "Feed-Forward Equalization (FFE)" on page 12

            ·   "Decision Feedback Equalization (DFE)" on page 19

Decision Feedback Equalization (DFE)

Decision Feedback Equalization (DFE) is an equalization technique so named because it makes logical decisions (whether a bit is a 0 or a 1) and then feeds that information back to help determine whether the current bit is a 1 or a 0.

There are multiple ways to implement DFE. This topic discusses the DFE algorithms used in the Infiniium oscilloscope.

Serial data receiver hardware cares only about the voltage of the serial waveform at the time of the clock edge. When the clock edge occurs, the waveform voltage is compared to a decision threshold calculated based on the DFE tap values. This determines the serial data level value (or symbol value if there are more than two levels) of the serial data.

Some serial data receiver hardware uses different sampling circuitry depending on the DFE tap values to determine the voltage at the time of the clock edge. Either way, it is the DFE tap values that determine the thresholds that are used for the sample.

As an oscilloscope user, you may want to see a real-time eye of the DFE-equalized waveform (even though this signal never really exists in serial data receiver hardware). In a hardware receiver, the input waveform does not change; it is typically just the decision threshold that changes.

Therefore, to display a DFE-equalized waveform or real-time eye, the oscilloscope must model the effects of equalization using the same DFE tap values used in hardware receivers.

The Infiniium oscilloscope software models DFE equalization in two different ways, depending on the **Equalize in place** or **Display as function** selection:

·   **Equalize in place** — Instead of moving the decision threshold, the oscilloscope leaves the decision threshold and voltage reference in place (constant) and moves each real-time eye unit interval plot up or down according to the DFE tap values. With this selection, no DFE-equalized waveform is displayed.

Note that real-time eye must be enabled on the source signal in order to use the DFE **Equalize in place** option.

DFE "in-place" is described in four different ways. Each description contains the same basic content, but lets you choose the viewpoint that best suits your understanding:

- **Display as function** — The DFE function waveform is created by summing the original unequalized signal with a DFE threshold waveform made up of concatenated decision thresholds based on the DFE tap values. For more information, see:

  - "The DFE "Display as function" Waveform" on page 26

A real-time eye can be displayed for each method of modeling DFE equalization; however, because the methods are different, the resulting real-time eyes are also different. For more information, see:

- "DFE "Equalize in place" vs. "Display as function" Real-Time Eyes" on page 27

The Infiniium oscilloscope can also perform Feed-Forward Equalization (FFE) or both DFE and FFE in combination.

NOTE   Jitter analysis can be performed when DFE is enabled (on non-return-to-zero signals). However, jitter graphs are not valid and are not displayed when DFE is enabled.

See Also
- "DFE Questions & Answers" on page 29
- "Feed-Forward Equalization (FFE)" on page 12
- "Continuous Time Linear Equalization (CTLE)" on page 18

Mathematical Description of DFE "in-place"

For the purpose of this discussion, assume the DFE algorithm we are using has two taps. Taps are unitless correction factors applied to voltage levels to correct them. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

Before looking at the mathematical description of DFE, it is important to understand the results of the algorithm. Generally, DFE calculates a correction value that is added to the logical decision threshold (the threshold above which the waveform is considered a logical high and below which the waveform is considered a logical low). Therefore, DFE results in the threshold shifting up or down so new logical decisions can be made on the waveform based on this new equalized threshold level.

The mathematical description of the correction value added to the decision threshold for a two-tap DFE is as follows:

$$V(k) = c_1 s(k - 1) + c_2 s(k - 2)$$

Where:

- $V(k)$ = the correction voltage added to the decision threshold used when determining the logic value of bit k.
- $s(k-n)$ = the logic value of the data bit located n bits prior to bit k.

The logic high value of s(n-k) is equal to the **Upper Target** field in the user interface while the logic low value of s(n-k) is equal to **Lower Target** (for more information on the Upper and Lower Target fields, see "DFE Taps Tab" on page 73).

- c_n = the correction coefficient (tap) for the bit n bits prior to the bit of interest.

So, for DFE to obtain the corrected voltage offset for the threshold level at the bit of interest, it first needs to be seeded with the correct bit values for the first several bits to get started. Assuming the logical decisions for the first several bits are correct, the algorithm can feed them forward to determine the logical value of the current bit.

For a two-tap DFE, the two bits previous to the current bit need to have their bit levels already determined. Then the algorithm multiplies their bit levels by their corresponding tap values. The sum of these two tap/bit level products gives the amount the decision threshold should be shifted. However, instead of shifting the threshold, the Infiniium oscilloscope leaves this constant and shifts the waveform relative to it.

Notice the DFE method above used two previous bits to correct the current bit. This is because we used two taps in the algorithm. If we had used four taps, the equation above would have two more terms added to the sum [c_3s(k-3) and c_4s(k-4)], which would mean the DFE algorithm would have used the previous four bits to correct the bit of interest's voltage level.

Also notice the bit of interest is not used in the summation. FFE uses previous bits and the current bit in its sum, but DFE uses only previous bits.

Description of DFE "in-place" Acting on an Eye Diagram

Another way to understand DFE is to see how it acts on an eye diagram.

First, look at the following segment of a serial data signal.
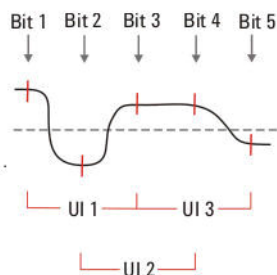
The vertical red bars on the waveform correspond to the clock edges for each bit and the red segments at the bottom of the figure delineate the three unit intervals (UI) that will be overlapped on the display to form an eye diagram. The gray line is the logical threshold level (bits with voltages above this are considered logical 1s and those with voltages below this are considered logical 0s).

Now if you overlap these three unit intervals to form an eye diagram, the logical high voltage levels for unit intervals 2 and 3 will not be as high as unit interval 1. In other words, the trajectories from unit intervals 2 and 3 will result in the partial closing of the eye.

Let us equalize the voltage for bit 4 using two taps to correct unit interval 3.

First, we need to know the tap values for the distortion present in our transmission channel. Determining these values is discussed in **"Taps"** on page 32. For now, assume these two tap values have been determined (call them Tap1 and Tap2 corresponding to c1 and c2 in the equation in **"Mathematical Description of DFE "in-place""** on page 20).

The DFE algorithm is first seeded with the correct bit values (high or low) for the first several bits to get it started. Assuming the algorithm has correctly determined the initial bit values, it can feed forward previous bit values to determine the current bit. So, for our example, we assume the oscilloscope has already determined that bit 2 is low and bit 3 is high.

The DFE algorithm would then take the ideal high value for the third bit (+1, for example) and multiply it by Tap1. Then it would add this product to the product of the ideal low value for the second bit (-1, for example) and Tap2.

This sum would then give the amount the decision threshold should be shifted. The resulting threshold shift might be the shift marked by the blue line in the plot below.
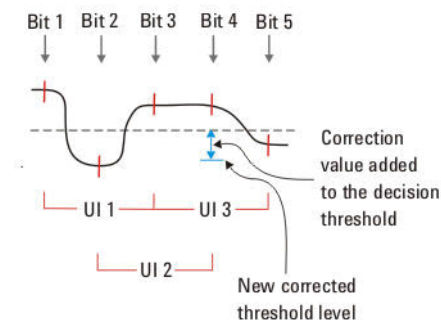

Many DFE algorithms would then shift the voltage threshold (the dashed gray line) to this new voltage level at the location of the fourth bit, but the Infiniium oscilloscope does the opposite. Instead of shifting the voltage threshold down in this example, it keeps the threshold constant and shifts the bit four voltage level up by the same amount (the amount marked by the blue arrow).

This would have the effect on the eye of raising the corresponding trajectory, opening up the eye in the process.

The algorithm would then shift forward one index so the new bit of interest would be bit five and the two bits used to equalize it would be bits three and four. This process repeats itself until the entire signal has been traversed.

DFE "in-place" Decision Diagram

It may also be useful to see the decision diagram for DFE.
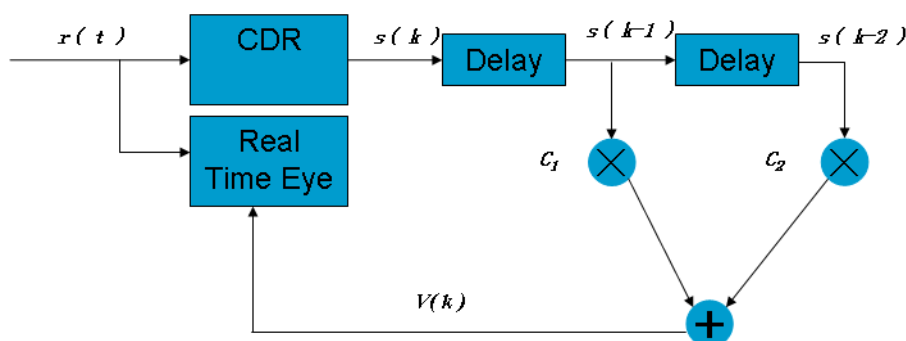
where:

- r(t) is the unequalized analog waveform voltage at time t.
- s(k-n) is the logic value (either Upper Target or Lower Target) for the bit n tap delays prior to the current bit.
- C_n is the nth coefficient (tap).
- CDR is Clock/Data Recovery.
- V(k) is the correction voltage added to the decision threshold used when determining the logic value of bit k. (Remember that Infiniium leaves the threshold constant and shifts the waveform by this amount in the opposite direction.)

This decision diagram is just another way of looking at the process described above.

The DFE algorithm is seeded with several correct bit values to get started. Then there is a tap delay (corresponding to each clock edge) between each bit. Each previous bit used in the algorithm is determined to be either high or low and is then multiplied by its corresponding tap value. These voltage/tap products are summed to determine how much to shift the waveform relative to the logical decision threshold.

Understanding DFE "in-place" Through Frequency Response Plots

One final way to understand DFE is in terms of the frequency response plot of your signal.
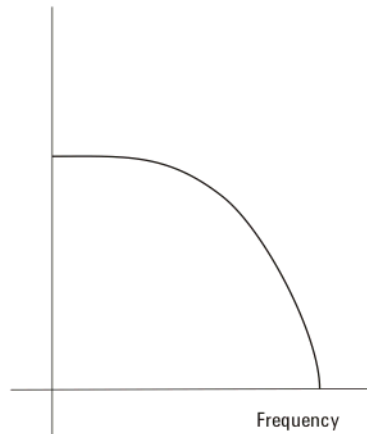
Ideally, you would like to transmit a signal through a channel whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented. However, the channel response sometimes attenuates the signal at higher frequencies (as shown in the simple plot above).

To correct for this attenuation, we can apply a peaking filter via equalization. With DFE, we apply a peaking filter response as shown in the plot below (dashed line) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).

Frequency
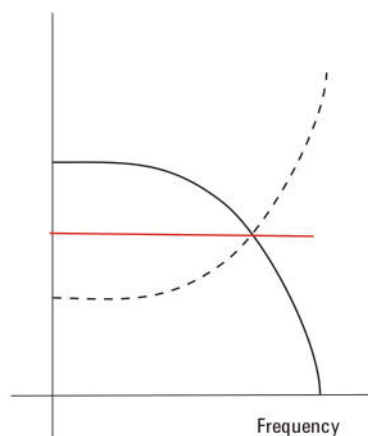
Remember inter-symbol interference (ISI) is a result of the non-flat response of the channel. When we flatten the response, we are eliminating the ISI.

To get the eye at the receiver to be the same size as the eye at the transmitter, we need to apply a gain term. To understand why the gain term is needed, remember what the frequency response looks like for FFE.
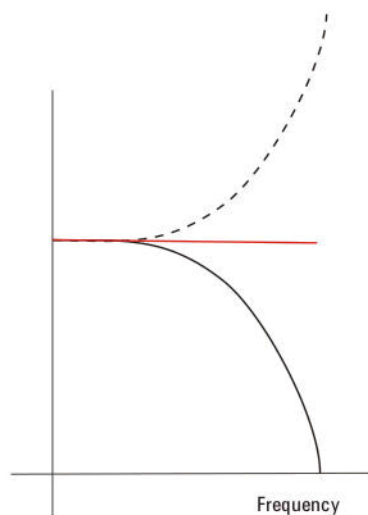
Frequency

You can see we did not attenuate the low frequencies, but instead left them alone.

However, with the DFE response, we are attenuating the low frequencies because we have shifted the entire FFE response down. This means the flat response we obtain in the end will be shifted down as well. You can use the gain factor to shift the flat response back up to its correct value.

This makes the eye at the receiver the same size as the eye leaving the transmitter.

The DFE "Display as function" Waveform

Beginning with the 6.40 version of Infiniium oscilloscope software, there is a new implementation of DFE that produces a waveform.

If a lane is using **Equalize in place** (modifying the waveform itself), no DFE waveform is produced, and the DFE affects the real-time eye only. This is how DFE worked in the Infiniium oscilloscope software versions before 6.40.

However, if **Display as function** (creating a separate equalized waveform) is selected for the equalization lane, you can view a waveform with DFE equalization applied.

How the DFE Waveform is Created   The DFE waveform is created by summing the original unequalized signal with the DFE threshold waveform.

1 First, a DFE threshold waveform is created.

The DFE threshold waveform is constructed by creating a waveform of concatenated decision threshold voltages, one per unit interval. The decision threshold for every unit interval is computed based on the DFE taps.

The time duration of the decision thresholds is the unit interval (bit time).

Unequalized waveform

Clocks

Decision threshold times

Threshold waveform
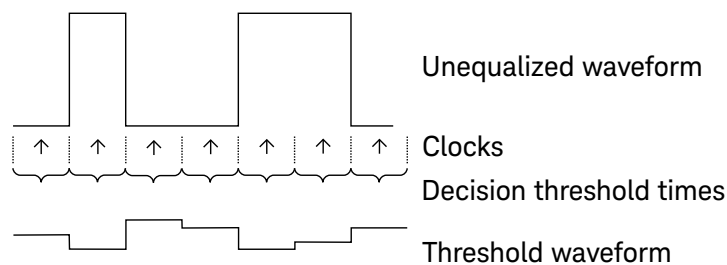
The DFE threshold waveform has abrupt changes in voltage at the unit interval boundaries.

2 Optionally, a low-pass filter is applied to the DFE threshold waveform.

When **Display as function** is selected in the Equalization dialog box, the **Threshold Bandwidth** mode specifies whether a low-pass filter is applied to the threshold waveform prior to summing it to the unequalized waveform.

· If **Off** is selected, no low-pass filter is applied.

- If **Track Source Bandwidth** is selected, the low-pass bandwidth 0.75 times the Nyquist bandwidth of the source (see **Global Bandwidth Limit** in the online help).

- If **Manual** is selected, you can specify the low-pass bandwidth.

3 Optionally, delay is applied to the DFE threshold waveform.

The **Threshold Delay** field allows the threshold waveform to be shifted in time relative to the unequalized waveform prior to summing them.

4 Finally, the original unequalized signal is summed with the (possibly filtered and delayed) DFE threshold waveform to create a DFE equalized waveform that is displayed as a function.

DFE "Equalize in place" vs. "Display as function" Real-Time Eyes

How DFE "Equalize in place" Real-Time Eyes are Created

When DFE **Equalize in place** is selected, no DFE waveform is produced, and the DFE affects the real-time eye only. This is how DFE worked in the Infiniium oscilloscope software versions before 6.40.

In the DFE "in-place" case, the Infiniium oscilloscope models moving the decision threshold on a stationary waveform (as in receiver hardware) by instead leaving the decision threshold and voltage reference in place (constant) and moving the real-time eye plot for each unit interval up or down according to the DFE tap values.

From one unit interval to the next, with decision thresholds changing (as in receiver hardware) or oscilloscope plots moving up or down, you might expect discontinuities in the real-time eye waveform at unit interval boundaries. However, in DFE "in-place" real-time eyes, the oscilloscope plots the entire width of the screen (which is typically two unit intervals wide) according to the DFE tap values for the unit interval at the clock location (in the middle of the screen). This way, there are no discontinuities present at the unit interval boundaries because the decision threshold extends beyond the unit interval boundary in time. Effectively, in this model, there is infinite bandwidth in the transition from one decision threshold to the next.

For more information on DFE "in-place", see:

- "Mathematical Description of DFE "in-place"" on page 20
- "Description of DFE "in-place" Acting on an Eye Diagram" on page 21
- "DFE "in-place" Decision Diagram" on page 23
- "Understanding DFE "in-place" Through Frequency Response Plots" on page 24

How DFE "Display as function" Real-Time Eyes are Created

When DFE **Display as function** is selected, the oscilloscope computes a new equalized waveform that is distinct from the original waveform (see "The DFE "Display as function" Waveform" on page 26). The DFE function waveform can then be plotted as a real-time eye, just like any other serial data waveform.

Typically, there are strange and unnatural looking aberrations in the waveform at the unit interval boundaries.

Differences in the DFE "Equalize in place" and "Display as function" Real-Time Eyes

Because of differences in the way DFE is applied "in-place" (that is, moving a waveform up or down based on DFE decision thresholds when creating a real-time eye) versus "as-a-function" (that is, adding a low-pass filtered DFE decision threshold waveform to the unequalized waveform to create a DFE function waveform), the real-time eye of a signal that is DFE-equalized "in-place" and the real-time eye of a signal that is DFE-equalized "as-a-function" will look quite different, even if the DFE settings are identical. This is expected behavior.
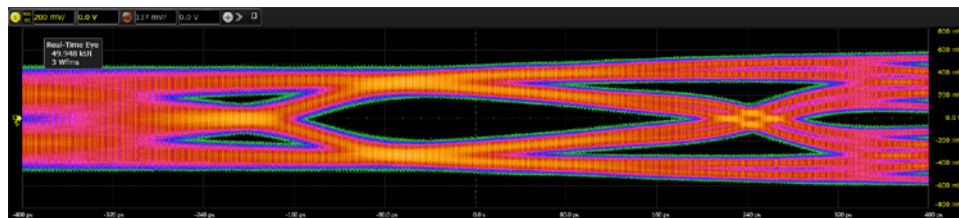

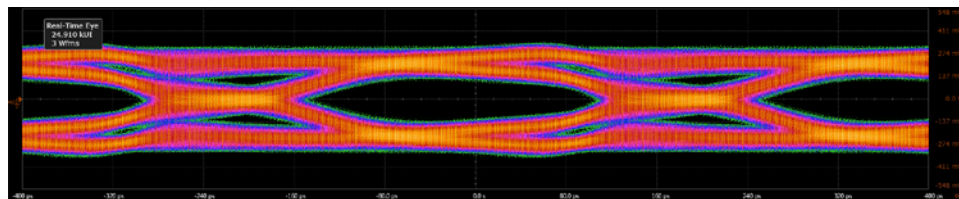**Figure 1**   Real-time eye of DFE "in-place"

Figure 2   Real-time eye of DFE "as-a-function" waveform

Which DFE Real-Time Eye Should You Use?

If the most important concern of receiver hardware is the voltage at the clock edge, either method of DFE real-time eye generation can be used because they both are primarily concerned with modeling equalization effects at the clock edge. Eye height and noise measurements at the clock edge are relevant to this concern and are very similar in both the "in-place" and "as-a-function" real-time eyes.

An advantage the DFE "as-a-function" method has over the DFE "in-place" method is that the function waveform allows other measurements or operations to be performed on it, and it can be saved to a file. There is no waveform associated with the DFE "in-place" method.

If receiver hardware is concerned about the decision threshold extending for the width of the unit interval, the DFE "in-place" real-time eye may give more useful information.

If receiver hardware is concerned about the transition of decision thresholds from one unit interval to the next (especially if you know the bandwidth of your circuit), the DFE "as-a-function" method provides more useful information. In this case, eye

width and jitter measurements may be useful. (The "compression to the left and fanning out to the right" trait of the DFE "in-place" real-time eye method — see "Why Does the DFE Real-Time Eye Have the Shape That It Does?" on page 30 — may disqualify it from providing useful eye width or jitter information.)

DFE Questions & Answers

- "How Do I Map DFE Voltage Tap Values to Unitless Tap Values?" on page 29
- "Why Does the DFE Real-Time Eye Have the Shape That It Does?" on page 30

How Do I Map DFE Voltage Tap Values to Unitless Tap Values?

In the Equalization Tap Setup dialog's **DFE Taps** group box, you find these DFE variables:

Upper Target	This is the DC voltage of the one level.
Lower Target	This is the DC voltage of the zero level.
Gain	Scale to be applied to the resultant waveform to display it relative to the original DC levels of the input.
T1 ... Tn	Unitless tap values (Tap 1 ... Tap n).

To convert unitless tap values into the voltage applied to the waveform, first calculate the Amplitude (A) and Offset (O):

Amplitude (A) = (Upper Target – Lower Target) / 2

Offset (O) = (Upper Target + Lower Target) / 2

Then, for 1 tap DFE:

Bit(0-1)	V
1	V = O + A * (T1 * 1)
0	V = O + A * (T1 * -1)

Or, for a 2 tap DFE:

Bit(0-2)	Bit(0-1)	V
0	0	V = O + A * (T1 * -1 + T2 * -1)
0	1	V = O + A * (T1 * 1 + T2 * -1)
1	0	V = O + A * (T1 * -1 + T2 * 1)
1	1	V = O + A * (T1 * 1 + T2 * 1)

Or, for an n tap DFE:

V = O + A * Sum(Tn * (Bit(0–n) == 1 ? 1 : -1))

where Bit(0) is 1 or 0 of current bit, Bit(0-1) is 1 or 0 of previous bit, Bit(0-n) is 1 or 0 of previous nth bit, etc.

So, if you have DFE voltage tap values, H1 ... Hn, you can find the unitless tap values as follows:

Hn = O + A * Tn

Tn * A = Hn – O

Tn = (Hn – O) / A

Substituting Upper and Lower values for Amplitude (A) and Offset (O):

Tn = (Hn – (Upper + Lower) / 2)) / (Upper – Lower) / 2

Tn = 2(Hn – (Upper + Lower) / 2)) / (Upper – Lower)

Tn = (2Hn – (Upper + Lower)) / (Upper – Lower)

Why Does the DFE Real-Time Eye Have the Shape That It Does?

That is, why is there compression of the eye to the left of the bit and fanning out to the right?

Consider a case with a one-tap DFE. When the previous bit was a 0 and the current bit is a 0, DFE correction compresses the data (moves toward zero).

DFE Correction Graph, Previous Bit Same as CurrentBit
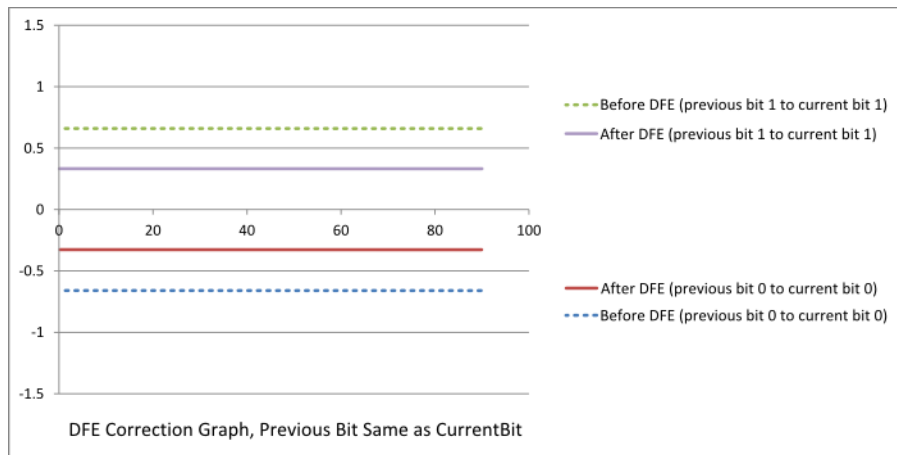
When the previous bit is a 0 and the current bit is a 1, correction is applied to shift the bit up. This compresses the data (moves toward zero) to the left of the bit of interest and causes it to fan out (moves away from zero) with greater amplitude to the right.

DFE Correction Graph, Previous Bit Different than Current Bit
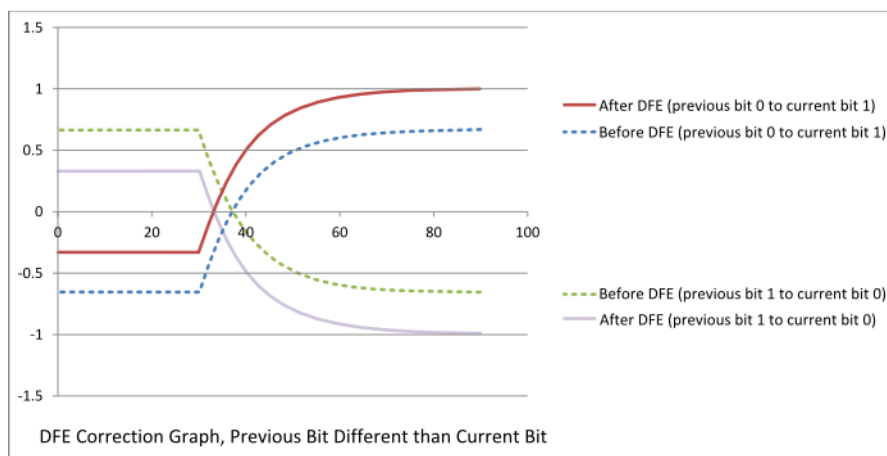
The same phenomenon occurs for 1 bits staying 1's. They are shifted down, compressing the data. 1 bits transitioning to 0's will be shifted down causing a compression on the left and fanning out on the right.

Taps

Taps are dimensionless correction factors applied to the bit voltage levels in serial data patterns during equalization. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

To see how taps are implemented in equalization, see "Mathematical Description of FFE" on page 12 and "Mathematical Description of DFE "in-place"" on page 20.

- "How Many Taps Should You Use?" on page 32
- "What Tap Values Should You Use?" on page 32

How Many Taps Should You Use?

Generally, two taps will open your eye diagram significantly. However, there may be times when the signal error may span over more than two bits and in these cases, more taps will be required.

A simple way to see how many taps you need is a trial-and-error approach. Simply set the number of taps (perhaps start with 3-4) and then have the oscilloscope optimize the tap values for your signal (optimization is discussed below). Look at the tap values after they have been optimized and see if the higher tap values (for example, start with tap 4, then 3, ...) are contributing much or if they are near zero. If there is a tap value near zero, this tap is unnecessary. If all the tap values are significant then add more taps, re-optimize, and investigate again. This is a good approach if you do not know how many tap values to use for your specific setup.

Notice the user interface lets you enter up to 40 taps. More than likely, you are not going to have an error that spans 40 bits, but the oscilloscope lets you use this many taps in case you want to use more than one tap per bit. For instance, if you found you needed four taps when you used 1 tap/bit, you would probably need eight taps if you used 2 taps/bit.

What Tap Values Should You Use?

There are two ways to set up the tap values for your specific application:

- If you know what tap values to use for your device under test (DUT), you can enter them in the "Equalization Tap Setup Dialog Box" on page 71:

  a   Choose **Analyze > Equalization...**.

  b   In the Equalization dialog box, click **Tap Setup...**.

  c   In the Equalization Tap Setup dialog box, select the **# of Taps**.

  d   Enter the tap values in the **Tap N** fields.

- If you do not know which tap values to use, you can have the oscilloscope determine them for you through optimization.

The optimization process occurs when using the Equalization Wizard, but you can also run the optimization process from the "Equalization Tap Setup Dialog Box" on page 71:

a Choose **Analyze > Equalization...**.

b In the Equalization dialog box, click **Tap Setup...**.

c In the Equalization Tap Setup dialog box, enter the **# of Taps**.

d Click **Auto Set...**.

e In the Equalization Auto Tap Setup dialog box, set the controls appropriately. See "Equalization Auto Tap Setup Dialog Box" on page 67.

f Click **Run Auto Set**.

The optimizer runs and reports the tap values to you in the **Auto Tap Status** box.

The optimized taps are automatically inserted into the Equalization Tap Setup dialog box.

Using the Equalization Wizard

The Equalization Wizard is one of two ways to set up the equalization feature. (The other way is to manually set it up—see "Manually Setting Up Equalization" on page 51.)

The wizard is the recommended setup method as it walks you through the process in a simplified, step-by-step fashion. Use the manual method only if you know exactly what you are doing and are familiar with equalization.

NOTE   The wizard does not set up your signal, so you need to have whatever channel, function, or waveform memory you want to perform equalization on already set up on the oscilloscope. The wizard will assume the signal is already there.

It is also recommended that only one signal be displayed prior to running the Wizard as it is difficult to discern one signal from another with real-time eye diagrams.

- "Step 1: Lane Selection" on page 34
- "Step 2: Configuration" on page 35
- "Step 3: Thresholds" on page 38
- "Step 4: Clock Recovery" on page 40
- "Step 5: Acquisition" on page 43
- "Step 6: CTLE Setup" on page 44
- "Step 7: FFE Setup" on page 47
- "Step 8: DFE Setup" on page 49

Step 1: Lane Selection

The first step after the welcome page is the first **Configuration** page that lets you select the lane that the Equalization Wizard will be setting up.
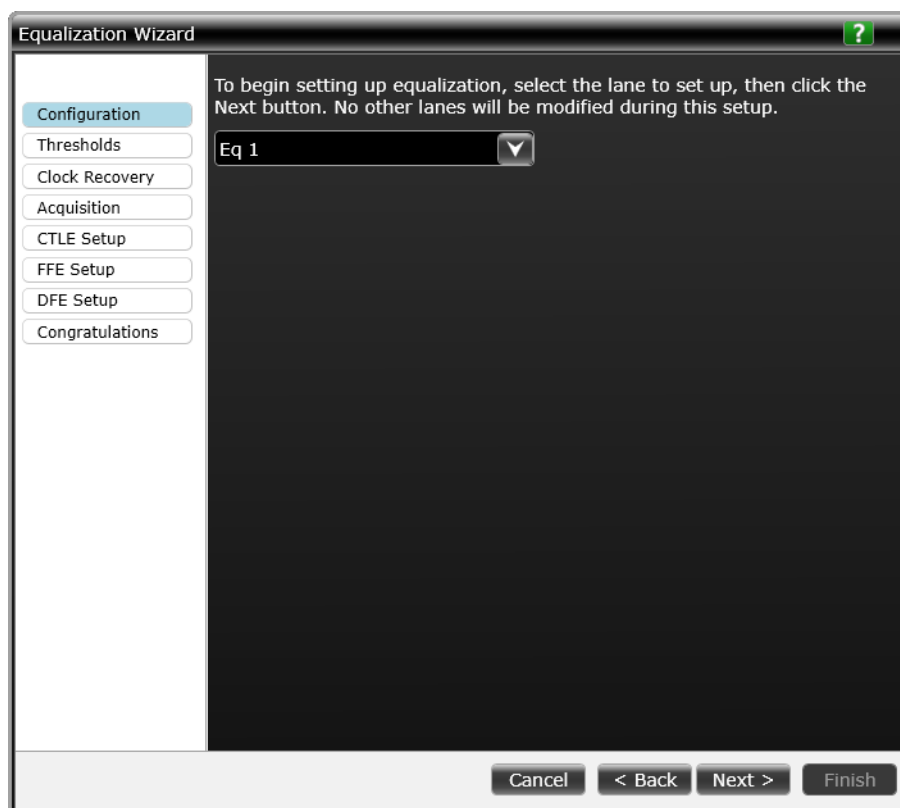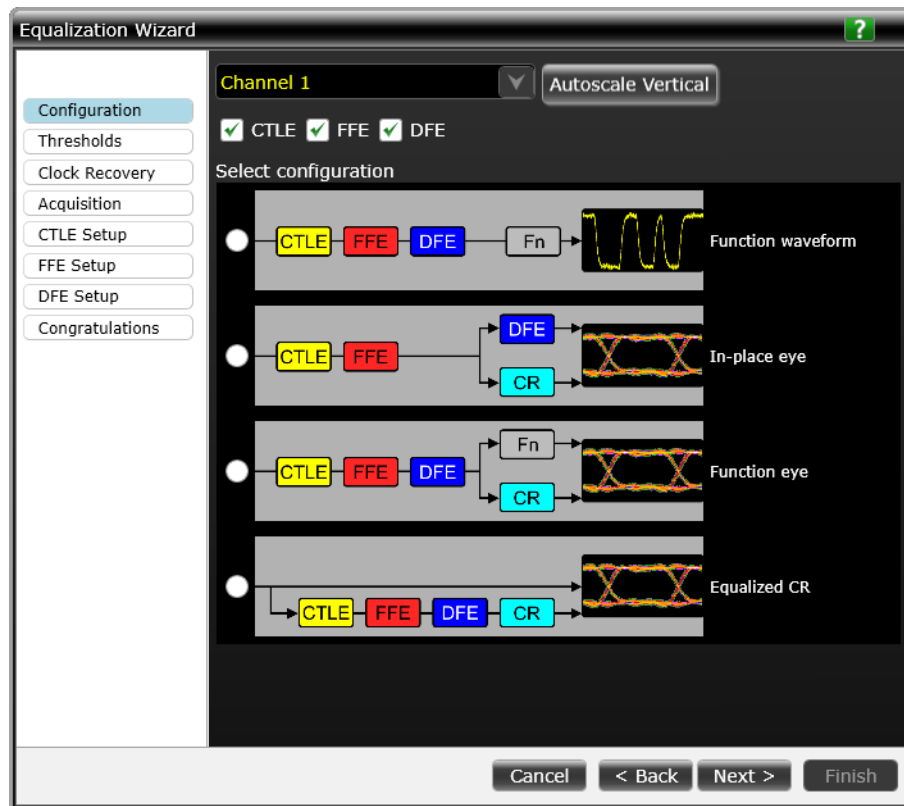
You can run the Equalization Wizard multiple times, selecting a different lane each time, to set up multiple lanes of equalization.

Next   ·   **"Step 2: Configuration"** on page 35

Step 2: Configuration

After selecting the lane, you select the equalization configuration.

- **Select the signal to equalize** — Select which source you want to equalize (channel inputs, functions, or waveform memories).

- **CTLE**, **FFE**, **DFE** — Select the type(s) of equalization that you would like to apply. You can apply any combination of the types of equalization to a lane.

- **Select configuration** — Click the option button next to the desired configuration.

  For linear equalization (CTLE and FFE), "in-place" means the equalization runs completely in hardware, greatly improving speed. For DFE, "in-place" means the pre-6.40 version of DFE is performed (that is, the DFE that modifies the display of the real-time eye only). For more information on displaying equalization "in-place" versus "as-a-function", see "Multiple "Lanes" of Equalization" on page 10.
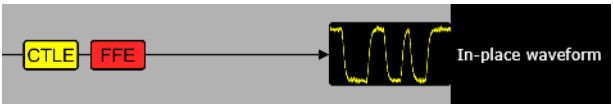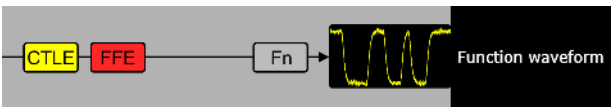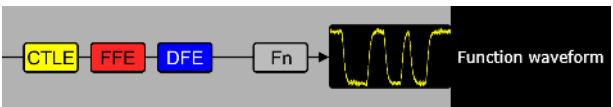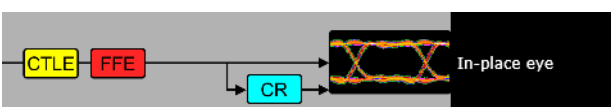
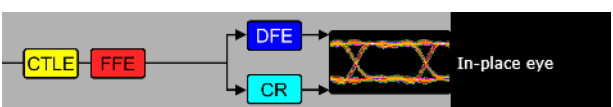**Table 1**   Equalization Configurations
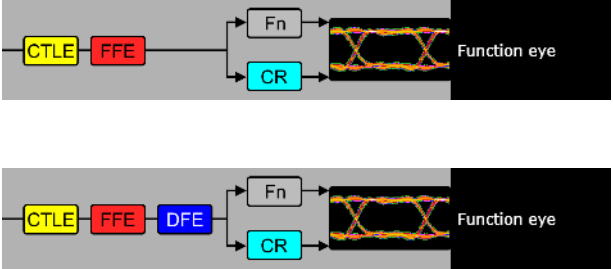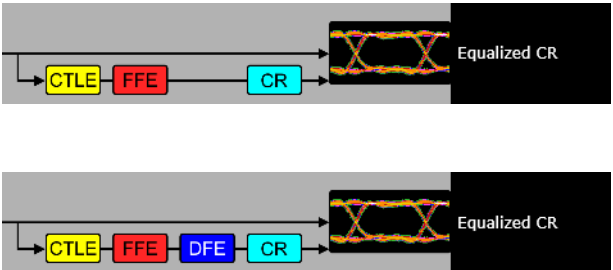
Configuration Type/Description	Example Configuration Selections
In-place waveform Equalization is applied to the source waveform in-place, but no real-time eye is displayed. The equalized source waveform is displayed. When **DFE** is selected, the **In-place waveform** configuration is not available.	
Function waveform Equalization is applied to the source waveform and displayed as a separate function waveform, but no real-time eye is displayed. The equalized function waveform is displayed.	
In-place eye Equalization is applied to the source waveform in-place, and clock recovery on the equalized waveform is used to display a real-time eye of the equalized source waveform.	

Table 1   Equalization Configurations (continued)

Configuration Type/Description	Example Configuration Selections
Function eye Equalization is applied to the source waveform and displayed as a separate function waveform. Clock recovery on the equalized waveform is used to display a real-time eye of the equalized function waveform.	
Equalized CR Equalization is used only to recover the clock, and the recovered clock is used to display a real-time eye of the original (unequalized) waveform. Clock recovery cannot be performed on a waveform if there is no open eye. So, in this case, equalization is used to create an open eye on a waveform so that clock recovery (and real-time eye analysis) can be performed.	

Step 3: Thresholds

This step sets up various parameters associated with the logical threshold applied to the serial data pattern.
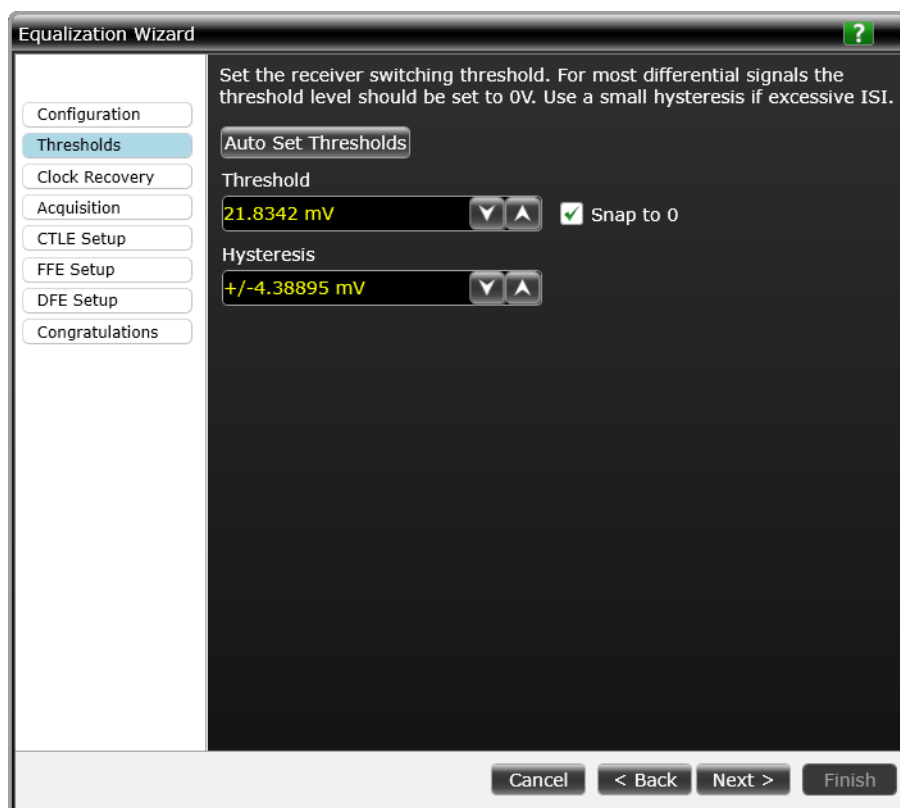
- **Auto Set Thresholds** — Clicking this button performs an analysis of the signal to automatically determine good threshold and hysteresis settings.

  The threshold level is defined to be the voltage level above which the signal is considered to be a logical 1 and below which the signal is considered to be a logical 0.

- **67 Threshold**, **56 Threshold**, **45 Threshold**, **34 Threshold**, **23 Threshold**, **12 Threshold**, **01 Threshold** — If you see these controls, the signal is a PAM signal. You have these options for the PAM signal types:

  - PAM-3: **12 Threshold**, **01 Threshold**
  - PAM-4: **23 Threshold**, **12 Threshold**, **01 Threshold**
  - PAM-6: **45 Threshold**, **34 Threshold**, **23 Threshold**, **12 Threshold**, **01 Threshold**
  - PAM-8: **67 Threshold**, **56 Threshold**, **45 Threshold**, **34 Threshold**, **23 Threshold**, **12 Threshold**, **01 Threshold**

  For more information on these controls, see the "PAM Measurements" help.

- **Threshold** — Lets you set the threshold voltage (switching) level.

  When **Snap to 0** is enabled, thresholds within ±10 mV are moved to zero. This is useful for differential signals.

- **Hysteresis** — The hysteresis helps define what is considered an edge by the oscilloscope.

  For example, if hysteresis is set to +/-5 mV, this means the signal must go through a 10 mV swing (-5 mV to 5 mV) to be considered an edge. If you make the hysteresis value too large, the oscilloscope may skip edges and incompletely recover the clock. If you make this value too small, the oscilloscope will mistake noise for edges. In most cases, +/-5 mV is a good setting, although you may need to increase or decrease this setting based upon your specific signal and the amount of noise.

Next   ·

Step 4: Clock Recovery

The clock recovery algorithms available in this step depend on the selected configuration. In this step, the wizard lets you:

- Select a clock recovery preset and specify a data rate (if known). You can enter the data rate or click **Measure Data Rate** to populate the **Nominal Data Rate** field with the data rate measured on the input signal.
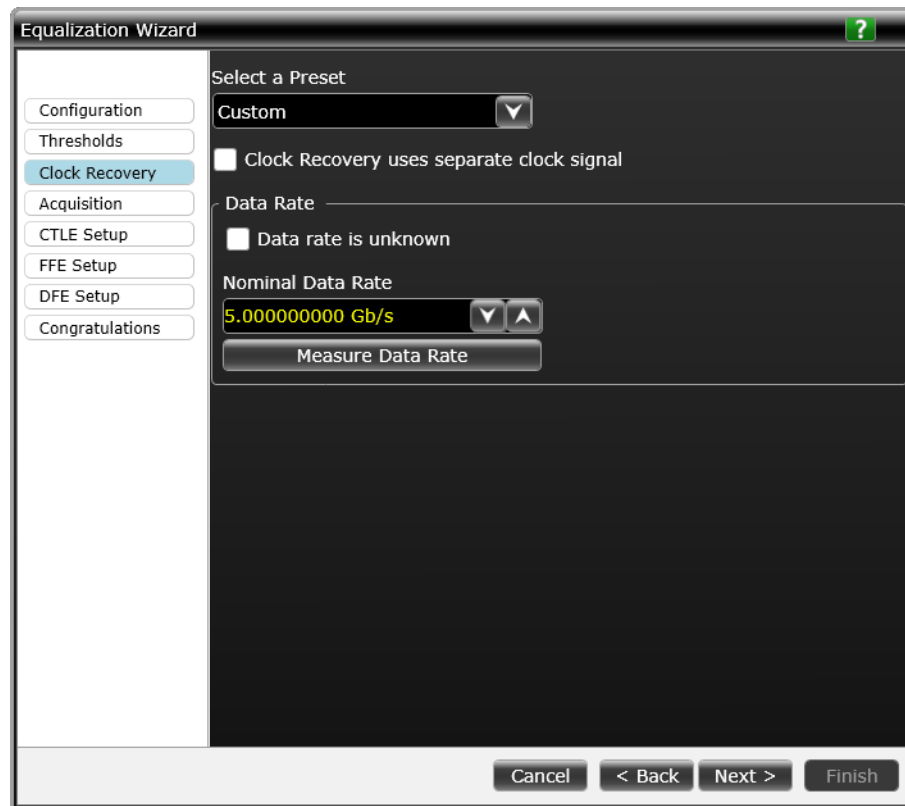
For more information on the clock recovery presets and nominal data rate settings, see Clock Recovery Dialog Box Controls in the online help.

If you see the **Nominal Symbol Rate** control, the signal is a PAM signal. For more information on this clock recovery control, see the "PAM Measurements" help.

- Specify the clock recovery method to use.
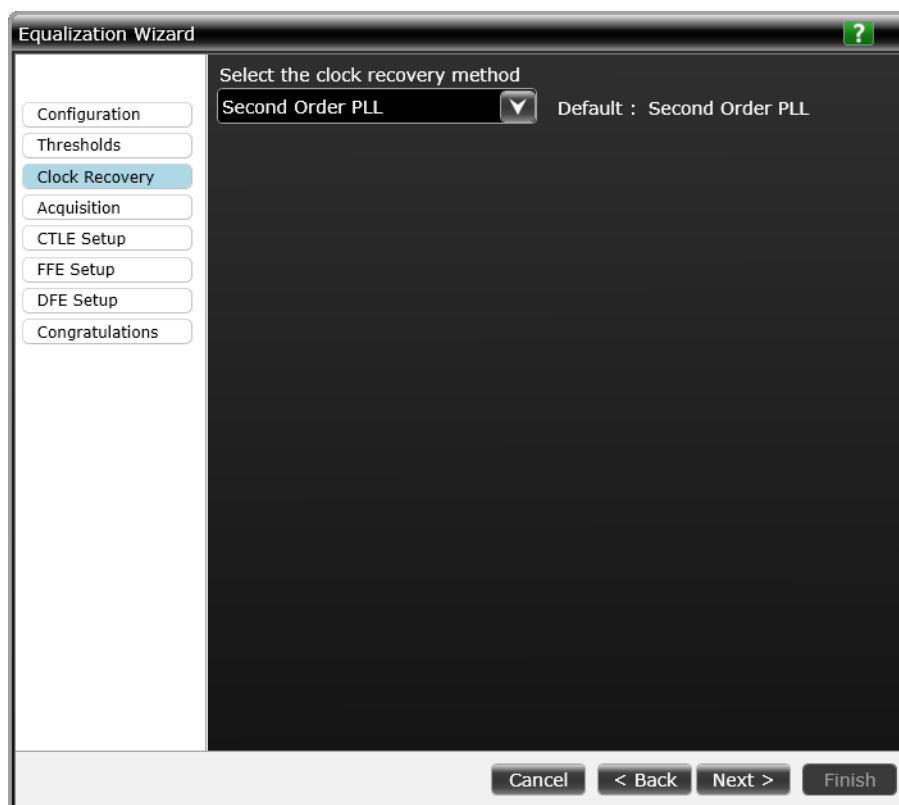
The algorithms will be a subset of the following list:

- Constant Frequency
- First Order PLL (Phase Lock Loop)
- Second Order PLL
- Third Order PLL
- Equalized First Order PLL
- Equalized Second Order PLL
- Equalized Third Order PLL
- Explicit Clock
- Explicit First Order PLL
- Explicit Second Order PLL
- Explicit Third Order PLL

Generally, you want to choose the method that approximates the method your receiver uses.

Equalized First, Second, or Third Order PLL means first FFE is applied to the closed eye to open it (because the oscilloscope cannot extract a clock from a closed eye) and then the corresponding PLL algorithm is used to extract the clock.

For more information on the clock recovery methods you can select, see Clock Recovery Dialog Box Controls in the online help.

· Specify the options for the selected clock recovery method.
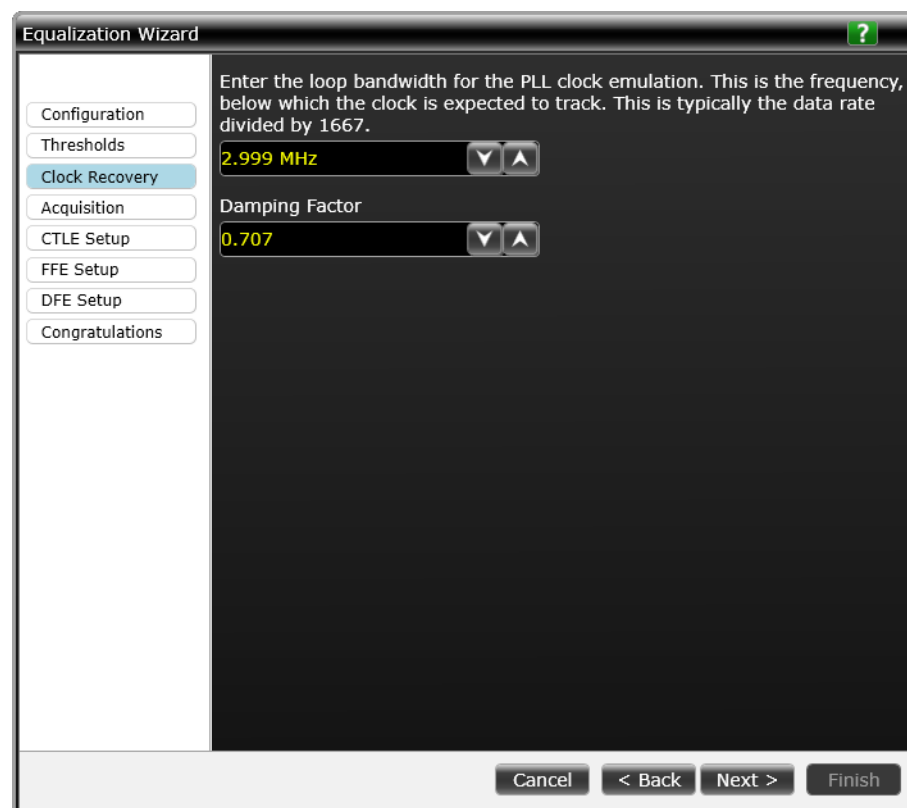
For more information on the clock recovery method options, see Clock Recovery Dialog Box Controls in the online help.

If you see the **Clock Skew** control, the signal is a PAM signal. For more information on these clock recovery controls, see the "PAM Measurements" help.

Next   ·

Step 5: Acquisition

This step sets up various parameters associated with the oscilloscope's acquisition of the signal.

The fields in this step will automatically be filled in based on the settings you chose in the clock recovery step. How these values were determined is explained in the text above each field in the user interface and is also discussed below.
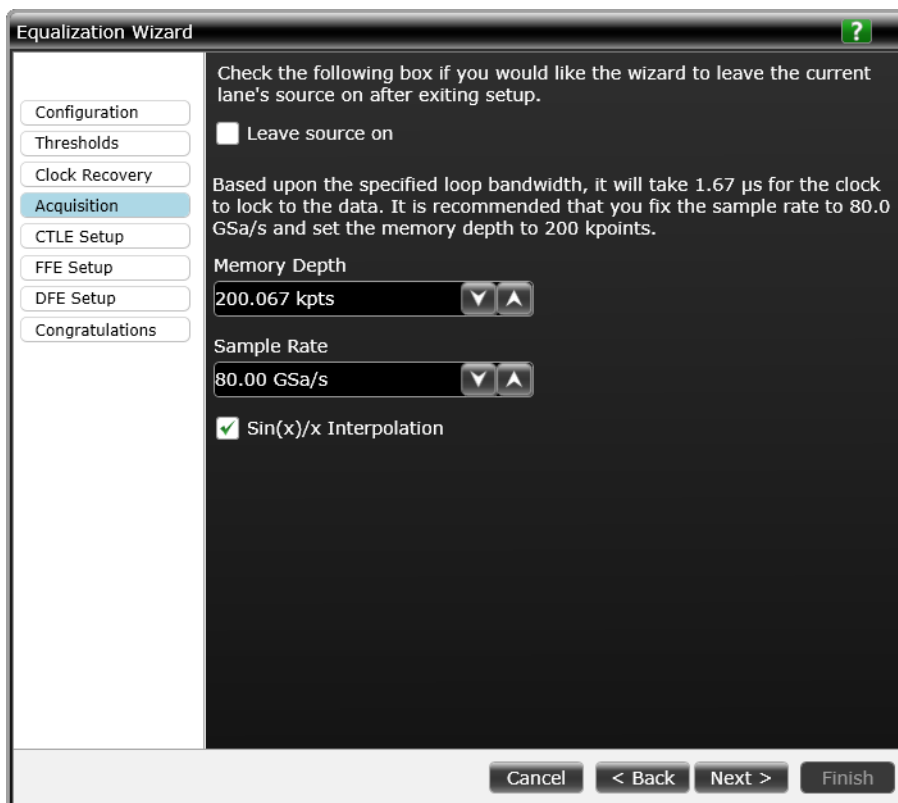
- **Leave source on** — If you asked for an equalization configuration that displays as a function (that is, not in-place), this option lets you choose whether or not the original waveform will be displayed after the wizard completes.
- **Memory Depth** — There are two ways this value is set:

  If you set Constant Frequency as your clock recovery method, the oscilloscope will determine how many samples per unit interval are acquired (it will tell you this value in the text at the top of the dialog box). The oscilloscope will then set the memory depth such that the oscilloscope displays 1,000 unit intervals [(samples /unit interval) x 1000]. Displaying 1,000 unit intervals is arbitrary. If you want to display a different number of unit intervals, multiply the number you want times the samples per unit interval.

If you use a PLL or Fibre Channel clock recovery algorithm, the oscilloscope will calculate how much time it takes for the clock to lock to the data (assumes five time constants for PLL to set up so this time is calculated via 5/(loop bandwidth)). The oscilloscope will then recommend the memory depth based upon the following equation:

Memory Depth = (5/loop bandwidth) x (3/2)

where 5 is the number of time constants for PLL to settle and 3/2 is used so some data after the PLL settles is included.

· **Sampling Rate** — The sampling rate is set to the maximum value possible for your specific oscilloscope model.

· **Sin(x)/x Interpolation** — Enable this for better effective resolution.

Next ·

Step 6: CTLE Setup

This step lets you set various parameters associated with CTLE. If the selected configuration did not include CTLE, this step will be skipped.

· **Data Rate** — This field is automatically filled in based on the settings you provided in the clock recovery step.

- **# of Poles** — Lets you select the number of poles in the CTLE filter.
  - **2 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2}}{\omega_Z} \cdot \frac{s + \omega_Z}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right)}$$

  - **3 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{p_3}}{\omega_Z} \cdot \frac{s + \omega_Z}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right) \cdot \left(s + \omega_{p_3}\right)}$$

  - **4 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{\omega_{p_1} \cdot \omega_{p_3} \cdot \omega_{p_4}}{\omega_{Z_1}} \bullet \frac{s + \omega_{Z_1} \cdot s + \omega_{p_2} \cdot A_{dc}}{s + \omega_{p_1} \cdot s + \omega_{p_2} \cdot s + \omega_{p_3} \cdot s + \omega_{p_4}}$$

  - **6 Pole, 2 Zeros** — The following equation or graph is shown:

$$\frac{\omega_{p_1} \cdot \omega_{p_3} \cdot \omega_{p_4} \cdot \omega_{p_5} \cdot \omega_{p_6}}{\omega_{Z_1} \cdot \omega_{Z_2}} \bullet \frac{s + \omega_{Z_1} \cdot s + \omega_{p_2} \cdot A_{dc} \cdot s + \omega_{Z_2}}{s + \omega_{p_1} \cdot s + \omega_{p_2} \cdot s + \omega_{p_3} \cdot s + \omega_{p_4} \cdot s + \omega_{p_5} \cdot s + \omega_{p_6}}$$
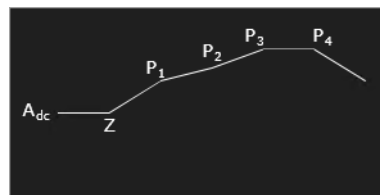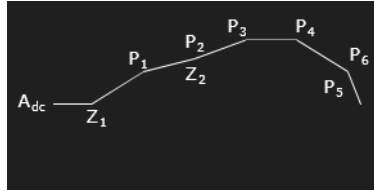

•   **3 Pole, 2 Zeros** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{p_3}}{\omega_{Z_1} \cdot \omega_{Z_2}} \cdot \frac{\left(s + \omega_{Z_1}\right) \cdot \left(s + \omega_{Z_2}\right)}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right) \cdot \left(s + \omega_{p_3}\right)}$$

•   **2 Pole, AC Gain** — (USB 3.1) The following equation or graph is shown:

$$H(s) = A_{ac} \cdot \omega_{p_2} \cdot \frac{s + \frac{A_{dc}}{A_{ac}} \cdot \omega_{p_1}}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right)}$$

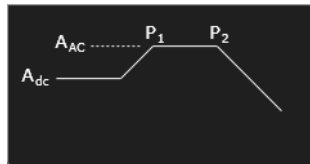
•   **2 Pole, 2 Gain** — The following equation or graph is shown:

$$H(s) = \frac{(\omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{LF}) \cdot \; s + \omega_Z \cdot 10^{\frac{A_{dc}}{20}} \; \cdot \; s + \omega_{LF} \cdot 10^{\frac{A_{dc2}}{20}}}{(\omega_Z \cdot \omega_{LF}) \cdot \; s + \omega_{p_1} \cdot \; s + \omega_{p_2} \; \cdot (s + \omega_{LF})}$$

- The remaining fields in the dialog box let you enter values for the equation's variables:

    - **DC Gain** (A_{dc})

    - **DC Gain (dB)** (A_{dc} in dB)

    - **DC Gain 2** (A_{dc2})

    - **DC Gain 2 (dB)** (A_{dc2})

    - **Zero Frequency** (ω_Z)

    - **Zero 1 Frequency** (ω_{Z1})

    - **Zero 2 Frequency** (ω_{Z2})

    - **AC Gain** (A_{ac})

    - **AC Gain (dB)** (A_{ac})

    - **Pole 1 Frequency** (ω_{p1})

    - **Pole 2 Frequency** (ω_{p2})

    - **Pole 3 Frequency** (ω_{p3})

    - **Pole 4 Frequency** (ω_{p4})

    - **Pole 5 Frequency** (ω_{p5})

    - **Pole 6 Frequency** (ω_{p6})

    - **LF Frequency** (ω_{LF})

Next   ·

Step 7: FFE Setup

This step lets you set various parameters associated with FFE. If the selected configuration did not include FFE, this step will be skipped.

- **Data Rate** — This field is automatically filled in based on the settings you provided in the clock recovery step.

- **# of Taps** — This field lets you enter the number of taps to use (up to 40). For more information on what taps are, how to determine the number of taps to use, and how to set their values, see "Taps" on page 32.

- **# of Precursor Taps** — This control lets you specify how many precursor taps you want to use.

  Remember that with standard FFE, the current bit is equalized by using its tap value along with tap values associated with bits prior to it.

  Precursor taps are tap values associated with bits in front of the current bit.

  For example, if you set **# of Precursor Taps** to zero and you were using three taps, the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you set the **# of Precursor Taps** control to one then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used.

  This gives you a chance to specify which bits surrounding the "bit of interest" are used in the equalization algorithm. The **# of Taps** is the total number you want to use, and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.

- **Bandwidth** — See "Bandwidth Limit Control" on page 61.
- **Tap Delay** — Usually when you set the number of taps, it will be at a rate of 1 tap/bit. If you are indeed using 1 tap/bit then the tap delay is defined as (1/data rate) and you can set it to this by selecting the **Track Data Rate** choice for this field.

  However, you can also set up the equalization such that multiple taps/bit are used. To do this, first set the **Tap Delay** field to **Manual**. Then compute how large to make the tap delay via the following equation:

  Tap delay = 1/[(data rate) x (# of taps per bit)]

- **Auto Set...** — After you set the number of taps, use this button to have the oscilloscope determine tap values through optimization:

  a Click **Auto Set...**.

  b In the Equalization Auto Tap Setup dialog box, set the controls appropriately. See "Equalization Auto Tap Setup Dialog Box" on page 67.

  c Then, click **Run Auto Set**.

  The optimizer runs and reports the tap values to you in the **Auto Tap Status** box.

  d Close the Equalization Auto Tap Setup dialog box to return to the Equalization Wizard.

  If you want to enter the tap values manually after completing the Equalization Wizard, go to the "Equalization Tap Setup Dialog Box" on page 71.

Next • "Step 8: DFE Setup" on page 49

Step 8: DFE Setup

This step lets you set various parameters associated with Decision Feedback Equalization. If the selected configuration did not include DFE, this step will be skipped.

- **# of Taps** — Specify the number of taps to use in the Decision Feedback Equalization. For more information on taps and how to choose the correct number of taps, see "Taps" on page 32.

- **Auto Set...** — After you set the number of taps, use this button to have the oscilloscope determine tap values through optimization:

  a Click **Auto Set...**.

  b In the Equalization Auto Tap Setup dialog box, set the controls appropriately. See "Equalization Auto Tap Setup Dialog Box" on page 67.
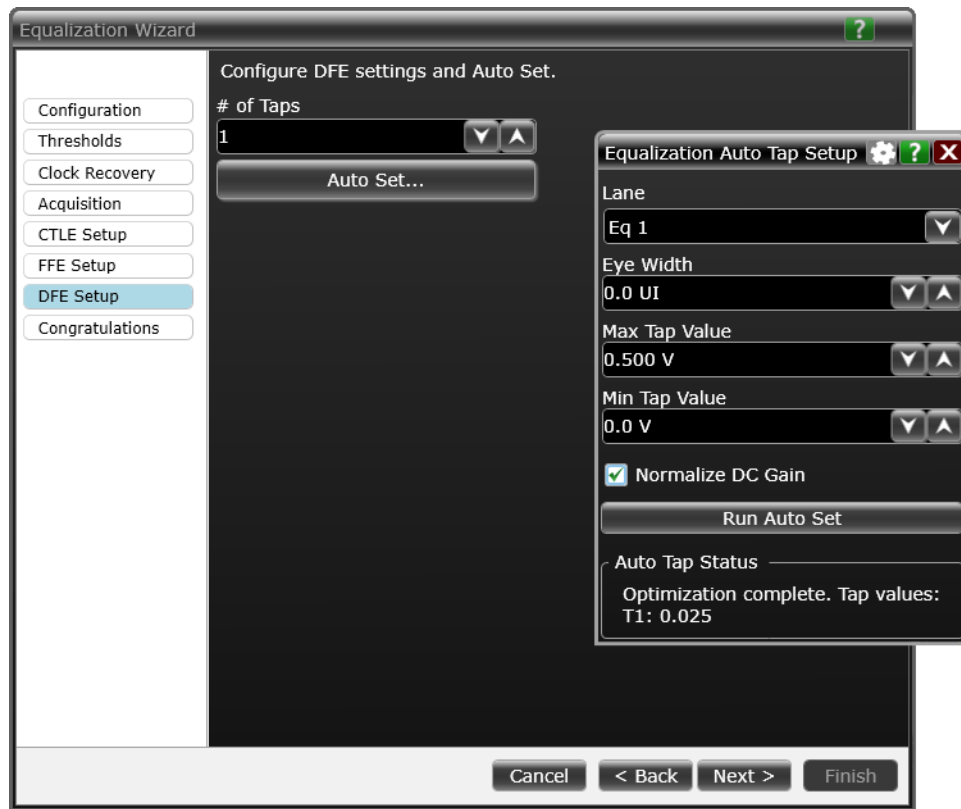
  c Then, click **Run Auto Set**.

  The optimizer runs and reports the tap values to you in the **Auto Tap Status** box.

  d Close the Equalization Auto Tap Setup dialog box to return to the Equalization Wizard.

  If you want to enter the tap values manually after completing the Equalization Wizard, go to the "Equalization Tap Setup Dialog Box" on page 71.

NOTE There is a situation where DFE cannot be enabled for a particular lane. See "Decision Feedback Eq (DFE) Tab" on page 64 for more information. The Equalization Wizard will place DFE in the next lane if necessary, or if you are setting up Lane 4, DFE is simply disabled.

Manually Setting Up Equalization

Manually entering the settings is one of the two ways to set up the Serial Data Equalization feature (the other way is to use the Equalization Wizard—see "Using the Equalization Wizard" on page 34).

The Equalization Wizard is the recommended setup method as it walks you through the process in a step-by-step fashion. Use the manual setup method only if you are comfortable with the manual controls and are familiar with equalization.

- "Setting Up Your Signal" on page 51
- "Equalization Setups" on page 52

Setting Up Your Signal

Before entering the Equalization dialog box, set up the channel, function, or waveform memory you want to equalize and turn on the real-time eye feature using the Real-Time Eye dialog box (**Analyze > Real-Time Eye...**). You will also need to set the sampling rate, memory depth, and timebase scale using the following recommendations.

1   **Sampling Rate** — (**Setup > Acquisition...**) This is generally set to the maximum possible value.

2   **Timebase Scale** — (**Setup > Horizontal...**) The timebase range (the width of the display) should be set to two-unit intervals to produce a nice display. Since the horizontal axis has 10 divisions, the timebase scale would then be calculated as:

Timebase scale = (2 x unit interval)/10

As an example, if your signal's data rate is 6.0 Gb/s, the unit interval would be 167 ps (1/data rate). Multiplying this by two and then dividing by ten would give you a timebase scale of 33.4 ps.

3   **Memory Depth** — (**Setup > Acquisition...**) If you are going to use Constant Frequency as your clock recovery method, you need to first determine how many samples per unit interval will be acquired based on the data rate of your signal and the sampling rate of your oscilloscope. As an example, assume your signal's data rate is 6 Gb/s (so your unit interval is [1/data rate] or 167 ps) and your sampling rate is 20 GSa/s. You could determine the samples per unit interval by multiplying your unit interval times your sampling rate and then converting units as in the following:

Samples/unit interval = (20 GSa/1 s) x (109 Sa/1 Gsa) x (1 s/1012 ps) x (167 ps/1 unit interval)

Samples/unit interval = 3.34 Sa/unit interval

Determine how many unit intervals you want to display and multiply the number you want times the samples per unit interval to calculate your needed memory depth. If you use a PLL or Fibre Channel clock recovery algorithm, the recommended memory depth is based on the following equation:
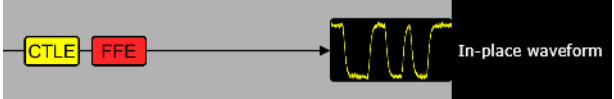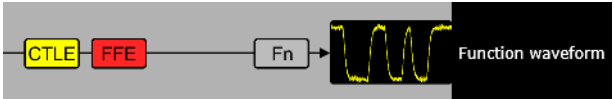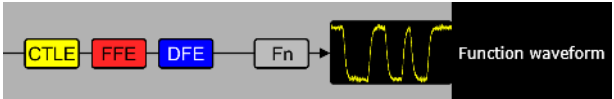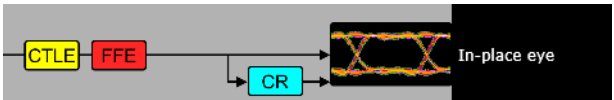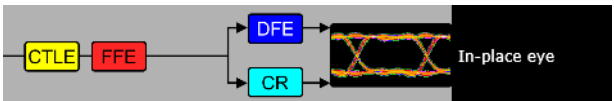
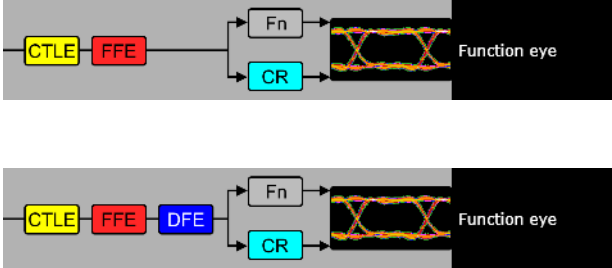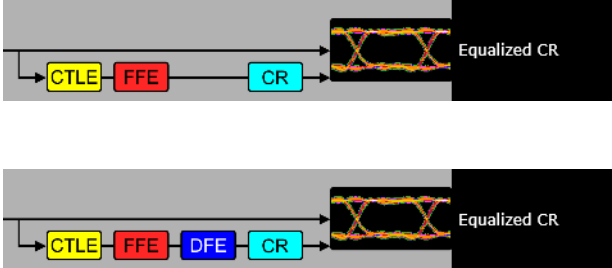Memory depth = (5/loop bandwidth) x (3/2)

where 5 is the number of time constants for PLL to settle and 3/2 is used so some data after the PLL settles is included.

4   **Thresholds** — (**Analyze > Real-Time Eye...** and then click the **Thresholds** button) In the Measurements Thresholds dialog box, set the **Thresholds** field to **Custom: Level +/- Hysteresis**. For most differential signals, set the Threshold Level to 0 V. You can also manually set the threshold level to whatever value you want or set it to 50% of the peak-to-peak voltage.

- **Set to 50% Vp-p** — Clicking this button automatically sets the logical threshold level to be (Vmax + Vmin)/2. The logical threshold level is defined to be the voltage level above which the signal is considered to be a logical 1 and below which the signal is considered to be a logical 0. Pressing this button also sets the hysteresis to be 5% of the source waveform's amplitude.

- **Hysteresis** — The hysteresis helps define what is considered an edge by the oscilloscope. For example, the default hysteresis is set to +/-5 mV. This means the signal must go through a 10 mV swing (-5 mV to 5 mV) to be considered an edge. If you make this value too large, the oscilloscope may skip edges and incompletely recover the clock. If you make this value too small, the oscilloscope will mistake noise for edges. In most cases +/-5 mV is a good setting, although you may need to increase or decrease this setting based on your specific signal and the amount of noise.

Equalization Setups

The following table shows possible equalization setups. The **How to Manually Set Controls to Obtain Setup** column assumes you are in the Equalization dialog box and are not using the Equalization Wizard. This column also covers only the initial setup variables. It does not cover parameters such as taps.

Graphical Representation and Description of Setup	How to Manually Set Controls to Obtain Setup
In-place waveform Equalization is applied to the source waveform in-place, but no real-time eye is displayed. The equalized source waveform is displayed. When **DFE** is selected, the **In-place waveform** configuration is not available.	**1** In the Real-Time Eye / Serial Data Analysis dialog box (**Analyze > Real-Time Eye...**), clear the **Real-Time Eye** check box. **2** Click the **Clock Recovery** button and choose your clock recovery method. **3** In the Equalization dialog box (**Analyze > Equalization...**), select **Equalize in place**.
Function waveform Equalization is applied to the source waveform and displayed as a separate function waveform, but no real-time eye is displayed. The equalized function waveform is displayed.	**1** In the Real-Time Eye / Serial Data Analysis dialog box (**Analyze > Real-Time Eye...**), clear the **Real-Time Eye** check box. **2** Click the **Clock Recovery** button and choose your clock recovery method. **3** In the Equalization dialog box (**Analyze > Equalization...**), select **Display as function**.
In-place eye Equalization is applied to the source waveform in-place, and clock recovery on the equalized waveform is used to display a real-time eye of the equalized source waveform.	**1** In the Real-Time Eye / Serial Data Analysis dialog box (**Analyze > Real-Time Eye...**), select the **Real-Time Eye** check box. **2** Click the **Clock Recovery** button and choose your clock recovery method. **3** In the Equalization dialog box (**Analyze > Equalization...**), select **Equalize in place**.

| Graphical Representation and Description of Setup	How to Manually Set Controls to Obtain Setup
Function eye Equalization is applied to the source waveform and displayed as a separate function waveform. Clock recovery on the equalized waveform is used to display a real-time eye of the equalized function waveform.	1 In the Real-Time Eye / Serial Data Analysis dialog box (**Analyze > Real-Time Eye...**), select the **Real-Time Eye** check box. 2 Click the **Clock Recovery** button and choose your clock recovery method. 3 In the Equalization dialog box (**Analyze > Equalization...**), select **Display as function**.
Equalized CR Equalization is used only to recover the clock, and the recovered clock is used to display a real-time eye of the original (unequalized) waveform. Clock recovery cannot be performed on a waveform if there is no open eye. So, in this case, equalization is used to create an open eye on a waveform so that clock recovery (and real-time eye analysis) can be performed.	1 In the Real-Time Eye / Serial Data Analysis dialog box (**Analyze > Real-Time Eye...**), select the **Real-Time Eye** check box. 2 Click the **Clock Recovery** button and choose one of the **Equalized ... PLL** clock recovery methods and select one of the equalization lanes as the **Clock Source**. 3 In the Equalization dialog box (**Analyze > Equalization...**) for the equalization lane you are using, set up the equalization; however, you do not need to select the **On** check box.

Equalization Dialog Box

The Equalization dialog box is where you launch the Equalization setup wizard (see "Using the Equalization Wizard" on page 34) or manually enter your equalization settings (see "Manually Setting Up Equalization" on page 51).

For background information on equalization settings, see:

- "Equalization Background / Theory" on page 6
- "Feed-Forward Equalization (FFE)" on page 12
- "Continuous Time Linear Equalization (CTLE)" on page 18
- "Decision Feedback Equalization (DFE)" on page 19

Controls
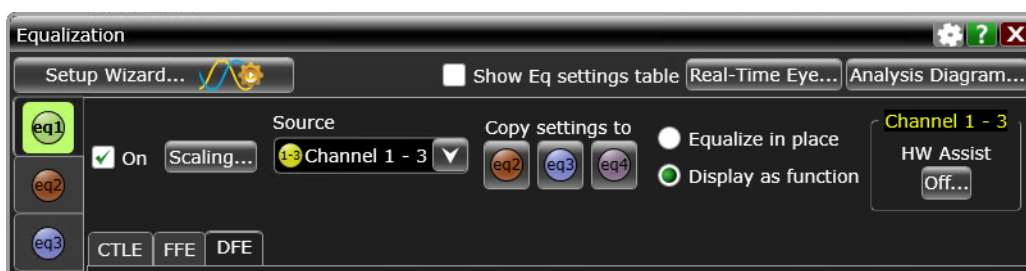
- **Setup Wizard...** — Opens the Equalization Wizard that walks you through the equalization set up process in a simplified, step-by-step fashion. See "Using the Equalization Wizard" on page 34.
- **Show Eq settings table** — Enables or disables the display of equalization settings in the Results pane at the bottom of the Infiniium oscilloscope window.
- **Real-Time Eye...** — Opens the Real-Time Eye / Serial Data Analysis dialog box where you can specify threshold voltage settings, specify clock recovery settings, and set up and enable real-time eye diagram displays.
- **Analysis Diagram...** — Opens the Analysis Diagram dialog box that shows the flow of input channel source signals through the various types of processing and analysis.
- **On** — Enables or disables equalization on the lane. The different types of equalization available are individually enabled or disabled in the CTLE, FFE, and DFE tabs.
- **Scaling...** — Opens the "Equalization Scaling Dialog Box" on page 66 where you can specify the vertical scaling options for the equalized signal waveform.
- **Source** — Lets you select the input signal source. You can choose between channel inputs, functions, or waveform memories.

> **NOTE**   Be sure you choose the correct signal source. It is also recommended that only one signal be displayed prior to running the equalization as it is difficult to discern one signal from another with real-time eye diagrams.

- **Copy settings to** — Lets you copy the equalization lane settings to another equalization lane. For more information, see "Multiple "Lanes" of Equalization" on page 10.

- **Equalize in place** or **Display as function** — Lets you choose to modify the waveform itself (**Equalize in place**) or create a separate equalized waveform (**Display as function**). In Infiniium oscilloscope software versions prior to 6.40, **Display as function** was used.

  For linear equalization (CTLE and FFE), "in-place" means the equalization runs completely in hardware, greatly improving speed. For DFE, "in-place" means the pre-6.40 version of DFE is performed (that is, the DFE that modifies the display of the real-time eye only).

  For more information, see "Multiple "Lanes" of Equalization" on page 10.

- **HW Assist** — This button can show:

  - **Off...** — Hardware-accelerated equalization is off and processing is done in software only. Click this button to open the Hardware Acceleration dialog box where you can resolve the issues preventing hardware acceleration to ultimately enable it.

  - **N/A** — Hardware-accelerated equalization is not available for this type of waveform.

  - **On...** — Hardware-accelerated equalization is on. Click this button to open the Hardware Acceleration dialog box.

For descriptions of the controls that appear in the CTLE, FFE, and DFE tabs of the Equalization dialog box, see:

- "Linear Eq, CTLE Tab" on page 57
- "Linear Eq, FFE Tab" on page 60
- "Decision Feedback Eq (DFE) Tab" on page 64

Accessing   To access the Equalization dialog box, choose **Analyze > Equalization...** from the main menu.

Linear Eq, CTLE Tab

- **Enable CTLE** — Enables the CTLE linear equalization in the equalization lane.
- **Nominal Data Rate** — Enter the data rate of your signal into this field.
- **Auto** — Automatically sets the optimal data rate.
- **Preset** — Lets you quickly set up the remaining options by selecting from a list of industry-standard equalization definitions.
- **# of Poles** — Lets you select the number of poles in the CTLE filter.
  - **2 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2}}{\omega_Z} \cdot \frac{s + \omega_Z}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right)}$$

· **3 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{p_3}}{\omega_Z} \cdot \frac{s + \omega_Z}{(s + \omega_{p_1}) \cdot (s + \omega_{p_2}) \cdot (s + \omega_{p_3})}$$

· **4 Pole, 1 Zero** — The following equation or graph is shown:

$$H(s) = \frac{\omega_{p_1} \cdot \omega_{p_3} \cdot \omega_{p_4}}{\omega_{Z_1}} \bullet \frac{s + \omega_{Z_1} \cdot s + \omega_{p_2} \cdot A_{dc}}{s + \omega_{p_1} \cdot s + \omega_{p_2} \cdot s + \omega_{p_3} \cdot s + \omega_{p_4}}$$

· **6 Pole, 2 Zeros** — The following equation or graph is shown:

$$\frac{\omega_{p_1} \cdot \omega_{p_3} \cdot \omega_{p_4} \cdot \omega_{p_5} \cdot \omega_{p_6}}{\omega_{Z_1} \cdot \omega_{Z_2}} \bullet \frac{s + \omega_{Z_1} \cdot s + \omega_{p_2} \cdot A_{dc} \cdot s + \omega_{Z_2}}{s + \omega_{p_1} \cdot s + \omega_{p_2} \cdot s + \omega_{p_3} \cdot s + \omega_{p_4} \cdot s + \omega_{p_5} \cdot s + \omega_{p_6}}$$
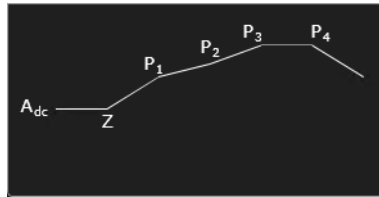

· **3 Pole, 2 Zeros** — The following equation or graph is shown:

$$H(s) = \frac{A_{dc} \cdot \omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{p_3}}{\omega_{Z_1} \cdot \omega_{Z_2}} \cdot \frac{\left(s + \omega_{Z_1}\right) \cdot \left(s + \omega_{Z_2}\right)}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right) \cdot \left(s + \omega_{p_3}\right)}$$

· **2 Pole, AC Gain** — (USB 3.1) The following equation or graph is shown:

$$H(s) = A_{ac} \cdot \omega_{p_2} \cdot \frac{s + \frac{A_{dc}}{A_{ac}} \cdot \omega_{p_1}}{\left(s + \omega_{p_1}\right) \cdot \left(s + \omega_{p_2}\right)}$$
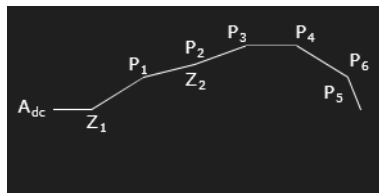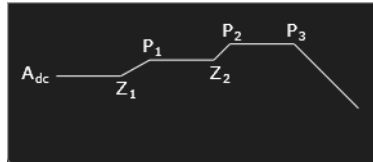

· **2 Pole, 2 Gain** — The following equation or graph is shown:

$$H(s) = \frac{(\omega_{p_1} \cdot \omega_{p_2} \cdot \omega_{LF}) \cdot \ s + \omega_Z \cdot 10^{\frac{A_{dc}}{20}} \ \cdot \ s + \omega_{LF} \cdot 10^{\frac{A_{dc2}}{20}}}{(\omega_Z \cdot \omega_{LF}) \cdot \ s + \omega_{p_1} \ \cdot \ s + \omega_{p_2} \ \cdot (s + \omega_{LF})}$$

· **Show Settings on Real-Time Eye** — Select this check box to display a pop-up box on the eye showing your equalization settings.
· The remaining fields in the dialog box let you enter values for the equation's variables:

- **DC Gain** (A_{dc})
- **DC Gain (dB)** (A_{dc} in dB)
- **DC Gain 2** (A_{dc2})
- **DC Gain 2 (dB)** (A_{dc2})
- **Zero Frequency** (ω_Z)
- **Zero 1 Frequency** (ω_{Z1})
- **Zero 2 Frequency** (ω_{Z2})
- **AC Gain** (A_{ac})
- **AC Gain (dB)** (A_{ac})
- **Pole 1 Frequency** (ω_{p1})
- **Pole 2 Frequency** (ω_{p2})
- **Pole 3 Frequency** (ω_{p3})
- **Pole 4 Frequency** (ω_{p4})
- **Pole 5 Frequency** (ω_{p5})
- **Pole 6 Frequency** (ω_{p6})
- **LF Frequency** (ω_{LF})

Linear Eq, FFE Tab

- **Enable FFE** — Enables the FFE linear equalization in the equalization lane.

- **Nominal Data Rate** — Enter the data rate of your signal into this field.
- **Auto** — Automatically sets the optimal data rate.
- **# of Taps** — Lets you set how many taps to use in the equalization. (For more information on how to determine the number of taps, see "Taps" on page 32.)
- **# of Precursor Taps** — Lets you specify how many precursor taps you want to use.

  Remember that with standard FFE, the current bit is equalized by using its tap value along with tap values associated with bits prior to it. Precursor taps are tap values associated with bits in front of the current bit.

  For example, if you set **# of Precursor Taps** to zero and you were using three taps, the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you set the **# of Precursor Taps** control to one then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used. This gives you a chance to specify which bits surrounding the "bit of interest" are used in the equalization algorithm.

  The **# of Taps** is the total number you want to use and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.

- **Auto Set...** — Opens the "Equalization Auto Tap Setup Dialog Box" on page 67 where you can have the oscilloscope determine tap values for you through optimization. See also: "What Tap Values Should You Use?" on page 32.
- **Tap Setup...** — Click this button to open the "Equalization Tap Setup Dialog Box" on page 71 where you can:
  - Choose to have the oscilloscope determine the tap values using optimization.
  - Manually enter tap values.
- **Tap Delay** — Usually when you set the number of taps, it will be at a rate of 1 tap/bit. If you are indeed using 1 tap/bit, the tap delay is defined as (1/data rate), and you can set it to this by selecting the **Track Data Rate** choice for this field.

  However, you can also set up the equalization such that multiple taps/bit are used. To do this, first set the **Tap Delay** field to **Manual**. Then compute how large to make the tap delay via the following equation:

  tap delay = 1/[data rate X (# of taps per bit)]

- **Linear Bandwidth** — See "Bandwidth Limit Control" on page 61.

See Also
- "Feed-Forward Equalization (FFE)" on page 12

Bandwidth Limit Control

To understand the **Bandwidth** limit control for equalization, we must first discuss some aspects of equalization.

As discussed in "Feed-Forward Equalization (FFE)" on page 12, one way to think of equalization is in terms of the frequency response of the channel your signal travels through. Ideally, you would like to transmit a signal through a channel whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented.

In practice, your frequency response plot may look something like the simplified plot below where the signal attenuates at higher frequencies. This is what causes ISI and makes the eye diagram close.
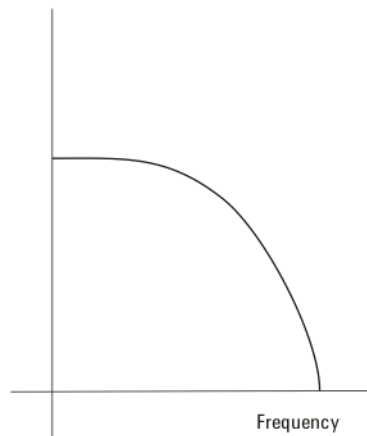
Frequency

To correct for this attenuation, we can apply a peaking filter response (dashed line in plot below) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).
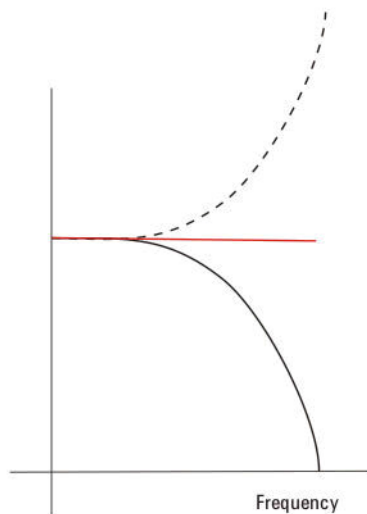
Frequency

Figuring out what this response should be and applying it is what you are doing when you perform equalization.

Now, in the above example, the unequalized response plot was very simplistic. A more realistic frequency response plot might look more like the following:
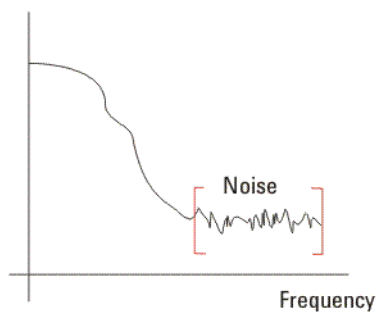
Noise

Frequency

Notice that after a certain frequency (or bandwidth) there is very little data and mostly noise. If you apply a response via equalization as discussed above, you will end up amplifying the noise as well as the data.

The **Bandwidth** control lets you minimize this noise gain by rolling off the response above the frequency where there is mostly just noise.

There are three options for the **Bandwidth** control:

- **Manual** — If you know at what frequency you want the response to roll off, you can enter it manually.
- **Track Source Bandwidth** (default) — This option simply generates a response up to the bandwidth of the source. For example, if your source is channel 1 and its bandwidth is set to 12 GHz, the response will go to 12 GHz before rolling off. This is the method that maximizes the noise gain.
- **Track Tap Delay** — This option forces the response to roll off at a typical ratio of bandwidth to data rate (0.75 x data rate). This will minimize the noise and is best used if you are unsure where in the frequency spectrum the noise is the dominant factor, but would like to minimize the noise gain.

See Also
- "Linear Eq, FFE Tab" on page 60
- "Step 7: FFE Setup" on page 47

Decision Feedback Eq (DFE) Tab
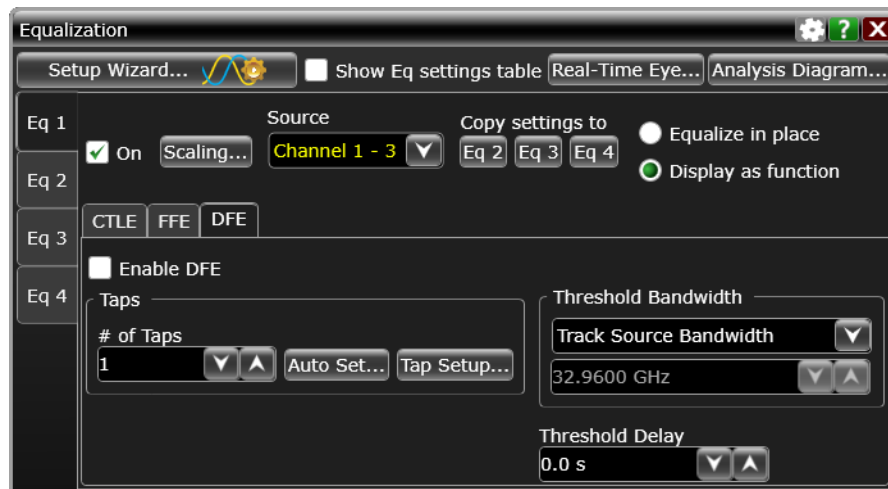
- **Enable DFE** — Enables or disables the DFE in the equalization lane.

  If a lane is equalized in-place, no DFE waveform is produced, and a real-time eye only DFE is applied. If a lane is displayed as a function, however, the a DFE waveform can be displayed. See "Multiple "Lanes" of Equalization" on page 10.

  If all of the following are true, DFE will be disabled in a lane, and cannot be turned on:
  - The lane is displayed as a function.

- CTLE and/or FFE is enabled in the lane.

- The lane's source is not using explicit or equalized clock recovery.

This is because DFE relies on the clock recovery (CDR) results of its source, but it cannot perform clock recovery in the middle of a lane (that is, after CTLE or FFE, but before DFE).

In this scenario, a warning icon appears in the DFE tab; clicking the icon opens a dialog box that explains the situation:
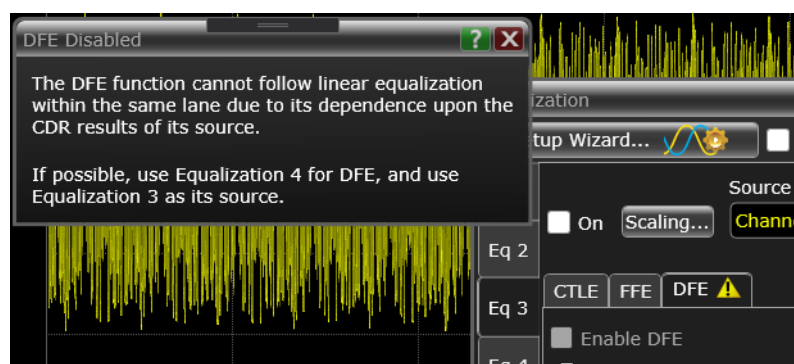
The Equalization Wizard allows setting up all three equalization types, but it places DFE in the next lane if necessary. If you are setting up Lane 4, DFE is simply disabled. The Equalization Wizard also display icons warning you about these situations.

- **# of Taps** — Lets you set how many taps to use in the equalization. (For more information on how to determine the number of taps, see "Taps" on page 32.)

- **Auto Set...** — Opens the "Equalization Auto Tap Setup Dialog Box" on page 67 where you can have the oscilloscope determine tap values for you through optimization. See also: "What Tap Values Should You Use?" on page 32.

- **Tap Setup...** — Click this button to open the "Equalization Tap Setup Dialog Box" on page 71 where you can:

  - Choose to have the oscilloscope determine the tap values using optimization.

  - Manually enter tap values.

- When **Display as function** is selected, the **Threshold Bandwidth** mode specifies whether a low-pass filter is applied to the DFE threshold waveform prior to summing it to the unequalized waveform.

  - If **Off** is selected, no low-pass filter is applied.

  - If **Track Source Bandwidth** is selected, the low-pass bandwidth 0.75 times the Nyquist bandwidth of the source.

  - If **Manual** is selected, you can specify the low-pass bandwidth.

- **Threshold Delay** — Sets a delay to move the decision threshold relative to the original waveform when creating the DFE equalized waveform.

    This delay is in contrast to the DFE Taps **Delay** value that moves the DFE equalized waveform relative to an explicit recovered clock in order to center the DFE eye post equalization (see "DFE Taps Tab" on page 73).

See Also   ·   "Decision Feedback Equalization (DFE)" on page 19

Equalization Scaling Dialog Box
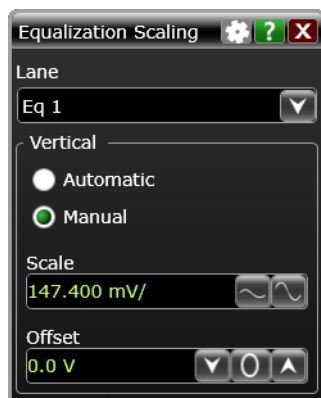
- **Lane** — Selects the equalization lane whose vertical scaling options are being set.
- **Vertical**:
    - **Automatic** — Automatically sets the vertical scale and offset of the equalized signal waveform.
    - **Manual** — Lets you set the vertical scale and offset of the equalized signal waveform.

Equalization Auto Tap Setup Dialog Box

Controls / 67

Accessing / 70

The Keysight-proprietary auto tap setup (optimization) algorithms for FFE and DFE are designed to produce an optimal eye opening. The values resulting from auto tap setup can be useful when initially evaluating transmission channels or other circuitry. However, when performing device interoperability testing, you will likely instead use the tap requirements specified by standards documents.

Controls

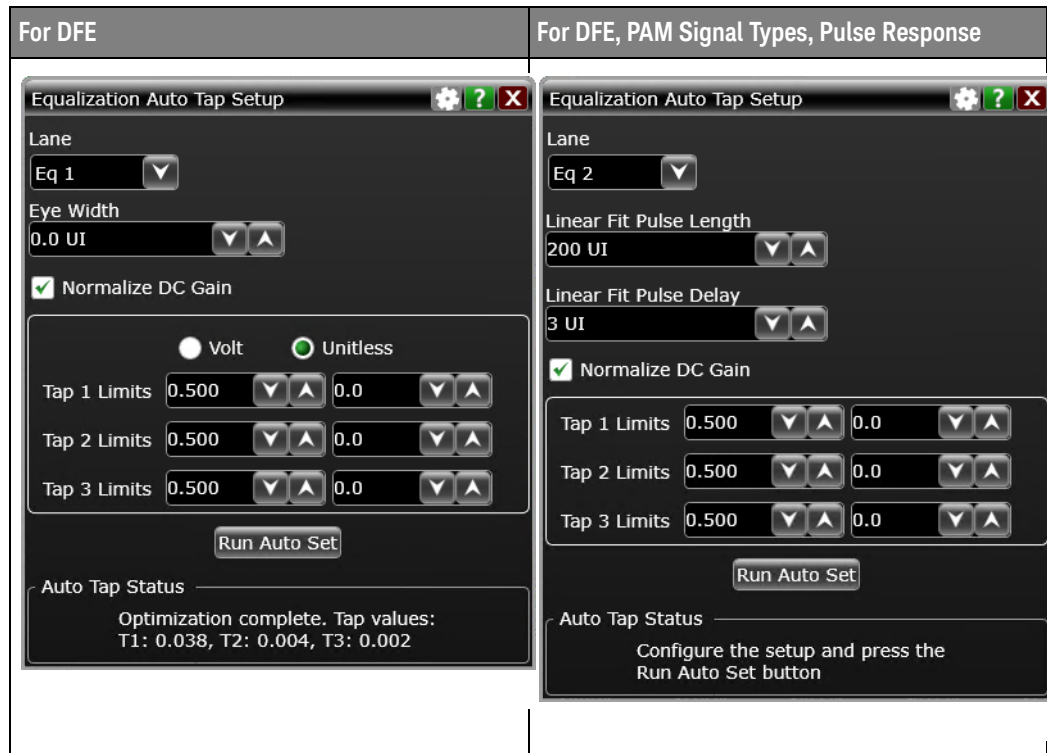| For DFE	For DFE, PAM Signal Types, Pulse Response
Equalization Auto Tap Setup ⚙ ? X Lane Eq 1 ▼ Eye Width 0.0 UI ▼ ▲ ☑ Normalize DC Gain ● Volt ● Unitless Tap 1 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Tap 2 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Tap 3 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Run Auto Set Auto Tap Status Optimization complete. Tap values: T1: 0.038, T2: 0.004, T3: 0.002	**Equalization Auto Tap Setup** ⚙ ? X Lane Eq 2 ▼ Linear Fit Pulse Length 200 UI ▼ ▲ Linear Fit Pulse Delay 3 UI ▼ ▲ ☑ Normalize DC Gain Tap 1 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Tap 2 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Tap 3 Limits 0.500 ▼ ▲ 0.0 ▼ ▲ Run Auto Set Auto Tap Status Configure the setup and press the Run Auto Set button

- **Lane** — Selects the equalization lane whose vertical scaling options are being set.

- When performing equalization on a PAM signal types (see the Signal Type Setup dialog box selection), you can choose the linear fit **Pulse Response** for the DFE tap optimization algorithm, and you can specify these filter parameters:

  - **Linear Fit Pulse Length** — This parameter is sometimes denoted using N_p.

  - **Linear Fit Pulse Delay** — This parameter is sometimes denoted using D_p.

  Tests that require a PRBS13Q or other specific signal pattern may suggest values for these filter parameters.

- **Eye Width** — Specifies the percentage of the unit interval, centered on the clock location, to optimize taps for.

  You can enter values from zero to one. For example, if the eye width is set to zero (0), taps are optimized for only the voltage specifically at the clock location of each unit interval. If the eye width is set to 0.2, taps are optimized for the voltage in 20% of the unit interval centered on the clock location. If the eye width is set to one (1), taps are optimized for voltage in the entire unit interval.

  - Typically with FFE, the receiver samples data not only at the location of the clock, but also in its vicinity. The default is set to 0.333 UI (unit interval) to account for this variance in the sample location. Typically values between 0.0 UI - 0.5 UI are the most useful.

- For DFE, true receivers sample data only at the location of the clock so a value of 0.0 UI (unit interval) is used as the default. If you want to include more of the eye, simply set this number to a higher value (0.0 UI – 0.5 UI would make sense for DFE).

  The jitter of the clock relative to the data will affect the eye width you should choose. If you have no jitter, you can choose an eye width of zero (0) because the data waveform will always be sampled at exactly the right time. If you want to accommodate some jitter, choose a higher eye width.

If you are unsure about what value to enter here, but do not want to use the default, think about whether you have more margin vertically or horizontally (where is your uncertainty?). If you optimize the tap values for a narrow portion of the unit interval, your eye will have a greater peak, but only near its center.

Narrow Eye
Width value
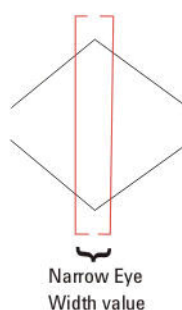
If you optimize the tap values for more of the unit interval, your eye will not be peaked as high, but will be more spread out horizontally.

Wide Eye Width
value
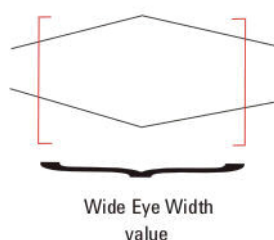
- **Normalize DC Gain** — Automatically scales the eye diagram so it is the same size as the transmitted eye.

  In the discussion on DFE (see "Decision Feedback Equalization (DFE)" on page 19), you can see that the eye drawn after DFE is applied is attenuated. To amplify the eye back to its original size (so you can directly compare the eye at the receiver to the eye at the transmitter) a gain factor needs to be applied.

- **Tap N Limits** — For each tap, you can enter upper and lower limits (maximum and minimum values) for the generated taps. You can specify limits in **Unitless** values or in **Volts**.

  Some serial data standards have limits on the tap values that can be used for DFE (you can look-up these limits in the standard specification documents for your technology).

<table>
<tr><td>**NOTE**</td><td>You can set negative tap values by entering a negative **Tap N Limit**.</td></tr>
</table>

- **Run Auto Set** — Runs the optimizer and reports the tap values to you in the **Auto Tap Status** box.

  The optimized taps are automatically inserted into the Equalization Tap Setup dialog box.

Accessing   To access the Equalization Auto Tap Setup dialog box:

  1 Choose **Analyze > Equalization...** from the main menu.

  2 In the Equalization dialog box FFE or DFE tab, click **Tap Setup...**.

  3 In the Equalization Tap Setup dialog box, click **Auto Set...**.

See Also

Equalization Tap Setup Dialog Box

Controls / 71

Accessing / 71

Controls

· **Lane** — Selects the equalization lane whose tap setup options are being set.

The remaining controls that appear in the Equalization Tap Setup dialog box depend on the selected tab:

· "FFE Taps Tab" on page 71

· "DFE Taps Tab" on page 73

· "Advanced Tab" on page 75

Accessing   To access the Equalization Tap Setup dialog box:

1   Choose **Analyze > Equalization...** from the main menu.

2   In the Equalization dialog box FFE or DFE tab, click **Tap Setup...**.

See Also   · "Taps" on page 32

FFE Taps Tab

- **Auto Set...** — Opens the "Equalization Auto Tap Setup Dialog Box" on page 67 where you can have the oscilloscope determine tap values for you through optimization. See also: "What Tap Values Should You Use?" on page 32.

- **Delay** — Equalization may cause your eye diagram to drift so the clock is no longer centered on the eye. The **Delay** field lets you specify this shift in order for the eye to be brought back to its original location relative to the unequalized signal.

- **# of Taps** — Adjusts the number of taps. See "How Many Taps Should You Use?" on page 32.

  The GUI lets you manually adjust up to 40 tap values.

- **# of Precursor Taps** — Lets you specify how many precursor taps you want to use.

  Remember that with standard FFE, the current bit is equalized by using its tap value along with tap values associated with bits prior to it. Precursor taps are tap values associated with bits in front of the current bit.

  For example, if you set **# of Precursor Taps** to zero and you were using three taps, the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you set the **# of Precursor Taps** control to one then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used. This gives you a chance to specify which bits surrounding the "bit of interest" are used in the equalization algorithm.

  The **# of Taps** is the total number you want to use and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.

- **Normalize** — This button makes the tap values sum to one for the output to have unity gain at DC. This lets you see the same size eye diagram at the receiver as you do at the transmitter.

  If you want to use this feature, first enter your tap values and then press the **Normalize** button. Your tap values will be adjusted so they sum to one.

- **Tap 0**, **Tap 1**, ... — Lets you manually enter FFE tap values.

- **Load Taps...**, **Save Taps...** — Lets you load (read) tap values from, or save tap values to, an .xml file.

- **Default Taps** — Sets the FFE tap values to their defaults.

See Also
- "Taps" on page 32
- "Feed-Forward Equalization (FFE)" on page 12

DFE Taps Tab

- When performing equalization on a PAM signal types (see the Signal Type Setup dialog box selection), you can choose the DFE tap optimization algorithm:

  - **Least Squared Error** — This is the traditional DFE tap optimization algorithm. This tap optimization is able to run on open eyes only.

  - **Pulse Response** — Because a specific signal pattern is required for this algorithm, the tap optimization is able to run with closed eyes.

    This DFE tap optimization algorithm uses a linear fit pulse response filter. When you automatically set tap values (see the following **Auto Set...** description), you are able to specify the linear fit pulse length and linear fit pulse delay parameters.

- **Auto Set...** — Opens the "Equalization Auto Tap Setup Dialog Box" on page 67 where you can have the oscilloscope determine tap values for you through optimization. See also: "What Tap Values Should You Use?" on page 32.

- **Upper Target**, **Lower Target** — These fields dictate the high and low logical values used in the DFE algorithm.

  For example, in DFE, when a bit is determined to be a logical high, its value will be equal to the Upper Target. Similarly, if a bit is determined to be a logical low, its value will be equal to the Lower Target.

- **Delay** — Equalization may cause your eye diagram to drift so the clock is no longer centered on the eye. The **Delay** field specifies a delay of the DFE equalized waveform relative to an explicit recovered clock in order to center the DFE eye, post equalization.

You can automatically set the tap delay to center the DFE eye by clicking the **Auto** button.

You can also affect the DFE eye location when creating the DFE equalized waveform by using the **Threshold Delay** field in the "Decision Feedback Eq (DFE) Tab" on page 64 to delay the DFE decision threshold.

- **Gain** — In the discussion on DFE theory/background (see "Decision Feedback Equalization (DFE)" on page 19), you can see that the eye drawn after DFE is applied is attenuated. To amplify the eye back to its original size (so you can directly compare the eye at the receiver to the eye at the transmitter) a gain factor needs to be applied. The **Gain** field lets you enter a constant that is applied to both the unequalized input signal and the corrected decision threshold value. Setting this field to 1 means there is no gain applied to the DFE eye diagram (like multiplying by 1).

- **# of Taps** — Adjusts the number of taps. See "How Many Taps Should You Use?" on page 32.

- **Tap 1**, **Tap 2**, ... — Lets you manually enter DFE tap values.

  For each DFE tap value, the entry field on the left is the unitless tap value. The field on the right shows the tap value in voltage.

  A red arrow next to a value indicates that the value was clamped according to the limits specified in the "Equalization Auto Tap Setup Dialog Box" on page 67.

  For information on how unitless tap values are converted to voltage values, see "How Do I Map DFE Voltage Tap Values to Unitless Tap Values?" on page 29.

NOTE   You can set negative DFE tap values by:

- a   Clicking **Auto Set...**.
- b   Entering a negative **Min Tap Value** in the Equalization Auto Tap Setup dialog box.
- c   Closing the Equalization Auto Tap Setup dialog box.
- d   Entering negative tap values (down to the specified **Min Tap Value**) in the **Tap 1**, **Tap 2**, ... fields.

- **Load Taps...**, **Save Taps...** — Lets you load (read) tap values from, or save tap values to, an .xml file.

- **Default Taps** — Sets the DFE tap values to their defaults.

See Also   · "Taps" on page 32

· "Decision Feedback Equalization (DFE)" on page 19

Advanced Tab

- **Seed DFE Taps** — Normally DFE will exclude the first several symbols from the real-time eye because they are used to seed the DFE. For example, if you are using a 4-tap DFE, the first four UIs are not included in the real-time eye.

  However, if you want to include all of the unit intervals in the real-time eye, you can enable **Seed DFE Taps**. When **Seed DFE Taps** is enabled, initial symbols are all seeded to the **Value** you enter so that the DFE can start working on the first UI.

  DFE tap seeding is useful for DDR buses that turn on and off frequently.

- **Load Taps...**, **Save Taps...** — Lets you load (read) tap values from, or save tap values to, an .xml file.

- **Default Taps** — Sets the DFE tap values to their defaults.

See Also   · "Taps" on page 32
            · "Decision Feedback Equalization (DFE)" on page 19

Index

A

acquisition setup, equalization wizard, 43
Advanced tab in the Equalization Tap Setup dialog box, 75
auto tap status, 33, 49, 50, 70

B

Bandwidth limit control, equalization, 61

C

chaining equalization lanes, 10
clock recovery, equalization wizard, 40
configuration, equalization wizard, 35
Constant Frequency clock recovery method, 41
Continuous Time Linear Equalization (CTLE), 36
Continuous Time Linear Equalization (CTLE), background, 18
copyright, 2
correction value, DFE, 20
crosstalk and Equalization, 6
CTLE (Continuous Time Linear Equalization), 36
CTLE setup, 36

D

data rate, equalization wizard, 44, 48
Decision Feedback Eq (DFE) tab in Equalization dialog box, 64
Decision Feedback Equalization (DFE), 49
Decision Feedback Equalization (DFE), background, 19
Decision Feedback Equalization (DFE), decision diagram, 23
Decision Feedback Equalization (DFE), frequency response plots, 24
Decision Feedback Equalization (DFE), how it acts on an eye diagram, 21
Decision Feedback Equalization (DFE), mathematical description, 20
DFE "Display as function" real-time eye, 27
DFE "Display as function" waveform, 26
DFE "Equalize in place" real-time eye, 27
DFE (Decision Feedback Equalization), 49

DFE and jitter analysis, 20
DFE real-time eye shape, 30
DFE real-time eyes, "Equalize in place" vs. "Display as function", 27
DFE real-time eyes, which should you use?, 28
DFE setup, equalization wizard, 49
DFE Taps tab in the Equalization Tap Setup dialog box, 73
DFE voltage tap values, 29
display as function equalization, 11, 20, 26

E

Equalization, 5
Equalization Auto Tap Setup dialog box, 67
Equalization background/theory, 6
Equalization dialog box, 55
equalization methods, 9
Equalization Scaling dialog box, 66
Equalization Tap Setup dialog box, 71
equalization wizard, 34
equalization, display as function, 11, 20, 26
equalization, in-place, 11, 19, 26
equalization, manual set up, 51, 52
equalization, manual signal set up, 51
equalization, multiple lanes of, 10
Equalization, problem it corrects, 6
Equalization, purpose of, 9
Equalized First Order PLL clock recovery method, 41
Equalized Second Order PLL clock recovery method, 41
Equalized Third Order PLL clock recovery method, 41
equalizer type, equalization wizard, 36
Explicit Clock clock recovery method, 41
Explicit First Order PLL clock recovery method, 41
Explicit Second Order PLL clock recovery method, 41
Explicit Third Order PLL clock recovery method, 41
eye width, auto tap setup, 68

F

Feed-Forward Equalization (FFE), 36
Feed-Forward Equalization (FFE), background, 12

Feed-Forward Equalization (FFE), decision diagram, 15
Feed-Forward Equalization (FFE), frequency response plots, 15
Feed-Forward Equalization (FFE), how it acts on a waveform, 13
Feed-Forward Equalization (FFE), mathematical description, 12
FFE (Feed-Forward Equalization), 36
FFE setup, 36
FFE Taps tab in the Equalization Tap Setup dialog box, 71
First Order PLL clock recovery method, 41

I

Infiniium Serial Data Equalization application, 7
in-place equalization, 11, 19, 26
Inter-Symbol Interference (ISI), 7, 17, 25
Inter-Symbol Interference (ISI) and Equalization, 6
ISI (Inter-Symbol Interference), 7

J

jitter analysis when DFE is enabled, 20

L

lane selection, equalization wizard, 34
lanes of equalization, copying, 10
lanes of equalization, legacy setups, 10
lanes of equalization, multiple, 10
leave source on, equalization wizard acquisition setup, 43
legacy equalization setups, 10
Linear Eq, CTLE tab in Equalization dialog box, 57
Linear Eq, FFE tab in Equalization dialog box, 60
logical decision threshold, correction value added by DFE, 20, 21
logical threshold level, DFE, 22

M

max/min tap value, DFE auto tap setup, 70