

# Continuous-Time Sigma-Delta (CTSD) Precision ADC Minitutorial



# Table of Contents

---

Chapter 1: How to Improve Your Precision ADC Signal Chain Design Time	PAGE 3
Chapter 2: CTSD Architecture Explained for Signal Chain Designers	PAGE 9
Chapter 3: Inherent Alias Rejection Made Possible	PAGE 14
Chapter 4: Ease of ADC Input and Reference Drive Simplify Signal Chain Design	PAGE 20
Chapter 5: Digital Data Interface Simplification with Asynchronous Sample Rate Conversion (ASRC)	PAGE 25
References	PAGE 32

# Chapter 1: How to Improve Your Precision ADC Signal Chain Design Time

Abhilasha Kawle, Senior Analog Design Engineer, and Wasim Shaikh, Applications Engineer

## Abstract

Precision signal chain designers are challenged to meet noise performance demands in medium bandwidth applications and often end up making a trade-off between noise performance and accuracy. Achieving faster time to market and getting their design right the first time can add further pressure. Continuous-time sigma-delta (CTSD) ADCs bring out inherent architectural benefits and simplify signal chain design to reduce solution size and help customers achieve faster time to market for their end products. In this minitutorial, we will explain the inherent architectural benefits of CTSD ADCs and how they are adapted to a wide range of precision medium bandwidth applications. We will take a deep dive into signal chain design to educate designers on the key advantages of CTSD technology and explore the [AD4134](#) precision ADC's ease of design features.

## Introduction

In many digital processing applications and algorithms, the demand to have better resolution and precision for all converter technologies has increased over the last two decades. The limited resolution/precision of ADCs was enhanced by using an external digital controller that would extract and deliver more precise results using software techniques such as averaging and optimized filtering schemes. To reduce extensive postprocessing at the digital microcontroller or DSP, designers could use a high performance precision ADC. This would reduce optimization time at the digital side, and a lower cost microcontroller or DSP could also be considered. The applications and markets for precision ADCs are widespread:

- ▶ Industrial instrumentation: vibration analysis, temperature/pressure/strain/flow measurements, dynamic signal analysis, acoustic analysis

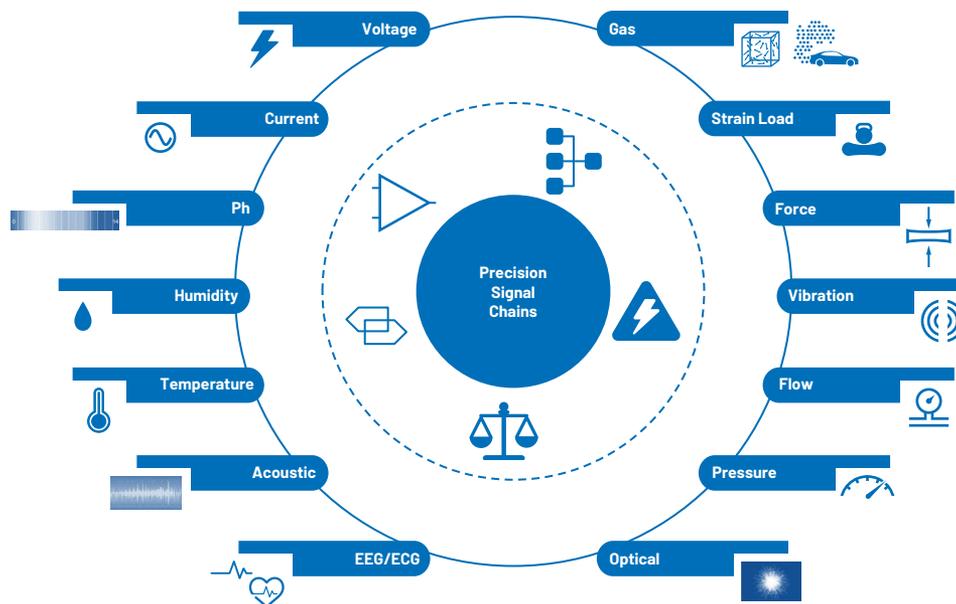


Figure 1. Precision ADC signal chain examples.

- ▶ Medical instrumentation: electrophysiology, blood analysis, electrocardiogram (EKG/ECG)
- ▶ Defense applications: sonar, telemetry
- ▶ Test and measurement: audio test, hardware in loop, power quality analysis

The analog input signal to be processed by an ADC could be a sensor signal with voltage, current output, or a feedback control loop signal with bandwidth ranging from dc to a few hundred kHz. The ADC digital output format and rate are dependent on the application and postprocessing required by the following digital controller. In general, signal chain designers follow the Nyquist sampling theorem and program the ADC's output data rate (ODR) for the digital controller to be at least twice the input frequency. Most ADCs provide the flexibility to tune the output data rate based on the signal frequency band of interest.

For currently available ADCs, there are several signal conditioning stages involved before the ADC can interact with the input signal. Signal conditioning circuits with stringent requirements need to be designed and tailored around specific and individual ADC technologies to ensure that ADC data sheet performance can be achieved. A signal chain designer's job doesn't stop after the selection of the ADC. Considerable time and effort are often required to design and fine tune this surrounding periphery. Analog Devices provides a high level of technical support in the form of design simulation tools and models to overcome most of these inherent design challenges.

## A New Approach: Easing the Design Journey with CTSD Architecture

CTSD architecture, which has been predominantly used in audio and high speed ADCs, is being tailored for precision applications to achieve the highest precision while leveraging its unique signal chain simplification properties. The advantages of this architecture remove the burdens involved with designing the periphery. Figure 2 shows a small snippet of how current ADC signal chains can be simplified and shrunk by 68% by using this new solution to enable high channel densities.

To illustrate the simplification that CTSD ADC technology brings to the signal chain, this article highlights some of the key challenges involved in incumbent signal chain design for general applications, as well as shows how CTSD ADCs ease these challenges.

So, let's start with a few design steps involved in incumbent signal chains with the very first task being the selection of the right ADC to best fit the targeted application.

### Step 1: Selecting the ADC

When selecting from the wide range of ADCs that are available, important considerations are resolution and accuracy, signal bandwidth, ODR, signal type, and the range to be processed. Generally, in most of the applications, digital controllers require their algorithms to process amplitude, phase, or frequency on the input signal.

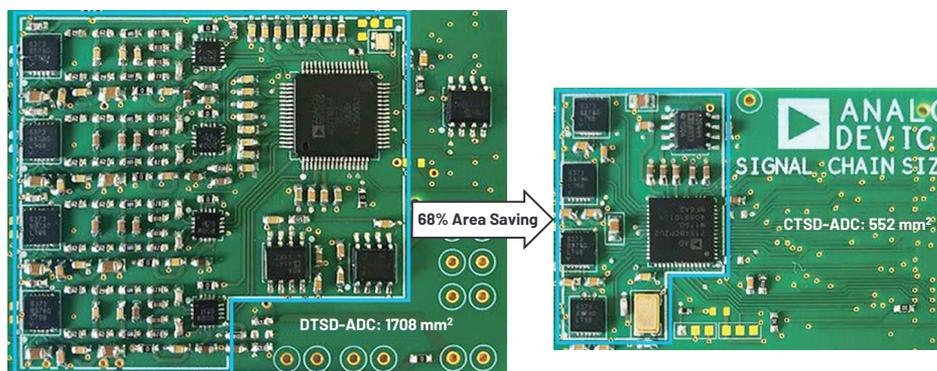


Figure 2. A compact size solution with ADI's new easy to use CTSD ADC.

To accurately measure any of the previous factors, the errors added in the process of digitization need to be minimal. The major errors and their corresponding measurement terminology are detailed in Table 1 and explained in further detail in the [Essential Guide to Data Conversion](#).

**Table 1. ADC Errors and Performance Metrics**

ADC Error	Associated Measurements in Data Sheets
1 Thermal and quantization noise	Signal-to-noise ratio (SNR), dynamic range (DR)
2 Distortion	Total harmonic distortion (THD), intermodulation distortion (IMD)
3 Interference	Crosstalk, alias rejection, power supply rejection ratio (PSRR), common-mode rejection ratio (CMRR)
4 Magnitude and phase error	Gain error, magnitude, and phase droop at frequency of interest
5 Delay from ADC input to final digital output	Latency, settling time

The performance metrics in Table 1 are related to signal amplitude and frequency, and are generally termed as ac performance parameters.

For dc or near dc applications such as power metering dealing with 50 Hz to 60 Hz input signals, ADC errors such as offset, gain, INL, and flicker noise would have to be accounted for. These dc performance parameters also require a certain level of temperature stability relating to an application's intended use.

ADI provides a wide range of industry-leading high performance ADCs to meet the system requirements of several applications, be they precision based, speed based, or based on a restricted power budget. Just comparing one set of ADC specifications to another is not the way to choose an ADC. The overall system performance and design challenges must be considered, and that's where the choice of ADC technology or architecture comes into play. There are two broad classifications of ADC architectures that are traditionally preferred. The most popular is the [successive approximation register \(SAR\)](#) ADC, which follows the simple Nyquist theorem. It states that a signal can be reconstructed if sampled at twice its frequency. The advantages of SAR ADCs are excellent dc performance and small form factors with low latency and power consumption scaling with ODR.

The second technology choice is a [discrete-time sigma-delta \(DTSD\) ADC](#), which works on the principle that the greater the number of samples, lesser is the information lost. So the sampling frequency is much higher than the stated Nyquist frequency, a scheme referred to as oversampling. An added advantage from this architecture is that the errors added due to sampling are minimized in

the frequency band of interest. Because of this, DTSD ADCs have both excellent dc and ac performance but a higher latency.

Figure 3 shows an illustration of the typical analog input bandwidths of both SAR and DTSD ADCs, with some popular product choices at various speeds and resolution. The Precision Quick Search feature can also be referenced to help in your choice of ADC.

Additionally, a new class of precision ADCs are now available. These are based on CTSD ADCs that are on par with the performance of DTSD ADCs but they are unique with regard to simplifying the entire signal chain design process. The challenges highlighted in the next few design steps of an incumbent signal chain can be addressed by this new ADC family.

## Step 2: Interfacing the Input to the ADC

Sensors whose outputs are to be processed by the ADC may have very high sensitivity. Designers must have a good understanding of the ADC input structure to which the sensor will be interfaced to ensure ADC errors don't mask or distort the actual sensor signal.

In conventional SAR, DTSD ADCs, the input structure is known as the switched capacitor sample-and-hold circuit, as shown in Figure 4. At every sampling clock edge when the sample switch changes its ON/OFF state, finite current demand needs to be supported to charge or discharge the hold capacitor to a new sampled input value. This current demand needs to be supplied by the input source, which, in our case of discussion, is the sensor. Additionally, the switch itself has some on-chip parasitic capacitors that inject some charge back onto the source, which is called charge injection kickback. This added error source also needs to be absorbed by the sensor to avoid corruption of the sensor signal.

Most of the sensors are incapable of providing such a magnitude of currents, indicating that they fall short of driving switching circuitry directly. In a different

scenario, say even if a sensor can support these current demands, the sensor's finite impedance would add an error at the ADC input. The charge injection current is a function of input and this current causes an input dependent voltage drop across the sensor impedance. As shown in Figure 4a, the input of the ADC is then in error. One solution to solve these issues is to place a driving amplifier between the sensor and ADC, as shown in Figure 4b.

Now we need to set the criteria for this amplifier. First and foremost, the amplifier should support the charging current and absorb the charge injection kickback. Next, this amplifier's output needs to be fully settled at the end of the sampling edge so that the ADC samples input without added errors. This means the amplifier should have the capability to provide instantaneous current steps that map to having a high slew rate and provide a fast settling response to these transient events, which maps to having high bandwidth. As the sampling frequency and resolution of the ADC increase, meeting these requirements becomes critical.

The big challenge for designers, especially those who work with medium bandwidth applications, is to identify the right amplifier for the ADC. As indicated earlier, ADI provides a set of simulation models and precision ADC driver tools to ease this step, but for a designer, it is an added design step to achieve the data sheet performance of the ADC. Some of the new age SAR and DTSD ADCs have mitigated this challenge by using novel sampling techniques to completely reduce the transient current demand or by having an integrated amplifier. But either solution limits the range of signal bandwidth or penalizes ADC performance.

**The CTSD ADC advantage:** CTSD ADCs address this challenge by providing an easy to drive resistive input instead of a switched capacitor input. This shows that there are no hard requirements of high bandwidth, large slew rate amplifiers. If sensors can directly drive this resistive load, they can be directly interfaced to a CTSD ADC; otherwise any low bandwidth, low noise amplifier could be interfaced between a sensor and a CTSD ADC.

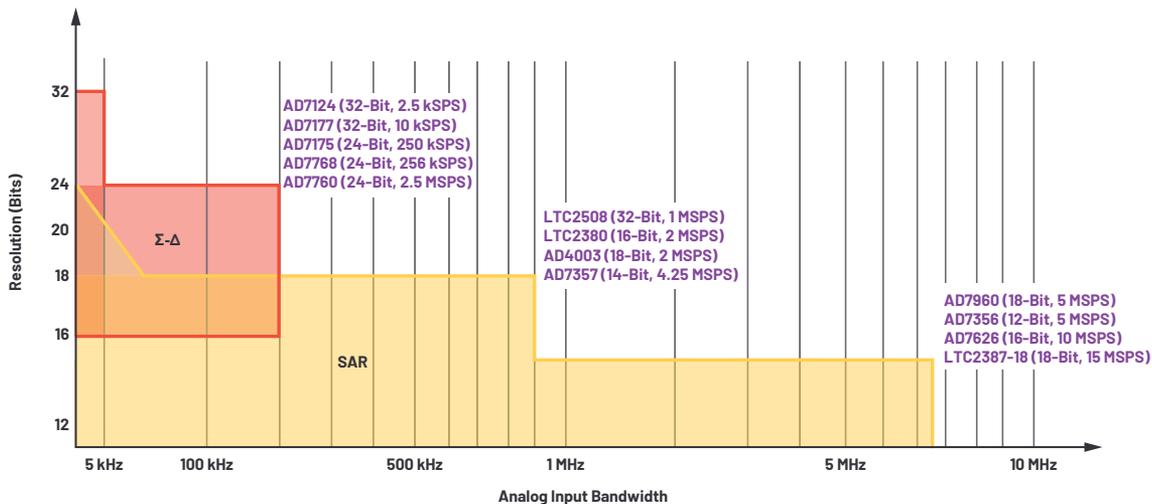


Figure 3. Precision ADC architecture positioning.

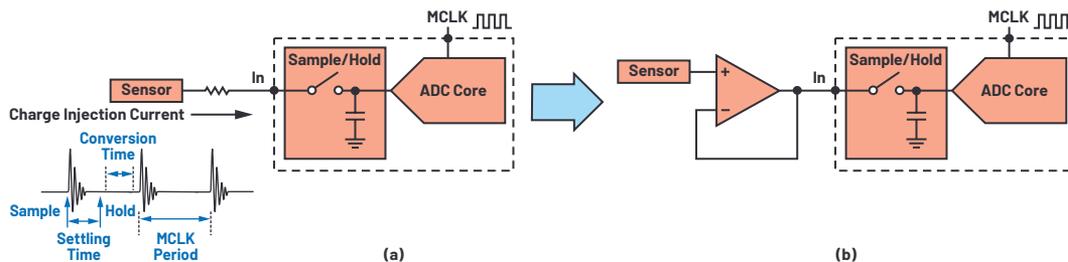


Figure 4. (a) Switched capacitor charge injection kickback into the sensor, and (b) isolating the kickback effect with an input buffer.

### Step 3: Interfacing the Reference to the ADC

The challenge involved with interfacing to a reference is similar to input interfacing. The reference input for conventional ADCs is also a switched capacitor. At every sampling clock edge, the reference source needs to charge the internal capacitors, thus demanding large switching current with good settling time.

The reference ICs available cannot support large switching current demand and have limited bandwidth. The second interfacing challenge is that noise from these references is large in comparison to an ADC's noise. To filter this noise, a first-order RC circuit is used. On one hand, we are band-limiting the reference for noise, while on the other hand, we are demanding fast settling time. These are two opposing requirements to satisfy. For this reason, a low noise buffer is used to drive the ADC reference pin, as shown in Figure 5b. Based on the sampling frequency and resolution of an ADC, the slew rate and bandwidth of this buffer is decided.

Again, like with our precision input driver tools, ADI has tools to simulate and select the correct reference buffers for an ADC. And similar to input, some of the new age SAR and DTSD ADCs also have the option of an integrated reference buffer, but they come with performance and bandwidth limitations.

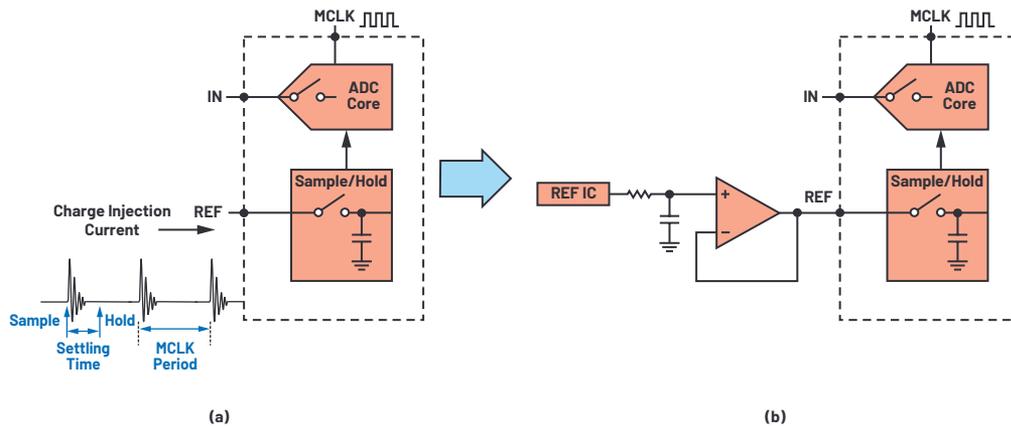


Figure 5. (a) Switched capacitor charge injection kickback into the reference IC and (b) isolating the kickback effect with a reference buffer.

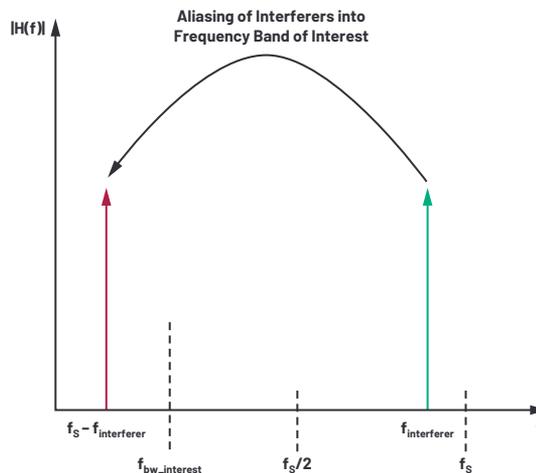


Figure 6. The aliasing/foldback of out-of-band interferers into the frequency band of interest because of sampling.

**The CTSD ADC advantage:** This design step can be completely skipped by using a CTSD ADC as it provides a new, easy option for driving a resistive load that doesn't require such a high bandwidth, large slew rate buffer. The reference IC with low-pass filter can directly be interfaced to the reference pin.

### Step 4: Making a Signal Chain Immune to Interference

Sampling and digitizing a continuous signal causes loss of information, which is termed as quantization noise. The sampling frequency and number of bits set the performance limit for an ADC architecture. After addressing the performance and interfacing challenges for the reference and input, the next struggle is to address the issue of high frequency (HF) interferers/noise folding into the low frequency bandwidth of interest. This is termed aliasing or folding back. These reflected images of the HF or out-of-band interferers into the bandwidth of interest cause signal-to-noise ratio (SNR) degradation. Citing the sampling theorem, any tone around the sampling frequency folds back in-band, as illustrated in Figure 6, which causes an unwanted information or error in the frequency band of interest. Further details on aliasing can be found in the tutorial [MT-002: What the Nyquist Criterion Means to Your Sampled Data System Design](#).

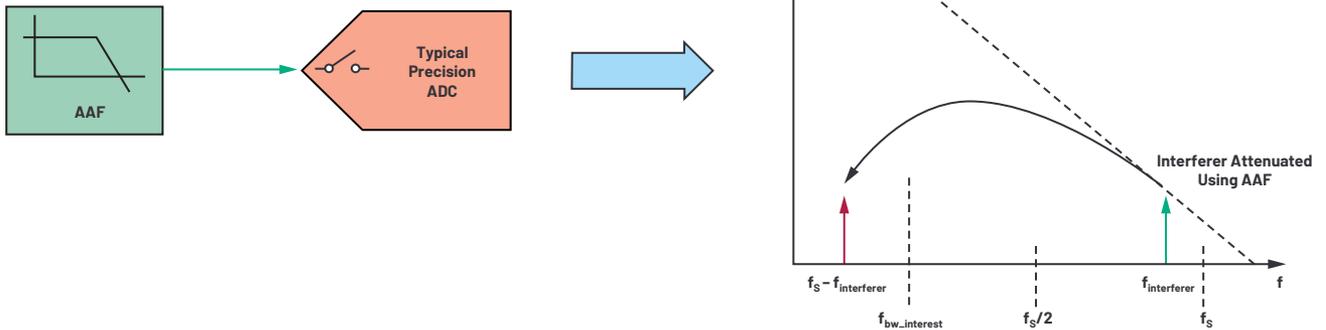


Figure 7. Use of antialiasing filter to mitigate the effect of aliasing on in-band performance.

One solution to mitigate the effect of foldback is to use a type of low-pass filter known as an antialiasing filter (AAF) to attenuate the unwanted interferer's magnitude such that when this attenuated interferer folds back in-band, the desired SNR is maintained. This low-pass filter is generally incorporated with a driver amplifier, as shown in Figure 7.

When designing this amplifier, the biggest challenge is finding a balance between faster settling and the low-pass filtering requirements. An added challenge is that this solution needs to be fine tuned for each application requirement, which limits the adoption of single platform design across various applications. ADI has many [antialiasing filter tool designs](#) to help designers overcome this challenge.

**The CTSD ADC advantage:** This immunity to interference is addressed by the inherent alias rejection property of the CTSD ADC itself, a feature that is unique only to CTSD ADCs. The AAF is not required for ADCs with this technology. So, we would be one step nearer to directly interfacing a CTSD ADC to a sensor without much effort.

### Step 5: Selecting the ADC Clock Frequency and the Output Data Rate

Next, let's discuss the clock requirements for the two classes of traditional ADCs we have discussed. The DTSD is an oversampled ADC, which means that the ADC is sampled at a higher than Nyquist sampling rate. But giving ADC oversampled data directly to the external digital controller implies we are overloading it with a lot of redundant information. In an oversampled system, the core ADC output is decimated using on-chip digital filters that enable the final ADC digital output at a lower data rate, which is usually twice the signal frequency.

For DTSD ADCs, the designer needs to plan for the provision of the high frequency sampling clock for the core ADC and program the desired output data rate. The ADC will give a final digital output at this desired ODR and the ODR clock. A digital controller uses this ODR clock to clock in the data.

Next, we address the clock requirements of SAR ADCs, which usually follow the Nyquist theorem. Here, the sampling clock of the ADC is provided by a digital controller and the clock also acts as an ODR. But there is less flexibility in the timing of this clock as the sample-and-hold timing needs to be well controlled to get optimum performance from the ADC, which also indicates that the timing of the digital output needs to be well aligned with these requirements.

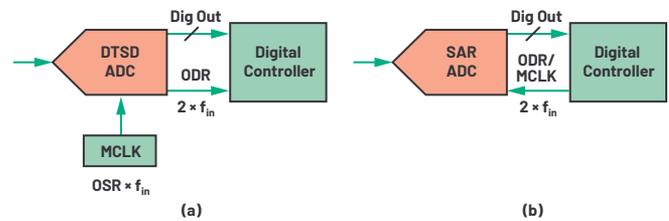


Figure 8. Clocking requirements in (a) a DTSD ADC and (b) a SAR ADC.

In understanding the clock requirements of both these architectures, we see that the ODR is coupled to the sampling clock of the ADC, which is a limitation in many systems where ODR can drift or change dynamically or needs to be tuned to the analog input signal frequency.

**The CTSD ADC advantage:** The CTSD ADC couples with a novel asynchronous sample rate converter (ASRC) that resamples the core ADC data at any desired ODR. The ASRC also enables designers to granularly set the ODR at any frequency and go beyond the age-old restriction of limiting ODR to a multiple of sampling frequency. The frequency and timing requirements of ODR are now purely a function of the digital interface and completely decoupled from the ADC sampling frequency. This feature eases digital isolation design for signal chain designers.

## Step 6: Interfacing with the External Digital Controller

Traditionally, there are two types of data interface modes for ADCs to communicate with the digital controller. One involves the ADC acting as a host, providing the digital/ODR clock, and deciding on the clock's edges for the digital controller to clock-in the ADC data. The other type is hosted mode (receiver mode), in which the digital controller is the host, provides the ODR clock, and decides the clock edges at which the ADC data will be clocked in.

Continuing from Step 5, if a designer selects a DTSD ADC, the ADC acts as host for the following digital controller since the ADC provides the ODR clock. If a SAR ADC is selected, the digital controller needs to provide the ODR clock implying SAR ADCs are always configured as hosted peripheral. So, the obvious limitation is that, once an ADC architecture is chosen, the digital interface is restricted to being in host mode or hosted mode. Currently, there is no flexibility in choosing the interface regardless of ADC architecture.

**The CTSD ADC advantage:** The novel ASRC that has been coupled with a CTSD ADC enables designers to independently configure the ADC data interface mode. This opens up a whole new opportunity for applications where high performing ADCs can be configured in any mode suitable for the digital controller of the application irrespective of ADC architecture.

## Putting It All Together

Figure 9 shows the building blocks of a traditional signal chain with an analog front end (AFE) comprising an ADC input driver, an alias rejection filter, and a reference buffer that can be drastically simplified by a CTSD ADC. Figure 10a illustrates an example signal chain with a DTSD ADC that requires significant design effort to fine tune and derive the data sheet performance of the ADC. To ease the customer journey, ADI has [reference designs](#) that can be reused or retweaked for various applications for these ADCs.

Figure 10b shows a signal chain with a CTSD ADC with its simplified analog input front end (AFE) because its ADC core does not have a switch capacitor sampler at the input and reference. The switch sampler is moved to a later stage of the ADC core, making the signal input and reference input purely resistive. This results in an almost nonsampling ADC, making a class of its own. Also, the signal transfer function of this class of ADCs mimics the antialiasing filter response, which means it inherently attenuates noise interferers. With CTSD technology, the ADC is reduced to an easy plug and play component.

In summary, CTSD ADCs simplify signal chain design while achieving a system solution with the same performance level as a traditional ADC signal chain, along with offering the following advantages:

- ▶ Provides alias free, low latency signal chain with excellent channel-to-channel phase matching
- ▶ Simplifies the analog front end with no added step of selection and fine tuning of high bandwidth input and reference driver buffers, enabling higher channel density
- ▶ Breaks barrier of ODR being a function of the sampling clock
- ▶ Gives independent control of interface to external digital controller
- ▶ Improves the signal chain reliability rating, which is a direct result of periphery component reduction
- ▶ Reduces size and has a 68% reduction in BOM, leading to faster time to market for customers

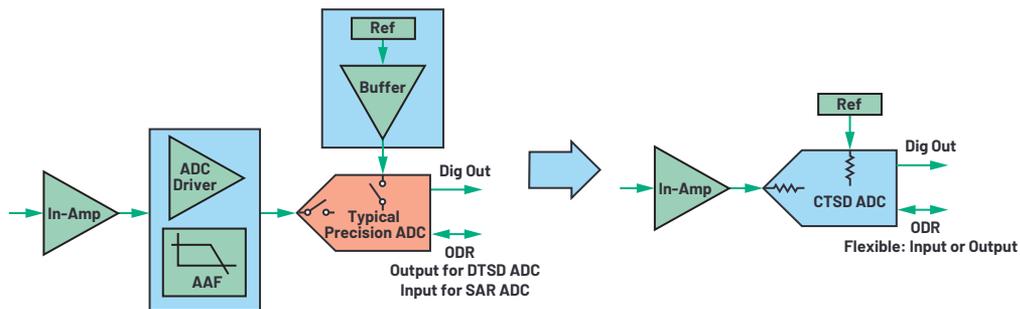


Figure 9. The building blocks of a signal chain with a traditional precision ADC vs. a CTSD ADC.

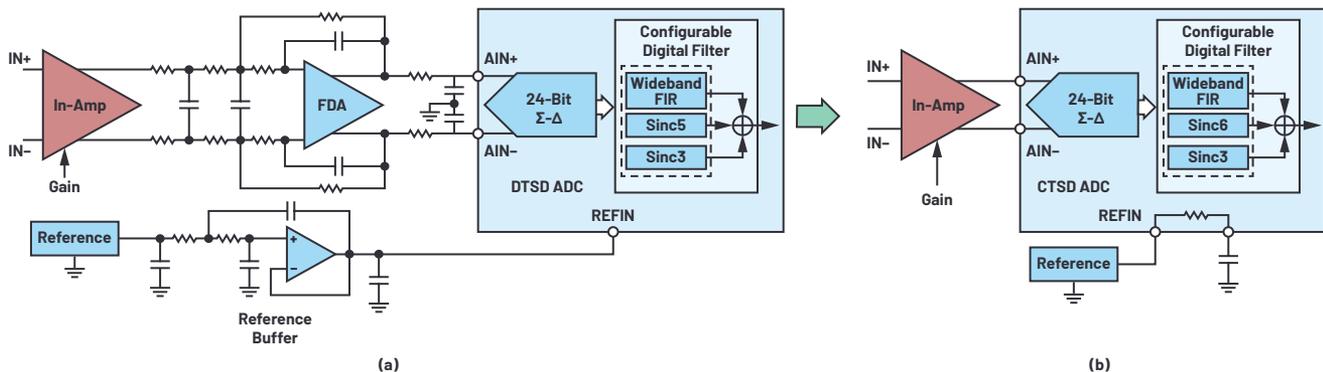


Figure 10. An example signal chain using (a) DTSD technology vs. (b) CTSD technology.

# Chapter 2: CTSD Architecture Explained for Signal Chain Designers

Abhilasha Kawle, Analog Design Manager

The traditional approach of explaining the concept of CTSD technology is by first understanding the basics of a discrete-time sigma-delta (DTSD) modulator loop and then substituting the discrete-time loop elements with equivalent continuous-time elements. While this method gives an in-depth understanding of sigma-delta functionality, we aim to provide a more intuitive understanding behind the inherent advantages of precision CTSD ADCs. To begin, we will outline a step-by-step approach to building a CTSD modulator loop starting with the widely known closed-loop inverting amplifier configuration and combining it with an ADC and a DAC. Finally, we will evaluate the basic sigma-delta functionality from the circuit we build.

## Step 1: Revisiting the Closed-Loop Inverting Amplifier Configuration

One of the key advantages of the CTSD ADC is that it offers an easy to drive continuous resistive input rather than a traditional switched capacitor sampler upfront. One of the circuits that has a similar input impedance concept is the inverting amplifier, which we will use as a starting block toward building a CTSD modulator loop.

A closed-loop op amp configuration has always been the go-to option for replicating an analog input with high fidelity, and Figure 1 shows one of the most popular op amp configurations, which is called an inverting amplifier configuration.<sup>1</sup> One of the measures of the fidelity is the output to input gain, also known as, in sigma-delta nomenclature, the signal transfer function (STF). Determining the parameters that affect the STF requires analyzing the circuit.

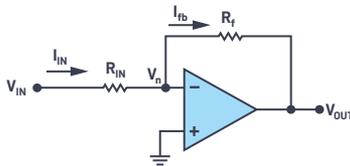


Figure 1. A closed-loop op amp in inverting amplifier configuration.

To refresh our mathematical skills, let's revisit the derivation of famous  $V_{OUT}/V_{IN}$ . In the first step, the open-loop gain of the op amp A is assumed to be infinite. This assumption directly leads to making negative input of op amp,  $V_n$ , at potential ground. The application of Kirchhoff's laws at this node gives

$$I_{IN} = \frac{V_{IN}}{R_{IN}}, I_{fb} = -\frac{V_{OUT}}{R_f} \quad (1)$$

Mapping this to  $V_{OUT}$  and  $V_{IN}$ , we get the gain or STF as shown in Equation 2:

$$STF = \frac{V_{OUT}}{V_{IN}} = -\frac{R_f}{R_{IN}} \quad (2)$$

Next let's go beyond the impractical assumption of infinite gain and rederive the STF with the finite gain of A for the op amp. The STF now looks like

$$STF = -\frac{R_f}{R_{IN}} \times \left( \frac{A}{\left(1 + \frac{R_f}{R_{IN}}\right) + A} \right) \quad (3)$$

From here, textbooks generally describe the sensitivity toward each of the parameters  $R_{IN}$ ,  $R_f$ , and A. For our case, let's proceed toward building the CTSD loop.

## Step 2: Introducing Discretization into the Amplifier

The requirement for our ADC signal chain is a digitized version of  $V_{IN}$ . In our next step, we introduce the digitization in this circuit. Rather than use the traditional way of putting a sampling ADC directly at the input signal, we will try a different approach and put a representative ADC that follows the amplifier output to get the digitized data. But the output of the ADC cannot be used as feedback directly, as it is required to be an analog voltage. So then, we need to follow up the ADC with a voltage digital-to-analog converter (DAC) as shown in Figure 2.

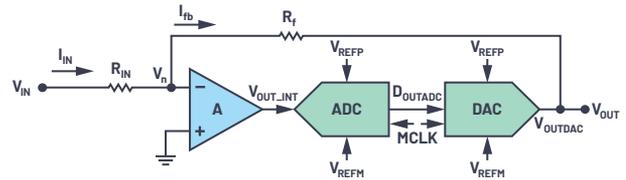


Figure 2. Introducing an ADC and DAC in an inverting amplifier configuration.

Because of the ADC and DAC,  $V_{OUT}$  is still a representation of  $V_{IN}$  but with quantization error due to the digitization added. So, nothing has changed in the signal flow from  $V_{IN}$  to  $V_{OUT}$ . One point of note here, to keep the functionality of loop symmetric about 0 V and ease our mathematical derivation, the references of ADC and DAC are chosen to be

$$V_{REFP} = V_{REF}/2, \text{ and } V_{REFM} = -V_{REF}/2 \quad (4)$$

## Step 3: Introducing the Analog Accumulator-The Integrator

Is the closed-loop configuration in Figure 2 stable? Both the ADC and DAC are discretization elements working on a sampling clock, MCLK. It has been an unachievable dream of converter specialists to design a delay free ADC or DAC. Since these loop elements are clocked, the input is generally sampled on one edge and processed on the other clock edge. So, the output of the ADC and DAC combination  $V_{OUT}$ , which is the feedback in Figure 2, is available only after 1 clock cycle delay.

Does this delay in feedback have any implication on stability? Let's trace how  $V_{IN}$  transfers along. For simplification let's assume  $V_{IN} = 1$ ,  $R_{IN} = 1$ ,  $R_f = 1$ , and the gain of op amp A is 100. At the first clock cycle, the input voltage is 1 and the DAC output feedback,  $V_{OUT}$  or  $V_{OUTDAC}$ , is 0 and is not available until the next clock edge. As we trace the error between the input and feedback to the output of the amplifier and ADC, we can see the output keeps growing exponentially and this is technically termed as the runaway problem.

**Table 1. Clock Edge Samples**

	$V_{IN}$	$V_{OUT} = V_{OUTDAC}$	$V_n = (V_{OUT} + V_{IN})/2$	$V_{OUT\_INT} = -A \times (V_n)$	$D_{OUTDAC}$
First Sample Edge	1	0	0.5	-50	-50
Second Sample Edge	1	-50	~ -25	~2500	2500
Third Sample Edge	1	2500	~1250	~-12,500	-12,500

This happened because the ADC input works on instantaneous error gained up by the amplifier; that is, the ADC decides even before the feedback is available, which was not required. If the ADC works on an accumulated, averaged error data so that the error due to 1 clock delay of feedback is averaged out, then the output of the system would be bounded.

The integrator is one such analog equivalent of an averaging accumulator. The gain of the loop is still high but only at low frequencies or, in other words, in the frequency bandwidth of interest. This ensures that the ADC is not presented with any instantaneous errors that can lead to a runaway situation. So, the loop is now amplifier modified as an integrator followed by the ADC and DAC, as shown in Figure 3a.

### Step 4: Simplifying the Feedback Resistor

Our element of interest is  $D_{OUTDAC}$ , so let's rearrange the loop elements to highlight  $D_{OUTDAC}$  as the output of the system, as shown in Figure 3b. Next, let's visit the simplification of the DAC and  $R_f$  path. And for that let's dig into the DAC's details. The purpose of the DAC is to convert a digital code,  $D_{IN}$ , to an equivalent analog current or voltage in proportion to the reference. To further extend the advantages of continuity to reference, what we have considered here is a general DAC architecture based on a resistor ladder that has no switching load on reference. Let's review a thermometric resistor DAC,<sup>2</sup> which converts  $D_{IN}$  to the DAC current, with relation to Equation 5.

$$I_{DAC} = \frac{V_{REF}}{R_f} \times \frac{D_{IN}}{2^N} \tag{5}$$

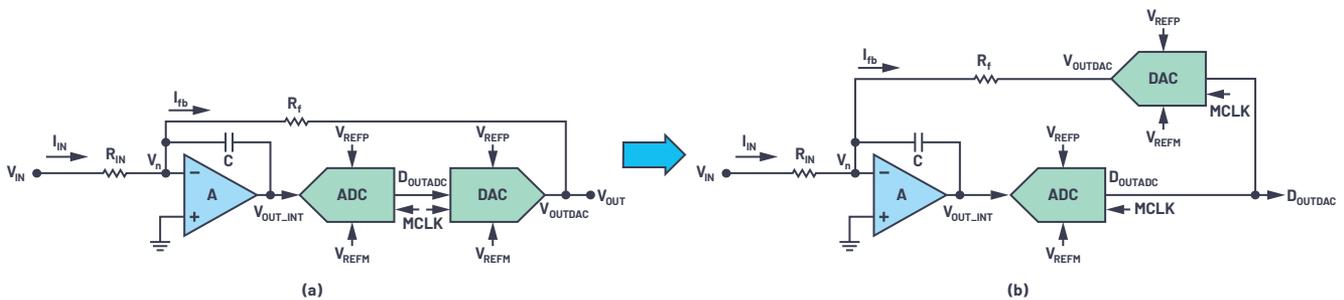


Figure 3. (a) Introducing the integrator into the loop. (b) Rearranging the loop to highlight  $D_{OUTDAC}$  as output.

Where  $V_{REF} = V_{REFP} - V_{REFM}$ , the total reference voltage across the DAC.

- ▶  $D_{IN}$  = Digital input in the thermometric code
- ▶  $R_f$  = Feedback resistor; split as each unit element
- ▶  $N$  = Number of bits

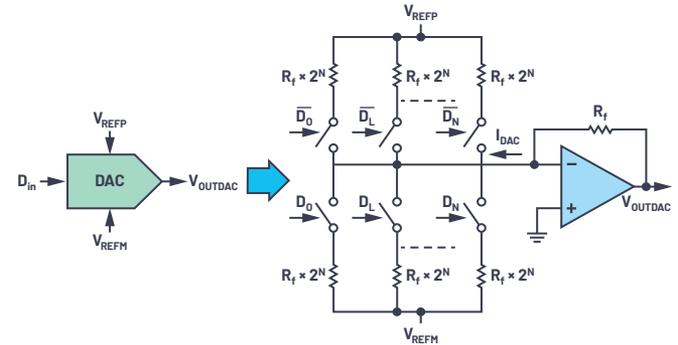


Figure 4. A general thermometric resistive DAC.

To get the voltage output, an  $I$  to  $V$  conversion follows by using an op amp in a transimpedance configuration,<sup>3</sup> as shown in Figure 4. So,

$$V_{OUTDAC} = I_{DAC} \times R_f \tag{6}$$

Going back to our discretized loop of Figure 3b, this  $V_{OUTDAC}$  is again converted back to current,  $I_{fb}$ , through the feedback resistor of the inverting amplifier, implying the signal flow is  $I_{DAC} \rightarrow V_{OUTDAC} \rightarrow I_{fb}$ . Mathematically,

$$I_{fb} = \frac{V_{OUTDAC}}{R_f} = I_{DAC} \tag{7}$$

From the above signal flow and formula, we see that converting  $V_{OUTDAC}$  to  $I_{fb}$  is a redundant step that can be bypassed. Removing the redundant elements and, for simplicity, representing  $(V_{REFP} - V_{REFM})$  as  $V_{REF}$ , let's redraw our loop, as shown in Figure 5.

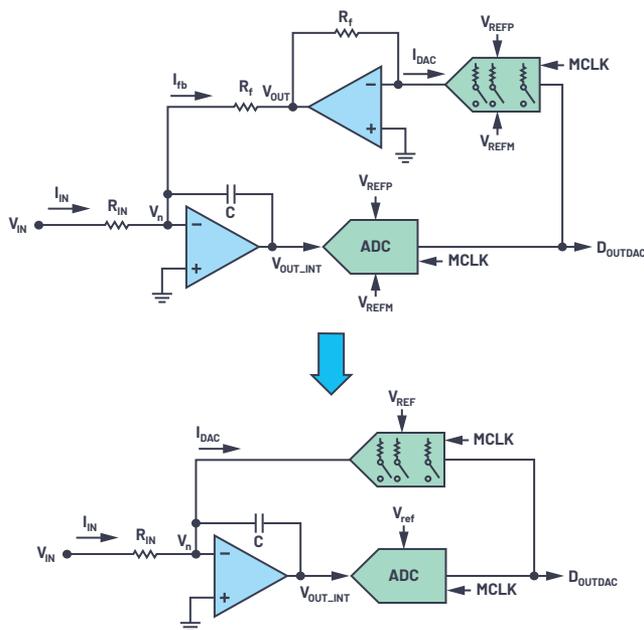


Figure 5. Removing redundant I to V and the feedback resistor.

And voila! We have built a first-order sigma-delta loop! And all by stitching together well-known elements—an inverting amplifier, an ADC, and a DAC.

### Step 5: Understanding Oversampling

We have up to now grasped the construction of a CTSD loop, but we have yet to appreciate the particularities offered by this fanciful loop. The first step toward that is understanding oversampling. ADC data is useful only if there are enough sampled and digitized data points to extract or interpret the analog signal information. The Nyquist theorem advises that, for faithful reconstruction of an input signal, the sampling frequency of the ADC should be at least twice the frequency of the signal. If we keep adding more data points over this minimum requirement, the error in interpretation would be further reduced. Following this line of thought, in sigma-delta the sampling frequency is selected to be much higher than the suggested Nyquist frequency and this is known as oversampling. Oversampling<sup>4</sup> helps reduce the quantization noise in the frequency band of interest by spreading the total noise over much higher frequency, as shown in Figure 6.

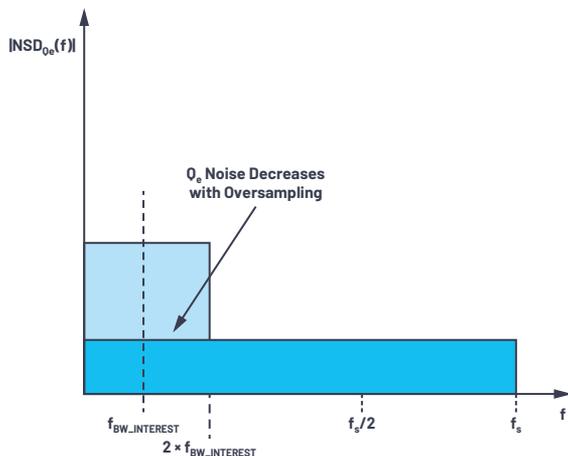


Figure 6. A noise spectral density comparison between Nyquist sampling and oversampling.

### Step 6: Understanding Noise Shaping

Signal chain designers shouldn't feel lost when sigma-delta experts use terms like noise transfer function (NTF) or noise shaping,<sup>4</sup> and our next step will help them get an intuitive understanding of these terms as they are unique to sigma-delta converter nomenclature. Let's revisit our simple inverting amplifier configuration and introduce the error  $Q_e$  at the output of the amplifier, as shown in Figure 7.

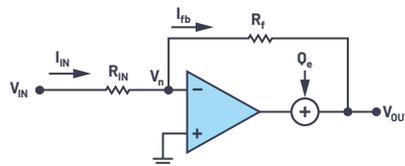


Figure 7. The introduction of an error in an inverting amplifier configuration.

The contribution of this error at the output is quantified as

$$V_{OUT} = \frac{Q_e}{\left(1 + \frac{A}{\left(1 + \frac{R_f}{R_{IN}}\right)}\right)} \quad (8)$$

The mathematical formula translates that the error  $Q_e$  is attenuated by the open-loop gain of the amplifier, which is just reiterating the advantage of a closed loop.

This understanding of the closed-loop advantage can be extended to quantization error  $Q_e$  of the ADC in CTSD loop, which is the error introduced due to digitization of the continuous signal at the output of the integrator, as shown in Figure 8.

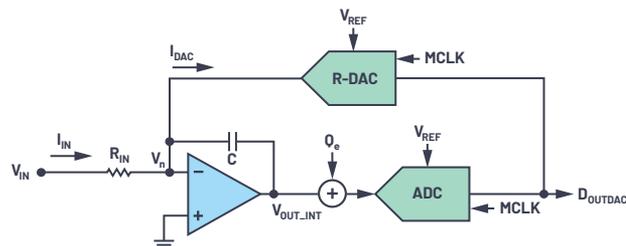


Figure 8. The introduction of quantization error  $Q_e$  in a sigma-delta loop.

We can now intuitively conclude that this  $Q_e$  would be attenuated by the integrator. The integrator TF is  $|H_{INTEG}(f)| = 1/|s \times RC| = 1/2\pi fRC$  and its corresponding frequency domain representation is shown in Figure 9. Its profile is equivalent to a low-pass filter profile with high gain at low frequencies, and the gain reduces linearly as frequency increases. Correspondingly, the attenuation for  $Q_e$  would then look like a high-pass filter.

The mathematical representation of this attenuation factor is the noise transfer function. For an interim, let's ignore the sampler in the ADC and the switches in the DAC. The NTF,  $V_{OUTDAC}/Q_e$ , can be evaluated by following the same exercise as we did for the inverting amplifier configuration, which in the frequency domain looks like a high-pass filter profile, as shown in Figure 10.

$$NTF_{int} = \frac{V_{OUTDAC}}{Q_e} = \frac{sR_fC}{(1 + sR_fC)} \quad (9)$$

In the frequency band of interest, the quantization noise is completely attenuated and pushed to "not to our concern" high frequencies. This is what is called noise shaping.

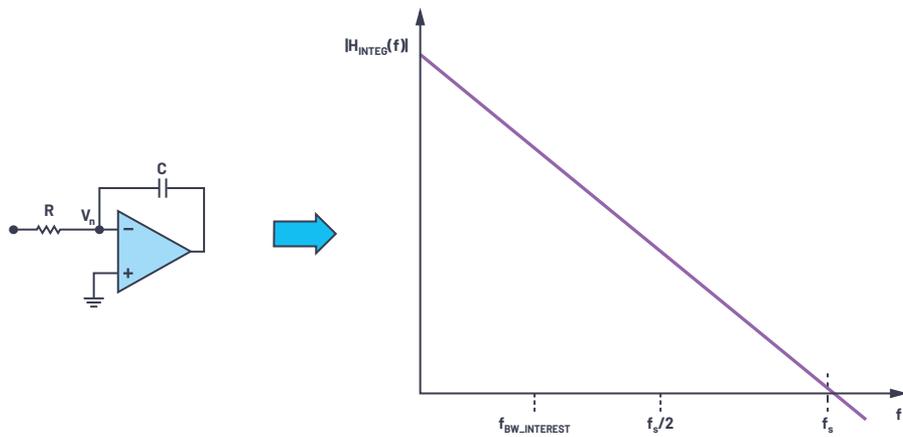


Figure 9. Integrator transfer function.

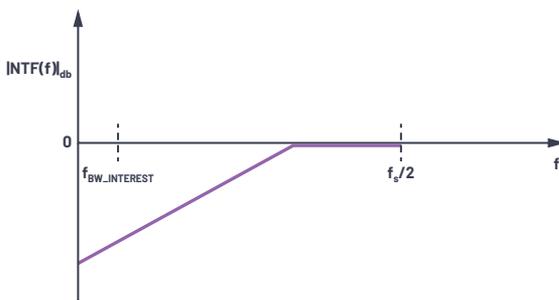


Figure 10. Noise transfer function without the sampler—has a high-pass filter profile.

With the sampler in loop, the quantization noise shaping analogy remains the same. The difference being the NTF frequency response would have replicated images at every multiple of  $f_s$ , as shown in Figure 10, thus creating notches at every integer multiple of the sampling frequency.

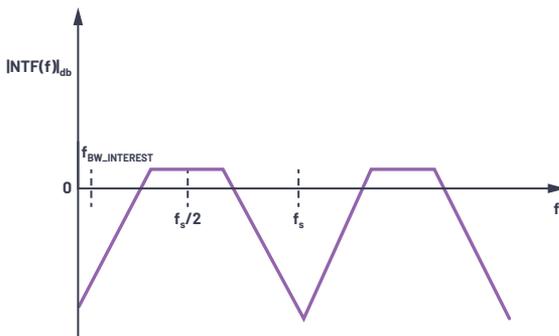


Figure 11. The noise transfer function of a CTSD ADC.

The uniqueness of sigma-delta architecture lies in the fact that putting an integrator and a DAC loop around a crude ADC—for example, a 4-bit ADC—and applying the concept of oversampling and noise shaping reduces the quantization noise significantly in the frequency bandwidth of interest and masks this crude ADC to a 16- to 24-bit precision ADC.

These basics of the first-order CTSD ADC can now be extended to any order of modulator loop. The sampling frequency, the crude ADC specifications, and the order of loop are top-level design decisions driven by the performance requirements of the ADC.

## Step 7: Completing the CTSD Modulator with a Digital Filter

Generally, in an ADC signal chain, the digitized data is postprocessed by an external digital controller for any signal information extraction. In sigma-delta architecture, as we know now, the signal is oversampled. If this oversampled digital data is directly given to the external controller, then there is a lot of redundant data that needs to be processed. This causes excess power and real estate cost overheads in the digital controller design. So, before data is presented to the digital controller, the data samples are dropped in an efficient way without affecting the performance. This process is called decimation and is done by digital decimation filters. Figure 11 shows a typical CTSD modulator with on-chip digital decimation filters.

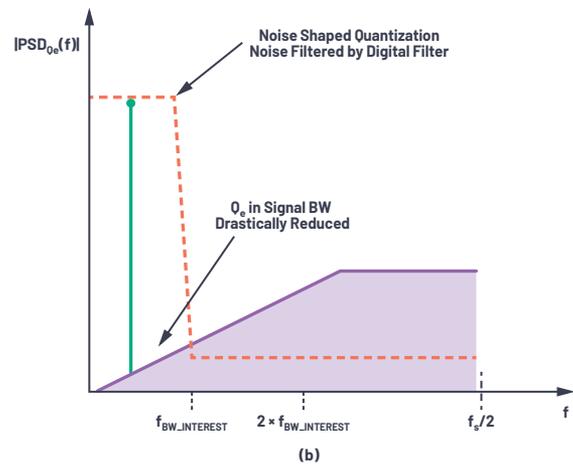
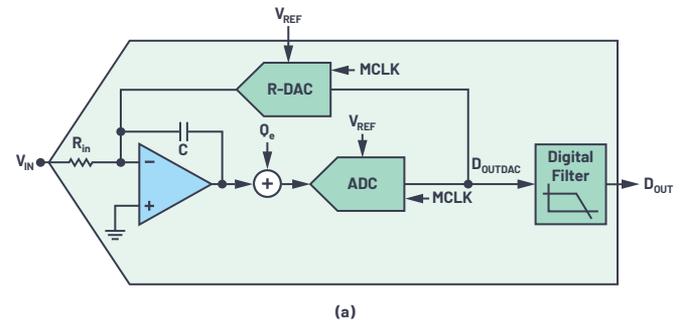


Figure 12. (a) A block diagram of a CTSD ADC modulator loop from an analog input to a digital

output. (b) A frequency spectrum representation of an input signal at the output of a modulator and the output of a digital filter.

Figure 12b shows frequency response for an in-band analog input signal. At the output of the modulator we observe the noise shaping of the quantization noise, drastically reducing it in the frequency band of interest. The digital filter helps attenuating the shaped noise beyond this frequency bandwidth of interest so that the final digital output,  $D_{OUT}$ , is at the Nyquist sampling rate.

## Step 8: Understanding the Clock Sensitivity of CTSD ADCs

So far, we have understood how CTSD ADCs keep the continuous integrity of the input signal, which significantly simplifies signal chain design. There are also a few limitations with this architecture, mainly dealing with the sampling clock, MCLK. The CTSD modulator loop works on the concept of integrating the error current between  $I_{IN}$  and  $I_{DAC}$ . Any error in this integrated value would cause the ADC in loop to sample the error and would reflect this in the output. For our first-order integrator loop, the integrated value over the sampling time period of  $T_s$  for constant  $I_{IN}$  and  $I_{DAC}$  is given by

$$\delta V_{out_{integ}} = \frac{T_s}{RC} \times (I_{IN} - I_{DAC}) \quad (10)$$

For an input of 0, the parameters that would affect this integration error are

- ▶ **MCLK frequency:** As indicated by Equation 10, if the MCLK frequency scales, then the RC coefficient that controls the slope of integration also needs to be returned to get back the same integrated value. This implies that a CTSD modulator is tuned for a fixed MCLK clock frequency and cannot support varying MCLK.
- ▶ **MCLK jitter:** The DAC code and, hence,  $I_{DAC}$  change every clock time period  $T_s$ . If the  $I_{DAC}$  time period randomly changes, then the average integrated value keeps changing, as shown in Figure 13. So, any error in the sampling clock time period in the form of jitter would affect the performance of the modulator loop.

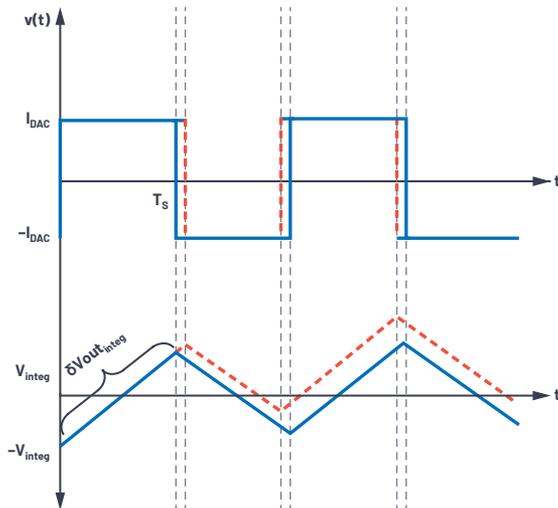


Figure 13. The clock sensitivity of a CTSD modulator.

CTSD ADCs are sensitive to the frequency and jitter of an MCLK because of the above reasons.<sup>5</sup> But ADI has identified solutions to work around these fallacies. For example, the challenges of generating and routing accurate, low jitter MCLK along the system to the ADC can be addressed using a local, low cost crystal and oscillator near the ADC. The fallacy around the fixed sampling frequency has been addressed by using innovative asynchronous sample rate conversion (ASRC) that enables a variable and independent digital output data rate for the digital controller irrespective of the fixed sampling MCLK. More information about this will be detailed later in this minitutorial.

## Step 9: Voila! All Set to Explain the CTSD Concept to Your Buddies!

The input impedance of a CTSD ADC is equivalent to the input impedance of the inverting amplifier, which is resistive and easy to drive. Using innovative techniques, the reference used by the modulator loop's DAC has also been made resistive. The sampler of the ADC is after the integrator and not directly at the input, which enables inherent alias rejection for interferers outside the frequency band of interest. We will deep dive into each of these advantages and their corresponding impact in a signal chain in the next few chapters of this minitutorial.

# Chapter 3: Inherent Alias Rejection Made Possible

Abhilasha Kawle, Analog Design Manager, and Smita Choudhury, Design Evaluation Manager

In many applications like sonar arrays, accelerometers, vibration analysis, etc., signals outside the signal bandwidth of interest are observed that are termed as interferers. The key challenge for signal chain designers is that the ADC sampling phenomenon causes these interferers to alias into the signal bandwidth of interest (in-band) and degrade the performance. Apart from this, in applications like sonar, the interferers aliasing in-band could be misinterpreted as an input signal, causing misdetection of objects around the sonars. The solutions to reject these aliases are one of the reasons why traditional ADC signal chain designs are quite complex. The unique inherent alias rejection property of CTSD ADCs provides a new simplified solution. Before arriving at this groundbreaking solution, our first stop for this article is at understanding the concept of aliasing.

## Revisiting the Nyquist Sampling Theorem

To understand the concept of aliasing, let's have a quick recap of the Nyquist sampling theorem. One could analyze a signal in either the time domain or frequency domain. In the time domain, the sampling of an analog signal is represented mathematically as multiplication of the signal—for example,  $x(t)$  with an impulse train,  $\delta(t)$ , having time period  $T_s$ .

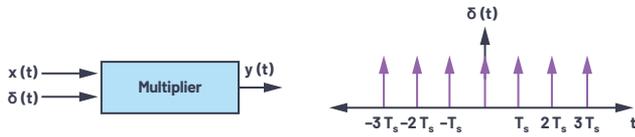


Figure 1. The time domain representation of a sampling process.

Equivalently in the frequency domain, the sampled output can be expressed using a Fourier series as,

$$Y(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s) \text{ where, } f_s = \frac{1}{T_s}, n = 0, \pm 1, \pm 2, \dots (1)$$

Equation 1 simply means that if the frequency axis is unfurled, images of the input signal are formed at every integer multiple of sampling frequency,  $f_s$ .

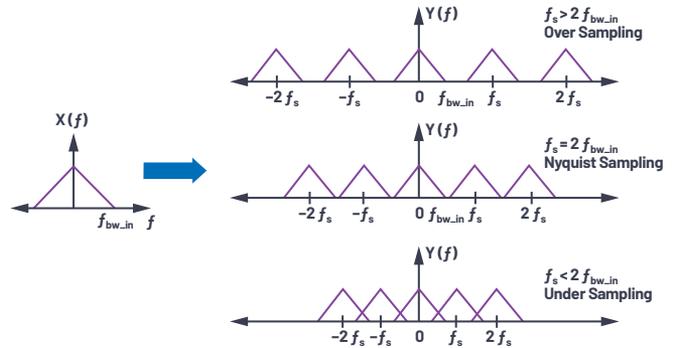


Figure 2. A representation of  $X(f)$  after being sampled by different sampling frequencies.

Equation 1 indicates that the signal content of  $X(f)$  at frequencies  $f = n \times f_s - f_{in}$ , where  $n = 0, \pm 1, \pm 2, \dots$ , will manifest itself at  $f_{in}$  after sampling, similar to the under-sampling scenario in Figure 2, which illustrates the sampling phenomenon under various conditions.

$$\begin{aligned} Y(f) &= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s) \\ &= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f_{in}) \forall f = n \times f_s - f_{in} \end{aligned} \quad (2)$$

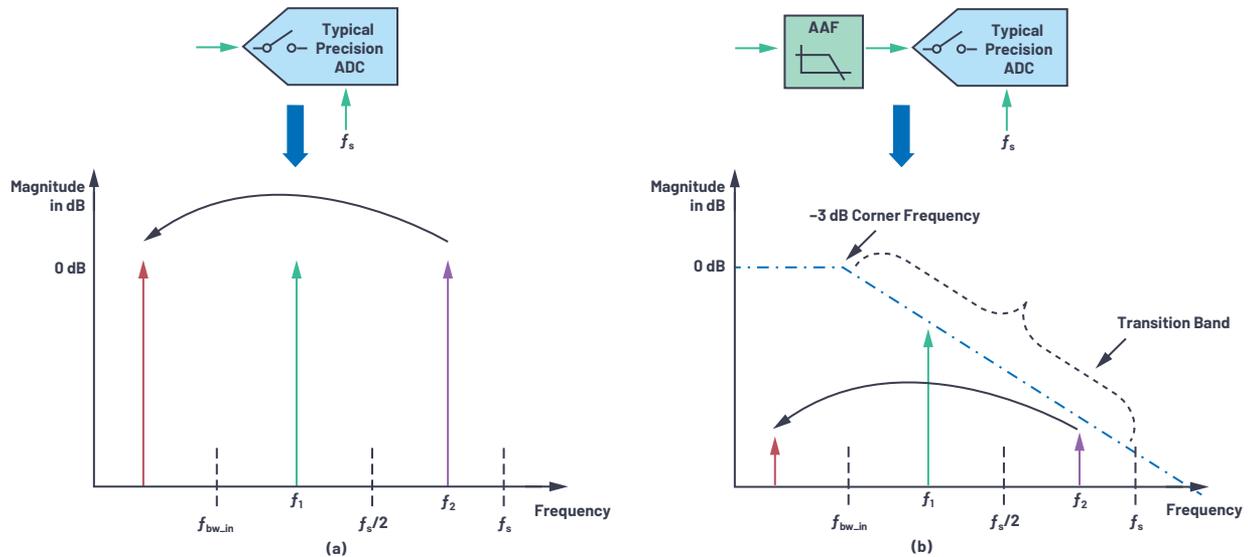


Figure 3. (a) Applying the sampling theorem to understand aliasing and (b) using the antialiasing filter to attenuate the aliasing frequencies.

In summary, the Nyquist theorem states that any signal greater than half the sampling frequency, folds or mirror backs to frequency less than  $f_s/2$  and can potentially fall into the frequency band of interest.

Assume an ADC is sampling at frequency  $f_s$  and there are two out-of-band tones/interferers in the system,  $f_1$  and  $f_2$  at the ADC input as shown in Figure 3. Applying the Nyquist theorem, we can infer that since the frequency of tone  $f_1$  is less than  $f_s/2$ , after sampling, its frequency remains the same. While the frequency of tone  $f_2$  is greater than  $f_s/2$ , it will alias itself in the frequency band of interest,  $f_{bw\_in}$  and degrade the performance of the ADC in this region, as shown in Figure 3a.

This theory can also be extended to any noise beyond  $f_s/2$ , which also folds back and manifests itself in-band to increase the in-band noise floor and degrade performance.

## An Incumbent Solution for Aliasing

A simple solution to avoid this performance degradation due to out-of-band (OOB) tone or noise foldback is to attenuate any signal content beyond  $f_s/2$  before being sampled by the ADC using a low-pass filter, which is known as an antialiasing filter (AAF). Figure 3b shows the transfer function of a simple AAF and illustrates the attenuation-to-alias tone at frequency  $f_2$  before it folds back in-band. The main characteristics of this AAF would be the order of the filter and -3 dB corner frequency. They are determined by pass-band flatness, the absolute attenuation required at certain frequencies (like sampling frequency) and the slope of attenuation required beyond input bandwidth (also called transition band). A few common filter architectures are Butterworth, Chebhevsev, Bessel, and Sallen-Key, which can be implemented using passive RC and op amps. [Filter design tools](#) are available to assist signal chain designers with AAF design for given architecture and requirements.

Let's take an example application to understand the antialiasing filter requirements. In a submarine system, the sonar sensor emits sound waves and analyzes the echoes underwater to estimate the position and distance of surrounding objects. The sensor has input bandwidth of 100 kHz and the system detects any tone of magnitude  $>-85$  dB at the ADC input as a valid source of echo. So, any interference from out-of-band would need to be attenuated by at least -85 dB by an ADC to avoid detection as input by the sonar system. For these requirements, in the next section we will build and compare the alias rejection solutions for different ADC architectures.

In traditional ADC architectures, such as successive approximation register (SAR) and discrete-time sigma-delta (DTSD) ADCs, the sampling circuit is at the analog input of the ADC, indicating that an AAF is required before the ADC input, as shown in Figure 3b.

### AAF Requirements for SAR/Nyquist Sampling ADCs

SAR ADCs generally have a sampling frequency set to two or four times the analog input frequency ( $f_{in}$ ). The AAF for such an ADC would need to have a narrow transition band beyond frequency  $f_{in}$ , implying a very high order filter is required. From Figure 4, we can see that a SAR ADC with a sampling frequency of approximately 1 MHz requires a fifth-order Butterworth filter to get -85 dB rejection for frequencies greater than 100 kHz. In terms of filter implementation, as the order of filter increases, the number of passives and op amps required increases. This means an AAF for SAR ADCs requires significant power consumption and area budget in signal chain design.

### AAF Requirements for DTSD ADCs

Sigma-delta ADCs are oversampled ADCs where sampling is much higher than the analog input frequency. And the region of aliasing to be considered for AAF design is  $f_s \pm f_{in}$ . The transition band requirement for the filter would be from  $f_{in}$  to very high  $f_s$ . This is a wider transition band in comparison with a SAR ADC AAF, showing that the order of AAF required is also lower. Figure 4 shows that, for a 6 MHz sampling frequency DTSD ADC, to get -85 dB rejection for frequencies around  $f_s - 100$  kHz, a second-order AAF is generally required.

In a practical scenario, interferers or noise could be anywhere in the frequency band and not restricted to being around sampling frequency  $f_s$ . Any frequency tone less than  $f_s/2$ , like the tone at frequency  $f_1$  in Figure 3, wouldn't manifest into in-band to degrade the ADC performance. Though the AAF may attenuate the tone  $f_1$  to a certain extent, it is still present in the ADC output and is unnecessary information that must be processed by the external digital controller. Could this tone be further attenuated so that it is not seen at the ADC output? One solution could be to use an AAF with a narrow transition band beyond frequency  $f_{in}$ , but then the filter design complexity would increase. Alternative solutions are on-chip digital filters that are part of sigma-delta modulator loops.

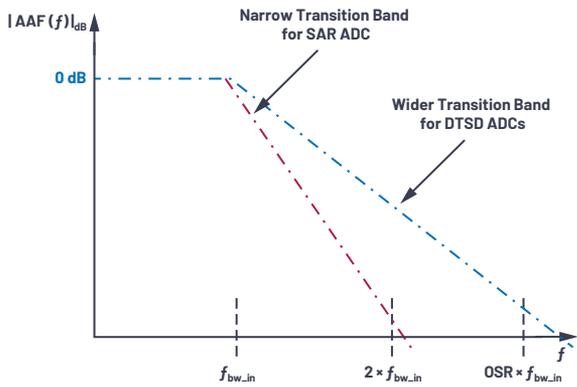


Figure 4. The complexity of an AAAF vs. an ADC architecture vs. frequency band of interest.

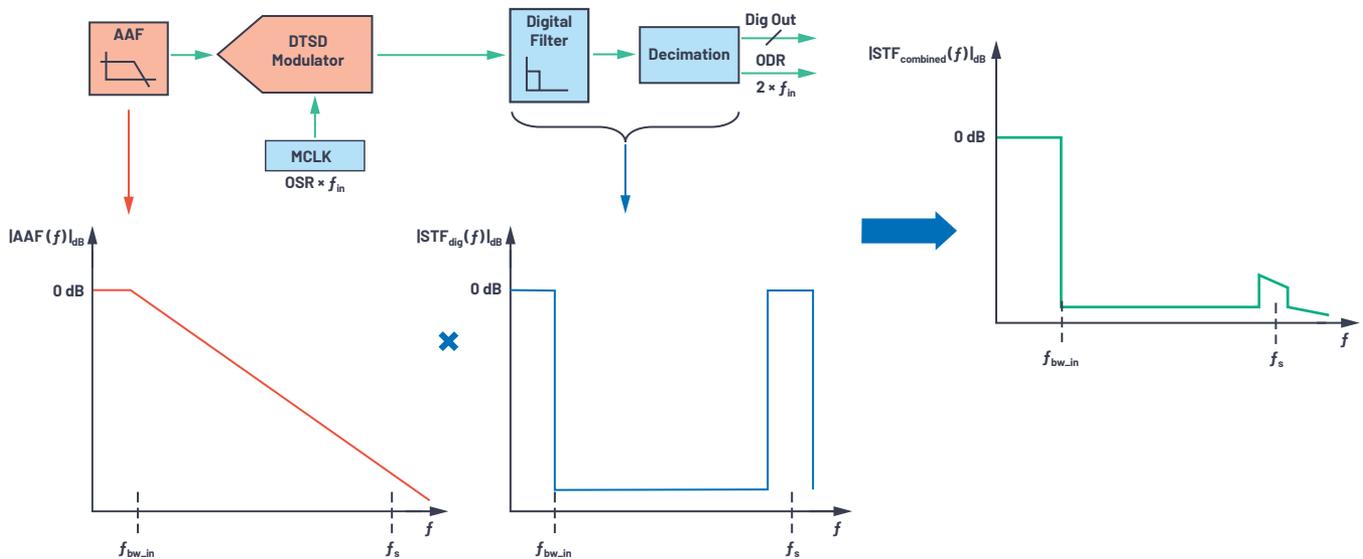
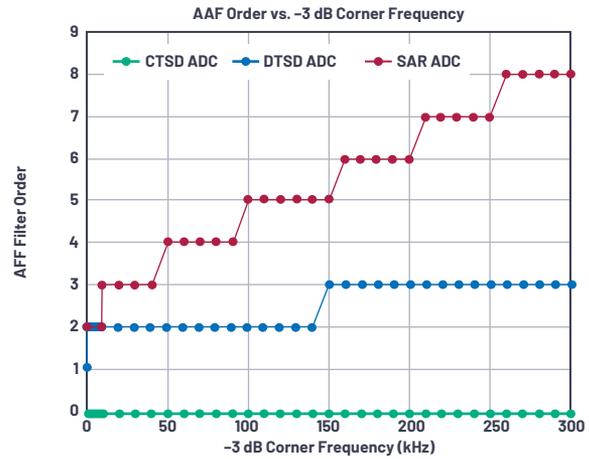


Figure 5. The STF of a DTSD ADC with an AAAF at the front end and digital filters at the back end.

### Digital Filters of Sigma-Delta Modulator Loops

In sigma-delta ADCs, because of oversampling and noise shaping, the modulator output contains a lot of redundant information and thus requires a large amount of processing by the external digital controller. This redundant information processing can be avoided if modulator data is averaged, filtered, and provided at a lower output data rate (ODR), which is generally  $2 \times f_{IN}$ . Decimation filters are used to convert the sampling rate from  $f_s$  to the required lower ODR. Sample rate conversion using a digital filter will be explained in future articles, but the key point here is that a discrete-time sigma-delta modulator is usually partnered with an on-chip digital filter. The combined signal transfer function (TF) for interferers with the analog filter in front and digital filter on the back end of a modulator is shown in Figure 5.

In conclusion, the AAAF for a DTSD ADC is designed based on the attenuation required for tones around alias region  $f_s$ . And the tones in a non-aliasing region like  $f_1$  are completely attenuated by the on-chip digital filters.

### Back-End Digital Filter vs. Front-End Analog Filter

A SAR ADC requires a narrow transition band in an AAAF, while a sigma-delta ADC requires a narrow transition band in a digital filter. Digital filters are low power and easy to integrate on-chip. Also, programming the order, bandwidth, and transition band of a digital filter is much simpler than with an analog filter.

Oversampling is advantageous in that it allows the use of a wide transition analog filter combined with a narrow transition digital filter on the back end, providing an optimized solution in terms of power, space, and immunity to interferers.

With the use of DTSD ADCs, the AAAF requirements, though relaxed, add design complexity to meet settling time requirements after every sampling event to avoid performance degradation of a signal chain. The challenge for signal chain designers is to fine-tune the AAAF to balance between alias rejection and output settling requirements.

The new class of precision CTSD ADCs simplifies the signal chain design by eliminating the need for front-end analog filter design.

### The Inherent Alias Rejection of CTSD ADCs

In chapter 2 of this minitutorial, a first-order CTSD modulator was built from a closed-loop resistive inverting amplifier, as shown in Figure 6. A CTSD modulator follows the same concept of oversampling and noise shaping as a DTSD modulator counterpart to achieve the desired performance, and has a resistive input rather than a switched capacitor input. The modulator building blocks include a continuous-time integrator, followed by a quantizer that samples and digitizes the integrator output and a feedback DAC that closes the loop at the input. Any noise at the input of a quantizer is noise shaped by the integrator's gain transfer function.

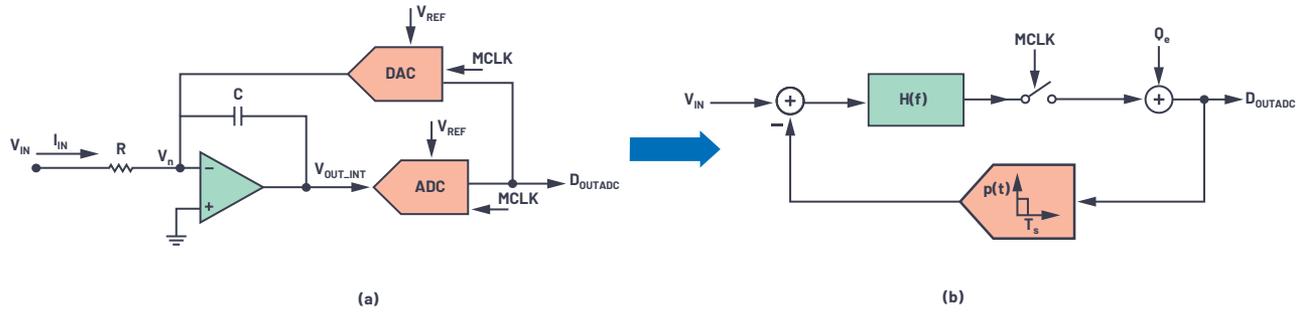


Figure 6. (a) The building blocks of a CTSD modulator loop and (b) a simplified block representation for mathematical analysis.

Expanding on the information from chapter 2, a simplified block representation for a CTSD modulator loop can be drawn with the following mathematical models:

- ▶ The integrator transfer function is generalized as  $H(f)$  and is also known as a loop filter. For a first-order integrator,  $H(f) = 1/2\pi RC$ .
- ▶ The functionality of the ADC is sampling and quantization. So, a simplified ADC model for analysis uses a sampler followed by an additive quantization noise source.
- ▶ The DAC is a block that multiplies in the input in the present clock cycle with a constant. So, it's a block with an impulse response that is constant during the sampling clock period and 0 the rest of the time.

The equivalent block diagram with these simplified models is shown in Figure 6b and is widely used for sigma-delta performance analysis. The transfer function from  $V_{in}$  to  $V_{out}$  is called signal TF (STF) and the  $Q_e$  to output is termed as noise TF (NTF).

One reasonable explanation about the inherent alias rejection property of a CTSD modulator loop would be that sampling occurs not directly at the input of the modulator but after the loop filter,  $H(f)$  as shown in Figure 6a. But to get a complete picture, a linear model without a sampler would be used to understand the concepts and the analysis would be extended to loop with the sampler.

### Step 1: STF and NTF Analysis Using a Linear Model

Ignoring the sampler for analysis simplification, the linear model would be as shown in Figure 7. The STF and NTF for this loop can be represented as

$$V_{OUTADC} = V_{IN} \times \frac{H(f)}{1 + H(f)} + Q_e \times \frac{1}{1 + H(f)} \quad (3)$$

$$= V_{IN} \times STF(f) + Q_e \times NTF(f)$$

From Equation 3, the STF can be rewritten as

$$STF(f) = H(f) \times NTF(f) \quad (4)$$

The frequency bandwidth of interest is low frequency, so mathematically it can be represented as  $f \rightarrow 0$ , while high frequency can be represented as  $f \rightarrow \infty$ . The magnitude of STF and NTF in dB as a function of frequency when plotted would be as shown in Figure 7.

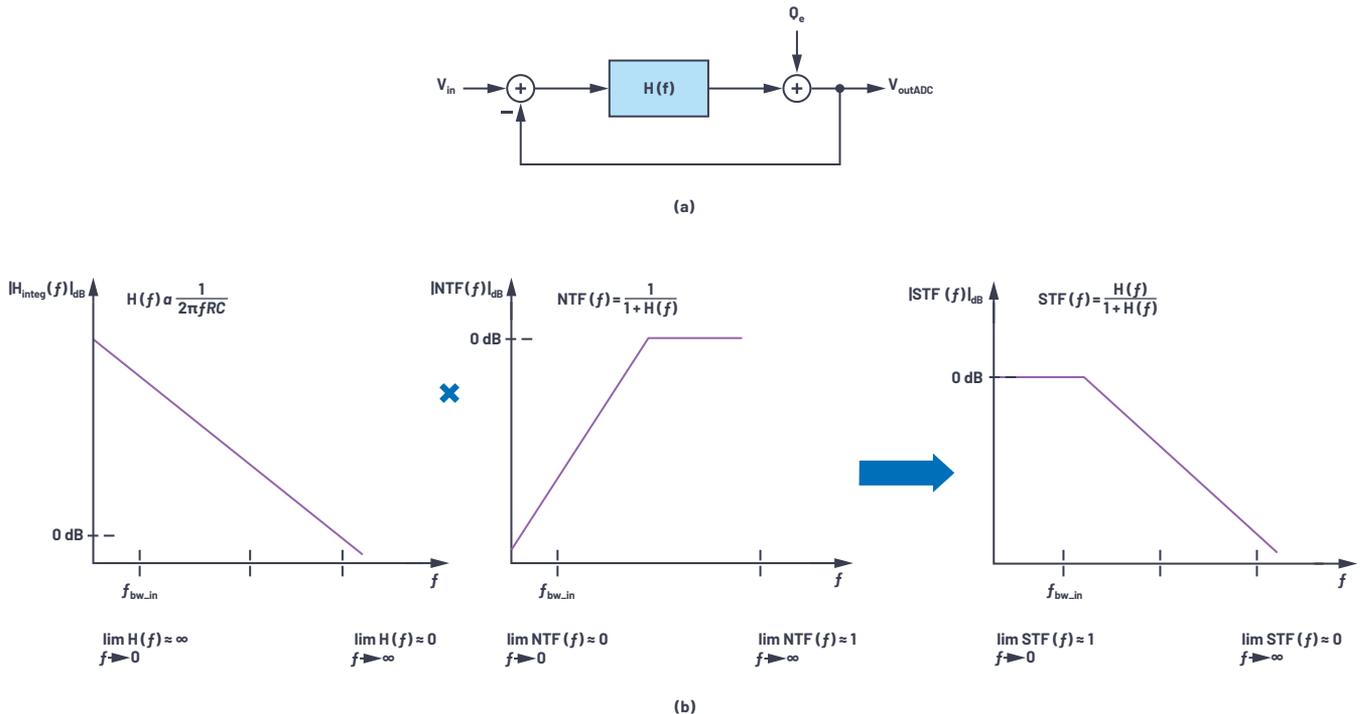


Figure 7. (a) A linear model for simplified analysis and (b)  $STF(f) = H(f) \times NTF(f)$ .

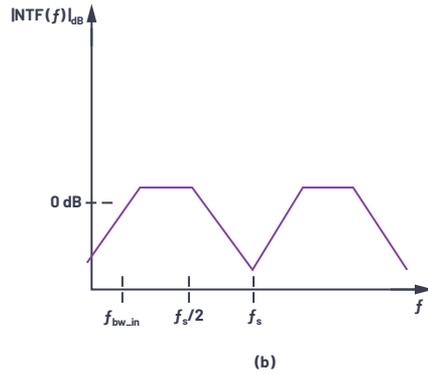
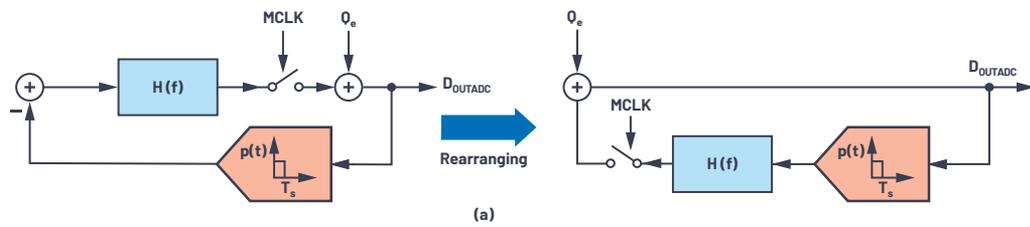


Figure 8. (a) A CTSD modulator loop diagram with input = 0 V and (b) an NTF of a modulator loop.

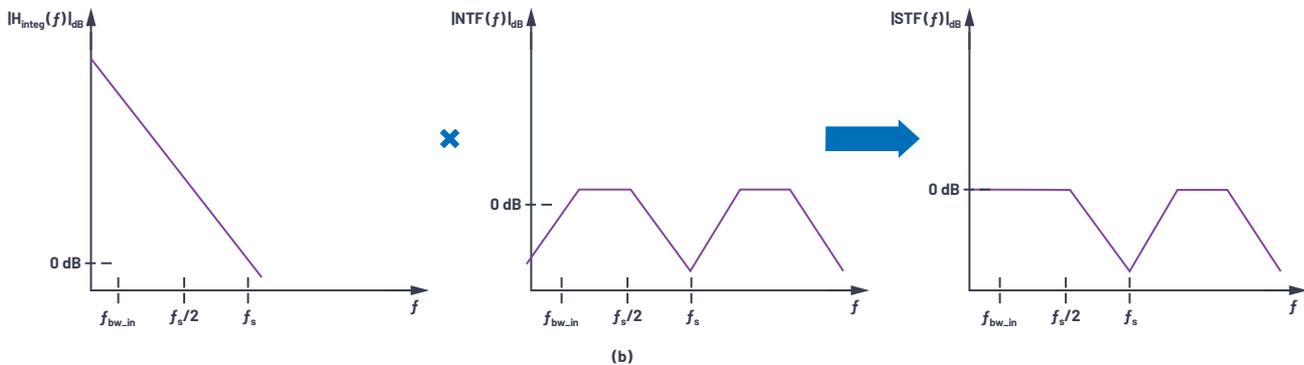
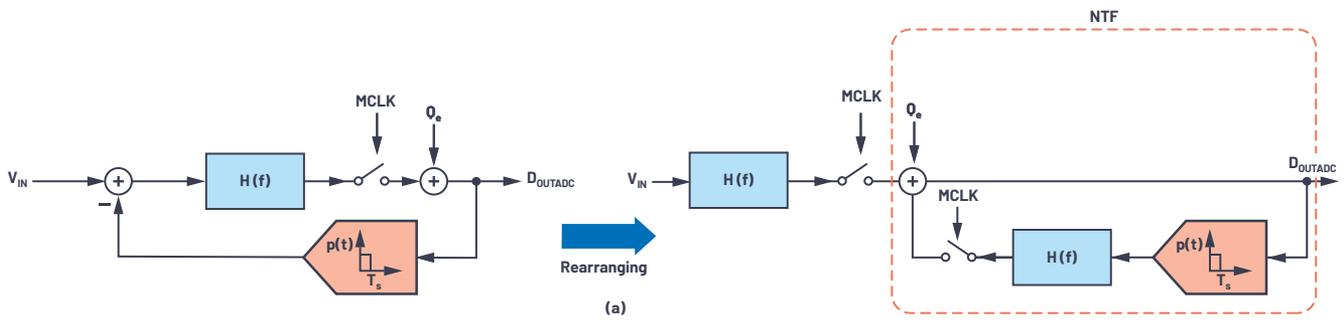


Figure 9. Rearrangement of the modulator loop to illustrate the inherent alias rejection.

The NTF resembles a high-pass filter and the STF resembles a low-pass filter with flat 0 dB magnitude for the frequency band of interest and attenuation for higher frequencies that is equivalent to AAF TF. Mathematically, the signal passes through  $H(f)$ , which has a high gain, low-pass filter profile and then is processed by the NTF loop. Now this understanding can be extended to loop with the sampler by first understanding the NTF block representation.

### Step 2: Block Diagram Representation for NTF

With input  $V_{IN}$  set to 0 V, the block diagram of the modulator loop can be rearranged as shown in Figure 8a and used for NTF representation. With the sampler in the

loop, the NTF response would be similar to a linear model, but with replicated images at every multiple of  $f_s$ , as shown in Figure 8b.

### Step 3: Rearranging the Modulator Loop to Visualize Upfront Filtering Action

If the loop filter  $H(f)$  and sampler of the modulator loop are moved to the input and feedback is as shown in Figure 9, there is no change with regards to the transfer function from input to output. The right side of this rearranged block diagram represents the NTF.

Similar to the linear model from Step 1, in the sampled equivalent system the input signal traverses through high gain  $H(f)$ , and then is sampled and processed through the NTF loop. The transversal of a signal through a loop filter creates a low-pass filter profile before it is sampled. This profile leads to the inherent alias rejection of a CTSD modulator. Thus, the STF for a CTSD modulator loop is as shown in Figure 9.

### Step 4: Complete STF with a Digital Filter

To reduce the redundant high frequency information, the CTSD modulator is partnered with on-chip digital decimation filters and the combined alias rejection TF is shown in Figure 10. Alias from around  $f_s$  is attenuated by the inherent alias rejection property of a CTSD while intermediate interferers are attenuated by a digital filter.

Figure 4 compares the order of AAF required for SAR ADCs, DTSD ADCs, and CTSD ADCs for -80 dB rejection at the sampling frequency vs. the input signal bandwidth. The order and, hence, complexity of AAF with SAR ADCs is the highest, while CTSD ADCs don't require an external AAF as alias rejection is inherent to their design.

### The Signal Chain Advantages Made Possible by a CTSD Architecture

In certain multichannel applications like sonar beamforming and vibration analysis, the phase information between channels is important. For example, the phases between channels need to be accurately matched with a requirement of  $0.05^\circ$  at 20 kHz.

For traditional ADC signal chains, the AAFs are designed using passive RC and op amps. The filter causes a certain magnitude and phase droop in-band that would be a function of corner frequency. For good channel-to-channel phase matching, all the channels need to have the same droop, which indicates the corner frequency of the filters for each channel need to be finely controlled and matched. A second-order Butterworth filter designed for -80 dB rejection at 16 MHz (sampling frequency) and  $f_{\text{dB}}$  of 160 kHz (input bandwidth) could have phase mismatch of  $\pm 0.15^\circ$  at 20 kHz with error tolerance of as low as 1% on the absolute values of RC. The availability of lesser error tolerance RC passives is limited and increases the bill of material (BOM).

Since the AAF is eliminated in a CTSD ADC signal chain, the channel-to-channel magnitude and phase matching is inherently achieved in the frequency band of interest. The phase mismatch is limited by on-chip mismatches of analog modulator loop design, which could be as low as  $\pm 0.02^\circ$  at 20 kHz.

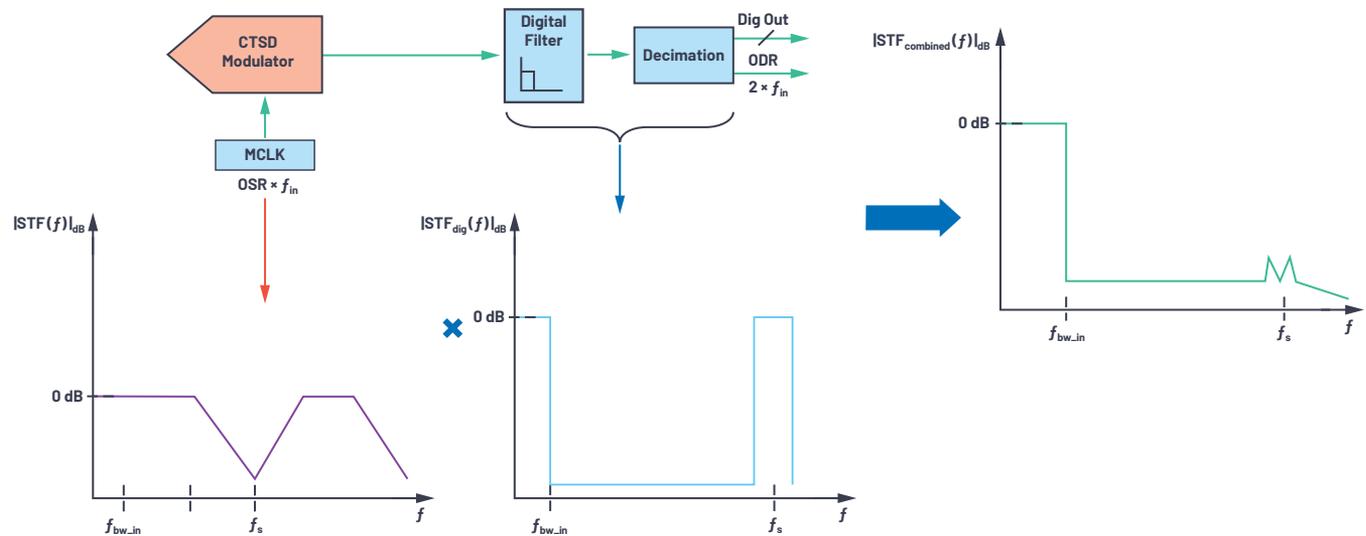


Figure 10. A CTSD modulator loop with back-end digital filters.

### Measuring and Quantifying the Inherent Alias Rejection

New functional checks to measure the alias rejection are introduced in the ADC data sheet of AD4134, which is a precision ADC based on the CTSD ADC architecture. The frequency of the analog input signal of the ADC is swept, and the impact of each out-of-band input signal is calculated by measuring the magnitude of tone folded back, if any, for the test frequency, with respect to the magnitude of the applied tone.

Figure 11 shows the alias rejection of AD4134 for out-of-band frequencies in the performance bandwidth of 160 kHz with a sampling frequency of 24 MHz. For a frequency of 23.84 MHz ( $f_s - 160$  kHz), alias rejection is -85 dB, which is the alias rejection specification of the ADC. It can also be observed that the rejection is better than -100 dB for other intermediate frequencies. Further details on inherent alias rejection with options to further increase this rejection can be found in the AD4134 data sheet.

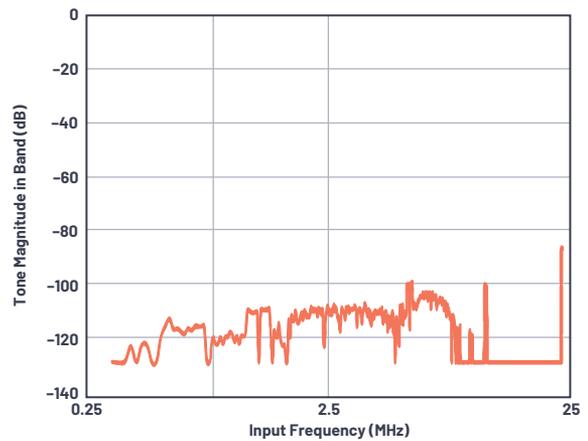


Figure 11. Alias rejection vs. the out-of-band frequency.

The CTSD ADC concepts explained so far can help signal chain designers envision the unique properties of the resistive input, resistive reference, and inherent alias rejection of this architecture. An easy to drive input and reference coupled with the elimination of AAF design for CTSD ADC signal chains, has led to a new simplified ADC front-end design for various applications.

# Chapter 4: Ease of ADC Input and Reference Drive Simplify Signal Chain Design

Abhilasha Kawle, Analog Design Manager, and Roberto Maurino, Design Engineer

## Front-End Design for Traditional ADCs

In this chapter, we will use the terms “sensor” or “input signal” interchangeably to represent any kind of voltage input to the ADC signal chain. The input signal for the ADC signal chain could be a sensor, a signal from some source, or the feedback of a control loop. It is well known that in traditional discrete-time sigma-delta (DTSD) ADCs and successive approximation register (SAR) ADCs, the [sampling network](#) at the input and reference is a switched capacitor load. When the switch turns ON, the capacitor is charged to the input, and when it turns OFF, the capacitor holds the sampled value. At every sampling clock edge when the switch reconnects the capacitor to the input, a finite current termed kickback current is required to charge or discharge the capacitor to the new sampled value. The profile of the current is shown in Figure 1a. Most sensors and reference ICs fall short of driving such [magnitudes of kickback currents](#) and if directly interfaced to the ADC, there is a high probability of the input signal or reference getting corrupted. One of the known solutions to avoid this corruption is to use a driving buffer amplifier to isolate the input sensor and the reference from the ADC. The driver amplifier should have the capability to absorb these kickback currents as shown in Figure 1b. This leads to the requirement of a high slew rate and a high bandwidth amplifier to support the required input charging/discharging currents and settle the kickbacks within one sampling time period.

These stringent requirements limit the choice of buffer amplifiers that can be used on input and reference paths for the traditional ADCs.

On the other hand, a low-pass antialiasing filter is required on the input to ensure high frequency noise and interferers are heavily attenuated so that when they fold back due to sampling into the frequency band of interest, the performance is not degraded. The challenge for incumbent ADC signal chain designers is to fine-tune the opposing requirements of alias rejection and output settling. The front-end design with driver and antialiasing filter for a DTSD ADC is shown in Figure 2.

The input path consists of an instrumentation amplifier (in-amp) that interfaces the sensor to a fully differential amplifier (FDA), which finally drives the ADC. The in-amp isolates the input sensor environment from the ADC circuit. For example, the common-mode (CM) signal of the sensor can be very high, up to 10s of volts. But most FDAs and ADCs don't support this high input common-mode voltage. A general in-amp has the capability to support wide input common mode while providing an output common mode suitable for the FDA and ADC. Another advantage of an in-amp is that it has high input impedance. This means that if the sensor cannot drive the input resistor of the FDA directly, then the sensor can be interfaced with an FDA using an in-amp. The FDA itself would require a high

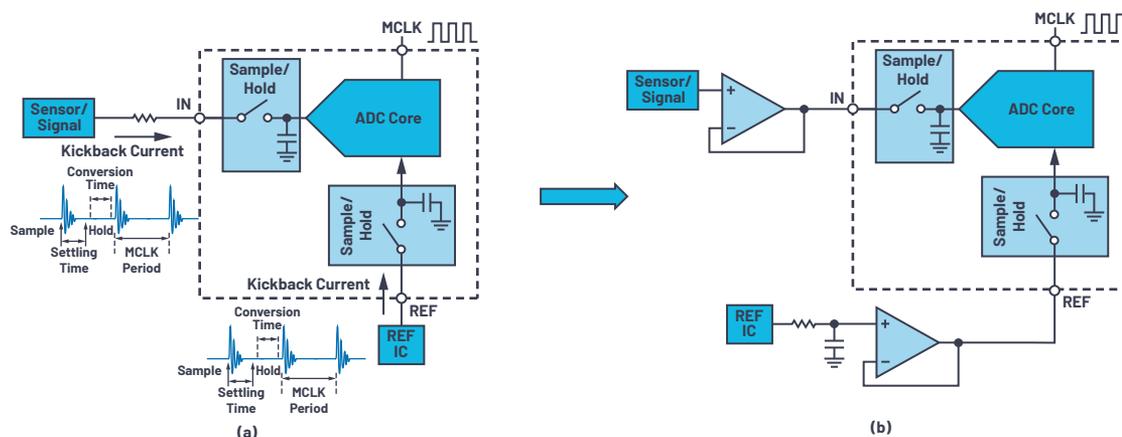


Figure 1. (a) Kickback current on the input and reference of a traditional ADC, and (b) the isolation of the kickback currents by the buffers on an input and reference.

bandwidth and slew rate for faster output settling. An active antialiasing filter (AAF) required for the interferer's immunity is built around the FDA.

The drivers of the input or the reference have conflicting requirements: on the one hand, high bandwidth is desired for settling, but on the other, low bandwidth is required to filter noise and interferers. On the reference path, the front-end design for the DTSD ADC signal chain is shown in Figure 2, which has a reference IC connected to a buffer that drives the reference load of the ADC. It also includes a noise filter that cuts off the noise of the reference IC and buffer beyond a certain frequency. The design requirements of this filter are discussed in a later section. The reference buffer has high bandwidth and slew rate requirements for faster settling of the sampling event disturbances.

## CTSD ADC Advantage: Resistive Input and Reference

In chapter 2, the CTSD ADC architecture was explained to signal chain designers with an unconventional approach to inverting closed-loop amplifiers. As noted in chapter 2, a CTSD ADC can be envisioned as a sigma-delta ADC with a resistive input and reference load. The input and reference structure are a simple resistive load, which means there are no high bandwidth or high slew rate drive requirements. Chapter 3 demonstrated the unique advantages of a CTSD that offers interference immunity due to its inherent alias rejection. In a traditional signal chain design, the external alias rejection filter needed to attenuate the interferers is an added challenge, while CTSD ADCs have no need for an external AAF. Due to the inherent alias rejection property of CTSD ADCs, the signal transfer function of the modulator loop equates to an antialiasing filter that attenuates the high frequency interferers. Because of the resistive input and inherent AAF, the input network is simplified, and the sensor can be directly connected to the ADC. In cases where sensors may not have the capability to drive such a resistive load, an

in-amp could be used to interface the sensor to the ADC. Similarly, on the reference side, due to the resistive load, the reference buffer is eliminated in the CTSD ADC signal chain. A simplified schematic with an in-amp is shown in Figure 3b.

Figure 4 shows further support for how CTSD ADCs help simplify input front-end design. For the DTSD ADC, the discontinuities in input current because of kickbacks when the input sampling switch changes state are noticeably seen. With the CTSD ADC, the input current is observed to be continuous, which maintains signal continuity.

## The Simplified Input Drive Design

We have established that the input drive of the CTSD ADC is resistive. This section will address the value of the input impedance  $R_{IN}$  in planning for the input drive of the ADC.  $R_{IN}$  is a function of the noise performance specified for the ADC. For example, in the AD4134, which is a precision CTSD ADC with a dynamic range of 108 dB with 4 V reference, the input impedance is 6 k $\Omega$  differential. This indicates that when a full-scale 8 V p-p differential input signal is applied to the ADC, the peak current requirement is 1.3 mA p-p. If the sensor can support the input current  $V_{IN}/R_{IN}$ , then it can be directly interfaced to the ADC. The scenarios where a simple amplifier would be required to drive this resistive load are:

- 1) When the sensor doesn't have the drive capability to provide the peak current of  $V_{IN}/R_{IN}$ .
- 2) Signal chain design dictates that gain or attenuation is required for the sensor output.
- 3) Isolating input sensor environment from the ADC circuit.
- 4) The sensor has a large output impedance.
- 5) The sensor is far from the ADC and the track routing could add significant resistance to the input.

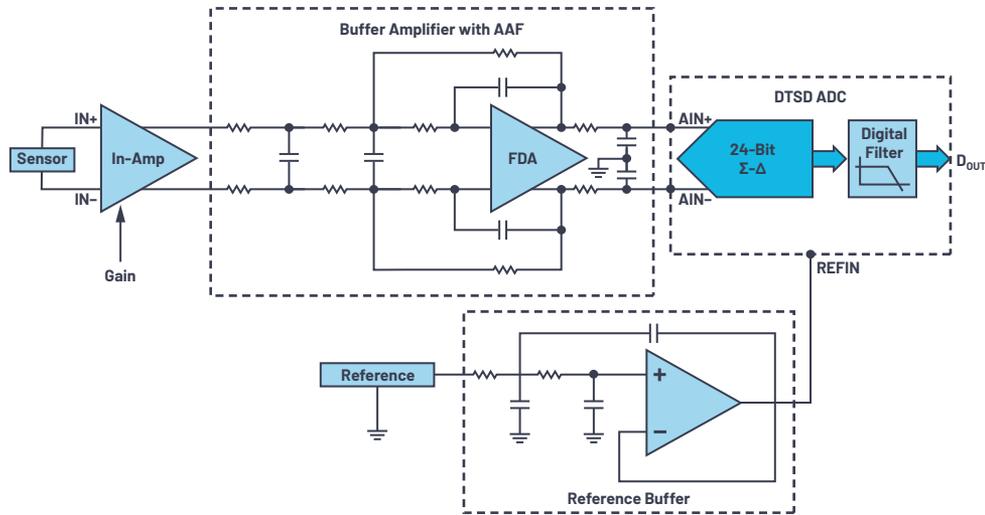


Figure 2. The front-end design of a discrete-time sigma-delta ADC.

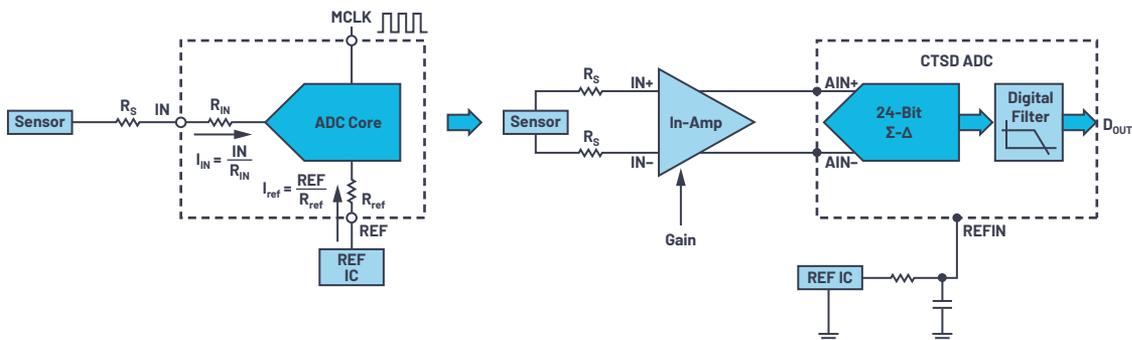


Figure 3. (a) A CTSD architecture offers resistive input and reference load, and (b) a direct in-amp with the reference driving a CTSD ADC.

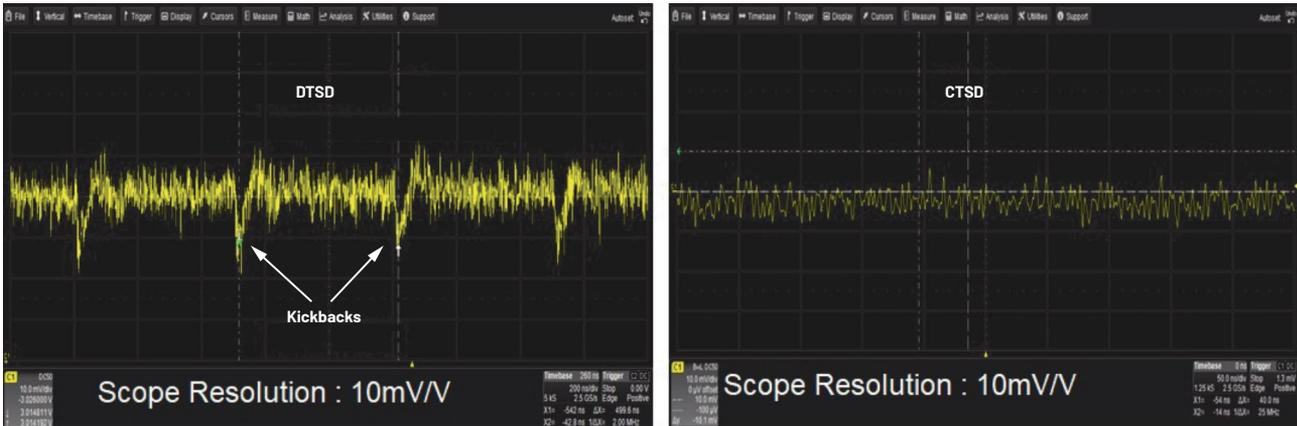


Figure 4. (a) Kickbacks in input current for a DTSD ADC, and (b) a continuous input current profile for a CTSD ADC.

In scenario 4 and 5, there would be a voltage drop across the extra external resistor  $R_{sr}$ , which indicates a signal loss at the ADC input. This leads to gain errors for the signal chain and drift of errors with temperature, which can lead to performance degradation. The temperature gain drift is caused by the different temperature coefficients of the external resistance and internal resistance. This problem can be solved with a simple amplifier to isolate the extra external resistance. Because the driving load for this amplifier is resistive, the selection criteria of this amplifier are:

- ▶ Input impedance: To avoid signal attenuation or loss, the impedance of the sensor should be matched with the amplifier input impedance.
- ▶ Output impedance: The output impedance should be sufficient to drive the resistive input load of the ADC.
- ▶ Output type: As a general signal chain design guideline, a **differential signaling** strategy is recommended for best signal chain performance. A differential output type amplifier or a design technique for single-ended to differential output is best suited for this task. Also, for the best performance, it is preferable to set the common mode of this differential signal to  $V_{REF}/2$ .
- ▶ Programmable gain: The input signal is generally gained or attenuated to map it to the full-scale range of the ADC. This is because the maximum performance can be obtained from an ADC signal chain when the full input range of its ADC is used.

Based on the application, this amplifier could be an in-amp or an FDA or combination of two single-ended op amps forming a differential output amplifier. With no rigid requirements of high slew rate or high bandwidth, a wide range of selection from ADI's amplifier portfolio is available to drive this CTSD ADC based on application requirements. Also, amplifier performance parameters are generally specified with a resistive load, which makes the selection more straightforward.

As an example, for the AD4134, one option for a performance-compatible in-amp with programmable gain options and fully differential outputs is the **LTC6373**. This in-amp provides high impedance to the input source and can easily drive the differential 6 k $\Omega$  impedance with noise and linearity performance on par with the ADC. With its wide range of input common-mode support and programmable gain options, any sensors or input signals with a wide range of signal magnitude can be interfaced with the ADC. An example of input front-end design with this direct in-amp drive is shown in Figure 4.

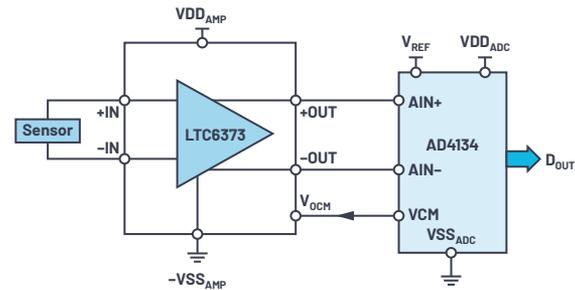


Figure 5. Input front-end design with CTSD ADC directly interfacing to in-amp.

Another example is a low voltage simple front-end design using a fully differential driver amplifier like the **LTC6363-0.5/LTC6363-1/LTC6363-2**, based on the gain or attenuation required, as shown in Figure 6. The scenario when FDAs could be used is when the sensor has the capability to drive the resistive load of the FDA but is single-ended type or has a common-mode that is not supported by the ADC or requires small gain/attenuation in the signal chain.

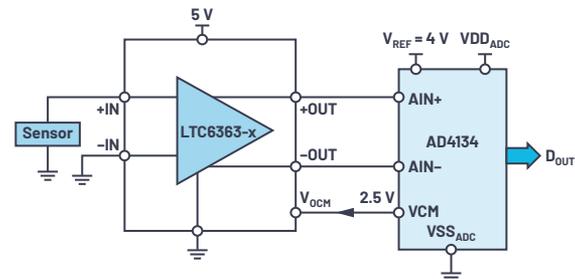


Figure 6. An input front-end design with a CTSD ADC directly interfacing to a fully differential amplifier.

Another example includes a low BOM option for single-ended input conversion to a fully differential signal at the ADC using two single-ended op amps, as shown in Figure 7.

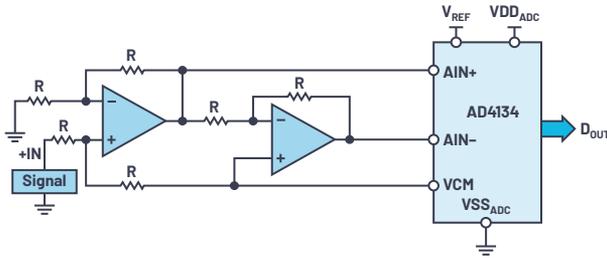


Figure 7. An input front-end design with a CTSD ADC with two single-ended amplifiers.

There are many other examples, like using a combination of a single-ended in-amp and single-ended op amp to build a differential output front end for a very high input CM or low drive strength single-ended type sensor. Any such combination can be selected based on performance, area, and BOM requirements to better suit the application.

A list of other amplifiers that are compatible with AD4134 are:

- ▶ Operational amplifiers: [ADA4625-2](#), [ADA4610-2](#), [AD8605](#), and [ADA4075-2](#).
- ▶ Fully differential amplifiers: [ADA4940-2](#), [LTC6363](#), and [ADA4945-1](#).
- ▶ Instrumentation amplifier: [AD8421](#).

ADI's [amplifier selection guide](#) can be used to select the best amplifier suited for an application. For example, for high linearity applications like audio test equipment, the ADA4945-1 is recommended. For photodiode applications where the most important consideration is very high input impedance, a transimpedance amplifier (TIA) like ADA4610-2 can be used.

With the input front end dramatically simplified with a CTSD ADC, let's have a look at the similar simplification of the reference drive.

## The Simplified Reference Design

The ADC output is a representation of its input and reference, as given by Equation 1.

$$D_{OUT} = V_{IN} \times \frac{2^N}{VREF_{ADC}} \quad (1)$$

Where  $V_{IN}$  = input voltage level,  $VREF_{ADC}$  = reference of the ADC,  $N$  = number of bits,  $D_{OUT}$  = ADC digital output.

Equation 1 indicates that, for the best ADC performance, a clean, uncorrupted reference is important. The three major performance metrics of the ADC signal chain that are affected by an error in the reference are:

- ▶ **Signal-to-noise ratio (SNR):** The major noise contributors to SNR are the input path, the ADC itself, and the reference. For a target total noise at the output of the ADC, accounting for the other noise contributors, the budget for the reference noise is generally  $\frac{1}{3}$  or  $\frac{1}{4}$  of standalone ADC output noise. The reference or reference buffers typically have higher noise than the ADC. If we look at any data sheet for a reference or reference buffer IC, spectral noise density, or  $Noise_{density}$  is one of the specifications. If we revisit [noise calculations basics](#), the total noise at the output of the reference or reference buffer is then given by

$$Total\ Reference\ Noise = Noise_{density} \times \sqrt{(Noise\ Bandwidth)} \quad (2)$$

We cannot control the  $Noise_{density}$  as it is fixed for the chosen reference or buffer. The only controllable parameter is the [noise bandwidth](#) (NBW). To

target low reference noise, we reduce the noise bandwidth of reference or reference buffer. This is generally done by connecting a first-order low-pass RC filter to the ADC, as shown in Figure 8. For a first-order RC filter, the NBW is given by

$$NBW = \frac{1}{2\pi RC} \times \frac{\pi}{2} = \frac{1}{4 \times RC} \quad (3)$$

The ADC reference current,  $I_{ADC}$ , flowing through the filter resistor  $R$  causes a voltage drop, which changes the actual reference value at the ADC. Therefore, it is recommended to choose a small value of  $R$  and a large value of  $C$  to meet the NBW requirements for low reference noise.

- ▶ **Gain error:** As evident from Equation 1,  $VREF_{ADC}$  determines the slope of the output to input transfer function just like in a straight equation such as  $y = mx$ . This slope is also termed as the gain of the ADC. Thus, if the reference changes, the gain of the ADC also changes.
- ▶ **Linearity:** For traditional DTSD ADCs and SAR ADCs, the reference current and the accompanying kickbacks are dependent on the input signal. Therefore, if the reference does not settle completely before the next sampling clock edge, the error seen on the reference will be input dependent and cause nonlinearity. Mathematically, the  $VREF_{ADC}$  is then represented as

$$VREF_{ADC} = (VREF - I_{ADC} \times R) = VREF + aV_{IN} + bV_{IN}^2 + cV_{IN}^3 \quad (4)$$

Referring to Equation 1, the ADC output  $D_{OUT}$  will have various higher order dependencies based on the input of the ADC, and this dependency causes harmonics and integral nonlinearity. Hence, for traditional ADCs there is a hard requirement on the high slew rate and bandwidth of the reference buffer to settle the reference output within the sampling time period.

If we carefully analyze the SNR and linearity, we see that the reference or reference buffer has quite conflicting requirements to satisfy. There is a low bandwidth requirement for noise and a high bandwidth requirement for faster settling. Tuning the fine balance between the two requirements has been an age-old challenge for signal chain designers. Some of the latest DTSD ADCs and SAR ADCs have the reference buffer incorporated on-chip to ease one step in signal chain design, but these solutions require additional power or come with certain performance penalties. Because CTSD ADCs don't need a fast settling buffer and have a resistive input to remove the need for a fast settling driver, they're able to avoid these performance problems.

The CTSD ADC addresses reference driver challenges with the following properties and design improvements:

- ▶ With the resistive load on reference, there is no settling requirement at every sampling clock edge. This allows designers to directly connect the reference IC to the ADC without requiring a dedicated reference buffer.
- ▶ Patented design techniques make the reference current independent of the input and force the reference current of the ADC,  $I_{ADC}$ , to be substantially constant. This is beneficial when an RC filter may be required to reduce the reference noise, as shown in Figure 8. The result is a constant voltage drop across the resistor with no input dependent terms added to  $VREF_{ADC}$ . A provision is designed to digitally correct for the gain error at the system level depending on the value of  $R$  and the voltage measured at the reference pin. Hence, this simple reference interface will not have gain or linearity errors.

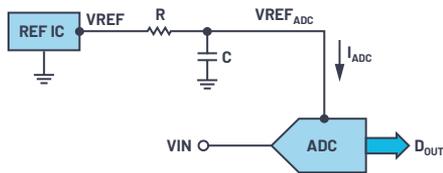


Figure 8. A resistive reference load enabling a direct connection to a reference IC with a passive filter.

Even though a provision has been implemented to digitally correct for the error caused by the voltage drop across  $R$ , one might wonder if this would limit the full-scale range of the CTSD ADC, as the actual reference ( $V_{REF\_ADC}$ ) of the ADC would be lesser than the applied  $V_{REF}$ .

For example, if the  $V_{REF}$  of the reference IC is trimmed and set to 4.096 V and the ADC reference current ( $I_{ADC}$ ) = 6 mA, then, for a filter resistance of  $R = 20 \Omega$ , the actual reference of the ADC ( $V_{REF\_ADC}$ ) is 3.967 V. This is shown in Equation 5. In such a case, when the specified full-scale differential input of  $2 \times V_{REF} = 8.192$  V p-p (which is greater than  $2 \times V_{REF\_ADC}$ ) is applied at the input of the ADC, is there a possibility of saturating the output of the ADC? The answer is “no.” CTSD ADCs are designed to support input magnitudes that are a few mV beyond the reference at the ADC pin, REF<sub>IN</sub>. In our example case of AD4134, this extended range limits the resistor value to a maximum of 25  $\Omega$ . The value of  $C$  for the noise filter is then chosen to satisfy the noise bandwidth calculated.

## Reference Drive Design Simplified

CTSD ADCs have eased the design of the reference drive, but there are still additional factors to consider when selecting the correct  $R$  for the filter followed by digital gain error correction of the voltage drop across the resistor. Digital gain error correction, also known as calibration, is a common feature in many ADCs, and it offers signal chain designers the freedom to compensate for errors in the signal chain at the digital output of an ADC. As such, it may not require an added design step but rather reuse of the same algorithm, which is common for many signal chains. At the face of it then, the selection of resistor doesn't seem to be a particularly involved design step, but there is one caveat: the temperature dependence of the voltage drop. The external filter resistor and  $I_{ADC}$  drift differently with temperature, which in turn causes  $V_{REF\_ADC}$  and the gain of the ADC to drift with temperature. For applications with stringent gain drift requirements, a crude solution consists of calibrating the signal chain periodically. But a much better and innovative solution is made possible because of CTSD technology. Since the ADC reference load current is a constant and function of resistive material used on-chip, it was possible to provide the filter resistor,  $R$ , with 20  $\Omega$  on-chip, as shown in Figure 9.

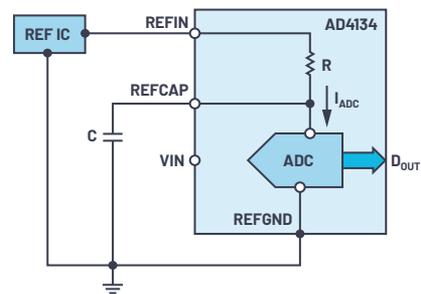


Figure 9. An on-chip reference noise filter resistor simplifying the reference front-end design for a CTSD ADC.

In the new front-end design, the reference IC is connected at the REF<sub>IN</sub> pin and the filter capacitor is connected at the REF<sub>CAP</sub> pin to form the noise filter for the reference IC noise. Since the resistance of the on-chip resistor  $R$  and the  $I_{ADC}$  are both functions of the same resistor material, there is no temperature drift on REF<sub>CAP</sub> ( $V_{REF\_ADC}$ ). AD4134 also uses a patented on-chip reference correction algorithm to digitally self-calibrate for the voltage drop across the on-chip resistor. Thus, the reference drive design is now simplified to the selection of the reference IC and capacitor value based on the performance requirements.

ADR444 is one of the low noise reference ICs that can be used as a companion for a CTSD ADC. The AD4134's data sheet has further details on the capacitor value selection and the internal and external digital gain calibration.

## Summary

CTSD ADCs eliminate many of the barriers to achieving optimal precision performance and simplified front-end design. In the next chapter, we will cover how a CTSD ADC modulator core's output is processed into its final digital output format for use by an external digital controller for optimum processing. From the sigma-delta basics explained throughout this minitutorial, we know the modulator output cannot be processed directly, as it is sampled at a much higher rate. There is a need to reduce the sample rate to the required output data rate (ODR) of the application. Next, we will introduce a novel asynchronous sample rate conversion (ASRC) technique that enables signal chain designers to tune the final ADC output at any desired ODR and go beyond the age-old restriction of limiting ODR to a multiple of sampling frequency.

# Chapter 5: Digital Data Interface Simplification with Asynchronous Sample Rate Conversion (ASRC)

Abhilasha Kawle, Analog Design Manager,  
Naiqian Ren, Applications Engineer, and  
Mayur Anvekar, Digital Design Manager

The job of an ADC is to sample the analog input signal and convert it into an equivalent digitized format. The sample rate at which an application requires the digital data for further processing needn't necessarily be the sample rate at which the ADC samples the analog signal. Each application requires a unique digital output sample rate. A sample rate converter maps the ADC data at the input sample rate to the desired output sample rate. This chapter starts with an overview of sample rate requirements in various applications and establishes the need for an ADC to support a wide range of output sample rates. We follow up with a quick recap of traditional sample rate conversion techniques in known ADC architectures and their shortcomings. Next we introduce the novel asynchronous sample rate conversion (ASRC), which can be paired with any ADC architecture to get any desired output sample rate and simplify the digital interface design with the external digital host. Pairing ASRC with a CTSD ADC offers the best of both worlds, simplifying the signal chain design not only on the analog input side but also on the digital output side of the ADC.

## Sample Rate Requirements

One of the major performance parameters for any ADC application that drives digital data sample rate selection is the accuracy expected from the ADC. The greater the number of samples in the digital data, the more accurate the representation of the analog input. But this would mean processing large amounts of data with its own penalty in external digital host interface design complexity and power. So, based on the accuracy required, the budget for power and design complexity, and the algorithm processing planned, each application decides the sample rate of the digital data. Most of the general sample rates required can be categorized into the following:

### Nyquist Sample Rate

The well-known Nyquist sampling<sup>1</sup> theorem states that the sample rate should be at least twice the input bandwidth for faithful digital representation of the analog input. Therefore, Nyquist sample rate applications have the digital sample rate at twice the input bandwidth of interest. A well-known example for such a sample rate is digital audio data storage on a CD, which is at 44.1 kSPS, where the input audio bandwidth of interest is up to 20 kHz. This is the upper frequency limit of human hearing.

### Oversample Rate

In a few applications, such as frequency harmonic analysis or time-domain analysis, the sampling rate required would be multiple times higher than the input bandwidth. One example of an oversample rate is the time-domain analysis of a transient signal in a shock detection environment, as shown in Figure 1. If the sample rate for such a signal is the Nyquist sample rate, we will not get the complete picture of the peaks and turfs. Having more sample points results in faithful reconstruction and analysis of the signal.

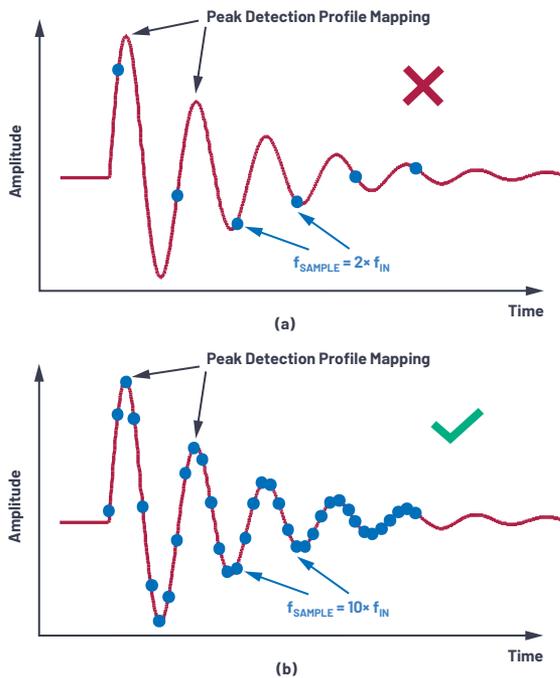


Figure 1. Time-domain analysis of a transient signal with (a) a Nyquist sample rate and (b) an oversample rate.

## Variable Sample Rate

In certain applications, such as coherent sampling, the requirement is to adjust the output sample rate based on the analog input frequency with good resolution. Power line monitoring is one such example application that requires coherent sampling to meet the Class A power quality meters specified in IEC 61000-4-30. The accuracy requirements in these standards dictate that the sample rate needs to track the input line frequency drift. In these applications, clock synthesizer circuitry on the power line generates the ADC's output digital data sample clock, as shown in Figure 2.

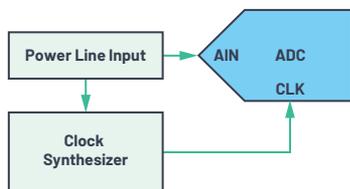


Figure 2. Variable sample rate: power line quality monitoring.

## Multisample Rate

In multichannel applications that detect and analyze a wide range and different types of analog inputs, such as oscilloscopes or data acquisition, the sample rate can be different for each channel. In this case, the ADCs used in the platform should have the flexibility to support a multisample rate.

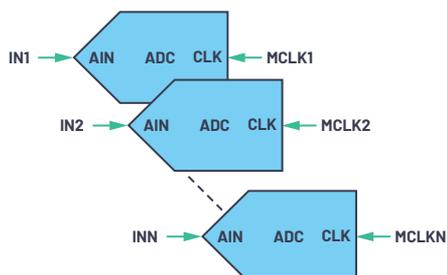


Figure 3. A multisample rate application.

Thus, we see that the digital data sample rate requirement is unique to the application, and there is no one-size-fits-all sample rate. Hence, a broad market ADC would need to support a wide range of programmable digital data sample rates.

Figure 4 showcases a generalized ADC digital data interface with the external digital host. A point to note here is that the digital data interface discussed in this article doesn't include the device configuration control interfaces such as SPI or I<sup>2</sup>C.

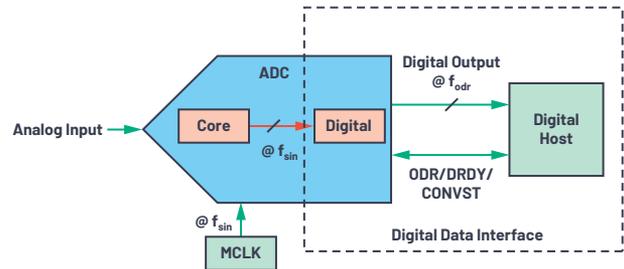


Figure 4. A generalized ADC digital data interface.

The core ADC samples the analog input with a sampling clock at rate  $f_{\text{sin}}$ , as shown in Figure 4. The input sampling clock itself is generally represented as MCLK in most data sheets. The final digital output data is at sample rate  $f_{\text{odr}}$ . Usually those pins are labeled as ODR or DRDY or CONVST clock in the data sheets. In this article, we will use the blanket term ODR clock to represent the digital output data clock.

The ADC core's sample rate,  $f_{\text{sin}}$ , depends on the ADC architecture. The digital output data rate,  $f_{\text{odr}}$ , depends on the data interface requirements with the external digital host. In most ADC signal chain applications,  $f_{\text{sin}}$  and  $f_{\text{odr}}$  can have different values and be uncorrelated. Hence, there is a need for a sample rate conversion that maps the ADC core's data at  $f_{\text{sin}}$  to the digital output data at  $f_{\text{odr}}$ . In the following sections, we will discuss traditional sample rate conversion techniques used in well-known ADC architectures such as Nyquist ADCs and oversampled ADCs. Also, we will get insights into other associated digital data interface requirements.

## Sample Rate Conversion in Nyquist Rate ADCs

In Nyquist rate converters, the ADC core's sampling frequency is twice the analog input bandwidth,  $f_{\text{in}}$ . The most common example under this category is the Nyquist rate SAR ADC, where the input and output sample rates are the same. Hence, the digital output data rate clock, ODR, can be reused as the ADC core sample clock, MCLK. In SAR ADC data sheets, the digital output data clock is represented as CONVST or DRDY. But, as indicated earlier, we will refer to all of these as ODR clock in this article. The ODR and MCLK combination leads to a simplified digital data interface, as shown in Figure 5, with only one clock routing. Since the clock is sourced and controlled either by an external clock source or an external digital host, the ADC is externally clocked. This means the ADC is running in an external hosted mode.

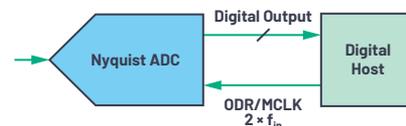


Figure 5. Simplified digital data interface of Nyquist rate converter in hosted mode.

It is easy to scale the sample rate  $f_{\text{odr}}$  based on the application requirements and the analog input bandwidth. With  $f_{\text{odr}}$  scaling, we are also scaling the ADC core's sample clock rate,  $f_{\text{sin}}$ . An added advantage is that as  $f_{\text{odr}}$  scales, the power of the entire ADC scales linearly. This simplified digital data interface leads to many other extended benefits, one of them being ease of synchronization in multichannel applications.

## Ease of Synchronization

In a single-channel ADC application, the local clock provided to the ADC would synchronize the digital data inherently to the given clock. In multichannel ADC applications, the challenge is to guarantee synchronous sampling of multiple analog inputs and synchronization of digital data to the clock edge of the ODR clock for further digital processing. There are many well-known examples of synchronized multichannel applications, such as an audio application where the left and right channels have specific synchronization requirements. Another typical example is monitoring various power lines in a power grid. Again, within each power line, synchronization is required between voltage, current, and power input measurements. With Nyquist rate ADCs, as shown in Figure 6, multichannel synchronization can be easily achieved by sharing and having well-planned routing of the ODR clock. Well-planned routing involves ensuring that the ODR clock propagates with equal delay to each of the ADCs and provides the best possible channel synchronization.

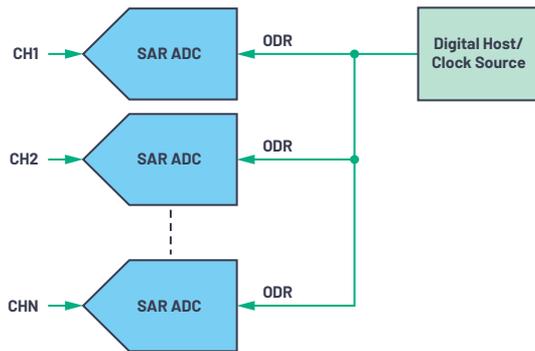


Figure 6. Ease of synchronization in a Nyquist rate sample rate converter.

A simplified digital data interface is a significant advantage of Nyquist rate converters. Let's discuss a few digital data interface challenges where it falls short.

## Limitations of Nyquist Rate Control

### Noise Scaling

In Nyquist rate converters based on the analog input bandwidth of the application, the digital data clock can be easily scaled. The clock scaling gives an advantage in power, but the ADC noise increases due to a phenomenon called alias foldback. The extension of the Nyquist sampling theorem is that any information beyond the Nyquist frequency folds back or aliases back into the frequency band of interest. The ADC's analog input would have a lot of unwanted information or noise from the source and the input analog circuitry, extending to very high frequency. The ADC sampling causes any input noise beyond  $f_{\text{sin}}/2$  to fold back, causing the noise in the input bandwidth of interest to increase. As seen in Figure 7, as the sampling rate reduces, more such external noise folds back, increasing the noise in the ADC's output.

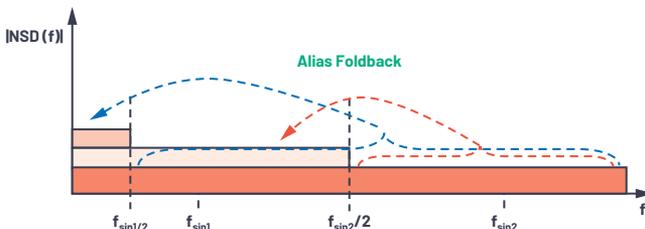


Figure 7. Input noise foldback vs. sampling frequency.

## Clock Timing Constraints

For SAR ADCs, the analog input sampling clock requires two phases, as shown in Figure 9a. One is the sampling phase, where the input sampling capacitors of the ADC charge to analog input, and the other is the conversion phase, where this sampled data is digitized. The sampling circuits of the ADC generally require some minimum sampling time for the best possible ADC performance. So, the external digital host or clock source generating this clock needs to adhere to these timing constraints.

### Clock Jitter

A clock routing on an application board is sensitive to the supply noise of the clock source or coupling to other signals on the board, as this noise adds uncertainty to the clock edges. The uncertainty in the clock edges is known as jitter, and there are various types of clock jitter on the sample clock that can affect the ADC's performance. The most common one is cycle-to-cycle rms jitter. It adds variability to the sampling point of the analog signal, resulting in performance degradation, as shown in Figure 8. More details on the effects of rms clock jitter on an ADC's performance are explained in various articles.<sup>2</sup>

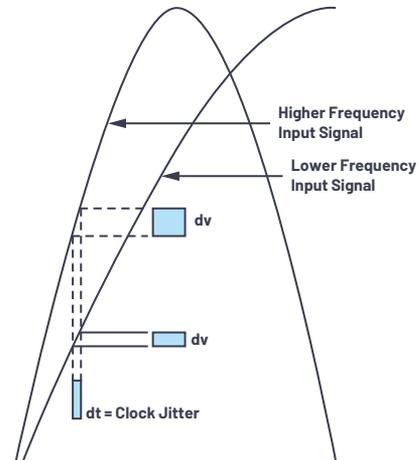


Figure 8. Clock jitter causing uncertainty in sampling point of analog input.

To summarize, the added error in the ADC data because of clock jitter can be quantified as degradation in signal-to-noise ratio (SNR),

$$SNR_j = -20 \times \log_{10}(2\pi\sigma_j f_{\text{sin}}) = -20 \times \log_{10}(2\pi\sigma_j \times 2f_{\text{in}}) \quad (1)$$

where  $\sigma_j$  is the rms jitter.

Equation 1 implies that to meet the required SNR, we either limit the input bandwidth or employ extra techniques to filter clock noise when the digital host or clock source is noisy.

Clock jitter is a more significant challenge in multichannel applications where balancing synchronization and jitter addition due to long clock routings requires good clock architecture planning.<sup>3</sup> Appropriate isolation and buffering are planned to ensure a low noise clock at the ADC in such scenarios. Isolation is implemented using commonly available digital isolators but requires an extra budget in design complexity and power.

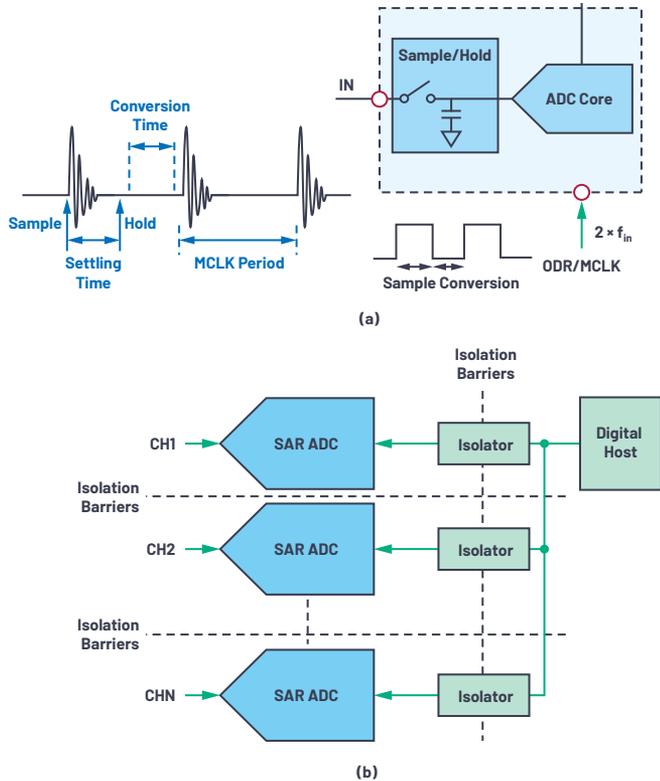


Figure 9. Limitations of a Nyquist rate converter data interface: (a) clocking time constraints and (b) isolation requirement in multichannel applications.

With an overview of sample rate control in Nyquist rate ADCs, let's next look into the sample rate control technique used in oversampled ADCs.

## Sample Rate Conversion in Oversampled ADCs

As illustrated in earlier chapters of this minitutorial, sampling and digitizing a continuous-time signal causes loss of information and introduces quantization noise in the sampled output. A class of ADCs follows the principle that the greater the number of samples, the better the accuracy and the lesser the quantization noise error. Hence, the analog input sample rate is higher than the Nyquist sampling rate and is termed oversampling. Some new precision SAR ADCs use this technique of oversampling and are called oversampled SAR ADCs. Figure 10a shows the noise advantage of oversampling SAR ADCs. Another class of ADCs that uses the oversampling concept is the sigma-delta ADC.<sup>4</sup> Here the quantization noise,  $Q_e$ , is further shaped and pushed out to improve performance in the input bandwidth of interest. Figure 10b shows the noise shaping characteristic of quantization noise for a sigma-delta modulator. Mathematically, the sampling frequency is  $OSR \times f_{odr}/2$ , where  $OSR$  is the oversampling ratio.

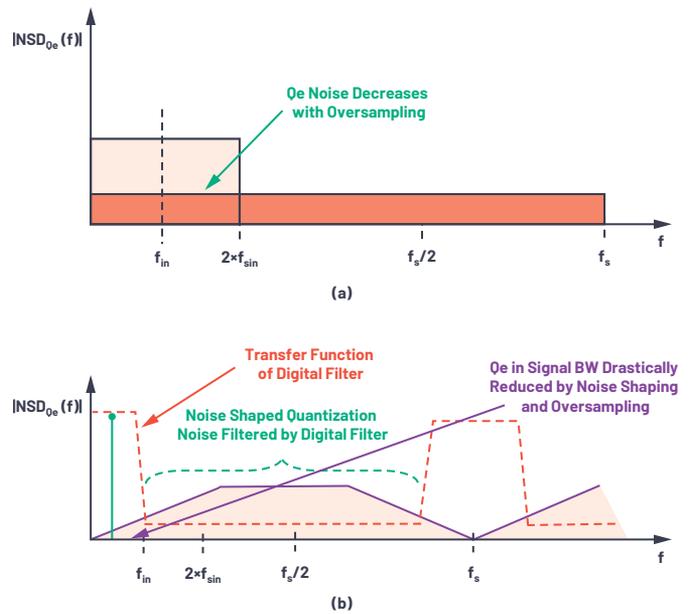


Figure 10. (a) The frequency spectrum of an oversampled SAR ADC and (b) the frequency spectrum for a sigma-delta ADC.

Directly interfacing the core ADC's oversampled data to the external digital host implies overloading it with a lot of redundant information. Moreover, in some cases, the host may not support the stringent timing constraints required for such a high digital data rate transmission and also causes high power dissipation. Therefore, it would be optimal if only the performance-optimized data in the input bandwidth of interest is provided. This would mean the output digital data rate should be reduced or decimated to the Nyquist rate, ( $2 \times f_{in}$ ), or a few multiples of the Nyquist rate, as desired by the application. Hence, a sample rate converter is needed to map the ADC's core data at a high sample rate of  $f_{sin}$  to the required  $f_{odr}$ .

Traditionally, a digital sample rate conversion technique called decimation is available that filters and decimates the core ADC data by multiple  $2^N$ , as shown in Figure 11. An input sampling clock termed MCLK is provided to the ADC. The desired digital output data sample rate (ODR/DRDY) clock, which is a divided version of MCLK, is provided as output. The division ratio is achieved by programming  $N$ , based on the decimation rate required. To get a much finer resolution on  $f_{odr}$  programming, the MCLK can also be scaled based on the input bandwidth requirement of the application. If we observe the digital data interface of oversampled ADCs, the ODR clock is given and controlled by the ADC. This means the ADC provides the clock, which is named the ADC in host mode.

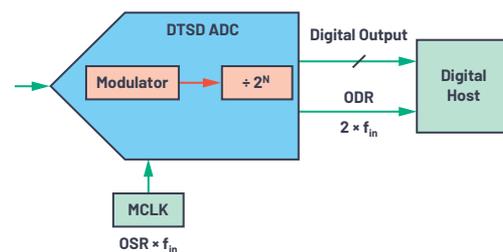


Figure 11. The digital data interface of a discrete time sigma-delta (DTSD) ADC.

Thus, with decimation as a sample rate conversion technique, the ADC can provide high performance digital data at a lower output data rate. But this technique has its own limitations.

## Limitations of Decimation as Sample Rate Control

### Nonlinear Noise, Power Scaling

In variable rate applications, the decimation rate, the MCLK, or both can be scaled. When only the decimation rate is increased, the  $f_{odr}$  reduces, and the noise decreases as the digital filter filters more quantization noise. Only the power in the digital filter reduces linearly. If MCLK is reduced as discussed in SAR ADCs, the power of the entire ADC decreases linearly, but noise increases due to alias foldback.

Many systems adjust both the ADC's MCLK and decimation rate to achieve a wide range of ODR, but this approach can result in an undesired step change in measurement noise performance or system power performance.

### Clock Jitter

Oversampled ADCs, since the input sampling clock frequency,  $f_{sin}$  is higher, are much more sensitive to clock jitter than the Nyquist rate SAR ADCs, as indicated by Equation 1. Therefore, the clock source and the clock routing for an MCLK are planned based on jitter noise tolerable by the application. Be it a single-channel or multichannel application signal chain, there would be many switching signals running across the application board. Coupling from such noisy signals can increase the clock jitter on an MCLK. Thus, isolation needs to be planned for an MCLK using digital isolators for optimum ADC performance. This extra design planning costs in area and power. As indicated earlier, for finer resolution in  $f_{odr}$  programming, an MCLK is also scaled. However, the availability of an MCLK clock source with the required  $f_{sin}$  value and jitter requirements may be limited.

### Synchronization

Achieving synchronization is another added challenge in oversampled ADCs. Generally, an extra pin called SYNC\_IN is provided for synchronization in sigma-delta ADCs. The trigger of the SYNC\_IN pin initiates simultaneous sampling of analog input and reset of decimation filters. After the digital filter settling time, the digital output data is synchronized. The digital output data during the settling of the digital filter is interrupted, as shown in Figure 12. It also assumes that the MCLK and SYNC\_IN command of all the ADCs is synchronized. Achieving such synchronization on a high sample rate clock, especially in the presence of isolators or synthesizers, would be a big challenge. One system solution identified toward solving the data interruption and synchronization challenge is a clock synthesizer circuit, such as a PLL, that would generate synchronized MCLKs for all the channels.

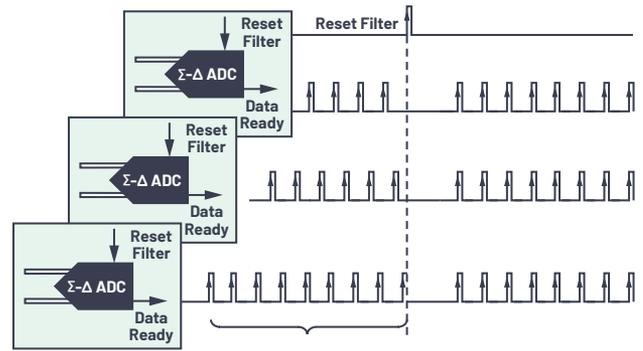


Figure 12. Synchronization in a DTSD ADC with data interruption.

To quickly summarize, when the SYNC\_IN pin is triggered, the PLL loop initiates the clock synchronization to a reference clock. During the time the PLL settles, the MCLK rate adjusts such that at the end of it, the input ADC sampling edge and the ODR clock edges are synchronized. The hows and whats of this solution can be found in “Newest Sigma-Delta ADC Architecture Averts Disrupted Data Flow When Synchronizing Critical Distributed Systems.”<sup>45</sup>

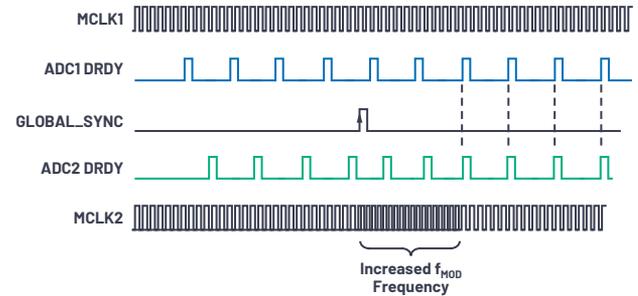


Figure 13. A PLL-based solution for synchronization in a DTSD ADC.

The takeaway is that, with extra on-board circuit, PLL, or clock synthesizer requirements when compared to SAR ADCs, the synchronization of sigma-delta ADCs or oversampled SAR ADCs adds to design complexity and power. ADI has explored another novel technique that helps ease the challenge of synchronization to a certain extent, called synchronous sample rate conversion.

### Synchronous Sample Rate Conversion (SRC)

A solution to a few of the discussed challenges of simple decimation is using a synchronous sample rate conversion.<sup>6</sup> The advantage of SRC is that the decimation rate can be any integer or fractional ratio of  $f_{sin}$ , allowing granular control of  $f_{odr}$ . ADI has explored this technique and paired it with a precision DTSD converter in the AD7770. More details on SRC can be found in the [data sheet](#) or [reference material](#) of the AD7770.

The highlight is that, with the possibility of fine resolution in  $f_{odr}$  programming in SRC, synchronization becomes easier. For example, instead of tuning the

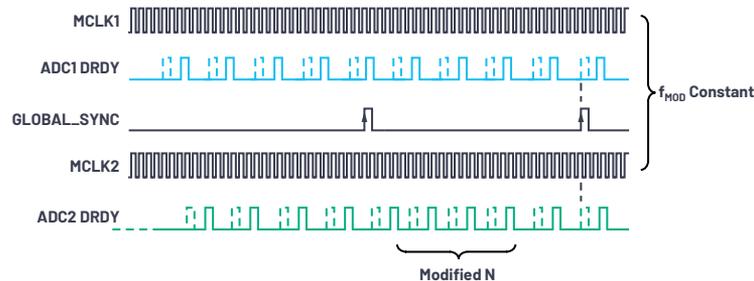


Figure 14. Multichannel synchronization using SRC.

external MCLK, the decimation rate is varied in very fine steps. So when SYNC\_IN is triggered, the channels would be synchronized, as shown in Figure 14.

Achieving finer  $f_{odr}$  without scaling the MCLK is an answer to most of the limitations discussed with the simple decimation technique. SRC also has its own limitations and challenges to solve.

## Limitations of SRC

The synchronization challenge of having the same MCLK for all channels is not addressed with SRC.

## Clock Jitter/Synchronization

SRC has the same limitations as a simple decimation sample rate control in terms of MCLK jitter. The sensitivity of ADC performance to clock jitter because of high  $f_{sin}$  needs to be addressed by planning isolation barrier or noise filtering circuits on the MCLK. This challenge further scales up in multichannel applications owing to the routing of the MCLK to multiple ADC channels. To achieve synchronization, the MCLK and SYNC\_IN pin signals need to be synchronized, as shown in Figure 16a. The challenge is that all clocks reach the ADCs at the same time, independent of the PCB distance from the clock and the possible delays through the isolation barrier. A carefully designed clock plan including the isolation barrier and the routing architecture needs to be built to ensure that all ADC channels equally see delays, even with isolators in the path.

## Interface Mode

The digital data interfaces we have discussed up until now are the host mode and the hosted mode, and there is a correlation to the ADC core architecture. For example, Nyquist rate ADCs' digital data clock is controlled and provided by an external clock source or digital host. Hence, they are limited to be programmed as hosted mode. Oversampled ADCs provide and control the digital clock to the external digital host. Hence, they are limited to be programmed as host mode. Thus, there is a general limitation in all the sample rate control techniques discussed, that the data interface cannot be independently planned.

A solution to most digital data interface challenges would be to decouple the MCLK clock and the ODR clock domains. Therefore, ADI reintroduces the novel asynchronous sample rate conversion technique that enables the independence of the ODR clock and the data interface clock—thus breaking the age-old barrier of ADC core architecture limiting the selection and control of the ODR clock.

## Asynchronous Sample Rate Conversion

ASRC resamples the core ADC data at  $f_{sin}$  in the digital domain and maps it to any desired output data rate. ASRC can be thought of as a digital filter that can achieve any noninteger decimation. However, an optimized implementation in terms of performance, area, and power would be the one where the ASRC handled the fractional decimation and was followed by a simple decimation filter to address integer decimation, as shown in Figure 15. The ASRC resamples the ADC core data and decimates the data by  $f_{sin}/N \times f_{odr}$ . The data at the output of ASRC is at the rate of  $N$  times  $f_{odr}$ . At the same time, decimation filters get the required  $\div N$  decimation.

In one form of ASRC implementation, the factor  $f_{sin}/N \times f_{odr}$  can be programmed by the signal chain designer based on the  $f_{sin}$  of the ADC and the required  $f_{odr}$  and  $N$  known from decimation filters implemented on the ADC. This is similar to programming the decimation rate in SRC—the difference being the decimation ratio can be an irrational ratio and a very fine resolution is possible. In this case, like in SRC, the ODR clock is synchronized to the MCLK and is an output generated on-chip by dividing the MCLK.

Another form of ASRC implementation is where the ODR clock is provided by an external clock source or digital host similar to Nyquist rate converters. In this case, ASRC has an internal clock synthesizer that will calculate the  $f_{sin}/N \times f_{odr}$  ratio and generate the required clocks for ASRC and decimation filters. The ODR needn't be synchronized to MCLK and can be independently set at any sample rate.

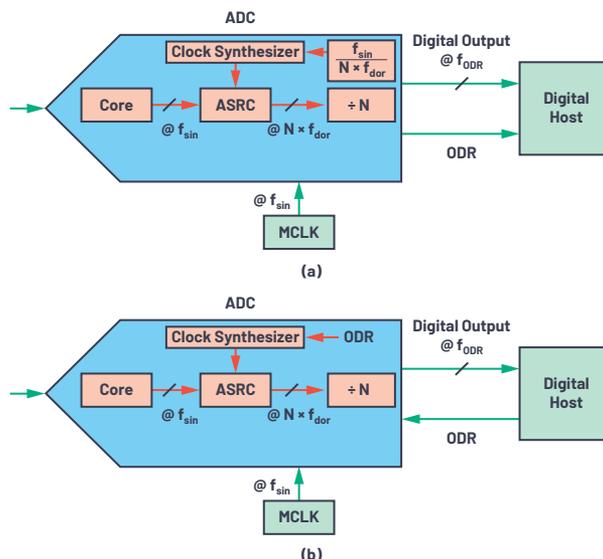


Figure 15. ASRC implementations: (a) programming the ratios and (b) on-chip calculation of the ratio.

Thus, in any form, the ASRC technique enables signal chain designers to granularly set  $f_{odr}$  and go beyond the age-old restriction of limiting  $f_{odr}$  to the integer or fractional ratio of the input sampling rate. As a result, the ODR clock's sample rate and timing requirements are now purely a function of the digital interface and completely decoupled from the ADC's input sampling frequency. In any of these two forms of implementations, we will see that the advantages of ASRC lead to ease of digital data interface design for signal chain designers.

## Value Proposition of ASRC

### Decoupling the MCLK and ODR Clock

In either form of its implementation, because of the possibility of finer resolution on  $f_{odr}$  programmability/scaling that can be adjusted by a fraction of a Hertz, ASRC allows independent selection of the MCLK and ODR clock rates. The MCLK rate,  $f_{sin}$ , can be chosen based on ADC performance and clock jitter requirements, while the ODR clock  $f_{odr}$  can be implemented based on digital data interface requirements.

### Clock Jitter

In both Nyquist rate converters and oversampled ADCs, we saw that MCLK and ODR are correlated. The MCLK was required to be scaled to achieve finer resolution in  $f_{odr}$ . But the availability of clock sources that match the clock jitter requirements of the MCLK at any  $f_{sin}$  rate was limited. Thus, there was a trade-off between ADC performance degradation due to MCLK jitter and possible resolution of  $f_{odr}$ . In the case of ASRC, the MCLK source can be selected to give the best possible clock jitter, as the value of  $f_{sin}$  can be chosen independently irrespective of ODR.

### Interface Mode

Since ASRC decouples the MCLK and ODR clock rates, it gives a degree of freedom on interface mode choice. Any ADC with an ASRC back end can independently be configured as host or hosted peripheral irrespective of ADC core architecture.

## Synchronization

In previously discussed techniques for multichannel synchronization, the MCLK clock routing has stringent requirements. Isolation barriers and clock architectures need to be planned to meet clock jitter and synchronization requirements. Now the MCLK source can be independent for each channel, as shown in Figure 16b. In the host mode of operation, the decimation rate can be programmed independently to achieve synchronization. And in hosted mode, as shown in Figure 16b, the ODR can be shared and synchronized. Since the rate of the ODR clock is low and is just a digital data strobing clock, it doesn't have jitter requirements as stringent as the MCLK. Hence, the stringent requirements of isolation barriers or clock routing are relaxed.

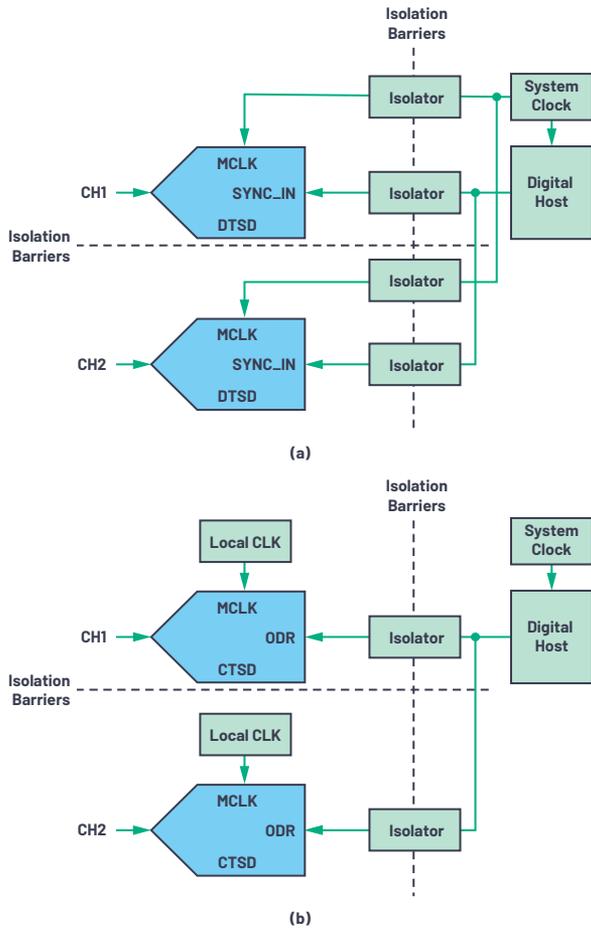


Figure 16. (a) Clock and SYNC\_IN distribution using SRC, and (b) simplified clocking and synchronization using ASRC.

In summary, ASRC opens up avenues to explore innovative and simplified ways of interfacing with external digital hosts. Furthermore, an MCLK can be independent, making it an ideal choice for pairing with CTSD ADCs.

## ASRC Pairing with CTSD ADCs

The CTSD ADC core also works on the sigma-delta concept of oversampling and noise shaping while giving architectural advantages of resistive input, reference drive, and inherent alias rejection. These traits drastically simplify the analog input front-end design. As discussed in chapter 2, since the core ADC loop is a continuous-time system, the loop coefficients are tuned to a fixed input sampling rate that would be specified in the data sheet.

The limitation of CTSD ADCs is that the MCLK is not scalable like in DTSD or SAR ADCs. If a CTSD ADC is paired with an SRC, then the ODR would have been a function of this fixed sampling clock. This would have limited the avenues where a CTSD ADC could be used. Applications can require ODR, which is an irrational ratio of this fixed  $f_{\text{sin}}$ . Also, the CTSD ADC requires that this MCLK be accurate and has low jitter for optimum ADC performance. For example, the order of requirements would be like  $\pm 100$  ppm accuracy in frequency and rms jitter of 10 ps. So, the MCLK would have required a well-planned clocking architecture to guarantee low jitter noise addition in a multichannel application. This magnitude of challenge increases because the MCLK is a high frequency clock.

ASRC, with its ability to decouple an MCLK and ODR, fits in well to address the limitation of a CTSD ADC architecture. The MCLK clock source can be local and near the ADC to avoid long clock routing and coupling to other signals that could increase jitter noise. Thus, combining ASRC with a CTSD ADC brings in a new class of ADC that leverages the architectural advantages of the CTSD ADC while addressing its limitations in the fixed, low jitter MCLK.

## Conclusion

ASRC gives independence to signal chain designers to select the required output data rate granularly. Another advantage is that with the input sampling clock and ODR clock dependencies decoupled, the digital isolations can be efficiently planned in multichannel applications. The freedom to configure the data interface irrespective of core ADC architecture is another simplification to the signal chain. This minitutorial helps understand the various advantages and simplifications ASRC brings to digital data interface over traditional sample rate conversion. In general, ASRC can be paired with any ADC core architecture, but pairing it with a CTSD ADC eases the complete signal chain design on the analog input end as well as the digital data end. With the need and value proposition of ASRC established, keep a lookout for the follow-up article, which will dive deeper into the concept of ASRC and give insights into the building blocks of ASRC. These details help signal chain designers understand the performance metrics associated with ASRC and leverage its advantages for their applications.

## References: Part 1

"Application Note AN-282: Fundamentals of Sampled Data Systems."

Analog Devices, Inc.

[Driving Precision Converters: Selecting Voltage References and Amplifiers.](#)

Analog Devices, Inc.

Kester, Walt. "MT-021 Tutorial: ADC Architectures II: Successive Approximation ADCs." Analog Devices, Inc., 2009.

[Sigma-Delta ADC Tutorial.](#) Analog Devices, Inc.

Shaikh, Wasim and Srikanth Nittala. "AC and DC Data Acquisition Signal Chains Made Easy." *Analog Dialogue*, Vol. 54, No. 3, August 2020.

## References: Part 2

<sup>1</sup>Hank Zumbahlen. "Mini Tutorial MT-213: Inverting Amplifier." Analog Devices, Inc., February 2013.

<sup>2</sup>Walt Kester. "MT-014 Tutorial: Basic DAC Architectures I: String DACs and Thermometer (Fully Decoded) DACs." Analog Devices, Inc., 2009.

<sup>3</sup>Luis Orozco. "Programmable-Gain Transimpedance Amplifiers Maximize Dynamic Range in Spectroscopy Systems." *Analog Dialogue*, Vol. 47, No. 2, May 2013.

<sup>4</sup>Walt Kester. "MT-022 Tutorial: ADC Architectures III: Sigma-Delta ADC Basics." Analog Devices, Inc., 2009.

<sup>5</sup>Pawel Czapor. "Sigma-Delta ADC Clocking—More Than Jitter." *Analog Dialogue*, Vol. 53, No. 3, April 2019.

Pavan, Shanthi, Richard Schreier, and Gabor C. Temes. *Understanding Delta-Sigma Data Converters*, 2<sup>nd</sup> edition. Wiley, January 2017.

## References: Part 3

[Antialiasing Filter Design Tool](#)

[Filter Design Tutorial](#)

Kester, Walt. "MT-002: What the Nyquist Criterion Means to Your Sampled Data System Design." Analog Devices, Inc., 2009.

## References: Part 4

"Driving Precision Converters: Selecting Voltage References and Amplifiers."

Analog Devices, Inc.

Mahaffey, Anna. "Driving SAR ADCs Part 1: Analog Input Model." Analog Devices, Inc.

Shah, Anshul. "Why Does Voltage Reference Noise Matter?" *Analog Dialogue*, Vol. 54, No. 1, March 2020.

## References: Part 5

<sup>1</sup>Walt Kester. "MT-002 Tutorial: What the Nyquist Criterion Means to Your Sampled Data System Design." Analog Devices, Inc., 2009.

<sup>2</sup>Derek Redmayne, Eric Trelewicz, and Alison Smith. "Design Note 1013: Understanding the Effect of Clock Jitter on High Speed ADCs." Linear Technology, 2006.

<sup>3</sup>Pawel Czapor. "Sigma-Delta ADC Clocking—More Than Jitter." *Analog Dialogue*, Vol. 53, No. 2, April 2019.

<sup>4</sup>Michael Clifford. "Fundamental Principles Behind the Sigma-Delta Topology: Part 1." Analog Devices, Inc., January 2016.

<sup>5</sup>Luis Beltran Gil. "Newest Sigma-Delta ADC Architecture Averts Disrupted Data Flow When Synchronizing Critical Distributed Systems." *Analog Dialogue*, Vol. 53, No. 3, September 2019.

<sup>6</sup>Anthony O'Shaughnessy and Petre Minciunescu. "AN-1388: Coherent Sampling for Power Quality Measurements Using the AD7779 24-Bit Simultaneous Sampling Sigma-Delta ADC." Analog Devices, Inc., February 2016.

## About the Authors

Abhilasha Kawle is an analog design manager at Analog Devices in the Linear and Precision Technology Group based in Bangalore, India. She graduated in 2007 from Indian Institute of Science, Bangalore, with a master's degree in electronic design and technology.

Naiqian Ren is an applications engineer with the Precision Converter Technology Group at Analog Devices in Limerick, Ireland. Naiqian joined ADI in 2007 and has a bachelor's degree in electrical engineering from the Dublin City University and a master's degree in VLSI systems from the University of Limerick.

Mayur Anvekar is a digital design manager at Analog Devices in the Linear Precision Technology Group based in Bangalore. Mayur has a master's in embedded systems with nearly 15 years of experience in digital design and verification.

Wasim Shaikh joined Analog Devices in 2015 as an applications engineer in the Precision Converters Department located in Bangalore, India. Wasim received his bachelor's degree from Pune University in 2003.

Roberto Maurino is a design engineer with the Precision ADC Group of Analog Devices in Newbury, UK. He received his Ph.D. from the Imperial College of London in 2005 and his Eng. degree from both the Politecnico di Torino, Italy, and the Institut National Polytechnique de Grenoble, France, in 1996.

Smita Choudhury is a design evaluation manager at Analog Devices in the Linear and Precision Technology Group based in Bangalore, India. She has a master's degree in VLSI design and embedded systems and has been with Analog Devices since 2012.

## Acknowledgements

The authors would like to thank silicon evaluation engineer, Sanjay Kuna, and senior test development engineer, Richard Escoto, for their efforts in testing and proving inherent alias rejection.



HEAD OF WHAT'S POSSIBLE™

For regional headquarters, sales, and distributors or to contact customer service and technical support, visit [analog.com/contact](http://analog.com/contact).

Ask our ADI technology experts tough questions, browse FAQs, or join a conversation at the EngineerZone Online Support Community. Visit [ez.analog.com](http://ez.analog.com).

©2022 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.

VISIT [ANALOG.COM](http://ANALOG.COM)

BR24041-12/22