

APPLICATION NOTE 3494

The Basics of Anti-Aliasing: Using Switched-Capacitor Filters

Abstract: Aliasing due to unwanted, spurious out-of-band signals is a problem in many applications that use A/D converters. These signals, if not filtered properly, can seriously impact the performance of a data-conversion system. This application note discusses the principles of anti-aliasing and its effect on system performance. It also offers advice, via an elegant and cost-effective, complete-system example, on using an integrated switched-capacitor device for this important function. It covers in detail all the important performance parameters and practical issues associated with designing a high-performance system.

What causes aliasing? The effect is explained in the Nyquist criterion. This criterion says that a time-continuous signal needs to be sampled more than twice per period if converted into a time-discrete signal. If the signal is not sampled often enough, information is irretrievably lost. This problem can be seen in **Figure 1**. If a signal is sampled once per period, a DC signal (with arbitrary amplitude) is generated (Figure 1a); if it is sampled exactly twice per period, a square wave is generated (Figure 1b). It is important to note that sampling at exactly twice the period of the input signal is a very special case and should always be avoided. Figure 1c shows a 190kHz signal sampled at 200kHz. The result is a perfect sine wave, but with a completely wrong frequency. This frequency change is caused by aliasing.

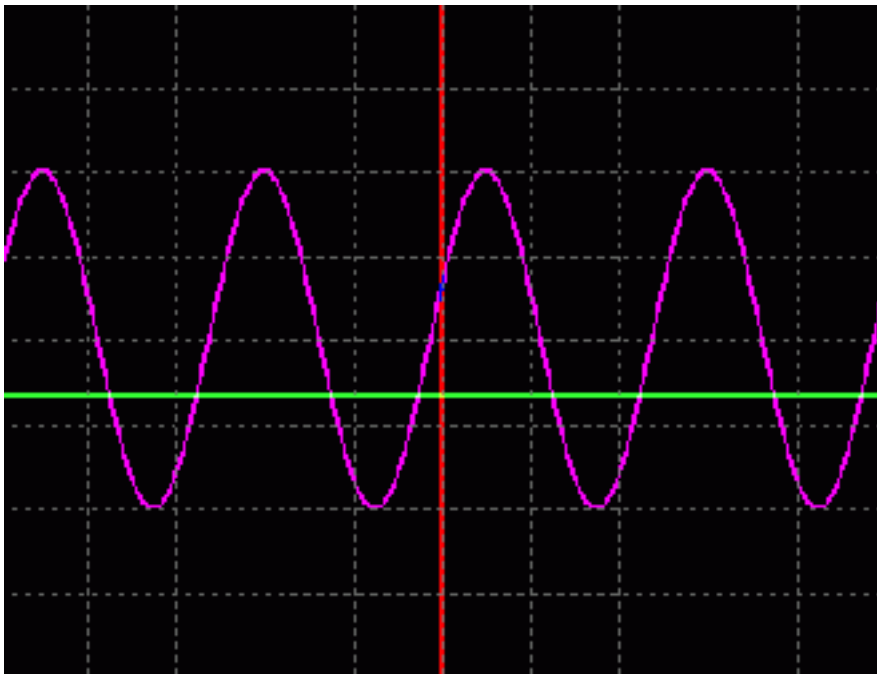


Figure 1a. A sine wave sampled only once per period results in a DC signal with arbitrary amplitude.



Figure 1b. The same sine wave sampled exactly twice per period generates a square wave—amplitude information is lost.

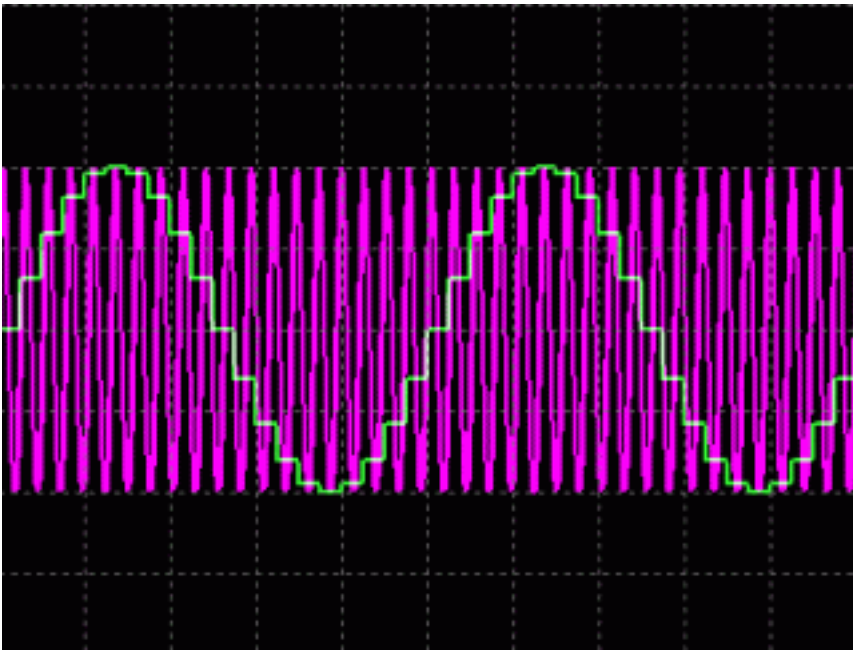


Figure 1c. $f_{\text{signal}} = 190 \text{ kHz}$, $f_s = 200 \text{ kHz}$ is an undersampled signal, resulting in aliasing.

Figure 2 shows this phenomenon in the frequency domain, where we can see that signals higher than $f \geq f_s/2$ are mirrored at $f_s/2$. The only possibility to avoid this is to ensure that no signals at higher frequencies are present in the signal. That means we have to know the highest frequency in the signal of interest. The sampling frequency needs to be larger than two times this frequency. An initial thought by some might be to do that in the digital domain, but this will not work because it is impossible to remove unwanted frequencies from the signal once it has been sampled and the frequencies are aliased into the band of interest. Anti-aliasing must be done in the analog domain, before the signal is sampled.

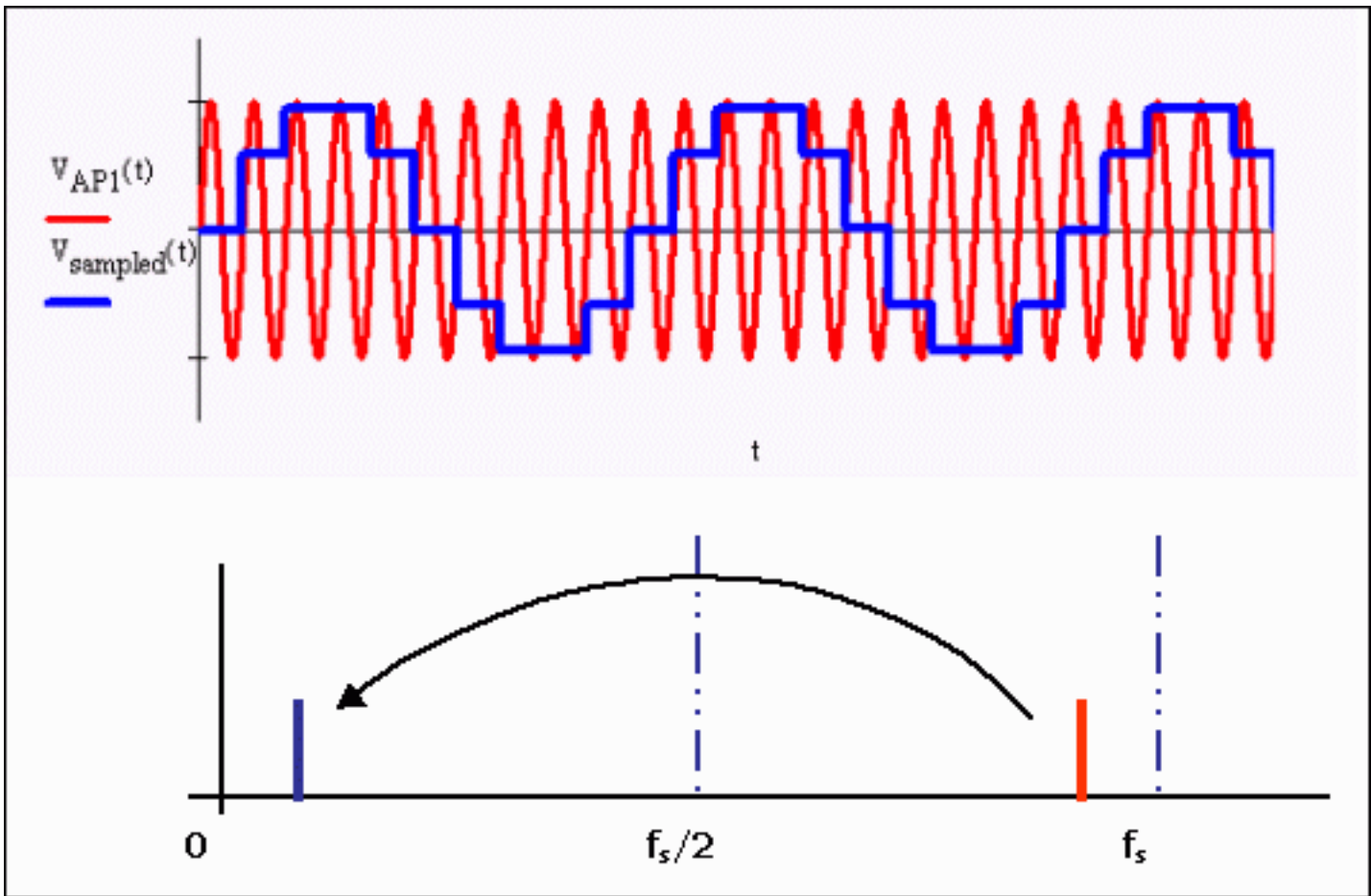


Figure 2. Aliasing in the frequency domain. A undersampled signal is mirrored at $f_s/2$.

The next step is to design the anti-aliasing filter. This involves first deciding on the desired filter characteristics (cutoff frequency, transition band steepness, etc.) then selecting the best filter approximation (sometimes called type) to meet the requirements. In general, it is valid to say that the more oversampling (higher sampling frequency) the easier the filter can be. However, oversampling is expensive, as faster ADCs are needed.

For an example, let's start with the following assumption: Oversampling by a factor of eight, i.e., the sampling frequency is eight times the highest signal frequency. This is a good compromise between ADC cost and necessary filter complexity. Further, we will assume an ADC resolution of 14 bits, with a signal-to-noise ratio (SNR) of 80dB. At one-half the sampling frequency (in this example, 4x signal frequency), the lowpass filter needs to reach 80dB of attenuation, to ensure that all possible error signals are attenuated well enough to not show up in the sampled signal. That means a transition band of 40dB per octave. This means that a high-order analog filter will be needed. The requirement can be met with a 7th-order Butterworth filter, but that might not be the best choice, depending on the application. Different filter approximations are suited to different applications. **Figure 3** shows a Butterworth, a Chebyshev, and an elliptic filter response. In this graph it can be observed that these approximations show different passbands and transition bands. Comparing the elliptic to the Butterworth filter shows that the elliptic will result in a much steeper transition band, but it comes at the expense of worse phase behavior. Which approximation is optimal has to be determined in every case—for an audio application, at least a Butterworth filter (if not a Bessel) is needed, while for generic data acquisition, with less demand for phase accuracy, a Chebyshev, or even an elliptic filter might be sufficient.

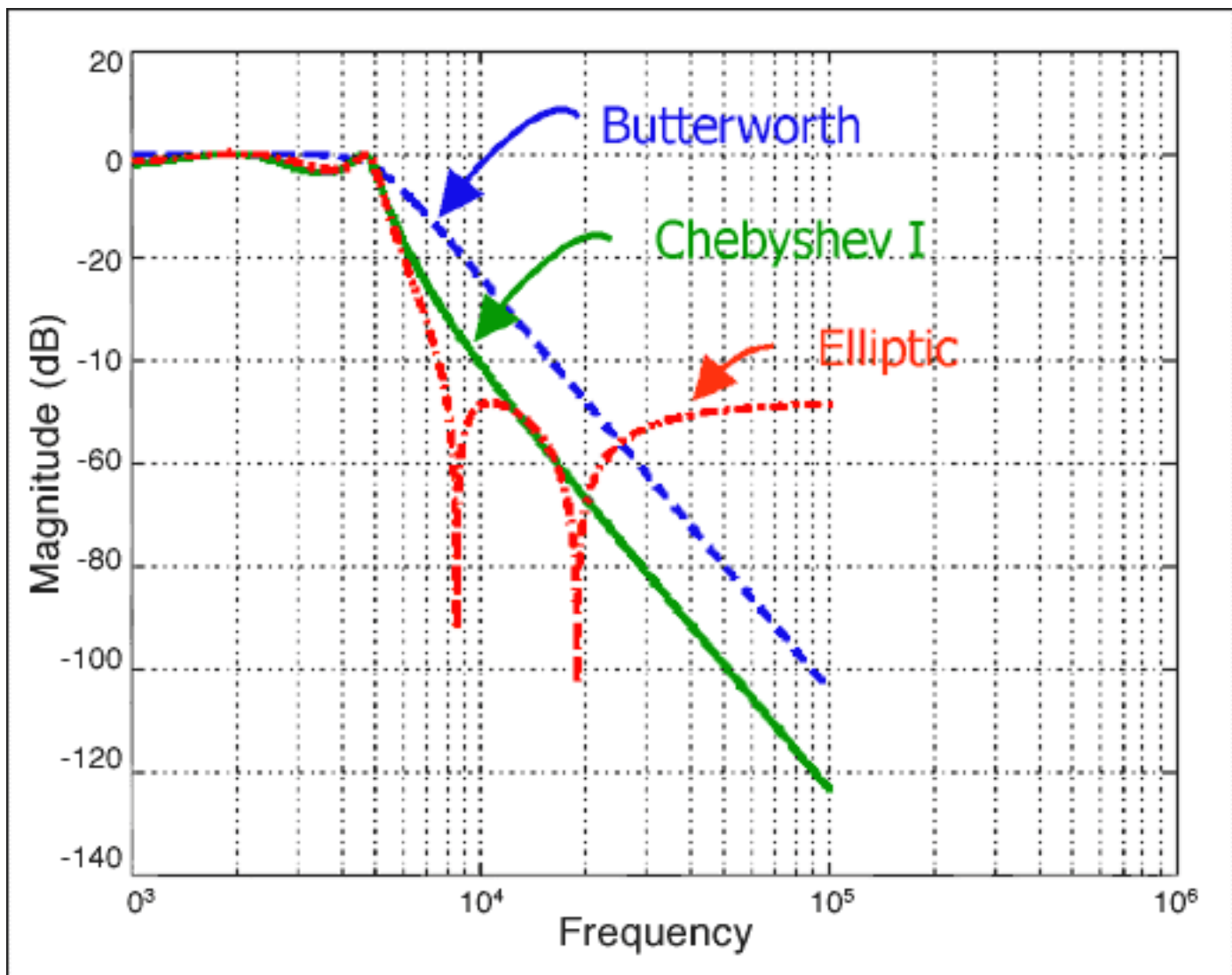


Figure 3. Comparison of different filter approximations.

There are several types of popular A/D converters: SAR (successive approximation), typically used for medium-fast applications; flash, typically used for fast to very-fast applications; and delta sigma (also called sigma delta), typically used for slow applications. They all have standard anti-aliasing requirements determined by the conversion rate and the desired input bandwidth, with the exception being the delta-sigma converter. This type of converter uses a conversion technique that utilizes a very high input sampling and conversion rate, followed by a digital decimation filter that reduces the effective throughput rate. This has the effect of increased resolution (dynamic range). The anti-aliasing requirements for the delta-sigma converter are determined from the ratio of the input sampling rate to the highest frequency of interest in the input signal. This reduction in the requirements is the same as simply oversampling any data converter; typically, a simple RC can be used. The tradeoff of a simpler anti-aliasing filter comes at the cost of longer propagation delays, which can make such converters very difficult to use in closed control-loop or multiplexed applications.

Many designers have an aversion to analog filters because they are difficult to design, have large tolerances and are difficult to manufacture, especially with the tight specs needed in our example. A good rule of thumb is to assume that tolerances of discrete components multiply. That means that with standard off-the-shelf resistors and capacitors, a large error in the corner frequency and transition band has to be expected. A good alternative to solve this problem is to look for integrated off-the-shelf solutions for these filters, which are available from IC manufacturers such as Maxim.

There are two types of integrated filters, continuous time and switched-capacitor filters. The continuous-time filters usually need some external components to adjust for corner frequency, making them limited in their flexibility. Switched capacitors can, due to their architecture, be very flexible. If used properly, they can be an excellent alternative to both discrete and integrated continuous-time filters.

The switched capacitor is a long-known architecture that can be reliably and repeatably integrated into today's

silicon technologies. Its functional principle and some mathematical basics are shown in **Figure 4**. The operation is as follows: Capacitors are charged and discharged by periodically opening and closing the switches located on either side of the capacitor. This generates a charge transfer that results in a pulsing current flow. The average current can be calculated and, if the switching frequency is high enough, this current will be equivalent to a current through a resistor. In basic terms, this means that a resistor is replaced by a capacitor. The amount of current—and indirectly the resistor value—depends on two variables: the size of the capacitor, and the switching frequency. The higher the frequency or larger the capacitor, the higher the current or conversely, the lower the resistor value. If a filter is built using this architecture, the frequency behavior can be changed by changing either the capacitor size or the switching frequency. In an integrated solution, the capacitor value is fixed, so the filter characteristic must be modified by changing the switching frequency. The schematic for such a filter is shown in **Figure 5**.

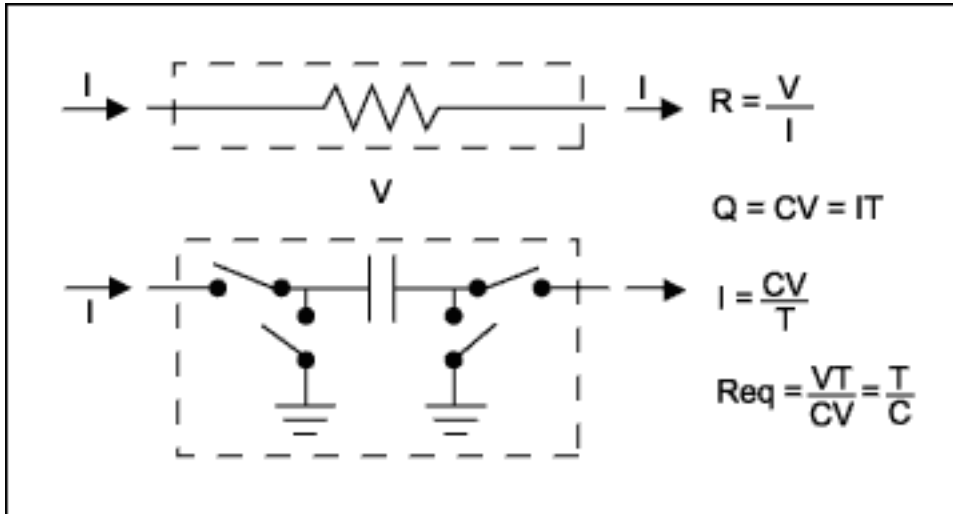


Figure 4. Switched capacitor schematic diagram.

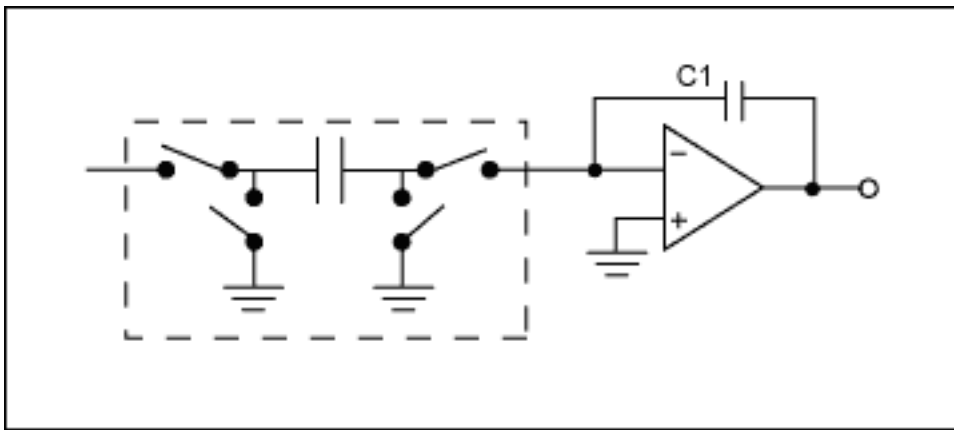


Figure 5. A simple filter realized using switched-capacitor techniques.

The accuracy of a filter is based on the tolerances of the individual components. In the case of a discrete filter, we can only use unmatched components, while in an integrated solution the matching of these components is extremely good (in the range of 0.1%). So we can expect to get very good control of the characteristics of integrated filters. For example, the MAX7490 specifies the corner frequency accuracy as 0.2%, a value that cannot be achieved with discrete off-the-shelf components. Also, temperature drift is excellent, e.g., 10 ppm/°C.

It is important to point out here that a switch-capacitor solution samples the signal. It converts the time-continuous signal to a time-discrete signal, which means that we again have to consider aliasing. The good news is that the sampling rate in such filters is very high, it is usually 100x oversampled. This means that a very simple anti-aliasing filter—a single resistor and capacitor—can be used. Another issue with such systems is the fact that, similarly as with ADCs, a phase jitter in the switching clock generates distortion. **Figure 6** shows the right signal sampled at the wrong time, generating amplitude errors.

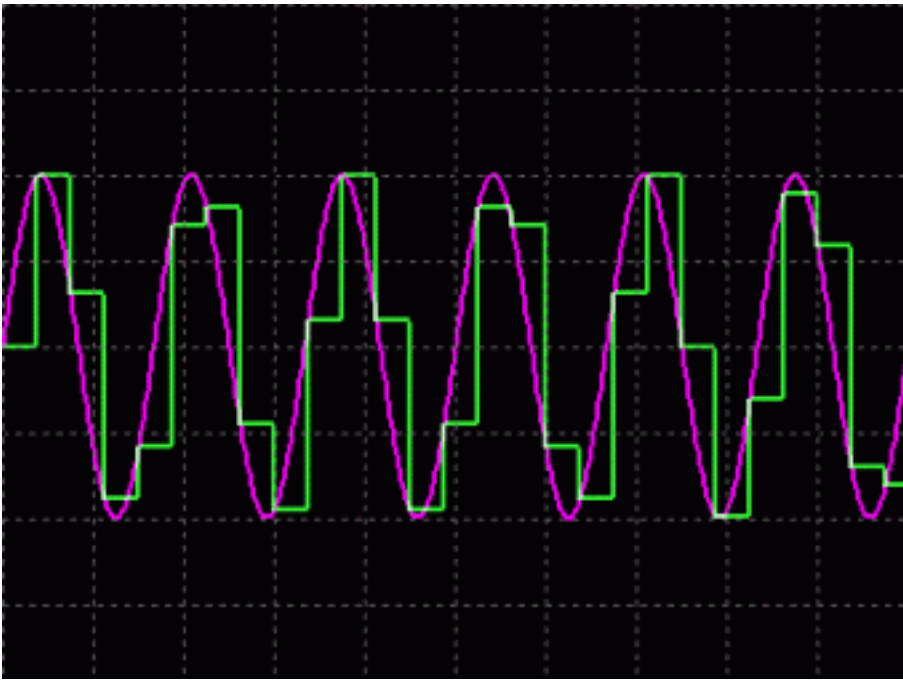


Figure 6. Amplitude distortion caused by undersampling.

Clock jitter can manifest itself in two ways. If the phase error has a random distribution, the noise floor rises; if the jitter is periodical, distortion (THD) rises. Jitter is specified as a time quality, e.g., ps peak/peak or RMS. How large can jitter be, to achieve a certain signal purity? Investigations (Reference 1) show that, in a 16-bit system, a clock jitter of one nspp (peak/peak) reduces the SNR from 98dB to 91dB. To limit the influence of jitter in such a system to 0.5dB, the jitter cannot be larger than 400ps_{pp}.

Such clock signals can easily be created with off-the-shelf clock oscillators, like the SaRonix NTH5 with 8ps_{RMS} (53ps_{pp}). The downside of this solution is that we are limited to one single frequency. In most systems, there are other components, like an ADC, a μ C, etc., which also need to be supplied with a clock. If these clocks are generated with additional oscillators, all these clocks will be asynchronous, generating all sorts of problems. ICs like the MAX7375 or DS1085 can generate several clocks, all synchronized to each other, with good enough jitter (160ps and 300ps, respectively) to achieve 90+ dB signal purity. Another advantage using these devices to generate the clocks is the fact that they can be programmed to different frequencies. This means that we get the capability to program via software the frequency behavior of the analog filter. We can create a very flexible system.

Figure 7 shows an implementation of a data-acquisition system, based on the constraints discussed above. An ADC with 14-bit resolution and a maximum sampling rate of 200ksps (MAX1067 family) is used. This device needs at least 24 clocks per conversion. For the anti-aliasing filter, a member of the MAX7418-21 family is used. Within this family, different variants with different filter approximations, like Bessel, Butterworth, or elliptic can be found. The corner frequency is set to $1/100 f_{CLK}$. Considering the DSP takes control of the sampling of the converter (only one timer is used for that function), the same clock can be used for both the filter and the ADC, giving perfect synchronization. The clock is generated using the DS1085, which can generate two clocks; the second clock can be used for the DSP. The DS1085 can be programmed through a two-wire interface, providing the possibility of reprogramming the system during operation to different sampling rates, or only once, to achieve different functions with only one PCB.

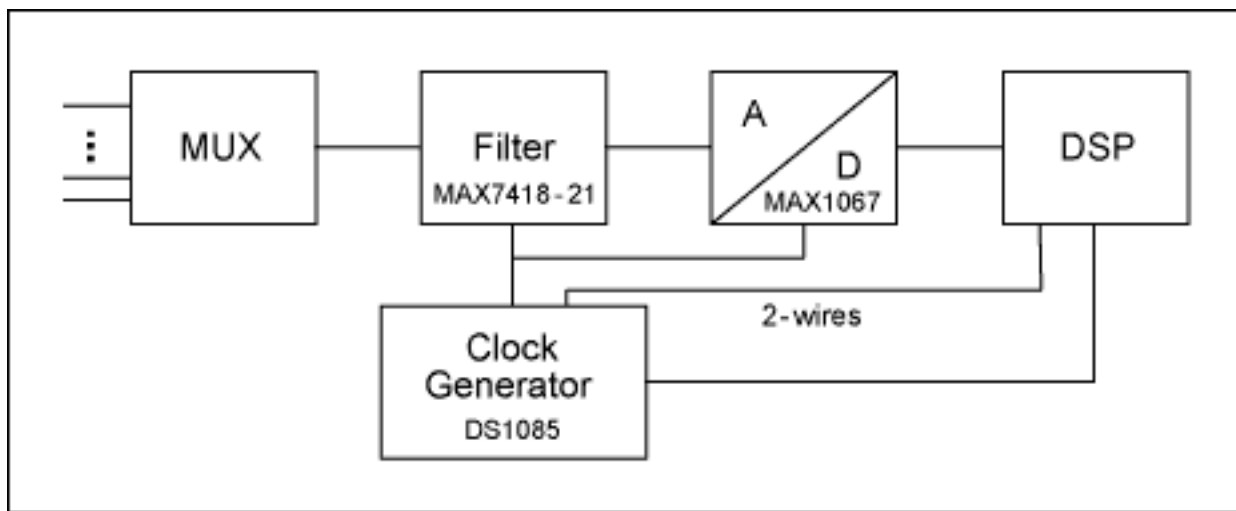


Figure 7. Schematic diagram of the realized data-acquisition system.

The system shown in Figure 7 is designed with devices coordinated with each other. Cost and complexity are in an acceptable range. An additional advantage is the programmability, which creates immense flexibility, saving both design and logistical costs.

References

1. The Effects of Sampling Clock Jitter on Nyquist Sampling Analog-to-Digital Converters and on Oversampling Delta-Sigma ADCs
 July 1990 JAES: AES preprint# 2844
 By Steven Harris

Also see

- [Maxim/Dallas Glossary definition of *aliasing*](#)
- [Maxim/Dallas Glossary definition of *anti-aliasing*](#)
- [Maxim/Dallas Glossary definition of *Nyquist*](#)
- [Application note 928/Tutorial: Filter Basics: Anti-Aliasing](#)

Application Note 3494: www.maxim-ic.com/an3494

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest?
[Sign up for EE-Mail™.](#)

Related Parts

DS1085: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX1067: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX7375: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAX7418: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX7419: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX7420: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX7421: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX7490: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN3494, AN 3494, APP3494, Appnote3494, Appnote 3494

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal