Acting on an Impulse: Equalization and Emphasis

Ransom Stephens, Ph.D.

April, 2008

Abstract:

The transmission path exposes a digital signal's vulnerable analog nature. Fortunately, equalization can recover its digital character. A channel's impulse response is encoded with the causes of eye-closure and gives us the clues we need to understand equalization. As usual in this business, when we make the transition from the theoretical to the realistic, trouble arises.

In Part 2 of this series (*The Case of the Closing Eye*) we saw that, at high data rates, the transmission path of a signal should be thought of as a complicated waveguide rather than as a simple trace on which digital signals propagate. Dispersion, the skin effect, multiple reflections, loss, and random noise all conspire to degrade the signal as it rumbles through the dielectric medium of a circuit board or cable. It's guided, but hardly confined by the conductor. These effects conspire to close the eye, but there is a conspicuous common feature of all but one of them. All but random noise are deterministic. If we understand a deterministic process, we should be able to compensate for it.

Techniques for fixing the problems caused by the transmission path or channel are called "equalization" and can be applied at the transmitter, receiver or both. They can also be continuous or discrete, linear or nonlinear, fixed or adaptive. In this paper, we'll figure out how to distinguish equalization techniques and, hopefully, gain an intuitive grasp of the electrodynamics upon which they are built.

The Dance of Conductors and Dielectrics

Pretend that you're a bit traveling through Flame Retardant Type 4 (FR-4) Printed Circuit Board (PCB), Figure 1. You have one hand on the conducting trace, like a mountain climber holding a rope, but your body is confined to the FR-4 dielectric. The medium, including both trace and dielectric, is slightly resistive, so you slowly lose power. The geometry of the PCB power and ground planes, as well as your trace and other traces on the board, form a convoluted capacitor and the whole system, what with all the other components and traces, experiences both self and mutual inductance. The result is a complicated nonlinear LRC filter. If that isn't bad enough, the trace you're hanging onto exhibits the skin effect, further attenuating your amplitude and filtering your high frequencies. The sharp edges that once defined you as a digital entity fade as you get farther from your transmitter. By the time you reach the receiver, your once sharp digital edges have spread into neighboring bits just as those neighboring bits have smeared into you.

At some point, your trace takes a sharp turn. Most of your energy makes the turn, but some is reflected back toward the transmitter. When it hits the transmitter, most is reflected a second time, interfering with some other bit way behind you.

Another thing can happen at a turn. Like a mountain climber reaching hand-over-hand on a rope, some of your energy skips the turn, exciting the conductor ahead. The effect is multi-path interference. Similarly, as your trace passes through the PCB, you're bound to see some other trace and, like a mountain climber grabbing another rope, some of your energy can excite that other trace – crosstalk.

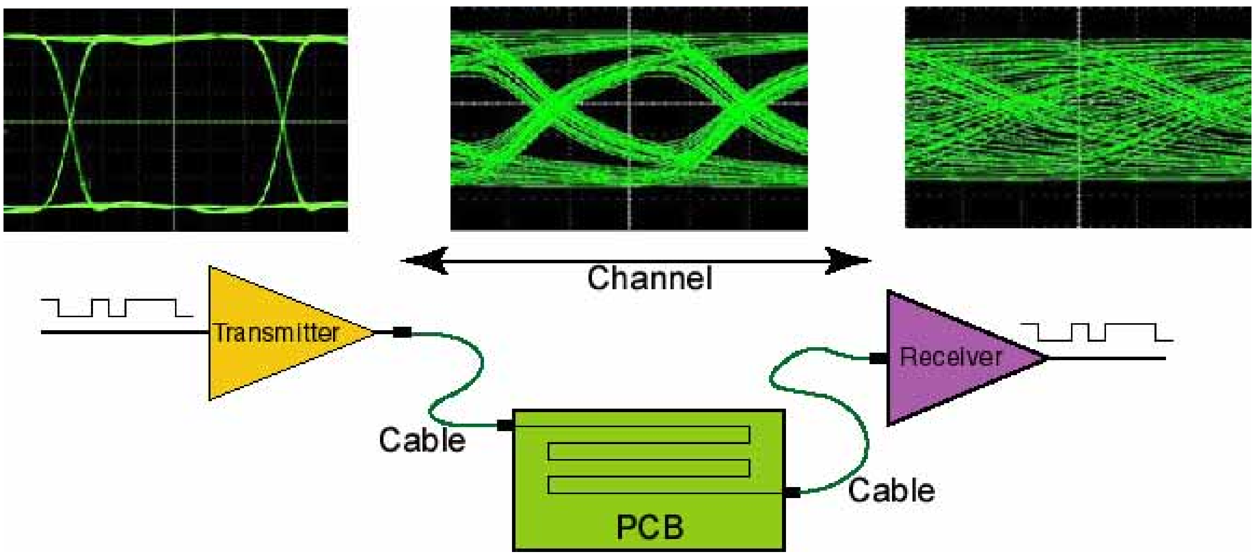

Figure 1: Serial data straw diagram.

Uh oh, up ahead, on your trace, a via! It's difficult to match the impedance of a trace at a via or some other junction like a connector. Some of your energy is reflected back like at the sharp turn, but worse – return loss and more multi-path interference.

When you finally get to the receiver, everything in the circuit has left its mark on you. You might be too beat up for the receiver to tell if you're a 1 or a 0, but all the bruises and scars tell a story.

The difference between your shape at the receiver and your shape right out of the transmitter encodes everything there is to know about where you've been. Everything.

Okay, not quite everything – there is random noise on your bit, almost all of which was generated by the transmitter. The effects of the transmission path, on the other hand, are deterministic. Attenuation, filtering, and multi-path interference cause one bit to affect the other bits of the same signal in different ways depending on the values of those bits. In other words, the fidelity of a given bit depends on the values of neighboring bits, which is why the eye-closing effects of the channel are called Inter-Symbol Interference (ISI). ISI causes both Data Dependent Noise (DDN) in voltage and Data Dependent Jitter (DDJ) in time.

The important point is that if you know (almost) everything about a system, only a fool would bet against you being able to fix it.

Impulse Response, Transfer Function and *S*-parameters

As it traverses the channel, the signal is encoded with a complete description of the signal-degrading properties of the circuit (except for random noise).

A special signal can be transmitted that is easy to decode. In 1822 Joseph Fourier (former aid to Napoleon Bonaparte and governor of Lower Egypt) showed that the time and frequency domains contain the same information and that a simple mathematical transformation carries us back and forth between them. A consequence of Fourier's theorem is that the narrower a pulse in the time-domain, the greater its frequency content. In the extreme, an infinitely narrow pulse has a perfect infinite bandwidth white frequency spectrum. (This is the part of Fourier's work that Werner Heisenberg reformulated as The Heisenberg Uncertainty Principle in 1927).

The *Impulse Response* shown in Figure 2 carries all the information about a transmission path in a form that is easy to decode.

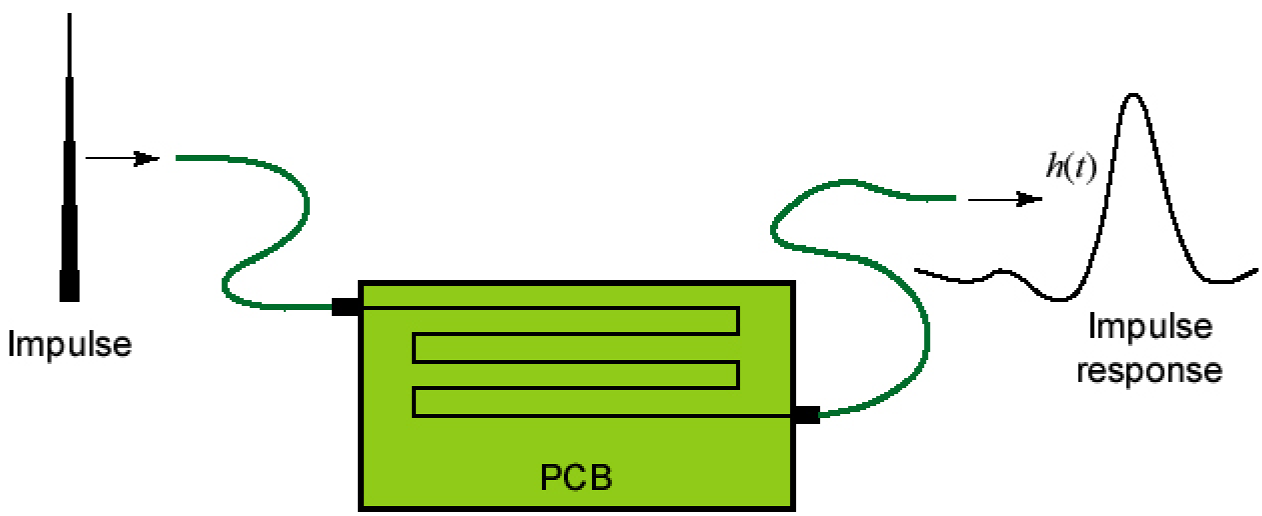

Figure 2: Impulse response, *h*(*t*).

Let's go back and think about a signal propagating through the circuit. A signal is made of a set of logic voltages that, for all practical purposes, is random. At any given instant, we can think of the circuit as being excited by many impulses, each multiplied by a coefficient so that their sum gives the observed signal.

If we know the impulse response, then we can calculate the output waveform by folding the impulse response over every piece of the transmitted signal. Mathematically, we say that the convolution of the impulse response, *h*(*t*), and the transmitted signal, *s*(*t*), gives the received signal, *r*(*t*).

$$r(t) = \int h(t-u)s(u)du \quad (1)$$

There are other ways to derive the impulse response. Since a sharp step is the integral of a narrow pulse, Figure 3, the derivative of the step response also gives the impulse response.

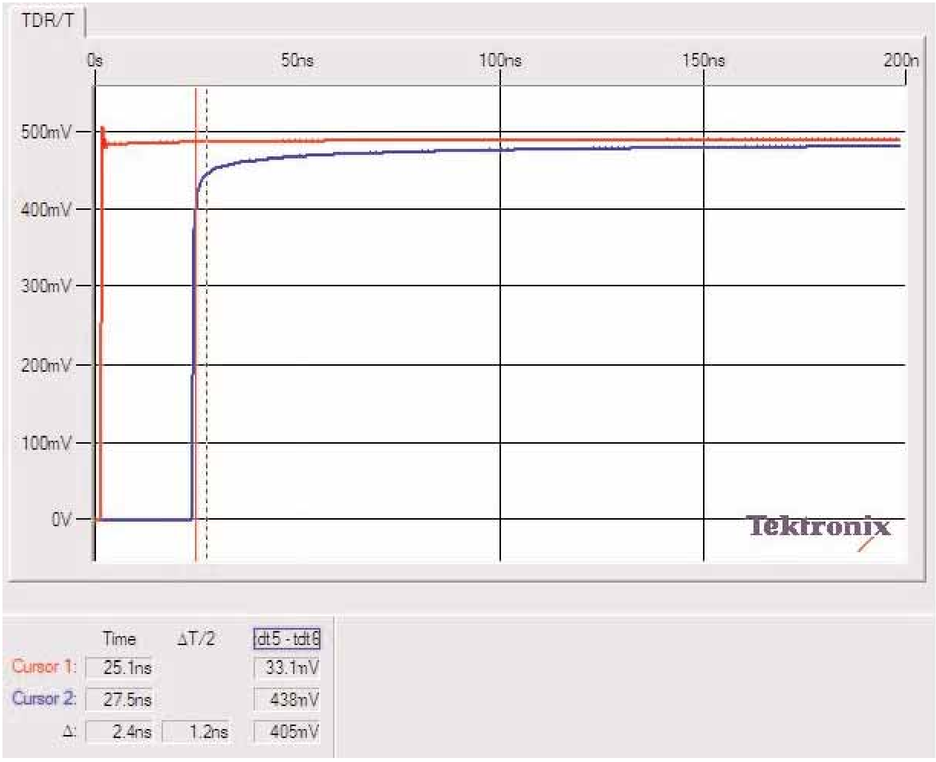

Figure 3: Step response.

Another common approach is to methodically analyze the responses of a series of single frequency signals. Each signal is sinusoidal and has the same amplitude. Their frequencies range from nearly DC to very high frequency (25 GHz would be fine for a 10 Gb/s link). The phase and magnitude of the response of each sinusoid is recorded. The result is all the information contained in the impulse response, but in the form of the scattering parameter, S_{21}. Since *S*-parameters have both phase and magnitude, they are complex quantities. $S_{21}(f)$ is related to the impulse response, *h*(*t*), by another mathematical transformation that was invented by another French mathematician, Pierre-Simon Laplace.

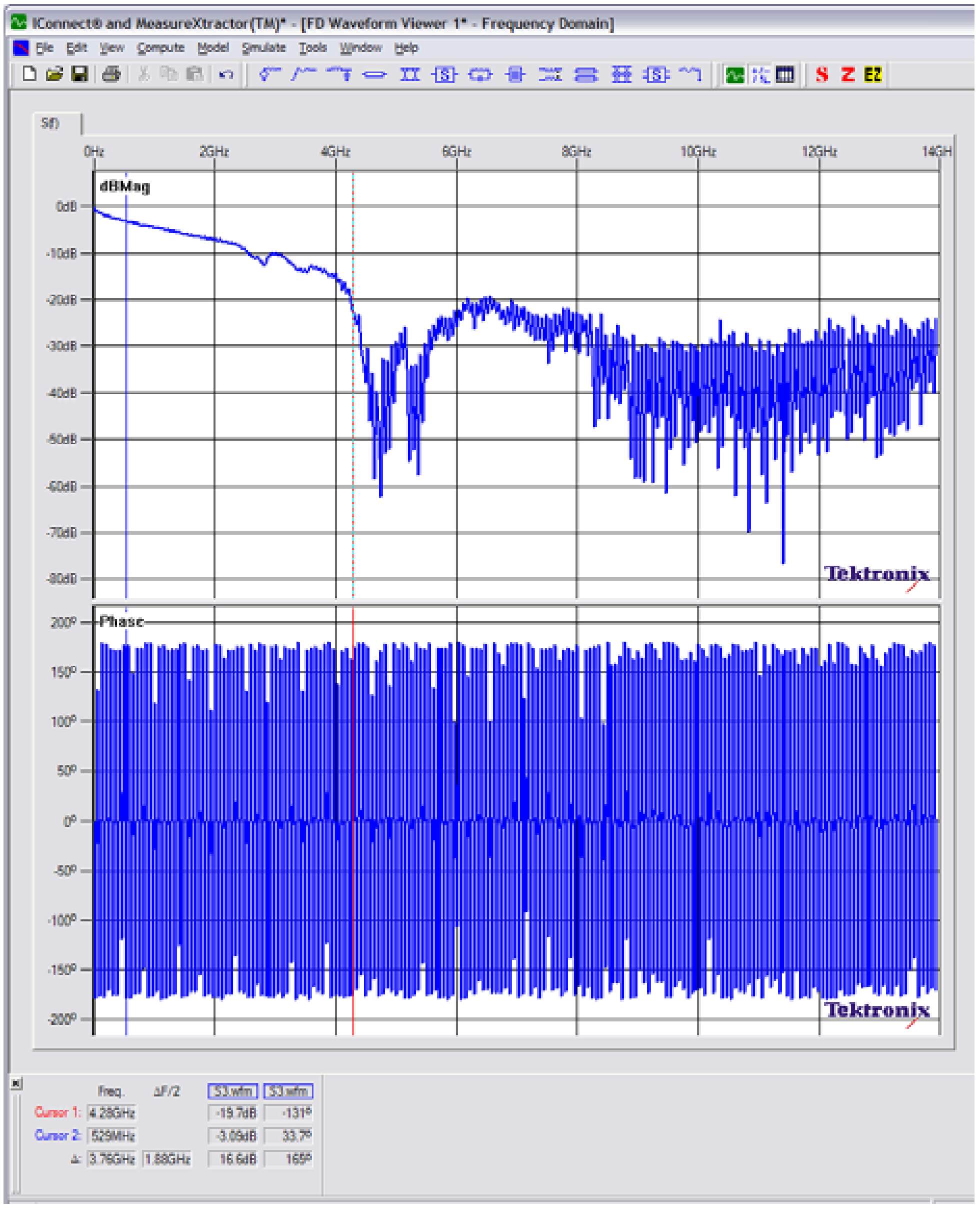

Figure 4: Frequency response of the channel whose step response is shown in Figure 3.

The *S*-parameter in frequency and phase for the step response of Figure 3 is given in Figure 4.

The *transfer function*, *G*(*s*), is given by the Laplace transform of the impulse response,

$$\begin{aligned} G(s) &= \mathrm{L}\left[h(t)\right] \\ &= \int_0^\infty e^{-st} h(t)\,dt \end{aligned} \qquad (2)$$

where *s* is the Laplace parameter. Since no information is either introduced or removed in the transform, it's okay to think of the impulse response, the transfer function, and the *S*-parameters as the same thing. After all, if you can measure one, you can calculate the others.

Linear Equalization

The obvious way to prevent the channel from degrading the signal is to use a less dispersive medium, carefully match impedances, and etch precisely consistent thin traces. Since this approach would make the design engineer's job straightforward it will never happen. Replacing FR-4 can increase cost by several hundred percent.

Instead, we stick with standard design techniques and implement clever technology at either the transmitter, receiver, or both. Equalization corrects a signal by amplifying the components of the signal that are suppressed by the transmission path.

To see how equalization works, we start in the tradition of engineers since they invented the wheel. First we do it, and then we ponder why it works.

Figure 5 shows several bits observed at the receiver overlaid on the same bits as they looked at the transmitter. Say we want to fix the bit labeled "0." We know the voltages of each bit that was received prior to bit 0. If we multiply the voltage levels of the preceding bits, r(*i*), by constants, k_i, and add them up, we should be able to improve the fidelity of bit 0.

$$e(0) = k_0 r(0) + k_1 r(1) + k_2 r(2) + k_3 r(3) + k_4 r(4) + k_5 r(5) + k_6 r(6) + k_7 r(7) . \qquad (3)$$

The voltages that we're using to fix the one bit are called *cursors* and the constants, k_i, are called *taps* because we're tapping into the cursors to fix the one bit.

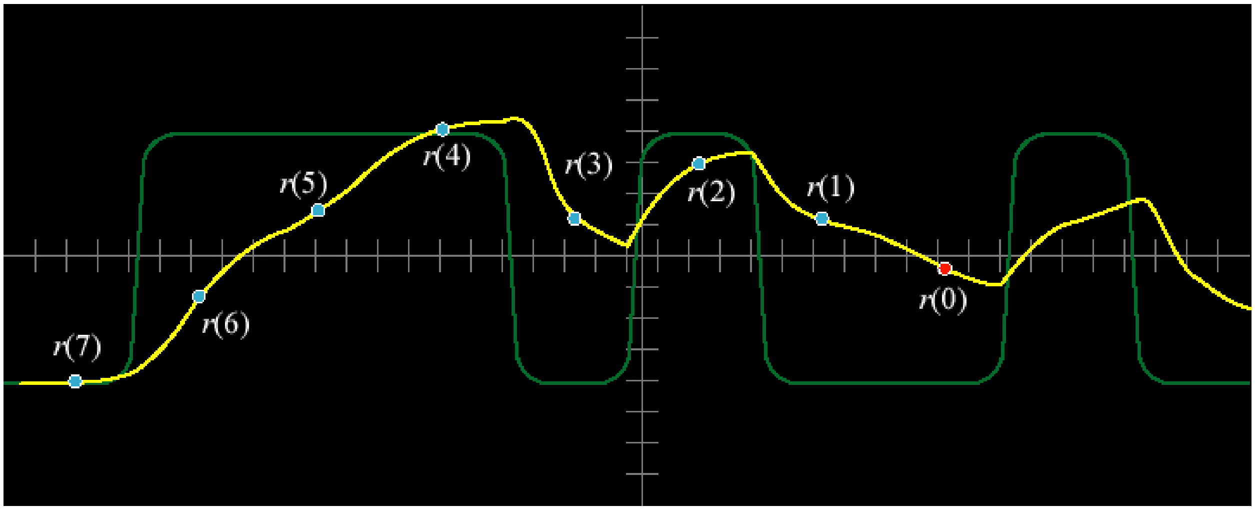

Figure 5: The ideal waveform of eight bits superimposed by the received waveform.

Of course, the trick is to pick constants (or taps) that improve not just bit 0 but can be applied to every bit in the signal and open the eye.

This is a good time to go back to the impulse response and transfer function. The signal at the receiver, *r*(*t*), is given by Eq. (1). If we operate on Eq. (1) with the Laplace transform, we get

$$R(s) = G(s)S(s) \tag{4}$$

where the Laplace transform of *r*(*t*) is *R*(*s*) and so forth. Since *S*(*s*) represents the transmitted signal, if we operate on Eq. (4) with the inverse of the transfer function then, up to the effects of random noise, we can extract the transmitted signal from the received signal,

$$G(s)^{-1}R(s) = G(s)^{-1}G(s)S(s) = S(s). \tag{5}$$

Now use the inverse Laplace transform to get back to the time domain

$$\mathrm{L}^{-1}[G(s)^{-1}R(s)] = s(t). \tag{6}$$

The inverse Laplace transform of a product is given by the convolution,

$$\mathrm{L}^{-1}[G(s)^{-1}R(s)] = \int g(u)r(t-u)du = s(t). \tag{7}$$

It is important at this point to remember that we are ignoring the effects of random noise.

By converting the integral in Eq. (7) to a sum, we recover the discrete form of the equalizer in Eq. (3) and can identify a good set of “taps” $\underline{k_i}$,

$$\begin{aligned} s(t) &= g(n) * r(n) = \sum_{i=0}^{N-1} g(i)r(n-i) \\ &= \sum_{i=0}^{N-1} k_i r(n-i) \end{aligned} \tag{8}$$

The discrete equalizer in Eqs. (3) and (8) is called a *Linear Feed-Forward Equalizer* (both LFE and FFE are common acronyms). The continuous form of this equalizer in Eq. (7) is called a Continuous Time Linear Equalizer (CTLE). We can think of the CTLE as an FFE with an infinite number of taps. In practice, since the receiver is designed with a bandwidth dictated by the data rate, in most cases the hardware can only support one tap per bit period. In principle, there’s no reason that correction terms should not be made of two, three, or more taps into each bit (or cursor).

The equalizers defined by Eq. (7) and (8) are called “feed forward” because the cursors that precede the bit being equalized are fed forward in time. In the language of Digital Signal Processing (DSP), Eq. (8) is also called a Finite Impulse Response (FIR) filter. Adding a delay of several bits would allow the use of cursors that follow the equalized bit too. Cursors that precede the equalized bit are called pre-cursors and those that follow, post-cursors.

Tap values can be derived from a simulation based on the measured *S*-parameters, but better taps can be determined by the Least Mean Square (LMS) technique. A transmitted waveform is simulated and the taps are varied until the sum of the differences of the transmitted and equalized waveforms is a minimum. Since ISI is caused mostly by neighboring bits, taps tend to decrease in magnitude for higher numbered cursors.

Since the sum in Eq. (8) has to be calculated at the full data rate, equalizers are typically implemented as hardware shift registers, Figure 6.

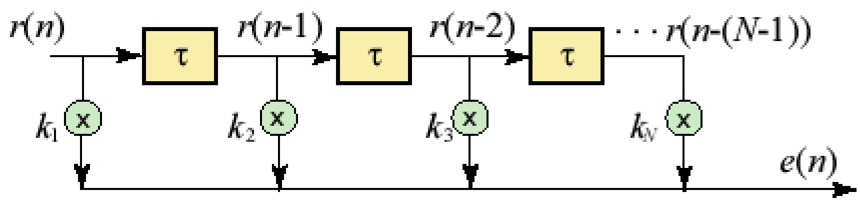

Figure 6: Linear Feed-Forward Equalizer (FFE) implemented in a shift register.

FFEs have two drawbacks. First, since the FFE is both discrete and limited in the number of taps, it can't remove all ISI. Second, a side effect of fixing the low pass nature of the channel, is that high frequency noise tends to be amplified by the FFE. Both of these problems can be addressed by introducing a nonlinear component.

Nonlinear and Adaptive Equalization

The Distributed Feedback Equalizer (DFE) is a common type of equalizer that supplements an FFE with a feedback loop. The output of the FFE is sent to a symbol decoder where logic levels are determined. The identified bits are delayed and then fed to a second shift register, the "feedback equalizer," which has its own taps, c_i. The output of the feedback equalizer is then combined with the output of the FFE, Figure 7.

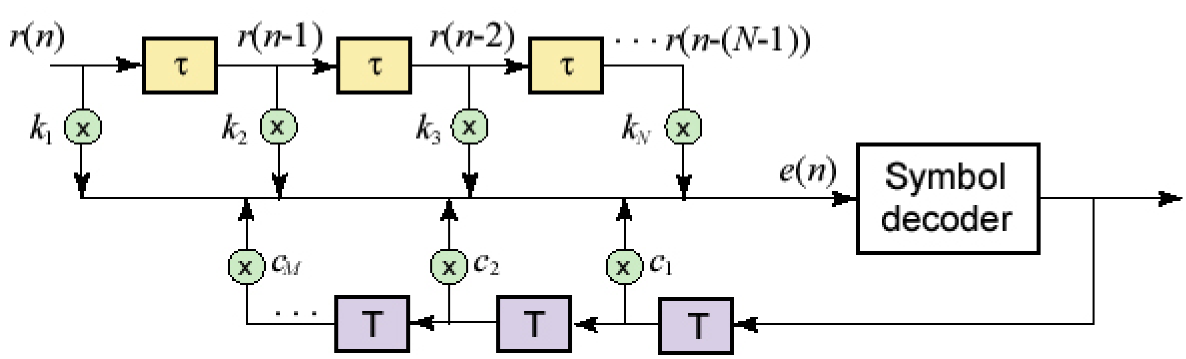

Figure 7: DFE block diagram

By determining the taps for both the linear and nonlinear components of the DFE at the same time, that is, by determining the FFE taps at the same time as the nonlinear feedback taps, the problem of amplified noise is reduced and more ISI is removed. The problem with a DFE is that, since the feedback loop operates on the digital output, when a bit is identified in error, an avalanche of subsequent errors can occur.

Both FFEs and DFEs are *fixed* equalizers. *Adaptive* equalizers can adjust on the fly to compensate for changes in the environment like temperature and humidity; electromagnetic interference, especially if it's just a few fixed frequencies that don't come and go too quickly; and even some components of random noise. Really clever adaptive equalizers use image reconstruction techniques like the maximum entropy method, neural networks and fuzzy logic. They are usually proprietary and they are always highly cool.

Equalization at the Transmitter

In principle we could transmit a signal that was appropriately contorted so that the effects of the channel would convert it to a nice digital-looking waveform. In practice, transmitter pre-emphasis emphasizes just the high frequency components of a signal to correct the low pass filtering effect of the channel. A pre-emphasized signal is one where a signal is given a voltage boost at every logic transition. We could call it a two tap transmitter equalizer.

Equalization techniques at the transmitter rarely employ more than three taps, but combining transmitter de-emphasis and receiver equalization can be very helpful in opening closed eyes.

Analyzing Closed Eyes

We started our Knowledge Series with receiver testing and here we are again. In stressed eye analysis, we measure the Bit Error Ratio (BER) of a receiver in response to the worst case, but compliant, signal. If the receiver fails the stressed eye test, or what is worse, passes the stressed eye test but fails in the system, then we need to analyze the behavior of the clock recovery circuit and the symbol decoder. To do so, we have to know what the signal looks like after equalization but before clock recovery, at the point marked by the star in Figure 8.

In most implementations of a standard, whether it's PCI Express generation 1 or 2, SATA, SAS, FibreChannel, Gigabit Ethernet, the specified clock recovery is modeled by a Phase Locked Loop (PLL), but the actual implementation is a nonlinear device, like a phase interpolator, whose behavior is more complicated. By probing the signal before the receiver, at the "+" in Figure 8, and applying a mathematical model of the equalization scheme, we can analyze the clock recovery and symbol decoder.

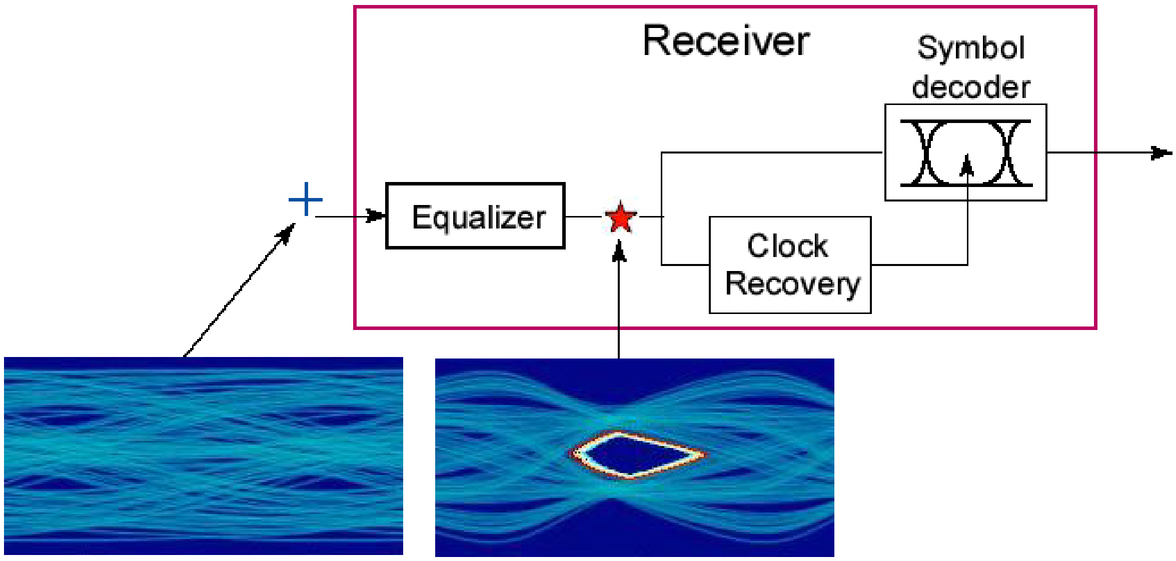

Figure 8: A receiver with equalizer and clock recovery.

The Tektronix DSA8200 equivalent time sampling oscilloscope equipped with 80SJNB Advanced software, Figure 9, provides virtual access to that point after the equalizer but before clock recovery in Figure 8. It has an FFE and a DFE built in so that you can analyze the equalizer output, Figure 10. In other words you can measure the jitter and noise that the clock recovery and slicer actually see.

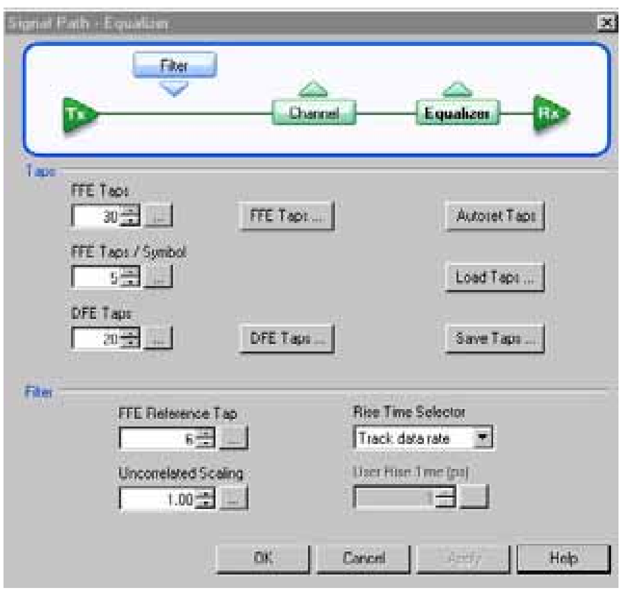

Figure 9: Tektronix 80SJNB Advanced analysis equalization menu.

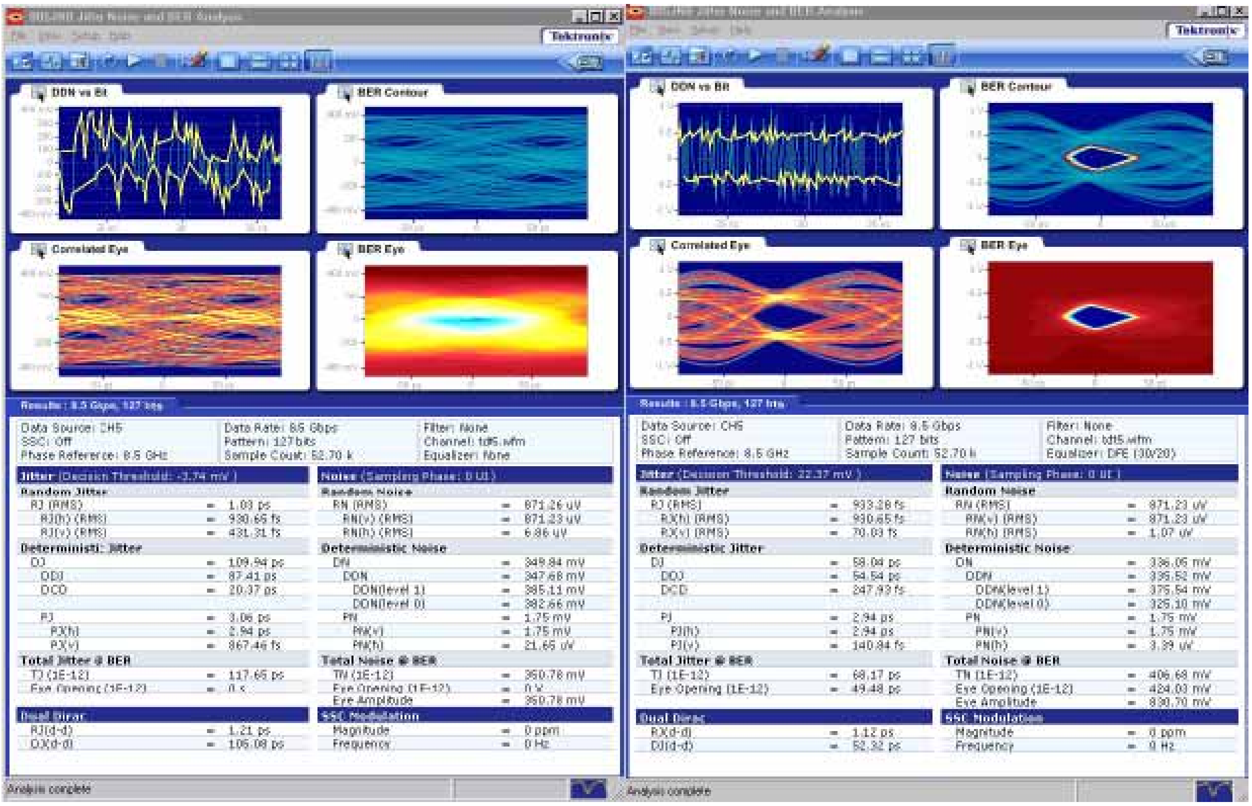

Figure 10: Tektronix 80SJNB Advanced analysis, closed eye results on the left, equalized on the right.

Conclusion

We've seen how the geometry and media of the path that a digital signal travels can destroy the tidy digital signal provided by the transmitter. If you understand the deterministic processes that close the eye then you have a fighting chance of fixing the problem with equalization. The mathematical purist's approach to equalization gives a prescription for inverting the impulse response as represented by the *S*-parameters. The theoretical result is recovery of the signal as it looked at the transmitter. Incorporating circuit realities like sampling and bandwidth, the continuous and infinite equalization scheme of Eq. (7) is reduced to the discrete and finite scheme of Eq. (8). A byproduct of this transition from an ideal Continuous Time Linear Equalizer (CTLE) to a realistic linear Feed-Forward Equalizer (FFE) is that some Inter-Symbol Interference (ISI) remains and high frequency random noise is amplified. To fix these byproducts, a feedback loop is added to the FFE resulting in a Distributed Feedback Equalizer (DFE).

The three equalization methods discussed here, transmitter pre-emphasis, FFE and DFE, are standard industrial techniques, but there's nothing to keep you from coloring outside the lines and implementing something better. Something that can adapt to changing conditions and address environmental changes, perhaps crosstalk, and maybe, if you're really good, random noise.