

# What did we do last time?

## PCB vs silicon

What are the differences when scaling up the geometries?

## Transmission lines

Termination and other properties of "large" wires

# What will we do today?

## Filters

A quick view of filters for an analog signal chain

## Supply filters

Decoupling (bypass) capacitors and supply filters

## Data converters

**(dependent on time, might be pushed out to next lecture)**

An overview of data converters

An understanding of the frequency domain

The oversampling data converter

# Analog filters

## Linear, frequency-selective filter

### Different types of filters

Band-select filters	(for telecom and noise reduction)
Anti-aliasing filters	(for the ADC)
Reconstruction filters	(for the DAC)
Notch filters	(for disturbances at specific frequencies)

### Effectively realizing an LTI transfer function

$$H(s) = H_0 \cdot \frac{a_N \cdot s^N + \dots + a_1 \cdot s + a_0}{b_M \cdot s^M + \dots + b_1 \cdot s + b_0}$$

# Analog filters, cont'd

A general transfer function for a linear system is given by

$$H(s) = \frac{Y(s)}{X(s)} = \frac{a_0 + a_1s + a_2s^2 + \dots}{b_0 + b_1s + b_2s^2 + \dots}$$

$$Y(s) \cdot (b_0 + b_1s + b_2s^2 + \dots) = X(s) \cdot (a_0 + a_1s + a_2s^2 + \dots)$$

$$Y(s) = X(s) \cdot \left( \alpha_0 + \alpha_1 \frac{1}{s} + \alpha_2 \frac{1}{s^2} + \dots \right) - Y(s) \cdot \left( \beta_1 \frac{1}{s} + \beta_2 \frac{1}{s^2} + \dots \right)$$

Create a “recursive” set of integrations

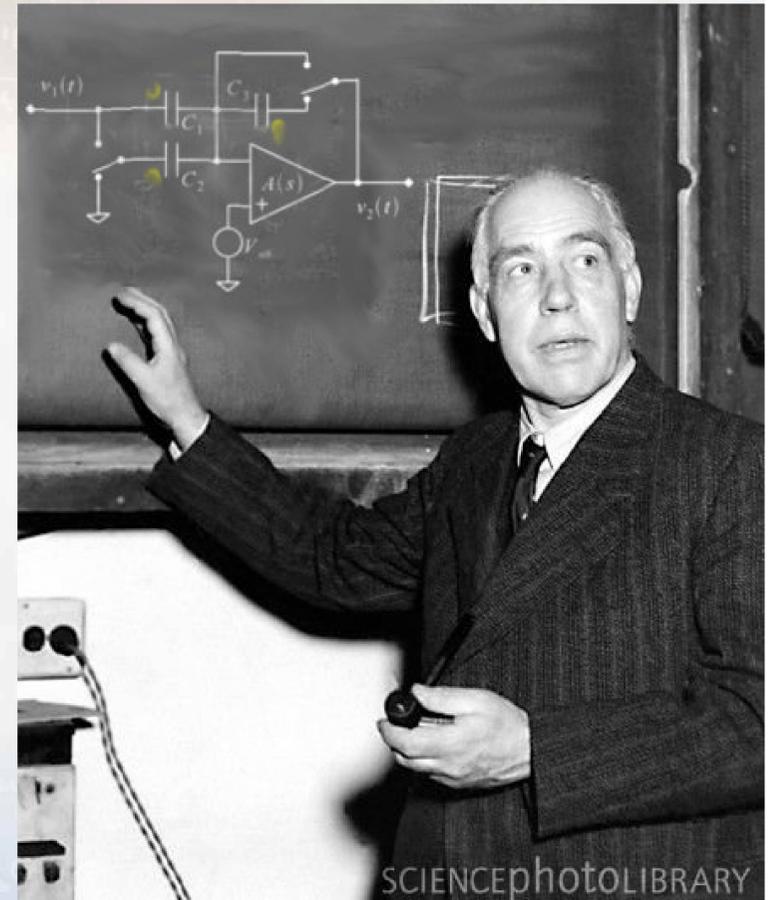
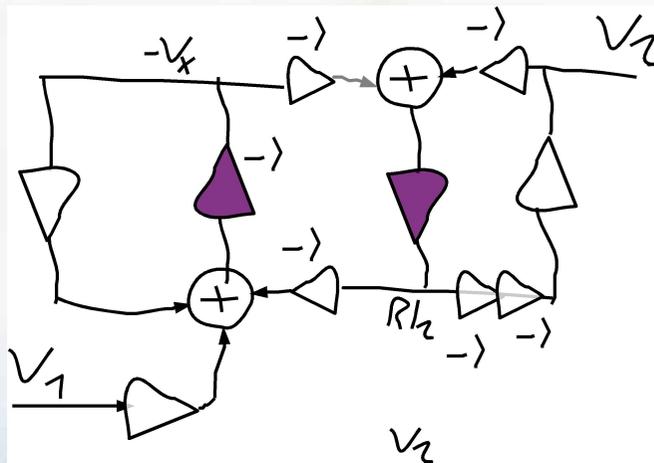
$$Y(s) = \alpha_0 \cdot X(s) + \frac{1}{s} \cdot \left( \alpha_1 X(s) - \beta_1 \cdot Y(s) + \frac{1}{s} \cdot \left( \alpha_2 \cdot X(s) - \beta_2 \cdot Y(s) + \frac{1}{s} \cdot (\dots) \right) \right)$$

# Continuous-time filters, flow graph

## Manipulation

Replacing with integrators

## Feedback, etc.



## Cascading of several stages

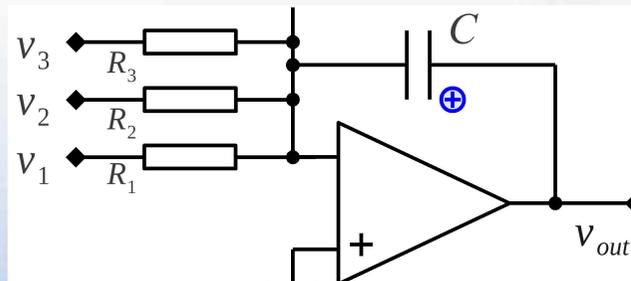
# Active-RC

Summation of different inputs is done with the resistors, i.e., we are summing up the currents in the virtual ground node:

$$-V_{out}(s) \cdot s C_L = I_1(s) + I_2(s) + I_3(s) = \frac{V_1(s)}{R_1} + \frac{V_2(s)}{R_2} + \dots$$

Which combined gives us the integration

$$V_{out}(s) = -\frac{V_1(s)}{s C_L R_1} - \frac{V_2(s)}{s C_L R_2} + \dots$$



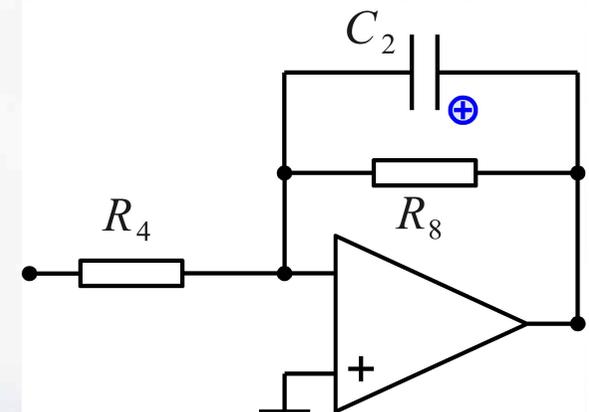
# Example, first-order pole with active-RC

Sum the currents in the virtual ground:

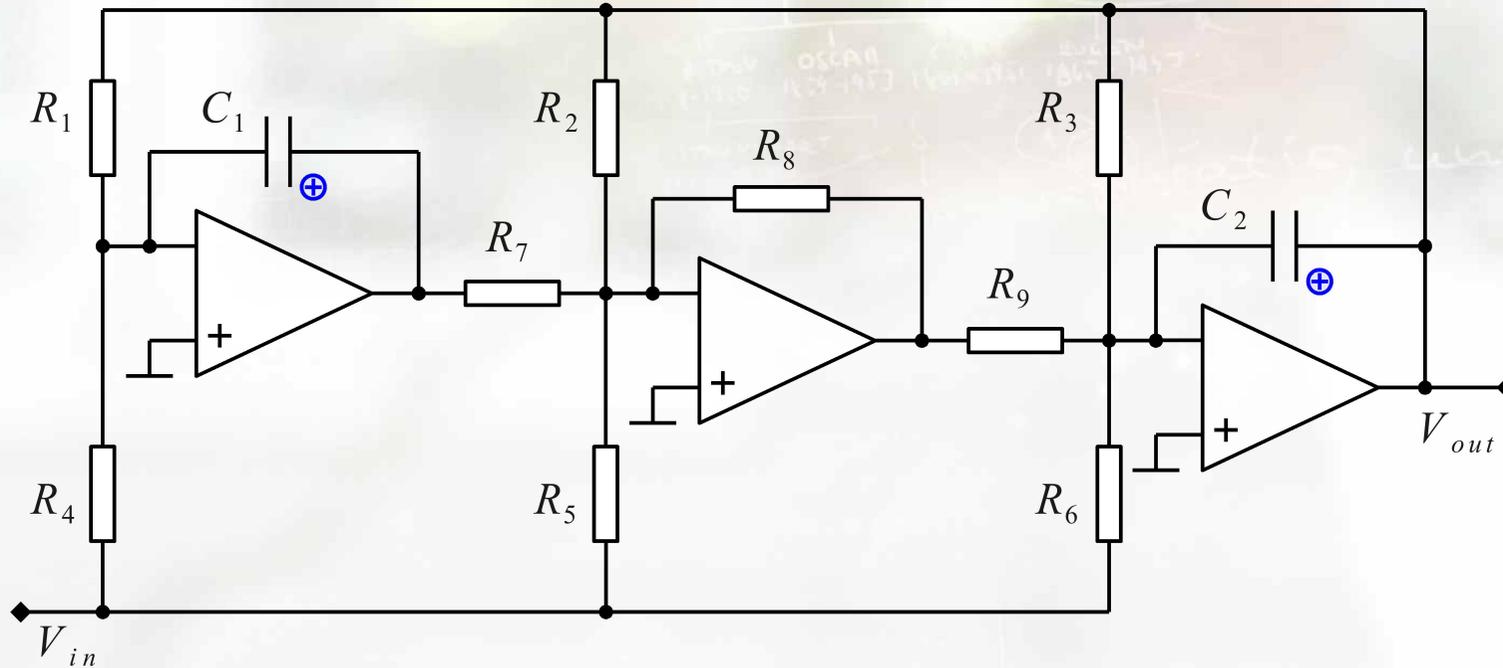
$$V_{out}(s) \cdot s C_2 = \frac{-V_{in}(s)}{R_4} - \frac{V_{out}(s)}{R_8}$$

such that

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{-R_8/R_4}{1 + \frac{s}{1/C_2 R_4}}$$



# Example: Tow-Thomas, Biquad, and more, ...



Output isolated and buffered with OP

Single-pole formed with single RC at output or input

# Analog filters, Architectures

## Ideally

Best performance is obtained with a leapfrog or ladder filter. Less sensitive to component variations

## Practically

Select a set of cascaded stages on the board instead.

Design for one extra pole whenever possible (two poles per opamp)

Quite a few software tools "out there" that provide you with the values once you know the specification.

Off-the shelf integrated analog circuits with filtering options

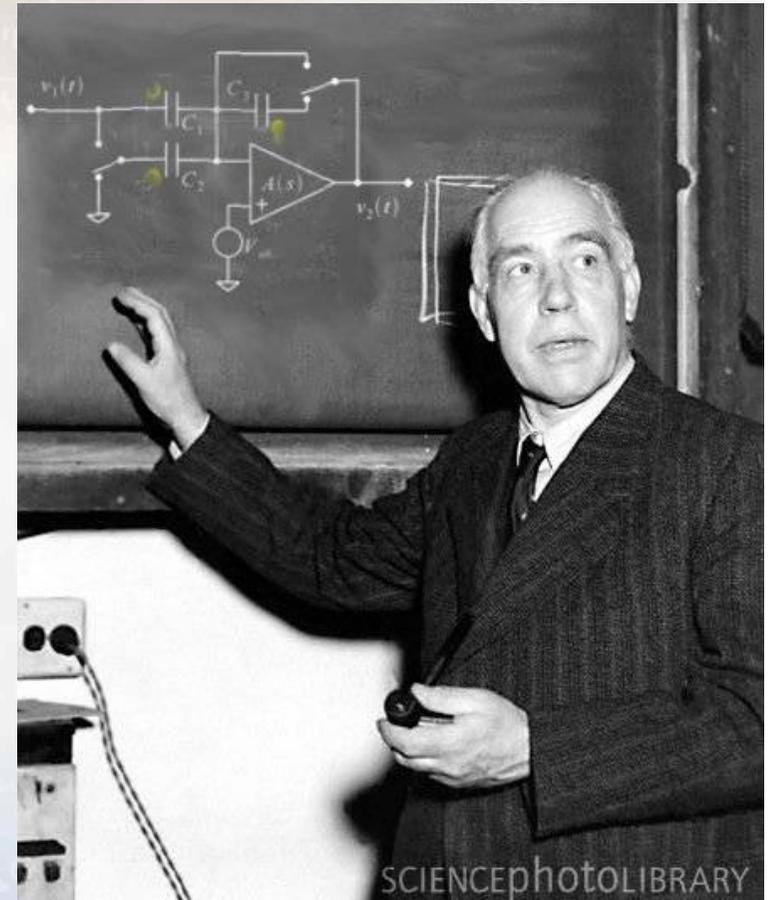
# Analog filters, Architectures, cont'd

## Comparison

Two lowpass filters with same, but different, properties.

## Scaling

Scale your signal levels to maximize the input/output range.



# Analog filters, opamp considerations

## Speed

Unity-gain frequency of the opamp should be considerably larger than the closed-loop gain times the cut-off frequency of your filter

$$f_{ug} > 100 \cdot A_{CL} \cdot f_{bw} \quad (\text{example})$$

## Current drive capability

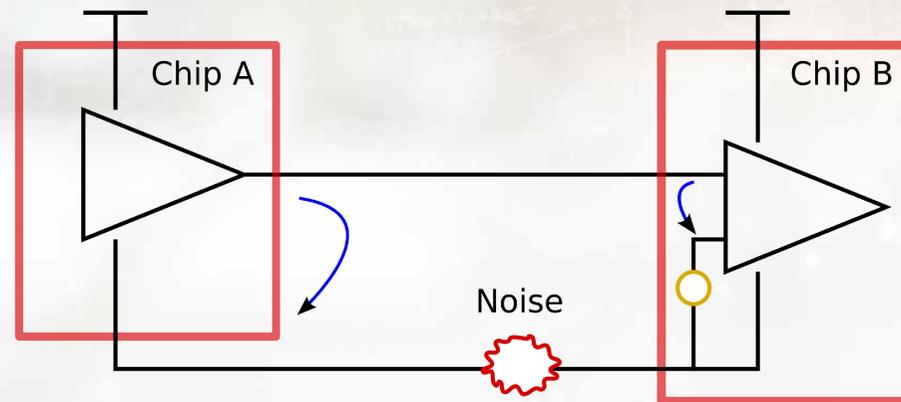
Slew rate must cope with the highest frequency of your filter:

$$|SR| > 2\pi \cdot f_{bw} \cdot V_{out,pp} \quad (\text{assuming sinusoid signal})$$

**Noise, limited gain we have discussed in previous lectures**

# Back to the PCB: supply filters

Receiver picks up noise on the ground (reference) wire



## Noise sources

External humm and interferers

Due to shared ground paths due to inductance in path

Differential signalling solves a lot (see next lecture)

# The power rules

## Three rules for power supplies/grounds

- 1) Use low-impedance ground connections between chips (gates).
- 2) The impedance between power pins on any two chips (gates) should be just as low as the impedance between ground pins.
- 3) There must be a low-impedance path between power and ground

# The power rules, cont'd

**These three rules can be satisfied if:**

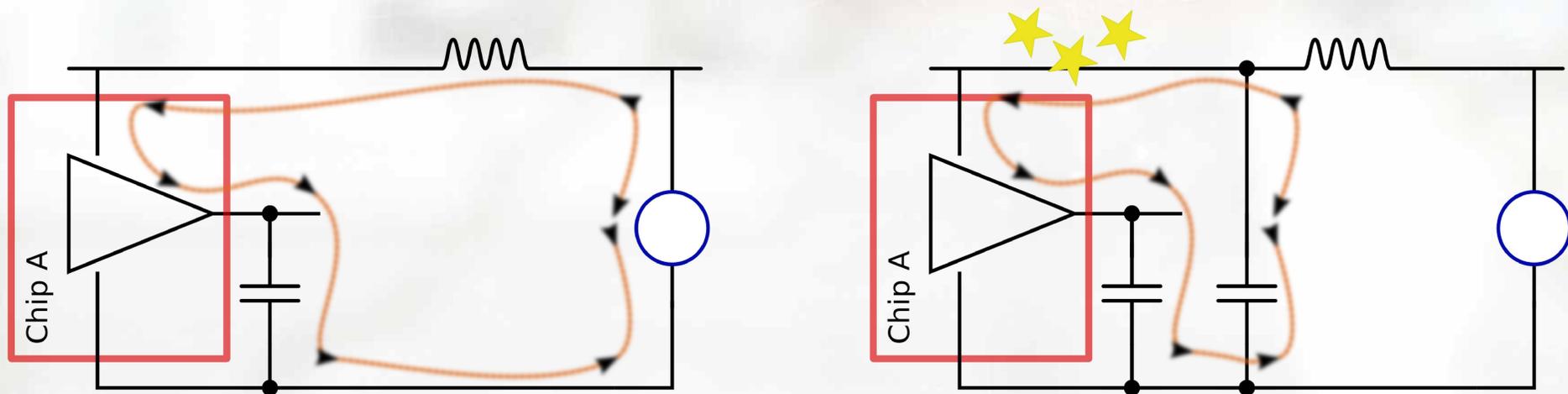
There is a good ground plane and there are bypass capacitors close to each chip/gate/component providing the low-impedance path.

Then, the supply wiring can be “sloppy”.

(However, best to use a power supply plane too).

# Supply filters (bypass capacitors)

Connecting a bypass capacitor locally also reduces the effect of inductance in the supply wires.



Supply has low impedance at low frequencies (inductance)

Capacitor has low impedance at high frequencies

# Example decoupling capacitor

## Values

Power inductance is  $L_P$  and that the max power drop  $\Delta V$

Chip switches  $N$  wires, each with a load of  $C_L$  during a time of  $t_{rf}$

## Some relationships

$$\Delta I = N \cdot C_L \cdot \frac{V_{dd}}{t_{rf}}, \quad \Delta V = X_{max} \cdot \Delta I, \quad X_{max} = 2\pi \cdot L_P \cdot f_x, \quad X_{max} = \frac{1}{2\pi \cdot N \cdot C_L \cdot f_x}$$

## Example values

$$V_{dd} = 1 V, N = 16, C_L = 10 pF, L_P = 100 nH, \Delta V = 100 mV, t_{rf} = 100 ps$$

# Example decoupling capacitor, cont'd

Combine and simplify

$$C_x = \frac{1}{2\pi \cdot f_{sw} \cdot Z_{max}} = \frac{1}{2\pi \cdot \frac{X_{max}}{2\pi L_P} Z_{max}} = \frac{L_P}{X_{max}^2} = \frac{L_P}{\left(\frac{\Delta V}{\Delta I}\right)^2} = \frac{L_P}{\left(\frac{\Delta V}{N \cdot C_L \cdot \frac{V_{dd}}{t_{rf}}}\right)^2}$$

The capacitance for which the reactance of C cancels the reactance of L

$$C_x = \frac{(N \cdot C_L \cdot V_{dd})^2 \cdot L_P}{(\Delta V \cdot t_{rf})^2} \text{ gives } C_x \approx 37 \mu F$$

# Decoupling (bypass) capacitors

## Parasitics

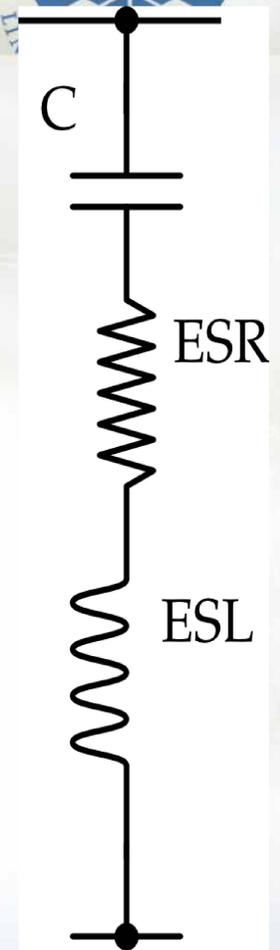
Lead/package/mounting inductance,  $L$

Equivalent series resistance due to parasitic resistance

## Effective inductance

$$Z_C(\omega) = \sqrt{ESR^2 + \left( \omega L - \frac{1}{\omega C} \right)^2}$$

The L and C can give rise to resonance effects when connected in parallel!



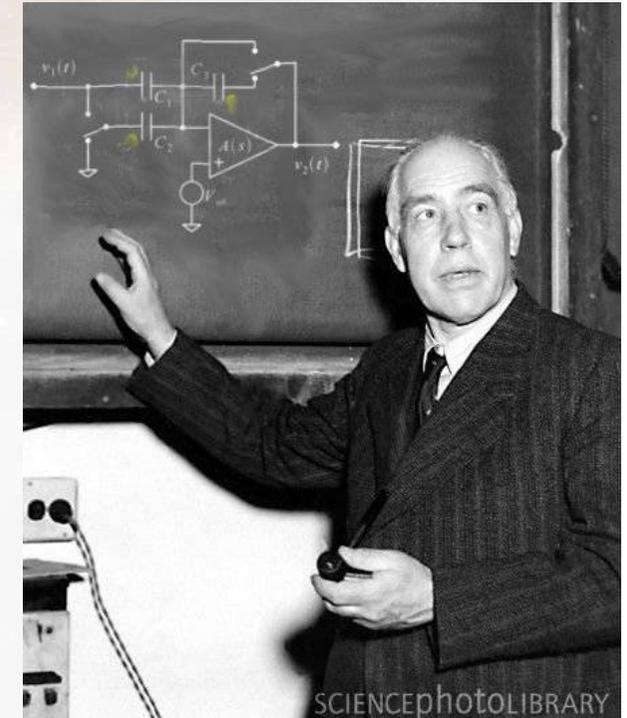
# Decoupling (bypass) capacitors

## Requirements on the reactance/impedance

0.1 to 1 Ohm

## Figure showing the impedance effects

- 1) Wiring inductance from supply
- 2) Decoupling capacitors
- 3) Decoupling capacitive bank
- 4) Power and ground plane



# Decoupling capacitors. Conclusions

Significant capacitance in the ground planes too, so put the ground and supply planes very close to each other!

The best way to get very low inductance is to parallel a larger set of smaller capacitors!

Laboratory 3 will guide you through this in more detail.

# Data converters fundamentals

## DAC

Represents a digital signal with an analog signal  
To control something  
To transmit something (a modulated signal)

## ADC

Represents an analog signal with a digital signal  
To measure something  
To receive something (a modulated signal)

## And there are others:

Time-to-digital converters  
Frequency-to-digital converters

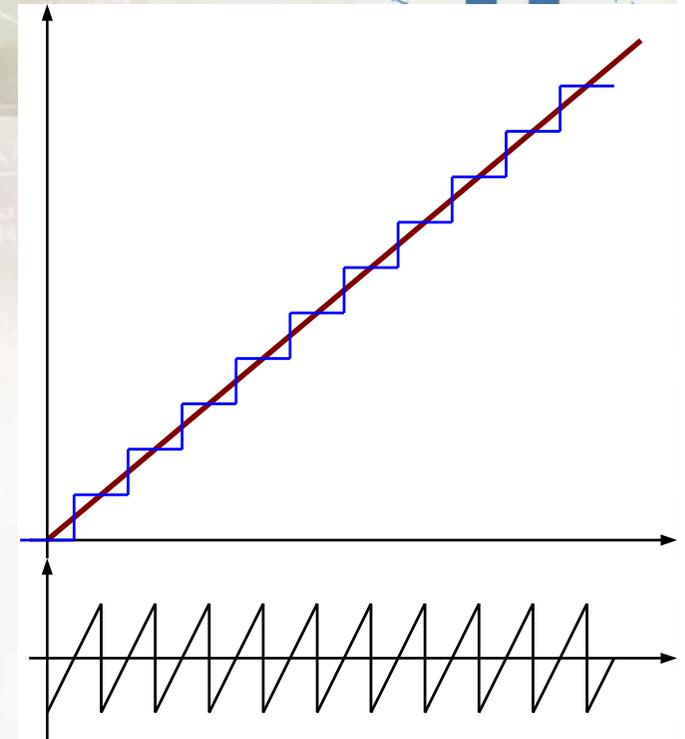
# The quantization process

Only distinct levels detected /represented

The error is the deviation from straight line

A range from 0 to  $V_{ref}$  gives the LSB step

$$\Delta = \frac{V_{ref}}{2^N}$$



The quantization error is bounded

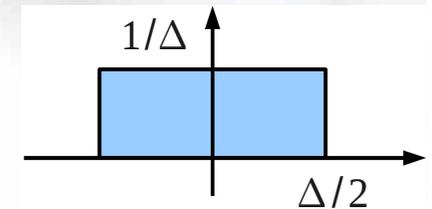
$$Q \in \left[ -\frac{\Delta}{2}, \frac{\Delta}{2} \right]$$

# Quantization process, cont'd

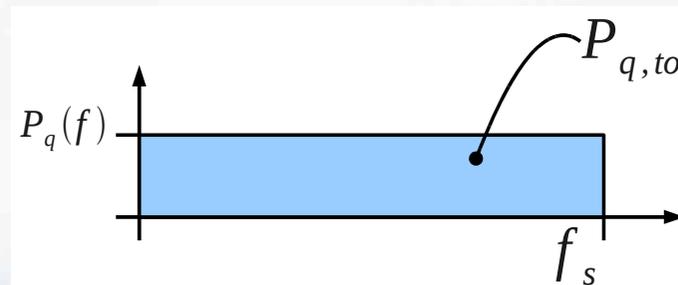
**Assume signal-independent (not true for low number of bits)**

Quantization assumed to be a stochastic process

White noise, uniformly distributed in  $[-\Delta/2, \Delta/2]$



**Noise power spectral density**



# Quantization process, cont'd

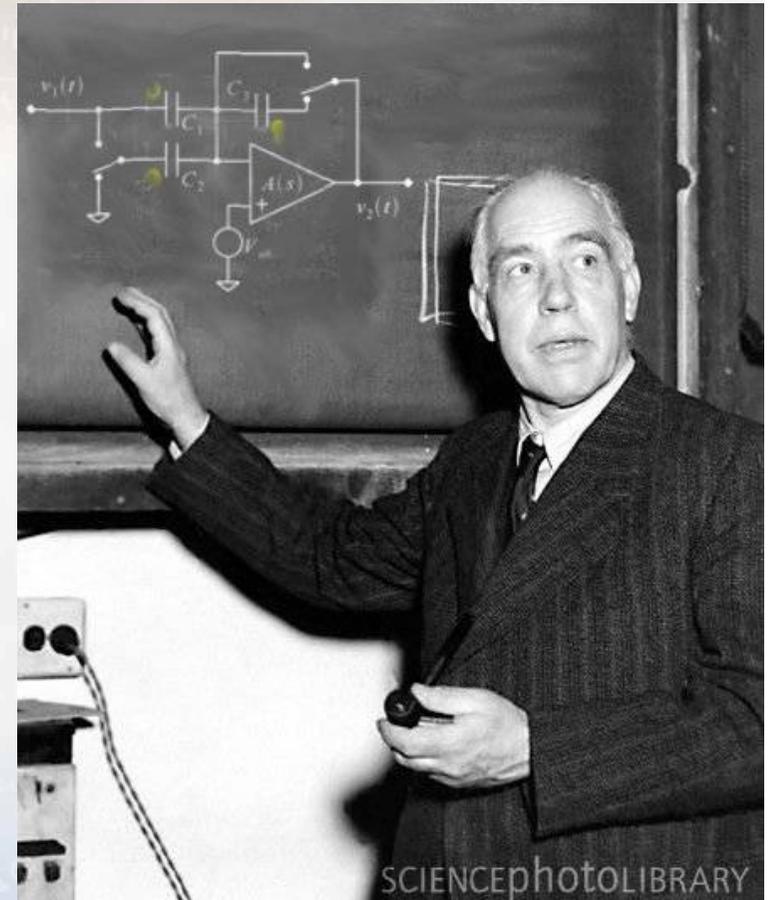
## Sigma of the probabilistic noise

## Noise model

Remember the superfunction

## Power spectral density

A certain bandwidth contains a certain amount of noise



# The quantization process, cont'd

Peak power assuming centered around the nominal DC level

$$P_{pk} = \left( \frac{V_{ref}}{2} \right)^2$$

Maximum, average sinusoidal power

$$P_{avg} = \frac{1}{2} \cdot \left( \frac{V_{ref}}{2} \right)^2 = \frac{1}{8} \cdot V_{ref}^2 = \frac{P_{pk}}{2}$$

Peak-to-average ratio (PAR) for a sinusoid

$$PAR = P_{pk} / P_{avg} = 2 \quad (1.76 \text{ dB})$$

# The quantization process, cont'd

Noise power given by the sigma:  $P_{q,tot} = \sigma^2 = \Delta^2 / 12$

Signal-to-quantization-noise ratio:  $SQNR = \frac{P_{avg}}{P_{q,tot}} = \frac{P_{pk}}{P_{q,tot} \cdot PAR}$

With values inserted

$$SQNR = \frac{V_{ref}^2 / 4}{\frac{1}{12} \cdot \left( \frac{V_{ref}}{2^N} \right)^2 \cdot PAR} = \frac{3 \cdot 2^{2N}}{PAR} \quad \text{or in a logarithmic scale:}$$

$SQNR \approx 6.02 \cdot N + 4.77 - PAR = 6.02 \cdot N + 1.76$  for our sinusoid.

# D/A conversion as such

**Amplitude is generated by scaling the digital bits and summing them**

$$A_{out}(nT) = \sum_{k=0}^{N-1} w_k(nT) \cdot 2^k$$

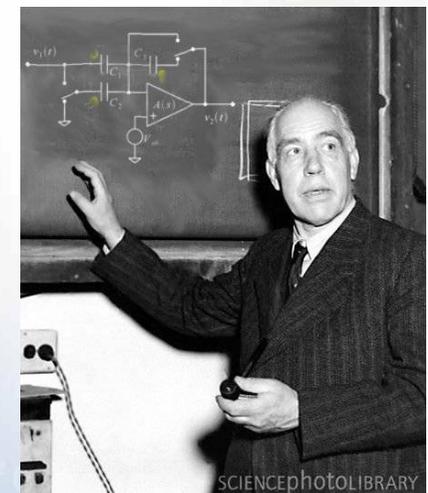
**The scaling does not necessarily have to be binary:**

Binary

Thermometer

Linear

Segmented



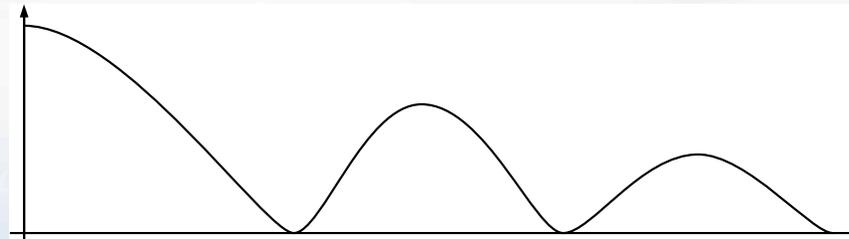
## D/A conversion, cont'd

The output is a pulse-amplitude modulated signal (PAM)

$$A_{out}(t) = \sum a(nT) \cdot p(t - nT)$$

$$A_{OUT}(j\omega) = A(e^{j\omega T}) \cdot P(j\omega)$$

A common pulse is the zero-order hold, since ideal reconstruction is impossible. In the frequency domain the output will be sinc-weighted:



A reconstruction filter is needed to compensate!

# D/A converter architectures

## Current-steering

Outputs summed by weighted current sources.

## Resistor-string

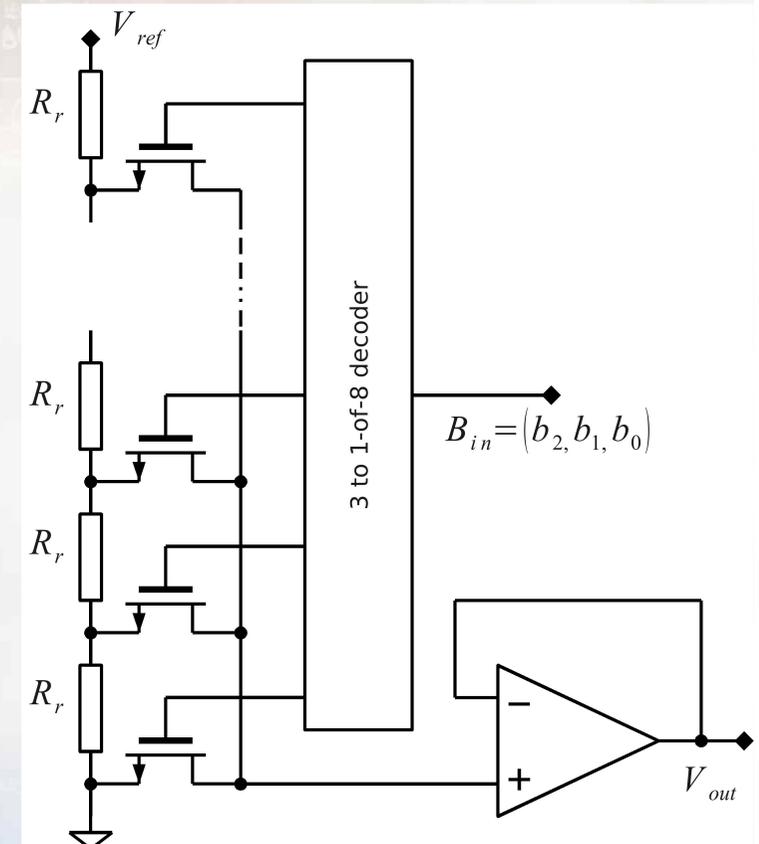
Select one tap out of many and buffer

## R-2R

Utilizes current dividers

## And many more

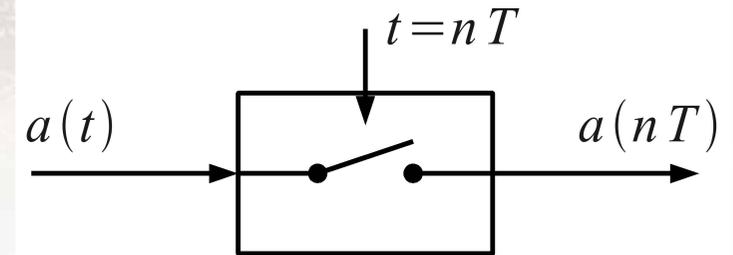
Oversampling DACs, etc.



# A/D conversion

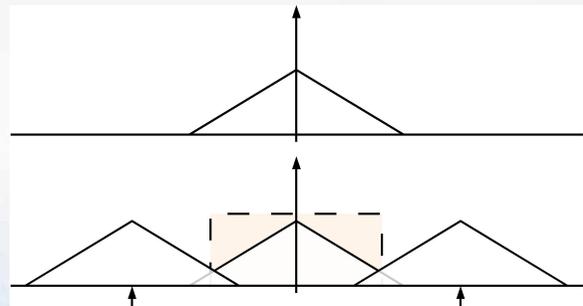
**A/D conversion is essentially a sampling process**

$$a(nT) = a(t)|_{t=nT}$$



**Poission's summation formula**

$$A(e^{j\omega T}) = \sum A(j(\omega - 2\pi k) \cdot T)$$



**Spectrum might repeat and overlap itself!**

# A/D conversion, cont'd

## To avoid folding:

meet the sampling theorem (theoretically minimizes error)

use an anti-aliasing filter (practically minimizes error)

**Practically, an amount of oversampling is required to meet the tough filter requirements**

## Analog input is mapped to a digital code

A range of the input mapped to a unique digital code

$$D(nT) = \sum_{k=0}^{N-1} w_k(nT) \cdot 2^k$$

# A/D converter architectures

## Flash

A set of comparators compares input signal with reference levels

## Sub-ranging

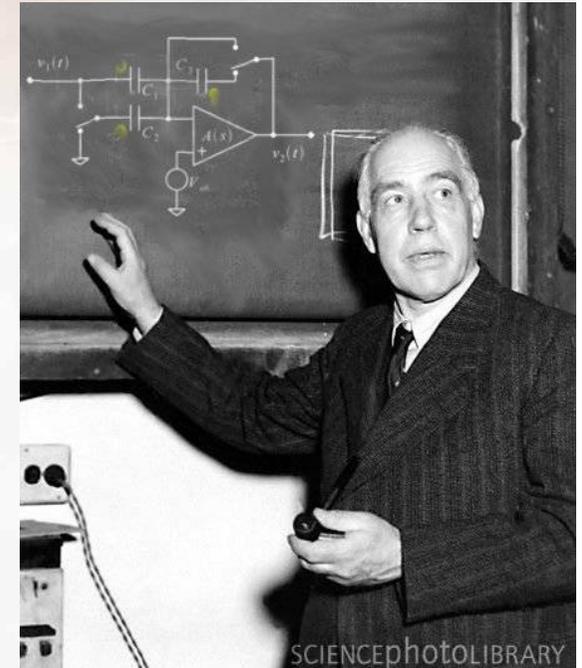
Use a coarse stage cascaded with a fine stage.

## Pipelined

A set of sub-ranging ADCs

## Successive approximation

One sub-ranging ADCs looping in time rather than a straight pipeline.



SCIENCEPHOTOLIBRARY

# Converter trade-offs, speed vs resolution

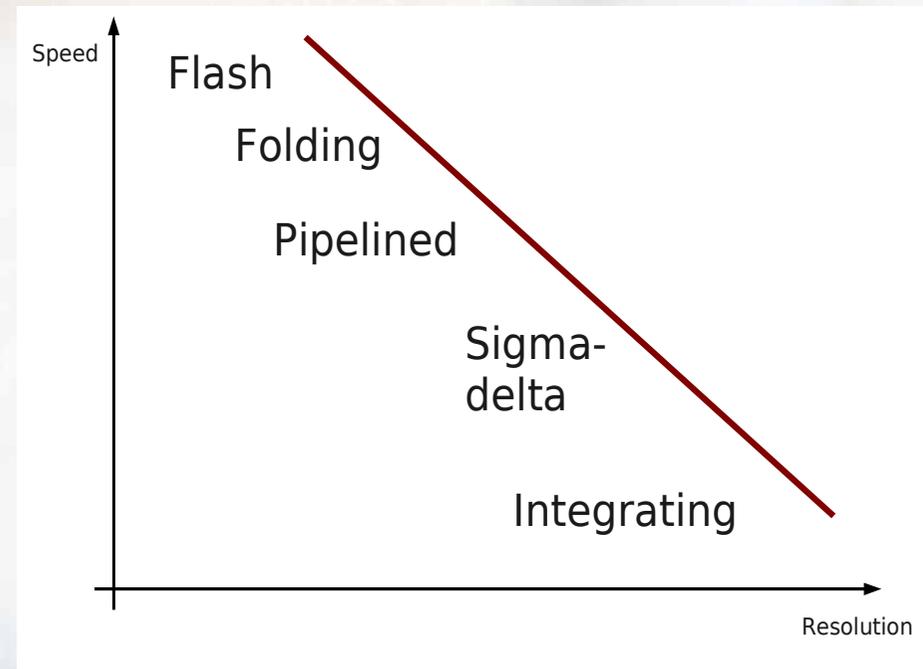
A common figure-of-merit:

$$\text{FOM} = \frac{4kT \cdot f_{bw} \cdot \text{DR}}{P}$$

Some conclusions from this formula

High-speed converters cost power

High-resolution converters cost area

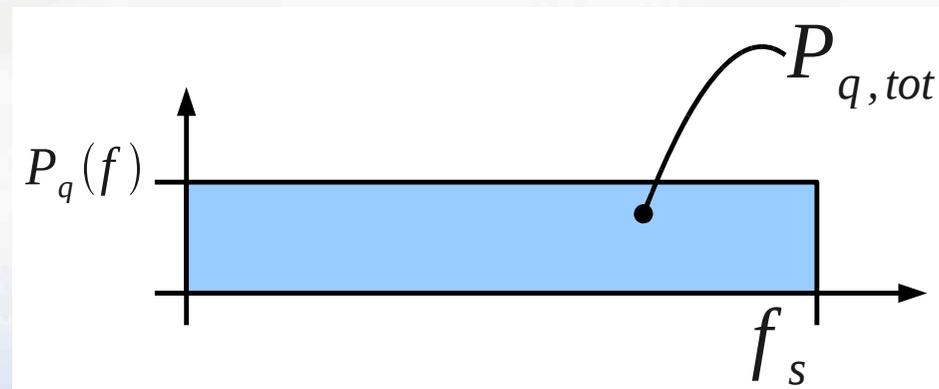


# Quantization noise revisited 1

Assume signal-independent (not true for low number of bits)

Assume white noise, uniformly in  $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$ , with  $\Delta = \frac{V_{ref}}{2^N}$

Noise power spectral density (PSD)



# Quantization noise revisited 2 (quiz ...)

Noise power given by the sigma:  $P_{q,tot} = \sigma^2 = \Delta^2/12$

Signal-to-quantization-noise ratio:  $SQNR = \frac{P_{avg}}{P_{q,tot}} = \frac{P_{pk}}{P_{q,tot} \cdot PAR}$

With values inserted

$$SQNR = \frac{V_{ref}^2/4}{\frac{1}{12} \cdot \left(\frac{V_{ref}}{2^N}\right)^2 \cdot PAR} = \frac{3 \cdot 2^{2N}}{PAR}$$

In logarithmic scale

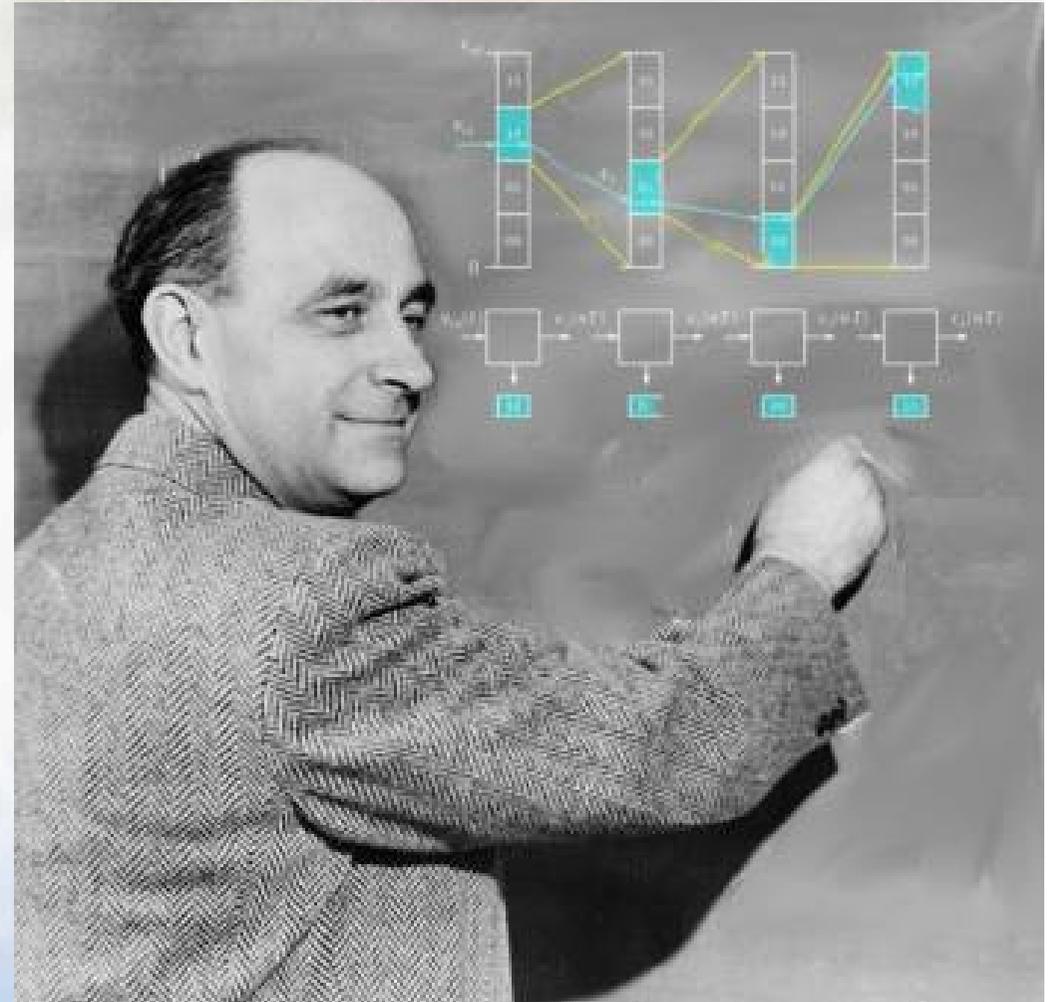
$SQNR \approx 6.02 \cdot N + 4.77 - PAR = 6.02 \cdot N + 1.76$  for our sinusoid.

# Oversampling 1

**Assume we have headroom to increase the sample frequency**

**Apply filtering to remove the excessive noise**

**We can effectively increase the performance! Or ... ?**



# Oversampling converters

Noise power over the entire Nyquist range

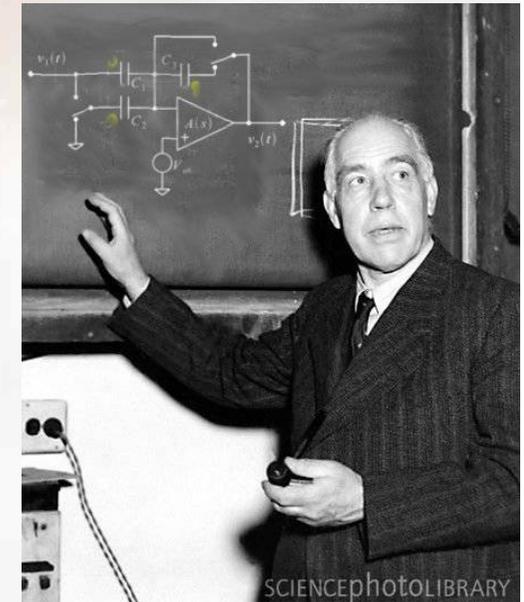
$$\text{SQNR} = 6.02 \cdot N + 1.76 \text{ [dB]}$$

Assume we oversample, or put it this way, we have a anti-aliasing/reconstruction filter there anyway. We get

$$\text{SQNR} = 6.02 \cdot N + 1.76 + 10 \cdot \log_{10} \frac{f_s}{2 \cdot f_{bw}}$$

where the oversampling ratio is  $\text{OSR} = \frac{f_s}{2 \cdot f_{bw}}$

"For each doubling of the sample frequency, we gain 3 dB"



# Oversampling converters

Assume we take a lower order converter to start with

$$\text{ENOB} = \frac{\text{SQNR} - 1.76}{6.02} = N + \frac{10 \cdot \log_{10} \text{OSR}}{6.02}$$

**A 16-bit resolution can be obtained using a 12-bit converter if we oversample 256 times.**

For some applications not an impossible scenario

**A 16-bit resolution can be obtained using a 1-bit converter if we oversample 1073741824 times.**

1 Hz would require 1 GHz of sampling frequency!

# Attacking the filtering problem

**Ideal reconstruction and sampling requires ideal filters**

**Increase your frequency range**

DAC: Interpolation and upsampling

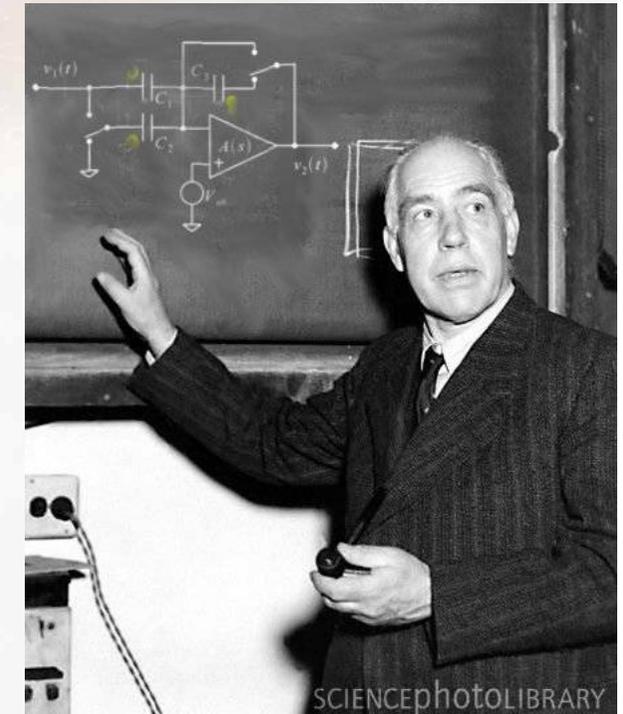
ADC: Decimation and downsampling

**Drawbacks**

Higher power consumption

More difficult to design

FOM limit



# Oversampling converters, cont'd

Since we are introducing another converter, and increasing the frequency - why not spice it a bit?

Create a converter that can also shape the new added noise

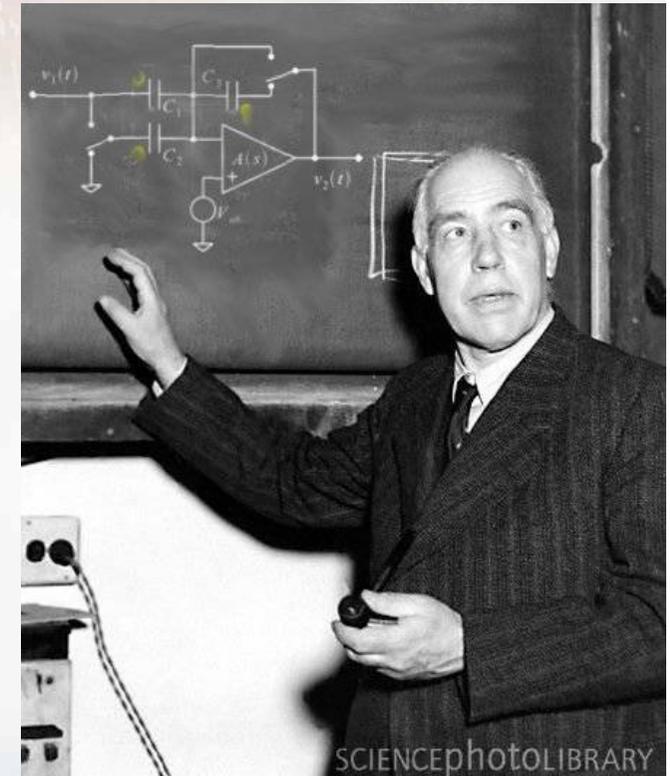
**This can be done through sigma-delta modulation**

High-pass filters the added noise

All-pass filters the signal

Designing a sigma-delta modulator is very much a filtering problem

**Notice that a DAC can never increase the number of bits!**



# Sigma-delta converters, cont'd

Consider the transfer function

$$Y = Q + A \cdot \underbrace{(X - B \cdot Y)}_{\epsilon} \Rightarrow Y = \frac{Q + A \cdot X}{1 + A \cdot B}$$

Noise and signal transfer functions:  $\text{NTF} = \frac{1}{1 + A \cdot B}$ ,  $\text{STF} = \frac{A}{1 + A \cdot B}$

Example, A is integrator and B is unity:  $\text{NTF} = 1 - z^{-1}$ ,  $\text{STF} = z^{-1}$

Order of the filters and oversampling determines the SQNR:

$$\text{SQNR} = 6.02 \cdot N + 1.76 + 10 \cdot (2 \cdot L + 1) \cdot \log_{10} \text{OSR} - 10 \cdot \log_{10} \frac{\pi^{2L}}{2L + 1}$$

# Sigma-delta converters, cont'd

## 1st-order modulator. 16-bit resolution can be obtained using:

12-bit converter if we oversample 16 times

1-bit converter if we oversample 1522 times (c.f. 1G-times before)

## Second-order modulator. 16-bit resolution ...

12-bit converter if we oversample 6 times.

1-bit converter if we oversample 116 times.

## Third-order modulator. 16-bit resolution ...

12-bit converter if we oversample 5 times.

1-bit converter if we oversample 40 times

**Momentum slightly lost and filtering problem recreated!**

# Sigma-delta, the audio example

## HIFI

16 bits, i.e., 100 dB

## Signal bandwidth

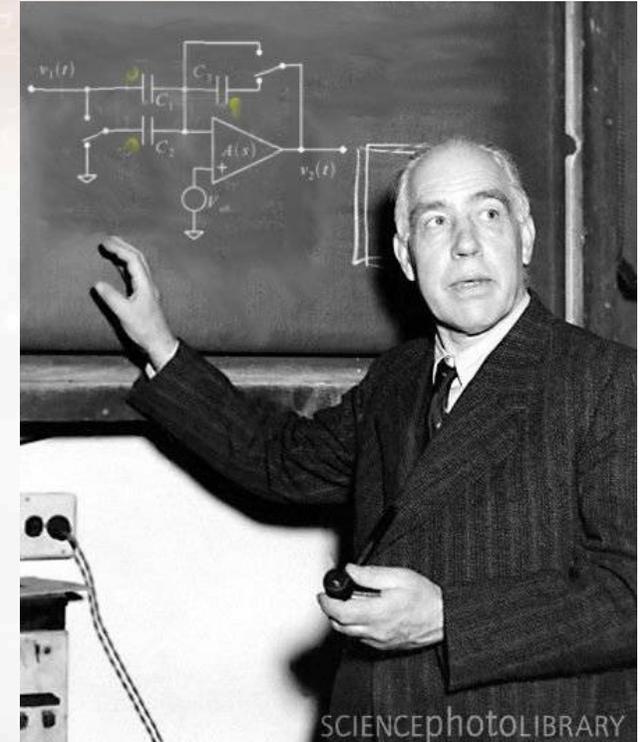
22 kHz

**Choose as few bits in coarse quantizer as possible**

**Choose minimum possible order**

**Choose minimum possible sample frequency**

What configurations are possible?



# Sigma-delta, the audio example

$$\text{SQNR} = 6.02 \cdot N + 1.76 + 10 \cdot (2 \cdot L + 1) \cdot \log_{10} \text{OSR} - 10 \cdot \log_{10} \frac{\pi^{2L}}{2L + 1}$$

gives us

```
>> antikAudioSigmaDelta
SNR = 100.2442 dB, L = 2, M = 1, OSR = 128 fs = 5.632 MHz.      *****
SNR = 115.2957 dB, L = 2, M = 1, OSR = 256 fs = 11.264 MHz.
SNR = 106.2642 dB, L = 2, M = 2, OSR = 128 fs = 5.632 MHz.
SNR = 112.2842 dB, L = 2, M = 3, OSR = 128 fs = 5.632 MHz.
SNR = 103.2527 dB, L = 2, M = 4, OSR = 64 fs = 2.816 MHz.
SNR = 112.8346 dB, L = 3, M = 1, OSR = 64 fs = 2.816 MHz.
SNR = 103.8025 dB, L = 3, M = 3, OSR = 32 fs = 1.408 MHz.
SNR = 109.8225 dB, L = 3, M = 4, OSR = 32 fs = 1.408 MHz.
```

# What did we do today?

## Filters

A quick glance on analog filters leading to supply filters.

## Data converter fundamentals

Quantization noise

Signal-to-noise ratio

Overview, since literature is more useful explaining many, many different architectures

DACs have no delay



# What will we do next time?

**Wrap-up data converters**

**Signal distribution**

**Timing and clocking**



# Lecture 7, Timing, clocking

Differential signaling schemes  
Timing and clocking



# What did we do last time?

## Filters

## Data converter fundamentals