

Using Equalization Techniques on Your Infiniium 90000A Series Oscilloscope

Application Note

This application note provides an overview of feed-forward and decision feedback equalization. It also discusses the features present in the Infiniium serial data equalization software.

To perform equalization techniques with your Infiniium 90000A Series oscilloscope, you must have the N5461A Infiniium serial data equalization software installed on your scope. You can buy a 90000A Series oscilloscope with the software already installed by ordering Option DSO90000A-012 or you can buy a server license to the software (5435A-025).

The Infiniium serial data equalization application is dependent upon having the E2688A serial data analysis option installed on your oscilloscope as well.



Table of Contents

Introduction to Equalization	2
Feed-Forward Equalization	6
Decision Feedback Equalization	11
N5461A Infiniium Serial Data Equalization Software	18
Related Agilent Literature	27
Warranty Information (Remove All Doubt)	Back cover
Contact Agilent.....	Back cover



Agilent Technologies

Introduction to Equalization

The problem

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

A transmitter sends a serial 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 cause a partially or completely closed eye diagram that makes it impossible for the receiver to extract the clock/data. To recover a clock or data from the eye diagram, it must be re-opened. This is where equalization can help.

Look at Figure 1 below. You can see that a signal with 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, see page 4) distort the signal. All of these factors can cause the eye to close.

You can use equalization to re-open the eye by correcting for the ISI because ISI is deterministic (in other words, the ISI can be determined prior to transmission).

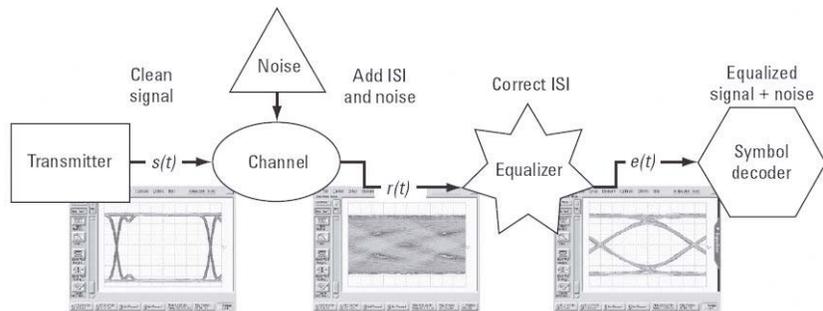


Figure 1. A transmitter sends a serial signal over a transmission channel. As it passes through the channel the signal is distorted.

Introduction to Equalization

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 two screen shots in Figure 2. They show a data signal before and after it traverses a backplane.

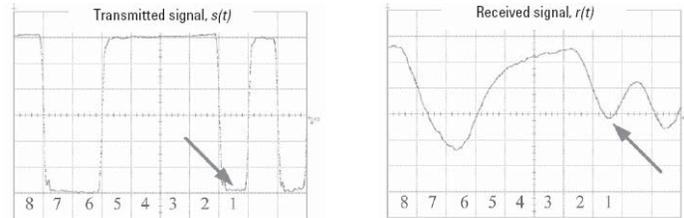


Figure 2. Serial data before it traverses a backplane (left) and after the traverse (right).

Looking at the five bits labelled **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 1. 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.

Introduction to Equalization

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 in Figure 3.

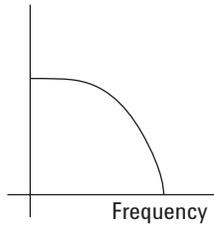


Figure 3. Simple example of a non-flat frequency response plot

You can see that at low frequencies, the frequency response is flat (like you want it to be), 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, 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, the pulses are somewhat rounded.

As this data sequence is sent through the channel, the medium it is travelling 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 (i.e. the design of the trace)
- The medium from which it is composed (i.e. the conductor or dielectric)

Each of these factors can be determined prior to signal transmission. Therefore, ISI is deterministic and you can correct for it with equalization techniques.

Introduction to Equalization

The solution: equalization

Now that the problem is clear, you can see how equalization is used as a solution.

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 (i.e., opens up the eye).

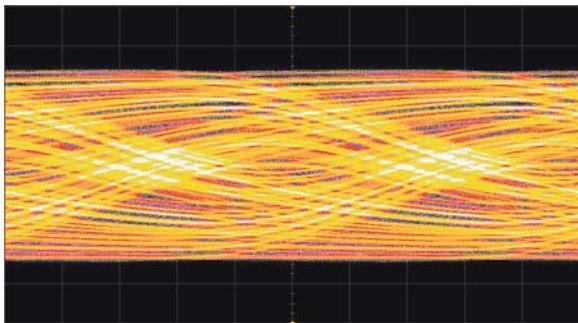


Figure 4. Unequalized 6-Gb/s eye diagram (notice the eye is completely closed)

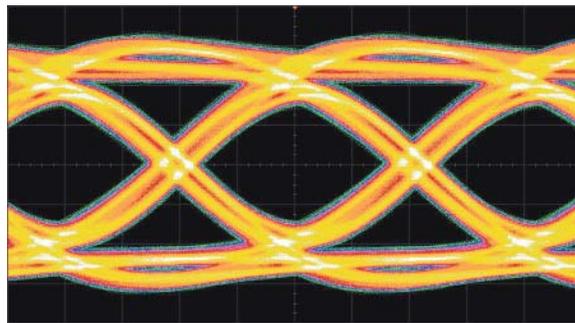


Figure 5. Equalized 6-Gb/s eye diagram (notice the eye is now open after equalization was applied)

There are two types of equalization methods used in the Infiniium 90000A Series oscilloscope: feed-forward equalization (FFE) and decision feedback equalization (DFE).

Feed-Forward Equalization

Feed-forward equalization (FFE) is an equalization technique that 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.

With FFE, 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 only concerned with correcting voltage levels in the waveform. You will see in the next section that this is in direct contrast to the way decision feedback equalization (DFE) works.

This section describes FFE four different ways. Each subsection contains the same basic content. You can choose the viewpoint that works best for you.

Mathematical description of FFE

For the purpose of this discussion, assume the FFE algorithm we are using has three taps. Taps are unitless correction factors applied to voltage levels in order 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:

$$e(t) = c_0 \cdot r(t - (0 \cdot T_d)) + c_1 \cdot r(t - (1 \cdot T_d)) + c_2 \cdot r(t - (2 \cdot T_d))$$

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, 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.

Feed-Forward Equalization

Description of FFE acting on a signal

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 one shown in Figure 6.

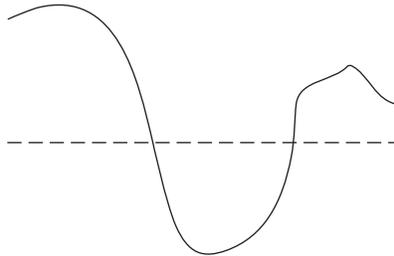


Figure 6. Unequalized waveform used for this example

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 for the purpose of this example.

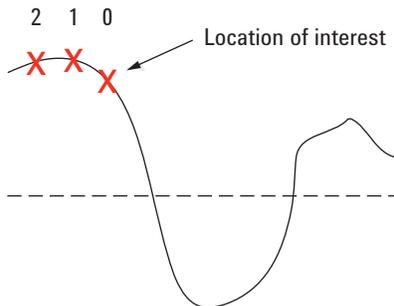


Figure 7. Location of interest and the two previous locations used in the equalization

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

Feed-Forward Equalization

First we need to know the tap values for the distortion present in our transmission channel. Determining these values will be discussed later. 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 multiplies it by the Tap0 value. Then the oscilloscope measures 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 sample rate to their next locations on the waveform.

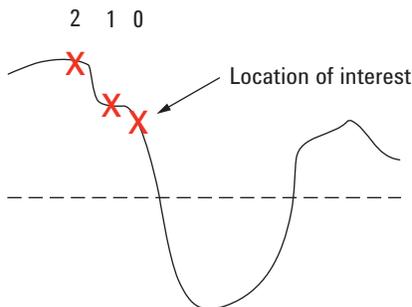


Figure 8. The new locations used for FFE

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 sample rate again to their new locations and repeat the entire process. This continues until the entire waveform has been traversed.

Feed-Forward Equalization

FFE decision diagram

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

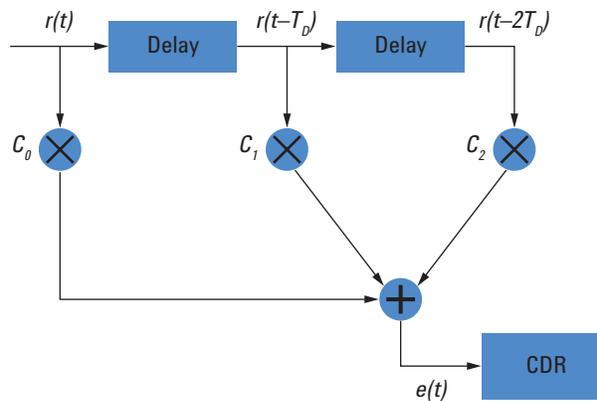


Figure 9.

where:

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

Feed-Forward Equalization

Understanding FFE through frequency response plots

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

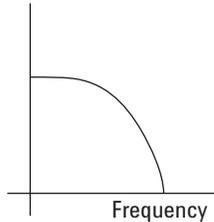


Figure 10. Simple example of a non-flat frequency response plot

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, as you saw in the discussion on ISI, the response sometimes attenuates the signal at higher frequencies (as shown in Figure 10).

To correct for this attenuation, we can apply a peaking filter via equalization. With FFE, we apply a filter response as shown in Figure 11 (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).

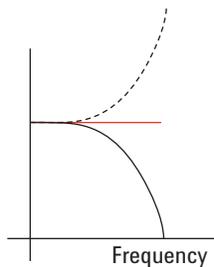


Figure 11. Plot showing the response generated by FFE (dashed) to correct for the non-flat frequency response of the channel (black)

Remember ISI is a result of the non-flat response of the channel. When we flatten the response, we are eliminating the ISI.

Decision Feedback Equalization

Decision feedback equalization (DFE) is an equalization technique that 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 section will discuss the DFE algorithm used in the Infiniium 90000A Series oscilloscope.

DFE will be described in four different ways. Each subsection will contain the same basic content. Choose the viewpoint that works best for you.

Before we look 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 upon this new equalized threshold level.

Decision Feedback Equalization

Mathematical description of DFE

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 in order 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 the correction value added to the decision threshold for a two-tap DFE:

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

where:

- $V(k)$ is the correction voltage added to the decision threshold used when determining the logic value of bit k
- $s(k-n)$ is the logic value of the data bit located n bits prior to bit k . The logic high value of $s(k-n)$ is equal to the upper target field in the Infiniium serial data equalization application GUI while the logic low value of $s(k-n)$ is equal to the lower target.
- c_n is the correction coefficient (tap) for the bit n bits prior to the bit of interest.

So, in order 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 $[(c_3 s(k-3) \text{ and } c_4 s(k-4))]$, 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 only uses previous bits.

Decision Feedback Equalization

Description of DFE acting on an eye diagram

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

First, look at the segment of a serial data signal in Figure 12.

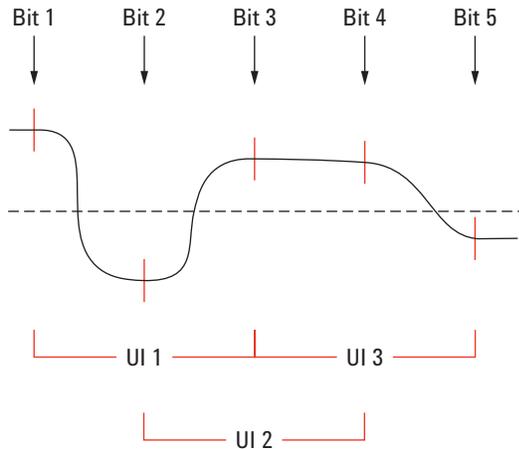


Figure 12. Example of a serial data stream

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's equalize the voltage for bit 4 using two taps to correct unit interval 3.

Decision Feedback Equalization

First, we need to know the tap values for the distortion present in our transmission channel. Determining these values is discussed in the next section. For now, assume these two tap values have been determined (call them Tap1 and Tap2 corresponding to c_1 and c_2 in the equation on page 12).

To get it started the DFE algorithm is first seeded with the correct bit values (high or low) for the first several bits. 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 shown in Figure 13.

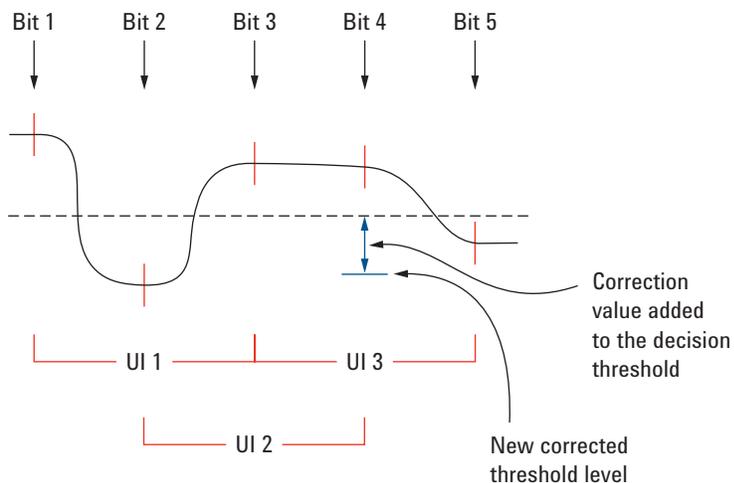


Figure 13. Blue arrow shows the new threshold level after DFE was applied to the fourth bit

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 90000A Series 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.

Decision Feedback Equalization

DFE decision diagram

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

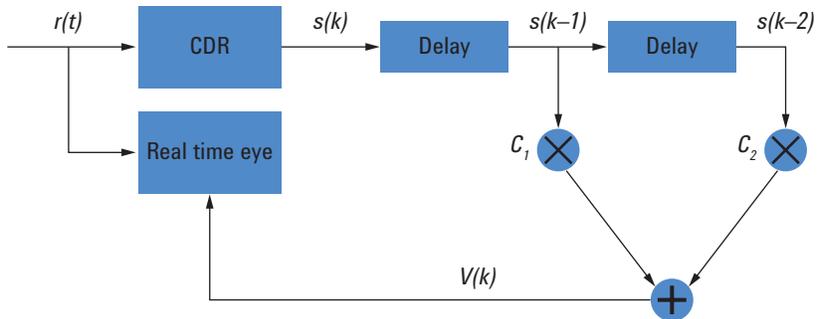


Figure 14. The DFE decision diagram

In this diagram:

- $r(t)$ 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 n th 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 the Infiniium oscilloscope actually 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.

Decision Feedback Equalization

Understanding DFE through frequency response plots

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

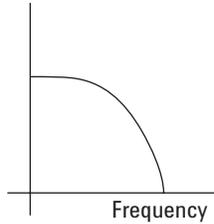


Figure 15. Simple example of a non-flat frequency response plot

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, as you saw in previously, the response sometimes attenuates the signal at higher frequencies (as shown in the plot in Figure 15).

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 in Figure 16).

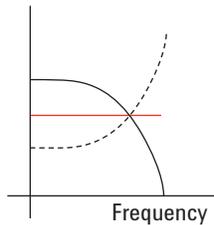


Figure 16. Plot showing the response generated by DFE (dashed) to correct for the non-flat frequency response of the channel (black)

Decision Feedback Equalization

Remember 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 (A) is needed, remember what the frequency response looks like for FFE.

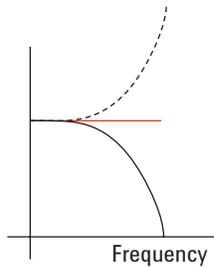


Figure 17. Plot showing the response generated by FFE (dashed) to correct for the non-flat frequency response of the channel (black)

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 (Figure 16). This means the flat response we obtain in the end will be shifted down as well. You can use the gain factor (A) 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.

N5461A Infiniium Serial Data Equalization Software

The Agilent N5461A Infiniium serial data equalization software enables you to perform both DFE and FFE on your 90000A Series oscilloscope. You can set up the equalization either manually or via the equalization wizard. This section will cover some of the features of this software.

For a more in-depth discussion on how to use the serial data equalization software, please refer to the *Agilent N5461A Infiniium Serial Data Equalization User's Guide* (N5461-97000) on www.agilent.com

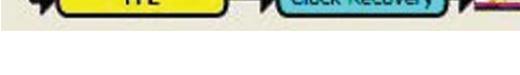
The equalization wizard

The equalization wizard is the preferred method for setting up equalization, as it walks you through the process step-by-step. It automatically adjusts controls based on your inputs so you do not have to search the oscilloscope GUI, but it still offers you the flexibility you need for your specific equalization case.

For example, the first step enables you to specify the exact combination of DFE/FFE you want to use. These options are graphically shown so you can easily view each choice. Table 1 shows the available selections.

N5461A Infiniium Serial Data Equalization Software

Table 1. Equalization wizard options.

Image from dialog box	Description
	<p>FFE is applied, but the real-time eye is not shown. Instead, the waveform resulting from applying FFE is displayed.</p>
	<p>No equalization is implemented and the unequalized real-time eye is displayed. This is useful if you want to have a reference to compare to. For instance, you can put the unequalized eye in the top grid and the equalized eye in the bottom grid so you can compare them on the same screen.</p>
	<p>FFE is only used to recover the clock, but the actual displayed real-time eye is unequalized. This is useful if you want to have a reference to compare to, but your unequalized eye is completely closed. You cannot recover a clock on a closed eye, so FFE is applied to open the eye so the clock can be recovered. In other words, this options serves the same purpose as the selection immediately above it, but is used with closed eyes.</p>
	<p>A feed-forward equalized eye is displayed on the oscilloscope. This is a standard FFE implementation and can be used on partially or completely closed eyes.</p>
	<p>A decision feedback equalized eye is displayed on the oscilloscope. This structure cannot be used on completely closed eyes as the clock cannot be recovered with this method. If the eye is closed and you want to implement DFE, use the option immediately below.</p>
	<p>FFE is used to recover the clock (but is not implemented in the real-time eye display) and a decision feedback equalized eye is displayed on the oscilloscope. This allows you to perform DFE on completely closed eyes.</p>
	<p>FFE is applied and then DFE is applied to the real-time eye. Additionally, since FFE is also used to recover the clock, this structure can be used on completely or partially closed eyes.</p>

N5461A Infiniium Serial Data Equalization Software

The equalization wizard also allows you to set up a reference configuration by splitting the display into an upper and lower grid. This enables you, for example, to display an unequalized eye in the upper grid and an equalized eye in the lower grid for comparison. You can also directly compare a DFE eye to an FFE eye.

The equalization wizard then gives you various choices for parameters associated with the clock recovery, thresholds, and acquisition. Each of these fields will be automatically defaulted to a specific value based upon your equalization setup, but you can also adjust these parameters if you do not want to use these defaults.

The clock recovery algorithms available in the equalization wizard include:

- First order PLL (phase lock loop)
- Second order PLL
- Equalized first order PLL
- Equalized second order PLL
- Constant frequency
- Explicit
- Explicit first order PLL
- Explicit second order PLL
- Fibre channel

Equalized first- and second-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.

The threshold controls include the threshold level, hysteresis, and a button to autoscale the vertical axis.

The acquisition controls include the timebase scale, sample rate, and memory depth.

Again, all of these fields will be determined automatically for you based on your equalization setup and signal. However, you can change them from these default values.

N5461A Infiniium Serial Data Equalization Software

Two additional parameters deal with FFE in particular:

Bandwidth

As discussed before, 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 plot in Figure 15 where the signal attenuates at higher frequencies. This is what causes ISI and makes the eye diagram close.

To correct for this attenuation, we can apply a peaking filter response (dashed line in Figure 16) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in Figure 16).

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

N5461A Infiniium Serial Data Equalization Software

In the above example, the unequalized response plot was very simplistic. A more realistic frequency response plot might look more like the plot in Figure 18.

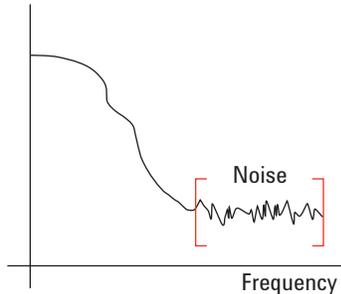


Figure 18. In a realistic frequency plot, you will see a lot of noise.

Notice 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** field allows you to minimize this noise gain by rolling off the response above the frequency where there is mostly just noise.

The three possible choices for the **Bandwidth** field are:

- **Track Scope BW** (default) – This option simply generates a response across the entire bandwidth of the oscilloscope. Therefore, if you are using a 12-GHz oscilloscope, the response will go to 12 GHz before rolling off. This is the method that maximizes the noise gain because the noise may be the dominant factor before the scope BW.
- **Track Tap Delay** – This forces the response to roll off at a typical ratio of bandwidth to data rate (0.75 x data rate). This choice 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.
- **Manual** – If you know at what frequency you want the response to roll off, you can enter it manually.

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:

$$\text{Tap delay} = \frac{1}{[(\text{data rate}) \cdot (\# \text{ of taps per bit})]}$$

N5461A Infiniium Serial Data Equalization Software

Taps

You can use up to 40 taps for each of your equalization methods (DFE and FFE), and there are two ways to determine what your tap values should be. The first way assumes you know the tap values and can then enter them manually. The second option is to have the oscilloscope optimize the tap values for you. The optimizer works differently for FFE and DFE, so each of these will be discussed separately.

Optimizing FFE Taps

The optimization algorithm needs to know the pattern being input to the oscilloscope in order to work. You can train the oscilloscope to a specific pattern signal and then use those tap values to equalize live traffic.

It is possible to train the tap values from a reference signal because the tap values are dependent only on the type of distortion, not the specific pattern. Therefore, you need to input a specific pattern signal with similar ISI characteristics to your live traffic signal. For example, a PRBS 2⁶-1 signal has five successive 1s and 0s similar to 8b/10b traffic.

It is best to use a short PRBS (pseudorandom binary sequence) as the reference signal for training. This is because:

- PRBS signals are rich and diverse in that they contain many different combinations of ones and zeros. The more combinations the optimizer “sees,” the better it is able to optimize the tap values. This is why CJT patterns do not work well as reference patterns.
- Using a short pattern is key because it does not require you to know the exact data rate. For instance, if you use a long pattern length and are off on the data rate, by the time the optimizer reaches the end of the pattern, the bits may not be in their correct locations.

The Equalizer Auto Tap Setup dialog box gives you several different PRBS patterns to choose from under the **Pattern** field. You can also set this field to **Custom** and then import a .ptrn file that describes the pattern you are using.

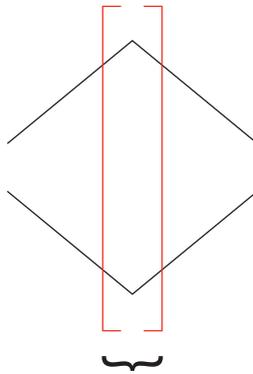


Figure 19. Equalizer Auto Tap Setup dialog box.

N5461A Infiniium Serial Data Equalization Software

The **Eye Width** field lets you choose how much of the eye to use in the optimization. This field ranges from zero to one. Entering zero means the eye is only optimized at the location of the clock. Entering one means the entire acquisition is used. 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 and 0.5 UI are the most useful.

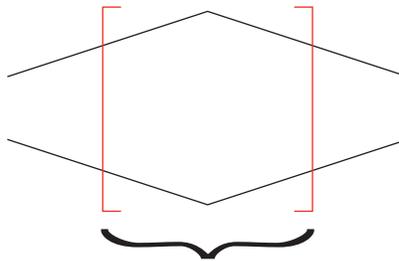
If you are unsure of the value to enter here, but do not want to use the default, think about whether you have more margin vertically or horizontally (where is your uncertainty?). If you use a narrow portion of the eye to optimize the tap values, your eye will have a greater peak, but only near its center.



Narrow eye width value

Figure 20. When a small eye width value is used for optimization, the eye diagram is greatly peaked only near its center.

If you use more of the eye to optimize the tap values, your eye will not be peaked as high, but will be more spread out horizontally.



Wide eye width value

Figure 21. When a larger eye width value is used for optimization, the peak is not as great, but the eye diagram is more spread out horizontally.

N5461A Infiniium Serial Data Equalization Software

Once you have these two fields set you can press **Run Auto Set** to start the optimizer. Once it is finished, it will report the tap values to you in the **Auto Tap Status** box.

The optimizer also looks for correlations between the input signal and the pattern you specify in the Pattern field. If it does not find a strong correlation, it will report a warning like the one circled in red in the screen shot in Figure 23.



Figure 22. If the optimizer does not find a strong correlation, it will issue a warning.

Notice the optimizer still generated tap values in case there was a weak correlation but everything was entered correctly. However, if you receive this warning, double check two things:

- Make sure your input pattern to the oscilloscope is the same pattern you specified in the **Pattern** field
- Make sure the data rate is set accurately

N5461A Infiniium Serial Data Equalization Software

Optimizing DFE taps

Some serial data standards have limits on the tap values that can be used for DFE (you can look up these limits in the standards for your technology). Therefore, this dialog box allows you to enter maximum/minimum values for the generated taps.

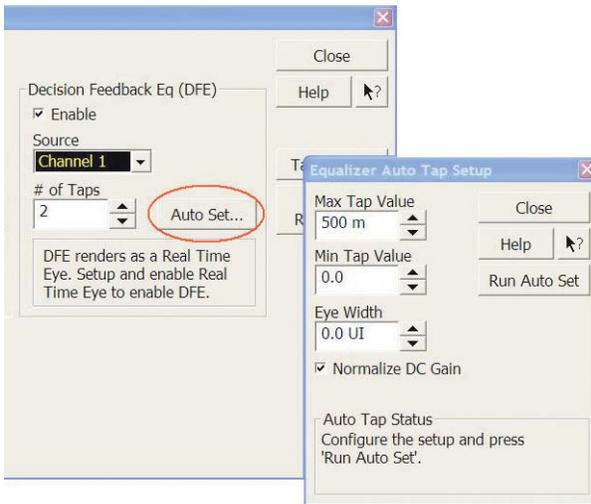


Figure 23. You can enter minimum and maximum tap values yourself.

An **Eye Width** field dictates how much of the eye to use in the optimization (see previous discussion in the *Optimizing FFE taps* section). For DFE, true receivers only sample data 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).

Remember in the discussion on DFE 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. Selecting the **Normalize DC Gain** check box automatically scales the eye diagram so it is the same size as the transmitted eye.

Once you have these two fields set, press **Run Auto Set**. The optimizer will run and report the tap values to you in the **Auto Tap Status** box.

N5461A Infiniium Serial Data Equalization Software

Determining how many taps to 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. 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.

Summary

This application note is meant to introduce you to equalization and provide an overview of the serial data equalization software. For a more complete look at using the software controls, please refer to the *Agilent N5461A Infiniium Serial Data Equalization User's Guide* (N5461-97000) on www.agilent.com

Related Agilent Literature

Publication title	Pub number
<i>Agilent Technologies N5461A Infiniium Serial Data Equalization Software for Infiniium 90000A Series Oscilloscopes Data Sheet</i>	5990-3330EN
<i>Agilent Technologies DSO/DSA 90000A Series Oscilloscopes Data Sheet</i>	5989-7819EN
<i>N5461A Infiniium Serial Data Equalization User's Guide</i>	N5461-97000

 **Agilent Email Updates**

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

 **Agilent Direct**

www.agilent.com/find/agilentdirect
Quickly choose and use your test equipment solutions with confidence.

Agilent
Open 

www.agilent.com/find/open
Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.

LXI

www.lxistandard.org
LXI is the LAN-based successor to GPIB, providing faster, more efficient connectivity. Agilent is a founding member of the LXI consortium.

Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements. For information regarding self maintenance of this product, please contact your Agilent office.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance, onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to:

www.agilent.com/find/removealldoubt

Product specifications and descriptions in this document subject to change without notice.

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Americas

Canada	(877) 894-4414
Latin America	305 269 7500
United States	(800) 829-4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Thailand	1 800 226 008

Europe & Middle East

Austria	01 36027 71571
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700*
	*0.125 €/minute
Germany	07031 464 6333
Ireland	1890 924 204
Israel	972-3-9288-504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
Switzerland	0800 80 53 53
United Kingdom	44 (0) 118 9276201

Other European Countries:

www.agilent.com/find/contactus

Revised: October 6, 2008

© Agilent Technologies, Inc. 2009
Printed in USA, February 17, 2009
5990-3585EN



Agilent Technologies