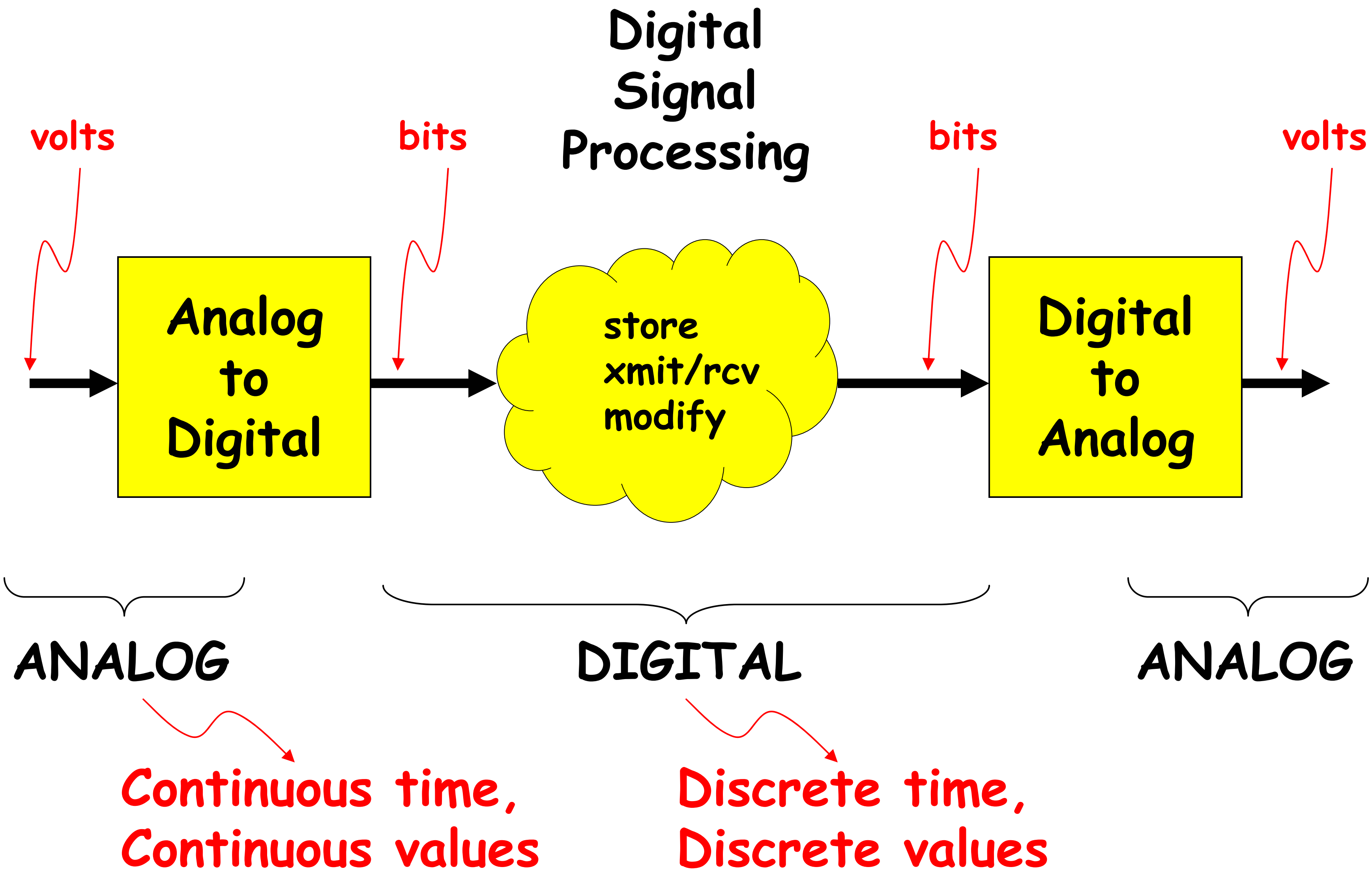
Analog↔Digital

- Sampling & Discrete Time
- Discrete Values & Noise
- Digital-to-Analog Conversion
- Analog-to-Digital Conversion

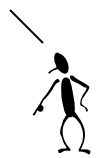
Plan: Mixed Signal Architecture



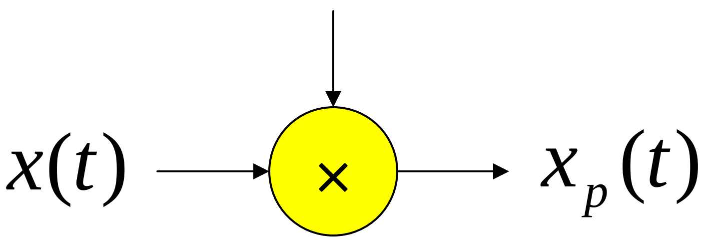
Discrete Time

Let's use an impulse train to sample a continuous-time function at a regular interval T:

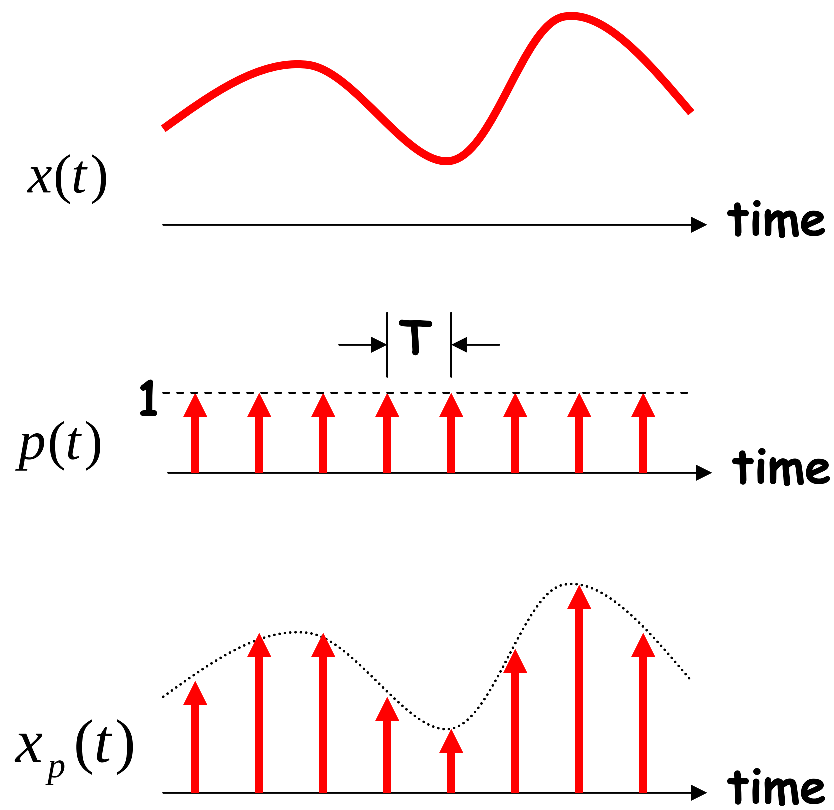
δ(x) is a narrow impluse at x=0, where $\int_{-\infty}^{\infty} f(t)\delta(t-a)dt = f(a)$



$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

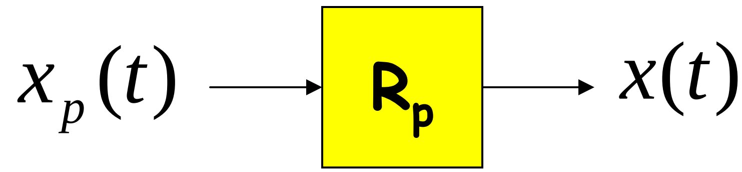


Time Domain

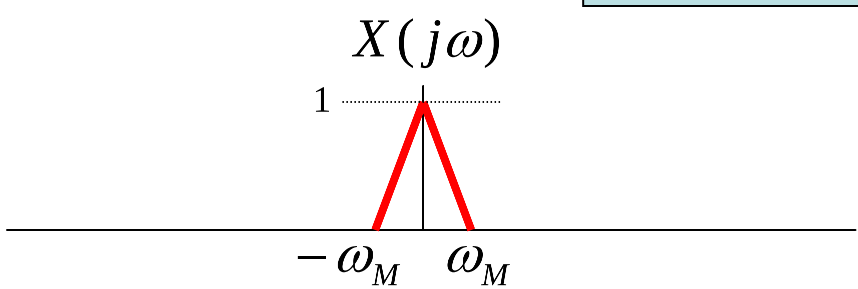


Reconstruction

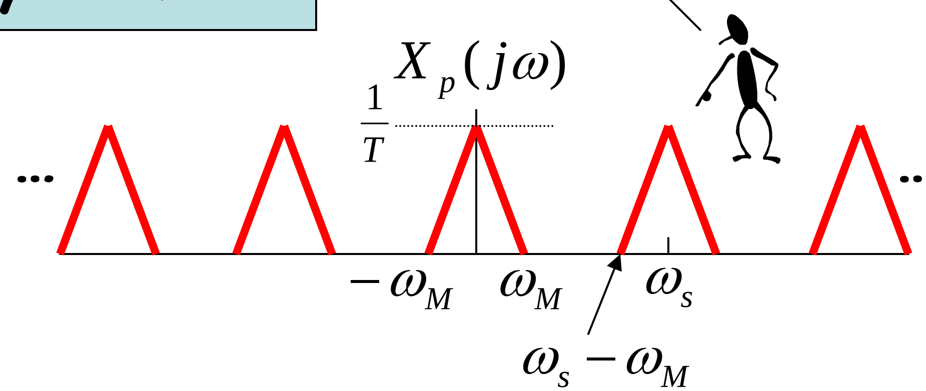
Is it possible to reconstruct the original waveform using only the discrete time samples?

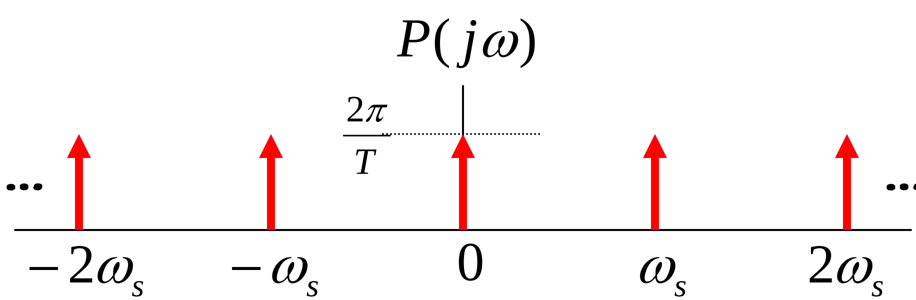


Frequency Domain



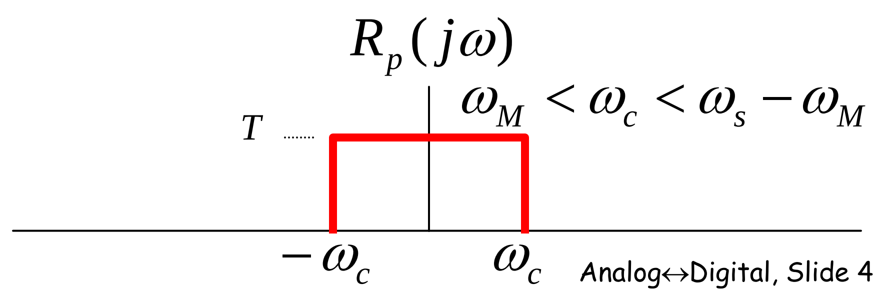
Looks like modulation by ω_s and its harmonics





$$\omega_s = \frac{2\pi}{T}$$

So, if $\omega_m < \omega_s - \omega_m$, we can recover the original waveform with a low-pass filter!



Sampling Theorem

Let x(t) be a band-limited signal with $X(j\omega)=0$ for $|\omega| > \omega_M$. Then x(t) is uniquely determined by its samples x(nT), n = 0, ±1, ±2, …, if

$$\omega_s > 2\omega_M$$

$2\omega_M$ is called the "Nyquist rate" and $\omega_s/2$ the "Nyquist frequency"

where

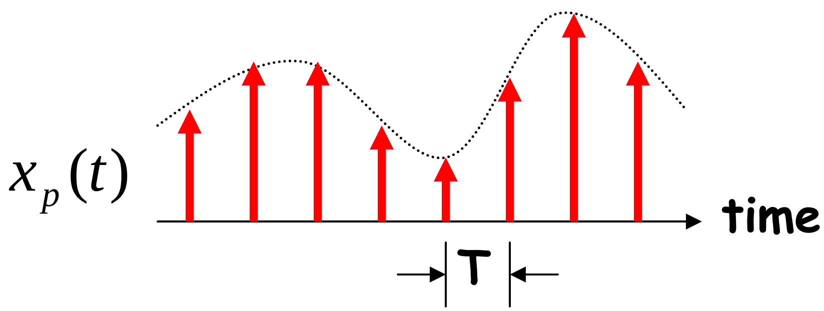
$$\omega_s = \frac{2\pi}{T}$$

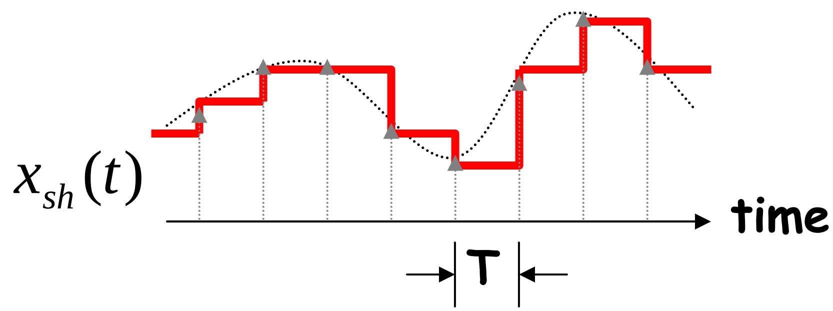
Given these samples, we can reconstruct x(t) by generating a periodic impluse train in which successive impulses have amplitudes that are successive sample values, then passing the train through an ideal LPF with gain T and a cutoff frequency greater than ω_M and less than $\omega_s - \omega_M$.

Zero-Order Sample & Hold

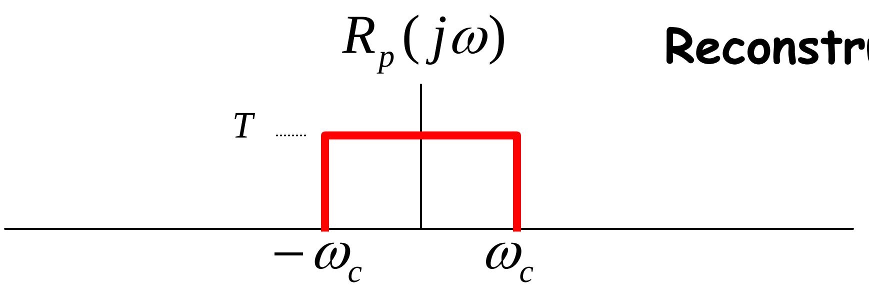
Impluses are hard to engineer, so a zero-order sample & hold is often used to produce the discrete time waveform.

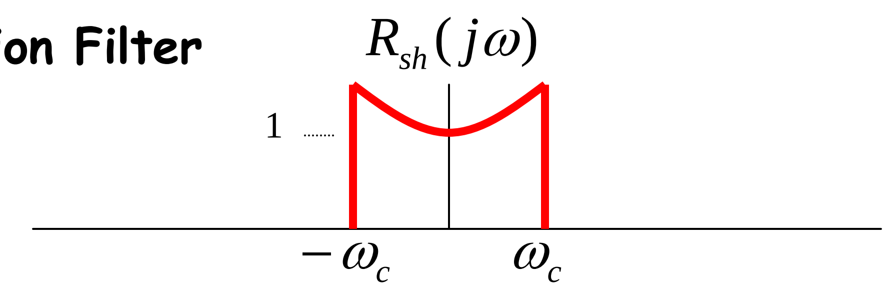
Sample Method





Reconstruction Filter





$$\omega_M < \omega_c < \omega_s - \omega_M$$

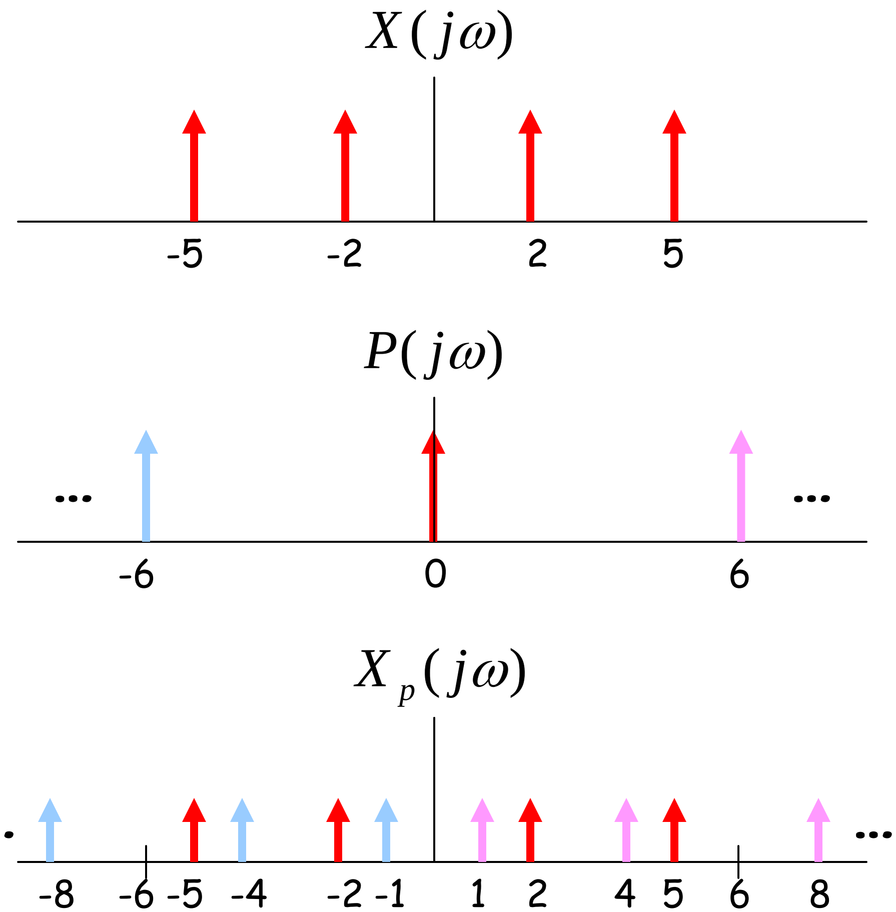
$$\omega_s = \frac{2\pi}{T}$$

See Chapter 7 in *Signals and Systems* by Oppenheim & Willsky (6.003)

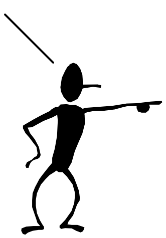
Undersampling → Aliasing

If $\omega_s \le 2\omega_M$ there's an overlap of frequencies between one image and its neighbors and we discover that those overlaps introduce additional frequency content in the sampled signal, a phenomenon called aliasing.

$$\omega_M = 5, \omega_s = 6$$



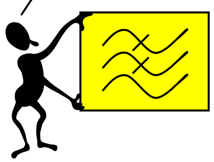
There are now tones at 1 (= 6 – 5) and 4 (= 6 – 2) in addition to the original tones at 2 and 5.



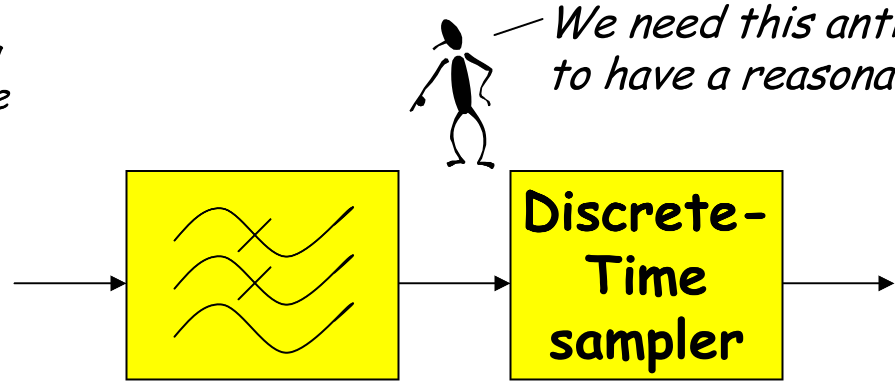
Antialias Filters

If we wish to create samples at some fixed frequency ω_s, then to avoid aliasing we need to use a low-pass filter on the original waveform to remove any frequency content $\geq \omega_s/2$.

This is the symbol for a low-pass filter – see the little "x" marks on the middle and high frequecies?



We need this antialiasing filter – it has to have a reasonably sharp cutoff



$$\omega_C = \frac{\omega_s}{2}$$

$$\omega_s$$

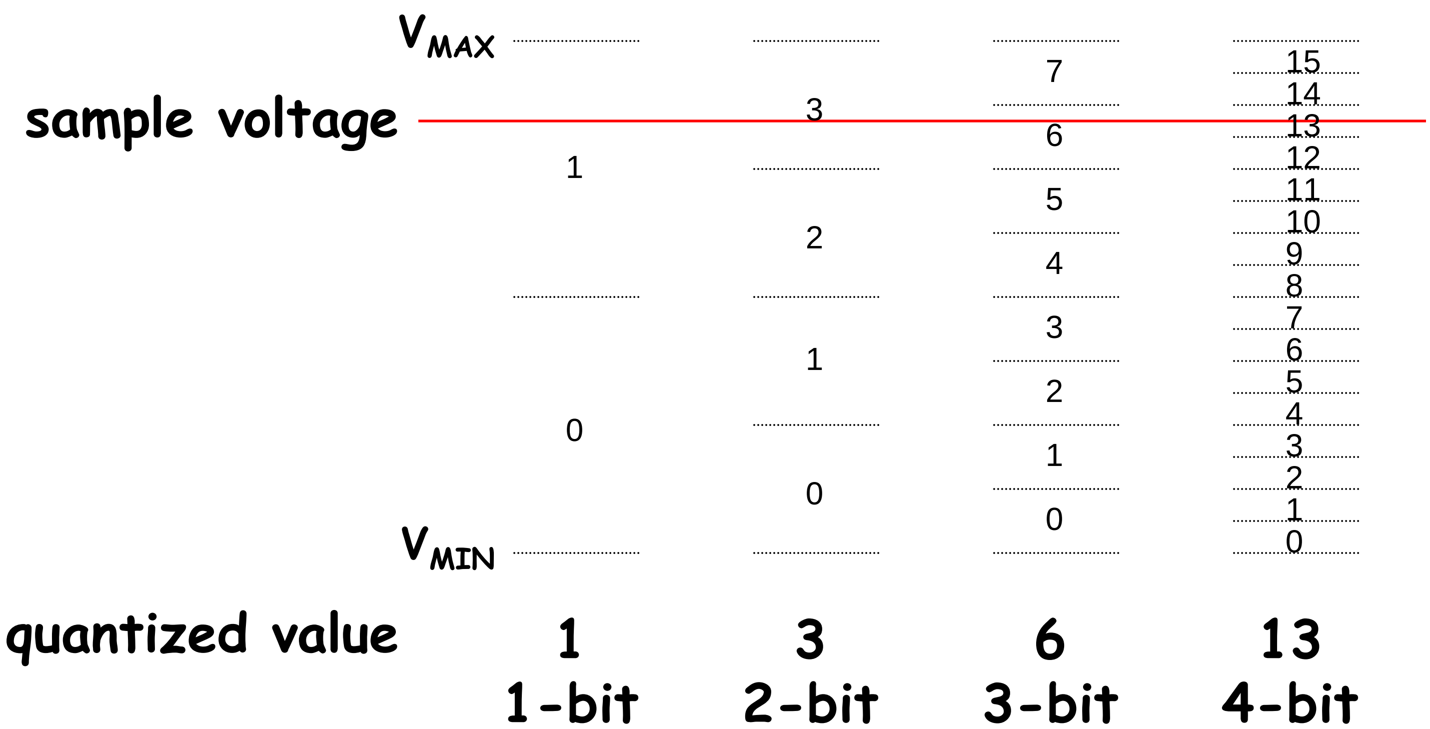
The frequency response of human ears essentially drops to zero above 20kHz. So the "Red Book" standard for CD Audio chose a 44.1kHz sampling rate, yielding a Nyquist frequency of 22.05kHz. The 2kHz of elbow room is needed because practical antialiasing filters have finite slope…

Why 44.1kHz? See http://www.cs.columbia.edu/~hgs/audio/44.1.html

Discrete Values

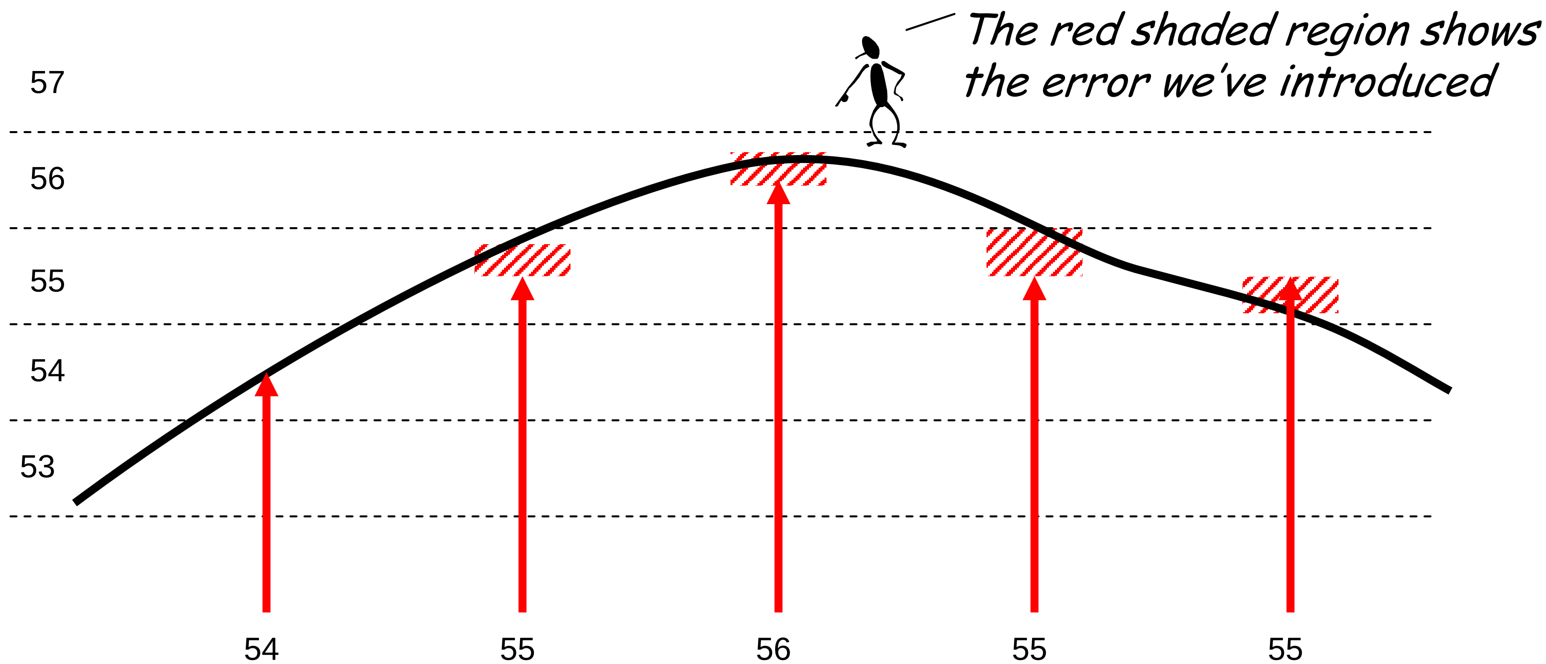
If we use N bits to encode the magnitude of one of the discrete-time samples, we can capture 2^N possible values.

So we'll divide up the range of possible sample values into 2^N intervals and choose the index of the enclosing interval as the encoding for the sample value.

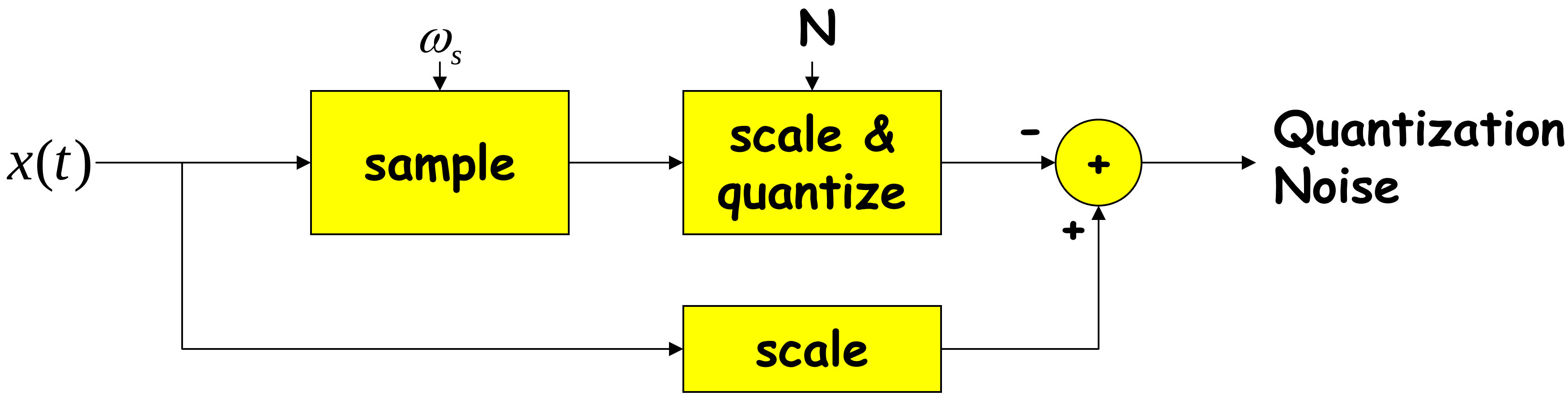


Quantization Error

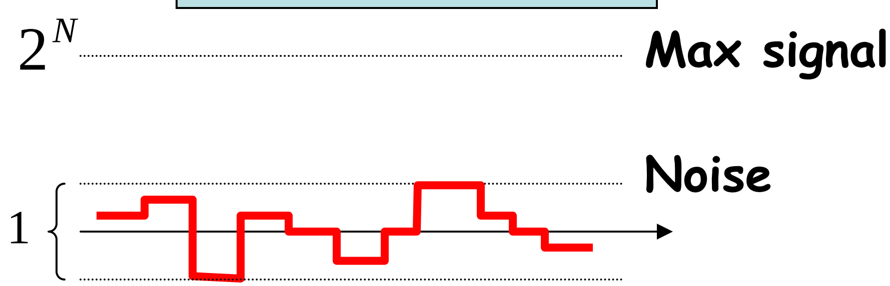
Note that when we quantize the scaled sample values we may be off by up to ±½ step from the true sampled values.



Quantization Noise



Time Domain

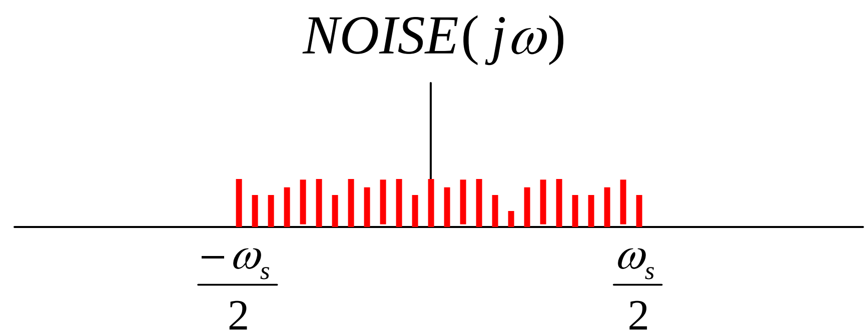


$$SNR = 20\log_{10}\left(\frac{A_{signal}}{A_{noise}}\right) = 20\log_{10}(2^N)$$

$$= N \cdot 6.02dB$$

+ 1.761dB if x(t) is a sine wave

Freq. Domain

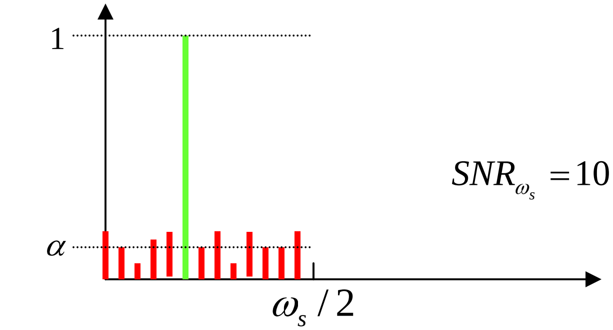


In most cases it's "white noise" with a uniform frequency distribution

Oversampling

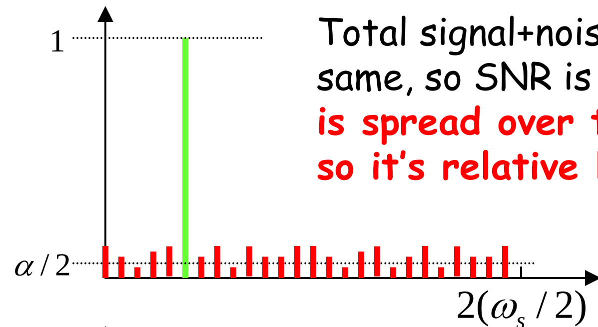
To avoid aliasing we know that ω_s must be at least $2\omega_M$. Is there any advantage to oversampling, i.e., $\omega_s = K \cdot 2\omega_M$?

Suppose we look at the frequency spectrum of quantized samples of a sine wave: (sample freq. = ω_s)



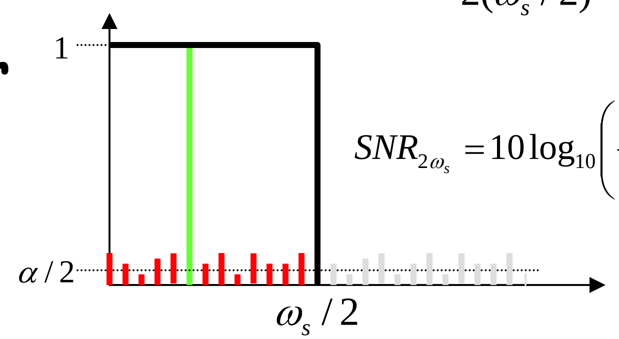
$$SNR_{\omega_s} = 10\log_{10}\left(\frac{P_{SIGNAL}}{P_{NOISE}}\right)$$

Let's double the sample frequency to $2\omega_s$.



Total signal+noise power remains the same, so SNR is unchanged. **But noise is spread over twice the freq. range so it's relative level has dropped.**

Now let's use a low pass filter to eliminate half the noise! Note that we're not affecting the signal at all…



$$SNR_{2\omega_s} = 10\log_{10}\left(\frac{P_{SIGNAL}}{P_{NOISE}/2}\right) = SNR_{\omega_s} + 3dB$$

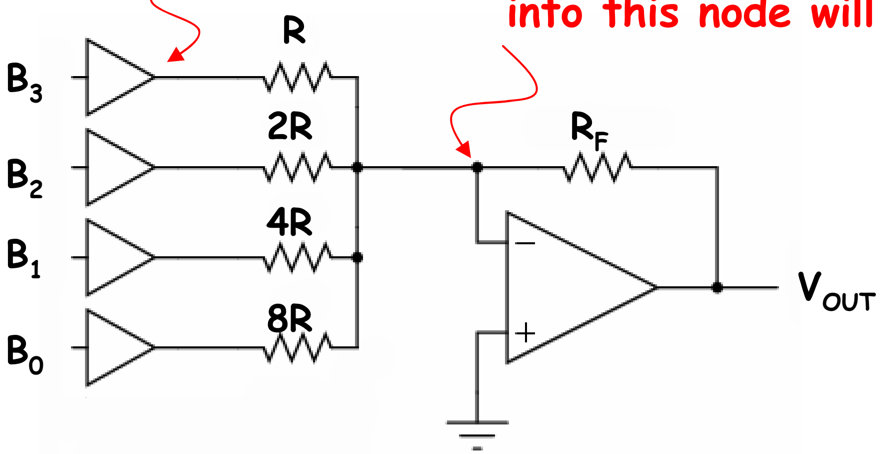
Oversampling+LPF reduces noise by 3dB/octave

DAC: digital to analog converter

How can we convert a N-bit binary number to a voltage?

V_i = 0 volts if B_i = 0
V_i = V volts if B_i = 1

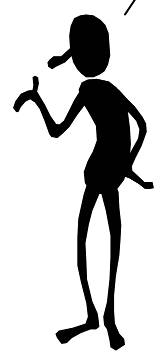
OPAMP will vary V_{OUT} to maintain this node at 0V, i.e., the sum of the currents flowing into this node will be zero.



$$\frac{V_{OUT}}{R_F} + \frac{B_3 V}{R} + \frac{B_2 V}{2R} + \frac{B_1 V}{4R} + \frac{B_0 V}{8R} = 0$$

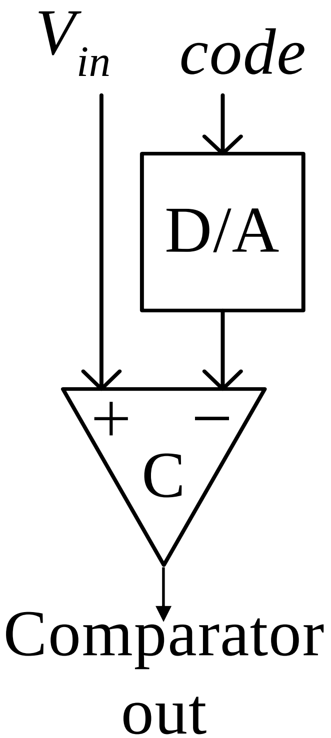
$$V_{OUT} = -\frac{R_F}{R} V \left(B_3 + \frac{B_2}{2} + \frac{B_1}{4} + \frac{B_0}{8} \right)$$

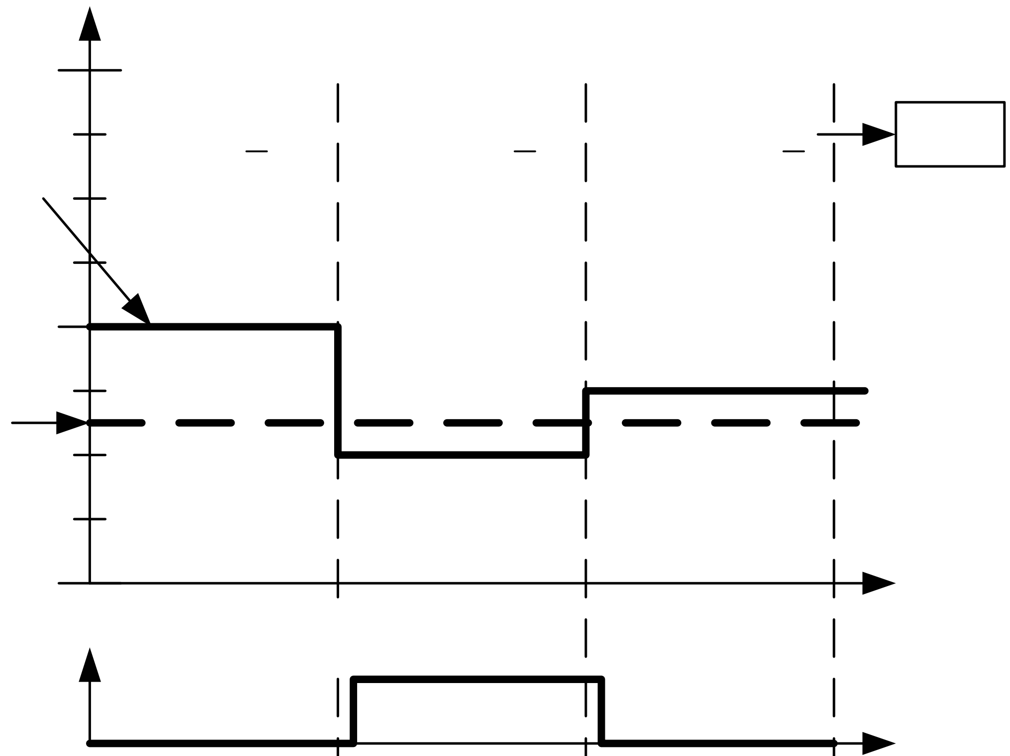
OKAY, this'll work, but the voltages produced by the drivers and various R's must be carefully matched in order to get equal steps.



Successive-Approximation A/D

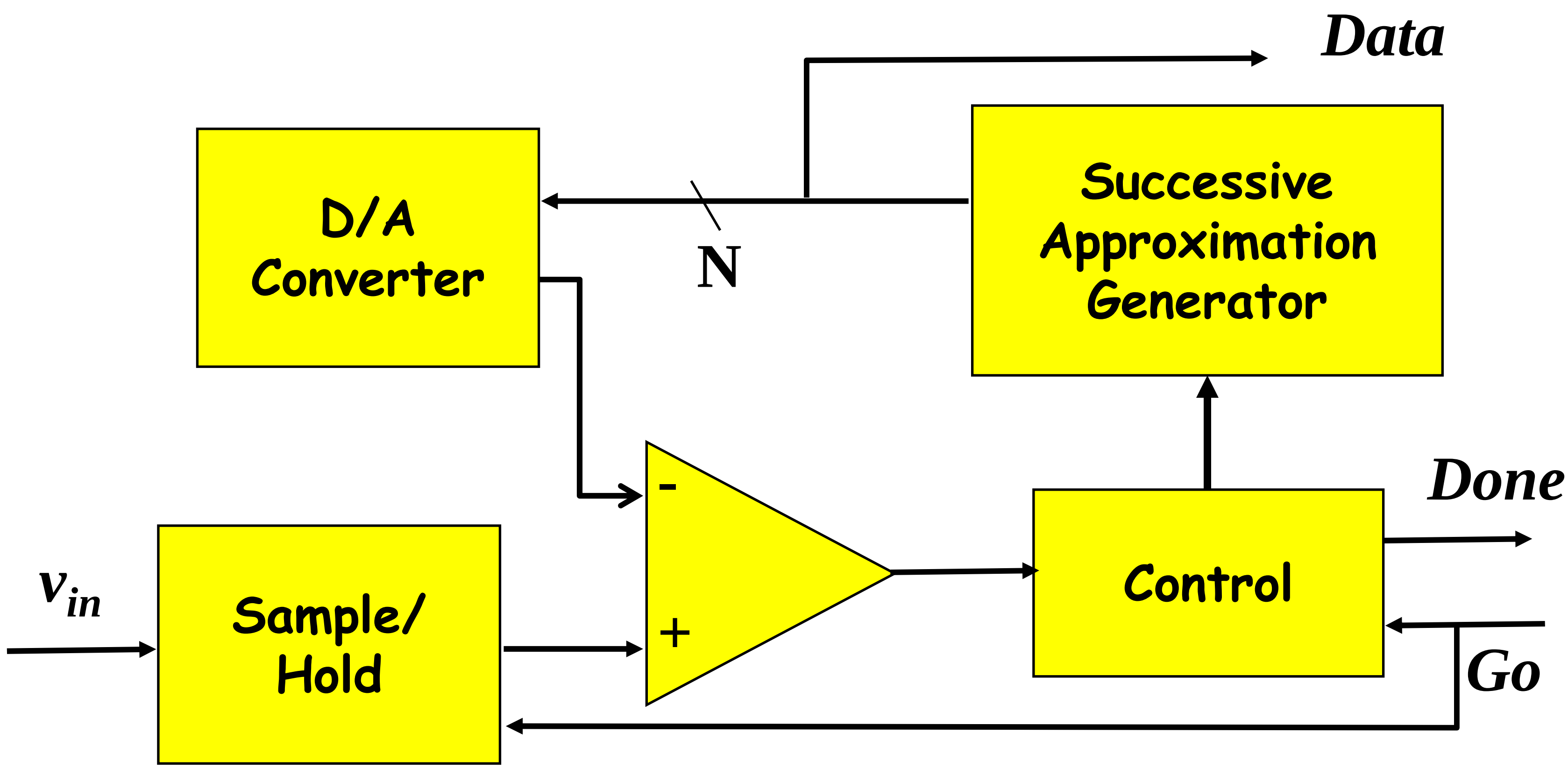
- D/A converters are typically compact and easier to design. Why not A/D convert using a D/A converter and a comparator?
- DAC generates analog voltage which is compared to the input voltage
- If DAC voltage > input voltage then set that bit; otherwise, reset that bit
- This type of ADC takes a fixed amount of time proportional to the bit length





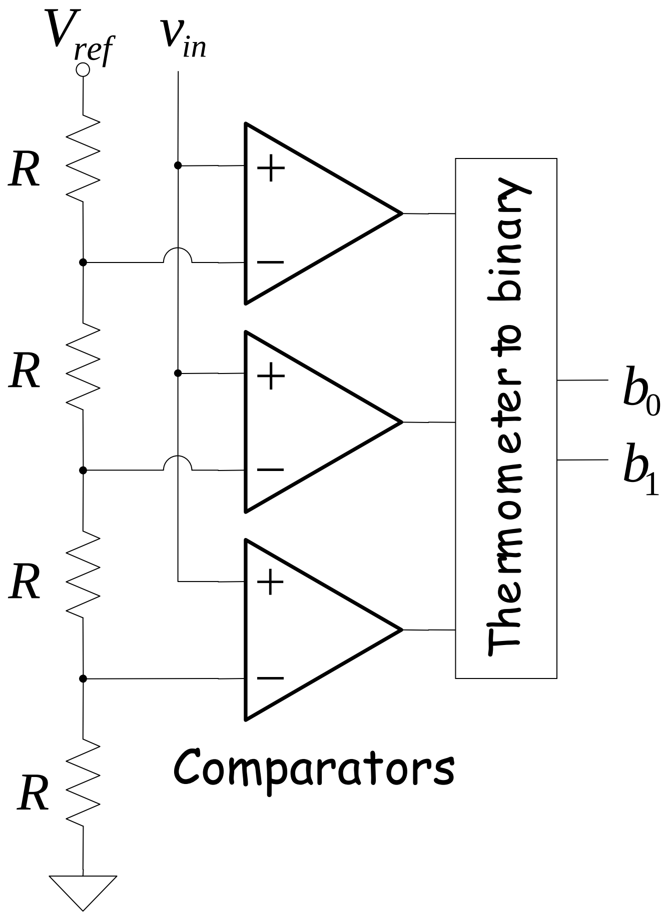
Example: 3-bit A/D conversion, 2 LSB < V_{in} < 3 LSB

Successive-Approximation A/D



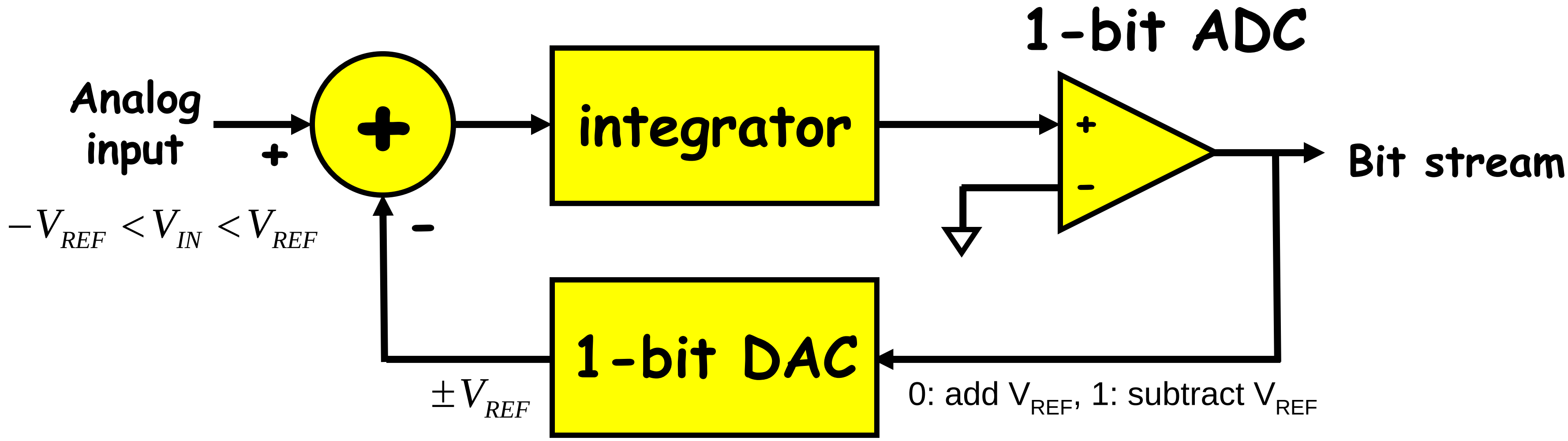
Serial conversion takes a time equal to $N(t_{D/A} + t_{comp})$

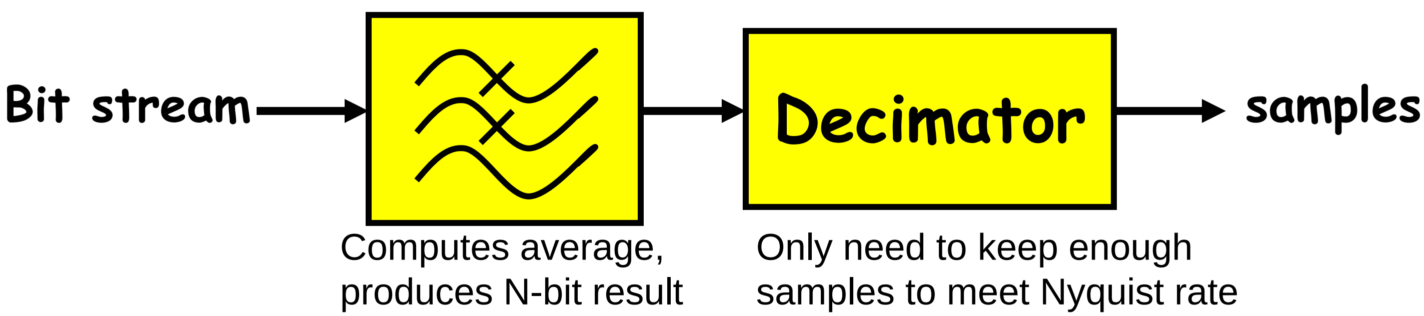
Flash A/D Converter



- Brute-force A/D conversion
- Simultaneously compare the analog value with every possible reference value
- Fastest method of A/D conversion
- Size scales exponentially with precision (requires 2^N comparators)

Sigma Delta ADC





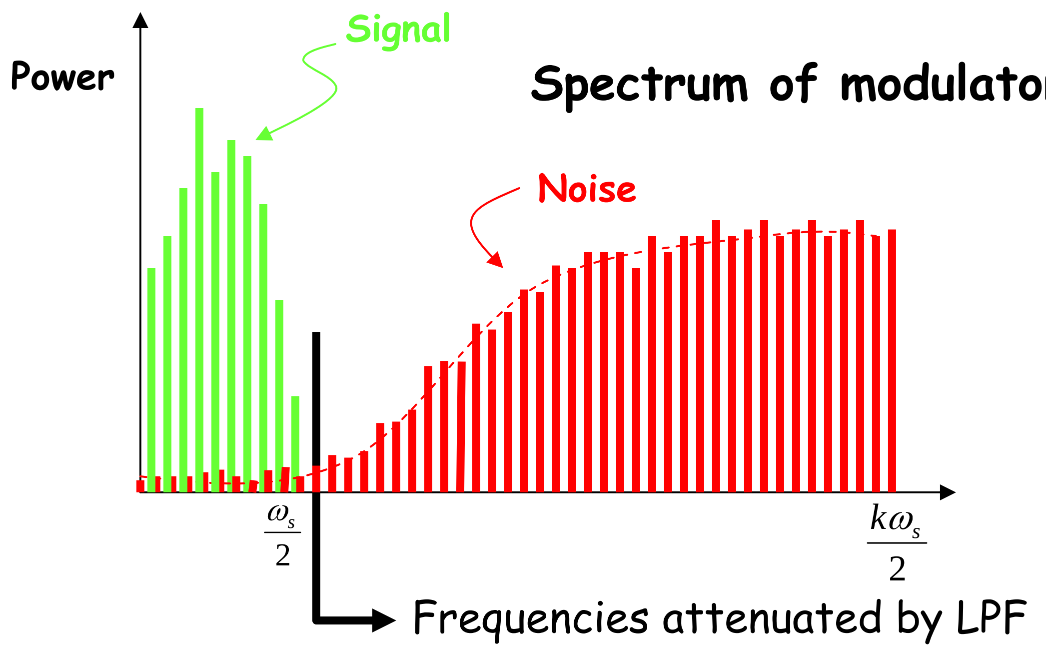
Average of bit stream ($1=V_{REF}$, $0=-V_{REF}$) gives voltage

With V_{REF}=1V: V_{IN}=0.5: 1110…, V_{IN}=-0.25: 00100101…, V_{IN}=0.6: 11110

http://www.analog.com/Analog_Root/static/techSupport/designTools/interactiveTools/sdtutorial/sdtutorial.html

So, what's the big deal?

- Can be run at high sampling rates, oversampling by, say, 8 or 9 octaves for audio applications; low power implementations
- Feedback path through the integrator changes how the noise is spread across the sampling spectrum.



- Pushing noise power to higher frequencies means more noise is eliminated by LPF: N^{th} order ΣΔ SNR = (3+N*6)dB/octave

Summary

- If we sample a band-limited signal x(t) to produce discrete-time samples x(nT), then if $\omega_s > 2\omega_M$, then we can reconstruct the original waveform by passing the impulse samples through a LPF.
- Undersampling ($\omega_s \leq 2\omega_M$) will result in aliasing. Usually an antialiasing filter is used before sampling to avoid this problem.
- Quantizing the sampled values into 2^N discrete levels introduces quantization noise (SNR ~ 6dB*N)
- Oversampling reduces noise by 3dB/octave
- DAC architectures: summing node
- ADC architectures: successive approx., flash, sigma delta (combines oversampling + noise shaping)