ADC Modeling Tools Speed Up Evaluation

By Umesh Jayamohan

Introduction

Consumer demand for faster, smarter, better products is pushing innovation to previously unseen levels. System designers are thus faced with a common dilemma: design a new product on a known platform, making only incremental changes; or use a brand-new platform with the most advanced products and capabilities. The former may be quick and low risk, but offers a low reward; whereas the latter offers better versatility, functionality, and value, but with higher risk.

Now, a new suite of simulation tools enables quick prototyping in software, minimizing development risk and providing designers with confidence that their new products will work as expected. The software offers insight into the capabilities of individual products such as analog-to-digital converters, clock ICs, and amplifiers, and makes it possible to combine devices (ADC and clock, for example) without having to obtain actual components. The decision to purchase hardware can be made after the software evaluation is complete, saving time and money.

This article demonstrates the versatility of ADIsimADC®, ADIsimCLK™, and the VisualAnalog® software suite to predict an ADC’s performance when combined with a sampling clock. The example uses the AD9467 16-bit, 250-MSPS analog-to-digital converter and the AD9523-1 low-jitter clock generator. The first section describes the software evaluation, simulating the ADC’s performance over frequency and showing how to connect the devices in software. The second section details a real-world hardware setup, using the evaluation boards and SPIController software. The example clocks the AD9467 at 245.76 MSPS. The AD9523-1 evaluation board uses an interactive GUI to configure the clock outputs.

Cosimulation Using ADIsimADC and ADIsimCLK

First, download and install VisualAnalog and the AD9523-1 evaluation software. ADIsimADC is packaged along with VisualAnalog. Upon starting VisualAnalog, a pop-up window will ask the user to choose a Canvas, as shown in Figure 1. The ADIsimADC model for the AD9467 is available under the ADC→Single→AD9467 menu selection. Figure 2 shows the ADIsimADC Average FFT Canvas.

Figure 1. VisualAnalog New Canvas window.

Figure 2. ADIsimADC Canvas showing AD9467 with single-tone FFT at 9.7 MHz.

Setting up ADIsimADC to Predict ADC Behavior

Input a frequency in the Tone Generator block and hit Tab. ADIsimADC automatically converts this frequency into a coherent frequency based on the sample rate and the sample size. Figure 3 shows the FFT of a single-tone input at 9.7 MHz using default settings.

Figure 3. ADIsimADC single-tone FFT at 9.7 MHz.
Setting up ADIsimCLK to Predict AD9523-1 Behavior

Next, download and install the ADIsimCLK software. Once installed, open the program and select File→New. A window appears with a selection of devices, as shown in Figure 4.

![Figure 4. ADIsimCLK device selection.](image)

Follow the setup method that best resembles the actual intended system implementation. In this example, an external 30.72-MHz clock provides the reference to the first PLL. A Crystek CVHD-950 is used as the VCXO for the first loop of the dual-loop PLL. The internal VCO frequency is set to 2949.12 MHz and internally divided by 3. A divide-by-4 on OUT7 provides a 245.76-MHz clock. This setup is shown in Figure 5.

![Figure 5. AD9523-1 setup in ADIsimCLK.](image)

ADIsimCLK also generates a report on the clock outputs, including output phase noise and jitter at various integration ranges. These reports are available on the tabs corresponding to the various outputs. In this setup, OUT7 is being used to clock the AD9467 evaluation board. The report page is shown in Figure 6. The key specification, Broadband Jitter, is highlighted.

![Figure 6. OUT7 report in ADIsimCLK.](image)

Simulating the AD9467 with the AD9523-1

ADIsimADC can predict the performance of the AD9467 when clocked with the AD9523-1. The broadband jitter specification from the ADIsimCLK report can be passed to the ADIsimADC canvas. In the FFT canvas, the ADC Model block enables the user to update the total jitter specification, as shown in Figure 7.

![Figure 7. Updating the jitter in the ADIsimADC model.](image)
The total jitter can be calculated as the root-sum-square (rss) of the individual jitter components. In this case, the aperture jitter is 60 fs and the broadband jitter is 215 fs. The rss jitter passed to ADIsimADC is 223.2 fs, producing the 97-MHz, single-tone FFT shown in Figure 8. Using the updated jitter, ADIsimADC can predict the expected performance at any input frequency.

A Short Note on Jitter
An ADC must take regular time samples of an analog signal. A stable sampling clock is required, as any nonideal clock source will produce some phase noise. Jitter is the integration of the phase noise over the period between two specified frequency offsets from the sample clock carrier. For ADCs, wideband noise is generally considered to be most important. ADIsimCLK calculates broadband jitter, which integrates the phase noise above a 1-kHz offset. This broadband jitter is passed to the ADIsimADC model to understand the effect of jitter on the ADC performance. For more detailed information on how sample clock jitter affects ADC performance, please refer to AN-756 Application Note: Sampled Systems and the Effects of Clock Phase Noise and Jitter.

Measured Performance
The ADIsimADC prediction was tested using the AD9467 evaluation board and the AD9523-1 evaluation board. The AD9523-1 was configured to produce a 245.76-MHz LVPECL clock on OUT7. This output was coupled to the AD9467 evaluation board, which was modified to accept a differential clock input on J200 and J201. This setup is shown in Figure 9.

The setup gathered data using analog input frequencies from 2 MHz to 400 MHz. Single-tone FFTs were captured using VisualAnalog, and data was compiled and plotted against the ADIsimADC prediction. Figure 10 shows the SNR vs. frequency. Note how well the simulation matches the actual measurements.

Figure 8. Single-tone FFT at 97 MHz input with updated jitter specification in ADIsimADC.

Figure 9. Hardware setup showing the AD9523-1 and AD9467 evaluation boards.

Figure 10. SNR vs. analog input frequency comparing ADIsimADC and actual measured data.

Figure 11. SFDR vs. analog input frequency comparing ADIsimADC and actual measured data.

The setup gathered data using analog input frequencies from 2 MHz to 400 MHz. Single-tone FFTs were captured using VisualAnalog, and data was compiled and plotted against the ADIsimADC prediction. Figure 10 shows the SNR vs. frequency. Note how well the simulation matches the actual measurements.

Figure 11 shows the SFDR data. These numbers do not completely agree, but the overall trend between simulated and measured data across frequency matches well. SFDR is highly dependent on PC board layout, components, clock amplitude, which can explain the discrepancy.
A better measure of distortion would compare the simulated and measured data for second and third harmonic distortion, as shown in Figure 12 and Figure 13. The simulated and measured HD2 performance matches very well, proving that the differential signal going into the ADC in the evaluation board is well balanced in terms of amplitude and phase, and that the layout of the evaluation board is good enough to not significantly affect the differential signal balance.

On the other hand, predicting HD3 performance over frequency can be tricky. The ADIsimADC model is developed by looking at the ADC performance and DNL data during characterization. The algorithm uses interpolation and extrapolation techniques to predict dynamic range at certain frequencies, but cannot accurately predict the HD3 performance at all points.

The actual HD3 performance is heavily dependent on real-world factors such as supply voltage, component selection, ADC input buffer, and clock signal quality.

The HD3 prediction may not always be spot on, but the overall trend over frequency provides good agreement between the simulated and measured data.

In many system designs, the primary performance metric is SNR. SFDR and dynamic range are dependent on a lot of other factors. The SNR numbers between the simulation and the measured data match very well, which gives system designers confidence in the choice of ADC and clock.

**Conclusion**

The need for shorter design cycles puts enormous pressure on system designers to evaluate new products for their designs. Hardware evaluation is almost always a necessity, but choosing the wrong hardware combination could end up costing money and time. Software evaluation can act as a quick and easy first look at ADC products. ADIsimADC and ADIsimCLK offer a simple and effective way for system designers to select an ADC and clocking IC. These software tools allow system designers to mix and match various ADCs and clocking ICs, allowing them to develop enough confidence in the selected components to perform a hardware evaluation.

**Acknowledgements**

Thanks to Jillian Walsh for all the hard work in the lab collecting the data for this paper and thanks to Kyle Slightom for helping with the AD9523-1 evaluation board and software setup.

**References**


MT-003 Tutorial. Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don’t Get Lost in the Noise Floor.
