From analog to digital

J. Belleman
CERN, Geneva, Switzerland

Abstract
Analog-to-digital conversion and its reverse, digital-to-analog conversion, are ubiquitous in all modern electronics, from instrumentation and telecommunication equipment to computers and entertainment. We shall explore the consequences of converting signals between the analog and digital domains and give an overview of the internal architecture and operation of a number of converter types. The importance of analog input and clock signal integrity will be explained and methods to prevent or mitigate the effects of interference will be shown. Examples will be drawn from several manufacturers' datasheets.

1 Introduction

Since the 1970s, the supremacy of numerical processing over analog signal treatment has become ever more evident. More and more functions that traditionally were in the analog realm are being replaced by digital electronic hardware. Yet before any numerical signal treatment is possible, the analog nature of most electrical signals must first be converted into a numerical representation. That task falls to an analog circuit that is not likely to disappear for some time to come: the Analog-to-Digital Converter (ADC).

The basic function of an ADC is to convert a voltage applied to its input into a number with a limited range of possible values. On top of that, it will only do so at a finite rate. Thus, it replaces the continuous input signal by a sequence of numbers, with discrete steps in both amplitude and time.

A huge variety of ADCs exist, using many different architectures and covering a wide range of resolution and speed.

1.1 Amplitude quantization

In order to make a signal suitable for treatment by numerical circuitry, it must first be represented in a numerical format, or quantized. That is, a continuous range of values is replaced by a limited set of values separated by discrete steps (Fig. 1).

Usually the number of steps is chosen to be a power of two, because that yields the most economical representation in binary digital electronics. Naturally, the quality of the approximation depends on the number of steps used to approximate the original signal.

It is customary to specify the accuracy of an ADC by comparing the power of an ideal full-scale sinusoidal input signal with its numerical representation, because it is relatively straightforward to produce a very clean sinewave signal.

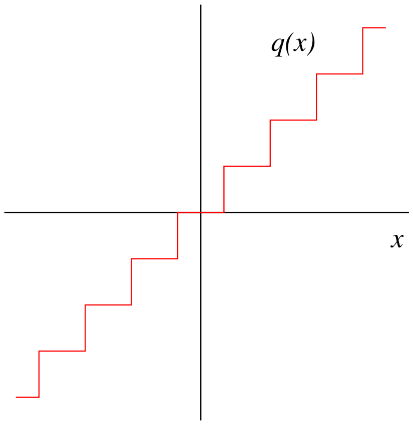


Fig. 1: Quantization function

Consider the example (Fig. 2), where a sinusoidal signal is compared with its numerical representation expressed in 2^n equidistant discrete steps, such that the difference between the sine and its quantized approximation is never greater than one half of the size of a step. This difference is referred to as the quantization error (ε).

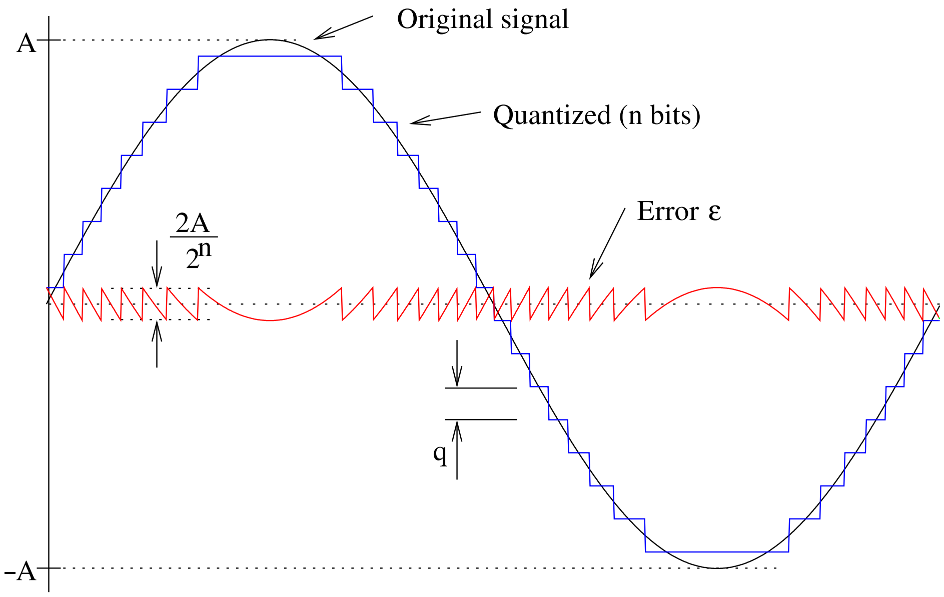


Fig. 2: Original signal, quantized signal, and quantization error

The mean power in a sinusoidal input signal with amplitude A is simply

$$P_s = \frac{A^2}{T}\int_0^T \sin^2 \omega t \mathrm{d}t = \frac{A^2}{2} \ . \tag{1}$$

Quantized to 2^n discrete levels, equivalent to n bits, one quantization step q has a size of

$$q = \frac{2A}{2^n} = A \cdot 2^{-(n-1)} \ . \tag{2}$$

We have arranged things so that the quantization error has a maximum value of plus or minus half the step size, so

$$|\varepsilon| \leq \frac{q}{2} \text{ and therefore, } \ |\varepsilon| \leq A \cdot 2^{-n} \ . \tag{3}$$

For a large enough number of quantization steps, the probability density function of the quantization error tends toward being flat [1]. Its value within the quantization error bounds is

$$p(\varepsilon) \approx \frac{1}{q} \ . \tag{4}$$

So we can calculate its mean power or variance as the 2^{nd} moment of its distribution:

$$P_\varepsilon = \int_{-q/2}^{q/2} p(\varepsilon)\varepsilon^2 \mathrm{d}\varepsilon \approx \frac{A^2 \cdot 2^{-2n}}{3} \ . \tag{5}$$

Thus, an ideal ADC would have a signal-to-noise ratio

$$\text{SNR} = \frac{P_s}{P_\varepsilon} = 1.5 \cdot 2^{2n} \ , \tag{6}$$

or, expressed in decibels, $1.76 + 6.02n$ dB [1]. This is an expression that appears in many texts on ADCs, and it is the *best* an n-bit ADC can do. It serves to set a standard against which to compare the performance of a real ADC. Usually, a real ADC performs substantially worse than that.

[1] Decibels: See Appendix A.

By measuring the actual signal-to-noise ratio of an ADC, and solving for n in Eq. (6), one can determine the Effective Number Of Bits (ENOB) of an ADC. This number is also often specified in ADC datasheets.

The quantization error is a distortion of the original signal. It is an irreversible loss of detail. Contrary to analog circuitry, in which distortion usually decreases for lower signal levels, the quantization error is comparatively worse for small signals. It is therefore important to scale the signal amplitude so as to fill the ADC range as completely as possible without running into saturation.

1.2 Quantization in the time domain

An ADC delivers values that correspond to discrete instants in time. Provided the sampling rate is at least twice the bandwidth of the signal and neglecting, for the moment, the effect of the finite resolution of the amplitude quantization, the information contained in the sample stream is sufficient to restore the original signal without loss of information. This was worked out in the first half of the twentieth century by E. Whittaker, V. Kotelnikov, H. Nyquist, R. Hartley and C. Shannon [2–6].

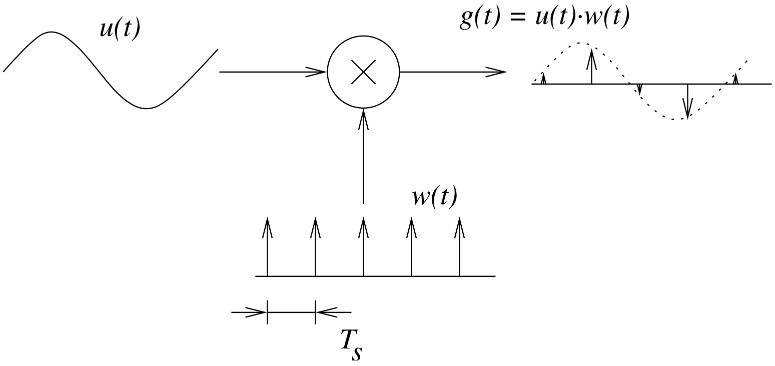


Fig. 3: The sampling process

Conceptually, sampling is a modulation process (Fig. 3). The input waveform $u(t)$ is multiplied by the modulating or sampling waveform $w(t)$. The modulating waveform is a sequence of equally spaced Dirac δ impulses:

$$w(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{7}$$

where T_s is the sampling period. Its reciprocal $1/T_s$ is the sampling frequency F_s. The output of the modulation process, $g(t)$, consists of a sequence of impulses with varying 'amplitude', according to the value of the input signal at each sampling instant. The value $u(t)$ taken in between the sampling instants does not affect the result. Only the values at the sampling instants matter.

$$g(t) = \sum_{n=-\infty}^{\infty} u(t) \cdot \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} u(nT_s) \cdot \delta(t - nT_s)\ . \tag{8}$$

Let us examine the properties of $g(t)$ in the frequency domain. To do so, we shall first derive the frequency-domain representation $W(f)$ of the modulating waveform $w(t)$:

$$W(f) = \mathscr{F}\{w(t)\} = \int_{-\infty}^{\infty} w(t)\mathrm{e}^{-j2\pi ft}\mathrm{d}t = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - nT_s)\mathrm{e}^{-j2\pi ft}\mathrm{d}t\ . \tag{9}$$

Integrating each term of the series separately, we obtain

$$W(f) = \sum_{n=-\infty}^{\infty} \mathrm{e}^{-j2\pi n f T_s} = 1 + 2\sum_{n=1}^{\infty} \cos 2\pi n f T_s = \sum_{n=-\infty}^{\infty} \delta(f - nF_s)\ . \tag{10}$$

This shows three different perspectives of the same expression. The spectrum of the sampling waveform turns out to be a repetition of spectral lines at multiples of the sampling frequency F_s.

In order to get the spectrum of the sampled signal $G(f)$, we can apply convolution:

$$\begin{aligned} G(f) &= U(f) * W(f) && (11)\\ &= \int_{-\infty}^{\infty} U(\phi) W(f-\phi)\mathrm{d}\phi && (12)\\ &= \int_{-\infty}^{\infty} U(\phi) \sum_{n=-\infty}^{\infty} \delta(f - nF_s - \phi)\mathrm{d}\phi && (13)\\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} U(\phi)\cdot\delta(f - nF_s - \phi)\mathrm{d}\phi\ . && (14)\\ G(f) &= \sum_{n=-\infty}^{\infty} U(f - nF_s)\ . && (15)\end{aligned}$$

Equation (15) shows that the spectrum of the original signal is repeated at all harmonics of the sampling waveform (Fig. 4). If the spectrum of the original signal $U(f)$ extends beyond $F_s/2$, adjacent sidebands would overlap and it would no longer be possible to tell to which image a given frequency in the spectrum belongs. Adjacent images get mixed up inseparably. This limit is known as the Nyquist criterion [4, 6]. We shall subsequently refer to $F_s/2$ as the *Nyquist frequency* F_N.

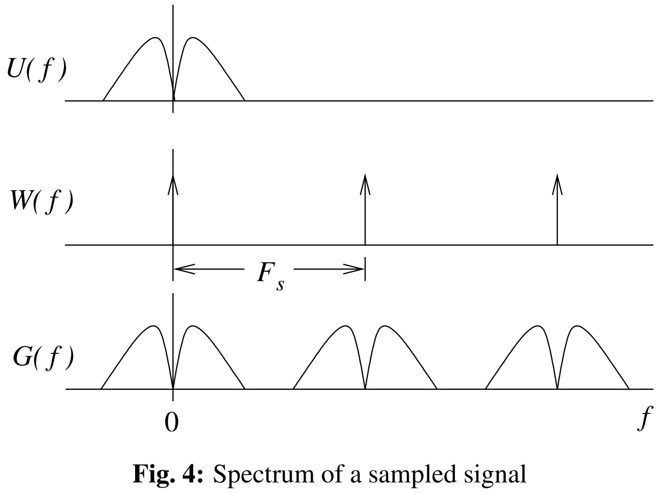


Fig. 4: Spectrum of a sampled signal

Note that Nyquist's criterion sets a limit on the *bandwidth* of a signal to be sampled; It says nothing of its frequency. Indeed, it is quite possible to sample a band-limited signal at a rate much slower than its actual frequency, while still keeping the spectral images separate. Consider for example a signal as depicted by $u(t)$ in Fig. 5. This signal has frequencies greater than the sampling rate. Its spectrum might look like $U(f)$ in Fig. 6. Let us describe it as frequency shifted by $m \cdot F_s$, with integer m. To get the spectrum of the sampled signal, we again apply convolution with the spectrum of the sampling waveform, as in Eq. (12), replacing $U(f)$ with $U(f + mF_s)$:

$$G(f) = U(f + mF_s) * W(f) = \int_{-\infty}^{\infty} U(\phi + mF_s) W(f - \phi)\mathrm{d}\phi\ . \tag{16}$$

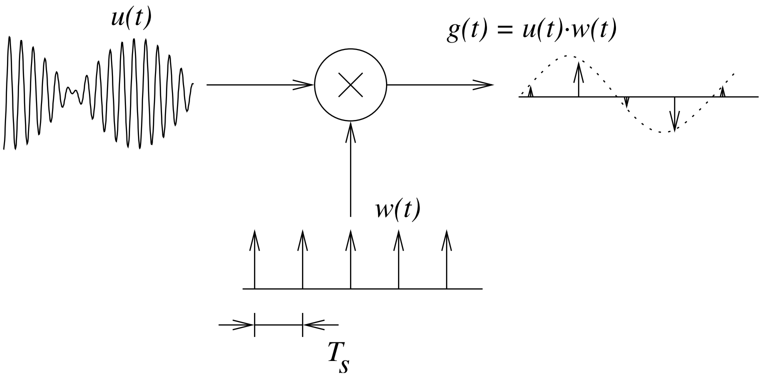


Fig. 5: Sub-sampling

$$G(f) = \sum_{n=-\infty}^{\infty} U\big(f + (m-n)F_s\big) \ . \tag{17}$$

Since the sum over n runs from $-\infty$ to ∞, we may add an arbitrary integer constant to n without affecting the value of the result, so let us choose $-m$. The resulting expression for the spectrum of the sampled signal is

$$G(f) = \sum_{n=-\infty}^{\infty} U(f - nF_s) \ , \tag{18}$$

which is exactly the same as in Eq. (15), above.

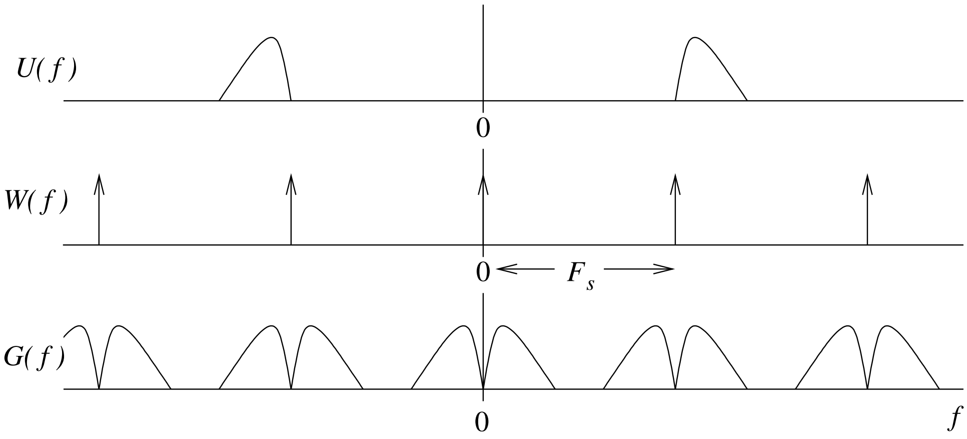


Fig. 6: Spectra with sub-sampling

The spectrum of the sampled signal does not change if the sampled signal is shifted by some integer times F_s. The practice of sampling a signal at a sampling frequency below the frequencies contained in the signal is called *sub-sampling*. Provided the signal bandwidth is less than half the sampling frequency, it is possible to reconstruct the original signal without loss of information. All the information of the signal is present in every spectral image. The multiple copying of frequencies in the spectrum of a sampled signal is known as *aliasing*.

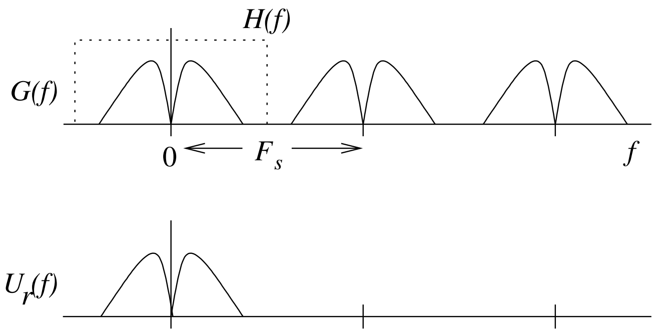


Fig. 7: Signal reconstruction by filtering

1.3 Signal reconstruction

To recover the original continuous signal, conceptually, we can filter the sampled signal using a filter $H(f)$ with a rectangular passband. In Fig. 7, we have chosen the baseband, which is the image immediately around $f = 0$, so $H(f) = 1$ for $-F_s/2 < f < F_s/2$ and zero elsewhere. Filtering is a multiplication in the frequency domain, and a convolution in the time domain. So, to obtain the expression for the reconstructed signal $u_r(t)$, we first get the inverse Fourier transform of the filter function $H(f)$ and then convolve that with the expression for the sample sequence:

$$h(t) = \mathscr{F}^{-1}\{H(f)\} = \int_{-\infty}^{\infty} H(f)\mathrm{e}^{j2\pi ft}\mathrm{d}f = \int_{-F_s/2}^{F_s/2} \mathrm{e}^{j2\pi ft}\mathrm{d}f = F_s\frac{\sin \pi F_s t}{\pi F_s t}| \,, \tag{19}$$

$$\begin{aligned} u_r(t) &= g(t) * h(t) && (20)\\ &= \int_{-\infty}^{\infty} g(\tau)\cdot h(t-\tau)\mathrm{d}\tau && (21)\\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} u(\tau)\cdot\delta(\tau - nT_s)\cdot h(t-\tau)\mathrm{d}\tau && (22)\\ &= \sum_{n=-\infty}^{\infty} u(nT_s)\cdot h(t-nT_s) && (23)\\ u_r(t) &= \sum_{n=-\infty}^{\infty} u(nT_s)\cdot F_s\frac{\sin \pi F_s(t-nT_s)}{\pi F_s(t-nT_s)}\,. && (24) \end{aligned}$$

The form $\sin(x)/x$ is the *cardinal sine* of x. It appears very often in signal processing mathematics and is also written as $\mathrm{sinc}(x)$. We can thus reconstruct the signal exactly by summing an infinite series of sinc functions, weighted and displaced according to each sample. (Fig. 8). This method is attributed to Kotelnikov [3] and Shannon [6].

It is also possible to reconstruct a signal from one of the other spectral images, thus obtaining a frequency translation of the signal by some integer multiple of F_s (Fig. 9). This is termed frequency conversion and is useful in, for example, digital receivers.

Let us look at one case in some more detail. The reconstruction filter is shifted by just F_s. The filter is then described by

$$H(f) = 1 \;\text{ for }\; F_s < |f| < \frac{3}{2}F_s \;\text{ and zero elsewhere.} \tag{25}$$

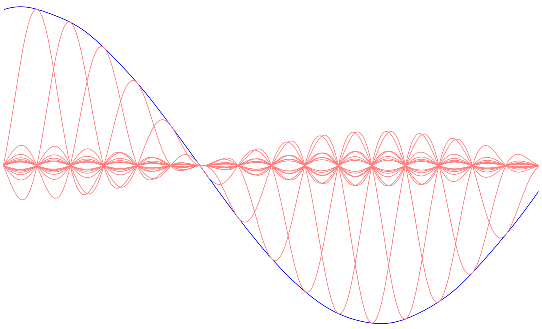


Fig. 8: Reconstruction of a continuous signal by summing sinc functions

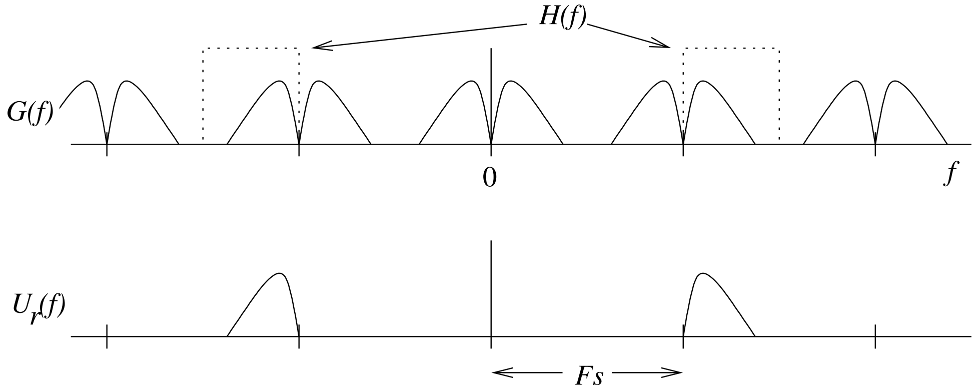


Fig. 9: Signal reconstruction from a different spectral image

The inverse Fourier transform of the filter function gives its time-domain representation:

$$\begin{aligned} h(t) &= \mathscr{F}^{-1}\{H(f)\} && (26)\\ &= \int_{-\infty}^{\infty} H(f)\mathrm{e}^{j2\pi ft}\mathrm{d}f && (27)\\ &= \int_{\frac{-3}{2}F_s}^{\frac{3}{2}F_s} \mathrm{e}^{j2\pi ft}\mathrm{d}f - \int_{F_s}^{F_s} \mathrm{e}^{j2\pi ft}\mathrm{d}f && (28)\\ h(t) &= 3F_s\mathrm{sinc}(3\pi tF_s) - 2F_s\mathrm{sinc}(2\pi tF_s)\ . && (29)\end{aligned}$$

Convolution of Eq. (29) with the time-domain representation of the sampled signal will then yield the reconstructed function $u_r(t)$, which is now centred on F_s rather than on $f = 0$.

$$u_r(t) = g(t) * h(t)\ . \qquad (30)$$

$$u_r(t) = \sum_{n=-\infty}^{\infty} u(nT_s)\{3F_s\mathrm{sinc}\big(3\pi(t-nT_s)F_s\big) - 2F_s\mathrm{sinc}\big(2\pi(t-nT_s)F_s\big)\}\ . \qquad (31)$$

The reconstructed signal consists of the sum of two infinite series of sinc functions, weighted and displaced according to the successive samples. The resultant waveform is illustrated in Fig. 10.

In practice, signal reconstruction is never done that way. Dirac impulses and filters with rectangular transfer functions are mathematical abstractions, and even if they could be realized to a sufficient

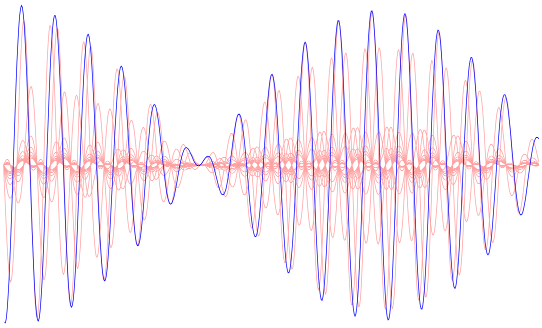


Fig. 10: Reconstruction with frequency conversion

degree of accuracy, each term of the series would extend far to both the past and the future, which is clearly impractical. Instead, each sample is held for a full sampling period. At that stage, the reconstructed signal looks like a staircase approximation of the reconstructed signal (Fig. 11). This sequence of rectangular steps is then smoothed by filtering it with a polynomial filter.

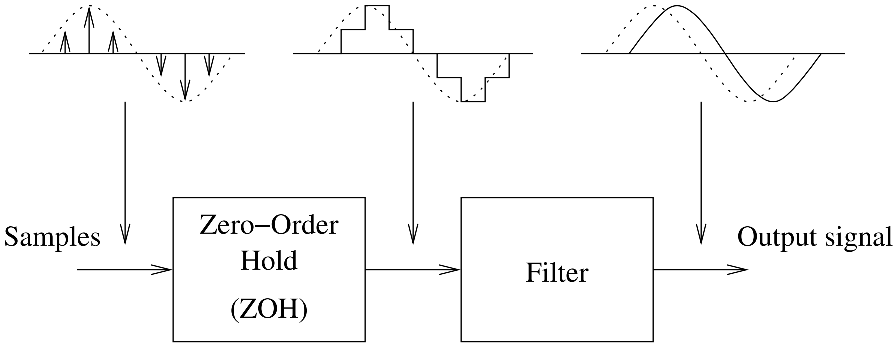


Fig. 11: Signal reconstruction with staircase approximation

The expression for the staircase approximation is

$$u_r(t) = \sum_{n=-\infty}^{\infty} g(nT_s) \cdot \sqcap(t - nT_s) \ , \tag{32}$$

where $\sqcap(t)$ is the rectangle function, of unit value for $0 < t < T_s$ and zero elsewhere (Fig. 12).

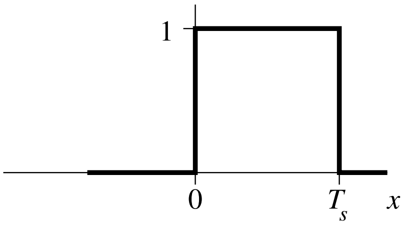


Fig. 12: The rectangle function $\sqcap(t)$

As a consequence of the staircase approximation, the spectrum of the reconstructed signal is distorted, convolved by the spectrum of the rectangular pulses of length T_s:

$$\prod(f) = \mathscr{F}\{\sqcap(t)\} = \int_{-\infty}^{\infty} \sqcap(t)\mathrm{e}^{-2j\pi ft}\mathrm{d}t \tag{33}$$

$$= \int_0^{T_s} \mathrm{e}^{-2j\pi f t}\mathrm{d}t \tag{34}$$

$$= T_s \mathrm{e}^{-j\pi f T_s} \cdot \mathrm{sinc}(\pi f T_s) \tag{35}$$

$$\prod(f) = \frac{1-\mathrm{e}^{-j2\pi f T_s}}{-j2\pi f}\,. \tag{36}$$

Since this function drops off slowly in the frequency domain (Fig. 13), frequencies from adjacent spectral images will leak into the reconstructed signal, so the polynomial filter following it is necessary to suppress them. Incidentally, this function is also known as the Zero-Order Hold (ZOH), because it holds its value constant during one sampling period. (The Laplace domain expression of the ZOH is readily found by substituting $s = j2\pi f$ in Eq. (36) above.) Higher order hold functions, implementing linear, parabolic, or even higher order interpolation between adjacent samples exist and offer improved reconstruction accuracy, albeit with increased delay and a considerable increase in complexity. In practice, higher order hold functions are never used.

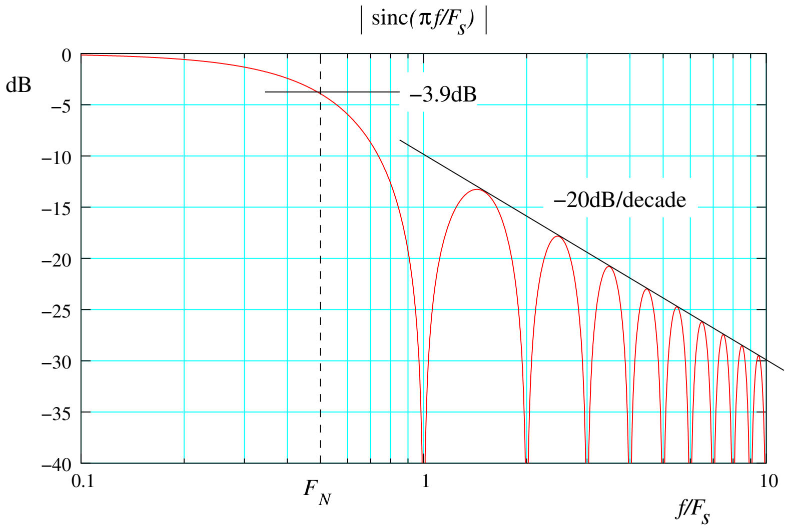


Fig. 13: Frequency response of a ZOH

The polynomial filter can be designed to compensate for the $\mathrm{sinc}(f)$ response of the staircase approximation. As an alternative, you may apply the compensation in the digital domain, before the DAC, and use a simpler flat filter. As a rule, the filter is some compromise between performance, cost, delay, and out-of-band signal rejection. While it is possible to use one of the harmonic bands to reconstruct a frequency-shifted signal, the frequency response of the ZOH drops off too fast to make this an interesting option.

1.4 Spectrum of the quantization error

It is clear from Fig. 2 that even though the quantization error is deterministic, it has a spectrum that bears little relation to $u(t)$, the signal being digitized. It does not yield so easily to analysis though [7–9]. A numerical Fourier transform shows that if the signal is busy enough, and the number of bits great enough, the quantization noise power Eq. (5) is basically evenly distributed over the full spectrum from 0 to F_N, thus appearing as white noise. (Frequencies beyond the Nyquist rate are aliased back into the range 0 to F_N and the spectrum repeats every integer multiple of F_s.) Compared to the level of a full-scale sinusoidal test signal, the noise floor ends up at

$$-\left(1.76 + 6.02n + 10\log_{10}\frac{F_s}{2}\right)\ \text{dBFS/Hz}\,, \tag{37}$$

with n the number of bits of the ADC. Usually the noise floor will be determined from a discrete Fourier transform of a block of samples. It is then more convenient to express the noise floor in dBFS/bin, because that can be read straight off the plot. Equation (37) is modified accordingly:

$$-\left(1.76 + 6.02n + 10\log_{10}\frac{N}{2}\right)\ \text{dBFS/bin}\ , \tag{38}$$

with N the number of samples over which the Fourier transform is taken. Figure 14 shows such a plot, the result of an 8 k sample long recording of a full-scale sinewave.

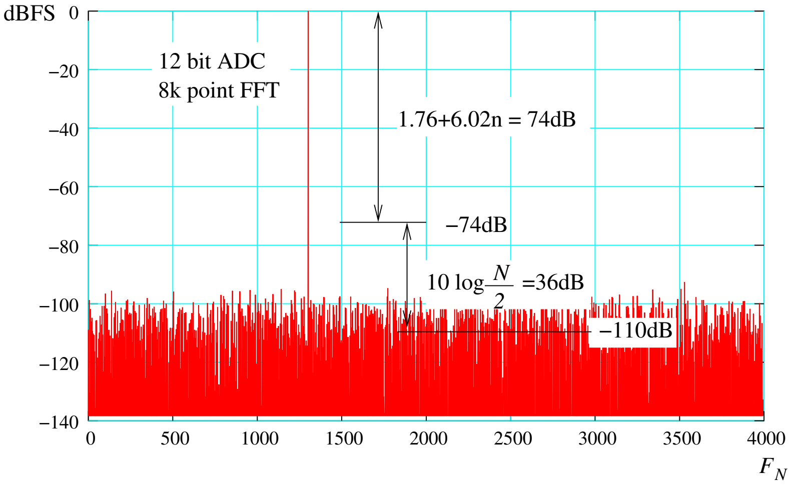


Fig. 14: Discrete Fourier transform of 8 k samples of ADC data

Unfortunately, the quantization noise does not always appear as white noise. Even for a hypothetically perfect ADC, for simple ratios between F_s and the frequency components of $u(t)$, some of the quantization noise power concentrates at discrete frequencies to form spurious signals or spurs. Nonlinearities in actual ADCs contribute harmonics of the frequencies in $u(t)$, which look like still more of these spurs (Fig. 15).

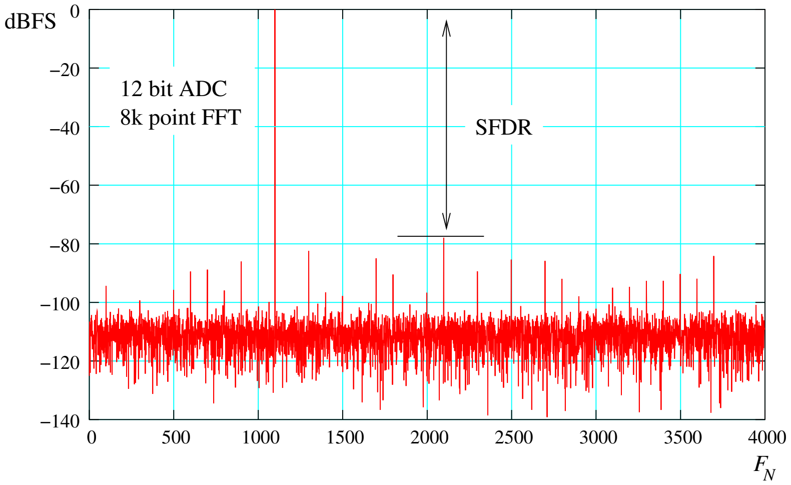


Fig. 15: Spurious components in the spectrum of an ADC

Spurs can be made less objectionable by applying *dither*, an intentional injection of low-level additive noise. This will spread out the energy of a spur over a greater frequency range, without seriously affecting the desired signal. If the spectrum of the dither is chosen so that it does not overlap with the spectrum of the desired signal, the loss of SNR can be made all but negligible.

Datasheets for ADCs used in signal processing will specify a number for the *Spurious-Free Dynamic Range* (SFDR) in dBc, which is the ratio in dB between the greatest spur and the applied test signal, or in dBFS, which is the ratio with respect to a full-scale signal. This is an important specification for digital radio receivers, for example, because a spur may hide or interfere with a weak neighbouring signal. Manufacturers are careful to avoid the problematic frequency ratios mentioned above when specifying the performance of their converters. In practice, the SFDR is measured using a near full-scale sinusoid and applying a Fourier transform to a recording of a few thousand samples. The sine wave must be spectrally clean, with harmonics and phase noise below the quantization noise floor. Even though clean sine waves are the simplest possible test signals, ADCs are getting so good nowadays that this is by no means easy (Fig. 16). (You may end up measuring the quality of your signal generator instead.)

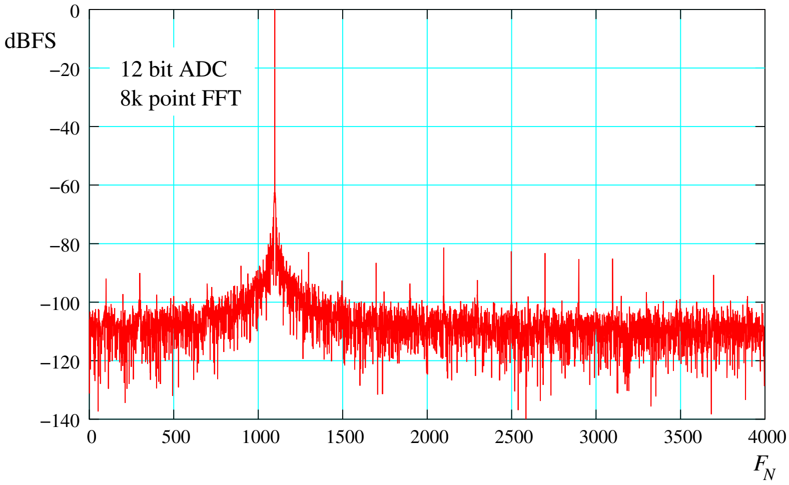


Fig. 16: The signature of excessive $1/f$ phase noise

The total power in the quantization noise depends on the number of bits of the converter, as determined previously [Eq. (6)]. By taking samples faster than required by the Nyquist criterion, the quantization noise is spread over a larger bandwidth. If we then pass the sample stream through a numerical low-pass filter, the noise beyond the signal bandwidth is removed and the SNR improved (Fig. 17). This is termed *oversampling*. The improvement is quite modest however: Only 3 dB for every doubling of the sampling rate, or equivalently, one bit for every factor of four increase. For DC inputs and very quiet ADCs, dither may be necessary to make this work at all. It is possible to do much better, as demonstrated by Σ-Δ ADCs.

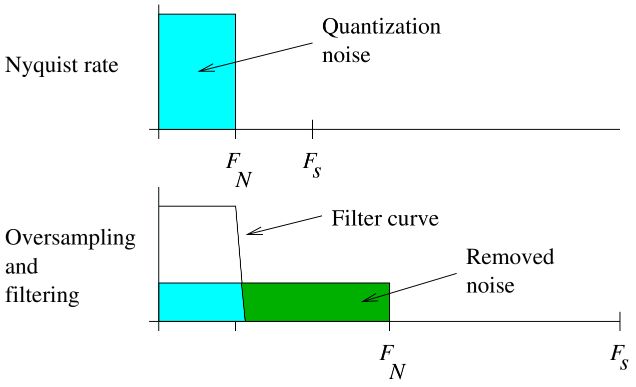


Fig. 17: Oversampling improves SNR

1.5 Limits on ADC resolution

With newer and better ADCs arriving on the scene almost weekly, one might ask if there are any limits to ADC speed and resolution. Of course, there are. As in any analog electronic circuit, thermal noise puts limits on the achievable resolution. There is no point in adding extra bits if all you get for your trouble is more noise.

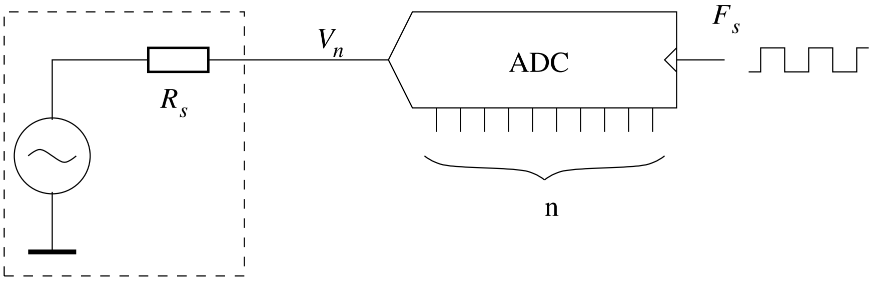


Fig. 18: Model for noise figure calculation

In order to get an idea of where these limits lie, we must make a simplified model (Fig. 18). Let us assume we have a signal source with a real internal resistance R_s. Let us furthermore assume an ADC with a high input impedance and with its input bandwidth limited to the Nyquist frequency $F_N = F_s/2$. In this model, the variance of the thermal noise voltage from the source is

$$V_n^2 = 4kTF_NR_s. \tag{39}$$

The variance of the quantization noise was given in Eq. (5). We can now equate the quantization noise with the thermal noise of the source and solve for n, the number of bits:

$$V_q^2 = V_n^2 \tag{40}$$

$$\frac{A^2 2^{-2n}}{3} = 4kTF_NR_s \tag{41}$$

$$n = \frac{1}{2}\log_2 \frac{A^2}{12kTF_NR_s}\ . \tag{42}$$

Filling in the numbers for Boltzmann's constant $k = 13.8 \cdot 10^{-24}$ J K^{-1} ($k = 13.8$ yW/HzK), the absolute temperature $T = 300$ K and the source resistance $R_s = 50\ \Omega$, this simplifies to

$$n = 29 - \frac{1}{2}\log_2 F_N + \log_2 A. \tag{43}$$

A number of commercially available ADC types have been plotted in Fig. 19, setting resolution against conversion rate. The continuous line is the noise limit according to Eq. (43), with $A = 1$ V. Note that the data for this plot come from a cursory reading of the cover page of the respective datasheets. The actual noise figure of many ADCs is some 25 dB, or equivalently, about 4 bits under the noise limit, and sometimes much worse. As an example, let us consider the LTC2255 from Linear Technology: this high-performance 125 MS/s, 14 bit ADC has a specified SNR of 72.4 dB and an input voltage range of 2 V$_{\text{pp}}$ ($A = 1$). The noise limit at the Nyquist bandwidth $F_N = 62.5$ MHz is $n = 29 - 13 + 0 = 16$ bits, corresponding to an SNR of 98 dB [via Eq. (6)]. The noise figure is defined as the total measured noise over the noise due to the source alone:

$$F = \frac{total\ noise}{source\ noise}\ , \tag{44}$$

(also often expressed in dB,) so the noise figure of this ADC is $F = 98 - 72.4 = 25.6$ dB. Several reasons can be advanced to explain why ADCs are so far from the noise limit:

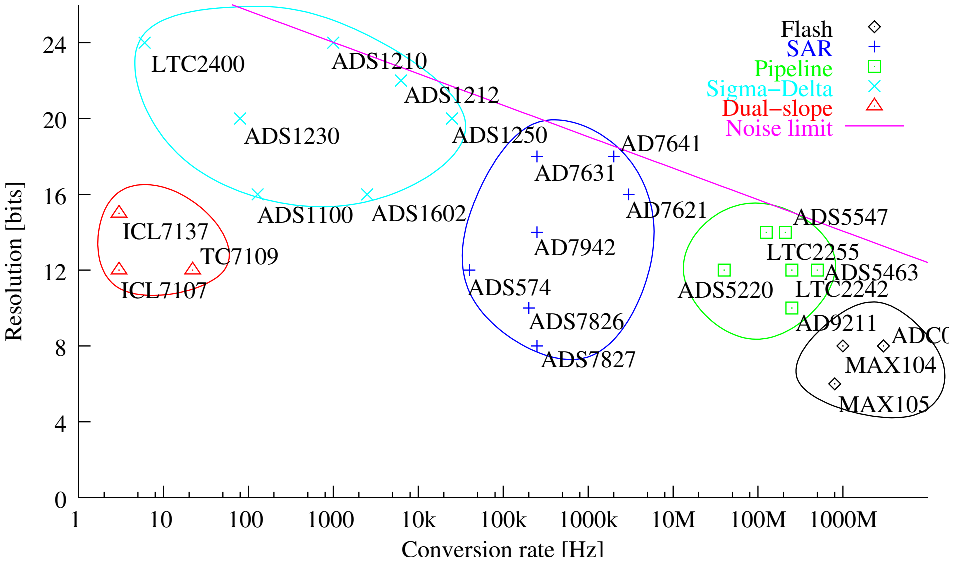


Fig. 19: Resolution vs. conversion rate for some ADCs

- The actual bandwidth of the ADC input is far greater than F_N.
- The ADC input circuitry contains its own noise sources: resistors, amplifiers, samplers, etc.
- Mixed signal devices like ADCs are subject to self-interference; in particular, the digital section of the chip will act back on the reference and analog input sections.

Finally, it is of course possible to trade input voltage range against noise figure by preceding the ADC with a low-noise amplifier or with passive noise-matching circuitry. Some ADCs have a programmable amplifier built in.

1.6 Gain and offset errors

The gain error is the ratio between the nominal and the actual full-scale value of the ADC. The offset error is the difference between the nominal and the actual value at zero input (Fig. 20). Exact ways of defining these values differ from one manufacturer to another, or from one ADC type to another.

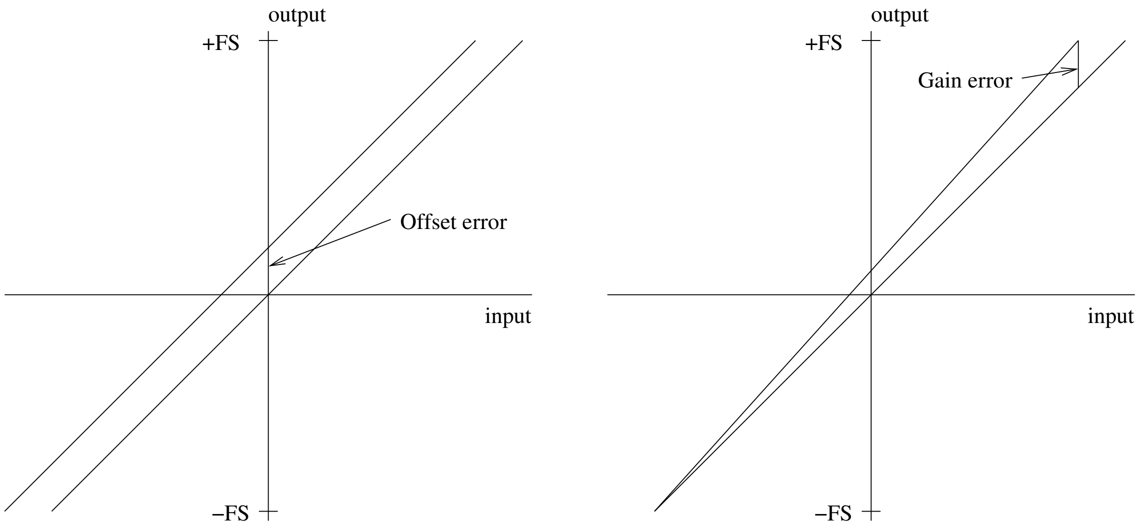


Fig. 20: Offset and gain error definitions

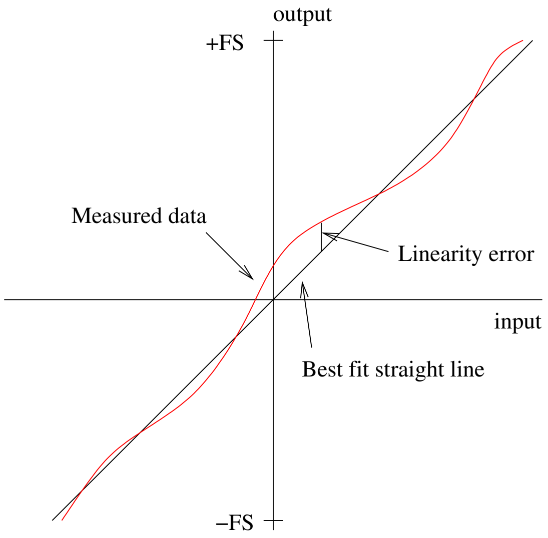


Fig. 21: Integral non-linearity error

1.7 Integral and differential non-linearity, SINAD

The *Integral Non-Linearity* (INL) is measured by comparing a set of measurements distributed over the whole range of the ADC with the linear regression through all points of the set. (Some manufacturers specify the deviation with respect to a straight line through the end points, Fig. 21. Read the datasheets.) Note that this measurement ignores gain and offset errors. The measurement setup used is something along the lines of Fig. 22. The digital comparator will drive a slow analog integrator to make the output from the ADC equal to the set value. A precision voltmeter then gives the corresponding ADC input value. For high-resolution ADCs, the principal difficulty lies in avoiding errors due to thermocouple effects. The shape of the linearity error curve yields information about the expected harmonic distortion products. For example, a parabolic component will yield a 2$^{\mathrm{nd}}$ harmonic. It is customary for manufacturers to quote the SNR of their products excluding the contribution of the first five harmonics. With all noise, harmonics and other spurs included, the performance measure is called SINAD. The abbreviation stands for SIgnal to Noise And Distortion.

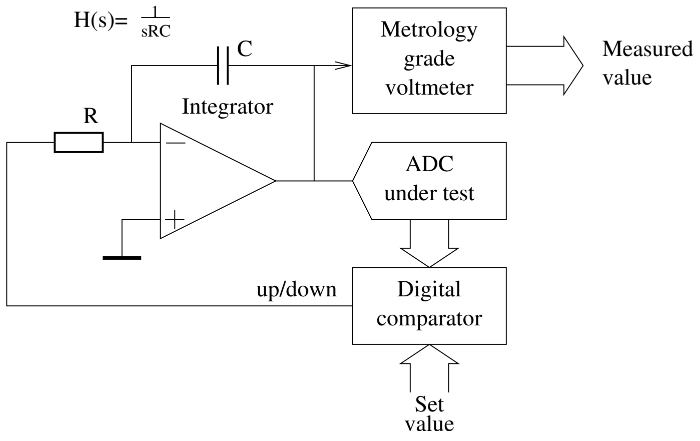


Fig. 22: Setup to measure integral non-linearity

The *Differential Non-Linearity* (DNL) measures the span of input values over which each possible digitized value occurs. In other words, it measures the width of each step of the amplitude quantization function. It is usually measured by histogramming converted values over the full range of the ADC, using an input signal with a flat (or at least with a *known*) distribution. This characteristic is important for ADCs used in closed-loop feedback systems and in spectrography applications. A poor DNL translates into greater than ideal quantization noise. Linearity errors can get bad enough to result in missing codes, or non-monotonicity, meaning that the converter output may actually go *down* for an increase in input signal at some places.

1.8 Clock jitter

ADCs used in signal processing applications need a stable, clean, clock signal. The effects of clock jitter do not depend on the sampling rate of the ADC, but rather on the rate of change of its input signal. With reference to Fig. 23, suppose an ADC digitizes an input signal $u(t)$. Let us assume that the sampling clock is in error by an amount Δt: Then the input sample is off by an amount

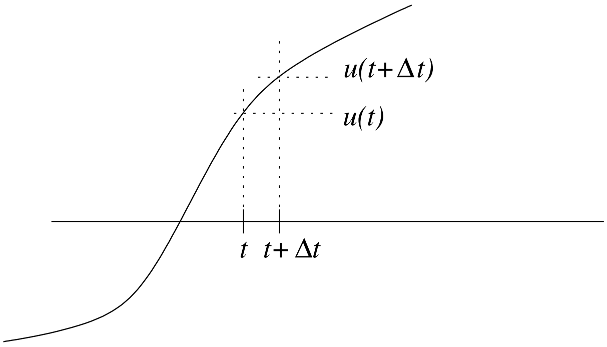


Fig. 23: The effect of clock jitter

$$\Delta U = \frac{\mathrm{d}u(t)}{\mathrm{d}t} \cdot \Delta t\ . \tag{45}$$

This quickly leads to surprisingly stringent demands on the jitter performance of the ADC conversion clock. As a point of example, consider an ADC digitizing a 100 MHz sine, driven by a clock source with the quite respectable jitter specification of 1 ps$_{\mathrm{rms}}$: Its effective resolution would be limited to about 10.5 bits at best (Fig. 24).

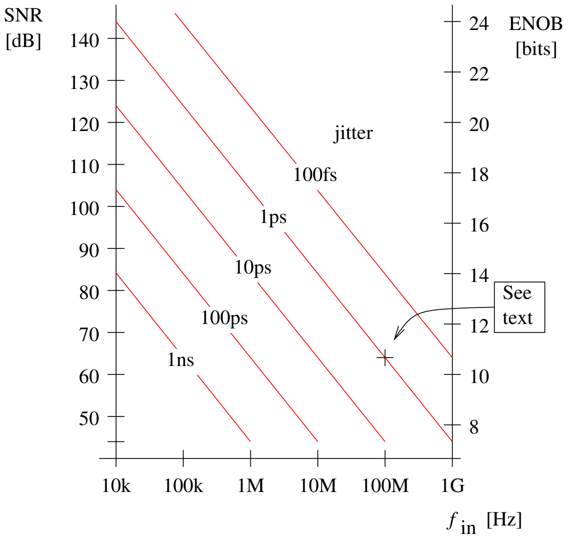


Fig. 24: ADC resolution vs. input frequency and clock jitter

In the light of these considerations, it is evident that the clock of high-performance ADCs should be generated using a high-quality oscillator, mounted close to the ADC and connected to the same ground plane as the ADC (Fig. 25). Owing to the finite rise time of the clock waveform, superimposed ground or power supply noise would deteriorate the jitter specifications of an otherwise good clock signal (Fig. 26). By the same relation as Eq. (45), we can see that only a few millivolts of ground noise on a clock signal with 1 ns rise and fall times are enough to introduce more than 1 ps of jitter. One way to alleviate this might be to use a differential clock signal.

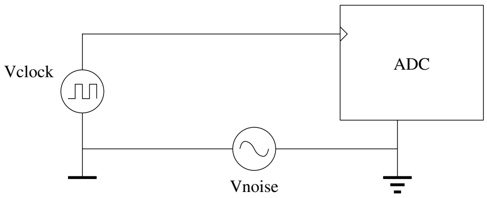


Fig. 25: Ground does not have the same potential throughout

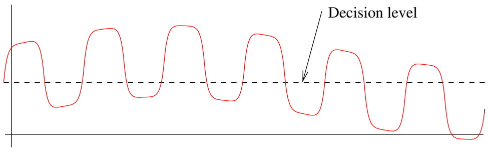


Fig. 26: Ground noise affects clock jitter

Some simple rules can be given to get a good clock signal. First, use a good oscillator. RC oscillators or logic gate oscillators, like the one in Fig. 27, have poor performance, with jitter usually worse than 100 $\mathrm{ps_{rms}}$. Colpitts type LC oscillators, as in Fig. 28, are much better, with around 10 $\mathrm{ps_{rms}}$ of jitter. A Pierce oscillator, see Fig. 29, with quartz or ceramic resonator can be quite good, with jitter at 1 $\mathrm{ps_{rms}}$ or below. Getting jitter much below 1 ps is an exercise in low-noise electronic design and is an art all by itself.

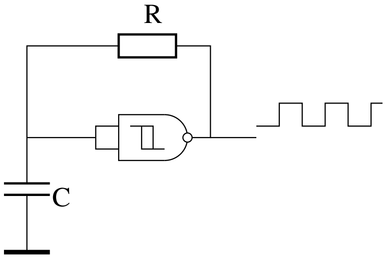


Fig. 27: An RC oscillator

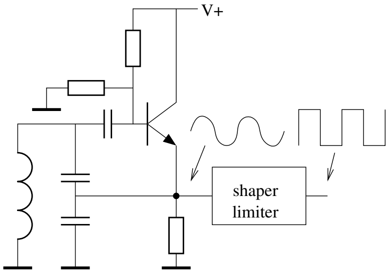


Fig. 28: A Colpitts oscillator

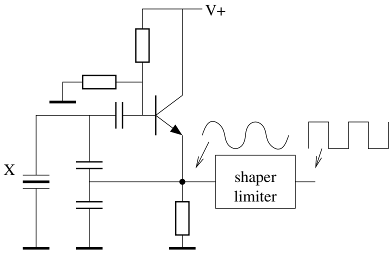


Fig. 29: A Pierce oscillator

It is all too easy to deteriorate a good clock by simple lack of care. In particular, do not feed the ADC from a clock taken from or via a nearby DSP or FPGA (Fig. 30), unless re-synchronized using a flipflop driven by the original, clean clock (Fig. 31). Do not use any remaining logic gates in the same package that buffers the clock signal for other purposes. Carefully filter the power supply of the

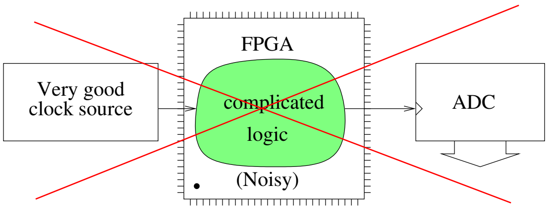


Fig. 30: Do not do this!

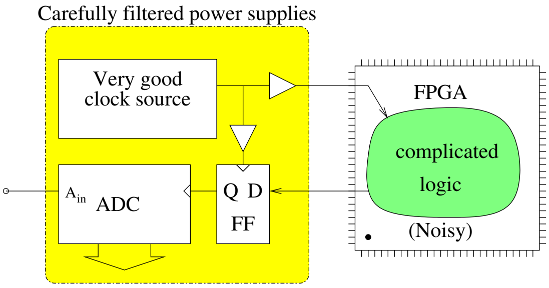


Fig. 31: But if you really must, do it like this

clock signal generation and distribution circuitry. Digital logic usually does not have very good isolation between gates in the same package, nor does it reject any power line noise: It has not been designed with that purpose in mind, because it usually does not need it. Crosstalk between different sections of a logic circuit can seriously deteriorate a clock signal. The clock signal should be kept clear of both the analog input and the digital outputs of the ADC. Even though the clock is a digital square wave, it should be treated with all the care that should be given to an analog signal.

The clock jitter is only a single figure to capture the behaviour of the sampler clock, exactly like the standard deviation is only a single parameter of a distribution. It tells you nothing about the shape of the distribution, nor does it give any information about its spectrum.

More generally, timing jitter is the consequence of oscillator phase noise, the collective effects of various noise sources inherent in their circuitry. Resistors contribute thermal noise and semiconductor devices add shot noise and $1/f$ noise. This noise modulates the generated output tone. Thus, its spectrum acquires sidebands that, for simple oscillators, have a $1/f$ slope for small offsets from the carrier, and that are flat farther out (Fig. 32). Synthesized frequency sources can deviate quite strongly from that simple model. In addition, they may have discrete spurious sidebands in their spectrum. Many manufacturers of oscillators publish phase noise measurements for their products. These may take the form of actual phase noise plots, or of specifications along the lines of: "–150 dBc at 100 Hz offset from the carrier".

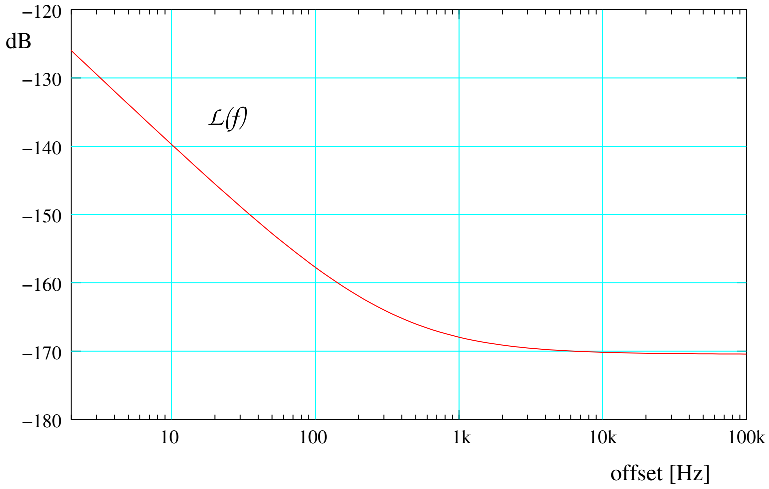


Fig. 32: Typical single-sideband phase noise spectrum

ADCs are clocked devices. They do not directly respond to phase variations, but only to variations in timing of the effective logic threshold. For the sake of argument, let us assume that the clocking waveform is a sinusoid and that the significant instants are the positive-going zero crossings. That approximation is quite accurate for high-speed ADCs with differential clock inputs:

$$V(t) = \sin\big(2\pi F_s t + \varphi(t)\big) \; , \tag{46}$$

where $\varphi(t)$ models the random phase variations. Two different zero crossings at, say, times t_1 and t_2, about N periods apart, satisfy the conditions

$$2\pi F_s t_1 + \varphi(t_1) = 0 \;\text{ and }\; 2\pi F_s t_2 + \varphi(t_2) = 2\pi N \tag{47}$$

and thus

$$2\pi F_s (t_2 - t_1) + \varphi(t_2) - \varphi(t_1) = 2\pi N \; . \tag{48}$$

The time between these instants is an integer number of periods plus a bit of random jitter:

$$t_2 - t_1 = \frac{N}{F_s} + \Delta t \; . \tag{49}$$

Substitution of Eq. (49) into Eq. (48) yields

$$2\pi F_s \left(\frac{N}{F_s} + \Delta t\right) + \varphi(t_2) - \varphi(t_1) = 2\pi N \tag{50}$$

and after solving for the random jitter Δt,

$$\Delta t = \frac{1}{2\pi F_s}\big(\varphi(t_1) - \varphi(t_2)\big) \; . \tag{51}$$

The expected value of the variance is obtained by squaring Eq. (51):

$$\langle \Delta t^2 \rangle = \frac{1}{4\pi^2 F_s^2}\left(\langle \varphi(t_1)^2 \rangle - 2\langle \varphi(t_1)\varphi(t_2) \rangle + \langle \varphi(t_2)^2 \rangle\right) \; . \tag{52}$$

Both t_1 and t_2 are affected by the same statistical jitter, so

$$\langle \varphi(t_1)^2 \rangle = \langle \varphi(t_2)^2 \rangle = \langle \varphi(t)^2 \rangle \; . \tag{53}$$

The variance of this term is the same, whether regarded in the time or in the frequency domain (Parseval's theorem), so, with $S_\varphi(f)$ the spectral density of the phase error,

$$\langle \varphi(t)^2 \rangle = \int_0^\infty S_\varphi(f)\mathrm{d}f \; . \tag{54}$$

The variance of the middle term of Eq. (52) is obtained by taking the cosine transform

$$\langle \varphi(t_1)\varphi(t_2) \rangle = \int_0^\infty S_\varphi(f)\cos(2\pi f\tau)\mathrm{d}f \; , \tag{55}$$

with $\tau = t_1 - t_2$. Substituting Eq. (54) and (55) back into Eq. (52) gives

$$\langle \Delta t^2 \rangle = \frac{1}{2\pi^2 F_s^2}\int_0^\infty S_\varphi(f)\big(1 - \cos(2\pi f\tau)\big)\mathrm{d}f = \frac{1}{\pi^2 F_s^2}\int_0^\infty S_\varphi(f)\sin^2(\pi f\tau)\mathrm{d}f \; . \tag{56}$$

And finally, the standard deviation of the timing error as a function of the phase noise spectrum becomes

$$\Delta t = \frac{1}{\pi F_s}\sqrt{\int_0^\infty S_\varphi(f)\sin^2(\pi f\tau)\mathrm{d}f} \; , \tag{57}$$

with τ the time between the significant instants of the clock waveform, usually equal to $1/F_s$. It is seen that the timing jitter depends on the total energy in the sidebands of the clock signal, weighted so that noise at small offsets and at offset frequencies near NF_s contributes little to the total jitter [10]. In practice, the integration interval is constrained by the reciprocal of the measurement time for the lower bound, and by the bandwidth of the ADC's sampler for the upper bound. Finally, the phase noise is often specified as single-sideband phase noise, denoted $\mathcal{L}_\varphi(f)$. Normally, the phase noise spectrum above and below the carrier is uncorrelated, but has equal power density, so that $S_\varphi(f) = 2\mathcal{L}_\varphi(f)$.

2 Converter architectures

There are many different ways of constructing analog-to-digital converters, and for some architectures, there are hundreds of different ADC implementations on the market. Technology constantly evolves, yielding ever higher performance and deeper integration. Each has its own peculiarities and strengths and depending on the application, some type of architecture may be preferable over another.

Table 1: Some converter architectures

Architecture	Speed	Resolution	Linearity	Applications
Flash	Very fast (GS/s)	Poor (8 bits)	Poor	Oscilloscopes, transient recorders
Successive approximation	Fast (MS/s)	Fair (14 bits)	Fair	DSP, digital receivers, instrumentation
Σ-Δ	Slow (kS/s)	Excellent (24 bits)	Excellent	Process control, audio, weight, pressure, temperature measurement
Dual slope integration	very slow (S/s)	Very good (18 bits)	Very good	Bench-top and hand-held measuring instruments, battery powered devices

The main architectures, with their typical application domains and ballpark specifications, are outlined in Table 1. In the paragraphs following, we shall give an overview of the most prevalent architectures in existence, with some selected specific models of each. For some architectures, the choice available is so overwhelmingly large that there is no hope of giving a comprehensive overview.

Converters exist that combine some aspects of different architectures. For example, flash and successive approximation (SA) can be combined to yield better resolution than pure flash, at a higher speed than pure SA. Besides the main architectures mentioned in Table 1, several less commonly used ADC types exist, for example, voltage-to-frequency converters, Wilkinson converters, tracking converters.

Many converters nowadays use switched-capacitor technology internally. Since charge stored on capacitors leaks away over time, this implies that such ADCs not only have a maximum conversion rate, but a minimum conversion rate as well. Moreover, changing the conversion rate of such ADCs on the fly may result in transient distortion effects. These phenomena do not usually appear in the data sheets.

Increasingly, one finds converters that are complete data acquisition subsystems, with many configuration options programmed by setting internal registers. Such ADCs cannot be used without some sort of processor or microcontroller to set all configuration registers.

2.1 Flash ADCs

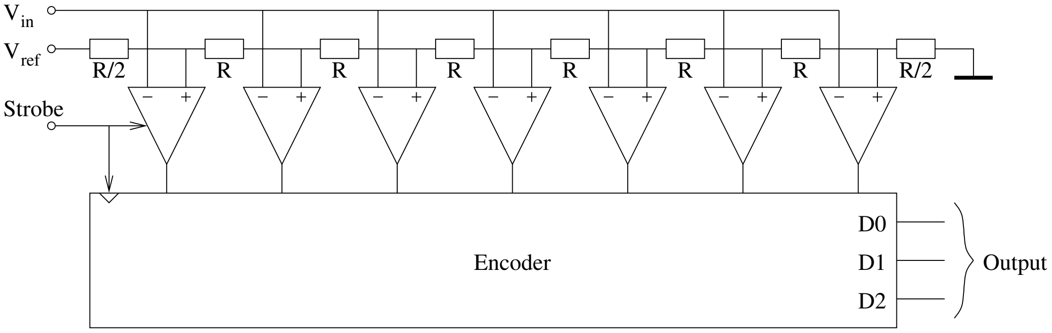


Fig. 33: Principle of a flash ADC

This is the fastest ADC architecture in existence. An n-bit flash ADC has $2^n - 1$ comparators, simultaneously comparing the input signal with the potential on as many taps of a resistor chain divider (Fig. 33). A logic encoder circuit takes the output of the comparators and turns it into a binary code. The converter is very fast: Its conversion time is basically the delay of the comparators plus that of the logic circuits, which added together can be under a nanosecond. The disadvantage is that for every bit extra, the number of comparators roughly doubles, which quickly gets out of hand. The input capacitance goes up linearly with the number of comparators, which quickly leads to unreasonable demands on the driving amplifier. Flash converters commonly have 8 bits, but up to 10 bits is possible.

Owing to differing offsets in individual comparators, the DNL of this kind of ADC is often rather poor. The converter may even be non-monotonous or have missing codes. Differences in switching speed between the comparators may cause transient *sparkle* values to appear at the outputs. Moreover, a comparator's switching speed depends on the magnitude of the difference signal it sees. At least one comparator sees a very small difference and therefore, it will be comparatively slow to settle. (There is even an infinitesimal chance that it will not settle at all, so called *metastability*.) Flash converters are often clocked or strobed to hide transient codes. For high input frequencies, the dynamic performance of the ADC may be much improved by preceding it with a Sample-and-Hold Amplifier (SHA), timed in such a way that the input signal is kept steady while conversion takes place. Flash converters are used in oscilloscopes and transient digitizers.

The digital interface of flash ADCs is sometimes de-multiplexed to reduce the necessary transfer rate on the digital buses. Differential signalling is used extensively, both for analog and digital signals. The chip always has multiple ground and power supply connections. Thus, even though a flash ADC may digitize to only eight bits, it is usually housed in packages with over a hundred pins. The power consumption of high-performance flash converters is quite high, often exceeding 1 W.

Table 2: Some flash converters

Type	Bits	Rate	INL	DNL	Power	Interface
ADC081500	8	1.5 GS/s	0.3 LSB	0.15 LSB	1.2 W	LVDS
MAX104	8	1 GS/s	0.25 LSB	0.25 LSB	5.25 W	diff PECL

2.2 Successive approximation ADCs

A very common and popular architecture is the successive approximation ADC (Fig. 34). It starts a conversion by first comparing the half scale value with the input. If the input is greater, the corresponding bit is set; If not, it is cleared. The approximation is then refined with successively smaller increments until the required precision is reached, at which time the conversion is complete.

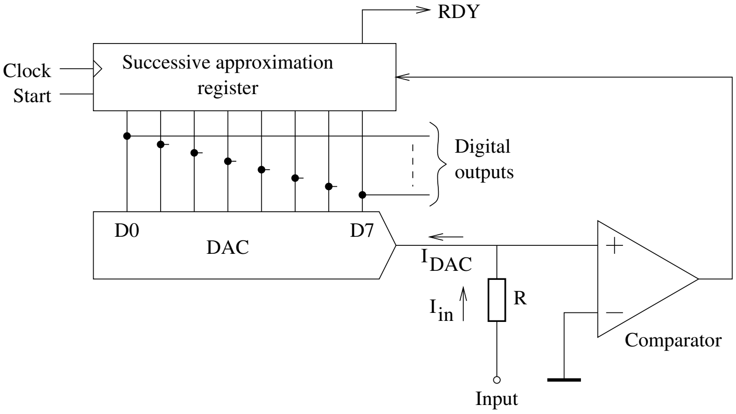


Fig. 34: Principle of a successive approximation ADC

The analog input bandwidth of SAR ADCs often well exceeds the Nyquist frequency. This can be useful in undersampling applications, like in digital receivers, but it makes the use of an anti-alias filter mandatory. It is important to keep the input value steady while conversion is going on. Most converters nowadays have a built-in hold circuit. A SHA should be used if this is not the case and if the input signal can vary during conversion. Even if the converter has its own hold circuit, if undersampling is used, performance may still benefit from a wide-band sample-and-hold preceding the ADC.

The digital interface of SAR ADCs is usually parallel, but lately a lot of serial interface devices have appeared on the market. These are handy for use with small microcontrollers and are available in tiny packages with surprisingly few pins.

The DNL of this type of ADC can be notably poor, sometimes to the point that certain output values never occur: So-called missing codes. SAR converters can never be non-monotonous. If you care about DNL, such as when performing spectrography, avoid using successive approximation ADCs.

Table 3: Some successive approximation converters

Type	Bits	Conv. rate	BW	SFDR	INL	DNL	Notes
AD7476	12	1 MS/s	6.5 MHz	80 dB	1 LSB	0.75 LSB	6-lead SOT23
LTC2356	14	3.5 MS/s	50 MHz	86 dB	0.5 LSB	0.4 LSB	10-lead MSOP
LTC1279	12	600 kS/s	5 MHz	82 dB	1 LSB	1 LSB	SOL-24

2.3 Pipelined converters

These may be regarded as a combination of flash and successive approximation. They are sometimes also referred to as segmented or sub-ranging ADCs. At each clock pulse, the input value is digitized by a coarse, first-stage, flash ADC. This first stage may have just three or four bits of resolution. The coarse approximation is subtracted from the original input and the residue is converted by the next flash stage on the next clock tick. This may be repeated one or two more times to reach the desired resolution. The partial conversion results are combined using logic circuits. Each converter stage usually spans a little more than the ideal residue of the preceding stage, so that a converter such as the one in Fig. 35, with one 4-bit stage and two 5-bit stages, may finally deliver an overall result of only 12 bits. This is done to accommodate linearity errors in the sub-sections (Fig. 36).

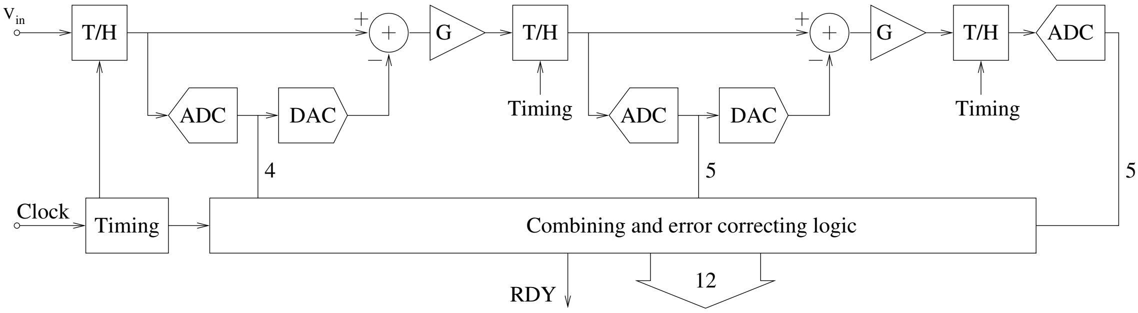


Fig. 35: Architecture of a pipelined ADC

The ADC delivers one output value every clock cycle, but each such value refers to the input level of several clock cycles previous (Fig. 37). This may be important if the ADC is part of a closed-loop feedback system, or if it is multiplexed across several input signals. These ADCs are used in fast signal processing applications like digital radio, radar, and medical ultrasound. The pipeline architecture is currently the subject of intense competition between several manufacturers and new types are introduced almost every month.

2.4 Σ-Δ ADCs

This architecture is very popular for audio converters and high-resolution, low-speed measurement applications such as weighing scales and pressure gauges. It is simple and undemanding with respect to

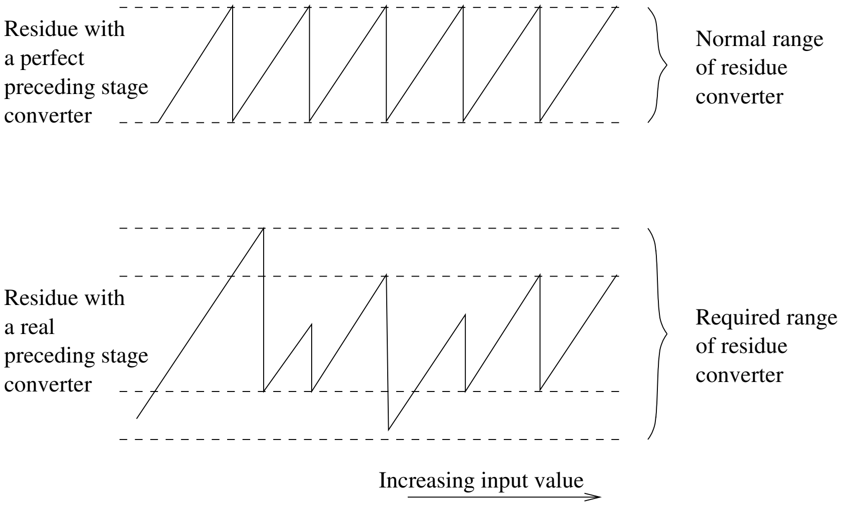


Fig. 36: The effect of imperfect sub-converter linearity

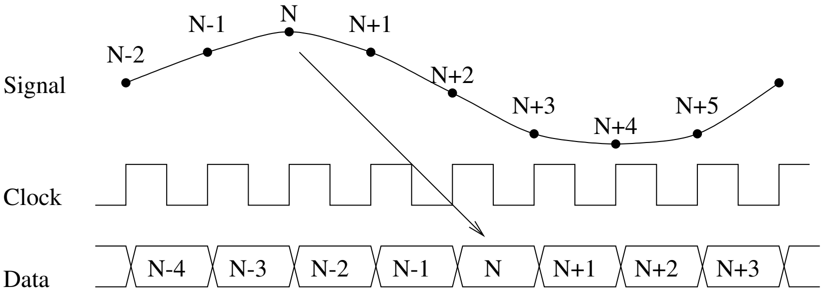


Fig. 37: Pipelined ADCs have several clock periods of latency

Table 4: Some pipelined converters

Type	Bits	Conv. rate	BW	SFDR	INL	DNL
AD872	12	10 MS/s	35 MHz	75 dB	1.75 LSB	0.5 LSB
AD9432	12	105 MS/s	500 MHz	80 dB	0.5 LSB	0.25 LSB
LTC2255	14	125 MS/s	640 MHz	88 dB	1 LSB	0.5 LSB
LTC2242	12	250 MS/s	1.2 GHz	78 dB	1 LSB	0.4 LSB
ADS5232	12	65 MS/s	300 MHz	85 dB	0.4 LSB	0.3 LSB
TDA9910	12	80 MS/s	370 MHz	70 dB	2 LSB	0.6 LSB

precision of components, but it requires a lot of digital post-processing, lending itself well to low-cost CMOS IC implementation. It is characterized by excellent linearity and resolution, but has low conversion rate and high latency.

Its principle is shown in Fig. 38. The output of an integrator is sampled by a strobed comparator which effectively operates as a single-bit ADC. The integrator continuously integrates the difference between the input and the comparator output. The output is a sequence of pulses with the average width representing the value of the input. The clock rate is normally many times higher than the Nyquist rate. The feedback loop, comprising the integrator and comparator is called a Σ-Δ modulator. A digital low-pass filter calculates the average over a number of samples, trading sampling speed for resolution, producing a multiple-bit sample stream at a fraction of the clock rate. This reduction of sampling rate is called *decimation*.

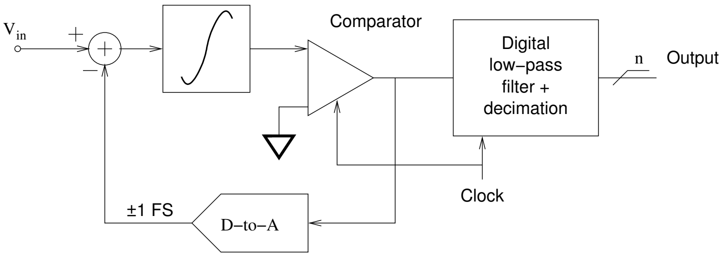


Fig. 38: Principle of a first-order Σ-Δ ADC

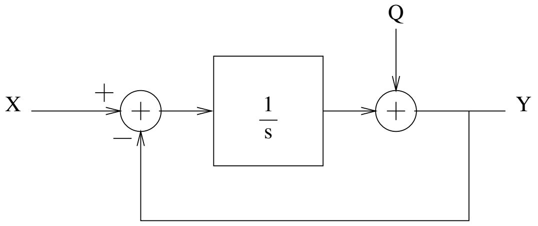


Fig. 39: Block diagram of a first-order Σ-Δ ADC

With some stretch of the imagination [11], this converter can be modelled as a linear feedback system, with the quantization due to the comparator considered as an independent source of random uncorrelated noise (Fig. 39). Examination of the loop transfer function teaches that the input signal X is subjected to a low-pass response (Fig. 40).

$$\frac{Y}{X} = \frac{1}{s+1} \; . \tag{58}$$

The transfer function from the point of view of the sampling noise Q is a high-pass

$$\frac{Y}{Q} = \frac{s}{s+1} \; . \tag{59}$$

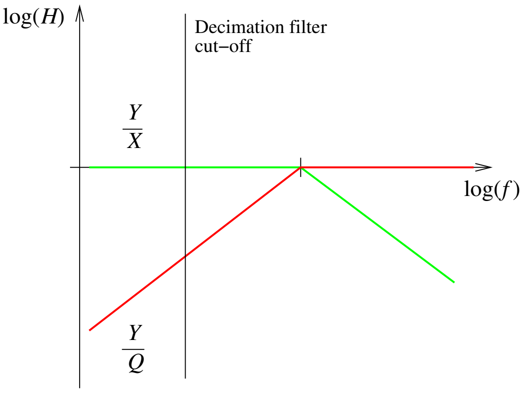


Fig. 40: Noise shaping for a first-order Σ-Δ ADC

With a suitable choice of sampling frequency and integrator time constant, most of the quantization noise is beyond the bandwidth of interest and removed by the decimation filter. The combined response of the Σ-Δ modulator and the decimation filter affords 9 dB of resolution improvement per octave of oversampling ratio: 6 dB for the modulator and 3 dB for the filter. The low-pass response with respect to the input signal and the high oversampling ratio ease the requirements put on the anti-aliasing filter, sometimes to the point of making it superfluous altogether.

The performance of Σ-Δ ADCs can be significantly improved by using higher order modulator loops and/or multi-bit quantizers (Fig. 41). However, ensuring stability of higher order modulators after an input overload event becomes problematic.

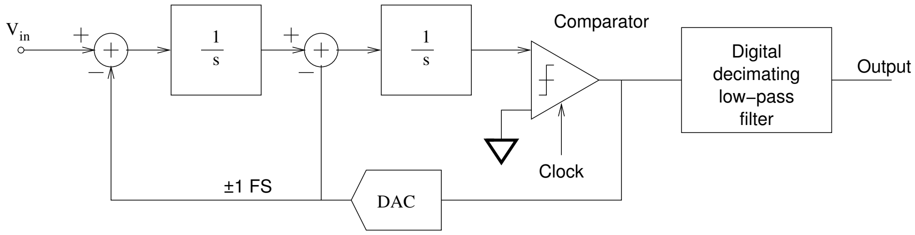


Fig. 41: Block diagram of a second-order Σ-Δ ADC

The decimation filter is usually implemented as a FIR filter with several tens, or even up to a few thousand, taps. Because of the long latency of such filters, this architecture is less suitable for use in feedback systems or in applications where many different signals are multiplexed into a single converter. Some converters have programmable filters, sometimes combining IIR and FIR filters, to trade settling time against resolution according to the needs of the application.

Σ-Δ converters designed for instrumentation are usually DC-accurate. Some have integrated signal conditioning amplifiers, for example, the AD7799 has a built-in instrumentation amplifier (as well as programmable conversion rate and filter bandwidth, and several other fancy features). Those used for audio signal processing have good dynamic properties, but usually very poor gain and zero error specifications.

Table 5: Some Σ-Δ converters

Type	Bits	Sampling rate	Delay	Decimation	Notes
ADS1610	16	60 MS/s	3 μs	6	Instrumentation
AD1871	24	6.1 MS/s	460 μs	64	Audio
AD7400	16	10 MS/s	NA	NA	Σ-Δ mod. (isolated)
AD7799	24	64 kS/s	240 ms	15 k	Instrumentation
LTC2400	24	154 kS/s	160 ms	256	Instrumentation

2.5 Dual-slope integration

This architecture offers good resolution and linearity, combined with low power consumption, but is usually limited to low conversion rates (tens of conversions per second), Fig. 42. It is very tolerant of component value drift and lends itself well to integration on low-cost semiconductor processes. It is often used in measurement instruments like volt meters and weighing scales. It is available in monolithic ICs containing not only the converter, but also auto-calibration and auto-ranging circuitry and even display driver logic. This technology has lost a lot of ground in favour of Σ-Δ converters.

Its operation has two phases: In the first, a current proportional to the input signal is integrated onto a capacitor for a fixed number of clock periods, Fig. 43. Then, a constant current I_{ref} is used to linearly discharge the capacitor, while at the same time incrementing a digital counter at the same clock rate. The counter is stopped when the capacitor voltage reaches zero. The counter then holds the digital representation of the input value. It is clear that the exact value of the capacitor is of no importance (but beware of dielectric absorbtion effects), and neither is the exact clock frequency. Often, the duration of the charging phase is chosen to be an integer number of mains power periods, in order to reject power line interference. The integral over one period of mains interference is usually zero.

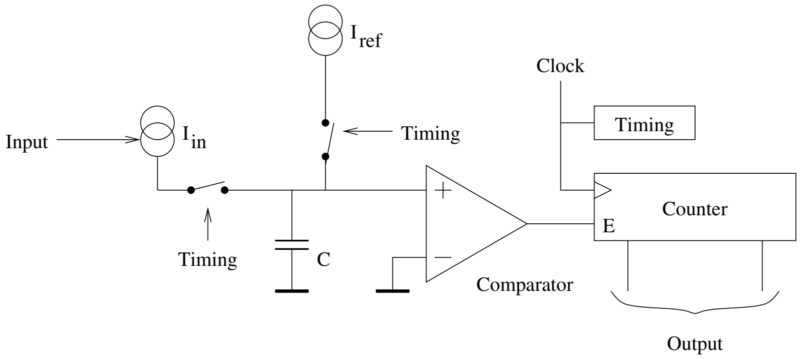


Fig. 42: Principle of a dual-slope ADC

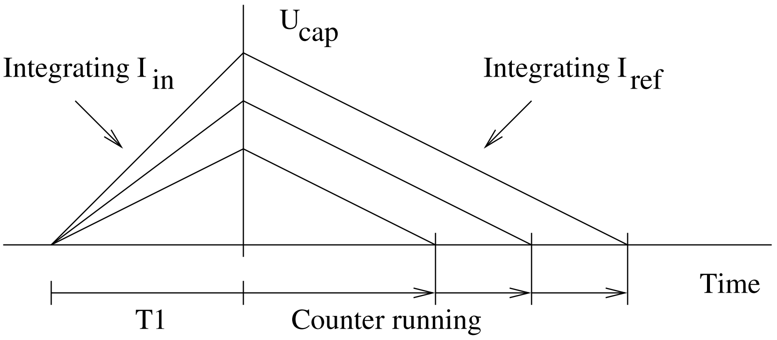


Fig. 43: Operation of a dual-slope ADC

Because of their excellent DNL, variants of this architecture are still widely used in high-energy physics to measure charge from photomultipliers or wire chamber detectors. In this application, they are referred to as ‘Wilkinson run-down ADCs’. Another variant implements a time-to-digital converter, by gating a constant current into the capacitor between a START and a STOP pulse input during the first phase. This method easily yields timing resolution in the picosecond domain, well beyond the reach of direct counting logic.

Table 6: Some dual-slope converters

Type	Resolution	Conversion rate	Linearity	Notes
ICL7106	$3\frac{1}{2}$ digits	3 S/s	10^{-4}	LCD display driver
TC7109	12 bits	30 S/s	$5 \cdot 10^{-5}$	μC compatible
ALD500	$5\frac{1}{2}$ digits	2 S/s	$4 \cdot 10^{-5}$	No counters
MQT300	18 bits	10 μs	10^{-5}	Wilkinson

2.6 Voltage-to-frequency converters

This is a very cheap and simple converter, yet it can deliver excellent linearity (10^{-4} ballpark). Its output signal, square pulses at a rate proportional to the input voltage, can very easily be sent over long distances with little risk of corruption. A simple counter can be used to acquire the signal. It is very easy to make integrating measurements: Just do not reset the counter. Converting the signal back into an analog voltage is also very simple: A low-pass filter will often do. These properties make it very interesting for remote measurement in industrial process control applications, e.g., in oil refineries and other large chemical plants. It is used in energy meters, flow, pressure and temperature meters, fuel gauges, and speed measurement.

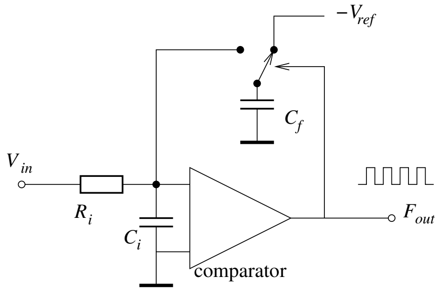


Fig. 44: Principle of a voltage-to-frequency converter

Its operation can be easily understood from Fig. 44. The input voltage is integrated on the input capacitor C_i. Every time the voltage on C_i goes positive, the comparator discharges the feedback capacitor C_f into the input, thus removing a fixed amount of charge from C_i and taking its potential negative again. Immediately after, C_f is once more connected to the (negative) reference and charged up with a new fixed amount of charge. The rate at which this repeats depends linearly on the value of the input voltage. Note that as long that $C_i \gg C_f$, the exact value of C_i does not matter. To first order, the circuit's conversion constant is

$$\frac{F_{out}}{V_{in}} = \frac{1}{R_i V_{ref} C_f} \ . \tag{60}$$

Table 7: Some voltage-to-frequency converters

Type	Range	Linearity	$F_{\max}$	
LM331	1 kHz/V	0.003%	10 kHz	
AD537	1 mA	0.25%	150 kHz	
AD7740	2.5 V	1%/kHz	500 kHz	Synchronous
XR4151	1 kHz/V	0.05%	10 kHz	
VFC110	400 μA	0.02%	4 MHz	

2.7 Other architectures

Some physical quantities lend themselves well to particular conversion techniques. Linear or rotational movement or displacement can be converted into digital format using coding rulers or disks (Fig. 45). Gray-code patterns are used to avoid glitches due to multiple simultaneous bit transitions. Proximity detectors, optical or Hall effect detectors can detect the passage of gear teeth to measure speed or position in vehicles and machinery.

For voice communication, ADCs have been designed with non-uniform quantization functions, with decreasing step sizes near zero (Fig. 46). This was done to improve the SNR for small signals, common in speech. Nowadays, these are advantageously replaced by ordinary linear Σ-Δ converters, followed by dynamic range compression in the digital domain. There have been attempts to create floating point format ADCs, such as MicroNetworks' MN5420. While the idea seems attractive, none of these were particularly successful commercially.

3 Digital-to-analog converters

There exist many different DACs of varying architectures. Some are optimized for instrumentation applications, requiring good DC accuracy. Others may be optimized for the generation of RF signals,

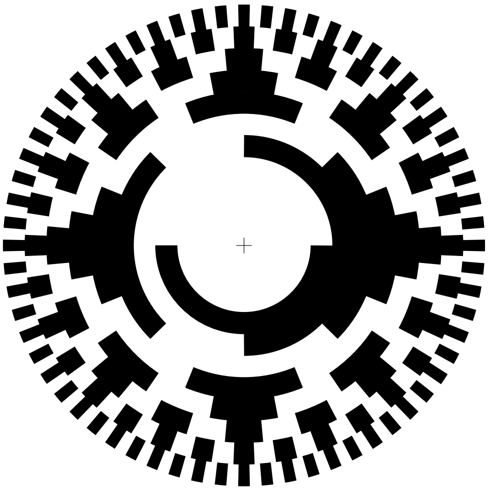


Fig. 45: A Gray-coded resolver disc as used in angular sensors

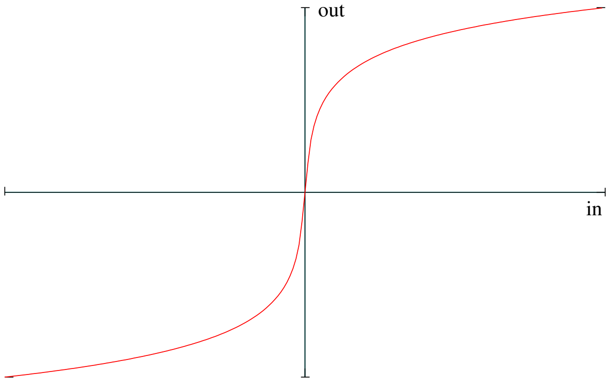


Fig. 46: A-law compression curve used for voice communication converters

and the emphasis there lies on speed and the lowest possible spurious and harmonic components. Audio DACs are optimized for cost and distortion.

Table 8: Some DAC architectures

Architecture	Properties
Kelvin–Varley divider	Very accurate, monotonous.
Thermometer DAC	Monotonous. Limited number of bits.
Binary weighted ladder	Very common, but subject to glitches.
R–2R ladder	Widely used. Not very power efficient.
$\Sigma - \Delta$	Linear, accurate, but complex.

A Digital-to-Analog Converter (DAC) produces a staircase approximation of the desired signal. As demonstrated in Section 1.3, this signal must be filtered to remove the out-of-band energy.

In addition to the aliases of the desired signal, a DAC output also contains harmonics, due to the INL of the DAC, and spurious signals that may have multiple origins. Note that harmonics above the Nyquist rate also have aliases, so the spectrum of a DAC output can look positively busy. If an alias of a harmonic falls within the desired signal bandwidth, it will be a source of trouble. It is impossible to filter this out! Careful frequency planning is used to avoid these pitfalls.

The quality of the DAC clock is just as important as it is for ADCs. The DAC output signal is convolved with the spectrum of the clock source.

Many DACs have an internal fixed full-scale reference voltage. Some can be used with an externally applied reference and a few will even accept reference voltages of either polarity and with a reasonable bandwidth. These are called *multiplying* DACs and can be very useful in certain applications.

3.1 The Kelvin–Varley divider

The grandfather of all DACs is perhaps the Kelvin–Varley Divider (KVD), which, despite its age, is still one of the most accurate D-to-A converters in existence. It is still used in metrology applications. Its basic building block is a divider made of a string of carefully matched resistors (Fig. 47). It is guaranteed to be monotonic. The two switches selecting the taps always move together. This keeps the total resistance between the end points constant. KVDs are available with up to 7 decades of resolution. The output impedance of a KVD depends on the selected output value, and thus is accurate only at zero current. It can therefore only be used in either nulling bridge measurements, or buffered with a high input impedance precision amplifier. KVD-like structures are used as subsections of segmented DACs.

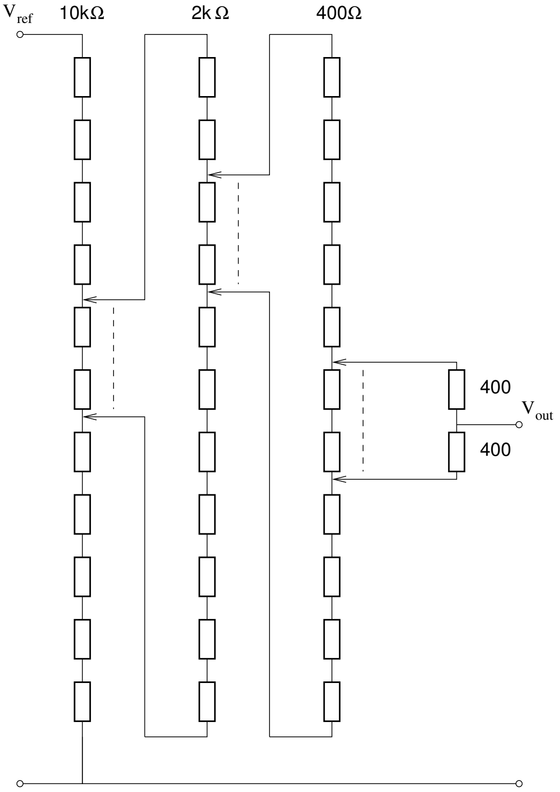


Fig. 47: The Kelvin–Varley divider

Considering that commercially available, manually switched KVDs are basically nothing more than resistors and switches, they can be surprisingly expensive. However, they are precision instruments, targeted at calibration and standards laboratories.

Table 9: Kelvin–Varley dividers

Type	Input resistance	Resolution	Accuracy	Notes
KVD720A	100 kΩ	10^{-7}	10^{-7}	Precision, metrology
GR1455A	10 kΩ	10^{-4}	$1.5 \cdot 10^{-4}$	Routine lab use
DAC081S101	NA	8 bits	$7 \cdot 10^{-4}$	String DAC
DAC5571	NA	8 bits	0.5 LSB	String DAC

3.2 The binary-weighted ladder DAC

This very simple DAC architecture adds together the currents flowing through a set of resistors dimensioned such that each differs by a factor of two from its immediate neighbours. The device as depicted in Fig. 48 (albeit with 8 bits and an additional single-transistor output buffer amplifier) was used by Hybrid Systems as a half-serious, give-away, promotional item at an exposition in the 1970s, but it proved so popular that it is still available today as the DAC371-8. In the schematic as drawn, the currents are set with resistors and the switches are diodes. The output should feed a zero impedance point, like the virtual signal ground at the negative input of an operational amplifier buffer. In more serious implementations of this principle, current sources and transistor differential switches may be used, and the output voltage may be allowed to vary.

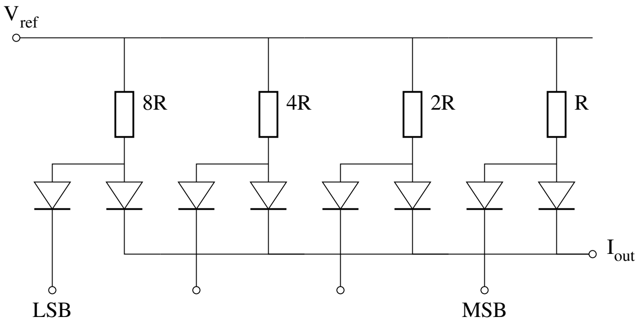


Fig. 48: Binary-weighted DAC

Both current output and voltage output variants are possible. The figure is an example of the former. Current output DACs have only a limited *compliance*, i.e., they can deliver their nominal current over only a limited range of output voltage, often not even 500 mV. Amplifiers are then needed to bring up the signal to a useful level. The total capacitance of the internal current switches is often enough to push the amplifier into instability unless extra frequency compensation is used.

The wide range of resistor values required makes this architecture less suitable for integration. The different currents in the switches affect the switching speeds, leading to glitches in the output. This architecture is not inherently monotonic.

Table 10: Some binary-weighted DACs

Type	Bits	Update rate	Settling time	glitch	Application
THS5641A	8	100 MHz	35 ns	5 pVs	Video, communication
TLC5602	8	30 MHz	30 ns	–	Video

3.3 The R–2R ladder

This very common architecture uses only two different resistor values, and all the switches conduct the same current, thus correcting two of the deficiencies of the binary-weighted ladder, Fig. 49. Its disadvantage is that only a fraction of the current of each stage contributes to the output, making it less attractive for low-power applications. Otherwise this architecture is very similar to the binary-weighted ladder. It is not inherently monotonic.

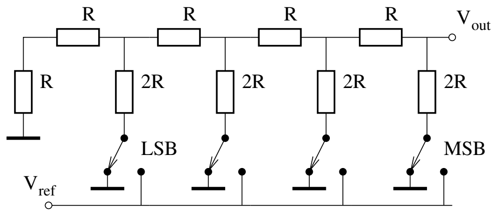


Fig. 49: R–2R ladder DAC

Table 11: Some R–2R DACs

Type	Bits	Update rate	Settling time	Glitch	Notes
AD5445	12	20.4 MHz	80 ns	2 nVs	Multiplying DAC
LTC7545	12	9 MHz	1 μs	2 nVs	4Q multiplying DAC
MAX7524	8	6 MHz	400 ns	–	CMOS multiplying DAC

3.4 Thermometer DAC

This architecture is most often used as a sub-circuit in segmented DACs (see below). For N bits, it uses 2^N identical current sources, with digital decoder logic to switch on the appropriate number of sources. To remain practical, the number of bits is limited, Fig. 50. It is inherently monotonic and glitch-free.

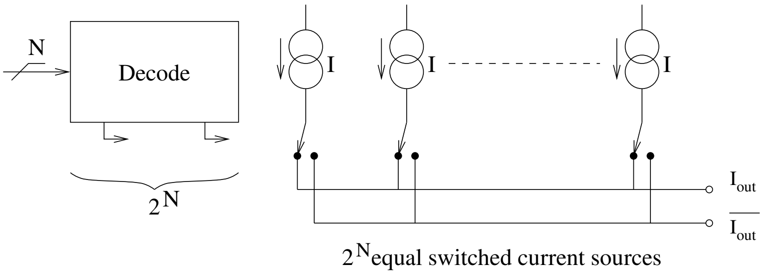


Fig. 50: Thermometer DAC

3.5 Segmented DACs

Segmented DACs combine several architectures in an attempt to strike a balance between speed, power consumption, distortion, glitch energy and maybe other considerations. Most high-performance DACs, such as those used in communication equipment and signal generators use some variant of this architecture. Figure 51 shows an example that combines a Kelvin–Varley first stage with an R–2R ladder secondary stage. A few bits of the applied digital input are used to select the KVD tap, and the remaining bits control the switches of the R–2R section. Other combinations implying binary-weighted and thermometer DAC sections are also encountered.

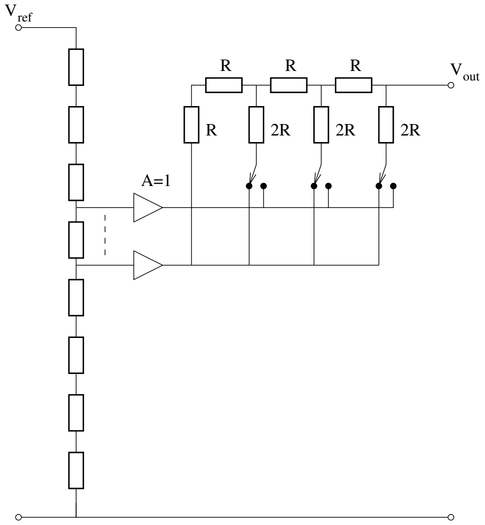


Fig. 51: A segmented DAC

Table 12: Some segmented DACs

Type	Bits	Update rate	Settling time	SFDR	Notes
AD9753	12	300 MS/s	11 ns	69 dB	Communication
AD9735	12	1.2 GS/s	1 ns	75 dB	Communication
LTC1591	14	9 MS/s	1 μs	94 dB	4Q multiplying DAC
TDA9935	14	80 MS/s	–	73 dB	Dual, communication

3.6 Pulse-width modulation

This is a very simple and cheap method that finds application in motor controllers, audio amplifiers (class-D) and single-chip micro controllers. The duty cycle of a fixed-amplitude square pulse is varied according to the desired output level, Fig. 52. A low-pass filter averages the sequence of pulses into a smooth output signal. The architecture is inherently linear. In power applications, such as portable audio amplifiers and motor controllers, its excellent power efficiency is a decisive advantage.

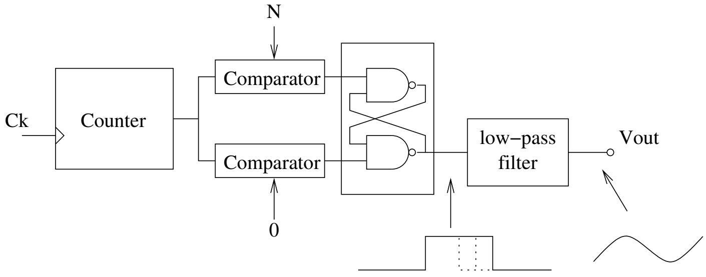


Fig. 52: Pulse-width modulation DAC

3.7 Σ-Δ DACs

This architecture is similar in its general idea to the pulse-width modulation (PWM) DAC, but much more sophisticated in its execution, Fig. 53. At the same sample rate, its resolution comfortably surpasses that of the PWM DAC. A fully digital implementation of a Σ-Δ modulator produces a train of output pulses with an average on–off ratio proportional to the desired output value. The pulse rate is many times higher than the maximum signal frequency. A low-pass filter smooths the pulses into a continuous output signal. Like for Σ-Δ ADCs, the modulator may be of higher order, and the single-bit internal DAC may be replaced by a multi-level device.

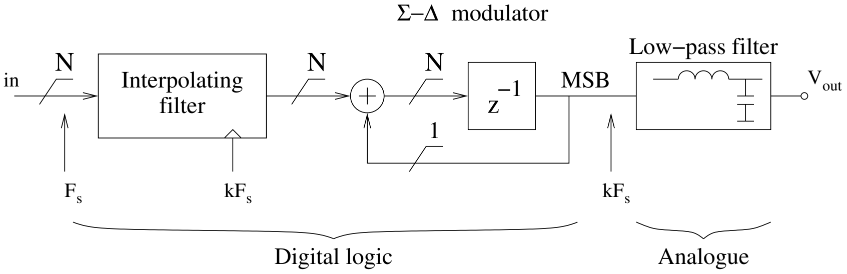


Fig. 53: Architecture of a first-order Σ-Δ DAC

This architecture is relatively recent, because of the complexity of the digital Σ-Δ modulator. It is inherently linear and monotonic and can be built with excellent resolution. It is used in instrumentation and audio equipment. It is available as a logic design file (IP, Intellectual Property) that can be programmed into FPGAs, leaving the designer only the burden of designing the output stage and filter. The principle can also be applied to extend the resolution of ordinary DACs.

Table 13: Some Σ-Δ DACs

Type	Bits	Rate	SNR	
AD1955	24	192 kS/s	120 dB	Audio
MAX5556	16	50 kS/s	86 dB	Audio
CWda30	≤24	NA	Variable	IP core

3.8 DAC ailments

DACs suffer from basically the same imperfections as ADCs regarding linearity and harmonic distortion, leading to spurs, harmonics of the signal, and clock feedthrough artefacts in the spectrum of the output signal. However, where an ADC would have missing codes, a DAC would be non-monotonous. DACs used in closed-loop feedback systems should be monotonous, or the loop might hang on such imperfections. At some points in its transfer function, a DAC may generate glitches, where the output value briefly departs from the vicinity of the final output value, even for small changes in the digital input. These typically occur when many input bits change value simultaneously. Manufacturers will usually specify the importance of these glitches in units of volt·seconds, improperly referred to as *glitch energy*.

4 Signal integrity

This is a whole subject by itself and cannot possibly be properly dealt with in just a few pages. We shall merely give an overview. For more detailed treatment, refer to the literature [12, 13]. First some generalities: We distinguish between noise, which is an inherent property of the circuit components, and interference, which comes from elsewhere. Noise may be present on the input signal as an inherent property of the input transducer, or generated by components in the circuit itself. The term noise is in practice often used for what is actually interference. Examples of noise are the shot noise from a photodiode or the thermal noise of resistors. Examples of interference may be mains hum, power supply noise, or radio frequency leakage.

Interference is hard to deal with because it is usually determined by undesirable, parasitic circuit 'components' that do not appear explicitly in any schematic diagram, and that are usually considered negligible. Most often, it involves inductance and resistance of solid conductors, which are usually thought of as having the same potential throughout, or capacitive and inductive coupling between parts of the circuit that have no explicit components connecting them. This makes identifying the causes of interference difficult and requires a fair idea of the location and importance of these parasitic components.

On top of that, there is usually no catch-all solution that will solve all problems completely. Reducing one particular source of interference may aggravate another. Compromises must sometimes be made. However, in all but the most obstinate cases, an acceptable solution can usually be found.

There are three main coupling mechanisms by which interfering signals can get into a circuit:

Common impedance coupling: This is probably the most frequent cause of interference in any electronic circuitry. Conductors, wires or printed circuit board (PCB) traces have a finite non-zero impedance, both inductive and resistive. If two distinct circuits share a piece of wire or PCB trace, here modelled as Z_p (Fig. 54), the current flowing in one circuit will cause some fraction of its signal to be superimposed on the other. To reduce this effect, common current return paths should be generously dimensioned, to minimize Z_p. Bear in mind that the inductive part of Z_p is often dominant. On PCBs, this leads to using full-surface ground planes. Alternatively, you may consider giving some or all circuits their own individual return path to some common point (Fig. 55). Taken to the extreme, this results in a star topology. This latter strategy is most appropriate to low-frequency circuits.

Inductive coupling: If any closed loop is traversed by a changing magnetic flux, a voltage and/or current will be induced in that loop. Conversely, any loop carrying a current produces a flux (Fig. 56). Keep loop areas small, use twisted-pair wires or coax cable, and put continuous, uninterrupted ground planes under PCB tracks. Keep direct and return paths close together. Keep loops with high dI/dt well away from loops that carry sensitive low-level signals. Consider using differential signalling. Decoupling or bypass capacitors also serve to confine fast changing currents to small loops. Magnetic shielding is rarely practical, but at high frequencies, a simple conductive shield may be effective.

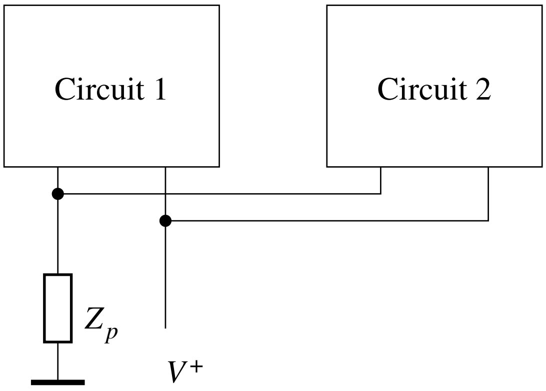


Fig. 54: Common impedance coupling

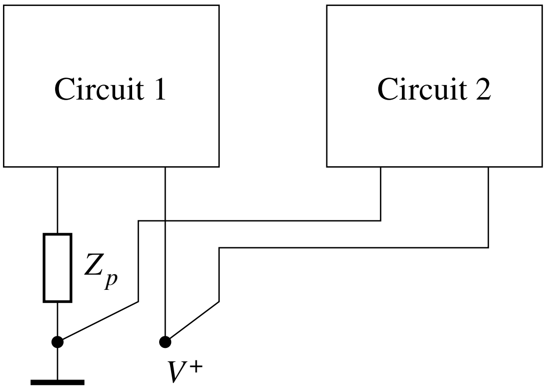


Fig. 55: Using a star layout to reduce common impedance coupling

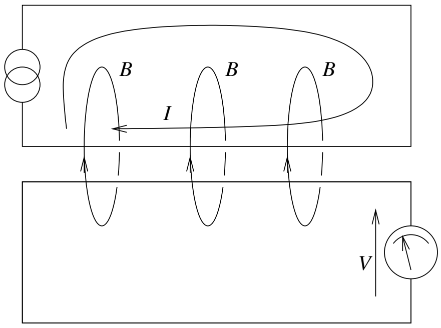


Fig. 56: Inductive coupling

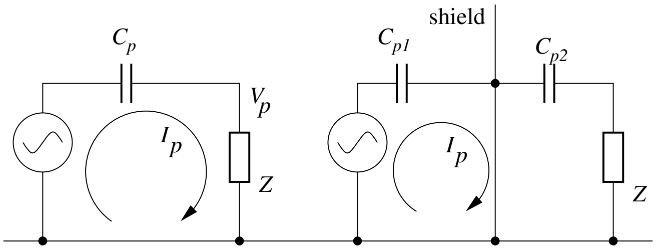


Fig. 57: Capacitive coupling and shielding

Capacitive coupling: Changing electric fields will also induce currents in nearby conductors (Fig. 57), effectively acting as parasitic capacitance C_p between circuit elements. This current produces a voltage V_p across victim impedance Z. Keep nodes with rapidly changing voltages compact. Keep high-impedance nodes far away from the first kind. If possible, reduce the $\mathrm{d}E/\mathrm{d}t$ of aggressor nodes and lower the impedance of victim nodes. Put grounded shields (coax, again) or guard tracks between them (so-called Faraday shields), so that the parasitic current I_p flows on the shield rather than through the victim impedance.

4.1 Input-signal conditioning

Many considerations guide the design of input-signal conditioning circuitry. This circuitry should adapt the input signal to the ADC input; filter, scale and offset it if needed, protect against out-of-bound inputs, reject common mode interference, etc. Many circuit topologies exist to deal with each of these requirements. Manufacturers will usually recommend some conditioning circuits with suitable device type numbers and component values.

A simple amplifier can buffer and possibly scale the signal to the ADC, presenting a high impedance to the signal source and a stiff source to the ADC (Fig. 58). A low-pass RC circuit isolates the amplifier from the capacitive ADC input and absorbs sampler kick-back. The amplifier may actually be an instrumentation amplifier (Fig. 59), in cases where considerable gain is required to reveal a small useful signal riding on a large common-mode voltage. This arrangement is good for low-frequency signals, as produced by Wheatstone-bridge-like temperature or force transducers, for example.

Recent ADCs often have differential inputs in order to retain a reasonable dynamic range in spite of ever lower supply voltages. A pair of operational amplifiers can be used to convert a single-ended signal

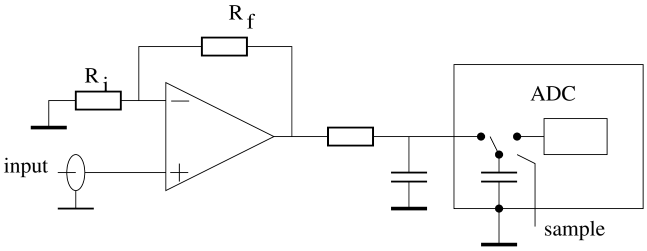


Fig. 58: Simple signal buffer

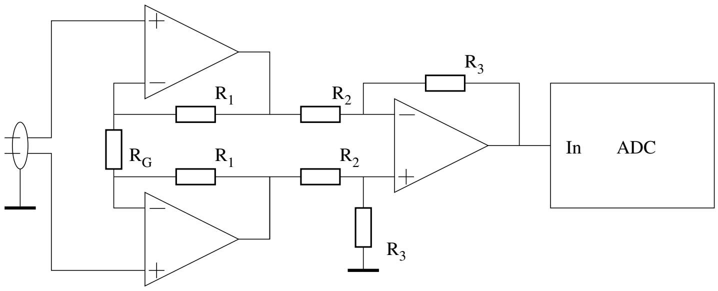
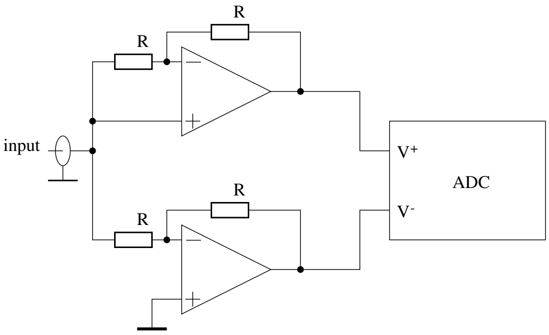


Fig. 59: Instrumentation amplifier

into a differential one (Fig. 60). A number of manufacturers produce monolithic differential amplifiers, which have both differential inputs and outputs for this sort of application (Fig. 61). (An example is the Analog Devices ADA4937.)



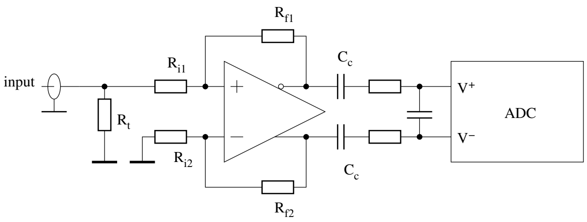


Fig. 60: Converting a single-ended signal to differential

Fig. 61: Using a fully differential buffer amplifier

High input impedance is not always desirable, as it is more susceptible to interference and requires shielding. For higher frequencies, inputs are usually terminated, in order to avoid reflecting signal back to the source. Termination consists of a resistor, of value equal to the characteristic impedance of the cable, across the input terminals. The frequency at which termination becomes necessary depends on the length of the cable between the signal source and the ADC.

For higher frequency signals, the single-ended to differential conversion can be obtained using baluns (Fig. 62) or transformers (Fig. 63). Even if the input signal is already differential, it may be beneficial to use a balun or transformer in order to reject common-mode interference. A transformer will not pass DC and low-frequency signals, whereas a balun will. Concerning common-mode rejection, a balun is poor at low frequencies, but it works very well at high frequencies. For the transformer, it is the reverse (due to unavoidable capacitive coupling between the windings). Both may be designed to match the input signal amplitude to best fit it into the ADC input range. Neither will prevent sampler kick-back from travelling back to the source.

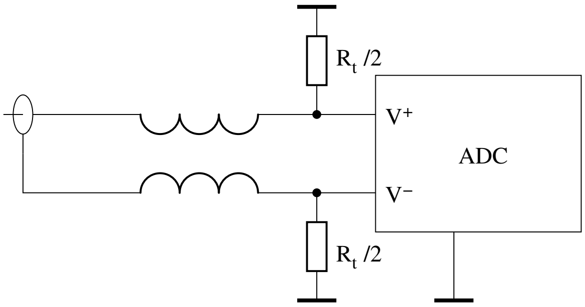


Fig. 62: Using a balun for single-ended to differential conversion

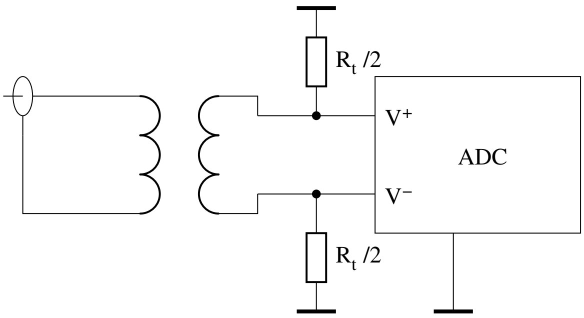


Fig. 63: Using a transformer for single-ended to differential conversion

4.2 Ground pins

Precision or high-speed ADCs have separate pins labelled AGND and DGND. Despite what is often believed, this does not mean that these pins should be connected to separate ground planes. In fact, AGND is the reference zero level for the analog part of the chip and DGND is the pin that carries the return current from the digital in- and outputs. These pins are brought out separately in order to reduce common impedance coupling between the analog and digital sections of the chip. Both should normally be connected to the same solid ground plane.

The often seen advice of splitting the ground plane into a digital and an analog section, with a single connecting link under the ADC, is not very practical. If both ground planes were also to be tied together at the power supply, this creates a loop with unknown geometry, inviting interference. Another loop may be formed by the screens of the cables carrying the input signal from the—probably grounded—source to the ADC. And where should the link go when your gadget has several ADCs and DACs?

A much better way is to use a single uninterrupted ground plane, and to route digital signals so that their return currents do not find their way into the analog sections. The unavoidable differences in ground potential between the acquisition system and the signal source should be dealt with in the signal conditioning at the ADC input, for example, using differential amplifiers, baluns or transformers.

The digital signals of the ADC itself are an important source of noise in any case. The layout should carefully keep digital, clock, and analog signals apart. For differential signals, pay attention to symmetry. Digital outputs should drive as light a load as possible. This will reduce the intensity of the return currents. Never connect an ADC to long, shared bus lines. Always connect it to a buffer first, using the shortest possible connections, and then connect the buffer outputs to the bus instead. It may be useful to insert small-valued series resistors ($\approx 50\ \Omega$) in the data lines between the ADC and the buffer to limit transient digital signal currents.

5 Conclusion

The digital revolution has afforded signal treatment with high fidelity. Even though ADCs cannot yet match the dynamic range of good analog circuitry, analog signals are increasingly to be found only at the very ends of the signal processing chain, the intent seemingly being to eliminate any remaining analog hardware altogether. There are a huge number of different A-to-D and D-to-A converters, with properties tailored to a variety of applications. Many manufacturers compete for a share of the market. The rate of development is furious, with many new, faster and better converters appearing on the market every year.

Appendix

A Decibels

Signal levels in electronics, and also in control system theory, are often specified in decibels (dB). In fact, the dB is a value that relates a given level to some specified or implied reference. The definition in terms of power levels is

$$\mathrm{dB} = 10 \cdot \log_{10} \frac{P}{P_r} \,. \tag{A.1}$$

Since the amplitude of a signal is proportional to the square root of its power, the definition in terms of amplitude is

$$\mathrm{dB} = 20 \cdot \log_{10} \frac{A}{A_r} \,. \tag{A.2}$$

Implied is the assumption that both signals are working into the same resistance. This is usually, but not systematically, adhered to by RF specialists, and totally ignored by control system engineers.

Often, some reference level is specified by one or more trailing symbols. For example, the ratio of a signal with respect to 1 mW is given in dBm. Other forms that appear frequently are dBV or dBμV, meaning dB with respect to a signal with r.m.s. value 1 V, respectively 1 μV. Also often encountered are dBc, meaning decibels with respect to the level of a carrier signal, or dBFS, meaning decibels with respect to a full-scale signal.

Even though the dB is one tenth of something called a *bel*, it is never used with other SI multiplier prefixes.

References

[1] B. Widrow, I. Kollár, Ming-Chang Liu, Statistical theory of quantization, *IEEE Trans. Instrum. Meas.*, Vol. 45, April 1996, pp. 353–361.

[2] E.T. Whittaker, On the functions which are represented by the expansions of the interpolation theory, *Proc. Royal Soc. Edinburgh*, Sec. A, Vol. 35, 1915, pp. 181–194.

[3] V.A. Kotelnikov, On the carrying capacity of the ether and wire in telecommunications, *Izd. Red. Upr. Svyazi RKKA*, Moscow, 1933 (Russian), `http://ict.open.ac.uk/classics/1.pdf`

[4] H. Nyquist, Certain topics in telegraph transmission theory, *Trans. AIEE*, Vol. 47, April 1928, pp. 617–644.

[5] R.V.L. Hartley, Transmission of information, *Bell Syst. Tech. J.*, Vol. 7, July 1928, pp. 535–563.

[6] C. Shannon, Communication in the presence of noise, *Proc. IRE*, Vol. 37, January 1949, pp. 10–21.

[7] W.R. Bennett, Spectra of quantized signals, *Bell Syst. Tech. J.*, Vol. 27, July 1948, pp. 446–472.

[8] B. Widrow, A study of the rough amplitude quantization by means of Nyquist sampling theory, *IRE Trans. Circ. Theory*, Vol. CT-3, 1956, pp. 226–276.

[9] R.M. Gray, Quantization noise spectra, *IEEE Trans. Inf. Theory*, Vol. 37, No. 6, November 1990, pp. 1220–1244.

[10] B. Drakhlis, Calculate oscillator jitter by using phase noise analysis – Part 1, *Microwaves & RF*, Vol. 50, No. 1, January 2001, pp. 82–90, 157.

[11] N. Thao, Asymptotic MSE law of n^{th}-order Σ-Δ modulators, *IEEE Trans. Circ. Syst. II: Analog and Digital Signal Processing*, Vol. 50, No. 5, May 2003, pp. 234–238.

[12] H. Ott, *Noise-Reduction Techniques in Electronic Systems*, 2nd ed., (Wiley, New York, 1988).

[13] R. Morrison, *Grounding and Shielding Techniques*, 4th ed., (Wiley, New York, 1998).