

UiO : Department of Informatics
University of Oslo

IN5220

Summary

Jørgen A. Michaelsen and Dag T. Wisland

Spring 2026

Outline

- Practical information
- Curriculum overview

Examination

Oral exam 30 min per candidate

Date: June 8th

More details on schedule will come from Studadm

All printed and written material allowed

Final grade: Character grade A – F

Oral exam (60 %)

Project (40 %)

Course content

From the course webpage:

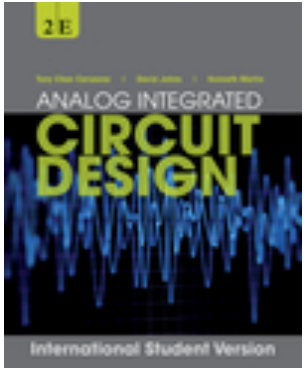
"The course provides the know-how and skills needed to design advanced mixed-signal CMOS integrated circuits using modern EDA tools. Key focus will be on circuits and systems design involving both digital- and analog/RF components. The starting point will be to introduce sampled systems and how the sampling process affects signals. Switched-capacitor design technique and circuit examples will be covered in detail. In addition, data converter methods/circuits and phase-locked loops are key aspects. A central part of the course will be a design project where students will implement a mixed-signal system in CMOS using modern design tools."

Learning outcomes

The main learning outcome is the ability to understand and design advanced mixed-signal integrated circuit. In particular you will be able to:

- Understand the concept of sampling and how the sampling process affects the involved signals
- Implement filters and other circuit functions using switched-capacitor design methodology
- Understand and design data converters, both analog-to-digital and digital-to-analog converters including oversampling and noise-shaping systems
- Understand and design phase-locked-loops (PLL) and its key building blocks
- Design and implement mixed-signal circuits in CMOS from schematic to layout using state-of-the-art design tools and methods

Curriculum



Chapter 2.2, 2.3, 2.4
Chapter 11.1, 11.2
Chapter 13
Chapter 14.1, 14.2, 14.3, 14.4, 14.5, 14.6, 14.7, 14.8
Chapter 15.1, 15.2, 15.3, 15.5
Chapter 16
Chapter 17.1, 17.2, 17.3, 17.5, 17.6, 17.7, 17.9
Chapter 18.1, 18.2 (Except: 18.2.6, 18.2.7), 18.3, 18.5, 18.7, 18.8
Chapter 19.1, 19.2, 19.3,19.5

DEVICE MISMATCH

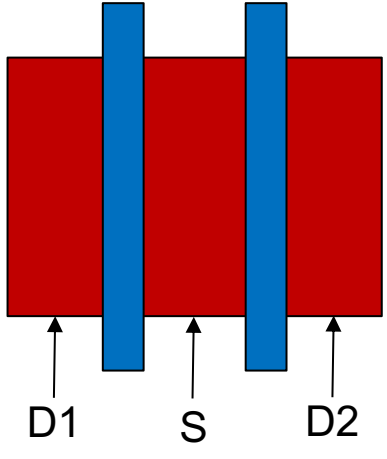
Device Mismatch

- The physical manufacturing process gives rise to imperfections in the manufactured devices
  - “Identical devices” (i.e. identically drawn devices) comes out slightly different on silicon
  - Systematic differences vs random differences
  - Local variations vs global variations

Device Mismatch

Mismatch is random mismatch between two identically drawn devices, as close as possible in the layout, usually sharing source. E.g.

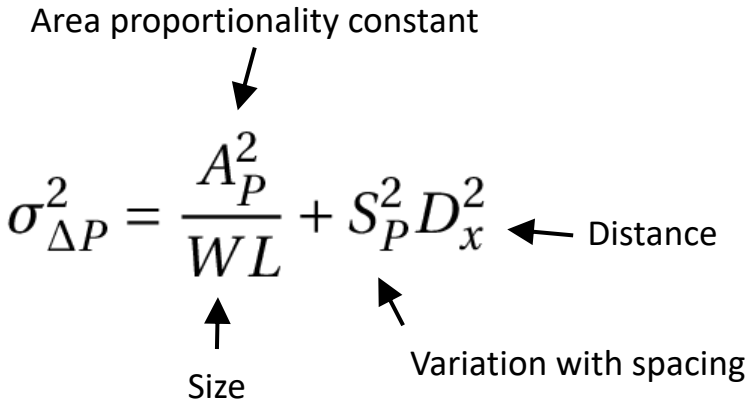
- Differential pair input devices
- Current mirrors



Hand calculation of matching

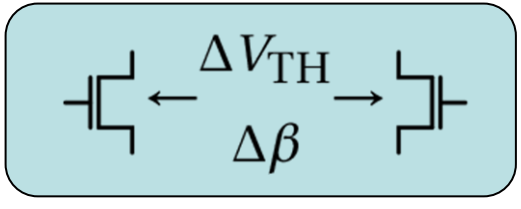
Matching of parameter, P, between two identically drawn devices

Parameters of devices closely spaced exhibit a random variance **inversely proportional to area**.



Hand calculation of matching

Mismatch between *two* identically drawn adjacent transistors. We use the previous formula to find ΔV_{th} and $\frac{\Delta\beta}{\beta}$. Then we use these results to find $\frac{\Delta I_D}{I_D}$, V_{OS}, etc.



$$I_D = \frac{\mu_n C_{\mathrm{ox}}}{2}\frac{W}{L}\left(V_{\mathrm{GS}} - V_{\mathrm{TH}}\right)^2 = \frac{\beta}{2}\left(V_{\mathrm{GS}} - V_{\mathrm{TH}}\right)^2$$

LAYOUT

Layout

As circuit designers we must carefully consider how to draw layout for critical/sensitive parts of the circuit to get **robust** and **predictable** performance. Simulation results from schematic is not the final answer (but necessary and a good indication).

Lithography system

Pattern on wafer is distorted by imperfections in the lithography system.

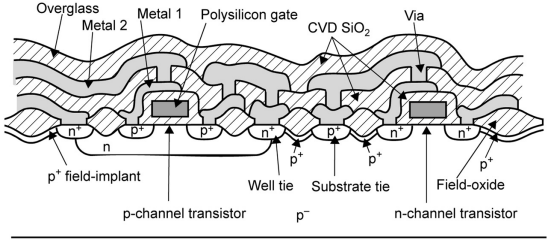
Wavelength of light is a limitation for feature size.

Depth of focus (DOF)

As the wafer is built layer by layer the geometry becomes uneven.

Wafer is planarized between each step with chemical mechanical polishing (CMP)

inherent tradeoff between DOF and resolution (better NA, finer pitch, more narrow DOF)



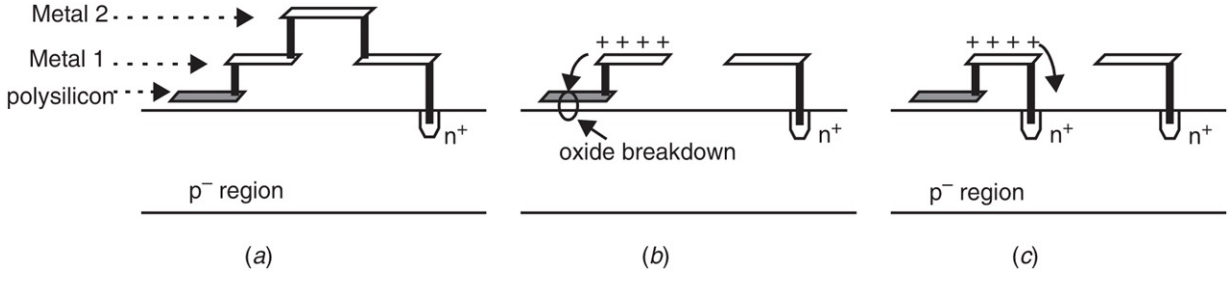
Design rules

The layout we draw is not perfectly reproduced on the wafer.

We must comply with a set of rules to ensure that the layout we draw is manufacturable. This is a minimum requirement, and does *not* guarantee robust and predictable performance. *Necessary but not sufficient.*

Antenna design rules

Large metal area connected to a MOSFET gate collects ions during manufacturing and potentially breaks down the gate oxide (irreversibly).

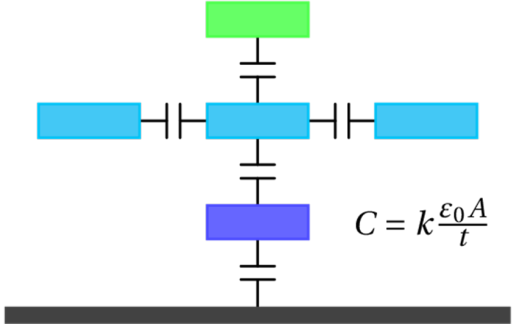


Interconnect parasitics

Overlap capacitance

Low-κ dielectric helps reduce interconnect capacitance.

Fringe capacitance also important.



Interconnect parasitics

Consider resistance and capacitance (and inductance).

Typical metal resistance is 0.1 Ω/□

10 Ω per via (never use one single via)

Current handling capability (electromigration)

Process documentation lists actual values

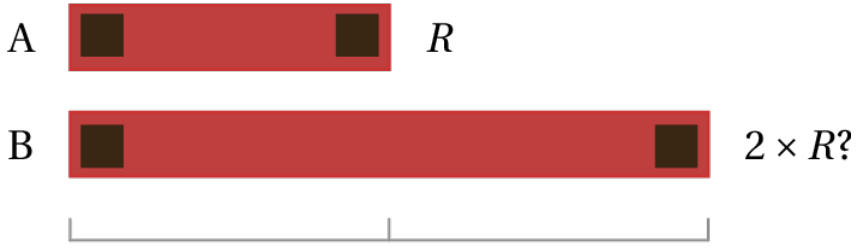
Interconnect parasitics

Sizing metal lines is a trade-off between capacitance and resistance (and current handling capability)

Wide lines, fewer squares, less resistance, but potentially more overlap capacitance.

Resistance and capacitance vary due to dishing and erosion.

Unit elements

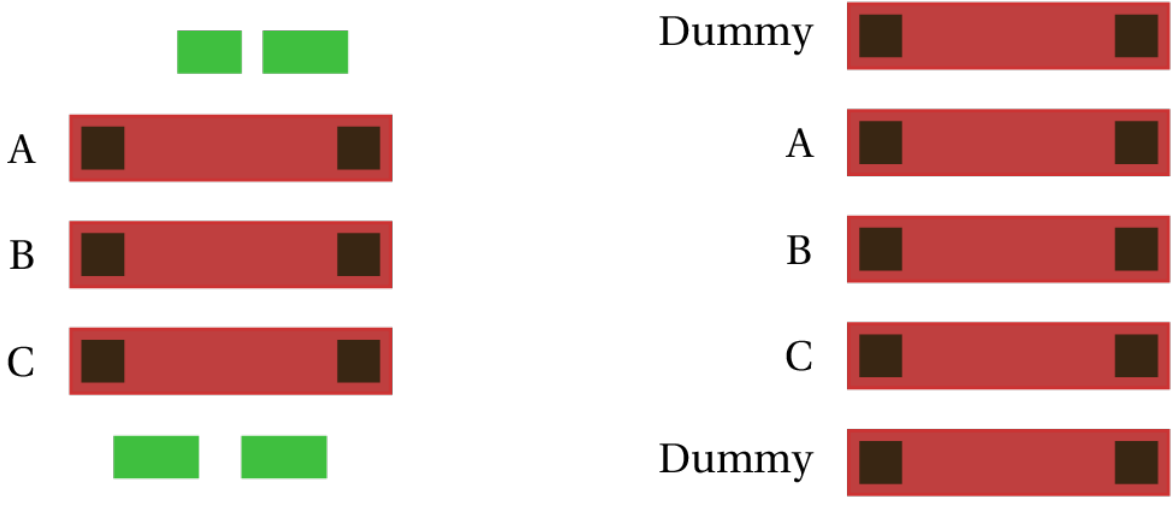


Instead, make *identical* unit elements.



Dummy elements

Make sure matched elements see the *exact* same surroundings



Interdigitated layout



Process gradient almost evenly distributed between components A and B.

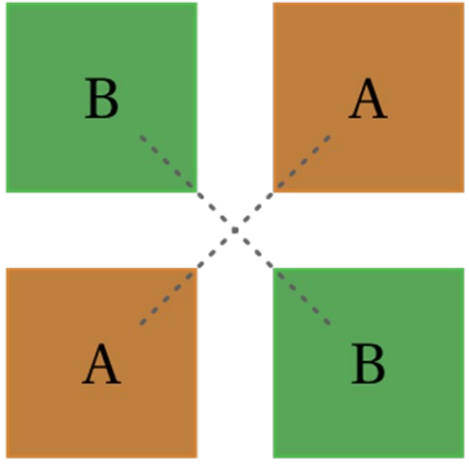
A and B are split into units

Common centroid

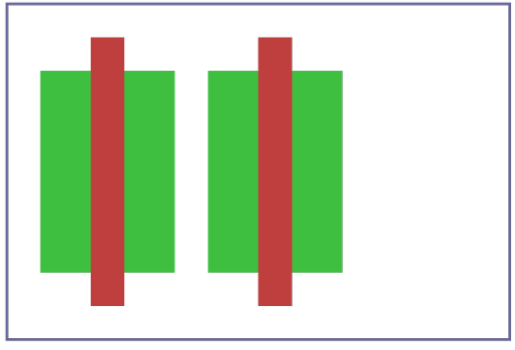
Perfect cancelation of linear gradients.

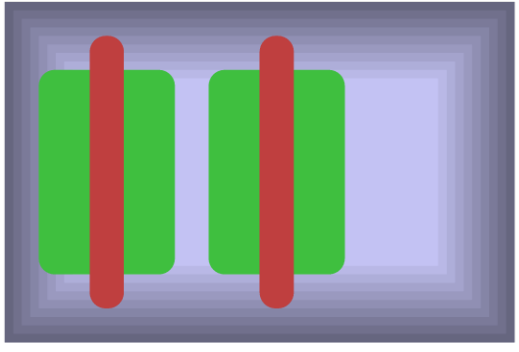
Several patterns are possible.

Can split A and B into more than two units.



Well proximity effect (WPE)

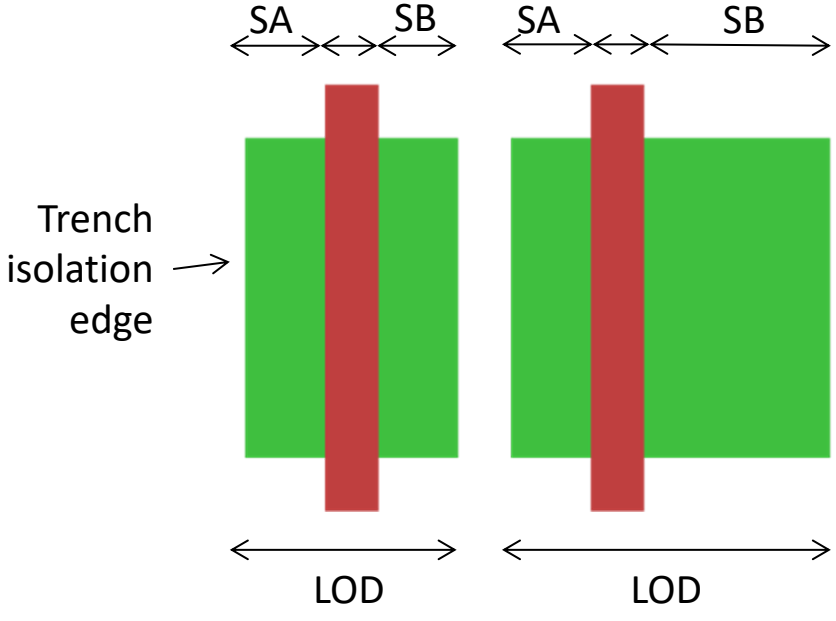




As with S/D asymmetry, implantation angle may render the scattering and doping asymmetric.

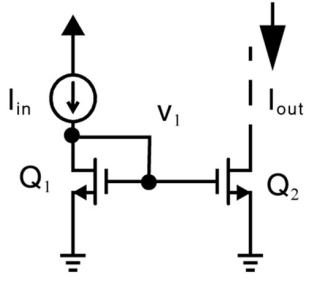
Affects threshold voltage, body effect, and mobility.

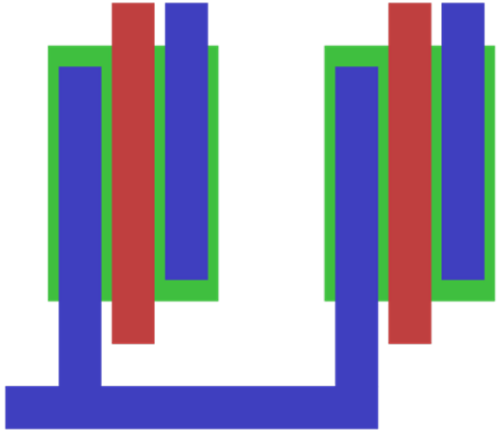
STI stress (LOD)



Transistor interconnect

Unbalanced metal routing: Transistors see different source voltage.



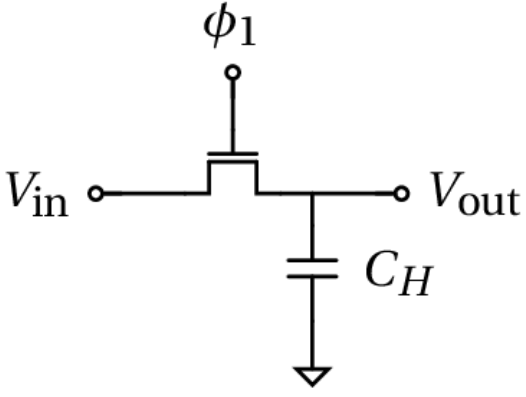


Distribute references as *currents*, not bias voltage.

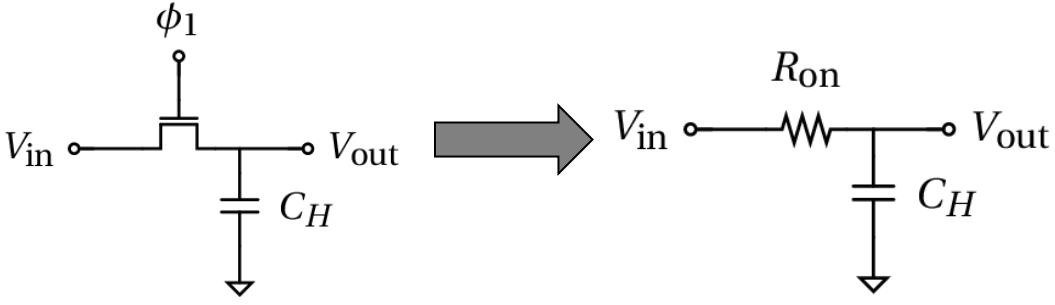
SAMPLE AND HOLD, AND DISCRETE-TIME SIGNALS

Basic circuit

- Sample and hold vs. track and hold

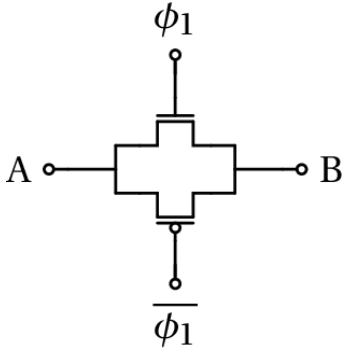


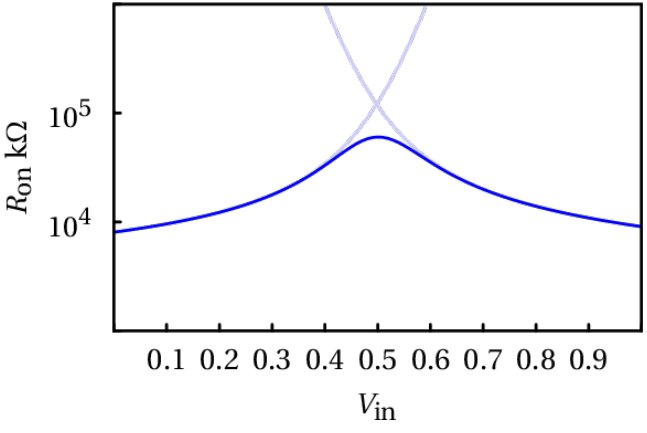
Finite speed



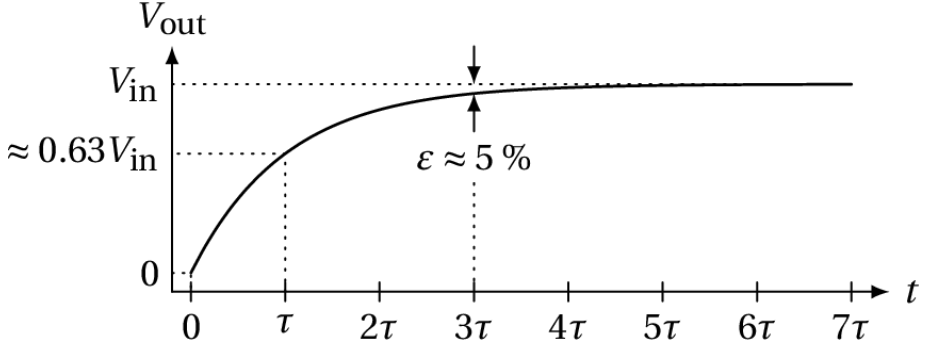
Finite speed

$$R = \frac{1}{\beta(V_{GS} - V_{tn})}$$





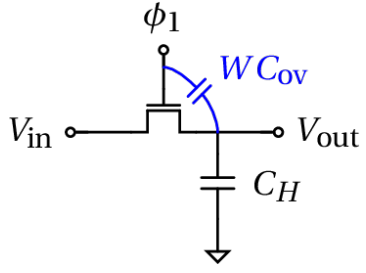
Settling time



$$t_s = \tau \ln \frac{1}{\varepsilon}$$

Clock feed-through

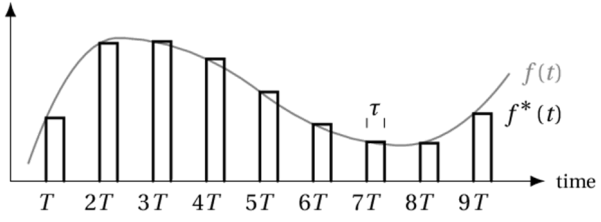
- Overlap capacitance
- $\Delta V = -\frac{C_{\mathrm{ox}} W L_{\mathrm{ov}} V_{\mathrm{dd}}}{C_H}$
- Constant, depends on clock voltage only

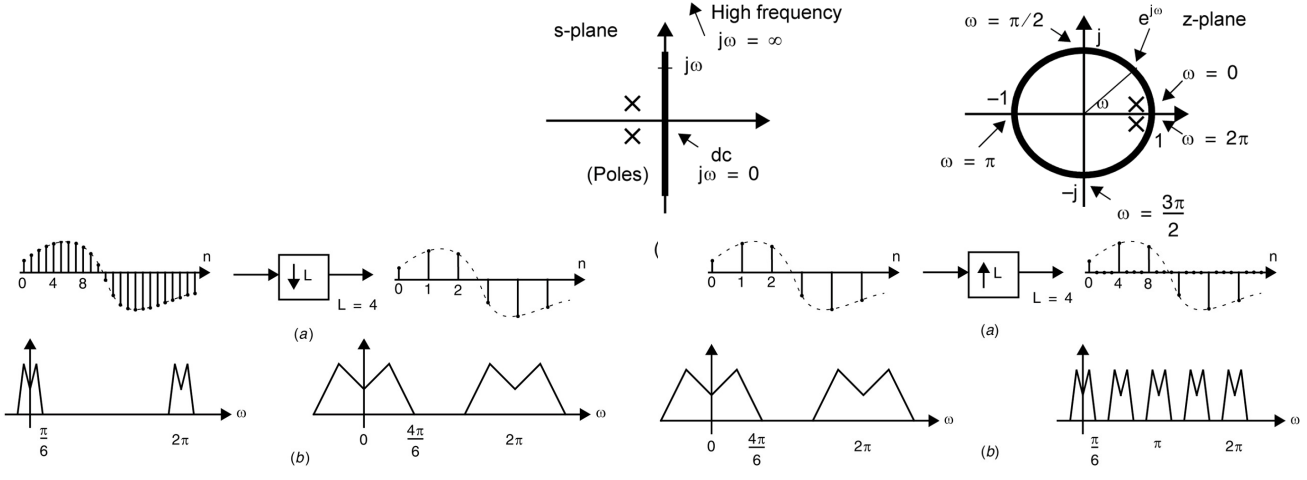


Discrete time (Ch 13)

Important to understand how sampling affects the signal.

z-transform to analyze sampled systems.





The z-transform

$$z \equiv e^{sT}$$

$$X(z) \equiv \sum_{n=0}^{\infty} x(nT) \cdot z^{-n} = \sum_{n=0}^{\infty} x[n] \cdot z^{-n}$$

- Delay by k samples, $z^{-k} \cdot X(z)$
- Convolution in time $\leftrightarrow$ multiplication in z

Frequency response

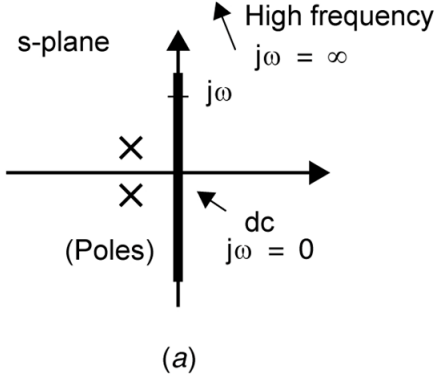
- Use the Laplace domain description of the sampled signal
- As before, substitute $s = j\omega$

$$X(j\omega) = \sum_{n=-\infty}^{\infty} x(nT) \cdot e^{-j\omega nT}$$

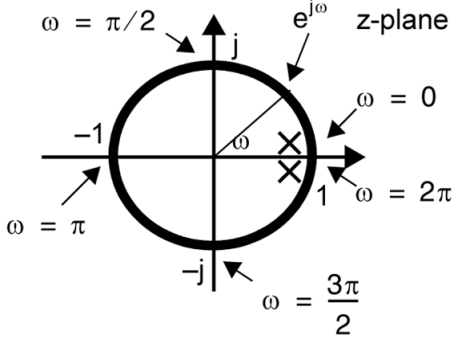
- $X(j\omega)$ is the Fourier transform of the impulse sampled input signal, $x(t)$.
- e^{jx} is cyclic, $e^{jx} = \cos x + j \sin x$

Frequency response

- We go from the z-transform to the frequency response by substituting $z = e^{j\omega T}$
- As we sweep ω we trace out the unit circle



(*a*)



(*b*)

Frequency response

Rewriting to use frequency (Hz), rather than radian frequency

$$X(f) = \sum_{n=-\infty}^{\infty} x(nT) \cdot e^{-j2\pi fnT}$$

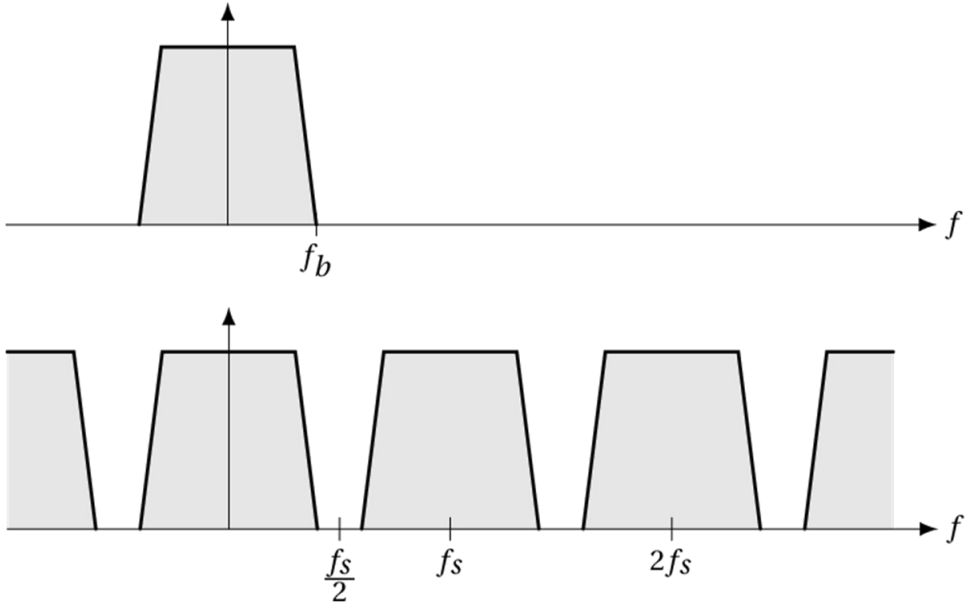
Because e^{jx} is cyclic, $f_1 = k \cdot f_s + f_1$, where f_1 is an arbitrary frequency, k is any integer and f_s is the sampling frequency (T^{-1}).

Frequency response

The frequency spectrum repeats. We can only uniquely represent frequencies from DC to $\frac{f_s}{2}$ (the Nyquist frequency).

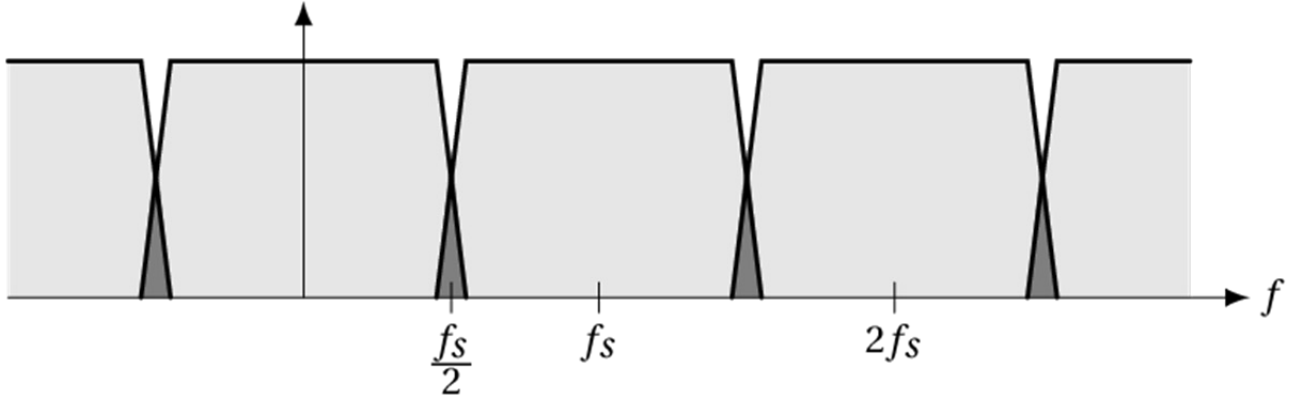
Important practical consequence: We must band limit the signal before sampling to avoid *aliasing*. A non-linear distortion.

Frequency response



Frequency response

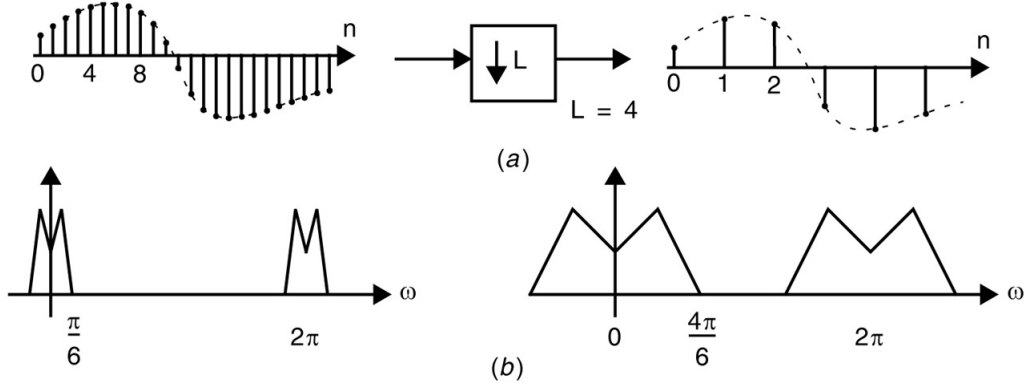
If the signal contains frequencies beyond $\frac{f_s}{2}$, sampling results in in aliasing. Images of the signal interfere.



Sampling rate conversion

- Changing the sampling rate after sampling
- We come back to this when discussing oversampled converters
- Oversampling = sampling faster than the Nyquist frequency would indicate
- Upsampling is increasing the sampling rate (number of samples per unit of time)
- Downsampling is decreasing the sampling rate

Downsampling

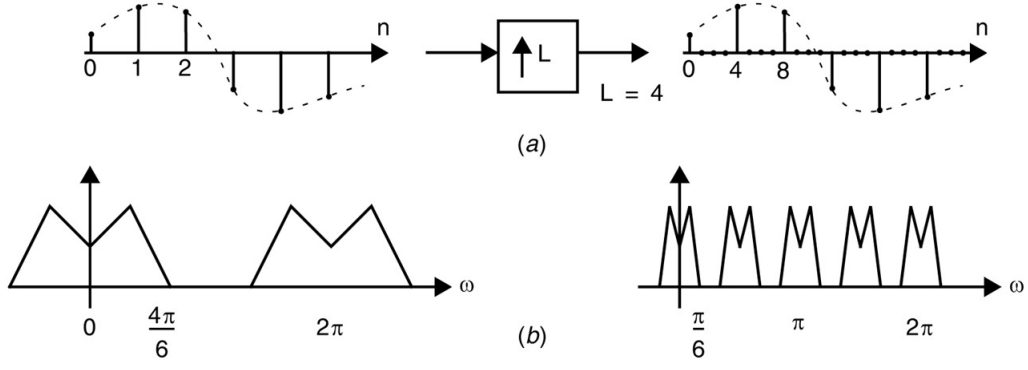


Keep every n-th sample.

Downsample too much: Aliasing

Upsampling

Insert *n* zero valued samples between each original sample, and low-pass filter. Requires gain to maintain the signal level.



Discrete time filters

Analog filters use integrators, s^{-1}, as building blocks to implement filter functions. Discrete time filters use delay, z^{-1}.

Example:

Time domain: $y[n+1] = bx[n] + ay[n]$

z-domain: $zY(z) = bX(z) + aY(z)$

$$H(z) \equiv \frac{Y(z)}{X(z)} = \frac{b}{z-a}$$

Discrete time filters

Frequency response, $z = e^{j\omega}$ (ω normalized to the sampling frequency, really $z = e^{j\omega T}$)

$$H\left(e^{j\omega}\right) = \frac{b}{e^{j\omega} - a}$$

DC is $z = e^{j0} = 1$.

The sampling frequency is $z = e^{j2\pi} = 1$ (also).

Sufficient to evaluate the frequency response from 0 to π due to symmetry (for real signals).

Stability

$$y[n+1] = bx[n] + ay[n]$$

If $a > 1$, the output grows without bounds. Not stable.

$$H(z) = \frac{b}{z - a}$$

In a stable system, all poles are inside the unit circle

- $a = 1$: Discrete time integrator
- $a = -1$: Oscillator

IIR filters

$y[n+1] = bx[n] + ay[n]$ is an infinite impulse response filter.

Single impulse input ($x[0] = 1, 0\ \text{otherwise}$) results in an output that decays towards zero, but (in theory) never reaches zero. If we try to characterize the filter by its impulse response, we need an infinite number of outputs to characterize it.

FIR filters

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

Is a FIR (finite impulse response filter).

$$H(z) = \frac{1}{3}\sum_{i=0}^{2} z^{-i}$$

FIR filters are inherently stable but require higher order (more delay elements) than IIR.

Bilinear transform

- Mapping between continuous and discrete time
- Design the filter as a continuous time transfer function and map it to the *z*-domain

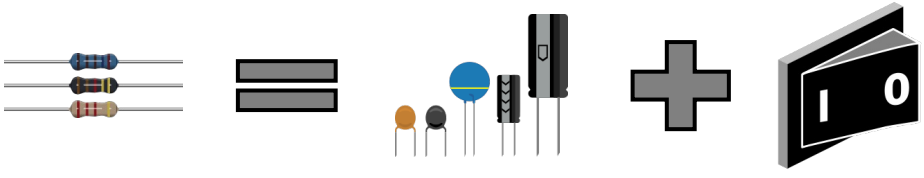
$$s = \frac{z-1}{z+1}, \text{ conversely, } z = \frac{1+s}{1-s}$$

- $s = 0$ maps to $z = 1$ (DC)
- $s = \infty$ maps to $z = -1$ $\left(\frac{f_s}{2}\right)$
- First order approximation

SWITCHED CAPACITOR CIRCUITS

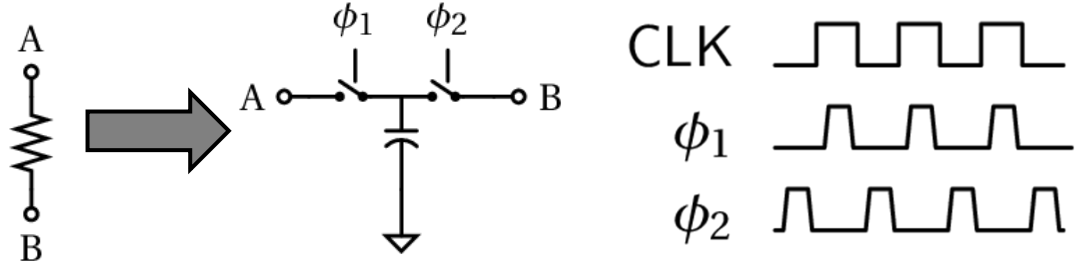
Switched capacitor circuits

Discrete time analog signal processing



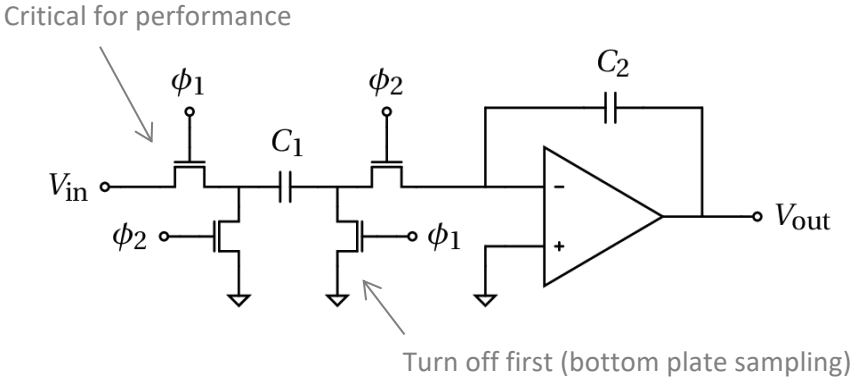
Why?

Introduction



The arrangement of switches and the capacitor approximates a resistor.

Parasitic insensitive integrators



Again, we analyze the charge transfer from one clock phase to the next to find the transfer function.

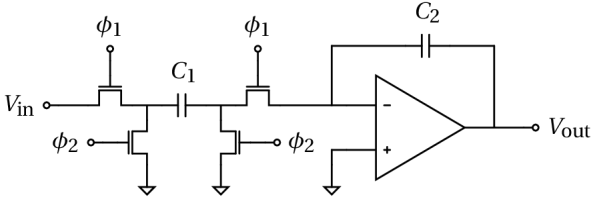
Delay free integrator

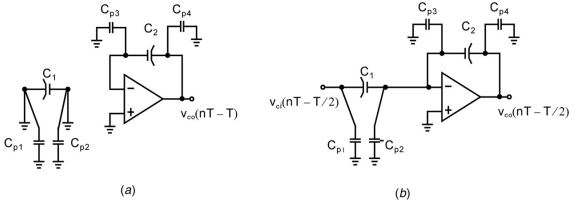
Same circuit as before, but modified clocking of the switches.

$$C_2 v_o(nT) = C_2 v_o\left(nT - \frac{T}{2}\right) - C_1 v_i(nT)$$
$$= C_2 v_o(nT - T) - C_1 v_i(nT)$$

$$v_o[n] = v_o[n-1] - \frac{C_1}{C_2} v_i[n]$$

$$H(z) = \frac{V_o(z)}{V_i(z)} = -\frac{C_1}{C_2}\frac{1}{1 - z^{-1}}$$
$$= -\frac{C_1}{C_2}\frac{z}{z - 1}$$



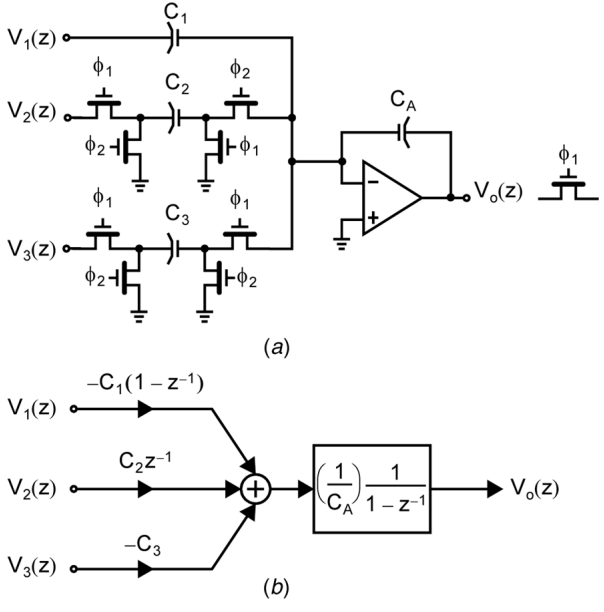


Signal flow graph analysis

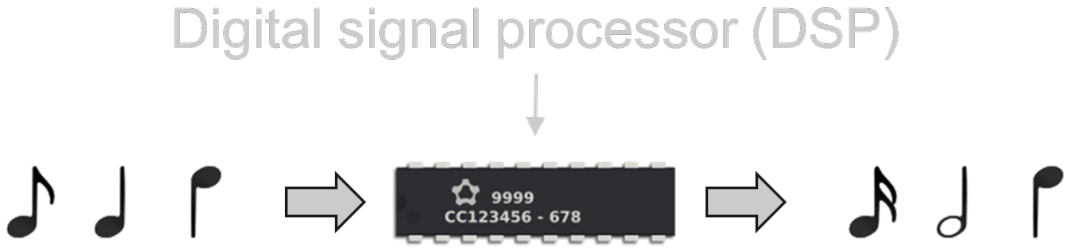
$$H_1(z) = \frac{V_o(z)}{V_1(z)} = -\frac{C_1}{C_A}$$
(inverting gain stage)

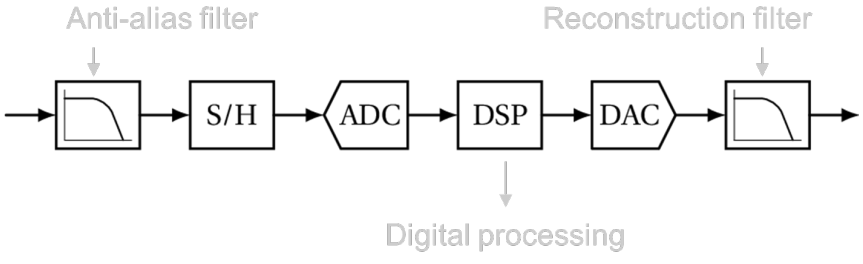
$H_2(z)$ is the non-inverting delaying integrator

$H_3(z)$ is the inverting delay-free integrator



Data converters (Ch 15–18)





DATA CONVERTER FUNDAMENTALS

Least significant bit

We want each bit to represent a fixed (constant) voltage

$$V_{LSB} \equiv \frac{V_{ref}}{2^N} \quad [\text{Volt}]$$

We relate many performance metrics to the unitless least significant bit (LSB)

$$1\ \text{LSB} = \frac{1}{2^N}$$

Analog to digital conversion

Ideally, the analog input has infinite precision

- Limited by noise
- Distortion sets a practical limit

The output has a finite number of information carrying units (bits)

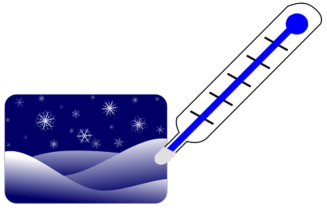
The ADC **quantizes** the input voltage to a finite number of bits

Quantization

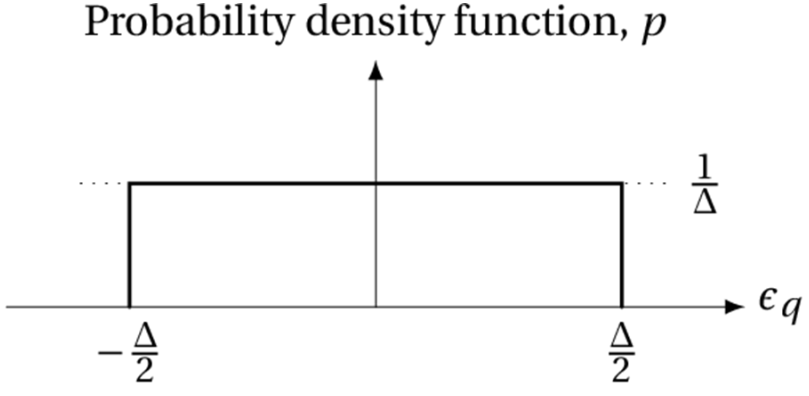
Data converters must represent continuous values in a range using a set of discrete values. A binary code is used to represent the value.

Information is lost!

- Hot or cold
- Freezing, cold, warm, or hot
- $-20\ °C, -19.5\ °C, \ldots, 20\ °C$



Quantization noise



$$P_q = \int_{-\infty}^{\infty} \epsilon_q^2 \, p(\epsilon_q) \, d\epsilon_q = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{\epsilon_q^2}{\Delta} \, d\epsilon_q = \frac{\Delta^2}{12}$$

Quantization noise

Assuming sine wave input

$$P_s = \frac{1}{T}\int_0^T \frac{x_{\text{fs}}^2}{4}\sin^2(2\pi f t)\,dt = \frac{x_{\text{fs}}^2}{8} = \frac{\Delta^2 2^{2N}}{8}$$

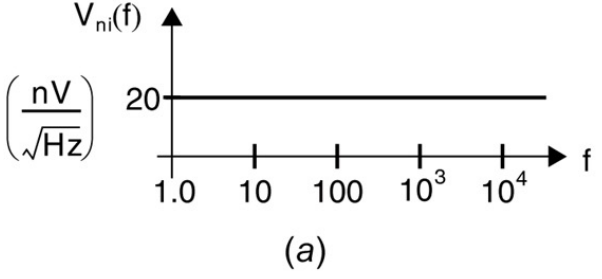
SNR due to quantization noise

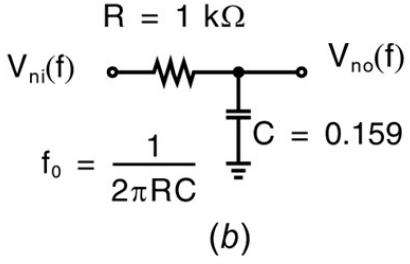
$$\text{SNR}_q = \frac{P_s}{P_q} = \frac{3\cdot 2^{2N}}{2} \approx 6.02N + 1.76\,\text{dB}$$

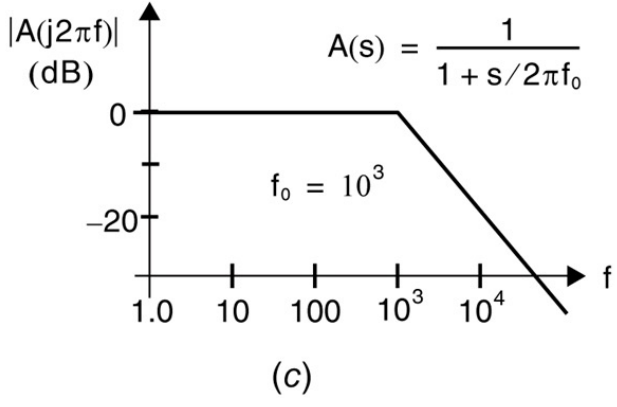
kT/C Noise (I)

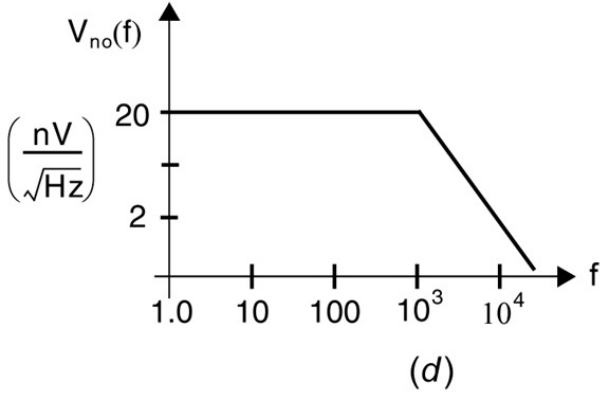
- SNR drives power consumption
- Signal headroom is reduced when the supply voltage is reduced
- Supply voltage is reduced because of scaling (beneficial for digital, less power consumption)
- Noise is constant

kT/C Noise (II)


$V_{ni}(f)$
$\left(\frac{nV}{\sqrt{Hz}}\right)$
20
1.0
10
100
10^3
10^4
f
(a)


R = 1 kΩ
$V_{ni}(f)$
$V_{no}(f)$
C = 0.159
$f_0 = \frac{1}{2\pi RC}$
(b)


$|A(j2\pi f)|$
(dB)
$A(s) = \frac{1}{1 + s/2\pi f_0}$
0
−20
$f_0 = 10^3$
1.0
10
100
10^3
10^4
f
(c)


$V_{no}(f)$
$\left(\frac{nV}{\sqrt{Hz}}\right)$
20
2
1.0
10
100
10^3
10^4
f
(d)

Signal to noise ratio

$$\text{SNR} \equiv 10 \log_{10} \frac{\text{signal power}}{\text{noise power}} \text{ [dB]}$$

Example:

Best case signal swing, $V_A = \frac{V_{DD}}{2}$, noise is $\frac{kT}{C}$.

$$\text{SNR} = \frac{CV_{DD}^2}{8kT}$$

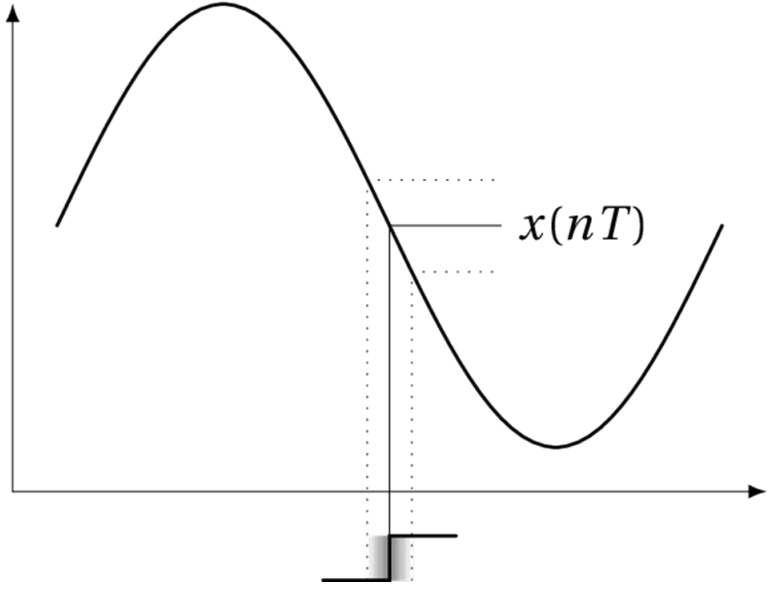
SNR vs power

- Increasing C to improve SNR has a power penalty
- If we need one more bit of resolution
  - This is the same as ~ 6 dB increase in the SNR
  - $10\times \log_{10} \frac{CV_{DD}^2}{8kT} \rightarrow 6.02\ \mathrm{dB}$
  - I.e. we need $4\times C$
- With higher C, to maintain $\mathrm{UGF} = \frac{g_m}{C_L}$ we also need $4\times g_m$
  - **Power increases $4\times$** because of $\frac{2I_D}{g_m}$

Sampling jitter

Sampling time uncertainty translates to an error in the input voltage.

Need a low noise sampling clock to get high accuracy.

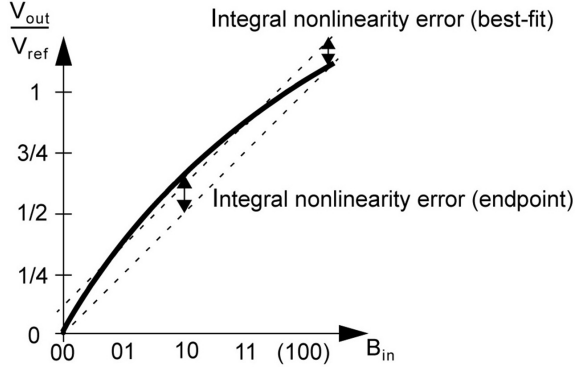


Integral non-linearity error

INL measures deviation from a straight line (best fit straight line or based on end points), after correcting for offset and gain error.

Result usually given in LSBs.

Static, meaning measured from DC inputs.



Differential non-linearity error

DNL is a measure of the step size error. Ideally the distance between two codes are exactly 1 LSB (after correcting for gain and offset).

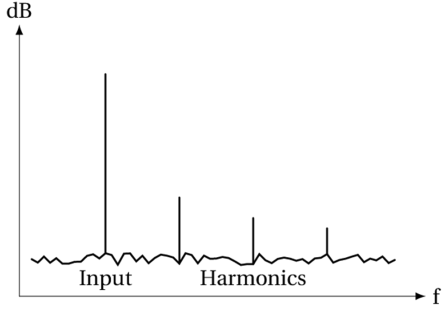
Like INL, DNL is measured at DC (no dynamic influence).

Both INL and DNL are common measures of data converter accuracy.

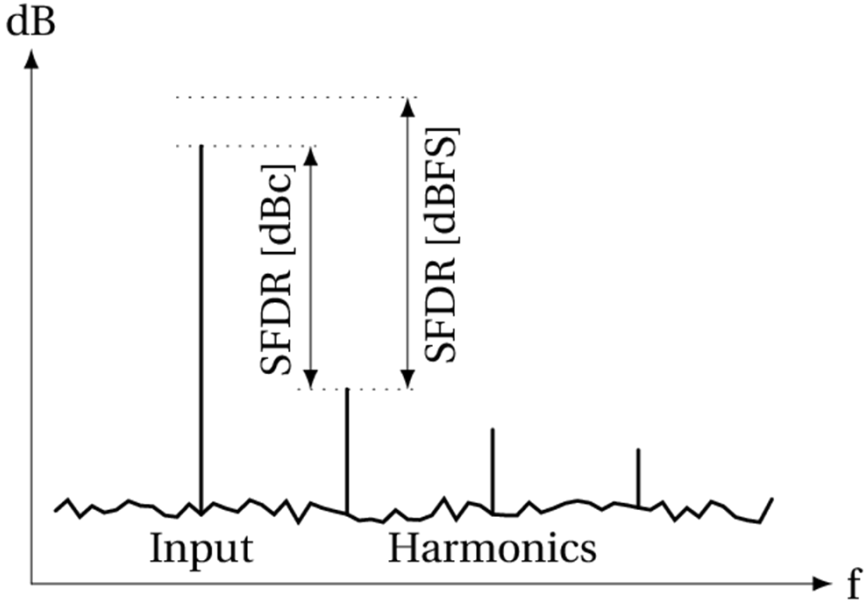
Dynamic specifications

Measure the performance when the input is a sine wave. Look at the converter output spectrum.

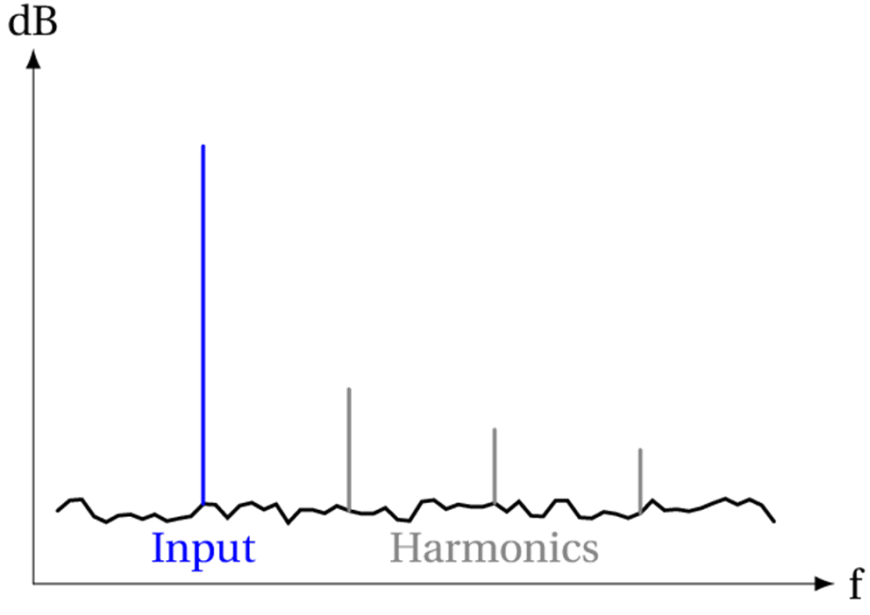
- SNR
- SINAD (SNDR)
- SFDR
- THD
- DR



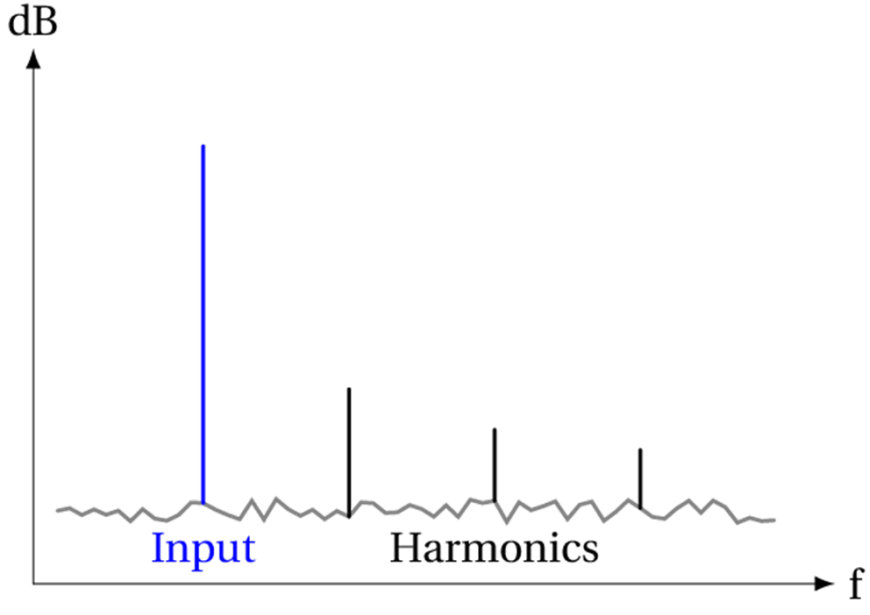
SFDR



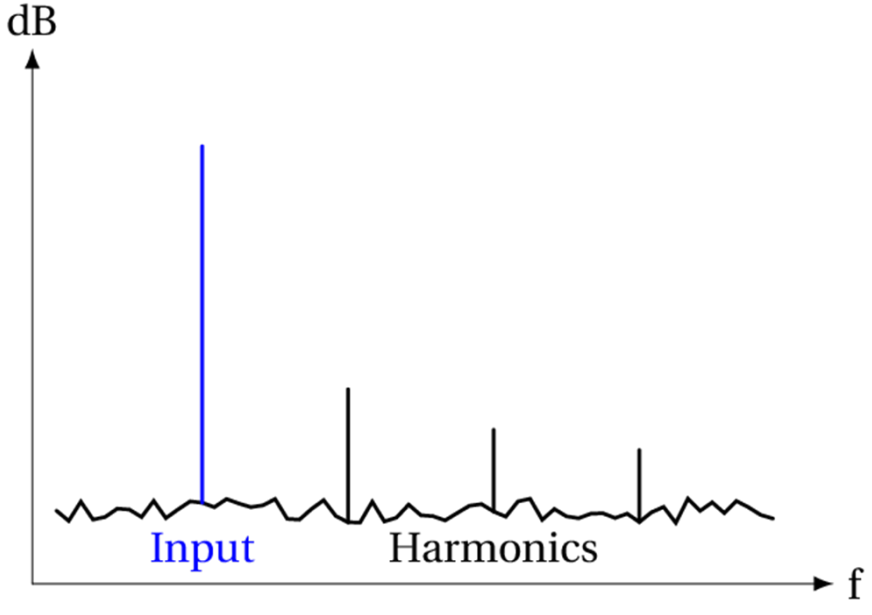
SNR



THD



SINAD



Dynamic specifications

Other dynamic specifications

- Intermodulation distortion
- Glitching
- …

Effective number of bits

Data converter resolution is the number of “physical” bits in the system. The effective number of bits (ENOB) tells us how many of these bits contains useful information. Relate the data converter performance to a converter limited only by quantization noise

$$\mathrm{ENOB} = \frac{\mathrm{SINAD} - 1.76\ \mathrm{dB}}{6.02\ \mathrm{dB}}$$

Figure of merit

Try to come up with an equation that tells us how good the data converter is. Several possibilities. Non are perfect. One commonly used figure of merit (FoM) is

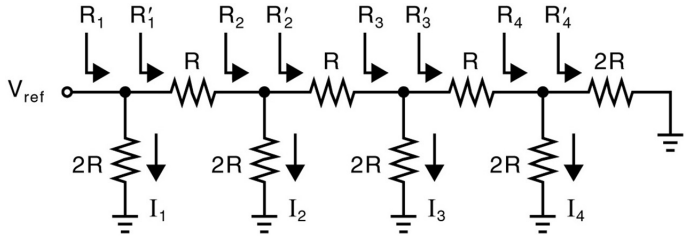
$$\mathrm{FoM} = \frac{P}{2^{ENOB} f_N}$$

Not necessarily a fair comparison, but tells us that if we want one bit improvement, we should expect the power consumption to double.

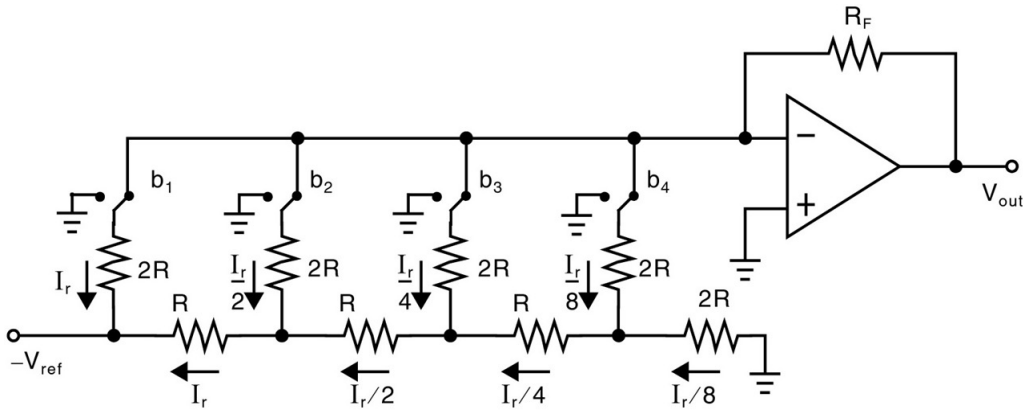
DIGITAL TO ANALOG CONVERTERS

R-2R based DAC

Popular choice for resistive DACs.

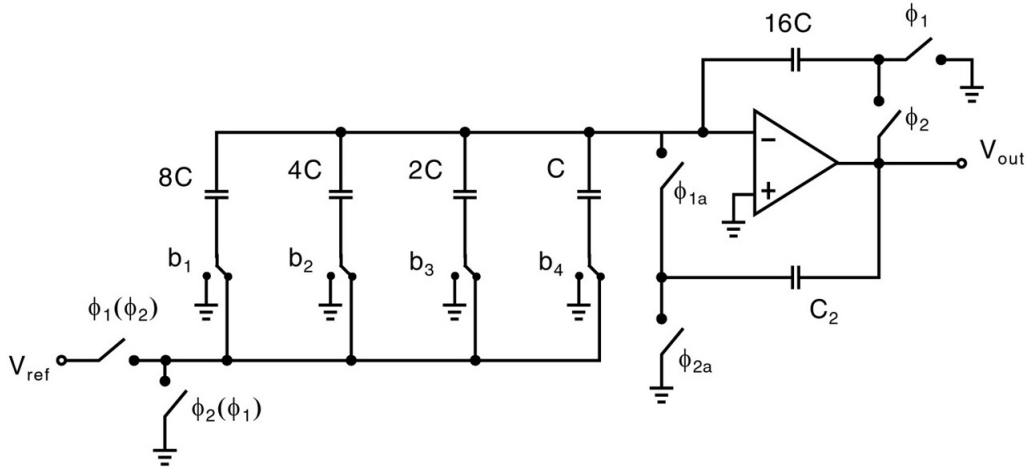


Low spread in component values: R and 2R. Only $2N$ resistors.



Charge redistribution DAC

Can be viewed as a SC gain circuit, amplifying a fixed reference voltage, where the gain is programmed by selecting capacitors.

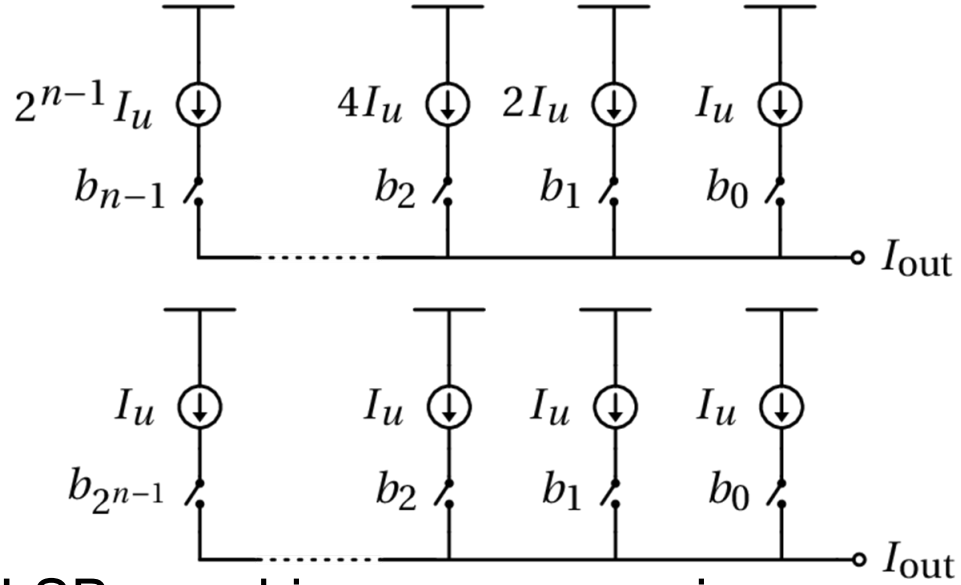


Segmented current steering DAC

Unit element
CS DAC,
good DNL,
but 2^N units

Binary weighted,
fewer elements,
but worse DNL

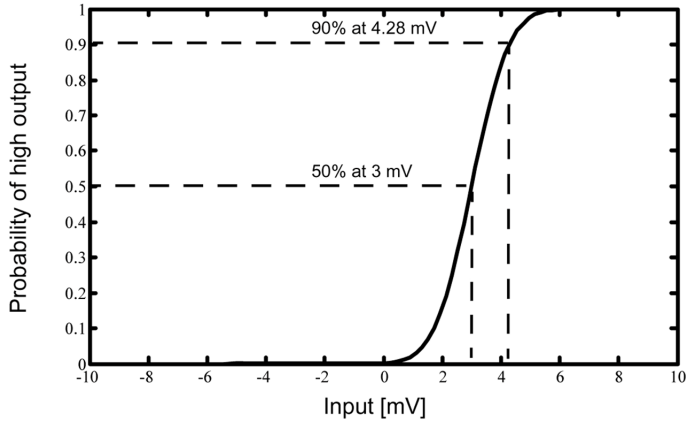
Segmented: MSBs
are unit elements and LSBs are binary, compromize.



ANALOG TO DIGITAL CONVERTERS

Comparator performance

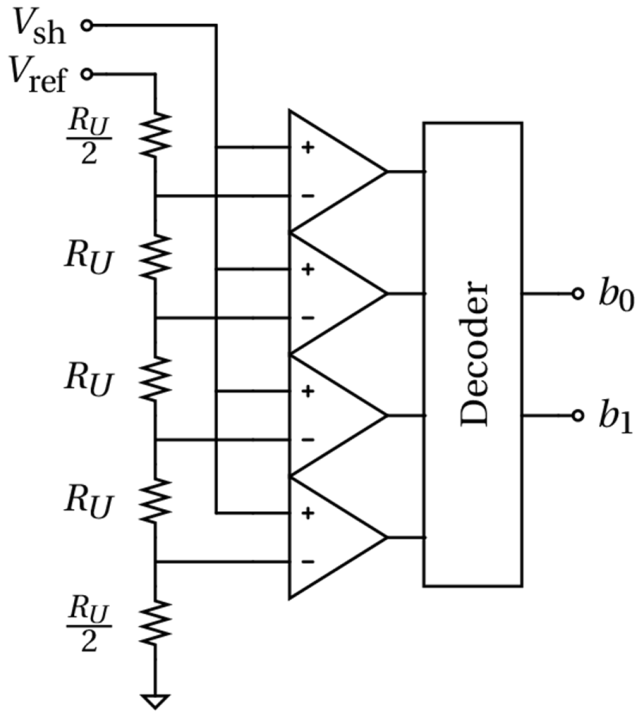
Non-ideal behavior limits performance.
→ Not a one-to-one mapping between input voltage and output, but a probability.



- Offset (mismatch)
- Noise
- Hysteresis (‘0’ → ‘0’ and ‘1’ → ‘1’ more likely)

Flash ADC

- The Kelvin divider is used to generate 2^N reference voltages
- Comparators compare the sample and held inputs to each derived reference voltage
- Thermometer output decoded to binary

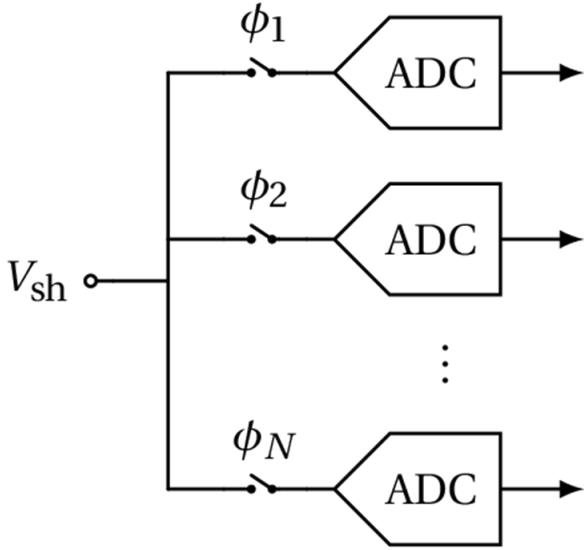


Time-interleaved ADC

Technique to increase the sampling rate by interleaving.

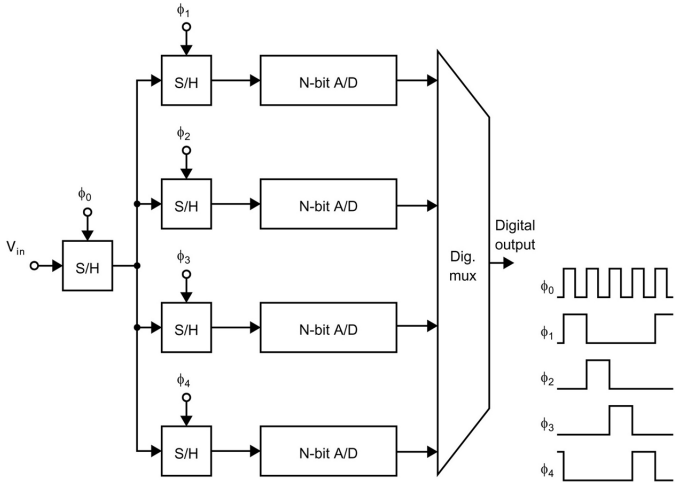
Run N ADCs in parallel.
Requires fast S/H

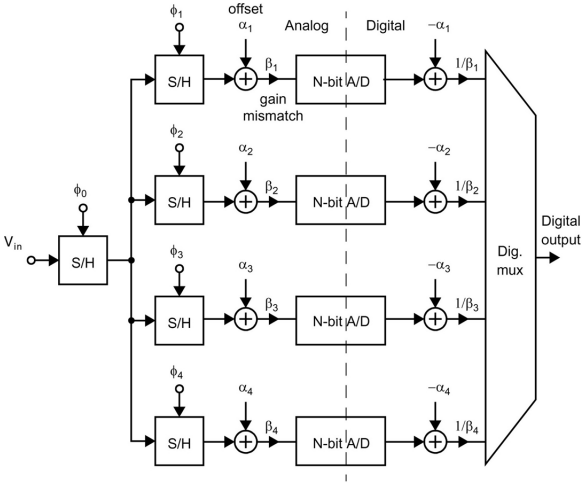
Offset and gain mismatch between channels and clock misalignment



Time-interleaved ADC

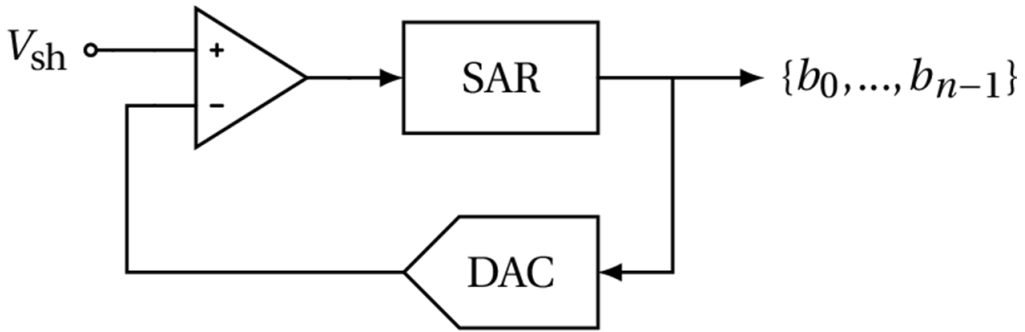
Digital correction of gain and offset





SAR ADC

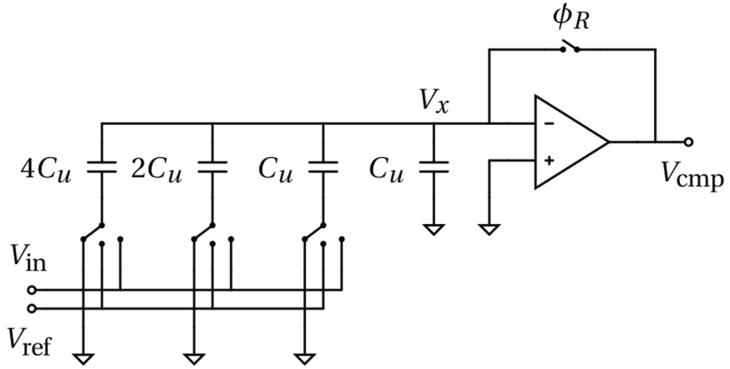
The successive approximation register (SAR) tests each bit sequentially (MSB first, one clock period per bit), and decides whether too keep the bit or not based on the comparator's output.



Charge redistribution SAR ADC

SAR ADCs are the most energy efficient ADCs (in terms of FoM).

Limited in resolution by capacitor matching (~ 10 bits).



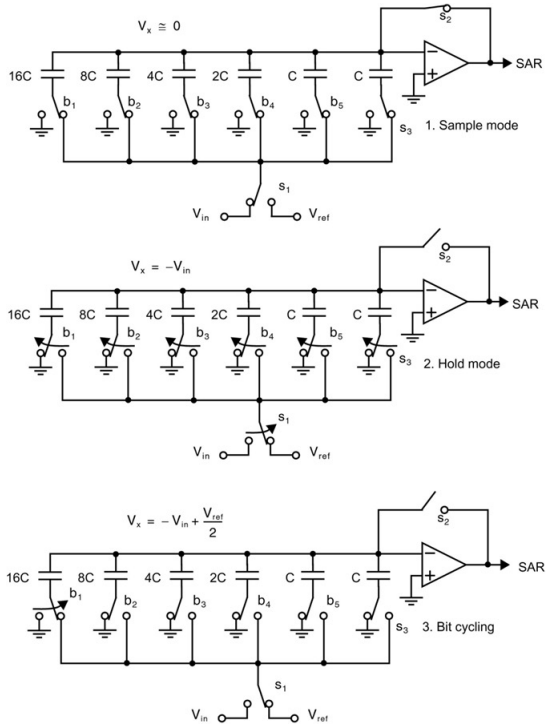
1. Sample mode
2. Hold mode
3. Bit cycling

- Inherent S/H function.
- Auto zero of the comparator is optional

Charge redistribution SAR ADC

No amplifiers needed (except for the comparator preamp). Saves power.

Comparator and charging of the capacitive array determines power consumption.

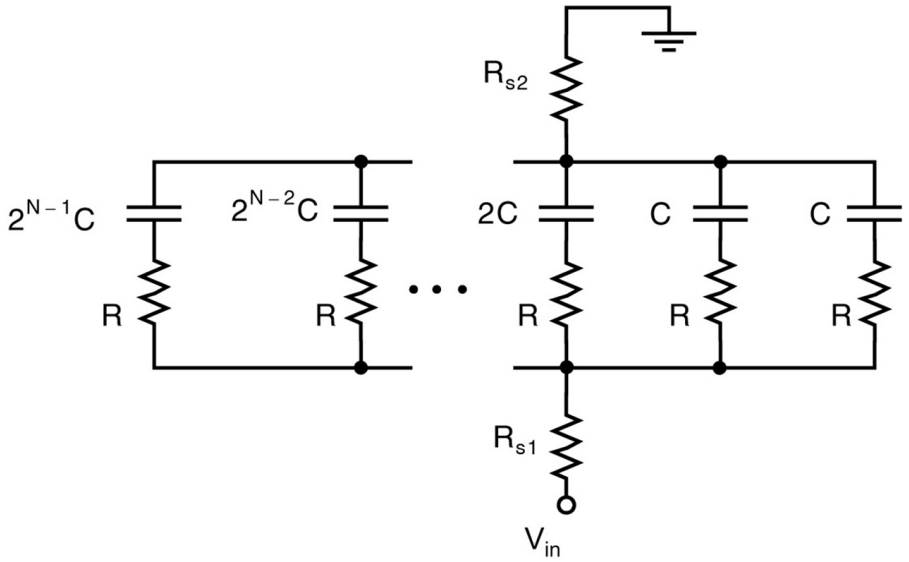


Charge redistribution SAR ADC speed

Speed limitation due to the RC time constant of the capacitor array and switches. $T > 0.69(N+1)\tau_{eq}$

τ_{eq} estimated from zero value time constant.

Charging time, T, assuming better than $^1/_2$ LSB.



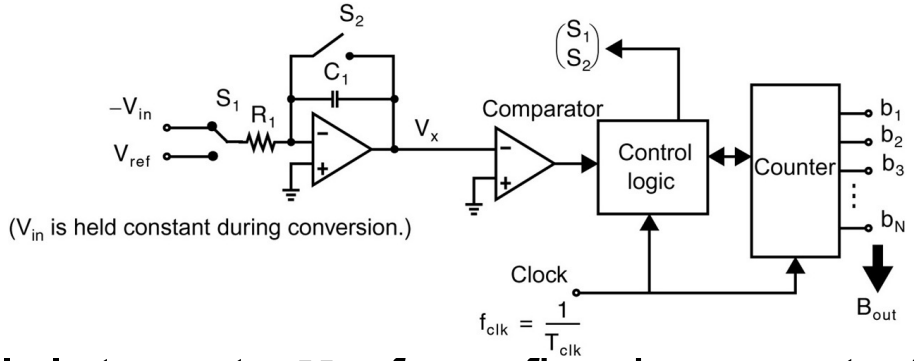
Charge redistribution SAR ADC

Error correction systems can be used to extend the resolution beyond the ~ 10 bit set by capacitor matching. Example in the textbook, fig. 17.11

We have considered SAR ADCs that determine one bit per clock cycle. To speed up, could consider converters that determine more than one bit per clock cycle.

Integrating ADC

High accuracy, but low speed.



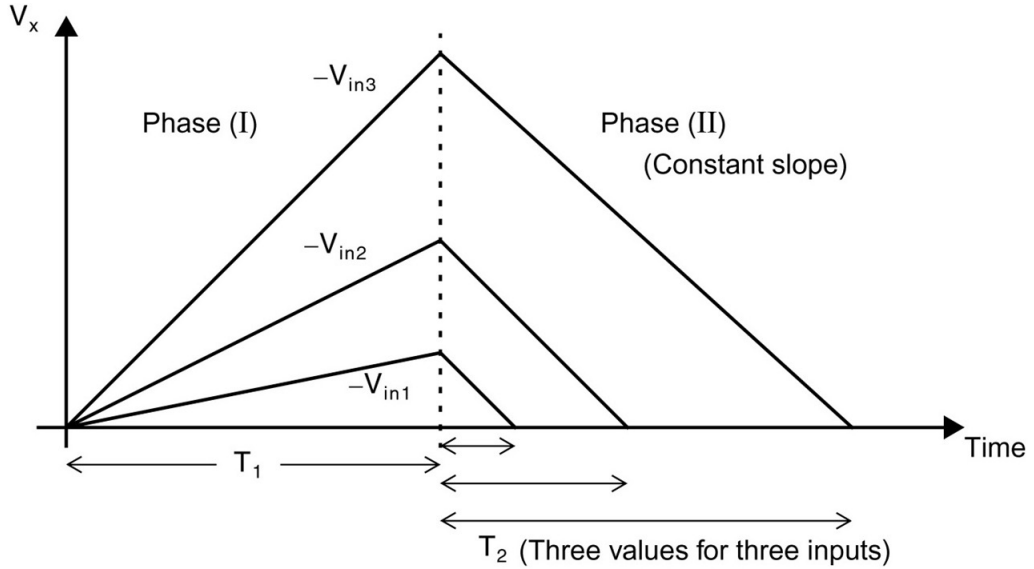
Phase I: Integrate V_{in} for a fixed amount of time

Phase II: Measure the time it takes to discharge the integrator to zero with a fixed voltage

Integrating ADC

Phase I lasts for $T_1 = 2^N \cdot T_{clk}$, counter value when phase II completes is the digital output.

Output word does not depend on RC.



OVERSAMPLING CONVERTERS

Introduction

Instead of relying only on the number of quantization levels, we can use signal processing to improve the resolution. ΔΣ modulator based data converters relies on **oversampling** and **noise shaping** to improve the resolution.

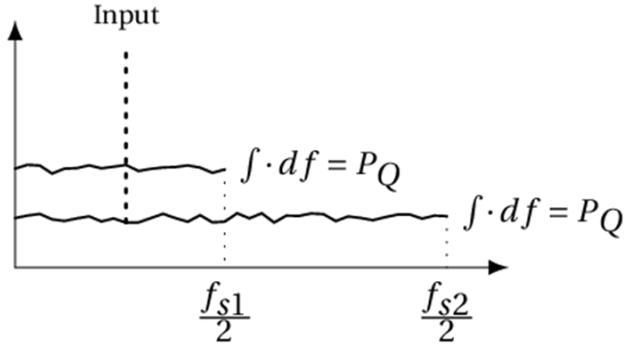
Introduction

Oversampling means that the sampling rate is increased to several times what is required just to avoid aliasing. Shortly, we will see why this helps improving the resolution.

Noise shaping means that the quantization noise is moved away from the signal band that we are interested in. (Oversampling implies that or signal band occupies a fraction of the full spectrum).

Oversampling

Important observation: The quantization noise depends on the *quantization step size only*. It does not depend on sampling frequency.



Sampling at a higher frequency reduces the amount of quantization noise inside our signal band.

Oversampling

$$\mathrm{OSR} \equiv \frac{F_s}{2F_b}$$

OSR is the fraction of the full sampled spectrum that our signal band occupies. The quantization noise (that we need to consider) is reduced proportionately. (Assume that we can remove the excess part of the spectrum with a “brick wall” filter).

$$P_Q = \frac{\Delta^2}{12}\frac{1}{\mathrm{OSR}}$$

Oversampling

The signal energy, assuming a full scale sine wave input, is not changed

$$P_S = \frac{\Delta^2 2^{2N}}{8}$$

$$\text{SQNR} = 10 \log \frac{P_S}{P_Q} = 6.02N + 1.76 + \underbrace{10 \log \text{OSR}}$$

Contribution from oversampling

Noise shaping

We can not magically remove noise (fundamental limit). However, we can “move” noise from the signal band to the oversampled part of the spectrum, which we are removing anyway.

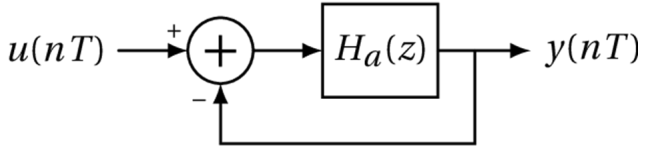
The idea is to **highpass filter the noise** (without spectrally shaping the signal). Done correctly, will **suppress noise in the signal band**, at the cost of **amplifying noise in the oversampled** part of the spectrum. (So noise is not moved as such).

Noise shaping

Signal transfer function, STF, $H_s = \frac{Y(z)}{U(z)}$. Analyze by setting the quantization noise input to 0 (superposition).

$$Y(z) = H_a(z)[U(z) - Y(z)]$$

$$H_s(z) = \frac{Y(z)}{U(z)} = \frac{H_a(z)}{1 + H_a(z)}$$

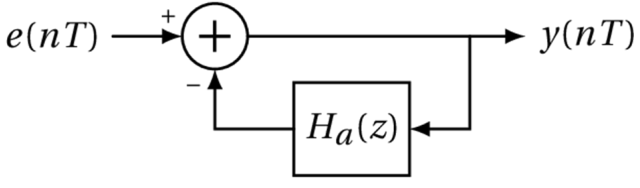


Noise shaping

Noise transfer function, NTF, $H_n(z) = \frac{E(z)}{Y(z)}$

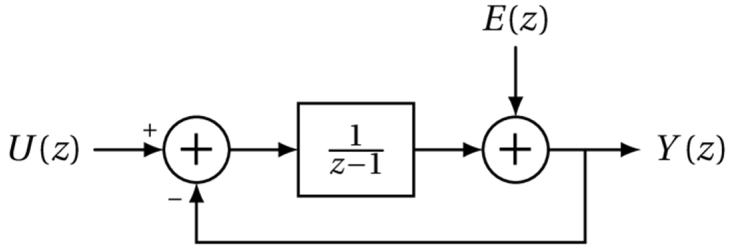
$$Y(z) = E(z) - H_a(z)Y(z)$$

$$H_n(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + H_a(z)}$$



Noise shaping

If we choose $H_a(z)$ to be an integrator …



Simple delay (signal)

$$H_s(z) = z^{-1}$$

Highpass filter (noise)

$$H_n(z) = 1 - z^{-1}$$

Noise shaping

$$P_Q = \int_0^{f_b} \frac{2f}{f_b}\frac{\Delta^2}{12} \cdot \underbrace{4\sin^2\frac{\pi f}{f_s}}_{\text{NTF}} df$$

$$\approx \left(\frac{2f_b}{f_s}\right)^3 \frac{\pi^2}{3}\frac{\Delta^2}{12}$$

Noise shaping

Assuming a full-scale sine wave input as before

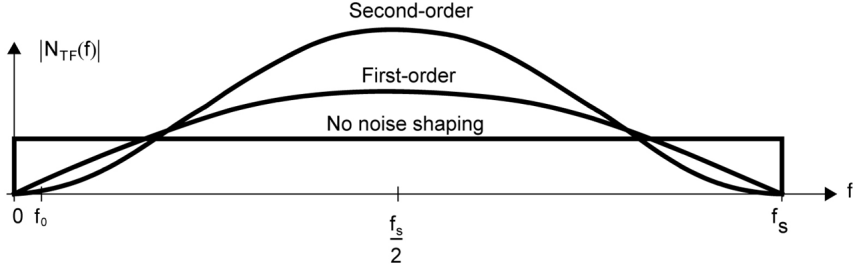
$$\mathrm{SQNR} = 6.02N + 1.76 - 5.17 + 30\log_{10}\mathrm{OSR}$$

Doubling the OSR improves the SQNR by 1.5 bits.
(compared to oversampling only, 0.5 bits).

Second order noise shaping

We can design the modulator such that it achieves a second order high-pass filter instead.

More suppression in the signal band (assuming sufficient OSR).



Idle tones

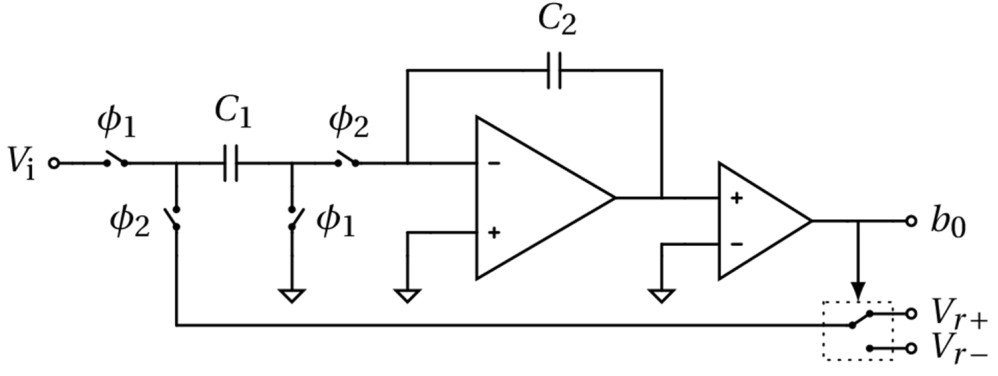
Sine or DC input and single bit quantization gives rise to quantization noise that is not white. Some inputs cause the output pattern to repeat after a deterministic number of samples.

Shows up as tones in the output spectrum and may occur inside the signal band.

Intentionally add noise to decorrelate the quantization noise pattern, *dithering.*

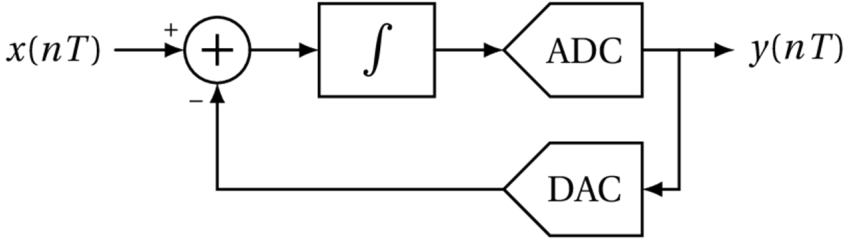
Circuit example

First order ΔΣ modulator discrete time implementation example (single bit quantizer).



Multi-bit quantization

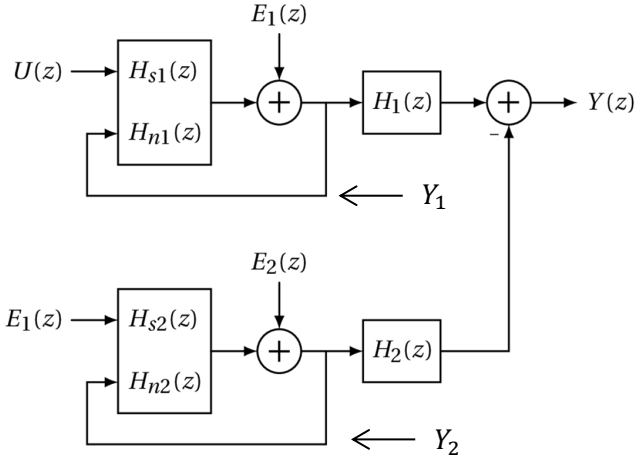
Multi-bit quantization helps stability and idle tones. Less inherent quantization noise. Quantizer non-linearity is shaped by the NTF, but the DAC must be linear to the full resolution of the system.



Multi-stage noise shaping (MASH)

A cascade structure where a higher order modulator is realized by two lower order modulators.

Input to the second modulator is the quantization noise from the first modulator.



Multi-stage noise shaping (MASH)

We want to cancel the quantization noise from the first modulator.

$$Y_1(z) = H_{s1}(z)U(z) + H_{n1}(z)E_1(z)$$
$$Y_2(z) = H_{s2}(z)E_1(z) + H_{n2}(z)E_2(z)$$
$$Y(z) = H_1(z)Y_1(z) - H_2(z)Y_2(z)$$

Choose $H_1(z)$ and $H_2(z)$ such that $E_1(z)$ cancels.

$$H_1(z)H_{n1}(z) - H_2(z)H_{s2}(z) = 0$$

Multi-stage noise shaping (MASH)

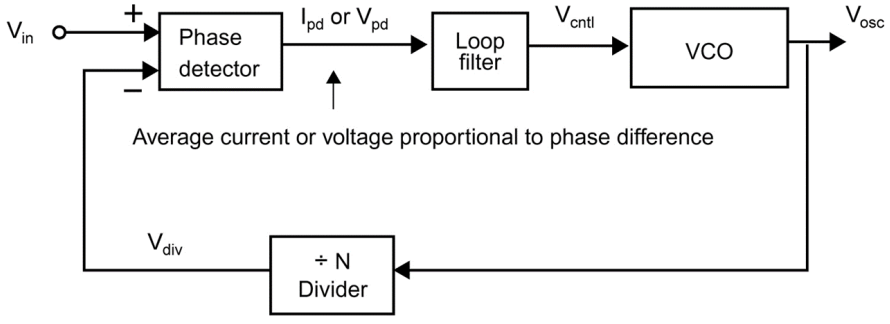
This requirement is satisfied if we choose

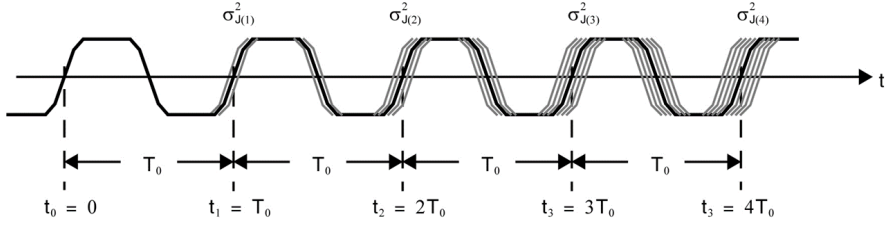
$$H_1(z) = H_{S2}(z) \text{ and } H_2(z) = H_{N1}(z)$$

Easy enough, but we must take into account that $H_1(z)$ and $H_2(z)$ are in digital and we must match them to $H_{S2}(z)$ and $H_{N1}(z)$ which are analog functions. Imperfections in the analog circuitry (offset, gain, etc.) will deteriorate the noise suppression.

Phase locked loops (Ch 19)

- Frequency multiplication
- Frequency synthesis
- Clock deskew (PLL or DLL)
- Clock recovery (from serial data)
- Demodulation





PHASE LOCKED LOOPS

Oscillator model

$$\phi(t) = \phi(0) + \int_0^t \omega(\tau)\, d\tau$$

Arbitrary initial phase

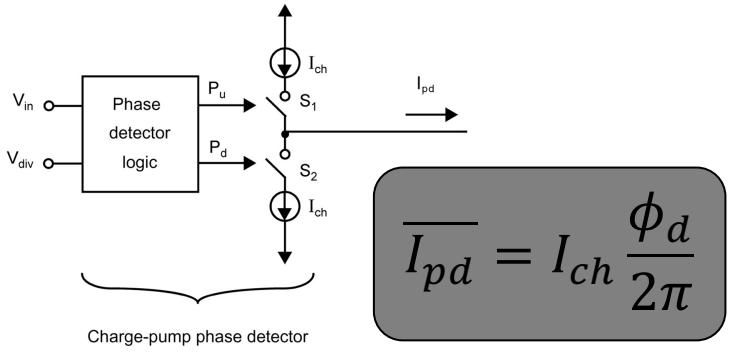
Laplace transform:

$$\phi(s) = \frac{\omega(s)}{s} = \frac{K_{osc} V_{ctl}(s)}{s}$$

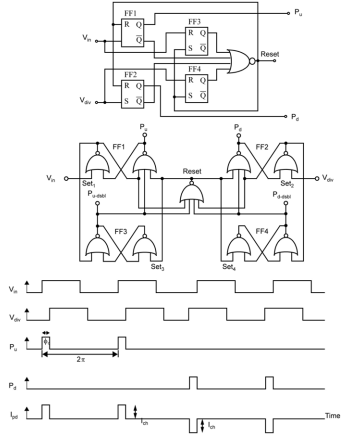
K_{osc} is a mapping from the control voltage to the frequency deviation. Assumed to be constant.

Charge pump PD

Generate a pulsed output who's pulse width is equal to the phase difference (ϕ_d). If REF comes first the VCO must speed up. Otherwise the VCO must slow down.

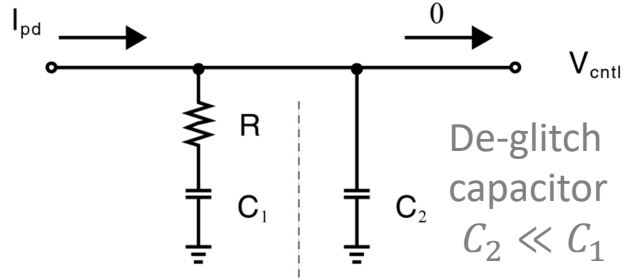


$$\overline{I_{pd}} = I_{ch}\frac{\phi_d}{2\pi}$$



Loop filter (LP)

$$\frac{V_{cntl}(s)}{I_{pd}(s)} = \frac{Z_1(s)Z_2(s)}{Z_1(s)+Z_2(s)} = \frac{1+C_1Rs}{s(C_1+C_2+C_1C_2Rs)}$$

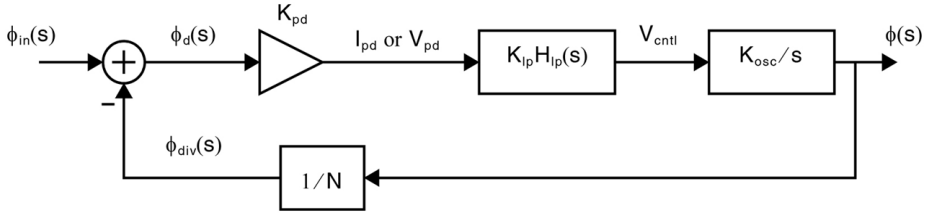


Simplify this as:

$$\frac{V_{cntl}(s)}{I_{pd}(s)} = K_{lp}H_{lp}(s) \approx \frac{1}{C_1}\frac{1+sRC_1}{s}$$

$K_{lp} = C_1^{-1}$
$\omega_z = (RC_1)^{-1}$

Small signal PLL model



$$A(s) = \frac{K_{pd}K_{lp}H_{lp}(s)K_{osc}}{s}, \qquad \beta = \frac{1}{N}$$

Phase difference (error) transfer function

$$\frac{\phi_d(s)}{\phi_{in}(s)} = \frac{1}{1+\beta A(s)} = \frac{Ns}{K_{pd}K_{lp}H_{lp}(s)K_{osc} + Ns}$$

Small signal PLL model

Similarly we find the output transfer function

$$H(s) = \frac{\phi(s)}{\phi_{in}(s)} = \frac{A(s)}{1 + \beta A(s)} = \frac{NK_{pd}K_{lp}H_{lp}(s)K_{osc}}{K_{pd}K_{lp}H_{lp}(s)K_{osc} + Ns}$$

$$H(s) = N\frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_z} + \frac{s^2}{\omega_{pll}^2}}$$

Second order denominator

$$1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}$$

Resonant frequency

$$\omega_0 = \omega_{pll}$$

$$Q = \frac{\omega_z}{\omega_{pll}}$$

Jitter and phase noise

However, noise introduces a random deviation from the ideal zero crossing time points

$$t_k = kT_0 + \tau_k$$

τ_k is the absolute jitter. Can also be expressed in radians

$$\phi_k = \tau_k \frac{2\pi}{T_0}$$

Total PLL phase noise

$$S_\phi(f) = S_{\phi,in}(f)|H(f)|^2 + V_n^2(f)|H_n(f)|^2 + S_{\phi,osc}(f)|H_{osc}(f)|^2$$

