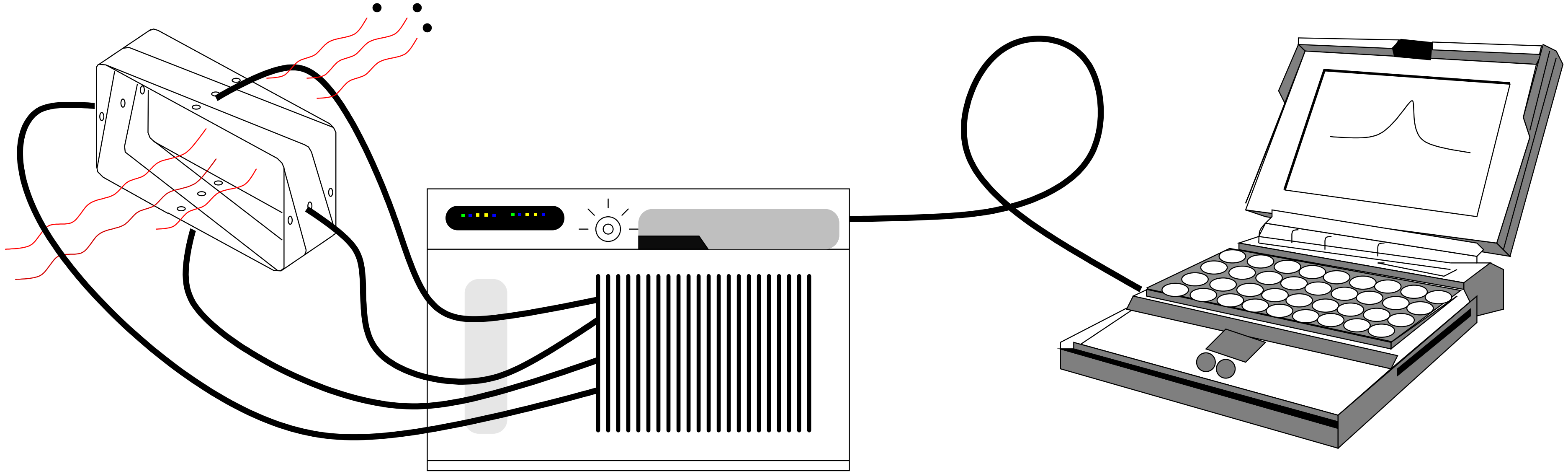
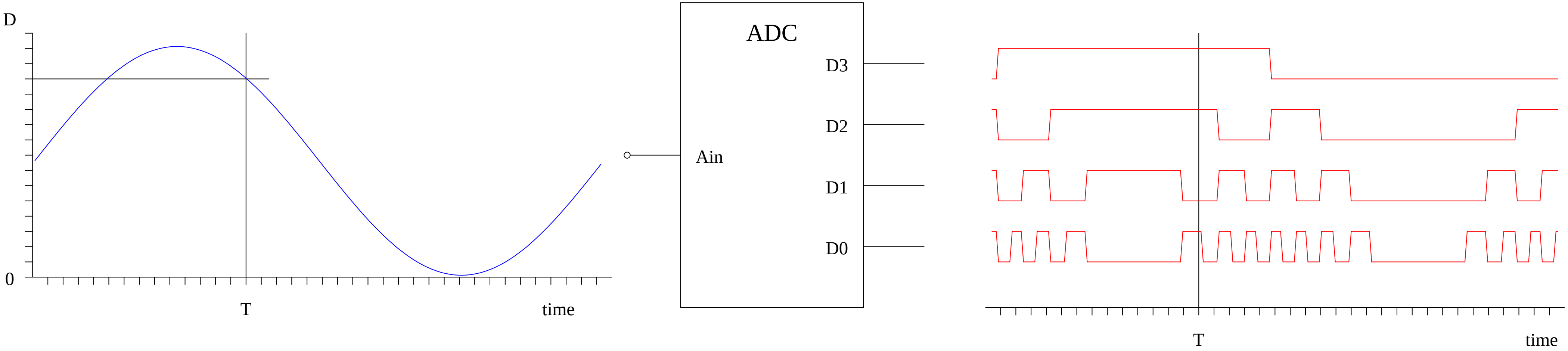


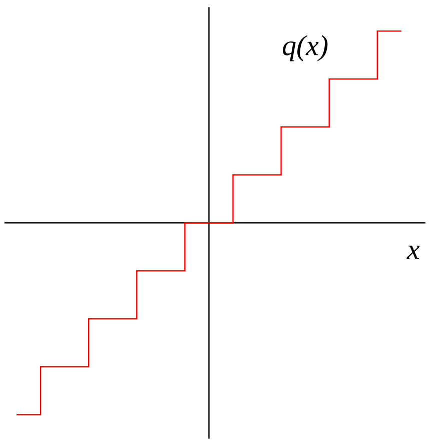
- Quantization of amplitude and time
- Noise, jitter & distortion
- ADC architectures
- Signal conditioning

## Analog to Digital Converter

An ADC converts a continuously variable signal, a voltage or a current, into a sequence of numbers, represented by logic levels on a group of wires.



# Amplitude quantization

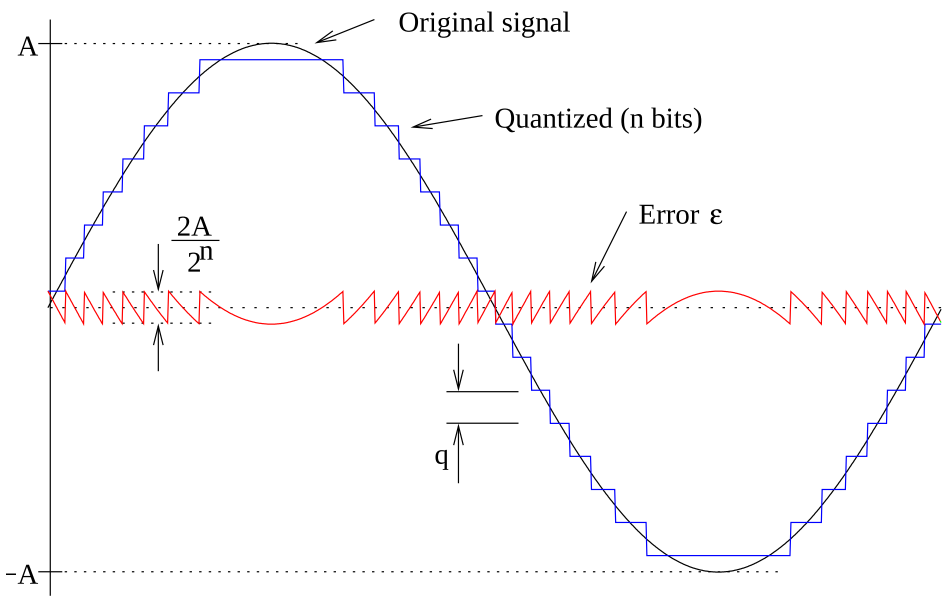



- Quantization replaces a range of continuous values by a set of discrete ones.
- Usually the number of levels is a power of 2.
- The difference between the original signal and the discrete representation is the quantization error.

# Amplitude quantization

Quantized to *n* bits, one quantum is:

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




Maximum quantization error:

$$\varepsilon=\frac{\pm q}{2}=\pm A\cdot 2^{-n}\left(with\ p(\varepsilon)\approx\frac{1}{q}\right)$$

Power in quantization error:

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

Power in original signal:

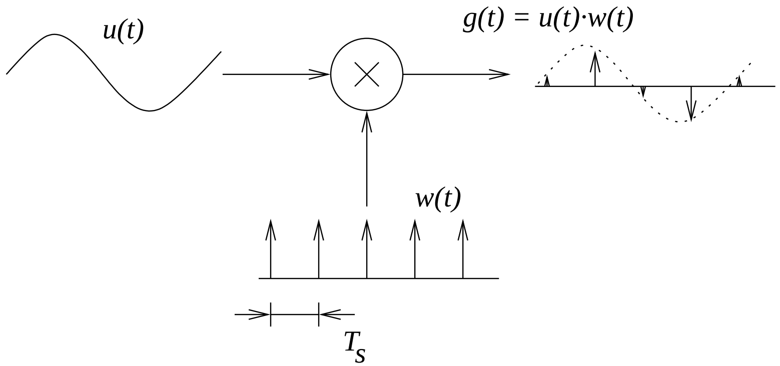
$$P_s=\frac{A^2}{T}\int_0^T \sin^2\omega t\,dt=\frac{A^2}{2}$$

Thus $SNR=\frac{P_s}{P_\varepsilon}=1.5\cdot 2^{2n}$

In dB: $10\cdot\log_{10}\frac{P_s}{P_\varepsilon}=1.76+6.02\,n$

# Time quantization or sampling

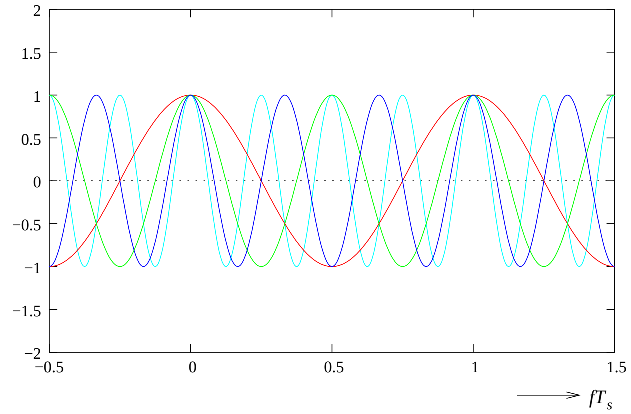
Multiplication of the signal by a train of impulses *w(t)* with period $T_s$ (= $1/F_s$):




$$g(t)=u(t)\cdot w(t)$$

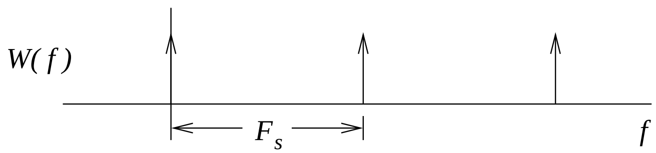
$$g(t)=u(t)\sum_{n=-\infty}^{\infty}\delta(t-nT_s)$$

$$g(t)=\sum_{n=-\infty}^{\infty}u(nT_s)\,\delta(t-nT_s)$$



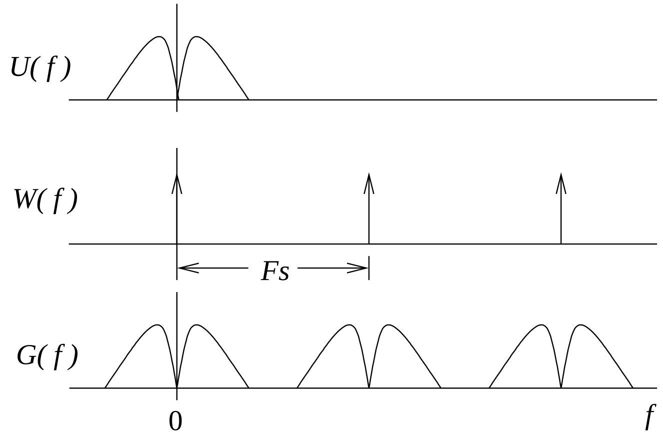

Fourier transform of *w(t)*:

$$W(f)=\int_{-\infty}^{\infty}w(t)\,e^{-j2\pi f t}\,dt$$




$$W(f)=\sum_{n=-\infty}^{\infty}e^{-j2\pi n f T_s}=1+2\sum_{n=1}^{\infty}\overbrace{\cos 2\pi n f T_s}=\sum_{n=-\infty}^{\infty}\delta\left(f-\frac{n}{T_s}\right)$$

The spectrum of the sampled signal
is the convolution of *U(f)* and *W(f)*




$$G(f)=U(f)*W(f)=\int_{-\infty}^{\infty} U(\phi)W(f-\phi)\,d\phi$$

$$G(f)=\int_{-\infty}^{\infty} U(\phi)\sum_{n=-\infty}^{\infty}\delta\left(f-\frac{n}{T_s}-\phi\right)d\phi$$

$$G(f)=\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} U(\phi)\delta\left(f-\frac{n}{T_s}-\phi\right)d\phi$$

$$G(f)=\sum_{n=-\infty}^{\infty} U\left(f-\frac{n}{T_s}\right)$$

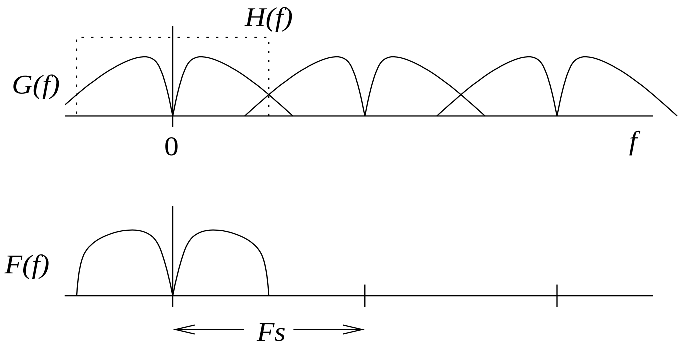
After sampling, the signal spectrum repeats for all multiples of $F_s$

If the sampling rate $F_s$ is less than twice the signal bandwidth, the spectral images overlap. The avoid this, the following condition must be fulfilled:

$$F_s > 2 \cdot BW$$

<u>This is the Nyquist criterion</u>

One way this condition can be fulfilled is by filtering the analogue signal prior to digitizing it, using what is called an anti-aliasing filter. Since brick-wall filters cannot be made, the sampling rate should usually be comfortably greater than twice the signal bandwidth.

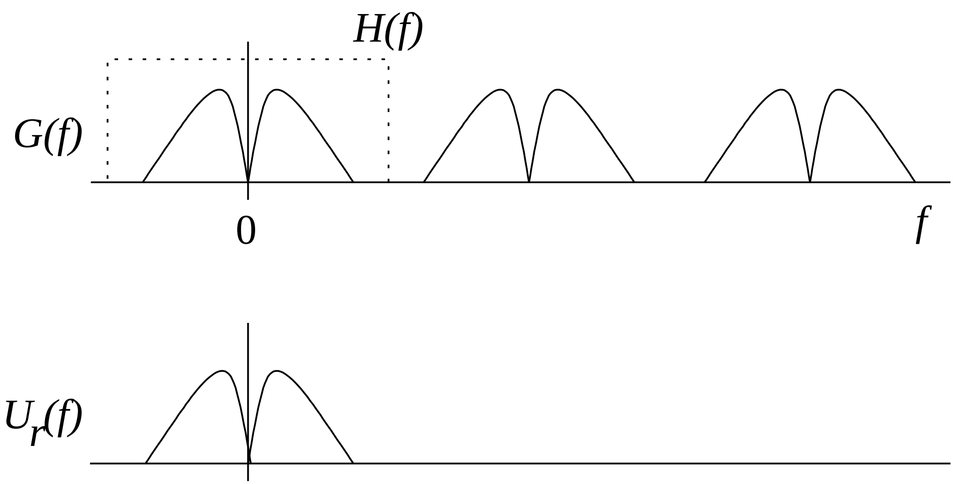


# Reconstruction of the original signal

Each of the images in the spectrum of the sampled signal contains all the information needed to reconstruct the original. They are *aliases*.

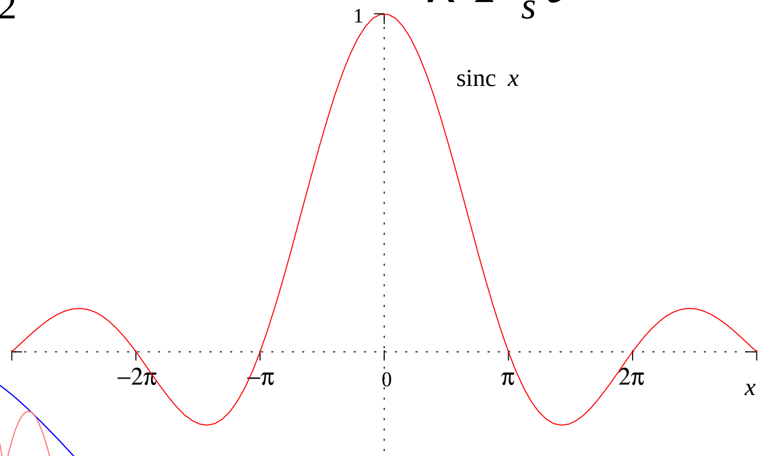
We might reconstruct the original signal with a filter that rejects everything except the original frequency band. After filtering, the spectrum is exactly that of the original signal, in other words, no information is lost. We have recovered the original signal exactly.

This is Shannon's theorem



Filter the baseband using a rectangular filter *H(f)*. The filter time-domain response is the inverse Fourier transform of its frequency-domain shape:

$$h(t)=\mathscr{F}^{-1}\{H(f)\}=\int_{-\infty}^{\infty}H(f)e^{j2\pi f t}df=\int_{-F_s/2}^{F_s/2}e^{j2\pi f t}df=F_s\frac{\sin\pi F_s t}{\pi F_s t}$$



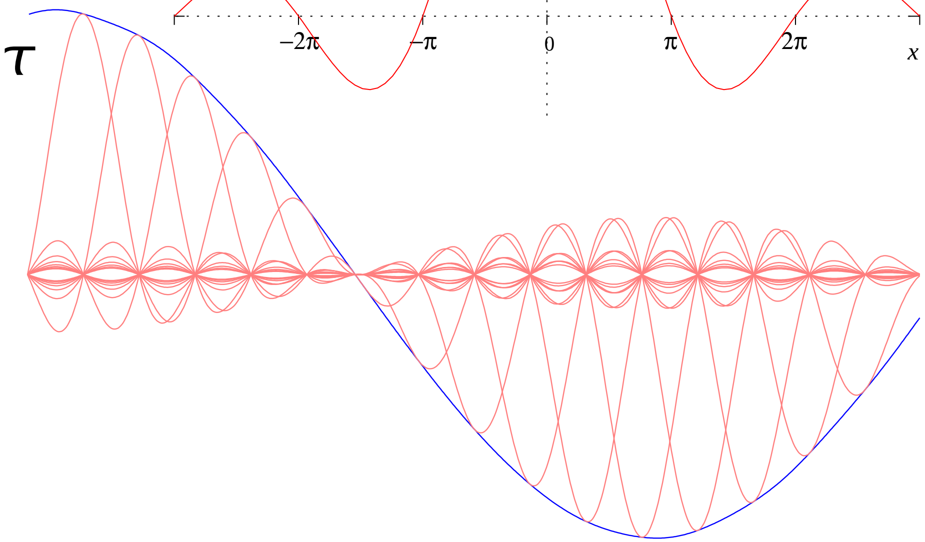

Convolution of filter with sample stream:

$$u_r(t)=g(t)*h(t)$$

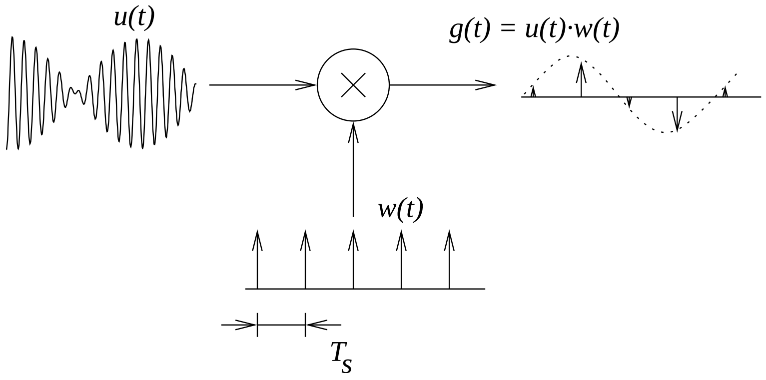
$$u_r(t)=\int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty}u(\tau)\cdot\delta(\tau-nT_s)\cdot h(t-\tau)\,d\tau$$

$$u_r(t)=\sum_{n=-\infty}^{\infty}u(nT_s)\cdot h(t-nT_s)$$

$$u_r(t)=\sum_{n=-\infty}^{\infty}u(nT_s)\cdot F_s\frac{\sin\pi F_s(t-nT_s)}{\pi F_s(t-nT_s)}$$

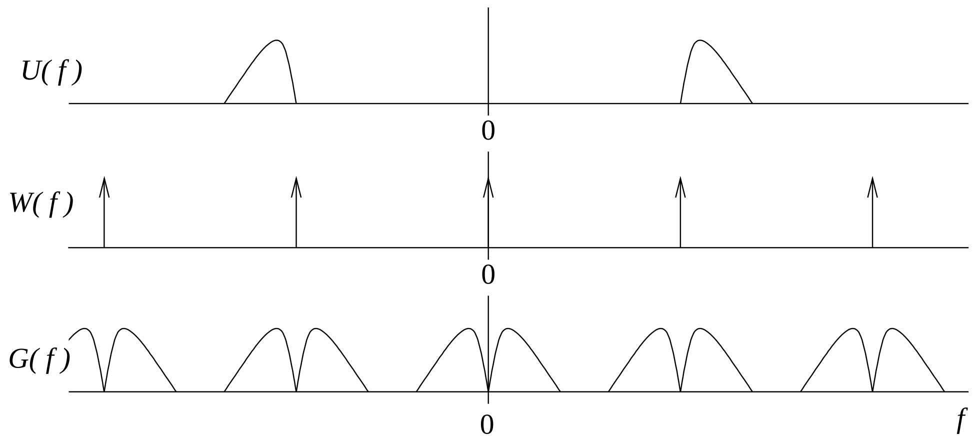


Note that exactly the same spectrum results for any signal frequency band displaced by $m \cdot F_s$ (for integer $m$)




This goes by the name of *sub-sampling*

$$G(f) = \int_{-\infty}^{\infty} U(\Phi + mF_s) W(f - \Phi)\, d\Phi$$



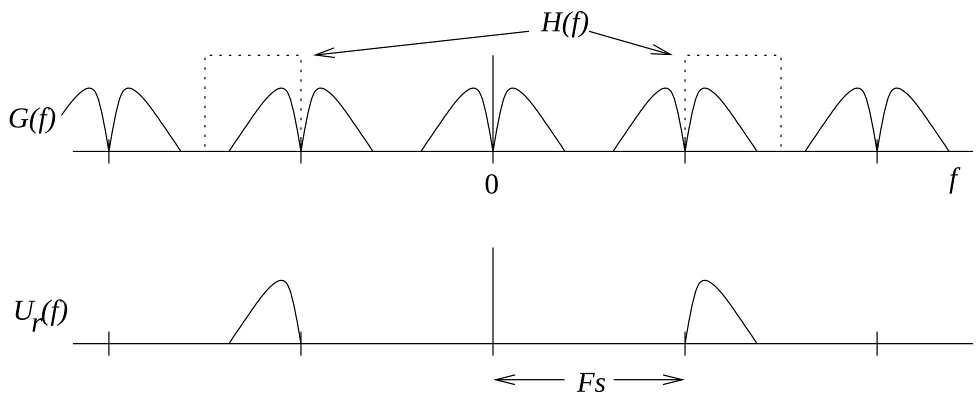

$$G(f) = \sum_{n=-\infty}^{\infty} U\left(f - \frac{n}{T_s} + mF_s\right)$$

$$G(f) = \sum_{n=-\infty}^{\infty} U\left(f + \frac{m-n}{T_s}\right)$$

$$G(f) = \sum_{n=-\infty}^{\infty} U\left(f - \frac{n}{T_s}\right)$$

We could also choose a different spectral image to (re)construct the signal:

First work out the time-domain representation of the filter:

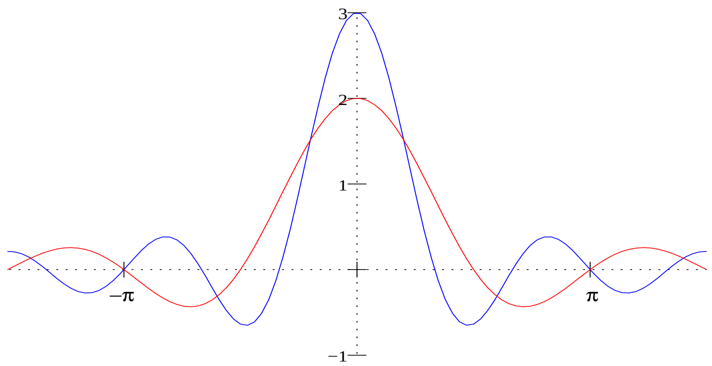



$$H(f)=1 \quad \text{for} \quad F_s<|f|<\frac{3}{2}F_s$$

$$H(f)=0 \quad \text{everywhere else}$$

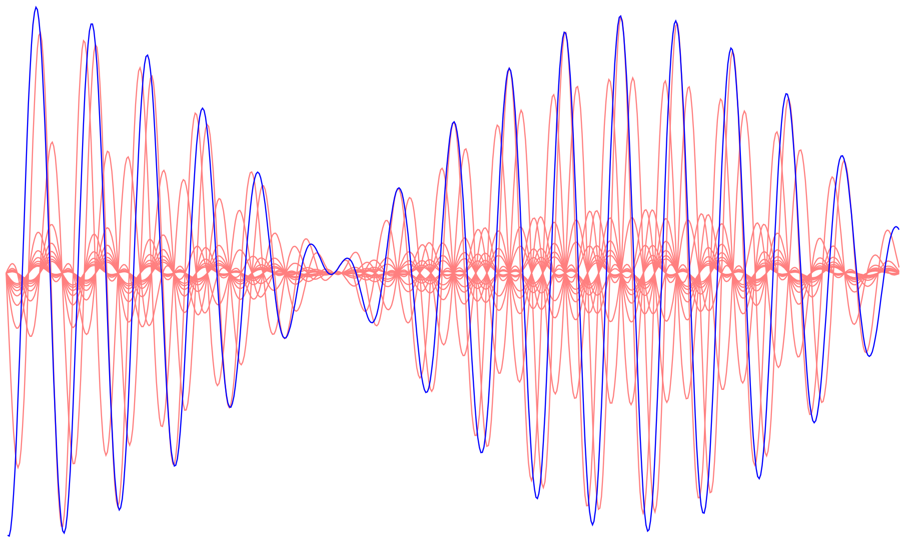
$$h(t)=\mathscr{F}^{-1}\{H(f)\}=\int_{-\infty}^{\infty} H(f)e^{j2\pi f t}df$$

$$h(t)=\int_{\frac{-3}{2}F_s}^{\frac{3}{2}F_s} e^{j2\pi f t}df-\int_{-F_s}^{F_s} e^{j2\pi f t}df$$




$$h(t)=3F_s\text{sinc}(3\pi t F_s)-2F_s\text{sinc}(2\pi t F_s)$$

Then convolve the sample stream
with the filter function:



$$u_r(t) = g(t) * h(t)$$

$$u_r(t) = \int_{-\infty}^{\infty} g(\tau) \cdot h(t-\tau)\, d\tau$$

$$u_r(t) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} u(\tau) \cdot \delta(\tau - nT_s) \cdot h(t-\tau)\, d\tau$$

$$u_r(t) = \sum_{n=-\infty}^{\infty} u(nT_s) \cdot h(t - nT_s)$$

$$u_r(t) = \sum_{n=-\infty}^{\infty} u(nT_s) \{ 3F_s \operatorname{sinc}(3\pi(t-nT_s)F_s) - 2F_s \operatorname{sinc}(2\pi(t-nT_s)F_s) \}$$

# Some history

1915: E.T. Whittaker : Interpolation theory

First use of the sinc function for the interpolation of bandwidth-limited functions

1928: H. Nyquist : Telegraph transmission theory

Classic! Deals with signal distortion in transmission channels like undersea cables, which were a hot subject, at the time.

1933: V.A. Kotelnikov : Carrying capacity of the ether

Detailed demonstration that band-limited signals can be represented by a sum of sinc functions, apparently independently from Whittaker.

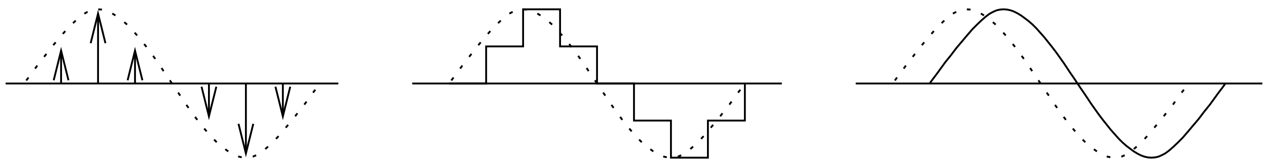
1949: C. Shannon : Communication in the presence of noise

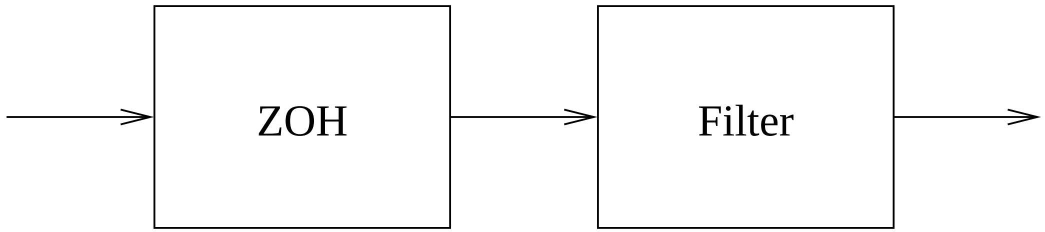
Classic! Gives transmission capacity of a channel as a function of bandwidth and signal to noise ratio. The sampling theorem is dealt with in section II.

Other names: R.V.L Hartley, J.M. Whittaker, C-J. de la Vallée Poussin, H. Raabe, Ogura, I. Someya, Weston ...

Although mathematically Dirac deltas, brick-wall filters and infinite sums are quite nice to handle, in real electronic circuitry, you can't have them.

The Dirac δ is replaced by an (almost) rectangular pulse of one sampling period duration, and filters are described by finite polynomials, with finite-slope band edges. So, the output is held constant during each sampling period, which is functionally called a zero-order hold (ZOH), and a low-pass filter smooths over the steps.



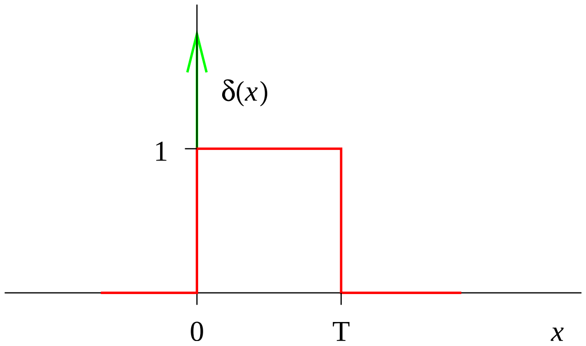


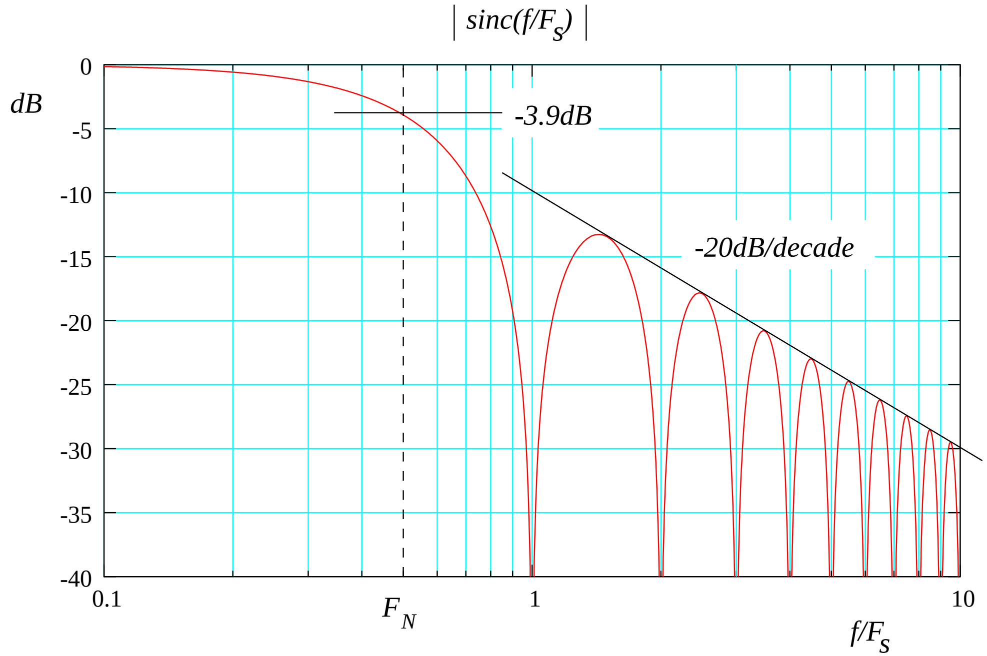
# Practical signal reconstruction

As a consequence, the reconstructed signal spectrum is convolved with a sinc($f/F_s$) function and some energy from adjacent spectral images leaks into the desired band. Note that at the Nyquist frequency, $F_N = F_s/2$, the response is down by 3.9dB.

If this is a problem, the reconstruction filter may be designed to compensate. (You can also pre-compensate in the digital domain.)

$$H_{ZOH}(f) = \frac{1 - e^{-j2\pi f T_s}}{-j2\pi f} = T_s e^{-j\pi f T_s} \operatorname{sinc} \pi f T_s$$

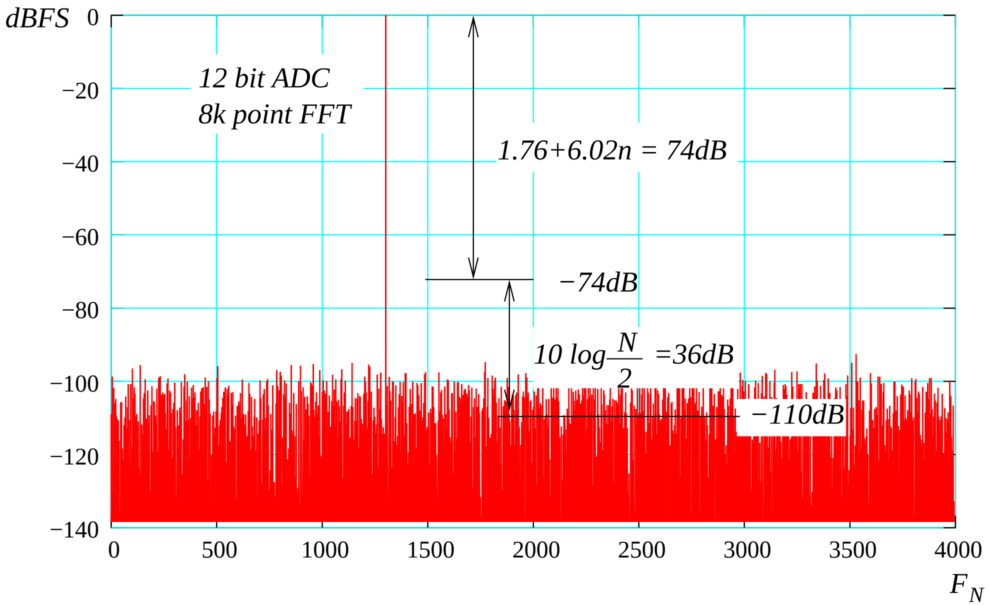


# Spectrum of quantization error

For 'large enough' and 'busy enough' signals, the quantization error is a random variable with a flat distribution.

→ Quantization noise is white and spread out evenly over $0<f<F_s/2$.

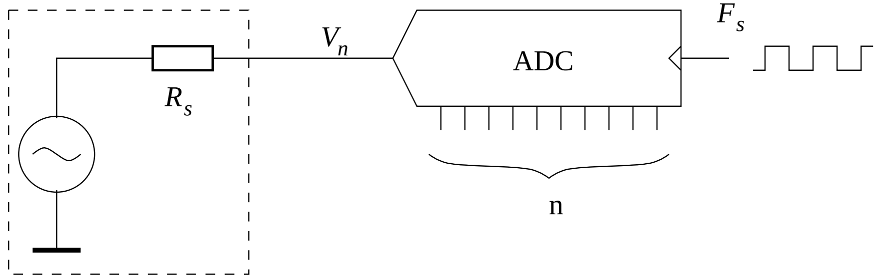



FFT Noisefloor: $$-\left(1.76+6.02\,n+10\log_{10}\frac{N}{2}\right)dBFS$$

(N = number of samples)

# ADCs and thermal noise

Signal source noise: $V_n^2 = 4kTBR_s$

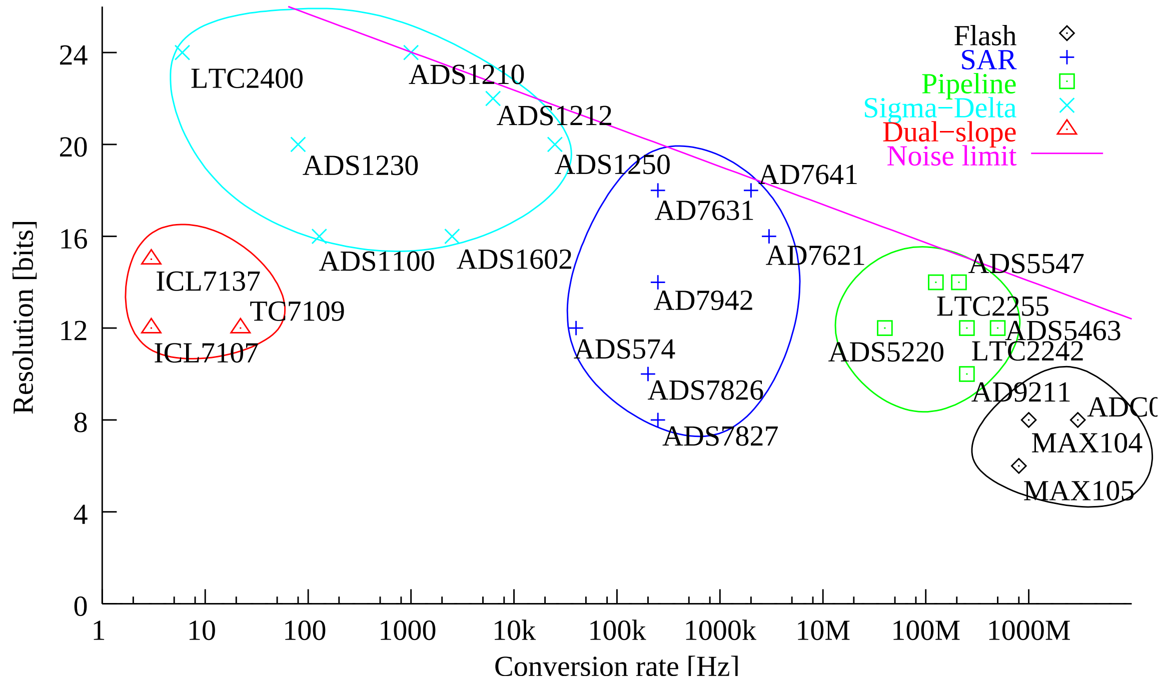



ADC quantization noise: $V_q^2 = \dfrac{A^2 2^{-2n}}{3}$

$V_n^2 = V_q^2$ and solve for n: $n = \dfrac{1}{2}\log_2 \dfrac{A^2}{12\,kTBR_s}$

Say, $T$=300K and $R_s$=50 Ω:

$$n = 29 - \frac{1}{2}\log_2 B + \log_2 A$$




The continuous line in the plot is for $A$=1

# ADC Noise figure

$$F = \frac{Total\,noise}{Source\,noise}$$

(Often expressed in dB.)

Example: The LTC2255, 14 bits, 125MS/s, SNR=72.4dB, ENOB 11.7 bits, 2Vpp full scale. Assume B = Fs/2 (Nyquist BW).

$$n = 29 - \frac{1}{2}\log_2 B + \log_2 A = 29 - 13 + 0 = 16\,bits$$

Noise limited SNR: $1.76 + 6.02 \cdot 16 = 98\,dB$

F = 98-72.4 = 25.6dB

# Spurious Free Dynamic Range (SFDR)

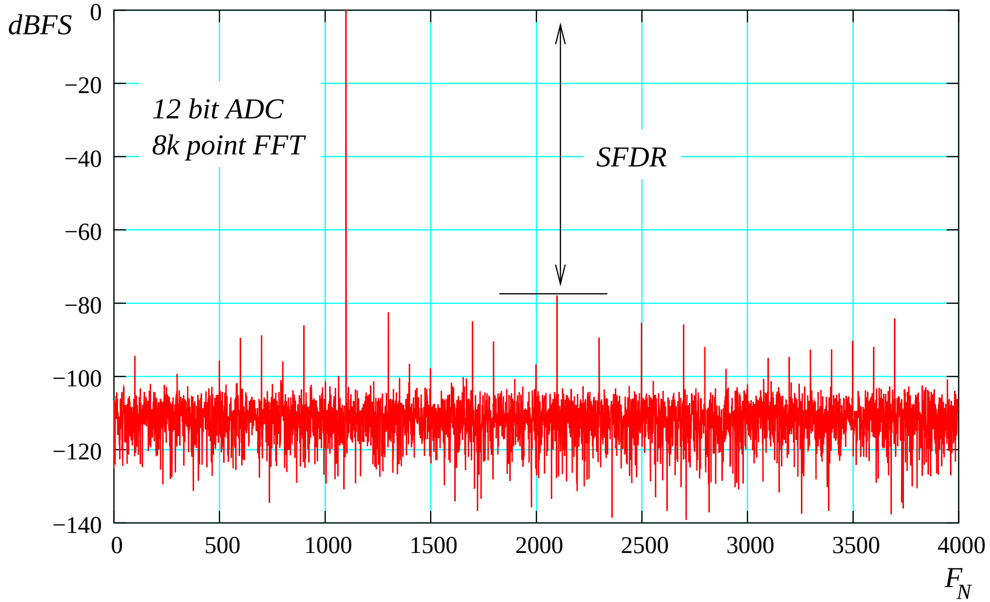
Unfortunately, quantization noise isn't always white:

- Simple ratios between $F_{in}$ and $F_s$ cause some of the quantization noise power to concentrate in discrete spectral lines
- ADC non-linearities cause harmonics of the input signal

Spurs appear in the spectrum:

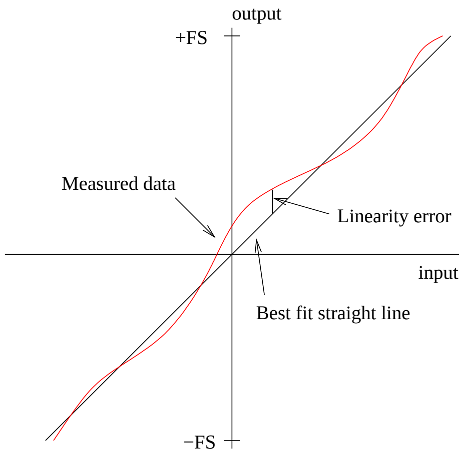
SFDR is the distance between the input signal and the greatest spur.

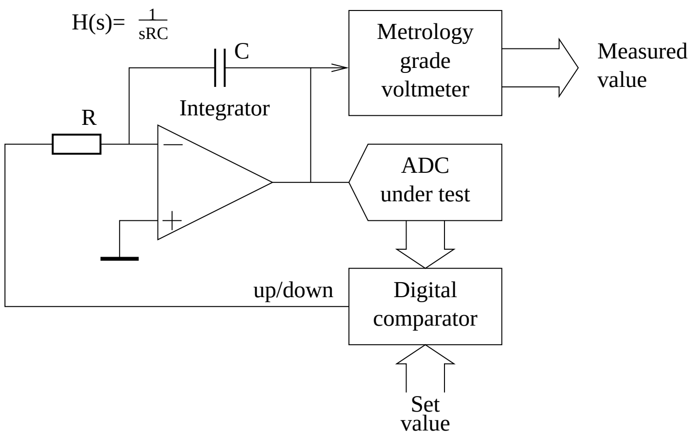
A little bit of dither can help to reduce spurs.

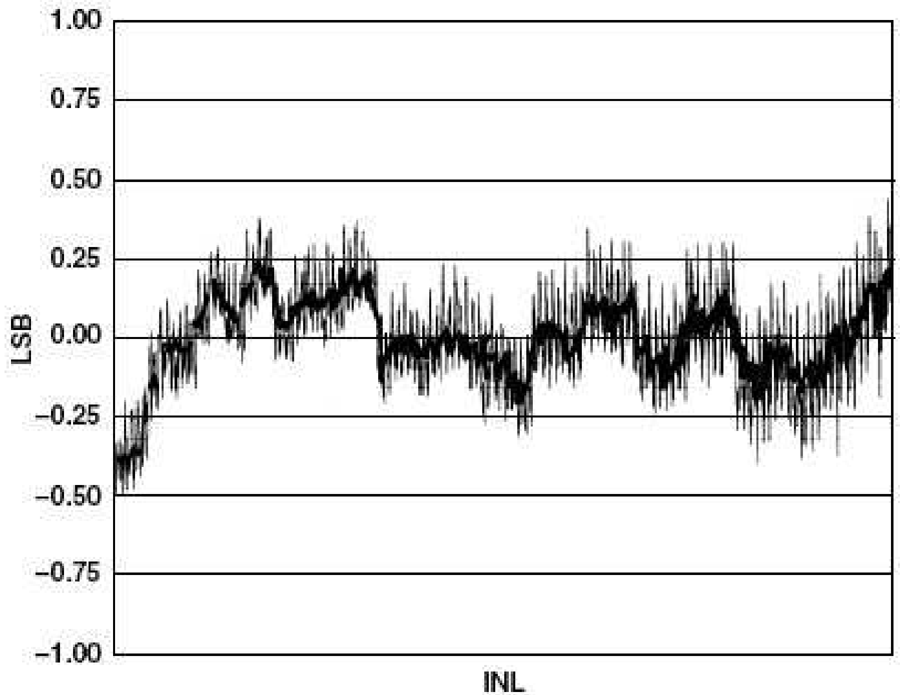



(Dither is the intentional injection of a little bit of noise.)

INL measures the deviation of the ADC characteristic from a straight line through the end points (or sometimes from a least squares fit)



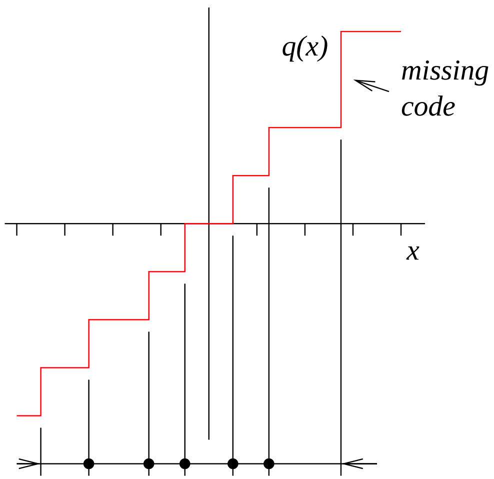



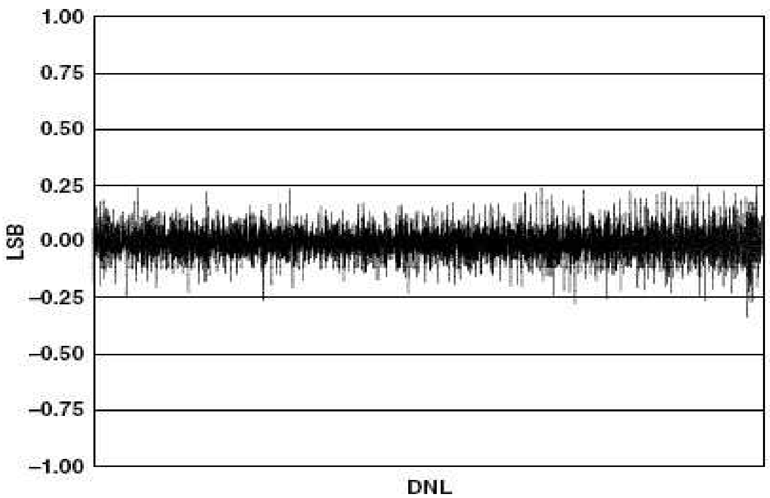

AD9432 12-bit 105MS/s pipeline ADC

# Differential Non-Linearity (DNL)

In real ADCs, the quantization function isn't perfectly uniform

For an input signal with a uniform distribution, the distribution of output values is no longer uniform.
The DNL measures the normalized error of the nominal size of each quantization step.




AD9432 12-bit 105MS/s pipeline ADC

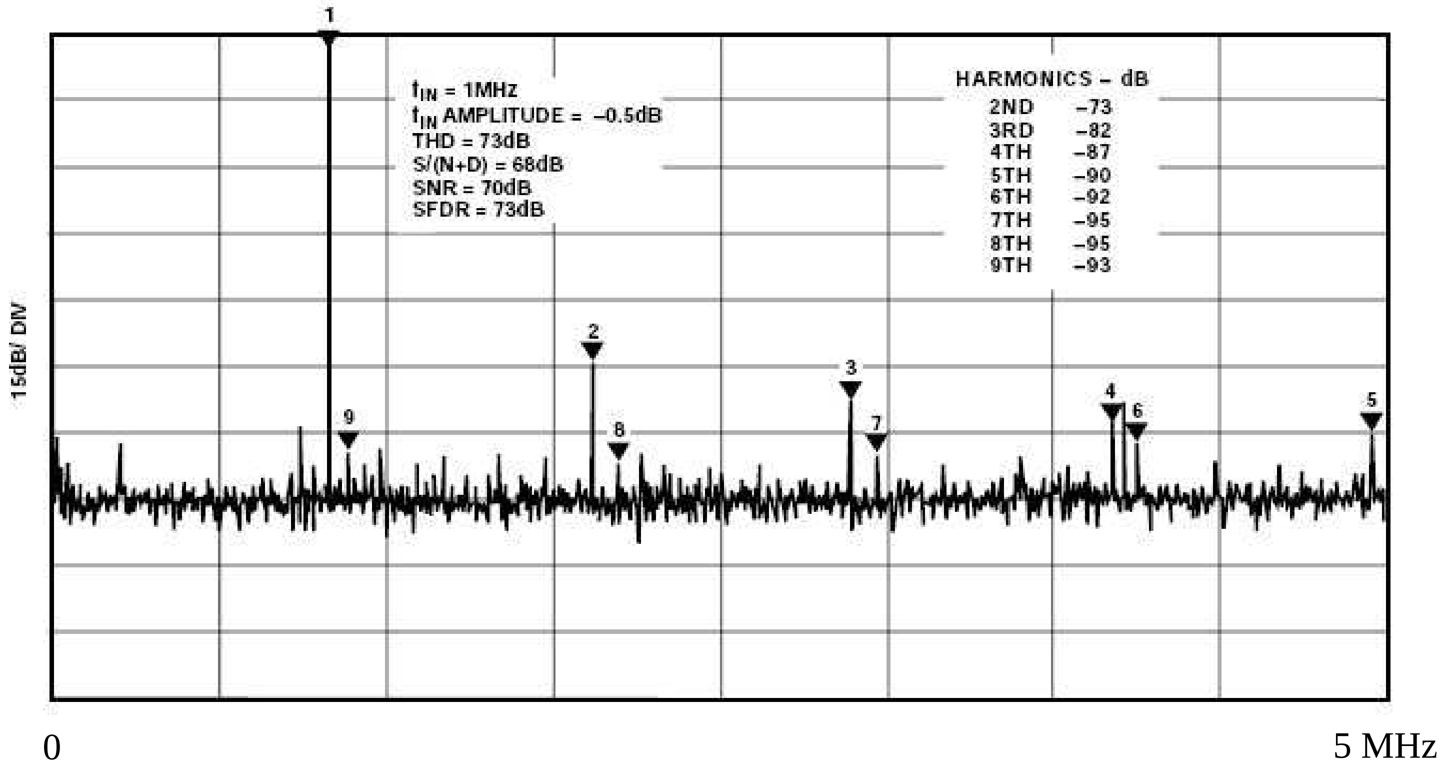
Missing codes

Non-monotonicity

# Total Harmonic Distortion (THD)

Non-linearity creates harmonics

THD is the rms sum of the first 6 harmonics compared to the input signal, in dB

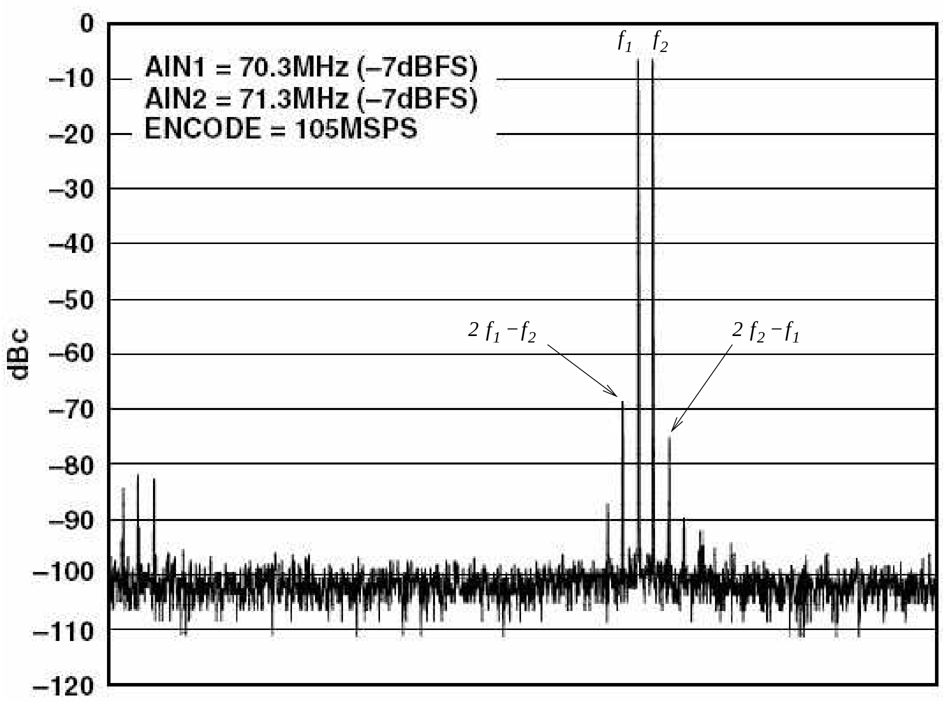



AD872A FFT plot , $f_{in}$=1 MHz, -0.5dBFS

Non-linearity also causes inter-modulation distortion. (Creating sum and difference frequencies from two applied tones $f_{imd} = \pm n f_1 \pm m f_2$ .)

IMD is the rms sum of the inter-modulation products compared to the rms sum of the input signals (usually in dB).

The order of an IMD product is |n|+|m|




AD9432, IMD FFT plot

# Effective Number Of Bits (ENOB):

Apply a (nearly) full-scale sinusoidal signal.
Measure $P_\varepsilon$, as the sum over all frequencies, ignoring DC, the input signal and the first five harmonics, and solve for *n* after substitution into the formula for the SNR of an ideal ADC:

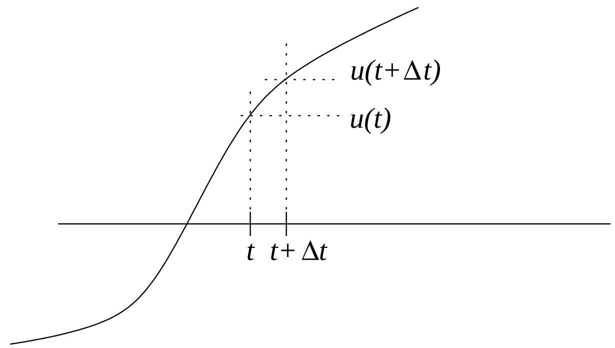
$$SNR = \frac{P_s}{P_\varepsilon} = 1.5 \cdot 2^{2n}$$

If you choose to also add in all harmonics into the calculation of $P_\varepsilon$, you would get the SINAD.
(SIgnal over Noise-And-Distortion)
(Which looks a little bit worse, of course)

SNR and SINAD are usually expressed in dBc or dBFS

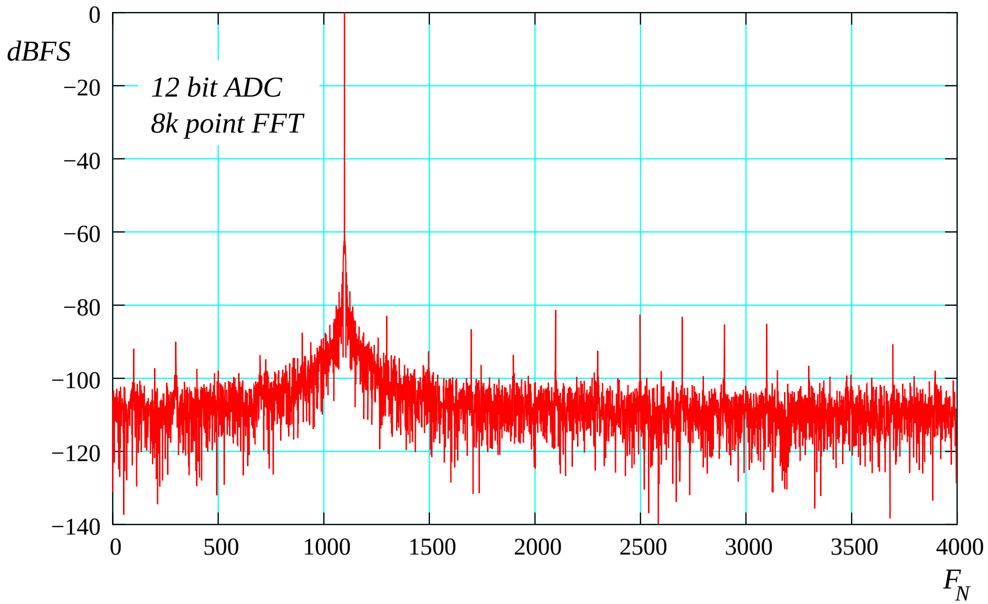
# Clock jitter

Importance of clock jitter depends on rate of change of <u>analogue input</u> signal




A clock timing noise *Δt* yields an amplitude error:

$$\Delta u = \frac{d\,u(t)}{dt} \cdot \Delta t$$



# The effect of clock jitter

Assuming noise due to jitter should not exceed quantization noise:

$$\Delta t < A \cdot 2^{-n} \frac{dt}{d\,u(t)}$$

This is a severe condition!

Example:
Suppose we want to digitize a 100MHz sinusoid to 10 bits:

$$u(t) = \sin 2\pi 10^8 t \quad \Rightarrow \quad \frac{d\,u(t)}{dt} = 2\pi 10^8 \cos 2\pi 10^8 t$$

So at the steepest slope:

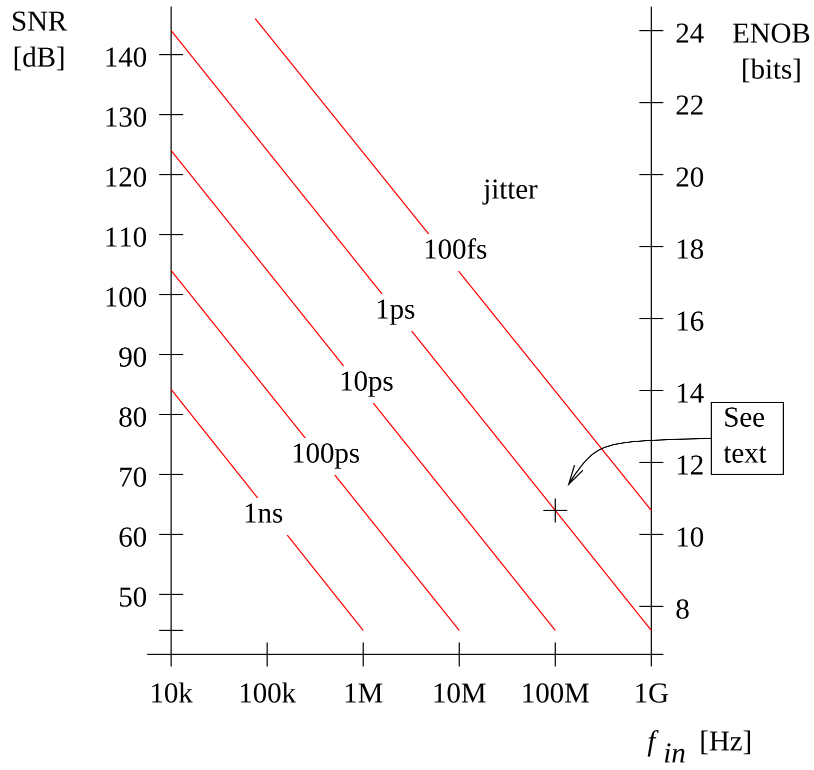
$$\Delta t = \frac{2^{-10}}{2\pi 10^8} \approx 1.6\,ps \quad \Rightarrow$$

A good quartz or ceramic resonator oscillator is needed

# Clock jitter limits vs. SNR and $f_{in}$

$$SNR = 20 \log_{10} \frac{1}{2\pi f \Delta t}$$

$$ENOB = \log_2 \frac{1}{2\pi f \Delta t}$$



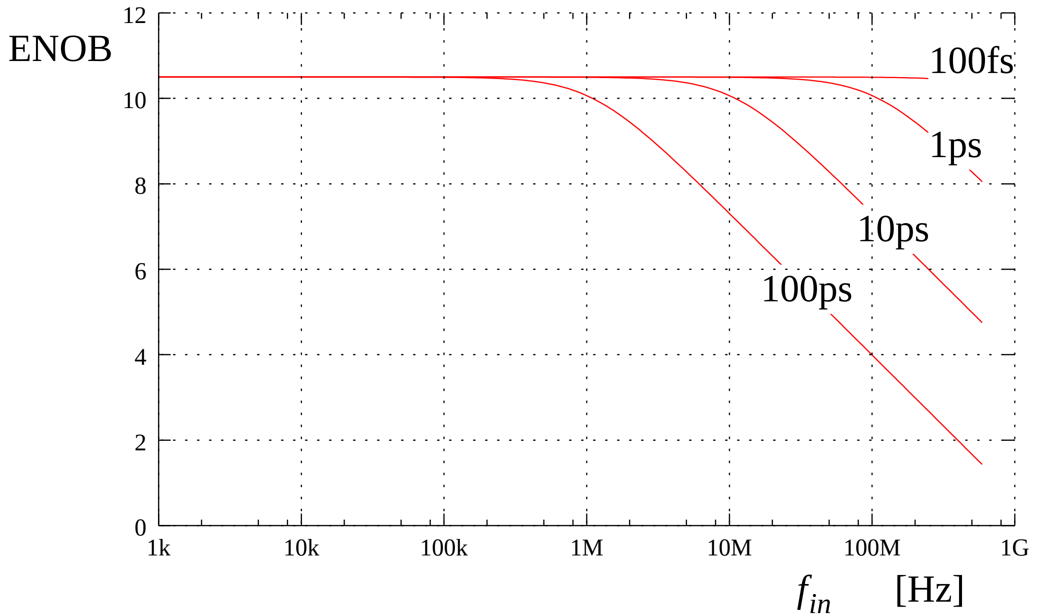

With 1 ps of jitter, a 100 MHz signal can only be digitized to 10.5 bits

Relaxed by root(decimation ratio) for Σ-Δ converters

# Clock jitter limits vs. SNR and $f_{in}$

In conclusion, the resolution of an ADC is bounded by its SNR or ENOB rating, by the clock jitter and by the analogue input signal frequency.

For example, for an actual 12-bit ADC, the ENOB vs. $f_{in}$ plot with clock jitter as the parameter might look like this:



# Calculating jitter from oscillator phase noise

Clock signal: $V(t)=\sin(2\pi F_s+\varphi(t))$

Significant instants: $2\pi F_s t_1+\varphi(t_1)=0$ and $2\pi F_s t_2+\varphi(t_2)=2\pi N$

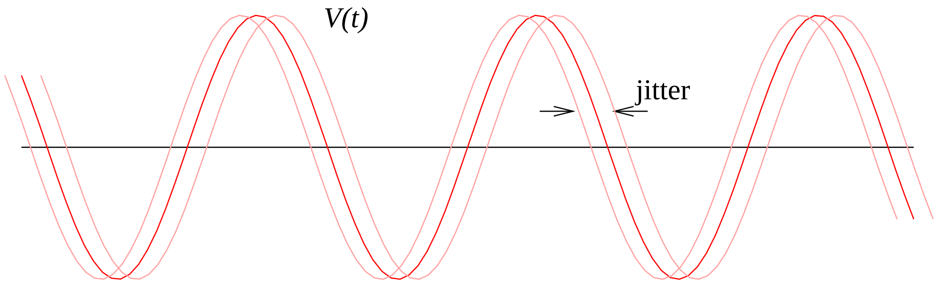
$$2\pi F_s(t_2-t_1)+\varphi(t_2)-\varphi(t_1)=2\pi N$$

The time between these instants is an integer number of periods plus some jitter:

$$t_2-t_1=\frac{N}{F_s}+\Delta t$$

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

$$\Delta t=\frac{1}{2\pi F_s}(\varphi(t_1)-\varphi(t_2))$$



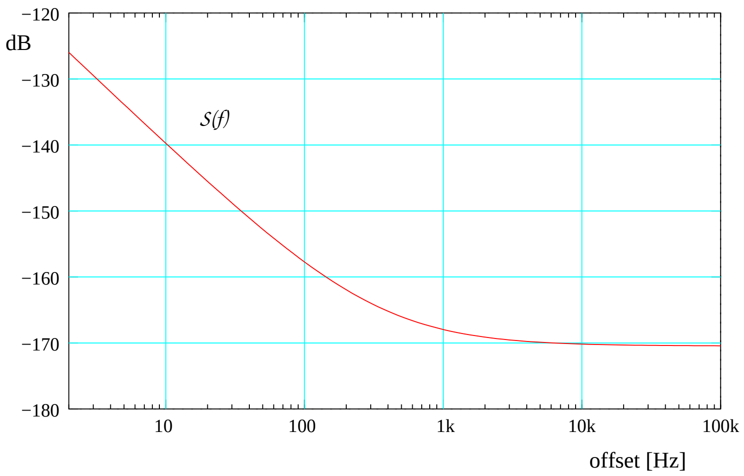
# Phase noise to jitter - 2

Expected value of variance:

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

Jitter does not depend on chosen instant:

$$\langle \varphi(t_1)^2 \rangle = \langle \varphi(t_2)^2 \rangle = \langle \varphi(t)^2 \rangle = \int_0^\infty S_\varphi(f)\, df$$



Oscillator phase noise spectral density

Variance of middle term is a cross covariance: Use cosine transform:

$$\langle \varphi(t_1) \varphi(t_2) \rangle = \int_0^\infty S_\varphi(f) \cos(2\pi f \tau)\, df \quad \text{with} \quad \tau = t_1 - t_2$$

# Phase noise to jitter - 3

Total variance of jitter:

$$\langle \Delta t^2 \rangle = \frac{1}{2\pi^2 F_s^2} \int_0^{\infty} S_\varphi(f)(1 - \cos(2\pi f \tau))\, df$$

$$\langle \Delta t^2 \rangle = \frac{1}{\pi^2 F_s^2} \int_0^{\infty} S_\varphi(f)(\sin^2(\pi f \tau))\, df$$

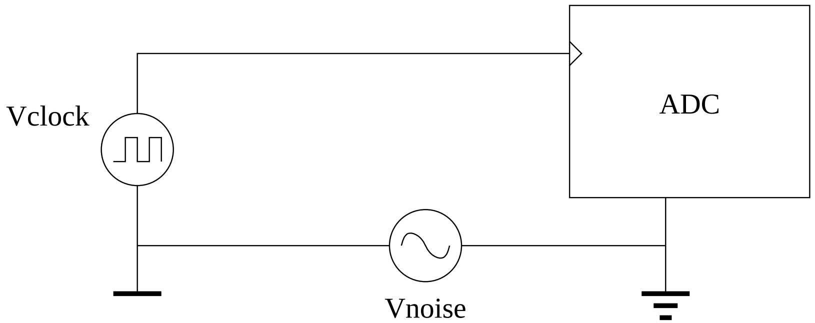
RMS jitter

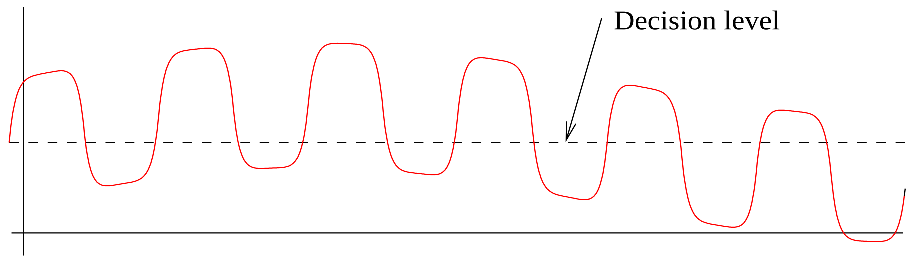
$$\langle \Delta t \rangle = \frac{1}{\pi F_s} \sqrt{\int_0^{\infty} S_\varphi(f)(\sin^2(\pi f \tau))\, df}$$

Integration bounds are set by measurement time for the lower bound and by sampler bandwidth for the upper bound.

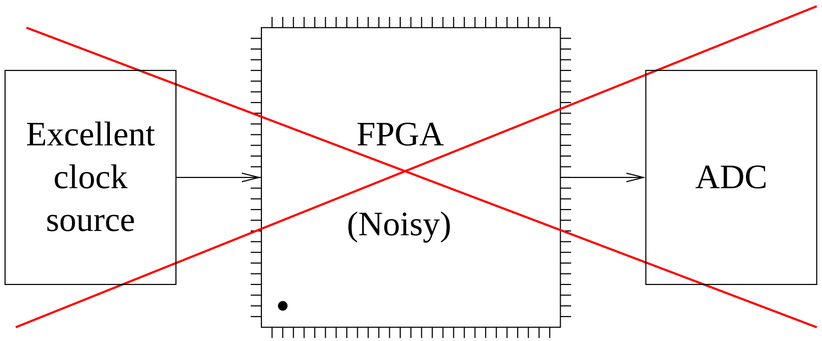
# Clock jitter and signal integrity

## Ground noise between clock source and ADC aggravates jitter



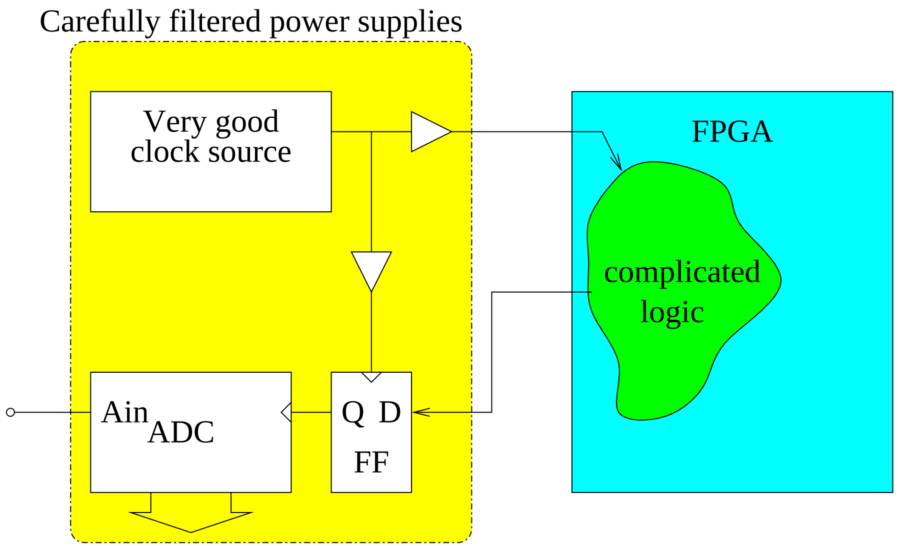




Noise may be due to magnetic interference or common impedance coupling. Possible remedy: Differential clock.




Don't route clocks through FPGAs!

(Complex logic circuits cause all sorts of interference)




But if you must, then resynchronize with the original clean clock

# Clock jitter summary

- ADCs (and DACs too!) need good quality clock sources
- Digital electronics is not optimized for low crosstalk
- PLLs in FPGAs usually have *very* poor jitter specs

Treat your clock like a sensitive analogue signal

- Filter and bypass clock generation & distribution power supply extra carefully
- Keep PCB layout tight and compact, minimize loop areas
- Refer clock source to the same GND as the ADC
- Do not route an ADC clock through an FPGA
- Don't use left-over gates in clock buffer package for other purposes

# ADC architecture

| *Architecture* | *Speed* | *Resolution* | *Linearity* | *Applications* |
|---|---|---|---|---|
| Flash | Very fast (GS/s) | Poor (8 bits) | Poor | Oscilloscopes<br>Transient recorders |
| Successive approximation | Fast (MS/s) | Fair (14 bits) | Fair | DSP, video, 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 |

## Other architectures:

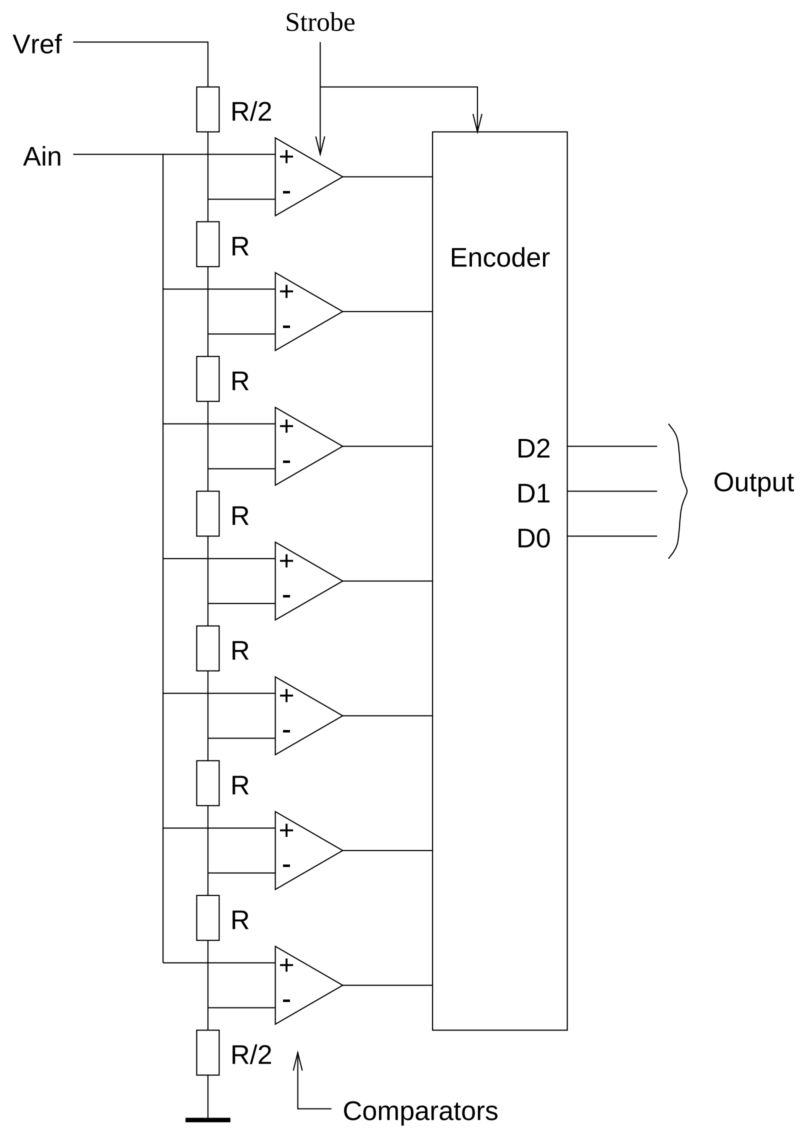
- Mixed forms (E.g. flash with SA, or flash with Σ-Δ)
- Tracking ADC
- Voltage-to-frequency converters

# The flash ADC

Flash ADCs are the fastest:

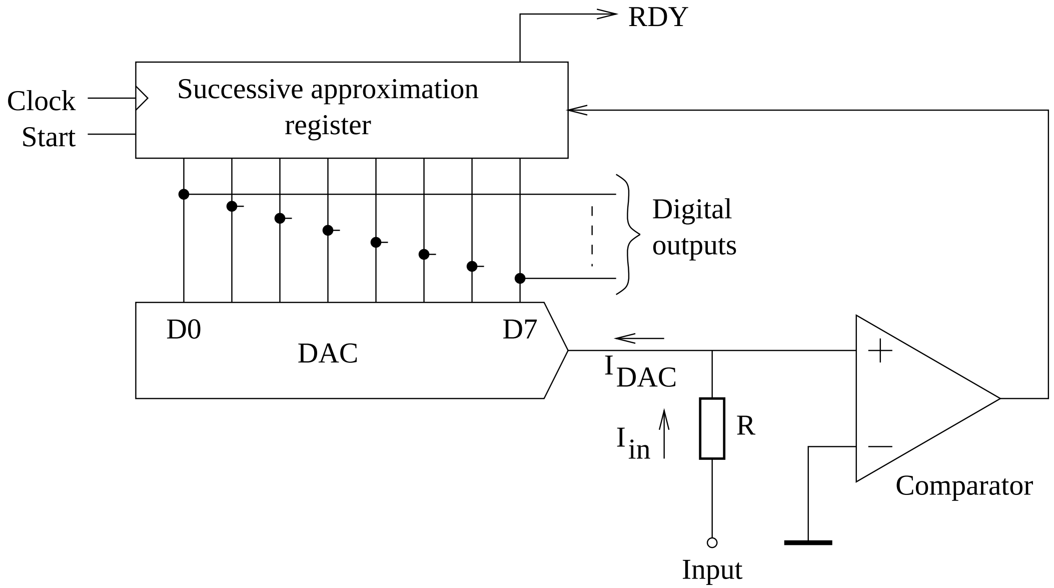
- A resistor divider chain creates all possible decision levels from a single reference.
- $2^n$-1 comparators compare each level with the input signal.
- Digital logic converts "thermometer" code into binary.
- Sensitive to 'sparkle' codes
- Metastability
- Input capacitance
- Poor linearity

Usually 8 bits, rarely more than 10



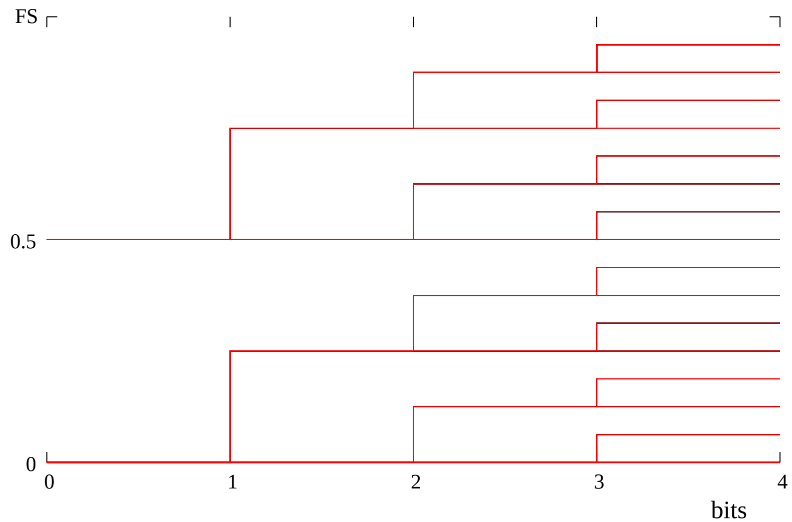
## Sequence of operation:

- Compare input with half scale, keep if greater.
- Add one quarter and compare, keep if input greater.
- Add 1/8 etc...




Basically a binary search algorithm

Usually clocked or strobed
Serial data interface
Often fixed conversion rate
Sometimes poor DNL

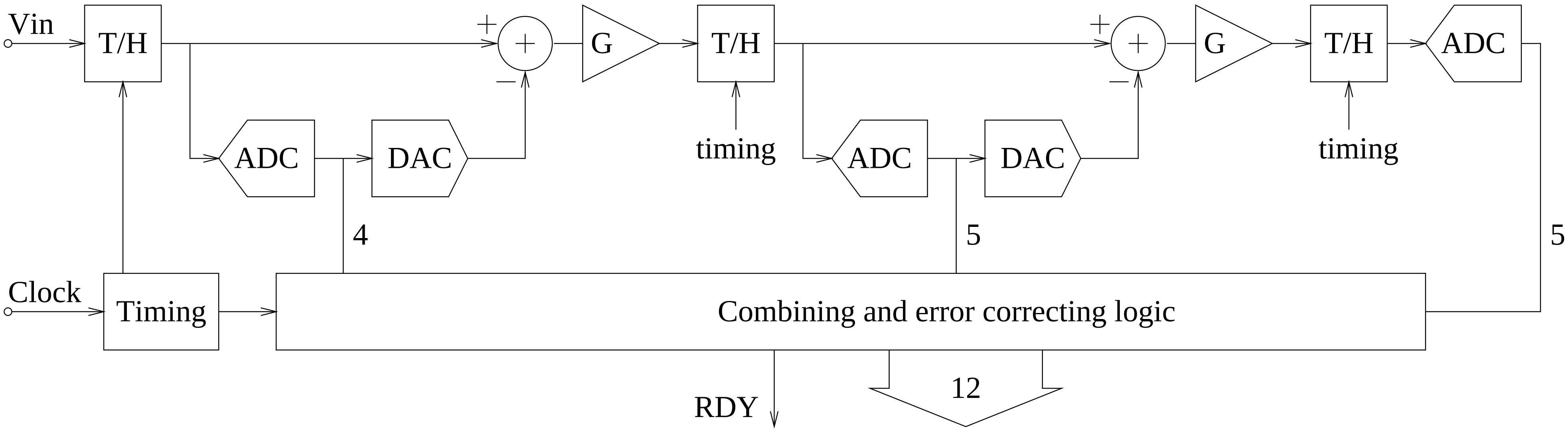



SAR ADC binary decision tree picture

# Mixed architecture ADCs

Segmented or pipelined ADC:

- Sample rate comparable to flash ADCs, but with several clock periods of latency.
- Resolution comparable to successive approximation architecture.



# The effect of oversampling

Increase sample rate:

➔ Quantization noise is spread over a larger BW.

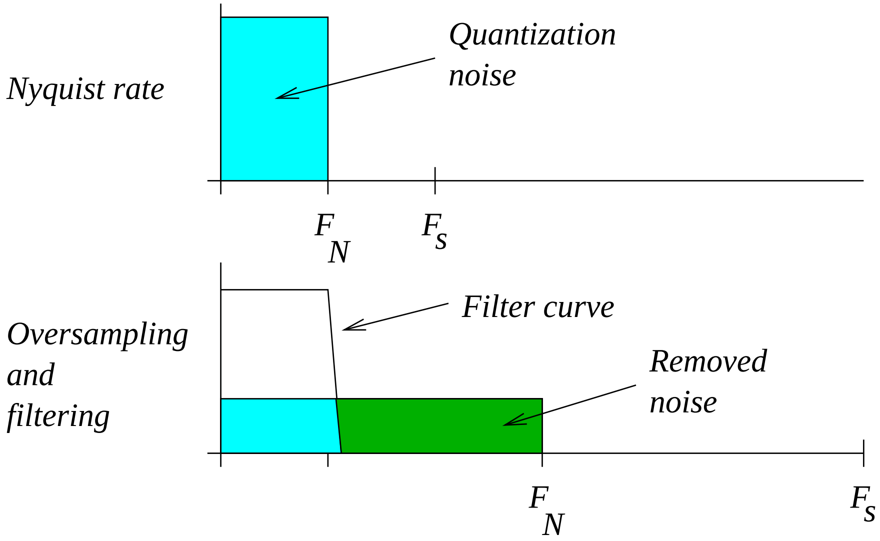
Numerically low-pass filter the sample stream:

➔ Out-of-band quantization noise power is removed.

Conclusion:

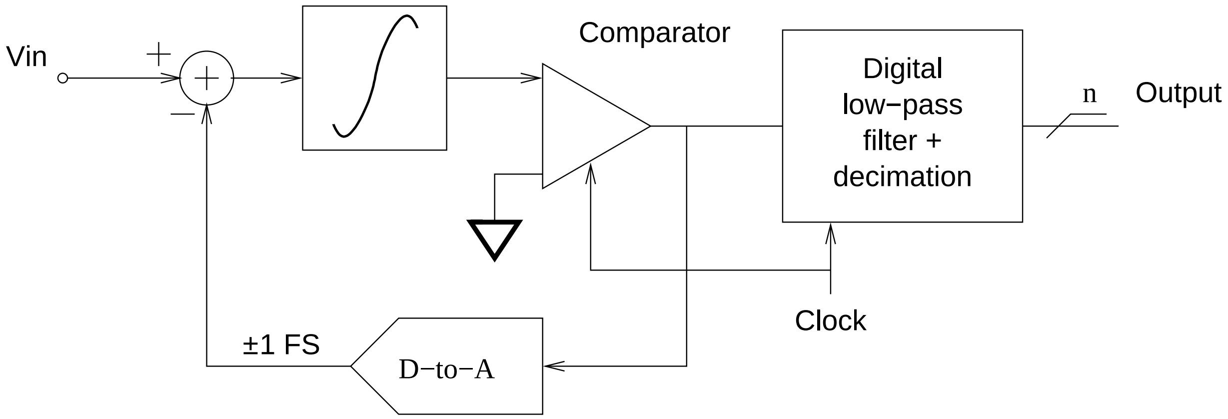
➔ SNR gets better by 3dB/octave of oversampling rate.

Dither may be necessary for very quiet ADCs.




It's possible to do much better using Σ–Δ modulation.

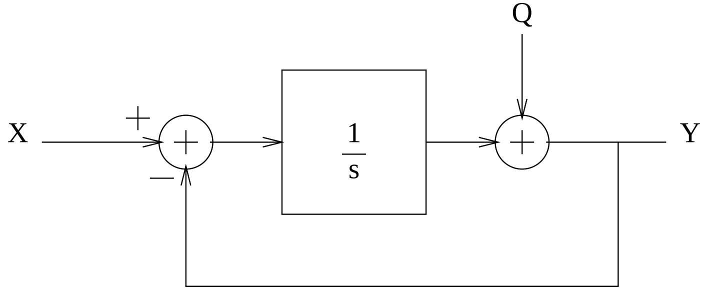
# The Σ/Δ ADC




Average duty cycle of comparator output reflects input value.
Digital low-pass decimation filter trades sample rate for resolution.
Good DNL, good resolution, slow.

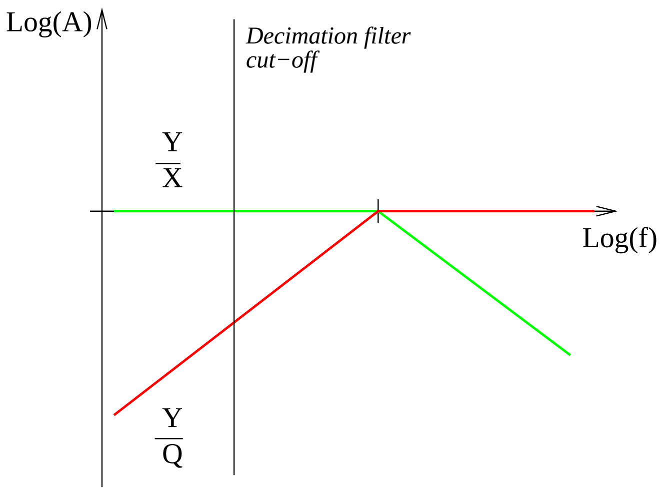
Model the Σ/Δ modulator as a linear feedback system with noise:

For the input: $$\frac{Y}{X}=\frac{1}{s+1}$$




Pretend that quantizer contributes random uncorrelated noise:

For the noise: $$\frac{Y}{Q}=\frac{s}{s+1}$$

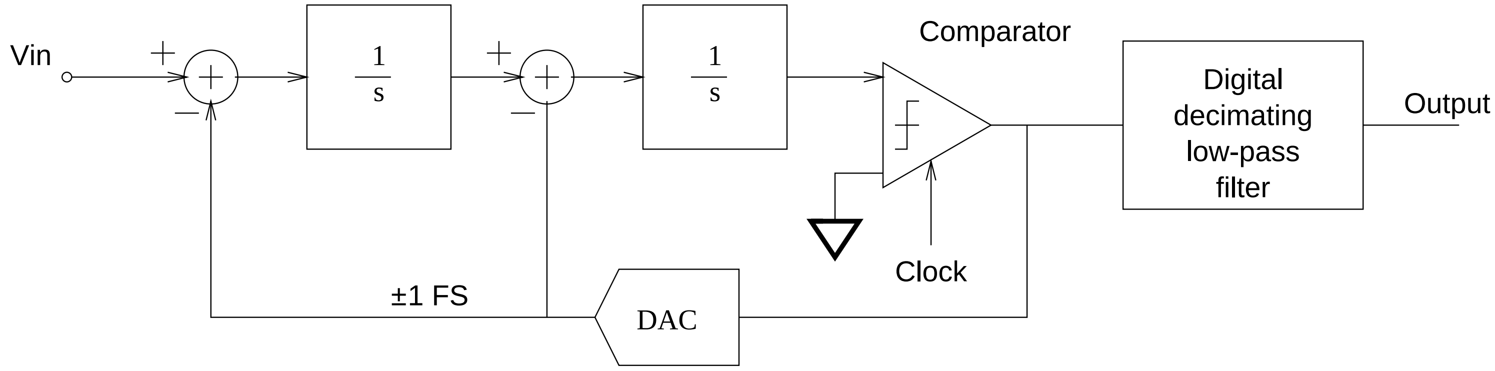



Input signal is low-pass filtered
Quantization noise is high-pass filtered
Decimation filter rejects high frequency
➔ Resolution is improved

Noise shaping!

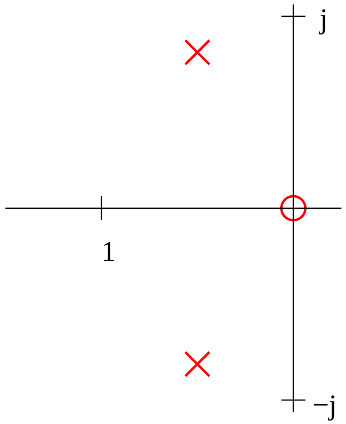
# The Σ/Δ ADC

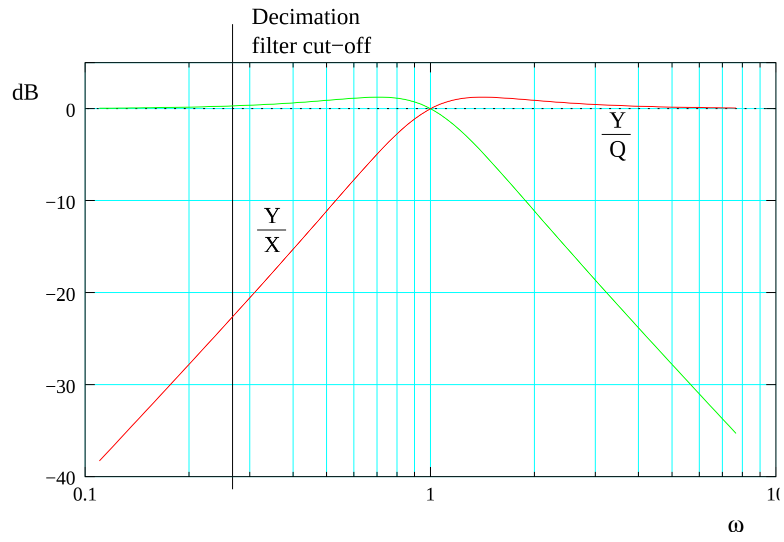
Higher order loops and noise shaping, E.g., a 2nd order modulator:




For the input:
$$\frac{Y}{X}=\frac{1}{s^2+s+1}$$

For the quantization noise:
$$\frac{Y}{Q}=\frac{s^2}{s^2+s+1}$$



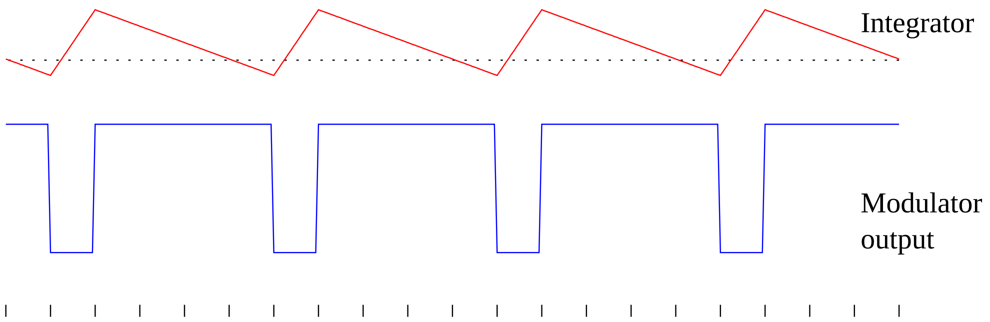




Decimating filter rejects noise

# Σ/Δ ADC ailments

## Side tones, idling patterns, birdies

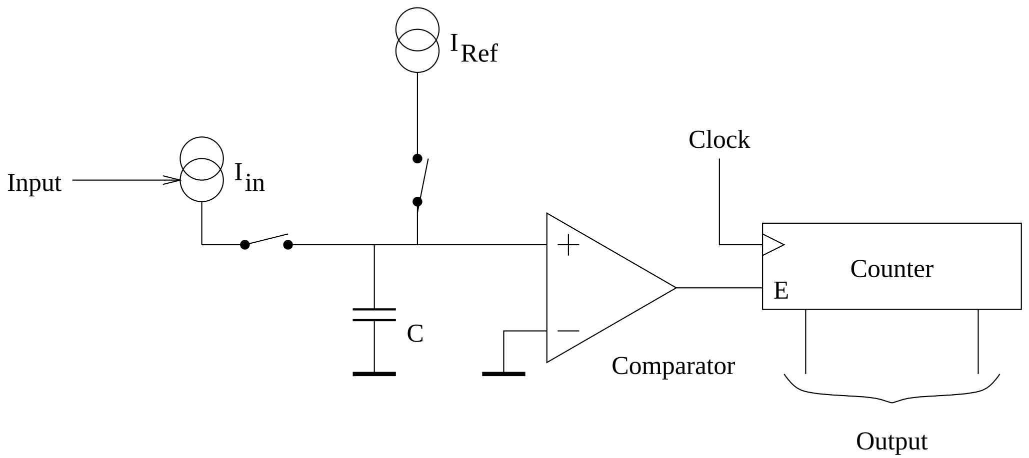
- For some input values, the Σ/Δ modulator can produce repeating patterns with repetition rates well below the sampling frequency
- These may leak though the decimation filter, causing a side tone or 'birdie'



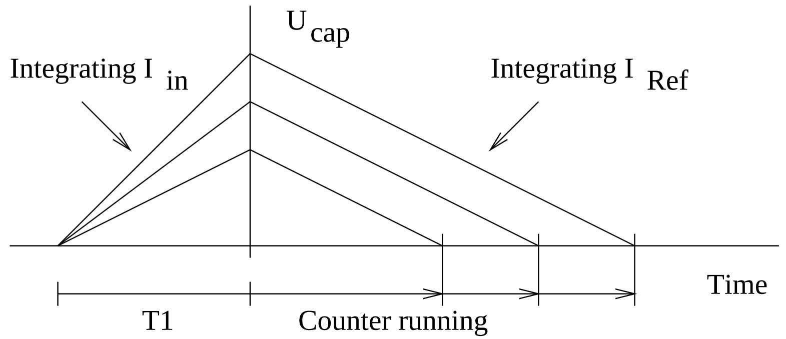

- Possible remedies include using higher order modulators and dither, to randomize things
- High-order modulators behave badly when overdriven

# The dual slope integration ADC (Wilkinson ADC)

Integrate input during T1
Then:
Integrate reference until zero is reached, while counting clock pulses
Final count is proportional to input



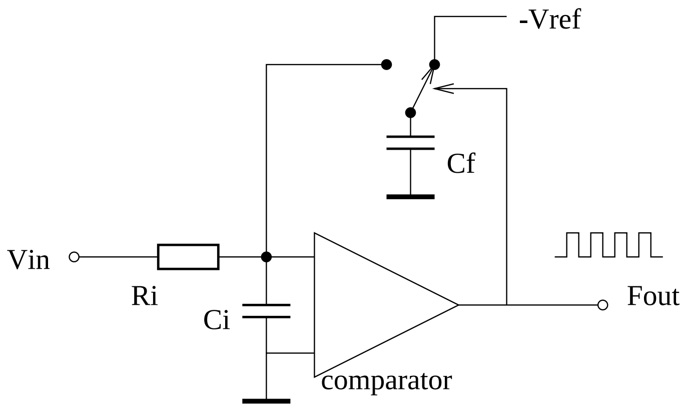

Can be built for low power operation, good for battery powered equipment (Multimeters etc.)
Integration time often chosen to reject power line interference
Excellent DNL




Sometimes the signal source is inherently a pulsed current source (Photomultiplier). The digitized quantity is then electrical charge.
Variants: Time to Digital Converter (TDC)

# Voltage-to-Frequency converters

Input voltage is integrated onto Ci
A fixed-size packet of charge is removed each time the switch connects Cf to the input.

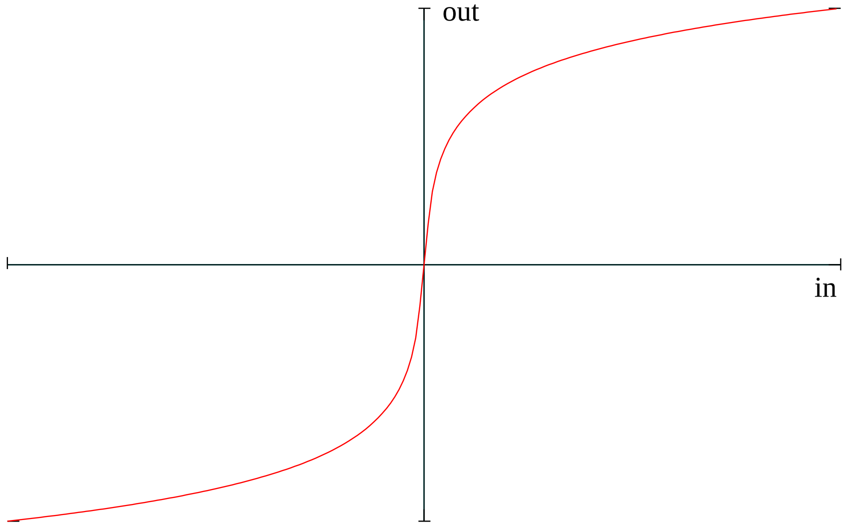



Applications:
Process control
Easy to use as integrating converter (Just add a counter)
In combination with F-to-V converter: Cheap isolation amplifier

Low-cost
Slow!
Signal easy to transmit over large distances
Very good linearity

# Special converters

Non-linear converters, A-law, μ-law, used in telephony.




A-law compression curve

Digital potentiometers (DS1669, MAX5438, AD5259, etc.
Digital output sensors, temperature, acceleration (AD7414, AD16006)
Capacitance-to-digital converters (AD7745)

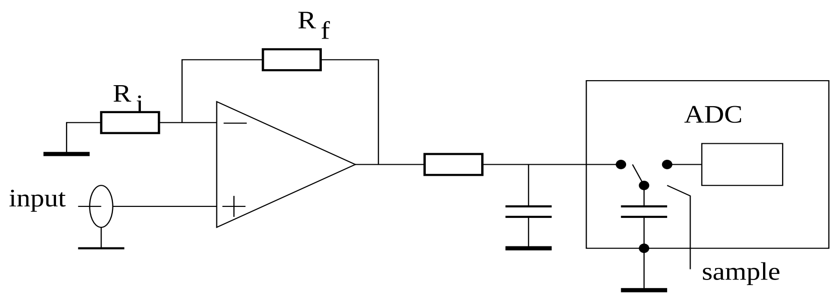
# Input signal conditioning

Purpose:

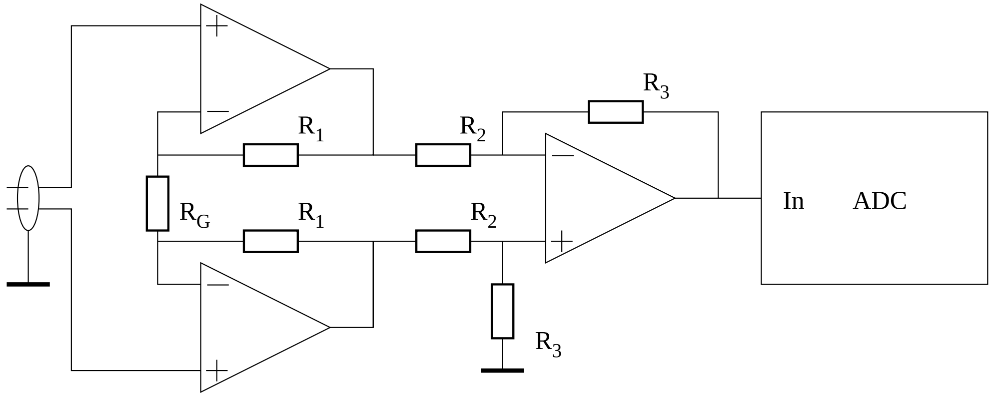
- Adapting signal range to ADC input range (Scaling & level shifting)
- Conversion between single-ended and differential signals
- Protecting the ADC input from overload
- Terminate long cables into their characteristic impedance...
- ...or, to the contrary, provide a high impedance to avoid loading the source
- Rejecting interference
- Filtering out-of-band frequencies (Anti-alias filter, noise reduction)
- Holding input constant while conversion takes place

# Active signal conditioning

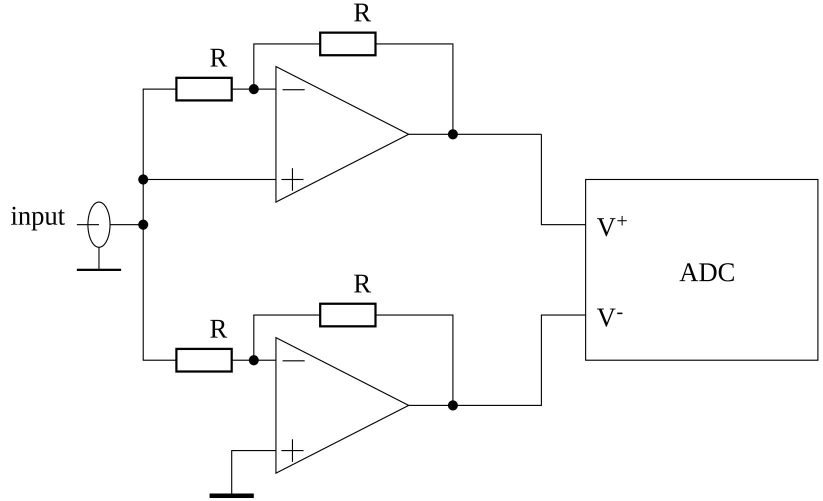
Buffer amplifier adapts signal to ADC and prevents sampler kickback
RC circuit at ADC input isolates amplifier from ADC input capacitance




Instrumentation amplifier: Good common mode rejection, high gain, but only at low frequency

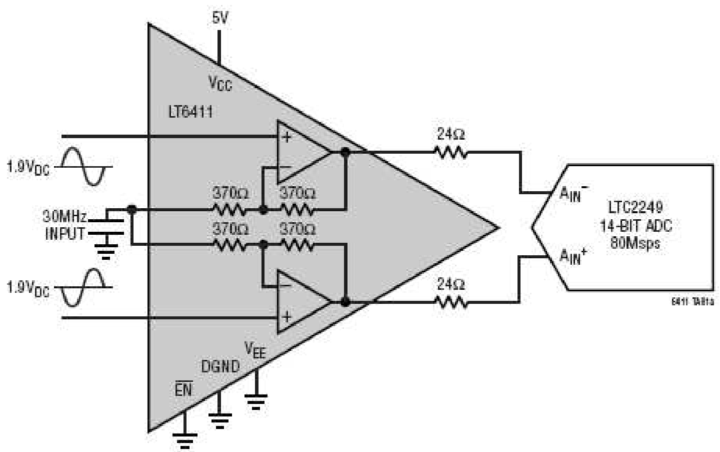


Single-ended to differential conversion
(Often used for high-performance or
low voltage ADCs)




Baluns and transformers can also be used for this.

Or use monolithic differential buffer amplifiers (ADA4941, AD8351, LT6411, THS4503, etc.)



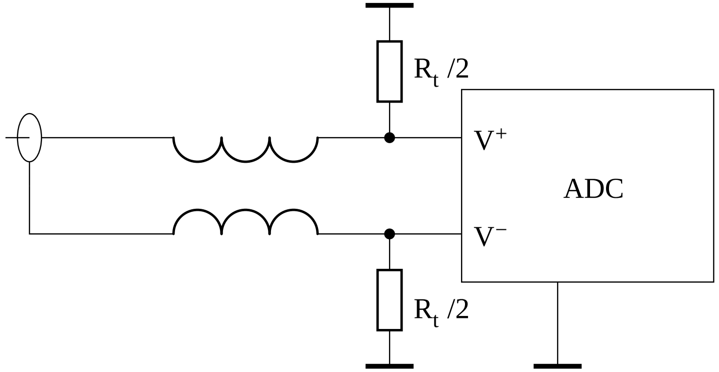
# Passive signal conditioning

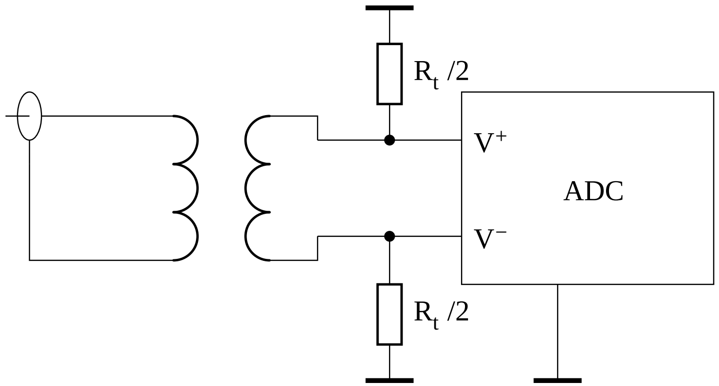
Baluns and transformers:

- Good common mode rejection at high frequencies.
- Linear, noise free, no power needed.

But:

- No DC response.
- Low impedance circuits only.
- Big.



# Conclusions

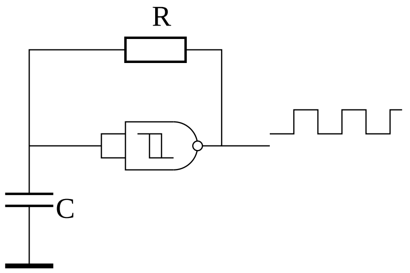
- There are ADCs and DACs for almost any imaginable application.
- Performance is ever getting better.
- 
- Prices keep going down. (15 years ago, a 12 bit 10 MS/s ADC cost 1 k$. Now it's around 10 $)
- Going digital does not end your analogue worries.

Digital is here to stay!

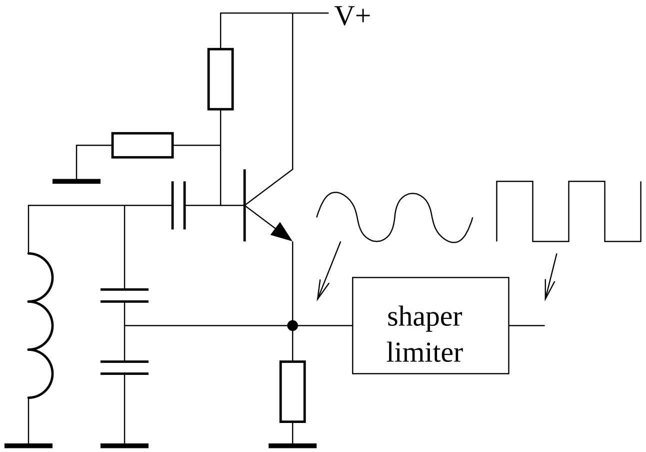
# END

# Typical jitter specs of common clock sources

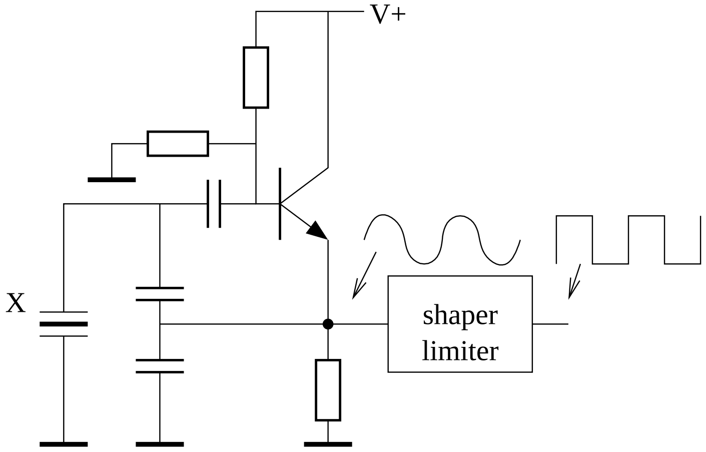
RC and logic gate oscillators — Poor
Jitter >100ps




LC oscillators — Fair
Jitter 10 - 100ps

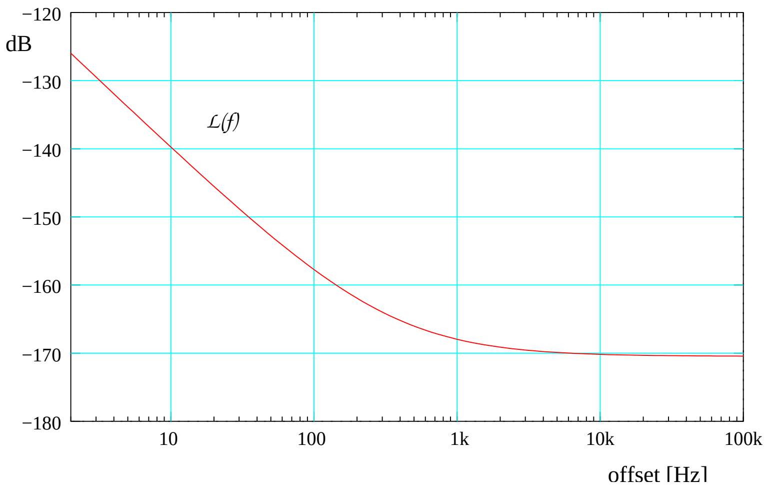



Pierce oscillator — Good.
Usually better than 1ps



# Clock jitter & phase noise

Upconverted thermal, schottky and 1/f noise



SSB phase noise of an oscillator

$$\Delta t_{rms} = \frac{T_0}{2\pi} \sqrt{\int_0^{\infty} S_{\varphi}(f) \cdot 4 \sin^2(\pi f \tau)\, df}$$

$S_{\varphi}(f)$ is the spectral density of the phase noise

(Function of Fourier frequency *f* and sum of both sidebands) $S_{\varphi}(f) = 2 * L_{\varphi}(f)$

$\sin^2(\pi f \tau)$ is a weighting function

(For low frequencies in *S*, the phase can't drift very far, when τ = *n/f*, the contribution cancels, and there are maxima in between.)

τ is the time between two events (Usually $\tau = T_0$)

The term $\frac{T_0}{2\pi}$ converts phase into time

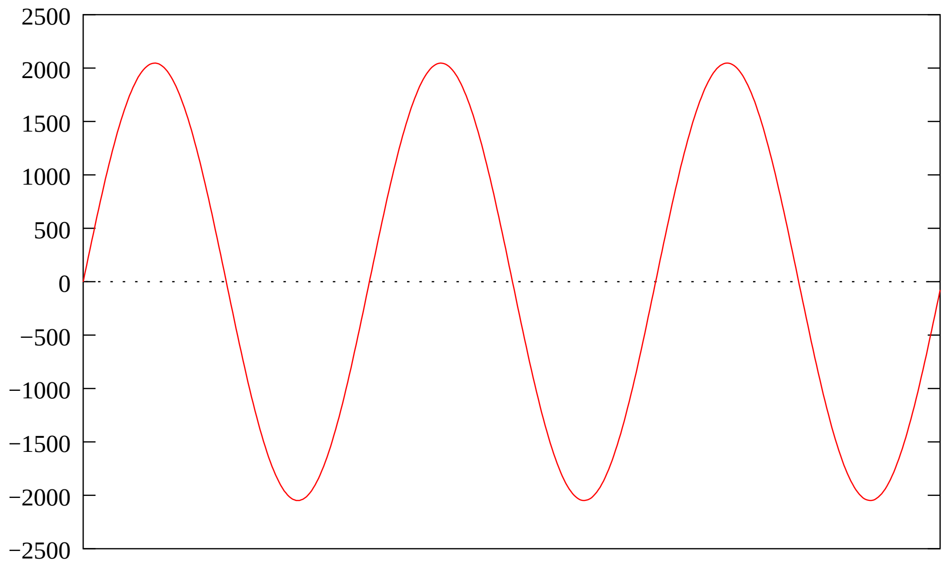
# Data formats

| | | |
|---|---|---|
| Straight binary | 1111 1111 | — max |
| | ... | |
| | 0000 0000 | — 0 |
| Offset binary | 1111 1111 | — max/2 |
| | ... | |
| | | — 0 |
| | ... | |
| | 0000 0000 | — -max/2 |
| Two's complement | 0111 1111 | — max/2 -1 |
| | ... | |
| | 0000 0000 | 0 |
| | 11111 111 | -1 |
| | .... | |
| | 1000 0000 | — -max/2 |

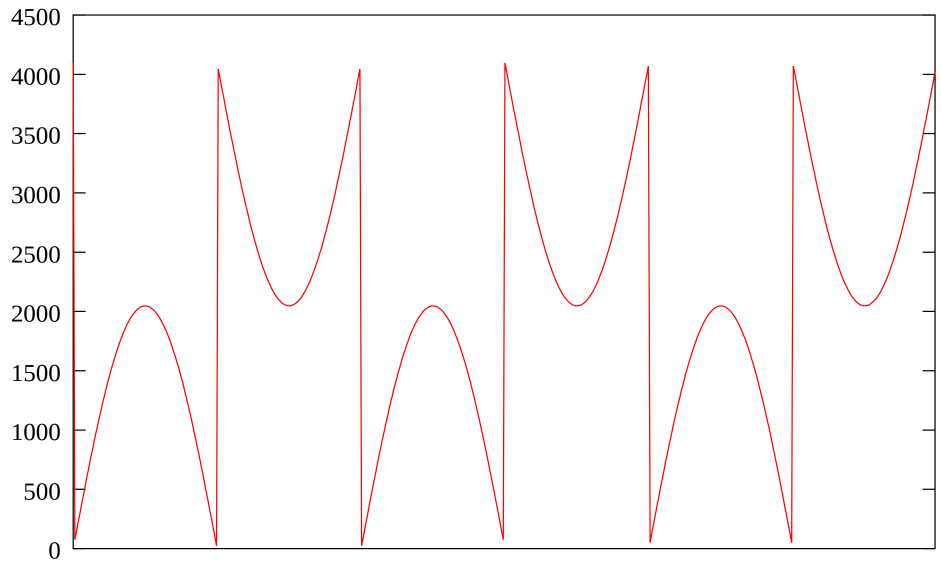
Note: To convert offset binary into 2's complement, simply invert the most significant ADC bit.

# Data formats

You were expecting this:

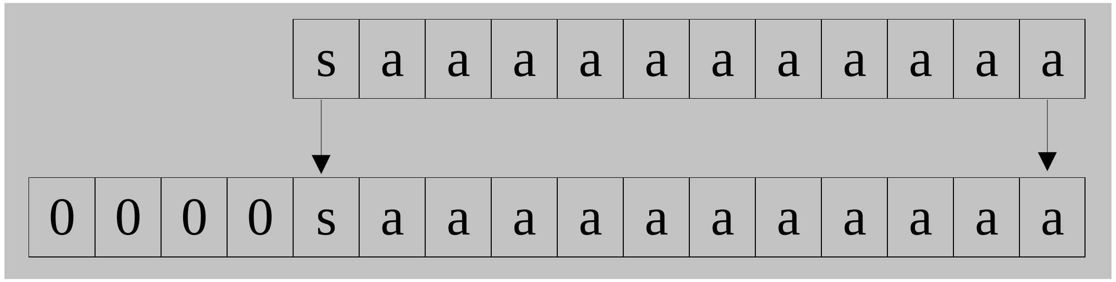


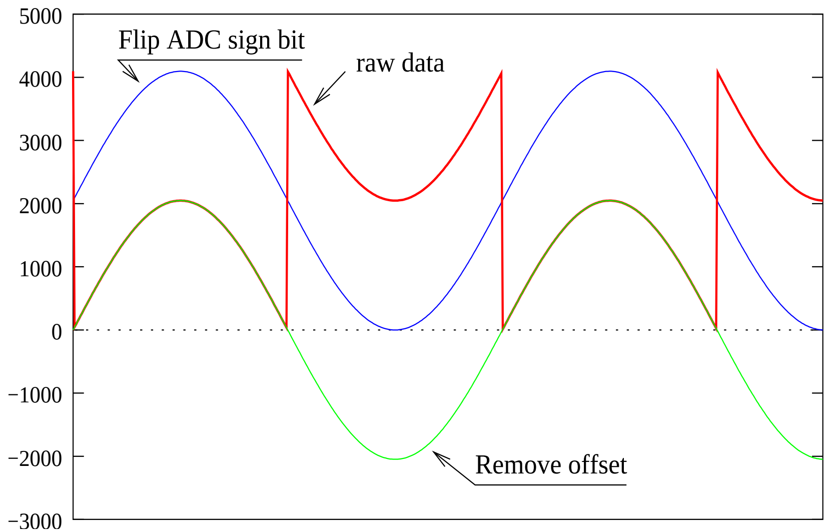

but you got this:



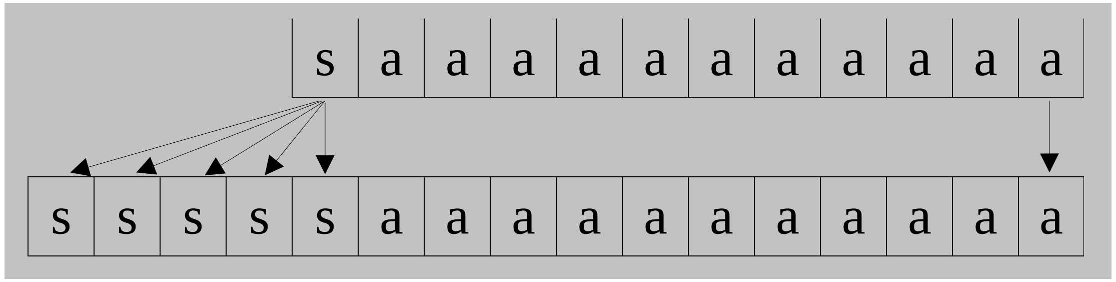

Reason: The ADC delivers 2's complement numbers and the sign bit is not in the right place.

# The sign bit isn't in the right place:

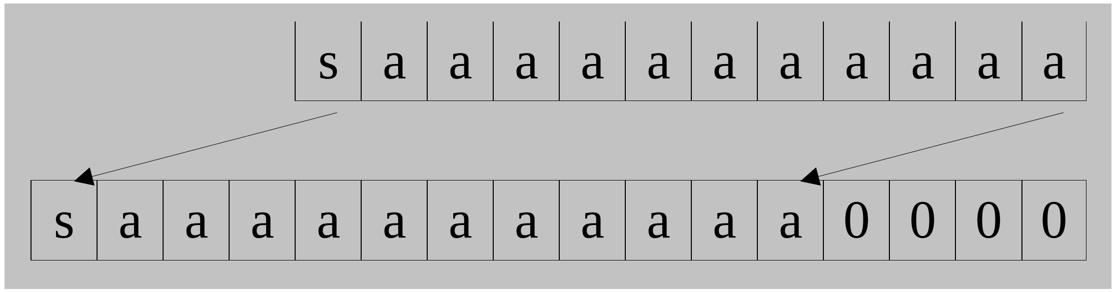



## Apply sign extension: `a = (a^0x800)-2048;`

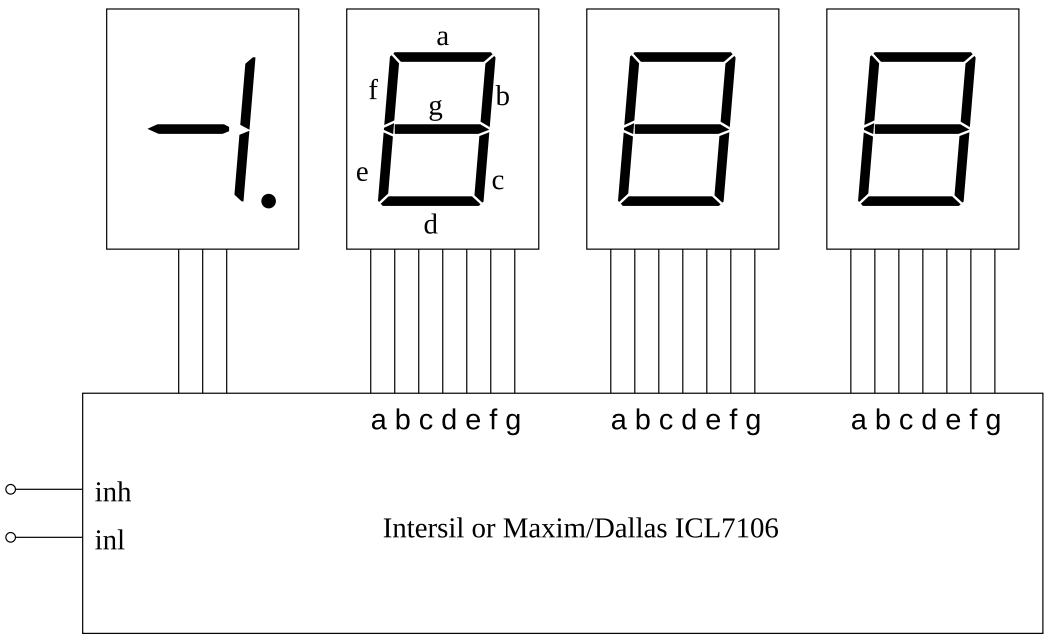


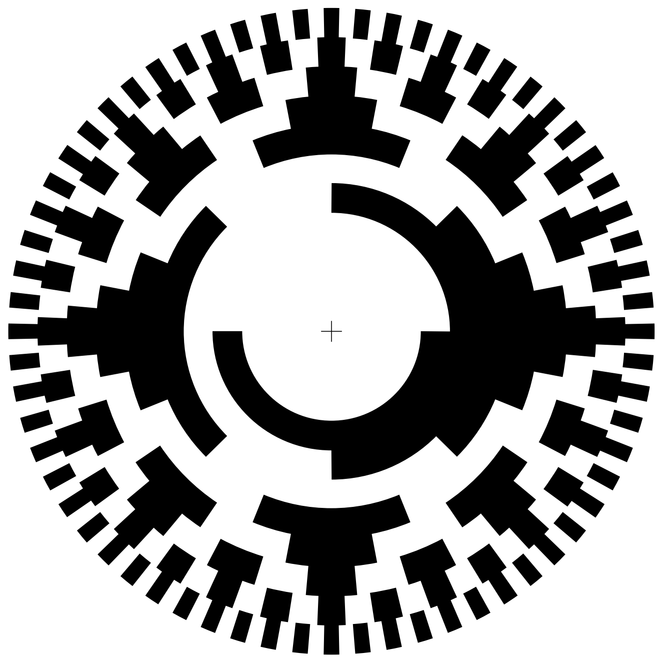

## or logical left shift: `a<<=4;`



BCD and display driver outputs. Handy for stand-alone instruments, panel meters, hand-held multi-meters.

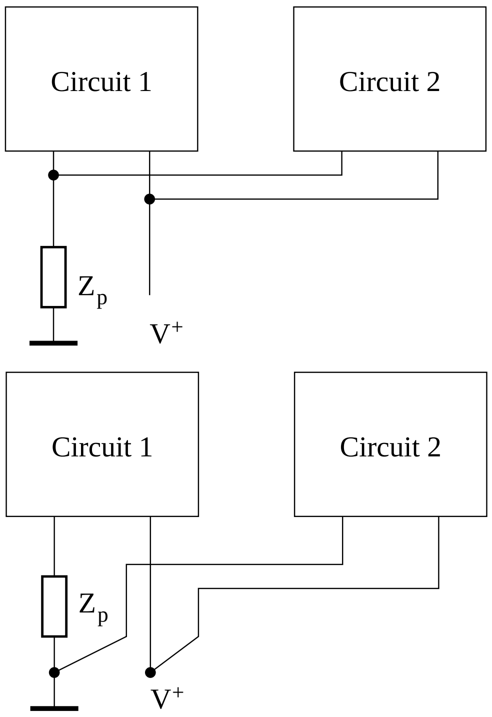
Gray code: Only one bit changes between adjacent values. No glitches. Resolver disks. Angular and linear transducers.



... or how to get your signal digitized cleanly:

Interference (comes from elsewhere) and noise (inherent in the circuit)




Coupling paths:

· Common impedance coupling

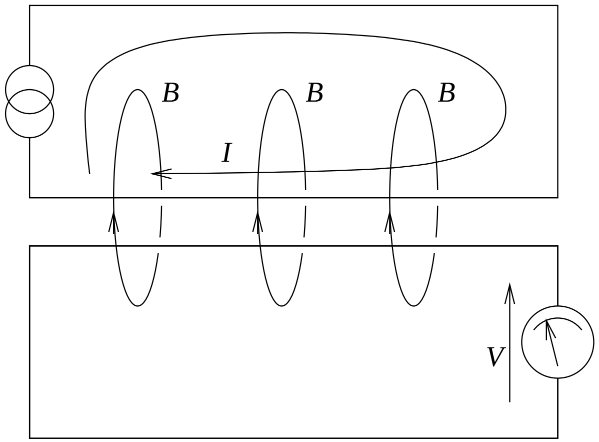
(Use generously dimensioned conductors or a star layout)

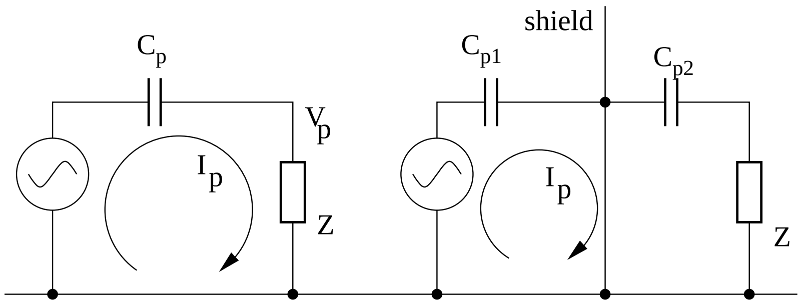
· Inductive coupling

(Keep loop areas small and put distance between them)

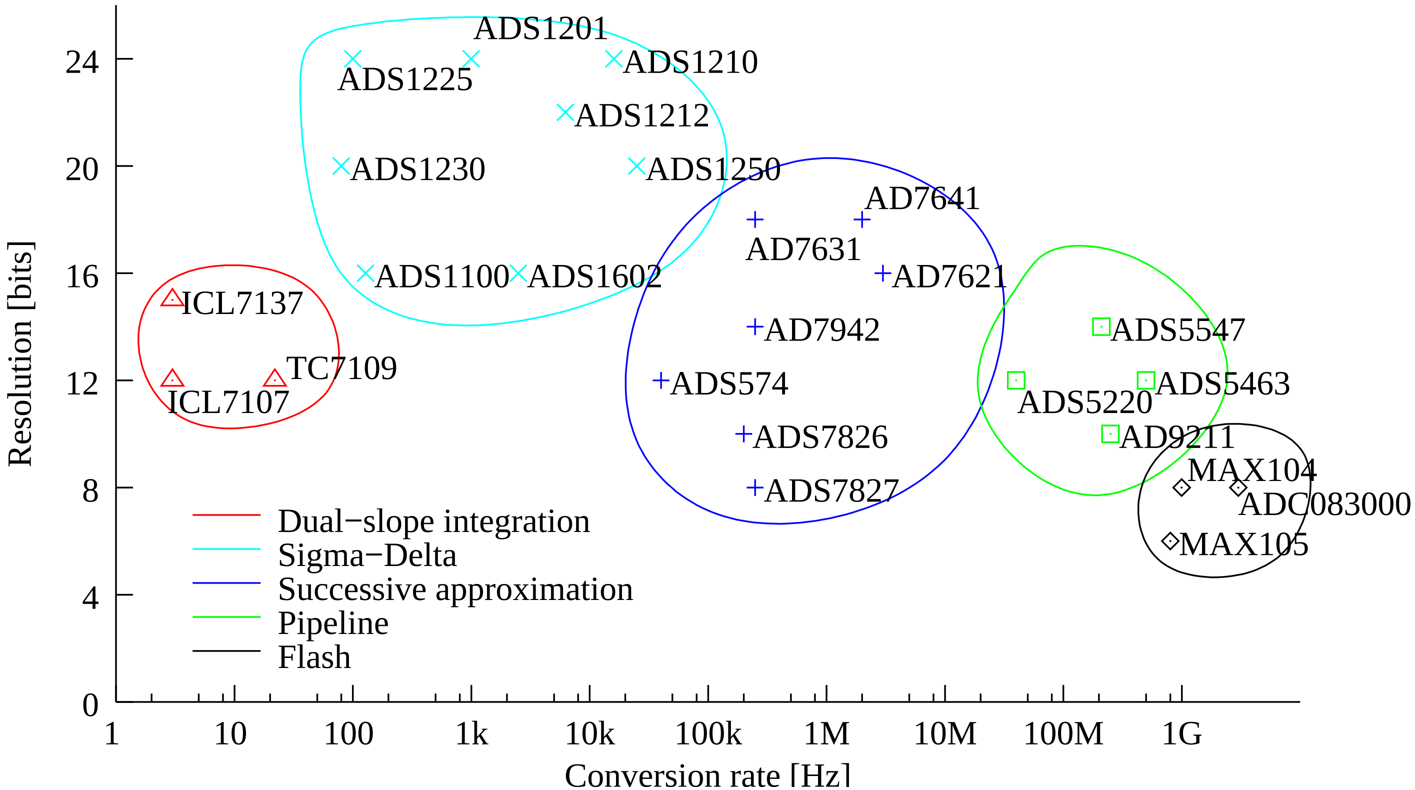
· Capacitive coupling

(Keep  high-Z nodes and nodes with high dE/dt far apart, or put grounded shields between them)



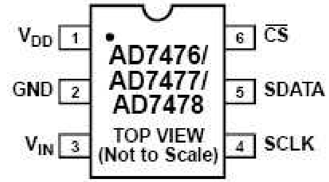



# The ADC landscape

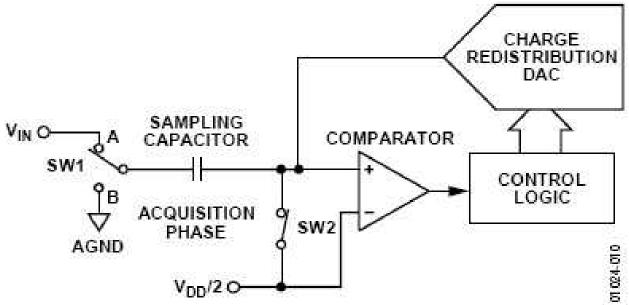


# Example of a SAR ADC: AD7474

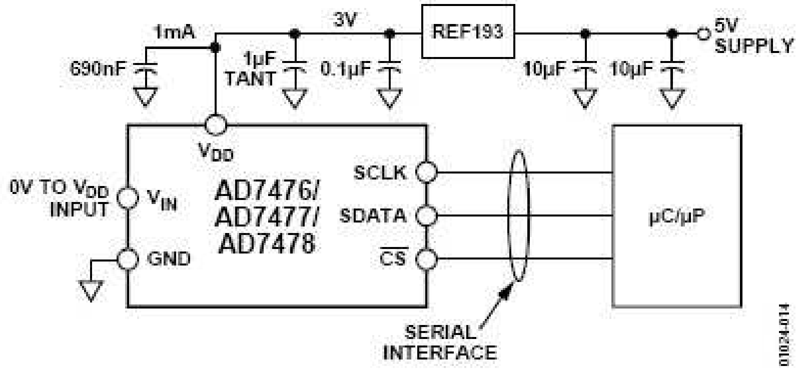
## 12-bit, 1 MS/s, serial interface SAR ADC

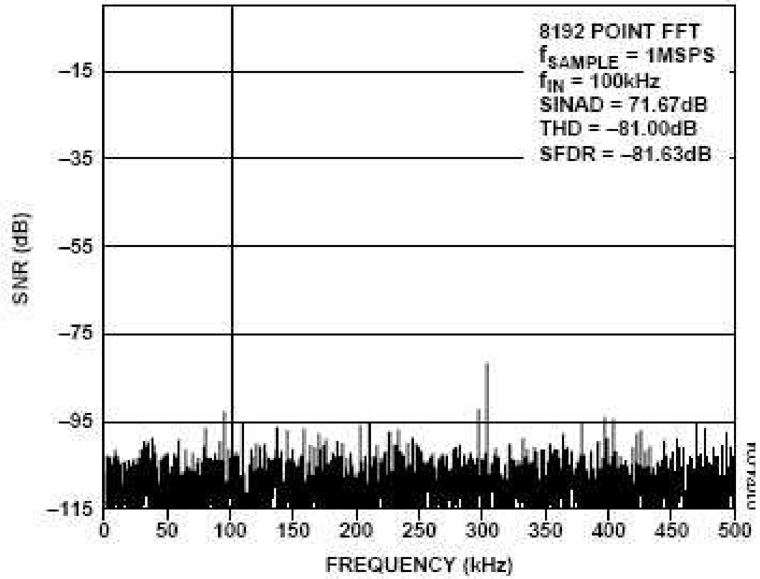


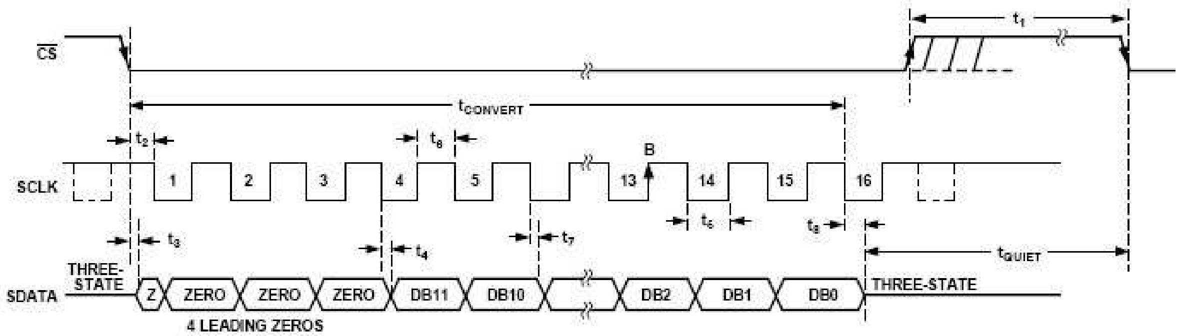

SOT23-6
(1.6 x 2.9mm)




Input sampler ($C_s$=30pF)



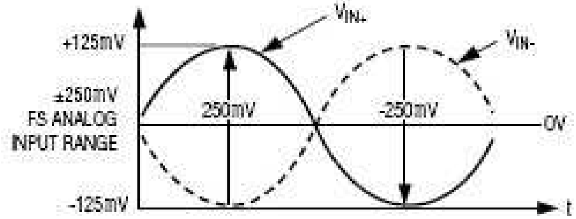



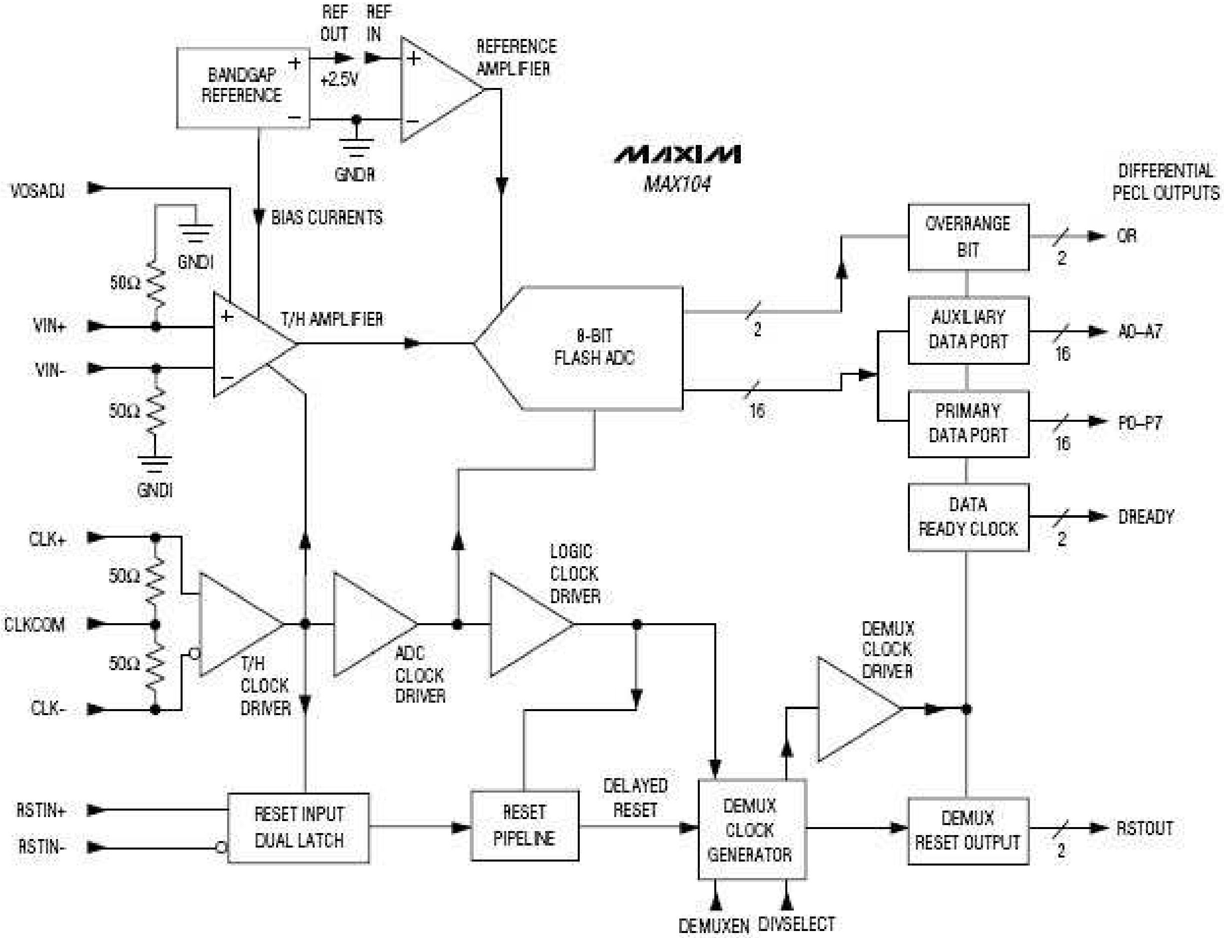

Serial interface timing

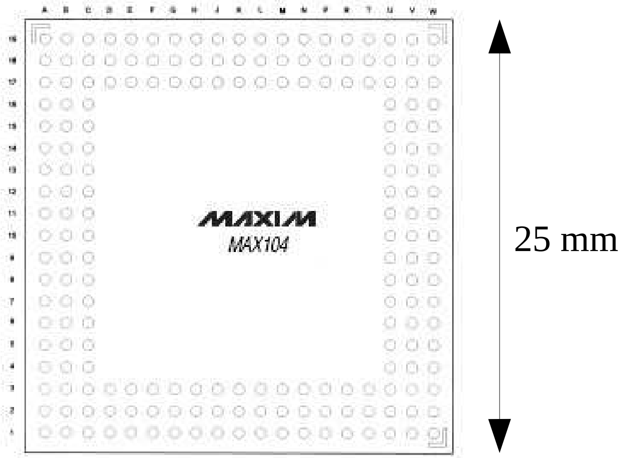
# Example of a flash ADC: MAX104

- 8-bit, 1 GS/s flash ADC
- 50 Ω differential inputs
- ± 250 mV input range
- Metastability error rate $10^{-16}$
- Differential PECL outputs
- De-multiplexer
- Needs 3 supply voltages
- +/-5 V and 3.3V
- Many GND and Supply pins




$P_d$ = 5.25 W
ENOB = 7.5 bits
$BW_{in}$ = 2.2 GHz

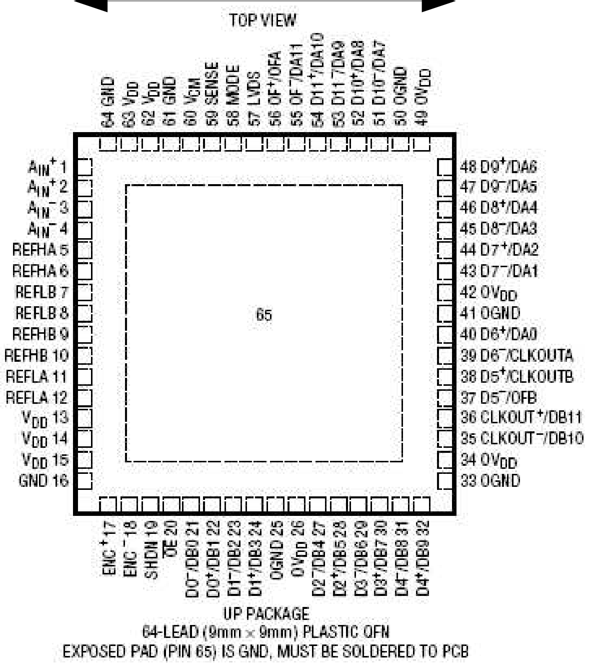


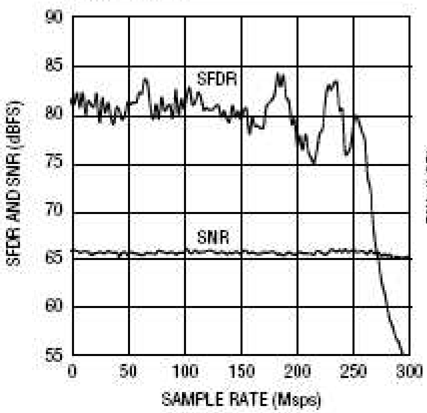

192-contact BGA

12-bit, 250 MS/s, 5-stage pipeline ADC
Differential LVDS <u>or</u>
de-multiplexed outputs
2pF sampling capacitor
Differential analogue
input, BW 1.2 GHz

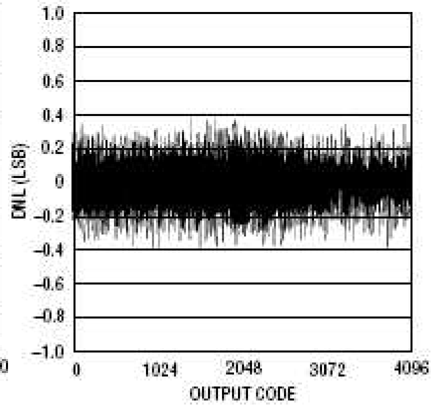
9 mm



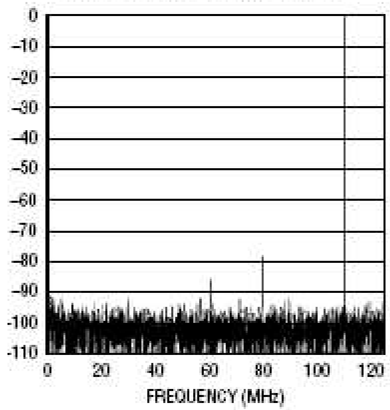

SFDR and SNR vs Sample Rate, 2V Range, $f_{IN}$ = 30MHz, –1dB, LVDS Mode



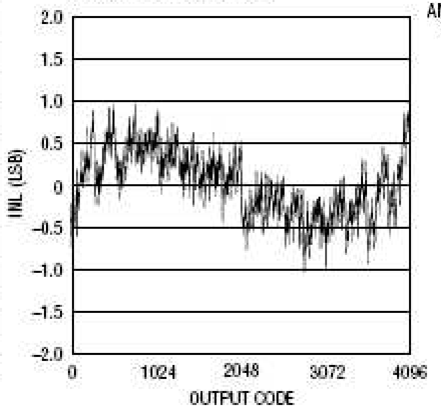

Differential Nonlinearity

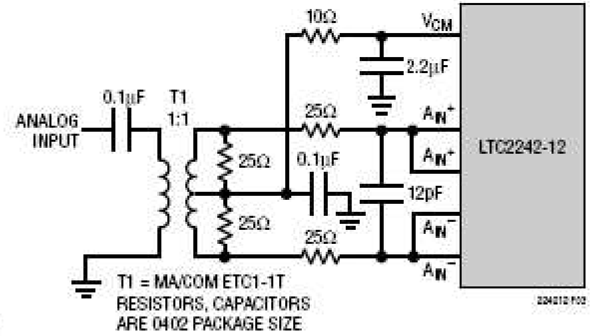


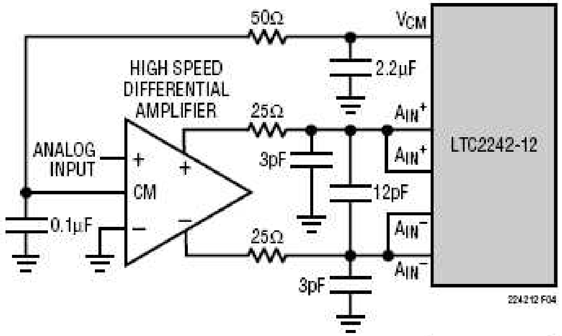

8192 Point FFT, $f_{IN}$ = 140MHz, –1dB, 2V Range, LVDS Mode

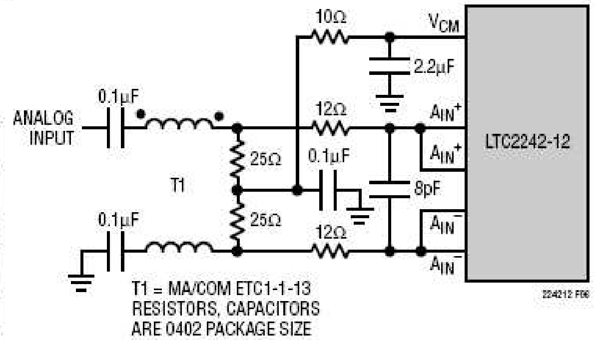



Integral Nonlinearity



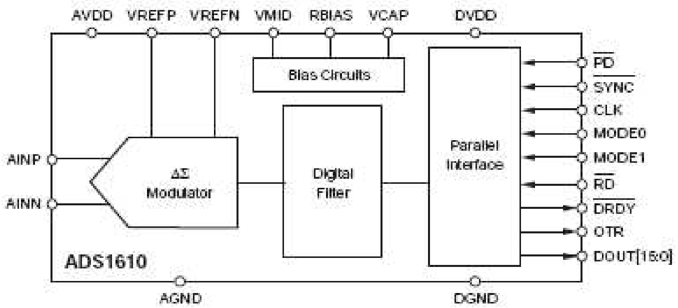



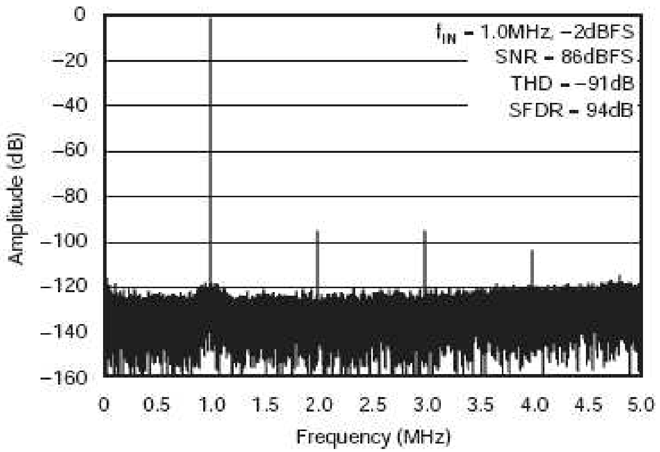

Single-ended to differential conversion

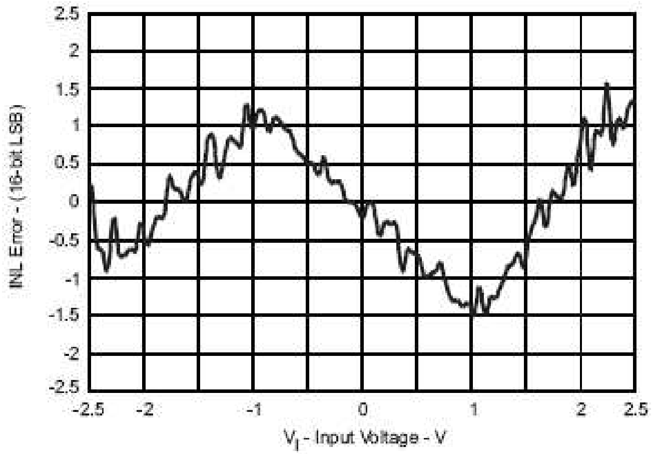
# Example of a Σ-Δ ADC: ADS1610

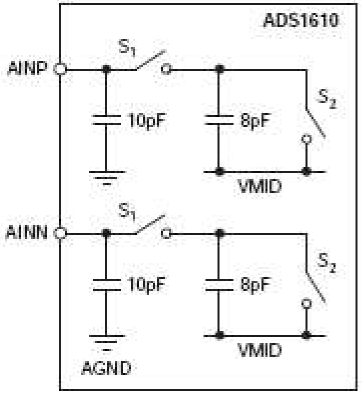
Programmable, instrumentation ADC
12-16 bits, depending on filter settings



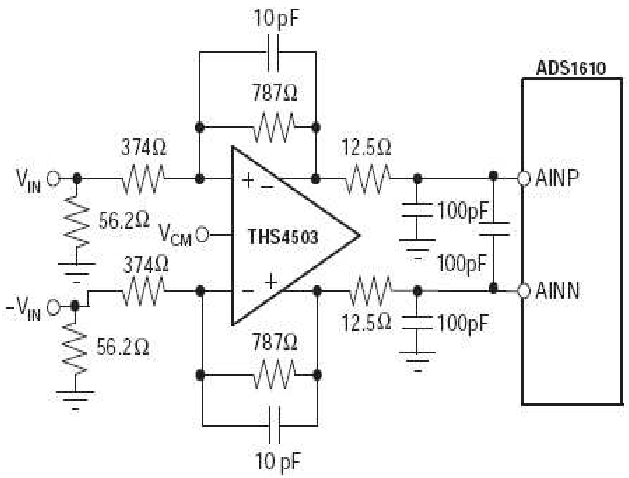

ADS1610 block diagram



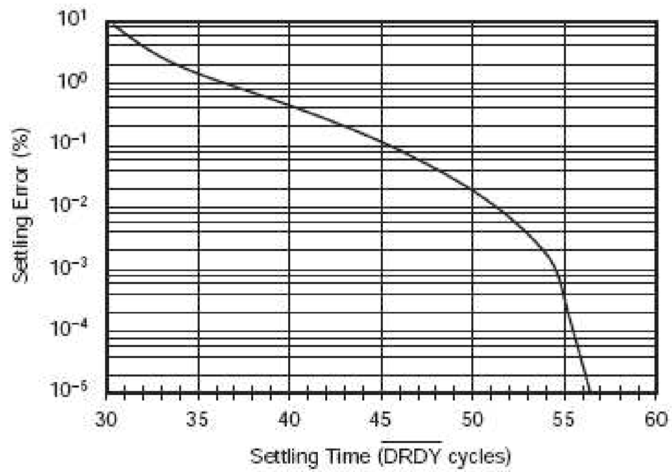




Circuit model of inputs

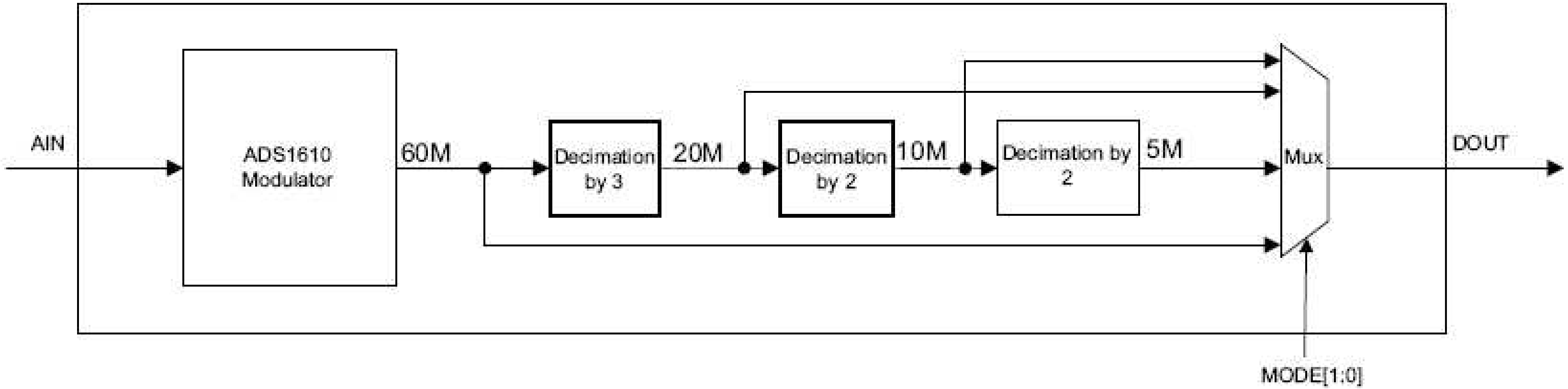



Input conditioning

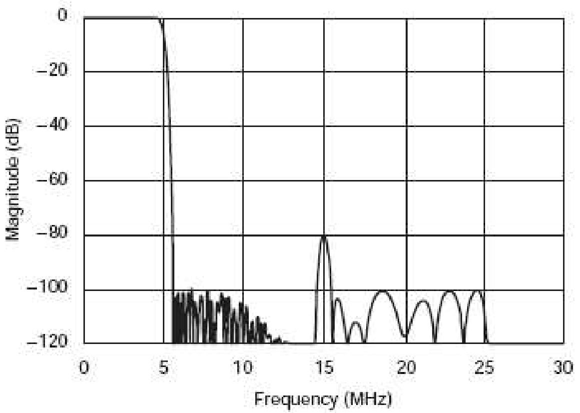


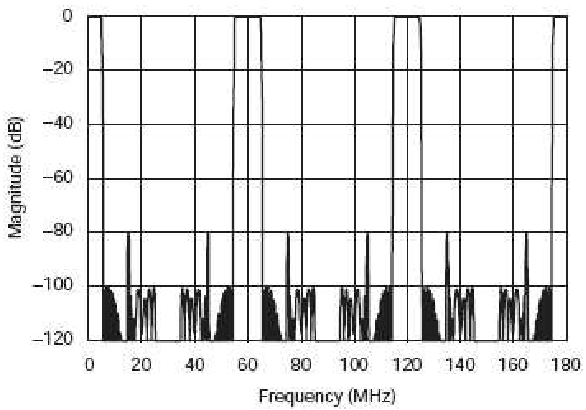
# Example of a Σ-Δ ADC: ADS1610

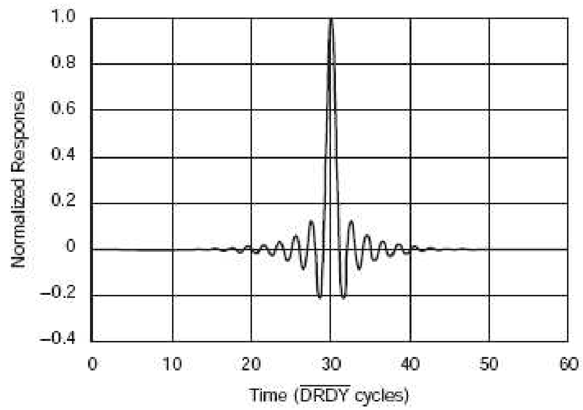
Programmable, instrumentation ADC
12-16 bits, depending on filter settings




| Mode 1 | Mode 0 | OUTPUT RATE | OSR | SNR (TYP) | BITS | SETTLING TIME (DRDY cycles) |
|---|---|---|---|---|---|---|
| 0 | 0 | Default 10MHz mode | 6 | 86dBFS | 16 | 55 |
| 0 | 1 | 20MHz | 3 | 74dBFS | 14 | 25 |
| 1 | 0 | 5MHz | 12 | 91dBFS | 16 | 55 |
| 1 | 1 | 60MHz bypass mode | 1 | 55dBFS | 12 | NA |

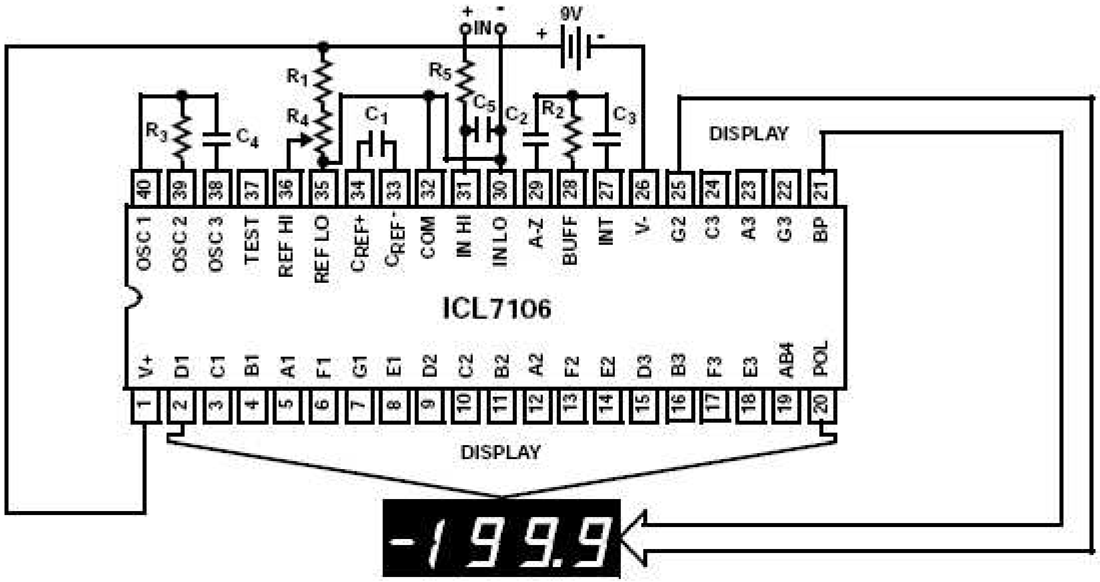


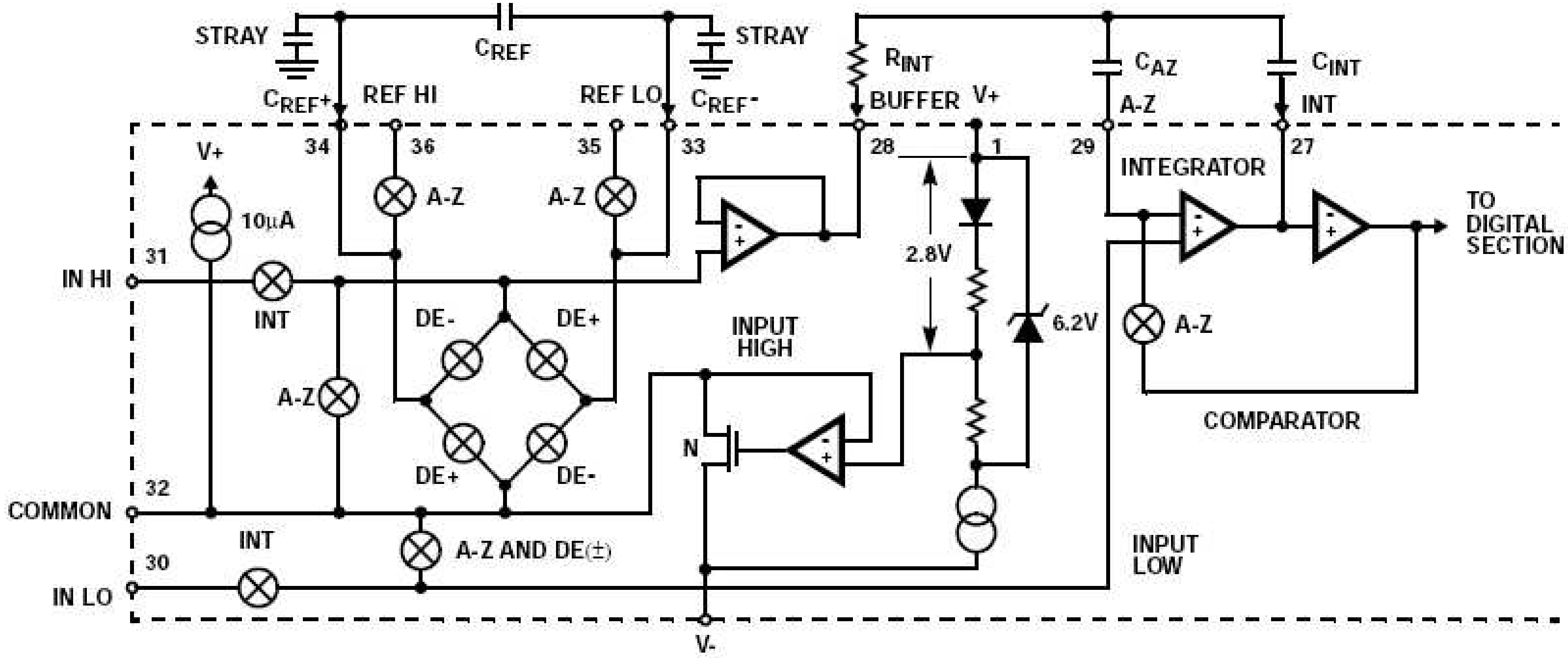
# Example of a dual slope integrating ADC: ICL7106

Directly drives an LCD display
±200 mV full scale
Conversion time 300 ms
Consumes 1 mA @ 9 V




$C_1$ = 0.1μF
$C_2$ = 0.47μF
$C_3$ = 0.22μF
$C_4$ = 100pF
$C_5$ = 0.02μF
$R_1$ = 24kΩ
$R_2$ = 47kΩ
$R_3$ = 100kΩ
$R_4$ = 1kΩ
$R_5$ = 1MΩ

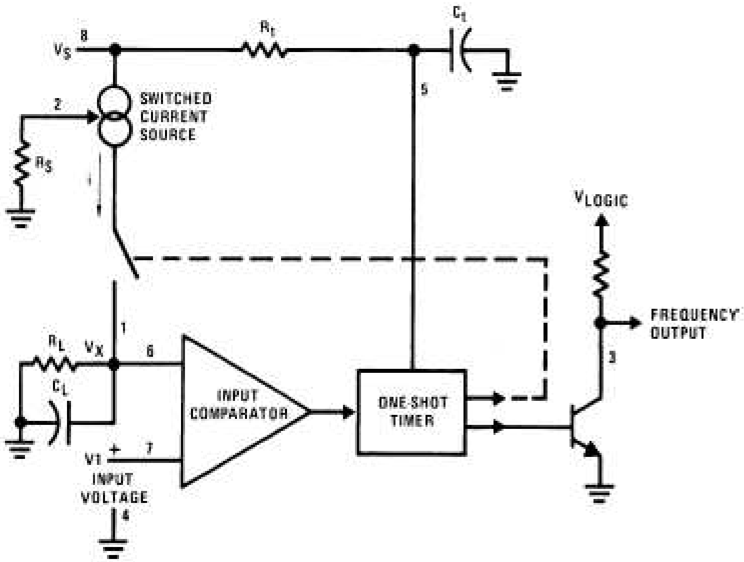
# Example of a dual slope integrating ADC: ICL7106

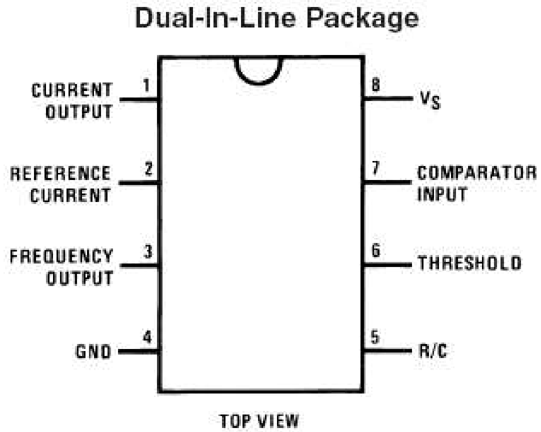


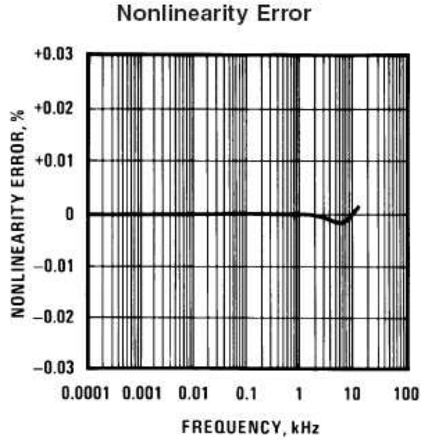

Analogue section of ICL7106

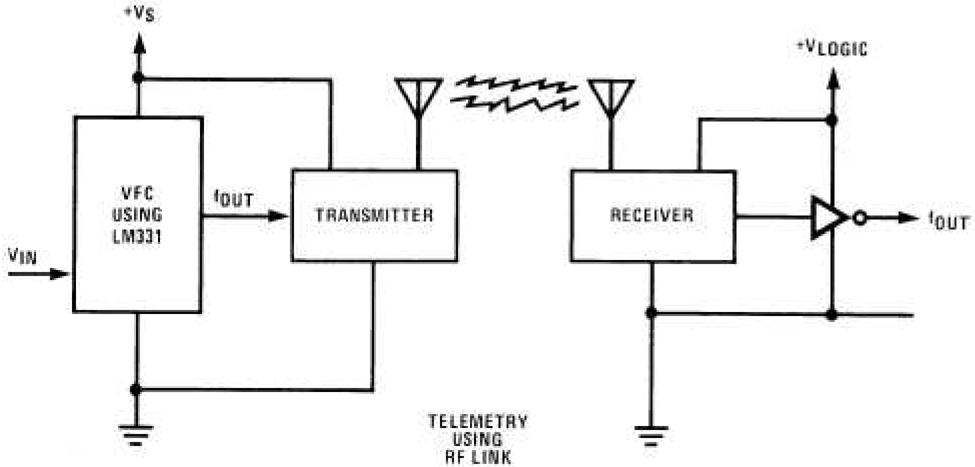
# Example of a V-to-F converter: LM331

8-pin DIP
Linearity 0.003%
Fmax 10 kHz



# DAC architectures

| *Architecture* | |
|---|---|
| Kelvin-Varley divider | Accurate, monotonous. Mainly as building block in integrated DACs |
| 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 |
| Σ-Δ | Linear, accurate, but complex. |

## Mixed architectures:

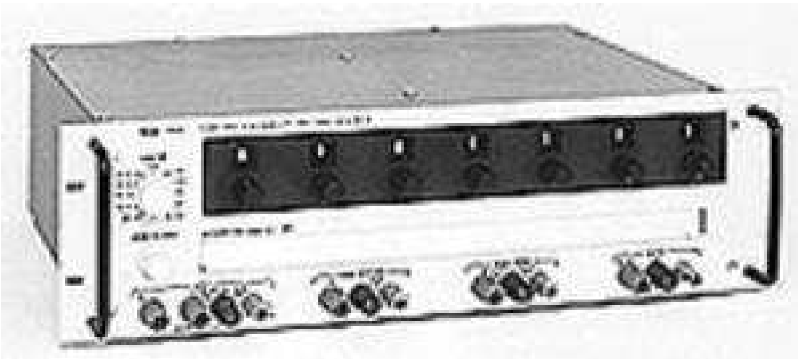
- Segmented DAC
- Interpolating DAC

## Variants:

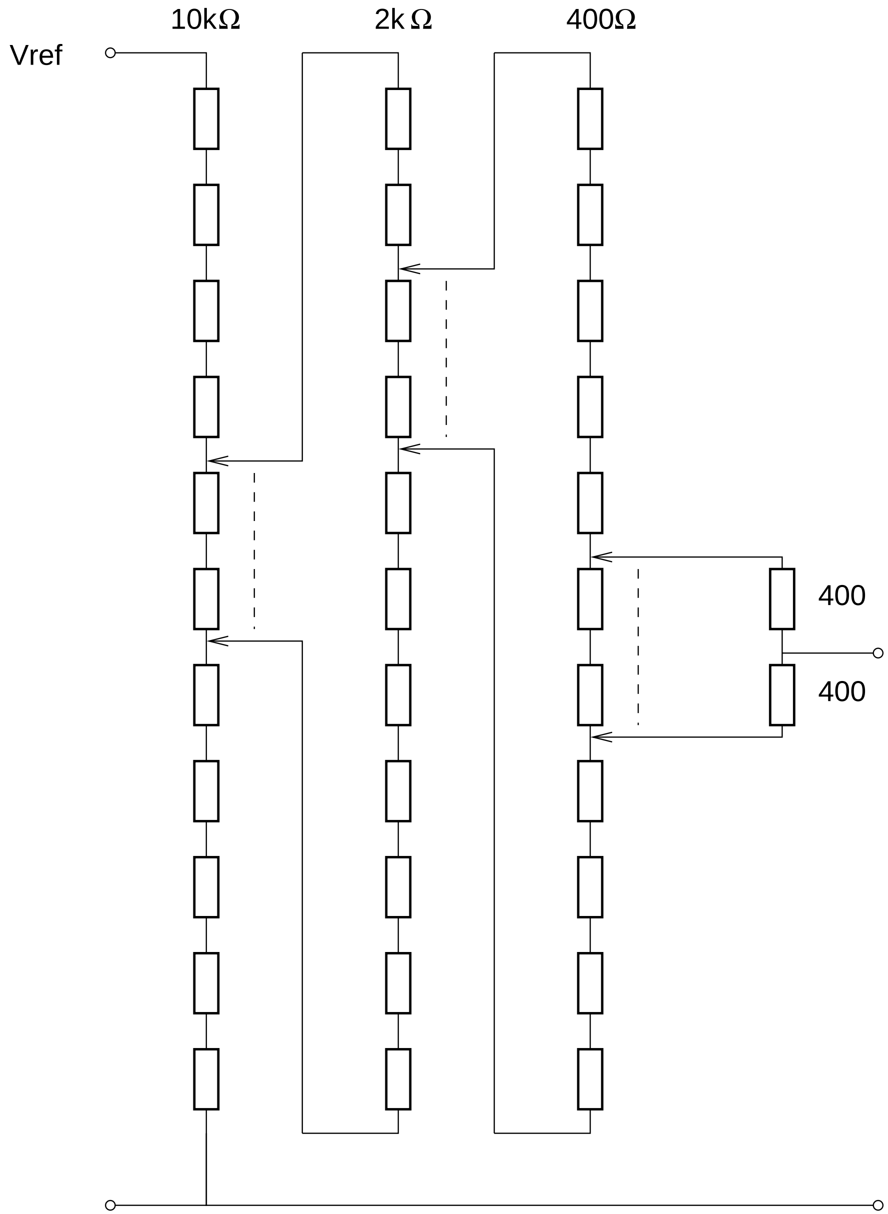
- Multiplying DACs
- Current or voltage output
- Differential or single-ended

# Kelvin-Varley divider

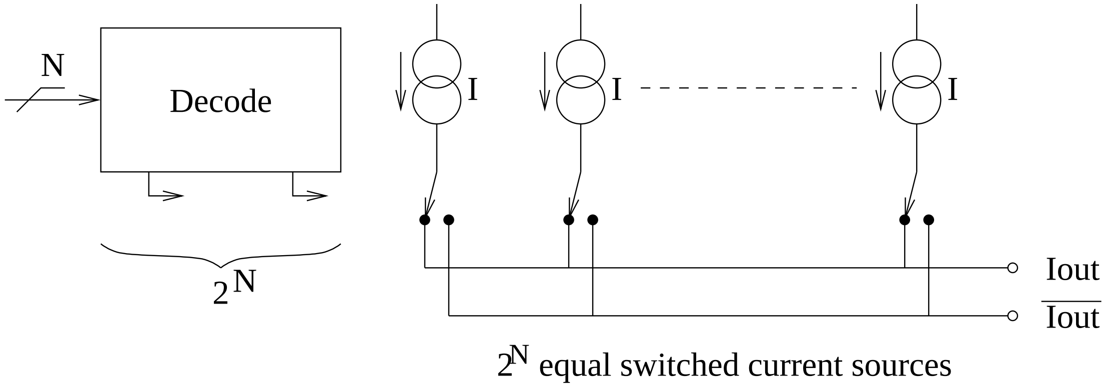
- The ancestor of all DACs
- Still rivals the best modern DACs
- Available as rack mounted units (Expensive!)
- Used as sub-circuit in IC DACs
- Variable $Z_{out}$: Buffer amplifier needed



Example: Fluke 720A
Linearity, resolution : $10^{-7}$
Input resistance : 100 kΩ, 0.005%



# 'Thermometer' DAC



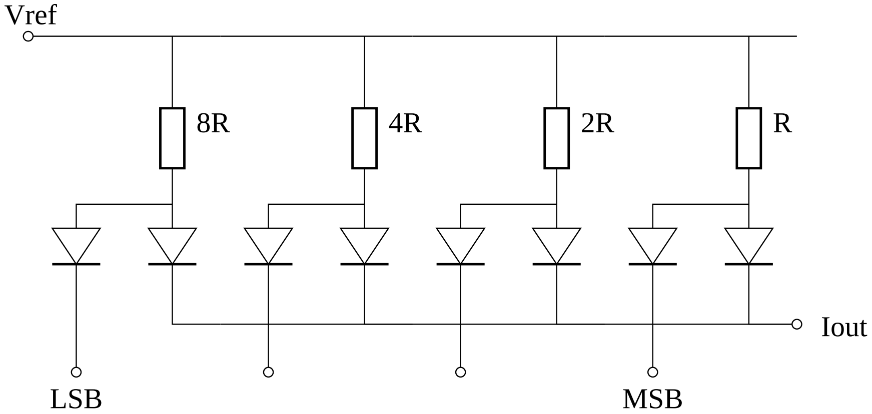

- Inherently monotonic
- No glitches
- Limited number of bits ($2^N$ current sources!)
- Used as a sub-circuit in segmented DACs

# Binary weighted DAC

$$I_{out} = 2\frac{V_{ref}}{R}\cdot\frac{N}{2^n}$$

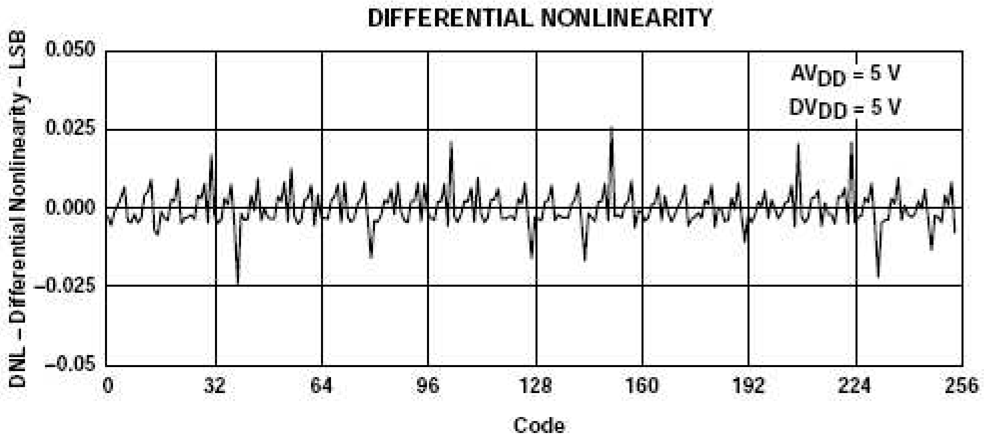
Number of bits *n*

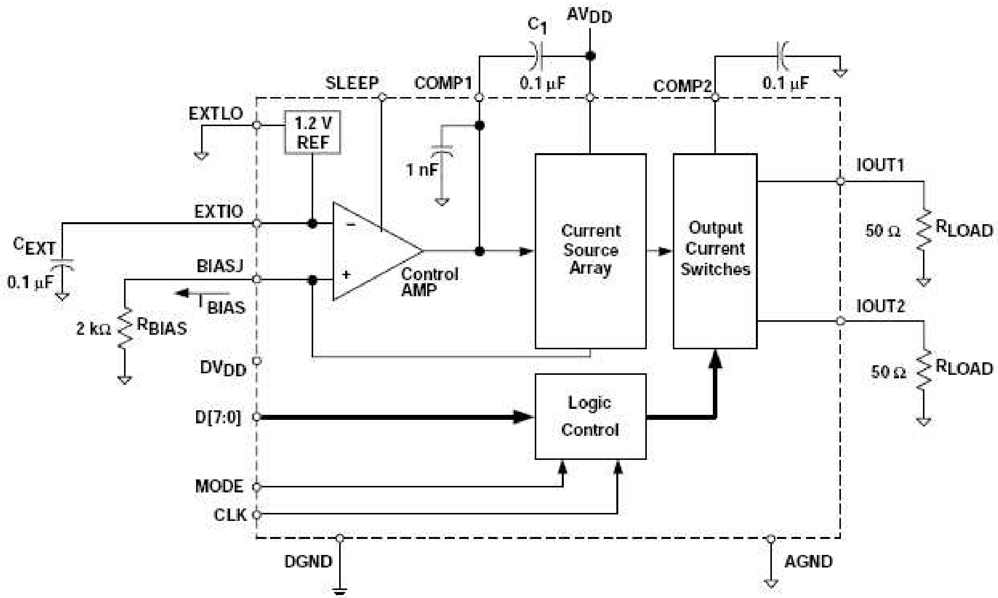
Applied input value *N*




Binary weighted resistor DAC:

## Example: THS5641, 8 bit, 100 MS/s, 35 ns settling time

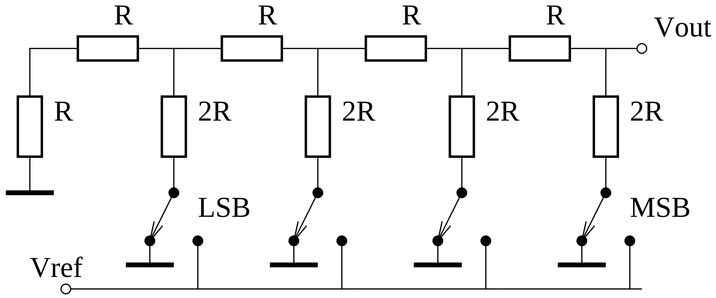


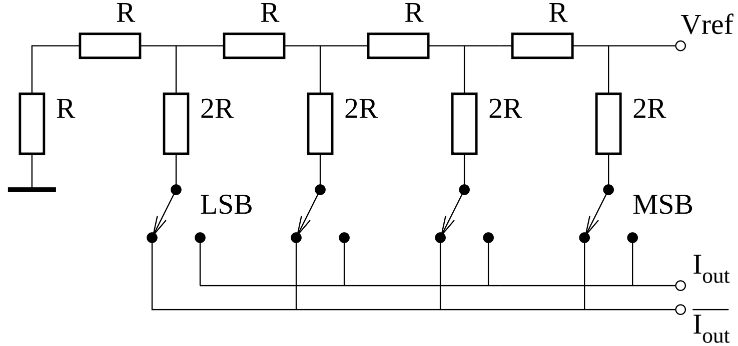
# R-2R Ladder

- Very common architecture
- Uses only two resistor values
- Voltage or current output
- Not very power efficient
- Often as multiplying DAC

$$V_{out} = V_{ref} \frac{N}{2^n}$$

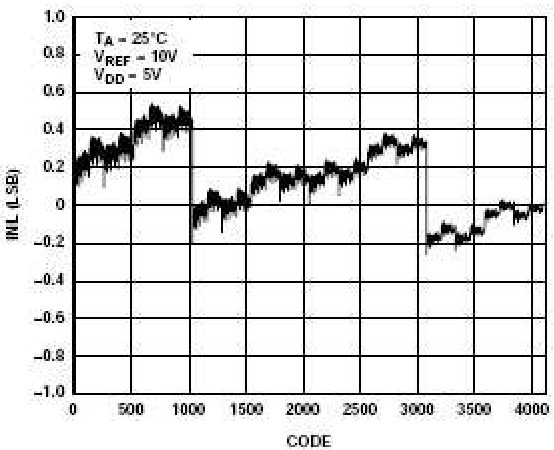


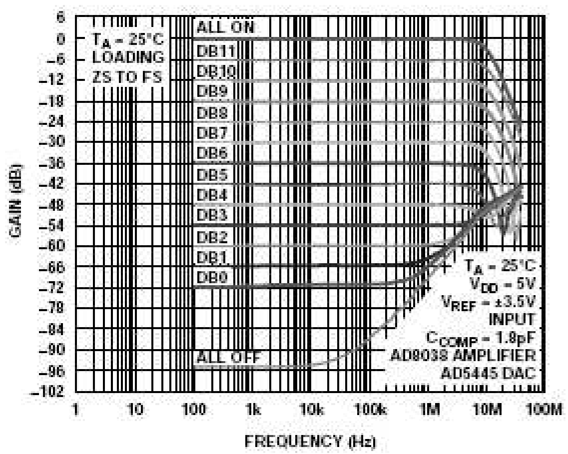

Voltage output R-2R DAC

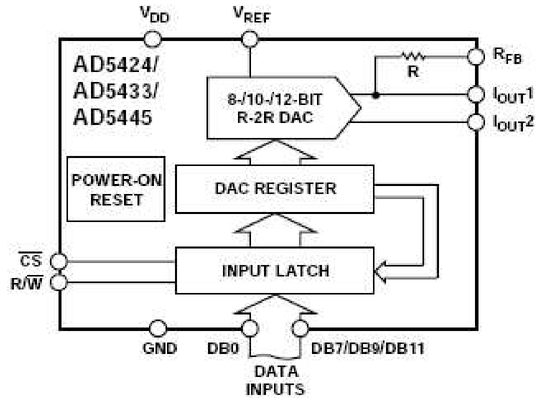



Current output R-2R DAC

Example: AD5445, 12 bit current output, 20.4 MS/s, 80 ns settling time

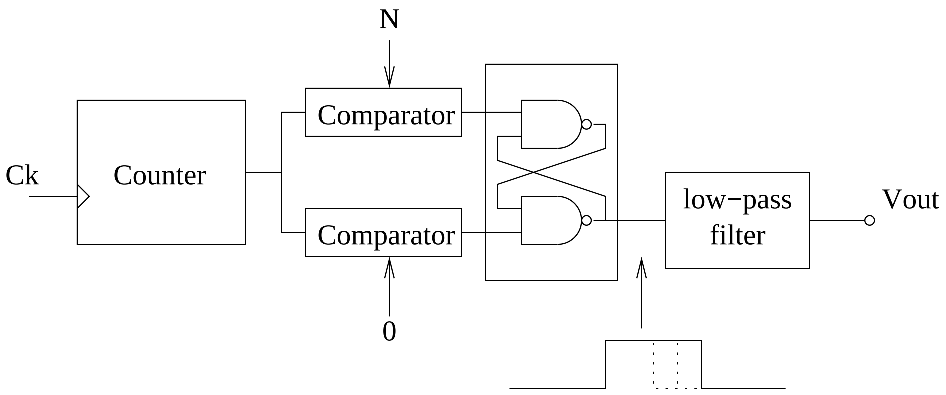



Reference input response vs. frequency and code

# PWM DAC

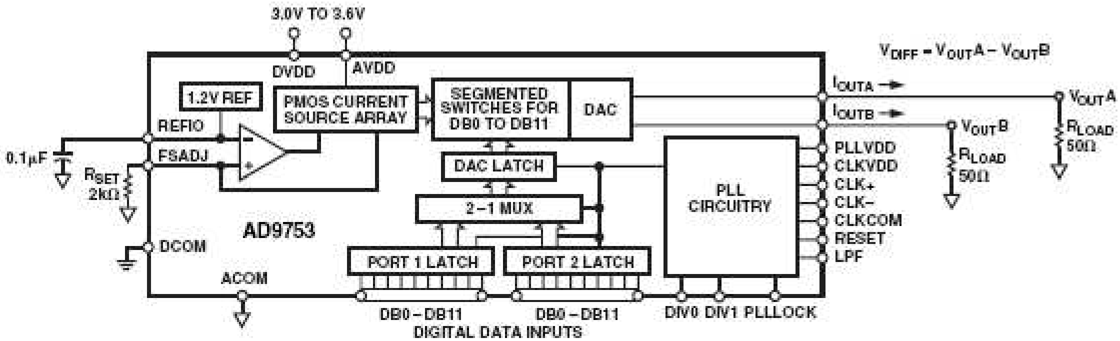



- Pulse Width Modulation
- Inherently linear
- Limited resolution
- Often used in μ-controller chips
- Applications: Motor control

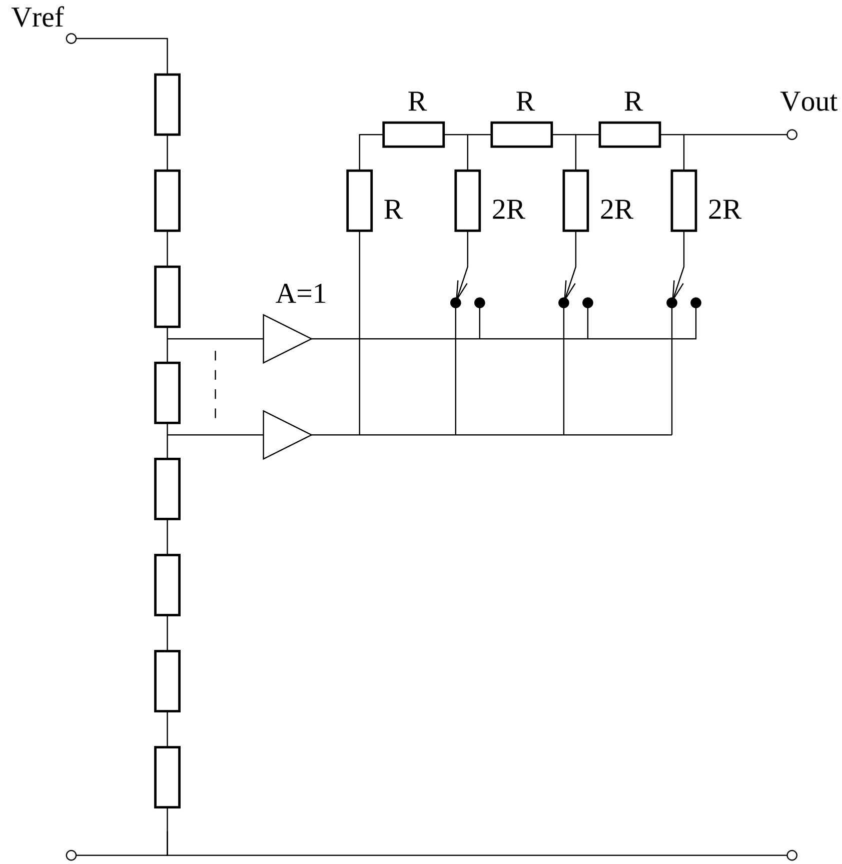
# Segmented DACs

Combination of other architectures, to optimize speed, linearity, SFDR, glitch energy, etc.
Common for high performance DACs used in instrumentation and communication equipment.

Example: AD9753, (12 bit, 300 MS/s) combines two thermometer (5 and 4 bits) sections and a binary weighted stage (3 bits).

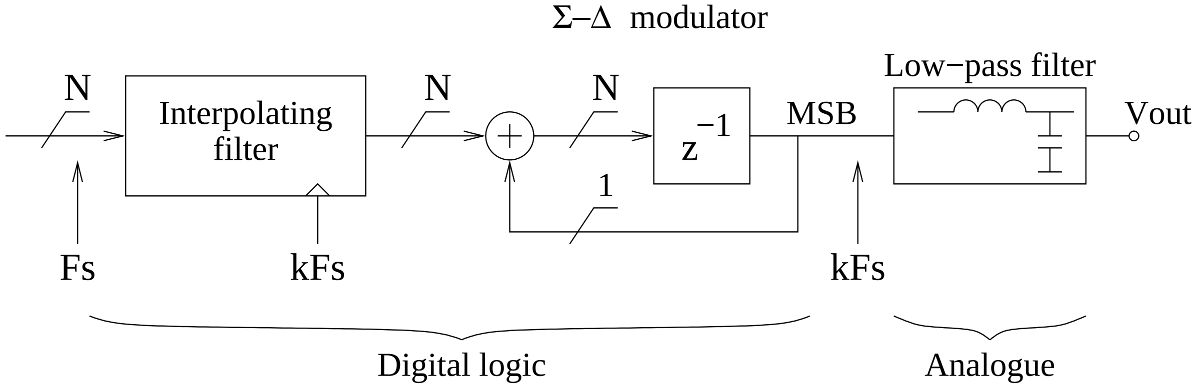


AD9753 block diagram



Combining a Kelvin divider with an R2R ladder

# Σ-Δ DAC




- Inherently linear
- Complex
- Σ–Δ modulator entirely digital
- Usually has a 1-bit DAC, sometimes multi-bit
- Large oversampling ratio eases output filter design

Example: AD1955 audio DAC
Update rate 192 kHz, SNR 120 dB