Introduction

Clock Jitter is commonly used to describe the performance requirements of oscillators and PLLs when driving high performance components like FPGAs, Microprocessors, and PHYs. But the plethora of clock jitter types, measurement methodologies, and corresponding specifications is broad and often leads to confusion. It is important to use the jitter type that is most appropriate for the application of interest. The purpose of this Application Note is to provide some basic guidelines for selecting and quantifying the relevant jitter specification for a given application, as summarized in Table 1.

***Note**: This Application Note is limited in scope in that it does not cover all jitter types, but the more common types are presented in detail.*

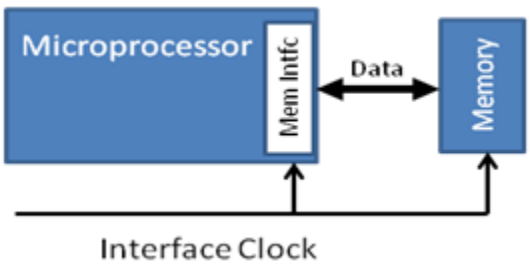
Table 1: Summary of Relevant Jitter by Application

Application (examples)	Relevant Jitter	Specification (units)	Qualifier
✓ **Synchronous Interface** ✓ **Synchronous Logic**	Period Jitter	Pk-Pk Period Jitter (time)	Bit Error Rate
		RMS Period Jitter (time)	10K cycles
✓ **Micro Complex** ✓ **Motherboard Design** ✓ **Spread Spectrum**	High Freq Jitter	C2C Jitter (time)	1K consecutive cycles
✓ **High Speed SerDes** ✓ **ADC (RF Signal Chain)**	Frequency Domain Jitter	Phase Noise, PN Plot (dBc/Hz) (Frequency Domain Measurement)	Carrier Offset Freq
		RMS Phase Jitter (time) (extrapolated from a PN Plot)	Integration Band (i.e. filter)
		Pk-Pk Total Jitter UI (time) (SerDes Eye Closure Specification)	Random Jitter Budget, Integration Band (i.e. filter)
		Spurs (dBc) (Deterministic Jitter)	Carrier Offset Freq
		"Additive" RMS Phase Jitter (time) (Fan Out Buffer "relevance")	Integration Band (i.e. filter)

Period Jitter

Period Jitter is a measurement of the "Actual" Clock period deviation, relative to its "Average" (mean) clock period. The application relevance of Period Jitter is intuitive and well understood. Consider the **Synchronous Interface** use case of the interface clock for a Microprocessor and its companion synchronous Memory device, as shown in Figure 1 and Figure 2.

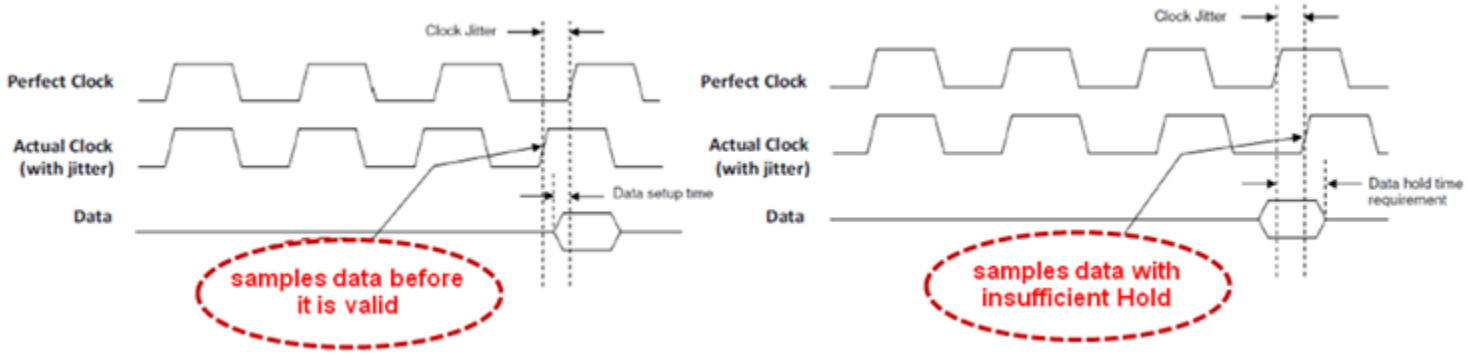
Figure 1. Period Jitter (Synchronous Interface Use Case)



 ©2014-2022 Renesas Electronics Corporation

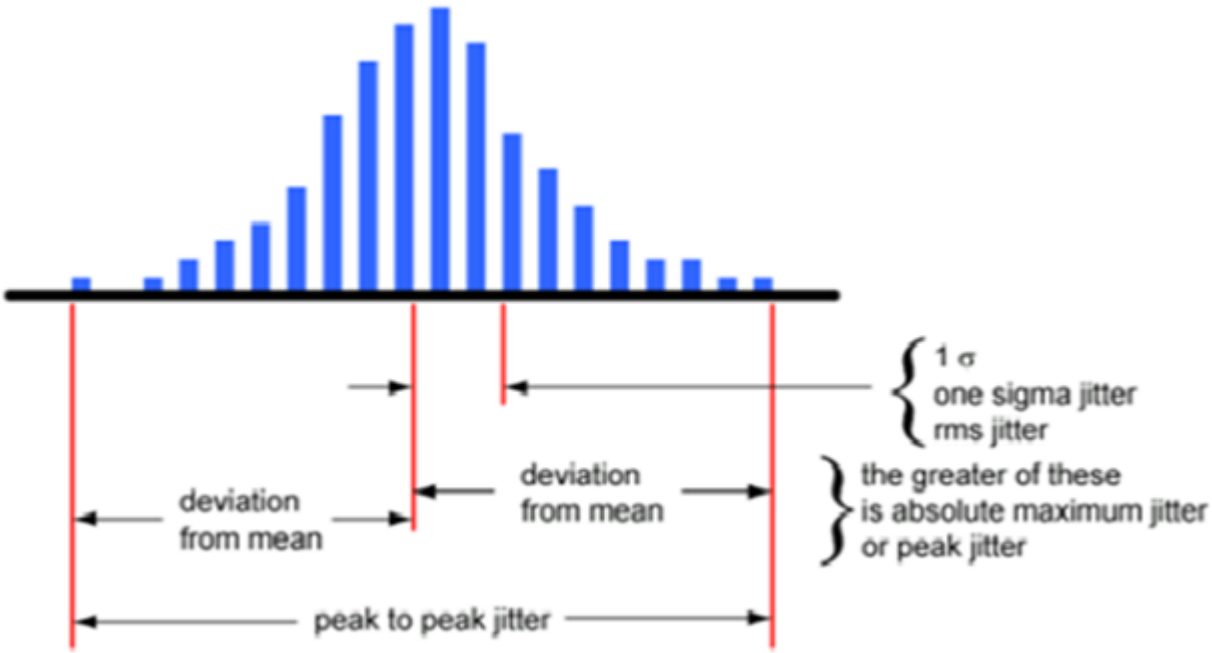
In this example the same clock is used to drive both the memory interface of the Microprocessor and the Memory itself. The corresponding Data setup and hold specifications for the input data latch/register of the receiving device must be met. But, interface clock jitter eats into the timing budget margin for meeting this requirement. And at higher frequencies, this margin is further reduced since there is less period (i.e. time) for use. A similar use case is the Period Jitter for the clock used for the **Synchronous Logic** design implemented in an FPGA.

Figure 2. Period Jitter (Synchronous Logic Use Case)



It is important to keep in mind that the skirts of the Gaussian distribution of “random” Period Jitter extend forever, as shown in Figure 3. In other words, the measured **Peak-to-Peak (Pk-Pk) Period Jitter** gets worse as the number of measurement cycles increases. This is because random jitter is unbounded.

Figure 3. Gaussian Distribution of “random” Period Jitter



For this reason, the absolute maximum Pk-Pk Period Jitter is not a practical measurement. But, a probability can be assigned to jitter exceeding a point on the Gaussian distribution. Bit Error Rate (BER) is application specific, and is commonly used for this purpose. With the known acceptable BER for a given application, the required **Root Mean Square (RMS) Period Jitter** can then be calculated using the corresponding Gaussian Probability Density Function, as shown in Figure 4.

Figure 4. RMS Multiplier for a Given Application's Acceptable Error Rate

RMS Background

RMS = Root Mean Square, also known as the standard deviation, or one Sigma (σ). RMS is a statistical measure of the magnitude of a varying quantity. It can be calculated for a series of discrete values or for a continuously varying function (i.e. RMS Phase Jitter from a continuous Phase Noise Plot). The term RMS is an abbreviation for "the square root of the mean of the squares", and is calculated in this manner

1) Take the square of each sample
2) Calculate the mean all the squares (sum & divide)
3) Take the square root of the mean

This statistical measurement is especially useful when samples are positive and negative (i.e. sinusoidal waveforms). The square of a negative sample is positive. So, RMS is a true measure of amplitude. Example use case – an alternating current through a resistor results in both positive and negative current numbers. The simple mean calculation would yield a total current of zero. But, of course this is inaccurate as we know the resistor dissipates power (I_{RMS}^2*R) regardless of the direction of the current flow. The corresponding RMS calculation yields a non-zero number, and is thus more representative of what is actually happening.

$(10)^{-12}$ commonly used for many high speed interfaces

BER	Sample Size	RMS Multiplier
$(10)^{-3}$	1,000	±3.291
$(10)^{-4}$	10,000	±3.891
$(10)^{-5}$	100,000	±4.417
$(10)^{-6}$	1,000,000	±4.892
$(10)^{-7}$	10,000,000	±5.327
$(10)^{-8}$	100,000,000	±5.731
$(10)^{-9}$	1,000,000,000	±6.109
$(10)^{-10}$	10,000,000,000	±6.467
$(10)^{-11}$	100,000,000,000	±6.807
$(10)^{-12}$	1,000,000,000,000	±7.103
$(10)^{-13}$	10,000,000,000,000	±7.441
$(10)^{-14}$	100,000,000,000,000	±7.739
$(10)^{-15}$	1,000,000,000,000,000	±8.014

For a clock period sample size of 1000, the Pk-Pk Period Jitter for 999 of those clocks will fall within 6.582 σ (±3.291 σ) from the mean clock period. Only 1 in 1000 samples, on average, will have a clock period deviation more than this value.

RMS Period Jitter is specified by Renesas for a clock period sample size of 10K. This is compliant to JEDEC Standard JESD65B (http://www.jedec.org/sites/default/files/docs/jesd65b.pdf). As an example, consider the Renesas **ICS83421 PLL Synthesizer**, datasheet excerpt shown below. The datasheet specifies a typical RMS Period Jitter of 3.0ps, measured across 10K cycles. Assuming an acceptable error rate of 10^{-12} for the given application, one can then predict a Pk-Pk Period Jitter ~ (14) x 3.0ps = 42ps, or ± 21pS.

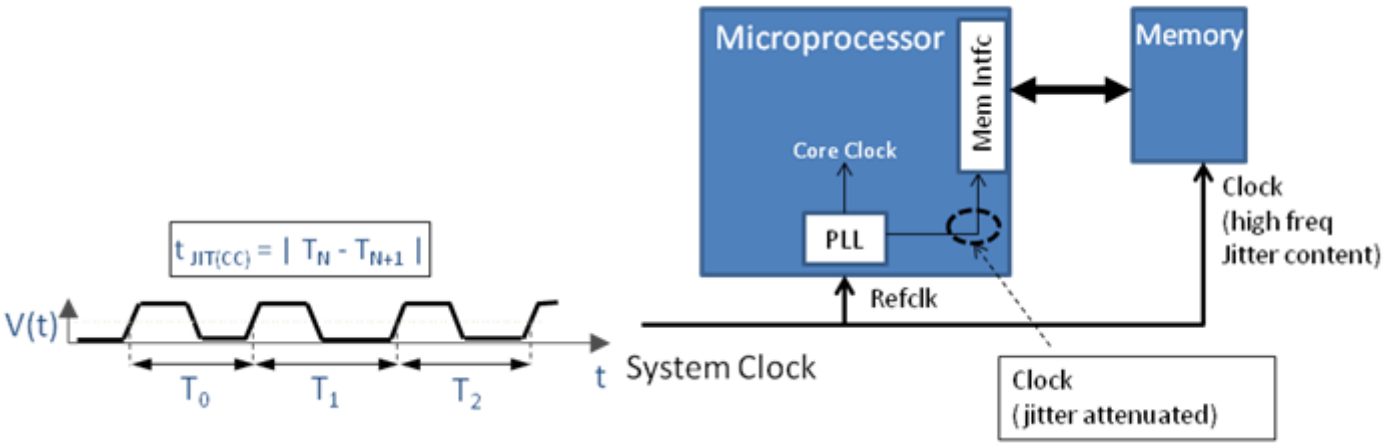
TABLE 7. AC CHARACTERISTICS, $V_{CC} = V_{CCA} = V_{CCO}$ = 3.3V±5%, TA = 0°C TO 70°C

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum	Units
F_{OUT}	Output Frequency		103.3		260	MHz
*t*jit(per)	Period Jitter, RMS; NOTE 1			3	4.5	ps
*t*sk(o)	Output Skew; NOTE 2, 3				15	ps

High Frequency Jitter

Cycle-to-Cycle (C2C) Jitter is the maximum variation in period between two consecutive cycles. There is no reference to an average clock for comparison, as is the case for Period Jitter. C2C Jitter is classified as a High Frequency Jitter, and its application relevance has its roots in the **Microprocessor Complex** specific to early motherboard design, as shown in Figure 5.

Figure 5. C2C Jitter (Motherboard Clock Use Case)



Typically, a Microprocessor has an internal Phase Lock Loop (PLL) that accepts a low frequency external System Clock as its Refclk input. This internal PLL then generates the higher internal frequencies needed for its pipeline core and memory interface. But, like all PLLs, the Microprocessor internal PLL acts as a low pass filter of the noise presented on the Refclk. As a result, the high frequency jitter content is not passed along to its internal memory interface. However, the same external System Clock is used to drive the companion memory device, and so the high frequency jitter is presented directly to the memory. In earlier generation motherboard design, memory interface errors were traced back to this synchronization difference between these two domains. As a result, the C2C Jitter specification was introduced as a means for quantifying High Frequency jitter. Newer generation Microprocessors typically synthesize their own external memory interface clock, and so this use case for C2C Jitter is no longer as relevant. Still, many modern day Microprocessors datasheets still include C2C Jitter specifications, in addition to traditional Period Jitter specifications.

It is worth noting that C2C Jitter is also commonly used to illustrate the stability of a **Spread Spectrum** generated clock, where low frequency modulation is intentionally induced onto the output clock of a PLL. Frequency spreading, in this manner, serves to reduce Electromagnetic Interference (EMI) which is an important criterion for many consumer electronics and radio applications. Period Jitter is sensitive to low frequency modulation, and the corresponding jitter measurement results of a Spread Spectrum generated clock can be significantly skewed. But, C2C jitter measurements of a Spread Spectrum generated clock are not as impacted by this modulation.

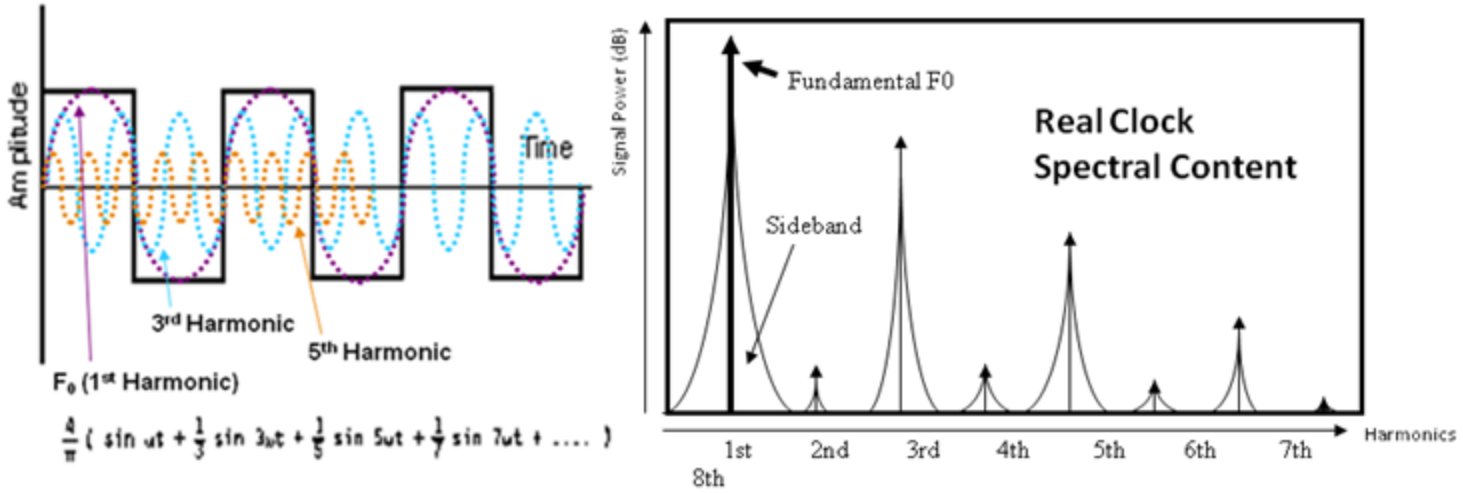
C2C Jitter is specified by Renesas for a sample size of 1K consecutive cycles. This is compliant to JEDEC Standard JESD65B (http://www.jedec.org/sites/default/files/docs/jesd65b.pdf). C2C Jitter is typically reported as a Peak value, or sometimes as a Pk-Pk value. As an example, consider the **Renesas 5P49V5901 VersaClock® 5 Programmable Clock**, datasheet excerpt shown below. The datasheet specifies a typical C2C Jitter of 30ps (Pk-Pk), or 15pS Peak.

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
t6	Clock Jitter	Cycle-to-Cycle jitter (Peak-to-Peak), multiple output frequencies switching, differential outputs (1.8V to 3.3V nominal output voltage)		30		ps
		Cycle-to-Cycle jitter (Peak-to-Peak), multiple output frequencies switching, LVCMOS outputs (1.8 to 3.3V nominal output voltage)		30		ps

Frequency Domain Jitter

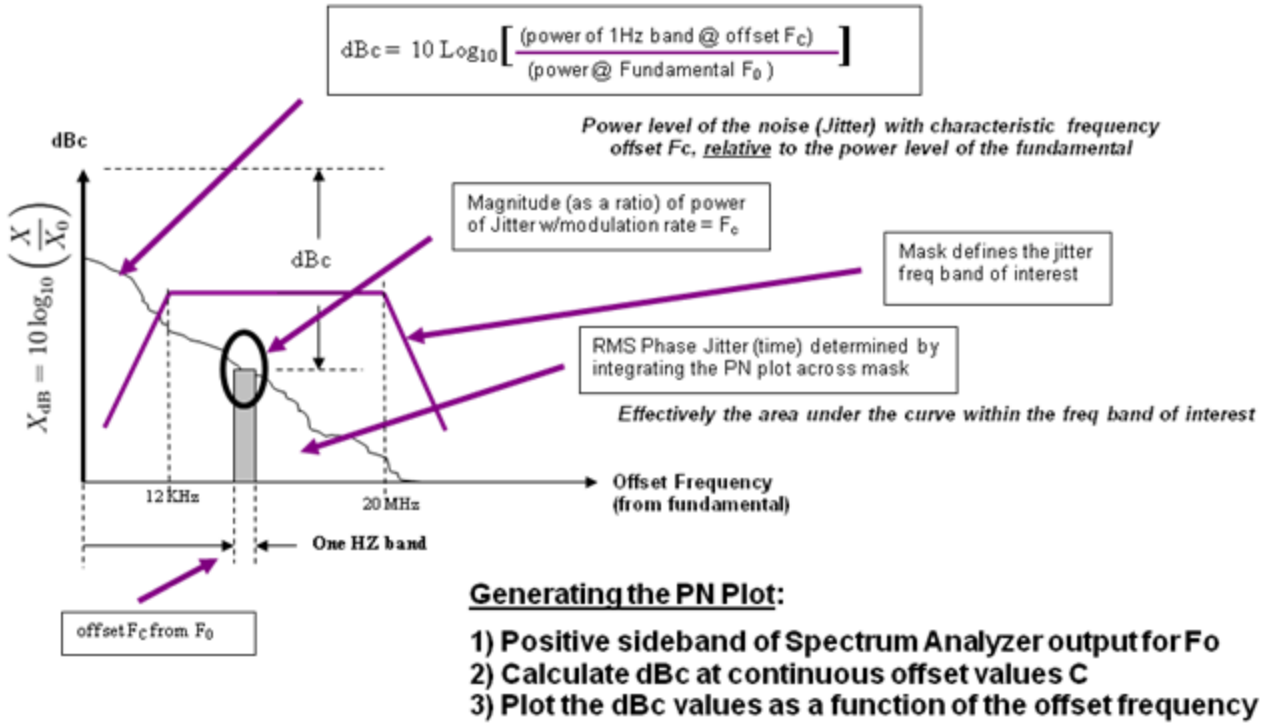
According to Fourier, any periodic wave (i.e. clock signal) can be constructed as an infinite sum of a series of Sine Waves, as shown in Figure 6. Thus, a clock signal can be described as a set of frequencies and corresponding amplitudes. The nominal frequency of a clock is called the Fundamental (also called F_0, or the 1st harmonic). Integer multiples of F_0 are called the Harmonics. The "Perfect Clock" is built with only odd Harmonics, in ever decreasing amplitude.

Figure 6. Fourier Representation of a Clock Signal (plus, corresponding spectral representation)



No real world clock signal is perfect, however. Sideband spectral content on the Fundamental implies frequency deviation, or timing jitter, which is what clock jitter specifications attempt to quantify. Note that there is both positive and negative sideband content, which is characteristic of random jitter. The **Phase Noise (PN)** of a clock signal, represented as a **PN Plot** as shown in Figure 7, is a quantification of this Fundamental sideband content. Typically generated by Spectrum Analyzer test equipment, a PN Plot is particularly useful for seeing the frequency characteristics contributions of clock jitter that would otherwise take very long sample times and long record lengths to measure in the time domain.

Figure 7. PN Plot – Frequency Domain Representation of Clock Jitter



deciBel (dB) Background

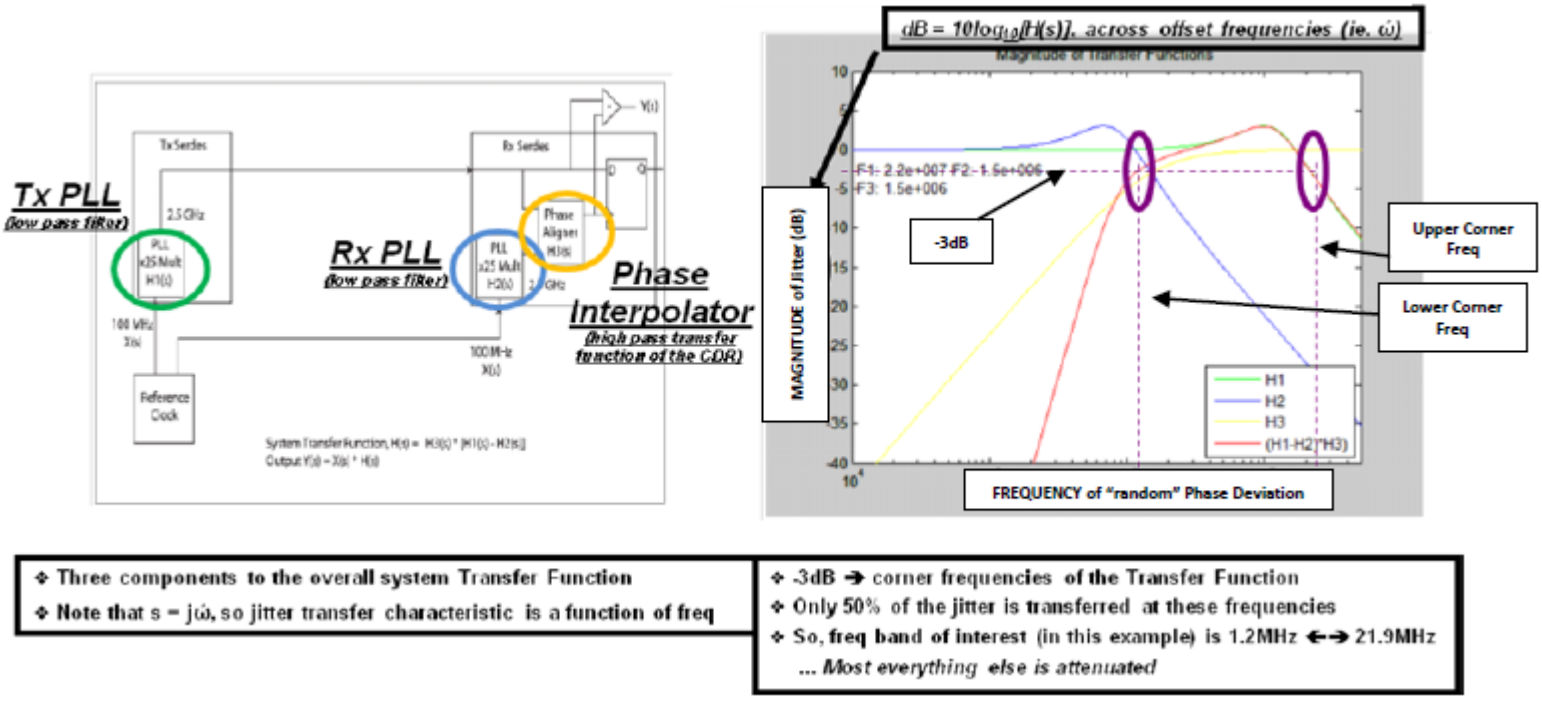
dB is used to quantify the magnitude, usually power, for a sample (i.e. power @ F_C), relative to a specified or implied reference level (i.e. power of F_0). dBm implies that the reference quantity is 1mW. dBc implies that the reference signal is the fundamental frequency F_0 (of the carrier). It is effectively a ratio, and thus a dimensionless unit. Using a logarithmic scale allows for very large and very small ratios to be represented conveniently. The decibel calculation is ten times the base-10 logarithm of the ratio of the measured quantity to the reference:

- $dBc = 10\log_{10}[P_{jitter}/P_{fundametal}]$, at carrier offset frequency C
  (for dBm, simply replace $P_{fundametal}$ with 1mW)

The term Phase Noise is not to be confused with Phase Jitter (or, Accumulated Jitter). Phase Noise is the term most commonly used to describe the "randomness" quality of instantaneous phase fluctuations. The corresponding frequency fluctuation, and thus period fluctuation, is a function of these instantaneous phase fluctuations. Thus, a PN Plot is a frequency domain representation of clock jitter, and is effectively the magnitude of jitter at the specific frequency deviation points from the mean. And, the magnitude (or power) of the jitter at a particular frequency deviation is a function of how often that deviation occurs. So, in summary, a PN Plot is effectively an indication of how often a particular period deviation occurs. And, the lower the dBc value for a given carrier offset frequency, the better.

The term **RMS Phase Jitter** is used to denote that the RMS Jitter value that is extrapolated from a frequency domain Phase Noise Plot. The terminology is intended to differentiate this jitter type from "RMS Jitter" previously detailed in this paper, which is strictly a time domain measurement of Period Jitter. As detailed by Parseval's Theorem, the conversion to an RMS Phase Jitter value from a PN Plot is largely an integral function (i.e. area under the PN Plot). For this reason, the integration range for RMS Phase Jitter is a necessary qualifier for this specification. This integration range is also referred to as the "mask", and is effectively the jitter frequency filter specific to the application. The purpose of this jitter mask is to restrict the quantification of jitter to the range of jitter frequencies that are not filtered by the application transfer function. The quantification of jitter in the frequency domain, represented by an RMS Phase Jitter specification, makes this clock jitter type particularly relevant for **SerDes** use cases (i.e. SDH, SONET, Ethernet, XAUI, PCIe, sRIO), as shown in Figure 8.

Figure 8. RMS Phase Jitter (SerDes Use Case)

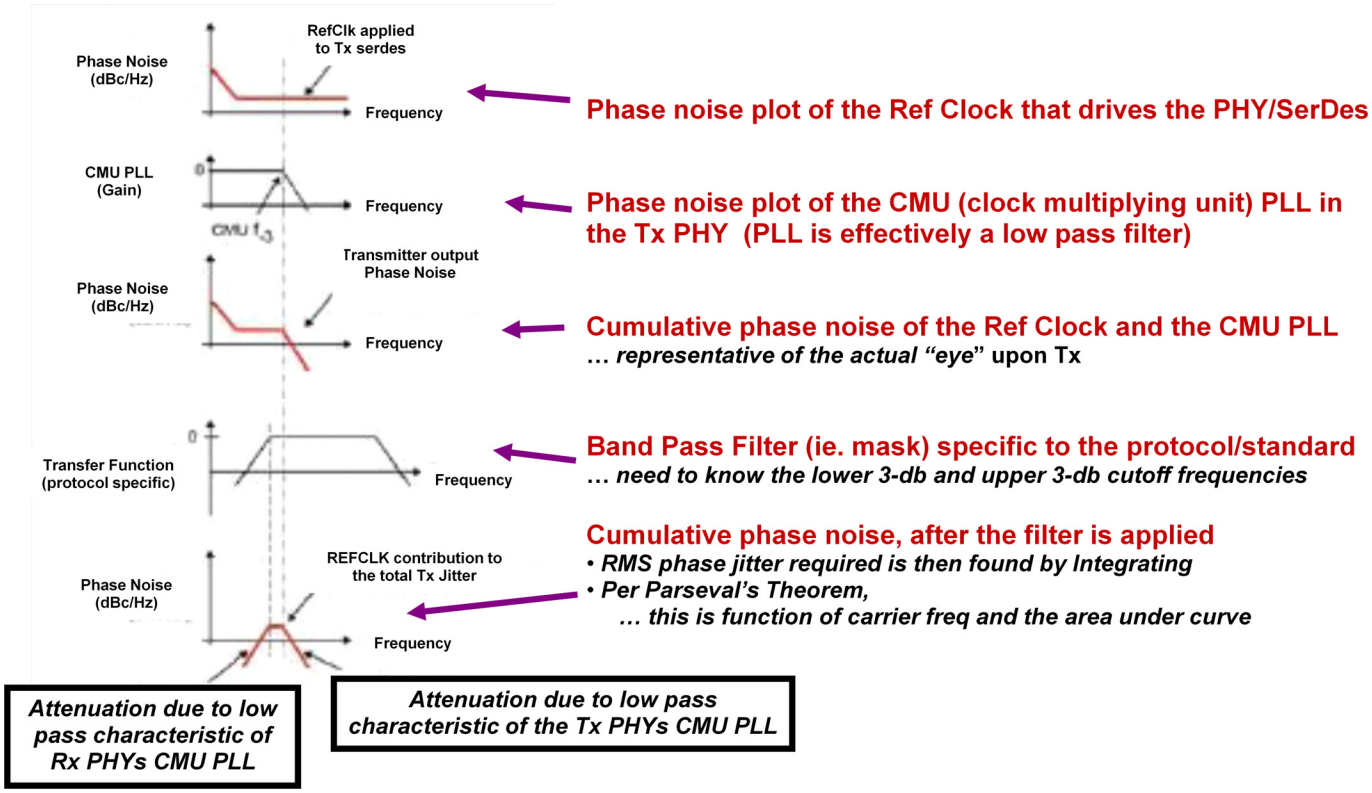


The Phase Interpolator is part of the Rx SerDes CDR circuitry and its transfer function is effectively a high pass filter (H3). But the Tx SerDes PLL (H1) and the Rx SerDes PLL (H2) transfer functions are effectively low pass filters that attenuate high frequency noise present on their respective inputs. And, the total transfer function of the system is given by: Ht(s) = [H1(s)-H2(s)]*H3(s).

A very important benchmark for a given PLL is its bandwidth, also referred to as its "-3dB corner frequency". This is effectively the nominal frequency offset at which only ½ the power of the input clock jitter is transferred to the output. Note that the composite SerDes transfer function is comprised of both an upper -3dB corner frequency and a lower -3dB corner frequency. And, clock jitter outside these corner frequencies is effectively attenuated.

The origin of the upper and lower corner frequencies for a SerDes transfer function is further detailed in Figure 9. The Tx SerDes Clock Multiplying PLL acts as a low pass filter. As a result, high frequency jitter present on the Tx Refclk is not transferred to the output of that PLL. This effectively defines the upper -3dB corner frequency of the integration band of interest.

Figure 9. Corner Frequencies (i.e. jitter mask) of a SerDes Transfer Function



Similarly, the Rx SerDes CDR complex on the receiving end of a serial data link used to recover the in-band clock also uses an internal PLL, and thus will also only pass the low frequency jitter on the Refclk. This is what effectively defines the lower -3dB corner frequency for the integration band of interest. Typically, both the Tx PLL and the Rx PLL track low frequency jitter and attenuate high frequency jitter. So, it is the mid range frequencies of most interest since the PLLs do not necessarily track these equally. Together, these effects bound the SerDes Transfer Function for the given application. If the RMS Phase Jitter is too high for the integration band of interest, then FIFO over/under-runs will increase beyond an acceptable error rate for the given application.

Thus, the RMS Phase Jitter requirements for a clock used to drive a SerDes must be qualified with the application specific jitter mask, or filter. Some example application specific filters include:

- Fibre Channel (637 KHz <<>> 10 MHz)
- 10 Gigabit Ethernet XAUI (1.875 MHz <<>> 20 MHz)
- SONET OC-48 (12KHz <<>> 20MHz)

- SONET OC-192 (20KHz <<>> 80MHz)
- SATA/SAS (900 KHz <<>> 7.5 MHz)

The various network communications standards (i.e. GE, 10GE, etc.) often specify **Pk-Pk Total Jitter Unit Interval (UI)** as a percentage of 1 UI. This is actually a SerDes eye closure specification that must be met in order to meet the acceptable BER, which is typically 10^{-12} for many of the standards. This specification is still bounded by an integration range of interest. But, a Pk-Pk Total Jitter UI specification requires conversion to a corresponding random jitter specification in order that one may quantify the required “random” RMS Phase Jitter value. Here are some common system level jitter budget assumptions that may be applied in order to arrive at a jitter budget for the clock driving the SerDes.

- ✓ A typical Pk-Pk Total Jitter (TJ) limit is 0.65 UI (bit errors occur when the eye is only 35% open)
- ✓ Random Jitter (RJ) is budgeted ¼ the Total Jitter
- ✓ Most of the RJ comes from the clock and SerDes
  - – But, conservatively assume the clock is budgeted ¼ the RJ budget
- ✓ 10^{-12} BER is most common, so remember 14 as the number of standard deviations
- ✓ Divide TJ_{pk-pk} (0.65 * 1 UI) by 224 (14*4*4) to get the acceptable random RMS Phase Jitter.
- ✓ If not specified, then;
  - – For long range communication, use a 12KHz - 20MHz filter
  - – For short range communication, divide baud rate by 1667 to get lower -3dB corner frequency

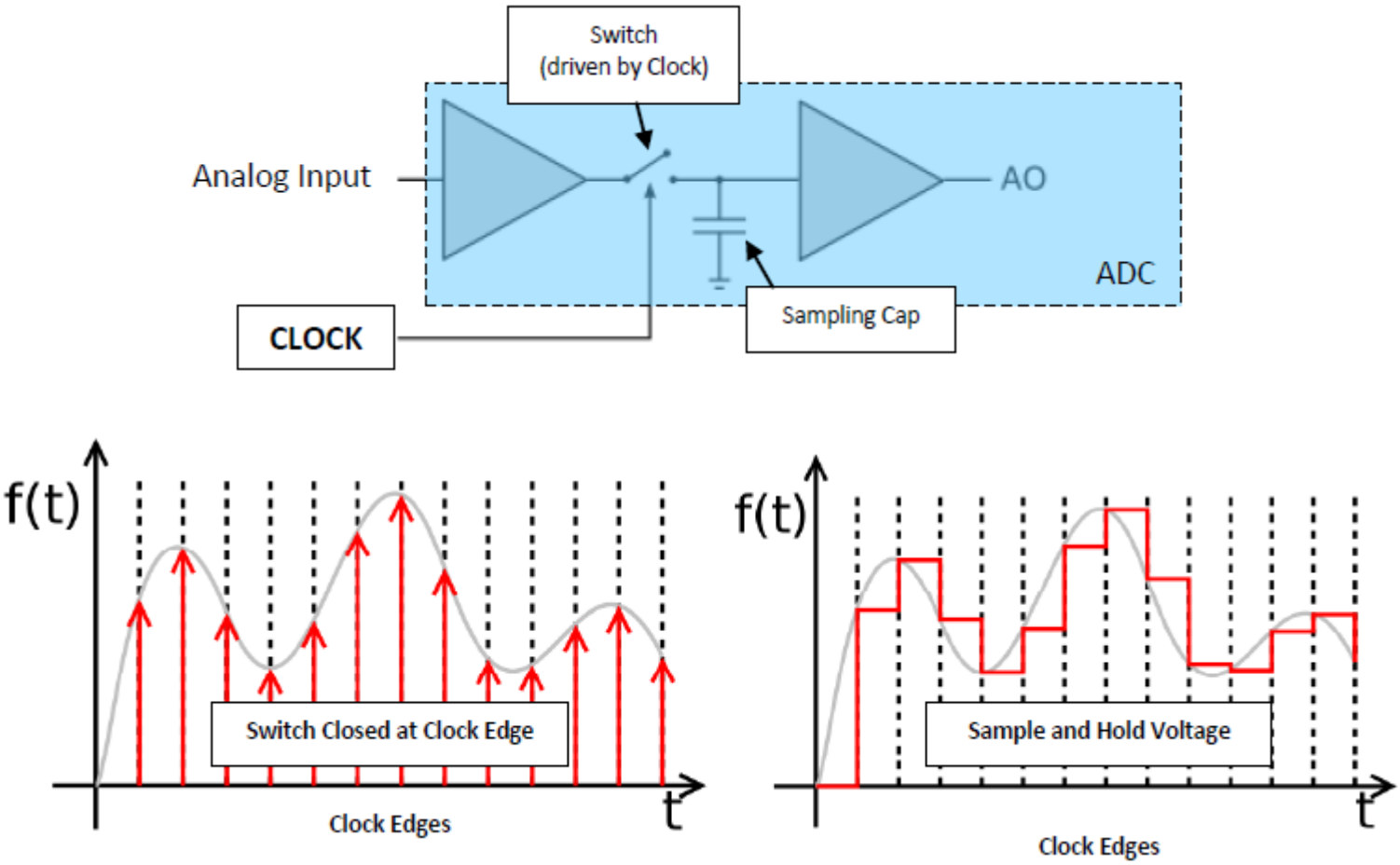
For example, consider 10 Gigabit Ethernet implemented as four XAUI SerDes lanes each running 3.125Gbps. The corresponding RMS Phase Jitter requirement is calculated as follows.

- One UI for 3.125Gbps = 320pS
- Pk-Pk Total Jitter = (0.65 * 320pS) = 208pS
- 10 GE specifies a jitter mask from 1.875MHz -to- 20MHz
- Thus, the “random” RMS Phase Jitter = (208pS)/224 = 930fs (1.875MHz <> 20MHz)

It is important to note that the acceptable magnitude of RMS Phase Jitter, and the integration band of interest, is often dictated by the PHY vendor's Refclock jitter specifications. And, the Refclock specifications from the PHY vendors often differ slightly from the industry standard and the application assumptions used in this section.

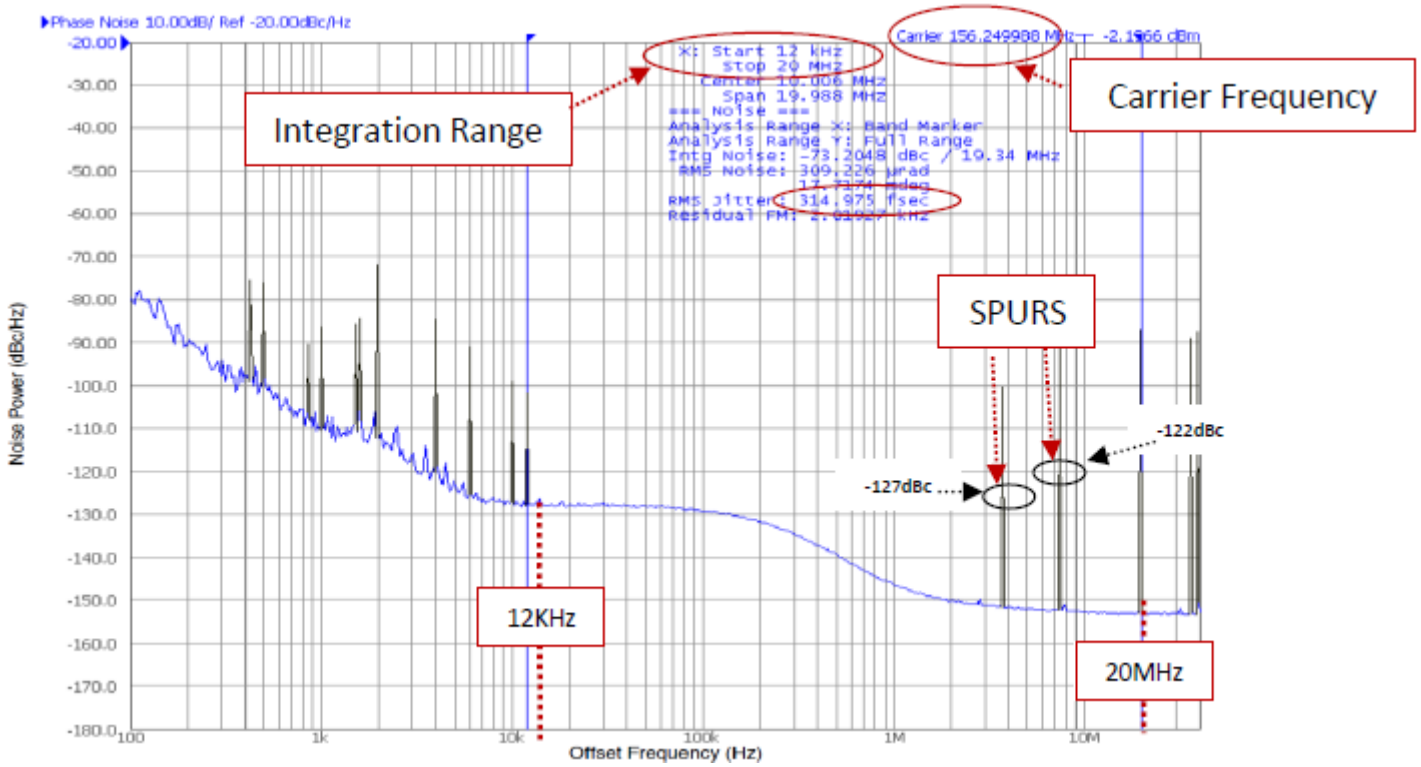
Phase Noise and RMS Phase Jitter are also relevant for **RF Signal Chain** design. Consider the use case of a high performance ADC in the signal chain path. Typically, **ADC** architectures implement a sample-and-hold (S&H) circuit that takes a snapshot of the ADC input at an instant in time, as shown in Figure 10.

Figure 10. Sample & Hold Circuit (ADC Use Case)



When the S&H switch is closed, the network at the input of the ADC is connected to the sample capacitor. At the instant the switch is opened, one half-clock period later, the voltage on the capacitor is recorded and held. Variation in the time at which the switch is opened is known as aperture uncertainty (i.e. jitter), and will result in an error voltage that is proportional to the magnitude of the sampling clock jitter and the sampled analog input signal slew rate. RMS Phase Jitter on the sampling clock is convenient expression of this jitter as it yields a single number useful for calculation of ADC SNR degradation due to aperture jitter. But, a detailed look at the actual spectral content in a PN Plot can be more instructive. Elevated wideband noise may not produce a poor RMS Phase Jitter result, but will degrade SNR. Close-in phase noise causes the fundamental signal to spread into adjacent frequency bins of an FFT, thereby reducing dynamic range. Broadband phase noise will uniformly elevate the noise floor throughout the Nyquist zone thus reducing the overall ADC SNR performance.

Renesas includes PN Plots and corresponding RMS Phase Jitter specifications in the datasheets of high performance PLL devices targeted for wired (i.e. SerDes) and wireless (i.e. RF Signal Chain) communications applications. As an example, consider the PN Plot and the corresponding RMS Phase Jitter measurement from the device datasheet of the Renesas **8T49N282 Universal Frequency Translator**, shown below. Note that the carrier frequency and integration range need to be specified for the quoted RMS Phase Jitter. With these qualifiers, the RMS Phase Jitter for a 156.25MHz carrier, integrated from 12KHz <<>> 20MHz, is measured to be ~ 314fs for this device.



A PN Plot is primarily intended to be a representation of random clock jitter. But, there are actually two main classifications of jitter; random jitter, and deterministic jitter. Random jitter is what is represented by the characteristic curve of the PN Plot, and is the primary measure of PLL jitter quality. But, real world applications always have some amount of deterministic jitter. And, this deterministic jitter can be quantified by the corresponding Spurs that show up in the PN Plot. For instance, the 8T49N282 example PN Plot (above) exhibits a -127dBc Spur centered at ~3.8MHz offset frequency and another -122dBc Spur centered at ~7.6MHz offset frequency.

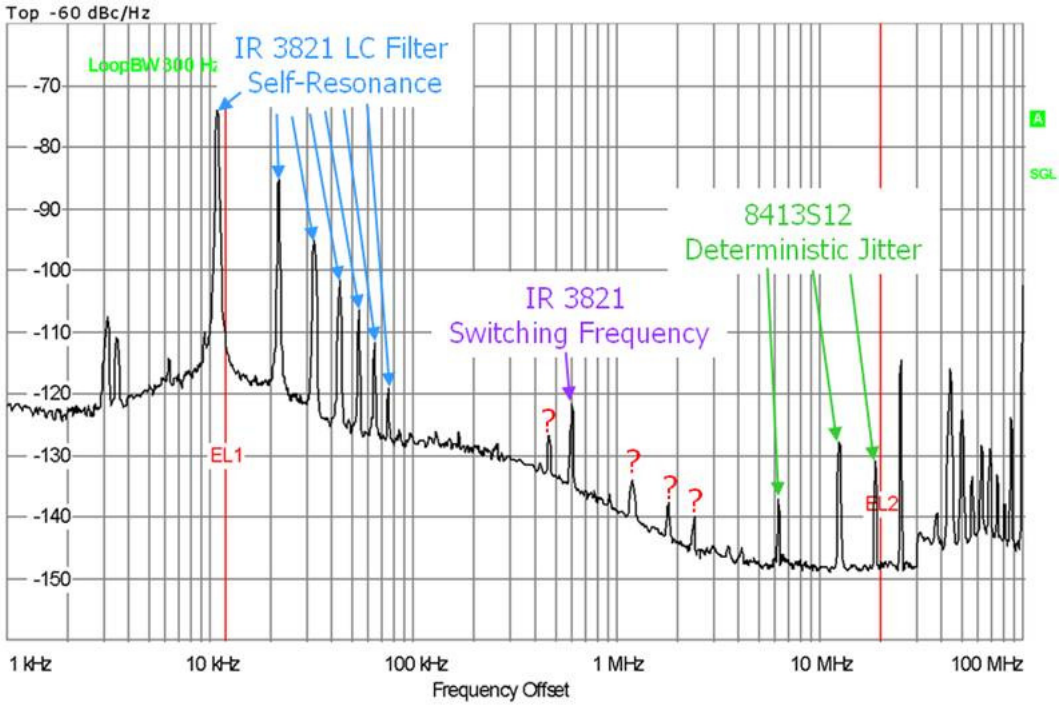
It is worth noting that it is not always easy to root cause the source of spurious content. Spurs can manifest themselves as second and third order sums and/or differences of different frequencies generated from within and/or outside the PLL device. Often times, the only way to root cause spurious content is to turn off (or modify) one at a time each potential modulation source and note any corresponding changes in the PN Plot.

Deterministic jitter is always bounded in amplitude and has a very specific (non-random) root cause. The potential sources of deterministic spurious content can be the PCB design itself, including;

- **Crosstalk** – The incremental inductance of a current carrying PCB trace will induce a magnetic field that can then impact a nearby parallel trace. This nearby trace will convert this magnetic field to an induced current superimposed on its otherwise typical drive characteristic. This induced current will affect the corresponding voltage, which will manifest itself as a deterministic jitter source that shows up as a Spur on a PN Plot. Component placement and PCB routing are important considerations for alleviating this type deterministic jitter
- **EMI** – RF Signal sources and AC power lines are examples of sources of Electromagnetic Interference (EMI). EMI sources can induce a noise current on a clock signal path. This phenomena and considerations are similar to that detailed for crosstalk.
- **Power Supply Switcher** – Often times Spurs seen on the PN Plot of a PLL output can be traced to the switching frequency of a Synchronous Buck Switching Regulator used as power source for the same PLL. Spurs will often show up at an offset frequency equal to the switcher frequency, and/or its harmonics. Understanding the frequency characteristic of the power supply switcher can be useful for designing a corresponding power supply filter circuit.

Consider the PN Plot of Renesas' 8413S12 used with a noisy power supply, as shown in Figure 11.

Figure 11. PN Plot, with Spurs

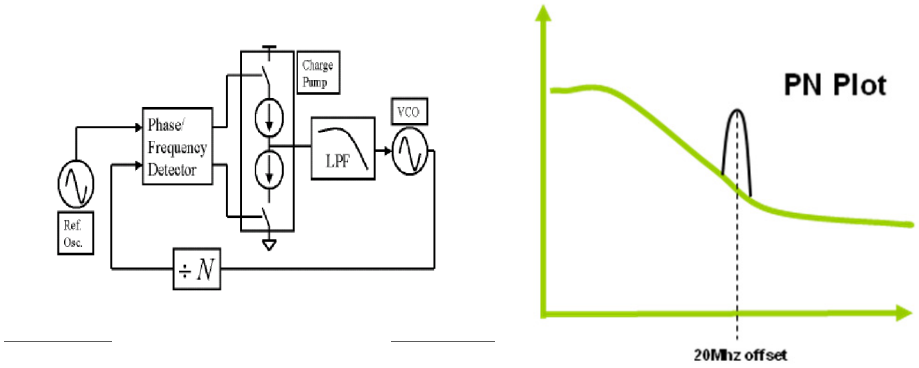


This design has significant spurious content observed at the output of the PLL. Root cause analysis made use of an EMI sniffer to trace back the Spurs to the Synchronous Buck Switching Regulator used in this design. Several layout modifications were implemented in order to reduce amount of noise that the 8413S12 picked-up from the regulator. Additional PCB design modifications were also implemented;

- The 11 KHz Spur, and its harmonics, is due to the regulator's L-C filter circuit. This Spur was moved to 1 KHz offset with simple passive component changes to the L-C filter, placing the Spur outside the 12KHz <<> 20MHz integration range of interest.
- The 600KHz Spur, and its harmonics, is a direct result of the regulator's switching frequency of ~600kHz. This Spur was alleviated by adding an additional 100µF cap to the power supply filter circuit at the PLL's analog power pin.

However, some of the spurious content comes from the 8413S12 device itself, as is also shown in Figure 11. Any clock synthesizer device with multiple outputs providing different frequencies can create undesirable sum and/or difference beat frequencies, which may be significant enough to show up as a Spur on a PN Plot. Also, the PLL architecture itself may explain some of the spurious content. Consider a simple Integer PLL architecture, as shown in Figure 12.

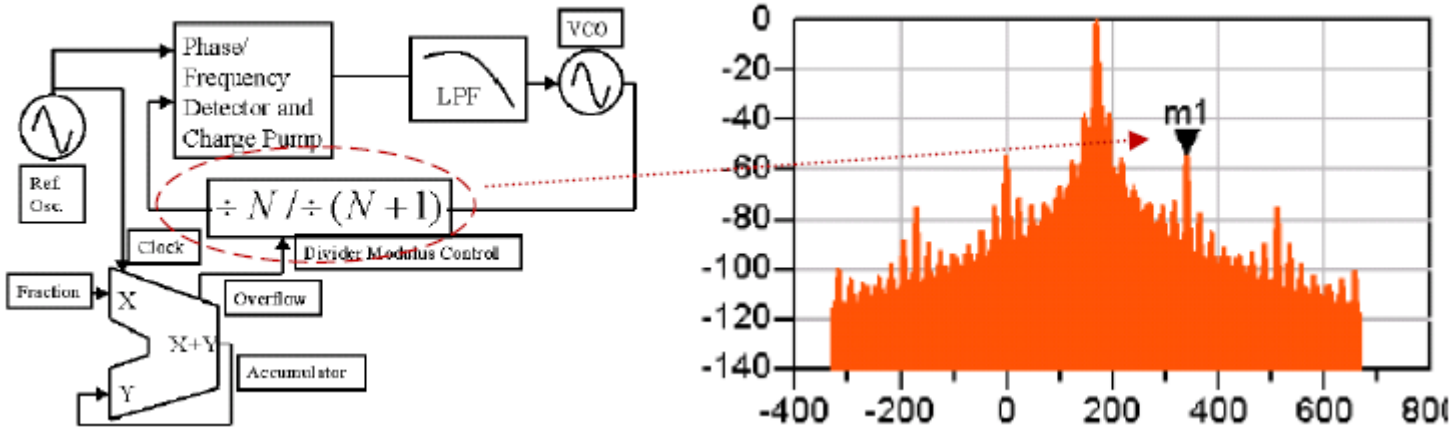
Figure 12. Integer PLL, corresponding Spur



The Phase Detector (PD) and the Charge Pump deliver +/- pulses to the Loop Filter, which integrates these pulses to generate the tuning voltage for the Voltage Controlled Oscillator (VCO). But, even when the PLL is locked, the Charge Pump still outputs small charges due to mismatches in PLL's positive & negative charge pump circuits and other non-idealities. These pulses can create a Spur at an offset frequency equal to the PD frequency, which is running at the rate dictated by the input clock reference.

This phenomenon is further complicated by more complex PLL designs being implemented today. For instance, the feedback divider of a Fractional-N (Frac-N) PLL, and/or the Frac-N output dividers, used in today's complex PLL designs can be the source of internally generated deterministic jitter. Consider the Frac-N PLL architecture, as shown in Figure 12. The feedback divider dynamically changes between N and N+1. The Accumulator sums the desired fraction to itself upon each reference cycle. And, when the accumulator overflows upon reaching a summed value of 1, then the feedback divider is changed to (N + 1) for one reference cycle. For a desired fractional value for the feedback divider of 0.1, then the overflow occurs every 10th reference cycle. If desired fraction is 0.5, then the overflow occurs every other reference cycle. The end result is that the “average” output frequency is equal to (N + fraction) * Fref.

Figure 13. Frac-N PLL, corresponding Spurs



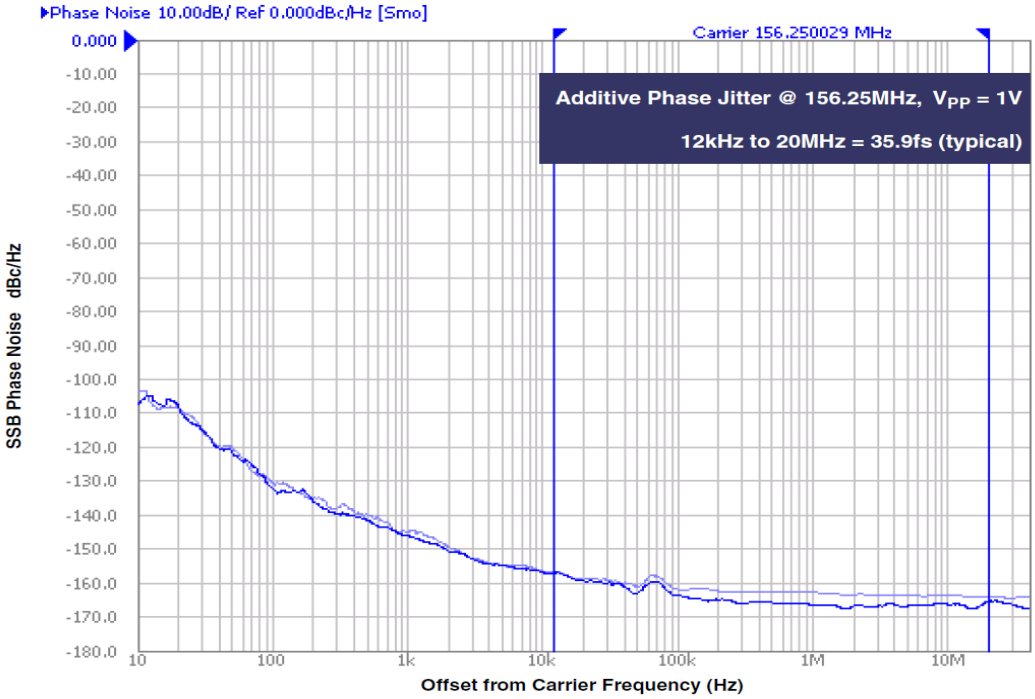
The benefit of a Frac-N PLL is that it is not limited to integer multiples of the input reference frequency, as it can generate any output frequency from any input frequency. But, the modulating feedback divider is now an additional deterministic noise source, and Spurs may be generated at frequencies related to this modulation frequency. As an example, if desired fraction is 0.5, then the feedback divider alternates between N and (N+1) every other cycle, and a corresponding Spur can occur at an offset equal to ½ of the reference frequency. This spurious behavior can become more relevant for a given application when desired fraction is close to 0 or 1. This is because the modulation rate of the feedback divider becomes very low. If, for example, the desired fraction is 0.01, then the PLL would divide by (N + 1) for only 1 out of every 100 reference cycles. The characteristic offset freq of noise would be 1/100 of the reference freq. This is often well below the loop bandwidth of the PLL and would show up on the device output.

To address this issue, advanced PLL designs implement Delta-Sigma-Modulation (i.e. DSM) techniques to improve upon the simple Frac-N PLL architecture. The basic structure is same as that for Frac-N PLL, but DSM is used to “rapidly” shift between many divider values. This way, the divider value never stays at the same setting for more than a few cycles, which keeps the divider modulation rate very high. So, the corresponding Spurs are pushed out to high frequencies which are easily filtered out by the loop bandwidth of the PLL.

The relevance of any spurious content depends on the sensitivity of the given application, as well as on the magnitude of the Spurs and the offset freq at which these Spurs occurs. It is very important that spectrum analyzer equipment used to generate a PN Plot is configured with “spurs turned on” in order to get the complete jitter representation. If the deterministic jitter source is external to the PLL, then identification of such may allow for an appropriate filter design, layout modifications, etc. If the source is the PLL itself, then configuration changes to the PLL may be tried to eliminate the offending Spur, or to at least move it outside of the integration range of interest.

The “additive” jitter for a Fan-Out Buffer (FOB) is often overlooked because it is considered negligible. And many FOBs do not even have jitter specified in their datasheet. However, this is something that should be quantified, particularly for high performance designs. But, simply subtracting the output jitter from the input jitter is not the correct way to specify “additive” RMS Phase Jitter. Consider the PN Plot of a FOB, as shown in Figure 14.

Figure 14. PN Plot of a FOB



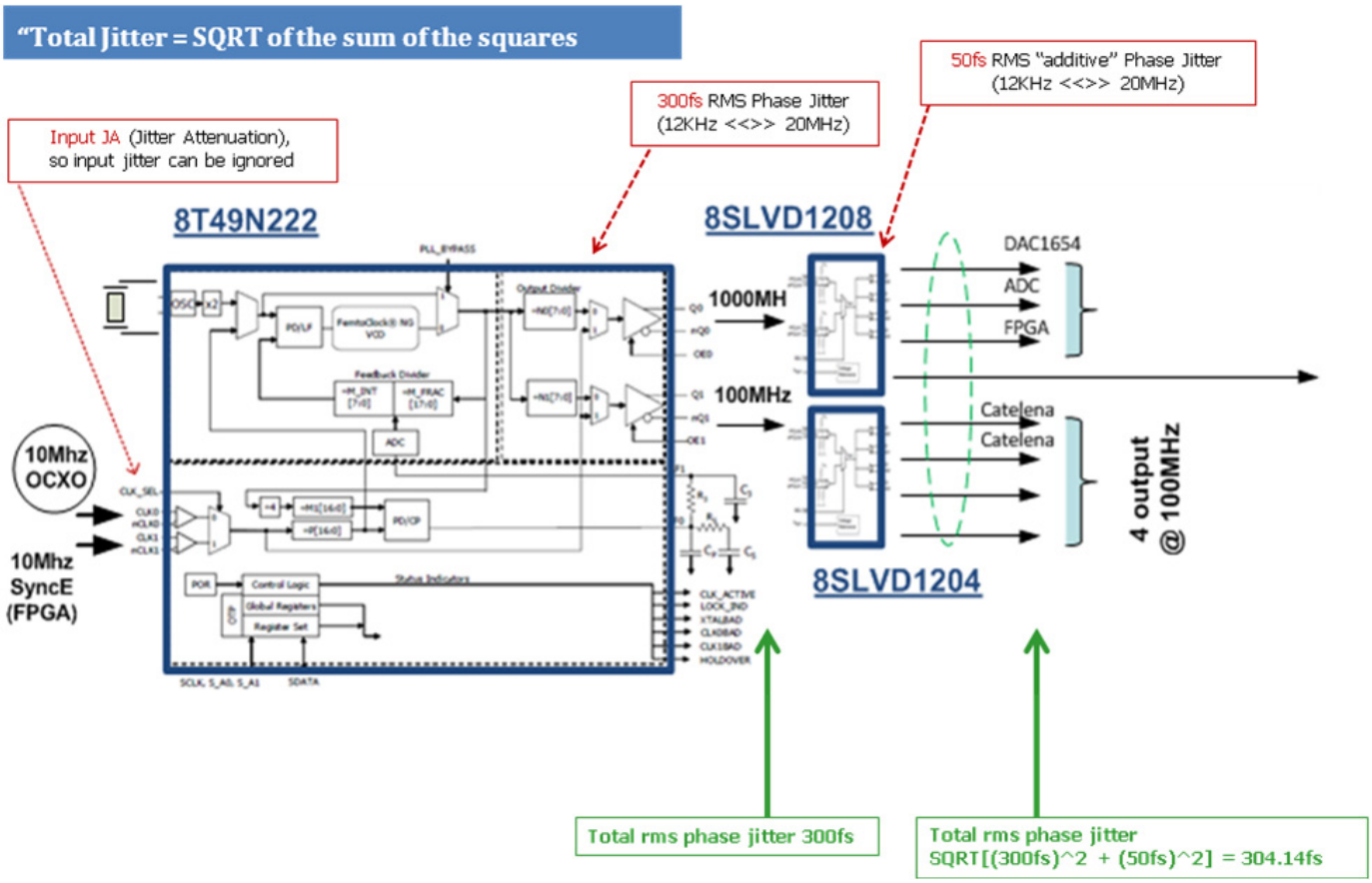
To get the correct number, one needs to square the RMS Phase Jitter of the output (light blue PN curve), subtract the square of the RMS Phase Jitter of the input (dark blue PN curve), and then take the square root of the result; effectively performing the ***Square Root of the Difference of the Squares***. Note that this value is still dependent on both the frequency and the integration range of interest. So, both must be qualified when specifying the “additive” RMS Phase Jitter of a Fan-Out Buffer. This same Figure 14 is actually the PN Plot from the device datasheet of the **Renesas 8SLVP1208 Low Phase Noise 1:8 LVPECL Fan-Out Buffer**. For this device, the “additive” RMS Phase Jitter for a carrier frequency of 156.25MHz, integrated from 12KHz-to-20MHz, is ~ 35.9fs.

Quantifying the “additive” RMS Phase Jitter in this manner allows the total RMS Phase Jitter to then be correctly calculated as the Square Root of the Sum of the Squares. Consider the 1GHz ADC/DAC Clock Distribution example shown in Figure 15. In this example, the first stage PLL has 300fs of intrinsic jitter generation. And, the subsequent FOB driven by this PLL has an “additive” RMS Phase Jitter specification of 50fs. The resultant total RMS Phase Jitter of the 1GHz clock provided to the ADC is calculated as follows:

total RMS Phase Jitter = SQRT[(300fs)^2 + (50fs)^2] = 304.14fs.

It is worth noting that the “additive” effect of the FOB used in this example is still very low, adding only ~4fs to the total RMS Phase Jitter.

Figure 15. “additive” RMS Phase Jitter (1GHz ADC/DAC Clock Distribution)



Conclusion/Summary

Three commonly used types of jitter were discussed in this paper:

- **Period Jitter –** the time difference of the jittered clock period minus the nominal clock period. This is a time domain measurement, and can be expressed as RMS Period Jitter measured across 10K cycle. If expressed as P2P Period Jitter, then this must also be qualified with an assumed BER, otherwise meaningless. This type of jitter is relevant for worst case timing analysis in Synchronous Interface and Synchronous Logic design.
- **High Frequency Jitter** – typically quantified as C2C Jitter, which is the difference in the Period for successive cycles. This is a time domain measurement, and is measured across 1K cycles, and can be expressed as Peak value, and sometimes a P2P value. This type of jitter is relevant for Micro Complex, Motherboard Design, and Spread Spectrum generated clocks.
- **Frequency Domain Jitter** – typically measured as Phase Noise and is a measure of the instantaneous phase deviations from the ideal; effectively a frequency domain quantification of clock jitter. Typically, this type of jitter is expressed as a PN Plot that shows the dBc values at the various carrier frequency offset points. It can also be quantified as RMS Phase Jitter, qualified with the application appropriate integration band of interest. This type of jitter is relevant for High Speed Serial communication and ADC use cases, to name just two. In addition, spurious content from a PN Plot should also be quantified in order to characterize the total clock jitter.

Revision History

Revision Date	Description of Change
March 1. 2022	Corrected minor spelling typo.
April 24, 2014	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2021 Renesas Electronics Corporation. All rights reserved.