KÄRNTEN
University of
Applied Sciences

Learning IC design and the role of Coside enabling the use of SystemC (AMS) in education

COSEDA
usergroupmeeting

Wolfgang Scherr, Johannes Sturm

v1.3, 4th of December 2024

Agenda

- Overview of Carinthia Institute for Microelectronics and the Integrated Systems and Circuits Master program

- The role of SystemC (AMS) and Coside as integral part of the Master program plus some use cases

- Outlook

Carinthia Institute for Microelectronics „CIME“ … „at a glance“

KÄRNTEN University of Applied Sciences

CARINTHIA INSTITUTE FOR MICROELECTRONICS

Lead: J. Sturm

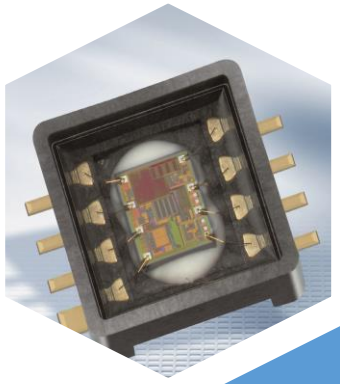
The **Carinthia Institute for Microelectronics** is a pool of experts with the clear passion for leading edge integrated circuit design.

We are a diverse team consisting of young talents, engineers, Post-Doc scientists and “old hands” with decades of industry experience.

Our research focus is on design and modelling of integrated circuits for different fields of applications like integrated sensors or wire-less and wire-line high speed communications.

Dr. Johannes Sturm

Head of Carinthia Institute for Microelectronics



RESPECT

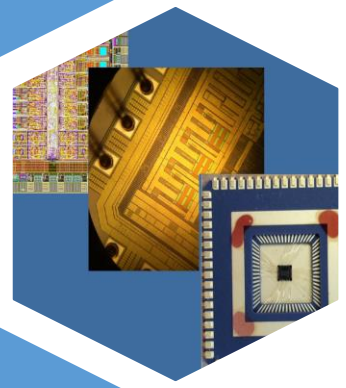
Research group for Modelling and Design of Integrated Systems and Circuits

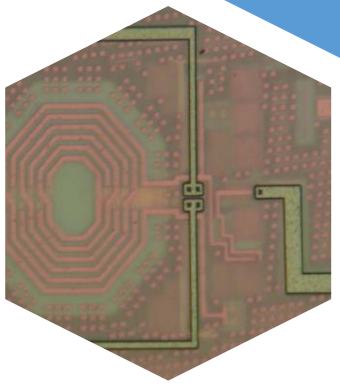
Lead: W. Scherr

Lead: J. Sturm

SIMS

Research group for Sensor Integration in Mechatronic Systems





RFFE-Lab

Research Lab for Radio Frequency Frontends

Lead: J. Sturm, S. Sondon

Key facts (2022/23):

Staff:

- 5 Key Researcher
- 3 Senior Researcher
- 3 Researcher
- 5-10 Master/PhD Students

Research Projects:

- > 700k€ / year

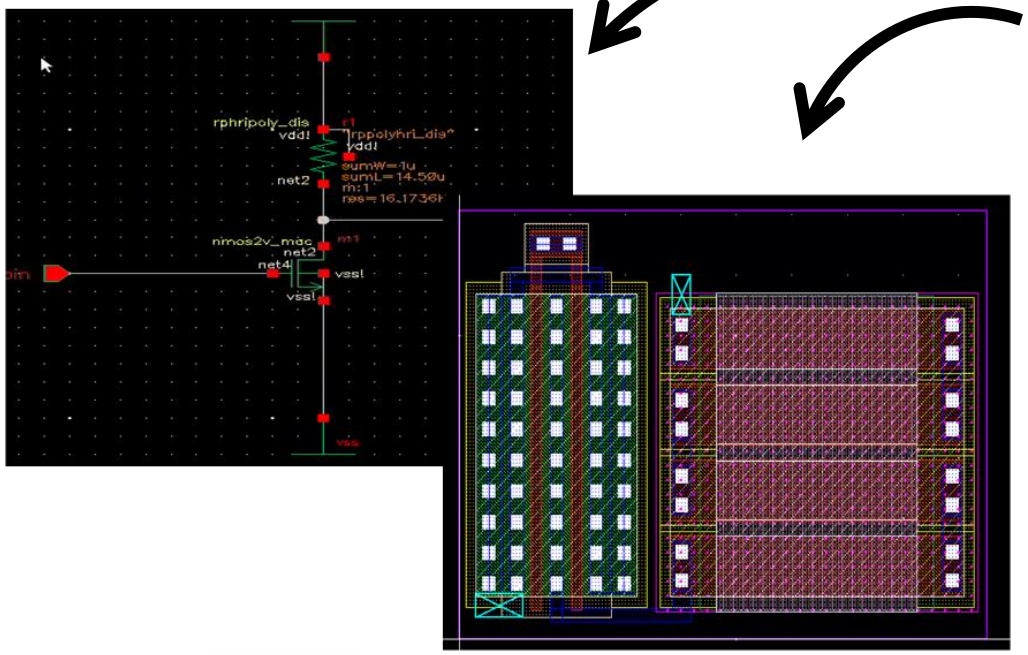
Project partners:

- Silicon Austria Labs
- University of Klagenfurt
- Johannes Kepler Univ. Linz
- Infineon Technologies
- Coseda Technologies
- NXP
- IHP
- IMEC
- Joanneum Research
- ….

www.cime.at

SODA: Austrian Josef Ressel Centre for System on Chip Design Automation

Item	Value
Preamplifier gain	18 dB (typical), 17 dB (min)
Preamplifier noise figure	0.9 (typical), 1.0 (max)
Preamplifier impedance	50 ohms, dc grounded
Receiver sensitivity	<0.3 μV
Receiver IF	1st: 10.7 MHz, 2d: 455 kHz
Receiver signal level range	50 dB
Receiver tuning	Software selectable in 5-kHz steps
Receiver output level	0–4 V
APT decoder sampling rate	Up to 5670 samples s^{-1}
APT decoder audio levels	100 mV pp–2.5 V pp



First publications @ IEEE and Accellera available.
Plans to release of Python framework “CUAS Cell Creator”.

Curriculum – ISCD Master

Master degree program "ISCD - Integrated Systems and Circuits Design - ISCD"

	Circuit Design and Technology						Design Methodology						System Solutions						Implementation and Verification				Master Thesis					
ECTS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**22**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
1st semester	Analog Integrated Circuits 1 ILV - 3,5 SWS / 5 ECTS					Digital Integrated Circuits 1 ILV - 3,5 SWS / 5 ECTS					Integrated Circuits Technology ILV - 3,5 SWS / 5 ECTS					Computer Aided Design 1 ILV - 3 SWS / 5 ECTS					Methods in Systems and Circuits Theory ILV - 3,5 SWS / 5 ECTS				Introduction to Integrated Circuits Design Project PA - 3 SWS / 5 ECTS					
2nd semester	Analog Integrated Circuits 2 ILV - 3,5 SWS / 5 ECTS					Digital Integrated Circuits 2 ILV - 3,5 SWS / 5 ECTS					Computer Aided Design 2 ILV - 3,5 SWS / 5 ECTS					System Modeling and Verification ILV - 3,5 SWS / 5 ECTS					Design and Implementation of Analog Circuits and Systems PA - 3 SWS / 5 ECTS				Design and Implementation of Digital Circuits and Systems PA - 3 SWS / 5 ECTS					
3rd semester	Radio-Frequeny Circuits and Systems ILV - 3 SWS / 5 ECTS					Integrated Sensors ILV - 3 SWS / 5 ECTS					Smart Power Integrated Circuits ILV - 3 SWS / 5 ECTS					Verification of Integrated Circuits and Systems ILV - 3 SWS / 5 ECTS					Advanced Topics in Mixed Signal Design (elective) ILV - 3 SWS / 5 ECTS				Advanced Topics in System-on-Chip Design (elective) ILV - 3 SWS / 5 ECTS					
4th semester	Master Thesis MT - 0,5 SWS / 24 ECTS																								Master Thesis Seminar SEM - 3 SWS/3 ECTS					Master Exam ME - 0 SWS/3 ECTS

Curriculum update, 2023 → Lecturers e.g. from CUAS, Infineon Technologies, AMS OSRAM, University Ljubljana and others.

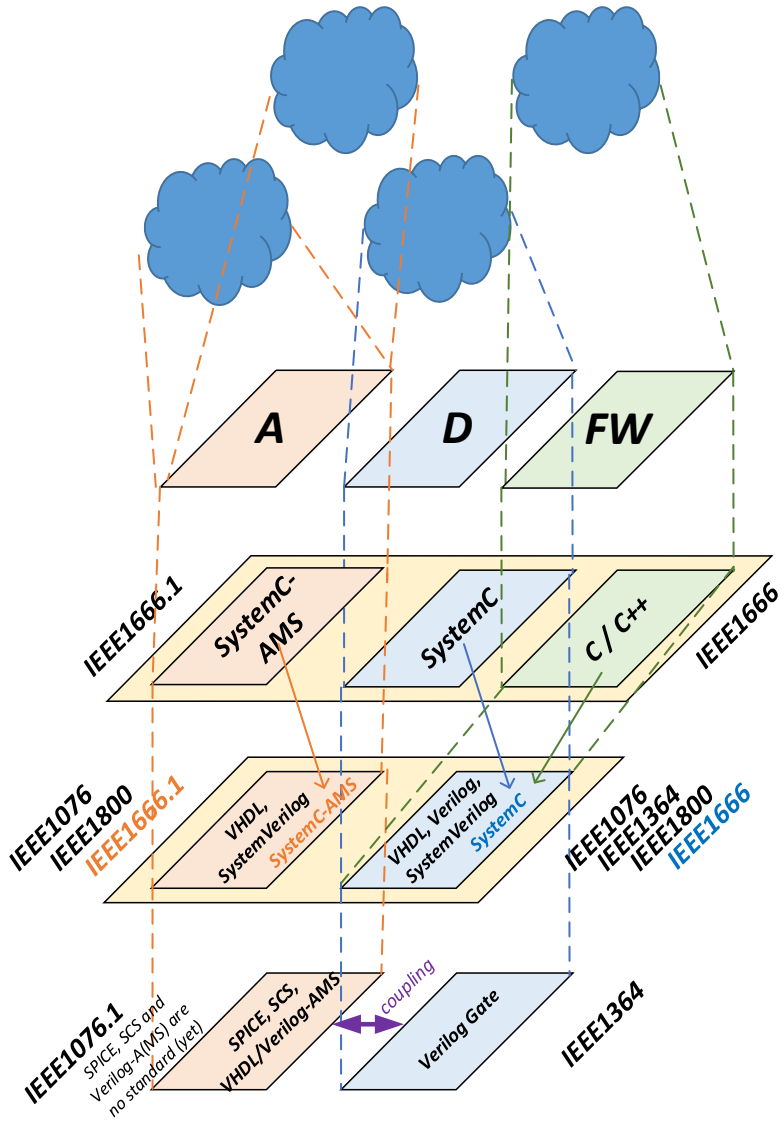
We are planning to offer micro-credentials as on-site, hands-on training in English (some details still TBD)

.) analogue/digital design basics: 2 intensive weeks without exam and "attended certificate (or with online exam afterwards → 7.5 ECTS certificate)
.) full-chip team project design: 2 intensive weeks without exam and "attended certificate (or with online exam afterwards → 7.5 ECTS certificate)
.) Minimum prerequisites will be an electronics Bachelor degree, there will be also a selection process as places are limited.

Students (PhDs) or young engineers will learn the whole mixed-signal microelectronics flow using industry-standard tools in a compact format, based on a Master degree program continuously improved since 2006!

The first courses we plan to offer around summer 2025 (not online yet).

Our backbone: industry EDA tools, PDKs, standards



Matlab, Simulink

C/C++

Office Tools

Python

XILINX Vivado, Pynq

COSEDA Technologies Coside

Mentor A Siemens Business Questa, Catapult

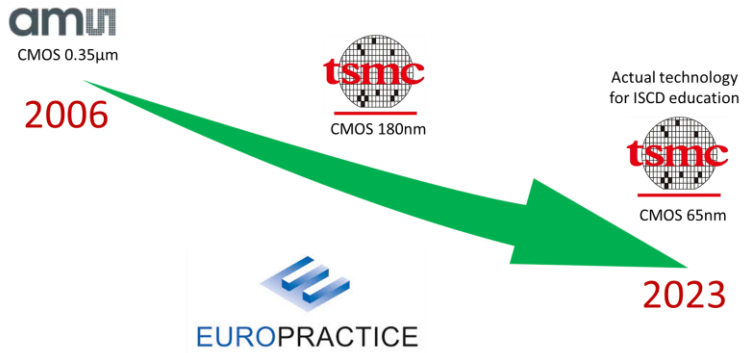
cadence Virtuoso, ADE, Spectre, ...

SYNOPSYS Design Compiler, Formality, PrimePower, ...

Mentor A Siemens Business Calibre, ...

cadence Innovus, ...

- State-of-the-art IC technologies

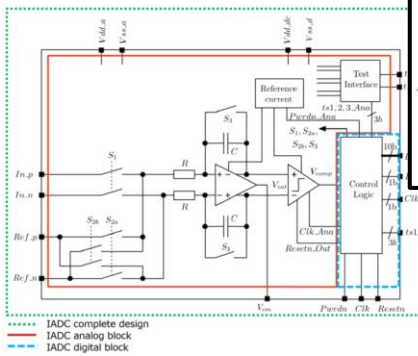


- Close interaction between study and research projects (internships, master theses,...)
- High-quality lectures - lecturers with long academic & industry background

The student project: create a whole M/S IC in 1 year!

KÄRNTEN University of Applied Sciences

Including lab characterisation approx. 20 ECTS / ~500hrs



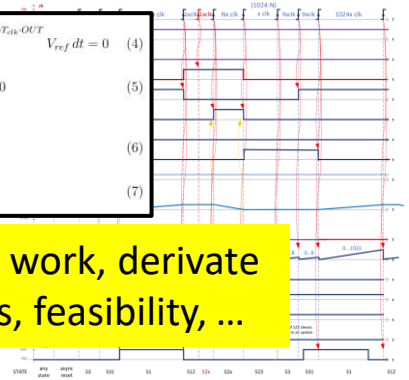
$$\frac{1}{RC}\int_0^{T1} V_{in}\,dt - \frac{1}{RC}\int_0^{T2} V_{ref}\,dt = \frac{1}{RC}\int_0^{T_{clk}\cdot 2^N} V_{in}\,dt - \frac{1}{RC}\int_0^{T_{clk}\cdot OUT} V_{ref}\,dt = 0 \quad (4)$$

$$\frac{1}{RC}\cdot T_{clk}\cdot 2^N\cdot V_{in} - \frac{1}{RC}\cdot T_{clk}\cdot OUT\cdot V_{ref} = 0 \quad (5)$$

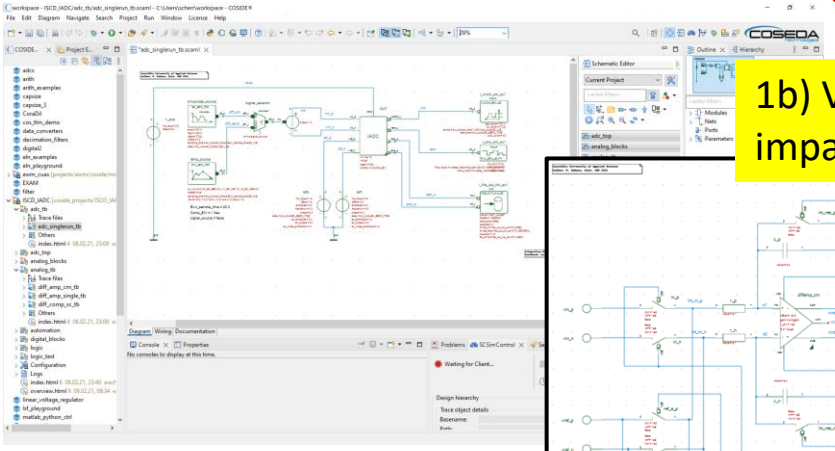
As T_{clk}, R and C is always positive, we can reduce to this equation.

$$2^N\cdot V_{in} - OUT\cdot V_{ref} = 0 \quad (6)$$

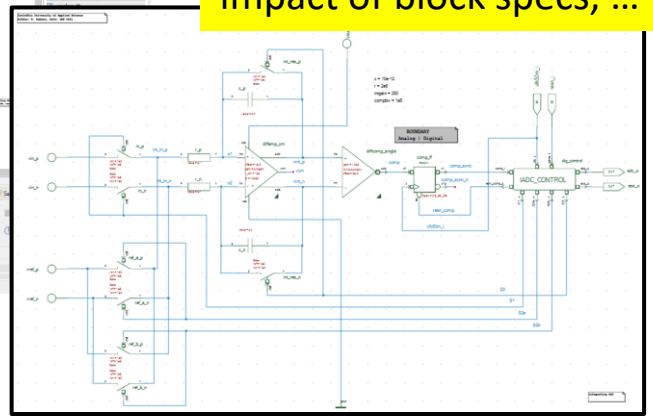
$$OUT = 2^N\cdot\frac{V_{in}}{V_{ref}} \quad (7)$$



1a) Methods, concept work, derivate initial block level specs, feasibility, …

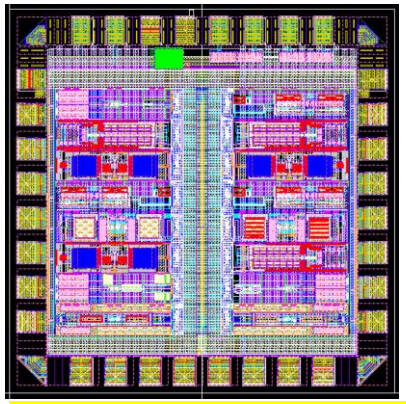


1b) Virtual prototyping, impact of block specs, …



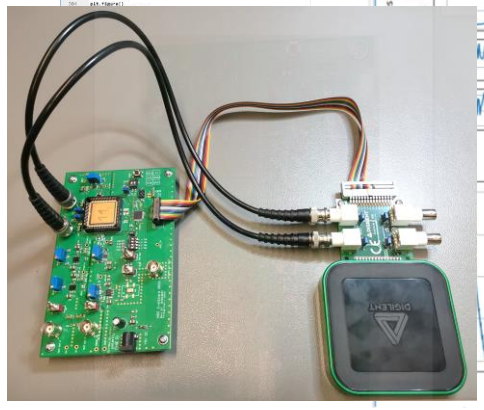
3a) Blocklevel & toplevel (M/S) verification

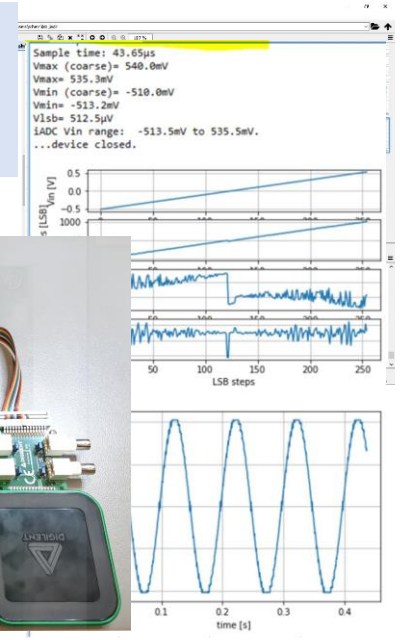

```
architecture rtl of control is  -- rtl style architecture of control

-- state data type and signal declaration
-- **** defines FSM states as enumerated datatype ********************
type cntrl_states is(state_r, state_s3x, state_s3, state_s31, state_s1,
                     state_s2x, state_s2a, state_s2b, state_s23);

signal state    : cntrl_states; -- state vector register
attribute fsm_state : string;
attribute fsm_state of state : signal is "BINARY";      -- encoding sty

signal count    : unsigned(9 downto 0);  -- core 10bit counter
signal neg      : std_logic;             -- helps to detect negative cou
signal cres     : std_logic;             -- sync. reset of counter
signal cres_del: std_logic;              -- sync. reset of counter
signal cres8    : std_logic;             -- sync. reset of counter on co
signal cnt1023 : std_logic;              -- signal end of counting (one
signal cnt0     : std_logic;             -- signal end of counting (over

signal capture  : std_logic_vector(9 downto 0); -- internal output regi
signal cap_neg  : std_logic;                    -- internal output regi
signal cap      : std_logic;                    -- capture control for
signal upd      : std_logic;                    -- update control for f
signal out_en   : std_logic;                    -- supress clkout until
signal dout_int : std_logic_vector(9 downto 0); -- internal dout to all
signal sgn_int  : std_logic;                    -- internal sign to all

-- output signals for synchronisation
signal s1_int         : std_logic;
signal s2a_int        : std_logic;
```



3b) TSMC 65nm test chip TO with 16 student blocks

4) Lab automation & ADC characterisation (in the 3rd semester)





2a) Analogue design & fullcustom flow

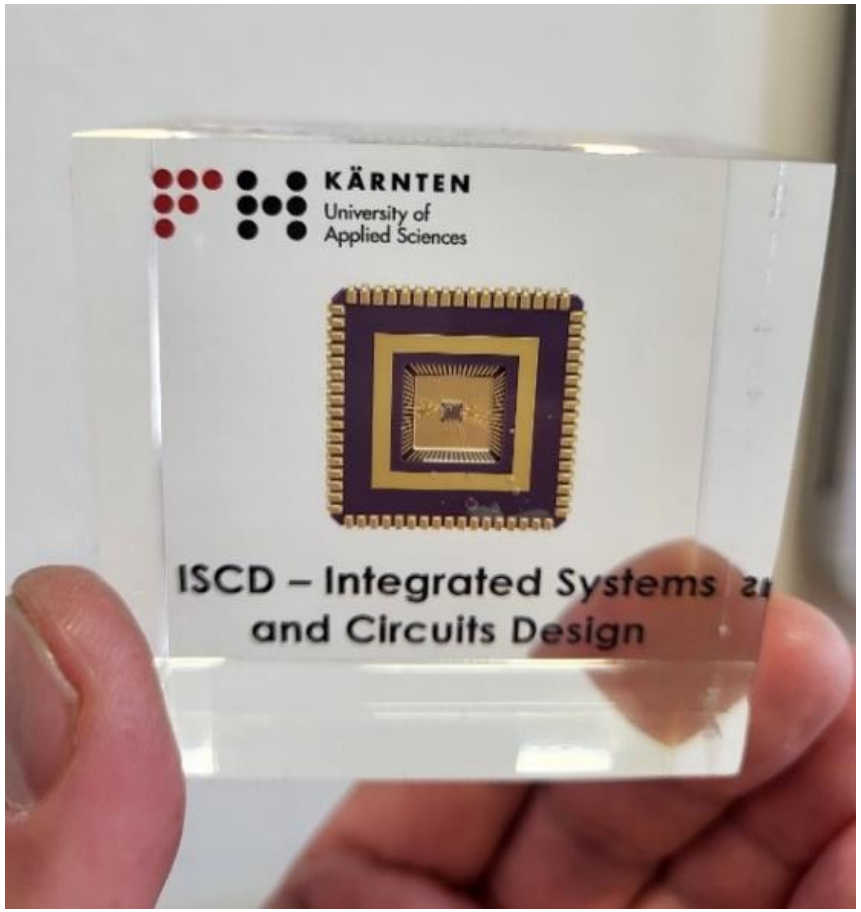
2b) Digital design & semicustom flow

see also: https://www.fh-kaernten.at/fileadmin/documents/studienbereiche/engineering-it/Siemens-SW-Carinthia-UAS-CS-85054-D3_90__1_.pdf
and: https://systemc-ams.at/ISCD_IADC/overview.html

WWW.FH-KAERNTEN.AT

The famous trophy of 4 semesters tough work…

- Every student finishing the ISCD Master program successfully will get it ☺



Job opportunities - several even in close vicinity

ISCD graduates can work as specialists in the semiconductor industry (foundries, fabless foundries, and engineering companies) as well as for businesses providing electronic system solutions.

Their fields of activities include

- research/feasibility
- concept/design
- modelling/prototyping
- implementation
- verification/validation/testing
- assembly
- integration
- technical support

of analogue, digital and mixed-signal integrated circuits and systems-on-chip (SoCs).

Agenda

- Overview of Carinthia Institute for Microelectronics and the Integrated Systems and Circuits Master program

- The role of SystemC (AMS) and Coside as integral part of the Master program plus some use cases

- Outlook

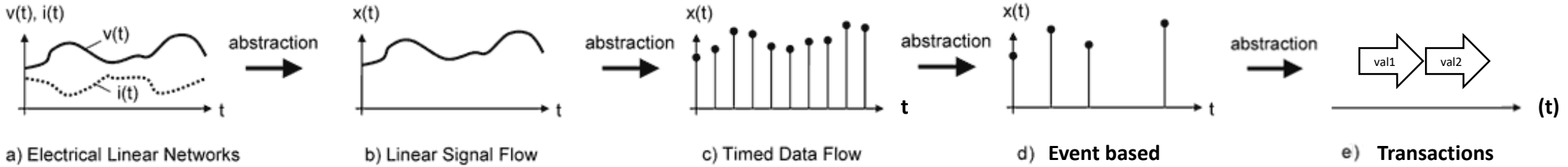
Coside and SystemC AMS in Education

- SystemC AMS offers different MoC (models of computation) in one environment
  - “transferable” abstraction for the topics at hand, from circuit problems to software design
- Fostering standards
  - taught at university, relevant throughout the industry career
- Coside is a great platform for lecturing
  - low entry barrier, fast turn-around for interactive use, efficient tools
- The setup is still not perfect, though… ;-)
  - potential improvements, especially for education/beginners level users

Abstraction helps in understanding

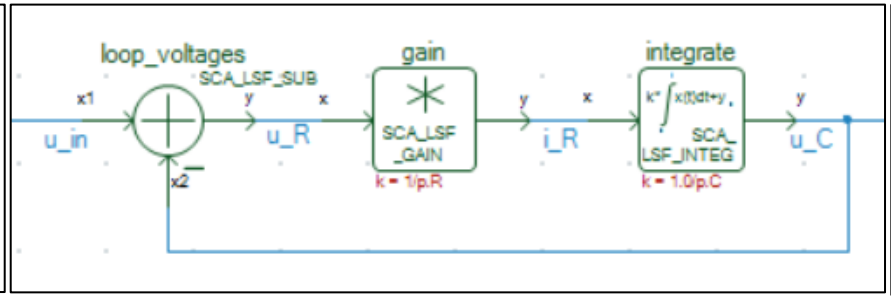
- Learn basic steps of (macro) modelling on many abstraction levels w/o switching tools (but usually easily transferrable to other tools/languages)

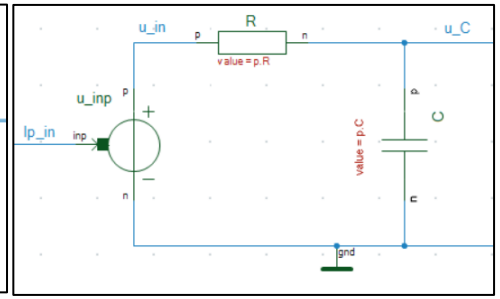
Abstraction for speed



Abstraction for understanding

```
static double Ts = 1.0/p.fs;
static double a = exp(-Ts/(p.R*p.C));

while(true) {
    s.z_int = a*s.z_int + (1.0-a)*inp.read();
    outp.write(s.z_int);

    wait(Ts, SC_SEC);
}
```





Partly taken and modified from "AMS Modeling Standard SystemC AMS Extensions : Solving the Need for Speed", OSCI Whitepaper, M. Barnasconi

Not to under-estimate:
from standard implementation to lecture content

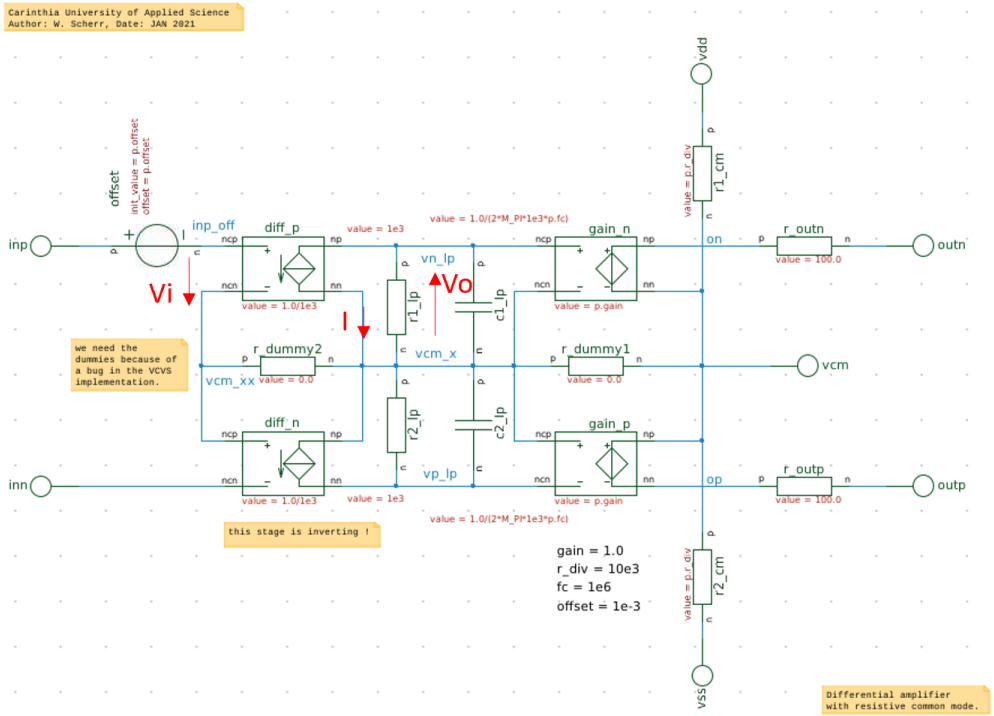
- Teaching based on (open) standards ensure sustainability later in the industry
- CUAS is university member of Accellera and the AMS working group
- Feedback on issues found during lecturing (examples, assignments, …) to the WG
- News/insights out of the WG can be directly included in lecture content

Low entry barrier of Coside, flexible environment

- Compared to other tools, Coside allows students to get started quickly
  - Graphical entry for all relevant MoCs of SystemC
  - The tool allows to focus on electrical engineering, not computer science
  - Interactive simulation of models w/o need for writing testbench code (SCSimControl)
- Schematics are easy to create and read, with different levels of abstractions
  - Very well usable as lecture material
- Embedded waveform editor with Python capabilities
  - Tracing all signals allows simple circuit analysis and post processing in the waveform tool
  - For high-quality figures with annotations, also useful as lecture material
- The generated code is human-readable
  - Students can (re-)use code for (own) models later or check out how things are coded
- Embedding SystemC in other tools
  - Links to Virtuoso, Matlab/Simulink/Octave, Xcelium, Python, … exist for advanced topics

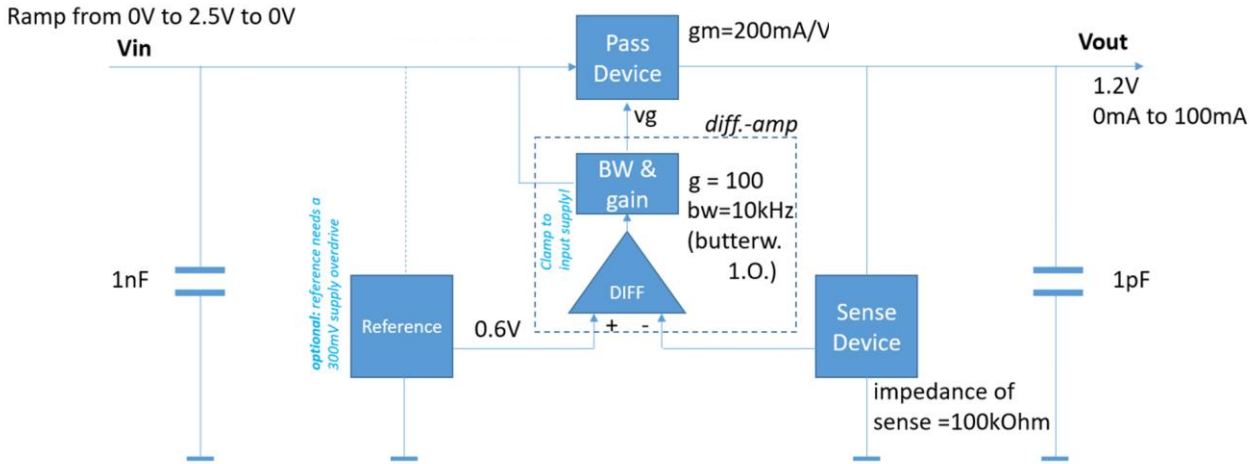
Analogue examples - learning macro models

- Band-limited, fully differential operational amplifier with resistive CM setup



- High-level voltage regulator optimisation example as assignment

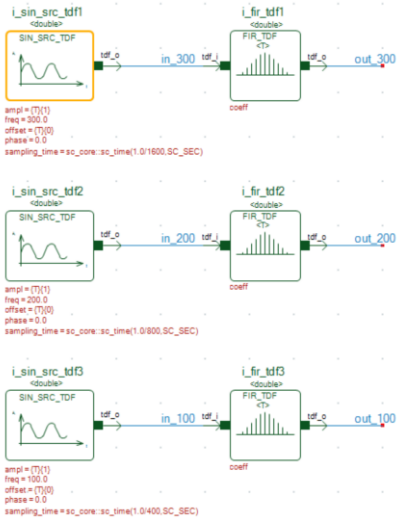
You have to set up a model based on this basic regulator architecture:

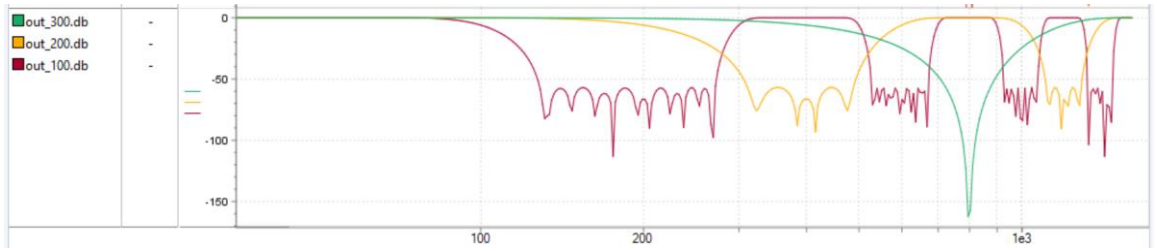


Using SystemC-AMS, much more “interactive” teaching because of significantly faster simulations (compared to SPICE, Simulink, …).

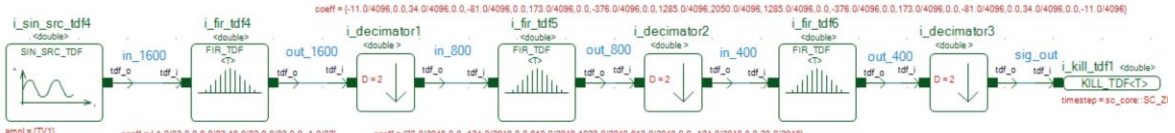
Digital signal processing example

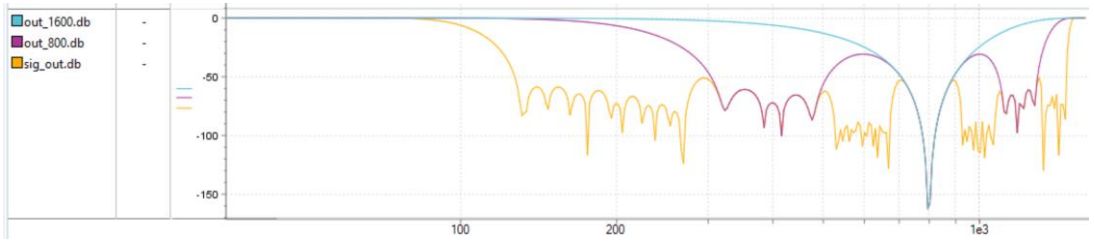
- Multirate FIR filters- transient and frequency domain analysis



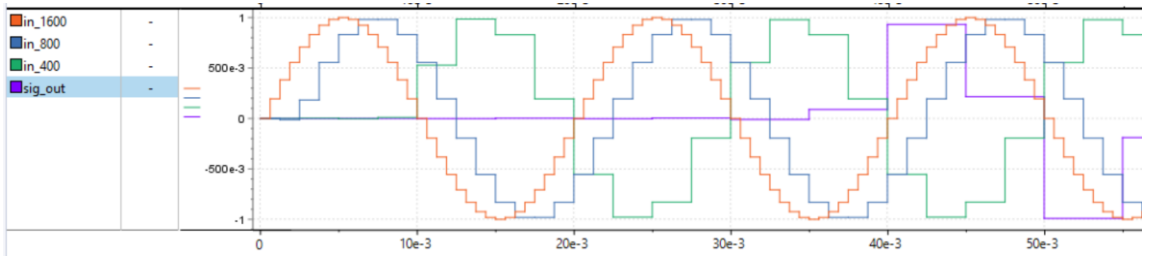


Graphical approach is extremely useful for learning such topics, especially when allowing direct visualisation of results as well.



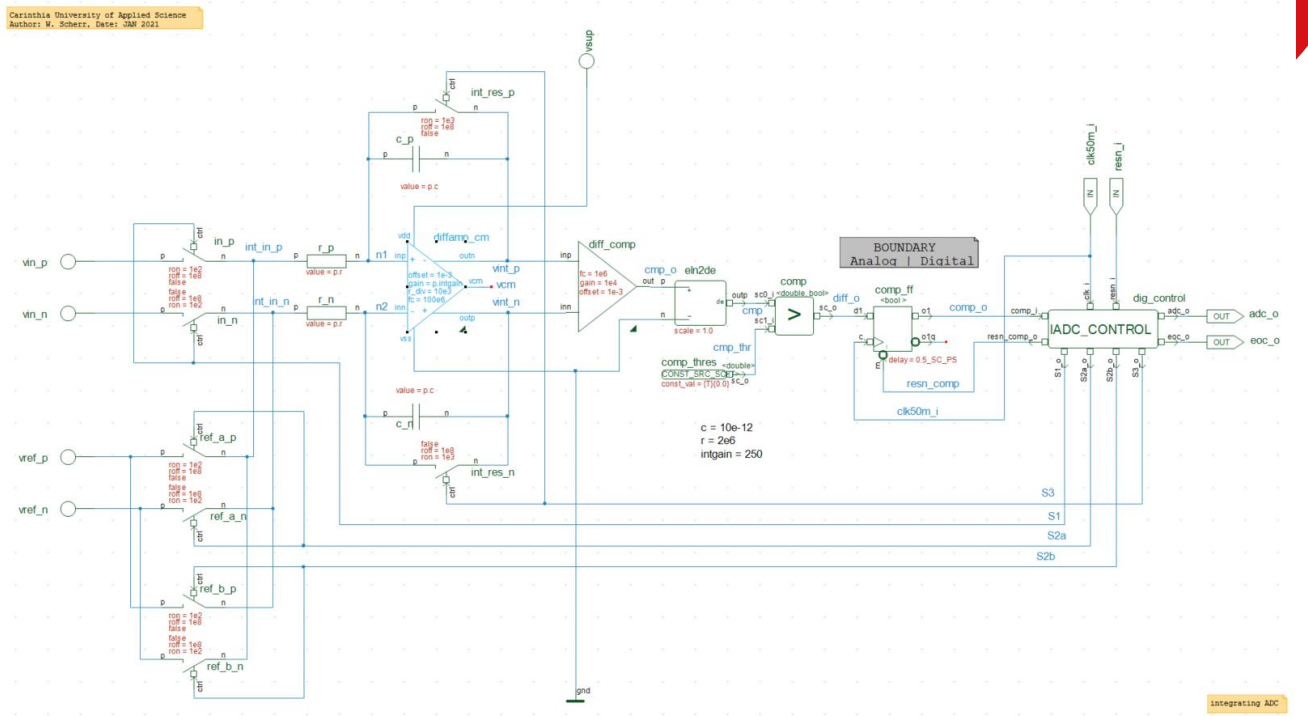


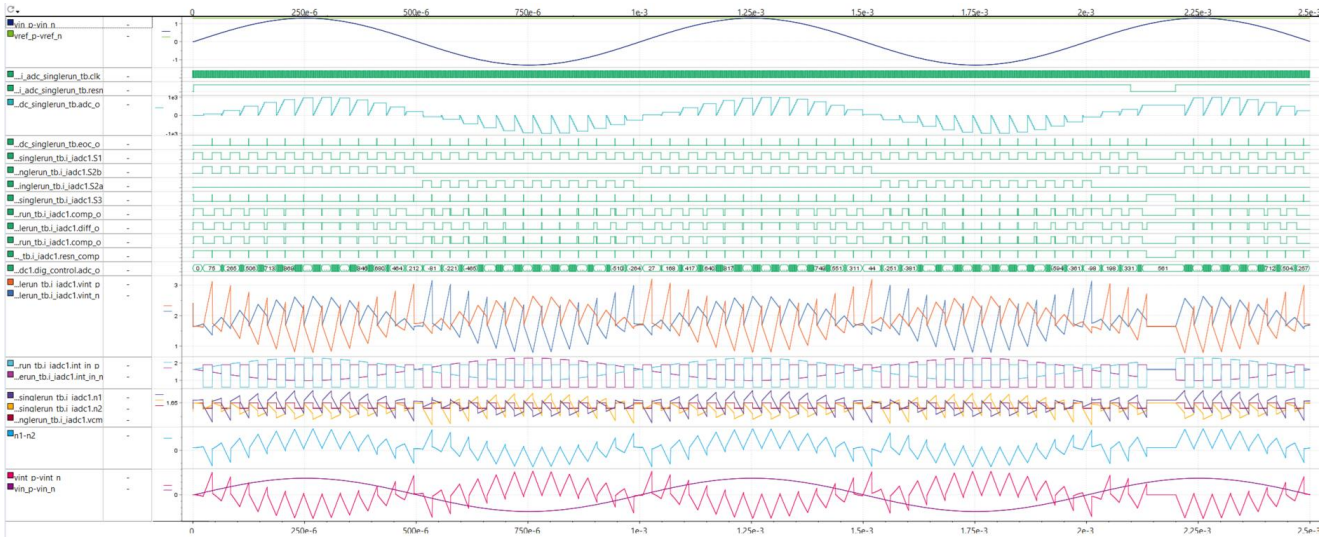
Transferring TDF models to HDL code is a straight-forward process afterwards (where simple AC analysis is also not possible).

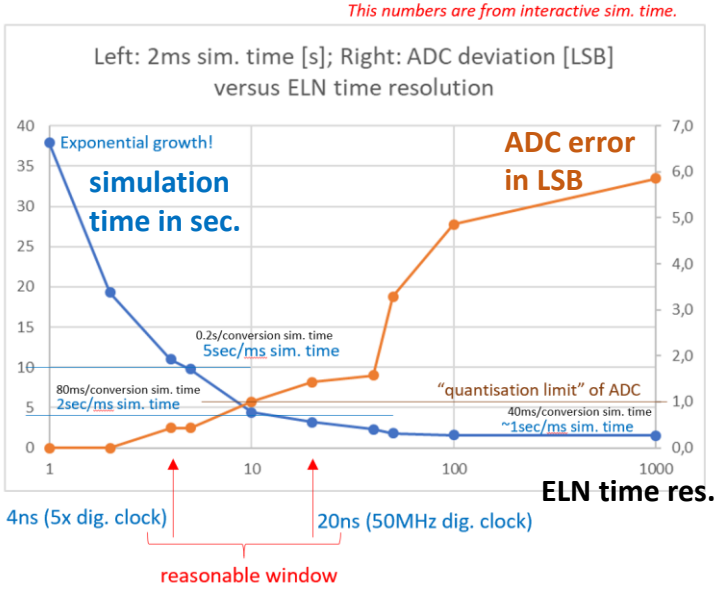


Mixed signal example

- Integrating ADC: study effects of block specs on overall performance
  - Hand calculations and estimations first
  - Then verify with models
  - Then implement design with proper checks in TSMC 65nm
- Study modelling trade-offs







Improvement potentials and what must be kept…

- Because models are C++ based, if something gets wrong, error messages can be quite “cryptic” for beginners → Coside already captures a lot!
  - The GUI approach is quite attractive to start-up quickly, C++ knowhow is not (really) needed
  - Thus, potential issues should be caught early → please keep up the good support here!
- Interactive AC analysis would be a really nice add-on…
  - Allow to set up min/max frequency and step and start AC analysis e.g. in a second tab of SCSimControl (ideally, also at any time during an active transient run)
- Python support is great and appreciated!
  - Python is often requested by students; it is an easy to learn scripting language and commonly used
- Model exchange (on “atomic” block level) is not trivial and quite proprietary…
  - Students would like to exchange sometimes single blocks (like they can easily do in Simulink, as it is usually a single file), but several files, namespaces etc. are “connected” to those blocks
  - Simple “file copy & paste” is dangerous, import/export from/to projects is a bit cumbersome
    - Both options can anyhow easily “mess up” a given Coside project
    - Not to speak of blocks made with other Coside versions (prev. semesters) when trying to copy them in…

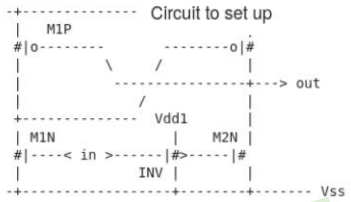
Agenda

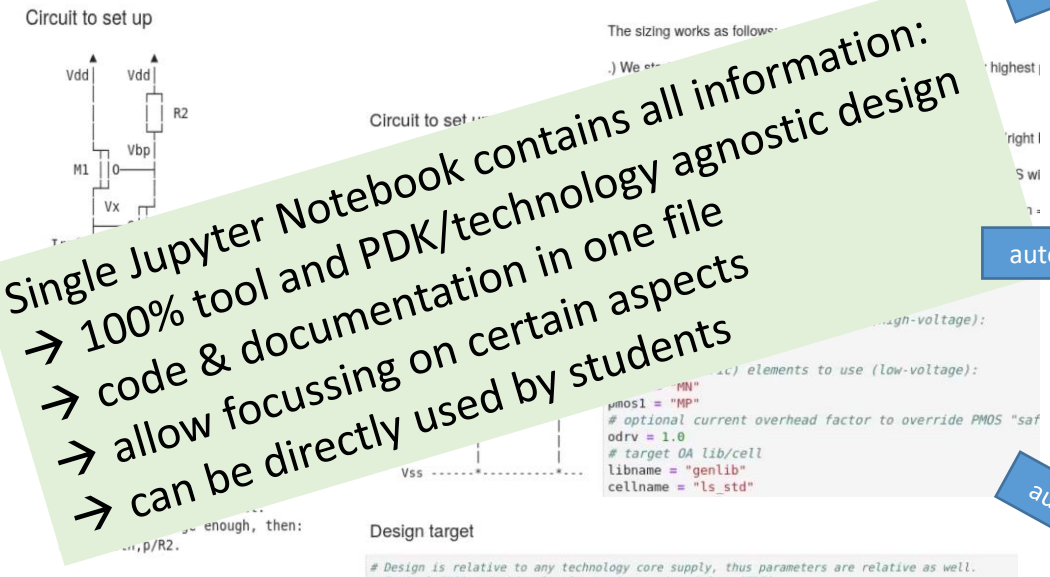
- Overview of Carinthia Institute for Microelectronics and the Integrated Systems and Circuits Master program

- The role of SystemC (AMS) and Coside as integral part of the Master program plus some use cases

- Outlook

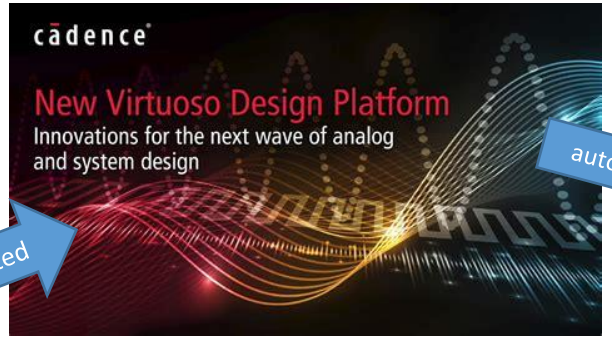
Research on automation also towards education



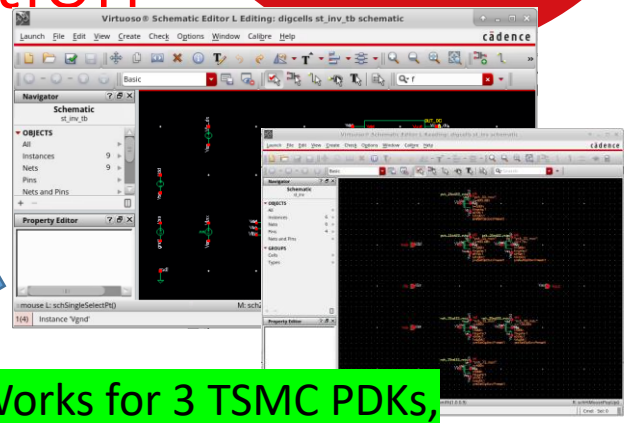


Single Jupyter Notebook contains all information:
→ 100% tool and PDK/technology agnostic design
→ code & documentation in one file
→ allow focussing on certain aspects
→ can be directly used by students

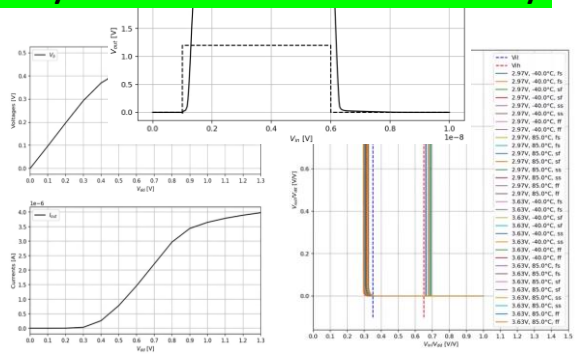
automated



automated



Works for 3 TSMC PDKs, partly used in lectures already.



E.g. basic analogue circuit blocks including PDK independent sizing.

automated

Open version (using GPDKs and open tools) in preparation.

E.g. models of (any) circuit blocks, more advanced CPU/digital setups.

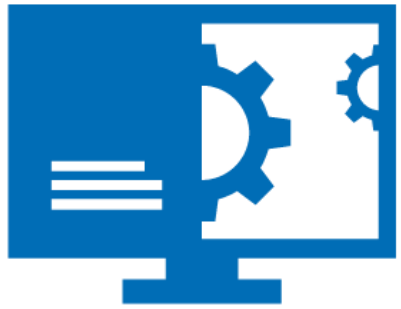
automated

automated

Ongoing, not yet used in lectures. Looking forward to Python extensions.

COSEDA technologies

KÄRNTEN
University of
Applied Sciences



THANK YOU FOR YOUR ATTENTION

COSEDA
usergroupmeeting