TI Technology Days 2010

High Speed Clocking Basics

Ioannis Tsikouris-Willgers

Business Development Manager

ioannis@ti.com

Texas Instruments

Agenda

- Introduction
- Skew
- Jitter & phase noise
- Jitter Optimization
- Solution Examples

Skew

Skew

Skew is the time delta between multiple clk signals which are ideally expected to be in phase with one another.

Types of skew:

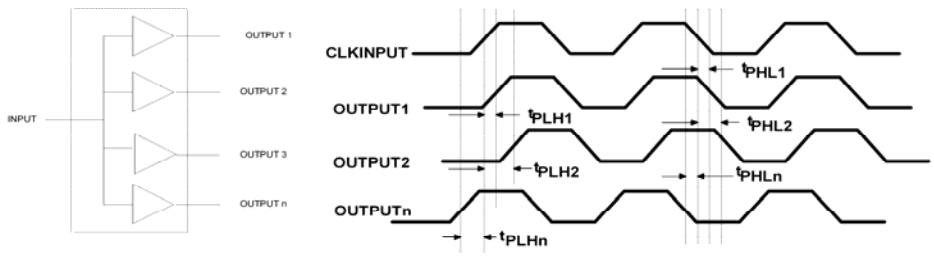
- ***Output skew***
- ***Part-to-part skew***
- ***Pulse skew***
- ***Process skew***
- ***Limit skew***
- ***Bank skew***
- ***Input skew***
- ***Board skew***
- ***Tracking skew***

Output Skew

Output skew ($t_{sk(O)}$): the difference between propagation delays of any two outputs of the same device at identical transitions (i.e., compares t_{pLH} versus t_{pLH} or t_{pHL} versus t_{pHL} for any two outputs). For example, if the propagation delay of the fastest output (t_{pLHn}) is 2.0ns and that of the slowest output is (t_{pLH1}) 2.165ns, then the output skew is:

$$t_{sk(O)} = t_{pLHn} - t_{pLH1} = 165 \quad ps$$

also known as pin-to-pin skew

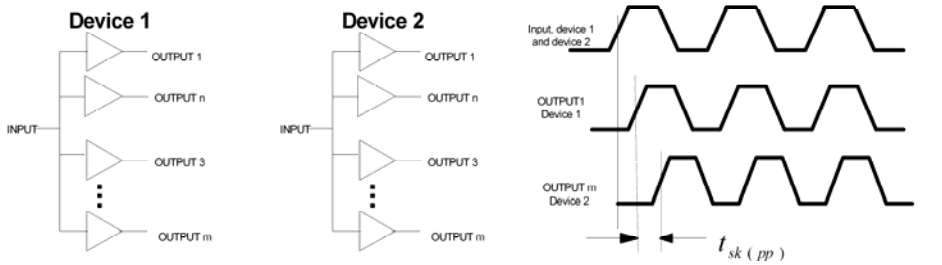


OUTPUT SKEW, tsk(O)

Part-to-Part Skew

Part-to-part skew ($t_{sk(pp)}$): similar to output skew except that it applies to two or more identical devices. Therefore is defined as, the magnitude of the difference in propagation delays between any specified outputs of two separate devices operating at identical conditions. The devices must have the same input signal, supply voltage, ambient temperature, package, and same load, environment, etc.

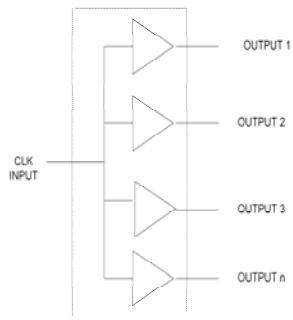
also known as package skew and device-to-device skew

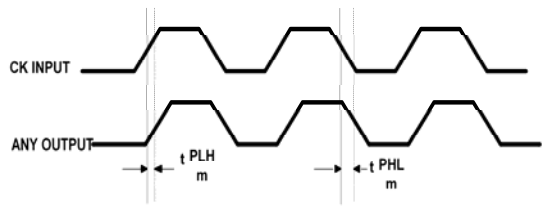


Pulse Skew

Pulse skew: the magnitude of the time difference between the high-to-low (t_{PHL}) and the low-to-high (t_{PLH}) propagation delays.

Also known as pulse width distortion or duty cycle skew



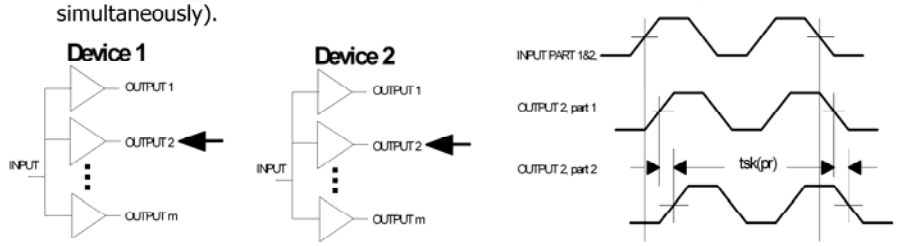


$$\textit{Pulse skew,}\quad t_{sk(p)} = \left| t_{PLH} - t_{PHL} \right|$$

Process Skew

Process skew: the difference in propagation delay times between corresponding outputs on any two like devices when both devices operate under identical conditions. Quantifies skew due to process variation in the manufacturing process (skew caused by lot-to-lot variation). It excludes variations in supply voltage, operating temperature, output loading, input edge rate, input frequency, etc.

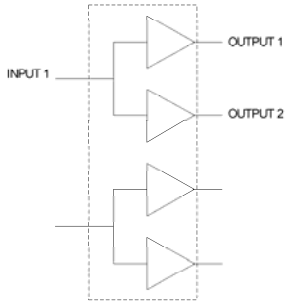
Process skew is generally specified and production tested under firm/stable conditions (e.g., Vcc=3.3V, TA=25C, CL=25pF, all inputs switching simultaneously).

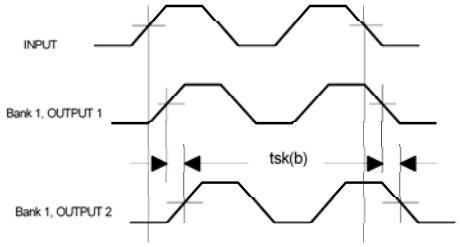


process skew, $t_{sk(pr)}$

Bank Skew

Bank skew: the output skew between outputs (at same bank), of a single device with a single driving input terminal. The main difference between bank skew and output skew is that, the latter is the worst-case delta between outputs in any output bank.

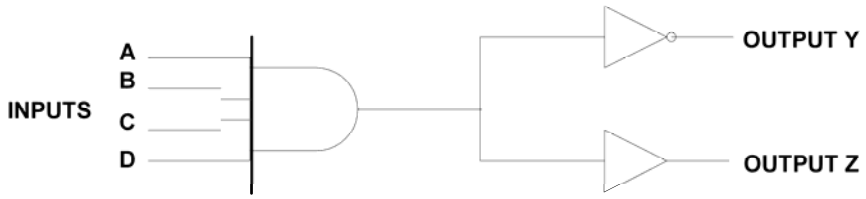




Bank skew, $t_{sk(b)}$

Input Skew

Input skew ($t_{sk(i)}$): the difference between any two propagation delay times that originate at different inputs and terminate at a single output. Input skew describes the ability of a device to manipulate (stretch, shrink, or chop) a clock signal. This is typically accomplished with a multi-input gate wherein one of the inputs acts as a controlling signal to pass the clock through. $t_{sk(i)}$ described the ability of the gate to shape the pulse to the same duration regardless of the input used as the controlling input.

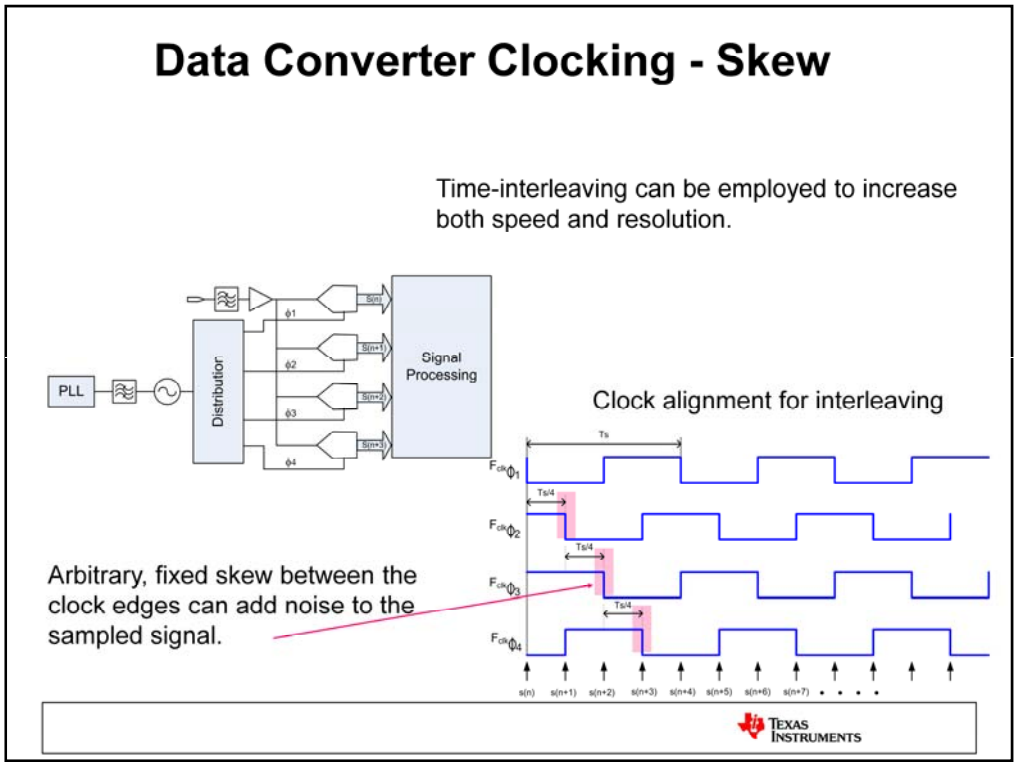


Example of a Gate with Input Skew

Limit Skew and Board Skew

Limit skew ($t_{sk(l)}$): the difference between the greater of the maximum specified values of t_{PLH} and t_{PHL} the lesser of the minimum specified values t_{PLH} and t_{PHL}. Limit skew is observed directly on a device, rather it is calculated from the datasheet limits of t_{PLH} and t_{PHL}. $t_{sk(l)}$ quantifies how much variation in propagation delay times will be induced by operation over the entire ranges of V_{CC}, T_A, output load, process variation and any other specified operating conditions.

Board skew ($t_{sk(pcb)}$): is introduced into the timing system by unequal trace lengths and loads. It is independent of skew generated by the clock driver.

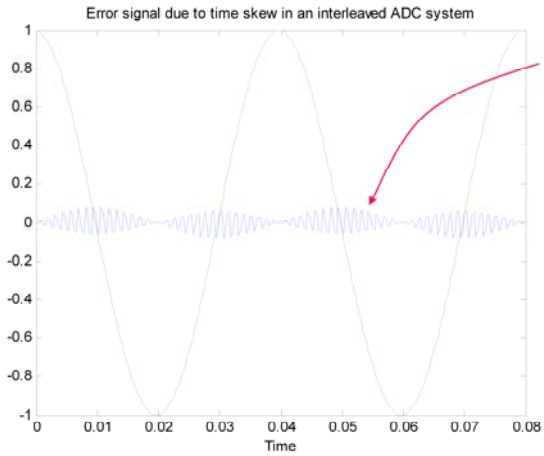


- Clock skew is a relative, *fixed* delay between two or more clock signals.
- It becomes relevant in systems that require fixed relationships between clocks.
  - Examples are designs that use multiple ADCs to capture data, or in synchronous data clocking.

- The ADC applications tend to fall into two categories:
  - Interleaved ADC sampling for the purpose of achieving higher sample rates AND higher resolution (test and measurement, instrumentation)
  - Parallel multiple event capture (medical imaging, instrumentation, test & measurement)

- Interleaved sampling systems are used when off-the-shelf ADCs offer the required resolution but not the required sampling rate.
  - These systems use multiple ADCs clocked at the same rate. The effective increase in sample rate is achieved by a round-robin sampling scheme.
  - The timing diagram shows the ideal relationship between the clocks.
  - Each ADC runs at a rate of Fs. However, by staggering the clock phase between ADCs, the effective sampling rate becomes M * Fs, M= number of ADCs.
  - The ADC sample streams can be multiplexed into a single sample stream if needed

Error due to Skew in a waveform captured via interleaved sampling



Skew error =

ideal signal – signal with skew

This plot shows a sinusoid along with the error component due to skew.

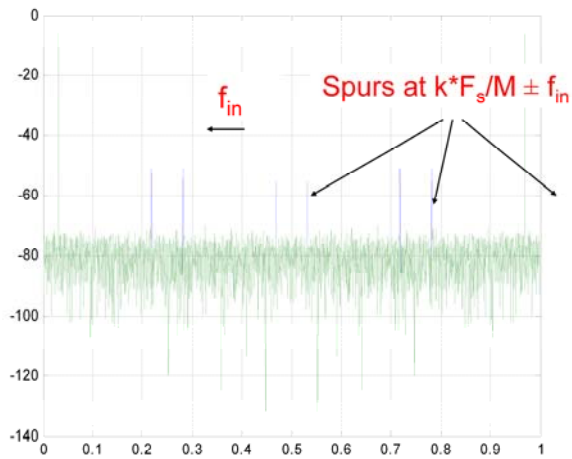
•If we could subtract the sampled signal with skew (green trace) from the ideal signal,

we would be left with the error signal (noise) due to clock skew, illustrated by the blue trace in the plot shown above.

•Note that the error reaches a maximum where the signal slope is maximum, and vice versa.

•How is this manifested in the frequency domain?

Skew in the frequency domain – interleaved sampled signal

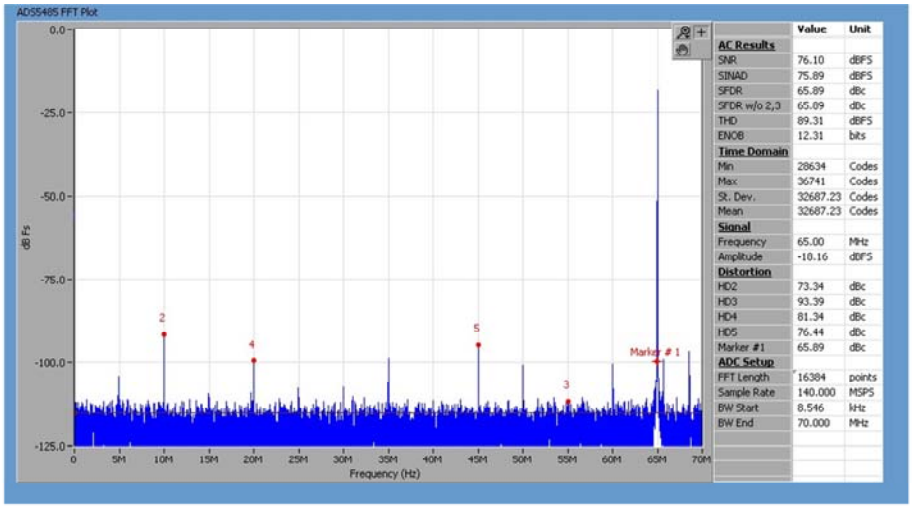


F_s = sample frequency

M = # of ADC channels

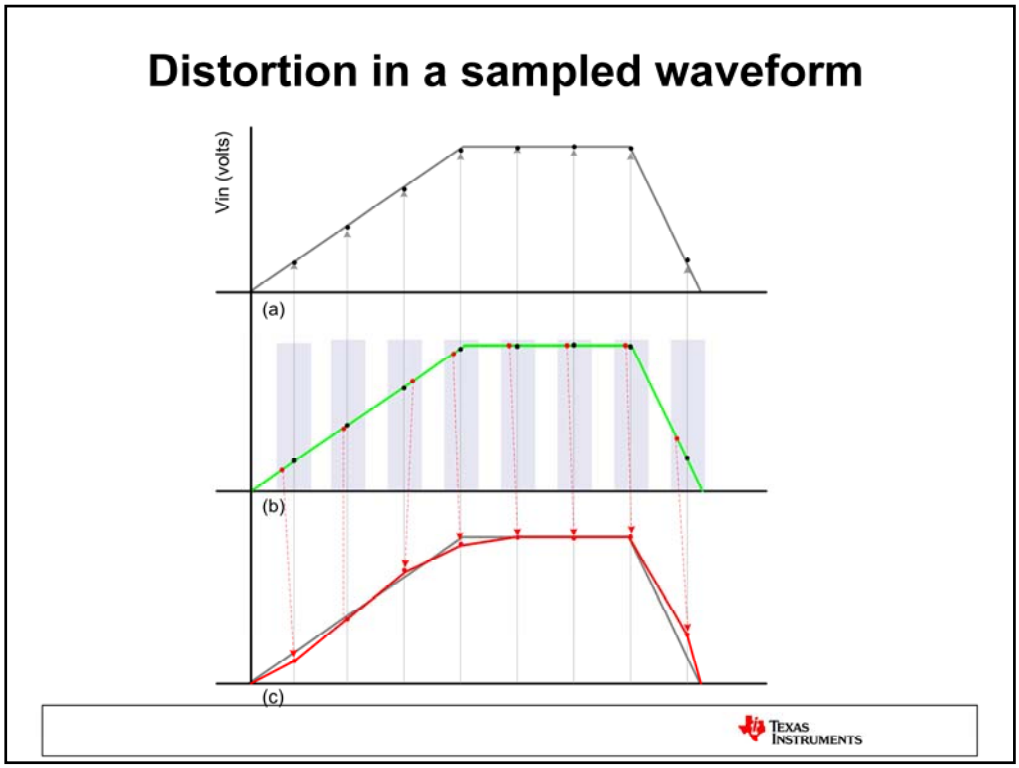
k = 0,…,M-1

- This plot shows that spurs are generated when clock skew is present.
  - The spur frequencies (blue traces) are determined by the frequency of the input signal and the number of ADC channels.
  - The spur amplitudes are determined by the frequency of the input signal and the relative values of the skew.


AD5548S FFT Plot
0.0
-25.0
-50.0
-75.0
-100.0
-125.0
dB Fs
0
5M
10M
15M
20M
25M
30M
35M
40M
45M
50M
55M
60M
65M
70M
Frequency (Hz)
2
4
5
3
Marker # 1
Value
Unit
AC Results
SNR 76.10 dBFS
SINAD 75.89 dBFS
SFDR 65.89 dBc
SFDR w/o 2,3 65.09 dBc
THD 89.31 dBFS
ENOB 12.31 bits
Time Domain
Min 28634 Codes
Max 36741 Codes
St. Dev. 32687.23 Codes
Mean 32687.23 Codes
Signal
Frequency 65.00 MHz
Amplitude -10.16 dBFS
Distortion
HD2 73.34 dBc
HD3 93.39 dBc
HD4 81.34 dBc
HD5 76.44 dBc
Marker #1 65.89 dBc
ADC Setup
FFT Length 16384 points
Sample Rate 140.000 MSPS
BW Start 8.546 kHz
BW End 70.000 MHz

Clocking HS ADCs

Texas Instruments



Consider the three plots above.

The center plot represents a trapezoidal pulse (perhaps as captured on an oscilloscope). It is a continuous waveform (i.e. for each position in time on the horizontal axis there is a corresponding voltage on the horizontal axis).

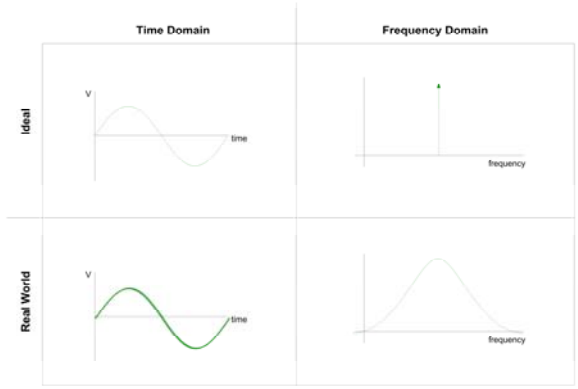
The Black dots represent ideal sample points and these points are transposed to the plot on the top. The original waveform can be determined by connecting the points.

Superimposed over the center plot are regions which represent a possible location of a sampling point if there is jitter on the sample clock. While the top plot contains points which appear in the exact center of these regions, the actual sample point can appear randomly anywhere in the region. Some possible sample points are show as red dots on the center plot. These values are transposed to the plot on the bottom. Notice the value recorded is not the correct voltage level because it is sampled at the incorrect location. It is transposed to the point in time in which the sample was ideally to be taken (in the center of the sampling region). This results in the signal being distorted as shown by the red trace on the bottom plot.

Phase Noise

$$v(t) = \sin(2\pi f t)$$

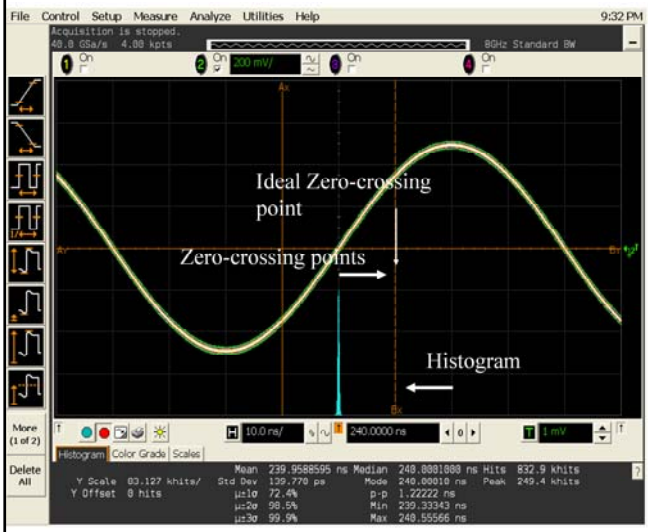
$$\tilde{v}(t) = \sin(2\pi f t + \phi(t))$$



Phase Noise Definition:

Rapid, short-term, random fluctuations in the phase of a waveform, caused by time domain instabilities.

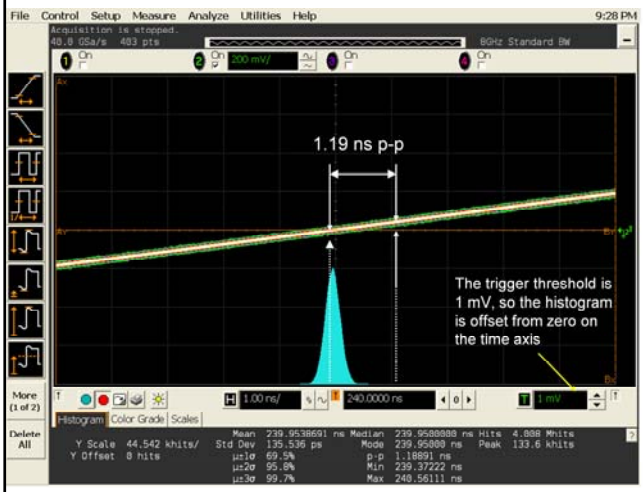
Phase Noise in the time domain (1)



A digital scope is useful for viewing jitter in the time domain but is not adequate for measuring jitter due to limited time resolution

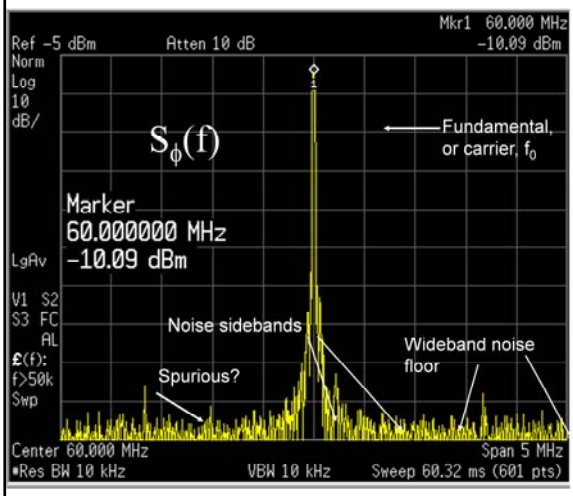
Texas Instruments

Phase noise in the time domain (2)



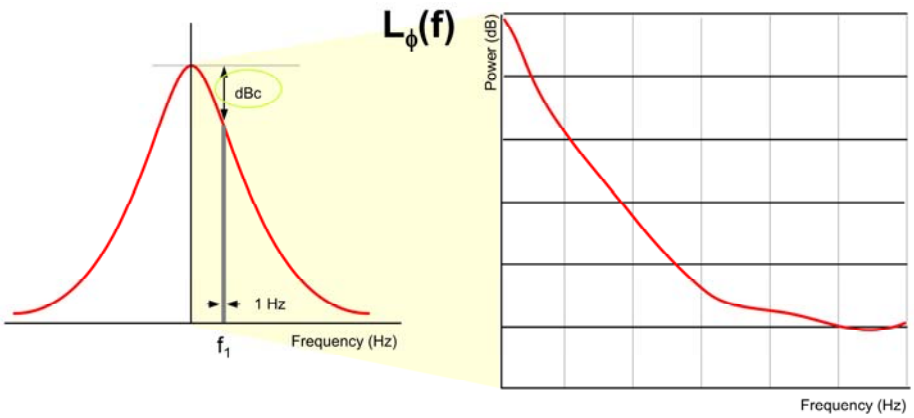
Expanding the time scale shows the Gaussian-like distribution of the jitter around the zero-crossing point

Phase Noise in the Frequency Domain



In the frequency domain, the noisy clock is not an ideal impulse. It appears as a spike with noise side bands that cause the base to spread.

Phase Noise Power Spectral Density (PSD)



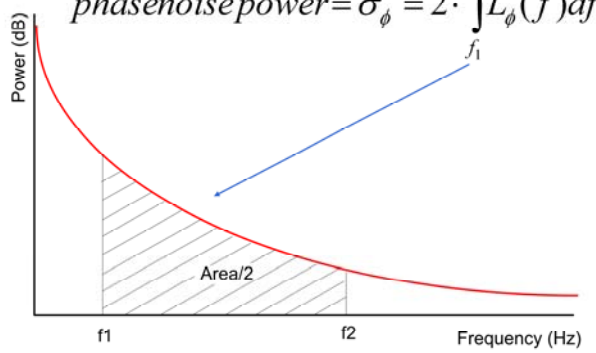
Phase noise power is measured in a 1 Hz bandwidth at offsets from the carrier that cover a specified range of frequencies

- Conventionally, the power spectral density (PSD) of the phase noise spectrum is specified in units of dBc/Hz.
- At a particular offset frequency (f_1), the spectrum analyzer measures the power in a 1 Hz bandwidth. Call this power P_{f1} Watts.
- To convert this to dBc, we take 10 times the log of the ratio of the carrier power (in Watts) to the phase noise power (in Watts):
  - $P_{dBc} = 10*\log(P_{carrier}/P_{f1})$
- By normalizing the phase noise power in this way, we can compare different oscillators without being concerned about the absolute power level of the oscillator.

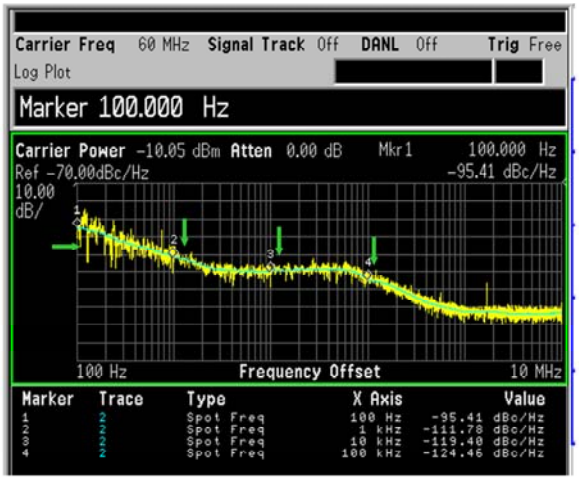
Integrated Phase Noise Power

$$power = \int_{-\infty}^{+\infty} S_{PSD}(f)df$$

$$phasenoisepower = \sigma_{\phi}^{2} = 2 \cdot \int_{f_1}^{f_2} L_{\phi}(f)df$$

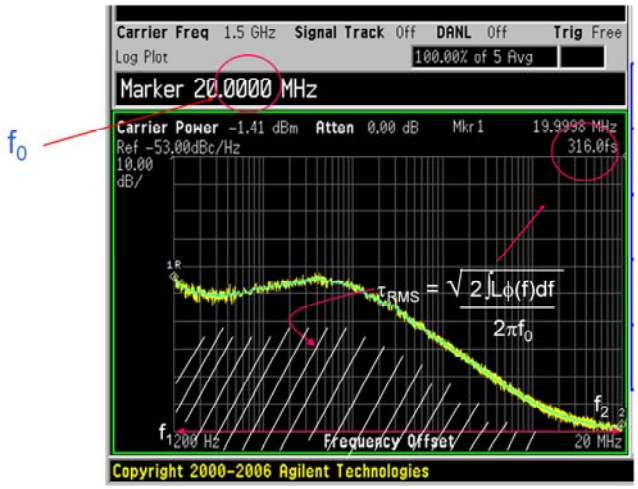


Phase Noise: Frequency Domain



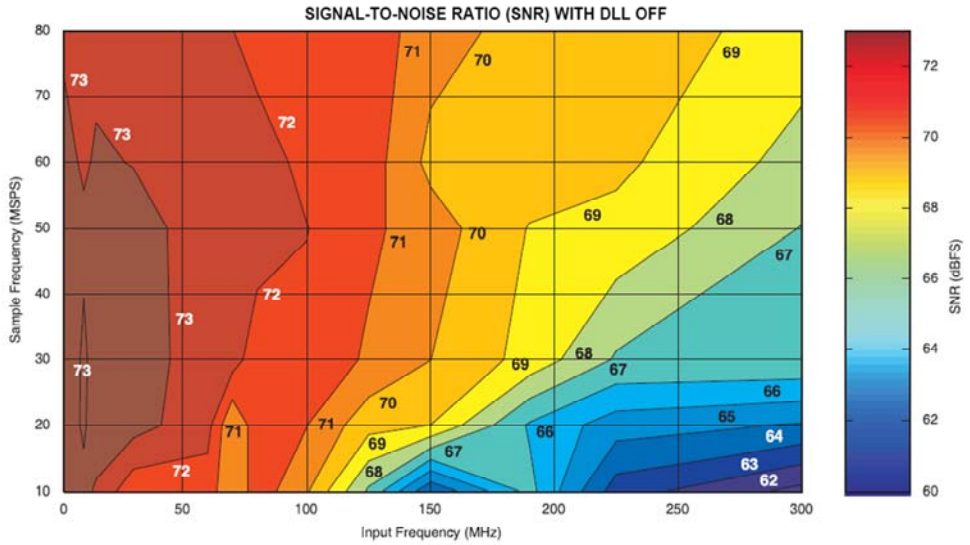
Specialized spectrum analyzers allows us to view and measure the phase noise PSD.

Converting Phase Noise to RMS Jitter



Texas Instruments

Typical SNR Diagram



TEXAS INSTRUMENTS

Signal to Noise Ratio

Assuming a sinusoidal input for V_{in} we have,

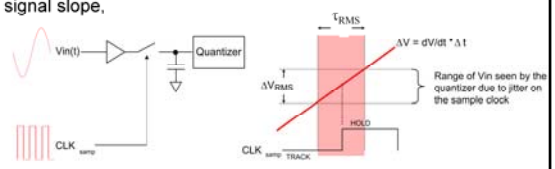
$$v(t) = V_o \sin 2\pi f t$$

Differentiating with respect to time yields the signal slope,

$$\frac{dv}{dt} = 2\pi f V_o \cos 2\pi f t$$

Taking the RMS value,

$$\frac{dv}{dt}_{rms} = \frac{2\pi f V_o}{\sqrt{2}} = \frac{\Delta V_{RMS}}{\tau_{RMS}}$$



Therefore, the RMS error voltage due to jitter is

$$\Delta V_{RMS} = \frac{2\pi f V_0 \tau_{RMS}}{\sqrt{2}}$$

Signal to noise ratio (SNR) is defined by,

$$SNR = 20\log_{10}\left[\frac{signal}{noise}\right] = 20\log_{10}\left[\frac{V_o/\sqrt{2}}{\Delta V_{rms}}\right]$$

Therefore, the SNR component due to jitter is:

$$SNR_{jitter} = 20\log_{10}\left[\frac{1}{2\pi f_{\max}\tau_{RMS}}\right]$$

Texas Instruments

The slide above shows the derivation of theoretical data converter SNR (Signal to Noise Ratio) due to jitter. It provides the following insight:

1. The frequency of the sample clock is not a factor.
2. The sample frequency of the data converter is not a direct factor, the input bandwidth of the data converter is (fmax).
3. The jitter value (tj) in Equation (6) represents total jitter which is the root sum squared of the sample clock jitter and the converter aperture jitter.

For many communications systems the bandwidth of the channel is fixed; however, the channel capacity needs to grow. Why?

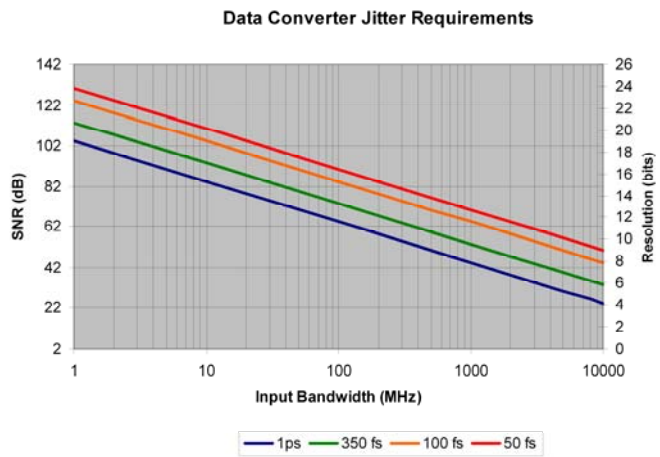
1. Multimedia content requires more capacity
2. High definition content requires more capacity
3. Growing populations and an expanding subscriber base requires more capacity.

In fixed bandwidth applications, SNR is the only lever engineers have to control channel capacity? What types of applications have fixed bandwidths?

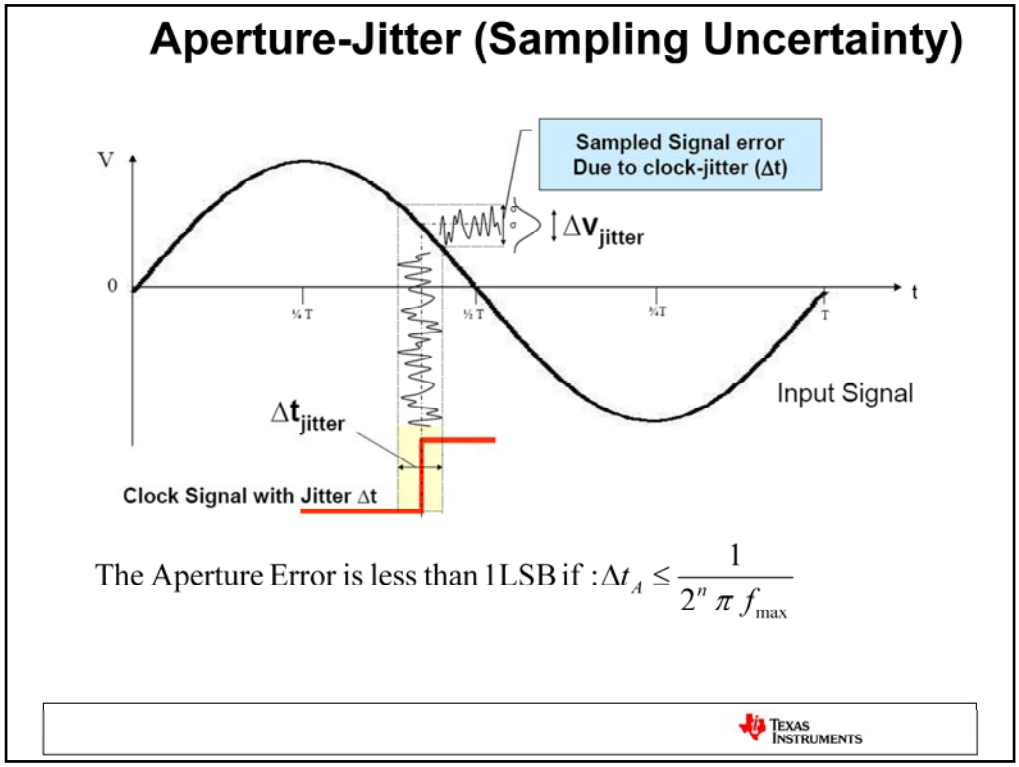
Data Converter Clocking

$$SNR_{ADC} = 6.02N + 1.76$$

$$SNR_{jitter} = 20\log_{10}\left[\frac{1}{2\pi f_{max}\tau_{RMS}}\right]$$



- This figure plots SNR vs. input frequency for fixed values of jitter.
- We can note a couple of things about this graph
  - Lower jitter = better SNR
  - For a constant jitter, SNR decreases as the frequency of the sampled signal increases



Aperture uncertainty is a key ADC concern when performing IF sampling. The terms aperture jitter and aperture uncertainty are synonymous and are frequently interchanged in the literature. Aperture uncertainty is the sample-to-sample variation in the encoding process. It has three distinct effects on system performance.

First, it can increase system noise.

Second, it can contribute to the uncertainty in the actual phase of the sampled signal itself giving rise to increases in error vector magnitude.

Third, it can heighten intersymbol interference (ISI).

If the aperture time varies by the time Δta error is caused which is equal to the change Δv in the voltage. This results into a degradation of the SNR of the ADC. To calculate the maximum time Δta which results into an error less than 1LSB, a sine wave with maximum frequency fmax as input is considered.

V(t) = Vp . sin(ωt)

The slope is

dv/dt= Vp . ω . cos (ωt)

Max. slope occurs when cos (ωt) = 1 or at the zero-crossing point thus max (dv/dt) = Vp . ω .

In order to limit the error in the change of the voltage to less than 1LSB:

Δta = ΔV / (Vp . ω)

1LSB = 2Vp / 2^n

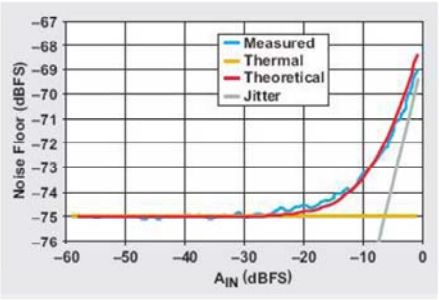
All above results that

Δta ≤ 1 / (2^n . Π . fmax)

ADC – Noise Sources

Three major contributors:

- **1. Quantization Noise**
  - SNR = 6.02B . n + 1.76dB
- **2. Thermal Noise**
  - (kTC, switching noise)
- **3. Jitter (total)**



$$SNR_{[dBc]} = 10\log_{10}\left[\frac{\left(\frac{A}{\sqrt{2}}\right)^2}{Thermal + Quantization + \frac{1}{2}\left(\frac{2\pi A}{T}\right)^2 Jitter_{total}^2}\right]$$

For a given ADC, below a certain input amplitude, the SNR is dominated by other sources like thermal noise and quantization noise and jitter becomes irrelevant. One way of showing this is plotting the noise floor in dBFS with respect to the Ain. The figure above shows a real noise-floor example of the ADS5542 operating at 78 MSPS and 230-MHz input.

This effect is especially important when we compute the theoretical effects of jitter in our system and try to choose the right ADC and clocking. Specifically, it shows that for certain systems, ADC data taken at –1 dBFS may give us too pessimistic a result, as the signal may seldom reach those levels.

Clock Considerations

- **Maximum SNR is Jitter Limited**

$$SNR_{[dBc]} = -20\log_{10}\left[2\pi f_{in}\, t_{Jtotal}\right]$$

- **Clock 'Quality' is a major factor for Undersampling *!***
- **Very low Jitter required to maintain good SNR**
- **Independent Jitter Sources are summed by Root-Sum Square**

$$t_{jtotal} = \sqrt{\left(t_{J_{ADC}}^2 + t_{j_{Ext}}^2\right)} \quad [t_{in}\, ps]$$

- **Examine Phase Noise (jitter) of clock source.**
- **Differential configuration allows for doubling of**
- **Amplitude and common mode noise rejection.**

Texas Instruments

Noise on the ADC power supply can directly affect the internal clock circuitry and may lead to increased jitter.

Where can I get the data?

ADS5424

ADC Internal Jitter = 150fs

TEXAS INSTRUMENTS
www.ti.com

SLWS157A – JANUARY 2005 – REVISED MAY 2005

TIMING CHARACTERISTICS(3)

Over full temperature range, AV_{DD} = 5 V, DRV_{DD} = 3.3 V, sampling rate = 105 MSPS

PARAMETER	DESCRIPTION	MIN	TYP	MAX	UNIT
Aperture Time					
t_A	Aperture delay		500		ps
t_J	Clock slope independent aperture uncertainity (jitter)		150		fs
k_J	Clock slope dependent jitter factor		50		µV
Clock Input					
t_{CLK}	Clock period		9.5		ns
t_{CLKH}(1)	Clock pulsewidth high		4.75		ns
t_{CLKL}(1)	Clock pulsewidth low		4.75		ns
Clock to DataReady (DRY)					
t_{DR}	Clock rising 50% to DRY falling 50%	2.8	3.9	4.7	ns
t_{C_DR}	Clock rising 50% to DRY rising 50%		t_{DR} + t_{CLKH}		ns
$t_{C_DR_50\%}$	Clock rising 50% to DRY rising 50% with 50% duty cycle clock	7.6	8.7	9.5	ns
Clock to DATA, OVR(4)					
t_r	Data V_{OL} to data V_{OH} (rise time)		2		ns
t_f	Data V_{OH} to data V_{OL} (fall time)		2		ns
L	Latency		9		Cycles
$t_{su(C)}$	Valid DATA(2) to clock 50% with 50% duty cycle clock (setup time)	1.8	3.4		ns
$t_{h(C)}$	Clock 50% to invalid DATA(2) (hold time)	2.6	3.6		ns
DataReady (DRY) to DATA, OVR(4)					
$t_{su(DR)_50\%}$	Valid DATA(2) to DRY 50% with 50% duty cycle clock (setup time)	1.8	2.6		ns
$t_{h(DR)_50\%}$	DRY 50% to invalid DATA(2) with 50% duty cycle clock (hold time)	3.9	4.4		ns

TEXAS INSTRUMENTS

So, considering all the above, Texas Instruments has started to include on its datasheet (see ADS5423/4) two values that the circuit designer can use to compute his estimated final jitter, depending not only on his/her clock jitter, but also on the shape of his/her clock:

1. A term that is constant, independent of clock amplitude. This would basically be the N3 of our model.

2. A term that divided by the clock slope will account for this new addition to the model.

Previous datasheets show a single jitter term that would correspond to a unique clock shape. Circuit designer should be careful with specmanship from some manufacturers who could specify the ADC clock jitter under ideal conditions (with input clock slope very fast).

Calculation for Clock Jitter

- **Solve for the RMS jitter required of the system in order to get at least 70dB SNR at 100MHz f_{in}**

$$SNR_{[dBc]} = -20\log_{10}\left[2\pi f_{in} t_{jtotal}\right] \Leftrightarrow$$

$$\Leftrightarrow t_{jtotal} = \frac{10^{\left(\frac{-SNR_{[dBc]}}{20}\right)}}{2\pi f_{in}} = \frac{10^{\left(\frac{-70}{20}\right)}}{2\pi 100^{6}} = 503\,fs$$

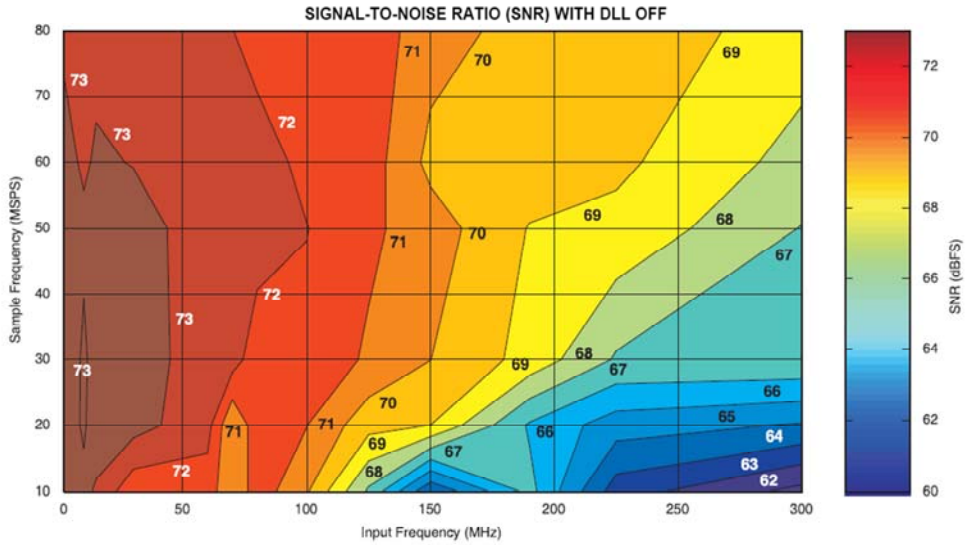
- **now solve for required Clock Jitter with 105fs of internal ADC aperture jitter**

$$t_{jtotal} = \sqrt{\left(t_{J_{ADC}}^{2} + t_{j_{Ext}}^{2}\right)} \Rightarrow t_{jExt} = \sqrt{t_{jtotal}^{2} - t_{jADC}^{2}} = \sqrt{\left(503^{-15}\right)^{2} - \left(105^{-15}\right)^{2}} = 492\,fs$$

- **This might require a bandpass filter on the clock**

Texas Instruments

Typical SNR Diagram

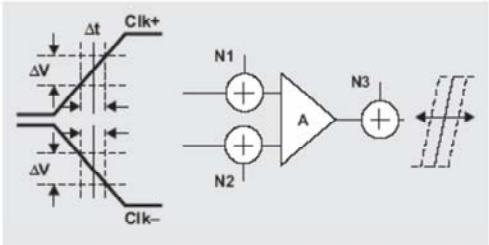


TEXAS INSTRUMENTS

Internal ADC jitter

$$\left(ADC_{Jitter\,Total}\right)^2 = \left(Jitter_{N_3}\right)^2 + \left(Jitter_{N_1,N_2}\right)^2 =$$

$$= \left(Jitter_{N_3}\right)^2 + \left(\frac{K_{(N_1,N_2)}}{Clock\ Slope}\right)^2$$



Where clock_slope for a sinsoidal signal is :

$$V = Amplitude \sin(\omega t)$$

$$Clock_Slope = \frac{dV}{dt} = Amplitude\ \omega \cos(\omega t)$$

$$Max\ Slope = Amplitude.\omega = Amplitude\ 2\pi f$$

ADC Jitter depends on the clock slope and indirectly the Clock Amplitude

So, let's model the internal ADC jitter as, jitter due to the addition of noise by the ADC clock amplifier on a finite clock slope (N1 and N2), and by the internal ADC clock distribution jitter (N3 - which is not affected by the clock amplitude). The slope of a sine wave is determined by taking a derivative of the sinewave. The maximum slope happens at the 0 crossing, so the cosine term = 1 and you are left with an amplitude dependent jitter term.

One

Calculation for ADC internal jitter (more exact)

- **What is the total internal jitter of the ADS5424 ADC when fed a 3.0Vpp 105M sinewave clock due to aperture jitter and the clock amplitude/slope?**

t_J	Clock slope independent aperture uncertainity (jitter)	150	fs
k_J	Clock slope dependent jitter factor	50	µV

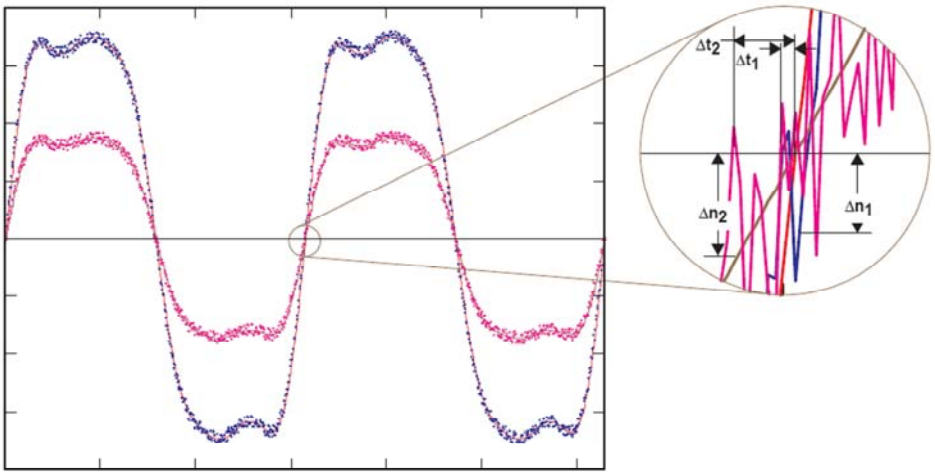
$$(ADC_{Jitter})^2 = (Jitter_{N_3})^2 + \left(\frac{K_{(N_1, N_2)}}{V_p \, 2\pi f}\right)^2 =$$

$$= \sqrt{(150 \; fs)^2 + \left(\frac{50 \; ns}{1.5 \, V \; 2\pi \; 105 \; MHz}\right)^2} = 158 \; fs$$

- …and now a 0.5Vp sinewave clock = 338 fs
- That is a big jump in jitter (from 158 fs to 338 fs) due to clock amplitude reducing from 3Vpp to 0.5Vpp
- This dependency on clock level (and therefore slope) usually shows up in the curves of SNR/SFDR vs Clock Level

Texas Instruments

Thermal Noise Effect for Different Slope Clock Signals



ADC internal jitter Spec

t_J	Clock slope independent aperture uncertainity (jitter)	150	fs
k_J	Clock slope dependent jitter factor	50	µV

- **Notice, even at 3Vpp, the calculated total jitter with a sinewave clock is 158fs, but spec says 150fs**
  - **The 150fs assumes no dependency on clock amplitude/slope**
  - **Therefore, it is usually a good idea to design the external clock jitter with the assumption that the ADC total internal jitter is worse than the spec**
  - **In practice, customers do not and will not go through the calculations using the clock slope dependent jitter factor simply because very few ADC datasheets provide this value and because the customers are not educated on how to use it**
  - **Therefore, estimates are usually made using the aperture jitter value, the simple calculation provided earlier and the plots of SNR/SFDR vs Clock Level (Amplitude)**

Texas Instruments

Where can I get the Clock Amplitude data?

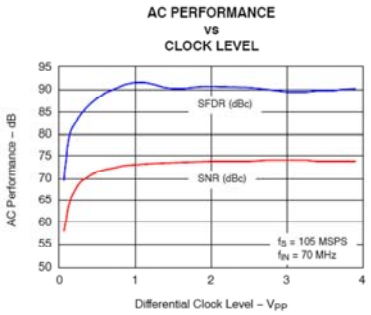


Figure 22

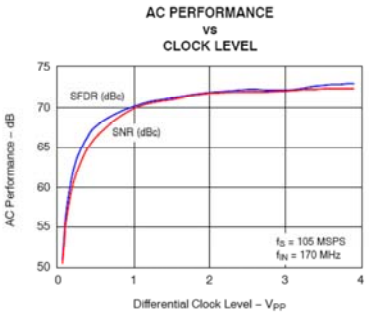
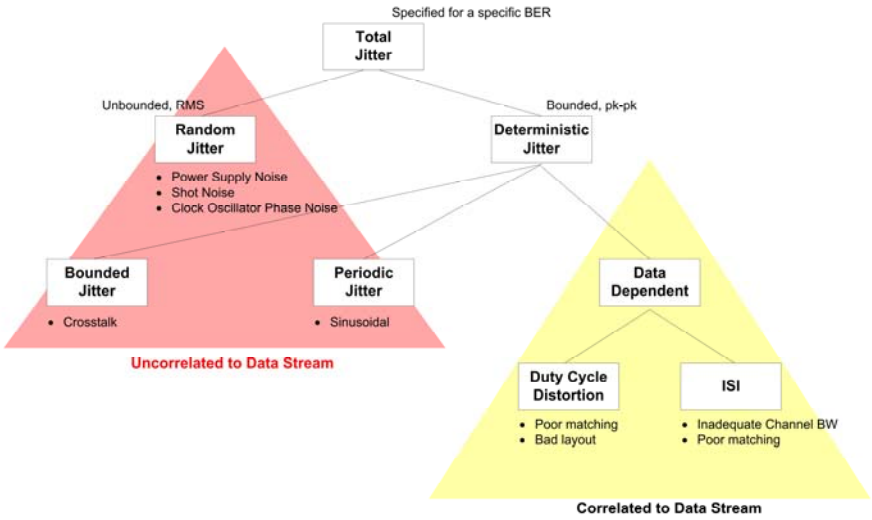


Figure 23

These are from ADS5424

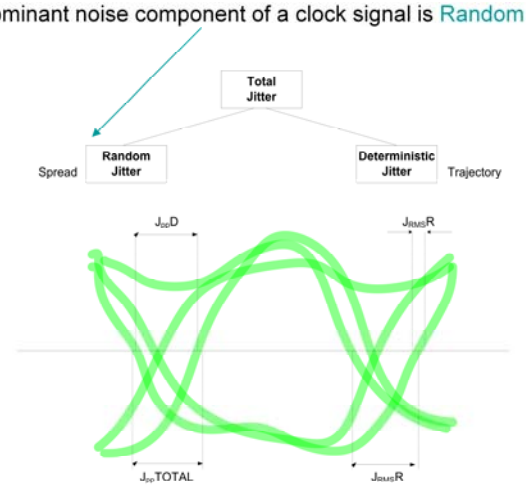
So, considering all the above, Texas Instruments has started to include on its datasheet (see ADS5424) plots that the circuit designer can use to estimate the required clock amplitude.

Jitter Tree


Specified for a specific BER
Total Jitter
Unbounded, RMS
Bounded, pk-pk
Random Jitter
Power Supply Noise
Shot Noise
Clock Oscillator Phase Noise
Deterministic Jitter
Bounded Jitter
Crosstalk
Periodic Jitter
Sinusoidal
Uncorrelated to Data Stream
Data Dependent
Duty Cycle Distortion
Poor matching
Bad layout
ISI
Inadequate Channel BW
Poor matching
Correlated to Data Stream

Elements of Total Jitter

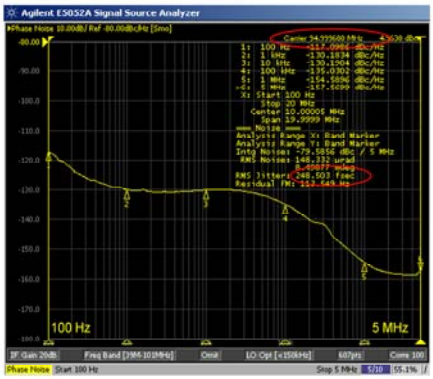
The dominant noise component of a clock signal is Random Jitter.



Random Jitter (RMS jitter) is derived from the phase noise plot of the clock signal.

Comparing Clock Performance

Can we compare these oscillators?



Texas Instruments

- When comparing RMS jitter or total phase noise between clocks, the ***limits of integration*** must be identical

  Comparing two clocks that have different phase noise bandwidths is invalid

- When comparing RMS jitter between clocks, ***the carrier frequencies*** must be identical

  Remember, jitter is proportional to the inverse of the clock frequency.

  - If two clocks have the same phase noise power but operate at different frequencies, the clock at the higher frequency will have lower jitter.

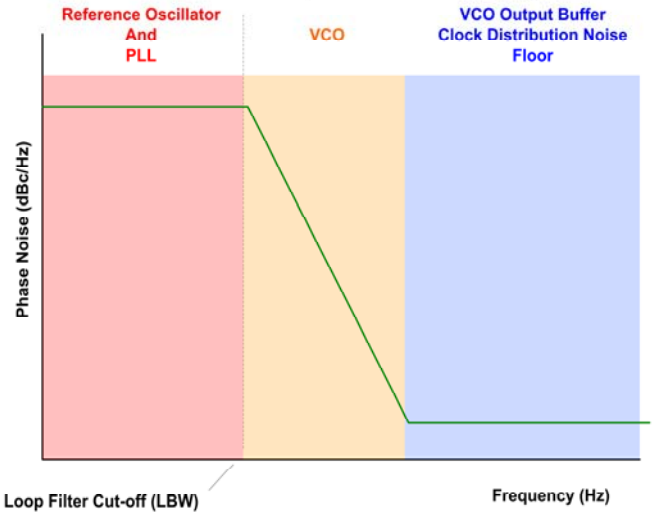
- Total (integrated) phase noise power says nothing about the distribution of the noise.
  - PSDs having different shapes may yield the same total phase noise power
- A good clock can be made to look bad if a wider region of integration is chosen relative to a lower quality oscillator

Jitter Optimization

Texas Instruments

Phase Noise Optimization

Phase Noise Components - Simplified



•The diagram above shows three primary regions in the frequency domain, in terms of offset from the carrier frequency of the clock.

•In each region, certain functions or components in the PLL will be the major noise sources

Close into the carrier, the noise of the reference oscillator dominates. Moving slightly away from the carrier, the noise of the PLL takes over. The point at which the VCO becomes a factor is determined by the bandwidth of the loop filter.

For the most part, the primary lever that a designer has to control the shape of the phase noise profile is the loop filter of the phase locked loop.

Let's look at some specific examples courtesy of EasyPLL.

Phase Noise Optimization - Levers

PLL Functional Block	To minimize Noise contribution…	Why?
Phase Detector/Charge Pump	Maximize charge pump gain (K_ϕ) (up to a certain point)	The phase detector noise contribution is proportional to $1/(K_\phi)^2$
R-counter and N-counter divide ratios	Maximize phase detector compare frequency → this minimizes N	The noise contribution of the R and N dividers is proportional to N^2.
Reference oscillator	Use highest frequency practical and use R > 1 if possible. If deciding between maximizing R and minimizing N, minimize N.	The noise contribution from the reference oscillator is proportional to $(N/R)^2$

Texas Instruments

•This table provides some guidance on design choices in PLL/VCO design that can be manipulated to improve phase noise performance.

•Keep in mind that no amount of clever design or optimization can overcome the use of poor quality components

•Using the best reference clock and VCO that meet cost targets is always a given.

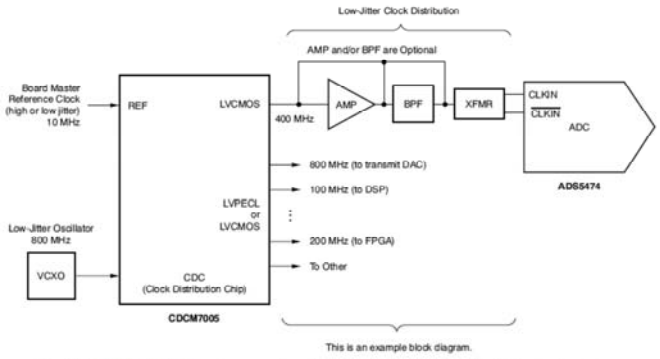
•If close-in phase noise is important for a particular application, then choose components that give the desired close-in performance and acceptable performance at higher offsets.

Selecting Loop BW

Condition	Loop Bandwidth	Why?
Noisy reference input. Good VCO.	A narrow loop bandwidth would be used to minimize the contribution of the reference input.	A narrow bandwidth allows the VCO performance to dominate. This why a high quality VCO is normally used.
Good reference. Noisy VCO	A wide loop bandwidth would be used to allow the reference to dominate inside the loop bandwidth	This will minimize the overall noise profile and total noise power because the VCO noise is suppressed inside the loop.

Solutions Example

How Do I Build the Low Jitter Clock Circuit?



Consult the CDCM7005 data sheet for proper schematic and specifications regarding allowable input and output frequency and amplitude ranges.

Figure 47. Optimum Jitter Clock Circuit

The jitter of this total clock solution at the clock input pins on the ADC will be determined primarily by

1) **The jitter of the VCO/VCXO chosen**
2) **The additive jitter in the CDC chip**
3) **The characteristics of the BPF**

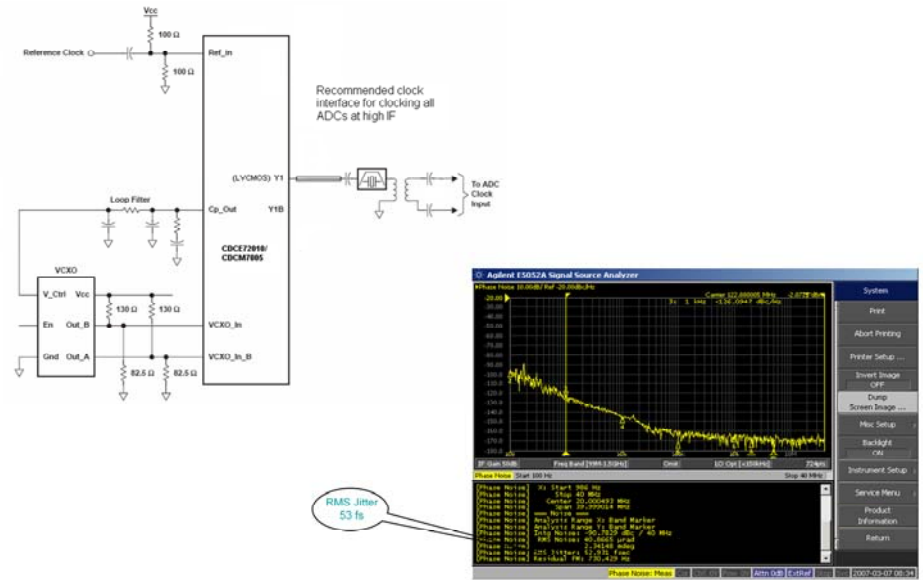
The jitter of this total clock solution at the clock input pins on the ADC will be determined primarily by

1) The jitter of the VCO/VCXO chosen
2) The additive jitter in the CDC chip
3) The characteristics of the BPF

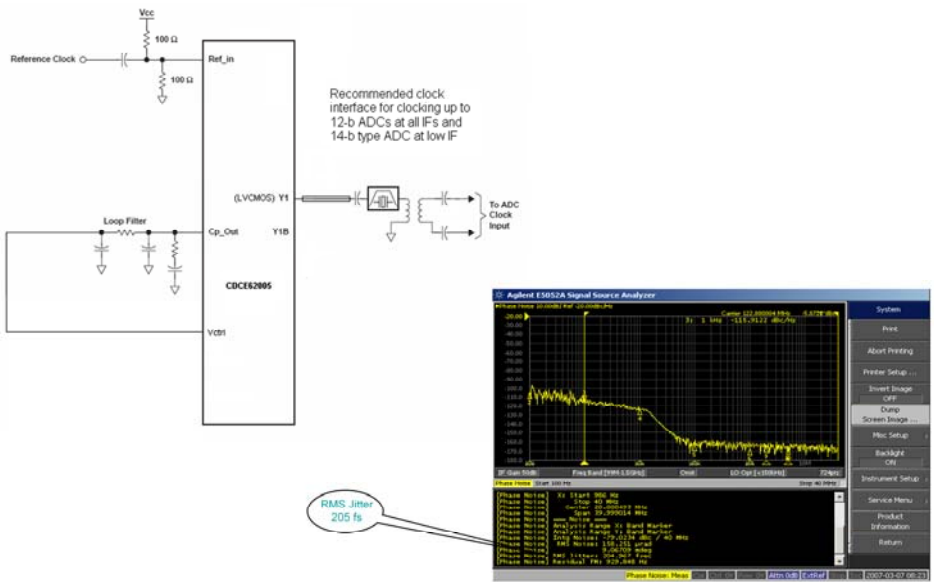
The BPF (bandpass filter) is sometimes required to reject the clock noise that is not near the clock frequency to improve the overall jitter.

The amp is optional if the insertion loss of the BPF (assuming it is required as well) causes the clock amplitude to be too low at the ADC clock pins.

CDCE72010 + BPF Clocking an ADC



CDCE62005 + BPF Clocking an ADC



Texas Instruments

CDCE72010 ADC/Macro BTS Clock
2:10 Ultra Low Jitter Cleaner w/ext. VC(X)O

Features

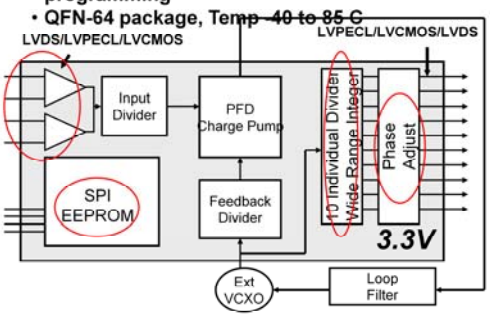
- Input frequencies up to 180MHz SE/500MHz Differential LDVS/LVPECL/LVCMOS
- Output frequencies up to 1.175GHz
- Output up to 10 LVPECL/10 LVDS/20 LVCMOS
- Optional High Swing LVPECL output
- Wide-range integer divide selectable by output
- Individual Phase Adjust
- Ultra low additive jitter (<35fs, RMS)
- Low Output Skew (~ 20ps, typ)
- Provides holdover functionality
- On-chip EEPROM

Applications

- Wireless BTS (Macro, Micro Cells)
- SONET
- Data Communications
- Test Equipment
- Jitter Cleaners

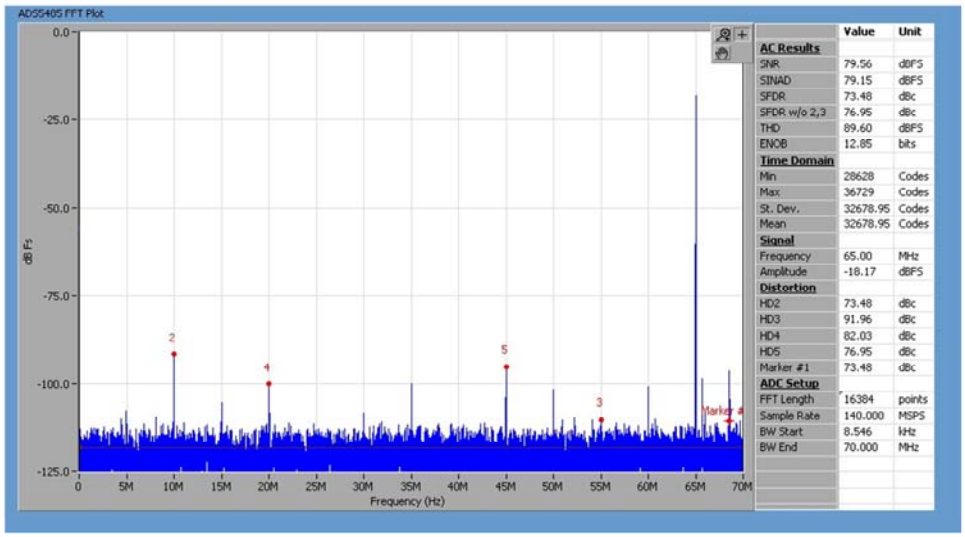
Benefits

- Wide input/output frequency range supports high and low end of frequency standards
- Selectable input/output standards reduces translation logic
- External loop filter provides maximum flexibility
- EEPROM saves default start-up settings
- SPI interface provides in-system programming
- QFN-64 package, Temp -40 to 85 C



Texas Instruments

Clean Jitter



	Value	Unit
AC Results		
SNR	79.56	dBFS
SINAD	79.15	dBFS
SFDR	73.48	dBc
SFDR w/o 2,3	76.95	dBc
THD	89.60	dBFS
ENOB	12.85	bits
Time Domain		
Min	28628	Codes
Max	36729	Codes
St. Dev.	32678.95	Codes
Mean	32678.95	Codes
Signal		
Frequency	65.00	MHz
Amplitude	-18.17	dBFS
Distortion		
HD2	73.48	dBc
HD3	91.96	dBc
HD4	82.03	dBc
HD5	76.95	dBc
Marker #1	73.48	dBc
ADC Setup		
FFT Length	16384	points
Sample Rate	140.000	MSPS
BW Start	8.546	kHz
BW End	70.000	MHz

Texas Instruments

Thank you!

Texas Instruments