Radio-Frequency Integrated Circuits

Harald Pretl
Johannes Kepler University
harald.pretl@jku.at

2025-11-08

Table of contents

1 Introduction

This is the material for an introductory radio-frequency integrated circuits course. The contents are largely based on [1] and [2]; these two books are an excellent introduction into this topic and are highly recommended! For a general introduction into RF and microwave [3] is a great read!

It is assumed that readers are familiar with the contents of this Analog Circuit Design course.

> **!** Important
>
> All course material (source code of this document, Jupyter notebooks for calculations, Xschem circuits, etc.) is made publicly available on GitHub (follow this link) and shared under the Apache-2.0 license.
>
> Please feel free to submit pull requests to fix typos or add content! If you want to discuss something that is not clear, please open an issue.
>
> The production of this document would be impossible without these (and many more) great open-source software products: `VS Code`, `Quarto`, `Pandoc`, `LaTeX`, `Typst`, `Jupyter Notebook`, `Python`, `Xschem`, `ngspice`, `CACE`, `pygmid`, `schemdraw`, `Numpy`, `Scipy`, `Matplotlib`, `Pandas`, `Git`, `Docker`, `Ubuntu`, `Linux`, ...

1.1 Wireless Transmission

In wireless transmission, we usually want to transmit data via a transmitter (TX) and a connected antenna to a receiver (RX) using an electromagnetic (EM) wave. This arrangement is shown in Figure 1.

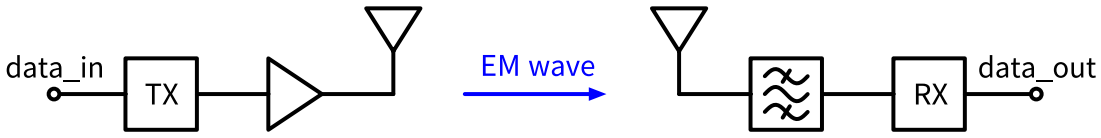

Figure 1: The block diagram of a simple wireless system.

Unfortunately, wireless transmission is hard. The wireless channel, i.e., the usage of electromagnetic waves to transmit information from a transmitter to a receiver, while tremendously useful, unfortunately has quite a few undesired features:

- The wireless channel is shared between all users.
- As a consequence, the available bandwidth is shared; this means that bandwidth is a scarce resource.
- The wireless channel has significant losses.
- The channel is time variant, as usually the transmitter and/or the receiver move, and/or the environment changes.

In order to estimate the power P_R of the wireless transmission at the receiver we can use Friis' transmission formula [3]:

$$P_\mathrm{R} = \frac{P_\mathrm{T}}{4\pi d^2} \cdot A_\mathrm{R} = P_\mathrm{T} \cdot \frac{A_\mathrm{R} \cdot A_\mathrm{T}}{d^2 \lambda^2} \tag{1}$$

Here, A_R (and A_T) is the effective area of the receive/transmit antenna, while d is the distance (line of sight) between the two antennas. The effective area of an antenna depends on the type and construction, but generally we can say that

$$A \propto \lambda^2$$

For an isotropic antenna (a theoretical construct where the radiation is equal in all directions) $A = \lambda^2/(4\pi)$, while for a $\lambda/2$-dipole $A = 0.13\lambda^2$. Of course, the speed of light c relates frequency f and wavelength λ of an electromagnetic wave by

$$c = \lambda f.$$

Generally speaking, the size of an electromagnetic antenna is proportional to the wavelength of the EM wave used for transmission. For many devices, we seek antennas on the order of a few centimeters, and this is why frequencies in the hundreds of MHz to GHz are so popular. Table 1 lists a few typical applications and their frequency and wavelength.

Table 1: Typical RF applications with their operating frequencies and corresponding wavelengths

Application	Frequency	Wavelength
FM Radio	88–108 MHz	2.8-3.4 m
WiFi (lowband)	2.4 GHz	12.5 cm
WiFi (highband)	5 GHz	6 cm
Bluetooth	2.4 GHz	12.5 cm
Cellular	0.6–5 GHz	6-50 cm
GNSS	1.575 GHz	19 cm

As you can see in Table 1 many of these antennas would not fit into the used device form factors, i.e., often we have to use electrically small antennas.

Note 1: Wavelength Calculation

Let's calculate the wavelength for a Bluetooth signal at 2.4 GHz. Given:

- Frequency $f = 2.4\ \text{GHz} = 2.4 \times 10^9\ \text{Hz}$
- Speed of light $c = 3 \times 10^8\ \text{m/s}$

Using the relationship $c = \lambda f$, we can solve for wavelength:

$$\lambda = \frac{c}{f} = \frac{3 \times 10^8\ \ \text{m/s}}{2.4 \times 10^9\ \ \text{Hz}} = 0.125\ \ \text{m} = 12.5\ \ \text{cm}$$

This means that a quarter-wavelength monopole antenna for 2.4 GHz Bluetooth would be approximately 3.1 cm long, which easily fits into most mobile devices.

In order to get a feeling for the attenuation experienced in wireless communication, we now calculate the following exemplary transmission. We will use the unit of dBm which is often used in RF design and is defined as

$$P\ |_{\text{dBm}} = 10 \cdot \log_{10}\left(\frac{P\ |_{\text{W}}}{1\,\text{mW}}\right) \tag{2}$$

Note 2: Wireless Transmission

We use the following parameters:

- Transmit power $P_{\mathrm{T}} = 1\ \mathrm{W}$
- Frequency $f = 2.4\ \mathrm{GHz}$
- Communication distance $d = 10\ \mathrm{km}$
- Using $\lambda/2$ dipoles on both ends

Using Equation 1 we calculate

$$P_{\mathrm{R}} = P_{\mathrm{T}} \cdot \frac{0.13\lambda^2 \cdot 0.13\lambda^2}{d^2\lambda^2} = P_{\mathrm{T}} \cdot 0.13^2 \left(\frac{\lambda}{d}\right)^2 = 2.64\,\mathrm{pW} = -85.8\,\mathrm{dBm}$$

With the transmit power of 1 W = 30 dBm we have an attenuation of 116 dB! This is a very large number!

The loss we calculated in Note 2 is called the free-space path loss (FSPL). It is the minimum loss we can expect in a wireless communication system. In reality, the situation is often even worse. The free-space path loss FSPL (in dB) can be calculated as

$$\mathrm{FSPL} = 20 \cdot \log_{10}(d/\mathrm{m}) + 20 \cdot \log_{10}(f/\ \mathrm{Hz}) + 20 \cdot \log_{10}\left(\frac{4\pi}{c}\ \mathrm{m/s}\right). \tag{3}$$

Equation 3 can be readily derived from Equation 1 and Equation 2 assuming isotropic antennas at transmitter and receiver. Using Equation 3 we can easily calculate the FSPL for different distances and frequencies. The results are shown in Figure 2. It should be noted that the FSPL increases by 20 dB per decade of distance and 20 dB per decade of frequency, making higher frequencies and longer distances very challenging.

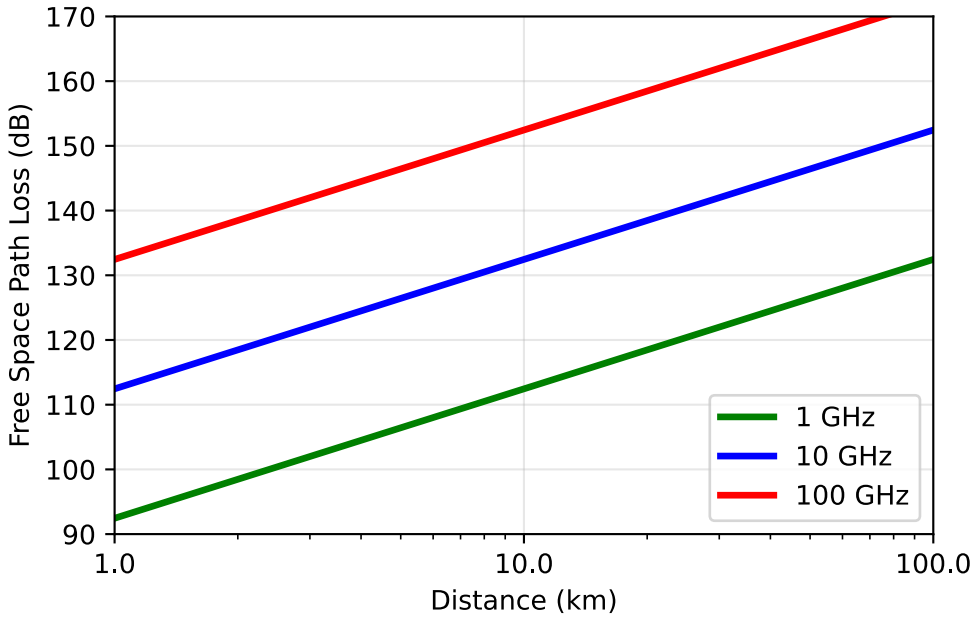

Figure 2: Free space path loss vs. distance for different frequencies (1 GHz, 10 GHz, and 100 GHz).

As dire as the situation of Figure 2 already looks, this is not even all factors considered:

- The given attenuation is for line-of-sight paths; often, the attenuation is significantly higher than this due to blockage by buildings, mountains, rain, or foliage.
- In the absence of a direct line-of-sight path, the EM wave is redirected by reflections, causing additional attenuation, and potential destructive interference by multi-path reception.

The consequences of this are (among others):

- The transmitter needs to generate enough **transmit power** to overcome the transmission loss; this has to be done often with high **efficiency**, as the transmit device is battery operated or limited by cooling.
- The receiver has to be able to process **weak signals**, i.e., the **noise** level of the signal processing has to be very low.
- Often, the receive signal is very weak, while there are strong signals at other frequencies (i.e., other wireless transmitters are located close to the receiver). This means the receiver has to be able to process a weak signal while simultaneously tolerating **large interfering signals** (called blockers).
- Since the frequency spectrum is shared among many users and wireless applications, the transmit information has to be packed efficiently into a **small bandwidth**.
- Very often, wireless devices are battery-operated. This means transmit and receive functions have to be implemented using **minimum power consumption**.

As stated in the beginning, designing wireless systems is hard.

1.2 Wireless Standards

In order to allow communication between different devices, different operators, and different manufacturers, wireless communication is standardized. There are many different standards, each with its own characteristics. Wireless standards define every aspect of wireless commu-

nication, and can be documents with hundreds or thousands of pages. Here, we mainly focus on the radio-frequency and analog aspects of wireless standards. Summarized in Table 2 are a few popular wireless standards with their main characteristics.

Table 2: Comparison of wireless communication standards

Standard	GSM (2G)	WCDMA (3G)	LTE (4G)**5G NR**	WiFi	Bluetooth	GNSS
Frequency range (MHz)	850, 900, 1800, 1900	850, 900, 1700, 1900, 2100	Multiple bands 450…7100 **(FR1)**, **24000…48000 (FR2)**	2400, 5000, 6000	2400	1500, 1200
Modulation	GMSK, 8PSK (EDGE)	QPSK (DL), BPSK (UL), 16QAM (HSPA), 64QAM (HSPA)	QPSK, 16QAM, 64QAM (DL+UL) 256QAM (DL**+UL**)	BPSK, QPSK, 16QAM, 64QAM, 256QAM, 1024QAM, 4096QAM	GFSK (m=0.28…0.35), π/4-DQPSK, 8DPSK	BPSK, QPSK
Transmission/ multiple access	TDMA, FDMA	DS-CDMA	OFDMA (DL**+UL**), SC-FDMA/**DFT-s-OFDM (UL)**	OFDM, CSMA/CA	FHSS	CDMA
Duplex	FDD	FDD	FDD, TDD	TDD	TDD	n/a
Channel bandwidth	200 kHz	5 MHz	1.4, 3, 5, 10, 15, 20, …, **100 MHz (FR1), 400 MHz (FR2)**	10, 20, 40, 80, 160, 320 MHz	1 MHz	16…24 MHz
Symbol rate	270.833 ksym/s	3.84 Msym/s	15/**30/60** ksym/s	312.5 ksym/s	1 Msym/s	50 sym/s
Pulse shaping	Gaussian (BT=0.3)	Root Raised Cosine (α =0.22)	Rectangular	Rectangular	Gaussian (BT=0.5)	Rectangular
Transmit power	1…2 W	250 mW	200 mW (FDD), 400 mW (TDD)	100 mW	1…100 mW	n/a
PAR (UL)	0 dB (GMSK), 3 dB (8PSK)	3…8 dB	6…8 dB	Up to 12 dB	0 dB (GFSK), 3 dB (8DPSK)	n/a
MIMO	no	Not realized (DL 2x2)	DL 4x4 (up to 8x8)	2x2 (up to 8x8)	no	no
Channel bond	no	Up to 4x5 MHz	Up to 7x20/4x100 MHz	Up to 80+80+80+80 MHz	no	no

During this course, we will learn what these terms mean and how they impact the design of RF integrated circuits.

As you can see in Table 2, since LTE (4G) and 5G NR, many additional bands have been defined in the sub-6 GHz range (FR1) and also in the mm-wave range (FR2, 24.25 to 52.6 GHz). This means that modern wireless devices have to support many different frequency bands, which makes the design of RF frontends even more challenging. A good overview of the different frequency bands is given here for LTE and 5G NR.

RFIC design is a multidisciplinary field, requiring knowledge from various engineering domains, as shown in Figure 3. This makes RFIC design challenging, but also very interesting!

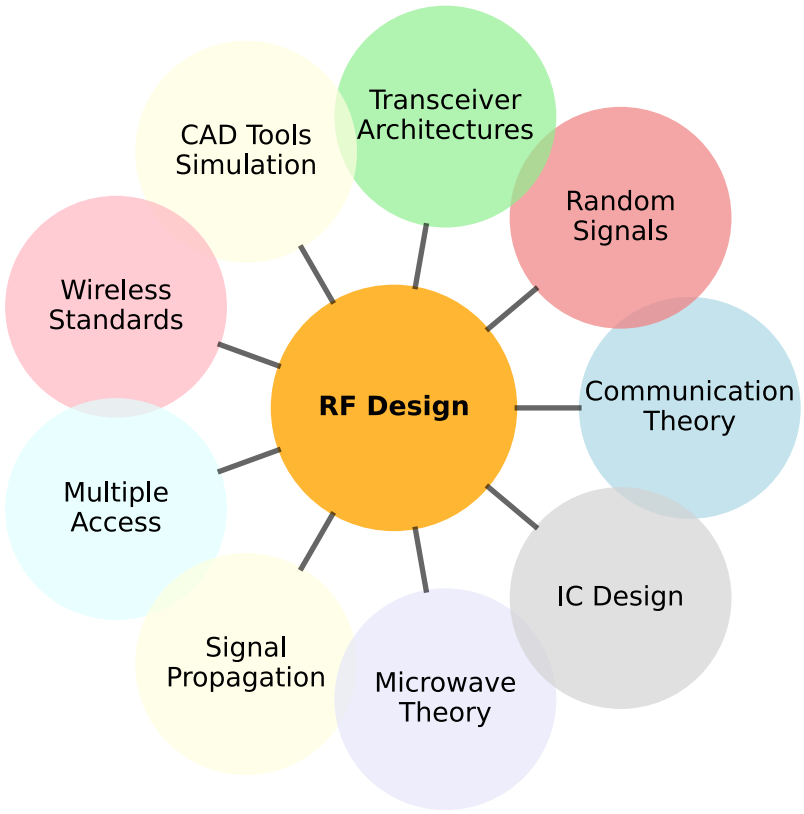

Figure 3: RF design as a multidisciplinary field requiring knowledge from various engineering domains (adapted from [1]).

Further, RFIC design requires careful consideration of many different aspects, as shown in Figure 4. Many parameters are often tightly coupled, requiring careful trade-offs during the design process.

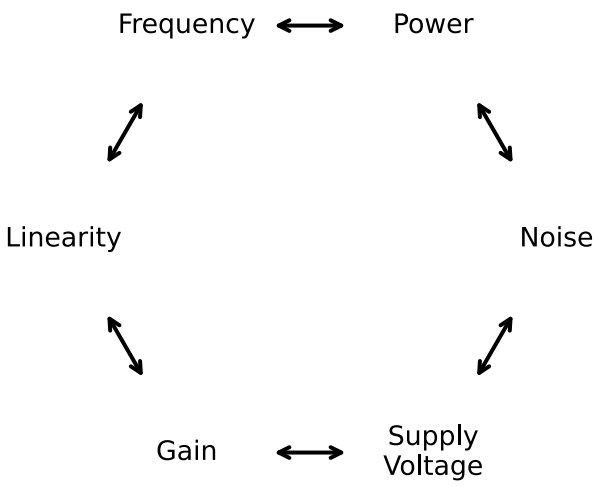

Figure 4: RFIC require careful design considerations and trade-offs (adapted from [1]).

2 Fundamentals

In this section, we will discuss a few important concepts which will be instrumental in the further study of RF circuits and systems. As signals in RF circuits and systems are often limited on the top end by linearity, and on the bottom end by noise, we will discuss these two topics in some detail.

2.1 Channel Capacity

In Section 1 we have already discussed the fact that we need to pack information into a minimum bandwidth, as the available spectrum is limited. To appreciate the limits of information transfer, we need to understand how much information can be transmitted over a given bandwidth. This limit is given by the *Shannon-Hartley theorem*, which gives the maximum data rate C (in bit/s) that can be transmitted over a communication channel with bandwidth B (in Hz) and signal-to-noise ratio SNR (with linear units):

$$C = B \cdot \log_2(1 + \mathrm{SNR}) \tag{4}$$

This formula gives us a theoretical upper limit on the data rate that can be achieved with a given bandwidth and SNR *under optimal conditions*. It is important to note that this limit is only achievable with ideal coding and modulation schemes, which are not practical in real-world systems. However, it provides a useful benchmark for evaluating the performance of communication systems.

i Note 3: Channel Capacity Example

Let us calculate the channel capacity for a system with a bandwidth of 2 MHz and an SNR of 7 dB (the BW and minimum SNR of Bluetooth LE for 1 Mbps). First, we need to convert the SNR from dB to linear units:

$$\text{SNR} = 10^{7/10} \approx 5$$

Now we can use Equation 4 to calculate the channel capacity:

$$C = B \cdot \log_2(1 + \text{SNR}) = 2\,\text{MHz} \cdot \log_2(1 + 5) = 2\,\text{MHz} \cdot 2.585 \approx 5.2\,\text{Mbps}$$

This sounds reasonable, as the user data rate for Bluetooth LE is 1 Mbps for the given SNR, which allows for quite some overhead for coding and protocols.

2.2 Linearity

As we have already seen in Section 1.1 the transmitter has to process large signals without distorting them, while the receiver has to process small signals in the presence of large signals. Both situations mean we need metrics and models to quantify and discuss linearity properties.

We are going to use a very simple, time-invariant model to study linearity, based on a Taylor polynomial.

! Linearity and Time Invariance in RF Systems

We use time invariance to simplify the mathematics. In practice, many circuits and systems will show time variant behavior which leads to quite a few very interesting and important phenomena! A time-invariant nonlinear system is also called a "memoryless" system, as the output at time t only depends on the input at time t.

In contrast, a system with memory (i.e., time-variant) will have an output at time t which depends on the input at time t and also on past inputs (e.g., at times $t - \Delta T$, $t - 2\Delta T$, etc.). Examples of systems with memory are filters, which have a frequency-dependent response, or power amplifiers with thermal memory effects.

We model a nonlinear circuit block with the following Taylor polynomial:

$$y(t) = \alpha_0 + \alpha_1 x(t) + \alpha_2 x(t)^2 + \alpha_3 x(t)^3 + \ldots \tag{5}$$

Usually, the blocks under study will have higher order nonlinear terms but we often stop at 3rd order to keep things simple. For practical work, higher order terms should be included if necessary.

Which $x(t)$ should we use to study wireless systems? Often, the bandwidth f_{BW} of a transmit signal is much smaller than the center frequency f_0, i.e., $f_{\text{BW}} \ll f_0$. In this case using a sinusoidal signal as a model is both simple to handle and approximately correct.

2.2.1 Single-Tone Linearity

We thus set (with A being the amplitude of the input signal and $\omega = 2\pi f$ the angular frequency)

$$x(t) = A\cos(\omega t)$$

and insert it into Equation 5. After some simple trigonometric manipulations we are at

$$\begin{aligned} y(t) = & \underbrace{\frac{1}{2}\alpha_2 A^2}_{\text{dc component}} + \underbrace{\left(\alpha_1 A + \frac{3}{4}\alpha_3 A^3\right)\cos(\omega t)}_{\text{fundamental}} \\ & + \underbrace{\frac{1}{2}\alpha_2 A^2 \cos(2\omega t)}_{\text{2nd harmonic}} + \underbrace{\frac{1}{4}\alpha_3 A^3 \cos(3\omega t)}_{\text{3rd harmonic}} \end{aligned} \tag{6}$$

Looking at Equation 6 we can make a few interesting observations:

- Even-order nonlinearity (α_2) creates low-frequency components; it effectively adds frequency components related to the envelope A. If A is a constant then this results in a dc term; if $A(t)$ is time variant it will create a squared version of it at low frequencies.
- The α_1 term is the gain of the circuit block.
- Odd-order nonlinearity (α_3) can impact the gain of the fundamental term passing through the block. Depending on the sign of α_3 this can lead to gain contraction or expansion.
- Even- and odd-order nonlinearities create additional frequency components, so-called harmonics of the fundamental frequency. These harmonics are often unwanted, as they are far outside the wanted transmission frequency range, and need to be minimized, by either
  1. use a lowpass filter to filter these harmonics, or
  2. increase the linearity, i.e., make the α_2, α_3, etc., small enough.

The created harmonics are illustrated in Figure 5. Note that measuring harmonics to quantify the nonlinearity metrics like α_2 and α_3 is often not very accurate, as these harmonics are often filtered in bandwidth-limited systems.

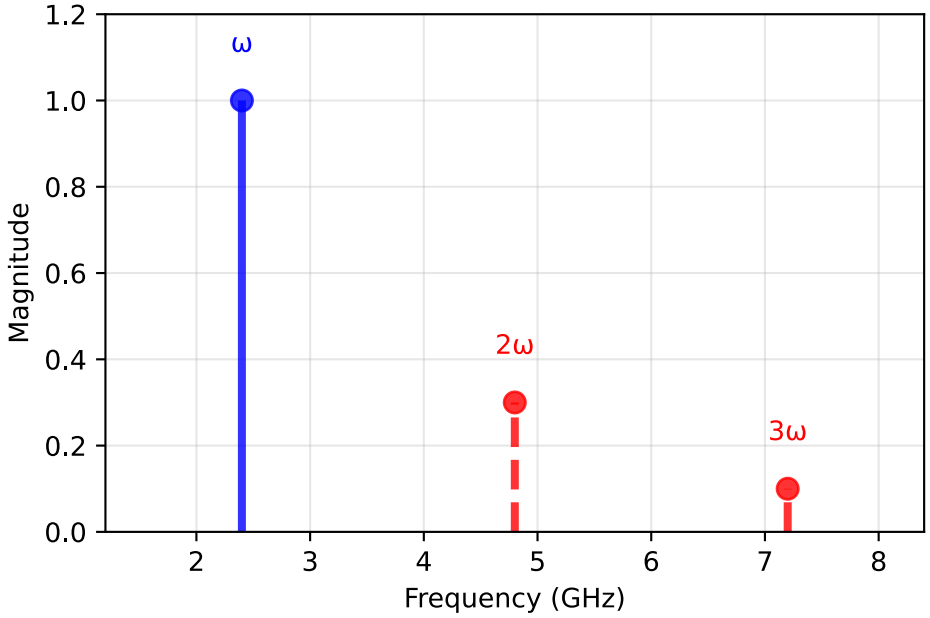

Figure 5: Single-tone test showing created harmonics at 2ω and 3ω.

How can we quantify the nonlinearity with a one-tone test? We can sweep the input signal $x(t)$ in amplitude, and observe the output $y(t)$. If the observed gain drops by 1 dB from the small-signal value we note the input power, and call this point the **1dB compression point** ($P_{1\text{dB}}$). We should always add whether this 1dB compression point is input- or output-referred to avoid ambiguity. The diagram in Figure 6 shows this test ($\alpha_1 = 100$, $\alpha_3 = -0.2$).

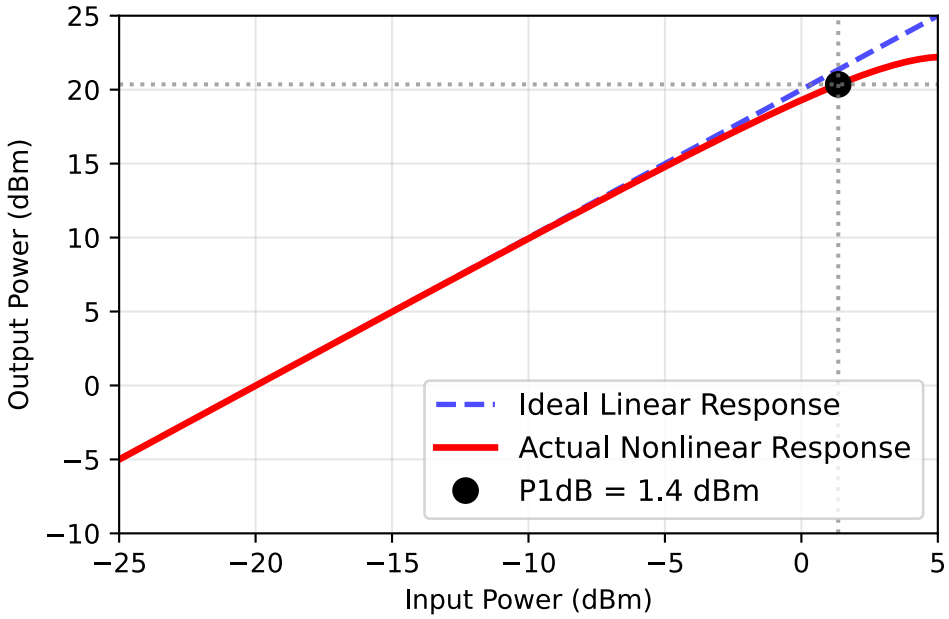

Figure 6: 1dB compression point test showing input vs output power relationship and the definition of P1dB.

! Compressive vs. Expansive Behavior

Note that for compressive behaviour, α_3 and α_1 have different signs, while for expansive behaviour, they have the same sign.

At some point, every circuit block will show compressive behavior, as the maximum signal amplitude will be limited by power supply voltages, device breakdown voltages, etc.

2.2.2 Multi-Tone Linearity

We now elevate our investigations and apply two sinusoids with different frequencies and different amplitudes and see which signals we get at the output of the nonlinear block. The two-tone test and resulting third-order intermodulation products (IM3) are illustrated in Figure 7.

$$x(t) = A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t)$$

We apply the above stimulus to our nonlinear model described by Equation 5 and again, after some trigonometric manipulations, arrive at:

$$y(t) = y'(t) + y''(t) + y'''(t) \tag{7}$$

As many different frequency components are created by this simple two-tone test (and nonlinearity only up to 3rd order) we split the result into different equations and look at the result separately.

First, we start with the fundamental tones:

$$\begin{aligned} y'(t) = &\left(\underbrace{\alpha_1 A_1 + \frac{3}{4}\alpha_3 A_1^3}_{\text{compression/expansion}} + \underbrace{\frac{3}{2}\alpha_3 A_1 A_2^2}_{\text{cross-modulation/desens}} \right) \cos(\omega_1 t) \\ &+ \left(\underbrace{\alpha_1 A_2 + \frac{3}{4}\alpha_3 A_2^3}_{\text{compression/expansion}} + \underbrace{\frac{3}{2}\alpha_3 A_2 A_1^2}_{\text{cross-modulation/desens}} \right) \cos(\omega_2 t) \end{aligned} \tag{8}$$

As shown in Equation 8, interesting things happen:

- We (again) have the gain compression/expansion effect as already discussed in Section 2.2.1.
- In addition, we have **cross-modulation**, i.e., the envelope of one tone (e.g., $A_2(t)$ of the tone at ω_2) impacts the envelope of the other tone at ω_1. This can lead to unwanted signal distortion, even if there is a large frequency separation between ω_1 and ω_2!
- Further, since the sign of α_3 is usually opposite to α_1, this can also lead to **desensitization** ("desens"). If, for example, $A_2 \gg A_1$, then there would be no compression due to the tone ω_1 itself, however, the large tone at ω_2 will lead to gain compression of the tone at ω_1; this effect is called desense.

We now look at the next class of generated tones:

$$\begin{aligned} y''(t) = &\frac{1}{2}\alpha_2 A_1^2 + \frac{1}{2}\alpha_2 A_2^2 \\ &+\alpha_2 A_1 A_2 \cos[(\omega_1 - \omega_2)t] \\ &+\alpha_2 A_1 A_2 \cos[(\omega_1 + \omega_2)t] \end{aligned} \tag{9}$$

As we can see in Equation 9 new tones are created (besides the low frequency components we already know from the single-tone test) at the sum and difference of ω_1 and ω_2. These new frequency components are called "**intermodulation products of second order**" (IM2). These tones are created by the even-order nonlinearity (α_2). These IM2 products are far away from the wanted tones, so are often not very problematic in amplifiers (but there can be exceptions!). However, they can be very problematic in frequency conversion blocks like mixers. We will come back to this point when discussing zero-IF receivers.

We now investigate the next couple of tones:

$$
\begin{aligned}
y'''(t) = &\frac{3}{4}\alpha_3 A_1^2 A_2 \cos[(2\omega_1 + \omega_2)t] \\
&+\frac{3}{4}\alpha_3 A_1^2 A_2 \cos[(2\omega_1 - \omega_2)t] \\
&+\frac{3}{4}\alpha_3 A_1 A_2^2 \cos[(2\omega_2 + \omega_1)t] \\
&+\frac{3}{4}\alpha_3 A_1 A_2^2 \cos[(2\omega_2 - \omega_1)t]
\end{aligned} \tag{10}
$$

The tones shown in Equation 10 are called "**intermodulation products of third order**" (IM3), and are caused by the odd nonlinearities (like α_3). While the IM3 tones located at $2\omega_1 + \omega_2$ and $\omega_1 + 2\omega_2$ are similar to the sum IM2 tone and far away from ω_1 and ω_2, the other two tones are concerning.

Expressing $\Delta\omega = \omega_2 - \omega_1$ (and assuming $\omega_1 < \omega_2$), the building law of $2\omega_1 - \omega_2 = \omega_1 - \Delta\omega$ and $2\omega_2 - \omega_1 = \omega_2 + \Delta\omega$ results in new tones right besides ω_1 and ω_2, with a frequency separation only defined by $\Delta\omega$. This situation is illustrated in Figure 7.

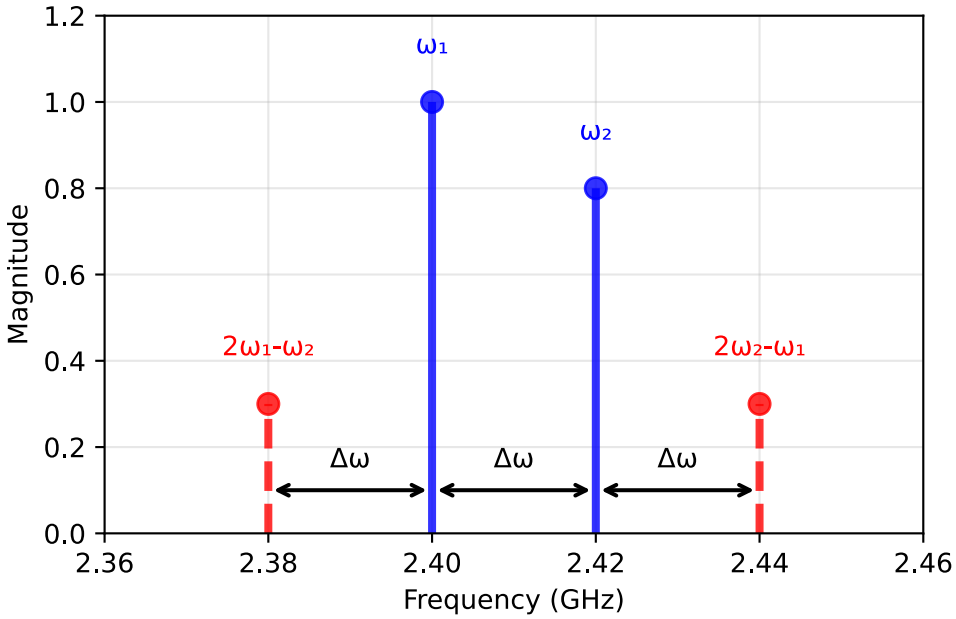

Figure 7: Two-tone test showing fundamental frequencies ω_1, ω_2 and third-order intermodulation products (IM3) at $2\omega_1$-ω_2 and $2\omega_2$-ω_1.

This close localization of the IM3 tones can also be utilized to characterize nonlinear performance. Using gain compression or harmonic generation (H3) it can be very difficult to extract nonlinearity of third order (α_3). However, using a two-tone test, the IM3 tones can be readily measured, even if the measured signal path shows a **bandpass characteristic**! As RF systems frequently employ bandpass filters to suppress out-of-band signals, this is a very important property of the two-tone test.

The resulting test is called a two-tone test yielding the third-order intercept point (IP3). This test is widely used in RF design to characterize the linearity of amplifiers, mixers, and complete

transceiver systems. The power relationship between fundamental tones and IM3 products as a function of input power is shown in Figure 8.

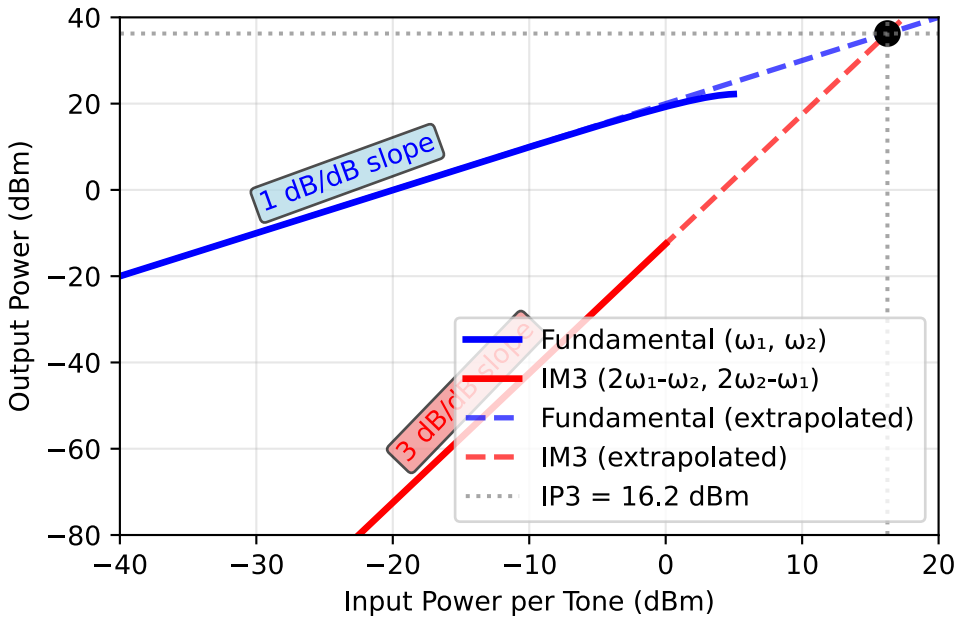

Figure 8: Two-tone IM3 test showing fundamental and IM3 product power vs. input power, with IP3 intercept point definition. Equal input power per tone is assumed.

Note that, as shown in Figure 8, the IM3 products rise with a slope of 3 dB/dB, i.e., if the input power is increased by 1 dB, the IM3 products increase by 3 dB. The fundamental tones rise with a slope of 1 dB/dB (as long as we are in the linear region). The IP3 point is defined as the intersection of the **extrapolated** linear lines of fundamental and IM3 products. As both lines have different slopes, this intersection point is usually far outside the actual operating range of the circuit block under test!

When calculating the IIP3 (input-referred IP3) we can use the following formula, assuming equal input power per tone. It is important to always check the slope of the IM3 products to ensure that we are indeed in the third-order region! If the input power per tone is P_{in} (in dBm) and the input-referred power of one IM3 tone is P_{IM3} (in dBm), then the input-referred IP3 is given by

$$\text{IIP3} = P_{\text{in}} + \frac{P_{\text{in}} - P_{\text{IM3}}}{2} \tag{11}$$

Further, for midly nonlinear systems (i.e., α_3 is dominating), the IIP3 can be approximated from the 1dB compression point as

$$\text{IIP3}\,|_{\text{dBm}} \approx P_{\text{1dB}}|_{\text{dBm}} + 9.6\,\text{dB}\ . \tag{12}$$

If we have two blocks which are cascaded, and we know the gain and IIP3 of both blocks, we can calculate the overall IIP3 of the cascade with the following approximation. An exact calculation is very involved, as the nonlinearities of the first block (and the resulting tones) will be processed by the second block, creating even more tones; this process escalates very quickly. However, for practical purposes, the following approximation is often sufficient:

$$\frac{1}{\mathrm{IIP3}_{\mathrm{total}}} \approx \frac{1}{\mathrm{IIP3}_1} + \frac{G_1}{\mathrm{IIP3}_2} + \frac{G_1 G_2}{\mathrm{IIP3}_3} \tag{13}$$

Here G_1 is the linear gain of the first block, and $\mathrm{IIP3}_1$, $\mathrm{IIP3}_2$ are the input-referred IP3 of the first and second block, respectively. Note that all powers have to be in linear units (i.e., Watts) when using Equation 13. An even more simplified version of Equation 13 can be used with all quantities given in dBm and dB, respectively:

$$\mathrm{IIP3}_{\mathrm{total}} \approx \min\{\mathrm{IIP3}_1, \mathrm{IIP3}_2 - G_1, \mathrm{IIP3}_3 - G_1 - G_2\} \tag{14}$$

A typical RF system cascade with multiple blocks and their individual IIP3 contributions is shown in Figure 9.

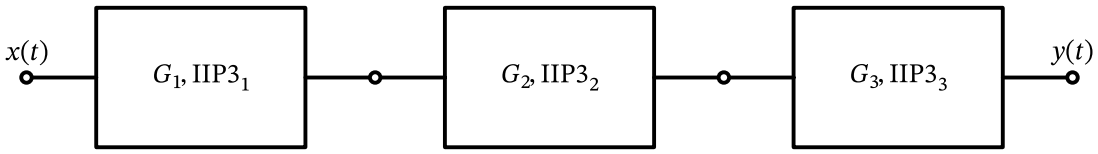

Figure 9: Block cascade for IIP3 calculation showing multiple stages with gains and individual IIP3 values.

i Note 4: Simple IIP3 Cascade Calculation

Let's calculate the overall IIP3 of two cascaded blocks. The first block is a low-noise amplifier with an IIP3 of −10 dBm and a gain of 20 dB. The second block is a mixer that has a gain of 10 dB and an IIP3 of 5 dBm. What is the overall IIP3?

Using Equation 14 we can quickly estimate:

$$\mathrm{IIP3}_{\mathrm{total}} \approx \min\{-10\,\mathrm{dBm}, 5\,\mathrm{dBm} - 20\,\mathrm{dB} = -15\,\mathrm{dBm}\} = -15\,\mathrm{dBm}$$

We see that the overall IIP3 is limited by the linearity of the second block, as the first block amplifies all signals (including blockers) by 20 dB before they reach the second block.

2.3 Noise

Just as nonlinearity is a limiting factor for large signals, noise is the limiting factor for small signals. Noise is present in all electronic circuits and systems, and it is impossible to avoid it. However, we can try to minimize its impact on system performance.

Noise is usually characterized by its power spectral density (PSD) in units of Watts per Hertz (W/Hz). For example, thermal noise at room temperature has a PSD of approximately $kT = 4 \times 10^{-21}$ W/Hz, or −174 dBm/Hz (with the Boltzmann constant $k = 1.38 \times 10^{-23}$ J/K). This means that if we have a bandwidth of 1 MHz, the total thermal noise power would be:

$$P_{\mathrm{thermal}} = \mathrm{PSD} \cdot B = -174\,\mathrm{dBm/Hz} + 10\log_{10}\left(\frac{1\,\mathrm{MHz}}{1\,\mathrm{Hz}}\right) = -114\,\mathrm{dBm}$$

The PSD of noise can be flat vs. frequency (which is called "white noise"), or can decrease with frequency (e.g., "flicker noise" or "1/f noise"). Further, noise can be generated by resistors (thermal noise), semiconductors (shot noise, generation-recombination noise), etc. A detailed discussion of noise sources can be found in [4] or [5].

2.3.1 Types of Noise Generation

Resistors generate thermal noise, which is white noise with a PSD of $4kTR$ (in V^2/Hz) when looking at the voltage across the resistor, or $4kT/R$ (in A^2/Hz) when looking at the current through the resistor. This noise is generated by the random thermal motion of charge carriers in the resistor.

! Thermal Noise

Note that the simple approximation given above is only valid for reasonably high frequencies and typical temperatures, and is known as the Rayleigh-Jeans approximation of Planck's blackbody radiation accounting for quantum effects and is given by [3]

$$\text{PSD} = \frac{4Rhf}{e^{hf/kT} - 1}$$

where h is the Planck constant ($h = 6.626 \times 10^{-34}$ Js) and f is the frequency. The Rayleigh-Jeans approximation is valid for $f \ll kT/h$, which is approximately 6 THz at room temperature (290 K).

We can integrate the above PSD over the full frequency range and show the rms noise voltage of a resistor R is bounded to

$$\overline{v_\text{n}^2} = \int_0^\infty \frac{4Rhf}{e^{hf/kT} - 1} df = \frac{2(\pi kT)^2}{3h} \cdot R$$

which equates to approximately 13 mVrms noise voltage for a 1 kΩ resistor at room temperature (which is impossible to measure in practice, as there will be some form of bandwidth limitation in any real measurement setup).

MOSFETs generate several types of noise, the most important ones being the thermal noise of the channel and flicker noise.

The thermal noise of the channel can be modeled as a current noise source between drain and source with a PSD of $\overline{I_\text{n}^2} = 4kT\gamma g_{d0}$ (in A^2/Hz), where γ is a process-dependent parameter (usually between 2/3 and 2). The parameter g_{d0} is the small-signal output conductance of the MOSFET in triode, i.e., $g_{d0} = g_\text{ds}$, or equal to $g_{d0} = g_\text{m}$ when in saturation.

In saturation, it is often useful to express the thermal noise as a voltage noise source at the gate with a PSD of $\overline{V_\text{n}^2} = 4kT\gamma/g_\text{m}$ (in V^2/Hz). We can see that we can lower this noise of the MOSFET by increasing the transconductance g_m, which can be achieved by increasing the bias current.

In addition, at high frequencies, the MOSFET also has induced gate-current noise, which is correlated with the channel thermal noise. A detailed discussion of this noise source can be found in [5].

Flicker noise is usually modeled as a voltage noise source at the gate with a PSD of $K_f/(C'_\text{ox}WLf)$ (in V^2/Hz), where K_f is a process-dependent parameter, C'_ox is the oxide capacitance per unit area, L and W are the length and width of the MOSFET, and f is the frequency. Note that we can lower the flicker noise by increasing the area of the MOSFET

(WL), however, this increases the parasitic capacitances associated with the MOSFET, and this often prohibitive for RF operation!

In **bipolar junction transistors (BJTs)**, the most important noise source is the shot noise due to the diffusion current in the base-emitter junction. Its PSD can be modeled as a current noise source between collector and emitter with a PSD of $2qI_\mathrm{C}$ (in A^2/Hz), where q is the elementary charge ($q = 1.6 \times 10^{-19}$ C) and I_C is the DC collector current.

> ! Equivalence of Shot and Thermal Noise
>
> Note that it has been shown in [6] that thermal noise and shot noise are actually equivalent, as both are generated by the random, thermally agitated motion of charge carriers!

Ideal **capacitors** and **inductors** do not generate noise, however, real capacitors and inductors have parasitic resistances which generate thermal noise.

In RF systems additional noise sources can be present. One noteworthy example is the **cosmic microwave background** radiation, which can be modeled as a noise temperature of approximately 3 K. While this is negligible compared to thermal noise at room temperature (approximately 290 K), it can be significant in very low-noise systems, such as radio telescopes pointing to the sky. Another important noise source in RF systems is the **atmospheric noise**, which is generated by natural phenomena like lightning or in the ionosphere.

> i A Note on Circuit Noise Calculations
>
> When doing circuit noise calculations, it is instructive to keep the following points in mind:
>
> - For circuit calculations involving noise sources it is convenient to replace the power spectral density by equivalent sinusoidal generators in small bandwidths.
> - The noise power spectral density in a small bandwidth Δf is given by $\overline{V_\mathrm{n}^2} = \overline{v_\mathrm{n}^2}/\Delta f$ and $\overline{I_\mathrm{n}^2} = \overline{i_\mathrm{n}^2}/\Delta f$.
> - The quantities $\overline{V_\mathrm{n}^2}$ and $\overline{I_\mathrm{n}^2}$ can be considered the mean-square value of sinusoidal generators. Using these values, network noise calculations reduce to familiar sinusoidal circuit-analysis calculations using V_n and I_n.
> - Multiple *independent* noise sources can be calculated individually at the output, and the total noise in bandwidth Δf is calculated as a mean-square value by adding the individual mean-square contributions from each sinusoid.

2.3.2 Noise in Impedance-Matched Systems

We now want to calculate the maximum noise power that can be extracted from a noisy source. We assume the following situation as shown in Figure 10. Note that the voltage source $\overline{V_\mathrm{n,s}^2}$ models the thermal noise of the source resistor R_s resulting in a Thevenin equivalent circuit.

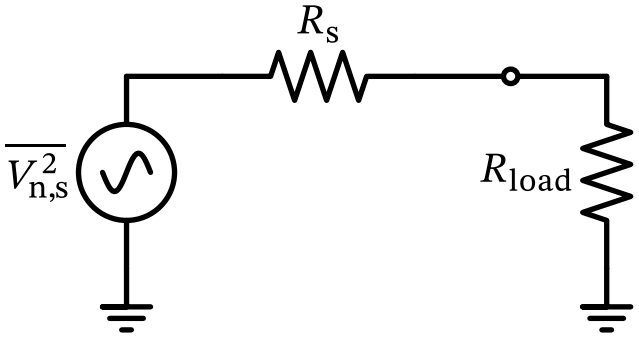

Figure 10: A noise-matched system with source and load impedances.

We know that the noise of the source resistor is given by $\overline{V_\mathrm{n,s}^2} = 4kTR_\mathrm{s}$. We assume the load resistor R_load as noiseless and matched to the source resistor, i.e., $R_\mathrm{load} = R_\mathrm{s}$ for maximum power transfer. The noise power spectral density delivered to the load resistor is then given by

$$P_\mathrm{n,load} = \frac{\overline{V_\mathrm{n,load}^2}}{R_\mathrm{load}} = \frac{\overline{V_\mathrm{n,d}^2}}{4R_\mathrm{s}} = kT \tag{15}$$

The calculation of Equation 15 confirms the initial statement that the maximum noise power spectral density that can be extracted from a noisy source is kT (in W/Hz). This result is independent of the actual value of the source resistance R_s.

We can further generalize the thermal noise of any impedance as

$$\overline{V_\mathrm{n}^2} = 4kT\Re\{Z\} \tag{16}$$

as for example in the complex impedance Z_ant of an antenna. Since an antenna is a reciprocal device, if we measure its radiation impedance Z_rad (for example with a vector network analyzer), we can calculate its thermal noise with Equation 16 to $\overline{V_\mathrm{n}^2} = 4kT\Re\{Z_\mathrm{rad}\}$.

2.3.3 Noise Figure

In RF systems, we often want to quantify the noise performance of a circuit block or a complete system. The most widely used metric is the **noise factor (F)**, which is defined as the ratio of the signal-to-noise ratio (SNR) at the input to the SNR at the output of a circuit block or system. If we express the noise factor in dB, we call it the **noise figure (NF)** [3]. The noise factor is given by

$$F = \frac{\mathrm{SNR}_\mathrm{in}}{\mathrm{SNR}_\mathrm{out}} = \frac{\left(P_\mathrm{s}/P_\mathrm{n}\right)_\mathrm{in}}{\left(P_\mathrm{s}/P_\mathrm{n}\right)_\mathrm{out}} \tag{17}$$

where P_s is the signal power and P_n is the noise power. The noise factor is always larger than or equal to 1 (or 0 dB), as no circuit can improve the SNR!

! SNR Improvement

Note that the SNR can be improved by filtering, as filtering reduces the noise power. If the noise bandwidth is larger than the signal bandwidth, then the SNR can be improved without affecting the signal. However, this is not considered in the noise factor, as the noise factor assumes that both signal and noise pass through the same bandwidth.

Let us look at a simple model of a noise circuit block as shown in Figure 11. The input signal S_{in} is accompanied by noise N_{in}. By definition it is assumed that the input noise power results from a matched resistor at $T_0 = 290\,\text{K}$, so that $N_{\text{in}} = kT_0$. The circuit block has a power gain G and adds its own noise N_{dut} to the output signal. For simplicity, we assume that the input and output of the circuit block are **impedance matched** to avoid reflections.

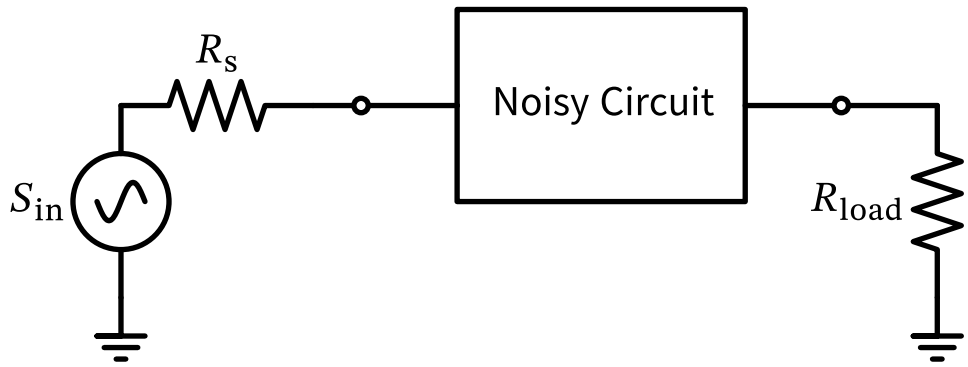

Figure 11: A noise-matched system with source and load impedances and a noisy circuit block.

The output signal and noise powers are then given by

$$S_{\text{out}} = GS_{\text{in}}$$

$$N_{\text{out}} = GN_{\text{in}} + N_{\text{dut}}$$

The resulting noise factor can then be calculated as

$$F = \frac{S_{\text{in}}/N_{\text{in}}}{S_{\text{out}}/N_{\text{out}}} = \frac{1}{G}\frac{GN_{\text{in}} + N_{\text{dut}}}{N_{\text{in}}} = 1 + \frac{N_{\text{dut}}}{GN_{\text{in}}},$$

in other words, the noise factor is 1 plus the ratio of the noise added by the device under test (DUT) to the amplified input noise.

Note that a noiseless block ($N_{\text{dut}} = 0$) has a noise factor of $F = 1$. A passive block with loss factor L (and impedance matched at input and output) has a noise factor of $F = L$ (in linear units), as it attenuates the signal and $N_{\text{out}} = N_{\text{in}} = kT$ if everything is in thermal equilibrium.

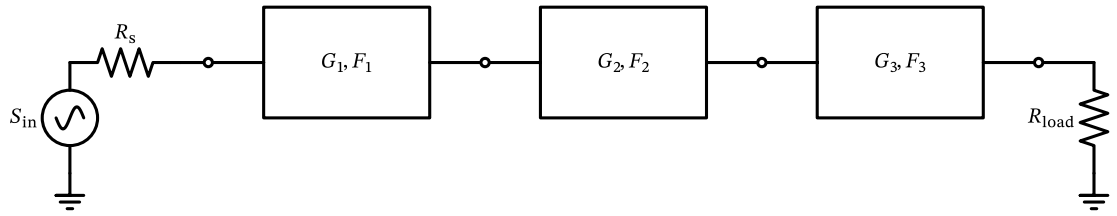

Figure 12: Block cascade for noise factor calculation showing multiple stages with gains and individual noise factors.

If we have a cascade of multiple blocks, as shown in Figure 12, we can calculate the overall noise factor with the **Friis formula** [3]

$$F_{\text{total}} = 1 + (F_1 - 1) + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} \tag{18}$$

where F_i and G_i are the noise factor and power gain of the i-th block, respectively. Note that all gains have to be in linear units (not dB) when using Equation 18. We can interpret Equation 18 as follows:

- The overall noise factor F_{total} is always larger than or equal to the noise factor of the first block (F_1).
- The noise factor of the first block is the most important one, as the noise factors of the following blocks are reduced by the gain of all preceding blocks. This is especially important in RF receivers, where the first block is usually a low-noise amplifier (LNA) with a very low noise figure (e.g., 1 dB or less) and a high gain (e.g., 10 dB or more). This ensures that the noise of the following blocks is negligible.
- The noise factor of the last block is reduced by the gain of all preceding blocks, so it is usually not very important.

Here we also see a trade-off between noise and linearity, as shown by Equation 13 and Equation 18. For low noise, we should try to maximize G_1, however, this will affect linearity (IIP3) in a negative way. As in many other situation in RF design, we have to find a good compromise between conflicting requirements.

2.3.4 Sensitivity

In RF receivers, we often want to know the minimum input signal power that can be detected with a certain SNR. This minimum input signal power is called the **sensitivity** of the receiver. The sensitivity can be calculated as

$$P_{\text{in,min}} = P_{\text{n}} \cdot \text{SNR}_{\text{min}} \cdot F \tag{19}$$

where P_{n} is the noise power at the input, SNR_{min} is the minimum detectable SNR, and F is the noise factor of the receiver. The input noise power can be calculated as

$$P_{\text{n}} = kTB$$

where k is the Boltzmann constant, T is the temperature in Kelvin, and B is the bandwidth of the receiver. Expressing Equation 19 in dBm we get the following formula:

$$P_{\text{in,min}}|_{\text{dBm}} = -174\,\text{dBm/Hz} + \text{NF} + 10\log_{10}(B/\ \text{Hz}) + \text{SNR}_{\text{min}}|_{\text{dB}} \tag{20}$$

where −174 dBm/Hz is the thermal noise PSD at room temperature (290 K). We can see that the sensitivity improves with lower noise figure, smaller bandwidth, and lower minimum detectable SNR.

Note 5: Sensitivity Calculation for WiFi

Let's calculate the sensitivity of a WiFi receiver operating at 5 GHz with a bandwidth of $B = 80\,\text{MHz}$, a noise figure of $NF = 7\,\text{dB}$, and a minimum detectable SNR of 25 dB. This high SNR means that a high-order modulation scheme (like 64-QAM) is used for high data rates.

Using Equation 20 we get:

$$P_{\text{in,min}} = -174\,\text{dBm/Hz} + 7\,\text{dB} + 10\log_{10}(80 \times 10^6) + 25\,\text{dB} \approx -63\,\text{dBm}$$

This means that the minimum input signal power that can be detected by the WiFi receiver is approximately −63 dBm.

2.4 Modulation

In order to transmit information via an EM wave, we need to modulate the EM wave with the information signal. Looking at a simple sinusoidal carrier wave

$$s(t) = A\cos(\omega_0 t + \varphi)$$

we see that we can change one or more of the following parameters to encode information:

- Amplitude $A(t)$ (amplitude modulation, AM; the digital form is called amplitude-shift keying, **ASK**)
- Frequency $\omega_0(t)$ (frequency modulation, FM; the digital form is called frequency-shift keying, **FSK**)
- Phase $\varphi(t)$ (phase modulation, PM; the digital form is called phase-shift keying, **PSK**)
- Amplitude $A(t)$ and phase $\varphi(t)$ (quadrature amplitude modulation, **QAM**)

The modulation formats FM and PM have the advantage that the carrier amplitude is constant, which makes them more robust against nonlinear distortion.

QAM is widely used in modern communication systems, as it allows to transmit more bits per symbol by combining amplitude and phase modulation. The form with 4 different symbols is called QPSK. Higher-order modulation like 16-QAM, for example, uses 16 different symbols, which can encode 4 bits per symbol (as $2^4 = 16$). Even higher-order QAM formats like 64-QAM (6 bits per symbol), 256-QAM (8 bits per symbol), 1024-QAM (10 bits per symbol), or 4096-QAM (12 bits per symbol) are also used in modern systems like WiFi or LTE.

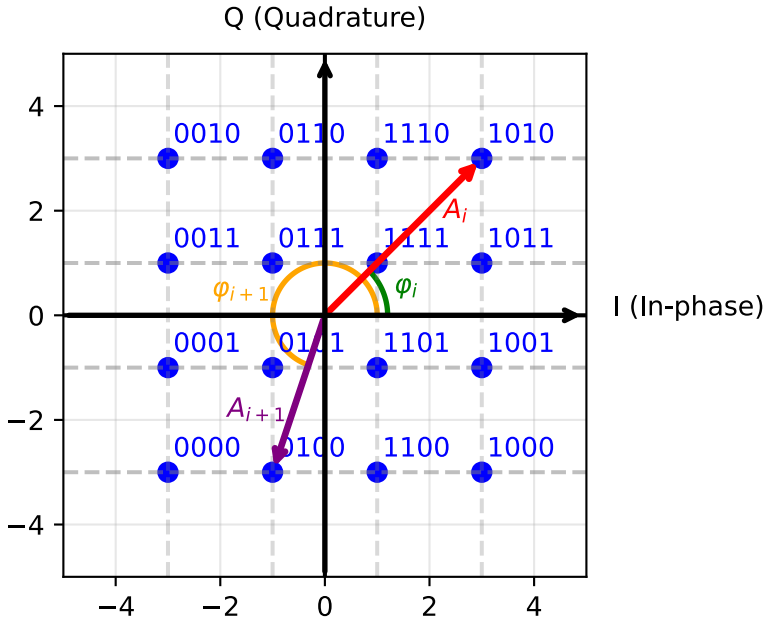

Figure 13: 16-QAM constellation diagram with Gray code labeling of constellation points.

Shown in Figure 13 is the "constellation diagram" of a 16-QAM modulation format. The constellation points are arranged in a square grid, with each point representing a unique combination of amplitude and phase. The distance between the constellation points determines the robustness against noise and interference; larger distances result in better performance, but also require more bandwidth. The mapping of bits to constellation points is called "bit mapping" or "symbol mapping". The example in Figure 13 uses a Gray code mapping, which minimizes the number of bit errors in case of a symbol error.

The constellation diagram can be imagined as a complex plane, where the x-axis represents the in-phase component (I) and the y-axis represents the quadrature component (Q) of the modulated signal. During transmission of a specific symbol, the RF carrier is modulated to the corresponding amplitude and phase, resulting in a specific point in the constellation diagram. In Figure 13, the amplitude and phase information for two consecutive symbols, A_i/φ_i and A_{i+1}/φ_{i+1}, is shown. If we have a bitrate with a bit duration of T_b, the symbol duration for 16-QAM (4 bits per symbol) is $T_\mathrm{s} = 4T_\mathrm{b}$. During the first T_s, the carrier is modulated to A_i/φ_i, and during the next T_s, it is modulated to A_{i+1}/φ_{i+1}.

The table below shows the **SNR requirements for different modulation formats** to achieve a bit error rate (BER) of 10^{-5} in an additive white Gaussian noise (AWGN) channel. As we can see, higher-order modulation formats require higher SNR to achieve the same BER. Note that for the SNR values of this table **no error correction coding** is assumed; with error correction coding the required SNR can be significantly reduced!

Table 3: SNR requirements for different modulation schemes to achieve BER = 10^{-5} in AWGN channel

Modulation	Bits/Symbol	Required SNR (dB)
BPSK	1	9.6
QPSK	2	12.6
16-QAM	4	18.2
64-QAM	6	24.4
256-QAM	8	30.6
1024-QAM	10	36.9
4096-QAM	12	43.2

2.5 Pulse Shaping and Spectral Efficiency

When we modulate symbols onto a carrier, we usually do not transmit the symbols as pure sinusoids, but rather as pulses with a certain shape. The pulse shape determines the bandwidth of the transmitted signal and its spectral efficiency. A common pulse shape is the **rectangular pulse**, which has a sinc-shaped spectrum. However, the sinc function $\sin(\pi x)/\pi x$ has side lobes that extend to infinity, which can cause interference with adjacent channels.

For reference, the spectrum of a random binary sequence with equal probability of 0s and 1s, using rectangular pulses with a duration of T_b is given by ($S(f)$ is the two-sided power spectral density):

$$S(f) = \frac{T_\mathrm{b}}{4}\mathrm{sinc}^2(fT_\mathrm{b}) + \frac{1}{4}\delta(f) = \frac{T_\mathrm{b}}{4}\left(\frac{\sin(\pi f T_\mathrm{b})}{\pi f T_\mathrm{b}}\right)^2 + \frac{1}{4}\delta(f)$$

To avoid this, we can use pulse shapes that have better spectral properties, such as the raised cosine pulse or the root-raised cosine pulse. The **raised-cosine pulse** has a roll-off factor α that determines the excess bandwidth beyond the Nyquist bandwidth. The root-raised cosine (RRC) pulse is used in practical systems (with half the pulse filter implemented at the TX, and half at the RX), as it can be implemented with a matched filter at the receiver.

The raised-cosine pulse $p(t)$ (with a spectrum shaped like a raised cosine) is given by:

$$p(t) = \frac{\sin(\pi t/T_\mathrm{b})}{\pi t/T_\mathrm{b}} \cdot \frac{\cos(\alpha\pi t/T_\mathrm{b})}{1-(2\alpha t/T_\mathrm{b})^2}$$

Setting $\alpha = 0$ results in a sinc pulse in the time domain (with a perfect bandwidth containment in the frequency domain), while $\alpha = 1$ results in a pulse with double the Nyquist bandwidth. The pulse shape for $\alpha = 0$ and $\alpha = 0.22$ (used in 3G) is shown in Figure 14.

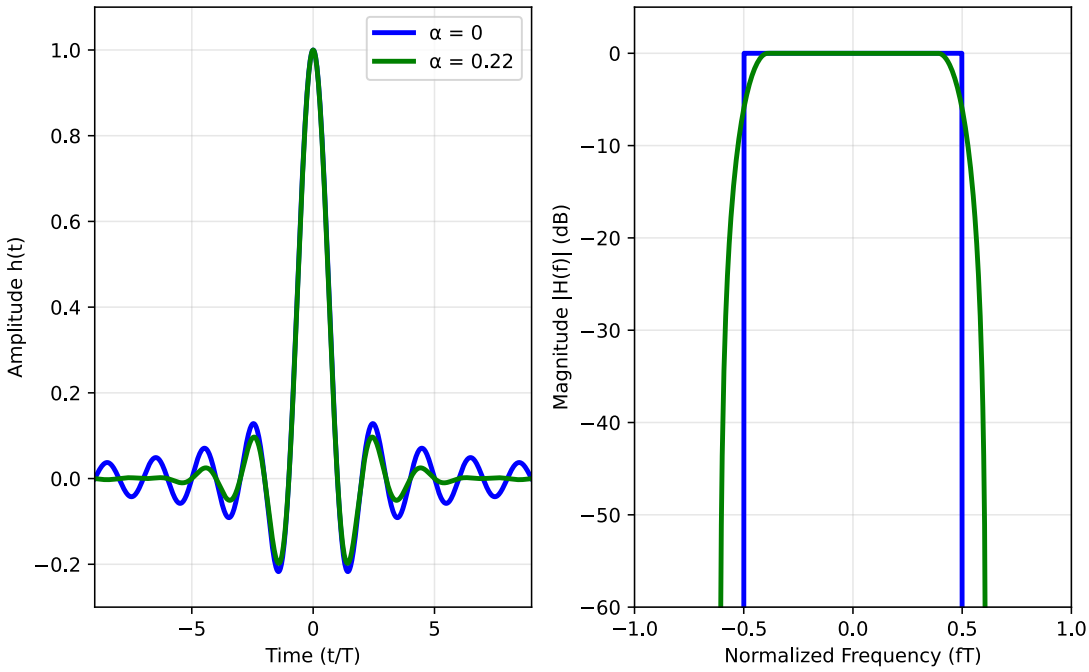

Figure 14: Raised cosine pulse shaping in time and frequency domain for different roll-off factors α.

Another often-used pulse shape is the **Gaussian pulse**, which is used in Gaussian minimum-shift keying (**GMSK**, used in 2G) modulation, or in Gaussian frequency-shift keying (**GFSK**, used in Bluetooth). The Gaussian pulse has a smooth shape and a narrow spectrum. The Gaussian pulse is given by:

$$p(t) = \frac{\sqrt{\pi}}{\alpha} e^{-(\pi t/\alpha)^2} \quad \text{with} \quad \alpha = \frac{\sqrt{\ln 2}}{\sqrt{2}} \cdot \frac{T_\mathrm{b}}{BT_\mathrm{b}}$$

where BT_b controls the width of the pulse. The spectrum of the Gaussian pulse is also Gaussian-shaped, which helps to minimize inter-symbol interference (ISI).

The Gaussian pulse for $BT = 0.5$ as used in Bluetooth is shown in Figure 15.

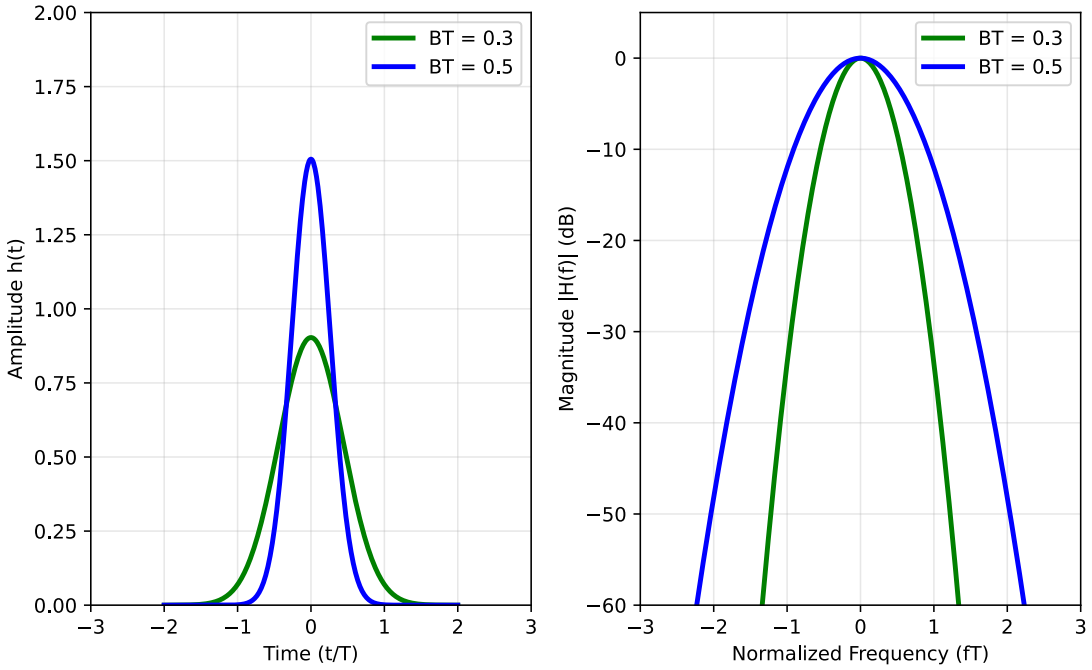

Figure 15: Gaussian pulse shaping in time and frequency domain for different bandwidth-time products BT.

For both the raised-cosine and Gaussian pulse, the trade-off between time- and frequency-domain containment is clearly visible. This is also captured in "**Küpfmüller's uncertainty principle**", which states that the product of the time duration and the bandwidth of a pulse is lower-bounded by a constant. In other words, if we want to have a pulse that is very short in time, it will have a wide bandwidth, and vice versa.

2.6 Orthogonal Frequency-Division Multiplexing (OFDM)

As we have seen in the previous section, if we make the symbol rate high, we need to use pulses with a wide bandwidth. The problem with a wide bandwidth in wireless communication is **multi-path propagation**, which causes frequency-selective fading. This means that some frequencies are attenuated more than others, which can cause errors in the received signal. Equalizing such a frequency-selective channel can be very complex, especially if the channel changes rapidly (as in mobile communication). We now face a dilemma: How can we achieve high data rates (which require high symbol rates and thus wide bandwidth) while avoiding frequency-selective fading? The key idea, implemented in **OFDM**, is to split the wideband channel into multiple narrowband sub-channels (subcarriers), each with a low symbol rate. This way, each subcarrier experiences flat fading, which is much easier to equalize.

The key question is now how to implement this idea efficiently, as we now have to apply modulation to hundreds or thousands of individual subcarriers. The solution is to use the **inverse fast Fourier transform (IFFT)** at the transmitter to generate the time-domain OFDM signal from the frequency-domain symbols, and the **fast Fourier transform (FFT)** at the receiver to recover the frequency-domain symbols from the time-domain OFDM signal. This is illustrated in Figure 16.

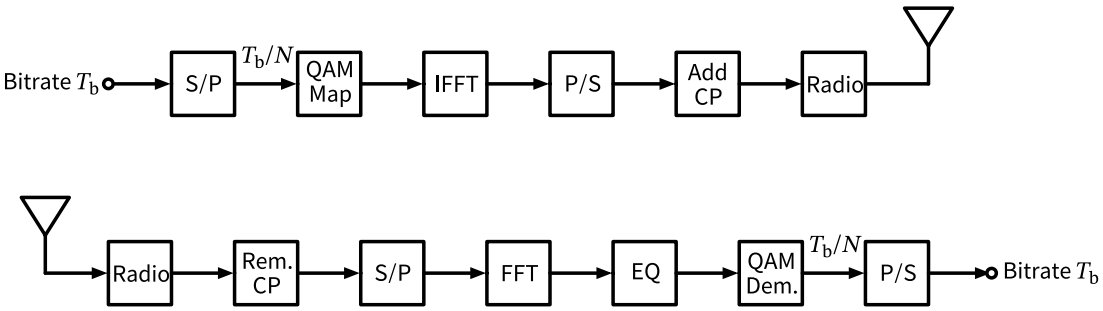

Figure 16: OFDM transmission system block diagram showing transmitter and receiver processing chains.

The OFDM transmitter takes a block of N symbols (e.g., 64-QAM symbols) and maps them onto N subcarriers, thereby reducing the symbol rate for each subcarrier to T_b/N. The IFFT then generates the time-domain OFDM signal, which is transmitted over the wireless channel. Before transmission the **cyclic prefix (CP)** is added to each OFDM symbol.

At the receiver, first the CP is removed, and then the FFT recovers the frequency-domain symbols, which can then be equalized (fairly simply by multiplying each subcarrier with a complex factor to correct amplitude and phase) and demodulated.

A key property of OFDM is that the subcarriers are **orthogonal** to each other, which means that they do not interfere with each other, even if they overlap in frequency. This is achieved by choosing the subcarrier spacing Δf such that it is equal to the symbol rate $1/T_\mathrm{b}$, i.e., $\Delta f = 1/T_\mathrm{b}$. This way, the integral of the product of two different subcarriers over one symbol period is zero, which means that they are orthogonal.

To further improve the robustness against multi-path propagation, a CP is added to each OFDM symbol. The CP is a copy of the last part of the OFDM symbol, which is added to the beginning of the symbol. This way, if there are delayed copies of the OFDM symbol due to multi-path propagation, they will still fall within the CP and will not cause inter-symbol interference (ISI). The length of the CP should be longer than the maximum delay spread of the channel.

Note 6: OFDM in LTE

In LTE OFDM is used for the downlink (base station to user equipment) with the following parameters:

- Subcarrier spacing: 15 kHz
- CP length: 5.2 µs (normal), 4.7 µs (extended)
- Number of subcarriers: 1200 (for 20 MHz bandwidth)
- Modulation: QPSK, 16-QAM, 64-QAM, 256-QAM

From the subcarrier spacing we can calculate the symbol duration as $T_\mathrm{b} = 1/\Delta f = 1/15\,\mathrm{kHz} \approx 66.7\,\mu\mathrm{s}$.

We can calculate the raw bitrate for a 20 MHz LTE channel as

$$\mathrm{Bitrate} = N_\mathrm{sc} \cdot N_\mathrm{sym} \cdot \frac{1}{T_\mathrm{b} + T_\mathrm{CP}} = 1200 \cdot 8 \cdot \frac{1}{66.7\,\mu\mathrm{s} + 5.2\,\mu\mathrm{s}} \approx 133\,\mathrm{Mbps}$$

Without the overhead for control channels and error correction coding a user datarate of approximately 100 Mbps can be achieved in a 20 MHz LTE channel.

2.7 Multiple Access Techniques

In wireless communication systems, multiple users need to share the same frequency spectrum. This is achieved by using **multiple access techniques**, which allow multiple users to transmit and receive data simultaneously without interfering with each other. The most common multiple access techniques are:

1. **Time division multiple access (TDMA)**: Users are assigned specific time slots for transmission, allowing multiple users to share the same frequency channel by dividing the time into slots.
2. **Frequency division multiple access (FDMA)**: Users are assigned specific frequency bands within the overall frequency spectrum, allowing multiple users to transmit simultaneously on different frequencies.
3. **Code division multiple access (CDMA)**: Users are assigned unique spreading codes, allowing them to transmit simultaneously over the same frequency band. The receiver uses the code to extract the desired signal. A variant of CDMA is frequency-hopping spread spectrum (FHSS), where the carrier frequency is changed rapidly according to a pseudo-random sequence known to both the transmitter and receiver. This is used in Bluetooth.
4. **Orthogonal frequency division multiple access (OFDMA)**: A variant of OFDM, where multiple users are assigned different subcarriers for transmission, allowing for efficient use of the frequency spectrum. This is used in 4G LTE and 5G NR.
5. **Spatial division multiple access (SDMA)**: Uses multiple antennas to create spatially separated channels, allowing multiple users to transmit simultaneously in the same frequency band.

In addition, all of these techniques can be combined to create more efficient and flexible communication systems. For example, OFDMA can be used in conjunction with SDMA to

allow multiple users to share the same frequency resources while also taking advantage of spatial diversity. Also, TDMA can be combined with FDMA to create a hybrid multiple access scheme (which has been used in 2G GSM).

3 Transceivers

Nowadays, the various small-signal RF functions for receive and transmit are integrated into so-called transceivers (TRX). A TRX is a device that can both transmit and receive signals, and is usually called an "RFIC". While high monolithic integration is certainly the norm for radio-frequency devices intended for standards like Bluetooth, WiFi, cellular, etc., it is increasingly used also for mm-wave frequencies for applications like automotive radar and 5G cellular.

Typically TRX include components like amplifiers, mixers, filters, oscillators, and phase-locked loops. When digital interfaces are used for the baseband data transport also functions like analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) are integrated together with digital signal processing (DSP) blocks and potentially high-speed interfaces.

In this lecture we will focus on the RF part of a TRX, which is responsible for the upconversion of baseband or intermediate frequency (IF) signals to the desired transmit frequency during transmission, and the downconversion of received signals from the carrier frequency to baseband or IF during reception. For filters, low-frequency amplifiers, ADCs, DACs, and DSP blocks we refer to related courses and literature, for example our analog circuit design course.

3.1 Direct-Conversion Transceiver

The following typical functions have to be performed by a TRX:

- Pulse-shaping filtering of the baseband signal (can be implemented analog or in most cases digital).
- Modulating the baseband signal onto a carrier frequency (upconversion) in the TX or downconversion in the RX.
- Contain the RF signal in a small bandwidth (TX), or single out the wanted signal in the RX.
- Adapt gain (and linearity) to the signal strength in the RX, and to the output power in the TX.
- Generate the carrier frequency (local oscillator, LO) with low phase noise.

The dominant architecture for the TRX is the so-called direct-conversion (or Zero-IF) architecture, where the upconversion and downconversion is performed in a single step. This is in contrast to superheterodyne architectures, where the signal is first converted to an intermediate frequency (IF) before being converted to baseband. The direct-conversion architecture has the advantage of reduced complexity and cost, as it requires fewer components and less filtering. However, it also has some disadvantages, such as increased susceptibility to DC offsets and I/Q imbalance. A typical TRX block diagram is shown in Figure Figure 17.

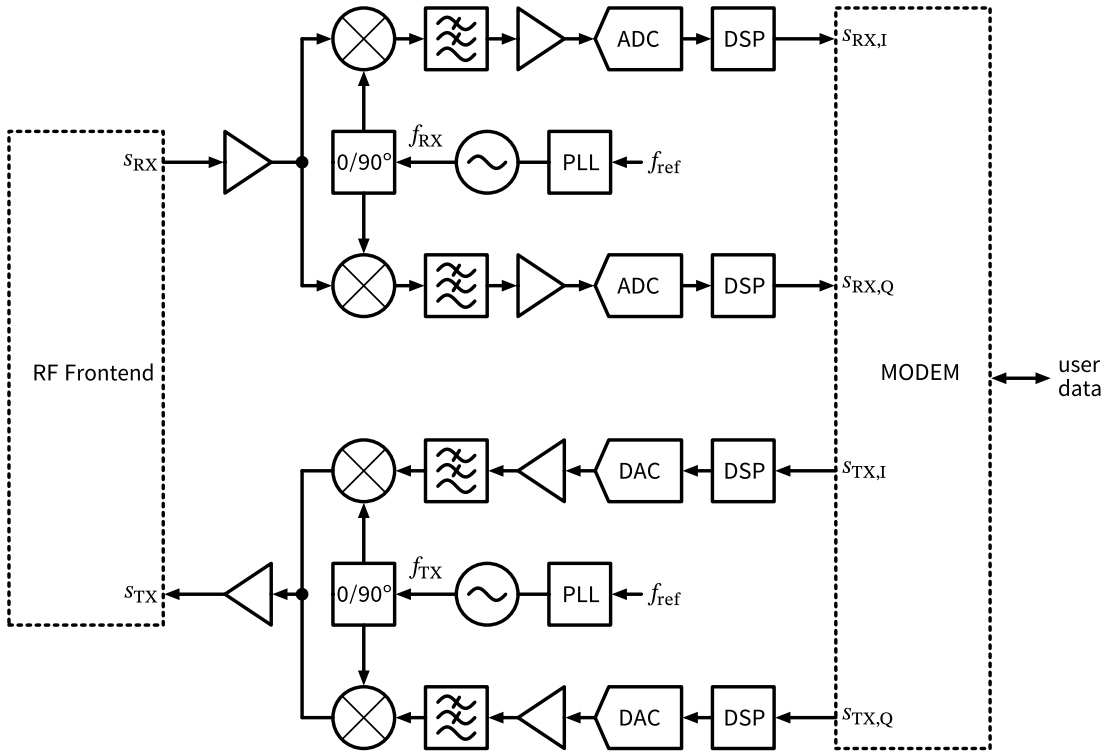

Figure 17: Block diagram of a typical transceiver (TRX) showing the main functional blocks of RX and TX. The modem provides the digital baseband processing and interfaces to the rest of the system. For implementation options of the RF front-end, see Section 3.5.

As can be seen in Figure 17, this generic example can be adapted in various ways. Generally, the amplifier gains are adjustable to adapt to different signal levels. If various channel bandwidths are to be supported, the corner frequencies of the low-pass filters (LPF) can be adjusted, as well as (optionally) the sampling rate of the ADCs and DACs. The local oscillator (LO) frequency is generated by a phase-locked loop (PLL) synthesizer, which can be tuned to the desired carrier frequency. In case of frequency-division duplex (FDD) operation, two PLLs are used to generate the TX and RX LO frequencies, which are separated by the duplex distance. In time-division duplex (TDD) operation, a single PLL is sufficient, supplying the LO signal to both RX and TX.

The modem that is shown in Figure 17 is responsible for the digital baseband processing, including functions like channel coding/decoding, modulation/demodulation, equalization, and error correction. The modem is usually implemented as a digital System-on-Chip (SoC) consisting of (multiple) CPUS, DSPs, and fixed-function blocks for time-critical processing. For an in-depth discussion we recommend [7] or [8].

3.2 Modulation and Demodulation

Modulation is the process of varying a carrier signal at frequency f_c in order to transmit information. The complex baseband signal (after converting the real-valued digital s_I and s_Q signals to analog and pulse-shaping filtering) is represented as ($\tilde{s}_{BB} \in \mathbb{C}$; $s_I, s_Q \in \mathbb{R}$):

$$\tilde{s}_{BB}(t) = s_I(t) + j \cdot s_Q(t).$$

We want to shift this signal to the carrier frequency f_c, which can be done by multiplying with a complex exponential ($e^{j\omega_\mathrm{c}t}$, with $\omega_\mathrm{c} = 2\pi f_\mathrm{c}$):

$$\tilde{s}_\mathrm{RF}(t) = \tilde{s}_\mathrm{BB}(t) \cdot e^{j\omega_\mathrm{c}t} = \big[s_\mathrm{I}(t) + j \cdot s_\mathrm{Q}(t)\big] \cdot [\cos(\omega_\mathrm{c}t) + j \cdot \sin(\omega_\mathrm{c}t)].$$

The real-valued RF signal is obtained by taking the real part of this expression ($s_\mathrm{RF} \in \mathbb{R}$, $\tilde{s}_\mathrm{RF} \in \mathbb{C}$):

$$s_\mathrm{RF}(t) = \Re\{\tilde{s}_\mathrm{RF}(t)\} = s_\mathrm{I}(t)\cos(\omega_\mathrm{c}t) - s_\mathrm{Q}(t)\sin(\omega_\mathrm{c}t). \tag{21}$$

The process formulated in Equation 21 is done in the TX, as shown in Figure 18.

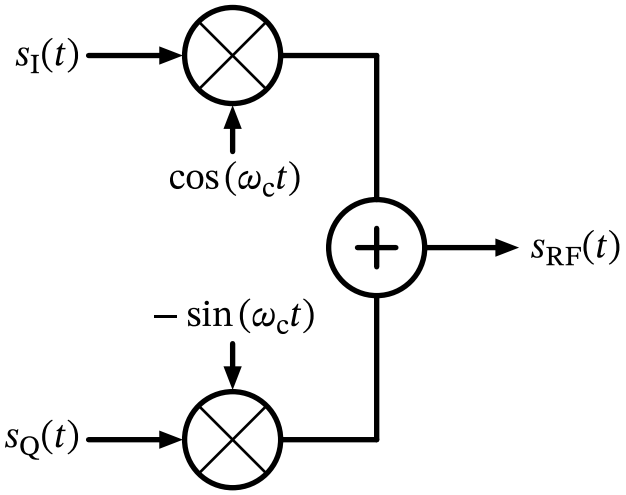

Figure 18: TX modulator.

The RF signal generation according to Equation 21 is called quadrature modulation. This is the modulation used most often in modern communication systems, as it allows to transmit two independent signals (I and Q) in the same bandwidth. The I and Q signals are also called quadrature components, as they are 90° out of phase with each other.

Alternatively, a modulation called polar modulation can be used, where the amplitude and phase of the carrier are varied according to the baseband signal. This is done by converting the I and Q signals to polar coordinates

$$s_\mathrm{RF}(t) = \Re\big\{A(t) \cdot e^{j\varphi(t)} \cdot e^{j\omega_\mathrm{c}t}\big\}$$

with

$$A(t) = \sqrt{s_\mathrm{I}^2(t) + s_\mathrm{Q}^2(t)}, \quad \varphi(t) = \tan^{-1}\left[\frac{s_\mathrm{I}(t)}{s_\mathrm{Q}(t)}\right].$$

As the mathematical operations required for the cartesian to polar transformation are quite nonlinear, the $A(t)$ and $\phi(t)$ signals are wideband. Some wireless standards allow efficient use of polar modulation, for example Bluetooth, where basically all TX are realized as polar modulators.

In the RX, the received RF signal is downconverted to baseband by a similar process, as shown in Figure 19.

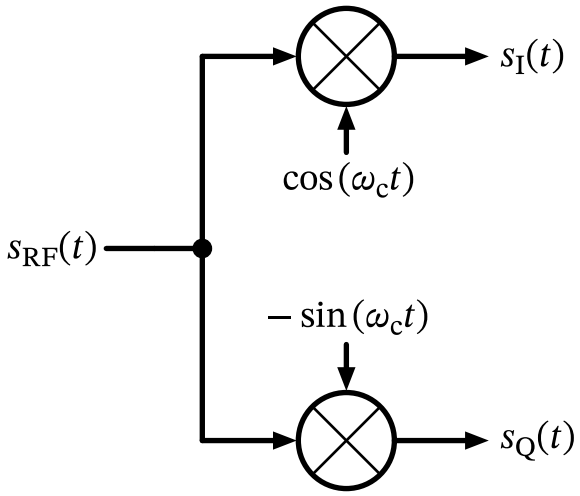

Figure 19: RX demodulator.

For demodulation we have to shift the RF signal down to baseband, which mathematically is done by multiplying with the complex conjugate of the carrier ($e^{-j\omega_{\mathrm{c}}t}$)

$$\tilde{s}_{\mathrm{BB}}(t) = s_{\mathrm{RF}}(t) \cdot e^{-j\omega_{\mathrm{c}}t} = s_{\mathrm{RF}}(t) \cdot [\cos(\omega_{\mathrm{c}}t) - j \cdot \sin(\omega_{\mathrm{c}}t)] \tag{22}$$

3.3 Filtering

Filtering is an essential function in both TX and RX. In the TX, filtering is used to limit the bandwidth of the transmitted signal to the allocated channel bandwidth, and to suppress out-of-band emissions. In the RX, filtering is used to select the wanted signal from a crowded spectrum, and to suppress unwanted signals (blockers) that can cause interference or desensitization of the RX. A typical example of filtering in the RX is shown in Figure 20, where a bandpass filter is used to attenuate strong unwanted blockers while only slightly attenuating the wanted signal.

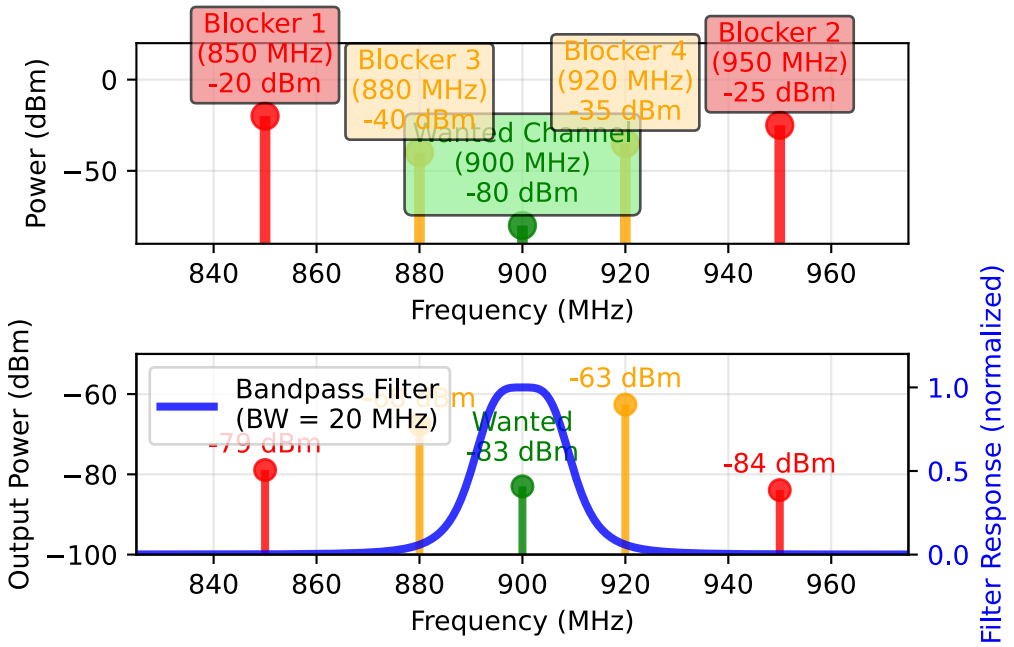

Figure 20: Filtering of wanted channel amid strong unwanted blockers. Exemplary shown in an RX scenario around 900 MHz. The strong blockers (top figure) are attenuated by an RF bandpass filter (bottom figure) with a bandwidth of 20 MHz, achieving more than 40 dB rejection of the blockers while only slightly attenuating the wanted signal.

In any filter there exists a fundamental trade-off between selectivity (steepness of the filter skirts), bandwidth, and insertion loss. A very selective filter with steep skirts and large BW will have a high insertion loss. Conversely, a filter with low insertion loss will have a gentle roll-off and may not sufficiently suppress unwanted signals. A useful metric to quantify the performance of a filter is the quality factor Q, defined as

$$Q = \frac{f_c}{\Delta f}$$

where f_c is the center frequency and Δf is the −3 dB bandwidth of the filter. A higher Q indicates a more selective filter.

The achievable Q depends on the filter technology used. For example, on-chip LC filters can achieve Q values of around 10-20, while off-chip SAW or BAW/FBAR filters can achieve Q values of several hundreds, and a crystal filter can achieve Q values of several thousands. The choice of filter technology depends on the application requirements, such as frequency range, bandwidth, insertion loss, and cost. Generally speaking, the required filtering to single out the wanted signal in the RX spectrum and decrease the power of strong blockers to a tolerable level is one of the most critical design choices, and is usually distributed at different locations in the RX chain:

- **RF filters** (between antenna and LNA) provide a first level of filtering, and are usually implemented as off-chip SAW or BAW/FBAR filters. They provide high Q and good selectivity, but have a fixed center frequency and bandwidth. They are used to pass the wanted band of interest, and to attenuate strong out-of-band blockers.

- **IF filters** (in case of a super-heterodyne receiver) provide additional filtering, and can be implemented as on-chip LC filters or off-chip SAW/BAW filters. They provide moderate Q and selectivity, and can be tuned to some extent.
- **BB filters** (after downconversion) provide the final level of filtering before entering the ADCs, and are usually implemented as on-chip active RC filters. They provide channel selection, and can be easily adjusted to different bandwidths.
- **Digital filters** (in the DSP block) provide the final level of filtering and signal processing, and can be implemented as FIR or IIR filters. They provide high flexibility and can be easily adapted to different standards and requirements. Digital filters show now variations, so they can be designed to be very selective.

It is important to note (because this dictates a lot of choices in RF design) that high-Q filters are usually fixed-frequency and fixed-bandwidth. Only baseband and digital filters can be easily adjusted to different bandwidths!

! Filter Technologies

Baseband filters (analog) are usually implemented as active RC filters on-chip. They are very flexible and can have adjustable bandwidth by either changing R and/or C. For medium frequencies $g_\mathrm{m} - C$ filters can be used, which are also tunable by changing the transconductance g_m and/or C. For even higher bandwidths, on-chip LC filters can be used, which have a limited Q of around 10-20.

Baseband filters (digital) are implemented as FIR or IIR filters in the DSP block. They are very flexible and can be easily adapted to different standards and requirements. Digital filters show no variations, so they can be designed to be very selective.

Surface acoustic wave (SAW) and **bulk acoustic wave (BAW/FBAR)** filters are off-chip components that can achieve high Q values of several hundreds. They have a fixed center frequency and bandwidth. Usually 1-2 such filters are required per supported band of interest.

Crystal filters can achieve very high Q values of several thousands, but are usually bulky and expensive.

LC filters can be either implemented off-chip (using discrete components) or on-chip. Off-chip LC filters can achieve higher Q values than on-chip LC filters, but are usually larger and more expensive. On-chip LC filters are limited in Q (around 10-20), but are very compact and can be integrated into the RFIC. Off-chip LC filters can achieve Q values of around 50-100, depending on the frequency and component quality.

Ceramic filters are another off-chip filter technology that can achieve moderate to high Q values (up to several hundreds). They are usually smaller and less expensive than SAW or BAW/FBAR filters, but also lower performance.

Waveguide filters are used at very high frequencies (above 10 GHz) and can achieve very high Q values (up to several thousands). They are usually bulky and expensive, and are not commonly used in mobile applications, but rather in fixed installations like base stations or satellite communication.

Fundamentally, the choice of filter technology is a trade-off between performance, size, cost, and flexibility. In most cases, a combination of different filter technologies is used to achieve the desired performance.

3.4 Direct-Conversion Architecture

The transceiver architecture shown in Figure 17 is called direct-conversion or zero-IF architecture, as the downconversion in the RX and upconversion in the TX is done in a single step. This architecture has several advantages:

- Per RX and TX a single LO is required (which can even be shared between RX and TX in TDD operation).
- There are a minimum number of RF blocks, which is good for cost and power consumption.
- This architecture is very flexible and can be easily adapted to different standards and requirements, and generally shows very good performance if the disadvantages can be overcome by good design.
- This architecture allows a high integration level, as basically all blocks can be implemented on-chip.
- Direct conversion is the de-facto standard architecture for cellular, WiFi, Bluetooth (with the exception of the TX), and GNSS.

However, the direct-conversion architecture also has some disadvantages:

- LO-RF coupling can cause self-mixing and desensitization of the RX, as well as LO leakage in the TX. This is an issue because the LO frequency is the same as the RF frequency.
- Even-order distortion products (especially IIP2) cause sensitivity degradation due to strong amplitude-modulated blockers.
- LO pulling can occur in the TX (again, LO and RF are at the same frequency).
- IQ errors (gain and phase mismatch) of the I and Q paths can cause constellation distortion leading to increased error vector magnitude (EVM).
- DC offsets can occur due to self-mixing of LO leakage and even-order distortion products.
- Flicker noise (1/f noise) upconversion can cause increased phase noise close to the carrier, as well as increased RX noise figure.

Nowadays there exist good design techniques to mitigate these disadvantages. However, in some cases (for example very high linearity requirements, or very high frequencies) other architectures like low-IF or super-heterodyne may be preferred.

3.5 Duplexing

In the block diagram of Figure 17, we have not yet considered how to share the antenna between RX and TX. Essentially, there are two main methods to achieve this: **frequency-division duplex (FDD)** and **time-division duplex (TDD)**.

3.5.1 Frequency-Division Duplex (FDD)

In FDD, the RX and TX operate at different frequencies, separated by a duplex distance. This allows simultaneous transmission and reception, which is beneficial for applications like voice communication where low latency is required. However, FDD requires two separate frequency bands, which can be a limitation in terms of spectrum availability. Additionally, FDD requires two PLLs to generate the RX and TX LO frequencies, which increases complexity and power consumption.

The RF RX and TX paths are connected to the antenna via a duplexer, which is a three-port device that allows signals to pass between the antenna and the RX or TX path, while isolating the RX and TX paths from each other. A typical FDD TRX block diagram is shown in Figure 21.

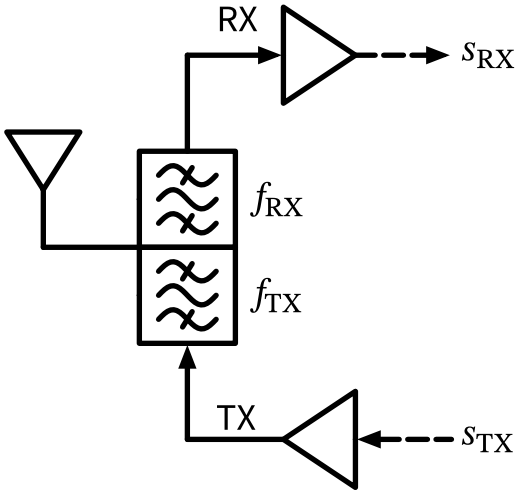

Figure 21: Block diagram of an FDD RF front-end.

Advantages of FDD:

- RX and TX can operate simultaneously, which is beneficial for low-latency applications.
- There is no need for fast switching between RX and TX, which simplifies the design.
- Relaxed synchronization requirements between RX and TX and different users.

Disadvantages of FDD:

- Duplexers are costly components with significant insertion loss depending on filtering requirements.
- Requires two separate frequency bands, which can be a limitation in terms of spectrum availability, and MIMO channel estimation.
- The strong TX causes severe desensitization of the RX, which requires high linearity and good filtering (50 dB to 60 dB).

3.5.2 Time-Division Duplex (TDD)

In TDD, the RX and TX share the same frequency band but operate at different times. This allows for more efficient use of the available spectrum, as the same frequency can be used for both transmission and reception. TDD is particularly well-suited for applications with asymmetric traffic patterns, where the data rate in one direction is significantly higher than in the other. However, TDD requires precise timing control to avoid interference between RX and TX periods, which can increase complexity. In TDD, a single PLL can be used to generate the LO frequency for both RX and TX, which reduces complexity and power consumption. The RF RX and TX paths are connected to the antenna via a switch, which alternates between connecting the antenna to the RX path and the TX path. A typical TDD TRX block diagram is shown in Figure 22.

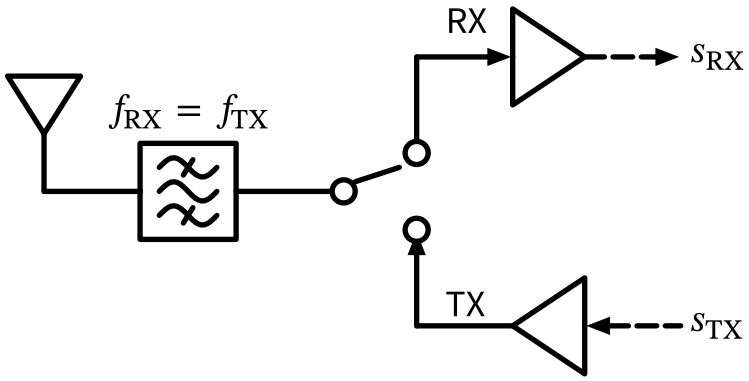

Figure 22: Block diagram of a TDD RF front-end.

Advantages of TDD:

- More efficient use of the available spectrum, as the same frequency can be used for both RX and TX.
- A single PLL can be used for both RX and TX, which reduces complexity and power consumption.
- No duplexer is required (just a single band filter), which reduces cost and insertion loss.
- No RX blocking by own TX, which relaxes linearity and filtering requirements.
- Easier to implement MIMO, as all antennas can operate in the same frequency band.

Disadvantages of TDD:

- RX and TX cannot operate simultaneously, which can be a limitation for low-latency applications.
- Requires precise timing control to avoid interference between RX and TX periods, which can increase complexity.
- Synchronization between RX and TX and different users is required, which can be challenging in some scenarios.

3.5.3 Comparison of FDD and TDD

Below is a summary of important wireless standards and their duplexing method as shown in Table 4:

Table 4: Comparison of duplexing methods used by major wireless standards

Wireless Standard	Duplexing Method	Comments
GSM (2G)	FDD & TDMA	TX and RX operate at different frequencies (FDD) and different times (TDMA)
UMTS (3G)	FDD	Traditional cellular standard using paired spectrum
LTE (4G)	FDD/TDD	FDD is used mostly <2.7 GHz, TDD is used >2.3 GHz
5G NR	FDD/TDD	FDD is used mostly <2.7 GHz, TDD is used >2.3 GHz
WiFi (802.11)	TDD	Unlicensed spectrum operation
Bluetooth	TDD	Short-range personal area network
Zigbee	TDD	Low-power IoT applications

As you can see in Table 4, there is a tendency to use FDD for lower frequencies and long communication distances, while TDD is preferred for higher frequencies and shorter distances.

3.6 Specialty Architectures

In some cases, other architectures may be preferred over the direct-conversion architecture. During the evolution of wireless communication, many different architectures have been proposed and used. However, only a few of them are still relevant today. Some examples are shown next.

3.6.1 Super-Heterodyne Architecture

The super-heterodyne architecture is a widely used approach in radio. It works by mixing the incoming/outgoing RF signal with an LO to produce an intermediate frequency (IF) signal. This IF signal is then amplified and processed, allowing for better selectivity and sensitivity compared to direct-conversion architectures. Super-heterodyne receivers/transmitters are known for their excellent performance in terms of image rejection and dynamic range, making them suitable for a variety of applications, including traditional analog TV and radio broadcasting. A simplified block diagram of a super-heterodyne transceiver is shown in Figure 23.

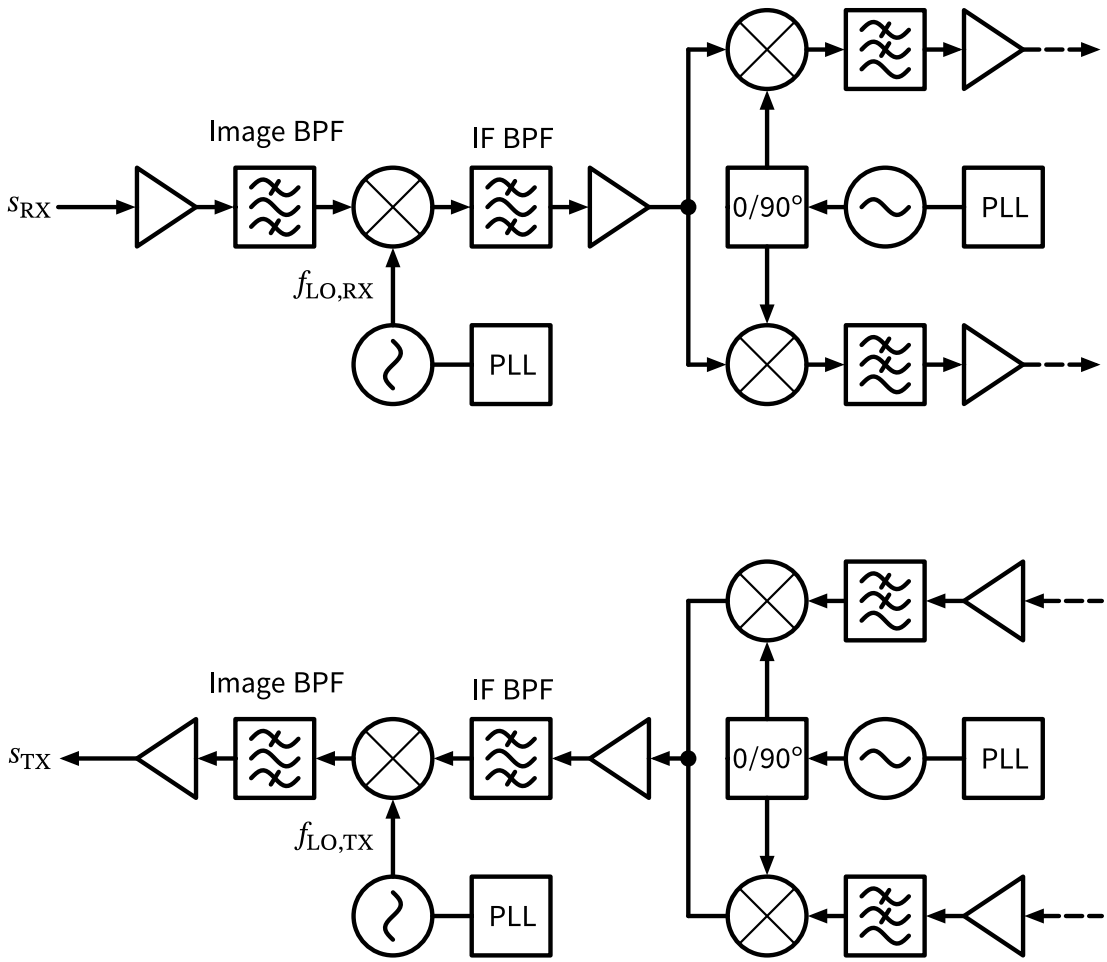

Figure 23: Block diagram of a super-heterodyne transceiver (TRX) showing the main functional blocks of RX and TX.

When you compare Figure 17 with Figure 23, you can immediately appreciate the increased complexity of the super-heterodyne architecture. It requires two PLLs to generate the RX and TX LO frequencies, as well as additional mixers and filters for the IF stage. This increases cost, power consumption, and size. However, the super-heterodyne architecture can provide better performance in terms of selectivity and sensitivity, especially in challenging RF environments with strong blockers, as it allows filtering at RF, IF, and baseband frequencies.

One important aspect of super-heterodyne receivers is the choice of the intermediate frequency (IF). The IF should be high enough to allow for effective filtering and **image rejection**, but low enough to avoid excessive complexity and power consumption. Common IF frequencies range from a few MHz to several hundred MHz, depending on the application and frequency band.

An important issue in super-heterodyne receivers is the **image frequency**. The image frequency is a spurious frequency that can interfere with the desired signal, and is located at $f_{\text{image}} = f_{\text{RF}} \pm 2f_{\text{IF}}$ (the sign depends on the choice of high-side or low-side mixing). To suppress the image frequency, an image-reject filter is either placed before (RX) or after (TX) the mixer. The design of this filter is critical, as it must provide sufficient attenuation of the image frequency while maintaining low insertion loss for the desired signal.

An alternative to image filtering is the use of active image rejection techniques, such as the **Hartley** or **Weaver** architectures. These techniques use additional mixers and phase shifters to cancel out the image frequency, allowing for improved performance without the need for a dedicated image-reject filter.

3.6.2 Low-IF Architecture

To avoid some of the issues of direct-conversion architectures (like dc offsets and flicker noise), a low-IF architecture can be used. In a low-IF architecture, the RX and TX signals are mixed to a low intermediate frequency (typically a few MHz to tens of MHz) instead of directly to baseband. This allows for easier filtering of DC offsets and flicker noise, while still maintaining the benefits of a single LO and reduced complexity compared to super-heterodyne architectures. A low-IF architecture is shown in Figure 24.

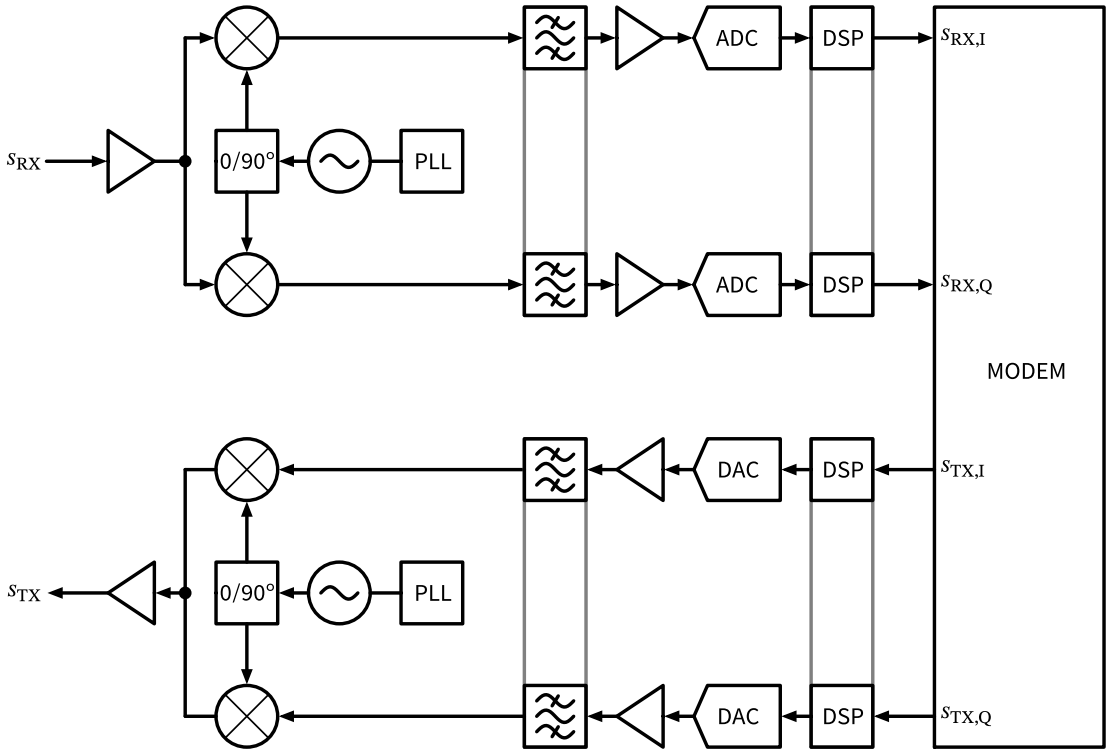

Figure 24: Block diagram of a low-IF transceiver (TRX) showing the main functional blocks of RX and TX. Note the usage of complex analog and digital baseband filters. Otherwise, the structure is similar to a zero-IF TRX as shown in Figure 17.

The low-IF architecture is the de facto standard for Bluetooth receivers. Its advantage compared to direct-conversion vanishes for larger channel bandwidths, which is why it is not used for cellular or WiFi (GSM receivers might be an exception).

One noteworthy disadvantage of low-IF architectures is the required 2xBW compared to direct-conversion. This might cause increased power consumption in the analog baseband filters and ADCs/DACs. Additionally, the low-IF architecture still requires careful design to mitigate issues like IQ imbalance and LO leakage, although these issues are generally less severe than in direct-conversion architectures.

3.6.3 Super Simple Architecture

For some applications with very low cost and low performance requirements, a super simple architecture can be used (think garage door opener). In this architecture, the RX and TX paths are stripped down to the bare minimum. A super simple receiver just uses a bandpass filter and an envelope detector, while a super simple transmitter uses an oscillator and power amplifier. These simplified architectures are shown in Figure 25.

Figure 25: Block diagram of a super simple TX and RX.

Despite the simple architecture, digital amplitude-shift-keying (ASK) or on-off-keying (OOK) can be used. If the receiver is able to discriminate between frequencies (e.g., by using two RF filters with an envelope detector each), also frequency-shift-keying (FSK) can be used.

3.7 I/Q Imbalance

In direct-conversion and low-IF architectures, the I and Q paths are used to process the in-phase and quadrature components of the signal. Ideally, these paths should have identical gain and a 90° phase difference. However, in practice, there are always some mismatches between the I and Q paths, leading to **I/Q imbalance**. This imbalance can cause constellation distortion, leading to increased error vector magnitude (EVM) and degraded system performance.

I/Q imbalance can be characterized by two parameters: gain mismatch (ΔG) and phase mismatch ($\Delta\varphi$). Gain mismatch refers to the difference in gain between the I and Q paths, while phase mismatch refers to the deviation from the ideal 90° phase difference. The impact of I/Q imbalance on system performance depends on the modulation scheme used, with higher-order modulations being more sensitive to these impairments.

There are two ways to quantify I/Q imbalance:

- **Image rejection ratio (IRR)**: The IRR is a measure of how well the receiver can reject the image frequency caused by I/Q imbalance. It is defined as the ratio of the power of the desired signal to the power of the image (unwanted) signal, typically expressed in dB. A higher IRR indicates better performance, with values above 30 dB to 40 dB generally considered acceptable for most applications.
- **Error vector magnitude (EVM)**: The EVM is a measure of the difference between the ideal transmitted signal and the received signal, expressed as a percentage of the signal's magnitude. It quantifies the overall distortion in the received signal, including the effects of I/Q imbalance. Lower EVM values indicate better performance, with typical requirements ranging from 1% to 10% depending on the modulation scheme and application.

The EVM (in rms) is defined as

$$\mathrm{EVM} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \left|s_{\mathrm{ideal}}(i) - s_{\mathrm{meas}}(i)\right|^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \left|s_{\mathrm{ideal}}(i)\right|^2}} \tag{23}$$

where $s_{\text{ideal}}(i)$ is the ideal transmitted symbol, $s_{\text{meas}}(i)$ is the measured received symbol, and N is the number of symbols. EVM is expressed either in percent or in dB using

$$\text{EVM}\ |_{\text{dB}} = 20 \cdot \log_{10}(\text{EVM}).$$

In order to make the I/Q mismatch sufficiently small, among the possible techniques are:

- Careful layout and matching of the components in the I and Q paths to minimize gain and phase mismatches. This usually involves good layout techniques. Further, the LO I/Q generation should be done with high accuracy.
- Calibration techniques can be used to measure and compensate for I/Q imbalance. This can be done either in the analog domain (e.g., using variable gain amplifiers and phase shifters) or in the digital domain (e.g., using digital signal processing algorithms). Digital compensation is usually preferred, as it is more flexible and can adapt to changing conditions. A CORDIC can be readily used for this purpose.

4 Low Noise Amplifiers

As shown in Section 2.3.4, the sensitivity of a receiver is determined (besides the channel bandwidth) mainly by the noise figure of the receiver. The noise figure is in turn determined by the noise figure of the first active component in the receive chain, which is usually a low noise amplifier (LNA), as exemplified in Equation 18. Hence, as shown in Figure 21 and Figure 22, low noise amplifiers (LNAs) are the first active building block in a receiver after the antenna and some initial RF filtering.

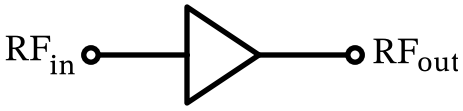

Figure 26: Block diagram of an LNA. Typically, the LNA input is impedance matched to 50 Ω, while the output is often not matched if the LNA is kept on chip. Often, the LNA gain is adjustable to allow for gain control in the receiver depending on the signal conditions. The LNA also might have a low-power bypass mode to reduce the power consumption of the LNA for sufficiently strong signals.

The LNA as a building block is shown in Figure 26. The main purpose of the LNA is to amplify the received signal with as little additional noise as possible. The LNA is usually designed for a specific frequency band, e.g., the 2.4 GHz ISM band or the 5 GHz WLAN band, and is typically designed for a specific impedance, e.g., 50 Ω, which is the standard impedance for RF systems. Impedance matching is usually required at the input; the output impedance matching is only required if the output of the LNA goes off-chip; if it is kept on-chip, impedance matching is often not required. The LNA is also designed to be sufficiently linear, i.e., to not introduce significant distortion to the amplified signal; however, compared to the noise requirements, linearity is often less critical.

Why is impedance matching at the input of the LNA so important?

- The antenna is usually designed for a specific impedance, e.g., 50 Ω, and if the LNA input is not matched to this impedance, a significant portion of the received signal will be reflected back to the antenna, resulting in a loss of signal power.

- Filters at the input of the LNA need to be terminated with the correct impedance to achieve the desired filter characteristics.
- Any transmission line in front of the LNA needs to be matched to avoid reflections and standing waves.

To quantify the "quality" of an impedance match, the reflection coefficient Γ is often used, which is defined as [3]:

$$\Gamma = \frac{Z_{\text{in}} - Z_0}{Z_{\text{in}} + Z_0}$$

where Z_0 is the characteristic impedance of the system (usually 50 Ω) and Z_{in} is the input impedance of the LNA. The reflection coefficient Γ is a complex number with a magnitude between 0 and 1, where 0 indicates a perfect match and 1 indicates a complete mismatch. This reflection coefficient can be represented on a Smith chart shown in Figure 27.

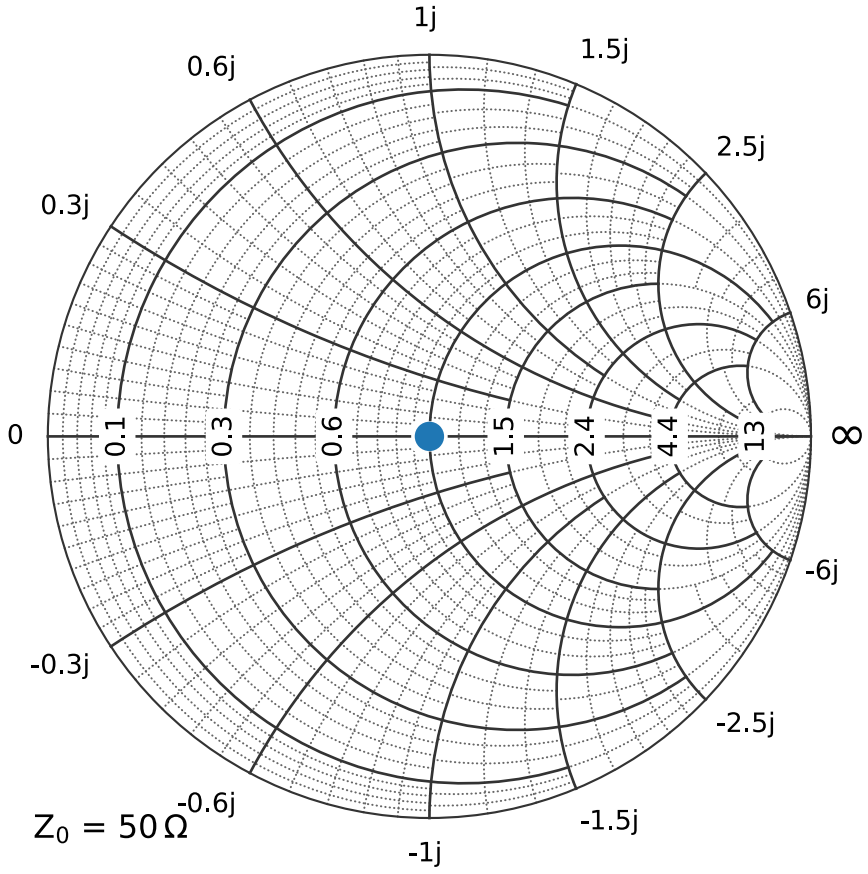

Figure 27: Smith chart showing constant resistance and reactance circles for impedance matching in RF circuits.

However, in practice, the more commonly used metric for impedance matching is the return loss (RL), which is defined as [3]:

$$\text{RL} = -20\log_{10}|\Gamma| = 20\log_{10}\left|\frac{Z_{\text{in}} + Z_0}{Z_{\text{in}} - Z_0}\right|.$$

A higher return loss indicates a better impedance match. A return loss of 10 dB indicates that 10% of the signal power is reflected back, while a return loss of 20 dB indicates that only 1% of the signal power is reflected back. In practice, a return loss of at least 10 dB is desired, with higher (positive) values being better [9].

4.1 Resistively Matched Common-Source LNA

The key question is now how to design an LNA with low noise figure and an input impedance matched to 50 Ω? In order to appreciate this design challenge, we will first try a naive approach, using a common-source amplifier with resistive termination, as shown in Figure 28.

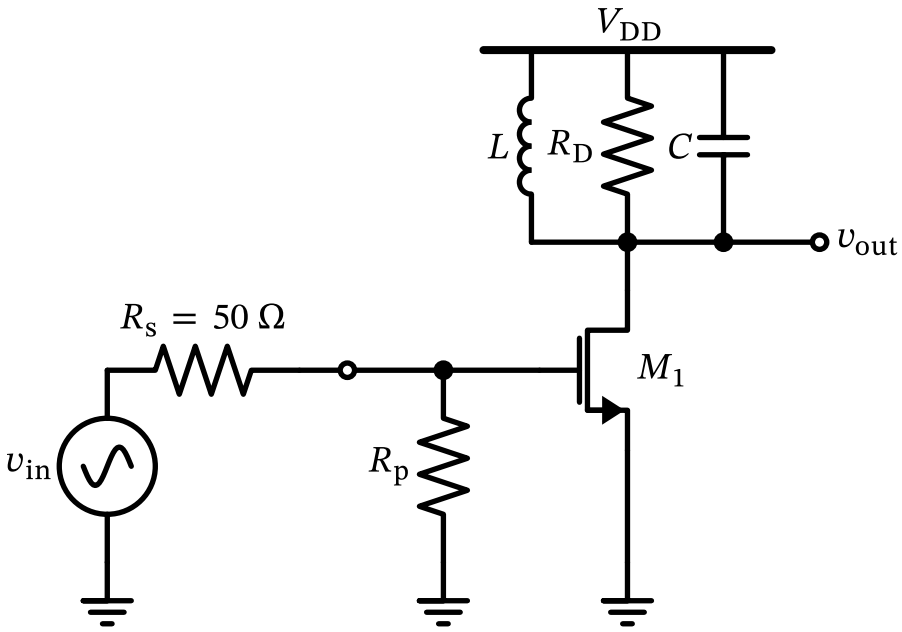

Figure 28: A simple LNA with resistive input matching and a tank circuit as a load (biasing details are omitted). The LNA is driven by a 50 Ω source.

If we assume the gate capacitance of M_1 is negligible, we can achieve good input impedance matching by choosing $R_s = R_p = 50\,\Omega$. The voltage gain of this simple common-source LNA is given by $A_v = -g_m R_D$, neglecting capacitances and g_{ds} of M_1 (we assume that the load tank is tuned to the desired frequency with $\omega_0 = 1/\sqrt{LC}$).

How can we calculate the noise figure of this simple LNA? We formulate

$$F = \frac{\text{total noise at output}}{\text{noise at output due to source only}}. \tag{24}$$

We derive a small-signal equivalent circuit of Figure 28, which is shown in Figure 29, to calculate the total noise at the output of the LNA.

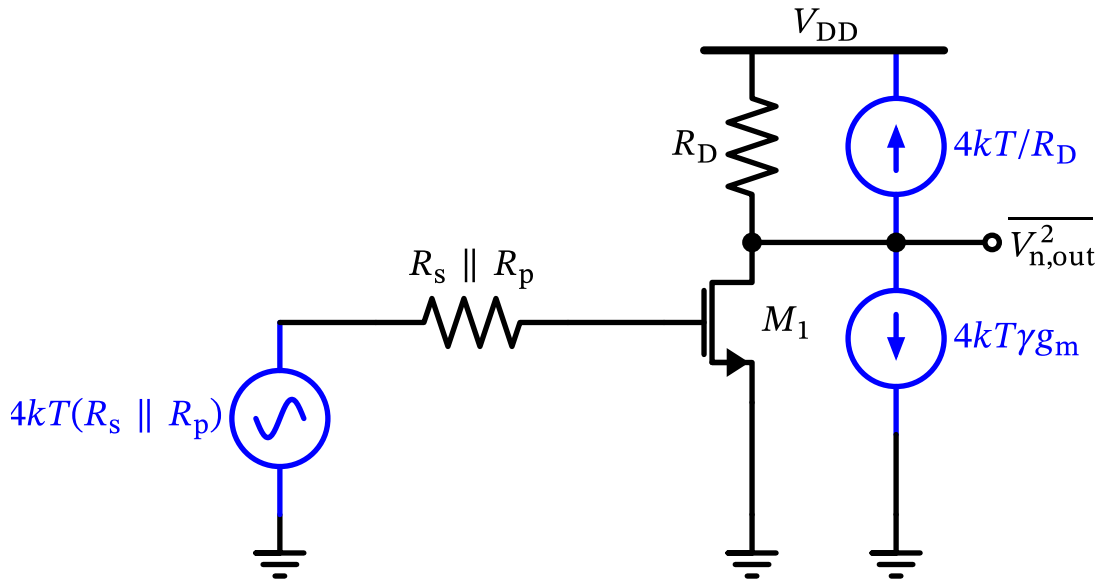

Figure 29: Equivalent circuit of resistively matched common-source LNA.

With the help of Figure 29, we can calculate the total output noise power spectral density as

$$\overline{V_{\mathrm{n,out,1}}^2} = A_\mathrm{v}^2 \cdot 4kT\left(R_\mathrm{s} \parallel R_\mathrm{p}\right) = \left(g_\mathrm{m} R_\mathrm{D}\right)^2 \cdot 4kT\left(R_\mathrm{s} \parallel R_\mathrm{p}\right)$$

and

$$\overline{I_{\mathrm{n,out}}^2} = 4kT\gamma g_\mathrm{m} + \frac{4kT}{R_\mathrm{D}}$$

$$\overline{V_{\mathrm{n,out,2}}^2} = R_\mathrm{D}^2 \cdot \overline{I_{\mathrm{n,out}}^2} = 4kT\gamma g_\mathrm{m} R_\mathrm{D}^2 + 4kT R_\mathrm{D}$$

so that in total

$$\overline{V_{\mathrm{n,out}}^2} = \overline{V_{\mathrm{n,out,1}}^2} + \overline{V_{\mathrm{n,out,2}}^2} = 4kT\left[\left(g_\mathrm{m} R_\mathrm{D}\right)^2\left(R_\mathrm{s} \parallel R_\mathrm{p}\right) + \gamma g_\mathrm{m} R_\mathrm{D}^2 + R_\mathrm{D}\right]. \tag{25}$$

We now need to find the output noise coming from the source only. For this we can use the equivalent circuit in Figure 30, to formulate the output noise due to the source only.

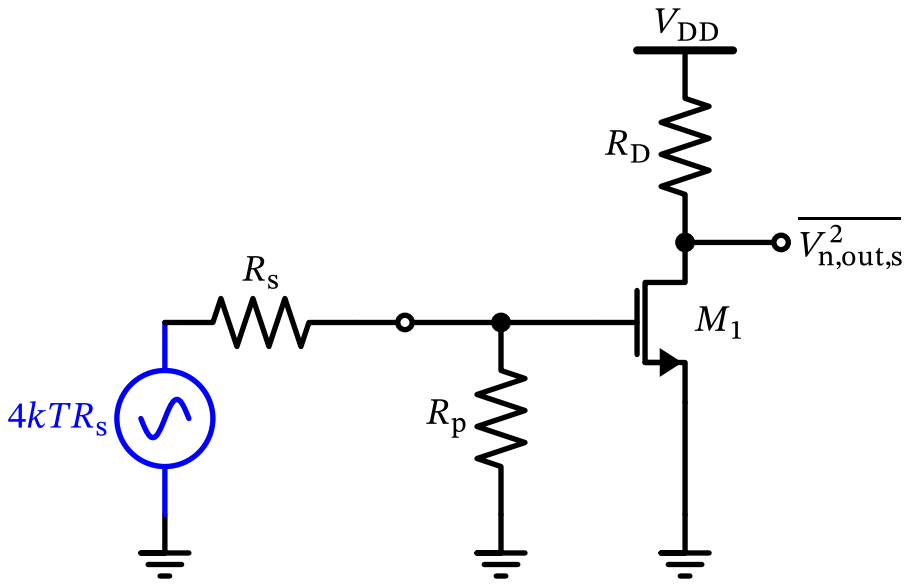

Figure 30: Eqivalent circuit to calculate the output noise from the input.

We find that

$$\overline{V^2_{n,out,s}} = A_v^2 \cdot 4kTR_s \cdot \left(\frac{R_p}{R_s + R_p}\right)^2. \tag{26}$$

Finally, we can use Equation 25 and Equation 26 with Equation 24 to calculate the noise figure of the simple resistively matched common-source LNA as

$$F = \frac{\overline{V^2_{n,out}}}{\overline{V^2_{n,out,s}}} = 1 + \frac{R_s}{R_p} + \frac{\gamma R_s}{g_m (R_s \parallel R_p)^2} + \frac{R_s}{g_m^2 (R_s \parallel R_p)^2 R_D}. \tag{27}$$

i Common-source LNA with Resistive Matching

As an exercise to calculate circuits with noise, re-confirm and derive yourself the result of Equation 27.

How can we interpret Equation 27? We see that we can minimize the noise factor by making g_m large. Then we have a noise factor of

$$F = 1 + \frac{R_s}{R_p} = 2$$

so we see that we are limited to a minimum noise figure of 3 dB, even if we spend the bias current to make g_m very large. We can go below 3 dB noise figure only if we choose $R_p > R_s$, however, this means that the input is no longer matched to 50 Ω, which is usually not acceptable. Hence, this simple resistively matched common-source LNA is not a good choice for a low-noise amplifier, with one exception: For very wideband amplifiers, where a NF of larger than 3 dB is acceptable, this configuration might be a good choice.

We see that we are stuck at high noise figures if we realize the real part of the input impedance with a resistor. This leaves us with the question of how to realize a real part of the input impedance then. We will answer this question in the next section.

4.2 Common-Gate LNA

We remember from our analog circuit design lecture that the common-gate configuration has an input impedance of $1/g_\mathrm{m}$, neglecting parasitic capacitances. Hence, if we choose $g_\mathrm{m} = 1/50\,\Omega = 20\,\mathrm{mS}$, we can achieve input matching to 50 Ω without using a resistor at the input. This is the key idea of the common-gate LNA, which is shown in Figure 31.

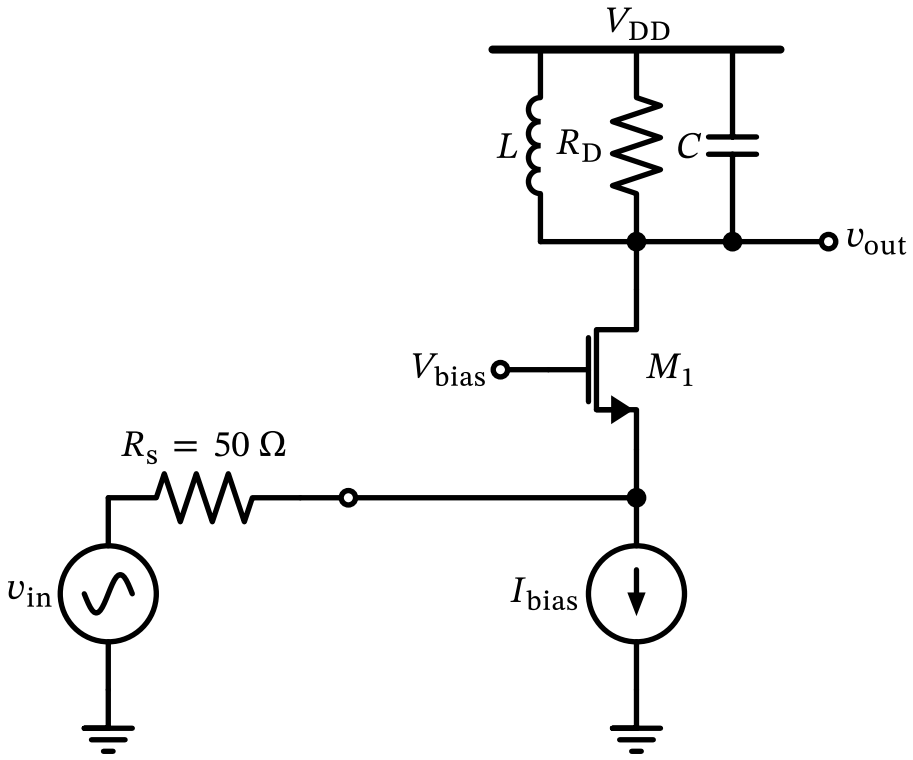

Figure 31: Circuit diagram of a common-gate LNA.

By inspecting Figure 31 and following the practice from Section 4.1, we can directly write down the output noise voltage as (with $1/g_\mathrm{m} = R_\mathrm{s}$)

$$\overline{V_\mathrm{n,out}^2} = kT\left[(g_\mathrm{m}R_\mathrm{D})^2 R_\mathrm{s} + \gamma g_\mathrm{m} R_\mathrm{D}^2 + 4R_\mathrm{D}\right] = kT\left(\frac{R_\mathrm{D}^2}{R_\mathrm{s}} + \gamma\frac{R_\mathrm{D}^2}{R_\mathrm{s}} + 4R_\mathrm{D}\right). \tag{28}$$

The output noise due to the source only is given by

$$\overline{V_\mathrm{n,out,s}^2} = \frac{R_\mathrm{D}^2}{R_\mathrm{s}}. \tag{29}$$

Finally, we can use Equation 28 and Equation 29 with Equation 24 to calculate the noise figure of the common-gate LNA as

$$F = 1 + \gamma + \frac{4R_\mathrm{s}}{R_\mathrm{D}} \overset{R_\mathrm{D} \gg R_\mathrm{s}}{\rightarrow} F = 1 + \gamma. \tag{30}$$

With a classical long-channel $\gamma = 2/3$, we can achieve a minimum noise figure of 2.2 dB, which is already better than the resistively matched common-source LNA. However, with modern short-channel devices, γ is often larger than 1, so that the minimum noise figure of the common-gate LNA is often larger than 3 dB [1].

4.3 Inductively-Degenerated Common-Source LNA

As we have seen in Section 4.2, using circuit techniques can realize a real part of an input impedance without the associated thermal noise of a resistor. We now try something different, in the hope that it will result in an even lower noise figure. We construct an LNA based on a common-source MOSFET amplifier, but we add an impedance Z_{deg} into the source line. This arrangement is shown in Figure 32.

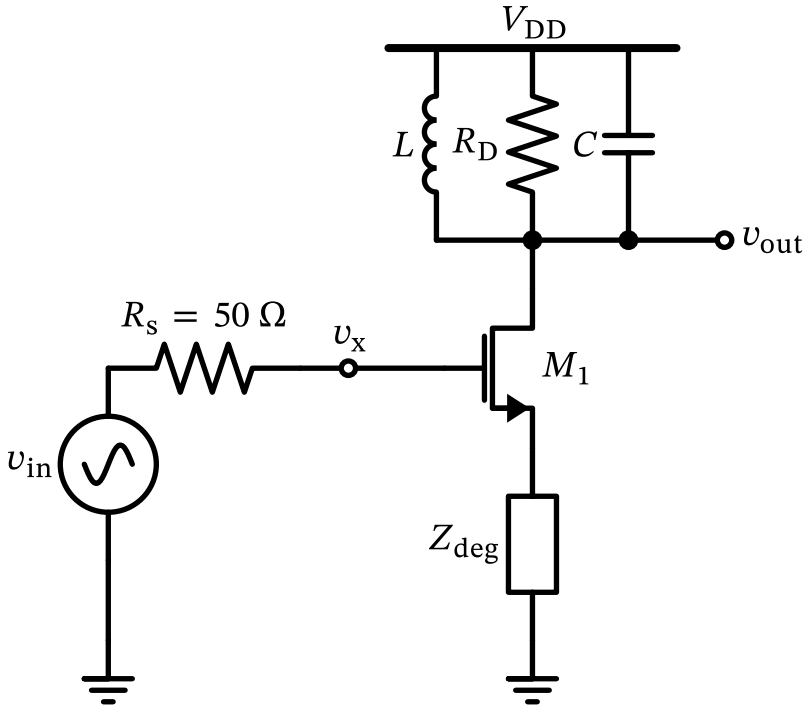

Figure 32: A common-source MOSFET stage with degeneration impedance.

We now extract the small-signal equivalent circuit of Figure 32, which is shown in Figure 33, to calculate the input impedance.

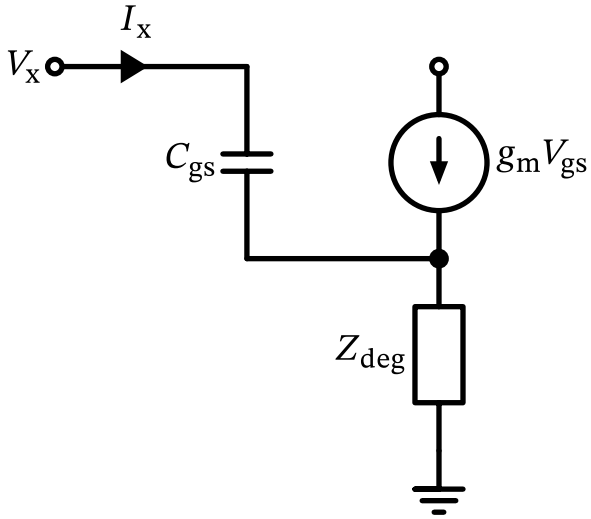

Figure 33: Equivalent small-signal circuit of the input stage around M_1.

We find that

$$V_x = V_{gs} + Z_{deg}(I_x + g_m V_{gs}), \quad V_{gs} = \frac{I_x}{sC_{gs}}$$

so that we can write the input impedance as

$$Z_{in} = \frac{V_x}{I_x} = \frac{1}{sC_{gs}} + Z_{deg} + \frac{g_m Z_{deg}}{sC_{gs}}. \tag{31}$$

The final term in Equation 31 is the interesting one: By choosing Z_{deg} to be inductive (which we can do by either use an on-chip or off-chip inductor), we can realize a real part of the input impedance. If we choose $Z_{deg} = sL$, we find that

$$Z_{in} = \frac{1}{sC_{gs}} + sL + \frac{g_m L}{C_{gs}}.$$

By proper choice of L and C_{gs}, we can achieve input matching to 50 Ω at the desired frequency ω_0. We find that the real part of the input impedance is given by

$$\Re\{Z_{in}\} = \frac{g_m L}{C_{gs}}$$

Without proof (refer to [1] or [2] for a derivation) we find for the noise factor of this input stage (with some simplification) as

$$F = 1 + \frac{\gamma R_s \omega_0^2 C_{gs}^2}{g_m}. \tag{32}$$

Finally, we have an LNA input stage configuration that allows us to achieve a noise figure below 3 dB, even with $\gamma > 1$, by proper choice of g_m. Making g_m large (by spending more bias current) results in (to first order) arbitrarily low noise figure. The inductively-degenerated common-source LNA is a widely used LNA input stage configuration in modern RFICs. A somewhat detailed schematic is shown in Figure 34.

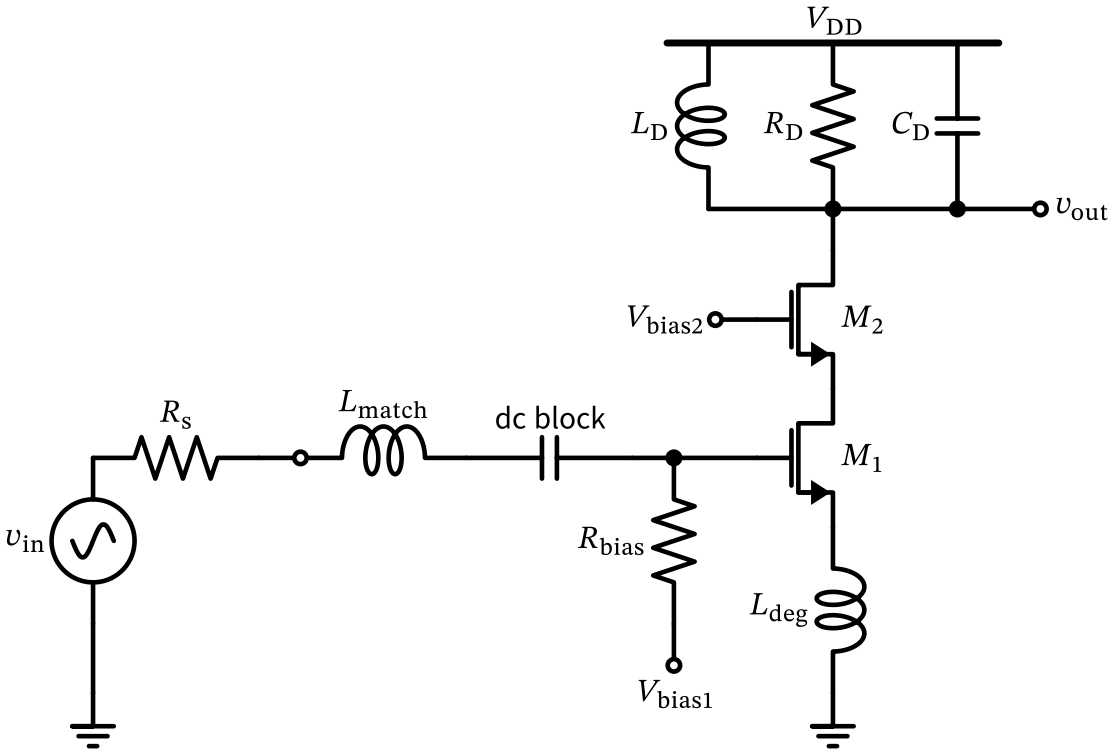

Figure 34: An (almost complete) common-source MOSFET stage with degeneration impedance and cascode.

The inductor L_{match} is used to match the input impedance to 50 Ω at the desired frequency, L_{deg} is used to realize the real part of the input impedance, R_{bias} is used to set the bias current of M_1, M_2 is a cascode transistor which increases the output impedance and thus the gain of the stage (plus it improves the reverse isolation), and R_{D}, L_{D}, and C_{D} form a load tank which provides high gain at the desired frequency. A dc block is used at the input so that the bias point of M_1 is not corrupted by the input signal source. The bias voltage V_{bias2} sets the operating point of the cascode transistor M_2.

What is missing in Figure 34 is any form of frequency tuning of the load to the frequency of interest, and the support of different gain modes. Apart from these details this LNA circuit is a good starting point for a practical LNA design.

4.4 Feedback LNA

One drawback of the inductively-degenerated common-source LNA is the usage of at least one inductor. If the inductor is placed on-chip, it has a comparatively large size, and if it is implemented in the package (via a bondwire) or on the PCB, it adds to the bill-of-materials (BOM) cost.

If the CMOS technology is sufficiently fast, a shunt feedback LNA, as shown in Figure 35, might be a good choice.

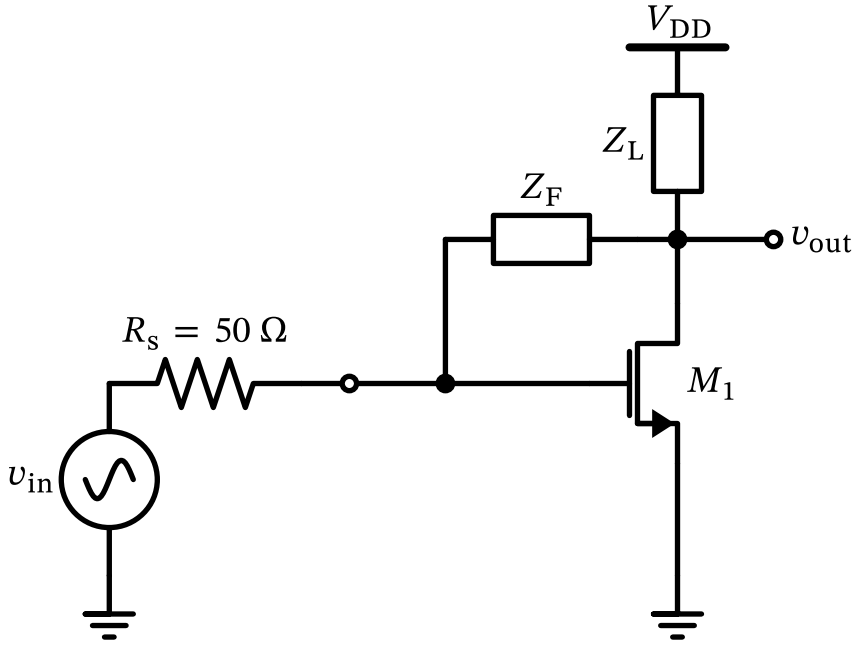

Figure 35: A shunt-feedback LNA.

Without proof, the input impedance of the shunt feedback LNA is given by

$$Z_{in} = \frac{Z_F + Z_L}{1 + g_m Z_L}. \tag{33}$$

The noise factor of the shunt feedback LNA is given by

$$F = 1 + \left| \frac{Z_F + R_s}{g_m Z_F + 1} \right|^2 \cdot \frac{\gamma g_m + \Re\{Y_L\}}{\Re\{Z_{in}\}} \tag{34}$$

As you can see from Equation 34, by making g_m large (by spending more bias current) the noise figure can be made arbitrarily small! Depending on the choice of Z_F and Z_L, the input impedance of this LNA can be changed in interesting ways.

By setting $Z_L \to \infty$ (e.g., by biasing with a current source and high-impedance loading), we find that

$$Z_{in} = \frac{1}{g_m}$$

which is independent of Z_F and is a well-known result for a common-source stage. The disadvantage of this configuration is the noise factor, which (given that Z_F is sufficiently large) tends to $F = 1 + \gamma$, which is the same as for the common-gate LNA.

A bit more interesting is the case when $g_m Z_L = A_0$ and $Z_L \gg Z_F$, which results in

$$Z_{in} = \frac{Z_L}{1 + A_0}$$

which is the well-known result that the input impedance of an amplifier with feedback is reduced by the factor $1 + A_0$, where A_0 is the open-loop gain of the amplifier. The noise factor can be made small by making g_m large, as we have already noted above.

A very interesting case can be achieved by choosing $Z_\mathrm{L} = 1/sC_\mathrm{L}$ and $Z_\mathrm{F} = 1/sC_\mathrm{F}$, which results in

$$Y_\mathrm{in} = \frac{1}{Z_\mathrm{in}} = \frac{g_\mathrm{m} C_\mathrm{F}}{C_\mathrm{L} + C_\mathrm{F}} + s\frac{C_\mathrm{L} C_\mathrm{F}}{C_\mathrm{L} + C_\mathrm{F}} \tag{35}$$

Looking at Equation 35, we see that the input admittance has a **real part**! By proper choice of components, we can achieve an input impedance matched to 50 Ω at the desired frequency. The noise factor can again be made small by making g_m large.

There is also an option, by proper choice of Z_F and Z_L, to achieve an inductive input impedance component, which can be used to resonate out the input capacitance of the LNA, similar to the inductively-degenerated common-source LNA. However, in contrast to the inductively-degenerated common-source LNA, no inductor is required in this case. This configuration is called a reactance-canceling LNA.

5 Mixers

As we have seen in Section 3.2, we are using mixers to "upconvert" and "downconvert" signals, i.e., we shift the frequency of a signal to a higher or lower frequency with the help of an auxiliary signal, called the "**local oscillator**" (LO) signal. The block diagram of a mixer is shown in Figure 37.

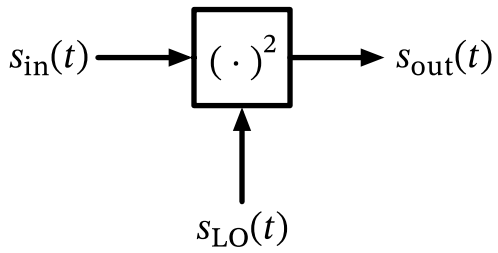

Figure 36: A squarer as a nonlinear mixer.

What we usually want in a mixer is that the input signal at ω_in is shifted by ω_LO, so that the output signal is located at

$$\omega_\mathrm{out} = \omega_\mathrm{in} \pm \omega_\mathrm{LO}.$$

It should be noted that the output signal usually contains additional frequency components, which are not desired. These so-called "**spurious**" components are usually filtered out in a subsequent filtering stage.

Recalling the properties of linear time-invariant (LTI) systems, we know that in an LTI system *no new frequency components* (which are not already present in the input signal) are generated. In conclusion, in order to generate new frequency components, a mixer has to be either

- a **non-linear** system, or
- a **time-variant** system.

Next we will look at two options on how to implement a mixer.

5.1 Non-Linear Mixer

Generally speaking, we can use any non-linear device to implement a mixer (often, this generation of new frequency components in a nonlinear system is highly undesired; in a mixer, we want this effect). As a simple example we look at the case of a square function, shown in Figure 36.

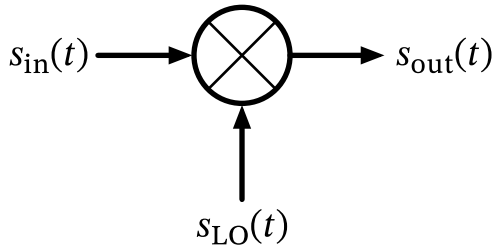

Figure 37: Mixer block diagram.

When we apply the sum of the signals $s_{\text{in}}(t) = \cos(\omega_{\text{in}} t)$ and $s_{\text{LO}}(t) = \cos(\omega_{\text{LO}} t)$ to the input of a squarer, we get the output signal

$$\begin{aligned} s_{\text{out}}(t) = \left[s_{\text{in}}(t) + s_{\text{LO}}(t)\right]^2 &= \cos(\omega_{\text{in}} t + \omega_{\text{LO}} t) + \cos(\omega_{\text{in}} t - \omega_{\text{LO}} t) \\ &\quad +1 + \frac{1}{2}\cos(2\omega_{\text{in}} t) + \frac{1}{2}\cos(2\omega_{\text{LO}} t). \end{aligned} \tag{36}$$

We see that the output signal contains the desired frequency components at $\omega_{\text{in}} \pm \omega_{\text{LO}}$, but also additional components at dc (0 Hz), $2\omega_{\text{in}}$, and $2\omega_{\text{LO}}$. These additional components are usually unwanted and have to be filtered out in a subsequent filtering stage. As a side-note, in some circuits a **frequency doubler** is desired, which can be implemented by simply filtering out the other frequency components. In a doubler no LO signal is required, as the input signal is simply squared.

A simple example circuit is a diode, which has a non-linear current-voltage characteristic. The block diagram of a simple diode mixer is shown in Figure 38.

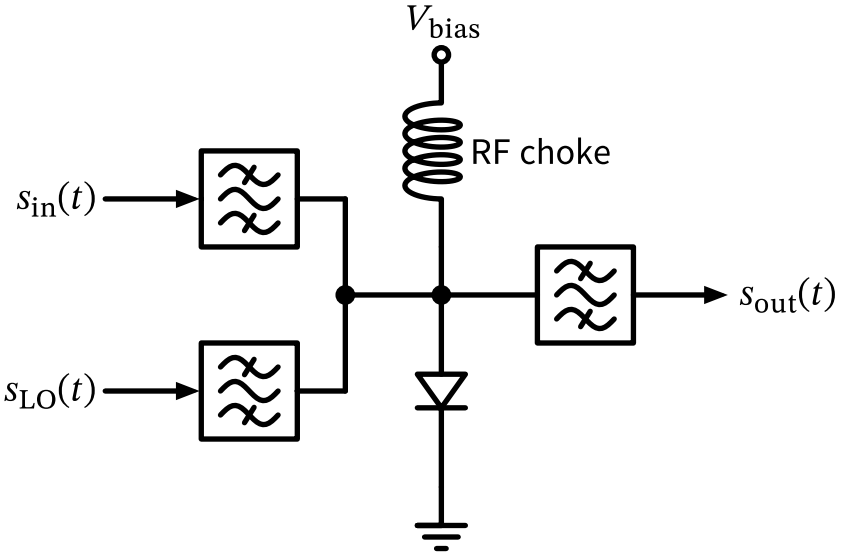

Figure 38: A diode mixer.

This structure is conceptionally simple, it just requires a (fast) diode, filters to couple the desired frequency components in and out, and an RF choke to provide a dc bias voltage for the diode. However, the performance of such a simple mixer is usually not very good, as the diode is a highly non-linear device, which generates many spurious frequency components. More advanced mixer circuits use more complex non-linear devices (for example, a ring of diodes) to improve the performance. The advantage of these mixers is that they can operate up to very high frequencies (in the mm-wave range and beyond). If you can make a nonlinear device, you can make a mixer (think also of nonlinear optics, for example).

5.2 Time-Variant Mixer

In contrast to the non-linear mixer, a time-variant mixer uses an ideally **linear** device, but changes its properties over time. A simple example is a switch, which is opened and closed at the LO frequency. The block diagram of such a mixer is shown in Figure 39. This system is ideally linear from input to output, but very nonlinear when considering the LO input.

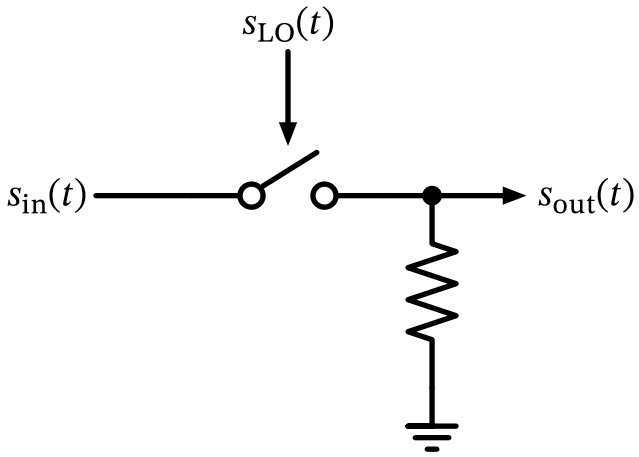

Figure 39: A switch as a time-variant mixer.

When the switch is closed ($s_{\mathrm{LO}} \geq 0$), the input signal is passed to the output, i.e., $s_{\mathrm{out}}(t) = s_{\mathrm{in}}(t)$; when the switch is open ($s_{\mathrm{LO}} < 0$), no signal is passed, i.e., $s_{\mathrm{out}}(t) = 0$. The output signal can be expressed as

$$s_{\mathrm{out}}(t) = s_{\mathrm{in}}(t) \cdot \frac{1}{2}\{1 + \mathrm{sgn}[s_{\mathrm{LO}}(t)]\},$$

where $\mathrm{sgn}(\cdot)$ is the sign function. The term in curly brackets is a square wave that switches between 0 and 1 at the LO frequency. This square wave can be expressed as a Fourier series, which contains the fundamental frequency at ω_{LO} and all odd harmonics $3\omega_{\mathrm{LO}}$, $5\omega_{\mathrm{LO}}$, and so on. In conclusion, if the input signal is expressed as $\cos(\omega_{\mathrm{in}}t)$, we get the output signal

$$s_{\mathrm{out}}(t) = \frac{1}{\pi}\cos(\omega_{\mathrm{in}}t \pm \omega_{\mathrm{LO}}t) + \ldots \tag{37}$$

Again, we see that the output signal contains the desired frequency components at $\omega_{\mathrm{in}} \pm \omega_{\mathrm{LO}}$, but also additional components at $3\omega_{\mathrm{LO}}$, $5\omega_{\mathrm{LO}}$, and so on. These additional components are usually unwanted and have to be filtered out in a subsequent filtering stage. The advantage of a time-variant mixer is that it can be implemented readily in CMOS, as shown in Figure 40.

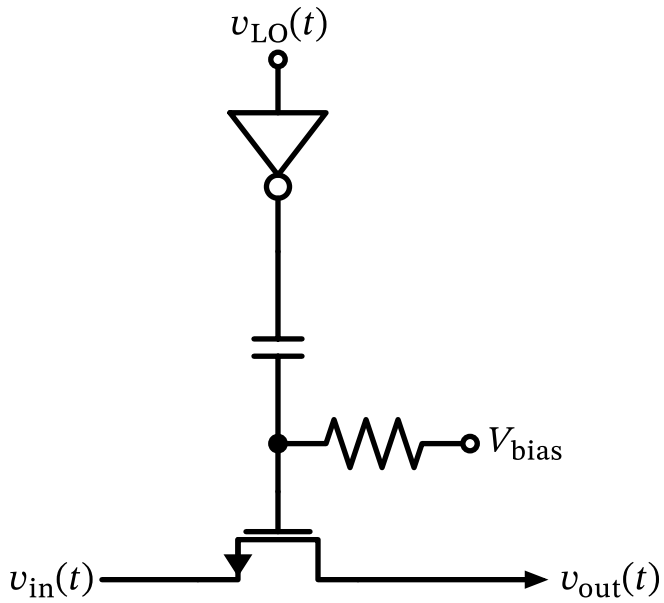

Figure 40: A MOSFET as a switch used as a mixer, with ac-coupled LO signal.

The implementation in Figure 40 uses a single NMOS transistor as a switch. The LO signal is ac-coupled to the gate of the transistor, so that the dc operating point is set by the bias voltage V_{bias}. When the LO signal is high enough, the transistor is switched on and the input signal $v_{\mathrm{in}}(t)$ is passed to the output; when the LO signal is low, the transistor is switched off and no signal is passed. Note that with the MOSFET we can switch voltages as well as currents, so the mixer can work in both modes, voltage mode or current mode.

A big disadvantage of this simple implementation is that the input signal is "lost" for half of the cycle when the MOSFET switch is open. A fully differential implementation (having differential ports at input, output and LO input) can alleviate this problem, as the input signal

is always connected to one of the two output ports. This configuration is called a "**double-balanced mixer**" and is widely used in practice. It is shown in Figure 41.

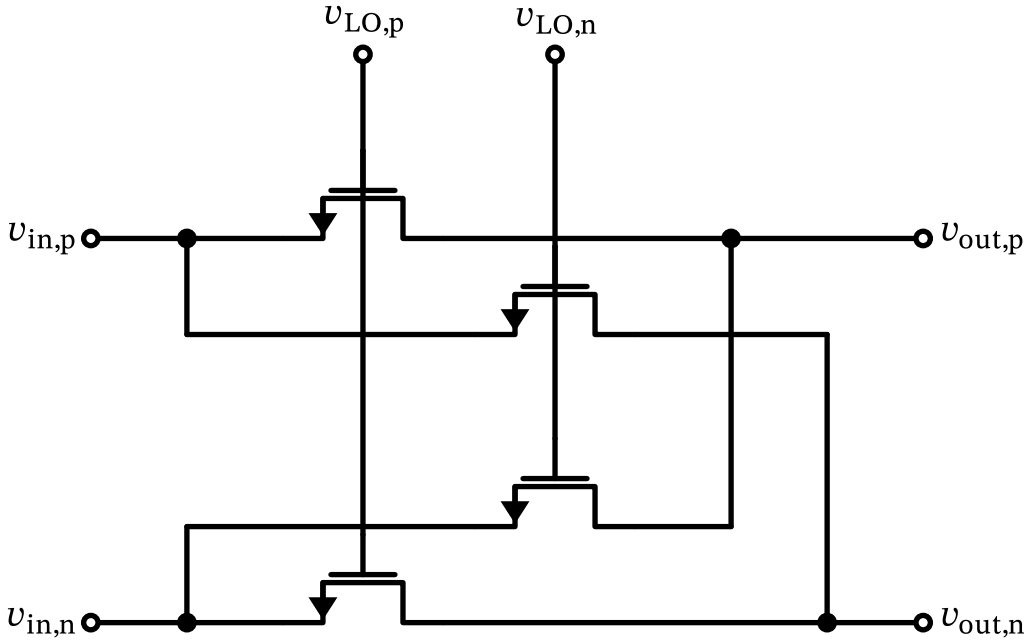

Figure 41: A fully-differential double-balanced MOSFET mixer.

Note that with a double-balanced mixer, we have 6 dB more conversion gain compared to the single-balanced mixer (refer to Equation 37), as can be seen in Equation 38. Note that the factor of $2/\pi$ represents the conversion loss of an ideal double-balanced mixer of 3.9 dB.

$$s_{\mathrm{out}}(t) = \frac{2}{\pi}\cos(\omega_{\mathrm{in}}t \pm \omega_{\mathrm{LO}}t) + ... \tag{38}$$

Note that no dc current needs to flow through the mixer structures shown in Figure 40 and Figure 41, which is a huge advantage when thinking of flicker noise. Flicker noise in a MOSFET is proportional to the dc current flowing through the device, so if no dc current flows, no flicker noise is generated!

In order to look at the full picture, we embed the mixer of Figure 41 in a complete RX front-end, as shown in Figure 42. An LNA (essentially a g_{m} cell) creates a current signal from the received voltage signal at the antenna. This current signal is then fed to the mixer in the current domain, and further sinked into a transimpedance amplifier (TIA), which also implements a lowpass pole. The simplified circuit is shown in Figure 42.

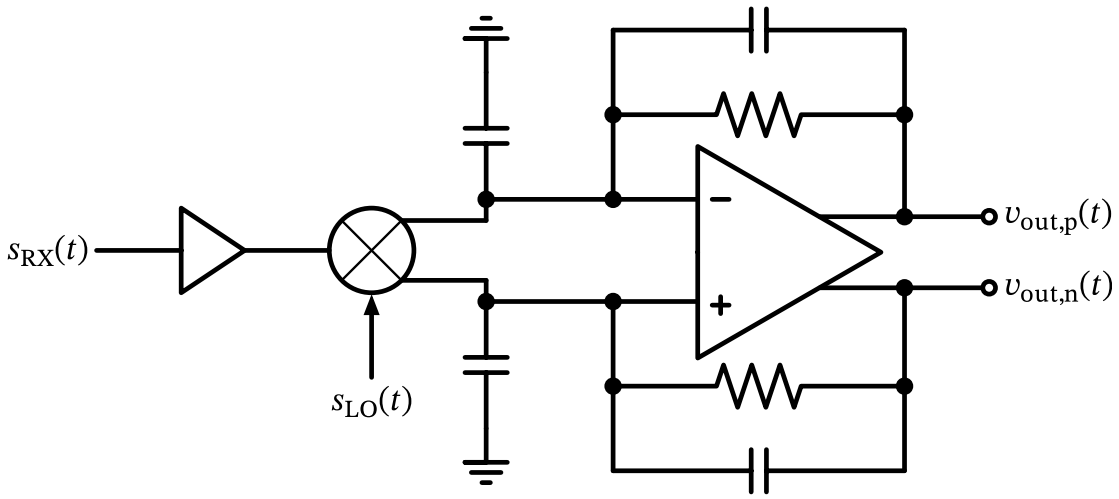

Figure 42: An RX front-end using a current-mode (passive) mixer.

The capacitors connected directly to ground at the mixer outputs are good practice, as they shunt high-frequency switching noise to ground, and in this way help the performance of the TIA, which otherwise would have to sink these currents. Note that all blocker currents originating at the LNA output and passing through the mixer are shunt by the feedback capacitors to the output of the TIA, and need to be actively driven by the TIA's output stage!

5.3 Gilbert Cell Mixer

As we have seen in Section 5.2, in CMOS we have quite a few options to implement a mixer as the MOSFET is a good current and voltage switch. This is in stark contrast to the BJT, as the bipolar transistor can only be used as an (excellent) current switch in the differential pair configuration. Hence, we need to implement a current-mode mixer in bipolar technology. The most widely used structure is the so-called "**Gilbert cell**", shown in Figure 43.

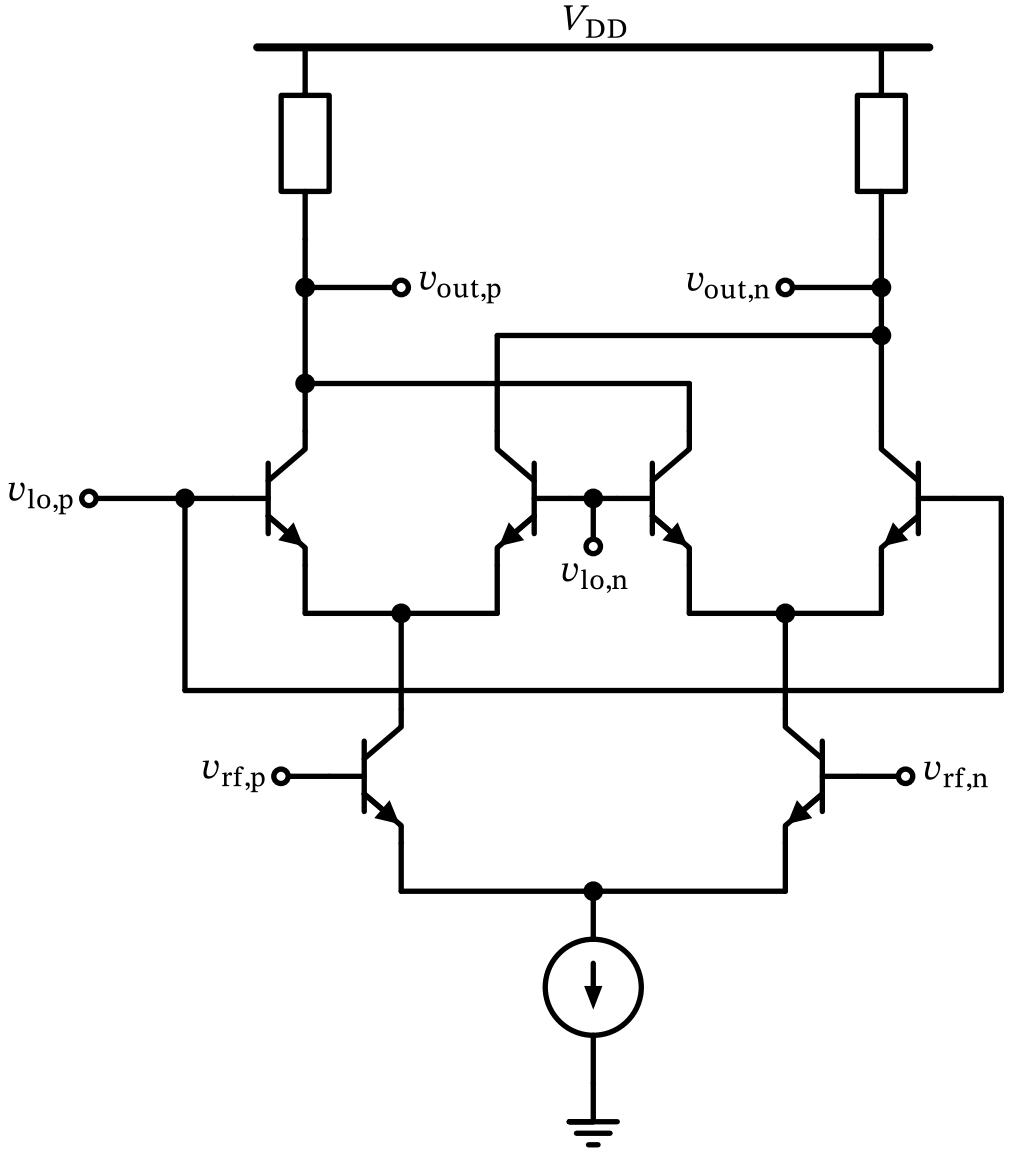

Figure 43: A Gilbert mixer based on bipolar differential pairs.

The idea is that an input transconductor creates currents, which are switched in the double differential pair, and converts the mixed currents back to voltage in the load impedances. There are many known variations of this circuit, like skipping the current source at the bottom to increase linearity and headroom, or swap the input transconductor for another structure. It might also be useful to "bleed" some bias current from the LO switching stage to improve noise.

5.4 N-Path Filter

Without much deliberation or diving into the background, it has to be stated that passive mixers (based on MOSFET switches) have a few very interesting properties. One of them is that they can be used to implement very high-quality bandpass filters, called "**N-path filters**" [10]. The basic idea is shown in Figure 44. We use 4 phases of an LO signal to switch the 4 capacitors to ground in a round-robin fashion. The duty cycle of each LO phase is 25% to have a non-overlapping clock.

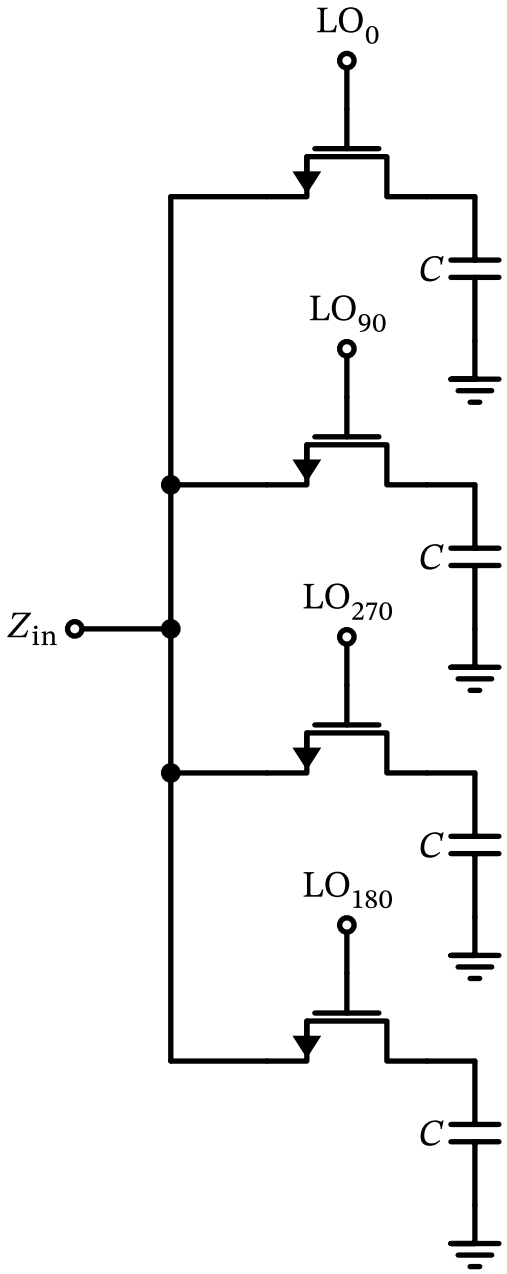

Figure 44: A 4-phase N-path filter.

The key observation to make is that a switch that is opened and closed at a certain frequency (the LO frequency) can be seen as a time-variant resistor. This time-variant resistor converts impedance seen at one one to the other end, and also changes the frequency of this apparent impedance. In other words, if we connect grounded capacitors at one end of the mixer switches and look into the other end, we see a **bandpass filter centered at the LO** frequency! In this way we can build bandpass filters (and also bandstop filters [11]) at very high frequencies which are precisely centered (without component variations) around ω_{LO}.

The impedance characteristics of such a tank circuit exhibit first-order bandpass behavior, as shown in Figure 45. The peak impedance reaches 5.3 times the switch resistance (R_{sw}), and importantly, the bandwidth is inversely proportional to the capacitance value, i.e., $\mathrm{BW} \propto$

$1/C$. This relationship allows for precise control of the filter bandwidth by adjusting the capacitor values.

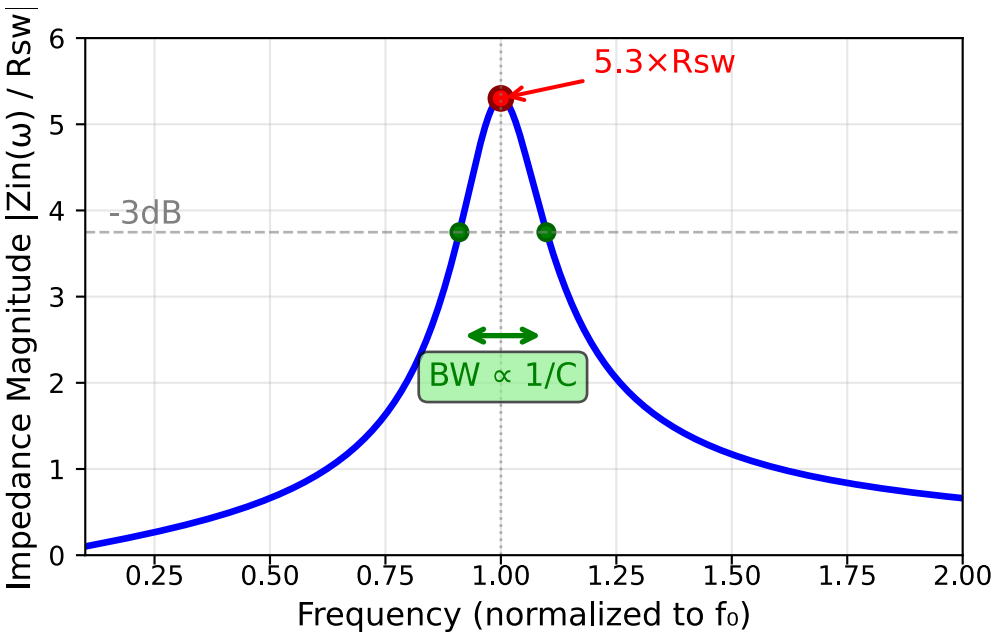

Figure 45: First-order bandpass behavior and impedance of an N-path filter.

5.5 LO Generation

As you have seen in Section 3.2, for complex-valued modulation schemes (like QPSK, QAM, OFDM, and so on) we need to generate quadrature LO signals (i.e., two LO signals with a 90° phase shift). There are many ways to generate such signals, and we will now study a few of them.

5.5.1 RC/CR Phase Shift Network

The probably simplest way to generate quadrature LO signals is to use a simple RC/CR phase shift network, as shown in Figure 46. The RC network (a first-order lowpass) generates a phase shift between 0° and −90°, while the CR network (a first-order highpass) generates a phase shift between 0° and +90°. By operating the network at $\omega_0 = 1/RC$, we can achieve a 90° phase shift between the two outputs.

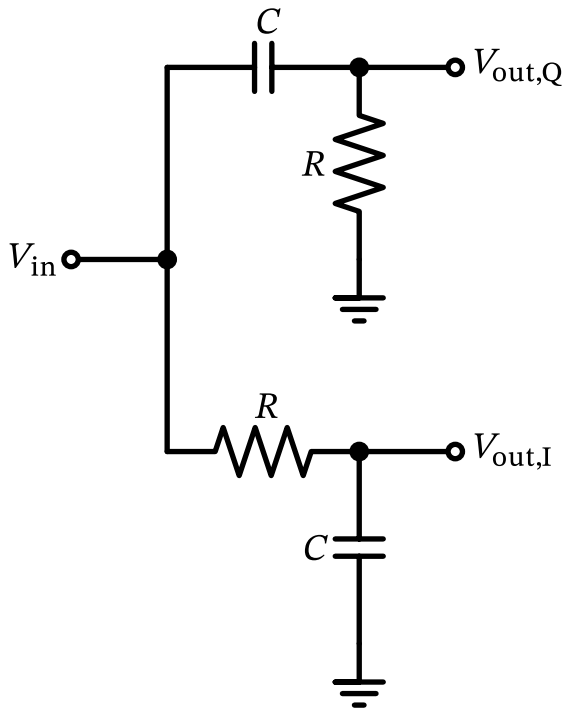

Figure 46: An RC/CR IQ generation network.

The advantage of this network is its simplicity, as it only requires a few passive components, and it can generate I and Q phases from an incoming signal at the target frequency. The disadvantage is that the phase shift is frequency-dependent, so the quadrature signals are only perfectly in phase at one frequency. Furthermore, the amplitude of the two outputs is not equal, which might require additional gain stages to equalize the amplitudes. Additionally, this passive network has a 3 dB loss, so we need additional gain stages to compensate for this loss.

5.5.2 Polyphase Filter

We can create a more advanced phase shift network (following the idea of Section 5.5.1) by using a so-called "**polyphase filter**" [12]. The idea is to use multiple RC sections to create a more ideal phase shift network. An example is shown in Figure 47.

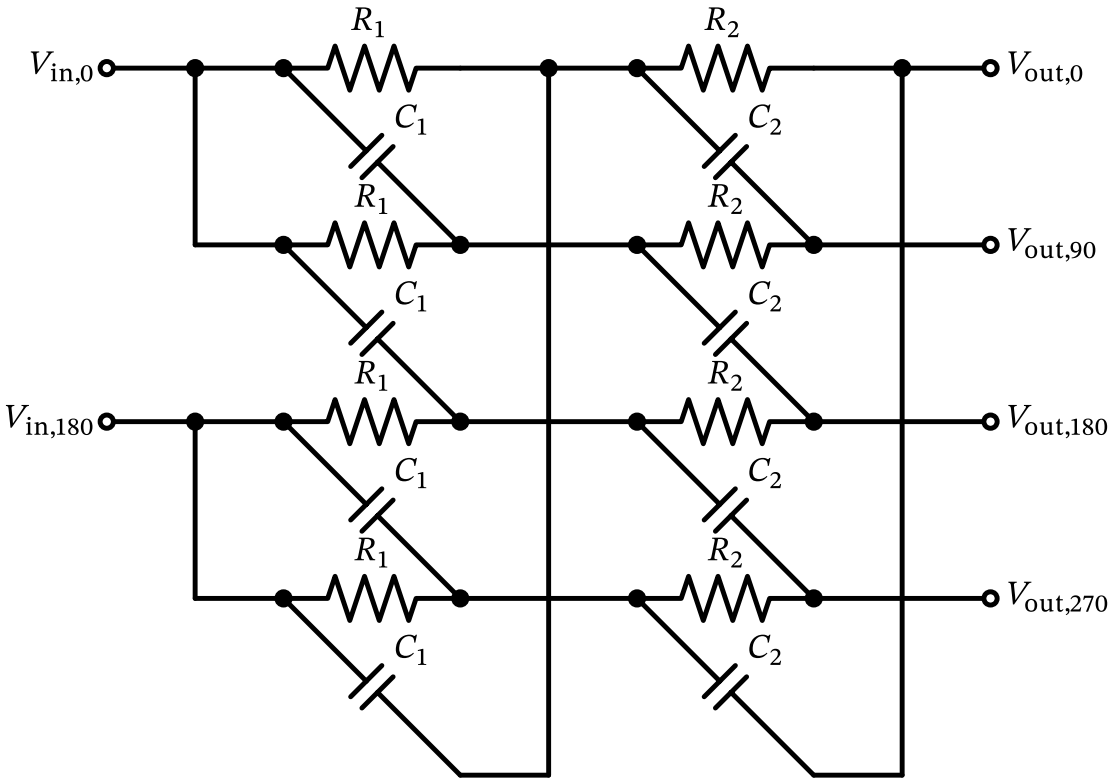

Figure 47: A two-stage polyphase network.

As shown in the figure above, we can use multiple stages (with $R_1C_1 \neq R_2C_2$) in cascade to broaden the frequency range over which we have a good 90° phase shift between the four outputs. The more stages we use, the better the performance, but also the more components are required. Note that this network is still passive, so it has an inherent loss, which also increases with the number of stages. The idea behind this network is that the network is transparent for positive frequencies, and blocks negative frequencies (or vice versa). We enter the network with real signals (having positive and negative frequency components), and at the output we get complex signals (having only positive or negative frequency components). In this sense this network is a complex filter.

Note that polyphase filters are popular when we have to create complex signals from real signals. We can use the polyphase filter in the LO path, or also in the signal path (e.g., in a receiver). The benefit is that we can work with a signal where input frequency is equal to the output frequency. This is in strong contrast to the approach we will discuss next.

5.5.3 Flip-Flop Based Phase Generation

The most widely used approach to generate quadrature LO signals is to use digital circuits, as shown in Figure 48. The idea is to use a flip-flop (or a latch) to divide the frequency of an incoming clock signal by 2. If we drive the two flip-flops (which are connected as toggle flip-flops) by inverted clocks, then the resulting output signals are 90° phase-shifted, as shown in Figure 49.

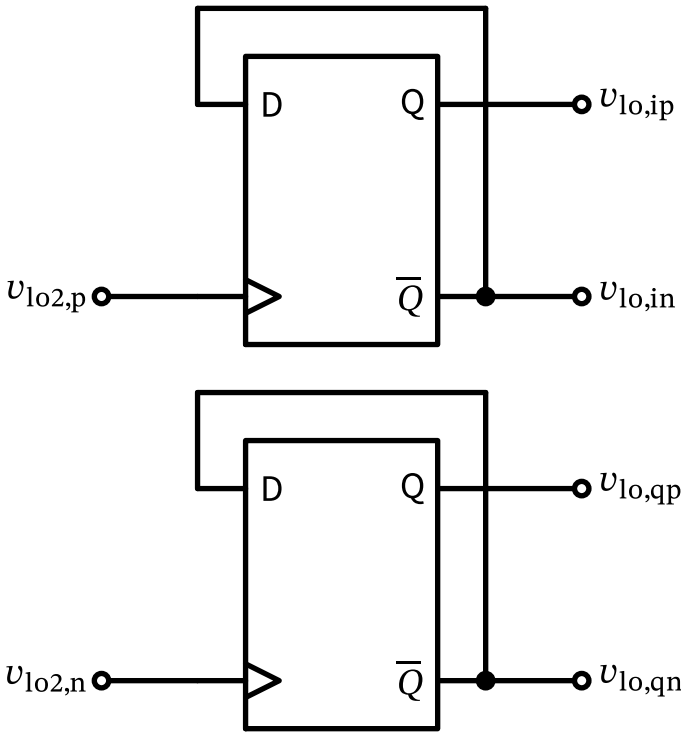

Figure 48: I/Q generation with a divide-by-2.

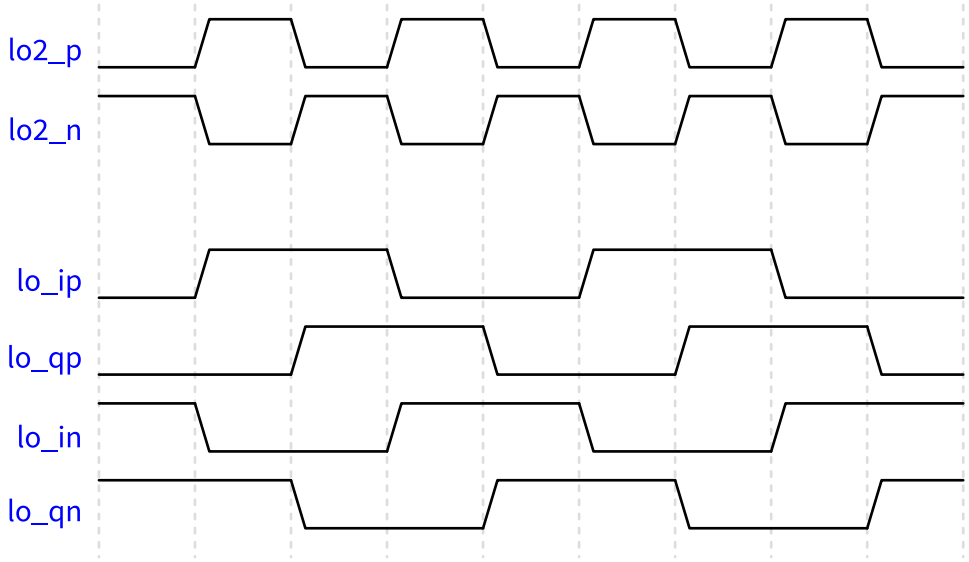

Figure 49: Input and output waveforms of I/Q generation with a divide-by-2.

It is important to note that the circuit shown in Figure 48 has to be implemented in a way so that there is no phase ambiguity, i.e., the I-phase is lagging the Q-phase by exactly 90°. This can be achieved by using a reset signal to set the flip-flops to a known state at startup, or by implementing flip-flops using a differential clock input.

If the input clock is not 2x the target frequency but higher (e.g., 4x), then we can use a frequency divider approach to generate more than four phases. This could be useful in an N-path filter, where we could need more than four phases (see Section 5.4).

Often, we need four-phase LO signals with 25% duty cycle (instead of the 50% duty cycle shown in Figure 49). This can be achieved by using additional logic gates to combine the outputs of the flip-flops, as shown in Figure 50. The resulting LO waveforms are shown in Figure 51.

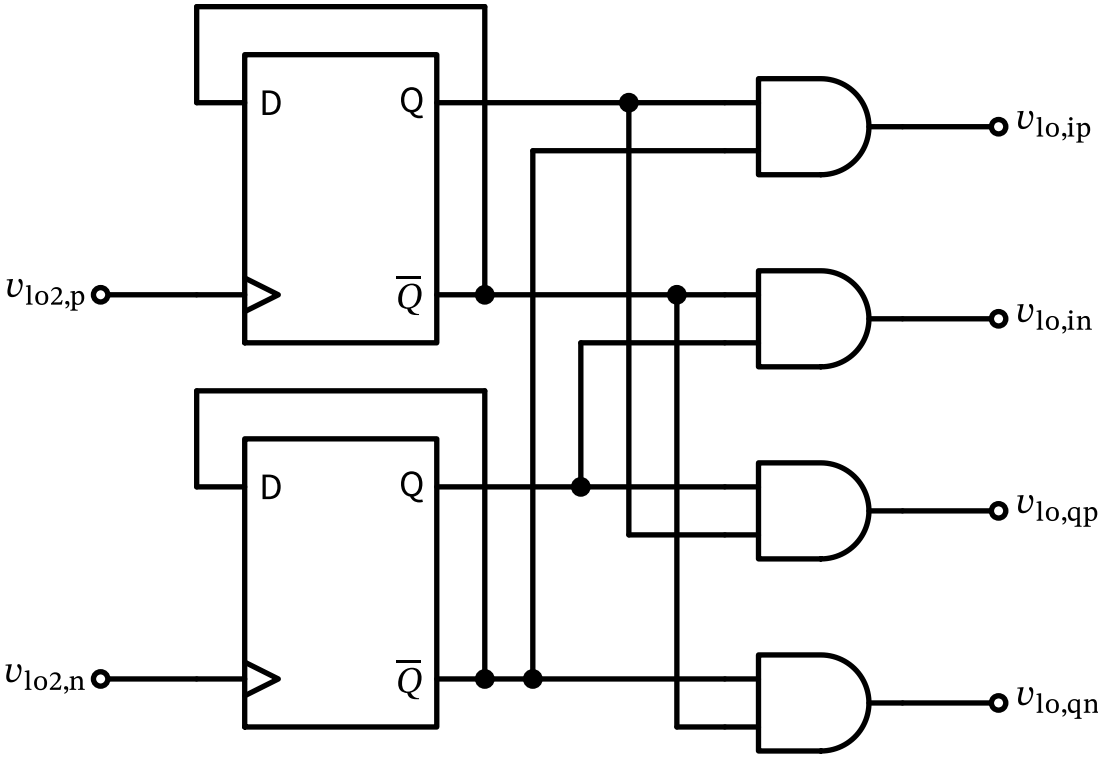

Figure 50: I/Q generation with a divide-by-2 and 25% duty cycle generation.

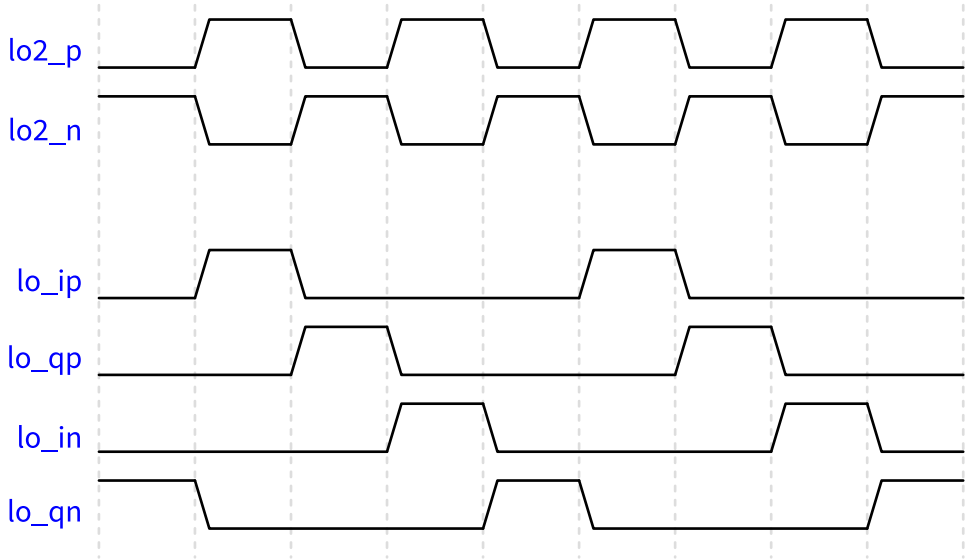

Figure 51: Input and output waveforms of I/Q generation with a divide-by-2 and 25% duty cycle generation.

The advantage of using flip-flops to generate multi-phase LO signals is that very precise phase shifts can be achieved, which can be produced over a very wide frequency range. They can be implemented in different logic styles, like CMOS, ECL, CML, and so on. The disadvantage

is that we need a high-frequency clock signal above the target frequency, which might be difficult to generate. Still, I/Q generation with flip-flops is the de-facto standard approach in modern RFICs.

5.5.4 Delay-Based Phase Generation

The final approach we want to mention is to use a delay line (or multiple ones) to generate phase shifts. Using transmission lines, this approach is widely used at very high frequencies, where other approaches (like the ones shown in Section 5.5.2 or Section 5.5.3) are difficult to implement. One well-known approach is the 90° hybrid coupler, realized with $\lambda/4$ transmission lines [3]. When the frequencies are high enough (and the resulting wavelengths short enough), this approach can even be implemented on-chip. The disadvantage is that the phase shift is frequency-dependent, so the quadrature signals are only perfectly in phase at one frequency. Furthermore, this passive network has a 3 dB loss, so we need additional gain stages to compensate for this loss. A branch-line hybrid coupler is shown in Figure 52.

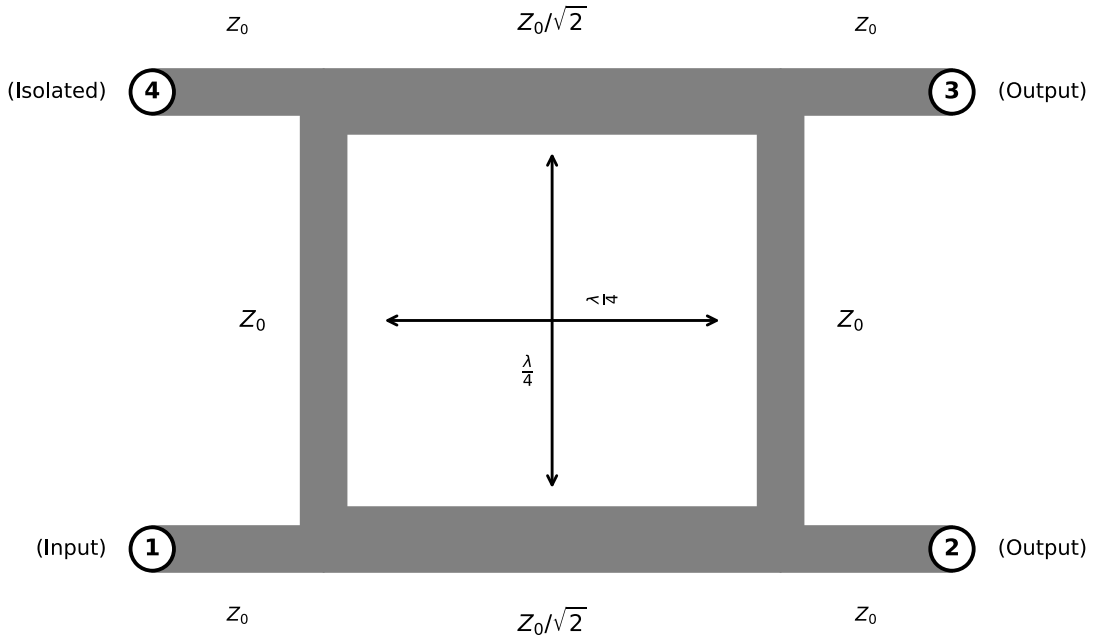

Figure 52: Branch-line hybrid coupler schematic showing the transmission line structure with characteristic impedances and λ/4 length sections.

The 3-port S-parameter matrix of an ideal branch-line coupler (assuming port 4 is poperly terminated with Z_0) is given by

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -j & -1 \\ -j & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}. \tag{39}$$

A feature of the coupler in Figure 52 is that it is single-ended; a differential implementation needs either two couplers or a coupler with subsequent baluns.

Another approach, which is similar to the flip-flop based approach, is to use a delay line to generate the required phase shifts. This can be done with a single delay line and multiple taps, or with multiple delay lines. The idea is that we delay the input clock signal by a certain amount of time, which corresponds to a 90° phase shift at the target frequency. This approach is shown in Figure 53. Since the delay of a delay line is frequency dependent, and the delay will change with process, voltage, and temperature (PVT) variations, we have to implement a tuning mechanism to adjust the delay. This implementation is called a delay-locked loop (DLL), which is similar to a phase-locked loop (PLL, see Section 7).

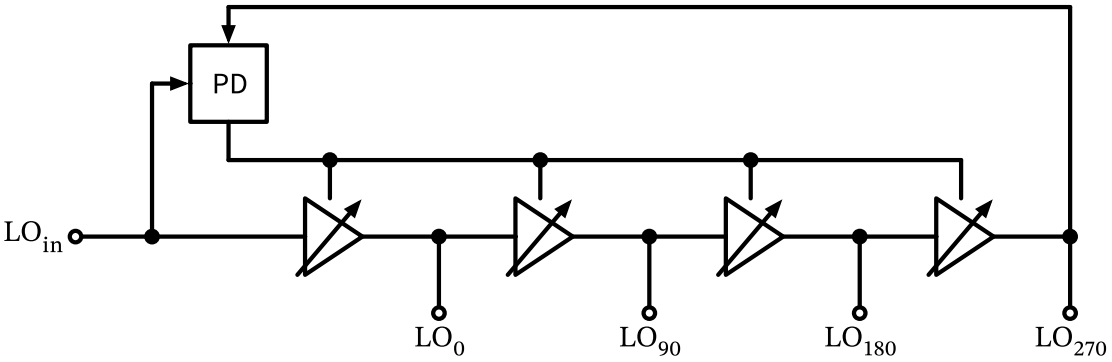

Figure 53: LO multiphase generation by delay-locked loop (DLL). An extension to more than four phases is straightforward.

With a phase detector (PD) we can compare the phase of the output signal with the input signal, and regulate the delay per stage so that after four delays the edges line up. Then we can tap the equally-spaced phases and use them as LO signals. The advantage of this approach is that we can flexibly create a required number of phases from an input clock. This feature is also used in wireline communication systems to generate multiple phases for the data sampling.

6 Oscillators

For the generation of the LO frequency to be used in a mixer for frequency conversion, oscillators are used. Ideally, oscillators produce a stable, noise-free sinusoidal signal, independent of environmental conditions like temperature or supply voltage variations. The circuit symbol of an oscillator is shown in Figure 54.

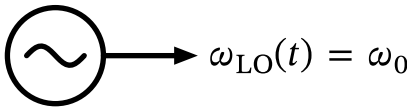

Figure 54: Oscillator symbol.

The question is how to construct an oscillator. In summary, we need to build something that oscillates, i.e., produces a sustained periodic signal with frequency ω_0. One way to achieve

this is to construct a feedback loop where $|H(s = j\omega_0)| = 1$ and $\angle H(s = j\omega_0) = n \cdot 2\pi$; these conditions are called the "Barkhausen criterion" (here, $H(s)$ is the loop gain around the feedback loop). A ring oscillator is one example of such a feedback oscillator. For the 3-stage single-ended *ring oscillator* shown in Figure 55, each inverter provides a phase shift of $2\pi/3$ at the oscillation frequency ω_0, resulting in a total phase shift of 2π for the three inverters. The gain condition is fulfilled by the gain of the inverters, which must be larger than unity to compensate for losses in the loop. By using an odd number of inverters, a stable locking point at dc is avoided.

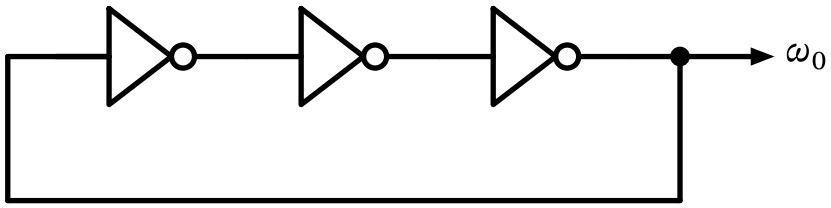

Figure 55: Three stage single-ended ring oscillator.

Note that the output frequency of the ring oscillator shown above is ill-controlled, as it only depends on the delay (i.e., phase shift) of the inverters, which are usually dependent on process, voltage, and temperature (PVT) variations. Also, the inherent quality factor of such an oscillator is low, as exemplified by the definition of the quality factor Q as [13]

$$Q = 2\pi \frac{\text{Average energy stored}}{\text{Energy loss per cycle}} = \frac{\omega_0}{\Delta\omega} = \frac{\omega_0}{2}\sqrt{\left(\frac{dA}{d\omega}\right)^2 + \left(\frac{d\varphi}{d\omega}\right)^2} \tag{40}$$

with a ring oscillator having very low $dA/d\omega$ and $d\varphi/d\omega$ slopes ($A(\omega)$ and $\varphi(\omega)$ are the open loop gain and phase shift, respectively). It can be shown that $Q \approx 1.3$ for a 3-stage ring oscillator [13]. For some applications, this might be sufficient, but in many cases, a higher Q is desired to reduce phase noise and improve frequency stability (as we will see later in this chapter).

According to Equation 40, a high Q can be achieved by using a resonator with high energy storage capability and low energy loss per cycle. We then add an amplifier in a feedback loop to compensate for the losses of the resonator. This principle is shown in Figure 56 for a parallel LC tank circuit as the resonator. Sometimes the action of the feedback loop around the amplifier is modelled as a negative resistance $-R_{\text{amp}}$ that compensates for the losses of the tank circuit represented by the resistor R_{p}. The single parallel resistor R_{p} models the losses of the inductor L and the capacitor C.

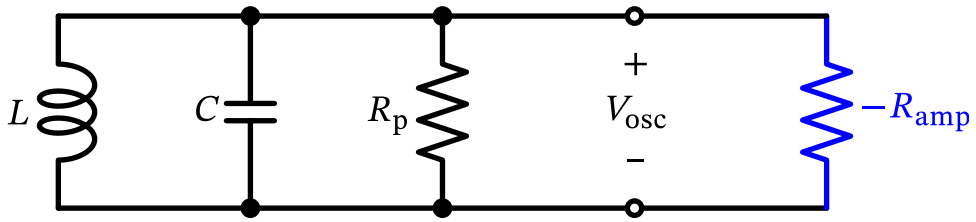

Figure 56: LC parallel tank with connected negative resistance forming an LC oscillator.

When $\left|-R_{\mathrm{amp}}\right| = R_{\mathrm{p}}$, the losses of the tank circuit are fully compensated, and we have a sustained oscillation at the resonance frequency $\omega_0 = 1/\sqrt{LC}$. In practice, the amplifier gain is usually set slightly higher than required for loss compensation to **start the oscillation from noise**. Then, as the oscillation amplitude increases, some **non-linear mechanism in the amplifier reduces the effective gain** until a stable oscillation amplitude is reached. Various implementations of such LC oscillators exist, which will be discussed in the following sections.

- An alternative implementation of setting the oscillation amplitude is to use an *automatic level control* (ALC) loop, which measures the oscillation amplitude and adjusts the amplifier gain accordingly to maintain a constant output amplitude. This approach can improve the stability of the oscillation amplitude over process, voltage, and temperature variations.
- As an alternative to start the oscillation from thermal noise (which can take considerable time depending on the Q of the resonator) is to provide a known initial condition to the resonator, e.g., by pre-charging the capacitor C to a certain voltage before enabling the amplifier. This way, the oscillation can start immediately from this initial energy stored in the resonator.

As a first hint on how to optimize an LC oscillator for low phase noise, we can try to maximize Q according to Equation 40. The energy stored in the capacitor at peak voltage is

$$E_{\mathrm{stored}} = \frac{C V_{\mathrm{osc,p}}^2}{2}$$

so to maximize this we should maximize both C and the peak oscillation voltage $V_{\mathrm{osc,p}}$. On the other hand, the energy loss per cycle is related to the power dissipated in the resistor R_{p} by

$$E_{\mathrm{loss}} = \frac{V_{\mathrm{osc,p}}^2}{2R_{\mathrm{p}}} \cdot \frac{1}{f_0}$$

which we can minimize by maximizing R_{p}. In summary, to maximize Q we should use a large capacitance C, a high oscillation amplitude $V_{\mathrm{osc,p}}$, and a high tank resistance R_{p}. Note that increasing C will reduce L for a given ω_0. Calculating Q from these expressions yields

$$Q = 2\pi \frac{E_{\mathrm{stored}}}{E_{\mathrm{loss}}} = \omega_0 C R_{\mathrm{p}}$$

which confirms the above observations.

Fundamentally, if we describe the output voltage of the oscillator as

$$v_{\mathrm{osc}}(t) = V_{\mathrm{osc,p}}(t) \cdot \cos[\omega_0(t)t + \varphi(t)]$$

we want to keep the amplitude variations small, i.e., $V_{\mathrm{osc,p}}(t) = V_{\mathrm{osc,p}}$, and also the frequency variations small, i.e., $\omega_0(t) = \omega_0$. We further want to minimize any phase fluctuations, i.e., $\varphi(t) = \varphi_0$.

As a side note, the definition of Q in Equation 40 can also be understood as describing how many oscillation cycles it takes until the energy stored in the resonator is dissipated. For example, if $Q = 1000$ at $f_0 = 1\,\mathrm{GHz}$, the energy stored in the resonator will last for approxi-

mately 1000 cycles, which is 1 µs. This time duration is sometimes called the "ring-down time" of the resonator.

6.1 Oscillator Noise

For calculating the noise of an oscillator, we assume that an LC-based oscillator as shown in Figure 56 is used. We assume that the oscillator is in steady-state operation, i.e., $-R_{\mathrm{amp}} = R_{\mathrm{p}}$. We can then simplify the circuit to the one shown in Figure 57.

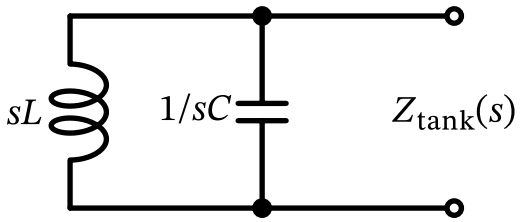

Figure 57: LC parallel tank with oscillator in steady-state operation.

We can calculate

$$Z_{\mathrm{tank}}(s) = \frac{sL\frac{1}{sC}}{sL + \frac{1}{sC}} = \frac{sL}{1 + s^2LC}.$$

At $s = j\omega_0$, we have $Z_{\mathrm{tank}}(s) \to \infty$, so let's approximate around ω_0:

$$Z_{\mathrm{tank}}(j\omega_0 + j\Delta\omega) = \frac{j(\omega_0 + \Delta\omega)L}{1 - (\omega_0 + \Delta\omega)^2 LC}$$

For $\Delta\omega \ll \omega_0$, we can approximate $(\omega_0 + \Delta\omega)^2 \approx \omega_0^2 + 2\omega_0\Delta\omega$ and $\omega_0 + \Delta\omega \approx \omega_0$, which yields (using $\omega_0^2 LC = 1$)

$$Z_{\mathrm{tank}}(j\omega_0 + j\Delta\omega) = -\frac{j}{2\Delta\omega C}.$$

We now use the correspondence $C = 1/\omega_0^2 L$ and express $L = R_{\mathrm{p}}/\omega_0 Q$ to get

$$|Z_{\mathrm{tank}}(j\omega_0 + j\Delta\omega)| = \frac{R_{\mathrm{p}}}{2Q}\left(\frac{\omega_0}{\Delta\omega}\right)$$

which provides us with an expression for the magnitude of the tank impedance around resonance. We now calculate the noise power if a noise current is injected into this impedance. We use the single-sided noise current of R_{p} (see Section 2.3.1), increased by a factor F contributed by the active circuit providing $-R_{\mathrm{amp}}$:

$$\overline{I_n^2} = \frac{4kTF}{R_{\mathrm{p}}}$$

The noise voltage across the tank at a frequency offset $\Delta\omega$ from ω_0 is then

$$\overline{V_n^2}(\Delta\omega) = |Z_{\mathrm{tank}}(j\omega_0 + j\Delta\omega)|^2 \cdot \frac{4kTF}{R_{\mathrm{p}}} = \frac{kTFR_{\mathrm{p}}}{Q^2}\left(\frac{\omega_0}{\Delta\omega}\right)^2$$

This absolute noise voltage is not of much interest per se. We normalize it to the oscillation amplitude V_p, and only account for 1/2 of the noise, as the total noise power is split equally into amplitude noise and phase noise [13], and we are only interested in the **phase noise**. This is because we assume the amplitude noise is removed by amplitude clipping in the LO chain routing the oscillator signal to the mixer. This is often a valid assumption. We introduce the symbol $\mathcal{L}\{\cdot\}$ to denote this normalized phase noise of the oscillator:

$$\mathcal{L}\{\Delta\omega\} = \frac{\frac{1}{2}\frac{kTFR_\mathrm{p}}{Q^2}\left(\frac{\omega_0}{\Delta\omega}\right)^2}{\left(\frac{V_\mathrm{p}}{\sqrt{2}}\right)^2} = \frac{kTFR_\mathrm{p}}{V_\mathrm{p}^2} \cdot \frac{1}{Q^2} \cdot \left(\frac{\omega_0}{\Delta\omega}\right)^2 \tag{41}$$

This equation is known as "**Leeson's equation**" [14], and it provides us with important insights on how to design low phase noise oscillators:

1. Use a resonator with high quality factor Q to reduce phase noise. On-chip LC tanks usually provide moderate Q values (e.g., 10 to 30), while off-chip crystals can provide very high Q values (e.g., 10,000 to 100,000).
2. Maximize the oscillation amplitude V_p to increase the stored energy in the resonator. In CMOS implementations, this is often limited by the supply voltage and/or device breakdown voltages.
3. Use an active circuit with low noise factor F to minimize the noise contribution of the negative resistance $-R_\mathrm{amp}$. A good value to aim for is $F \approx 2$ or lower.
4. Maximize the tank resistance R_p to minimize the thermal noise contribution. This can be achieved by using high-Q inductors and low-loss capacitors.

The phase noise $\mathcal{L}\{\Delta\omega\}$ is expressed in the unit of dBc/Hz, i.e., in decibels of phase noise relative to the carrier power per 1 Hz of bandwidth. When expressing the phase noise of an oscillator in dBc/Hz it is important to state both the oscillation frequency ω_0 and the offset frequency $\Delta\omega$ at which the phase noise is evaluated. For example, we could say that an oscillator has a phase noise of –137 dBc/Hz at 3 MHz offset from a carrier frequency of 2 GHz.

The phase noise expressed with Equation 41 describes an important region of the total oscillator phase noise where the phase noise decreases with $1/(\Delta\omega)^2$, i.e., with 20 dB per decade. This region is often called the "thermal noise region" as it is dominated by thermal noise from the tank resistor and the active circuit. However, at lower offset frequencies, other noise mechanisms (like flicker noise) can dominate, leading to different slopes of the phase noise vs. offset frequency curve. At larger offset frequencies, the phase noise can flatten out due to thermal noise floor limitation of buffer amplifiers following the oscillator. A typical phase noise plot of an LC oscillator is shown in Figure 58.

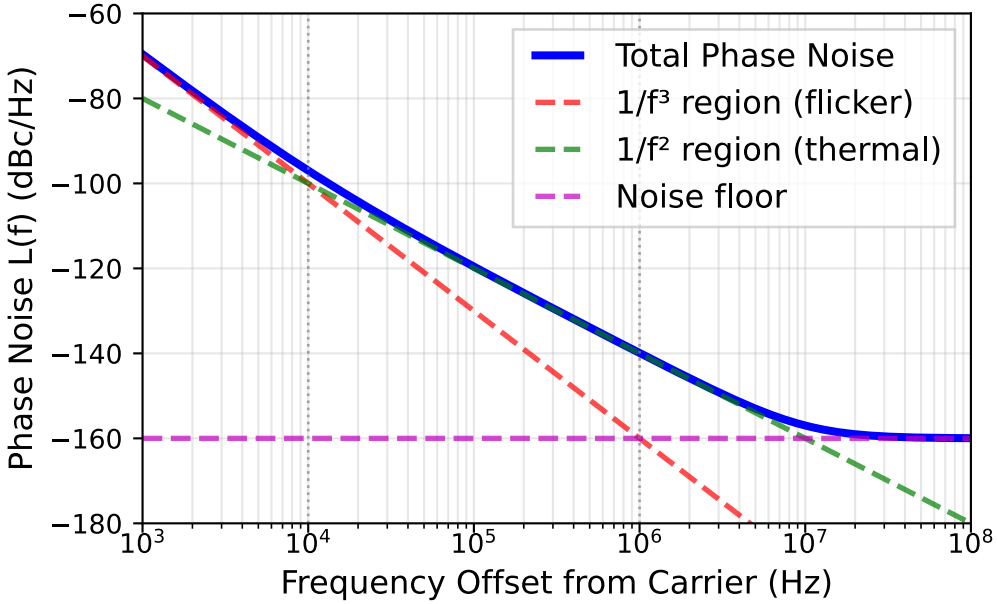

Figure 58: Phase noise spectrum of a typical LC oscillator showing characteristic $1/f^3$ and $1/f^2$ slopes vs. frequency offset from carrier. The flicker noise corner (where the $1/f^3$ region transitions to the $1/f^2$ region) is around 10 kHz, and the thermal noise floor onset is marked at 10 MHz.

It is important to note that the phase noise does not go to infinity as $\Delta\omega$ approaches zero, as Equation 41 might suggest. This is an artifact of the simplified model used to derive Leeson's equation. In reality, the phase noise has the form of a Lorentzian spectrum around the carrier frequency, which means that it flattens out at very low offset frequencies with a trend described by

$$\mathcal{L}\{\Delta\omega \ll\} \propto \frac{1}{\omega_{\mathrm{B}}^2 + \Delta\omega^2}$$

with ω_{B} being the half-power bandwidth of the oscillator spectrum.

Why is phase noise important?

1. In RX and TX the phase noise of the LO creates jitter, which degrades signal quality and contributes to EVM degradation.
2. In TX, phase noise of the LO creates noise sidebands around the carrier, which can lead to adjacent channel interference together with spectral regrowth due to nonlinearities and thermal noise.
3. In RX, reciprocal mixing of the phase noise sidebands with the desired signal creates in-band noise, which degrades the SNR and sensitivity of the RX.
4. In systems using multiple carriers or carriers with high-order modulation schemes (like in OFDM), phase noise can lead to inter-carrier interference (ICI) and symbol misinterpretation, further degrading system performance.

6.2 Reciprocal Mixing

Reciprocal mixing is a phenomenon that occurs in receivers when the phase noise sidebands of the local oscillator mix with strong adjacent channel signals, resulting in in-band noise that

degrades the signal-to-noise ratio of the desired signal. This effect is particularly pronounced in systems with high-order modulation schemes or closely spaced channels, where even small amounts of phase noise can lead to significant performance degradation.

To analyze reciprocal mixing, we consider a scenario where a strong interferer is present at a frequency offset Δf from the desired signal. The phase noise of the local oscillator at this offset frequency can be characterized by its power spectral density $\mathcal{L}\{\Delta f\}$. When the local oscillator mixes with the interferer, the phase noise sidebands effectively "fold" into the desired signal band, creating additional noise. This noise level can be estimated by multiplying the power of the interferer by the phase noise level at the offset frequency and considering the channel bandwidth B of the RX

$$P_{\mathrm{RM}} = P_{\mathrm{interferer}} + \mathcal{L}\{\Delta f\} + 10\log_{10}(B) \tag{42}$$

where P_{RM} is the power of the reciprocal mixing noise introduced into the desired signal band.

i Note 7: Reciprocal Mixing Example

Let us assume the following example from a Bluetooth LE receiver: The sensitivity target for the RX is −70 dBm with an SNR of 10 dB for a 1 MHz channel. An adjacent channel interferer is present at −27 dB channel to interferer ratio at an offset of 3 MHz. How much phase noise can the LO have to meet the sensitivity target?

From the sensitivity target and the SNR requirement, we can calculate the maximum allowable noise floor in the RX. We add a margin of 3 dB to account for implementation losses:

$$P_{\mathrm{noise,max}} = P_{\mathrm{sens}} - \mathrm{SNR} - P_{\mathrm{margin}} = -70\,\mathrm{dBm} - 10\,\mathrm{dB} - 3\,\mathrm{dB} = -83\,\mathrm{dBm}$$

The interferer power is:

$$P_{\mathrm{interferer}} = P_{\mathrm{sens}} - \mathrm{C/I} = -70\,\mathrm{dBm} + 27\,\mathrm{dB} = -43\,\mathrm{dBm}$$

Using Equation 42, we can express the maximum allowable phase noise at 3 MHz offset at 2.4 GHz as:

$$\mathcal{L}\{3\,\mathrm{MHz}\} = P_{\mathrm{noise,max}} - P_{\mathrm{interferer}} - 10\log_{10}(B) = -83\,\mathrm{dBm} + 43\,\mathrm{dBm} - 60\,\mathrm{dB} = -100\,\mathrm{dBc/Hz}$$

The reciprocal mixing scenario described in Note 7 is visualized in Figure 59.

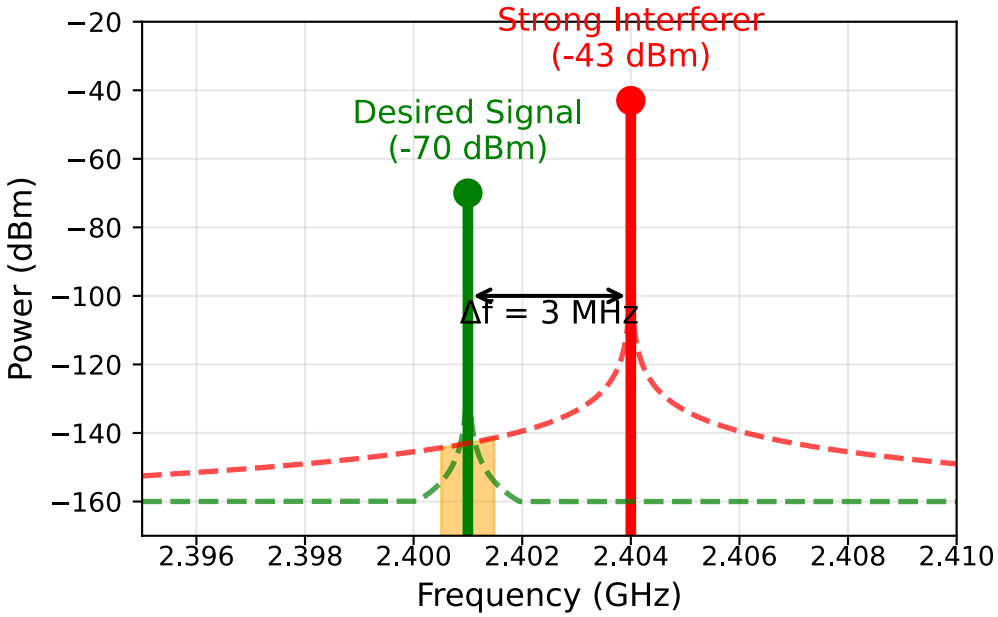

Figure 59: Reciprocal mixing in a receiver: A strong interferer at offset frequency mixes with LO phase noise sidebands, creating in-band noise that degrades the desired signal SNR. The desired signal shows the same, scaled phase noise profile around it as the interferer, as both mix with the same LO phase noise.

6.3 Single-Ended Oscillators

Single-ended oscillators are commonly used in RF applications due to their simplicity and ease of integration, especially in quartz oscillators. A negative resistance is implemented using a single transistor amplifier as is shown in Figure 60. The transistor is configured with two capacitors C_1 and C_2 to provide a phase-shifted feedback path.

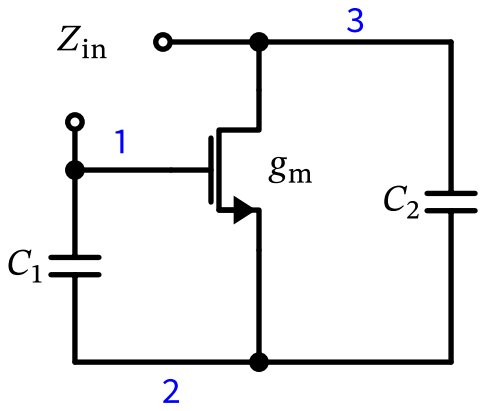

Figure 60: Circuit diagram of a single-ended negative resistance implementation.

It can be shown that the differential input impedance looking into the transistor gate and drain is given by

$$Z_{\text{in}}(s) = -\frac{g_{\text{m}}}{\omega^2 C_1 C_2} + \frac{C_1 + C_2}{sC_1C_2} = -R_{\text{amp}} + \frac{1}{sC_{\text{amp}}}$$

which consists of the series combination of a negative resistance $-R_{\mathrm{amp}}$ and a capacitance C_{amp}. Note that the circuit in Figure 60 does not show a ground symbol. In fact, any of the three nodes marked with blue numbers can be used as a reference node (ground), and this results in the following well-known oscillator topologies:

Reference Node	Oscillator Topology
Node 1 (Gate)	Colpitts oscillator
Node 2 (Source)	Pierce oscillator
Node 3 (Drain)	Clapp oscillator

When we investigate the equivalent electrical circuit of a quartz crystal, we find that it has inductive behavior between its series resonance frequency and its parallel resonance frequency, which are very close together. The quartz crystal equivalent circuit is shown in Figure 61.

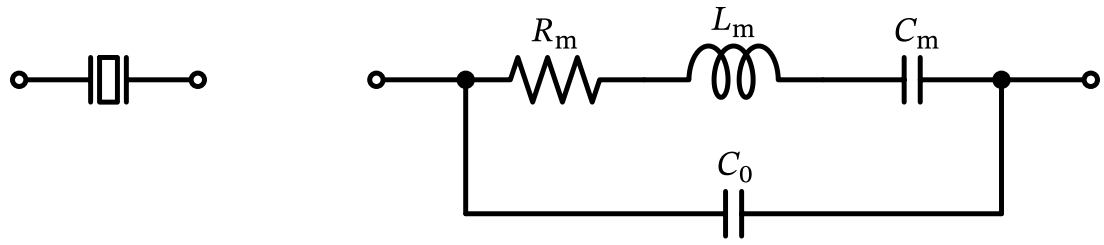

Figure 61: Quartz crystal equivalent circuit.

This means we can operate the quartz crystal as a high-Q inductor in an oscillator circuit, which results in the single-ended quartz crystal oscillator shown in Figure 62, which is a very popular choice for high-performance crystal oscillators. Note that in a simple implementation the current-bias MOSFET can be replaced by an inverter stage biased in the linear region.

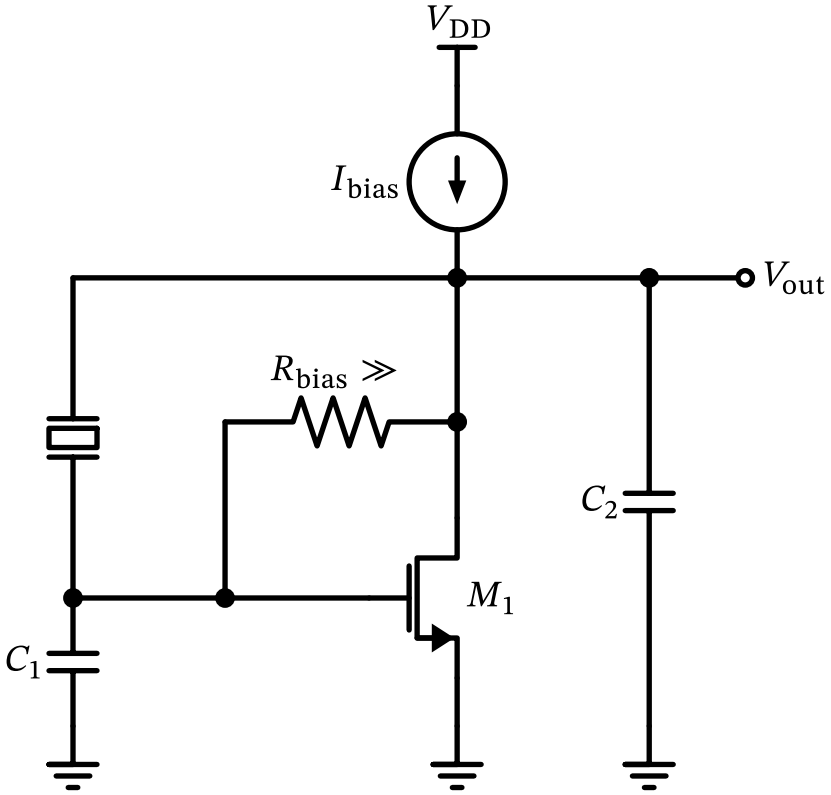

Figure 62: Circuit diagram of a single-ended Pierce crystal oscillator operating the quartz between series and parallel resonance where it acts as a large high-Q inductor. Note that the quartz crystal has no dc path, hence we need a high-ohmic bias resistor to connect M1 into a diode configuration.

The series combination of C_1 and C_2 provides the load capacitance required by the crystal to oscillate at its specified frequency. The oscillation frequency can be approximated by ($C_{\mathrm{load}}^{-1} = 1/C_1 + 1/C_2$):

$$f_0 \approx \frac{1}{2\pi\sqrt{L_{\mathrm{m}} C_{\mathrm{load}}}}$$

6.4 Differential Oscillators

After discussing single-ended oscillators in Section 6.3, we now turn to differential oscillator topologies, which are widely used in integrated LC oscillators due to their superior common-mode noise rejection and reduced even-order harmonics. A popular way to create a differential negative resistance is to use a cross-coupled pair of transistors as shown in Figure 63.

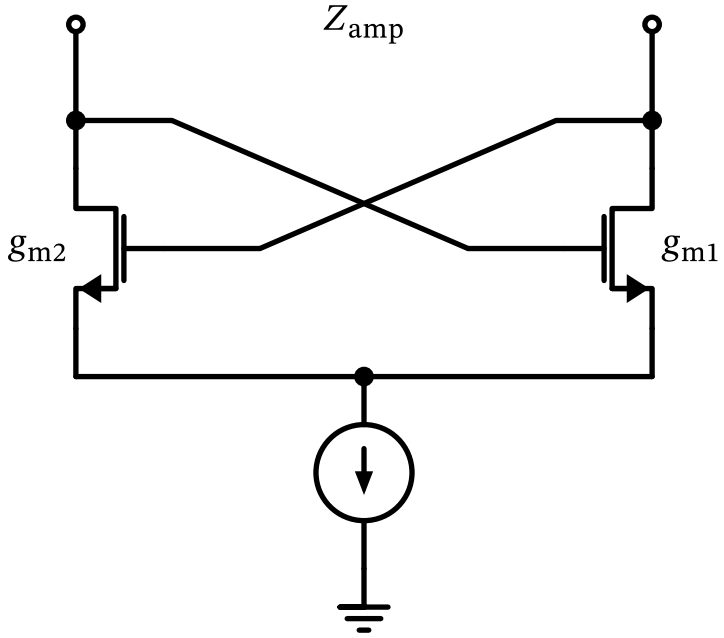

Figure 63: A cross-coupled differential pair.

We assume a symmetrical circuit by setting $g_{\text{m1}} = g_{\text{m2}} = g_{\text{m}}$. By analyzing the small-signal equivalent circuit, we can derive the differential input impedance looking into the gates of the transistors:

$$Z_{\text{in}} = -\frac{2}{g_{\text{m}}}$$

A practical implementation of a differential LC oscillator using the cross-coupled pair is shown in Figure 64 using a so-called “NMOS core”, as it is using an NMOS-based differential pair.

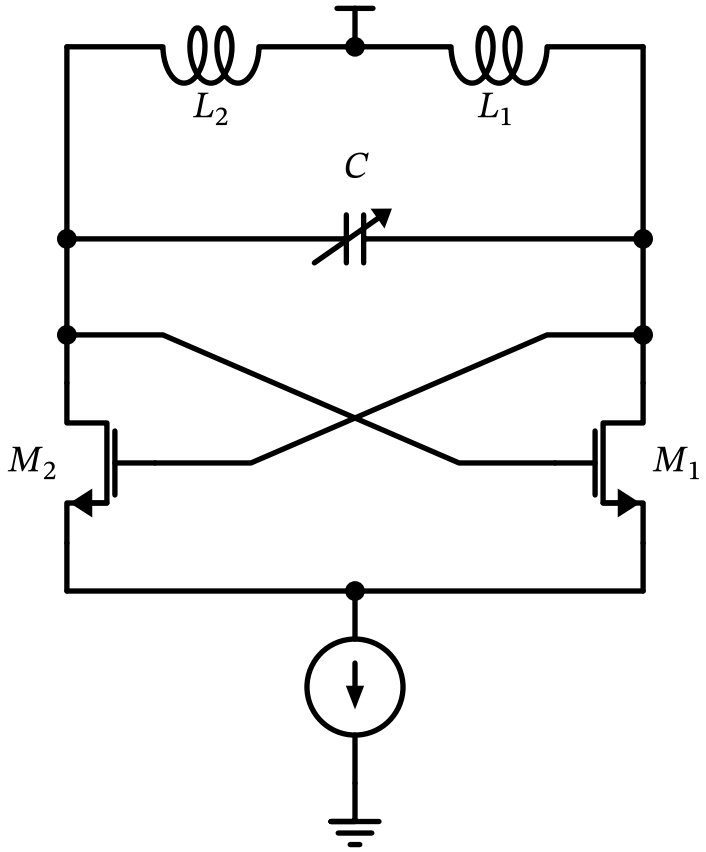

Figure 64: An LC differential oscillator using an NMOS cross-coupled differential pair. L_1 and L_2 are usually implemented as a single on-chip spiral inductor with a center tap.

By putting the circuit in Figure 64 onto its head we obtain a "PMOS core" oscillator, which is also widely used in integrated LC oscillators. This configuration is shown in Figure 65.

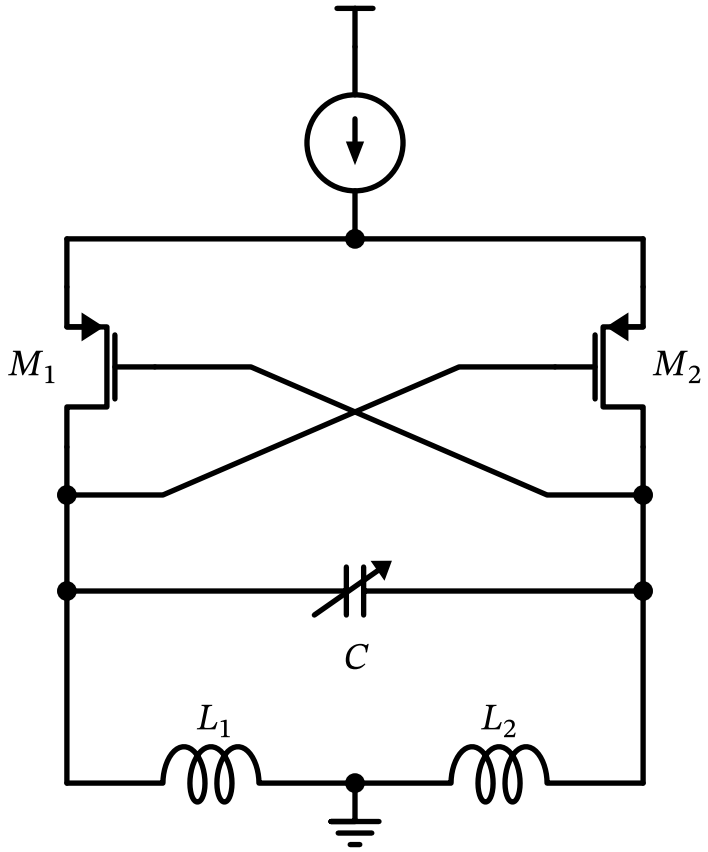

Figure 65: An LC differential oscillator using a PMOS cross-coupled differential pair. L_1 and L_2 are usually implemented as a single on-chip spiral inductor with a center tap.

In both oscillator topologies, the oscillation frequency is determined by the LC tank circuit formed by $L = L_1 + L_2$ and C. The oscillation frequency can be approximated by

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

As both technologies have their inductor middle point tied to a supply rail (ground for NMOS core, V_{DD} for PMOS core), the voltage swing across the tank can go well beyond the supply voltage, which is an advantage of these topologies. However, note that the maximum voltage swing is still limited by the device breakdown voltages of M_1 and M_2, which can be critical in advanced CMOS technologies with low breakdown voltages. To utilize both an NMOS and a PMOS cross-coupled pair in parallel, a so-called "complementary (or CMOS) LC oscillator" can be used, which is shown in Figure 66. Here, the voltage swing across the tank is limited to the supply voltage, so it is inherently safe regarding device breakdown. Also, the transconductance of both NMOS and PMOS devices contribute to the negative resistance, which can reduce power consumption for a given oscillation amplitude.

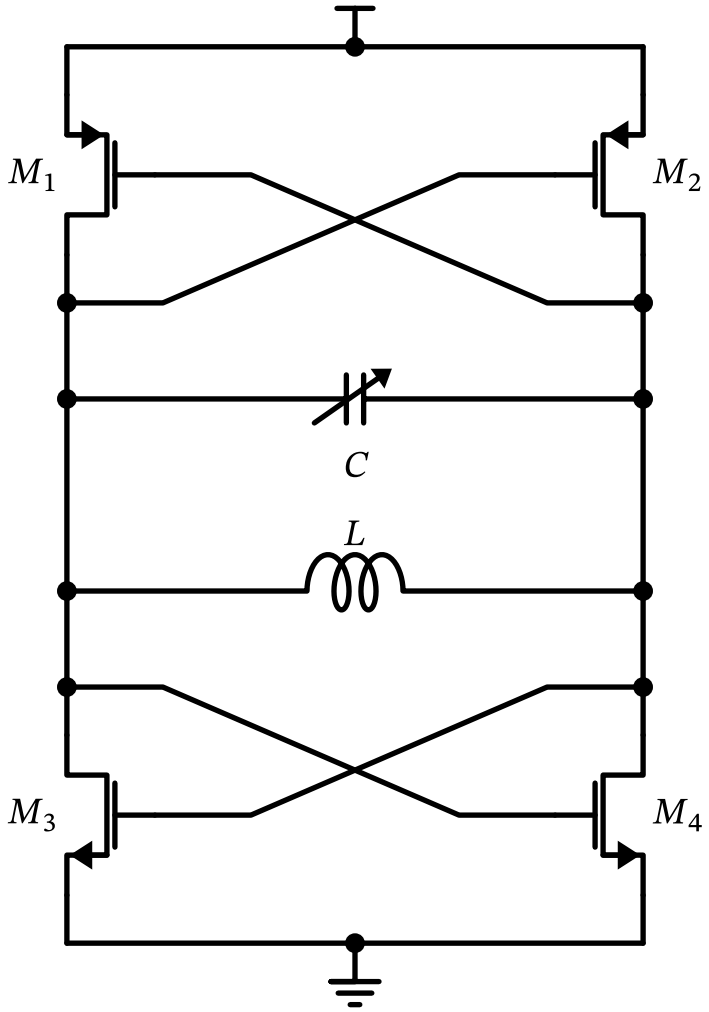

Figure 66: An LC differential oscillator using a PMOS and an NMOS cross-coupled differential pair.

Note that no bias current sources are shown in Figure 66, but can be added in practice. However, they require some voltage headroom, which lowers the maximum voltage swing across the tank, which is negative for phase noise performance according to Leeson's equation in Equation 41.

The inductor L used in LC tanks is often implemented as a differential spiral inductor, which can be realized on-chip using the top metal layers of the CMOS process. If higher Q is sought, then off-chip inductors must be used, which is more expensive and uses package pins to connect the inductor. Alternatively, bondwire inductors can be used, which utilize the bondwires connecting the die to the package leads as inductors. Bondwire inductors can provide high Q values and are a cost-effective solution for improving oscillator performance without requiring off-chip components. For mm-wave frequencies, transmission line stubs can be used as inductors, which can be implemented on-chip using coplanar waveguide (CPW) or microstrip structures.

6.5 Frequency Tuning of Oscillators

Note that all three LC oscillator topologies shown in Section 6.4 have a fixed oscillation frequency determined by the LC tank circuit. In practice, it is often desirable to be able to tune the oscillation frequency over a certain range. There are only two ways to achieve this, either by tuning the inductance L or the capacitance C of the tank circuit. In integrated LC oscillators, it is common to use a fixed inductor and a tunable capacitor, which is shown in Figure 64, Figure 65, and Figure 66.

The need to change the oscillator frequency can arise from various requirements:

1. We need to precisely tune the oscillator frequency to match a desired carrier frequency.
2. The oscillator frequency needs to be adjusted to compensate for process, voltage, and temperature (PVT) variations that can affect the resonant frequency of the LC tank.

In order to achieve a tunable capacitance, a so-called "varactor" (variable capacitor) is often used. A varactor is a semiconductor device that exhibits a capacitance that varies with the applied voltage. Common ways to implement a varactor in integrated circuits are:

1. Use the gate-to-channel capacitance of a MOSFET operated in reverse bias. By changing the voltage applied to the gate, the depletion region width in the channel changes, which in turn changes the capacitance.
2. Use a PN-junction diode operated in reverse bias. The depletion region width of the diode changes with the applied voltage, leading to a change in capacitance.
3. Use a gate-to-channel capacitance of special MOSFET structure in accumulation mode, which can be achieved by using a NMOS situated in an n-well (in contrast to the usual operation in a p-well). Changing the gate-to-channel voltage changes the accumulation capacitance [15].
4. Use a switched capacitor bank, where multiple capacitors are connected in parallel or series using switches (usually MOSFETs) to achieve discrete capacitance values.

In order to characterize the tuning sensitivity of an oscillator, the metric K_{VCO} (voltage-controlled oscillator gain) is often used, which is defined as

$$K_{\mathrm{VCO}} = \frac{df_0}{dV_{\mathrm{tune}}}$$

with V_{tune} being the control voltage applied to the varactor. The unit of K_{VCO} is usually expressed in MHz/V or GHz/V. A high K_{VCO} means that a small change in tuning voltage results in a large change in oscillation frequency, which can be beneficial for wide tuning ranges but can also make the oscillator more sensitive to noise on the tuning voltage line.

Note that the tuning sensitivity K_{VCO} is usually quite nonlinear over the tuning range, so it is common to specify K_{VCO} at a certain operating point or as an average value over the tuning range.

In order to achieve a wide tuning range while maintaining a sufficiently small K_{VCO} for phase noise reasons, a combination of coarse and fine tuning mechanisms can be used [16]. For example, a switched capacitor bank can provide coarse tuning steps, while a varactor can provide fine tuning within each step. It is important to ensure that the overall tuning range is free of dead zones, where the oscillator cannot be tuned to certain frequencies due to non-overlapping tuning ranges of the coarse and fine tuning elements.

While there are many ways to implement a switched capacitor for use in an oscillator tuning circuit, one popular way is shown in Figure 67. Many such switched capacitors with different values of C (often binary weighted) can be combined to form a capacitor bank for coarse frequency tuning.

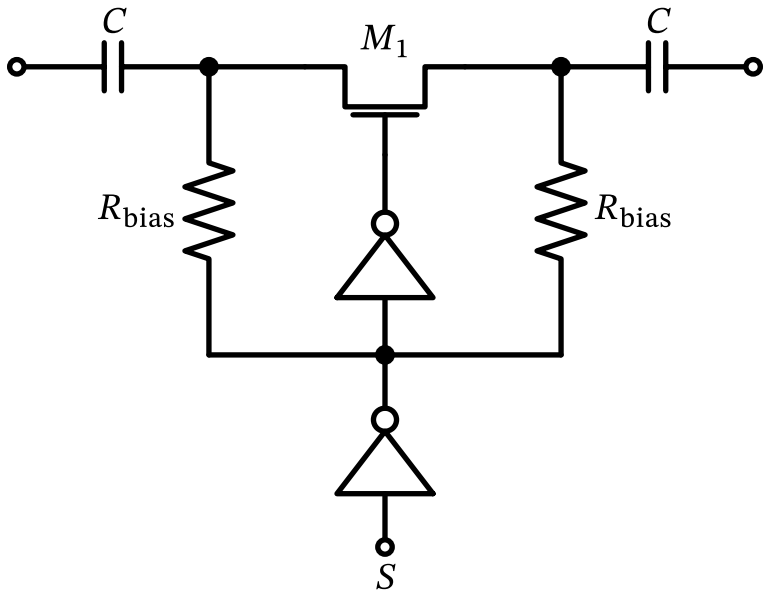

Figure 67: An switched capacitor for use in an oscillator switched capacitor tuning bank. The bias resistors tie the drain/source nodes to ground during turn on of M_1 (for low on resistance), while they tie the drain/source nodes to VDD during turn off of M_1 to prevent accidential turn on of the drain/source to bulk diodes of M_1.

Here, when the control signal S is high, the effective capacitance is $C/2$, with a parasitic series resistance R_{on} due to the switch. When S is low, the effective capacitance is C_{off}, which is the parasitic drain-source capacitance of the MOSFET switch.

Note that there exists a trade-off when designing the switched capacitor: A larger switch (with increased W) reduces R_{on}, but increases C_{off}. A large R_{on} leads to increased losses in the tank circuit, which degrades the quality factor Q and increases phase noise according to Equation 41. On the other hand, a large C_{off} reduces the effective tuning range of the switched capacitor, which can be detrimental if a wide tuning range is required.

In **digitally-controlled oscillators (DCOs)**, the tuning voltage V_{tune} is replaced by a digital control word that selects different capacitance values from a capacitor bank. This approach allows for precise and repeatable frequency tuning, which is beneficial in applications requiring frequency synthesis or channel selection. These fine tuning steps can also be implemented according to Figure 67, however, the value of C must be sufficiently small.

6.6 Oscillator Modelling

The instantaneous frequency of an oscillator can be expressed as

$$\omega_{\text{VCO}}(t) = \omega_0 t + K_{\text{VCO}} \cdot V_{\text{tune}}(t)$$

where ω_0 is the nominal oscillation frequency, K_{VCO} is the tuning sensitivity, and $V_{\mathrm{tune}}(t)$ is the tuning voltage applied to the varactor. Looking at the instantaneous phase $\varphi_{\mathrm{VCO}}(t)$ of the oscillator, we can integrate the instantaneous frequency $\omega_{\mathrm{VCO}}(t)$ to obtain

$$\varphi_{\mathrm{VCO}}(t) = \int_0^t \omega_{\mathrm{VCO}}(\tau)d\tau = \omega_0 t + K_{\mathrm{VCO}} \int_0^t V_{\mathrm{tune}}(\tau)d\tau.$$

Inspecting this equation, we see that with respect to phase, an oscillator is a **perfect** integrator of the tuning voltage over time! For simulation purposes, we can therefore model an oscillator as an integrator block as shown in Figure 68.

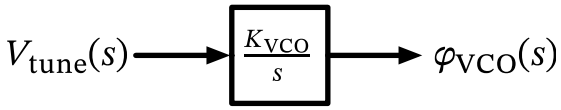

Figure 68: A model of a VCO as a perfect integrator for the excess phase in the s-domain.

In this model we look at the phase output of the oscillator as an *excess phase* with respect to the nominal phase $\omega_0 t$. This delta phase is obtained by integrating the tuning voltage $V_{\mathrm{tune}}(t)$ scaled by K_{VCO}.

7 Phase-Locked Loops

Thinking about oscillators, we face a dilemma:

- Good phase noise performance is reached with high Q in the resonator (see Equation 41).
- Oscillator tunability requires a tunable resonator, which usually results in a low-to-moderate Q.
- Oscillators have inherent frequency stability issues due to temperature variations, device aging, and supply voltage fluctuations.

7.1 Basic PLL Architecture

We solve this dilemma by using a **phase-locked loop (PLL)** to stabilize and control the frequency of a tunable oscillator, mostly in the form of a VCO. A PLL is a feedback control system that locks the phase of the VCO output to the phase of a reference signal, typically generated by a crystal oscillator with excellent frequency stability. The block diagram of a basic PLL is shown in Figure 69.

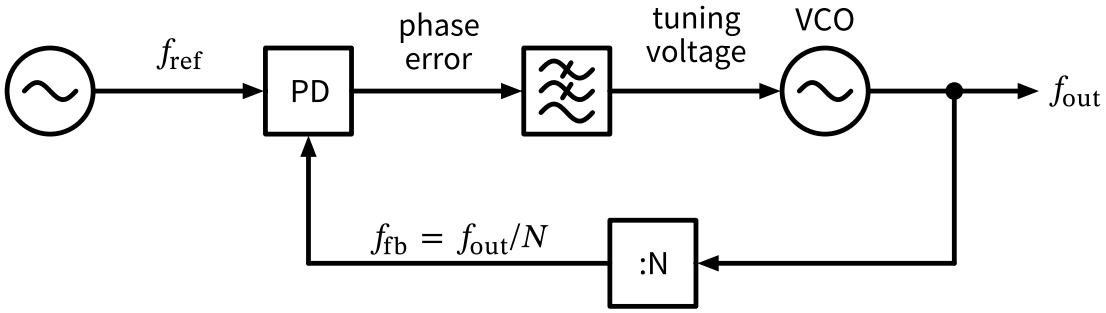

Figure 69: Block diagram of a PLL. A reference phase signal is compared to the phase of a VCO output signal in a phase detector. The phase error is low-pass filtered and used to tune the VCO frequency.

A reference frequency (often coming from a high-Q crystal oscillator) is compared to the phase of a VCO output signal in a phase detector (PD). The phase error is low-pass filtered and used to tune the VCO frequency. By continuously adjusting the VCO based on the phase difference, the PLL ensures that the VCO output remains synchronized with the reference signal, effectively combining the stability of the reference with the tunability of the VCO. When the output signal of the VCO is fed back to the phase detector through a frequency divider (with division ratio N), the PLL can generate output frequencies that are integer multiples of the reference frequency, given by $f_{\text{out}} = N \cdot f_{\text{ref}}$.

The PD compares the phase of the reference signal with the phase of the VCO output signal. It works in the time domain and produces an output voltage proportional to the arrival time difference (phase difference) between the two input signals. This output voltage, known as the phase error signal, indicates whether the VCO is leading or lagging the reference signal in phase, expressed as

$$\Delta\varphi = 2\pi \frac{\Delta t}{T_{\text{ref}}}$$

and

$$V_{\text{error}} = K_{\text{PD}} \cdot \Delta\varphi$$

where $\Delta\varphi$ is the phase difference (caused by the arrival time difference) in rad, K_{PD} is the phase detector gain in V/rad, Δt is the time difference between the two signals, and T_{ref} is the period of the reference signal.

An important consideration of the PD is its linear operating range. The PD can only provide a linear output voltage for small phase differences, typically within ±180° (±π rad). If the phase difference exceeds this range, the PD characteristic "wraps around," leading to ambiguity in the phase error signal. This can cause the PLL to lose lock or behave unpredictably. Therefore, the PLL design must ensure that the phase difference remains within the linear range of the PD during normal operation, which is especially troublesome if the output frequency f_{out} is not yet close enough to its final steady-state value. Note that a simple PD can be implemented by using an XOR gate for digital signals or a mixer for analog signals.

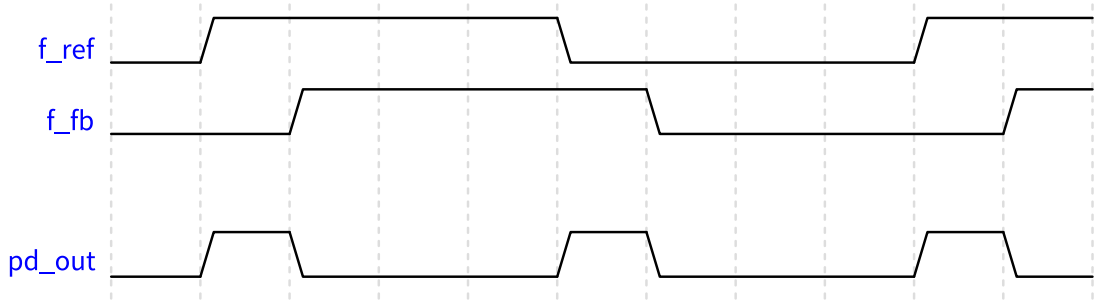

Figure 70: Input and output waveforms of an XOR-based phase-detector.

We can see in Figure 70 that the output of the XOR PD is high when the two input signals differ and low when they are the same. The output duty cycle is proportional to the phase difference between the two input signals. Note that this PD has no notion of frequency, only phase! Still, it can be used to lock the frequency of the VCO to the reference frequency, as

a constant phase difference implies equal frequencies; however, this only works if the initial frequency difference is small enough. Also, this XOR-based PD does not evaluate the edges of the input signals, only their logic states. Therefore, it is crucial that both input signals are square waves with a 50% duty cycle to ensure proper operation of the PD.

As the output signal of the PD contains high-frequency components (at least at twice the reference frequency), a low-pass filter (LPF) is used to smooth the phase error signal before it is applied to the VCO tuning input. The LPF also determines the dynamic response of the PLL, affecting its stability and transient behavior. A well-designed LPF ensures that the PLL can quickly respond to changes in the reference signal or disturbances while maintaining stability and minimizing overshoot or oscillations in the output frequency. Often, the LPF has voltage-mode inputs and outputs to connect to the PD and VCO, respectively.

Using a digital divider in the feedback path to only pass every N-th cycle of the VCO output to the PD allows the PLL to generate output frequencies that are integer multiples of the reference frequency. This is particularly useful in applications such as frequency synthesis, where a wide range of frequencies is required from a single stable reference source. By adjusting the division ratio $N \in \mathbb{N}$, the PLL can produce various output frequencies while maintaining phase lock with the reference signal. However, note that the frequency resolution of the output frequencies Δf_{out} is limited to integer multiples of the reference frequency!

Putting everything together into an s-domain block diagram, the PLL can be modeled as shown in Figure 71.

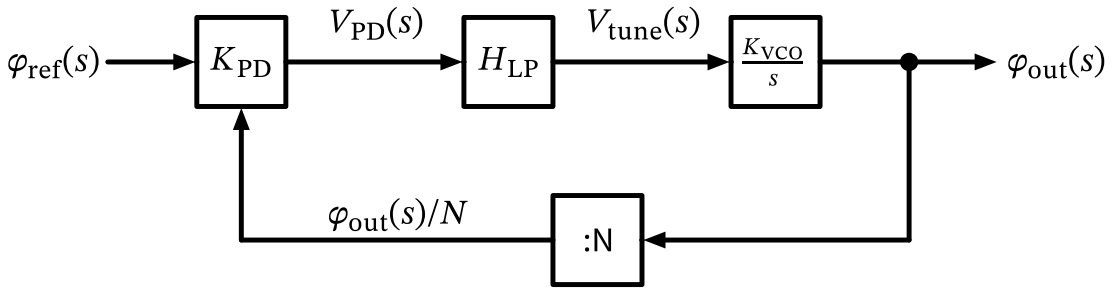

Figure 71: Laplace domain model of a PLL. Note that input and output signals are (excess) phase in rad.

For the loop filter transfer function $H(s)$ we assume a simple first-order low-pass filter with a cutoff frequency of $\omega_{\text{LP}} = 1/T_{\text{LP}}$ given by

$$H_{\text{LP}}(s) = \frac{1}{1 + sT_{\text{LP}}}.$$

! Why Regulate Phase Instead of Frequency?

Before we proceed, it is worth discussing why PLLs regulate the phase of the VCO output rather than its frequency directly. The reason lies in the relationship between phase and frequency: Frequency is the time derivative of phase. By controlling the phase, the PLL inherently controls the frequency as well. Additionally, even if the phase regulation has a steady-state phase error φ_{err}, the frequency error in steady state is **zero**, as can be appreciated when inspecting

$$\omega_{\text{out}}(t) = \frac{d\varphi_{\text{out}}(t)}{dt} = \frac{d}{dt}[N \cdot \varphi_{\text{ref}}(t) + \varphi_{\text{err}}] = N \cdot \frac{d\varphi_{\text{ref}}(t)}{dt} = N \cdot \omega_{\text{ref}}(t).$$

We now derive the closed-loop transfer function of the PLL from reference phase $\varphi_{\text{ref}}(s)$ to output phase $\varphi_{\text{out}}(s)$. First, we note that

$$V_{\text{PD}}(s) = K_{\text{PD}}\left[\varphi_{\text{ref}}(s) - \frac{\varphi_{\text{out}}(s)}{N}\right]$$

and

$$\varphi_{\text{out}}(s) = \frac{K_{\text{VCO}}}{s} \cdot H_{\text{LP}}(s) \cdot V_{\text{PD}}(s) = \frac{K_{\text{VCO}}}{s} \cdot \frac{1}{1 + sT_{\text{LP}}} \cdot K_{\text{PD}}\left[\varphi_{\text{ref}}(s) - \frac{\varphi_{\text{out}}(s)}{N}\right].$$

Not that this equation has one integrator $1/s$ (which we can also see in Figure 71). We call PLLs with one integrator in the loop a **Type-I PLL**. We can now rearrange to find the closed-loop transfer function $H(s)$ as

$$H(s) = \frac{\varphi_{\text{out}}(s)}{\varphi_{\text{ref}}(s)} = N \cdot \frac{K_{\text{VCO}} \cdot K_{\text{PD}} \cdot \omega_{\text{LP}}/N}{s^2 + s \cdot \omega_{\text{LP}} + K_{\text{VCO}} \cdot K_{\text{PD}} \cdot \omega_{\text{LP}}/N}. \tag{43}$$

We can use all techniques known from control theory to analyze the stability and dynamic response of the PLL based on Equation 43, however, we have to keep the following in mind:

- $H(s)$ is a small-signal model of the PLL around its locked operating point, only valid for small perturbations. Large-signal behavior, such as acquisition and lock range, are not captured by this model.
- A PLL is a sampled system, working at instances of T_{ref}. $H(s)$ is a continuous-time approximation, which is valid only if the loop bandwidth is much smaller (typically $1/10$) than the reference frequency f_{ref}. If this approximation is not valid, a discrete-time model of the PLL must be used, deriving the z-domain transfer function $H(z)$.

If we compare the canonical form of a second-order system given by

$$H(s) = K \cdot \frac{\omega_{\text{n}}^2}{s^2 + s \cdot 2\zeta\omega_{\text{n}} + \omega_{\text{n}}^2},$$

with Equation 43, we can identify the natural frequency ω_n and the damping factor ζ of the PLL as

$$\omega_\mathrm{n} = \sqrt{\frac{K_\mathrm{VCO} \cdot K_\mathrm{PD} \cdot \omega_\mathrm{LP}}{N}}$$

and

$$\zeta = \frac{1}{2}\sqrt{\frac{\omega_\mathrm{LP} \cdot N}{K_\mathrm{VCO} \cdot K_\mathrm{PD}}}.$$

This allows us to use standard control theory results to design the PLL parameters K_PD, K_VCO, ω_LP, and N to achieve the desired dynamic response and stability margins. Note that ω_n and ζ are not independent, as they both depend on the same set of PLL parameters! This seriously limits the design of the loop dynamics, and means that we need a more complex loop filter $H_\mathrm{LP}(s)$ to achieve more freedom in selecting ω_n and ζ independently. Often, we want to choose $\zeta = 1/\sqrt{2} \approx 0.707$ to achieve a Butterworth response for a good step response without overshoot.

The poles of a second-order system are given by

$$p_{1,2} = -\omega_\mathrm{n}\left(\zeta \pm \sqrt{\zeta^2 - 1}\right)$$

with $\zeta = 1/\sqrt{2}$ resulting in complex conjugate poles at

$$p_{1,2} = -\frac{\omega_\mathrm{n}}{\sqrt{2}} \pm j\frac{\omega_\mathrm{n}}{\sqrt{2}}.$$

To summarize: We can make a simple PLL using an XOR for a PD, use a first-order LPF, plus a feedback divider to get frequency multiplication of $f_\mathrm{out} = N \cdot f_\mathrm{ref}$. However, the control over the loop dynamics is very limited, and the locking procedure of the PLL is tricky, as the PD has a limited linear range of ±180°, requiring that the VCO frequency at the start of the locking procedure is already close enough to the desired output frequency. In addition, N is an integer, limiting the frequency resolution of the output frequencies to integer multiples of the reference frequency.

The question is how to improve the PLL design to overcome these limitations? We will explore advanced PLL architectures and techniques in the following section.

7.2 Charge-Pump PLL

The fundamental limitations of the simple PLL architecture presented in Section 7.1 can be addressed by using a more sophisticated phase detector known as a **phase-frequency detector (PFD)** in combination with a **charge pump (CP)** [17]. This combination allows for better control over the loop dynamics and improved locking behavior.

In order to drive the CP, we envision a PFD with two digital outputs: An “up” signal that indicates when the VCO phase lags behind the reference phase, and a “down” signal that indicates when the VCO phase leads the reference phase. The PFD compares both the phase and frequency of the reference and feedback signals, providing a more robust locking mechanism. The typical implementation of a PFD is shown in Figure 72.

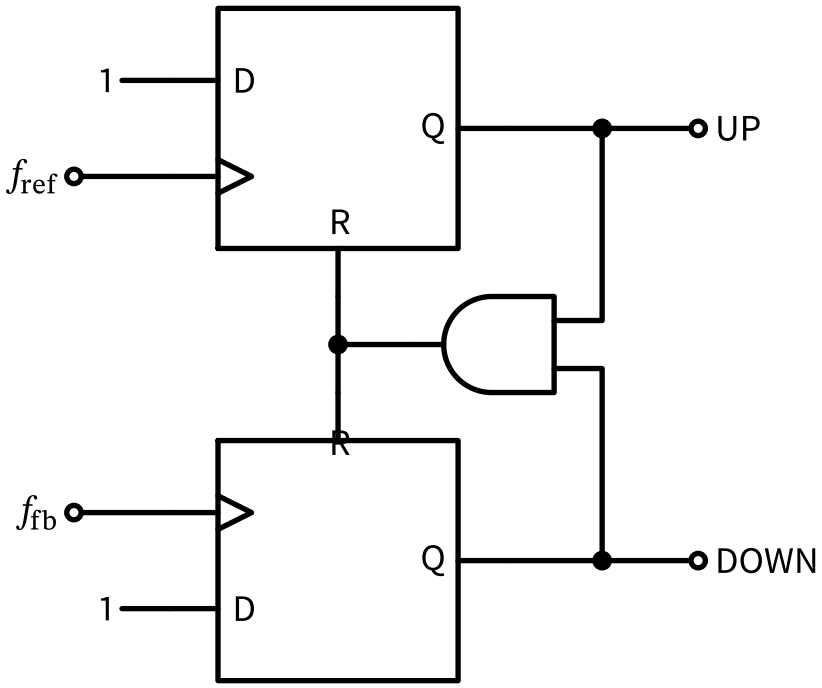

Figure 72: Implementation of a phase-frequency detector.

The PFD operates as follows:

- If the reference signal leads the feedback signal (its rising edge arrives first), the "UP" output is activated, causing the CP to source current into the integration capacitor, *increasing* the VCO tuning voltage and thus the VCO frequency.
- If the feedback signal leads the reference signal (its rising edge arrives first), the "DOWN" output is activated, causing the CP to sink current from the integration capacitor, *decreasing* the VCO tuning voltage and thus the VCO frequency.
- If both signals are aligned in phase and frequency, both outputs are only activated for a short time (the gate delays of the DFF and the AND), and the CP does not source or sink current; the integration capacitor in the loop filter holds its voltage, maintaining the VCO frequency.
- The rising edge that arrives last resets both outputs, ensuring that the CP only sources or sinks current for a duration proportional to the phase difference between the two signals.

In the following, we will look at four different cases of PFD operation to illustrate its behavior.

1. *Reference leads feedback*: The "UP" output is activated, and the CP sources current to the loop filter, increasing the VCO frequency.

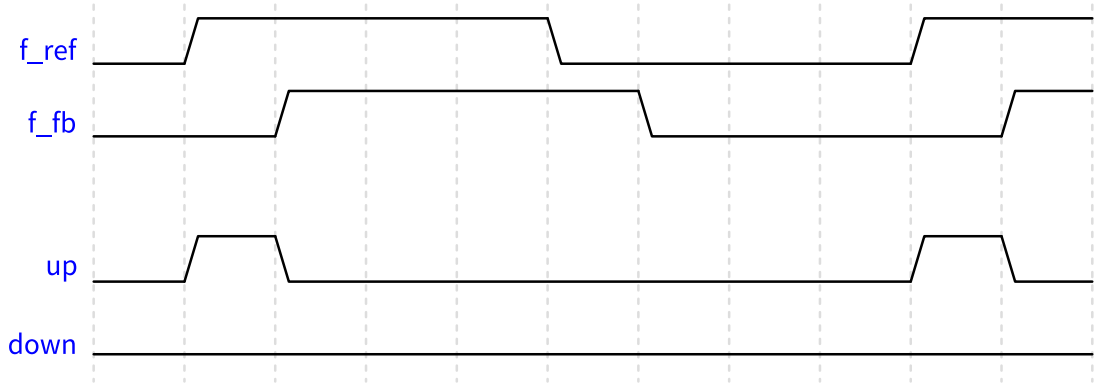

Figure 73: Reference leading the VCO feedback signal, frequencies are already aligned.

2. *Feedback leads reference*: The "DOWN" output is activated, and the CP sinks current from the loop filter, decreasing the VCO frequency.

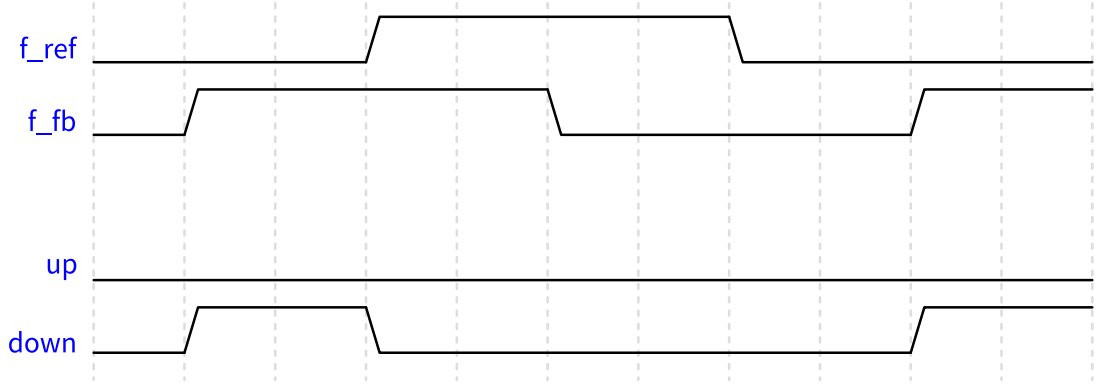

Figure 74: VCO feedback signal leading the reference, frequencies are already aligned.

3. *Both signals aligned*: Both outputs are briefly activated, but the CP does not source or sink current, maintaining the VCO frequency.

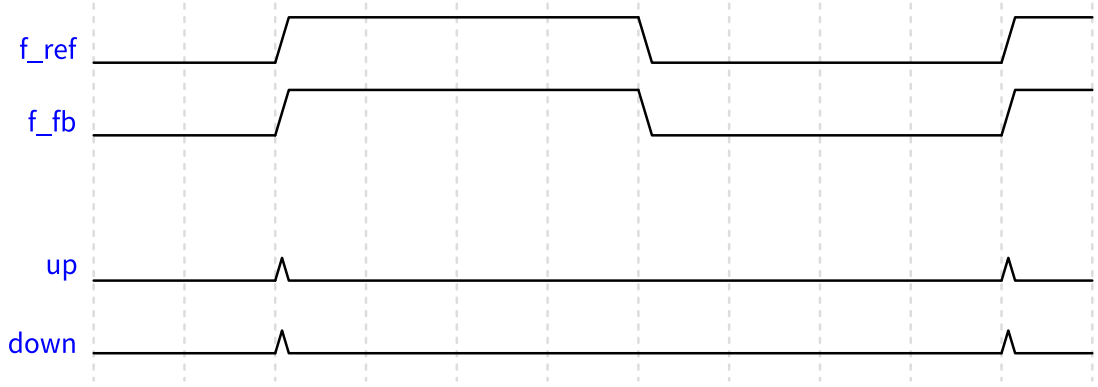

Figure 75: VCO feedback and reference signal are aligned in phase and frequency.

4. *Large frequency difference*: The PFD continues to source or sink current until the phases align, allowing the PLL to acquire lock even with large initial frequency differences.

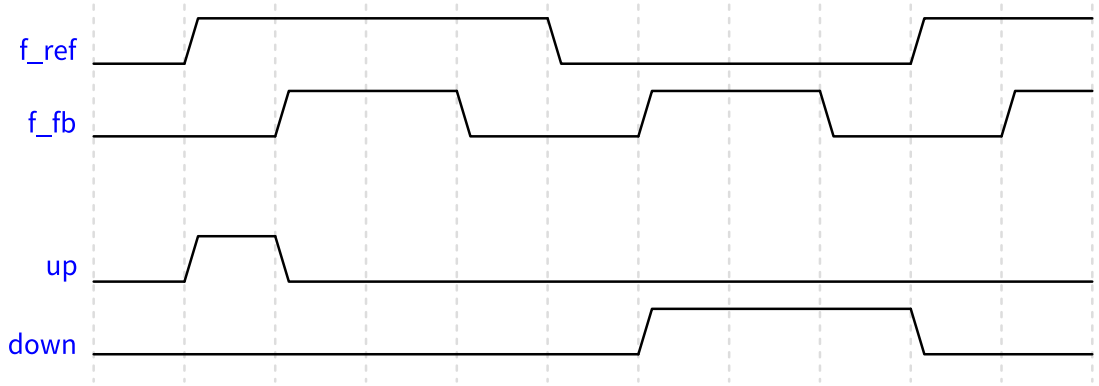

Figure 76: VCO feedback and reference signal have different frequencies (the VCO frequency is too high). There will be some periods with a wrong output (as shown here in the first cycle), but on average the phase-frequency detector output will indicate that the VCO frequency must be decreased.

Let us now investigate the circuit of the CP. The UP and DOWN signals from the PFD control two current sources/sinks connected to the loop filter capacitor, as shown in Figure 77.

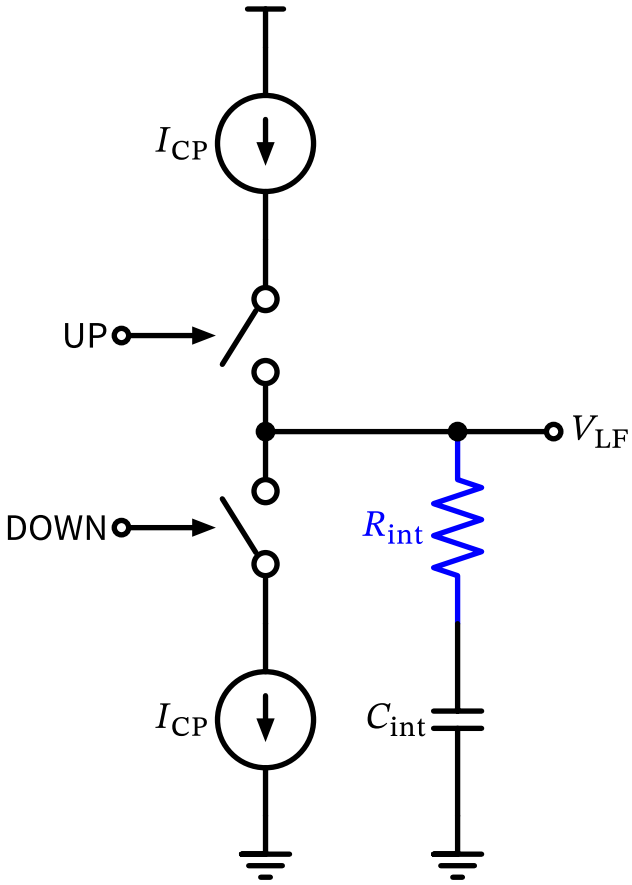

Figure 77: Charge pump consisting of two matched current sources and switches. The resistor in series with the capacitor introduces a zero.

When the UP signal is high, the CP *sources* a constant current I_{CP} into the loop filter capacitor C_{int}, increasing its voltage linearly over time according to the phase error. Conversely, when

the DOWN signal is high, the CP *sinks* a constant current I_{CP} from the capacitor, decreasing its voltage linearly.

Note that the top and bottom current sources must be well matched to ensure symmetric operation of the CP and should have a wide operating range. The output impedance of the CP is ideally very high so that the charge pump current does not depend on the voltage across the loop filter capacitor. Further, the switches introduce charge injection and clock feedthrough, which can cause unwanted ripples on the loop filter voltage, leading to spurious signals in the PLL output. Careful design of the switches and layout techniques are necessary to minimize these effects.

The voltage change on the capacitor can be expressed as

$$\Delta V_{\mathrm{LF}} = \frac{1}{C_{\mathrm{LF}}} \int_0^{t_{\mathrm{on}}} I_{\mathrm{int}} \cdot dt \tag{44}$$

where t_{on} is the duration for which the UP or DOWN signal is active, proportional to the phase difference between the reference and feedback signals.

Taking the Laplace transform of Equation 44, we find the transfer function of the CP from the UP/DOWN control signals to the loop filter voltage as

$$H_{\mathrm{CP}}(s) = \frac{I_{\mathrm{CP}}}{2\pi} \left(\frac{1}{sC_{\mathrm{int}}} + R_{\mathrm{int}} \right) \tag{45}$$

where we have added a resistor R_{int} in series with the capacitor C_{int} to introduce a zero in the loop filter transfer function, clearly showing the zero introduced by the resistor R_{int} in series with the capacitor C_{int}. The need for this zero will become apparent when we analyze the PLL loop dynamics next.

The resulting PLL using a PFD-CP combination is shown in Figure 78.

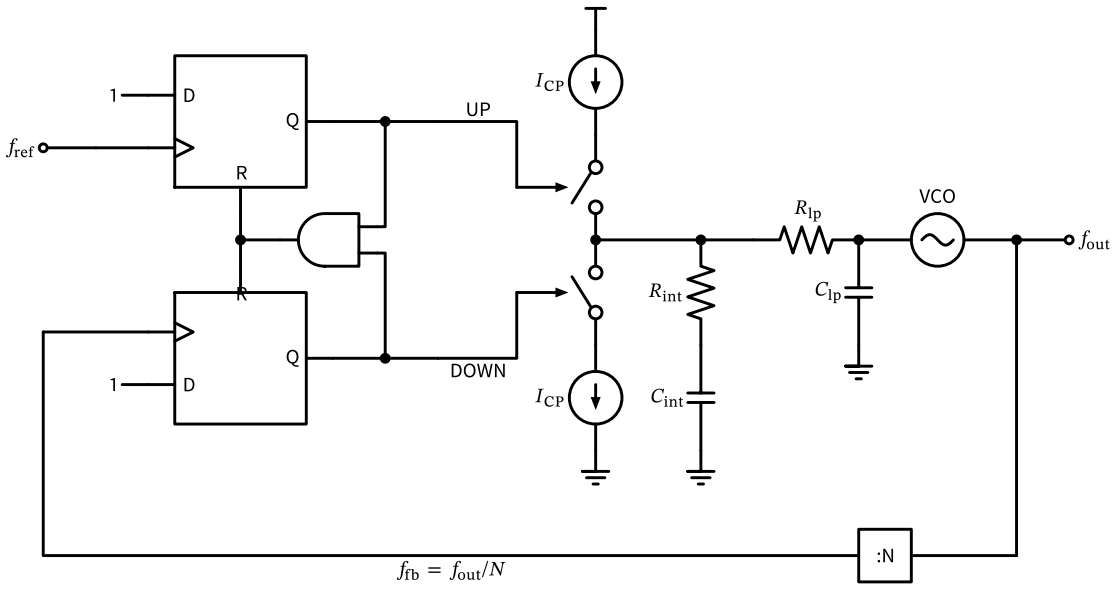

Figure 78: Diagram of a PFD-CP PLL. Note the added first-order lowpass which has been added to reduce the CP switching ripple.

Neglecting R_{lp} and C_{lp} for now, we can derive the closed-loop transfer function of the PLL from reference phase $\varphi_{\mathrm{ref}}(s)$ to output phase $\varphi_{\mathrm{out}}(s)$ as

$$H(s) = \frac{\varphi_{\mathrm{out}}(s)}{\varphi_{\mathrm{ref}}(s)} = N \cdot \frac{\frac{I_{\mathrm{CP}} K_{\mathrm{VCO}}}{2\pi C_{\mathrm{int}} N}(1 + sC_{\mathrm{int}} R_{\mathrm{int}})}{s^2 + s\frac{I_{\mathrm{CP}} K_{\mathrm{VCO}} R_{\mathrm{int}}}{2\pi N} + \frac{I_{\mathrm{CP}} K_{\mathrm{VCO}}}{2\pi C_{\mathrm{int}} N}} = N \cdot \frac{s \cdot 2\zeta\omega_{\mathrm{n}} + \omega_{\mathrm{n}}^2}{s^2 + s \cdot 2\zeta\omega_{\mathrm{n}} + \omega_{\mathrm{n}}^2} \tag{46}$$

with

$$\omega_{\mathrm{n}} = \sqrt{\frac{I_{\mathrm{CP}} K_{\mathrm{VCO}}}{2\pi C_{\mathrm{int}} N}} \tag{47}$$

and

$$\zeta = \frac{R_{\mathrm{int}}}{2}\sqrt{\frac{I_{\mathrm{CP}} K_{\mathrm{VCO}} C_{\mathrm{int}}}{2\pi N}}. \tag{48}$$

Looking at Equation 47 and Equation 48, we can see that we now have one *independent* parameter (R_{int}) to set the natural frequency ω_{n} and the damping factor ζ of the PLL independently. This allows us to design the loop dynamics more flexibly to achieve the desired performance.

With Equation 48, we can now appreciate the need for the resistor R_{int} in series with the capacitor C_{int} in the loop filter. Without this resistor, the damping factor ζ would be zero, leading to an undamped system with oscillatory behavior and poor transient response. The resistor introduces a zero in the loop filter transfer function, providing the necessary damping to stabilize the PLL and improve its dynamic response.

Looking at Figure 78 and Equation 45, we can also see that the PFD-CP introduces another integrator in the loop. This integrator, together with the integrator from the VCO, makes the PLL a **Type-II PLL**. A Type-II PLL can achieve zero steady-state phase error for step changes in the reference phase and zero steady-state frequency error for ramp changes in the reference frequency.

In conclusion:

- The PLL regulates the phase of a VCO to match the phase of a reference signal, using feedback control. Even with a steady-state phase error, the frequency error is zero (Type-I PLL). In a Type-II PLL, both phase and frequency errors are zero in steady state.
- The lowpass behavior of the loop filter determines the dynamic response and stability of the PLL.
- Further, the PLL transfer function from reference phase to output phase has the form of a lowpass, which passes the reference oscillator phase noise to the output inside the passband. The phase noise of the VCO is suppressed inside the loop bandwidth, as the PLL corrects for phase deviations.
- Outside the PLL loop bandwidth, the VCO phase noise dominates, as the PLL cannot correct for phase deviations fast enough.

This phase noise shaping behavior is illustrated in Figure 79, which shows how the reference and VCO phase noise contributions combine to form the total PLL output phase noise.

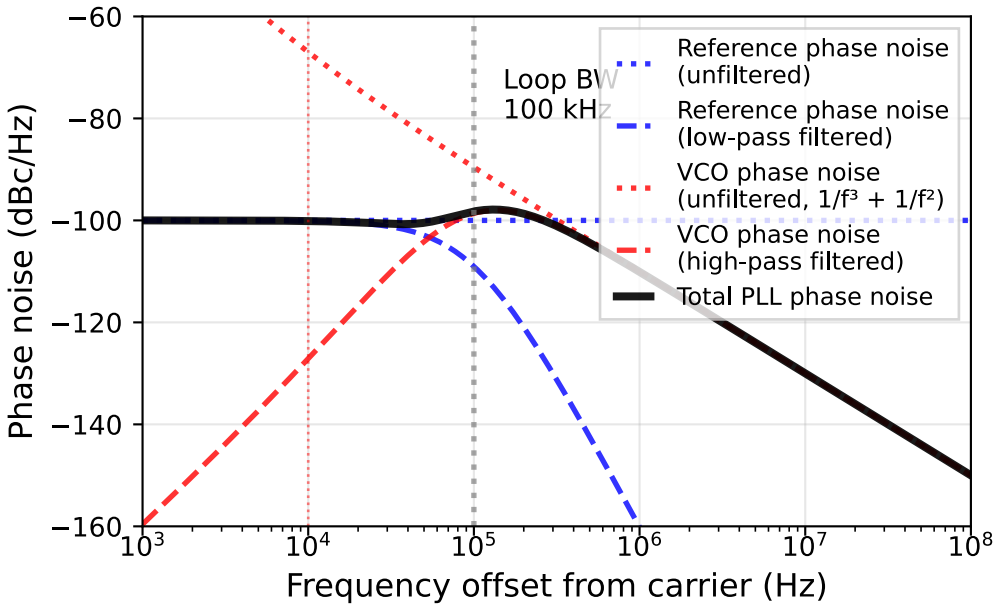

Figure 79: PLL phase noise contributions at the PLL output showing reference phase noise (increased by 20*log(N) and low-pass shaped), VCO phase noise (high-pass shaped), and total phase noise. The loop bandwidth determines the crossover frequency between reference and VCO phase noise contributions. For effective rejection of the 1/f³ VCO phase noise, a higher-order loop filter (shown is 3rd order) is beneficial.

Note that inside the PLL loop bandwidth the reference phase noise is multiplied by the division ratio N in power, i.e., $20 \log_{10}(N)$ in dB, as the PLL output frequency is N times the reference frequency. In Figure 79, $N = 100$ and a reference phase noise of −140 dBc/Hz is used. Large values of N can significantly increase the reference phase noise contribution at the PLL output, which is why low phase noise references are essential for high-performance PLLs.

7.3 All-Digital PLL

In the PLL implementations discussed so far, we have used analog components such as the VCO, loop filter, and charge pump. While these analog PLLs can achieve excellent performance, they also come with challenges such as component variability, temperature sensitivity, and integration complexity in modern digital processes. The question arises: Can we implement a PLL using only digital components to overcome these challenges? The answer is yes, leading to the concept of the **all-digital phase-locked loop (ADPLL)** [18], where a block diagram is shown in Figure 80.

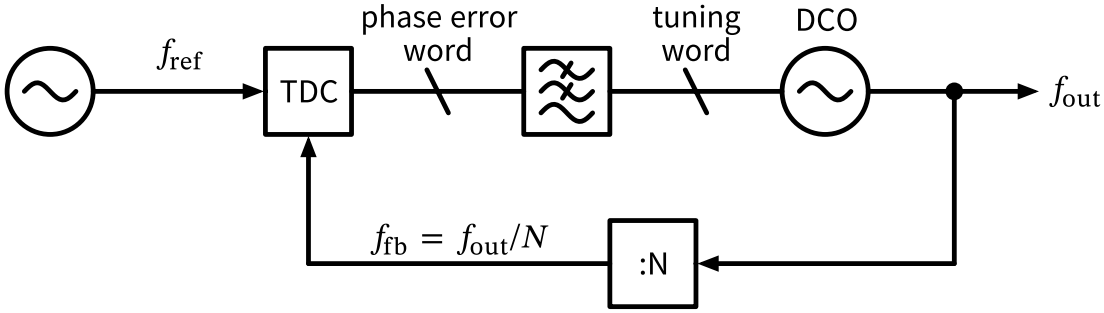

Figure 80: Block diagram of an all-digital PLL. A reference phase signal is compared to the phase of a VCO output signal in a time-to-digital converter. The phase error word is digitally low-pass filtered and used to tune the DCO frequency via a frequency control word.

Conceptually, the ADPLL in Figure 80 is similar to the analog PLL in Figure 69, with the key difference being that all components are implemented digitally. The phase detector is replaced by a time-to-digital converter (TDC), which measures the time difference between the reference and feedback signal edges and converts it into a digital phase error word. The loop filter is implemented as a digital filter (FIR, IIR, or similar), processing the phase error word to generate a tuning word for the digitally controlled oscillator (DCO). The DCO generates the output frequency based on the tuning word, completing the feedback loop.

The advantages of the ADPLL are that it can be fully integrated in a digital nm CMOS process, benefiting from scalability, low power consumption, small chip area, and immunity to analog component variations. Additionally, the digital nature of the ADPLL allows for easy programmability and adaptability to different applications. However, the ADPLL also faces challenges such as quantization noise from the TDC and DCO.

7.3.1 Time-to-Digital Converter

The TDC is a crucial component of the ADPLL, responsible for measuring the time difference between the reference and feedback signal edges and converting it into a digital phase error word. The TDC operates by sampling the time interval between two events (the rising edges of the reference and feedback signals) and quantizing this interval into discrete levels. TDCs can be implemented using various techniques. One popular method is using a digital delay line. An exemplary implementation is shown in Figure 81.

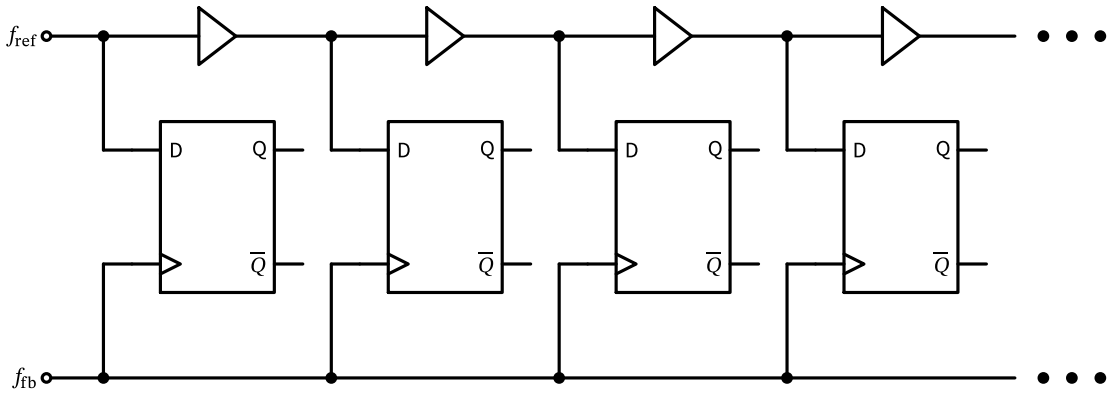

Figure 81: Basic TDC implementation as a delay line with parallel capture flip-flops. The TDC delay resolution is limited two inverter delays.

In Figure 81, the reference signal edge runs through a series of buffers (delay elements) connected in a delay line configuration. The feedback signal edge is used to sample the state of the delay line at the moment it arrives. The outputs of the DFFs represent the time difference between the reference and feedback signals, which can be encoded into a digital phase error word. The resolution of the TDC is determined by the delay between each stage in the delay line, with finer delays providing higher resolution. However, increasing the resolution also increases the complexity and power consumption of the TDC.

The phase noise of the TDC (at the output of the ADPLL) arising from this time measurement quantization can be modeled as

$$\mathcal{L}\{f\} = 10 \cdot \log\left[\frac{(2\pi)^2}{12 f_{\text{ref}}} \cdot \left(\frac{\Delta T_{\text{TDC}}}{T_{\text{DCO}}}\right)^2\right]$$

where ΔT_{TDC} is the time resolution of the TDC, T_{DCO} is the period of the DCO output frequency, and f_{ref} is the reference frequency. This equation shows that improving the TDC resolution (reducing ΔT_{TDC}) and increasing the reference frequency can help reduce the phase noise contribution from the TDC.

i Note 8: TDC Resolution Example

To get an idea of the performance requirements for the TDC, we assume a DCO output frequency of 2.4 GHz (e.g., for a Bluetooth application) and a reference frequency of 40 MHz. If we aim for a TDC phase noise contribution of −100 dBc/Hz at the DCO output, we can rearrange the above equation to find the required TDC time resolution:

$$\Delta T_{\text{TDC}} = T_{\text{DCO}} \cdot \sqrt{\frac{12 f_{\text{ref}} \cdot 10^{\mathcal{L}\{f\}/10}}{(2\pi)^2}} = \frac{1}{2.4 \times 10^9} \cdot \sqrt{\frac{12 \cdot 40 \times 10^6 \cdot 10^{-10}}{(2\pi)^2}} \approx 15\,\text{ps}$$

This number is challenging but achievable with modern TDC designs.

7.3.2 Digitally Controlled Oscillator

With the implementation of the TDC clarified in Section 7.3.1, we now turn our attention to the digitally controlled oscillator (DCO), which generates the output frequency of the ADPLL based on a digital tuning word. The DCO is a digital equivalent of the VCO used in analog PLLs, and its frequency is adjusted by changing the digital input value. Fundamentally, two possible implementations exist:

1. An analog-controlled oscillator (e.g., a VCO) is combined with a digital-to-analog converter (DAC) to convert the digital tuning word into an analog control voltage. This approach is shown in Figure 82.

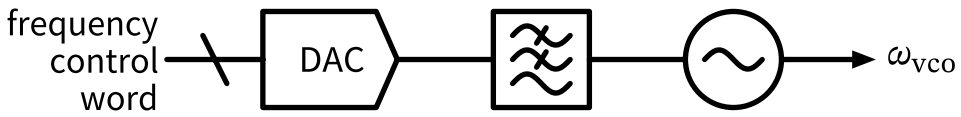

Figure 82: Digitally controlled oscillator using a DAC.

2. A digitally-controlled oscillator is using a large number of small varactors or switched capacitor banks to adjust the oscillation frequency directly based on the digital tuning word. An example implementation of a switched varactor is shown in Figure 83, or a switched capacitor like shown in Figure 67 is used.

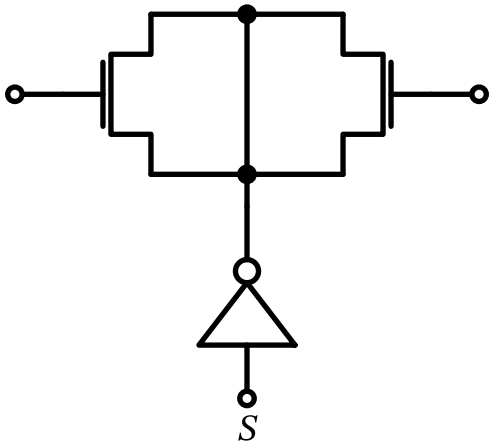

Figure 83: A switched differential varactor used for fine frequency control in a DCO.

It has to be noted that while an ADPLL seems conceptually simpler than an analog PLL, the design of high-performance TDCs and DCOs can be quite complex and requires careful consideration of quantization noise, linearity, and power consumption. Coupling effects between different circuit blocks can also introduce unwanted spurs and phase noise folding in the ADPLL output. Therefore, while ADPLLs offer many advantages, they also present unique design challenges that must be addressed to achieve optimal performance.

7.4 Fractional-N PLL

With the PLL architectures discussed so far, the output frequency resolution is limited to integer multiples of the reference frequency, i.e., $f_{\text{out}} = N \cdot f_{\text{ref}}$ with $N \in \mathbb{N}$. For a fine output frequency resolution, this requires a very low reference frequency. With the rule of thumb that the PLL loop bandwidth should be less than $f_{\text{ref}}/10$ to ensure stable operation, a low reference frequency also limits the achievable loop bandwidth, which is detrimental to phase noise performance and settling time. Further, we have learned in Section 7.2 that large values of N increase the reference phase noise contribution at the PLL output by $20\log_{10}(N)$ dB, making it challenging to achieve low phase noise performance with high frequency resolution.

How to break these fundamental limitations? The answer lies in the concept of the **fractional-N PLL**, which allows for non-integer frequency multiplication.

Assume we want to generate an output frequency of $f_{\text{out}} = 2.45\,\text{GHz}$ from a reference frequency of $f_{\text{ref}} = 40\,\text{MHz}$. This requires a frequency multiplication factor of $N = 61.25$, which is not an integer. To achieve this, the fractional-N PLL employs a feedback divider that can switch between two integer division ratios, N_1 and N_2, such that the average division ratio over time equals the desired fractional value (the averaging is performed by the lowpass loopfilter, providing the averaged tuning voltage to the VCO / the tuning word to the DCO). In this case, we can choose $N_1 = 61$ and $N_2 = 62$. By alternating between these two division ratios in a controlled manner, the PLL can effectively achieve an average division ratio of 61.25,

allowing it to generate the desired output frequency of 2.45 GHz. The following sequence illustrates this concept over four reference cycles:

$$N = N_1, N_1, N_1, N_2 \Longrightarrow \text{Average} \quad N = \frac{61 + 61 + 61 + 62}{4} = 61.25$$

A simple sequence like the one above works, but would lead to significant spurs in the PLL output spectrum due to the periodic nature of the division ratio switching, especially when the fractionality is close to an integer, and thus the sequence is long. Consider the case of $N = 61.01$, where we would need to switch to $N_2 = 62$ only once every 100 cycles of $N_1 = 61$:

$$N = \frac{99 \cdot 61 + 1 \cdot 62}{100} = 61.01$$

This periodic switching introduces spurious tones in the PLL output spectrum at multiples of $f_{\text{ref}}/100$, which can potentially pass through the loop filter and appear in the output spectrum, degrading the signal quality.

To minimize these spurs, more sophisticated techniques such as *delta-sigma modulation* are employed to *randomize* the switching sequence, spreading the quantization noise over a wider frequency range and reducing its impact on the PLL output. Additionally, the quantization noise introduced by the fractional division is *pushed to higher frequencies* where it can be more easily filtered out by the loop filter.

7.4.1 Delta-Sigma Modulator

For an in-depth understanding of delta-sigma modulation, we refer to [19]. Here, we briefly summarize the key concepts relevant to fractional-N PLLs. In a nutshell, a delta-sigma modulator (DSM) is a feedback system that shapes quantization noise to higher frequencies, allowing for high-resolution digital-to-analog conversion or frequency synthesis. The basic structure of a first-order DSM is shown in Figure 84.

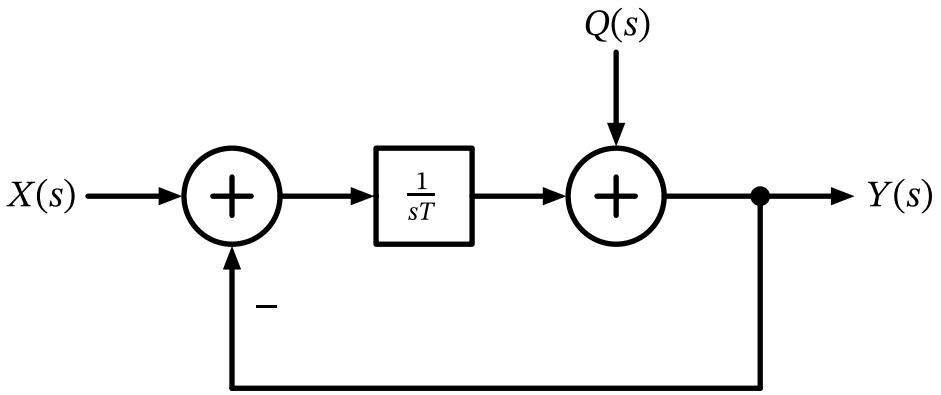

Figure 84: A first-order continuous time delta-sigma modulator.

In Figure 84, the input signal $X(s)$ is transferred to the output $Y(s)$ with the following transfer function (we set $Q = 0$):

$$Y(s) = X(s) \cdot \frac{1}{1 + sT} = X(s) \cdot H_{\text{LP}}(s)$$

The quantization noise $Q(s)$ introduced by the quantizer is shaped by the feedback loop, resulting in the following output contribution:

$$Y(s) = Q(s) \cdot \frac{sT}{1+sT} = Q(s) \cdot H_{\text{HP}}(s)$$

We can see that the quantization noise is highpass filtered, pushing it to higher frequencies, where it can be more easily filtered out by a subsequent lowpass filter.

The digital implementation of a first-order DSM is straightforward and one possible form is shown in Figure 85. At the point of truncation from $N+2$ bit to 1 bit (the MSB) we imagine the quantization noise $q[n]$ being added to the signal.

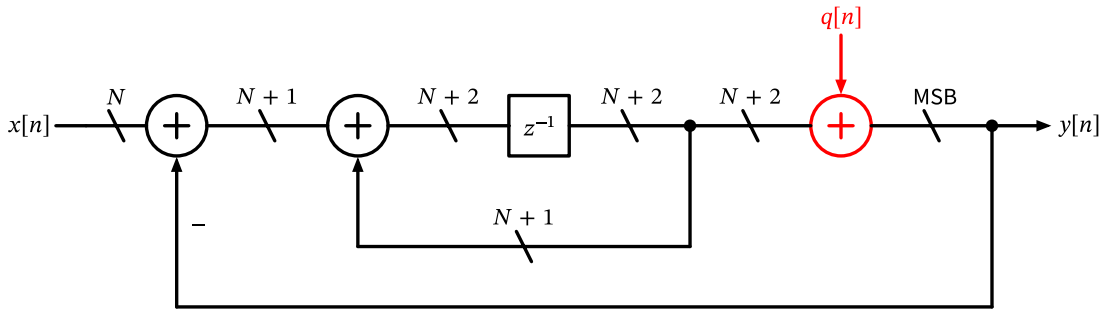

Figure 85: A first-order digital delta-sigma modulator.

Calculating the z-domain transfer functions from input $X(z)$ to the output $Y(z)$ (the signal transfer function, STF) and quantization noise $Q(z)$ to $Y(z)$ (the noise transfer function, NTF), we find

$$\frac{Y(z)}{X(z)} = \text{STF}\ (z) = z^{-1} \Longrightarrow \text{STF}\ (z) = z^{-1}$$

and

$$\frac{Y(z)}{Q(z)} = \text{NTF}\ (z) = (1 - z^{-1}) \Longrightarrow \text{NTF} = 1 - z^{-1}.$$

We see that the STF is a simple delay, while the NTF has a zero at DC, pushing the quantization noise to higher frequencies. Taking the Laplace transform of the NTF to find the frequency shaped PSD of the quantization noise, we find

$$S_{\text{y}}(f) = S_{\text{q}}(f)|H(f)|^2 = S_{\text{q}}(f) \cdot 2|1 - \cos(2\pi f T_{\text{s}})| \Longrightarrow H(f) = \sqrt{2|1 - \cos(2\pi f T_{\text{s}})|}.$$

with T_{s} being the sampling period of the DSM. The resulting $H(f)$ is shown in Figure 86. With the assumed PLL loop bandwidth of 100 kHz and a reference frequency of 40 MHz, we can see that the quantization noise is significantly attenuated inside the PLL loop bandwidth, minimizing its impact on the PLL output phase noise. The general trend is that higher-order DSMs and faster sampling frequencies lead to better noise shaping and lower in-band quantization noise.

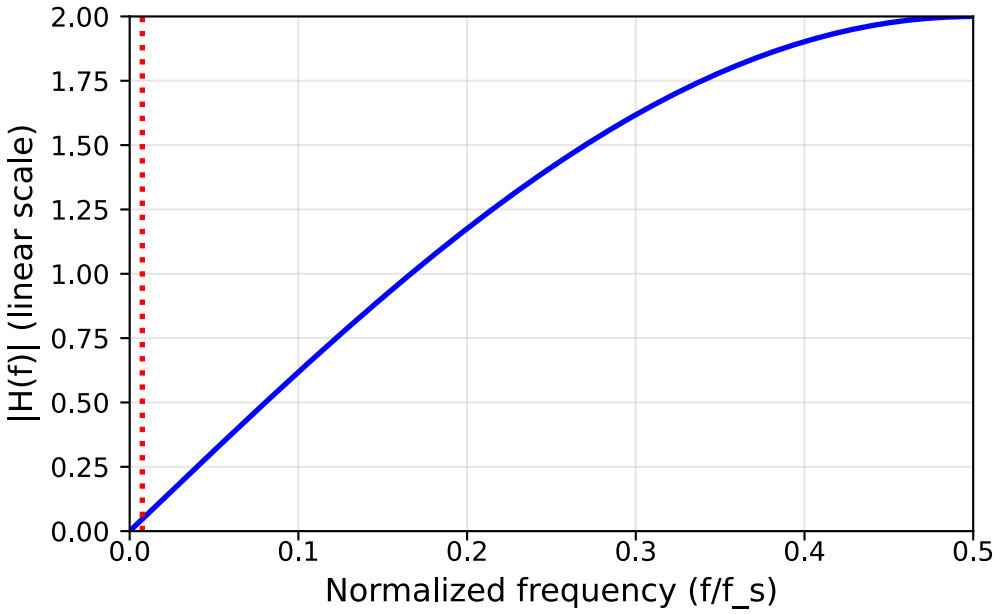

Figure 86: NTF H(f) of a first-order delta-sigma modulator showing the high-pass noise shaping characteristic. The quantization noise is suppressed at low frequencies and increases towards the Nyquist frequency. An exemplary loop filter bandwidth of 300 kHz for a 40 MHz reference frequency is indicated.

If we analyze the signal properties of the number sequence generated by the first-order DSM in Figure 85, we find that its randomization properties are not ideal, leading to spurs in the PLL output spectrum. To improve the randomization properties, higher-order DSMs are used in fractional-N PLLs, especially 2nd or 3rd order. A consequence of a higher-order DSM is that the produced sequence can take more values than just N and $N + 1$. The range of produced values is shown in Table 5.

Table 5: DSM output ranges and characteristics for different orders

DSM Order	Output Range	Noise Shaping
1	$N, N + 1$	20 dB/decade
2	$N - 1 \ldots N + 2$	40 dB/decade
3	$N - 3 \ldots N + 4$	60 dB/decade

7.4.2 Fractional-N PLL Implementation

A block diagram of a fractional-N PLL using a DSM and a multi-modulus divider (MMD) is shown in Figure 87. In order to ease the timing loop of the jittered clock coming from the MMD, the DSM is typically clocked by the MMD output.

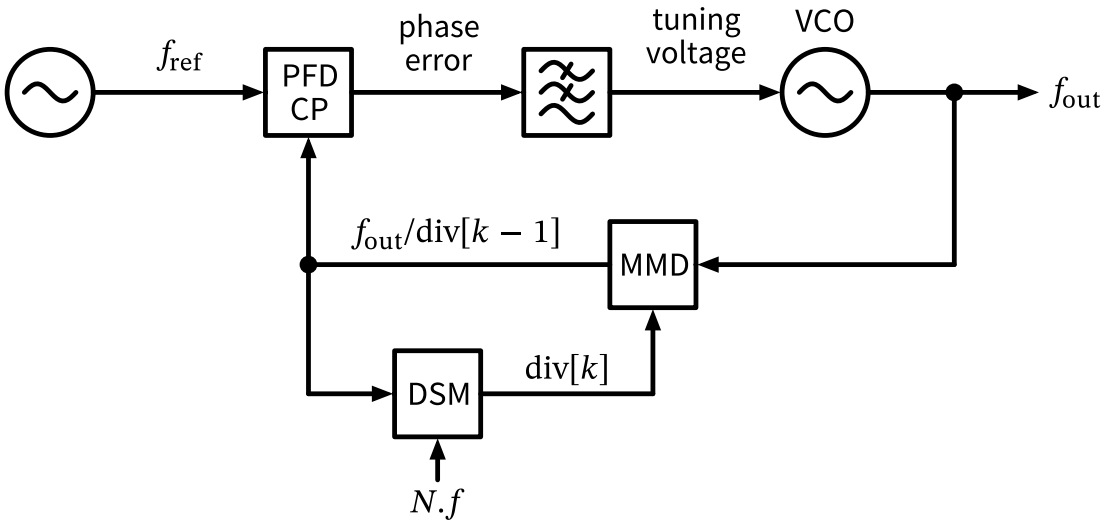

Figure 87: Block diagram of a fractional-N PLL. The VCO signal is divided by a sequence of integer divider values div[k] with (on average) is given by $N.f$, where N is the integer part of the divider value, and f the fractional part of the divider.

For a practical implementation of a fractional-N PLL it is of utmost importance that the path from the MMD output to the loop filter output is linear, as only the averaging/lowpass filtering of the noise-shaped quantization noise reduces the noise. Any *nonlinearity* in this path will lead to distortion of the quantization noise, causing quantization noise folding back into the PLL loop bandwidth, degrading the PLL output phase noise!

One prominent source of nonlinearity is the difference between the source and sink currents of the charge pump, as well as any nonlinearity present in the PFD/CP operating point of $\Delta\varphi \approx 0$ (like a deadzone). Since a Type-II PLL ideally operates at exactly this point, with random excursions around it due to the DSM-induced jitter, any nonlinearity here will seriously degrade the PLL performance. There are predominantly two options to mitigate this issue:

1. Use a Type-I PLL architecture with $\Delta\varphi \neq 0$.
2. Introduce a phase offset in a Type-II PLL to operate away from $\Delta\varphi = 0$.

A simple way to introduce a phase offset is to add a fixed current source/sink to the charge pump, as shown in Figure 88.

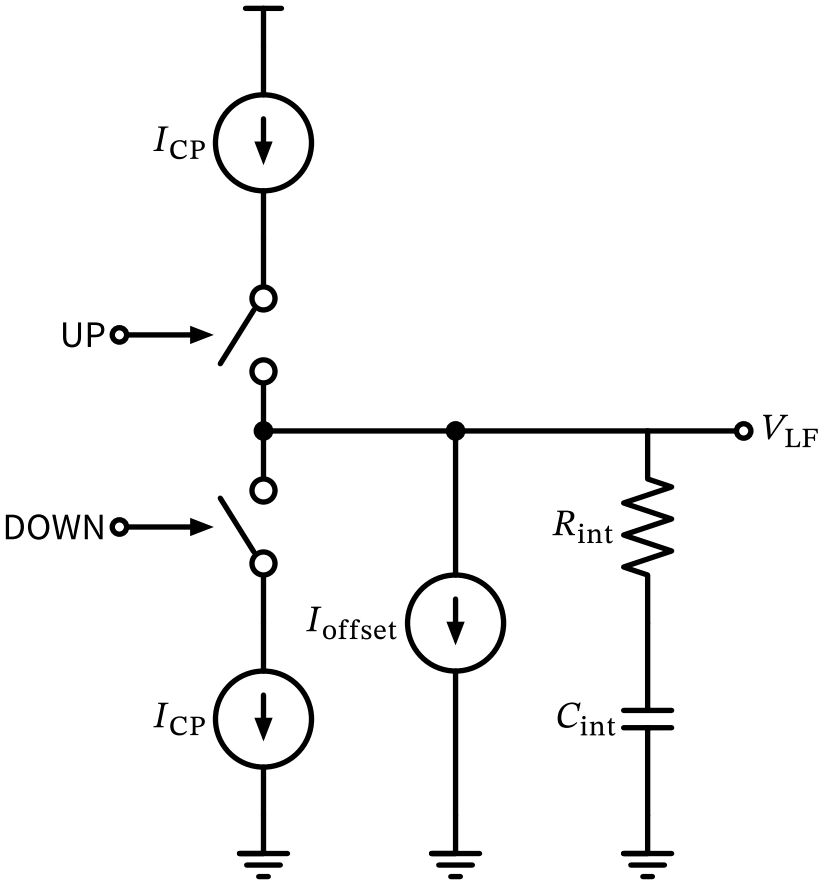

Figure 88: Charge pump with offset current for use in a fractional-N Type-II PLL.

As we have seen multiple times in the PLL discussion, using a higher f_{ref} is beneficial for phase noise performance, loop bandwidth, and TDC resolution (in an ADPLL). However, in many applications the choice of reference frequency is constrained by other factors. One way to achieve a higher reference frequency is by using a **frequency doubler**. As shown in Figure 89, a frequency doubler can be implemented using an XOR gate and a delay line. The delay line is set to a quarter of the period of the reference frequency, i.e., $T_{\text{d}} = T_{\text{ref}}/4$. The XOR gate then produces an output frequency of $f_{\text{double}} = 2 \cdot f_{\text{ref}}$. For stable operation, this delay line must be designed to be relatively insensitive to process, voltage, and temperature (PVT) variations, or must be adjusted with a control loop.

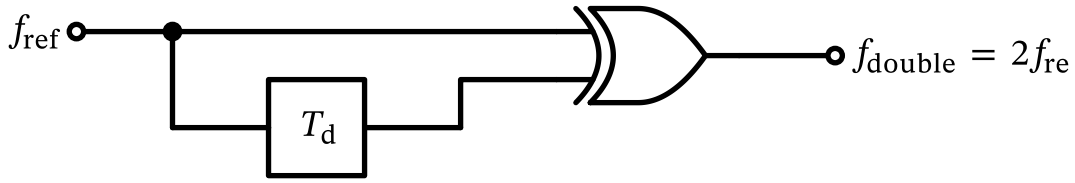

Figure 89: Implementation of reference frequency doubler.

As a final remark, we note that the output of the MMD in Figure 87 is a jittered clock due to the changing division ratio and the multiple gates and flip-flops making up the MMD. This jitter degrades the PLL phase noise performance. Fortunately, there is an easy fix to this issue

by using a retiming flip-flop after the MMD, clocked by the oscillator output, as shown in Figure 90.

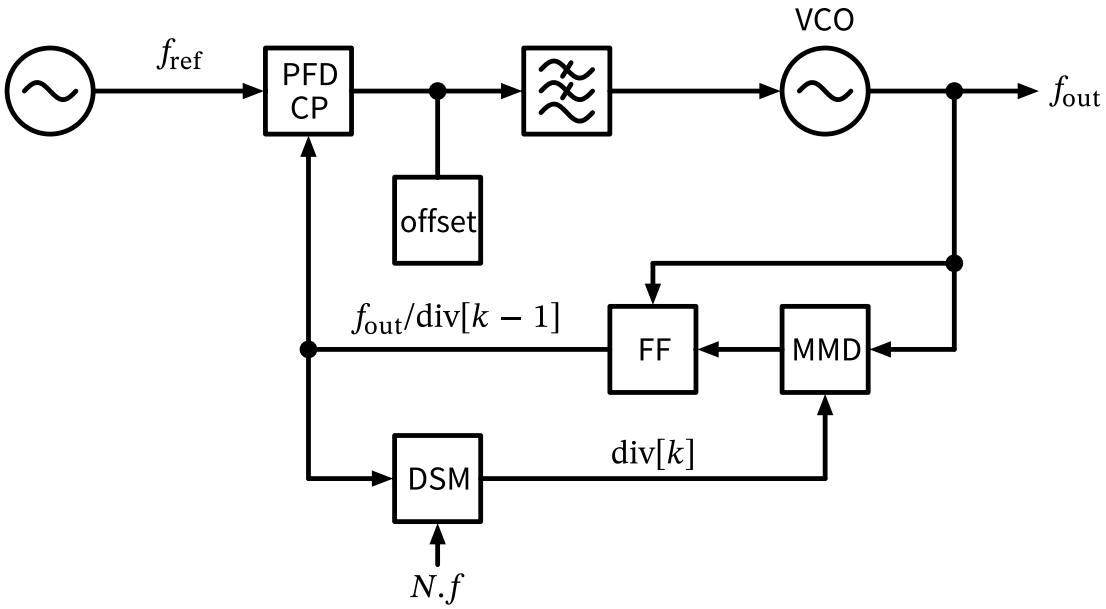

Figure 90: Block diagram of a fractional-N PLL including a retiming flip flop to reduce the MMD-related jitter.

This retiming flip-flop effectively synchronizes the MMD output to the oscillator clock domain, significantly reducing the jitter seen by the loop filter and improving the overall phase noise performance of the fractional-N PLL.

8 Power Amplifiers

To be added.

Bibliography

[1] B. Razavi, *RF Microelectronics*, 2nd edition. Pearson, 2011.

[2] H. Darabi, *Radio Frequency Integrated Circuits and Systems*, 2nd edition. Cambridge University Press, 2020.

[3] D. M. Pozar, *Microwave Engineering*. Wiley, 2011.

[4] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, Fifth. Wiley, 2009.

[5] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2017.

[6] R. Sarpeshkar, T. Delbruck, and C. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits and Devices Magazine*, vol. 9, no. 6, pp. 23–29, 1993, doi: 10.1109/101.261888.

[7] B. Sklar and F. J. Harris, *Digital Communications: Fundamentals and Applications*, 3rd edition. Pearson, 2020.

[8] A. F. Molisch, *Wireless Communications: From Fundamentals to Beyond 5G*, 3rd edition. Wiley-IEEE Press, 2022.

[9] T. S. Bird, "Definition and Misuse of Return Loss [Report of the Transactions Editor-in-Chief]," *IEEE Antennas and Propagation Magazine*, vol. 51, no. 2, pp. 166–167, 2009, doi: 10.1109/map.2009.5162049.

[10] W. R. Lepage, C. R. Cahn, and J. S. Brown, "Analysis of a comb filter using synchronously commutated capacitors," *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, vol. 72, no. 1, pp. 63–68, 1953, doi: 10.1109/tce.1953.6371974.

[11] R. Vazny, W. Schelmbauer, H. Pretl, S. Herzinger, and R. Weigel, "An Interstage Filter-Free Mobile Radio Receiver with Integrated TX Leakage Filtering," in 2010 IEEE Radio Frequency Integrated Circuits Symposium. IEEE, 2010, pp. 21–24. doi: 10.1109/rfic.2010.5477294.

[12] J. Kaukovuori, K. Stadius, J. Ryynänen, and K. A. I. Halonen, "Analysis and Design of Passive Polyphase Filters," *IEEE Transactions on Circuits and Systems–I: Regular Papers*, vol. 55, no. 10, pp. 3023–3037, 2008, doi: 10.1109/tcsi.2008.917990.

[13] B. Razavi, "A study of phase noise in CMOS oscillators," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 331–343, 1996, doi: 10.1109/4.494195.

[14] D. Leeson, "A simple model of feedback oscillator noise spectrum," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 329–330, 1966, doi: 10.1109/proc.1966.4682.

[15] P. Andreani and S. Mattisson, "On the use of MOS varactors in RF VCOs," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 905–910, 2000, doi: 10.1109/4.845194.

[16] A. Kral, F. Behbahani, and A. Abidi, "RF-CMOS oscillators with switched tuning," in Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (Cat. No.98CH36143). IEEE, 1998, pp. 555–558. doi: 10.1109/cicc.1998.695039.

[17] F. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849–1858, 1980, doi: 10.1109/TCOM.1980.1094619.

[18] R. Staszewski, D. Leipold, K. Muhammad, and P. Balsara, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS Process," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 815–828, 2003, doi: 10.1109/tcsii.2003.819128.

[19] R. Schreier, S. Pavan, and G. C. Temes, *Understanding Delta-Sigma Data Converters*. John Wiley & Sons, Ltd, 2017.