The Rules of Jitter Analysis

by Ransom Stephens, PhD
Agilent Technologies

The Rules of Jitter Analysis:

1. It's about the bit error ratio
2. Total jitter can only be *measured* on a bit error ratio tester
3. Measurements of jitter are always the comparison of a test clock and a reference clock
4. Timing noise and amplitude noise are not really separable (but we do it anyway)
5. It's **still** about the bit error ratio

Whenever technology advances there is an acronym explosion. The impact of jitter on advances to higher data rates is a perfect example: TJ, RJ, DJ, PJ, SJ, DDJ, BUJ1. And whenever there's an acronym explosion it's exhausting to try to keep up with the jargon and easy to overlook what really matters. The rules of jitter analysis offer an oasis of fundamental reality in a desert of abstract acronyms.

Between the acronyms, phase noise, timing noise, and "the deviation of significant instants of a digital signal from their ideal positions in time,"2 it's easy to forget why we care about jitter. The first rule of jitter analysis is it's about the bit error ratio.

We care about jitter for exactly the same reason that we care about signal-to-noise ratio: a low SNR means a high bit error ratio. Where signal-to-noise ratio describes the effect of voltage noise on a signal, jitter describes the effect of timing noise on a signal. Voltage noise causes bit errors when the signal voltage fluctuates vertically across the sampling point. Similarly, jitter causes errors when the timing of a signal transition fluctuates horizontally across the sampling point3.

The first rule of jitter analysis begs for a concrete way of describing jitter in terms of the bit error ratio (BER), we call it TJ (sorry about the acronyms, but these two are all you need for the moment.) Total jitter is defined in terms of the BER in the sense that it only makes sense to think of TJ in reference to a BER. We'll call it TJ(BER) because it changes depending on the BER of interest. Fig. 1 shows a data stream with jitter. The clock signals are indicated by dashed lines. In a system with no jitter, the logic transitions would coincide with the clock transitions. TJ(BER) defines the amount of jitter that causes a given BER. This way, if we specify that a system must perform with a BER that is less than say, 10^{-12} (no more than one error for every trillion transmitted bits), then the sum of all sources of jitter in the system, transmitter, channel, and receiver, must combine so that $TJ(10^{-12}) < Tb$, where Tb is a bit period. Defining jitter with respect to a BER is at once clever and annoying. It's clever because it gives us a way to link jitter and BER. It's annoying because of the second rule of jitter analysis: TJ(BER) can only be measured on a bit error ratio tester (BERT) and BERTs are expensive.

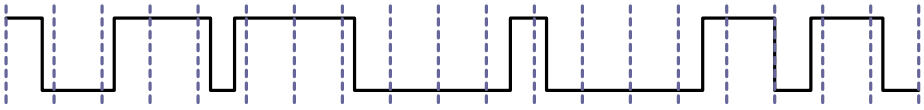

Fig. 1: Data Signal With Jitter
The solid line is a jittered data signal whose logic transitions ought to line up with the dashed lines. The dashed lines indicate the ideal transition times defined by a clock signal

To describe TJ(BER) it's easiest to explain how it's measured4. If we take the data stream of Fig. 1, use the clock for a trigger and look at the signal on an oscilloscope, we get an "eye-diagram:" the superposition of bits in a transmitted signal (Fig. 2a).

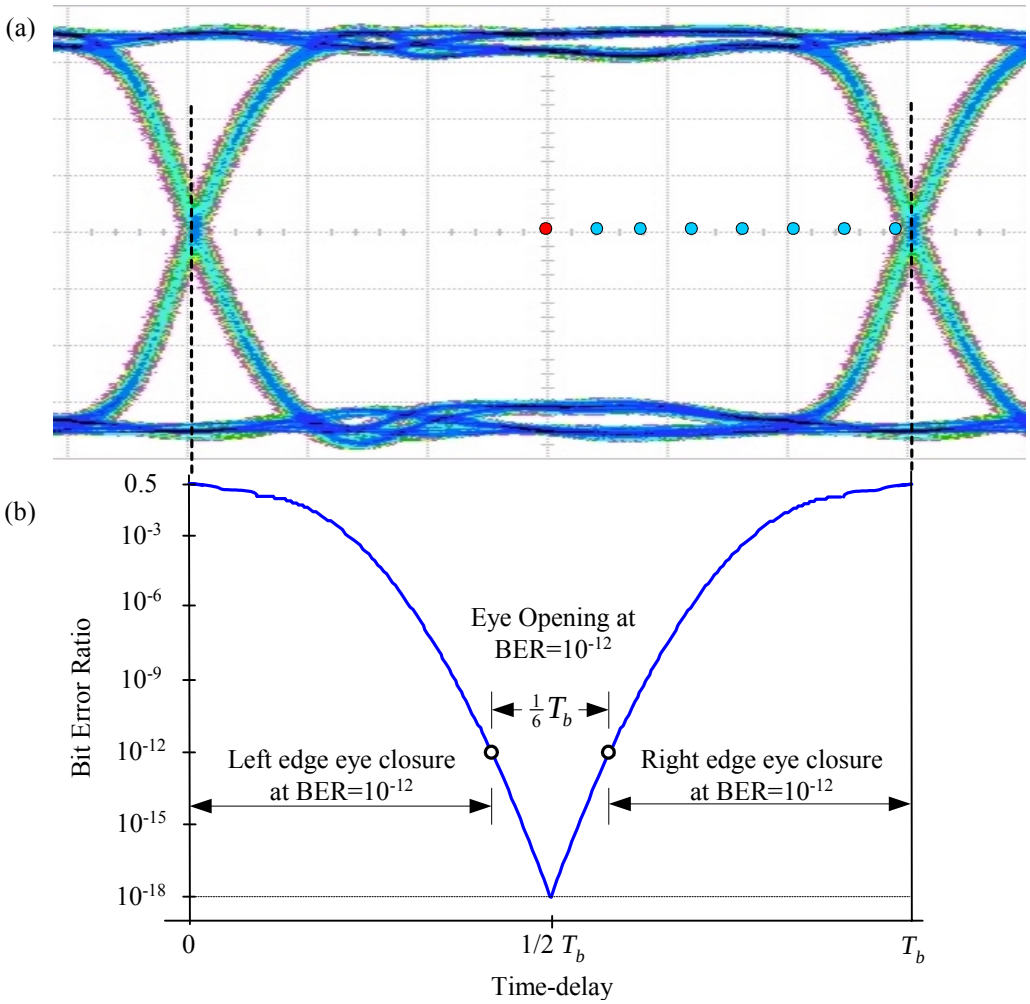

Fig. 2: (a) Eye Diagram With Ideal Sampling Point Indicated By Red Dot (b) Graph Of BER(t) Measured On BERT By Scanning The Sampling Point Across The Eye, Shown By The Blue Dots In (a)
The total jitter is the amount of eye closure defined at a given BER in (b); for example, TJ(10^{-12}) = 5/6 Tb and TJ(10^{-18}) = 0 because the eye is fully closed at BER=10^{-18}

Right now we don't really care about oscilloscopes (we will later, though), it's the eye-diagram that's useful. The ideal sampling point is set at the eye-center, the red dot; if the logic level of a bit is above the sampling point, it's a '1', if it's below, it's a '0'. A BERT transmits known patterns through a system and then detects the output. It counts the number of errors and keeps track of the total number of bits to get the ratio of errors received to bits transmitted; ie, the BER. To measure TJ(BER) the sampling point is moved horizontally across the eye as shown by the blue dots. As the sampling point gets closer to the crossing point, jitter causes the BER to increase, as shown in Fig. 2b. The time position of the sampling point is called the "time-delay" because it is set as a delay with respect to a reference clock. In other words, the time-delay is that distance in time from one of the dashed lines in Fig. 1 (the clock signals) to the point where the voltage of the corresponding bit is sampled to decide whether it is a '1' or '0'. Anyway, the BERT measures the BER as a function of the time-delay, BER(t), as illustrated in Fig. 2b. The distance between the left and right BER(t) curves gives the eye opening at the BER. TJ(BER) is the difference between the bit period and the eye opening, that is, TJ(BER) is the amount of eye closure at the BER. The eye is closed at the BER where the left and right slopes of the BER(t) curves intersect; in Fig. 2b, the curves intersect at BER = 10^{-18} so TJ(10^{-18}) = 0, but at BER =10^{-12}, we get TJ(10^{-12}) = 5/6 Tb.

The point to pay attention to here is that TJ(BER) is exactly what we need to stay firmly grounded in relating jitter to bit errors. While it can only be measured on a BERT – a measurement that takes a long time, eg, at 2.5 Gbit/s it takes at least an hour – it can be estimated very quickly (using a bunch of dubious approximations, some clever algorithms, a few guesstimates, and that mess of acronyms) on both real-time and equivalent-time-sampling oscilloscopes and time interval analyzers. BERTs can also make fast TJ(BER) estimates. The accuracy of the estimates is excellent on some test-sets but varies widely and depends on industry-wide assumptions about the statistical nature of different causes of jitter. In choosing a jitter analysis test set, take care to understand the evidence used for quoted accuracies. Getting back to Fig. 1, notice that the jitter of the signal is measured with respect to a clock signal, the dashed lines. The clock defines the "ideal positions in time." There are two important things to take away from this fact. One of them is subtle and one is not. The one that's not subtle is given by the third rule of jitter analysis: measurements of jitter are always the comparison of a test clock and a reference clock.

When measuring jitter on a data signal it's easy to forget that you're simply comparing the logic transition times to reference clock transition times. In describing TJ(BER) I compared the eye closure in Fig. 2b with the bit period, Tb; here, the bit period is where the reference clock is hidden. The subtle point in "the reference clock defines the ideal positions in time" is understanding what is meant by the word "clock," the true identity of Tb. In most serial data systems the clock is recovered from the data itself, a weird concept when you think about it; we're comparing the timing of the data with a clock that was reconstructed from the timing of the data.

Fig. 3 shows a typical serial data system, the clock is recovered from the data and the logic levels of the data are sampled at times determined by the clock. If everything were perfect then the recovered clock would have exactly the same jitter as the data. If the clock jittered the same amount as the data then the sampling point would dance with the jitter and jitter wouldn't cause any errors. Then we could stop caring about jitter. That the recovered clock shares some jitter with the data is one of the strengths of serial data systems: some of the jitter doesn't matter.

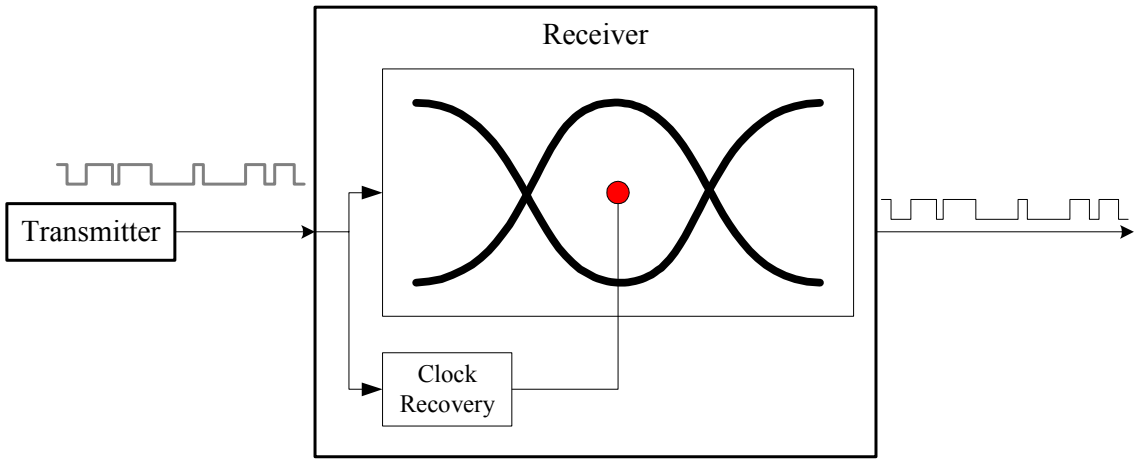

Fig. 3: System With Embedded Reference Clock Reconstructed By Clock Recovery Circuit
The clock recovery circuit sets the time position of the sampling point shown by the red dot

The trick is in discriminating the jitter we care about from that we don't. To get there we need to examine clock recovery more closely5. Clocks are recovered from data transitions by locking the phase of an oscillator to the phase of the data edges; a standard application of a phase locked loop (PLL). The bandwidth of the clock recovery circuit determines the jitter that can cause errors. The ideal clock recovery circuit would give clock and data transitions that jitter together in perfect harmony – an infinite bandwidth clock recovery circuit, which is impossible to realize. At the opposite extreme is the zero bandwidth clock recovery circuit where the clock stands still while the signal dances; the sampling point wouldn't track jitter at all – the technical equivalent of a distributed rather than recovered reference clock. Of course the real situation lies between the extremes: a nonzero, finite bandwidth clock recovery circuit. A clock recovery circuit with bandwidth, Δf, behaves like a high-pass jitter filter: the sampling point dances with the jitter at frequencies below Δf, but not above.

At this point we need to be careful with the term "frequency." The signal and jitter occupy two different frequency domains that shouldn't be confused. Fourier said that any periodic function can be described in terms of its harmonic frequencies – this is the frequency domain of the signal. But no real phenomenon is perfectly periodic, there is always something to cause a given cycle to differ from the rest. As usual the culprit is noise – from thermal oscillations, electromagnetic interference, or various acronyms – in this case, it is phase noise that causes deviations from the periodic ideal6. The phase

noise spectrum is what I mean by jitter-frequency spectrum; the frequency domain of the deviations from the ideal phase.

Think of a sinusoidal clock signal at some rate, *f*0 (Fig. 4a). The frequency domain of the clock signal is a spectrum with a sharp peak at *f*0 and smooth skirts dropping down around it (Fig. 4b). In a mathematically ideal world it would have a perfect peak at *f*0 with no skirts at all (Fig. 4c), but in real life the resonance of the oscillator isn't perfect; phase noise causes deviations from the resonant frequency resulting in the difference in the spectrums of Figs. 4b and 4c. For a clock signal, the jitter-frequency domain is the spectrum of the deviations from that sharp peak at *f*0 – the single-sideband or phase noise spectrum. A typical clock phase noise spectrum is shown in Fig. 57.

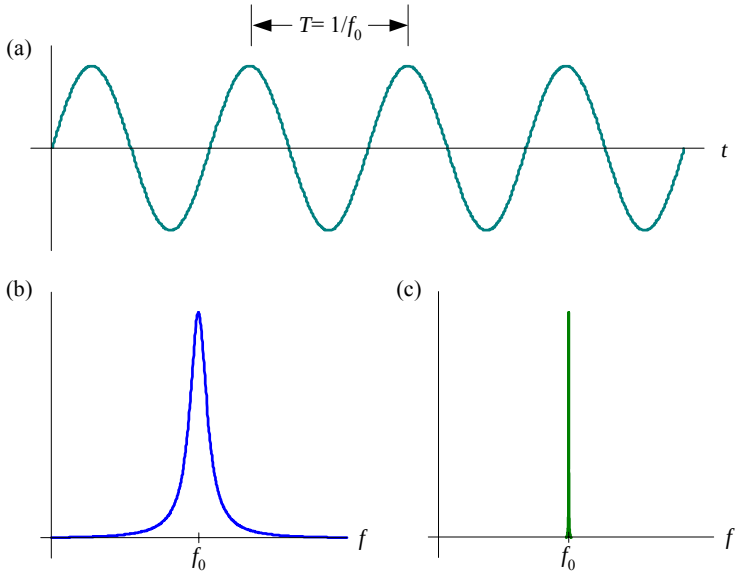

Fig. 4: (a) Sinusoidal Clock Of Frequency *f*0
(b) Frequency Spectrum Of Clock With Moderate But Nonzero Width
(c) Frequency Spectrum Of Ideal Clock With Zero Width

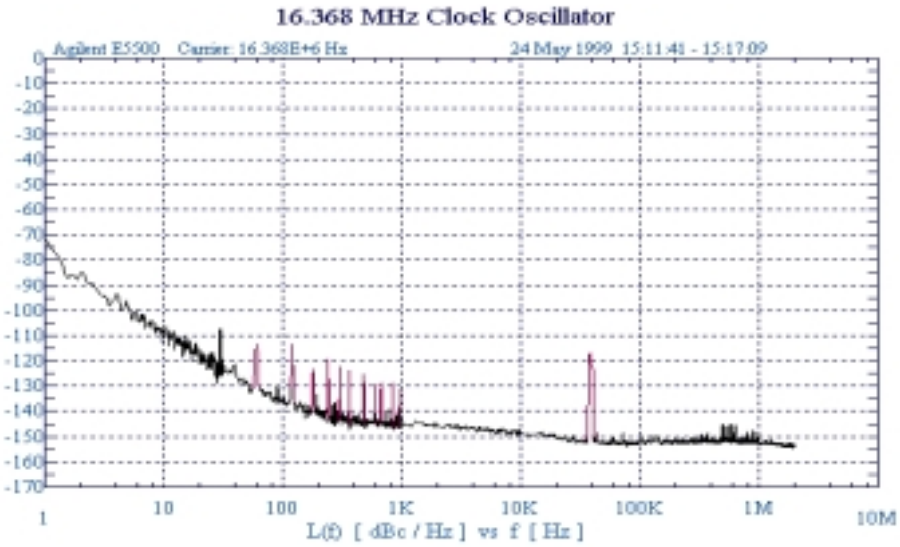

Fig. 5: Phase Noise Spectrum Of Clock Oscillator

The bandwidth of the clock recovery circuit tells us exactly what part of the jitter-frequency domain to worry about. Fig. 6 shows a typical clock recovery frequency response. The bandwidth rolls off at BW. The third rule of jitter analysis says we need to compare the jitter on the data with a reference clock that has the frequency response defined for that technology – the jitter that can cause errors (cf, the first rule of jitter analysis). The jitter that can cause errors is the jitter that doesn't dance in harmony with the clock; that is the jitter-frequencies above the clock recovery roll-off8. In most cases, especially when the clock is embedded in the data, regardless of the test equipment you use, the best approach is to reconstruct the clock from the data. On real-time oscilloscopes you can reconstruct the clock with a software model of the PLL defined for that technology. If you're working at data rates above a few Gbit/s, or if you want to analyze a signal with a wider bandwidth receiver than a real-time oscilloscope can provide, you'll need a hardware clock recovery circuit with the right bandwidth, a so-called golden PLL. Clock recovery circuits are available for almost every emerging technology and at higher bandwidths real-time oscilloscopes are so expensive that it's cost effective to use a truly wide bandwidth solution like an equivalent-time sampling oscilloscope anyway9.

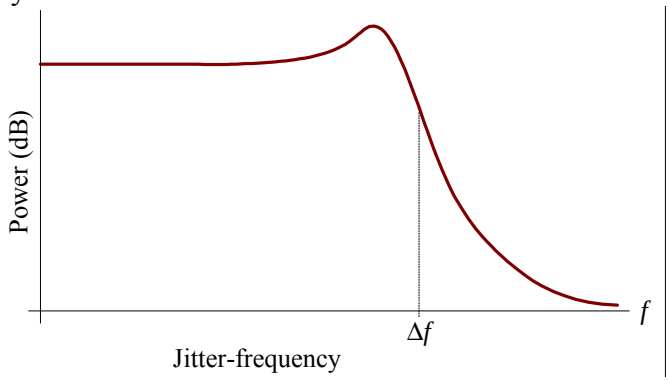

Fig. 6: Frequency Response Of Clock Recovery Circuit
Jitter Below Recovery Bandwidth, Δf, Causes No Errors; Jitter Above Can

We can summarize the first four rules of jitter analysis in the context of test equipment (which is of paramount importance for me to get paid each month): The second rule says that if you want to measure TJ(BER), you'll need a BERT (though you can make fast estimates of TJ(BER) with oscilloscopes and so forth). The third rule says you need to recover the clock from the data to make sure you compare the jitter on the signal to the correct reference clock. This brings us back to the first rule of jitter analysis because, by studying jitter with respect to the right reference clock, then you're only analyzing that jitter which can affect the bit error ratio.

The first three rules of jitter analysis are the keys to any jitter analysis issue. Yes, there are plenty of details – the dual-Dirac business, RJ, DJ, DDJ, correlated, uncorrelated, bounded, unbounded, etc – but the foundation of the whole mess is covered in the first three rules. If you weren't before, you are now ready to ask intelligent-sounding questions at signal integrity meetings – my apologies to your colleagues.

But here's the rub. You remember feeling like you'd peeked at the secrets of the universe after you took freshman mechanics and electromagnetism? And then the professor started talking about relativity and quantum mechanics and you got that, "so everything you just taught me is wrong" feeling? Well, here we are again. The fourth rule of jitter analysis is timing noise and amplitude noise are not really separable (but we do it anyway). In other words, everything I just told you is wrong. But only a little bit wrong.

The thing to remember is that noise is noise. The voltage noise that reduces the signal-to-noise ratio doesn't just cause amplitude fluctuations, it also causes timing fluctuations – jitter and voltage noise are not independent. Fortunately, just as an electrical engineer can design a circuit much better than a particle physicist (believe me, much better), separating voltage and timing noise, while not really right, is useful in the sense of the first rule of jitter analysis. The utility of the fourth rule is to watch out for cases where the separation breaks down. I suspect that thinking of jitter and voltage noise separately starts to break down between around 10 Gbit/s10.

I hope the rules of jitter analysis will help you focus on those design issues that really matter – which brings me to the fifth rule of jitter analysis, it's still about the bit error ratio!

About The Author

Ransom Stephens (ransom_stephens@agilent.com) specializes in the analysis of electrodynamics in high-rate digital systems and the marketing of analysis tools developed by the Digital Verification Solutions division of Agilent Technologies. He has spent the last four years analyzing timing noise and dispersion and developing new techniques for extracting signals from noise. He received his PhD in physics at UCSB in 1990 and spent the succeeding ten years in basic research making precise measurements of rare processes in particle physics at laboratories across the United States and Europe.

Footnotes

[1] A list of many of the acronyms is in Maxim Application Note 1916, "An introduction to jitter in communications systems," 2003, available from www.maxim-ic.com.
[2] NIST Technical Note 1337, "Characterization of clocks and oscillators," edited by D.B. Sullivan, D.W. Allan, D.A. Howe, F.L. Walls, 1990.
[3] For a terrific general reference, see Howard Johnson, *High speed signal propagation: advanced black magic*, (Prentice Hall, 2003).
[4] For a nice summary of jitter analysis with a bit error ratio tester, see Agilent Technologies' Application Note, "Jitter fundamentals: Agilent N4900 serial BERT series jitter injection and analysis capabilities," literature number 5989-0089EN, 2003, available from www.agilent.com.
[5] For an example of how to make a clock recovery circuit see Leonard Dieguez, Xilinx Application Note XAPP:250, "Clock and data recovery with coded data streams," 2004, available from www.xilinx.com.
[6] For a complete discussion of phase noise see, W. P. Robins, *Phase noise in signal sources (theory and applications),* (Peter Peregrinus Ltd., 1982); and for an example of a practical analysis, see, Texas

Instruments Application Report, SCAA067, “Phase noise (jitter) performance of CDC7005 with difference VCXOs,” 2003, available from www.ti.com.

[7] Agilent Technologies’ Application Note AN 1309, “Pulsed carrier phase noise measurements,” literature number 5968-2081E, 2000, available from www.agilent.com.

[8] Mike Li and Jan Wilstrup, “Statistical and system transfer function based method for jitter and noise in communication design and test,” proceedings of DesignCon 2004.

[9] Agilent Technologies Product Note 86100C-1, “Precision jitter analysis using the Agilent 86100C DCA-J,” literature number 5989-1146EN, 2004, available from www.agilent.com; Aeroflex Inc Application Note #1, “PN9000 phase noise measurement system,” available from www.aeroflex.com.

[10] For an example of how voltage and timing noise can be addressed without separation, see Anthony Sanders and Mike Resso, “Channel compliance testing utilizing novel statistical eye methodology,” proceedings of DesignCon2004.

as published in...
analog ZONE