Application Note

Leveraging TI Clock Tools For Data Converter Designs

Andrea Vallenilla

ABSTRACT

The performance of the sample clock is one of the most critical factors in maximizing the performance of a high-speed converter. This application note details a step-by-step process on how to use TI's clocking tools to select the best clock based on the requirements determined for the application.

Table of Contents

1 Introduction....2
2 Effects of a Phase Noise Curve of a Clock on the Performance of the Converter....2
3 Determine Clock Performance Target Values for a Selected Data Converter....3
4 Narrow Down Which TI Clock to Select Based on the Specified Requirements Determined....3
5 Analyze the Proposed Clock's Jitter on a Converter's Performance....6
6 Understanding Clocking Performance Effects on the SNR of the Converter....8
6.1 Limitations from TI Clocking Parts When Paired with TI High-Speed Converters....13
7 Summary....15
8 References....15
9 Appendix A: Clock Tree Architect (CTA) Detailed Step-By-Step Guide....16
10 Appendix B: Using PLLatinum Sim to Represent Clock Jitter for an Application's Integrated Bandwidth....22
11 Appendix C: PLLatinum Sim Phase Noise Curves Comparison Step-by-Step Guide....24
12 Appendix D: Comparing Phase Noise Curve Simulations of PLLatinum Sim with Measured Data....26

Trademarks

1 Introduction

TI offers several tools to select a clock based on the specified requirements determined and the selected converter. The rest of this application note is based on the following:

1. The ADC12DJ5200RF is used and configured to JMODE 3 (dual-channel) and sampled at 5200MSPS.
2. The analog input frequency is f_{in} = 900MHz.
3. The goal is to maximize the dynamic performance of the converter, or SNR (signal-to-noise ratio).

This high-level step-by-step example guide helps to select the best clock based on the above assumptions:

1. Determine Clock Performance Target Values for Your Selected Data Converter
   a. Halve the aperture jitter of the converter. This is equivalent to lowering the phase noise curve by 6dBc/Hz, which sets the jitter target of the clock.
   b. Subtract 6dBc/Hz from the noise floor of the converte at the sample frequency. This sets the noise floor target of the clock.
2. Narrow Down which TI Clock to Select Based on an Application's Requirements
   a. Clock Tree Architect (CTA) considers not only your performance target values from step 1, but also the overall power consumption, cost, area, and other clock features.
   b. Section 9 provides detailed instructions on how to use CTA.

Note

The jitter bandwidth used in CTA is limited to 12kHz to 20MHz and does not consider the jitter outside of this range. Jitter or noise outside this range can be considered white or Gaussian.

3. Analyze the Proposed Clock's Jitter on a Converter's Performance
   a. Enter the jitter value of the proposed clock given by CTA in the Excel SNR2Jitter_Curve comparison tool and determine if the performance achieved with the proposed clock is enough to meet the specified requirements.

2 Effects of a Phase Noise Curve of a Clock on the Performance of the Converter

A phase noise curve is a frequency-domain representation of the power spectral density of a signal's short-term phase fluctuations, showing how much energy has leaked from the ideal carrier frequency into sidebands. The more positive the phase noise curve is, the more energy has leaked from the ideal carrier frequency into the sidebands. Selecting a clock that has 6dBc/Hz lower phase noise than the phase noise of the converter is sufficient to maximize the performance of the converter. However, some applications can tolerate worse clock performance, depending on requirements. *Practical Clocking Considerations That Give Your Next High-Speed Converter Design an Edge* explains and uses practical examples to show how phase noise, jitter, and other clock characteristics affect a converter's performance. Before diagnosing clock degradation, consult this app note — issues may arise from slew rate violations, inadequate filtering, or incompatible output interfaces.

3 Determine Clock Performance Target Values for a Selected Data Converter

As mentioned previously, to sufficiently maximize the performance of the converter, the phase noise of the clock must be at least 6dBc/Hz better than the phase noise of the converter. This is equivalent to halving the aperture jitter of the converter. When using the ADC12DJ5200RF as the reference converter, the jitter of the clock must be at least 25fs of jitter (the aperture jitter of the ADC12DJ5200RF is 50fs). To determine the target noise floor for the clock, subtract 6dBc/Hz from the noise floor of the converter. The noise floor of the ADC12DJ5200RFis -151.8dBFS/Hz and therefore, the target clock noise floor is -157.8dBc/Hz.

To learn more about the relationship between phase noise and jitter, see Jitter and phase noise definition and Jitter and Phase Noise Measurement Techniques for BAW Oscillators.

4 Narrow Down Which TI Clock to Select Based on the Specified Requirements Determined

TI's Clock Tree Architect (CTA) is a synthesis tool that proposes clock trees based on system requirements. The tool searches through an extensive database of TI top clocking products to generate system-level clock trees.

Note

CTA uses an integration bandwidth from 12kHz to 20MHz and therefore, does not consider the proposed clock's phase noise below 12kHz offsets. If a user needs to consider the close-in phase noise of the clock, see Section 10.

Fill out all inputs required by CTA and be as specific as possible. Use the jitter and noise floor calculated from Section 3 for the jitter and noise floor boxes on the *Outputs* section (these boxes are optional, click on the *+ Add requirements* button to expose them). The jitter represents the integrated noise power from 12kHz to 20MHz and the noise floor represents the phase noise beyond 20MHz offset.

Apart from these two inputs, CTA allows the user to enter other characteristics about application and clocking needs. As seen on Figure 4-1, the user can input specific output requirements, input characteristics, and overall system requirements. All of this information allows CTA to give a more holistic clocking suggestion. For a more detailed CTA step-by-step how-to guide, see Section 9.

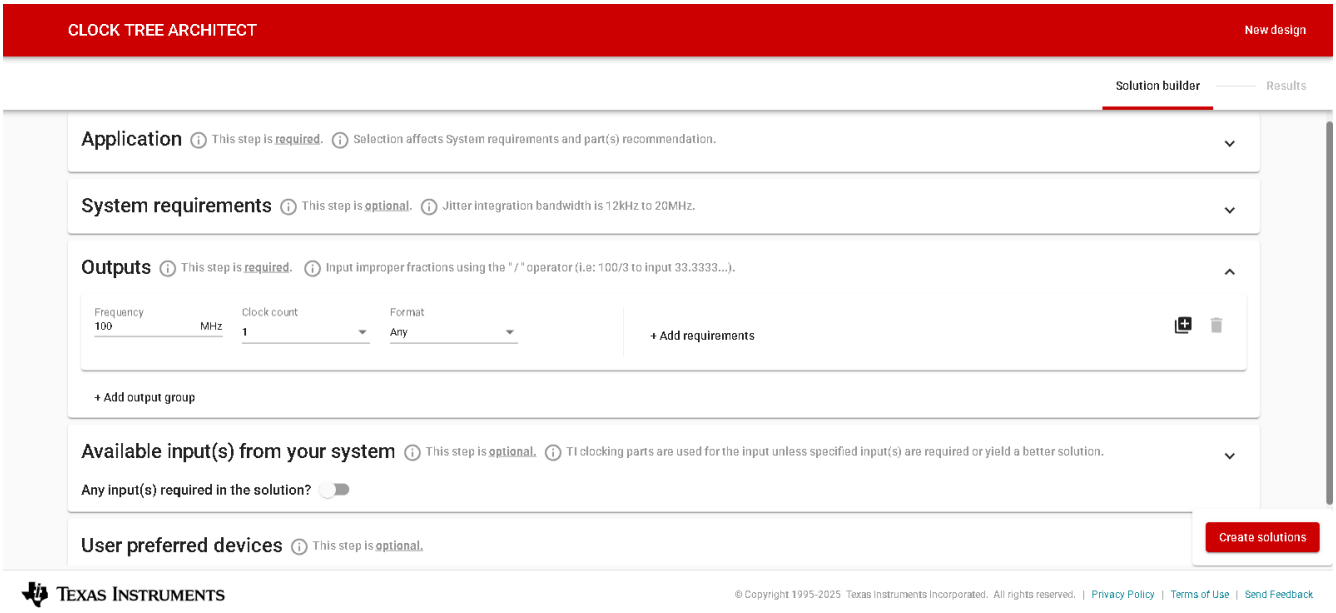


Figure 4-1. Clock Tree Architect Tool

After entering the required jitter, noise floor, and output frequency into CTA, the tool outputs either a list with valid clocking suggestions or a pop-up message stating that no design was found. In the latter case, CTA still provides a list of clocks that nearly met all requirements, offering alternative designs.

CTA can incorrectly reject suitable clocks for your application. The analog input frequency needs to be considered when the jitter of the clock is comparable to the jitter performance of the converter. For the ADC12DJ5200RF, CTA does not take the analog input frequency into account. Figure 4-2 demonstrates the inputs given to CTA and Figure 4-3 demonstrates the results from CTA when maximizing the performance of the ADC12DJ5200RF. Based on the results from CTA, there is no design that can achieve the specified application requirements. However, that is not true when considering an analog input frequency of 900MHz. Follow the steps in Section 5 to analyze how to select a clock when needing to consider the analog input frequency. For a detailed explanation of how the analog input frequency impacts the overall performance, see *Practical Clocking Considerations That Give Your Next High-Speed Converter Design an Edge*.

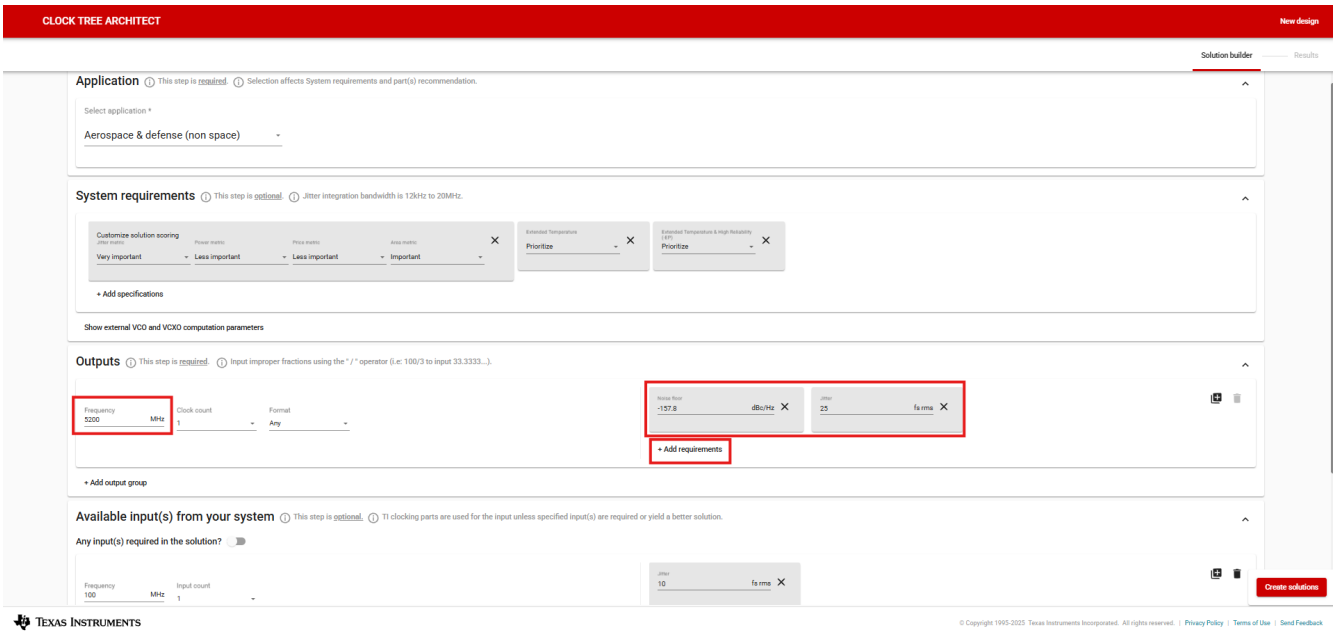


Figure 4-2. Clock Tree Architect Inputs to Obtain Clock Suggestions that Maximize the Performance of the ADC12DJ5200RF at 5200MSPS

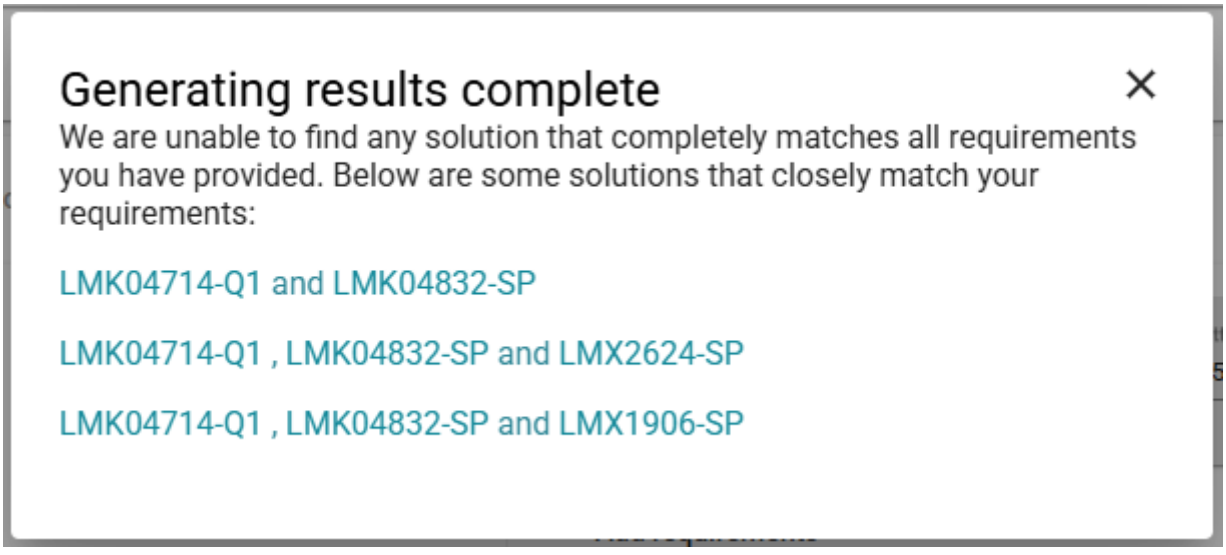


Figure 4-3. Clock Tree Architect Solutions When Maximizing the Performance of the ADC12DJ5200RF

Assume that the clock performance target values are relaxed. The clock jitter needs to be < 55fs and the noise floor < -155dBc/Hz. The list of clocking suggestions is shown in Figure 4-4

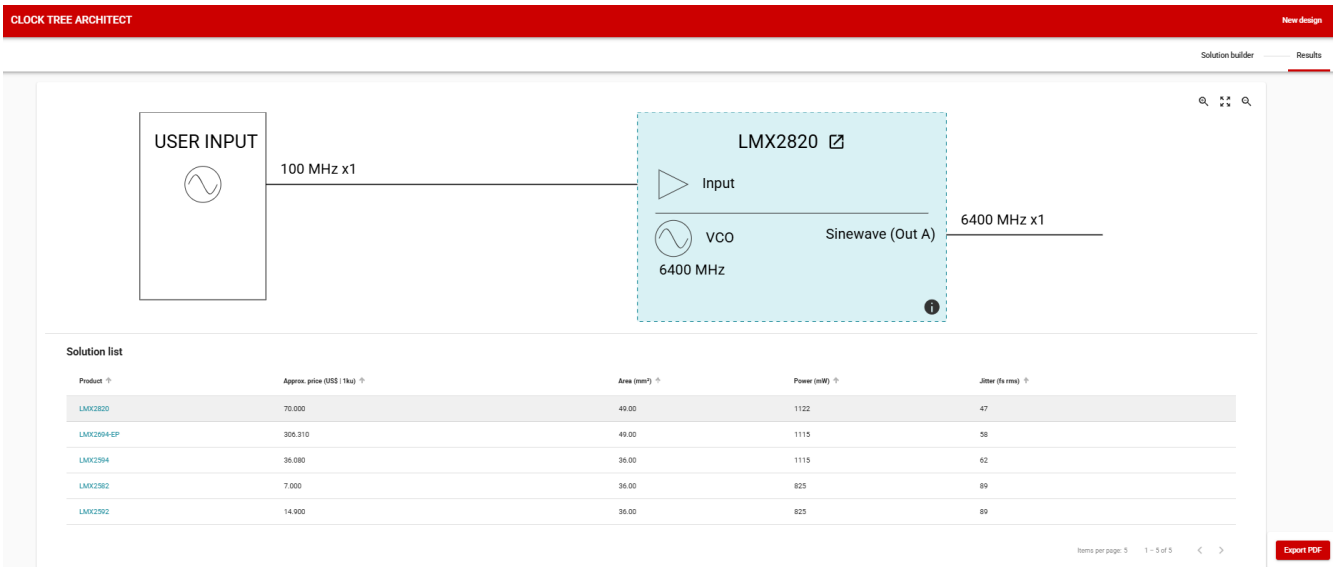


Figure 4-4. Clock Suggestions When Inputting Less Stringent Requirements

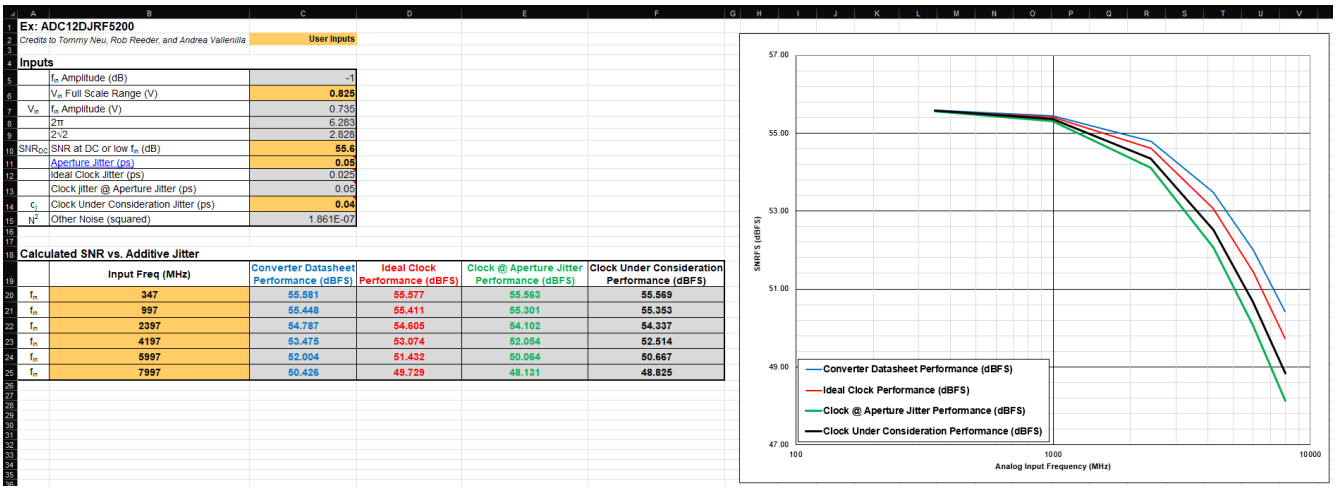
5 Analyze the Proposed Clock's Jitter on a Converter's Performance

Equation 1 demonstrates the effect of the SNR of the converter ($SNR_{converter}$) against the additive jitter of the clock (c_j) and analog input frequency (f_{in}). V_{in} represents the analog input full scale (FS) amplitude and N represents noise from other sources.

$$SNR_{converter} = 20 \times log_{10}\left[\frac{Vin}{2\pi \times fin \times \sqrt{\left[\left(\frac{Vin \times 2\pi \times fin \times 10^{-6} \times Cj}{2\sqrt{2}}\right)^2 + N^2\right]}}\right] \quad (1)$$

$$N = \frac{Vin}{10^{\frac{SNR_{DC}}{20}} \times 2\sqrt{2}} \quad (2)$$

The SNR2Jitter_Curves excel tool (downloadable here), shown on Figure 5-1, simulates Equation 1. This Excel tool uses the ADC12DJ5200RF as an example. Follow these steps to use this tool to determine the maximum clock jitter allowed to achieve desired performance for the specified analog input frequency.



Ex: ADC12DJRF5200

Credits to Tommy Neu, Rob Reeder, and Andrea Vallenilla — User Inputs

Inputs

	f_{in} Amplitude (dB)	-1
	V_{in} Full Scale Range (V)	0.825
V_{in}	f_{in} Amplitude (V)	0.735
	2π	6.283
	2√2	2.828
SNR_{DC}	SNR at DC or low f_{in} (dB)	55.6
	Aperture Jitter (ps)	0.05
	Ideal Clock Jitter (ps)	0.025
	Clock jitter @ Aperture Jitter (ps)	0.05
c_j	Clock Under Consideration Jitter (ps)	0.04
N^2	Other Noise (squared)	1.861E-07

Calculated SNR vs. Additive Jitter

	Input Freq (MHz)	Converter Datasheet Performance (dBFS)	Ideal Clock Performance (dBFS)	Clock @ Aperture Jitter Performance (dBFS)	Clock Under Consideration Performance (dBFS)
f_{in}	347	55.581	55.577	55.563	55.569
f_{in}	997	55.448	55.411	55.301	55.353
f_{in}	2397	54.787	54.605	54.102	54.337
f_{in}	4197	53.475	53.074	52.054	52.514
f_{in}	5997	52.004	51.432	50.064	50.667
f_{in}	7997	50.426	49.729	48.131	48.825

Figure 5-1. SNR2JItter_Curves Excel Tool

1. Find the full scale range specification of the analog input in the datasheet of the converter and enter it in cell C6. For the ADC12DJ5200RF, the V_{in} = 0.825V.
   a. In most cases, V_{in} is driven to 1dB below full scale (-1dBFS) to avoid clipping the ADC's analog input. If the application does not use 1dB below full scale, unlock the sheet and specify how far the analog input is driven from full scale in cell C5. The resulting analog input amplitude is calculated in C7.
2. Find the SNR of the converter at DC (SNR_{DC}) when driven to -1dBFS in the datasheet (or the SNR at the lowest f_{in} if SNR_{DC} is not provided). Enter this value in cell C10. For the ADC12DJ5200RF, the SNR_{DC} = 55.6dBFS (with a f_{in} = 347MHz).
   a. After inputting this value, other noise contributions are calculated in cell C13. This equation is represented in Equation 2.
3. Find the aperture jitter of the converter. Enter the value in cell C11.
4. Find the SNR of the converter in the datasheet. On cells B18 through B23, enter the analog input frequencies used for the SNR values. For the ADC12DJ5200RF, these frequencies are 347MHz, 997MHz, 2397MHz, 4197MHz, 5997MHz, and 7997MHz. The tool simulates the SNR across analog input frequencies of the converter.
5. Compare the SNR values in cells C18 through C23 with the ones in the datasheet of the converter. If the values are about 0.3dBFS from the values in the datasheet, then this is a good representation of the converter's performance. If not, estimate with the aperture jitter value until each SNR values at each f_{in} is close to the data sheet value.
6. On cell C14, enter the jitter value of the clock (c_j) shown on first clock suggestion from CTA.

   a. Remember that CTA only considers jitter from 12kHz to 20MHz. To obtain the jitter across a wider bandwidth, see Section 10. When using the LMX2820 as an example, the jitter from 100Hz to 100MHz at 5200MHz is 40fs. This is the clock used in the tool.

7. Use the graph on the SNR2Jitter_Curves tool to estimate the SNR achieved with the suggested clock at the desired f_{in}. For this example, f_{in} = 900MHz and the LMX2820 was used. The graph shows that at f_{in} = 900MHz, the LMX2820 does maximize the performance of the ADC12DJ5200RF (datasheet performance is 55.448dBFS vs. with the LMX2820, the performance of the ADC12DJ5200RF is 55.353dBFS). At lower analog input frequencies, the phase noise of the converter dominates the overall SNR performance instead of the clock; therefore, the LMX2820 does not deteriorate the performance of the converter at this f_{in}. To see measurements of the LMX2820 clocking the ADC12DJ5200RF, see Section 6.

Generally, obtaining 1dBFS lower than the SNR_{ADC} is still acceptable and is not a significant deterioration to the performance of the converter. Around 3dBFS is when the converter can start being affected. This tool can be used to pinpoint the acceptable clock jitter to meet the specified application requirements by changing the value in cell C14. To determine a clock with that jitter, work backwards and enter that as an input to CTA.

Another option to determine the impact of the clock jitter on the performance of the converter is to overlay and compare both the clock's phase noise curve and the converter's phase noise curve. If you have the converter's phase noise curve, use TI's PLLatinum Sim tool to simulates the phase noise curve of TI's clocking parts and upload the converter's phase noise curve. This allows the user to see a side-by-side comparison and determine if the phase noise of the clock is 6dBc/Hz lower than the phase noise of the converter across the desired bandwidth. If optimal performance is not required, being approximately 4dBc/Hz lower does not cause too much deterioration. Section 11 contains a step-by-step how-to guide on uploading an external phase noise curve to PLLatinum Sim and Section 12 demonstrates PLLatinum Sim's accuracy of TI clock's phase noise curves simulations.

6 Understanding Clocking Performance Effects on the SNR of the Converter

High-performance clocking designs often come with trade-offs in area, cost, and power or can require discrete implementation. Determine whether maximizing SNR or minimizing size, power, and cost is the higher priority for a given application.

This section demonstrates the effect of clocking the ADC12DJ5200RF with different performing clocks. Figure 6-1 shows the three different phase noise curves of three different performing clocks: the LMX2820, the LMX2594, and the LMX2572 at 5200MHz.

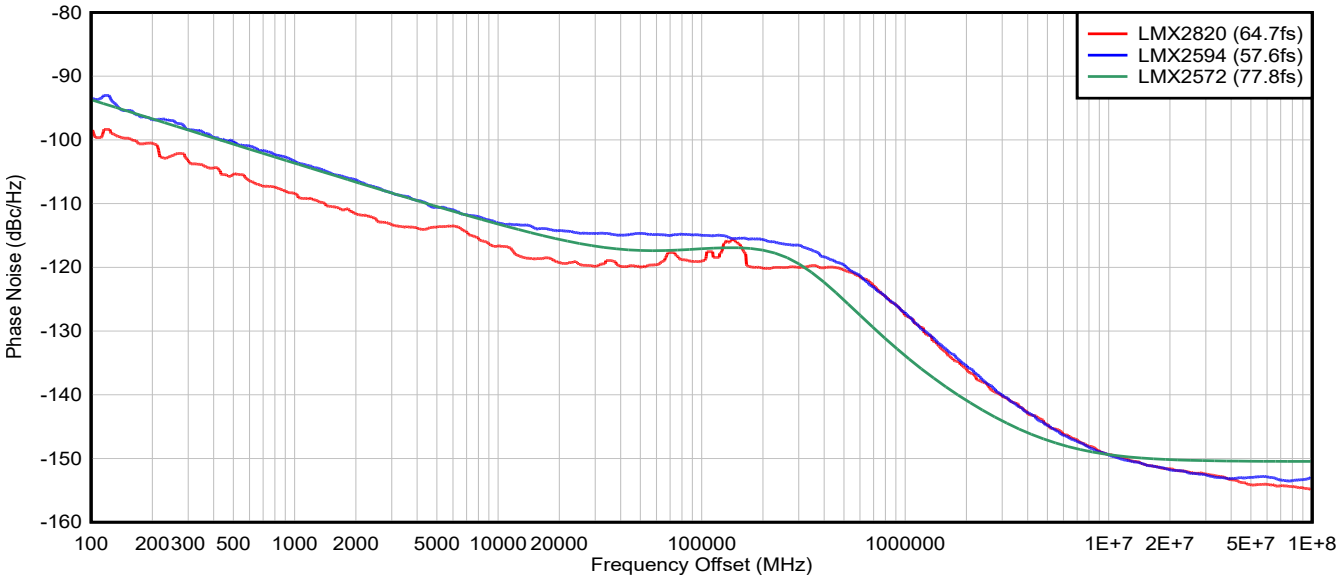


Figure 6-1. Overlay of the LMX2820, LMX2594, and LMX2572 Phase Noise Curves at 5200MHz

The ADC12DJ5200RF was configured with JMODE 3 at a 5200 MSPS sample rate and a f_{in} = 900MHz driven to full scale for all the following measurements. Figure 6-2 shows the datasheet performance for the ADC12DJ5200RF under these conditions by clocking the converter with a filtered Rhode & Schwartz SMA100B 5200MHz signal generator.

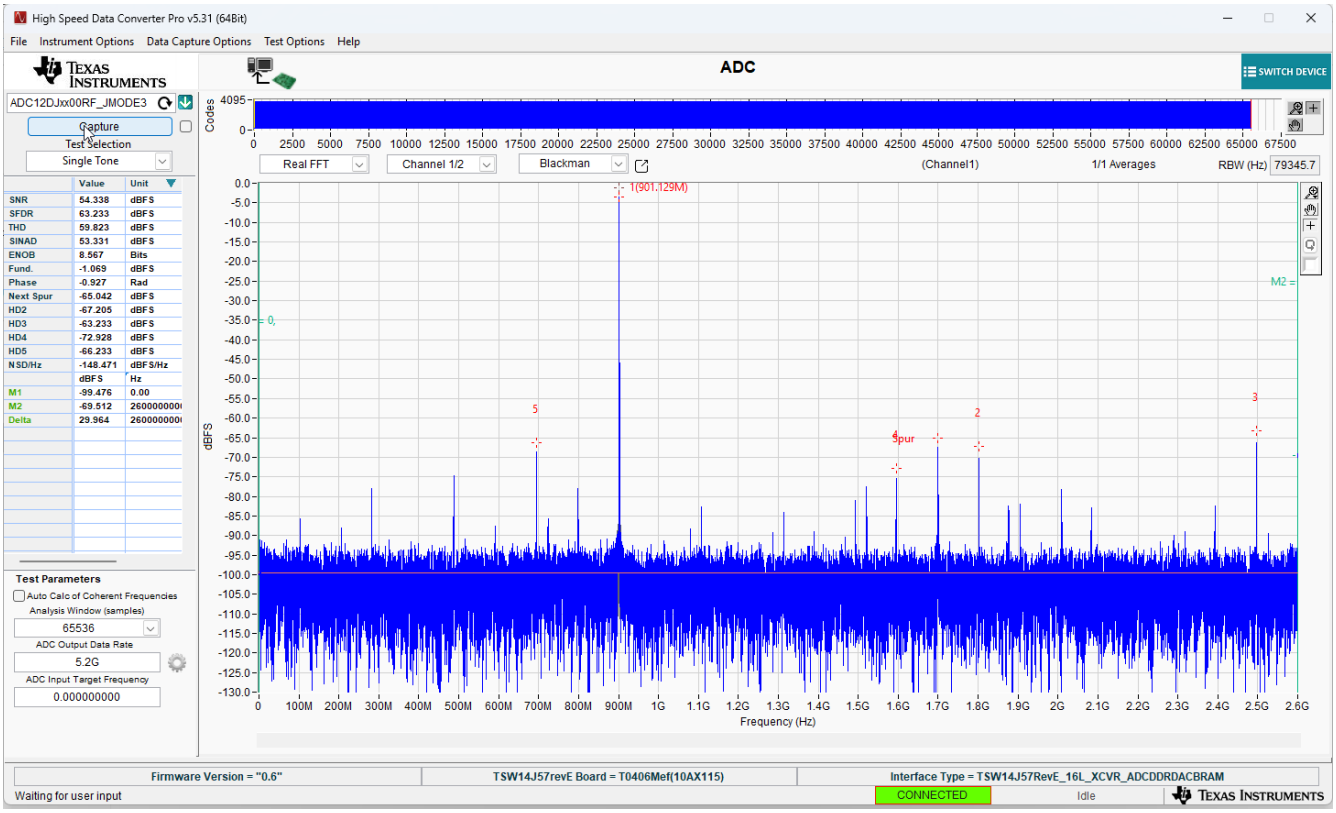


Figure 6-2. ADC12DJ5200RF Baseline FFT Plot with 5200MSPS, f_{in} = 900MHz, JMODE 3 Using the Rhode & Schwarz SMA100B with a Bandpass Filter as the Reference Clock

Figure 6-3 shows the performance of the ADC12DJ5200RF with the LMX2820. The resulting SNR is 54.161dBFS which matches the performance of the ADC12DJ5200RF datasheet, meaning the LMX2820 does not deteriorate the ADC's SNR at f_{in} = 900MHz.

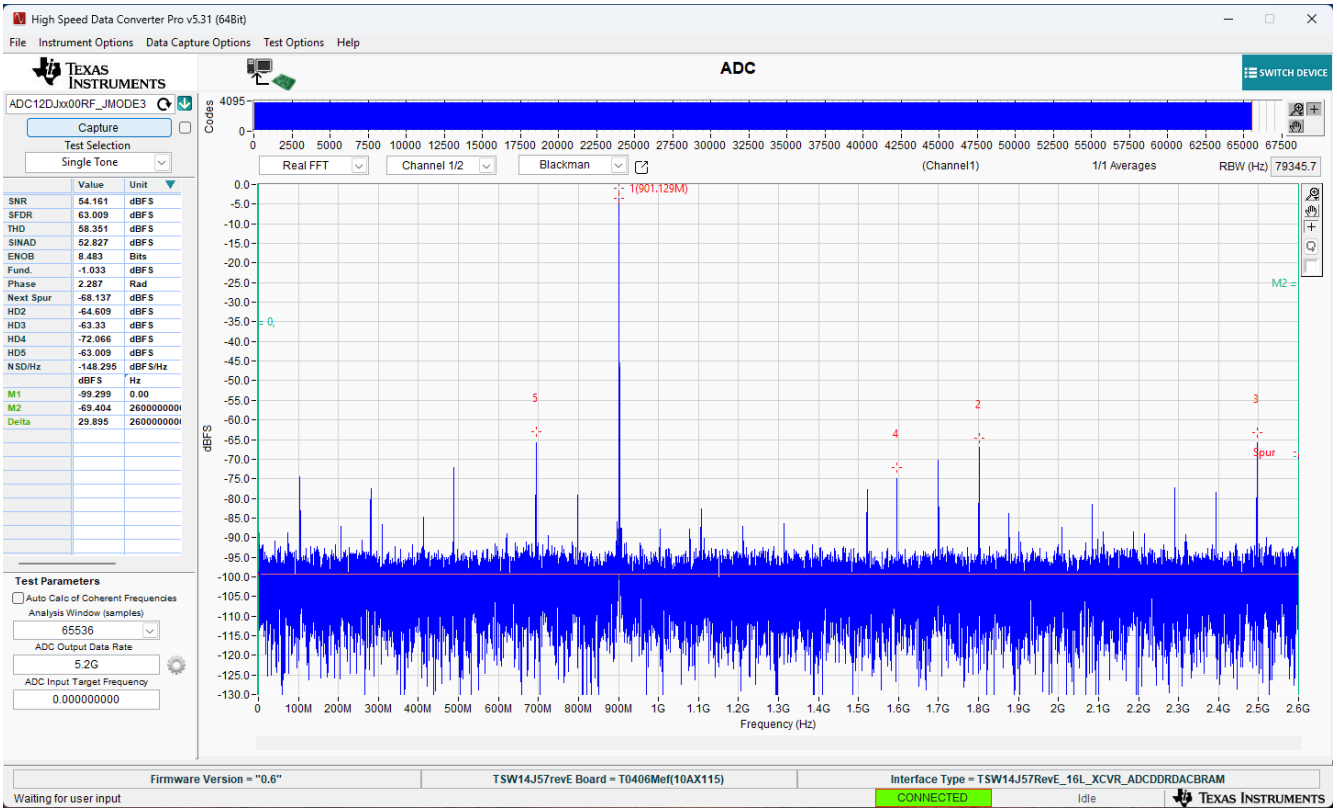


Figure 6-3. FFT of the LMX2820 and ADC12DJ5200RF at 5200MSPS and f_{in} = 900MHz and JMODE 3

On Figure 6-4, the ADC12DJ5200RF is paired with the LMX2594, an older and lower performing generation of the LMX2820, but cheaper, lower power, and smaller. The ADC12DJ5200RF still achieves 54.073dBFS, showcasing that even though the LMX2594 is 15.3fs higher than the LMX2820, the ADC12DJ5200RF's performance is still being maximized at f_{in} = 900MHz.

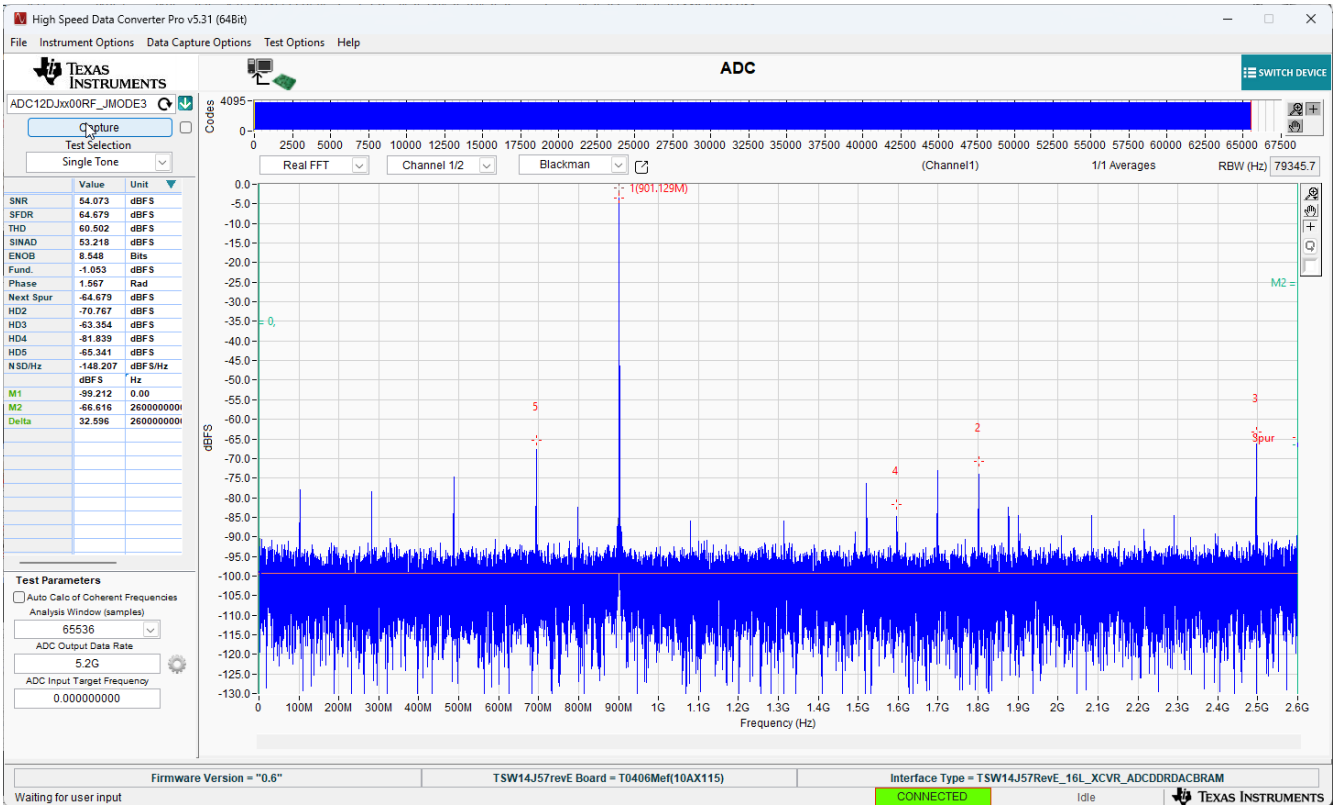


Figure 6-4. FFT of the LMX2594 and ADC12DJ5200RF at 5200MSPS and f_{in} = 900MHz and JMODE 3

Figure 6-5 demonstrates the effects of using the ADC12DJ5200RF with the LMX2572, a lower performing clock than the LMX2594, but cheaper and lower power. The jitter degradation between the LMX2820 and LMX2572 is 35.5fs, yet the SNR degradation is only 0.529dBFS lower compared to the baseline (the LMX2572 and ADC12DJ5200RF pairing result in 53.809dBFS). This performance difference is still not significant and does not degrade the performance of the ADC12DJ5200RF.

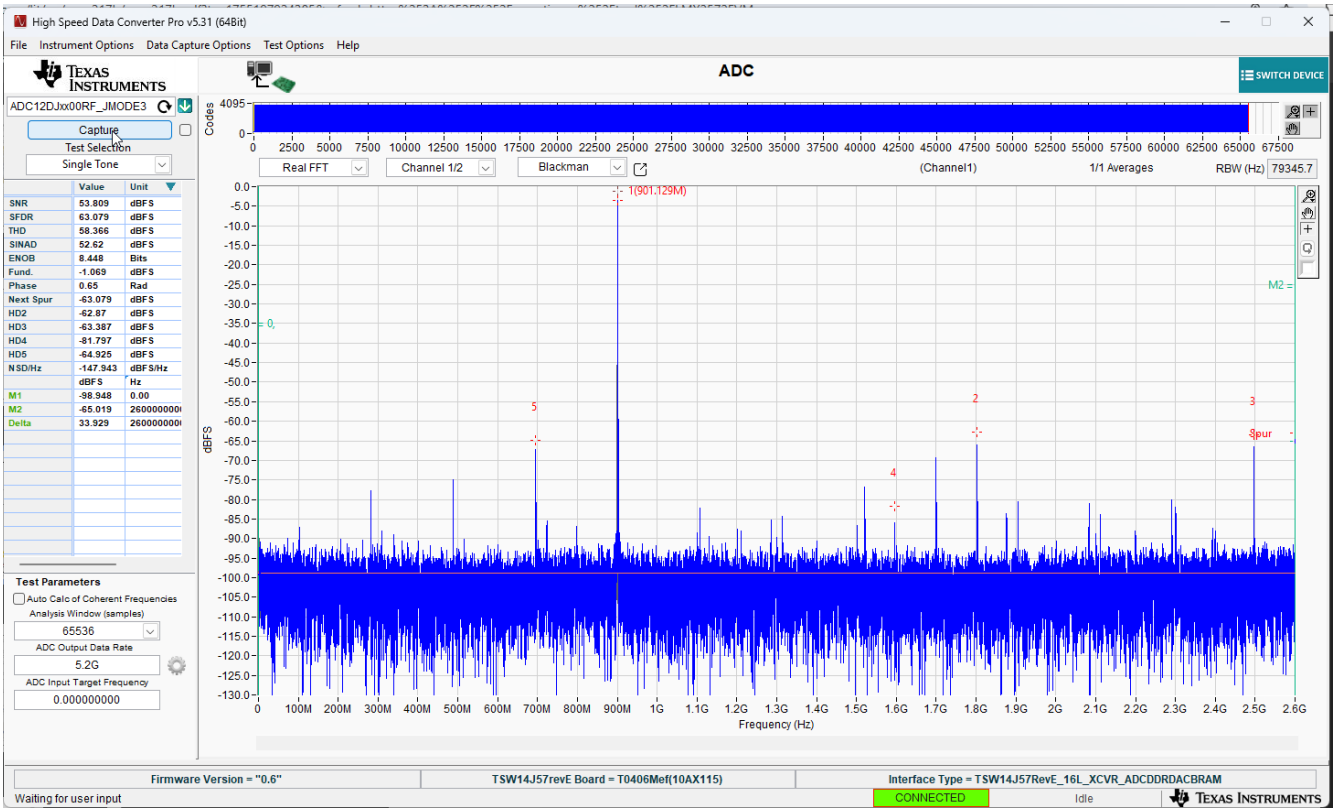


Figure 6-5. FFT of the LMX2572 and ADC12DJ5200RF at 5200MSPS and f_{in} = 900MHz and JMODE 3

For this example, the performance degradation seen when using the three different clocks was not significant and the LMX2572 is the clear winner for this application. This demonstrates that maximizing the performance of the converter does not always have as much of an impact as compared to the clock's power, cost, and size. Table 6-1 compares these three clocks.

Table 6-1. Comparison Between TI's LMX2820, LMX2594, and LMX2572

Specification	LMX2820	LMX2594	LMX2572
Jitter @ 5200MHz (BW: 100Hz to 100MHz)	42.3fs	57.6fs	77.8fs
Noise Floor at 5200MHz	−155.0dBc/Hz	−152.8dBc/Hz	−150.5dBc/Hz
SNRFS (f_{in} = 900MHz, 1dBFS)	54.1dBFS	54.07dBFS	53.8dBFS
Power	500mA	340mA	75mA
Size	49mm^2 (7mm x 7mm)	36mm^2 (6mm x 6mm)	36mm^2 (6mm x 6mm)
1ku Pricing	$84.000	$46.970	$19.404
Frequency Range	25MHz to 22.6GHz	10MHz to 15GHz	28MHz to 6.4GHz

All these examples show how different clocks impact the performance of the converter. But remember, the performance of a clock also depends on the configuration.

6.1 Limitations from TI Clocking Parts When Paired with TI High-Speed Converters

Some TI high-speed converters outperform all of TI's current clocking portfolio. For example, the ADC32RF55 is a dual-channel, 14-bit, 3-GSPS RF-sampling ADC with very low noise spectral density (NSD), 65.5dBFS SNR, and 75dBFS SFDR. Figure 6-6 demonstrates ADC32RF55 datasheet performance. The ADC32RF55 was configured with 4x averaging at a 2600MSPS sample rate with a filtered clock and a f_{in} = 500MHz driven to full scale for all figures in this section.

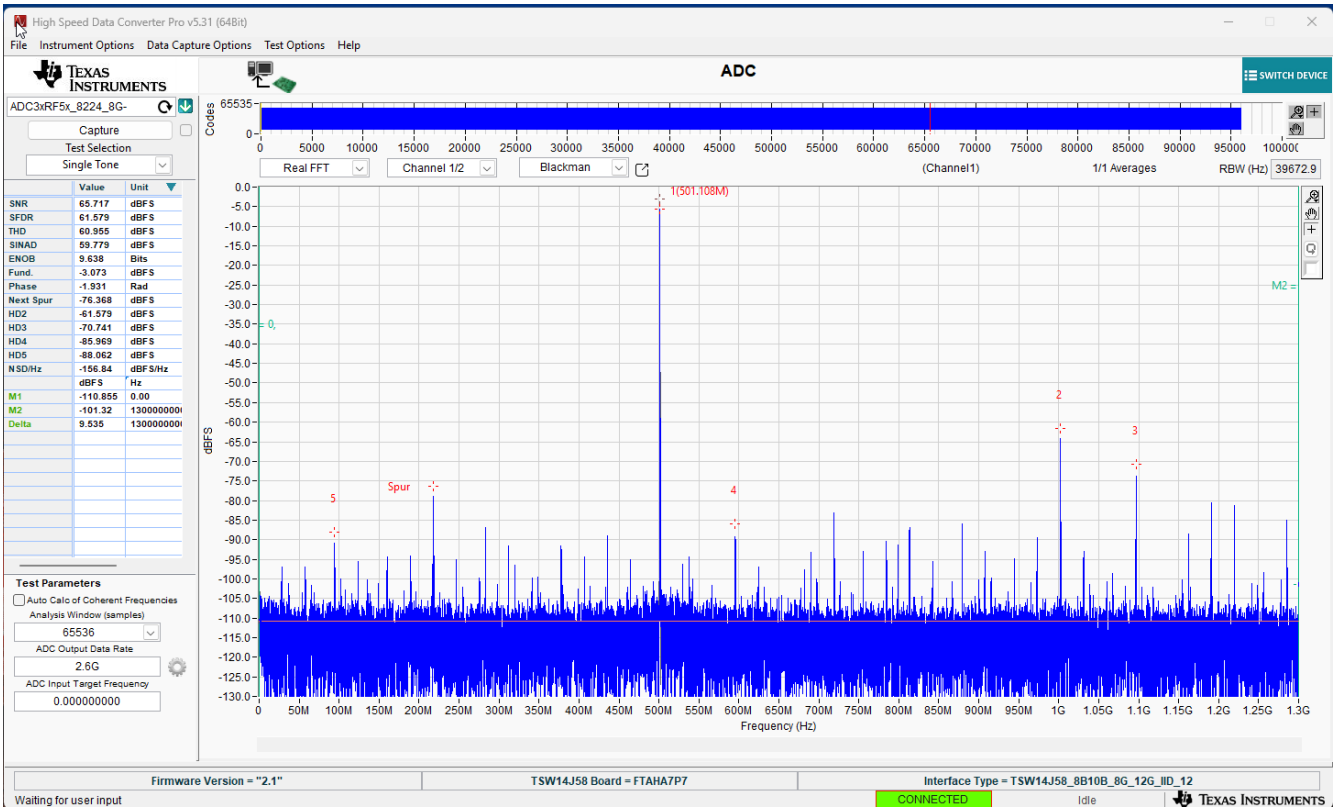


Figure 6-6. ADC32RF55 Datasheet Performance FFT Sampled at 2600MSPS with a Filtered Rhode & Schwarz SMA100B Signal Generator and f_{in} = 500MHz

The ADC32RF55 was driven with the same three clocks used previously with the ADC12DJ5200RF: the LMX2820 (TI's highest performing clock), the LMX2594, and the LMX2572. However, because the ADC32RF55 is better performing than the ADC12DJ5200RF, none of the clocks can mazimize the perfomance of the ADC32RF55, as demonstrated in Figure 6-7, Figure 6-8, and Figure 6-8. To maximize the performance of the ADC32RF55, a discrete, passive clock is needed but this approach adds area and cost to the design.

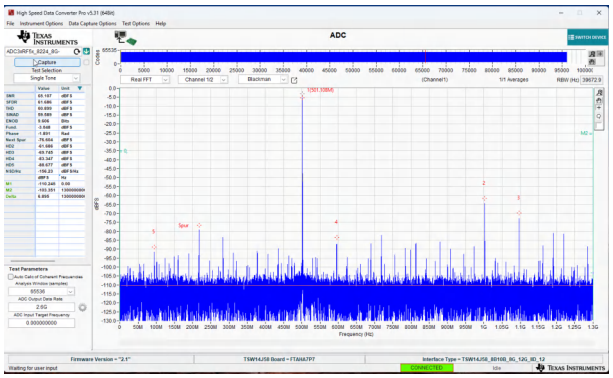


Figure 6-7. ADC32RF55 Performance FFT Sampled at 2600MSPS with a Filtered LMX2820 and f_{in} = 500MHz

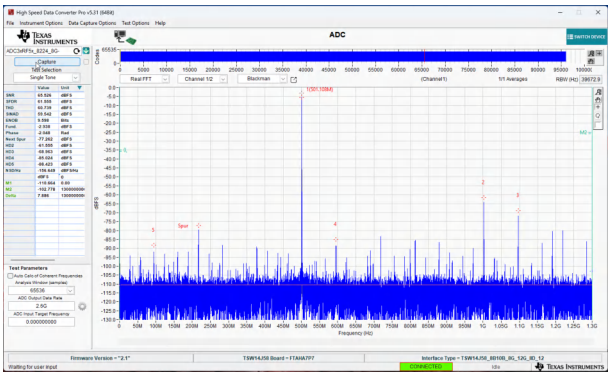


Figure 6-8. ADC32RF55 Performance FFT Sampled at 2600MSPS with a Filtered LMX2594 and f_{in} = 500MHz

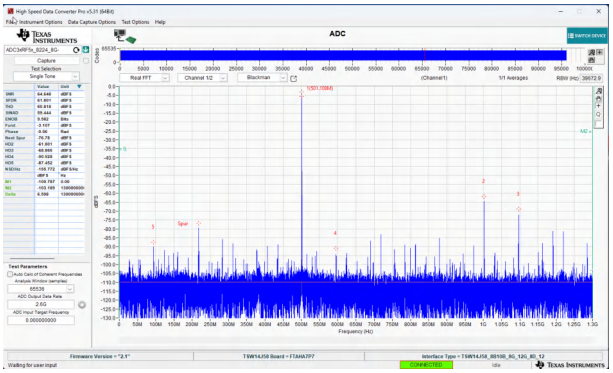


Figure 6-9. ADC32RF55 Performance FFT Sampled at 2600MSPS with a Filtered LMX2572 and f_{in} = 500MHz

If better performance is required than given by the LMX2820 (shown on Figure 6-7), consider using an external DRO. A DRO generally has better performance than the integrated VCO of the LMX2820. If that still not good enough, go to Section 5 and determine how much maximum jitter is allowed by the clock to achieve desired performance.

7 Summary

A clock with poor phase noise reduces the performance of the converter. To avoid performance degradation, target a clock that is 6dBc/Hz better in phase noise than the phase noise of the converter. Use Clock Tree Architect to see a list of clocking options that meet system requirements and use the SNR2Jitter_Curves Excel tool provided to see the effects of different performing clocks on the SNR of the converter. Using better performing clocks does not always outweigh other considerations such as cost, area, and jitter. Depending on the converter used, the application requirements, and the analog input frequency, a balance between jitter performance, cost, and area is needed to determine the best choice for the given application. Use Clock Tree Architect, SNR2Jitter_Curves Excel tool, and PLLatinum Sim to help balance performance, cost, area, and power to select the best TI clock for the given high-speed converter application.

8 References

1. Texas Instruments, *Practical Clocking Considerations That Give Your Next High-Speed Converter Design an Edge*, application note.
2. Texas Instruments, *TI Precision Labs – Clocks and Timing*, video.
3. Texas Instruments, *Jitter and Phase Noise Measurement Techniques for BAW Oscillators*, application note.
4. Texas Instruments, *Clock-Tree-Architect: Clock Tree Architect Proogramming Software*, product page.
5. Texas Instruments, *PLLATINUMSIM-SW Texas Instruments PLLatinum Simulator Tool* ., tool.

9 Appendix A: Clock Tree Architect (CTA) Detailed Step-By-Step Guide

To show how to use CTA, consider the following example:

- The user is are an aerospace and defense customer.
- The user prefers clocks with extended temperature or high-reliability (-EP) ratings.
- The user is using using a very clean 10fs 100MHz reference.
- The user wants to clock the ADC12DJ5200RF with a CML 5200MHz IC clock (not standalone passive components).
- The clock must generate a SYSREF signal for multi-converter synchronization.

Once you have defined your application's requirements:

1. Open Clock Tree Architect in the browser and on the first drop-down under *Application*, select your application. For this example, select *Aerospace & defense (non space)*. CTA auto-fills default values in the *System configuration* section and the *Available input(s) from your system* sections based on the selection.
2. Set your application requirements under *System requirements*. For this example, the aim is to maximize the converter's SNR, so set the *Jitter metric* to *Very Important* and all other parameters to *Less important*, as shown on Figure 9-1.

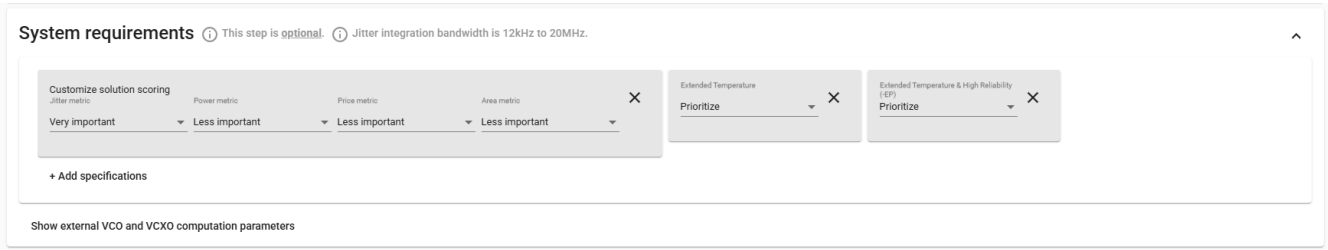


Figure 9-1. System Requirements Section

   a. Refine system requirements by clicking on the *+Add specifications* button. For this example, we selected *Extended Temperature & High Reliability (-EP)* and *Extended Temperature* and set both to *Prioritize*, as shown on Figure 9-1. Select *Require*, *Deprioritize*, and *Exclude*. *Require* and *Exclude* do not consider certain parts from the clocking database while *Prioritize* and *Deprioritize* consider all parts in the database and give certain parts more weight. Other system specifications include programming methods, standard compliance, and overall system power, area, jitter, and price budgets, as shown in Figure 9-2.

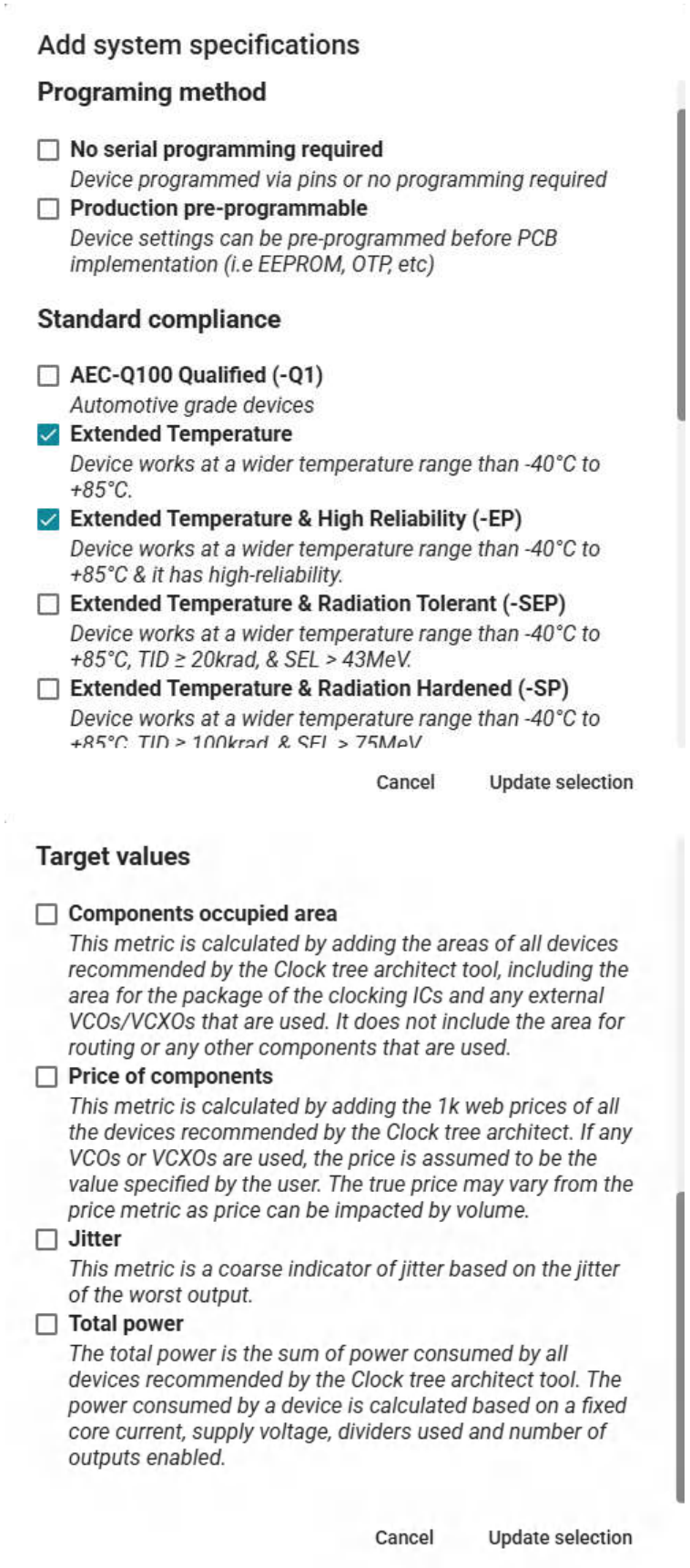


Figure 9-2. Complete List of System Requirements

3. Enter your output specifications under *Outputs*. Set *Frequency* to *5200MHz*, keep *Clock* count as 1, and set *Format* to *CML*, as illustrated in Figure 9-3.

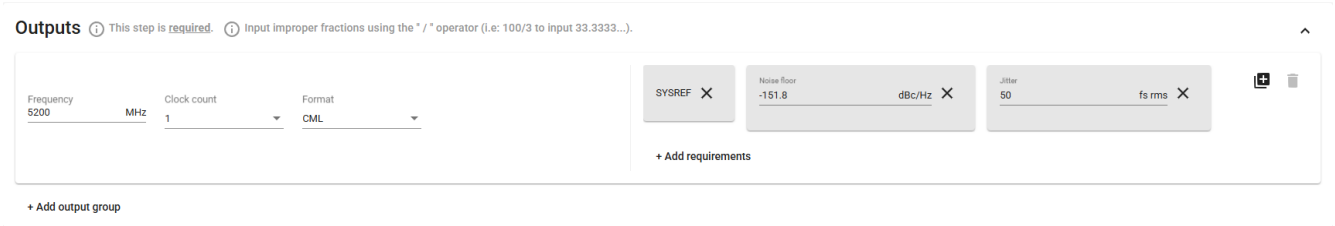
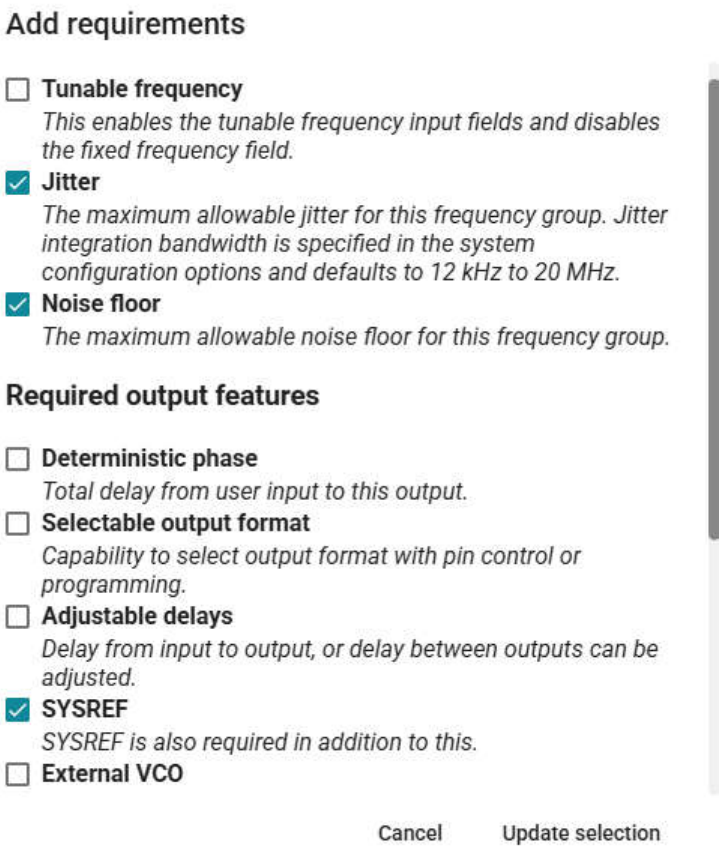


Figure 9-3. Outputs Section

a. Add requirements to your outputs by selecting *+Add requirements*. For this example, we need to set jitter and noise floor, and select SYSREF.
   i. Select *Jitter, Noise Floor,* and *SYSREF* options, as shown on Figure 9-4
   ii. Click on the *Update selection* button.
   iii. In the jitter and noise floor boxes, set the jitter and noise floor to the values of the ADC12DJ5200RF, which are 50fs and -151.8dBc/Hz, as demonstrated on Figure 9-3. These target values were chosen to relax the clock performance requirements and ensure CTA gives a design since these example wants to use an IC. Once the designs are given, they can be analyzed using Section 5

b. Set other output requirements such as adjustable output delays, selectable outputs formats, or deterministic phase as shown in Figure 9-4.



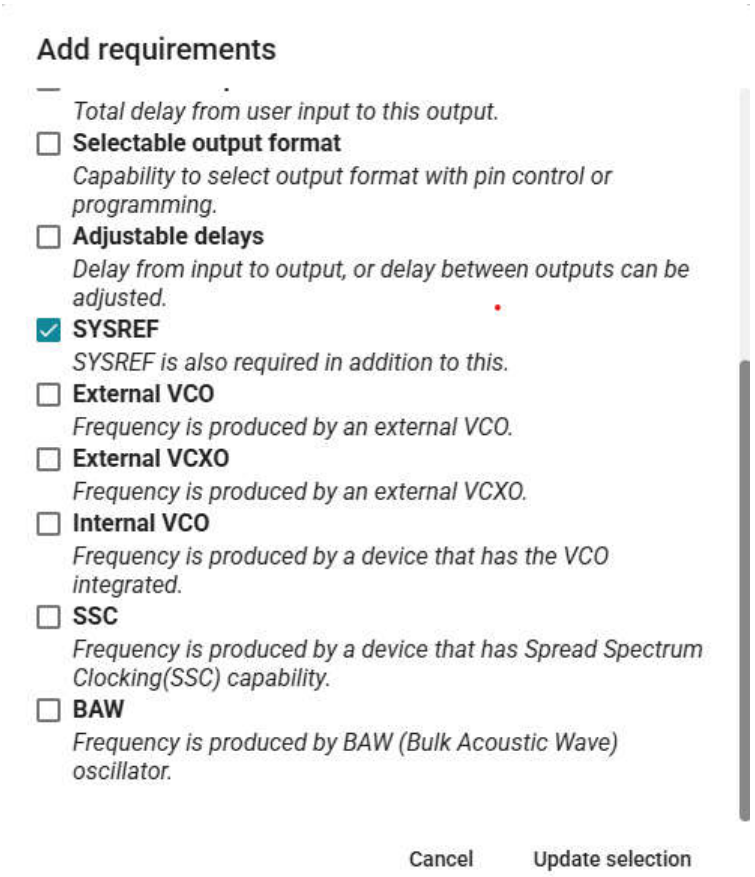


Figure 9-4. Complete List of Output Requirements

4. If the user wants CTA to consider a reference/clock that already exists in the system, click on *+ Add available input from your system* under the *Available input(s) from your system* section. For this example, the default of the *Aerospace & defense (non space)* already sets the input to 100MHz with 10fs of jitter, which matches this example. This is shown in Figure 9-5.

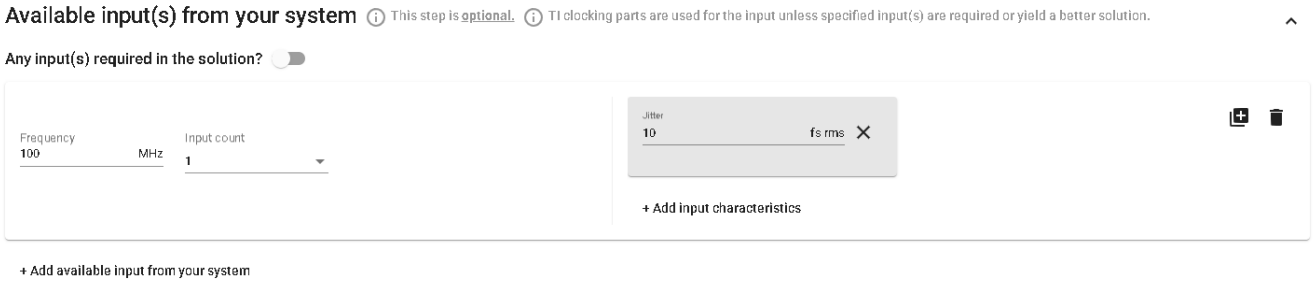


Figure 9-5. Available Input From Your System Section

   a. To force CTA to use this input, toggle *Any input(s) required in the design?*.
   b. To define input characteristics, click on *+Add input characteristics*, update any fields as needed based on the application, and click on *Update selection*, as shown on Figure 9-6.

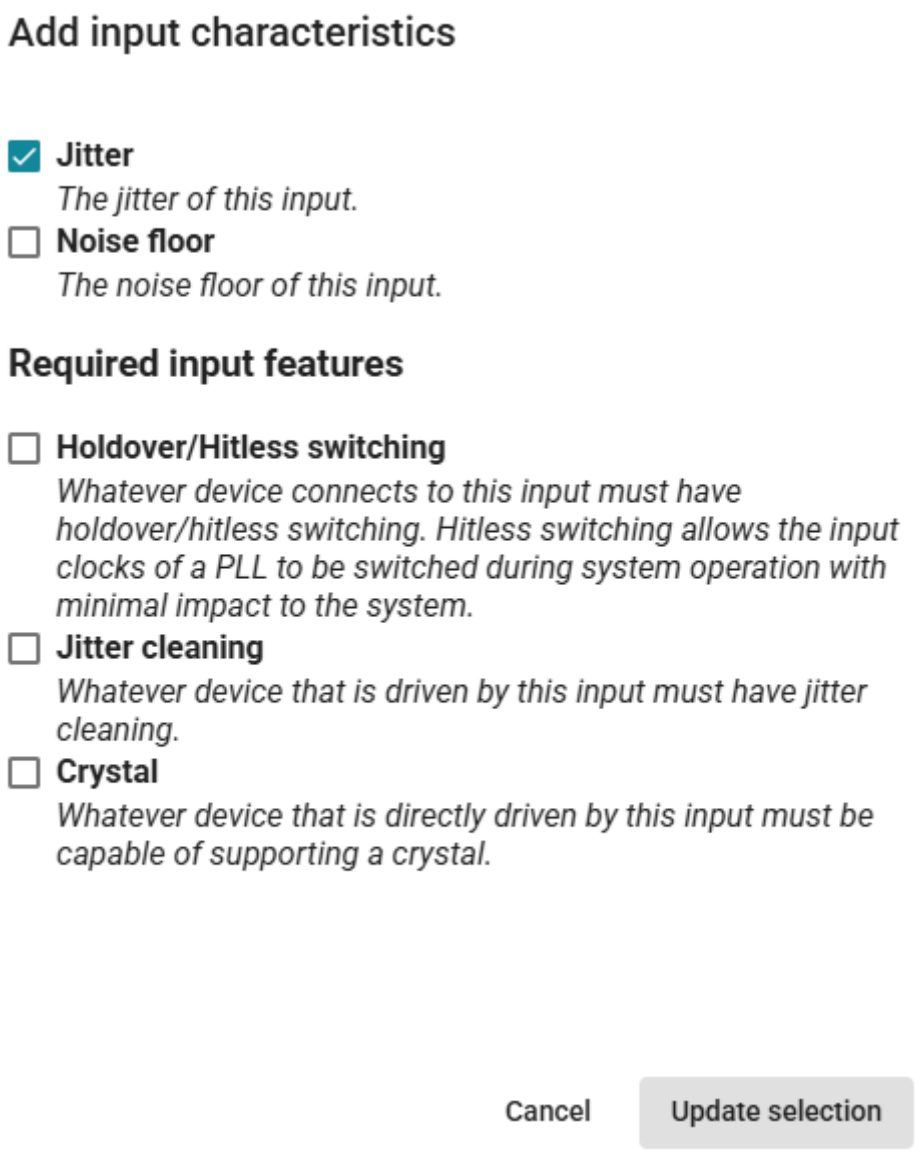


Figure 9-6. Complete List of Input Characteristics from an Already Selected Reference

c.

5. After filling in all sections that pertain to your application, click the *Create designs* button on the bottom right. Figure 9-7 demonstrates the finalized system setup for CTA to propose viable clocking suggestions.

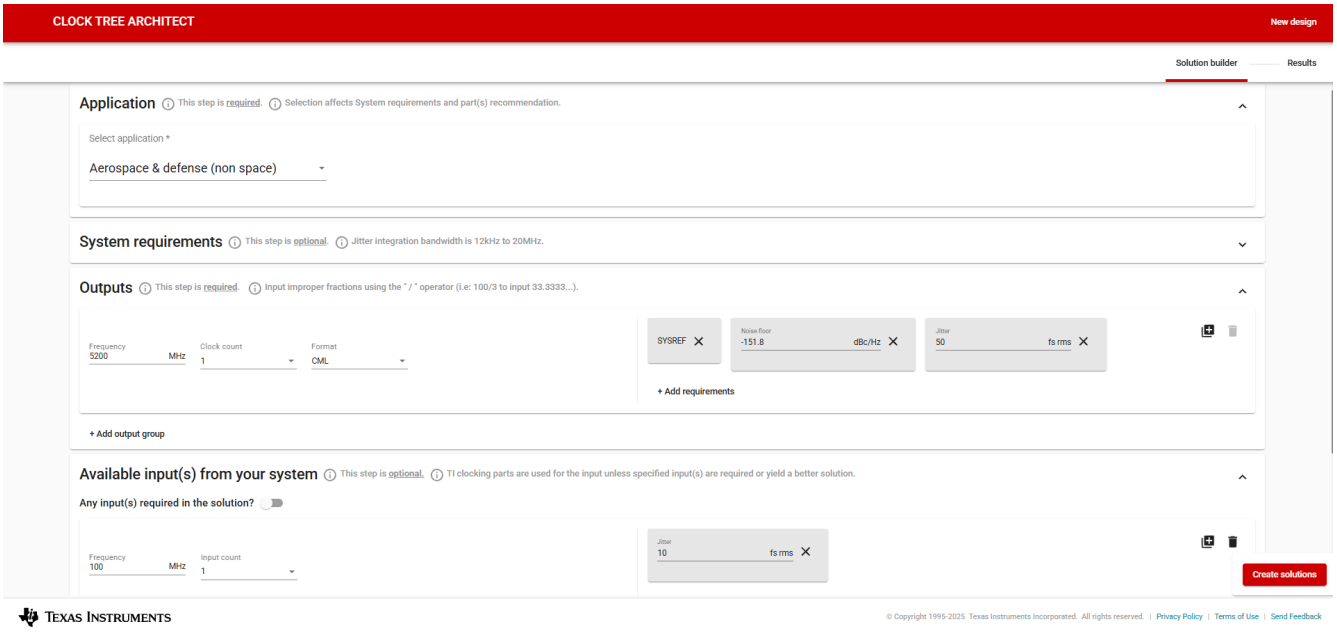


Figure 9-7. Finalized CTA Setup

6. After a few minutes (sometimes several minutes), CTA generates a list of clocking schemes that meet your system requirements. The results are organized from TI's best recommendation to worst. Figure 9-8 shows the results for this example, clearly recommending the LMX2820.
   a. For more details on the proposed design, click on the *Export PDF* button on the bottom right corner.

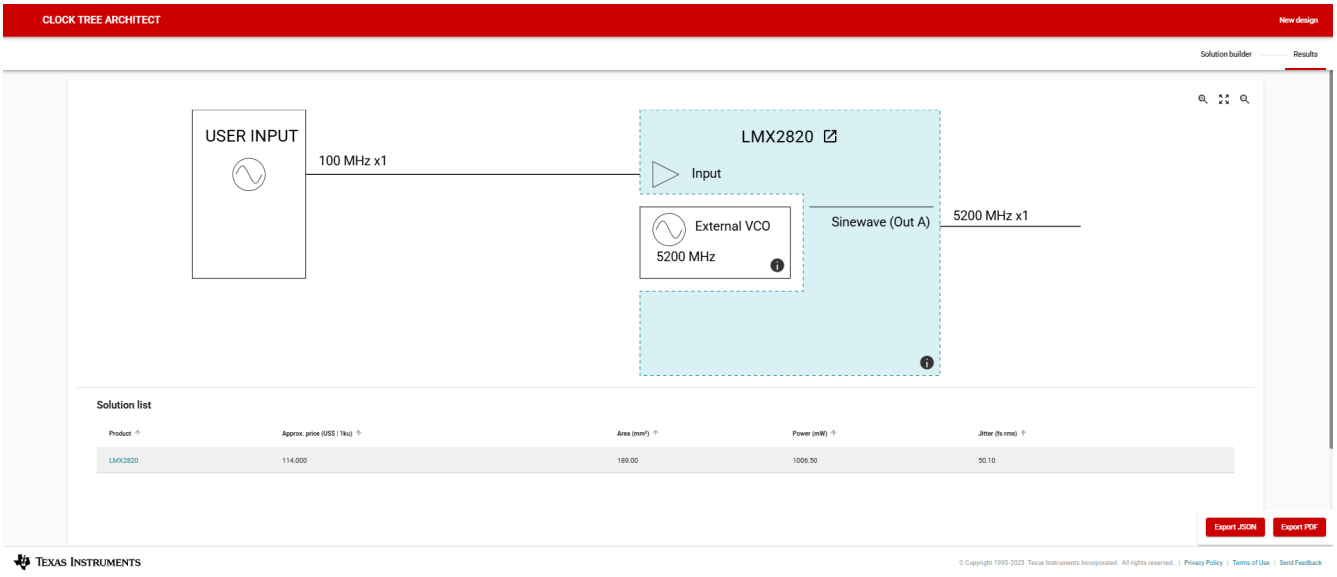


Figure 9-8. Results Page Showcasing Top Clocking Recommendations Matching Your Application's Requirements

   b. Sometimes, there is no clocking designs that meet all requirements. In that case, CTA recommends the closest possible match.

10 Appendix B: Using PLLatinum Sim to Represent Clock Jitter for an Application's Integrated Bandwidth

As mentioned previously, CTA cannot showcase the jitter below 12kHz. Also, the jitter for further out noise is also an estimate rather than an very close interpretation. Instead TI's PLLatinum Sim can be used to properly represent a clock's phase noise curve for the integrated bandwidth of the application. Use the example from Section 9 and specify that an integration bandwidth from 100Hz to 100MHz.is required.

1. Load the clock under consideration by selecting the desired clock under the *Select a Device* box and then clicking on the *Load Device* button, as shown on Figure 10-1. For this example, use the LMX2820.

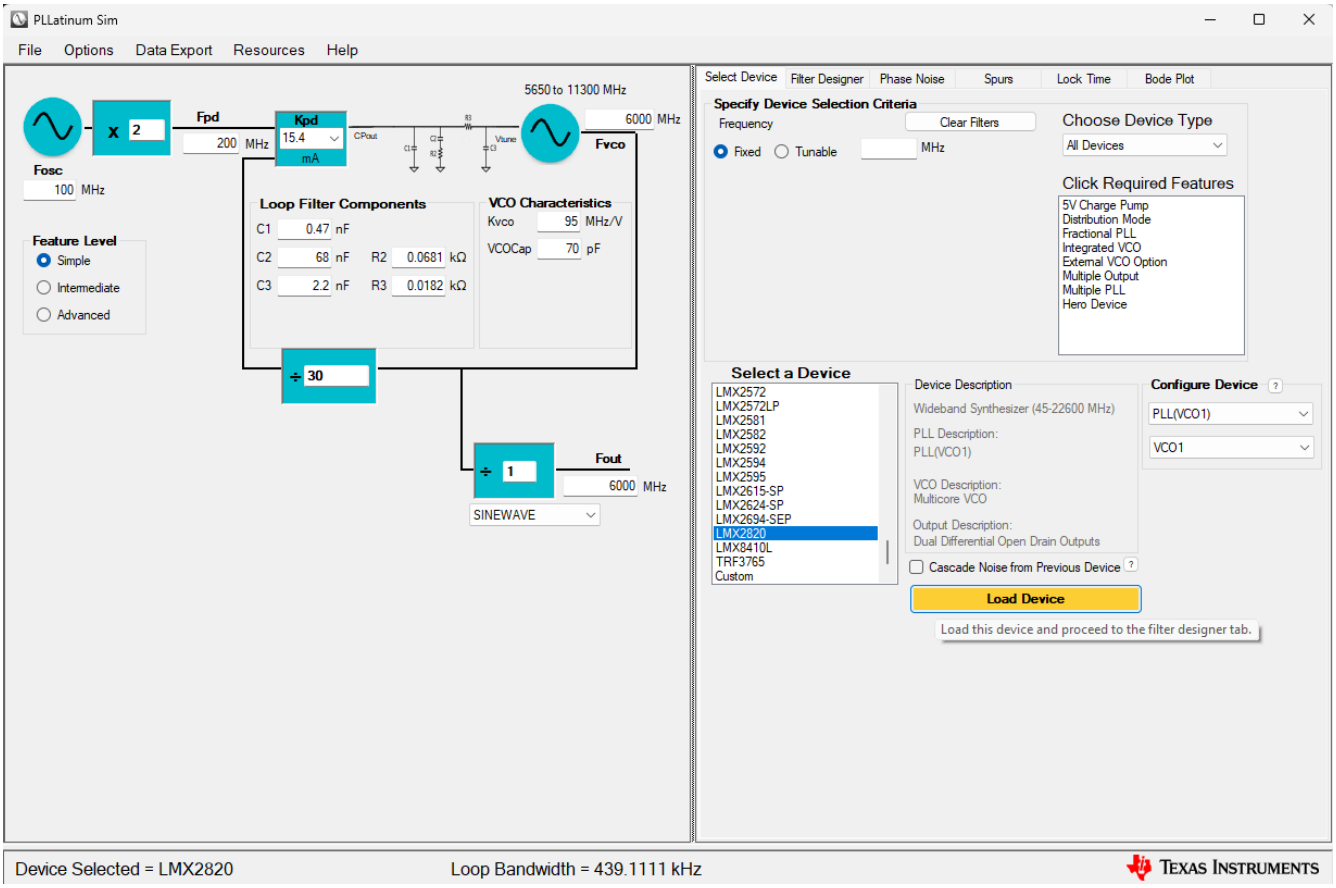


Figure 10-1. Selecting and Loading the LMX2820 into PLLatinum Sim

2. Input the F_{osc} and F_{out} based on system requirements. Following the system from Section 9, F_{osc} is 100MHz and F_{out} is 5200MHz. The step-by-step guide is demonstrated in Figure 10-2.
   a. Click on the *Advanced* option under the *Feature Level* box.
   b. Input F_{out} and then F_{osc}.
   c. Check the box next to *Loop Bandwidth* under the *Filter Parameters* box and then click on the *Calculate Loop Filter* button. This maximizes the jitter for the selected clock.

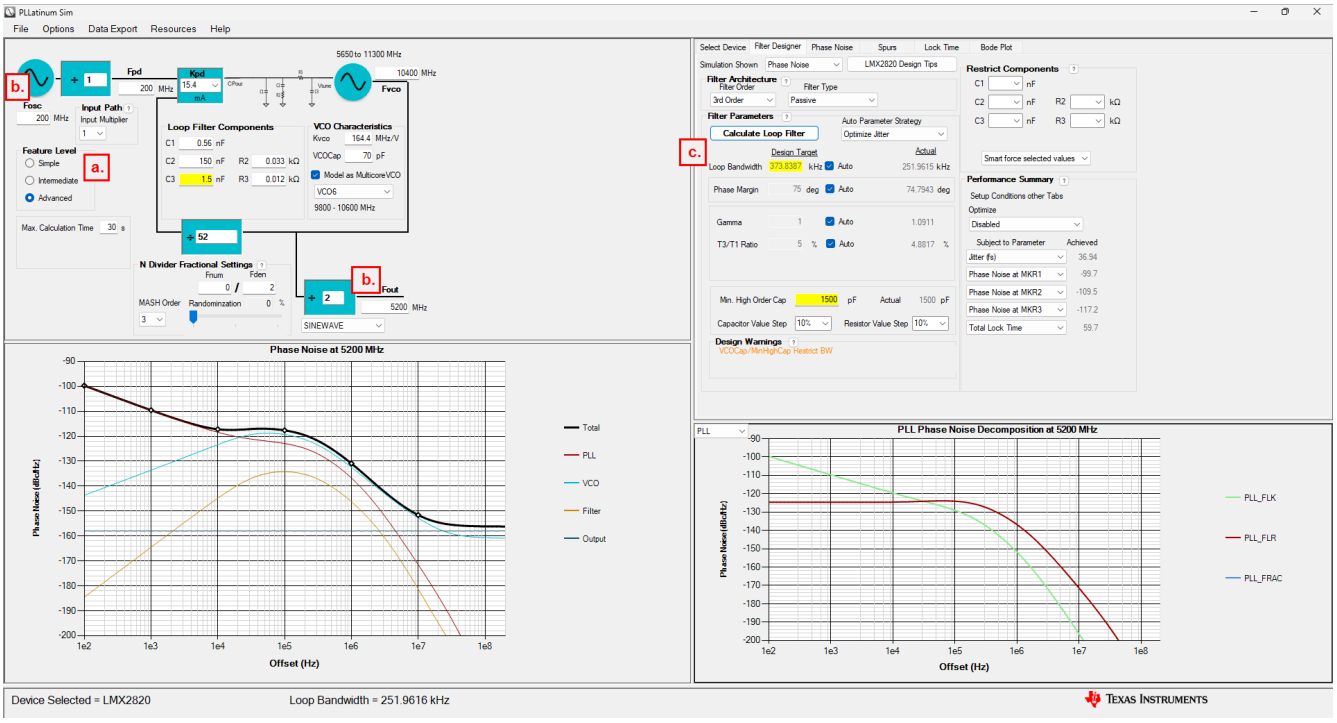


Figure 10-2. Setting up System Requirements of the Application in PLLatinum Sim

3. Click on the *Phase Noise* tab and specify the loop bandwidth under the *Integrated Noise* box. This results in the clock jitter value of the specified clock at the specified output frequency. For this example, the jitter is 38.15fs at 5200MHz output of the LMX2820 (also illustrated in Figure 10-3). This jitter value can then be used for better analysis when picking a clock for the selected data converter.

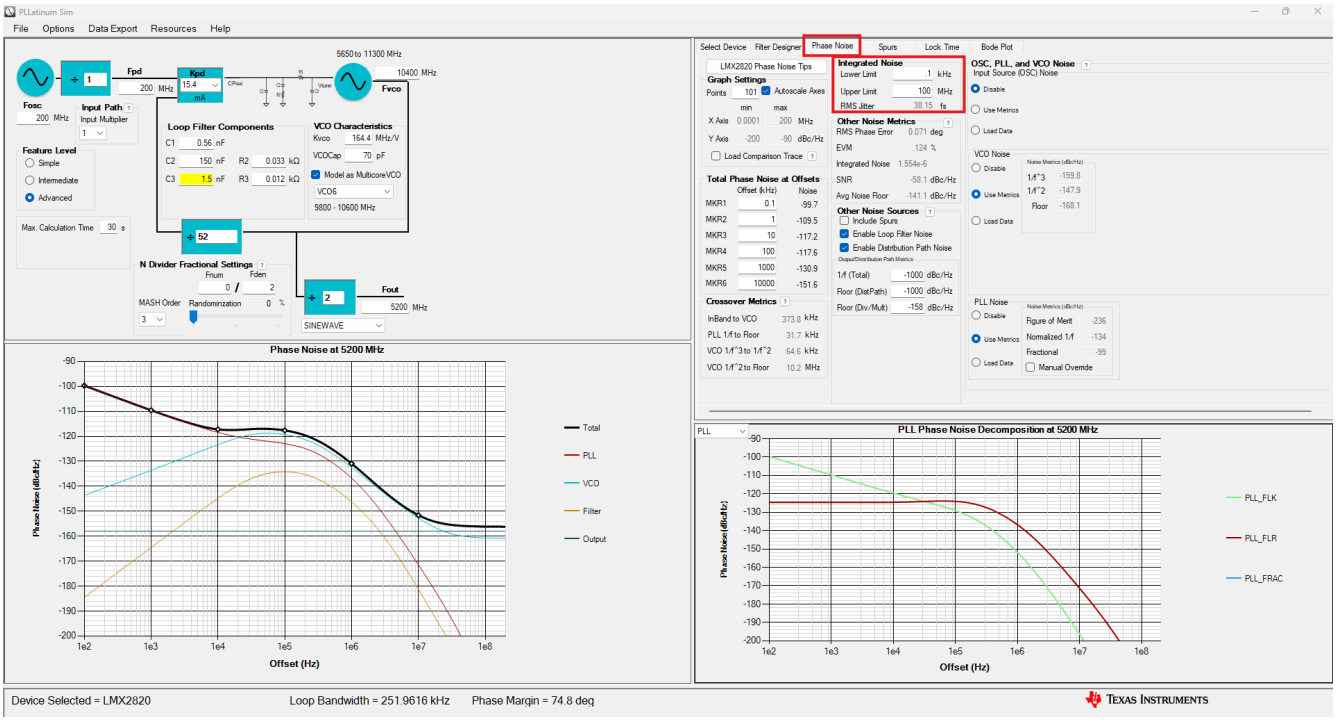


Figure 10-3. Using PLLatinum Sim to Determine Clock Jitter Across Integration Bandwidth of the Application

11 Appendix C: PLLatinum Sim Phase Noise Curves Comparison Step-by-Step Guide

As mentioned in this application note, the phase noise of the clock must be 6dBc/Hz lower then the phase noise of the converter to fully maximize the performance of the converter. This analysis can be done by overlaying the phase noise of both the clock and the converter using TI's PLLatinum Sim. The following steps guide the user on how to upload a phase noise curve on PLLatinum Sim to compare two phase noise curves.

1. Follow the steps from Section 10 to upload the LMX2820
2. Under the *Phase Noise* tab, select *Load Comparison trace* check box under the *Graph Settings* section, as shown in Figure 11-1.

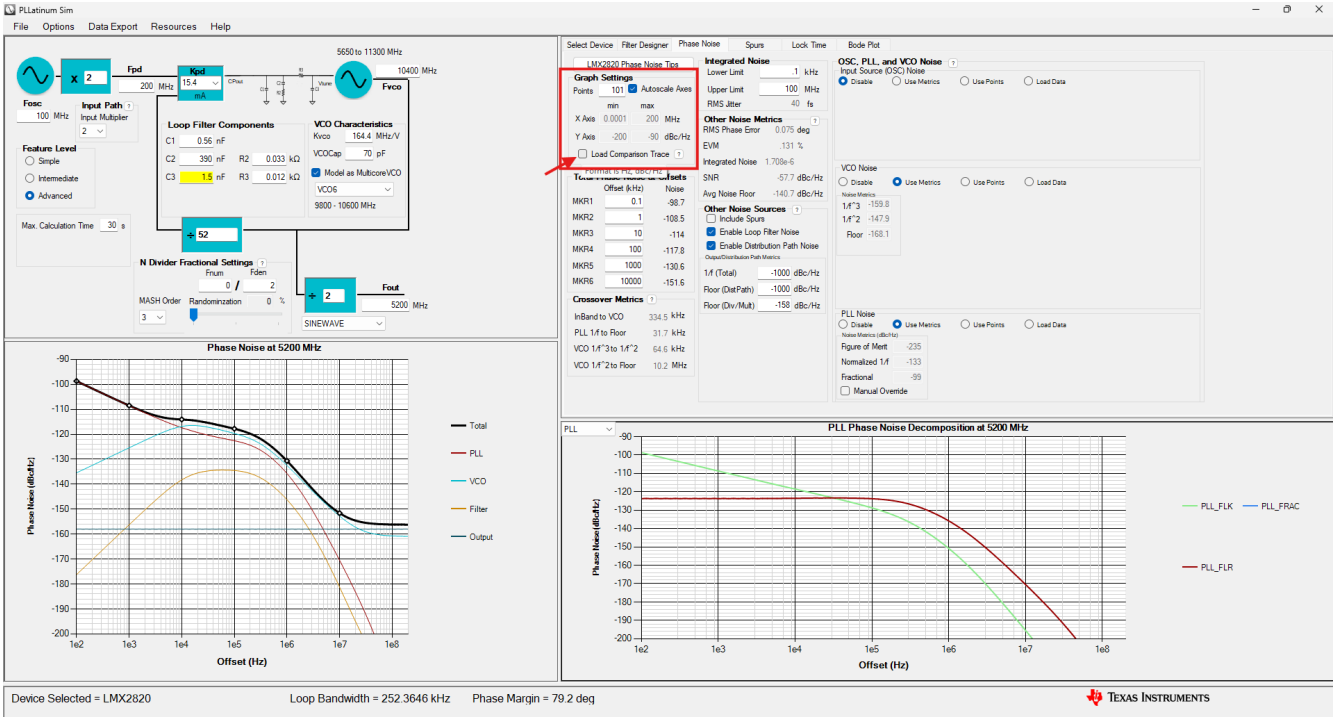


Figure 11-1. Loading a Trace to PLLatinum Sim

3. When the pop-up appears, navigate to the file folder where the phase noise curve of the converter is located, and select the that file. The converter's phase noise curve will appear in the bottom left plot of PLLatinum Sim, labeled *Comparison*, and represented with a red dashed line, as demonstrated in Figure 11-2.

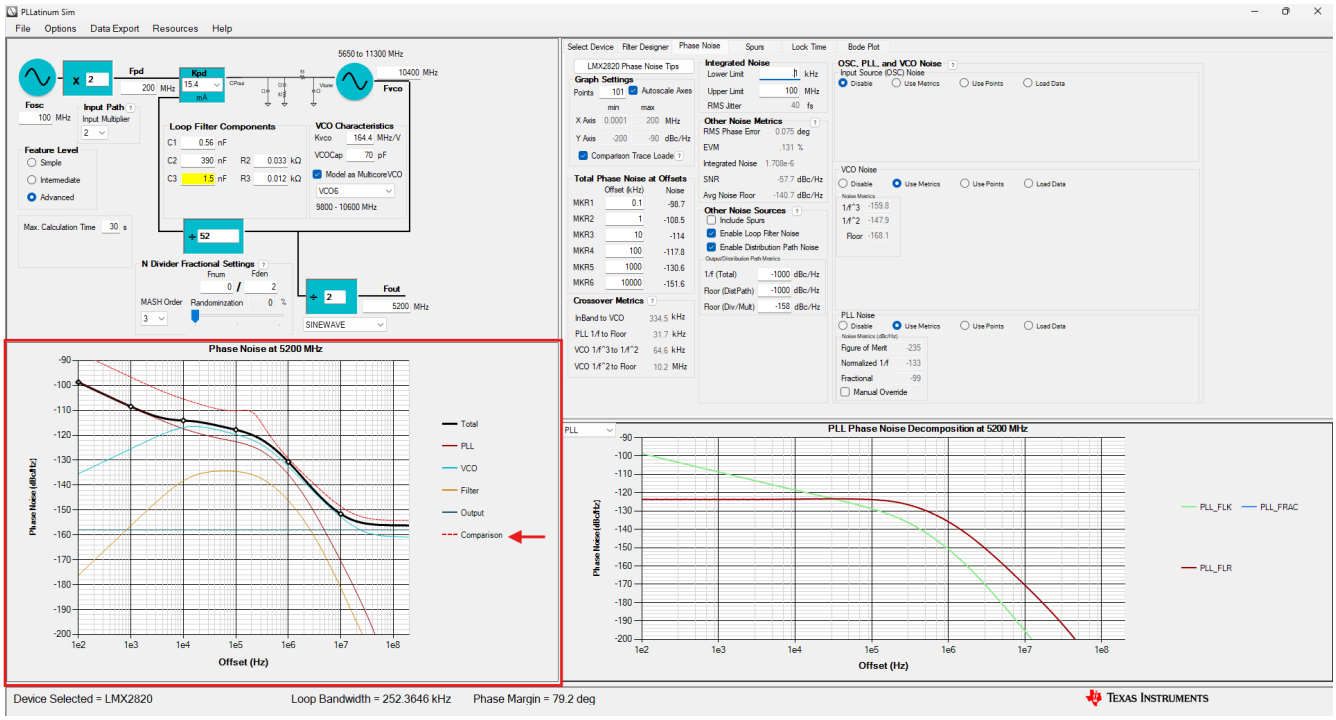


Figure 11-2. PLLatinum Sim Traces' Comparison

Note

Make sure the phase noise curve of the converter is at the same frequency as the output frequency of the clock for accurate comparison.

12 Appendix D: Comparing Phase Noise Curve Simulations of PLLatinum Sim with Measured Data

This appendix demonstrates PLLatinum Sim simulation accuracy. To find TI LMX2820 and LMX2594 clocks' raw phase noise curves, go to CLOCK-PERFDATA-DESIGN page and look for *LMX2820_Phase_Noise_Curves.zip* or *LMX2594_Phase_Noise_Curves.xlsx*.

1. For the LMX2820, download the .zip file of the part to compare, then download the specific clock phase noise curve from a *txt_* folder at the specific frequency to complete the comparison. TI recommends to always use the *txt_pow* folder to use phase noise data showcasing the entire phase noise profile. Phase noise data was taken with an Rhode & Schwartz FSWP with RBW = 3%, XCORR = 100, XCORR optimization on with threshold 10dB, with a sweep/avg count of two and an integration bandwidth from 100Hz to 100MHz. For the LMX2594, download the excel file and copy the offset frequency and the phase noise of the offset into a text file.
   a. To change the integration bandowidth in PLLatinum Sim, follow Section 10.
2. For the LMX2820, input the values for F_{osc}, F_{pd}, K_{pd}, and F_{out} based on the name of the file. For the LMX2594, use the values given in the *Setup* sheet. The LMX2820 is used as the example and was configured with F_{osc} = 200MHz, F_{pd} = 200MHz, K_{pd} = 15.4mA, and F_{out} = 5200MHz
3. Change the *Feature Level* to *Advanced*.
4. Select the check box next to *Loop Bandwidth* under *Filter Parameters* and click on the *Calculate Loop Filter* button. This simulates the measured phase noise curve.

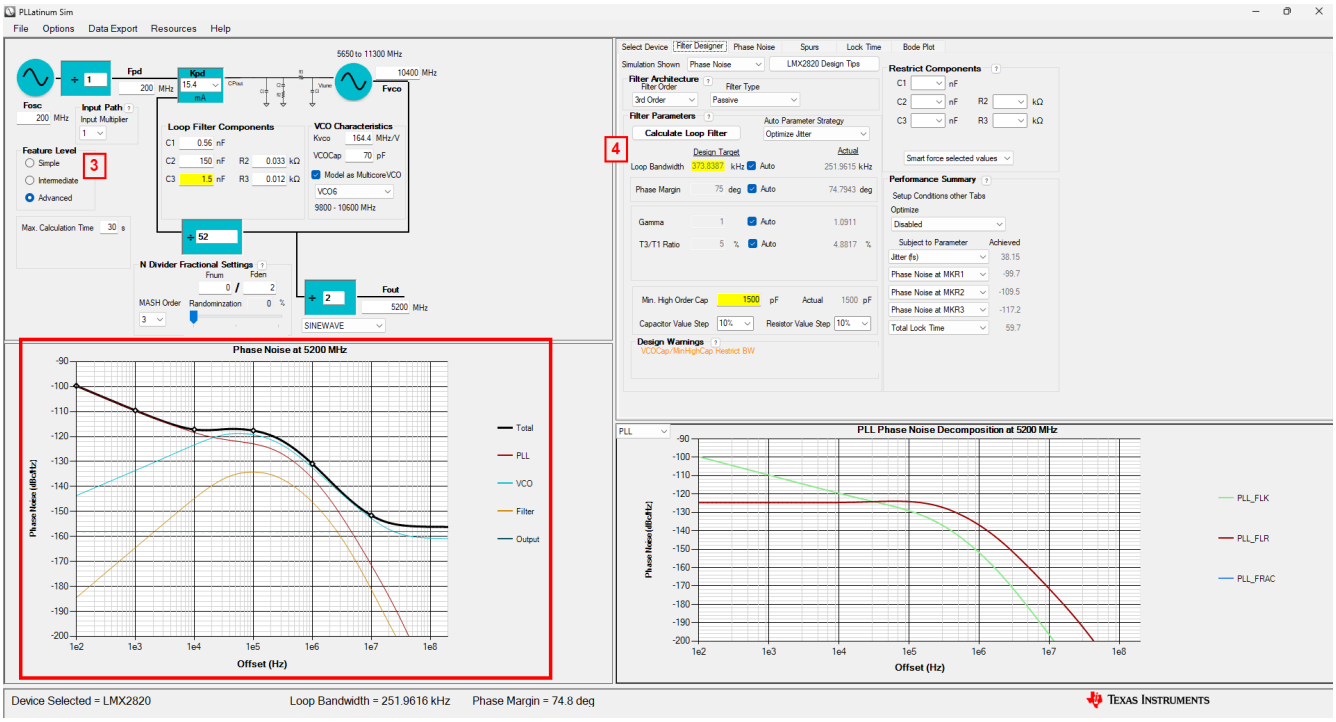


Figure 12-1. Loaded LMX2820 TICS Config onto PLLatinum Sim to Simulate Phase Noise Curve

5. For the next few steps, the LMX2820 is used. Upload the phase noise curve of the LMX2820 file downloaded in step 1 to PLLatinum Sim by following the steps from Section 11.
6. Compare the simulated phase noise curve (black) with the measured phase noise curve (red, dashed). As can be seen on Figure 12-2, the black and red dashed line have the same phase noise.

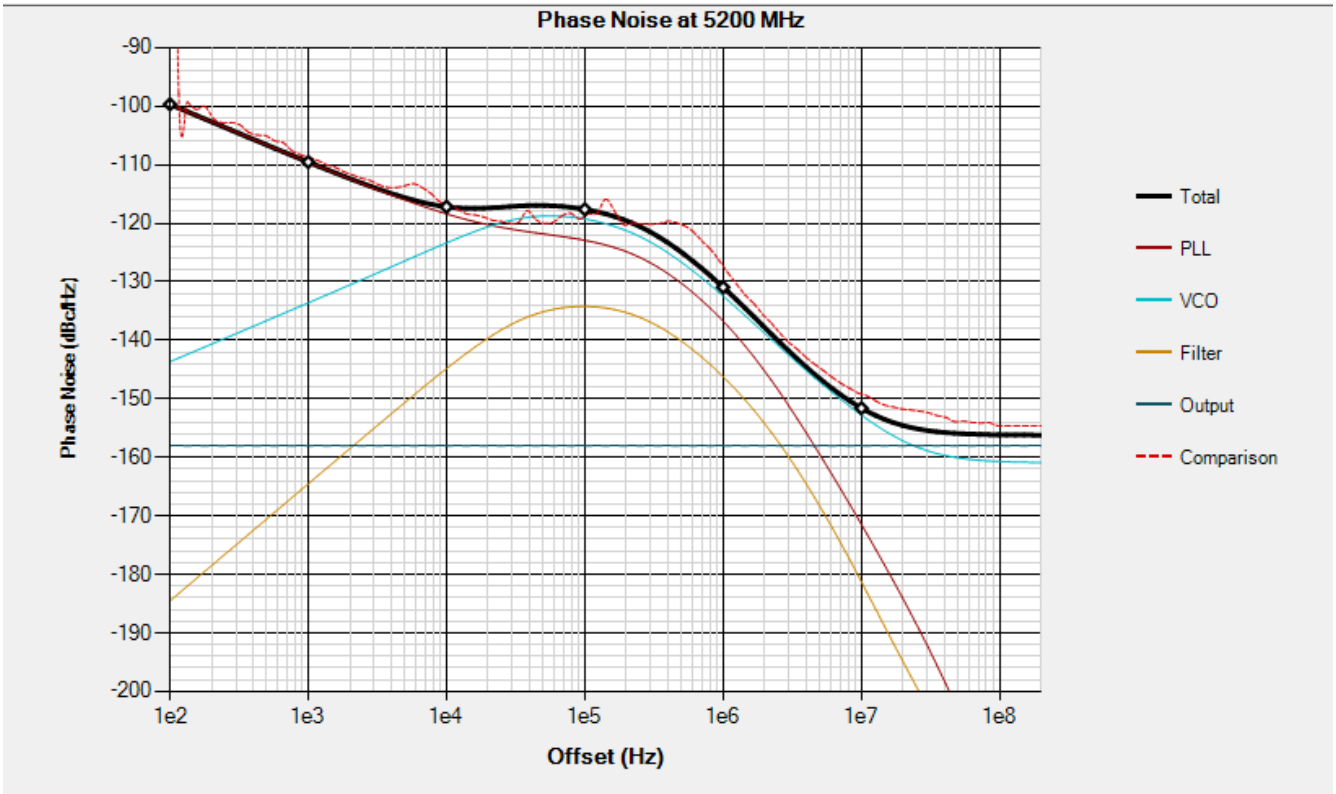


Figure 12-2. Loaded Comparison Trace to PLLatinum Sim to Complete Comparison Between Simulation and Measured Data

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL 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 AND 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 skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025