Analog Circuit Design

Harald Pretl Michael Koefinger Simon Dorrer

2025-09-08

Table of contents

1 Introduction

This is the material for an intermediate-level MOSFET analog circuit design course, held at JKU under course number 336.009 ("KV Analoge Schaltungstechnik").

The course makes heavy use of circuit simulation, using **Xschem** for schematic entry and **ngspice** for simulation. The 130nm CMOS technology **SG13G2** from IHP Microelectronics is used.

Tools and PDK are integrated in the **IIC-OSIC-TOOLS** Docker image, which will be used during the coursework.

> **!** 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 on GitHub to fix errors and omissions!
The production of this document would be impossible without these (and many more) great open-source software products: `VS Code`, `Quarto`, `Pandoc`, `TexLive`, `Jupyter Notebook`, `Python`, `Xschem`, `ngspice`, `CACE`, `pygmid`, `schemdraw`, `Numpy`, `Scipy`, `Matplotlib`, `Pandas`, `Git`, `Docker`, `Ubuntu`, `Linux`, …

! Important

Please use the IIC-OSIC-TOOLS image with tag `2025.03` or later, as the NMOS symbols changed in this version!

1.1 IHP's SG13G2 130nm CMOS Technology

SG13G2 is the name of a 130nm CMOS technology (strictly speaking BiCMOS) from IHP Microelectronics. It features low-voltage (thin-oxide) core MOSFET, high-voltage (thick-oxide) I/O MOSFET, various types of linear resistors, and 7 layers of Aluminum metallization (5 thin plus 2 thick metal layers). This PDK is open-source, and the complete process specification can be found at SG13G2 process specification. While we will not do layouts in this course, the layout rules can be found at SG13G2 layout rules.

For our circuit design, the most important parameters of the available devices are summarized in the following table:

Table 1: IHP SG13G2 devices

Component	Device Name	Specifications
Low-voltage (LV) NMOS	`sg13_lv_nmos`	operating voltage (nom.) $V_{DD} = 1.5\,V$, $L_{min} = 0.13\,\mu m$, $V_{th} \approx 0.5\,V$; isolated NMOS available
Low-voltage (LV) PMOS	`sg13_lv_pmos`	operating voltage (nom.) $V_{DD} = 1.5\,V$, $L_{min} = 0.13\,\mu m$, $V_{th} \approx -0.47\,V$
High-voltage (HV) NMOS	`sg13_hv_nmos`	operating voltage (nom.) $V_{DD} = 3.3\,V$, $L_{min} = 0.45\,\mu m$, $V_{th} \approx 0.7\,V$; isolated NMOS available
High-voltage (HV) PMOS	`sg13_hv_pmos`	operating voltage (nom.) $V_{DD} = 3.3\,V$, $L_{min} = 0.45\,\mu m$, $V_{th} \approx -0.65\,V$
Silicided poly resistor	`rsil`	$R_{\square} = 7\,\Omega \pm 10\%$, $TC_1 = 3100\,ppm/K$
Poly resistor	`rppd`	$R_{\square} = 260\,\Omega \pm 10\%$, $TC_1 = 170\,ppm/K$
Poly resistor high	`rhigh`	$R_{\square} = 1360\,\Omega \pm 15\%$, $TC_1 = -2300\,ppm/K$
MIM capacitor	`cap_cmim`	$C' = 1.5\,fF/\mu m^2 \pm 10\%$, $VC_1 = -26ppm/V$, $TC_1 = 3.6ppm/K$, breakdown voltage $> 15\,V$

Component	Device Name	Specifications
MOM capacitor	n/a	The metal stack is well-suited for MOM capacitors due to 5 thin metal layers, but no primitive capacitor device is available at this point.

1.2 Schematic Entry Using Xschem

Xschem is an open-source schematic entry tool with emphasis on integrated circuits. For up-to-date information of the many features of Xschem and the basic operation of it please look at the available online documentation. Usage of Xschem will be learned with the first few basic examples, essentially using a single MOSFET. The usage model of Xschem is that the schematic is hierarchically drawn, and the simulation and evaluation statements are contained in the schematics. Further, Xschem offers embedded graphing, which we will mostly use.

A summary of important Xschem keyboard shortcuts is provided in Section 19.

1.3 Circuit Simulation Using ngspice

ngspice is an open-source circuit simulator with SPICE dependency (Nagel 1975). Besides the usual simulated types like `op` (operating point), `dc` (dc sweeps), `tran` (time domain), `ac` (small-signal frequency sweeps), and `noise` (small-signal noise analysis), ngspice offers a script-like control interface, where many different simulation controls and result evaluations can be done. For detailed information please refer to the latest online manual.

Important ngspice simulation commands and options (e.g., how to control convergence settings) are listed in Section 20.

1.4 Integrated IC Design Environment (IIC-OSIC-TOOLS)

In order to make use of the various required components (tools like Xschem and ngspice, PDKs like SG13G2) easier, we will use the **IIC-OSIC-TOOLS**. This is a pre-compiled Docker image which allows to do circuit design on a virtual machine on virtually any type of computing equipment (personal PC, Raspberry Pi, cloud server) on various operating systems (Windows, macOS, Linux). For further information like installed tools, how to setup a VM, etc., please look at IIC-OSIC-TOOLS GitHub page.

> ⚠ Preparation
>
> Please make sure to receive information about your personal VM access ahead of the course start.

Experienced users can install this image on their personal computer, for JKU students the IIC will host a VM on our compute cluster and provide personal login credentials.

> ⚠ Linux
>
> In this course, we assume that students have a basic knowledge of Linux and how to operate it using the terminal (shell). If you are not yet familiar with Linux (which is basically a must when doing integrated circuit design as many tools are only available on Linux), then please check out a Linux introductory course or tutorial online, there are many resources available.
> A summary of important Linux shell commands is provided in Section 18.

1.5 Setting up the Design Directory

- Open your VM by entering the URL in your browser.
- Open a terminal (third icon in the taskbar at the bottom). You should get the following prompt: `/foss/designs >`
- Clone the git repository into the current directory: `git clone https://github.com/iic-jku/analog-`
- This GitHub repository includes a file called `.designinit`, which sets the PDK and certain paths. However, this must be located in `/foss/designs/`
- Therefore, we first need to copy it there: `cp analog-circuit-design/.designinit .`
- Then we adjust the variable `XSCHEM_USER_LIBRARY_PATH` by opening the file in an editor e.g. `nano .designinit`
  - Change the last line from `export XSCHEM_USER_LIBRARY_PATH=$DESIGNS/xschem` to `export XSCHEM_USER_LIBRARY_PATH=$DESIGNS/analog-circuit-design/xschem`
- To apply the changes, we need to close the current terminal window: `exit`
- Open again a terminal
- Test if the correct PDK gets selected: `echo $PDK` (you should get `sg13g2` as the answer)
- Change into the GitHub repository: `cd analog-circuit-design`
- Start xschem using `xschem` or directly open a specific schematic using `xschem xschem/dc_lv_nmos.sch`

1.5.1 Creating Backups

You can easily create backups of your work by creating a zip archive of the complete directory:

- Change to the parent directory: `cd /foss/designs`
- Create a zip archive from the complete design folder: `zip backup.zip analog-circuit-design -r`

1.5.2 Updating the Repository

- **Create a backup!**
- Go to directory: e.g. `cd /foss/designs/analog-circuit-design`
- Fetch newest changes from the origin: `git fetch origin`
- Merge changes from the origin into local branch 'master': `git merge origin/main`

⚠ Git Merge Conflicts

It is possible that `git merge` does not complete successfully. Either you are able to resolve the merge conflict manually, or it may be easier to make a fresh clone of the repository and adding your local changes manually from the backup.

! Important

Please think twice before executing any git command without a backup, as this could lead to permanent loss of data!

2 First Steps

In this first chapter we will learn to use Xschem for schematic entry, and how to operate the ngspice SPICE simulator for circuit simulations. Further, we will make ourself familiar with the transistor and other passive components available in the IHP Microelectronics SG13G2 technology. While this is strictly speaking a BiCMOS technology offering MOSFETs as well as SiGe heterojunction bipolar transistors (HBTs), we will use it as a pure CMOS technology, which is available from IHP under the name SG13C.

2.1 The Metal-Oxide-Semiconductor Field-Effect-Transistor (MOSFET)

In this course, we will **not** dive into semiconductor physics and derive the device operation bottom-up starting from a fundamental level governed by quantum mechanics or a simplified solid-state physics based approach resulting in the well-known square-law model. Instead, we will treat the MOSFET behaviorally by assuming a 4-terminal device, and the performance of this device regarding its terminal voltages and currents we will largely derive from the simulation model.

Since we have an emphasis on integrated circuit design in this course the size of the MOSFET can be adapted by changing its width W and its length L. As we will see later, L has a profound impact on the MOSFET performance allowing to trade-off speed versus output conductance versus device-to-device matching. The width W is more of a scaling parameter to adapt the current **density** (strictly speaking charge density) forming in the MOSFET channel to a desired current. More about this later.

The circuit symbol that we will use for the n-channel MOSFET is shown in Figure 1, and for the p-channel MOSFET it is shown in Figure 2. A control voltage between gate ("G") and source ("S") controls the current flow between drain ("D") and source. The MOSFET is a 4-terminal device, so the bulk ("B") can also control the drain-source current flow. Often, the bulk is connected to source, and then the bulk terminal is not shown to declutter the schematics.

MOSFET Background

Strictly speaking is the drain-source current of a MOSFET controlled by the voltage between gate and bulk (V_{GB}) and the voltage between drain and source (V_{DS}). Since bulk is often connected to source anyway, and many circuit designers historically were already familiar with the operation of the bipolar junction transistor (BJT), it is common to consider the gate-source voltage (besides the drain-source voltage) as the controlling voltage.
This focus on gate-source suggests that the source is special compared to the drain. In a typical physical MOSFET, however, the drain and source are constructed exactly the same (i.e., the MOSFET is a symmetric device), and which terminal is drain, and which terminal is source, is only determined by the applied voltage potentials, and can change dynamically during operation (think of a MOSFET operating as a switch… which side is the drain, which side is the source?).
Unfortunately, this focus on a "special" source has made its way into some MOSFET compact models. The model that is used in SG13G2 luckily uses the PSP model, which is formulated symmetrically with regards to drain and source, and is thus very well suited for analog and RF circuit design. For a detailed understanding of the PSP model please refer to the model documentation.

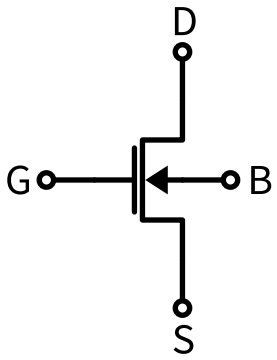


Figure 1: Circuit symbol of n-channel MOSFET.

For hand calculations and theoretical discussions we will use the following simplified large-signal model, shown in Figure 3. A current source I_D models the current flow between drain and source, and it is controlled by the three control voltages V_{GS}, V_{DS}, and V_{SB}. Note that in this way (since $I_D = f(V_{DS})$) also a resistive behavior between D and S can be modelled. In case that B and S are shorted then simply $V_{SB} = 0$ and C_{SB} is shorted.

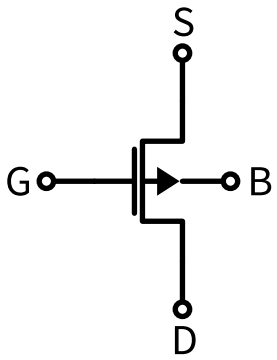


Figure 2: Circuit symbol of p-channel MOSFET.

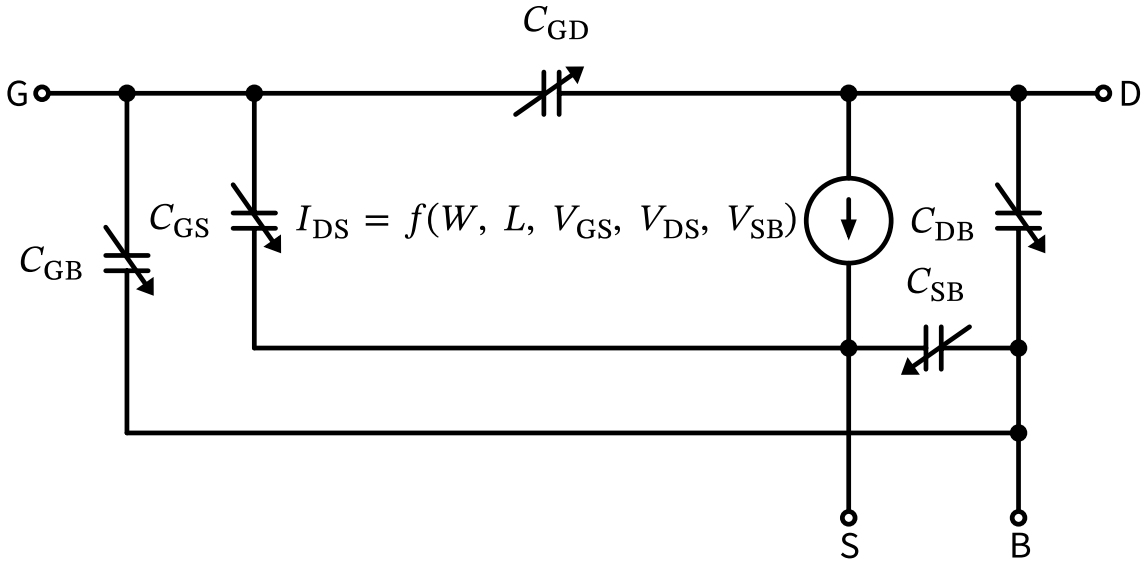


Figure 3: The MOSFET large-signal model. In general, all capacitors are nonlinear, i.e., they depend on their terminal voltages.

In an ideal MOSFET no dc current is flowing into the gate, the behavior is purely capacitive. We model this by two capacitors: $C_{GG} = C_{GS} + C_{GD} + C_{GB}$ is the total capacitance when looking into the gate of the MOSFET. C_{GS} is usually the dominant capacitance, and C_{GD} models the capacitive feedback between D and G, usually induced by a topological overlap capacitance in the physical construction of the MOSFET. This capacitance is often small compared to C_{GS}, but in situations where we have a large voltage swing at the drain this capacitance will be affected by the Miller effect (see Section 16.1). In hand calculations we will often set $C_{GD} = C_{GB} = C_{DB} = C_{SB} = 0$.

To model a physical MOSFET there will be also a requirement for resistors in the model to account for terminal access resistances (R_G, R_D, and R_S) as well as resistors to model second-order effects like non-quasistatic operation. For lower frequencies and bulk MOSFETs we will not consider these resistors, and just deal with the capacitive behavior.

MOSFET Bulk Terminal

In many situations we will connect the bulk and source terminals of a MOSFET together, which results in a simplified large-signal model. As an exercise, look at Figure 3 and draw this simplified model (hint: look at Figure 5 and Figure 6 for inspiration).

Now, as we are skipping the bottom-up approach of deriving the MOSFET large-signal behavior from basic principles, we need to understand the behavior of the elements of the large-signal model in Figure 3 by using a circuit simulator and observing what happens. And generally, a first step in any new IC technology should be to investigate basic MOSFET performance, by doing simple dc sweeps of V_{GS} and V_{DS} and looking at I_D and other large- and small-signal parameters.

As a side note, the students who want to understand MOSFET behavior from a physical angle should consult the MOSFET chapter from the JKU course "Design of Complex Integrated Circuits" (VL 336.048). A great introduction into MOSFET operation and fabrication is given in (Hu 2010), which is available freely online and is a recommended read. A very detailed description of the MOSFET (leaving usually no question unanswered) is provided in (Tsividis and McAndrew 2011).

Now, in order to get started, basic Xschem testbenches are prepared, and first simple dc sweeps of various voltages and currents will be done. But before that, please look at the import note below!

Mathematical Notation

Throughout this material, we will largely stick to the following notation standardized by IEEE:

- A **dc quantity** is shown with an upper-case variable name with upper-case subscripts, like V_{GS}.
- Double-subscripts denote **dc sources**, like V_{DD} and V_{SS}.
- An **ac (small-signal) quantity** (incremental quantity) has a lower-case variable

name with a lower-case subscript, like g_{m}.
- A **total quantity** (dc plus ac) is shown as a lowercase variable name with upper-case subscript, like i_{DS}.
- An upper-case variable name with a lower-case subscript is used to denote **RMS quantities**, like I_{ds}.

i A Comment on Active and Passive Devices and Linear vs. Nonlinear

In contrast to the **passive** devices resistor R, inductor L, and capacitor C, which can only dissipate energy (and are often treated in a linearized fashion), transistors (like the MOSFET) are called "**active**", since they can provide signal power amplification. However, transistors can not create energy out of thin air, but merely convert **dc** energy (supplied by the power supply) into **ac** energy. They have to have nonlinear transfer characteristics to do this, but it has been shown that a piecewise-linear characteristic is sufficient (Jakoby 2022). This is very good news for circuit design, as usually we strive for linear behaviour!

2.1.1 Large-Signal MOSFET Model

We start with an investigation into the large-signal MOSFET model shown in Figure 3 by using the simple testbench for the LV NMOS shown in Figure 4.

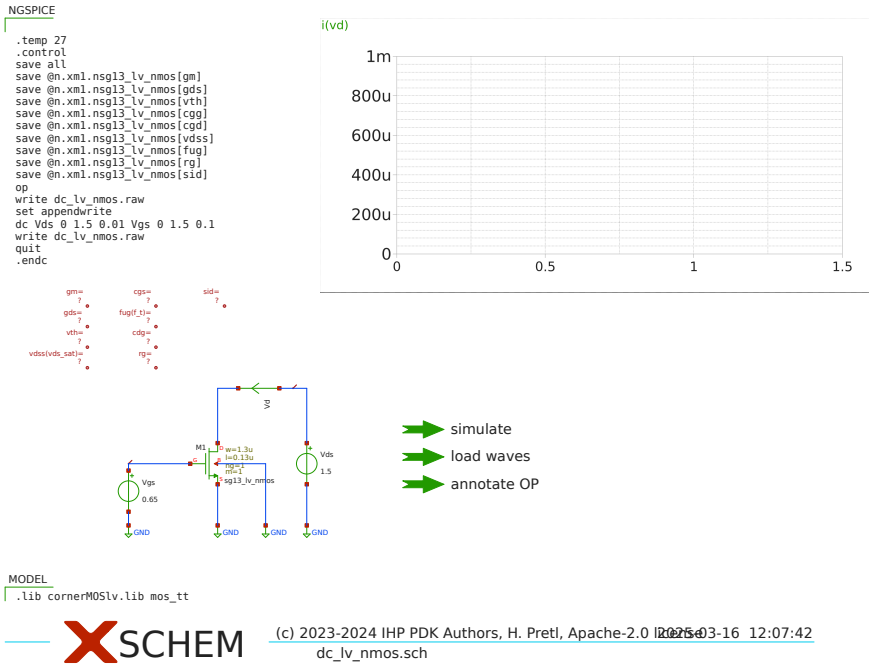


Figure 4: Testbench for NMOS dc sweeps.

MOSFET Simulation Model

For modelling the MOSFET behavior in a circuit simulator like ngspice different models are available. Some of these models have been widely adopted, like the BSIM (Berkeley Short-channel IGFET Model) or PSP (Philips Penn State) model. The PSP model version 103.6 is used in the IHP SG13G2 PDK for the LV and HV MOSFET. This model has several advantages:

- Physics-based surface-potential model
- Symmetric formulation with respect to drain and source
- Support for mobility reduction, velocity saturation, DIBL, gate current, lateral doping gradient effects, STI stress, NQS, etc.

The PSP 103.6 model documentation can be found here. In chapter 8 the dc operating point output of the model (these parameters can be queried in ngspice) is explained, which is helpful to interpret the simulation output.

Exercise: MOSFET Investigation

Please try to execute the following steps and answer these questions:

1. Get the LV NMOS testbench (available at https://github.com/iic-jku/analog-circuit-design/blob/main/xschem/dc_lv_nmos.sch) working in your IIC-OSIC-TOOLS environment.
2. Make yourself familiar with Xschem (change the schematic in various ways, run a simulation, graph the result).
3. Make yourself familiar with ngspice (run various simulations, save nets and parameters, use the embedded Xschem graphing, explore the interactive ngspice shell to look at MOSFET model parameters).
4. Explore the LV NMOS `sg13_lv_nmos`:
    1. How is I_D affected by V_{GS} and V_{DS}?
    2. Change W and L of the MOSFET. What is the impact on the above parameters? Can you explain the variations?
    3. Look at the capacitance values for C_{GS}, C_{GB}, C_{GD}, and C_{DB}. How are they affected by W and L and by changing the bias conditions (play with V_{GS} and V_{DS})?
    4. When looking at the model parameters in ngspice, you see that there is a C_{GD} and a C_{DG}. Why is this, what could be the difference? Sometimes these capacitors show a negative value, why? (Hint: Study Note 1)
5. Build testbenches in Xschem for the LV PMOS, the HV NMOS, and the HV PMOS. Explore the different results.
    1. For a given W and L, which device provides more drain current? How are the capacitances related?

2. If you would have to size an inverter, what would be the ideal ratio of W_p/W_n? Will you exactly design this ratio, or are the reasons to deviate?
3. There are LV and HV MOSFETs, and you investigated the difference in performance. What is the rationale when designing circuits for selection either an LV type, and when to choose an HV type?

6. Build a test bench to explore the body effect, start with LV NMOS.
   1. What happens when $V_{\mathrm{SB}} \neq 0$?

2.1.2 Small-Signal MOSFET Model

As you have seen in the previous investigations, the large-signal model of Figure 3 describes the behavior of the MOSFET across a wide range of voltages applied at the MOSFET terminals. Unfortunately, for hand analysis dealing with a nonlinear model is close to impossible, at the very least it is quite tedious.

However, for many practical situations, we bias a MOSFET with a set of dc voltages applied to its terminal, and only apply small signal excursions during operation. If we do this, we can linearize the large-signal model in this dc operating point, and resort to a small-signal model which can be very useful for hand calculations. Many experienced designers analyze their circuits by doing these kind of hand calculations and describing the circuit analytically, which is a great way to understand fundamental performance limits and relationships between parameters.

We will use the small-signal MOSFET model shown in Figure 5 for this course. The current-source $i_{\mathrm{d}} = g_{\mathrm{m}} v_{\mathrm{gs}}$ models the drain current I_{D} as a function of V_{GS} with

$$g_{\mathrm{m}} = \frac{\partial I_{\mathrm{D}}(V_{\mathrm{GS}}, V_{\mathrm{DS}}, V_{\mathrm{SB}})}{\partial V_{\mathrm{GS}}},$$

and the resistor g_{ds} models the dependency of the drain current by V_{DS}:

$$g_{\mathrm{ds}} = \frac{\partial I_{\mathrm{D}}(V_{\mathrm{GS}}, V_{\mathrm{DS}}, V_{\mathrm{SB}})}{\partial V_{\mathrm{DS}}}$$

The drain current dependency on the source-bulk voltage (the so-called “body effect”) is introduced by the current source $i_{\mathrm{d}} = g_{\mathrm{mb}} v_{\mathrm{sb}}$:

$$g_{\mathrm{mb}} = \frac{\partial I_{\mathrm{D}}(V_{\mathrm{GS}}, V_{\mathrm{DS}}, V_{\mathrm{SB}})}{\partial V_{\mathrm{SB}}}$$

As has been mentioned before, in many situations (and whenever we want to use a simplified model) we connect source and bulk of the MOSFET together. This results in the much simplified small-signal model shown in Figure 6.

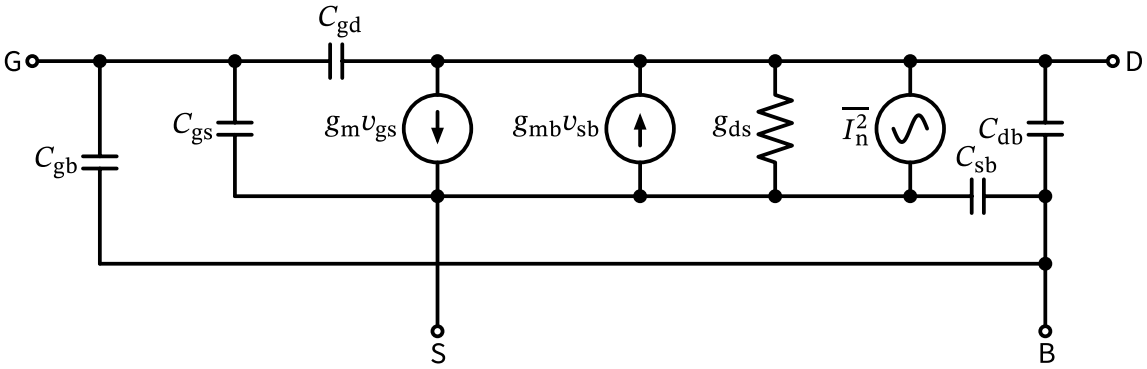


Figure 5: The MOSFET small-signal model.

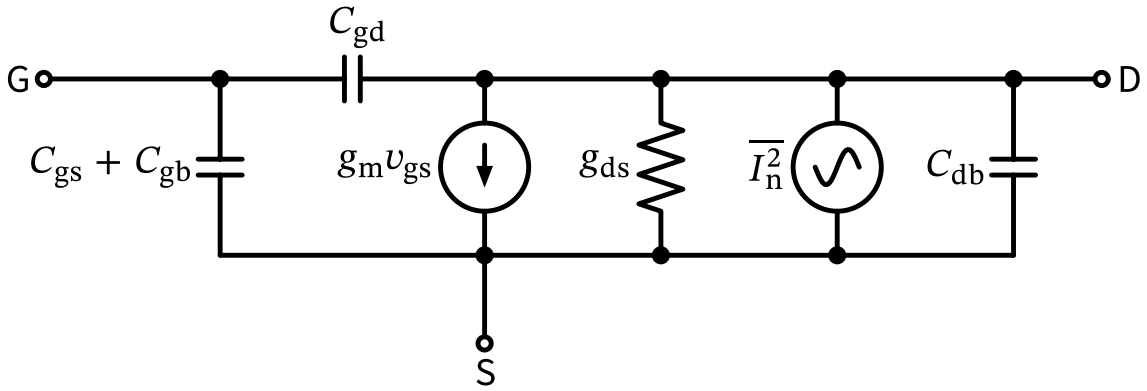


Figure 6: The MOSFET small-signal model when source and bulk are shorted.

As any electronic device the MOSFET introduces noise into the circuit. In this course we will only consider the **drain-source current noise** of the MOSFET, given by

$$\overline{I_{\mathrm{n}}^2} = 4kT\gamma g_{\mathrm{d0}}, \tag{1}$$

where $\overline{I_{\mathrm{n}}^2}$ is the one-sided power-spectral density of the noise in A^2/Hz; k is the Boltzmann constant; T is the absolute temperature; γ is a (fitting) parameter in simplified theory changing between $\gamma = 2/3$ in saturation and $\gamma = 1$ for triode operation; g_{d0} is equal to g_{m} in saturation and g_{ds} in triode).

> **i** MOSFET Triode and Saturation Region
>
> Sometimes we will refer to different operating modes of the MOSFET like "saturation" or "triode." Generally speaking, when the drain-source voltage is small, then the MOSFET acts as a voltage-controlled resistor (since the impact of both V_{GS} and V_{DS} on I_{D} is large), and this mode of operation we call "**triode**" mode.
> When the drain-source voltage V_{DS} is increased, at some point the drain-source current saturates and is only a weak function of the drain-source voltage, while still being well controlled by V_{GS}. This mode is called "**saturation**" mode.
> As you can see in the large-signal investigations, these transitions happen gradually, and it is difficult to define a precise point where one operating mode switches to the other one. In this sense we use terms like "triode" and "saturation" only in an approximate sense.

We can also consider an even more reduced small-signal MOSFET model compared to Figure 6, which is shown in Figure 7. In this, we just consider the transconductance g_{m}, the input capacitor C_{gg}, as well as the output conductance g_{ds}. Note that we can redraw the pi-model of Figure 7 into the τ-model of Figure 8. Depending on the circuit configuration, either the first or the second form results in simpler calculations of the circuit equations.

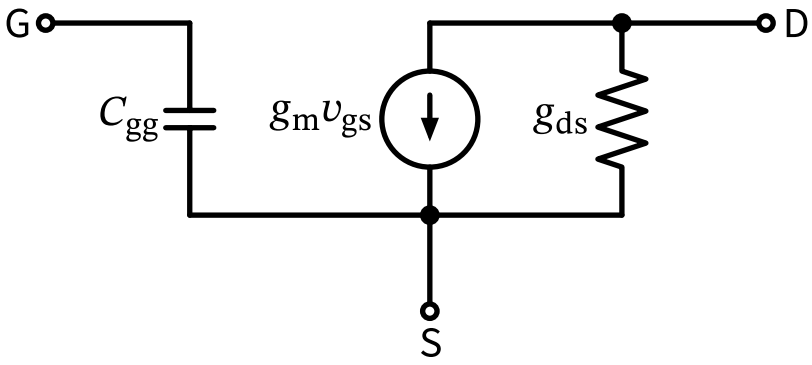


Figure 7: The MOSFET small-signal basic pi-model.

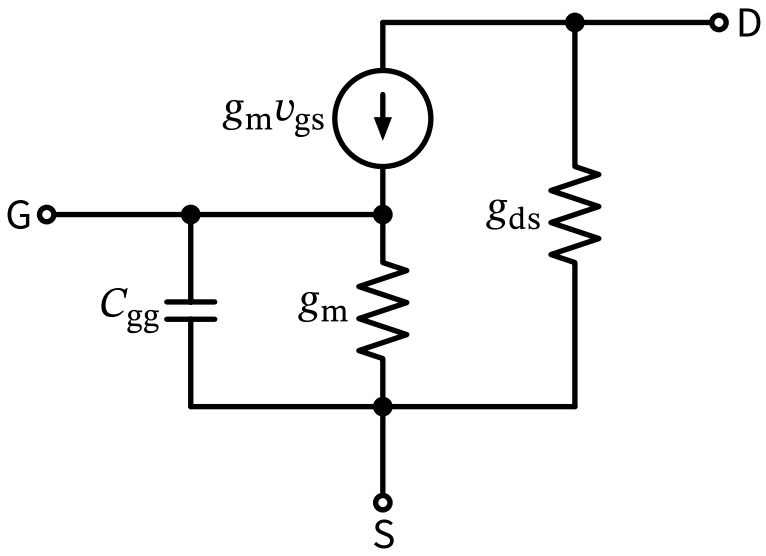


Figure 8: The MOSFET small-signal basic T-model.

> **Exercise: MOSFET Model Transformation**
>
> Can you show, with which circuit manipulations you can transform the pi-model of Figure 7 into the T-model of Figure 8?

A metric which is useful to assess the speed of a MOSFET is the so-called **transit frequency** f_T. It is defined as the frequency where the small-signal current gain (output current divided by the input current) of a MOSFET driven by a voltage-source at the input and loaded by a voltage source at the output drops to unity (reaches one). It can easily be derived using the simplified MOSFET small-signal model of Figure 6 by driving it with a voltage source and shorting the output to (neglecting the feed-forward current introduced by C_{gd})

$$\omega_T = 2\pi f_T \approx \frac{g_m}{C_{gg}} = \frac{g_m}{C_{gs} + C_{gd} + C_{gb}}. \tag{2}$$

This frequency is an extrapolated frequency where the MOSFET operation is dominated by several second-order effects (hence Equation 2 is not valid any longer). A rule-of-thumb is to use a MOSFET up to approximately $f_T/10$. In any case, f_T is a proxy of the speed of a MOSFET; in other words, how much input capacitance C_{gg} is incurred when creating a certain g_m.

> **Exercise: MOSFET Transit Frequency**
>
> As a home exercise, try to derive Equation 2 starting from Figure 6. By showing this transformation you can proof that indeed both circuits are electrically equivalent.

Now we need to see how the small-signal parameters seen in Figure 5 can be investigated and estimated using circuit simulation.

Exercise: MOSFET Small-Signal Parameters

Please try to execute the following steps and answer the following questions:

1. Reuse the LV NMOS testbench (available at https://github.com/iic-jku/analog-circuit-design/blob/main/xschem/dc_lv_nmos.sch).
2. Explore the LV NMOS `sg13_lv_nmos`:
    1. How are g_{m} and g_{ds} changing when you change the dc node voltages?
    2. What is the ratio of g_{m} to g_{mb}? What is the physical reason behind this ratio (you might want to revisit MOSFET device physics at this point)?
    3. Take a look at the device capacitances C_{gs}, C_{gd}, and C_{gb}. Why are they important? What is the f_{T} of the MOSFET?
    4. Look at the drain noise current according to the MOSFET model and compare with a hand calculation of the noise. In the noise equation there is the factor γ, which in triode is $\gamma = 1$ and in saturation is $\gamma = 2/3$ according to basic text books. Which value of γ are you calculating? Why might it be different?
3. Go back to your testbench for the LVS PMOS `sg13_lv_pmos`:
    1. What is the difference in g_{m}, g_{ds}, and other parameters between the NMOS and the PMOS? Why could they be different?

Note 1: Maxwell Capacitance Matrix

A Maxwell capacitance matrix (Maxwell 1873) provides the relation between voltages on a set of conductors and the charges on these conductors. For a given conductor set with N conductors (and thus N terminals) the relation is

$$\mathbf{Q} = \mathbf{C} \cdot \mathbf{V}$$

where $\mathbf{Q}$ is a vector of the charges on the N conductors, $\mathbf{C}$ is a $N \times N$ capacitance matrix, and $\mathbf{V}$ is the potential vector. In the case of two conductors and physical capacitances between them, $\mathbf{C}$ is given by

$$\mathbf{C} = \begin{pmatrix} C_{11} + C_{12} & -C_{12} \\ -C_{21} & C_{21} + C_{22} \end{pmatrix}$$

where $C_{xx} = \partial Q_x / \partial V_x$ is the auto capacitance from a conductor x towards infinity (ground), and $C_{xy} = \partial Q_x / \partial V_y$ is the mutual capacitance from node/conductor x to node/conductor y. For a physical capacitor $C_{xy} = C_{yx}$.
Using the above equation to calculate Q_1 (the charge on conductor 1) results in

$$Q_1 = (C_{11} + C_{12})V_1 - C_{12}V_2 = C_{11}(V_1 - 0) + C_{12}(V_1 - V_2)$$

which is the expected result.

Such a Maxwell capacitance formulation is also used in the MOSFET model to describe the charge at a terminal as a function of potential at another terminal. So,

$$C_{\mathrm{GD}} = \frac{\partial Q_{\mathrm{G}}}{\partial V_{\mathrm{D}}}$$

or

$$C_{\mathrm{GG}} = \frac{\partial Q_{\mathrm{G}}}{\partial V_{\mathrm{G}}}$$

with Q_{G} the charge at terminal G in response to either V_{D} or V_{G}. Note that in a MOSFET, generally $C_{xy} \neq C_{yx}$!

2.2 Conclusion

Congratulations for making it thus far! By now you should have a solid grasp of the tool handling of Xschem and ngspice, and you should be familiar with the large- and small-signal operation of both NMOS and PMOS, and the parameters describing these behaviors. If you feel you are not sufficiently fluent in these things, please go back to the beginning of Section 2.1 and revisit the relevant sections, or dive into further reading about the MOSFET operation, like in (Hu 2010).

3 Transistor Sizing Using gm/ID Methodology

When designing integrated circuits it is an important question how to select various parameters of a MOSFET, like W, L, or the bias current I_{D}. In comparison to using discrete components in PCB design, or also compared to a bipolar junction transistor (BJT), we have these degrees of freedom, which make integrated circuit design so interesting.

Often, transistor sizing in entry-level courses is based on the square-law model, where a simple analytical equation for the drain current can be derived. However, in nanometer CMOS, the MOSFET behavior is much more complex than these simple models. Also, this highly simplified derivations introduce concepts like the threshold voltage or the overdrive voltage, which are interesting from a theoretical viewpoint, but bear little practical use.

i MOSFET Square-Law Model

One of the many simplifications of the square-law model is that the mobility of the charge carriers is assumed constant (it is not). Further, the existence of a threshold voltage is assumed, but in fact this voltage is just existing given a certain definition, and depending on definition, its value changed. In addition, in nm CMOS, the threshold voltage is a function on many thing, like W and L.

An additional shortcoming of the square-law model is that it is only valid in strong inversion, i.e. for large V_{GS} where the drain current is dominated by the drift current. As soon as the

gate-source voltage gets smaller, the square-law model breaks, as the drain current component based on diffusion currents gets dominant. Modern compact MOSFET models (like the PSP model used in SG13G2) use hundreds of parameters and fairly complex equations to somewhat properly describe MOSFET behavior over a wide range of parameters like W, L, and temperature. A modern approach to MOSFET sizing is thus based on the thought to use exactly these MOSFET models, characterize them, put the resulting data into tables and charts, and thus learn about the complex MOSFET behavior and use it for MOSFET sizing.

Being a well-established approach we select the g_m/I_D methodology introduced by P. Jespers and B. Murmann in (Jespers and Murmann 2017). A brief introduction is available here as well.

The g_m/I_D methodology has the huge advantage that it catches MOSFET behavior quite accurately over a wide range of operating conditions, and the curves look very similar for pretty much all CMOS technologies, from micrometer bulk CMOS down to nanometer FinFET devices. Of course the absolute values change, but the method applies universally.

3.1 MOSFET Characterization Testbench

In order to get the required tabulated data we use a testbench in Xschem which sweeps the length L and the terminal voltages (V_{GS}, V_{DS} and V_{SB}) for a fixed device width $W = 5\,\mu m$ and records various large- and small-signal parameters, which are then stored in large tables. The testbench for the LV NMOS is shown in Figure 9, and the TB for the LV PMOS is shown in Figure 10.

> **Note on Characterization Testbench**
>
> The testbenches are relatively straightforward, with one exception: The drain current noise is sensed via the drain voltage source `vd` and converted to a noise voltage (node `n`) using a current-controlled voltage source (CCVS). This is necessary as the `.noise` simulation statement works with voltages.

We will use Jupyter notebooks to inspect the resulting data, and interpret some important graphs. This will greatly help to understand the MOSFET behavior.

> **Note on width W**
>
> In general, the device width could be included as a fifth sweep variable. However, this is not necessary since the parameters scale approximately linearly with W across the typical range encountered in analog design. For $W > 2\,\mu m$ the error lies within about 1.5% for g_m/I_D, g_m/g_{ds} and g_m/C_{gg} as described in (Jespers and Murmann 2017).

The resulting four-dimensional data is saved into a `.txt` file and imported into two Jupyter notebooks as visualized in Figure 11. The first notebook outputs `.mat` files, which are compatible with the `pygmid` Python package used for sizing. The second notebook generates

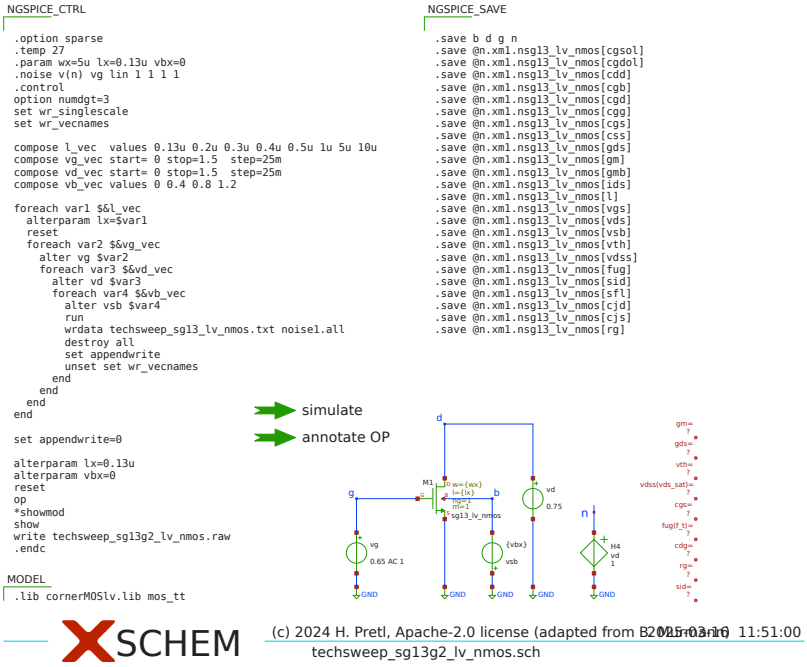


Figure 9: Testbench for LV NMOS $g_\mathrm{m}/I_\mathrm{D}$ characterization.

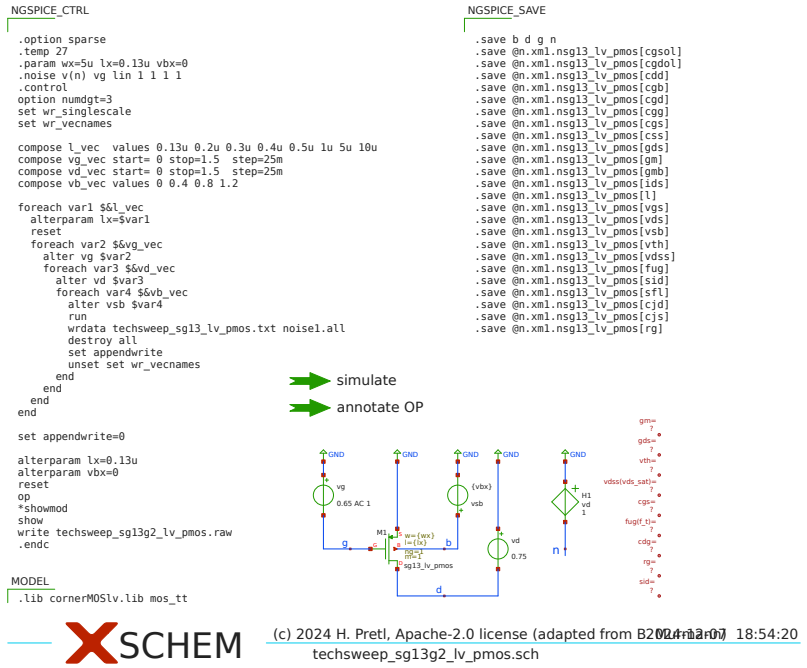


Figure 10: Testbench for LV PMOS $g_\mathrm{m}/I_\mathrm{D}$ characterization.

vector graphic characterization plots, which are shown in Section 3.2 for LV NMOS and Section 3.3 for LV PMOS. With these plots, an immediate understanding of the tradeoffs for the given transistor can be acquired. Furthermore, they can be used for fast hand calculations in small circuits.

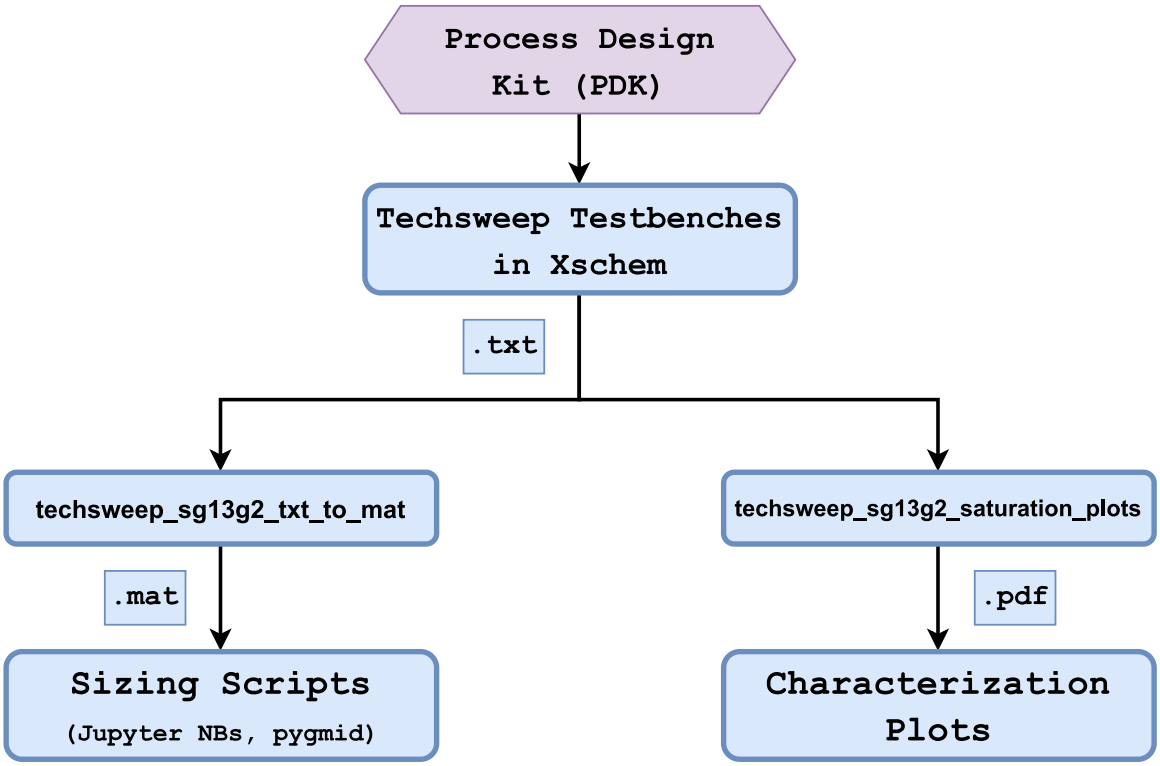


Figure 11: Overview of the g_m/I_D MOSFET characterization procedure (Dorrer 2025).

3.2 NMOS Characterization in Saturation

First, we will start looking at the LV NMOS. In Section 3.3 we have the corresponding graphs for the LV PMOS. In this lecture, we will only use the LV MOSFETs. While there are also the HV types available, they are mainly used for high-voltage circuits, like circuits connecting to the outside world. Here, we only will design low-voltage circuits running at a nominal supply voltage of 1.5 V, so only the LV types are of interest to us.

In the plots that follow we will set $V_{ds} = V_{DD}/2$ to keep the MOSFET in saturation (as this is the region of operation where most MOSFET are operated when working in class-A). We will later also look at the MOSFET performance in triode operation, as this is the operation mode where the MOSFET is used as a switch. While the g_m/I_D method is primarily intended to be used for circuits where the MOSFETs are held in saturation and are biased by a certain bias current (usually referred to class-A), the generated tables using the testbenches of Section 3.1 contain the data for all MOSFET bias points.

The first import graph is the plot of g_m/I_D and f_T versus the gate-source voltage V_{GS}. First let us answer the question why g_m/I_D is a good parameter to look at, and actually this is also the central parameter in the g_m/I_D methodology. In many circuits we want to get a large amplification from a MOSFET, which corresponds to a large g_m. We want to achieve this by spending the minimum biasing current possible (ideally zero), as we almost always design for lowest power consumption. Thus, a high g_m/I_D ratio is good.

> **Power Consumption**
>
> Designing for minimum power consumption is pretty much always mandated. For battery-operated equipment it is a paramount requirement, but also in other equipment electrical energy consumption is a concern, and often severely limited by the cooling capabilities of the electrical system.

However, as can be seen in the below plot, there exists a strong and unfortunate trade-off with device speed, characterized here by the transit frequency f_T. It would be ideal if there exists a design point where we get high transconductance per bias current concurrently to having the fastest operation, but unfortunately, this is clearly not the case. The g_m/I_D peaks for $V_{GS} < 0.3\,\mathrm{V}$, and the highest speed we get at $V_{GS} \approx 1.2\,\mathrm{V}$. The dashed vertical line plots the nominal threshold voltage, as you can see in this continuum of parameter space, it marks not a particularly special point.

Note that

$$\frac{g_m}{I_D} = \frac{1}{nV_T} \tag{3}$$

for a MOSFET in weak inversion (i.e., small gate-source voltage). n is the subthreshold slope, and $V_T = kT/q$ which is 25.8 mV at 300 K. We thus have $n \approx 1.38$ for this LV NMOS, which falls nicely into the usual range for n of 1.3 to 1.5 for bulk CMOS (FinFET have n very close to 1).

For the classical square-law model of the MOSFET in strong inversion, g_m/I_D is given as

$$\frac{g_m}{I_D} = \frac{2}{V_{GS} - V_{th}} = \frac{2}{V_{od}} \tag{4}$$

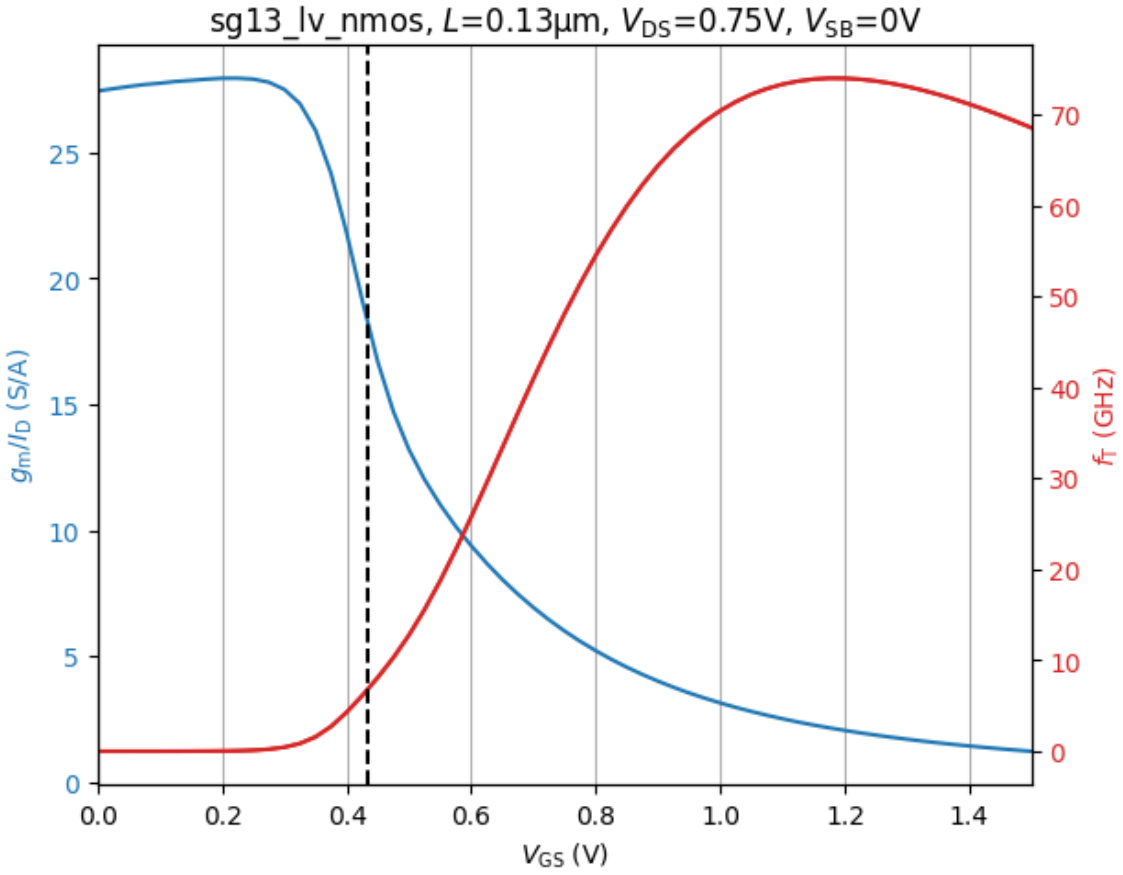
with V_{th} the threshold voltage and V_{od} the so-called "overdrive voltage." The latter is sometimes also dubbed the effective gate-source voltage V_{eff} (Carusone, Johns, and Martin 2011).

> **Why 300K?**
>
> Why are we so often using a temperature of 300 K for a typical condition? As this corresponds to roughly 27°C, this accounts for some self heating compared to otherwise cooler usual room temperatures. Further, engineers like round numbers which are easy to remember, so 300 K is used as a proxy for room temperature.

As we can also see from belows plot, the peak transit frequency of the LV NMOS is about 75 GHz, which allows building radio-frequency circuits up to ca. $f_T/10 = 7.5\,\mathrm{GHz}$, which is

a respectable number. It is no coincidence, that the transition for RF design in the GHz-range switched from BJT-based technologies to CMOS roughly in the time frame when 130nm CMOS became available (ca. 2000).

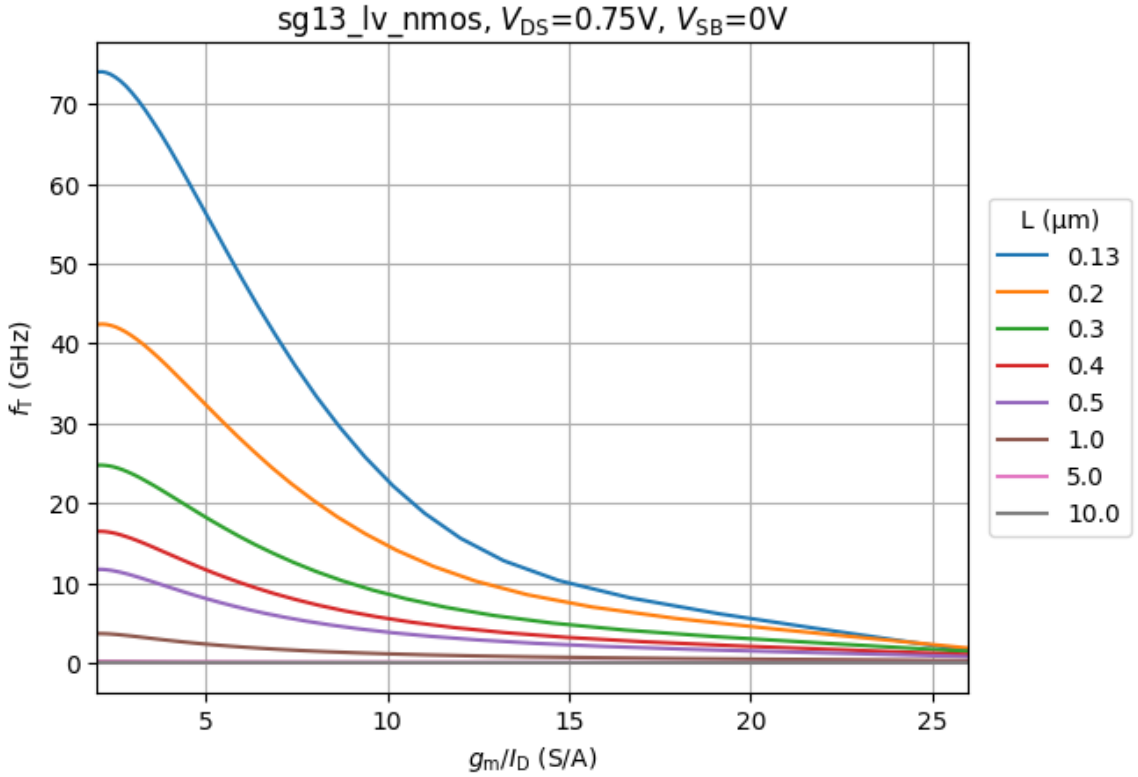
Note that f_T saturates and even decreases again at around $V_\mathrm{GS} = 1.2\,\mathrm{V}$ due to second-order effects of the transistor like velocity saturation and DIBL. Velocity saturation describes the saturation of the velocity of the electrons at a certain V_GS. As a consequence, also g_m saturates. Since $f_\mathrm{T} \propto g_\mathrm{m}/C_\mathrm{gs}$, f_T can not further increase. DIBL explains the effect when a too high V_GS is applied and therefore the channel is confined to a narrow region at the surface, leading to more carrier scattering and thus lower mobility.



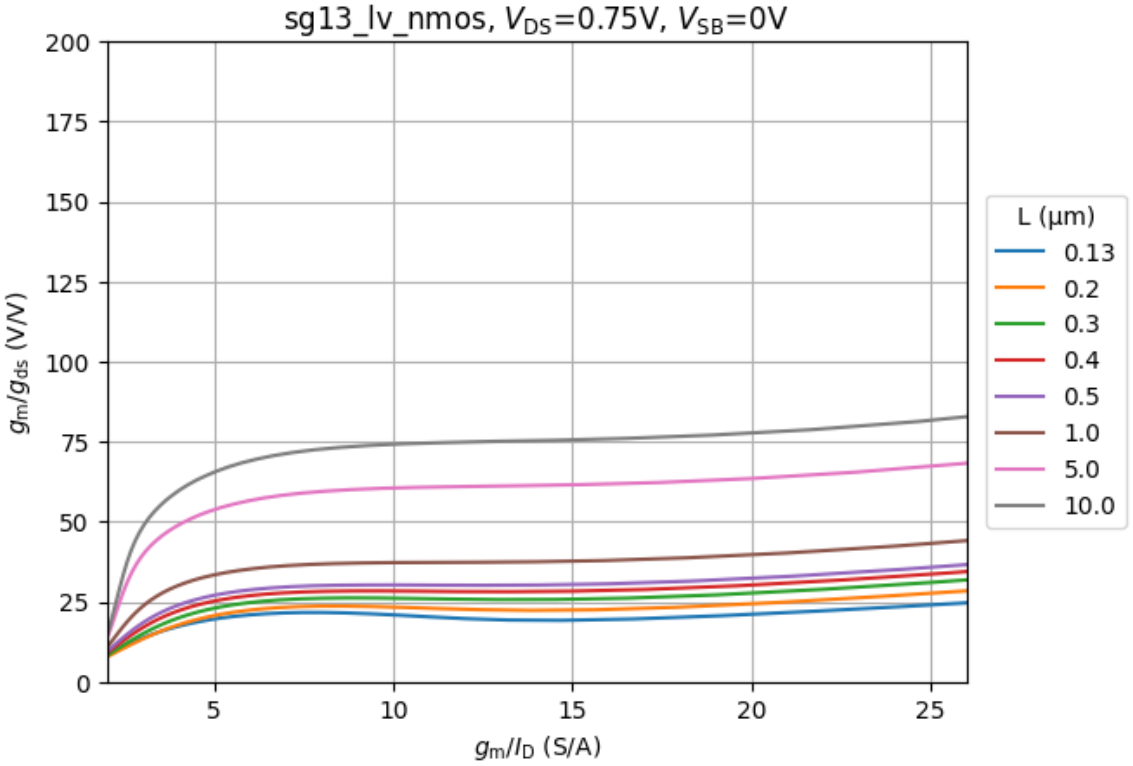
The following figure plots f_T against $g_\mathrm{m}/I_\mathrm{D}$ for several different L. As you can see, device speed maximizes for a low $g_\mathrm{m}/I_\mathrm{D}$ and a short L. Note, that the drain-source voltage is kept at $V_\mathrm{DS} = 0.75\,\mathrm{V} = V_\mathrm{DD}/2$, which is a typical value keeping the MOSFET in saturation across the characterization sweeps. Further, the source-bulk voltage is kept at $V_\mathrm{SB} = V_\mathrm{S} - V_\mathrm{B} = 0\,\mathrm{V}$, which means bulk and source terminals are connected. If $V_\mathrm{SB} \neq 0\,\mathrm{V}$, then the so-called body effect, bulk effect or back-gate effect occurs.

If $V_\mathrm{SB} > 0\,\mathrm{V}$, then V_th increases. If $V_\mathrm{SB} < 0\,\mathrm{V}$, then V_th decreases. At first glance, this effect may sound unwanted (and often also is), however, improved circuit designs by changing the bulk potential can be realized. For example, circuit 16 in (Pretl and Eberlein 2021) shows

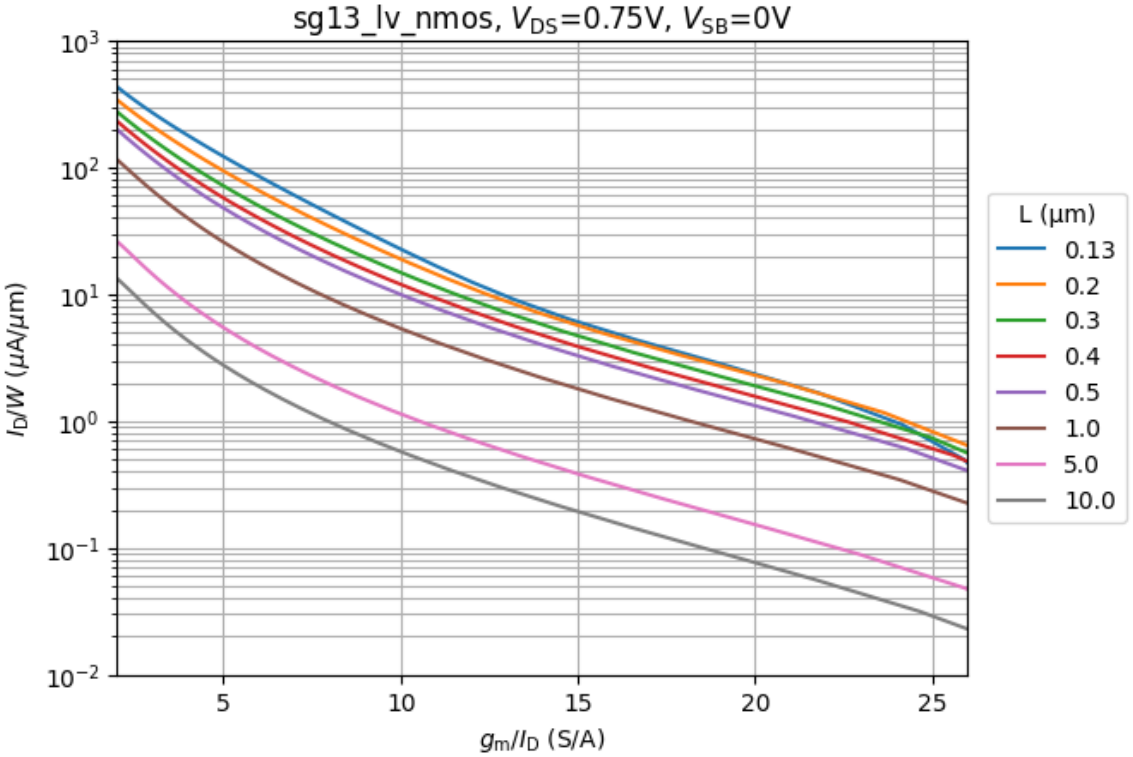
an implementation of a current mirror by exploiting the body effect. However, keep in mind that a deep-n-well (DNW), also called triple-well, is needed for NMOS transistors when the bulk is not tied to GND. This DNW may not be supported by the used PDK or can add additional costs to the wafer processing.



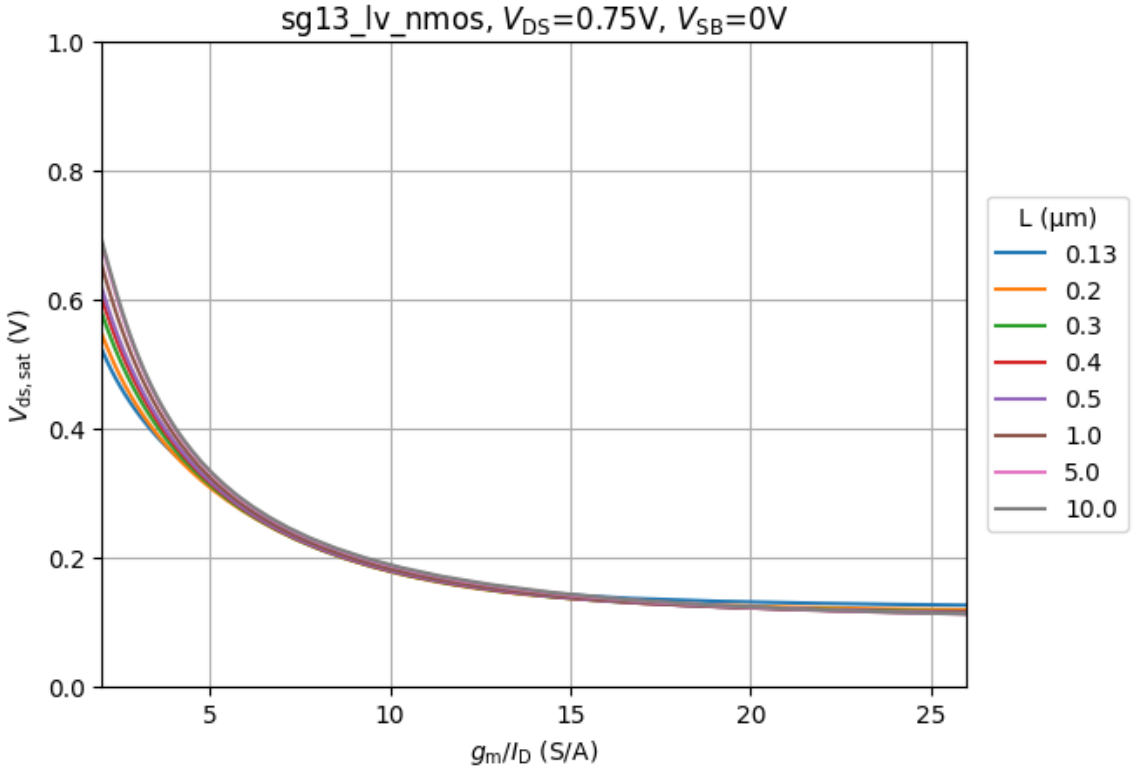
The next plot shows the ratio of g_m/g_{ds} versus g_m/I_D. The ratio g_m/g_{ds} is the so-called **"self-gain"** of the MOSFET, and shows the maximum voltage gain we can achieve in a single transistor configuration. As one can see the self gain increases for increasing L, but this also gives a slower transistor, so again there is a trade-off. This plot allows us to select the proper L of a MOSFET if we know which amount of self gain we need.



The following figure plots the drain current density I_D/W as a function of g_m/I_D and L. With this plot we can find out how to set the W of a MOSFET once we know the biasing current I_D, the L (selected according to self gain, f_T, and other considerations) and the g_m/I_D design point we selected. The drain current density I_D/W is a very useful normalized metric to use, because the physical action in the MOSFET establishes a charge density in the channel below the gate, and the changing of the W of the device merely transforms this charge density into an absolute parameter (together with L).



The following plot shows the minimum drain-source voltage $V_{\mathrm{ds,sat}}$ that we need to establish in order to keep the MOSFET in saturation. As you can see, this value is almost independent of L, and increases for small $g_\mathrm{m}/I_\mathrm{D}$. So for low-voltage circuits, where headroom is precious, we tend to bias at $g_\mathrm{m}/I_\mathrm{D} \geq 10$, wheres for fast circuits we need to go to small $g_\mathrm{m}/I_\mathrm{D} \leq 5$ requiring substantial voltage headroom per MOSFET stage that we stack on top of each other.

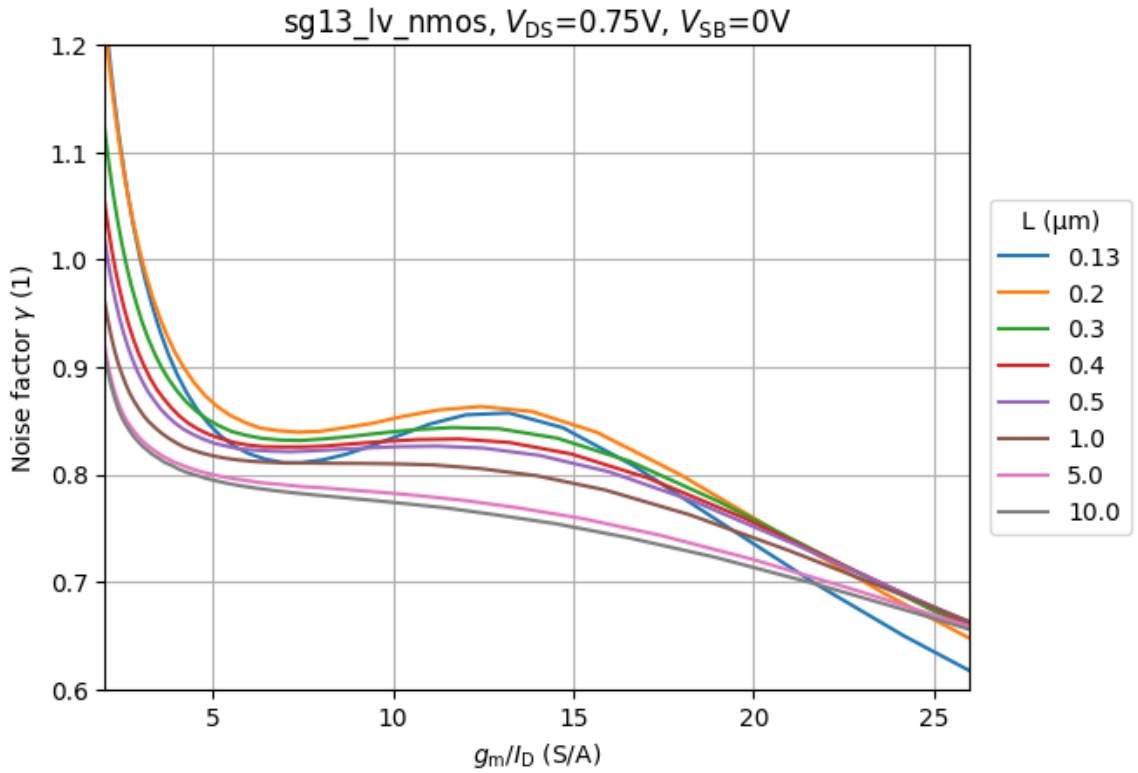


For analog circuits the noise performance is usually quite important. Thermal noise of a resistor (the Johnson-Nyquist noise) has a flat power-spectral density (PSD) given by $\overline{V_\mathrm{n}^2}/\Delta f = 4kTR$, where k is Boltzmann's constant, T absolute temperature, and R the value of the resistor (the unit of $\overline{V_\mathrm{n}^2}/\Delta f$ is V^2/Hz). This PSD is essentially flat until very high frequencies where quantum effects start to kick in.

> **Noise Notation**
>
> We usually leave the Δf away for a shorter notation, so we write $\overline{V_\mathrm{n}^2}$ when we actually mean $\overline{V_\mathrm{n}^2}/\Delta f$. In case of doubt look at the unit of a quantity, whether is shows V^2 or V^2/Hz or $\mathrm{V}/\sqrt{\mathrm{Hz}}$ (or I^2 or I^2/Hz or $\mathrm{I}/\sqrt{\mathrm{Hz}}$).
> Please also note that the pair of kT pretty much always shows up together, so when you do a calculation and you miss the one or the other, that is often a sign for miscalculation. Boltzmann's constant $k = 1.38 \cdot 10^{-23}\,\mathrm{J/K}$ is just a scaling factor from thermal energy expressed as a temperature T to energy $E = kT$ expressed in Joule.
> Further, when working with PSD there is the usage of a one-sided ($0 \geq f < \infty$) or two-sided power spectral density (PSD) ($-\infty < f < \infty$). The default in this lecture is the usage of the **one-sided PSD**.

In this lecture the only MOSFET noise we consider is the drain noise (as discussed in Section 2.1.2), showing up as a current noise between drain and source. For a realistic MOSFET noise model, also a (correlated) gate noise component and the thermal noise of the gate resistance needs to be considered.

The factor γ (Equation 1) is a function of many things (in classical theory, $\gamma = 2/3$ in saturation and $\gamma = 1$ in triode), and it is characterized in the following plot as a function of $g_\mathrm{m}/I_\mathrm{D}$ and L. So when calculating MOSFET noise we can lookup γ in the below plot, and use Equation 1 to calculate the effective drain current noise.



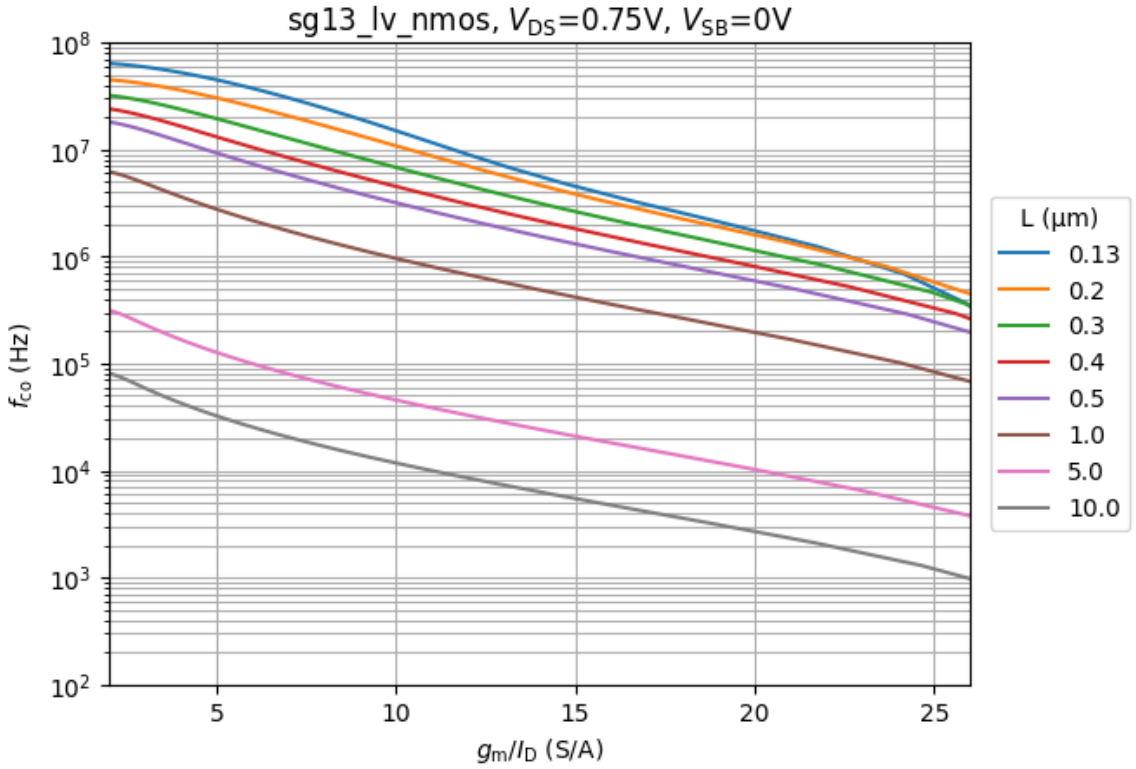
In a MOSFET, unfortunately, besides the thermal noise according to Equation 1, there is also a substantial low-frequency excess noise, called "flicker noise" due to its characteristic $\overline{I^2_\mathrm{d,nf}} = K_\mathrm{f}/f$ behavior (this means that this noise PSD decreases versus frequency). In order to characterize this flicker noise the following plot shows the cross-over frequency f_co, where the flicker noise is as large as the thermal noise. As can be seen in the below plot, this frequency is a strong function of L and $g_\mathrm{m}/I_\mathrm{D}$. Generally, the flicker noise is proportional to $(WL)^{-1}$, so the larger the device is, the lower the flicker noise. The parameter $g_\mathrm{m}/I_\mathrm{D}$ largely stays constant when we keep W/L constant, so for a given $g_\mathrm{m}/I_\mathrm{D}$ flicker noise is proportional to $1/L^2$. However, increasing L lowers device speed dramatically, so here we have a trade-off between flicker-noise performance and MOSFET speed, and this can have dramatic consequences for high-speed circuits.

MOSFET Flicker Noise

The physical origin of flicker noise is the crystal interface between silicon (Si) and the silicon dioxide (SiO_2). Since these are different materials, there are dangling bonds, which can capture charge carriers traveling in the channel. After a random time, these carriers are released, and flicker noise is the result. The amount of flicker noise is a

function of the manufacturing process, and will generally be different between device types and wafer foundries.

As you can see in the following plot, f_{co} can reach well into the 10's of MHz for short MOSFETs, significantly degrading the noise performance of a circuit.



During the design phase, it might be convenient to have an overview of the most relevant sizing plots of the NMOST on one page. This overview can be downloaded here.

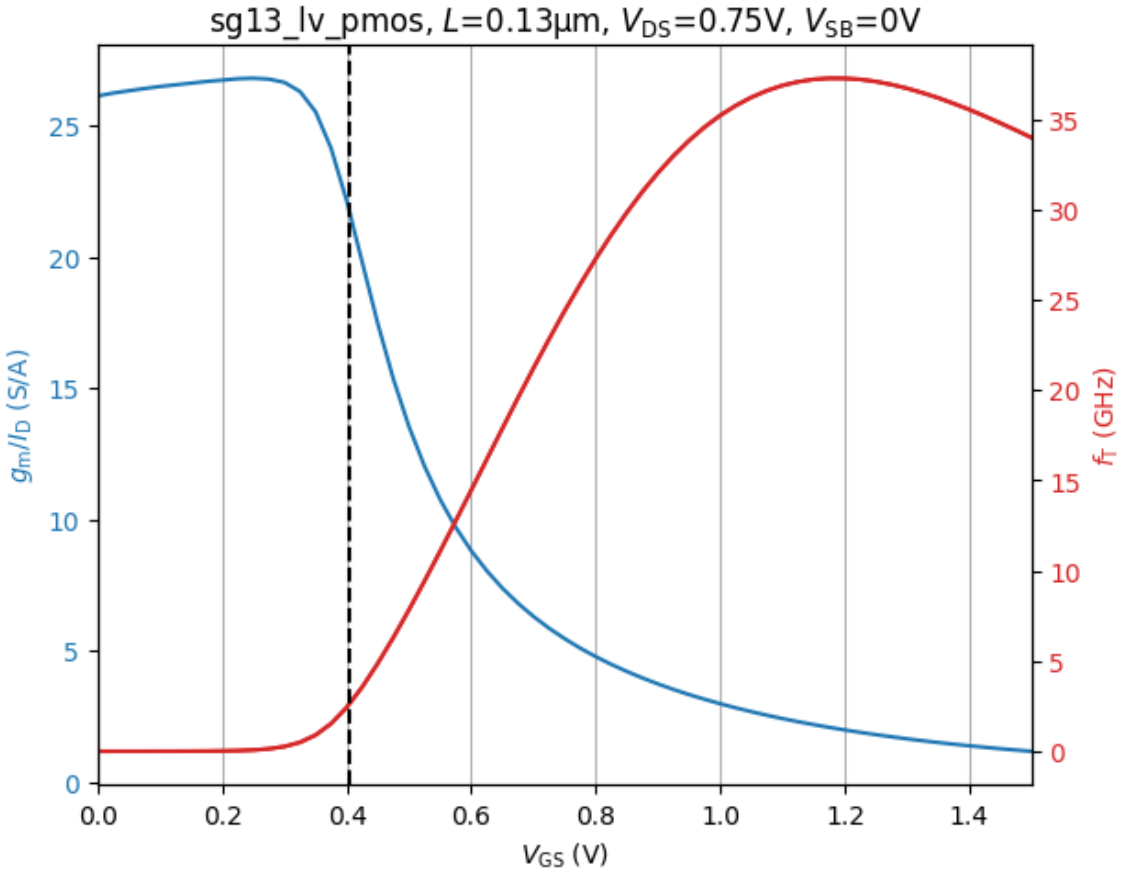
3.3 PMOS Characterization in Saturation

In the following, we have the same plots as discussed in Section 3.2, but now for the PMOS.

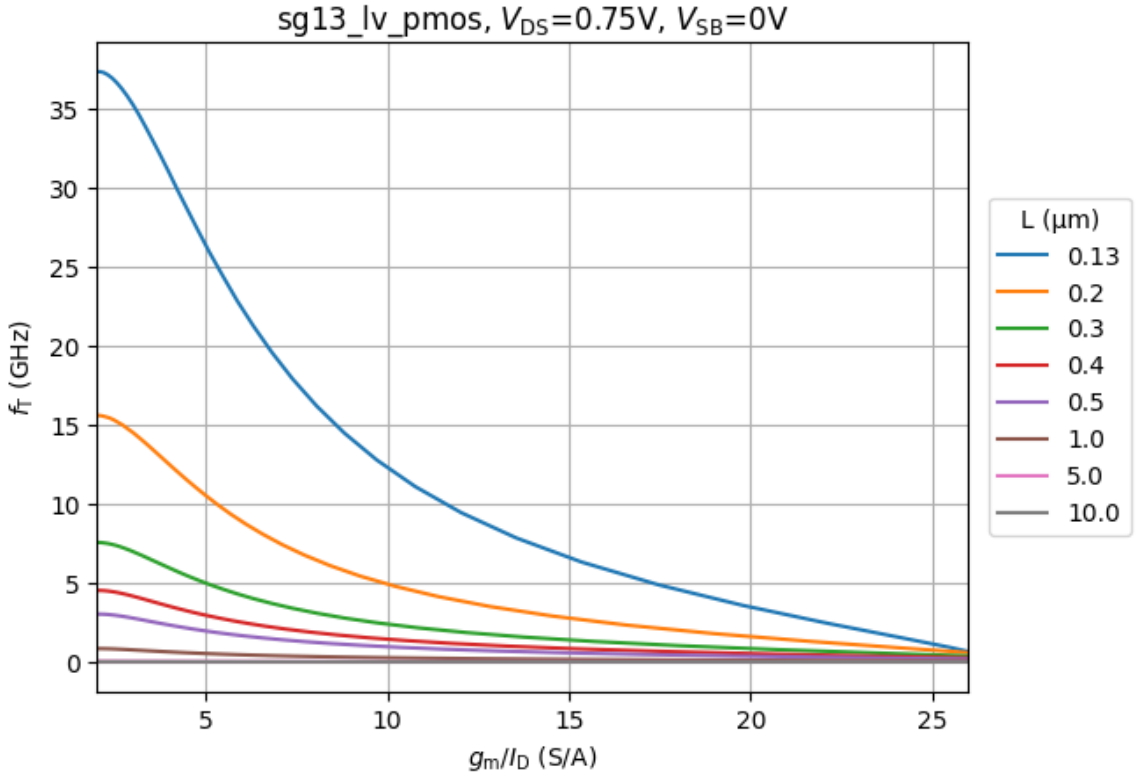
PMOS Sign Convention

In all PMOS plots we plot positive values for voltages and currents, to have compatible plots to the NMOS. Of course, in a PMOS, voltages and currents have different polarity compared to the NMOS.

g_m/I_D and f_T versus the gate-source voltage V_{GS}:



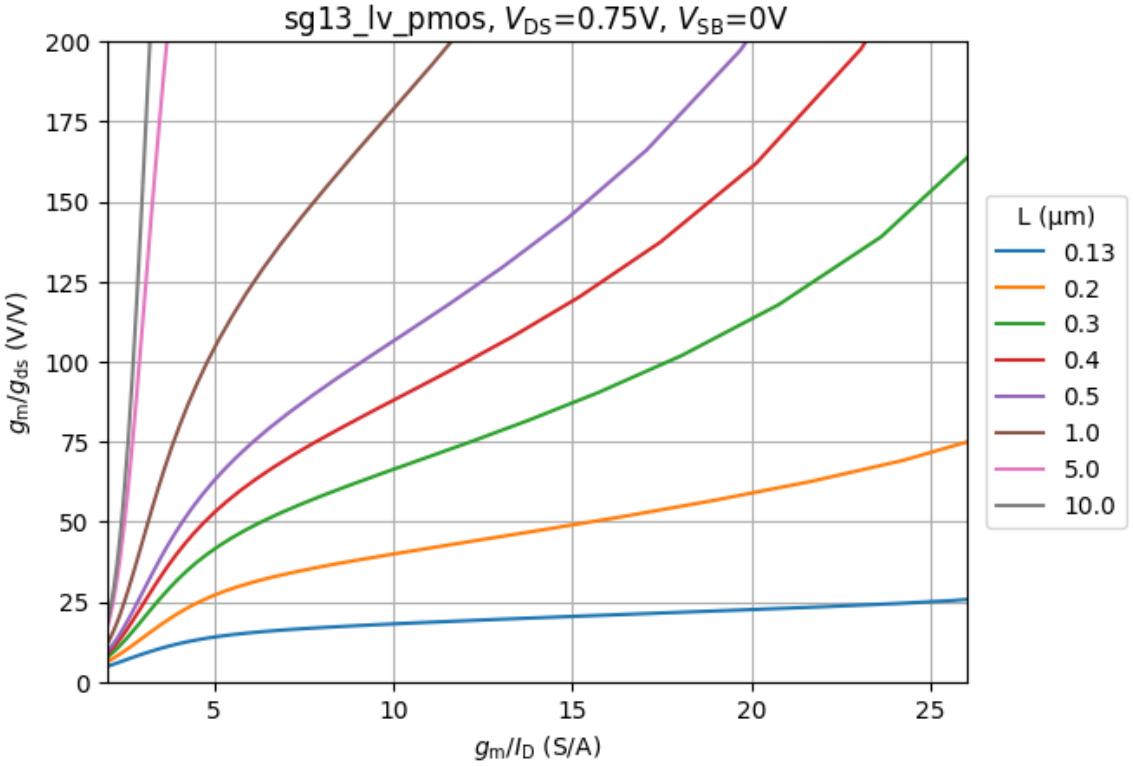
f_T against $g_\mathrm{m}/I_\mathrm{D}$ for several different L. One can see a significantly lower top speed for the PMOS compared to the NMOS, which means for high-speed circuits the NMOS should be used. The reason for this is the approximately two to three times higher mobility of electrons compared to the mobility of holes.



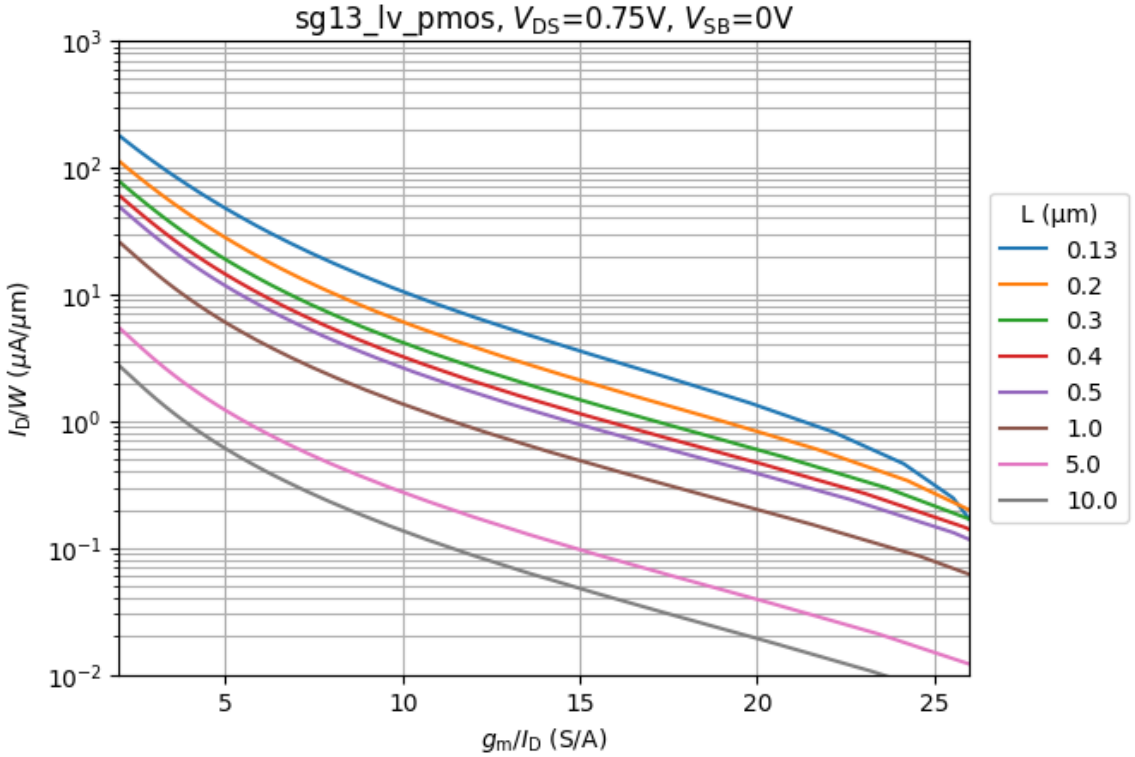
g_m/g_{ds} versus g_m/I_D. Unfortunately, one can see a modelling error for the PMOS in this plot. The self gain g_m/g_{ds} reaches non-physical values, which indicates an issue with the g_{ds} modelling for the PMOS. We can not use these values for our circuit sizing, so we will use the respective NMOS plots also for the PMOS.

! Beware of Modelling Issues

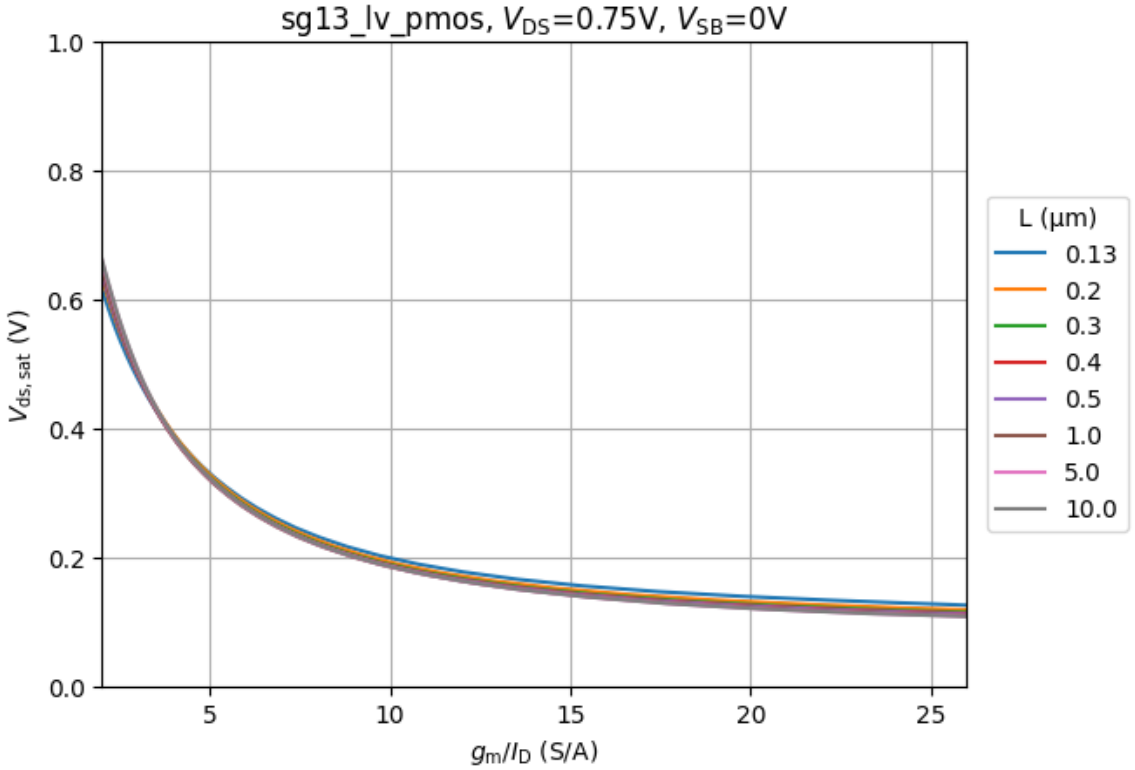
This example shows how important it is to benchmark the device models when starting to use a new technology. Modelling artifacts like the one shown are quite often happening, as setting up the device compact models and parametrize them according to measurement data is a very complex task. In any case, just be aware that modelling issues could exist in whatever PDK you are going to use!



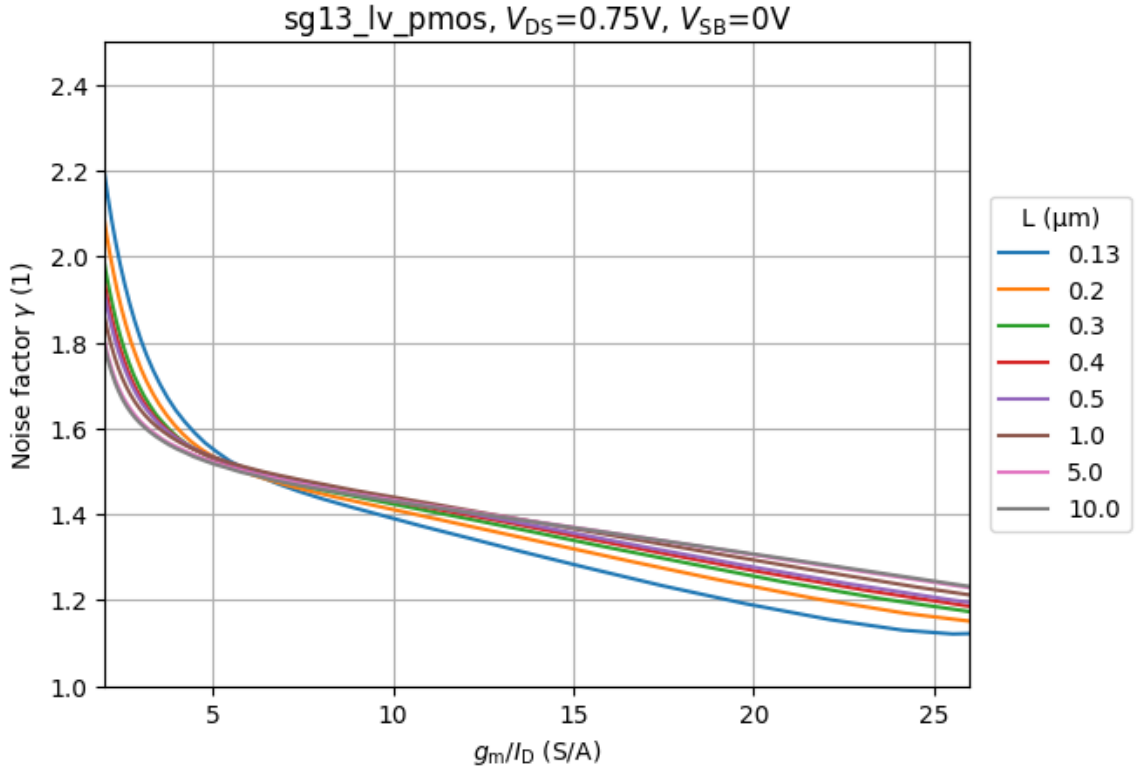
Drain current density I_{D}/W as a function of $g_{\mathrm{m}}/I_{\mathrm{D}}$ and L:



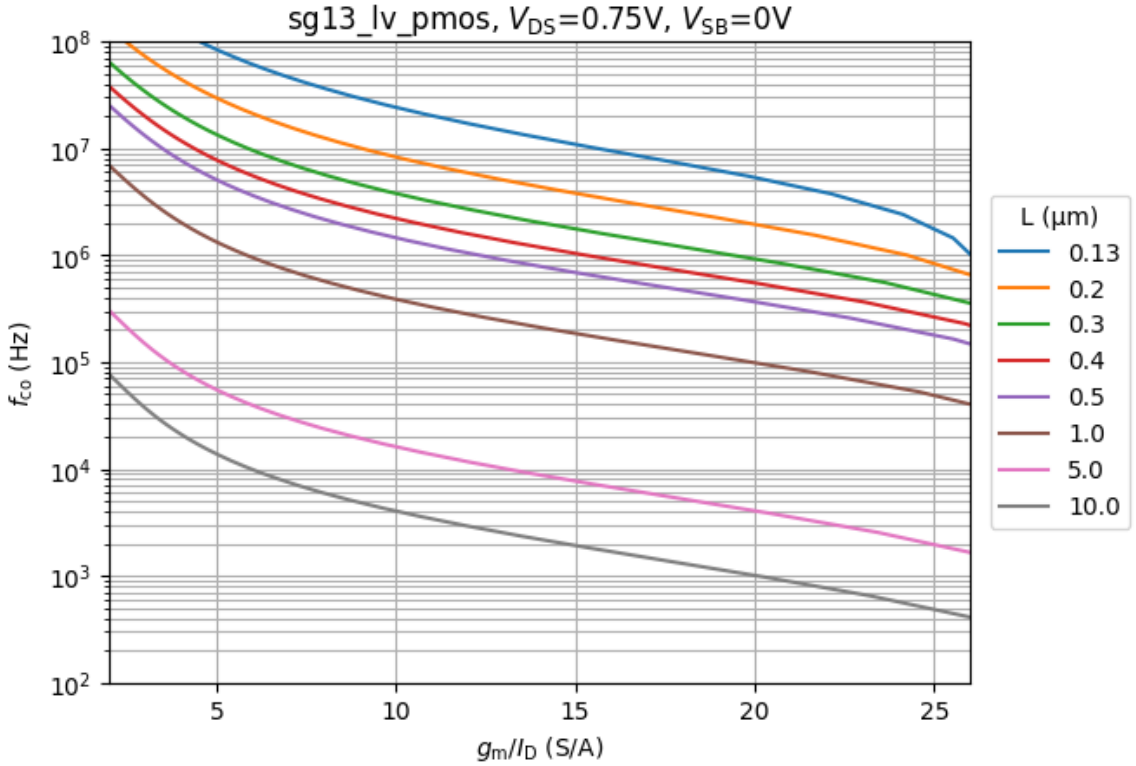
Minimum drain-source voltage $V_{\mathrm{ds,sat}}$ versus $g_{\mathrm{m}}/I_{\mathrm{D}}$ and L:



Noise factor γ versus $g_{\mathrm{m}}/I_{\mathrm{D}}$ and L:



Flicker noise corner frequency f_{co} versus g_m/I_D and L. If you compare this figure carefully with the NMOS figure you can see that for some operating points the flicker noise for the PMOS is lower than for the NMOS. This is often true for CMOS technologies, so it can be an advantage to use a PMOS transistor in places where flicker noise is critical, like an OTA input stage. Using PMOS has the further advantage that the bulk node can be tied to source (which for NMOS is only possible in a triple-well technology, which is often not available), which gets rid of the body effect.



During the design phase, it might be convenient to have an overview of the most relevant sizing plots of the PMOST on one page. This overview can be downloaded here.

3.4 Tradeoffs in Saturation

In Figure 12, an overview of the tradeoffs mentioned above is presented. The design parameters are only shown once for their best-case implementation. Their worst-case value is then located at the opposite arrow. In conclusion, strong inversion with short channels is used for high-speed applications, whereas weak inversion with long channels is for low-power and low-noise applications. Moderate inversion with longer channels is used for current sources to maximize their output resistance. Note that the two arrows for strong inversion with long channels and weak inversion with short channels are empty since they make practically no sense. This tradeoff overview simplifies the choice of the starting g_m/I_D and L values in the sizing scripts.

3.5 NMOS and PMOS Characterization in Triode

Besides using the MOSFET as a transconductor in saturation we often use the MOSFET as a switch in triode mode (to either switch voltages or currents). In this triode/switch mode of operation we are mainly interested in two parameters:

- The resistance of the switch/MOSFET when it is turned on ($R_{on} = 1/g_{ds}$).

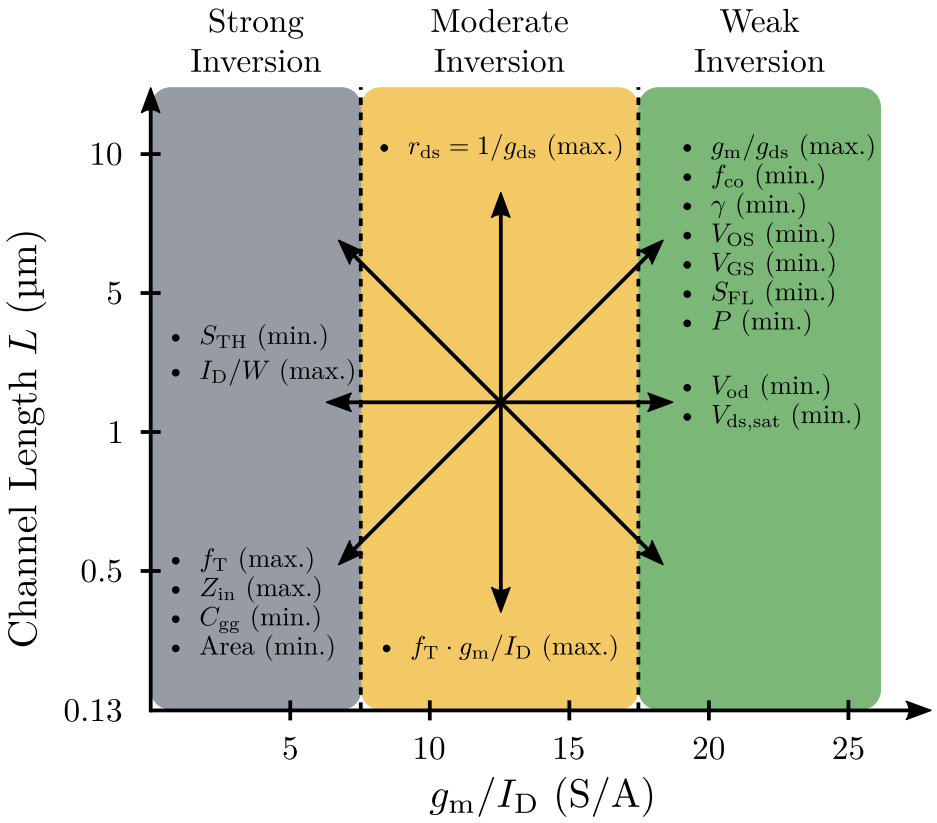
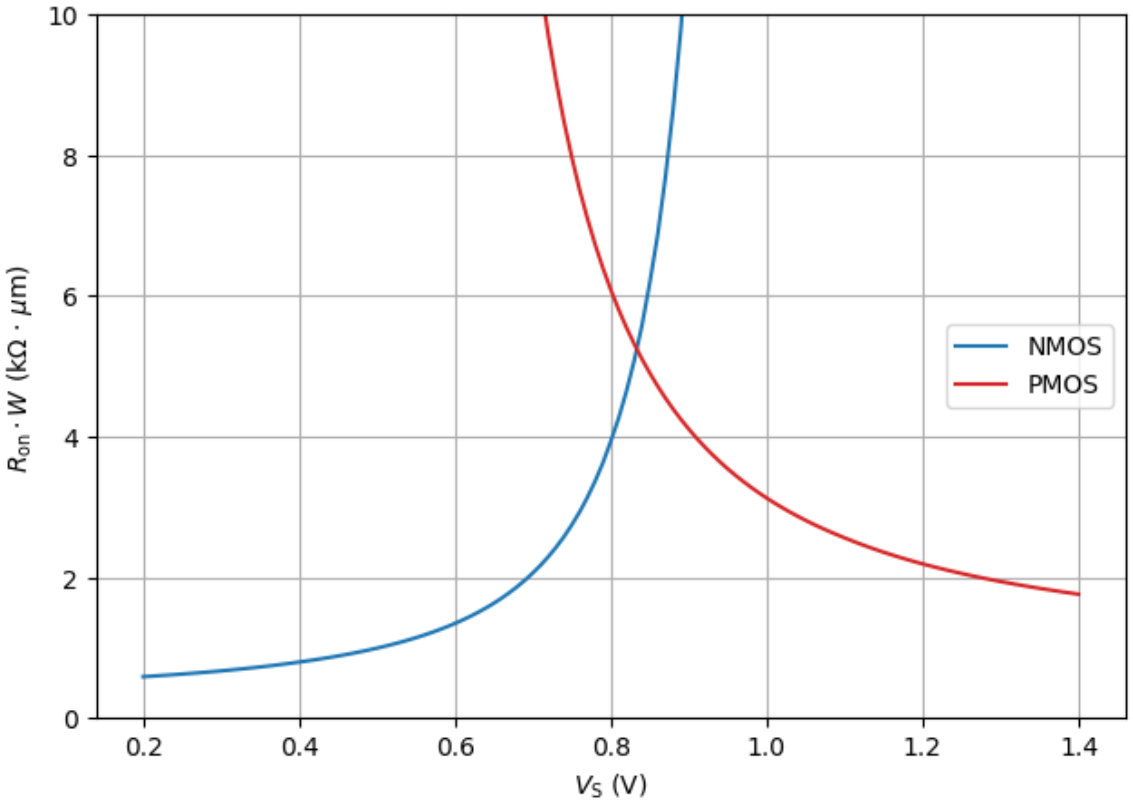


Figure 12: Overview of key design tradeoffs depending on the channel length L and transconductance efficiency g_m/I_D (Dorrer 2025).

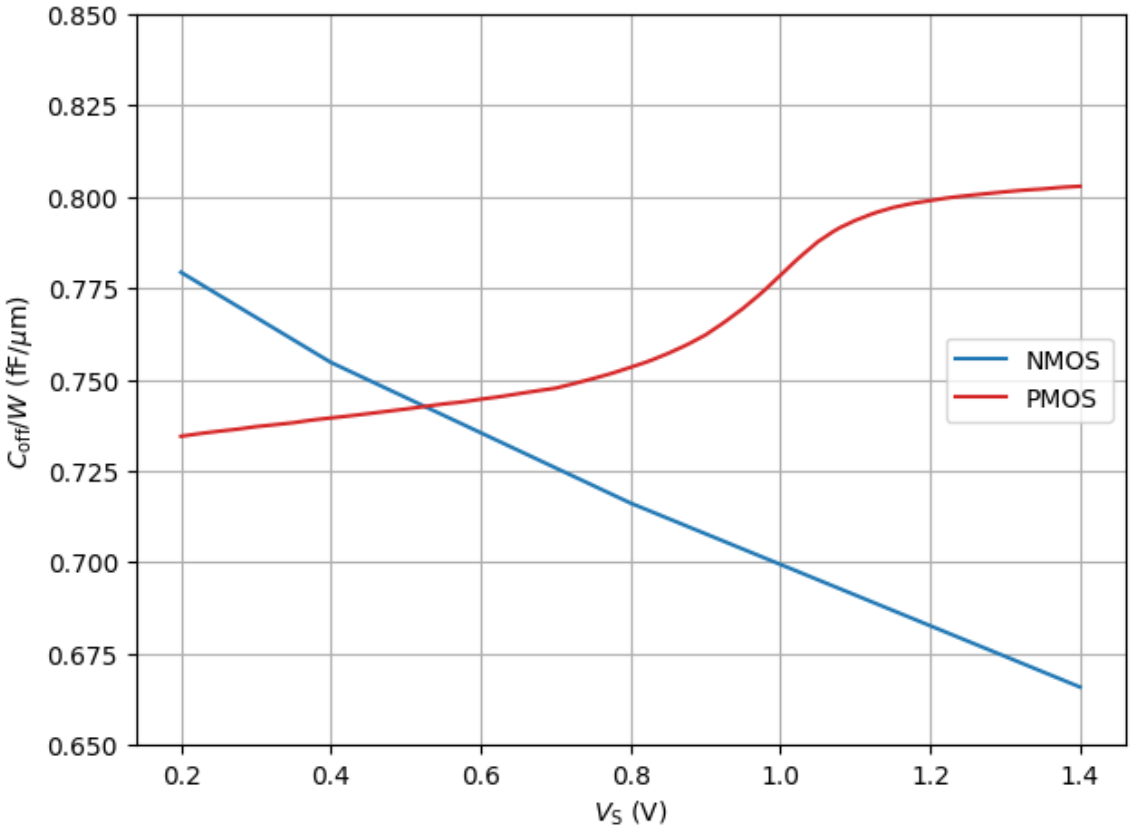
- The shunt capacitance of the switch when it is turned off (C_{off} is defined by the coupling capacitances between drain and source).

In the operation of the NMOS as a switch the gate is usually pulled to V_{DD} and the bulk is permanently connected to V_{SS} to achieve the lowest R_{on} (to turn the switch off the gate is pulled towards V_{SS}). Likewise, to turn on a PMOS, the gate is usually pulled to V_{SS} and the bulk is connected to V_{DD}. In this situation the drain/source potential is somewhere between V_{DD} and V_{SS}, so the MOSFET will experience a $V_{\text{SB}} \neq 0$, leading to a quite noticeable bulk effect. Once the drain/source potential is sufficiently high for NMOS (low PMOS) the switch resistance will drastically degrade.

In order to get a feeling for the behavior of the MOSFET as a switch the following characterization plots show R_{on} and C_{off} for the NMOS and the PMOS, respectively. Both plots are for $L = L_{\text{min}} = 0.13\mu m$, as for switches usually minimum length devices are used. Only for special applications (e.g., the drain-source leakage current in off-mode is a concern) MOSFETs with increased L are used.



As can be seen in the previous plot, an NMOS can be used to switch at potentials close to V_{SS}, while a PMOS is the better choice when switching at potentials close to V_{DD}. To construct a switch which can work for all voltage levels between V_{DD} and V_{SS} an NMOS is put in parallel to a PMOS, resulting in the well-known **transmission gate**.



As the switch on-resistance gets lower when increasing W, the off-capacitance gets larger. Thus, a good performance indicator for comparing switches in a given technology is the $R_{on}C_{off}$ product.

3.6 Tradeoffs in Triode

Some might think that sizing switches is a straightforward task, but in reality, they also come with several tradeoffs, and depending on the application, they can be incredibly complicated. The essential design parameters in switches are the on-resistance R_{on}, the off-capacitance C_{off}, the drain-source leakage current I_{leak}, and their impact on clock feedthrough and charge injection (Carusone, Johns, and Martin 2011). In Figure 13, an overview of the tradeoffs of these design parameters depending on the channel width W and the channel length L is presented. In conclusion, if W/L increases, R_{on} decreases, and the switching speed increases. If W/L decreases, I_{leak}, C_{off}, and the drain-source overlap capacitance decrease, effectively reducing clock feedthrough and charge injection (Gray et al. 2009). Practically, it does not make sense to make the switch both wide and long.

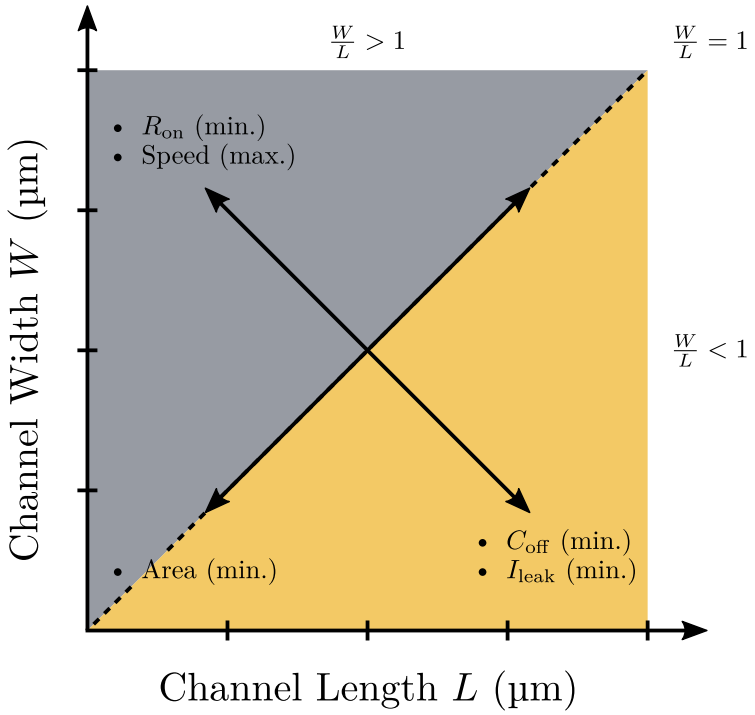


Figure 13: Overview of key design tradeoffs in switches depending on the channel width W and the channel length L (Dorrer 2025).

4 First Circuit: MOSFET Diode

The first (simple) circuit which we will investigate is a MOSFET, where the gate is shorted with the drain, a so-called MOSFET "diode", which is shown in Figure 14. This diode is one half of a current mirror, which we will investigate in Section 6.

Why looking at a single-transistor circuit at all? By starting with the simplest possible circuit we can develop important skills in circuit analysis (setting up and calculating a small-signal model, calculating open-loop gain, calculate noise, etc.) and Xschem/ngspice simulation testbench creation. We safely assume that also the Mona Lisa was not Leonardo da Vinci's first painting, so let's start slow.

This diode is usually biased by a current source, shown as I_{bias} in the figure. Depending on MOSFET sizing with W and L, a certain gate-source voltage V_{GS} will develop. This voltage can be used as a biasing voltage for other circuit parts, for example.

Feedback in the MOSFET Diode

It is important to realize that this configuration employs a feedback loop for operation. The voltage at the drain of the MOSFET is sensed by the gate, and the gate voltage changes until I_D is exactly equal to I_{bias}. In this sense this is probably the smallest feedback circuit one can build.

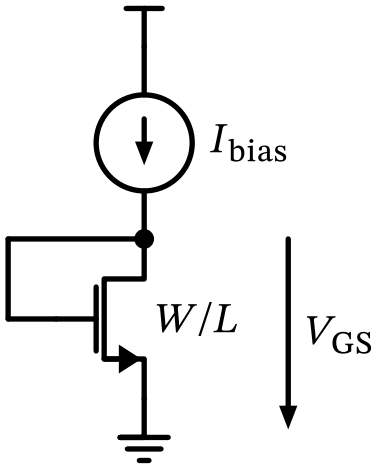


Figure 14: A MOSFET connected as a diode (drain shorted with gate).

4.1 MOSFET Diode Sizing

We will now build this circuit in Xschem. For sizing the MOSFET we will use the g_m/I_D methodology introduced in Section 3.

Exercise: MOSFET Diode Sizing

Please build a MOSFET diode circuit in Xschem where you use an LV NMOS, set $I_{\text{bias}} = 20\,\mu\text{A}$, $L = 0.13\,\mu\text{m}$, and we want to use $g_m/I_D = 10$ (often a suitable compromise between transistor speed and g_m efficiency).

1. Use the figures in Section 3.2 to find out the proper value for W.
2. What is the f_T for this MOSFET? What is the value for g_m and g_{ds}?
3. Draw the circuit in Xschem, and simulate the operating point. Do the values match to the values found out before during circuit sizing?

Before continuing, please finish the previous exercise. Once you are done, compare with the below provided solution.

Solution: MOSFET Diode Sizing

1. Using the fact that $I_{\text{bias}} = I_D = 20\,\mu\text{A}$ and $g_m/I_D = 10$ directly provides $g_m = 0.2\,\text{mS}$.
2. Using the self-gain plot, we see that $g_m/g_{ds} \approx 21$, so $g_{ds} \approx 9.5\,\mu\text{S}$. The f_T can easily be found in the respective plot to be $f_T = 23\,\text{GHz}$.
3. The W of the MOSFET we find using the drain current density plot and the given bias current. Rounding to half-microns results in $W = 1\,\mu\text{m}$.
4. Since we are looking at the graphs, we further find $\gamma = 0.84$, $V_{\text{ds,sat}} = 0.18\,\text{V}$, and $f_{\text{co}} \approx 15\,\text{MHz}$.

5. In addition, we expect $V_{GS} \approx 0.6\,\mathrm{V}$.

An example Jupyter notebook to extract these values accurately you can find here. An Xschem schematic for this exercise is provide as well.

4.2 MOSFET Diode Large-Signal Behavior

As discussed above, the MOSFET diode configuration is essentially a feedback loop. Before we will analyze this loop in small-signal, we want to investigate how this loop settles in the time domain, and by doing this we can observe the large-signal settling behavior. To simulate this, we change the dc bias source from the previous example to a transient current source, which we will turn on after some picoseconds. The resulting Xschem testbench is shown in Figure 15.

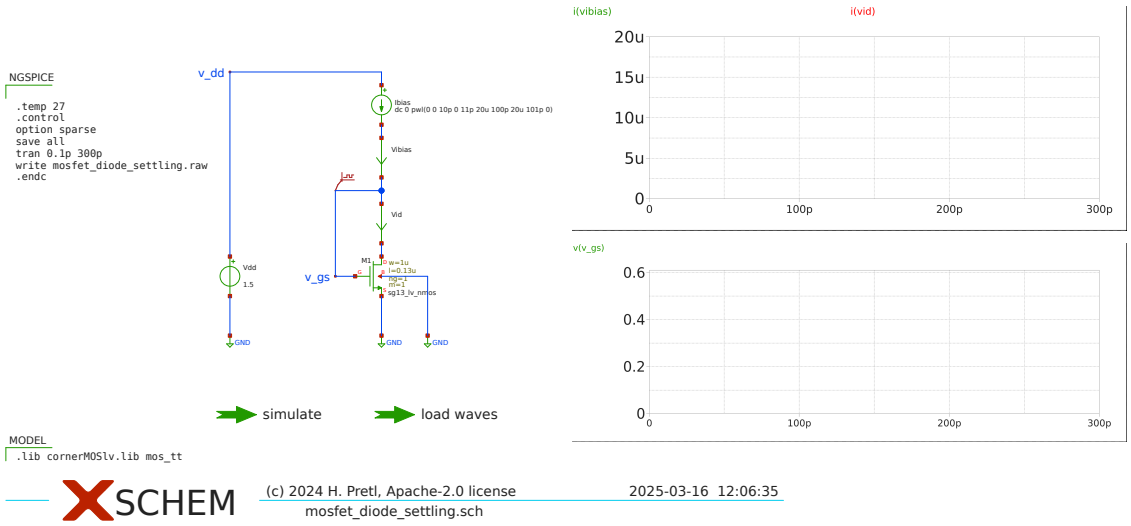


Figure 15: Testbench for MOSFET diode transient settling.

When simulating the circuit in Figure 15 another interesting effect can be observed: While the turn-on happens quite rapidly (essentially the bias current source charges the gate capacitance, until the gate-source voltage is large enough that the drain current counteracts the bias current), the turn-off shows a very long settling tail. This is due to the fact that as the gate capacitance is discharged by the drain current, the V_{GS} drops, which in turn reduces the drain current, which will make the discharge even slower. We have an effect similar to a capacitor discharge by a diode (Hellen 2003).

Power-Down Switches

It is thus generally a good idea to add power-down switches to the circuits to disable the circuit quickly by pulling floating nodes to a defined potential (usually V_{DD} or V_{SS}) and to avoid long intermediate states during power down. This will also allow a turn-on from a well-defined off-state.

Exemplary implementations of power-down switches are shown in the Xschem implementation of the improved OTA in Figure 63. Which W/L ratio should these transistors have? Well, in general, switches often have minimum length to be fast and to have low R_{on}. However, if there are no critical specifications (e.g., like power-down time), then these transistors are often used as dummy transistors for other circuit parts and are sized to fit the layout best.

4.3 MOSFET Diode Small-Signal Analysis

We now want to investigate the small-signal behavior of the MOSFET diode. Based on the small-signal model of the MOSFET in Figure 5 we realize that gate and drain are shorted, and we also connect bulk to source. We can thus simplify the circuit to the one shown in Figure 16.

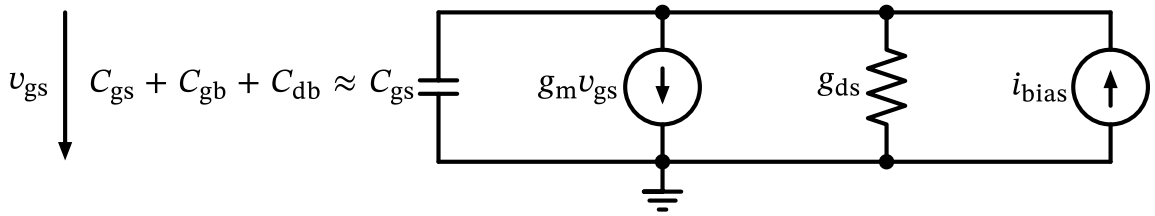


Figure 16: The MOSFET diode small-signal model (drain and gate are shorted, as well as source and bulk).

Ground Node Selection

For small-signal analysis we would not need to declare one node as the ground potential. However, when doing so, and selecting the ground node strategically, we can simplify the analysis, as we usually do not formulate KCL for the ground node (as we have only $N-1$ independent KCL equations, N being the number of nodes in the circuit), and the potential difference equations are simpler if one node is at $0\,V$.

For calculating the small-signal impedance of the MOSFET diode we formulate Kirchhoff's current law (KCL, also Kirchhoff's first law or Kirchhoff's junction rule) at the top node to get

$$i_{bias} - sC_{gs}v_{gs} - g_m v_{gs} - g_{ds}v_{gs} = 0.$$

It follows that

$$Z_{\text{diode}}(s) = \frac{v_{\text{gs}}}{i_{\text{bias}}} = \frac{1}{g_{\text{m}} + g_{\text{ds}} + sC_{\text{gs}}}. \tag{5}$$

When neglecting g_{ds}, at dc we get $Z_{\text{diode}} = 1/g_{\text{m}}$, which is an important result and should be memorized.

! The Admittance is Your Friend

In circuit analysis it is often algebraically easier to work with admittance instead of impedance, so please remember that Ohm's law for a conductance is $I = G \cdot V$, and for a capacitance is $I = sC \cdot V$. When writing equations, it is also practical to keep sC together, so we will strive to sort terms accordingly.

Looking at Equation 5 we see that for low frequencies, the diode impedance is resistive, and for high frequencies it becomes capacitive as the gate-source capacitance starts to dominate. The corner frequency of this low-pass can be calculated as

$$\omega_{\text{c}} = \frac{g_{\text{m}} + g_{\text{ds}}}{C_{\text{gs}}} \approx \omega_{\text{T}}$$

which is pretty much the transit frequency of the MOSFET!

4.4 MOSFET Diode Stability Analysis

i Open-Loop Gain, Closed-Loop Gain, and Loop-Gain—A Short Recap

Figure 17 shows a negative feedback system with input $X(s)$ and output $Y(s)$, where $H_{\text{ol}}(s)$ is the transfer function of the feed-forward path (also called **open-loop gain**) and $G(s)$ is the transfer function of the feedback network. The **loop-gain** is the product of both transfer functions $T(s) = H_{\text{ol}}(s)G(s)$ and is used for the stability analysis. The **closed-loop** gain is defined as $H_{\text{cl}}(s) = Y(s)/X(s)$ can be derived with $Y(s) = H_{\text{ol}}(s)[X(s) - Y(s)G(s)]$ to be

$$H_{\text{cl}}(s) = \frac{Y(s)}{X(s)} = \frac{H_{\text{ol}}(s)}{1 + H_{\text{ol}}(s)G(s)} = \frac{H_{\text{ol}}(s)}{1 + T(s)} \tag{6}$$

If the open-loop gain is sufficiently large $H_{\text{ol}}(s) \gg 1$, then the closed-loop gain simplifies to $H_{\text{cl}}(s) \approx 1/G(s)$. This result is convenient, since it is independent of $H_{\text{ol}}(s)$. Therefore, the overall gain is only set with the feedback gain $G(s)$ in operational amplifier circuits.
In the case of the MOSFET diode, $G(s) = 1$ and therefore $T(s) = H_{\text{ol}}(s)$ and $H_{\text{cl}}(s) \approx 1$. Therefore, this chapter uses open-loop gain and loop-gain as synonyms.

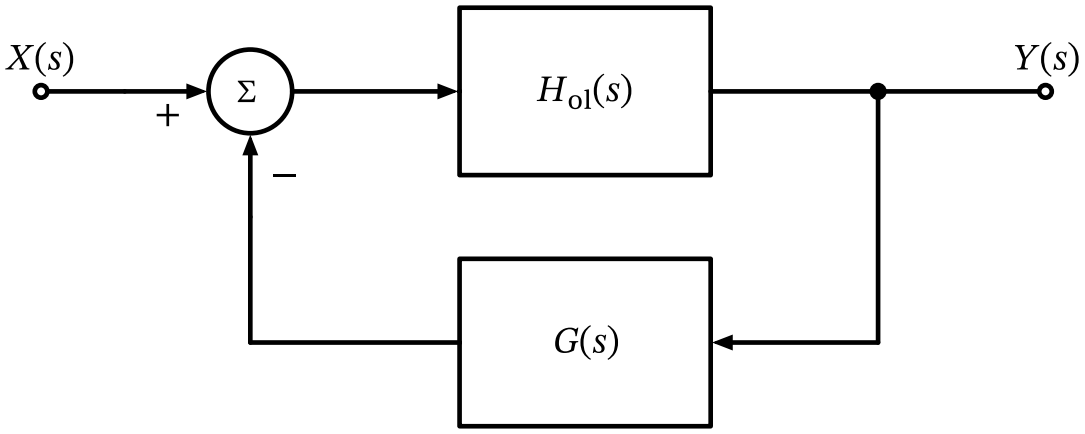


Figure 17: The block diagram of a negative feedback system.

Gain-Bandwidth Product in Feedback Systems

The gain-bandwidth product (GBP or GBWP) or transit frequency f_T of a first-order open-loop system is the product of the open-loop dc gain $H_\mathrm{ol,dc} = H_\mathrm{ol}(f = 0\,\mathrm{Hz})$ and the open-loop $-3\,\mathrm{dB}$ cut-off frequency $f_\mathrm{c,ol}$ of $H_\mathrm{ol}(s)$.

$$\mathrm{GBWP} = f_\mathrm{T,ol} = H_\mathrm{ol,dc} f_\mathrm{c,ol}$$

If a frequency-independent negative feedback G (e.g., a resistive divider) is applied to this open-loop system, the transit frequency changes to

$$f_\mathrm{T,cl} = f_\mathrm{T,ol}\sqrt{1 - G^2}$$

Hence, the closed-loop transit frequency is slightly lower than the open-loop transit frequency $f_\mathrm{T,cl} < f_\mathrm{T,ol}$.
The closed-loop $-3\,\mathrm{dB}$ cut-off frequency $f_\mathrm{c,cl}$ can then be calculated from the open-loop transit frequency and the feedback gain.

$$f_\mathrm{c,cl} = H_\mathrm{ol,DC} f_\mathrm{c,ol} G = f_\mathrm{T,ol} G$$

This theory might be interesting when Middlebrook's and Tian's methods for loop gain analysis are later compared in the MOSFET diode testbench (see Figure 19).

The diode-connected MOSFET forms a feedback loop. What is the open-loop gain? For calculating it, we are breaking the loop, and apply a dummy C^*_gs at the right side to keep the impedances correct. A circuit diagram is shown in Figure 18, we break the loop at the dotted connection. As we can see in this example, it is critically important when breaking up a loop for analysis (also for simulation!) to keep the terminal impedances the same. Only in special cases where the load impedance is very high or the driving impedance is very low is it acceptable to disregard loading effects!

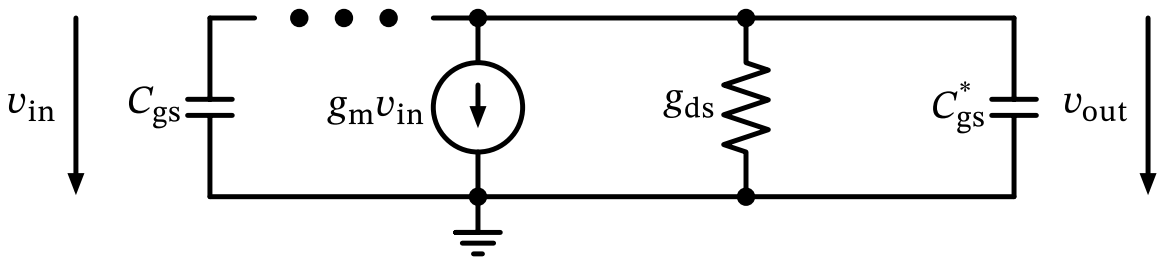


Figure 18: The MOSFET diode small-signal circuit for open-loop analysis.

We are now driving (with a voltage source) v_{in} and record the output voltage v_{out} to find the open-loop response $H_{ol}(s)$. By inspecting Figure 18 we see that

$$v_{out} = -g_m v_{in} \frac{1}{g_{ds} + sC_{gs}}.$$

The open-loop gain $H_{ol}(s)$ is thus

$$H_{ol}(s) = \frac{v_{out}}{v_{in}} = -\frac{g_m}{g_{ds} + sC_{gs}} = -\frac{g_m}{g_{ds}} \frac{1}{1 + s\frac{C_{gs}}{g_{ds}}}. \tag{7}$$

Inspecting Equation 7 we realize that:

1. The dc gain g_m/g_{ds} is the self-gain of the MOSFET, so $20\log_{10}(0.2 \cdot 10^{-3}/9.6 \cdot 10^{-6}) = 26.4\,\text{dB}$.
2. There is a pole at $\omega_p = -g_{ds}/C_{gs}$, which is at $9.6 \cdot 10^{-6}/(2\pi \cdot 1.4 \cdot 10^{-15}) = 1.1\,\text{GHz}$.

With this single pole location in $H_{ol}(s)$ this loop is perfectly stable at under all conditions (remember that a single pole results in a maximum phase shift of –90°).

The question is now how to simulate this open-loop gain, i.e., how to break the loop open in simulation? In general there are various methods, as we can use artificially large (ideal) inductors and capacitors to break loops open and still establish the correct dc operating points for the ac loop analysis. This is called **Rosenstark's method** (Rosenstark 1984). However, mimicking the correct loading can be an issue, and requires a lot of careful consideration.

There is an alternative method which breaks the loop open only by adding an ac voltage source in series (thus keeps the dc operating point intact), or injects current using an ac current source. Based on both measurements the open-loop gain can be calculated. This is called **Middlebrook's method** (Middlebrook 1975) and is based on double injection, and we will use it for our loop simulations. This method is detailed in Section 15.

There are several other methods like Tian's method (Tian et al. 2001), for example. A comprehensive overview can be found in (Neag et al. 2015) which describes ten different simulation-based loop gain analysis methods.

We now want to simulate the open-loop transfer function $H_{ol}(s)$ by using Middlebrook's method and confirm our analysis above.

Exercise: MOSFET Diode Loop Analysis

Please build a simulation testbench in Xschem to simulate the open-loop transfer function of the MOSFET diode. Confirm the dc gain and pole location as given by Equation 7.
If you are getting stuck you can look at this Xschem testbench, shown in Figure 19.

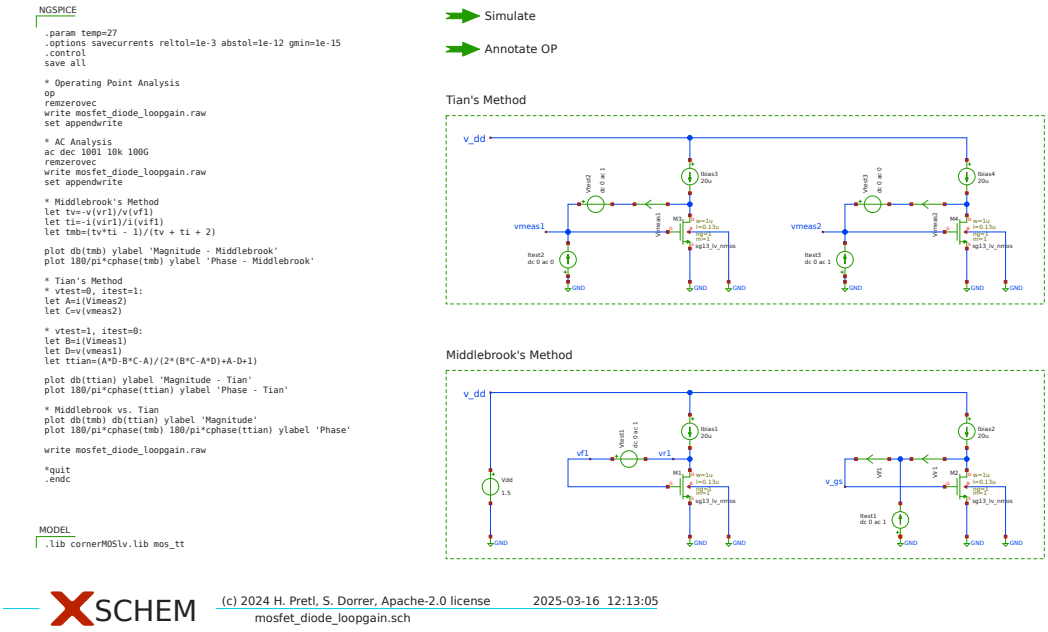


Figure 19: Testbench for MOSFET diode stability analysis.

From simulation we see that the open-loop gain is 24.9 dB at low frequencies, which matches quite well our prediction of 26.4 dB. In the Bode plot we see a low-pass with a −3 dB corner frequency of 1.4 GHz, which again is fairly close to our prediction of 1.1 GHz.

What About Large-Signal Stability?

Keep in mind that the above simulation only verifies the small-signal stability in one certain operating point. If we later look at the stability of an OTA it might be a good idea to verify the small-signal stability in different operating points.
Furthermore, one can apply a step response to the closed-loop system input and estimate the phase margin from the overshoot at the output (see *"Automatic Control"* lecture). One could also use a step-wise step response to simulate different operating points for a certain time (see *"Introduction in Integrated Circuit Design"* lecture).

4.5 MOSFET Diode Noise Calculation

As a final exercise on the MOSFET diode circuit we want to calculate the output noise when we consider V_{GS} the output reference voltage which is created when passing a bias current through the MOSFET diode. The bias current we will assume noiseless.

We are going to use the small-signal circuit shown in Figure 20.

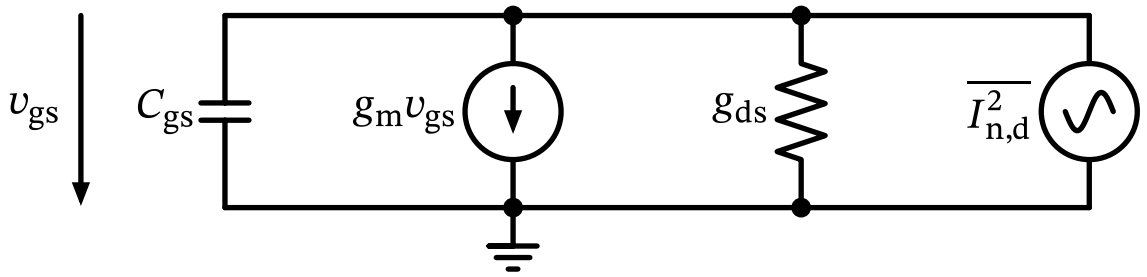


Figure 20: The MOSFET diode small-signal model with drain noise source.

As we have already calculated the small-signal diode impedance in Equation 5 we will use this result, and just note that the drain current noise of the MOSFET flows through this impedance. The noise voltage at v_{gs} is thus given as

$$\overline{V_{\mathrm{n}}^2} = |Z_{\mathrm{diode}}|^2 \overline{I_{\mathrm{n,d}}^2}.$$

The drain current noise of the MOSFET is given as (introduced in Section 2.1.2)

$$\overline{I_{\mathrm{n,d}}^2} = 4kT\gamma g_{\mathrm{m}}.$$

For low frequencies (ignoring g_{ds} and C_{gs}) we get

$$\overline{V_{\mathrm{n}}^2} = |Z_{\mathrm{diode}}|^2 \overline{I_{\mathrm{n,d}}^2} = \frac{1}{g_{\mathrm{m}}^2} 4kT\gamma g_{\mathrm{m}} = \frac{4kT\gamma}{g_{\mathrm{m}}}$$

which is the thermal noise of a resistor of value $1/g_{\mathrm{m}}$ enhanced by the factor γ.

We now calculate the full equation, and after a bit of algebra arrive at

$$\overline{V_{\mathrm{n}}^2}(f) = \frac{4kT\gamma g_{\mathrm{m}}}{(g_{\mathrm{m}} + g_{\mathrm{ds}})^2 + (2\pi f C_{\mathrm{gs}})^2}. \tag{8}$$

If we are interested in the PSD of the noise then Equation 8 gives us the result. If we are interested in the rms value (the total noise) we need to integrate this equation, using the following identity:

Useful Integral for Noise Calculations

$$\int_0^\infty \frac{a}{b^2 + c^2 f^2} df = \frac{\pi}{2} \frac{a}{b \cdot c} \tag{9}$$

Using the integral help in Equation 9, we can easily transform Equation 8 to

$$V_{\mathrm{n,rms}}^2 = \int_0^\infty \overline{V_{\mathrm{n}}^2}(f) df = \frac{kT\gamma g_{\mathrm{m}}}{(g_{\mathrm{m}} + g_{\mathrm{ds}}) C_{\mathrm{gs}}}. \tag{10}$$

The form of Equation 10 is the exact solution, but we gain additional insight if we assume that $g_{\mathrm{m}} + g_{\mathrm{ds}} \approx g_{\mathrm{m}}$ and then

$$V_{\mathrm{n,rms}}^2 = \frac{kT\gamma}{C_{\mathrm{gs}}}. \tag{11}$$

Inspecting Equation 11 we see our familiar kT/C noise multiplied by the factor γ!

Exercise: Total Output Noise of RC-Lowpass

If you have never calculated this before then you should work through the following: Calculate the total output noise of an RC-lowpass filter. Formulate the transfer function in the Laplace domain, and put the eqivalent resistor noise voltage source at the input, calculate the transfer to the output, and then integrate the output PSD (like we did for the MOSFET diode noise).
You fill find that the output noise is

$$V_{\mathrm{n,rms}}^2 = \frac{kT}{C}$$

which is independent of R! This is a surprising result, and is the well-known kT/C noise. Intuitively, we could argue that the noise increases with larger R, but at the same time, the bandwidth decreases and therefore R does not add additional noise. More detailed information and an intuitive explanation of kT/C noise can be found in (Sheikholeslami 2025).
Side note: The shortest derivation of this formula involves the equipartition theorem: Any system in thermal equilibrium with a reservoir of temperature T has a fluctuation energy of $kT/2$ per degree of freedom. This RC system has one degree of freedom in the voltage on the capacitor, and the stored energy in the capacitor is $CV_{\mathrm{rms}}^2/2$. Equating both energies we find that $V_{\mathrm{rms}}^2 = kT/C$ (Sarpeshkar, Delbruck, and Mead 1993).
To calculate the total output noise of a generalized passive network Bode's noise theorem is quite practical (see Section 16.2).

Calculating the rms noise voltage for our MOSFET diode we get

$$\sqrt{V_{\mathrm{n,rms}}^2} = \sqrt{1.38 \cdot 10^{-23} \cdot 300 \cdot 0.84/1.4 \cdot 10^{-15}} = 1.58\,\mathrm{mV},$$

which is a sizeable value! Think about it, can this rms noise voltage be measured with an oscilloscope? If not, why? We run circuits in this technology at $V_{DD} = 1.5\,V$, which leaves us with a signal swing of ca. $1.1\,V_{pp}$ (single-ended), resulting in a dynamic range in this case of $20\log_{10}(0.39/1.58 \cdot 10^{-3}) \approx 48\,dB$ assuming a sinusoidal signal. In order to get a feeling which dynamic range is "good", we can calculate the required dynamic range of a 16-bit audio ADC to be $6.02 \cdot 16 + 1.76\,dB = 98.08\,dB \approx 100\,dB$. This calculation should make clear that, for example, the correct sizing of the sample&hold capacitor is crucial for low rms noise voltage.

! Be Careful with Parasitic Capacitances in IC Design

In general, in integrated circuit design, we often have only small parasitic capacitances on many nodes that could sum up to unwanted high noise according to Equation 11. If one wants to lower the noise an increased capacitance could limit the bandwidth (and thus the kT/C noise).

! Large Bandwidth and Noise

Remember: Large bandwidth circuits integrate noise over a wide bandwidth resulting in (potentially) considerable rms noise. The way to lower the total noise is to lower the PSD of the noise contributions, which usually requires increased power consumption. So in a nutshell:
Large bandwidth plus small noise equals large power consumption.

Exercise: MOSFET Diode Noise

Please build a simulation testbench in Xschem to simulate the noise performance of the MOSFET diode, and confirm the rms noise value that we just calculated. Look at the rms value and the PSD of the noise, and play around with the integration limits. What is the effect? Can you see the flicker noise in the PSD? How much is its contribution to the rms noise? What is the value of f_{co}, and does it correspond to the above calculated one?
If you are getting stuck you can look at this Xschem testbench, shown in Figure 21.

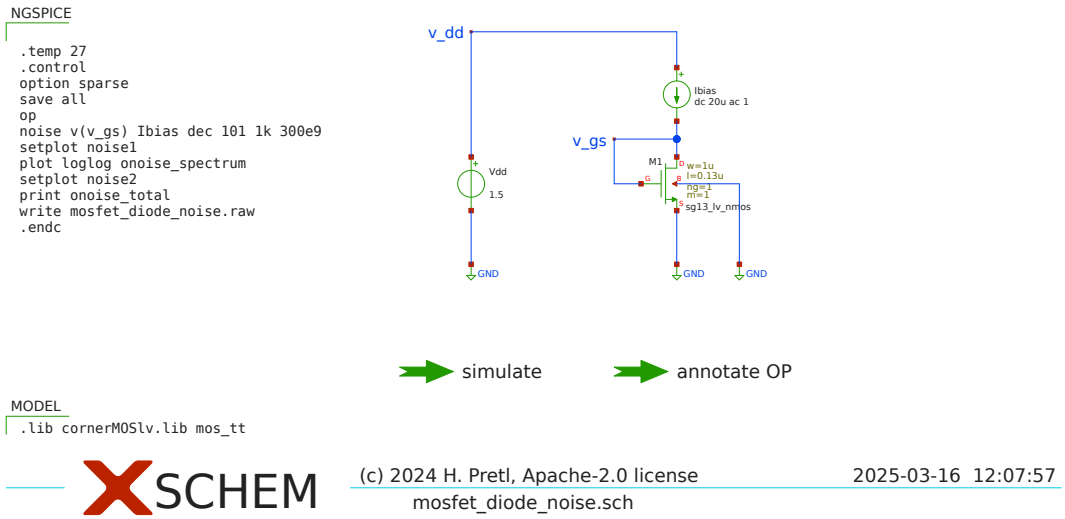


Figure 21: Testbench for MOSFET diode noise analysis.

4.6 Conclusion

In this section we investigated the simple MOSFET-diode circuit. We learned important skills like how to derive a small-signal model and how to calculate important features like noise and open-loop gain for stability analysis. We introduced Middlebrook's method to have a mechanism to open up loops in simulation (and calculation) without disturbing dc operating points or introduce errors by changing loading conditions.

If you feel that you have not yet mastered these topics or are uncertain in the operation of Xschem or ngspice, please go back to the beginning of the section and read through the theory and redo the exercises.

5 Common-Source Amplifier

We now want to step up our game, and use more components to design something useful. We will use a basic circuit structure, namely a single-ended common-source amplifier. The structure of this circuit, using a resistor as a load, is shown in Figure 22.

The function of this circuit is as follows: Assuming the MOSFET M_1 is kept in saturation, then a small-signal voltage v_{in} applied at the gate is converted into a drain current i_{d} by the MOSFET's transconductance g_{m}. Then, this current is converted into a voltage again in the resistor R_1. Ultimately, we have a dc voltage gain A_{v} of

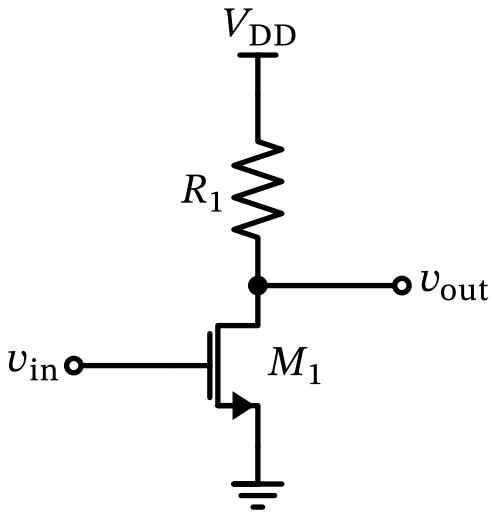


Figure 22: A MOSFET common-source amplifier with resistive load.

$$A_v = \frac{v_{out}}{v_{in}} \approx -\frac{g_m v_{in} R_1}{v_{in}} = -g_m R_1.$$

As explained above, it is a good approach to see electronic circuit components as

- voltage-to-current (MOSFET as common-source or common-gate; resistor),
- current-to-voltage (resistor),
- current-to-current (MOSFET as common-gate), and
- voltage-to-voltage (MOSFET as common-drain)

conversions for better understanding.

5.1 Sense Amplifier Driving 50 Ohm Matched Load

Let us now size and design an exemplary implementation of this amplifier (of course using the g_m/I_D method). In order to have useful real-life specifications, we want to build an amplifier which can be used to sense an on-chip voltage and drive off-chip measurement equipment. Often, this equipment has an input impedance of 50Ω, and we want to have an impedance-matched output. The voltage gain shall be set to 1 (essentially, we want to sense a voltage and drive the measurement equipment).

The resulting circuit is shown in Figure 23. As the load is usually ground-referred, and we want to avoid a dc-block at the output, we use a PMOS amplifier stage (compare with Figure 22).

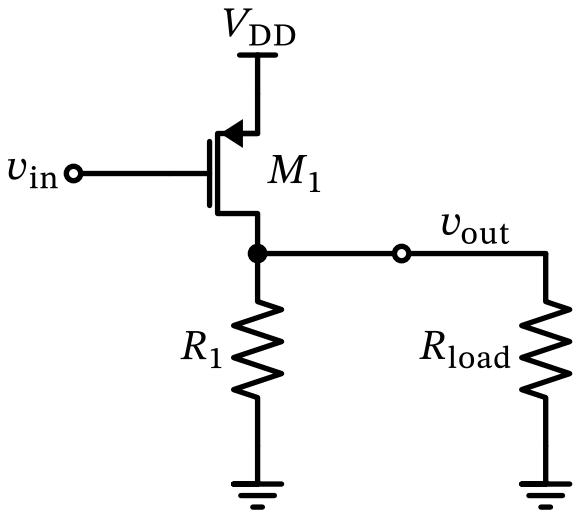


Figure 23: A MOSFET common-source amplifier with 50 Ohm load.

Exercise: PMOS-Based Measurement Amplifier

Please think about why exactly we want this measurement amplifier be based on a PMOS instead of an NMOS.

The power matching requirement at the output mandates that $R_1 = R_{\text{load}}$, so $R_1' = R_{\text{load}} \parallel R_1 = 25\,\Omega$. The voltage gain requirement of $|A_\text{v}| = 1$ results in $g_\text{m} = 1/R_1' = 40\,\text{mS}$.

We now need to find W and L of M_1 and calculate the required bias current I_D. We also need to find the proper V_GS to set this current. As usual, we use a Jupyter notebook to calculate these values. Since we require modest speed of this buffer we use a $g_\text{m}/I_\text{D} = 8$ and set $L = 0.13\,\mu\text{m}$. The notebook is available here. The resulting circuit including all component values is shown at the end of the notebook (the red input capacitor shows the C_gg of the MOSFET).

Important 1: MOSFET Parameters NG and M

When sizing the MOSFET for this example we found that we need a fairly large W, resulting in a MOSFET aspect ratio of $W/L \gg 1000$. When constructing an integrated circuit out of individual MOSFET we strive for an overall IC dimension that is roughly quadratic. For MOSFET with large aspect ratios we need to get them into a comfortable shape.
In order to achieve this, we construct the MOSFET out of smaller pieces, and the size of this pieces (called "gate fingers") are controlled by the parameter `ng`. These MOSFET gate fingers all have the same L, but their width is $W_\text{finger} = W/\text{ng}$. All this individual smaller MOSFET are connected in parallel.
In order to increase the MOSFET model accuracy, often the maximum value of W_finger is limited. In the case of SG13G2 $W_\text{finger} \leq 10\,\mu\text{m}$.

In order to construct even larger MOSFET, we can connect multiple MOSFET in parallel. We can do this in the circuit editor by placing and connected these MOSFET; but since this is often used there is a more convenient way: By using the parameter `m` ("multiplier" or "multiplicity") we instantiate m MOSFET connected in parallel.
When to use `ng` and when to use `m`? The use of `ng` results in a more compact IC layout, and is thus generally preferable. Only in certain instances (e.g., when using a really large W) `m` should be used. Further, the thoughtful use of `ng` **allows to construct all the NMOS and PMOS of a circuit out of the same gate finger elements**. This will result in a very compact layout!

Exercise: Measurement Amplifier Simulation

Please go through the sizing notebook of the measurement amplifier and double-check the calculations. Do you agree that the calculations are correct?
Once you agree with the circuit sizing please build an Xschem simulation testbench where you simulate the small-signal voltage gain A_v of this measurement amplifier if it is driven with an ideal voltage source. Keep in mind that the maximum MOSFET finger width is 10µm in this technology, so you need to set the parameter `ng` accordingly (see Important 1).

- What is the dc gain of this amplifier when loaded with 50Ω?
- The dc gain is likely not exactly 0dB. Why is this so?
- Increase the width W of the PMOS until the gain is correct. What is the W that you had to set, and how much is I_D now?
- What is the bandwidth (i.e., the -3dB corner frequency) of the output voltage, when the voltage source has a source resistance of 1kΩ?

If you get stuck, here is the solution to this exercise, and it is also shown in Figure 24.

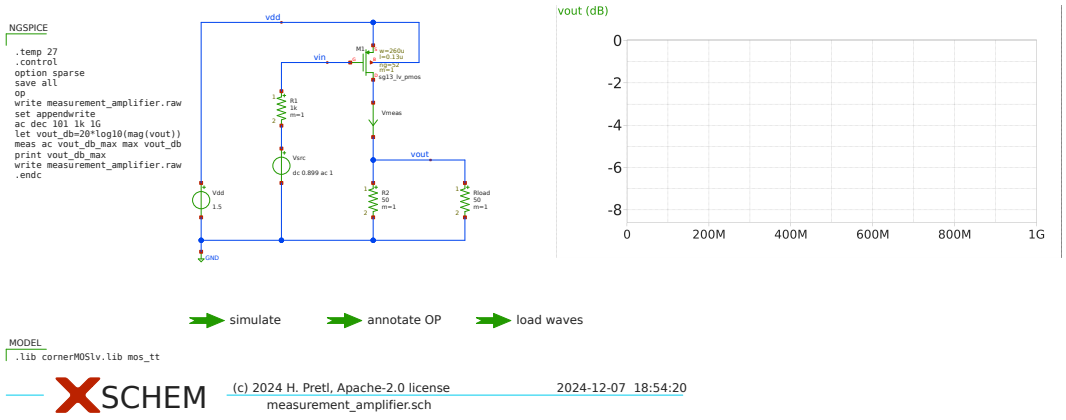


Figure 24: Simulation schematic of the common-source measurement amplifier.

By now we have designed a measurement amplifier based on a common-source stage. One problem with this stage is the relatively large input capacitance C_{GG} of approx. 0.3pF, which loads the input source. Another issue even more severe is that the fact that the bias point in this circuit is set by the dc voltage level at the input. In general, we want a setup where the bias points of the circuit are largely independent of the dc input voltages. This is why in integrated circuit design we often design **differential circuits** where the input and output signals are given by the differential voltages, and are largely independent from the common-mode voltages. This is usually an advantage.

6 Current Mirror

In this section we will look into a fundamental building block which is often used in integrated circuit design, the **current mirror** (R. Widlar 1965). A diagram is shown in Figure 25 with one MOSFET diode converting the incoming bias current into a voltage $V_{\mathrm{GS}} = f(I_{\mathrm{D1}}) = f(I_{\mathrm{bias}})$, and two output MOSFETs working as current sources, which are biased from the diode. By properly selecting all W and L the input current can be scaled, and multiple copies can be created at once. Shown in the figure are two output currents I_{out1} and I_{out2}, but any number of parallel branches can be realized (note that this is true for MOSFET as no gate current flows; for the case of BJTs, the base current flowing into the various transistors is subtracted from I_{bias}, so usually a compensation circuit is added).

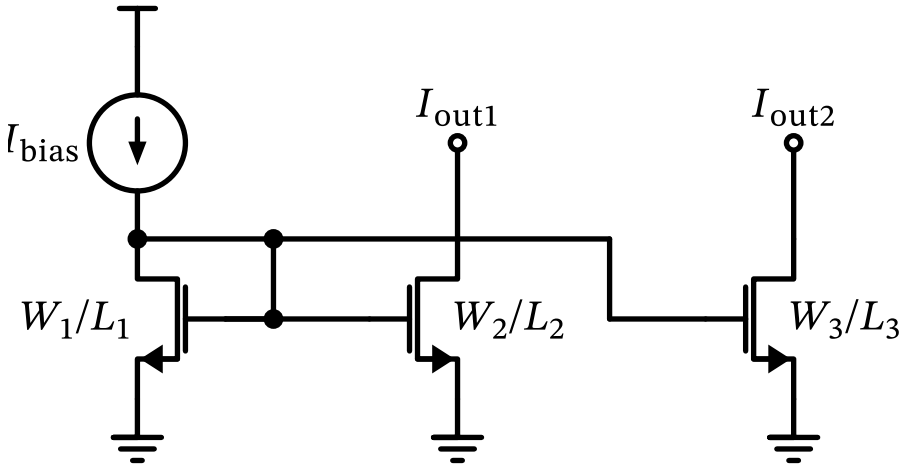


Figure 25: A current mirror with two output branches.

Neglecting the impact of g_{ds1} and g_{ds2}, the output current I_{out1} is then given by

$$I_{out1} \approx I_{bias} \frac{W_2}{L_2} \frac{L_1}{W_1}$$

and the output current I_{out2} is given by

$$I_{out2} \approx I_{bias} \frac{W_3}{L_3} \frac{L_1}{W_1}.$$

For good matching care has to be taken that the MOSFET widths and lengths are constructed out of **unit elements** of identical size, where an appropriate amount of these single units are then arranged in series or parallel configuration to arrive at the target W and L (remember MOSFET parameters `ng` and `m`, see Important 1).

As we know from earlier investigations of the MOSFET performance in Section 3 the drain current of a MOSFET is a function of V_{GS} and V_{DS}. As long as the MOSFET stays in saturation (i.e., $V_{DS} > V_{ds,dsat}$) the drain current is just a mild function of V_{DS} (essentially the effect of g_{ds}, which is the output conductance of the MOSFET). A fundamental flaw/limitation of the basic current mirror shown in Figure 25 is the mismatch of the V_{DS} of the MOSFETs. The input-side diode has $V_{GS} = V_{DS1}$, whereas the output current sources have a $V_{DS2,3}$ depending on the connected circuitry. Improved current mirrors exist (fixing this flaw), however, when a current mirror is required with mediocre performance requirements this structure can be used for its simplicity.

Exercise: Current Mirror

Please construct a current mirror based on the MOSFET-diode which we sized in Section 4. The input current $I_{bias} = 20\,\mu A$, and we want three output currents of size $10\,\mu A$, $20\,\mu A$, and $40\,\mu A$.
Sweep the output voltage of all three current branches and see over which voltage

range an acceptable current is created. For which output voltage range is the current departing from its ideal value, and why?
You see that the slope of the output current is quite bad, as g_{ds} is too large. We can improve this by changing the length to $L = 5\,\mu$m (for motivation, please look at the graphs in Section 3). In addition, for a current mirror we are not interested in a high g_m/I_D value, so we can use $g_m/I_D = 5$ in this case. Please size the current mirror MOSFETs accordingly (please round the W to half micron, to keep sizes a bit more practical). Compare this result to the previous one, what changed?
In case you get stuck, here are Xschem schematics for the original and the improved current mirrors.

7 Differential Pair

Like the current mirror in Section 6 the **differential pair** is an ubiquitous building block often used in integrated circuit design (Blumlein 1937). The fundamental structure is given in Figure 26.

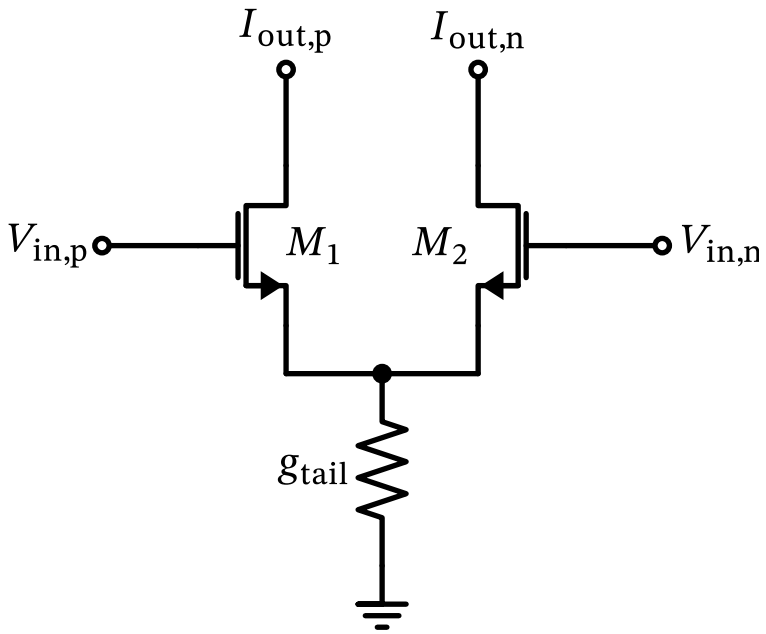


Figure 26: A differential pair.

In order to understand its operation it is instructive to separate the input condition into (1) a purely differential voltage, and (2) a common-mode voltage, and see what the impact on the output currents is.

7.1 Differential Operation of the Diffpair

For a differential mode of operation we assume that the input common mode voltage is constant, i.e., $(V_{\text{in,p}} + V_{\text{in,n}})/2 = V_{\text{CM}}$. The differential input voltage $v_{\text{in}} = V_{\text{in,p}} - V_{\text{in,n}}$, so that

$$V_{\text{in,p}} = V_{\text{CM}} + \frac{v_{\text{in}}}{2}$$

and

$$V_{\text{in,n}} = V_{\text{CM}} - \frac{v_{\text{in}}}{2}.$$

For a small-signal differential drive the potential at the tail point stays constant and we can treat it as a virtual ground. The output current on each side is then given by (neglecting g_{ds} and g_{mb} of M_1 and M_2)

$$i_{\text{out,p}} = g_{\text{m1}} \left(\frac{v_{\text{in}}}{2} \right)$$

and

$$i_{\text{out,n}} = g_{\text{m2}} \left(-\frac{v_{\text{in}}}{2} \right).$$

Usually we assume symmetry in the differential pair, so $g_{\text{m1}} = g_{\text{m2}} = g_{\text{m}}$. The differential output current i_{out} is then given by

$$i_{\text{out}} = i_{\text{out,p}} - i_{\text{out,n}} = g_{\text{m}} v_{\text{in}} \tag{12}$$

We see in Equation 12 that the differential output current is simply the differential input voltage multiplied by the g_{m} of the individual transistor. We also note that the bottom conductance g_{tail} plays no role for the small-signal differential operation.

7.2 Common-Mode Operation of the Diffpair

Usually, the source conductance g_{tail} is realized by a current source and ideally should be $g_{\text{tail}} = 0$. If this is the case, then the output currents are not a function of the common-mode input voltage (I_{tail} is set by the tail current source), and

$$I_{\text{out,p}} = I_{\text{out,n}} = \frac{I_{\text{tail}}}{2}.$$

However, if we assume a realistic tail current source then $g_{\text{tail}} > 0$. For analysis we can simply look at a half circuit since the circuit operation is symmetric. In order to simplify the analysis a bit we remove all capacitors from the MOSFET small-signal model and set $g_{\text{ds}} = g_{\text{mb}} = 0$. We then arrive at the small-signal equivalent circuit shown in Figure 27 (note that we set $v_{\text{in,p}} = v_{\text{in,n}} = v_{\text{in}}$ and $i_{\text{out,p}} = i_{\text{out,n}} = i_{\text{out}}$ under symmetry considerations).

Formulating KVL for the input-side loop we get

$$v_{\text{in}} = v_{\text{gs}} + \frac{i_{\text{d}}}{g_{\text{tail}}}.$$

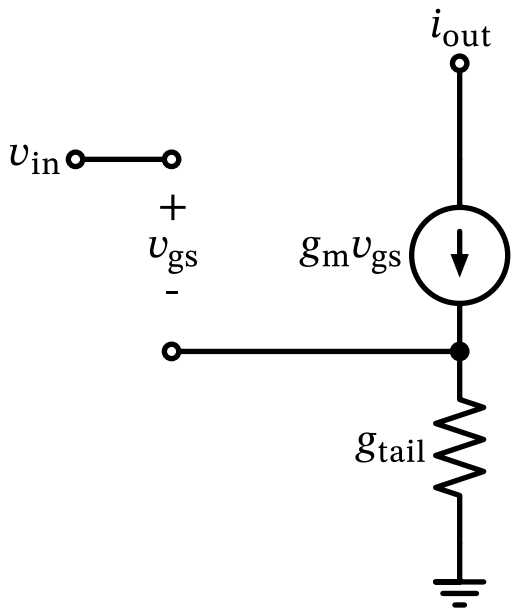


Figure 27: Small-signal model of the differential pair half-circuit in common-mode operation.

With $i_{out} = i_d = g_m v_{gs}$ we arrive at

$$i_{out} = \frac{g_m g_{tail}}{g_m + g_{tail}} v_{in} \tag{13}$$

Interpreting Equation 13 we can distinguish the following extreme cases:

1. If $g_{tail} = 0$ (ideal tail current source) then $i_{out} = 0$, the common-mode voltage variation from the input is suppressed and does not show up at the common-mode output current (which is constant due to the ideal tail current source). This is usually the case that we want to achieve.
2. If $g_{tail} \to \infty$ then $i_{out} = g_m v_{in}$, which means the output current is a function of the MOSFET g_m. If everything is perfectly matched, then the differential output current is zero, but the common-mode output current changes according to the common-mode input voltage. In special cases this can be a wanted behavior, this configuration is called a "pseudo-differential pair."

Note that the result of Equation 13 is also valid for the general case of a degenerated common-source transistor stage (see Figure 28). The effective transconductange g'_m is given by ($R_{degen} = g_{tail}^{-1}$)

$$g'_m = \frac{i_{out}}{v_{in}} = \frac{g_m g_{tail}}{g_m + g_{tail}} = \frac{1}{g_m^{-1} + R_{degen}}.$$

When no degeneration resistor is used, then $g'_m = g_m$. If a degeneration of $R_{degen} \gg g_m^{-1}$ is used then $g'_m = 1/R_{degen}$. This result is worth memorizing.

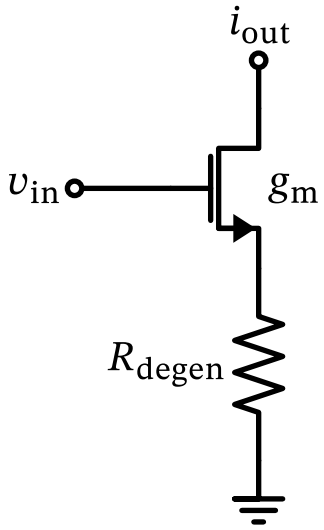


Figure 28: A MOSFET common-source amplifier with resistive degeneration.

8 A Basic 5-Transistor OTA

Suited with the insights of basic transistor operation (Section 2 and Section 3) and the working knowledge of the current mirror (Section 4 and Section 6) as well as the differential pair (Section 7) we can now start to design our first real circuit. A fundamental (simple) circuit that is often used for basic tasks is the 5-transistor operational transconductance amplifier (OTA). A circuit diagram of this 5T-OTA is shown in Figure 29.

> ⚠ Refresh MOSFET Basic Circuits
>
> While we repeat the basics of elementary MOSFET amplifier stages (like common-source stage, common-gate stage, and current mirror) in this course material, the following compendium (Murmann 2013) is recommended for review. It is freely available at https://github.com/bmurmann/Book-on-MOS-stages.
> In addition, we can highly recommend these references Razavi (2017) for further study.

The operation is as follows: $M_{1,2}$ form a differential pair which is biased by the current source M_5. $M_{5,6}$ form a current mirror, thus the input bias current I_{bias} sets the bias current in the OTA. The differential pair $M_{1,2}$ is loaded by the current mirror $M_{3,4}$ which mirrors the drain current of M_1 to the right side. Here, the currents from M_4 and M_2 are summed, and together with the conductance effective at the output node a voltage builds up.

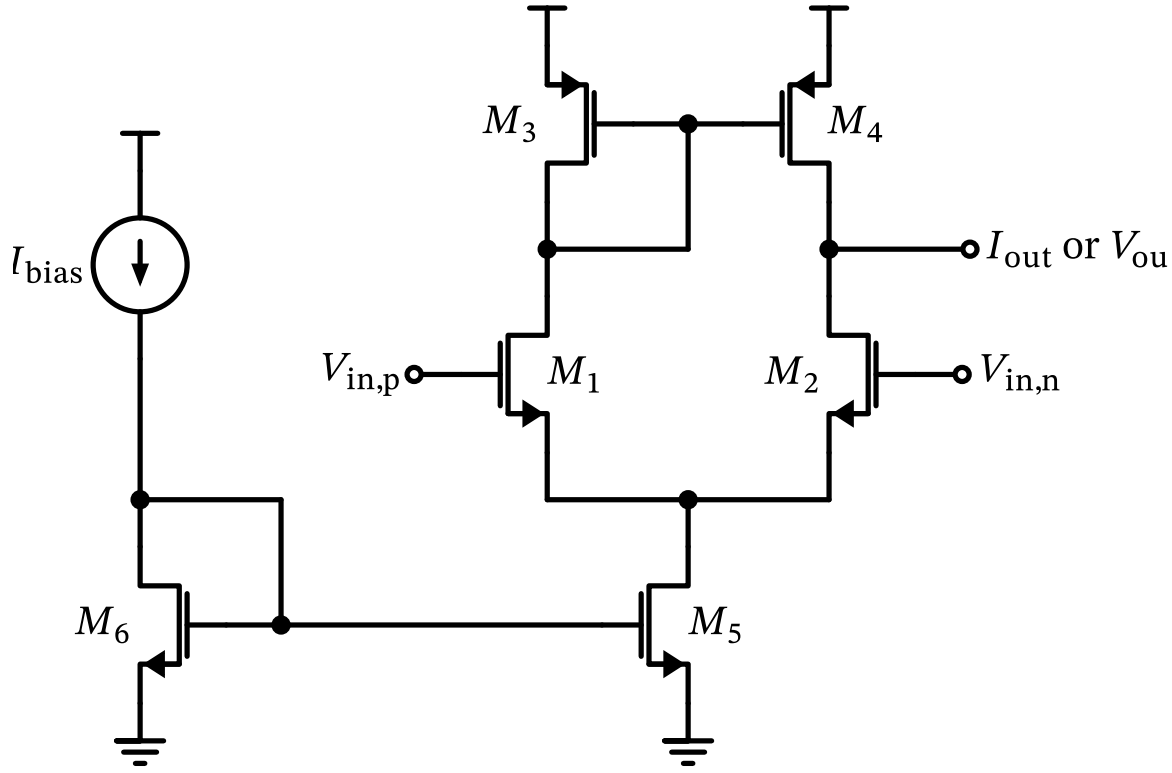


Figure 29: The 5-transistor OTA.

i Operational Amplifier (op-amp, OPA) vs. Operational Transconductance Amplifier (OTA)

An operational amplifier is a universal electronic building block characterized by (in the ideal case):

- infinite input impedance at the input ports ($R_{\text{in}} \to \infty$)
- zero output resistance, i.e., a voltage output ($R_{\text{out}} = 0$)
- infinite voltage gain ($A_{\text{v}} = V_{\text{out}}/V_{\text{in}} \to \infty$)

An operational transconductance amplifier is a building block characterized by (again in the ideal case):

- infinite input impedance at the input ports ($R_{\text{in}} \to \infty$)
- infinite output resistance, i.e., a current output ($R_{\text{out}} \to \infty$)
- infinite transconductance ($G = I_{\text{out}}/V_{\text{in}} \to \infty$)

In integrated circuits, we very often load an OPA/OTA high-ohmic, i.e., with a capacitive load. Hence, an OTA can be used to create a voltage-mode amplifier with high gain, approaching the properties of the OPA. If an OTA is used to drive a low-ohmic load, the voltage gain will be low, and we have to use this block as a transconductance amplifier. Since the output changes behavior depending on high- or low-ohmic loading, we label the output in Figure 29 accordingly.
Why then implement an OTA instead of an OPA? Usually, an OTA is a simpler structure than an OPA. As a general rule, the simplest circuit that can do a job is usually the best choice.

We note that $M_{1,2}$ and $M_{3,4}$ need to be symmetric, thus will have the same W and L dimensioning. $M_{5,6}$ we scale accordingly to set the correct bias current in the OTA.

8.1 Voltage Buffer with OTA

In order to design an OTA we need an application, and from this we need to derive the circuit specifications. We want to use this OTA to realize a voltage buffer which lightly loads an input voltage source and can drive a large capacitive load. Such a configuration is often used to, e.g., buffer a reference voltage that is needed (and thus loaded) by another circuit. The block diagram of this configuration is shown in Figure 30.

If the voltage gain of the OTA in Figure 30 is high, then $V_{\text{out}} \approx V_{\text{in}}$. We now want to design an OTA for this application fitting the following specification values (see Table 2). These values are rather typical of what could be expected for such a buffer design.

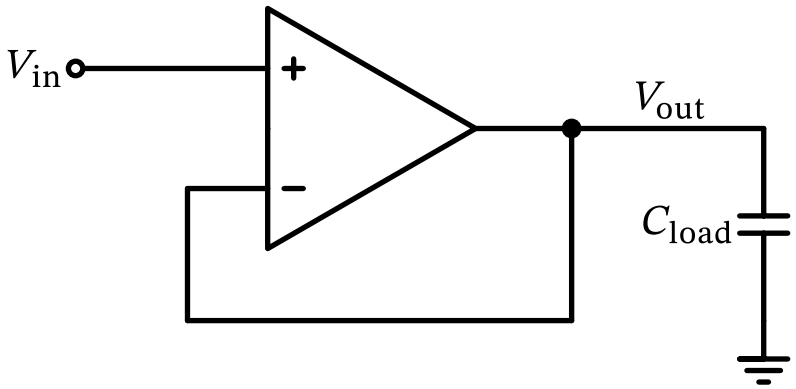


Figure 30: A voltage buffer (based on OTA) driving a capacitive load.

Table 2: Voltage buffer specification

Specification	Value	Unit
Supply voltage	$1.45 < \underline{1.5} < 1.55$	V
Temperature range (industrial)	$-40 < \underline{27} < 125$	degC
Load capacitance C_{load}	50	fF
Input voltage range (for buffering 2/3 bandgap voltage)	$0.7 < \underline{0.8} < 0.9$	V
Signal bandwidth (3dB)	> 10	MHz
Output voltage error	< 3	%
Total output noise (rms)	< 1	mV_{rms}
Supply current (as low as possible)	< 10	µA
Stability	stable for rated C_{load}	
Turn-on time (settled to with 1%)	< 10	µs
Externally provided bias current (nominal)	20	µA

8.2 Large-Signal Analysis of the OTA

The first step when receiving a design task is to look at the specifications, and see whether they make sense. The detailed performance of the design will be the result of the circuit simulation, but before we step into sizing we need to do a few simple calculations to (a) do back-of-the-envelope gauging if the specification makes sense, and (b) use the derived analytical equations as guide for the sizing procedure.

In terms of large-signal operation, we will now check whether the input and output voltage range, as well as the settling time can be roughly met. Since we do not know yet the resulting V_{GS} of the transistors we use 0.6 V as an initial guess. If we run into issues with that guess we know how to later steer the sizing procedure.

- When the input is at its maximum of 0.9 V, we see that we need to keep M_1 in saturation. We can calculate that $V_{\text{DS1}} = V_{\text{DD}} - |V_{\text{GS3}}| + V_{\text{GS1}} - V_{\text{in}} = 1.45 - 0.6 +$

$0.6 - 0.9 = 0.55\,\mathrm{V}$, which leaves enough margin.

- When the input is at its minimum of 0.7 V, we see that the V_{DS5} of M_5 is calculated as $V_{\mathrm{DS5}} = V_{\mathrm{in}} - V_{\mathrm{GS1}} = 0.7 - 0.6 = 0.1\,\mathrm{V}$, so this leaves little margin, but we can make V_{GS1} smaller, so it should work out.
- For the output voltage, when the output voltage is on the high side, it leaves $|V_{\mathrm{DS4}}| = V_{\mathrm{DD}} - V_{\mathrm{out}} = 1.45 - 0.9 = 0.55\,\mathrm{V}$, which is enough margin.

In summary, we think that we can make an NMOS-input OTA like the one in Figure 29 work for the required supply and input- and output-voltages. If this would not work out, we need to look for further options, like a PMOS-input OTA, or a NMOS/PMOS-input OTA.

Another large-signal specification item that we can quickly check is the settling time. Under slewing conditions, the complete bias current in the OTA is steered towards the output (try to understand why this is the case), so when the output capacitor is fully discharged, and we assume just a linear ramp due to constant-current charging of the output capacitor, the settling time is

$$T_{\mathrm{slew}} \approx \frac{C_{\mathrm{load}} V_{\mathrm{out}}}{I_{\mathrm{tail}}} = \frac{50 \cdot 10^{-15} \cdot 1.3}{10 \cdot 10^{-6}} = 6.5\,\mathrm{ns}$$

so this leaves plenty of margin for additional slow-signal settling due to the limited bandwidth, as well as reducing the supply current.

The small-signal settling (assuming one pole at the bandwidth corner frequency) leads to an approximate settling time (1% error corresponds to $\approx 5\tau$) of

$$T_{\mathrm{settle}} \approx \frac{5}{2\pi f_{\mathrm{c}}} = \frac{5}{2\pi \cdot 1 \cdot 10^{6}} = 0.8\,\mu\mathrm{s}.$$

which also checks out.

8.3 Small-Signal Analysis of the OTA

In order to size the OTA components we need to derive how the MOSFET parameters define the performance. The important small-signal metrics are

- dc voltage gain A_0
- gain-bandwidth product (GBW)
- output noise

The specification for GBW is directly given in Table 2, while the dc gain we have to calculate from the voltage accuracy specification. For a voltage follower in the configuration shown in Figure 30 the voltage gain is given by

$$\frac{V_{\mathrm{out}}}{V_{\mathrm{in}}} = \frac{A_0}{1 + A_0}. \tag{14}$$

So in order to reach an output voltage accuracy of at least 3% we need a dc gain of $A_0 > 30.2\,\mathrm{dB}$. To allow for process and temperature variation we need to add a bit of extra gain as margin.

! Small-Signal vs. Large-Signal Operation

In order to get the correct dc voltage per the specification we require the large-signal gain calculated with Equation 14. However, calculating the large-signal gain of a circuit is quite involved (usually mandating the use of a large-signal nonlinear model for the used components), so we typically resort to do a simpler small-signal calculation instead, like in Section 8.3. We deliberately introduce this error, but we should not get confused about the difference between large- and small-signal operation!

8.3.1 OTA Small-Signal Transfer Function

In order to derive the governing small-signal equations for the OTA we will make a few simplifications:

- We will set $g_{\mathrm{mb}} = 0$ for all MOSFETs.
- We will further set $C_{\mathrm{gd}} = 0$ for all MOSFETs except for M_4 where we expect a Miller effect on this capacitor, and we could add its effect by increasing the capacitance at the gate node of $M_{3,4}$ (for background please see Section 16.1). However, as this does not create a dominant pole in this circuit, we consider this a minor effect (see Equation 17). Thus, only C_{gs34} is considered at the gate node of the current mirror load.
- We assume $g_{\mathrm{m}} \gg g_{\mathrm{ds}}$, so we set $g_{\mathrm{ds1}} = g_{\mathrm{ds3}} = 0$.
- The drain capacitance of M_2 and M_4, as well as the gate capacitance of M_2 we can add to the load capacitance C_{load}. Note that C_{gs2} can be added because of the feedback connection between the inverting input and the output. However, this is not shown in the small-signal equivalent circuits below, because we are interested in the open-loop transfer function.

The resulting small-signal equivalent circuit is shown in Figure 31.

⚠ Refresh MOSFET Small-Signal Model

Please review the MOSFET small-signal equivalent model in Figure 5 at this point. For the PMOS just flip the model upside-down.

We can further simplify the output side by recognizing that the impedance (when looking from the output down) is g_{ds2} in series with $g_{\mathrm{ds5}} + g_{\mathrm{m12}}$. This is a valid simplification as we treat M_1 as a common-gate stage when looking from the output, and since it is loaded by a low impedance of g_{m34}^{-1}, we can approximate the impedance looking into the source of M_1 with g_{m12}^{-1}. With the approximation that $g_{\mathrm{m}} \gg g_{\mathrm{ds}}$ the parallel connection of g_{m12} and g_{ds5} is dominated by g_{m12} ($g_{\mathrm{m12}} + g_{\mathrm{ds5}} \approx g_{\mathrm{m12}}$) and its series connection with g_{ds2} ($g_{\mathrm{m12}}^{-1} + g_{\mathrm{ds2}}^{-1} \approx g_{\mathrm{ds2}}^{-1}$), resulting in the fact that we can ground the lower end of g_{ds2}. You should probably stop here and read this paragraph again, slowly.

Therefore, we can move $g_{\mathrm{ds2}} + g_{\mathrm{ds4}}$ in parallel to C_{load}. Further, assuming a differential drive with a virtual ground at the tailpoint we can remove g_{ds5}. The current source $g_{\mathrm{m34}} v_{\mathrm{gs34}}$ is

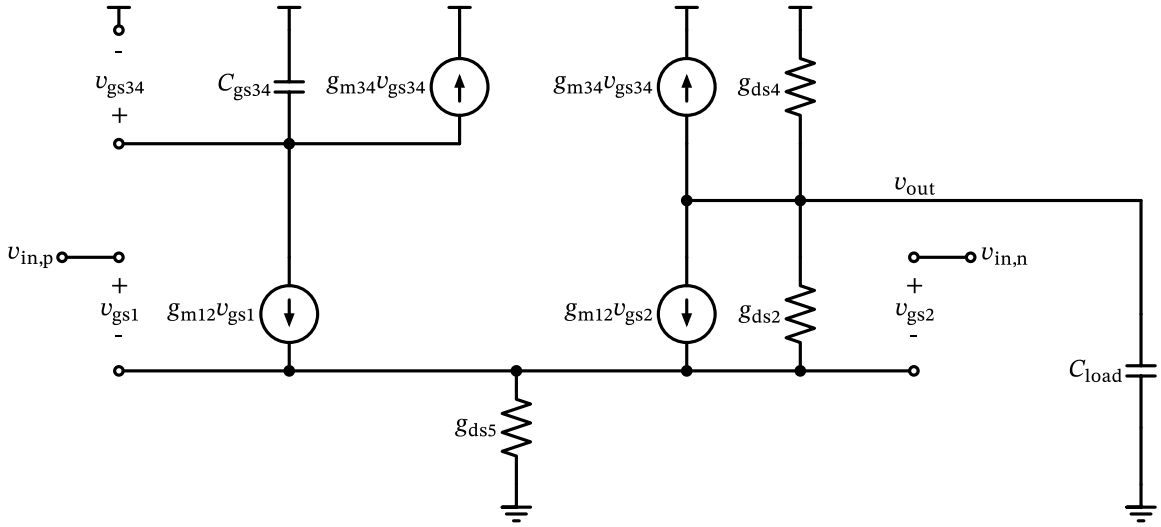


Figure 31: 5-transistor OTA small-signal model.

replaced with the equivalent conductance g_{m34}. All the previous steps result in the further simplified equivalent circuit shown in Figure 32.

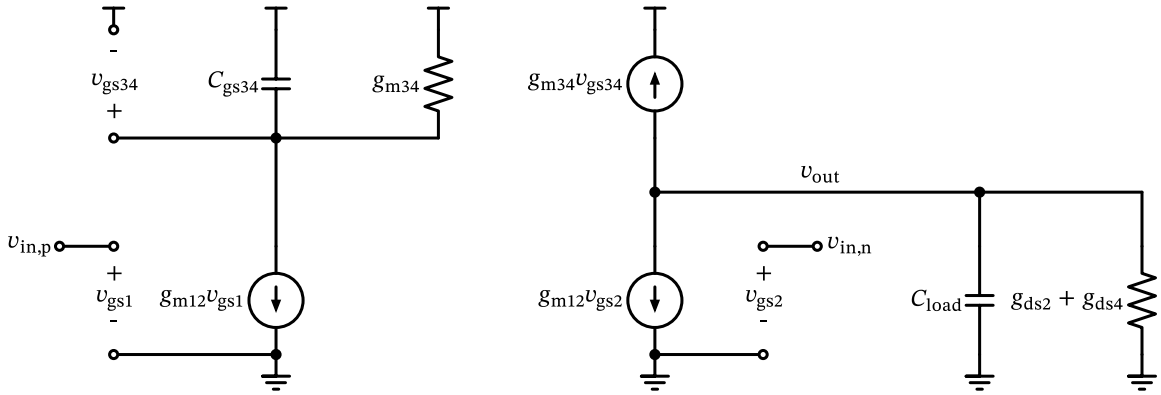


Figure 32: 5-transistor OTA small-signal model with further simplifications.

In the simplified circuit model in Figure 32 we can see that we have two poles in the circuit, one at the gate note of $M_{3,4}$, and one at the output. Realizing that $v_{in,p} = v_{in}/2$ and $v_{in,n} = -v_{in}/2$ we can formulate KCL at the output node to

$$-g_{m34}V_{gs34} - \left(-g_{m12}\frac{V_{in}}{2}\right) - V_{out}(g_{ds2} + g_{ds4} + sC_{load}) = 0. \tag{15}$$

We further realize that

$$V_{gs34} = -g_{m12}\frac{V_{in}}{2}\frac{1}{g_{m34} + sC_{gs34}}. \tag{16}$$

By combining Equation 15 and Equation 16 and after a bit of algebraic manipulation we

arrive at

$$A(s) = \frac{V_{\text{out}}(s)}{V_{\text{in}}(s)} = \frac{g_{\text{m12}}}{2} \frac{2g_{\text{m34}} + sC_{\text{gs34}}}{(g_{\text{m34}} + sC_{\text{gs34}})(g_{\text{ds2}} + g_{\text{ds4}} + sC_{\text{load}})}. \tag{17}$$

When we now inspect Equation 17 we can see that for low frequencies the (open-loop) gain is

$$A(s \to 0) = A_0 = \frac{g_{\text{m12}}}{g_{\text{ds2}} + g_{\text{ds4}}} \tag{18}$$

which is plausible, and confirms the requirement of a high impedance at the output node. For very large frequencies we get

$$A(s \gg) = \frac{g_{\text{m12}}}{2sC_{\text{load}}} \tag{19}$$

which is essentially the behavior of an integrator, and we can use Equation 19 to calculate the frequency where the gain drops to $|A(s_{\text{ug}})| = 1 = g_{\text{m12}}/|2j\omega_{\text{ug}}C_{\text{load}}|$:

$$f_{\text{ug}} = \frac{g_{\text{m12}}}{4\pi C_{\text{load}}}$$

When looking at Equation 17 we see that we have a dominant pole at s_{p} and a pole-zero doublet with $s_{\text{pd}}/s_{\text{zd}}$:

$$s_{\text{p}} = -\frac{g_{\text{ds2}} + g_{\text{ds4}}}{C_{\text{load}}} \tag{20}$$

$$s_{\text{pd}} = -\frac{g_{\text{m34}}}{C_{\text{gs34}}} \tag{21}$$

$$s_{\text{zd}} = -\frac{2g_{\text{m34}}}{C_{\text{gs34}}} \tag{22}$$

> **Why a Pole-Zero Doublet?**
>
> Looking at Equation 21 and Equation 22 we see that this pair is intimately linked by the same parameters and can only move together. Hence we call it a "doublet". The effects of pole-zero doublets on the frequency response and settling time of OTAs can be found in (Kamath, Meyer, and Gray 1974). This paper shows that doublets may cause severe degradation of settling time while only causing minor changes in the frequency response of the amplifier.

8.3.2 OTA Noise

For the noise analysis we ignore the pole-zero doublet due to C_{gs34} (we assume minor impact due to this) and just consider the dominant pole given by Equation 20. For the noise analysis at the output we set the input signal to zero, and thus we arrive at the simplified small-signal circuit shown in Figure 33.

We see that

$$\overline{V_{\text{gs34}}^2} = \frac{1}{g_{\text{m34}}^2} \left(\overline{I_{\text{n1}}^2} + \overline{I_{\text{n3}}^2} \right).$$

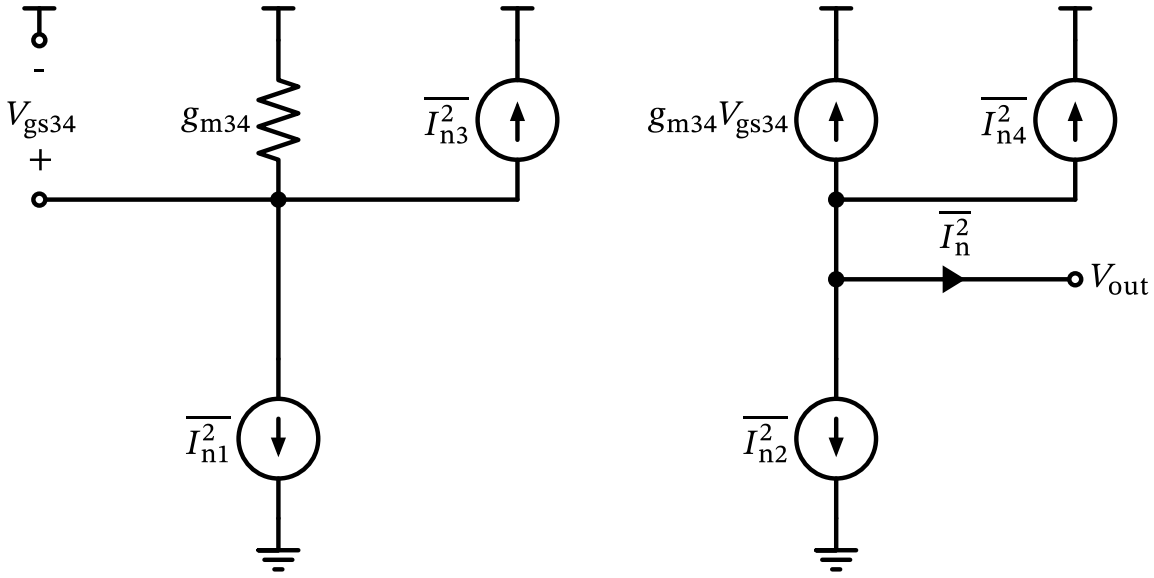


Figure 33: 5-transistor OTA small-signal model for noise calculation.

> **!** Noise Addition
>
> Remember that **uncorrelated** noise quantities need to be power-summed (i.e., $I^2 = I_1^2 + I_2^2$)!

We can then sum the output noise current $\overline{I_n}$ as

$$\overline{I_n^2} = \overline{I_{n2}^2} + \overline{I_{n4}^2} + g_{m34}^2 \frac{1}{g_{m34}^2} \left(\overline{I_{n1}^2} + \overline{I_{n3}^2}\right) = 2\left(\overline{I_{n12}^2} + \overline{I_{n34}^2}\right).$$

As a next step, let us rewrite the OTA transfer function $A(s)$ (see Equation 17) by getting rid of the pole-zero doublet as a simplifying assumption to get

$$A'(s) = \frac{g_{m12}}{g_{ds2} + g_{ds4} + sC_{load}}. \tag{23}$$

Inspecting Equation 23 we can interpret the OTA transfer function as a transconductor g_{m12} driving a load of $Y_{load} = g_{ds2} + g_{ds4} + sC_{load}$. We can thus redraw Figure 30 in the following way, injecting the previously calculated noise current into the output node. The result is shown in Figure 35.

> **i** Output Impedance of the Voltage Buffer
>
> First we short the input terminal to ground and then we connect a current source I_{out} at the output terminal, see Figure 34. Since we can neglect the gate leakage current

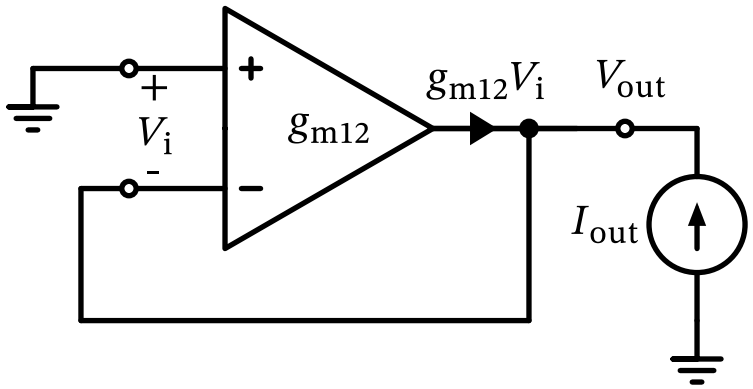


Figure 34: Output impedance calculation of a voltage buffer.

into the inverting input terminal of the OTA, KCL at the output node is simply:

$$I_{out} + g_{m12}\left(-V_{out}\right) = 0$$

Thus, the output impedance is easily calculated.

$$Z_{out} = \frac{V_{out}}{I_{out}} = \frac{V_{out}}{g_{m12}V_{out}} = \frac{1}{g_{m12}}$$

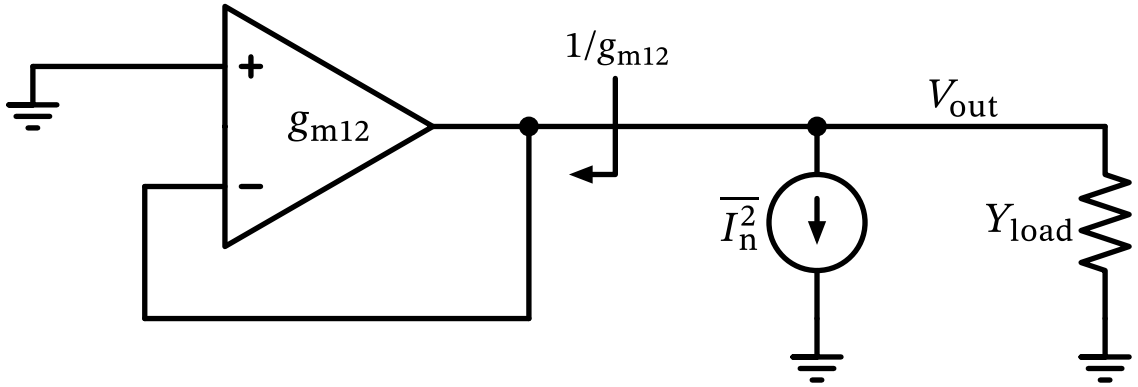


Figure 35: The voltage buffer redrawn for noise analysis.

We see that the feedback around the transconductor g_{m12} creates an impedance of $1/g_{m12}$. We can now calculate the effective load conductance of

$$Y'_{load} = g_{ds2} + g_{ds4} + sC_{load} + g_{m12} \approx g_{m12} + sC_{load}. \tag{24}$$

The output noise voltage is then (using Equation 1)

$$\overline{V_{n,out}^2}(f) = \frac{\overline{I_n^2}}{|Y'_{load}|^2} = \frac{\overline{I_n^2}}{g_{m12}^2 + (2\pi f C_{load})2^2} = \frac{8kT(\gamma_{12}g_{m12} + \gamma_{34}g_{m34})}{g_{m12}^2 + (2\pi f C_{load})^2}.$$

We can use the identity Equation 9 to calculate the rms output noise to

$$V_{\mathrm{n,out,rms}}^2 = \int_0^\infty \overline{V_{\mathrm{n,out}}^2}(f)df = \frac{kT}{C_{\mathrm{load}}}\left(2\gamma_{12} + 2\gamma_{34}\frac{g_{\mathrm{m34}}}{g_{\mathrm{m12}}}\right). \tag{25}$$

Inspecting Equation 25 we can see that the integrated output noise is the kT/C noise of the output load capacitor, enhanced by the γ_{12} of the input differential pair, plus a (smaller) contribution of the current mirror load $M_{3,4}$. Intuitively, this result makes sense.

> **Exercise: Derivation of 5T-OTA Performance**
>
> Please take your time and carefully go through the explanations and derivations for the 5-transistor-OTA in Section 8.2 and Section 8.3. Try to do the calculations yourself; if you get stuck, review the previous chapters.

8.4 5T-OTA Sizing

Outfitted with the governing equations derived in Section 8.3 we can now size the MOSFETs in the OTA, we remember that we have to size $M_{1,2}$ and $M_{3,4}$ equally.

First, we need to select a proper $g_{\mathrm{m}}/I_{\mathrm{D}}$ for the MOSFET. Remembering Section 3 we see that for the input differential pair we should go for a large g_{m}, thus we select a $g_{\mathrm{m}}/I_{\mathrm{D}} = 10$. As g_{ds} of M_2 could limit the dc gain (Equation 18) we go with a rather long $L = 5\,\mu\mathrm{m}$. For current sources a small $g_{\mathrm{m}}/I_{\mathrm{D}}$ is a good idea, so we start with $g_{\mathrm{m}}/I_{\mathrm{D}} = 5$ (because we can not go too low because of $V_{\mathrm{ds,sat}}$) and also an $L = 5\,\mu\mathrm{m}$. The $g_{\mathrm{m}}/I_{\mathrm{D}}$ is also useful to estimate the required drain-source voltage to keep a MOSFET in saturation (i.e., keep the g_{ds} small) with this approximate relationship:

$$V_{\mathrm{ds,sat}} = \frac{2}{g_{\mathrm{m}}/I_{\mathrm{D}}} \tag{26}$$

> **Exercise: 5T-OTA Sizing**
>
> Please size the 5T-OTA according to the previous $g_{\mathrm{m}}/I_{\mathrm{D}}$ and L suggestions. Please calculate the W of M_{1-6} and the total supply current. Please check wether gain error, total output noise, and turn-on settling is met with the calculated devices sizes and bias currents.

The sizing procedure and its calculation are best performed in a Jupyter notebook, as we can easily look up the exact data from the pre-computed tables:

Solution: 5T-OTA Sizing

Sizing for Basic 5T-OTA


```
# read table data
from pygmid import Lookup as lk
import numpy as np
lv_nmos = lk('sg13_lv_nmos.mat')
lv_pmos = lk('sg13_lv_pmos.mat')
# list of parameters: VGS, VDS, VSB, L, W, NFING, ID, VT, GM, GMB, GDS, CGG, CGB, CGD, C
# if not specified, minimum L, VDS=max(vgs)/2=0.9 and VSB=0 are used
```

```
# define the given parameters as taken from the specification table or inital guesses
c_load = 50e-15
gm_id_m12 = 10
gm_id_m34 = 5
gm_id_m56 = 5
l_12 = 5
l_34 = 5
l_56 = 5
f_bw = 10e6 # -3dB bandwidth of the voltage buffer
i_total_limit = 10e-6
i_bias_in = 20e-6
output_voltage = 1.3
vin_min = 0.7
vin_max = 0.9
vdd_min = 1.45
vdd_max = 1.55
```

```
# we get the required gm of M1/2 from the -3dB bandwidth requirement of the voltage buff
# note that the -3dB bandwidth of the voltage buffer with gain Av=1 is equal to the unit
# of the ota, hence we wet them equal here
# we add a factor of 3 to allow for PVT variation plus additional MOSFET parasitic loadi
gm_m12 = f_bw * 3 * 4*np.pi*c_load
print('gm12 =', round(gm_m12/1e-3, 4), 'mS')
```

```
gm12 = 0.0188 mS
```

```
# since we know gm12 and the gmid we can calculate the bias current
id_m12 = gm_m12 / gm_id_m12
i_total = 2*id_m12
print('i_total (exact) =', round(i_total/1e-6, 1), 'µA')
# we round to 0.5µA bias currents
i_total = max(round(i_total / 1e-6 * 2) / 2 * 1e-6, 0.5e-6)
id_m12 = i_total/2

print('i_total (rounded) =', i_total/1e-6, 'µA')
if i_total < i_total_limit:
    print('[info] power consumption target is met!')
else:
    print('[info] power consumption target is NOT met!')
```

```
i_total (exact) = 3.8 µA
i_total (rounded) = 4.0 µA
[info] power consumption target is met!
```

```
# we calculate the dc gain
gm_gds_m12 = lv_nmos.lookup('GM_GDS', GM_ID=gm_id_m12, L=l_12, VDS=0.75, VSB=0)
gm_gds_m34 = lv_pmos.lookup('GM_GDS', GM_ID=gm_id_m34, L=l_34, VDS=0.75, VSB=0)

gds_m12 = gm_m12 / gm_gds_m12
gm_m34 = gm_id_m34 * i_total/2
gds_m34 = gm_m34 / gm_gds_m34

a0 = gm_m12 / (gds_m12 + gds_m34)
print('a0 =', round(20*np.log10(a0), 1), 'dB')
```

```
a0 = 34.8 dB
```

```
# we calculate the MOSFET capacitance which adds to Cload, to see the impact on the BW
gm_cgs_m12 = lv_nmos.lookup('GM_CGS', GM_ID=gm_id_m12, L=l_12, VDS=0.75, VSB=0)
gm_cdd_m12 = lv_nmos.lookup('GM_CDD', GM_ID=gm_id_m12, L=l_12, VDS=0.75, VSB=0)
gm_cdd_m34 = lv_pmos.lookup('GM_CDD', GM_ID=gm_id_m34, L=l_34, VDS=0.75, VSB=0)

c_load_parasitic = abs(gm_m12/gm_cgs_m12) + abs(gm_m12/gm_cdd_m12) + abs(gm_m34/gm_cdd_m
print('additional load capacitance =', round(c_load_parasitic/1e-15, 1), 'fF')

f_bw = gm_m12 / (4*np.pi * (c_load + c_load_parasitic))
print('unity gain bandwidth incl. parasitics =', round(f_bw/1e6, 2), 'MHz')
```

```
additional load capacitance = 54.9 fF
unity gain bandwidth incl. parasitics = 14.3 MHz
```

```
# we can now look up the VGS of the MOSFET
vgs_m12 = lv_nmos.look_upVGS(GM_ID=gm_id_m12, L=l_12, VDS=0.75, VSB=0.0)
vgs_m34 = lv_pmos.look_upVGS(GM_ID=gm_id_m34, L=l_34, VDS=0.75, VSB=0.0)
vgs_m56 = lv_nmos.look_upVGS(GM_ID=gm_id_m56, L=l_56, VDS=0.75, VSB=0.0)

print('vgs_12 =', round(float(vgs_m12), 3), 'V')
print('vgs_34 =', round(float(vgs_m34), 3), 'V')
print('vgs_56 =', round(float(vgs_m56), 3), 'V')
```

```
vgs_12 = 0.367 V
vgs_34 = 0.729 V
vgs_56 = 0.591 V
```

```
# calculate settling time due to slewing with the calculated bias current
t_slew = (c_load + c_load_parasitic) * output_voltage / i_total
print('slewing time =', round(t_slew/1e-6, 3), 'µs')
t_settle = 5/(2*np.pi*f_bw)
print('settling time =', round(t_settle/1e-6, 3), 'µs')
```

```
slewing time = 0.034 µs
settling time = 0.056 µs
```

```
# calculate voltage gain error
gain_error = a0 / (1 + a0)
print('voltage gain error =', round((gain_error-1)*100, 1), '%')
```

```
voltage gain error = -1.8 %
```

```
# calculate total rms output noise
sth_m12 = lv_nmos.lookup('STH_GM', VGS=vgs_m12, L=l_12, VDS=0.75, VSB=0) * gm_m12
gamma_m12 = sth_m12/(4*1.38e-23*300*gm_m12)

sth_m34 = lv_pmos.lookup('STH_GM', VGS=vgs_m34, L=l_34, VDS=0.75, VSB=0) * gm_m34
gamma_m34 = sth_m34/(4*1.38e-23*300*gm_m34)

output_noise_rms = np.sqrt(1.38e-23*300 / (c_load + c_load_parasitic) * (2*gamma_m12 + 2
print('output noise =', round(output_noise_rms/1e-6, 1), 'µVrms')
```

```
output noise = 354.2 µVrms
```

```
# calculate all widths
id_w_m12 = lv_nmos.lookup('ID_W', GM_ID=gm_id_m12, L=l_12, VDS=vgs_m12, VSB=0)
w_12 = id_m12 / id_w_m12
w_12_round = max(round(w_12*2)/2, 0.5)
print('M1/2 W =', round(w_12, 2), 'um, rounded W =', w_12_round, 'um')

id_m34 = id_m12
id_w_m34 = lv_pmos.lookup('ID_W', GM_ID=gm_id_m34, L=l_34, VDS=vgs_m34, VSB=0)
w_34 = id_m34 / id_w_m34
w_34_round = max(round(w_34*2)/2, 0.5)
print('M3/4 W =', round(w_34, 2), 'um, rounded W =', w_34_round, 'um')

id_w_m5 = lv_nmos.lookup('ID_W', GM_ID=gm_id_m56, L=l_56, VDS=vgs_m56, VSB=0)
w_5 = i_total / id_w_m5
w_5_round = max(round(w_5*2)/2, 0.5)
print('M5 W =', round(w_5, 2), 'um, rounded W =', w_5_round, 'um')
w_6 = w_5_round * i_bias_in / i_total
w_6_round = max(round(w_6*2)/2, 0.5)
print('M6 W =', round(w_6_round, 2), 'um')
```

```
M1/2 W = 1.77 um, rounded W = 2.0 um
M3/4 W = 1.64 um, rounded W = 1.5 um
M5 W = 0.74 um, rounded W = 0.5 um
M6 W = 2.5 um
```

```
# print out final design values
print('5T-OTA dimensioning:')
print('--------------------')
print('M1/2 W=', w_12_round, ', L=', l_12)
print('M3/4 W=', w_34_round, ', L=', l_34)
print('M5   W=', w_5_round, ', L=', l_56)
print('M6   W=', w_6_round, ', L=', l_56)
print()
print('5T-OTA performance summary:')
print('---------------------------')
print('supply current =', round(i_total/1e-6, 1), 'µA')
print('output noise =', round(output_noise_rms/1e-6, 1), 'µVrms')
print('voltage gain error =', round((gain_error-1)*100, 1), '%')
print('unity gain bandwidth incl. parasitics =', round(f_bw/1e6, 2), 'MHz')
print('turn-on time (slewing+settling) =', round((t_slew+t_settle)/1e-6, 3), 'µs')
print()
print('5T-OTA bias point check:')
print('------------------------')
print('headroom M1 =', round(vdd_min-vgs_m34+vgs_m12-vin_max, 3), 'V')
print('headroom M4 =', round(vdd_min-vin_max, 3), 'V')
print('headroom M5 =', round(vin_min-vgs_m12, 3), 'V')
```

```
5T-OTA dimensioning:
--------------------
M1/2 W= 2.0 , L= 5
M3/4 W= 1.5 , L= 5
M5   W= 0.5 , L= 5
M6   W= 2.5 , L= 5

5T-OTA performance summary:
---------------------------
supply current = 4.0 µA
output noise = 354.2 µVrms
voltage gain error = -1.8 %
unity gain bandwidth incl. parasitics = 14.3 MHz
turn-on time (slewing+settling) = 0.09 µs

5T-OTA bias point check:
------------------------
headroom M1 = 0.188 V
headroom M4 = 0.55 V
headroom M5 = 0.333 V
```

8.1 5T-OTA Simulation

With the initial sizing of the MOSFETs of the 5T-OTA done, we can design the 5T-OTA circuit and setup a simulation testbench to check the performance parameters. Since this is the first time we draw a more complex schematic, and use a hierarchical design, we should note that drawing a schematic is an art, and there exists a set of rules and recommendations how to name pins, how to use annotations, and so on. Please read Section 21 before you start into your design work.

> Exercise: 5T-OTA Design and Testbench
>
> Please design the circuit of the 5T-OTA. Put the OTA circuit in a separate schematic, create a symbol for it, and use this symbol in a testbench you create in Xschem for this 5T-OTA used as a voltage buffer as shown in Figure 30. Use typical conditions for the simulation, and check how well the specification in Table 2 is met, and how well the derivations in Section 8.2 and Section 8.3 fit to the simulation results.
> If you get stuck, you can find the testbench and 5T-OTA schematic here (for the small-signal analysis) and here (for the large-signal settling simulation). For interested students, the loop gain analysis with Middlebrook's and Tian's method of the 5T-OTA can be found here.

8.2 MOSFET Mismatch

So far, we have assumed that implemented MOSFETs show no difference from device to device, which means that two transistors behave completely identical, and the resulting differential circuits are fully symmetric. However, due to manufacturing tolerances, this is not the case in reality. MOSFETs will show **mismatch** due to tiny random fluctuations in manufacturing, and we have to account for this.

For a typical MOSFET, we usually consider two main mismatch effects (Pelgrom, Duinmaijer, and Welbers 1989):

- A variation of the threshold voltage (mainly due to variations in doping levels).
- A variation of the critical dimensions of the MOSFET (W and L as well as vertical dimensions).

Both effects influence the drain current of the MOSFET, and they are indirectly proportional to the size of the MOSFET. So, if we want to reduce the mismatch, we have to increase the size of the MOSFET by increasing its gate area $W \cdot L$.

If we formulate the drain current mismatch $\Delta I_\mathrm{D}/I_\mathrm{D}$ of two nominally identical MOSFET we get

$$\frac{\Delta I_\mathrm{D}}{I_\mathrm{D}} = \frac{A_\mathrm{mosfet}}{\sqrt{WL}} \tag{27}$$

with A_{mosfet} being a mismatch parameter for a given CMOS technology.

MOSFET Mismatch

Usually, the mismatch in MOSFETs is characterized via two mismatch parameters (Pelgrom, Duinmaijer, and Welbers 1989):

- The threshold voltage mismatch

$$\Delta V_{\text{th}} = \frac{A_{\text{vth}}}{\sqrt{WL}} \tag{28}$$

- and the size mismatch

$$\frac{\Delta(W/L)}{(W/L)} = \frac{A_{\text{k}}}{\sqrt{WL}}. \tag{29}$$

The resulting input offset voltage V_{offs} of a differential pair is then given by (Razavi 2017)

$$V_{\text{offs}} = \sqrt{\left(\frac{V_{\text{GS}} - V_{\text{th}}}{2}\frac{\Delta(W/L)}{(W/L)}\right)^2 + \Delta V_{\text{th}}^2}. \tag{30}$$

The mismatch in the drain current of two current-mirror transistors is given by (Razavi 2017)

$$\frac{\Delta I_{\text{D}}}{I_{\text{D}}} = \frac{\Delta(W/L)}{(W/L)} - 2\frac{\Delta V_{\text{th}}}{V_{\text{GS}} - V_{\text{th}}}. \tag{31}$$

To minimize the input offset voltage in a differential pair we should strive to minimize V_{GS} (see Equation 30) and to minimize the mismatch in current mirrors we should target a large V_{GS} (see Equation 31). In both cases the MOSFET area $W \cdot L$ needs to be large enough (see Equation 28 and Equation 29).

How can we now cope with mismatch in design and simulation? We can account the transitor mismatch according to Equation 28 and Equation 29 in circuit analysis and quantify its effect. We can then take this into account when performing the circuit sizing procedure.

We can also simulate the effects of MOSFET mismatch on circuit performance by doing a **Monte Carlo** simulation. Here, a random realization of the variations of all circuit components (where mismatch is modelled) is taken and the circuit is simulated. Then, another random realization is simulated, and so on. In summary, we run the same simulation for different realizations N times and evaluate the resulting variations of the circuit parameters. If N is large enough then the distributions should approach a Gaussian distribution, and we can then estimate the variances of the circuit parameters. Using statistical analysis we can then assess the **yield** of a circuit, i.e., how many circuit realizations will meet the specification.

> **Number of Monte Carlo Simulation Runs**
>
> The number of Monte Carlo simulation runs N has to be large enough to approach Gaussian distributions to allow estimation of the variances. However, this comes at the cost of a large simulation time, so a balance has to be found. Often, $N = 250$ is a good compromise between simulation time and well-behaved parameter distributions.

As you can see in the previous discussion, running Monte Carlo simulations is a tedious process due to the large number of involved simulations and the required data processing of the heaps of simulation data. Luckily, CACE supports this type of simulation, and we will use it in Section 8.4.

8.3 Resistor Mismatch

Similar to the MOSFET mismatch discussed in Section 8.2, resistors will also show mismatch. Equivalently to Equation 27 the resistor mismatch $\Delta R/R$ can be characterized by

$$\frac{\Delta R}{R} = \frac{A_{\text{res}}}{\sqrt{WL}} \tag{32}$$

with A_{res} being a mismatch parameter for a given CMOS technology.

With Equation 32 we now have two criteria for how to select the width W of a specific resistor (the length L is then derived from the required resistance value with $R = R_{\square} \cdot L/W$):

1. The current handling capability (the larger the W, the more dc current a resistor can carry), and
2. the resistor mismatch (the larger the W, the larger the WL for a given L/W).

If a resistor's dimensions are not limited by the two criteria above then we usually choose the minimum width that is allowed for a given resistor in a specific technology (for SG13G2 it is $L_{\text{min}} = 0.5\,\mu\text{m}$) to save area and to minimize the parasitic capacitance of the resistor to substrate.

8.4 5T-OTA Simulation versus PVT and MC

As you have seen in Section 8.1 running simulations by hand is tedious. When we want to check the overall performance, we have to run many simulations over various conditions:

1. The supply voltage of the circuit has tolerances, and thus we need to check the performance against this variation.
2. The temperature at which the circuit is operated is likely changing. Also the performance against this has to be verified.

3. When manufacturing the wafers random variations in various process parameters lead to changed parameters of the integrated circuit components. In order to check for this effect, wafer foundries provide model files which shall cover these manufacturing excursions. Simplified, this leads to a slower or faster MOSFET, and usually NMOS and PMOS are not correlated, so we have the process corners **SS**, **SF**, **TT**, **FS**, and **FF**. So far, we have only used the **TT** models in our simulations.

The variations listed in the previous list are abbreviated as **PVT** (process, voltage, temperature) variations. In order to finalize a circuit all combinations of these (plus the variations in operating conditions like input voltage) have to be simulated. As you can imagine, this leads to a huge number of simulations, and simulation results which have to be evaluated for pass/fail.

There are two options how to tackle this efficiently:

1. As an experienced designer you have a very solid understanding of the circuit, plus based on the analytic equations you can identify which combination of operating conditions will lead to a worst case performance. Thus, you can drastically reduce the number of corners to simulate, and you run them by hand.
2. You are using a framework which highly automates this task of running a plethora of different simulations and evaluating the outcome. These frameworks are called simulation runners.

Luckily, there are open-source versions of simulation runners available, and we will use CACE in this lecture. CACE is written in Python and allows to setup a datasheet in YAML which defines the simulation problem and the performance parameters to evaluate against which limits. The resulting simulations are then run in parallel and the simulation data is evaluated and summarized in various forms.

There is a CACE setup available for our 5T-OTA. The datasheet describes the operating conditions and the simulations tasks. For each simulation a testbench template is needed, this one is used for ac simulations, this one is used for noise simulation, and this one is used for transient simulation.

> ⚠ Running CACE Simulation
>
> The CACE simulation run can be started with
>
> ```
> cace cace/voltage-buffer-ota.yaml
> ```
>
> The simulation results are then placed into the `cace/_docs` folder. If in addition to the default Markdown report an HTML output is needed for easier review then using Pandoc it can be easily converted and viewed with with
>
> ```
> cace/cace_view.sh cace/_docs/ota-5t_schematic.md
> ```

After a successful run, a documentation is automatically generated. The result of a full run of this OTA design is presented here:

Note 2: CACE Summary for 5T-OTA

CACE Summary for ota-5t

netlist source: schematic

Parameter	Tool	Result	Min Limit	Min Value	Typ Target	Typ Value	Max Limit	Max Value	Status
Output voltage ratio	ngspice	gain	0.97 V/V	0.987 V/V	any	1.000 V/V	1.03 V/V	1.007 V/V	Pass
Bandwidth	ngspice	bw	10e6 Hz	15550400.000 Hz	any	26912100.000 Hz	any	34052200.000 Hz	Pass
Output voltage ratio (MC)	ngspice	gain_mc	any	0.837 V/V	any	1.010 V/V	any	1.185 V/V	Pass
Bandwidth (MC)	ngspice	bw_mc	10e6 Hz	18988400.000 Hz	any	26567100.000 Hz	any	29443100.000 Hz	Pass
Output noise	ngspice	noise	any	0.308 mV	any	0.371 mV	1 mV	0.454 mV	Pass
Settling time	ngspice	tsettle	any	0.137 us	any	0.144 us	10 us	0.156 us	Pass

Plots

gain_vs_temp

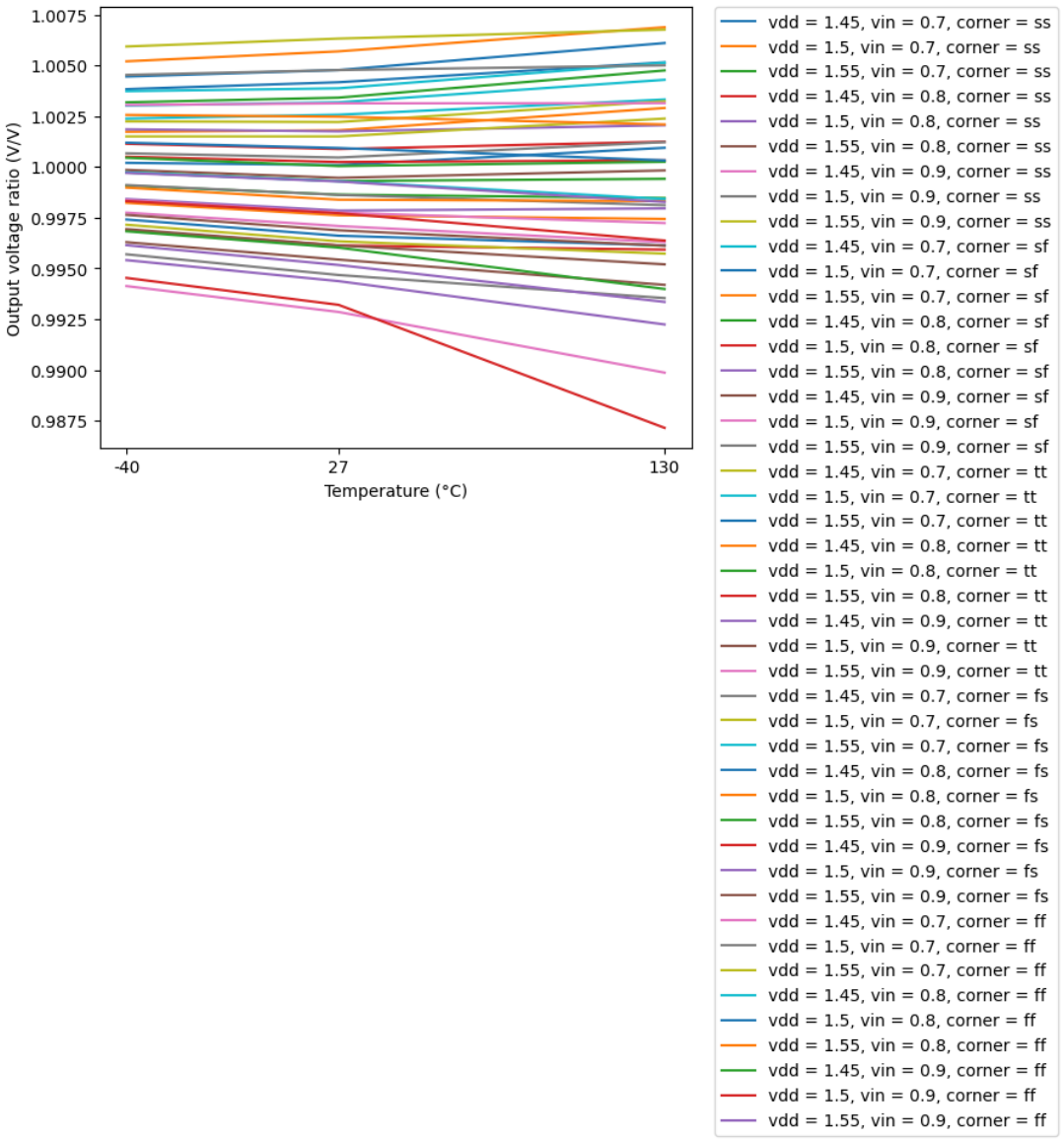


Figure 36: gain_vs_temp

gain_vs_vin

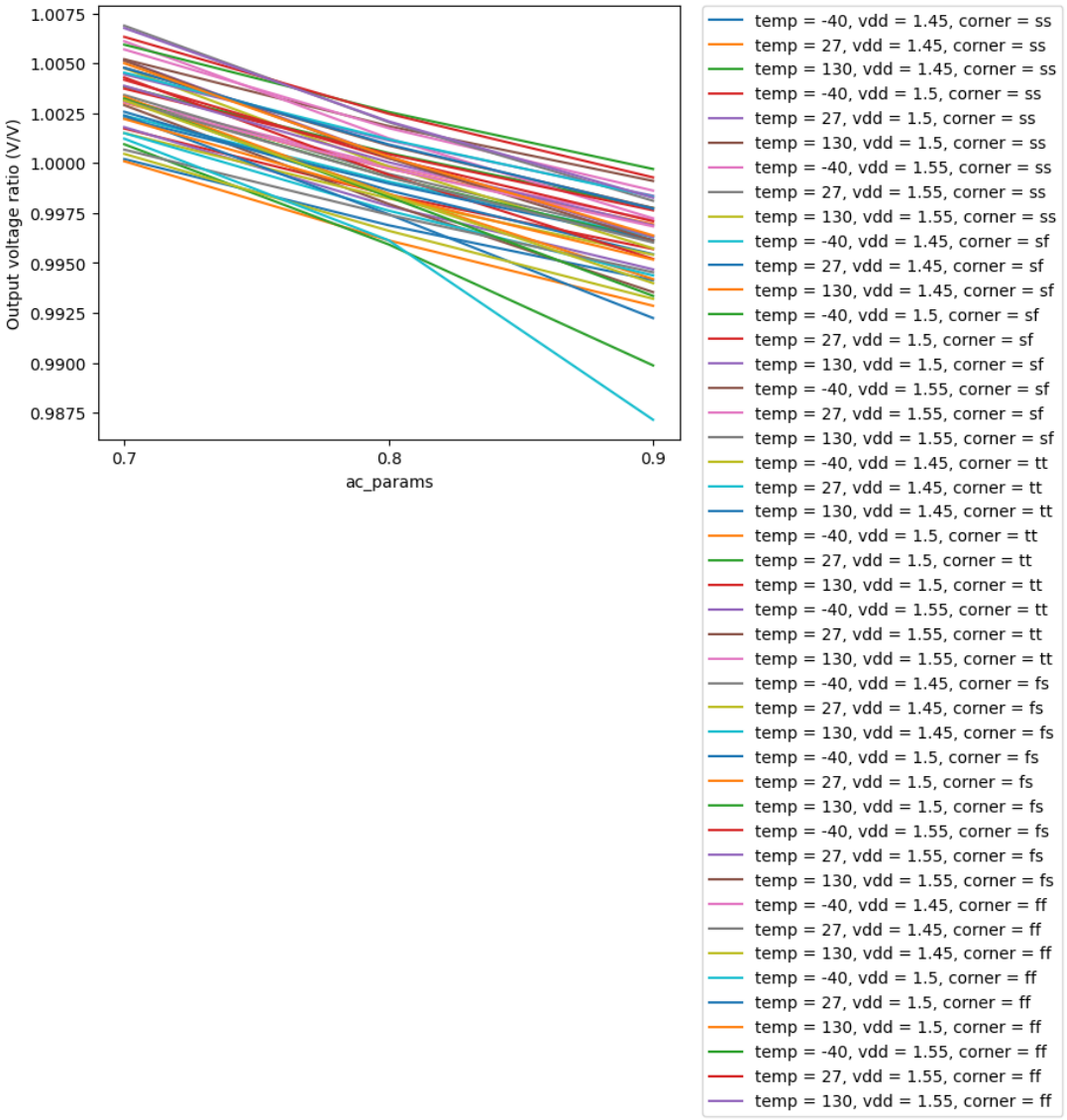


Figure 37: gain_vs_vin

gain_vs_vdd

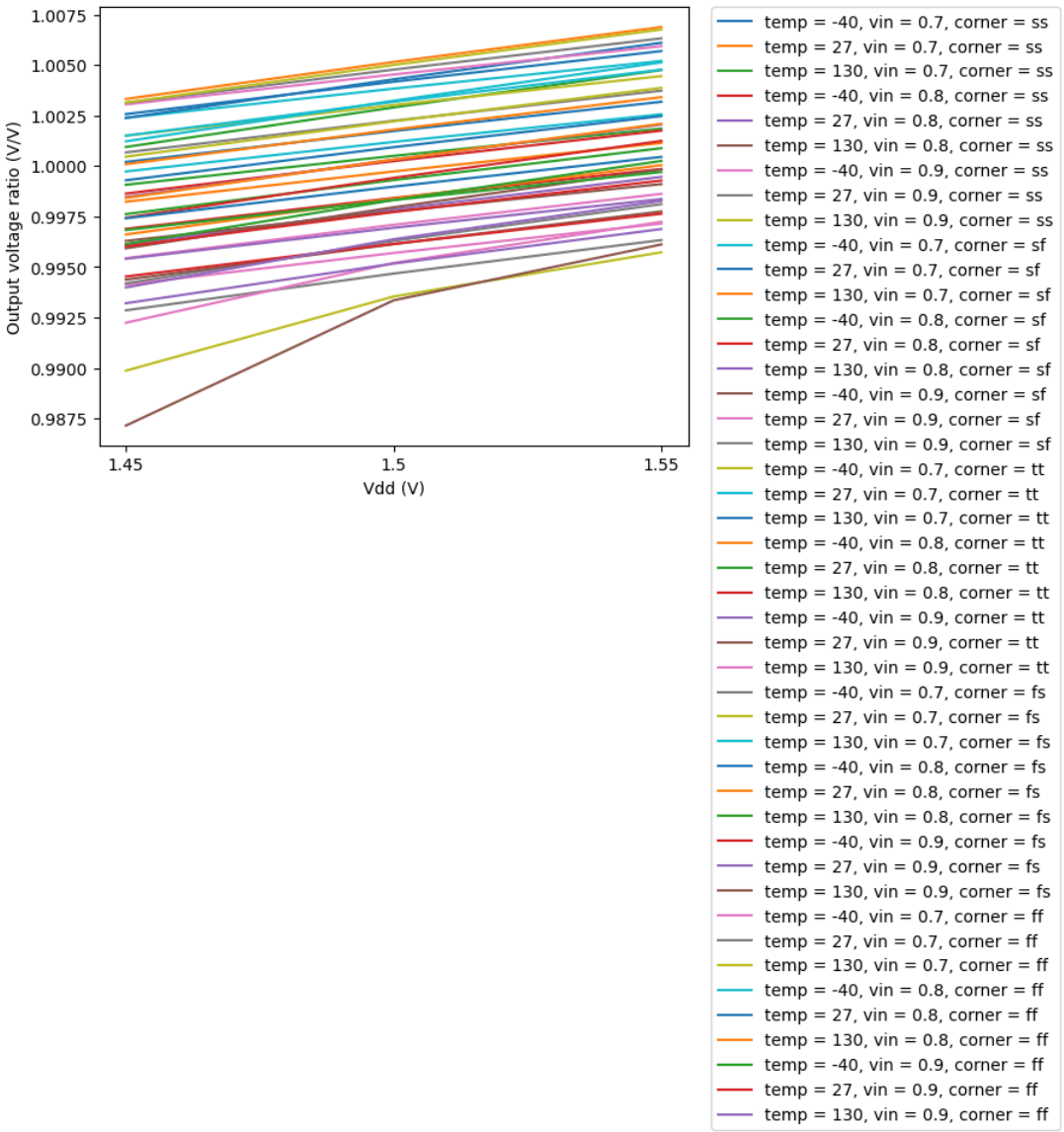


Figure 38: gain_vs_vdd

gain_vs_corner

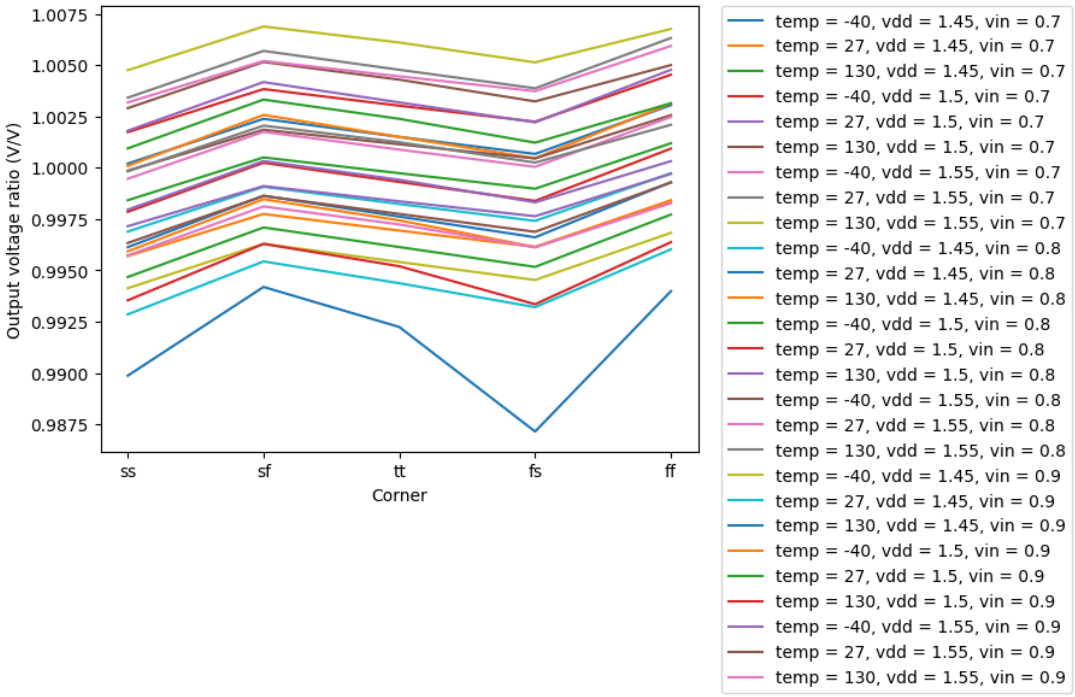


Figure 39: gain_vs_corner

bw_vs_temp

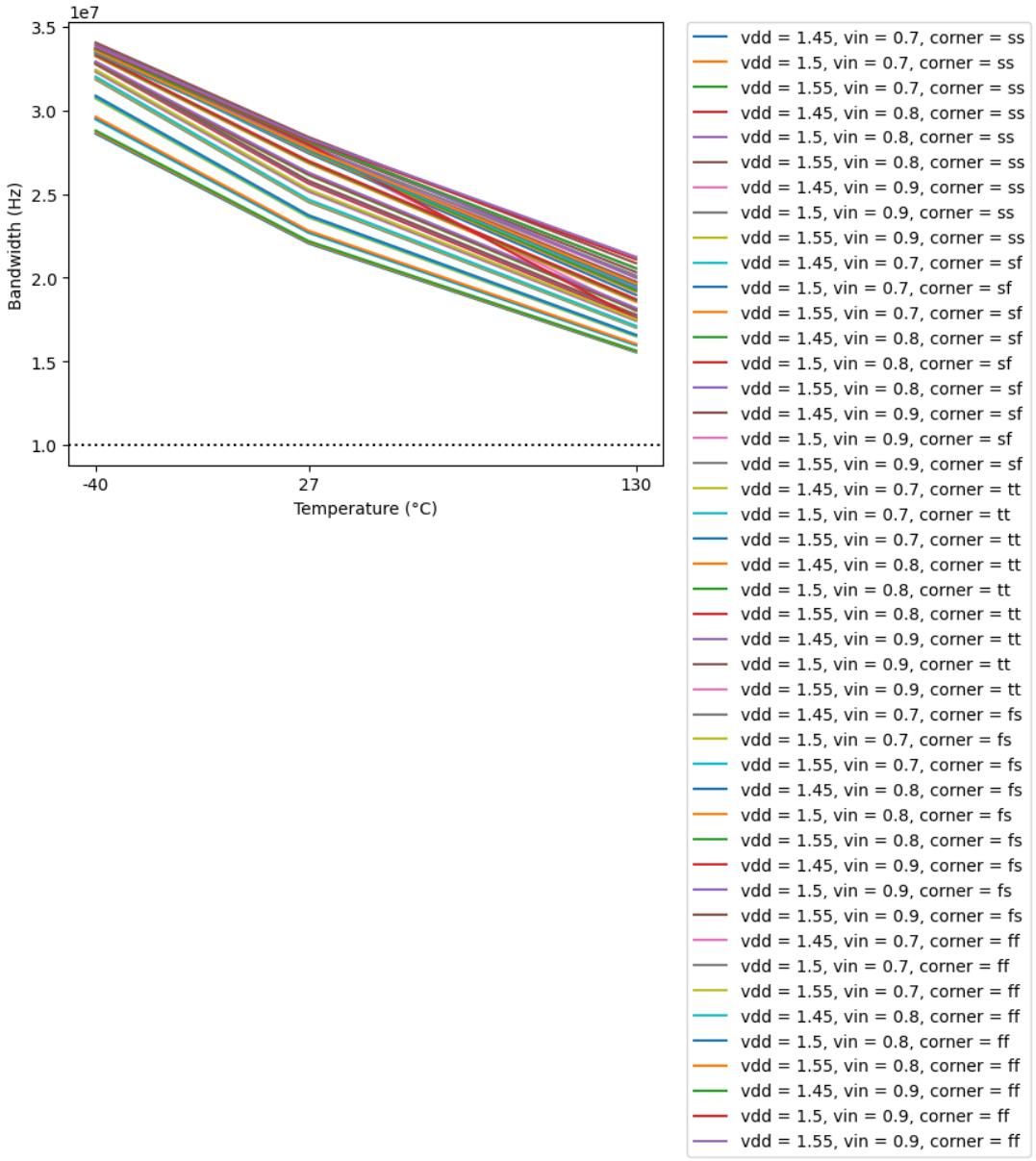


Figure 40: bw_vs_temp

bw_vs_vin

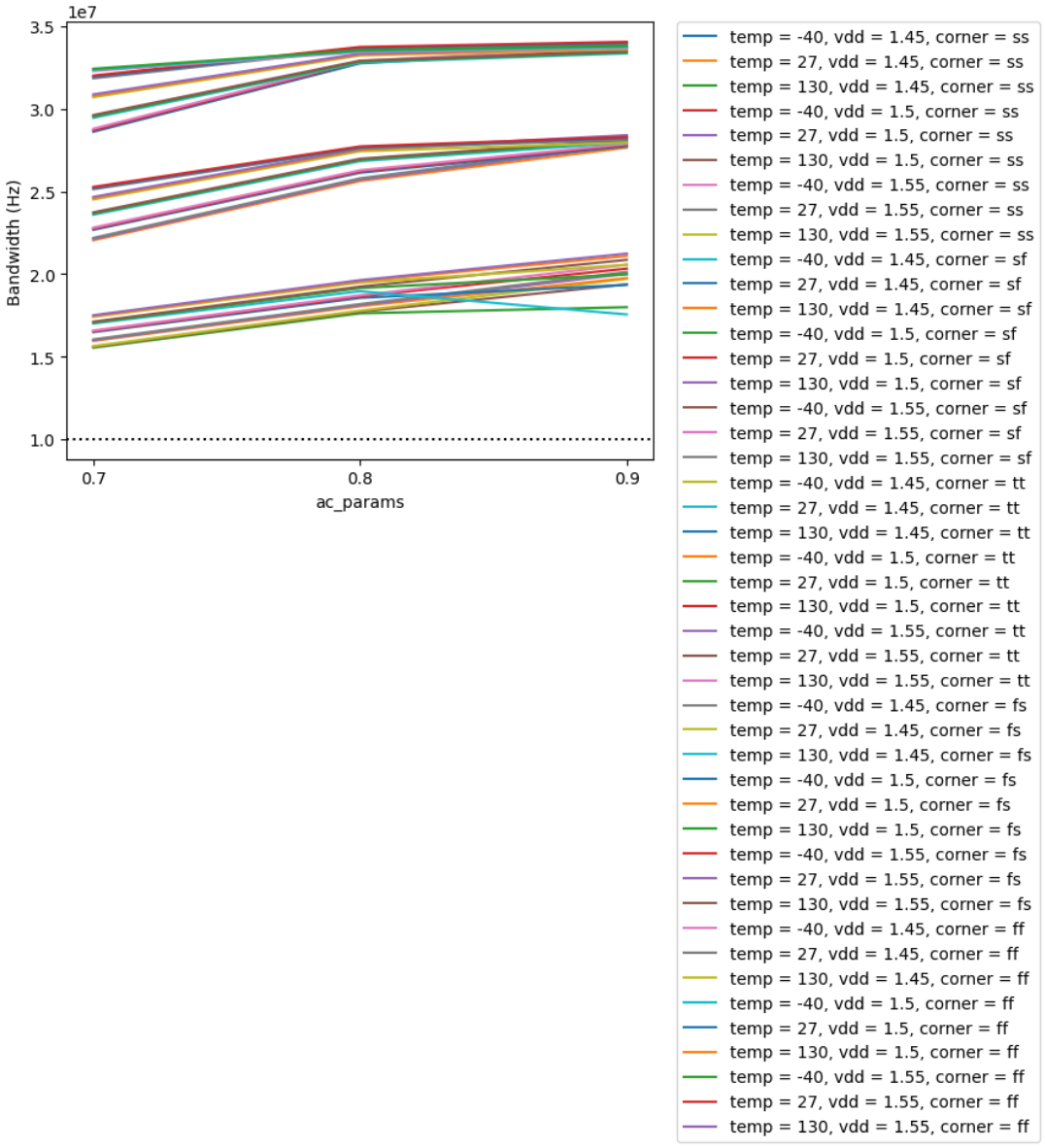


Figure 41: bw_vs_vin

bw_vs_vdd

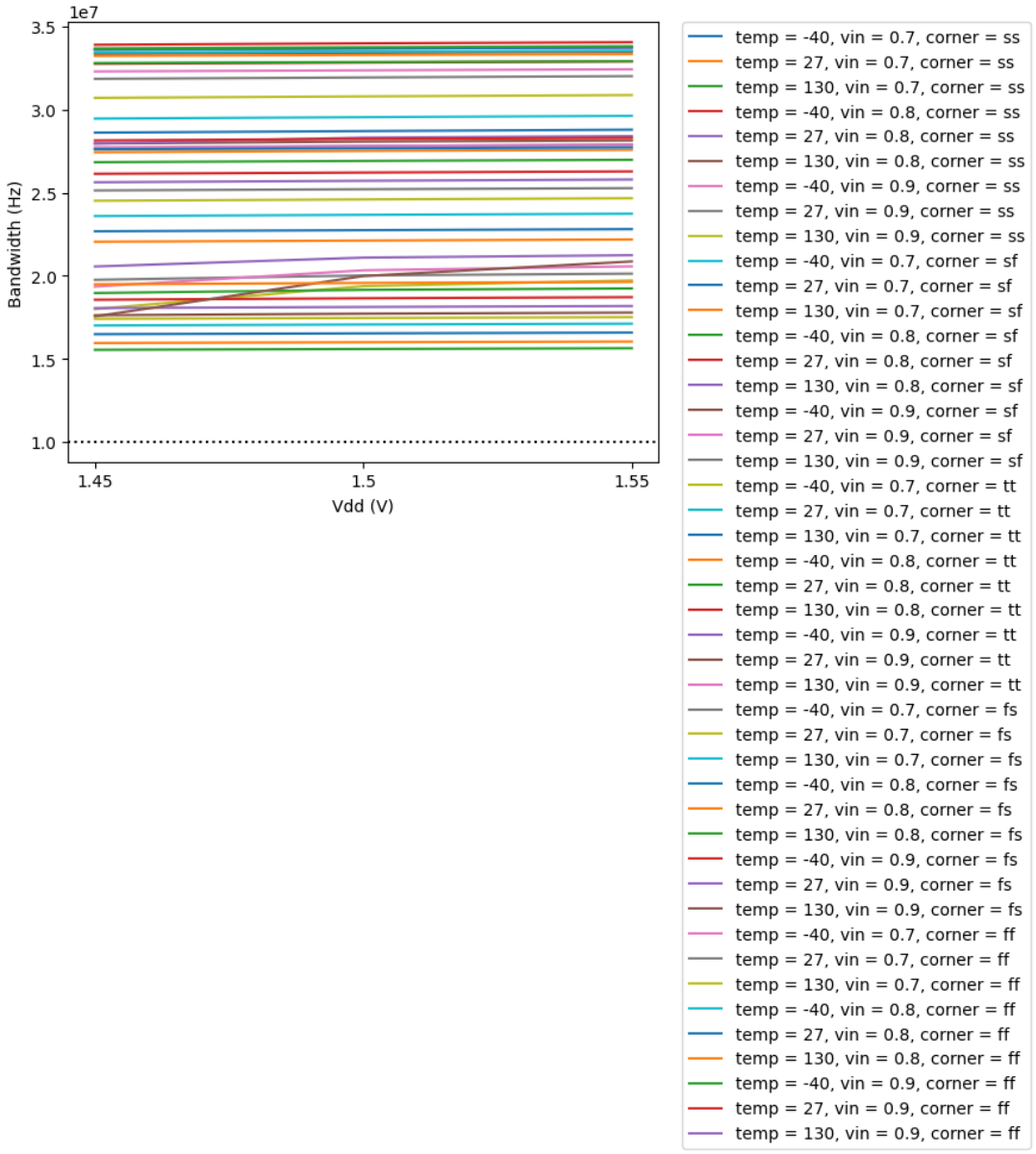


Figure 42: bw_vs_vdd

bw_vs_corner

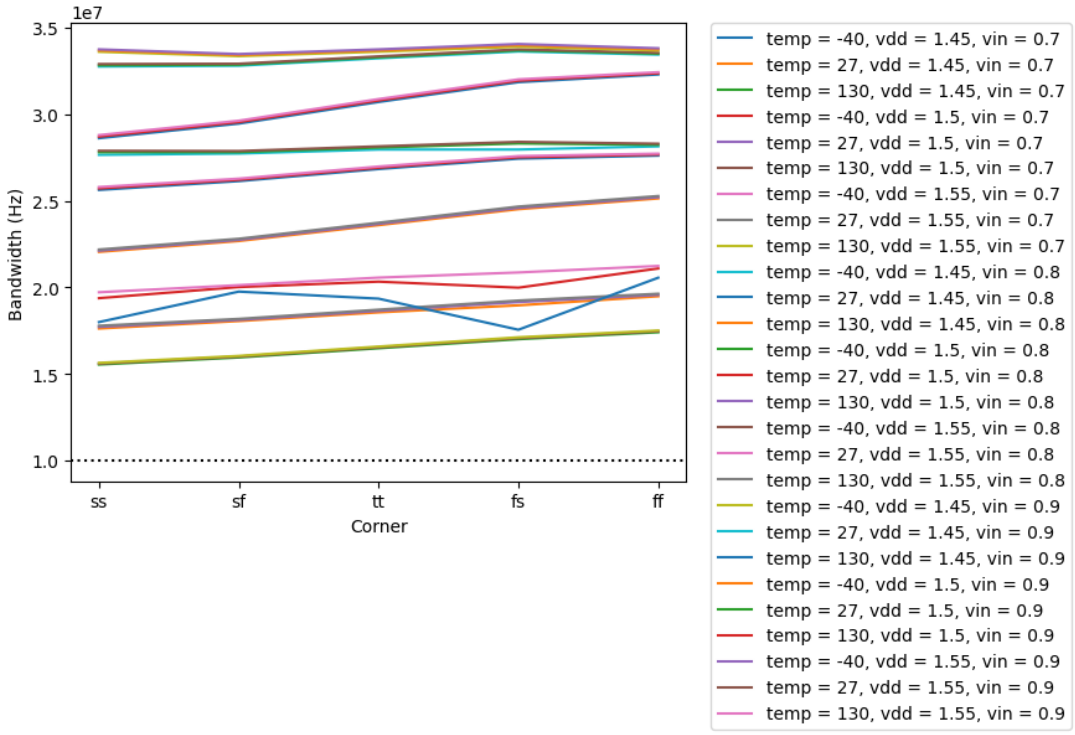


Figure 43: bw_vs_corner

gain_mc

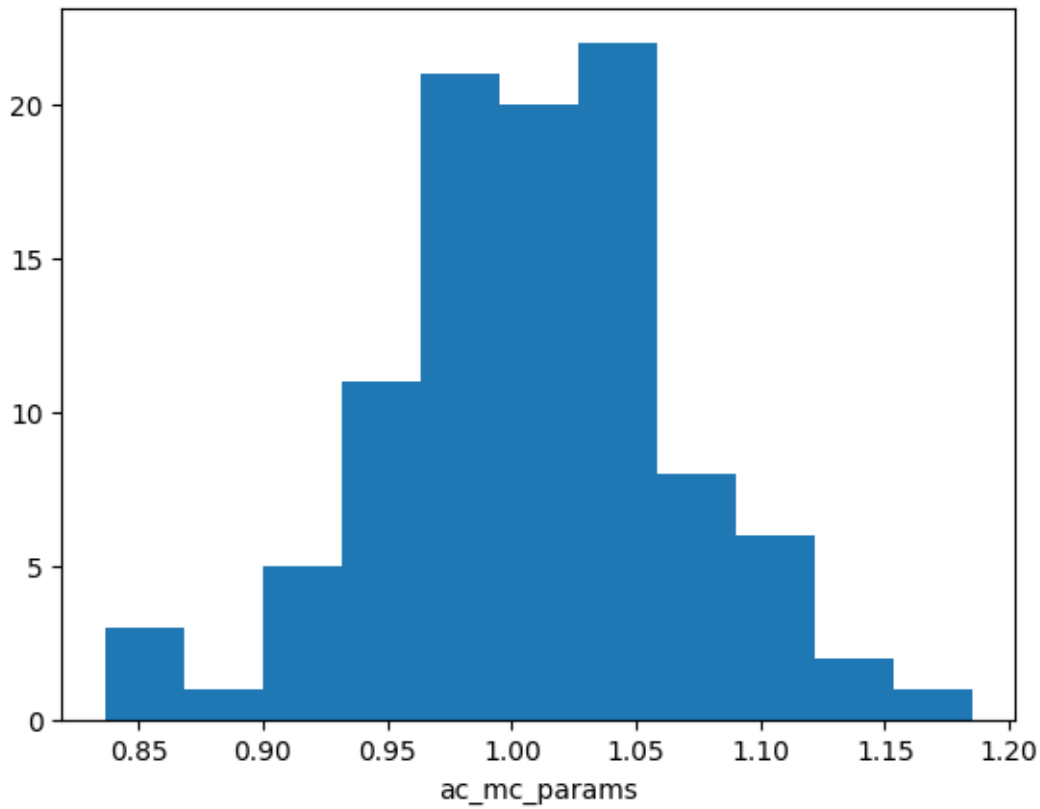


Figure 44: gain_mc

bw_mc

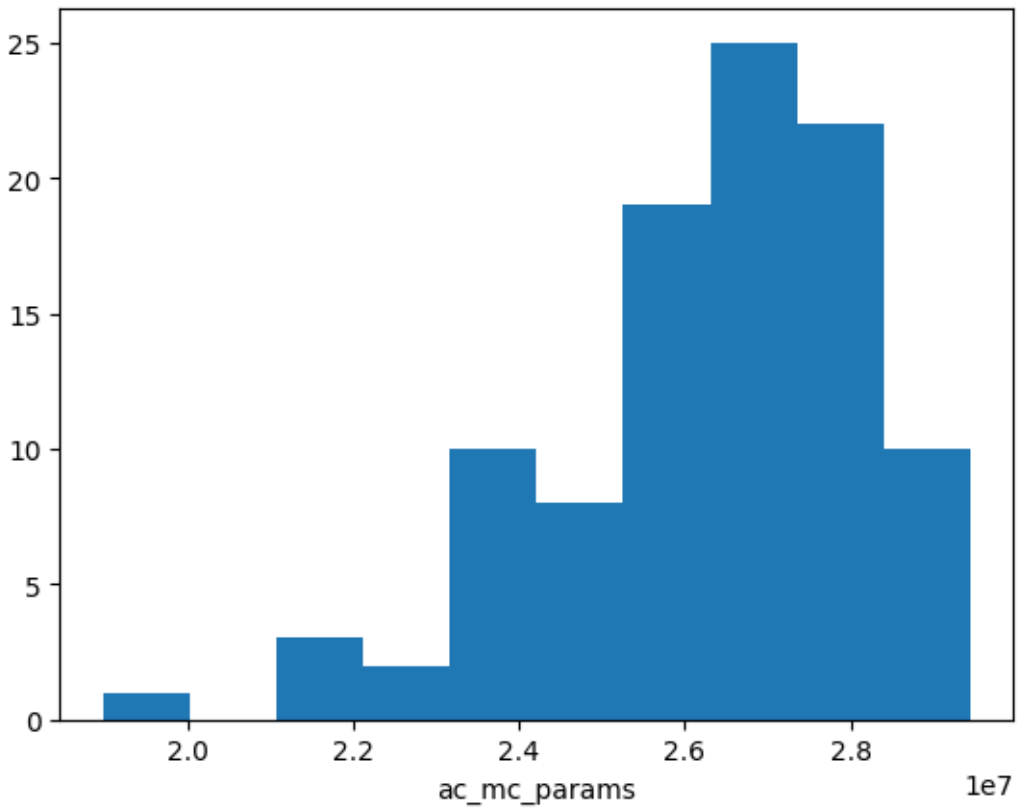


Figure 45: bw_mc

noise_vs_temp

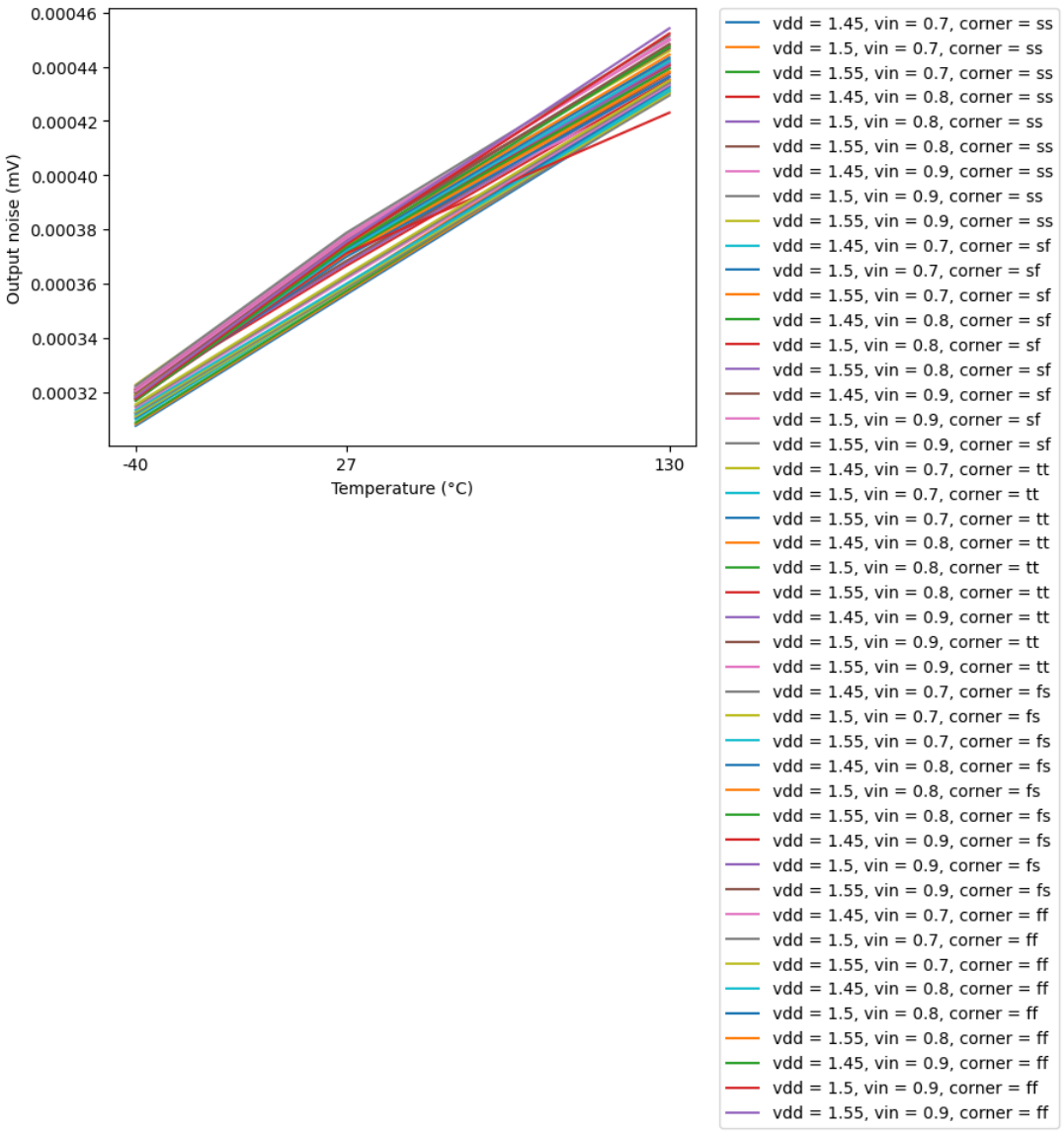


Figure 46: noise_vs_temp

noise_vs_vin

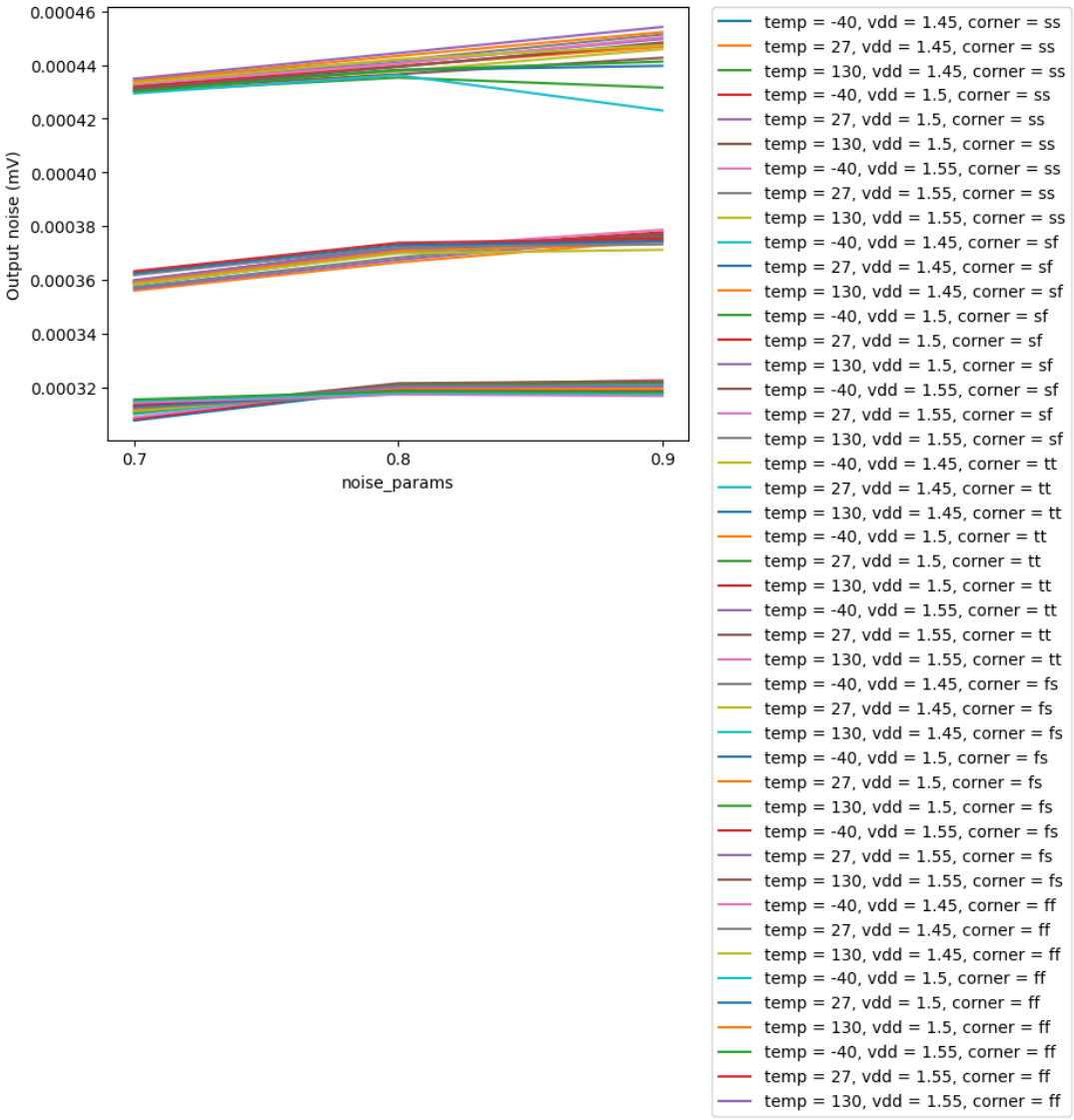


Figure 47: noise_vs_vin

noise_vs_vdd

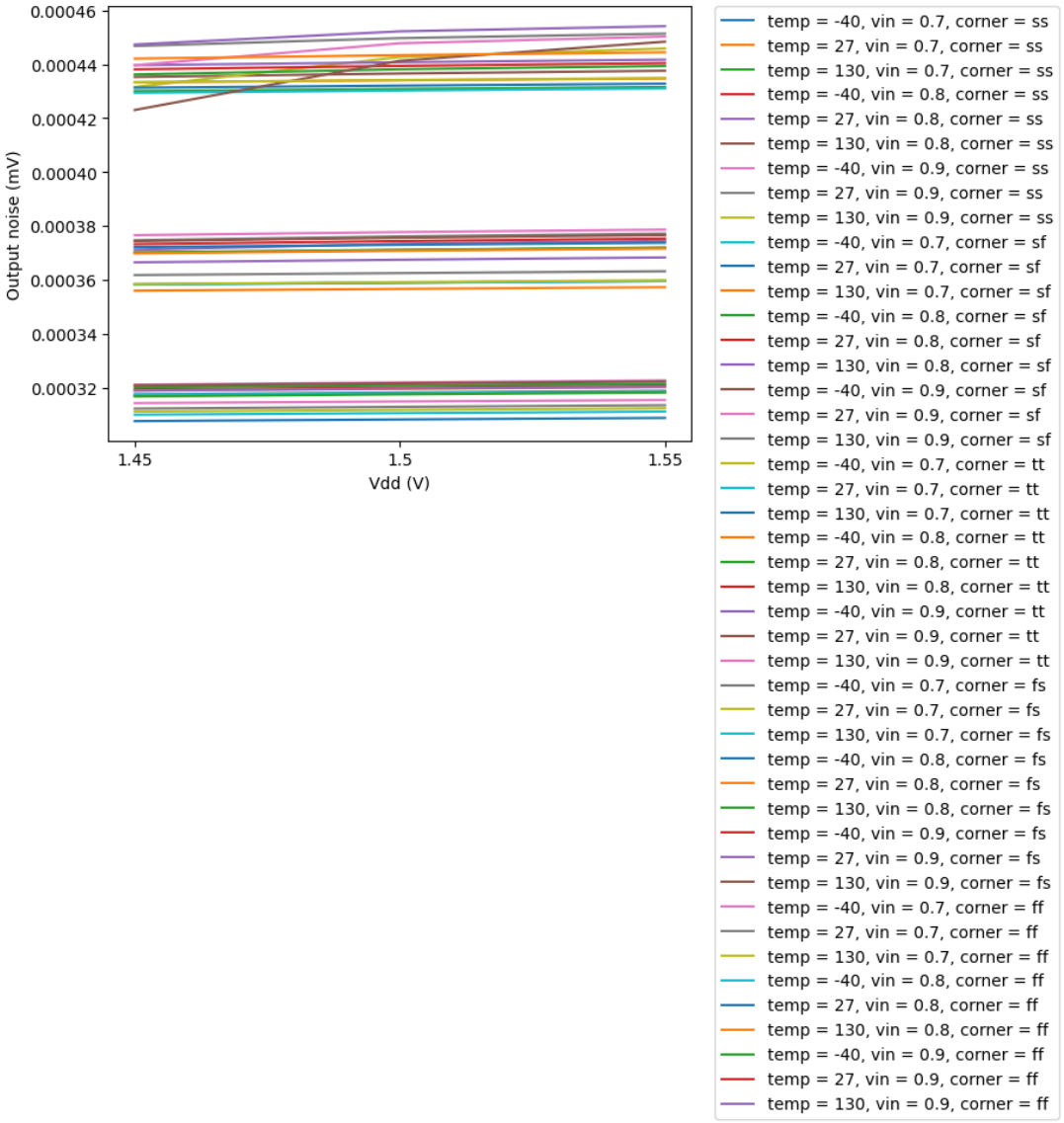


Figure 48: noise_vs_vdd

noise_vs_corner

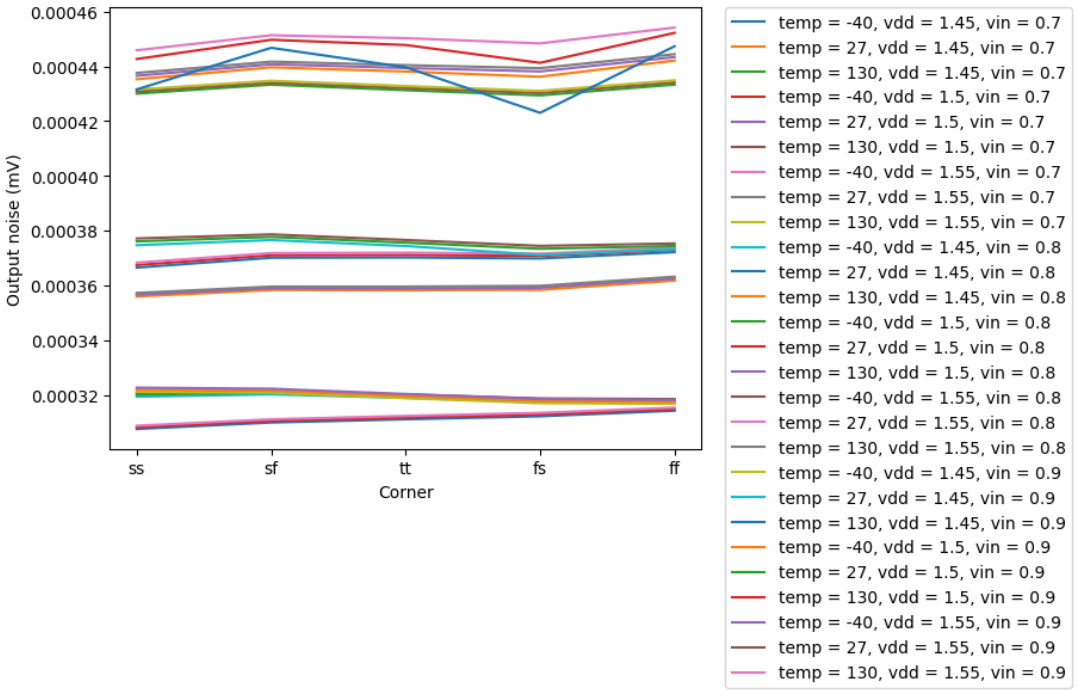


Figure 49: noise_vs_corner

settling_vs_temp

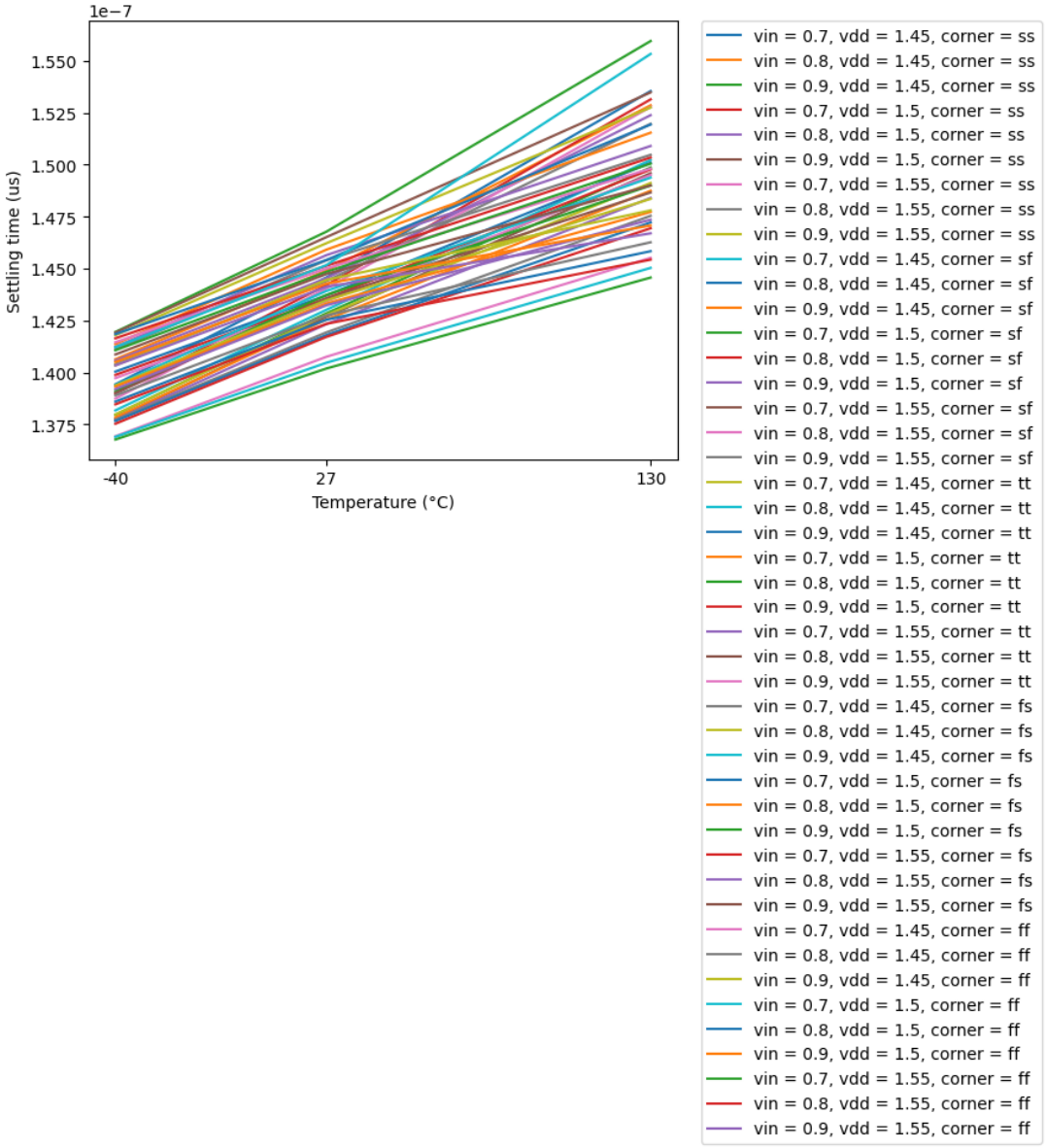


Figure 50: settling_vs_temp

settling_vs_vin

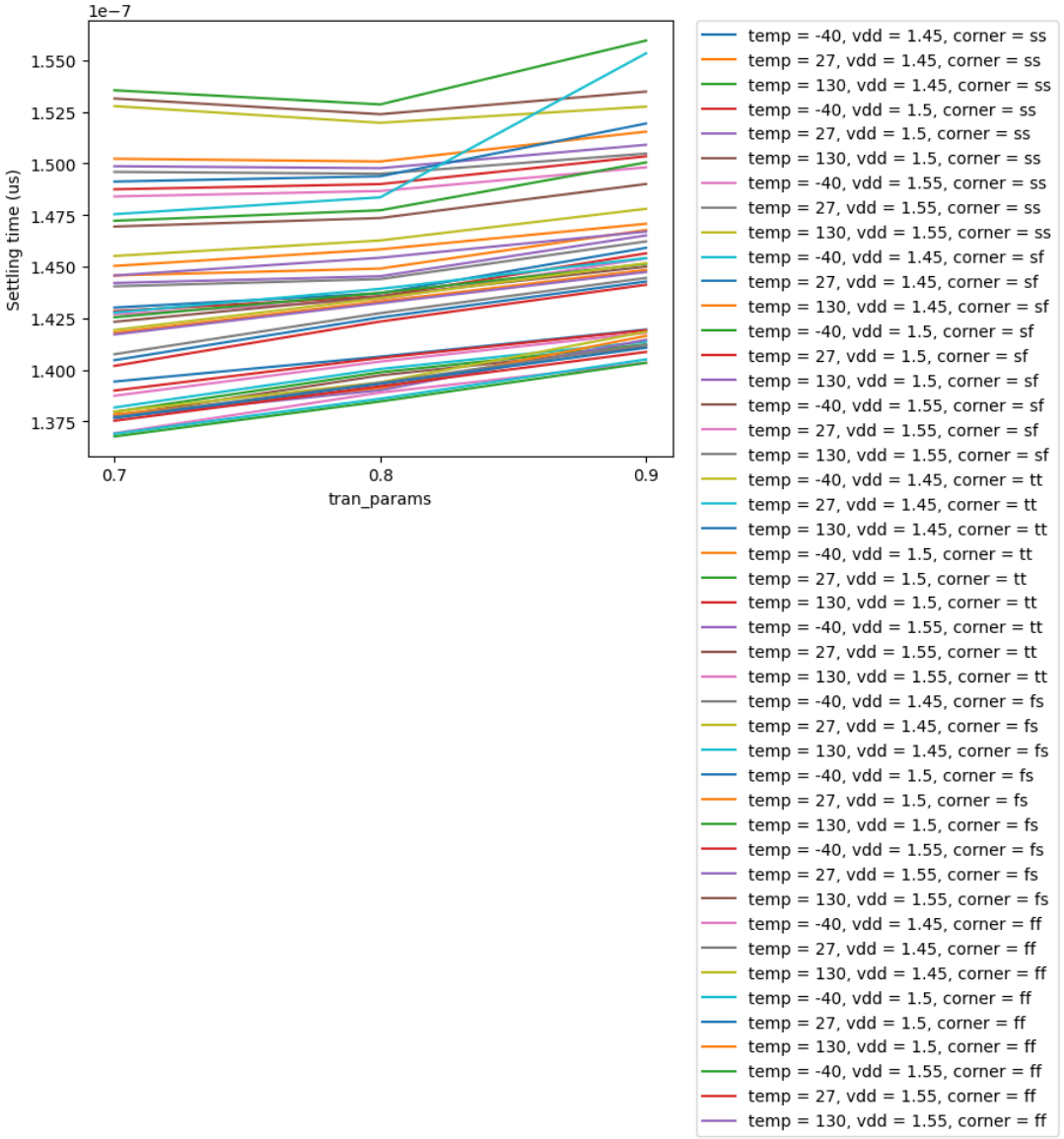


Figure 51: settling_vs_vin

settling_vs_vdd

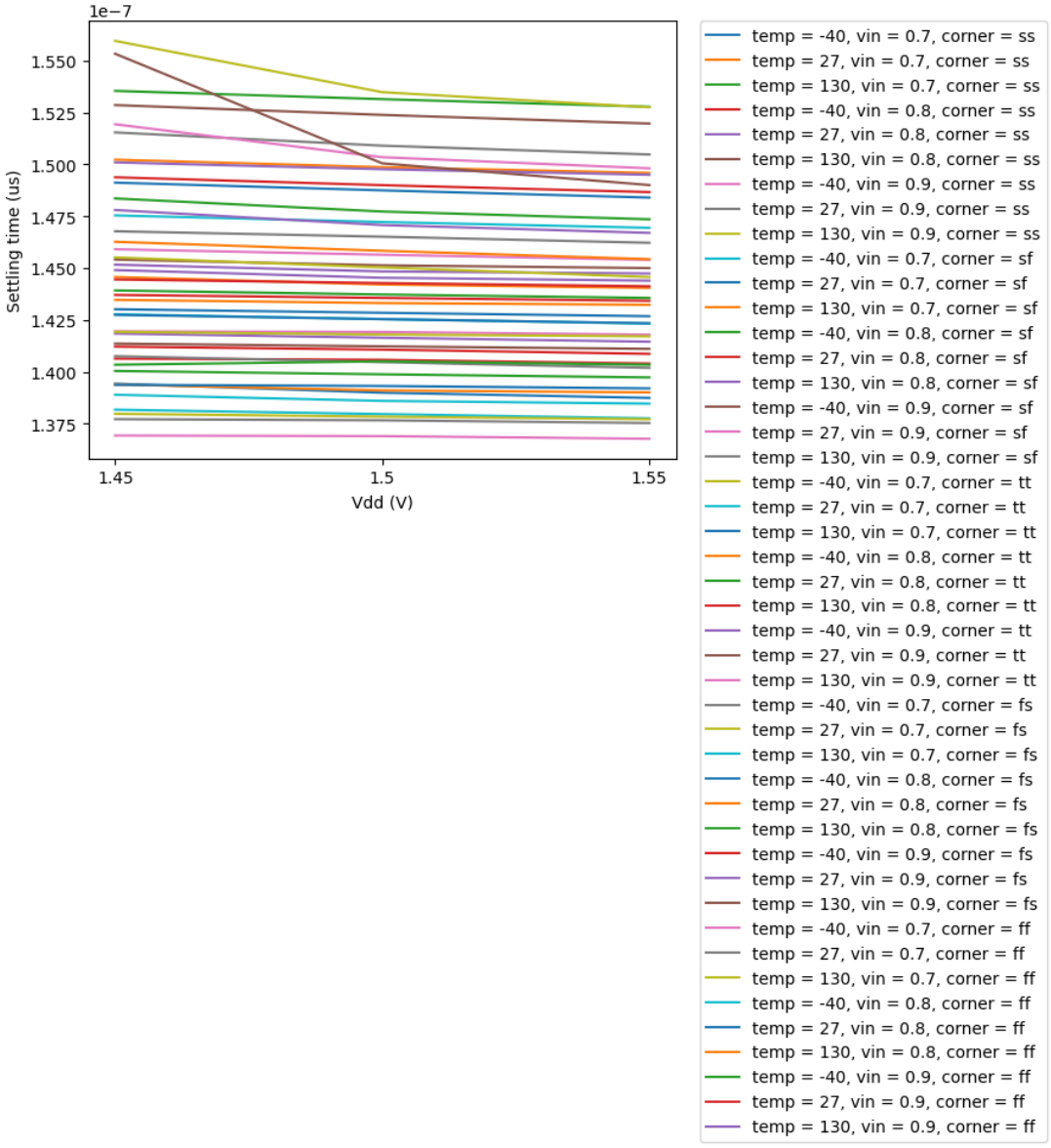


Figure 52: settling_vs_vdd

settling_vs_corner

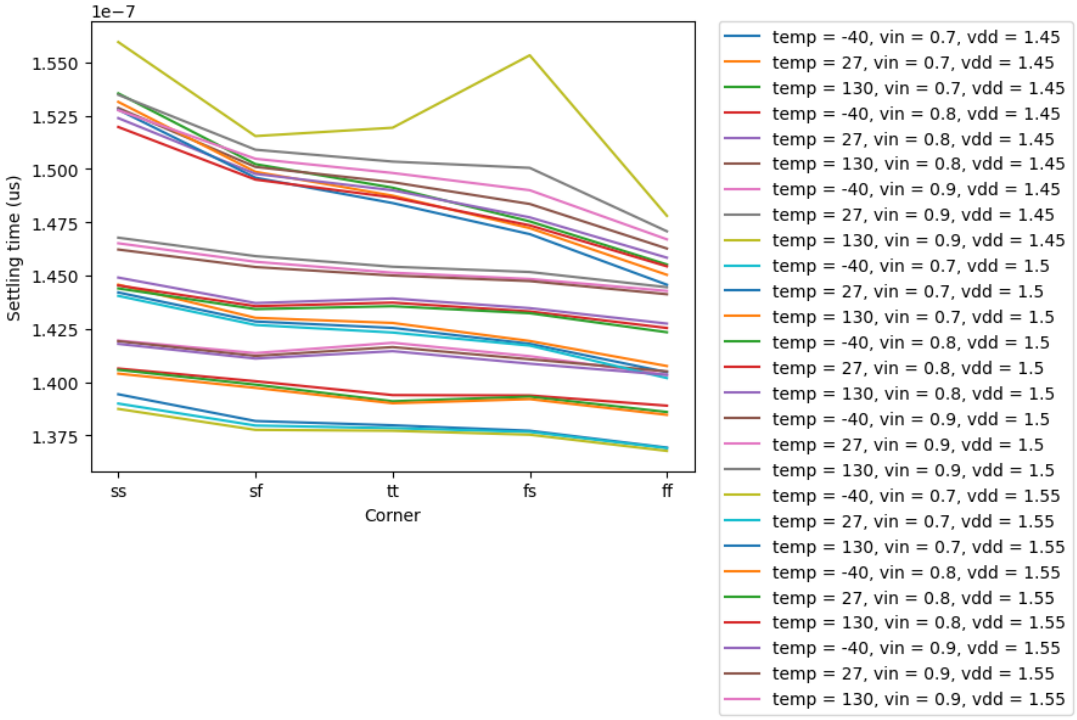


Figure 53: settling_vs_corner

8.0.1 PVT Simulation Analysis

Looking at the CACE report in Note 2 we see that (luckily) the specification is met for all parameters. This is great news! We now have a design that we carefully simulated across PVT and other corners, and which is ready for layout. Once we have the layout ready, we can extract the wiring parasitic (R and C) as well as other layout-dependent effects like well proximity. Using this augmented netlist we can then again use CACE to check performance across conditions and parameter variations, and if we still pass all specification points then our design is finished.

8.0.2 Monte Carlo Simulation Analysis

Looking at the CACE report in Note 2 we see that the output voltage specification is not met due to MOSFET mismatch! We have not considered transistor mismatch in the circuit sizing procedure, and as a result the selected MOSFET dimensions proof to be too small. We should now go back and change the transistor sizing (increasing L significantly while keeping the $g_\mathrm{m}/I_\mathrm{D}$ values). Likely we will find that some performance parameters are now

deteriorating due to the increased MOSFET dimensions, and we need to iterate until all performance metrics are met in the presence of transistor mismatch.

It is not unusual that the power consumption now increases, as we have to increase the size of the MOSFETs for matching, and these larger MOSFETs increase the parasitic capacitances which in turn lead to larger power consumption to keep the required bandwidth by increasing the g_m.

> Exercise: Re-Sizing of 5T-OTA for Mismatch
>
> Go back to Section 8.4 and repeat the sizing procedure of the 5T-OTA by increasing the L of the MOSFETs significantly. Focus first on the differential pair as it will likely have the biggest impact (see Equation 30). Then, tune the size of the output current mirror if necessary (see Equation 31).
> Once you are happy with the sizing result repeat the PVT simulations in CACE to confirm the performance of the voltage buffer including mismatch.

8.1 OTA Variants

Following, we are going to discuss two popular variants of the simple 5T-OTA. The first version provides an almost rail-to-rail single-ended output, and can thus be used for a variety of applications requiring this range. The arrangement is shown in Figure 54. It is a single-stage amplifier, as the only high-impedance point is at the output where significant voltage-gain is produced. The current mirrors can be used to scale up the current generated in the input differential pair to some extent in the output stage (or can be used to lower the current compared to the bias current in the diffpair); i.e., the bias currents of the input stage and the output stage can be set independently.

Being a single-stage amplifier stability is usually not an issue, as only the output node is high-ohmic; all other nodes feature a MOSFET diode so the node impedance is $\approx g_\mathrm{m}^{-1}$ and thus the according poles are located at high frequencies.

Another popular version is shown in Figure 55. Here, we have a two-stage amplifier able to provide higher voltage gain. The first stage (the diffpair loaded by a current mirror) is followed by a single-stage common-source amplifier with current-source load. Being a two-stage amplifier with two high-ohmic nodes, stability is a concern, so usually we need some form of compensation. Figure 55 shows a very simplistic Miller-compensation using C_M. Often, we would want to implement a more advanced scheme. Some examples can be found in (Baker, n.d.). An interesting technique is the indirect compensation with cascoded input differential pairs (see Figure 62) for higher power supply rejection. An advantage of this two-stage amplifier (compared with the simple 5T-OTA) is the dc-balanced load on top of the differential pair, as each side sees a voltage drop from V_DD of one V_GS (V_GS3 on the left side, V_GS5 on the right side).

Note that the circuit of Figure 55 can be easily modified into a low-dropout voltage regulator (LDO). This simple yet often effective circuit is shown in Figure 56. The pass transistor M_pass has to be sized according to the load current and the dropout voltage.

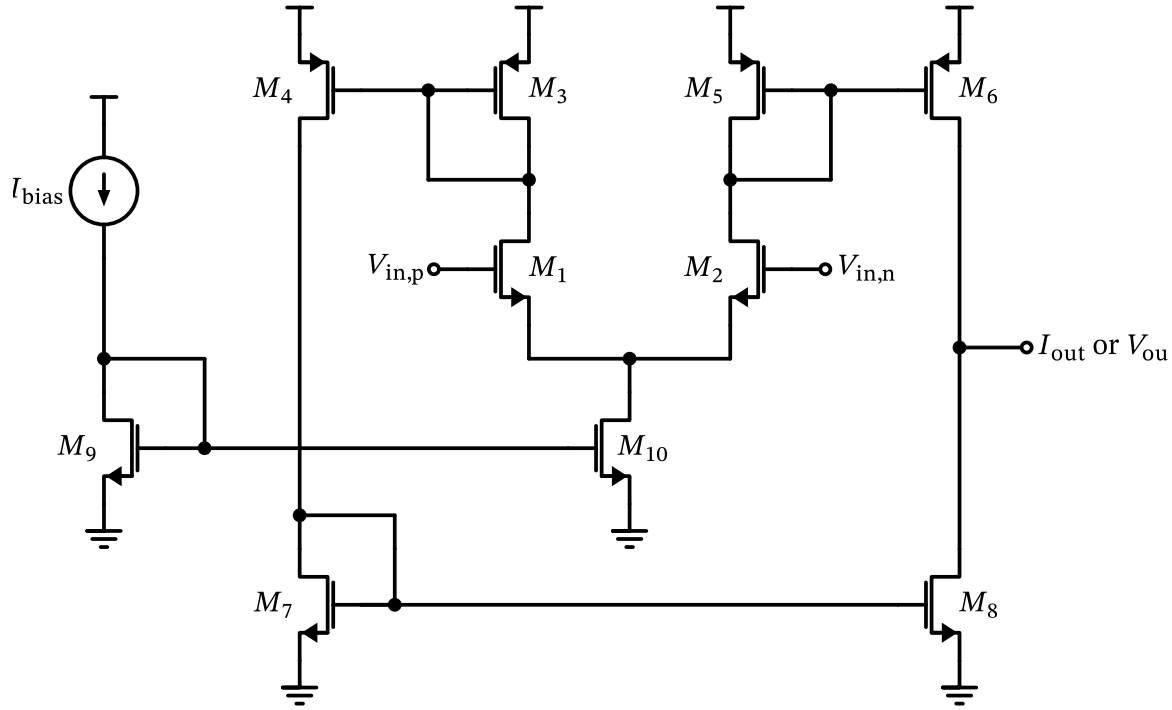


Figure 54: Single-ended OTA with rail-to-rail output stage.

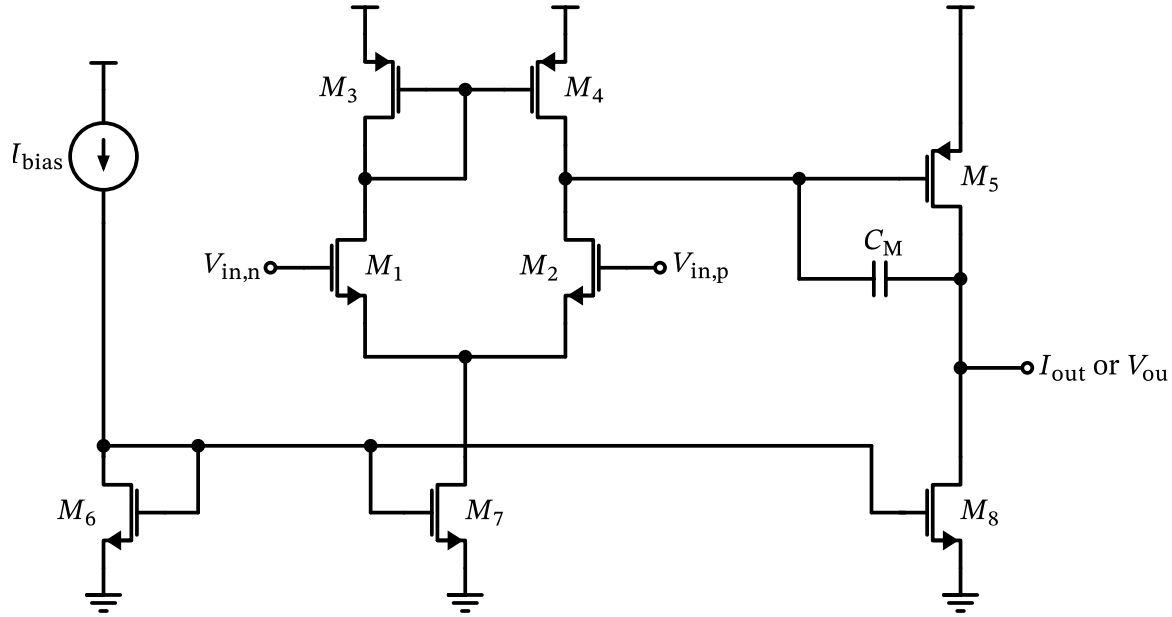


Figure 55: Single-ended two-stage OTA with rail-to-rail output.

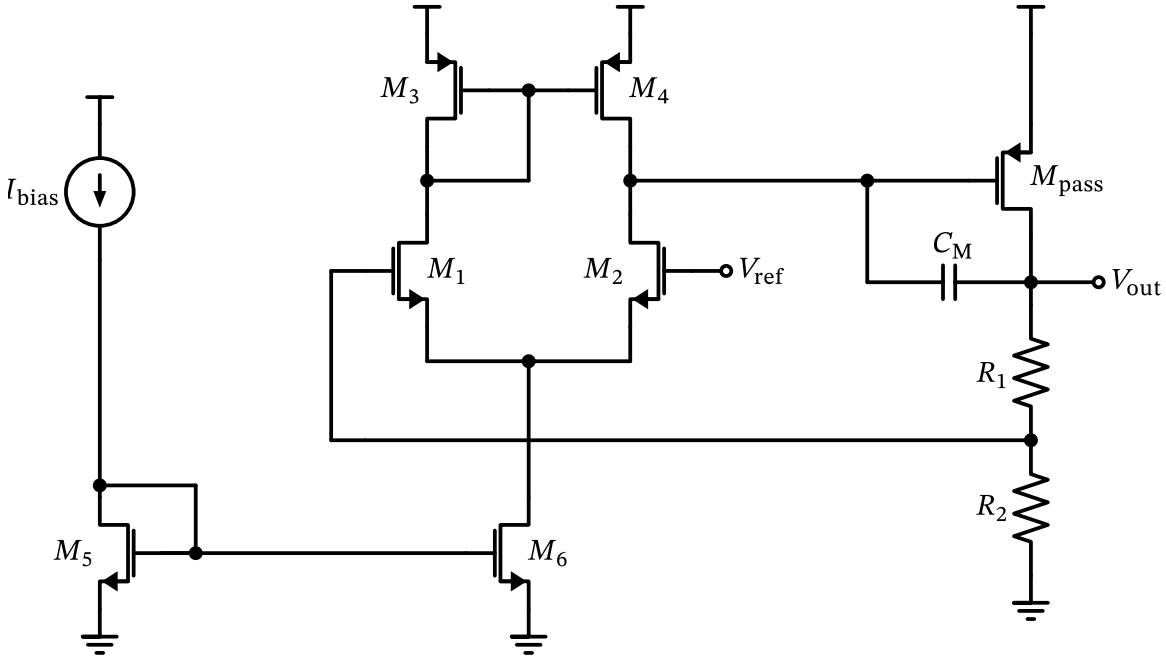


Figure 56: A basic low-dropout voltage regulator (LDO) with Miller compensation.

The reference voltage V_{ref} is scaled by R_1 and R_2, so that the output voltage V_{out} is given by

$$V_{\text{out}} \approx V_{\text{ref}} \left(1 + \frac{R_1}{R_2}\right)$$

if the gain of the OTA is sufficiently high. The quiescent current through $R_{1,2}$ establishes a minimum load current for the LDO, which is often good for stability. More information on LDOs can be found in (Razavi 2019).

9 Cascode Stage

As we have seen in Section 8 the performance of the OTA is generally quite acceptable (see Table 2), but we might want to aim for better output voltage accuracy. As our analysis has shown the output voltage tolerance is limited by the open-loop dc gain A_0 of the OTA (see Equation 14), which in turn is limited by the output conductance of M_2 and M_4 in Figure 29, which is also confirmed by the analytical result in Equation 18.

During the sizing procedure we have seen that the achievable $g_{\text{m}}/g_{\text{ds}}$ ratio of a single MOSFET is limited, even if we increase L. We are thus searching for a better option, and here (local) feedback in form of a **cascode** comes to help.

For analysis of a cascode, we use the following single-transistor stage shown in Figure 57.

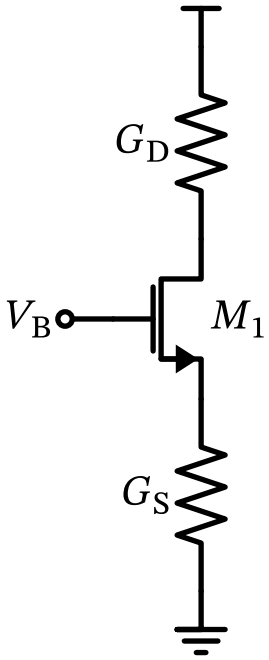


Figure 57: A MOSFET cascode circuit.

In order to derive the operation of the cascode analytically, we draw the small-signal equivalent circuit in Figure 58. We assume that V_B is a low-ohmic bias voltage, thus we replace it by ac ground. We further set $g_\mathrm{mb} = 0$.

Since the gate is assumed at a fixed potential, we can put C_gs in parallel to G_S as $G_\mathrm{S}^* = G_\mathrm{S} + sC_\mathrm{gs}$, and we can put C_gd in parallel to G_D as $G_\mathrm{D}^* = G_\mathrm{D} + sC_\mathrm{gd}$. As a result we will disregard these capacitors for now, and just consider G_S and G_D.

9.1 Cascode Output Impedance

As a first step, we want to calculate the output impedance at the drain of the MOSFET (i.e., looking into the drain). For this, we replace G_D with a current source. The resulting small-signal equivalent circuit is shown in Figure 60.

We realize that i_out flows through G_S and drops v_gs (note the sign):

$$v_\mathrm{gs} = -\frac{i_\mathrm{out}}{G_\mathrm{S}}$$

Further, $v_\mathrm{out} = -v_\mathrm{gs} + v_\mathrm{ds}$. Calculating KCL at the output node results in

$$i_\mathrm{out} - g_\mathrm{m} v_\mathrm{gs} - g_\mathrm{ds} v_\mathrm{ds} = 0.$$

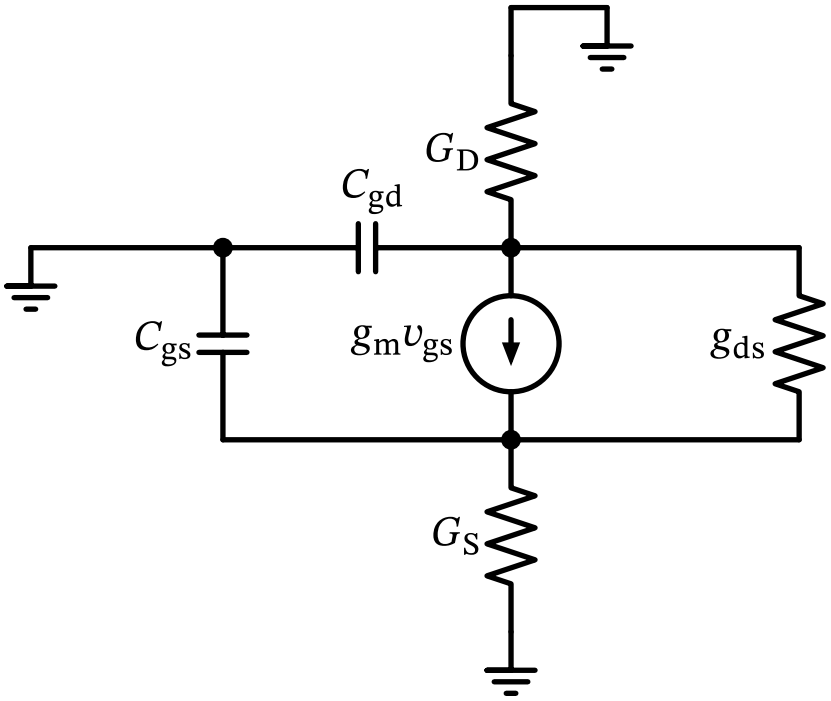


Figure 58: The MOSFET cascode small-signal model.

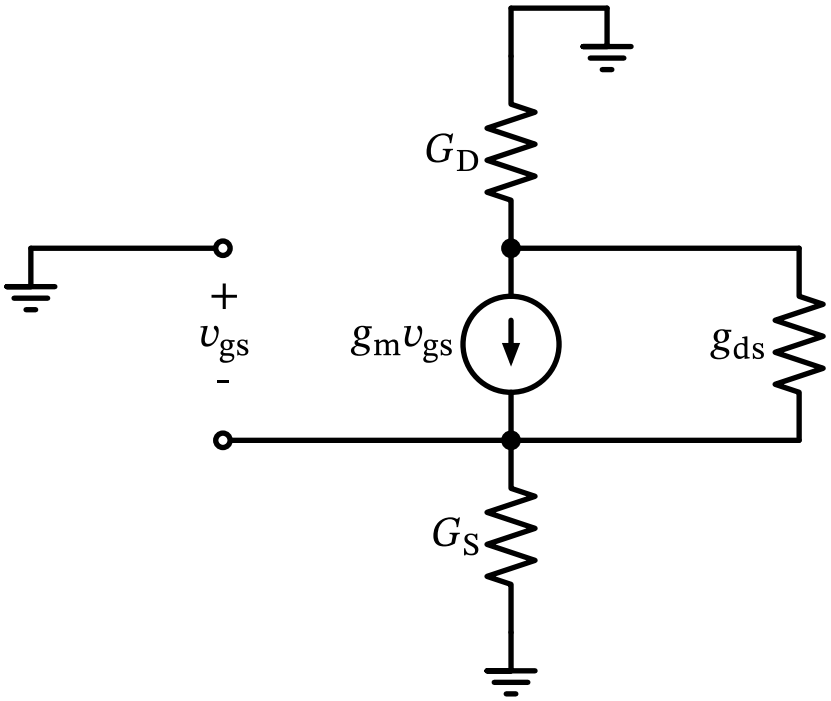


Figure 59: The simplified MOSFET cascode small-signal model.

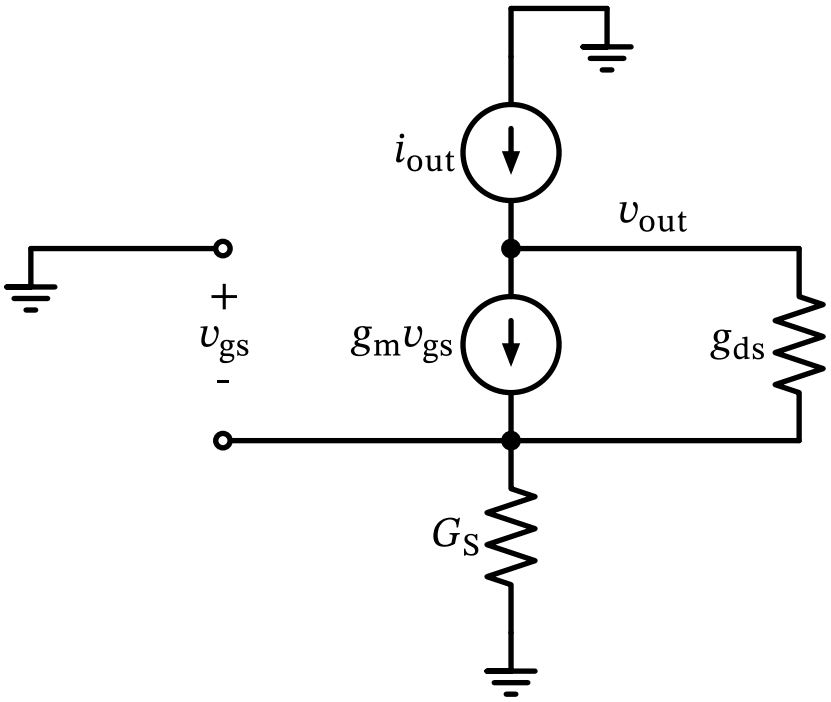


Figure 60: The simplified MOSFET cascode small-signal model for calculation of the output impedance.

Using the previously found identities, and after a bit of algebraic manipulations we arrive at

$$g_{out} = \frac{i_{out}}{v_{out}} = \frac{g_{ds}}{1 + \frac{g_m + g_{ds}}{G_S}} = \frac{g_{ds} \cdot G_S}{G_S + g_m + g_{ds}} \tag{33}$$

We find that if $G_S = 0$ (an open) then $g_{out} = 0$, and if $G_S = \infty$ (a short) then $g_{out} = g_{ds}$. We can calculate the benefits of a cascode if we assume we put a cascode on top of a common-source transistor stage (thus $G_S = g'_{ds}$) and get

$$g_{out} = \frac{g_{ds} \cdot g'_{ds}}{g'_{ds} + g_m + g_{ds}} \approx g'_{ds} \frac{g_{ds}}{g_m}. \tag{34}$$

! Benefit of Cascode (Output)

The output impedance of the lower MOSFET ($r_{ds} = 1/g'_{ds}$) is **increased** by the self-gain of the cascode transistor! This is a powerful technique to increase the output impedance of a transistor stage by cascoding, much better than increasing L.

9.2 Cascode Input Impedance

To calculate the input impedance of a cascode (i.e., looking into the source) we replace G_S with a current source. The resulting small-signal equivalent circuit is shown in Figure 61.

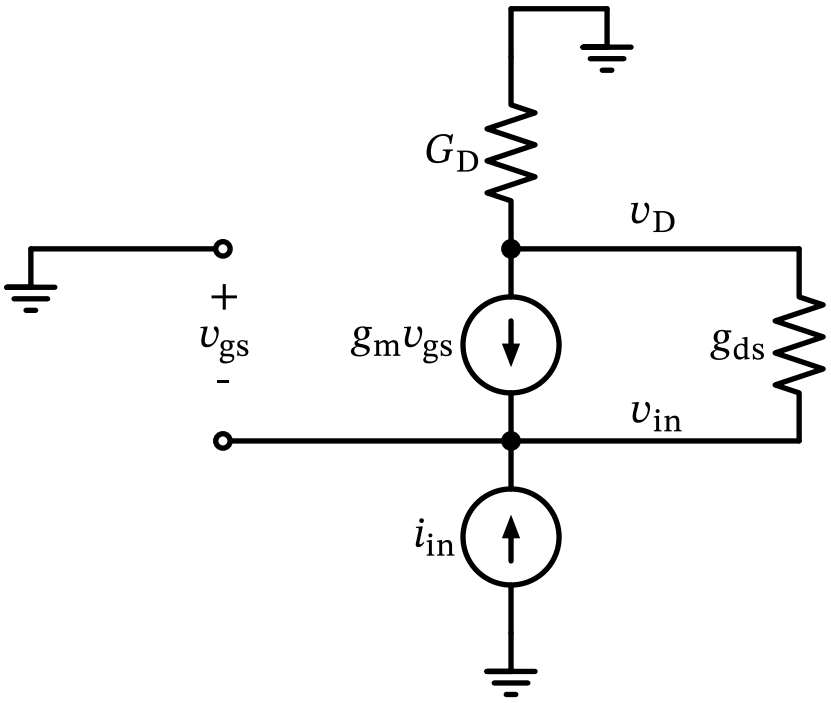


Figure 61: The simplified MOSFET cascode small-signal model for calculation of the input impedance.

We note that $v_{gs} = -v_{in}$ and that i_{in} flows through G_D, resulting in $v_D = i_{in}/G_D$. Note that $v_{ds} = v_D - v_{in}$. Formulating KCL at the input node results in

$$i_{in} + g_{ds}v_{ds} + g_m v_{gs} = 0.$$

After some manipulation we find that

$$g_{in} = \frac{i_{in}}{v_{in}} = \frac{(g_m + g_{ds}) \cdot G_D}{g_{ds} + G_D}. \tag{35}$$

Setting $G_D = 0$ (an open) results in $g_{in} = 0$ as well, so the input impedance of the cascode is very large when the drain impedance is large.

However, setting $G_D = \infty$ (a short or low-ohmic impedance) results in the well-known result of $g_{in} = g_m + g_{ds} \approx g_m$, which means that the input impedance looking into a cascode is approximately $1/g_m$.

! Benefit of Cascode (Input)

This has the practical benefit that a capacitance connected at this node results in a high-frequency pole, which is often not critical in terms of stability. Further, the voltage swing at a cascode input node is small due to the often small impedance, and this minimizes the Miller effect at connected inter-node capacitors (see Section 16.1).

10 Improved OTA

With the new learned know-how of the cascode stage we can set out to improve our original basic 5T-OTA design. Essentially this means to add cascodes to M_2 and M_4 in Figure 29. For symmetry reasons we will add cascodes to both sides, and the resulting schematic is shown in Figure 62.

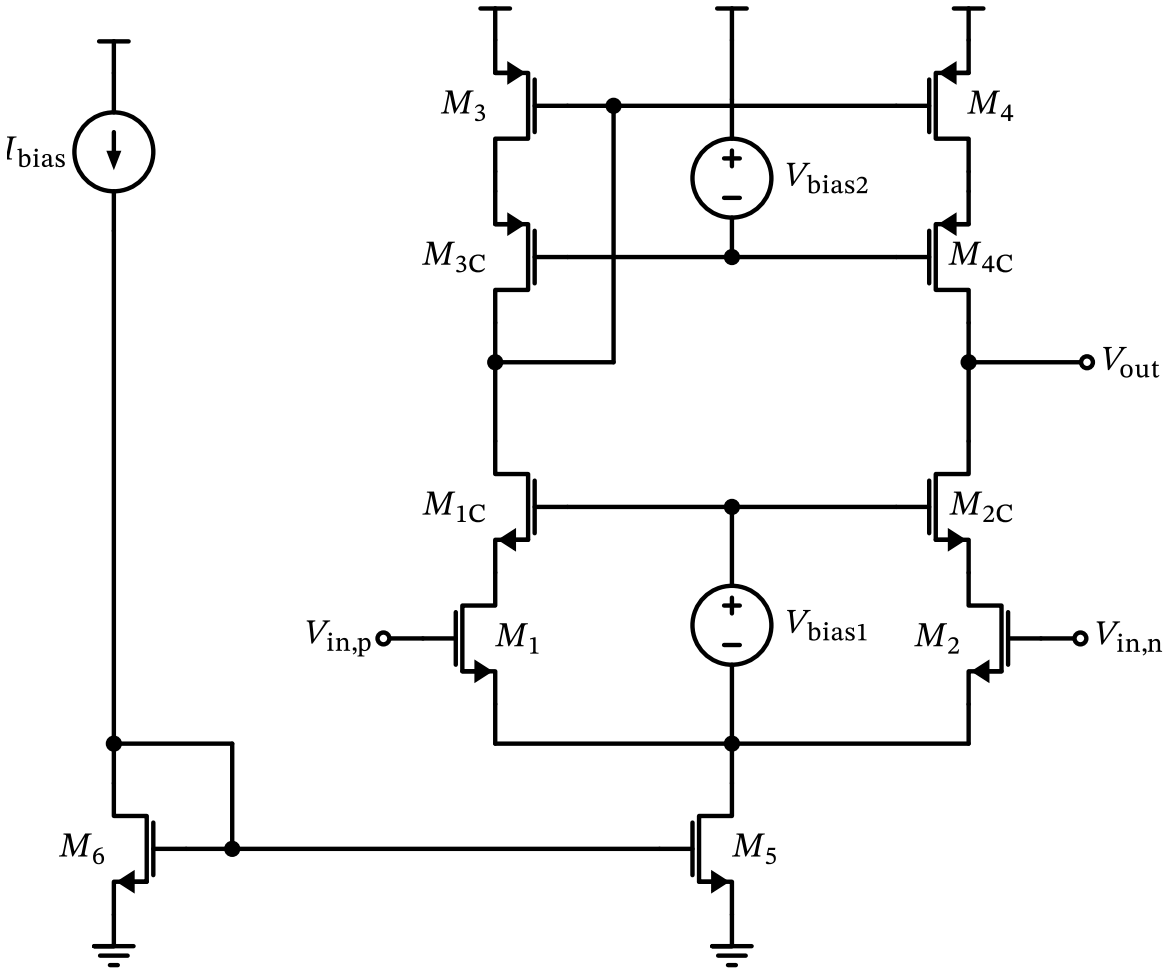


Figure 62: The improved OTA based on the 5T-OTA design.

The transistor name appendix "C" indicates a cascode device sitting atop its base transistor. The bias voltage V_{bias2} is referenced to V_{DD} (it is shown differently in Figure 62 to simplify the schematic), and the floating bias voltage V_{bias1} creates a voltage bias for $M_{1\text{C}}$ and $M_{2\text{C}}$ relative to the tail point, so that the V_{DS} of $M_{1,2}$ stays constant with a changing common-mode input voltage.

> **!** Cascode Bias Voltage Generation
>
> It is critically import for a stable performance across PVT that the bias voltages for the cascode gates are created in a manner that tracks variations with process, temperature, and supply voltage!

The current mirror constructed out of $M_{3,3\text{C}}$ and $M_{4,4\text{C}}$ is a special kind of **cascode current mirror for low-voltage operation**, also referred to as high-swing cascode current mirror (Jespers and Murmann 2017). This type is very often used, as it forces the V_{GS} and V_{DS} of $M_{3,4}$ to be equal, so the current mirror ratio is independent of g_{ds}.

> Exercise: Cascode Current Mirror vs. High-Swing Cascode Current Mirror
>
> Try to verify the above statement of equal drain-source voltages by deriving both, an equation for V_{DS4} assuming a high-swing cascode current mirror (Figure 62) and V_{DS4} in case of a simple cascode current mirror, where the reference branch ($M_{3,3\text{C}}$) is comprised of two MOSFET diodes.

Further, by properly selecting the bias voltages of the cascode a low-voltage operation is achieved as V_{DS} can be minimized, allowing even triode operation of the current-mirror MOSFETs (as, noted above, a large g_{ds} is not a big issue).

A simplified small-signal gain calculation of this improved OTA uses the result of Equation 18 and Equation 34 to arrive at the approximate dc gain of

$$A_0 \approx \frac{g_{\text{m12}}}{g_{\text{ds2}}\frac{g_{\text{ds2C}}}{g_{\text{m2C}}} + g_{\text{ds4}}\frac{g_{\text{ds4C}}}{g_{\text{m4C}}}} \tag{36}$$

leading to a significant boost in dc gain due to cascoding. We will use this increased gain to reduce the L of all MOSFET to

1. save area (a smaller L will lead to a smaller W for a given W/L ratio) and
2. push the additional poles and zeros at the inner nodes of the cascode transistors (e.g., the connection of the drain of M_4 to the source of $M_{4\text{C}}$) to higher frequencies to result in stable behavior and a reasonable gain transfer function (too many poles and zeros in the pass band of the amplifier create many issues with stability margin).

10.1 Sizing the Improved OTA

Like the sizing of the 5T-OTA in Section 8.4 we will again use the $g_{\text{m}}/I_{\text{D}}$ method using a Python notebook. Instead of using $L = 5\,\mu\text{m}$ we will this time use a reduced $L = 0.5\,\mu\text{m}$

for $M_{1/1C,2/2C,3/3C,4/4C}$ (for speed reasons) and $L = 5\,\mu$m for $M_{5,6}$ for better common-mode rejection (the tail current mirror is less critical in terms of speed and stability).

We set $g_m/I_D = 13$ across the board for a good trade-off between speed, current efficiency, and voltage headroom for the MOSFETs (this is now way more critical than in the basic 5T-OTA as we stack now double as many MOSFET at the same supply voltage). Please look at Section 3 to confirm this choice.

Improved OTA Sizing

Sizing for Basic (Improved) OTA


```
# read table data
from pygmid import Lookup as lk
import numpy as np
lv_nmos = lk('sg13_lv_nmos.mat')
lv_pmos = lk('sg13_lv_pmos.mat')
# list of parameters: VGS, VDS, VSB, L, W, NFING, ID, VT, GM, GMB, GDS, CGG, CGB, CGD, C
# if not specified, minimum L, VDS=max(vgs)/2=0.9 and VSB=0 are used
```

```
# define the given parameters as taken from the specification table or initial guesses
c_load = 50e-15
gm_id_m12 = 13
gm_id_m12c = 13
gm_id_m34 = 13
gm_id_m34c = 13
gm_id_m56 = 13
l_12 = 0.5
l_12c = 0.5
l_34 = 0.5
l_34c = 0.5
l_56 = 5
f_bw = 10e6 # -3dB bandwidth of the voltage buffer
i_total_limit = 10e-6 # we plan 2x5uA in addition for additional bias voltage generation
i_bias_in = 5e-6
output_voltage = 1.3
vin_min = 0.7
vin_max = 0.9
vdd_min = 1.45
vdd_max = 1.55
vds_headroom = 0.2
```

```
# we get the required gm of M1/2 from the -3dB bandwidth requirement of the voltage buff
# note that the -3dB bandwidth of the voltage buffer with gain Av=1 is equal to the unit
# of the ota, hence we wet them equal here
# we add a factor of 3 to allow for PVT variation plus additional MOSFET parasitic loadi
# we also add an additional factor of 2 to get more dc gain (and there is power still in
gm_m12 = f_bw * 3 * 4*np.pi*c_load * 3
print('gm12 =', round(gm_m12/1e-3, 4), 'mS')
```

```
gm12 = 0.0565 mS
```

```
# since we know gm12 and the gmid we can calculate the bias current
id_m12 = gm_m12 / gm_id_m12
i_total = 2*id_m12
print('i_total (exact) =', round(i_total/1e-6, 1), 'µA')
# we round to 0.5µA bias currents
i_total = max(round(i_total / 1e-6 * 2) / 2 * 1e-6, 0.5e-6)
# here is a manual override to set the current; we keep a reserve of 2µA for bias branch
i_total = 8e-6
id_m12 = i_total/2

print('i_total (rounded) =', i_total/1e-6, 'µA')
if i_total < i_total_limit:
    print('[info] power consumption target is met!')
else:
    print('[info] power consumption target is NOT met!')
```

```
i_total (exact) = 8.7 µA
i_total (rounded) = 8.0 µA
[info] power consumption target is met!
```

```
# we calculate the dc gain
gm_gds_m12 = lv_nmos.lookup('GM_GDS', GM_ID=gm_id_m12, L=l_12, VDS=vds_headroom, VSB=2*v
gm_gds_m12c = lv_nmos.lookup('GM_GDS', GM_ID=gm_id_m12c, L=l_12c, VDS=vds_headroom, VSB=
gm_gds_m34 = lv_pmos.lookup('GM_GDS', GM_ID=gm_id_m34, L=l_34, VDS=vds_headroom, VSB=0)
gm_gds_m34c = lv_pmos.lookup('GM_GDS', GM_ID=gm_id_m34c, L=l_34c, VDS=vds_headroom, VSB=
# conductance of lower cascoded differential pair
gds_m12 = gm_m12 / gm_gds_m12
gds_m12_casc = gds_m12 / gm_gds_m12c
# conductance of upper cascoded current mirror
gm_m34 = gm_id_m34 * i_total/2
gds_m34 = gm_m34 / gm_gds_m34
gds_m34_casc = gds_m34 / gm_gds_m34c

print('gds_12 =', round(gds_m12/1e-6, 3), 'µs')
print('gm_12c/gds_12c =',round(float(gm_gds_m12c), 1))
print('gds_34 =', round(gds_m34/1e-6, 3), 'µs')
print('gm_34c/gds_34c =', round(float(gm_gds_m34c), 1))

a0 = gm_m12 / (gds_m12_casc + gds_m34_casc)
print('a0 =', round(20*np.log10(a0), 1), 'dB')
```

```
gds_12 = 4.026 µs
gm_12c/gds_12c = 13.4
gds_34 = 2.031 µs
gm_34c/gds_34c = 24.9
a0 = 43.4 dB
```

```
# we calculate the MOSFET capacitance which adds to Cload, to see the impact on the BW
gm_cgs_m12 = lv_nmos.lookup('GM_CGS', GM_ID=gm_id_m12, L=l_12, VDS=vds_headroom, VSB=2*v
gm_cdd_m12c = lv_nmos.lookup('GM_CDD', GM_ID=gm_id_m12c, L=l_12c, VDS=vds_headroom, VSB=
gm_cdd_m34c = lv_pmos.lookup('GM_CDD', GM_ID=gm_id_m34c, L=l_34c, VDS=vds_headroom, VSB=

c_load_parasitic = abs(gm_m12/gm_cgs_m12) + abs(gm_m12/gm_cdd_m12c) + abs(gm_m34/gm_cdd_
print('additional load capacitance =', round(c_load_parasitic/1e-15, 1), 'fF')

f_bw = gm_m12 / (4*np.pi * (c_load + c_load_parasitic))
print('unity gain bandwidth incl. parasitics =', round(f_bw/1e6, 2), 'MHz')
```

```
additional load capacitance = 5.5 fF
unity gain bandwidth incl. parasitics = 81.15 MHz
```

```
# we can now look up the VGS of the MOSFET
vgs_m12 = lv_nmos.look_upVGS(GM_ID=gm_id_m12, L=l_12, VDS=vds_headroom, VSB=2*vds_headro
vgs_m12c = lv_nmos.look_upVGS(GM_ID=gm_id_m12c, L=l_12c, VDS=vds_headroom, VSB=3*vds_hea
vgs_m34 = lv_pmos.look_upVGS(GM_ID=gm_id_m34, L=l_34, VDS=vds_headroom, VSB=0.0)
vgs_m34c = lv_pmos.look_upVGS(GM_ID=gm_id_m34c, L=l_34c, VDS=vds_headroom, VSB=vds_headr
vgs_m56 = lv_nmos.look_upVGS(GM_ID=gm_id_m56, L=l_56, VDS=vds_headroom, VSB=0.0)

print('vgs_12  =', round(float(vgs_m12), 3), 'V')
print('vgs_12c =', round(float(vgs_m12c), 3), 'V')
print('vgs_34  =', round(float(vgs_m34), 3), 'V')
print('vgs_34c =', round(float(vgs_m34c), 3), 'V')
print('vgs_56  =', round(float(vgs_m56), 3), 'V')
```

```
vgs_12  = 0.436 V
vgs_12c = 0.458 V
vgs_34  = 0.475 V
vgs_34c = 0.512 V
vgs_56  = 0.318 V
```

```
# calculate settling time due to slewing with the calculated bias current
t_slew = (c_load + c_load_parasitic) * output_voltage / i_total
print('slewing time  =', round(t_slew/1e-6, 3), 'µs')
t_settle = 5/(2*np.pi*f_bw)
print('settling time =', round(t_settle/1e-6, 3), 'µs')
```

```
slewing time  = 0.009 µs
settling time = 0.01 µs
```

```
# calculate voltage gain error
gain_error = a0 / (1 + a0)
print('voltage gain error =', round((gain_error-1)*100, 1), '%')
```

```
voltage gain error = -0.7 %
```

```
# calculate total rms output noise
sth_m12 = lv_nmos.lookup('STH_GM', VGS=vgs_m12, L=l_12, VDS=vds_headroom, VSB=2*vds_head
gamma_m12 = sth_m12/(4*1.38e-23*300*gm_m12)

sth_m34 = lv_pmos.lookup('STH_GM', VGS=vgs_m34, L=l_34, VDS=vds_headroom, VSB=0) * gm_m3
gamma_m34 = sth_m34/(4*1.38e-23*300*gm_m34)

output_noise_rms = np.sqrt(1.38e-23*300 / (c_load + c_load_parasitic) * (2*gamma_m12 + 2
print('output noise =', round(output_noise_rms/1e-6, 1), 'µVrms')
```

```
output noise = 555.4 µVrms
```

```
# calculate all widths
id_w_m12 = lv_nmos.lookup('ID_W', GM_ID=gm_id_m12, L=l_12, VDS=vds_headroom, VSB=2*vds_h
w_12 = id_m12 / id_w_m12
w_12_round = max(round(w_12*2)/2, 0.5)
print('M1/2  W =', round(w_12, 2), 'um, rounded W =', w_12_round, 'um')

id_m12c = id_m12
id_w_m12c = lv_nmos.lookup('ID_W', GM_ID=gm_id_m12c, L=l_12c, VDS=vds_headroom, VSB=3*vd
w_12c = id_m12c / id_w_m12c
w_12c_round = max(round(w_12c*2)/2, 0.5)
print('M1/2c W =', round(w_12c, 2), 'um, rounded W =', w_12c_round, 'um')

id_m34 = id_m12
id_w_m34 = lv_pmos.lookup('ID_W', GM_ID=gm_id_m34, L=l_34, VDS=vds_headroom, VSB=0)
w_34 = id_m34 / id_w_m34
w_34_round = max(round(w_34*2)/2, 0.5)
print('M3/4  W =', round(w_34, 2), 'um, rounded W =', w_34_round, 'um')

id_m34c = id_m12
id_w_m34c = lv_pmos.lookup('ID_W', GM_ID=gm_id_m34c, L=l_34c, VDS=vds_headroom, VSB=vds_
w_34c = id_m34c / id_w_m34c
w_34c_round = max(round(w_34c*2)/2, 0.5)
print('M3/4c W =', round(w_34c, 2), 'um, rounded W =', w_34c_round, 'um')

id_w_m5 = lv_nmos.lookup('ID_W', GM_ID=gm_id_m56, L=l_56, VDS=vds_headroom, VSB=0)
w_5 = i_total / id_w_m5
w_5_round = max(round(w_5*2)/2, 0.5)
print('M5    W =', round(w_5, 2), 'um, rounded W =', w_5_round, 'um')

w_6 = w_5_round * i_bias_in / i_total
print('M6    W =', round(w_6, 2), 'um')
```

```
M1/2  W = 0.83 um, rounded W = 1.0 um
M1/2c W = 0.8 um, rounded W = 1.0 um
M3/4  W = 3.28 um, rounded W = 3.5 um
M3/4c W = 2.99 um, rounded W = 3.0 um
M5    W = 14.2 um, rounded W = 14.0 um
M6    W = 8.75 um
```

```
# Print out final design values
print('Improved OTA dimensioning:')
print('--------------------------')
print('M1/2  W=', w_12_round, ', L=', l_12)
print('M1/2c W=', w_12c_round, ', L=', l_12c)
print('M3/4  W=', w_34_round, ', L=', l_34)
print('M3/4c W=', w_34c_round, ', L=', l_34c)
print('M5   W=', w_5_round, ', L=', l_56)
print('M6   W=', round(w_6, 2), ', L=', l_56)
print()
print('Improved OTA performance summary:')
print('---------------------------------')
print('supply current =', round(i_total/1e-6, 1), 'µA')
print('output noise =', round(output_noise_rms/1e-6, 1), 'µVrms')
print('voltage gain error =', round((gain_error-1)*100, 1), '%')
print('unity gain bandwidth incl. parasitics =', round(f_bw/1e6, 2), 'MHz')
print('turn-on time (slewing+settling) =', round((t_slew+t_settle)/1e-6, 3), 'µs')
print()
print('Improved OTA bias point check:')
print('------------------------------')
print('headroom M1+M1c =', round(vdd_min-vgs_m34+vgs_m12-vin_max, 3), 'V')
print('headroom M4+M4c =', round(vdd_min-vin_max, 3), 'V')
print('headroom M5     =', round(vin_min-vgs_m12, 3), 'V')
```

```
Improved OTA dimensioning:
--------------------------
M1/2  W= 1.0 , L= 0.5
M1/2c W= 1.0 , L= 0.5
M3/4  W= 3.5 , L= 0.5
M3/4c W= 3.0 , L= 0.5
M5   W= 14.0 , L= 5
M6   W= 8.75 , L= 5

Improved OTA performance summary:
---------------------------------
supply current = 8.0 µA
output noise = 555.4 µVrms
voltage gain error = -0.7 %
unity gain bandwidth incl. parasitics = 81.15 MHz
turn-on time (slewing+settling) = 0.019 µs

Improved OTA bias point check:
------------------------------
headroom M1+M1c = 0.512 V
```

```
headroom M4+M4c = 0.55 V
headroom M5     = 0.264 V
```

Looking at this sizing result we see that we achieve an improved $A0 > 43\,\text{dB}$ while meeting also the other performance requirements of Table 2 with margin. In addition, we check the voltage headroom of the critical MOSFET to see if we can squeeze it into the available supply voltage range, and see that this is possible with our above choice selection of parameters.

> **Exercise: Improved OTA Sizing**
>
> Please take a detailed look at the above sizing notebook and play with the numbers and calculations. Do you find a better trade-off for the input parameters? Can you understand the thinking process behind the choices and calculations?

10.1 Designing the Improved OTA

Based on the collected experience in this lecture and the result of the sizing procedure in Section 10.1 you should be able to design this OTA. If you want, please go ahead and try an implementation and check its performance with CACE.

As an alternative there is a prepared OTA design shown in Figure 63 which we will discuss in detail next.

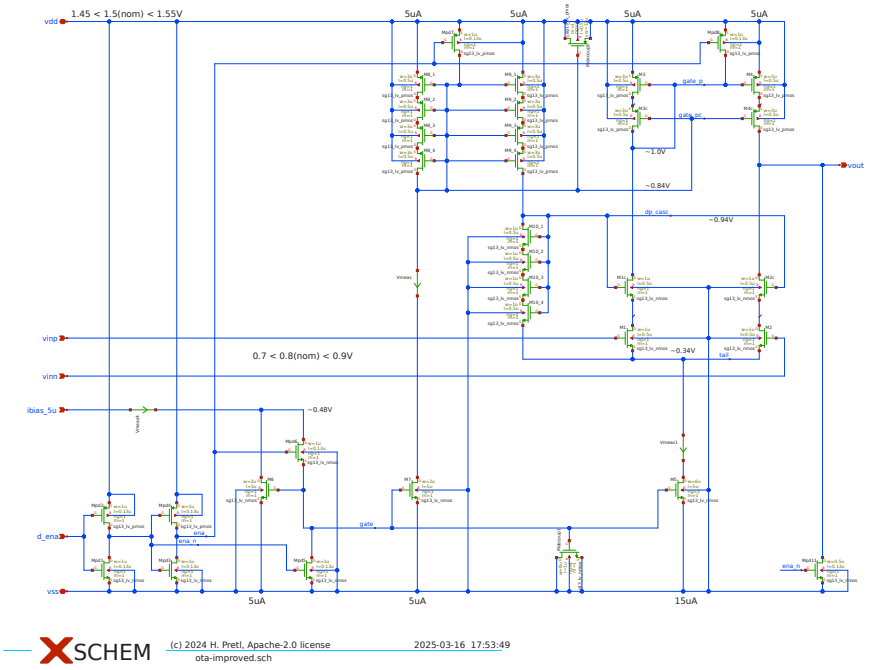


Figure 63: Improved OTA design in Xschem.

10.1.1 Discussion of the OTA Design

We will now do an analysis of the circuit design of the OTA including all the complications which make this design practical.

1. For easier navigation, the device identifier are consistent with the circuit sketch in Figure 62.
2. Some MOSFET dimensions are rounded to make a better fit in the IC layout. Please also look carefully at W, L, and ng. The parameter ng defines how the total W of a MOSFET should be split into individual MOSFET fingers with $W_\mathrm{f} = W/\mathrm{ng}$. This is done to arrive at a suitably sized MOSFET physical implementation. As we will not deal with IC layout in this lecture we will leave it at that.
3. In order to allow good matching in the IC layout, MOSFETs (and other components) have to be constructed from equal pieces. To that end, W/L scaling is done using unit elements (see finger width W_f). Sometimes, besides W the length L has to be scaled, and this leads to the oddly-looking series stacking of some MOSFET (easily recognizable by the connected gates). In order to increase circuit readability, a subcircuit could be constructed hiding this series stacking of MOSFET, but it is sometimes easier to avoid subcircuits. There is a fine line in this trade-off, sometime a depth of 4 is the decision point between subcircuit use/no-use.
4. As you can (hopefully) see the circuit is carefully drawn to ease readability. Important nets are named, text comments state certain properties like nominal voltage levels, bias currents, etc. Current sensing elements are added to directly see the dc currents in the circuit simulation.
5. The bias voltage generation for the cascodes is included as well. The voltage drop for the bottom transistors is developed by properly scaling the MOSFETs in the reference branch. We reduce the W/L ratio to increase the V_GS to create a voltage headroom for the bottom MOSFET. We are using a dummy branch for bias generation (constructed with M_{7-10}).
6. The floating bias voltage V_bias1 is created by implementing a current source from V_DD (M_9), then a MOSFET diode M_{10}, and an increased current towards V_SS through M_5.
7. Power-down transistors $M_{\mathrm{pd},x}$ are added to allow a proper shutdown of the circuit with a digital enable input. It is generally a good idea to clamp floating nodes in off-mode so that no issues during power-down (like increased leakage currents) or delayed startup or shutdown are occurring. It is further a good design principle to buffer all incoming digital signals with inverters ($M_{\mathrm{pd},1-4}$) connected to the local supply. This lowers the risk of unwanted noise coupling or excessive slew rates on the incoming digital signals.
8. Sensitive bias nodes are buffered with decoupling capacitors. We are using MOSFETs as nonlinear capacitors, which is not an issue in this application, but we value the increased capacitive density. Please note how the MOSFET are connected (some are tied to V_DD while others are tied to V_SS).

> **Parallel Connection**
>
> Note that a parallel connection of devices is effectively possible using the multiplier notation of Xschem.

10.2 Simulation of Improved OTA

Now that the circuit design of the improved OTA is done, we can use the same simulation testbenches as for the basic OTA. The testbenches are shown in Figure 64, Figure 65, and Figure 66.

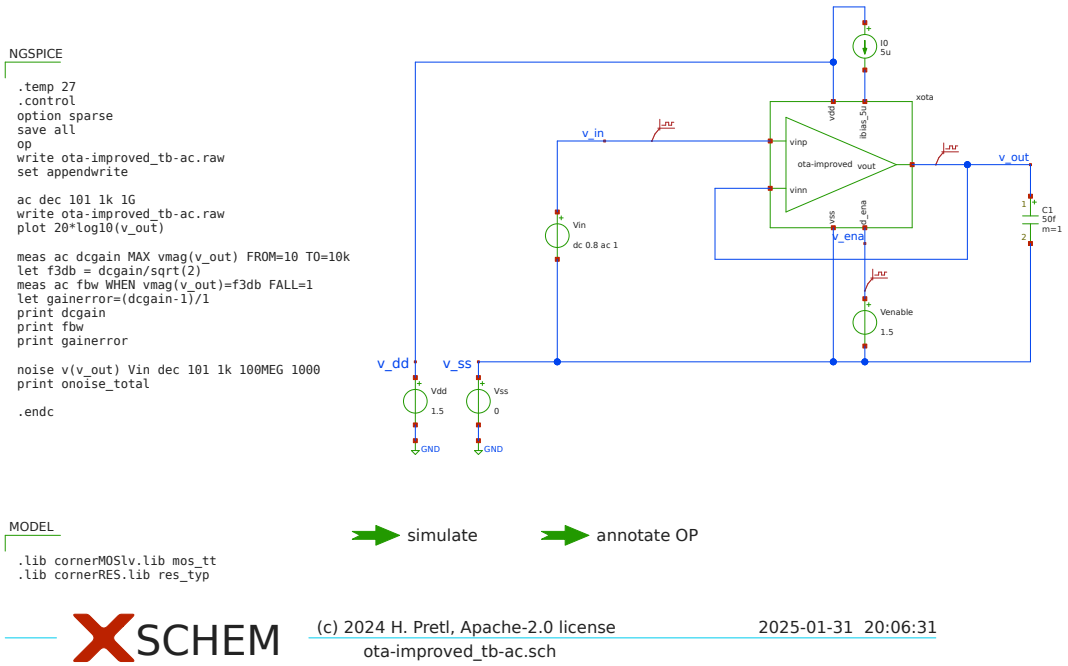


Figure 64: Simulation testbench of the improved OTA design (small-signal).

> **Exercise: Improved OTA Initial Simulation**
>
> Please use the above testbenches to simulate the improved OTA:
>
> 1. Check the dc bias points. Are they good? How stable are they across PVT variations?
> 2. What are the small-signal parameters like gain, noise and bandwidth? Are they fitting the specification?
> 3. What is large-signal performance? Is the settling fast enough? Is the settling well behaved, i.e., are there overshoots or other strange ringing indicating potential stability issues?
> 4. Try to improve the design. Change various device parameters and see what

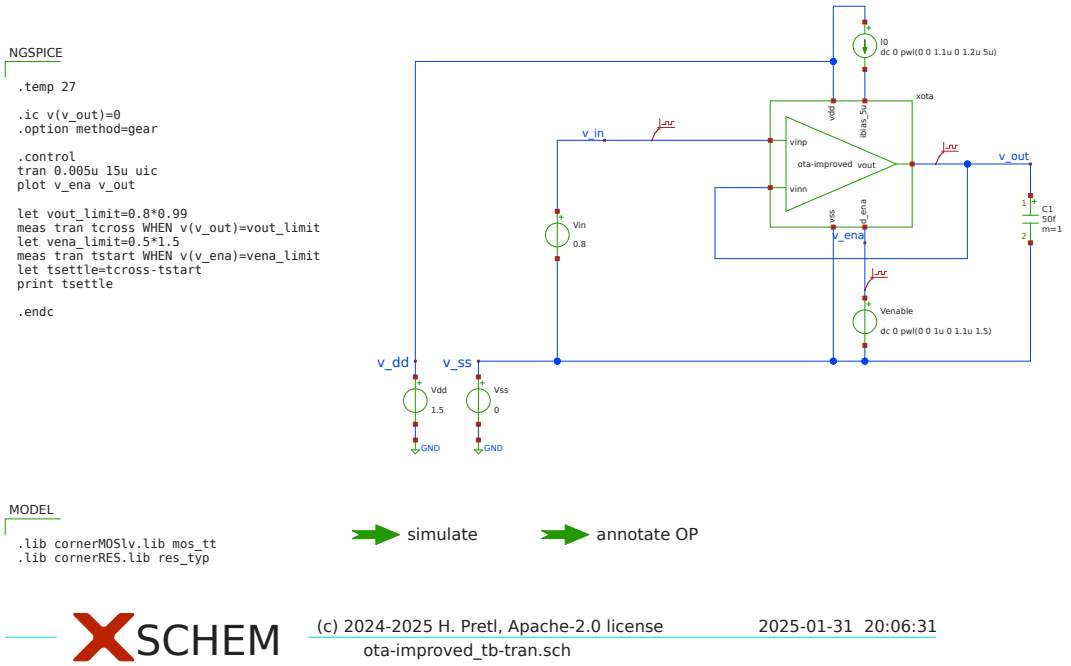


Figure 65: Simulation testbench of the improved OTA design (large-signal).

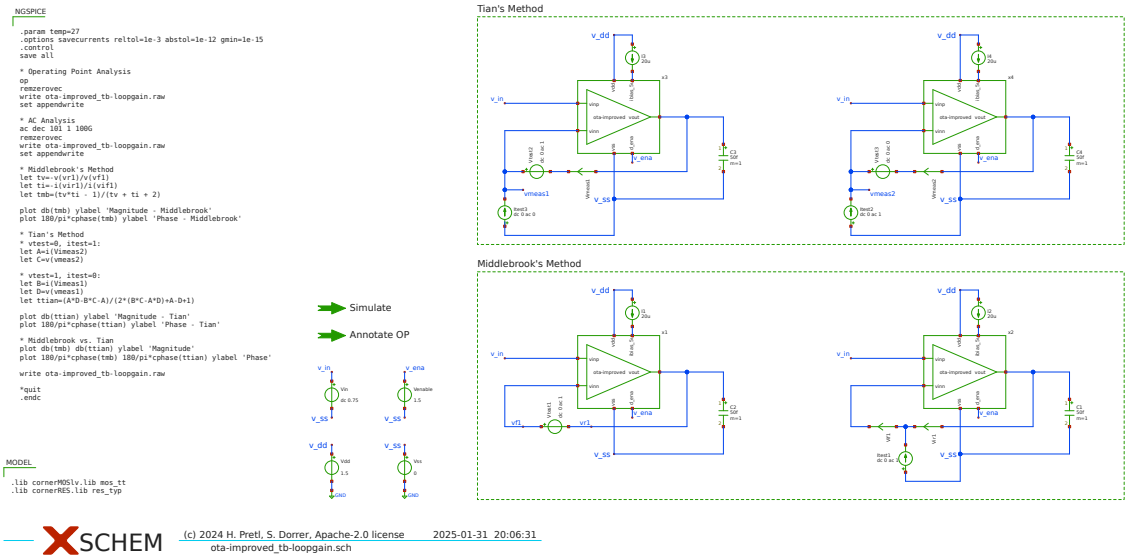


Figure 66: Simulation testbench of the improved OTA design (loop gain analysis).

happens. Whenever you change something, check the dc operating point first. If the dc operating point is not good no further simulations make sense.

10.3 Corner Simulation of Improved OTA

Just like for the basic OTA we use the CACE system to check the performance of the improved OTA design holistically across variations like PVT and input signal variations. The results of the CACE run are shown below in Note 3.

Note 3: CACE Summary for Improved OTA

CACE Summary for ota-improved

netlist source: schematic

Parameter	Tool	Result	Min Limit	Min Value	Typ Target	Typ Value	Max Limit	Max Value	Status
Output voltage ratio	ngspice	gain	0.99 V/V	1.000 V/V	any	1.002 V/V	1.01 V/V	1.006 V/V	Pass
Bandwidth	ngspice	bw	10e6 Hz	146600000.000 Hz	any	206653000.000 Hz	any	254164000.000 Hz	Pass
Output voltage ratio (MC)	ngspice	gain_mc	any	0.809 V/V	any	1.023 V/V	any	1.304 V/V	Pass
Bandwidth (MC)	ngspice	bw_mc	10e6 Hz	24169400.000 Hz	any	171443500.000 Hz	any	279056000.000 Hz	Pass
Output noise	ngspice	noise	any	0.309 mV	any	0.391 mV	0.6 mV	0.530 mV	Pass
Settling time	ngspice	tsettle	any	0.134 us	any	0.141 us	1 us	0.151 us	Pass

Plots

gain_vs_temp

Figure 67: gain_vs_temp

gain_vs_vin

Figure 68: gain_vs_vin

gain_vs_vdd

Figure 69: gain_vs_vdd

gain_vs_corner

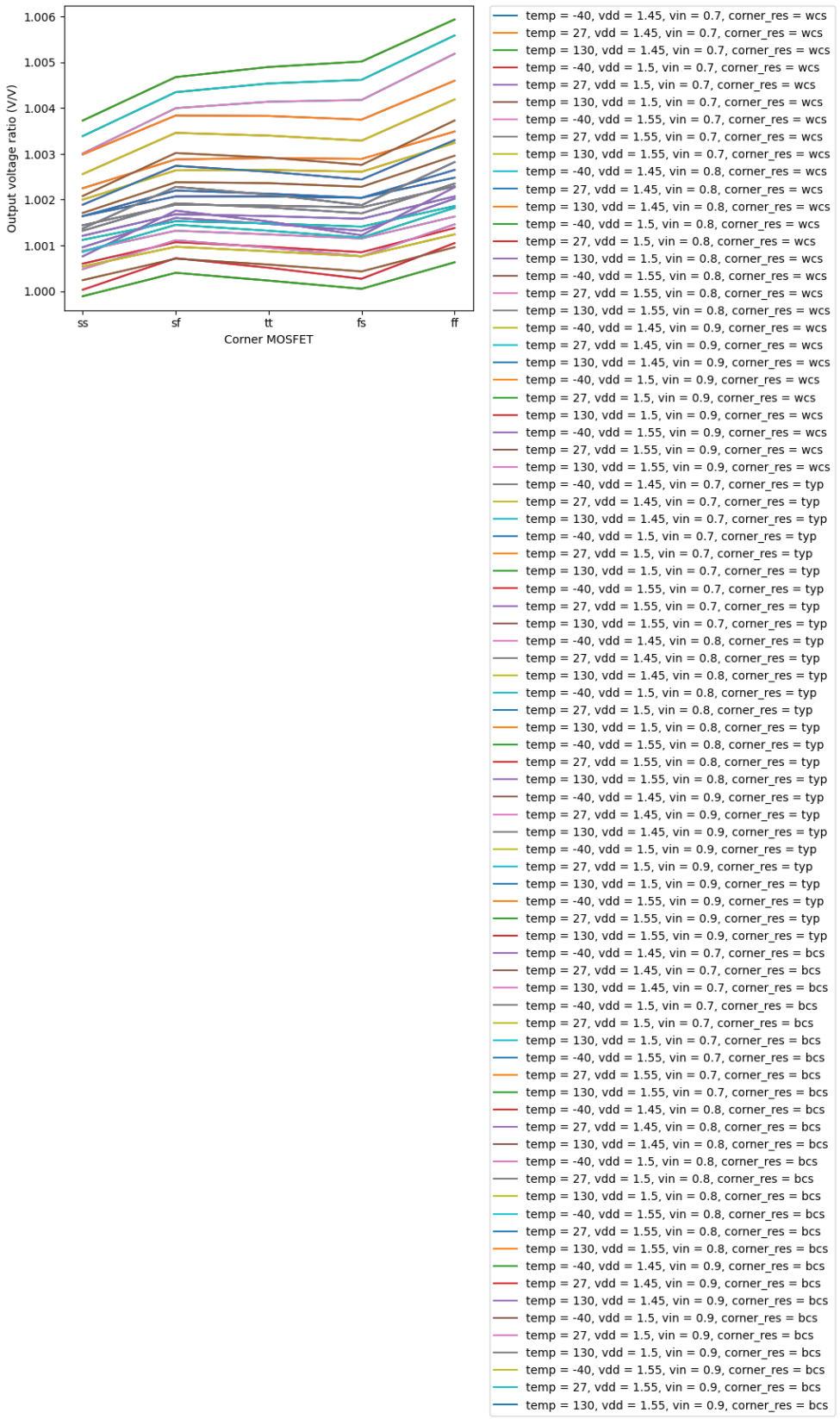


Figure 70: gain_vs_corner

bw_vs_temp

Figure 71: bw_vs_temp

bw__vs__vin

Figure 72: bw__vs__vin

bw_vs_vdd

Figure 73: bw_vs_vdd

bw_vs_corner

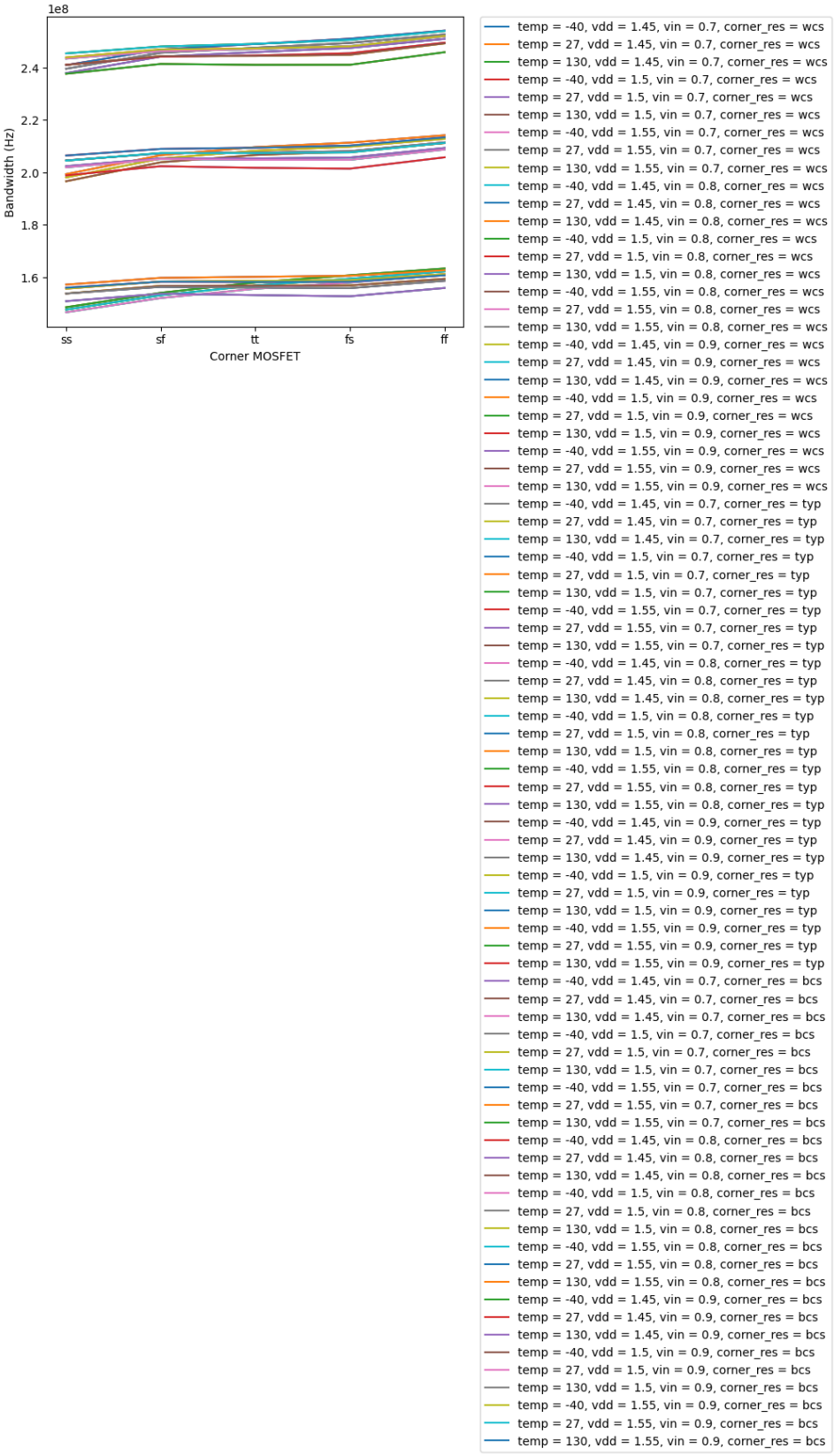


Figure 74: bw_vs_corner

gain_mc

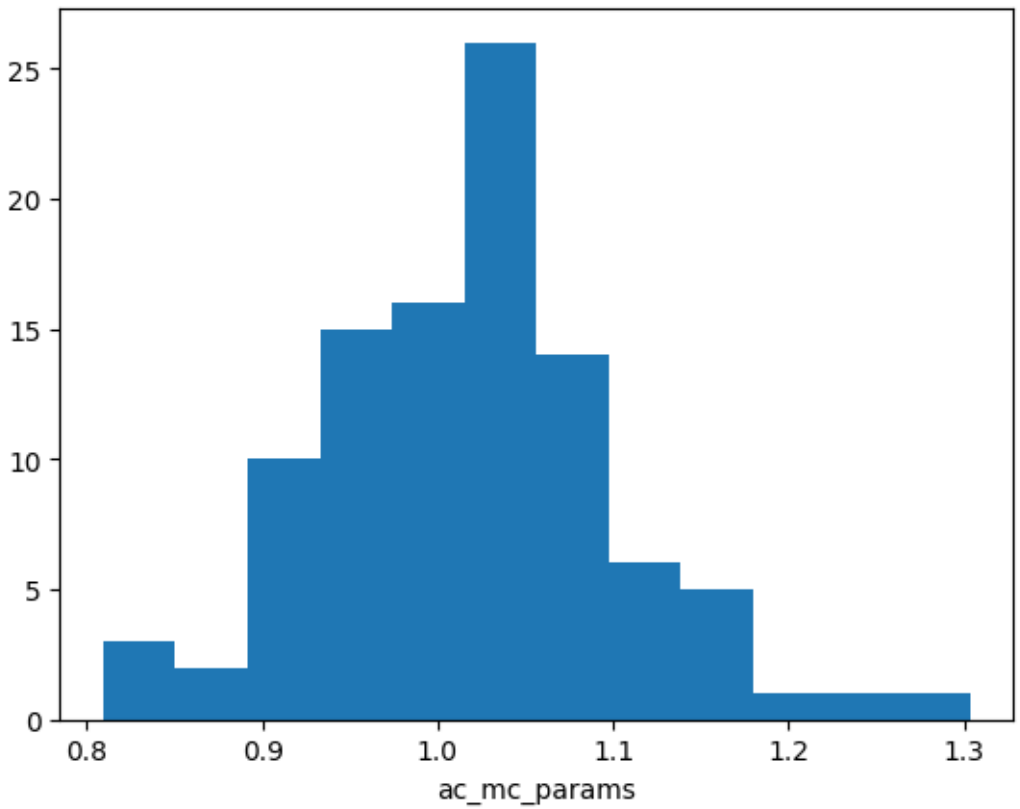


Figure 75: gain_mc

bw_mc

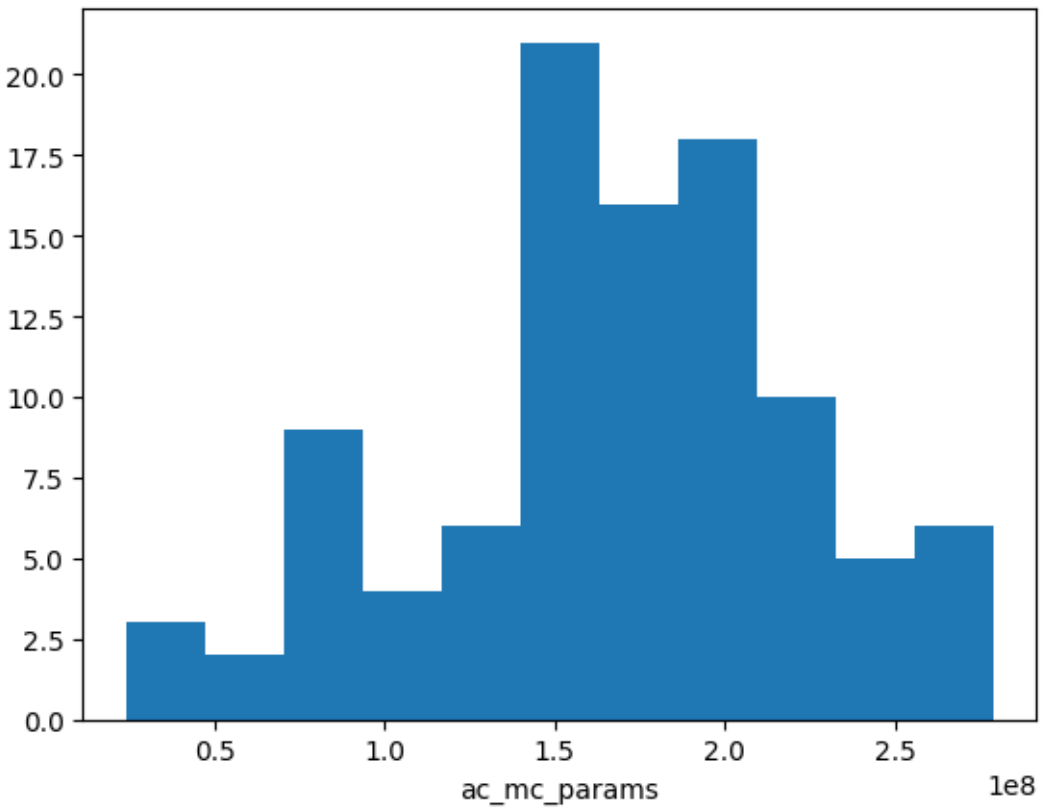


Figure 76: bw_mc

noise_vs_temp

Figure 77: noise_vs_temp

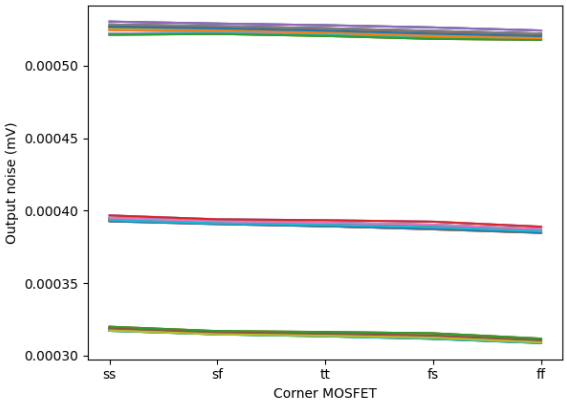
noise_vs_vin

Figure 78: noise_vs_vin

noise_vs_vdd

Figure 79: noise_vs_vdd

noise_vs_corner



temp = -40, vdd = 1.45, vin = 0.7, corner_res = wcs
temp = 27, vdd = 1.45, vin = 0.7, corner_res = wcs
temp = 130, vdd = 1.45, vin = 0.7, corner_res = wcs
temp = -40, vdd = 1.5, vin = 0.7, corner_res = wcs
temp = 27, vdd = 1.5, vin = 0.7, corner_res = wcs
temp = 130, vdd = 1.5, vin = 0.7, corner_res = wcs
temp = -40, vdd = 1.55, vin = 0.7, corner_res = wcs
temp = 27, vdd = 1.55, vin = 0.7, corner_res = wcs
temp = 130, vdd = 1.55, vin = 0.7, corner_res = wcs
temp = -40, vdd = 1.45, vin = 0.8, corner_res = wcs
temp = 27, vdd = 1.45, vin = 0.8, corner_res = wcs
temp = 130, vdd = 1.45, vin = 0.8, corner_res = wcs
temp = -40, vdd = 1.5, vin = 0.8, corner_res = wcs
temp = 27, vdd = 1.5, vin = 0.8, corner_res = wcs
temp = 130, vdd = 1.5, vin = 0.8, corner_res = wcs
temp = -40, vdd = 1.55, vin = 0.8, corner_res = wcs
temp = 27, vdd = 1.55, vin = 0.8, corner_res = wcs
temp = 130, vdd = 1.55, vin = 0.8, corner_res = wcs
temp = -40, vdd = 1.45, vin = 0.9, corner_res = wcs
temp = 27, vdd = 1.45, vin = 0.9, corner_res = wcs
temp = 130, vdd = 1.45, vin = 0.9, corner_res = wcs
temp = -40, vdd = 1.5, vin = 0.9, corner_res = wcs
temp = 27, vdd = 1.5, vin = 0.9, corner_res = wcs
temp = 130, vdd = 1.5, vin = 0.9, corner_res = wcs
temp = -40, vdd = 1.55, vin = 0.9, corner_res = wcs
temp = 27, vdd = 1.55, vin = 0.9, corner_res = wcs
temp = 130, vdd = 1.55, vin = 0.9, corner_res = wcs
temp = -40, vdd = 1.45, vin = 0.7, corner_res = typ
temp = 27, vdd = 1.45, vin = 0.7, corner_res = typ
temp = 130, vdd = 1.45, vin = 0.7, corner_res = typ
temp = -40, vdd = 1.5, vin = 0.7, corner_res = typ
temp = 27, vdd = 1.5, vin = 0.7, corner_res = typ
temp = 130, vdd = 1.5, vin = 0.7, corner_res = typ
temp = -40, vdd = 1.55, vin = 0.7, corner_res = typ
temp = 27, vdd = 1.55, vin = 0.7, corner_res = typ
temp = 130, vdd = 1.55, vin = 0.7, corner_res = typ
temp = -40, vdd = 1.45, vin = 0.8, corner_res = typ
temp = 27, vdd = 1.45, vin = 0.8, corner_res = typ
temp = 130, vdd = 1.45, vin = 0.8, corner_res = typ
temp = -40, vdd = 1.5, vin = 0.8, corner_res = typ
temp = 27, vdd = 1.5, vin = 0.8, corner_res = typ
temp = 130, vdd = 1.5, vin = 0.8, corner_res = typ
temp = -40, vdd = 1.55, vin = 0.8, corner_res = typ
temp = 27, vdd = 1.55, vin = 0.8, corner_res = typ
temp = 130, vdd = 1.55, vin = 0.8, corner_res = typ
temp = -40, vdd = 1.45, vin = 0.9, corner_res = typ
temp = 27, vdd = 1.45, vin = 0.9, corner_res = typ
temp = 130, vdd = 1.45, vin = 0.9, corner_res = typ
temp = -40, vdd = 1.5, vin = 0.9, corner_res = typ
temp = 27, vdd = 1.5, vin = 0.9, corner_res = typ
temp = 130, vdd = 1.5, vin = 0.9, corner_res = typ
temp = -40, vdd = 1.55, vin = 0.9, corner_res = typ
temp = 27, vdd = 1.55, vin = 0.9, corner_res = typ
temp = 130, vdd = 1.55, vin = 0.9, corner_res = typ
temp = -40, vdd = 1.45, vin = 0.7, corner_res = bcs
temp = 27, vdd = 1.45, vin = 0.7, corner_res = bcs
temp = 130, vdd = 1.45, vin = 0.7, corner_res = bcs
temp = -40, vdd = 1.5, vin = 0.7, corner_res = bcs
temp = 27, vdd = 1.5, vin = 0.7, corner_res = bcs
temp = 130, vdd = 1.5, vin = 0.7, corner_res = bcs
temp = -40, vdd = 1.55, vin = 0.7, corner_res = bcs
temp = 27, vdd = 1.55, vin = 0.7, corner_res = bcs
temp = 130, vdd = 1.55, vin = 0.7, corner_res = bcs
temp = -40, vdd = 1.45, vin = 0.8, corner_res = bcs
temp = 27, vdd = 1.45, vin = 0.8, corner_res = bcs
temp = 130, vdd = 1.45, vin = 0.8, corner_res = bcs
temp = -40, vdd = 1.5, vin = 0.8, corner_res = bcs
temp = 27, vdd = 1.5, vin = 0.8, corner_res = bcs
temp = 130, vdd = 1.5, vin = 0.8, corner_res = bcs
temp = -40, vdd = 1.55, vin = 0.8, corner_res = bcs
temp = 27, vdd = 1.55, vin = 0.8, corner_res = bcs
temp = 130, vdd = 1.55, vin = 0.8, corner_res = bcs
temp = -40, vdd = 1.45, vin = 0.9, corner_res = bcs
temp = 27, vdd = 1.45, vin = 0.9, corner_res = bcs
temp = 130, vdd = 1.45, vin = 0.9, corner_res = bcs
temp = -40, vdd = 1.5, vin = 0.9, corner_res = bcs
temp = 27, vdd = 1.5, vin = 0.9, corner_res = bcs
temp = 130, vdd = 1.5, vin = 0.9, corner_res = bcs
temp = -40, vdd = 1.55, vin = 0.9, corner_res = bcs
temp = 27, vdd = 1.55, vin = 0.9, corner_res = bcs
temp = 130, vdd = 1.55, vin = 0.9, corner_res = bcs

Figure 80: noise_vs_corner

settling_vs_temp

Figure 81: settling_vs_temp

settling__vs__vin

Figure 82: settling__vs__vin

settling_vs_vdd

Figure 83: settling_vs_vdd

settling_vs_corner

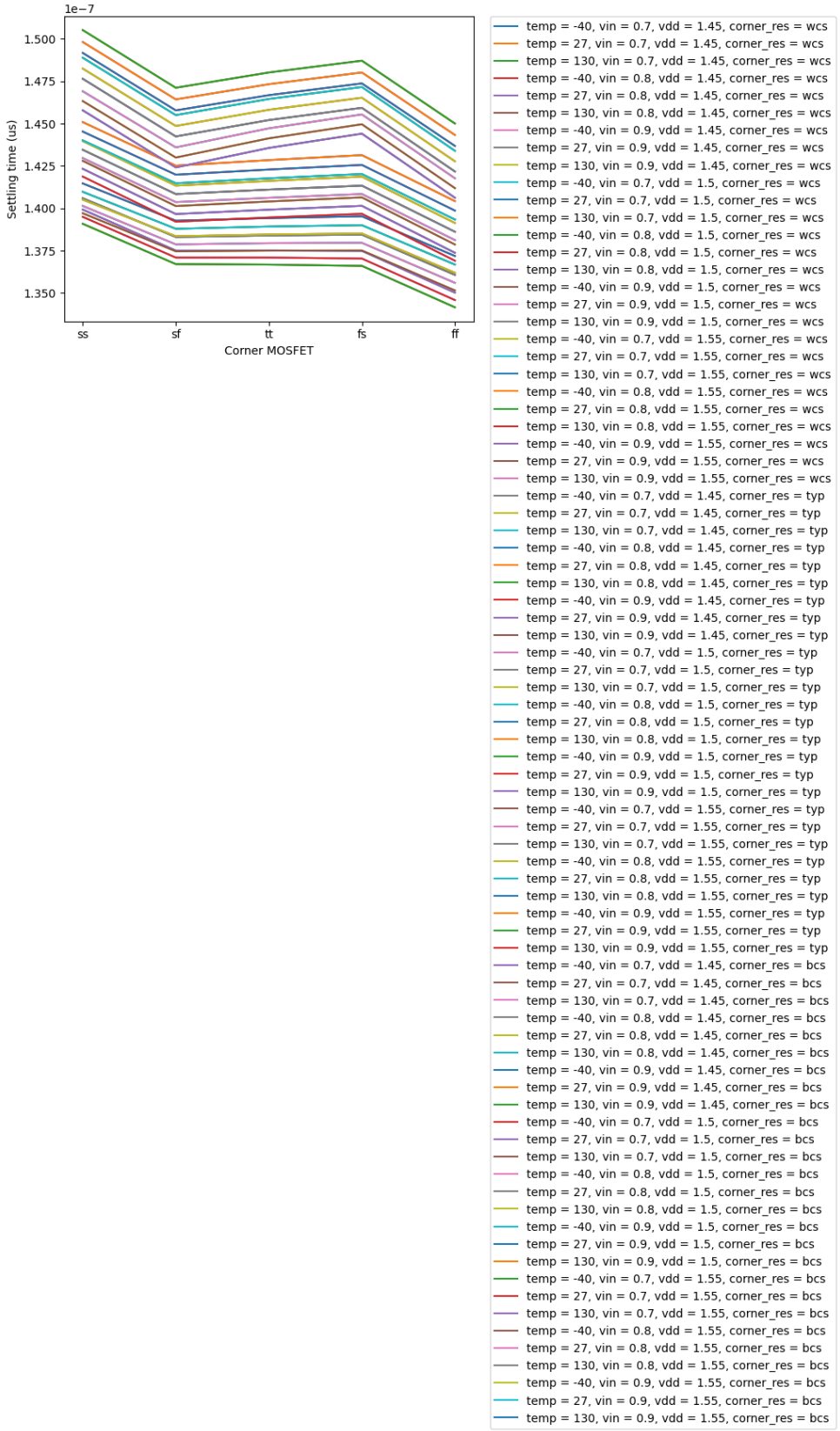


Figure 84: settling_vs_corner

The improved performance allows to improve the specifications in a few important points, notably the output voltage tolerance which is an important metric for a reference voltage buffer. We have intentionally increased the power consumption a little bit, but we negotiated with the chip lead designer a changed bias current level, so overall the situation is even slightly improved. The new situation with the improved design is summarized in Table 5 (unchanged entries are not shown).

Table 5: Voltage buffer specification

Specification	Basic 5T-OTA	Improved OTA	Unit
Output voltage error	< 3	< 1	%
Total output noise (rms)	< 1	< 0.6	$\mathrm{mV_{rms}}$
Supply current (as low as possible)	< 10	< 20	µA
Turn-on time (settled to with 1%)	< 10	< 1	µs
Externally provided bias current (nominal)	20	5	µA

11 Biasing

So far, we have pushed one unresolved issue in front of us. We have studied ways to multiply and scale bias currents using current mirrors (see Section 6) and we have also found ways to buffer (and scale) bias voltages (see Section 8.1, scaling can be readily achieved by using a resistive divider in the feedback path). We can generate bias voltages by running a current through a resistor. However, what has not been discussed so far is **how to generate one stable bias current in the first place**.

If we would have a stable reference voltage, then we can use the arrangement shown in Figure 85 to generate a bias current that is given by (note the use of feedback using an error amplifier which could be a simple OTA)

$$I_{\mathrm{bias}} = \frac{V_{\mathrm{ref}}}{R_1}.$$

Using A_1 and M_1 the voltage V_{ref} is regulated across R_1 and available at the drain of M_1.

The resistor R_1 will show variation with the process tolerance, but we can either (a) use an external resistor for a precise current generation, or (b) use some sort of trimming to correct this resistor value. Depending on the further use of the bias currents a variation of the resistor might no be a bad thing. Assuming we use the resulting bias current to generate somewhere a bias voltage V_{bias} by running the current through resistor R_2, then this voltage is given by

$$V_{\mathrm{bias}} = R_2 I_{\mathrm{bias}} = \frac{R_2}{R_1} V_{\mathrm{ref}}.$$

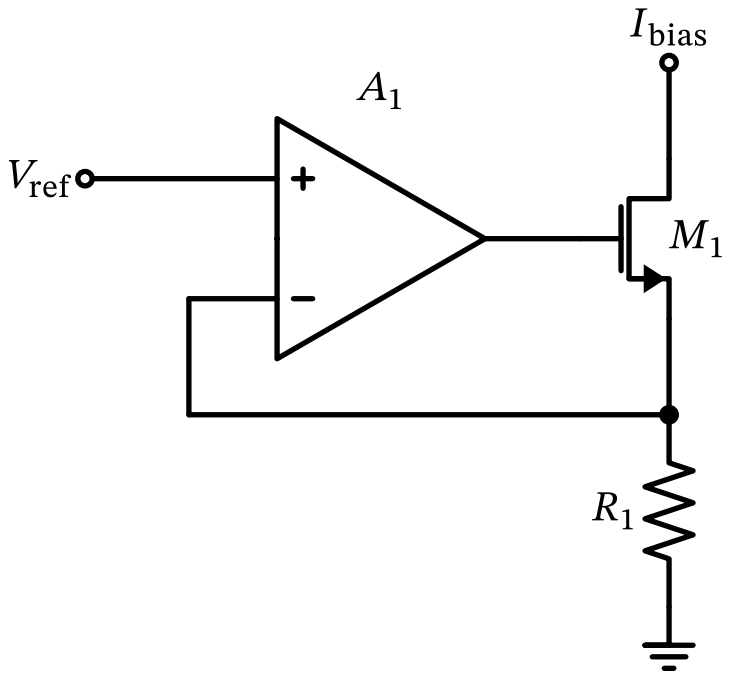


Figure 85: A constant-current generator based on OTA.

We can see in the above equation that the absolute values of R_1 and R_2 do not play a role, only the *ratio* is important. Luckily, components in integrated circuits match very well, so we can multiply and scale one reference voltage V_{ref} across our IC.

Since all components on chip will experience manufacturing tolerances of at least $\pm 10\%$ we strive for something more accurate. We can resort to an off-chip property like an externally provided reference current or voltage, or use the power supply as a voltage reference (often, power supply rails are specified to $\pm 5 \ldots 10\%$).

The only option left is to use a material property of the silicon itself for stable reference voltage generation.

11.1 Bandgap Reference

It has been realized that a bipolar junction transistor (BJT) has the wonderful property, that the base-emitter voltage V_{BE} has the following approximate relationship vs. temperature (R. J. Widlar 1971)

$$V_{\text{BE}} \approx V_{\text{g0}} \left(1 - \frac{T}{T_0}\right) + V_{\text{BE0}} \left(\frac{T}{T_0}\right) \tag{37}$$

where $V_{\text{g0}} = 1.205\,\text{V}$ is the bandgap voltage of silicon at 0 K and V_{BE0} is the base-emitter voltage of a BJT as reference temperature T_0. Further, the difference in V_{BE} of two BJT operated at different emitter current densities J_1 and J_2 is given as

$$\Delta V_{\mathrm{BE}} = \frac{kT}{q} \ln\left(\frac{J_1}{J_2}\right) \tag{38}$$

with k the Boltzmann constant, and q the elementary charge.

Adding Equation 37 and Equation 38 results in a reference voltage of value

$$V_{\mathrm{ref}} = V_{\mathrm{g0}}\left(1 - \frac{T}{T_0}\right) + V_{\mathrm{BE0}}\left(\frac{T}{T_0}\right) + \frac{kT}{q} \ln\left(\frac{J_1}{J_2}\right) \tag{39}$$

which can be made temperature-insensitive when the terms, which are a function of T, cancel each other, and only

$$V_{\mathrm{ref}} = V_{\mathrm{g0}} \tag{40}$$

remains. We thus have created an on-chip reference (called the *bandgap voltage reference*) which is almost independent of manufacturing tolerances with zero temperature coefficient. Of course, this is only true neglecting second-order effects, but nevertheless, reference accuracies of $\pm 1 \ldots 3\%$ without trimming are perfectly possible.

The original implementation in (R. J. Widlar 1971) uses NPN transistors. The question is, where do we find BJT in a CMOS process? Luckily, when looking at the typical implementations, we find that there is a layer sandwich of P-N-P available. While the PNPs constructed parasitically out of this available layers are available for free without extra processing cost, they are very slow, show unusually small $\beta < 10$, and the collector is tied to V_{SS} as it is the substrate. Still, for bandgap references, they are very useful.

A simple implementation of a bandgap reference circuit is shown in Figure 86. If we scale R_1 and R_2 correctly then we can achieve Equation 40. Note that the output voltage is ca. 1.2 V, so operating this circuit on low supply voltages will be problematic.

M_1 to M_4 are scaled in a way that in both branches the same bias current is flowing. Further, M_1 and M_2 ensure that there is the same potential at their sources. Since the PNP are scaled by the factor n (thus the current density is different) so that the following voltage develops across R_1:

$$\Delta V_{\mathrm{BE}} = \frac{kT}{q} \ln m$$

Hence, the bias current in all the branches is given by

$$I_{\mathrm{bias}} = \frac{\Delta V_{\mathrm{BE}}}{R_1} = \frac{1}{R_1}\frac{kT}{q} \ln m. \tag{41}$$

Inspecting Equation 41 we see that $I_{\mathrm{bias}} = k_1 T$ is a linear function of temperature T, a property that is very useful and called **PTAT** (proportional to absolute temperature).

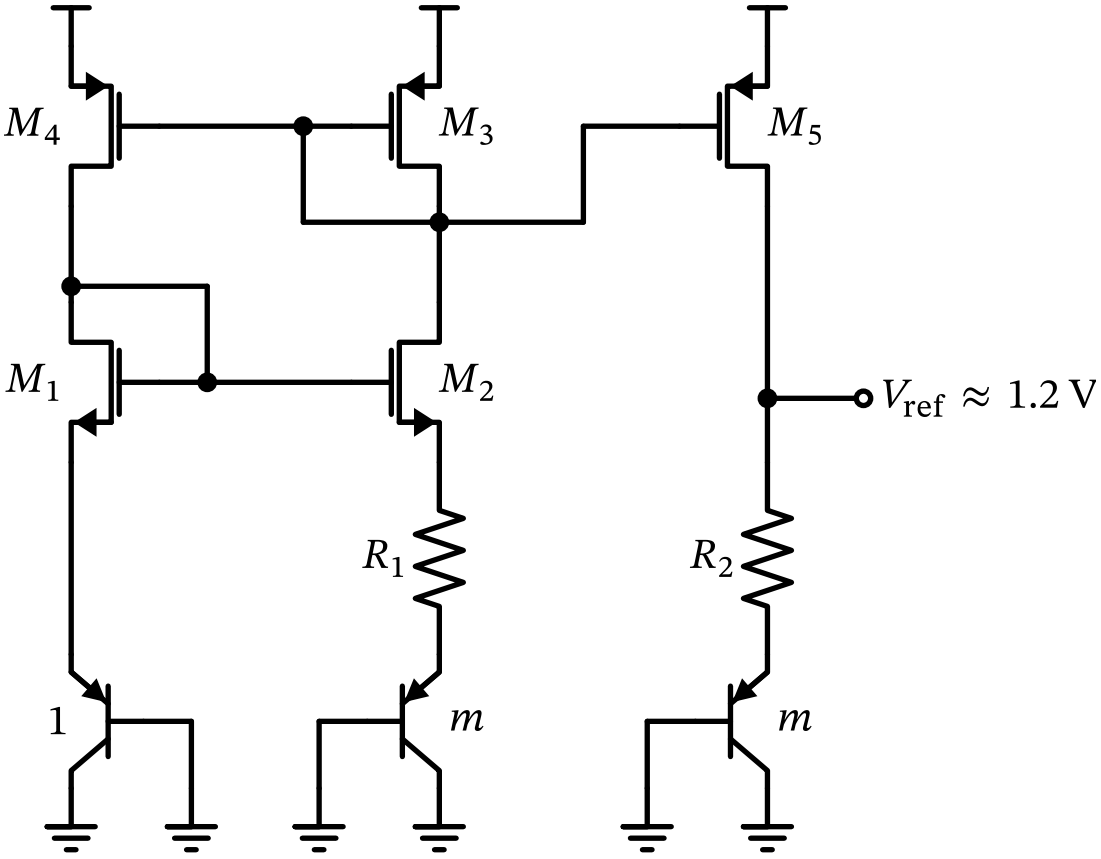


Figure 86: A simple bandgap reference.

With M_5 we mirror this bias current into the output branch, and the output voltage V_{ref} is then given by

$$V_{\text{ref}} = V_{\text{BE}} + \frac{R_2}{R_1}\frac{kT}{q}\ln m.$$

By proper selection of R_1, R_2 and m we can satisfy Equation 39 to result in Equation 40.

> **Improved Bandgap Reference**
>
> For an improved implementation of Figure 86, the current mirrors should be cascoded, and a startup circuit should be included to guarantee proper operation after enabling it. Further, Equation 37 and Equation 38 build on the relationship $I_{\text{C}} = f(V_{\text{BE}})$, while we control I_{E} in this circuit. If β is large then $I_{\text{C}} \approx I_{\text{E}}$, but this is not the case for the used PNPs.

The circuit of Figure 86 has been implemented in Xschem and is shown in Figure 87. The current sources have been improved by using cascodes. We are using the low-voltage current mirror type already introduced in Section 10. The bias voltages for the cascodes are generated via the voltage drops of R_3 and R_4, respectively.

No base current compensation for the BJTs is implemented, as it is assumed that the β of the PNP are similar although they are operated at different emitter current densities.

Note the addition of a startup branch with M_{startup} which is inactive during normal operation but will inject a startup current if no proper bias point has yet been found.

There is no circuitry added for enabling/disabling the circuit, which would also be needed for a practical implementation. As usual, the MOSFET sizing has been done in this notebook.

The resistors R_{1-4} have been implemented out of unit elements of ca. 5 kΩ for optimum matching. Building a bandgap for the first time on silicon likely will show a slightly deviating temperature coefficient, which is why we keep a few dummy resistors around in R_2 to compensate the TC in a redesign.

The Xschem schematic is available here and the simulated reference voltage vs. temperature is shown in Figure 88. For a typical process we achieve a TC of ±0.2%.

Please note how tight the dc operating point is in this design to keep all MOSFET saturated. We only use 100 mV nominally as headroom. The circuit in Figure 87 works only marginally at $V_{\text{DD}} = 1.5\,\text{V}$, but would not work at 1.2 V or lower. Improved circuit architectures for $< 1\,\text{V}$ operation exist Eberlein, Panagopoulos, and Pretl (2018).

11.2 Banba Bandgap Reference

The Banba reference (Banba et al. 1999) is quite a bit trickier to design than the classical bandgap shown in Figure 86. It requires the use of an error amplifier; luckily, we can use the 5T-OTA which we designed in Section 8. Since a loop is involved the startup of this

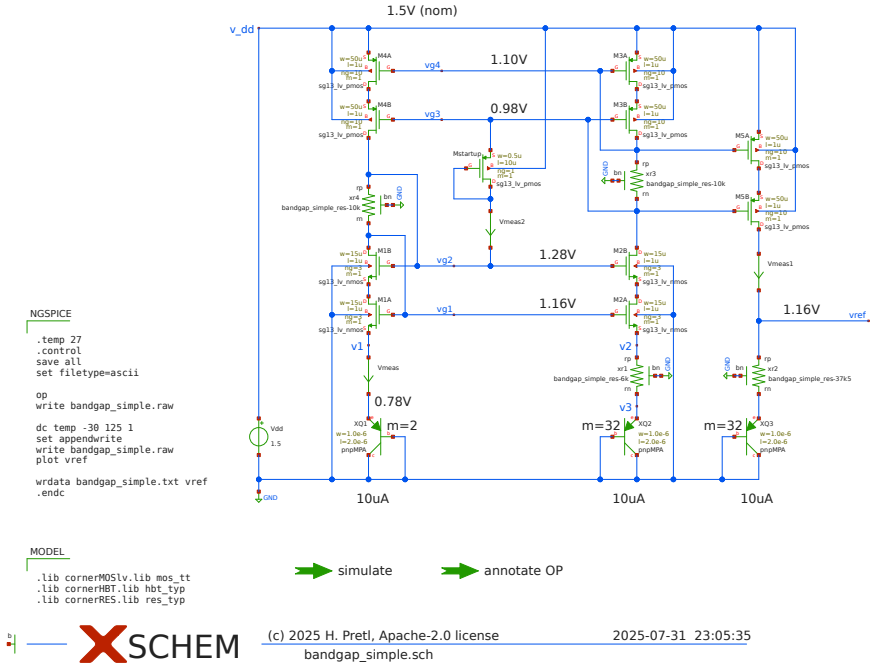


Figure 87: Simple bandgap reference circuit in Xschem.

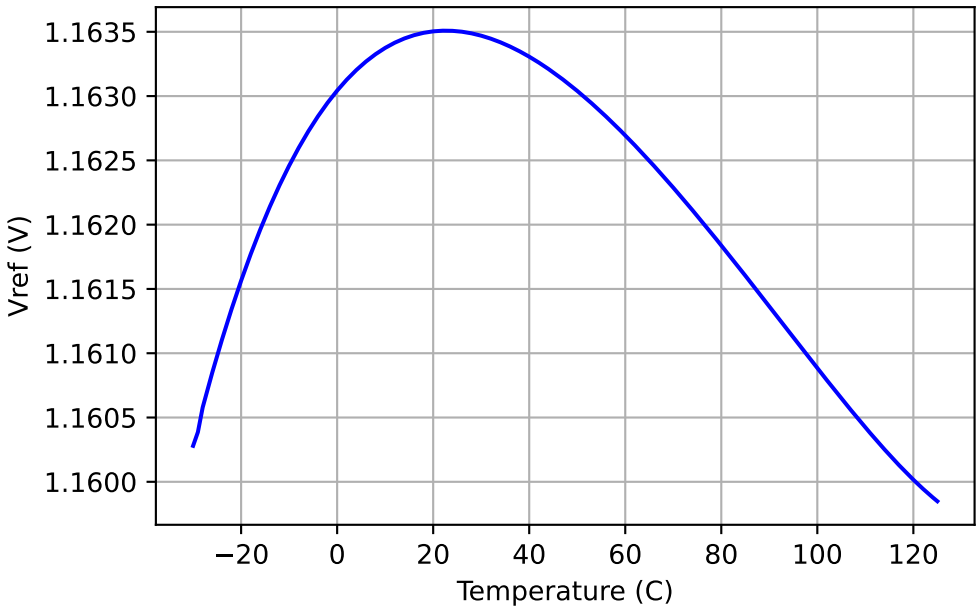


Figure 88: Reference voltage from simulated bandgap circuit.

circuit is not easy and requires the use of a transient simulation and a proper pre-charge of critical nodes. We can use the ngspice scripting language to (a) set the temperature for a sweep, (b) run a transient simulation, and (c) capture the final reference voltage and save it.

A first design has been implemented and is shown in Figure 89, and the testbench is shown in Figure 90. The supply voltage (which could be lower than 1.5 V) is limited by our OTA design; however, it works well at $V_{\mathrm{DD}} = 1.5\,\mathrm{V}$. The simulated reference voltage (which is scaled to roughly $V_{\mathrm{bandgap}}/2$) is shown in Figure 91.

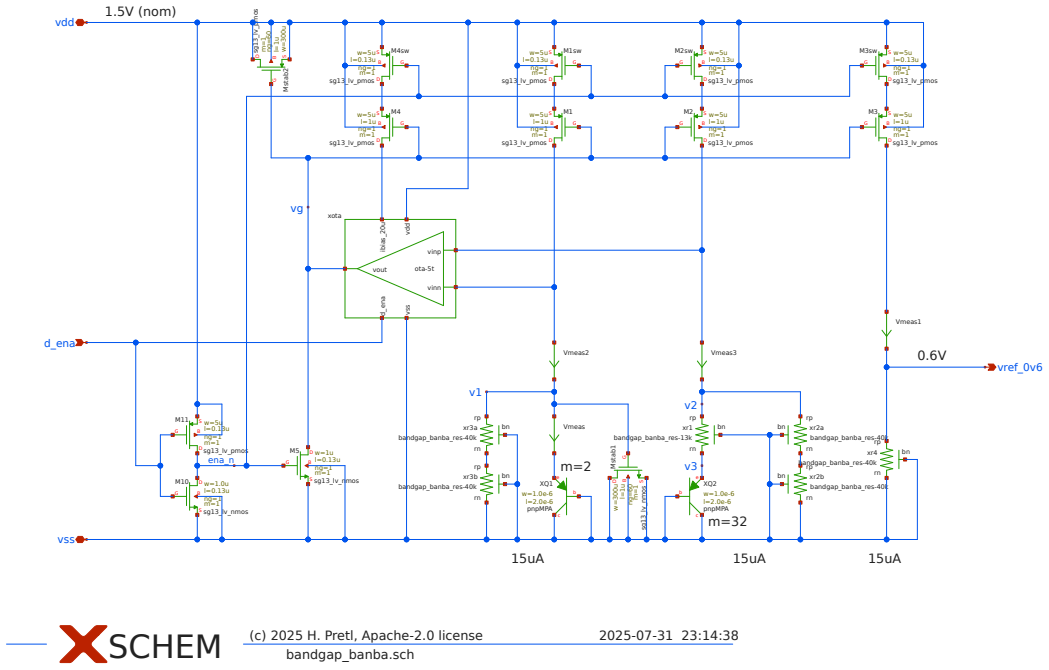


Figure 89: Banba bandgap reference circuit in Xschem.

Note the self-biasing of the OTA with the current generated in the reference branches. The feedback loop needs capacitors for stabilization, and we use area-efficient MOSFET for this task. For a detailed explanation of this circuit please refer to (Banba et al. 1999), but in brief, the operation is as follows:

The CTAT and PTAT currents required for compensating each others TC's are built using a ΔV_{BE} cell (given the PTAT current) with parallel resistors (providing the CTAT current). Voltages are sensed using an error amplifier and current sources are controlled to achieve matching currents in both branches. This current can then also be mirrored to flow through another resistor which can be scaled to produce the output voltage.

In comparison to the classical bandgap reference shown in Figure 86 (where the developed currents in all branches are PTAT) the currents developed in the Banba reference are constant vs. temperature.

The minimum supply voltage for the Banba reference is the V_{BE} of the PNP plus the

saturation voltage of one MOSFET current source, so ca. $V_{\mathrm{DDmin}} \geq 0.8\,\mathrm{V} + 0.2\,\mathrm{V} \approx 1\,\mathrm{V}$. Of course we also need to design an OTA which can work at this low supply.

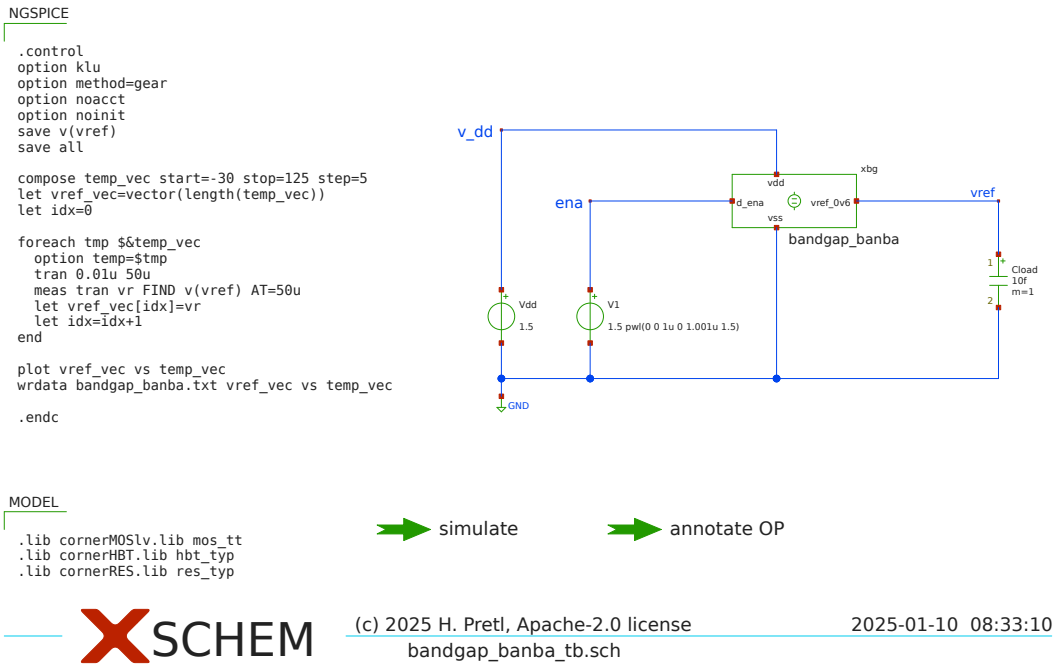


Figure 90: Banba bandgap testbench Xschem.

A well described design of a Banba bandgap reference (including a two-stage OTA and a regulated cascode for the output current mirror), covering much more details than discussed here, can be found in [Razavi_2021_bandgap].

> Exercise: Improved Low-Voltage Bandgap
>
> As an *optional* exercise for advanced users: Design a bandgap circuit following [Razavi_2021_bandgap]. Implement the shown two-stage OTA and the regulated cascode.
> As a starting point, the design of Section 11.2 can be used. As this design will contain more blocks, please build up a hierarchical design, with the OTAs designed in separate subcircuits.

12 Differential OTAs

So far, we have discussed the implementation of OTAs with a differential input and a single-ended output. Often, in integrated circuits, we want to implement fully differential signal chains, as this allows an almost-rail-to-rail swing around a common-mode voltage. Further, noise pickup due to limited power-supply rejection ratio (PSRR) or coupling into

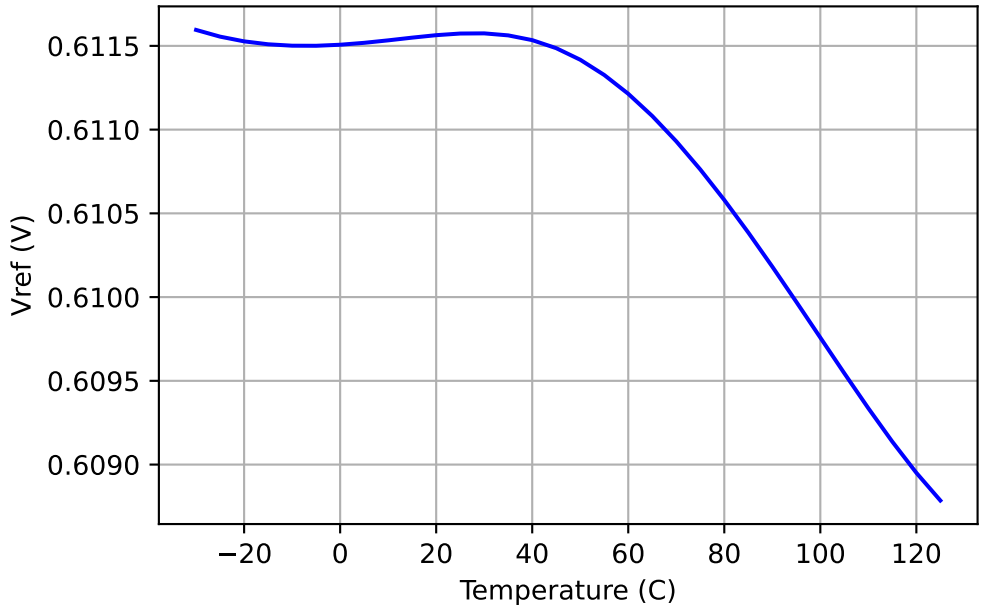


Figure 91: Reference voltage from simulated Banba bandgap circuit.

the differential signal routing can be suppressed by designing amplifiers with high common-mode rejection ratio (CMRR).

The OTA presented in Section 8.1 can be readily adapted for differential output. Striving for maximum utility, we are discussing a popular two-stage differential-output OTA with a special kind of load, shown in Figure 92.

This OTA shows a load on top of the differential pair that we have not yet studied. It is instructive to analyze this structure in terms of differential and common-mode operation.

For common-mode operation (i.e., injecting the same current into the drain of M_3 and M_4) we have the same voltage at the drains of $M_{3,5}$, thus $V_{\mathrm{DS3}} = V_{\mathrm{DS5}}$. We have thus no current flowing through $R_{1,2}$ and as a result $V_{\mathrm{DS3}} = V_{\mathrm{DS5}} = V_{\mathrm{X}} = V_{\mathrm{GS3}} = V_{\mathrm{GS5}} = V_{\mathrm{GS3,5}}$. We realize that M_3 and M_5 are diode-connected for common-mode operation. We thus have well-defined dc operating points, and also low common-mode gain, since the common-mode load is $g_{\mathrm{m3}}^{-1} \parallel g_{\mathrm{m5}}^{-1}$.

For pure differential operation, the mid-point X of $R_{1,2}$ acts as a virtual ground. The bias voltage $V_{\mathrm{GS3,5}}$ is thus not changed (it is set only by the common-mode operation) and thus M_3 and M_5 operate as current sources. The differential load impedance is high and is given by $R_{\mathrm{load}} = (R_1 + R_2) \parallel (g_{\mathrm{ds3}}^{-1} + g_{\mathrm{ds5}}^{-1})$.

In addition, we realize that $V_{\mathrm{GS4}} = V_{\mathrm{GS3}} = V_{\mathrm{GS5}} = V_{\mathrm{GS6}}$ for common-mode operation, hence the quiescent current of $M_{4,6}$ is set in a current-mirror-like fashion by the diode-connected $M_{3,5}$.

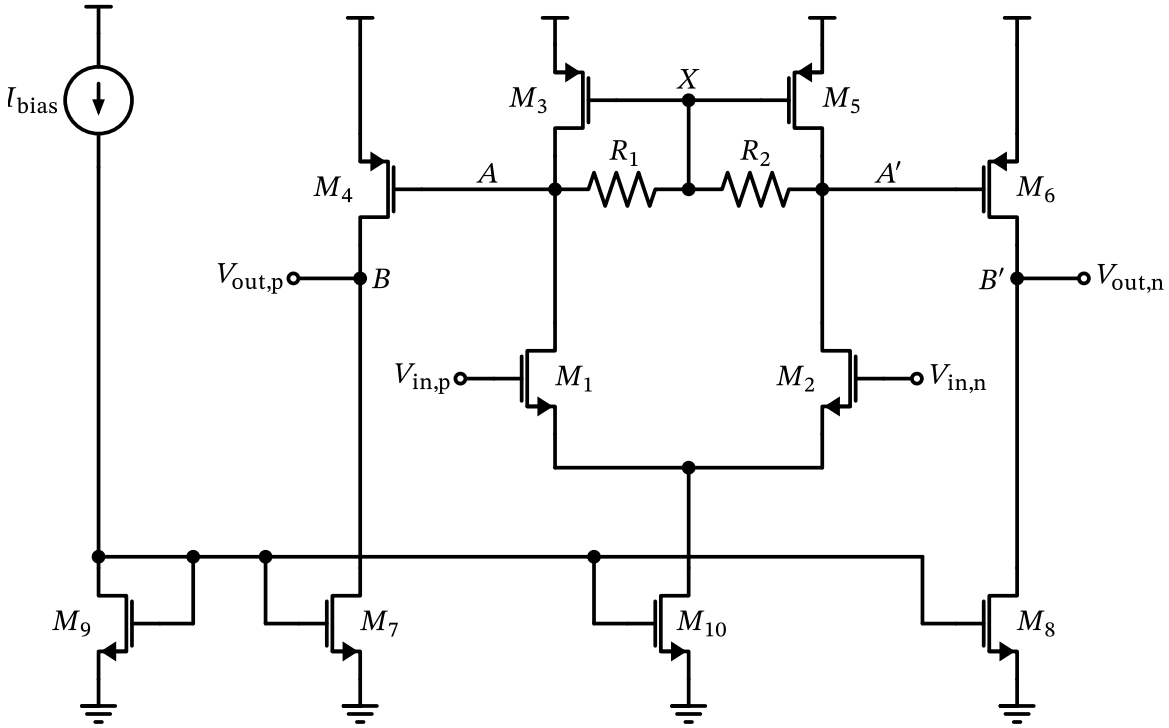


Figure 92: Differential two-stage OTA with resistive load.

For differential operation, the differential pair of $M_{1,2}$ is loaded by R_{load} and provides fairly high gain. The second gain stage is formed by current-source-loaded common-source stages $M_{4,6}$ and provides additional gain (of course, this is a function of the load impedance).

As soon as we implement a two-stage amplifier we need to look into stability. We likely have more than two poles, and in the case of the amplifier in Figure 92 we have a low-frequency pole at the drains of $M_{1,3}$ and $M_{2,5}$ and a further low-frequency pole at the output. Any additional pole will add further phase shift making stability critical. We now need a method to stabilize this amplifier and thus we will look into *Miller compensation*.

Of course, loop gain analysis of differential OTAs can and should be carried out. An exemplary testbench where the loop gain is simulated with Rosenstark's, Middlebrook's, and Tian's method can be found here. Note that there is no underlying circuit right now and it should only show how it could be done.

12.1 Miller Compensation

A popular way to stabilize a multi-pole feedback-system is to make one pole dominant, and try to shift the other poles to sufficiently high frequencies, that we have enough phase margin in the closed-loop system. We may strive for 60° as this will only cause a minor peaking in the frequency response (Gray et al. 2009).

The question now is where to implement this dominant pole. In order to create a low-frequency pole we need a high-impedance point and a large capacitance. Placing just a large capacitor is unwelcome, as this causes large area consumption on chip.

Luckily, we know from the analysis in Section 16.1 that we can use voltage gain to increase the apparent value of a capacitor by feedback. Inspecting our circuit in Figure 92 we see that we have voltage gain from node A to node B and node A' and B', respectively. This means we have an opportunity to strap a capacitor between those nodes, and create a dominant pole at node A (and A').

We can now add these so-called "Miller capacitors" to our circuit. The result is shown in Figure 93. (We have also added resistors in series with the capacitors; we ignore these resistors for the time being).

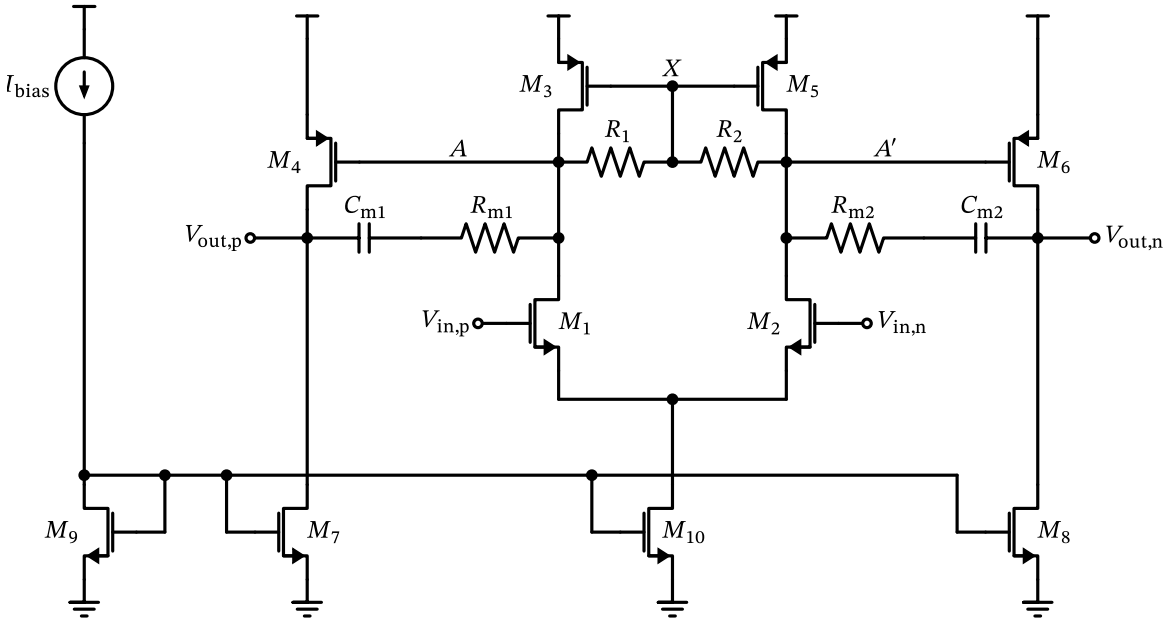


Figure 93: Differential two-stage OTA with resistive load and Miller compensation.

It is instructive to look at the small-signal equivalent circuit of the common-source stage M_4 loaded by current-source M_7. The resulting model is shown in Figure 94 (we are ignoring the bulk effect of M_4, and lump components of $M_{4,7}$ into the input and the output impedances formed by C_L, C_in, g_in and g_out).

In order to analyze the transfer function and thus the poles and zeros of this configuration we formulate KCL at the input and output node and the current through C_m (we set $R_\mathrm{m} = 0$ for now):

$$I_\mathrm{in} + I_\mathrm{m} - V_\mathrm{gs}(sC_\mathrm{in} + g_\mathrm{in}) = 0 \tag{42}$$

$$-I_\mathrm{m} - g_\mathrm{m}V_\mathrm{gs} - V_\mathrm{out}(sC_\mathrm{L} + g_\mathrm{out}) = 0 \tag{43}$$

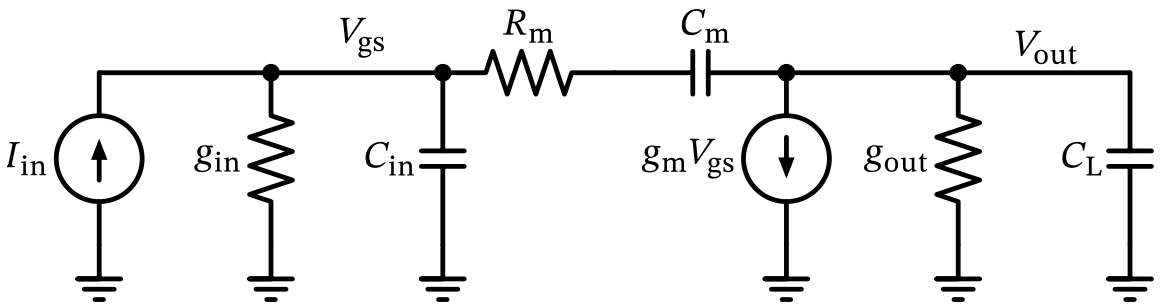


Figure 94: Small-signal model of common-source stage with Miller compensation.

$$I_m = sC_m(V_{out} - V_{gs}) \tag{44}$$

Using Equation 44 in Equation 42 and Equation 43 and then calculating the transfer function we arrive at

$$\frac{V_{out}}{I_{in}} = \frac{sC_m - g_m}{(sC_m + sC_L + g_{out})(sC_m + sC_{in} + g_{in}) - (sC_m - g_m)sC_m}. \tag{45}$$

In order to check Equation 45 we can set $sC_m = 0$ and see whether we can interpret the result:

$$\frac{V_{out}}{I_{in}} = -\frac{g_m}{(sC_L + g_{out})(sC_{in} + g_{in})} = -\frac{g_m}{g_{out}g_{in}} \frac{1}{\left(1 + \frac{sC_L}{g_{out}}\right)} \frac{1}{\left(1 + \frac{sC_{in}}{g_{in}}\right)} \tag{46}$$

We find that Equation 46 looks reasonable, as we have the correct dc gain of $-g_m/(g_{out}g_{in})$ and two poles, one at the input and one at the output.

We now return to the more interesting case of $sC_m \neq 0$. We use the reasonable assumption that $g_m \gg g_{in,out}$ to simplify the algebra and result in an interpretable result. After quite a few pages of algebraic manipulations (please try for yourself!) we arrive at

$$\frac{V_{out}}{I_{in}} = -\frac{g_m}{g_{out}g_{in}} \frac{\left(1 - \frac{sC_m}{g_m}\right)}{\left(1 + s\frac{C_L + C_{in} + \frac{C_{in}C_L}{C_m}}{g_m}\right)\left(1 + s\frac{C_m \frac{g_m}{g_{out}}}{g_{in}}\right)}. \tag{47}$$

Looking at Equation 47 we can identify important changes compared to Equation 46. We have the intended low-frequency pole s_{p1} at the input where the capacitor C_m is increased by the dc gain of the common-source stage g_m/g_{out}:

$$s_{p1} = -\frac{g_{in}}{C_m \frac{g_m}{g_{out}}}$$

We further have a high frequency pole s_{p2} where the pole at the output has been shifted to higher frequencies! This is a very welcome effect called *pole splitting*, and it helps to stabilize the feedback system, as the nondominant (output) pole is shifted out in frequency while the dominant (input) pole is pulled in.

$$s_{p2} = -\frac{g_m}{C_L + C_{in} + \frac{C_{in}C_L}{C_m}}$$

Together, the movement of poles s_{p1} and s_{p2} is a great deal in terms of stability. However, not all is rosy, as we have to also look at the numerator of Equation 47. Here we see that a zero s_z has been formed, unfortunately a quite bad one. Calculating its location as

$$s_z = +\frac{g_m}{C_m} \tag{48}$$

we see that it is located in the *right half-plane* of the s-domain. Such a zero leads to a rise of the magnitude of the transfer function (this is generally not a bad thing), but the phase contribution of this zero is negative. This means that we are loosing phase margin, yet we push available gain to higher frequencies; in other words, we are degrading phase- and gain-margin!

A circuit-level interpretation of this effect is that while the Miller capacitor is wanted at the input of the amplifier (i.e., the feedback path), it also allows the input signal to pass to the output (i.e., the forward path). Since in normal operation the signal is inverted and in feed-forward mode it is not we have this unwanted effect of phase shift.

Luckily, there are several techniques to break the feed-forward path while keeping the feedback path (e.g., using a source follower to drive the output side of the Miller capacitor). For our purposes, we use a slightly simpler technique of adding a resistor in series to the Miller capacitor (see Figure 93 and Figure 94). Doing this we can modify the location of this zero, and even push it into the left-half-plane. This is excellent news for stability, as now this zero helps to improve gain- and phase-margin!

Repeating the calculation of the transfer function V_{out}/I_{in} including R_m we see that the zero location is changed and can be calculated as (the pole locations are also slightly changed due to the addition of R_m, but we will not discuss the resulting equations here)

$$s_z = \frac{g_m}{C_m(1 - g_m R_m)} \tag{49}$$

If we do not use the resistor (i.e., $R_m = 0$) then Equation 49 collapses to Equation 48. If $g_m R_m < 1$ then the zero stays in the right half-plane; if $g_m R_m > 1$ then the zero moves into the left half-plane (this is what we want). If $g_m R_m = 1$ then we have compensated the zero at it moves to $-\infty$; however, in practice exact compensation is not easy to establish across conditions, so we want to move the zero into the left half-plane. Adding a bit of margin want to size

$$R_\mathrm{m} > \frac{2}{g_\mathrm{m}}.$$

12.2 Common-Mode Regulation

In fully-differential (i.e., differential inputs and outputs) OTAs we have a new issue concerning common-mode voltage control: Depending on the feedback network around the OTA we might or might not have a defined dc operating point (common-mode wise). Think of the following scenario: If implement an integrator with an OTA then the feedback network from output to input consists of a capacitor. This means that the output of the OTA is loaded very high ohmic, and any small current mismatch between M_4/M_7 or M_6/M_8 (see Figure 93) will cause a strong deviation of the dc operating point at the output!

We can not accept that the dc operating points in a circuit are ill defined. We thus need a way to establish the dc operating point. Using the diode-resistive load for the differential pair in Figure 93 the dc operating point there is well-defined by the diode-connected $M_{3,5}$. However, the output stage is different, and we need to add circuitry to also control the dc operating point there.

One well-known way is to sense the common-mode voltage using two resistors (similar to $R_{1,2}$ in Figure 93), and compare this measured common-mode voltage to a reference voltage using an error amplifier. Then the output of the error amplifier controls the common-mode voltage, e.g., by driving the gates of $M_{7,8}$ in Figure 93.

> **!** Differential and Common-Mode Loops
>
> When using an error amplifier to regulate a common-mode point keep in mind that you need to check the differential and common-mode stability of these various loops! This can lead to tricky situations, especially under large-signal excitation where the common-mode sensing might not work as expected!
> In order to get a differential circuit stable one often has to make the common-mode loop faster than the differential loops. So simple, high-speed error amplifiers are an advantage.
> In summary, stability investigations are critically important for differential circuits, and never forget to check for common-mode stability as well!

Instead of a common-mode regulation loop (and all its complications regarding stability) often a common-mode setting is sufficient (after all, a somewhat imprecise setting of the dc points is good enough). A common-mode setting has the advantage that no error amplifier is required. We will also use this approach of a common-mode loop setting in the adapted differential OTA shown in Figure 95.

Resistors $R_{3,4}$ sense the output voltages and create a replica of the common-mode point (assuming $R_3 = R_4$). This common-mode point is connected to the gates of $M_{7,8}$ to essentially connect $M_{7,8}$ like a diode (only for common-mode operation); in differential mode, M_7 and M_8 act as current source, like the load of the differential pair $M_{3,5}$. However, since

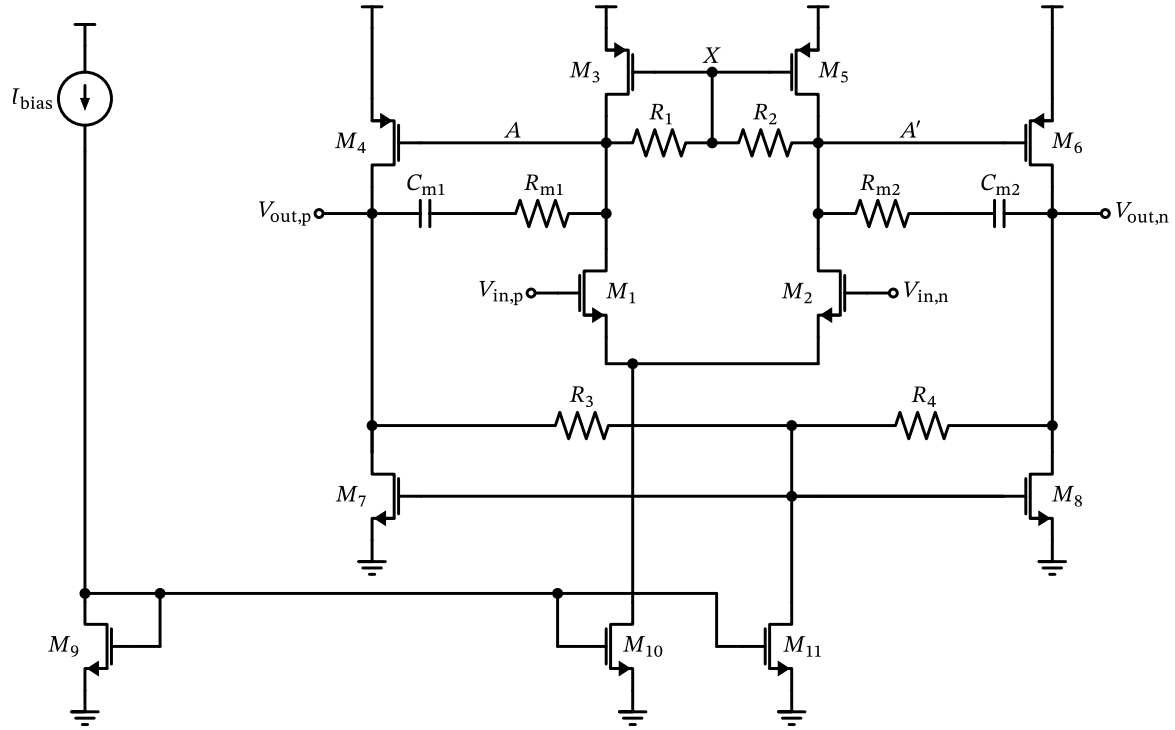


Figure 95: Differential two-stage OTA with resistive load, Miller compensation and output common-mode control.

we want to set the common-mode voltage at the output independently from $V_{\mathrm{GS7,8}}$ we also pull a current through $R_{3,4}$ to cause $V_{\mathrm{DS7,8}} \neq V_{\mathrm{GS7,8}}$; essentially, this is a diode connection including a voltage shift! The output common-mode voltage is then given by

$$V_{\mathrm{out,cm}} = V_{\mathrm{GS7,8}} + \frac{R_{3,4} I_{\mathrm{D11}}}{2}.$$

Modify Bias Points with Currents

Keep the technique shown in Figure 95 (using $R_{3,4}$ and M_{11}) in mind: You can always modify a bias point by injecting a dc current into a node, or by pulling a dc current out of a node (or do both to increase or lower the quiescent current through a resistor or transistor)! Since we likely have already current mirrors in the circuit it is usually a minor effort adding MOSFETs to create these bias currents.

13 An RC-OPAMP Filter

To be added in a future release.

14 Summary & Conclusion

By now, you should be familiar with the use of a schematic entry tool (Xschem) and circuit simulator (ngspice). You have learned the basic performance trade-offs, and the large- and small-signal behavior of the MOSFET. You can use the $g_{\mathrm{m}}/I_{\mathrm{D}}$ method to size MOSFET for class-A operation. You can design simple amplifiers based on OTA structures. In summary, you are on a good way to become a good analog or mixed-signal circuit designer!

Feedback

We hope you have enjoyed these lecture notes! If you have feedback, suggestions, additions, or corrections, please send us an e-mail, create a GitHub issue, or provide a GitHub pull request. Thank you in advance for your contributions!

Further Reading

For interested circuit designers, Chapter 5 of (Dorrer 2025) is recommended as a further read. It explains additional circuits, such as the beta-multiplier reference, an OTA with a telescopic input stage, and a push-pull output stage, designs them with the $g_{\mathrm{m}}/I_{\mathrm{D}}$ methodology, and simulates them in Xschem and with CACE. Hence, the design flow is the same as proposed in this course.

15 Appendix: Middlebrook's Method

If we want to do a closed-loop gain analysis (for stability or other investigations), we have the need to break the loop at one point, apply a stimulus, and monitor the response on the other end. By doing this we want to keep the loading (i.e. the impedance) on both ends similar to the original case. To achieve this, we break the loop at one point by inserting (1) an ac voltage source, and (2) attach an ac current source, as shown in Figure 96 and Figure 97. The derivation of this approach is presented in (Middlebrook 1975), and has the big advantage that loading is not changed compared to the closed-loop situation, and the bias points are also unchanged.

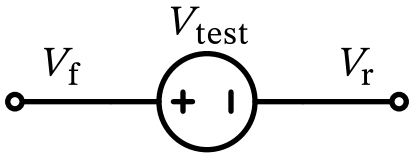


Figure 96: Middlebrook voltage loop gain simulation.

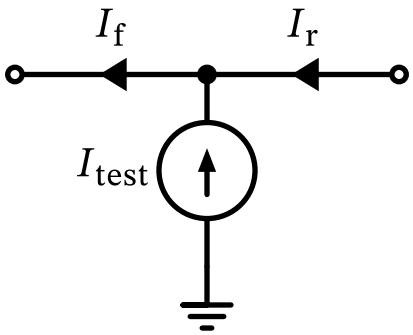


Figure 97: Middlebrook current loop gain simulation.

For both cases we do an ac analysis, and find the corresponding transfer functions T_{v} and T_{i} as

$$T_{\text{v}} = -\frac{V_{\text{r}}}{V_{\text{f}}}$$

and

$$T_{\text{i}} = -\frac{I_{\text{r}}}{I_{\text{f}}}.$$

Then, we can calculate the open-loop transfer function $T(s) = H_{\text{ol(s)}}$ as

$$T(s) = \frac{T_{\text{v}}T_{\text{i}} - 1}{T_{\text{v}} + T_{\text{i}} + 2} = \frac{V_{\text{r}}I_{\text{r}} - V_{\text{f}}I_{\text{f}}}{2V_{\text{f}}I_{\text{f}} - V_{\text{r}}I_{\text{f}} - V_{\text{f}}I_{\text{r}}}.$$

The four ac quantities V_{f}, V_{r}, I_{f}, and I_{r} we can readily find by circuit simulation or calculation.

Please note that Middlebrook's method works well for $T \gg 1$, so it will show inaccuracies at the crossover frequency of the open-loop gain at $T(s) \approx 0$. An improved method (slightly more complicated) can be found in (Tian et al. 2001).

16 Appendix: Useful Circuit Theorems

16.1 Miller's Theorem

Using Miller's theorem we can find the equivalent circuit of an impedance connected between two nodes, and we know the transfer function between these nodes. The given situation is shown in Figure 98, and the equivalent circuit is shown in Figure 99.

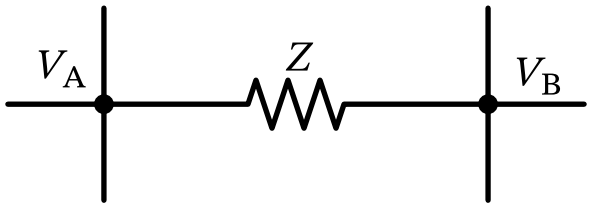


Figure 98: An impedance connected between two nodes A and B.

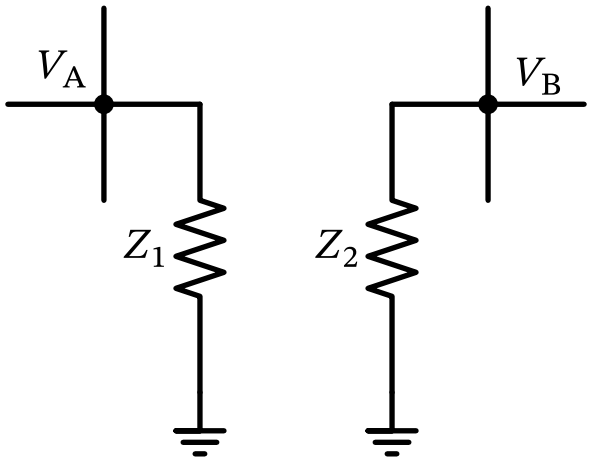


Figure 99: An equivalent circuit using Miller's theorem.

Using Miller's theorem (Sheikholeslami 2015) we can calculate

$$Z_1 = \frac{Z}{1-A} = \frac{Z}{1-V_\mathrm{B}/V_\mathrm{A}}$$

and

$$Z_2 = \frac{Z}{1-A^{-1}} = \frac{Z}{1-V_\mathrm{A}/V_\mathrm{B}}$$

to arrive at an equivalent circuit, given that $A = V_B/V_A$ is the voltage gain between nodes A and B. A derivation of this theorem is relative straightforward considering the current through Z when looking into the impedance from either node A or node B and calculating an equivalent impedance causing the same current.

Note that if $V_A = V_B$ then there is no current flow through Z, and accordingly the impedances $Z_1 = Z_2 = \infty$.

Miller's theorem can be quite handy when an impedance is strapped between two nodes, and we want to break this connection in a calculation, e.g., considering the effect of C_{GD} in a MOSFET.

> **Miller's Secret**
>
> Note that Miller's compensation is so much more than just making a big capacitor out of a small one. There are layers upon layers of subtlety, and huge hidden benefits which can be read in (Mangelsdorf 2025).

16.2 Bode's Noise Theorem

The total integrated noise of any (no matter how complicated) RLC network (interpreted as a one-port) is given by

$$\overline{V_n^2} = kT\left(\frac{1}{C_\infty} - \frac{1}{C_0}\right),$$

where C_∞ is the capacitance looking into the network with all resistors and inductors open-circuited, and C_0 is the capacitance looking into the circuit when all inductors and resistors are shorted (Pavan 2019).

Reference (Pavan 2019) is an excellent read deriving Bode's noise theorem from different angles.

17 Appendix: 5T-OTA Small-Signal Output Impedance

This section gives additional details to the analysis presented in Section 8.3. Here we provide the full calculation of the output impedance/conductance of the 5T-OTA for frequencies below the dominant pole, i.e. we neglect any capacitors.

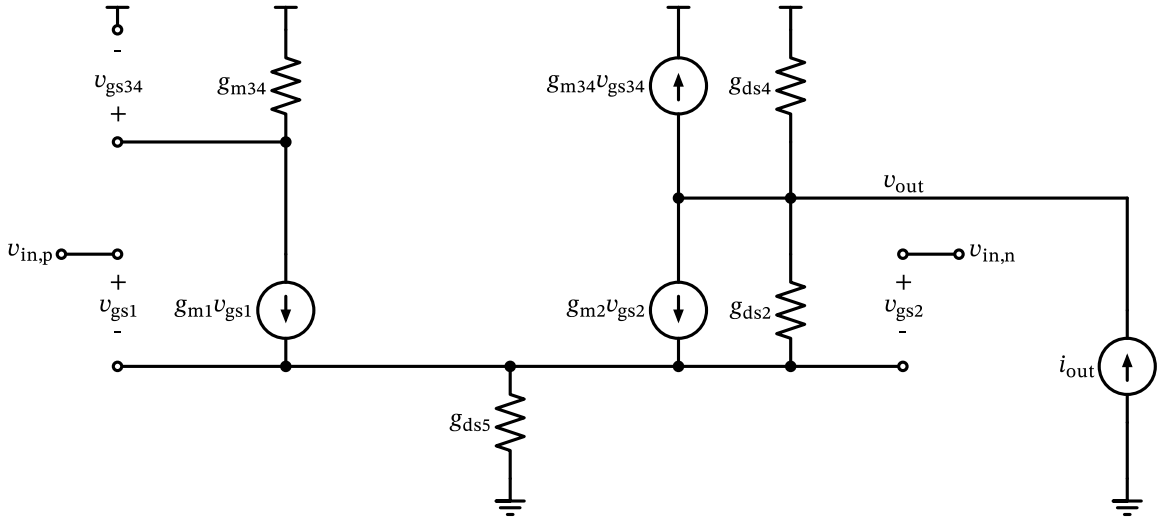


Figure 100: 5-transistor OTA small-signal model for output impedance calculations.

17.1 Open-Loop Configuration

For the open-loop case, the gates of M_1 and M_2 are tied to ground and thus, both v_{gs} are equal.

$$v_{in,p} = v_{in,n} = 0 \text{ V}$$
$$v_{gs1} = v_{gs2} \tag{50}$$

KCL at the output node:

$$i_{out} - g_{ds4}v_{out} - g_{m34}v_{gs34} - i_{g_{ds2}} - g_{m2}v_{gs2} = 0 \tag{51}$$

KCL at the tail node:

$$g_{m1}v_{gs1} + g_{m2}v_{gs2} + i_{g_{ds2}} + g_{ds5}v_{gs2} = 0$$

Using Equation 50 we can eliminate v_{gs1} and solve for $i_{g_{ds2}}$.

$$i_{g_{ds2}} = -\left(g_{m1} + g_{m2} + g_{ds5}\right) v_{gs2} \tag{52}$$

Furthermore, we need an expression for v_{gs34}. Ohm's law at the conductance g_{m34} will suffice.

$$v_{gs34} = -\frac{g_{m1}}{g_{m34}} v_{gs1} \tag{53}$$

KVL from the output node down to ground (over g_{ds2} and g_{ds5}) in combination with Equation 52 gives us an expression for v_{gs2}

$$v_{gs2} = -\frac{g_{ds2}}{g_{m1} + g_{m2} + g_{ds2} + g_{ds5}} v_{out} \tag{54}$$

Now, we can plug in all quantities into Equation 51. First, Equation 52 is inserted, which provides an expression for the current through the output conductance g_{ds2} of M_2.

$$i_{out} - g_{ds4}v_{out} - g_{m34}v_{gs34} + (g_{m1} + g_{ds5})\, v_{gs2} = 0$$

Second, v_{gs34} is substituted by Equation 53. Since we have assumed a matched pair of transistors for the current mirror comprised of M_3 and M_4, g_{m34} perfectly cancels out of the equation, and is effectively replaced by the transconductance g_{m1} of the input transistor M_1.

$$i_{out} - g_{ds4}v_{out} + (2g_{m1} + g_{ds5})\, v_{gs2} = 0$$

Third, Equation 54 gives as an expression for the last remaining unknown v_{gs2}. Thus, the factor in front of v_{out} defines the conductance at the output node.

$$i_{out} - \left[g_{ds4} + (2g_{m1} + g_{ds5})\, \frac{g_{ds2}}{g_{m1} + g_{m2} + g_{ds2} + g_{ds5}}\right] v_{out} = 0 \tag{55}$$

Before, we interpret this result, we use are assumption of matched input transistors ($g_{m12} = g_{m1} = g_{m2}$) and slightly rearrange the equation to give us more insight.

$$i_{out} - \left[g_{ds4} + \frac{g_{ds2} \cdot (2g_{m12} + g_{ds5})}{g_{ds2} + (2g_{m12} + g_{ds5})}\right] v_{out} = 0 \tag{56}$$

Now, we can identify the common equation of the total resistance of two parallel resistors. However, we are dealing with conductances here, so the same equation describes the total conductance of two conductances in series, while parallel conductances are simply summed. In parallel to g_{ds4}, there is effectively the series connection of g_{ds2} and $(2g_{m12} + g_{ds5})$ at work. If we apply the general assumption of $g_m \gg g_{ds}$, only the parallel connection of g_{ds4} and g_{ds2} remains. Therefore, moving $g_{ds2} + g_{ds4}$ in parallel to C_{load} in Section 8.3 was valid.

$$\frac{i_{out}}{v_{out}} \approx g_{ds4} + g_{ds2} \tag{57}$$

17.2 Closed-Loop Configuration

In contrast to the open-loop case, we keep the gate of M_1 connected to ground and tie the input of M_2 to the output node v_{out}.

$$v_{in,n} = v_{out} \tag{58}$$

KCL at the output node:

$$i_{out} - g_{ds4}v_{out} - g_{m34}v_{gs34} - g_{ds2}v_{gs2} - g_{m2}v_{gs2} = 0 \tag{59}$$

We use KVL from the output node down to ground to find an expression for v_{gs2}.

$$v_{gs2} = v_{out} + v_{gs1} \tag{60}$$

KCL at the tail node:

$$g_{m1}v_{gs1} + g_{m2}v_{gs2} + g_{ds2}v_{gs2} + g_{ds5}v_{gs2} = 0 \tag{61}$$

Using Equation 60 to substitute v_{gs2} in {#eq-app-vbufzout-kcl-vtail-cl} we find an equation for v_{gs1}.

$$v_{gs1} = -\frac{g_{m2} + g_{ds2}}{g_{m1} + g_{m2} + g_{ds2} + g_{ds5}} v_{out} \tag{62}$$

Again, we derive the output conductance by plugging Equation 60, Equation 53 and Equation 62 step by step into Equation 59. First, we use Equation 60 to eliminate v_{gs2}.

$$i_{out} - (g_{ds4} + g_{ds2} + g_{m2})\, v_{out} - g_{m34}v_{gs34} - (g_{ds2} + g_{m2})\, v_{gs1} = 0$$

Second, Equation 53 also holds for the closed-loop case and lets us eliminate v_{gs34}.

$$i_{out} - (g_{ds4} + g_{ds2} + g_{m2})\, v_{out} - (g_{ds2} + g_{m2} - g_{m1})\, v_{gs1} = 0$$

Third, we use Equation 62 to eliminate the remaining unknown v_{gs1}.

$$i_{out} - (g_{ds4} + g_{ds2} + g_{m2})\, v_{out} + (g_{ds2} + g_{m2} - g_{m1})\, \frac{g_{m2} + g_{ds2}}{g_{m1} + g_{m2} + g_{ds2} + g_{ds5}} v_{out} = 0$$

A more simpler result can be obtained, if we neglect g_{ds2} and g_{ds5} in Equation 62 first ($g_m \gg g_{ds}$) and then plug it into our main equation. Additionally, we use $g_{m12} = g_{m1} = g_{m2}$ to further simplify the equation.

$$i_{out} - \left(g_{ds4} + \frac{3}{2}g_{ds2} + g_{m12}\right) v_{out} \approx 0$$

If we apply $g_m \gg g_{ds}$ again, we arrive at the same result which was used for the noise calculation in Section 8.3, compare the expression for Y'_{load} given by Equation 24 .

$$i_{out} - (g_{m12})\, v_{out} \approx 0$$

18 Appendix: Linux Cheatsheet

The most useful commands for the Linux command line are:

- `ls` to list files and directories
- `cd` to change directory (e.g. `cd analog-circuit-design/xschem`)
- `cd ..` to move one directory level down
- `mkdir` to create a new directory (e.g. `mkdir my_directory`)
- `touch` to create an empty file (e.g. `touch file.txt`)
- `rm` to remove files (e.g. `rm file.txt`)
- `rm -r` to remove recursively, for example a directory (e.g. `rm -r my_directory`)

- `cp` to copy files (e.g. `cp file.txt destination`)
- `cp -r` to copy recursively a directory (e.g. `cp -r directory destination`)
- `mv` to rename files (e.g. `mv file.txt new_name.txt`)
- `mv` to move files into other directories (e.g. `mv file.txt directory`)
- `cat` to view the contents of a file (e.g. `cat file.txt`)
- `find` to search for files and directories (e.g. `find /path -name "*.txt"`)
- `nano` to edit file (e.g. `nano file.txt`)
- `Ctrl + C` to forcefully terminate a running process
- `htop` to open the "task manager"

More advanced commands can be found under https://www.geeksforgeeks.org/linux-commands-cheat-sheet.

19 Appendix: Xschem Cheatsheet

When opening Xschem, using `Help -> Keys` a pop-up windows comes up with many useful shortcuts. The most useful are:

19.0.0.1 Moving around in a schematic:

- `Cursor keys` to move around
- `Ctrl-e` to go back to parent schematic
- `e` to descend into schematic of selected symbol
- `i` to descend into symbol of selected symbol
- `f` full zoom on schematic
- `Shift-z` to zoom in
- `Ctrl-z` to zoom out

19.0.0.2 Editing schematics:

- `Del` to delete elements
- `Ins` to insert elements from library
- `Escape` to abort an operation
- `Ctrl-#` to rename components with duplicate names
- `c` to copy elements
- `Alt-Shift-l` to add wire label
- `Alt-l` to add label pin
- `m` to move selected objects
- `Shift-R` to rotate selected objects
- `Shift-F` to mirror / flip selected objects
- `q` to edit properties
- `Ctrl-s` to save schematic
- `t` to place a text
- `Shift-T` to toggle the `ignore` flag on an instance

- `u` to undo an operation
- `w` to draw a wire
- `Shift-W` draw wire and snap to close pin or net point
- `&` to join, break, and collapse wires
- `A` to make symbol from schematic
- `Alt-s` to reload the circuit if changes in a subcircuit were made

19.0.0.3 Viewing/Simulating Schematics

- `5` to only view probes
- `k` to highlight selected net
- `Shift-K` to unhighlight all nets
- `Shift-o` to toggle light/dark color scheme
- `s` to run a simulation
- `a & b` to add cursors to an in-circuit simulation graph
- `f` full zoom on y- or x-axis in in-circuit simulation graph

20 Appendix: ngspice Cheatsheet

Here is an unsorted list of useful ngspice settings and command:

20.1 Commands

- `ac dec|lin points fstart fstop` performs a small-signal ac analysis with either linear or decade sweep
- `dc sourcename vstart vstop vincr [src2 start2 stop2 incr2]` runs a dc-sweep, optionally across two variables
- `display` shows the available data vectors in the current plot
- `echo` can be used to display text, `$variable` or `$&vector`, can be useful for debugging
- `let name = expr` to create a new vector; `unlet vector` deletes a specified vector; access vector data with `$&vec`
- `linearize vec` linearizes a vector on an equidistant time scale, do this before an FFT; with `set specwindow=windowtype` a proper windowing function can be set
- `meas` can be used for various evaluations of measurement results (see ngspice manual for details)
- `noise v(output <ref>) src (dec|lin) pts fstart fstop` runs a small-signal noise analysis
- `op` calculates the operating point, useful for checking bias points and device parameters
- `plot expr vs scale` to plot something
- `print expr` to print it, use `print all` to print everything
- `remzerovec` can be useful to remove vectors with zero length, which otherwise cause issues when saving or plotting data
- `rusage` plot information about resource usage like memory

- `save all` or `save signal` specifies which data is saved during simulation; this lowers RAM usage during simulation and size of RAW file; do save before the actual simulation statement
- `setplot` show a list of available plots
- `set var = value` to set the value of a variable; use variable with `$var`; `unset var` removes a variable
- `set enable_noisy_r` to enable noise of behavioral resistors; usually, this is a good idea
- `shell cmd` to run a shell command
- `show : param`, like `show : gm` shows the g_m of all devices after running an operating point with `op`
- `spec` plots a spectrum (i.e. frequency domain plot)
- `status` shows the saved parameters and nodes
- `tf` runs a transfer function analysis, returning transfer function, input and output resistance
- `tran tstep tstop <tstart <tmax>>` runs a transient analysis until `tstop`, reporting results with `tstep` step size, starting to plot at `tstart` and performs time steps not larger then `tmax`
- `wrdata` writes data into a file in a tabular ASCII format; easy to further process
- `write` writes simulation data (the saved nodes) into a RAW file; default is binary, can be changed to ASCII with `set filetype=ascii`; with `set appendwrite` data is added to an existing file

20.2 Options

Use `option option=val option=val` to set various options; important ones are:

- `abstol` sets the absolute current error tolerance (default is 1pA)
- `gmin` is the conductance applied at every node for convergence improvement (default is 1e-12); this can be critical for very high impedance circuits
- `klu` sets the KLU matrix solver
- `list` print the summary listing of the input data
- `maxord` sets the numerical order of the integration method (default is 2 for Gear)
- `method` set the numerical integration method to `gear` or `trap` (default is `trap`)
- `node` prints the node table
- `opts` prints the option values
- `temp` sets the simulation temperature
- `reltol` set the relative error tolerance (default is 0.001 = 0.1%)
- `savecurrents` saves the terminal currents of all devices
- `sparse` sets the sparse matrix solver, which can run noise analysis, but is slower than `klu`
- `vntol` sets the absolute voltage error tolerance (default is 1µV)
- `warn` enables the printing of the SOA warning messages

20.3 Convergence Helper

- `option gmin` can be used to increase the conductance applied at every node
- `option method=gear` can lead to improved convergence
- `.nodeset` can be used to specify initial node voltage guesses
- `.ic` can be used to set initial conditions

21 Appendix: Circuit Designer's Etiquette

21.1 Prolog

A consistent naming and schematic drawing style, as well as VHDL/Verilog coding scheme, is a huge help in avoiding errors and increasing productivity. Even if just one person works on a design, the error rate is lowered. If multiple persons work together in a team, a consistent working style is a big help for smooth cooperation without misunderstanding each other's intentions. Consistency also helps to reuse existing blocks. In a well-done design, the documentation is included in the schematic/source code, so there is no searching for a piece of documentation somewhere else (which is often not found anyway).

21.2 Pins

- Name package pins (interfacing with the outside the IC) in **UPPERCASE**, and all internal signals in **lowercase**.
- Supply voltages like VDD/VCC and ground like VSS/GND need to start with either `VDD`, `VCC`, `VSS`, `VEE` or `GND`, plus a suitable suffix. Examples: `VDD1`, `VDD_AMP`, `vdd_ldo_out`, `VSS_ANA` (uppercase means connected to a pin, lowercase means a VDD is created on-chip by, e.g., an LDO).
- Preferred are `VDD`/`VSS` for CMOS and `VCC`/`GND` for bipolar circuits. In BiCMOS circuits `VDD`/`VSS` are preferred, as usually, the digital content is the major part.
- Digital signals in an analog schematic should start with `di_` (for digital input) or `do_` (for digital output). Example: `di_ctrl1`. In the rare case of a bi-directional digital signal `dio_` can be used.
- Name digital signals consistently: `di_pon` is active-high, `di_pon_b` is active-low (`_b` standing for the negating "bar"); as an alternative, this last signal could be named `di_disable`. `di_reset` is an active-high reset, but often a reset is active-low, so it needs to be named `di_reset_b` (an alternative is `di_resetn`).
- In mixed-voltage designs, it might be useful to append the voltage level of a signal to avoid connecting incompatible inputs and outputs. Example: `do_comp_1v2` or `di_poweron_3v3`.
- Digital buses always have the MSB to the left and LSB to the right. Example: `do_adc[7:0]`.

- Analog signals should start with a `v` for a voltage signal or `i` for a current signal. It is often useful to include a value for bias signals or make the naming meaningful. RF signals, which are often neither voltage nor current signals, start the name with `rf_`. Examples: Signal and pin names like `ibias_30u` (30uA of bias current), `vbg_1v2` (a bandgap voltage of 1.2V), `vin_p`, `v_filt_out_n`, and `rf_lna_i` speak for themselves.
- Appending analog signals with `_i` and `_o` might be useful if a clear direction is obvious in the signal flow. If a signal is bi-directional, it is better to skip `_io`. If using `_p` or `_n` in combination with `_i` or `_o` then use `_pi`/`_ni` or `_po`/`_no`.
- Consistently use pin types `input`, `output`, or `inout` to indicate signal flow. Power supply pins are of `inout` type.

21.3 Schematics

- In analog schematics, add a textual note about basic circuit performance. For example, in an amplifier, note things like suitable supply range, typical and w.c. current consumption, gain, GBW, input voltage range, PSRR, and other useful information.
- If a circuit has a quirk or is particularly clever, add a note on how it works, so others can understand the function without excessive analysis (reviewing a circuit should not be a brain teaser).
- Use provided borders or drawing templates for schematics, and fill the data in, like circuit designer name, date, change history, project name, etc.
- Use a versioning system for your data, and check in often. This avoids data loss, and going back to an earlier design stage is simple. `SVN` is often preferable to `GIT` for binary data.
- Draw uncluttered clear circuits. Ideally, the circuit function is apparent by inspection **quickly**. Everyone can obscure an inverter so that it takes 5 minutes to recognize it, but this is not a good design.
- Don't alter the standard grid setting while drawing schematics (also make sure that the pins in your drawn symbols are on the standard grid)! Off-grid schematic elements will haunt you and your colleagues forever!
- Once a schematic is finished, take the time to name component instances properly (you can use speaking names like `Rstab` or simply use `R1`, `R2`, etc.). Use iterated instances to clean up the circuit. Use wire bundles to clean up circuits where useful. A clever technique is to use bundles and iterated instances to efficiently draw large resistor ladders, for example (however, use with care).
- Avoid connection-by-name, as it makes the circuit hard to read. However, there is a fine line to not cluttering circuits. Signals with many connections (`vdd`, `vss`, `pon`, `pon_b`) are often better done with connection-by-name instead of drawing a wire.
- Some tools allow the use of colored wires, which might be used to mark signal paths, bias lines, etc. However, this should not be overdone; use it with care.
- If you add auxiliary elements like current probes, ensure they get proper treatment when creating the netlist for the LVS (some elements should be shorted, and some elements simply taken out). Ideally, only use a single schematic for simulation, LVS, etc. By using tool features this can usually be done, and avoids the need to keep multiple schematics of one block in sync.

- Use annotations in the schematics to (1) denote current levels in branches, (2) denote bias voltage levels, (3) explain the function of logic input signals, and (4) put in logic tables if not obvious.
- Add comments concerning the layout, like matching devices, certain considerations of placement, sensitive nodes, etc.
- Add simple ASCII diagrams for timing signals if useful.
- Name internal signals (signals connected to pins are anyway named like the port) in a meaningful way; this makes tracking signals in simulation or layout much easier (automatic net names like `net0032` are of not much help).
- Properly name instances, not just `I1` or `I2`; better is `amp1`, `inv2`, etc. (a descriptor in a tool output like `I1/I13/I5/net017` is not helpful; compare that to `adc1/bias/bg/vref_int`).
- On check-and-save, never ignore warnings; just fix them! They will annoy you and others forever and might flag critical design flaws.
- Name cells interpretably, ideally making the function clear already by the name. It is often useful to prefix or postfix a cell by the project name and design iteration. Example: In the project `GIGAPROJECT`, the cells which are changed in the second design step are prefixed with `g2_`, like `g2_amp_bias`. Of course, more letters as a project abbreviation are useful if a name collision is likely to happen.
- Cell names in lowercase are a good choice, as otherwise, capitalization leads to inconsistency in cell names. Use `_` to break words instead of `CamelCase`, like `amp_bias_startup`.
- When building a design, start with the hierarchy first; plan a suitable design structure, and define all interfaces. Implement simple behavioral models for every circuit block (either with controlled sources or using Verilog-A or VHDL/Verilog digital models). In this way, you can simulate the overall design early and find issues in the hierarchy or the interconnects. Then, populate the hierarchy with the detailed circuit designs in the leaf cells. At each point in the design process, you have a design that can be simulated, with some blocks as behavioral models and some blocks already designed. Try to avoid scattered circuit elements (digital or analog) in the hierarchy; it is better to push all components into the leaf cells.
- Avoid huge schematics, better break them down into smaller, maintainable, and self-contained blocks, and provide a simulation test bench for these simple blocks. In this way, later re-simulation across the hierarchy is easily possible.
- When building up the hierarchy, choose pin names and signal names as consistently as possible. Example: use the signal name `vref_int` when connecting two leaf cells with the pin names `vref_int_o` and `vref_int_i`.
- Avoid the excessive use of net breakers like small resistors, as they inhibit net tracing and can lead to simulation convergence issues. If a net breaker is needed (or a current should be probed) use a 0V dc voltage source.

21.4 Symbols

- Spend time drawing nice symbols! Ideally, the underlying circuit functionality is apparent by just looking at the symbol.

- Arrange the pins in a meaningful way.
- Group pins that belong together. An often useful arrangement is to locate the inputs on the left side, outputs on the right, digital control inputs at the bottom, and supplies at the top.
- Make the origin of a symbol in the top-left corner. In this way, symbols can be changed more easily, for example, by swapping out different versions of blocks.
- The cell name (and potentially library name) should be visible in the symbol, not only in the properties.

21.5 Design Robustness

- It is good practice to buffer incoming digital signals with a local inverter (connected to the local block supply) before connecting it to internal nodes. This improves the slew rate of the control signal and lowers the chance of unwanted cross-talk.
- Consider dummy elements for good matching, and try to make useful unit sizes of components. This will make the layout creation much smoother.
- The golden rule of good analog performance is good matching, and good matching is achieved by identical components (size, orientation, surroundings)! If the layout does not look nice (humans like symmetry), it will not perform well.
- Consider supply decoupling and bias voltage decoupling inside the cells. Often, dummy elements can be used for that. Be aware, however, of unwanted supply resonances (think bond wire `L` and decoupling `C`) and slow transients of bias nodes after disturbance.
- Always implement a proper power-down mode. Avoid floating nodes in off-mode. The better defined the on- as well as the off-mode are, the less the chance of leakage currents. Always simulate both modes (on and off), and also simulate a transient power-up of a circuit to identify issues with slow bias start or insufficient turn-off, or nasty feedback loop instabilities during transients.
- When drawing the first schematic, add parasitic capacitances to each node. If all nodes are labeled, a capacitor bank is easily put into one corner of the schematic with parasitic caps tied to the ground. Use 5fF as a starting value (and replace it later with the correct value from parasitic extraction). This accounts for some wiring parasitics in layout and helps to account for these layout impairments early in the design phase and later when simulating the schematic instead of the extracted netlist with parasitics.

21.6 Rules for Good Mixed-Signal and RF Circuits

- Separate analog and digital power supply, connect to package pins with multiple bond wires/bumps, and separate noisy and clean `vdd`/`vss` from each other!
- Prevent supply loops; keep `vdd` and `vss` lines close to each other (incl. bond wires and PCB traces)! This minimizes `L` and coupling factor `k`.

- Some prefer a massive (punched) ground plane, which is possible if you have enough metal levels. With a ground plane, the return path of a signal or supply line is just a few microns away.
- Use chip-internal decoupling capacitors, and decouple bias voltages to the **correct** potential (`vdd` or `vss`, or another node, depending on the circuit)!
- Use substrate contacts and guard rings to lower substrate crosstalk but use a quiet potential for connection; use triple-well if available! Connecting a guard-ring/substrate contact to a noisy supply is a prime noise injector (usually unwanted).
- Physically separate quiet and noisy circuits (at least by the epi thickness)!
- Reduce circuit noise generation as much as possible (avoid switching circuits if possible, use constant-current circuits instead, and use series/shunt regulators for supply isolation).
- Reduce sensitivity of circuits to interference (by using a fully differential design with high PSRR/CMRR, symmetrical layout parasitics, and good matching)!

21.7 VHDL/Verilog Coding Guide

These recommendations are specifically targeted at Verilog; however, they apply similarly to VHDL.

- Use automatic checkers (linters) to see whether your code contains errors or vulnerabilities. Commercial or open-source tools allow this, e.g., Icarus Verilog (`iverilog -g2005 -tnull FILE.v`) or Verilator (`verilator --lint-only -Wall FILE.v`).
- Write readable and maintainable code; use speaking variable names, and use a naming convention for inputs (ending with `_i`) and outputs (ending with `_o`). IO signals are using `_io`. Active-low signals have an `n` or `b` in their name (coming before the direction), like `reset_ni`. Use comments to explain the intention.
- With a synchronous reset reset-related racing conditions are often avoided. If an asynchronous reset is desirable (which is often the case), ensure the reset signals are free from race conditions.
- Module-local registers and wires could append `_w` (for Verilog `wire`) or `_r` (for Verilog `reg`) to make their function clear. This is not required in SystemVerilog where the unified type `logic` should be used.
- Use an `assign` statement for logic as this often is easier to read than an `always @(*)` block. The ternary operator `COND ? TRUE : FALSE` can help with conditional assignments and is often a better choice than a (nested) `if ... else` statement.
- Declare all outputs explicitly with either `reg` or `wire`.
- Use local parameter definitions with `localparam` in a module to make the code easier to follow. Name parameters in **UPPERCASE**.
- Take care to reset all registers to a defined state (in simulation and HW).
- Use the rule of "one file per module." The filename shall match the module declaration.
- Use `` `default_nettype none `` at the beginning of a file containing a module definition. After the module you can use `` `default_nettype wire ``. This will add a safety net against typos in signal names.

- In a logic assign block, use `assign @(*) begin ... end` instead of spelling out the signals in the sensitivity list. Forgetting a signal could lead to serious mismatches between simulation and HW.
- Make your code flexible by making bit widths and other values parameterized using a `localparam` or module parameter.
- Be cautious of implicit type conversions and bit-width adaptions; better make explicit conversions and match bit widths in assignments.
- Use only blocking assignments (`=`) in `always @(*)` blocks, and only non-blocking assignments (`<=`) in clocked `always @(posedge ...)` blocks.

A comprehensive coding style guide for Verilog/SystemVerilog can be found here, and it is highly recommended to follow it.

21.8 Further Reading

- Good information about drawing schematics, design testbenches, etc: https://circuit-artists.com
- Sutherland/Mills, *Verilog and SystemVerilog Gotchas - 101 Common Coding Erorrs and How to Avoid Them*, Springer, 2010
- B. Razavi, *The Analog Mind*, recurrent column in IEEE Solid-State Circuits Magazine

Baker, R. J. n.d. "Bad Circuit Design 3 - Compensating an Op-Amp." https://cmosedu.com/cmos1/bad_design/bad_design3/bad_design_3.htm.

Banba, H., H. Shiga, A. Umezawa, T. Miyaba, T. Tanzawa, S. Atsumi, and K. Sakui. 1999. "A CMOS Bandgap Reference Circuit with sub-1-V Operation." *IEEE Journal of Solid-State Circuits* 34 (5): 670–74. https://doi.org/10.1109/4.760378.

Blumlein, A. D. 1937. "Thermionic Valve Amplifying Circuit." U.S. Patent 2,185,367.

Carusone, Tony C., David Johns, and Kenneth Martin. 2011. *Analog Integrated Circuit Design*. Wiley.

Dorrer, Simon. 2025. "An Open-Source Adaptive Event-Based ADC for Bio-Signal Acquisition in 130 Nm CMOS." 2025. https://epub.jku.at/obvulihs/content/titleinfo/12118473?query=dorrer.

Eberlein, Matthias, Georgios Panagopoulos, and Harald Pretl. 2018. "A 40nW, Sub-IV Truly 'Digital' Reverse Bandgap Reference Using Bulk-Diodes in 16nm FinFET." *2018 IEEE Asian Solid-State Circuits Conference (A-SSCC)* 00 (November): 99–102. https://doi.org/10.1109/asscc.2018.8579306.

Gray, Paul R., Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer. 2009. *Analysis and Design of Analog Integrated Circuits*. Wiley.

Hellen, Edward H. 2003. "Verifying the diode–capacitor circuit voltage decay." *American Journal of Physics* 71 (8): 797–800. https://doi.org/10.1119/1.1578070.

Hu, Chenming. 2010. *Modern Semiconductor Devices for Integrated Circuits*. Pearson.

Jakoby, Bernhard. 2022. "Achieving signal power amplification using energetically passive devices." *E&i Elektrotechnik Und Informationstechnik* 139 (6): 477–84. https://doi.org/10.1007/s00502-022-01046-9.

Jespers, Paul G. A., and Boris Murmann. 2017. *Systematic Design of Analog CMOS Circuits: Using Pre-Computed Lookup Tables*. Cambridge University Press.

Kamath, B. Y. T., R. G. Meyer, and P. R. Gray. 1974. "Relationship Between Frequency Response and Settling Time of Operational Amplifiers." *IEEE Journal of Solid-State Circuits* 9 (6): 347–52. https://doi.org/10.1109/JSSC.1974.1050527.

Mangelsdorf, Chris. 2025. "Miller's Secret [Shop Talk: What You Didn't Learn in School]." *IEEE Solid-State Circuits Magazine* 17 (1): 21–27. https://doi.org/10.1109/MSSC.2024.3503792.

Maxwell, James Clerk. 1873. *A Treatise on Electricity and Magnetism*. Vol. 1. Clarendon Press.

Middlebrook, R. D. 1975. "Measurement of loop gain in feedback systems." *International Journal of Electronics* 38 (4): 485–512. https://doi.org/10.1080/00207217508920421.

Murmann, Boris. 2013. *Analysis and Design of Elementary MOS Amplifier Stages*. NTS Press.

Nagel, Laurence W. 1975. "SPICE2: A Computer Program to Simulate Semiconductor Circuits." PhD thesis, EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/1975/9602.html.

Neag, Marius, Raul Oneţ, István Kovács, and Paul Mărtari. 2015. "Comparative Analysis of Simulation-Based Methods for Deriving the Phase- and Gain-Margins of Feedback Circuits with Op-Amps." *IEEE Transactions on Circuits and Systems I: Regular Papers* 62 (3): 625–34. https://doi.org/10.1109/TCSI.2014.2370151.

Pavan, Shanthi. 2019. "An Alternative Approach to Bode's Noise Theorem." *IEEE Transactions on Circuits and Systems II: Express Briefs* 66 (5): 738–42. https://doi.org/10.1109/TCSII.2019.2907860.

Pelgrom, M. J. M., A. C. J. Duinmaijer, and A. P. G. Welbers. 1989. "Matching Properties of MOS Transistors." *IEEE Journal of Solid-State Circuits* 24 (5): 1433–39. https://doi.org/10.1109/JSSC.1989.572629.

Pretl, Harald, and Matthias Eberlein. 2021. "Fifty Nifty Variations of Two-Transistor Circuits: A Tribute to the Versatility of MOSFETs." *IEEE Solid-State Circuits Magazine* 13 (3): 38–46.

Razavi, Behzad. 2017. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill.

———. 2019. "The Low Dropout Regulator [A Circuit for All Seasons]." *IEEE Solid-State Circuits Magazine* 11 (2): 8–13. https://doi.org/10.1109/mssc.2019.2910952.

Rosenstark, S. 1984. "Loop Gain Measurement in Feedback Amplifiers." *International Journal of Electronics* 57 (3): 415–21. https://doi.org/10.1080/00207218408938921.

Sarpeshkar, R., T. Delbruck, and C. A. Mead. 1993. "White noise in MOS transistors and resistors." *IEEE Circuits and Devices Magazine* 9 (6): 23–29. https://doi.org/10.1109/101.261888.

Sheikholeslami, Ali. 2015. "Miller's Theorem [Circuit Intuitions]." *IEEE Solid-State Circuits Magazine* 7 (3): 9–10. https://doi.org/10.1109/mssc.2015.2446457.

———. 2025. "Noise and Distortion, Part IV [Circuit Intuitions]." *IEEE Solid-State Circuits Magazine* 17 (3): 29–31. https://doi.org/10.1109/MSSC.2025.3582840.

Tian, M., V. Visvanathan, J. Hantgan, and K. Kundert. 2001. "Striving for small-signal stability." *IEEE Circuits and Devices Magazine* 17 (1): 31–41.

Tsividis, Yannis, and Colin McAndrew. 2011. *Operation and Modeling of the MOS Transistor*. Oxford University Press.

Widlar, R. 1965. "Some Circuit Design Techniques for Linear Integrated Circuits." *IEEE Transactions on Circuit Theory* 12 (4): 586–90. https://doi.org/10.1109/tct.1965.1082512.

Widlar, R. J. 1971. "New developments in IC voltage regulators." *IEEE Journal of Solid-State Circuits* 6 (1): 2–7. https://doi.org/10.1109/jssc.1971.1050151.