Vol 53 No 3, September 2019

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

AnalogDialogue

Newest Sigma-Delta ADC Architecture Averts Disrupted Data Flow When Synchronizing Critical Distributed Systems

Lluis Beltran Gil

Abstract

This article introduces the way distributed data acquisition systems have been traditionally synchronized for both SAR ADC-based systems and sigma-delta (Σ-Δ) ADC-based systems, and explores the differences among the two architectures. Additionally, we will discuss the typical inconveniences when trying to synchronize multiple sigma-delta ADCs. Finally, a novel approach based on AD7770's sample rate converter (SRC) is presented, which shows how synchronization is achieved on sigma-delta ADC-based systems without interrupting the data flow.

Introduction

Have you ever imagined yourself in a supersonic aircraft breaking the sound barrier? Since Concorde was retired, this seems like an impossible dream, unless you are a military pilot or an astronaut.

As an electronic engineer, I'm fascinated with how everything works such as, for example, in a cuckoo clock, and how every single system works harmoniously in perfect synchronization with the others.

This applies to every single aspect of our life. As we live in an interconnected world, everything is synchronized—from bank servers to the alarms of our smartphones. The only difference is the magnitude or complexity of the problem to solve in each particular case, the synchronization of different systems vs. the accuracy required—or tolerable error—or the size of the system to synchronize.

Distributed Systems

In a standalone design, the synchronization is inherent to the local clock or oscillator used. But when the standalone design should be integrated into a wider system—let's call it a distributed system—the perspective of the problem changes and the standalone design should be designed accordingly for the use case.

In a system, to calculate the instantaneous power consumption of an electric appliance, both current and voltage must be measured simultaneously.

By performing a quick analysis, you could solve the problem in three different ways:

- Use two synchronized single-channel ADCs to measure current and voltage.
- Use a multichannel simultaneous sampling ADC, either it has one ADC or one sample-and-hold circuit per channel.
- Use one multiplexed ADC and interpolate the measurements, in order to compensate for the time shift between the voltage and the current measurement.

At this point, you may have a solid solution to solve the problem, but let's expand the system requirements from the original single electric appliance to an application where the power must be measured in every single ac power socket in a factory. Now, your original instantaneous power consumption design should be distributed around a factory and somehow designed in such a way that the power in each ac power socket is measured and calculated at the same time.

You are now dealing with a distributed system that consists of a set of subsystems located apart from each other, but closely interrelated. Each subsystem needs to provide data sampled at the exact same point in time, such that the total instantaneous power consumption in the factory can be calculated.

Finally, if we keep expanding the hypothetical application example, imagine that your original design is going to be integrated into your country's power grid. Now, you are sensing millions of power watts, and any failure in a link could have terrible results, like damage to power lines due to stressful conditions that, in turn, may lead to a power outage, with dramatic consequences such as a wildfire or a hospital running without an energy supply.

So, everything needs to be precisely synchronized—that is, the data captured in the power grid shall be captured at the exact same point in time, independent of their geographical situation, as shown in Figure 1.

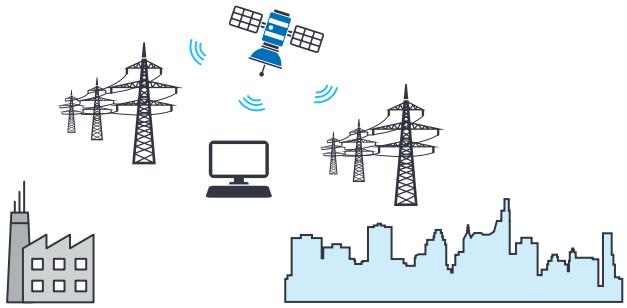

Figure 1. Power grid synchronization.

Under these circumstances, you may consider this as a critical distributed system, and you must get a continuous, fully synchronized stream of data from every single sense node.

Similar to the power grid example, these requirements apply to many other examples of critical distributed systems that may be found within the aerospace or the industry market, among others.

Nyquist and Oversampling ADCs

Before moving into explaining how to synchronize the sampling instant of many ADCs, it is good to understand how each ADC topology determines when to sample the analog input signal, and the benefits and drawbacks of each architecture.

- Nyquist or SAR ADC: This converter's maximum input frequency is dictated by Nyquist, or half-sampling frequency.
- Oversampling or sigma-delta ADC: The maximum input frequency is typically a fraction of the maximum sampling frequency, typically around 0.3×.

On one hand, the SAR ADC's input signal sampling instant is controlled through an external pulse applied to the conversion start pin. If a common conversion start signal is applied to every SAR ADC in the system being synchronized, as shown in Figure 2, they will all trigger the sampling simultaneously on the conversion start signal's edge. By just making sure there are no significant delays between the signals—that is, the conversion start pulses reach every SAR ADC at the same moment in time—the system synchronization can easily be achieved. Note the propagation delay between the pulse reaching the conversion start pin and the actual sampling instant should not vary from device to device, and it is not significant in precision ADCs where the sampling speed is relatively low.

Sometime after the conversion start pulse is applied, which is called conversion time, the conversion result will be available through the digital interface in all ADCs.

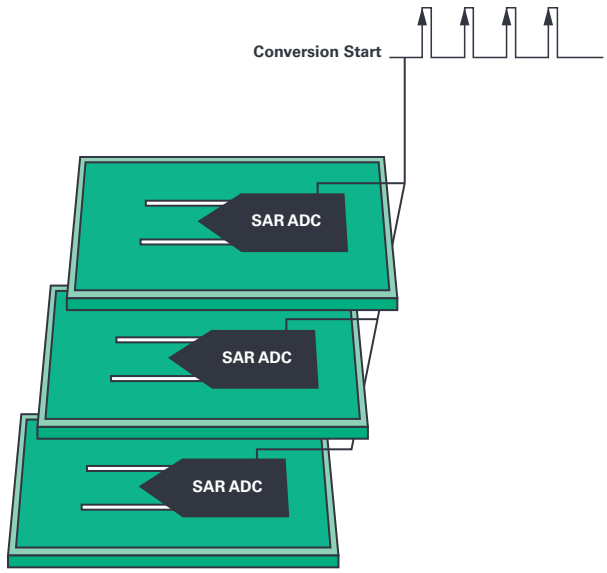

Figure 2. Synchronizing a SAR ADC-based distributed system.

On the other hand, sigma-delta ADC operation is slightly different due to its architecture. In this type of converter, the internal core—the modulator—samples the input signal at a much higher frequency (modulator frequency, f_{MOD}) than the minimum frequency dictated by Nyquist, which is the reason why it is called an oversampling ADC.

By sampling at much higher frequency than strictly needed, an extra number of samples are gathered. All the ADC data is then postprocessed through an average filter for two reasons:

- The noise decreases 1 bit for every 4 averaged samples.
- An average filter transfer function is a low-pass filter. As the sigma-delta architecture pushes its quantization noise toward high frequencies, this shall be removed, as shown in Figure 3. So, this filtering is accomplished by this averaging filter.

The number of samples averaged, known as the decimation ratio (N), dictates the output data rate (ODR) that is the rate at which the ADC provides the conversion results, in samples per second, as indicated in Equation 1. The decimation ratio is typically an integer number with a set of predefined values discretely programmable (that is N = 32, 64, 128, etc.) on the digital filter. So, by keeping f_{MOD} constant, the ODR will be configured depending on the value of N, within the set of predefined values.

$$ODR = \frac{f_{MOD}}{N} \quad (1)$$

The averaging process is typically implemented internally by a sinc filter, and the analogous conversion start pulse for the modulator is generated internally as well, so there is no external control on triggering the conversion process. This type of converter is indeed continuously sampling,

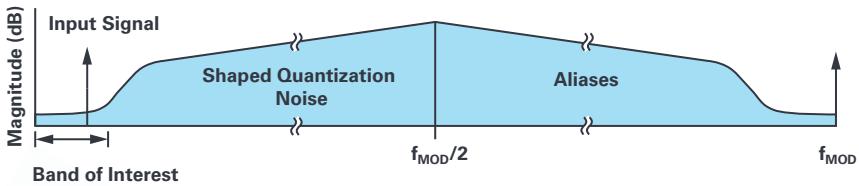

Figure 3. Sigma-delta noise shaping.

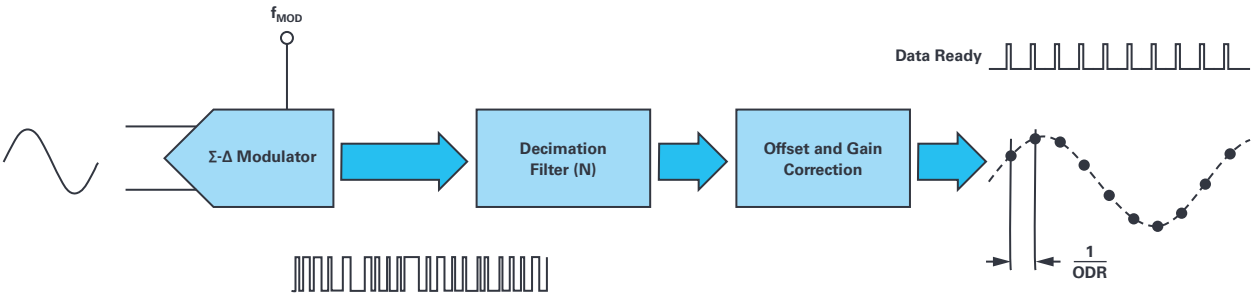

Figure 4. Sigma-delta ADC flow.

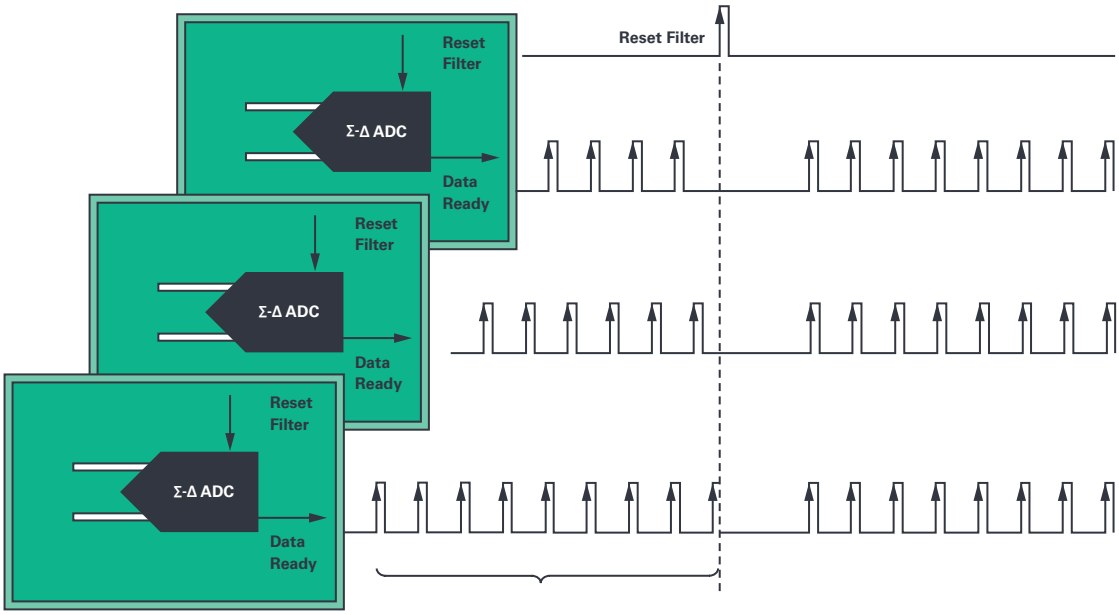

Figure 5. Sigma-delta system reset synchronization.

tracking the input signal, and processing the data acquired. Once the process (sampling and averaging) is completed, the converter generates a data ready signal to indicate to the controller that the data can be read back through the digital interface.

As shown in Figure 4, the flow for a sigma-delta can be summarized in four major steps,

- The modulator samples the signal at f_{MOD} frequency.
- The samples are averaged through the sinc digital filter.
- The result from the sinc filter is offset and gain corrected.
- The data ready pin toggles, indicating that the conversion result is ready to be readback by the controller.

As there is no external control on when the internal sampling triggers, in order to synchronize multiple sigma-delta ADCs within the distributed system, all the digital filters must be reset simultaneously as it is the digital filter that controls the start of the averaged conversion.

Figure 5 shows the effect on the synchronization assuming same ODR, and f_{MOD} in all the sigma-delta ADCs.

Similar to the SAR ADC-based system, it must be ensured here that the reset filter pulse reaches all subsystems at the same time.

However, note that every time the digital filter is reset, the data flow is interrupted as the filter must settle again. The duration of the data disruption in this case depends on the digital filter order, the f_{MOD}, and the decimation rate. An example is shown in Figure 6 where the LPF nature of the filter delays the time until a valid output is generated.

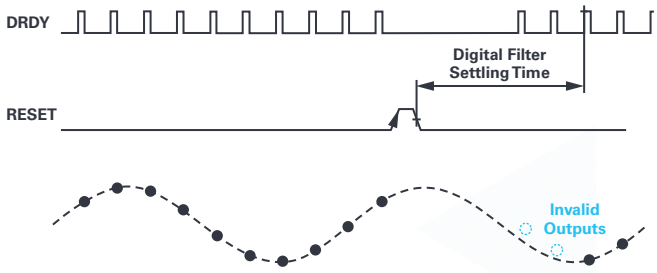

Figure 6. Data disruption due to digital filter settling time.

Implication on Synchronizing the Sampling in Distributed Systems

In a distributed system, the global synchronization signal (let's call it Global_SYNC) is shared across all the modules/subsystems. This synchronization signal could be generated by the master or by a third-party system, like the GPS 1 pps, as shown in Figure 1.

Once the Global_SYNC signal is received, each module must resynchronize the instantaneous sampling of each converter, and most probably its local clock, to guarantee simultaneity.

In a SAR ADC-based distributed system, the resynchronization is intrinsically easy as described in the previous section: the local clock (which manages the conversion start signal) realigns with the Global_SYNC signal, getting these signals in synchrony from then on.

This has an implication in terms of generating frequency spurs because, during the synchronization, there is one sample that has been gathered at different time and distance, as highlighted in blue in Figure 7. In distributed applications these spurs may be acceptable, while disrupting the data flow would have been indeed critical in applications like the power line monitoring mentioned earlier, for instance.

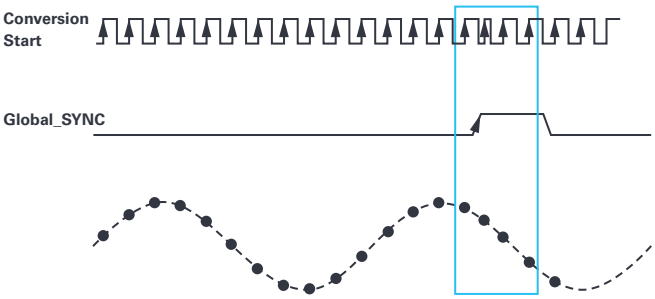

Figure 7. Aligning the SAR ADC conversion process to a global synchronization signal.

In sigma-delta-based distributed systems, the resynchronization with regards to a Global_SYNC signal is a little bit more complicated because the modulator is continuously sampling the analog input signal, and the conversion process is not externally controlled as it is in an SAR ADC.

The easiest way to synchronize multiple sigma-delta-based distributed systems is by resetting the digital filter: all modulator samples gathered and stored to be used on the average filter are dumped, and the digital filter is emptied. That means it will take some time, depending on the digital filter order, to settle its output again, as shown in Figure 5 and Figure 6.

Once the digital filter is settled, it will provide valid conversion data again, but resetting the digital filter on sigma-delta ADC implied data disruption may not be acceptable considering the amount of time the settling takes. The more often the distributed system needs to be resynchronized, the more data flow interruptions there will be, which can make sigma-delta ADCs impractical for critical distributed systems due to the continuous data flow disruption.

Traditionally, a method to minimize the data disruption has been the use of a tunable clock, like a PLL that decreases the error between the global sync and the f_{MOD} frequencies.

Once the Global_SYNC pulse is received, the uncertainty between the sigma-delta ADC conversion start and the Global_SYNC pulse can be calculated following a process similar to:

- The controller calculates the time difference between the sampling instant (calculated backwards from the data ready signal by knowing the group delay, as shown in Figure 8) and the Global_SYNC pulse. The group delay is a data sheet specification that accounts for the time between when the input is sampled until the data ready pin toggles, indicating that the sample is ready to be read.
- If there is a discrepancy between the sampling instant and Global_SYNC, the local controller quantifies the time difference (t_{ahead} or $t_{delayed}$), as shown in Figure 9.

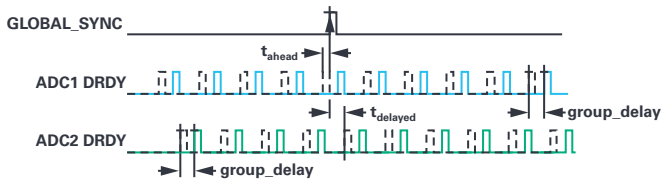

Figure 9. Quantifying the discrepancy from each ADC's sampling instant (provided the group delay is known) to the global synchronization signal.

- If there is a discrepancy, the sigma-delta filter could be reset or the f_{MOD} could be modified in order to adjust the sigma-delta sampling during a few samples. In both cases, a few samples would be missed. Note that by changing the local clock frequency (f_{MOD}), the sigma-delta ADC is modifying its output data rate (ODR = f_{MOD}/N), such that the ADC samples its analog input either slower or faster, with the intention for this ADC to catch up the rest of the ADCs on the system and with the Global_SYNC.
- If the f_{MOD} is updated, once in synchrony, the master clock frequency is reverted back to original frequency to return to previous ODR, while the subsystem becomes synchronous from then on.

This process of changing f_{MOD} during a certain period of time is shown in Figure 10.

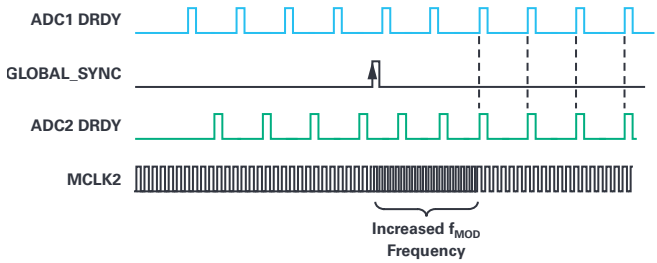

Figure 10. Synchronization method using PLL to tune the modulator frequency.

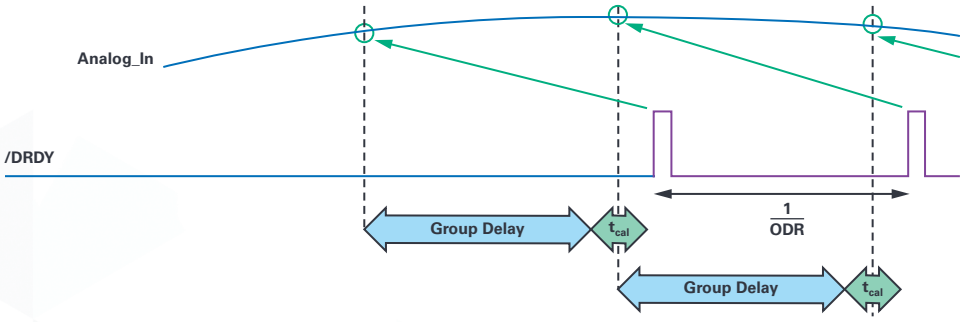

Figure 8. Time delay between the analog input being sampled and the data ready toggling.

This method may not be appropriated in some cases as there are a couple of details to consider:

- Changing the modulator frequency to non-multiple values may be unpractical.
- If fine frequency tuning is possible, the frequency steps while changing must be small, otherwise the digital filter may go out of bounds, so the lead time for the synchronization routine becomes longer.
- If the ODR change needed is big enough, it could be solved by changing the decimation rate (N) instead of the modulator frequency (f_{MOD}), but this would also imply missing samples.
- Using a PLL implies an extra amount of power being consumed plus its own settling time until reaching the desired modulator frequency.

In general, the overall system complexity and cost increase, scaling up with the system size, especially compared to SAR ADCs where this problem was solved more easily by just aligning the conversion start to the Global_SYNC signal. In addition, in many cases the use of a sigma-delta ADC must be not possible as per the system restrictions and limitations explained above.

Easily Resynchronize Sigma-Delta ADCs with No Data Disruption

The AD7770 family (which includes AD7770, AD7771, and AD7779) has a built-in SRC. With the introduction of this novel architecture, the restriction of the decimation factor (N) being a fixed value is no longer valid.

The SRC allows you to program decimal numbers, not only integers, as the decimation rate (N), which allows you to program any desired output data rate. On the previous synchronization method, as N was fixed, the external clock needed to change to adjust f_{MOD} in order to perform the synchronization routine.

Using AD7770 family of products, as the N is flexibly programmable and reprogrammable on-the-fly, any ODR can be programmed without having to change f_{MOD} and without data disruption.

This novel method for resynchronizing sigma-delta-based subsystems, making use of the SRC simplifies the resynchronization, minimizing the complexities described in the previous sections.

The new method is as follows:

- When the Global_SYNC signal is received, each subsystem checks if it is sampling synchronously or not, taking the data ready signal as a reference and using the group delay to find the actual sampling instant.
- If there is a discrepancy between the sampling instant and when the Global_SYNC signal has been received, the local controller quantifies the time difference (t_{ahead} or $t_{delayed}$) as shown in Figure 9.
- A new ODR is programmed to temporarily generate a faster or slower ODR through modifying the decimation factor (N) by means of the SRC. The whole operation of resynchronization will always take 4 samples (or 6 if the sinc5 filter is enabled on AD7771), but without interrupting the data flow due to these samples which are still valid and 100% settled.
- Once the required amount of DRDY has been received, the decimation factor is reprogrammed again to return to the desired ODR, which guarantees that the sigma-delta converter is synchronized with the rest of subsystems, as shown in Figure 11, with no data flow disruption.

Conclusions

Critical distributed systems require synchronous conversion in all the subsystems and a continuous data flow.

SAR converters provide an intuitive way to resynchronize the sampling by readjusting and aligning the conversion start signal with the Global_SYNC pulse.

In applications that require high dynamic range (DR) or signal-to-noise ratio (SNR) specifications, SAR is not an option, but traditional sigma-delta converters become a challenge to use due to their inflexibility to readjust without disrupting the data flow.

As seen in the example, the SRC provides a seamless synchronization routine with minimum latency and much lower cost and complexity than other solutions.

There are plenty of applications where the SRC can also be advantageous. As with the power line monitoring example, any line frequency change can be compensated for by immediately changing the decimation rate on the fly. That way, the power line is always sampled at a coherent sampling frequency. Here, in critical distributed systems, it's demonstrated that the SRC can also be very useful for resynchronizing the system without having to interrupt the data flow and with no need for extra components like PLLs. The AD7770 solves the traditional problem of synchronizing sigma-delta ADC-based distributed systems, without missing samples and avoiding the added cost and complexity of the PLL-based method.

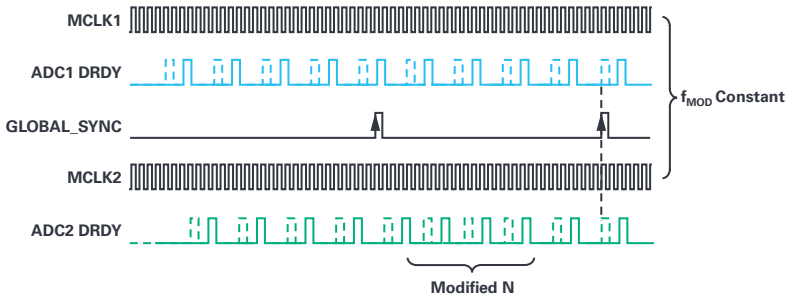

Figure 11. The sample rate converter adjusts the ODR on the fly in order to resynchronize the sampling on all devices.

About the Author

Lluis Beltran Gil received his B.S. degree in electronics engineering in 2009 and in industrial engineering in 2012, both from the Universitat Politècnica de València, UPV (Technical University of Valencia). After graduation, Lluis joined Analog Devices in 2013 as an applications engineer in the Precision Converter Group in Limerick, supporting temperature sensors. Currently, Lluis is working on the SAR ADC Applications Team within the Precision Converters Group, and he is based in Valencia, Spain. He can be reached at *lluis.beltrangil@analog.com*.