JOHANNES KEPLER
UNIVERSITY LINZ

Design of a Low-Power 12-bit Non-Binary Charge-Redistribution SAR-ADC Utilizing the SKY130 Open-Source Technology

Manuel Moser, Patrick Fath, Harald Pretl
Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
2023-07-11 FSiC2023 (Paris)

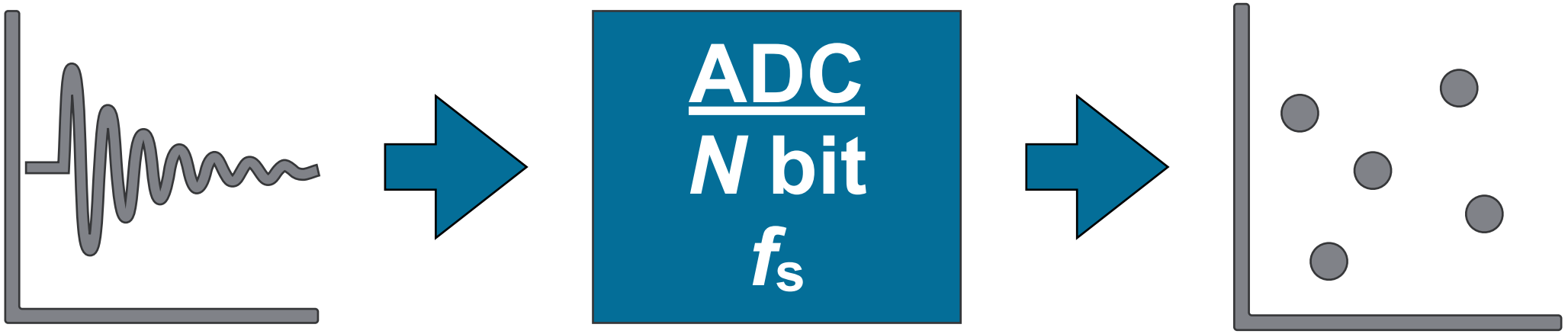
JOHANNES KEPLER
UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

The presented ADC IP is not yet silicon-proven; all results are based on simulation!

Agenda

- Introduction
- Open-source IC design environment
- ADC design details
- Conclusion

What is an Analog-to-Digital Converter (ADC)?



Signal
(voltage, current, charge, etc.)

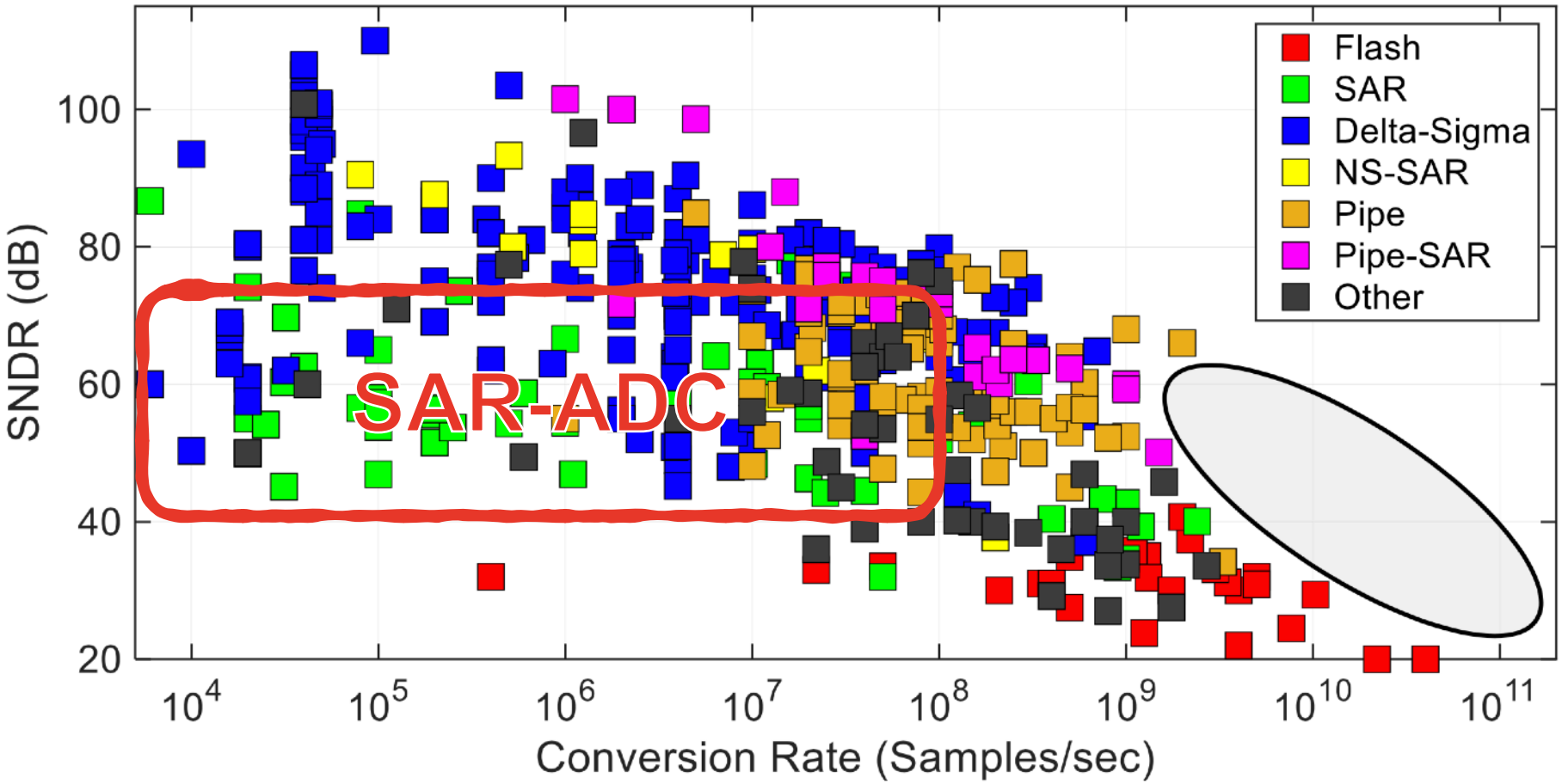
<u>Continuous</u> in value and time

Digital representation of signal compared to a reference

<u>Discrete</u> in value and time
Sampled each $\Delta T_s = 1 / f_s$
2^N digital values

ADC Types: SNDR vs. Conversion Speed vs. Complexity

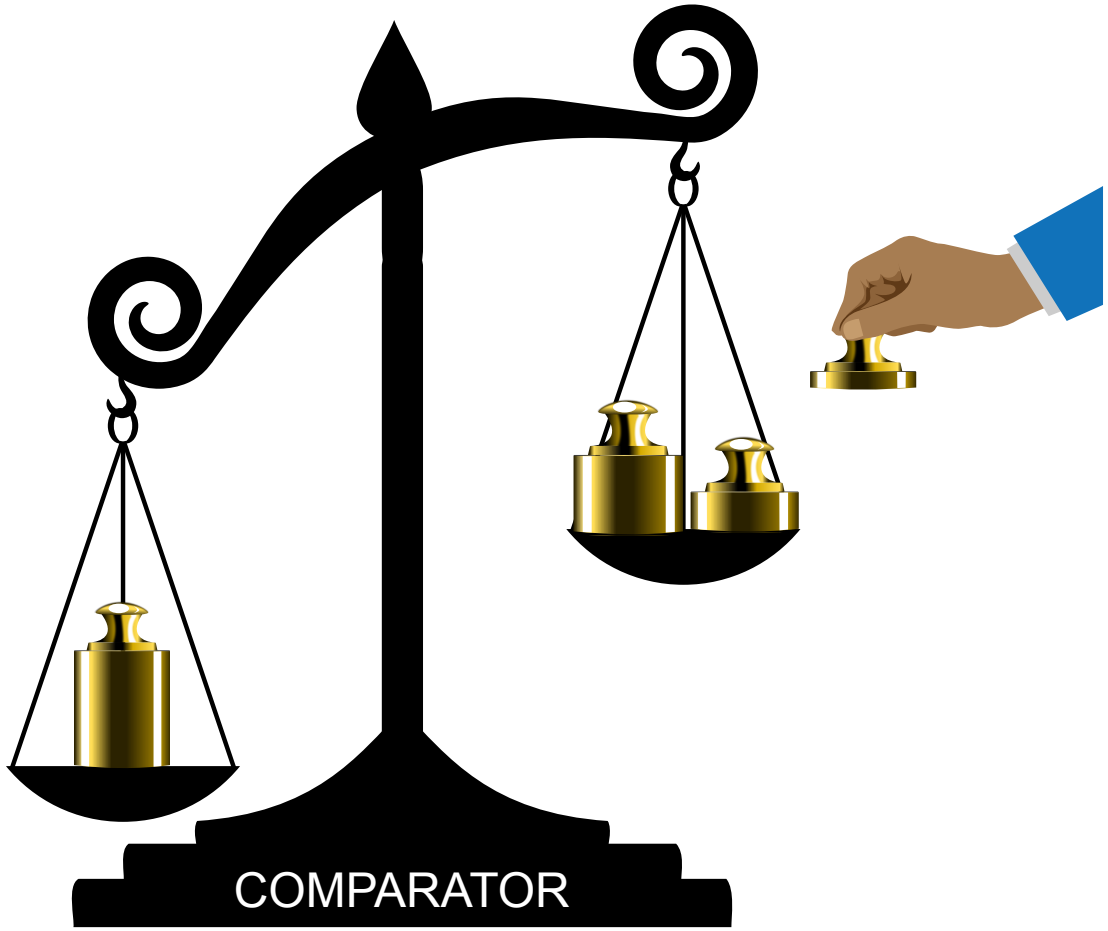
$\propto 2^N$



f_S

Source: B. Murmann, "Introduction to ADCs/DACs: Metrics, Topologies, Trade Space, and Applications," ISSCC Tutorial, 2022.

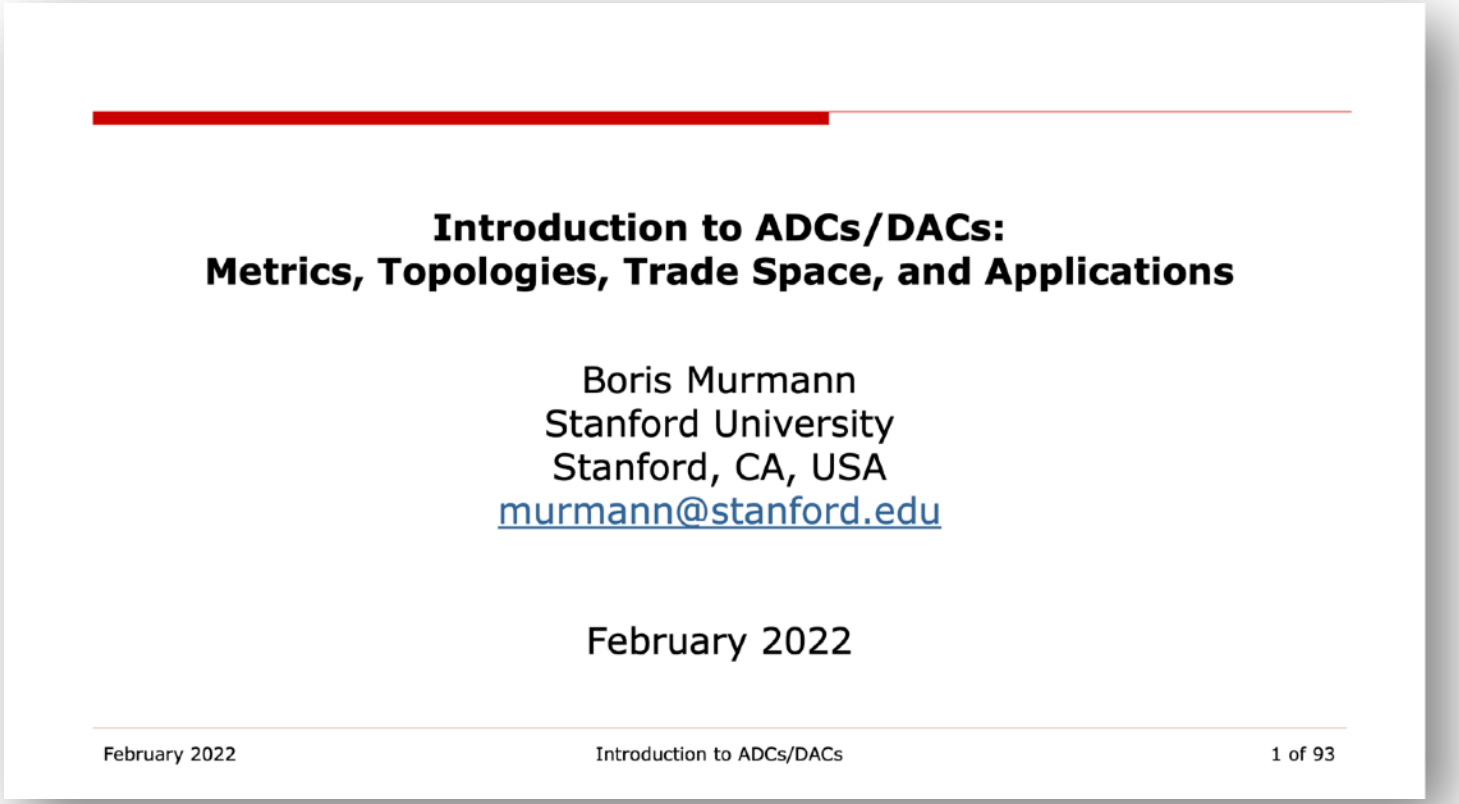
Successive Approximation Register Analog-to-Digital Converter (SAR-ADC)



Source: pixabay.com

- SAR-ADC works like scales
- Something with unknown weight is taken and held constant („sampled“), and known weights are successively added or subtracted until the scales are in balance.
- In a binary SAR-ADC, these weights are binary-scaled (but could also be otherwise).
- Taking a measurement requires multiple steps (so need a clock higher than the sampling rate).

You Want to Lean More About ADC/DAC?



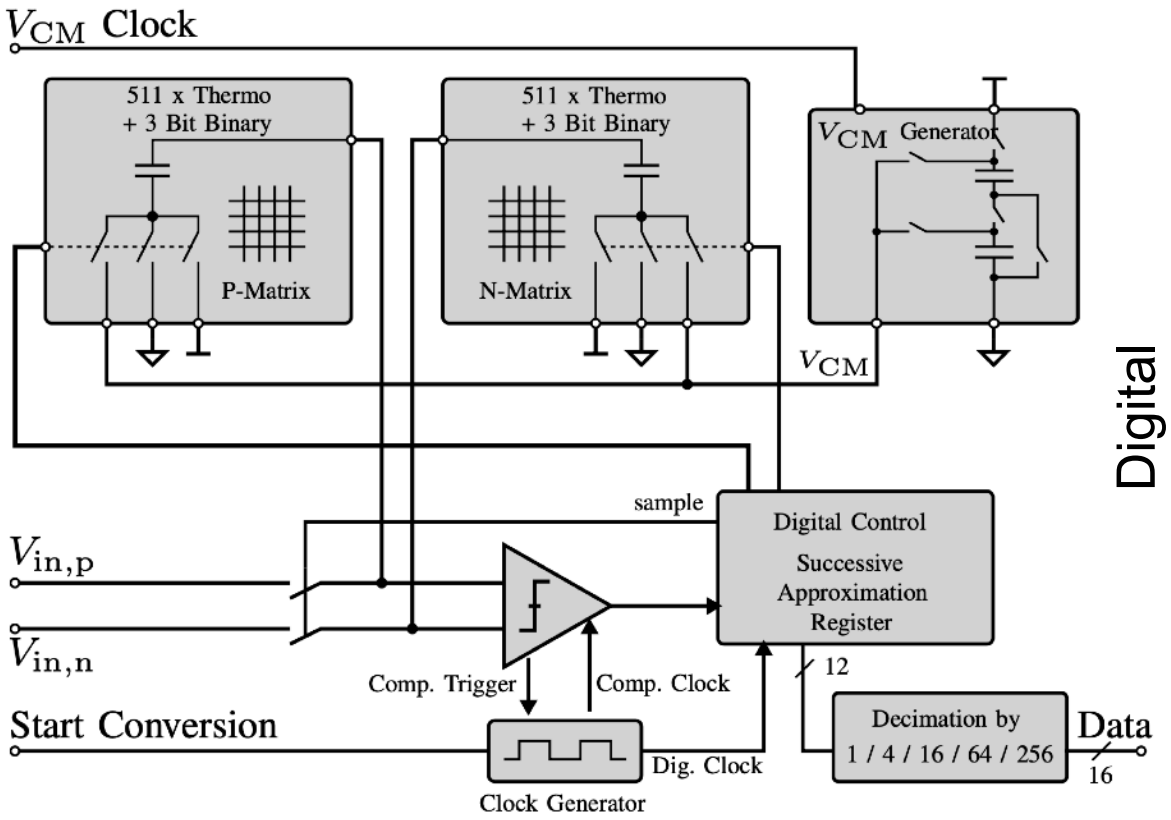
Introduction to ADCs/DACs:
Metrics, Topologies, Trade Space, and Applications

Boris Murmann
Stanford University
Stanford, CA, USA
murmann@stanford.edu

February 2022

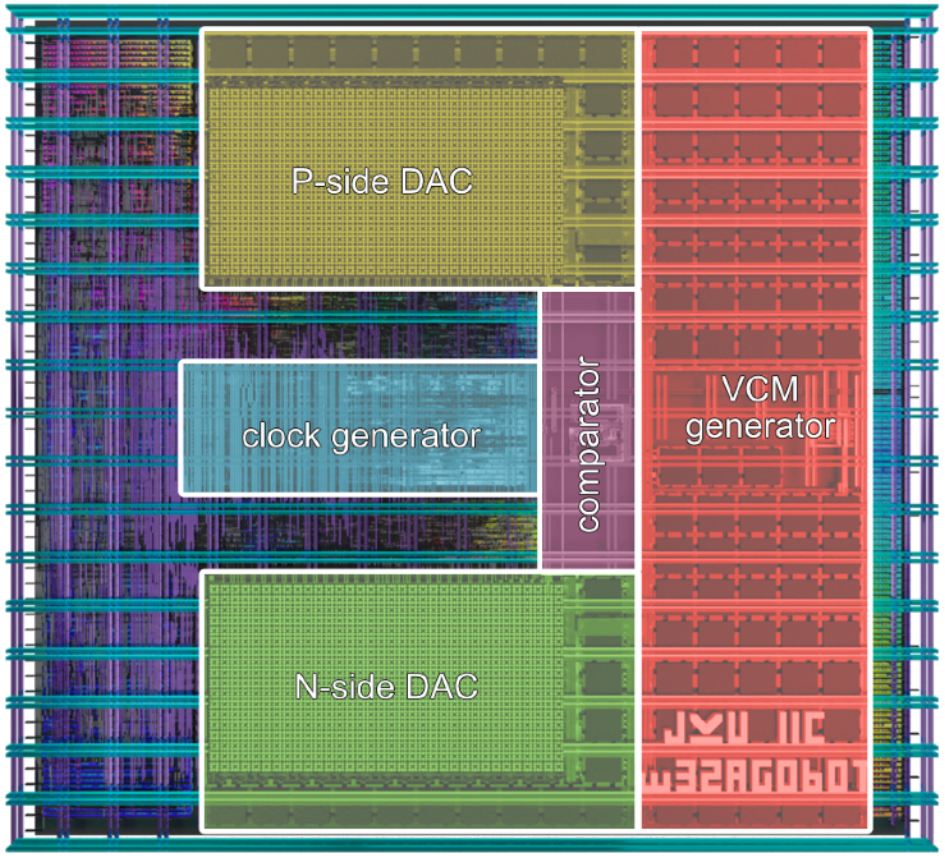
B. Murmann, "Introduction to ADCs/DACs: Metrics, Topologies, Trade Space, and Applications," ISSCC 2022 Tutorial. Available: https://github.com/bmurmann/ADC-survey/blob/main/pdf/ISSCC2022-Short%20Course-Murmann.pdf.

SAR-ADC [1]
https://github.com/iic-jku/SKY130_SAR-ADC1



SAR-ADC block diagram

Digital



Analog

Floorplan of the SAR-ADC

[1] 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,” ICECS, 2020.

Agenda

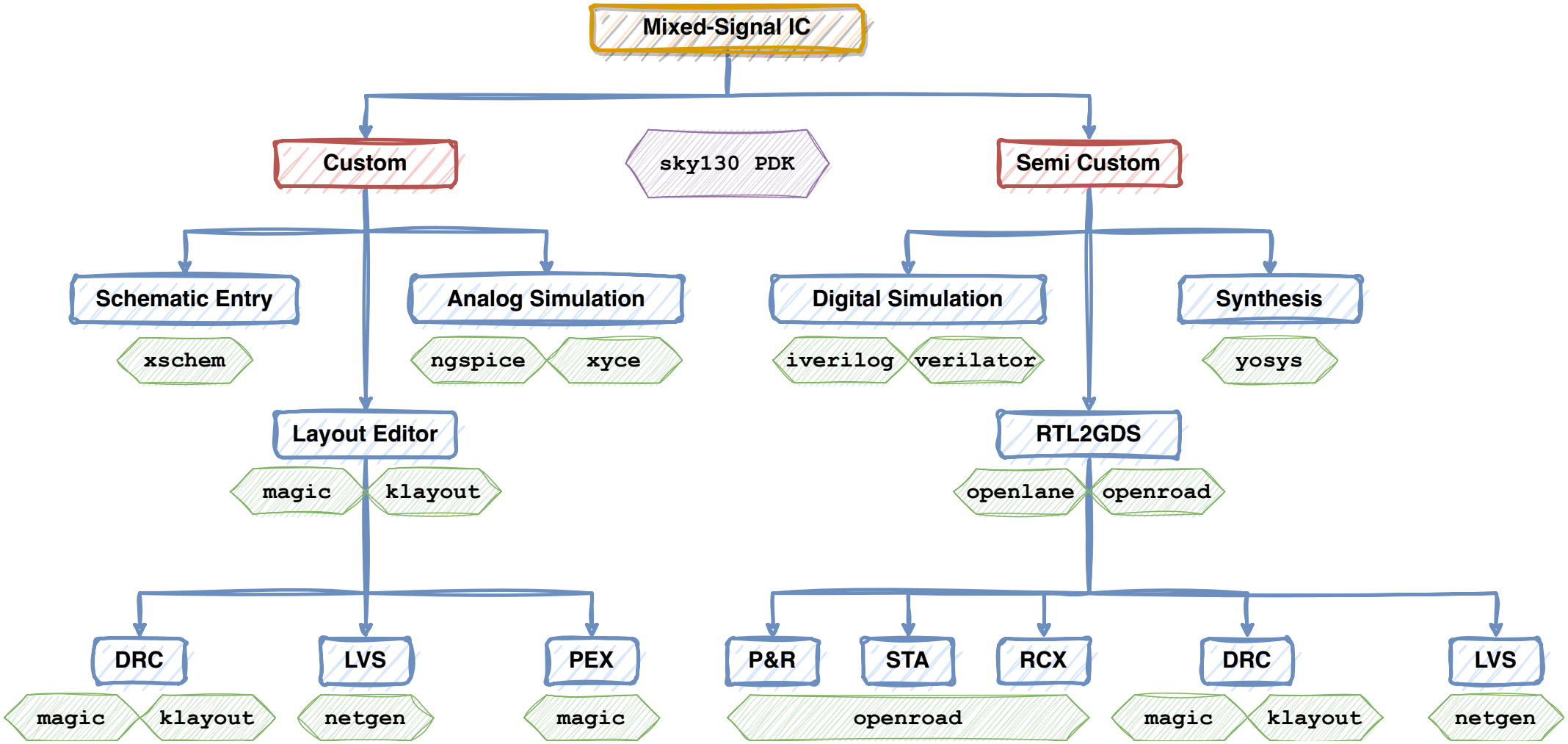
- Introduction

- **Open-source IC design environment**

- **ADC design details**

- **Conclusion**

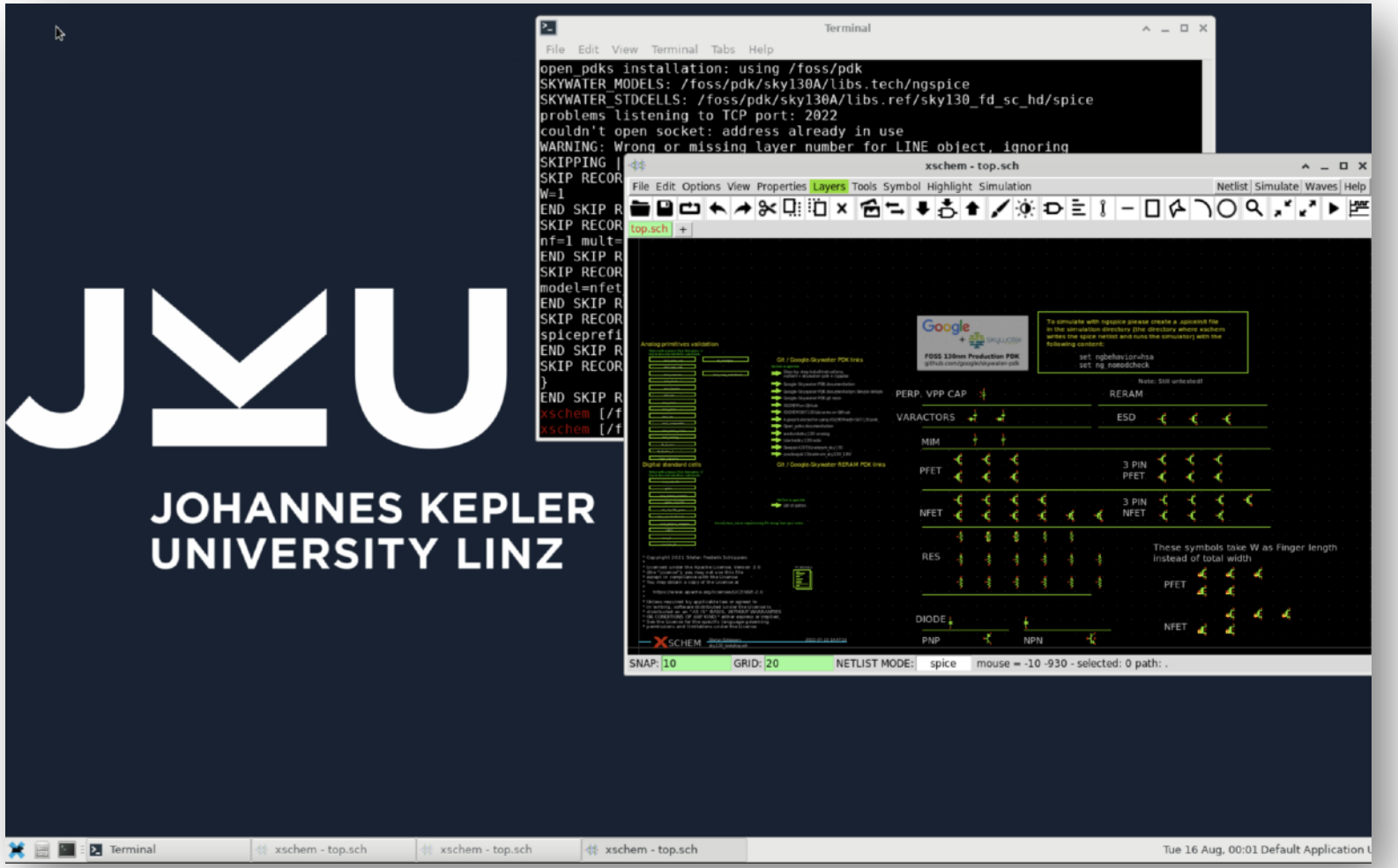
IC Design Tools Used in This Project



All-in-One Docker Image

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

Forked from efabless.com



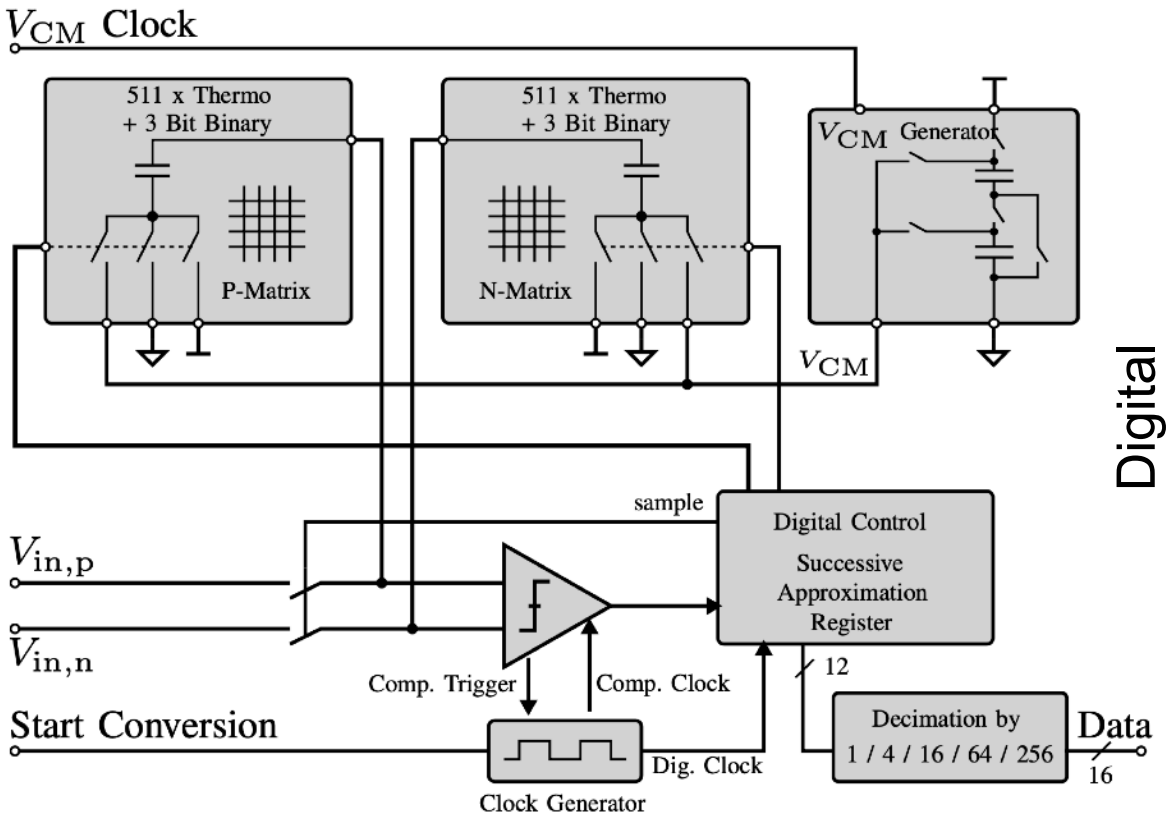
https://github.com/iic-jku/IIC-OSIC-TOOLS
Currently Installed Tools & PDKs (list growing...)

- **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)

Agenda

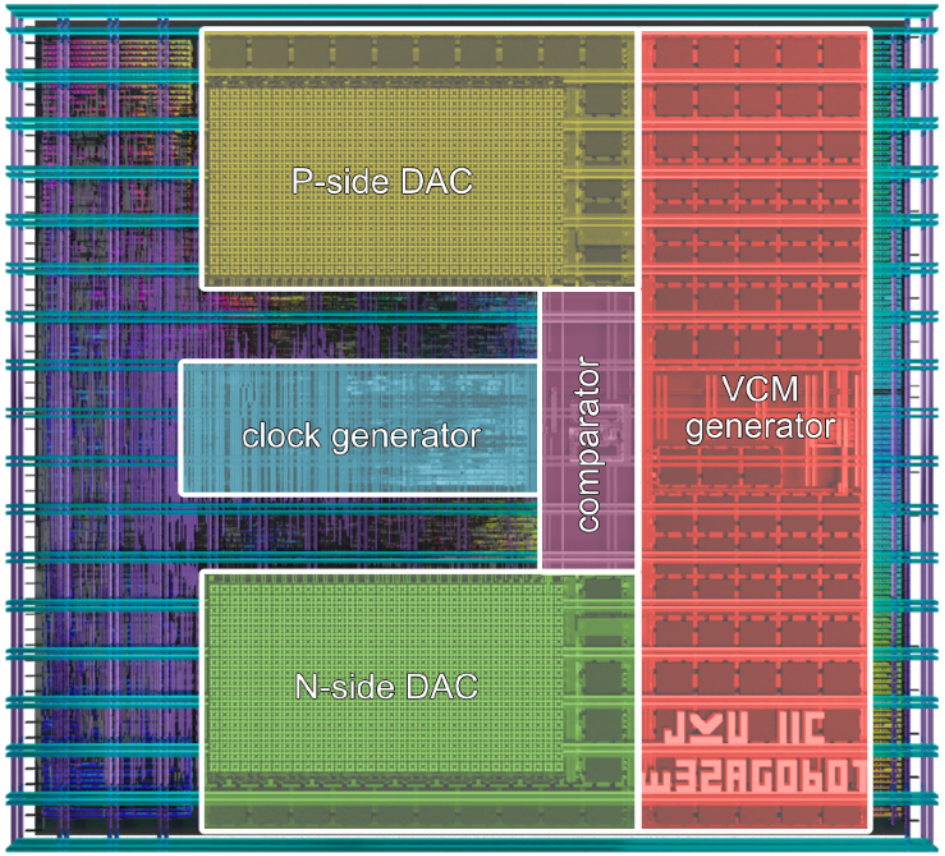
- Introduction
- Open-source IC design environment
- **ADC design details**
- **Conclusion**

SAR-ADC [1]
https://github.com/iic-jku/SKY130_SAR-ADC1



SAR-ADC block diagram

Digital



Analog

Floorplan of the SAR-ADC

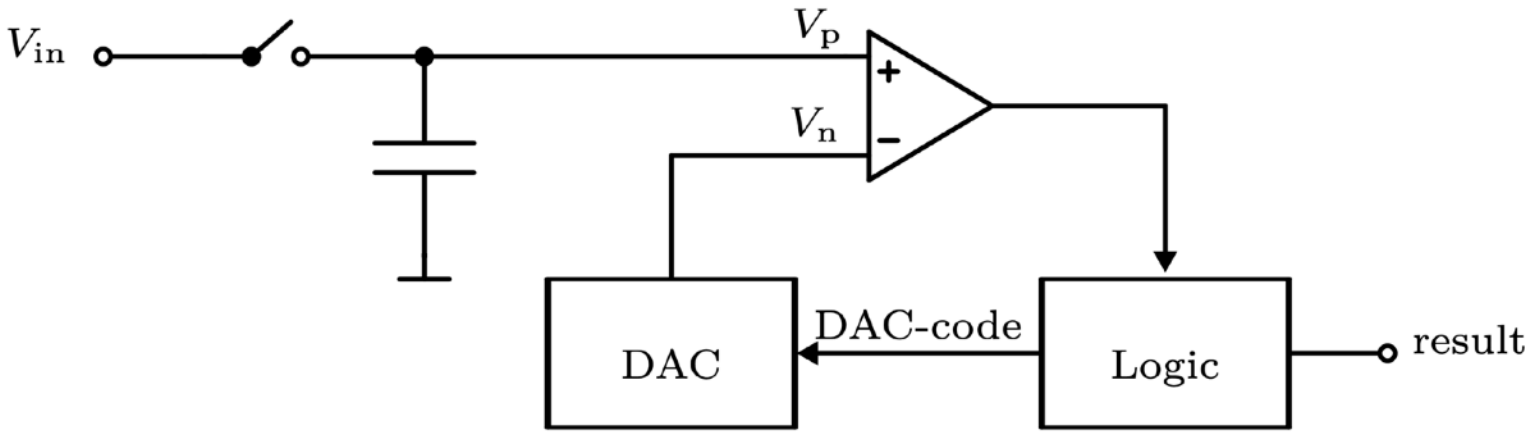
[1] S. Schmickl, T. Faseth, and H. Pretl, “An Untrimmed 14-bit Non-Binary SAR-ADC Using 0.37 fF-Capacitors in 180 nm for 1.1 µW at 4 kS/s,” ICECS, 2020.

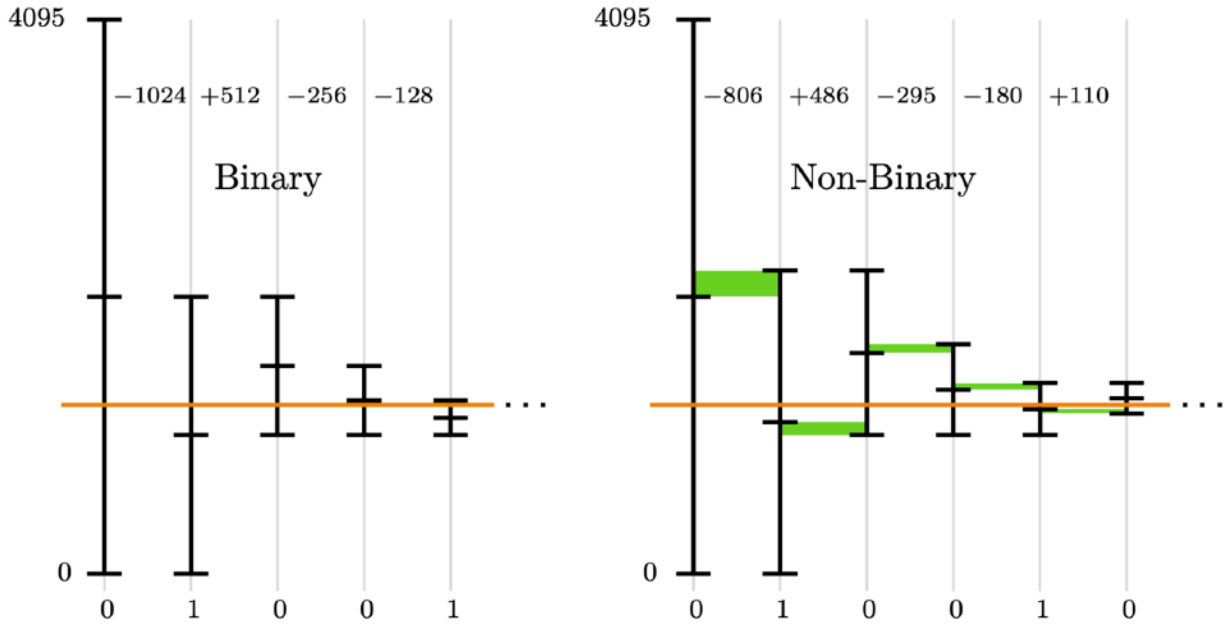
SAR-ADC — Features

- Versatile, easy-to-use IP: **SAR-ADC**, **28S/s (16b) to 1.2MS/s (12b), 0.18mm²**, **no calibration**
- **Fully-differential** design: Robust against interference
- **Class-B operation**: No quiescent current, no bias generation
- **Self-clocked**: No high-speed clock needed; configurable self-clock delay
- **Non-binary**: Robust against disturbance and comparator noise
- Integrated **decimation filter**: Supports OSR=1/4/16/64/256
- **LSB averaging**: Trade-off speed vs. comparator noise
- **Supply-referenced**: No bias generation and bias buffers needed
- Integrated **common-mode voltage generator**: Based on charge-pump
- **Selectable DAC activation** style: To trade off INL vs. DNL
- **Published** with the Apache 2.0 license on GitHub [1], **documented** in a master thesis [2]

[1] https://github.com/iic-jku/SKY130_SAR-ADC1
[2] https://epub.jku.at/obvulihs/content/titleinfo/8694768

SAR-ADC — Overview





Binary search:

Easy to realise
Very simple state machine
No error correction/redundancy
N steps to get result

Non-binary search [1]:

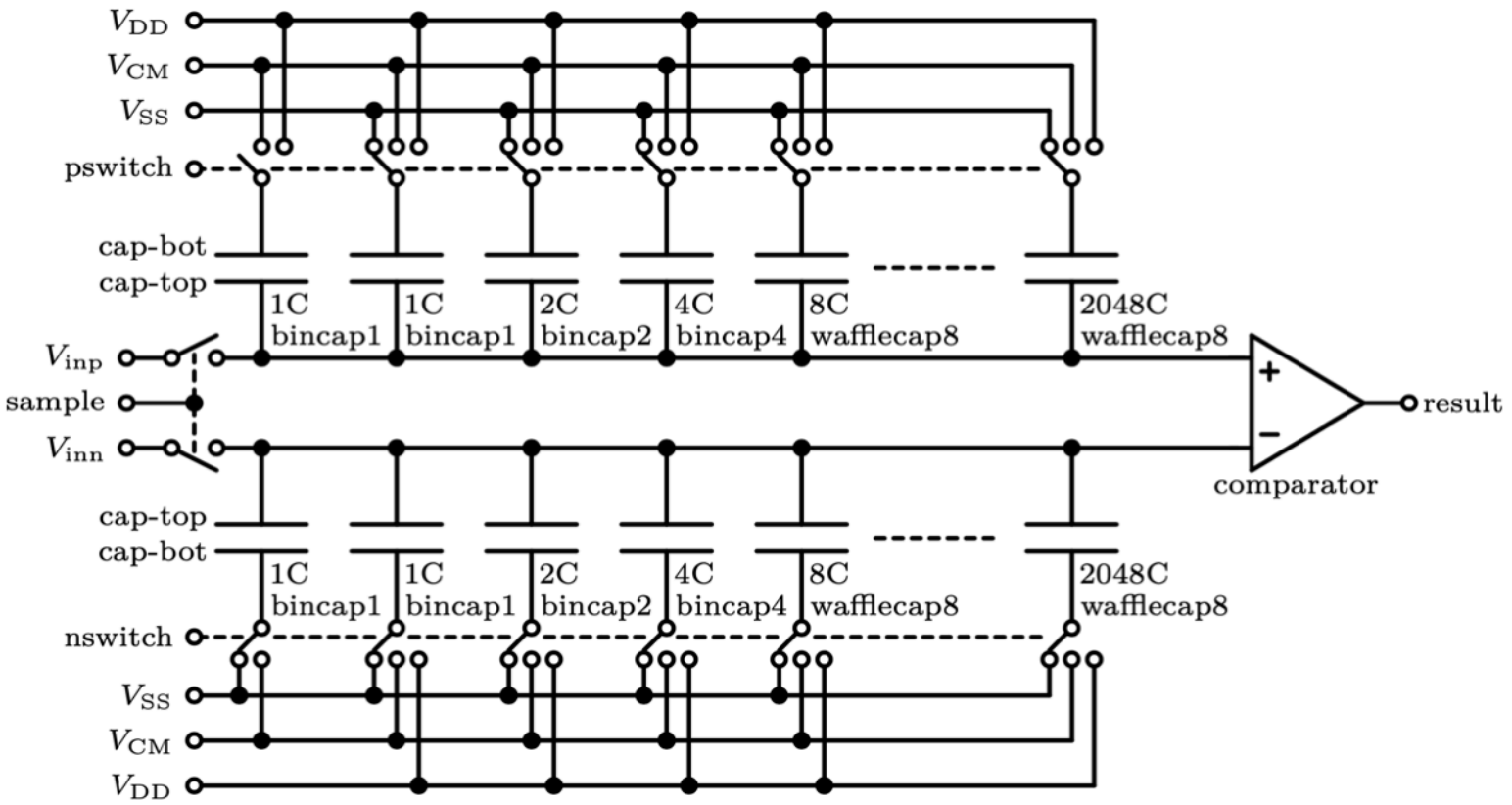
Calculation of result
Requires Verilog
Error recovery
>*N* steps needed

JOHANNES KEPLER UNIVERSITY LINZ

[1] F. Kuttner, "A 1.2V 10b 20MSample/s non-binary successive approximation ADC in 0.13µm CMOS," ISSCC, 2002.

SAR-ADC — Analog DAC / Sampling-Capacitor

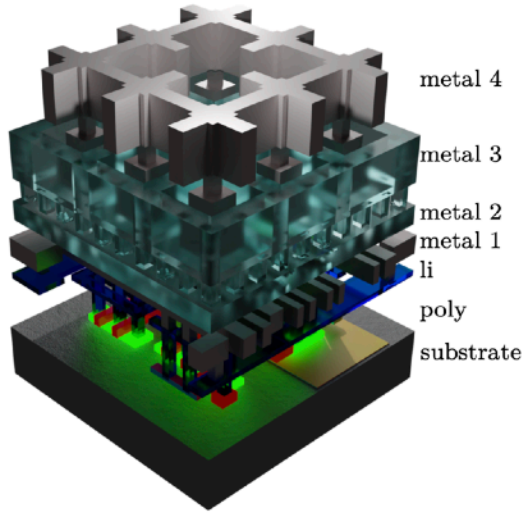
- Digital-to-Analog Converter (DAC) based on charge-redistribution [1]
- Voltage is sampled onto capacitive array (top-plate sampling)
- Segmented 12-bit capacitor array
- 9-bit thermometer code (511 cells of 8*C* unit caps with *C* = 447aF in 25µm²) arranged as a matrix [2]
- 3-bit binary constructed from thermometer unit cell (1*C*/2*C*/4*C*)
- Activation style of meander or common-centroid

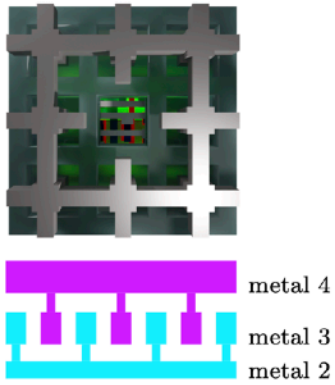


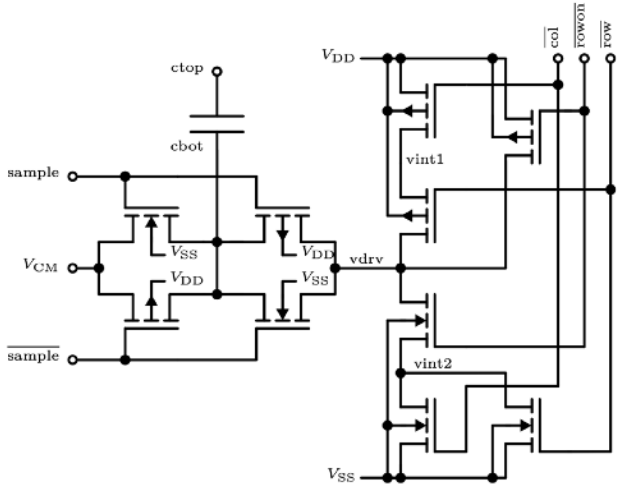
[1] J. L. McCreary and P. R. Gray, “All-MOS charge redistribution analog-to-digital conversion techniques. I,” JSSC, 1975.

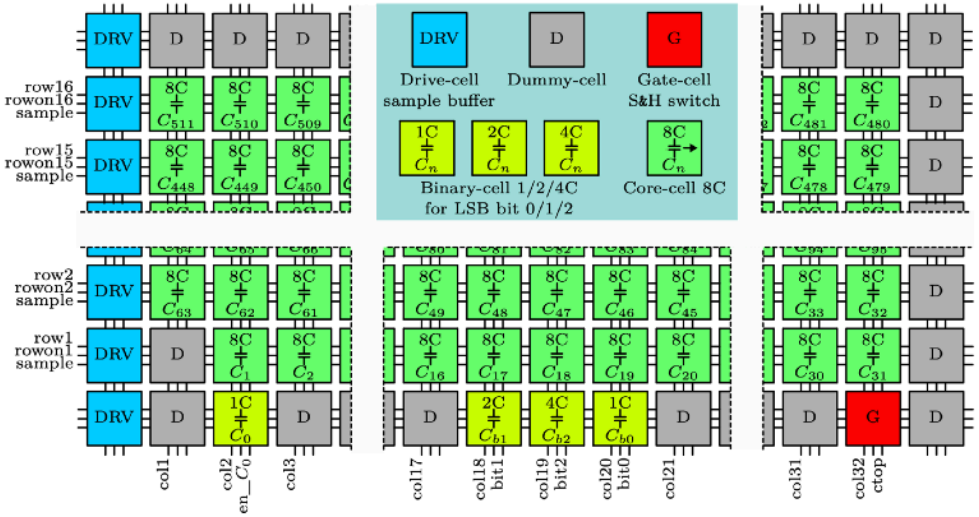
[2] T. Miki, et al., “An 80-MHz 8-bit CMOS D/A converter,” JSSC, 1986.

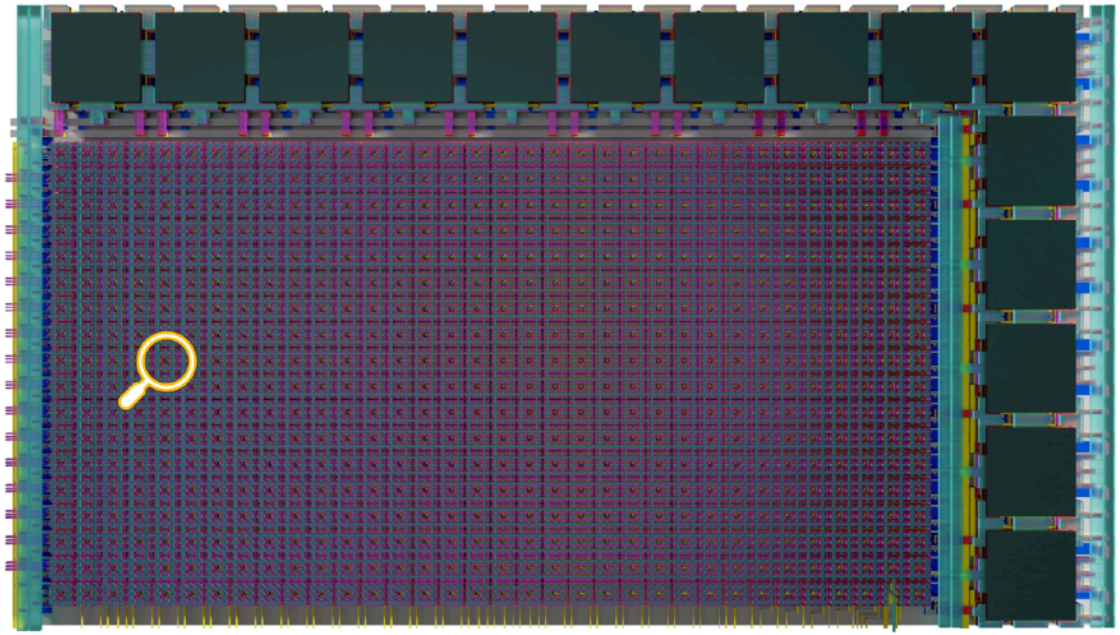
SAR-ADC — Analog DAC / Sampling-Capacitor (cont'd)





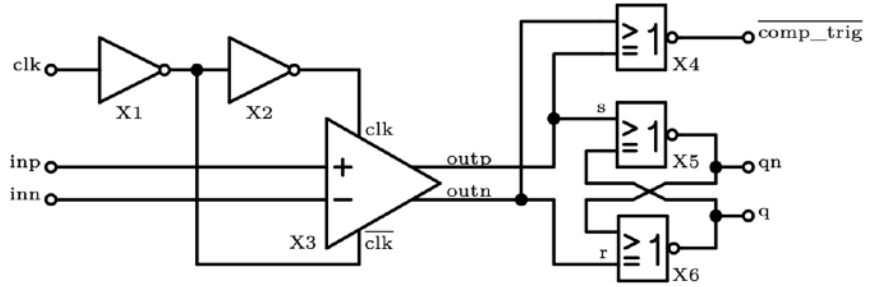


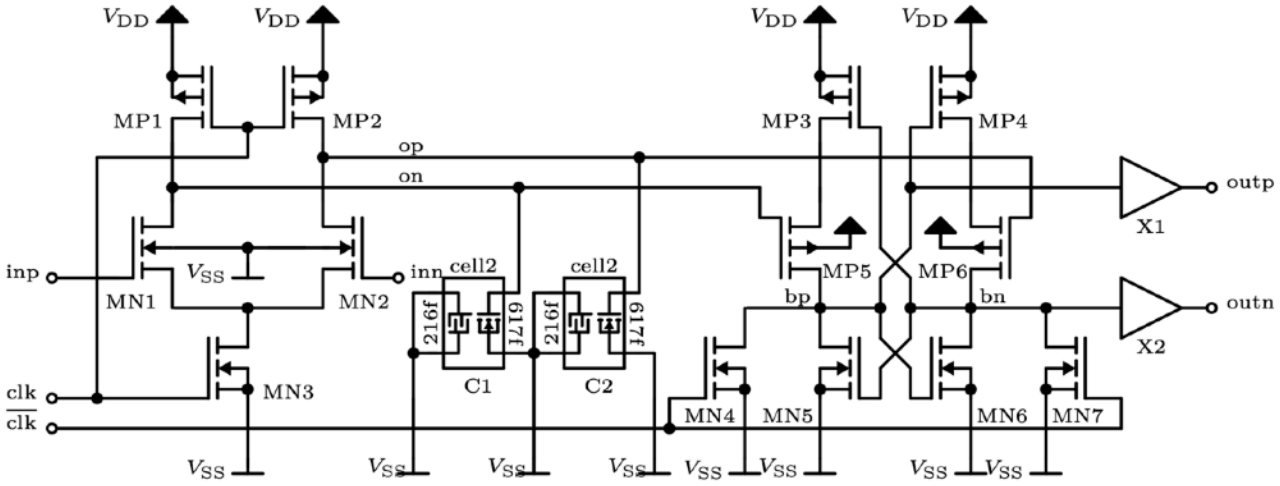


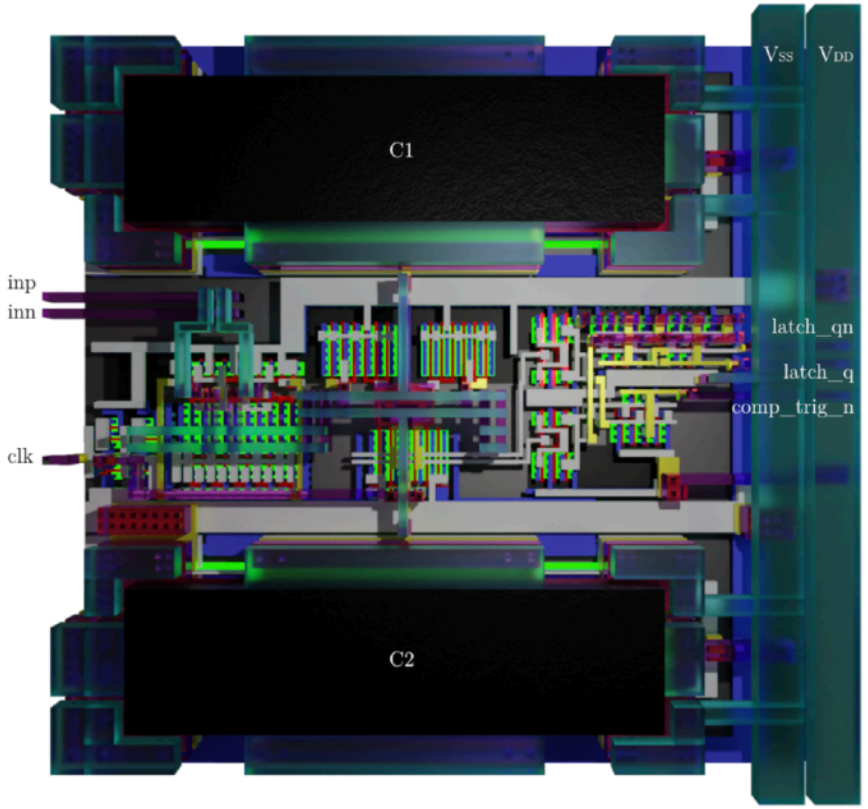


SAR-ADC — Analog Comparator / Latch

- Two-stage dynamic comparator [1]
- 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`**)
- **Issue**: Noise simulation not possible in open-source (no large signal noise simulation in **`ngspice`** or **`xyce`**)



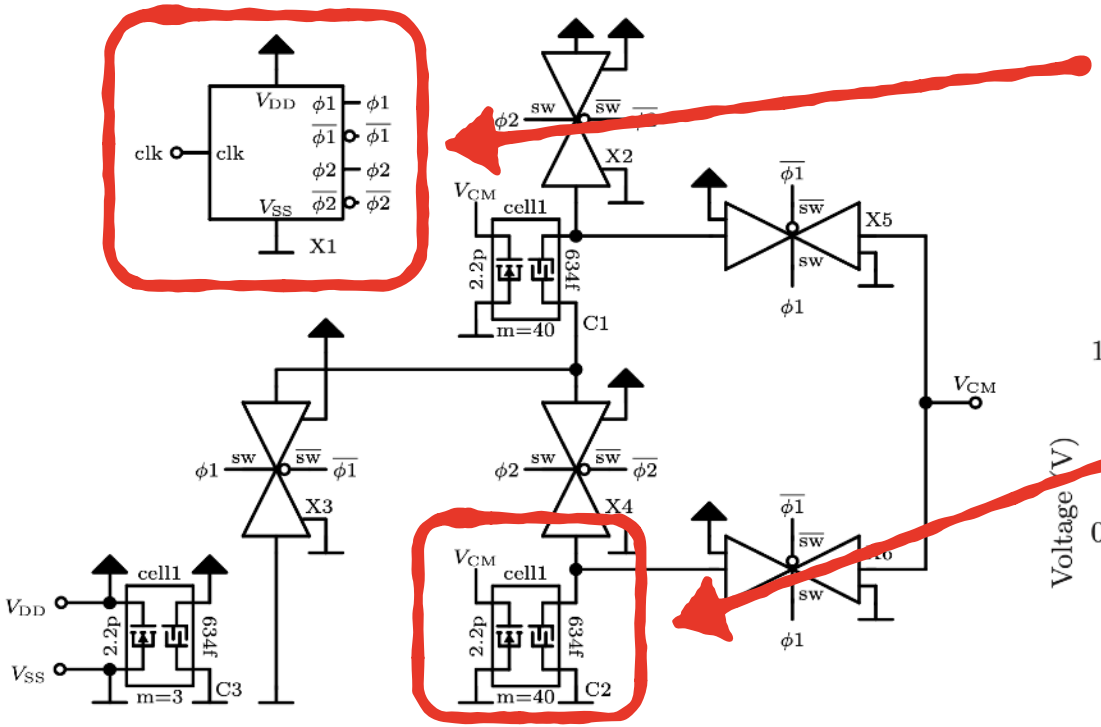


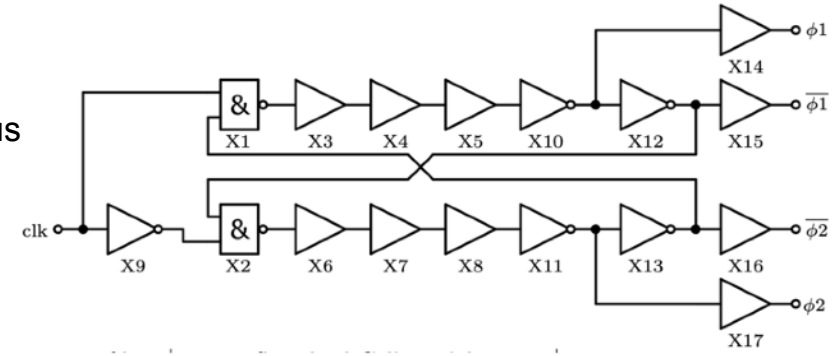


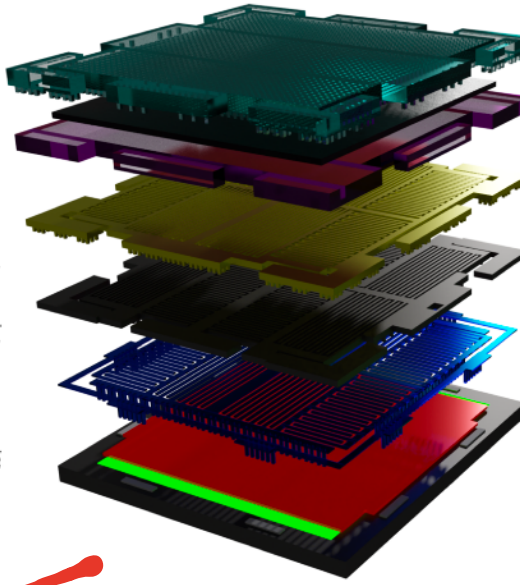
[1] M. Van Elzakker, et al., “A 1.9µW 4.4fJ/conversion-step 10b 1MS/S charge- redistribution ADC,” ISSCC, 2008.

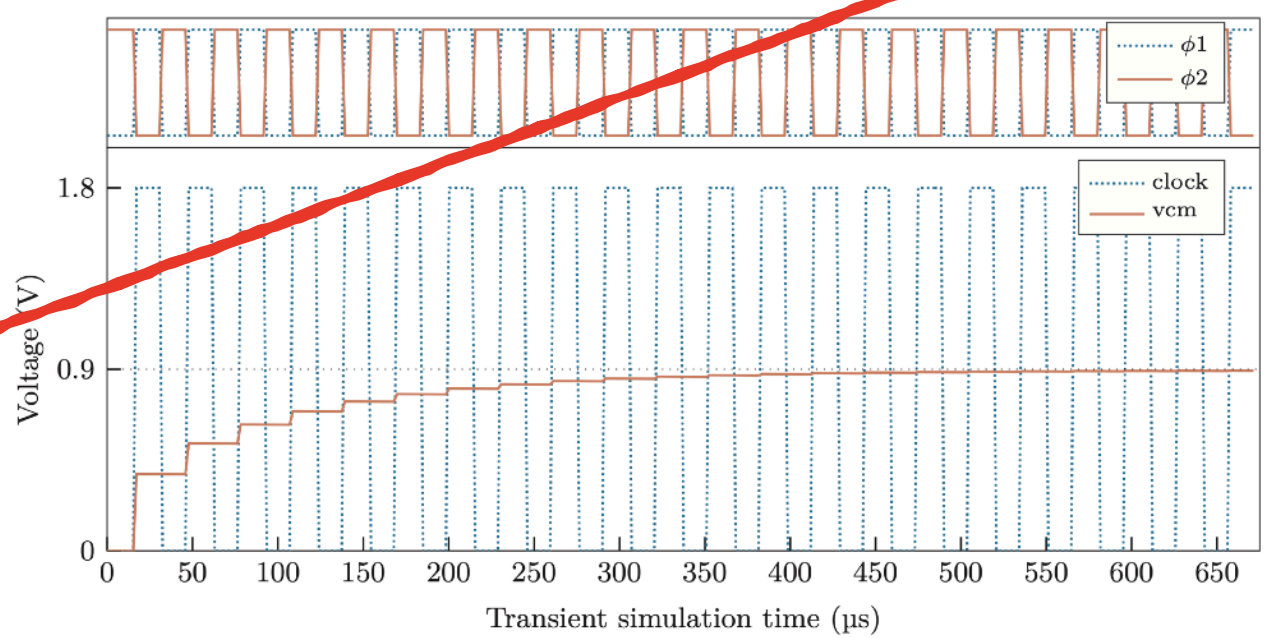
SAR-ADC — Analog Common-Mode Voltage Charge Pump

- A buffer-less charge-pump-generated $V_{CM}=V_{DD}/2$ is used during sampling
- Non-overlapping clock generator in **`xschem`** and layout using **`magic`**, simulation with **`ngspice`**
- A tile-able stacked capacitor cell (MOSCAP plus metal finger caps plus MIMCAP) is used with 7.2fF/µm2





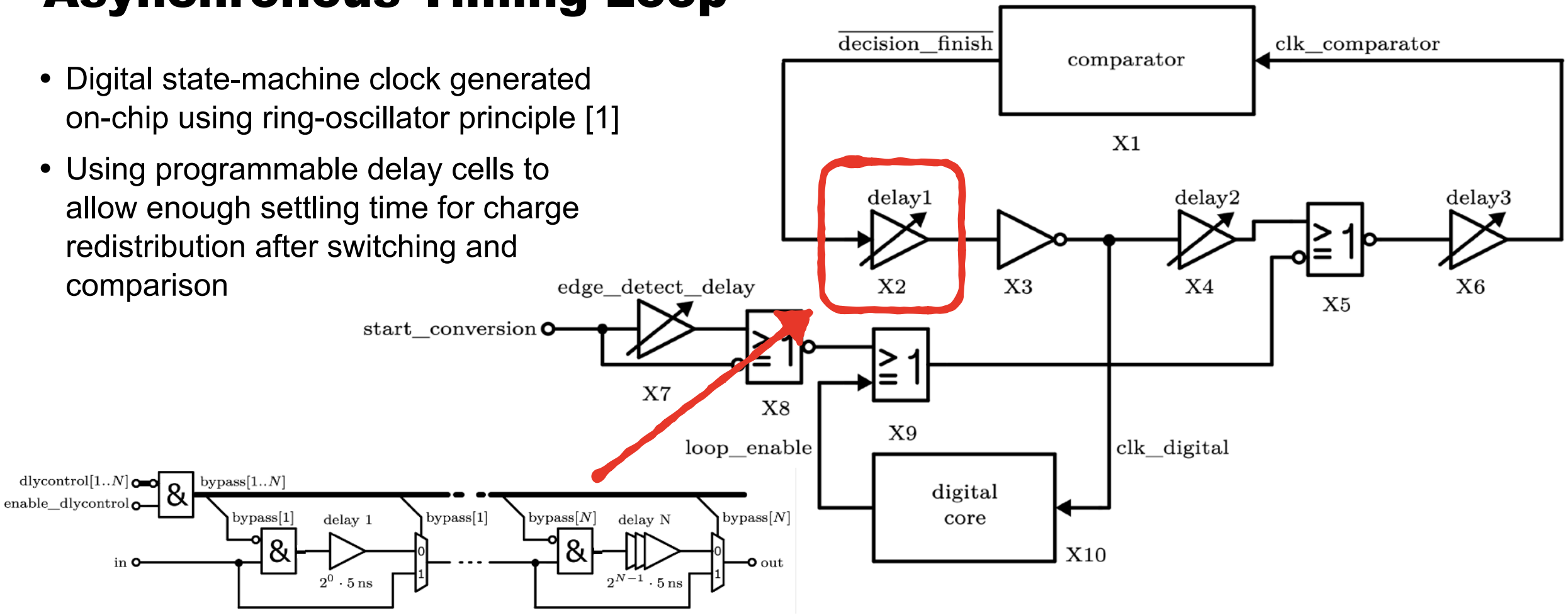




JOHANNES KEPLER UNIVERSITY LINZ

SAR-ADC — Digital Asynchronous Timing Loop

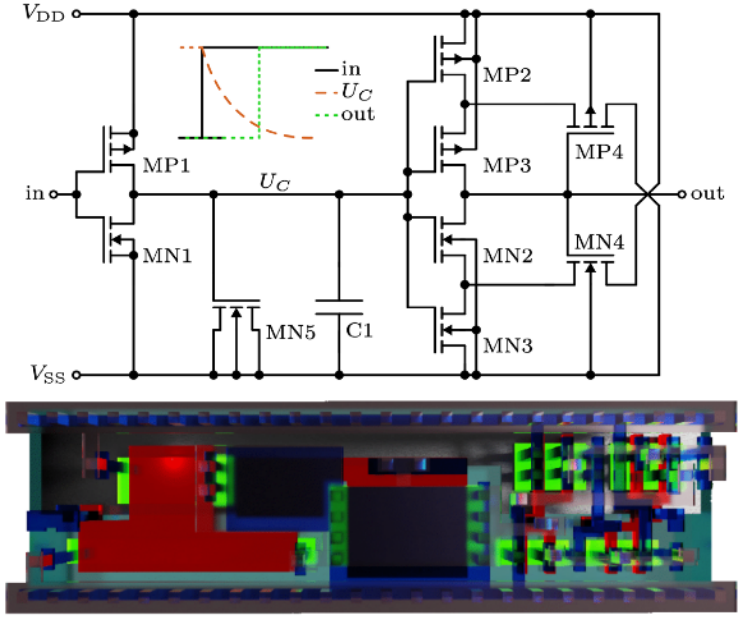
- Digital state-machine clock generated on-chip using ring-oscillator principle [1]
- Using programmable delay cells to allow enough settling time for charge redistribution after switching and comparison



[1] S.-W. M. Chen and R. W. Brodersen, “A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13-µm CMOS,” JSSC, 2006.

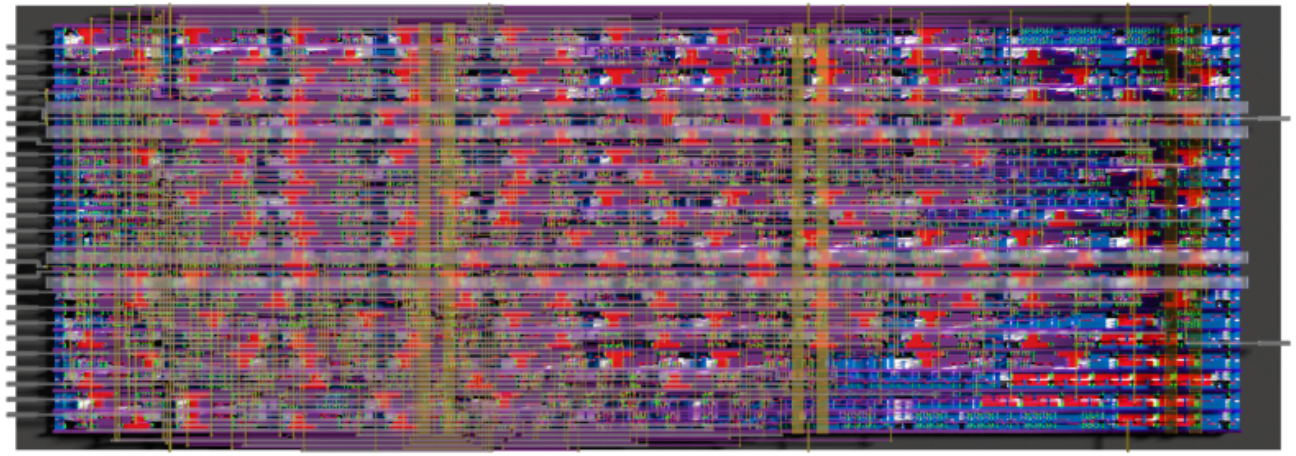
SAR-ADC — Digital
Custom standard cells (5ns delay)

- Custom standard cell (using **`xschem`**, **`magic`**, **`ngspice`**) for increased delay steps (5ns)
- P&R of programmable delay generator (5ns–100ns) implemented in Verilog, using **`openlane`**/ **`openroad`**, simulation with **`ngspice`** after PEX




```
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
```

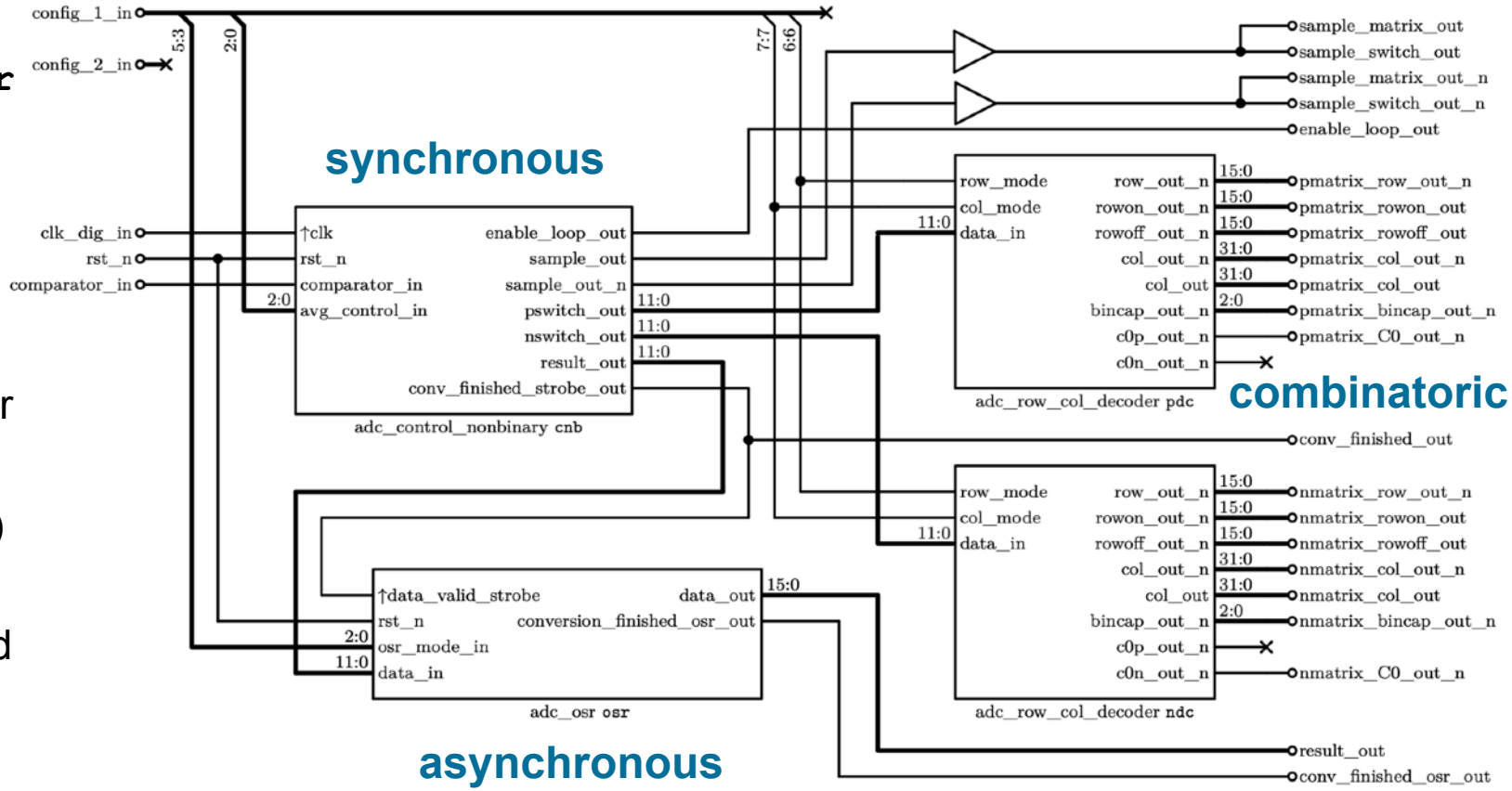
Gate level description



Layout automatic P&R

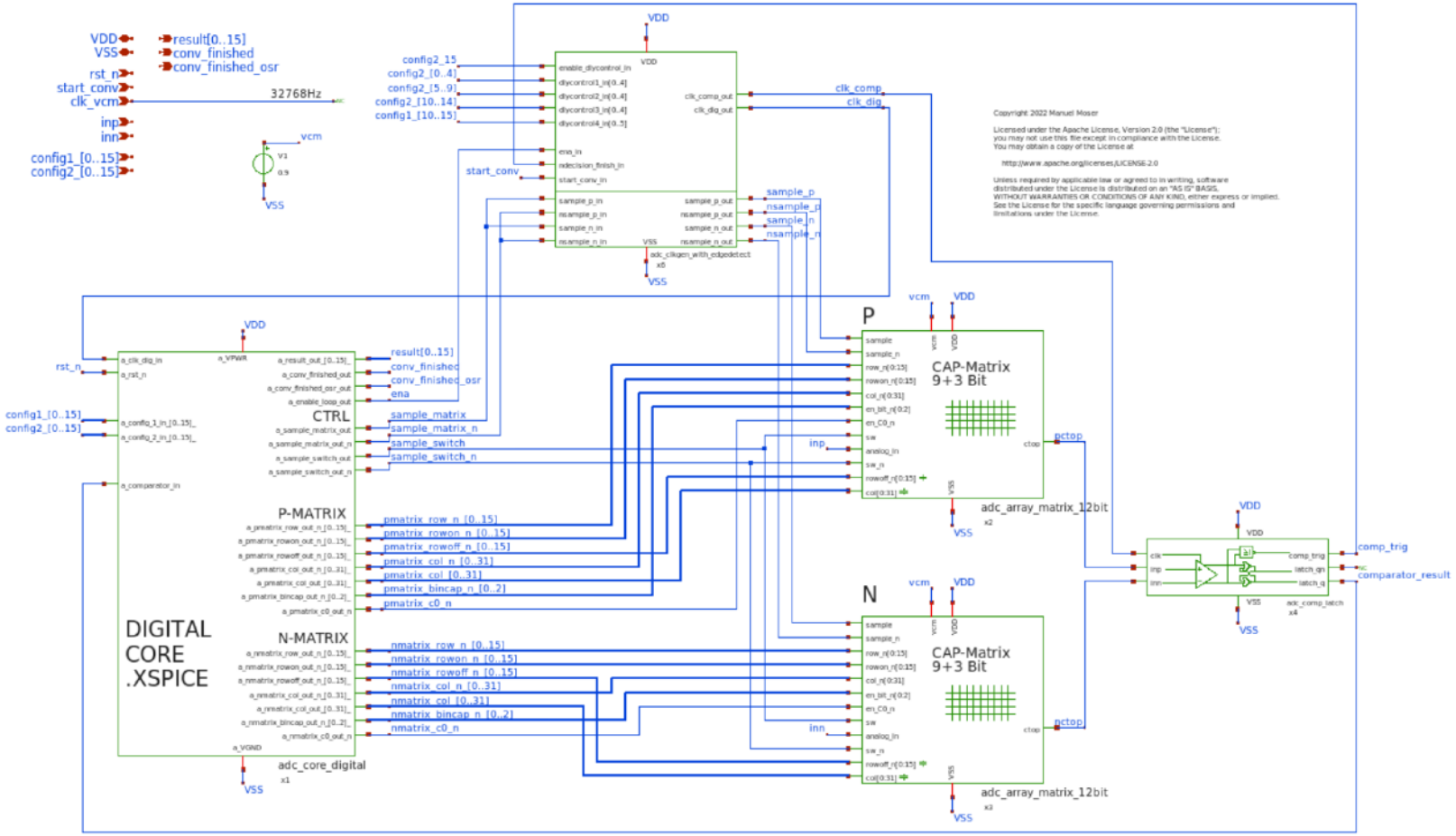
SAR-ADC — Digital
State Machine, Decoders, FIR Decimation Filter

- Design of digital logic in Verilog using **`iverilog`** for simulation, **`gtkwave`** for visualization of results, and **`verilator`** for linting
- Behavioral implementation of
  - **non-binary SAR** algorithm (x1.65)
  - **4-LSB averaging** (1/3/7/15/31)
  - **row/column decoders** with meander or common-centroid activation
  - **decimation FIR** filter (1/4/16/64/256)
- **No manual routing** between digital P&R block by proper pin placement and routing shape



SAR-ADC — Verification
Top-Level Mixed-Mode Simulation

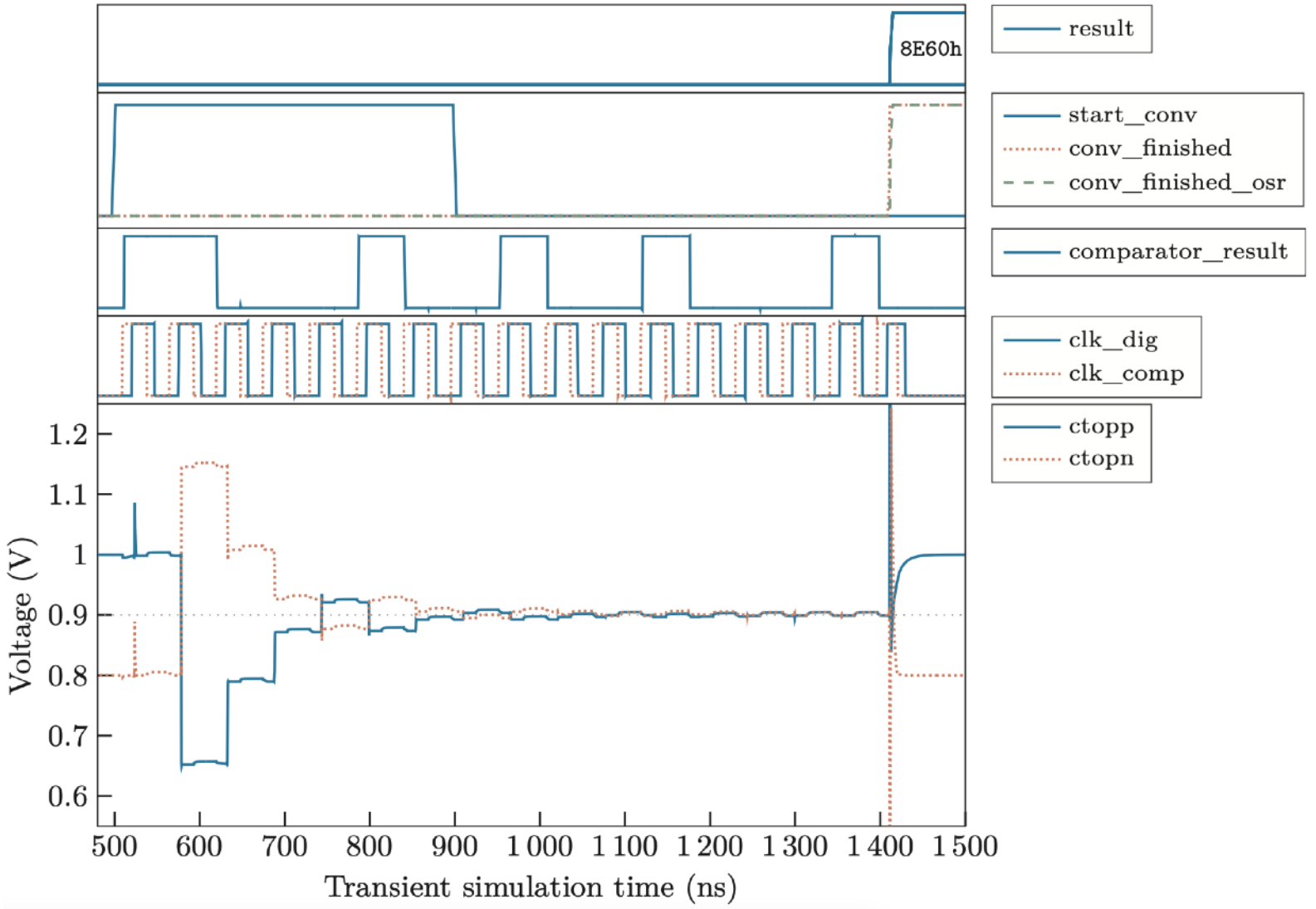
- Top-level schematic and testbench in **`xschem`** for pre-layout functional verification
- Mixed-mode simulation in **`ngspice`**, using `.spice` netlists for analog schematics, `.xspice` for Verilog blocks
- Creation of `.xspice` from Verilog using synthesis with **`yosys`** and gate-to-xspice conversion using **`qflow`** scripts(*)



(*) `vlog2verilog`, `verilog2spice`, `spi2xspice`

SAR-ADC — Verification
Top-Level Mixed-Mode Simulation (cont'd)

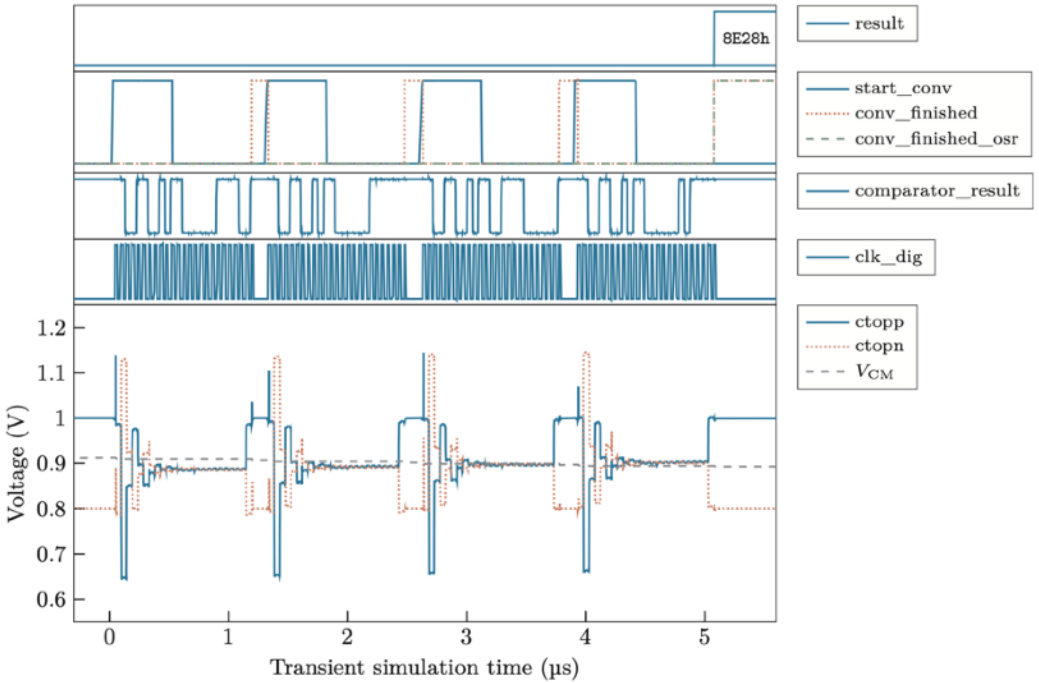
$V_{in,diff}$ = 200mV

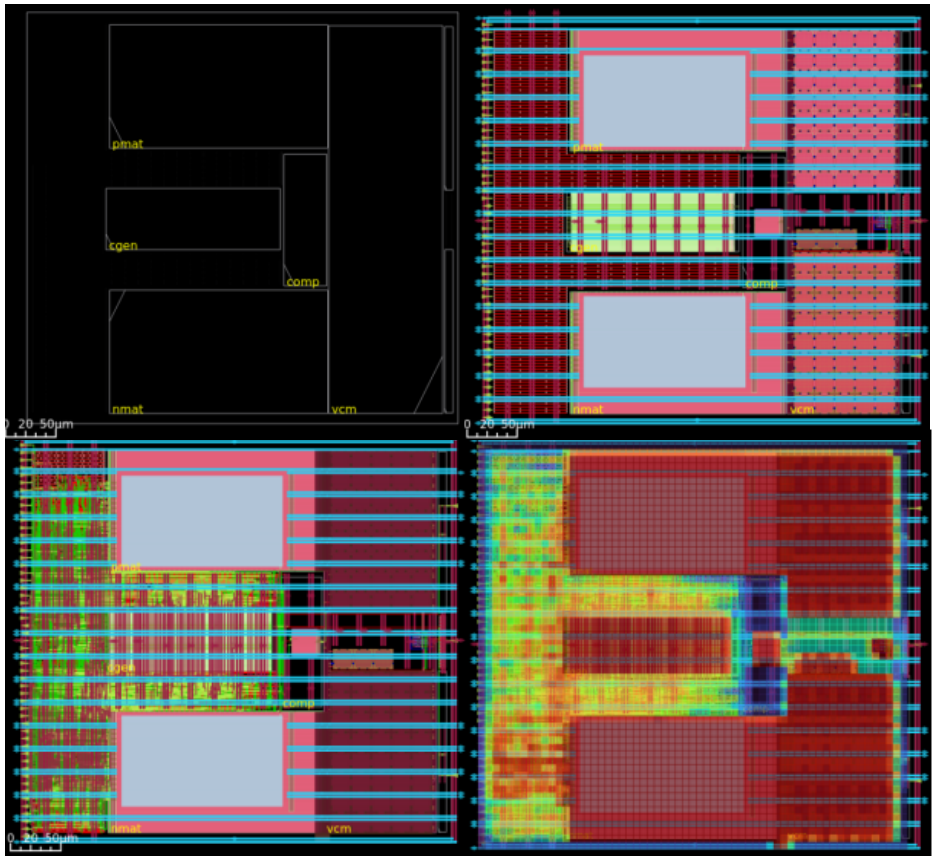


SAR-ADC — Top-Level Integration
Physical Integration & Verification

- Top-level physical integration using **`openlane`**/**`openroad`**: placement, routing between analog and digital blocks, supply grid
- PEX of top-level layout using **`iic-pex.sh`** (C-decoupled, C-coupled, RC)
- Post-layout simulation of C-extracted netlist using **`xyce`** (incl. digital blocks on transistor-level: 76k capacitors, 27k BSIM4, 250 resistors)

$V_{in,diff}$ = 200mV
N_{OSR} = 4
N_{AVG} = 3
$T_{DEL1/2/3}$ = 5ns
f_s = 820kS/s
f_{BW} = 100kHz
P_{diss} = 335µW





`openroad` views of the hardening process

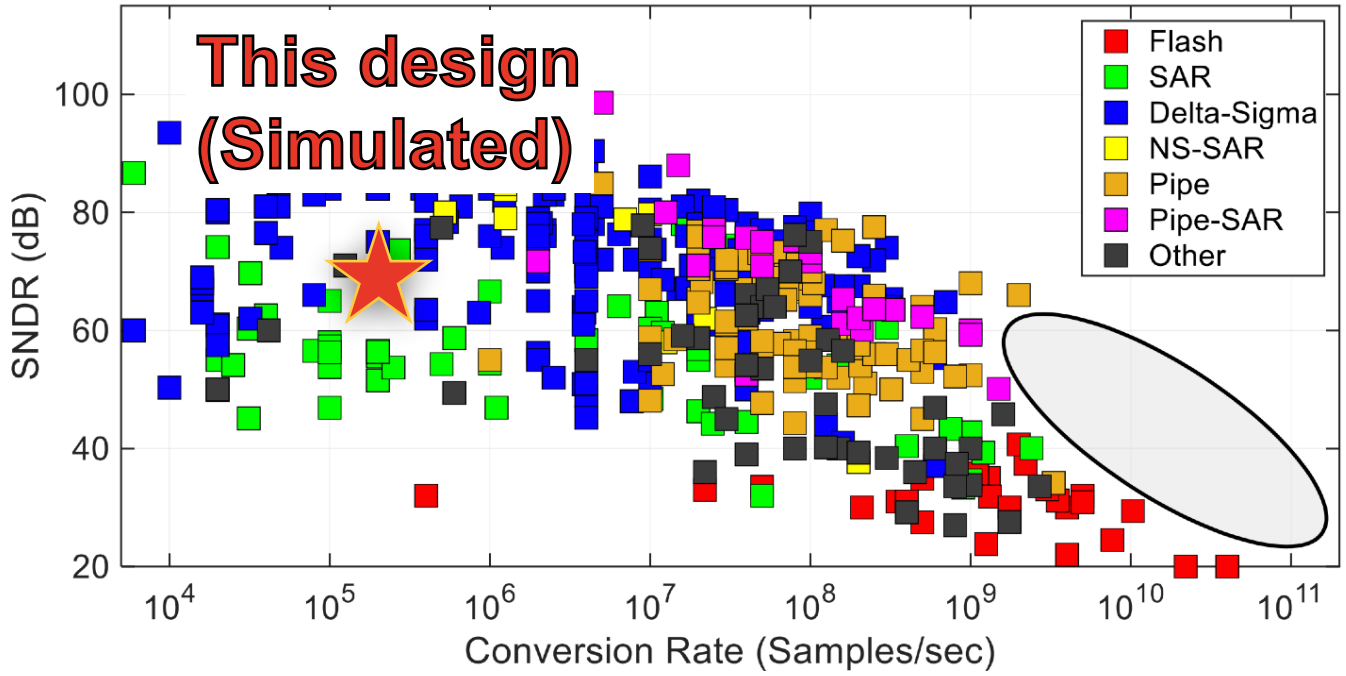
JOHANNES KEPLER UNIVERSITY LINZ

Agenda

- Introduction
- Open-source IC design environment
- ADC design details
- **Conclusion**

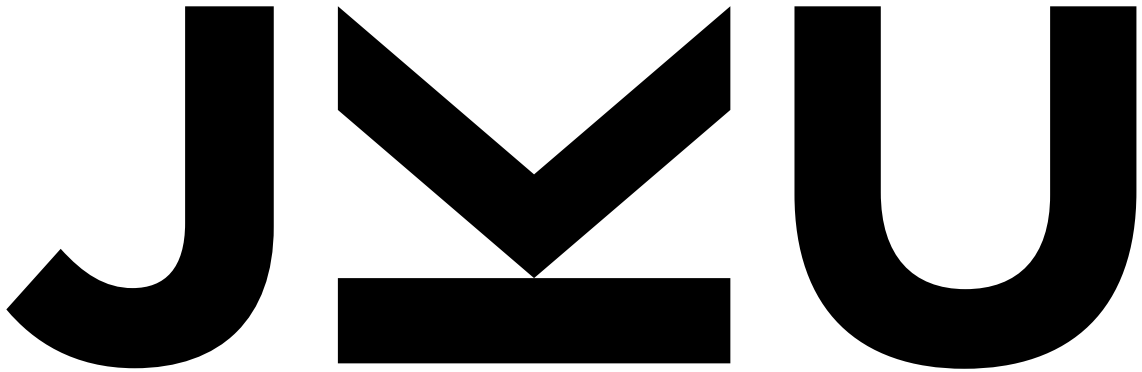
Conclusion

- A versatile 12-bit SAR-ADC has been designed using exclusively open-source IC design tools for SKY130.
- Custom design and layout blocks, as well as semi-custom digital synthesized P&R blocks, have been created.
- A digital-on-top flow is used to integrate the analog and digital macros into the top-level design.
- Pre- and post-layout simulation has been done to check functionality and performance.
- The complete design is open-sourced, incl. all intermediate files, and documentation is available.



SNR = 68dB @ 200kS/s assuming ENOB = 11b with N = 13b (OSR = 4)

Source: B. Murmann, "Introduction to ADCs/DACs: Metrics, Topologies, Trade Space, and Applications," ISSCC Tutorial, 2022.



JOHANNES KEPLER
UNIVERSITY LINZ