

Phase Noise and Jitter Requirements for Serial IO Applications

1 Introduction

Historically, the clock phase jitter in the frequency range of 12 kHz to 20 MHz is used to specify the clock oscillator or clock generator device phase jitter. This frequency range is the legacy SONET / SDH applications, which have been dominant in telecom industry 20 to 30 years ago. Such specification, however, is not relevant to a significant portion of today's widely used application and systems, such as asynchronous data communication standards, including PCI-Express, Ethernet, FibreChannel, SATA, etc. Using this specification blindly causes some system designers to overlook many other options that easily meet their application requirements and may provide them with significant advantage in other areas, such as power, size, delivery time, price, control options, programming flexibility, etc. This application note reviews the phase jitter requirement for several applications and shows why the phase jitter in 12 kHz to 20 MHz is not relevant for those applications and how to analyze their phase jitter requirements.

The remainder of this document is organized as follow. Section 2 provides a short overview of relationship between phase noise and time interval error (TIE), and phase jitter. Section 3 describes phase jitter filters applicable to asynchronous serial standards, while Section 4 provides jitter budget estimation analysis for such applications. Jitter budgets for various serial IOs when using SiTime high performance oscillators as reference clock are provided in Section 5.

2 Phase Noise and Time Interval Error (TIE) Relationship with Jitter

Let us assume that a clock signal or its first harmonic can be specified with a sinusoidal waveform $v(t)$, as defined below:

$$v(t) = A \cdot \cos(2\pi f_c \cdot t + \varphi_n) \quad \text{Equation 1}$$

where f_c is the clock frequency, and φ_n is the noise of the phase component. The noise φ_n causes noise in the signal $v(t)$; The spectrum of the normalized amplitude noise around the carrier frequency is referred to as "phase noise". On the other hand, the spectrum of the φ_n is referred to as "phase noise spectrum" or "phase jitter spectrum". Both "phase noise" and "phase noise spectrum" are expressed in dBc/Hz unit versus offset frequency relative to the carrier frequency f_c . Strictly speaking, "phase noise" and "phase noise spectrum" are not the same because one represents the noise in the signal amplitude and the other the noise in the signal phase, but for most practical applications the two are very close and can be used

interchangeably, especially at offset frequencies greater than 100Hz or so. We denote phase noise spectrum by $PN(f)$ for the remainder of this document.

Different types of jitter specifications can be derived from the phase noise spectrum by using the following formula:

$$\text{RMS Jitter (filtered)} = \frac{\sqrt{2 \int_{f_1}^{f_2} \left(10^{\frac{PN(f)}{10}} \right) |H(f)|^2 df}}{2\pi f_c} \quad \text{Equation 2}$$

The jitter filter response $H(f)$ and the integration range limits f_1 and f_2 are dependent on the type of the jitter and the specific applications.

Time interval error (TIE) is defined as the deviation of signal edges from their idea positions in time, as illustrated in the Figure 1.

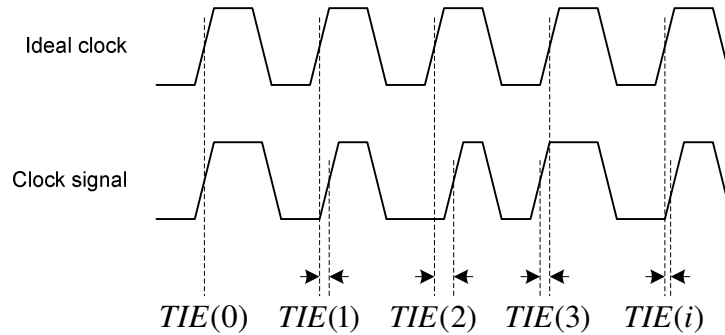


Figure 1: TIE definition

With a close look at TIE definition, one can see that the TIE sequence represents the phase variations of the signal sampled at the signal threshold-crossing; in other words, it is equivalent to the phase jitter signal, φ_n being sampled with a sampling frequency of the carrier, f_c , but expressed in the time unit. Equation 3 shows the relationship between phase noise signal and TIE:

$$TIE(i) = \frac{\varphi_n(i.T_c)}{2\pi f_c} \quad \text{Equation 3}$$

$$PN(f) = 20 \cdot \log_{10} (2\pi f_c \cdot F\{TIE\})$$

where $T_c = 1/f_c$ is the ideal signal period, and $F\{TIE\}$ is the Fourier transform of the TIE sequence normalized to 1Hz. In practice, f_c and T_c are the average frequency and period of the signal, respectively. Since TIE and φ_n are related with a constant factor, much of the jitter specifications can also be computed from TIE by combining Equation 2 and Equation 3 to obtain the following formula:

$$\text{RMS Jitter (filtered)} = \sqrt{2 \int_{f_1}^{f_2} |F\{TIE\}|^2 \cdot |H(f)|^2 \cdot df} \quad \text{Equation 4}$$

where $F\{TIE\}$ is the Fourier transform of the TIE sequence.

The sampled nature of TIE sequence means that TIE spectrum is subject to aliasing, which results in phase noise components at frequency offsets above half the sampling rate, i.e., $f_c/2$ to alias to frequency range between 0 and $f_c/2$. In other words, the high frequency phase noise spectrum components which may not contribute to jitter when using phase noise spectrum (Equation 2), will contribute to jitter when using TIE (Equation 3). In digital applications, TIE is a more appropriate sequence for estimating jitter because those circuits sample the phase at clock edges, which means they are subject to aliasing intrinsically. However, in some applications, the high frequency phase noise is typically lower than the low frequency components and as such do not contribute to jitter significantly. This allows using either phase noise or TIE for measuring jitter without significant error.

TIE jitter is often defined as the statistics of the $TIE(i)$ sequence. It can be shown that the RMS value of $TIE(i)$, $i = 1, \dots, M$, can also be computed from Equation 3 by choosing $H(f) = 1$. However, TIE jitter is usually not relevant in most case because $H(f) \neq 1$ for majority of practical applications, such as high-speed serial interfaces. We will review the jitter filter response $H(f)$ for high-speed serial data communication sections in section 3.

3 High-Speed Asynchronous Serial Data Communication (Serial IO)

Many asynchronous data communication standards and methodologies have been developed, each crafted for different segments of market, such as computing, storage, telecom, and consumer products. Two clocking methods are primarily used in serial IO applications, embedded clocking and common reference clocking, which are described in the following sections. A list of major serial IO standards and their clocking method are given in Table 1.

Table 1: Major serial IO applications

Serial IO standard	General applications	Clocking method
FibreChannel	Storage	Embedded
Serial ATA (SATA)	Storage	Embedded
Serial SCSI (SAS)	Storage	Embedded
PCI-Express	Computing	Common RefClk, embedded
Serial Giga-bit Ethernet (GbE), GPON	Networking	Embedded
Serial 10Giga-bit Ethernet (10GbE)	Networking	Embedded

Infiniband	Servers	Embedded
XAUI	Networking	Embedded
RapidIO	Computing	Embedded
Fully-buffered DIMM (FBDIMM)	Computing	Common RefClk, embedded
IEEE1394b (Firewire)	General	Embedded
USB2.0	General	Embedded
USB3.0	General	Embedded

3.1.1 Embedded Clocking

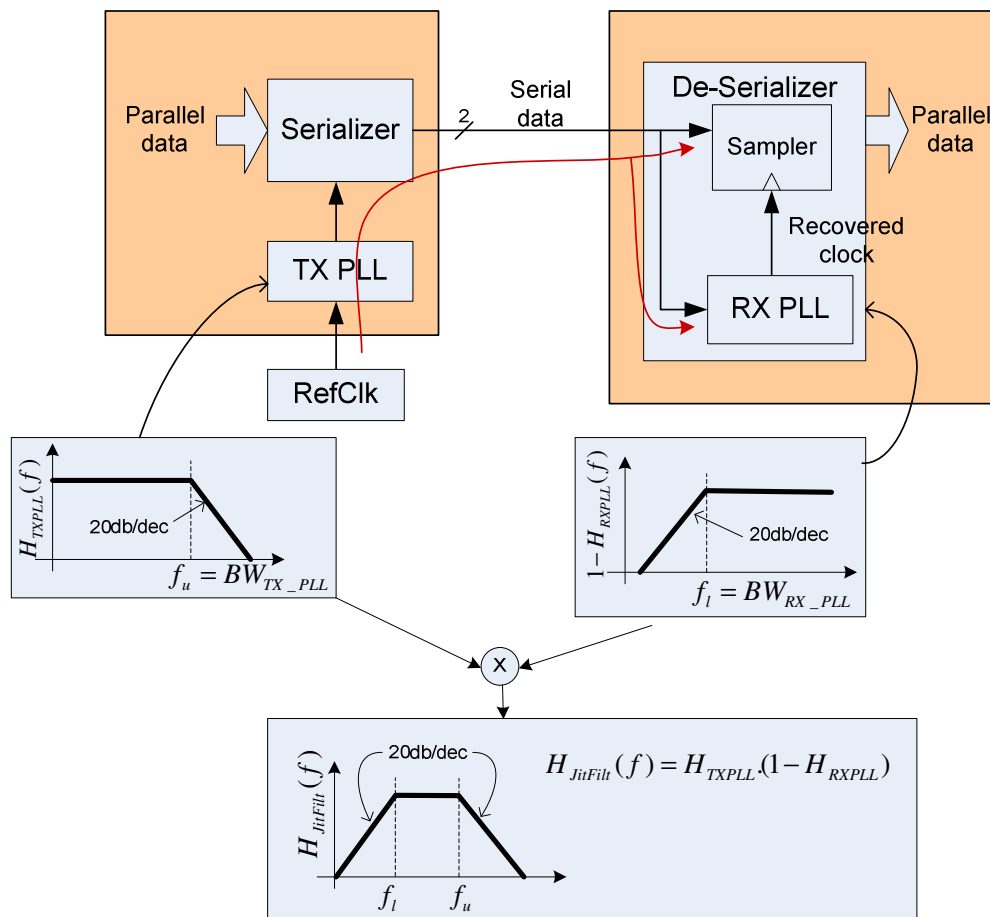


Figure 2: Embedded clocking methodology

The embedded clocking method relies on serial data coding schemes to ensure sufficient transition density in the data stream. The receiver typically employs a clock/data recovery (CDR) circuit which uses the data transitions to recover clock and sample the data in middle of a bit optimally. Most CDR circuit use phase-locked loops (PLL) to recover the clock. On the

transmitter side, typically another PLL is used to synthesize a high-speed clock for clocking out the data from a local reference clock.

The transmitter PLL has a wide bandwidth to minimize the impact of the transmitter PLL's often noisy VCO on the output jitter. The maximum TX PLL bandwidth is roughly 1/10th of the reference clock to allow proper operation of the TX PLL, i.e:

$$BW_{TX-PLL} = \frac{f_{RefClk}}{10} \quad \text{Equation 5}$$

For example, a popular reference clock for serial ATA (SATA) application is 75MHz, which leads to the maximum TX PLL bandwidth of 7.5MHz. The TX PLL passes the reference clock phase jitter from low frequency offsets up to its bandwidth, but attenuates the clock phase noise impact at higher frequency offsets based on the PLL frequency response (20dB/dec for second order response). The TX PLL may have second or higher order response, but it is typically assumed to be second order for more conservative jitter estimates. The overall impact of TX PLL is a low-pass filtering of the clock reference phase noise, as shown in Figure 2.

At the receiver CDR circuit, the RX PLL bandwidth is typically much smaller than the TX PLL. The RX PLL bandwidth is selected low enough to average the effect of irregular data transition sufficiently and reduce the jitter that such irregularities may cause at the RX PLL output. Simultaneously, the RX PLL bandwidth should be as high as possible to allow the receiver to track the jitter of the input signal and improve the overall jitter margins. FibreChannel standard is one of the earliest standards to analyze the optimal selection of the RX PLL bandwidth and concluded that best choice is as specified below:

$$BW_{RX-PLL} = \frac{f_{baud}}{1667} \quad \text{Equation 6}$$

where f_{baud} is the electrical data rate in bit-per-second (bps). Many other standards that use embedded clocking incorporated the FibreChannel recommendations and use the same RX PLL bandwidth relationship. The overall impact of the RX PLL on reference clock jitter is high-pass behavior. Again, the PLL is assumed to have second order behavior, leading to the response shown in Figure 2.

Some standards, such as High-Speed USB2.0, specify jitter specs based on jitter overall a number of cycles. It can be shown that such specification can be mapped to a similar response and a first-order high-pass filter; the derivation of such relationship is beyond the scope of this application note.

Table 2 lists most popular serial IO standards and their associated RX PLL and TX PLL bandwidths.

Table 2: Upper and lower corner frequencies of clock jitter filter response for different serial IO applications

Serial IO standard	Most popular reference clock frequencies (MHz)	Signaling rate (Gbps)	Maximum RX PLL bandwidth	Minimum TX PLL bandwidth
FibreChannel	106.25	1.0625 2.125 4.25	637 kHz 1.275 MHz 2.55 MHz	10 MHz
Serial ATA (SATA) & Serial SCSI (SAS)	37.5 75 120 150	1.5 3 6	900 MHz 1.8 MHz 3.6 MHz	4 MHz 7.5 MHz 12 MHz 15 MHz
Serial Giga-bit Ethernet (GbE)	25 125	1.25	637 kHz *	2.5 MHz 12.5 MHz
Serial 10Giga-bit Ethernet (10GbE)	161.1328 156.25	10.3125 3.125 (x 4 lanes)	4 MHz 1.875 MHz	20 MHz * 20 MHz *
Infiniband	100 200	2.5 5	1.5 MHz 3 MHz	10 MHz 20 MHz
XAUI	156.25	3.125	1.875 MHz	20 MHz *
RapidIO	100	0.5 1 2.5 3.125	0.3 MHz 0.6 MHz 1.5 MHz 1.875 MHz	10 MHz * 20 MHz *
IEEE1394b (Firewire)	98.304	0.49152 0.98304 1.96608	295 kHz 590 kHz 1.19 MHz	10 MHz
USB2.0	12 24 48	0.48	500 kHz	5 MHz *
USB3.0	50 250	5	3 MHz	5 MHz 25 MHz
* These values follow Equation 5 and Equation 6 only approximately; they have become more established limits in the industry.				

3.1.2 Common Reference (RefClk) Clocking

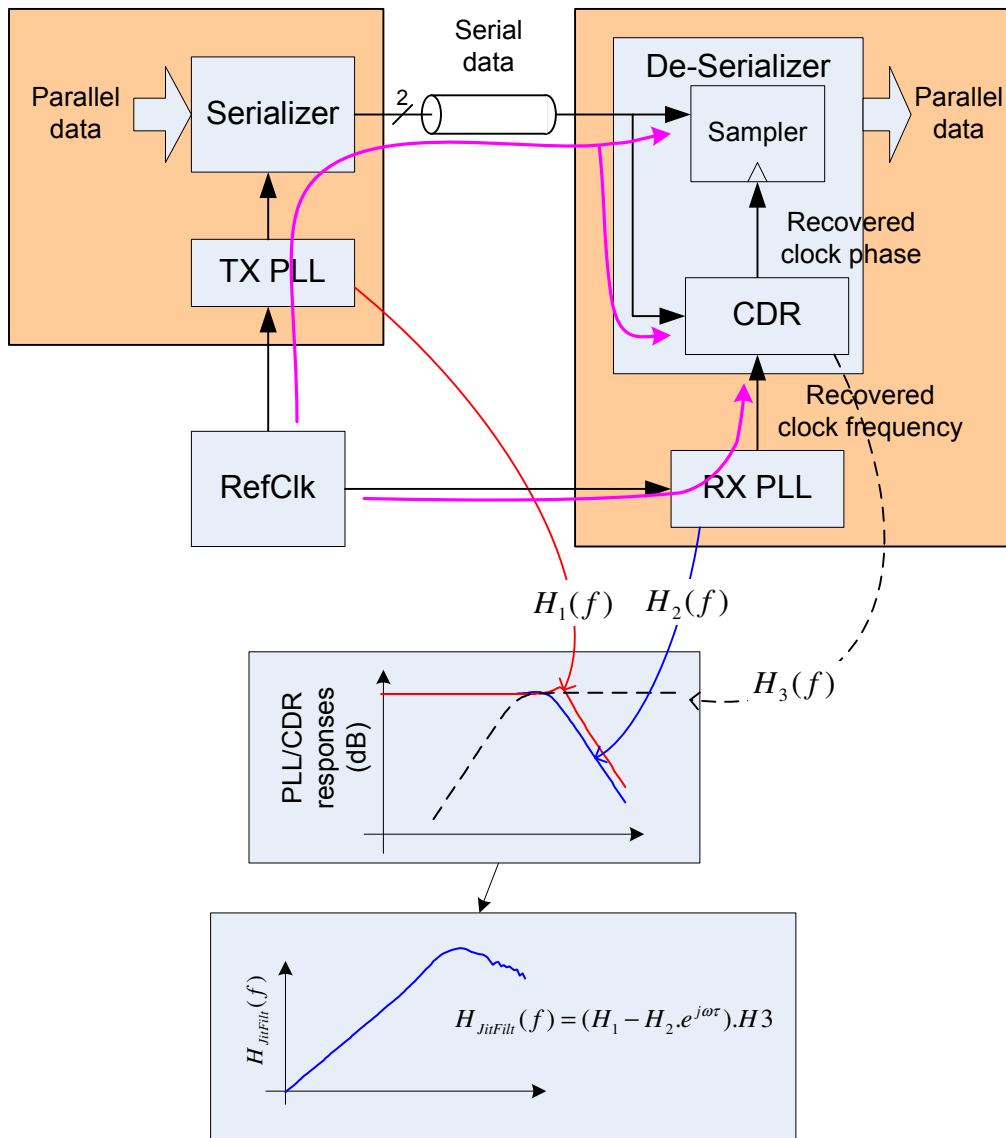


Figure 3: Reference clocking for serial IOs

Another clocking methodology for serial interfaces routes the transmitter reference clock (RefClk) to the receiver. The transmitter includes a PLL to generate high frequency signal for clocking out serial data. Such a PLL is modeled as a second order system with a given response peaking. Receiver uses another PLL to synthesize the correct data sampling clock frequency from the reference clock and employs one clock/data recovery (CDR) circuit for each receiver lane to align locally synthesized clock edge with the center of data stream bits. The common RefClk clocking at the receiver replaces the CDR PLLs per lane with a single PLL for all lanes and one DLL per lane. This can reduce the receiver silicon area and power. The primary examples of standards that use reference clocking are PCI-Express and Fully-buffered DIMM (FBDIMM). This architecture attenuates the impact of phase noise spectrum at clock low

frequency offsets (frequency range that fall in the pass-bands of the TX PLL and RX PLL) significantly because such noise is common in the transmitter and receiver and is tracked closely by the receiver PLL. The dominant noise is concentrated in the region that fall in the pass-band of the TX PLL, but in the stop-band of the receiver PLL. The phase noise that falls in stop-bands of both PLLs is also attenuated significantly.

3.1.3 Estimating Clock Jitter for Different Serial IOs

As explained in previous sections, the system jitter filter responses for the reference clocks in both common RefClk clocking and embedded clocking strategies have a bandpass behavior. Such filters attenuate the impact of phase noise at low and high frequency offsets. Figure 4 shows examples of simulated phase noise profiles for the two 75MHz reference oscillator and filtered phase noise response for SATA-1.5Gbps (900kHz to 7.5MHz) applications. This figure clearly shows how the impact of low and high frequency phase noise is attenuated by the application jitter filter.

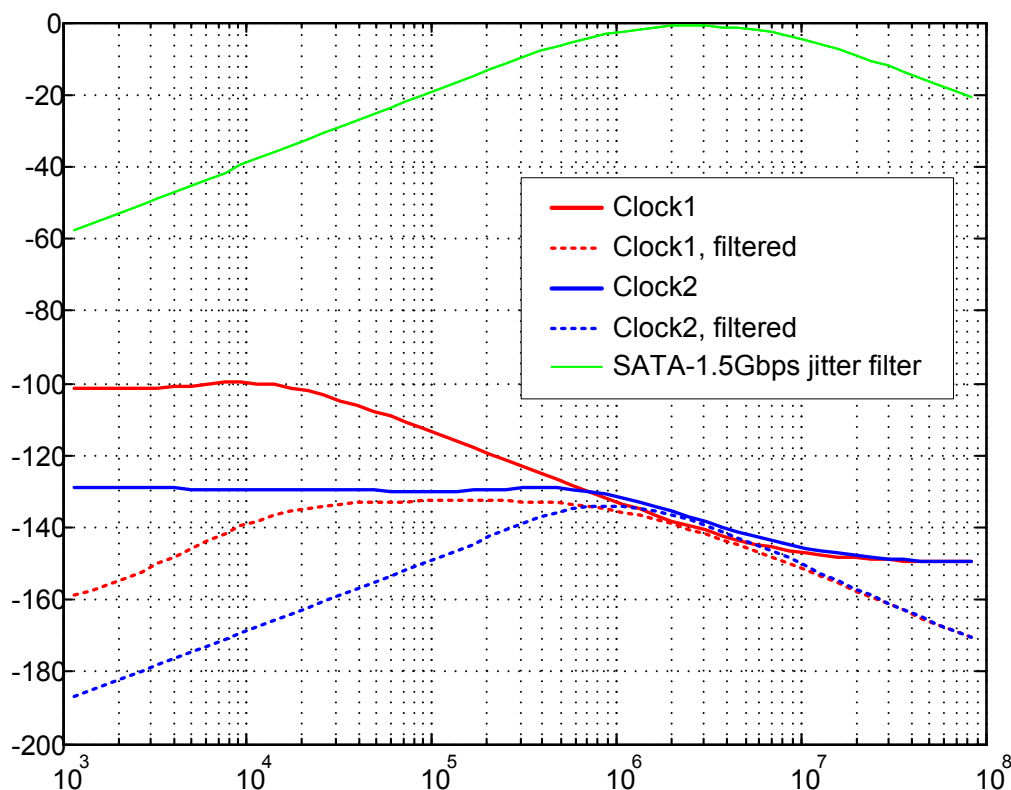


Figure 4: Phase noise profile examples

Table 3 lists the unfiltered phase jitter between 12 kHz to 20MHz as well as filtered jitter for SATA-1.5Gbps in the same frequency range for the four the clock1 and clock2 above. This table shows that clock1 shows higher jitter than clock2 when measuring jitter from unfiltered phase noise spectrum, but clock1 is in fact slightly better than clock2 for SATA-1.5Gbps application. This example highlights the importance of considering the application when evaluating a clock phase jitter for specific applications.

Table 3: Random jitter (RJ) of two example 75MHz clocks for SATA-1.5Gbps application

Clock example	Application RMS random jitter (ps)	12kHz to 20MHz jitter (ps)
Clock1	0.92	5.71
Clock2	0.96	1.66

4 Jitter Budget and Margin Estimation Methodology

The total jitter budget in serial IO applications depends on various components in the IO, with each component consuming certain percentage of the total budget. This section analyzes the remaining jitter budget for all components of the system except clock when using a clock with a given jitter specifications. We analyze transmitter budget because it typically imposes tighter limits on the clock jitter requirements.

The total jitter in a serial IO transmitter is caused by the following sources:

1. Transmitter chipset driver:
The TX driver contributes both random Jitter (RJ) and deterministic Jitter (DJ). The DJ includes both data-dependent jitter (DDJ) and periodic jitter (PJ).
2. Reference clock (RefClk) source:
The RefClk jitter includes RJ and DJ. The major source of RefClk DJ is periodic jitter due to the RefClk phase noise spurious or switching power supply noise.
3. Physical interface between the TX driver and the connector:
This physical interface includes the driver termination components, the board traces and the connector itself. The jitter from this interface is typically dominated by DDJ because virtually all the components in the path are passive; therefore they contribute a negligible amount of RJ.

Jitter from the different sources can be represented by an overall RJ and DJ:

$$RJ_{RMS_TX} = \sqrt{RJ_{RMS_driver}^2 + RJ_{RMS_RefClk}^2} \quad \text{Equation 7}$$

$$DJ_{PP_TX} = DJ_{PP_driver} + DJ_{PP_RefClk} + DJ_{PP_IF} \quad \text{Equation 8}$$

Where RJ_{RMS_TX} , RJ_{RMS_driver} , and RJ_{RMS_RefClk} are the RMS random jitters of the overall transmitter, the driver, and the reference clock, respectively; and DJ_{PP_TX} , DJ_{PP_driver} , DJ_{PP_RefClk} , DJ_{PP_IF} are the peak-to-peak deterministic jitters of the overall transmitter, the driver, the reference clock, and the physical interface, respectively.

The peak-to-peak DJ components are summed directly, while RMS RJ is calculated as the square root of the sum of squares (RSS) of the individual components due to the statistically independent random nature of the RJ components.

Important note: The peak-to-peak values of the RJ components should not be summed directly to compute the overall RJ peak-to-peak value. Instead, the RMS value should be computed first

using Equation 7 and then the peak-to-peak RJ is calculated by multiplying the RMS value by 14.1 for a bit error rate of 10e-12 (as specified by most serial IO standards).

Jitter specifications are defined differently for various standards, but they usually fall into the following categories:

1. Reference clock is directly specified. PCI-Express Gen-II is an example, but this is not the case for most other standards.
2. System level peak-to-peak total jitter (TJ), and DJ or RJ are specified separately. Knowing TJ and RJ, DJ limits can be computed, while RJ limits can be deduced if TJ and DJ are specified.
3. System-level TJ is specified; no separate limits for RJ and DJ given. To deduce the clock referred jitter, we will use the following assumptions:
 - a. Of the total system peak-to-peak jitter budget, 50% is allocated to random jitter (RJ), and 50% to the deterministic jitter (DJ). The DJ includes inter-symbol interference (ISI) due to the package and board bandwidth limitations, power supply injected noise, and clock source periodic jitter.
 - b. The RMS integrated timing jitter is estimated by dividing the peak-to-peak RJ by 14.1.

Using Equation 7 and Equation 8, the jitter budgets of the transmitter chip and the board can be defined as:

$$RJ_{RMS_TX} = \sqrt{RJ_{RMS_dev}^2 - RJ_{RMS_RefClk}^2} \quad \text{Equation 9}$$

$$DJ_{PP_TX} + DJ_{PP_IF} = DJ_{PP_dev} - DJ_{PP_RefClk} \quad \text{Equation 10}$$

The total jitter budget estimates are computed using the following equation (for the BER of 10e-12):

$$TJ_{PP_TX} = (14.1)RJ_{RMS_TX} + DJ_{PP_TX}$$

5 Jitter Budgets with SiTime oscillators for Reference Clock

In this section, we use the analysis in Section 4 to estimate the transmitter jitter budget excluding the clock when SiTime high performance oscillators are used for the reference clock.

An Agilent E5052B signal source analyzer was used to measure the phase jitter of SiT8102 oscillators running at different frequencies, relevant to some of the applications in Table 2. The phase jitter spectrum was processed using the appropriate filters defined in Table 2 and integrated from 100Hz to 20MHz using Equation 2. The results are listed in Table 4. This table shows that the transmitter budget consumed by the SiTime reference oscillator ranges from 3% to 26% for different applications with majority of cases being in 10% range; these are excellent margins to ensure highly reliable function of the serial IOs. Note that the values in Table 4 are only for the transmitter. When considering the whole system, including the receiver, the TX jitter

budget excluding reference clock improves even further to the range of 1% to 9% for these applications.

SiTime high performance differential oscillators, SiT9102 and SiT9002 exhibit slightly better margins than the ones in Table 4 because of their very low levels of phase noise spectrum at high frequency offsets.

Table 4: Jitter budget with SiTime high performance oscillator SiT8102

Serial IO standard	Jitter budget allocated to the TX excluding the clock	Total TX Jitter budget consumed by the reference clock
FibreChannel	91% (1Gbps) 88% (2Gbps) 80% (4Gbps)	9% 12% 20%
SATA & SAS	96% (1.5Gbps) 94% (3Gbps) 88% (6Gbps)	4% 6% 12%
GPON	79%	21%
10GbE	74% (10GBASE-LR) 91% (10GBASE-LX/CX)	26% 9%
Infiniband	92%	8%
XAUI	91%	9%
IEEE1394b (Firewire)	93% (400Mbps) 89% (800Mbps) 87% (1600Mbps)	7% 11% 13%
USB2.0	97% (Device) 93% (Hub)	3% 7%
PCI-Express, Gen-I	95% (embedded clock) 93% (common RefClk, Gen-I, 2.5Gbps) 88% (common RefClk, Gen-II, 5Gbps)	5% 7% 12%

6 Conclusion

Choosing a clock to meet the jitter requirements for many application, such serial IO ones, involves more than integrating phase noise from 12kHz to 20MHz. The knowledge of the application jitter filters in key for evaluating the jitter and its impact on the system function properly and reliably. The analysis of the phase noise contributions for jitter, we showed that almost most serial applications use a bandpass jitter filter to compute effective jitter. The lower cutoff frequencies of such filters range between 300 kHz to 6 MHz, while upper cutoff frequencies range between 2.5MHz to 20MHz.

Using the analysis in this document, we showed that SiTime high performance single-ended (SiT8102, SiT9001) and differential (SiT9102, SiT9002) oscillators easily and reliably meet the jitter requirement for almost all serial IO application, while leaving excellent budget for the rest of the IO system.

SiTime Corporation
990 Almanor Avenue
Sunnyvale, CA 94085
USA
Phone: 408-328-4400
<http://www.sitime.com>

© SiTime Corporation, 2008-2009. The information contained herein is subject to change at any time without notice. SiTime assumes no responsibility or liability for any loss, damage or defect of a Product which is caused in whole or in part by (i) use of any circuitry other than circuitry embodied in a SiTime product, (ii) misuse or abuse including static discharge, neglect or accident, (iii) unauthorized modification or repairs which have been soldered or altered during assembly and are not capable of being tested by SiTime under its normal test conditions, or (iv) improper installation, storage, handling, warehousing or transportation, or (v) being subjected to unusual physical, thermal, or electrical stress.

Disclaimer: SiTime makes no warranty of any kind, express or implied, with regard to this material, and specifically disclaims any and all express or implied warranties, either in fact or by operation of law, statutory or otherwise, including the implied warranties of merchantability and fitness for use or a particular purpose, and any implied warranty arising from course of dealing or usage of trade, as well as any common-law duties relating to accuracy or lack of negligence, with respect to this material, any SiTime product and any product documentation. Products sold by SiTime are not suitable or intended to be used in a life support application or component, to operate nuclear facilities, or in other mission critical applications where human life may be involved or at stake.