

Project name: "ASIC design with open source tools"

Developing of an low resolution ADC. As example an 4-Bit ADC with conversion of the thermometer code into binary code.

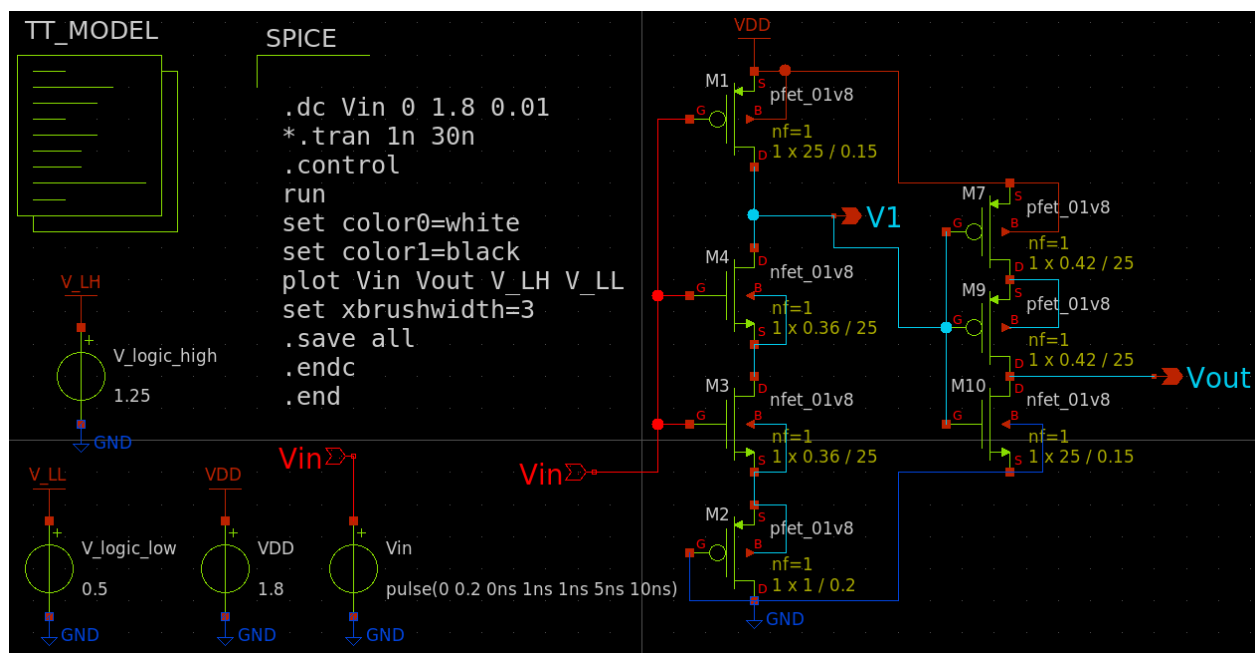
Steps:

A small CMOS inverter with xschem and the sky130a technology (PDK) and simulate the circuit with ngspice:

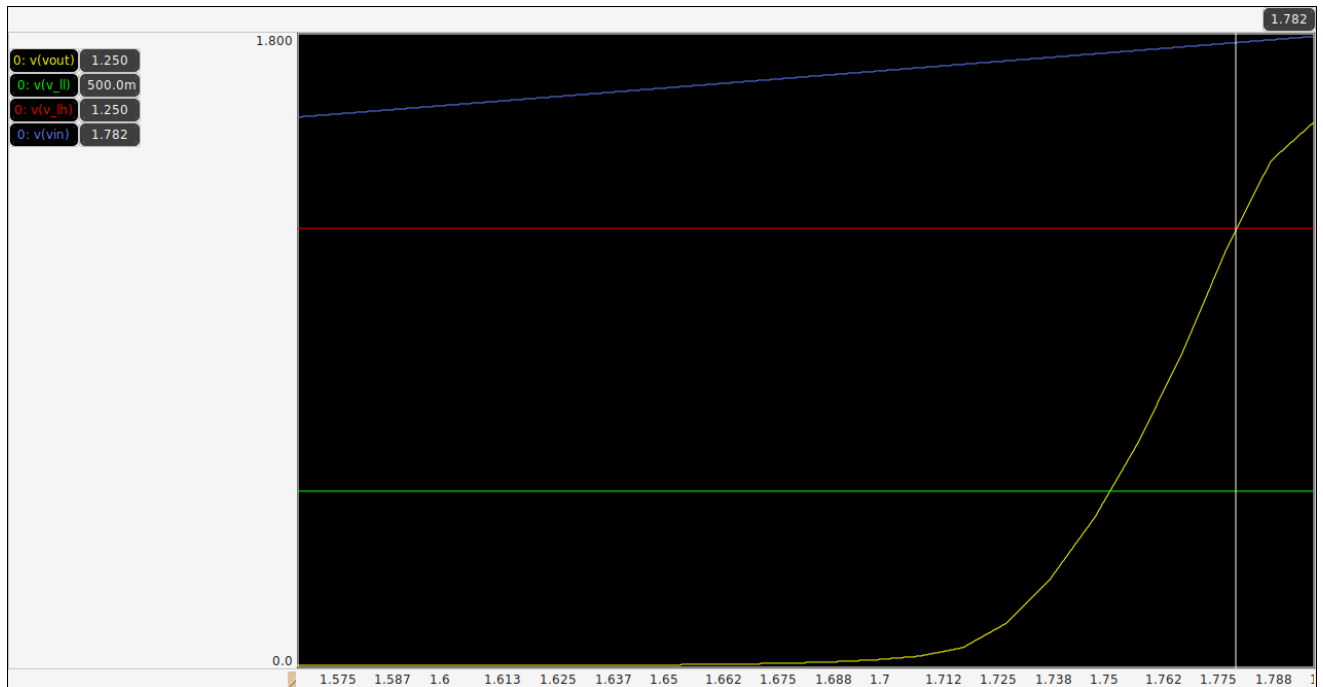
Binary Logic level assumption

https://en.wikipedia.org/wiki/Logic_level

Technology	L Voltage	H Voltage	note
CMOS	0 to 30% of Vdd	70% of Vdd to Vdd	Vdd = Supply voltage
	≈0.5 Volt	≈1.25 Volts	Vdd = 1.8 Volts



The above figure is the schematic for 15th threshold point of the analogue part of the ADC. The plot is as follows:



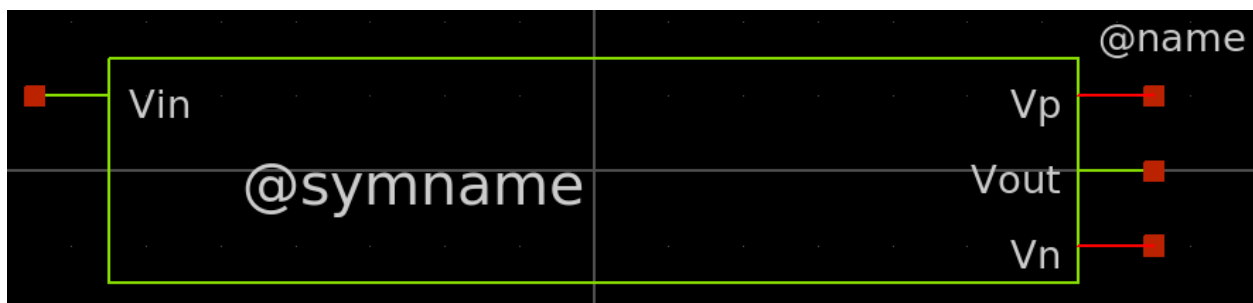
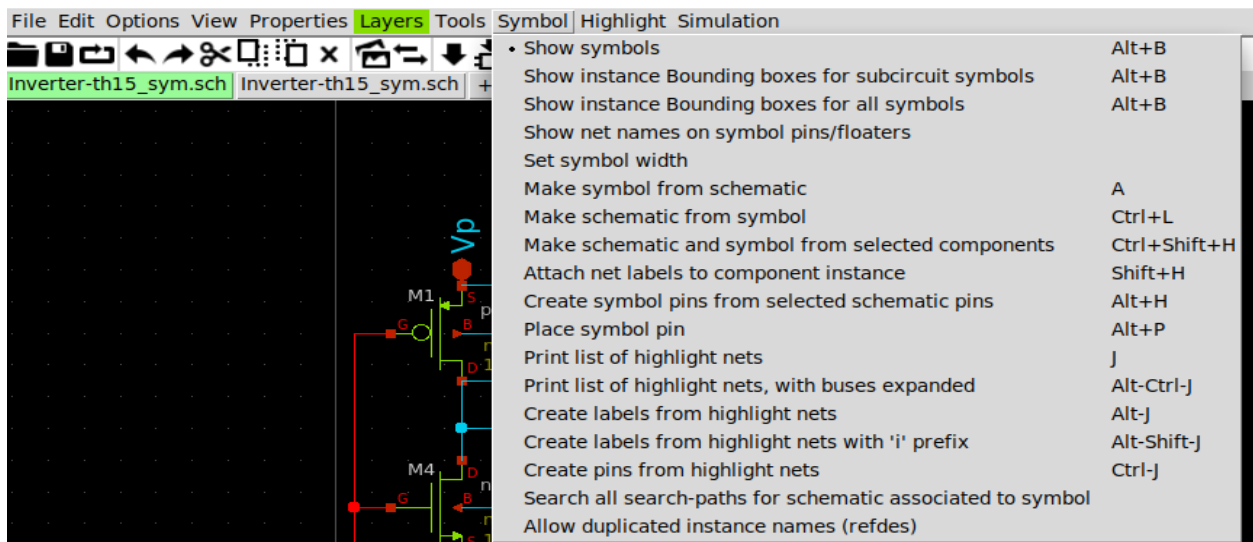
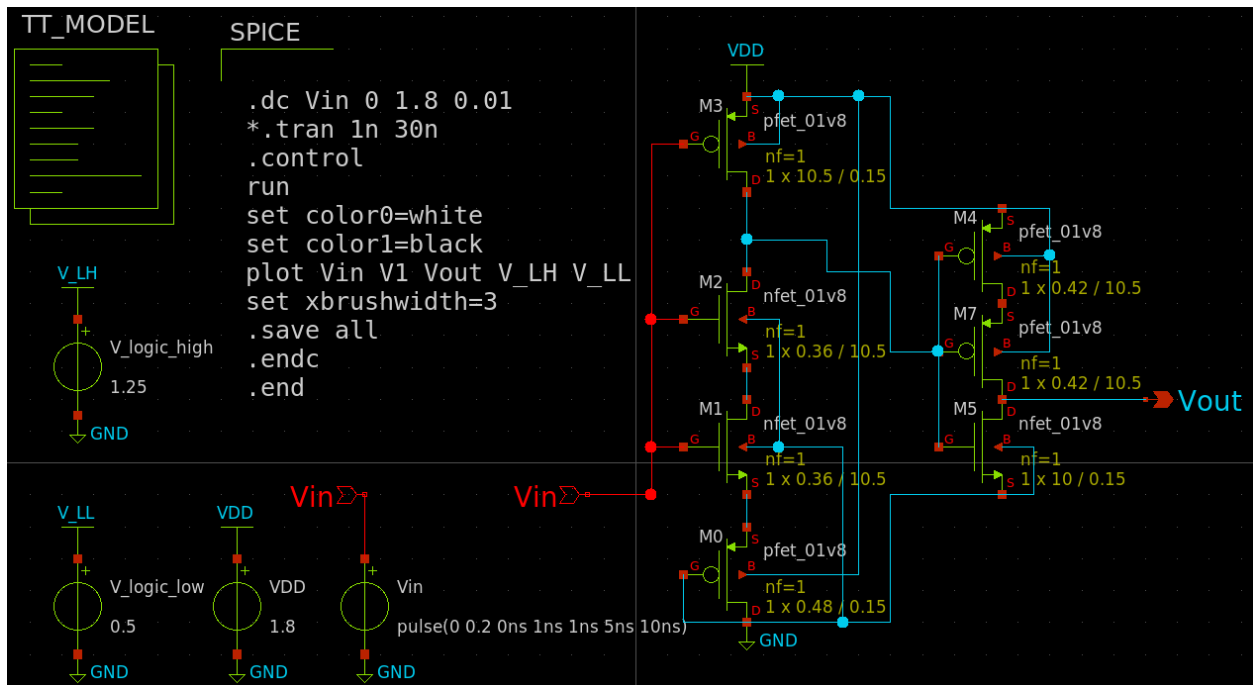
The inverter will switch from logic-low to logic-high when input voltage exceeds 1.782 volts. So, for this inverter the threshold is 1.782 volts.

For a 4-bit ADC we need 16 voltage level including zero (0~1.782 volts). The quantization should be $1.782/15=0.1188$ volts.

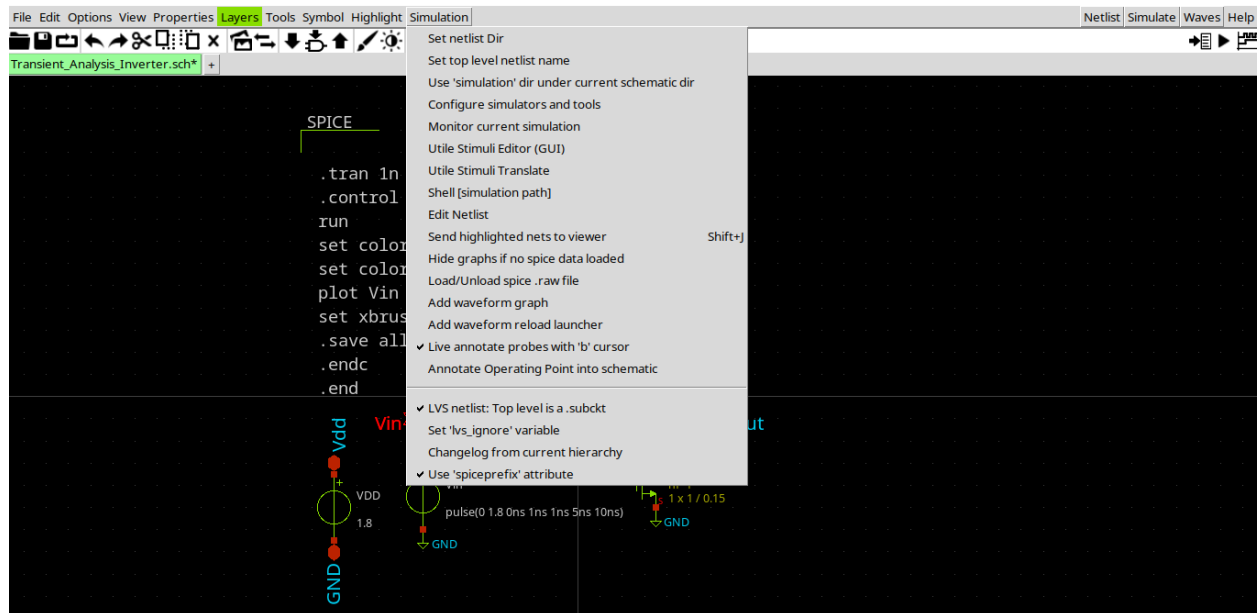
Vdd = 1.8 V

Th. 15: 1.782

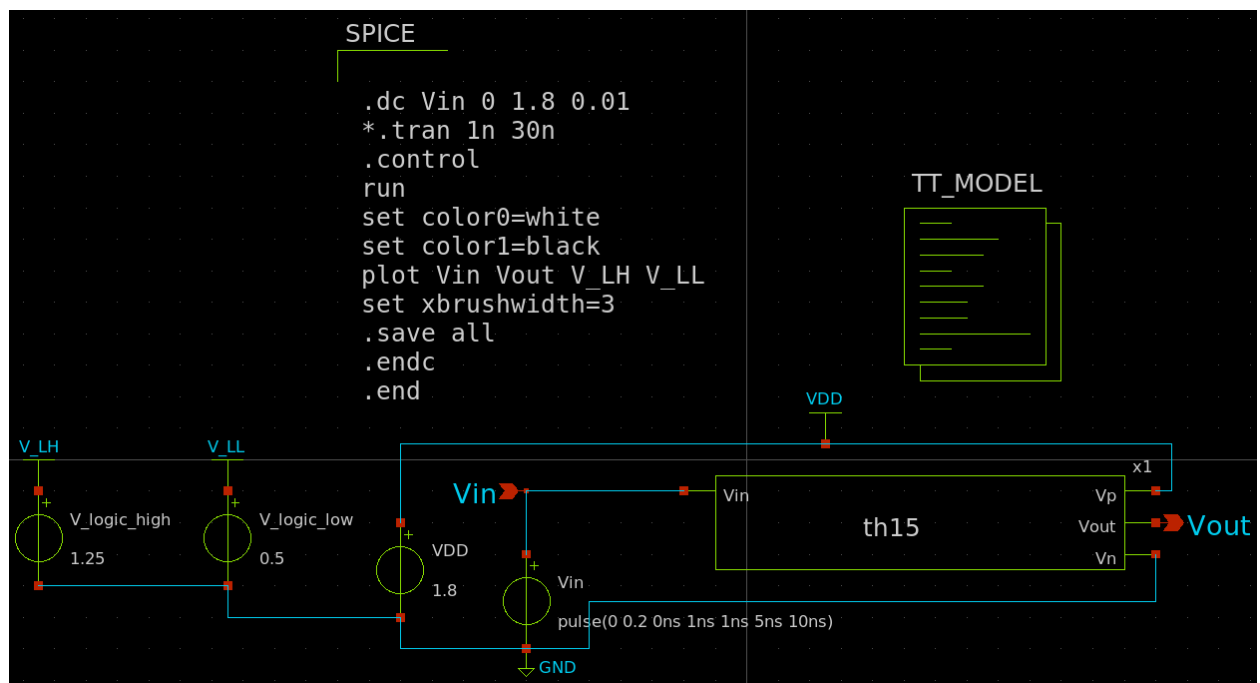
For calculation the schematic and the plot already mentioned earlier. Remove all source to make symbol from schematic. As the symbol will have to be connected to Vdd and GND, those pins are replaced with Vp and Vn accordingly.



Magic=>before importing netlist to ngspice check LVS net

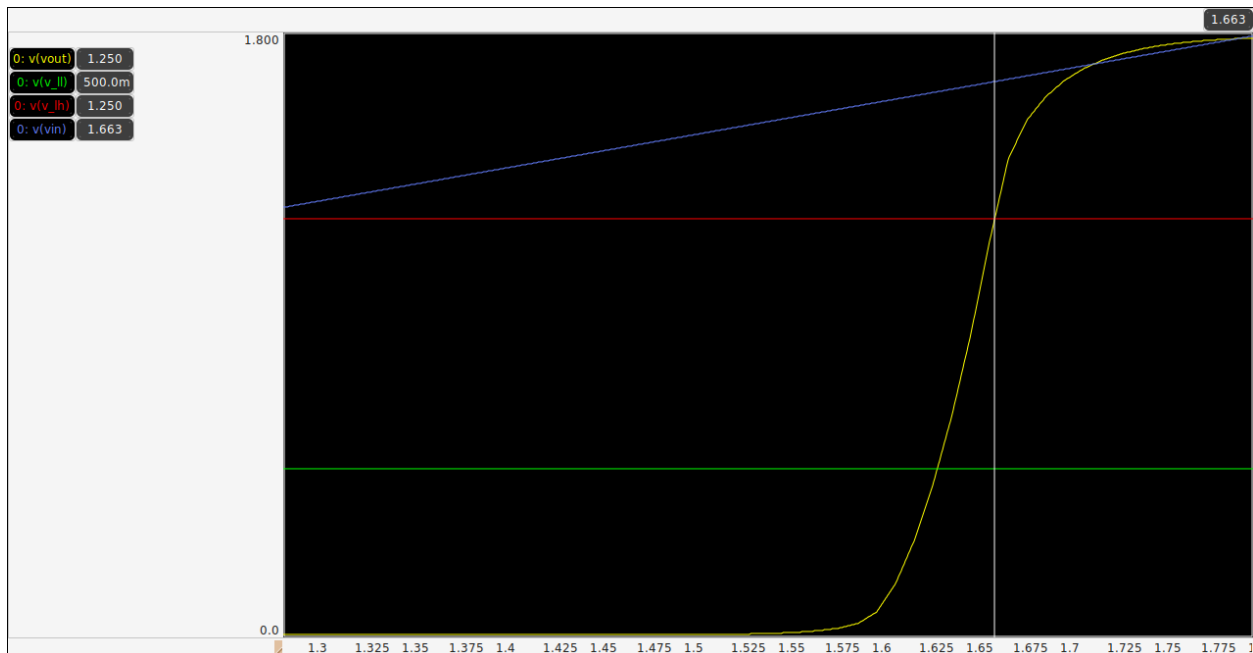
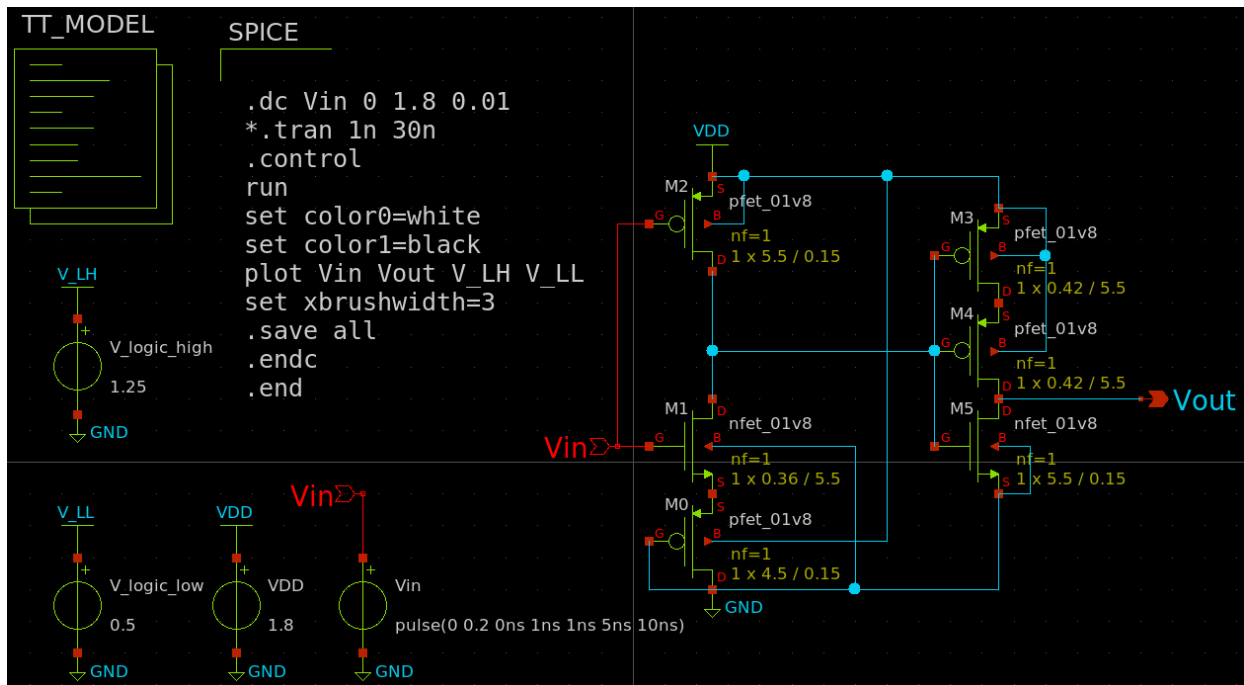


The simplified schematic is as follows:



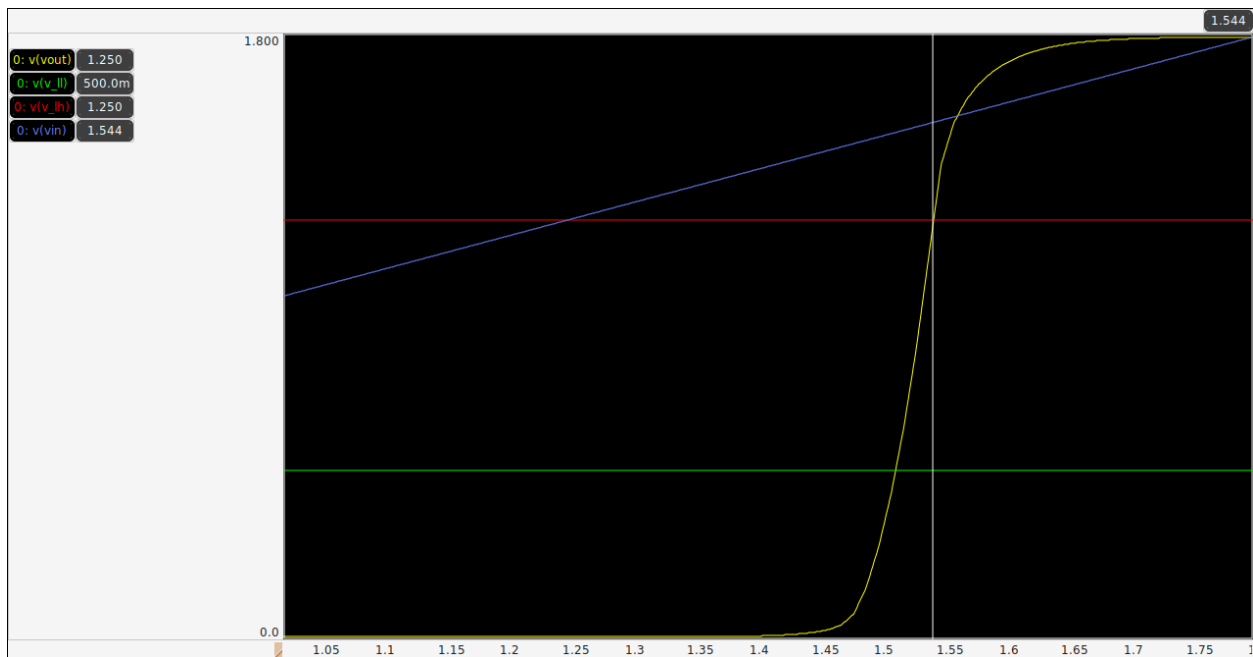
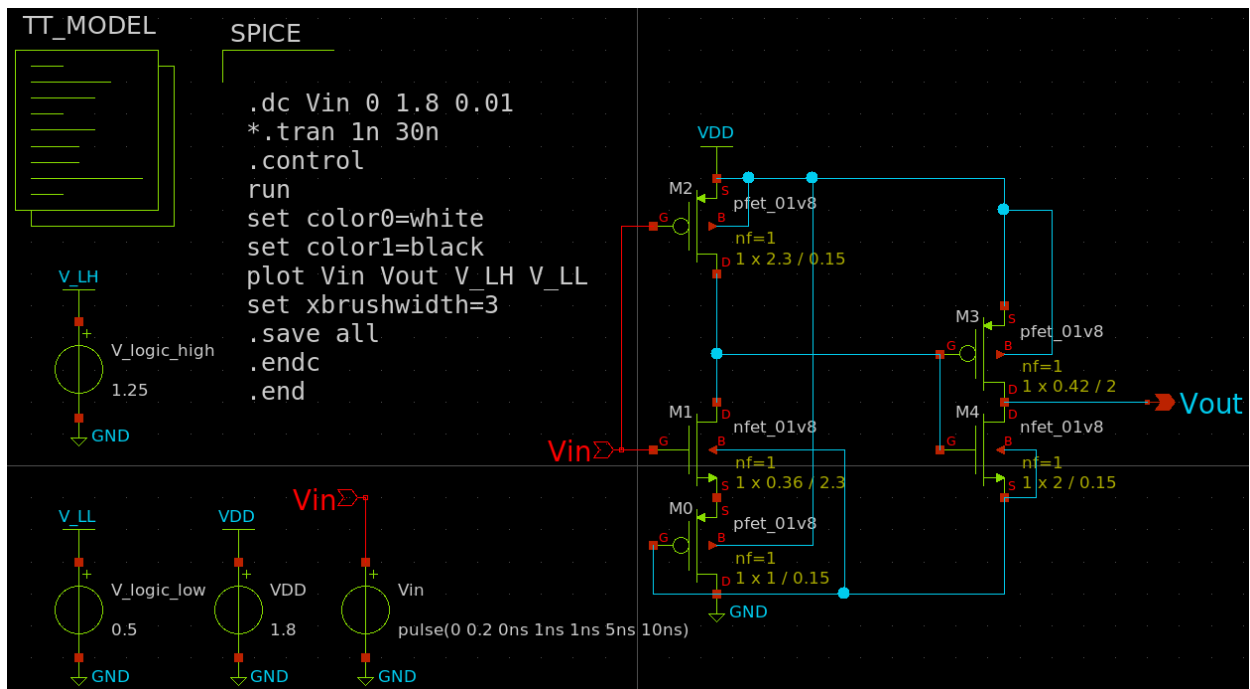
The plot is similar to the previous figure.

Th. 14: **1.6632**

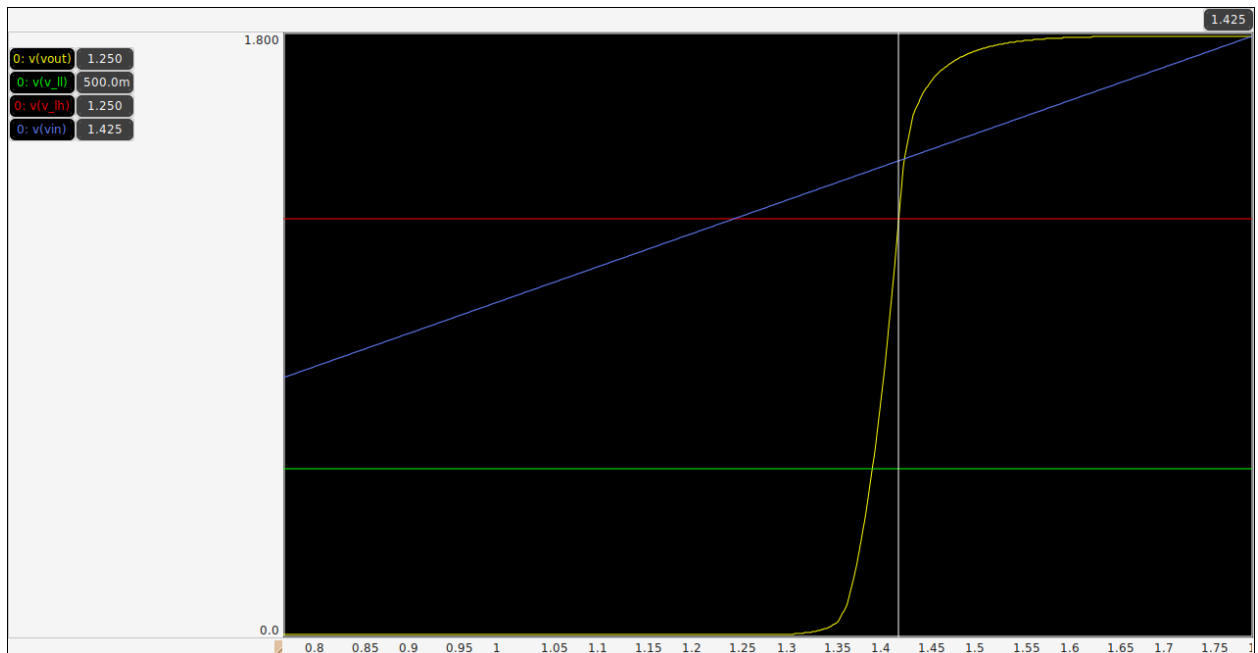
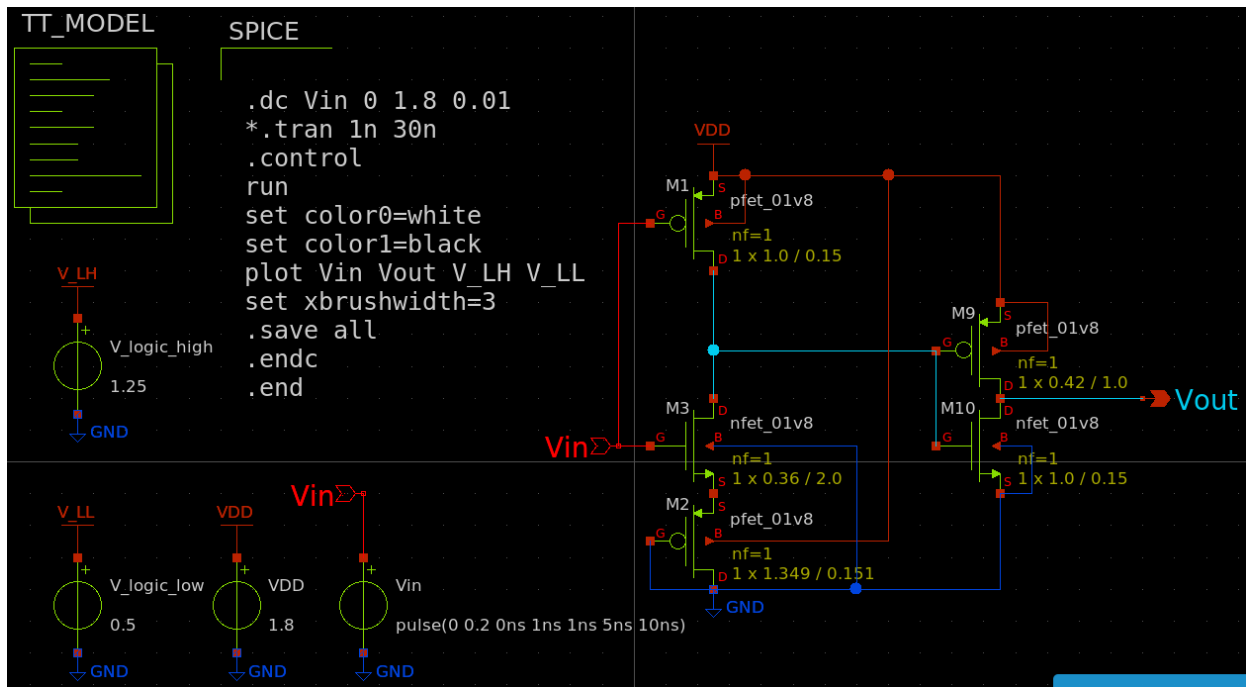


Rest of the figures are similar to the previous figures.

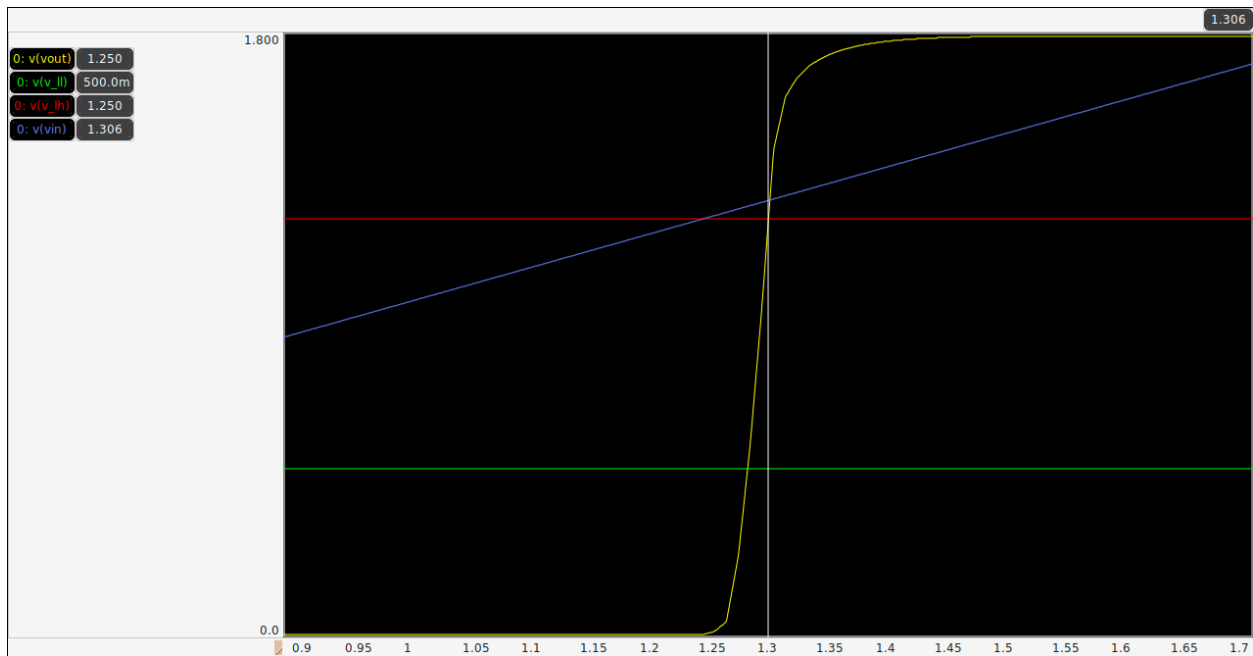
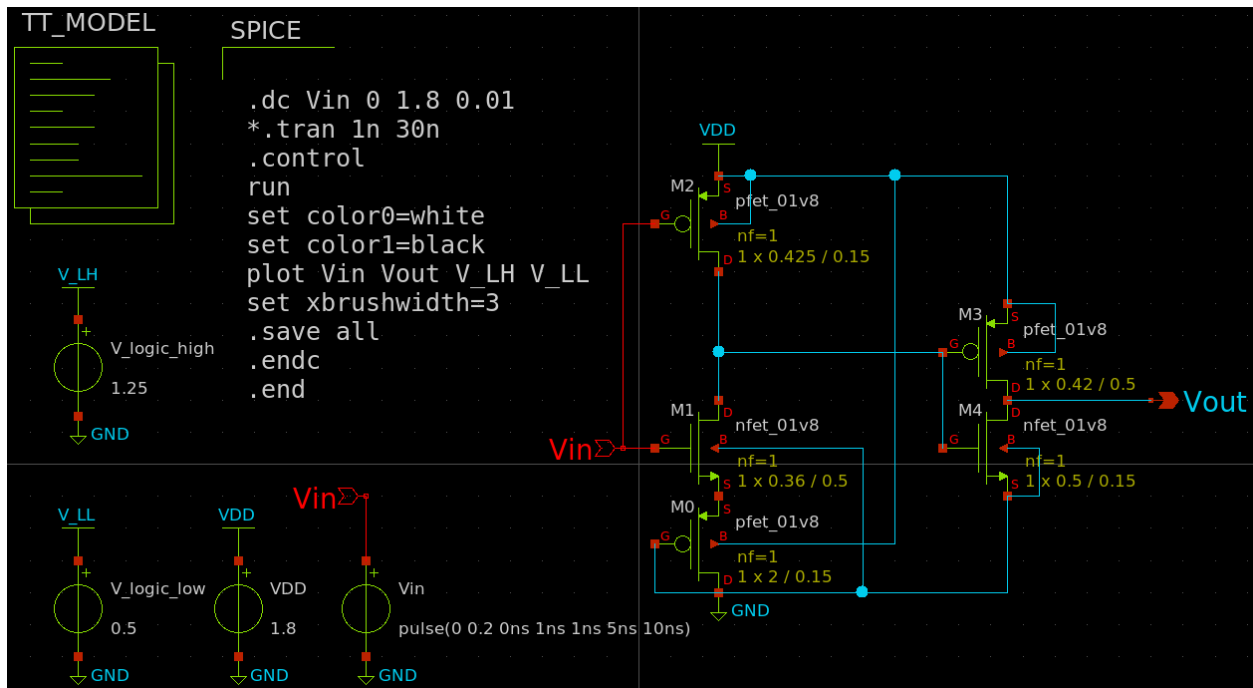
Th. 13: **1.5444:**



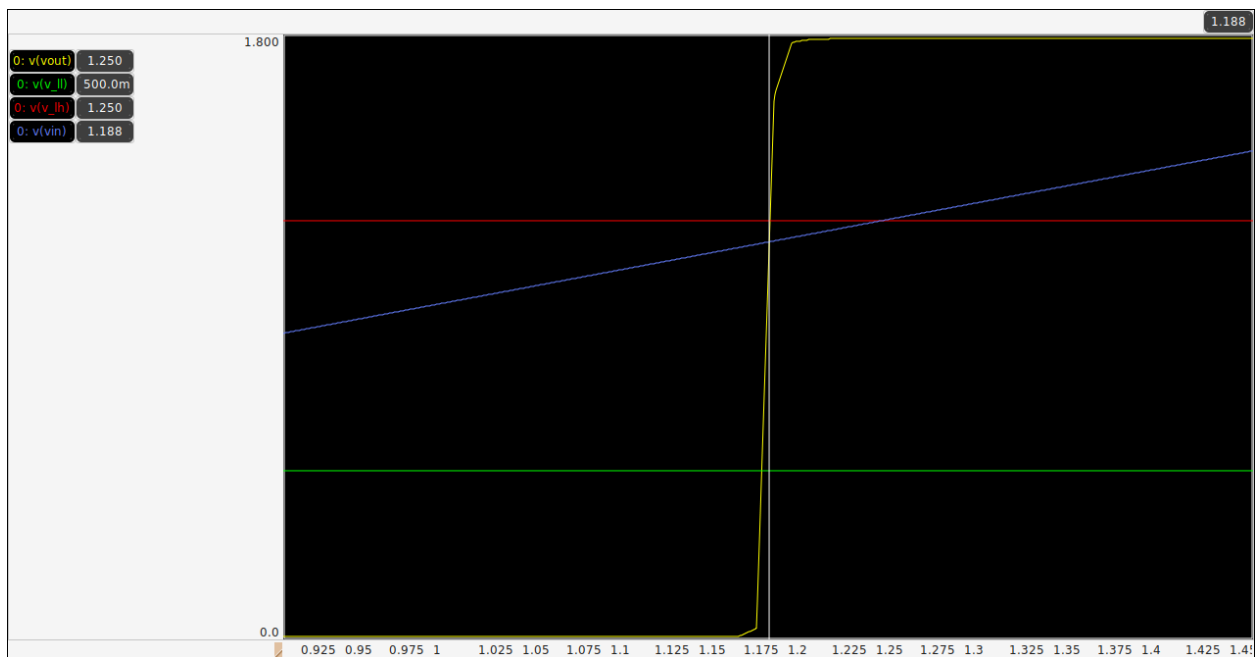
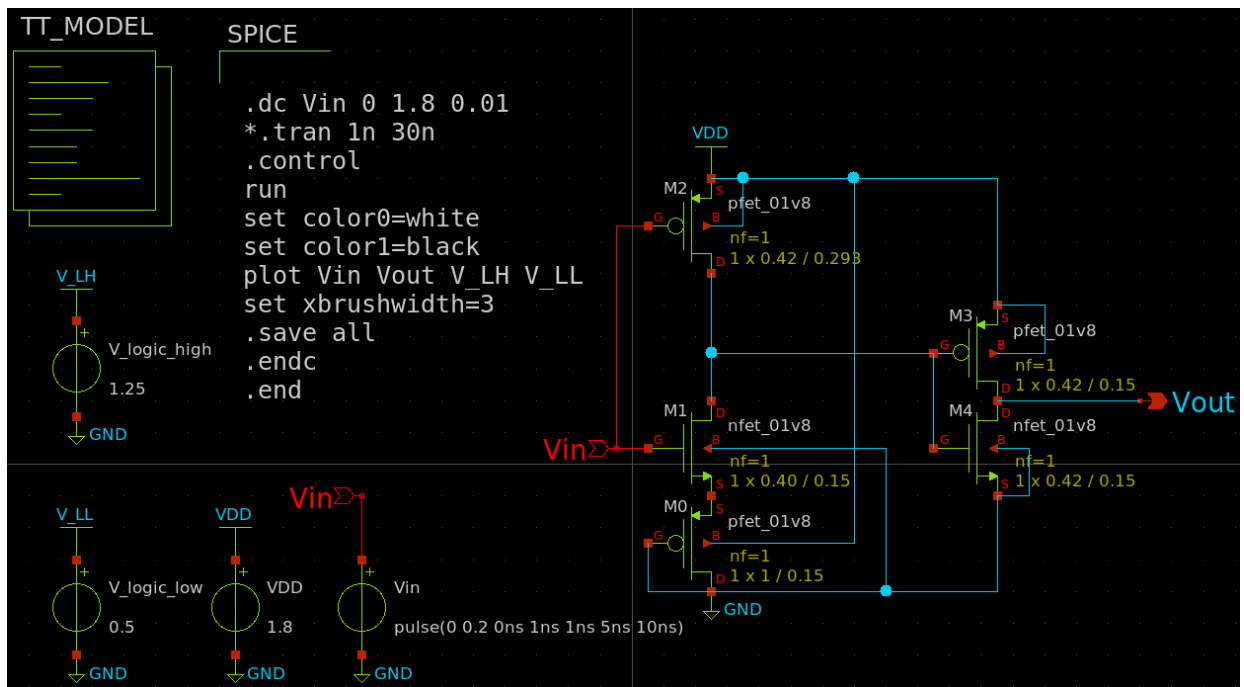
Th. 12: **1.4256**



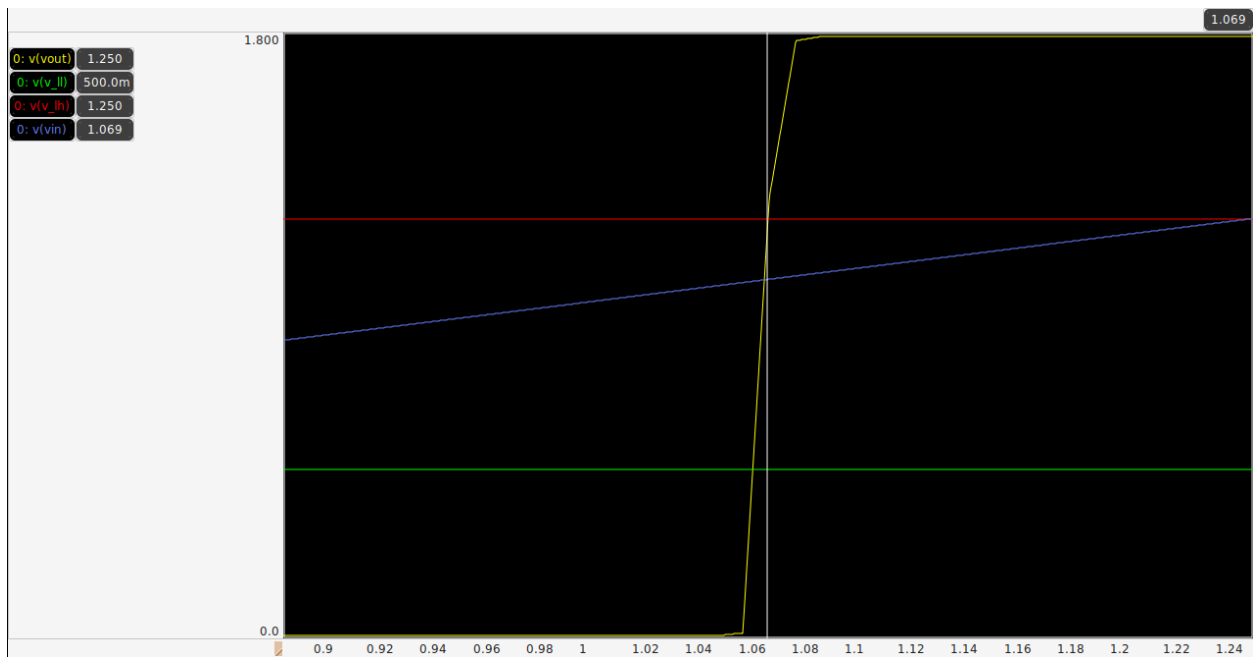
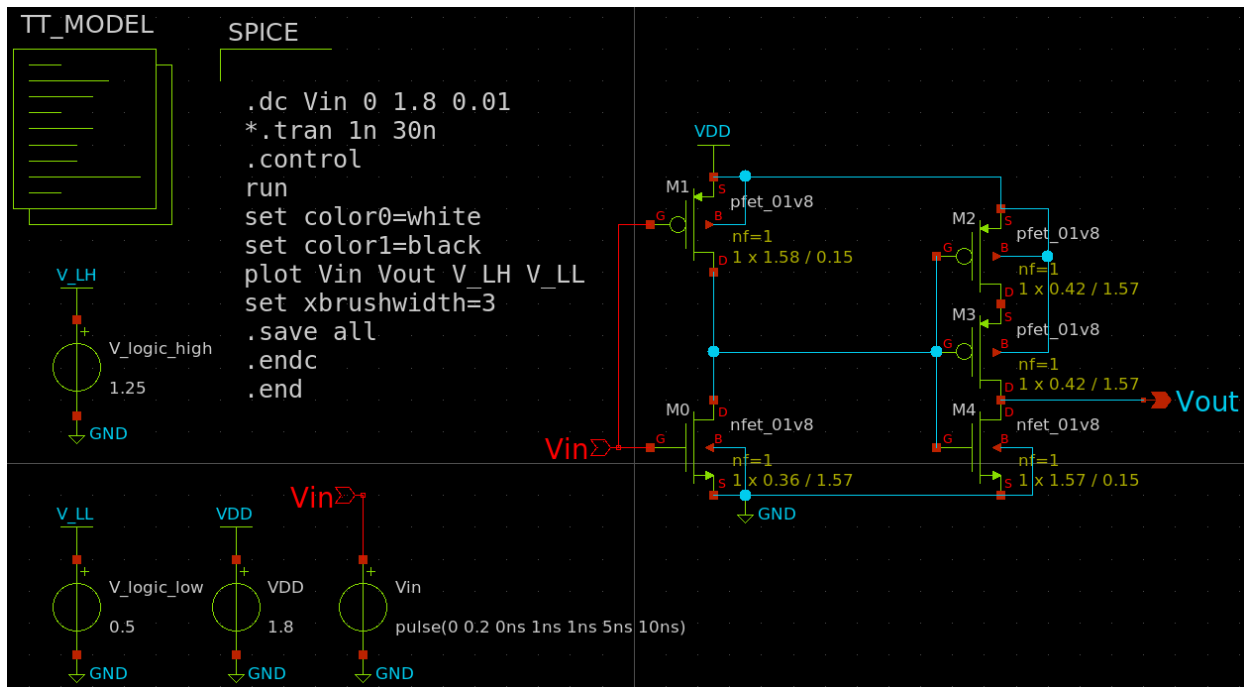
Th. 11: 1.3068



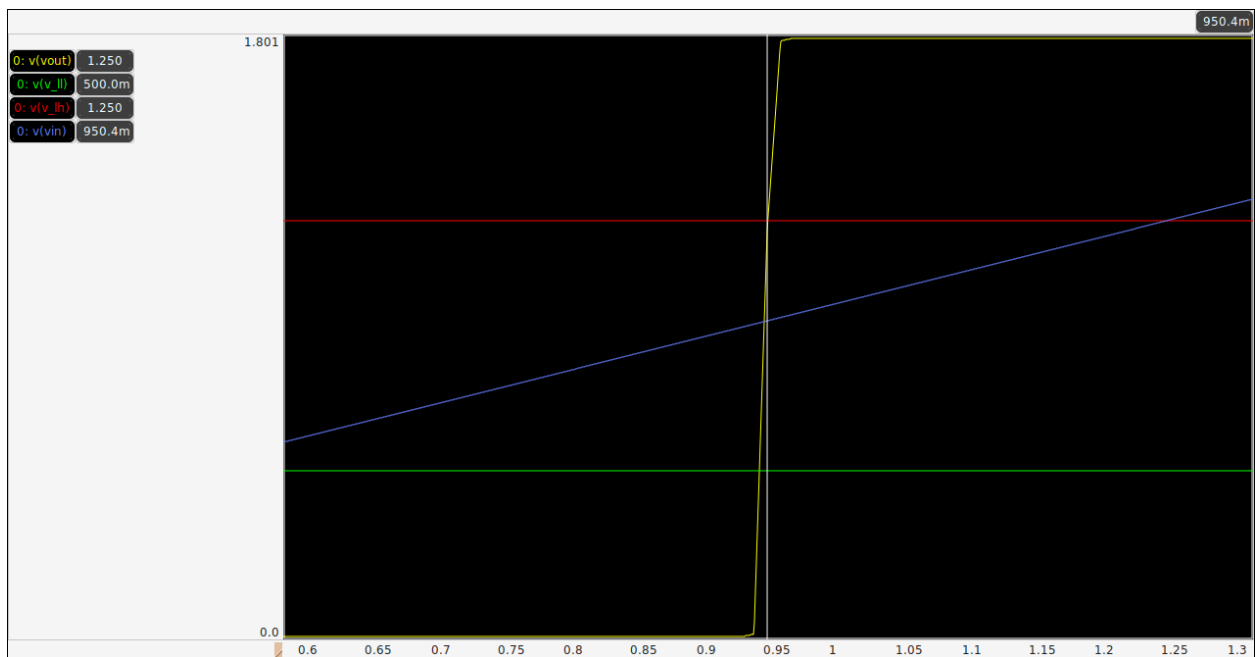
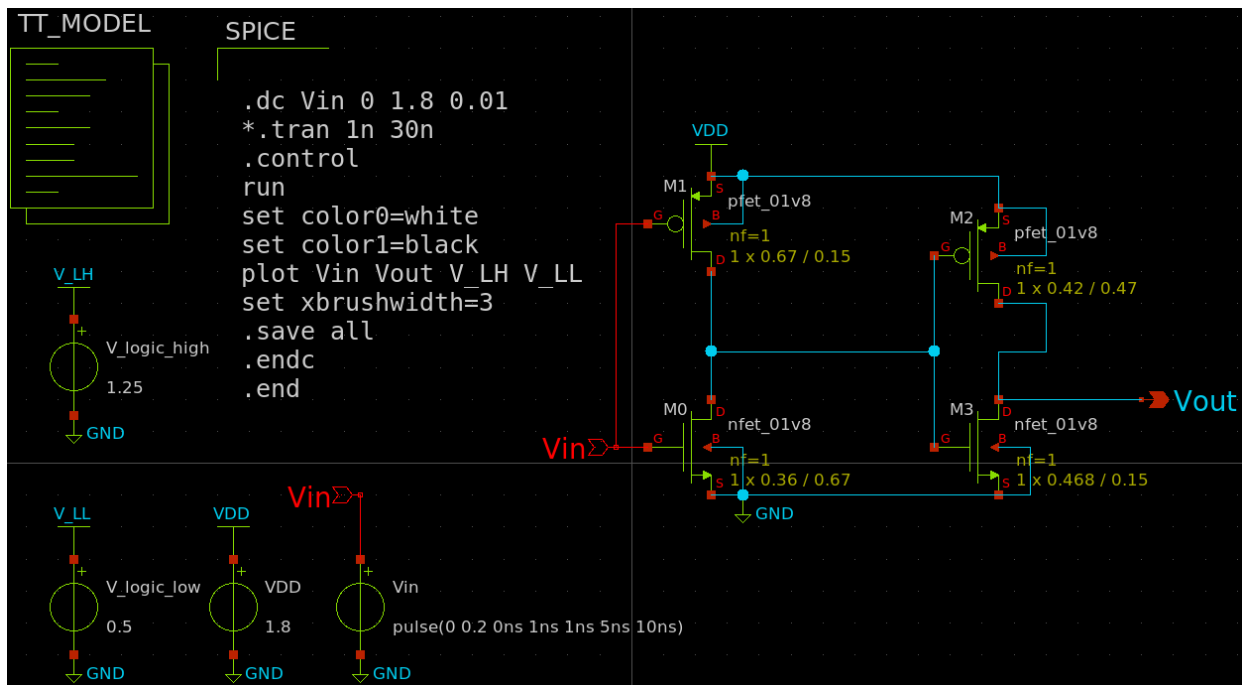
Th. 10: 1.188



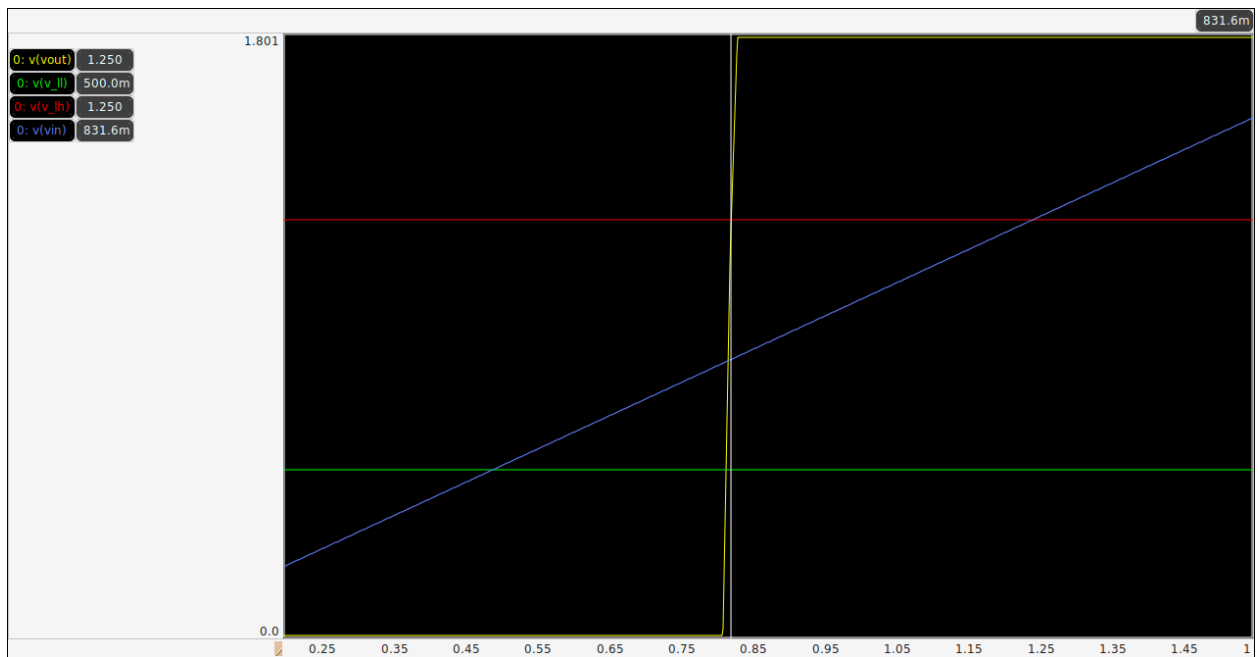
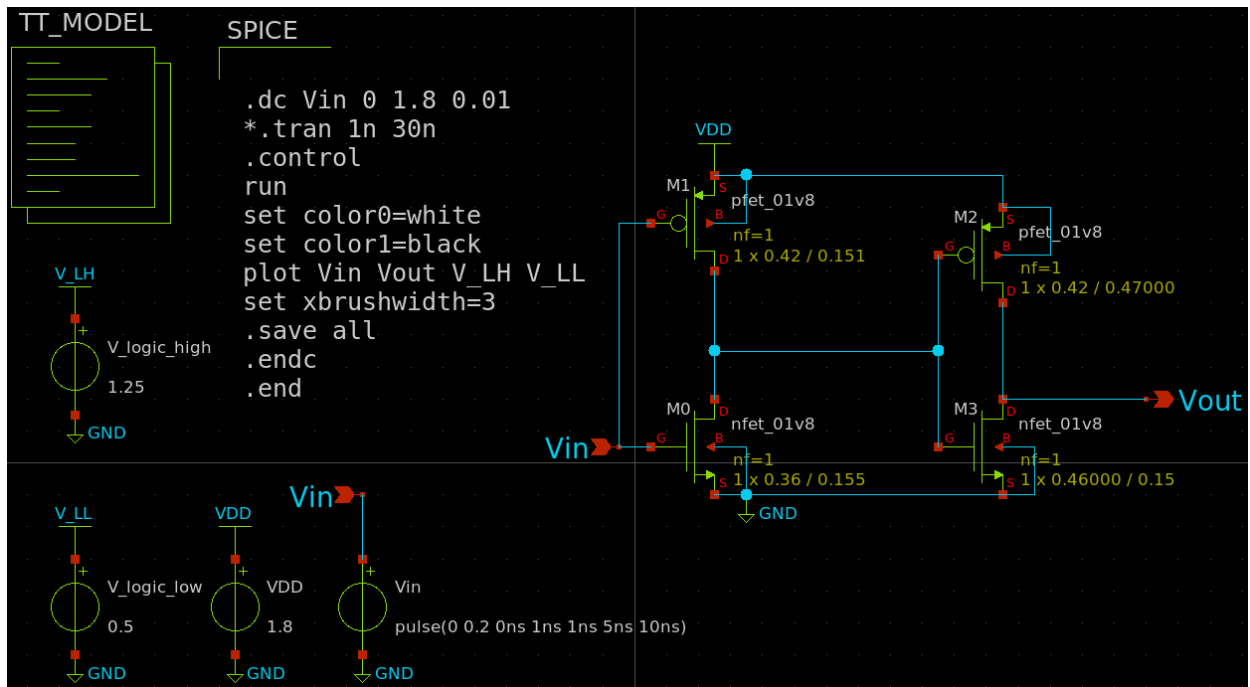
Th. 9: 1.0692



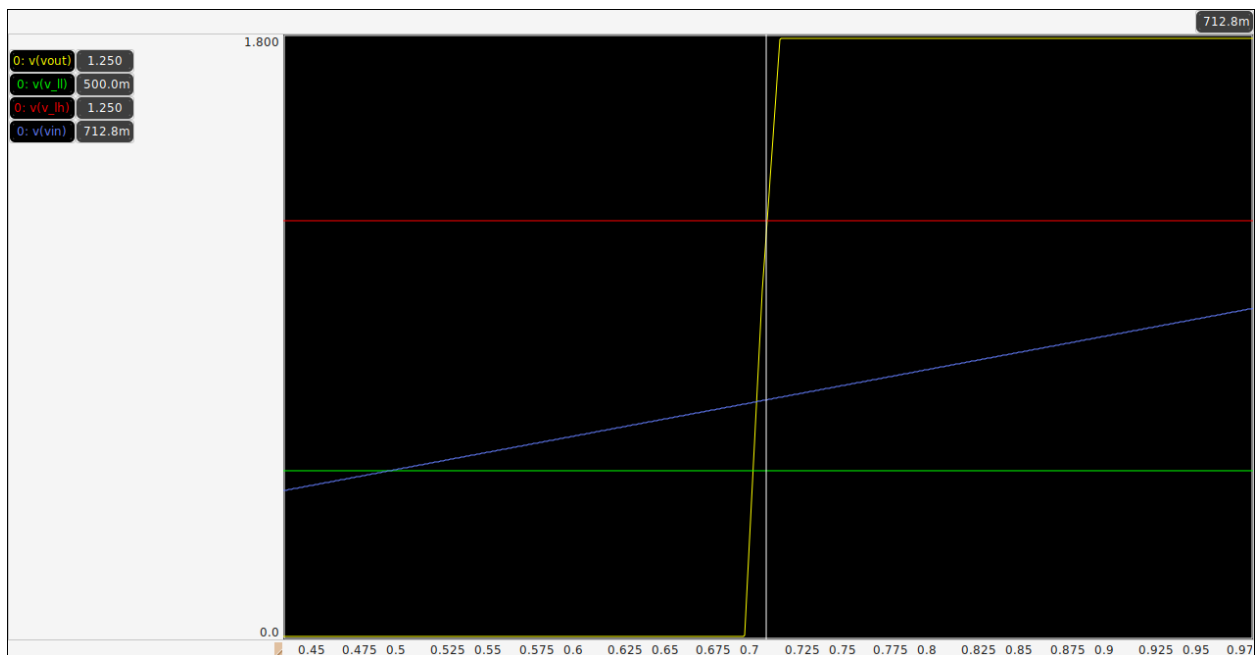
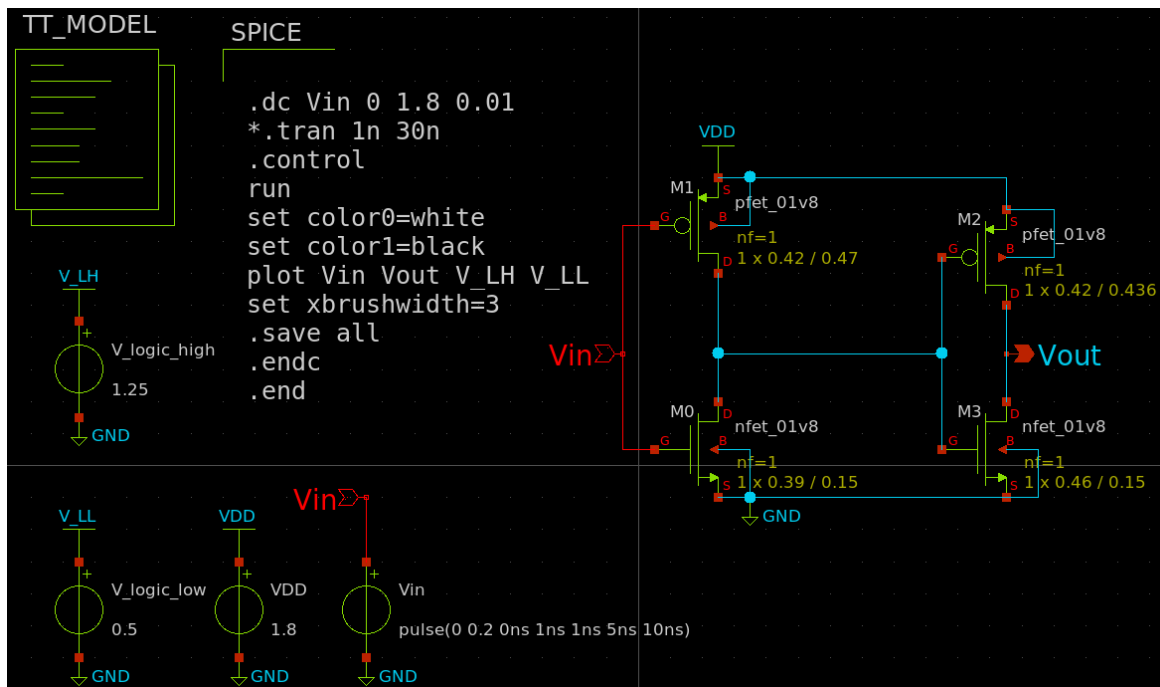
Th. 8: 0.9504



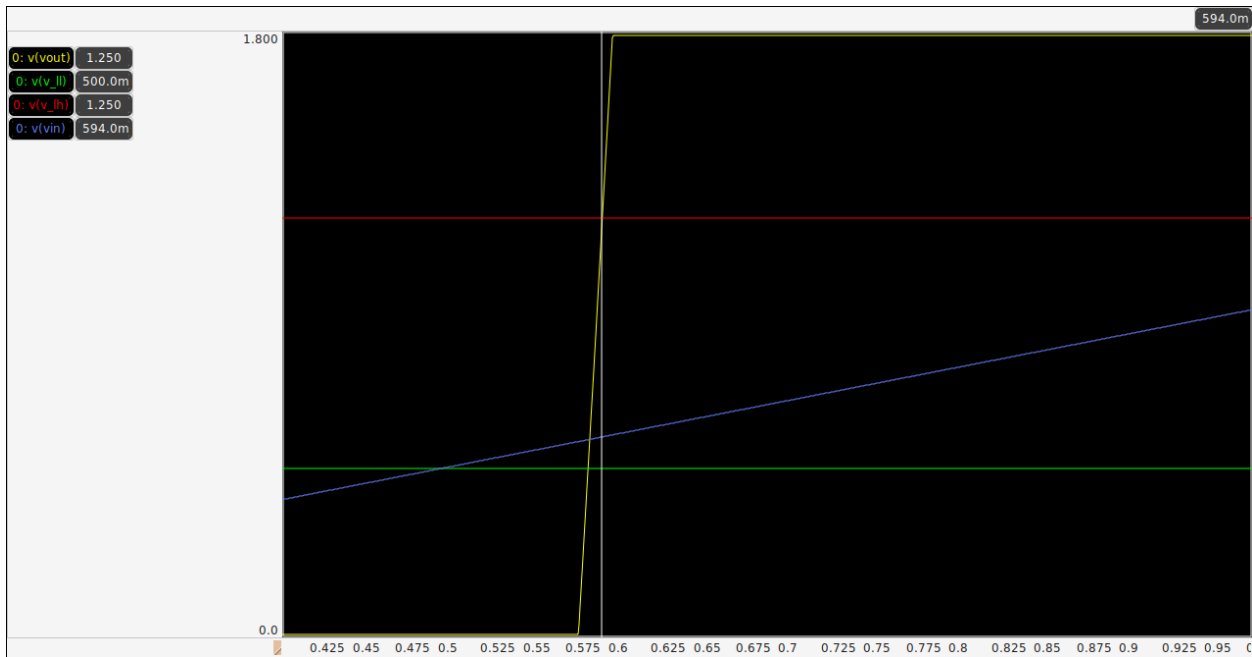
Th. 7: 0.8316



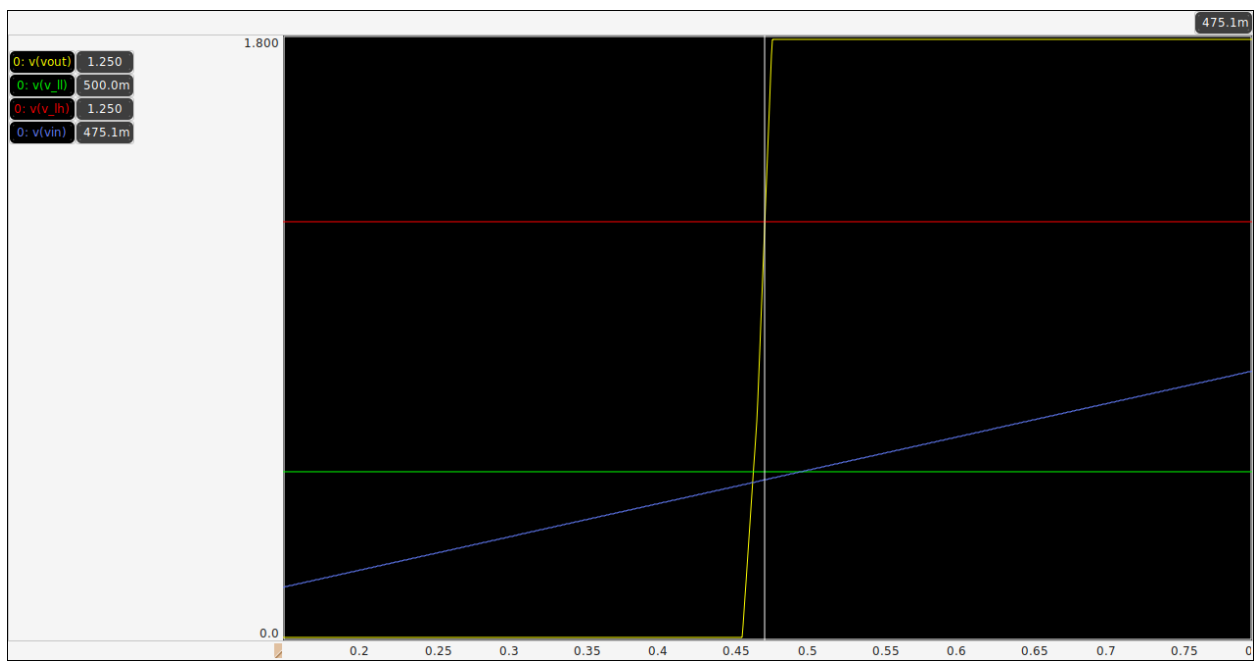
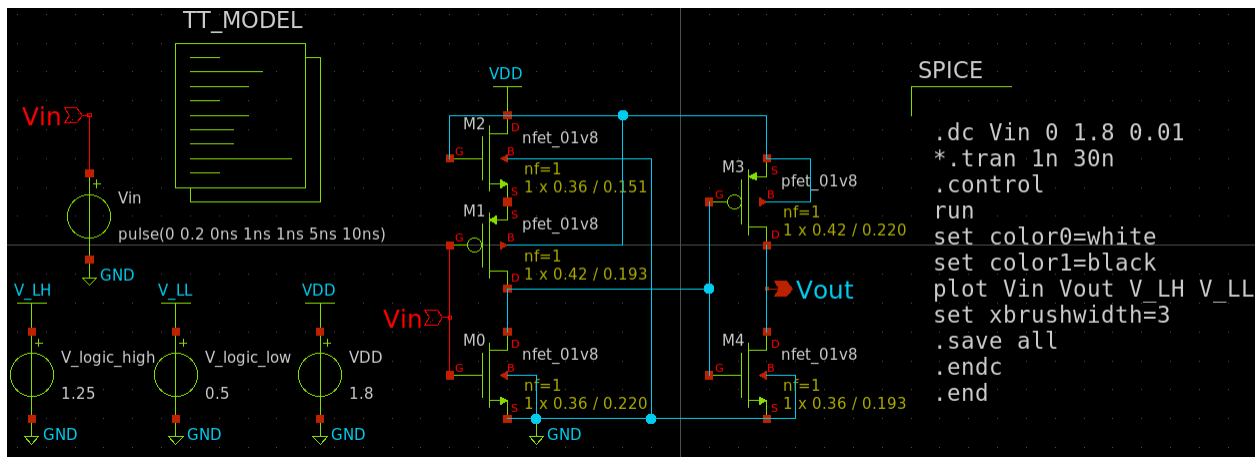
Th. 6: 0.7128



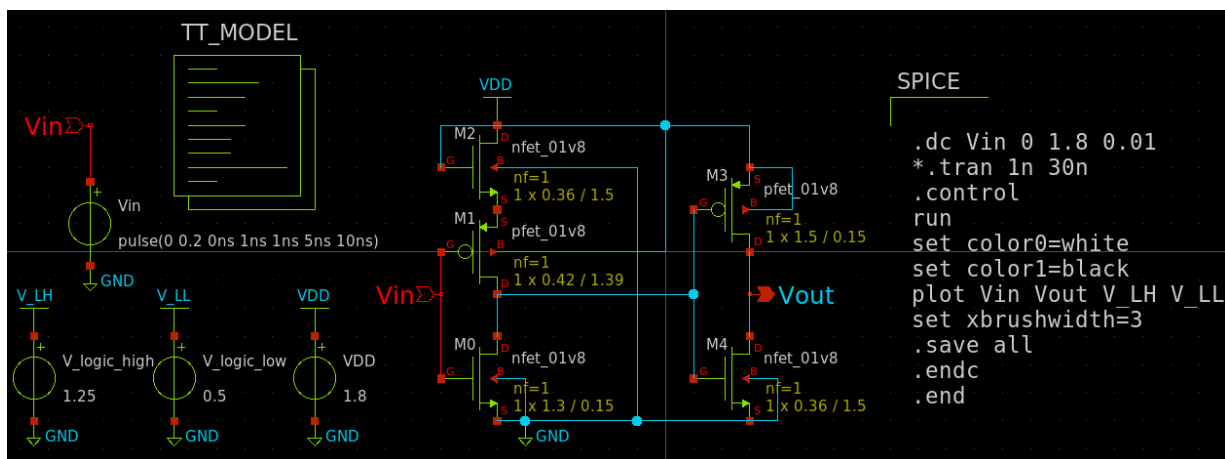
Th. 5: 0.594



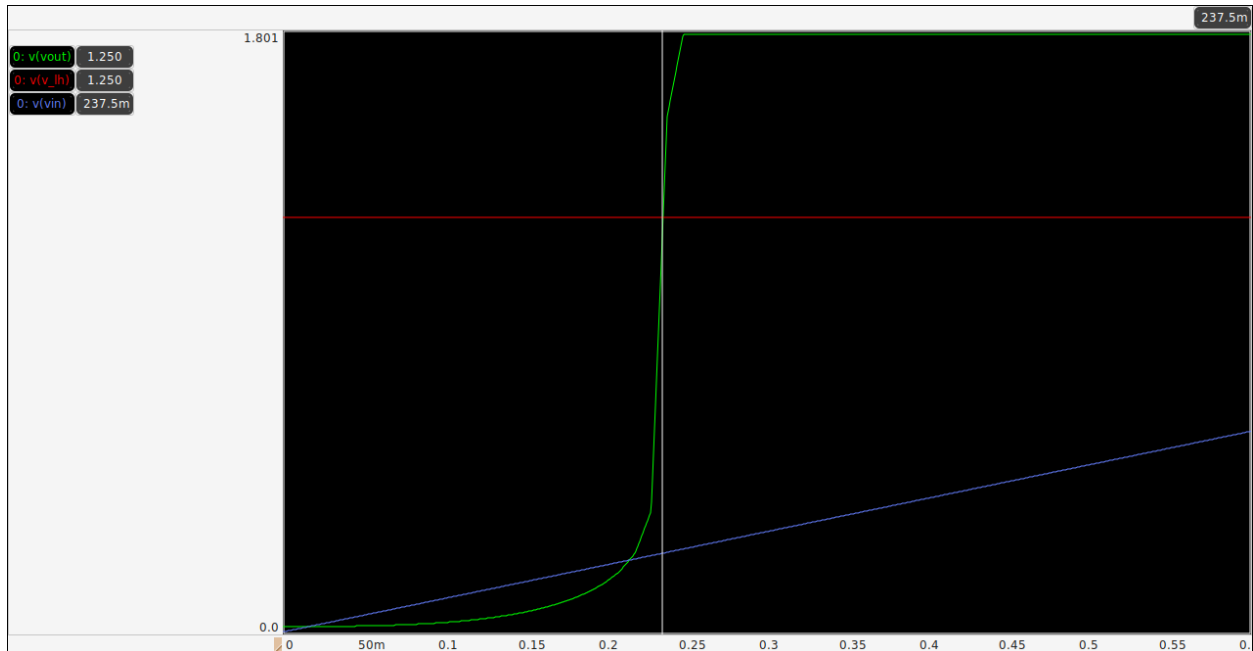
Th. 4: 0.4752



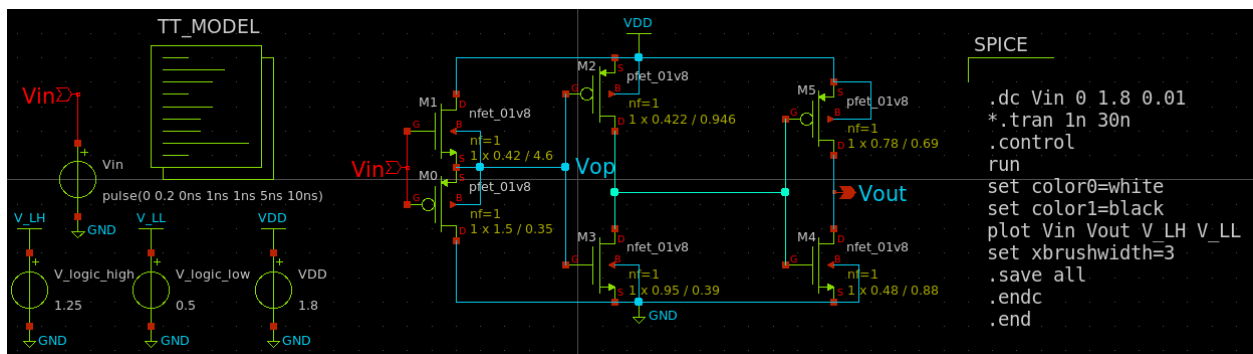
Th. 3: 0.3564



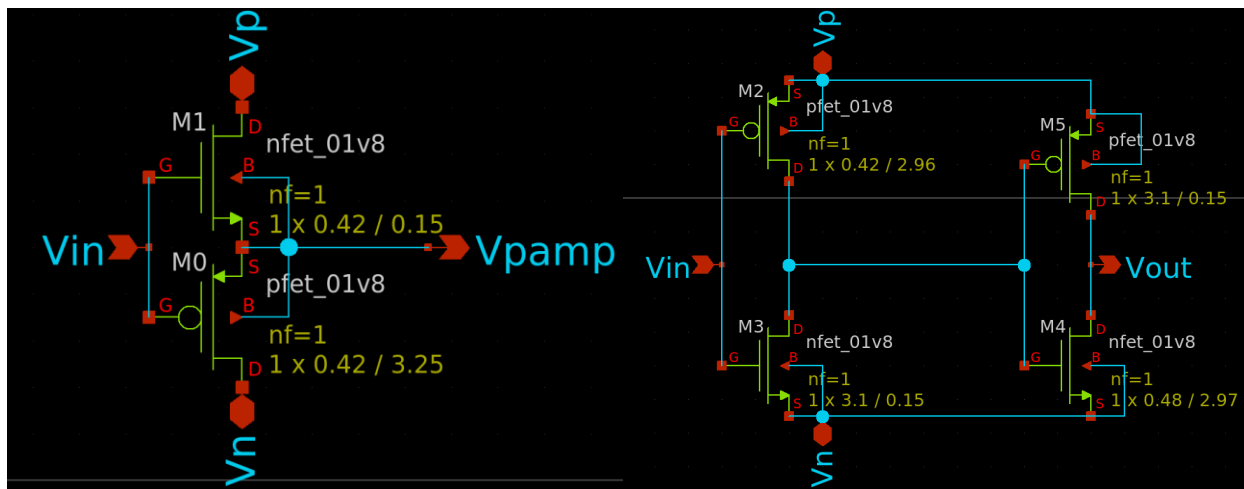


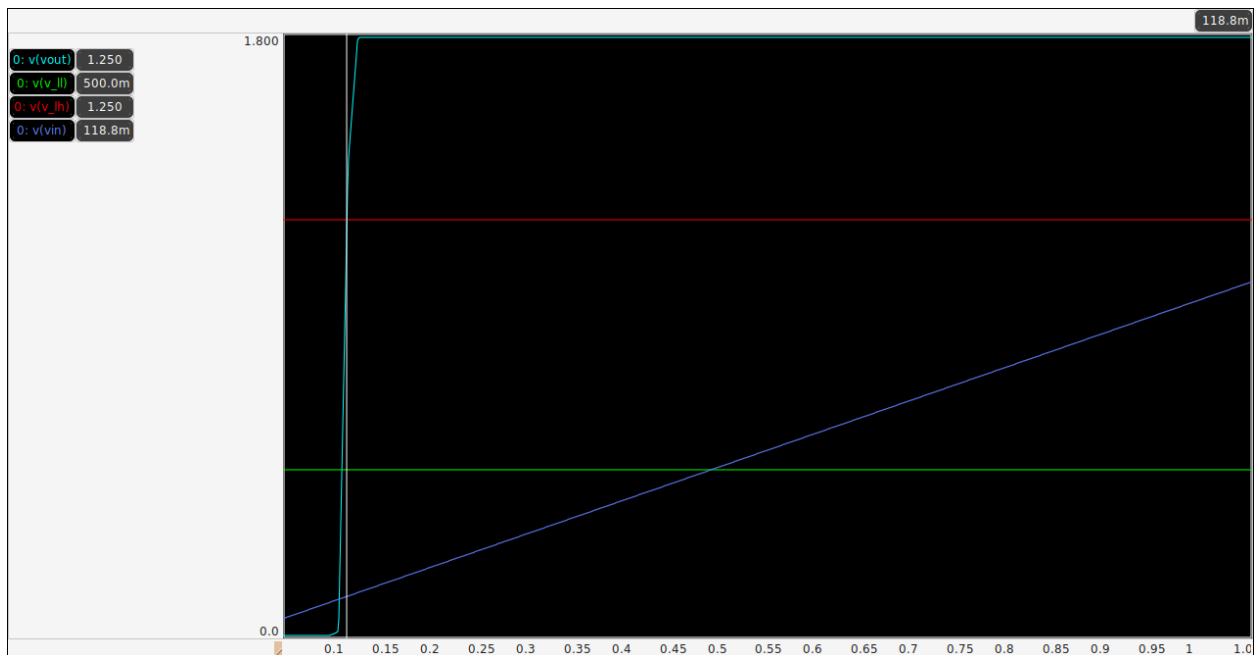


Th. 1: 0.1188

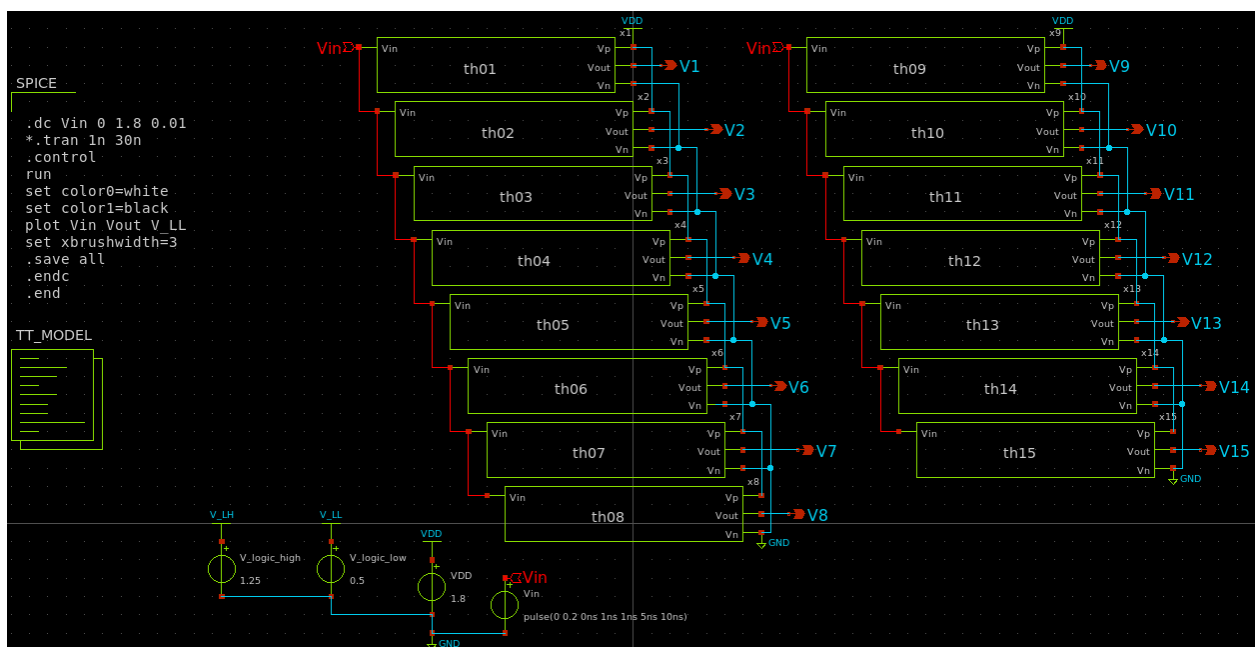


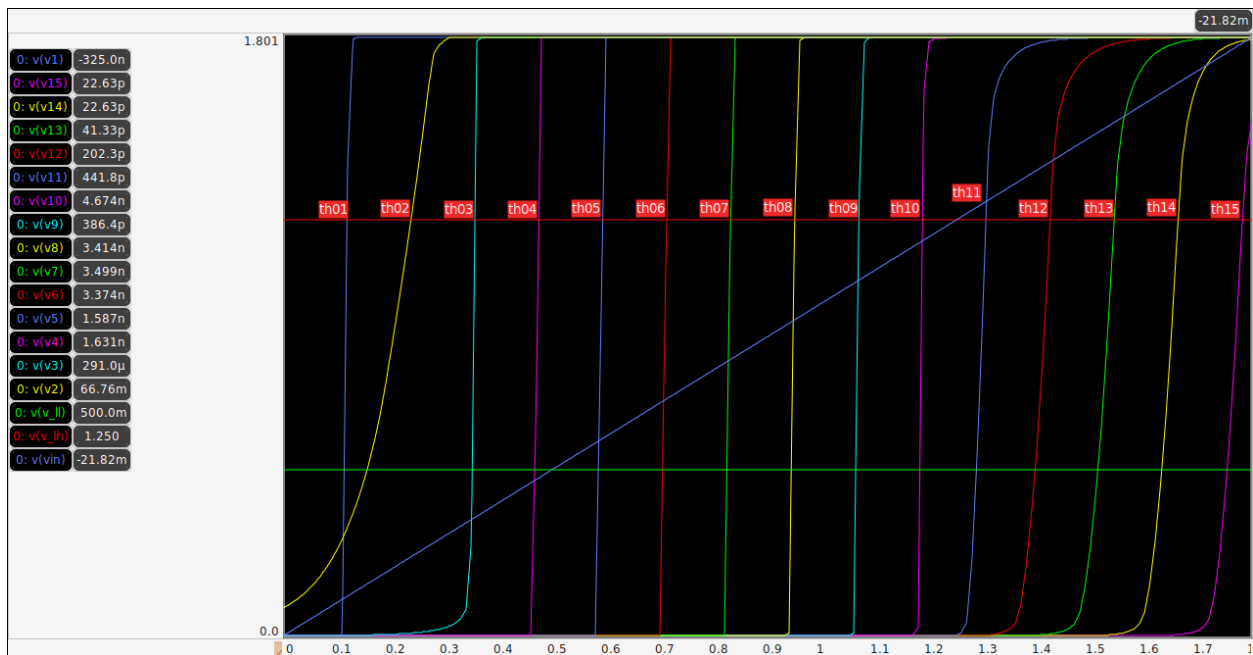
With combination of all thresholds the length and width need to be further adjusted.





Combined Analog sch.





<http://web02.gonzaga.edu/faculty/talarico/vlsi/xschemTut.html>

<https://ngspice.sourceforge.io/ngspice-control-language-tutorial.html>

Alter subckt tutorial—

<https://sourceforge.net/p/ngspice/discussion/133842/thread/5ad086c79f>

<https://sourceforge.net/p/ngspice/discussion/133842/thread/9a75acf2/#bf90/45b3/ea48>

Pyspice ngspice interpreter

<https://pyspice.fabrice-salvaire.fr/releases/v1.3/examples/ngspice-shared/ngspice-interpreter.html>

<https://github.com/ashwith/ngspicepy>

Read spice raw output data

<https://gist.github.com/snmishra/27dcc624b639c2626137/raw/742fe0dd59c7b2c41b71a1c8c1c2506d13affc53/rawread.py>

ngspice wrapper for python

<https://github.com/eps82/lyngspice/wiki>

write some verilog code to convert the thermometer code (from the inverter stages) into the binary code

```
module thermometer_to_binary (  
    input [14:0] ith,  
    output reg [3:0] binary  
);  
  
always @(*) begin  
    case (ith)  
        16'b0000000000000000: binary = 4'b0000; // ~0.0000  
        16'b0000000000000001: binary = 4'b0001; // ~0.1188  
        16'b0000000000000011: binary = 4'b0010; // ~0.2376  
        16'b0000000000000111: binary = 4'b0011; // ~0.3564  
        16'b0000000000001111: binary = 4'b0100; // ~0.4752  
        16'b0000000000011111: binary = 4'b0101; // ~0.5940  
        16'b0000000000111111: binary = 4'b0110; // ~0.7128  
        16'b0000000001111111: binary = 4'b0111; // ~0.8316  
        16'b0000000111111111: binary = 4'b1000; // ~0.9504  
        16'b0000001111111111: binary = 4'b1001; // ~1.0692  
        16'b0000011111111111: binary = 4'b1010; // ~1.1880  
        16'b0000111111111111: binary = 4'b1011; // ~1.3068  
        16'b0001111111111111: binary = 4'b1100; // ~1.4256  
        16'b0011111111111111: binary = 4'b1101; // ~1.5444  
        16'b0111111111111111: binary = 4'b1110; // ~1.6632  
        16'b1111111111111111: binary = 4'b1111; // ~1.7820  
  
        default: binary = 4'bxxxx; // Don't care  
    endcase  
end
```

endmodule

<http://www.asic-world.com/verilog/veritut.html>

<https://nandland.com/introduction-to-verilog-for-beginners-with-code-examples/>

https://www.youtube.com/playlist?list=PLfGJELQIDBN0VsXQ68_FEYyqcym8CTDN

<https://github.com/Swagatika-Meher/msvsd2bitcomp>

ALIGN Tool-

https://www.youtube.com/playlist?list=PLvXKBNlvcSm30Y0zu1765oG_x-ECU8tVG

<https://github.com/syedimaduddin/msvsd4bituc/tree/main/Week-1>

<https://github.com/ALIGN-analoglayout/ALIGN-public>

<https://learning.edx.org/course/course-v1:HarveyMuddX+ENGR85A+3T2021/home>

<https://www.vlsiuniverse.com/digital-thermometer-code-in-verilog-vhdl-flash-adc-binary-encoder/>

<https://www.classcentral.com/subject/vlsi?free=true>

<https://www.eng.biu.ac.il/temanad/digital-vlsi-design/>

https://www.youtube.com/playlist?list=PLZU5hLL_713x0_AV_rVbay0pWmED7992G

https://youtu.be/BlqLk23hE90?list=PLZU5hLL_713x0_AV_rVbay0pWmED7992G

get familiar with the yosis syntesis tool to convert the verilog code into a CMOS circuit

<https://www.mehmetburakaykenar.com/synthesis-n-bit-counter-with-open-source-yosys-synthesizer/294/>

https://github.com/spdy1895/RTL_synthesis_using_sky130

<https://github.com/lmellal/RTL-workshop-Sky130-PDK>

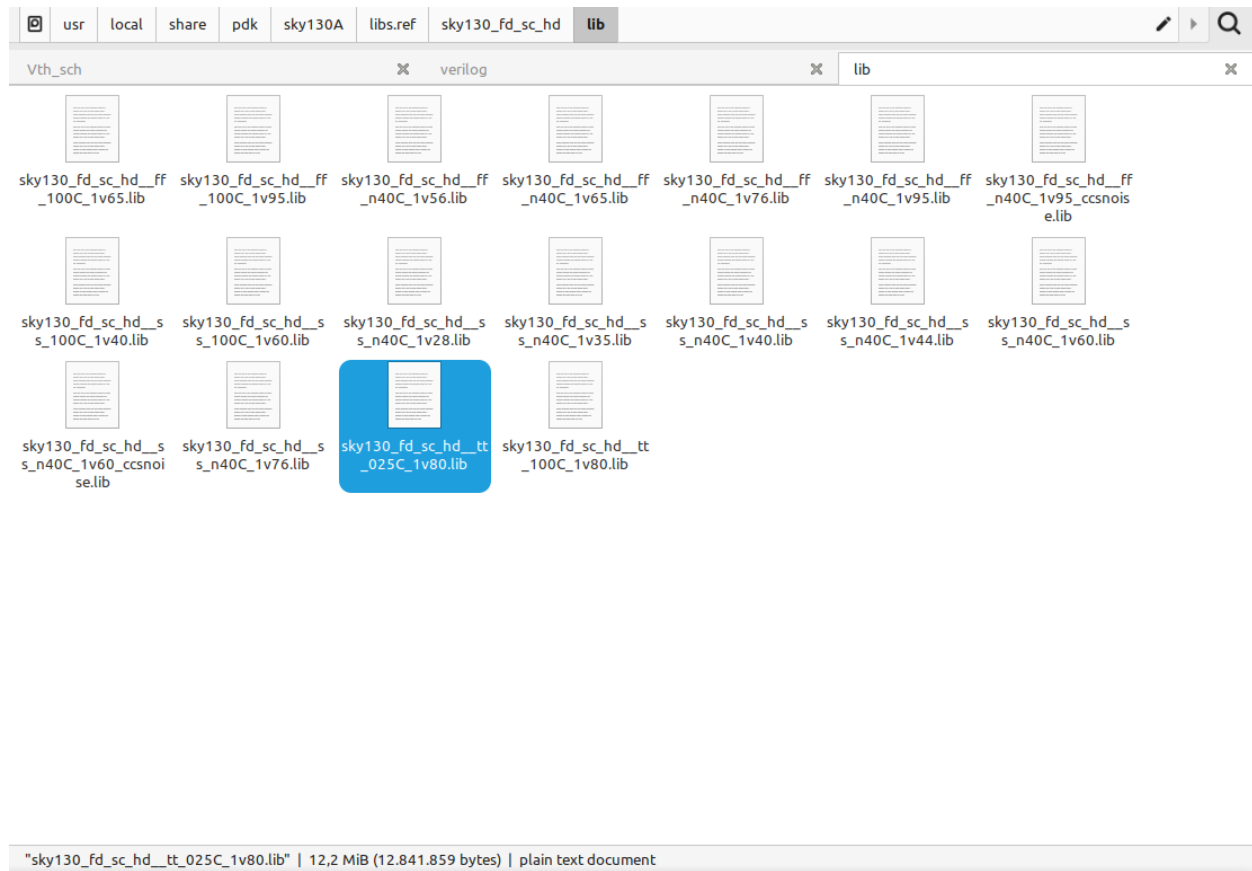
After install YoSys (see the bash file in the repository), run the following commands—

```
>> yosys          // open yosys console in terminal
```

```
>> read_liberty -lib
```

```
/usr/local/share/pdk/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd__tt_025C_1v80.lib
```

```
// reading the suitable liberty file (see the screenshot below)
```



```
>> read_verilog therm_code.v // read the verilog file
```

```
Terminal - exotic@exotic-virtual-machine: ~/Desktop/ASIC_ADC/verilog
File Edit View Terminal Tabs Help

exotic@exotic-virtual-machine: ~/Desktop/yosys x exotic@exotic-virtual-machine: ~/Desktop/ASIC_ADC/... x

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

-----/

Yosys 0.35+36 (git sha1 c95298225, gcc 11.4.0-1ubuntu1~22.04 -fPIC -Os)

yosys> read verilog therm_code.v
1. Executing Verilog-2005 frontend: therm_code.v
Parsing Verilog input from `therm_code.v' to AST representation.
Generating RTLIL representation for module `thermometer_to_binary'.
Successfully finished Verilog frontend.

yosys> 
```

>> **synth -top thermometer_to_binary** // synthesize the module

>> **abc -liberty**

/usr/local/share/pdk/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd__tt_025C_1v80.lib

>> **flatten** // // to invoke flat synthesis after netlist generation

// netlist generation

>> **show** // view graphical

>> **write_verilog therm.v** // write the generated netlist to a new module/verilog file for verification

Or

>> **write_verilog -noattr therm.v** // // to write verilog netlist without attributes(clean)

Optimize:

>> **flatten**

>> **opt_clean -purge** // running optimization

>> **show**

>> **write_spice therm.spice**

>> **show -prefix therm -format svg**//save diagram in .svg

Or

>> **show -prefix therm -format dot**//save diagram in .dot

Or

>> **show -enum -prefix therm -format dot** //save diagram in .dot with gate numbers

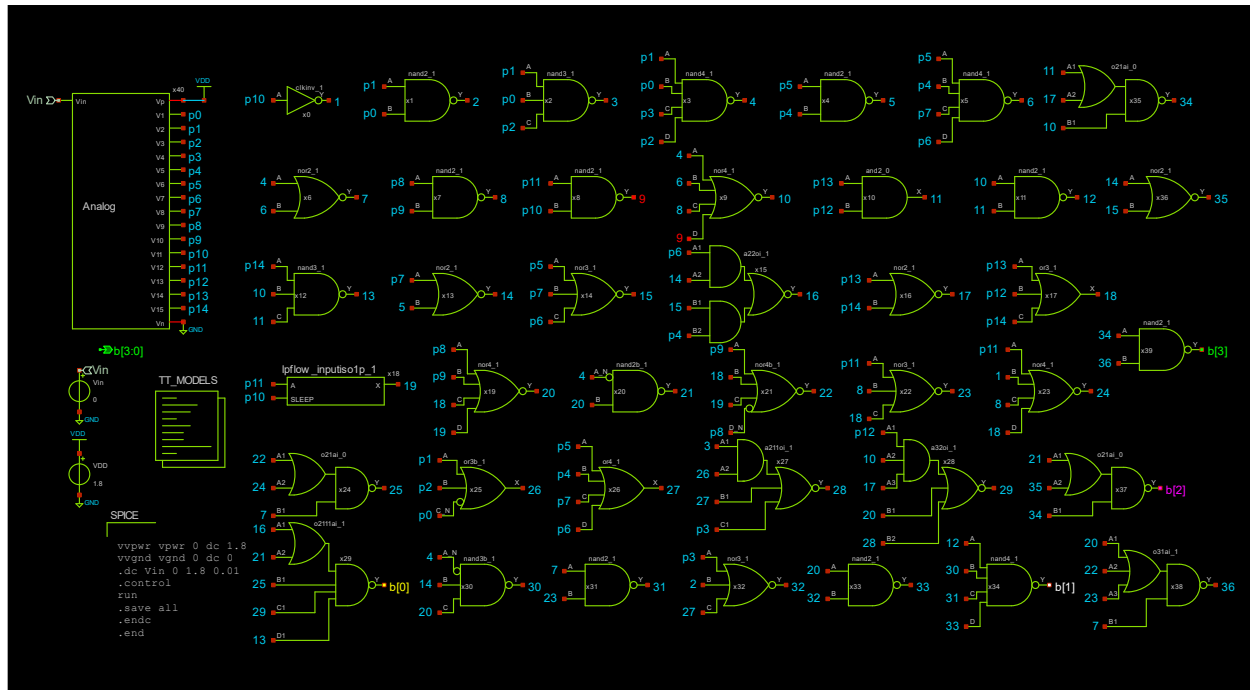
Digital flow with qflow:

http://opencircuitdesign.com/qflow/tutorial_nogui.html

Layout editor

<https://www.youtube.com/watch?v=K-6GMwmc4>

<https://www.layouteditor.org/layout/basics/installation>



Even though Yosys is a useful tool is useful for synthesis, still it is not enough for layout digital logic. It can generate the Verilog netlist and schematic netlist, but to implement the above logic gates either it can be done manually with Magic VLSI (which would be time-consuming, stressful, and prone to error), or using some automation tools flow (ALIGN, qflow, or OpenLane). ALIGN is suitable for analog design, and qflow and OpenLane have already Yosys integrated. For this project I preferred OpenLane. Using qflow or OpenLane the above diagram is not necessary, this steps are already taken care of in the processes. However, the above schematic is still beneficial to compare the pre-layout and post-layout results.

OpenLane Procedures:

Install OpenLane- <https://github.com/The-OpenROAD-Project/OpenLane>

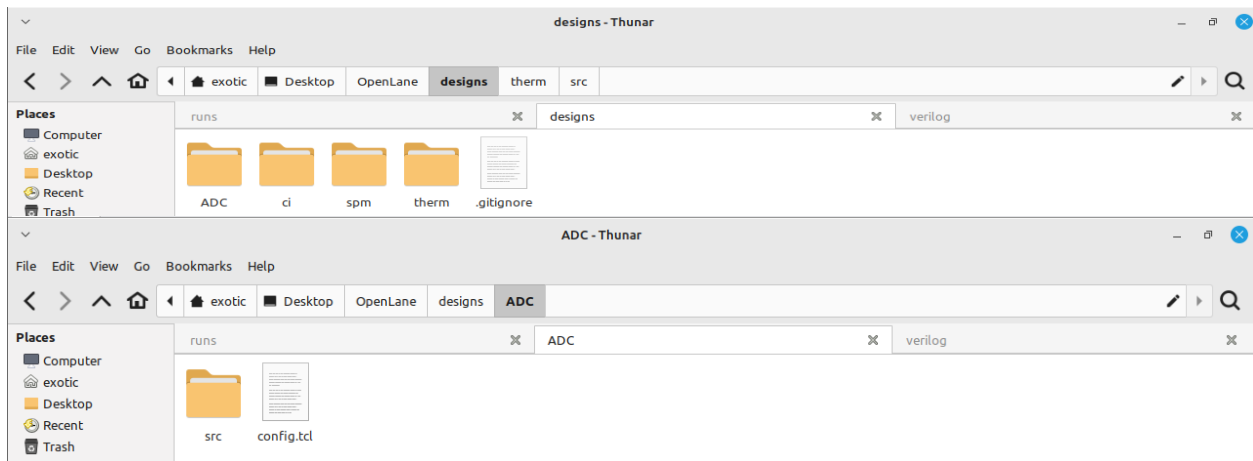
Add a new design in the designs folder:

```
./flow.tcl -design ADC -init_design_config -add_to_designs
```

By default it will create config.json file

If config.ct1 is preferred the following command is to be entered:

```
./flow.tcl -design ADC -init_design_config -add_to_designs -config_file config.tcl
```



Now the config.tcl has to be configured accordingly (resolving the errors) and the verilog file need to be copied to src folder, the verilog file name should be same as the design name, otherwise specified in the command.

At first let's try run the processes with default config:

```
./flow.tcl -design ADC
```

Here I have used the verilog code without any clock and I have errors.

```
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[INFO]: Running linter (Verilator) (log: designs/ADC/runs/RUN_2023.12.26_11.29.39/logs/synthesis/linter.log)...
[ERROR]: 2 errors found by linter
[ERROR]: Step 0 (verilator lint check) failed with error:
-code 1 -level 0 -errorcode NONE -errorinfo {
  while executing
  "throw error"
    (procedure "run_verilator" line 86)
    invoked from within
    "run_verilator"
      (procedure "run_verilator_step" line 3)
      invoked from within
      "run_verilator_step" -errorline 1
```

Lets have look at the log file named linter.log

```
therm.v x  spm.v x  pin_order.cfg x  config.tcl x  linter.log x
%Error: Specified --top-module 'ADC' was not found in design.
%Error: Exiting due to 1 error(s)
```

As the design/file name is ADC, but the module name in the code is different, it threwed the errors. I have changed the module name to ADC, and lets run it again.

```
[ERROR PDN-0175] Pitch 5.9890 is too small for, must be atleast 6.6000
Error: pdn_cfg.tcl, 92 PDN-0175
child process exited abnormally
```

Now there is Pitch error. Let's edit config.tcl file, adding the following lines:

```
set ::env(FP_PDN_AUTO_ADJUST) 0
set ::env(FP_PDN_VPITCH) 7.0
set ::env(FP_PDN_VOFFSET) 7.0
```

after adding above lines the error is gone but new error appeared.

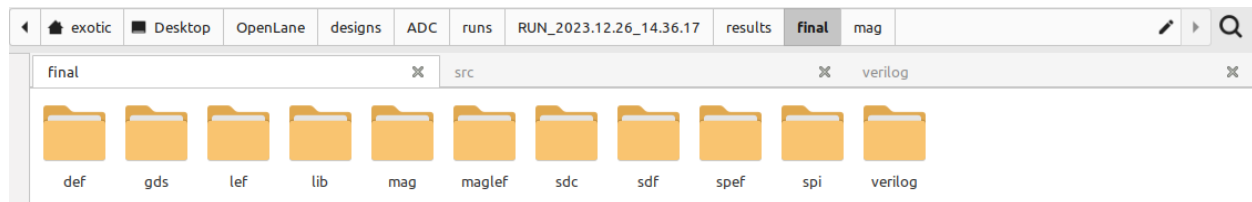
```
[ERROR]: during executing openroad script /openlane/scripts/openroad/gpl.tcl
[ERROR]: Log: designs/ADC/runs/RUN_2023.12.26_14.25.51/logs/placement/6-global_skip_io.log
[ERROR]: Last 10 lines:
[INFO GPL-0017] NonPlaceInstsArea: 66313600
[INFO GPL-0018] PlaceInstsArea: 294032000
[INFO GPL-0019] Util(%): 64.74
[INFO GPL-0020] StdInstsArea: 294032000
[INFO GPL-0021] MacroInstsArea: 0
[ERROR GPL-0302] Use a higher -density or re-floorplan with a larger core area.
Given target density: 0.60
Suggested target density: 0.65
Error: gpl.tcl, 75 GPL-0302
child process exited abnormally

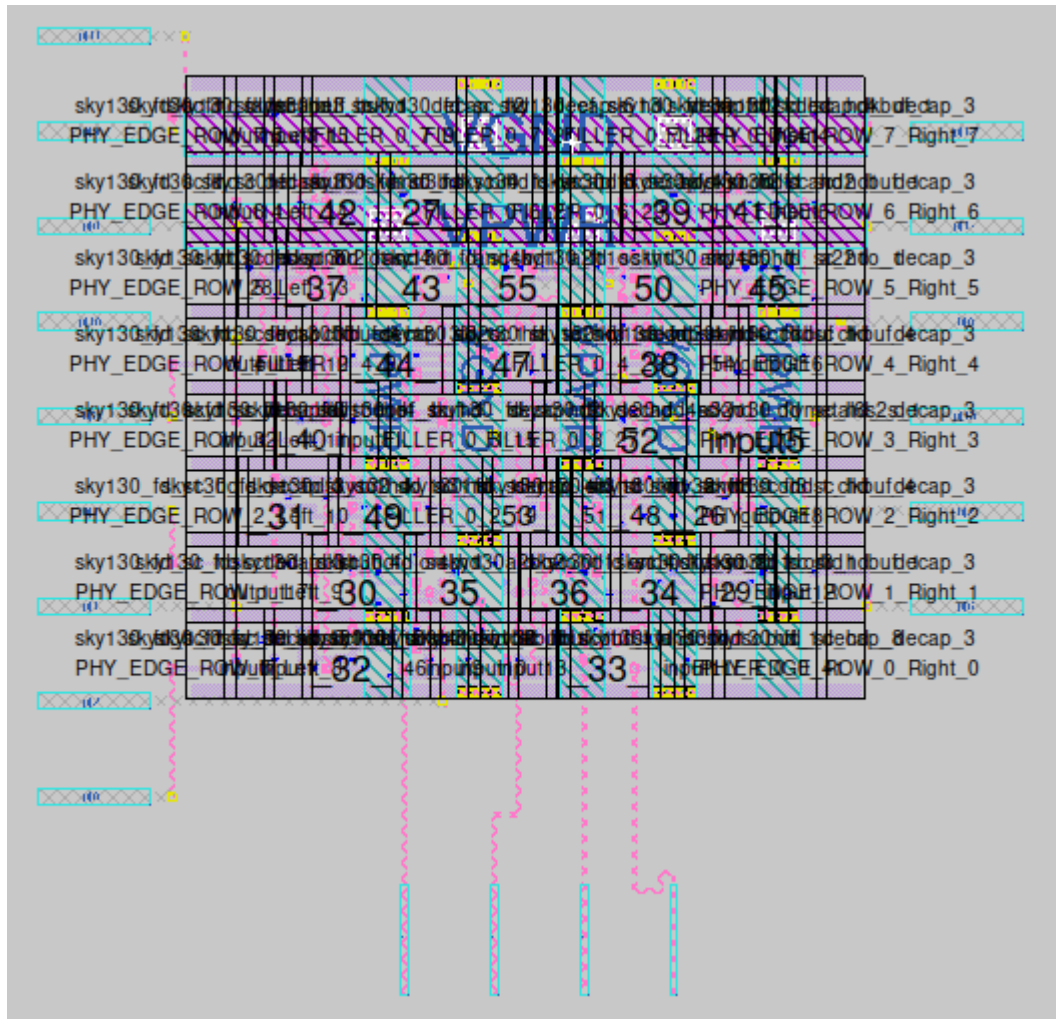
[ERROR]: Creating issue reproducible...
```

Adding another line to fix that error:

```
set ::env(PL_TARGET_DENSITY) "0.65"
```

now the layout has generated in different formats.





It is the digital part of the layout. For the analog part I preferred ALIGN.

Install ALIGN VLSI:

<https://github.com/ALIGN-analoglayout/ALIGN-pu>
<https://github.com/syedimaduddin/msvsd4bituc/tree/main/Week-2blic>

- simulate the complete (ultra small) ADC

Prelayout simulation:

To run these standard cells the "TT_MODELS" content should be:

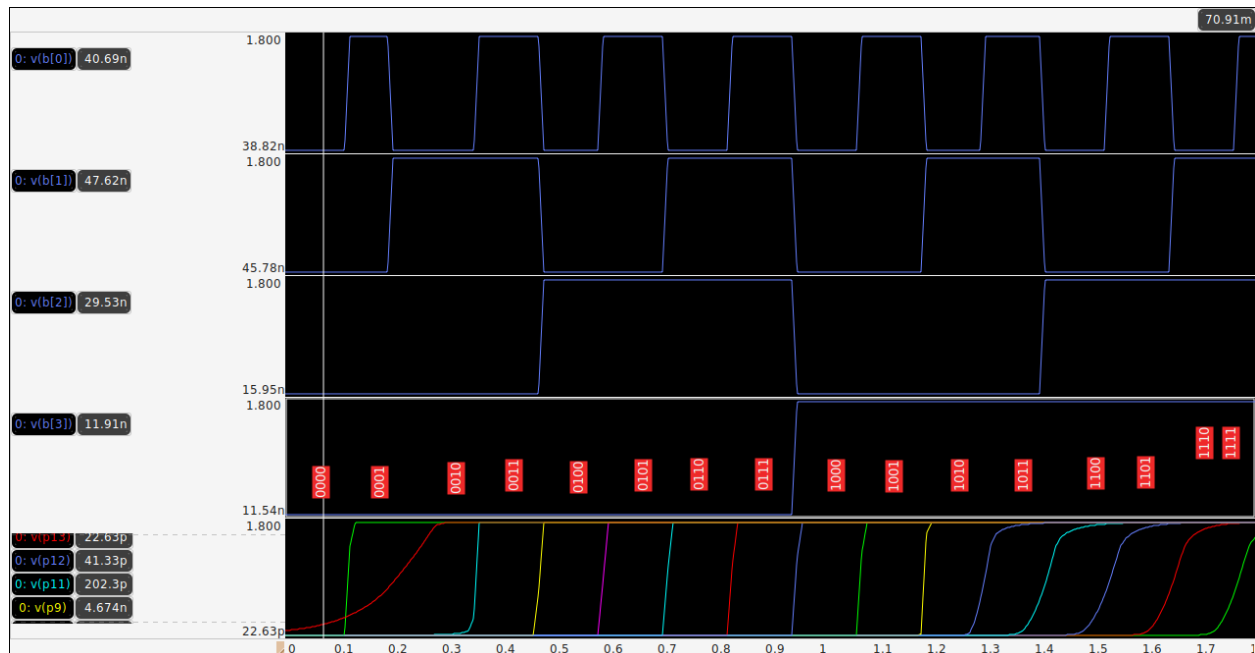
'''

name=TT_MODELS
only_toplevel=true
format="tcllevel(@value)"

```

value=".lib $::SKYWATER_MODELS/sky130.lib.spice tt
.include $::SKYWATER_STDCELLS/sky130_fd_sc_hd.spice"
Spice_ignore=false
'''

```



References:

<https://github.com/bluecmd/learn-sky130/blob/main/schematic/xschem/getting-started.md>

OpenLane:

https://openlane.readthedocs.io/en/latest/usage/exploration_script.html

Magic VLSI command/long commands

check channel before or after routing

:channel

Clear the channel/any feedback view

:feedback clear

Open netlist window

:specialopen netlist

Save netlist (save and autowrite)

```

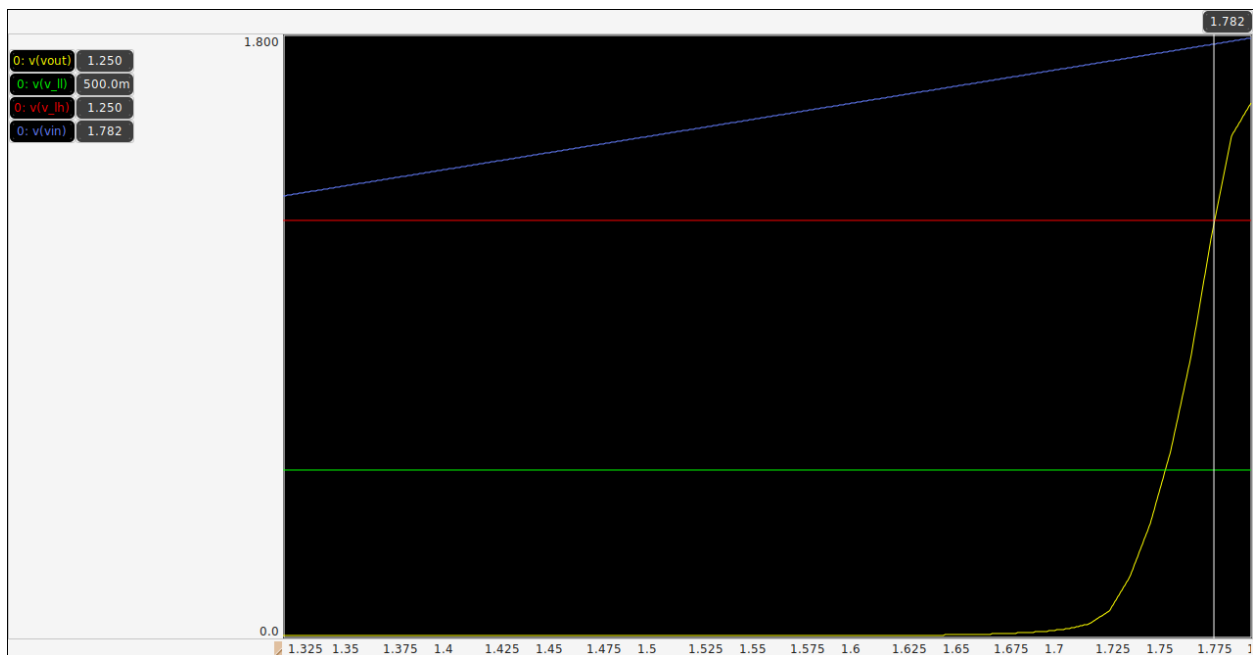
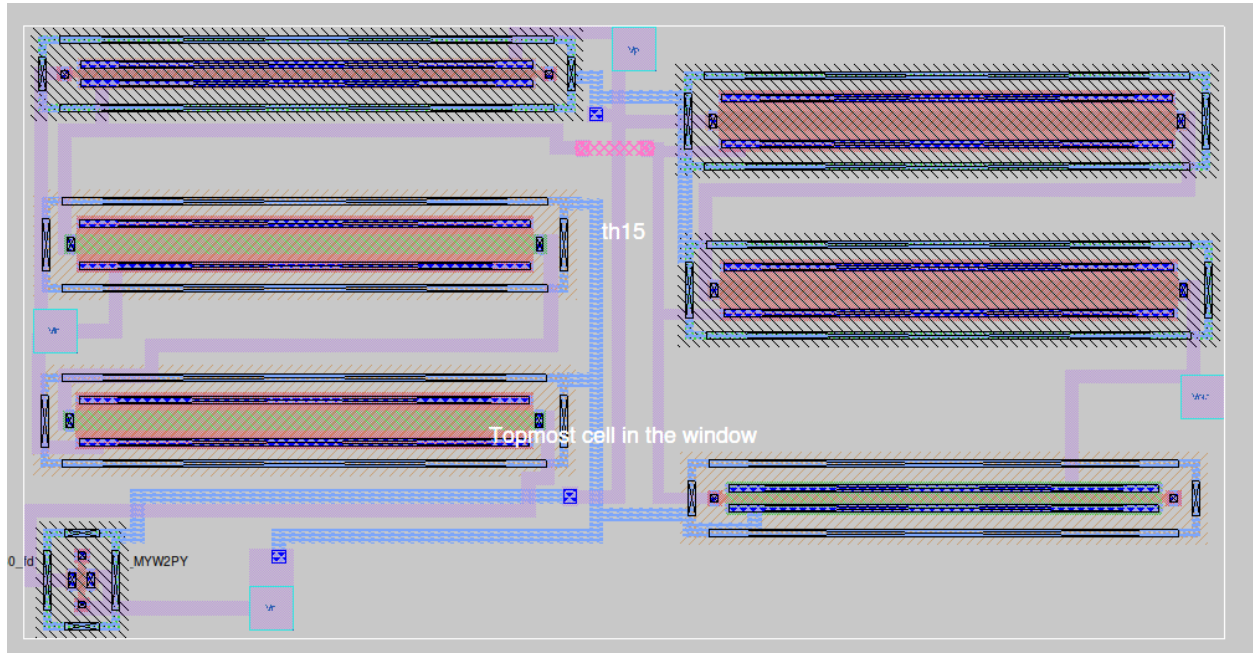
extract do local
extract all
ext2spice lvs
ext2spice cthresh 0 rthresh 0

```

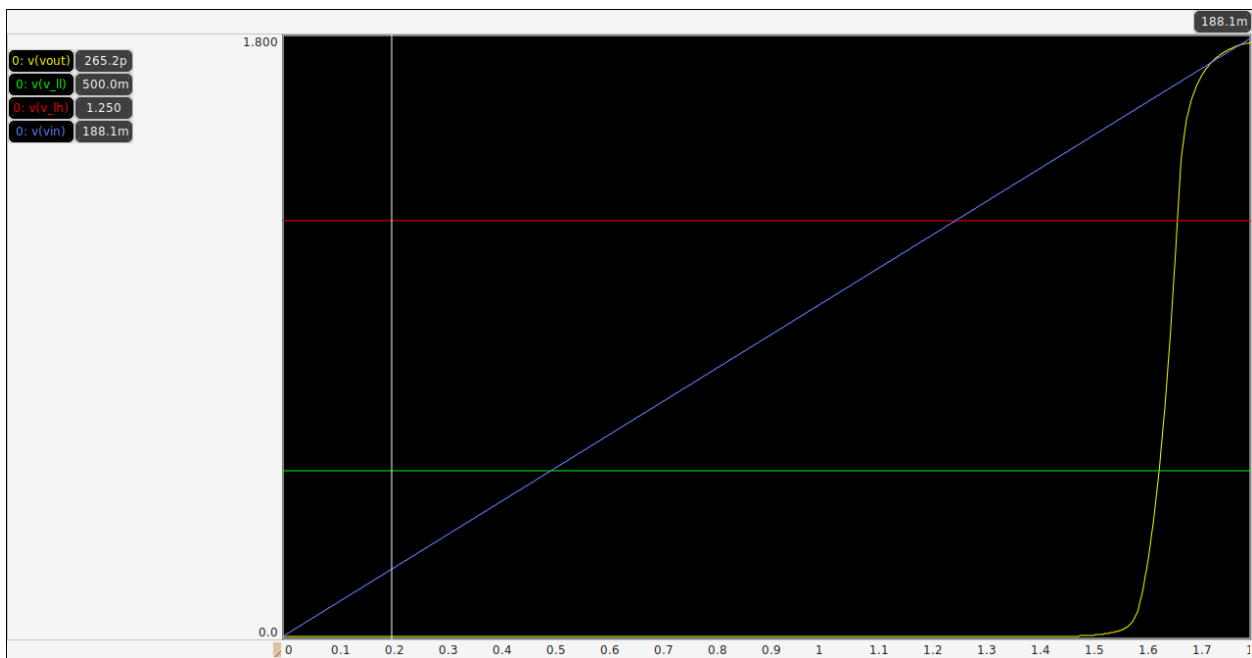
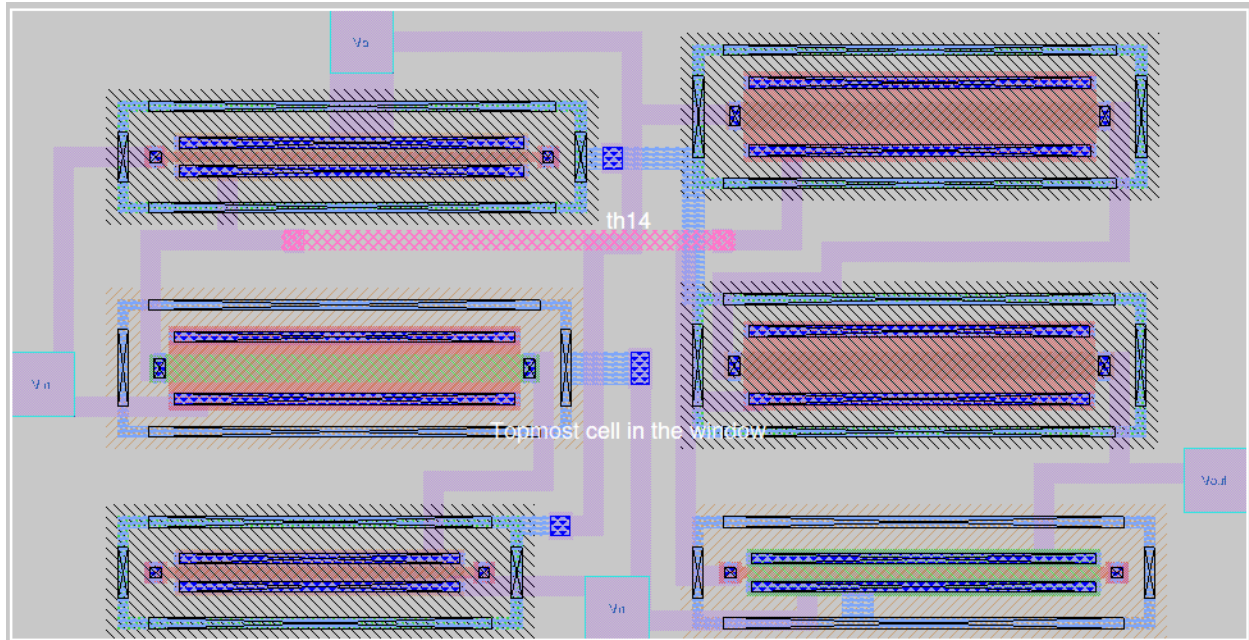
ext2spice

Analog Design:

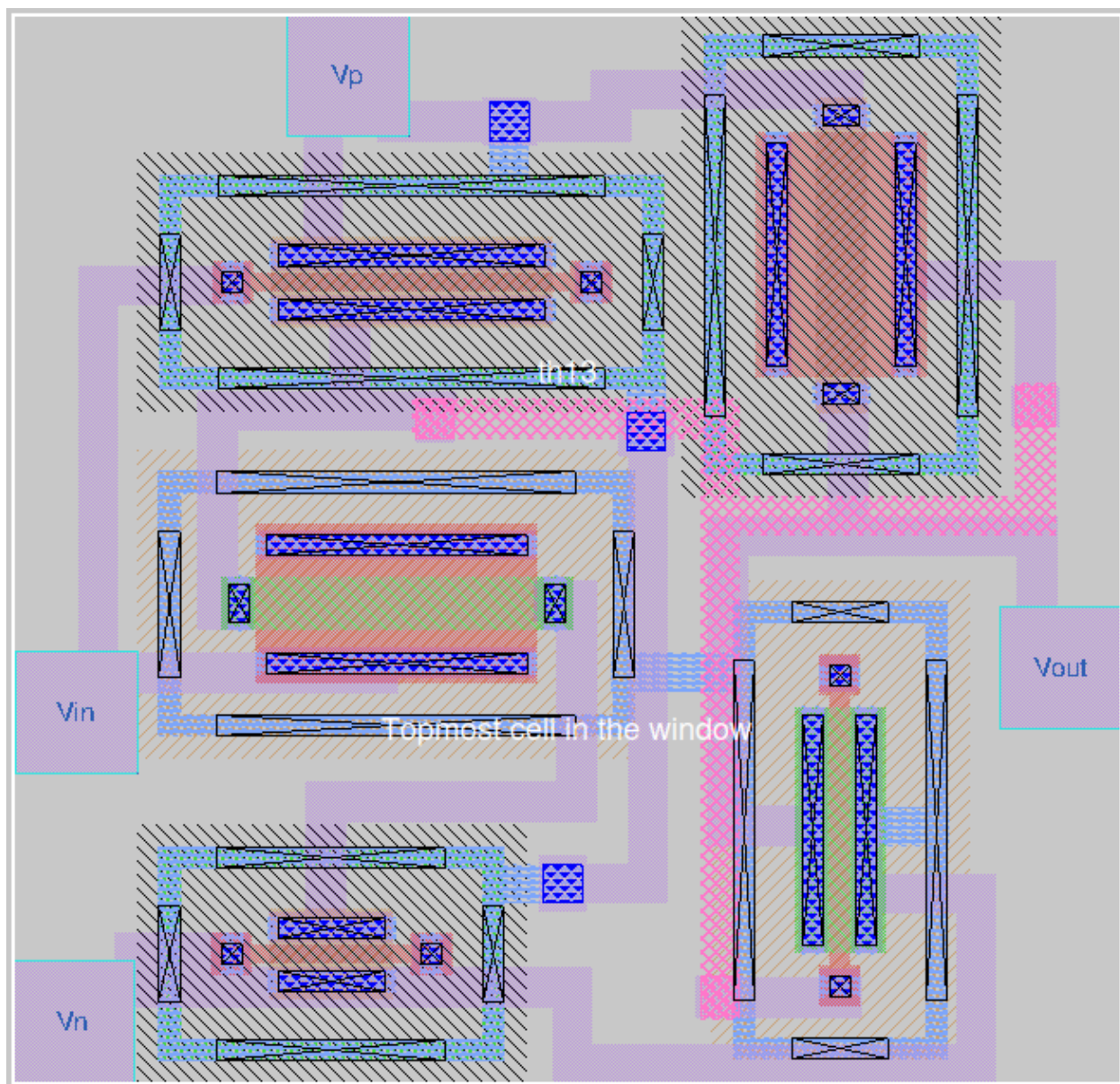
Th. 15 ([1.782](#))

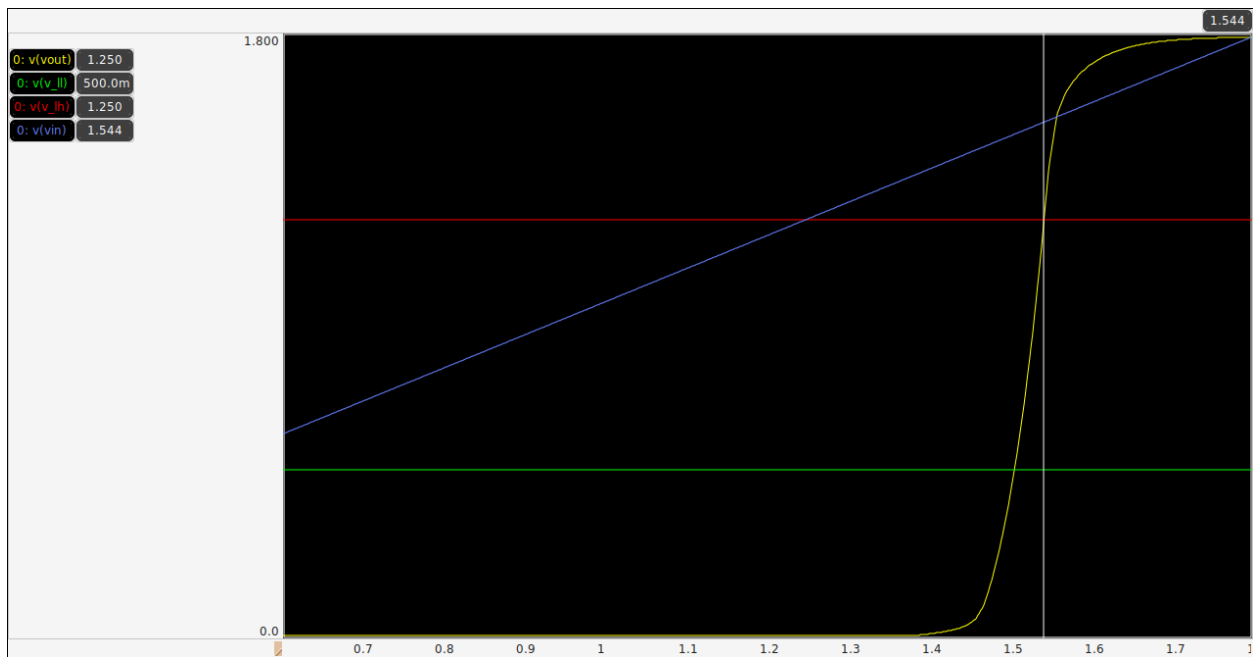


Th. 14 ([1.663](#))

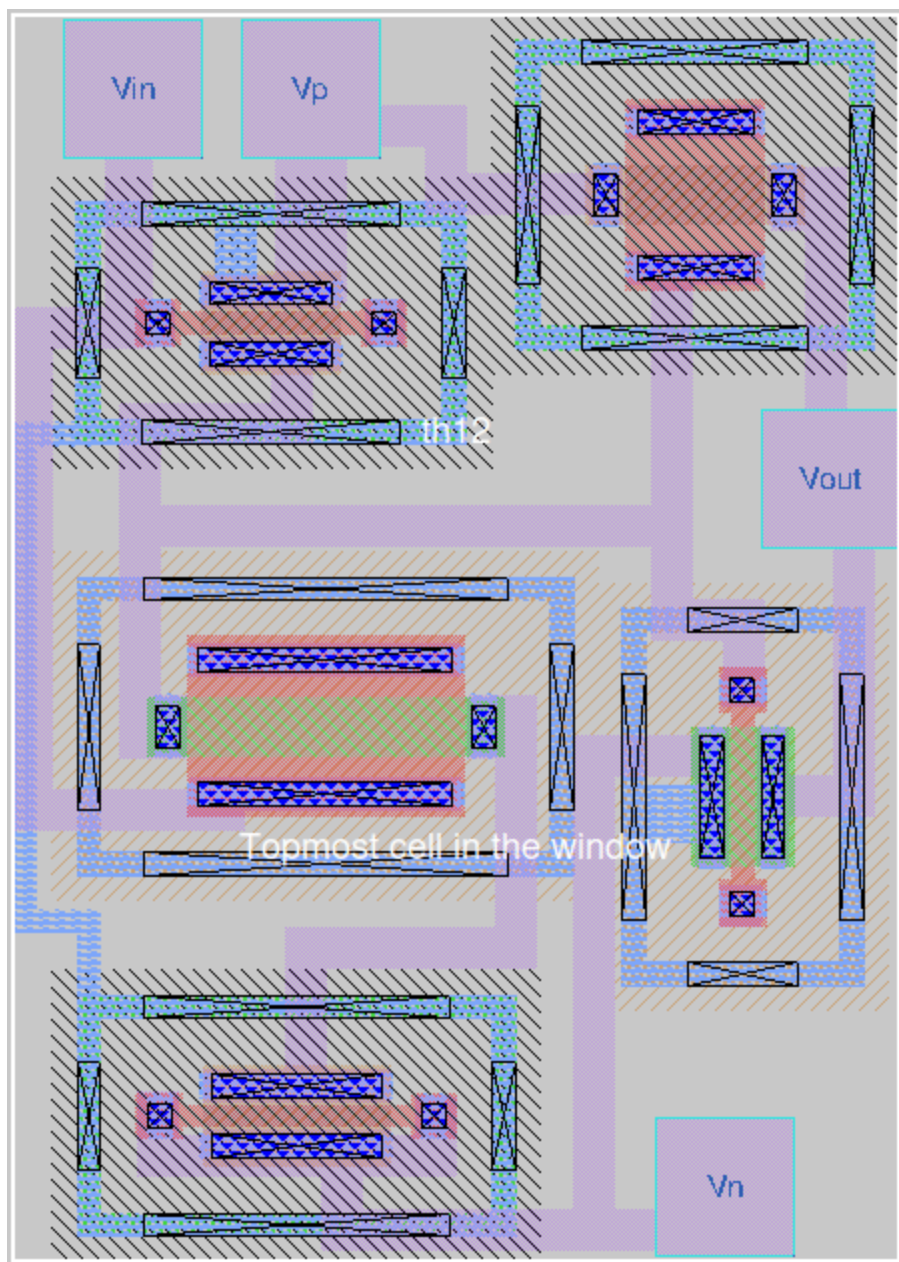


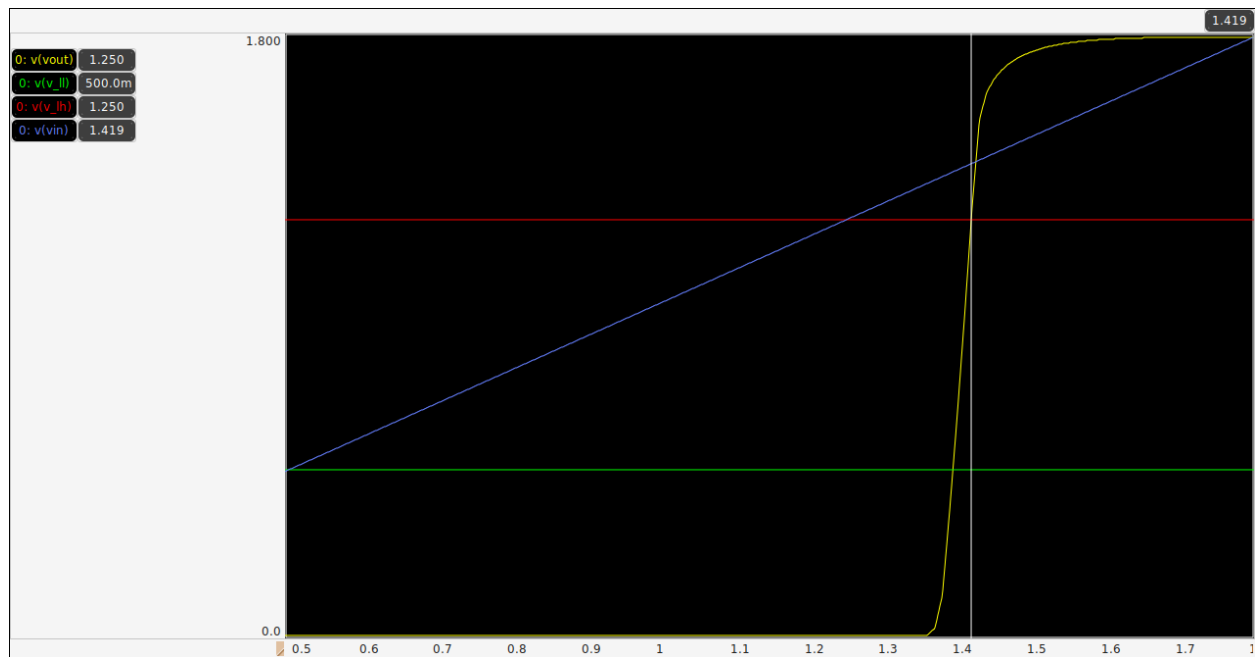
Th. 13 (1.544)



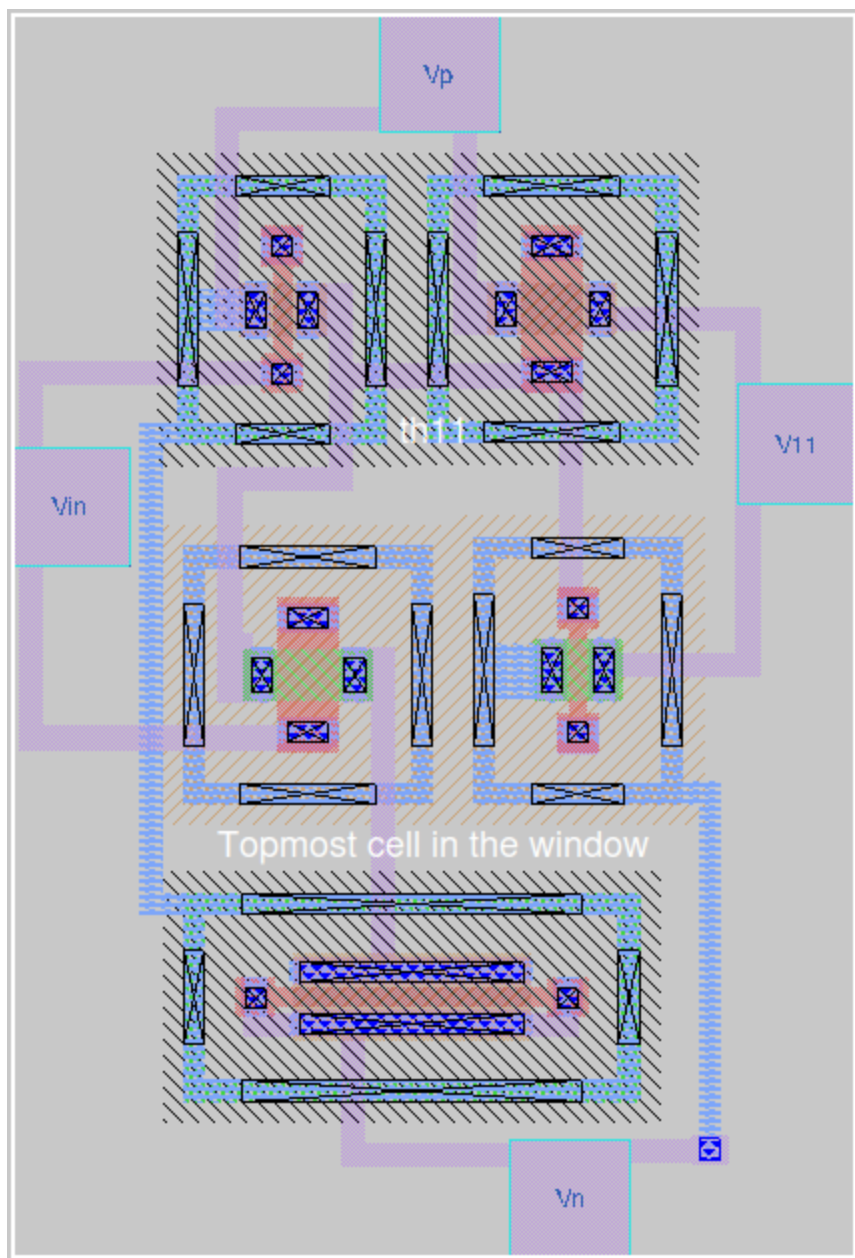


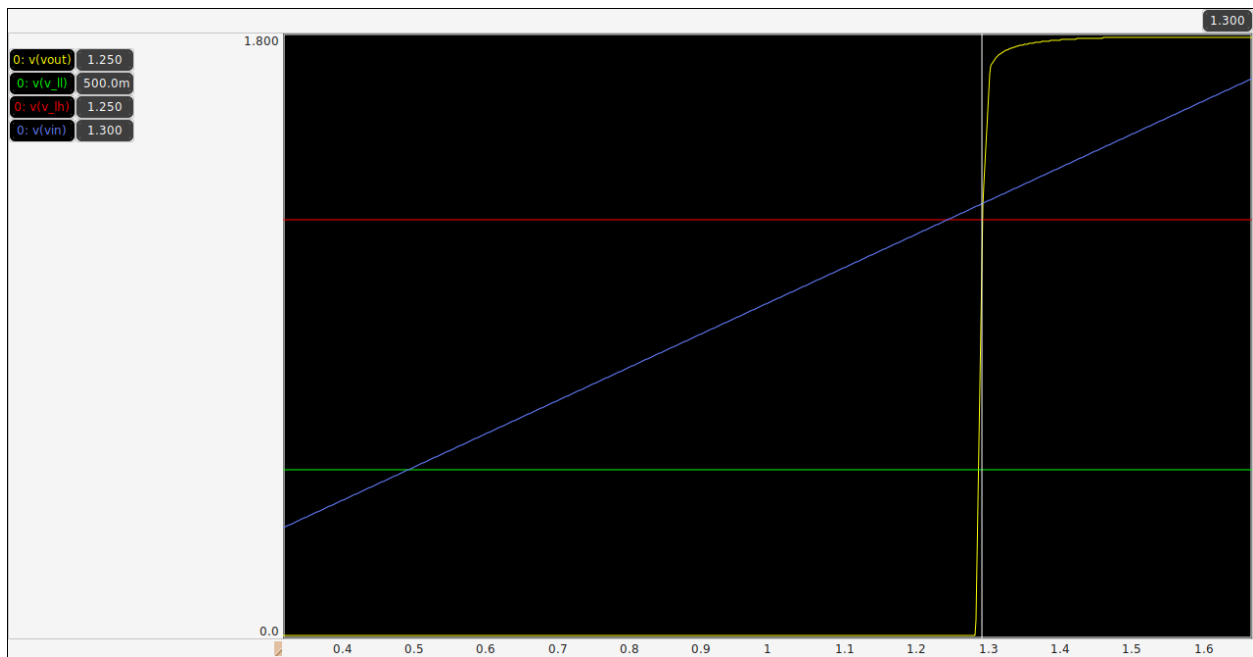
Th. 12 (1.419)



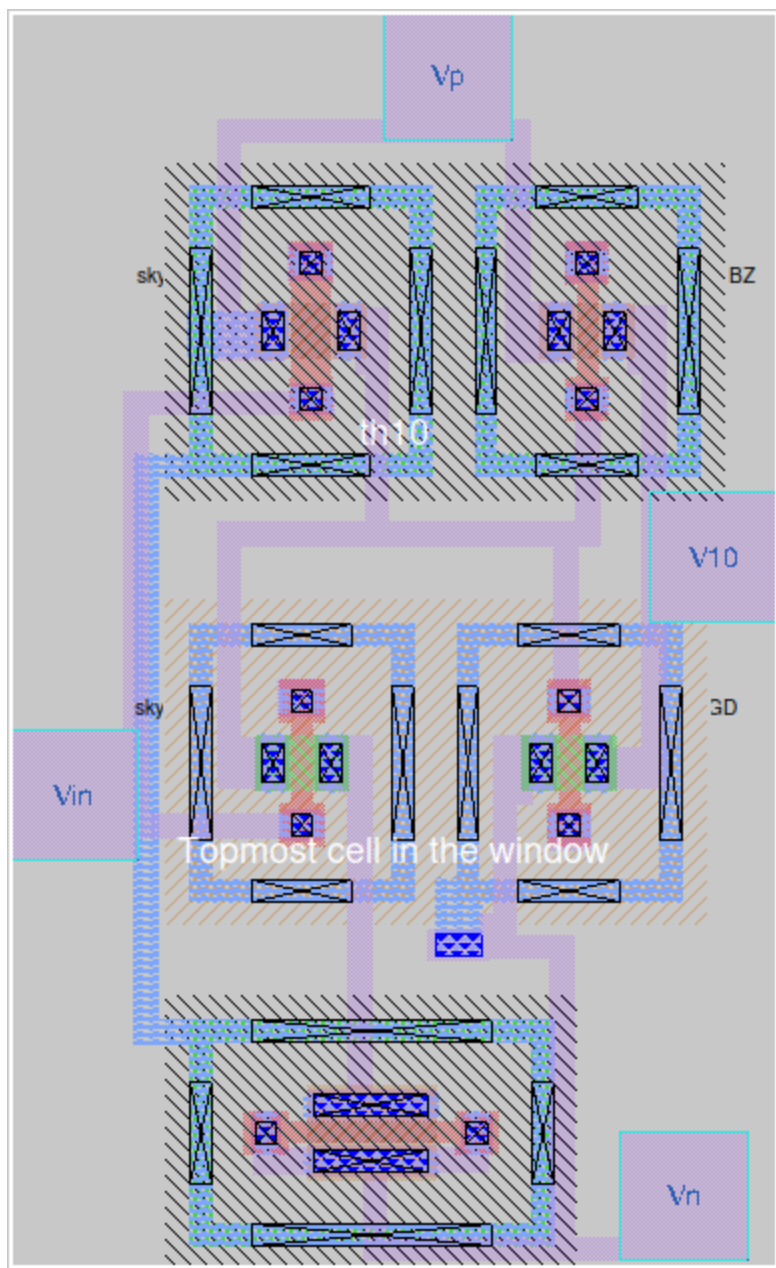


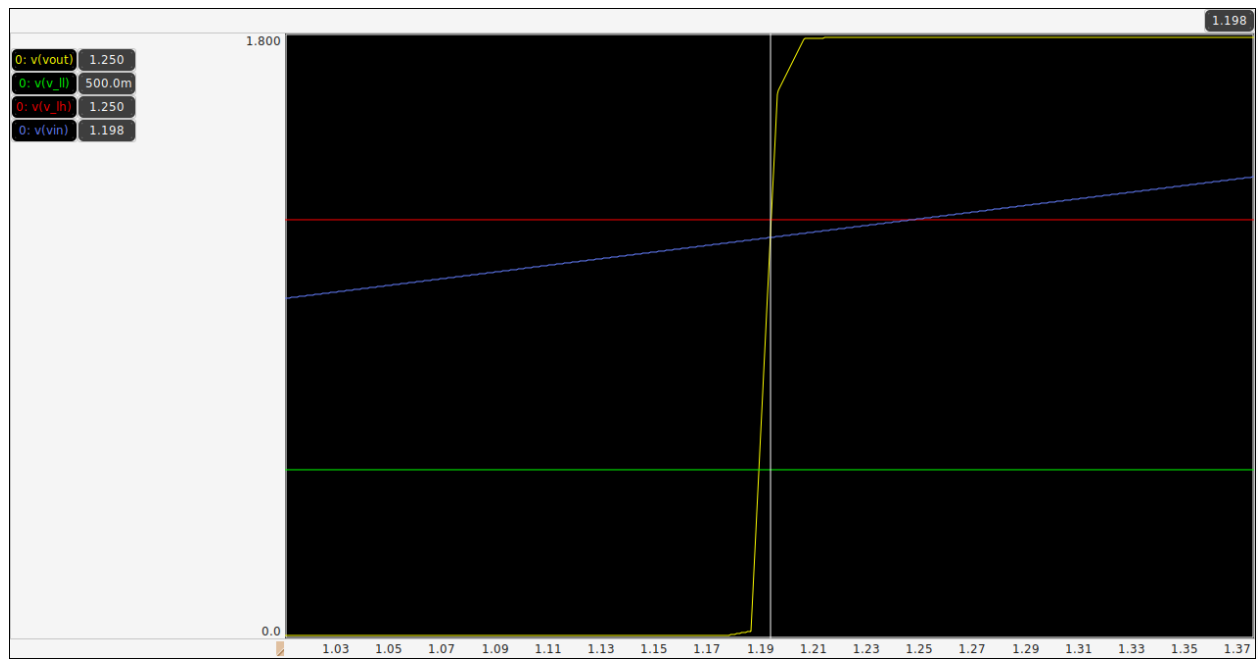
Th. 11 (1.30)



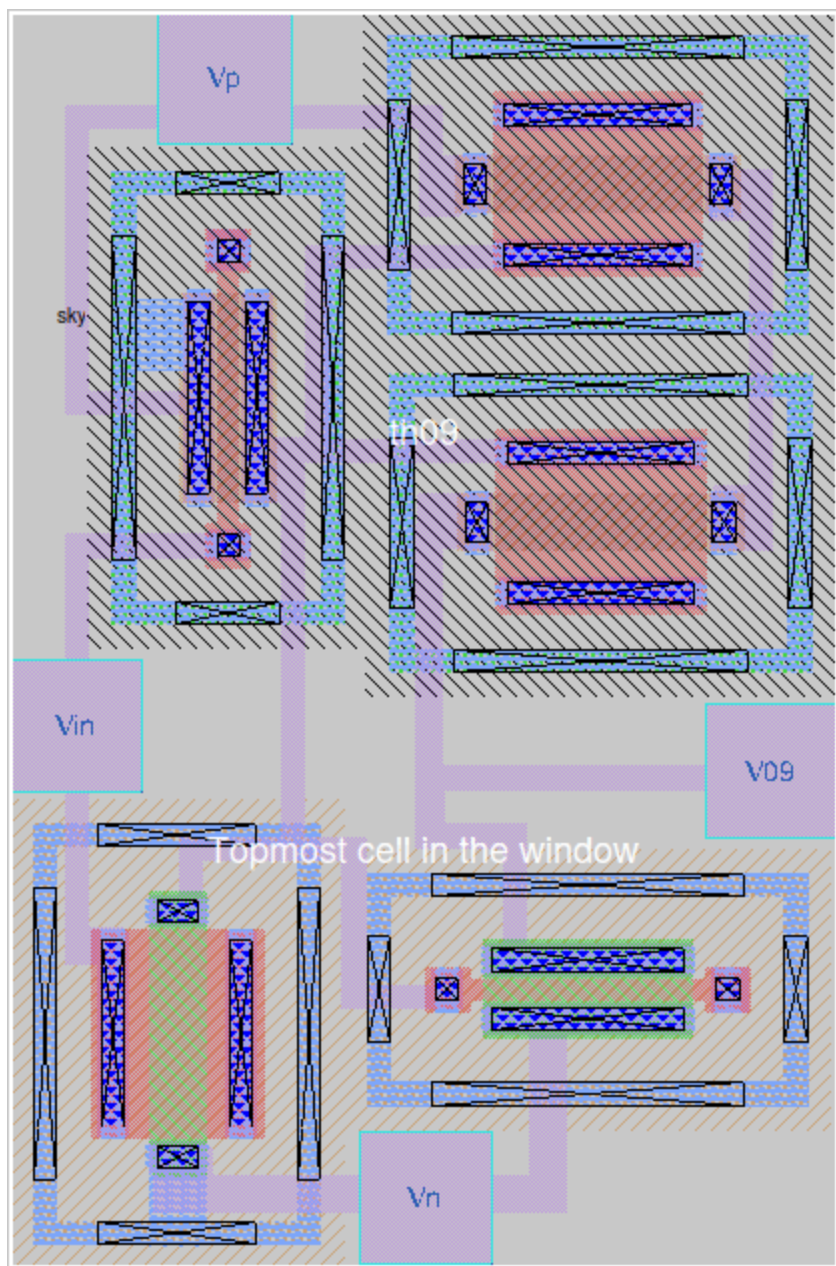


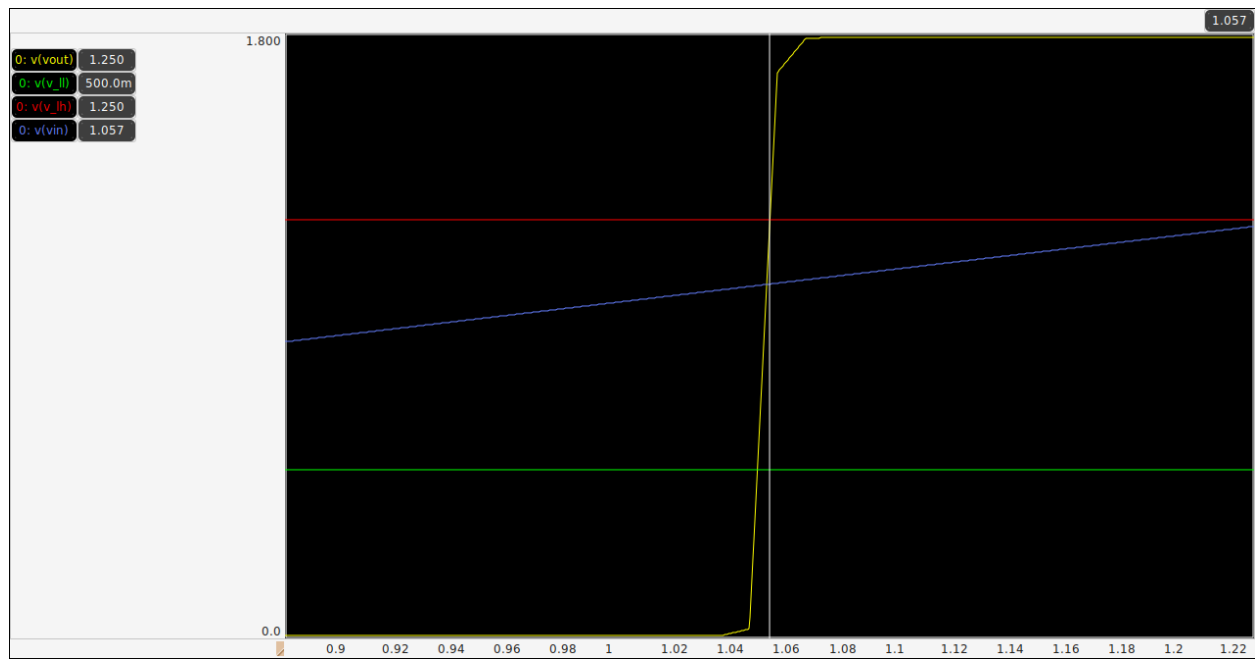
Th. 10 (1.198)



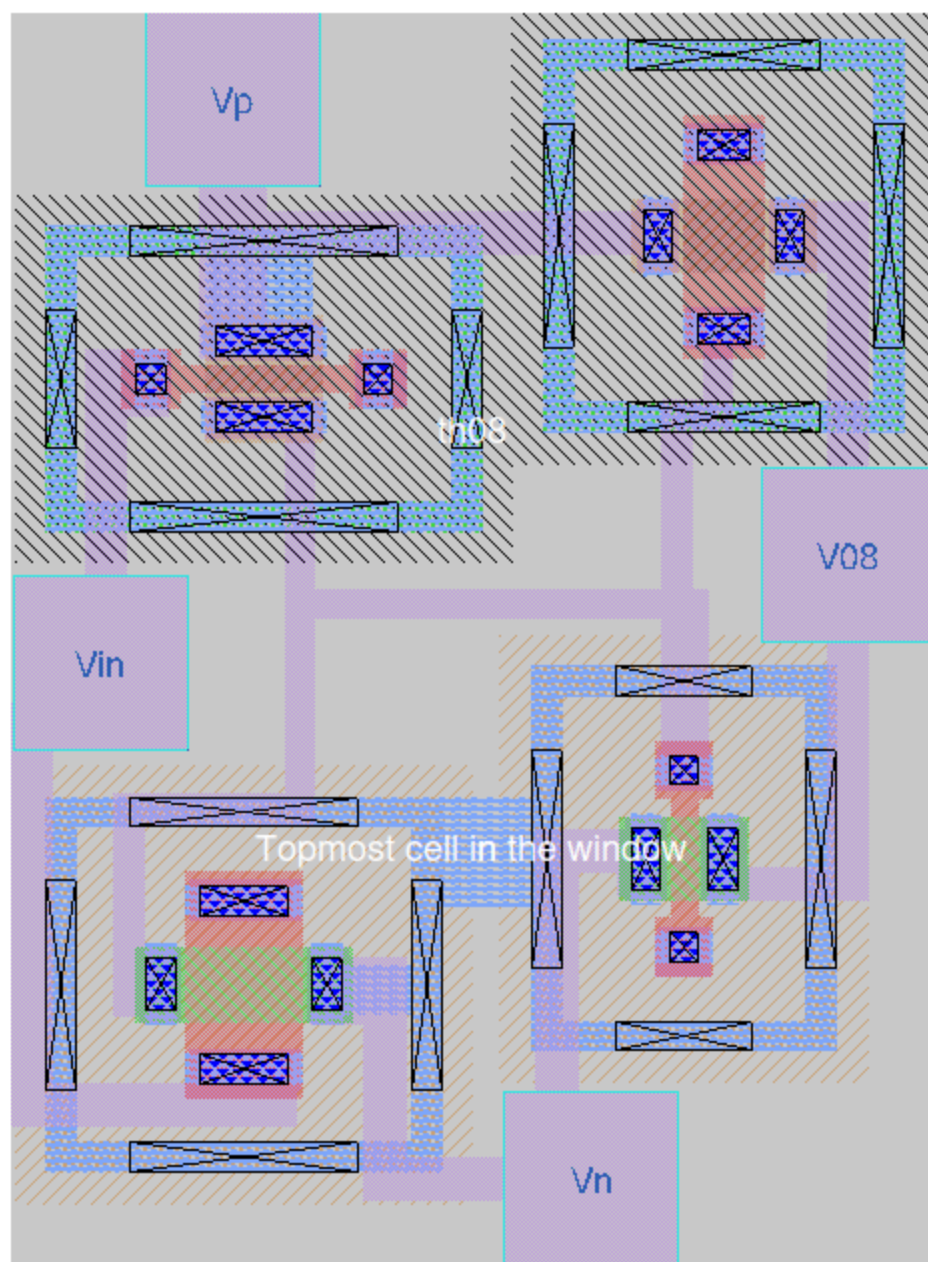


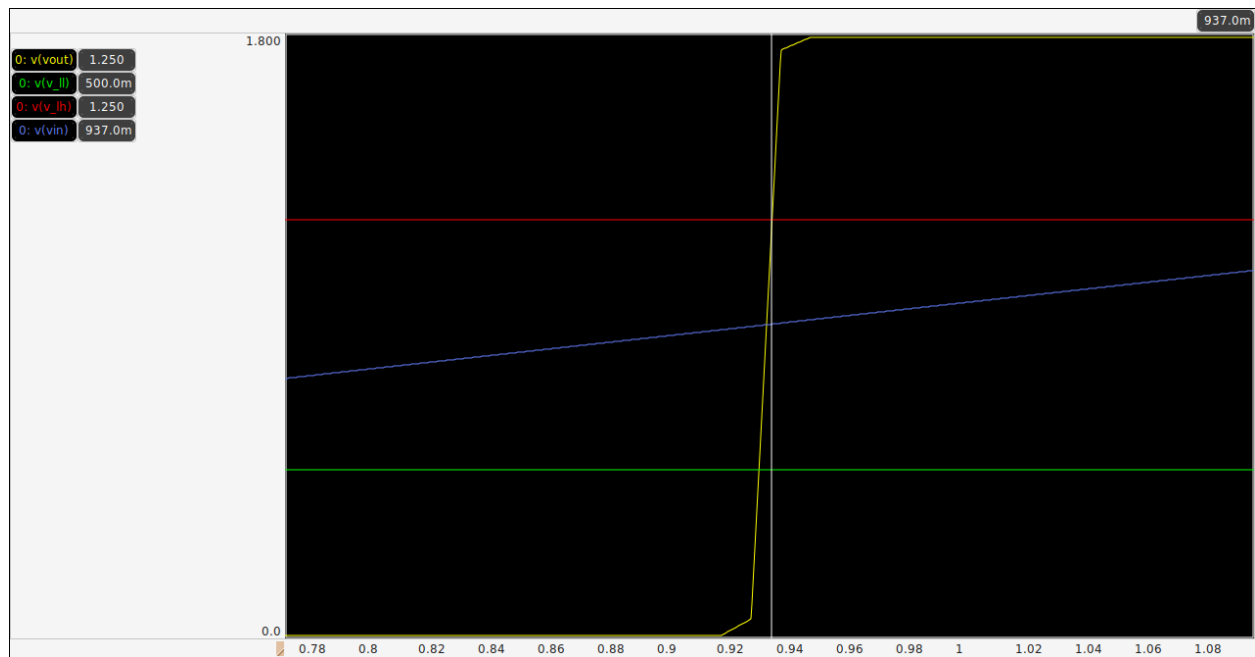
Th. 09 (1.057)



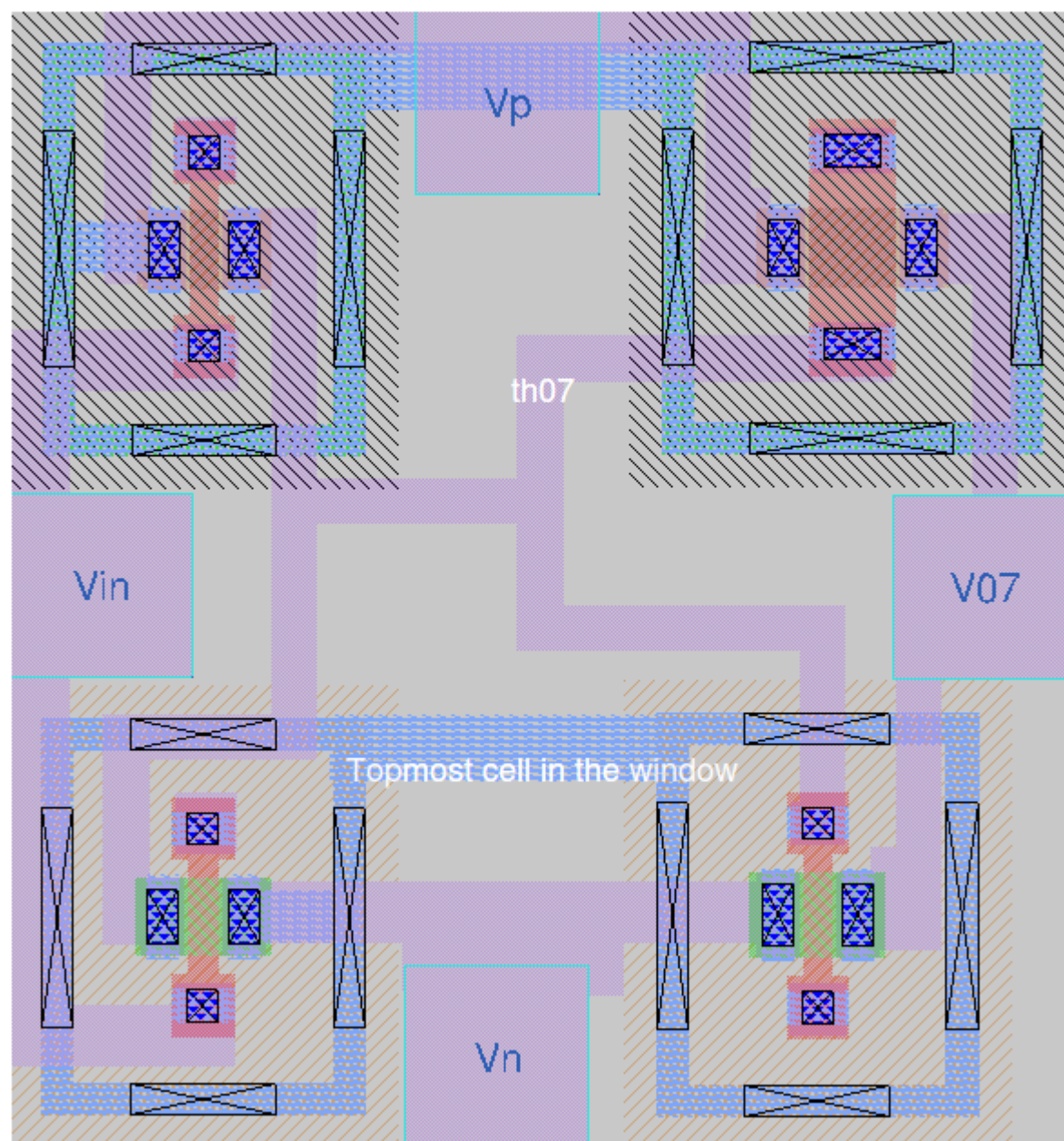


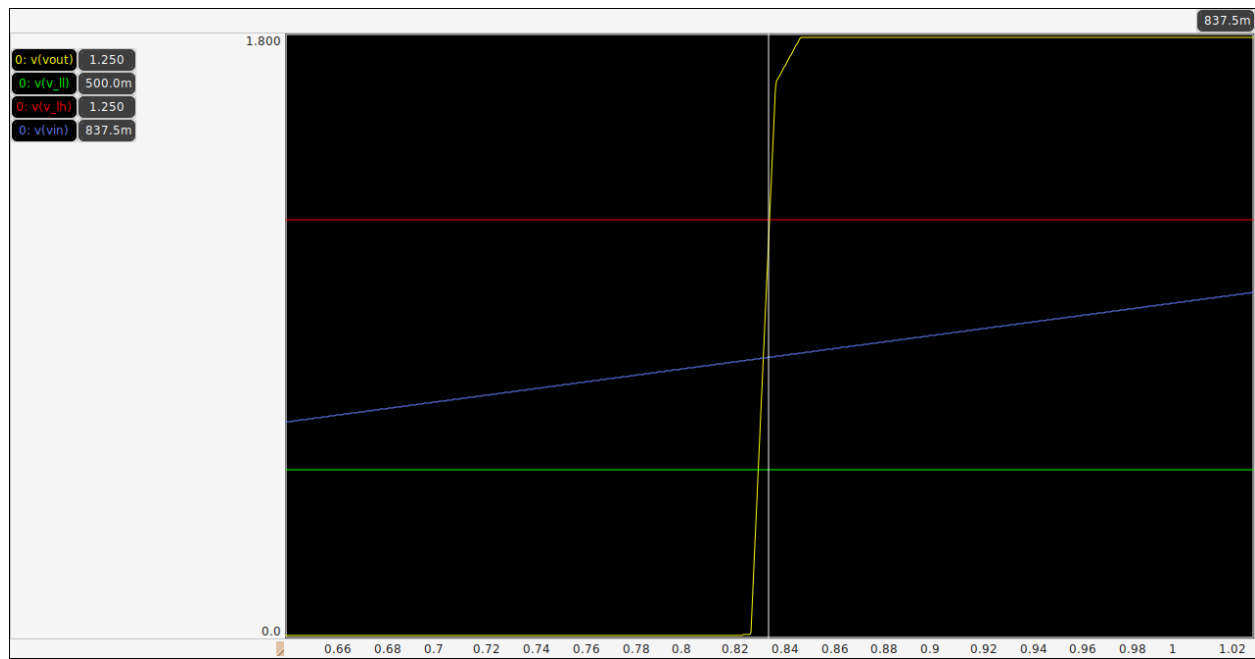
Th. 08 (0.937)



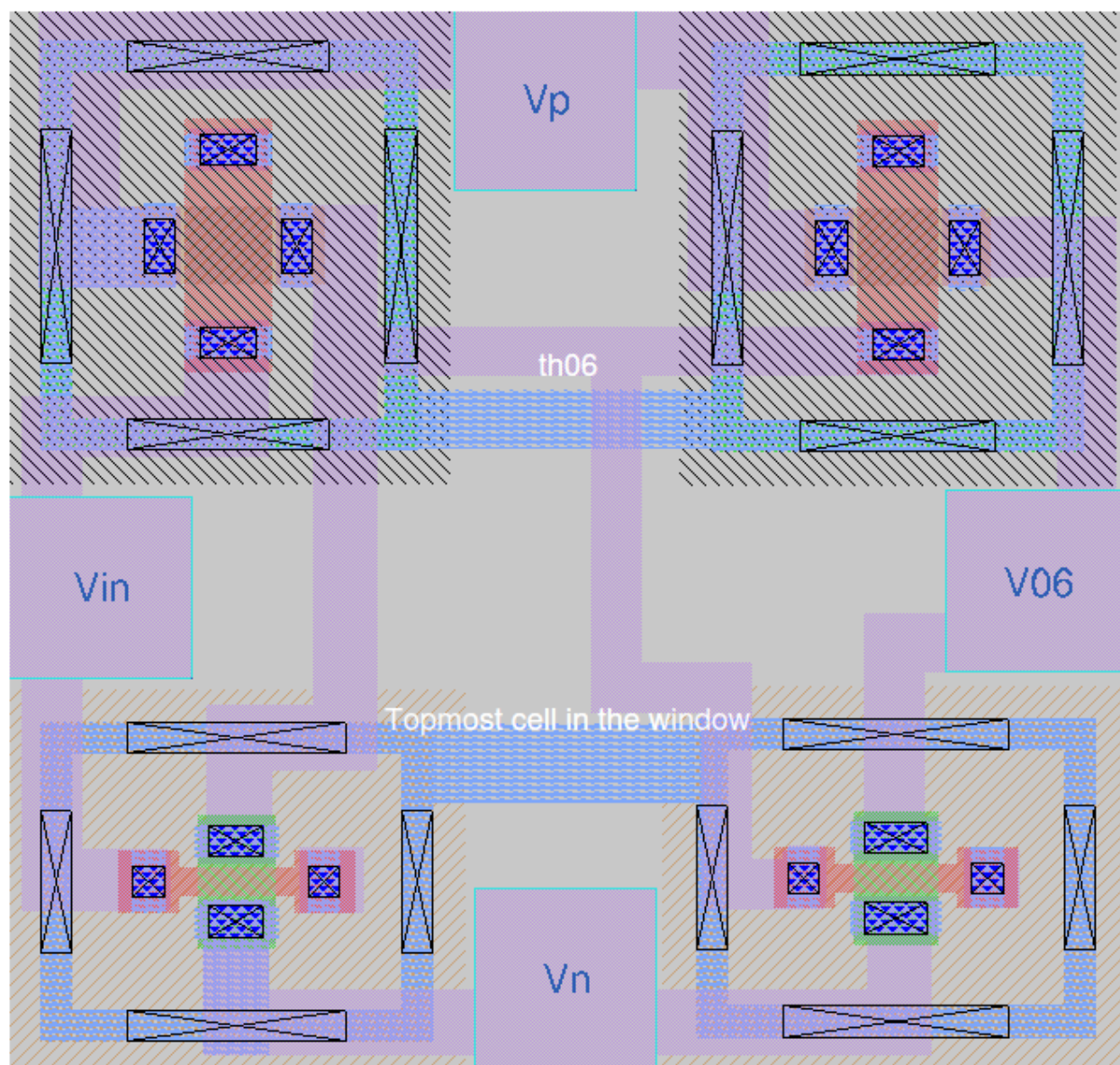


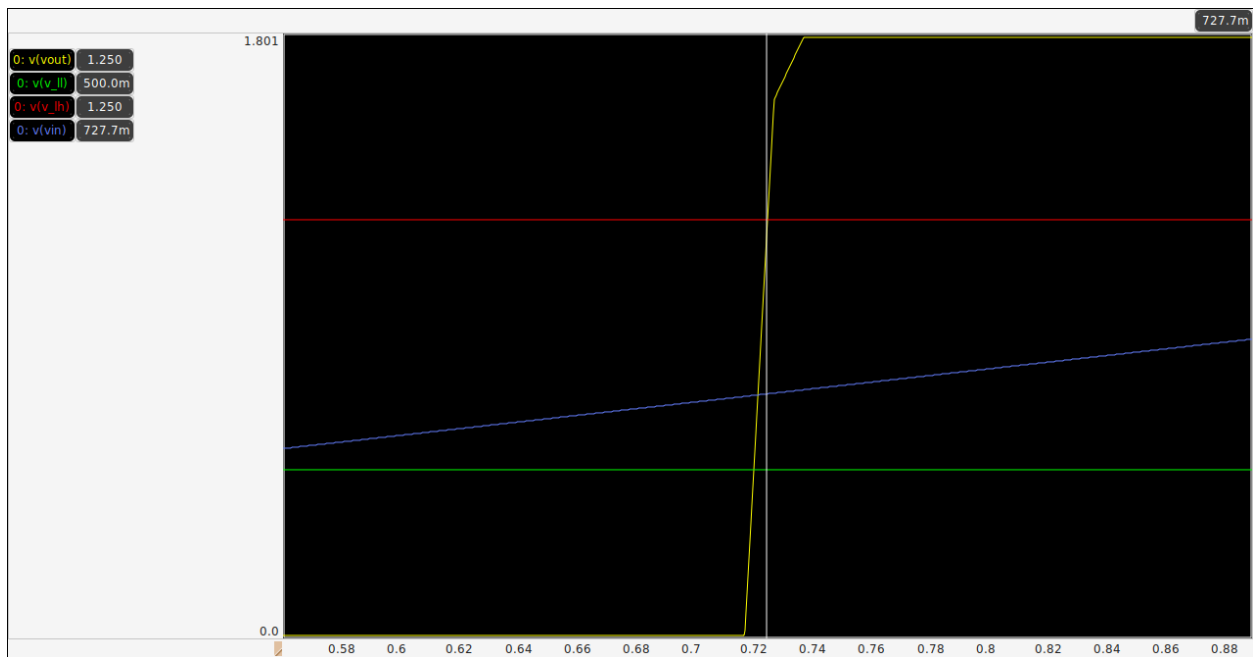
Th. 07 (0.8375)



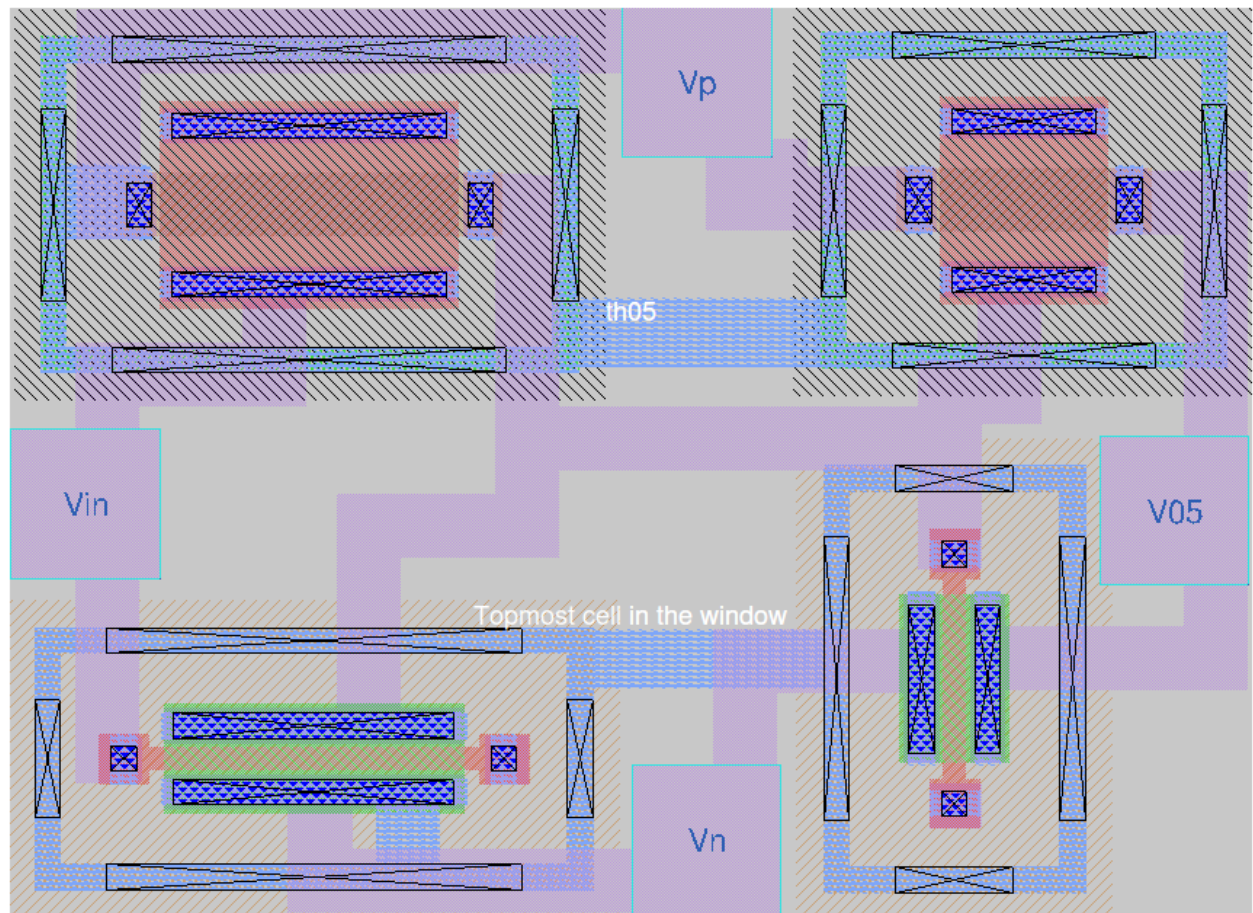


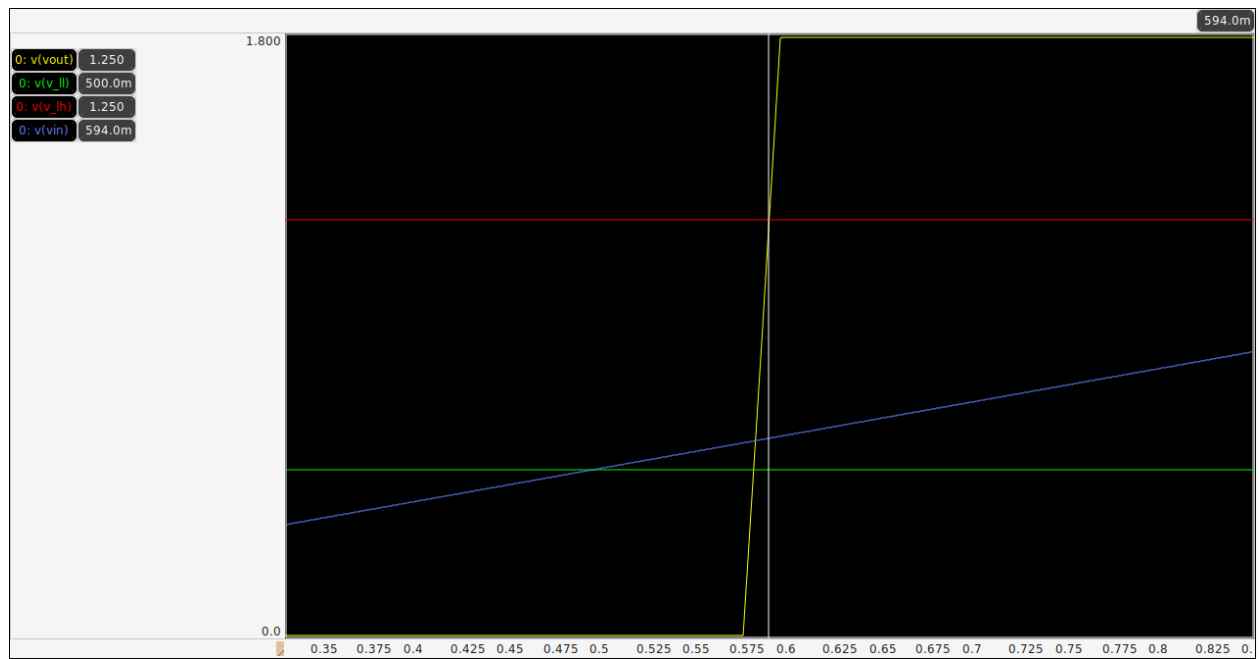
Th. 06 (0.7278)



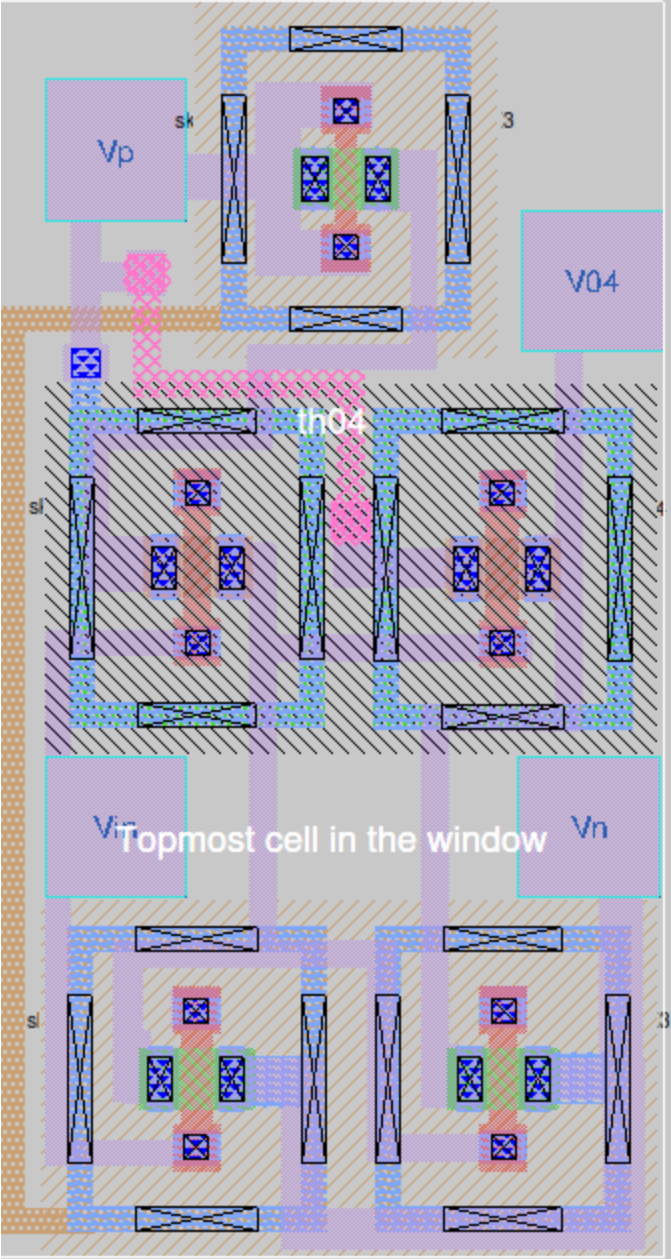


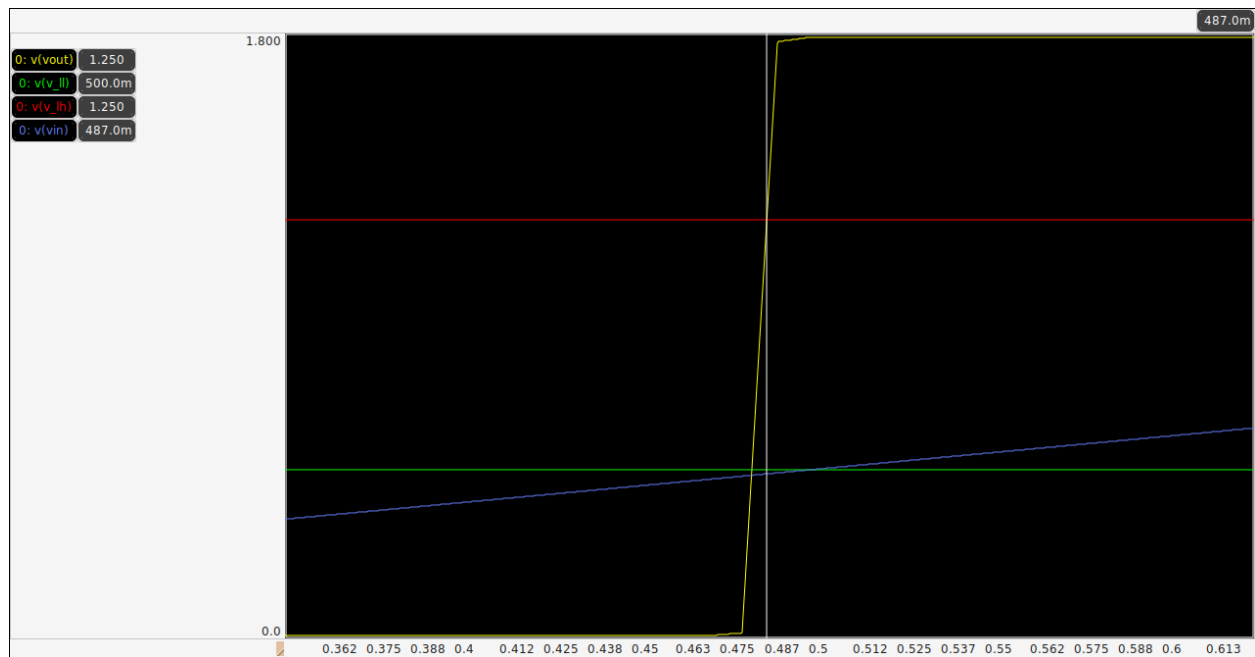
Th. 05



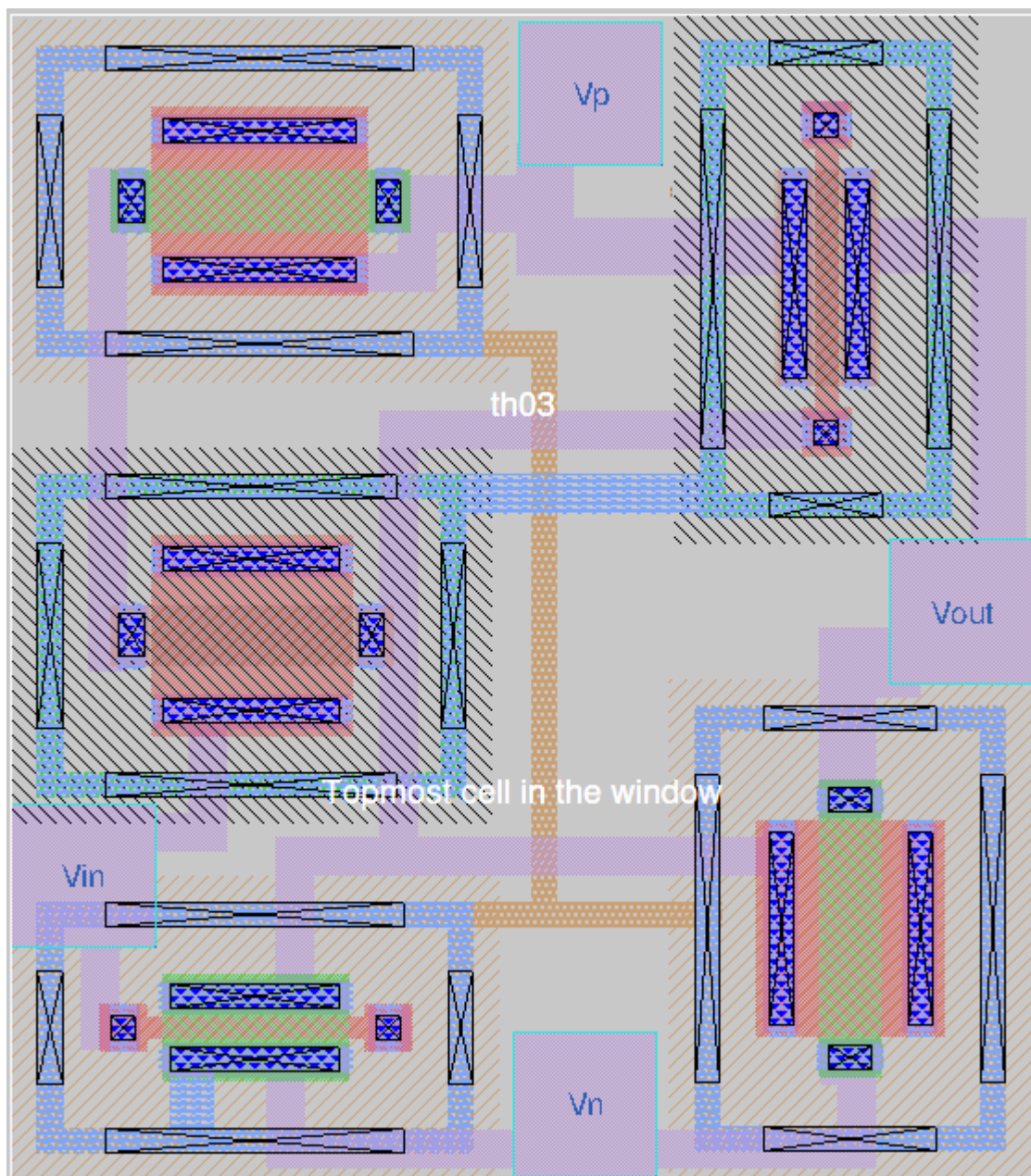


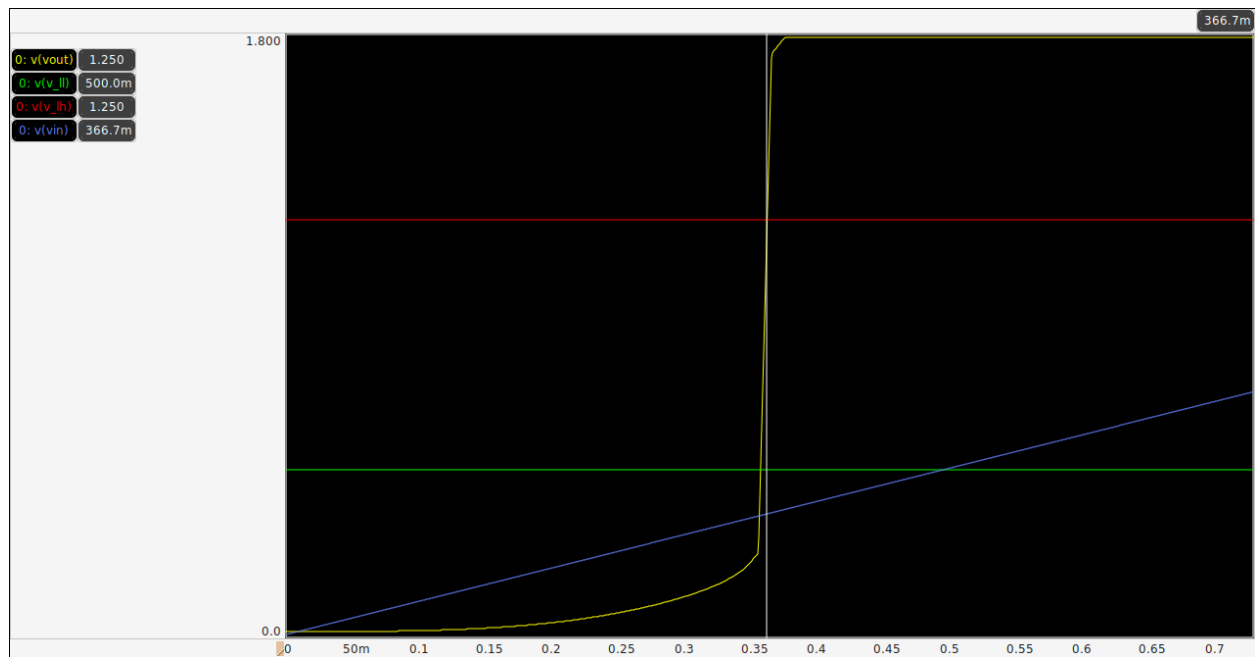
Th. 04 (0.487)



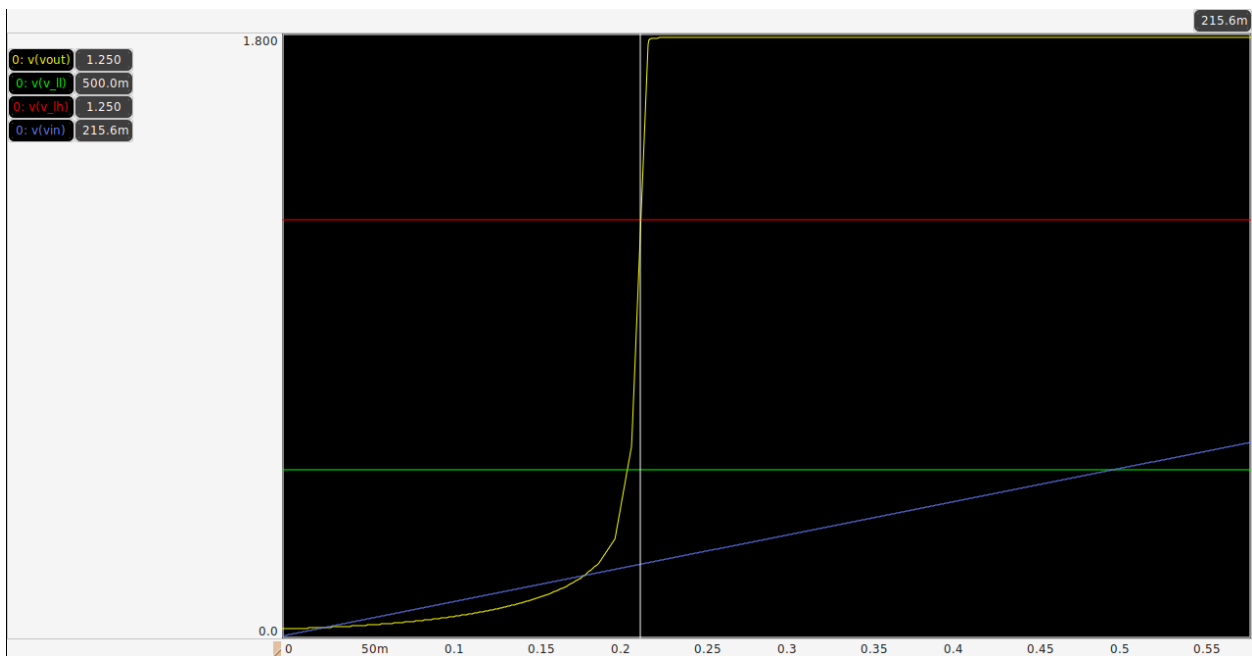
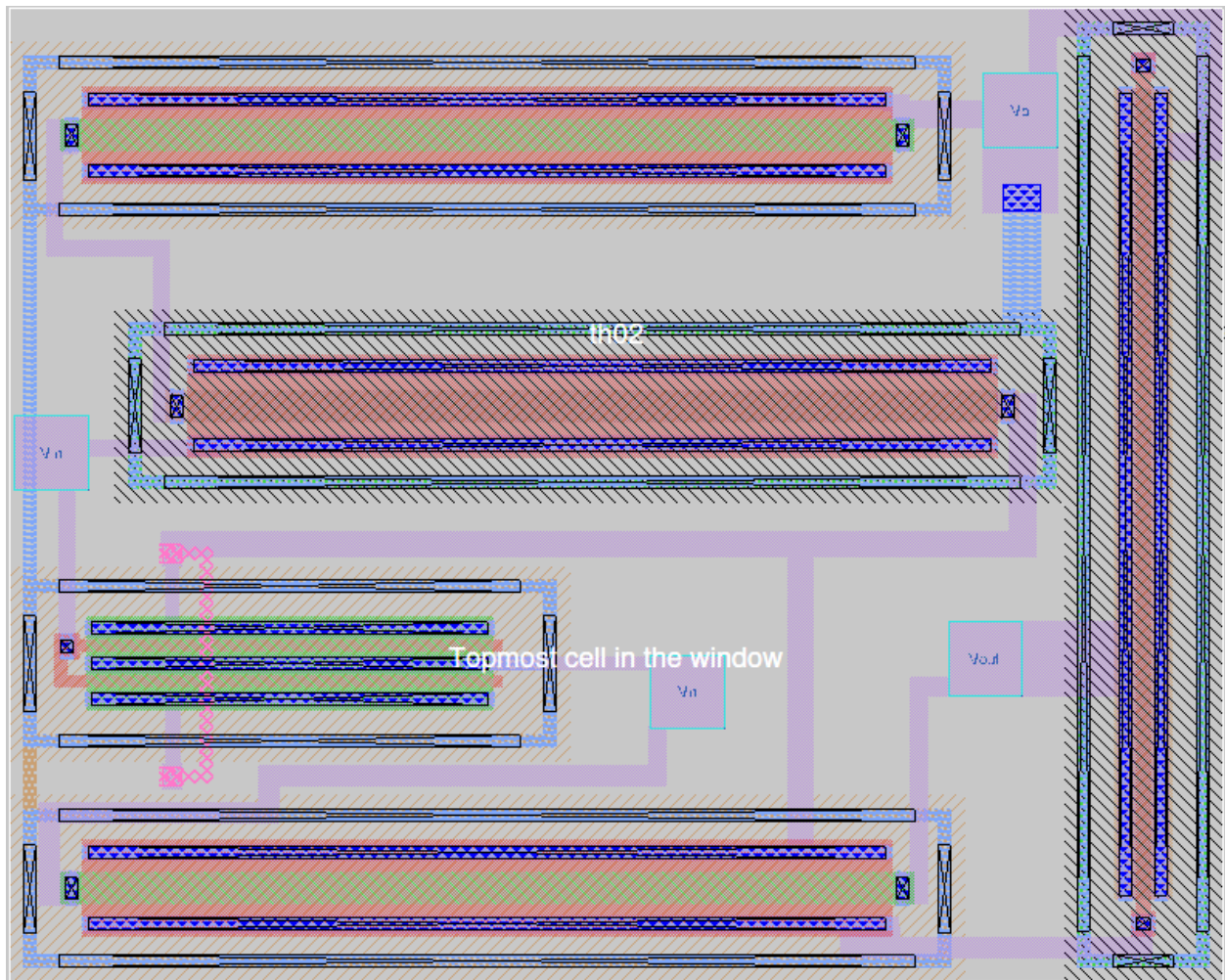


Th. 03 (0.3667)

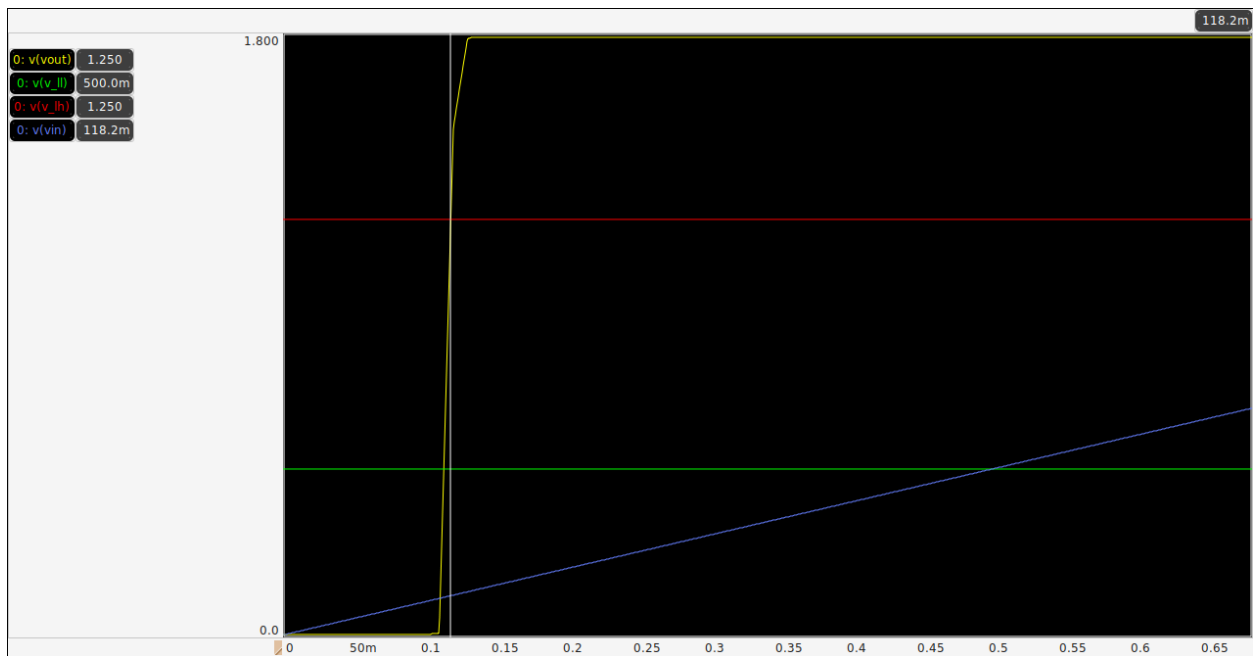
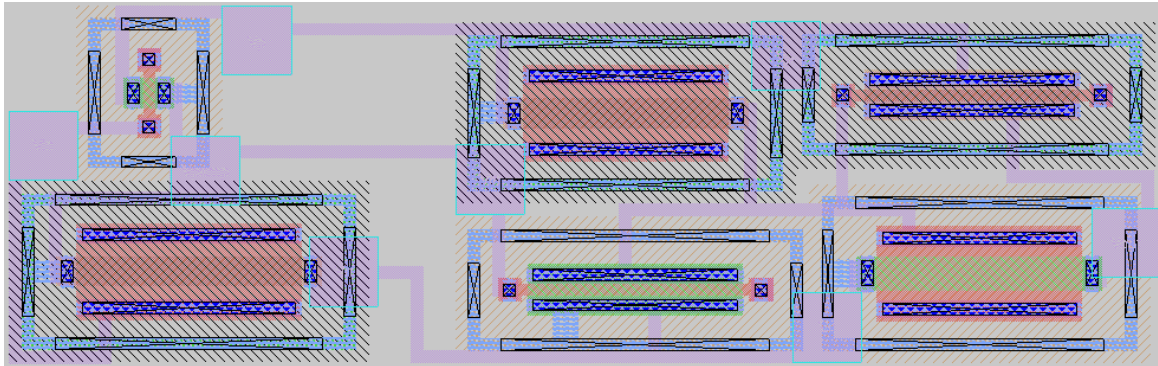




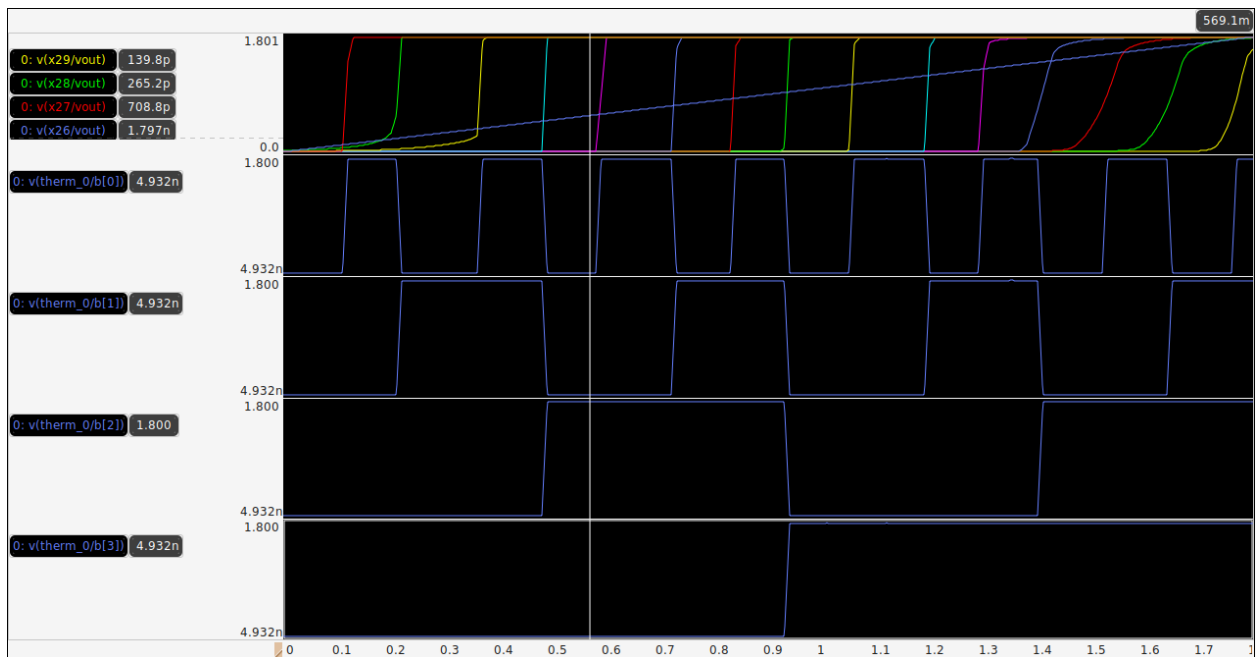
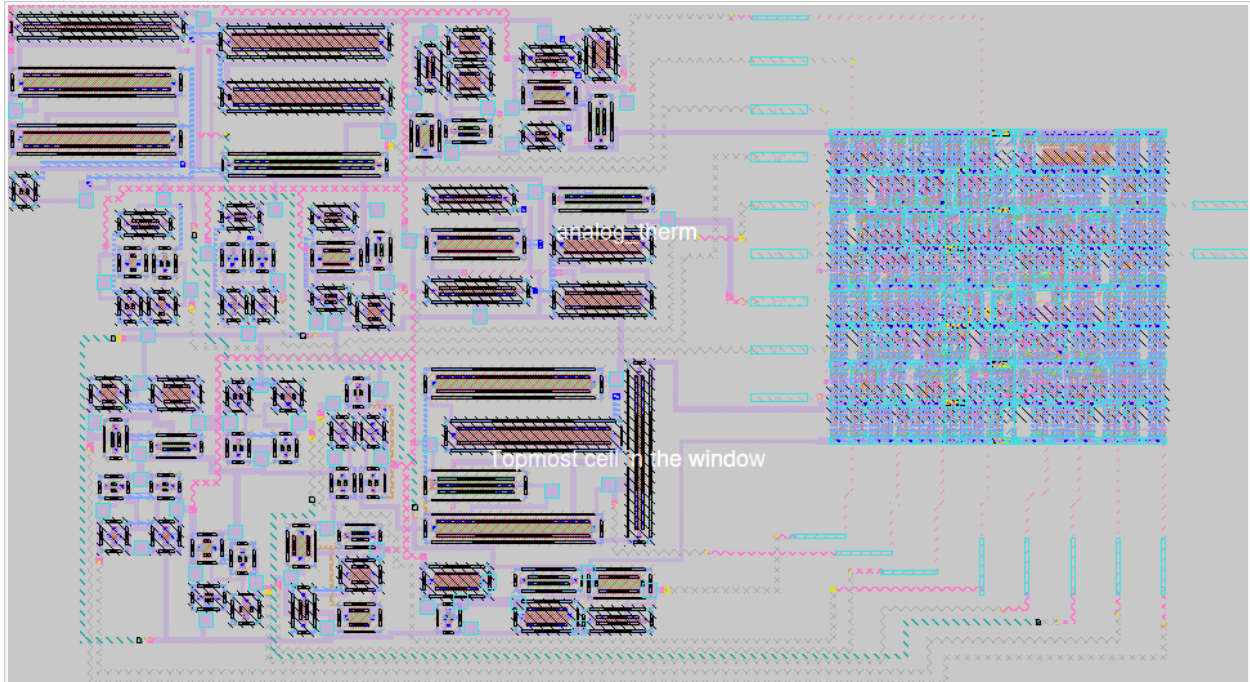
Th. 02 (0.2156)



Th. 01 (0.1182)



Combined Layout:



Run iverilo/icarus

iverilog file.v

vvp a.out
