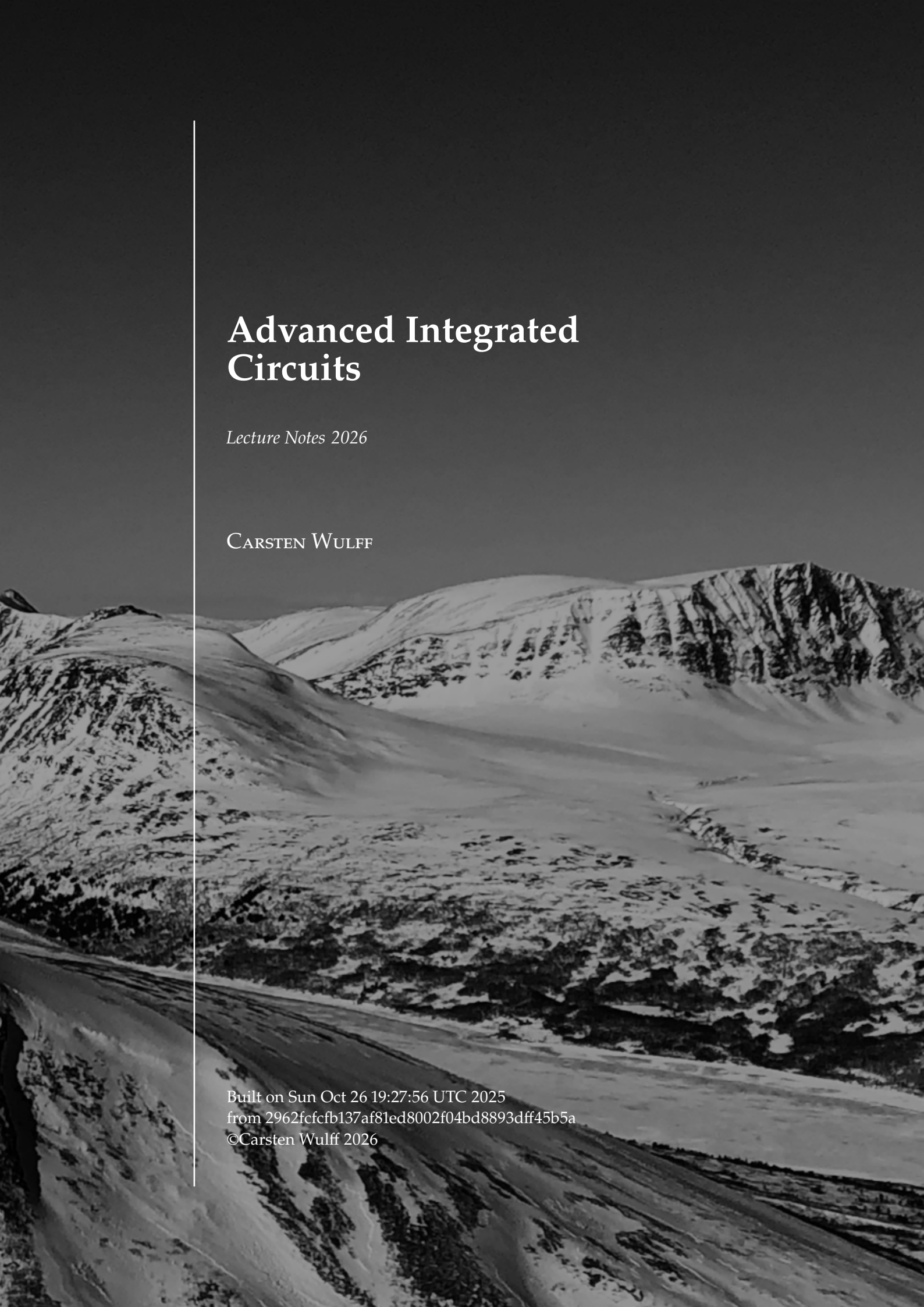
Advanced Integrated Circuits

Lecture Notes 2026

Carsten Wulff



Built on Sun Oct 26 19:27:56 UTC 2025
from 2962fcfcfb137af81ed8002f04bd8893dff45b5a

Contents

Background 1

In the spring of 2026 I lectured Advanced Integrated Circuits for the fifth time. I have an inherent need to make things better, and the course is no different.

In 2022 I noticed that little of what I had on slides, or said in lectures, made it into the student brain. That annoyed me, and I realized that things needed to change.

In 2023 I moved to complete open source project, and the project was without grade. There should have been a grade on the project.

I feel the lectures have gotten better. I did not take attendance in 2023, but there were 19 students that took the exam in 2024. I don't have all the dates, but an average attendance of 76 %.

Date	Attendance
2024-02-02	19
2024-02-09	17
2024-02-16	16
2024-03-01	14
2024-03-07	14
2024-03-15	12
2024-03-22	13
2024-04-12	16
2024-04-19	10

In 2025 there were 23 students that took the exam, however, 26 different students showed up to the lectures (more than a few times). The average attendance was around 80 %.

Wk	Attendance
2	21
3	21
4	23
5	20
6	22
7	24
9	20
9	24
11	20
12	17
14	16
15	14

In 2024 I finally felt I achieved a balance. I spent Thursday's preparing for the lecture, writing these notes, making a YouTube video (so I'll remember next year what I wanted to talk about). I passed 1k subscribers on Youtube. Friday's I had the lecture and the group work.

For the group work I forced students into groups, and I forced that they for the first 5-10 minutes do a check-in. That I need to do next year too.

For the check in, they had go around in the group and answer one of the following questions:

- What is one thing that is going on in your life (personal or professional)?
- What is one thing that you're grateful for right now?
- What is something funny that happened?

The check-in led to excellent team work for those students that showed up.

In 2025 I made a few tweaks. One change was the grading of the project, I used github actions to do the GDS,DRC,LVS,SIM and docs. The grading did not really work that well, although, it was a good way to get students to get the designs correct on github. The first milestones with the sim and the doc did not work. The last milestone actions worked well.

For 2026 I should do the following changes:

- Add GPT as part of learning: generate verification plan, checklists etc.
- Wait until after M0 for group selection
- Talk about layout early. Force full M0 tutorial
- Make them do TR layout early
- Re-introduce milestone 3
- Write a detailed project description and milestone and expectation description
- Reduce time for milestone 1. Maybe make a ready schematic hierarchy to force names? ideal OTA?
- Find a good sigma delta intro circuit
- Add to analog systemverilog
- Write about FFTs

I love programming and automation. Not much makes me more happy than using the same source (the slide markdowns), to generate the lecture notes, to translate into the book your looking at right now.

If you find an error in what I've made, then fork aic2026, fix , commit, push and create a pull request. That way, we use the global brain power most efficiently, and avoid multiple humans spending time on discovering the same error.

The Story of Jayn | 2

I want to tell you a story about Jayn. Jayn is an electron. Just one among countless others. There's nothing particularly special about Jayn, but Jayn plays an important role in our story.

At one point in Jayn's long life, without knowing it, Jayn would cause problems for me – but that's not where Jayn's story begins.

Jayn's story begins 13.8 billion years ago. Jayn popped into existence out of the emptiness, seemingly alone in the world. Well, not entirely alone. Surrounded by cousins and siblings, an ocean of elementary particles. There were quarks, there were electrons, and they all swam together in a sea of raw energy and matter.

Before long, the quarks got tired of floating around on their own. They clumped together, forming protons and neutrons – massive beasts compared to little Jayn. Jayn didn't like this new, crowded world. Jayn loved the freedom of zipping around the universe unbound. But eventually, Jayn felt a tug – a deep, irresistible attraction to one of those giant protons. And with that, Jayn became part of something new: a hydrogen atom.

Jayn lived in that hydrogen atom for billions of years. Over time, Jayn joined with other hydrogen atoms, coalescing into a star. Jayn basked in the warmth of the stars outer layers, content as the star burned brightly in the universe.

Eventually, though, Jayn's hydrogen atom drifted toward the star's core. There, something new happened. Instead of circling just one proton, Jayn now orbited two. A new proton had joined, forming helium. This meant there was space for another electron too – Jayn's first real companion.

They shared the same orbital space, something Jayn had never experienced before. Usually, if another electron came too close, one of them would yield into another energy state. But not this time. This was different. They could exist in the same place, with the same energy. How? Physicists would later call it "spin". Jayn didn't know what "spin" meant, it was weird, but Jayn liked it. It meant Jayn wasn't alone.

Jayn was happy in the helium atom, but the universe never stands still. Another proton came, then another, and eventually Jayn was part of a silicon atom, orbiting a nucleus with 14 protons and – of course – 14 electrons.

Jayn was no longer close to the nucleus. Now, Jayn was far out on the edge of the atom, in the outermost orbital shell. From there, Jayn could feel the presence of everything around – not just the self atom, but all the neighboring atoms as well.

One day, everything changed. The star exploded – a supernova – flung Jayn into the universe.

For a time, Jayn was adrift. Not alone, but not close to anything familiar. Eventually, the gravity of a forming planet – the one we now call Earth – caught Jayn's atom. The silicon atom joined with other atoms, bound together by Jayn's shared orbital shell between the atoms. It was easier to share the electrons than to be apart. The allure of the other atoms wasn't strong, but it was always there.

Jayn spent billions of years as part of the silicon atom, tumbling through Earth's oceans, sometimes bonding with other atoms, sometimes drifting free. One day, Jayn washed up on a beach, part of a grain of sand. And there Jayn stayed for a long, long time.

Now and then, out to sea Jayn went, then returned, living a simple, chaotic, quiet life.

But even quiet lives face change. One day, scooped up in a bucket, melted down, and turned into something new. Melted just meant more energy, Jayn had experienced that before, just more vibration. Sometimes, because of the vibrations, Jayn even had enough energy to escape the atom briefly before settling back down. But this time was different.

This time, Jayn became part of something incredibly uniform: a crystal lattice where every atom was another silicon atom, each in perfect order, with each atom sharing it's four outermost electrons in orbital shells with the neighbors. For the first time, Jayn could feel the full, equal pull of the electrons and nuclei around – like a invisible ocean of charges. It was electrifying.

Then one day, it happened. Another electron struck. It hit hard, and knocked Jayn from the orbital shell. For a brief moment, Jayn was free. Jayn flew through the crystal lattice, disoriented, then through something that was not a crystal lattice, but rather a jumbled mess of crystal pieces, until Jayn found another open spot, an empty energy state, where Jayn could settle again. It was strange but exhilarating.

And did you know? The presence of just that one electron – Jayn – in the gate oxide of a transistor was enough to shift the threshold voltage, change the flow of bias current, alter the frequency of an oscillator, cause my phone to loose the Bluetooth link to my door lock, and made me swear a number of times until the Bluetooth link finally reconnected, many, many, many seconds later.

And yet, Jayn's story doesn't end. Because Jayn, like all electrons, never really ends. Jayn may pop in and out of existence, but is always there – unchanged, identical to all the siblings.

The only differences between the electrons are where they are, their momentum and yes, their "spin". They are responsible for all chemical reactions in the universe. And their path through space-time described by the complex mathematics of the Schrödinger

equation. Or if you want to include relativity, the Lagrangian of Quantum Electrodynamics.

So ends the story of Jayn – our 13.8 billion-year-old troublemaker.

Introduction 3

Status: 1.0

3.1 Who

My name is

Carsten Wulff carstenw@ntnu.no

I finished my Masters in 2002, and did a Ph.D on analog-to-digital converters finished in 2008.

Since that time, I've had a three axis in my work/hobby life.

I work at Nordic Semiconductor where I've been since 2008. The first 7 years I did analog design (ADCs, DC/DCs, GPIO). The next 7 years I was the Wireless Group Manager. The Wireless group make most of the analog and RF designs for Nordic's short-range products. Now I'm the IC Scientist, and focus on technical issues with our integrated circuits that occur before we go into volume production.

I work at NTNU where I did a part time postdoc from 2014 - 2017. From 2020 I've been working on and teaching Advanced Integrated Circuits

I have a hobby trying to figure out how to make a new analog circuit design paradigm. The one we have today with schematic/simulation/layout/verification/simulation is too slow

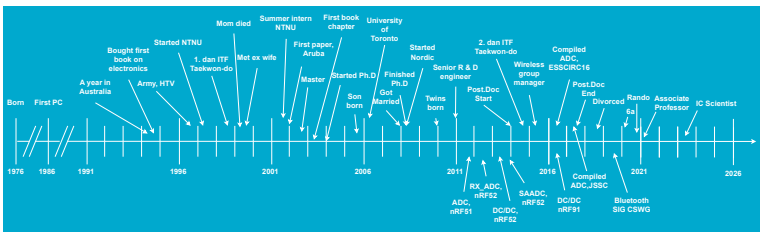


Figure 1: My life

3.2 How I see our roles

In Figure 2 you can see how I think about the research universe. There are things we know to be possible, things that actually are impossible (travel back in time, breaking thermodynamics, travel with a speed beyond light).

Between the impossible, and the possible, lies the unknown. I consider our roles as follows:

Professors: Guide students on what is impossible, possible, and hints on what might be possible

Ph.D students: Venture into the unknown and make something (more) possible

Master students: Learn all that is currently possible

Bachelor students: Learn how to make complicated into easy

Industry: Take what is possible, and/or complicated, and make it easy

Figure 2: Research Universe

3.3 I want you to learn the skills necessary to make your own ICs

In 2020 the global integrated circuit market was 437.7 billion dollars! The market is expected to grow to 1136 billion in 2028. Integrated circuits enable all technologies.

I will be dead in approximately 50 years, and will retire in approximately 20 years. Everything I know will be gone (except for the small pieces I've left behind in videos or written word)

Someone must take over, and to do that, they need to know most of what I know, and hopefully a bit more.

That's were some of you come in. Some of you will find integrated circuits interesting to make, and in addition, you have the stamina, patience, and brain necessary to learn some of the hardest topics in the world.

> Making integrated circuits (that work reliably) is not rocket science, it's much harder.

3.4 There will always be analog circuits, because the real world is analog

In this course, we'll focus on analog ICs, because the real world is analog, and all ICs must have some analog components, otherwise they won't work.

The steps to make integrated circuits is split in two. We have an analog flow, and a digital flow, as shown in Figure 3.

It's rare to find a single human that do both flows well. Usually people choose, and I think it's based on what they like and their personality.

If you like the world to be ordered, with definite answers, then it's likely that you'll find the digital flow interesting.

If you're comfortable with not knowing, and an insatiable desire to understand how the world *really* works at a fundamental level, then it's likely that you'll find analog flow interesting.

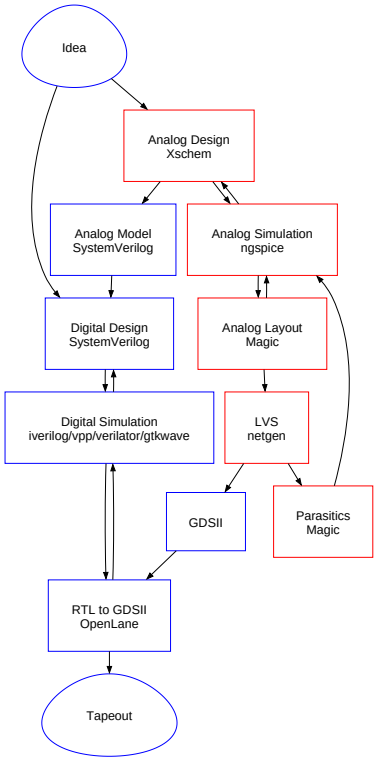


Figure 3: Analog and Digital design process

–>

3.5 Will you tape-out an IC?

Something that would make me really happy is if someone is able to tapeout an IC in this course.

It's now possible without signing an NDA or buying expensive software licenses.

In 2020 Google and Skywater joined forces to release a 130 nm process design kit to the public. In addition, they have fueled a renaissance of open source software tools.

Together with Efabless there are cheap alternatives, like tinytapeout, which makes it possible for a private citizen to tape-out their own integrated circuit.

3.5.1 What the team needs to know to design ICs

There are a multitude of tools and skills needed to design professional ICs. It's not likely that you'll find all the skills in one human, and even if you could, one human does not have sufficient bandwidth to design ICs with all it's aspects in a reasonable timeline

That is, unless we can find a way to make ICs easier.

The skills needed are

- *Project flow support*: **Confluence**, JIRA, risk management (DFMEA), failure analysis (8D)
- *Language*: **English**, **Writing English (Latex, Word, Email)**
- *Psychology*: Personalities, convincing people, presentations (Powerpoint, Deckset), **stress management (what makes your brain turn off?)**
- *DevOps*: **Linux**, bulid systems (CMake, make, ninja), continuous integration (bamboo, jenkins), **version control (git)**, containers (docker), container orchestration (swarm, kubernetes)
- *Programming*: Python, C, C++, Matlab *Since 1999 I've programmed in Python, Go, Visual BASIC, PHP, Ruby, Perl, C#, SKILL, Ocean, Verilog-A, C++, BASH, AWK, VHDL, SPICE, MATLAB, ASP, Java, C, SystemC, Verilog, Assembler, and probably a few I've forgotten.*
- *Firmware*: signal processing, algorithms, software architecture, security
- *Infrastructure*: **Power management**, **reset**, **bias**, **clocks**
- *Domains*: CPUs, peripherals, memories, bus systems

- *Sub-systems*: **Radio's**, **analog-to-digital converters**, **comparators**
- *Blocks*: **Analog Radio**, Digital radio baseband
- *Modules*: Transmitter, **receiver**, de-modulator, timing recovery, state machines
- *Designs*: **Opamps**, **amplifiers**, **current-mirrors**, adders, random access memory blocks, standard cells
- *Tools*: **schematic**, **layout**, **parasitic extraction**, synthesis, place-and-route, **simulation**, (System)Verilog, **netlist**
- *Physics*: transistor, pn junctions, quantum mechanics

3.5.2 Zen of IC design (stolen from Zen of Python)

When you learn something new, it's good to listen to someone that has done whatever it is before.

Here is some guiding principles that you'll likely forget.

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts (especially schematics).
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one **and preferably only one** obvious way to do it.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.

3.5.3 IC design mantra

To copy an old mantra I have on learning programming

> Find a problem that you really want to solve, and learn programming to solve it. There is no point in saying "I want to learn programming", then sit down with a book to read about programming, and expect that you will learn programming that way. It will not happen. The only way to learn programming is to do it, a lot. – Carsten Wulff

And run the perl program

```
s/programming/analog design/ig
```

3.5.4 Analog Design Process

- Define the problem, what are you trying to solve?
- Find a circuit that can solve the problem (papers, books)
- Find right transistor sizes. What transistors should be weak inversion, strong inversion, or don't care?
- Write a verification plan. Plan to simulate everything that could go wrong.
- Check operating region of transistors (.op)
- Check key parameters (.dc, .ac, .tran)
- Check function. Exercise all inputs. Check all control signals
- Check key parameters in all corners. Check mismatch (Monte-Carlo simulation)
- Do layout, and check it's error free. Run design rule checks (DRC). Check layout versus schematic (LVS)
- Extract parasitics from layout. Resistance, capacitance, and inductance if necessary.
- On extracted parasitic netlist, check key parameters in all corners and mismatch (if possible).
- If everything works, then your done.

On failure, go back as far as necessary

3.6 My Goal

Don't expect that I'll magically take information and put it inside your head, and you'll suddenly understand everything about making ICs.

You are the one that must teach yourself everything.

I consider my role as a guide, similar to a mountain guide. I can't carry you up the mountain, you need to walk up the mountain , but I know the safe path to take and increase the likelihood that you'll come back alive.

I want to:

- Enable you to read the books on integrated circuits
- Enable you to read papers (latest research)
- Correct misunderstandings on the topic
- Answer any questions you have on the chapters

I'm not a mind reader, I can't see inside your head. That means, you must ask questions. Only by your questions can I start to understand what pieces of information is missing from your head, or maybe somehow correct your understanding.

At the same time, and similar to a mountain guide, you should not assume I'm always right. I'm human, and I will make mistakes. And maybe you can correct my understanding of something. All I

care about is to *really* understand how the world works, so if you think my understanding is wrong, then I'll happily discuss.

3.7 Syllabus

The syllabus will be from Analog Integrated Circuit Design (CJM) and Circuits for all seasons.

These lecture notes are a supplement to the book. I try to give some background, and how to think about electronics. It's not my goal to repeat information that you can find in the book.

Buy a hard-copy of the book if you don't have that. Don't expect to understand the book by reading the PDF.

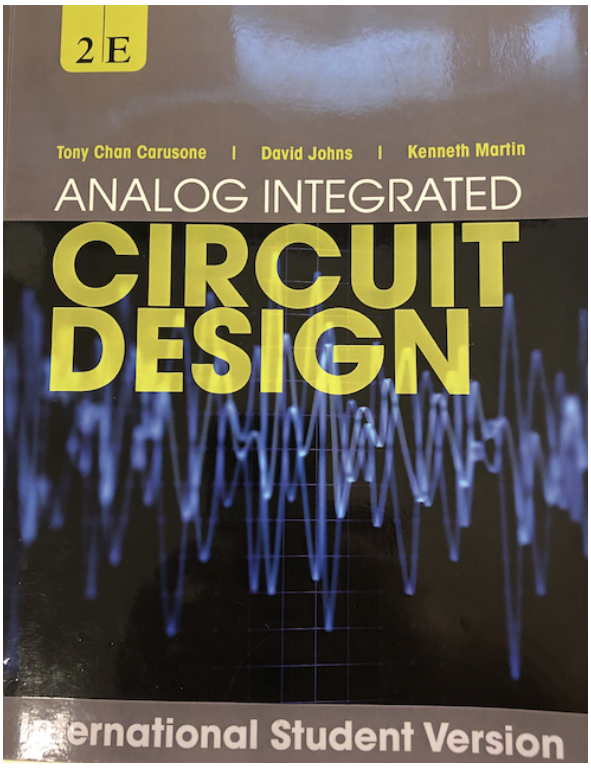


Figure 4: The book we'll use

3.8 Software

We'll use professional Open source software (xschem, ngspice, sky130A PDK, Magic VLSI, netgen)

I've made a rather detailed (at least I think so myself) tutorial on how to make a current mirror with the open source tools. I strongly recommend you start with that first.

Skywater 130 nm Tutorial

I've also made some more complex examples, that can be found at the link below. There are digital logic cells, standard transistors, and few other blocks.

aicex

A Refresher 4

Status: 0.8

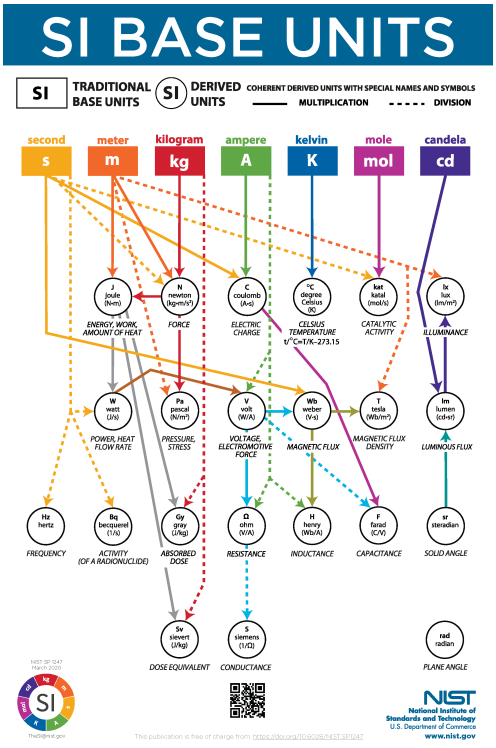
4.1 There are standard units of measurement

All known physical quantities are derived from 7 base units (SI units)

- second (s) : time
- meter (m) : space
- kg (kilogram) : weight
- ampere (A) : current
- kelvin (K) : temperature
- candela (cd) : luminous intensity

All other units (for example volts), are derived from the base units.

I don't go around remembering all of them, they are easily available online. When you forget the equation for charge (Q), voltage (V) and capacitance (C), look at the units below, and you can see it's $Q = CV$ *



* Although you do have to keep your symbols straight. We use "C" for Capacitance, but C can also mean Columbs. Context matters.

Figure 1: Si base units, from https://www.nist.gov/pml/owm/metric-si/si-units

4.2 Electrons

Electrons are fundamental, they cannot (as far as we know), be divided into smaller parts. Explained further in the standard model of particle physics

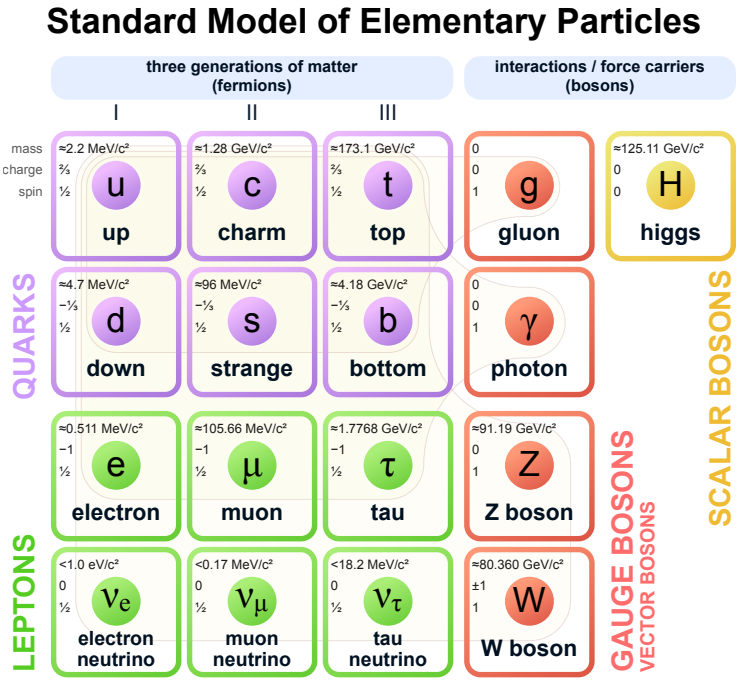


Figure 2: Standard model of particle physics, Wikipedia

Electrons have a negative charge of $q \approx 1.602 \times 10^{-19}$. The proton a positive charge. The two charges balance exactly! If you have a trillion electrons and a trillion protons inside a volume, the net external charge will be 0 (assuming we measure from some distance away). I find this fact absolutely incredible. There must be a fundamental connection between the charge of the proton and electron. It's insane that the charges balance out so exactly.

All electrons are the same, although the quantum state can be different.

An electron cannot occupy the same quantum state as another. This rule that applies to all Fermions (particles with spin of 1/2)

The quantum state of an electron is fully described by it's spin, momentum (p) and position in space (r).

4.3 Probability

The probability of finding an electron in a state as a function of space and time is

$$P = |\psi(r,t)|^2$$

, where ψ is named the probability amplitude, and is a complex function of space and time. In some special cases, it's

$$\psi(r,t) = Ae^{i(kr-\omega t)}$$

, where A is complex number, k is the wave number, r is the position vector from some origin, ω is the frequency and t is time.

The energy is $E = \hbar\omega$, where $\hbar = h/2\pi$ and h is Planck Constant and the momentum is $p = \hbar k$

The probability amplitude is also called the wave function. Type of wave function depends on the scenario, and does not have to take on the solution above. The possible wave functions are those equations that fits with the time evolution of quantum states given by the Schrodinger equation.

4.4 Uncertainty principle

We cannot, with ultimate precision, determine both the position and the momentum of a particle, the precision is

$$\sigma_x \sigma_p \geq \frac{\hbar}{2}$$

From the uncertainty (Unschärfe) principle we can actually estimate the size of the atom

4.5 States as a function of time and space

The time-evolution of the probability amplitude is

$$i\hbar\frac{d}{dt}\psi(r,t) = H\psi(r,t)$$

, where H is named the Hamiltonian matrix, or the energy matrix or (if I understand correctly) the amplitude matrix of the probability amplitude to change from one state to another.

For example, if we have a system with two states, a simplified version of two electrons shared between two atoms, as in H_2, or hydrogen gas, or co-valent bonds, then the Hamiltonian is a 2 x 2 matrix. And the ψ is a vector of $[\psi_1, \psi_2]$

Computing the solution to the Schrodinger Equation can be tricky, because you must know the number of relevant states to know the vector size of ψ and the matrix size of H. In addition, the H can be a function of time and space (I think).

Compared to the equations of electric fields, however, Schrodinger is easy, it's a set of linear differential equations.

4.6 Allowed energy levels in atoms

Solutions to Schrodinger result in quantized energy levels for an electron bound to an atom.

Take hydrogen, the electron bound to the proton can only exists in quantized energy levels. The lowest energy state can have two electrons, one with spin up, and one with spin down.

From Schrodinger you can compute the energy levels, which most of us did at some-point, although now, I can't remember how it was done. That's not important. The important is to internalize that the energy levels in bound electrons are discrete.

Electrons can transition from one energy level to another by external influence, i.e temperature, light, or other.

The probability of a state transition (change in energy) can be determined from the probability amplitude and Schrodinger.

4.7 Allowed energy levels in solids

If I have two silicon atoms spaced far apart, then the electrons can have the same spin and same momentum around their respective nuclei. As I bring the atoms closer, however, the probability amplitudes start to interact (or the dimensions of the Hamiltonian matrix grow), and there can be state transitions between the two electrons.

The allowed energy levels will split. If I only had two states interacting, the Hamiltonian could be

$$H = \begin{bmatrix} A & 0 \\ 0 & -A \end{bmatrix}$$

and the new energy levels could be

$$E_1 = E_0 + A$$

and

$$E_2 = E_0 - A$$

In a silicon crystal we can have trillions of atoms, and those that are close, have states that interact. **That's why crystals stay solids**. All chemical bonds are states of electrons interacting! Some are strong (co-valent bonds), some are weaker (ionic bonds), but it's all quantum states interacting.

The discrete energy levels of the electron transition into bands of allowed energy states.

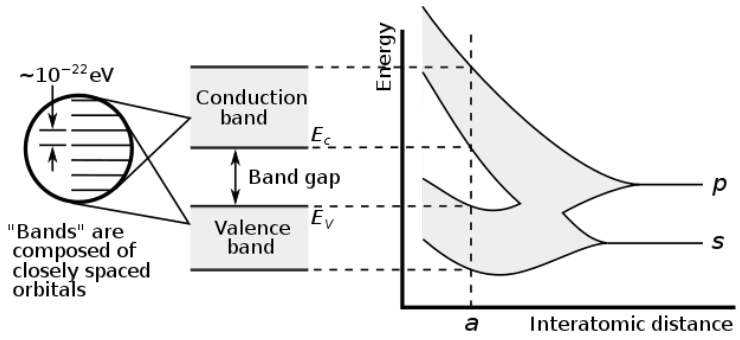


Figure 3: Electronic band structure, Wikipedia

For a crystal, the allowed energy bands is captured in the band structure

4.8 Silicon Unit Cell

A silicon crystal unit cell is a diamond faced cubic with 8 atoms in the corners spaced at 0.543 nm, 6 at the center of the faces, and 4 atoms inside the unit cell at a nearest neighbor distance of 0.235 nm.

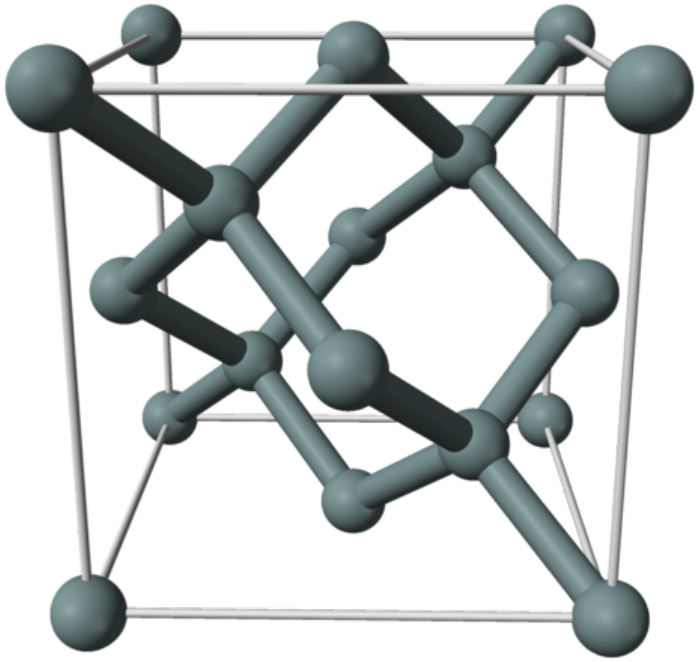


Figure 4: Silicon, Wikipedia

4.9 Band structure

The full band structure of a silicon unit cell is complicated, it's a 3 dimensional concept

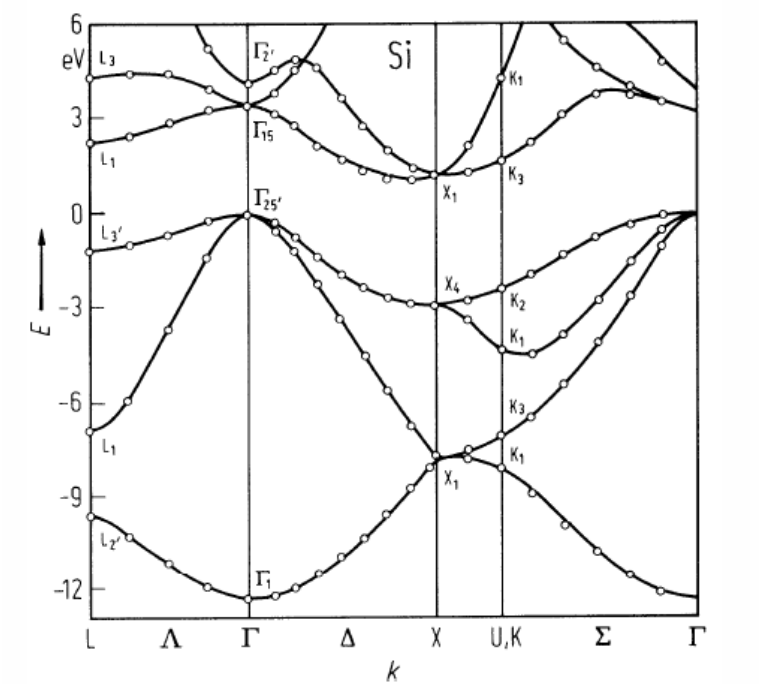


Figure 5: Silicon Band Structure

4.10 Valence band and Conduction band

For bulk silicon we simplify, and we think of two bands, the conduction band, and valence band

In the conduction band (E_C) is the lowest energy where electrons are free (not bound to atoms). The valence band (E_V) is the highest band where electrons are bound to silicon atoms.

The difference between E_C and E_V is a property of the material we've named the band gap.

$$E_G = E_C - E_V$$

4.11 Fermi level

From Wikipedia's Fermi level

> In band structure theory, used in solid state physics to analyze the energy levels in a solid, the Fermi level can be considered to be a hypothetical energy level of an electron, such that at thermodynamic equilibrium this energy level would have a 50% probability of being occupied at any given time

The Fermi level is closely linked to the Fermi-Dirac distribution

$$f(E) = \frac{1}{e^{(E-E_F)/kT} + 1}$$

If the energy of the state is more than a few kT away from the Fermi-level, then

$$f(E) \approx e^{(E_F-E)/kT}$$

The equation above is one of the reasons the structure $e^{E/kT}$ or $e^{qV/kT}$ shows up all over the place. You'll see it in the equations for current in a diode, $I_D = I_s(e^{qV_D/nkT} - 1)$, the subthreshold conduction of a mosfet $I_D \propto e^{qV_{gs}/nkT}$ and even the Arrhenius Equation $k = Ae^{-E_a/kT}$.

It seems like any time you have something related to chemical reactions (state transitions of electrons, breaking bonds, forming bonds), or current in solids, there is a relation to the equation above. To me, that makes sense.

The Fermi-Dirac function also explains why there are more free carriers, and reaction rates increase, at high temperature. The part of the equation that is $e^{-E/kT}$ will approach one at high temperatures.

4.12 Metals

In metals, the band splitting of the energy levels causes the valence band and conduction band to overlap.

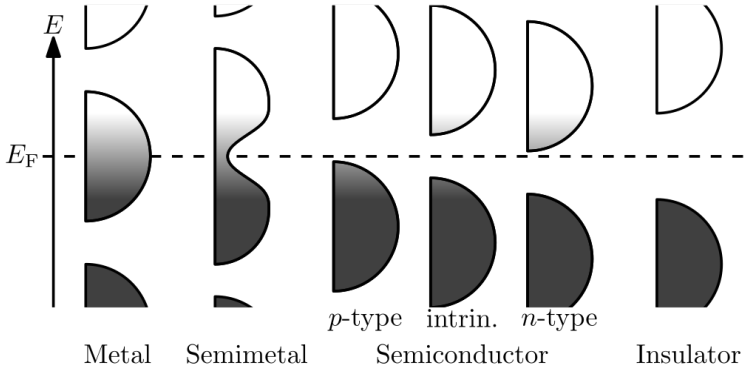


Figure 6: Band splitting in materials. Electronic Band Structure, Wikipedia

Electrons can easily transition between bound state and free state. As such, electrons in metals are shared over large distances, and there are many electrons readily available to move under an applied field, or difference in electron density. That's why metals conduct well.

4.13 Insulators

In insulating materials the difference between the conduction band and the valence band is large. As a result, it takes a large energy to excite electrons to a state where they can freely move.

That's why glass is transparent to optical frequencies. Visible light does not have sufficient energy to excite electrons from a bound state.

That's also why glass is opaque to ultra-violet, which has enough energy to excite electrons out of a bound state.

Based on these two pieces of information you could estimate the bandgap of glass.

```
from scipy import constants
#- We must use the "correct" units for planck's constant to get energy in eV
h = constants.physical_constants["Planck constant in eV/Hz"][0]
c = constants.physical_constants["speed of light in vacuum"][0]

lambda_optical = 450e-9
e_optical = h * c/lambda_optical

lambda_ultra = 380e-9
e_ultra = h * c/lambda_ultra

print("Bandgap of glass is above %.2f eV, maybe around %.2f eV " %(e_optical,e_ultra))
```

4.14 Semiconductors

In silicon the bandgap is lower than an insulator, approximately

$$E_G = 1.12\ eV$$

At room temperature, that allows a small number of electrons to be excited into the conduction band, leaving behind a "hole" in the valence band.

4.15 Band diagrams

A band diagram or energy level diagrams shows the conduction band energy and valence band energy as a function of distance in the material.

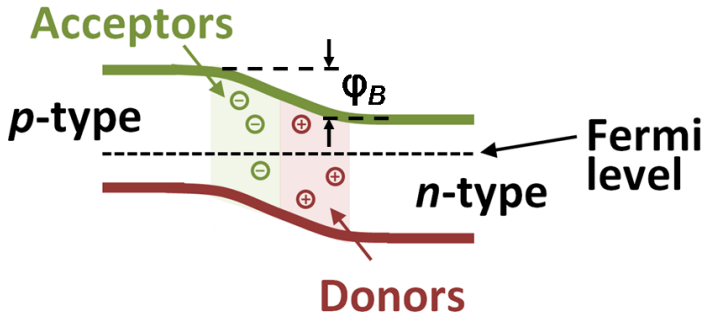


Figure 7: Band diagram of a PN junction, Wikipedia

The horizontal axis is the distance in the material, the vertical axis is the energy.

4.16 Density of electrons/holes

There are two components needed to determine how many electrons are in the conduction band. The density of available states, and the probability of an electron to be in that quantum state.

The probability is the Fermi-Dirac distribution. The density of available states is a complicated calculation from the band-structure of silicon.

For details see the Diodes chapter.

$$n_e = \int_{E_C}^{\infty} N(E) f(E) dE$$

The Fermi level is assumed to be independent of energy level, so we can write

$$n_e = e^{E_F/kT} \int_{E_C}^{\infty} N(E) e^{-E/kT} dE$$

for the density of electrons in the conduction band.

4.17 Fields

There are equations that relate electric field, magnetic field, charge density and current density to each-other.

$$\oint_{\partial\Omega} \mathbf{E} \cdot d\mathbf{S} = \frac{1}{\epsilon_0} \iiint_V \rho \cdot dV$$

,relates net electric flux to net enclosed electric charge

$$\oint_{\partial\Omega} \mathbf{B} \cdot d\mathbf{S} = 0$$

,relates net magnetic flux to net enclosed magnetic charge

$$\oint_{\partial\Sigma} \mathbf{E} \cdot d\ell = -\frac{d}{dt} \iint_{\Sigma} \mathbf{B} \cdot d\mathbf{S}$$

,relates induced electric field to changing magnetic flux

$$\oint_{\partial\Sigma} \mathbf{B} \cdot d\ell = \mu_0 \left(\iint_{\Sigma} \mathbf{J} \cdot d\mathbf{S} + \epsilon_0 \frac{d}{dt} \iint_{\Sigma} \mathbf{E} \cdot d\mathbf{S} \right)$$

,relates induced magnetic field to changing electric flux and to current

These are the Maxwell Equations, and are non-linear time dependent differential equations.

Under the best of circumstances they are fantastically hard to solve! But it's how the real world works.

4.18 Permittivity and Permeability

The permittivity of free space is defined as

$$\epsilon_0 = \frac{1}{\mu_0 c^2}$$

, where c is the speed of light, and μ_0 is the vacuum permeability, which, in SI units, is now

$$\mu_0 = \frac{2\alpha}{q^2}\frac{h}{c}$$

, where α is the fine structure constant.

4.19 Quantum electrodynamics

The quantum electrodynamics (QED) is a full description of interactions between light and matter. The equations describe both quantum mechanical effects, electromagnetism and is in agreement with special relativity.

The equations are rather complicated, but it's based on Lagrangian physics. Maxwell's equations actually fall out of the QED Lagrangian when one assumes local phase symmetry.

The QED Lagrangian is

$$\mathcal{L} = \bar{\psi}[i\hbar c\gamma^\mu \partial_\mu - mc^2]\psi - q[\bar{\psi}\gamma^\mu\psi]A_\mu - \frac{1}{16\pi}F_{\mu\nu}F^{\mu\nu}$$

For more information, have a look at Electromagnetism as a Gauge Theory

4.20 Voltage

The electric field has units voltage per meter, so the electric field is the derivative of the voltage as a function of space.

$$E = \frac{dV}{dx}$$

4.21 Current

Current has unit A and charge C has unit As, so the current is the number of charges passing through a volume per second.

The current density J has units A/m^2 and is often used, since we can multiply by the surface area of a conductor, if the current density is uniform.

$$I = \text{Area} \times J$$

4.22 Drift current

Charge carriers (electrons, holes, ions) in an electric field will give rise to a drift current.

We know from Newtons laws that force equals mass times acceleration

$$\vec{F} = m\vec{a}$$

If we assume a zero, or constant magnetic field, the force on a particle is

$$\vec{F} = q\vec{E}$$

The current density is then

$$\vec{J} = q\vec{E} \times n \times \mu$$

where n is the charge density, and μ is the mobility (how easily the charges move) and has units m^2/Vs

Assuming

$$E = V/m$$

, we could write

$$J = \frac{C}{m^3}\frac{V}{m}\frac{m^2}{Vs} = \frac{C}{s}m^{-2}$$

So multiplying by an area A with unit meters squared

$$I = qn\mu AV$$

and we can see that the conductance

$$G = qn\mu A$$

, and since

$$G = 1/R$$

, where R is the resistance, we have

$$I = GV \Rightarrow V = RI$$

Or Ohms law

4.23 Diffusion current

A difference in charge density will give rise to a diffusion current. The current density is

$$J = -qD_n \frac{d\rho}{dx}$$

,where D_n is a diffusion constant, and ρ is the charge density.

4.24 Why are there two currents?

I struggled with the concepts diffusion current and drift current for a long time. Why are there two types of current? It was when I read The Schrödinger Equation in a Classical Context: A Seminar on Superconductivity I realized that the two types of current come directly from the Schrodinger equation, there is one component related to the electric field (potential energy) and a component related to the momentum (kinetic energy).

In the absence of an electric field electrons will still jump from state to state set by the probabilities of the Hamiltonian. If there are more electrons in an area, then it will seem like there is an average movement of charges away from that area. That's how I think about drift and diffusion currents. We can kinda see it from the Schrödinger equation below.

$$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial^2 x}\psi(x,t) + V(x)\psi(x,t) = i\hbar\frac{\partial}{\partial t}\psi(x,t)$$

4.25 Currents in a semiconductor

Both electrons, and holes will contribute to current.

Electrons move in the conduction band, and holes move in the valence band.

Both holes and electrons can only move if there are available quantum states.

For example, if the valence band is completely filled (all states filled), then there can be no current.

To compute the total current in a semiconductor one must compute

$$I = I_{n_{drift}} + I_{n_{diffusion}} + I_{p_{drift}} + I_{p_{diffusion}}$$

where n denotes electrons, and p denote holes.

4.26 Resistors

We can make resistors with many materials. The behavior of the charge carrier may be different between materials.

In metal the dominant carrier depends on the metal, but it's usually electrons. As such, one can often ignore the hole current.

In a semiconductor the dominant carrier depends on the Fermi level in relation to the conduction band and valence band.

If the Fermi level is close to the valence band the dominant carrier will be holes. If the Fermi level is close to the conduction band, the dominant carrier will be electrons.

That's why we often talk about "majority carriers" and "minority carriers", both are important in semiconductors.

4.27 Capacitors

A capacitor resists a change in voltage

$$I = C\frac{dV}{dt}$$

and store energy in an electric field between two conductors with an insulator between.

4.28 Inductors

An inductor resist a change in current

$$V = L\frac{dI}{dt}$$

and store energy in the magnetic fields in a loop of a conductor.

Diodes 5

Status: 1.0

5.1 Why

Diodes are a magical * semiconductor device that conduct current in one direction. It's one of the fundamental electronics components, and it's a good idea to understand how they work.

If you don't understand diodes, then you won't understand transistors, neither bipolar, or field effect transistors.

A useful feature of the diode is the exponential relationship between the forward current, and the voltage across the device.

To understand why a diode works it's necessary to understand the physics behind semiconductors.

This paper attempts to explain in the simplest possible terms how a diode works †

5.2 Silicon

Integrated circuits use single crystalline silicon. The silicon crystal is grown with the Czochralski method which forms a ingot that is cut into wafers. The wafer is a regular silicon crystal, although, it is not perfect.

A silicon crystal unit cell, as seen in Figure 1 is a diamond faced cubic with 8 atoms in the corners spaced at 0.543 nm, 6 at the center of the faces, and 4 atoms inside the unit cell at a nearest neighbor distance of 0.235 nm.

* It doesn't stop being magic just because you know how it works. Terry Pratchett, The Wee Free Men

† Simplify as much as possible, but no more. Einstein

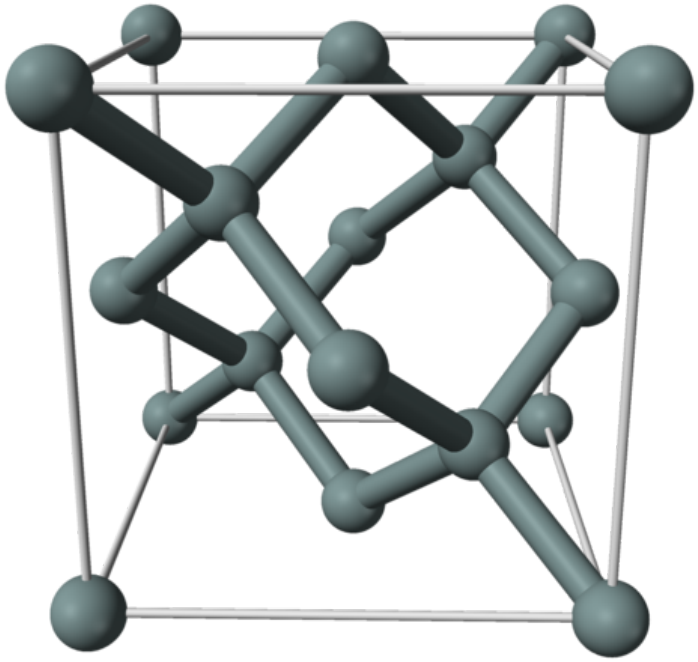


Figure 1: Silicon crystal unit cell

As you hopefully know, the energy levels of an electron around a positive nucleus are quantized, and we call them orbitals (or shells). For an atom far away from any others, these orbitals, and energy levels are distinct. As we bring atoms closer together, the orbitals start to interact, and in a crystal, the distinct orbital energies split into bands of allowed energy states. No two electrons, or any Fermion (spin of 1/2), can occupy the same quantum state. We call the outermost "shared" orbitial, or band, in a crystal the valence band. Hence covalent bonds.

If we assume the crystal is perfect, then at 0 Kelvin all electrons will be part of covalent bonds. Each silicon atom share 4 electrons with its neighbors. What we really mean when we say "share 4 electrons" is that the wave-functions of the outer orbitals interact, and we can no longer think of the orbitals as belonging to either of the silicon nuclei. All the neighbors atoms "share" electrons, and nowhere is there an vacant state, or a hole, in the valence band.

If such a crystal were to exist, where there were no holes in the valence band, and a net neutral charge, the crystal could not conduct any drift current. Electrons would move around continuously, swapping states, but there could be no net drift of charge carriers.

In an atom, or a crystal, there are also higher energy states where the carriers are "free" to move. We call these energy levels, or bands of energy levels, conduction bands. In singular form "conduction band", refers to the lowest available energy level where the electrons are free to move.

Due to imperfectness of the silicon crystal, and non-zero temperature, there will be some electrons that achieve sufficient energy to jump to the conduction band. The electrons in the conduction band leave vacant states, or holes, in the valence band.

Electrons can move both in the conduction band, as free electrons, and in the valence band, as a positive particle, or hole. Both bands can support drift and diffusion currents.

5.3 Intrinsic carrier concentration

The intrinsic carrier concentration of silicon, or the density of free electrons and holes at a given temperature, is given by

$$n_i = \sqrt{N_c N_v} e^{-E_g/(2kT)} \tag{1}$$

where E_g is the bandgap energy of silicon (approx 1.12 eV), k is Boltzmann's constant, T is the temperature in Kelvin, N_c is the density of states in conduction band, and N_v is the density of states in the valence band.

The density of states are

$$N_c = 2\left[\frac{2\pi kTm_n^*}{h^2}\right]^{3/2} \quad N_v = 2\left[\frac{2\pi kTm_p^*}{h^2}\right]^{3/2}$$

where h is Planck's constant, m_n^* is the effective mass of electrons, and m_p^* is the effective mass of holes.

Leave it to engineers to simplify equations beyond understanding. Equation (1) is complicated, and the density of states includes the effective mass of electrons and holes, which is a parameter that depends on the curvature of the band structure. To engineers, this is too complicated, and n_i has been simplified so it "works" in daily calculation.

Through engineering simplification, however, physics understanding is lost.

In [1] they claim the intrinsic carrier concentration is a constant, although they do mention n_i doubles every 11 degrees Kelvin.

In BSIM 4.8 [2] the intrinsic carrier concentration is

$$n_i = 1.45e10\frac{TNOM}{300.15}\sqrt{\frac{T}{300.15}}\exp^{21.5565981-\frac{E_g}{2kT}}$$

Comparing the three models in Figure 2, we see the shape of BSIM and the full equation is almost the same, while the "doubling every 11 degrees" is just wrong.

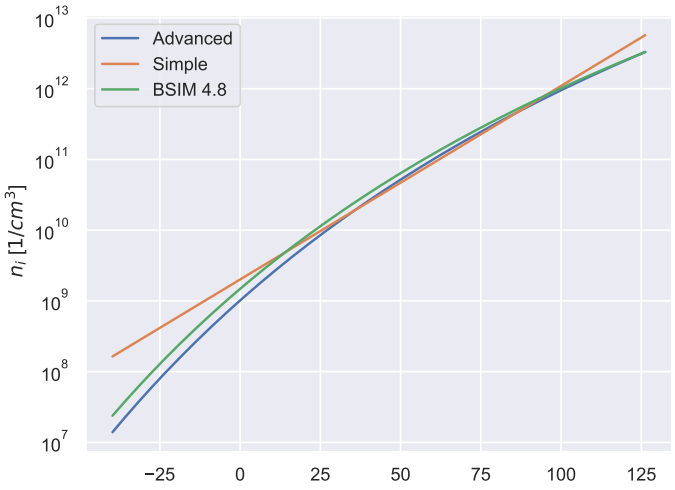


Figure 2: Intrinsic carrier concentration versus temperature

At room temperature the intrinsic carrier consentration is approximately $n_i = 1 \times 10^{16}$ carriers/m^3.

That may sound like a big number, however, if we calculate the electrons per um^3 it's $n_i = \frac{1\times10^{16}}{(1\times10^6)^3}$ carriers/μm^3 < 1, so there are really not that many free carriers in intrinsic silicon.

From Figure 2 we can see that n_i changes greatly as a function of temperature, but the understanding "why" is not easy to get from "doubling every 11 degrees". To understand the temperature behavior of diodes, we must understand Eq (1).

So where does Eq (1) come from? I find it unsatisfying if I don't understand where things come from. I like to understand why there is an exponential, or effective mass, or Planck's constant. If you're like me, then read the next section. If you don't care, and just want to memorize the equations, or indeed the number of intrinsic carrier concentration number at room temperature, then skip the next section.

5.4 It's all quantum

There are two components needed to determine how many electrons are in the conduction band. The density of available states, and the probability of an electron to be in that quantum state.

For the density of states we must turn to quantum mechanics. The probability amplitude of a particle can be described as

$$\psi = Ae^{i(\mathbf{k}\mathbf{r}-\omega t)}$$

where k is the wave number, and ω is the angular frequency, and $\mathbf{r}$ is a spatial vector.

In one dimension we could write $\psi(x,t) = Ae^{i(kx-\omega t)}$

In classical physics we described the Energy of the system as

$$\frac{1}{2m}p^2 + V = E$$

where $p = mv$, m is the mass, v is the velocity and V is the potential.

In the quantum realm we must use the Schrodinger equation to compute the time evolution of the Energy, in one space dimension

$$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial^2 x}\psi(x,t) + V(x)\psi(x,t) = i\hbar\frac{\partial}{\partial t}\psi(x,t)$$

where m is the mass, V is the potential, $\hbar = h/2\pi$.

We could rewrite the equation above as

$$\widehat{H}\psi(x,t) = i\hbar\frac{\partial}{\partial t}\psi(x,t) = \widehat{E}\psi(x,t)$$

where $\widehat{H}$ is sometimes called the *Hamiltonian* and is an operator, or something that act on the wave-function. In Feynman's Lectures on Physics Feynman called the Hamiltonian the *Energy Matrix* of a system. I like that better. The $\widehat{E}$ is the energy operator, something that operates on the wave-function to give the Energy.

We could re-arrange

$$[\widehat{H} - \widehat{E}]\psi(r,t) = 0$$

This is an equation with at least 5 unknowns, the space vector in three dimensions, time, and the energy matrix $\widehat{H}$.

The dimensions of the energy matrix depends on the system. The energy matrix further up is for one free electron. For an atom, the energy matrix will have more dimensions to describe the possible quantum states.

I consider all energy matricies as infinite dimensions, but most state transitions are so unlikely that they can be safely ignored.

I was watching Quantum computing in the 21st Century and David Jamison mentioned that the largest system we could today compute would be a system with about 30 electrons.

We know exactly how the equations of quantum mechanics appear to be, and they've proven extremely successful, we must make simplifications before we can predict how electrons behave in

complicated systems like the silicon lattice with approximately 0.7 trillion electrons per cube micro meter. You can check the calculation

$$\left[\frac{1\ \mu\text{m}}{0.543\ \text{nm}}\right]^3 \times 8 \text{ atoms per unit cell} \times 14 \text{ electrons per atom}$$

5.4.1 Density of states

To compute "how many Energy states are there per unit volume in the conduction band", or the "density of states", we start with the three dimensional Schrodinger equation for a free electron

$$-\frac{\hbar^2}{2m}\nabla^2\psi = E\psi$$

I'm not going to repeat the computation here, but rather paraphrase the steps. You can find the full derivation in Solid State Electronic Devices.

The derivation starts by computing the density of states in the k-space, or momentum space,

$$N(dk) = \frac{2}{(2\pi)^p}dk$$

Where p is the number of dimensions (in our case 3).

The band structure $E(k)$ is used to convert to the density of states to a function of energy $N(E)$. The simplest band structure, and an approxmiation of the lowest conduction band is

$$E(k) = \frac{\hbar^2 k^2}{2m^*}$$

where m^* is the effective mass of the particle. It is within this effective mass that we "hide" the complexity of the actual three-dimensional crystal structure of silicon.

The effective mass when we compute the density of states is

$$m^* = \frac{\hbar^2}{\frac{d^2E}{dk^2}}$$

as such, the effective mass depends on the localized band structure of the silicon unit cell, and depends on direction of movement, strain of the silicon lattice, and probably other things.

In 3D, once we use the above equations, one can compute that the density of states per unit energy is

$$N(E)dE = \frac{2}{\pi^2}\frac{{m^*}^{3/2}}{\hbar^2}E^{1/2}dE$$

In order to find the number of electrons, we need the probability of an electron being in a quantum state, which is given by the Fermi-Dirac distribution

$$f(E) = \frac{1}{e^{(E-E_F)/kT}+1} \tag{2}$$

where E is the energy of the electron, E_F is the Fermi level or checmical potential, k is Boltzmann's constant, and T is the temperature in Kelvin.

Fun fact, the Fermi level difference between two points is what you measure with a voltmeter.

If the $E - E_F > kT$, then we can start to ignore the +1 and the probability reduces to

$$f(E) = \frac{1}{e^{(E-E_F)/kT}} = e^{(E_F-E)/kT}$$

A few observiation on the Fermi-Dirac distribution. If the Energy of a state is at the Fermi level, then $f(E) = \frac{1}{2}$, or a 50 % probability of being occupied.

In a metal, the Fermi level lies within a band, as the conduction band and valence band overlap. As a result, there are a bunch of free electrons that can move around. Metal does not have the same type of covalent bonds as silicon, but electrons are shared between a large part of the metal structure. I would also assume that the location of the Fermi level within the band structure explains the difference in conductivity of metals, as it would determined how many electrons are free to move.

In an insulator, the Fermi level lies in the bandgap between valence band and conduction band, and usually, the bandgap is large, so there is a low probability of finding electrons in the conduction band.

In a semiconductor we also have a bandgap, but much lower energy than an insulator. If we have thermal equilibrium, no external forces, and we have an un-doped (intrinsic) silicon semiconductor, then the fermi level E_F lies half way between the conduction band edge E_C and the valence band edge E_V.

The bandgap is defined as the $E_C - E_V = E_g$, and we can use that to get $E_F - E_C = E_C - E_g/2 - E_C = -E_g/2$. This is why the bandgap of silicon keeps showing up in our diode equations.

The number of electrons per delta energy will then be given by

$$N_e dE = N(E)f(E)dE$$

, which can be integrated to get

$$n_e = 2\left(\frac{2\pi m^* kT}{h^2}\right)^{3/2} e^{(E_F - E_C)/kT}$$

For intrinsic silicon at thermal equlibrium, we could write

$$n_0 = 2\left(\frac{2\pi m^* kT}{h^2}\right)^{3/2} e^{-E_g/(2kT)} \tag{3}$$

As we can see, Equation (3) has the same coefficients and form as the computation in Equation (1). The difference is that we also have to account for holes. At thermal equilibrium and intrinsic silicon $n_i^2 = n_0 p_0$

5.4.2 How to think about electrons (and holes)

I've come to the realization that to imagine electrons as balls moving around in the silicon crystal is a bad mental image.

For example, for a metal-oxide-semiconductor field effect transistor (MOSFET) it is not the case that the electrons that form the inversion layer under strong inversion come from somewhere else. They are already at the silicon surface, but they are bound in covalent bonds (there are literaly trillions of bound electrons in a typical transistor).

What happens is that the applied voltage at the gate shifts the energy bands close to the surface (or bends the bands in relation to the Fermi level), and the density of carriers in the conduction band in that location changes, according to the type of derivations above.

Once the electrons are in the conduction band, then they follow the same equations as diffusion of a gas, Fick's law of diffusion. Any charge density concentration difference will give rise to a diffusion current given by

$$J_{\text{diffusion}} = -qD_n \frac{\partial \rho}{\partial x} \tag{4}$$

where J is the current density, q is the charge, ρ is the charge density, and D is a diffusion coefficient that through the Einstein relation can be expressed as $D = \mu kT$, where mobility $\mu = v_d/F$ is the ratio of drift velocity v_d to an applied force F.

To make matters more complicated, an inversion layer of a MOSFET is not in three dimensions, but rather a two dimensional electron gas, as the density of states is confined close to the silicon surface. As such, we should not expect the mobility of bulk silicon to be the same as the mobility of a MOSFET transistor.

5.5 Doping

We can change the property of silicon by introducing other elements, something we've called doping. Phosphor has one more electron than silicon, Boron has one less electron. Injecting these elements into the silicon crystal lattice changes the number of free electron/holes.

These days, we usually dope with ion implantation, while in the olden days, most doping was done by diffusion. You'd paint something containing Boron on the silicon, and then heat it in a furnace to "diffuse" the Boron atoms into the silicon.

If we have an element with more electrons we call it a donor, and the donor concentration N_D.

The main effect of doping is that it changes the location of the Fermi level at thermal equilibirum. For donors, the Fermi level will shift closer to the conduction band, and increase the probabilty of free electrons, as determined by Equation (2).

Since the crystal now has an abundance of free electrons, which have negative charge, we call it n-type.

If the element has less electrons we call it an acceptor, and the acceptor concentration N_A. Since the crystal now has an abundance of free holes, we call it p-type.

The doped material does not have a net charge, however, as it's the same number of electrons and protons, so even though we dope silicon, it does remain neutral.

The doping concentrations are larger than the intrinsic carrier concentration, from maybe 10^{21} to 10^{27} carriers/m^3. To separate between these concentrations we use $p-, p, p+$ or $n-, n, n+$.

The number of electrons and holes in a n-type material is

$$n_n = N_D \ , p_n = \frac{n_i^2}{N_D}$$

and in a p-type material

$$p_p = N_A \ , n_p = \frac{n_i^2}{N_A}$$

In a p-type crystal there is a majority of holes, and a minority of electrons. Thus we name holes majority carriers, and electrons minority carriers. For n-type it's opposite.

5.6 PN junctions

Imagine an n-type material, and a p-type material, both are neutral in charge, because they have the same number of electrons and protons. Within both materials there are free electrons, and free holes which move around constantly.

Now imagine we bring the two materials together, and we call where they meet the junction. Some of the electrons in the n-type will wander across the junction to the p-type material, and visa versa. On the opposite side of the junction they might find an opposite charge, and might get locked in place. They will become stuck.

After a while, the diffusion of charges across the junction creates a depletion region with immobile charges. Where as the two materials used to be neutrally charged, there will now be a build up of negative charge on the p-side, and positive charge on the n-side.

5.6.1 Built-in voltage

The charge difference will create a field, and a built-in voltage will develop across the depletion region.

The density of free electrons in the conduction band is

$$n = \int_{E_C}^{\infty} N(E)f(E)dE$$

, where $N(E)$ is the density of states, and $f(E)$ is a probability of a electron being in that state (Equation (2)).

We could write the density of electrons on the n-side as

$$n_n = e^{E_{F_n}/kT} \int_{E_C}^{\infty} N_n(E)e^{-E/kT}dE$$

since the Fermi level is independent of the energy state of the electrons (I think).

The density of electrons on the p-side could be written as

$$n_p = e^{E_{F_p}/kT} \int_{E_C}^{\infty} N_p(E)e^{-E/kT}dE$$

If we assume that the density of states, $N_n(E)$ and $N_p(E)$ are the same, and the temperature is the same, then

$$\frac{n_n}{n_p} = \frac{e^{E_{F_n}/kT}}{e^{E_{F_p}/kT}} = e^{(E_{F_n} - E_{F_p})/kT}$$

The difference in Fermi levels is the built-in voltage multiplied by the unit charge.

$$E_{F_n} - E_{F_p} = q\Phi$$

and by substituting for the minority carrier concentration on the p-side we get

$$\frac{N_A N_D}{n_i^2} = e^{q\Phi_0/kT}$$

or rearranged to

$$\Phi_0 = \frac{kT}{q} ln\left(\frac{N_A N_D}{n_i^2}\right)$$

5.6.2 Current

The derivation of current is a bit involved, but let's try.

The hole concentration on the p-side and n-side could be written as

$$\frac{p_p}{p_n} = e^{-q\Phi_0/kT}$$

The negative sign is because the built in voltage is positive on the n-type side

Asssume that $-x_{p0}$ is the start of the junction on the p-side, and x_{n0} is the start of the junction on the n-side.

Assume that we lift the p-side by a voltage qV

Then the hole concentration would change to

$$\frac{p(-x_{p0})}{p(x_{n0})} = e^{q(V-\Phi_0)/kT}$$

while on the n-side the hole concentration would be

$$\frac{p(x_{n0})}{p_n} = e^{qV/kT}$$

So the excess hole concentration on the n-side due to an increase of V would be

$$\Delta p_n = p(x_{n0}) - p_n = p_n \left(e^{qV/kT} - 1 \right)$$

The diffusion current density, given by Equation (4) states

$$J(x_n) = -qD_p \frac{\partial \rho}{\partial x}$$

Thus we need to know the charge density as a function of x. I'm not sure why, but apparently it's

$$\partial \rho(x_n) = \Delta p_n e^{-x_n/L_p}$$

where L_p is a diffusion length. I think the equation above, the exponential decay as a function of length, is related to the probabilty of electron/hole recombination, and how the rate of recombination must be related to the exceess hole concentration, as such related to Exponential decay.

Anyhow, we can now compute the current density, and need only compute it for $x_n = 0$, so you can show it's

$$J(0) = q \frac{D_p}{L_p} p_n \left(e^{qV/kT} - 1 \right)$$

which start's to look like the normal diode equation. The p_n is the minority concentration of holes on the n-side, which we've before estimated as $p_n = \frac{n_i^2}{N_D}$

We've only computed for holes, but there will be electron transport from the p-side to the n-side also.

We also need to multiply by the area of the diode to get current from current density. The full equation thus becomes

$$I = qAn_i^2 \left(\frac{1}{N_A} \frac{D_n}{L_n} + \frac{1}{N_D} \frac{D_p}{L_p} \right) \left[e^{qV/kT} - 1 \right]$$

where A is the area of the diode, D_n,D_p is the diffusion coefficient of electrons and holes and L_n,L_p is the diffusion length of electrons and holes.

Which we usually write as

$$I_D = I_S(e^{\frac{V_D}{V_T}} - 1), \text{ where } V_T = kT/q$$

5.6.3 Forward voltage temperature dependence

We can rearrange I_D equation to get

$$V_D = V_T \ln\left(\frac{I_D}{I_S}\right)$$

and at first glance, it appears like V_D has a positive temperature coefficient. That is, however, wrong.

First rewrite

$$V_D = V_T \ln I_D - V_T \ln I_S$$

$$\ln I_S = 2\ln n_i + \ln Aq\left(\frac{D_n}{L_n N_A} + \frac{D_p}{L_p N_D}\right)$$

Assume that diffusion coefficient [‡], and diffusion lengths are independent of temperature.

That leaves n_i that varies with temperature.

$$n_i = \sqrt{B_c B_v} T^{3/2} e^{\frac{-E_g}{2kT}}$$

where

$$B_c = 2\left[\frac{2\pi k m_n^*}{h^2}\right]^{3/2} \quad B_v = 2\left[\frac{2\pi k m_p^*}{h^2}\right]^{3/2}$$

$$2\ln n_i = 2\ln\sqrt{B_c B_v} + 3\ln T - \frac{V_G}{V_T}$$

with $V_G = E_G/q$ and inserting back into equation for V_D

$$V_D = \frac{kT}{q}(\ell - 3\ln T) + V_G$$

Where ℓ is temperature independent, and given by

$$\ell = \ln I_D - \ln\left(Aq\frac{D_n}{L_n N_A} + \frac{D_p}{L_p N_D}\right) - 2\ln\sqrt{B_c B_v}$$

[‡] From the Einstein relation $D = \mu kT$ it does appear that the diffusion coefficient increases with temperature, however, the mobility decreases with temperature. I'm unsure of whether the mobility decreases with the same rate though.

From equations above we can see that at 0 K, we expect the diode voltage to be equal to the bandgap of silicon. Diodes don't work at 0 K though.

Although it's not trivial to see that the diode voltage has a negative temperature coefficient, if you do compute it as in vd.py, then you'll see it decreases.

The slope of the diode voltage can be seen to depend on the area, the current, doping, diffusion constant, diffusion length and the effective masses.

Figure 3 shows the V_D and the deviation of V_D from a straight line. The non-linear component of V_D is only a few mV. If we could combine V_D with a voltage that increased with temperature, then we could get a stable voltage across temperature to within a few mV.

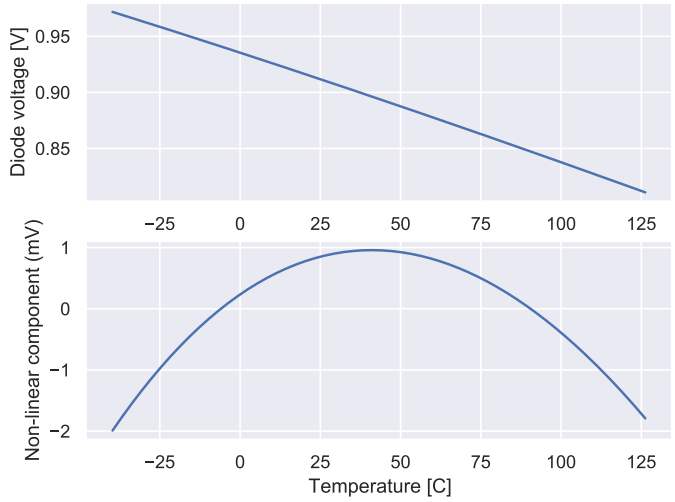


Figure 3: Diode forward voltage as a function of temperature

5.6.4 Current proportional to temperature

Assume we have a circuit like Figure 4.

Here we have two diodes, biased at different current densities. The voltage on the left diode V_{D1} is equal to the sum of the voltage on the right diode V_{D2} and voltage across the resistor R_1. The current in the two diodes are the same due to the current mirror. A such, we have that

$$I_S e^{\frac{qV_{D1}}{kT}} = N I_S e^{\frac{qV_{D2}}{kT}}$$

Taking logarithm of both sides, and rearranging, we see that

$$V_{D1} - V_{D2} = \frac{kT}{q} \ln N$$

Or that the difference between two diode voltages biased at different current densities is proportional to absolute temperature.

In the circuit above, this ΔV_D is across the resistor R_1, as such, the $I_D = \Delta V_D / R_1$. We have a current that is proportional to temperature.

If we copied the current, and sent it into a series combination of a resistor R_2 and a diode, we could scale the R_2 value to give us the exactly right slope to compensate for the negative slope of the V_D voltage.

The voltage across the resistor and diode would be constant over temperature, with the small exception of the non-linear component of V_D.

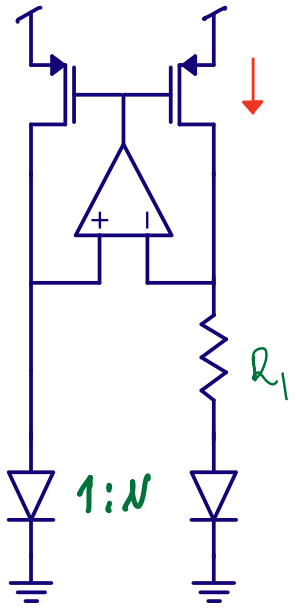


Figure 4: Circuit to generate a current proportional to kT

5.7 Equations aren't real

> Nature does not care about equations. It just is.

We know, at the fundamental level, nature appears to obey the mathematics on quantum mechanics, however, due to the complexity of nature, it's not possible today (which is not the same as impossible), to compute exactly how the current in a diode works.

We can get close, by measuring a diode we know well, and hope that the next time we make the same diode, the behavior will be the same.

As such, I want to warn you about the "lies" or "simplifications" we tell you. Take the diode equation above, some parts, like the intrinsic carrier concentration n_i has roots directly from quantum mechanics, with few simplifications, which means it's likely solid truth, at least for a single unit cell.

But there is no reason nature should make all unit cells the same, and infact, we know they are not the same, we put in dopants. As we scale down to a few nano-meter transistors the simplification that "all unit cells of silicon are the same, and extend to infinity" is no longer true, and must be taken into account in how we describe reality.

Other parts, like the exact value of the bandgap E_g, the diffusion constant D_p or diffusion length L_p are macroscopic phenomena, we can't expect them to be 100 % true. The values would be based on measurement, but not always exact, and maybe, if you rotate your diode 90 degrees on the integrated circuit, the values could be different.

You should realize that the consequence of our imperfection is that the equations in electronics should always be taken with a grain of salt.

Nature does not care about your equations. Nature will easily have the superposition of trillions of electrons, and they don't have to agree with your equations.

But most of the time, the behavior is similar.

References

[1] T. C. Carusone, D. Johns, and K. Martin, *Analog integrated circuit design*. Wiley, 2011 [Online]. Available: https://books.google.no/books?id=1OIJZzLvVhcC

[2] Berkeley, "Berkeley short-channel IGFET model." [Online]. Available: http://bsim.berkeley.edu/models/bsim4/

MOSFETs 6

Status: 0.3

I'm stunned if you've never heared the word "transistor". I think most people have heard the word. What I find funny is that almost nobody understand in full detail how transistors work.

Through my 30 year venture into the world of electronics I've met "analog designers", or people that should understand exactly how transistors work. I used to hire analog designers, and I've interviewed hundred plus "analog designers" in my 8 years as manager and I've met hundreds of students of analog design. I would go as far as to say none of them know everything about transistors, including myself.

Most of the people I've met have a good brain, so that is not the reason they don't understand. Transistors are incredibly complicated! I say this, because if at some point in this document, **you** don't understand, then don't worry, you are not alone.

In this document I'm focusing on Metal Oxide Semiconductor Field Effect Transistors (MOSFETs), and ignore all other transistors.

6.1 Metal Oxide Semiconductor

The first part of the MOSFET name illustrates the 3 dimensional composition of the transistor. Take a semiconductor (Silicon), grow some oxide (Silicon Oxide, SiO2), and place a metal, or conductive, gate on top of the oxide. With those three components we can build our transistor.

Something like the cartoon below where only the Metal (gate) of the MOS name is shown.

The oxide and the silicon bulk is not visible, but you can imagine them to be underneath the gate, with a thin oxide (a few nano meters thick) and the silicon the transparent part of the picture.

The length (L), and width (W) of the MOS is annotated in blue.

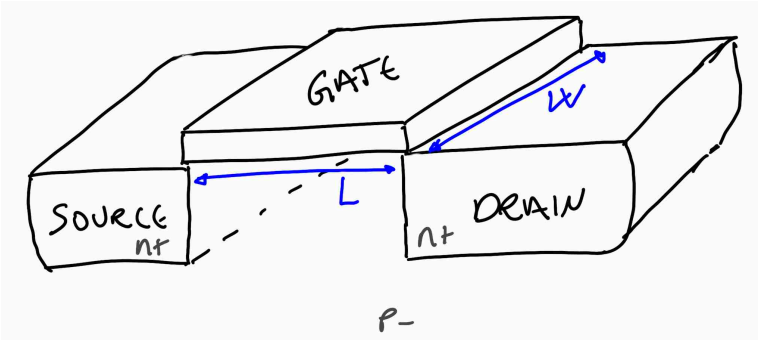


Figure 1: 3D crossection of a transistor

MOSFETs come in two main types. There is NMOS, and PMOS. The symbols are as shown below. The NMOS is MN1 and PMOS is MP1.

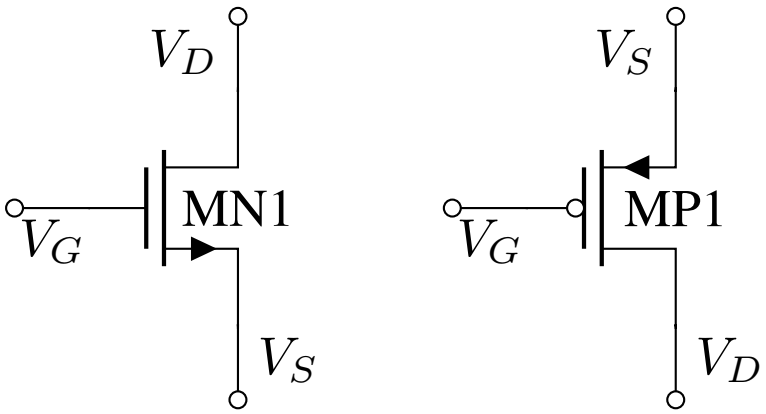


Figure 2: Transistor symbols

The MOS part of the name can be seen in MN1, where V_G is the gate connected to a vertical line (metal), a space (oxide), and another vertical line (the silicon substrate or silicon bulk).

On the sides of the gate we have two connections, a drain V_D and a source V_S.

If we have a sufficient voltage between gate and source V_{GS}, then the transistor will conduct from drain to source. If the voltage is too low, then there will not be much current.

The "source" name is because that's where the charge carrier (electrons) come from, they come from the source, and flow towards the drain. As you may remember, the "current", as we've defined it, flows opposite of the electron current, from drain to source.

The PMOS works in a similar manner, however, the PMOS is made of a different type of silicon, where the dominant charge carrier is holes in the valence band. As a result, the gate-source voltage needs to be negative for the PMOS to conduct.

In a PMOS the holes come from the source, and flow to the drain. Since holes are positive charge carriers, the current flows from source to drain.

In most MOSFETs there is no physical difference between source and drain. If you flip the transistor it would work almost exactly the same.

6.2 Field Effect

Imagine that the bulk (the empty space underneath the gate), and the source is connected to 0 V. Assume that the gate is 0 V.

In the source and drain parts of the transistor there is an abundance of **free** electrons that can move around, exactly like in a metal conductor, however, underneath the gate there are almost no **free** electrons.

There are electrons underneath the gate though, trillions upon trillions of electrons, but they are stuck in co-valent bonds between the Silicon atoms, and around the nucleus of the Silicon atoms. These electrons are what we call bound electrons, they cannot move, or more precisely, they cannot contribute to current (because they do move, all the time, but mostly around the atoms).

Imagine that your eyes could see the free electrons as a blue fluorescent color. What you would see is a bright blue drain, and bright blue source, but no color underneath the gate.

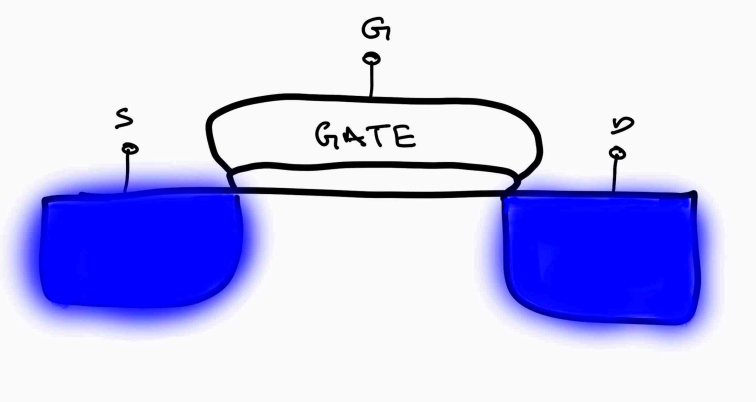


Figure 3: MOSFET in "off" state

As you increase the gate voltage, the color underneath the gate would change. First, you would think there might be some blue color, but it would be barely noticeable.

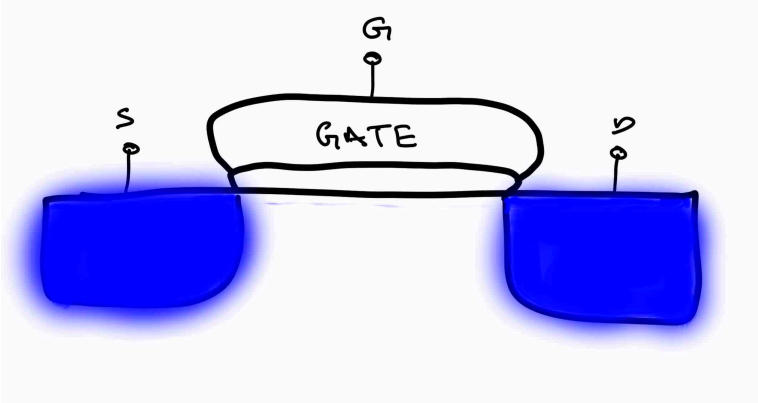


Figure 4: MOSFET in subthreshold

At a certain voltage, suddenly, there would be a thin blue sheet underneath the gate. You'd have to zoom in to see it, in reality it's a ultra thin, 2 dimensional electron sheet.

As you continue to increase the gate voltage the blue color would become a little brighter, but not much.

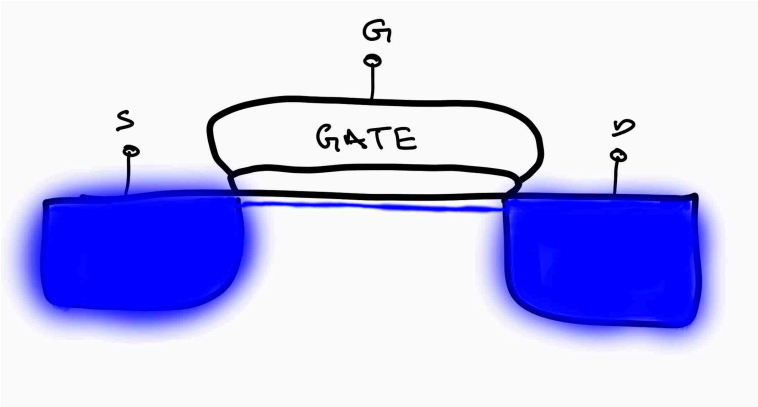


Figure 5: MOSFET in strong inversion

This thin blue sheet extend from source to drain, and create a conductive channel where the electrons can move from source to drain (or drain to source), exactly like a resistor. The conductance of the sheet is the same as the brightness, higher gate source voltage, more bright blue, higher conductance, less resistance.

Assume you raise the drain voltage. The electrons would move from source to drain proportional to the voltage. How many electrons could move would depend on the gate voltage.

If the gate voltage was low, then there is low density of electrons in the sheet, and low current.

If the gate voltage is high, then the electron density in the sheet is high, and there can be a high current, although, the electrons do have a maximum speed, so at some point the current does not change as fast with the gate voltage.

At a certain drain voltage you would see the blue color disappear close to the drain and there would be a gap in the sheet.

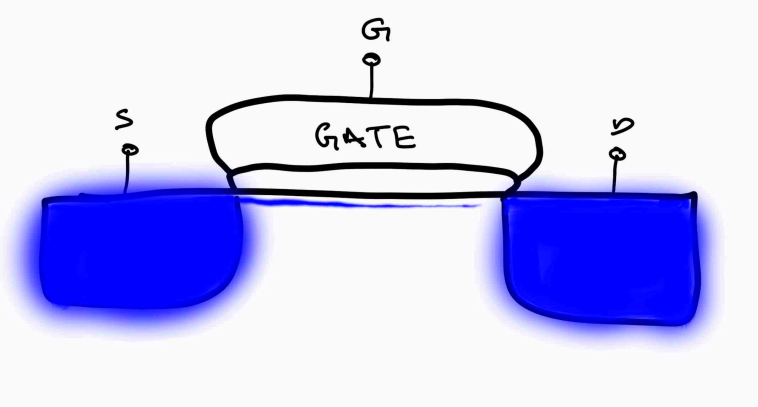


Figure 6: MOSFET in strong inversion and saturation

That could make you think the current would stop, but it turns out, that the electrons close to drain get swept across the gap because the electric field is so high from the edge of the sheet to the drain.

As you continue to increase the drain voltage, the gap increases, but the current does not really increase that much. It's this exact feature that make transistor so attractive in analog circuits. I can create a current from drain to source that does not depend much on the drain to source voltage! That's why we sometimes imagine transistors as a "trans-conductance". The conductance between drain and source depends on the voltage somewhere else, the gate-source voltage.

And now you may think you understand how the transistor works. By changing the gate voltage, we can change the electron current from source to drain. We can turn on, and off, currents, creating a 0 and 1 state.

For example, if I take a PMOS and connect the source to a high voltage, the drain to an output, and an NMOS with the source to ground and the drain to the output, and connect the gates together, I would have the simplest logic gate, an inverter, as shown below.

If the input V_{in} is a high voltage, then the output V_{out} is a low voltage, because the NMOS is on. If the input V_{in} is a low voltage, then the output V_{out} is a high voltage, because the PMOS is on.

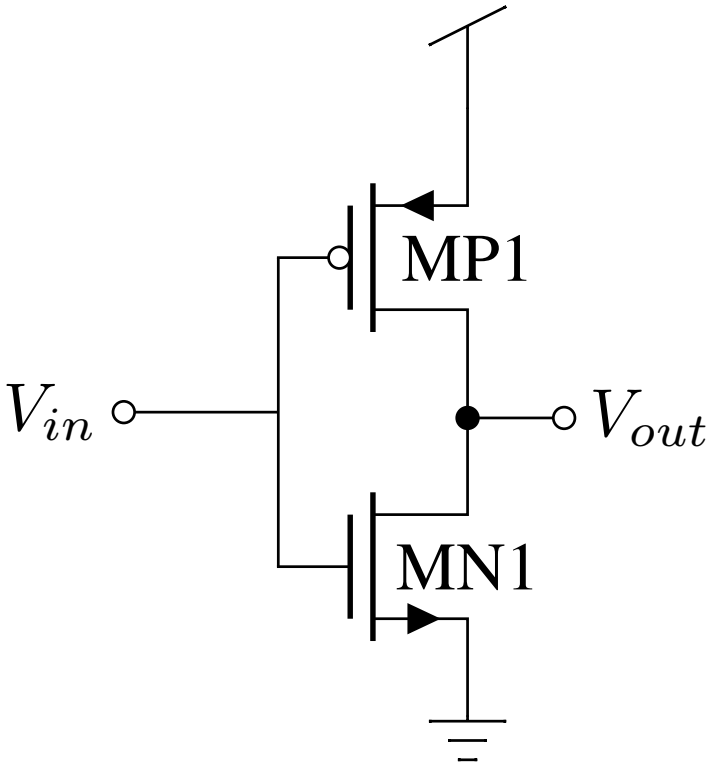


Figure 7: Inverter

I can now build more complex "logic gates". The one below is a Not-AND gate (NAND). If both inputs (A and B) are high, then the output is low (both NMOS are on). Otherwise, the output is high.

I find it amazing that all digital computers in existence can be constructed from the NAND gate. In principle, it's the only logic gate you need. If you actually did construct computers from NANDs only, they would be costly, and consume lots of power. There are smarter ways to use the transistors.

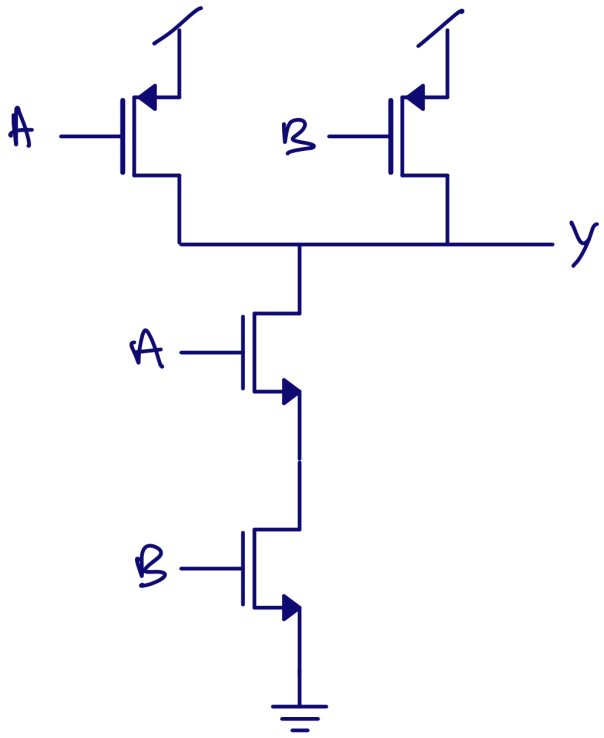


Figure 8: NAND

You may be too young to have seen the Matrix, but now is the time to decide between the red pill and the blue pill.

The red will start your journey to discover the reality behind the transistor, the blue pill will return you to your normal life, and you can continue to think that you now understand how transistors work.

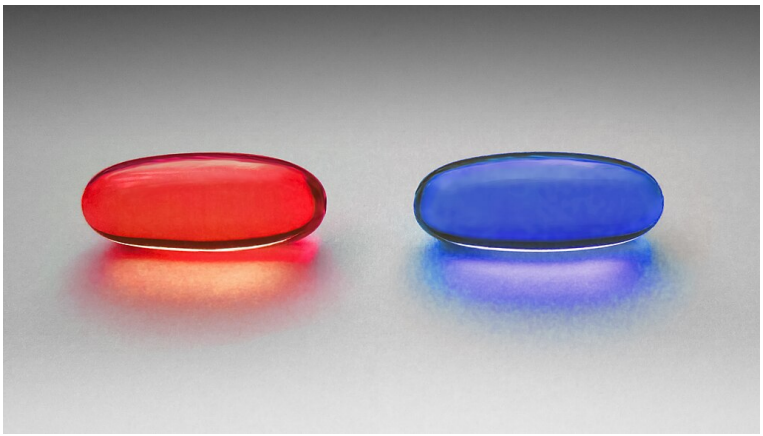


Figure 9: The choice

Because:

- Why did the area underneath the gate turn blue?
- Why is it only a thin sheet that turns blue?
- Where did the electrons for the sheet come from?
- Why did the blue color change suddenly?

- How does the brightness of the blue change with gate-source voltage?
- How can the electrons stay in that sheet when we connect the bulk to 0 V?
- Why is there not a current from the bulk (0 V) to drain?
- Why does not the electrons jump from source to drain? It's a gap, the same as from the sheet to drain?

And did you realize I never in this chapter explained how the field effect worked?

Someday, I may write all the details, if I ever understand it all. For now, I hope that the sections below will help you a bit.

6.3 Analog transistors in the books

In the books we learn the equations for weak inversion

$$I_D \propto (e^{(V_{gs}-V_{th})/U_T} - 1)$$

, where I_D is the drain current, V_{gs} is the gate source voltage, V_{th} is the threshold voltage and $U_T = kT/q$, where k is Boltzmann's constant, T is the temperature in Kelvin and q is the unit charge

The equation is similar to bipolar and diode equations, because the physics is the same.

The drain current in weak inversion is mostly a diffusion current and relates to the density of electrons in the conduction band (for an NMOS), which can be computed from the density of available energy states, and the Fermi-Dirac distribution.

$$n = \int_{E_C}^{\infty} N(E) \frac{1}{e^{(E-E_F)/kT} + 1} dE$$

, where n is the density of electrons in the conduction band, $N(E)$ is the density of available energy states, E is the integration variable (and the energy) and E_F is the Fermi-level.

Maybe the equation looks complicated, but it's really "Multiply the available energy state with the probability of being in that state, and sum for all available energy states".

Changing the voltage changes the number of free electrons, simply because we bring the conduction band closer to the Fermi level.

The Fermi level is just something we invented, and just means "If there was an quantum state at the Fermi level Energy, then it would have a 50 % probability of being occupied by a electron".

In the equation above, moving the conduction band edge is equivalent to reducing the E_C. As such, more of the Fermi-Dirac distribution has available energy states $N(E)$, and the density of electrons n in conduction band becomes higher.

In strong inversion, the MOSFET is more like a voltage controlled resistor with a conductance that is proportional to gate-source voltage.

The density of electrons increases because we bend the conduction band beyond the Fermi level, as a result, most of the available energy states in the conduction band are filled by electrons.

Electrons are only free to move, however, close to the surface of the silicon, as far away from the surface, we don't feel the effects of the gate-source voltage, and the conduction band stays at the same energy. As a result, electrons form a 2 dimensional electron gas close to the silicon surface. What we call an inversion layer.

Once we have that electron gas, or inversion layer, we have a connection between the drain and source n-type regions, and the current can be estimated by a drift current. Parts of the diffusion current will still be there, but much smaller magnitude than the drift current, so we drop the diffusion current, and get

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{gs} - V_{th})^2$$

The equations in the books are good to give a physical understanding of what happens. Although, we tend to forget that everybody forgets.

We teach quantum physics one year, and how to compute the density of states $N(E)$ from Schrodinger, the wave-function and Fermi-Dirac distribution.

Next year we talk about semiconductors, crystal lattice, band structure (density of states as a function of space), energy diagrams (band structure is complex, so we just use the lowest conduction band and highest valence band), doping to shift the Fermi level, and how we can create PN-junctions, bipolars and MOSFETS.

The year after we teach the current equations for MOSFETs, and the books don't have the link back to solid-state physics, after all, we already told the students that, they should remember!

I think, quite often, we just end up with confused students. And I don't think it's necessary to end up with confused students. Maybe sometimes we end up with confused students because the Professors can't necessarily remember where the equations come from either, nor how electrons and holes really behave.

It's not necessary for an analog design student to remember how to compute the density of available energy states from Schrodinger and the wave function. If we wanted to use the relativistc version

of Schrodinger (which includes magnetic fields, and if you did not know, magnetic fields is just a relativistic effect of the electric field) and the wave function to compute how an Silicon atom actually behaves, I don't think we can. As far as I've been able to figure out, it's not possible to have a closed form solution (symbolic), nor is it possible with supercomputers to do a numeric time-evolution of the states in a single Silicon atom with all the inter-particle interactions, space, momentum, spins, electric fields and magnetic fields.

But we can make sure we connect the links from Schrodinger to the MOSFET equations, the short version of that was above, but the following sections tries to explain with words how the transistor actually works.

I'm not going to give all the equations and all the maths. For that, there are excelent books and resources. I would recommend Mark Lundstrom for the best in detail description of MOSFETs.

6.4 Transistors in weak inversion

Consider the cartoon below which shows the hole concentration in the valence band, and electron concentration in the conduction band versus the x direction of the transistor.

For the moment we'll ignore the field effect of the gate, and how that modulates the hole concentration underneath the gate.

If you're familiar with bipolars, then you may think I've drawn the wrong transistor, because you see an NPN bipolar transistor. The picture is correct, however, this is how a normal MOSFET looks. It's actually also a NPN bipolar transistor, but we don't usually use that part (you'll see more when we get to ESD)

In the source we've doped with donors, and have an abundance of free electrons. Underneath the gate, or the bulk, we have doped with acceptors, and have an abundance of holes.

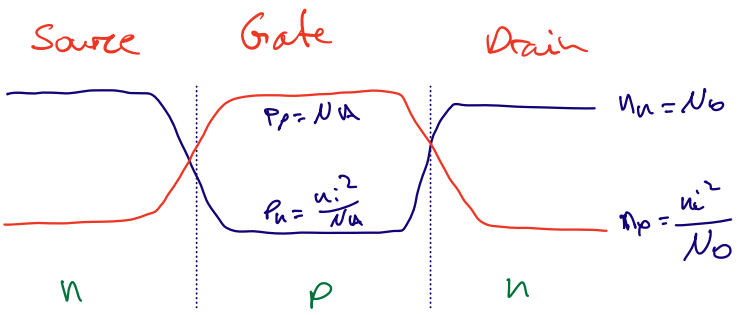


Figure 10: Charge carrier density in a MOSFET

Let's consider electron current for now, and only look at the conduction band.

An electron in the source would see a energy barrier of ϕ_B, and most electrons would be turned around at the barrier. Some, however, do have the energy to traverse the barrier and flow through the bulk. Not all of them would reach the bulk, due to recombination, but let's assume the bulk is short, and all electrons injected into the bulk show up at the drain.

At the drain side they would fall down the potential barrier to the drain. The same process would happen in reverse, from drain to source.

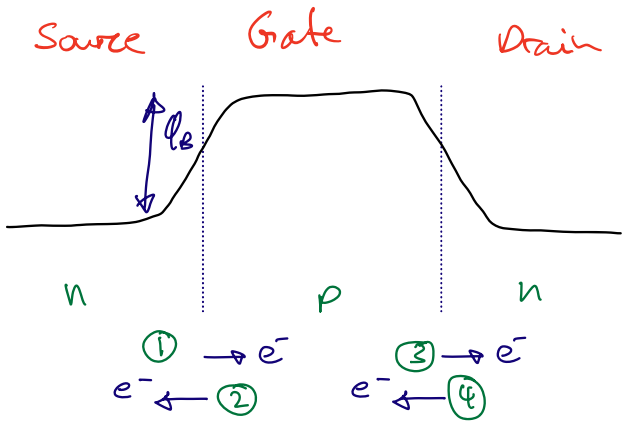


Figure 11: MOSFET subthreshold , $V_{DS} = 0$

There would also be hole currents flowing between source/bulk/drain and visa versa

Assume source and drain are at the same potential, then the sum of all currents (1,2,3,4) for both electrons and holes in Figure 11 must equal zero.

Assume that we increase the drain voltage, as shown in Figure 12. Increasing the drain voltage is the same as reducing the conduction band in the drain.

Since there now is a higher barrier from drain to bulk, it's now much less probable that electrons are injected from drain to bulk.

Now the sum of all currents would not equal zero, as the 1 and 3 currents are larger than 2 and 4.

As such, there would be a net flow of electron current from source to drain.

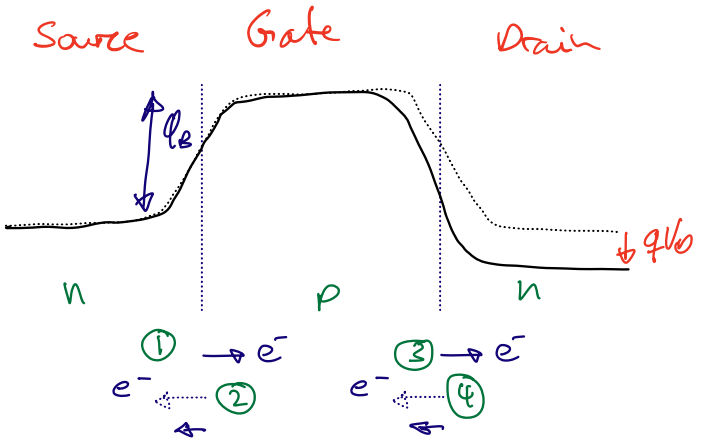


Figure 12: MOSFET subthreshold, $V_S = 0\ V, V_D > 0\ V$

Notice that if we increase the drain voltage further, then the electron injection from drain to bulk would quickly approach zero.

At that point, even though we increase the drain voltage further, the current does not really change. As the current is only now given by the barrier height at the source.

The barrier height at the source is the built in voltage of the junction, and as we've seen before, that voltage depends on doping concentration. If we increase the hole concentration in bulk, then we increase the barrier height, and it's less probable that the electrons have enough energy to be injected from source to bulk.

If we only need to consider the electrons and holes at source for the subthreshold current (assuming the drain voltage is high enough), then we should expect the equation look very similar to a diode, and indeed it does.

The drain current, which is mostly a diffusion current, is given by

$$I_D = I_{D0}\frac{W}{L}e^{q(V_{GS}-V_{TH})/nkT}$$

where

$$n = (C_{ox} + C_{j0})/C_{ox}$$

$$I_{D0} = (n-1)\mu_n C_{ox}\left(\frac{kT}{q}\right)^2$$

This is not exactly the same as the diode equation, but we can see that it looks similar. Most of the quantum mechanics is baked into the V_{TH}

The transconductance (dI_D/dV_{GS}) in weak inversion is then

$$g_m = \frac{I_D}{nV_T}$$

A big difference from the diode equation is the fact that the gate-source voltage seems to determine the current, and not the voltage across the pn junction.

6.5 Transistors in strong inversion

Consider the band diagram in Figure 13, in the figure we're looking at a cross section of the transistor. From left we're in the gate, then we have the oxide, and then the bulk of the transistor.

We don't see the drain and source, as the source would be towards you, and the drain would be into the picture.

The cartoon is not a real transistor. I don't think there is necessarily a combination of semiconductor and metal where we end up with the same Fermi level (E_F) without some bending of the conduction band and valence band, but for illustration, let's assume that's the case.

We can see the Fermi level in the semiconductor is shifted towards the valence band, and thus we have a P-type semiconductor.

The gate is metallic, so it does not have a bandgap, and we assume that the Fermi level is at the conduction band edge.

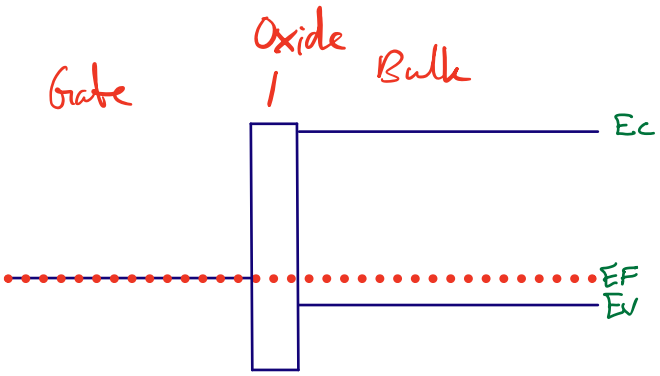


Figure 13: Band diagram of a fictive MOSFET.

Assume we increase the gate-source voltage. In a band diagram that corresponds to shifting the energy down.

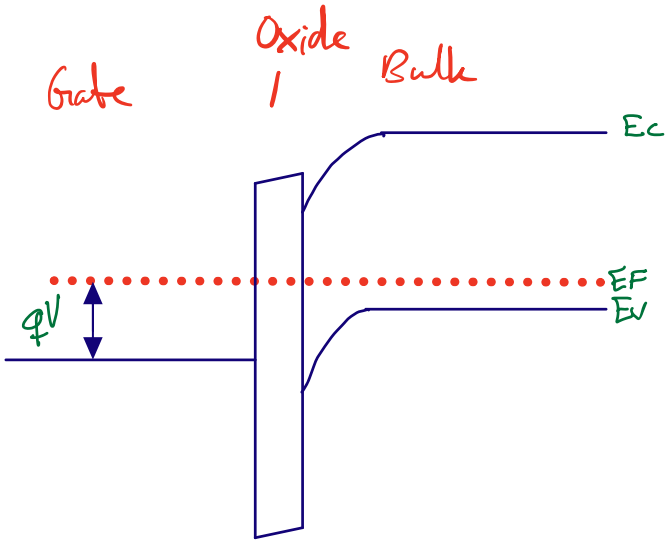


Figure 14: Band diagram with gate-source voltage applied

Moving the gate down has the effect of bending the bands in the semiconductor. We'll lose some voltage across the oxide, but not necessarily that much.

The bending of the valence band will decrease the hole concentration close to the silicon surface, and the semiconductor will be depleted of mobile charge carriers.

The valence band bending will also reduce the barrier height in Figure 12, which increases the number of carriers that can be injected at source/bulk interface, so the subthreshold current will start to increase.

At some point, the band bending of the conduction band will become so large that the electron concentration underneath the gate will increase signficantly. The gate-source voltage where the electron concentration equals the bulk hole concentration far away from the silicon surface is called the "threshold voltage".

As you continue to increase the gate-source voltage there is a limit to how much the electron concentration increases. When the band bending of the conduction band passes the Fermi level, then over 50 percent of the available states in the conduction band are filled with electrons.

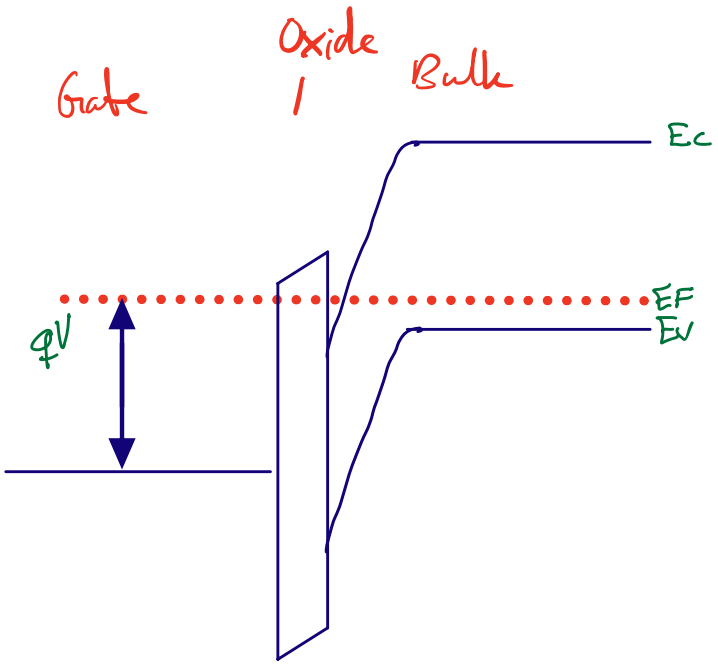


Figure 13: Band diagram with high gate-source voltage applied

The conditions to be in strong inversion is that the gate/source voltage is above some magic values (threshold voltage), and then some.

The quantum state of the electron is fully determined by it's spin, momentum and position in space. How those parameters evolve with time is determined by the Schrodinger equation. In the general form

$$i\hbar\frac{d}{dt}\Psi(r,t) = \widehat{H}\Psi(r,t)$$

The Hamiltonian (H) is an "energy matrix" operator and may contain terms both for the momentum and Columb force (electric field) experienced by the system.

But what does the Schrodinger equation tell us? Well, the equation above does not tell me much, it can't be "solved", or rather, it does not have a single solution. It's more a framework for how the wave function, and the Hamiltonian, describes the quantum states of a system, and the probability ampltiudes of transition between states.

The Schrodinger equation describes the time evolution of the bound electrons shared between the Silicon atoms, and the fact that applying a electric field to silicon can free co-valent bonds.

As the gate-source voltage increases the wave function that fits in the Schrodinger equation predicts that the free electrons will form a 2d sheet underneath the gate. The thickness of the sheet is only a few nano meters.

In Figure 2 in

Carrier transport near the Si/SiO2 interface of a MOSFET

you can see how the free electron density is located underneath the gate.

I would really recommend that you have a look at Mark Lundstrom's lecture series on Essentials of MOSFETs. It's the most complete description of electrons in MOSFET's I've seen

6.6 How should I size my transistor?

The method that makes most sense to me, is to use the inversion-coefficient method, described in Nanoscale MOSFET Modeling: Part 1 and Nanoscale MOSFET Modeling: Part 2.

The inversion coefficient tells us how strongly inverted the MOSFET channel (inversion layer) is. A number below 0.1 is weak inversion, between 0.1 and 10 is moderate inversion. A number above 10 is strong inversion.

There are also some blog posts worth looking at Inversion Coefficient Based Circuit Design and My Circuit Design Methodology.

I should caveat my proposal for method. For the past 7 years I've not had the luxury to do full time, hardcore, analog design. As my career progressed, most of my time is now spent telling others what I think is a good idea to do, and not doing hardcore analog design myself. I think, however, I have a pretty decent understanding of analog circuits, and how to design them, so I think I'm correct in the proposal. If I were to start hardcore analog design now, I would go all in on inversion-coefficient based transistor size selection.

6.7 Introduction to behavior

Let's assume we know nothing about how transistors work, but we do know how to simulate them in ngspice.

We could sit down, and try and figure out how the transistors work.

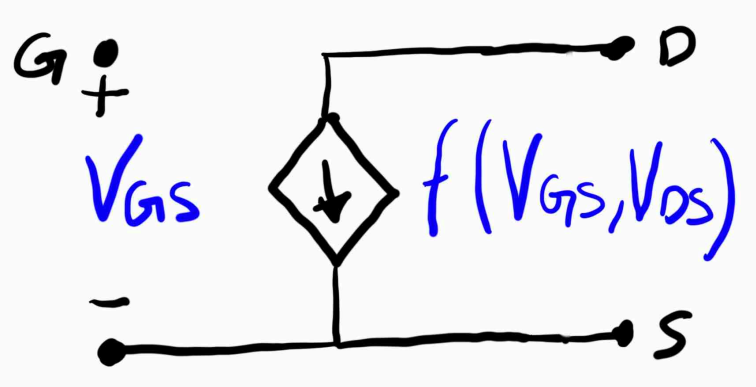
You can find the testbenches at Testbenches at dicex/sim/spice/N-CHIO

6.7.1 Drain Source Current

Let's see what happens to the drain to source current when we change the voltages. We would expect the drain to source current to change as a function of the drain to source, V_{DS}, and gate to source V_{GS} voltages. Or mathematically

$$I_{DS} = f(V_{GS}, V_{DS}, ...)$$

or symbolically



The symbolic model above is what we call a "Large Signal Model". We could expand the function above to

$$I_{DS} = f(V_{GS}, V_{DS}) = G_m(V_{GS}, V_{DS}, I_{DS})V_{GS} + G_{ds}(V_{GS}, V_{DS}, I_{DS})V_{DS}$$

, where the G_m is a trans-conductance (the current depends on a voltage somewhere else), and G_{ds} is a conductance (current depends on the voltage across the conductance).

Even now we can see that the model above is complicated. The transconductance and conductance of the transistor is a function of the other voltages, and the output current. It's a non-linear system!

If the transistor was linear, then we would expect that the current increased proportionally to gate/source voltage, but how does the current look when we change the gate source voltage?

6.7.2 Gate-source voltage

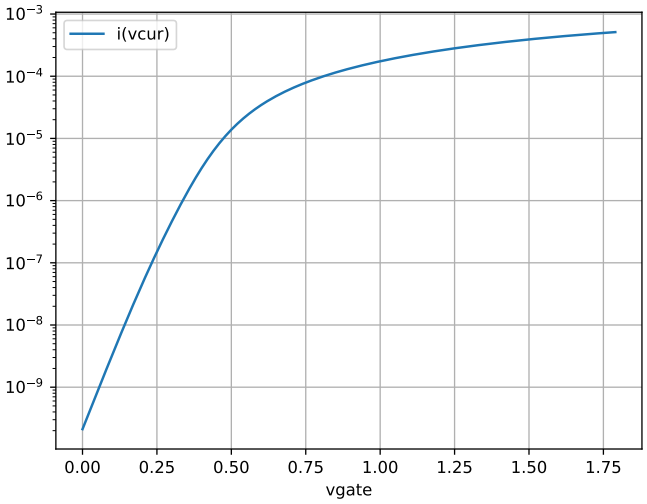
Below are the conditions I've used in the testbench. Notice there is a V_B that is the $p-$ substrate, or bulk, of the transistor. When we draw symbols of a transistor we don't always include the bulk node, because that's most of the time connected to ground for NMOS.

But sometimes, we connect the bulk to another voltage, so the bulk terminal will be in our schematics.

Param	Voltage
VGS	0 to 1.8
VDS	1.0
VS	0
VB	0

In the plot below we can see the sweep of the gate voltage.

$$i(vcur) = I_{DS}$$



6.7.3 Inversion level

Define

$$V_{eff} \equiv V_{GS} - V_{tn}$$

, where

$$V_{tn}$$

is the "threshold voltage"

Veff	Inversion level
less than 0	weak inversion or subthreshold
0	moderate inversion
more than 100 mV	strong inversion

Weak inversion

The drain current is low, but not zero, when

$$V_{eff} << 0$$

$$I_{DS} \approx I_{D0} \frac{W}{L} e^{V_{eff}/nV_T} \text{ if } V_{DS} > 3V_T$$

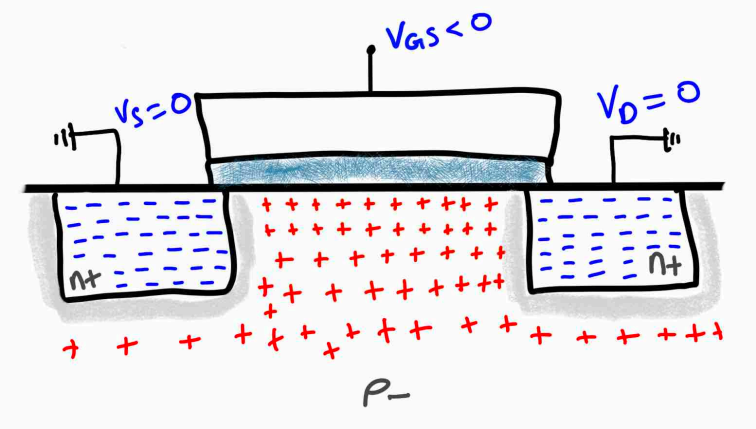
$$n \approx 1.5$$

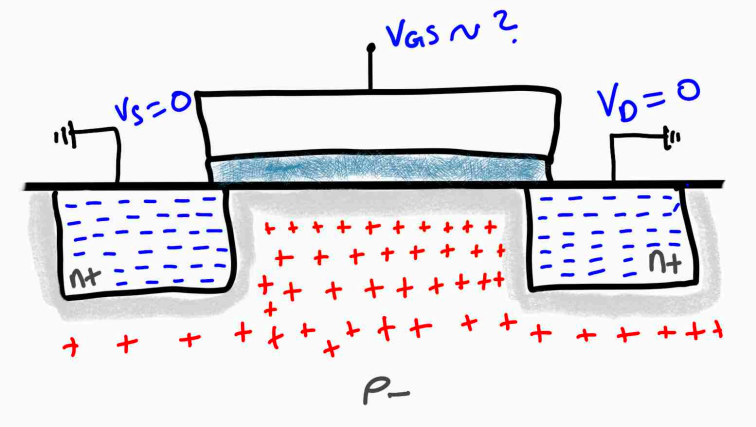
Moderate inversion

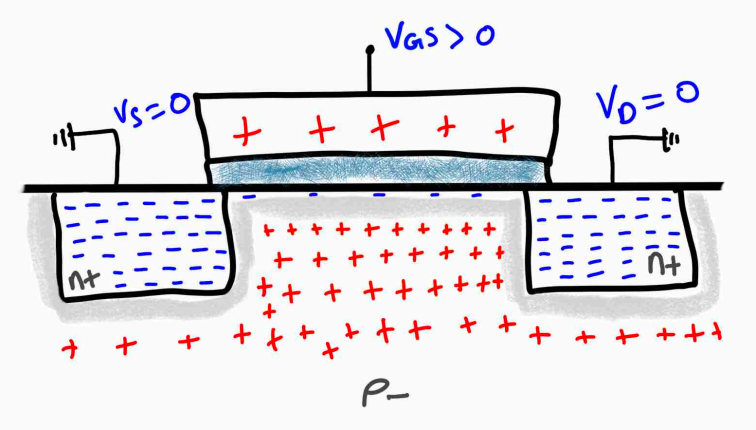
Very useful region in real designs. Hard for hand-calculation. Trust the model.

Strong inversion

$$I_{DS} = \mu_n C_{ox} \frac{W}{L} \begin{cases} V_{eff} V_{DS} & \text{if } V_{DS} << V_{eff} \\ V_{eff} V_{DS} - V_{DS}^2/2 & \text{if } V_{DS} < V_{eff} \\ \frac{1}{2} V_{eff}^2 & \text{if } V_{DS} > V_{eff} \end{cases}$$



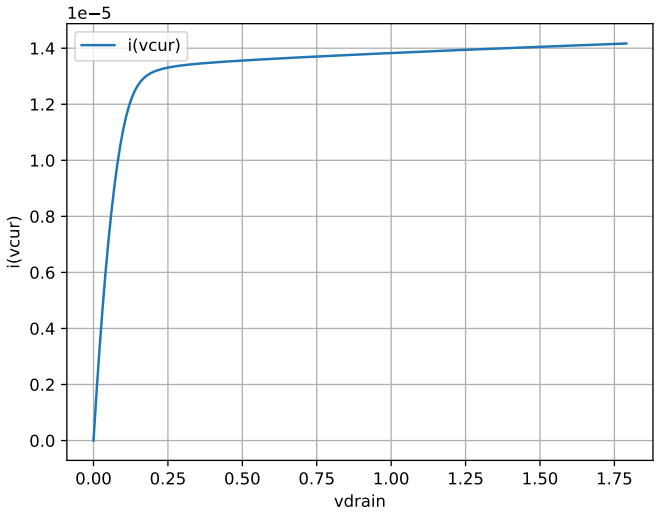




6.7.4 Drain source voltage

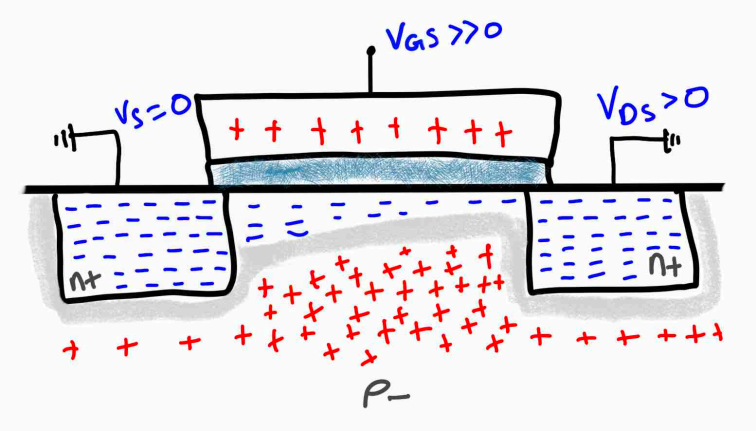
Param	Voltage [V]
VGS	0.5
VDS	0 to 1.8
VS	0
VB	0

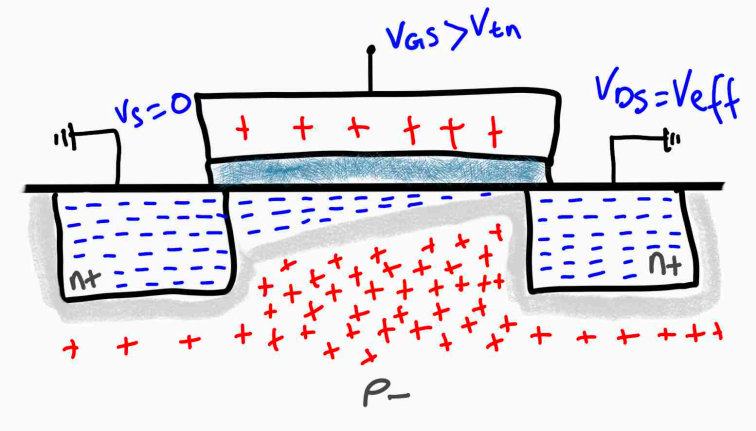
$$i(vcur) = I_{DS}$$

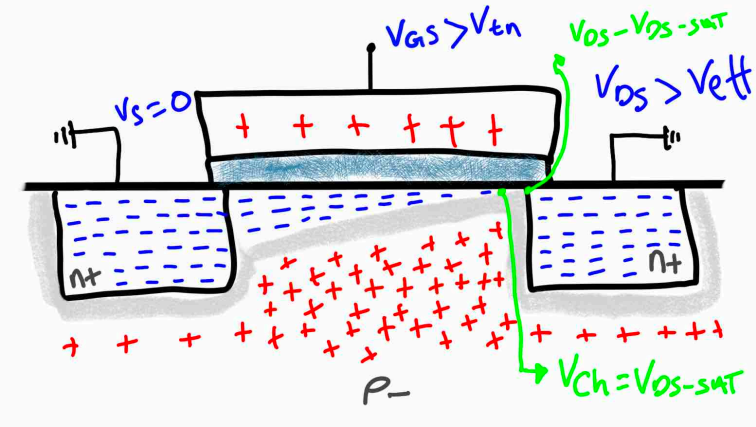


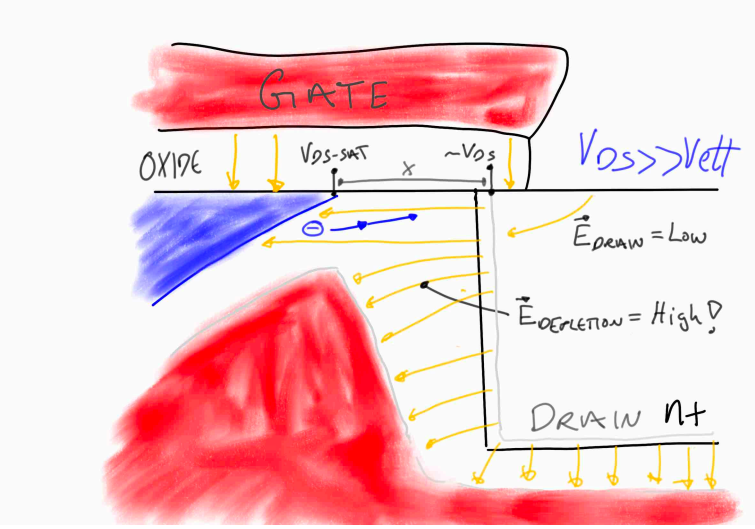
6.7.5 Strong inversion

$$I_{DS} = \mu_n C_{ox} \frac{W}{L} \begin{cases} V_{eff} V_{DS} & \text{if } V_{DS} << V_{eff} \\ V_{eff} V_{DS} - V_{DS}^2/2 & \text{if } V_{DS} < V_{eff} \\ \frac{1}{2} V_{eff}^2 & \text{if } V_{DS} > V_{eff} \end{cases}$$





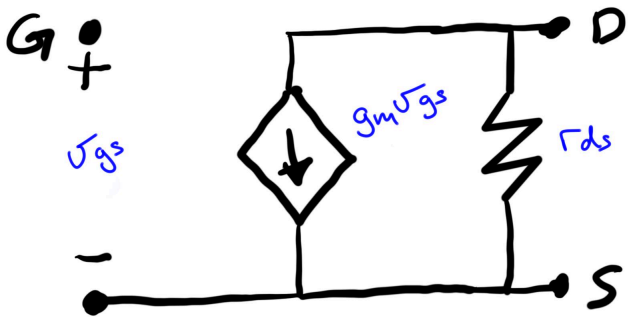




6.7.6 Low frequency model

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}}$$

$$g_{ds} = \frac{1}{r_{ds}} = \frac{\partial I_{DS}}{\partial V_{DS}}$$



6.7.7 Transconductance

Define

$$\ell = \mu_n C_{ox} \frac{W}{L}$$

and

$$V_{eff} = V_{GS} - V_{tn}$$

$$I_D = \frac{1}{2}\ell(V_{eff})^2$$

and

$$V_{eff} = \sqrt{\frac{2I_D}{\ell}}$$

and

$$\ell = \frac{2I_D}{V_{eff}^2}$$

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} = \ell V_{eff} = \sqrt{2\ell I_D}$$

$$g_m = \ell V_{eff} = 2\frac{I_D}{V_{eff}^2}V_{eff} = \frac{2I_D}{V_{eff}}$$

Define

$$\ell = \mu_n C_{ox}\frac{W}{L}$$

and

$$V_{eff} = V_{GS} - V_{tn}$$

$$I_D = \frac{1}{2}\ell V_{eff}^2[1 + \lambda V_{DS} - \lambda V_{eff})]$$

$$\frac{1}{r_{ds}} = g_{ds} = \frac{\partial I_D}{\partial V_{DS}} = \lambda\frac{1}{2}\ell V_{eff}^2$$

Assume channel length modulation is not there, then

$$I_D = \frac{1}{2}\ell V_{eff}^2$$

which means

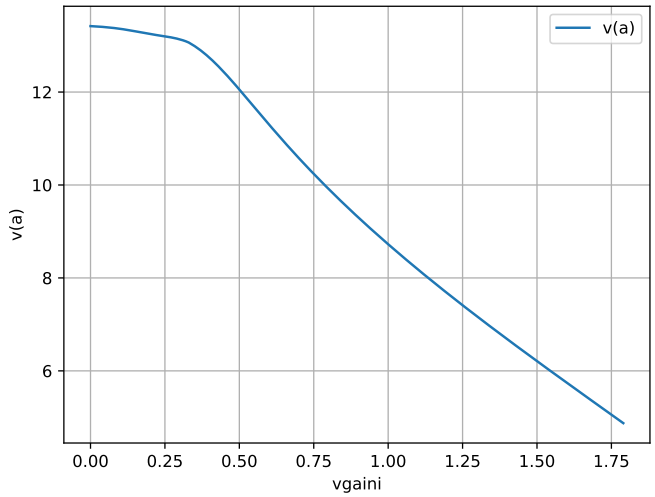
$$\frac{1}{r_{ds}} = g_{ds} \approx \lambda I_D$$

6.7.8 Intrinsic gain

Define intrinsic gain as

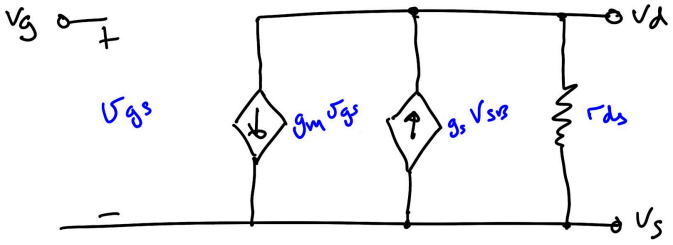
$$A = \left|\frac{v_{out}}{v_{in}}\right| = g_m r_{ds} = \frac{g_m}{g_{ds}}$$

$$A = \frac{2I_D}{V_{eff}} \times \frac{1}{\lambda I_D} = \frac{2}{\lambda V_{eff}}$$

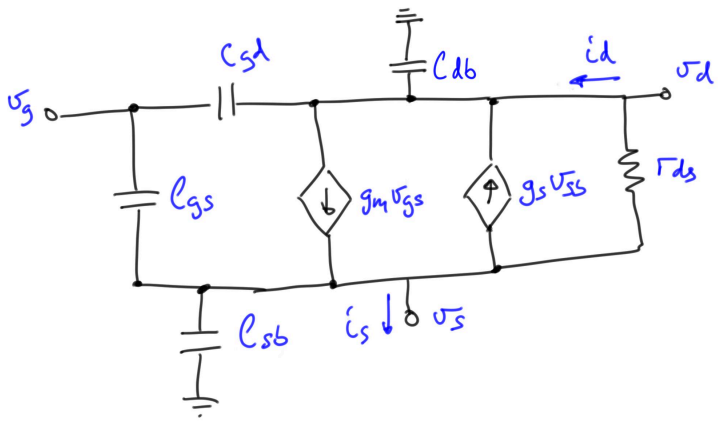


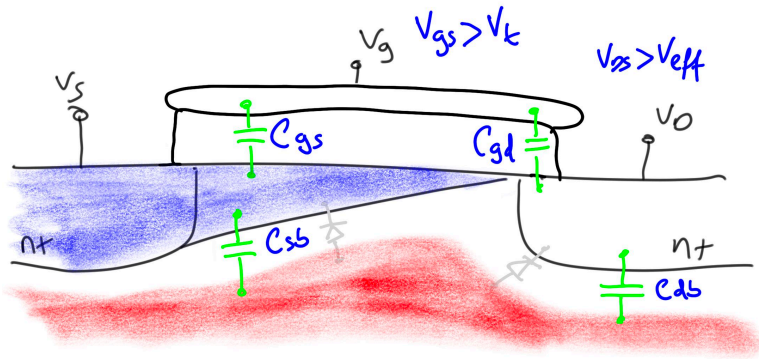
vgaini = Gate Source Voltage =

$$V_{eff} + V_{tn}$$



6.7.9 High frequency model





$$C_{gs}$$

and

$$C_{gd}$$

$$C_{gs} = \begin{cases} WLC_{ox} & \text{if } V_{DS} = 0 \\ \frac{2}{3}WLC_{ox} & \text{if } V_{DS} > V_{eff} \end{cases}$$

$$C_{gd} = C_{ox}WL_{ov}$$

$$C_{sb}$$

and

$$C_{db}$$

Both are depletion capacitances

$$C_{sb} = (A_s + A_{ch})C_{js}$$

$$C_{js} = \frac{C_{j0}}{\sqrt{1 + \frac{V_{SB}}{\Phi_0}}}$$

$$\Phi_0 = V_T ln\left(\frac{N_A N_D}{n_i^2}\right)$$

$$C_{db} = A_d C_{jd}$$

$$C_{js} = \frac{C_{j0}}{\sqrt{1 + \frac{V_{DB}}{\Phi_0}}}$$

6.7.10 Be careful with Cgd (blame Miller)

If

$$Y(s) = 1/sC$$

then

$$Y_1(s) = 1/sC_{in}$$

and

$$Y_2(s) = 1/sC_{out}$$

where

$$C_{in} = (1 + A)C$$

,

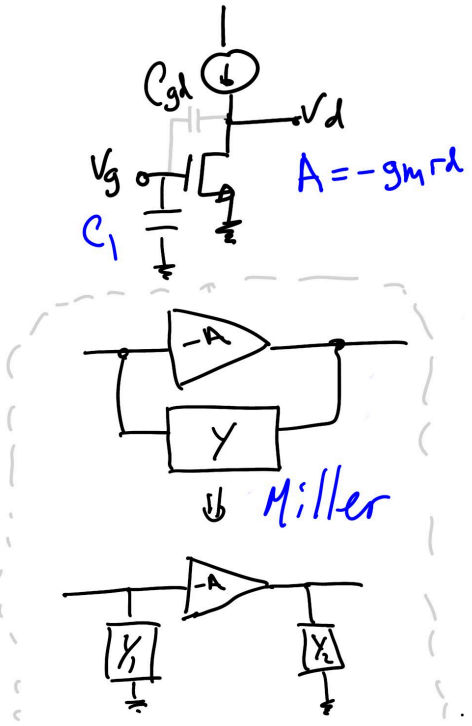
$$C_{out} = (1 + \frac{1}{2})C$$

$$C_1 = C_{gd} g_m r_{ds}$$

C_{gd}

can appear to be 10 to 100 times larger!

if gain from input to output is large



6.8 Weak inversion

If

$$V_{eff} < 0$$

diffusion currents dominate.

$$I_D = I_{D0} \frac{W}{L} e^{V_eff/nV_T}$$

, where

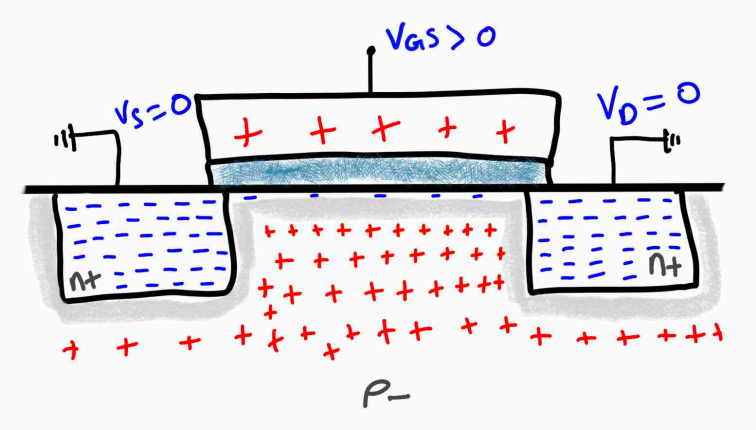
$$V_T = kT/q$$

,

$$n = (C_{ox} + C_{j0})/C_{ox}$$

$$I_{D0} = (n-1)\mu_n C_{ox} V_T^2$$

$$g_m = \frac{I_D}{nV_T}$$



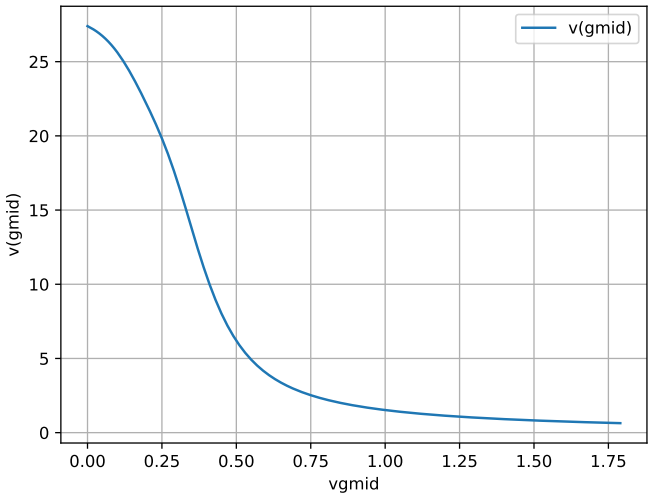
Bang for the buck

Subthreshold:

$$\frac{g_m}{I_D} = \frac{1}{nV_T} \approx 25.6 \text{ [S/A] @ 300 K}$$

Strong inversion:

$$\frac{g_m}{I_D} = \frac{2}{V_{eff}}$$



6.9 Velocity saturation

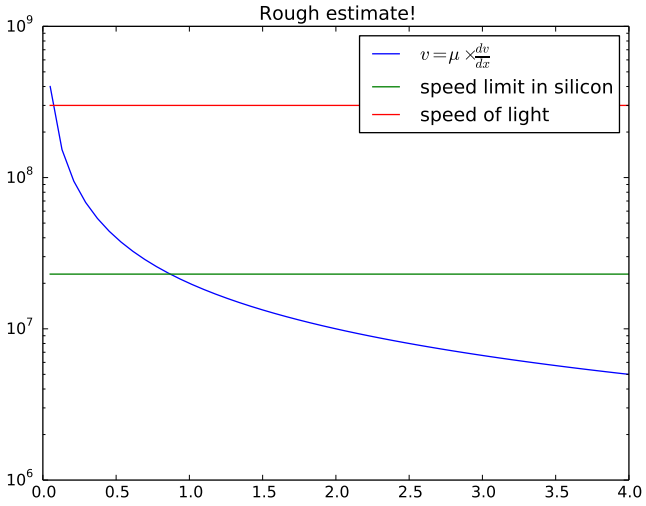
Electron speed limit in silicon

$$v \approx 10^7 cm/s$$

$$v = \mu_n E = \mu_n \frac{dV}{dx}$$

$$\mu_n \approx 100 \text{ to } 600\ cm^2/Vs$$

in nanoscale CMOS



6.9.1 Square law model

$$Q(x) = C_{ox}\left[V_{eff} - V(x)\right]$$

$$v = \mu_n E = \mu_n \frac{dV}{dx}$$

$$\ell = \mu_n C_{ox} \frac{W}{L}$$

$$I_D = WQ(x)v = \ell L\left[V_{eff} - V(x)\right]\frac{dV}{dx}$$

$$I_D dx = \ell L\left[V_{eff} - V(x)\right]dV$$

$$I_D \int_0^L dx = \ell L \int_0^{V_{DS}} \left[V_{eff} - V(x)\right]dV$$

$$I_D\,[x]_0^L = \ell L\left[V_{eff}V - \frac{1}{2}V^2\right]_0^{V_{DS}}$$

$$I_D L = \ell L\left[V_{eff}V_{DS} - \frac{1}{2}V_{DS}^2\right]$$

$$@V_{DS} = V_{eff} \Rightarrow I_D = \frac{1}{2}\ell V_{eff}^2$$

6.9.2 Mobility Degradation

Multiple effects degrade mobility

- Velocity saturation
- Vertical fields reduce channel depth => more charge-carrier scattering

$$\ell = \mu_n C_{ox} \frac{W}{L}$$

$$\mu_{n\_eff} = \frac{\mu_n}{([1 + (\theta V_{eff})^m])^{1/m}}$$

$$I_D = \frac{1}{2}\ell V_{eff}^2 \frac{1}{([1 + (\theta V_{eff})^m])^{1/m}}$$

From square law

$$g_m = \frac{\partial I_D}{\partial V_{GS}} = \ell V_{eff}$$

With mobility degradation

$$g_{m(mob-deg)} = \frac{\ell}{2\theta}$$

6.9.3 What about holes (PMOS)

In PMOS holes are the charge-carrier (electron movement in valence band)

$$\mu_p < \mu_n$$

In intrinsic silicon:

$$\mu_n \leq 1400[cm^2/Vs] = 0.14[m^2/Vs]$$
$$\mu_p \leq 450[cm^2/Vs] = 0.045[m^2/Vs]$$

$$\mu_n \approx 3\mu_p$$

$$v_{n\_max} \approx 2.3 \times 10^5[m/s]$$
$$v_{p\_max} \approx 1.6 \times 10^5[m/s]$$

Doping (

$$N_A \textbf{or} N_D$$

) reduces

$$\mu$$

6.10 OTHER

As we make transistors smaller, we find new effects that matter, and that must be modeled.

_ which is an opportunity for engineers to come up with cool names _

CONTRIBUTED PAPER

Analog Circuit Design in Nanoscale CMOS Technologies

Classic analog designs are being replaced by digital methods, using nanoscale digital devices, for calibrating circuits, overcoming device mismatches, and reducing bias and temperature dependence.

By Lanny L. Lewyn, *Life Senior Member IEEE*, Trond Ytterdal, *Senior Member IEEE*, Carsten Wulff, *Member IEEE*, and Kenneth Martin, *Fellow IEEE*

https://ieeexplore.ieee.org/document/5247174

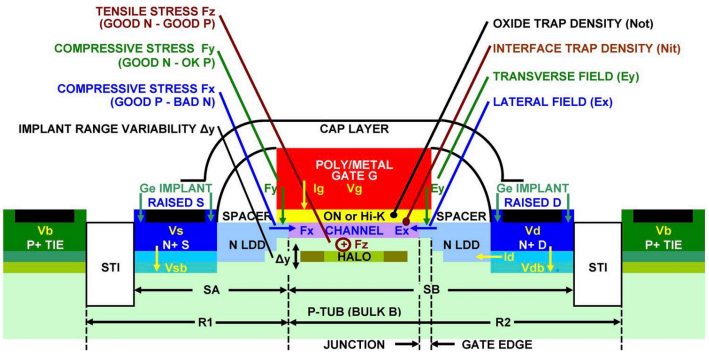
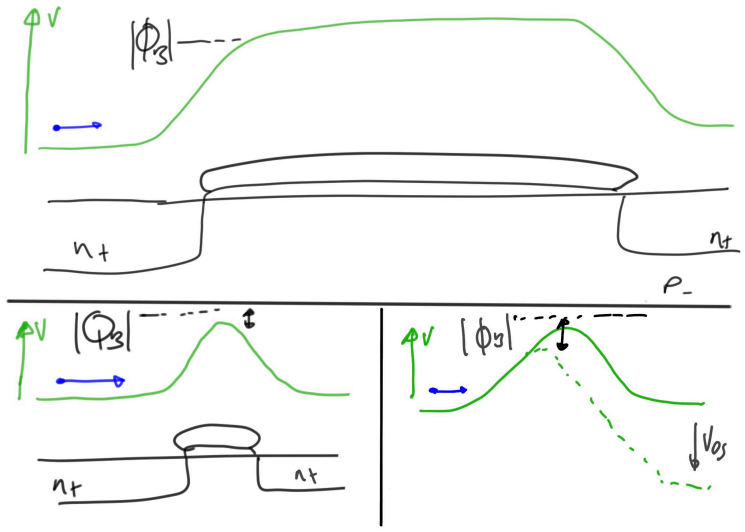
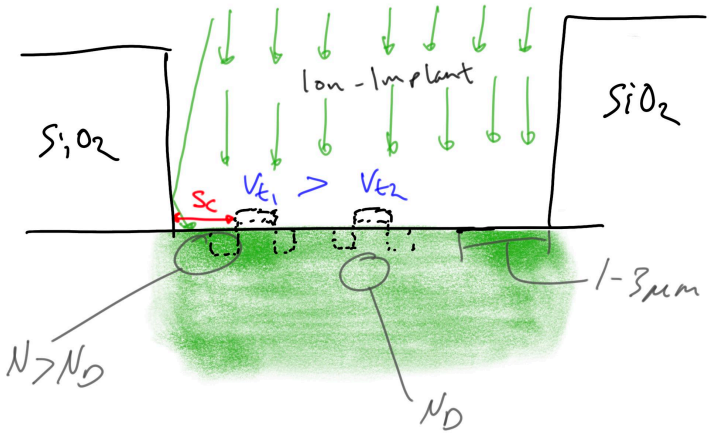


Fig. 2. *NMOS cross-section. In addition to stress from cap layers and Ge raised source-drain (S-D) implants, device dimensions such as distance from source-channel boundary to nearby STI (SA and SB), proximity and regularity of overlying metal patterns, and short distances to other device patterns within the local (< 2 μm) stress field induce transverse (F_y) and lateral (F_x and F_z) stress components, which affect threshold and mobility. Increasing the distance to P+ ties increases local tub (bulk) resistance components R1 and R2, which isolate the device MOS model substrate node from the device subcircuit symbol V_b node and degrade HF performance. Hot carrier reliability stress is dependent on the sum of transverse and lateral fields E_y and E_x. These fields are increased near the drain by increasing source to bulk (V_{sb}) and drain (V_d) to gate (V_g) or source (V_s) voltages in various combinations. As hot carrier stress increases, damage to channel from interface trap density (N_{it}) affects threshold and mobility, while gate oxynitirde (ON) or high-dielectric-constant (Hi-K) insulator trap density (N_{ot}) affects threshold and gate leakage.*

6.10.1 Drain induced barrier lowering (DIBL)



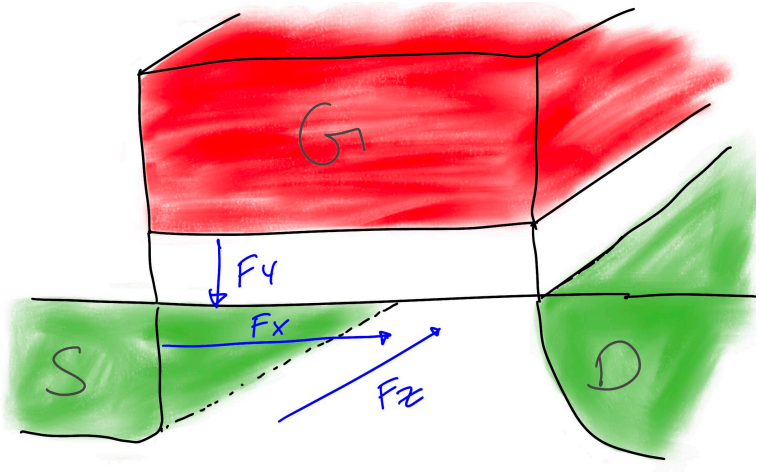
6.10.2 Well Proximity Effect (WPE)



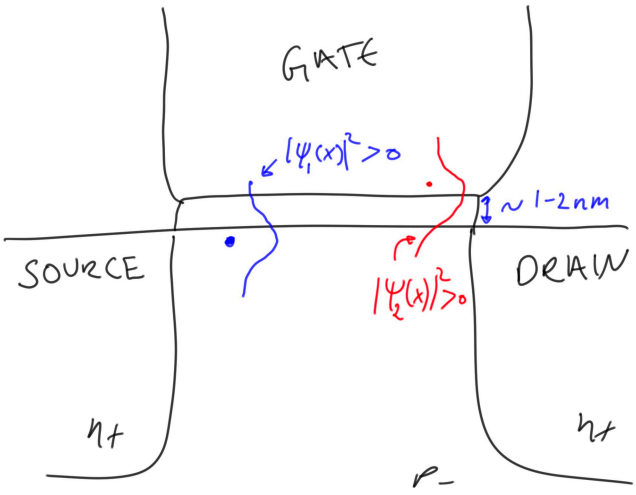
6.10.3 Stress effects

Stress	PMOS	NMOS
Stretch Fz	Good	Good
Compress Fy	OK	Good
Compress Fx	Good	Bad

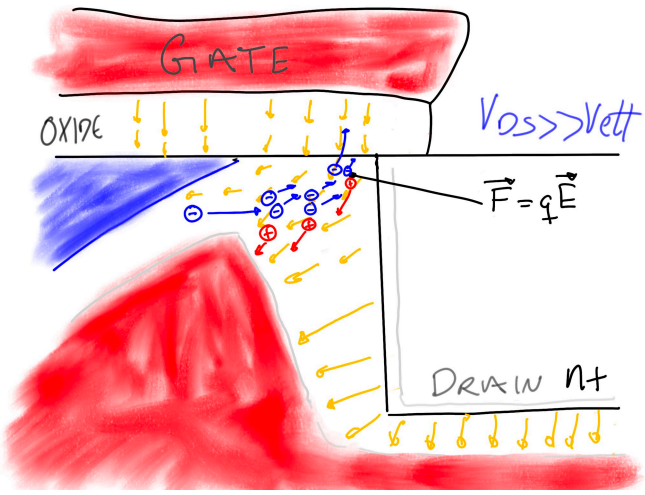
What can change stress?



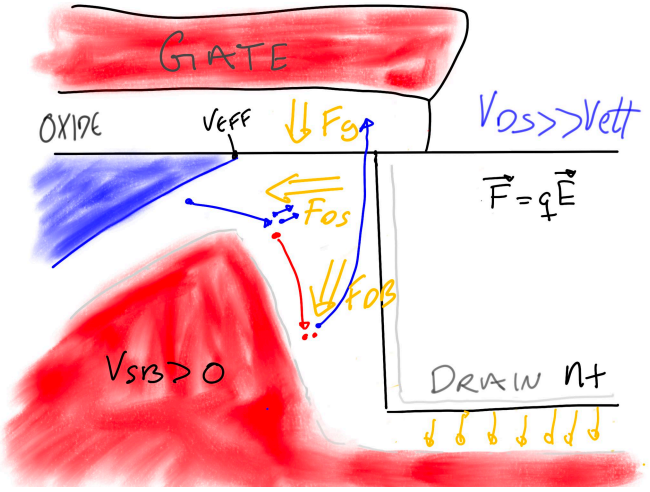
6.10.4 Gate current



6.10.5 Hot carrier injection



6.10.6 Channel initiated secondary-electron (CHISEL)



6.11 Variability

Provide

$$I_2 = 1\mu A$$

Let's use off-chip resistor

$$R$$

, and pick

$$R$$

such that

$$I_1 = 1\mu A$$

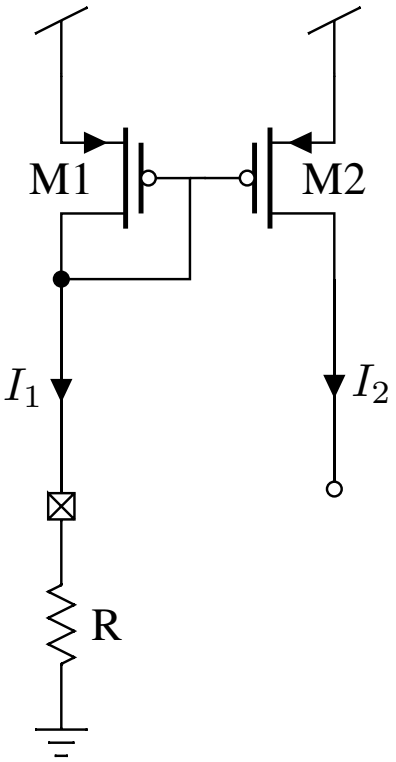
Use

$$\frac{W_1}{L_1} = \frac{W_2}{L_2}$$

What makes

$$I_2 \neq 1\mu A$$

?



- Voltage variation
- Systematic variations
- Process variations
- Temperature variation
- Random variations
- Noise

6.11.1 Voltage variation

$$I_1 = \frac{V_{DD} - V_{GS1}}{R}$$

If

$$V_{DD}$$

changes, then current changes.

Fix: Keep

$$V_{DD}$$

constant

6.11.2 Systematic variations

If

$$V_{DS1} \neq V_{DS2} \rightarrow I_1 \neq I_2$$

If layout direction of

$$M_1 \neq M_2 \rightarrow I_1 \neq I_2$$

If current direction of

$$M_1 \neq M_2 \rightarrow I_1 \neq I_2$$

If

$$V_{S1} \neq V_{S2} \rightarrow I_1 \neq I_2$$

If

$$V_{B1} \neq V_{B2} \rightarrow I_1 \neq I_2$$

If

$$WPE_1 \neq WPE_2 \rightarrow I_1 \neq I_2$$

If

$$Stress_1 \neq Stress_2 \rightarrow I_1 \neq I_2$$

...

6.11.3 Process variations

Assume strong inversion and active

$$V_{eff} = \sqrt{\frac{2}{\mu_p C_{ox} \frac{W}{L}} I_1}$$

,

$$V_{GS} = V_{eff} + V_{tp}$$

$$I_1 = \frac{V_{DD} - V_{GS}}{R} = \frac{V_{DD} - \sqrt{\frac{2}{\mu_p C_{ox} \frac{W}{L}} I_1} - V_{tp}}{R}$$

$$\mu_p$$

,

$$C_{ox}$$

,

$$V_{tp}$$

will all vary from die to die, and wafer lot to wafer lot.

6.11.4 Process corners

Common to use 5 corners, or Monte-Carlo process simulation

Corner	NMOS	PMOS
Mtt	Typical	Typical
Mss	Slow	Slow
Mff	Fast	Fast
Msf	Slowish	Fastish
Mfs	Fastish	Slowish

6.11.5 Fix process variation

Use calibration: measure error, tune circuit to fix error

For every single chip, measure voltage across known resistor

$$R_1$$

and tune

$$R_{var}$$

such that we get

$$I_1 = 1\mu A$$

Be careful with multimeters, they have finite input resistance (approximately 1 M

$$\Omega$$

)

6.11.6 Temperature variation

Mobility decreases with temperature

Threshold voltage decreases with temperature.

$$I_D = \frac{1}{2}\mu_n C_{ox}(V_{GS} - V_{tn})^2$$

High

$$I_D =$$

fast digital circuits

Low

$$I_D =$$

slow digital circuits

What is fast? High temperature or low temperature?

6.11.7 It depends on

$$V_{DD}$$

Fast corner - Mff (high mobility, low threshold voltage) - High

$$V_{DD}$$

- High or low temperature

Slow corner - Mss (low mobility, high threshold voltage) - Low

$$V_{DD}$$

- High or low temperature

6.11.8 How do we fix temperature variation?

Accept it, or don't use this circuit.

If you need stability over temperature, use 7.3.2 and 7.3.4 in CJM (SUN_BIAS_GF130N)

6.11.9 Random Variation

$$\ell = \mu_p C_{ox} \frac{W}{L}$$

$$I_D = \frac{1}{2}\ell(V_{GS} - V_{tp})^2$$

Due to doping , length, width,

$$C_{ox}$$

,

$$V_{tp}$$

, . . . random varation

$$\ell_1 \neq \ell_2$$

$$V_{tp1} \neq V_{tp2}$$

As a result

$$I_1 \neq I_2$$

, but we can make them close.

6.11.10 Pelgrom's* law

Given a random gaussian process parameter

$$\Delta P$$

with zero mean, the variance is given by

$$\sigma^2(\Delta P) = \frac{A_P^2}{WL} + S_P^2 D^2$$

where

$$A_P$$

and

$$S_P$$

are measured, and

$$D$$

is the distance between devices

Assume closely spaced devices (

$$D \approx 0$$

)

$$\Rightarrow \sigma^2(\Delta P) = \frac{A_P^2}{WL}$$

6.11.11 Transistors with same

$$V_{GS}$$

†

$$\frac{\sigma_{I_D}^2}{I_D^2} = \frac{1}{WL}\left[\left(\frac{gm}{I_D}\right)^2 \sigma_{vt}^2 + \frac{\sigma_\ell^2}{\ell}\right]$$

Valid in weak, moderate and strong inversion

$$\frac{\sigma_{I_D}^2}{I_D^2} = \frac{1}{WL}\left[\left(\frac{gm}{I_D}\right)^2 \sigma_{vt}^2 + \frac{\sigma_\ell^2}{\ell}\right]$$

$$\frac{\sigma_{I_D}}{I_D} \propto \frac{1}{\sqrt{WL}}$$

---

* M. J. M. Pelgrom, C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," IEEE J. Solid-State Cir- cuits, vol. 24, no. 5, pp. 1433–1440, Oct. 1989.

† Peter Kinget, see CJM

Assume

$$\frac{\sigma_{I_D}}{I_D} = 10\%$$

, We want

$$5\%$$

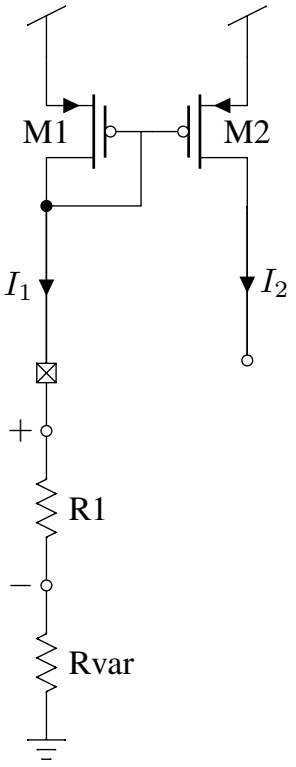
, how much do we need to change WL?

$$\frac{\frac{\sigma_{I_D}}{I_D}}{2} \propto \frac{1}{2\sqrt{WL}} = \frac{1}{\sqrt{4WL}}$$

We must quadruple the area to half the standard deviation

$$1\%$$

would require **100** times the area



6.11.12 What else can we do?

$$\frac{\sigma_{I_D}^2}{I_D^2} = \frac{1}{WL}\left[\left(\frac{gm}{I_D}\right)^2 \sigma_{vt}^2 + \frac{\sigma_\ell^2}{\ell}\right]$$

Strong inversion

$$\Rightarrow \frac{gm}{I_D} = \frac{1}{2V_{eff}} = low$$

Weak inversion

$$\Rightarrow \frac{gm}{I_D} = \frac{q}{nkT} \approx 25$$

Current mirrors achieve best matching in strong inversion

$$\frac{\sigma^2_{I_D}}{I^2_D} = \frac{1}{WL}\left[\left(\frac{gm}{I_D}\right)^2 \sigma^2_{vt} + \frac{\sigma^2_\ell}{\ell}\right]$$

$$\sigma^2_{I_D} = \frac{1}{WL}\left[gm^2\sigma^2_{vt} + I^2_D\frac{\sigma^2_\ell}{\ell}\right]$$

Offset voltage for a differential pair

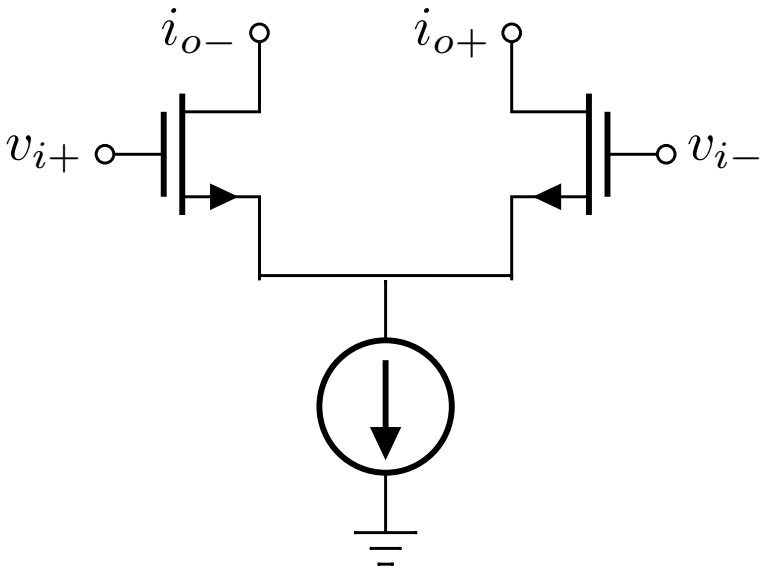
$$i_o = i_{o+} - i_{o-} = g_m v_i = g_m(v_{i+} - v_{i-})$$

$$\sigma^2_{v_i} = \frac{\sigma^2_{I_D}}{gm^2} = \frac{1}{WL}\left[\sigma^2_{vt} + \frac{I^2_D}{gm^2}\frac{\sigma^2_\ell}{\ell}\right]$$

High

$$\frac{gm}{I_D}$$

is better (best in weak inversion)



6.11.13 Transistor Noise

Thermal noise Random scattering of carriers, generation-recombination in channel?

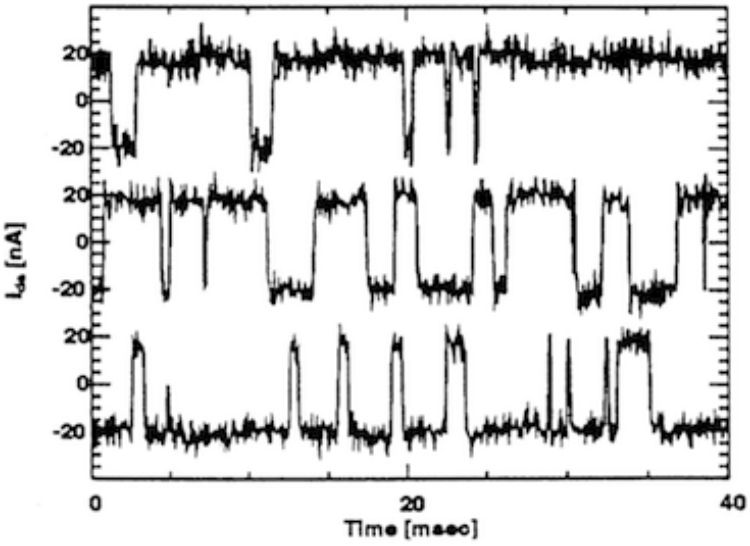
$$PSD_{TH}(f) = \text{Constant}$$

Popcorn noise Carriers get "stuck" in oxide traps (dangling bonds) for a while. Can cause a short-lived (seconds to minutes) shift in threshold voltage

$$PSD_{GR}(f) \propto \text{Lorentzian shape} \approx \frac{A}{1 + \frac{f^2}{f_0}}$$

Flicker noise Assume there are many sources of popcorn noise at different energy levels and time constants, then the sum of the spectral densities approaches flicker noise.

$$PSD_{flicker}(f) \propto \frac{1}{f}$$



Circuits 7

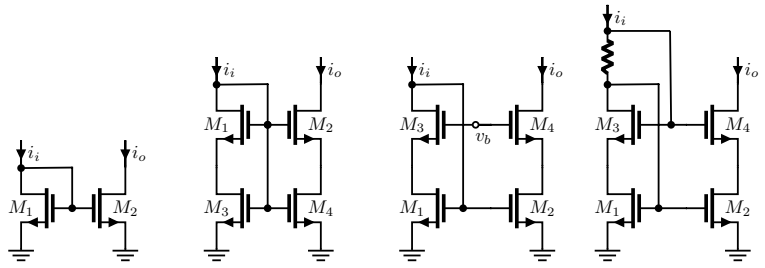
7.1 Current Mirrors

MOSFETs need a current for the transistor to be biased in the correct operating region. The current must come from somewhere, we'll look at bias generators later. Usually there is a central bias circuit that provides a single, good, reference current.

On an IC, however, there will be many circuits, and they all need a bias current (usually). As such, we need a circuit to copy a current.

In the figure below you can see a selection of current mirros. They all do the same thing. Try to ensure that i_i and i_o are the same current.

Which one we choose is usually determined by what we mean by $i_i = i_o$. Do we mean "within ± 10 %", or "within ± 2 %".



a) "normal" b) Self Cascode c) Cascode d) Lazy Cascode

7.1.1 Normal current mirror

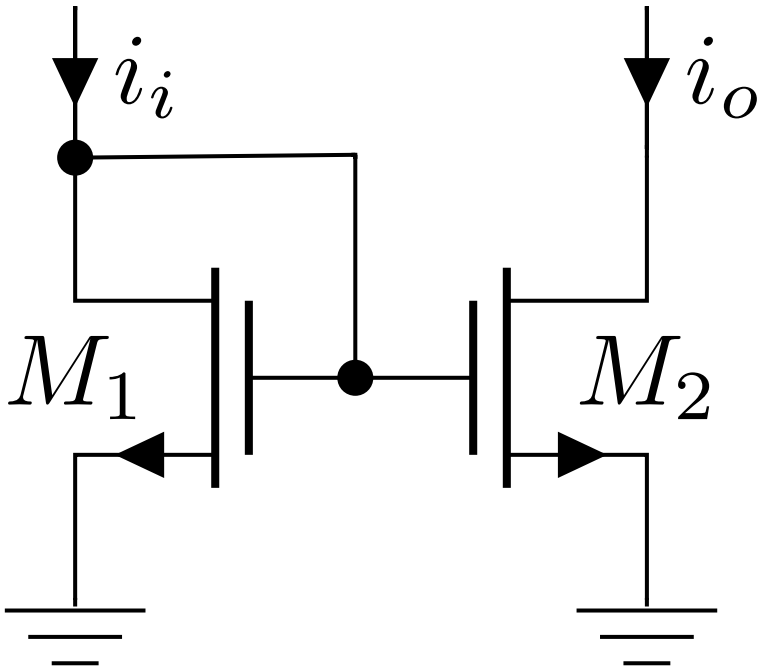
The normal current mirror consists of a diode connected transistor (M_1) and a common source transistor M_2.

If we assume infinite output resistance of the MOSFETs, then the drain voltage does not affect the current.

If the two transistors are the same size, threshold voltage, mobility, etc, and they have the same gate-source voltage, then the current in them must be the same.

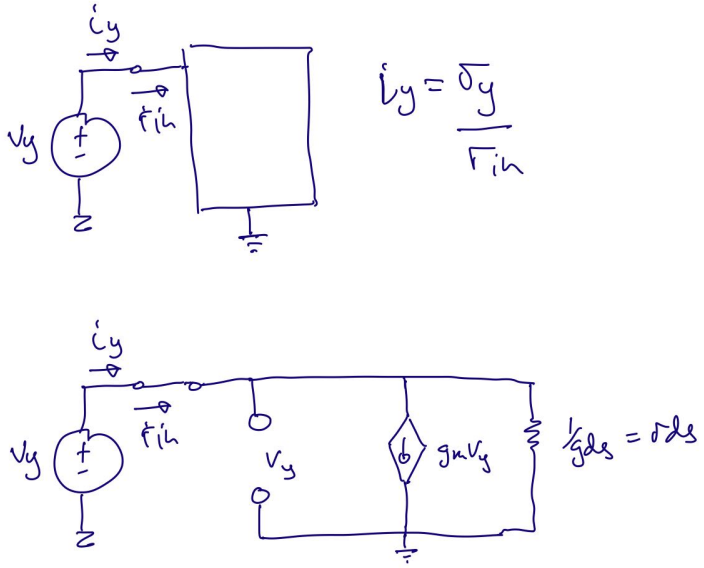
A current pushed into M_1 will cause the V_{GS1} to rise, and at some point, find a stable point where the current pushed in is equal to the current in M_1

M_2 will see the same $V_{GS1} = V_{GS2}$ so the current will be the same, provided the voltage at i_o is sufficient to pinch-off the channel of M_2, or the $V_{DS2} \approx 3kT/q$ if the transitor is in weak-inversion.



7.1.1.1 Input resistance

To see the small signal input resistance we can apply a test voltage to the diode connected resistor, as shown in the figure below.



Observe the current

$$i_y = g_{ds} v_y + g_m v_y$$

While the input resistance

$$r_{in} = \frac{v_y}{i_y} = \frac{1}{g_m + g_{ds}}$$

which, assuming $g_{ds} >> g_m$, reduces to

$$r_{in} \approx \frac{1}{g_m}$$

.

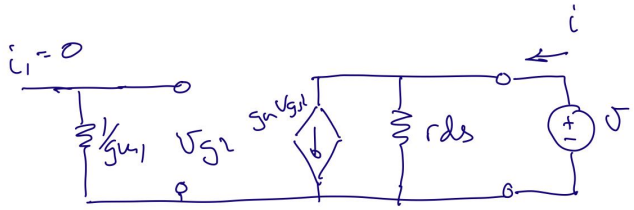
Assume now I apply 1 μA current to the diode connected transistor, and the $g_m = 1\ \mu S$.

Would the voltage be $v_y = r_{in} i_y = \frac{1\ \mu A}{1\ \mu S} = 1\ V$? NO! It's important to understand the difference between the small signal input resistance, and the large signal impedance.

The large signal impedance is a highly non-linear function (we've seen before that the current in a MOSFET has both an exponential, and a square-law, and sometimes a linear with voltage), as such, there is no single function describing what the gate-source voltage will be.

To see the DC voltage, apply a current in SPICE, and use a simulator to find the voltage.

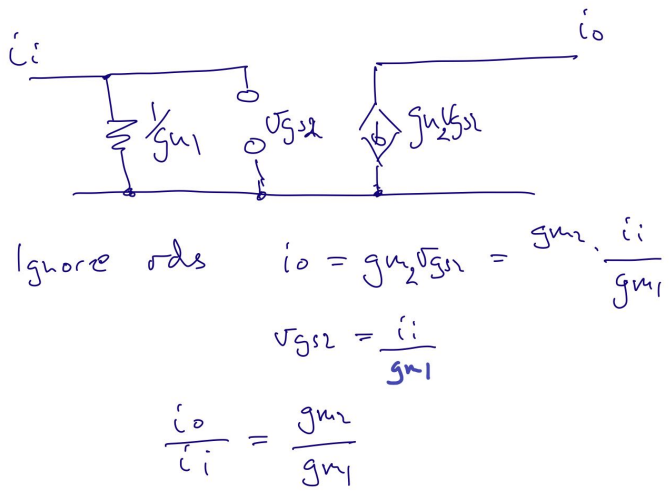
7.1.1.2 Output resistance



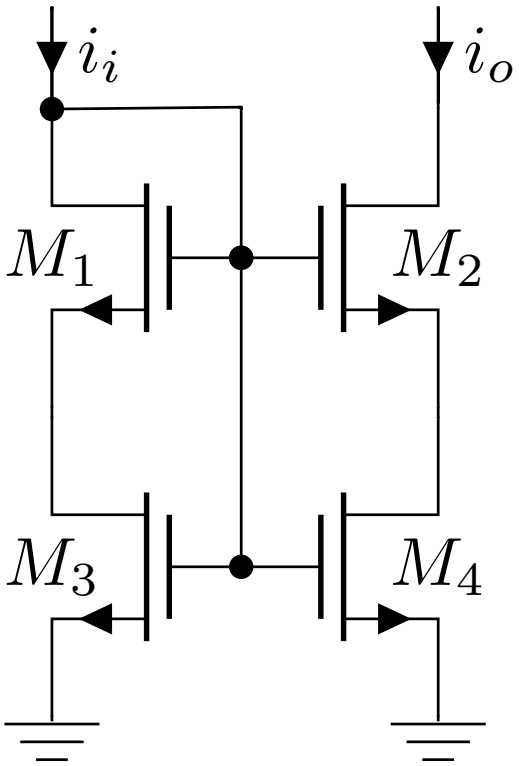
V_y does not affect V_{gs2}

$r_{out} = r_{ds}$

7.1.1.3 Current gain



7.1.2 Source degeneration



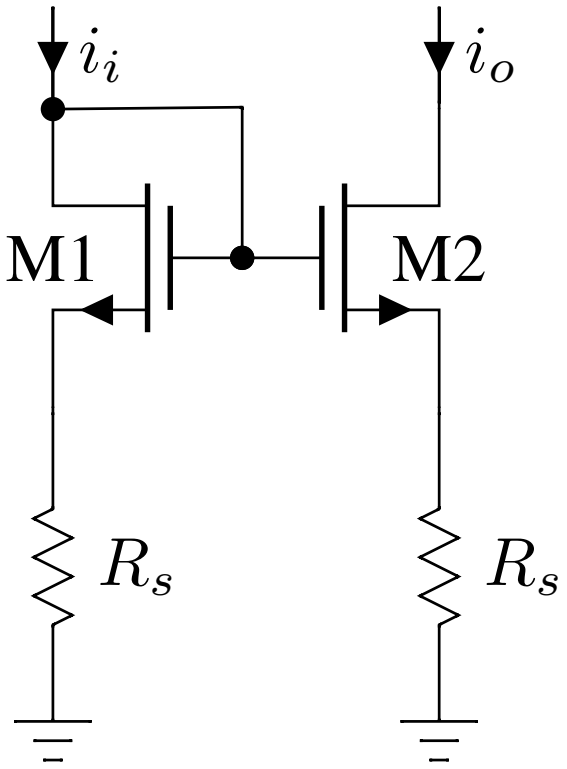
What is the operating region of M3 and M4?

What is the operating region of M1 and M2?

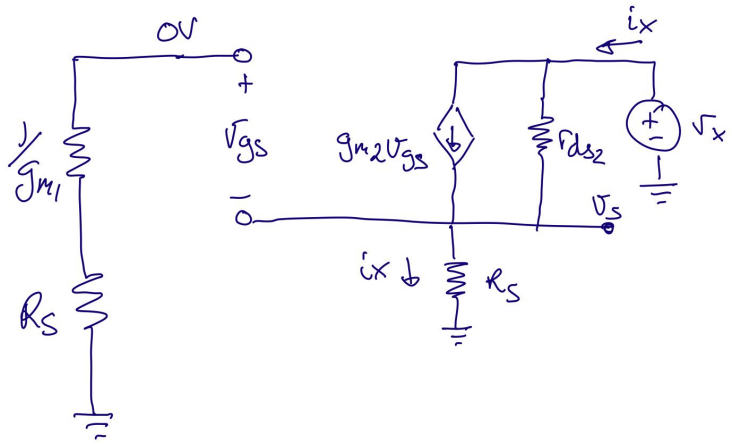
7.1.2.1 Input resistance

M1 and M2 are in linear region, can be simplified to resistors

$$r_{in} = \frac{1}{g_{m1}} + R_s$$



7.1.3 Output resistance



$$v_{gs} = -v_s$$

,

$$v_s = i_x R_s$$

,

$$r_{out} = \frac{v_x}{i_x}$$

$$i_x = g_{m2} v_{gs} + \frac{v_x - v_s}{r_{ds2}}$$

$$i_x = -i_x g_{m2} R_s + \frac{v_x - i_x R_s}{r_{ds2}}$$

$$v_x = i_x \left[r_{ds2} + R_s (g_{m2} r_{ds2} + 1) \right]$$

Rearranging

$$r_{out} = r_{ds2}[1 + R_s(g_{m1} + g_{ds2})] \approx r_{ds2}[1 + g_{m1} R_s]$$

7.1.3.1 Cascode output resistance

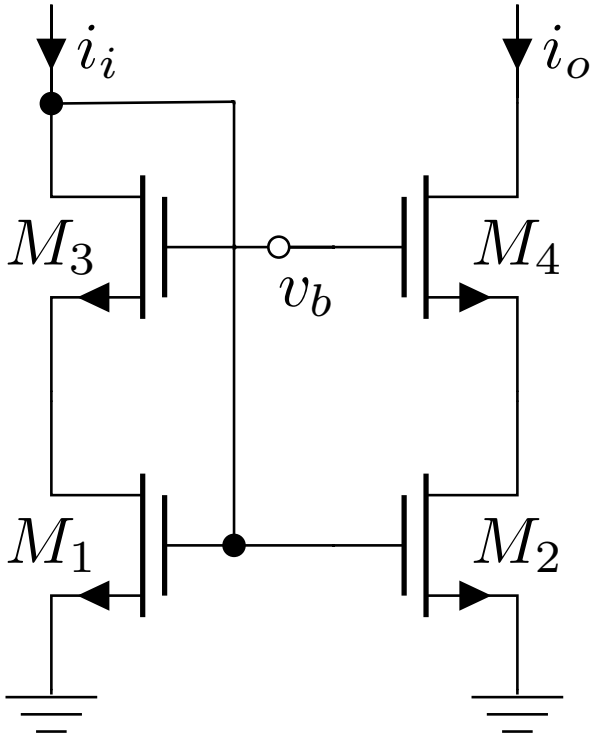
From source degeneration (ignoring bulk effect)

$$r_{out} = r_{ds4}[1 + R_s(g_{m4} + g_{ds4})]$$

$$R_S = r_{ds2}$$

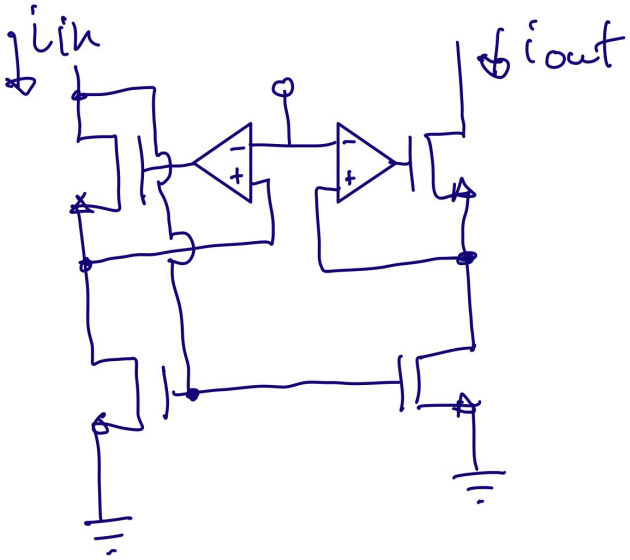
$$r_{out} = r_{ds4}[1 + r_{ds2}(g_{m4} + g_{ds4})]$$

$$r_{out} \approx r_{ds2}(r_{ds4} g_{m4})$$



7.1.3.2 Active cascodes

$$r_{out} \approx r_{ds2}(Ar_{ds4}g_{m4})$$



7.2 Amplifiers

7.3 Source follower

Input resistance

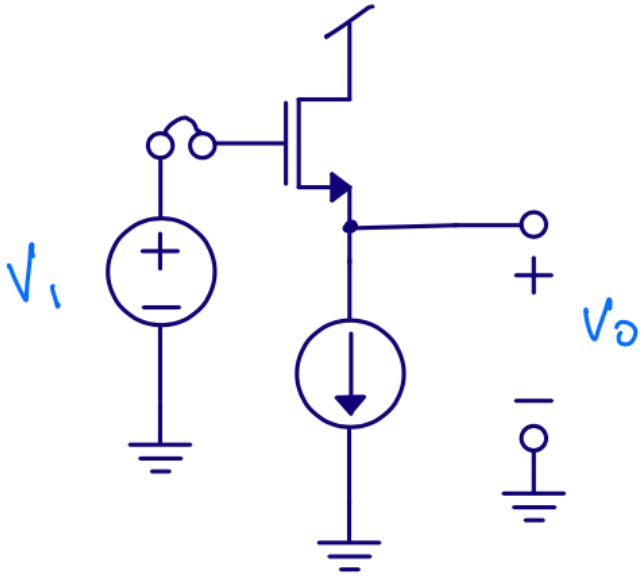
$$\approx \infty$$

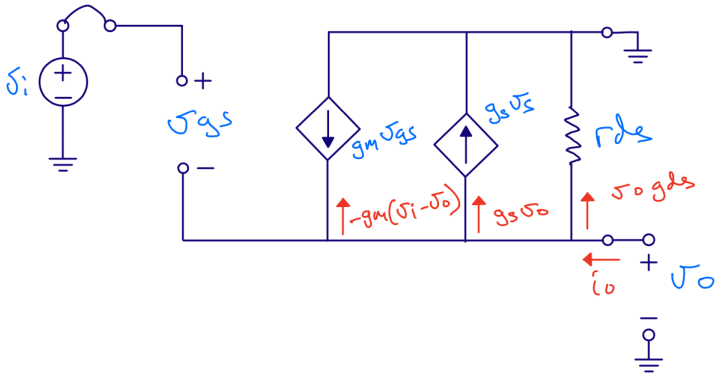
Gain

$$A = \frac{v_o}{v_i}$$

Output resistance

$$r_{out}$$





7.3.1 Output resistance

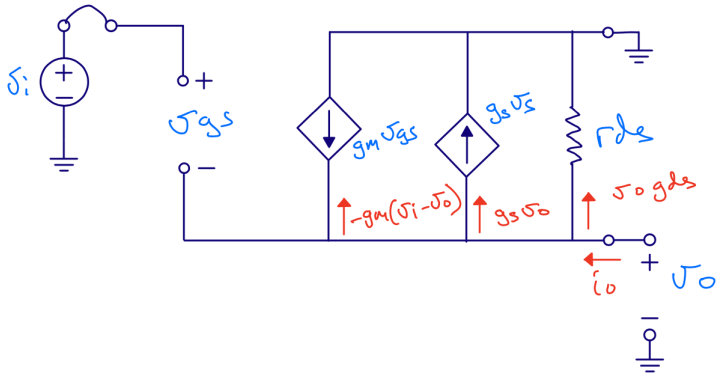
$$i_o = v_o(g_{ds} + g_s) - g_m v_i + v_o g_m$$

$$v_i = 0$$

$$i_o = v_o(g_{ds} + g_s + g_m)$$

$$r_{out} = \frac{v_o}{i_o} = \frac{1}{g_m + g_{ds} + g_s}$$

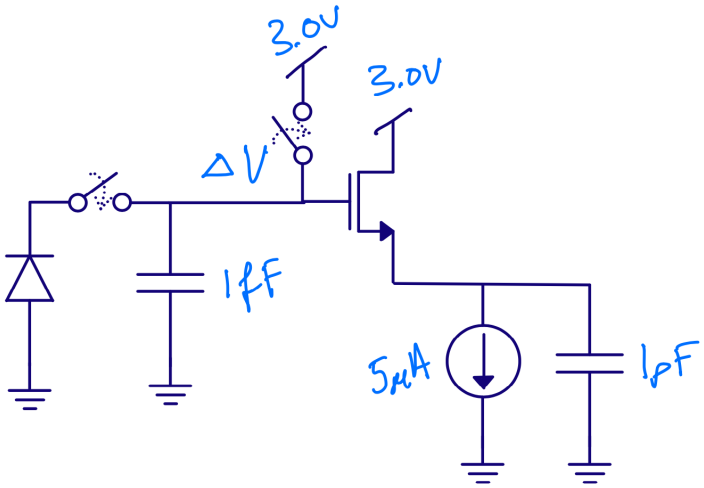
$$r_{out} \approx \frac{1}{g_m}$$



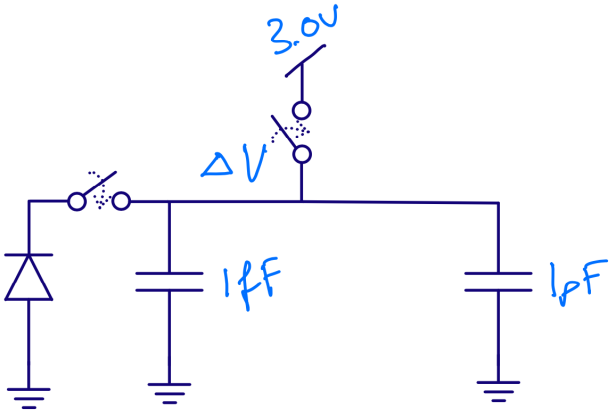
7.3.2 Why use a source follower?

Assume 100 electrons

$$\Delta V = Q/C = -1.6 \times 10^{-19} \times 100/(1 \times 10^{-15}) = -16 \text{ mV}$$



$$\Delta V = Q/C = -1.6 \times 10^{-19} \times 100/(1 \times 10^{-12}) = -16 \text{ uV}$$

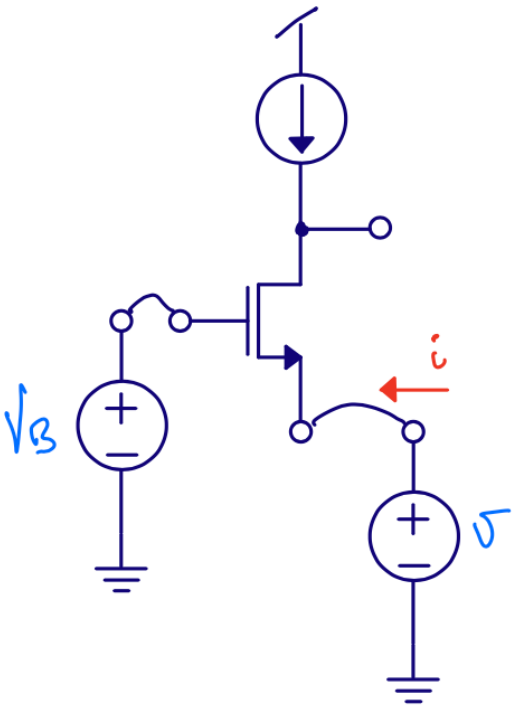


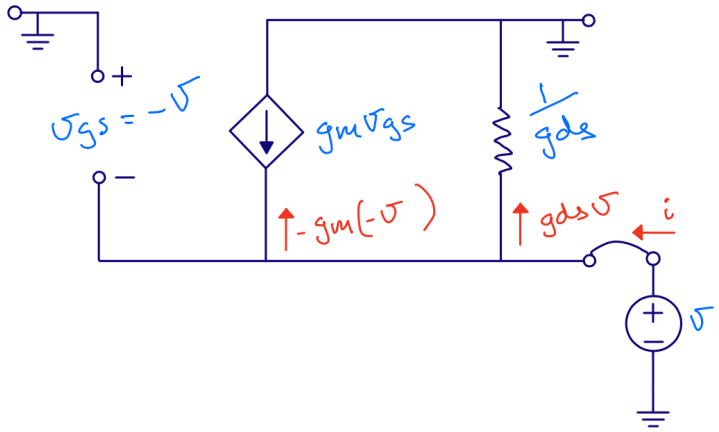
7.4 Common gate

Input resistance

Gain

Output resistance





7.4.1 Input resistance

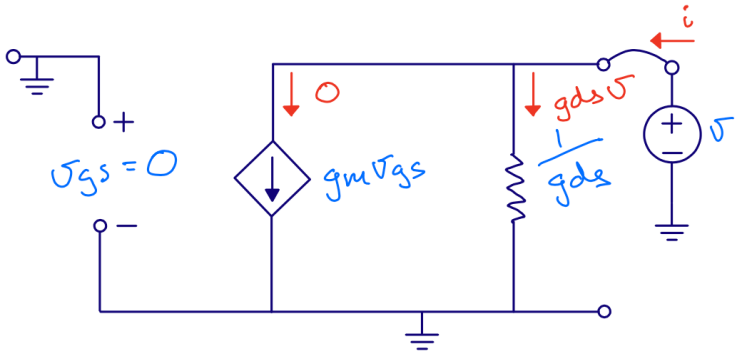
$$i = g_m v + g_{ds} v$$

$$r_{in} = \frac{1}{g_m + g_{ds}} \approx \frac{1}{g_m}$$

However, we've ignored load resistance.

$$r_{in} \approx \frac{1}{g_m}\left(1 + \frac{R_L}{r_{ds}}\right)$$

7.4.2 Output resistance



7.4.3 Gain

$$i_o = -g_m v_i + \frac{v_o - v_i}{r_{ds}}$$

$$i_o = 0$$

$$0 = -g_m v_i r_{ds} + v_o - v_i$$

$$v_i(1 + g_m r_{ds}) = v_o$$

$$\frac{v_o}{v_i} = 1 + g_m r_{ds}$$

We've ignored bulk effect (g_s), source resistance (R_S) and load resistance (R_L)

$$A = \frac{(g_m + g_s + g_{ds})(R_L || r_{ds})}{1 + R_S \left(\frac{g_m + g_s + g_{ds}}{1 + R_L / r_{ds}} \right)}$$

If $R_L >> r_{ds}$, $R_S = 0$ and $g_s = 0$

$$A = \frac{(g_m + g_{ds}) r_{ds}}{1} = 1 + g_m r_{ds}$$

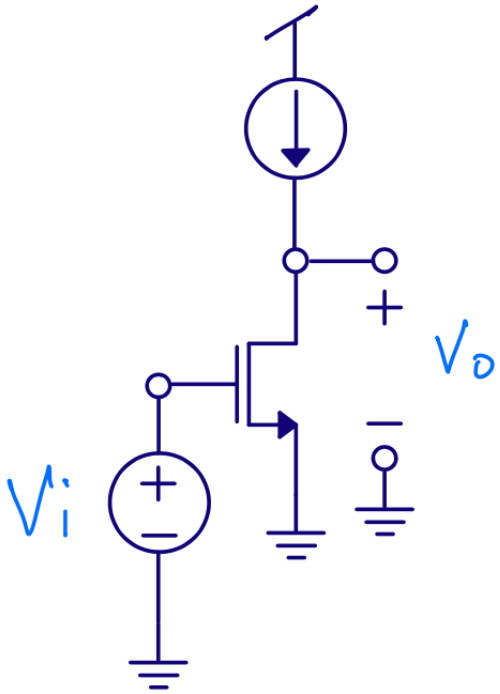
7.5 Common source

$$r_{in} \approx \infty$$

$$r_{out} = r_{ds}$$

, it's same circuit as the output of a current mirror

Gain



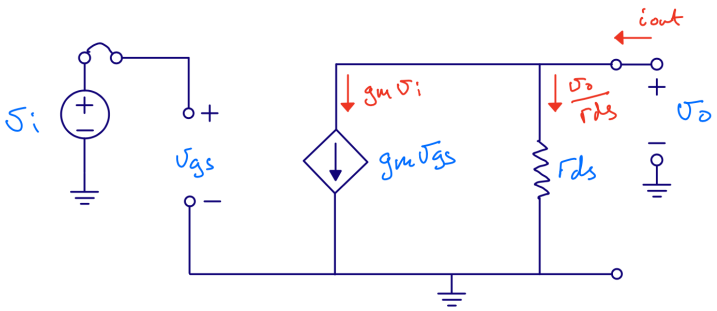
7.5.1 Gain

$$i_o = g_m v_i + \frac{v_o}{r_{ds}}$$

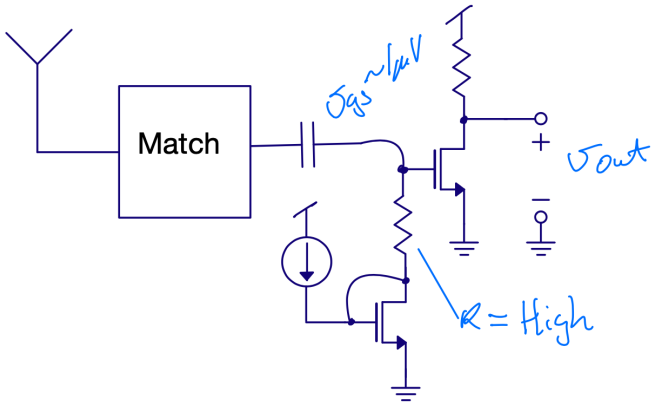
$$i_o = 0$$

$$-g_m v_i = \frac{v_o}{r_{ds}}$$

$$\frac{v_o}{v_i} = -g_m r_{ds}$$



7.5.2 Why common source?



7.6 Differential pair

Input resistance

$$r_{in} \approx \infty$$

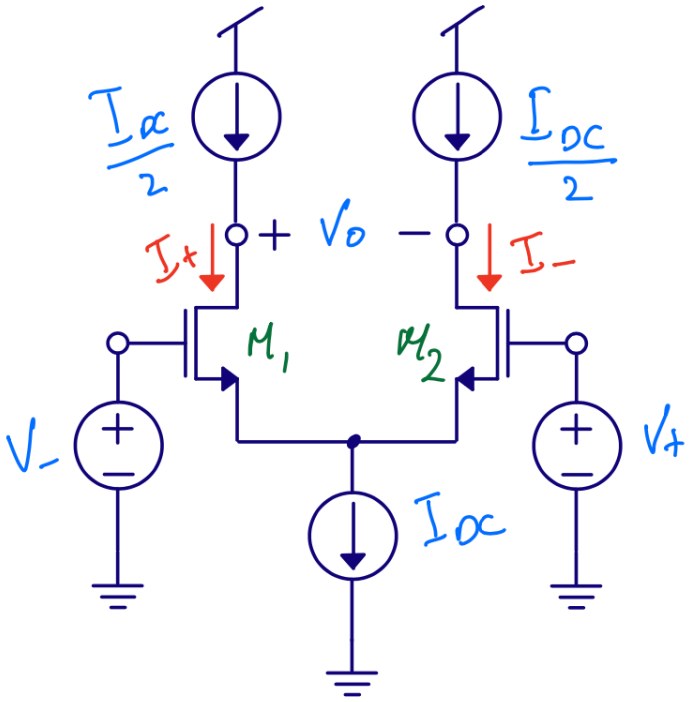
Gain

$$A = g_m r_{ds}$$

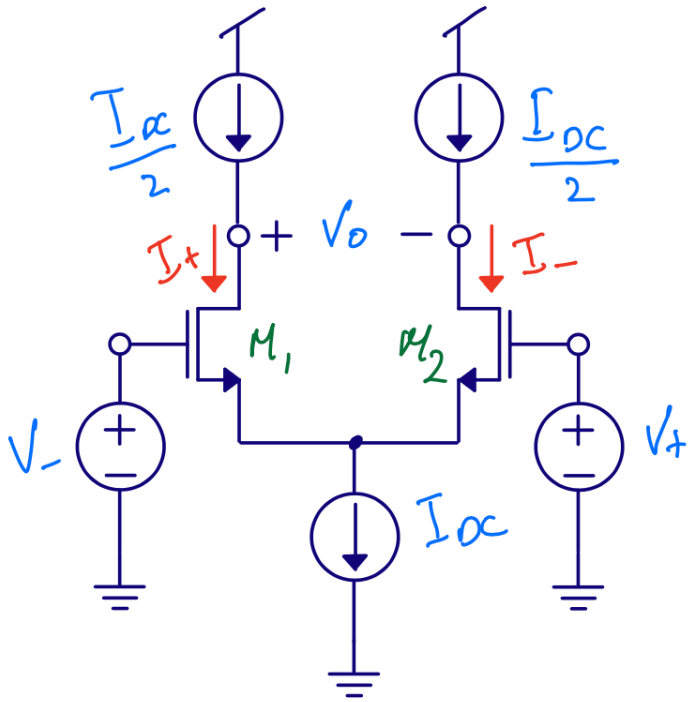
Output resistance

$$r_{out} = r_{ds}$$

Best analyzed with T model of transistor (see CJM page 31)



7.6.1 Diff pairs are cool



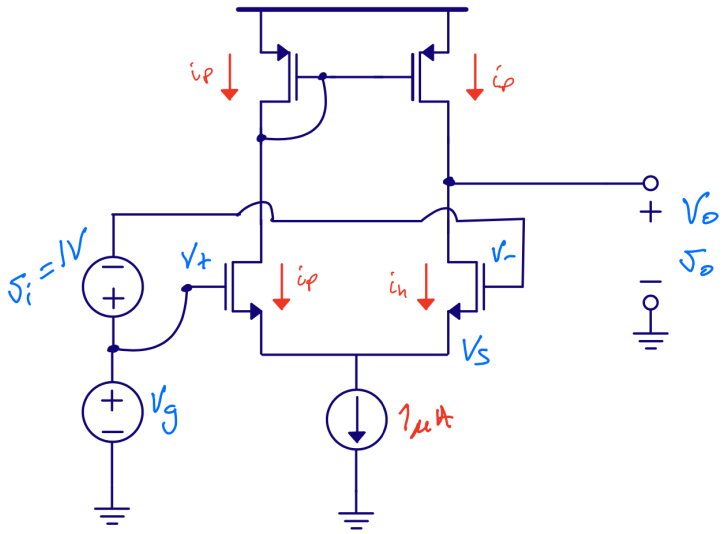
Can choose between

$$v_o = g_m r_{ds} v_i$$

and

$$v_o = -g_m r_{ds} v_i$$

by flipping input (or output) connections



Integrated Passives 8

Status: 0.3

8.1 Metal in ICs is not wire in schematic

Metal wires in an integrated circuit comes in two types, copper and aluminium.

Most of the routing layers will be copper. To ensure that the copper ions don't diffuse into the silicon-oxide a barrier material surrounds all copper interconnect.

Copper is too stiff to be wire-bonded. As such, the top layer metals would be aluminium.

Since the routing is so small, we have to care about the parasitic properties of the routing. Below is a table with some common quantities for copper. For example, if we have 1000 μm metal wire with 1 μm width, then it would be approximately 150 Ω, 1 nH , 1 pF and tolerate a maximum of 1 mA DC current.

Parameter	Typ. Value	Unit
Resistance	150	mΩ/□
Capacitance	1	fF/μm
Inductance	1	nH/mm
Max DC current	1	mA/□

The type of circuit we have determine what we must simulate. Everything needs to be simulated with parasitc capacitance and max current. Only RF, however, usually needs to be simulated with resistance, capacitance, inductance and maximum current.

Circuit type	Must simulate/know
All	C Imax
Analog, Power	R C Imax
Some RF, Some Power	R L C Imax

To simulate the effects of parasitics, we need a description of the technology. A Process Design Kit (PDK). Most PDKs are closely guarded secrets, as they describe many things about the way the foundry makes the integrated circuits.

Some PDKs are open source, however, see Skywater 130 nm and IHP-Open-PDK

In addition to the PDK, we need tools that can calculate from the layout the parasitic elements. Some of the tools are

Layout parasitic extraction tools

- Calibre xRC
- Synopsys StarRC
- Cadence Quantus
- Magic VLSI

3D EM Simulators

- Keysight ADS
- HFSS

Transistor CAD (TCAD)

- Synopsys TCAD

8.2 Resistors

Sometimes we want a specific resistance. In general, any resistance on IC will vary in absolute value by maybe up to ± 20 %. The relative size, however, can be controlled to within 0.1 %.

In other words, you can't rely on a 1 kOhm resistor actually being 1 kOhm, it might be 0.8 kOhm. If you have two, however, you can trust that both of them will be 0.8 kOhm.

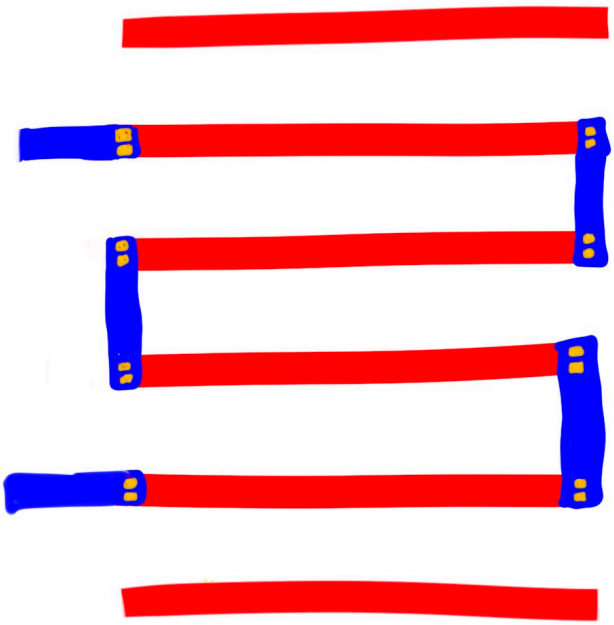
That's why almost all analog circuits rely on the relative sizes of passives, not the absolute value. If a circuit does rely on absolute values, then it usually needs to be trimmed in production.

8.2.1 Polysilicon

Can be both N-doped, and P-doped

Often with two flavors, with, and without silicide

Silicide reduces resistance of polysilicon



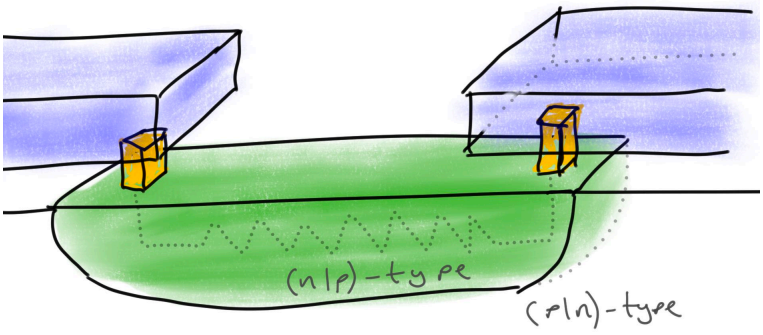
8.2.2 Diffusion

Use doped region as resistor

Usually without silicide

Non-linear capacitance

Tricky temperature dependence

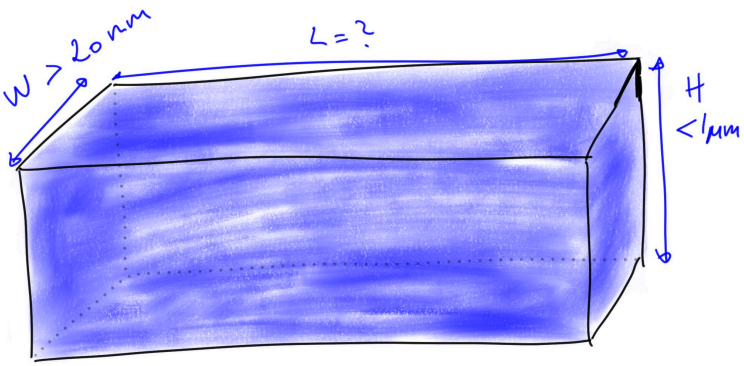


8.2.3 Metal

Usually too low omhic to be a useful resistor

Useful for “separating nets” in schematic and layout

Must be considered for power supply and ground routing (high currents)



8.3 Capacitors

8.3.1 What is S, M, L, XL on a chip?

nRF52832

$$3200\mu m \times 3000\mu m = 9600k\mu m^2$$

S

$$< 5\ k\mu m^2$$

M

$$< 50\ k\mu m^2$$

L

$$< 200\ k\mu m^2$$

XL

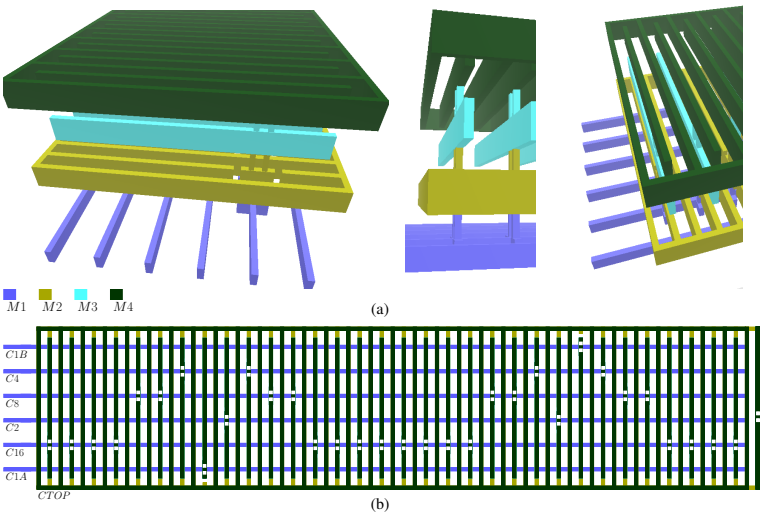
$$> 200\ k\mu m^2$$

8.3.2 Metal-Oxide-Metal finger capacitors

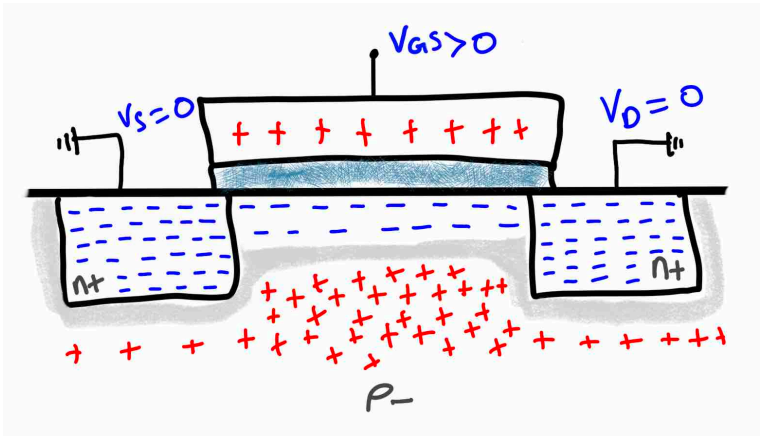
Unit capacitance

$$\approx 1fF/\mu m^2/layer$$

$$10pF = 100\mu m \times 100\mu m = 10k\mu m^2$$



8.3.3 MOS capacitors




```
dicex/sim/spice/NCHIO/vcap.cir
* gate cap

.include ../../../models/ptm_130.spi

vdrain D 0 dc 1
vgaini G 0 dc 0.5
vbulk B 0 dc 0
vcur S 0 dc 0

M1 D G S B nmos  w=1u  l=1u

.op
```

Moscap is

$$\approx 10fF/\mu m^2$$

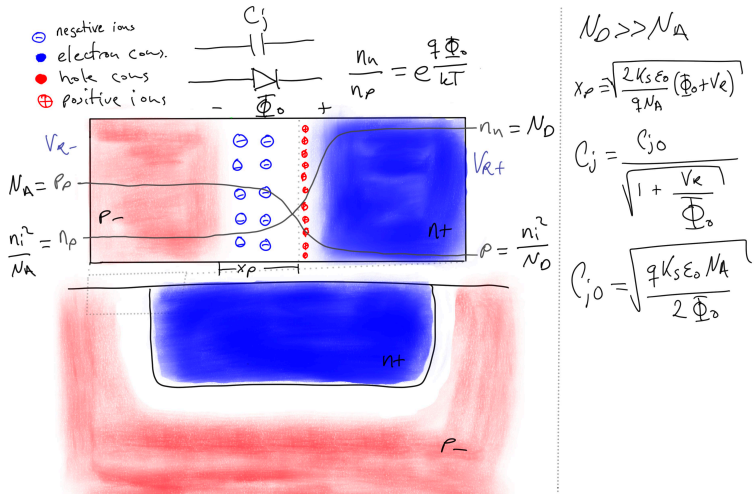
$$10pF = 31\mu m \times 31\mu m \approx 1k\mu m^2$$

```
dicex/sim/spice/NCHIO/vcap.vlog
Device m1:
    Vgs     (gate-source voltage)        [V] : 0.5
    Vgd     (gate-drain voltage)         [V] : -0.5
    Vds     (drain-source voltage)       [V] : 1
```

```
Vbs      (bulk-source voltage)          [V] : 1.90808e-12
Vbd      (bulk-drain voltage)           [V] : -1
Id       (drain current)                [A] : 7.32634e-06
Is       (source current)               [A] : -7.32633e-06
Ibd      (bulk-drain current)           [A] : -1.01e-12
Ibs      (bulk-source current)          [A] : 9.581e-25
Vt       (threshold voltage)            [V] : 0.378198
Vgt      (gate overdrive voltage)       [V] : 0.121802
Vgsteff  (effective vgt)                [V] : 0.12515
Gm       (transconductance)             [S] : 8.44164e-05
Gmb      (bulk bias transconductance)   [S] : 2.00071e-05
Ueff     (mobility)              [cm^2/Vs] : 417.675
Gds      (channel conductance)          [S] : 1.95043e-07
Rds      (output resistance)         [Ohm] : 5.12708e+06
Vdsat    (drain saturation voltage)     [V] : 0.14171
IC       (inversion coefficient)          [] : 4.42478
Cgs      (gate-source capacitance)      [F] : 9.98457e-15
Csg      (source-gate capacitance)      [F] : 5.86932e-15
Cgd      (gate-drain capacitance)       [F] : 3.98239e-16
Cdg      (drain-gate capacitance)       [F] : 3.91086e-15
Cds      (drain-source capacitance)     [F] : 4.30968e-15
Cgg      (gate-gate capacitance)        [F] : 1.05198e-14
Cdd      (drain-drain capacitance)      [F] : 1.05198e-14
Css      (source-source capacitance)    [F] : 0
Cgb      (gate-bulk capacitance)        [F] : 1.05198e-14
Cbg      (bulk-gate capacitance)        [F] : 1.74123e-15
Cbs      (bulk-source capacitance)      [F] : 8e-16
Cbd      (bulk-drain capacitance)       [F] : 3.97768e-16
```

8.3.4 Varactors

A varactor is a “variable capacitor”, usually it’s a device that varies the capacitance with the voltage across the device.

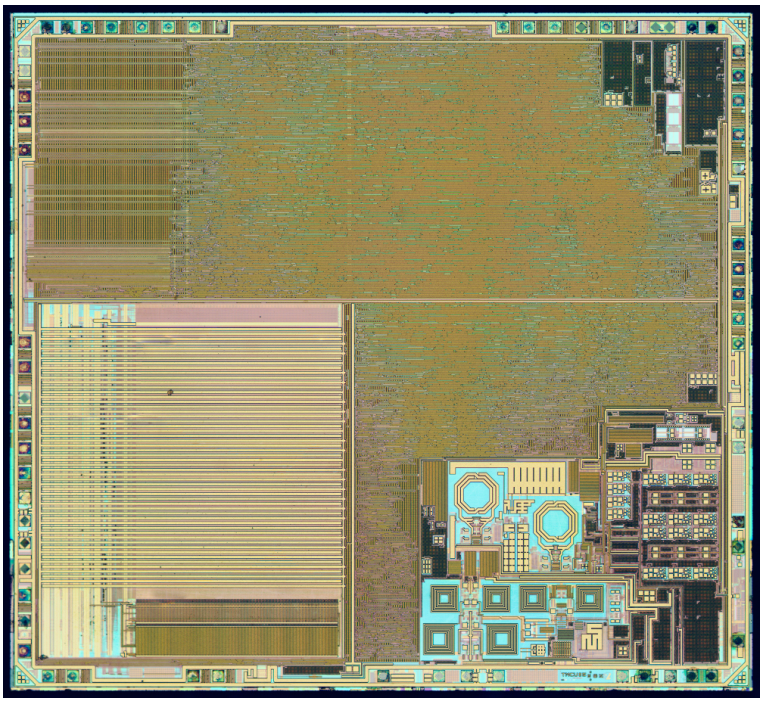


8.4 Inductors

Usually two top metals, because they are thick (low ohmic)

Use foundry model

3D electro magnetic simulation often needed



8.5 Variation in passives

Absolute value for resistors and capacitors

$$\approx \pm 10$$

% to

$$\pm 20$$

%

Relative precision for closely spaced devices

$$\approx$$

0.1 % to 1 %

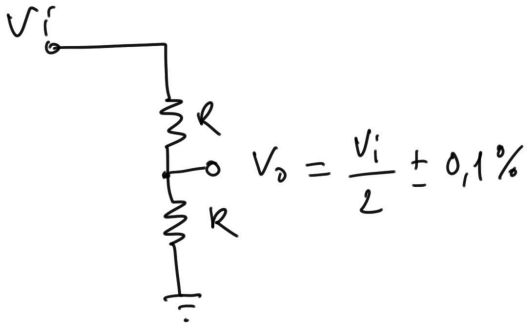
Relative precision for devices on same die

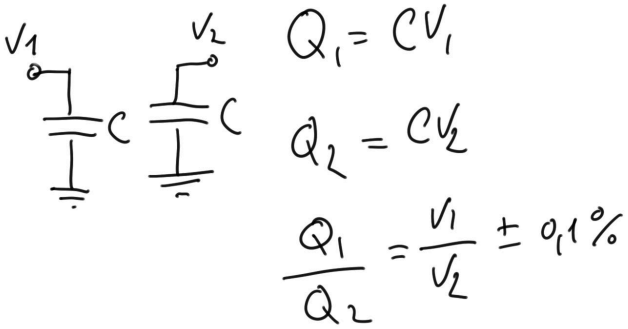
$$> 2$$

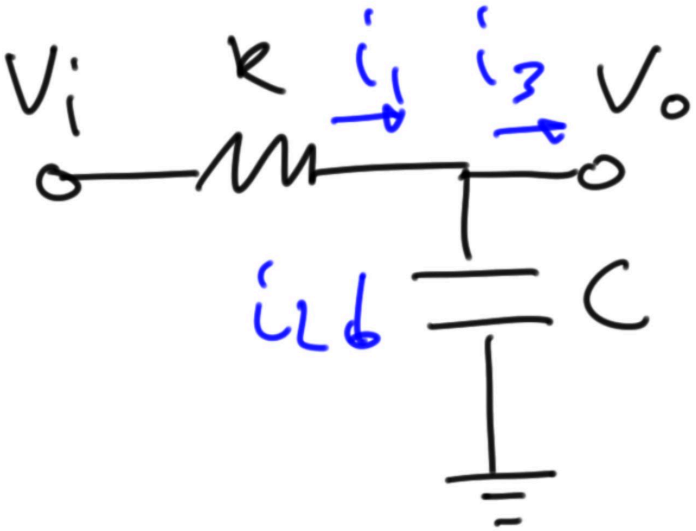
% or more

8.6 Relative precision

Resistors and Capacitors can be matched extremely well







$$i_3 = 0 = i_1 - i_2$$

$$0 = \frac{V_i - V_o}{R} - \frac{V_o}{1/sC}$$

$$0 = V_i - V_o - V_o sRC$$

$$V_o(1 + sRC) = V_i$$

$$\frac{V_o}{V_i} = \frac{1}{1 + sRC}$$

Assume standard deviation (

$$\sigma$$

)* of

$$\sigma_R = 20$$

%,

$$\sigma_C = 20$$

%

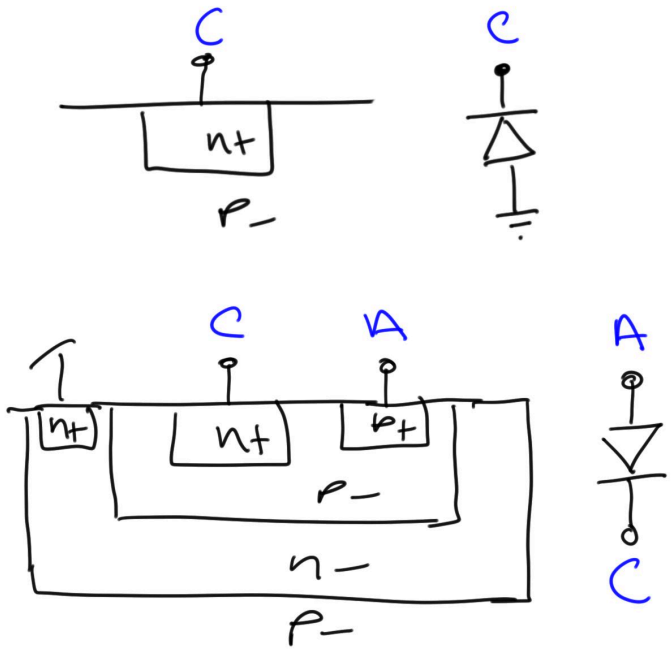
$$\sigma_{RC} = \sqrt{0.2^2 + 0.2^2} = 28$$

%

8.7 Diodes

Many, many ways

Reverse bias diodes to ground are useful for signals with long routing to transistor gate. Protects gate from breakdown during chemical mechanical polish.



* If you don't remember how standard deviation works, read Introduction to mathematics of noise sources

Noise 9

Status: 0.5

9.1 Noise

Noise is a phenomena that occurs in all electronic circuits. It places a lower limit on the smallest signal we can use. Many now have super audio compact disc (SACD) players with 24bit converters, 24 bits is around $2^{24} = 16.78$ Million different levels. If 5V is the maximum voltage, the minimum would have to be $\frac{5V}{2^{24}} \approx 298nV$. That level is roughly equivalent to the noise in a 50 Ohm resistor with a bandwith of 96kHz. There exist an equation that relates number of bits to signal to noise ratio 1, the equation specifies that $SNR = 6.02 * Bits + 1.76 = 146.24dB$. As of 12.2005 the best digital to analog converter (DAC) that Analog Devices (a very big semiconductor company) has is a DAC with 120dB SNR, that equals around $Bits = (120 - 1.76)/6.02 = 19.64$. In other words, the last four bits of your SACD player is probably noise!

9.2 Statistics

The mean of a signal x(t) is defined as

$$\overline{x(t)} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x(t)dt$$

The mean square of x(t) defined as

$$\overline{x^2(t)} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x^2(t)dt$$

The variance of x(t) defined as

$$\sigma^2 = \overline{x^2(t)} - \overline{x(t)}^2$$

For a signals with a mean of zero the variance is equal to the mean square. The auto-correlation of x(t) is defined as

$$\begin{aligned} R_x(\tau) &= \overline{x(t)x(t+\tau)} \\ &= \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x(t)x(t+\tau)dt \end{aligned}$$

9.3 Average Power

Average power is defined for a continuous system as ([eq:powcont]) and for discrete samples it can be defined as ([eq:powsamp]).

P_{av} usually has the unit A^2 or V^2, so we have to multiply/devide by the impedance to get the power in Watts. To get Volts and Amperes we use the root-mean-square (RMS) value which is defined as $\sqrt{P_{av}}$.

$$P_{av} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x^2(t)dt$$

$$P_{av} = \frac{1}{N} \sum_{i=0}^{N} x^2(i)$$

If x(t) has a mean of zero then, according to ([eq:var]), P_{av} is equal to the variance of x(t).

Many different notations are used to denote average power and RMS value of voltage or current, some of them are listed in Table [t:avgpow] and Table 2. Notation can be a confusing thing, it changes from book to book and makes expressions look different.

It is important to realize that it does not matter how you write average power and RMS value. If you want you can invent your own notation for average power and RMS value. However, if you are presenting your calculations to other people it is convenient if they understand what you have written. In the remainder of this paper we will use $\overline{e_n^2}$ for average power when we talk about voltage noise source and $\overline{i_n^2}$ for average power when we talk about current noise source. The n subscript is used to identify different sources and can be whatever.

Voltage	Current
V_{rms}^2	I_{rms}^2
$\overline{V_n^2}$	$\overline{I_n^2}$
$\overline{v_n^2}$	$\overline{i_n^2}$

Voltage	Current
V_{rms}	I_{rms}
$\sqrt{\overline{V_n^2}}$	$\sqrt{\overline{I_n^2}}$
$\sqrt{\overline{v_n^2}}$	$\sqrt{\overline{i_n^2}}$

9.4 Noise Spectrum

With random noise it is useful to relate the average power to frequency. We call this Power Spectral Density (PSD). A PSD plots how much power a signal carries at each frequency. In literature $S_x(f)$ is often used to denote the PSD. In the same way that we use V^2 as unit of average power, the unit of the PSD is $\frac{V^2}{Hz}$ for voltage and $\frac{A^2}{Hz}$ current. The root spectral density is defined as $\sqrt{S_x(f)}$ and has unit $\frac{V}{\sqrt{Hz}}$ for voltage and $\frac{I}{\sqrt{Hz}}$ for current.

The power spectral density is defined as two times the Fourier transform of the auto-correlation function 2

$$S_x(f) = 2\int_{-\infty}^{\infty} R_x(\tau)e^{-j2\pi f\tau}d\tau$$

This can also be written as

$$\begin{aligned}
S_x(f) = \quad & 2\left[\int_{-\infty}^{\infty} R_x(\tau)\cos(\omega\tau)d\tau - \int_{-\infty}^{\infty} R_x(\tau)j\sin(\omega\tau)d\tau\right] \\
= \quad & 2\left[\int_{-\infty}^{0} R_x(\tau)\cos(\omega\tau)d\tau + \int_{0}^{\infty} R_x(\tau)\cos(\omega\tau)d\tau\right] \\
- \quad & 2j\left[\int_{-\infty}^{0} R_x(\tau)\sin(\omega\tau)d\tau + \int_{0}^{\infty} R_x(\tau)\sin(\omega\tau)d\tau\right] \\
= \quad & 4\int_{0}^{\infty} R_x(\tau)\cos(\omega\tau)d\tau \\
- \quad & 2j\left[-\int_{0}^{\infty} R_x(\tau)\sin(\omega\tau)d\tau + \int_{0}^{\infty} R_x(\tau)\sin(\omega\tau)d\tau\right] \\
= \quad & 4\int_{0}^{\infty} R_x(\tau)\cos(\omega\tau)d\tau
\end{aligned}$$

, since $e^{-j\omega\tau} = \cos(\omega\tau) - j\sin(\omega\tau)$, $R_x(\tau)$ and $\cos(\omega\tau)$ are symmetric around $\tau = 0$ while $\sin(\omega\tau)$ is asymmetric around $\tau = 0$.

The inverse of power spectral density is defined as

$$R_x(\tau) = \frac{1}{2}\int_{-\infty}^{\infty} S_x(f)e^{j2\pi f\tau}df = \int_{0}^{\infty} S_x(f)\cos(\omega\tau)df$$

If we set $\tau = 0$ we get

$$\overline{x^2(t)} = \int_{0}^{\infty} S_x(f)df$$

which means we can easily calculate the average power if we know the power spectral density. As we will see later it is common to express noise sources in PSD form.

Another very useful theorem when working with noise in the frequency domain is this

$$S_y(f) = S_x(f)|H(f)|^2$$

, where $S_y(f)$ is the output power spectral density, $S_x(f)$ is the input power spectral density and $H(f)$ is the transfer function of a time-invariant linear system.

If we insert ([eq:psd_hf]) into ([eq:ms_psd]), with $S_x(f) = a\ constant = D_v$ we get

$$\overline{x^2(t)} = \int S_y(f)df = D_v \int |H(f)|^2 df = D_v f_x$$

, where f_x is what we call the noise bandwidth. For a single time constant RC network the noise bandwidth is equal to

$$f_x = \frac{\pi f_0}{2} = \frac{1}{4RC}$$

where f_x is the noise bandwidth and f_0 is the 3dB frequency.

We haven't told you this yet, but thermal noise is white and white means that the power spectral density is flat (constant over all frequencies). If $S_x(f)$ is our thermal noise source and $H(f)$ is a standard low pass filter, then equation ([eq:psd_hf]) tells us that the output spectral density will be shaped by $H(f)$. At frequencies above the f_x in $H(f)$ we expect the root power spectral density to fall by 20dB per decade.

9.5 Probability Distribution

Theorem 1 (Central limit theorem). *The sum of n independent random variables subjected to the same distribution will always approach a normal distribution curve as n increases.*

This is a neat theorem, it explains why many noise sources we encounter in the real world are white.* Take thermal noise for example, it is generated by random motion of carriers in materials. If we look at a single electron moving through the material the probability distribution might not be Gaussian. But summing probability distribution of the random movments with a large number of electrons will give us a Gaussian distribution, thus thermal noise is white.

* Gaussian distribution = normal distribution. Noise sources with Gaussian distribution are called white

9.6 PSD of a white noise source

If we have a true random process with Gaussian distribution we know that the autocorrelation function only has a value for $\tau = 0$. From equation ([eq:autocor]) we have that

$$\begin{aligned} R_x(\tau) = & \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x(t)x(t-\tau)dt \\ = & \left[\lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x^2(t)dt \right] \delta(\tau) \\ = & \overline{x^2(t)}\delta(\tau) \end{aligned}$$

The reason being that in a true random process $x(t)$ is uncorrelated with $x(t+\tau)$ where τ is an integer. If we use equation ([eq:psd]) we see that

$$\begin{aligned} S_x(f) = & 2\int_{-\infty}^{\infty} \overline{x^2(t)}\delta(\tau)e^{-j2\pi f\tau}d\tau \\ = & 2\overline{x^2(t)} \int_{-\infty}^{\infty} \delta(\tau)e^{-j2\pi f\tau}d\tau \\ = & 2\overline{x^2(t)} \end{aligned}$$

, since

$$\int \delta(\tau)e^{-j2\pi f\tau}d\tau = e^0 = 1$$

This means that the power spectral density of a white noise source is flat, or in other words, the same for all frequencies.

9.7 Summing noise sources

Summing noise sources is usually trivial, but we need to know why and when it is not. We if we write the time dependant noise signals as

$$v_{tot}^2(t) = (v_1(t) + v_2(t))^2 = v_1^2(t) + 2v_1(t)v_2(t) + v_2^2(t)$$

The average power is defined as

$$\begin{aligned}
\overline{e_{tot}^2} = & \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{+T/2} v_{tot}^2(t)dt \\
= & \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{+T/2} v_1^2(t)dt \\
+ & \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{+T/2} v_2^2(t)dt \\
+ & \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{+T/2} 2v_1(t)v_2(t)dt \\
= & \overline{e_1^2} + \overline{e_2^2} + \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{+T/2} 2v_1(t)v_2(t)dt
\end{aligned}$$

If $\overline{e_1^2}$ and $\overline{e_2^2}$ are uncorrelated noise sources we can skip the last term in ([eq:noisesum]) and just write

$$\overline{e_{tot}^2} = \overline{e_1^2} + \overline{e_2^2}$$

Most natural noise sources are uncorrelated.

9.8 Signal to Noise Ratios

Signal to Noise Ratio (SNR) is a common method to specify the relation between signal power and noise power in linear systems. It is defined as

$$\begin{aligned}
SNR = & \quad 10\log\left(\frac{Signal\ power}{Noise\ power}\right) \\
= & \quad 10\log\left(\frac{\overline{v_{sig}^2}}{\overline{e_n^2}}\right) \\
= & \quad 20\log\left(\frac{v_{rms}}{\sqrt{\overline{e_n^2}}}\right)
\end{aligned}$$

Another useful ratio is Signal to Noise and Distortion (SNDR), since most real systems exibit non-linearities it is useful to include distortion in the ratio. One can calculate SNR and SNDR in many ways. If we don't know the expression for $\overline{e_n^2}$ we can do a FFT of our output signal. From this FFT we sum spectral components except at the signal frequency to get noise and distortion. SNR is normally calculated as

$$SNR = 10\log\left(\frac{Signal\ power}{Noise\ power\ -\ 6\ first\ harmonics}\right)$$

And SNDR is calculated as

$$SNDR = 10\log\left(\frac{Signal\ power}{Noise\ power}\right)$$

9.9 Noise figure and Friis formula

Noise factor is a measure on the noise performance of a system. It is defined as

$$F = \frac{\overline{v_o^2}}{source\ contribution\ to\ \overline{v_o^2}}$$

where $\overline{v_o^2}$ is the total output noise.

The noise figure is defined as (noise factor in dB)

$$NF = 10\log(F)$$

The noise factor can also be defined as

$$F = \frac{SNR_{input}}{SNR_{output}}$$

This brings us right into what is known as Friis formula. If we have a multistage system, for example several amplifiers in cascade, the total noise figure of the system is defined as

$$F = 1 + F_1 - 1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + ...$$

Here F_i is the noise figures of the individual stages and G_i is the available gain of each stage. This can be rewritten as

$$F = F_1 + \sum_{i=1}^{N} \frac{F_{i+1} - 1}{\prod_{k=1}^{i-1} G_i}$$

Friiss formula tells us that it is the noise in the first stage that is the most important if G_1 is large. We could say that in a system it is important to amplify the noise as early as possible!

9.10 Spectral Density

Warning: This is not an introduction to spectral density. If the subject is completely unfamiliar I'd advise reading another source. For example chapter 4 in 1 or chapter 7 in 3.

9.10.1 Definition of Spectral Density

There are two different definitions of spectral density used in the literature. They differ by a factor of two. The one used in signal processing books, like 4, is

$$S_{x1}(f) = \int_{-\infty}^{\infty} R_{x1}(\tau) e^{-j\omega\tau} d\tau$$

And the one often used in books about noise, like 2, is

$$S_{x2}(f) = 2\int_{-\infty}^{\infty} R_{x2}(\tau) e^{-j\omega\tau} d\tau$$

In both cases $R_{xi}(\tau)$ is the auto-correlation function defined as

$$R_{xi}(\tau) = \overline{x_i(t)x_i(t+\tau)}$$

As we can plainly see

$$S_{x1}(f) \neq S_{x2}(f)$$

, there is no way these two can be made equal if

$$R_{x1}(\tau) = R_{x2}(\tau)$$

This is ok, there is no problem having two different definitions for two different functions. In reality $S_{x1}(f)$ and $S_{x2}(f)$ are different functions of frequency, and we could say that

$$S_{x2}(f) = 2S_{x1}(f)$$

if ([eq:rxequal]) is true.

9.10.2 Sources of Confusion

The problem with spectral density arises when reading literature from different communities, for example 4 and 2 where $S_x(f)$ is used for both $S_{x1}(f)$ and $S_{x2}(f)$. When I started investigating spectral densities this lead me to believe that different sources defined the same measure "spectral density" in two different ways. The more sources I investigated the more unsure I was about which of the two definitions that was correct. After months of searching (not actively, but sporadicly) I eventually found the original source of the definition of spectral density 5. Having the original source helped, but I still don't know when the original definition split into ([eq:psd1]) and ([eq:psd2]). However, I'm pretty sure the it's just a matter of convenience. To see why ([eq:psd2]) is the most common among sources concerning noise we look at the inverse Fourier Transform. By the way, if you had not noticed yet, ([eq:psd1]) says

that *Spectral density is the Fourier Transform of the Auto-Correlation function*. The inverse Fourier Transform of ([eq:psd1]) is

$$R_{x1}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{x1}(f)e^{j\omega\tau}dw = \int_{-\infty}^{\infty} S_{x1}(f)e^{j\omega\tau}df$$

,since $dw = df\,dw/df = 2\pi df$. And for ([eq:psd2])

$$R_{x2}(\tau) = \frac{1}{2}\int_{-\infty}^{\infty} S_{x2}(f)e^{jw\tau}df$$

Before we proceed lets get rid of the e's. We know that $e^{j\alpha} = \cos\alpha + j\sin\alpha$. So we could rewrite ([eq:psd1]) as

$$S_{x1}(f) = \int_{-\infty}^{\infty} R_{x1}(\tau)[\cos(\omega\tau) + j\sin(\omega\tau)]d\tau$$

and it turns out that since $R_{x1}(\tau)$ is an even function we can drop the $j\sin\omega\tau$ term. $S_{x1}(f)$ is also an even function since the Fourier Transform of an even function is even.

The definitions then become

$$\begin{aligned} S_{x1}(f) &= \int_{-\infty}^{\infty} R_{x1}(\tau)\cos(\omega\tau)d\tau \\ R_{x1}(\tau) &= \int_{-\infty}^{\infty} S_{x1}(f)\cos(\omega\tau)df \end{aligned}$$

and

$$\begin{aligned} S_{x2}(f) &= 2\int_{-\infty}^{\infty} R_{x2}(\tau)\cos(\omega\tau)d\tau \\ R_{x2}(\tau) &= \frac{1}{2}\int_{-\infty}^{\infty} S_{x2}(f)\cos(\omega\tau)df \end{aligned}$$

We can rewrite $R_{x2}(\tau)$ as

$$R_{x2}(\tau) = \overline{x_2(t)x_2(t+\tau)} = \int_0^{\infty} S_{x2}(f)\cos(\omega\tau)df$$

and if $\tau = 0$

$$\overline{x_2^2(t)} = \int_0^{\infty} S_{x2}(f)df$$

So using spectral density definition ([eq:psd2]) we see that average power (mean square value of $x_2(t)$) is equal to the integral from 0 to infinity of the spectral density. If we use ([eq:psd1]) average power would be

$$\overline{x_1^2(t)} = 2\int_0^\infty S_{x1}(f)df$$

But if $R_{x1}(\tau) = R_{x2}(\tau)$ then

$$\overline{x_2^2(t)} = \overline{x_1^2(t)}$$

even though $S_{x1}(f) \neq S_{x2}(f)$.

Definition ([eq:psd1]) is called the two-sided spectral density and ([eq:psd2]) is called the one-sided spectral density.

9.10.3 Example: Thermal Noise

The spectral density of thermal noise in electronic circuit should be known to anyone that has studied analog electronics. We normally define the voltage spectral density of thermal noise as

$$S_{th}(f) = 4kTR$$

where k is Boltzmann's constant, T the temperature in Kelvin and R the resistance. But ([eq:othermal]) is the spectral density when it is defined as in ([eq:psd2]). If we were to use ([eq:psd1]) then the spectral density of thermal noise would be

$$S_{th}(f) = 2kTR$$

Both these spectral densities would give the same average power value if we use the inverse Fourier Transform of ([eq:psd1]) and ([eq:psd2]).[†]

9.10.4 Einstein: The source

In his 1914 paper 5 Albert Einstein described, supposedly for the first time, the auto-correlation function and what we have come to know as the spectral density. He defined the auto-correlation function as

$$\mathfrak{M}(\Delta) = \overline{F(t)F(t+\Delta)}$$

and the intensity (spectral density) as

$$I(\theta) = \int_0^T \mathfrak{M}(\Delta)\cos(\pi\frac{\Delta}{\theta})d\Delta$$

[†] Note that if you calculate the average power of $S_{th}(f)$ you'll get infinity. You have to include the bandwidth of the circuit you are considering for average power to have a finite value.

,where the period $\theta = T/n$ and T is a very large value. The paper is very short, only 1 page, but it is worth reading. Note that ([eq:psd1]) is often referred to as the *Wiener-Khintchine* theorem.

The Tools 10

Status: 1.0

10.1 Tools

I would strongly recommend that you install all tools locally on your system. There is a video that describe the install procedure. It's a few years old, but should still be able to guide you https://youtu.be/DRppsdjo2Rc?si=x8cJsa1lpncvSFmu.

For the analog toolchain we need some tools, and a process design kit (PDK).

- Skywater 130nm PDK. I use open_pdks to install the PDK
- Magic VLSI for layout (Version 8.3 revision 541)
- ngspice for simulation (version 45.2)
- netgen for LVS (1.5.295)
- xschem (3.4.8RC)
- verilator (5.034)
- python > 3.10

The tools are not that big, but the PDK is huge, so you need to have about 50 GB disk space available.

10.1.1 Setup WSL (Applicable for Windows users)

Install a Linux distribution such as Ubuntu 24.04 LTS by running the following command in PowerShell on Windows and follow the instructions.

```
wsl --install -d Ubuntu-24.04
```

When you have installed the Linux distribution and signed into it, install make

```
sudo apt install make
```

10.1.2 Setup public key towards github

Do

```
ssh-keygen -t rsa
```

And press "enter" on most things, or if you're paranoid, add a passphrase

Then

```
cat ~/.ssh/id_rsa.pub
```

And add the public key to your github account. Settings - SSH and GPG keys

10.1.3 Provide git with author identity

There are interactions with git that require an author identity. You are supposed to use one of these interactions a lot during the project, namely, `git commit`. What you need to provide is an email address and a name. If you would like to keep your real email address private/secret, read what it says on GitHub at your user settings page under emails. Use the below commands to provide the author identity information to git.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

10.1.4 Get AICEX and setup your shell

You don't have to put aicex in `$HOME/pro`, but if you don't know where to put it, chose that directory.

```
cd
mkdir pro
cd pro
git clone --recursive https://github.com/wulffern/aicex.git
```

You need to add the following to your `~/.bashrc` (note that ~ refers to your home directory `$HOME/.bashrc` also works, or `$HOME/.bash_profile` on some newer macs)

```
export PDK_ROOT=/opt/pdk/share/pdk
export LD_LIBRARY_PATH=/opt/eda/lib
export PATH=/opt/eda/bin:$HOME/.local/bin:$PATH
```

10.1.5 On systems with python3 > 3.12

On newer systems it's not trivial to install python packages because python is externally managed. As such, we need to install a python environment.

```
#- Find a package similar to name below
sudo apt-get update
sudo apt install python3.12-venv
sudo mkdir /opt
sudo mkdir /opt/eda
sudo mkdir /opt/eda/python3
sudo chown -R $USER:$USER /opt/eda/python3/
python3 -m venv /opt/eda/python3
```

Modify the `~/.bashrc` to include the python environment

```
export PATH=/opt/eda/bin:/opt/eda/python3/bin:$HOME/.local/bin:$PATH
```

10.1.6 Install Tools

Make sure you load the settings before you proceed

```
source ~/.bashrc
```

Hopefully the commands below work, if not, then try again, or try to understand what fails. There is no point in continuing if one command fails.

```
cd aicex/tests/
make requirements
make tt
```

On a mac, you probably need to add bison to the path

```
export PATH="/opt/homebrew/opt/bison/bin:$PATH"
```

I've split the install of each of the tools. It's possible to run the commented out lines instead, but they often fail

```
#make eda_compile
#sudo make eda_install
make magic_compile  magic_install
make netgen_compile netgen_install
make xschem_compile xschem_install
make iverilog_compile iverilog_install
make ngspice_compile # Sometimes fails
make ngspice_compile ngspice_install
```

On Mac, do

```
brew install yosys verilator
```

On Linux, do

```
make yosys_compile yosys_install
```

On all, do

```
python3 -m ensurepip --default-pip

python3 -m pip install matplotlib numpy click svgwrite \
    pyyaml pandas tabulate wheel setuptools tikzplotlib
source install_open_pdk.sh
```

10.1.7 Install cicconf

cIcConf is used for configuration. How the IPs are connected, and what version of IPs to get.

```
cd
cd pro/aicex/ip/cicconf
git checkout main
git pull
python3 -m pip install -e .
cd ../
```

Update IPs

```
cicconf clone --https
cd ../..
```

10.1.8 Install cicsim

cIcSim is used for simulation orchestration.

```
cd aicex/ip/cicsim
git checkout main
git pull
python3 -m pip install -e .
cd ../..
```

10.1.9 Install cicpy

CicPy is used to generate layout

```
cd aicex/ip/cicpy
git checkout master
git pull
python3 -m pip install -e .
cd ../..
cd aicex/ip/cicspi
git checkout main
git pull
python3 -m pip install -e .
cd ../..
```

10.1.10 Setup your ngspice settings

Edit `~/.spiceinit` and add

```
set ngbehavior=hsa     ; set compatibility for PDK libs
set ng_nomodcheck      ; don't check the model parameters
set num_threads=8      ; CPU hardware threads available
set skywaterpdk
option noinit          ; don't print operating point data
option klu
optran 0 0 0 100p 2n 0 ; don't use dc operating point,
option opts
```

10.2 Check that magic and xschem works

To check that magic and xschem works

```
cd ~/pro/aicex/ip/sun_sar9b_sky130nm/work
magic ../design/SUN_SAR9B_SKY130NM/SUNSAR_SAR9B_CV.mag &
xschem -b ../design/SUN_SAR9B_SKY130NM/SUNSAR_SAR9B_CV.sch &
```

Sky130nm tutorial 11

Status: 0.5

If the commands don't work, then you have not installed the tools. Check The Tools chapter first.

11.0.1 Create the IP

I've made some scripts to automatically generate the IP.

To see what files are generated, see `tech_sky130A/cicconf/lelo.yaml`

```
cd aicex/ip
cicconf newip ex --project lelo --technology sky130A --ip  tech_sky130A/cicconf/lelo.yaml
```

11.0.2 The file structure

It matters how you name files, and store files. I would be surprised if you had a good method already, as such, I won't allow you to make your own folder structure and names for things. I also control the filenames and folder structure because there are many scripts to make your life easier (yes, really) that rely on an exact structure. Don't mess with it.

11.0.2.1 Github workflows

On github it's possible use something called workflows to run things every time you push a new version. It's really nice, since it can then check that your design is valid.

The workflows are defined below.

```
.github
   workflows
   docs.yaml # Generate a github page
   drc.yaml  # Run Design Rule Checks
   gds.yaml  # Generate a GDS file from layout
   lvs.yaml  # Run Layout Versus Schematic
             # and Layout Parasitic Extraction
```

11.0.2.2 Configuration files

Each IP has a few files that define the setup, you'll need to modify at least the `README.md` and the `info.yaml`.

```
.gitignore  # files that are ignored by git
README.md   # Frontpage documentation
config.yaml # What libraries are used. Used by  cicconf
info.yaml   # Setup names, authors etc
media       # Where you should store images for documentation
tech -> ../tech_sky130A  # The technology library
```

11.0.2.3 Design files

A "cell" in the open source EDA world should consists of the following files

- Schematic (.sch)
- Layout (.mag)
- Documenation (.md)

The files must have the same name, and must be stored in `design/<LIB>/` as shown below.

Note there are also two symbolic links to other libraries. These two libraries contain standard cells and standard analog transistors (ATR) that you should be using.

```
design
  LELO_EX_SKY130A
  LELO_EX.sch
  JNW_ATR_SKY130A -> ../../jnw_atr_sky130a/design/JNW_ATR_SKY130A
  JWN_TR_SKY130A -> ../../jnw_tr_sky130a/design/JNW_TR_SKY130A
```

For example, if the cell name was `LELO_EX`, then you would have

- `design/LELO_EX_SKY130A/LELO_EX.sch`: Schematic (xschem)
- `design/LELO_EX_SKY130A/LELO_EX.sym`: Symbol (xschem)
- `design/LELO_EX_SKY130A/LELO_EX.mag`: Layout (Magic)
- `design/LELO_EX_SKY130A/LELO_EX.md` : Markdown documentation (any text editor)

All these files are text files, so you can edit them in a text editor, but mostly you shouldn't (except for the Markdown)

11.0.2.4 Simulations

All simulations shall be stored in `sim`. Once you have a Schematic ready for simulation, then

```
cd sim
make cell CELL=LELO_EX
```

This will make a simulation folder for you. Repeat for all your cells.

```
sim
  Makefile
  cicsim.yaml -> ../tech/cicsim/cicsim.yaml
```

11.0.2.5 The work

All commands (except for simulation), shall be run in the `work` folder.

In the `work/` folder there are startup files for Xschem (xschemrc) and Magic (.magicrc). They tell the tools where to find the process design kit, symbols, etc. At some point you probably need to learn those also, but I'd wait until you feel a bit more comfortable.

```
work
 .magicrc
 Makefile
 mos.24bit.dstyle -> ../tech/magic/mos.24bit.dstyle
 mos.24bit.std.cmap -> ../tech/magic/mos.24bit.std.cmap
 xschemrc
```

11.0.3 Github setup

Create a repository on github. The name of the repository that you make on GitHub has to be the same as what is written after `<your username>` in the last command below. In this example, that is `lelo_ex_sky130a`.

```
cd lelo_ex_sky130a
git remote add origin \
 git@github.com:<your username>/lelo_ex_sky130a.git
```

11.0.4 Start working

11.0.4.1 Edit README.md

Open README.md in your favorite text editor and make necessary changes.

11.0.4.2 Familiarize yourself with the Makefile and make

I write all commands I do into a Makefile. There is nothing special with a Makefile, it's just what I choose to use 20 years ago. I'm not sure I'd choose something different now.

```
cd work
make
```

Take a look inside the file called Makefile.

11.0.5 Draw Schematic

The block we'll make is a current mirror with a 1 to 4 scaling.

A schematic is how we describe the connectivity, and the types of devices in an analog circuit. The open source schematic editor we will use is XSchem.

Open the schematic:

```
xschem -b ../design/LELO_EX_SKY130A/LELO_EX.sch &
```

11.0.5.1 Add Ports

Add IBPS_5U and IBNS_20U ports, the P and N in the name signifies what transistor the current comes from. So IBPS must go into a diode connected NMOS, and N will be our output, and go into a diode connected PMOS somewhere else.

11.0.5.2 Add transistors

Use 'I' or 'Shift+i' (note the letter case) to open the library manager. Click the `lelo_ex_sky130A/design` path, then `JNW_ATR_SKY130A` and select `JNWATR_NCH_4C5F0.sym`

The naming convention for these transistors is `<number of contacts on drain/source>C<times minimum gate length>F`, so the number before the C is the width, and the number before/after the F is the length. The absolute size does not matter for now. Just think "4C5F0 is a 4 contact wide long transistor", while a "4C1F2 is a 4 contact wide, short transistor".

Select the transistor and press 'c' to copy it, while dragging, press 'shift-f' to flip the transistor so our current mirror looks nice. 'shift-r' rotates the transistor, but we don't want that now.

Place two transistors for the output transistor, as shown in the figure below.

Press ESC to deselect everything

Select the input transistor, and change the name to 'xo1'

Select the first output transistor, and change the name to 'xo0[1:0]'. Using bus notation on the name will create 2 transistors.

Select the second output transistor and give it the name 'xo1[1:0]'.

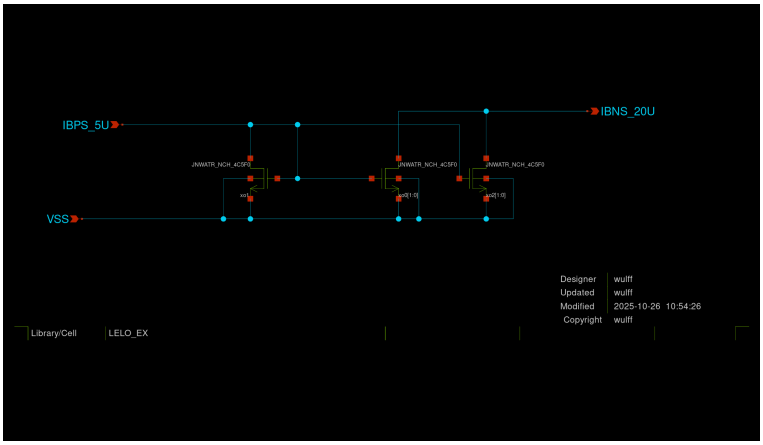
Select ports, and use 'm' to move the ports close to the transistors.

Press 'w' to route wires.

Use 'shift-z' and z, to zoom in and out

Use 'f' to zoom full screen

Remember to save the schematic



11.0.5.3 Netlist schematic

Check that the netlist looks OK

In work/

```
make xsch CELL=LELO_EX
cat xsch/LELO_EX.spice
```

11.0.6 Typical corner SPICE simulation

I've made cicsim that I use to run simulations (ngspice) and extract results

11.0.6.1 Setup simulation environment

Navigate to the `lelo_ex_sky130a/sim/` directory.

Make a new simulation folder

```
cicsim simcell  LELO_EX_SKY130A LELO_EX \
    ../tech/cicsim/cell_spice/template.yaml
```

I would recommend you have a look at simcell_template.yaml file to understand what happens.

11.0.6.2 Familiarize yourself with the simulation folder

I've added quite a few options to cicsim, and it might be confusing. For reference, these are what the files are used for

File	Description
Makefile	Simulation commands
cicsim.yaml	Setup for cicsim
summary.yaml	Generate a README with simulation results
tran.meas	Measurement to be done after simulation
tran.py	Optional python script to run for each simulation

File	Description
tran.spi	Transient testbench
tran.yaml	What measurements to summarize

The default setup should run, so

```
cd LELO_EX
make typical
```

11.0.6.3 Modify default testbench (tran.spi)

Delete the VDD source

Add a current source of 5uA, and a voltage source of 1V to IBNS_-20U

```
IBP 0 IBPS_5U dc 5u
V0  IBNS_20U 0 dc 1
```

Save the current in V0 by adding i(V0) to the save statement in the testbench

Save the voltage by adding v(IBPS_5U) to the save statement

```
.save i(V0) v(IBPS_5U)
```

11.0.6.4 Modify measurements (tran.meas)

Add measurement of the current and VGS. It must be added between the "MEAS_START" and "MEAS_END" lines.

```
let ibn = -i(v0)
meas tran ibns_20u find ibn at=5n
meas tran vgs_m1 find v(ibps_5u) at=5n
```

Run simulation

```
make typical
```

and check that the output looks okish.

Try to run the simulation again

```
make typical
```

If everything works, then the simulation now should **not** be run. Every time cicsim runs (provided the `sha: True` option is set in `cicsim.yaml`) cicsim will compute a SHA hash of all files (stored in output_tran/*.sha*) that is referenced in the `tran.spi`. Next time cicsim is run, it checks the hash's and does not re-run if there is no need (no files changed).

Sometimes you want to force running, and you can do that by

```
make typical OPT="--no-sha"
```

Often, it's the measurement that I get wrong, so instead of rerunning simulation every time I've added a "–no-run" option to cicsim. For example

```
make typical OPT="--no-run"
```

will skip the simulation, and rerun only the measurement. This is why you should split the testbench and the measurement. Simulations can run for days, but measurement takes seconds.

11.0.6.5 Modify result specification (tran.yaml)

Add the result specifications, for example

```
ibn:
  src:
    - ibns_20u
  name: Output current
  min: -5%
  typ: 20
  max: 5%
  scale: 1e6
  digits: 3
  unit: uA

vgs:
  src:
    - vgs_m1
  name: Gate-Source voltage
  typ: 0.6
  min: 0.3
  max: 0.8
  scale: 1
  digits: 3
  unit: V
```

Re-run the measurement and result generation

```
make typical OPT="--no-run"
```

Open `results/tran_Sch_typical.html`

11.0.6.6 Check waveforms

You can either use ngspice, or you can use cicsim, or you can use something I don't know about

Open the raw file with

```
cicsim wave output_tran/tran_SchGtKttTtVt.raw
```

Load the results, and try to look at the plots. There might not be that much interesting happening

Searching waveforms On the left side of the window you'll see a text box in the middle between the filename, and the wave names. This is a regex search field, and you can easily search for waveforms (like `i(v0)`) that you want to find.

Note that the search field uses regular expressions. If you don't know regex, then it's time to learn. I always use the perl regular expression variants.

For example, searching for "i(v0)" won't acctually show anything, because the `()` are special characters. "i(v0)" will find it though.

I could search for both ibps and v0 at the same time with `ibps|i\(`, so it's well worth learning.

A great resource is Mastering Regular Expressions

11.0.7 All corners SPICE simulations

Analog circuits must be simulated for all physical conditions, we call them corners. We must check high and low temperature, high and low voltage, all process corners, and device-to-device mismatch.

11.0.7.1 Remove Vh and Vl corners (Makefile)

For the current mirror we don't need to vary voltage, since we don't have a VDD.

Open Makefile in your favorite text editor.

Change all instances of "Vt,Vl,Vh" and "Vl,Vh" to Vt

11.0.7.2 Run all corners

To simulate all corners do

```
make typical etc mc
```

where etc is extreme test condition and mc is monte-carlo.

Wait for simulations to complete.

11.0.7.3 Get creative with python

Open `tran.py` in your favorite editor, try to read and understand it.

The `name` parameter is the corner currently running, for example `tran_SchGtAmcttTtVt`.

The measured outputs from ngspice will be added to `tran_SchGtAmcttTtVt.yaml`

Delete the "return" line.

Add the following lines (they automatically plot the current and gate voltage)

```
import cicsim as cs
fname = name +".png"
print(f"Saving {fname}")
cs.rawplot(name + ".raw","time","v(ibps_5u),i(v0)" \
  ,ptype="",fname=fname)
```

Re-run measurements to check the python code

```
make typical etc mc OPT="--no-run"
```

You'll see that cicsim writes all the png's. Check with `ls -l output_tran/*.png`.

You'll also notice it will slow down the simulation, so maybe remove the lines from `tran.py` again ;-)

11.0.7.4 Generate simulation summary

Run

```
make summary
```

Install pandoc if you don't have it

Run

```
pandoc -s   README.md -o README.html
```

to generate a HTML slideshow that you can open in browser. Open the HTML file.

11.0.7.5 Viewing results without GUI browser

If your on a system without a browser, or indeed a GUI, then it's possible to view the results in the terminal.

Check if `lynx` is installed, if it's not installed, then

On linux

```
sudo apt-get install lynx
```

On Mac

```
brew install lynx
```

Then

```
lynx README.html
```

11.0.7.6 Think about the results

From the corner and mismatch simulation, we can observe a few things.

- The typical value is not 20 uA. This is likely because we have a M2 VDS of 1 V, which is not the same as the VDS of M1. As such, the current will not be the same.
- The statistics from 30 corners show that when we add or subtract 3 standard deviation from the mean, the resulting current is outside our specification of +- 5 %.

11.0.8 Draw Layout

A foundry (the factory that makes integrated circuits) needs to know how we want them to create our circuit. So we need to provide them with a "layout", the recipe, or instruction, for how to make the circuit. Although the layout contains the same components as the schematic, the layout contains the physical locations, and how to actually instruct the foundry on how to make the transistors we want.

Open Magic VLSI

```
cd work
magic ../design/LEL0_EX_SKY130A/LEL0_EX.mag
```

Now brace yourself, Magic VLSI was created in the 1980's. For it's time it was extremely modern, however, today it seems dated. However, it is free, so we use it.

11.0.8.1 Magic VLSI

Try google for most questions, and there are youtube videos that give an intro.

- Magic Tutorial 1
- Magic Tutorial 2
- Magic Tutorial 3
- Magic command reference
- Magic Documentation

Default magic start with the BOX tool. Mouse left-click to select bottom corner, left-click to select top corner.

Press “space” to select another tool (WIRING, NETLIST, PICK).

Type “macro help” in the command window to see all shortcuts

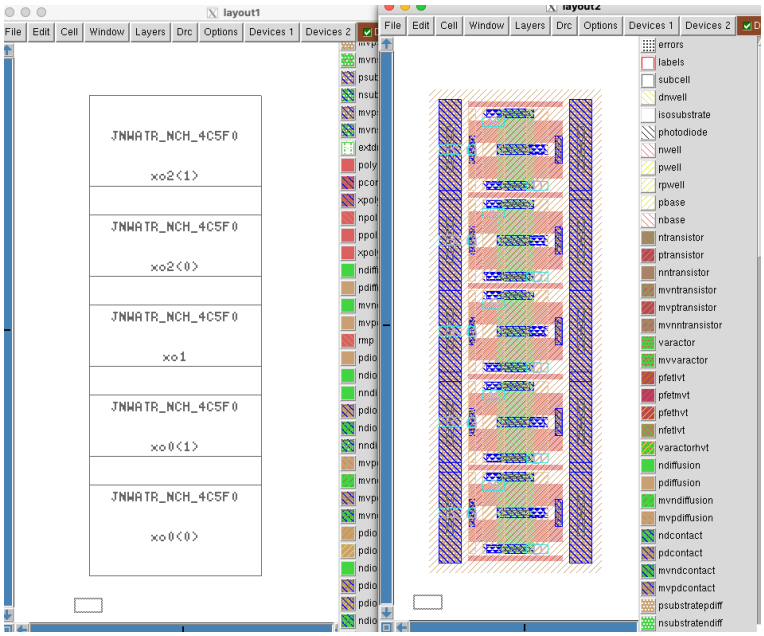
Hotkey	Function
v	View all
shift-z	zoom out
z	zoom in
x	look inside box (expand)
shift-x	don’t look inside box (unexpand)
u	undo
d	delete
s	select
Shift-Up	Move cell up
Shift-Down	Move cell down
Shift-Left	Move cell left
Shift-Right	Move cell right

11.0.8.2 Add transistors

Open Cell -> Place Instance. Navigate to the right transistor.

Place it. Hover over the transistor and select it with ‘s’. Now comes a bit of tedious thing. Select again, and copy. It’s possible to align the transistors on-top of eachother, but it’s a bit finicky.

Place all transistors on top of each other as shown below in the picture.



11.0.8.3 Place devices

You will find that one of the more time consuming things with analog layout is to place the devices, and to follow the design rules from foundry. I detest tedious work. As such, I've tried for the past 25 years to simplify analog layout. I've not finished yet, but maybe you'll find some of the scripts useful.

Note that the command below will override all your hard work ;-)

```
cd work
make xsch
cicpy sch2mag LELO_EX_SKY130A LELO_EX
```

11.0.8.4 Add Ground

In the command window, type

```
see no *
see viali
see locali
see m1
see via1
see m2
```

Make a box around the layout by left cliking bottom left, and right clicking top right. Press 'x' to expand.

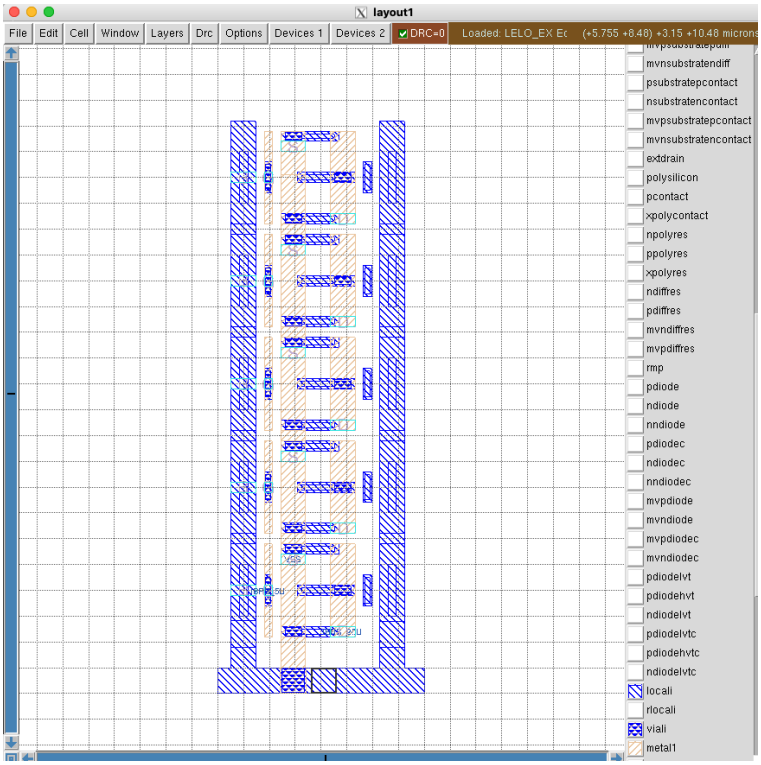
Change grid to 1 um. Set "Window->Snap to grid on"

Select a 1 um box below the transistors and paint the rectangle with locali (middle click on locali)

Change to the ‘wire tool’ with spacebar. Set “Window-> Snap to grid off”

Connect guard rings to ground.

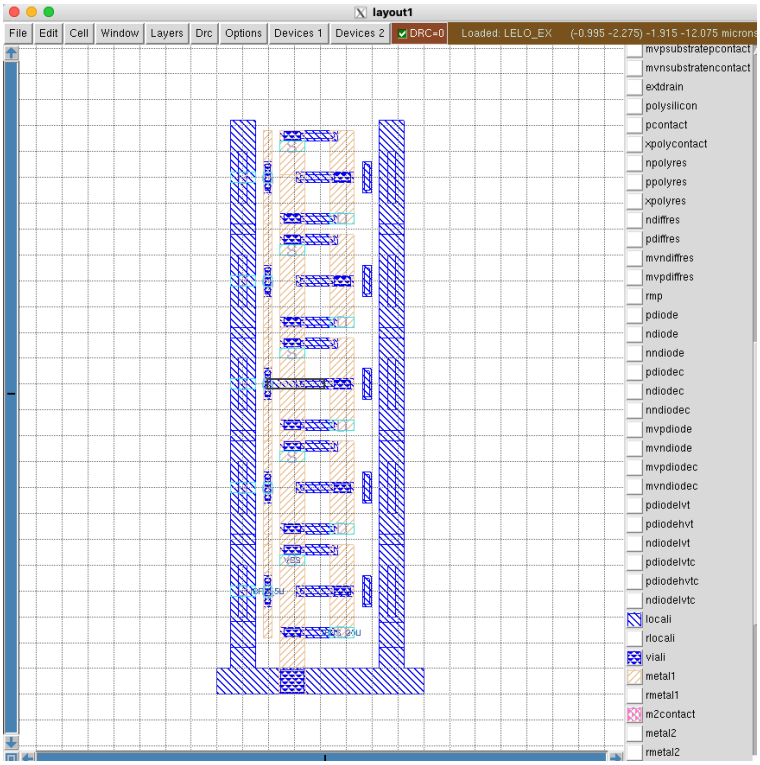
Press the top transistor ‘S’ and draw all the way down to connect all of the transistors’ source terminals. Use ‘shift-right click’ to change layer down



11.0.8.5 Route Gates

Press “space” to enter wire mode. Left click on the top gate to start a wire, and right click to end the wire.

The drain of M1 transistor needs a connection from gate to drain. We do that for the middle transistor. Change to the box tool (spacebar a few times). Create a box that matches the locali. Connect the drain to the gate in locali.

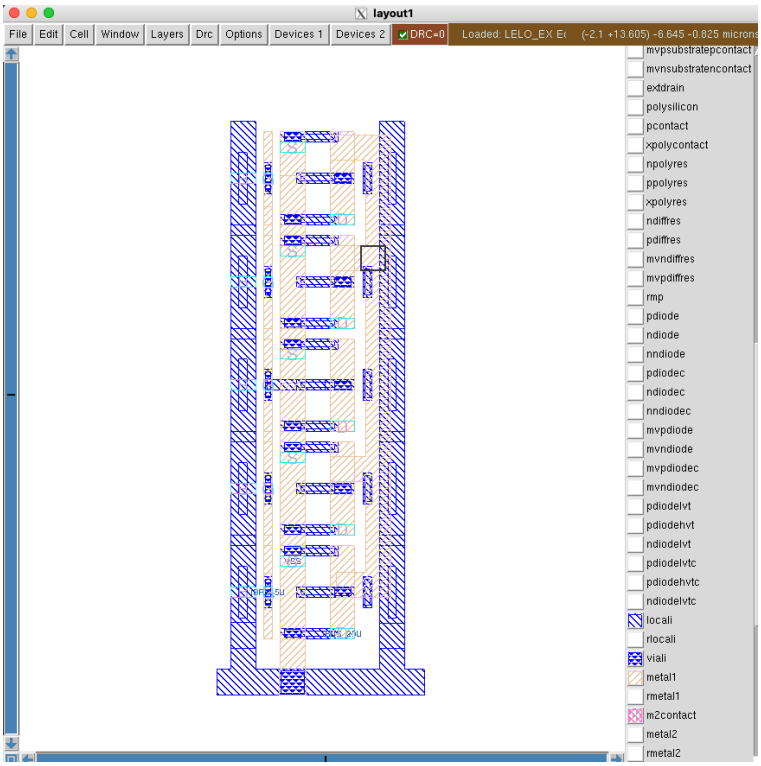


11.0.8.6 Drain of M2

Use the wire tool to draw connections for the drains.

To add vias you can do “shift-left click” to move up a metal, and “shift-right click” to go down.

It’s a very good idea to have direction rules for metal layers. I would recommend that you route metal1 vertical, metal2 horizontal, metal3 vertical etc. For locali it’s usually all over the place.



11.0.8.7 Add labels

All ports must be named (IBPS_5U, IBNS_20U, VSS). The cicpy script may add ports, but not necessarily where you want them.

Select a box on a metal, and use “Edit->Text” to add labels for the ports. Select the port button.

11.0.9 Layout verification

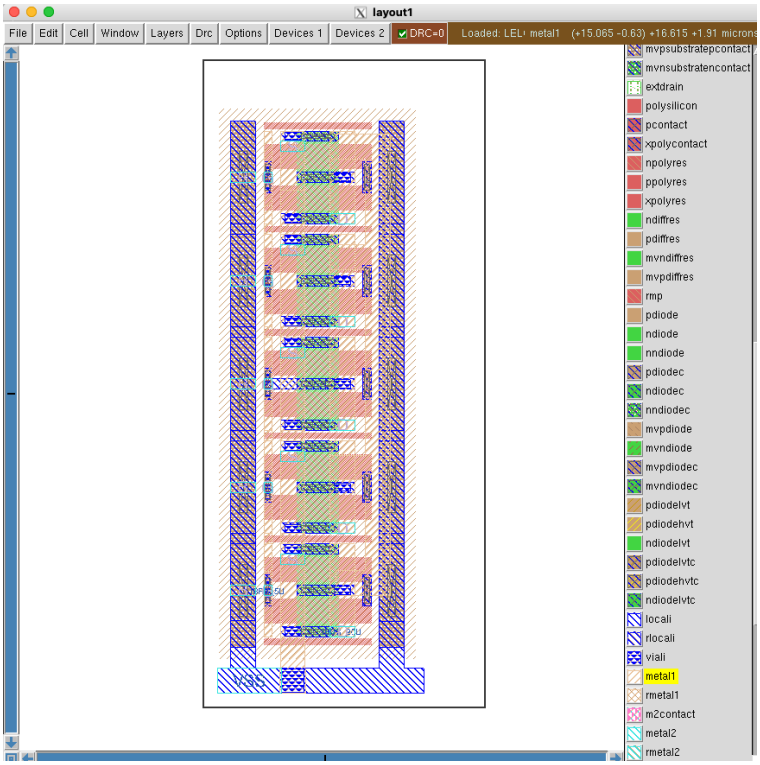
The DRC can be seen directly in Magic VLSI as you draw.

To check layout versus schematic navigate to work/ and do

```
make cdl lvs
```

Remember to save the layout first.

If you’ve routed correctly, then the LVS should be correct.



11.0.10 Extract layout parasitics

With the layout complete, we can extract parasitic capacitance.

```
make lpe
```

Check the generated netlist

```
cat lpe/LELO_EX_lpe.spi
```

11.0.11 Simulate with layout parasitics

Navigate to sim/LELO_EX. We now want to simulate the layout.

The default `tran.spi` should already have support for that.

Open the Makefile, and change

```
VIEW=Sch
```

to

```
VIEW=Lay
```

11.0.11.1 Typical simuation

Run

```
make typical
```

11.0.11.2 Corners

Navigate to sim/LELO_EX. Run all corners again

```
make all
```

11.0.11.3 Simulation summary

Open `summary.yaml` and add the layout files.

```
    - name: Lay_typ
      src: results/tran_Lay_typical
      method: typical
    - name: Lay_etc
      src: results/tran_Lay_etc
      method: minmax
    - name: Lay_3std
      src: results/tran_Lay_mc
      method: 3std
```

Run summary again

```
make summary
pandoc -s  README.md -o README.html
```

Open the README.html and have a look a the results. The layout should be close to the schematic simulation.

11.0.12 Make documentation

Make a file (or it may exists) `design/LELO_EX_SKY130A/LELO_EX.md` and add some documentation of what you've made.

Add the simulation results to your git repository to keep track

```
git add sim/LELO_EX/results/*.html
git add sim/LELO_EX/README.md
```

11.0.13 Edit info.yaml

Finally, let's setup the `info.yaml` so that all the github workflows run correctly.

Mine will look like this.

You need to setup the url (probably something like `<your username>.github.io`) to what is correct for you.

I've added the doc section such that the workflows will generate the docs.

The sim is to run a typical simulation.

```
library: LELO_EX_SKY130A
cell: LELO_EX
author: Carsten Wulff
github: wulffern
tagline: The answer is 42
email: carsten@wulff.no
url: wulffern.github.io
doc:
  libraries:
    LELO_EX_SKY130A:
      - LELO_EX
```

11.0.14 Setup github pages

Go to your GitHub repository (repo). Press Settings. Press Pages. Choose source under Build and Deployment -> GitHub Actions

Wait for the workflows to build. And check your github pages. Mine is https://wulffern.github.io/lelo_ex0_sky130a/.

11.0.15 Frequency asked questions

Q: My GDS/LVS/DRC action fails, even though it works locally.

Sometimes the reference to the transistors in the magic file might be wrong. Open the .mag file in a text editor and check. The correct way is

```
use JNWATR_NCH_4C5F0  JNWATR_NCH_4C5F0_0 ../LELO_ATR_SKY130A
```

It's the last `../JNW_ATR_SKY130A` that sometimes is missing.

Bibliography

D. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, Inc., 1997.

A. V. D. Ziel, *Noise in Solid State Devices and Circuits*. John Wiley & Sons Inc, 1986.

B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.

R. M. Gray and L. D. Davisson, *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004, no. ISBN-0521838606.

A. Einstein, "Method for the determinination of the statistical values of observations concerning quantities subject to irregular fluctuations," vol. October, 1987.