Crossing the Digital-Analog Divide

White Paper

Digital signals are an idealization and as data rates climb above a few Gb/s, they betray their microwave analog reality.

To understand how to cope with the physical nature of signals that we might prefer to think of as bits, nibbles and bytes, let's start with an ideal digital waveform. To get nice square edges, the perfect waveform requires an infinite sum of harmonic frequencies at odd multiples of the data rate. It's easiest to see in the Fourier series representation of a square wave where f_d is the data rate in Gb/s.

$$\text{Square Wave}(t) = V_{pp} \sum_{odd\ n} \frac{4}{n\pi} sin(n\pi f_d t)$$

To extend Eq. (1) from a simple square wave to a digital data signal requires subharmonics to accommodate all runs of consecutive identical bits. For example, a 0011 sequence requires frequency components at half the data rate, a run of three identical bits requires components at 1/3 the data rate, and so on.

The first challenge is that the transmitter bandwidth limits the actual high frequency content to a handful of higher frequencies, usually just f_d, $3f_d$ and maybe $5f_d$. The resulting waveform has logic transitions whose trajectories have finite slopes and whose edges have some ringing that could be seen in either time or frequency domains, see Figure 1 and Figure 2.

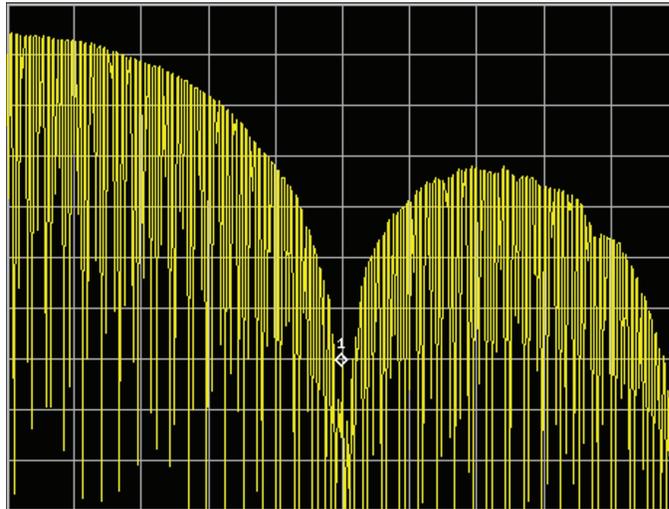Figure 1: A realistic looking waveform at the transmitter

Figure 2: Frequency spectrum of the signal as shown in Figure 1

Anticipate ___*Accelerate* ___*Achieve*

Agilent Technologies

Most people find the time domain a more intuitive description of the system than the frequency domain, though we know they are equivalent representations. In principle, no information is gained or lost in transforming between the time and frequency domains. In practice, there are limits to the equality of the waveform and the spectrum. Windowing of the time domain and limited bandwidth in the frequency domain reduce the precision with which we can revert from one domain to the other. Let us explore a specific example.

It's nice to think of a PCB trace as simply the route that a signal takes from transmitter to receiver and, at low data rates, it's not an unreasonable concept. In the DC ideal, current flows in the conducting trace with a constant magnetic field of cylindrical symmetry around the trace – according to Ampere's Law. However, during logic transitions, the current changes. Changing currents induce an electromotive force which creates eddy currents within the conductor in a direction counter to the change – according to Faraday's Law. The more abrupt the change, the faster the rise or fall, therefore, the stronger the counter eddy current. At high data rates, around 1 Gb/s, the signal current and the induced current begin to cancel, and the net current is restricted to an ever thinner skin at the conductor surface; the skin effect.

The skin effect increases the effective resistance and reduces the inductance of the circuit. Enhanced resistance causes loss. Reduced inductance alters circuit impedance in a way that depends on PCB layout. Impedance variations change the phases of the harmonics and subharmonics. Those phases are what give analog waveforms their sharp-edged digital character. As the phases vary, the signal degrades and reveals its messy analog nature.

Now consider the medium through which the signal propagates. A PCB trace is a very complex waveguide. Plus the FR-4 medium is a fiberglass weave of dubious symmetry and uniformity. At high data rates, it helps to think of the signal as an electromagnetic wave propagating through FR-4 with just a tenuous grasp of the trace. Since the dielectric constant of FR-4 is not constant, rather, it depends on frequency. Different components propagate at slightly different speeds. Over the course of the channel, this dispersion changes the phases of the harmonics even more, further degrading the digital signal.

As it careens through the dielectric, fractions of the wave can propagate through the medium and rejoin the trace farther along. Parts of the wave reflect at small impedance variations caused by turns in the trace or variations in trace width or worse, at major impedance mismatches such as vias and connectors. These multiple reflections and wild transmission paths interfere with that part of the signal that remained faithful to the conducting trace.

The combination of the skin effect, multipath interference and dispersion changes the shape of the waveform. Since the specific changes – attenuation of the peak-to-peak voltage, reduced rise/fall times and delays in logic transitions – depend on the frequency content, and since that frequency content depends on the symbol sequence of the signal, the result is called inter-symbol interference (ISI).
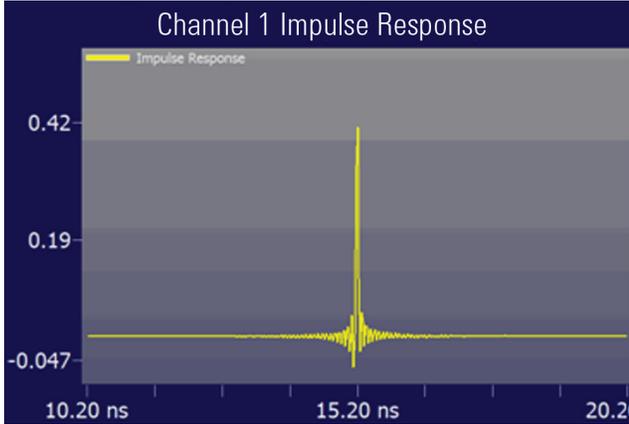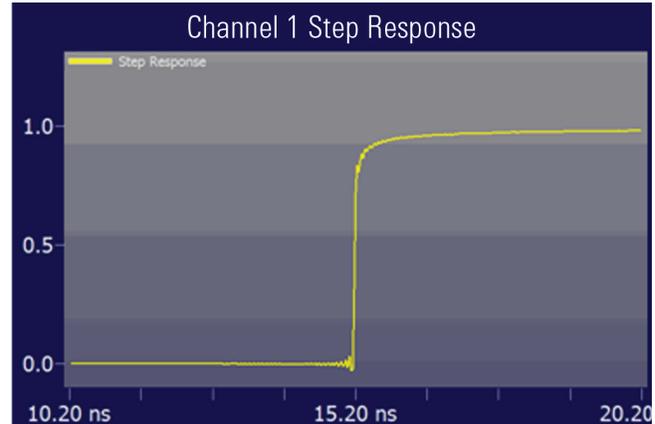
Figure 3a. Impulse response

Figure 3b. Step response

To understand ISI in terms of bit errors, which is the only terms we actually care about, it's easiest to consider the impulse response in the time domain first and then the scattering matrix, or S-parameters, in the frequency domain.

The impulse response is measured by transmitting a single narrow pulse through the circuit. The ideal impulse would be an infinitely thin, very high amplitude signal. As it propagates, the impulse spreads into a function whose shape at the receiving end incorporates the complete channel response as shown in Figure 3a. Imposing the impulse response on a transmitted signal produces that signal's output waveform. In mathematical terms, we think of convolution which amounts to folding the impulse response over every element of the input signal to produce the received signal.

The reason processing an impulse through the channel reproduces the output waveform is simple. A sufficiently narrow pulse is composed of an even distribution of frequency components. Thus, as the impulse suffers the skin effect, dispersion and multipath interference, the frequency response of the channel is encoded in the waveform that emerges.

The impulse response can also be derived from the step response as shown in Figure 3b, because the derivative of a sharp step is an impulse. The pulse response, that is, the response of a single 1 within a long string of 0s, has the same information, as well. Even a data signal can be used if the transmitted waveform has sharp enough edges.

The scattering matrix is the frequency domain equivalent of the impulse response. It can be obtained from a vector network analyzer (VNA) by measuring the channel response to individual sine waves in frequency steps from DC to the maximum transmission bandwidth. The response is measured for all four possible cases: transmission and reflection from both the transmitter and receiving ends. S-parameters are the elements of the scattering matrix, see Figure 4.

Since sine waves are easy to produce at precise frequencies and amplitudes, whereas infinitely narrow impulses and perfect steps only exist in our imaginations, the frequency domain approach on a VNA with physical layer test software is usually more accurate at higher frequencies. Though technology is constantly improving, right now the time domain approach is accurate to about 12 GHz, while the frequency domain approach maintains fidelity over 70 GHz.

Since signal degradation from ISI can be predicted from fixed and measurable characteristics of the channel, we should be able to correct it. Receiver equalization and signal de-emphasis are two approaches. The most obvious characteristic of the channel is its low pass nature. In building a square wave, the high frequency Fourier components are responsible for making nice square, digital-looking edges. De-emphasis amplifies those high frequencies at the transmitter by applying extra large voltage swings to logic transitions and then reducing, or "de-emphasizing," the peak-to-peak voltage of logic sequences that follow transitions. For example, in a 0111 sequence, the voltage swing of the first 1 is larger than that of the second or third 1.

Figure 4. The S-parameters

At the receiver, even with closed eyes, equalization techniques can distinguish 1s and 0s with error ratios better than the standard one in a trillion. There are three essential techniques. First, the continuous time linear equalizer (CTLE) is a band pass filter that amplifies the frequencies at the high end of the spectrum — sort of the back-end equivalent of transmitter de-emphasis.

The second technique is feed forward equalization (FFE). FFE uses the received voltage levels of surrounding bits to help determine the value of a given bit. The coefficients applied to the surrounding voltage levels are directly related to the impulse response. The decision feedback equalizer (DFE), the third technique, adds a clever nonlinear digital layer to the FFE.
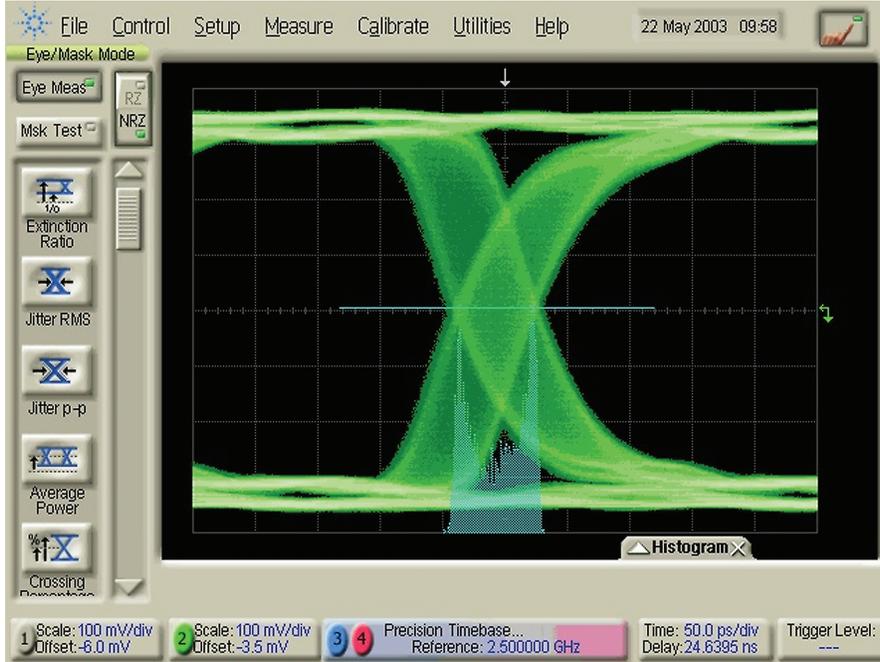
Figure 5. A jitter histogram

Equalization opens eyes that have been closed by predicting signal integrity problems, but what about noise?

As a digital signal is guided by the trace through the PCB dielectric, it is endangered by both external and internal sources of noise. Internal noise sources are random voltage noise from the transmitter and random phase noise from the underlying clock; both contribute to random jitter (RJ) and, due to its random nature, equalization doesn't help.

Siting voltage noise as a source of jitter might be counterintuitive. After all, isn't voltage noise what we call "noise" and phase noise what we call "jitter"? Not quite.

Jitter is the displacement of the timing of logic transitions from their ideal time positions. Figure 5 shows a jitter distribution histogram of the crossing point of an eye diagram.

To understand the distinction between phase noise and jitter and how voltage noise contributes to jitter, let's represent a digital waveform like this:

$$\text{digital waveform} = (V_{pp} + \delta V(t))$$
$$S(2\pi ft + \varphi(t))$$

S indicates the sum of harmonics and subharmonics around the fundamental frequency that make up the digital data signal; $\delta V(t)$ is the voltage noise and $\varphi(t)$ is the phase noise. On an oscilloscope, $\delta V(t)$ causes vertical fluctuations and $\varphi(t)$ causes horizontal fluctuations.

The sampling point is the time-delay at which the voltage slicer determines whether a bit is a 0 or a 1; if the voltage is above the slice-threshold, it's a 1, if below, it's a 0.
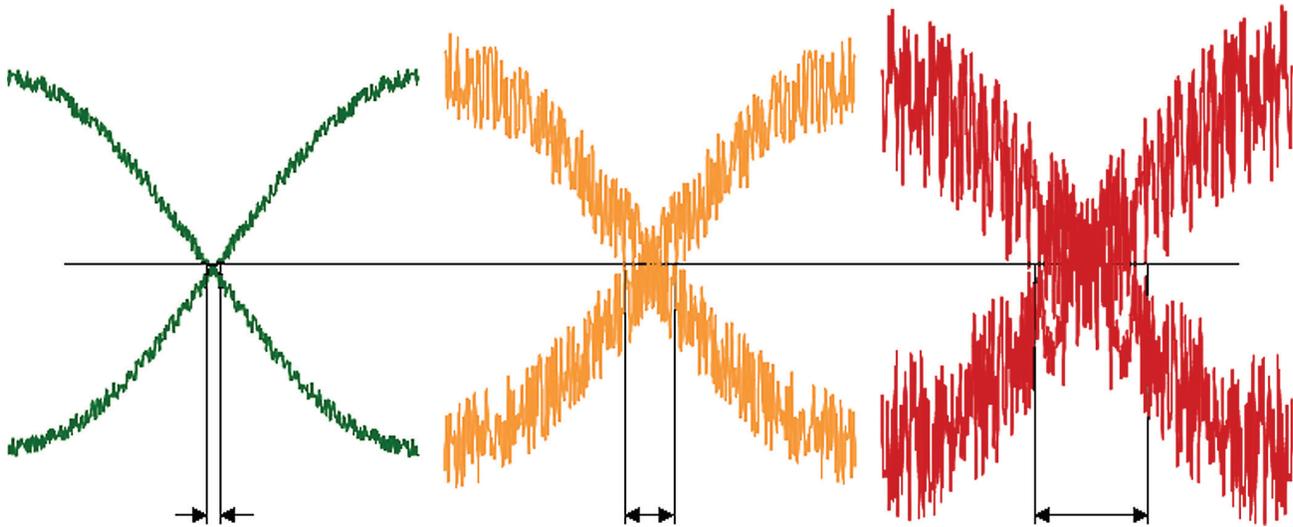
Figure 6. How voltage noise causes jitter

Now picture the trajectory of a $0 \rightarrow 1$ transition. The trajectory is a continuous line with a finite slope and nonzero rise time. If voltage noise pushes the trajectory down, the point where that trajectory crosses the voltage threshold shifts toward the sampling point, see Figure 6. This is how voltage noise causes jitter.

To analyze phase noise, we need a signal that isn't cluttered with all the subharmonics in the data so we use a clock-like signal, a repeating 1010 sequence. Figure 7 shows the resonant shapes of some oscillators. The greater the quality, the narrower the width and higher the amplitude. A perfect oscillator would be a single narrow line. Phase noise broadens the line and reduces the amplitude of the peak in a way that leaves the total power unchanged.
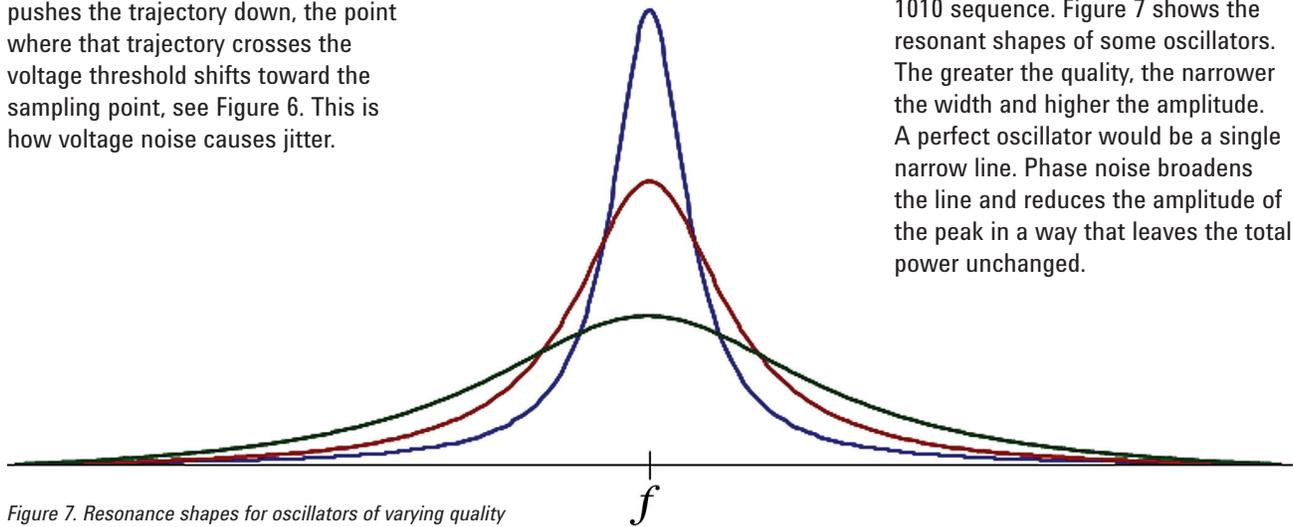
Figure 7. Resonance shapes for oscillators of varying quality

To get the phase spectrum, let's start with the voltage spectrum of the clock-like signal as seen in Figure 8.

Phase noise analyzers extract the phase from the clock-like signal to get $\varphi(t)$ which is then transformed to the phase-frequency domain. The frequency axis of this "phase spectral density" is the offset from the oscillator frequency, $f - f_c$. Since we want to see deep into the gritty noise, it's plotted on a log-log scale.

Let's retrace our steps for a minute. To get to the frequency domain, we transformed from the signal as a function of time to the signal as a function of signal frequency. To get the phase noise, we transformed from the phase as a function of time to the phase as a function of phase frequency which is phase noise. The horizontal axis in Figure 9 is the frequency of the phase, not the frequency of the signal. The business of phase frequency as opposed to signal frequency can be confusing because it's sort of like a picture within a picture.

Recall that jitter is the displacement of the timing of logic transitions from their ideal time positions. Phase noise is the deviation of the phase from its ideal. The majority of random jitter originates in the oscillator that drives the entire system. If we integrate the phase spectral density we get the RJ contribution due to phase noise.

Since almost everything in nature (and electronics) either has the properties of an oscillator (e.g., atoms, molecules, hearts, stars, crystals, ...), is governed by an oscillator or is a component of an oscillator, the phase noise spectrum says a lot about what's going on at the core of the system. The smooth shape of the spectrum comes from contributions of FM and AM random walk, flicker
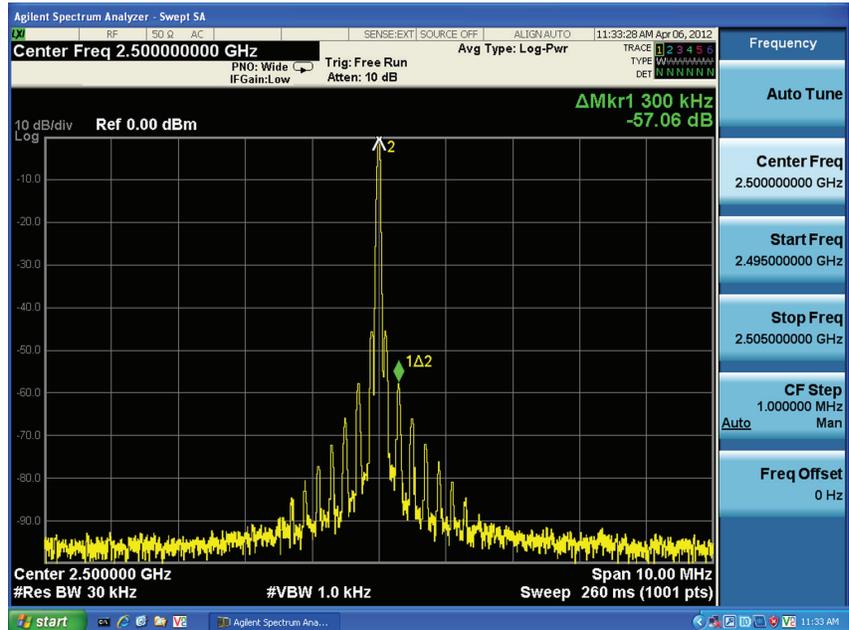
Figure 8. The frequency spectral density
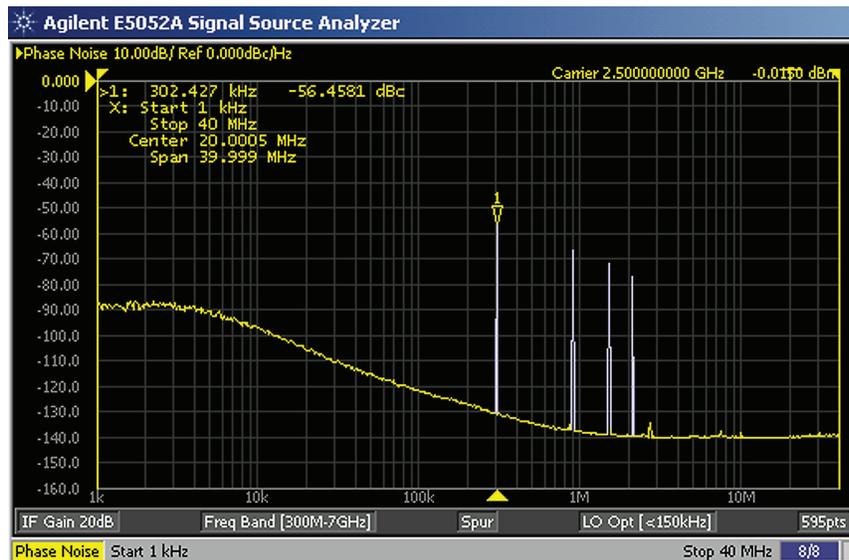
Figure 9. The phase spectral density

and white noise. The shape of the spectrum indicates specific oscillator properties and defects like shock, temperature sensitivity, microcracks in a crystal, problems with frequency multipliers, thermal noise in associated components, and so on. For a

good oscillator, the real trouble shows up as spikes as seen in Figure 9. Spikes in the phase spectrum betray sources of sinusoidal and periodic jitter (SJ and PJ) which are a symptoms of electromagnetic interference.

One of the main reasons for the success of high speed serial technologies is that embedded clocking reduces errors caused by jitter. By embedding the system clock in the data and reconstructing it at the receiver, jitter at frequencies below that of the clock recovery bandwidth remains on both the data and the clock. Since the timing of the sampling point is set by the clock, it jitters in harmony with the data and, therefore, the jitter below the bandwidth of the clock recovery doesn't cause errors. Embedded clocking essentially provides a low pass phase-frequency filter as shown in Figure 10.

Another key reason for the success of high speed serial technology is differential signaling. By transmitting both positive and negative versions of the signal on nearby traces and then using the difference of the two to identify bits at the receiver, problems with interference and crosstalk are diminished.

Crosstalk is electromagnetic interference picked up on a signal line. It can be a crippling problem with parallel architectures. Going back to our vision of a signal flying through PCB as guided by a conducting trace, any other signal on the PCB can excite eddy currents on our trace and degrade the signal.

In an ideal differential signaling configuration, the two lines are exactly the same length, are identical in geometry, and perfectly overlap. The two lines would be infected by exactly the same interference and, in combining the signal with its complement, the receiver would cancel it perfectly. Of course, if the lines overlapped in the real world, they would short out; at best they're separated by a millimeter or so. They're also never quite the same length, so the cancellation is never perfect. As data rates exceed 10 Gb/s, this differential lane skew approaches the length of individual bits and the situation deteriorates. Interference, especially crosstalk, can be a detrimental to digital designs..

While it's not yet clear if equalization techniques can rescue us from crosstalk, it is clear that more clever solutions will have to emerge in order to account for the analog problems digital designers will face with ever increasing data rates...
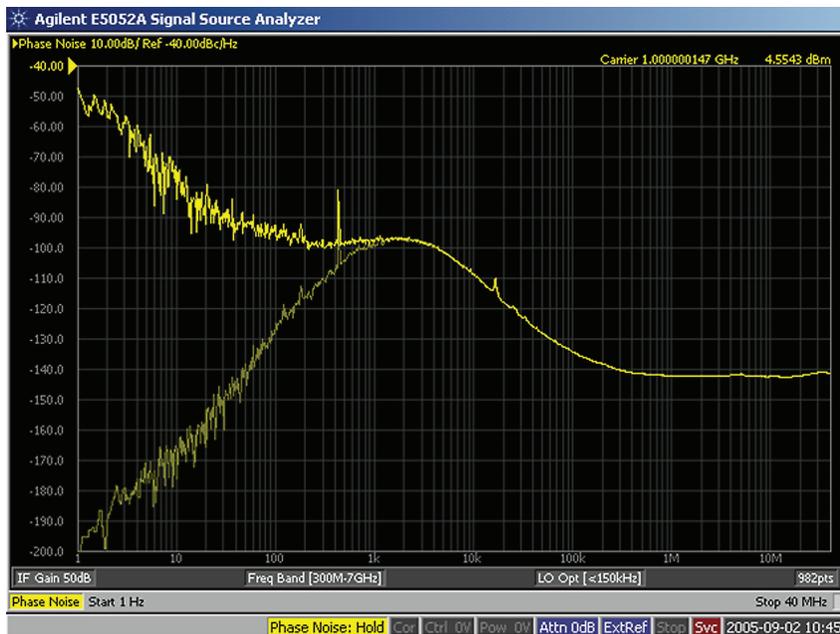
Figure 10. Phase noise with and without the filtering effect of clock recovery