End-to-End Bandgap Reference Design: From Schematic to Layout Essentials

Leibniz Institute for high performance microelectronics

Phillip Ferreira Baade–Pedersen (Design Engineer)
Dr. Ing. Christian Wittke (Scientist)
Frankfurt (Oder), IHP - 20/05/2025

Projects: BMBF -> FMD-QNC (16ME0831)

Agenda For today

09:00 - 09:15 | Recap | Questions?
Review of Key Takeaways from Yesterday

09:15 – 09:30| Module Overview
Overview and Introduction to Today's Module

09:30 - 10:45 | Expert Talk
Dr. Prabhat Kumar Dubey: Insights on Model Generation in the Open PDK

10:45 - 12:00 | Introduction | Hands On
Designing the Testbench for OTA and Bandgap Reference

12:00 - 13:00 | Lunch Brake | Catching Up

13:00 - 15:00 | Continuation of Previous Session's Work
Completing Testbenches with Mismatch Analysis and Starting Layout Design

15:00 – 15:30 | Coffee Break | Caching Up

15:30 - 17:00 | Hands-On Session
Guided Layout Session or Continuation of Testbench Design and Simulations

Leibniz Institute for high performance microelectronics

Key Takeaways from Yesterday

- Introduction to the IHP Open PDK
- Learning the basics in Xschem related to simulations
- Basic Klayout maneuvering
- First examples on physical verification

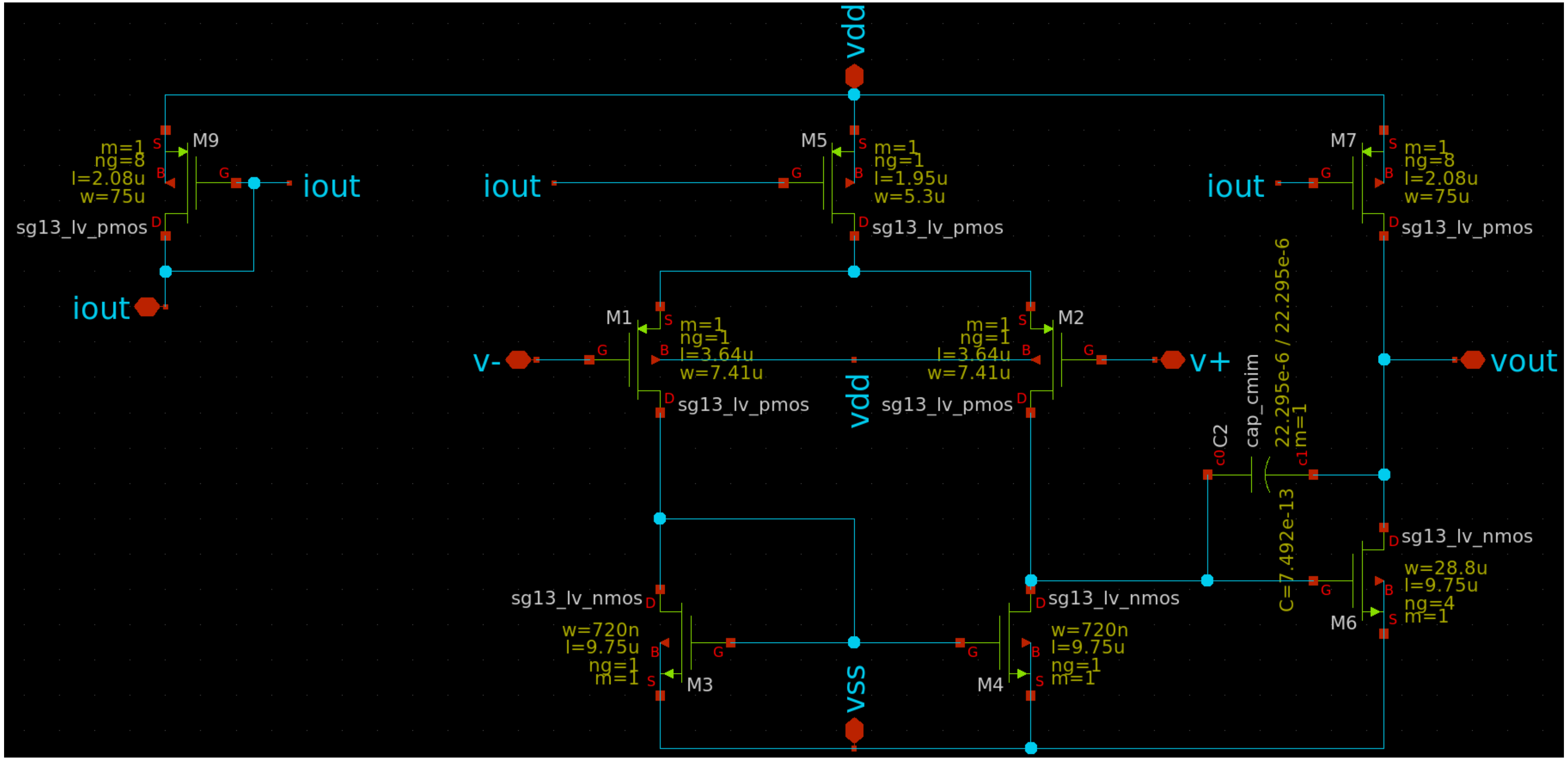
Questions?

- Was something from yesterday unclear?
- Are we going to fast?
- Do you need more time for the small exercise?

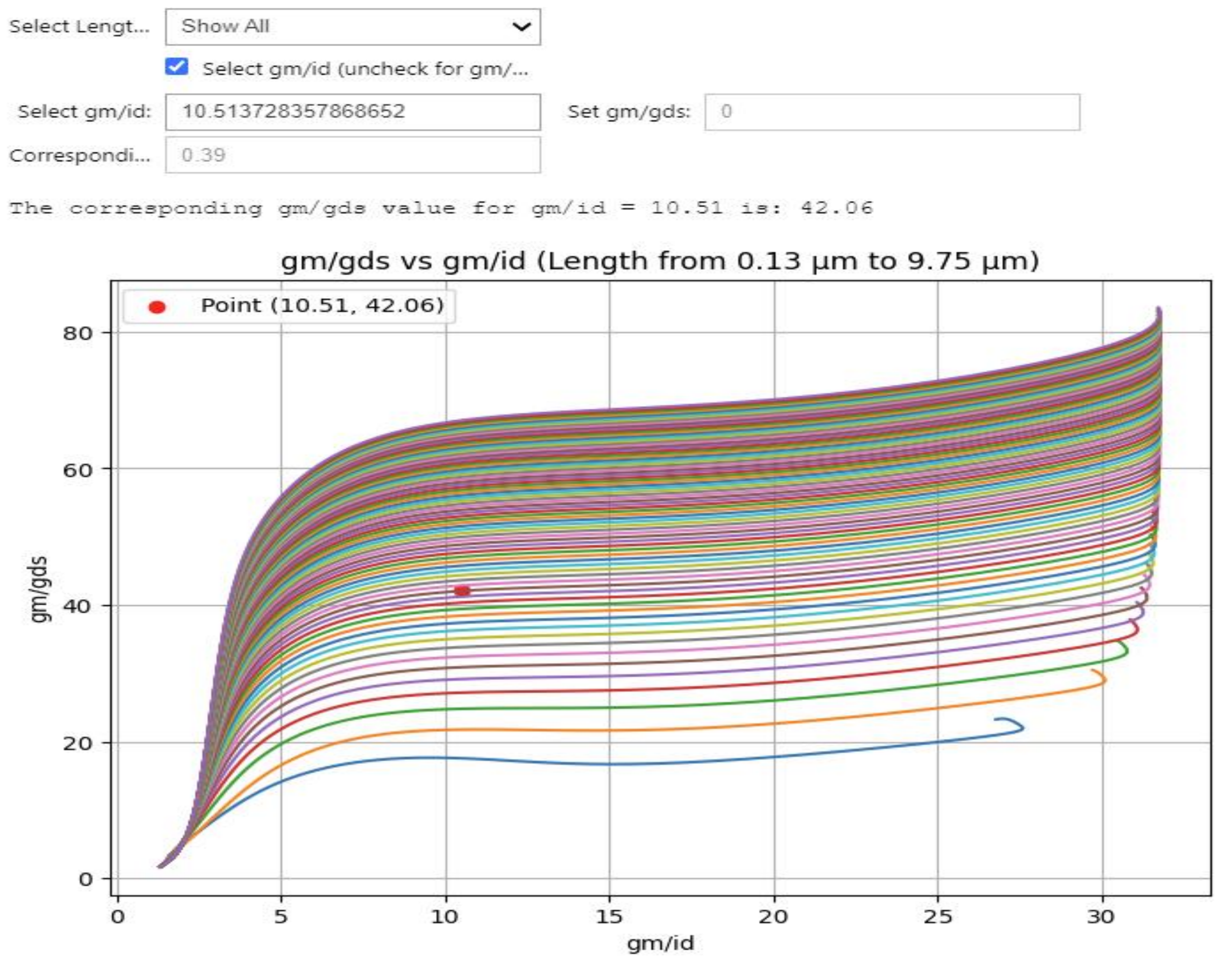
Overview Of Todays Plan

- **OTA Design & Analysis:** Perform DC and AC analysis to evaluate key design metrics.
- **Bandgap Voltage Reference:** Conduct DC, transient, and mismatch analysis using Monte Carlo sampling.
- **From Design to Tape-Out:** Explore the final layout and discuss the steps required for tape-out in the open-source domain.
- **Layout Competition:** Compete to design the best OTA layout with the fewest DRC/LVS errors.

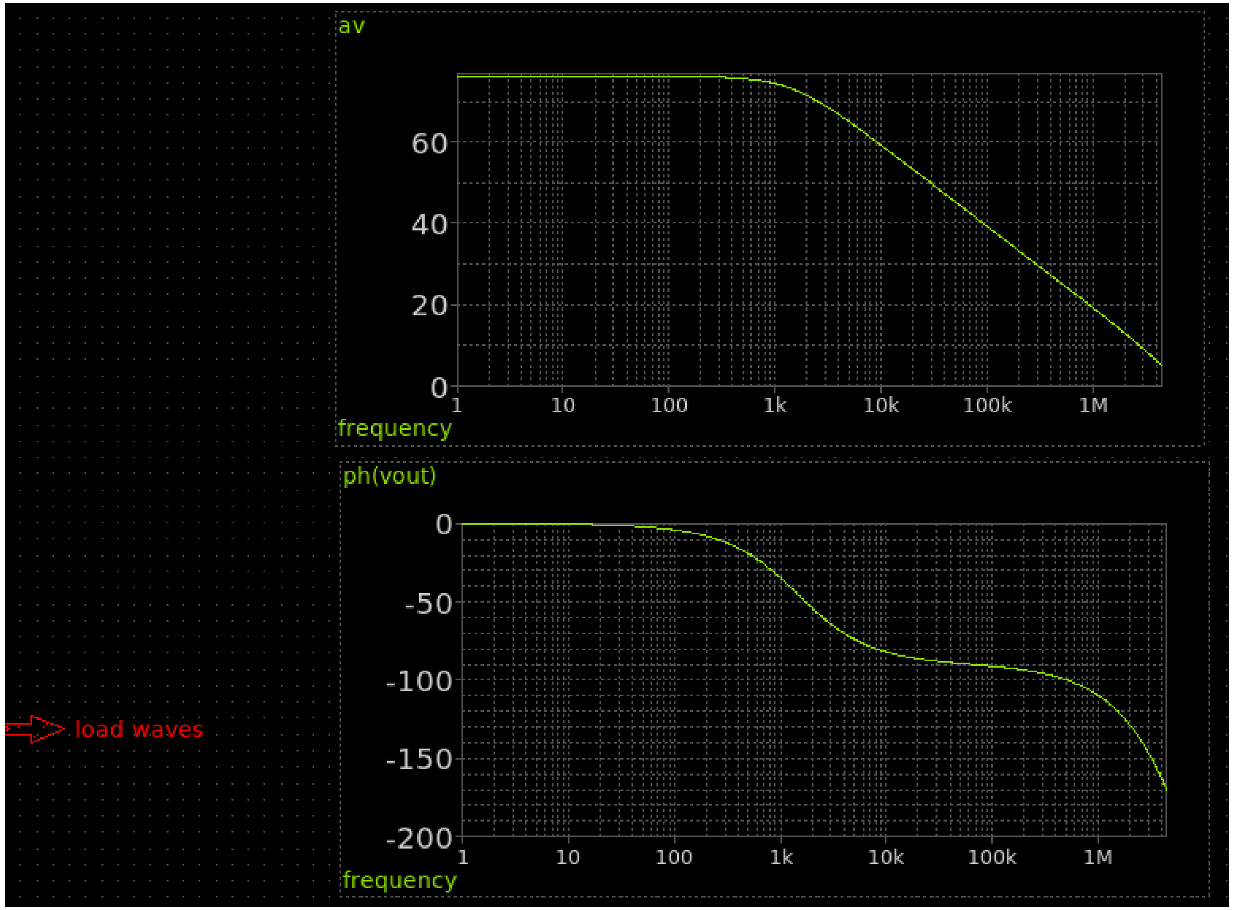
OTA Design (Two Stage Miller Compensated)

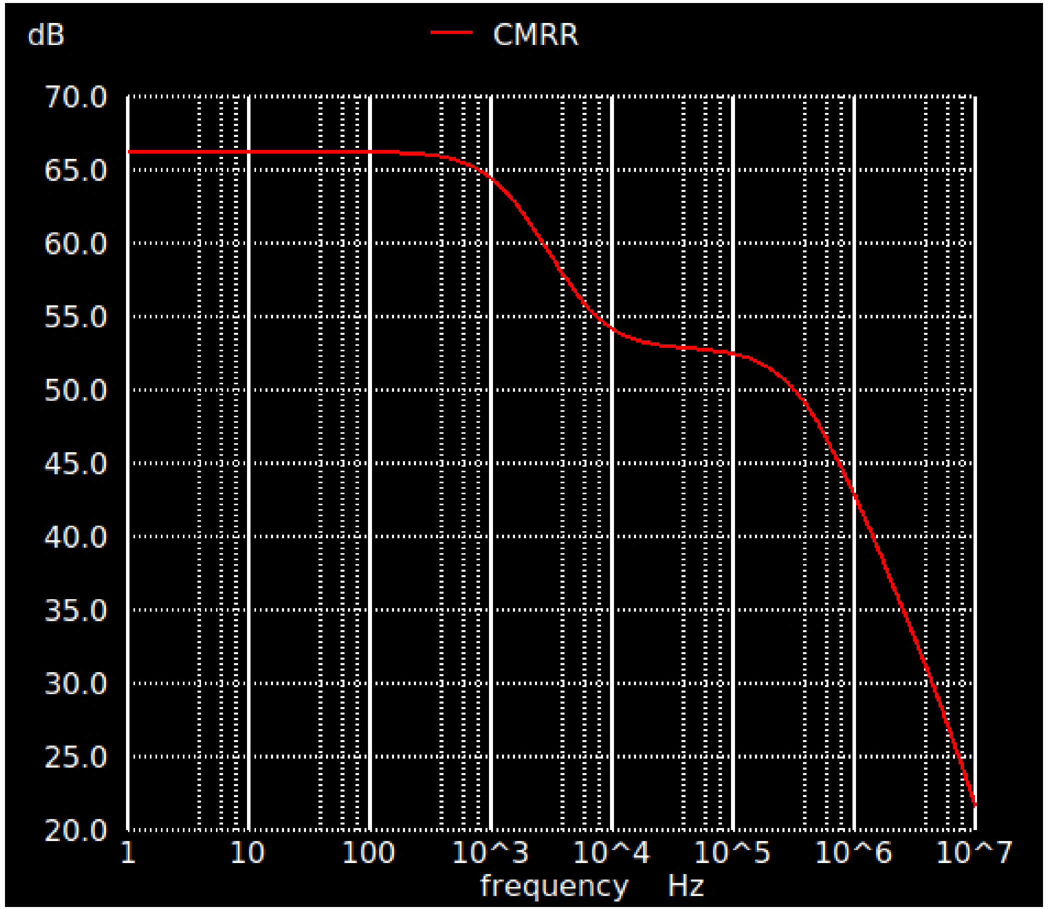


Design Methodology In OS (Gm/ID)

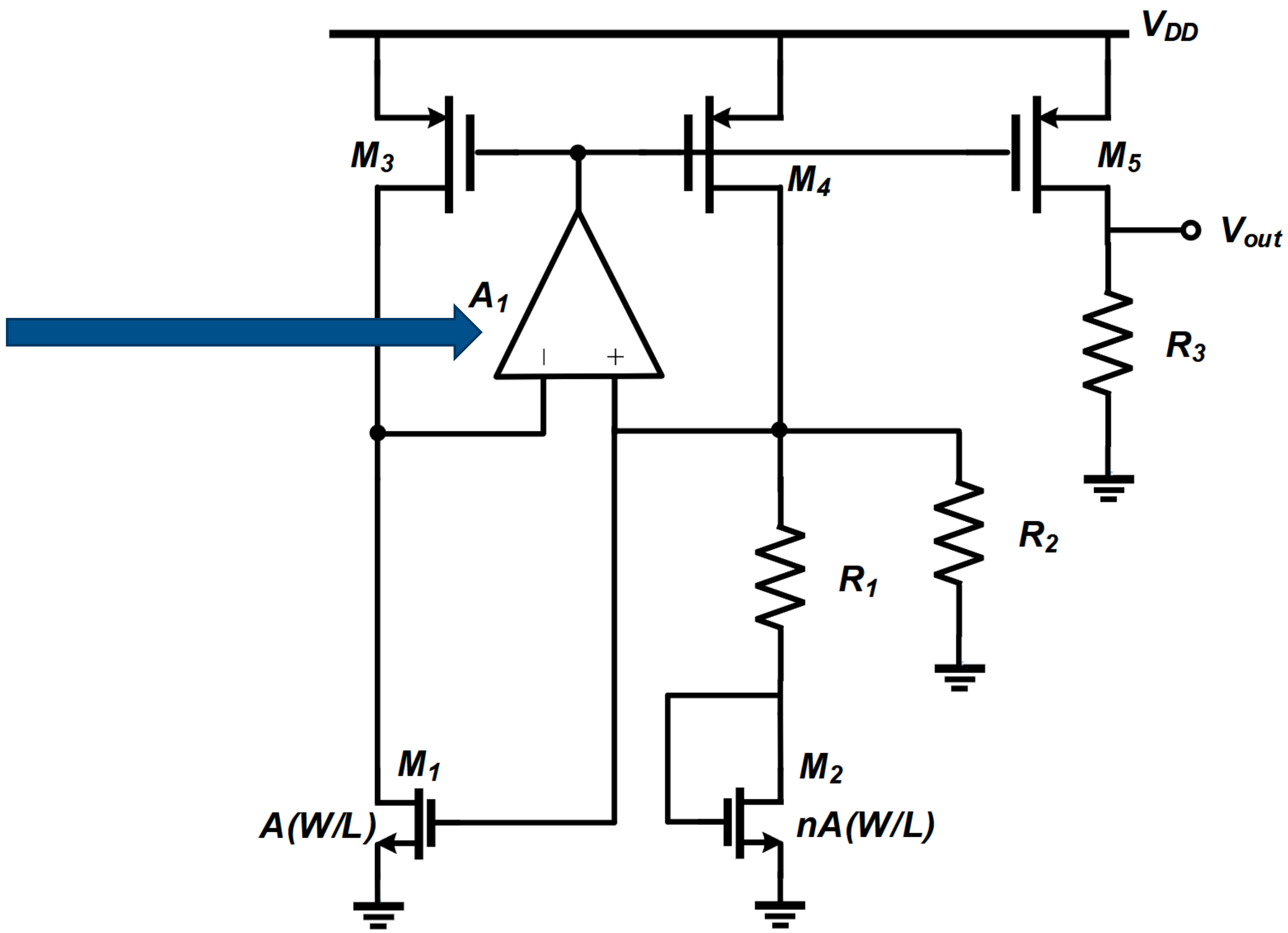


Testbenches

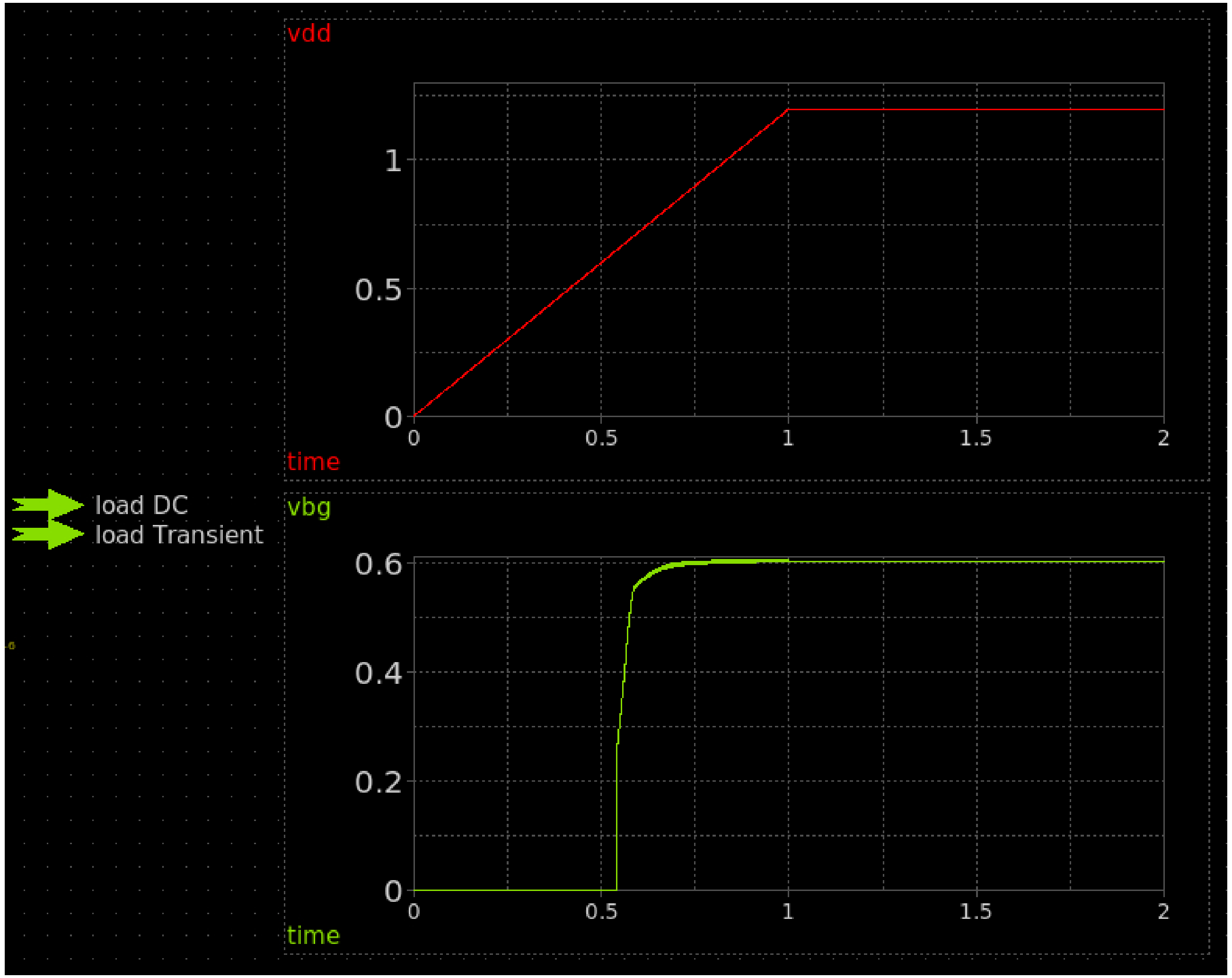


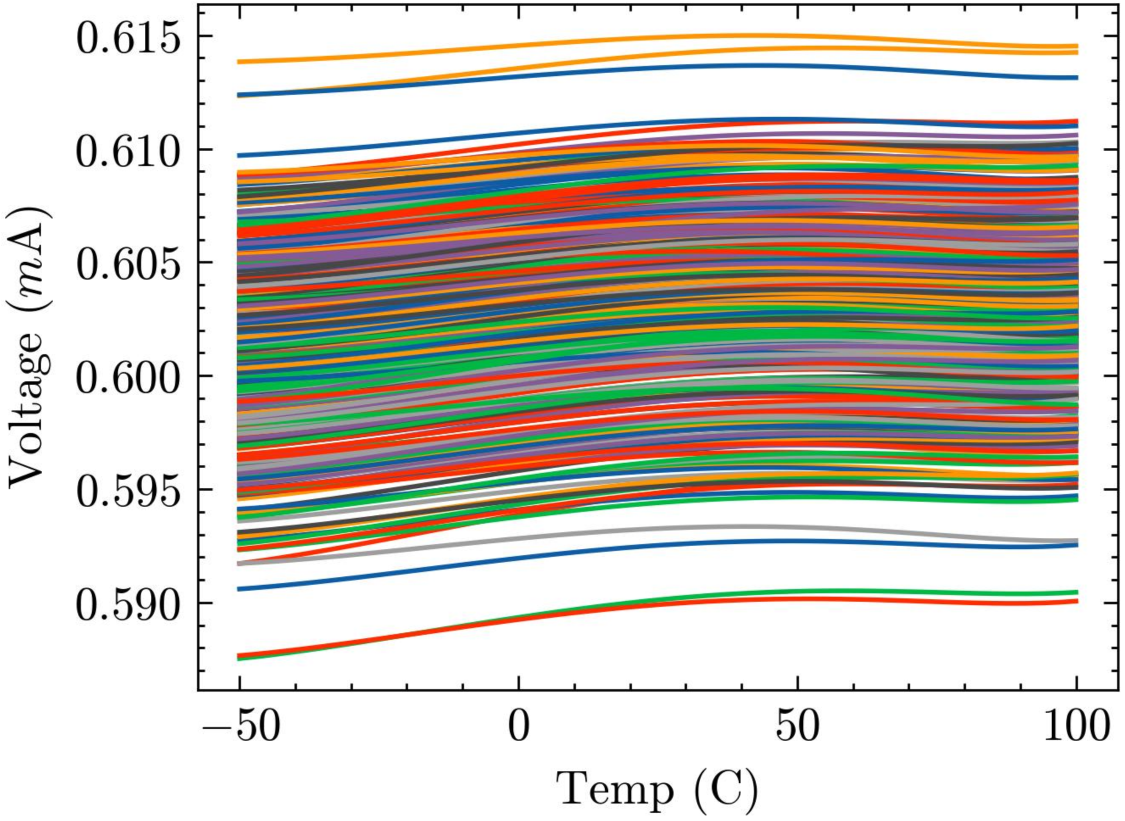


Bandgap Reference

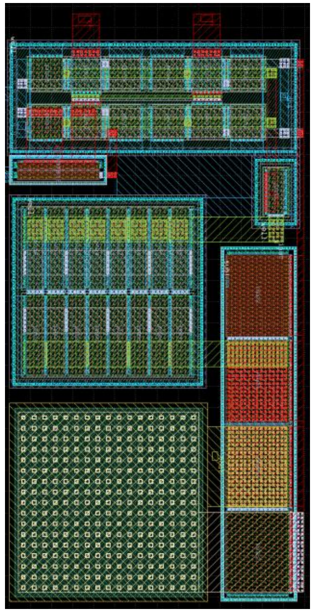


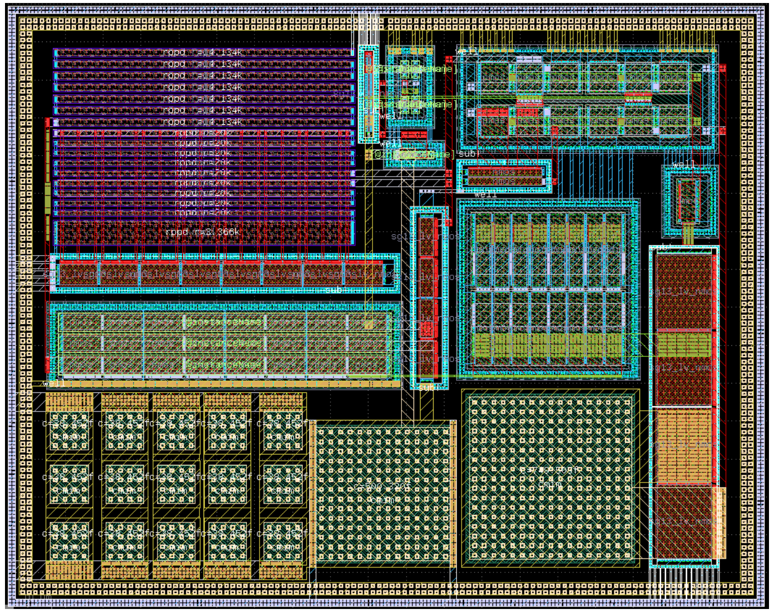
Evaluating Overall Performance

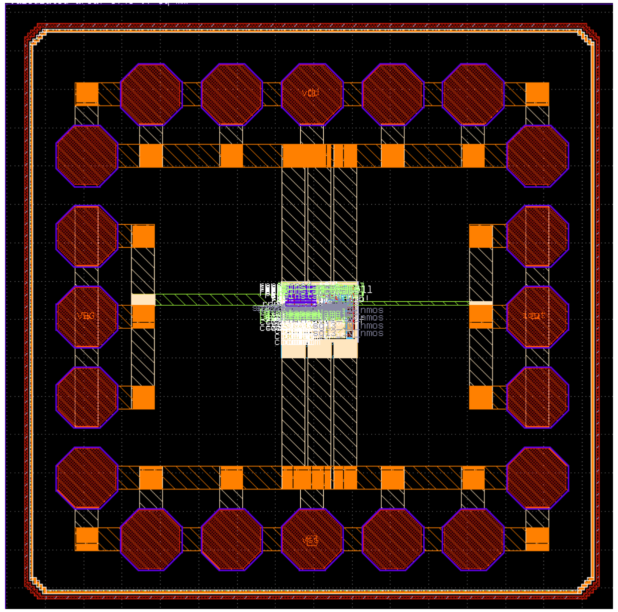




Bandgap Reference (Layout)







Expert Talk

Dr. Prabhat Kumar Dubey

Insights on Model Generation in the Open PDK

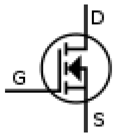
Leibniz Institute for high performance microelectronics

Device Modeling and Test Chip Characterization

Prabhat Kumar Dubey

IHP Microelectronics, Frankfurt Oder Germany

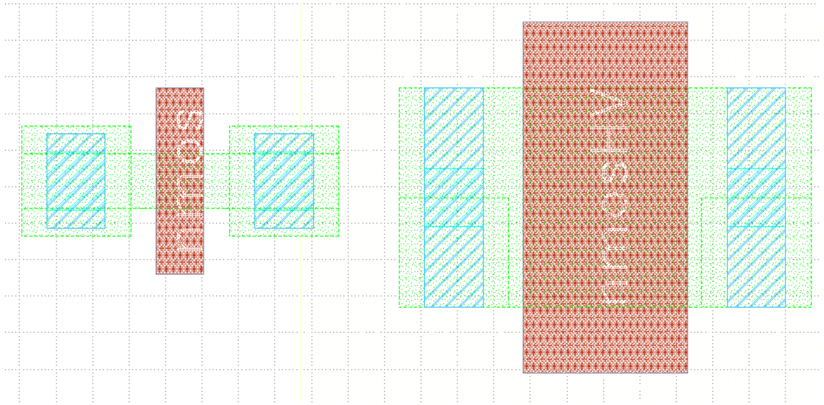
OpenPDK Device/Cell



Symbol View
(Xschem/Quecs)

* SPICE instance NMOS Card
N1 (d g s b) nmos <i_par=value>
* Model Card
.model nmos psp <m_par=value>

Spice View



Layout View/PCells

SPICE Models at IHP Microelectronics

PSP Model:
- nmos/pmos
- nmosHV/pmosHV
- rfnmosHV/rfpmosHV
- rfnmosHV/rfpmosHV
- SVaricap

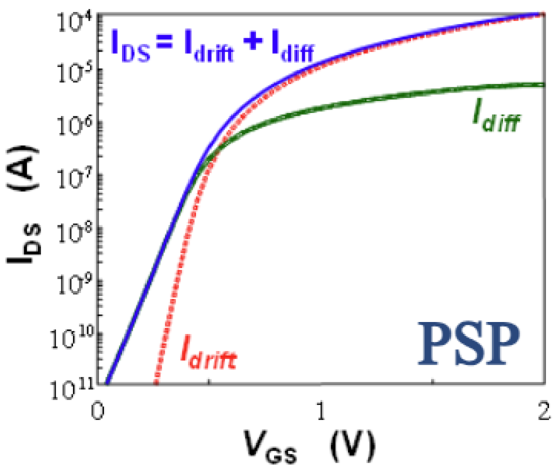
HiCUM Model:
- npn13G2 transistors
- npn13G2L transistors
- npn13G2V transistors

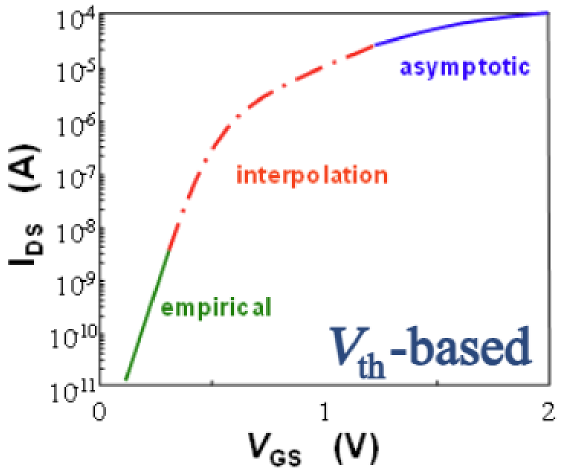
VBIC Model:
- HBT npn transistors
- npn13G2L transistors
- npn13G2V transistors

VIBIC model could be used up to 10GHz frequency above that it can give inaccuracy in results.

PSP Model: Overview

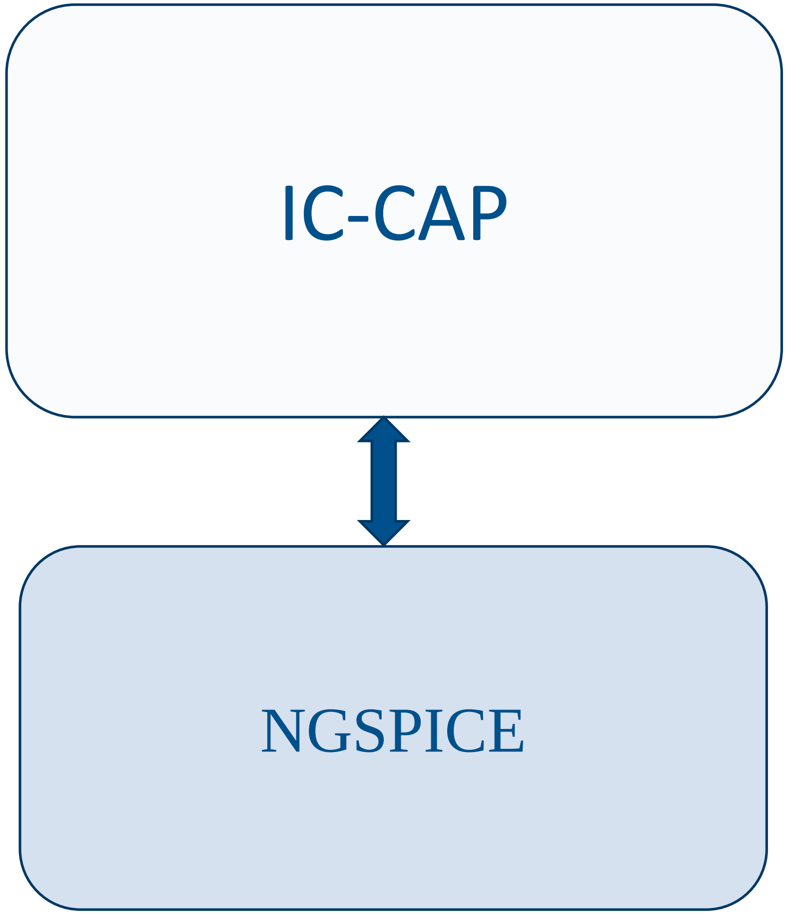
- PSP is a surface potential based compact model
  - Nonuniform doping
  - Electric field dependent mobility/ velocity saturation
  - Channel length modulation/DIBL, series resistance
  - Short channel effect, narrow width effect
  - Gate poly depletion, quantum corrections, overlap capacitance
  - Impact ionization, gate-leakage, GIDL/GISL
  - Junction Diode I-V, C-V, diode reverse breakdown
  - Noise (1/f, thermal, gate induced, shot noise)
  - Non-quasi-static effects
  - Gate and bulk resistances
  - STI stress effect

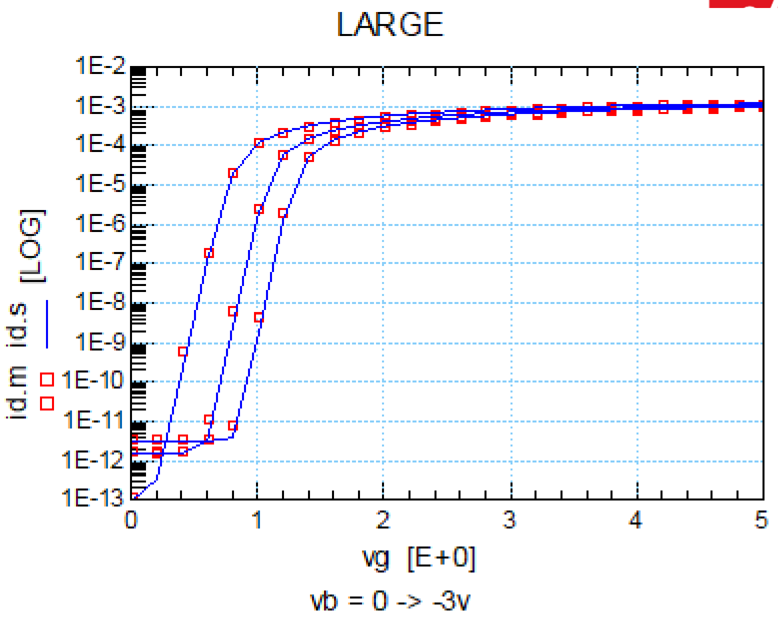


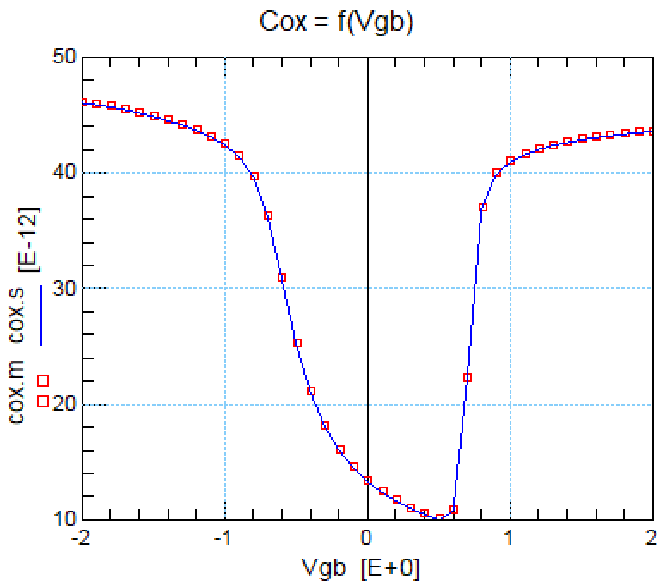


[REF] G. Gildenblat; PSP family of compact models overview and recent development. MOS-AK 2008 13 Dec. 2008

Parameter Extraction

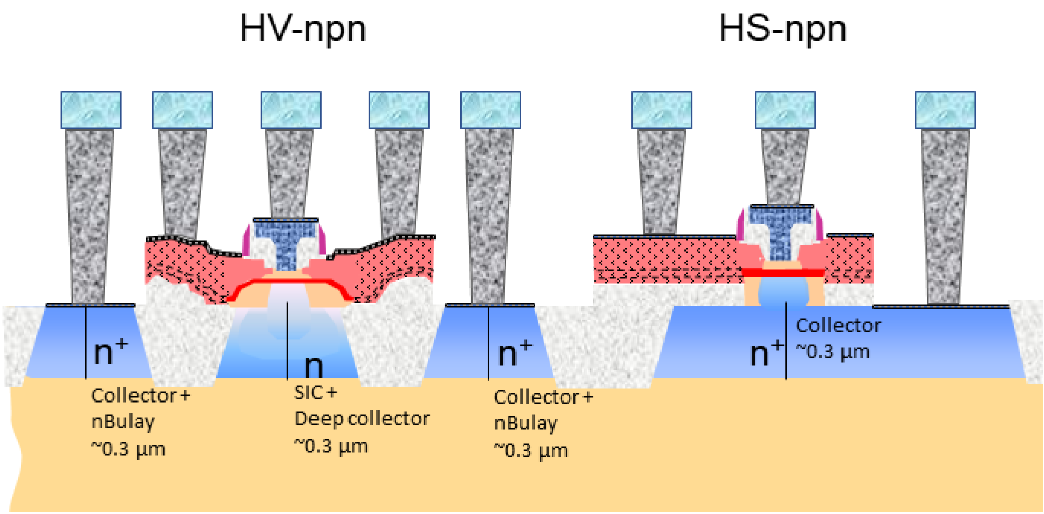






VBIC Model

- BJT model that was developed as public domain replacement for SPICE Gummel-Poon model
  - Improved Early effect modeling
  - Quasi-saturation modeling
  - Parasitic substrate transistor model
  - Parasitic fixed capacitance model
  - Avalanche multiplication model
  - Decoupled base current from collector
  - Electrothermal Modeling
  - Smooth, continuous model

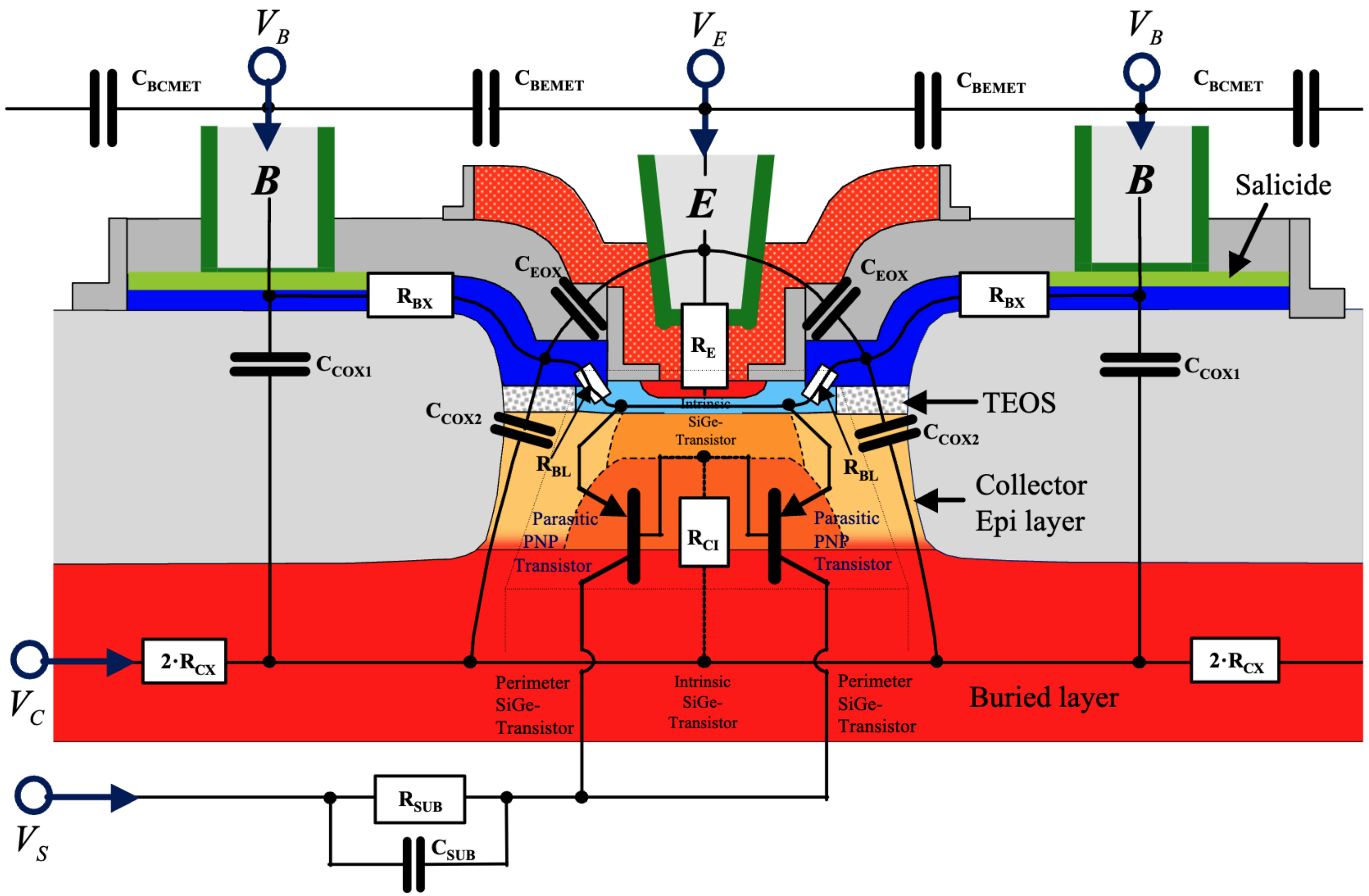


High Current Model (HiCUM)

- Standard compact model for BJTs and HBTs
- Applicable for SiGe HBTs and InP HBTs
- Useful for high current densities and high frequency

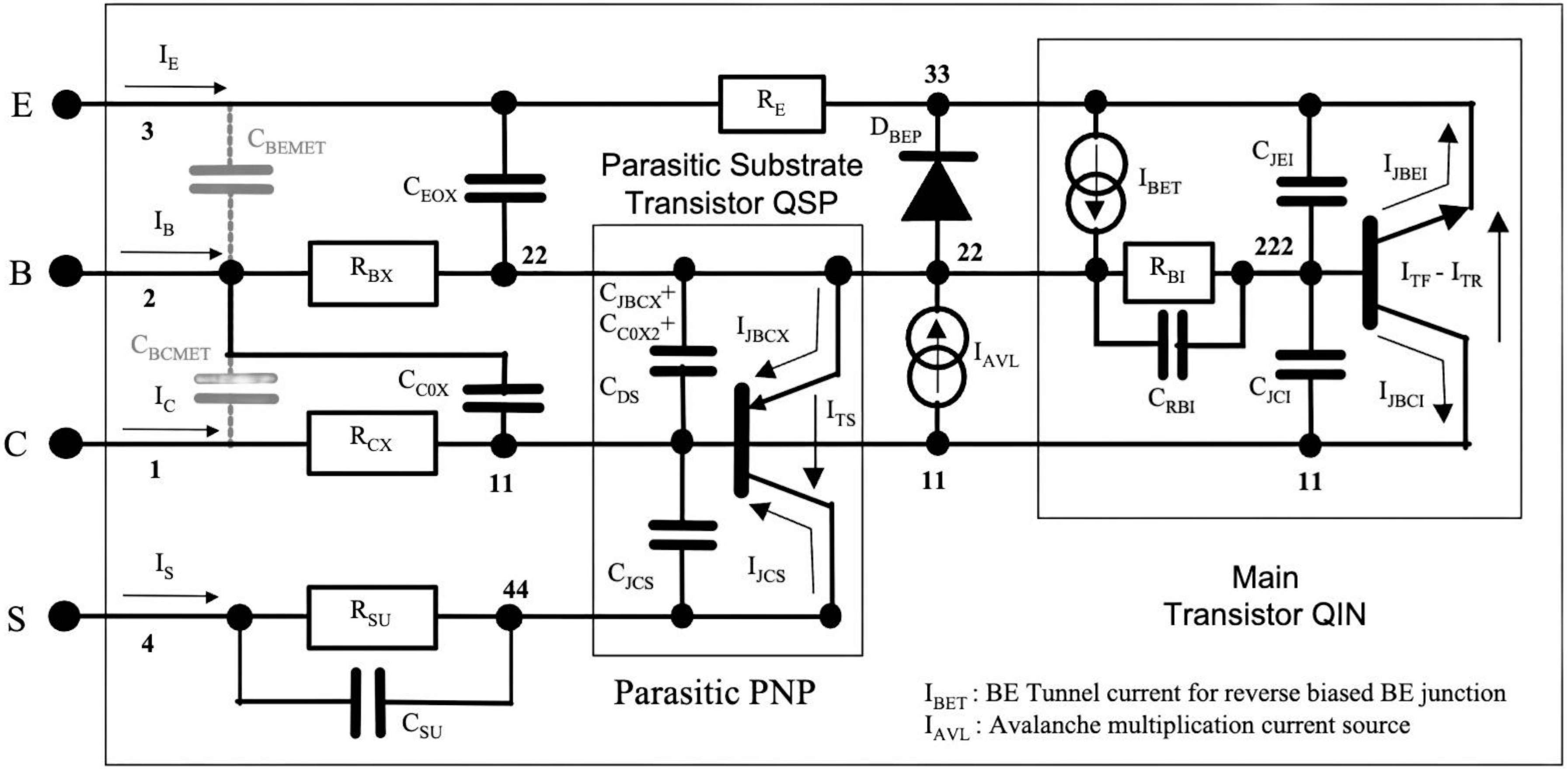
  - Self-heating, high-injection effects, and quasi-saturation
  - Scalable for different transistor geometries
  - More accurate than VBIC model at high current and high frequency applications

SiGe NPN Cross-Section



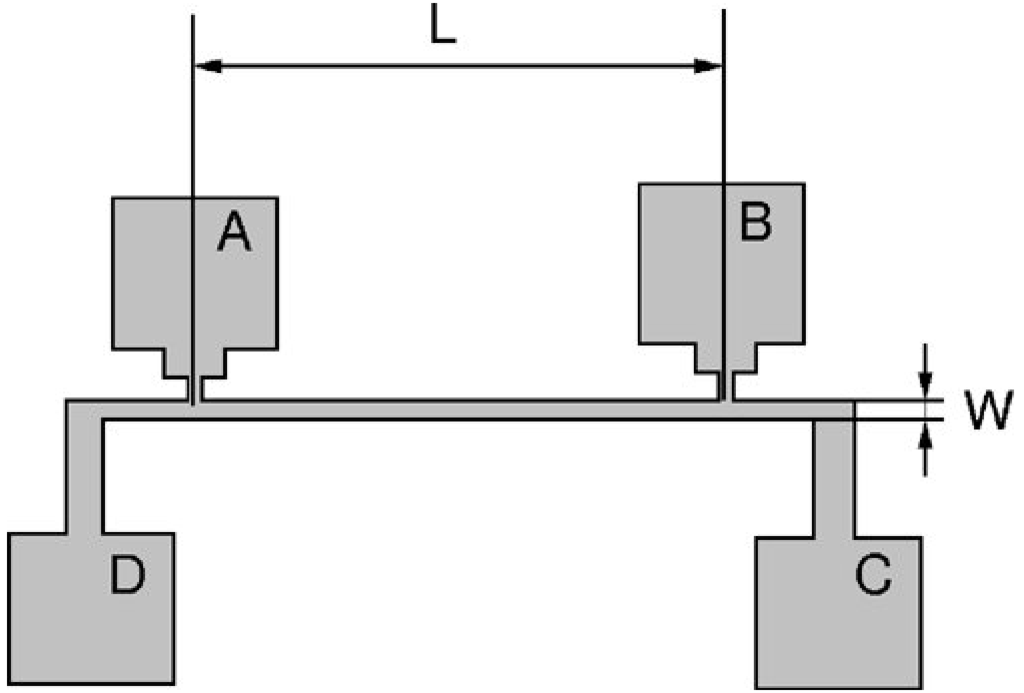
[REF] Pietro Brenner; Test Structures and a new approach for fast and accurate extraction of specific NPN parameters. Bipolar Arbeitskreis 2002 7. Oct. 2002

HICUM transistor subcircuit model for SiGe NPN HBT



[REF] Pietro Brenner; Test Structures and a new approach for fast and accurate extraction of specific NPN parameters. Bipolar Arbeitskreis 2002 7. Oct. 2002

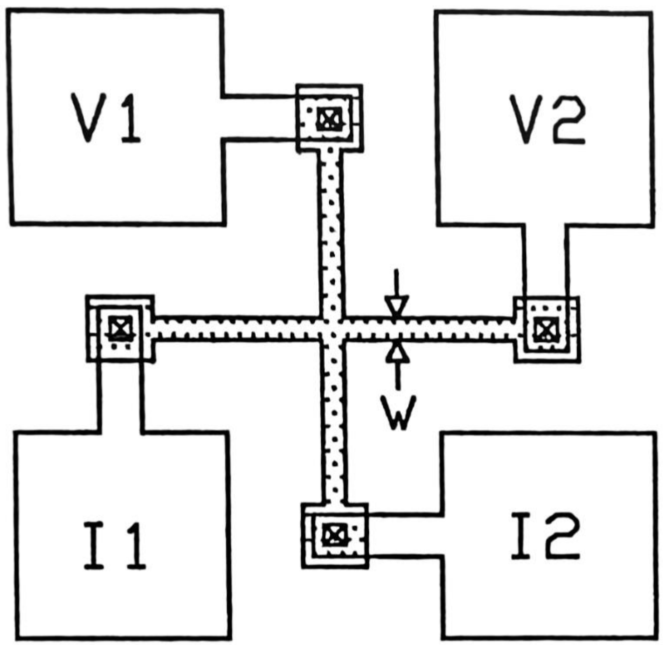
Bridge test structure for sheet resistance measurement

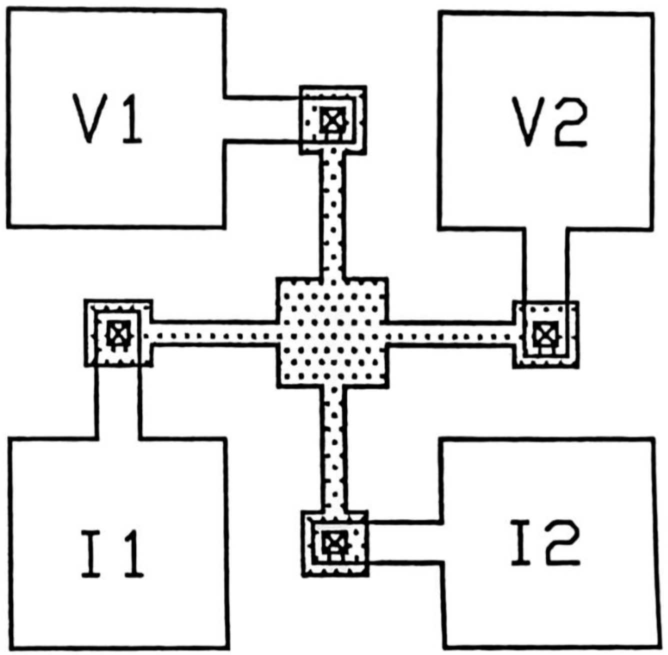


Bridge resistor measurements use kelvin contacts, where current is injected at taps that are displaced from voltage measuring taps.
The distance between the taps, L, is taken at the center line of the voltage taps, thus over etching of the conducting layer does not affect L provided the taps are etched identically

[REF] S. Enderling et al., "Sheet resistance measurement of non-standard cleanroom materials using suspended Greek cross test structures," in IEEE Transactions on Semiconductor Manufacturing, vol. 19, no. 1, pp. 2-9, Feb. 2006, doi: 10.1109/TSM.2005.863248.

Greek and Box Cross Resistor



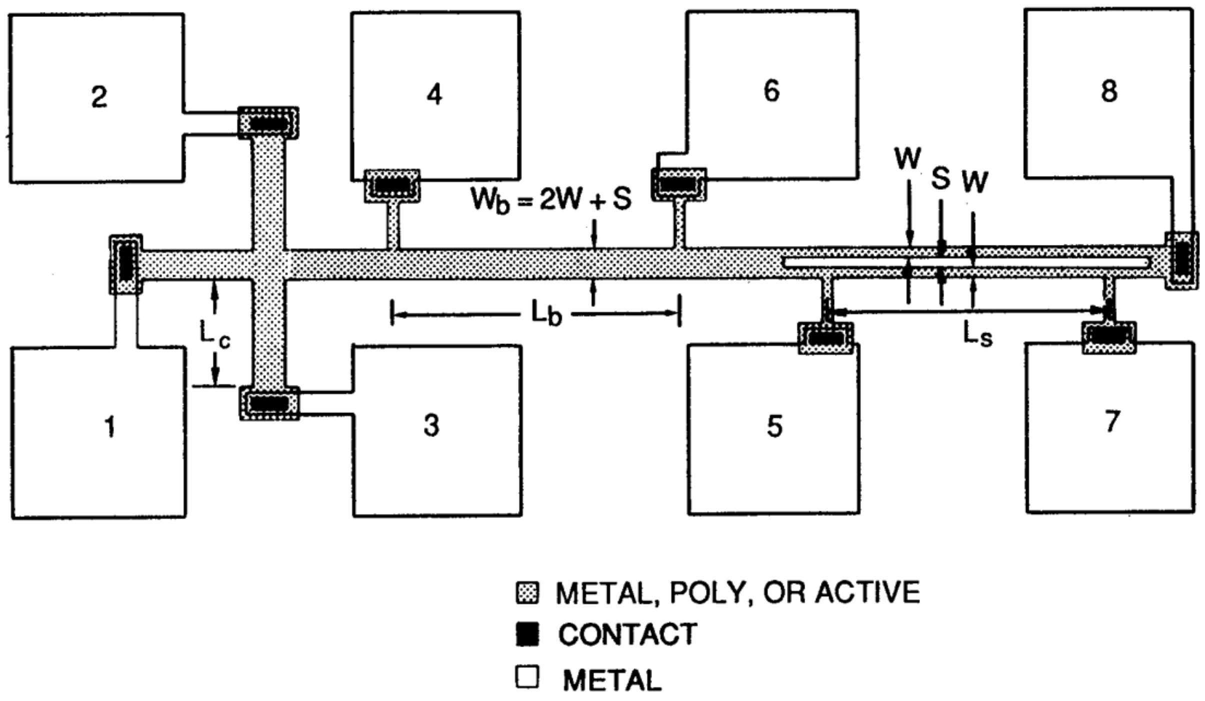


BOX CROSS

Greek cross resistor provides a measure of the local sheet resistance; if the layer contains grains that are equal to or larger than the line width, the sheet resistance will be highly variable from structure-to-structure.

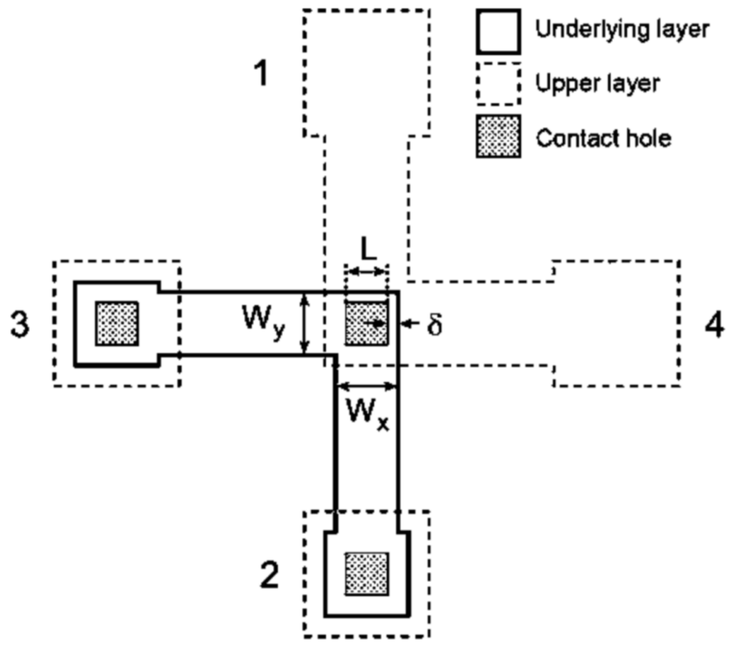
Box cross resistor provides a measure of the average sheet resistance.

Split-Cross-Bridge Resistor



Split-Cross-Bridge Resistors are used to measure sheet resistance as well as width, spacing, and pitch of conducting lines designed to their minimum rule.

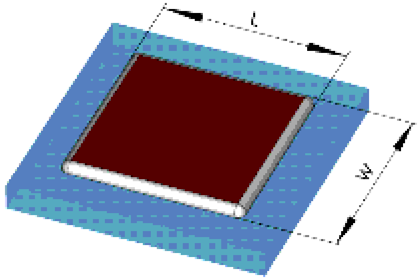
Contact Resistor

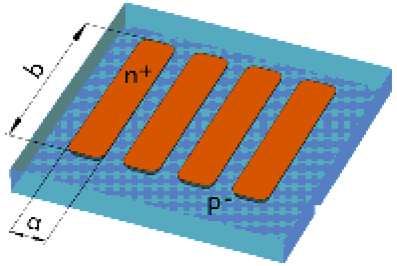


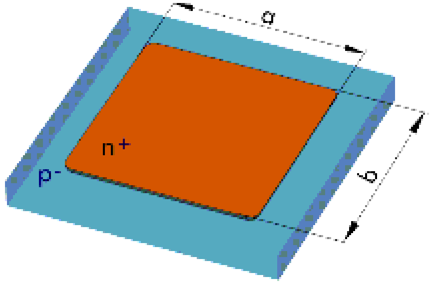
Contact (Cross-Bridge Kelvin Resistor: CBKR) structures for low specific contact resistances

[REF] Scorzoni, Andrea, Finetti, Kaj Grahn, Ilkka Suni, and P. Cappelletti. "Current crowding and misalignment effects as sources of error in contact resistivity measurements—Part I: Computer simulation of conventional CER and CKR structures." IEEE transactions on electron devices 34, no. 3 (1987): 525-531

Test Structures for CV Measurements







C_Oxide_m (Gate oxide)

Large area MOS capacitor

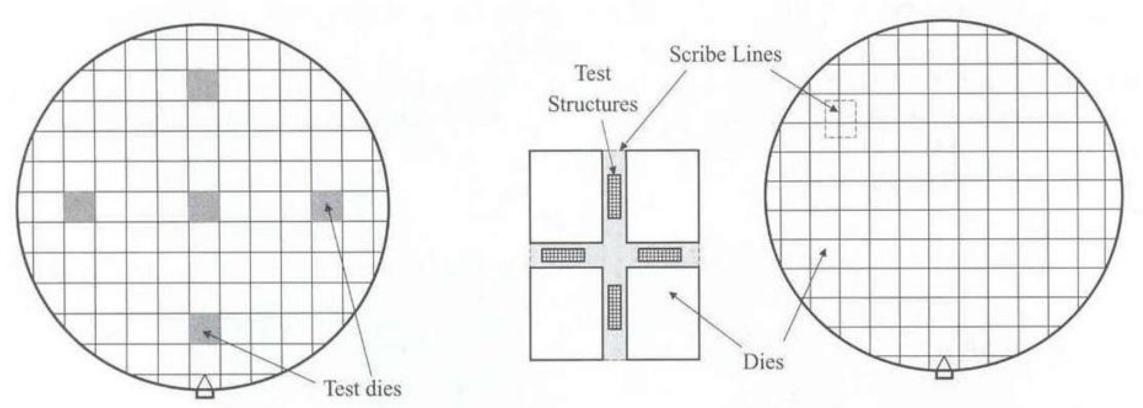
Diode_Perim_m

Finger diode with a large perimeter and a small area (shown here for an n-type device)

Diode_Area_m

Area diode with a large area and a small perimeter (shown here for an n-type device)

PCM Test Chips/Dies vs. Scribe Lines Monitors

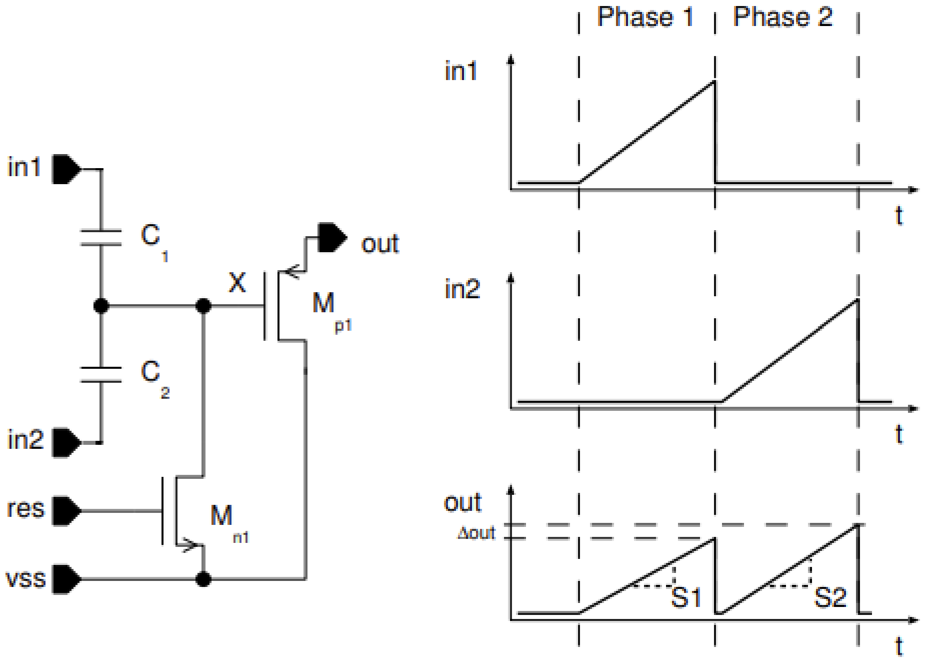


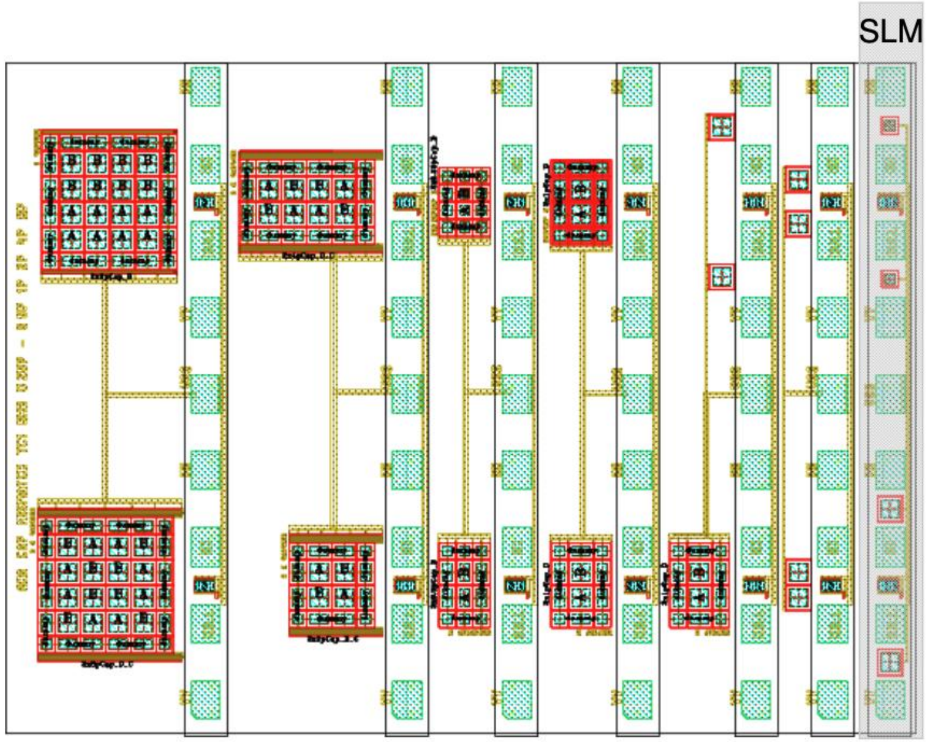
Test Wafer

Production Wafer

[REF] Udo Sobe, Karl-Heinz Rooch and Dietmar Mörtl; Simulation and Analysis of Analog Circuit and PCM (Process Control Monitor) Test Structures in Circuit; Design ZMD Zentrum Mikroelektronik, Dresden AG

Capacitor Mismatch

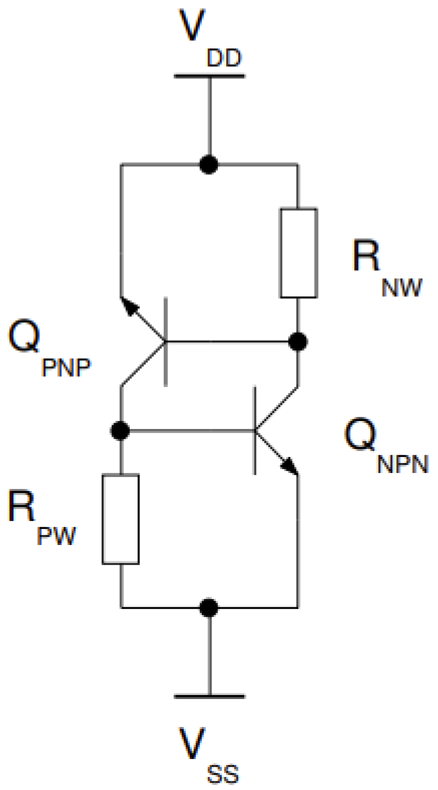


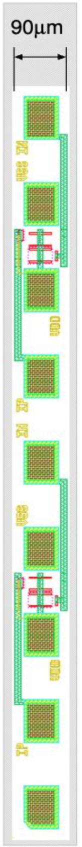


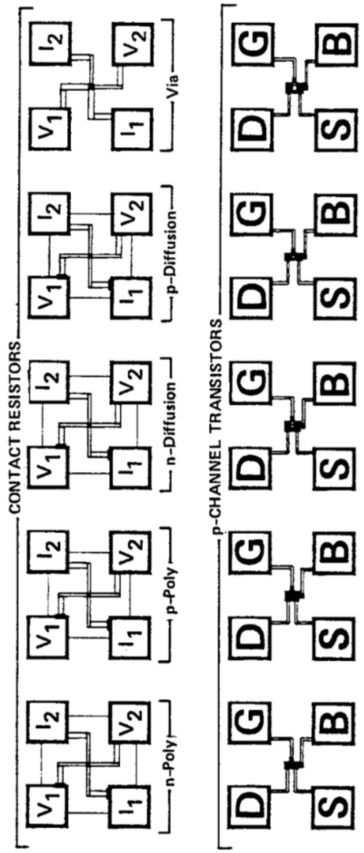
$$\Delta C = C_1 \frac{\Delta S}{S_1} \quad \text{with} \quad \Delta S = S_2 - S_1, \Delta C = C_2 - C_1$$

[REF] Udo Sobe, Karl-Heinz Rooch and Dietmar Mörtl; Simulation and Analysis of Analog Circuit and PCM (Process Control Monitor) Test Structures in Circuit; Design ZMD Zentrum Mikroelektronik, Dresden AG

SLM, Standard Pad Frames (2 by N) and Test Chip

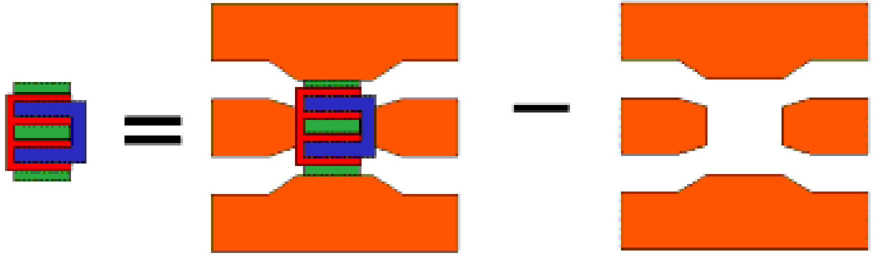






[REF] Udo Sobe, Karl-Heinz Rooch and Dietmar Mörtl; Simulation and Analysis of Analog Circuit and PCM (Process Control Monitor) Test Structures in Circuit; Design ZMD Zentrum Mikroelektronik, Dresden AG

RF characterization and De-embedding procedures



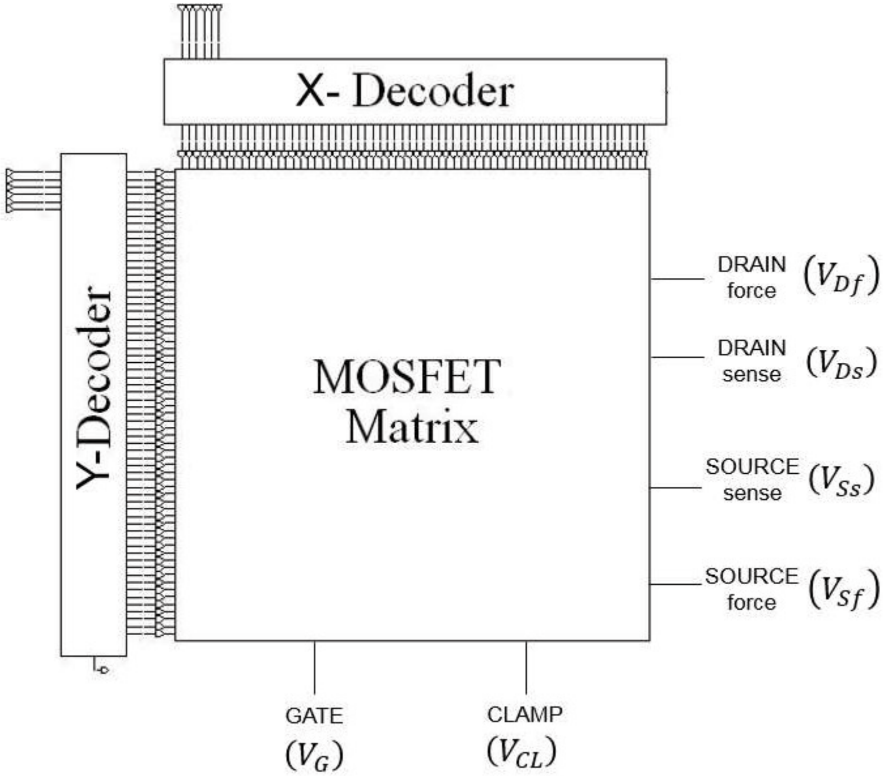
$$Y_{DUT} = Y_{Total} - Y_{OPEN}$$

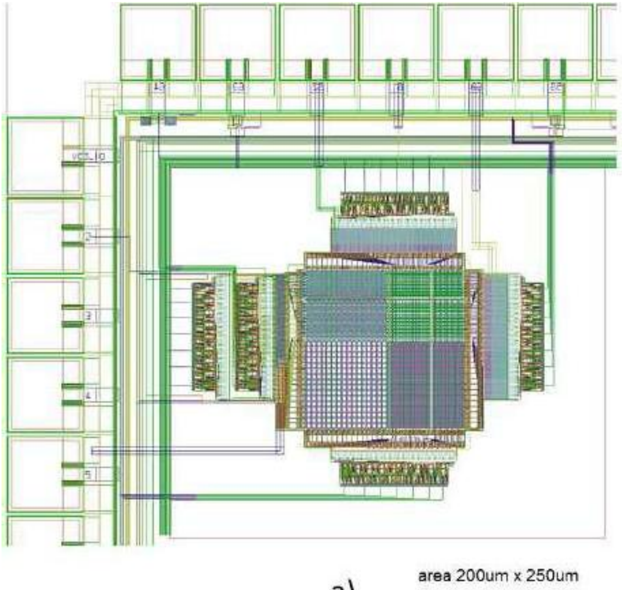
The OPEN device is measured and the S-parameters of the DUT are calculated as shown next:

Stotal -> Ytotal, Sopen -> Yopen Ydut -> Sdut

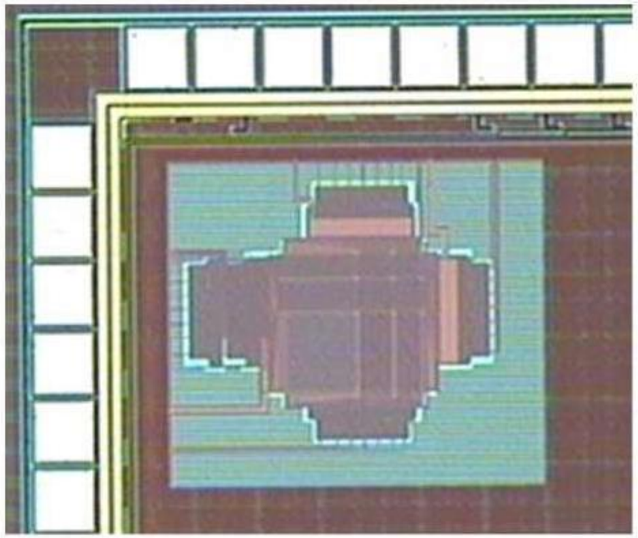
where:
Stotal -> measured S-parameters of the DUT including parasitics
Sopen -> measured S-parameters of the OPEN test structure
Sdut -> S-parameters of the DUT without influence of the parasitics
Yxxx -> transformed Y-parameters with: Yxxx = TwoPort(Sxxx,"S","Y")

Addressable MOSFET Matrix Test Structure





a)

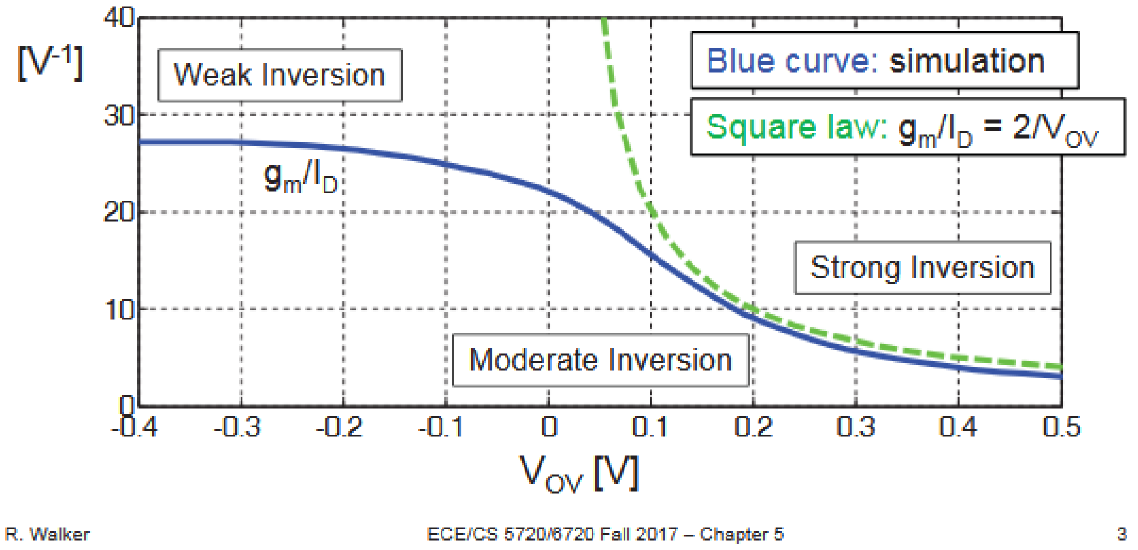


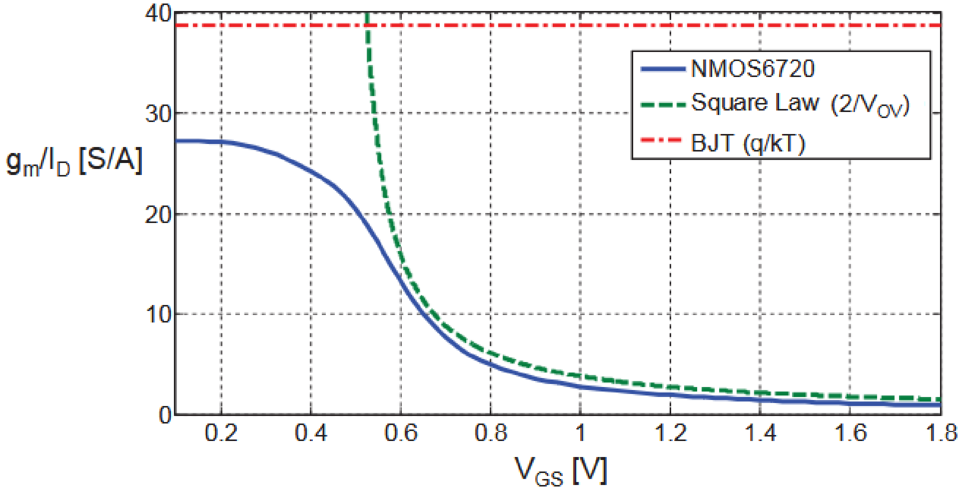
b)

Top view of the MOSFET Matrix IC, its GDSII layout and the microscopic view
IC designed for the study of geometrical aspects of the MOSFET mismatch

Questions?

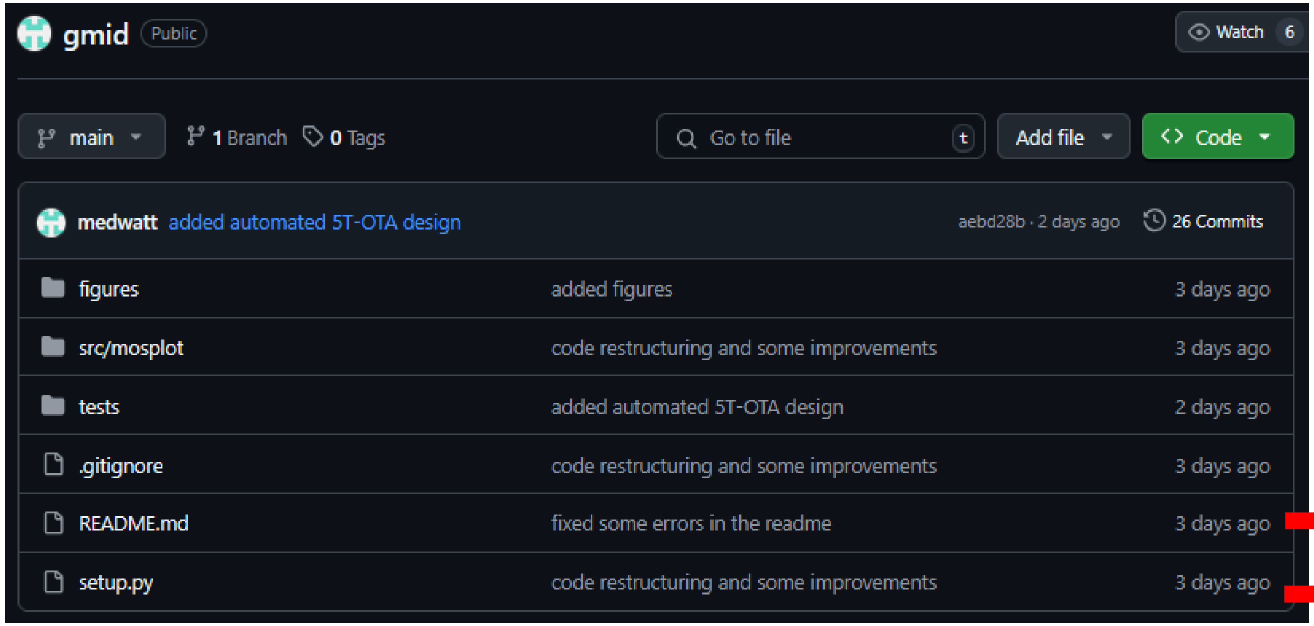
Small Insights In The Design Phase





Ross Walker ECE/CS 5720/6720 Fall 2017 University of Utah Partly adapted from Stanford's analog circuit design sequence

Small Insights In The Design Phase (Live Demo)



Original Repository: https://github.com/medwatt/gmid

Small Insights In The Design Phase (Live Demo)

- Setup Procedure:

Clone metwatts gmid repository	***https://github.com/medwatt/gmid***
Create virtual env	***python3 –m venv gmid_env***
Source the enviroment	***source gmid_env/bin/activate***
Inside the gmid repo run	***pip install .***

Small Insights In The Design Phase (Live Demo)

```python
from mosplot.lookup_table_generator.simulators import NgspiceSimulator, HspiceSimulator
from mosplot.lookup_table_generator import LookupTableGenerator, TransistorSweep
# One of `include_paths` or `lib_mappings` must be specified.
# The rest are optional.

ngspice = NgspiceSimulator(
    # Provide path to simulator if simulator is not in system path.
    simulator_path="ngspice",

    # Default simulation temperature. Override if needed.
    temperature=27,

    # All parameters are saved by default. Override if needed.
    parameters_to_save=["id", "vth", "vdsat", "gm","gds","vgs"],


    # Files to include with `.LIB`.
    lib_mappings = [
        ("/home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/ngspice/models/cornerMOSlv.lib", " mos_tt") # Put your own path to the corner lib
    ],

    # If the transistor is defined inside a subcircuit in
    # the library files, you must specify the symbol used (first entry)
    # and the hierarchical name (second entry). Override if needed.
    mos_spice_symbols = ("XM1", "n.xm1.nsg13_lv_nmos"),



    # Specify the width. For devices that do not take a width,
    # you can specify other parameters such as the number of fingers.
    # The keys are exactly those recognized by the model.
    device_parameters = {
        "w": 10e-6,
    }
)

nmos_sweep = TransistorSweep(
    mos_type="nmos",
    vgs=(0, 1.2, 0.01),
    vds=(0, 1.2, 0.01),
    vbs=(0, -1.2, -0.1),
    length = [130e-9, 260e-9, 390e-9, 520e-9, 650e-9, 780e-9, 910e-9, 1040e-9, 1170e-9, 1300e-9, 1430e-9, 1560e-9, 1690e-9, 1820e-9, 1950e-9, 2080e-9, 2210e-9, 2340e-9, 2470e-9, 2600e-9, 2730e-9,
2860e-9, 2990e-9, 3120e-9, 3250e-9, 3380e-9, 3510e-9, 3640e-9, 3770e-9, 3900e-9, 4030e-9, 4160e-9, 4290e-9, 4420e-9, 4550e-9, 4680e-9, 4810e-9, 4940e-9, 5070e-9, 5200e-9, 5330e-9, 5460e-9, 5590e-9,
5720e-9, 5850e-9, 5980e-9, 6110e-9, 6240e-9, 6370e-9, 6500e-9, 6630e-9, 6760e-9, 6890e-9, 7020e-9, 7150e-9, 7280e-9, 7410e-9, 7540e-9, 7670e-9, 7800e-9, 7930e-9, 8060e-9, 8190e-9, 8320e-9, 8450e-9,
8580e-9, 8710e-9, 8840e-9, 8970e-9, 9100e-9, 9230e-9, 9360e-9, 9490e-9, 9620e-9, 9750e-9, 9880e-9]
)
```

IHP-AnalogAcademy/modules/module_0_foundations/sg13_nmos_lv.py
IHP-AnalogAcademy/modules/module_0_foundations/sg13_pmos_lv.py

Small Insights In The Design Phase (Live Demo)

- Setup Procedure:

Create the LUTs from module_0_foundations/sg13_nmos_lv.py **_python3 sg13_nmos_lv.py_**

Now navigate to module_0_foundations/scripting and run **_pip install -r requirements.txt_**

From the same location launch the jupyter lab file **_jupyter lab gmid_test.ipynb_**

For the jupyter lab file you may only get a link you can paste into chrome/firefox

Small Insights In The Design Phase (Live Demo)

```
----- Sample Netlist -----
* Lookup Table Generation *
.lib '/home/pedersen/IHP-Open-PDK/ihp-sg13g2/libs.tech/ngspice/models/cornerMOSlv.lib'  mos_tt
VGS NG 0 DC=0
VBS NB 0 DC=0.0
VDS ND 0 DC=0
XM1 ND NG 0 NB sg13_lv_nmos  L=1.3e-07 w=1e-05
.options TEMP = 27
.options TNOM = 27
.control
save i(vds)
save @n.xm1.nsg13_lv_nmos[vth]
save @n.xm1.nsg13_lv_nmos[vdsat]
save @n.xm1.nsg13_lv_nmos[gm]
save @n.xm1.nsg13_lv_nmos[gds]
dc VDS 0 1.2 0.01 VGS 0 1.2 0.01
let i_vds = -i(vds)
write output all
.endc
.end
--------------------------
Total simulation jobs: 65
Progress: 1/65 jobs completed
Progress: 2/65 jobs completed
Progress: 3/65 jobs completed
Progress: 4/65 jobs completed
Progress: 5/65 jobs completed
Progress: 6/65 jobs completed
Progress: 7/65 jobs completed
Progress: 8/65 jobs completed
Progress: 9/65 jobs completed
Progress: 10/65 jobs completed
Progress: 11/65 jobs completed
Progress: 12/65 jobs completed
Progress: 13/65 jobs completed
Progress: 14/65 jobs completed
Progress: 15/65 jobs completed
Progress: 16/65 jobs completed
Progress: 17/65 jobs completed
Progress: 18/65 jobs completed
Progress: 19/65 jobs completed
Progress: 20/65 jobs completed
```

Saves the data in LUTs folder in the .npz format

Small Insights In The Design Phase (Live Demo)

[101]:
```python
import matplotlib.pyplot as plt
import numpy as np
from mosplot.plot import load_lookup_table, Mosfet, Expression
import ipywidgets as widgets
from ipywidgets import interactive
from ipywidgets import interactive_output, HBox, VBox
import matplotlib.ticker as ticker
```

[102]:
```python
lookup_table_nmos = load_lookup_table('../sg13_nmos_lv_LUT.npz')
lookup_table_pmos = load_lookup_table('../sg13_pmos_lv_LUT.npz')
```

[103]:
```python
print(lookup_table.keys())
```

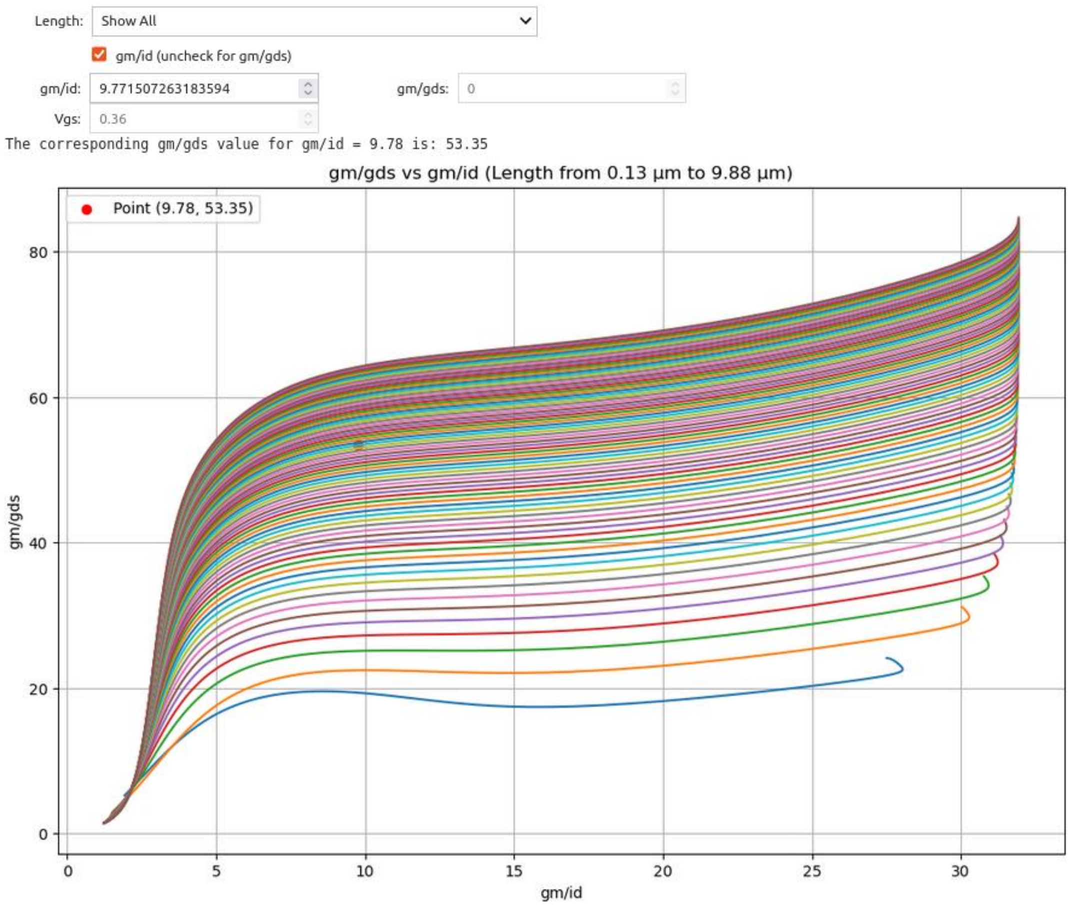
```
dict_keys(['sg13_lv_nmos ', 'sg13_lv_pmos', 'description', 'simulator', 'parameter_names', 'device_parameters'])
```

[108]:
```python
nmos = Mosfet(lookup_table=lookup_table_nmos, mos="sg13_lv_nmos ", vbs=0.0, vds=0.6)
pmos = Mosfet(lookup_table=lookup_table_pmos, mos="sg13_lv_pmos", vbs=0.0, vds=-0.6, vgs=(-1.2, -0.15))

rows_0, cols_0 = np.shape(nmos.extracted_table['gm']) # just for getting the shape of the data
rows_1, cols_1 = np.shape(pmos.extracted_table['gm']) # just for getting the shape of the data
reshaped_lengths_nmos = np.tile(nmos.length[:, np.newaxis], (1, cols_0))
reshaped_lengths_pmos = np.tile(pmos.length[:, np.newaxis], (1, cols_1))
```

```python
def plot_data_vs_data(x_values, y_values, z_values, length, x_axis_name, y_axis_name='y', y_multiplier=1, log=False):•••
```

NMOS GMID

[110]:
```python
width_values = nmos.width
id_values = nmos.extracted_table['id']
gm_values = nmos.extracted_table['gm']
gds_values = nmos.extracted_table['gds']
vgs_values= nmos.extracted_table['vgs']

plot_data_vs_data(gm_values/id_values, gm_values/gds_values, vgs_values, reshaped_lengths_nmos, 'gm/id', 'gm/gds')
```



Part 1

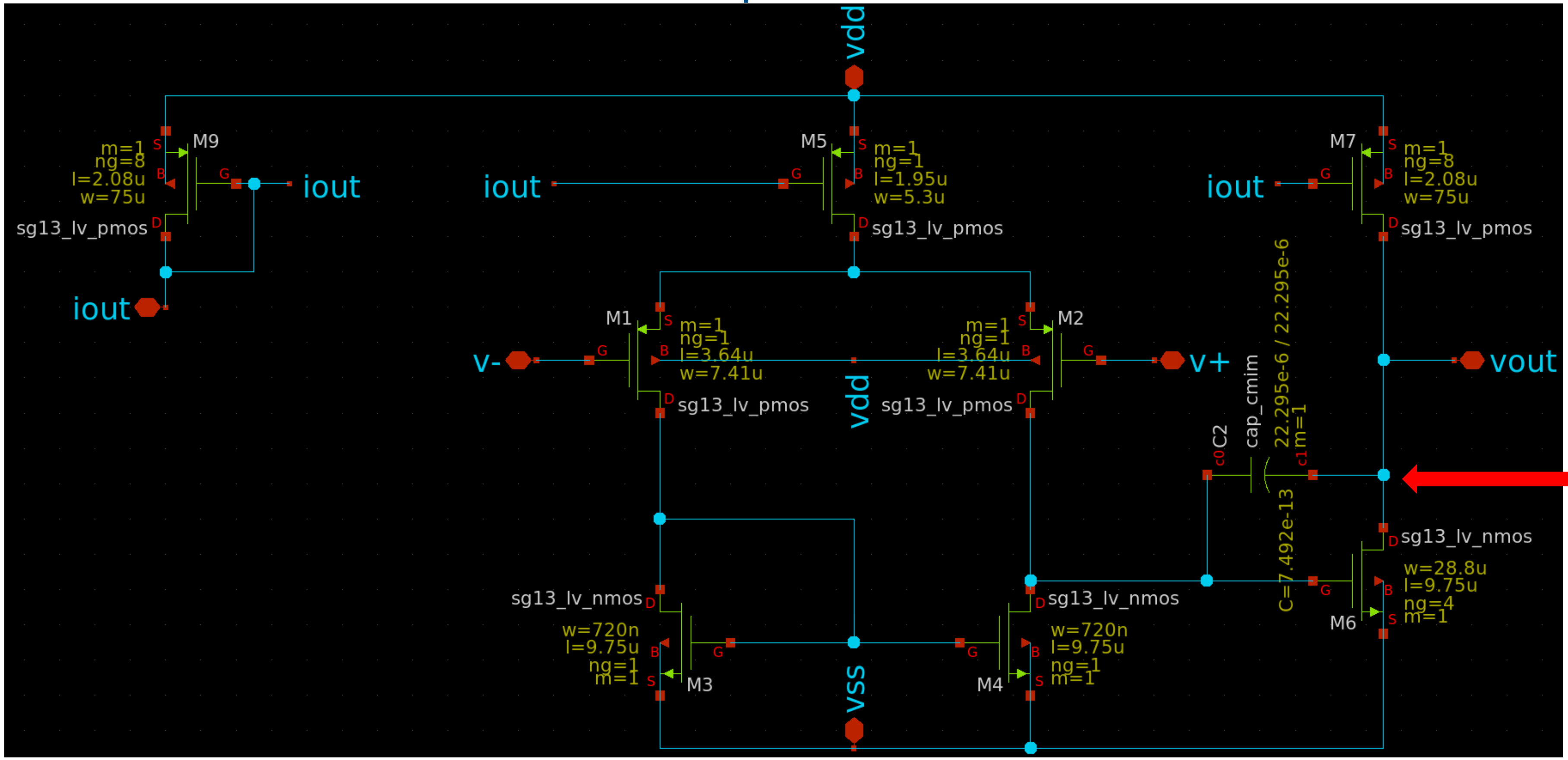
Miller Compensated OTA-Design

Reference: /modules/module_1_bandgap_reference/part_1_OTA

Leibniz Institute for high performance microelectronics

Schematic & Symbol

- Create a folder for this module and organize yourself within
- Create a Schematic for the OTA seen in the picture



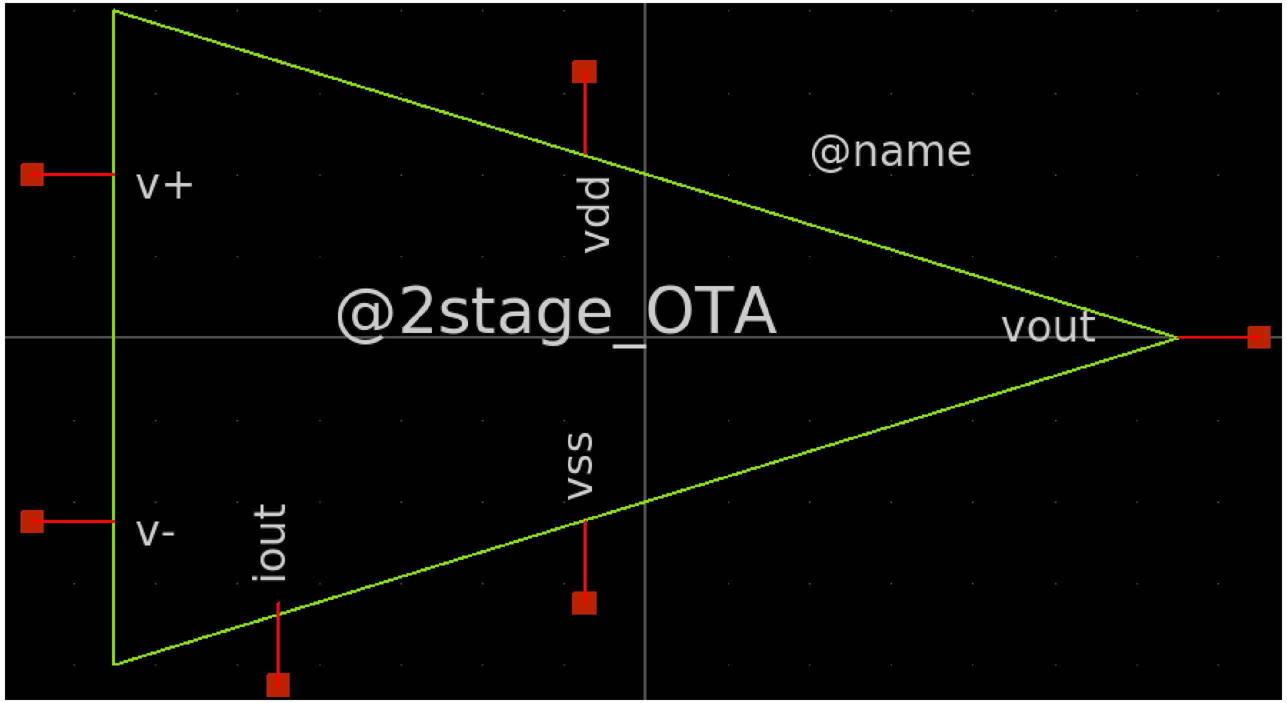
CMIM

Specs cmim:

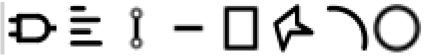
W = L = 22.295e-6

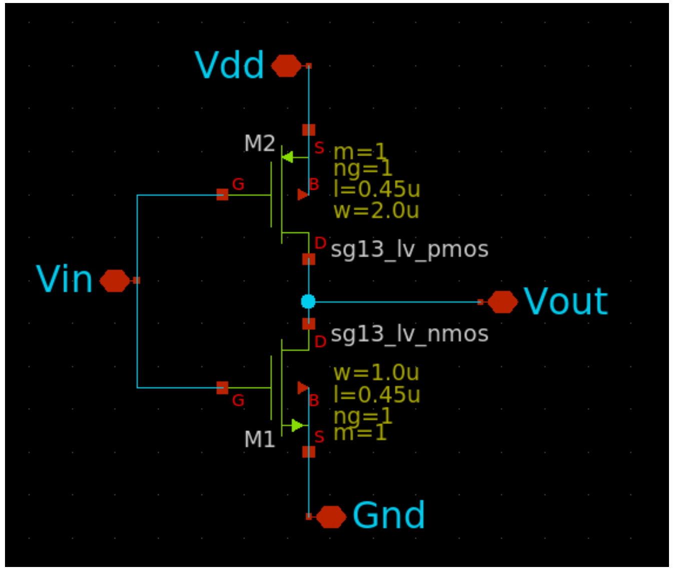
Schematic & Symbol

- Create the OTA symbol in your preferred style
- The next slide provides a reminder on how to create a symbol

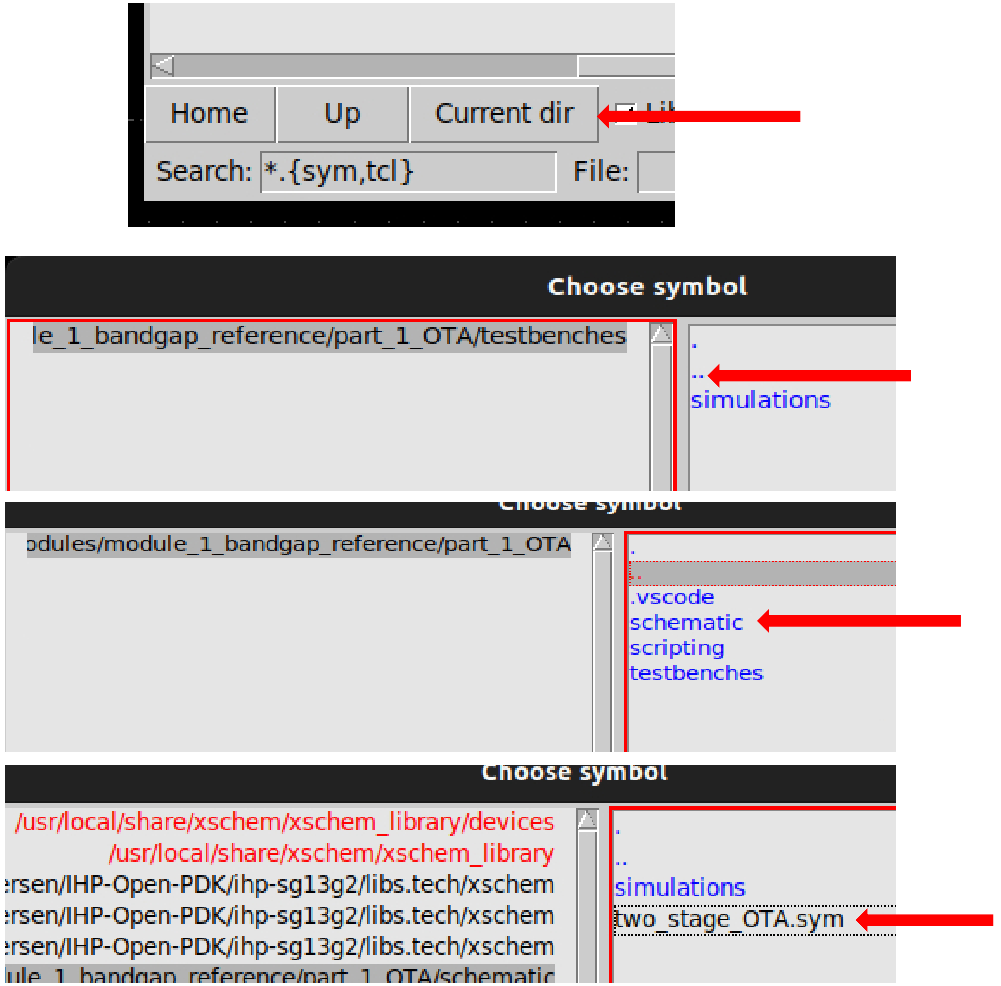


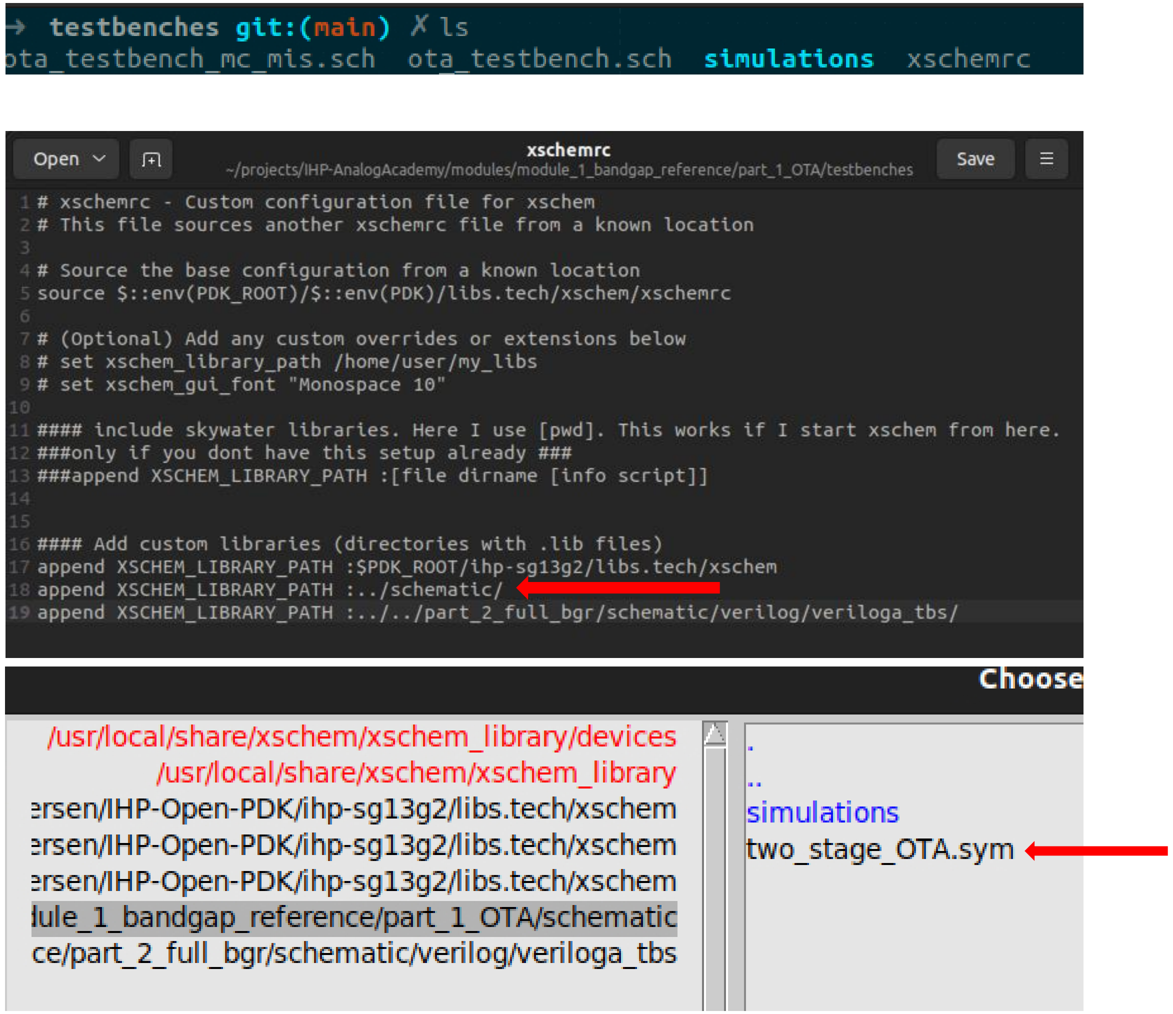
Creating a Symbol (from yesterday)

- For Creating the Symbol press **A**
- This creates a symbol file in the directory of the schematic, navigate to this file and open it by writing (**xschem inverter.sym**)
- Now you can drag the already existing lines to create the boundaries or use the tools seen in the toolbar to define custom shapes 

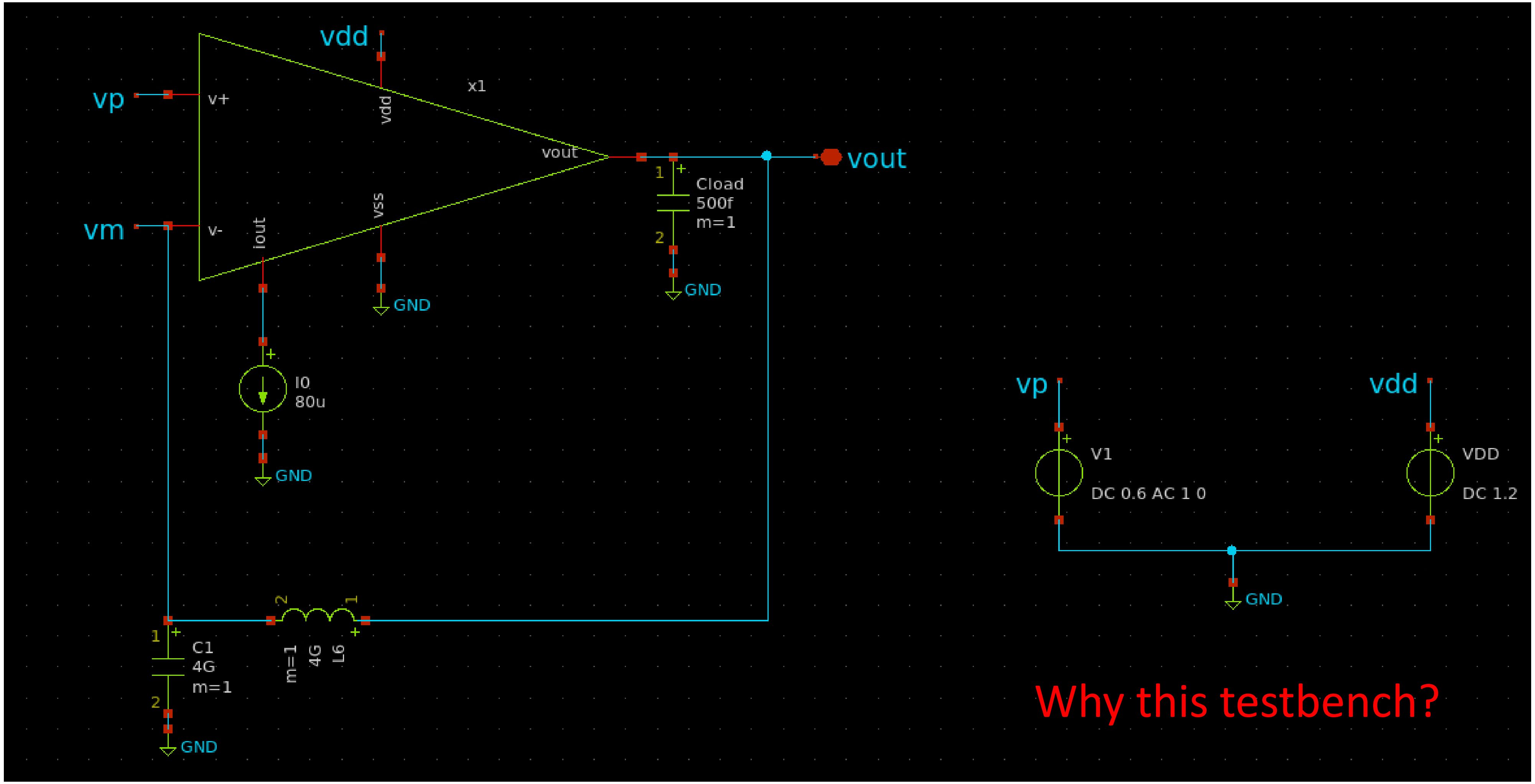


Accessing the Symbol (Two Options)





OTA Testbench



OTA Testbench: Finish The Testbench

Model Definition

Path: /usr/local/share/xschem/xschem_library/devices

Symbol devices/code_shown.sym OK Cancel

☐ No change properties ☐ Preserve unchanged props ☐ Copy cell

```
name=MODEL only_toplevel=true
format="tcleval( @value )"
value="
.lib $::SG13G2_MODELS/cornerCAP.lib cap_typ
.lib cornerMOSlv.lib mos_tt
"
```

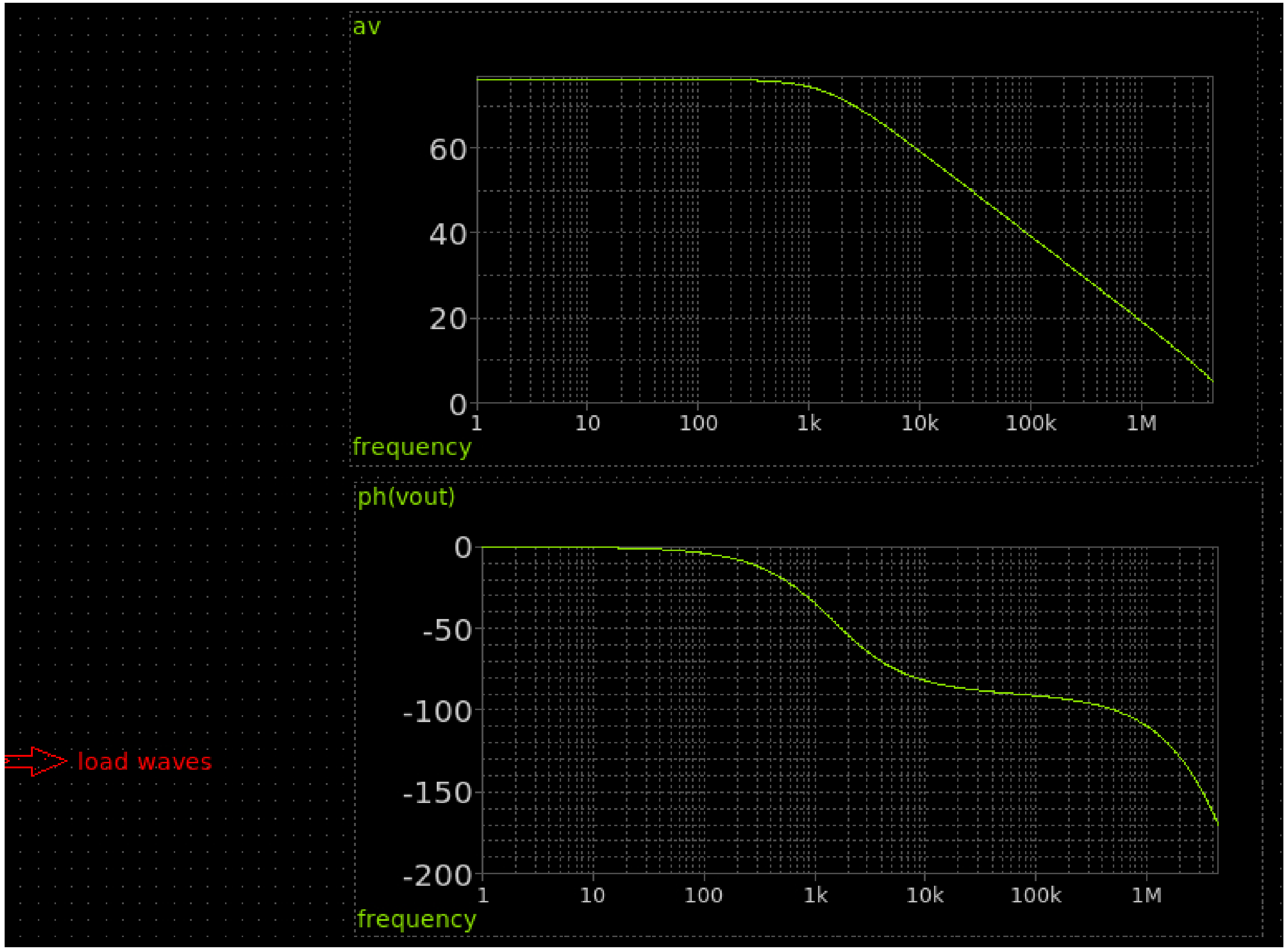
Simulation Code

DC Sim

```
name=NGSPICE only_toplevel=false
value="
.control
op
save all
write tb_OTA_op.raw
.endc
```

AC Sim

```
.control
op
ac dec 100 1 10e6
save all
let Av = db(v(vout))
let phase    = 180*cph(vout)/pi
write output_file.raw
.endc
"
```

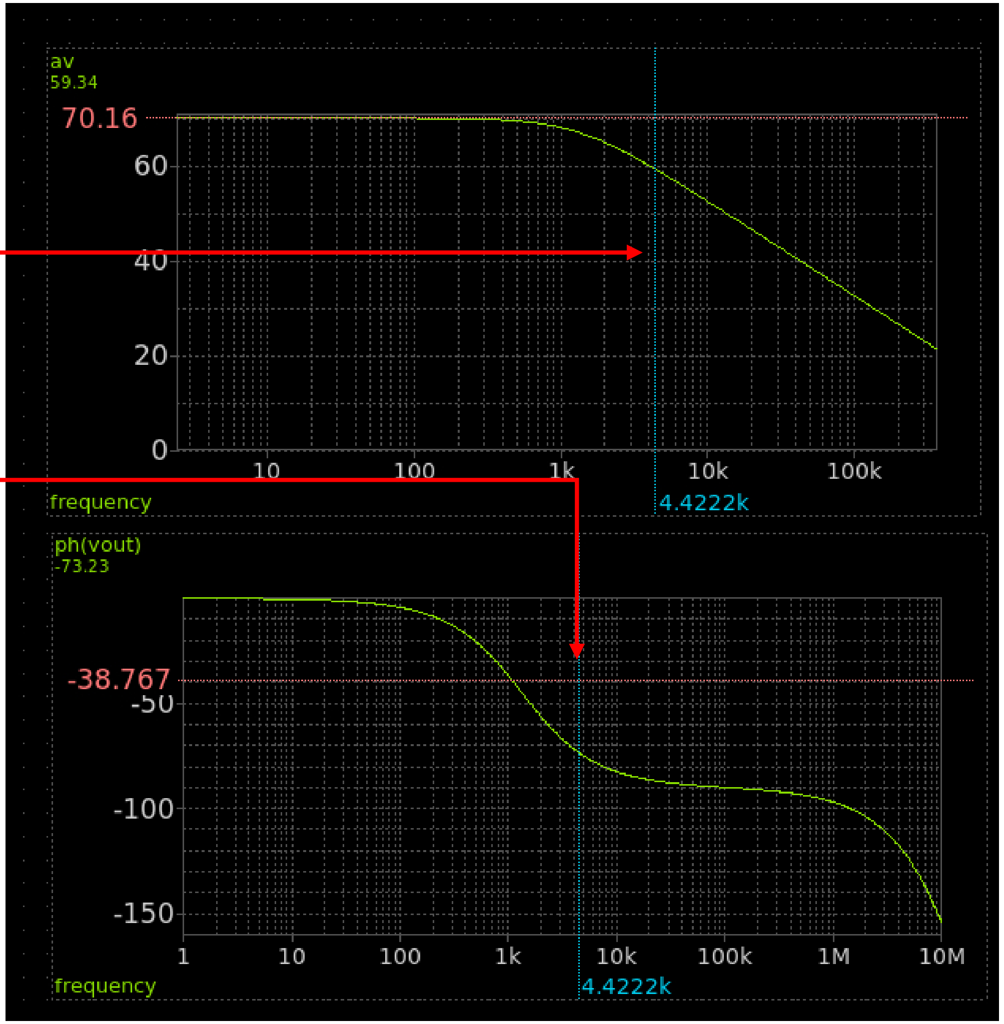


OTA Testbench: Finish The Testbench

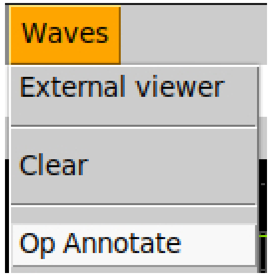
Shortcuts for measuring in plot:

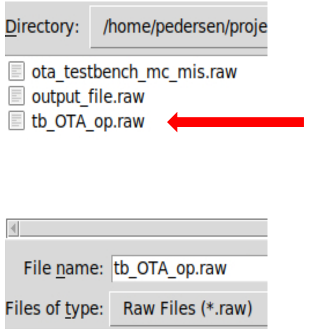
Vertical line: A

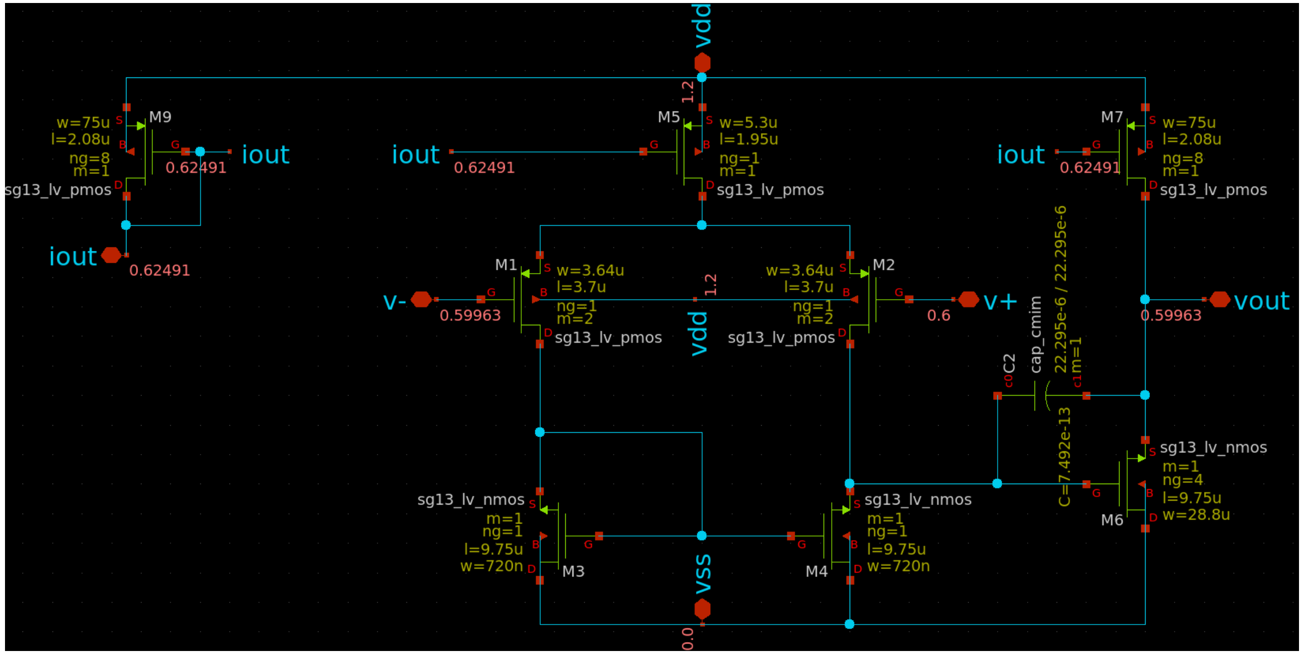
Horizontal Line: Shift + A

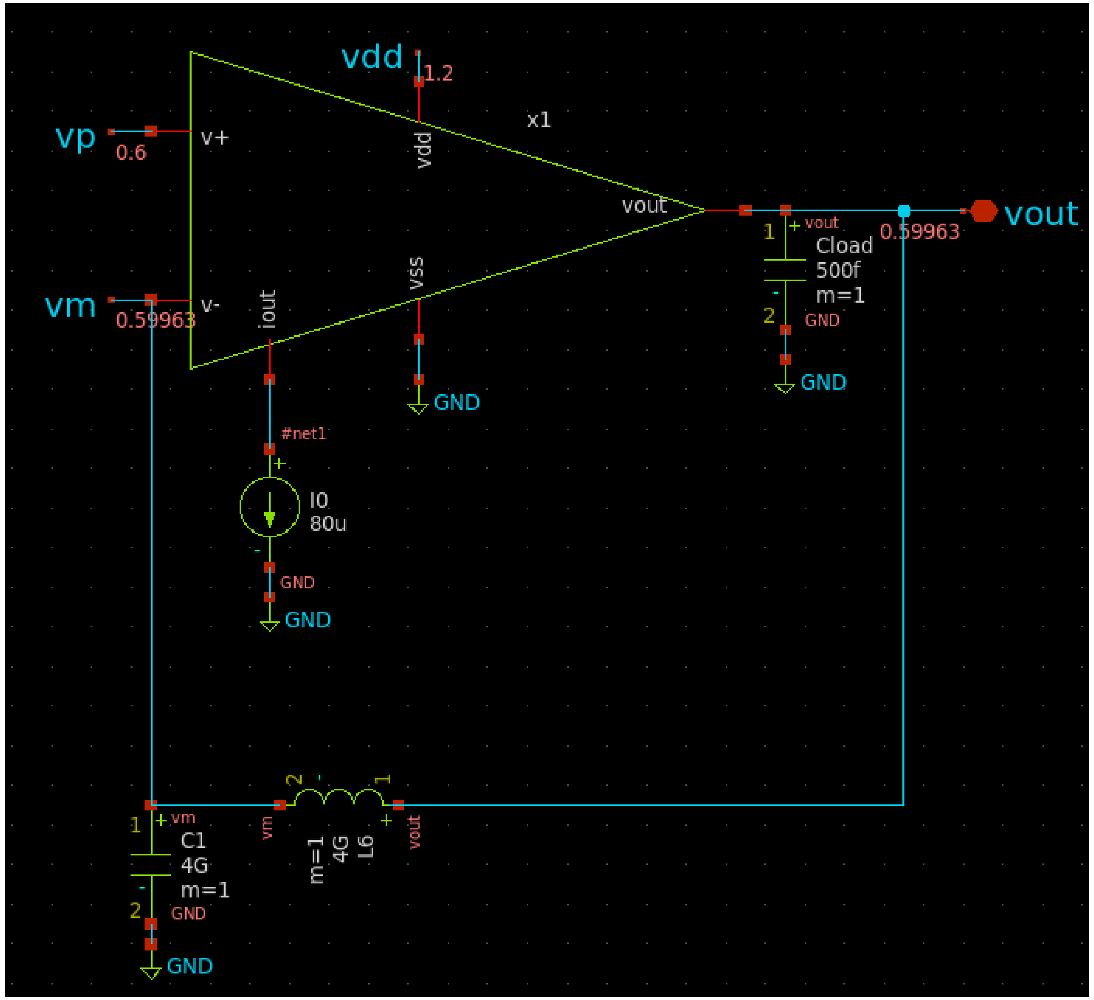


OTA Testbench: Op Annotation



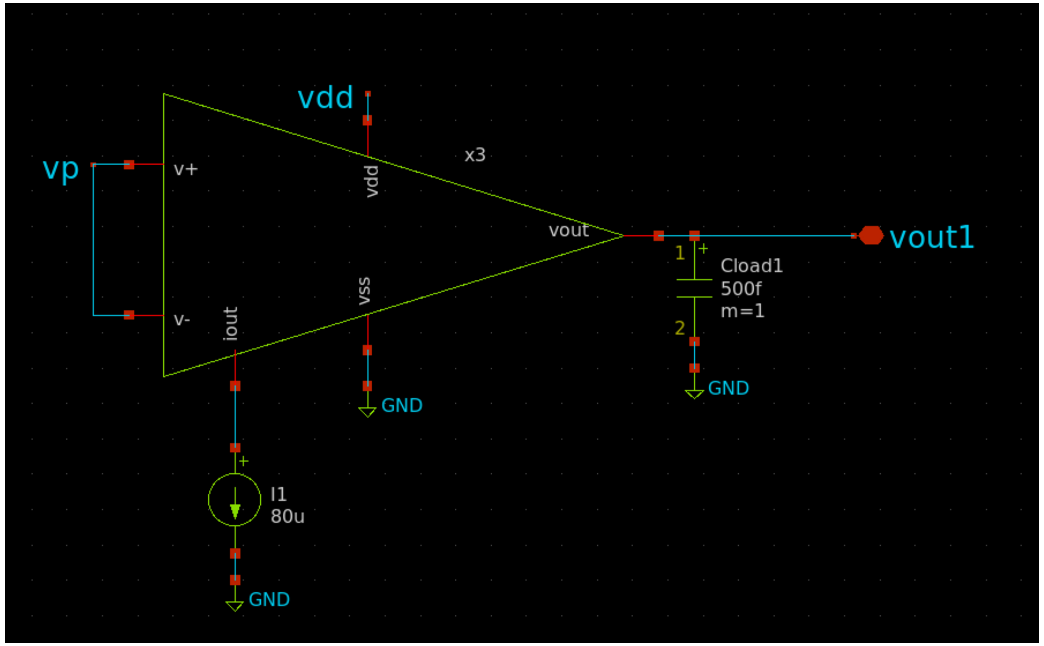


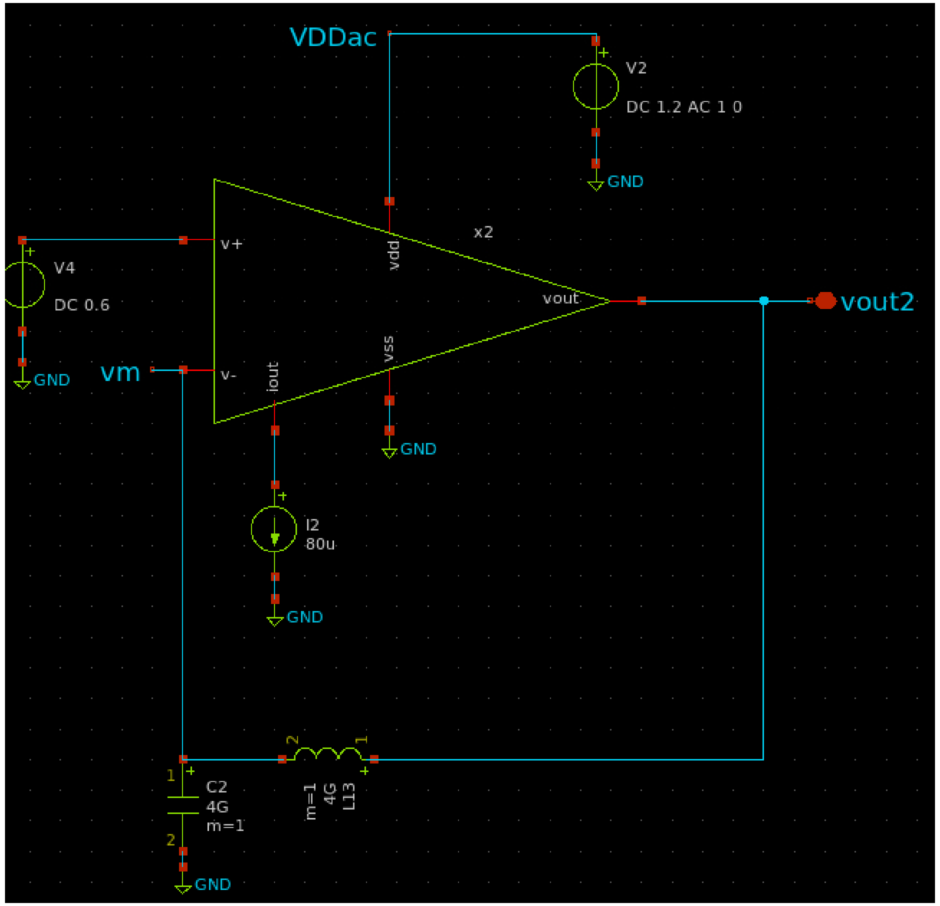




OTA Testbench: CMRR & PSRR

HINTS



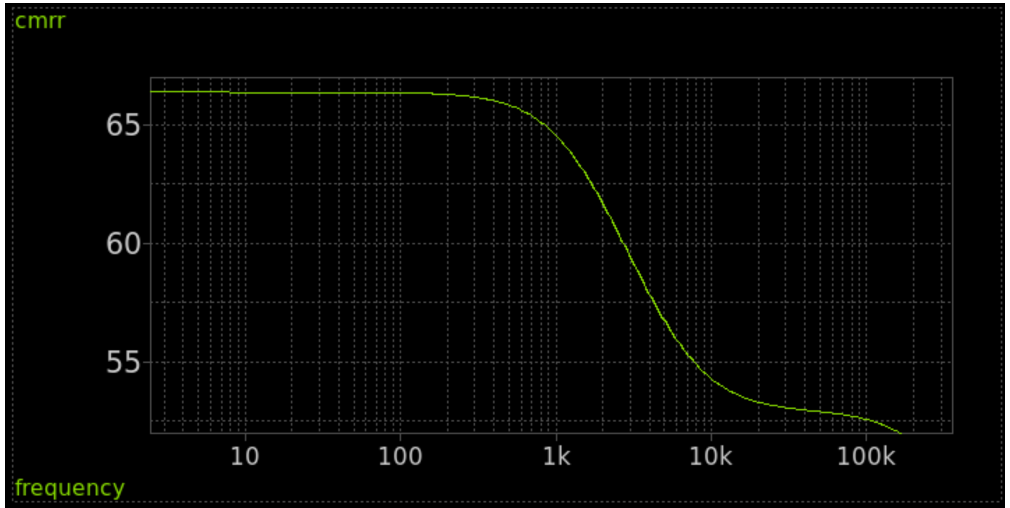


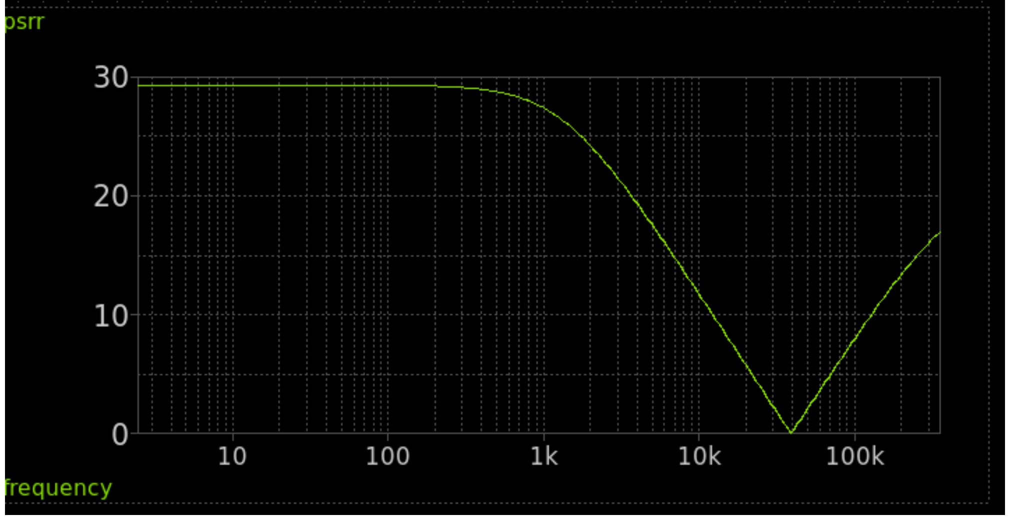
$$CMRR_{dB} = 20Log_{10}\left(\frac{A_{diff}}{A_{cm}}\right)$$

$$PSRR_{dB} = 20Log_{10}\left(\frac{\Delta V_{out}}{\Delta V_{DD(AC)}}\right)$$

OTA Testbench: CMRR & PSRR

```
.control
op
ac dec 100 1 10e6
save all

let Av = db(v(vout))

let PSRR = db(v(vout2)/v(VDDac))

let CMRR = db((v(vout)/v(vp))/(v(vout1)/v(vp)))

let phase = 180*cph(vout)/pi

write output_file.raw
.endc
```





Catching Up / Lunch Time !

Next session:
Bandgap Reference Design

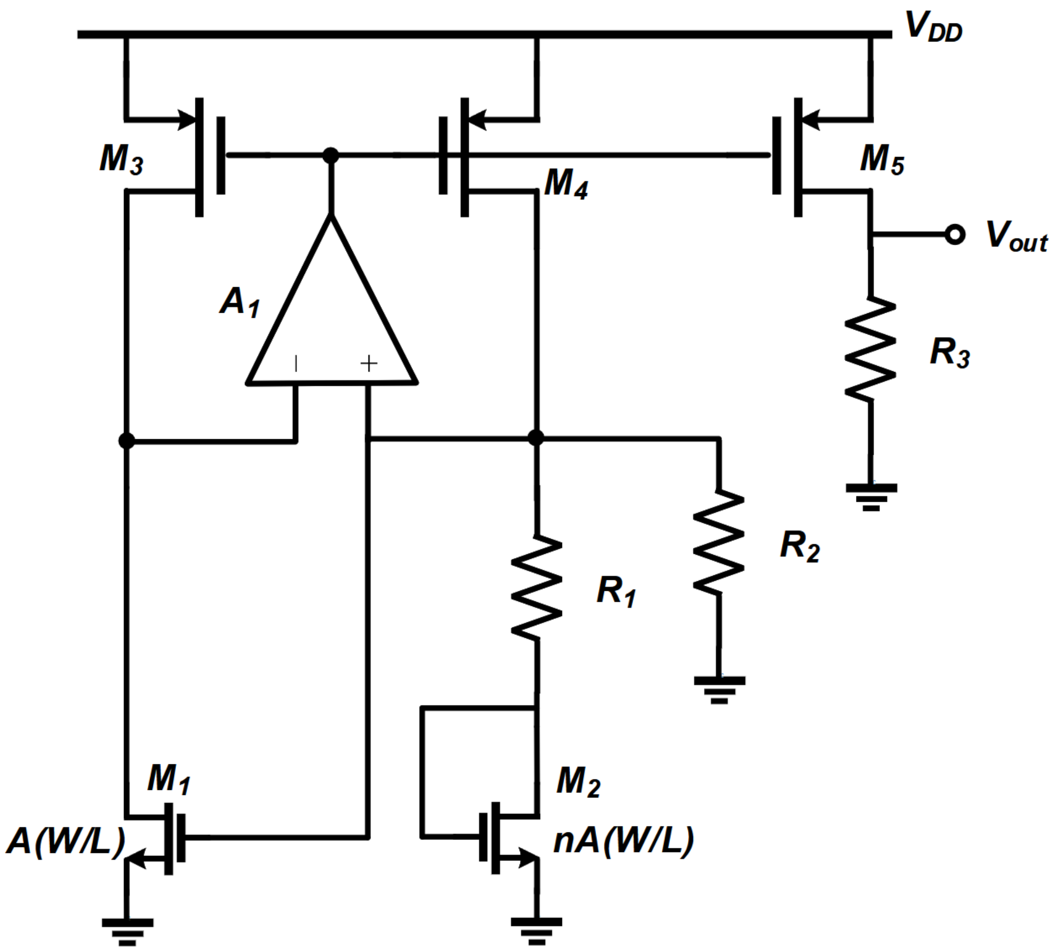
Leibniz Institute
for high
performance
microelectronics

Leibniz Institute
for high
performance
microelectronics

Part 2

Bandgap Reference Design
Reference: /modules/module_1_bandgap_reference/part_2_full_bgr

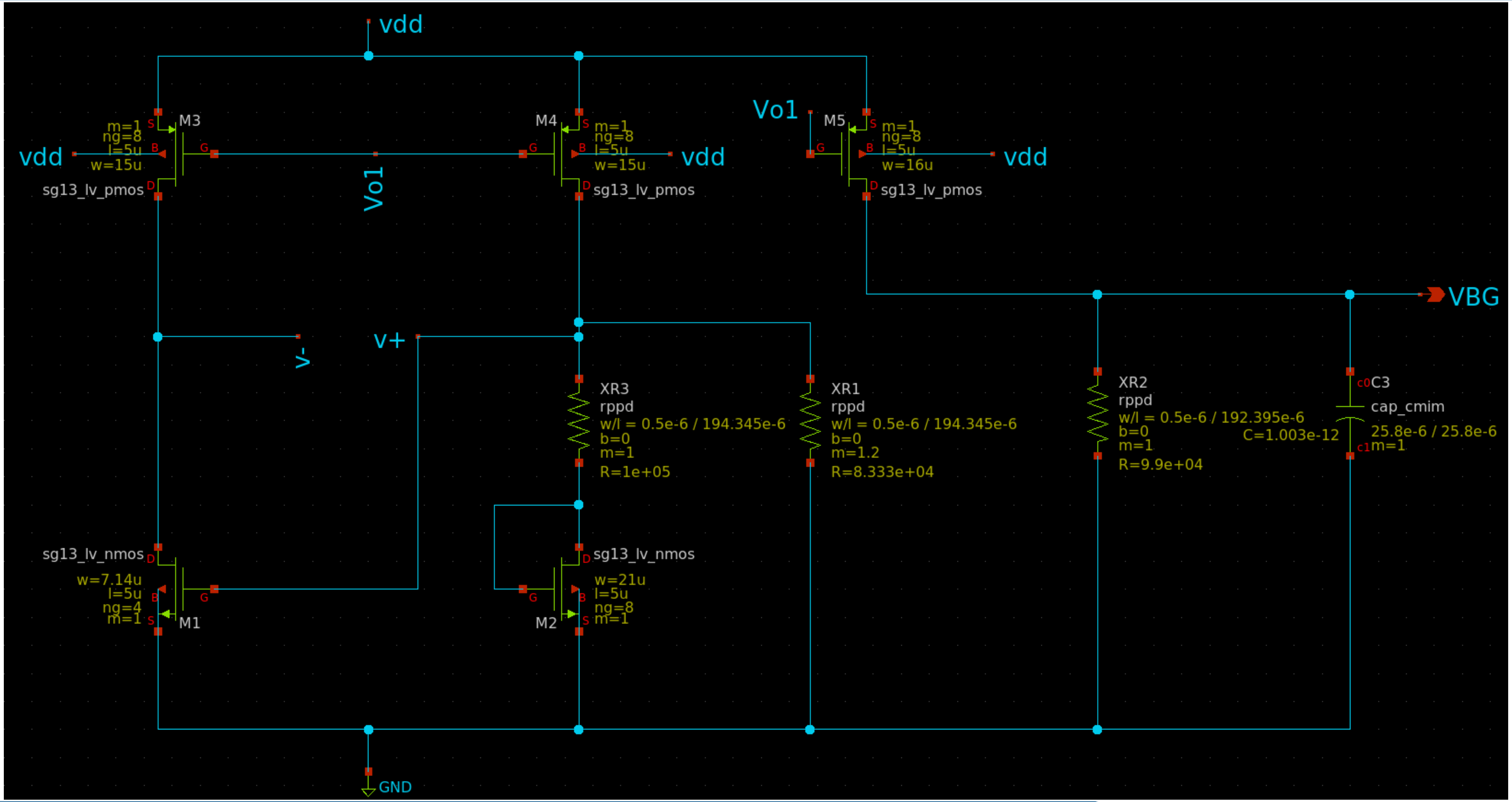
Bandgap Reference



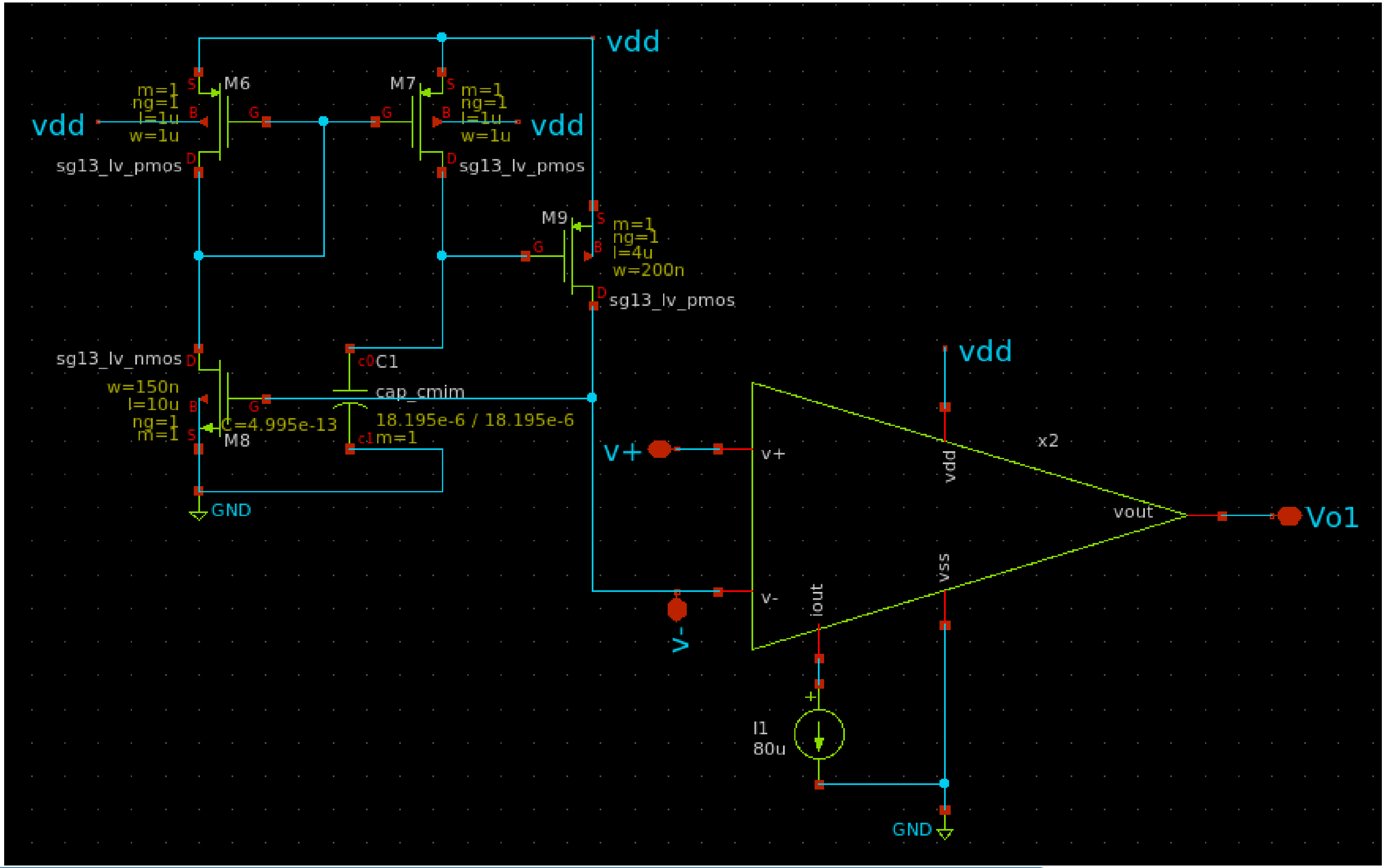
Bandgap Reference

Resistor Type	Material	Sheet Resistance	Temperature Coefficient (TC)	Applications
Rsil	Salicided n-doped polysilicon	7 Ω/□	3100 ppm/K	Low-resistance paths, non-precision use
Rppd	Unsalicided p-doped polysilicon	260 Ω/□	170 ppm/K	Precision and temperature-sensitive designs
Rhigh	High-resistance structures	Very high	Variable	High-impedance paths

Bandgap Reference Core Schematic



Start-up Circuit Schematic



Import Of Models

```
name=MODEL only_toplevel=true
format="tcleval( @value )" value="
.lib $::SG13G2_MODELS/cornerCAP.lib cap_typ
.lib $::SG13G2_MODELS/cornerRES.lib res_typ
.lib cornerMOSlv.lib mos_tt "
```

Simulation

DC Simulation

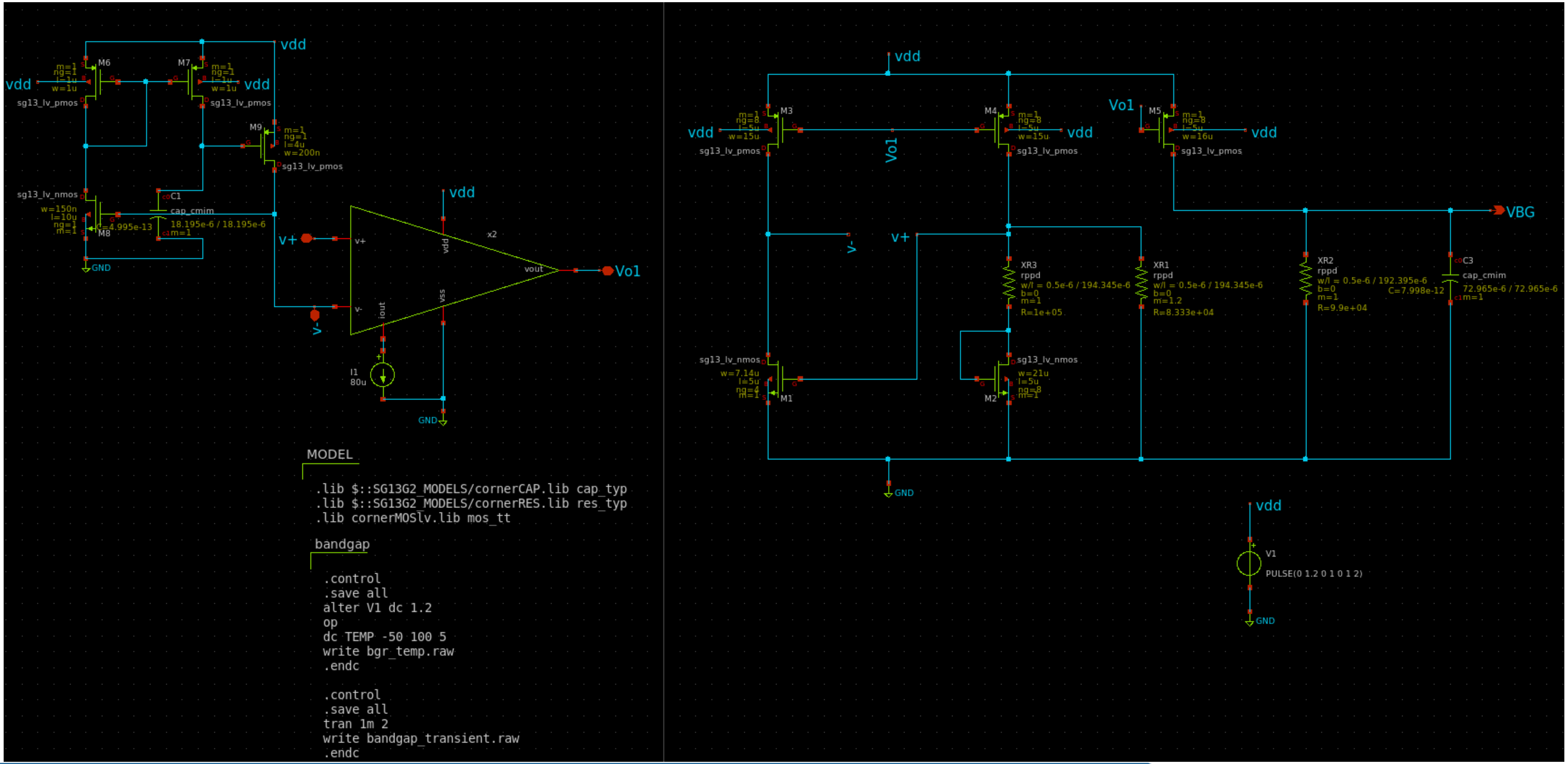
```
.control
.save all
alter V1 dc 1.2
op
dc TEMP -50 100 5
write bgr_temp.raw .endc
```

Transient Simulation

```
.control .save all
tran 1m 2
write bandgap_transient.raw .endc
```

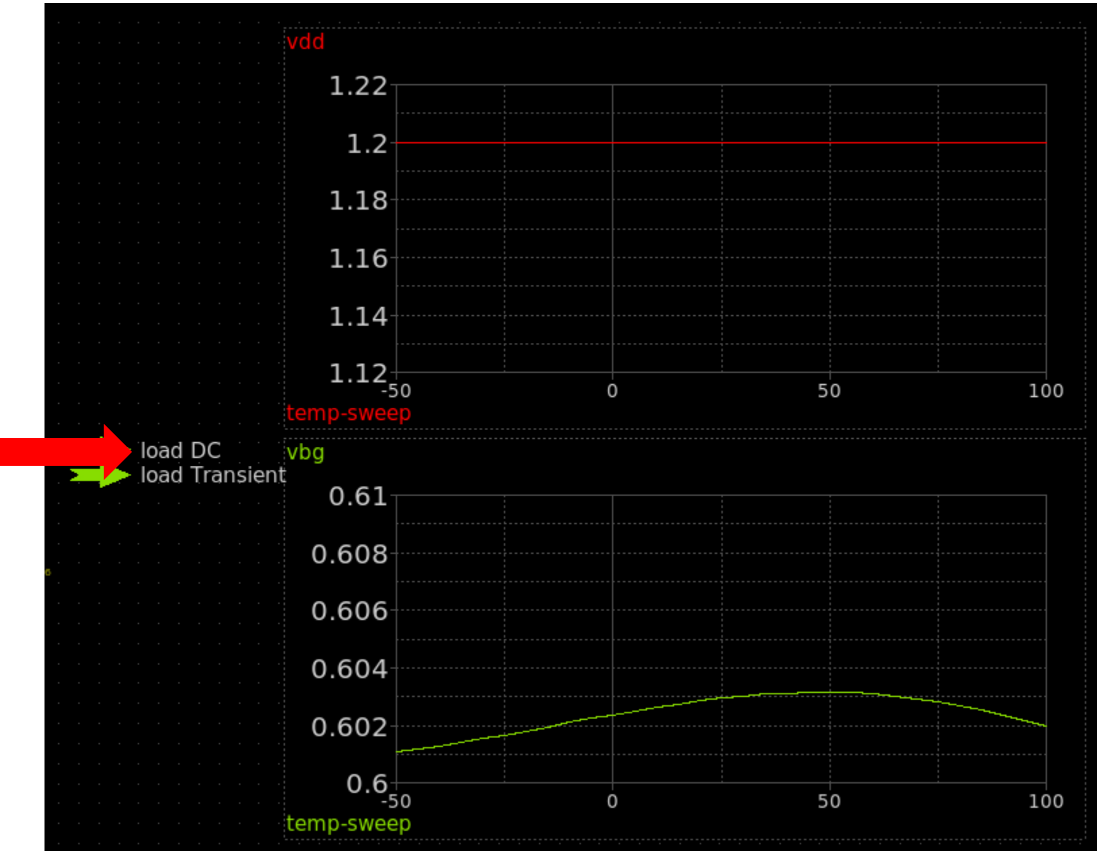
VDD → "PULSE(0 1.2 0 1 0 1 2)"

Simulation

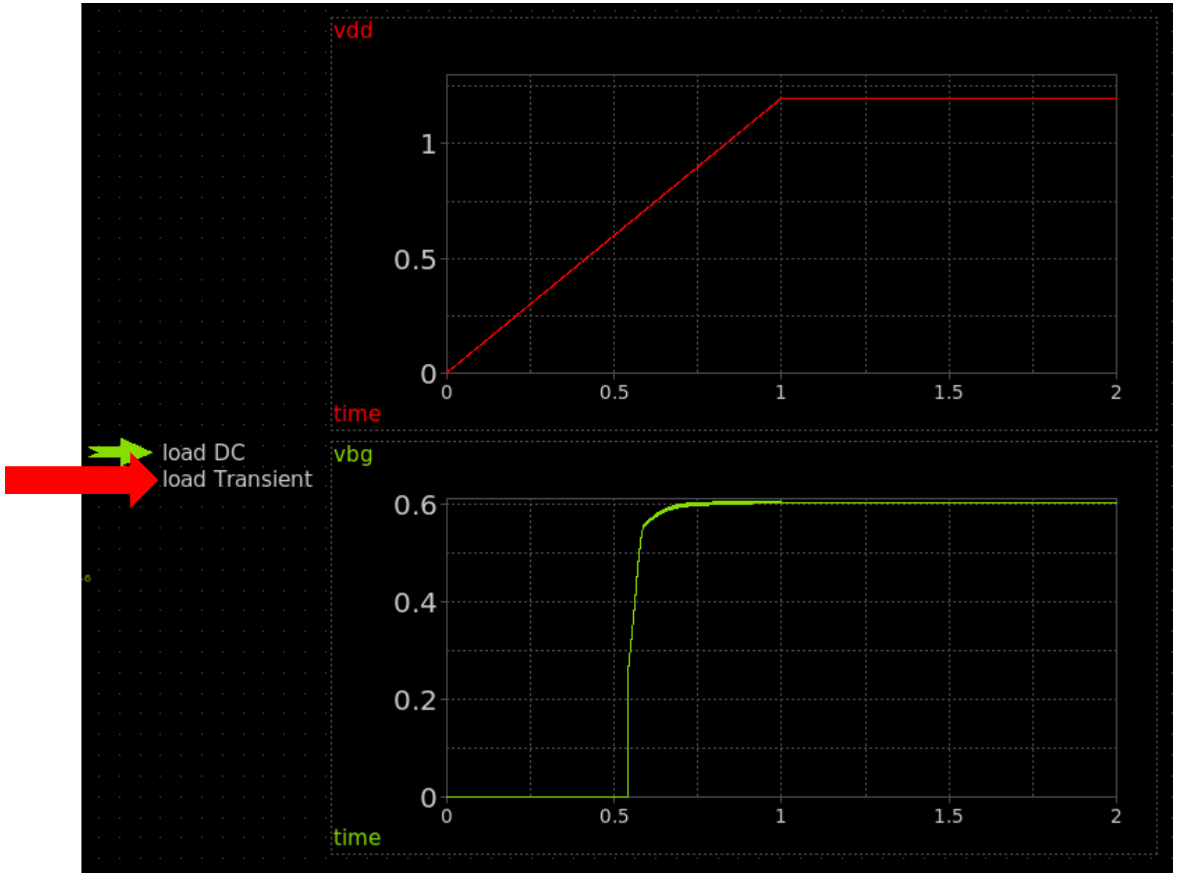

vdd
M6
M7
M9
sg13_lv_pmos
sg13_lv_nmos
C1
cap_cmim
M8
v+
v-
vout
x2
vdd
vss
iout
Vo1
I1
80u
GND
MODEL
.lib $::SG13G2_MODELS/cornerCAP.lib cap_typ
.lib $::SG13G2_MODELS/cornerRES.lib res_typ
.lib cornerMOSlv.lib mos_tt
bandgap
.control
.save all
alter V1 dc 1.2
op
dc TEMP -50 100 5
write bgr_temp.raw
.endc
.control
.save all
tran 1m 2
write bandgap_transient.raw
.endc
M3
M4
M5
Vo1
VBG
XR3
rppd
w/l = 0.5e-6 / 194.345e-6
R=1e+05
XR1
rppd
w/l = 0.5e-6 / 194.345e-6
R=8.333e+04
XR2
rppd
w/l = 0.5e-6 / 192.395e-6
R=9.9e+04
C3
cap_cmim
C=7.998e-12
72.965e-6 / 72.965e-6
M1
M2
V1
PULSE(0 1.2 0 1 0 1 2)

plots

bgr_temp.raw



bandgap_transient.raw



Mismatch and Monte Carlo Simulations in Analog Design

🔍 **Mismatch in IC Manufacturing**

- Caused by **process variations**: geometry, doping, threshold voltage etc.
- Critical in **symmetry-based circuits** (e.g., differential pairs, bandgap references).
- Leads to **imbalanced currents**, voltage offsets, and **reduced accuracy**.

Monte Carlo Simulations: Why They Matter

- Simulate **random device-level variations**.
- Evaluate **circuit robustness** under realistic conditions.
- **1σ (~68%)**, **2σ (~95%)**, **3σ (~99.7%)** coverage of variation.

Mismatch Models and Code

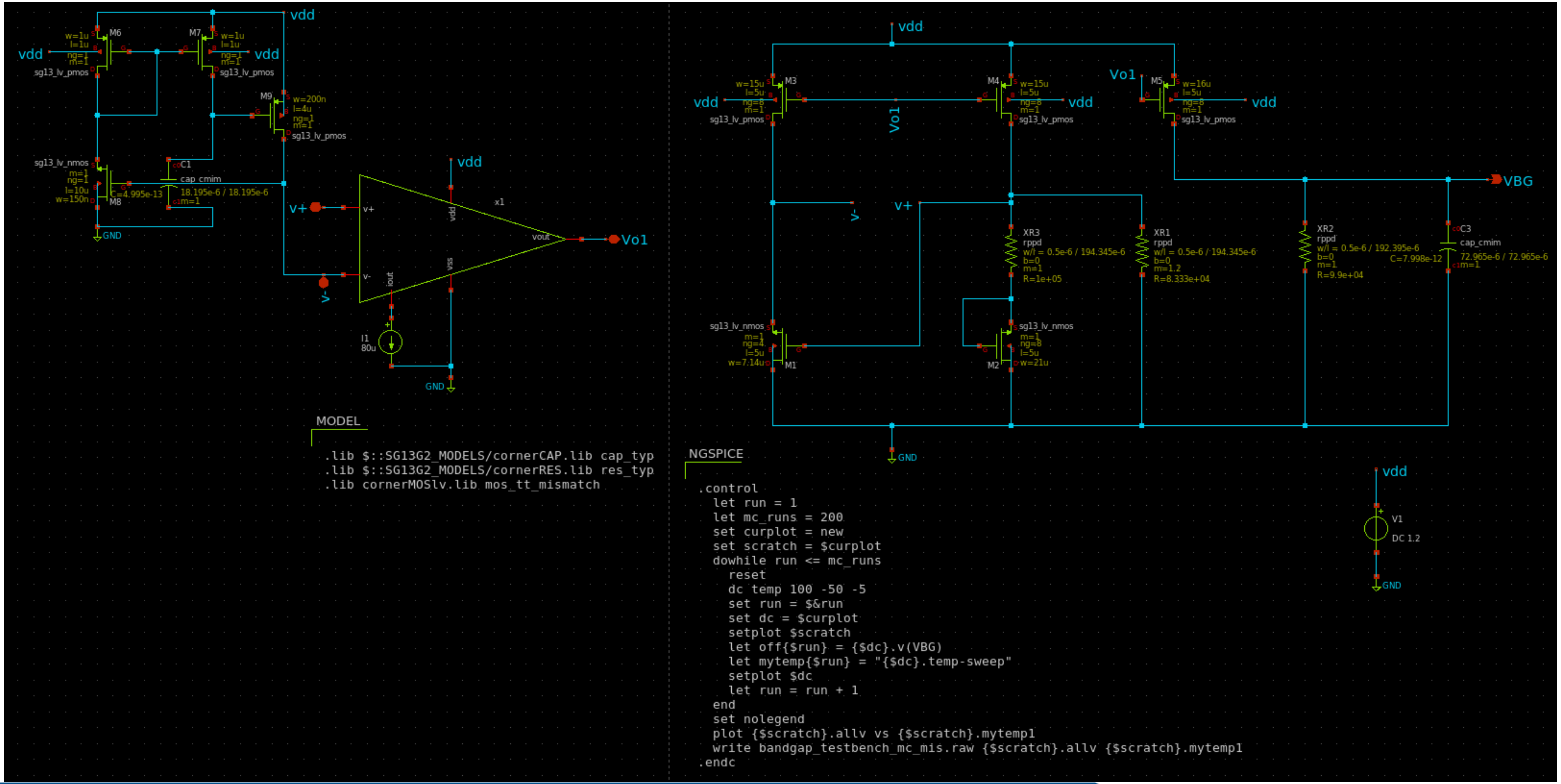
```
name=NGSPICE only_toplevel=false
value="
.control
  let run = 1                                   Initializes the run counter
  let mc_runs = 200                             Sets the number of Monte Carlo runs
  set curplot = new                             Creates a new plot and assigns to scratch
  set scratch = $curplot
  dowhile run <= mc_runs                        Starts a loop that runs until run > 200
     reset                                      Resets the data to a clean state for each run
     dc temp 100 -50 -5                         Performs a DC sweep over temperature range
     set run = $&run                            Using the run value as a variable
     set dc = $curplot                          Saves the current plot name to variable dc
     setplot $scratch                           Switches to the scratch plot to store results
     let off{$run} = {$dc}.v(VBG)               Extracts the VBG voltage and stores it under the current run
     let mytemp{$run} = \"{$dc}.temp-sweep\“    Saves the temp data for the current run
     setplot $dc                                Returns to the dc plot
     let run = run + 1                          increments the run counter
  end
  set nolegend                                  Disables legend
  plot {$scratch}.allv vs {$scratch}.mytemp1    Plots all the waves against the temperature

  write bandgap_testbench_mc_mis.raw {$scratch}.allv {$scratch}.mytemp1
.endc
```

Mismatch Models and Code

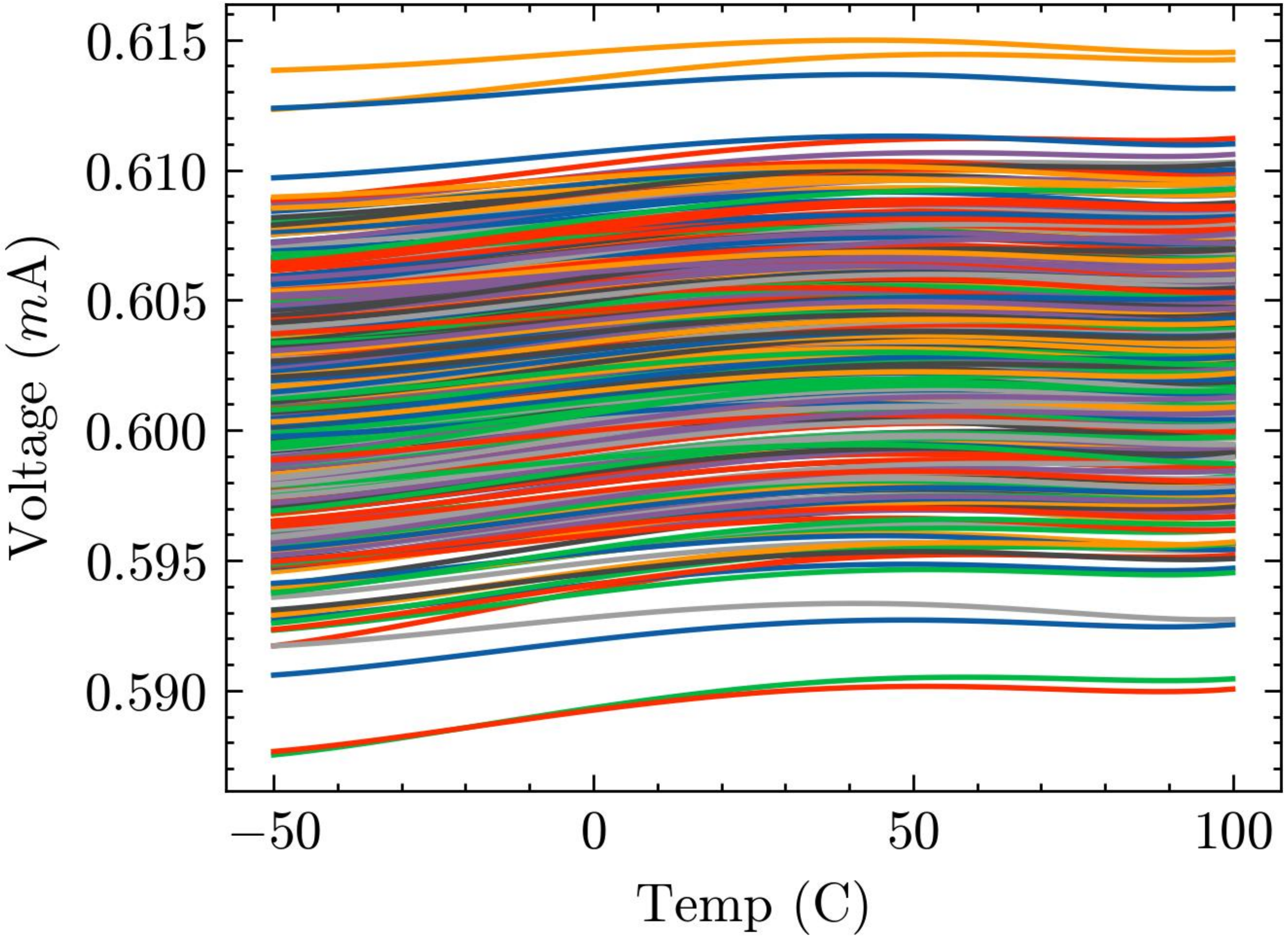
```
.lib $::SG13G2_MODELS/cornerCAP
.lib cap_typ .lib $::SG13G2_MODELS/cornerRES
.lib res_typ .lib cornerMOSlv
.lib mos_tt_mismatch
```

Use .lib mos_tt_stat for statistical variation i.e wafer to wafer

Testbench


vdd
Vo1
V+
V-
VBG
GND
MODEL
.lib $::SG13G2_MODELS/cornerCAP.lib cap_typ
.lib $::SG13G2_MODELS/cornerRES.lib res_typ
.lib cornerMOSlv.lib mos_tt_mismatch
NGSPICE
.control
let run = 1
let mc_runs = 200
set curplot = new
set scratch = $curplot
dowhile run <= mc_runs
reset
dc temp 100 -50 -5
set run = $&run
set dc = $curplot
setplot $scratch
let off{$run} = {$dc}.v(VBG)
let mytemp{$run} = "{$dc}.temp-sweep"
setplot $dc
let run = run + 1
end
set nolegend
plot {$scratch}.allv vs {$scratch}.mytemp1
write bandgap_testbench_mc_mis.raw {$scratch}.allv {$scratch}.mytemp1
.endc

Results



Coffee Break!! / Catching Up

Leibniz Institute for high performance microelectronics

Leibniz Institute for high performance microelectronics

Part 3

Layout

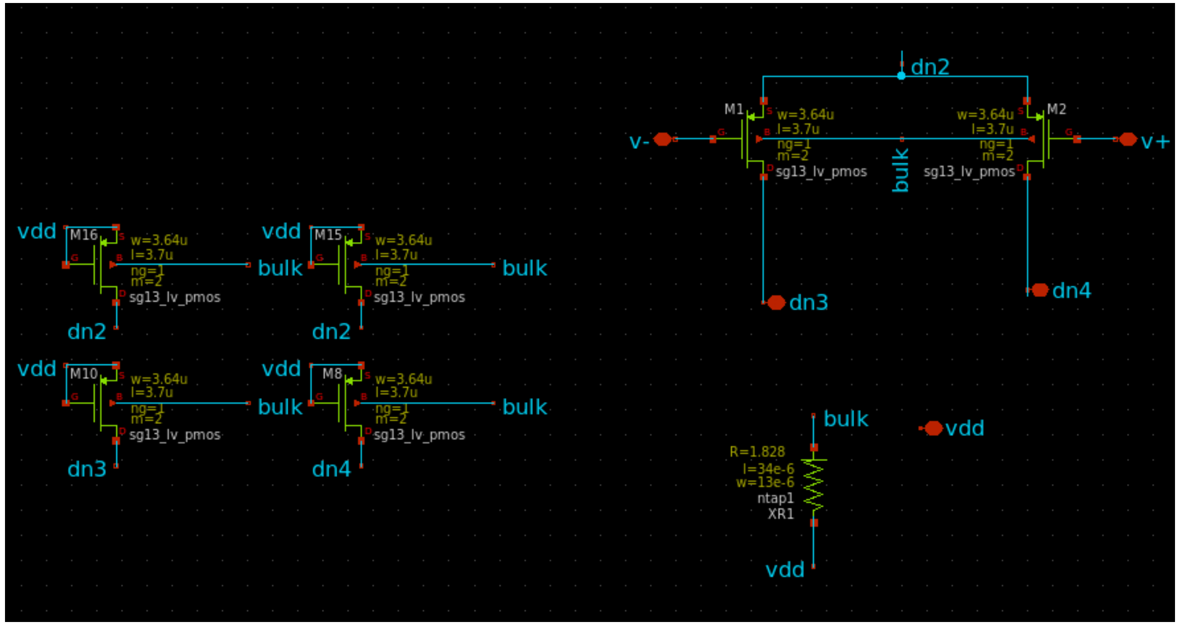
Reference: /modules/module_1_bandgap_reference/part_3_layout

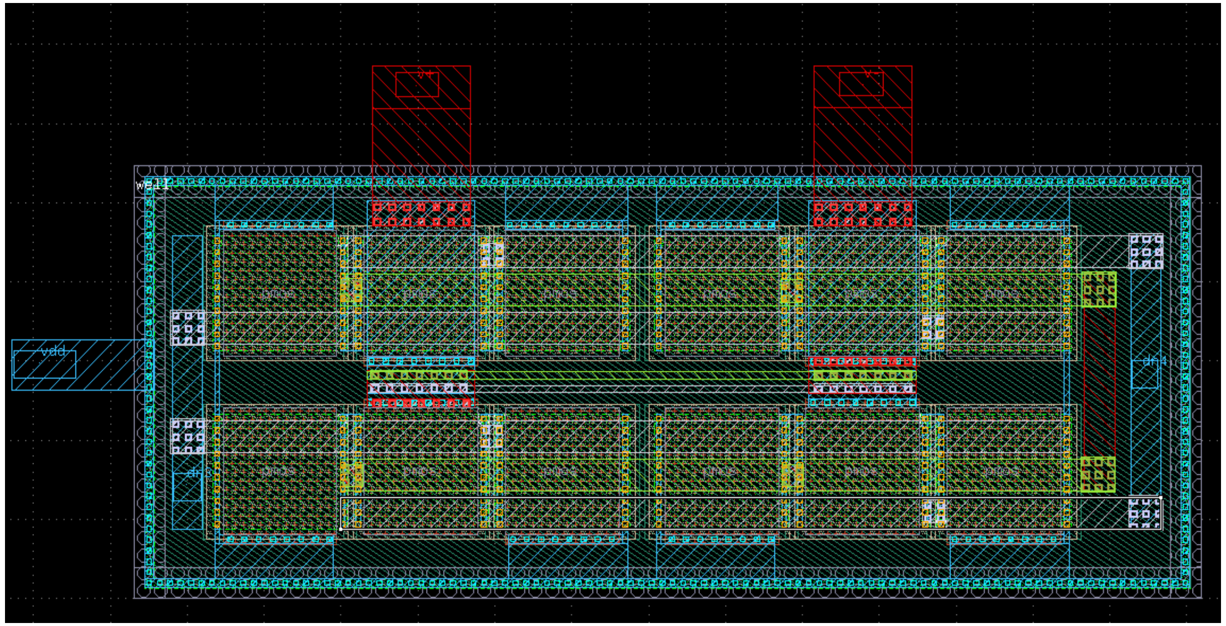
Layout Competition!

- **Concept:** OTA layout, get as far as you can!
- **Motivation:** Technical skills and awesome secret gift
- **Evaluation:** On Friday we will look at each layout and a winner will be selected
- **Metrices:** DRC/LVS errors, style and completeness
- The remaining time over the next few days can be dedicated to the layout; it doesn't need to be completed today

I will be available to offer assistance!

Back Annotation





Objects		Layout	Reference
▼ input_common_centroid ⇔ INPUT_COI	⛔	input_common_centroid	INPUT_COMMON_CENTROID
▶ Pins			
▶ Nets			
▼ Devices			
▶ ntap1 ⇔ NTAP1	⚠	$13 / ntap1 [A=28.7556, P=0.18552k]	1 / NTAP1 [A=0.442k, P=94]

Back Annotation

```
*.PININFO v-:B v+:B vdd:B dn3:B dn4:B
M1 dn3 v- dn2 bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M2 dn4 v+ dn2 bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M8 dn4 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M10 dn3 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M15 dn2 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M16 dn2 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
R1 vdd bulk ntap1 A=4.42e-10 P=9.4e-05
.ends
```

```
.subckt input_common_centroid v- v+ vdd dn3 dn4
*.PININFO v-:B v+:B vdd:B dn3:B dn4:B
M1 dn3 v- dn2 bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M2 dn4 v+ dn2 bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M8 dn4 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M10 dn3 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M15 dn2 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
M16 dn2 vdd vdd bulk sg13_lv_pmos w=3.64u l=3.7u ng=1 m=2
R1 vdd bulk ntap1 A=28.7556e-12 P=0.18552e-03
.ends
```

Objects	Layout	Reference
input_common_centroid ⇔ INPUT_COM	input_common_centroid	INPUT_COMMON_CENTROID
Pins		
Nets		
Devices		
ntap1 ⇔ NTAP1	$13 / ntap1 [A=28.7556, P=0.18552k]	1 / NTAP1 [A=28.7556, P=0.18552k]
sg13_lv_pmos ⇔ SG13_LV_PMOS	$4 / sg13_lv_pmos [L=3.7, W=7.28, AS=2.	1 / SG13_LV_PMOS [L=3.7, W=7.28]
sg13_lv_pmos ⇔ SG13_LV_PMOS	$3 / sg13_lv_pmos [L=3.7, W=7.28, AS=2.	2 / SG13_LV_PMOS [L=3.7, W=7.28]
sg13_lv_pmos ⇔ SG13_LV_PMOS	$5 / sg13_lv_pmos [L=3.7, W=7.28, AS=2.	8 / SG13_LV_PMOS [L=3.7, W=7.28]
sg13_lv_pmos ⇔ SG13_LV_PMOS	$6 / sg13_lv_pmos [L=3.7, W=7.28, AS=2.	10 / SG13_LV_PMOS [L=3.7, W=7.28]
sg13_lv_pmos ⇔ SG13_LV_PMOS	$1 / sg13_lv_pmos [L=3.7, W=14.56, AS=‹	15 / SG13_LV_PMOS [L=3.7, W=14.56]

Alternatively

- Complete unfinished tasks from today or yesterday
- Explore other design metrices for the OTA or Bandgap
- Ask questions/ get help if something wasn't clear today
- Relax ☺

What Will We Do Tomorrow?

- **QUCS-S:** Getting familiar with QUCS-S and Xyce
- **50GHz MPA:** Perform S-parameter analysis, along with DC and Harmonic Balancing
- **Matching:** Performing simple matching using online tool and tuning in QUCS-S
- **OpenEMS:** Expert talk on OpenEMS and how to interface with it using Python
- **EM Simulation:** EM simulating components for the 50GHz MPA