

Analog Workshop



Simon Dorrer

HeiChips 2026

About Me – Simon Dorrer

- PhD student @ JKU
- Research and teaching assistant @ IICQC
- Open-source projects:
 - AMS and RF design flows
 - mm-wave six-port receivers
 - TinyWhisper SoC
 - Event-based ADCs
 - Tutorials and templates
- Email: simon.dorrer@jku.at



Raise your hand if:

- You have synthesized your own digital ASIC
- You have designed an analog (mixed-signal) ASIC
- You have drawn a schematic in Xschem
- You have sized transistors (with the g_m/I_D method)
- You have drawn a DRC and LVS clean layout in KLayout
- You have run a post-layout simulation
- You have automated your design flow
- You had your own chip or tile manufactured



Motivation

- **Analog integrated circuit design is complex!**
 - lots of data to organize and transfer between tools
 - steep learning curve of tools and skills
 - **Mission:** Lower the entry barrier for analog mixed-signal chip design
 - Course: <https://github.com/iic-jku/analog-circuit-design>
 - Tutorials and workshops
 - Reproducible and automated design flows
 - Starting point with **Makefile-driven** and **recursive** folder structure
- **An Open-Source Analog Mixed-Signal Chip Design Template & Tutorial**

Analog Mixed-Signal Design Flow & Template



GitHub Template

- **Solderpad Hardware License v2.1**
- [Use this template](#) or `git clone ...`
- `make all`
- **Tapeout-ready AMS chip!**
- **Tutorial** based on Quarto
- **Regression tests** via GitHub actions
- **Unbelievable?**
 - Try it out now! →
- Tapeout proven?



An Open-Source Analog Mixed-Signal Chip Design Template & Tutorial for the ihp-sg13g2 Open-PDK

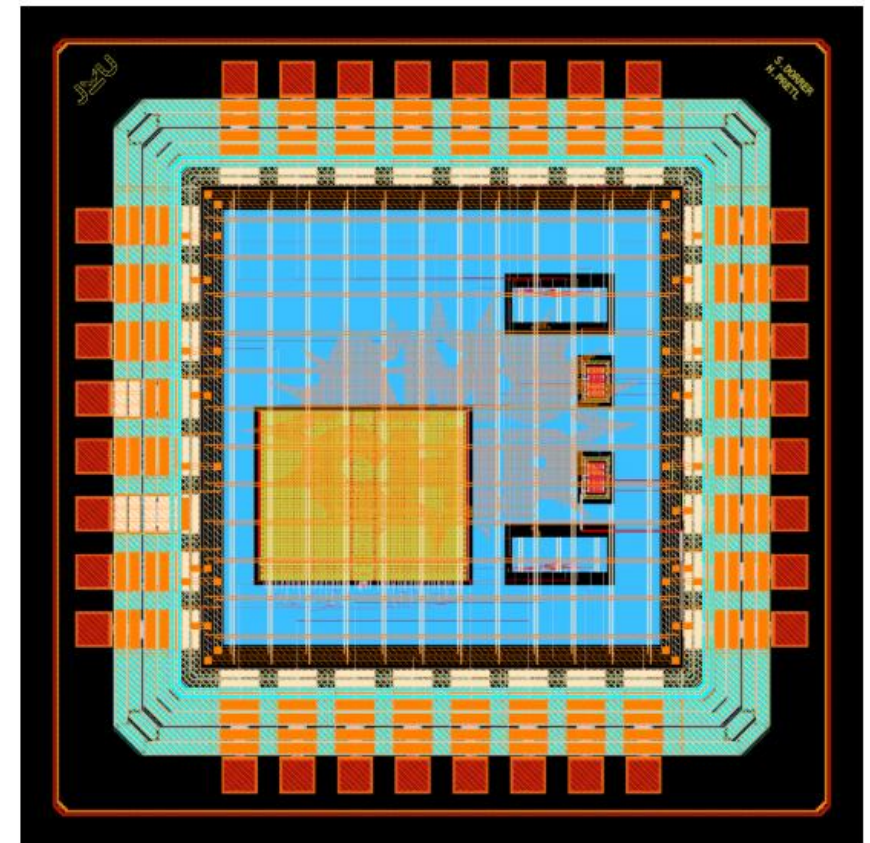
License [Solderpad Hardware License v2.1](#) [quarto-publish](#) [passing](#) [regression](#) [passing](#) [Tutorial](#) [online](#)
DOI [10.5281/zenodo.20129233](#)

(c) 2026 Simon Dorrer and Harald Pretl

Institute for Integrated Circuits and Quantum Computing, Johannes Kepler University (JKU), Linz, Austria

Important

This repository requires the [IIC-OSIC-TOOLS](#) container with tag `2026.06` or later.



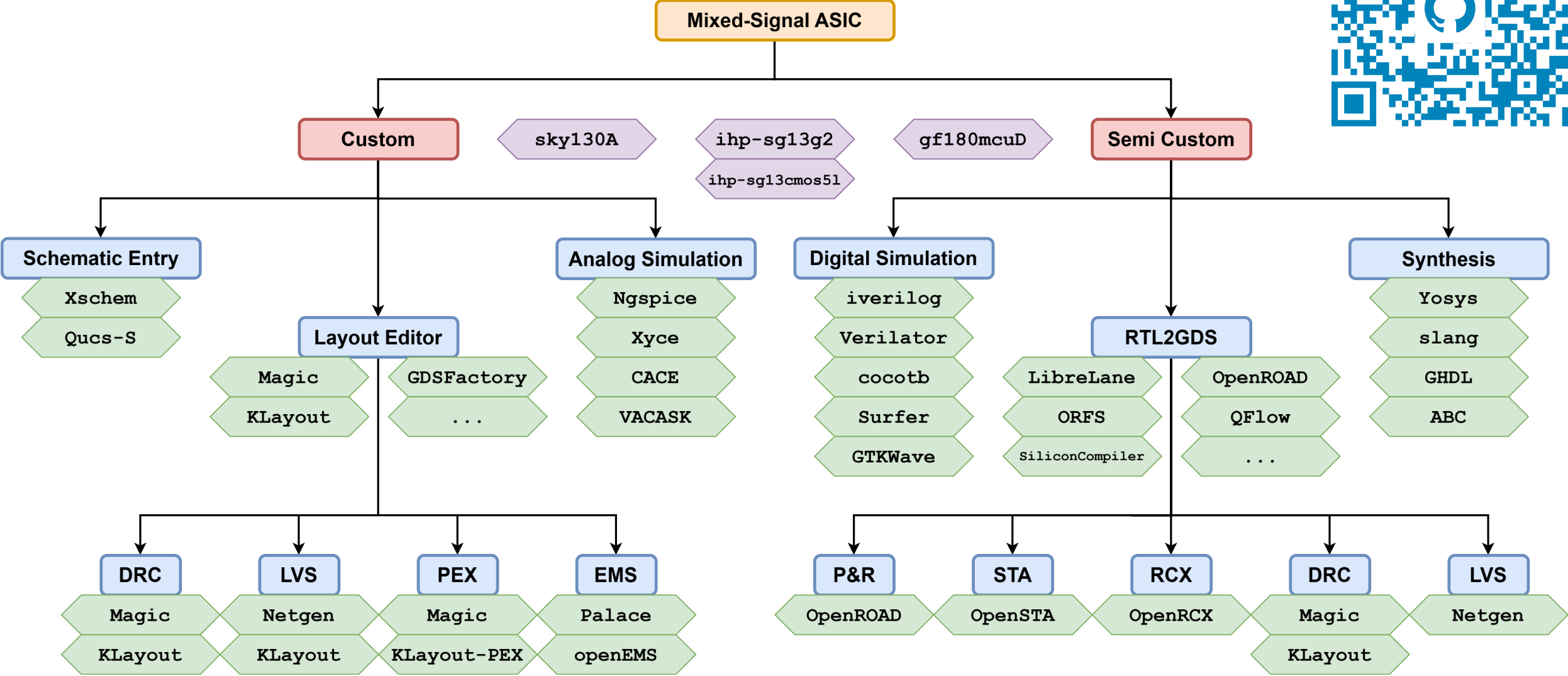
Chip render of the ihp-sg13g2 analog-mixed signal template chip (1.6 mm x 1.6 mm).

Flow Overview



- Simulation, linting, and hardening of RTL
- Layout checks and reports of hardened digital macro
- Simulation of all analog testbenches (incl. PVT and MC)
- Layout checks of analog blocks (DRC, LVS, and PEX)
- Digital-on-top top-level assembly and layout check (DRC)
 - Seal ring and pad frame generation
 - Macro placement and wiring also with non-default rules (NDRs)
 - Logo and fill insertion
- Full top-level LVS and simulation without black boxing
- Bonding diagram

IIC-OSIC-TOOLS

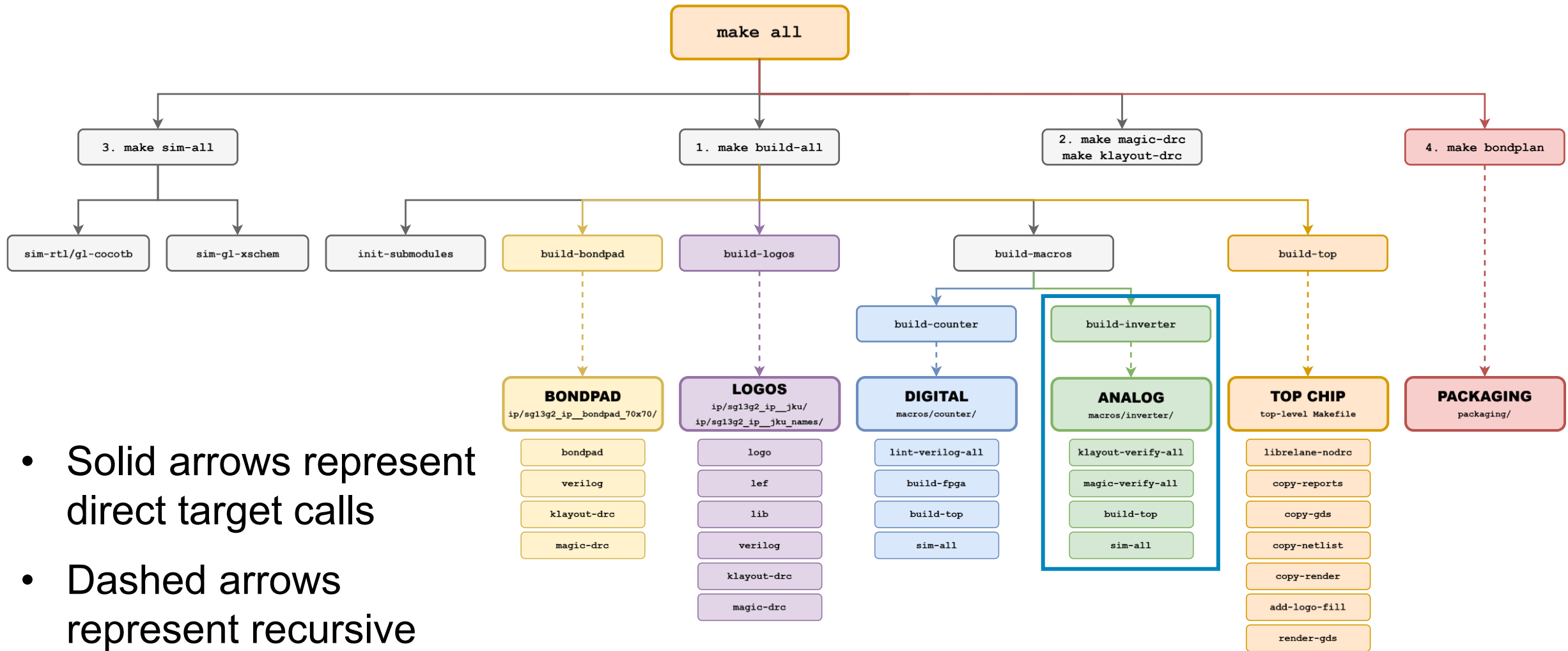


Folder Structure

- Structure discussed with FOSSi community
- Easily share and reproduce results of other macros



Makefile Structure



- Solid arrows represent direct target calls
- Dashed arrows represent recursive target calls

Tutorial

- Walk through all IC design steps to get started
- Using IHP's 130 nm CMOS process
- Simple exercises to learn by doing



Open-Source Analog-Mixed Signal Chip Design Tutorial

HomePDF

A Tutorial on Open-Source Analog-Mixed Signal Chip Design with the ihp-sg13g2 Open-PDK

AUTHORS

Simon Dorrer

Harald Pretl

PUBLISHED

July 5, 2026

Important

This tutorial and the underlying template repository require the [IIC-OSIC-TOOLS](#) container with tag [2026.06](#) or later. All commands below must be executed from within the container, and the working directory must be a folder containing the corresponding [Makefile](#).

Note

We happily accept [pull requests](#) to fix typos or add content! If you want to discuss something that is not clear, please [open an issue](#)!

1 Tutorial Overview

This tutorial is a step-by-step walk-through of the open-source analog-mixed signal (AMS) chip flow based on the [ihp-sg13g2-ams-chip-template](#) repository for the ihp-sg13g2 130nm Open-PDK. It covers:

- [Overview](#) and [Installation](#) of [IIC-OSIC-TOOLS](#)
- [Digital macro design](#)
 - Design files written in SystemVerilog
 - Linting with [Verilator](#)
 - Simulation with [Icarus Verilog](#) and [cocotb](#)
 - Viewing waveforms with [GTKWave](#) and [Surfer](#)
 - Synthesis, placement, routing, and verification with [Yosys](#), [OpenROAD](#), and [LibreLane](#)
- [Analog macro design](#)
 - Schematic entry in [Xschem](#)
 - Simulation with [Ngspice](#) and [VACASK](#)
 - Monte Carlo / mismatch characterisation with [CACE](#)
 - Layout in [KI layout](#)

On this page

1 Tutorial Overview

1.1 Guidelines & Cheatsheets

2 Tools

2.1 IIC-OSIC-TOOLS Overview

2.2 IIC-OSIC-TOOLS Installation

2.3 PDK Selection

3 Repository Overview

3.1 Purpose

3.2 Examples

3.3 Directory Structure

3.4 Workflow

3.4.1 Top-level targets

3.4.2 Digital macro targets

3.4.3 Analog macro targets

4 Digital Macro Implementation

4.1 Directory Structure

4.2 Reading the RTL

4.2.1 counter.sv

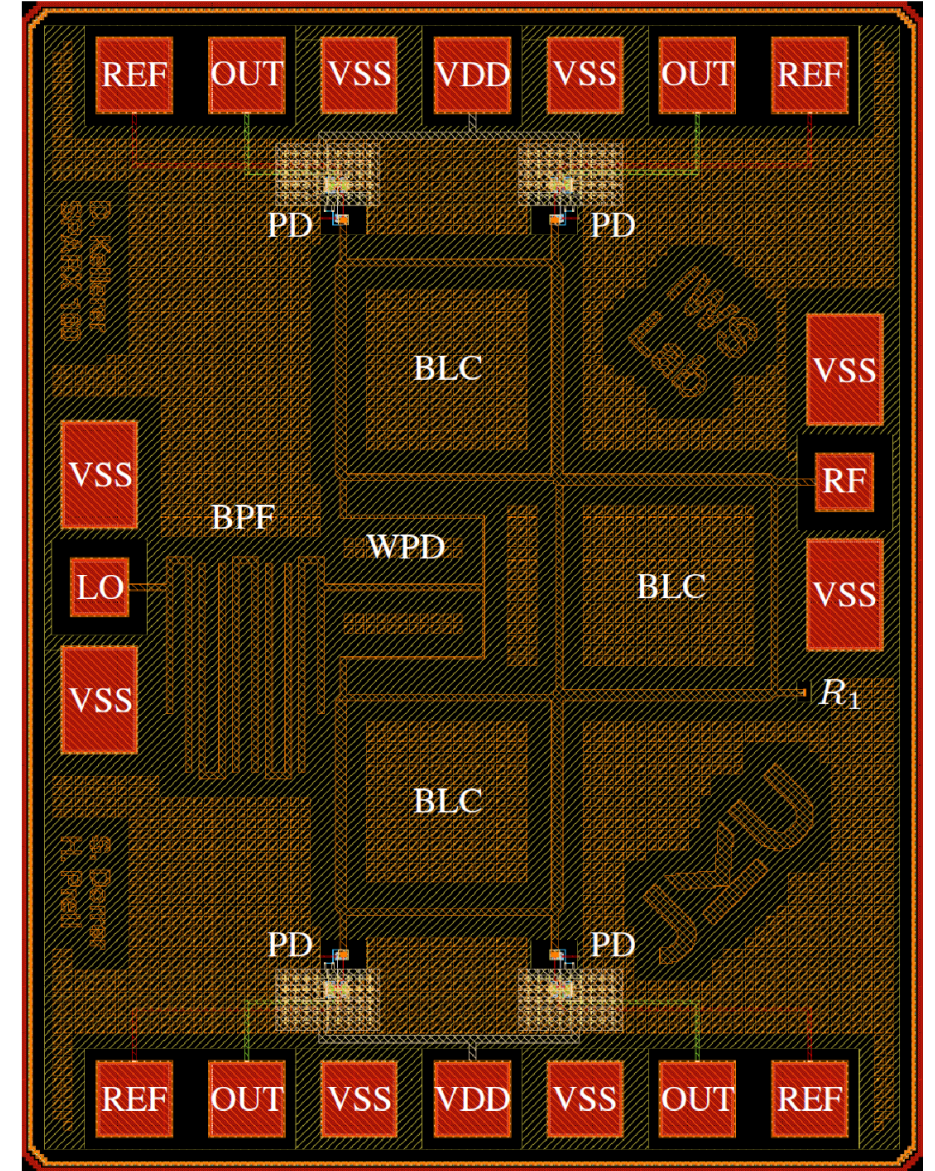
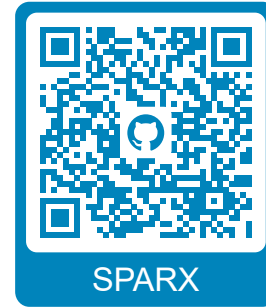
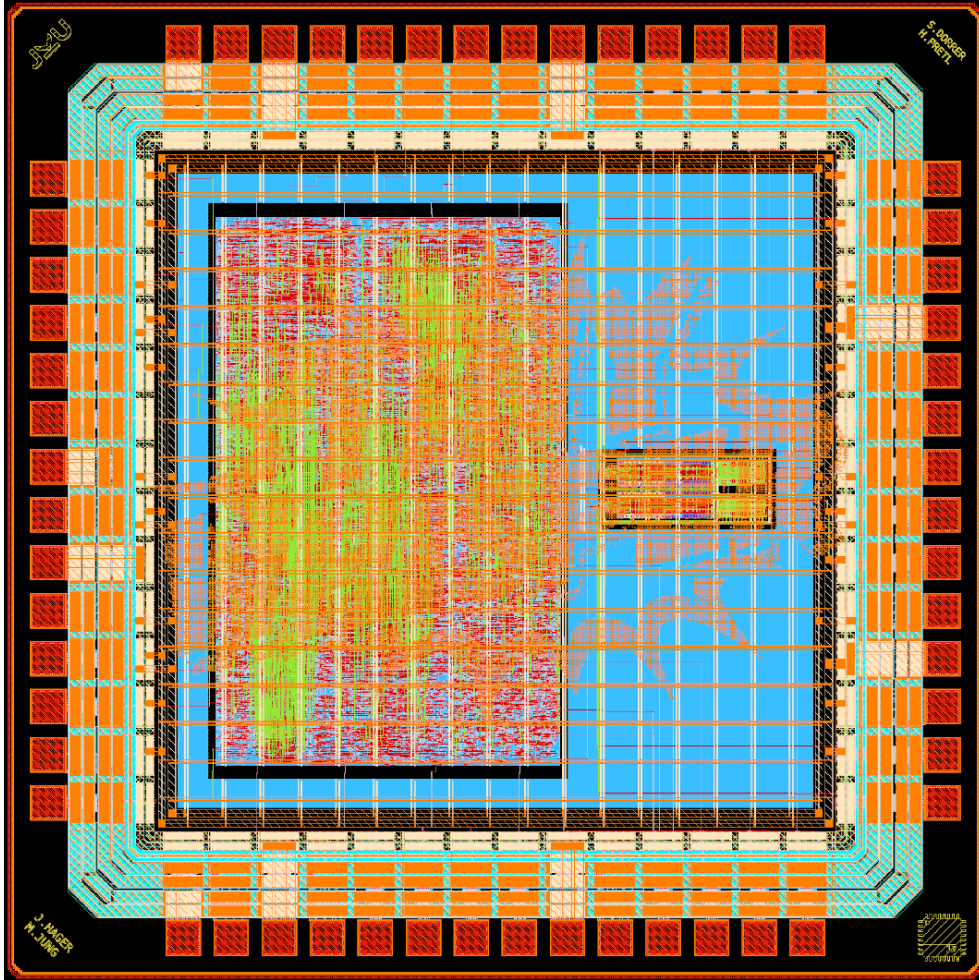
4.2.2 counter_top.sv

4.3 Linting

4.4 Simulation

4.4.1 RTL with SystemVerilog testbench

Tapeouts



Analog @ HeiChips?

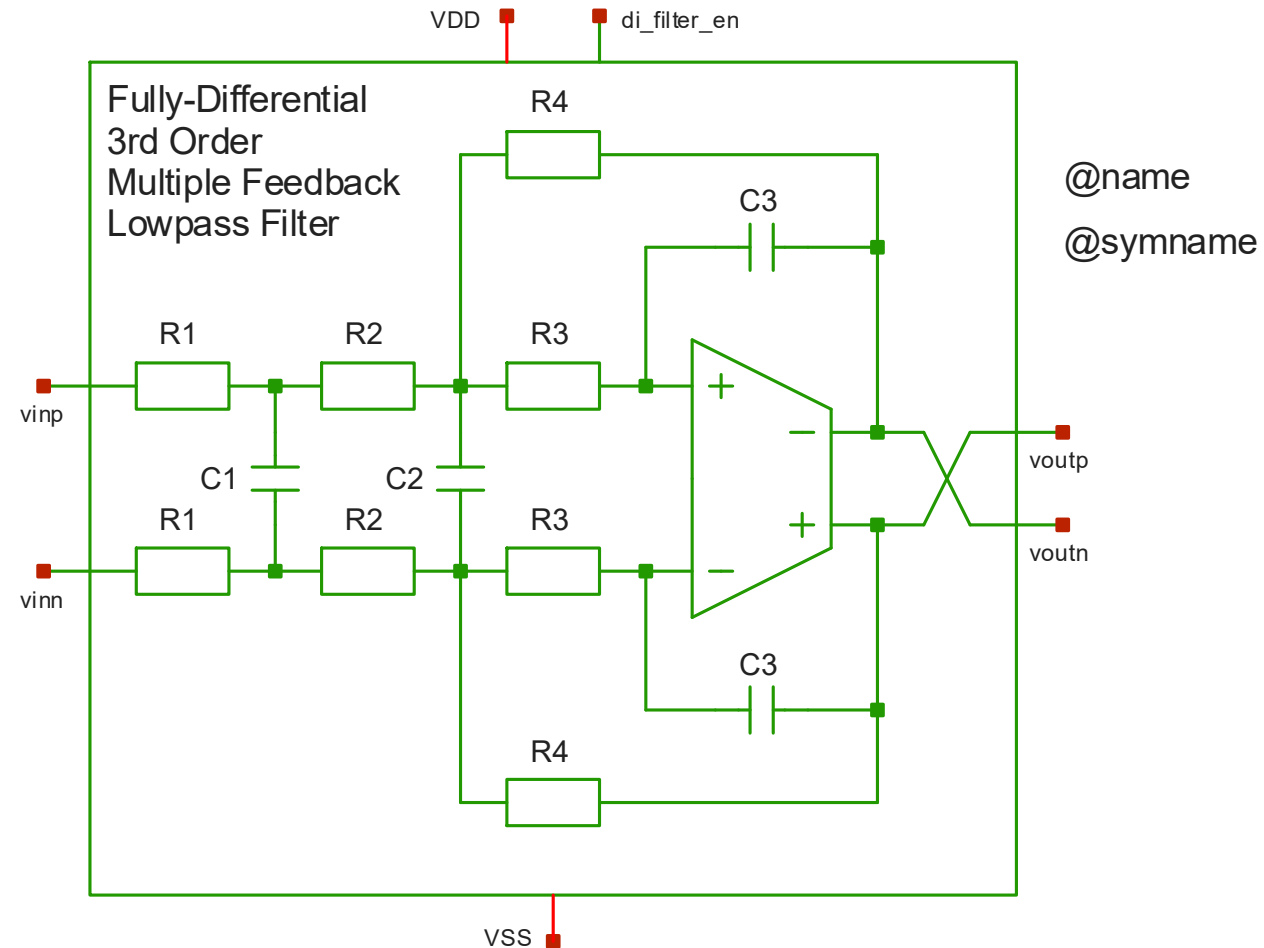
Let's do it!



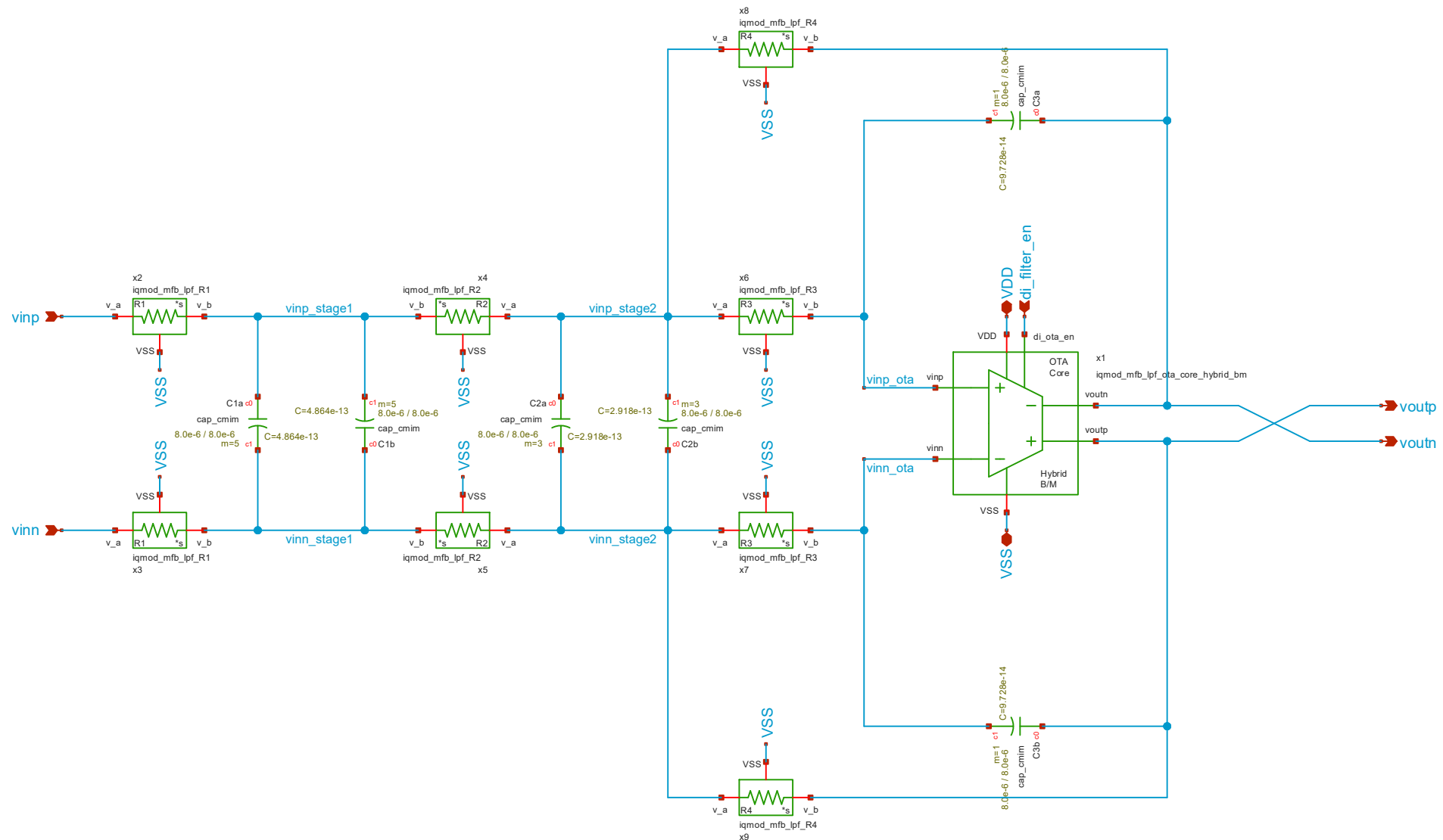
Overview

- Workshop and templates are based on **Nix** shell & **CMOS5L**
- Part 1:
 - Draw schematics and symbols (Xschem)
 - Draw testbenches and simulate (Xschem, Ngspice)
- Part 2:
 - Draw DRC & LVS clean layout and PEX (KLayout, Magic+Netgen)
 - Run post-layout simulation (Xschem, Ngspice)
- Part 3:
 - Self-Biased Single-Ended Inverter Amplifier
 - Three-Stage Ring Oscillator with Output Buffer

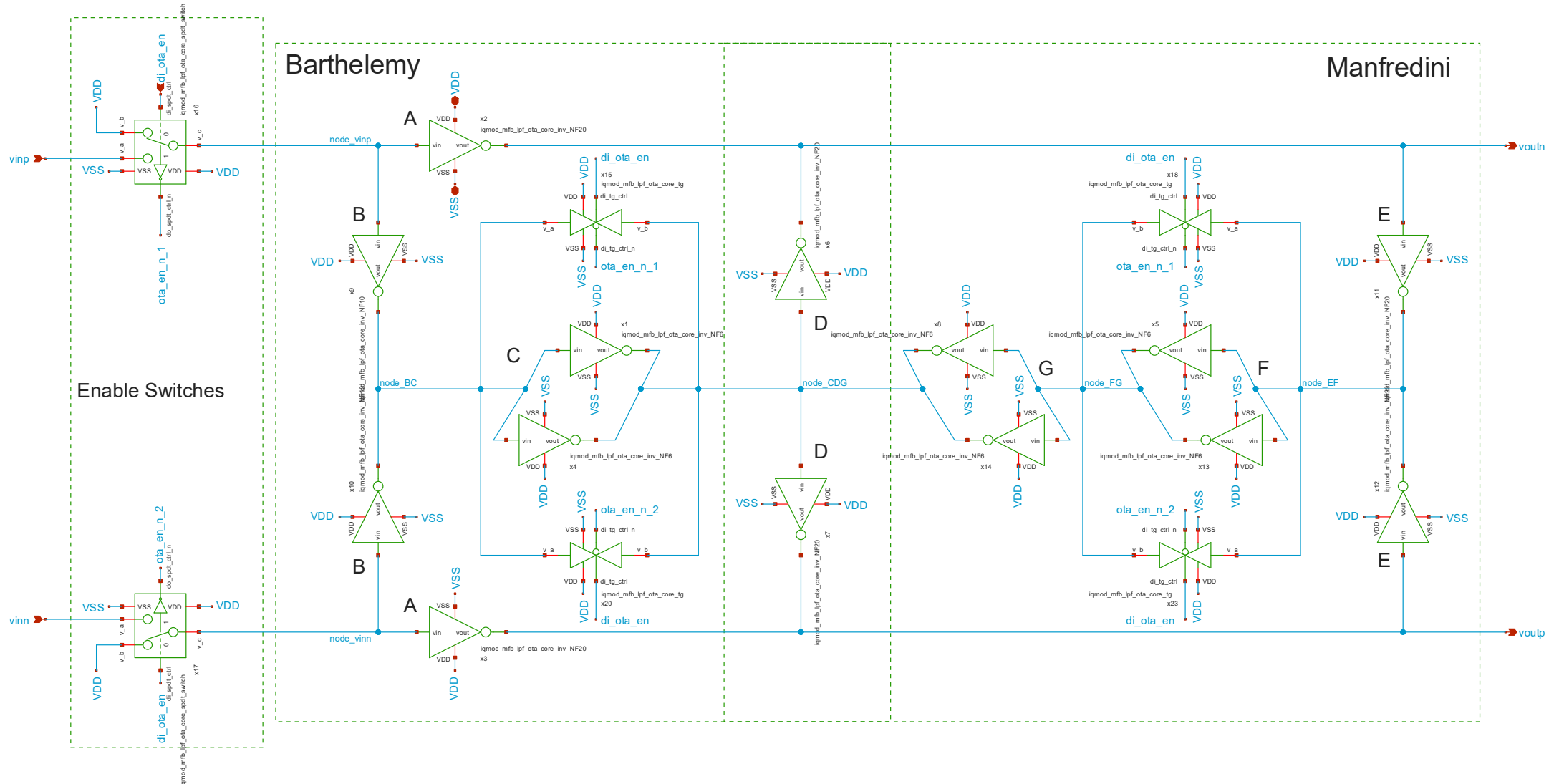
Symbol Drawing is Art!



Schematic Drawing is Art!



Schematic Drawing is Art!



Testbench Drawing is Art!

Testbench for AC CL analysis - Third Order MFB Lowpass Filter

NGSPICE

```
.include ../../netlist/pe/igmod_mfb_lpf_pex.spice
.param VDD=1.5
.param Vcm=VDD/2
.param temp=27
.param cload=10p
.options savecurrensts klu method=gear reitot=1e-3 abstot=1e-12 gmin=1e-12 rshunt=1e12
.control

* optran 0 0 0 1u 1m 0

save all
* save v(voutp) v(voutn) v(vout) v(vin)

set wr_vecnames
set wr_singlescale

* User Constants
let f_min = 1
let f_max = 1G
let fdc = 10

* Operating Point Analysis
op
remzerovec
write igmod_mfb_lpf_tb_ac_cl.raw
set appendwrite

* AC Analysis
ac dec 101 $const.f_min $const.f_max
remzerovec
write igmod_mfb_lpf_tb_ac_cl.raw
set appendwrite

* Plotting
let Amag = v(vout)/v(vin)
let Amag_db = vdb(Amag)
let Aarg = 180/Pi*cphase(Amag)

plot Amag_db ylabel 'Magnitude'
plot Aarg ylabel 'Phase'
plot Amag_db Aarg ylabel 'Magnitude, Phase'

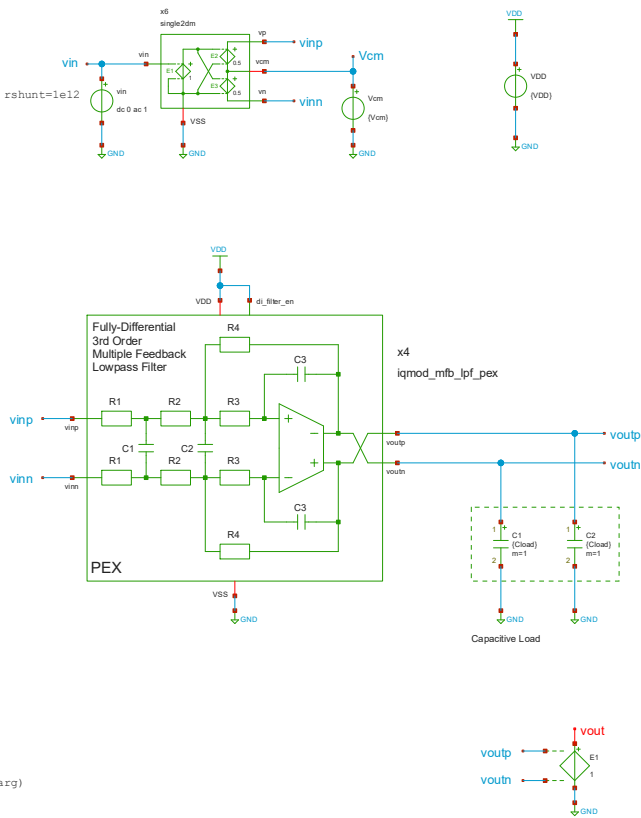
* Measurements
meas ac Adc_cl_db find Amag_db when frequency = fdc
print Adc_cl_db

let Amag_fc = Adc_cl_db-3
* meas ac fcl find frequency when Amag_db = Amag_fc RISE=1
* print fcl

meas ac fcu find frequency when Amag_db = Amag_fc FALL=1
print fcu

* Write Data
unset appendwrite
set wr_vecnames
set wr_singlescale
wrdata ../plot_simulations/data/igmod_mfb_lpf_tb_ac_cl.txt v(Amag_db) v(Aarg)

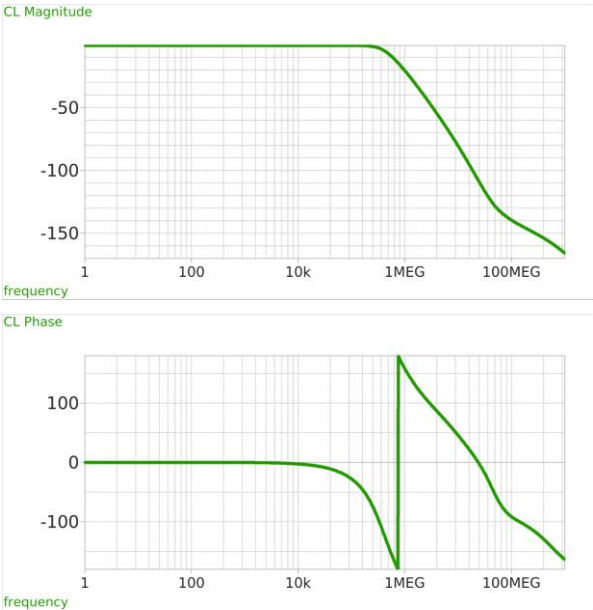
*quit
.endc
```



- ➡ Simulate
- ➡ Annotate OP
- ➡ Load waves

MODEL

```
.lib cornerMOSlv.lib mos_tt
.lib cornerMOShv.lib mos_tt
.lib cornerHBT.lib hbt_typ
.lib cornerRES.lib res_typ
.lib cornerCAP.lib cap_typ
.lib cornerDIO.lib dio_tt
```



Simon Dorrer

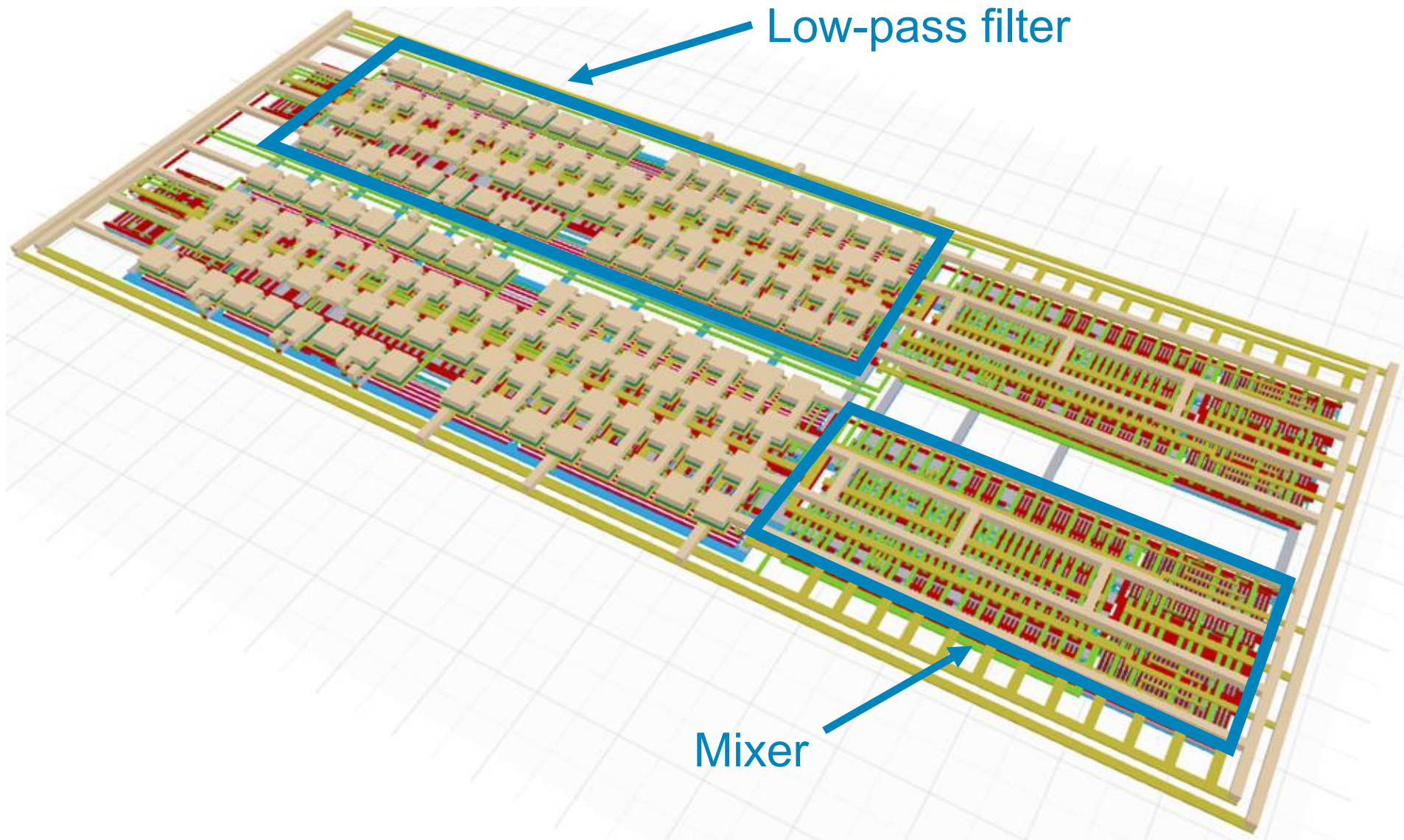
Page of

Rev. 1.0

2026-07-31 18:38:09



IQ Modulator



Let's START!



GitHub repository



Documentation



**JOHANNES KEPLER
UNIVERSITY LINZ**