NGSPICE - XSPICE ELEMENTAL DEVICES MADE AVAILABLE IN KICAD

Holger Vogt

Duisburg, Germany, | 01.02.2025

Fraunhofer IMS

Elektronische Bauelemente & Schaltungen

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

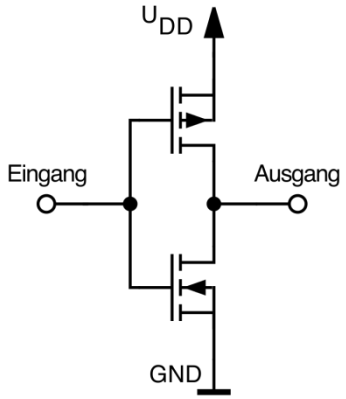
Contents

- Intro to the ngspice circuit simulator
- What is new in ngspice ?
- XSPICE code models in ngspice
- KiCad supporting code models
- Simulation examples
- What is next in ngspice ?

ngspice – what is it ?

Circuit simulator that numerically solves equations describing (electronic) circuits made of passive and active devices for (time varying) currents and voltages.

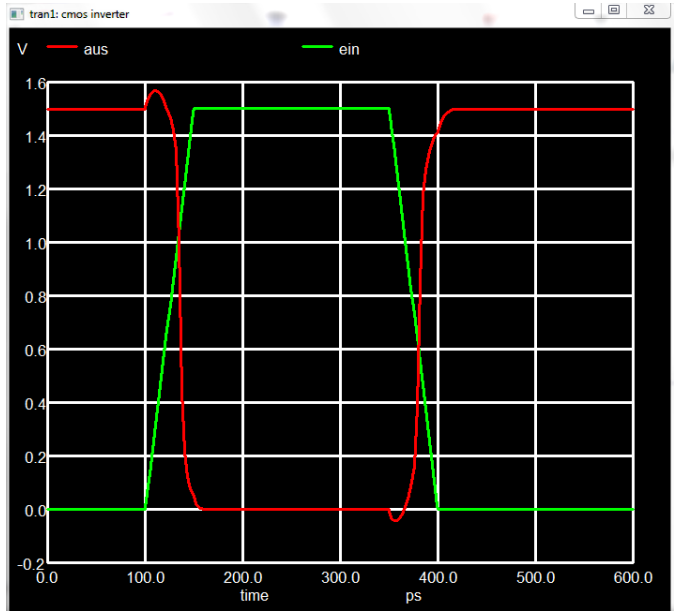
Open source successor of venerable spice3f5 from Berkeley



the circuit

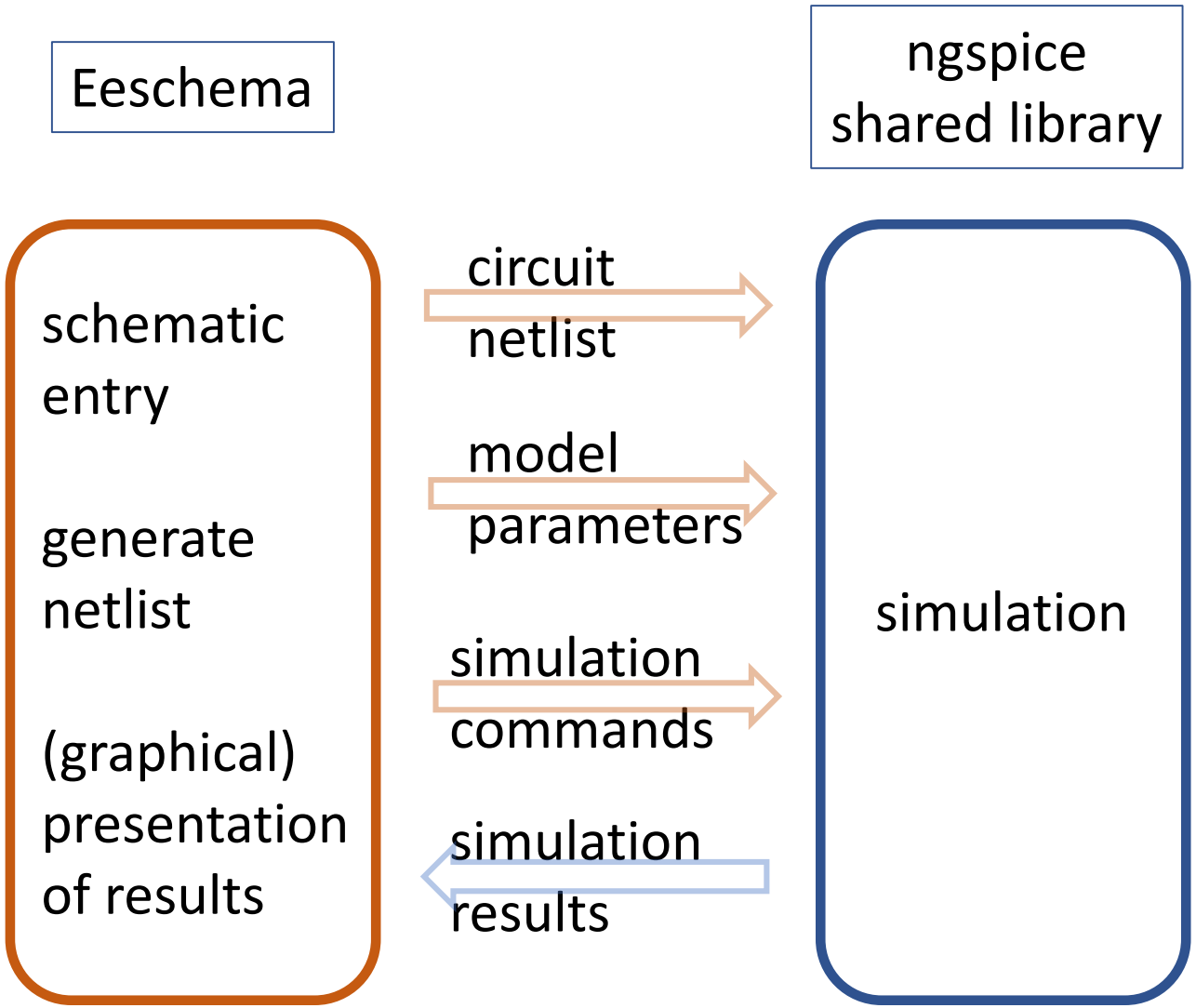
```
CMOS inverter

.include ./bsim4soi/nmos4p0.mod
.include ./bsim4soi/pmos4p0.mod
.option TEMP=27C

Vpower VD 0 1.5
Vgnd VS 0 0

Vgate Ein VS PULSE(0 1.5 100p 50p 50p 200p 500p)

MN0 Aus Ein VS VS N1 W=10u L=0.18u
MP0 Aus Ein VD VS P1 W=20u L=0.18u

.tran 3p 600ps

.control
  run
  plot Ein Aus
.endc

.END
```

the input

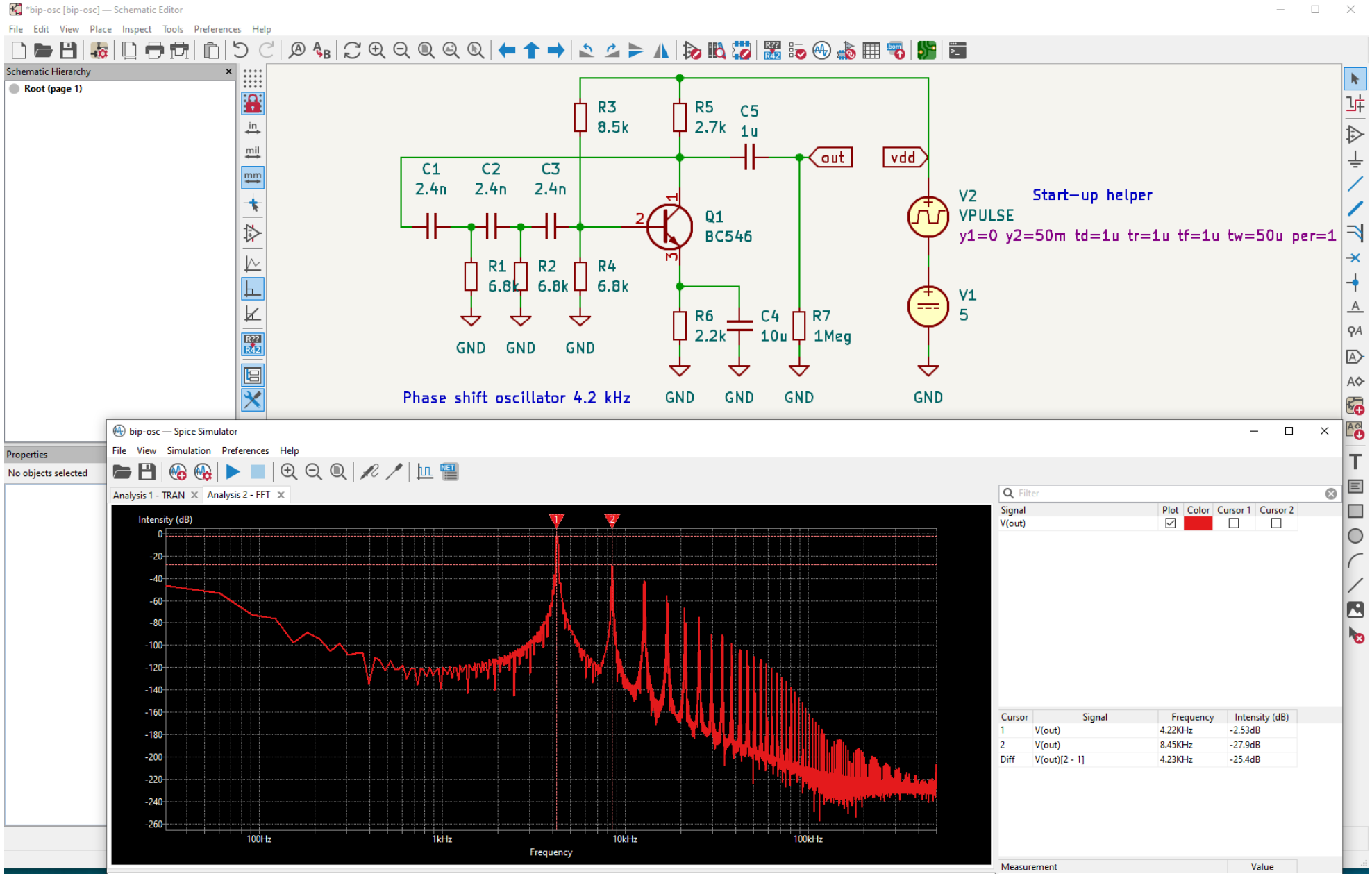


the output

KiCad/ngspice Interface



KiCad/Eeschema simulator window



What's new in ngspice?

- Current release ngspice-44.2 from Jan. 11th, 2025
- Support for Verilog-A compact device models and analog building blocks
- Co-simulation ngspice mixed-signal – Verilog digital (using Verilator, or Icarus Verilog)
- Co-simulation ngspice mixed-signal – VHDL digital (using GHDL) *
- Co-simulation ngspice mixed-signal – C-coded digital
- Degradation simulation (stress, radiation)

https://sourceforge.net/projects/ngspice/files/ng-spice-rework/44.2/ https://ngspice.sourceforge.io/

* https://sourceforge.net/p/ngspice/ngspice/ci/pre-master-45/tree/ (development branch)

ngspice and Verilog-A co-simulation

```
// importing libs

`include "discipline.h"

module diff_amp(
  output electrical out,
  input electrical in1,
  input electrical in2);

parameter real gain = 40; // setting gain to 40 of the differential amplifier
parameter real vcc = 3;   // swing from -vcc to +vcc
parameter real offset = 3; // added offset

analog begin

    V(out) <+ offset / 2 + vcc / 2 * tanh( gain / vcc * 2 * V(in1, in2));

end
endmodule
```

Compile with OpenVAF → Model in shared library

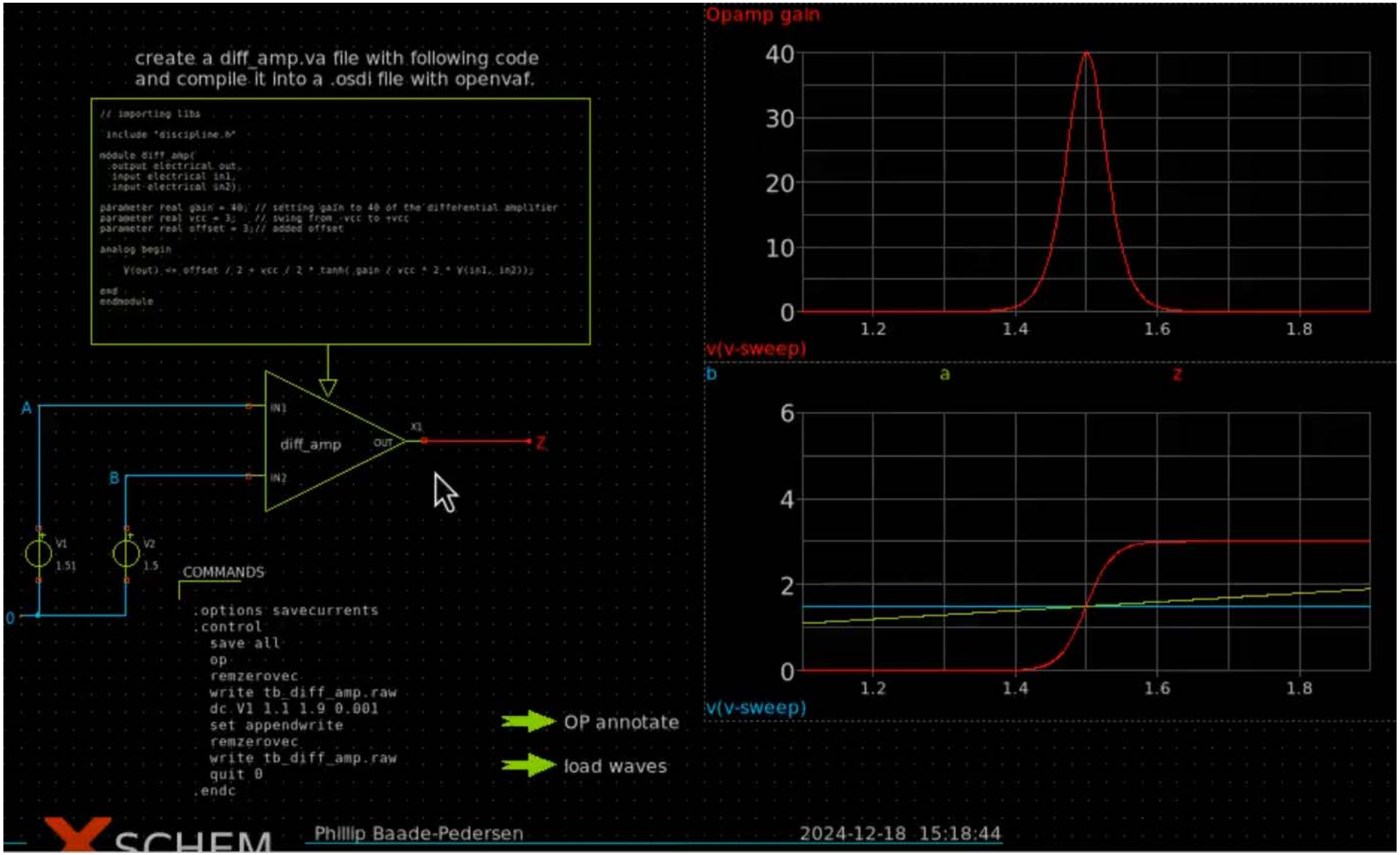
```
.subckt diff_amp_cell OUT IN1 IN2
N1 out in1 in2 diff_amp_model
.ends diff_amp_cell

.model diff_amp_model diff_amp
```

Call to model in the ngspice netlist

Verilog-A model of a simple differential amplifier

Ngspice and Verilog-A cosimulation with Xschem
Author Stefan Schippers
https://www.youtube.com/watch?v=g9WPjZ_e8co

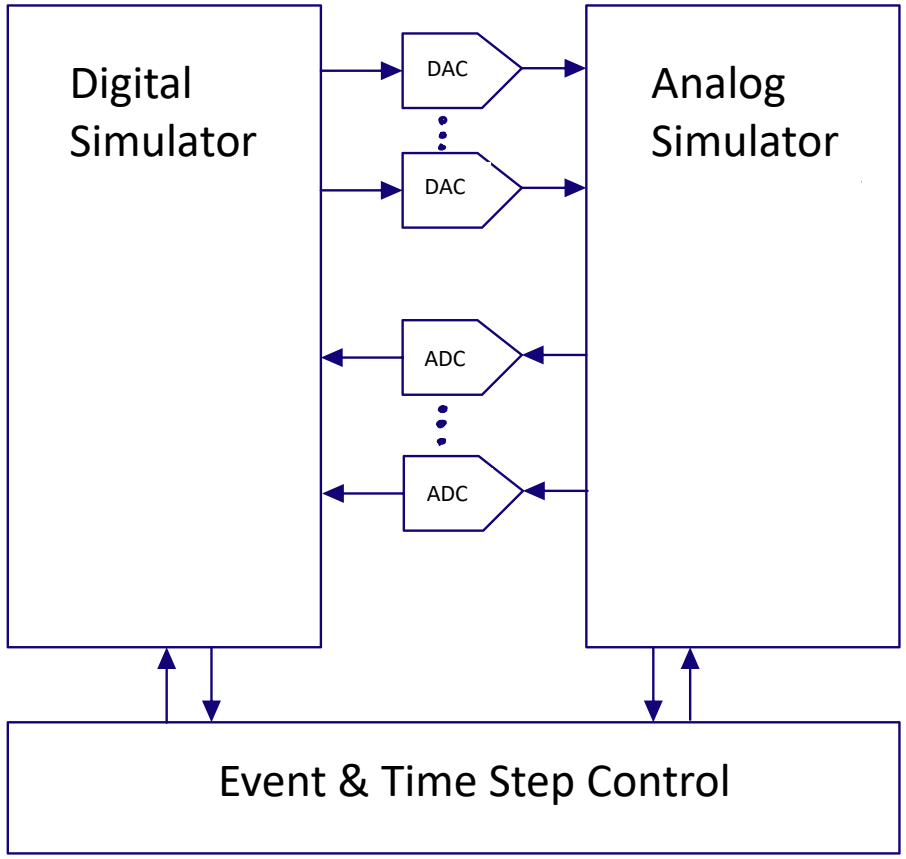
ngspice and Verilog-A



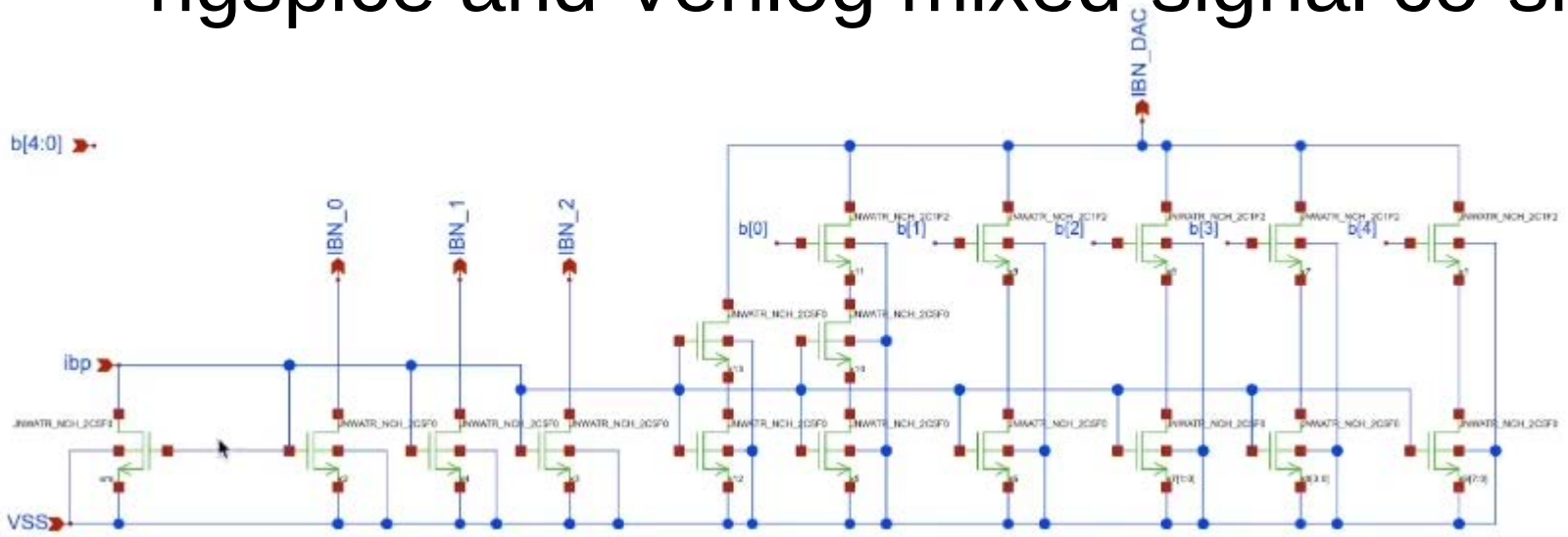
ngspice and Verilog/VHDL mixed-signal co-simulation

ngspice analog and Verilator/Iverilog/ GHDL event based co-simulation

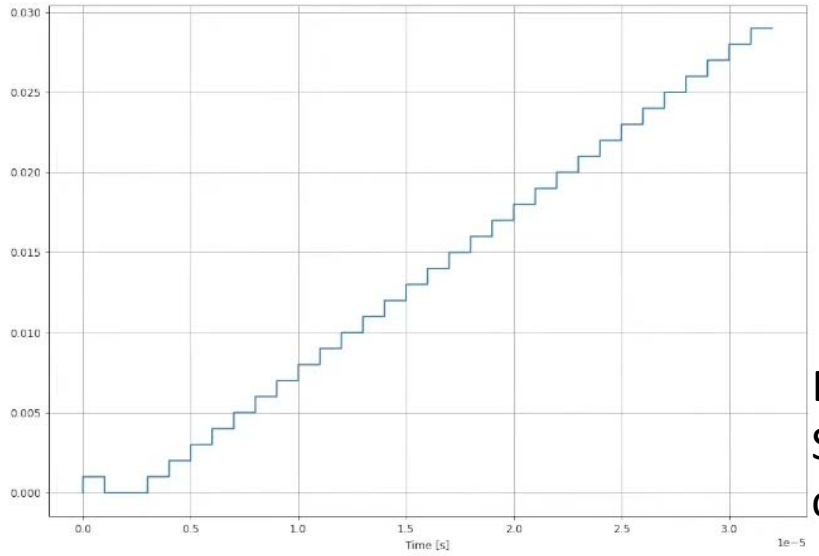
(but also Ngspice analog and internal XSPICE event based co-simulation)



ngspice and Verilog mixed-signal co-simulation



Analog module:
DAC with weighted current mirrors and switches



DAC output:
Summed-up current

Digital module:
Counter

Compile with Verilator
Link to ngspice

```
module dig(
            input wire          clk,
            input wire          reset,
            output logic [4:0] b
            );

   logic                        rst = 0;

   always_ff @(posedge clk) begin
      if(reset)
        rst <= 1;
      else
        rst <= 0;
   end

   always_ff @(posedge clk) begin
      if(rst)
        b <= 0;
      else
        b <= b + 1;
   end // dig
endmodule
```

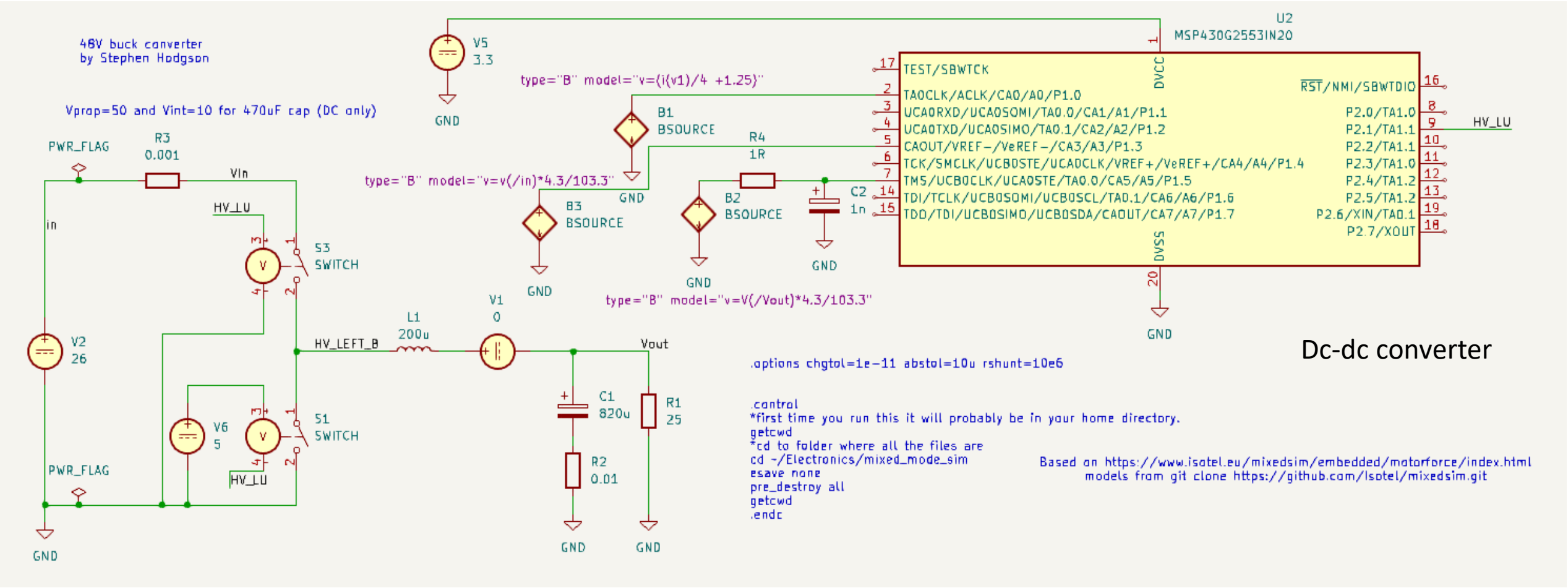
Author: Carsten Wulff

ngspice and Verilog mixed-signal co-simulation

Mixed Signal Simulation in Ngspice
Author: Carsten Wulff
https://www.youtube.com/watch?v=vEZPCIInwmQ&t=1689s
https://analogicus.com/aic2025/2024/12/06/Mixed-Signal-Simulation-in-NGSPICE.html
Tutorial at http://analogicus.com/jnw_sv_sky130a/
Repository at https://github.com/wulffern/jnw_sv_sky130a
Assumes knowledge of https://analogicus.com/aic2025/2025/01/01/Sky130nm-tutorial.html

Interface to ngspice: d_cosim code model: Giles Atkinson

ngspice and C code mixed-signal co-simulation



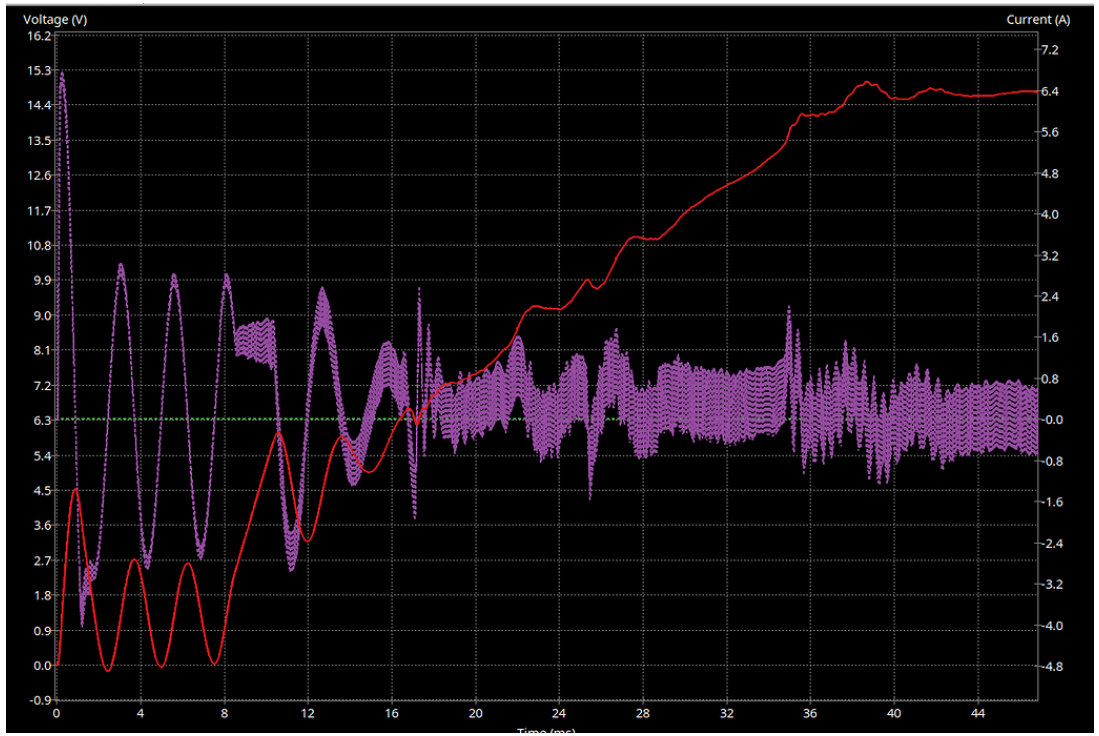
Author: Stephen, @slh

https://forum.kicad.info/t/easier-mixed-mode-simulation-and-real-world-comparison/56530

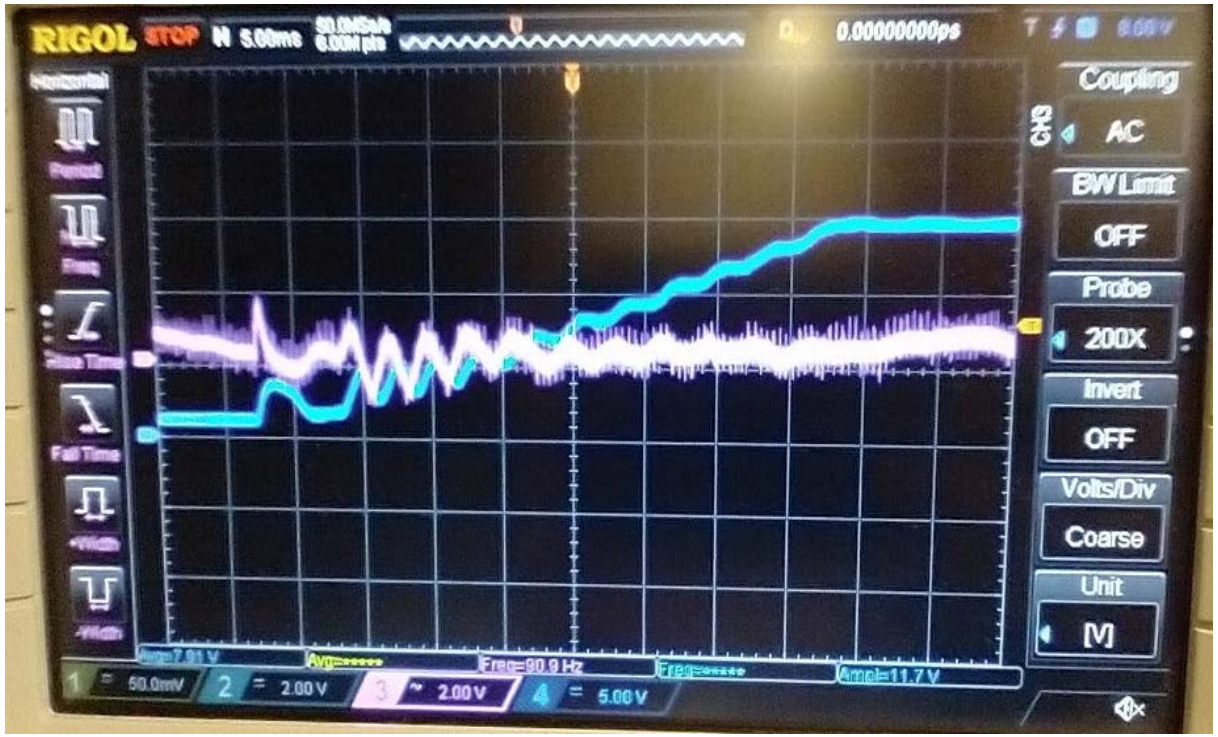
https://forum.kicad.info/t/mixed-mode-simulation-including-c-code-and-verilog/49752

Interface to ngspice: d_process code model: Uros Platise, Brian Taylor

ngspice and C code mixed-signal co-simulation

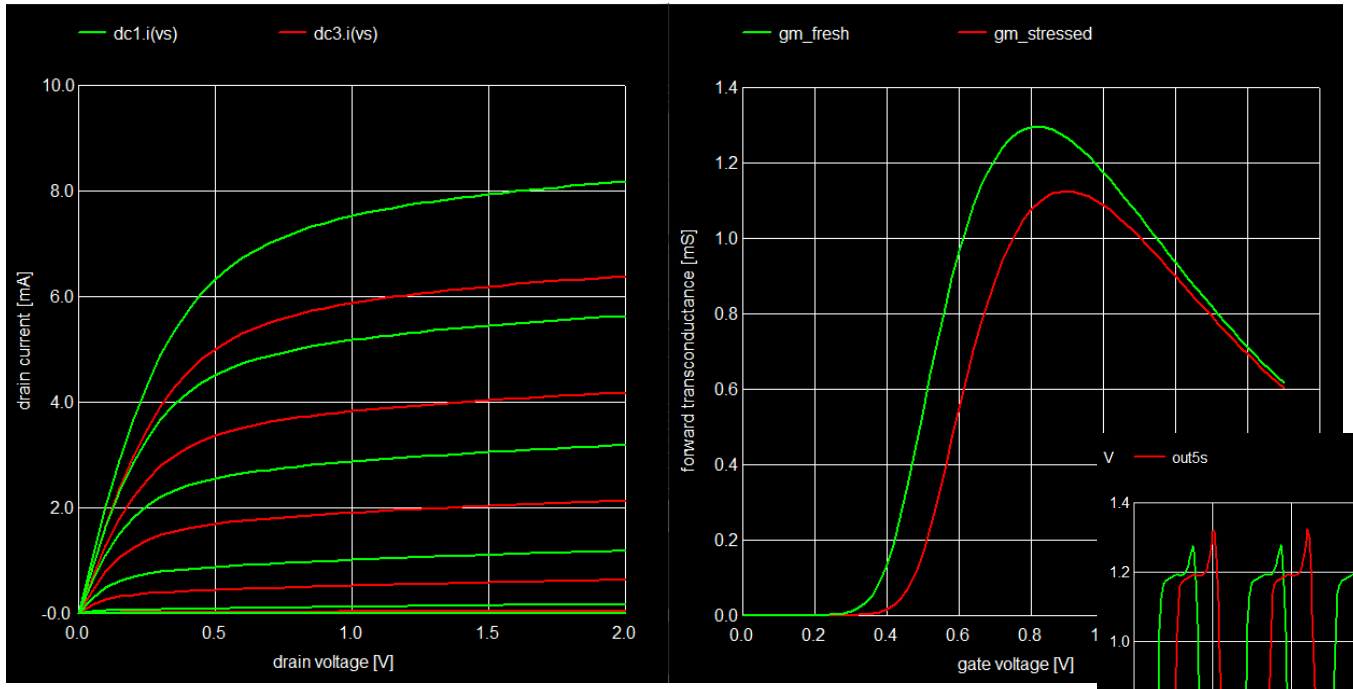


DC-DC-converter Simulation



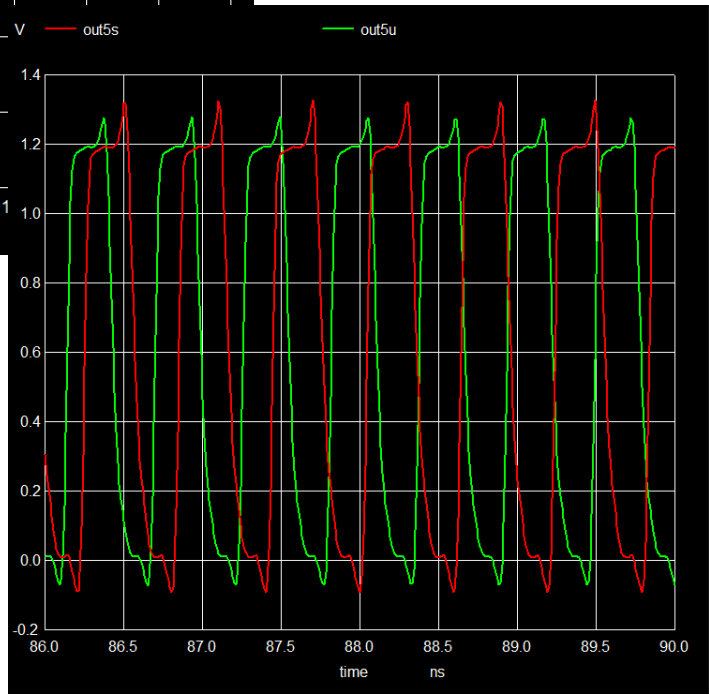
Measurement

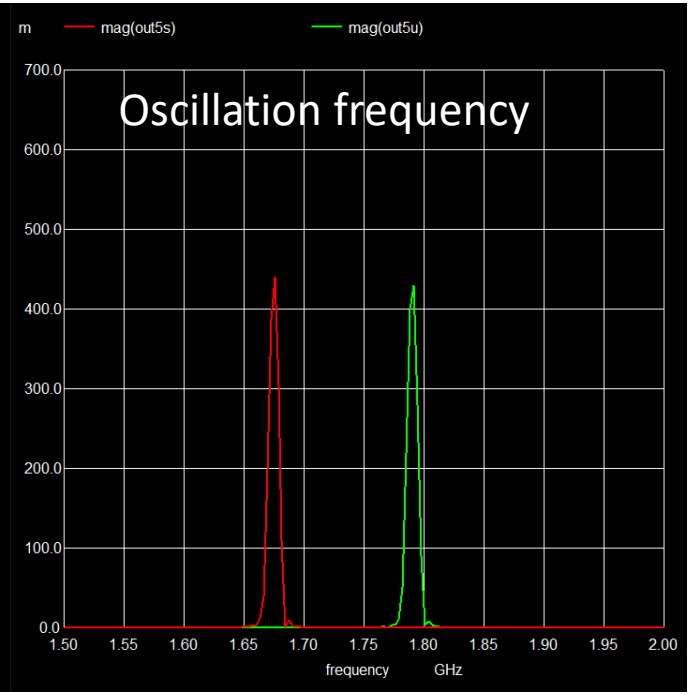
Simulation of a CMOS Circuit after Hot Carrier Degradation



Green: before voltage stress
Red: after voltage stress

Hot Carrier Degradation of a MOS transistor



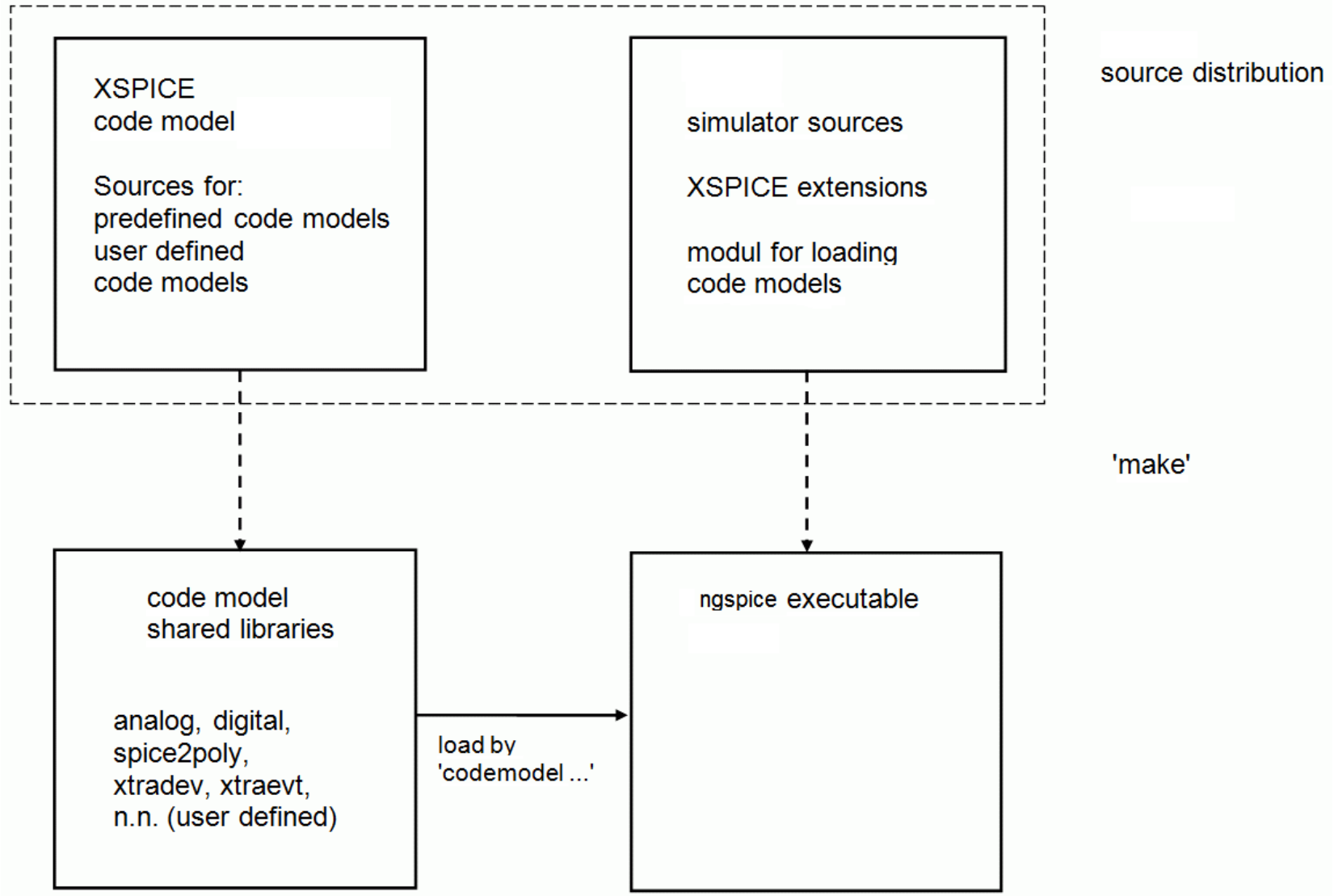


Ring Oscillator

XSPICE in ngspice

- Conceived by Georgia Tech in 1990/91 as an extension of Spice3f.
- Fully integrated into ngspice since its beginning.
- Enables co-simulation of analog blocks and digital event based blocks.
- More than 60 blocks (named ‘code models’) are provided with the ngspice distribution.
- Digital simulation is fast, but lacks waveform details.
- User may create his/her own code models (C code with predefined macros).

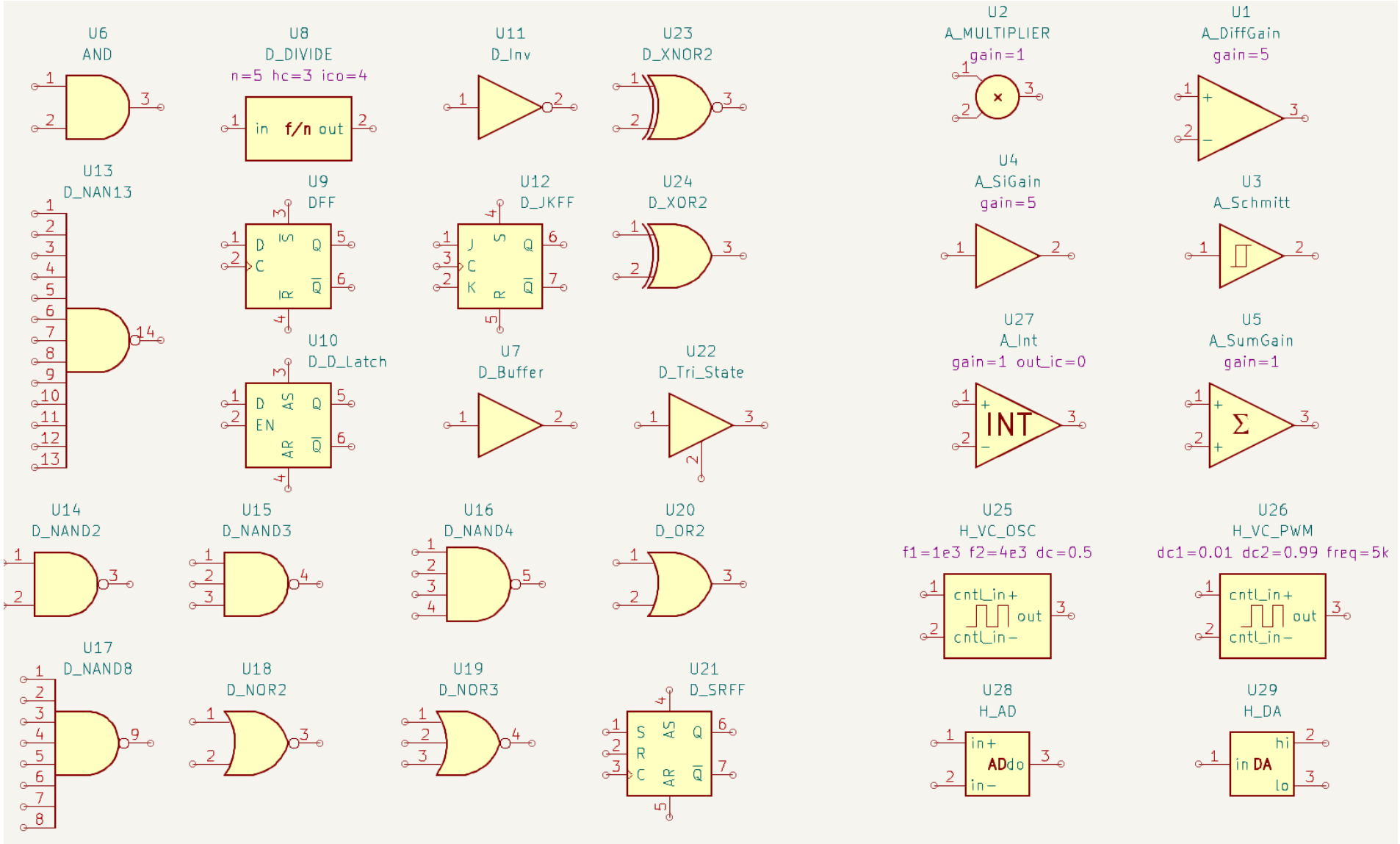
ngspice/XSPICE Top-Level Diagram



Why XSPICE/ngspice support in KiCad?

- Task: create and simulate electronic systems.
- Use KiCad/Eeschema/ngspice simulation capability.
- Facilitate the use of the code models.
- Schematic entry at block level.
- Building blocks for system evaluation, not linked to PCB design.
- May be combined with the Verilog-A and HDL digital approaches described before.

XSPICE Elemental Devices

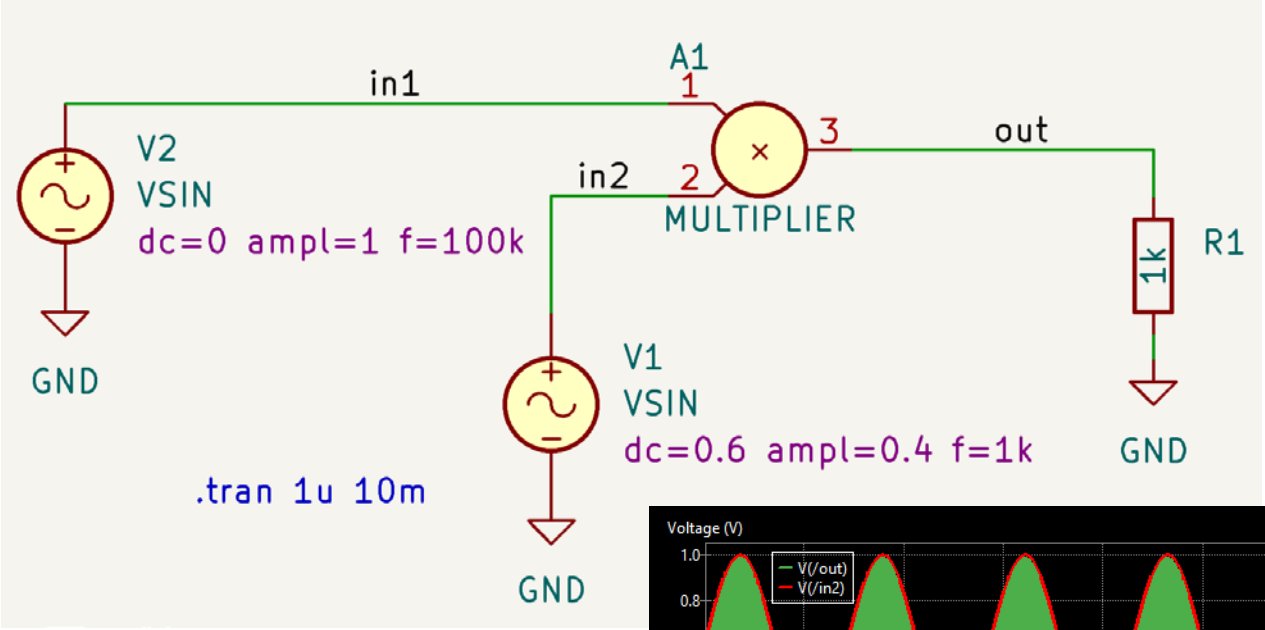


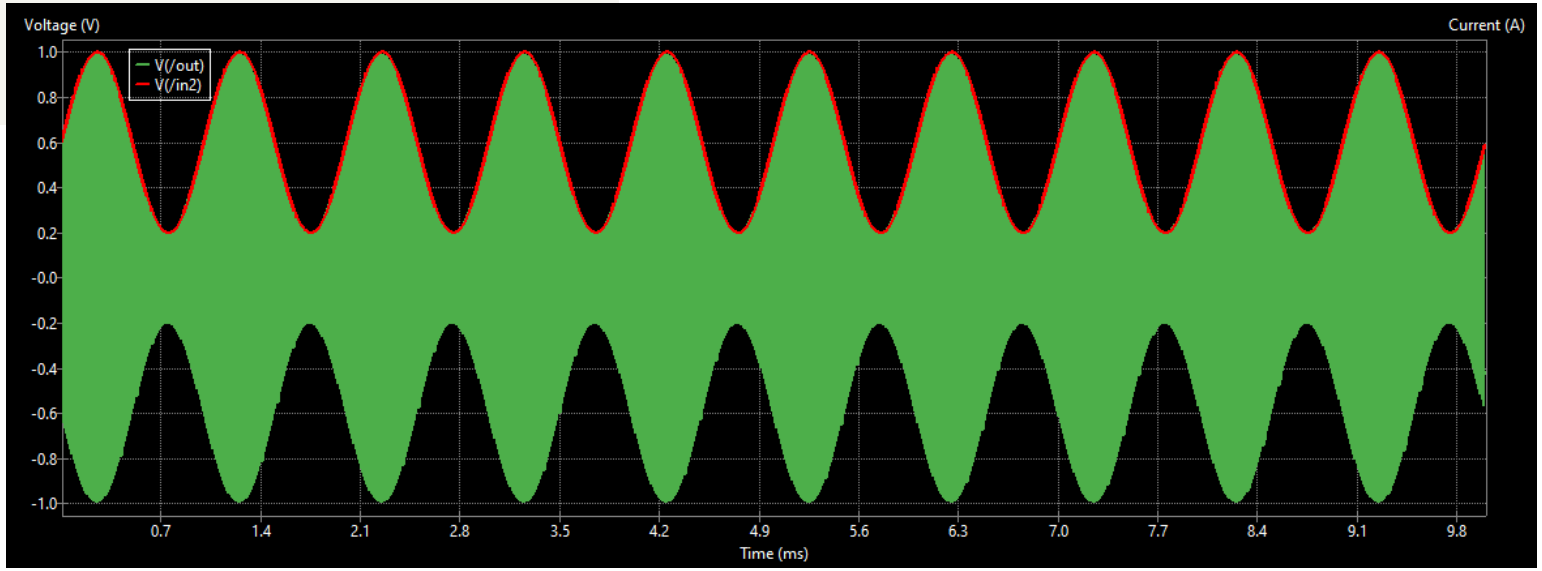
https://forum.kicad.info/t/simulation-with-xspice-code-models/56384

How to install the code models in KiCad?

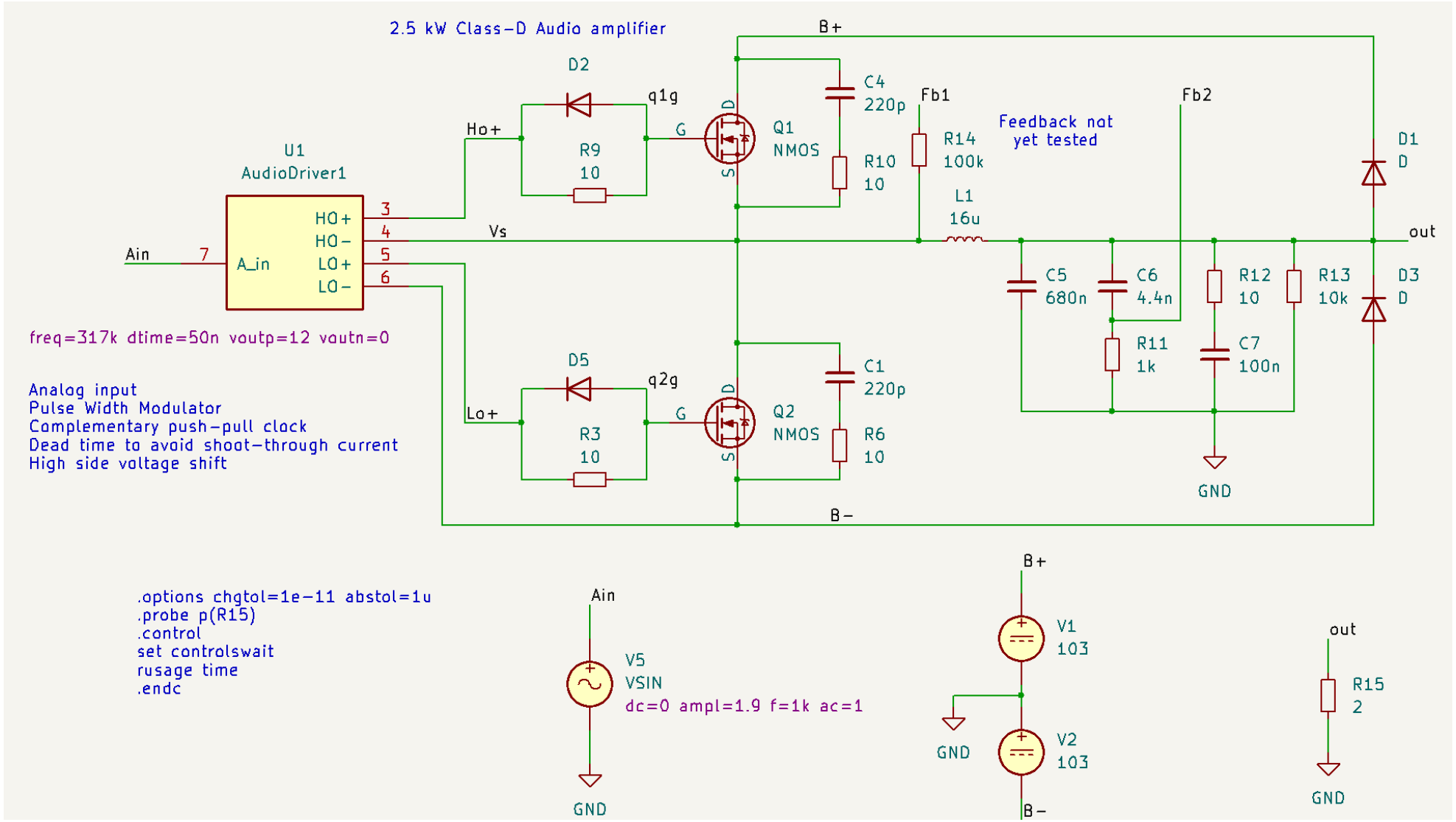
- Download XSPICE-symbols-models-4.7z from https://forum.kicad.info/t/simulation-with-xspice-code-models/56384/3 .
- Expand and save both files into the KiCad symbol directory (C:\Program Files\KiCad\8.0\share\kicad\symbols on MS Windows). Simulation_XSPICE.kicad_sym contains the symbols, Simulation_XSPICE.sp the associated ngspice models.
- Add the symbols to the list of available symbols via the KiCad Symbol Editor.

Example: Amplitude Modulation





2.5 kW class D audio amplifier

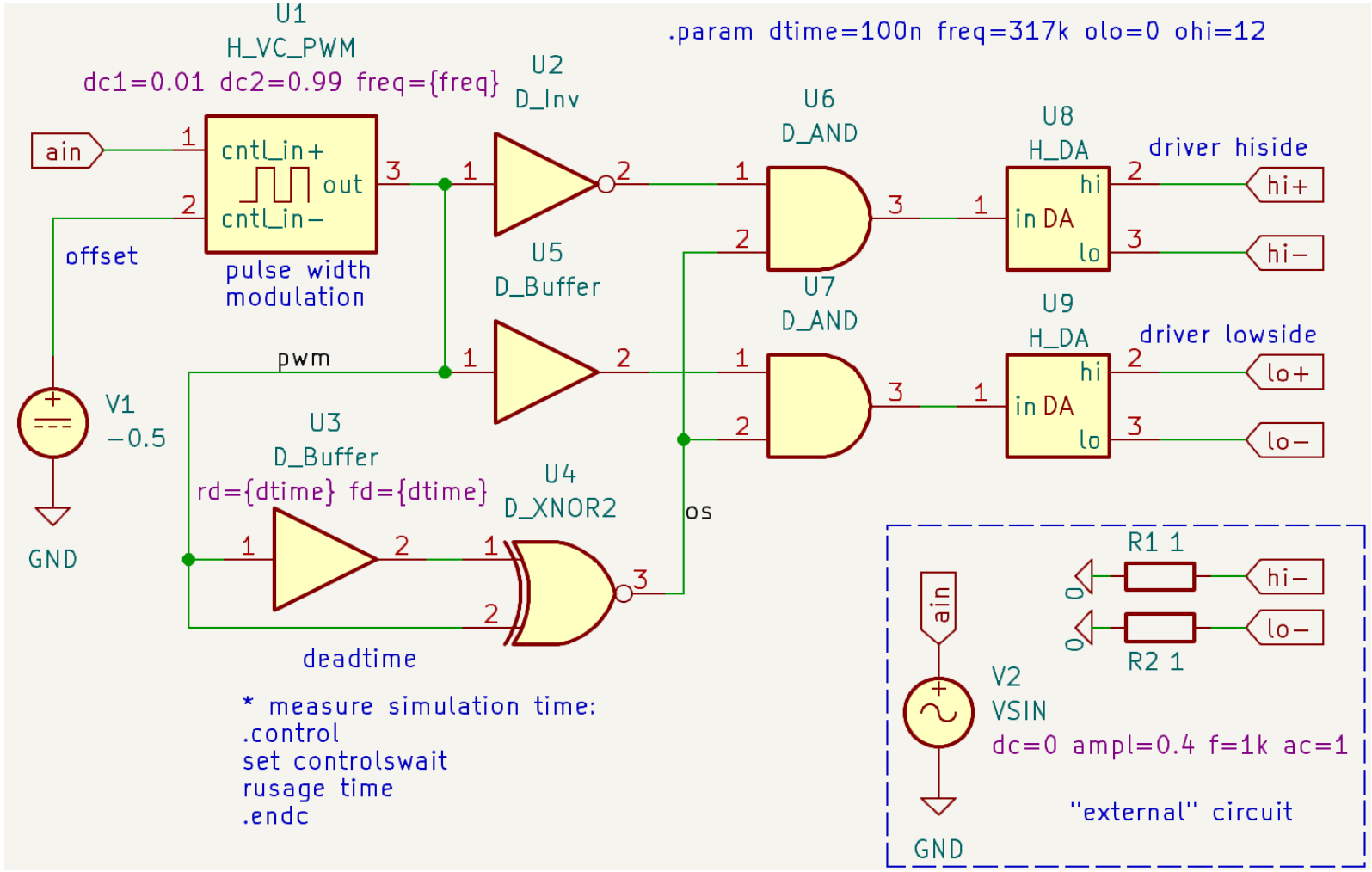


Example: Audio Driver

```
* Class D audio amp frontend (to drive a power MOS half bridge)
* analog input
* pwm clock generator
* digital one-shot
* non-overlapping clock
* two floating half-bridge drivers


* Calling the subcircuit
* Xpwm ain lo+ lo- hi+ hi- DAudioDriver freq = 500k dtime = 100n voutp = 1.4 voutn = 0

.subckt DAudioDriver ain lo+ lo- hi+ hi- params: freq = 317k dtime = 50n voutp = 12 voutn = 0
apwm1 ain dfast1 pwm_osc
.model pwm_osc d_pwm(cntl_array = [-2 -1.99 1.99 2]
+ dc_array = [0.1 0.1 0.9 0.9]
+ frequency = {freq} init_phase = 90.0

a6 dfast1 _d1 inv1
.model inv1 d_inverter(rise_delay = 0.3e-9 fall_delay = 0.3e-9
+ input_load = 0.5e-12)

* equalize d1 and _d1
abuf2 dfast1 d1 buff2
.model buff2 d_buffer(rise_delay = 0.3e-9 fall_delay = 0.3e-9
+ input_load = 0.5e-12)

*** one-shot ***
* buffer
abuf1 dfast1 d2 buff1
.model buff1 d_buffer(rise_delay = {dtime} fall_delay = {dtime}
+ input_load = 0.5e-12)
* one-shot 1->0 output
a9 [dfast1 d2] dos xnor3
.model xnor3 d_xnor(rise_delay = 0.2e-9 fall_delay = 0.2e-9
+ input_load = 0.5e-12)
***

* outputs: inverted, non-overlapping
aand1 [d1 dos] dout1 and1
aand2 [_d1 dos] dout2 and1
.model and1 d_and(rise_delay = 0.4e-9 fall_delay = 0.4e-9
+ input_load = 0.5e-12)

* analog out, differential
abridge1 [dout1] [%vd(lo+ lo-)] dac1
abridge2 [dout2] [%vd(hi+ hi-)] dac1
.model dac1 dac_bridge(out_low = {voutn} out_high = {voutp} out_undef = 0
+ input_load = 5.0e-12 t_rise = 20e-9
+ t_fall = 20e-9)

* test
* do we have overlap?
* aandtest [dout1 dout2] dtest and1

.ends
```

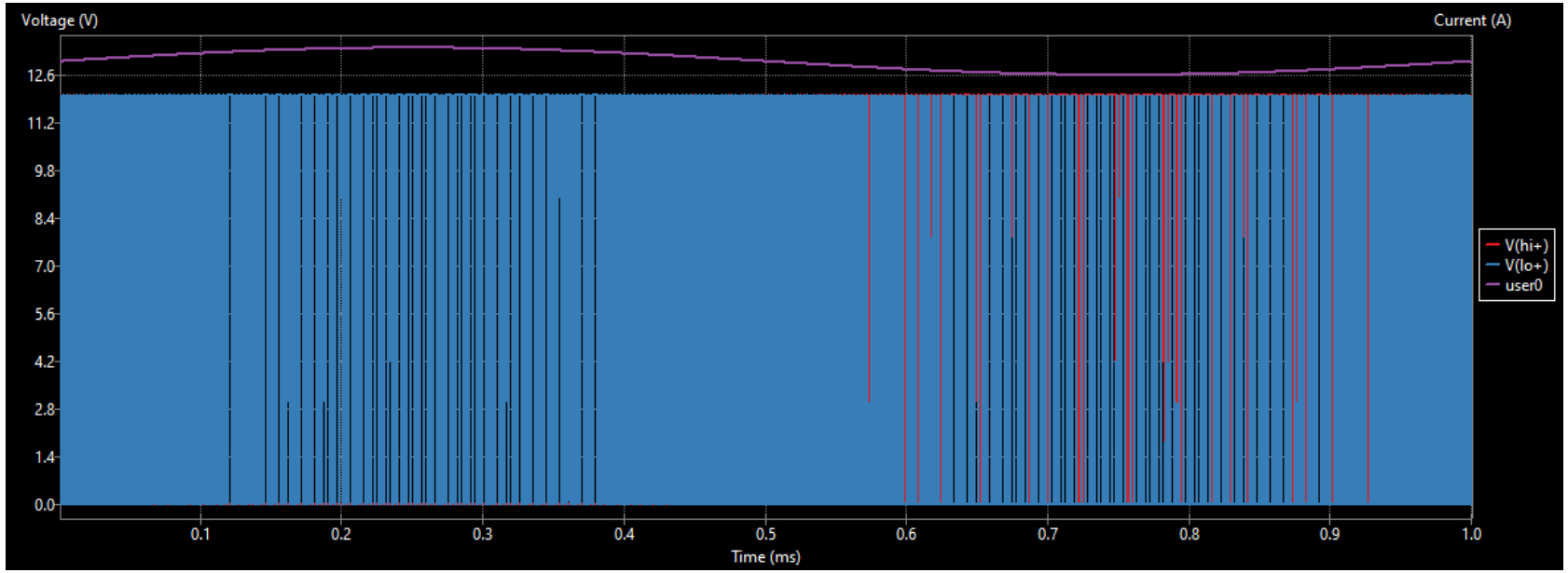
Started with netlist, difficult to read after some time

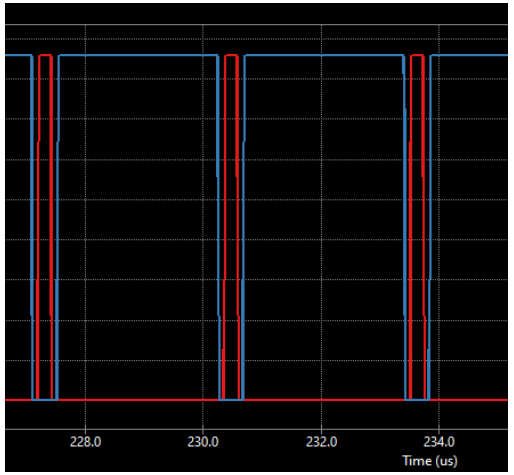
Much better to read, create netlist by exporting

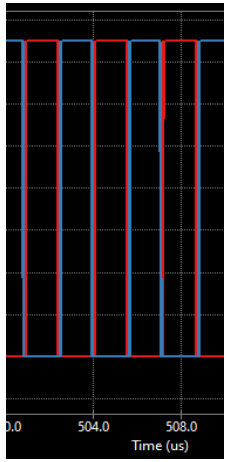


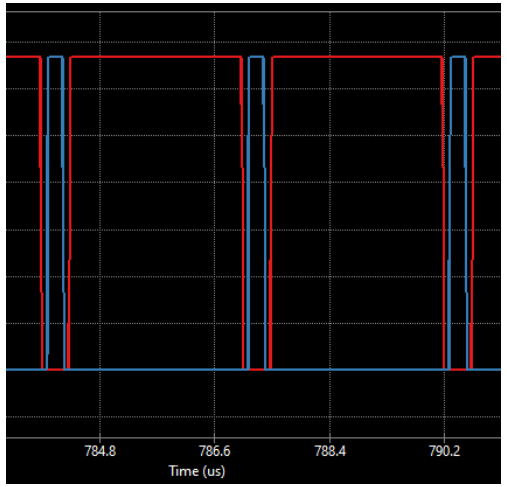
Example: Audio Driver

Simulation of the half bridge power transistor driver over one input period









Output
max, mid, and min
of sine input

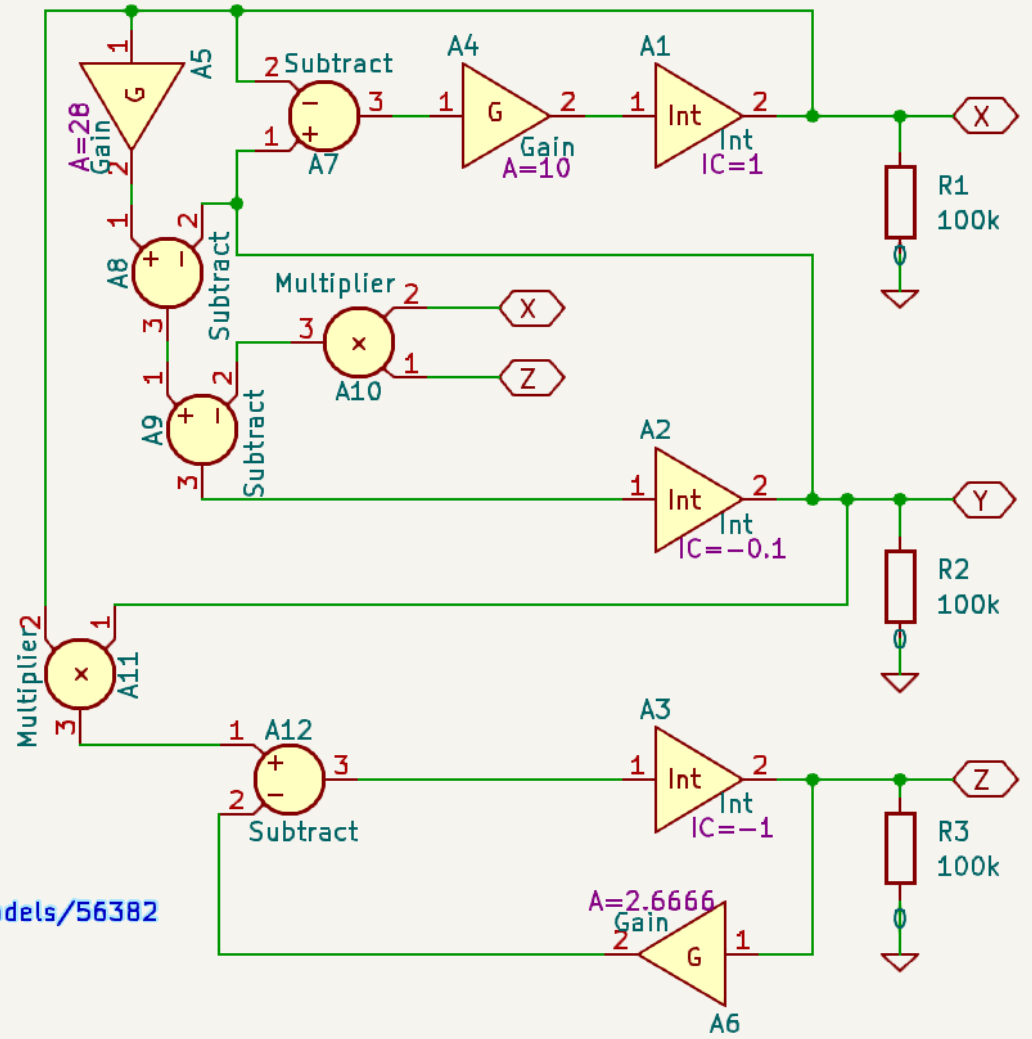
Example: Lorenz Attractor

$$\frac{dx}{dt} = 10(y - x)$$

$$\frac{dy}{dt} = 28x - y - xz$$

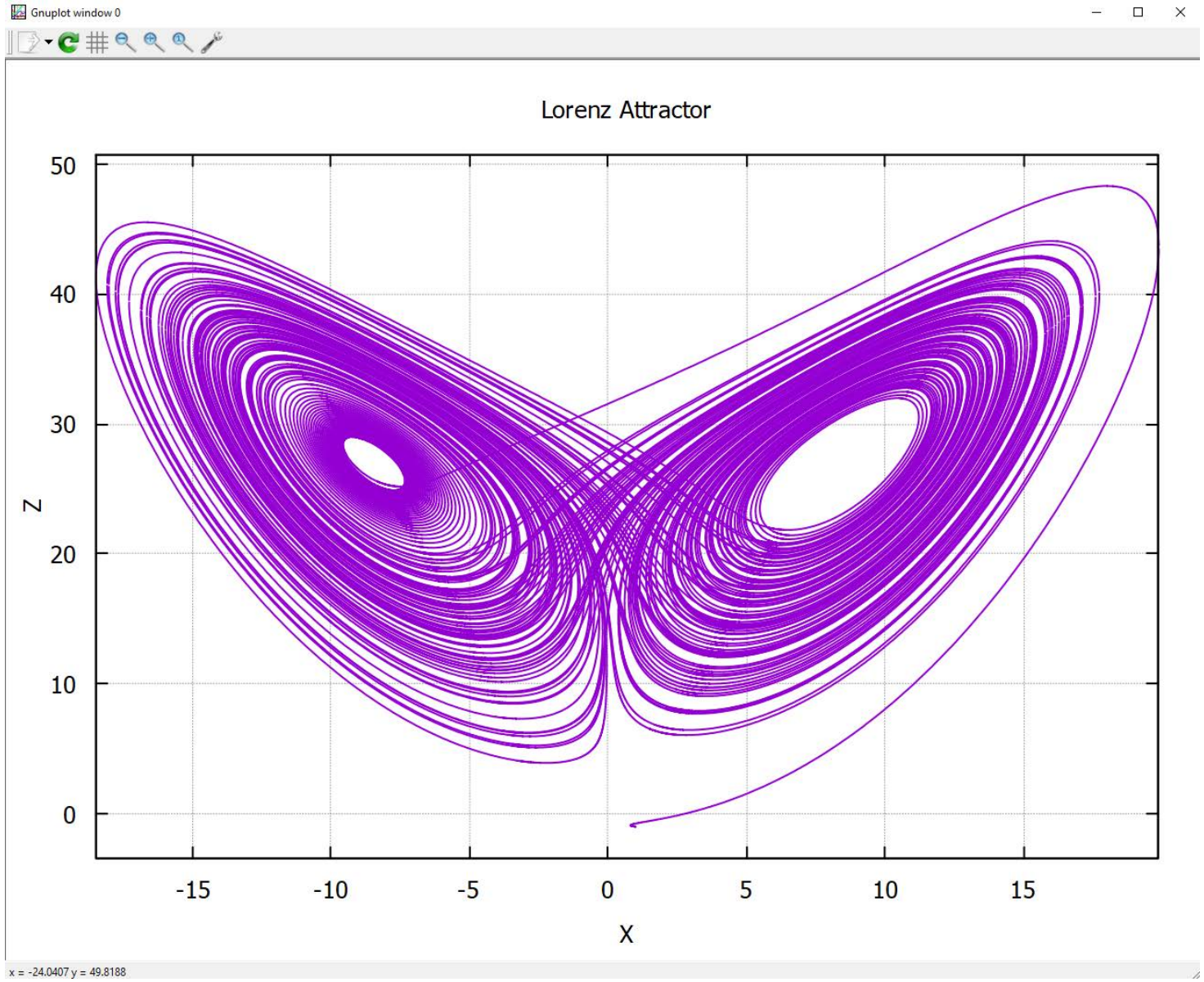
$$\frac{dz}{dt} = xy - \frac{8z}{3}$$

```
.control
set controlswait
set nolegend
gnuplot D:\gnuplot Z vs X title 'Lorenz Attractor' xlabel 'X' ylabel 'Z'
.endc
```



Author: cblas
https://forum.kicad.info/t/x-y-plot-lorenz-attractor-using-a-models/56382

Example: Lorenz Attractor



Independently developed by cblas, using his own symbols with the ngspice code models

An analog computer, solving three dependend differential equations, simulated on a digital machine.

Uses gnuplot for the X-Z plot

What's next in ngspice?

Some ideas, some more or less fixed plans, some actual activities:

- Improve RF capability by adding Harmonic Balance ongoing
- Support for reliability and degradation simulation ongoing
- Simulation of transient noise
- Improve usability of KiCad/ngspice graphics interface ongoing
- Support for ngspice mixed signal building blocks in Eeschema ongoing
- Enhance compatibility with LTSPICE models (A devices?)
- Test and improve the Verilog/VHDL/C interfaces
- Use AI for faster model building and simulation

Current development branch at git: https://sourceforge.net/p/ngspice/ngspice/ci/pre-master-45/tree/

Support

Ngspice discussion forums https://sourceforge.net/p/ngspice/discussion/

Ngspice Manual https://ngspice.sourceforge.io/docs.html

KiCad-Info forum https://forum.kicad.info/

KiCad Manual (V8) https://docs.kicad.org/8.0/en/eeschema/eeschema.html#simulator

Tutorials https://ngspice.sourceforge.io/tutorials.html

Models, model parameters https://ngspice.sourceforge.io/modelparams.html

Simulation examples
https://sourceforge.net/p/ngspice/ngspice/ci/master/tree/examples/
https://forum.kicad.info/t/simulation-examples-for-kicad-eeschema-ngspice/34443
https://forum.kicad.info/t/more-simulation-examples-for-kicad-eeschema-ngspice/45546
https://forum.kicad.info/t/simulation-with-xspice-code-models/56384/2