



# Practical Analysis of Refclk Jitter for SerDes Applications

Dr. Gary Giust, SiTime Corp.  
ggiust@sitime.com

## **Abstract**

The traditional methodology for analyzing SerDes reference clock (refclk) jitter based on a 12 kHz to 20 MHz brick wall filter served the industry well for decades but is no longer suited for modern high-speed designs due to numerous sources of error leading to suboptimal link performance. Instead, we recommend an easy, practical, and accurate methodology to evaluate and specify refclk jitter that minimizes total system error rates. Industry adoption of this methodology will ensure SerDes vendors and their customers optimize link performance while simplifying refclk selection.

## **Author(s) Biography**

Dr. Gary Giust is a technical director at SiTime developing industry-leading timing solutions. Prior to SiTime, Gary founded JitterLabs, and previously worked at Applied Micro, PhaseLink, Supertex, Cypress Semiconductor, and LSI Logic. Gary obtained a Ph.D. from Arizona State University, Tempe, an MS from the University of Colorado, Boulder, and a BS from the University of New Hampshire, Durham, all in Electrical Engineering.

## Motivation

Proper evaluation of reference clock (refclk) jitter is critical to optimize system performance in high-speed serial links. Figure 1 shows how the traditional 12 kHz to 20 MHz refclk jitter analysis methodology, established in 1992, can mislead today’s designers to select components that degrade rather than improve link performance. Using a more accurate “4-16A” phase-jitter filter methodology (discussed below) predicts at least a 50% improvement (45 vs 94 fs RMS) in system jitter. This paper reveals the limitations of the traditional 12 kHz to 20 MHz methodology and recommends the industry adopt a more meaningful methodology to easily evaluate then select refclk sources that optimize system performance in high-speed serial links.

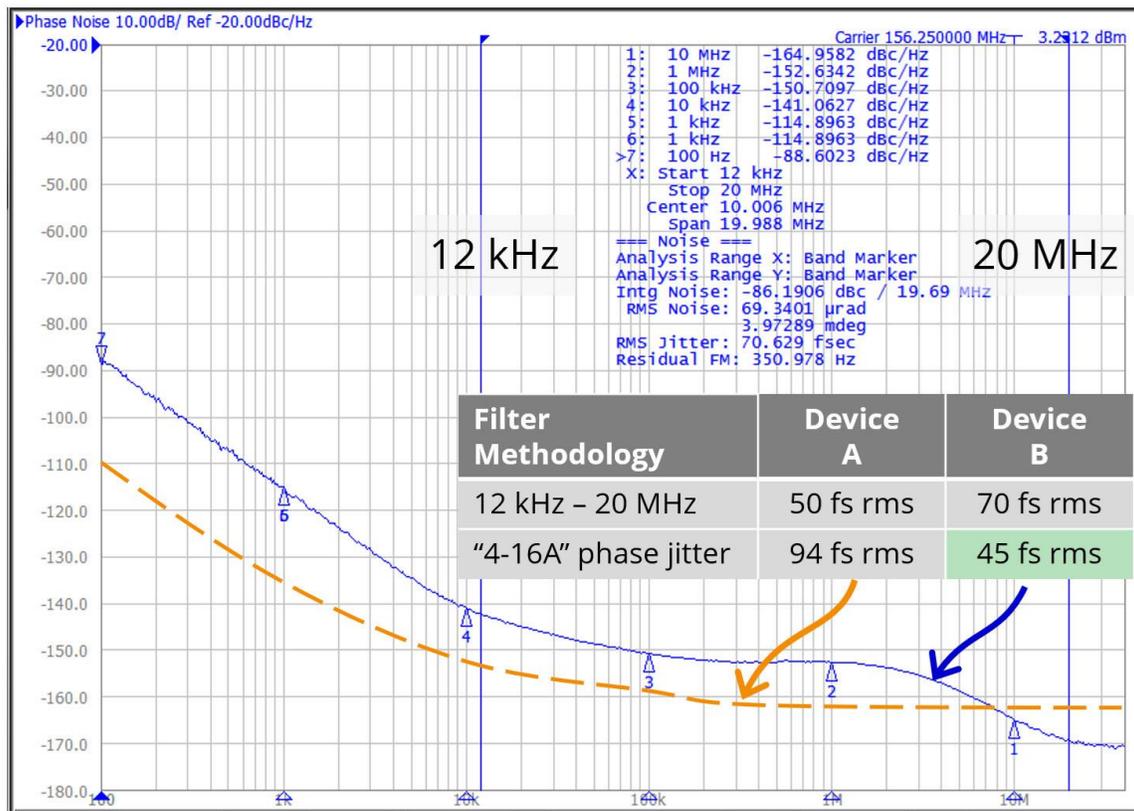


Figure 1. Two products analyzed with 2 different methodologies lead to opposite conclusions. The “4-16A” phase jitter methodology more accurately predicts in-system performance for high-speed serial links versus the traditional method.

## Traditional Refclk Jitter Analysis

The public’s insatiable demand for data requires high-bandwidth links to transport data quickly and efficiently. To meet this demand, silicon serializer-deserializer (SerDes) vendors must keep pace with ever-increasing line rates. One hurdle to designing faster SerDes chips is complying with shrinking jitter budgets with each new generation of silicon. In general, as data rates double, the jitter must cut in half to keep the same timing

margin. Part of the SerDes transmitter jitter budget specified by high-speed standards is consumed by an external refclk driving the transmitter. In most cases, to add flexibility, the exact jitter budget allocated to this refclk is left up to the implementor, and not explicitly defined by standards. Nevertheless, in the absence of standards, a “de facto” refclk jitter methodology emerged and has been applied for over three decades.

The banner specification in clock and oscillator datasheets used in jitter-critical applications is commonly referred to as “phase jitter”. A system analysis of refclk phase jitter in serial links is illustrated in Figure 2. Although the figure is drawn for the popular embedded-clock architecture, the proposed methodology applies equally to other clocking architectures with proper selection of transmitter (TX) and receiver (RX) jitter-transfer functions.

As shown in Figure 2, a portion of refclk phase noise (or, jitter) passes through the low-pass TX’s phase-locked loop (PLL) jitter transfer function to appear on the serial data output by the SerDes. This jitter subsequently gets high-pass filtered by the RX’s observed jitter transfer function. Therefore, the overall system band-pass filters the refclk phase noise. The phase jitter observed by the system due to the refclk is computed by integrating the filtered refclk phase noise.

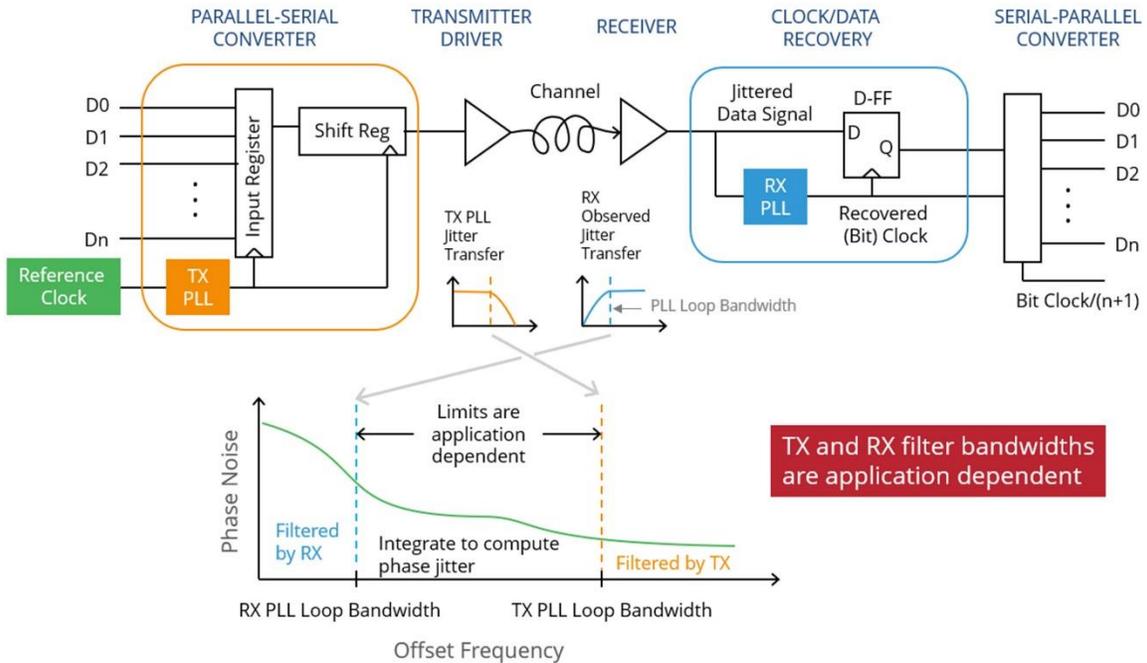


Figure 2. A generic serial link (top) uses a transmit PLL and receive CDR to low and high pass filter, respectively, refclk phase noise. This establishes a band-pass system filter (bottom) for deriving refclk phase jitter.

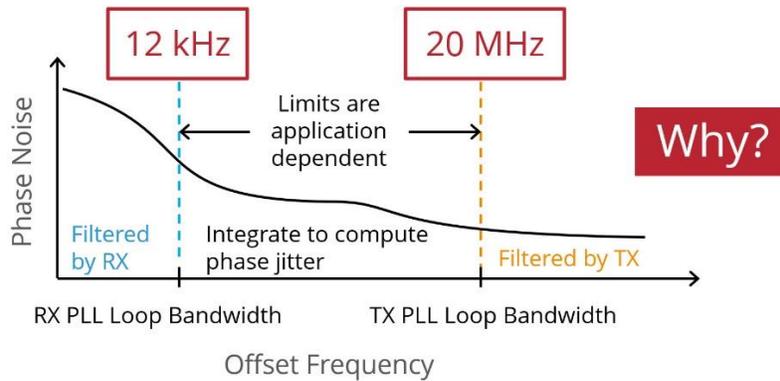


Figure 3. In 1992, clock and oscillator datasheets, for the first time, specified phase jitter using a 12 kHz to 20 MHz brick-wall filter to support sales into the SONET OC-48 (2.488 Gbps) telecom market.

For over three decades, timing vendors have quantified phase jitter by measuring then integrating refclk phase noise over an offset frequency range of 12 kHz to 20 MHz, as shown in Figure 3. Why is that?

This brick-wall band-pass filter between 12 kHz and 20 MHz was initially specified in clock and oscillator datasheets to support refclk sales into, what was at the time (1992), the leading Telcordia data rate of 2.488 Gbps for SONET OC-48. Specifically, Telcordia GR-253-CORE [1] defined low and high-pass jitter generation filter bandwidths for OC-48 to be 12 kHz and 20 MHz, respectively. The 12 kHz related to the system's lowest RX clock-data-recovery (CDR) bandwidth. However, a few years later the telecom industry evolved beyond OC-48 to OC-192 (~10 Gbps) and OC-768 (~40 Gbps). In parallel, SerDes applications expanded and fragmented into many diverse markets (i.e. storage, displays, etc.), each having their own application-specific jitter filter requirements. Despite this, the datasheets provided by clock and oscillator vendors never evolved beyond the original 12 kHz to 20 MHz filter requirements for OC-48. Figure 4 illustrates the evolution of interconnection speed and standards fragmentation, throughout which clock and oscillator datasheets continued to adopt the defunct OC-48 filter specification to this day.

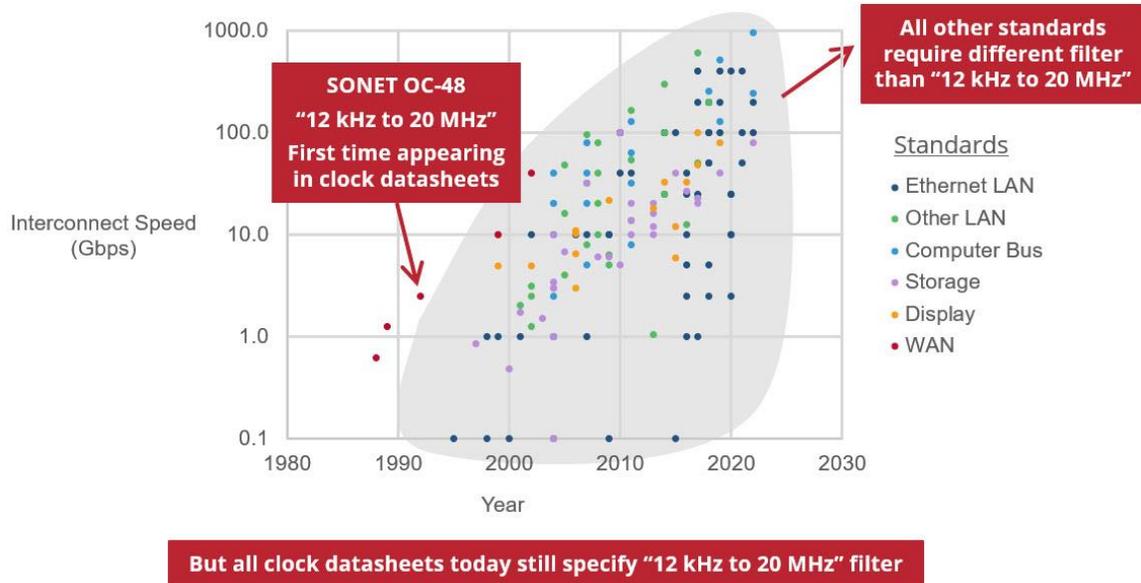


Figure 4. Clock and timing datasheets continue to specify phase jitter for OC-48 using a 12 kHz to 20 MHz brick wall filter despite standards evolving away from this filter in the late 1990s.

## Problems with Traditional Refclk Jitter Analysis

Today, the industry has moved well beyond the 2.488 Gbps data rate of OC-48. A lot more is also known about how to measure and specify phase jitter to emulate what real systems observe. It's time to update the traditional refclk jitter methodology, which unintentionally became the de facto specification for all clocks and oscillators selling into jitter-critical applications since 1992. Evaluating refclk jitter based on the legacy 12 kHz to 20 MHz filter is problematic because:

1. The filter corner frequencies are incorrect, which are application specific and can significantly change the computed phase jitter.
2. Real systems don't use "brick-wall" filters, but rather include tail regions that roll off slowly and contribute significantly to total jitter.
3. The methodology ignores aliased phase noise, which real systems observe, and is often the dominant source of refclk phase jitter.
4. Errors as small as tens of femtoseconds rms are significant and can't be ignored given today's higher data rates and tighter jitter margins.

Figure 5 illustrates how standards evolved away from the 12 kHz to 20 MHz filter frequencies shown in the first row. Specifically, the SONET high-pass filter bandwidth increased to 4 MHz then 16 MHz when OC-48 evolved to OC-192 then OC-768 [1]. The low-pass filter bandwidths also increased, resulting in the system observing an entirely different portion of the spectrum. Furthermore, as serial standards fragmented into other markets, the trend repeats. Ethernet, for example, which started out with a high-pass filter bandwidth of 20 kHz, evolved to 10 MHz as data rates increased.

Application	Standard	Technology	Data Rate (single lane)	HPF bw	LPF bw	Roll off
Telecom networking	GR-1244-CORE	SONET	2.488 Gbps	12 kHz	20 MHz	1 <sup>st</sup> order
Telecom networking	GR-1244-CORE	SONET	9.95328 Gbps	4 MHz	80 MHz	1 <sup>st</sup> , 3 <sup>rd</sup> order
Telecom networking	GR-1244-CORE	SONET	39.81312 Gbps	16 MHz	320 MHz	1 <sup>st</sup> , 3 <sup>rd</sup> order
Passive optical network	100BASE-BX10	Ethernet	0.1 Gbps	20 kHz	Depends on SerDes vendor	1 <sup>st</sup> order
Passive optical network	1000BASE-BX10	Ethernet	1 Gbps	637 kHz		1 <sup>st</sup> order
Backplane	1000BASE-KX	Ethernet	1 Gbps	750 kHz		1 <sup>st</sup> order
Chip-to-chip	XAUI	Ethernet	3 Gbps	1.875 MHz		1 <sup>st</sup> order
Backplane	10GBASE-KR4	Ethernet	10 Gbps	4 MHz		1 <sup>st</sup> order
Backplane	100GBASE-KR4	Ethernet	26 Gbps	10 MHz		1 <sup>st</sup> order
Storage	INCITS-16GFC	Fibre channel	14 Gbps	5.1 MHz		1 <sup>st</sup> order
Storage	INCITS-128GFC	Fibre channel	28 Gbps	10 MHz		1 <sup>st</sup> order
Data center interconnect	OIF2021.144.14	Coherent optics	800 Gbps	3 MHz		1 <sup>st</sup> order
Chip-to-chip	OIF-CEI-6G-SR	Common interface	6 Gbps	3.82 MHz		1 <sup>st</sup> order
Chip-to-chip	OIF-CEI-11G-SR	Common interface	11 Gbps	6.72 MHz	1 <sup>st</sup> order	
Chip-to-chip	OIF-CEI-28G-SR	Common interface	28 Gbps	16.86 MHz	1 <sup>st</sup> order	
Computer interface	USB3.1-GEN1	USB	5 Gbps	4.9 MHz	1 <sup>st</sup> order	
Computer interface	USB3.1-GEN2	USB	10 Gbps	15 MHz	1 <sup>st</sup> order	

Figure 5. The arrows indicate the evolution of different standards over the years. Notice that the high-pass filter (HPF) bandwidth increases with time, and the low-pass filter (LPF) bandwidth is usually defined by the SerDes vendor rather than standards. Thus, the error introduced by applying the traditional 12 kHz to 20 MHz methodology only increases with time.

Standards increase the high-pass filter bandwidth (RX CDR) as data rates increase to minimize the impact of inter-symbol interference (ISI) that results when high-speed data passes through a bandwidth-limited channel. That is, increasing the receiver’s CDR bandwidth reduces the data-dependent component of jitter resulting from ISI as the data passes through a bandlimited channel.

The low-pass filter bandwidth (TX PLL) also increases over time. This is done to minimize the intrinsic jitter contribution from the TX PLL. This bandwidth also tends to increase over time to enable the TX PLL to accept higher-frequency refclks. A higher TX PLL bandwidth makes the loop more stable (ensures adequate phase margin) for higher refclk frequencies.

For all the above reasons, selecting a refclk using the traditional 12 kHz to 20 MHz approach make it increasingly more likely to result in worse system bit error rate performance as time goes by, and often at the expense of higher component cost and system complexity.

We therefore recommend here a more meaningful methodology to evaluate and specify refclk jitter for high-speed SerDes applications. This methodology is based on using relevant application-specific jitter filters and accounting for phase-noise aliasing. Originally developed within the PCI-SIG Electrical Workgroup for PCI Express Revision 5.0, the following section extends this methodology for use in other high-speed interfaces.

## **Recommended Refclk Jitter Methodology**

### **Background and Overall Description**

PCI Express (PCIe) is one of the few standards that explicitly specifies refclk jitter. The PCI Express Base Revision 4.0 Specification [2] published in 2017 inherited the longstanding real-time oscilloscope-based refclk jitter methodology from earlier revisions, but cut the 16 GT/s refclk jitter limit in half to 500 fs rms. In doing so, the noise intrinsic to the oscilloscope was on the order of, or in many cases higher than, the clock and oscillators being tested for PCIe refclk jitter compliance. This spurred IDT, SiLabs and JitterLabs to independently develop alternative methodologies [3-6] to address the intrinsic oscilloscope noise from the refclk jitter measurement. Eventually, PCI-SIG adopted one of JitterLabs' methodologies (described below) into PCI Express Base Revisions 5.0 and 6.0 [7, 8]. An overview of this methodology was published in 2019 as reference [9].

The dominant source of oscilloscope noise is usually ADC quantization noise (i.e. vertical noise) that couples into the measurement when the signal is sampled. To mitigate this, some real-time oscilloscope models today employ 10 or 12 bit ADCs, which dramatically reduce this noise compared to 8 bit ADC models. Regardless, phase noise analyzers remain the tool of choice for clock and oscillator vendors because they have the lowest intrinsic noise floor of any instrument, data acquisition is quick and easy, and post-processing is simple. Also, phase noise data is typically available anyway from clock and oscillator vendors. Thus, vendor-provided PCIe refclk jitter compliance data is typically derived from phase noise rather than oscilloscope data. However, the oscilloscope methodology is still used in applications where the signal must be probed, since probes are not available for direct phase noise analyzer instruments today.

One key difference between oscilloscope and phase noise measurements is how aliased phase noise is captured during the measurement. Real-time oscilloscopes observe aliased phase noise just like the actual application (assuming the signal is bandwidth limited similarly). However, phase noise analyzers include an anti-aliasing filter that prevents aliased noise from being measured. In fact, phase noise analyzers can only observe roughly 30% to 40% away from the fundamental clock frequency (e.g. 30 MHz to 40 MHz offset from a 100 MHz clock signal). Although spectrum analyzers can observe a wider spectrum, they cannot distinguish between amplitude and phase noise and so are not permitted by PCI-SIG to measure PCIe refclk jitter compliance.

Figure 6 illustrates phase noise aliasing in the real application, in which the waveforms are drawn as sine waves for simplicity. The refclk drives the SerDes TX PLL, which multiplies the refclk frequency higher. The first block that the refclk signal enters in the TX PLL is the phase detector. Being a digital block, the phase detector samples the mid-point of both refclk and feedback signals. Their phase difference forms an error signal that the loop minimizes during operation. To avoid duty-cycle distortion jitter, the phase detector typically samples either the rising or falling edge only (i.e., one clock-edge polarity), rather than all clock edges.

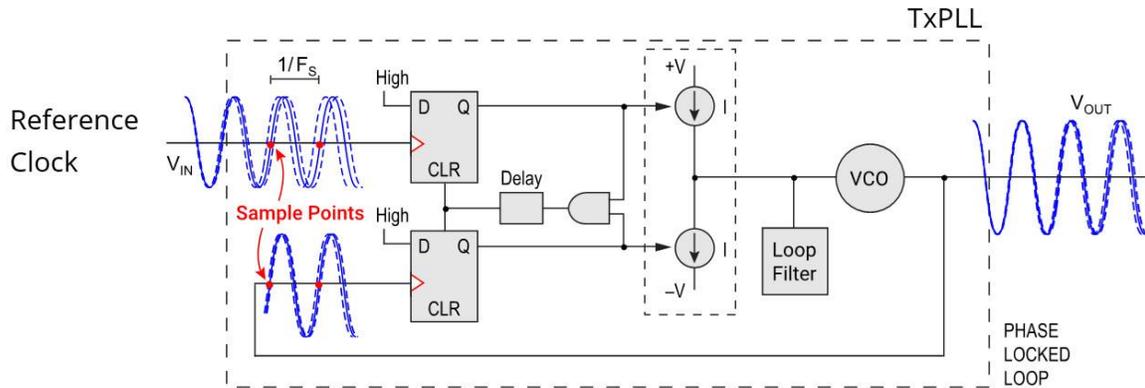


Figure 6. The TX PLL phase detector samples the refclk, causing noise to alias below the Nyquist frequency ( $F_S/2$ ), where  $F_S$  is the sample rate of phase detector. The aliased phase noise is then filtered by the PLL jitter transfer function, and later by the receiver’s observed jitter transfer function.

For example, a 100 MHz refclk would have a phase detector sampling rate of 100 MHz, which has a Nyquist frequency of 50 MHz. In this scenario, phase noise above 50 MHz aliases below 50 MHz. Specifically, the noise spectrum from 50 MHz to 100 MHz offset frequency “folds” (or mirrors) below the 50 MHz offset-frequency spectral boundary. The same occurs at higher Nyquist zones that separated by integer multiples of 50 MHz (e.g. 100, 150, 200, etc.) offset frequency. Meaning, each 50 MHz spectrum-segment mirrors across Nyquist-zone boundaries to end up in the first Nyquist zone (i.e. below 50 MHz).

Like the phase detector, a real-time oscilloscope also samples the midpoint of the refclk waveform to measure jitter. As such, the oscilloscope aliases phase noise just like the real application. But unlike the oscilloscope, a phase noise analyzer cannot measure aliased noise. Thus, we must manually add aliased noise into the phase noise spectrum to more accurately emulate the noise observed by the real system. This is done by extending the last measured phase-noise data point flat to the 3<sup>rd</sup> harmonic, as shown by the flat red line shown in Figure 7.

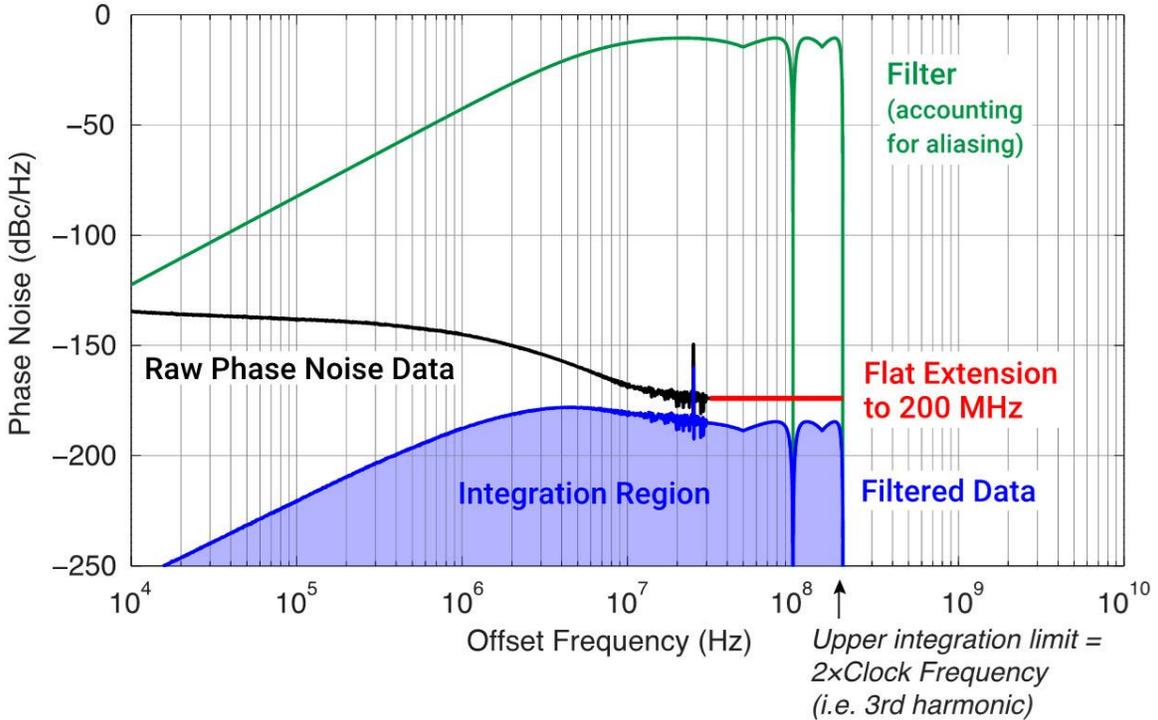


Figure 7. Aliased phase noise is added to the original measured 100 MHz phase noise data (black curve) by extending the last measured phase noise data flat (red curve) to the 3<sup>rd</sup> harmonic. The system filter (green curve) is mirrored across Nyquist-zone boundaries at 50, 100 and 150 MHz offset frequencies, before applying to the measured-plus-extended phase noise data, resulting in the filtered phase noise data (blue curve). The filtered curve is then integrated to compute the refclk contribution to overall phase jitter observed by the system.

The system filter is mirrored across Nyquist zones (green curve) and applied to the measured-plus-extended data to create filtered phase noise data (blue curve). The filtered phase noise is then integrated across offset frequencies up to the 3<sup>rd</sup> harmonic to derive a measure of phase jitter.

## Discussion

Why is the phase noise extended flat to the 3<sup>rd</sup> harmonic? Studies within PCI-SIG’s Electrical Workgroup [10] suggest that extending the measured phase noise data flat to the 3<sup>rd</sup> harmonic (or, twice the fundamental frequency in the offset-frequency axis) accurately estimates worst-case phase jitter across various timing vendors. Beyond this, phase noise tends to roll off quickly and can be ignored. Figure 8 illustrates such a scenario for a timing device with negligible amplitude noise at high offset frequencies (so the spectrum analyzer data can be interpreted as dominated by phase noise). Here, the 3<sup>rd</sup> harmonic appears as a bump at 200 MHz offset. A -10 dB/decade reference line is shown, which has the unique property [11, 12] that its intersection when lowered onto the phase noise curve identifies the offset region that dominates the phase jitter value. Beyond the 3<sup>rd</sup> harmonic, the phase noise rolls off and doesn’t contribute significantly to phase jitter.

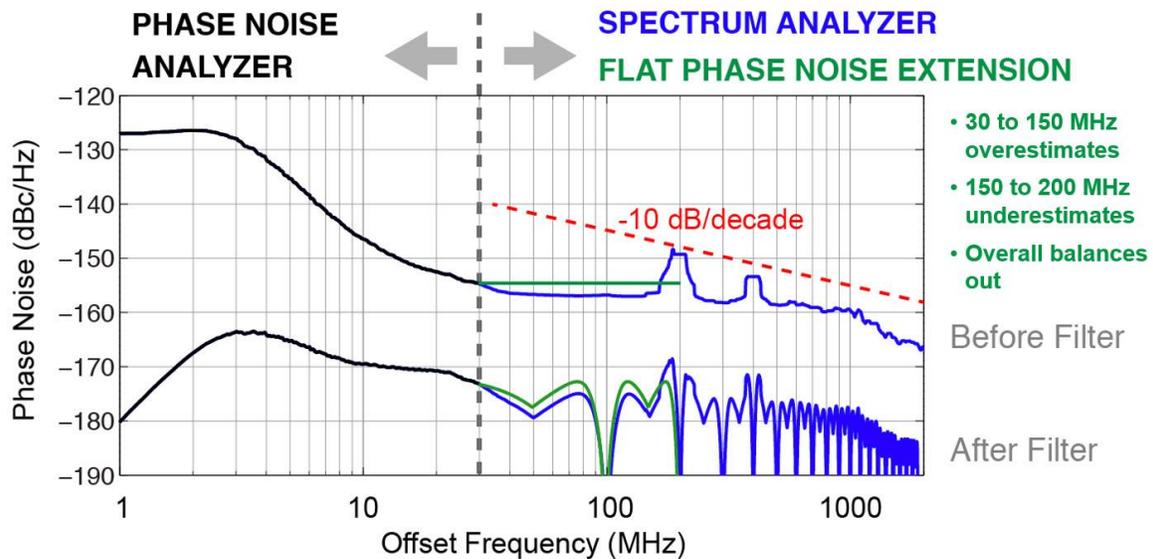


Figure 8. Illustration how phase noise rolls off above the 3<sup>rd</sup> harmonic. This is for a 100 MHz refclk device known to have negligible amplitude noise.

In addition to the above empirical analysis, the data converter industry has been using the 3<sup>rd</sup> harmonic phase noise integration limit since at least 2008. In theory, the upper integration limit for analog-to-digital converters is set by the encode bandwidth [13], which can extend well beyond the 3<sup>rd</sup> harmonic. In practice [14, 15], this limit is much lower than the encode bandwidth since there is very little phase noise spectral energy density past the 3<sup>rd</sup> harmonic. This was validated by deriving phase jitter ( $t_{jitter}$ ) from independent measurements of signal-to-noise ratio (SNR) using the following equation.

$$SNR_{Ideal} = 20 \log_{10} \left( \frac{1}{2\pi f t_{jitter}} \right)$$

Extending phase noise data flat to the 3<sup>rd</sup> harmonic and integrating it provides a value of phase jitter that correlates well with “golden” phase jitter derived from these independent SNR measurements.

One might ask why the flat extension (green line) ends exactly at the 3<sup>rd</sup> harmonic in Figure 8 and doesn't include energy from the entire harmonic (that is, the right half of the 3<sup>rd</sup> harmonic bump). This is simply because the flat extension only extends far enough to accurately estimate the true phase jitter value (obtained via SNR measurements as discussed above, or in the case of the PCI-SIG work, jitter measurements obtained with a real-time oscilloscope minus the intrinsic jitter introduced by the oscilloscope [5]). Ending the phase noise extension at some offset frequency before or after the 3<sup>rd</sup> harmonic simply results in a less accurate correlation to the true phase jitter that was independently measured.

Note that this methodology is not perfect, however. In general, it provides an accurate estimate of phase jitter for the majority of clock and oscillator devices on the market. In practice, some devices' phase noise may roll off earlier than the 3<sup>rd</sup> harmonic, resulting in overestimating phase jitter. Nevertheless, this methodology may be used to design robust systems.

Another interesting aspect of this methodology is that there are two ways to filter aliased phase noise. One method, as illustrated in Figures 7 and 8, is to mirror the system filter higher across multiple Nyquist-zone boundaries until reaching the 3<sup>rd</sup> harmonic. Alternatively, the extended phase noise can be mirrored lower across the same Nyquist-zone boundaries until falling within the first Nyquist zone. See Figure 2 in reference [9] for more details. Figure 9 illustrates both approaches. While both are mathematically equivalent, the left chart in Figure 9 is perhaps more insightful since the entire spectrum is shown. Aliasing is observed in the right chart where the filtered trace (green curve) has higher phase noise than the unfiltered trace (orange curve) above 10 MHz offset.

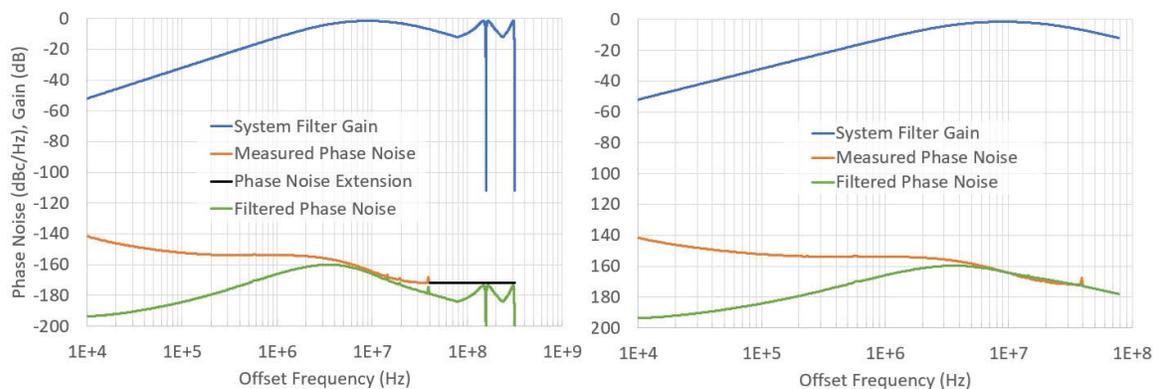


Figure 9. Illustration of two equivalent processes to filter aliased phase noise in a 156.25 MHz refclk. The left chart extends (black) the measured phase noise (orange) to the 3<sup>rd</sup> harmonic (312.5 MHz), mirrors the filter (blue) across higher Nyquist zones (78.125, 156.25, 234.375 MHz) before deriving the filtered phase noise (green). Alternatively, the right chart aliases the extended phase noise (not shown) below the Nyquist frequency (78.125 MHz) before filtering (green). Both filtered curves produce the same phase jitter value.

## Short-hand Notation

Figure 10 presents a flexible short-hand notation for describing this filter methodology using “#-#A”. Here, the first and second numbers “#” are replaced with RX CDR and TX PLL bandwidths, respectively, and assume first-order (20 dB/decade) roll offs unless specified otherwise. The letter “A” is explicitly noted to indicate that aliasing is included in the computed phase jitter value. For example, “4-16A” phase jitter denotes 4 MHz RX and 16 MHz TX bandwidths with aliasing. Here, 4 MHz represents the most common serial standard, IEEE 802.3 Ethernet, which typically specifies a 4 MHz CDR bandwidth for 10 Gbps and higher link rates, and 16 MHz represents a worst-case estimate for TX PLL bandwidth (the PLL becomes unstable at higher bandwidths when driven by a refclk frequency of 156.25 MHz). This short-hand notation makes it easy to communicate variations. For example, “2-10A” phase jitter describes the same methodology but for 2 MHz RX CDR and 10 MHz TX PLL bandwidths. In practice, the exact bandwidths are application dependent. Overall, the “4-16A” designation, being the most common, would be a good choice to replace the de-facto “12 kHz to 20 MHz” brick wall filter in new clock and oscillator datasheets.

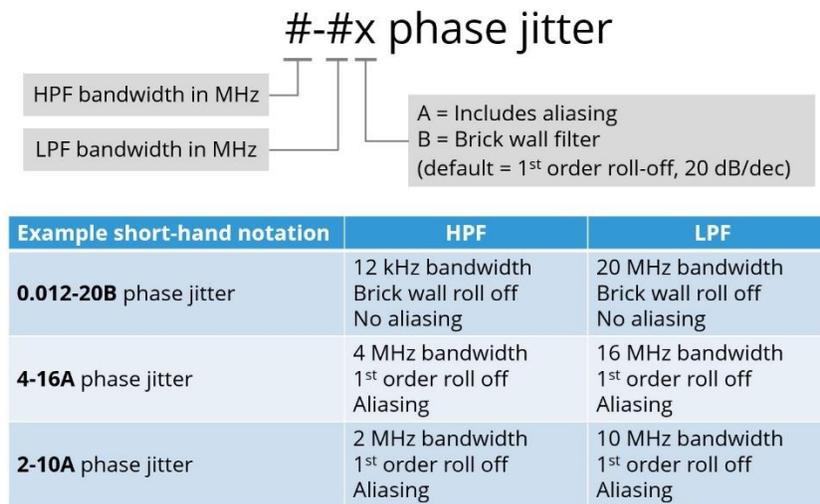


Figure 10. Illustration of short-hand notation for use with the recommended refclk jitter methodology. This notation provides a flexible framework to easily communicate filter characteristics, of which three examples are shown here.

## Applicability

The recommended refclk jitter methodology may be applied to any high-speed serial link. Also, any clocking architecture (embedded clock, common clock, independent clock, etc.) is supported by applying the appropriate system transfer function, to emulate the overall system filtering effect on the refclk phase noise. While currently used in a few standards (those that specify refclk jitter, namely: PCIe, CXL, CK440), the intent here is to extend this methodology to cover all other serial standards. For example, standards associated with Ethernet, Fibre Channel, USB, OIF-CEI, and so on.

## Summary and Example Application

The following summarizes a procedure to evaluate refclk phase jitter using the recommended methodology:

1. Measure a device with a direct phase-noise analyzer instrument (not a spectrum analyzer or oscilloscope). For example, using Keysight E5052B or Rhode & Schwarz FSWP (discarding spectrum analyzer data collected by FSWP, which occurs at offsets above 30% of the clock frequency).
2. Extend flat the measured phase noise data to the third harmonic.
  - a. Note that the third harmonic in the signal spectrum equals twice the clock frequency in the offset-frequency spectrum.
3. Account for aliasing by folding the phase noise data into the first Nyquist zone.
  - a. Alternatively, as shown in Figures 6 and 8, mirror the system filter across Nyquist-zone boundaries up to the 3<sup>rd</sup> harmonic.
  - b. Note that this process is patented by JitterLabs LLC [16]. However, free public software is available with this IP embedded to enable widespread adoption – see reference [17] for example.
4. Apply the system jitter transfer function to the measured-plus-extended phase noise data.
  - a. Model the system transfer function as accurately as possible (e.g. include filter roll offs) to extract the relevant phase noise observed by the system.
  - b. Typically, industry standards define the RX CDR filter and the SerDes vendor's silicon defines the worse-case TX PLL filter.
  - c. For a conservative estimate, use the highest expected TX PLL bandwidth across process, voltage, and temperature.
5. Integrate the filtered phase noise across offset frequency to derive a value of phase jitter.
  - a. Begin the integration at least an order of magnitude below the lowest filter corner frequency. It's OK to integrate to lower offset frequencies, but the resulting phase jitter is unlikely to change.
6. For consistency, report (or specify) refclk phase jitter using the template in the next section.

As a practical example, Figure 10 compares traditional versus recommended refclk jitter methodologies for two example oscillators. The top chart, based on the traditional 0.012-20B phase jitter methodology, clearly favors the mid-offset frequency region of the phase noise plot. The bottom chart, based on the 4-16A phase jitter methodology, favors a dramatically higher-offset frequency region. In this example, Device A fares better in the traditional methodology with 20 fs rms less phase jitter, whereas Device B fares better in the recommended methodology with 49 fs rms less phase jitter.

It's interesting to note that the different refclk jitter methodologies shown in Figure 10 arrive at opposite conclusions. However, this is not always the case, as the phase noise curvature of the two devices, in combination with the jitter filter for each methodology, dictates the resulting phase jitter. Regardless, the 4-16A phase jitter methodology will always more accurately predicts the refclk jitter observed by the system because:

1. The application-specific filter is applied, including roll offs.
2. Phase-noise aliasing is included.

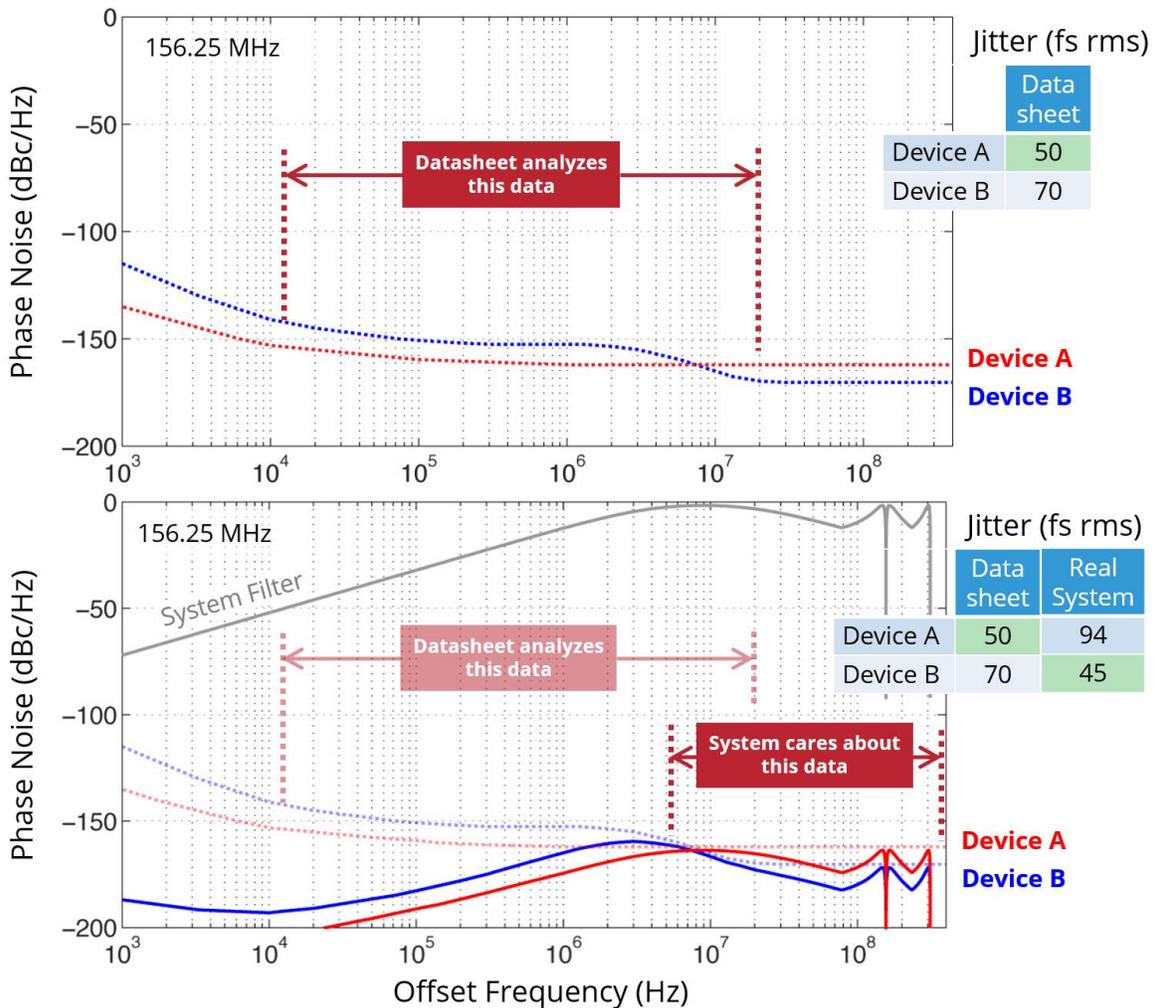


Figure 10. Comparison of traditional 12 kHz to 20 MHz brick wall filter analysis (top chart) versus the recommended 4-16A phase jitter analysis (bottom chart). It's interesting to note that the two methodologies arrive at opposite conclusions for the example devices shown. For the reasons outlined here, the 4-16A analysis more accurately predicts the in-system performance of the two devices.

## Why to Avoid Phase Noise Masks

Sometimes SerDes vendors specify phase noise masks instead of phase jitter values. This methodology seems attractive since one simply needs to select a refclk device whose phase noise falls below the mask. However, while this simplifies analysis by avoiding the filtering step, it includes several unintended consequences that can result in non-optimal system performance. Consider Figure 11 below comparing 3 different oscillators' phase noise against a SerDes vendor's compliance mask. Which device would you select?

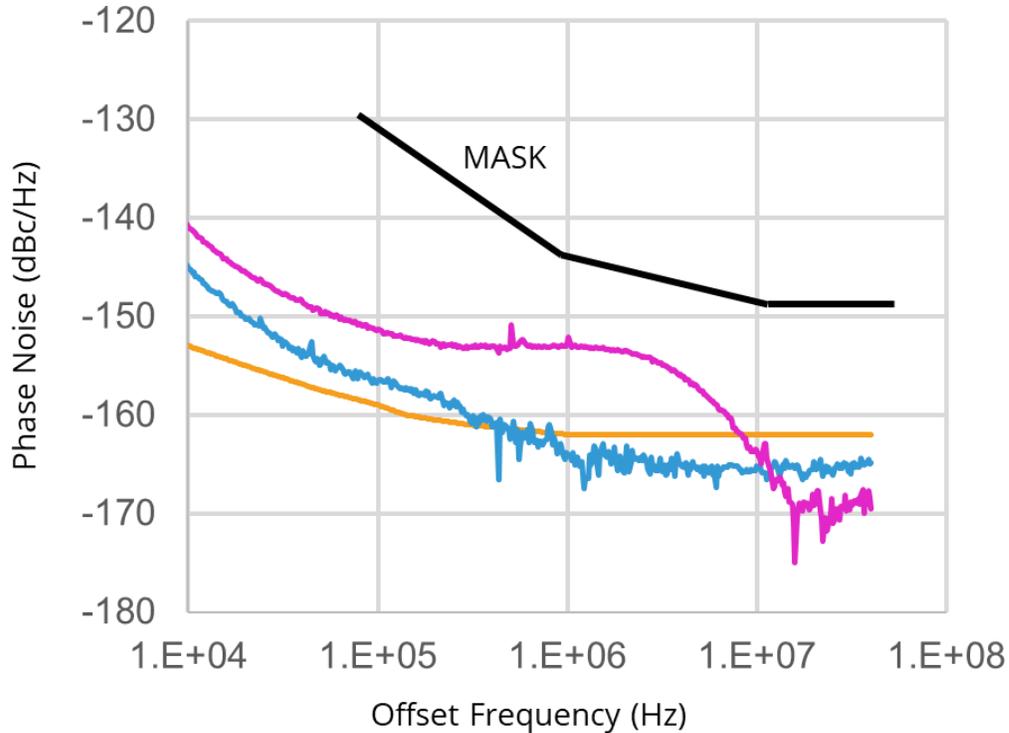


Figure 11. Three timing devices' phase-noise characteristics are compared against a phase-noise compliance mask. Which product would you select for your design? Counter-intuitively, the device with the least margin to the mask is observed by the system to have the lowest jitter. For this reason, we recommend SerDes datasheets specify phase jitter values after appropriately filtering as discussed herein.

Although all 3 devices comply with the mask, which one performs best in the application? Most people will simply do a visual inspection to eliminate devices having the least margin to the mask. In doing so, this “spot analysis” reduces each curve to a single point that has the least margin to the mask. However, the system does not observe a single point, but rather applies a filter and integrates the resulting phase noise. Figure 12 summarizes the results.

Product	“0.012-20B” Phase Jitter	“4-16A” Phase Jitter	Mask Margin
Product A	37 fs	67 fs	17 dB
Product B	51 fs	94 fs	12 dB
Product C	73 fs	54 fs	9 dB

Figure 12. Analysis of 3 timing devices’ performance versus the phase-noise compliance mask shown in Figure 11. A “spot-frequency” analysis shows Product C performance has the least margin to the mask. But considering the entire spectrum and system filtering effects, Product C adds the least jitter to the system.

In this example, Product A has the lowest 0.012-20B phase jitter and highest margin to the mask. Many customers will select it on either basis. However, Product C has the lowest 4-16A phase jitter, which is what the system observes in practice. However, it also has the worst margin to the mask.

The key point is that while a mask simplifies compliance analysis, it cannot be used to select devices for optimal link performance. To do this, one must evaluate phase noise as the system does by applying the appropriate filters and integrating to obtain a phase jitter value. Since this phase jitter value is what the application observes, it makes the most sense to specify it in SerDes datasheets.

Another issue is the phase noise mask is often created without accounting for the characteristic phase-noise “hump” associated with PLL-based clocks and oscillator devices. Doing so eliminates PLL-based devices in favor of non-PLL based solutions such as fundamental or 3<sup>rd</sup> overtone crystal-based devices. This is an unintended consequence since SerDes vendors should be motivated to enable as many refclk vendors (and fabrication technologies) as possible for their customer base. By specifying a phase noise mask, they potentially eliminate a large portion of the market that can outperform “compliant” devices, as illustrated in Figures 11 to 12.

## Conclusions

Established in 1992 for SONET OC-48 telecom line rates, the traditional 12 kHz to 20 MHz brick wall jitter filter served as a golden reference to evaluate refclk jitter for over 30 years. Today, this filter is used in nearly all clock and timing datasheets. It's popularity has further spread into most SerDes datasheet refclk-jitter specifications, to simplify customer adoption. However, the results it provides no longer correlate with system performance and can create sub-optimum link performance. Sources of error include incorrect corner frequencies, unrealistic brick-wall roll offs and a lack of accounting for aliased phase noise. Errors of tens of femtoseconds are significant today and will become more significant tomorrow as data rates increase.

We therefore recommend the industry adopt a more accurate refclk jitter methodology. This methodology, established by the PCI-SIG Electrical Workgroup in 2018 [6-10], is applied here to other high-speed standards. This methodology is based on:

- specifying measurements using a phase-noise analyzer
- including aliased phase noise
- specifying meaningful (application specific) jitter filters, including
  - corner frequencies
  - roll off

Figure 13 provides a template for vendors to specify this methodology in datasheets for SerDes, clock and oscillator components. The highlighted text is product specific, and the values shown are for illustration only.

**Table x: Refclk Jitter – 156.25 MHz**

	Parameter	Symbol	Max	Unit	Condition
Traditional →	0.012-20B Phase Jitter	J_clk_legacy	xxx	fs rms	Integrated phase noise from 12 kHz to 20 MHz offset.
Recommended →	4-16A Phase Jitter	J_clk_416a	xxx	fs rms	Measured with phase noise analyzer, extending (flat) phase noise to 3rd harmonic (i.e., 312.5 MHz offset), folding phase noise below the Nyquist frequency (i.e., 78.125 MHz offset), filtering and integrating from 10 kHz to Nyquist. Filter comprises 4 MHz low pass and 16 MHz high pass cut off frequencies, each with 20 dB/dec roll off.

For adoption, SerDes datasheets simply need to update the highlighted text in Figure 11

Figure 13. Framework (or, template) for specifying the recommended refclk jitter methodology in a datasheet (either SerDes, clock or oscillator). The first row specifies traditional refclk jitter as commonly written today. The second row specifies the recommended methodology for refclk jitter. The highlighted text is provided for example only and must be updated as appropriate for the actual product. Although both specifications are shown in this template, we recommend the industry transition to specify only the recommended methodology to establish a new de-facto standard. Adoption and alignment between SerDes, clock and oscillator vendors will ensure easy refclk selection and optimum link performance.

according to the relevant application requirements (RX CDR) and SerDes silicon (TX PLL) characteristics.

Clock and oscillator products have more generic datasheets that sell into a wide variety of SerDes applications. It's common for these components to specify a single datasheet

phase jitter value, which serves as a proxy for performance across all SerDes applications. In this case, we recommend replacing the traditional 0.012-20B phase jitter specification with the 4-16A phase jitter specification. This enables customers to accurately estimate in-system performance even if the system filters are slightly different. For example, if the actual system is best emulated using 2-14A phase jitter, the 4-16A phase jitter value is still a better proxy for performance than the traditional 0.012-20B value.

Free public software [17] is available to automate analysis using this methodology.

By adopting the recommended refclk jitter methodology in datasheets, SerDes and clock/oscillator vendors are focusing their business on the performance that matters. Furthermore, alignment between vendors will establish a new, more meaningful, de facto refclk jitter standard than the current 12 kHz to 20 MHz methodology. Overall, this advances the industry to more easily evaluate and select the best refclk for an application, to ensure optimal link performance for SerDes vendors and their customers.

## References

- [1] Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria,” Telcordia GR-253-CORE, Issue 1 (1994).
- [2] PCI Express Base Specification, Rev. 4.0, Ver. 1.0 (PCI-SIG, 2017), [www.pcisig.com](http://www.pcisig.com).
- [3] "PCI Express Measurement Techniques for Gen5 and Beyond White Paper," R. Wade, J. Hsu, J. Tajnai, J. Bal, IDT (Jan. 9, 2018), [www.renesas.com](http://www.renesas.com).
- [4] "Refclk Fanout Best Practices for 8GT/s and 16GT/s Systems," G. Richmond, SiLabs, PCI-SIG Developers Conference in Santa Clara, CA (June 7, 2017), [www.pcisig.com](http://www.pcisig.com).
- [5] ”[Removing Oscilloscope Noise from RMS Jitter Measurements](#),” G. Giust, F. Benford, Technical Note 5, JitterLabs (July 25, 2017), republished as AN10074 at [www.sitime.com](http://www.sitime.com).
- [6] "Methodologies for PCIe5 Refclk Jitter Analysis," G. Giust, PCI-SIG Electrical Workgroup Meeting (Jan. 19, 2018), [www.pcisig.com](http://www.pcisig.com).
- [7] PCI Express Base Specification, Rev. 5.0, Ver. 1.0 (PCI-SIG, 2019) , [www.pcisig.com](http://www.pcisig.com).
- [8] PCI Express Base Specification, Rev. 6.0, Ver. 1.0 (PCI-SIG, 2022), [www.pcisig.com](http://www.pcisig.com).

- [9] G. Giust, "[How to evaluate reference-clock phase noise in high-speed serial links](#)," Signal Integrity Journal online (2019), [www.signalintegrityjournal.com](http://www.signalintegrityjournal.com).
- [10] "Study to Determine a Phase-noise Methodology that Matches Jitter Measurements from a Scope," G. Giust, rev. 6, PCI-SIG Electrical Workgroup Meeting (May 24, 2018), [www.pcisig.com](http://www.pcisig.com)
- [11] "[Determine the Dominant Source of Phase Noise by Inspection](#)", JitterLabs (2016), republished as AN10072, [www.sitime.com](http://www.sitime.com) (2021).
- [12] "Determine the Dominant Source of Phase Noise by Inspection," SiTime (2022) (<https://www.youtube.com/watch?v=2eIHk3v45Pk>)
- [13] "[Sampled Systems and the Effects of Clock Phase Noise and Jitter](#)," B. Brannon, app note N-756, Analog Devices, (2004).
- [14] "[Analog-to-Digital Converter Clock Optimization, a Test Engineering Perspective](#)," R. Reeder, W. Green, R. Shillito, Analog Dialogue, 42-02, Analog Devices (Feb. 2008).
- [15] "[Interfacing to Data Converters](#)," section 6.5 (Sampling Clock Generation), W. Kester, Analog Devices
- [16] "Method and apparatus for analyzing phase noise in a signal from an electronic device," JitterLabs LLC, U.S. Patents 10,802,074 (2020), 11,231,459 (2022), 11,592,480 (2023).
- [17] "SiTime Studio" desktop software, available for free at <http://www.sitime.com/sitime-studio>.

## **Acknowledgements**

All opinion, judgments, and recommendations presented herein are the opinions of the author and do not necessarily reflect on the opinions of the PCI-SIG®.

PCI, PCI Express, PCIe, and PCI-SIG are trademarks or registered trademarks of PCI-SIG.