KU LEUVEN

BRUGES
9-12 SEPTEMBER
ESSERC 2024

micas
KU LEUVEN

QUCS-S - a central tool in the openPDK IC design flow

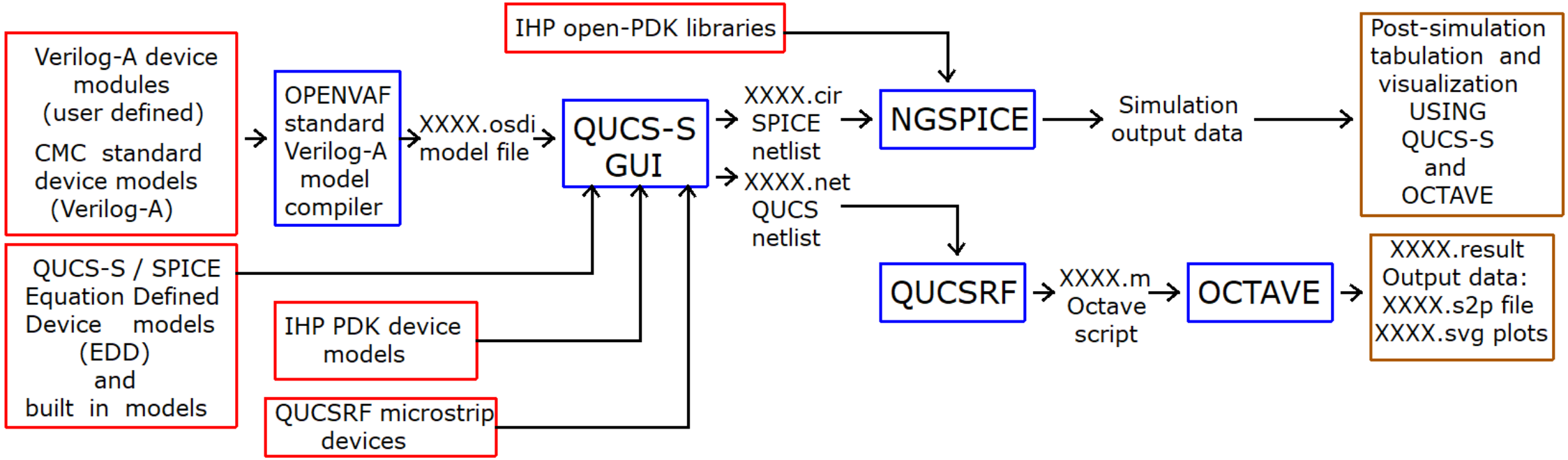
Mike Brinson
Centre for Communications Technology
London Metropolitan University, UK
brinsonm@londonmet.ac.uk

IEEE SSCS Electron Devices Society CAS

Outline

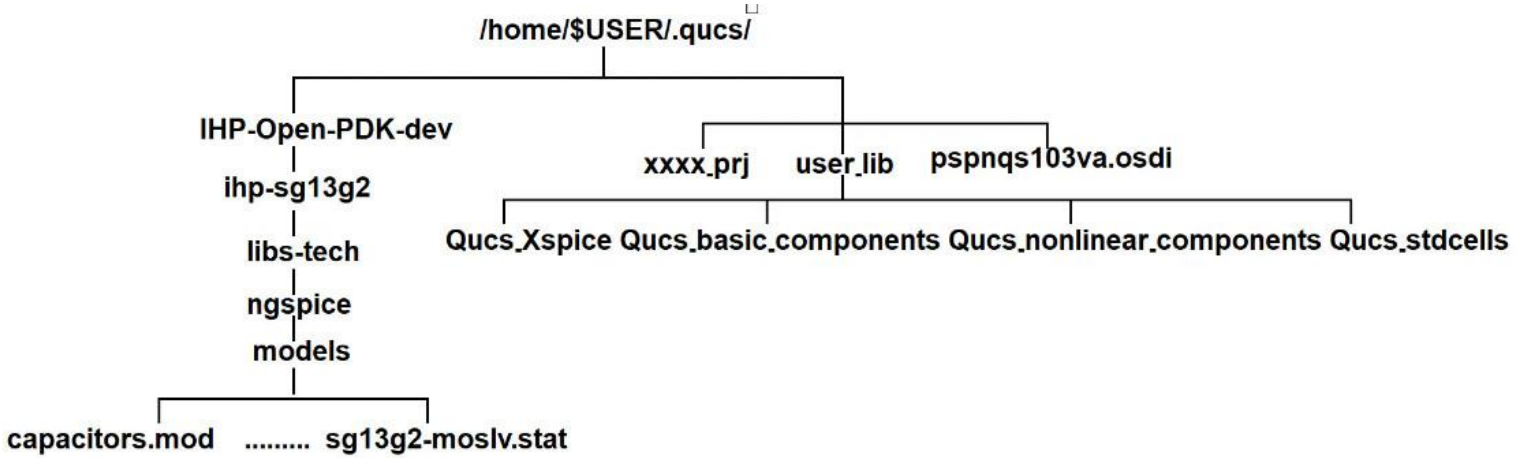
- Qucs-S structure and features
- Qucs-S/IHP Open-PDK file layout
- Example model symbols
- RC, diode and MOS performance test benches
- MOS corner simulation
- DC and AC Monte Carlo analysis
- Analog and digital simulation examples
- XSPICE digital simulation
- QucsRF Octave/openEMF microstrip simulation
- Summary
- Acknowledgments and FOSS links

Qucs-S structure and features



- Red boxes: device and open source PDK data
- Blue boxes: Free and open source circuit simulation software (FOSS)
- Brown boxes: Simulation output data

Qucs-S/IHP Open-PDK file layout

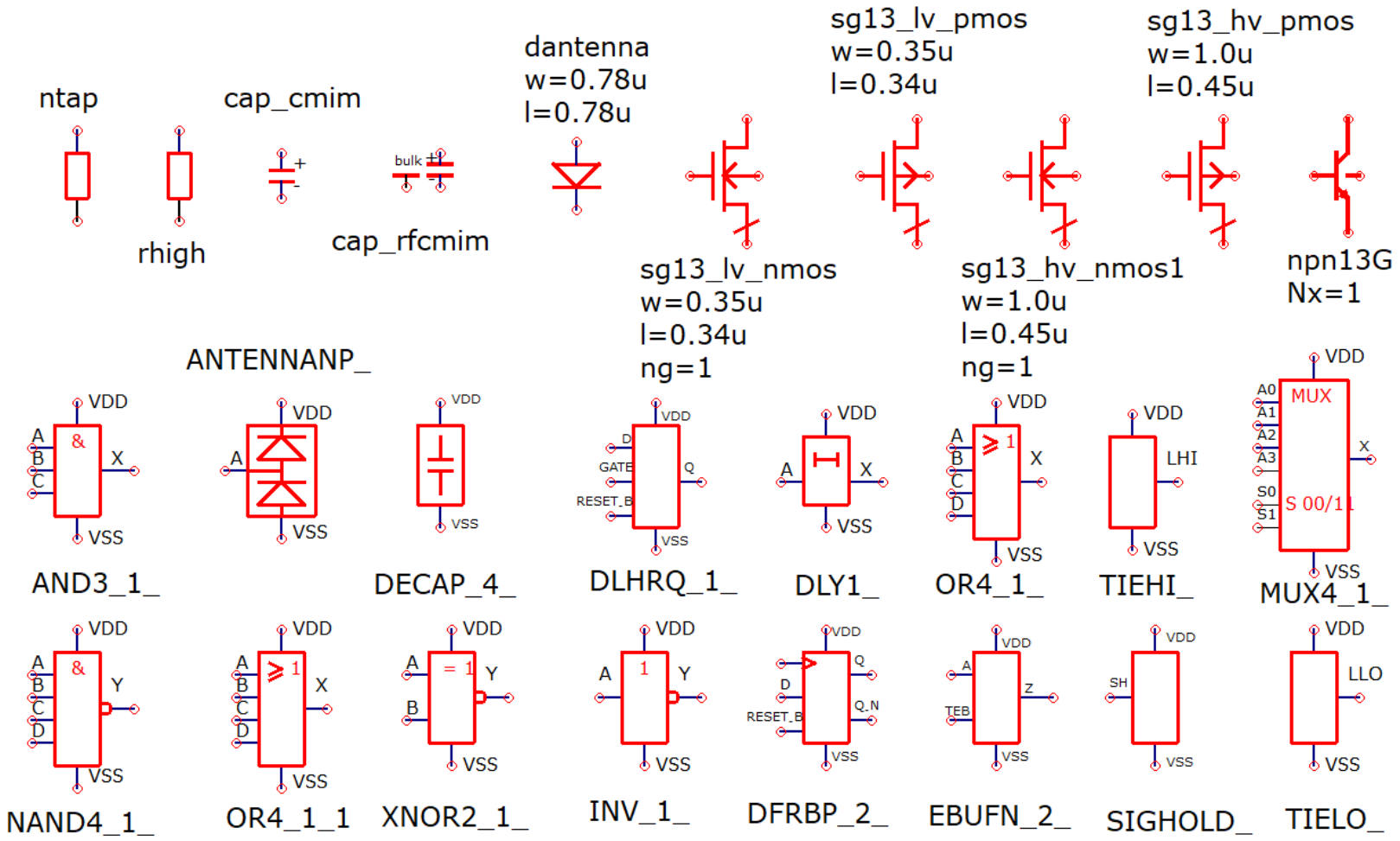


.spiceinit

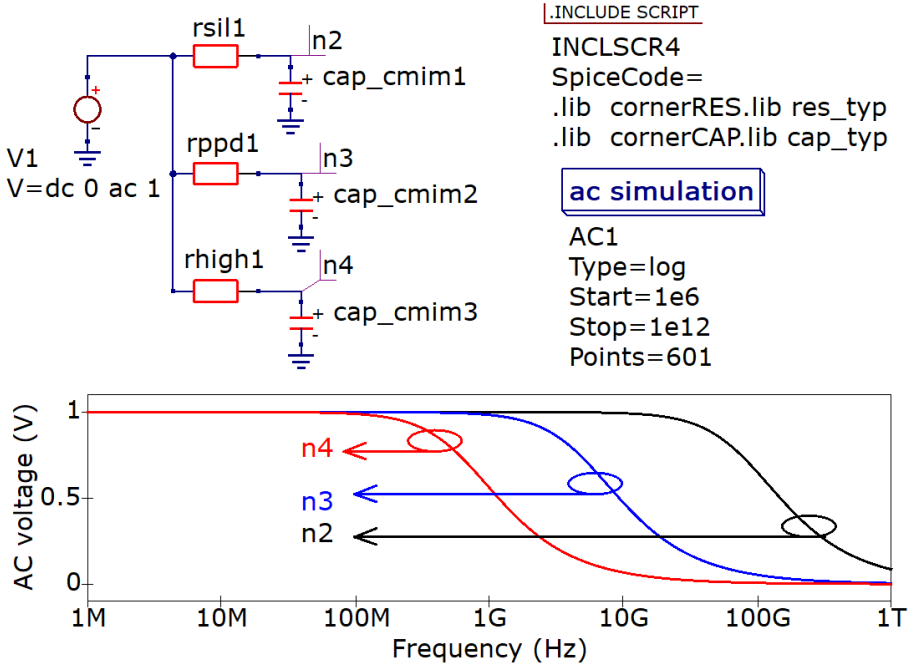
```
*
* Qucs-S/Ngspice global initialization file.
* This file must be copied to directory /home/$USER.
* It is read after Ngspice spinit file and prior to passing an Ngspice netlist.
* NOTE: 1. File ".spiceinit" is an Ngspice user defined iniitialisation file that works with the Linux operating systems.
*       2. Parameter $USER is replaced by the "user name of the computer running Qucs-S.  It is unique for each computer.
*          It is determined by Ngspice during initialisation.
*       3. Using parameter $USER removes the need for absolute addressing.
*       4. The Qucs-S/IHP PDK port assumes that projects and associated data can be found at location /home/$USER/.qucs,
*          for example /home/mike/.qucs where $USER is name mike.
*       5. See Qucs-S  drop down menu "Applications settings" -> Locations -> Qucs Home: /home/mike/.qucs.  The name mike
*          will be different for different computers
*
* Author Mike Brinson: R1 July 2024.
*********************************************************************************
setcs sourcepath = (  $sourcepath /home/$USER/.qucs/IHP-Open-PDK-dev/ihp-sg13g2/libs.tech/ngspice/models )
setcs sourcepath = (  $sourcepath /home/$USER/.qucs/IHP-Open-PDK-dev/ihp-sg13g2/libs.ref/sg13g2_stdcell/spice )
*
*load psp103_nqs.osdi as a global item so that it becomes available to an entire Ngspice netlist.

osdi /home/$USER/.qucs/pspnqs103va.osdi
```

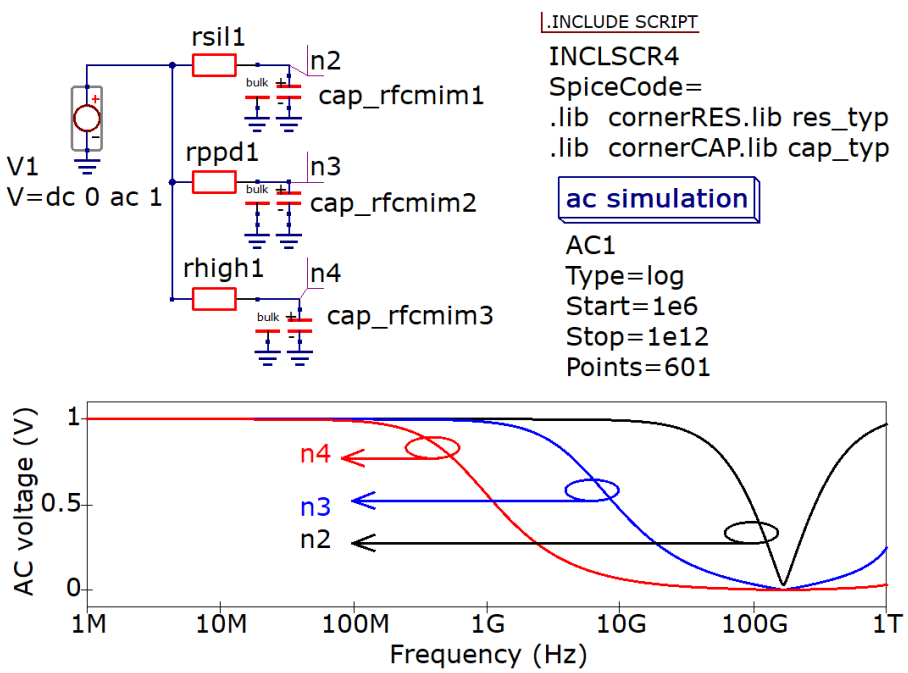
Example model symbols

- Analog devices
- Example digital standard cells



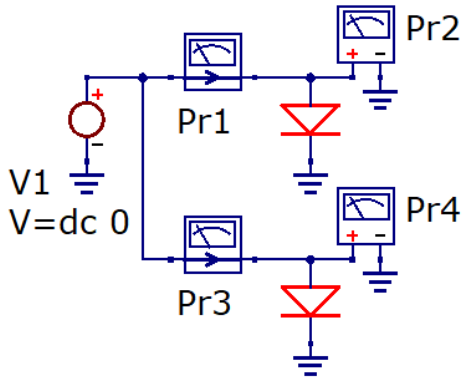
RC circuit simulation





- PDK model symbols colored red
- PDK resistors - name: rxxxx, value: .lib cornerRES.lib res_typ
- PDK capacitors – names: cmim and rfcmim, value: .lib cornerCap.lib cap_typ
- Other model symbols and simulation command ICONS - Qucs-S

Diode model simulation



dantenna1
w=0.78u
l=0.78u

dpantenna1
w=0.78u
l=0.78u

Parameter sweep

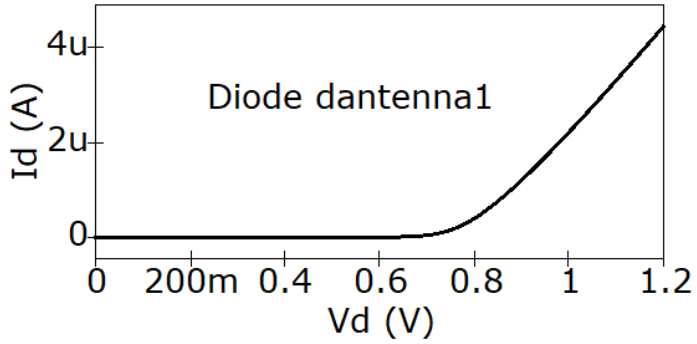
SW1
Sim=DC1
Type=lin
Param=V1
Start=0
Stop=1.2
Points=2001

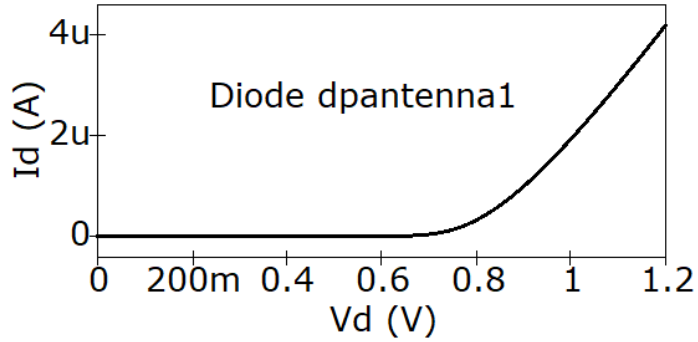
dc simulation

DC1

.INCLUDE SCRIPT

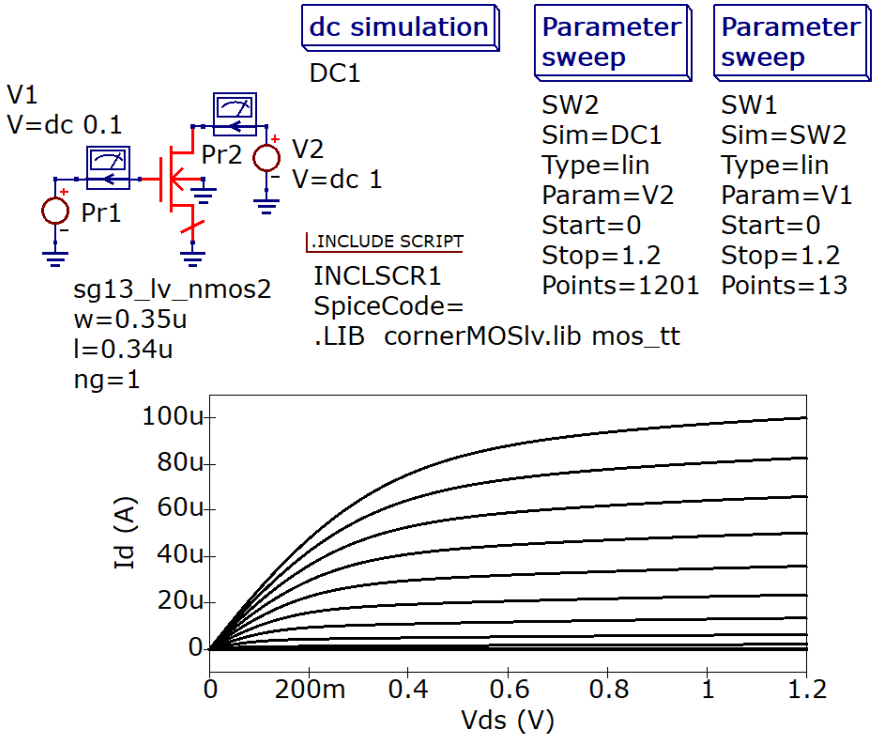
INCLSCR1
SpiceCode=
.INCLUDE diodes.lib

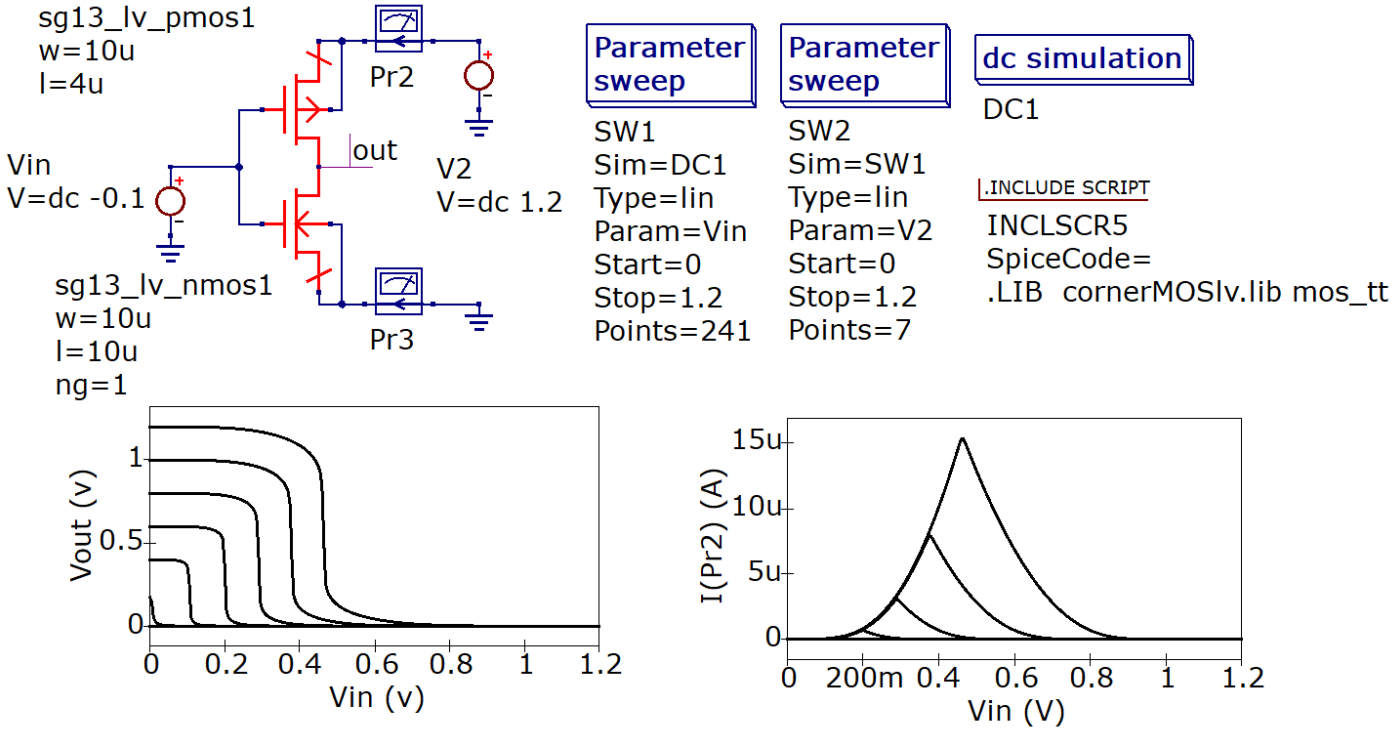




- No “Corner” values for IHP 130nm PDK diodes
- Ngspice .INCLUDE diodes.lib accesses IHP 130nm PDK diode parameters

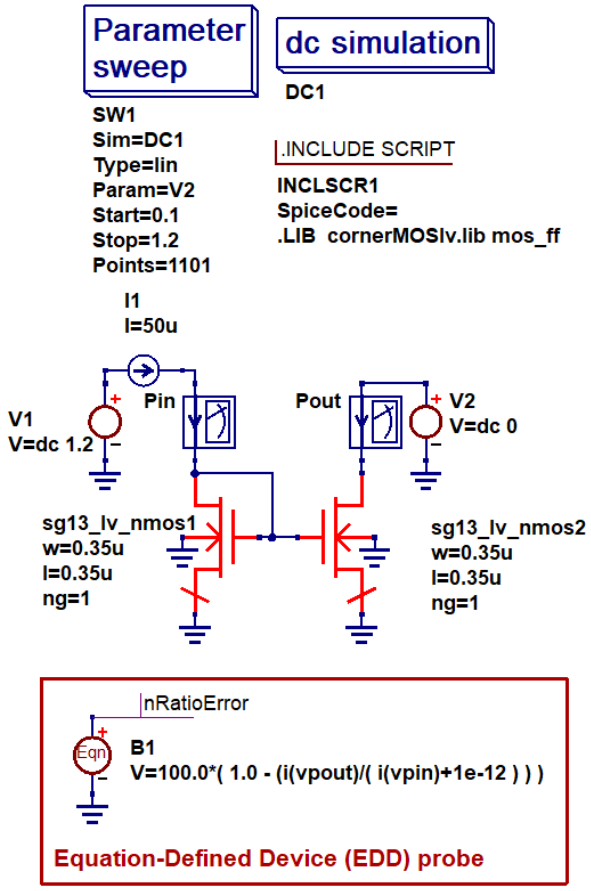
MOS device dc simulation



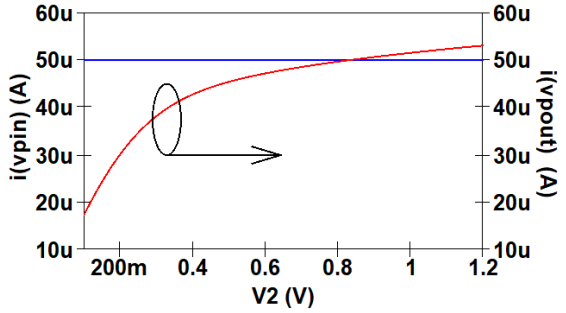


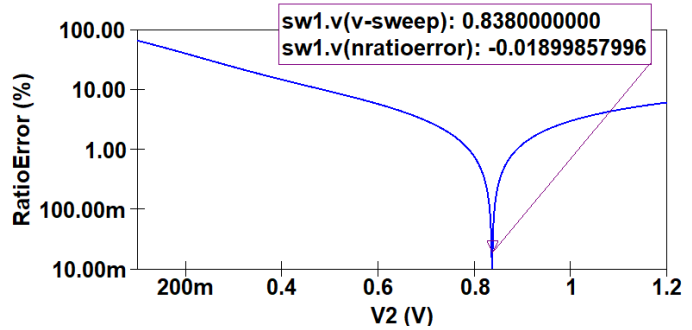
- Left figure: nMOS output characteristic test bench and sg13_lv_mos data
- Right figure: MOS inverter test bench and performance data

MOS corner simulation



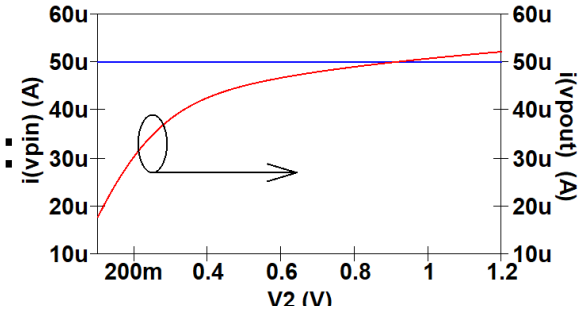
Fast n and p: mos_ff

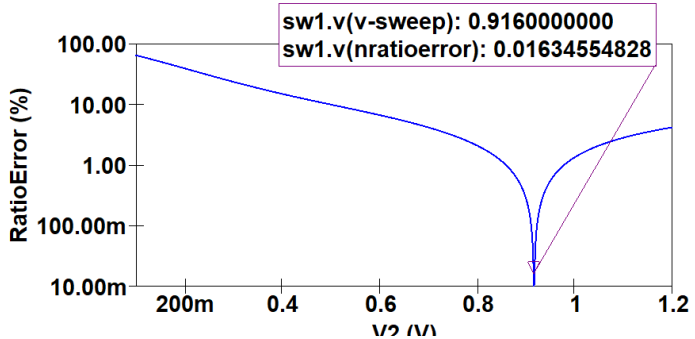




mos_ff

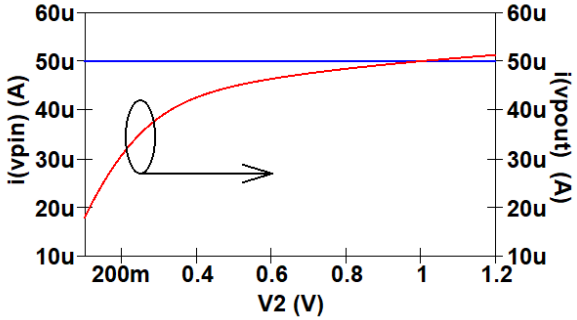
Typical n and p: mos_tt

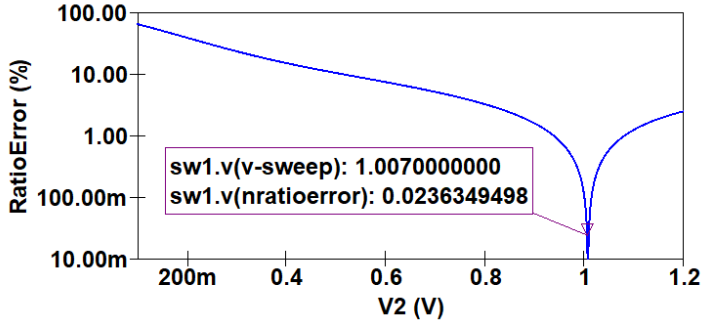




MOS_tt

Slow n and p: mos_ss

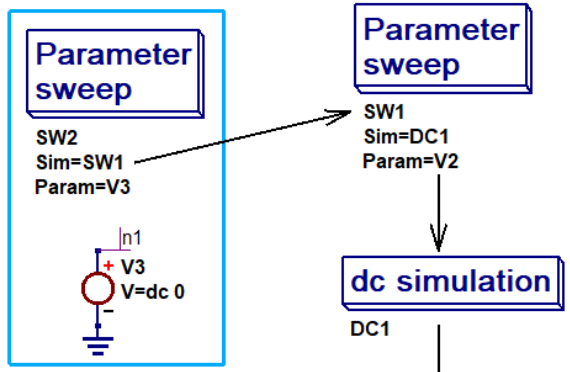




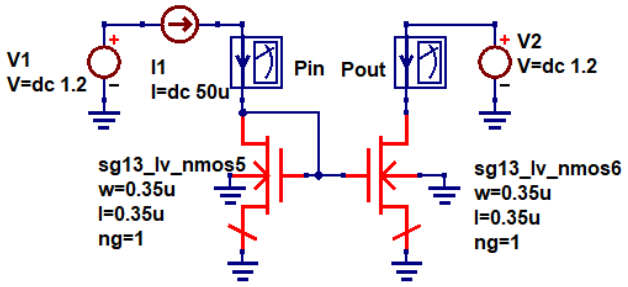
mos_ss

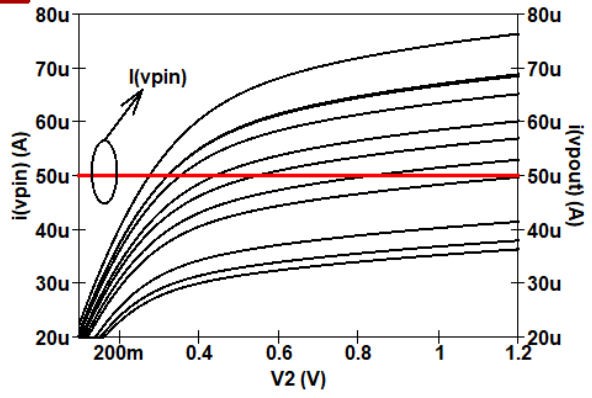
DC Monte Carlo simulation: 1

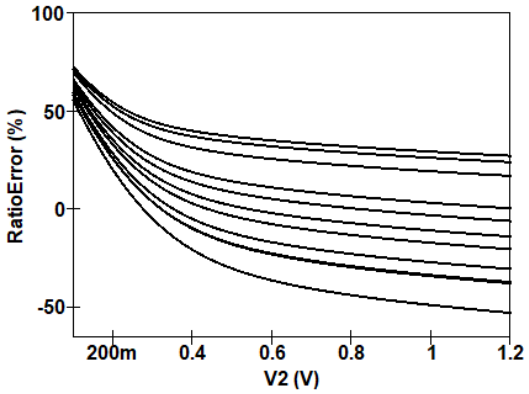
```
.INCLUDE SCRIPT
INCLSCR1
SpiceCode=
.control
let number_v3 = 0
let nruns = 10
echo "STEP sw1.v3" > spice4qucs.sw1.cir.res
dowhile $&number_v3  le  $&nruns
   reset
   alter v3 = $&number_v3
   dc v2 0.1 1.2 0.01
   let RatioError = 100.0*( 1.0 - (i(vpout)/( i(vpin)+1e-12 ) ) )
   write spice4qucs.sw1._swp.plot i(VPin) i(VPout) v(n1) RatioError
   set appendwrite
   echo "$&number_v3 " >> spice4qucs.sw1.cir.res
   let number_v3 =number_v3 + 1
end
unset appendwrite
destroy all
reset
exit
.endc
.end
```




```
.INCLUDE SCRIPT
INCLSCR2
SpiceCode=
.PARAM mc_ok = 1.0
.LIB  cornerMOSlv.lib mos_tt_stat
*
```

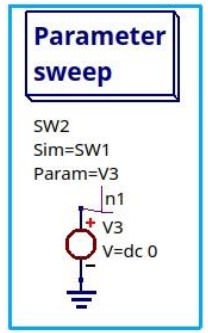


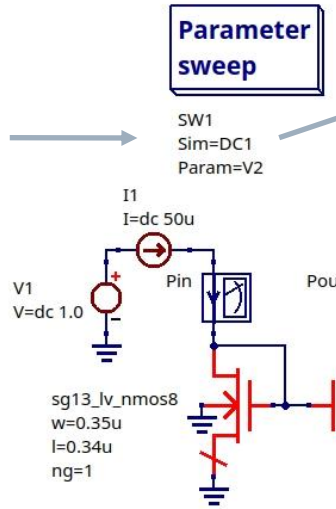


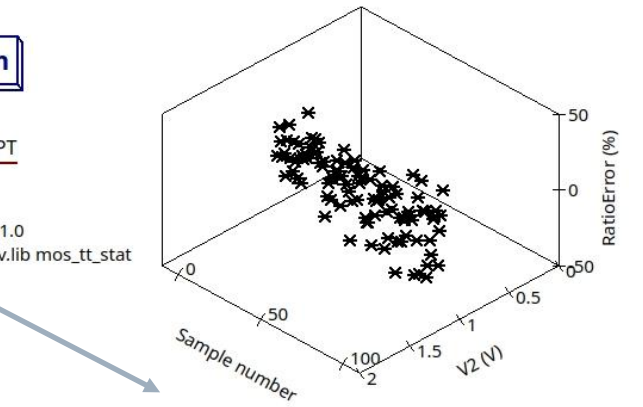


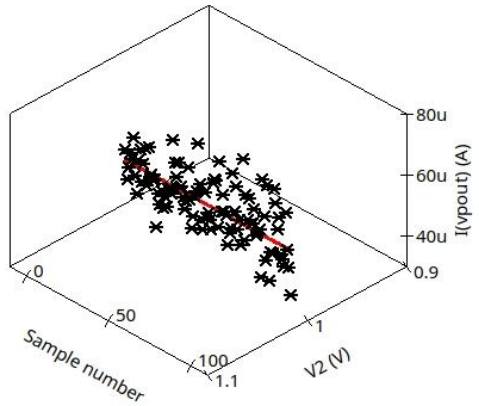
DC Monte Carlo simulation: 2

```
.INCLUDE SCRIPT
INCLSCR5
SpiceCode=
.control
let number_v3 = 0
let nruns = 100
echo "STEP sw1.v3" > spice4qucs.sw1.cir.res
dowhile $&number_v3  le  $&nruns
   reset
   alter v3 = $&number_v3
   dc v2 1.0 1.1 1.0
   let RatioError = 100.0*( 1.0 - (i(vpout)/( i(vpin)+1e-12 ) ) )
   write spice4qucs.sw1._swp.plot i(VPin) i(VPout) v(n1) RatioError
   set appendwrite
   echo "$&number_v3 " >> spice4qucs.sw1.cir.res
   let number_v3 =number_v3 + 1
end
unset appendwrite
destroy all
reset
exit
.endc
.end
```

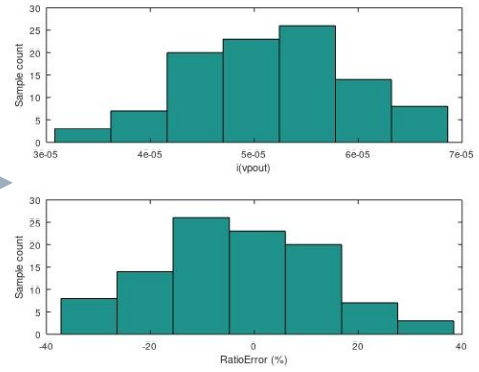




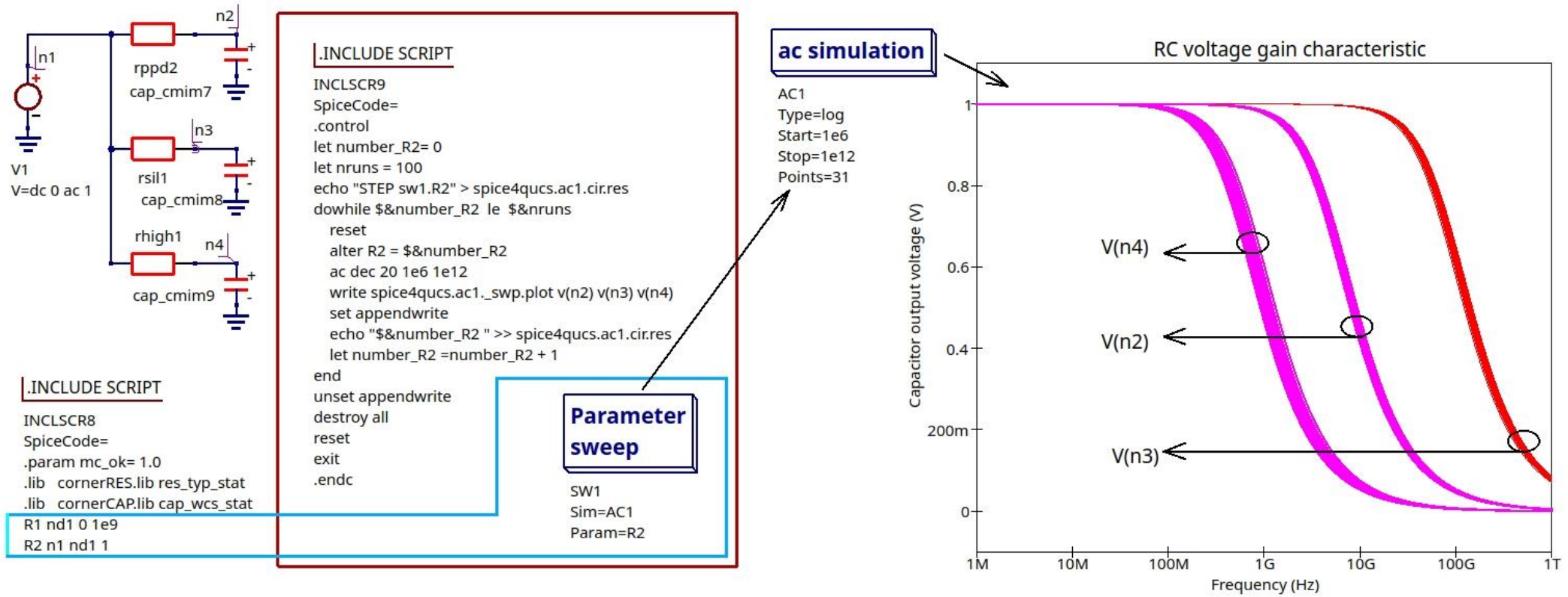





```
% Octave histogram plot.
DataSetFile = "CM_MC_S2_2A.dat.ngspice"
Y1name = "sw1.i(vpout)"
Y1label = "Sample count"
X1label = "i(vpout)"
Y2name = "sw1.ratioerror"
Y2label = "Sample count"
X2label =  "RatioError (%)"
nbins = 7
plotHistogram2V(DataSetFile, Y1name, Y1label, X1label, Y2name, Y2label, X2label, nbins)
```

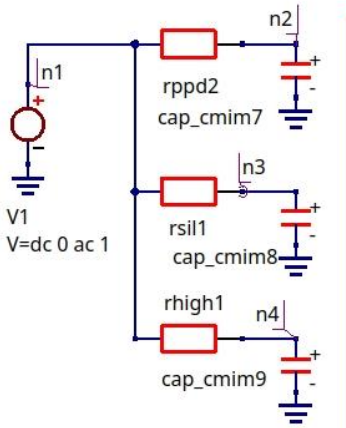


AC Monte Carlo simulation: 1



- Dummy component **R2** and parameter **nruns** link Ngspice script and Qucs-s schematics
- Note use of IHP res-typ_stat and cap_wcs_stat libraries

AC Monte Carlo simulation: 2



.INCLUDE SCRIPT

```
INCLSCR8
SpiceCode=
.param mc_ok= 1.0
.lib   cornerRES.lib res_typ_stat
.lib   cornerCAP.lib cap_wcs_stat
```

.INCLUDE SCRIPT

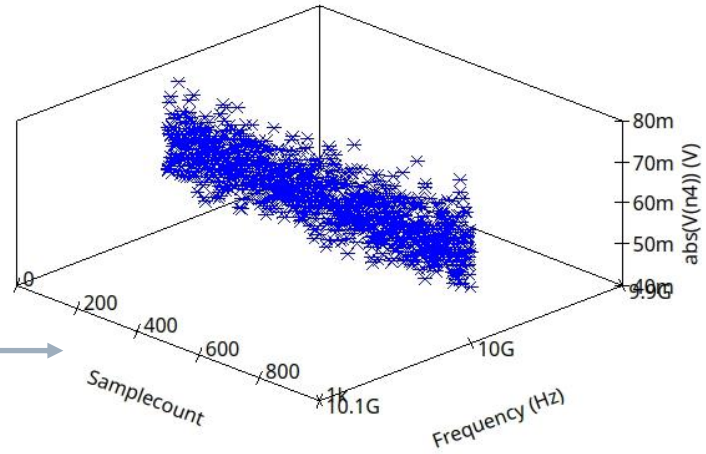
```
INCLSCR6
SpiceCode=
*
R1 nd1 0 1e9
R2 n1 nd1 1
.control
let number_r2 = 0
let nruns = 1000
echo "STEP sw1.r2" > spice4qucs.ac1.cir.res
dowhile $&number_r2 le $&nruns
 reset
 alter R2 = $&number_r2
  ac lin 1 1e10 10.1e10
  write spice4qucs.ac1._swp.plot v(n1) v(n2) v(n3) abs(v(n4))
  set appendwrite
  echo "$&number_r2 " >> spice4qucs.ac1.cir.res
  let number_r2 = number_r2 + 1
end
unset appendwrite
destroy all
reset
exit
.endc
.END
```

Parameter sweep

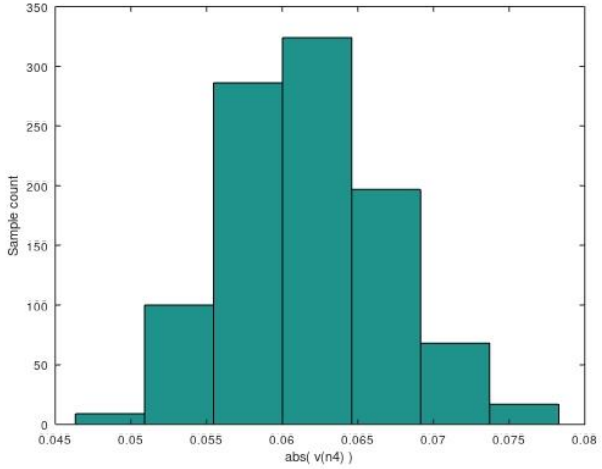
SW1
Sim=AC1
Param=R2

ac simulation

AC1
Type=const
Values=[10g]

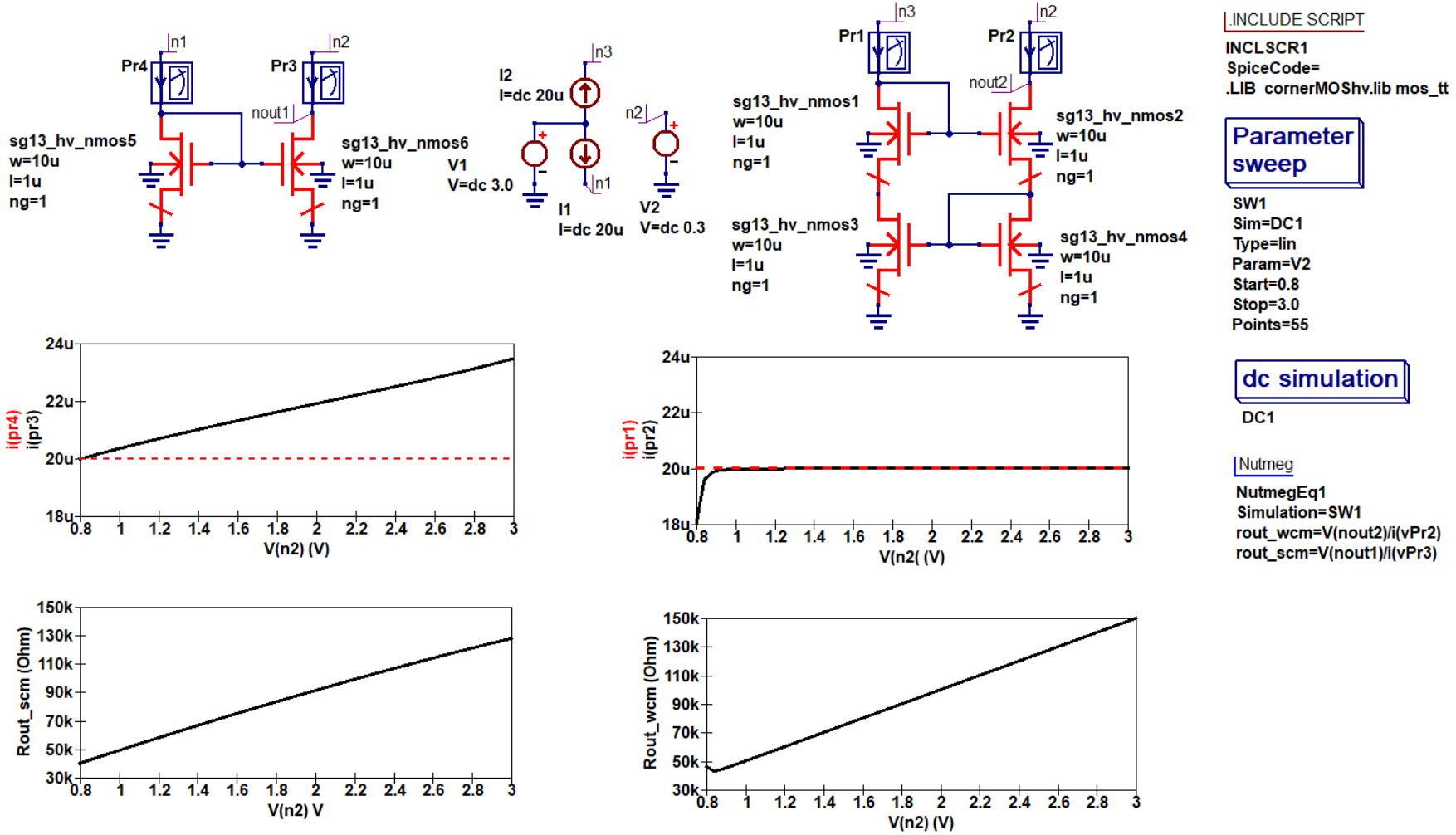


Octave script

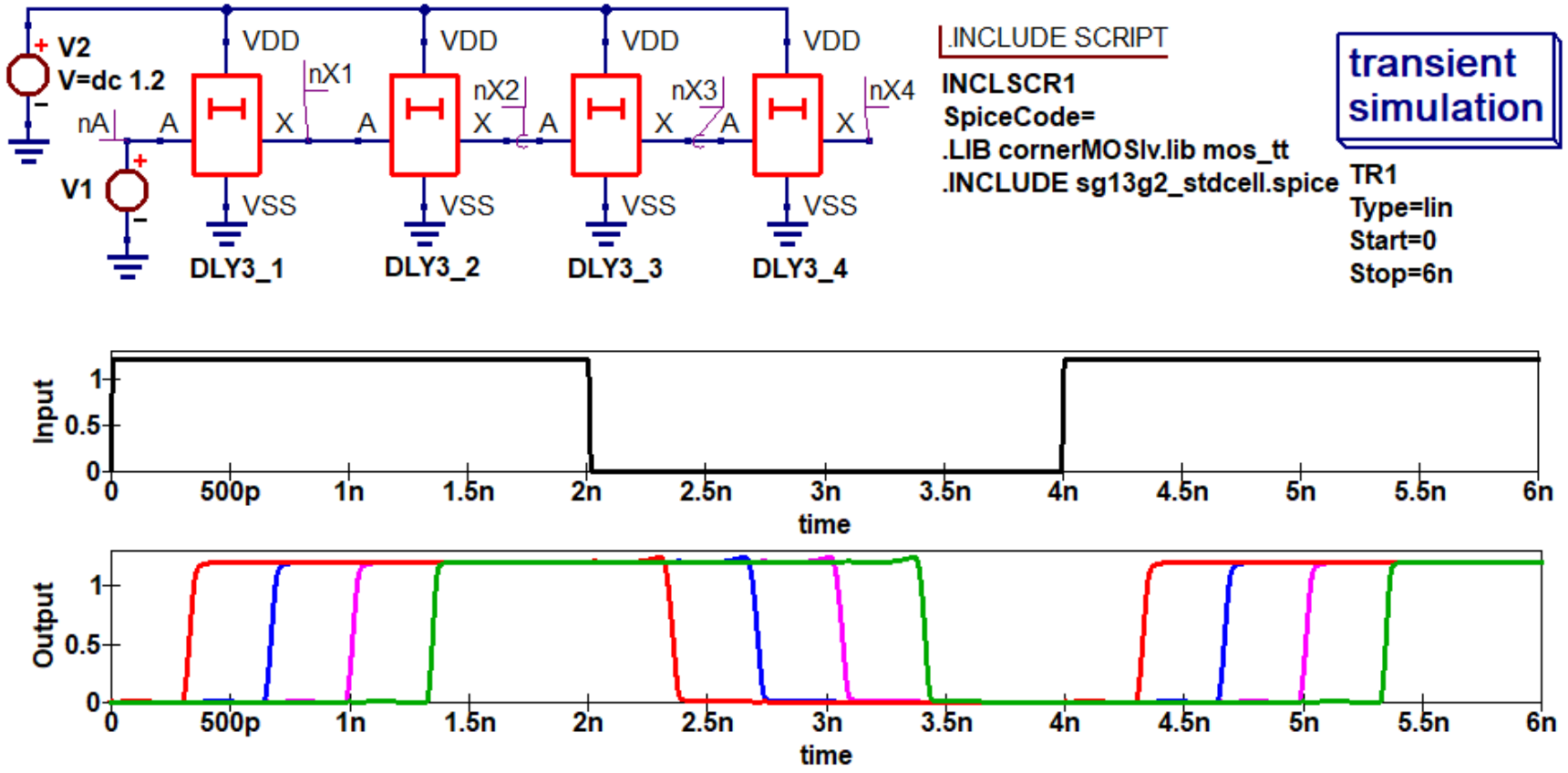


- Frequency set at 10GHz
- nruns = 1000

Example sg13g2 hv analog CM blocks

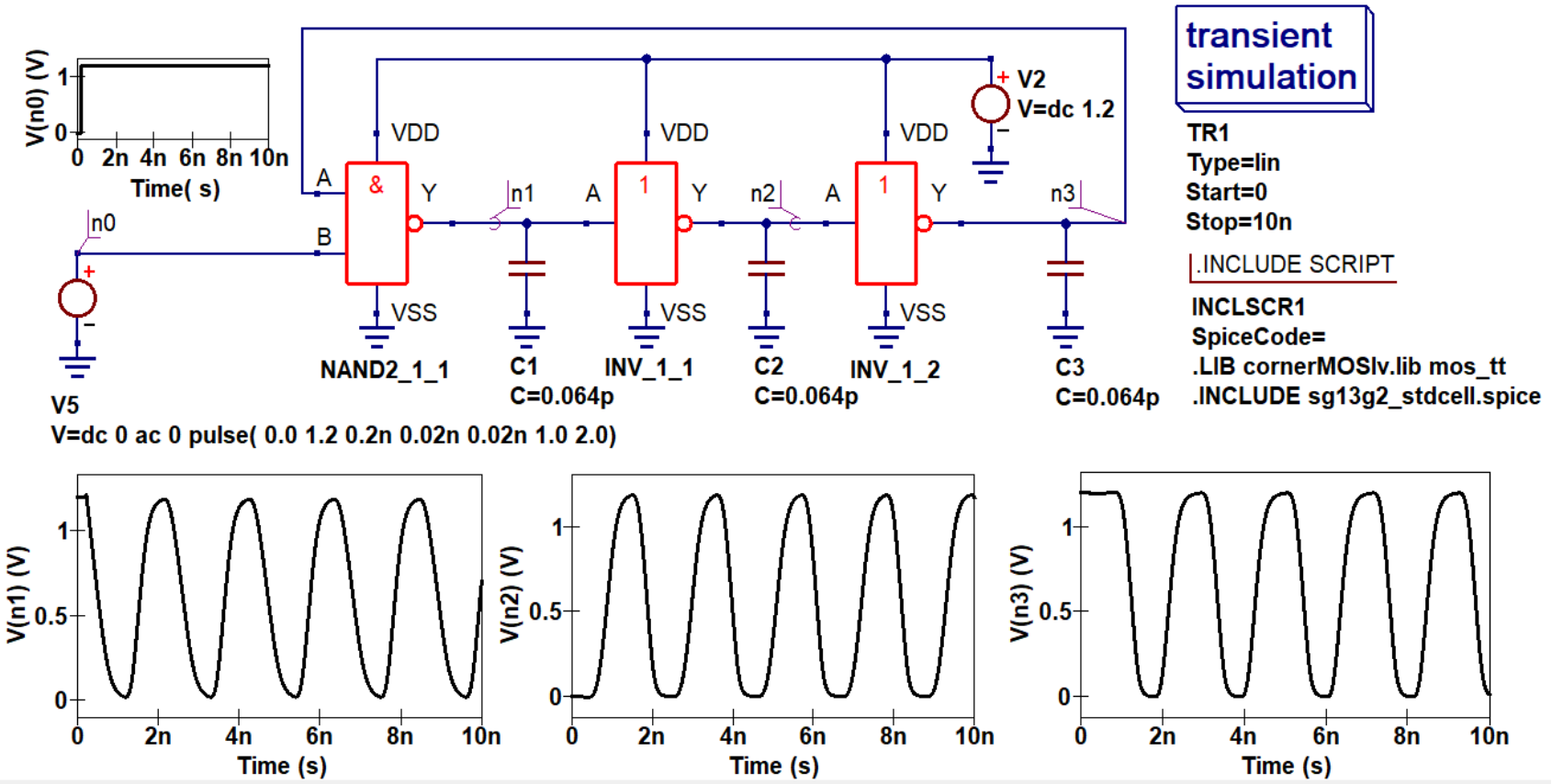


sg13g2 digital stdcell simulation



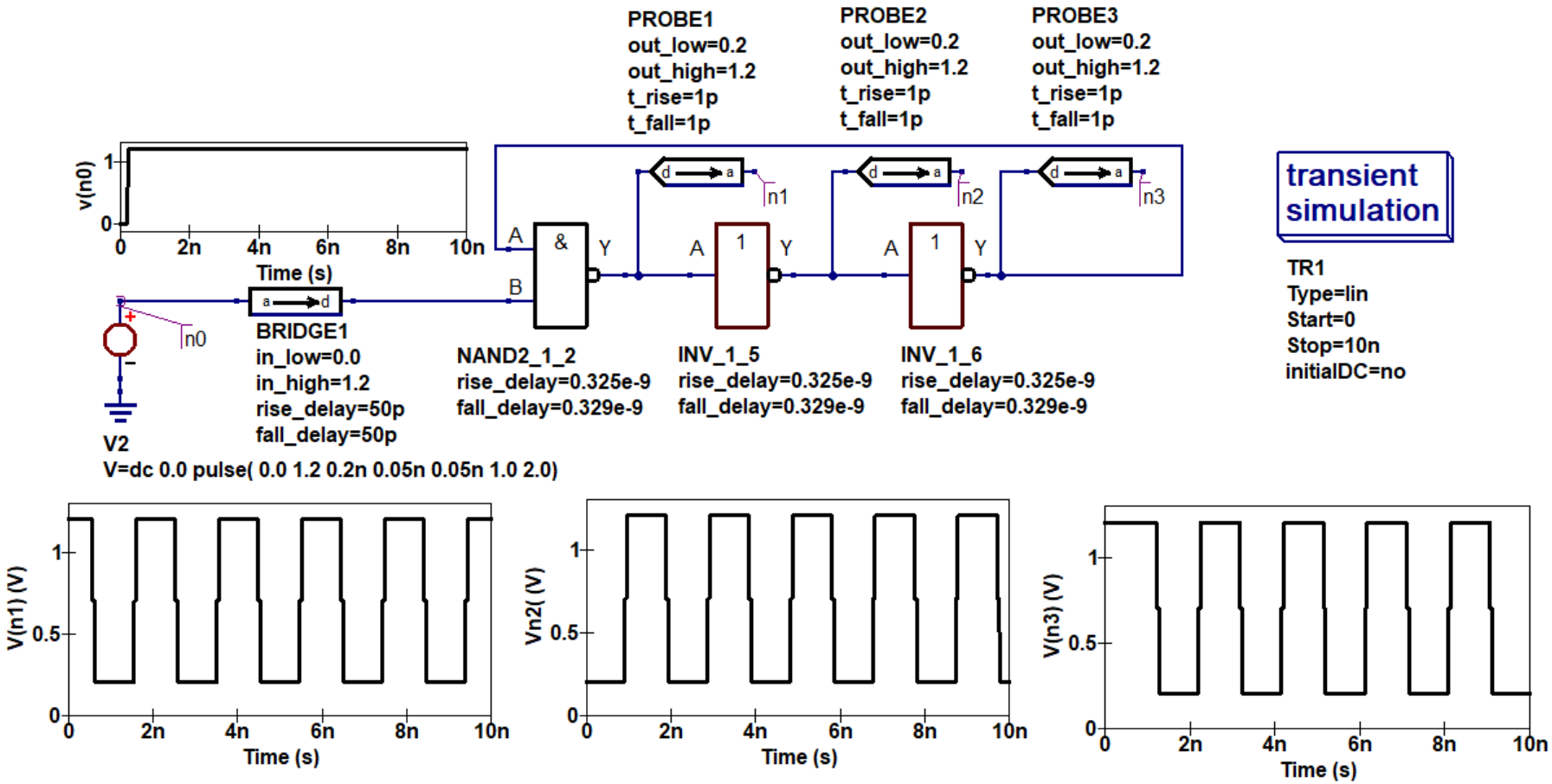
- Logic 0 = 0.0V and 1 = 1.2V
- Output signals: DLY3_1 = red; DLY3_2 = blue; DLY3_3 = mauve; DLY3_4 = green

Mixed analog and digital simulation



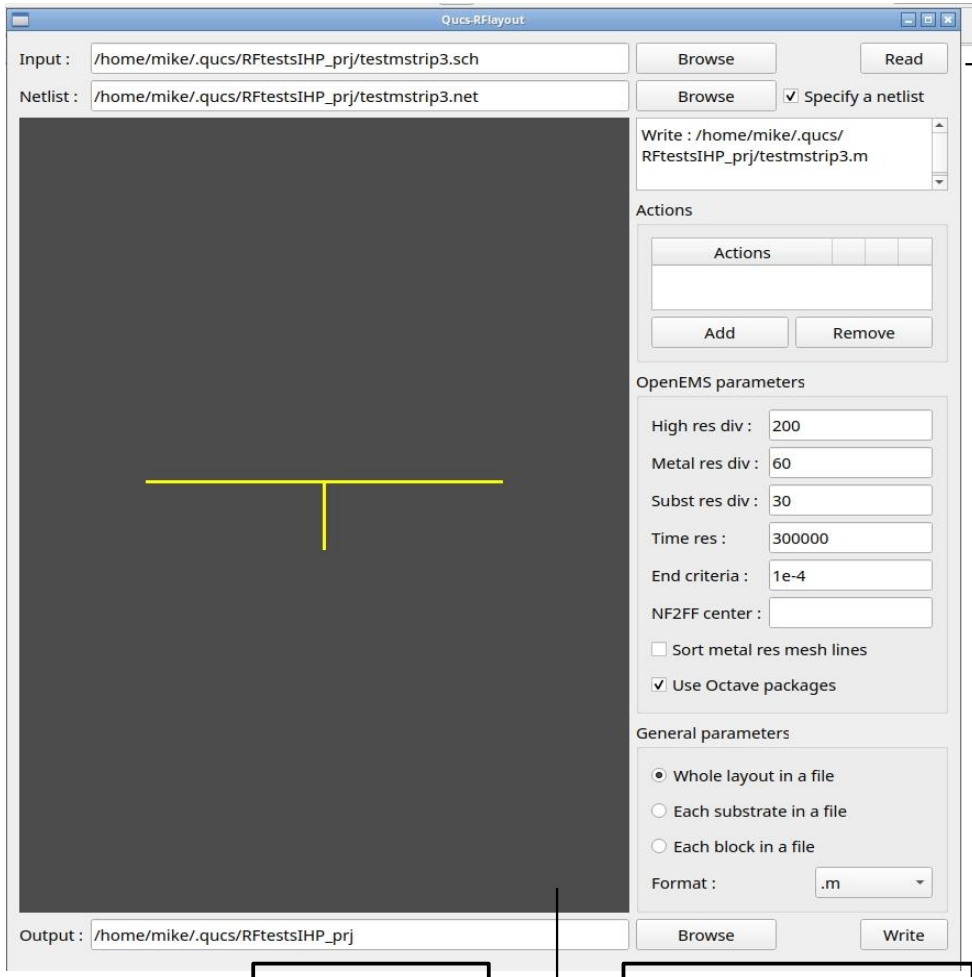
- Qucs-S/Ngspice IHP PDK digital stdcell ring oscillator
- C1, C2 and C3 set to measured values for "mid" size devices

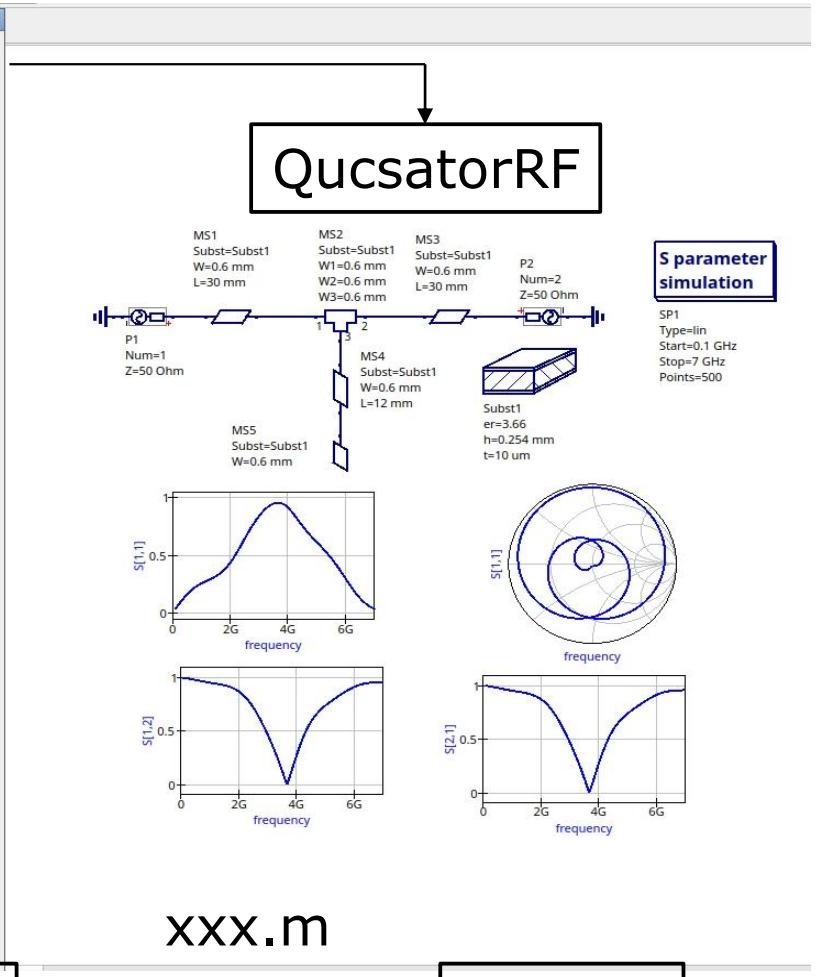
XSPICE digital simulation

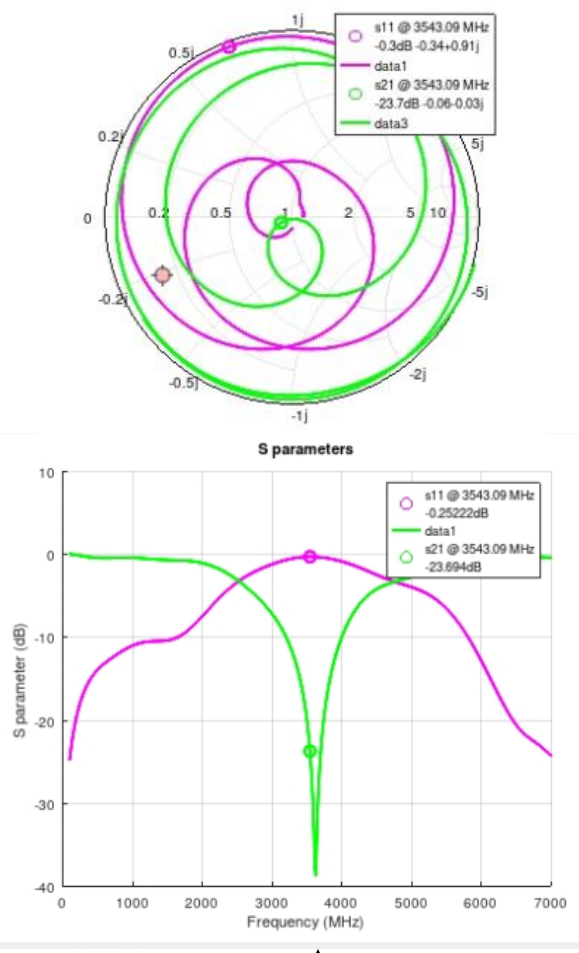


- Qucs-S/XSPICE digital stdcell ring oscillator
- Logic levels: 0 = 0.0V, U = 0.6V and 1 = 1.2V

QucsatorRF/openEMF simulation







KiCAD ↔ QucsatorRF → Octave

QUCS-S - a central tool in the openPDK IC design flow

Summary

• The recent rapid developments in semiconductor technologies and devices has increased the pressure on circuit design tools to keep in step with user needs. This is particularly true in areas driven by the release of open source analogue, RF and mixed signal PDKs.

• This presentation reports on developments in the latest Qucs-S and Ngspice FOSS tools for IC design using the IHP 130 nm BiCMOS Open Source PDK (sg13g2).

• A series of Qucs-S/Ngspice/IHP PDK simulation test benches outline the application of Free Open Source (FOSS) tools, or equivalent licence software, for IC design.

• All the software tools outlined in this presentation are freely available from the web sites listed on slide 20.

• Future work will concentrate on developing a series of analogue/RF/mixed signal test structures for production by IHP and subsequent performance measurement. This will allow the accuracy of the Qucs-S/Ngspice simulation output data to checked.

Acknowledgments and FOSS links

The IHP (Leibniz-Institut für innovative Mikroelektronik) Open PDK Development Team (project leader René Scholtz) : https://github.com/IHP-GmbH/IHP-Open-PDK,

The Qucs-S (Qucs with SPICE) Development Team, https://ra3xdh.github.io/ : Qucs-S 24.3.0, 23 July 2024

The Ngspice (next generation open source SPICE) Development Team https://ngspice.sourceforge.io) : ngspice-43_64.7z, 13 July 2024

The openEMS (open electromagnetic field solver) Development Team : https://github.com/thliebig/openEMS-Project, openEMS 0.0.36, 22 October 2023

The openVAF (next-Generation Verilog-A compiler) Development Team https://openvaf.semimod.de/ : openVAF with noise support (Beta), 1 January 2024