

---

# Phase Noise and Jitter Requirements for Serial IO Applications

## Table of Contents

1	Introduction .....	2
2	Phase Noise and Phase Jitter .....	2
3	High-Speed Asynchronous Serial Data Communication (Serial IO) .....	3
3.1	Embedded Clocking.....	4
3.2	Common Reference Clocking.....	7
3.3	Estimating Clock Jitter for Different Serial IOs .....	8
4	Jitter Budget and Margin Estimation Methodology.....	9
5	Jitter Margin with SiTime Oscillators as Reference Clock.....	10
6	Conclusion .....	11
7	Reference .....	12

## 1 Introduction

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

The remainder of this document is organized as follows. Section 2 provides a short overview of phase noise and phase jitter. Section 3 describes phase jitter filters applicable to asynchronous serial standards, while Section 4 provides a jitter budget estimation analysis for such applications. Jitter budgets for various serial IOs when using SiTime high-performance oscillators as reference clocks are provided in Section 5.

## 2 Phase Noise and Phase 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  $\varphi_n$  is the noise of the phase component. The noise  $\varphi_n$  causes noise in the signal  $v(t)$ . The spectrum of the normalized amplitude ( $v(t)$ ) noise around the carrier frequency is referred to as "phase noise". On the other hand, the spectrum of the  $\varphi_n$  is referred to as the "phase noise spectrum" or the "phase jitter spectrum". Both "phase noise" and "phase noise spectrum" are expressed in dBc/Hz units versus offset frequency relative to the carrier frequency  $f_c$ . Strictly speaking, "phase noise" and "phase noise spectrum" are not the same [1], because one represents the noise in the signal amplitude and the other the noise in the signal phase. For most practical applications, however, the two are very close and can be used interchangeably, especially at offset frequencies greater than 100 Hz 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:

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

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

Time interval error (TIE) also represents error in phase, and is sometimes used to compute phase noise. In this document, we will only use phase noise. (Refer to Application Note AN10007 for the relationship between TIE and phase noise.)

### 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 the 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 methods is given in Table 1.

**Table 1: Major serial IO applications**

Serial IO standard	General applications	Clocking method
Fibre Channel	Storage	Embedded
Serial ATA (SATA)	Storage	Embedded
Serial SCSI (SAS)	Storage	Embedded
PCI Express	Computing	Common RefClk, embedded
Serial Gigabit Ethernet (GbE), GPON	Networking	Embedded
Serial 10 Gigabit Ethernet (10 GbE)	Networking	Embedded
Infiniband	Servers	Embedded
XAUI	Networking	Embedded
RapidIO	Computing	Embedded
Fully-buffered DIMM (FBDIMM)	Computing	Common RefClk, embedded
IEEE 1394b (FireWire)	General	Embedded
USB 2.0	General	Embedded
USB 3.0	General	Embedded

### 3.1 Embedded Clocking

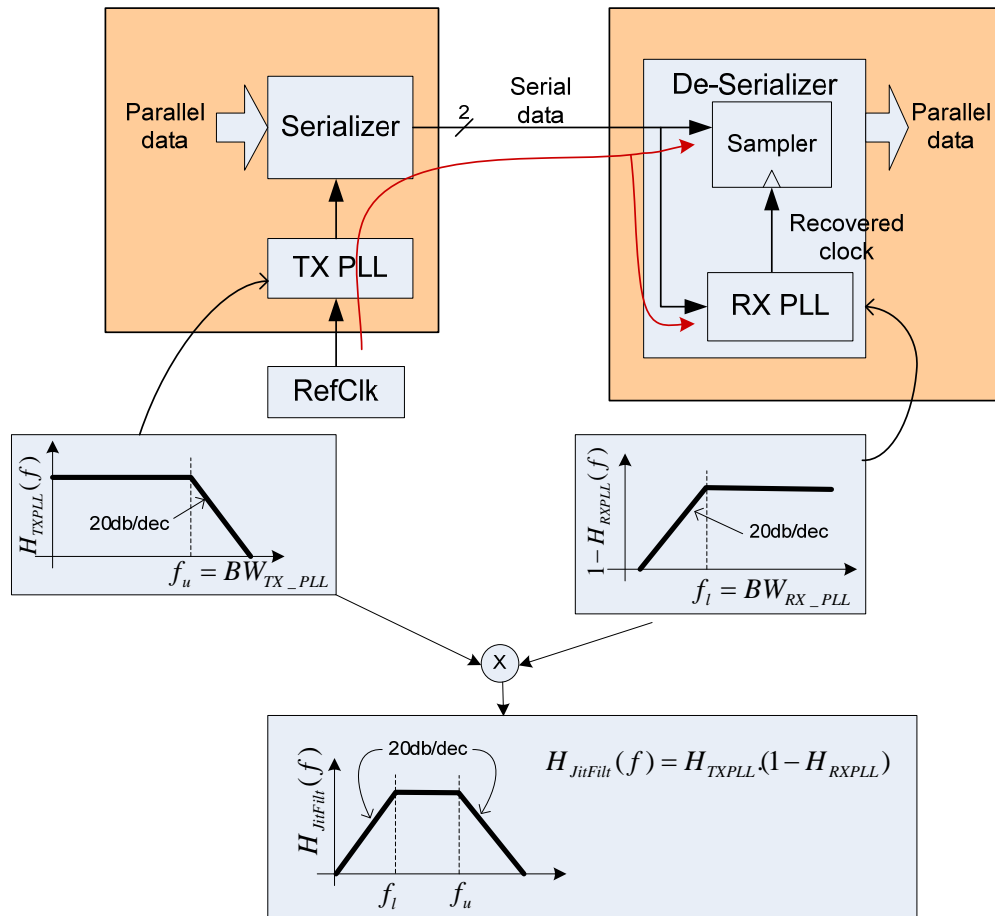


Figure 1: Embedded clocking methodology

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, optimally in the middle of a bit. Most CDR circuits use phase-locked loops (PLLs) 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 its often noisy VCO on the output jitter. The maximum TX PLL bandwidth is roughly 1/10 of the reference clock to allow proper operation of the TX PLL, i.e:

$$BW_{TX\_PLL} \approx \frac{f_{RefClk}}{10} \quad \text{Equation 3}$$

For example, a popular reference clock for serial ATA (SATA) applications is 75 MHz, which leads to the maximum TX PLL bandwidth of 7.5 MHz. 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 (20 dB/dec). 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 the TX PLL is a low-pass filtering of the clock reference phase noise, as shown in Figure 1.

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 sufficiently average the effect of irregular data transitions 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. The Fibre Channel standard was one of the earliest standards to analyze the optimal selection of the RX PLL bandwidth. Its developers conclude that the best choice is as shown below:

$$BW_{RX-PLL} = \frac{f_{baud}}{1667} \quad \text{Equation 4}$$

where  $f_{baud}$  is the electrical data rate in bits-per-second (bps). Many other standards that use embedded clocking incorporated the Fibre Channel recommendations and use the same RX PLL bandwidth relationship. The overall impact of the RX PLL on the reference clock jitter follows high-pass behavior. The PLL is assumed to have first-order behavior, leading to the response shown in Figure 1. In practice, most CDR PLLs have second or third order response, yielding much steeper roll-off in the stop-band region.

Some standards, such as High-Speed USB 2.0, require jitter specifications based on jitter over a number of cycles. It can be shown that such specifications can be mapped to a similar response and a first-order high-pass filter; the derivation of such relationships is beyond the scope of this application note.

Table 2 lists the 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
Fibre Channel	106.25 212.5	1.0625 2.125 4.25 8.5	637 kHz 1.275 MHz 2.55 MHz 5.1 MHz	10 MHz 20 MHz
Serial ATA (SATA) & Serial SCSI (SAS)	37.5 75 120 150	1.5 3 6 12	900 MHz 1.8 MHz 3.6 MHz 7.2 MHz	4 MHz 7.5 MHz 12 MHz 15 MHz
Serial Gigabit Ethernet (GbE)	25 125	1.25	637 kHz *	2.5 MHz 12.5 MHz
Serial 10 Gigabit Ethernet (10 GbE)	25 161.1328	10.3125 3.125 (x 4 lanes)	4 MHz 1.875 MHz	2.5 MHz 20 MHz *

	156.25 312.5 644.53125			20 MHz * 30 MHz 60 MHz
Infiniband	100 200	2.5 5 10	1.5 MHz 3 MHz 6 MHz	10 MHz 20 MHz
XAUI	156.25 312.5	3.125	1.875 MHz	20 MHz * 30 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
USB 2.0	12 24 48	0.48	500 kHz	5 MHz *
USB 3.0	50 250	5	3 MHz	5 MHz 25 MHz
* These values follow Equation 3 and Equation 4 only approximately; they have become more established limits in the industry.				

### 3.2 Common Reference Clocking

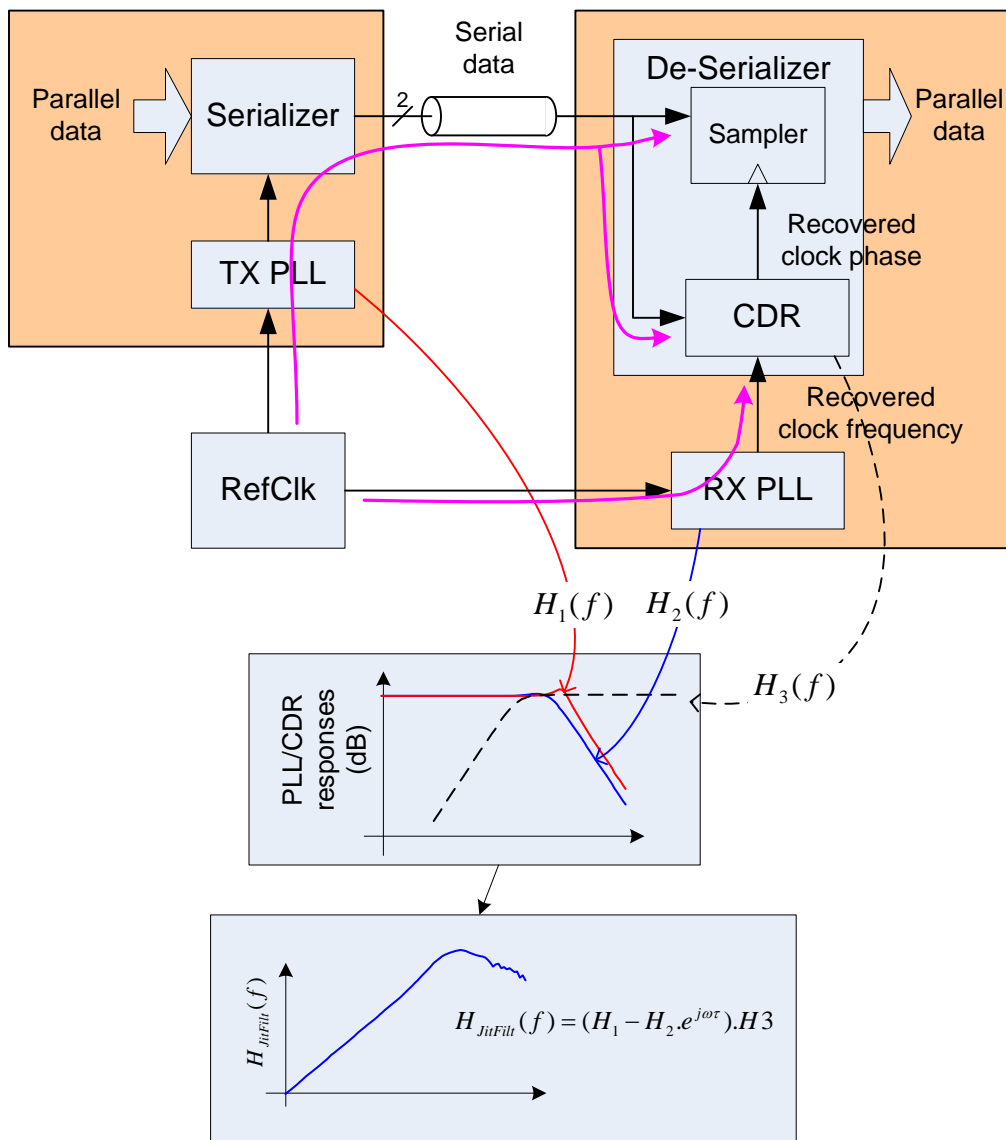


Figure 2: Reference clocking for serial IOs

Another clocking methodology for serial interfaces routes the transmitter reference clock (RefClk) to the receiver. The transmitter includes a PLL to generate a high-frequency signal for clocking out serial data. Such a PLL is modeled as a second-order system with a given response. The 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 the locally synthesized clock edge with the center of the 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 the phase noise spectrum at low frequency offsets that fall in the passbands of both RX and TX PLLs because noise in that

region is common in the transmitter and receiver and is tracked closely by the receiver PLL. The dominant noise is concentrated in the region that falls in the pass-band of the TX PLL, but in the stop-band of the receiver PLL. The phase noise that falls in the stop-bands of both PLLs is also significantly attenuated.

### 3.3 Estimating Clock Jitter for Different Serial IOs

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

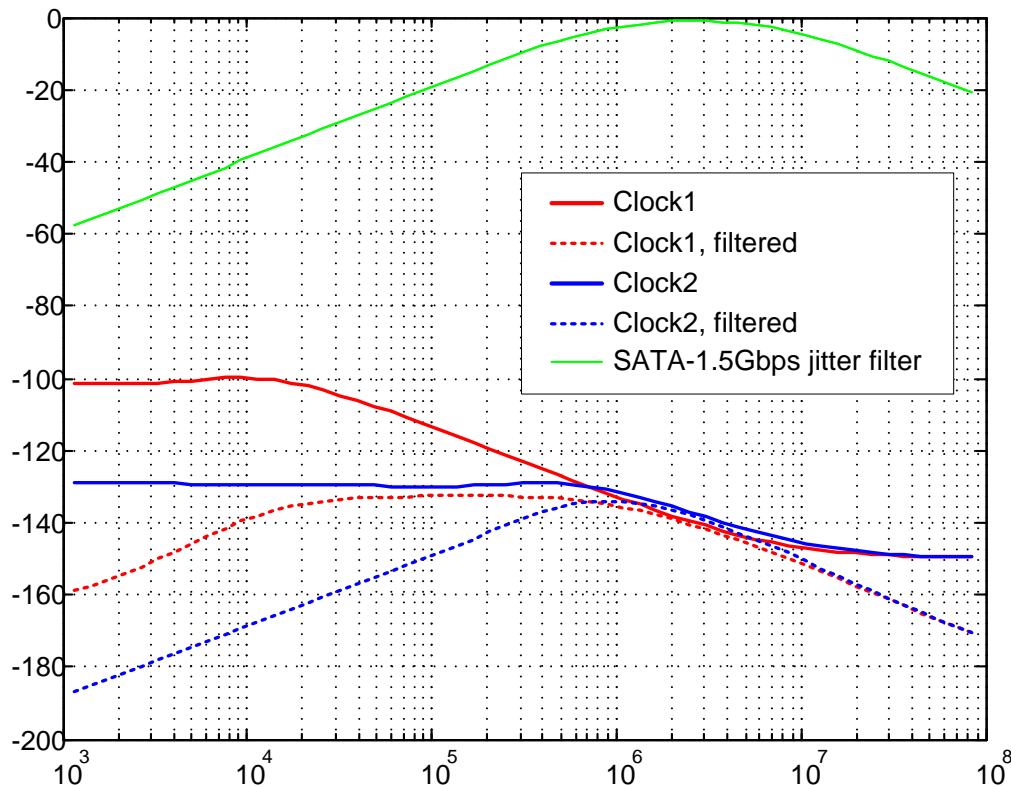


Figure 3: Phase noise profile examples

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

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

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 jitter components in the IO interface, with each component consuming a certain percentage of the total jitter budget. This section analyzes the ratio of the remaining jitter budget for all components of the system, excluding the clock, to the total jitter budget, which includes the clock. The total jitter budget typically is defined by the relevant standard for a particular application. We analyze the transmitter budget because it imposes tighter limits on clock jitter requirements.

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

1. The transmitter chipset driver:  
The TX driver contributes both random jitter (RJ) and deterministic jitter (DJ). DJ includes both data-dependent jitter (DDJ) and periodic jitter (PJ).
2. The reference clock (RefClk) source:  
RefClk jitter includes RJ and DJ. The major source of RefClk DJ is periodic jitter due to the RefClk spurious phase noise or switching power supply noise.
3. The 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 these different sources can be represented by an overall RJ and DJ:

$$RJ_{RMS\_TXout} = \sqrt{RJ_{RMS\_TXphy}^2 + RJ_{RMS\_RefClk}^2} \quad \text{Equation 5}$$

$$DJ_{PP\_TXout} = DJ_{PP\_TXphy} + DJ_{PP\_RefClk} + DJ_{PP\_IF} \quad \text{Equation 6}$$

Where  $RJ_{RMS\_TXout}$ ,  $RJ_{RMS\_TXphy}$ , and  $RJ_{RMS\_RefClk}$  are the RMS random jitters of the overall transmitter, the driver, and the reference clock, respectively; and  $DJ_{PP\_TXout}$ ,  $DJ_{PP\_TXphy}$ ,  $DJ_{PP\_RefClk}$ ,  $DJ_{PP\_IF}$  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.

**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 5 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. The reference clock is directly specified. PCI Express Gen 2 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 deterministic jitter (DJ). 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 5 and Equation 6, the jitter budgets of the transmitter chip and the board can be defined as:

$$RJ_{RMS\_TXphy} = \sqrt{RJ_{RMS\_TXout}^2 - RJ_{RMS\_RefClk}^2} \quad \text{Equation 7}$$

$$DJ_{PP\_TXphy} + DJ_{PP\_IF} = DJ_{PP\_TXout} - DJ_{PP\_RefClk} \quad \text{Equation 8}$$

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

$$TJ_{PP\_TXphy} = (14.1)RJ_{RMS\_TXphy} + DJ_{PP\_TXphy} \quad \text{Equation 9}$$

Subsequently, the jitter margin is computed as below:

$$TX_{Jit\_margin} = TJ_{PP\_TXphy} / TJ_{PP\_TXout} \quad \text{Equation 10}$$

## 5 Jitter Margin with SiTime Oscillators as Reference Clock

In this section, we use the analysis in Section 4 to estimate the transmitter jitter margin when using SiTime high-performance oscillators as 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 shown in Table 2. The phase jitter spectrum was processed using the appropriate filters defined in Table 2 and integrated from 100 Hz to 20 MHz 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 the majority of cases being in the 10% range; these are

excellent margins to ensure highly reliable functioning of the serial IO. 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.

**Table 4: Jitter budget with SiTime oscillators**

Serial IO standard	Baud rate (Gbps)	Transmitter jitter margin		
		SiT8103/ SiT8003/ SiT8004 (LVCMOS output)	SiT9102/SiT9107 (LVPECL, LVDS, HCSL, CML outputs)	SiT8102 (LVCMOS output)
Fibre Channel	1.0625	97.5%	91%	90%
	2.125	96.5%	88%	86%
	4.25	95%	80%	79%
	8.5	93%	77%	75%
SATA & SAS	1.5	98.8%	96%	96%
	3	97.7%	94%	93%
	6	96.5%	88%	87%
GBE, GE-PON	1.25	96%	86%	84%
	2.5	95%	88%	86%
10 GbE –XAUI 1GBASE-R	3.125	95.5%	92%	90%
	10.3125	92.5%	81%	78%
Infiniband	2.5	96%	85%	84%
	5	93.5%	89%	87%
IEEE 1394b (FireWire)	0.4	98.5%	NA	93%
	0.8	97.5%		89%
	1.6	96.5%		87%
USB 2.0	0.48	99% (Device) 98% (Hub)	NA	97% (Device) 93% (Hub)
PCIe-Gen 1 – Embedded RefClk	2.5	98%	95%	94%
PCIe-Gen 1 – Common RefClk	2.5	99%	93%	91%
PCIe-Gen 2 – Common RefClk	5	96%	88%	87%

## 6 Conclusion

Choosing a clock to meet the jitter requirements for many applications, such as those with serial IOs, involves more than integrating phase noise in the 12 kHz to 20 MHz range. Knowledge of application jitter filters is key for evaluating jitter and its impact on the system's ability to function properly and reliably. Our analysis of the phase noise contributions to jitter showed that 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 the upper cutoff frequencies range between 2.5 MHz to 20 MHz.

Using the analysis in this document, we showed that SiTime high-performance single-ended (SiT8102) and differential (SiT9102, SiT9107) oscillators easily and reliably meet the jitter requirements of all serial IO applications, while leaving an excellent budget for the rest of the IO system.

## 7 Reference

- [1] R. Navid, et. Al, "An Analytical formulation of phase noise of signals with Gaussian-distributed jitter", IEE Transaction on Circuits and Systems-II: Express briefs, Vol. 52, No. 3, March 2005.

---

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

© SiTime Corporation, 2008-2011. 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.