JOHANNES KEPLER
UNIVERSITY LINZ

An Ultra-Low-Power High-Density Wireless Biomedical Sensing System, or

How to Build Mixed-Signal SoC with Open-Source Tools

Harald Pretl (harald.pretl@jku.at | github:hpretl | slack:hpretl)
Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
2023-06-27 IHP OpenPDK Workshop

JOHANNES KEPLER UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Agenda

- Introduction
- Open-source IC design environment
- IIC designs in open-source
- Conclusion

Personal Background

Personal Background

- Located in Linz, Austria

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools
- Since 2015 professor at Johannes Kepler University Linz,
  chair of the Institute for Integrated Circuits (https://iic.jku.at)

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools
- Since 2015 professor at Johannes Kepler University Linz,
  chair of the Institute for Integrated Circuits (https://iic.jku.at)
- Background in analog, mixed-signal, and radio-frequency integrated circuits

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools
- Since 2015 professor at Johannes Kepler University Linz,
  chair of the Institute for Integrated Circuits (https://iic.jku.at)
- Background in analog, mixed-signal, and radio-frequency integrated circuits
- Interest in open-source IC design since 2021; using open-source tools in
  the electronics curriculum at JKU

Personal Background

- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools
- Since 2015 professor at Johannes Kepler University Linz, chair of the Institute for Integrated Circuits (https://iic.jku.at)
- Background in analog, mixed-signal, and radio-frequency integrated circuits
- Interest in open-source IC design since 2021; using open-source tools in the electronics curriculum at JKU
- Founding member of IEEE Solid-State Circuits Society “Technical Committee on the Open Source Ecosystem” (TC-OSE)

Personal Background

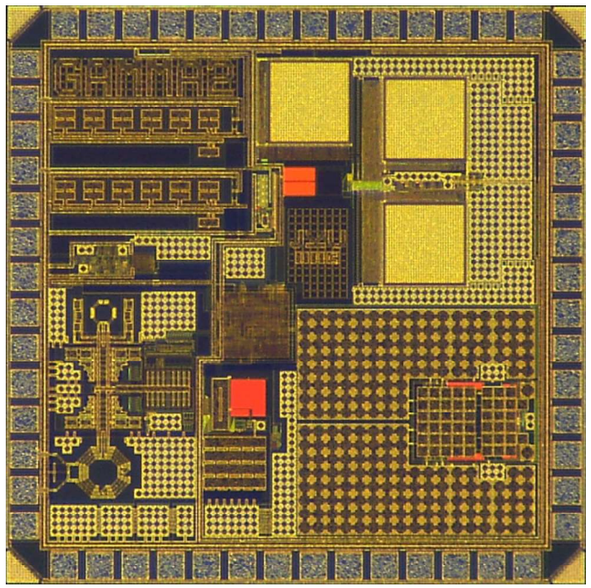
- Located in Linz, Austria
- IC design for 25 years with Infineon Technologies, Intel, etc.
- Radio-frequency IC for mobile communications using commercial tools
- Since 2015 professor at Johannes Kepler University Linz,
  chair of the Institute for Integrated Circuits (https://iic.jku.at)
- Background in analog, mixed-signal, and radio-frequency integrated circuits
- Interest in open-source IC design since 2021; using open-source tools in the electronics curriculum at JKU
- Founding member of IEEE Solid-State Circuits Society “Technical Committee on the Open Source Ecosystem” (TC-OSE)
- Maintainer of **`IIC-OSIC-TOOLS`** (https://github.com/iic-jku/iic-osic-tools)

Open-source HW and IC design are important for small/ medium-sized companies and academia to push innovation, allow exchange, and train the next generation of IC engineers.

Examples of What We Do At IIC:

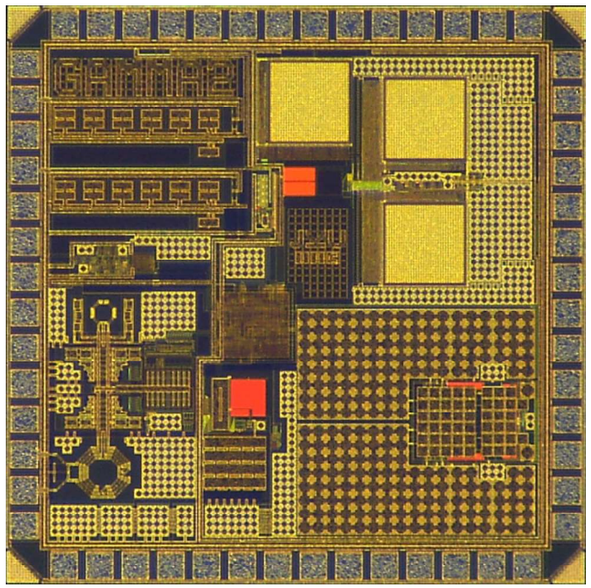
Examples of What We Do At IIC:

Mixed-signal/RF 180nm CMOS

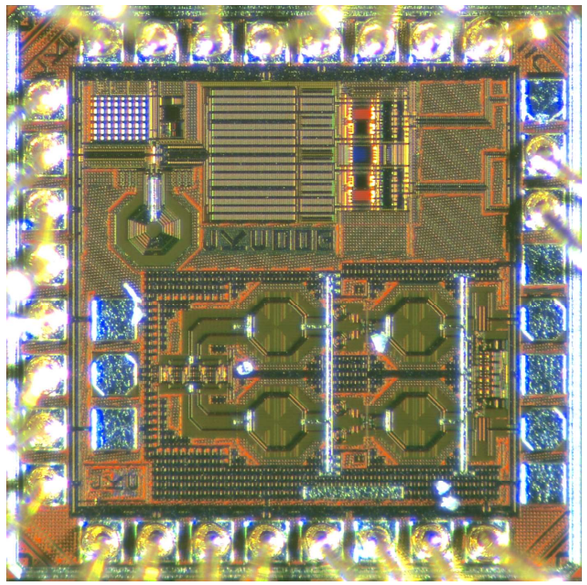


Examples of What We Do At IIC:

Mixed-signal/RF 180nm CMOS

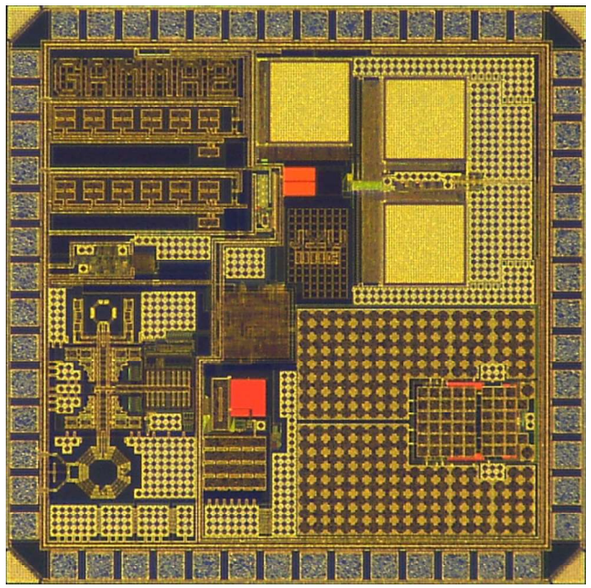


RF/mm-wave 28nm CMOS

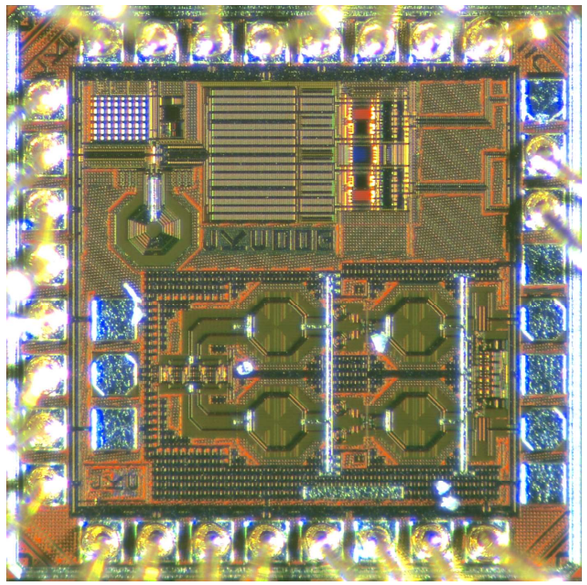


Examples of What We Do At IIC:

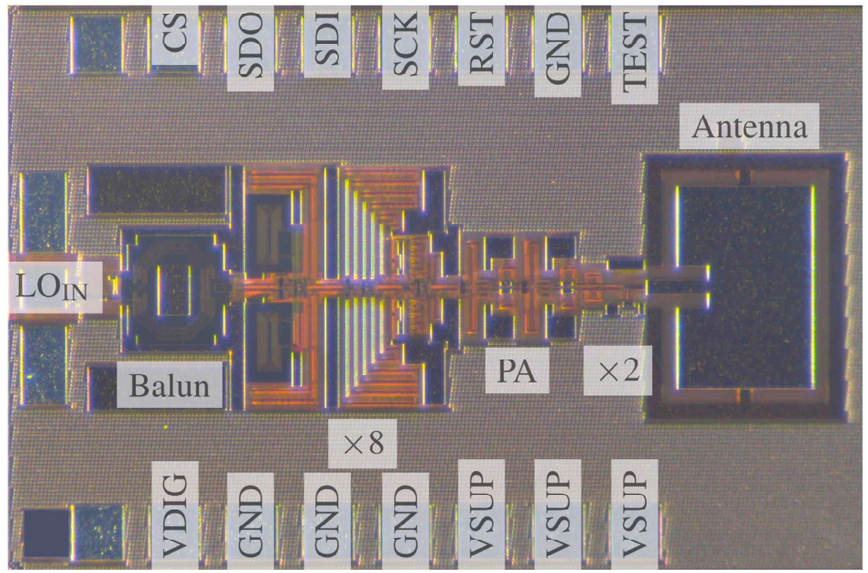
Mixed-signal/RF 180nm CMOS



RF/mm-wave 28nm CMOS

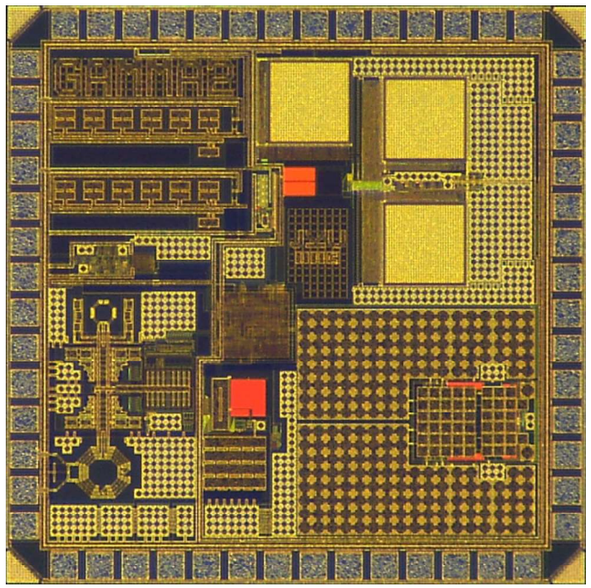


mm-wave/THz 130nm SiGe:C BiCMOS

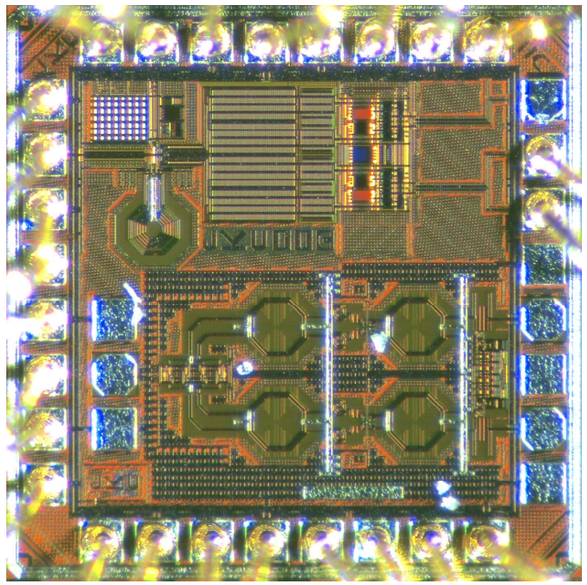


Examples of What We Do At IIC:

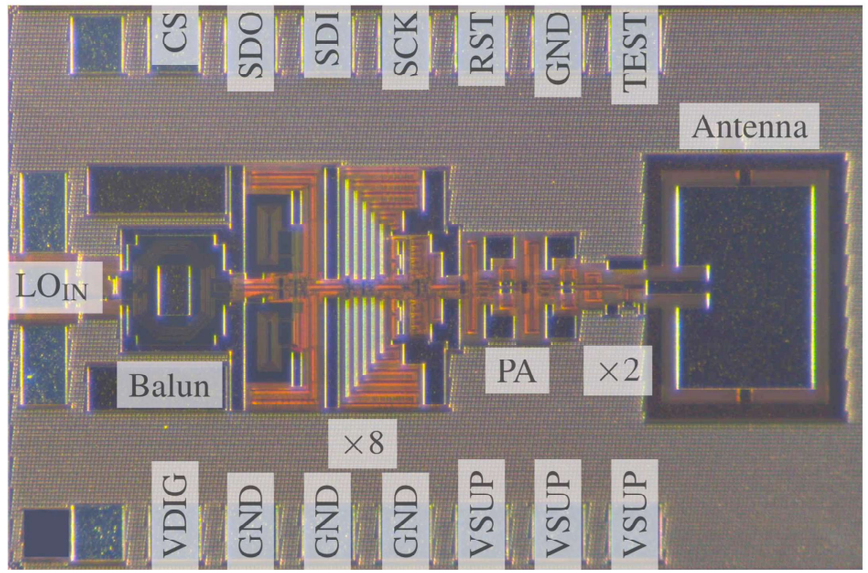
Mixed-signal/RF 180nm CMOS

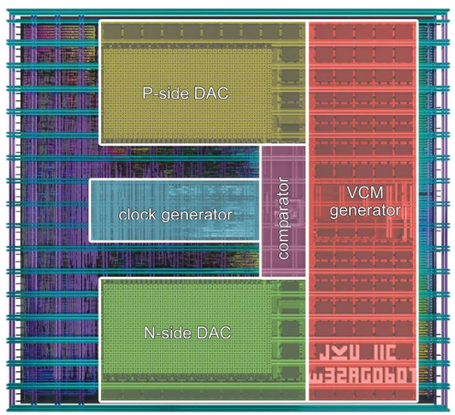


RF/mm-wave 28nm CMOS



mm-wave/THz 130nm SiGe:C BiCMOS

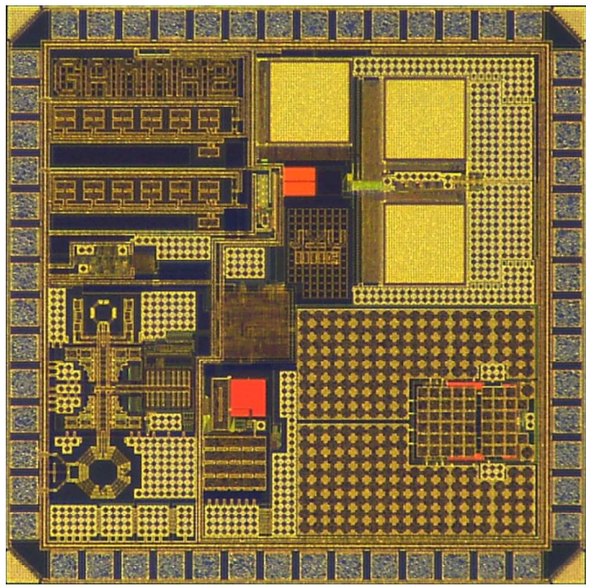




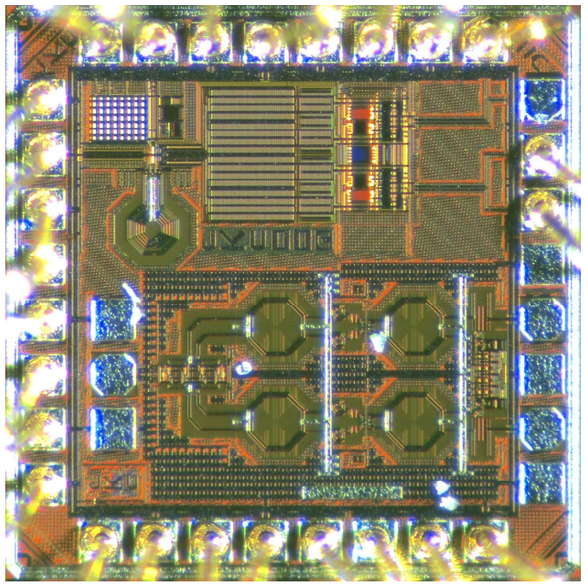
Mixed-signal in 130nm CMOS Open-Source Tools & PDK

Examples of What We Do At IIC:

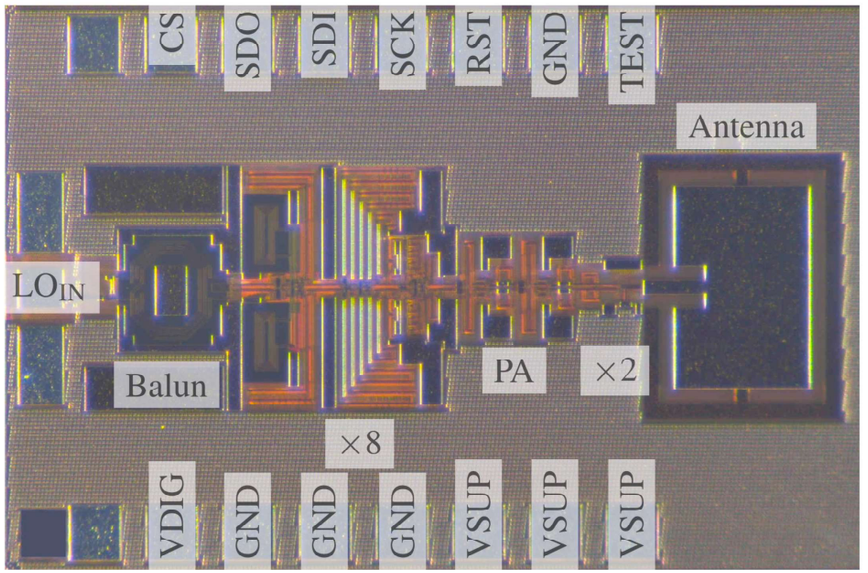
Mixed-signal/RF 180nm CMOS



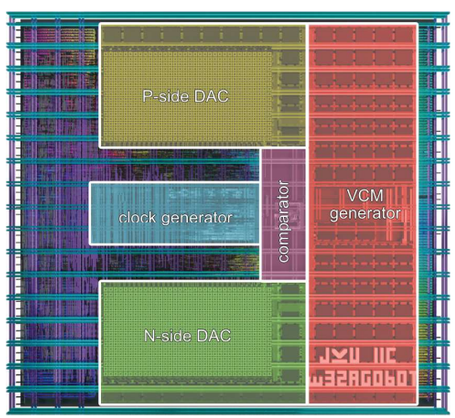
RF/mm-wave 28nm CMOS



mm-wave/THz 130nm SiGe:C BiCMOS



[1] S. Schmickl, T. Faseth, and H. Pretl, “An RF-Energy Harvester and IR-UWB Transmitter for Ultra-Low-Power Battery-Less Biosensors,” *IEEE TCAS-I*, vol. 67, no. 5, pp. 1459–1468, May 2020, DOI:10.1109/tcsi.2020.2970765.
[2] S. Schmickl, T. Faseth, and H. Pretl, “An Untrimmed 14-bit Non-Binary SAR-ADC Using 0.37 fF-Capacitors in 180 nm for 1.1 µW at 4 kS/s,” *27th IEEE ICECS*, 2020, DOI:10.1109/icecs49266.2020.9294971.
[3] S. Schmickl, T. Schumacher, P. Fath, T. Faseth, and H. Pretl, “A 350-nW Low-Noise Amplifier With Reduced Flicker-Noise for Bio-Signal Acquisition,” *Austrochip*, 2020, DOI:10.1109/austrochip51129.2020.9232981.
[4] T. Schumacher, M. Stadelmayer, and H. Pretl, “A Mixer-First Receiver With Class-F Adiabatic Switching,” *IEEE SSC-L*, vol. 5, pp. 45–48, 2022, DOI:10.1109/lssc.2022.3152571.
[5] M. Stadelmayer, T. Schumacher, and H. Pretl, “A 66% Tuning-Range, 60-GHz Quadrature Ring Oscillator Using Current-Combining for Frequency Multiplication,” *IEEE MWC-L*, vol. 32, no. 2, pp. 141–144, 2022, DOI:10.1109/lmwc.2021.3118324.
[6] G. Zachl, C. Mangiavillano, R. K. R. Mitta, T. Schumacher, H. Pretl, A. Stelzer, “A 0.32-THz 6.6-dBm Single-Chain CW Transmitter Using On-Chip Antenna With 2.65% DC-to-THz Efficiency,” *presented at the IEEE RFIC*, 2023.
[7] M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1



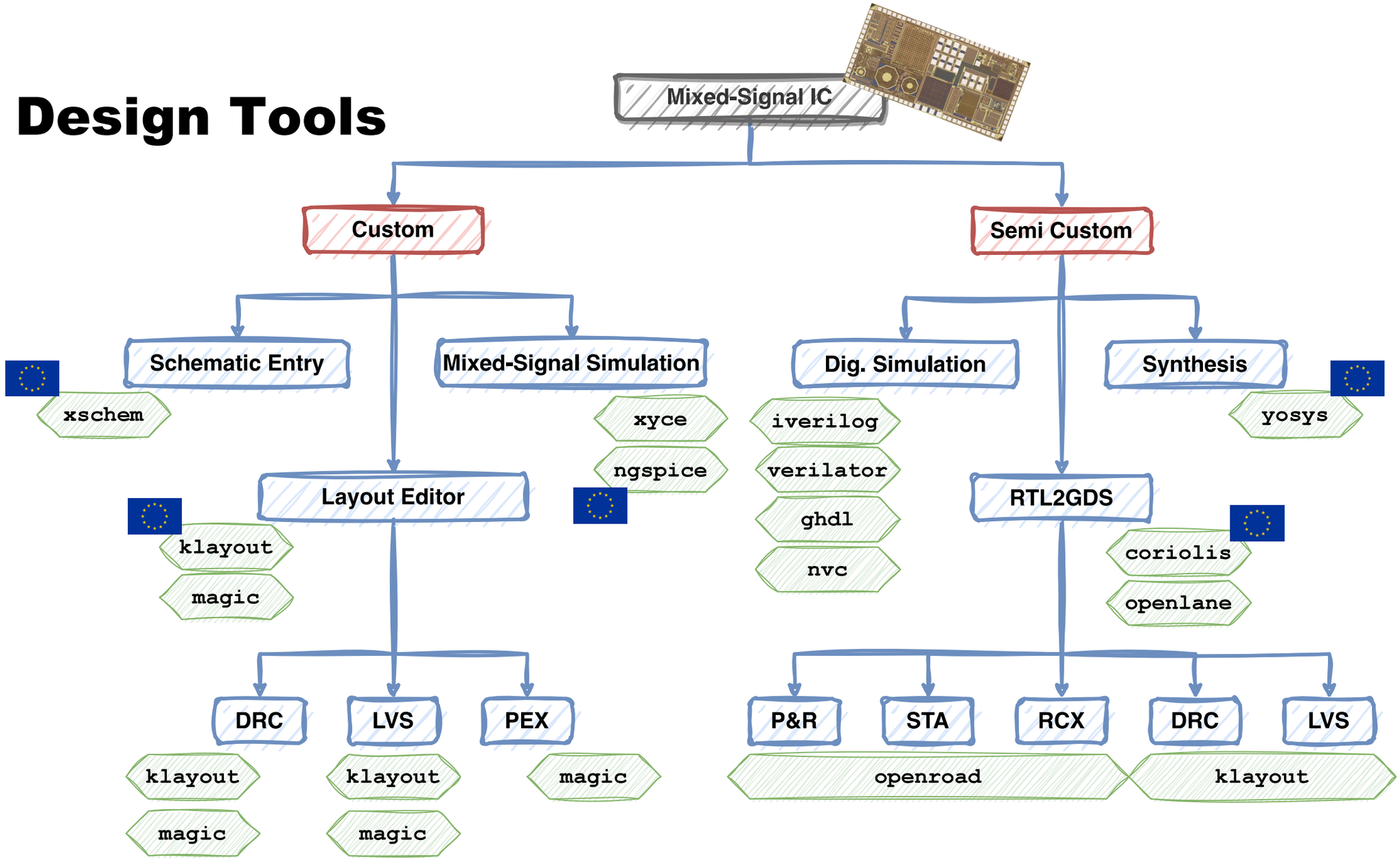
Mixed-signal in 130nm CMOS Open-Source Tools & PDK

JOHANNES KEPLER UNIVERSITY LINZ

Agenda

- Introduction
- **Open-source IC design environment**
- **IIC designs in open-source**
- **Conclusion**

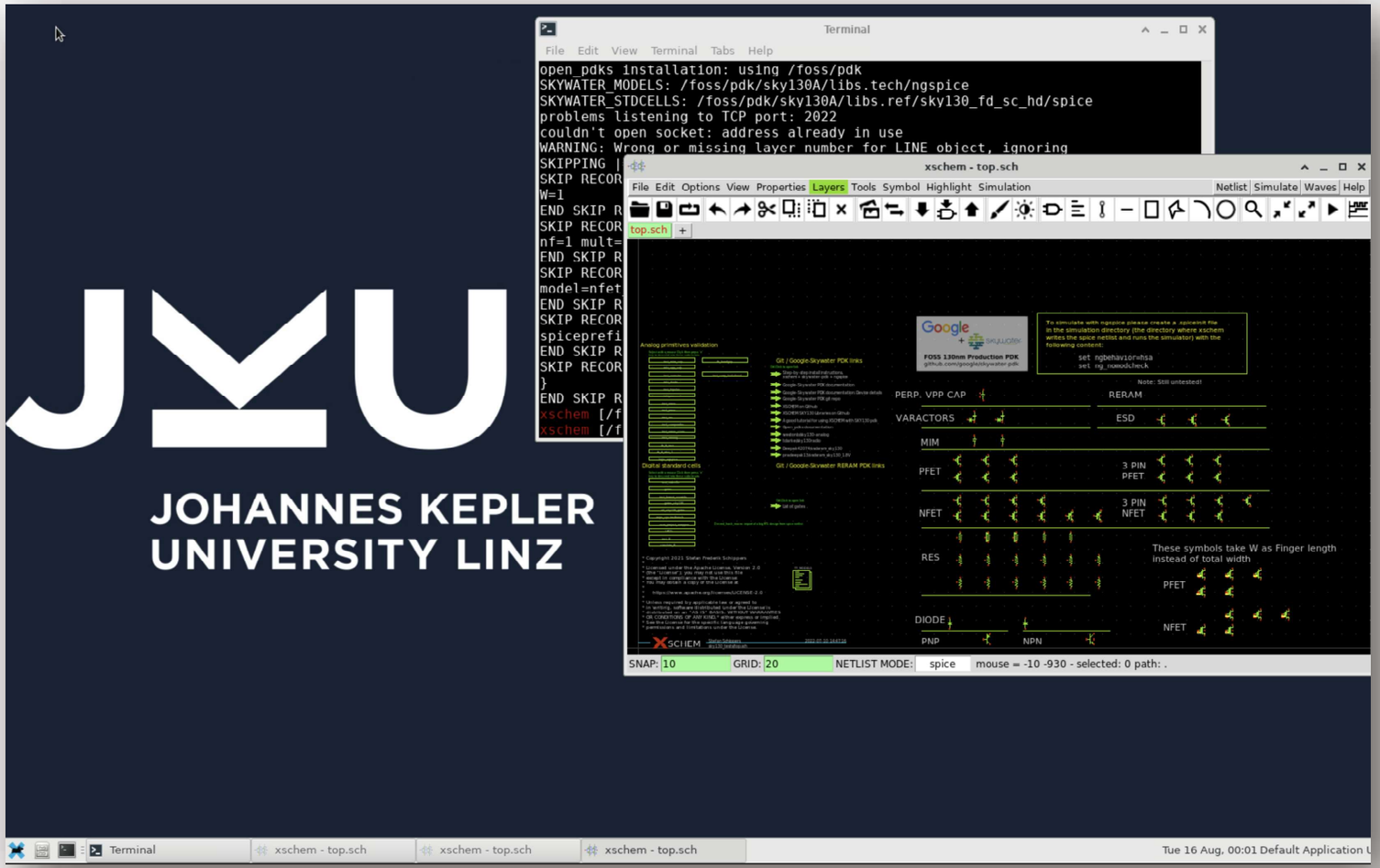
IC Design Tools



Our Contribution: All-in-One Docker Image

`https://github.com/iic-jku/IIC-OSIC-TOOLS`

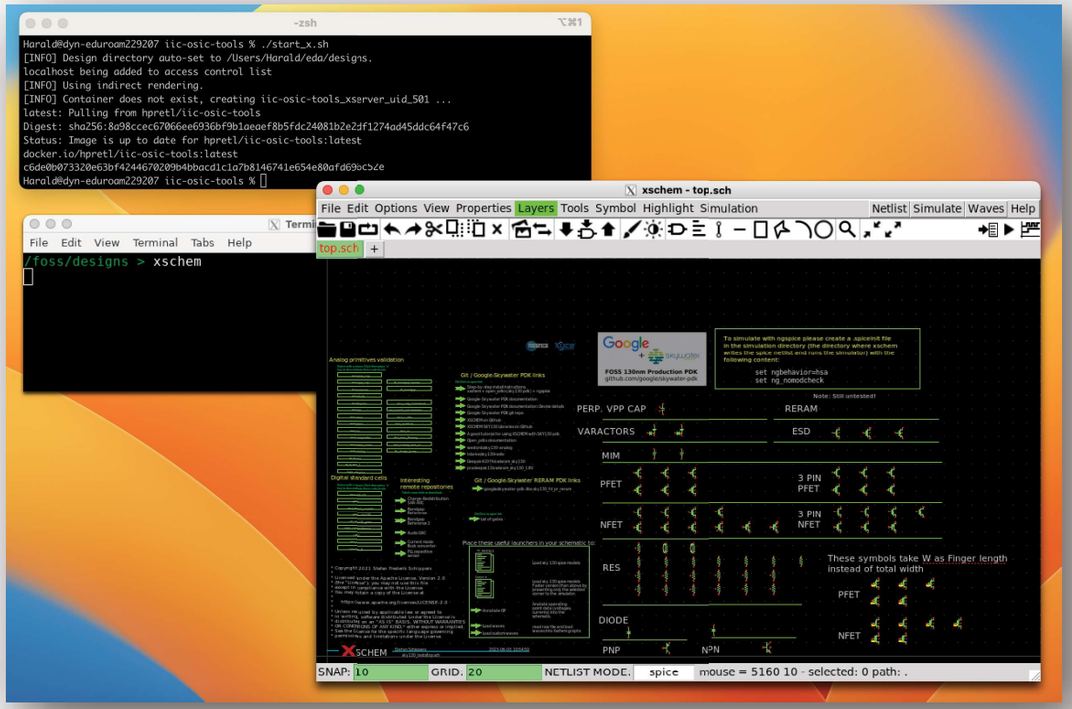
Forked from efabless.com



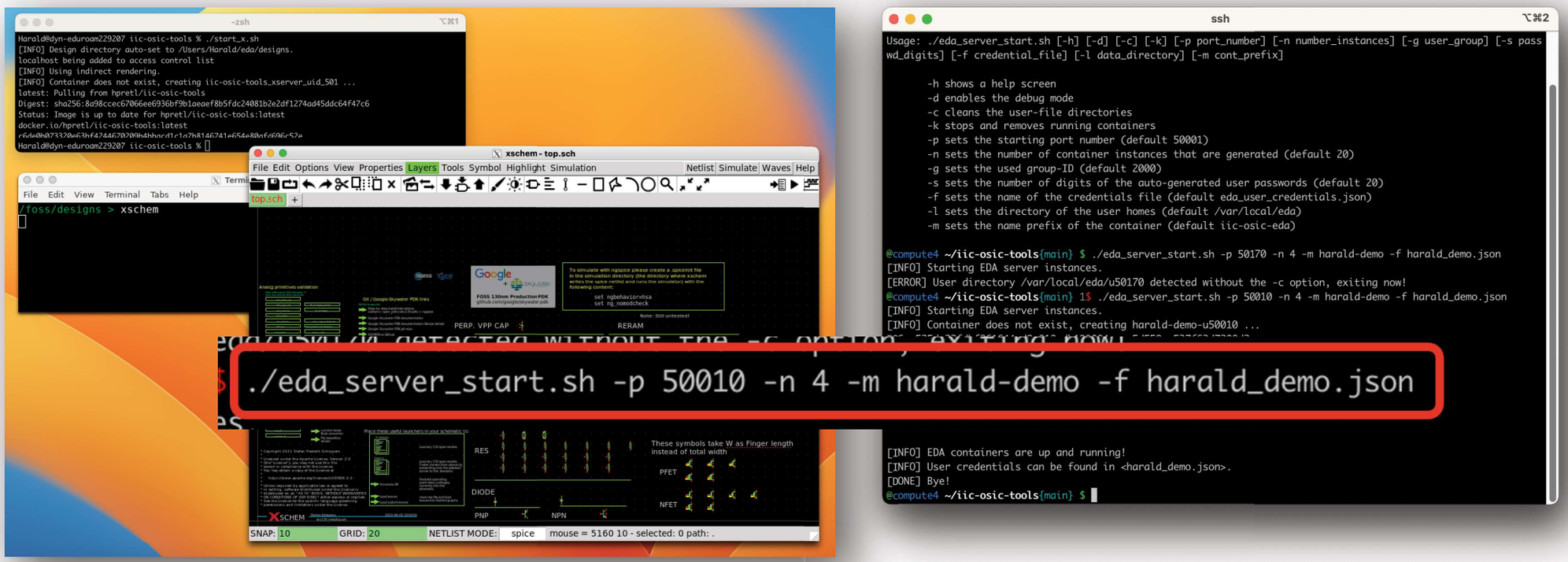
`https://github.com/iic-jku/IIC-OSIC-TOOLS`

Usage: VNC, seamless via X11, cloud-based, on `amd64` & `arm64`

https://github.com/iic-jku/IIC-OSIC-TOOLS
Usage: VNC, seamless via X11, cloud-based, on `amd64` & `arm64`

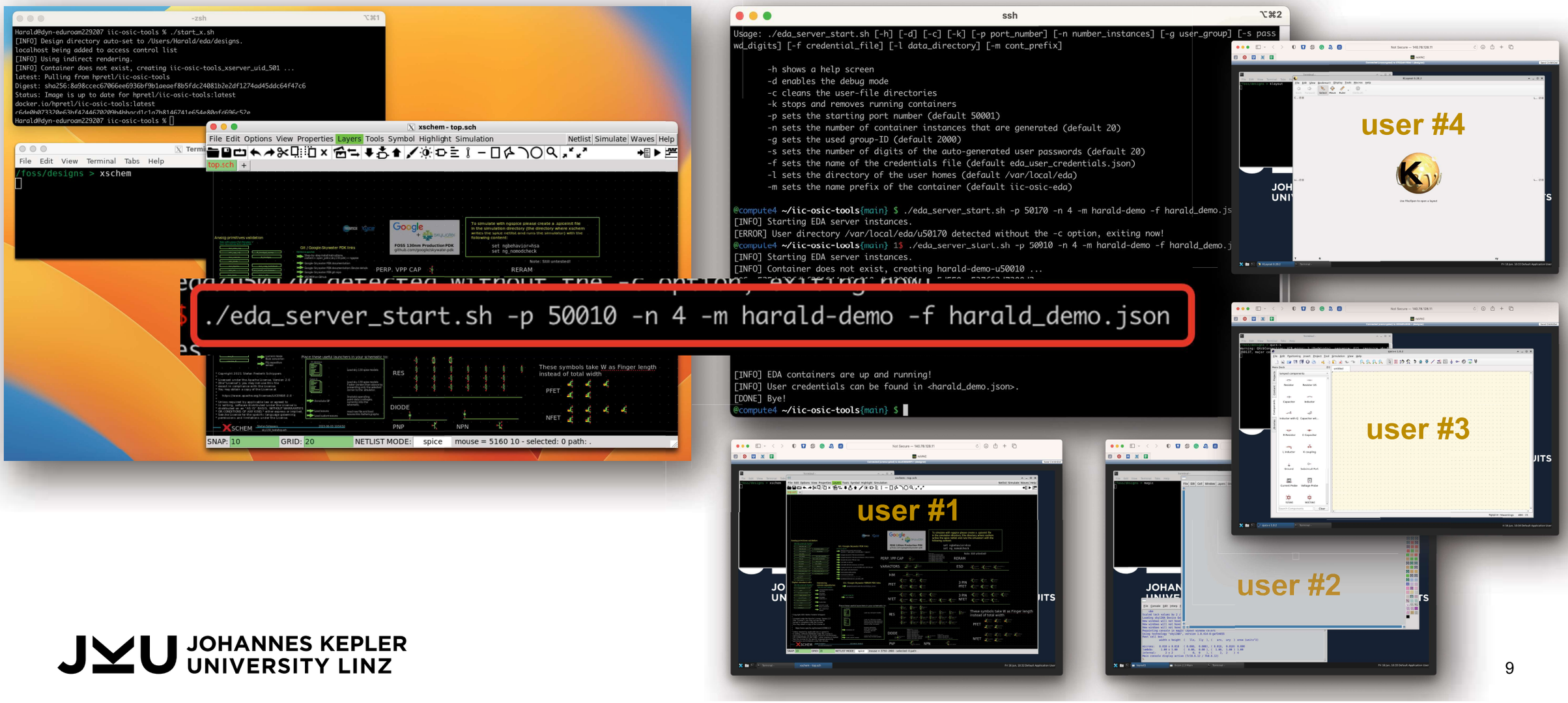


https://github.com/iic-jku/IIC-OSIC-TOOLS
Usage: VNC, seamless via X11, cloud-based, on amd64 & arm64



https://github.com/iic-jku/IIC-OSIC-TOOLS

Usage: VNC, seamless via X11, cloud-based, on `amd64` & `arm64`

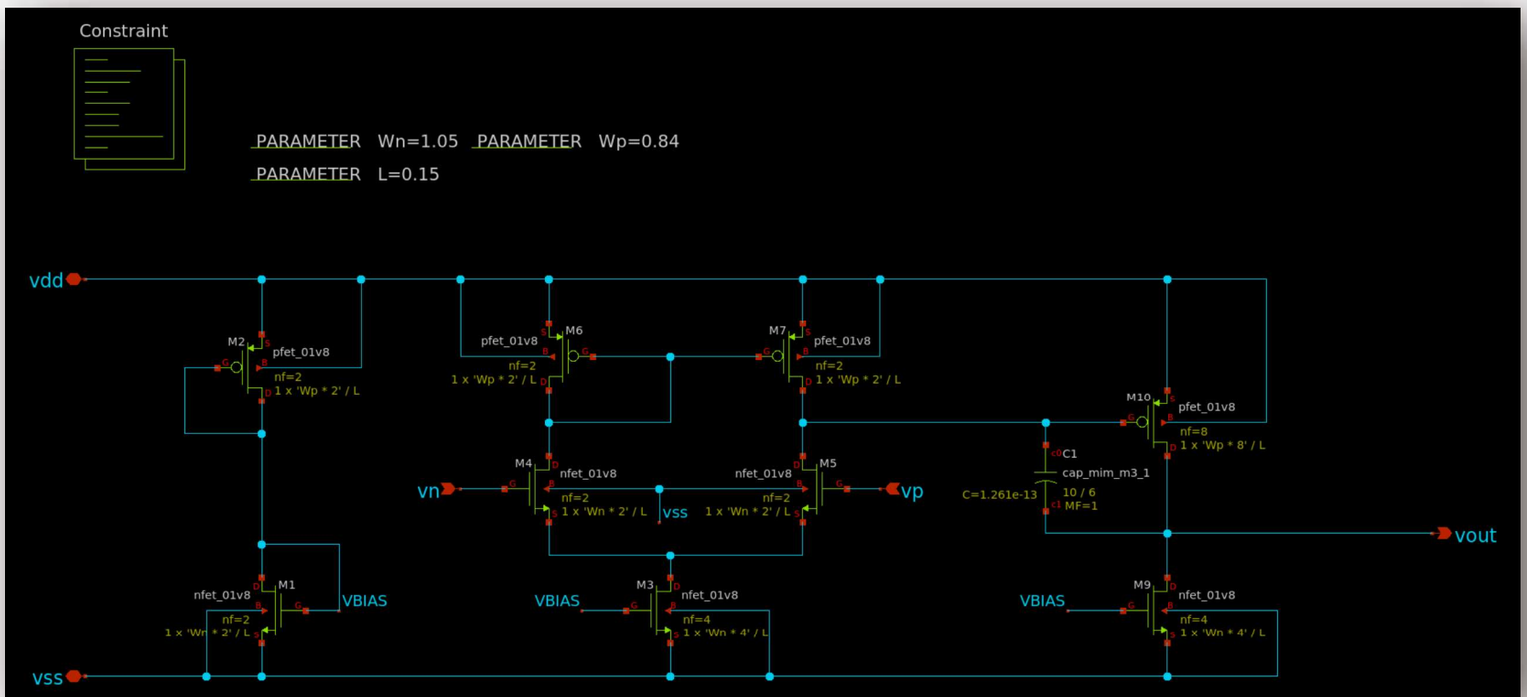


https://github.com/iic-jku/IIC-OSIC-TOOLS
Currently installed tools/PDKs (list growing...)

- **amaranth** a Python-based HDL toolchain
- **cocotb** simulation library for writing VHDL and Verilog test benches in Python
- **covered** Verilog code coverage
- **cvc** circuit validity checker (ERC)
- **edalize** Python abstraction library for EDA tools
- **fault** design-for-test (DFT) solution
- **fusesoc** package manager and build tools for SoC
- **gaw3-xschem** waveform plot tool for **xschem**
- **gdsfactory** Python library for GDS generation
- **gdspy** Python module for creation and manipulation of GDS files
- **gds3d** a 3D viewer for GDS files
- **gf180mcu** GlobalFoundries 180nm CMOS PDK
- **ghdl** VHDL simulator
- **gtkwave** waveform plot tool for digital simulation
- **ihp-sg13g2** IHP Microelectronics 130nm SiGe:C BiCMOS PDK (partial PDK yet)
- **irsim** switch-level digital simulator
- **iverilog** Verilog simulator
- **klayout** layout viewer and editor for GDS and OASIS
- **magic** layout editor with DRC and PEX
- **netlistsvg** draws SVG netlist from a **yosys** JSON netlist
- **netgen** netlist comparison (LVS)
- **ngspice** SPICE analog and mixed-signal simulator
- **ngspyce** Python bindings for **ngspice**
- **nvc** VHDL simulator and compiler
- **open_pdks** PDK setup scripts
- **openlane** digital RTL2GDS flow
- **openlane2** rewrite of **openlane** in Python, 2nd generation
- **openram** OpenRAM Python library
- **openroad** RTL2GDS engine used by **openlane** and **openlane2**
- **osic-multitool** collection of useful scripts and documentation (DRC, LVS, PEX)
- **padring** padring generation tool
- **pyopus** simulation runner and optimization tool for analog circuits
- **pyrtl** collection of classes for pythonic RTL design
- **pyspice** interface **ngspice** and **xyce** from Python
- **pyverilog** Python toolkit for Verilog
- **RF toolkit** with **FastHenry2**, **FasterCap**, and **openEMS**
- **qucs-s** simulation environment with RF emphasis
- **rggen** code generation tool for configuration and status registers
- **spyci** analyze/plot **ngspice**/**xyce** output data with Python
- **qflow** Verilog file conversion
- **volare** version manager (and builder) for open-source PDKs
- **risc-v toolchain** GNU compiler toolchain for RISC-V RV32I cores
- **siliconcompiler** modular build system for hardware
- **sky130** SkyWater Technologies 130nm CMOS PDK
- **verilator** fast Verilog simulator
- **xschem** schematic editor
- **xyce** fast parallel SPICE simulator (incl. **xdm** netlist conversion tool)
- **yosys** Verilog synthesis tool (with **ghdl** plugin for VHDL synthesis)

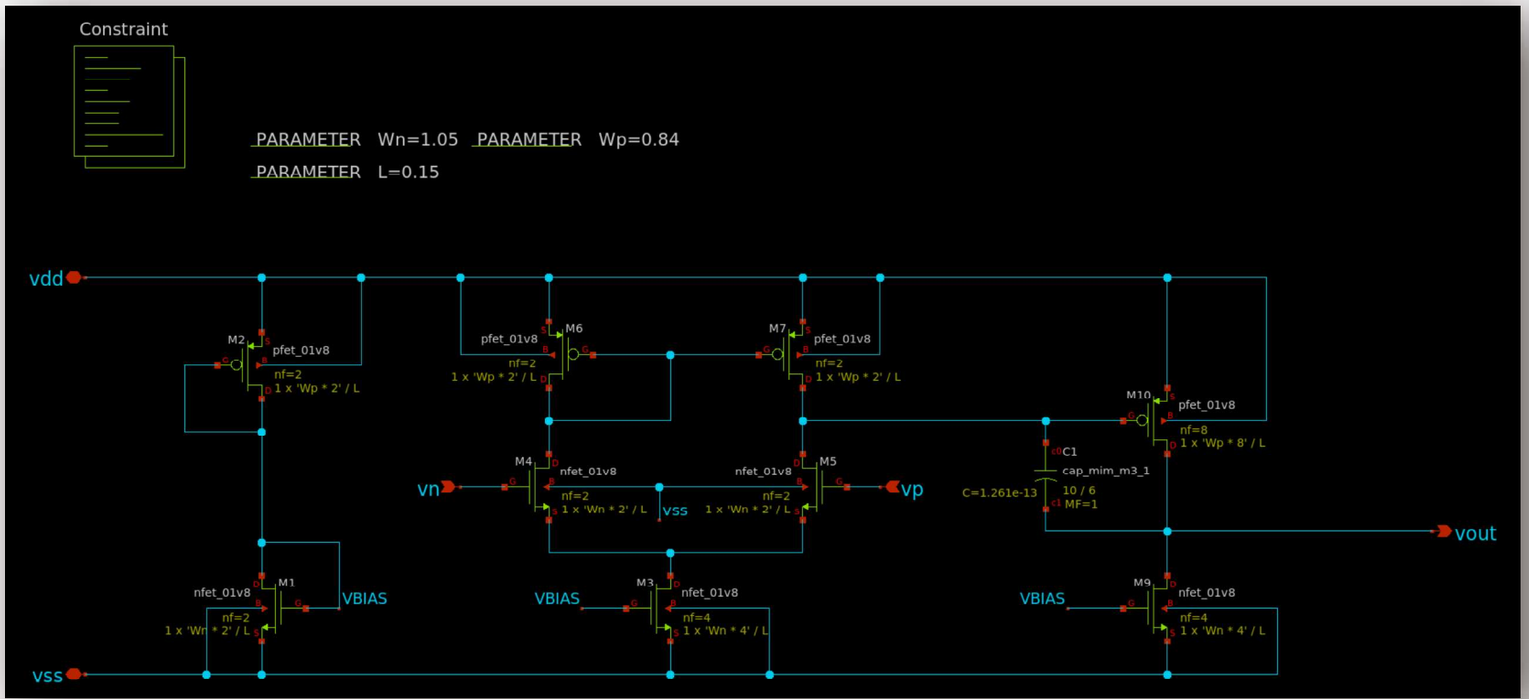
https://github.com/iic-jku/IIC-OSIC-TOOLS
Coming soon — continue to close the gaps

- Integrate **`align`** (automatic layout generation; `amd64` done, `arm64` build fail to be fixed)



https://github.com/iic-jku/IIC-OSIC-TOOLS
Coming soon — continue to close the gaps

- Integrate **`align`** (automatic layout generation; `amd64` done, `arm64` build fail to be fixed)

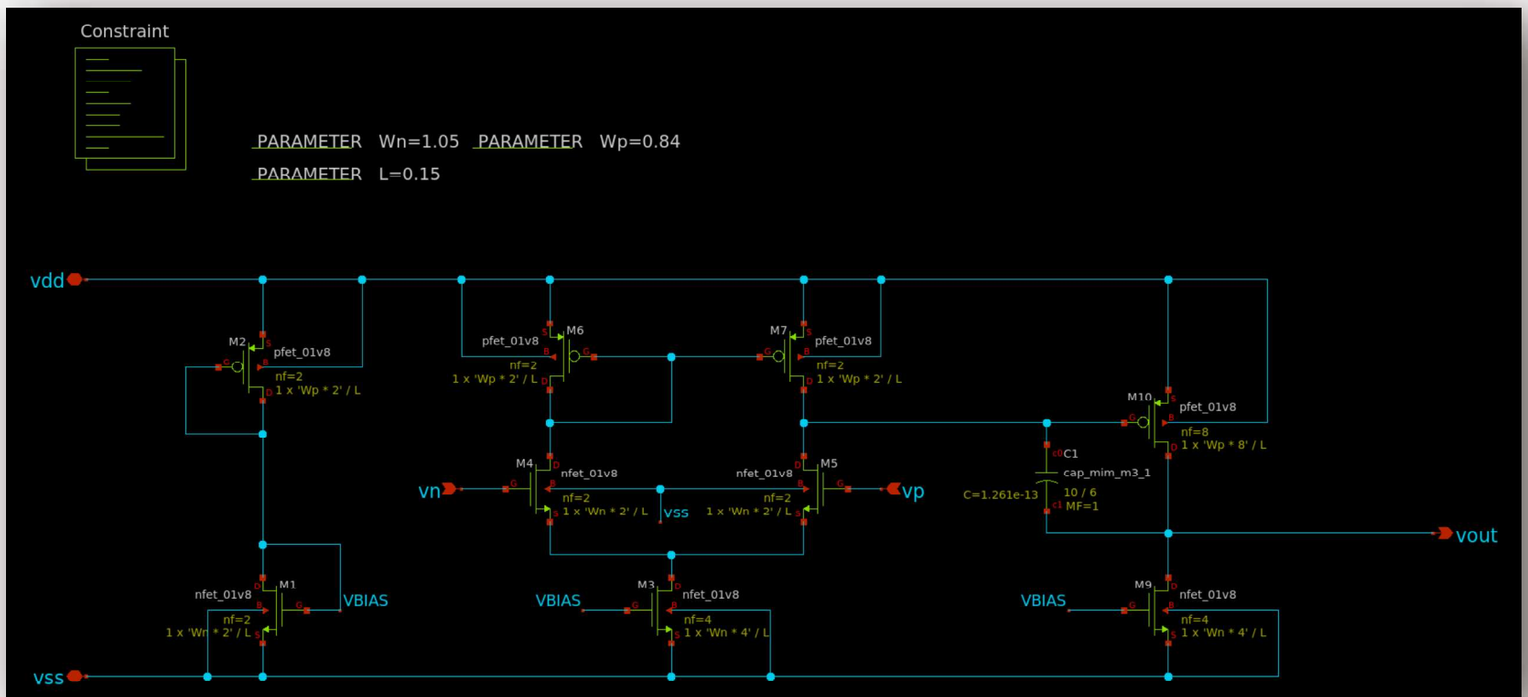



```
/foss/designs/IIC-ALIGN/SKY130-PDK/ALIGN/two_stage_opamp > bash ../iic-align.sh
two_stage_opamp_compensated -d -e 10
```

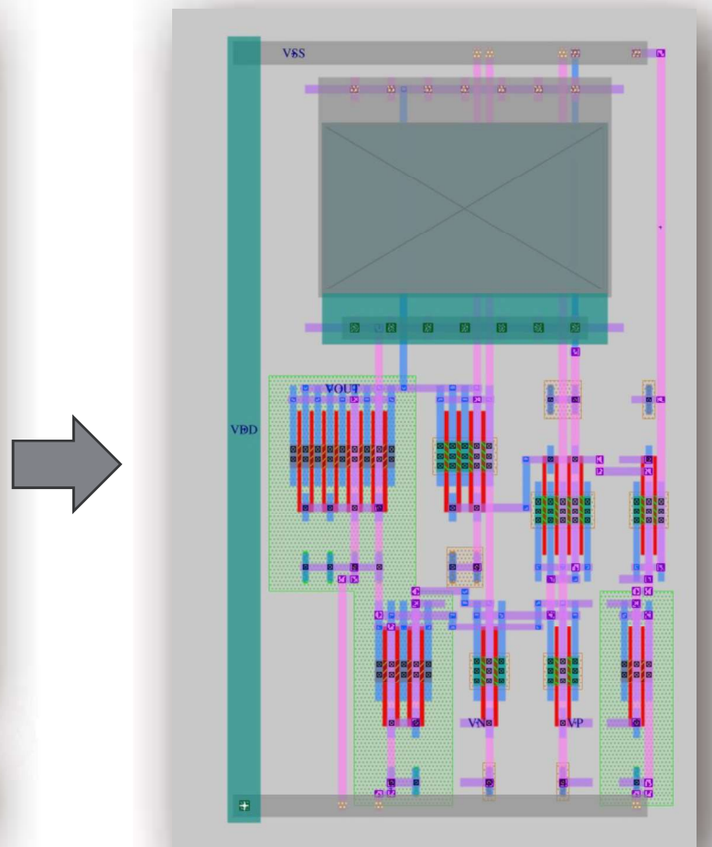
JOHANNES KEPLER UNIVERSITY LINZ

https://github.com/iic-jku/IIC-OSIC-TOOLS
Coming soon — continue to close the gaps

- Integrate **`align`** (automatic layout generation; `amd64` done, `arm64` build fail to be fixed)

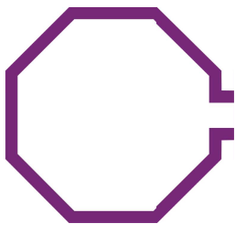



```
/foss/designs/IIC-ALIGN/SKY130-PDK/ALIGN/two_stage_opamp > bash ../iic-align.sh
two_stage_opamp_compensated -d -e 10
```



https://github.com/iic-jku/IIC-OSIC-TOOLS
Coming soon — continue to close the gaps

- Provide a **coil generator tool**

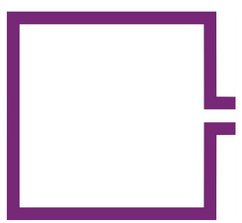


octagonal

```
/foss/designs/alpamp > ./octGen.sh -w 10 -l 150 -f 7e9 -c coil1
**********Inductor generator Started**********
- Magic layout file generated: coil1.mag
- Fasthenry2 input file generated: coil1_flat.fh
**********FastHenry2 Started******************
- Simulation done!
**********Extract LC Started******************
Not part of any matrix: Row 1:  na1  to  na12
Reading Frequency 7e+09
Inductance= 291.50 pH
Resistance= 1.549 ohm
- spice model created!: coil1.spice
Runtime : .638136873
**********Inductor generator done!************
```

File Types➔ | .mag | .spice | .fh | .sym

square



```
/foss/designs/alpamp > ./sqGen.sh -w 15 -l 100 -f 7e9 -c coil2
**********Inductor generator Started**********
- Magic layout file generated: coil2.mag
- Fasthenry2 input file generated: coil2_flat.fh
**********FastHenry2 Started******************
- Simulation done!
**********Extract LC Started******************
Not part of any matrix: Row 1:  na1  to  na8
Reading Frequency 7e+09
Inductance= 157.38 pH
Resistance= 0.843 ohm
- spice model created!: coil2.spice
Runtime : .974005266
**********Inductor generator done!************
```

Sweep parameters

```
/foss/designs/alpamp > ./iterator.sh -y 6 -x 15 -l 100 -f 7e9 -n 9 -c coilw
**********Inducgen Sweep Started**********
**********Iteration 1 ********************
Length 100, width:6.0000000000000000000
- Magic layout file generated
- Fasthenry2 input file generated
- Simulation done!
Not part of any matrix: Row 1:  na1  to  na8
Reading Frequency 7e+09
Inductance= 238.99 pH
Resistance= 1.967 ohm
Runtime : .979178593
**********Iteration 2 ********************
Length 100, width:7.0000000000000000000
- Magic layout file generated
- Fasthenry2 input file generated
- Simulation done!
Not part of any matrix: Row 1:  na1  to  na8
Reading Frequency 7e+09
Inductance= 225.65 pH
Resistance= 1.702 ohm
Runtime : .902555284
**********Iteration 3 ********************
Length 100, width:8.0000000000000000000
- Magic layout file generated
- Fasthenry2 input file generated
- Simulation done!
Not part of any matrix: Row 1:  na1  to  na8
Reading Frequency 7e+09
Inductance= 213.87 pH
Resistance= 1.505 ohm
Runtime : .936969451
**********Iteration 4 ********************
Length 100, width:9.0000000000000000000
- Magic layout file generated
- Fasthenry2 input file generated
- Simulation done!
Not part of any matrix: Row 1:  na1  to  na8
Reading Frequency 7e+09
Inductance= 203.31 pH
Resistance= 1.351 ohm
Runtime : .900903295
```

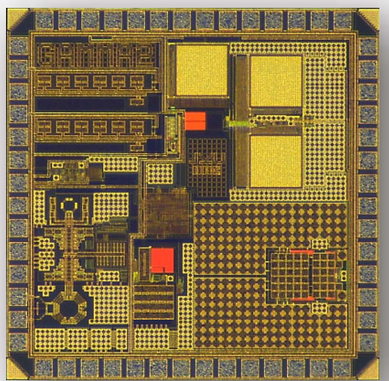
`https://github.com/iic-jku/IIC-OSIC-TOOLS`
Plans going forward — continue to close the gaps

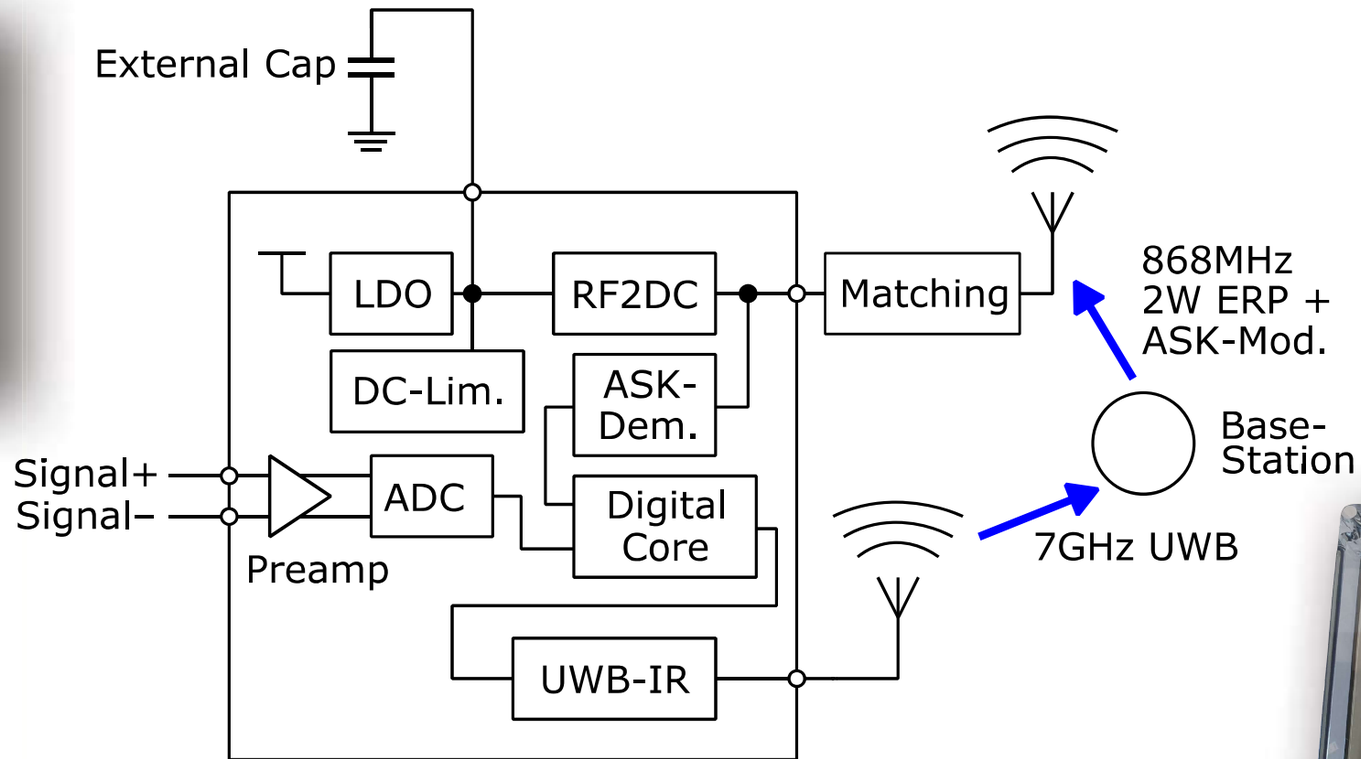
- Integrate **ALIGN**
- Provide a **coil generator tool**
- Integrate **SG13G2** and use it for designs
- Support **`riscv64`** (besides `amd64` and `arm64`)

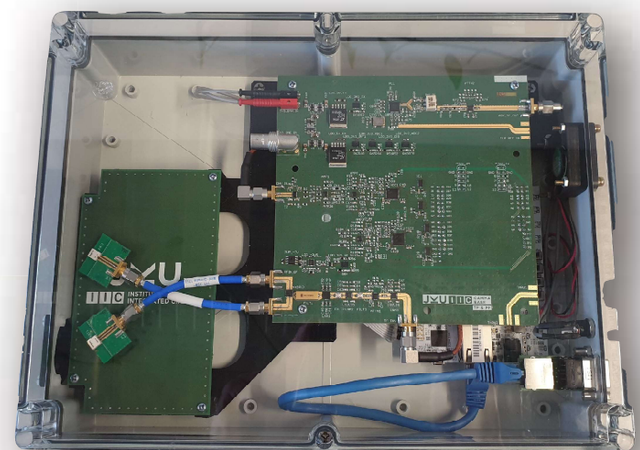
Agenda

- Introduction
- Open-source IC design environment
- **IIC designs in open-source**
- Conclusion

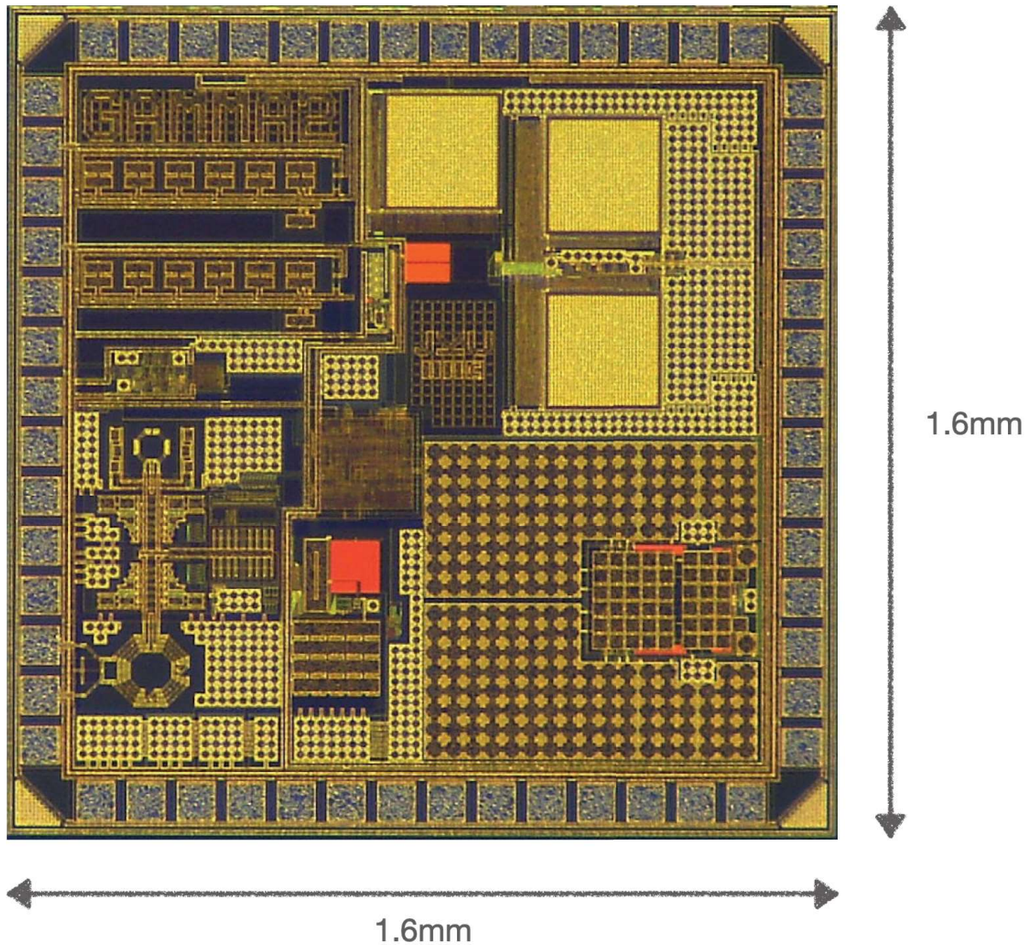
Ultra-Low Power Bio-Sensing SoC (closed-source): RF-powered SoC, wireless TX to base-station



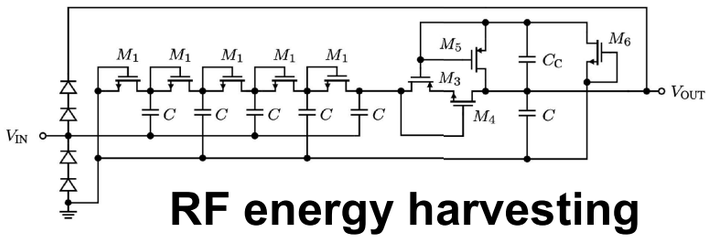




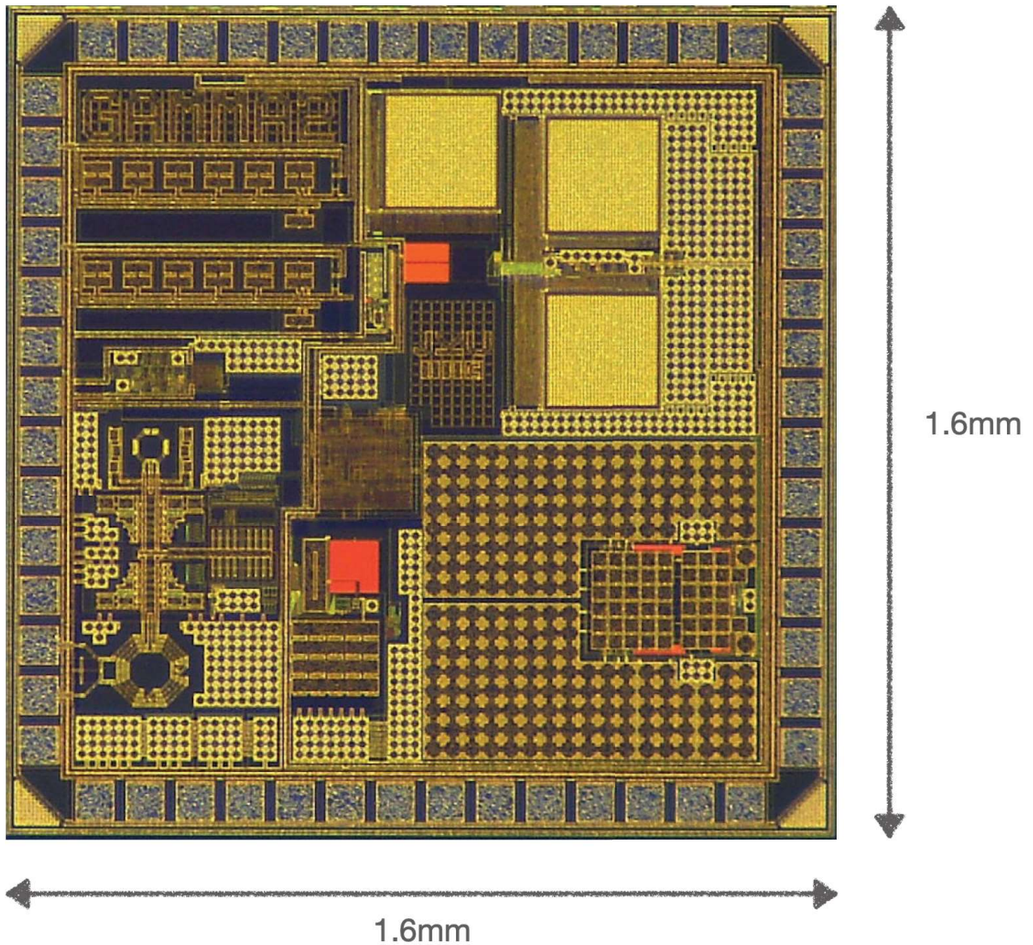
Ultra-Low Power Bio-Sensing SoC (closed-source): Low-noise amplifier, SAR ADC, UWB TX, RF harvester



Ultra-Low Power Bio-Sensing SoC (closed-source): Low-noise amplifier, SAR ADC, UWB TX, RF harvester

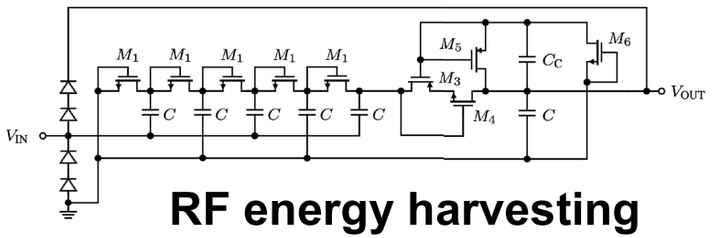


RF energy harvesting

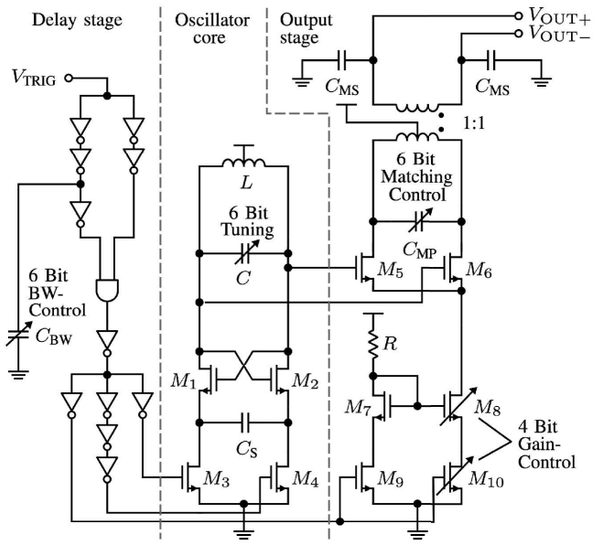


JOHANNES KEPLER UNIVERSITY LINZ

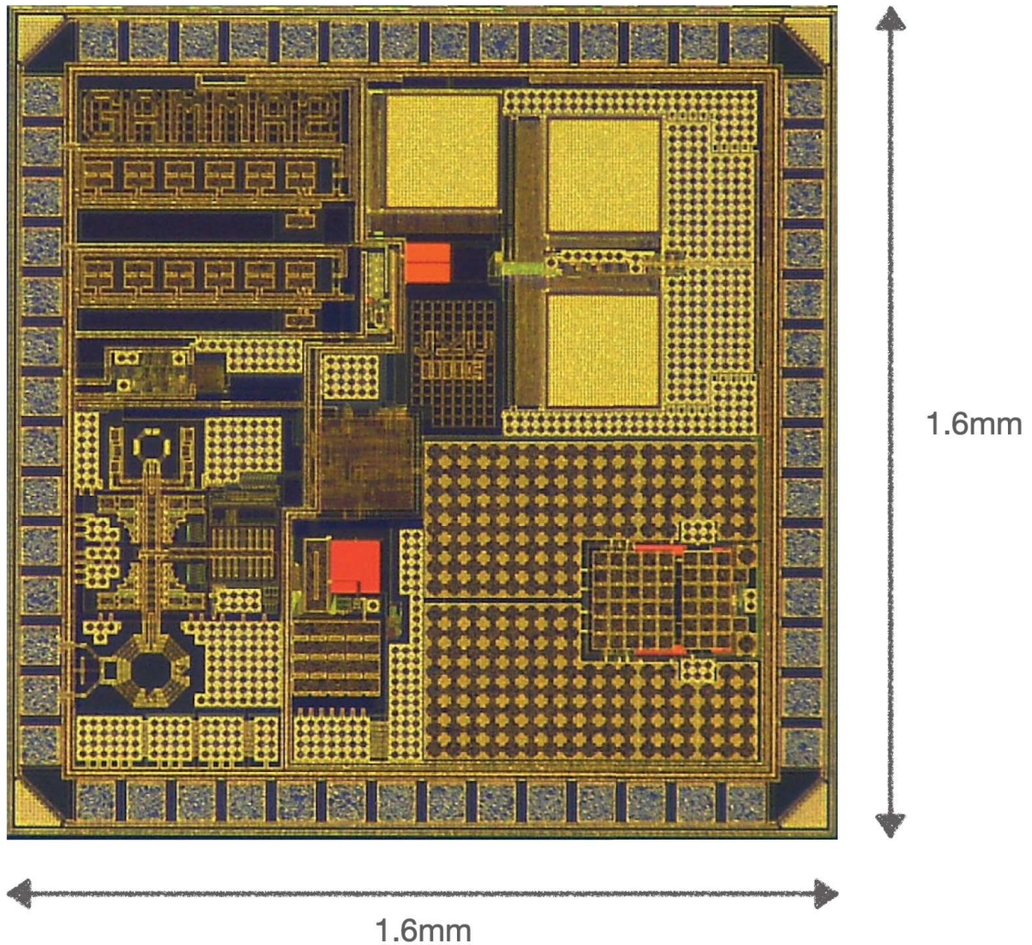
Ultra-Low Power Bio-Sensing SoC (closed-source): Low-noise amplifier, SAR ADC, UWB TX, RF harvester



RF energy harvesting

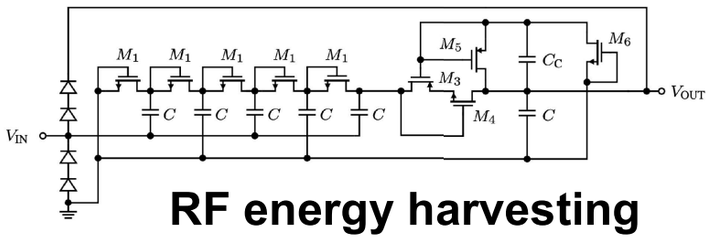


IR UWB transmitter

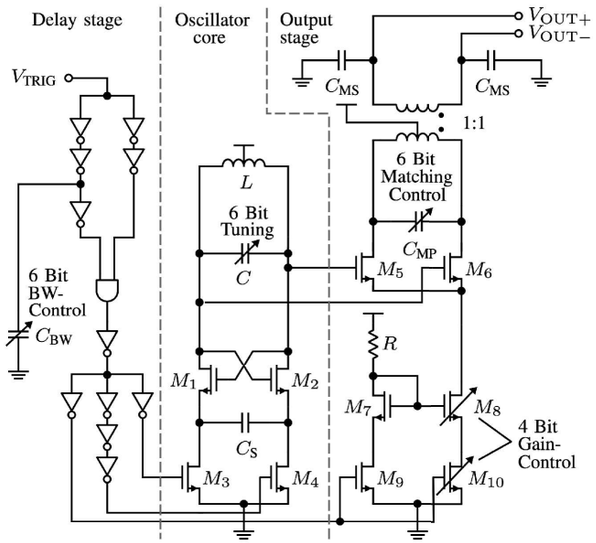


JOHANNES KEPLER UNIVERSITY LINZ

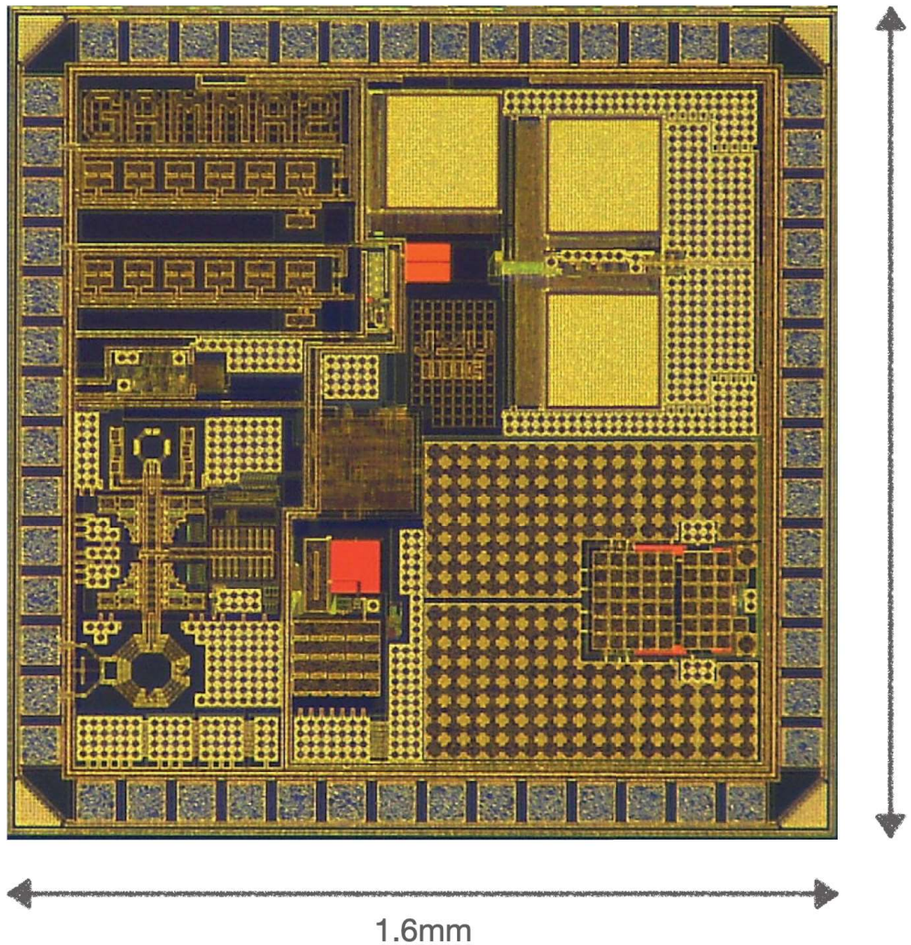
Ultra-Low Power Bio-Sensing SoC (closed-source): Low-noise amplifier, SAR ADC, UWB TX, RF harvester

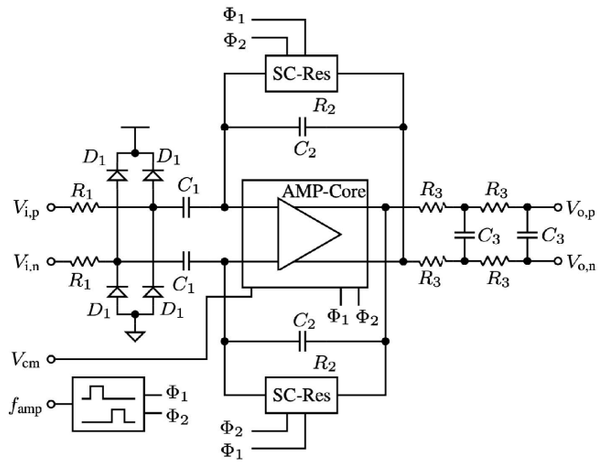


RF energy harvesting



IR UWB transmitter

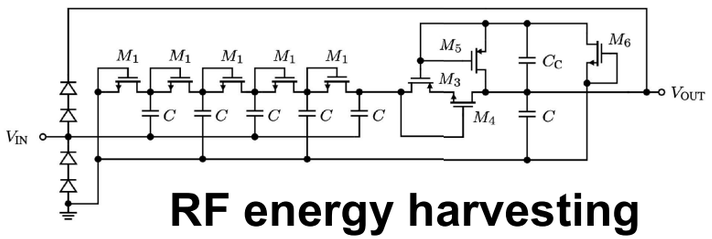




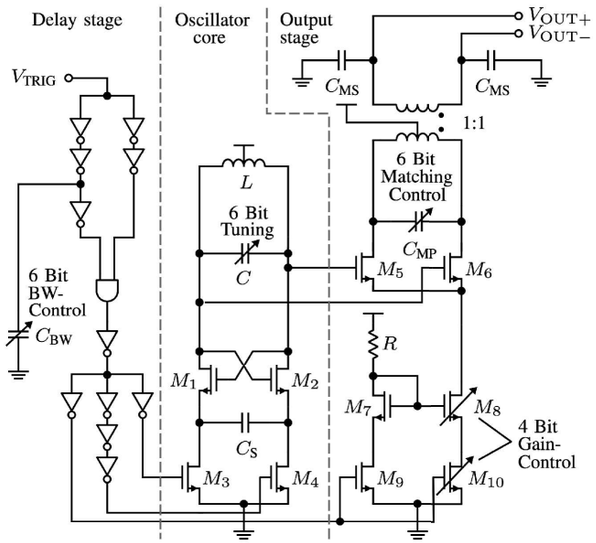
0.1Hz–120Hz Low-noise amplifier

JOHANNES KEPLER UNIVERSITY LINZ

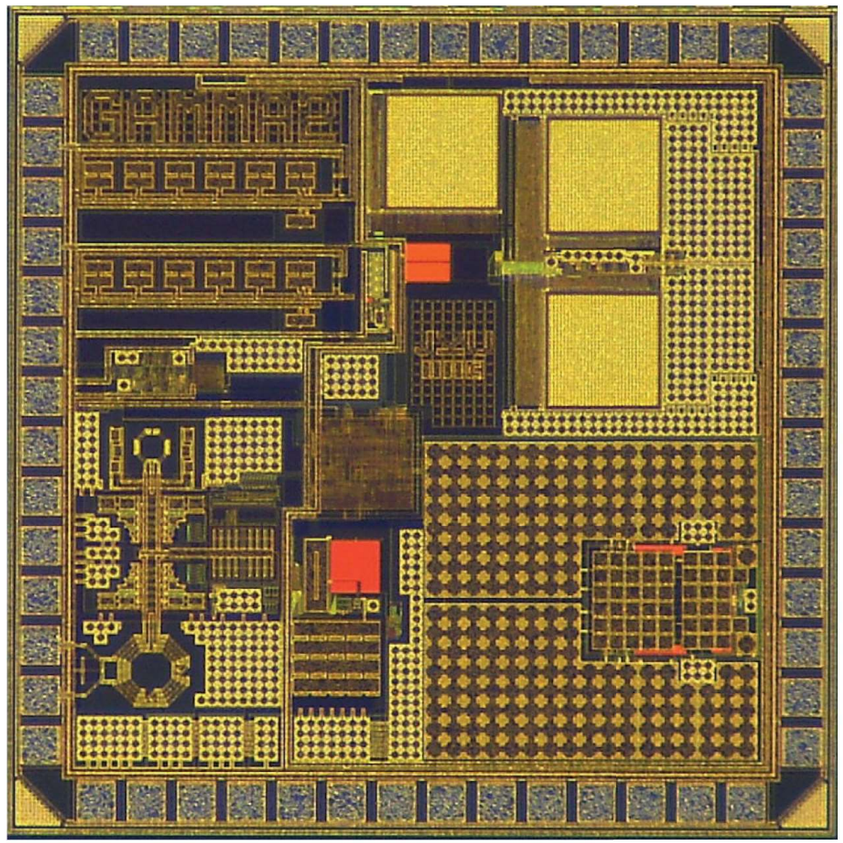
Ultra-Low Power Bio-Sensing SoC (closed-source): Low-noise amplifier, SAR ADC, UWB TX, RF harvester

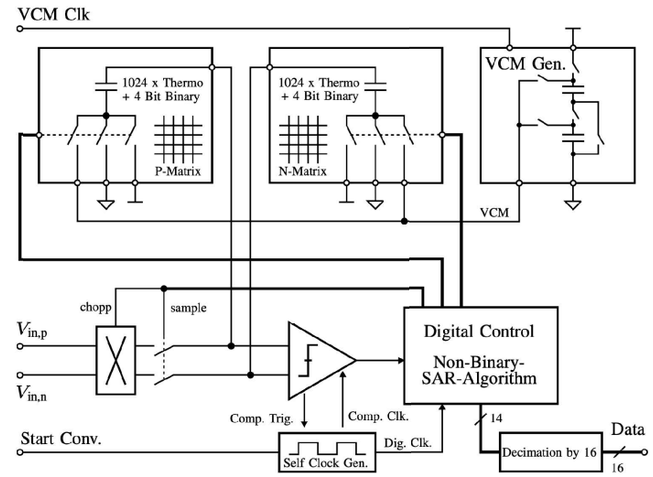


RF energy harvesting

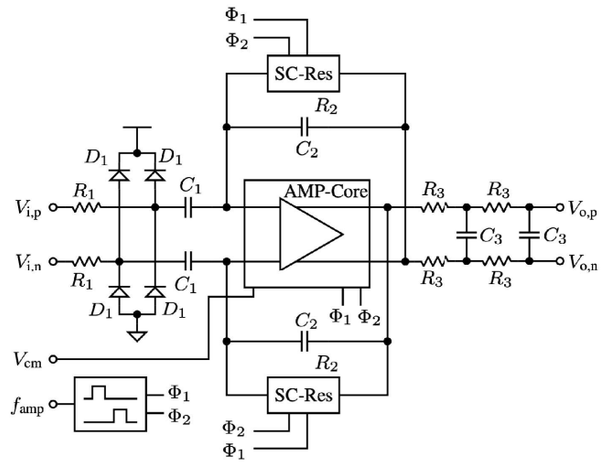


IR UWB transmitter





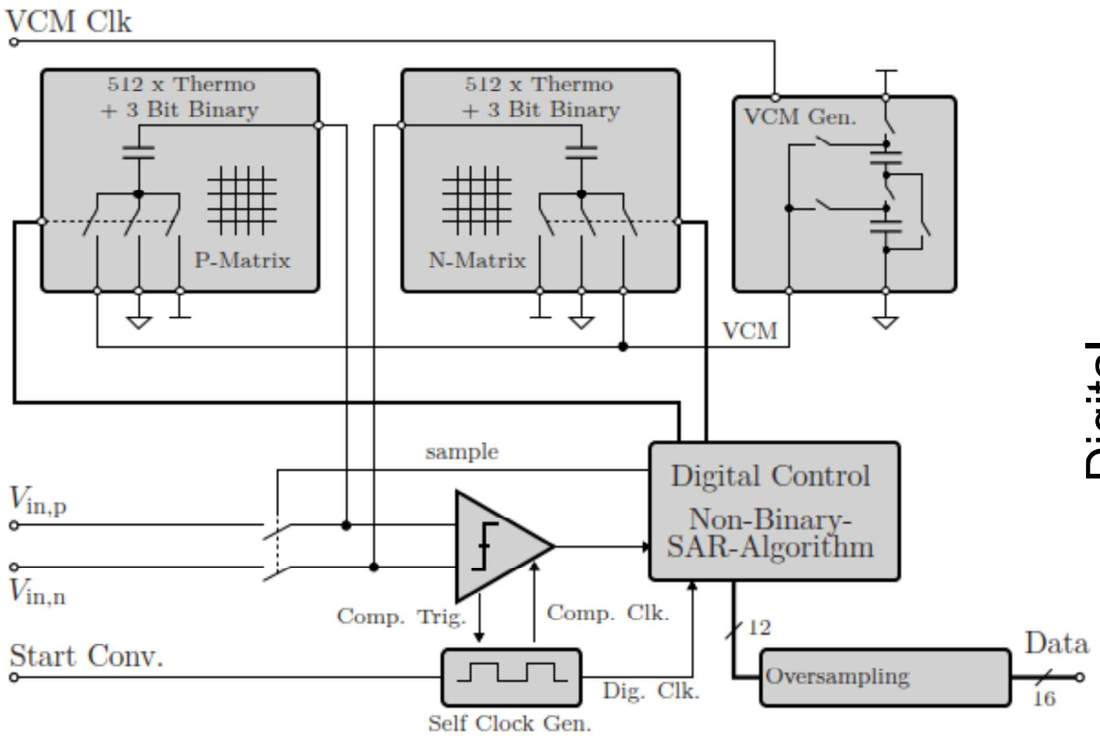
14b Self-clocked SAR ADC



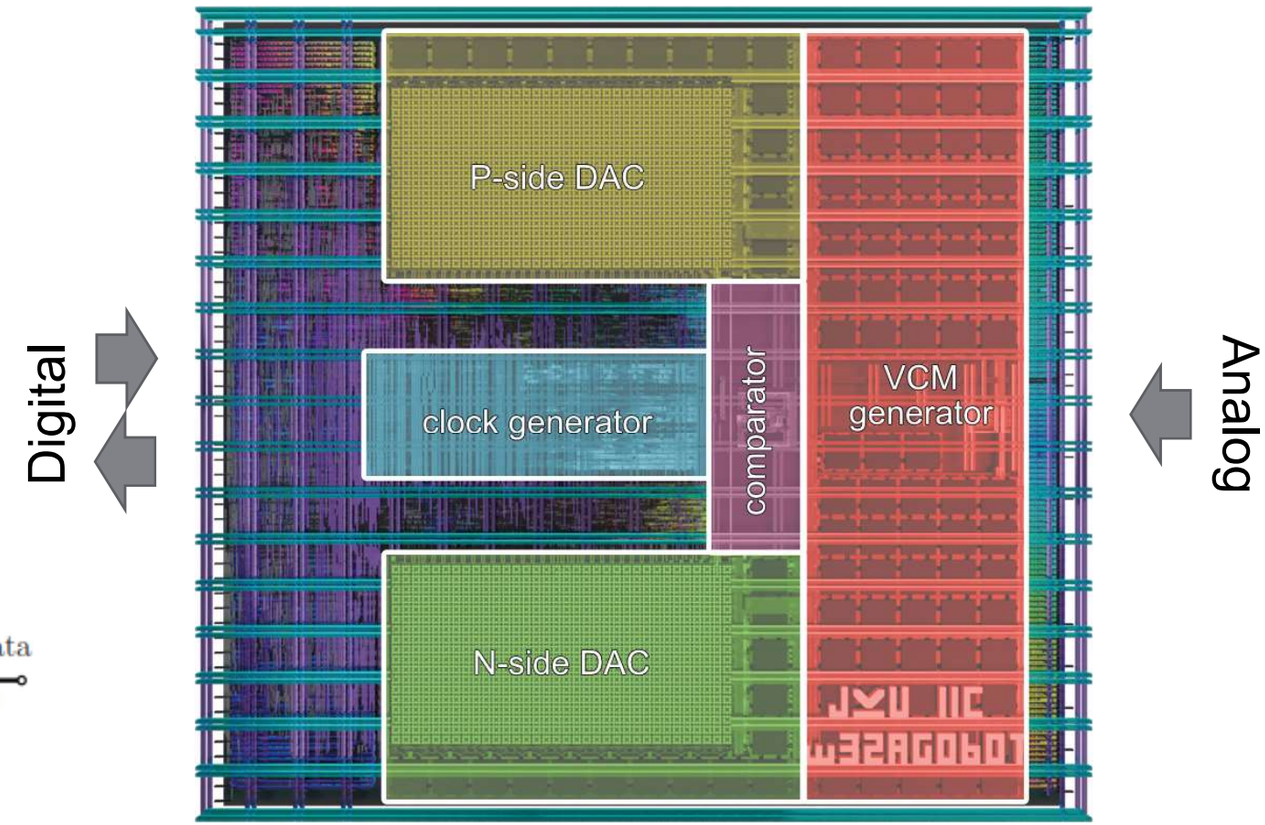
0.1Hz–120Hz Low-noise amplifier

JOHANNES KEPLER UNIVERSITY LINZ

SAR ADC (open-source)
https://github.com/iic-jku/SKY130_SAR-ADC1



SAR-ADC block diagram

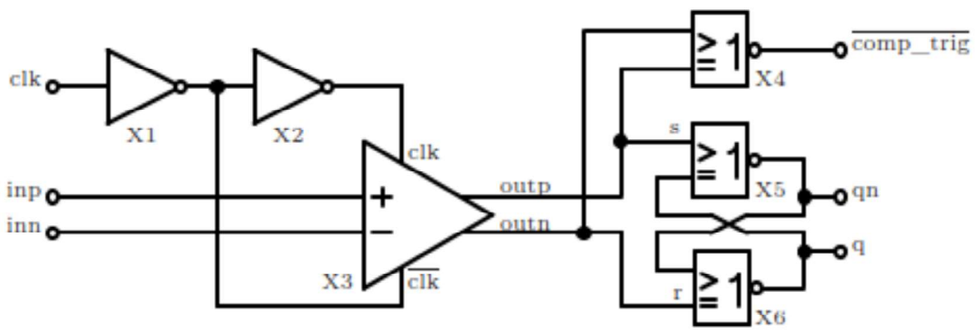


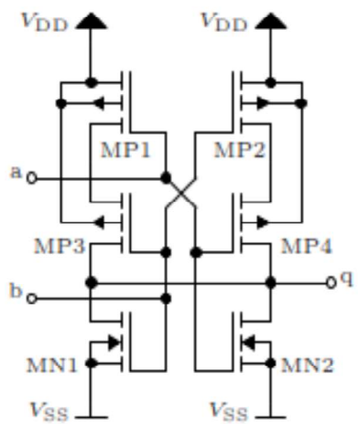
Floorplan of the SAR-ADC

S. Schmickl, T. Faseth, and H. Pretl, “An Untrimmed 14-bit Non-Binary SAR-ADC Using 0.37 fF-Capacitors in 180 nm for 1.1 µW at 4 kS/s,” *27th IEEE ICECS*, 2020, DOI:10.1109/icecs49266.2020.9294971.

SAR ADC (open-source)
Full-custom blocks like comparator

- Two-stage dynamic comparator
- Latched output, conversion-finished signal, symmetric NOR gates; custom capacitors using metal finger caps
- Design entry (**`xschem`**), layout (**`magic`**), analog simulation (**`ngspice`**), DRC/LVS/PEX (**`iic-drc.sh, iic-lvs.sh, iic-pex.sh`**)

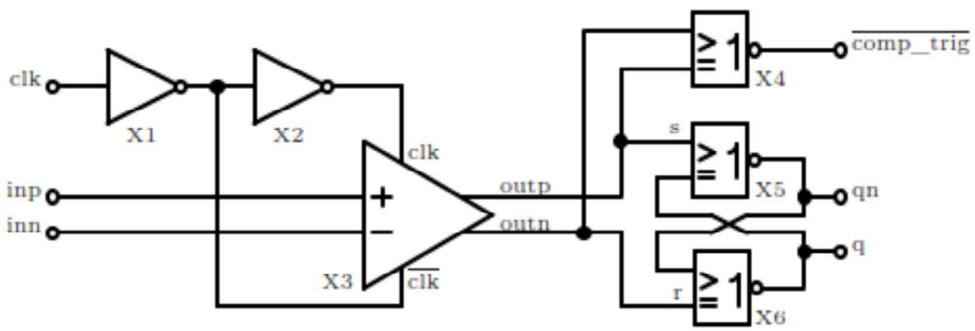


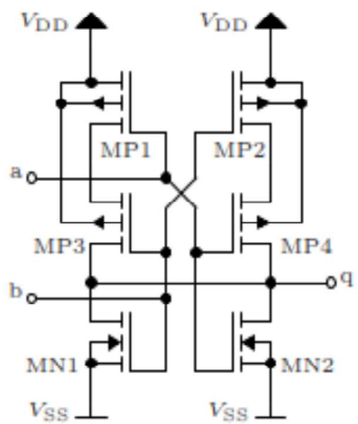


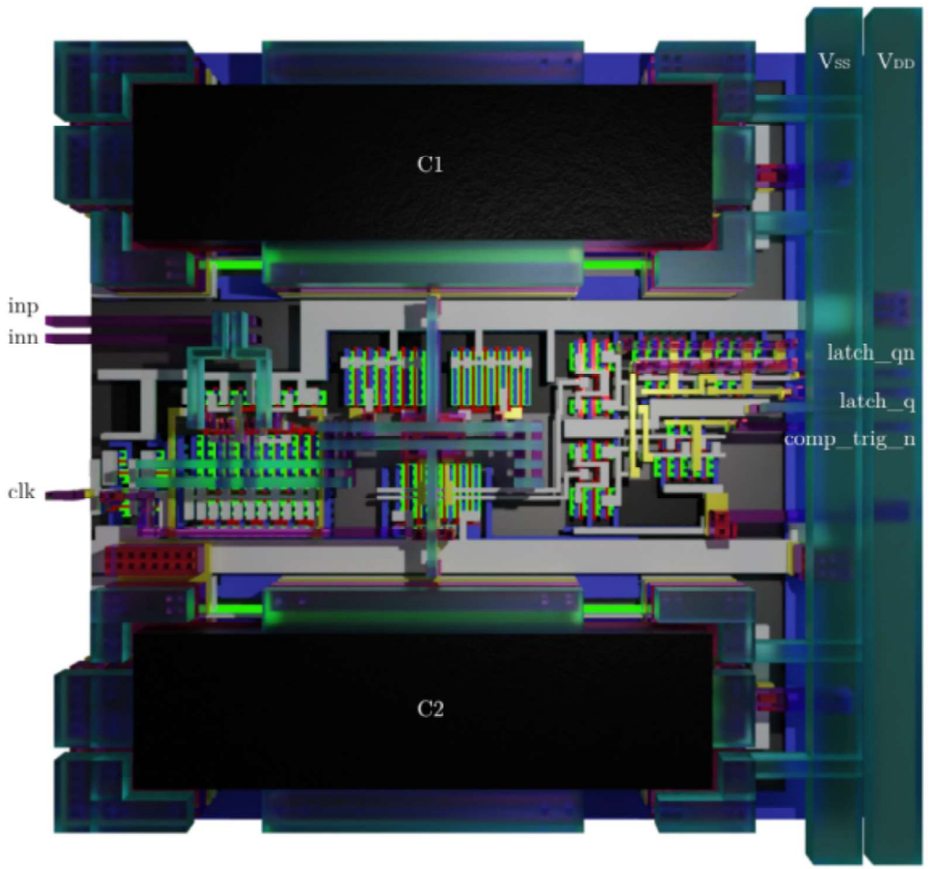
M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Full-custom blocks like comparator

- Two-stage dynamic comparator
- Latched output, conversion-finished signal, symmetric NOR gates; custom capacitors using metal finger caps
- Design entry (**`xschem`**), layout (**`magic`**), analog simulation (**`ngspice`**), DRC/LVS/PEX (**`iic-drc.sh, iic-lvs.sh, iic-pex.sh`**)



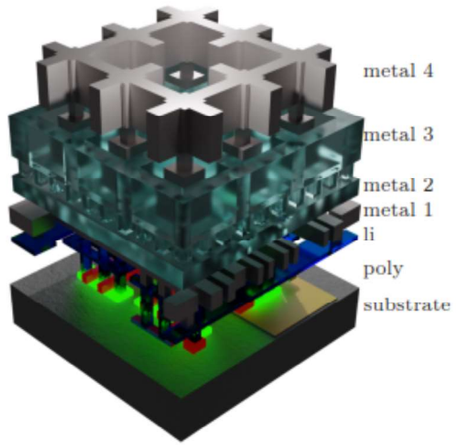


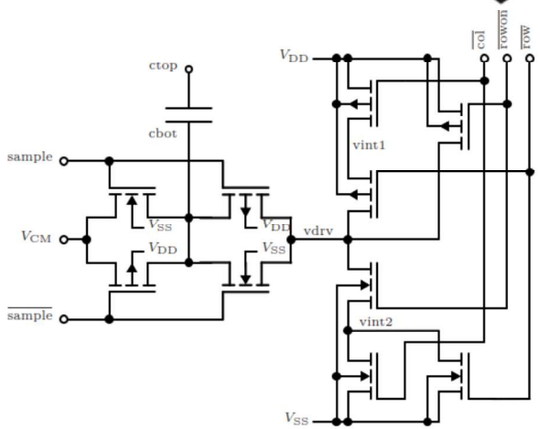


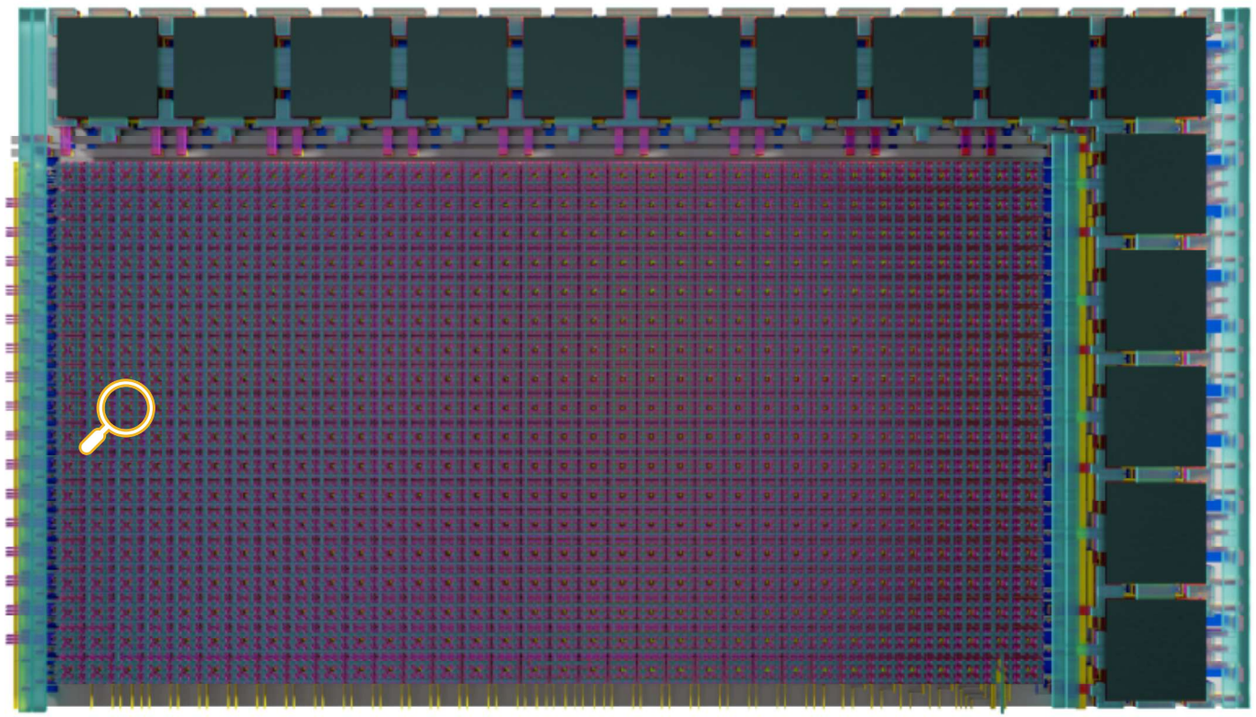
M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
12b (9b thermo, 3b binary) switched-capacitor DAC

- 25µm², 8 unit caps á 447aF
- Requires special LVS handling (`lvs_ignore` in **`xschem`**)



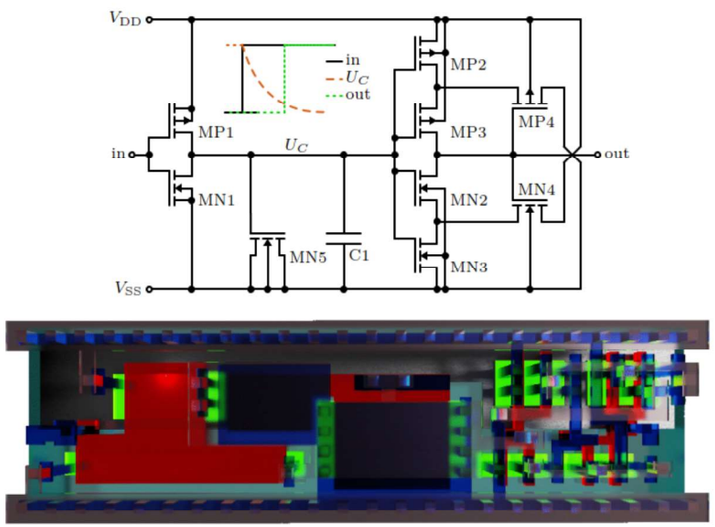




M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Custom standard cells (5ns delay)

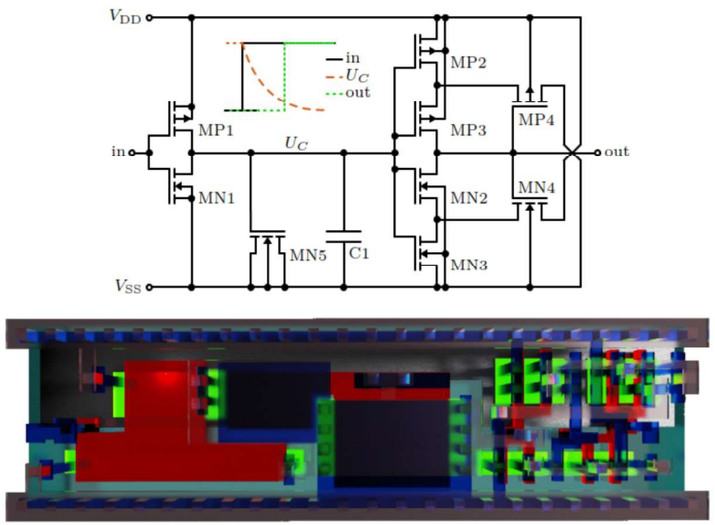
- Custom standard cell (using **`xschem`**, **`magic`**, **`ngspice`**) for increased delay (5ns)
- P&R of programmable delay generator implemented in Verilog, using **`openlane`**/ **`openroad`**, simulation with **`ngspice`**



M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

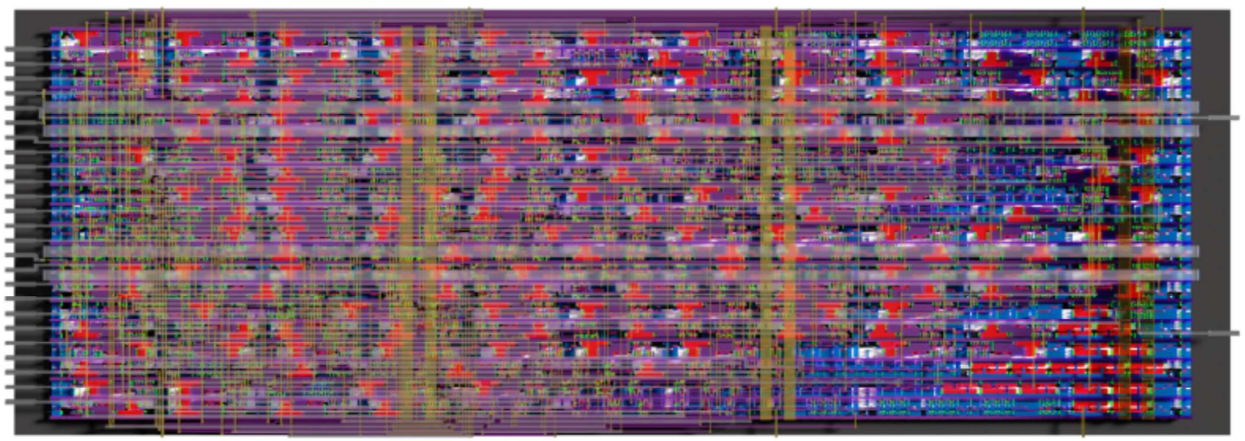
SAR ADC (open-source)
Custom standard cells (5ns delay)

- Custom standard cell (using **`xschem`**, **`magic`**, **`ngspice`**) for increased delay (5ns)
- P&R of programmable delay generator implemented in Verilog, using **`openlane`**/ **`openroad`**, simulation with **`ngspice`**




```
generate
   for(j=0;j<N_TIMES_5NS;j=j+1) begin
      sky130_mm_sc_hd_dlyPoly5ns delay_unit (.in(signal_w[j]), .out(signal_w[j+1]));
   end
endgenerate
sky130_fd_sc_hd__and2b_1 and_bypass_switch (.A_N(bypass_in),.B(in),.X(signal_w[0])); // 2 input and, A inverted
sky130_fd_sc_hd__mux2_1 out_mux (.A0(signal_w[N_TIMES_5NS]),.A1(in),.S(bypass_in),.X(out)); //2 input mux
```

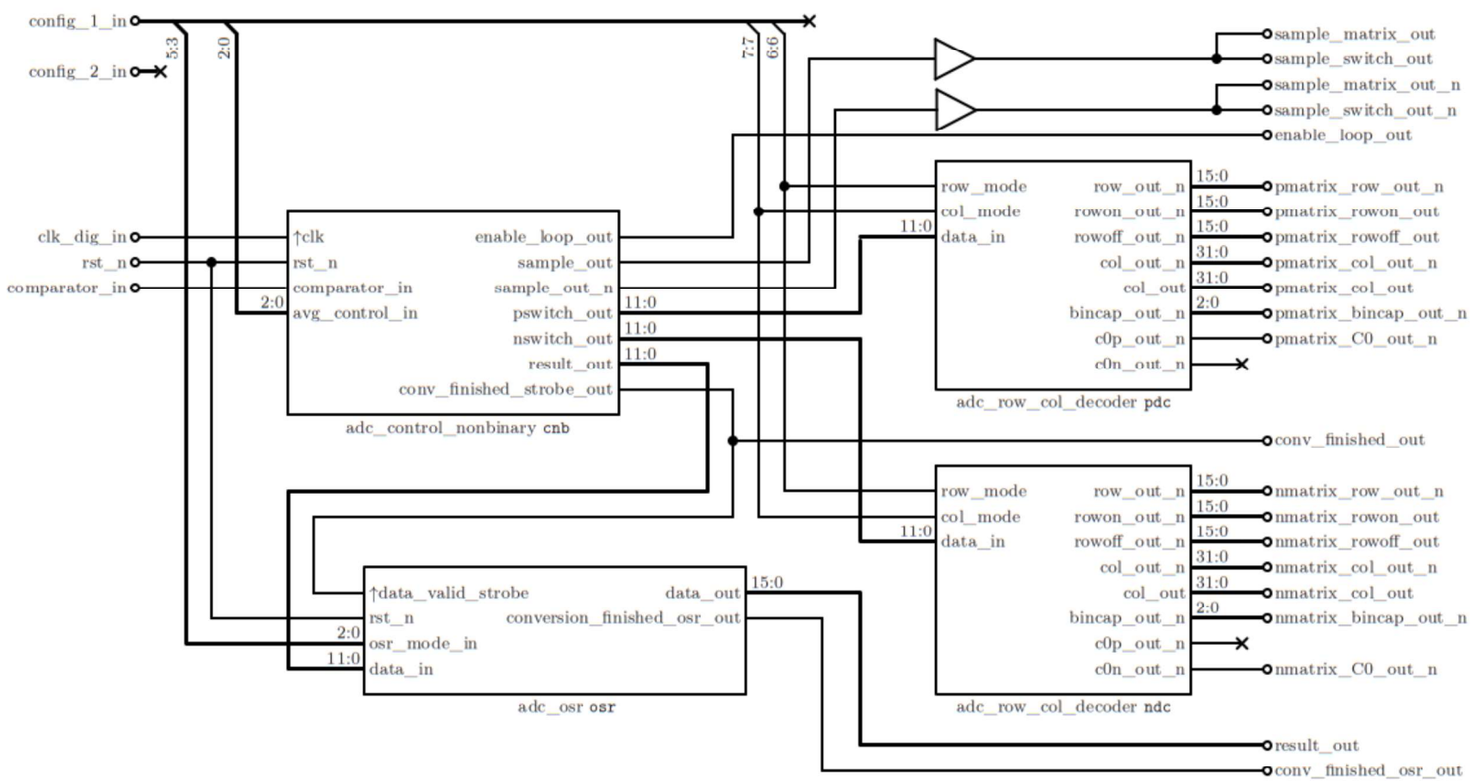
Gate level description



M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Digital design (state machine, decoders, FIR filter)

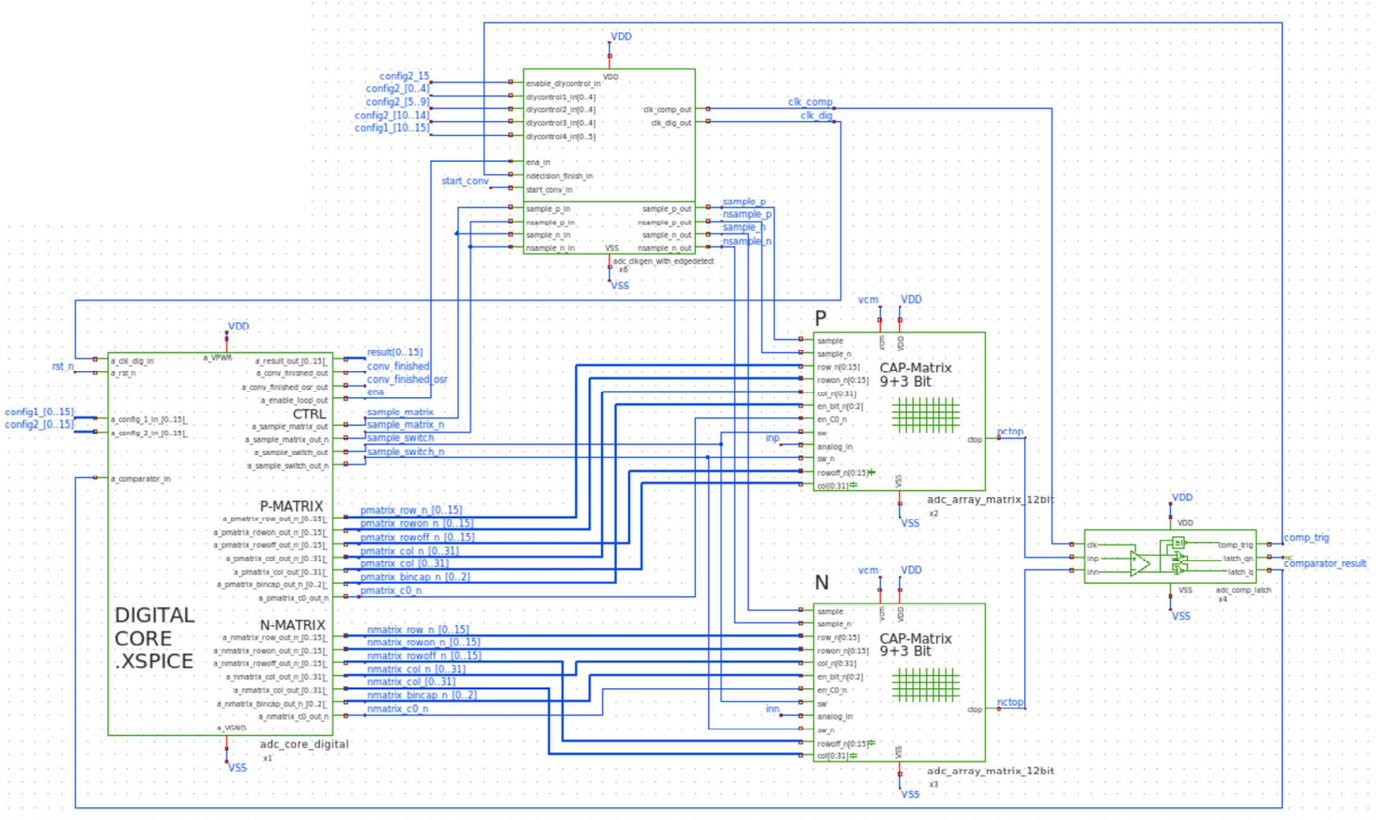
- Design of digital logic using **`iverilog`** for simulation and **`gtkwave`** for visualization of results, linting with **`verilator`**
- Verilog implementation of non-binary SAR algorithm, LSB averaging, row/column decoders including meander or common-centroid activation, decimation FIR filter (1/4/16/64/256)



M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Toplevel simulation circuit-level

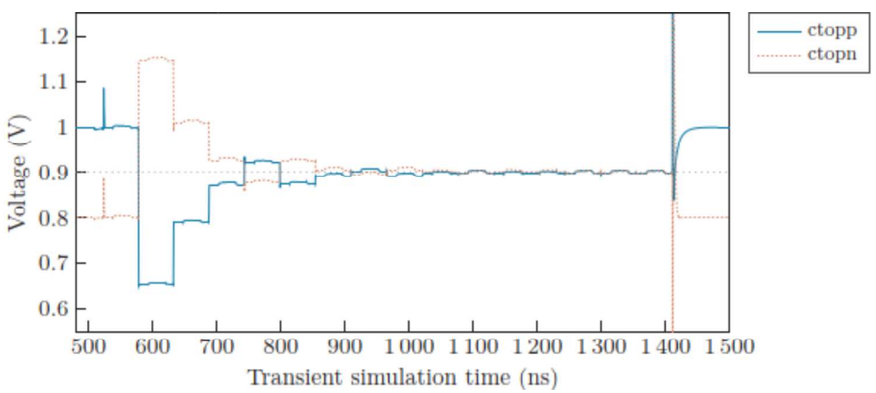
- Toplevel schematic and testbench in **`xschem`**
- Mixed-mode simulation in **`ngspice`**, using `.spice` netlists for analog blocks, `.xspice` for digital blocks
- Creation of `.xspice` from Verilog using synthesis with **`yosys`** and gate-to-xspice conversion using **`qflow`**

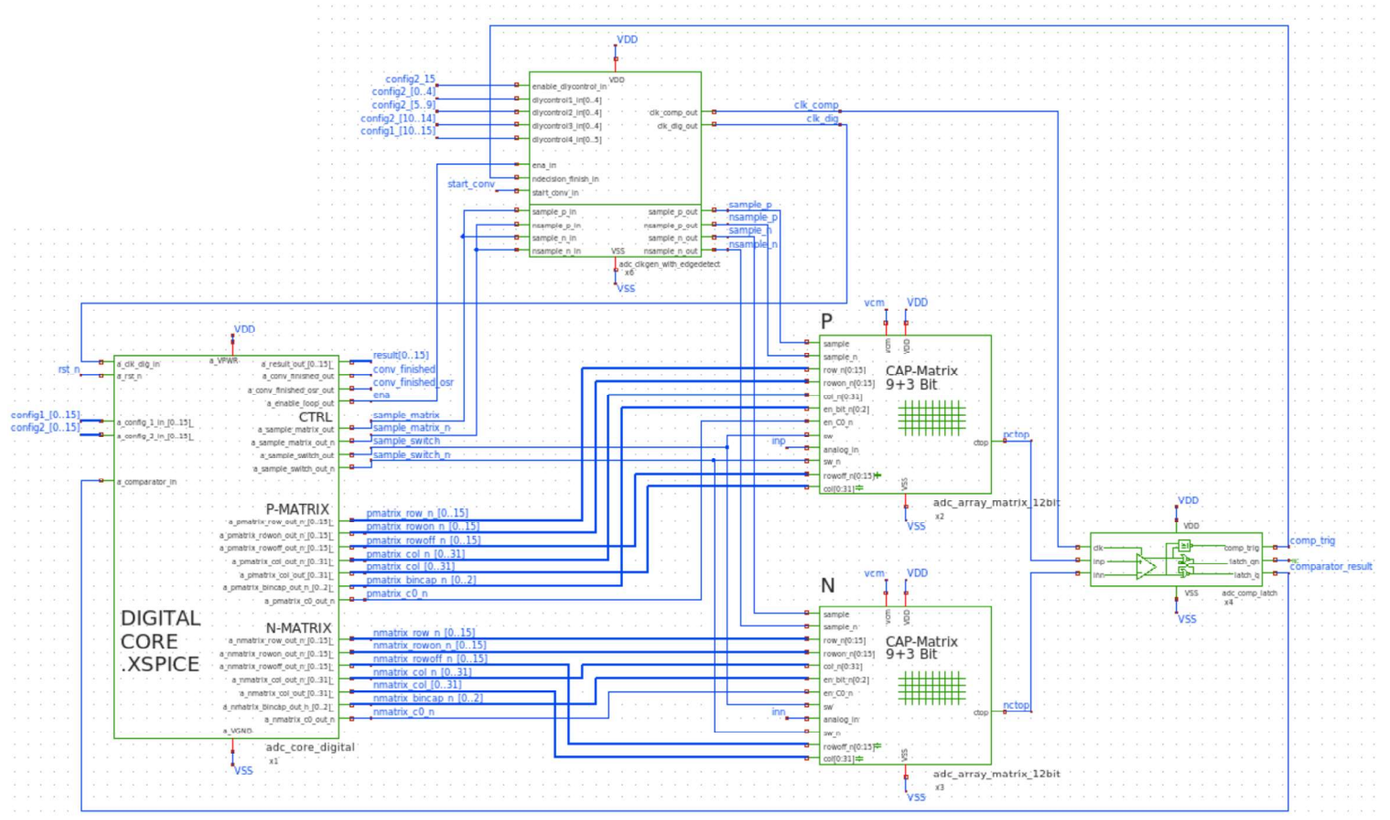


M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Toplevel simulation circuit-level

- Toplevel schematic and testbench in **`xschem`**
- Mixed-mode simulation in **`ngspice`**, using `.spice` netlists for analog blocks, `.xspice` for digital blocks
- Creation of `.xspice` from Verilog using synthesis with **`yosys`** and gate-to-xspice conversion using **`qflow`**

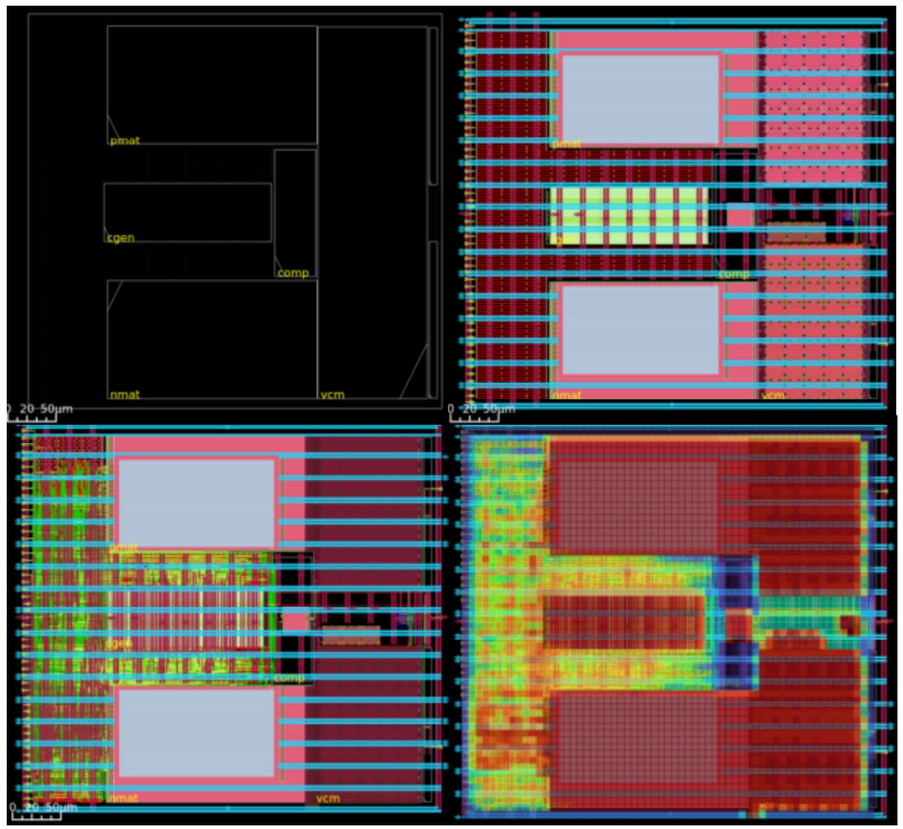




M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Top-level physical integration

- Toplevel physical integration using **`openlane`/`openroad`**
- PEX of top-level layout using **`iic-pex.sh`** (C-decoupled, C-coupled, RC)
- Post-layout simulation of C-extracted netlist using **`xyce`** (76k capacitors, 27k BSIM4, 250 resistors)

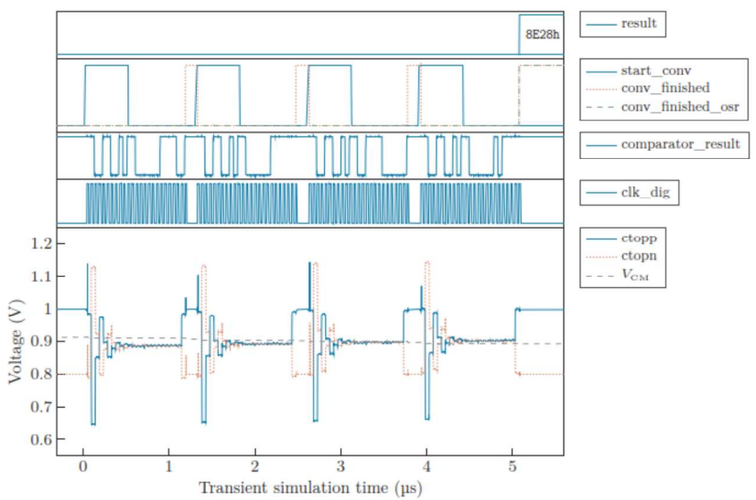


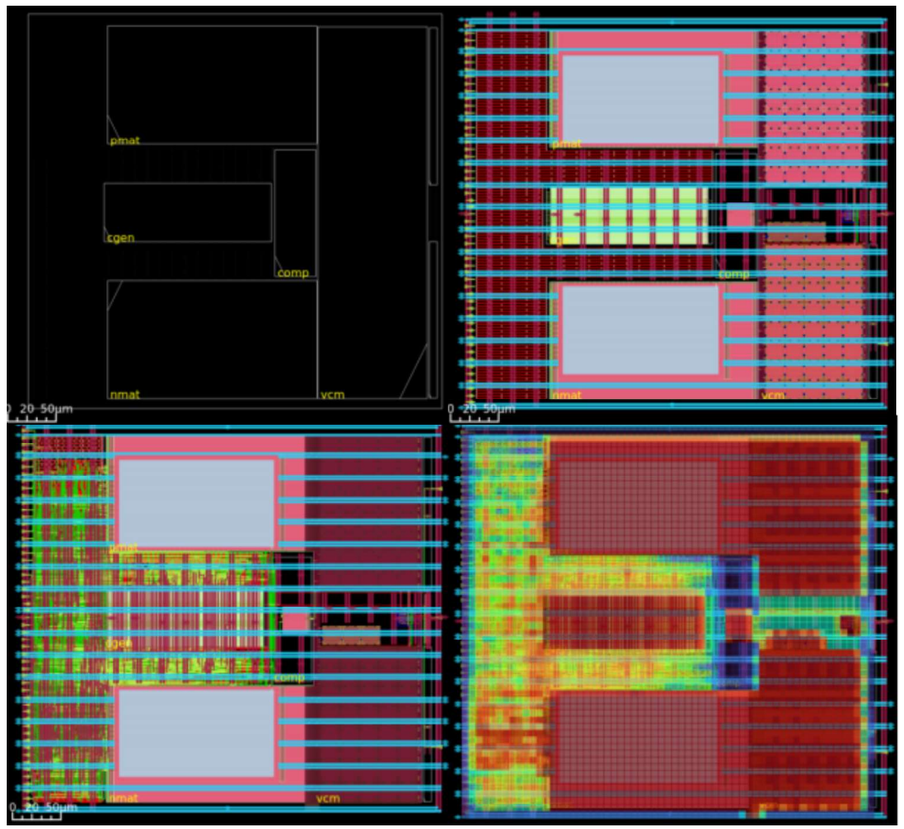
`openroad` views of the hardening process

M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

SAR ADC (open-source)
Top-level physical integration

- Toplevel physical integration using **`openlane/openroad`**
- PEX of top-level layout using **`iic-pex.sh`** (C-decoupled, C-coupled, RC)
- Post-layout simulation of C-extracted netlist using **`xyce`** (76k capacitors, 27k BSIM4, 250 resistors)





`openroad` views of the hardening process

M. Moser, “Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC utilizing the SKY130 Open-Source Technology,” MA thesis, 2023, https://digital.obvsg.at/urn/urn:nbn:at:at-ubl:1-62352, https://github.com/iic-jku/SKY130_SAR-ADC1

Agenda

- Introduction
- Open-source IC design environment
- IIC designs in open-source
- **Conclusion**

Open-Source EDA:
Missing pieces and open questions

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither `ngspice` nor `xyce` supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).
4. **Fewer interface breaks, better QA** of the various components: Naturally, since many different SW projects, but maybe it can be minimized with an overarching regression suite (we definitely need to implement regression tests for our Docker image build).

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).
4. **Fewer interface breaks, better QA** of the various components: Naturally, since many different SW projects, but maybe it can be minimized with an overarching regression suite (we definitely need to implement regression tests for our Docker image build).
5. Support of **EM structures**: EM simulators (with tech files), plus S-parameter handling in simulators.

Open-Source EDA: Missing pieces and open questions

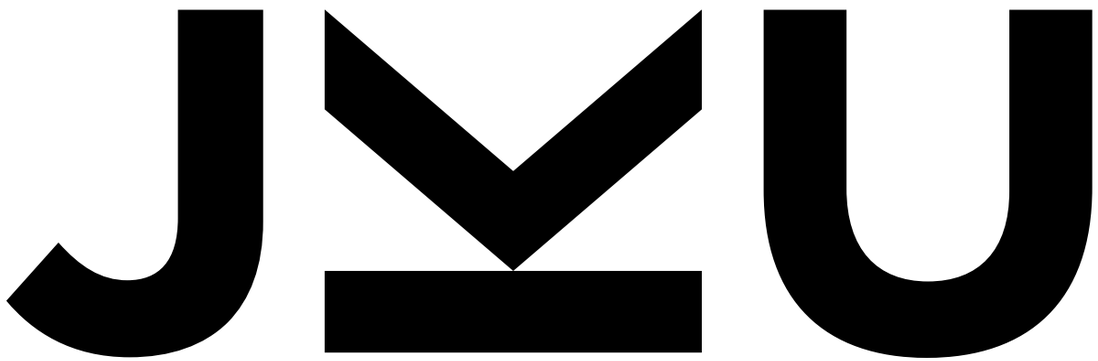
1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).
4. **Fewer interface breaks, better QA** of the various components: Naturally, since many different SW projects, but maybe it can be minimized with an overarching regression suite (we definitely need to implement regression tests for our Docker image build).
5. Support of **EM structures**: EM simulators (with tech files), plus S-parameter handling in simulators.
6. **Magic support?** Yes, please, this is needed!

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).
4. **Fewer interface breaks, better QA** of the various components: Naturally, since many different SW projects, but maybe it can be minimized with an overarching regression suite (we definitely need to implement regression tests for our Docker image build).
5. Support of **EM structures**: EM simulators (with tech files), plus S-parameter handling in simulators.
6. **Magic support?** Yes, please, this is needed!
7. **GUI for an integrated toolset?** Not required; scripting is highly efficient and allows automation (like regression testing, and continuous integration).

Open-Source EDA: Missing pieces and open questions

1. **Large-signal noise simulation**: Neither **`ngspice`** nor **`xyce`** supports large-signal noise analysis, but this is highly desired for mixed-signal (switching circuits) or RF/mm-wave (oscillators, mixers).
2. **Accurate PEX and fast analog simulators**: Need to simulate large netlists for serious designs.
3. **Scan insertion and pattern generation** for digital: Not (so) important for academic IC, but once commercial ICs are developed, this is needed (`fault` is available but not tested yet by us).
4. **Fewer interface breaks, better QA** of the various components: Naturally, since many different SW projects, but maybe it can be minimized with an overarching regression suite (we definitely need to implement regression tests for our Docker image build).
5. Support of **EM structures**: EM simulators (with tech files), plus S-parameter handling in simulators.
6. **Magic support?** Yes, please, this is needed!
7. **GUI for an integrated toolset?** Not required; scripting is highly efficient and allows automation (like regression testing, and continuous integration).
8. Need **update-to-date documentation, tutorials, and well-done examples**.



JOHANNES KEPLER
UNIVERSITY LINZ