



Leibniz Institute  
for high  
performance  
microelectronics

# 50 GHz Medium Power Amplifier: Exploring Qucs-s & OpenEMS

Phillip Ferreira Baade–Pedersen (Design Engineer)  
Dr. Ing. Christian Wittke (Scientist)  
Frankfurt (Oder), IHP - 21/05/2025

**Projects:** BMBF → FMD-QNC (16ME0831)

# Agenda For today



Leibniz Institute  
for high  
performance  
microelectronics

**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.-Ing Volker Mühlhaus: Python Interfacing with OpenEMS using IHP Stackup*

**10:45 - 12:00 | Introduction | Hands On**

*Introduction To QUCS-S / Starting the MPA design*

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

**13:00 - 15:00 | Design of 50 GHz MPA**

*Build a simple schematic and set up the RF design flow, including S-parameter and nonlinear analysis. (Starting EM simulations)*

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

**15:30 - 17:00 | EM Simulations**

*Perform EM simulation of schematic components and analyze their impact on circuit behavior*



# Key Takeaways from Yesterday

- 0 Insights on models within the IHP Open PDK
- 0 Building the testbench for two stage OTA
- 0 Building the testbench for the BGR
- 0 Exploring mismatch simulations
- 0 Beginning the layout of two stage OTA

Questions?

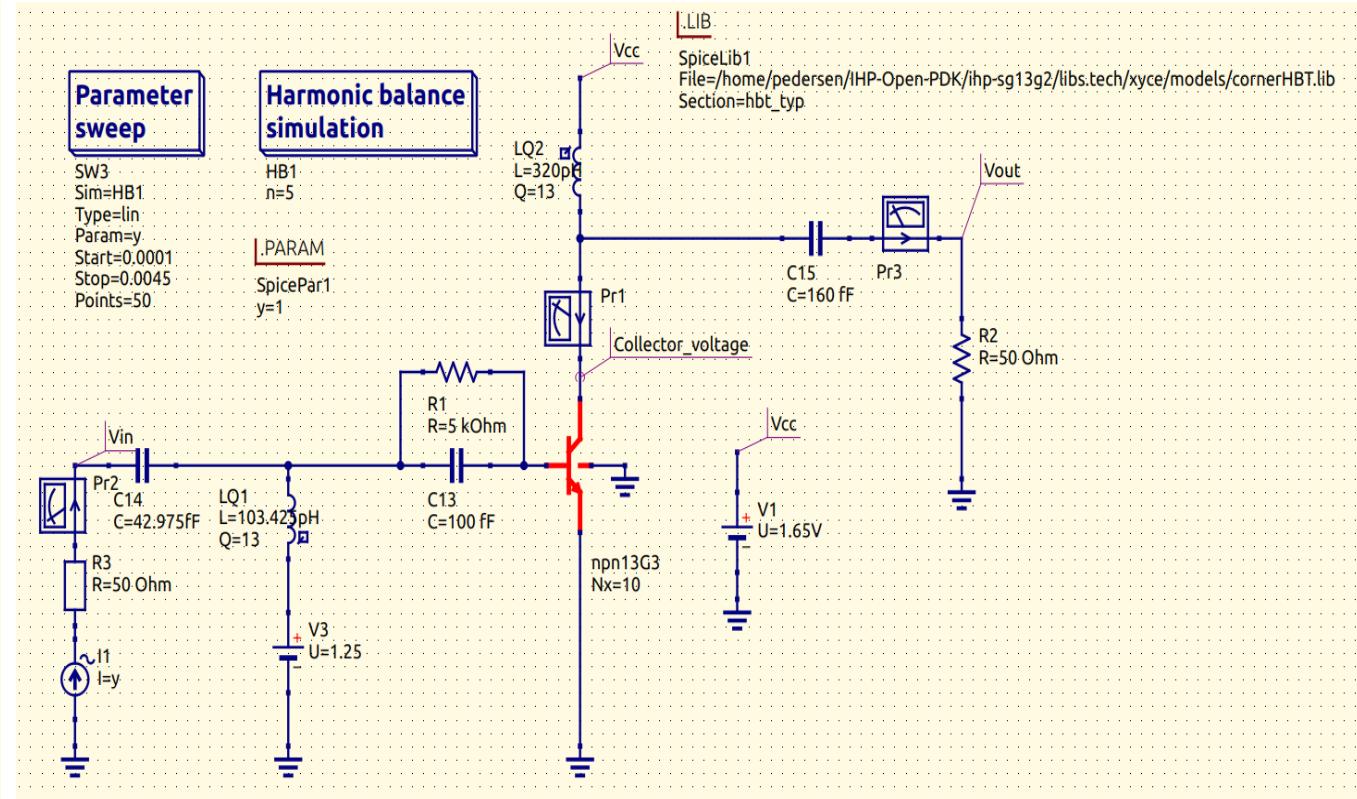
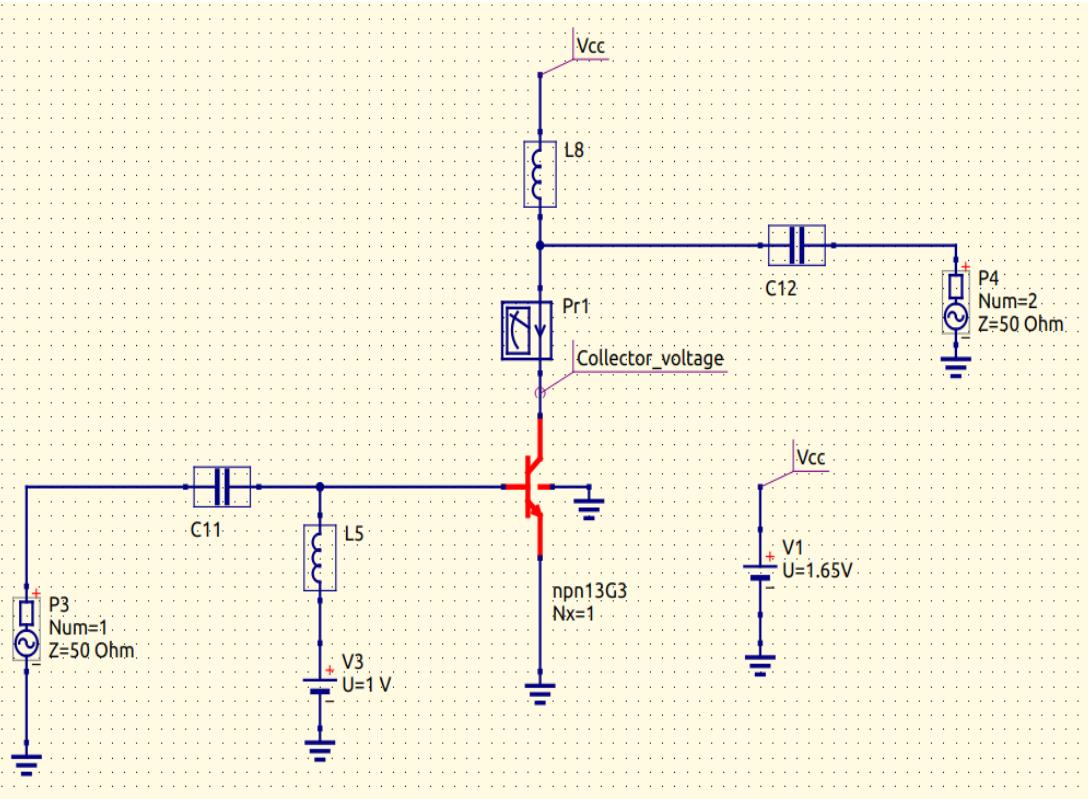
- 0 Was something from yesterday unclear?
- 0 Are we going to fast?
- 0 Do you need more time for the small exercise?

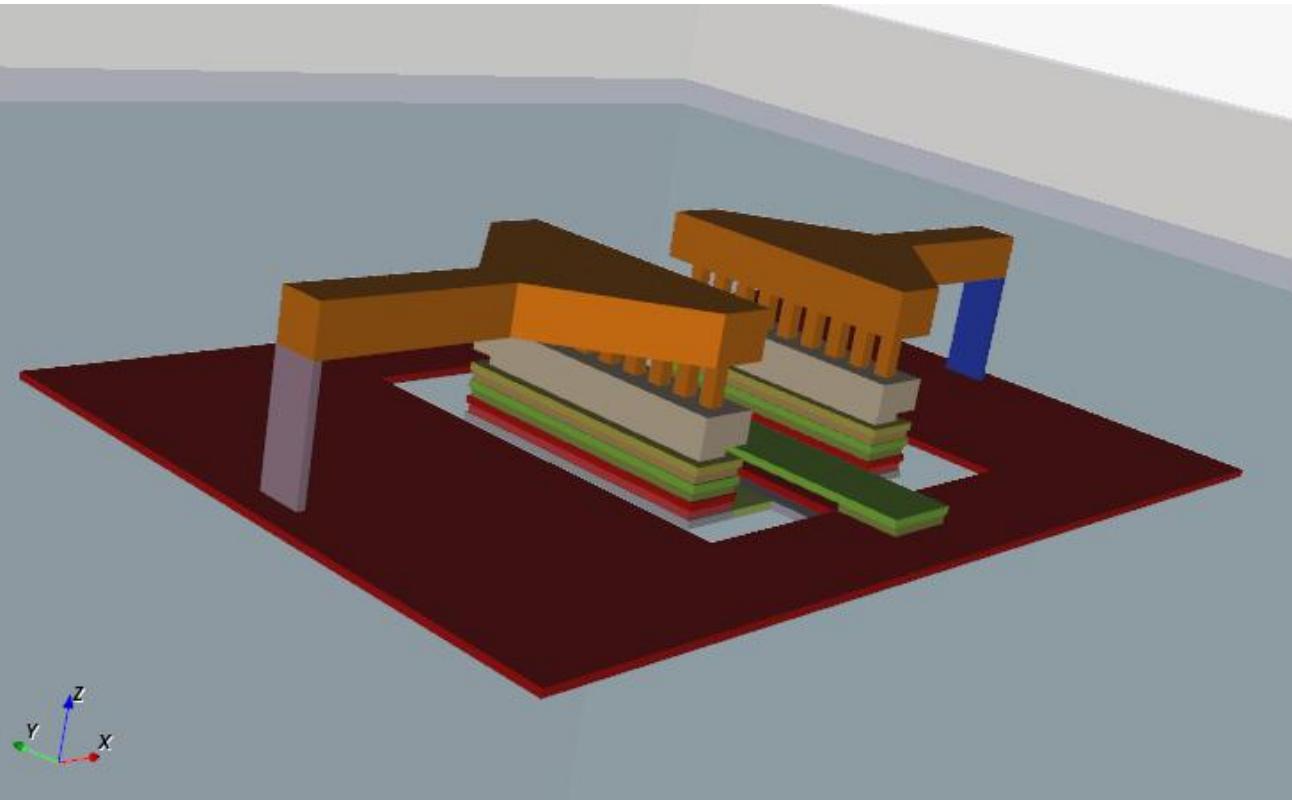
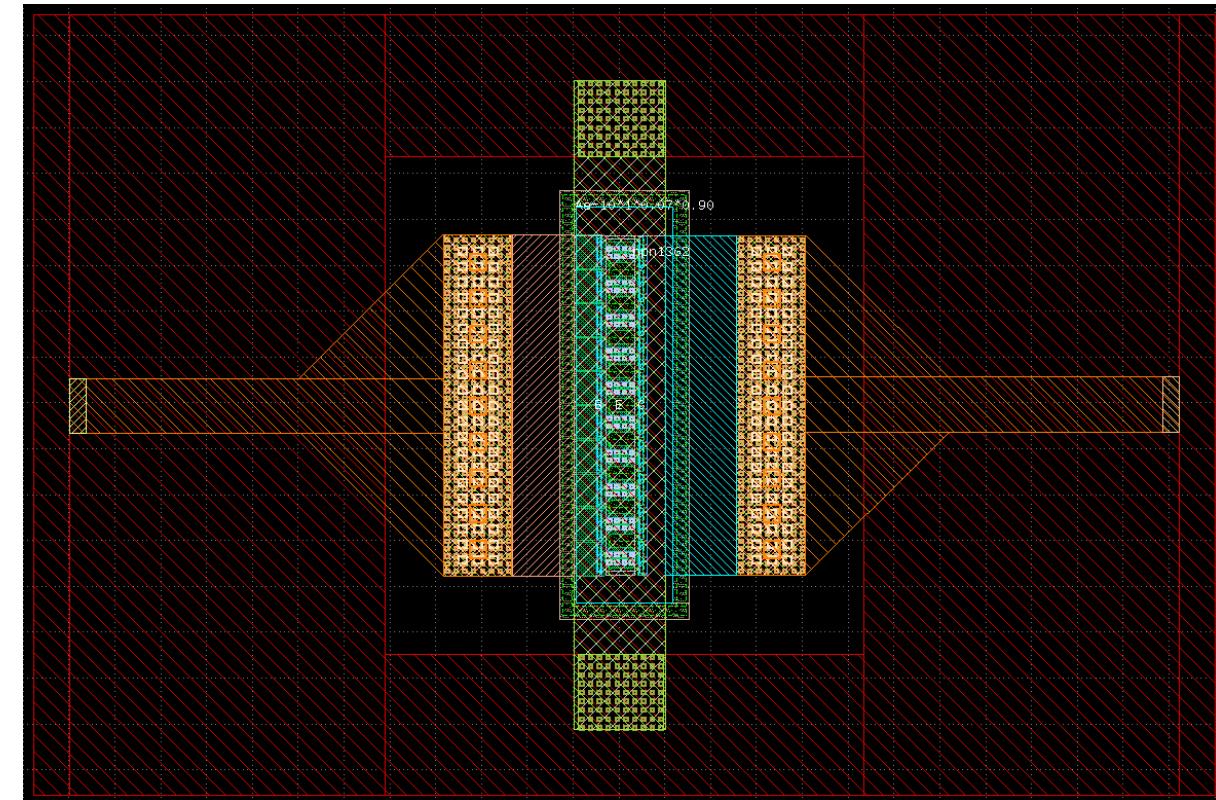


# Overview Of Todays Plan

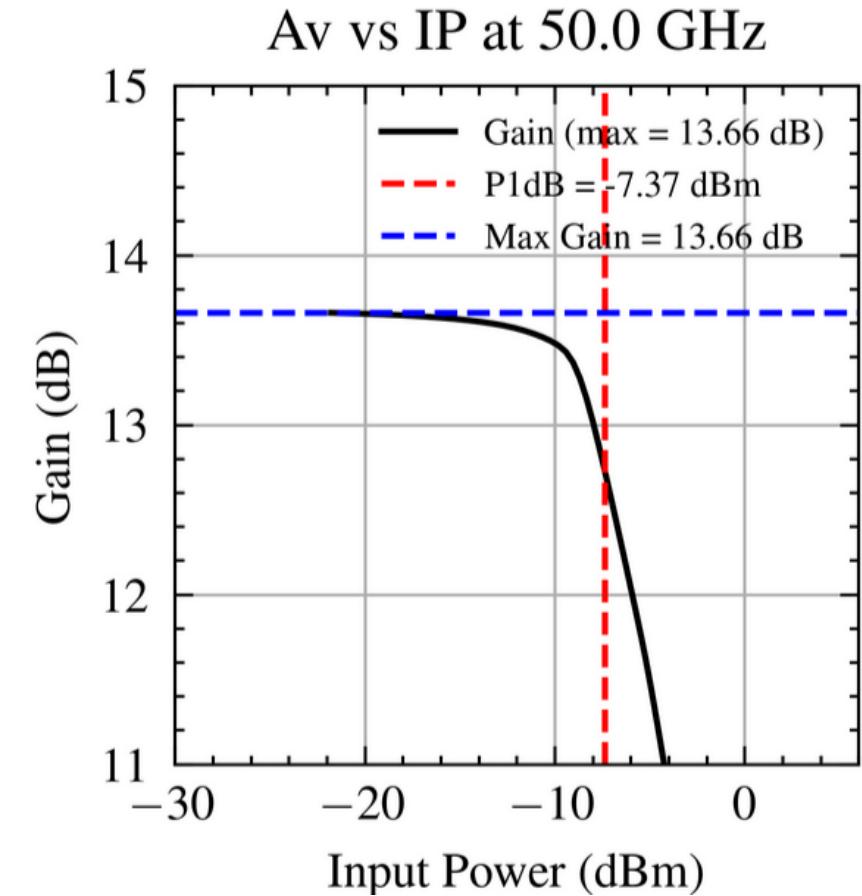
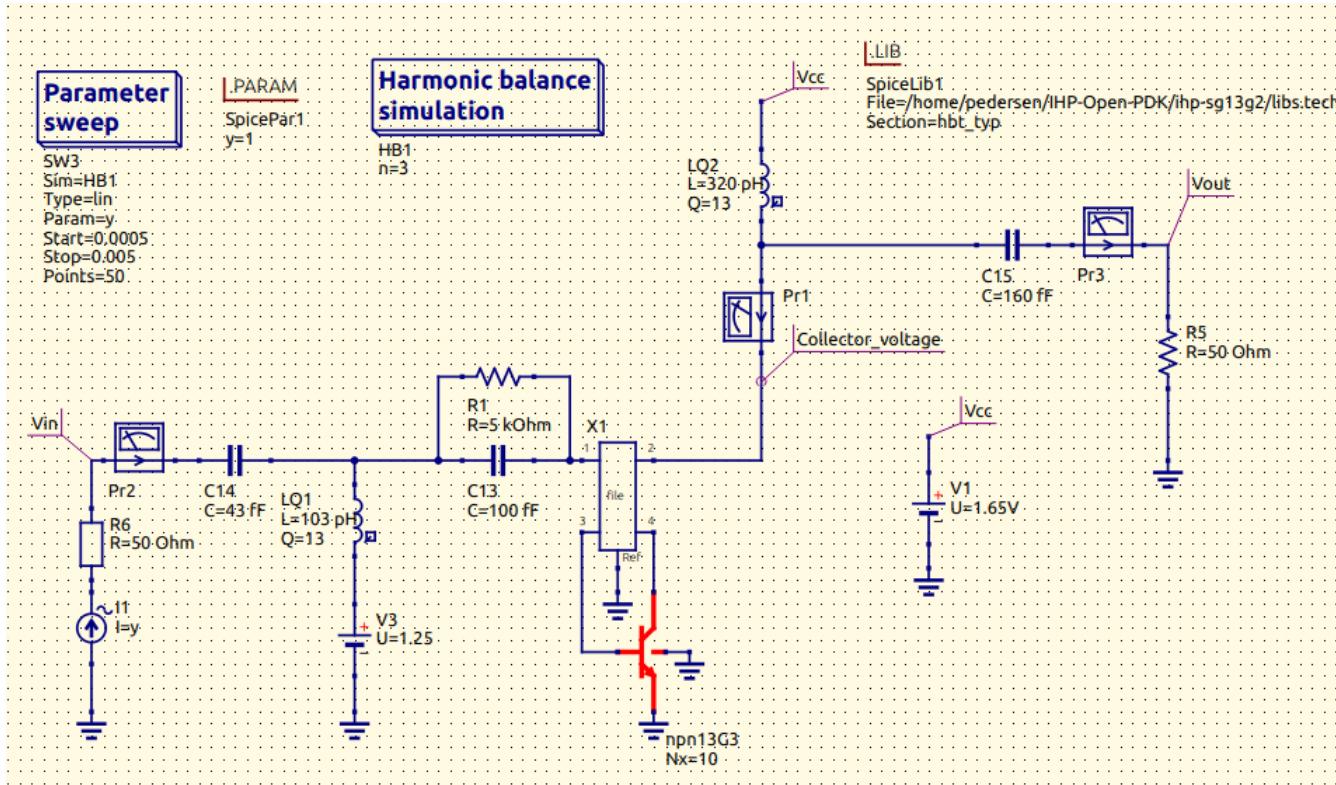
- 0 **Exploring QUCS-S:** Small introduction to Qucs-s
- 0 **MPA Schematic:** Biasing and Initial matching
- 0 **Compression Point:** Non-Linear analysis using xyce to determine compression point
- 0 **OpenEMS:** EM Simulation using OpenEMS
- 0 **Misc:** Continue OTA layout or perform EM simulations for the remaining MPA components.

# Single Stage 50 GHz MPA





# Post Processing





# Small Disclaimer

## RF Workflow in OS is challenging !

- 0 Harmonic Balancing in Xyce is challenging... Why? (sweeps, equations etc.)
- 0 This workflow propose a procedure to extract results and metrices in a semi automated way
- 0 RF workflow is based on repetition so be patient ☺

## Good News?

- 0 The OS echo system is pushing for a lot of development in this area
- 0 EM simulation with python interfacing is convenient and performs well!
- 0 For ***HB*** simulations Ngspice developers are currently implementing this



Leibniz Institute  
for high  
performance  
microelectronics

# Expert Talk

## Dr.-Ing Volker Mühlhaus

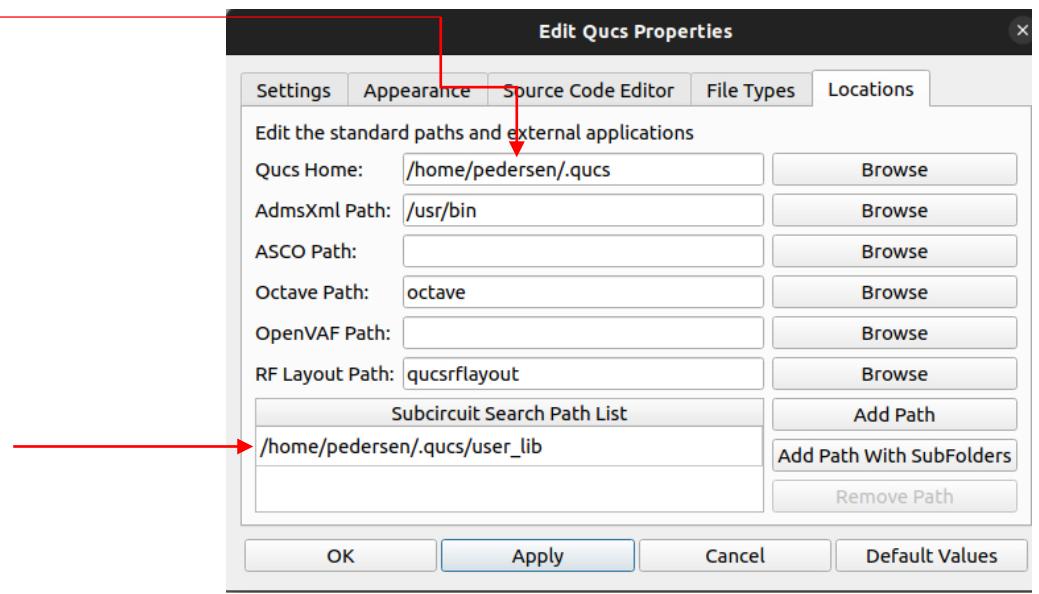
Python Interfacing with OpenEMS using IHP Stackup

# Setting Up Qucs-s



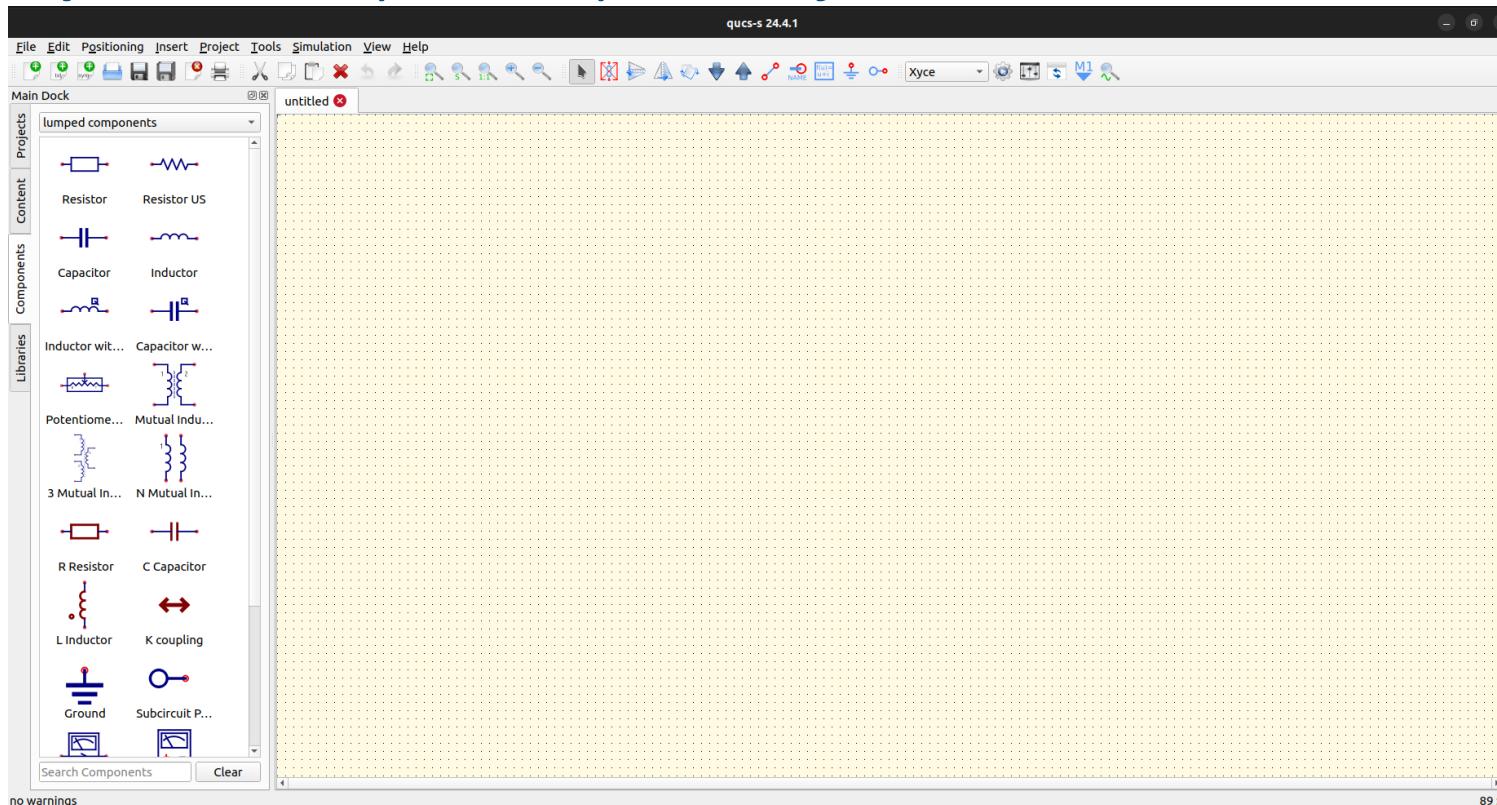
- 0 Navigate to the following directory: IHP-Open-PDK/ihp-sg13g2/libs.tech/qucs
- 0 Run the following: `python3 install.py`
- 0 Launch Qucs by writing the following in the command line: `qucs-s`
- 0 Navigate to the following location in Qucs: file -> Application settings -> Locations tab

Change to your own path!



# Simple Navigation in QUCS-S

- o To launch without specific file reference: *qucs-s*
- o Opening a specific file: *qucs-s -i path/to/file.sch*

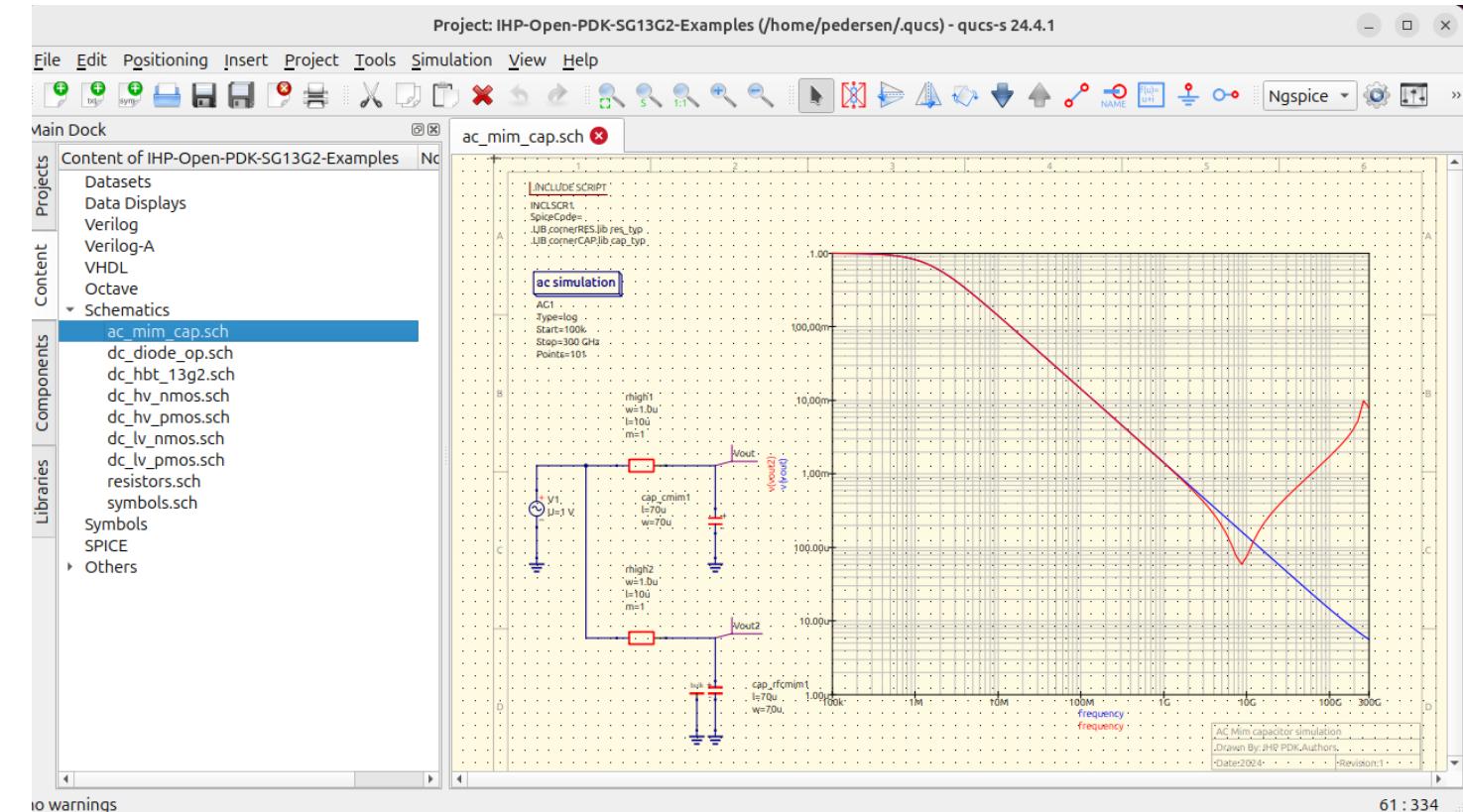
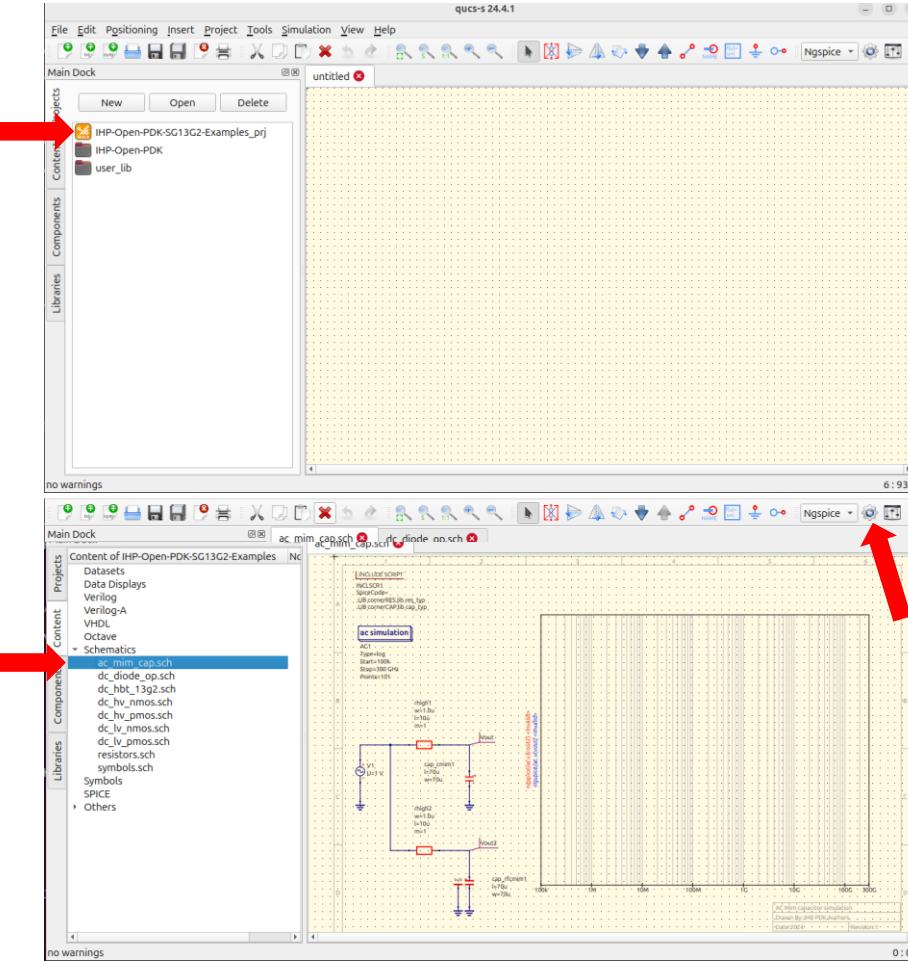


Qucs Shortcuts: <https://qucs-help.readthedocs.io/en/0.0.18/short.html>

# Simple Navigation in QUCS-S



## -0 Running An Example

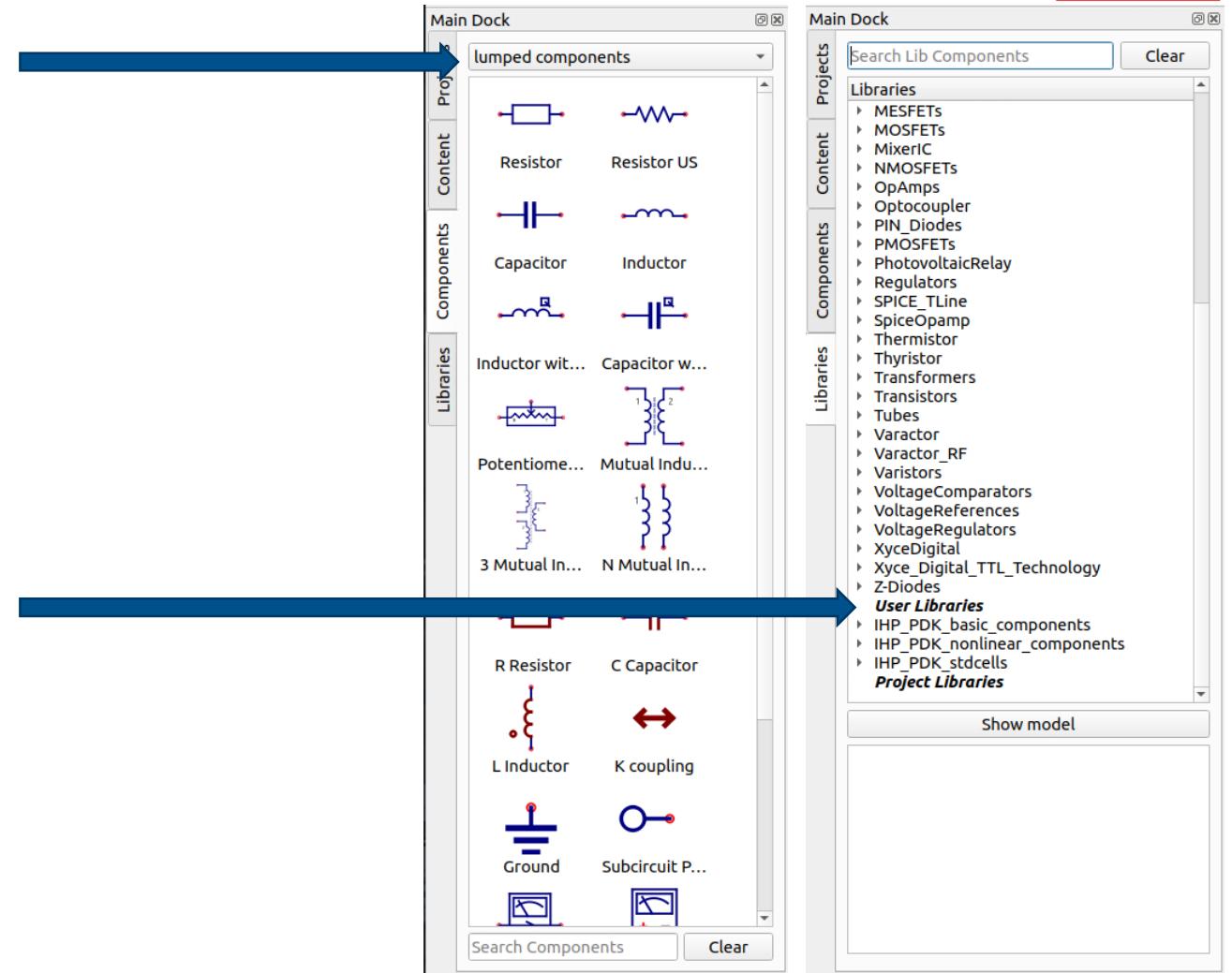


# Simple Navigation in QUCS-S

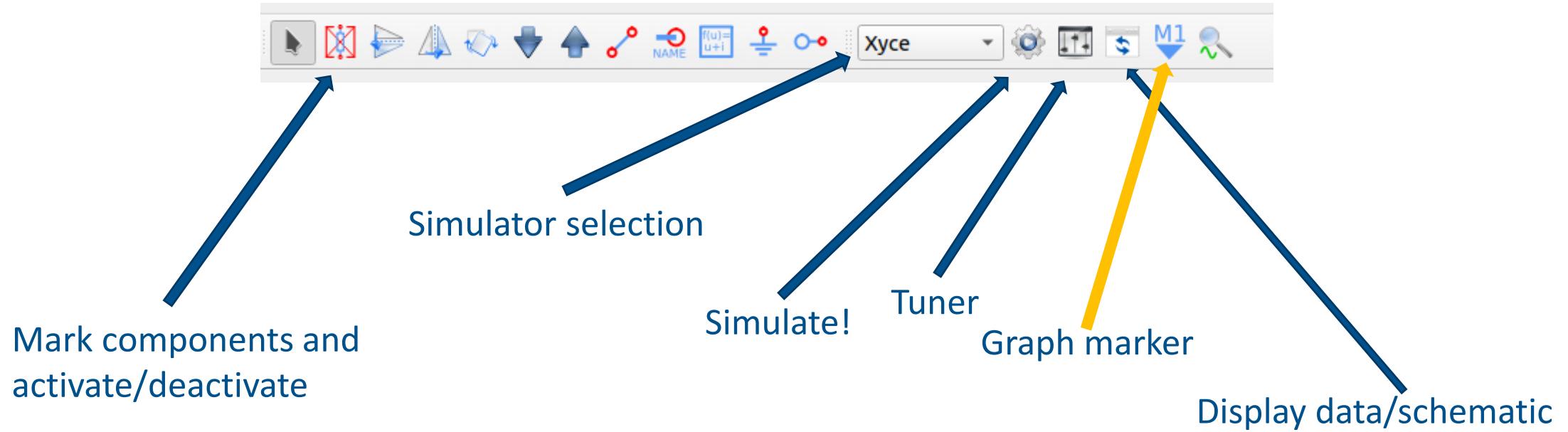


## Main Dock

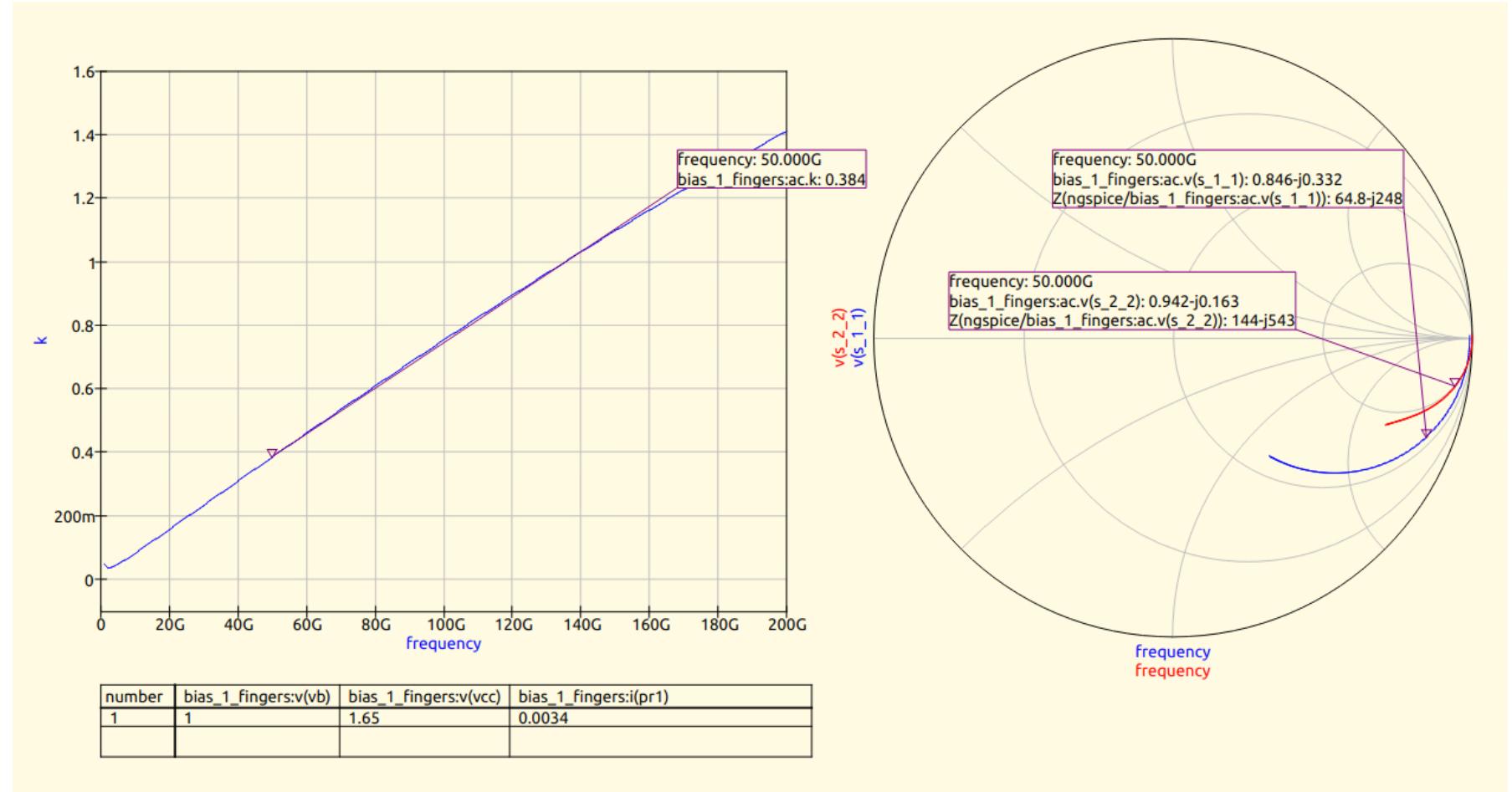
- Components: Stores standard libs and simulation blocks etc..
- The library panes stores the Open PDK components with the symbol passives (simple drag and drop)



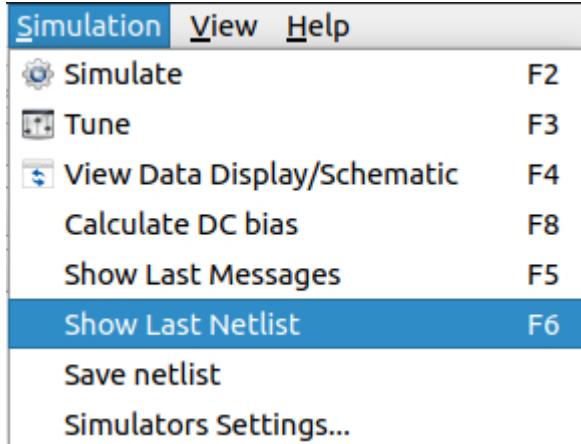
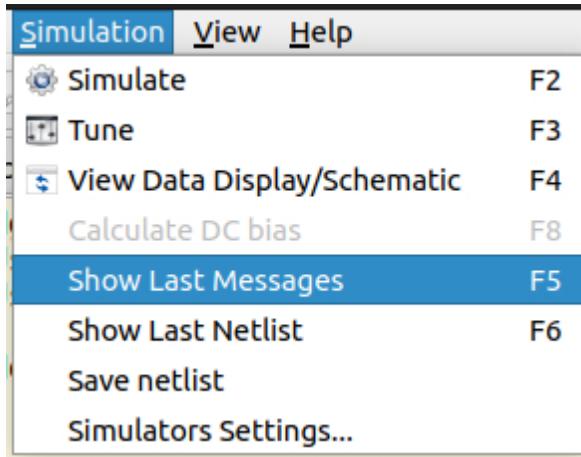
# Simple Navigation in QUCS-S



# Simple Navigation in QUCS-S



# Simple Navigation in QUCS-S



```
Ngspice started...
Using SPARSE 1.3 as Direct Linear Solver
Using SPARSE 1.3 as Direct Linear Solver

Note: No compatibility mode selected!

Circuit: * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

No. of Data Rows : 200
binary raw file "spice4qucs.sp1.plot"
Reset re-loads circuit * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
Circuit: * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

No. of Data Rows : 1
Reset re-loads circuit * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
Circuit: * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
ngspice-43 done
* Qucs 24.4.1 /home/pedersen/projects/IHP-AnalogAcademy/modules/module_2_50GHz_MPA/part_1_biasing/schematic/bias_1_fingers/bias_1_fingers.sch
.INCLUDE "/usr/local/share/qucs-s/xspice_cmlib/include/ngspice_mathfunc.inc"
.SUBCKT IHP_PDK_nonlinear_components_npn13G2 gnd c b e bn Nx=1
X1 c b e bn npn13G2 Nx={Nx}
.ENDS

.LIB cornerHBT.lib hbt_typ
VP3 _net0 0 dc 0 ac 0.632456 SIN(0 0.632456 1MEG) portnum 1 z0 50
C11 _net0 _net1 1U
L5 _net2 _net1 1U
L8 _net3 Vcc 1U
VPr1 _net3 Collector_voltage DC 0
V3 _net2 0 DC 0.97
C12 _net1 _net4 1U
VP4 _net4 0 dc 0 ac 0.632456 SIN(0 0.632456 1MEG) portnum 2 z0 50
V1 Vcc 0 DC 1.65

Xnpn13G3 0 Collector_voltage _net1 0 gnd IHP_PDK_nonlinear_components_npn13G2 Nx=10
.control
SP LIN 200 1G 200G
let k = (1 - abs(s_1_1)^2 - abs(s_2_2)^2 + abs(s_1_1 * s_2_2 - s_1_2 * s_2_1)^2) / (2 * abs(s_1_2 * s_2_1))
write spice4qucs.sp1.plot S_1_1 Y_1_1 Z_1_1 S_1_2 Y_1_2 Z_1_2 S_2_1 Y_2_1 Z_2_1 S_2_2 Y_2_2 Z_2_2 k
destroy all
reset

op
print v(Collector_voltage) i(VPr1) v(Vcc) > spice4qucs.dcl.ngspice.dc.print
destroy all
reset

exit
.endc
.END
```

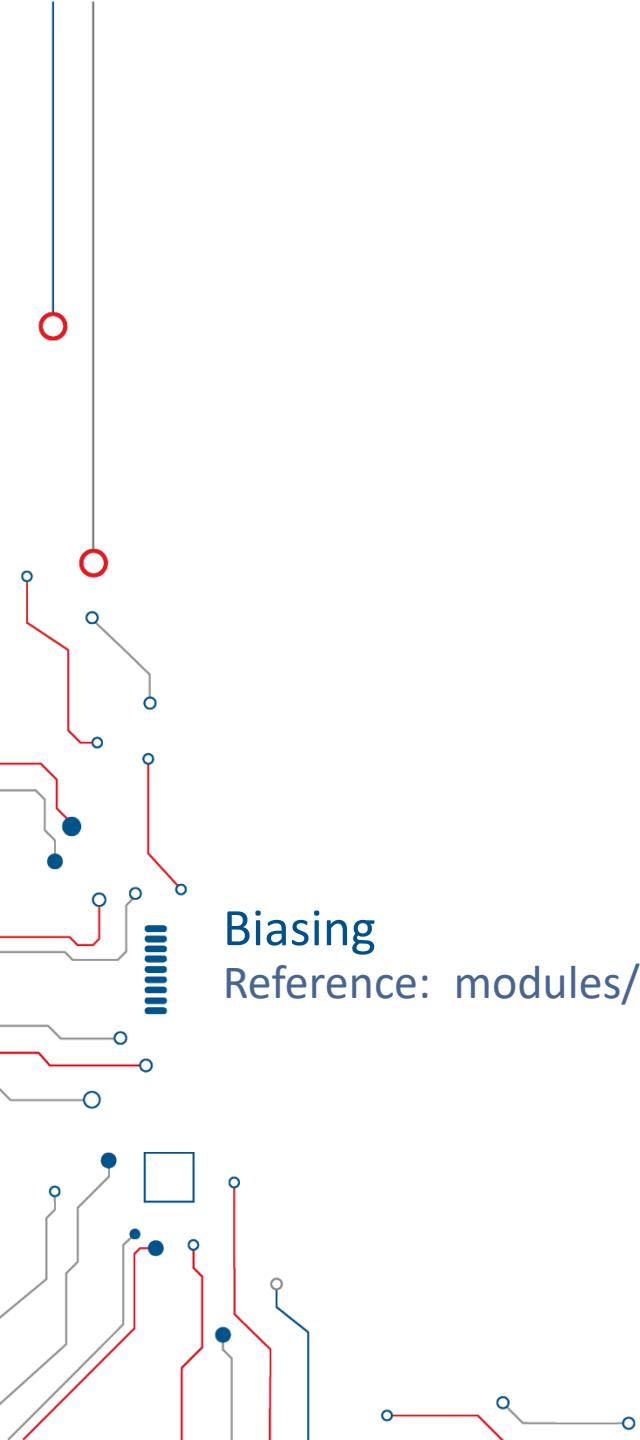
# Catching Up / Lunch Time !

Next session:  
Biasing

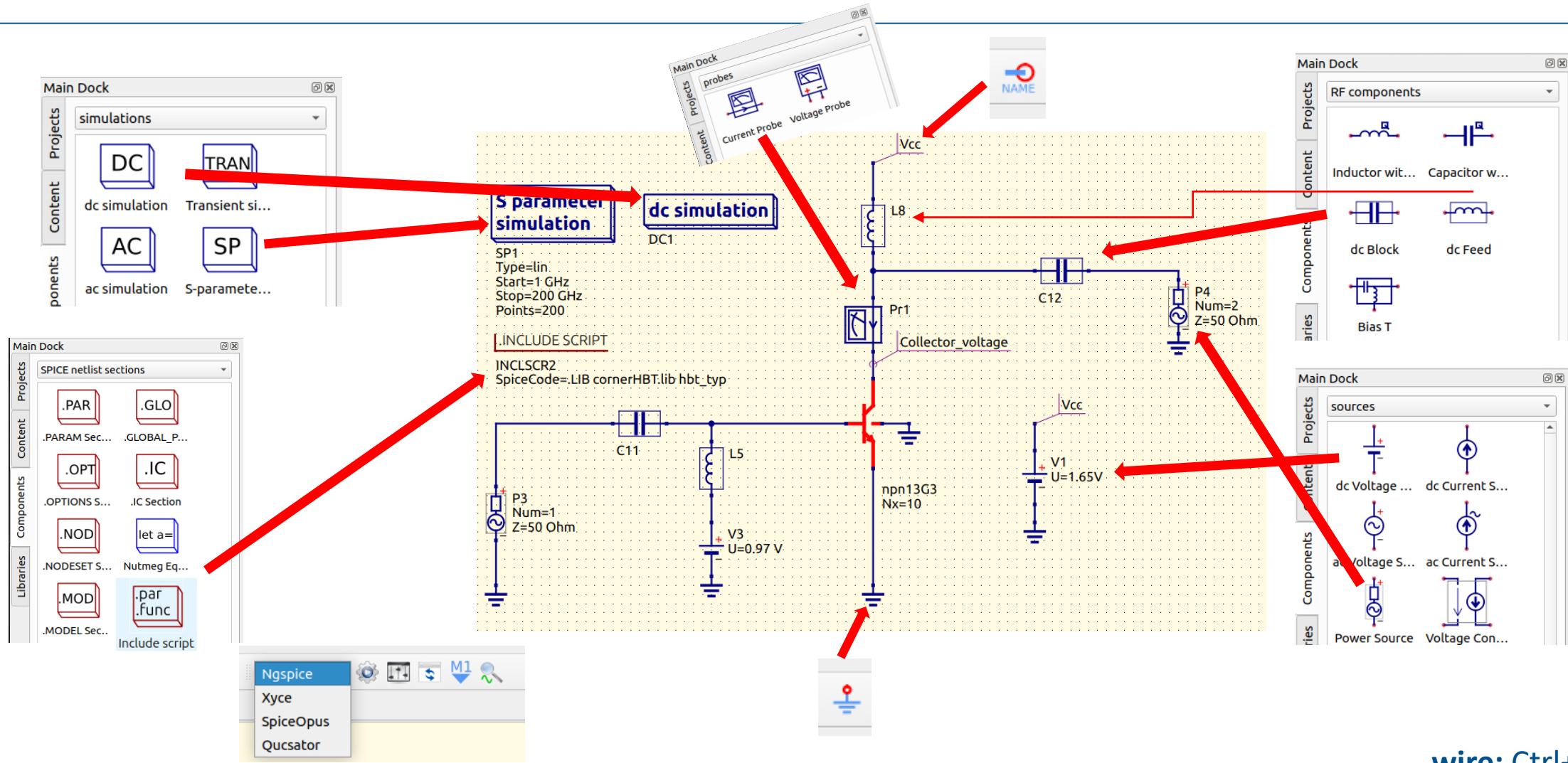
# Part 1

## Biasing

Reference: [modules/module\\_2\\_50GHz\\_MPA/part\\_1\\_biasing](#)



# Building The Schematic



## wire: Ctrl+W

# Building The Schematic



Main Dock

Search Lib Components Clear

Projects

Libraries

- PWM\_Controller
- PhotovoltaicRelay
- SPICE\_TLine
- SpiceOpamp
- Thermistor
- Thyristor
- Transformers
- Transistors
- Tubes
- Varactor
- Varactor\_RF
- Varistors
- VoltageComparators
- VoltageReferences
- VoltageRegulators
- Xanalogue
- Z-Diodes

User Libraries

- IHP\_PDK\_basic\_components
- IHP\_PDK\_nonlinear\_components
  - dantenna
  - dpantenna
  - sg13\_lv\_nmos
  - sg13\_hv\_nmos
  - sg13\_lv\_pmos
  - sg13\_hv\_pmos
  - npn13G2
  - npn13G2l
  - npn13G2v

IHP\_PDK\_stdcells

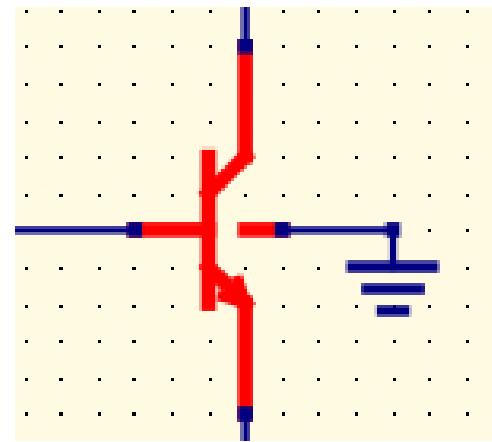
Project Libraries

**User Libraries**

- IHP\_PDK\_basic\_components
- IHP\_PDK\_nonlinear\_components
  - dantenna
  - dpantenna
  - sg13\_lv\_nmos
  - sg13\_hv\_nmos
  - sg13\_lv\_pmos
  - sg13\_hv\_pmos
  - npn13G2
  - npn13G2l
  - npn13G2v

**IHP\_PDK\_stdcells**

**Project Libraries**

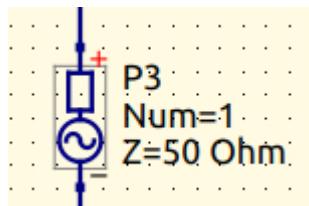


# Check Component Parameters



You have to set different parameters that aren't necessarily shown!

Example:



Double Click

!!!

Edit Component Properties

ac power source

Name: P3  display in schematic

Properties

Name	Value	display	Description
Num	1	yes	number of the port
Z	50 Ohm	yes	port impedance
P	0 dBm	no	(available) ac power in Watts
f	50 GHz	no	frequency in Hertz
Temp	26.85	no	simulation temperature in degree Celsius
EnableTran	true	no	enable transient model as sine source [true,false]

Num  
number of the port  
  
   
 display in schematic

Add Remove  
Move Up Move Down  
Fill from SPICE .MODEL

OK Apply Cancel

Same for Port 2

# Running The Simulation



Simulate with external simulator

Simulation console

```
Ngspice started...
Using SPARSE 1.3 as Direct Linear Solver
Using SPARSE 1.3 as Direct Linear Solver

Note: No compatibility mode selected!

Circuit: * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/
bias_1_fingers.sch

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

No. of Data Rows : 200
binary raw file "spice4qucs.sp1.plot"
Reset re-loads circuit * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/
bias_1_fingers.sch

Circuit: * qucs 24.4.1 /home/pedersen/projects/ihp-analogacademy/modules/module_2_50ghz_mpa/part_1_biasing/schematic/bias_1_fingers/
bias_1_fingers.sch

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
```

! Simulation started on: Fri Apr 25 10:22:34 2025  
✓ Simulation successful. Now place diagram on schematic to plot the result.

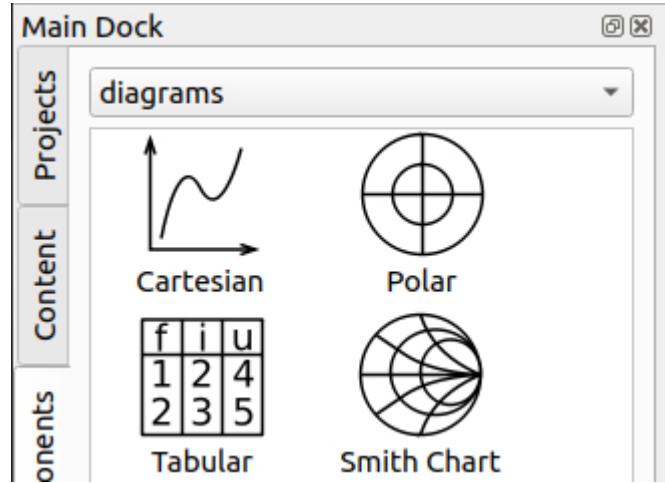
100%

Stop Save netlist Exit

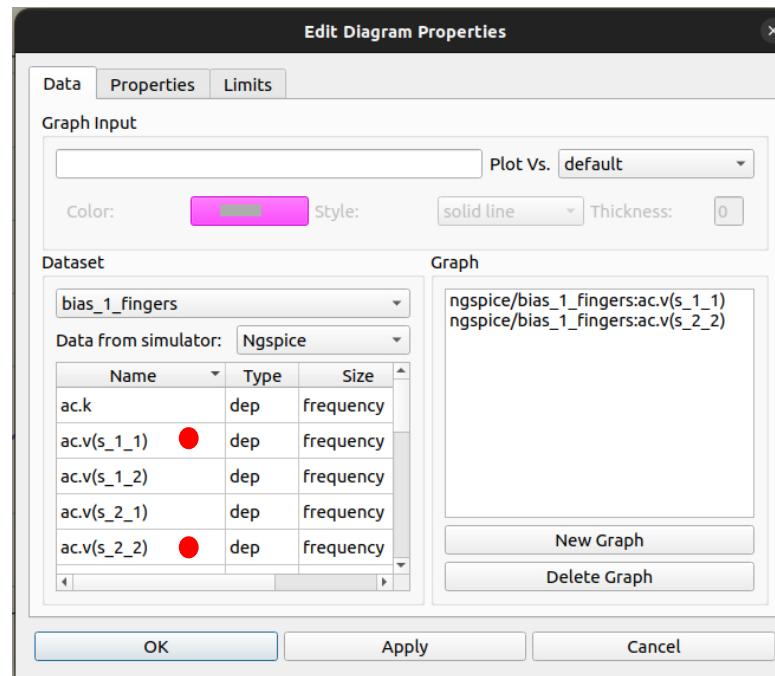
# Viewing The First Results



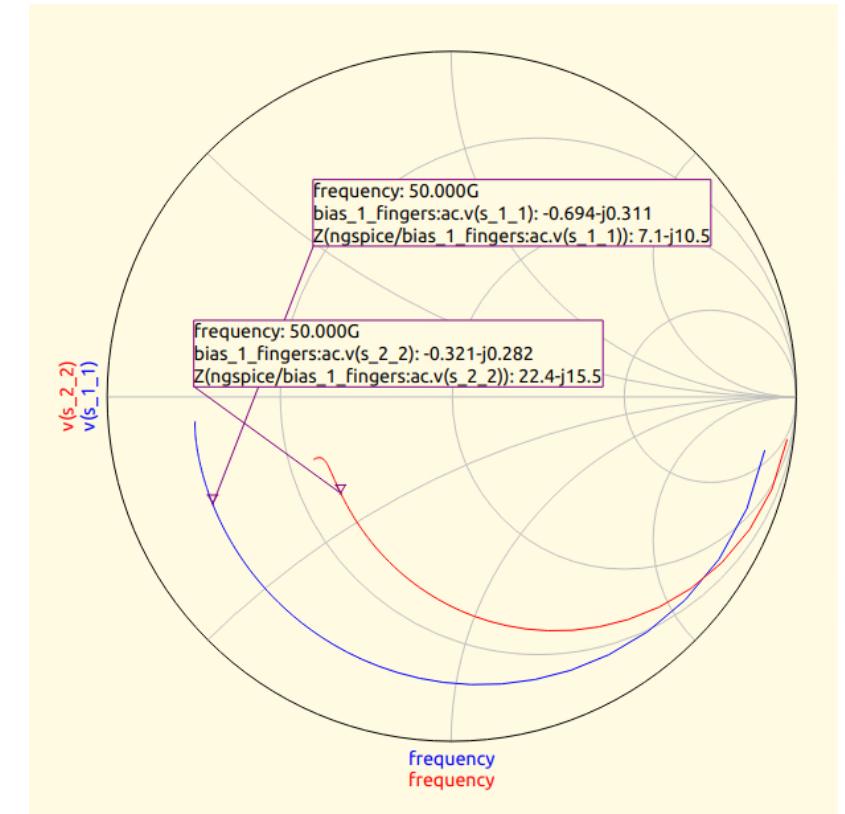
Start by switching to data display



Select the Smith Chart



Double click the S11 and S22



View the data and insert markers with M1  
Use arrow keys to move the marker

# Inserting Equation

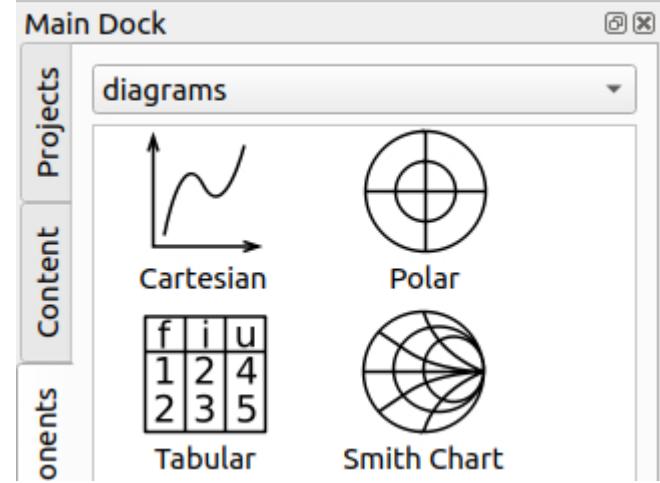


The screenshot illustrates the workflow for inserting an equation into a simulation setup:

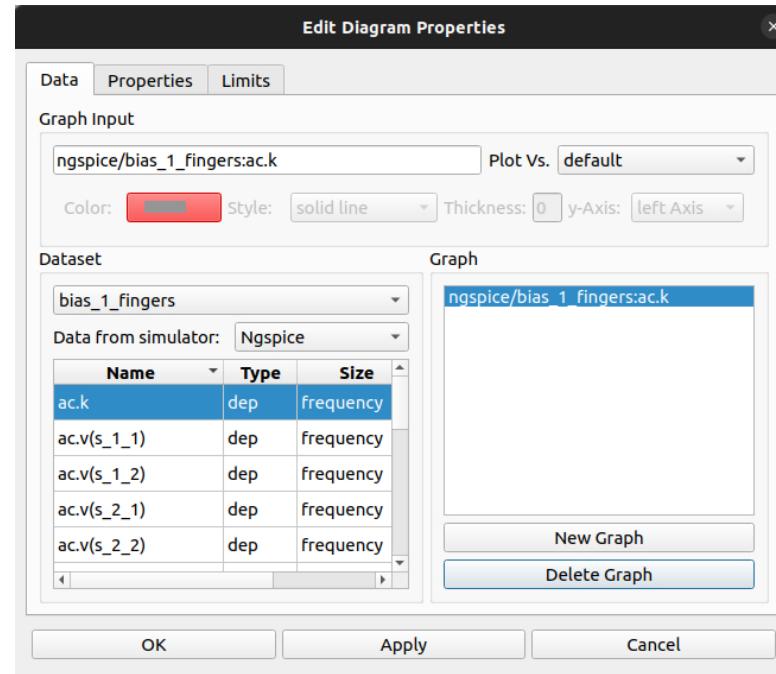
- Toolbar:** Shows various icons for circuit elements like resistors, capacitors, voltage sources, and current sources.
- Schematic Area:** Displays a component labeled "Nutmeg" with the following properties:
  - Name: NutmegEq1
  - Simulation: SP1
  - Value:  $k = (1 - \text{abs}(s_{1\_1})^2 - \text{abs}(s_{2\_2})^2 + \text{abs}(s_{1\_1} * s_{2\_2} - s_{1\_2} * s_{2\_1})^2) / (2 * \text{abs}(s_{1\_2} * s_{2\_1}))$
  - Description: Simulation name
- Properties Panel:** Shows the component's properties with the following table:

Name	Value	display	Description
Simulation	SP1	yes	Simulation name
k	$(1 - \text{abs}(s_{1\_1})^2 - \text{abs}(s_{2\_2})^2 + \text{abs}(s_{1\_1} * s_{2\_2} - s_{1\_2} * s_{2\_1})^2) / (2 * \text{abs}(s_{1\_2} * s_{2\_1}))$	yes	
- Equation View:** A summary of the inserted equation:  $k=(1 - \text{abs}(s_{1\_1})^2 - \text{abs}(s_{2\_2})^2 + \text{abs}(s_{1\_1} * s_{2\_2} - s_{1\_2} * s_{2\_1})^2) / (2 * \text{abs}(s_{1\_2} * s_{2\_1}))$
- Stability / K-Factor:** A gear icon representing the analysis or simulation step.

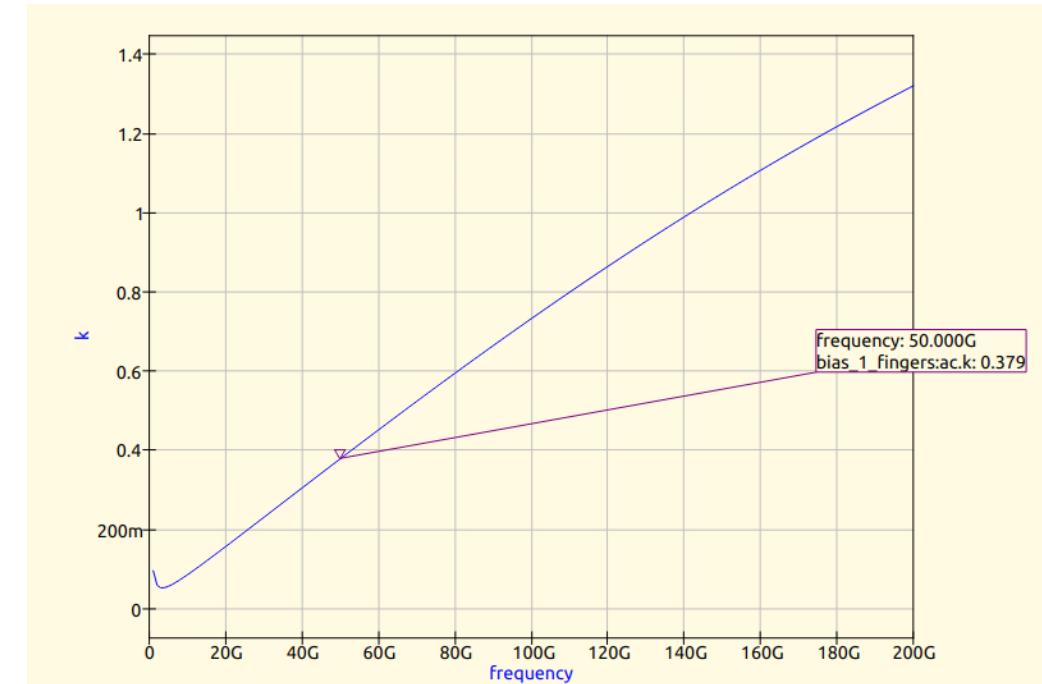
# Viewing The K-Factor



Select the Cartesian plot

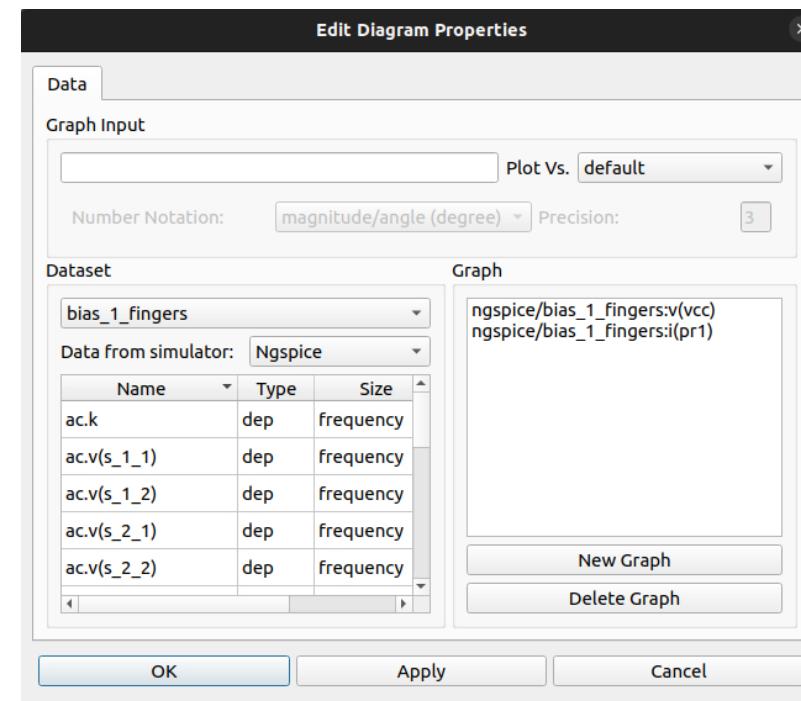
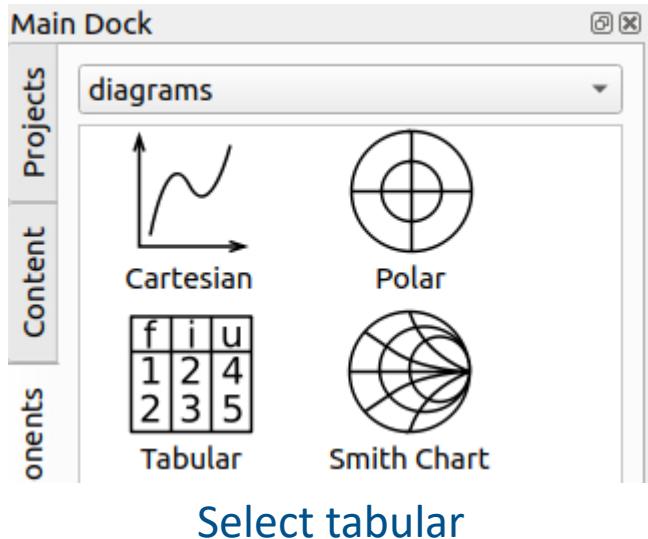


Select the K-factor



Clearly not stable!

# Viewing Data From Table



Select the collector voltage and current

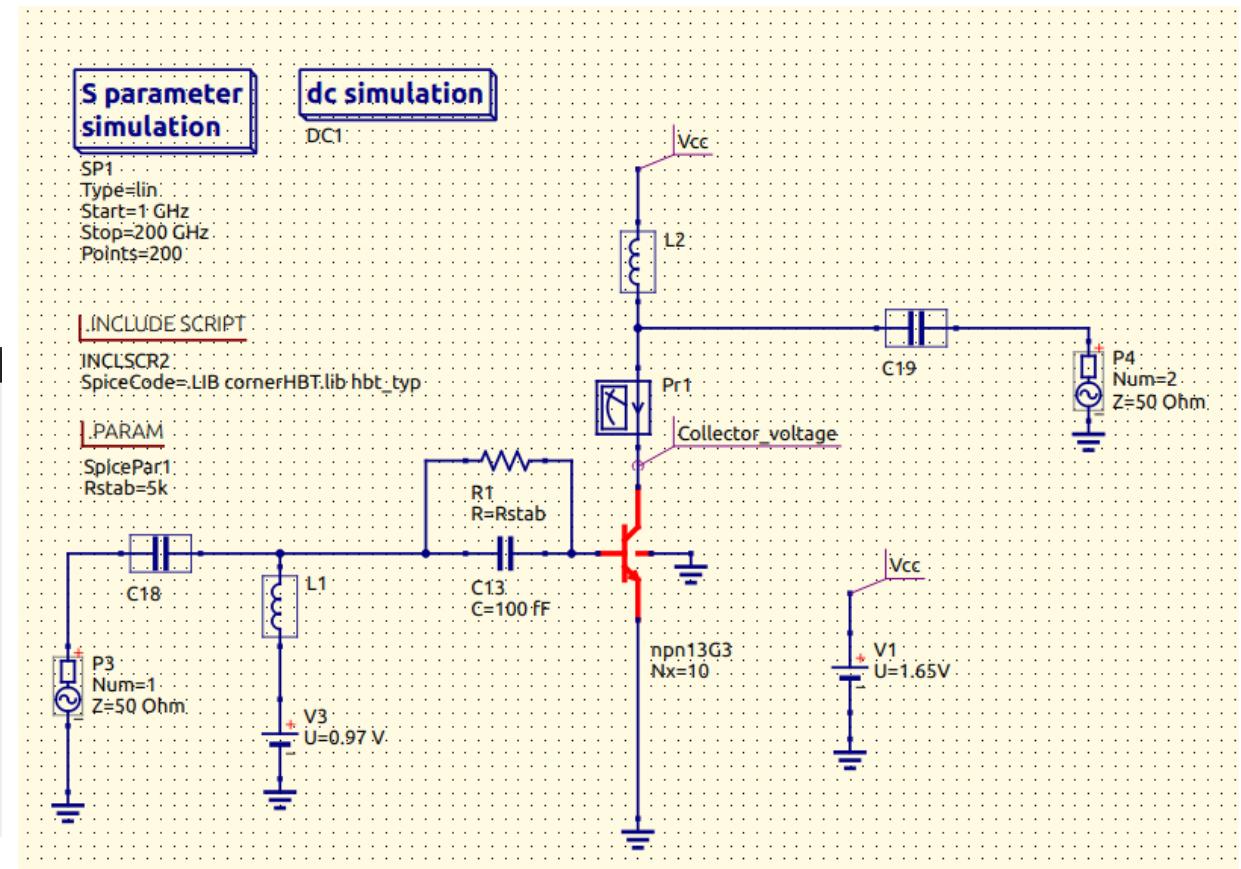
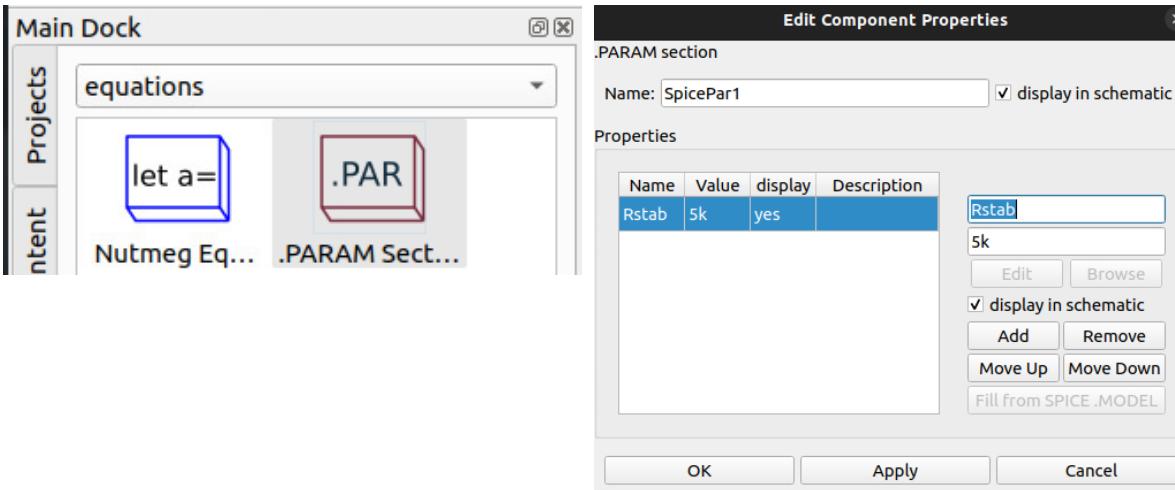
number	bias_1_fingers:v(vcc)	bias_1_fingers:i(pr1)
1	1.65	0.0269

Checking some bias points

# Tuning Components

## Introducing Stability Comps

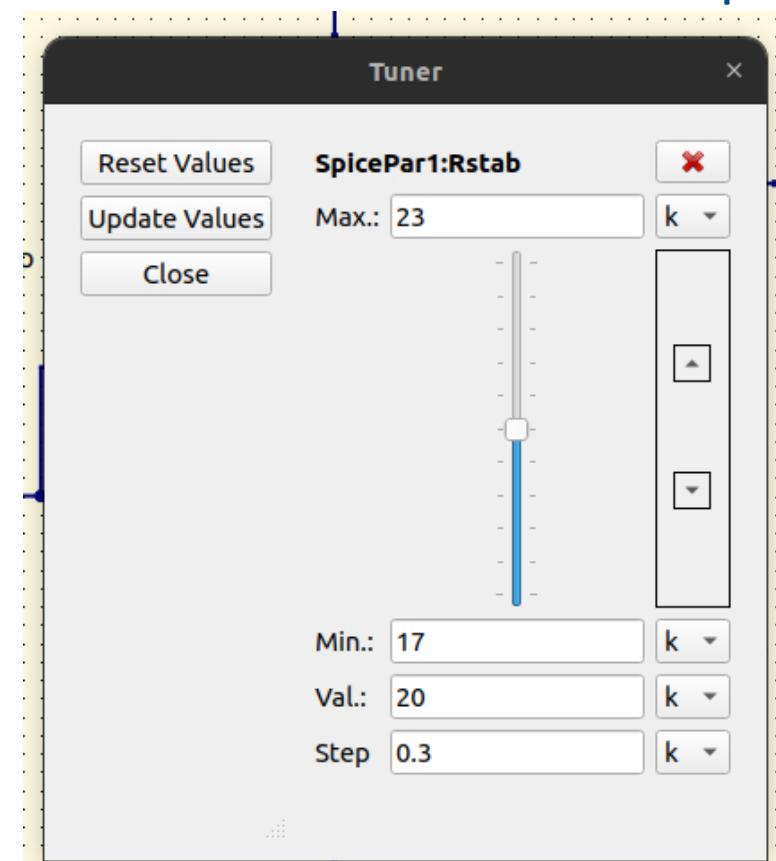
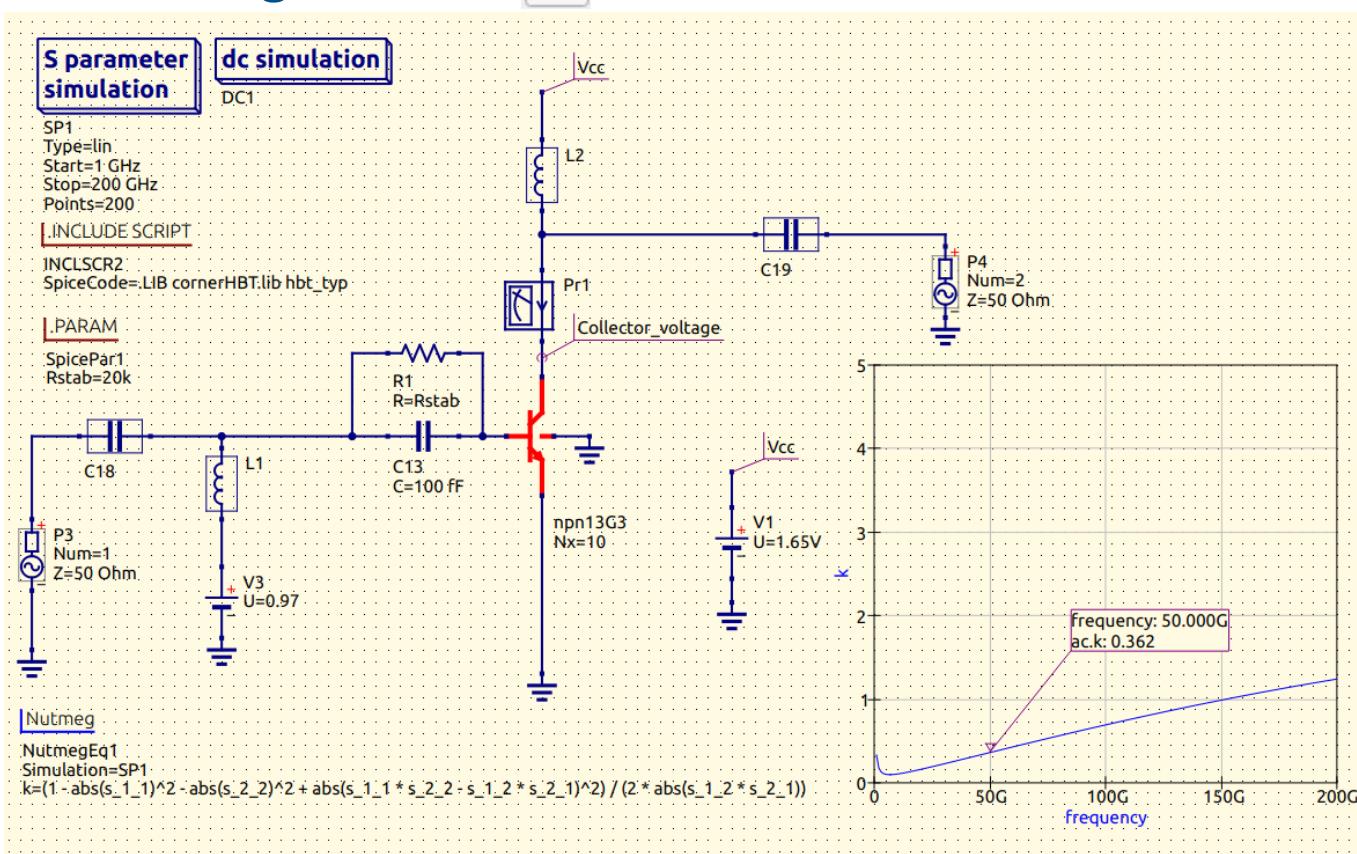
- We want to tune the resistance of the base resistor!
- We introduce a .Param block:



- Also set the parameter in the resistor

# Tuning Components

Instantiate the Cartesian plot inside the schematic and view the K-factor From here open the tuning function



Select the Rstab parameter in the .param block and start tuning



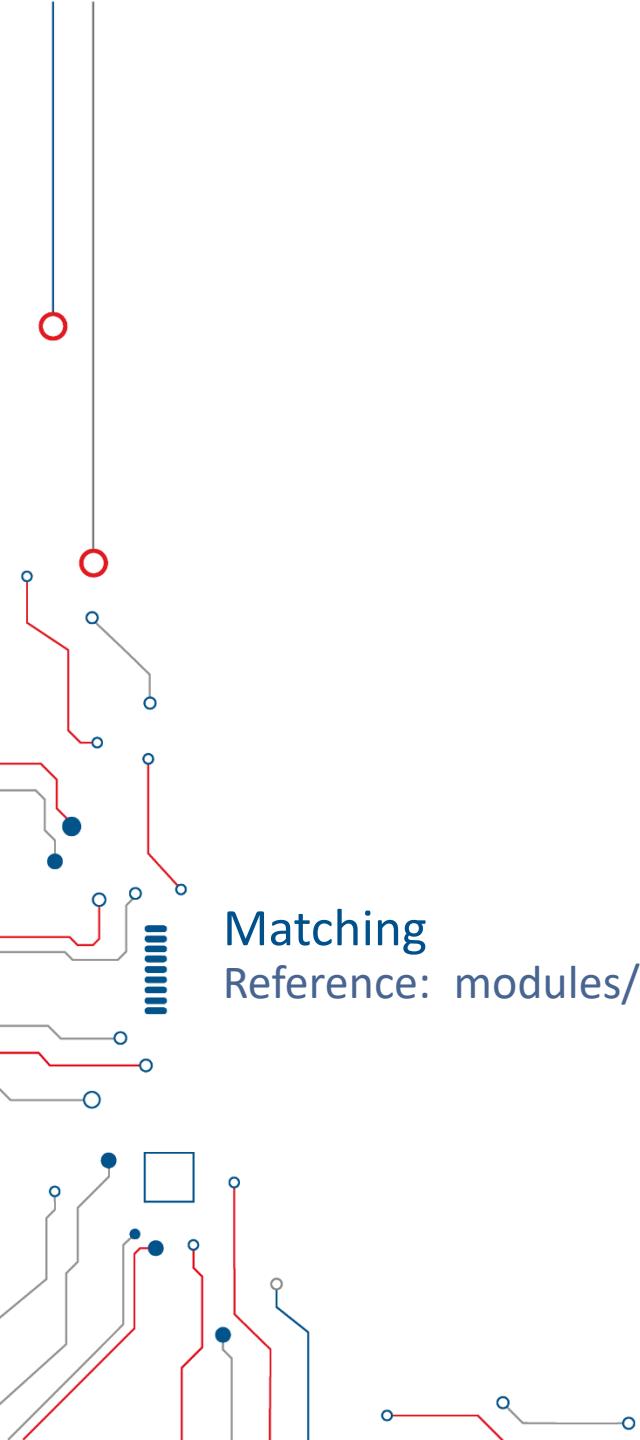
# Stabilize The Amplifier ☺

- 0 Try as best as you can to stabilize the amplifier
- 0 You can employ any method you like (emitter degeneration, biasing etc...)
- 0 Its not critical to stabilize the amplifier since we only want to create a appropriate flow ☺
- 0 If unsure, stabilize the amp by adjusting components for  $K > 1$  at 50 GHz.

# Part 2

## Matching

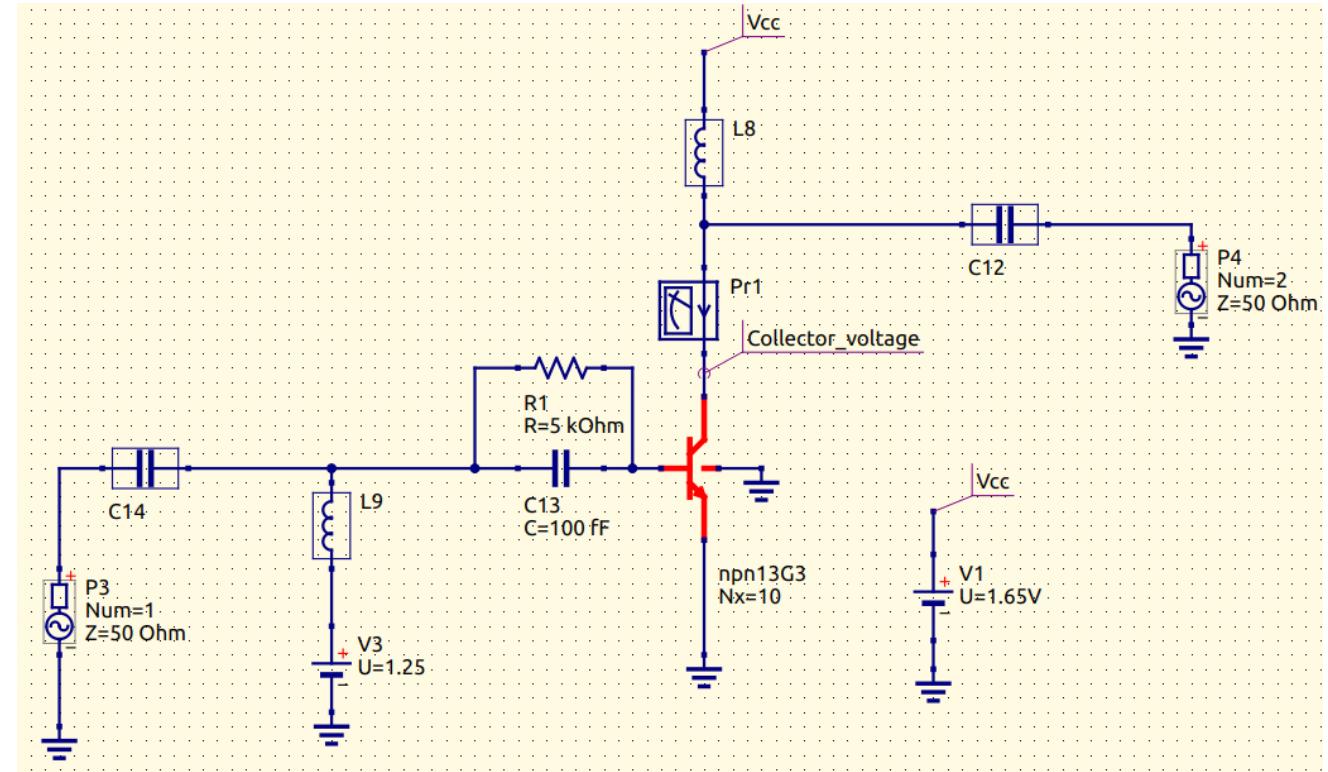
Reference: [modules/module\\_2\\_50GHz\\_MPA/part\\_2\\_matching\\_ideal](#)



# Matching



- To move forward we will use the schematic seen to the right
- This is simply done to keep quite consistent with the first iteration of the design earlier
- You may continue with your own schematic to keep better stability factor 😊

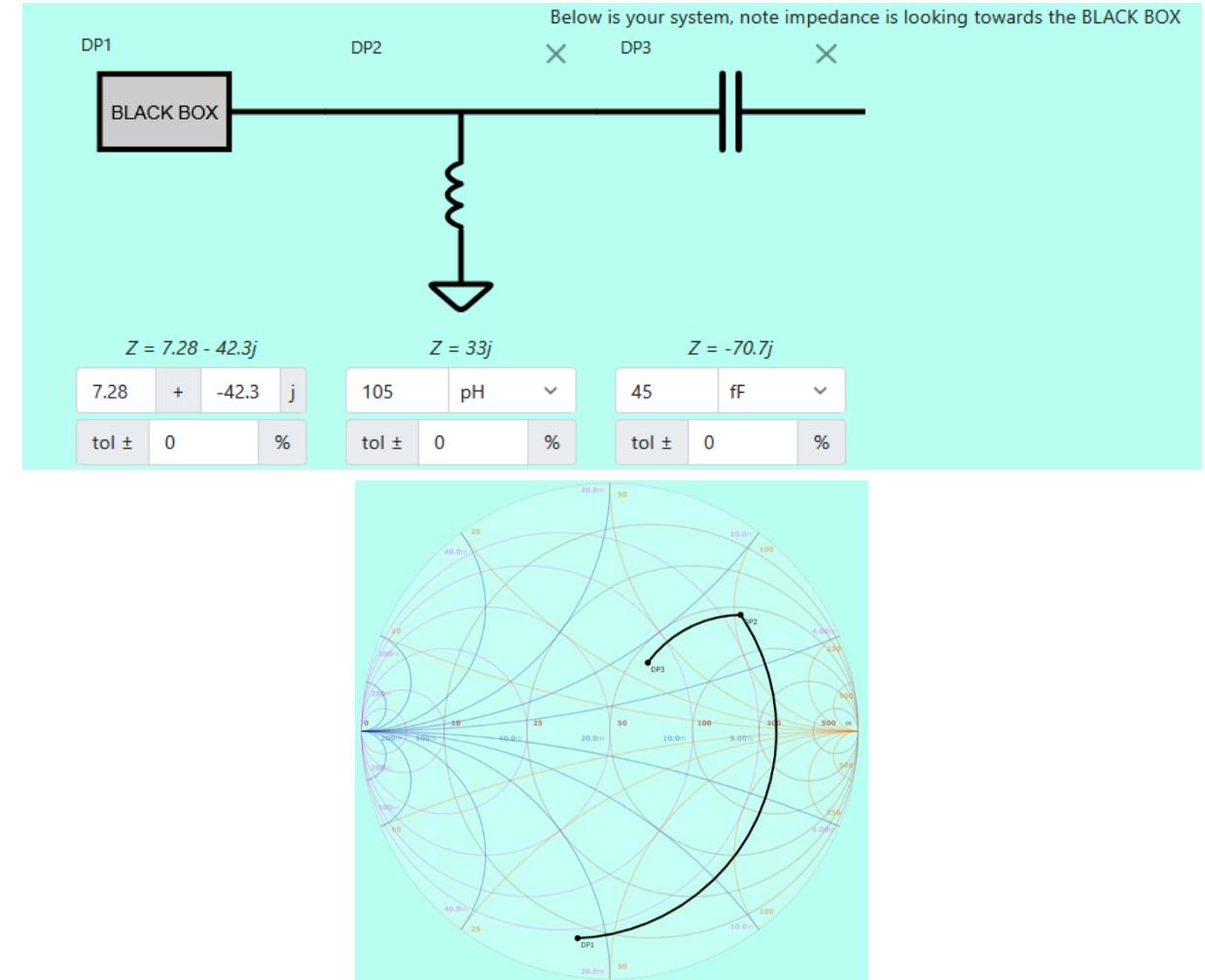


# Online Matching Tool



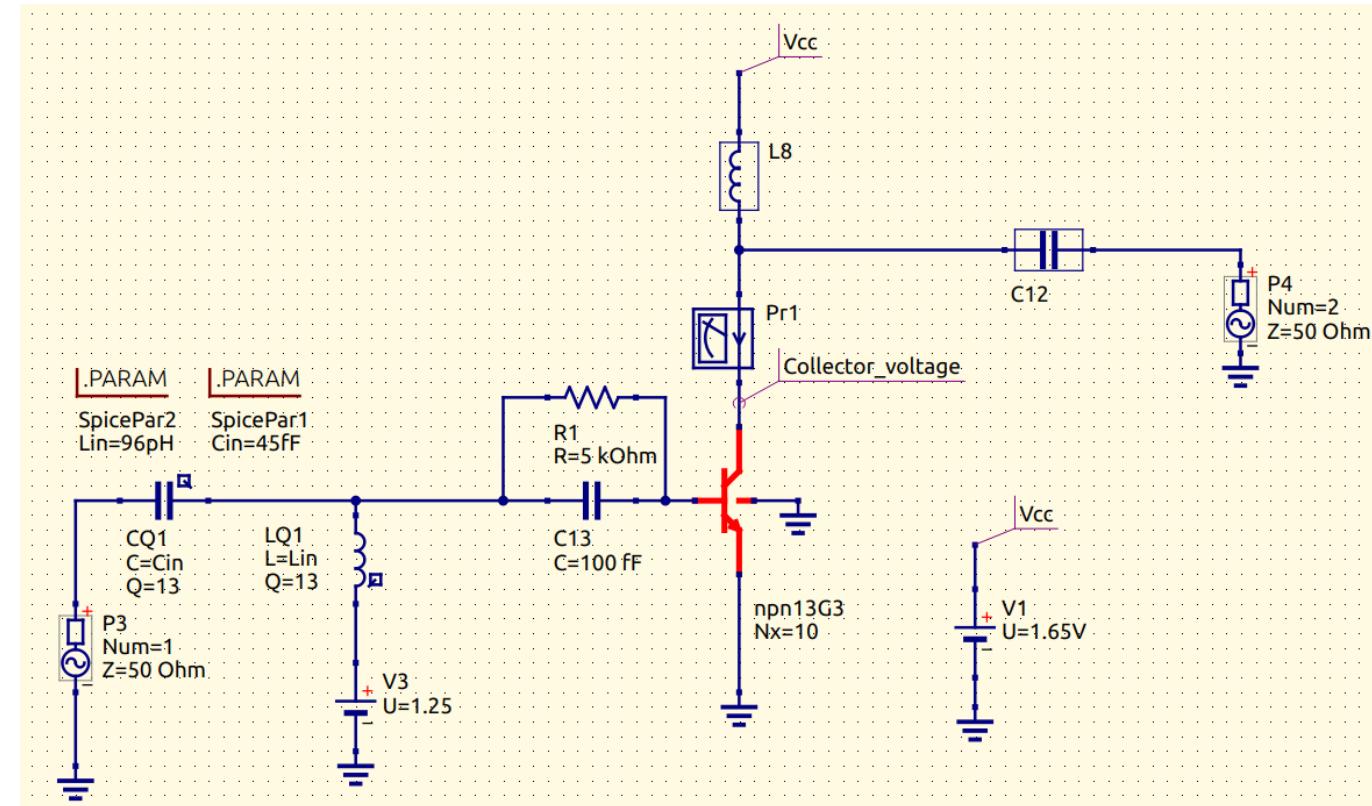
- 0 At this point you can navigate to the following link to match your input section:  
[https://www.will-kelsey.com.smith\\_chart/](https://www.will-kelsey.com.smith_chart/)

- 0 You can do the matching in QUCS-S using the matching tool under tools, but not for this kind of section that we wish to create
- 0 When an appropriate matching have been found you can refer to the next slide to insert the components and tune the parameters

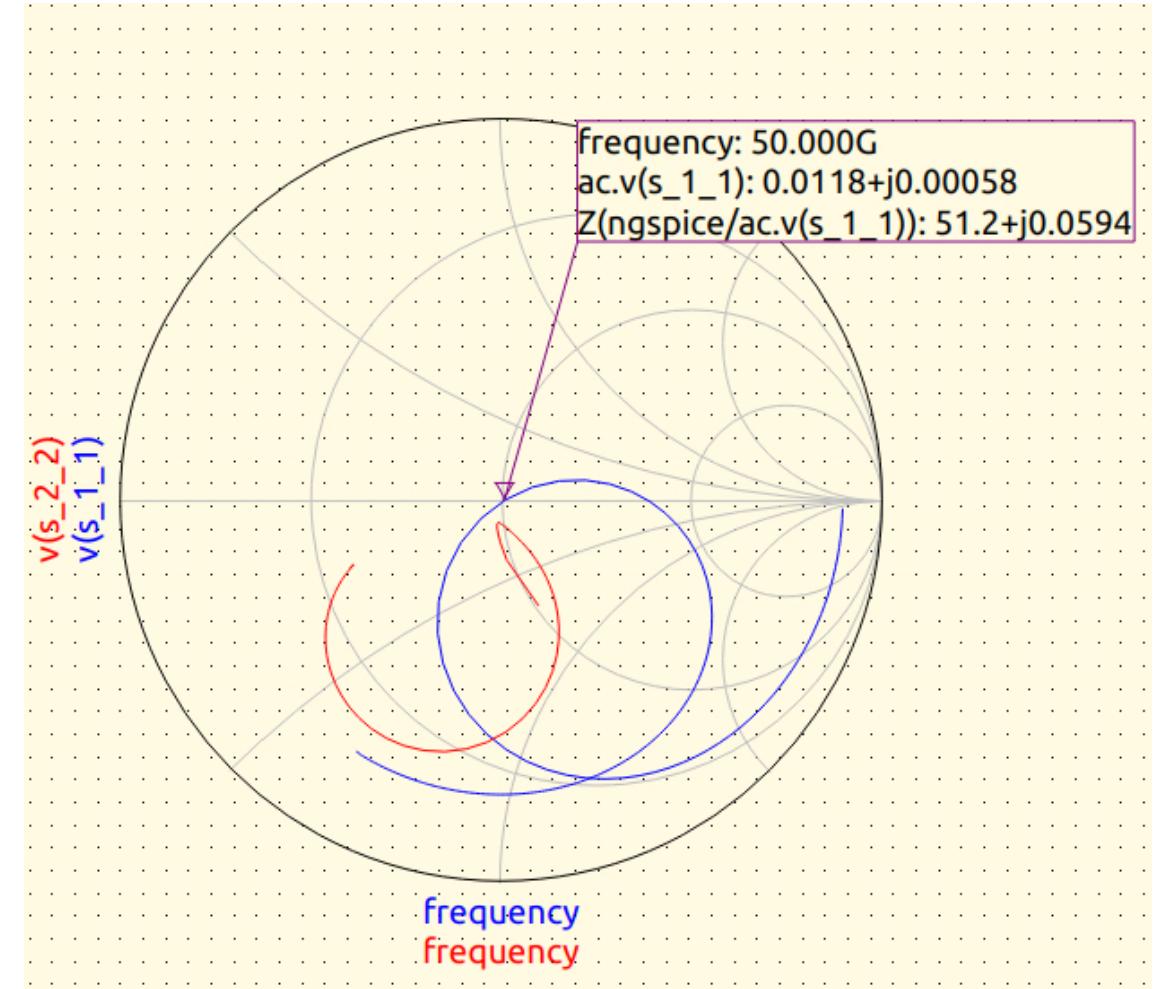
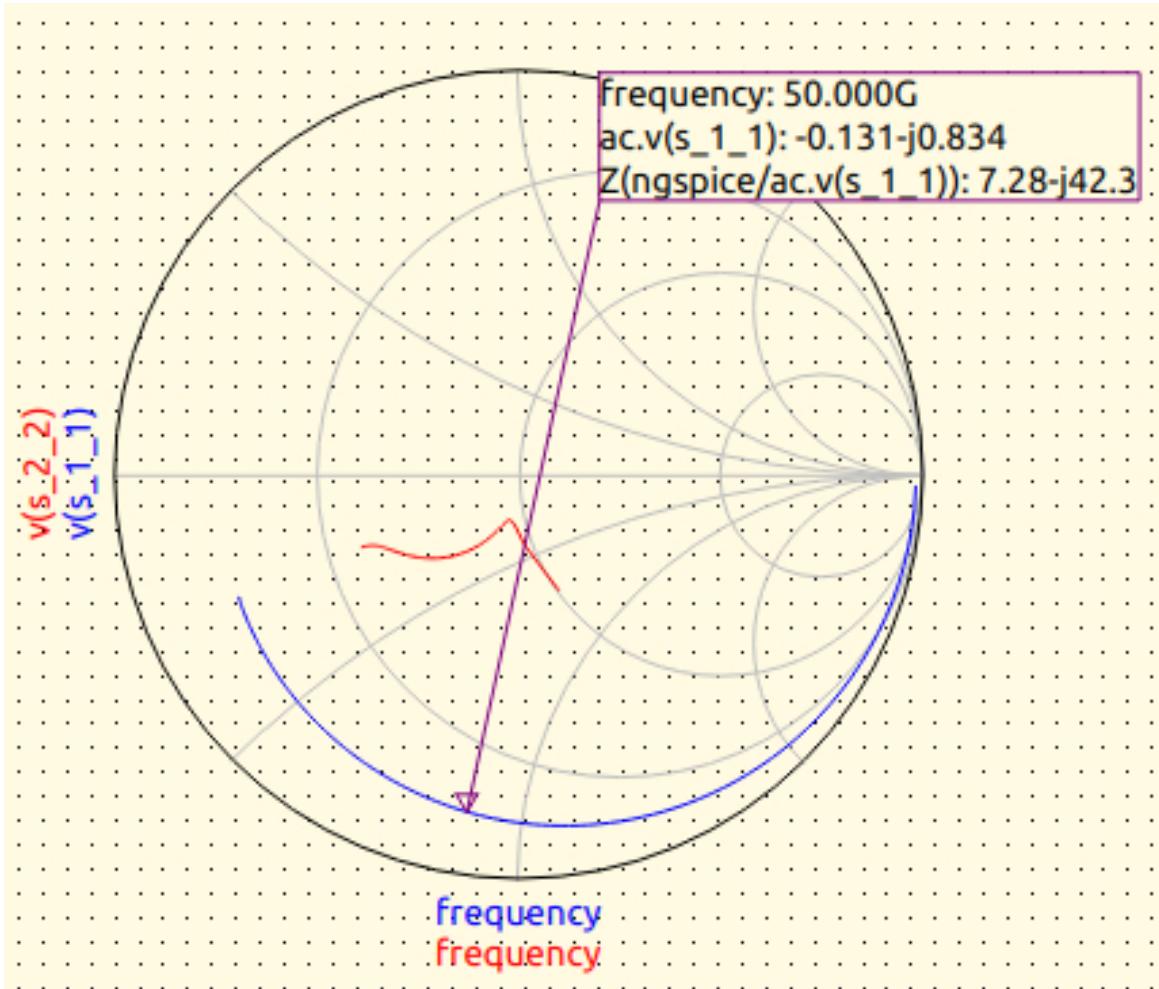


# Matching

- 0 We will start by inserting “real” capacitor and inductor in the input and (remember to set frequency of Q-factor components)
- 0 Since we define variables for the capacitance and inductance we also need to include **.param** blocks for this
- 0 This will be used to tune the input sizes to get best input matching
- 0 C and L values are tuned based on parameters from the previous slide—yours may vary 😊



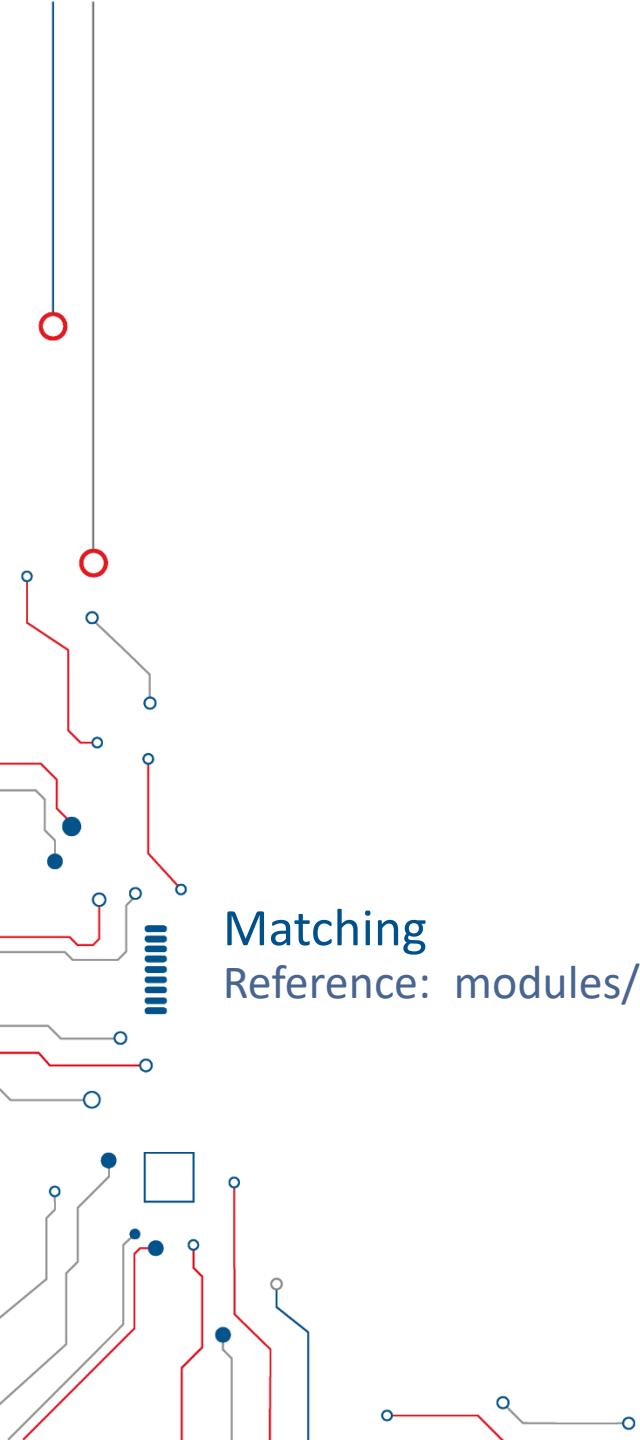
# Matching



# Part 3

## Matching

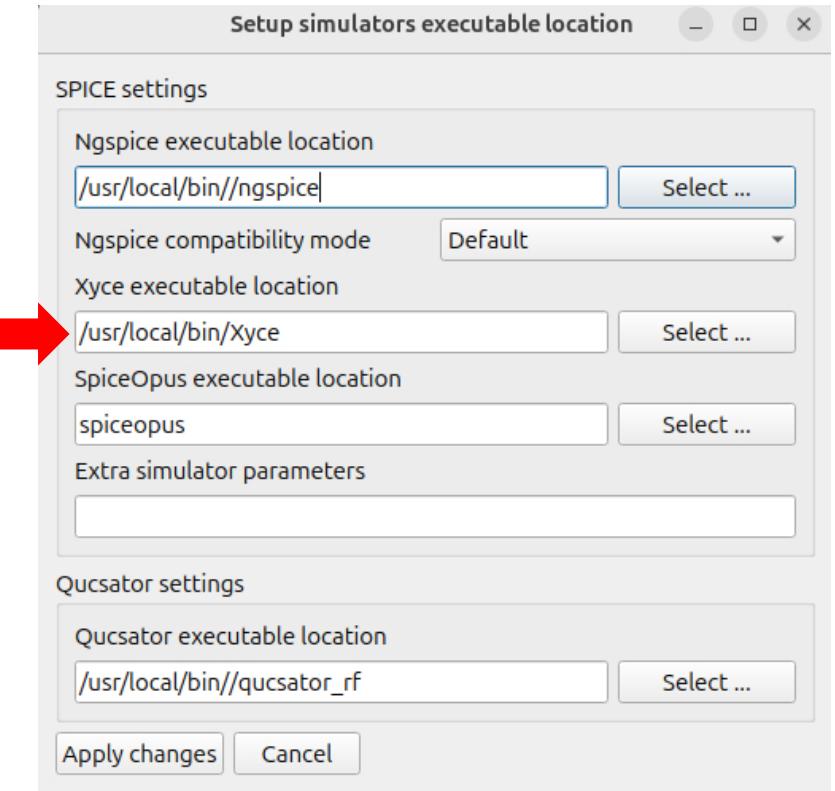
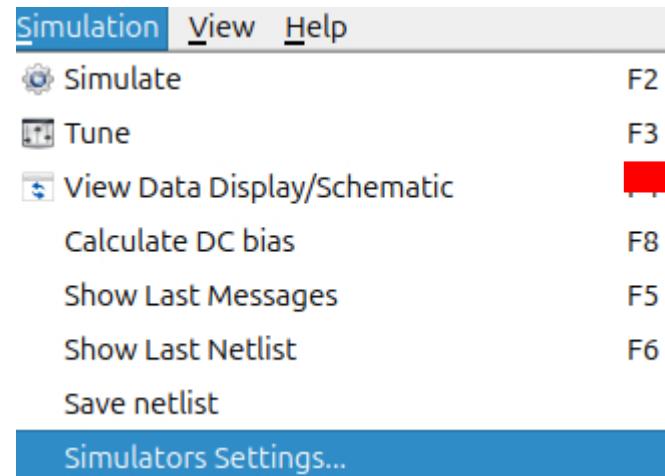
Reference: [modules/module\\_2\\_50GHz\\_MPA/part\\_3\\_nonlinear\\_analysis](#)



# Xyce Setup



-0 Make sure the path to the simulator is set correctly!



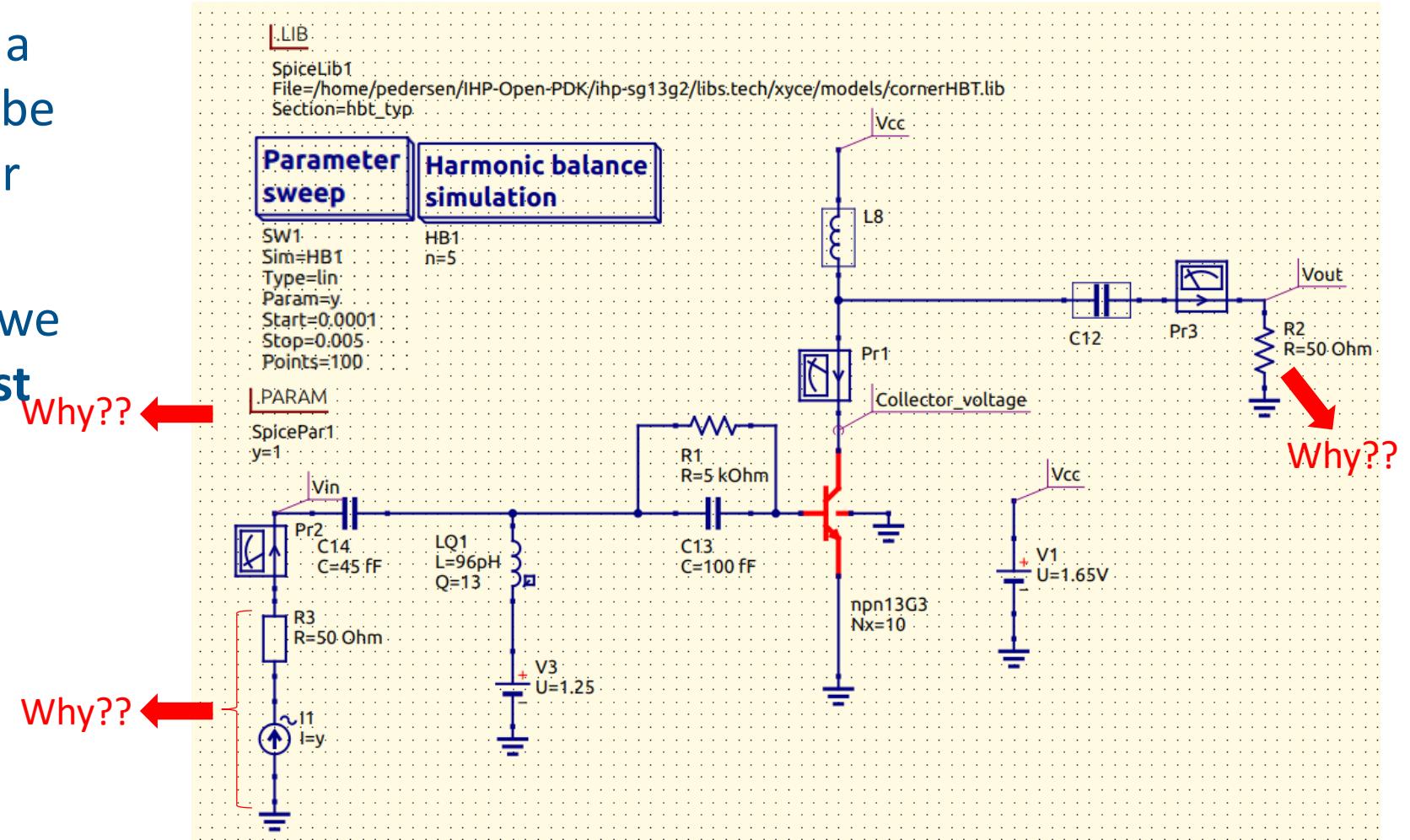
# Nonlinear Analysis



- 0 We switch simulator to Xyce
- 0 This imposes some issues!!
- 0 Math calculations for plotting is not possible!
- 0 Sweeping is somehow challenging
- 0 Pushing to hard into the Nonlinear Domain corrupts the data from xyce
- 0 Lets look at the schematic setup on the next slide!

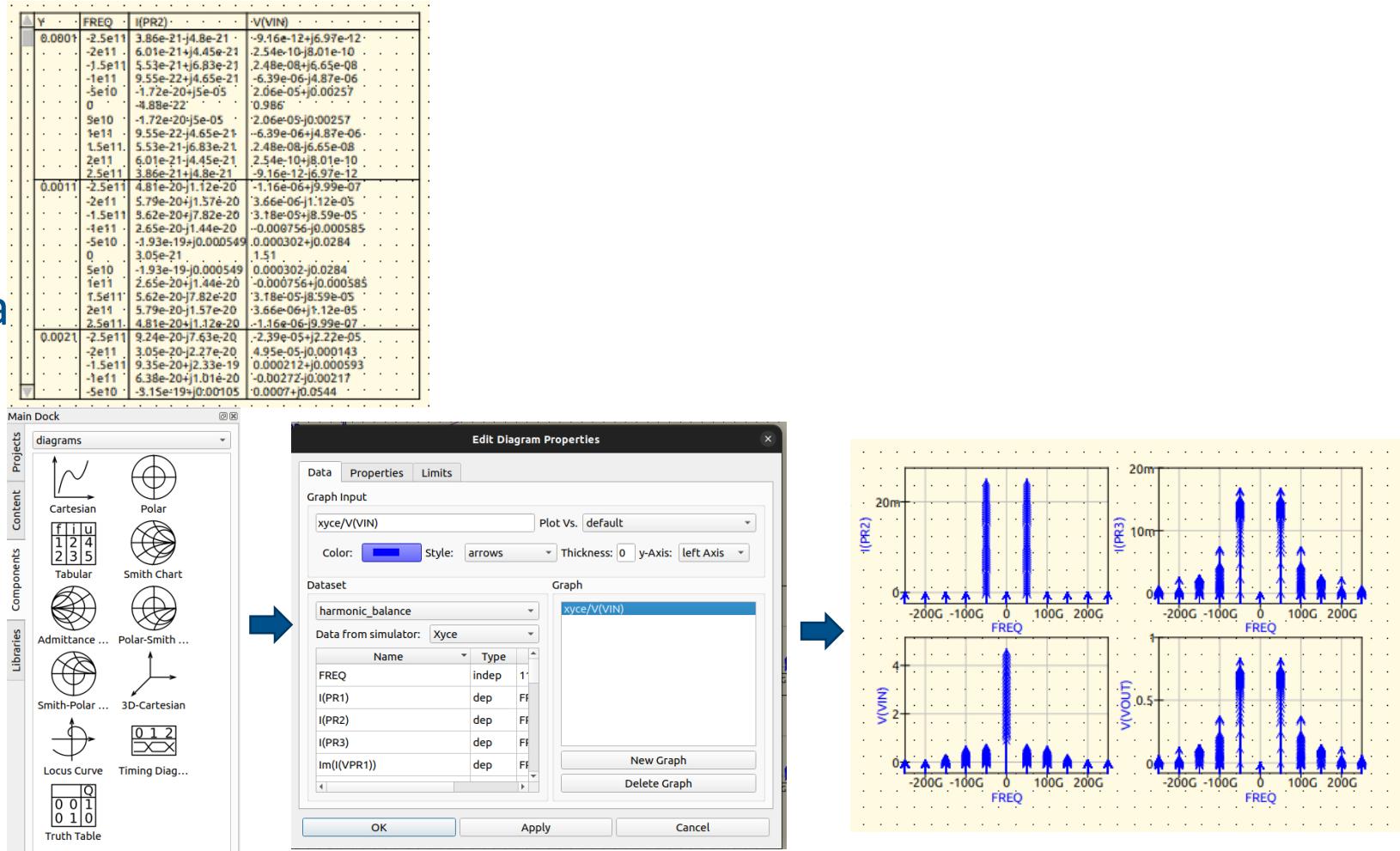
# Nonlinear Analysis (Compression Point)

- 0 We node we have to include a parameter sweep which can be found in the main dock under simulation
- 0 Also for the model inclusion we have to go under **SPICE netlist section** in the main dock and choose the **.Lib directive**
- 0 Remember to switch the simulator to Xyce!
- 0 Run the Simulation



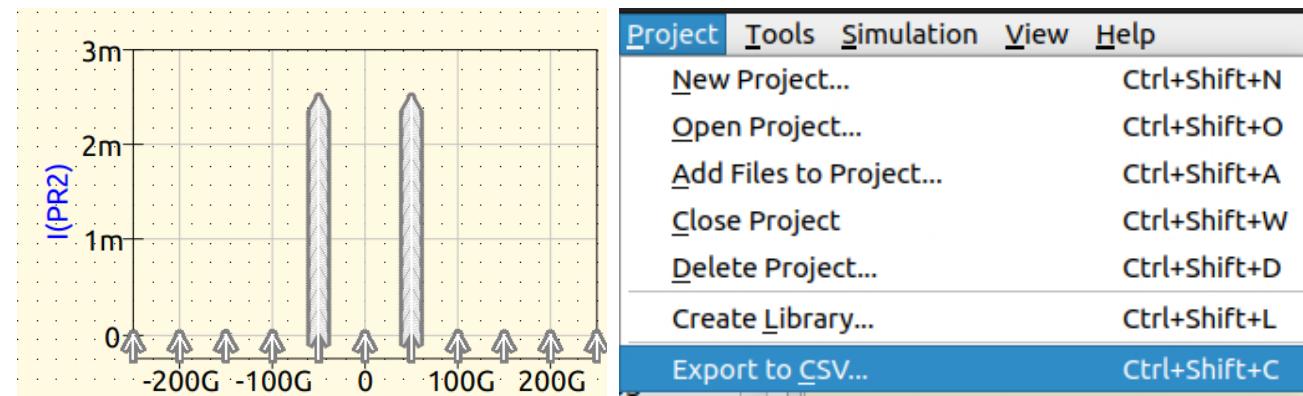
# Visualizing The Results

- 0 After setting up the simulation, we can run it and check the results. Upon completion, the first task is to **instance a table** to inspect the output data and ensure proper formatting.
- 0 To extract the data for post-processing we need to plot data, which we will do with a **Cartesian** plot seen on the following slide



# Visualizing The Results

- 0 Mark the data on the graphs and navigate to the following
- 0 Once the all the data for the parameters have been saved, the Jupyter-lab file at the following path can be opened: *module\_2\_50GHz\_MPA/part\_3\_nonlinear\_analysis/python jupyter lab post\_processing.ipynb*



Name
Vout.csv
Vin.csv
IPR3.CSV
IPR2.CSV

# Visualizing The Results

- 0 Demonstrates how to plot extracted data using Python
- 0 Ensure parameter names match those specified for successful extraction
- 0 If schematic labels differ from the slides, open the CSV files to check and adjust the parameter names

```
[1]: # Import libraries
import pandas as pd
pd.set_option('display.float_format', lambda x: '%.22f' % x)
import numpy as np
import matplotlib.pyplot as plt
import os
import scienceplots
plt.style.use(['science', 'ieee'])
# creates figs directory
output_dir = 'figs'
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

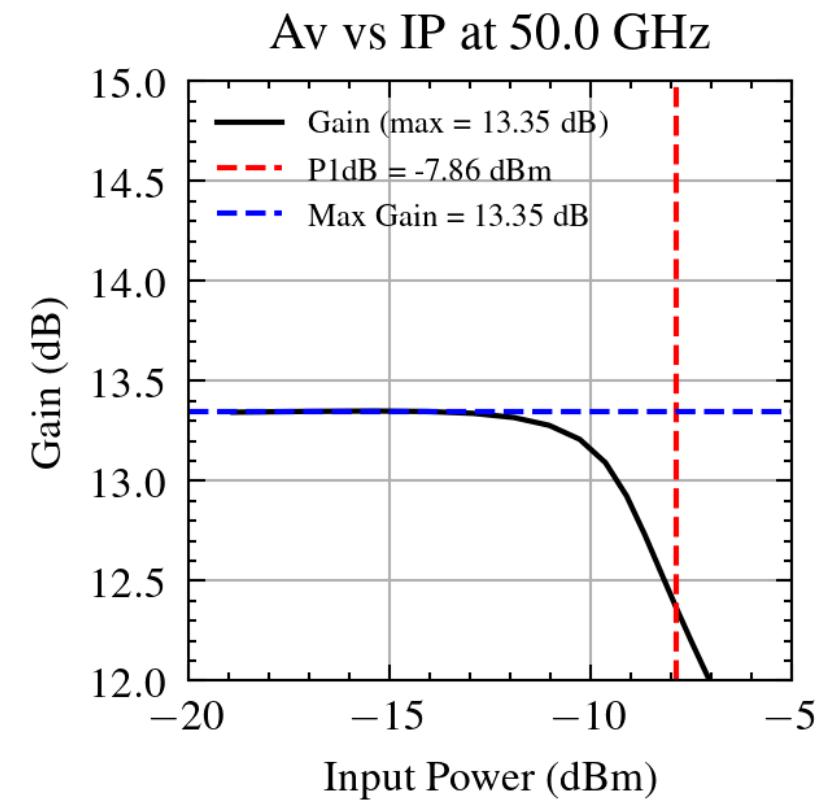
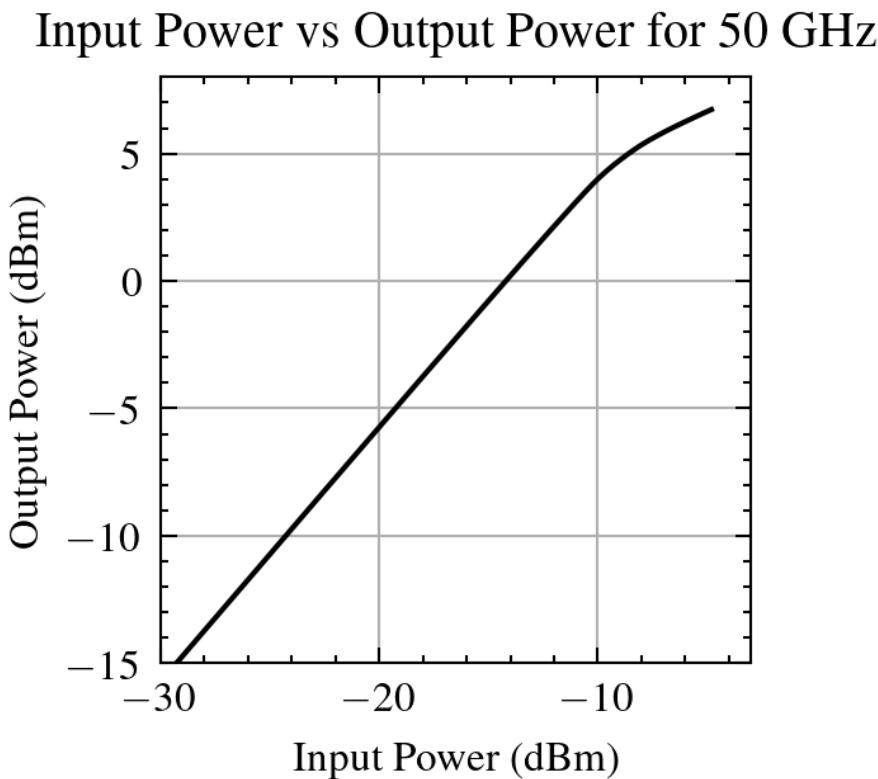
[2]: # Read CSV data (change paths accordingly)
Iin_csv = pd.read_csv('../schematic/compression_1/csv/IPR2.csv', delimiter=';', comment='#')
Iout_csv = pd.read_csv('../schematic/compression_1/csv/IPR3.csv', delimiter=';', comment='#')
Vin_csv = pd.read_csv('../schematic/compression_1/csv/Vin.csv', delimiter=';', comment='#')
Vout_csv = pd.read_csv('../schematic/compression_1/csv/Vout.csv', delimiter=';', comment='#')

# Initialize an empty DataFrame with the correct number of rows based on Iin_csv
power_df = pd.DataFrame({
    'Freq': Iin_csv['FREQ'],
    'P_in': np.zeros(len(Iin_csv)), # Preallocate with zeros
    'P_out': np.zeros(len(Iin_csv)) # Preallocate with zeros
})

# Create complex voltage and current arrays
vin_complex = []
vout_complex = []
Iout_complex = []
Iin_complex = []

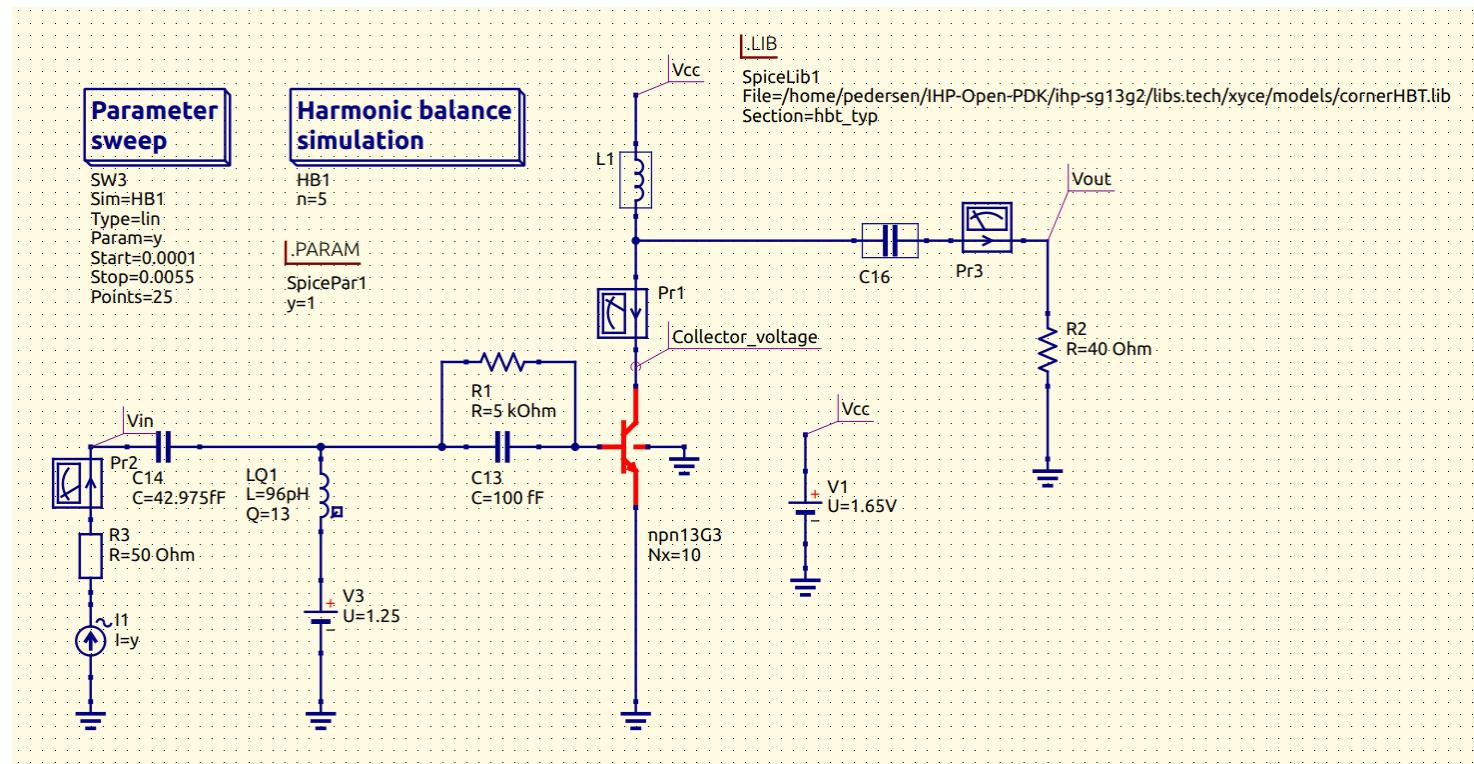
# Populate the complex lists
for i in range(len(Vin_csv['r_xyce/V(VIN)'])):
    vin_complex.append(complex(Vin_csv['r_xyce/V(VIN)'][i], Vin_csv['i_xyce/V(VIN)'][i]))
    vout_complex.append(complex(Vout_csv['r_xyce/V(VOUT)'][i], Vout_csv['i_xyce/V(VOUT)'][i]))
    Iout_complex.append(complex(Iout_csv['r_xyce/I(PR3)'][i], Iout_csv['i_xyce/I(PR3)'][i]))
    Iin_complex.append(complex(Iin_csv['r_xyce/I(PR2)'][i], Iin_csv['i_xyce/I(PR2)'][i]))
```

# Visualizing The Results



# Load Pull

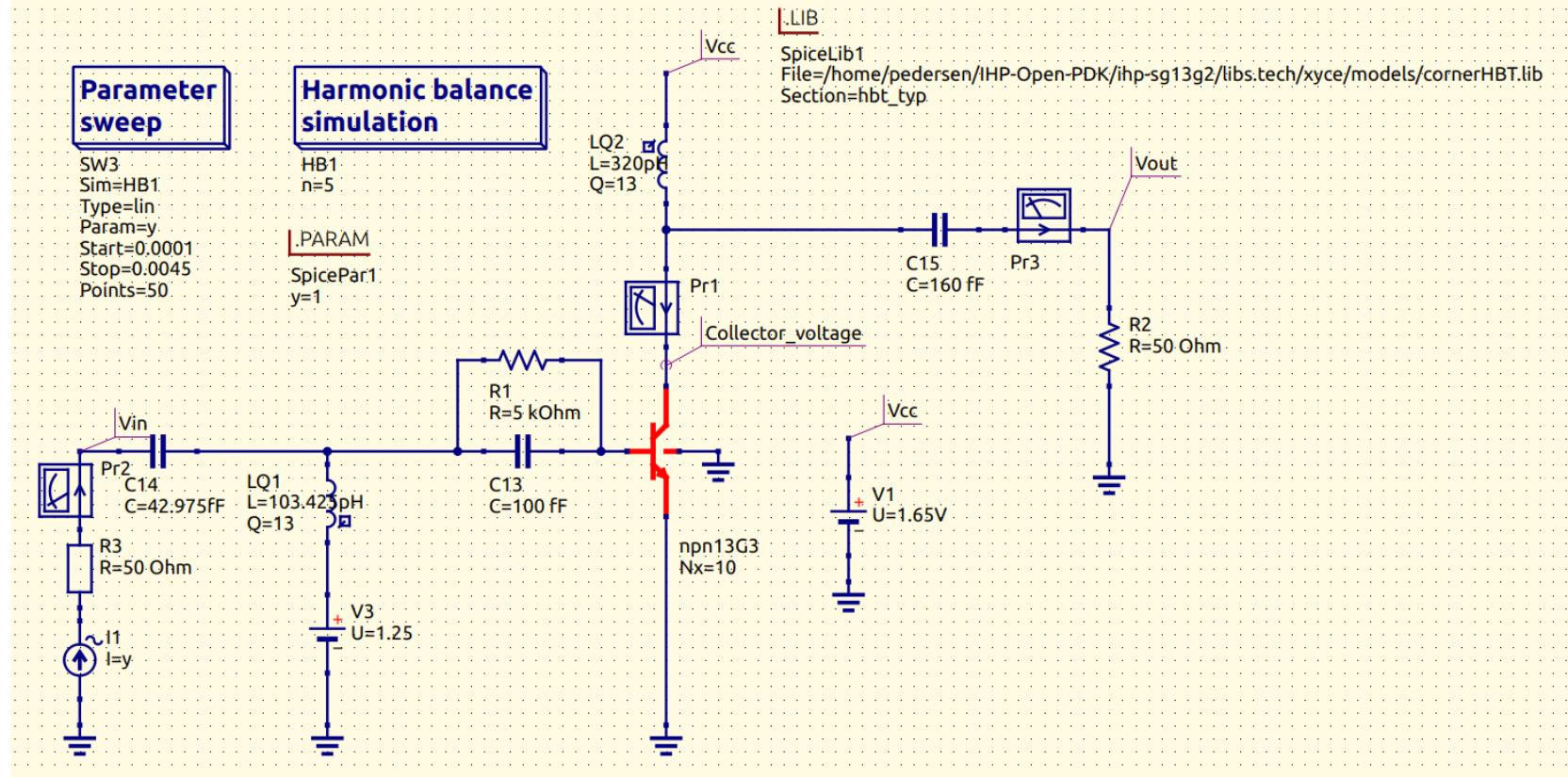
- 0 The goal of the load-pull analysis is to optimize the **output network** for maximum linearity
- 0 Explored a straightforward approach for multi-parameter sweeps in Xyce, but faced challenges due to limited documentation and unexpected complexities.
- 0 Current simple approach: vary resistance and evaluate performance around 50 Ω.



# Load Pull



-0 The schematic to the right is the one we will continue with..





Leibniz Institute  
for high  
performance  
microelectronics

# Coffee Break!! / Catching Up

# Part 4

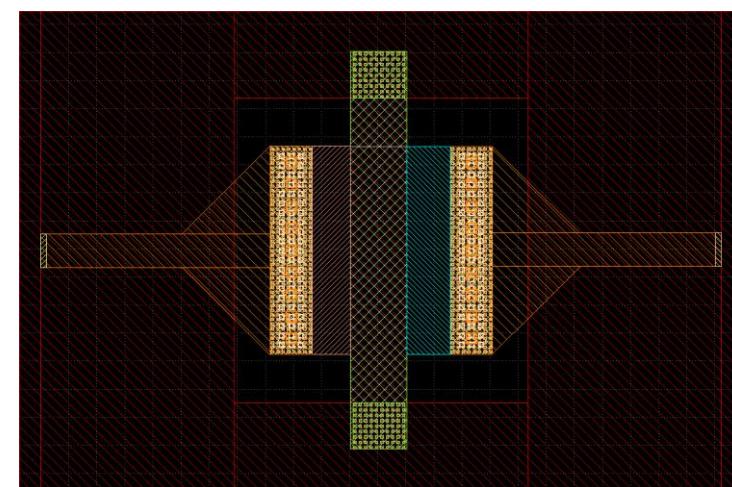
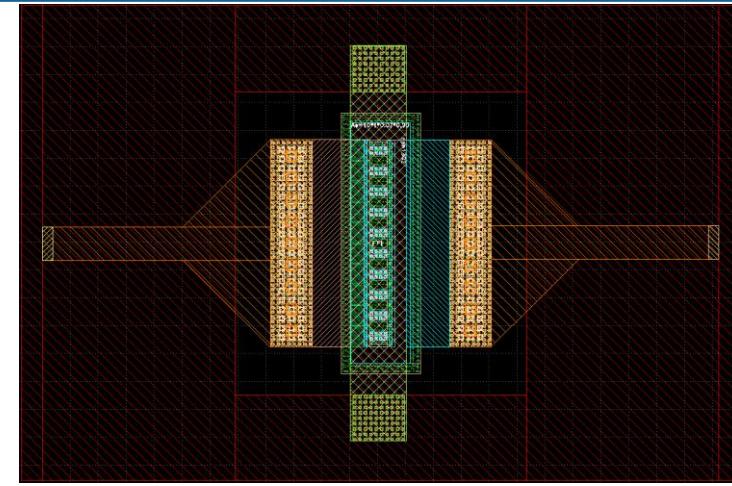
OpenEMS

Reference: [modules/module\\_2\\_50GHz\\_MPA/part\\_4\\_layout\\_EMsims](#)

# EM Simulation



- 0 At this point we want to include the EM simulated components into QUCS-S for post processing!
- 0 We will use the approach proposed by **Dr.-Ing Volker Mühlhaus**



# EM Simulation

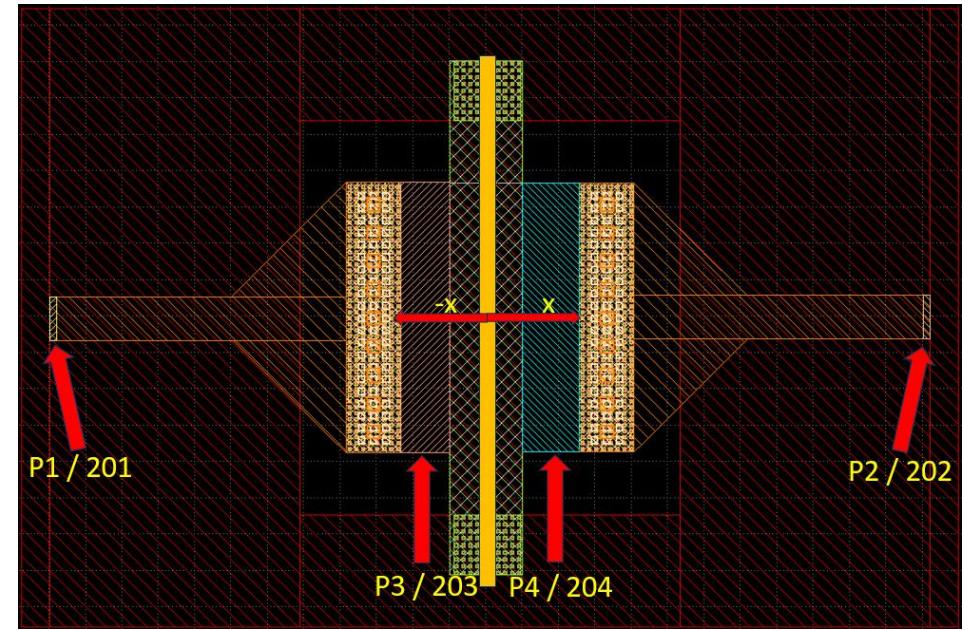


- 0 Define Input Ports on following layers
- 0 The input ports should be referenced as described in the earlier presentation. For a detailed walkthrough of the code for this specific structure, refer to the following markdown file:

[module\\_2\\_50GHz\\_MPA/part\\_4\\_layout\\_EMsims/EM\\_simulation.md](#)



201/0
202/0
203/0
204/0



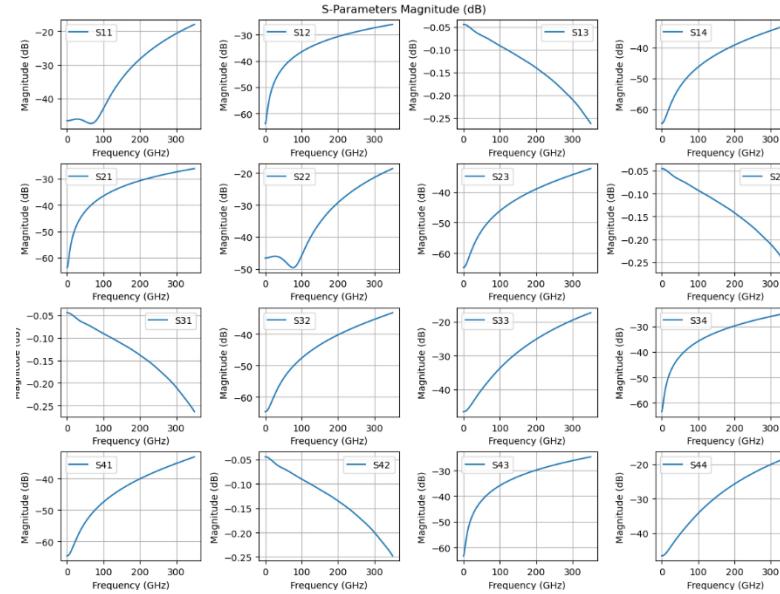
# Post EM-Simulations

## -0 Viewing the S-Parameters:

`openems/output/run_core_50ghz_mpa_data/run_core_50ghz_mpa_data`

`openems /output/run_core_50ghz_mpa_data/spar_plot.py` (this can be used for plotting the S-parameters)

```
python3 spar_plot.py your_spar_file.s4p
```



# Post EM-Simulations



## -0 Importing The EM model into QUCS-S:

The screenshot shows the QUCS-S software interface. On the left, the Main Dock displays a library of components under the 'Components' tab. Components shown include X, L, Subcircuit, SPICE library, S, s1p, 1-port S par..., SPICE netlist, s2p, 2-port S par..., n-ports S par..., Z, A, SPICE gener..., and XSPICE gener... . A red arrow points from the 'n-ports S par...' component towards the 'Edit Component Properties' dialog on the right.

**Edit Component Properties**

S parameter file

Name: X1  display in schematic

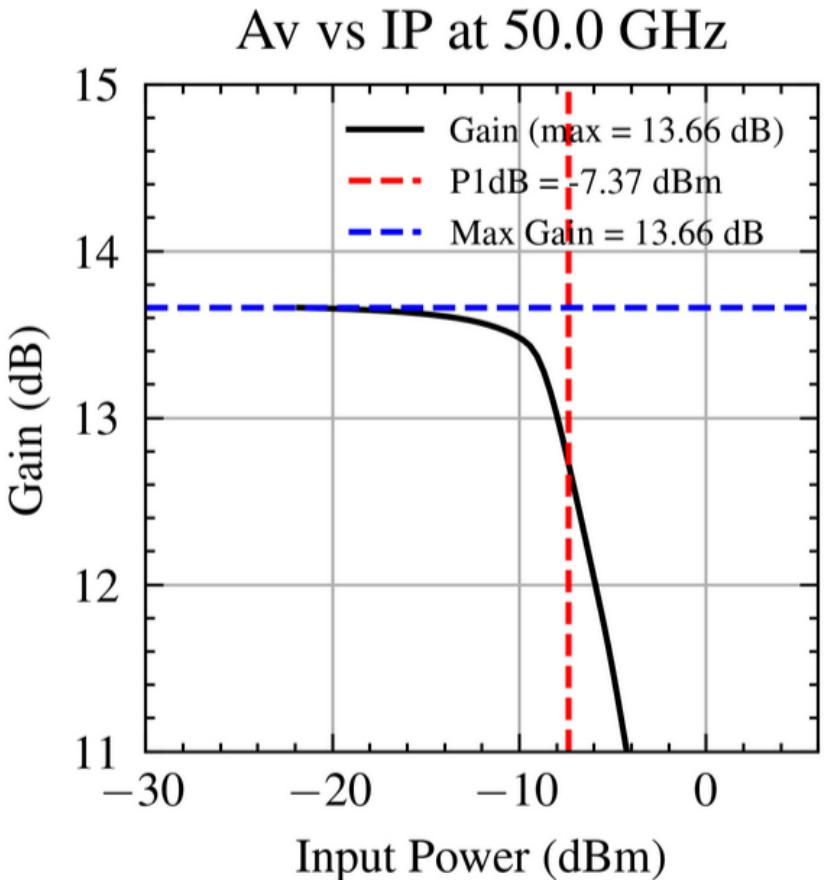
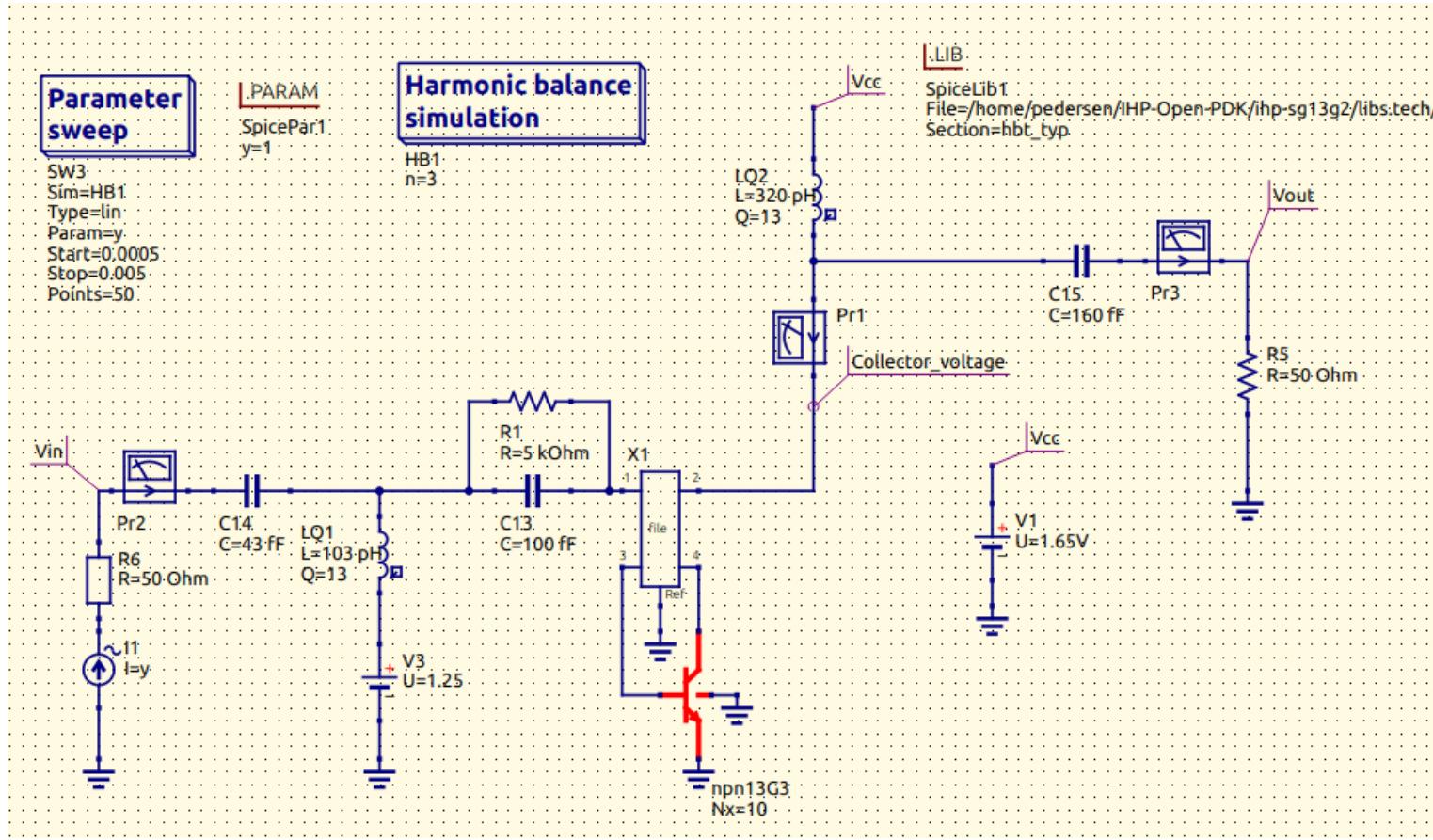
Properties

Name	Value
File	..../openems/output/run_core_50ghz_mpa_data/run_core_50ghz_mpa.s4p
Data	rectangular
Interpolator	linear
duringDC	open
Ports	4

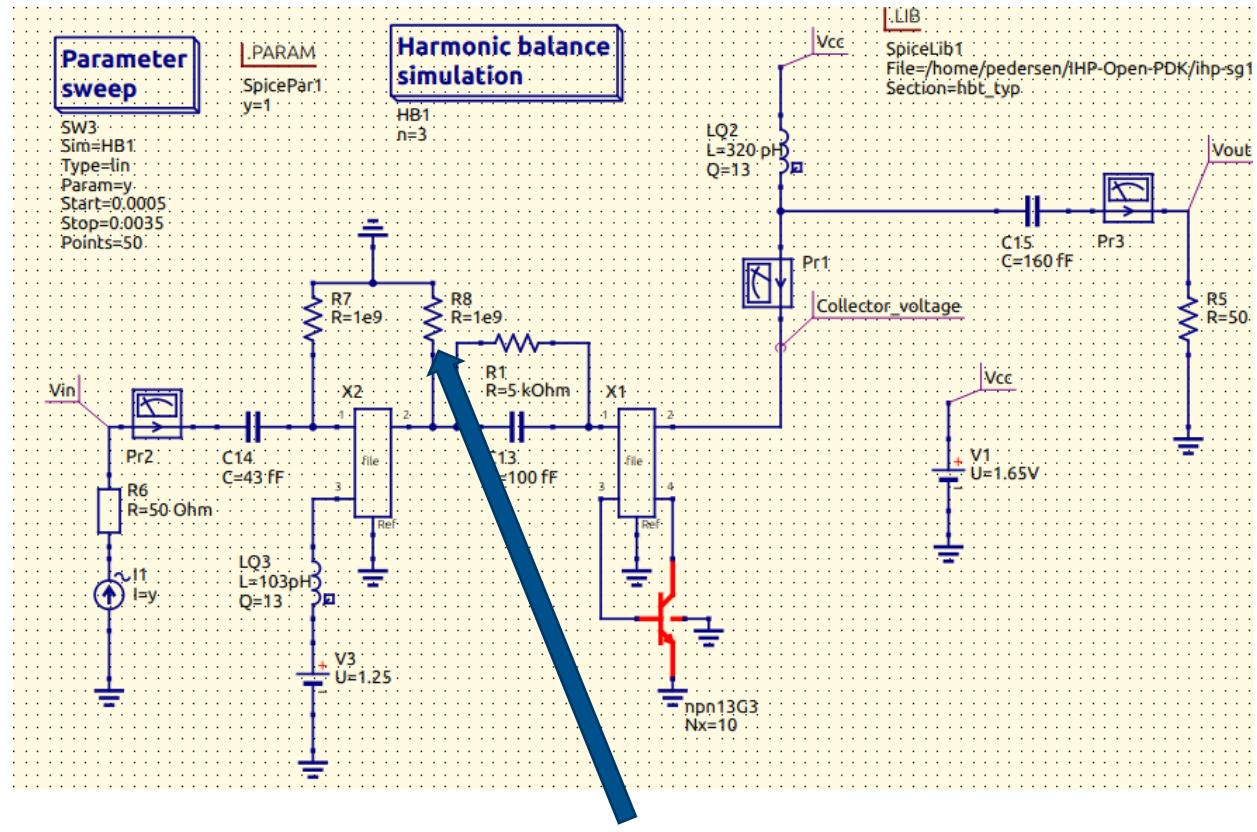
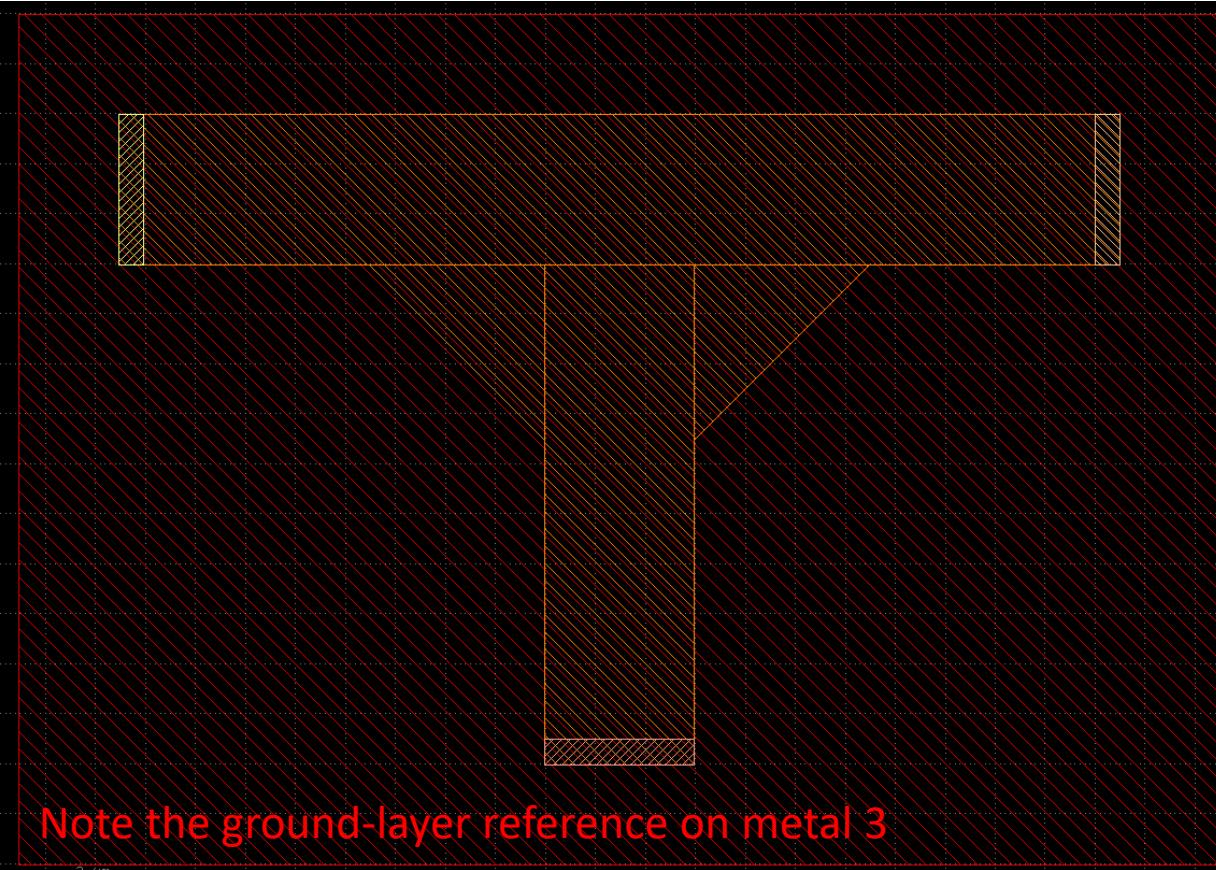
File  
name of the s parameter file  
a/run\_core\_50ghz\_mpa.s4p  
Edit Browse ←

Add Remove  
Move Up Move Down  
Fill from SPICE .MODEL  
OK Apply Cancel

# Post EM-Simulations



# Post EM-Simulations





# Remaining Agenda for Today

- 0 Make additional components to be EM simulated (inductors, T-connections etc.):** Use this change to work on a machine with the full tool flow and get help
- 0 Design Competition:** Work on your layout for the big design competition!!
- 0 Catch-Up:** Work on material from today or the other days to catch up or clarify points that wasn't clear
- 0 Relax ☺**

# What Will We Do Tomorrow?



- 0 **Xschem Practice:** Create small building blocks for a larger circuit and perform analyses such as Monte Carlo simulations based on your prior experience
- 0 **Verilog:** Write and simulate simple Verilog code using open-source tools.
- 0 **Analog-Mixed-Signal Integration:** Learn to integrate digital blocks into analog designs within Xschem.
- 0 **8-bit SAR ADC Simulation:** Build and simulate a basic 8-bit successive approximation register (SAR) ADC using the day's designs.
- 0 **Post-Processing with Python:** Plot and analyze transient simulation results using Python.