Topic of the project:
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:

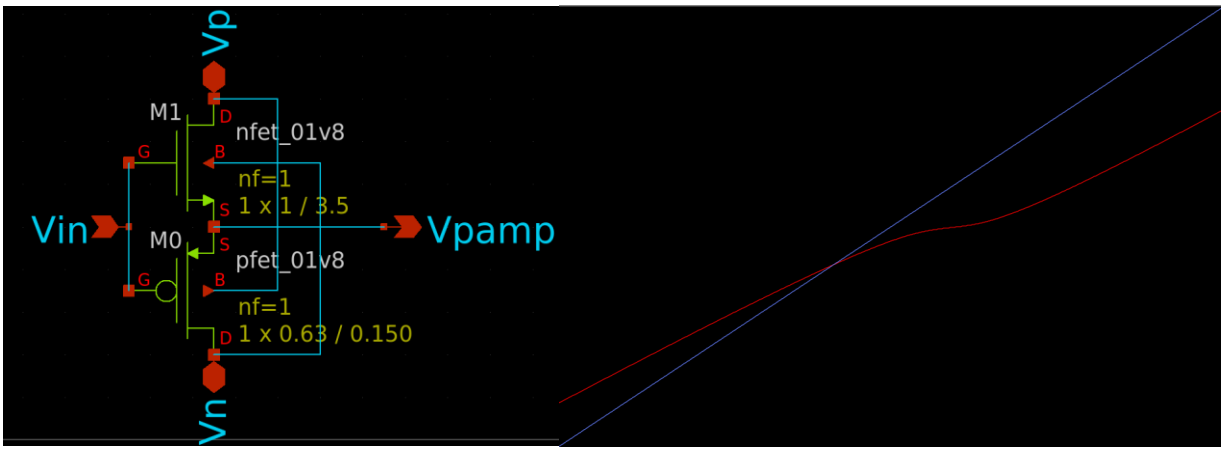
A small CMOS inverter with xschem and the sky130a technology (PDK) and simulate the circuit with ngspice:

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

Preamp:



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

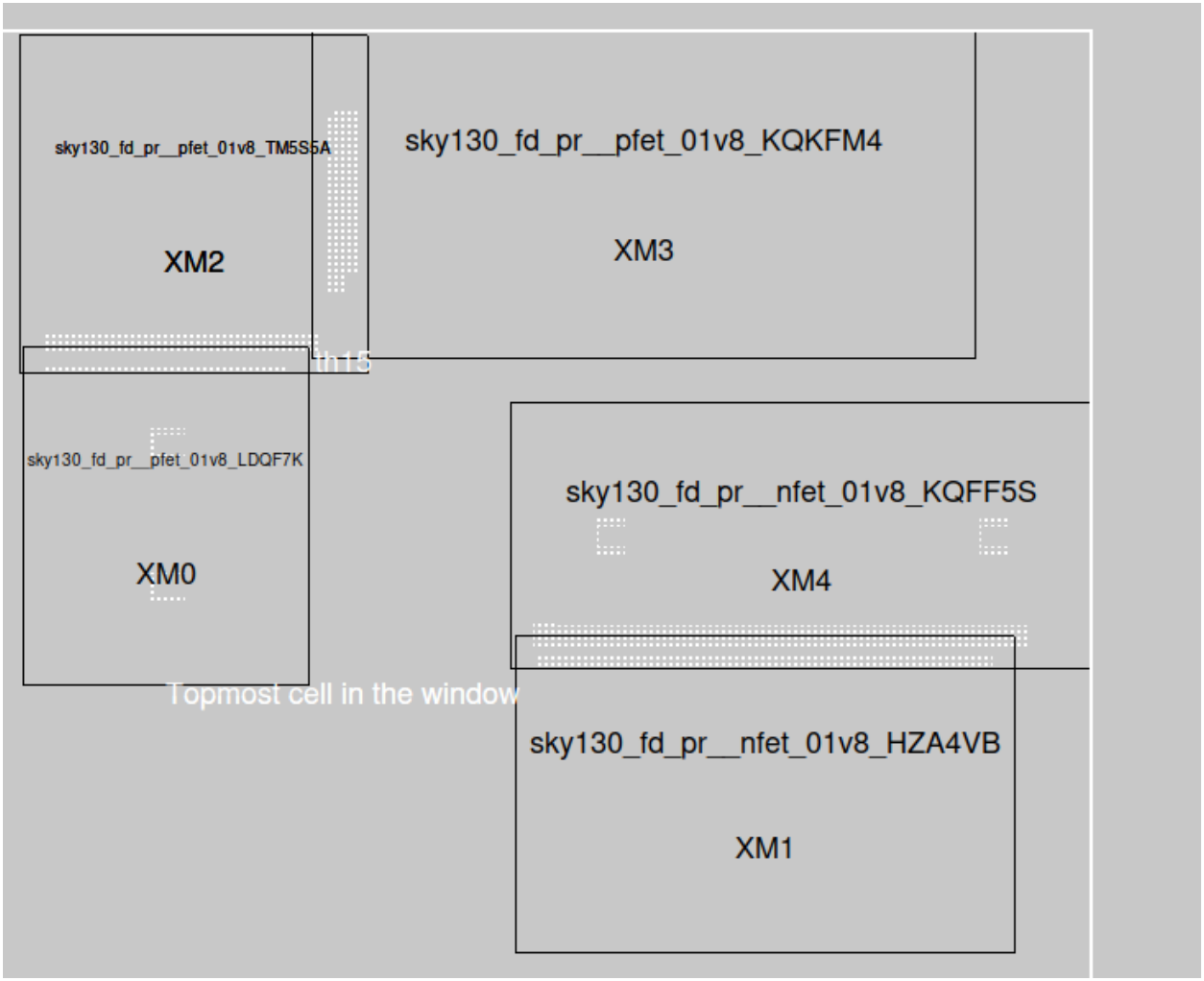
The inverter will switch from logic-low to logic-high when input voltage exceeds 1.785 volts. So, for this inverter the threshold is 1.785 volts.

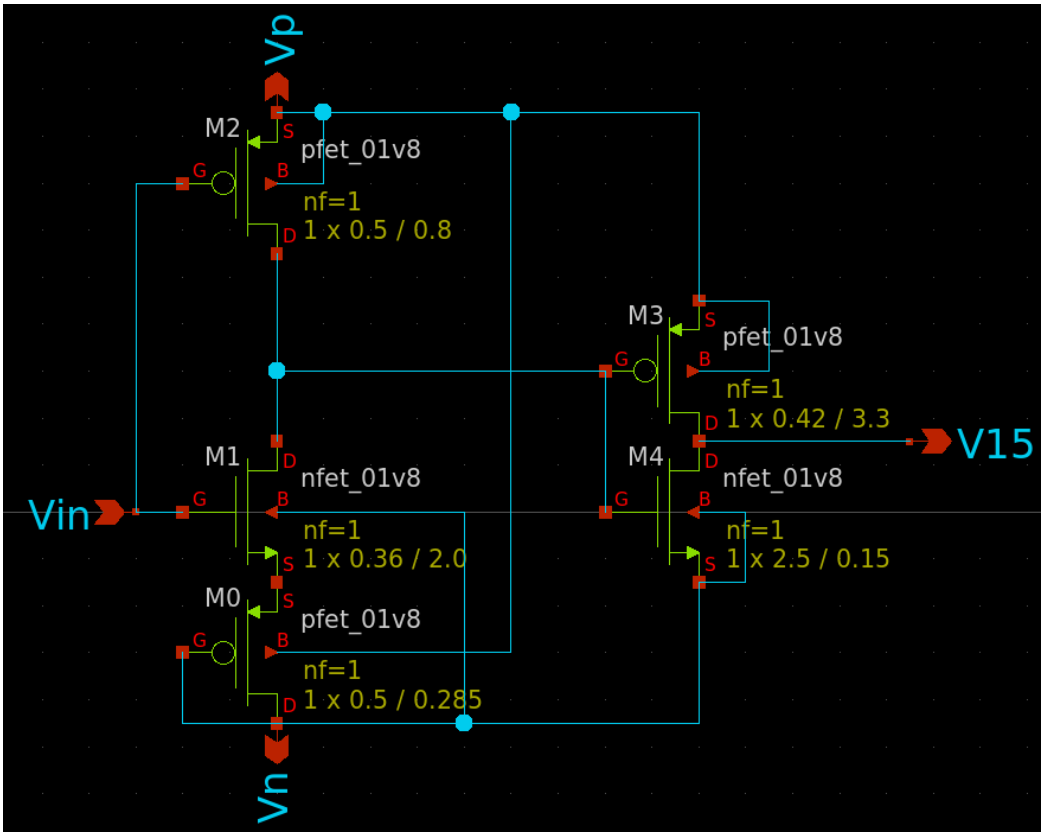
For a 4-bit ADC we need 16 voltage level including zero (0~1.785 volts). The quantization should be 1.785/15=0.119 volts.

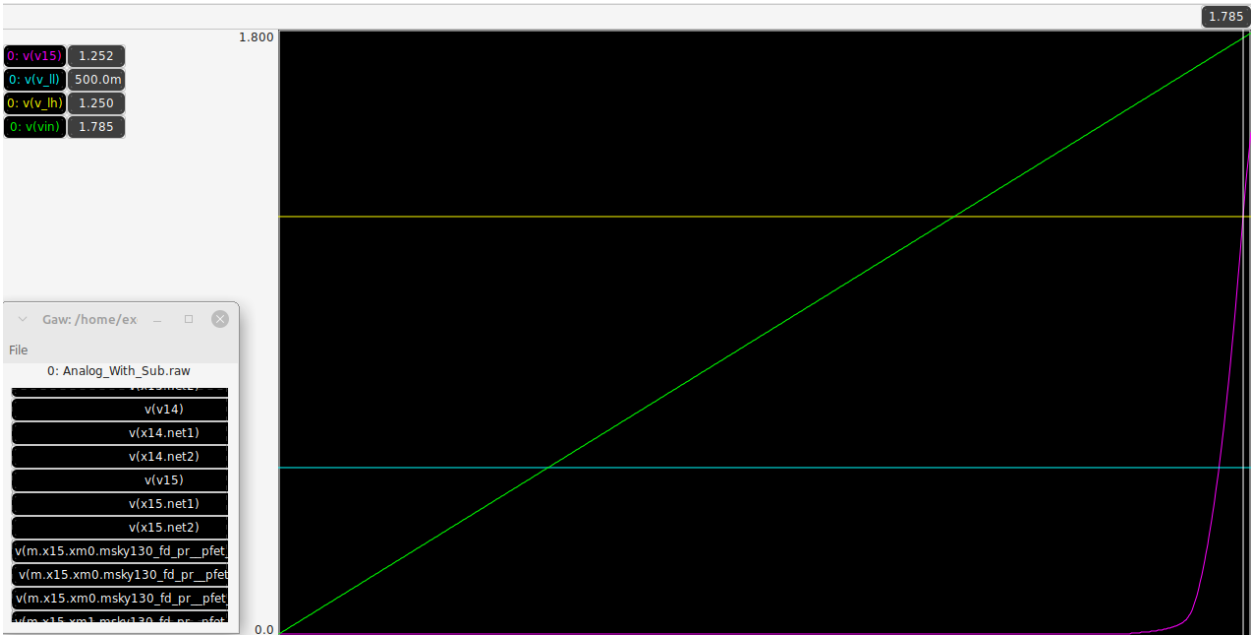
Vdd = 1.8 V

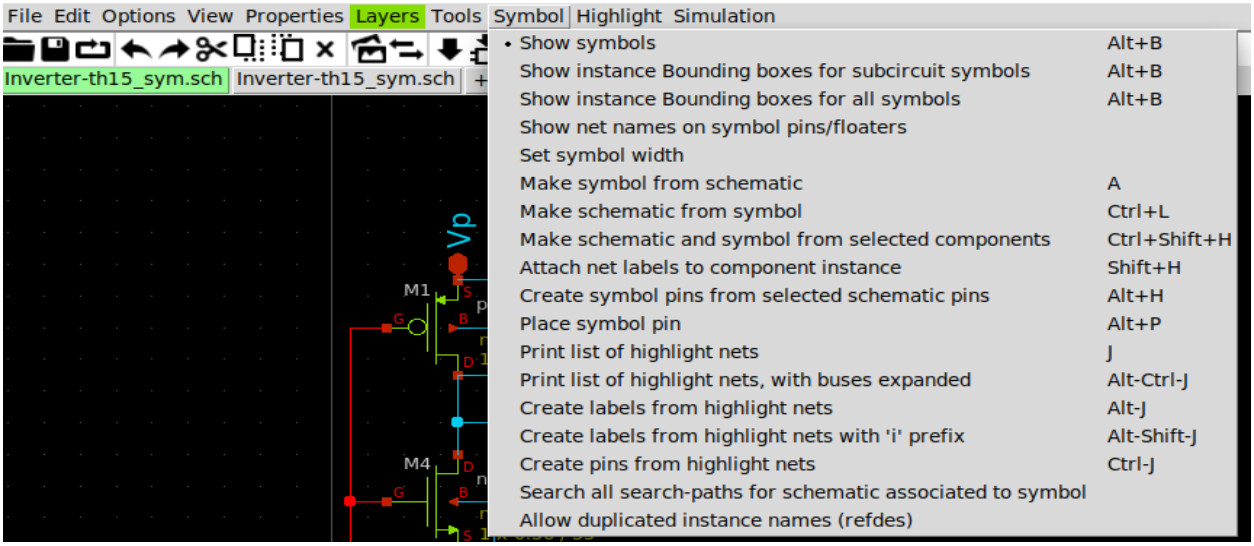
Th. 15: 1.785

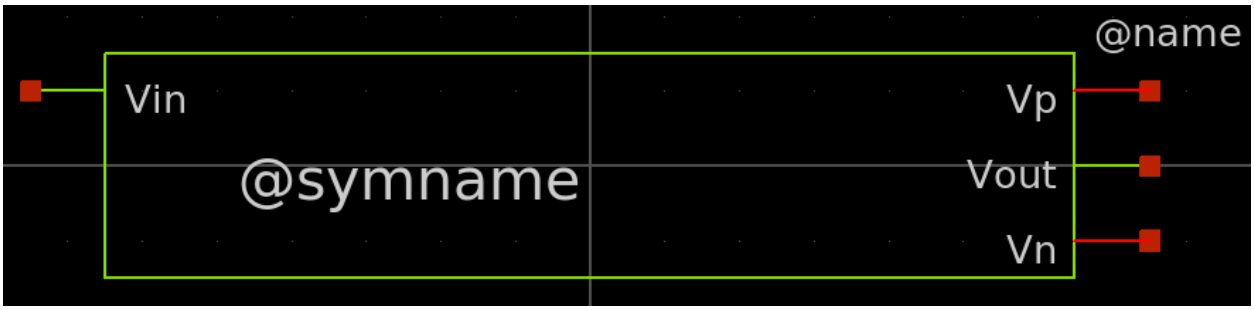
For calculation the schematic and the plot already mentioned earlier. Remove all source to make symbol from schematic. As the symbol will have to be connected to Vdd and GND, those pins are replaced with Vp and Vn accordingly.



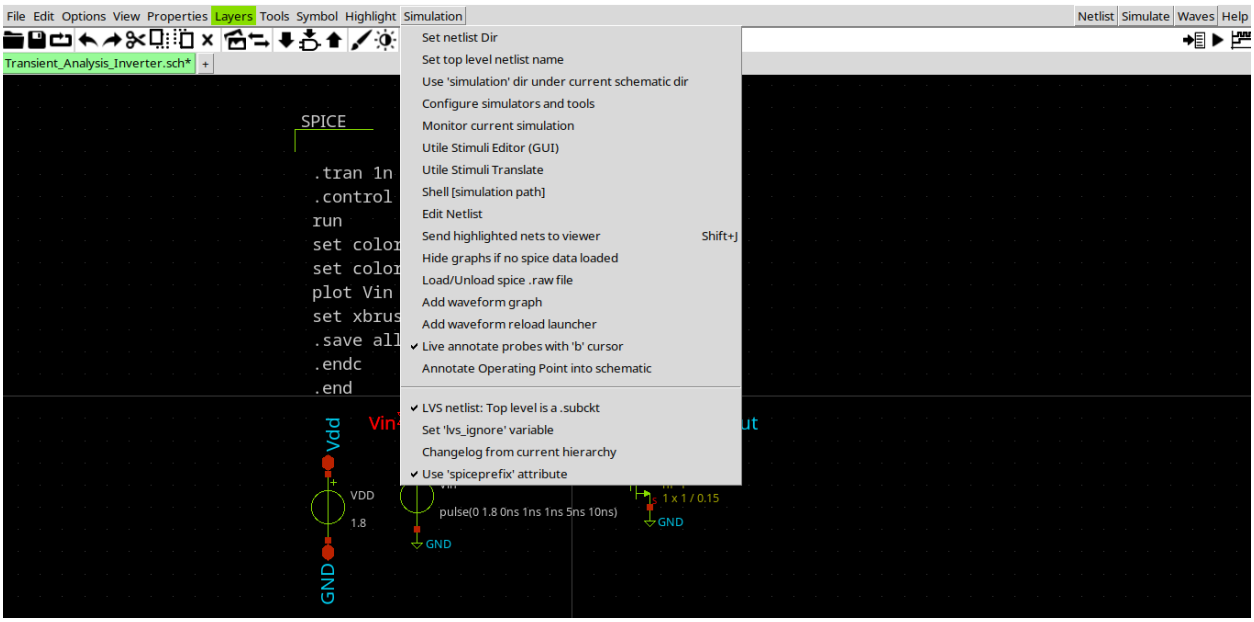

Vp
M2
pfet_01v8
nf=1
1 x 0.5 / 0.8
M3
pfet_01v8
nf=1
1 x 0.42 / 3.3
V15
M1
nfet_01v8
nf=1
1 x 0.36 / 2.0
M4
nfet_01v8
nf=1
1 x 2.5 / 0.15
Vin
M0
pfet_01v8
nf=1
1 x 0.5 / 0.285
Vn


1.785
1.800
0: v(v15) 1.252
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.785
Gaw: /home/ex
File
0: Analog_With_Sub.raw
v(v14)
v(x14.net1)
v(x14.net2)
v(v15)
v(x15.net1)
v(x15.net2)
0.0

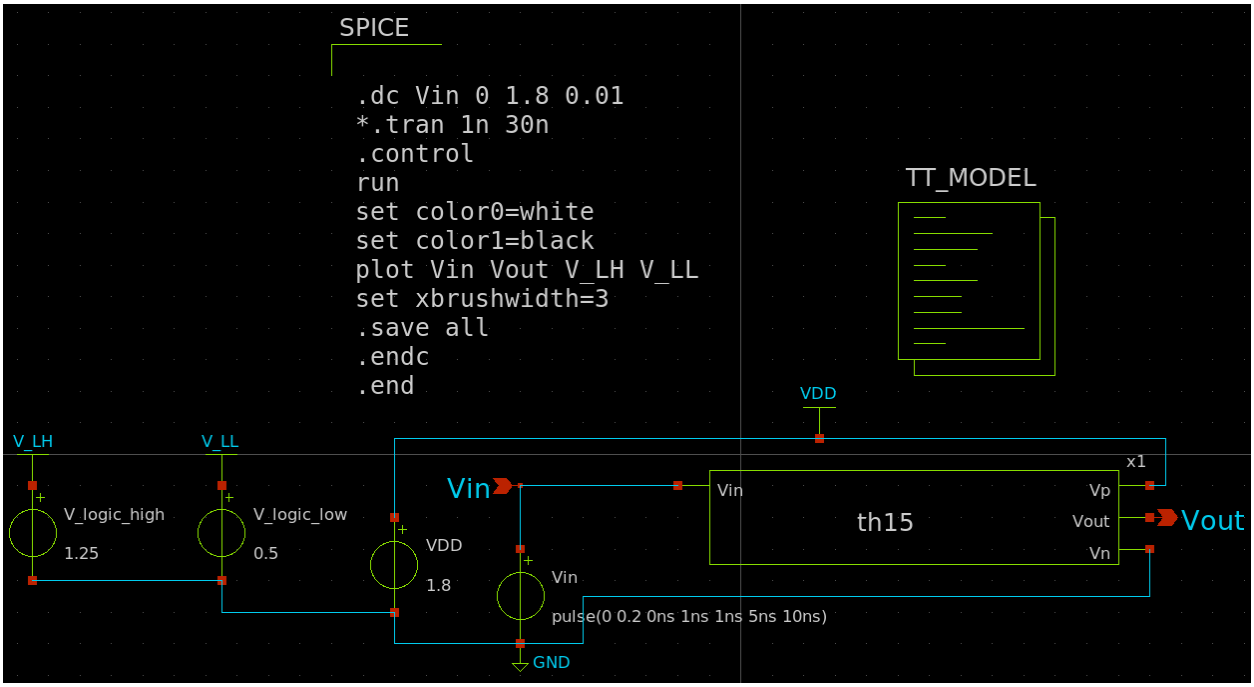




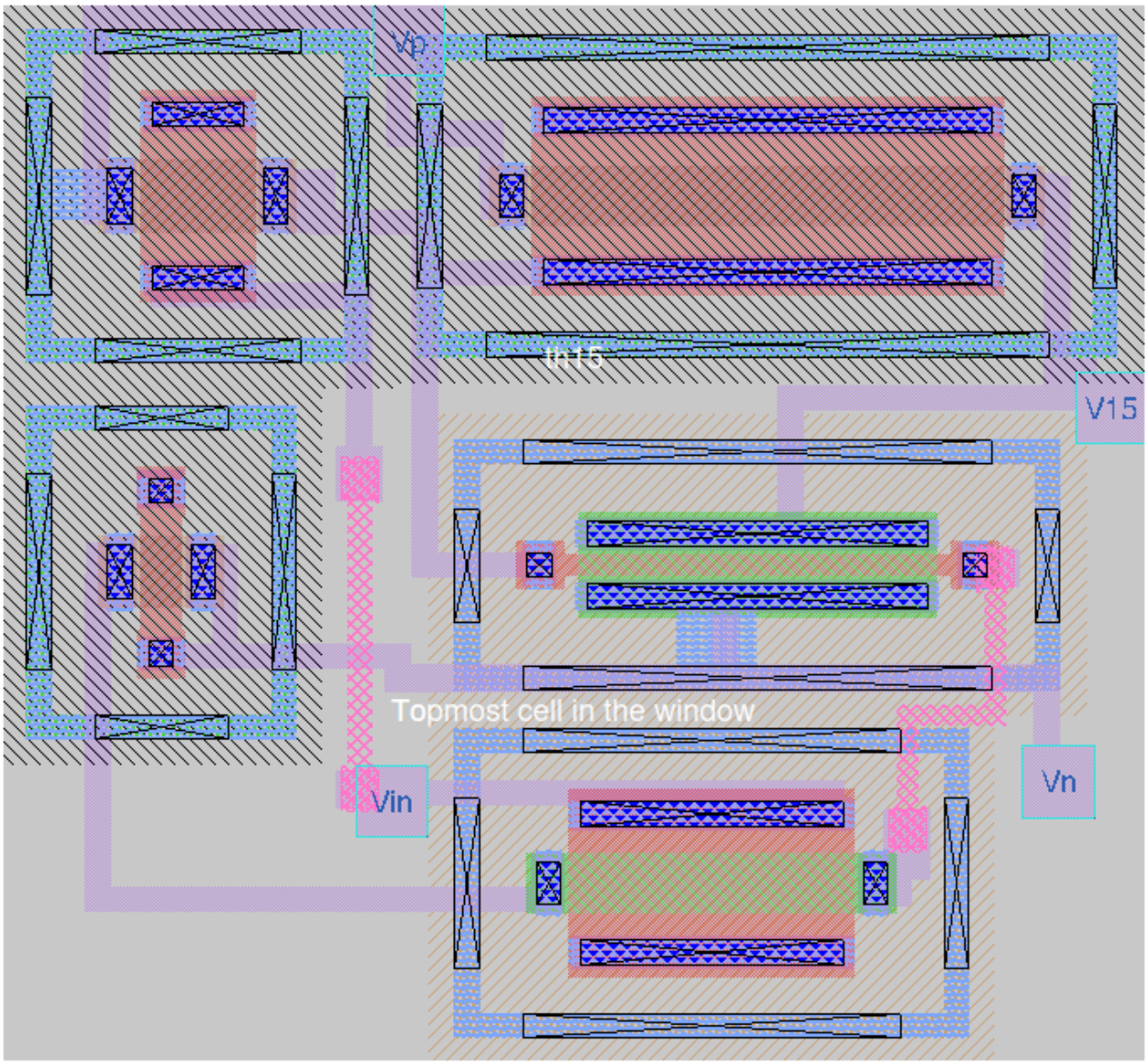
Magic=>before importing netlist to ngspice check LVS net



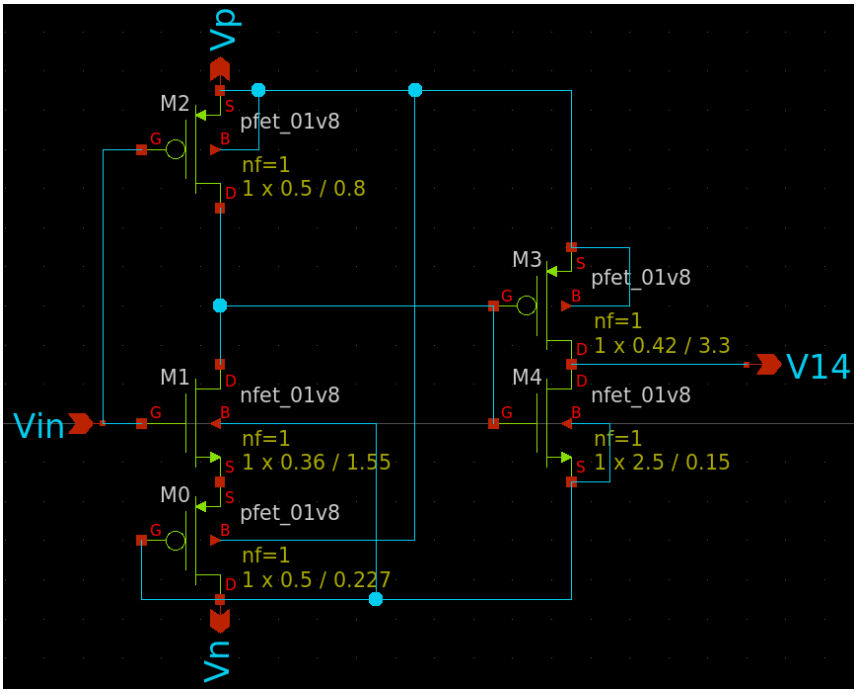
The simplified schematic is as follows:

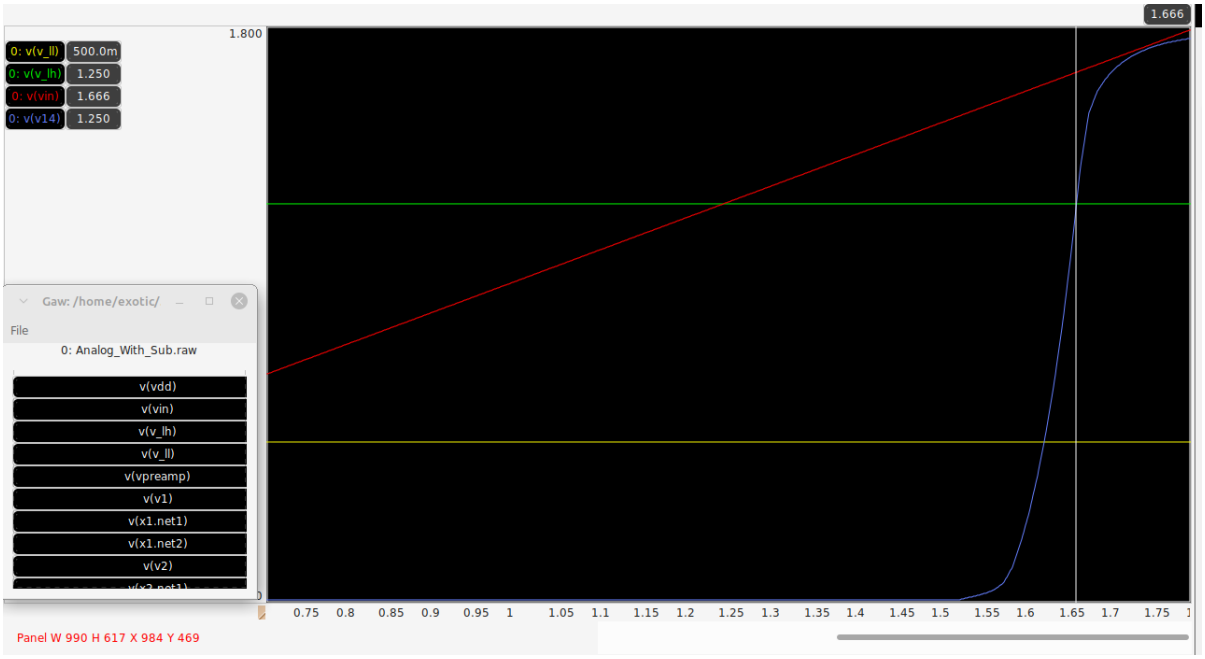


The plot is similar to the previous figure.

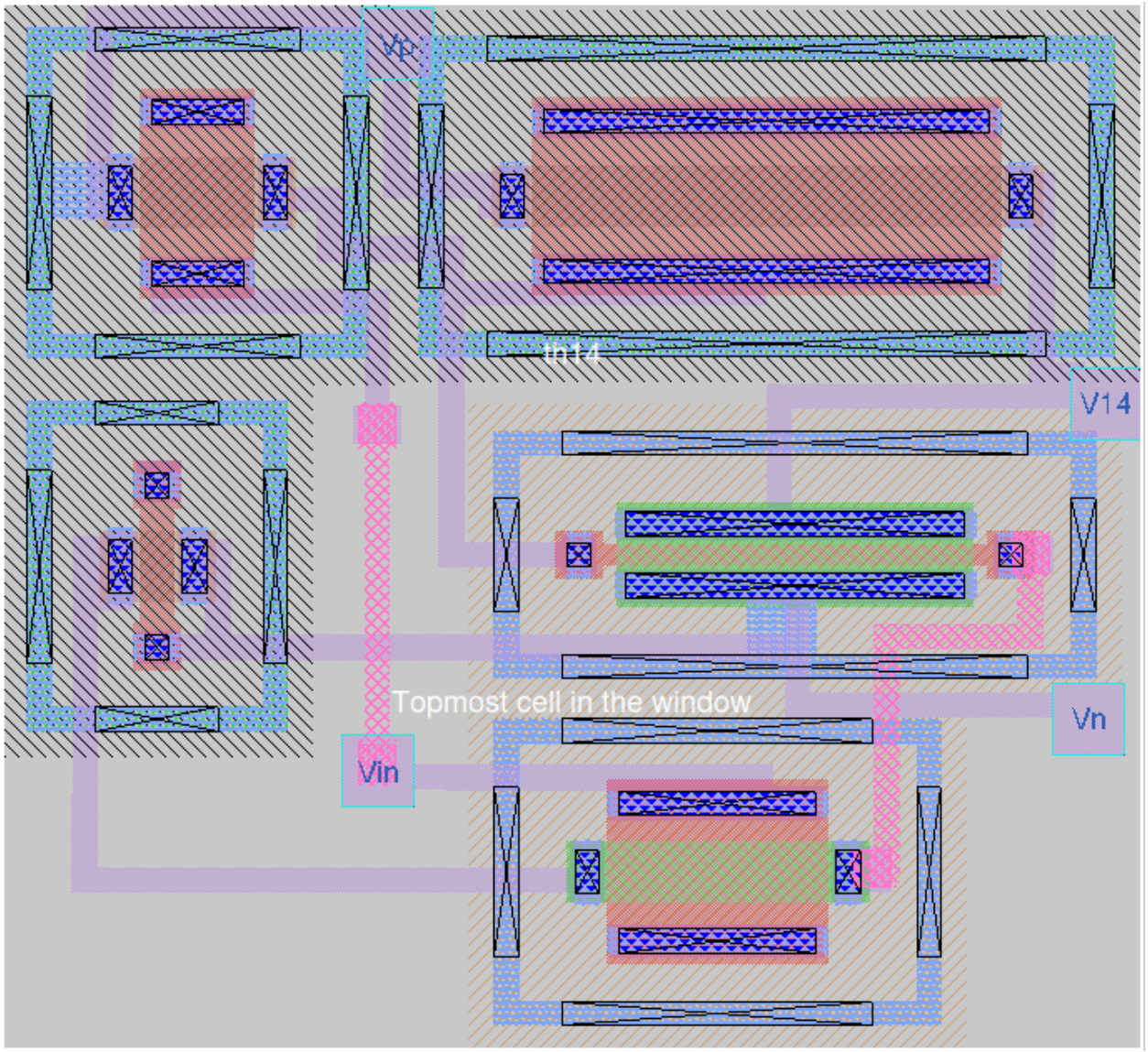


Th. 14: **1.666**

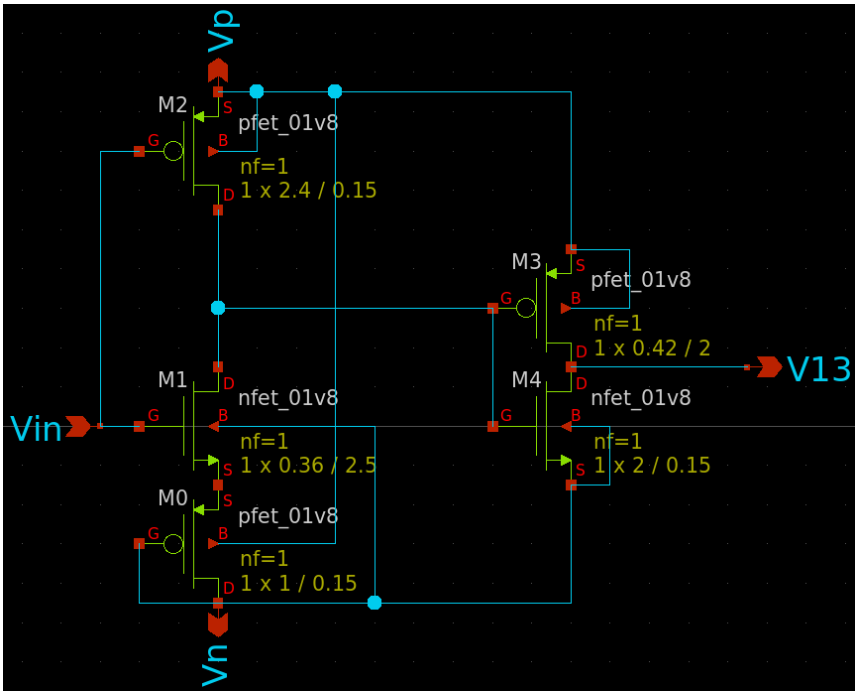


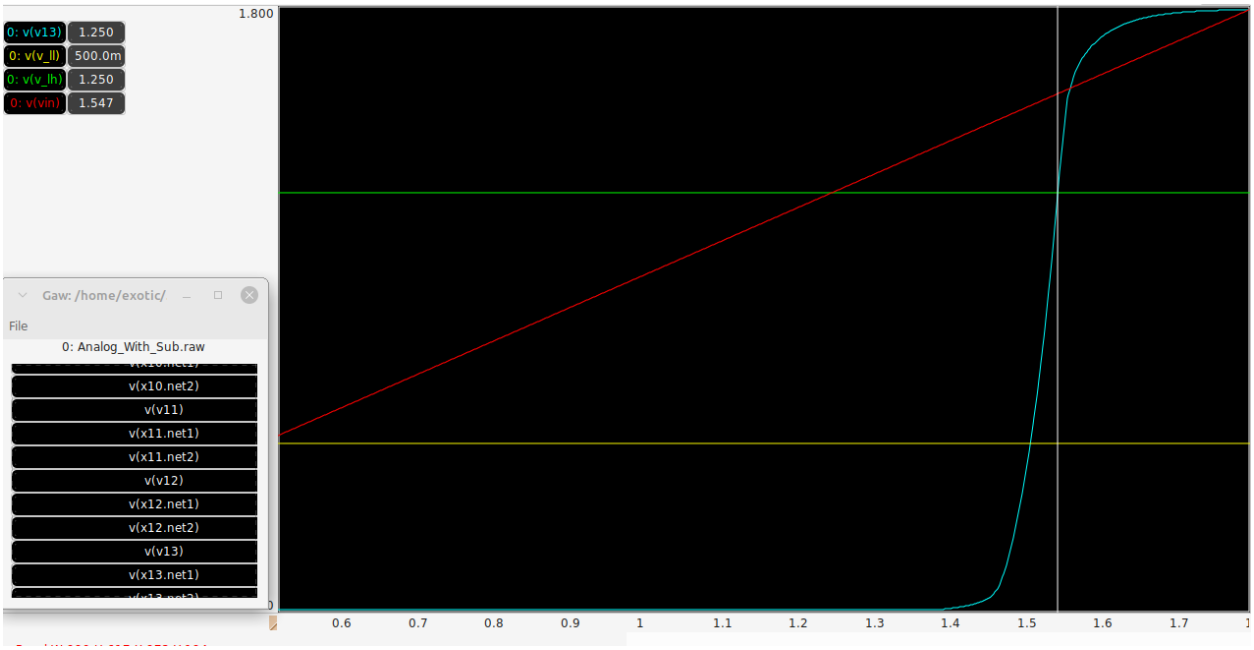


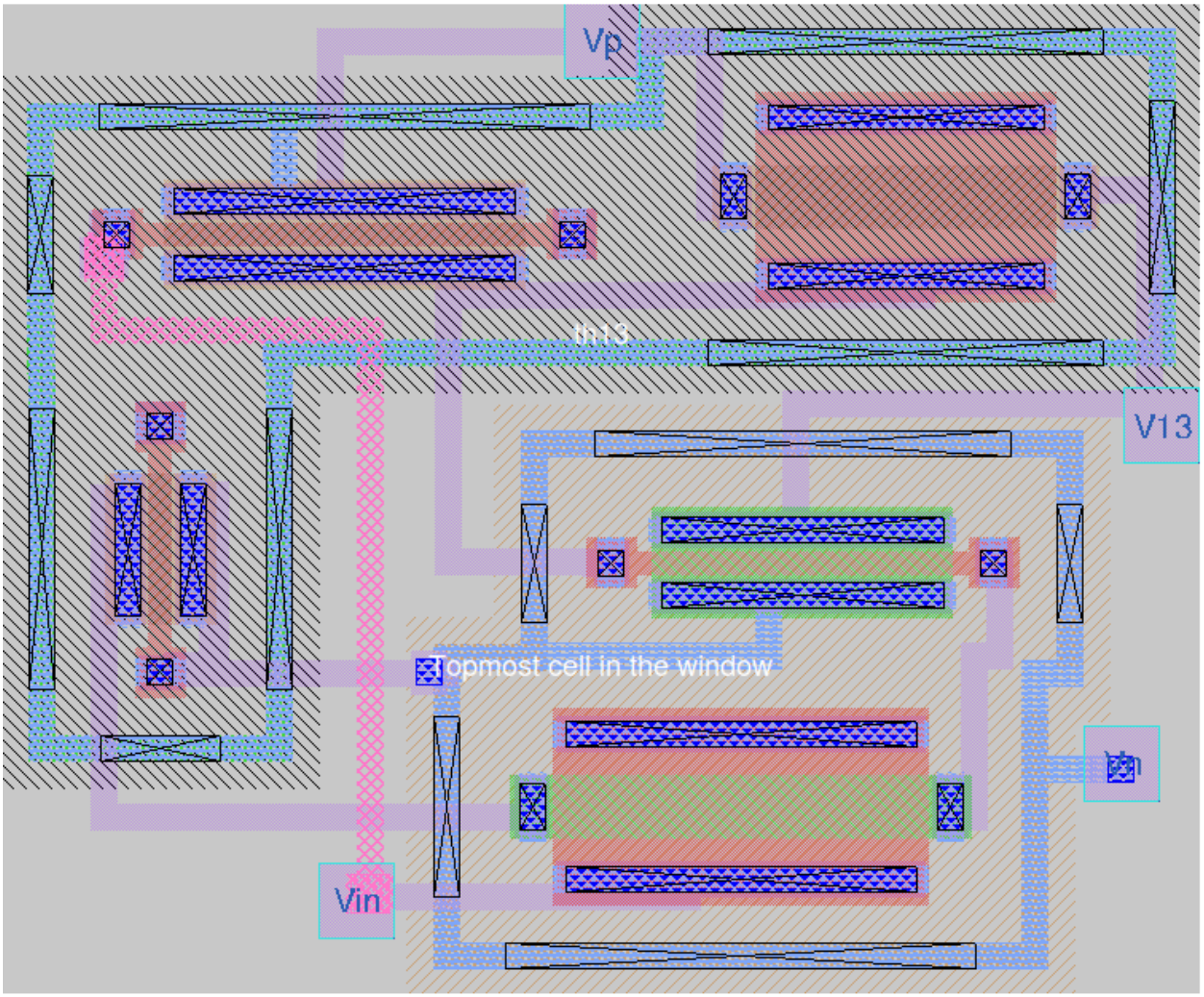
Rest of the figures are similar to the previous figures.



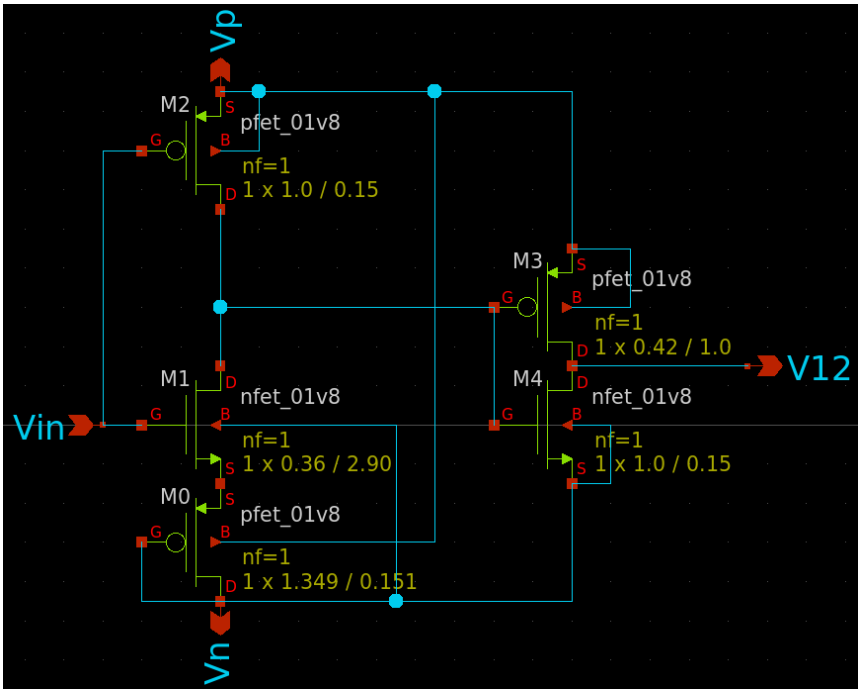
Th. 13: 1.547:

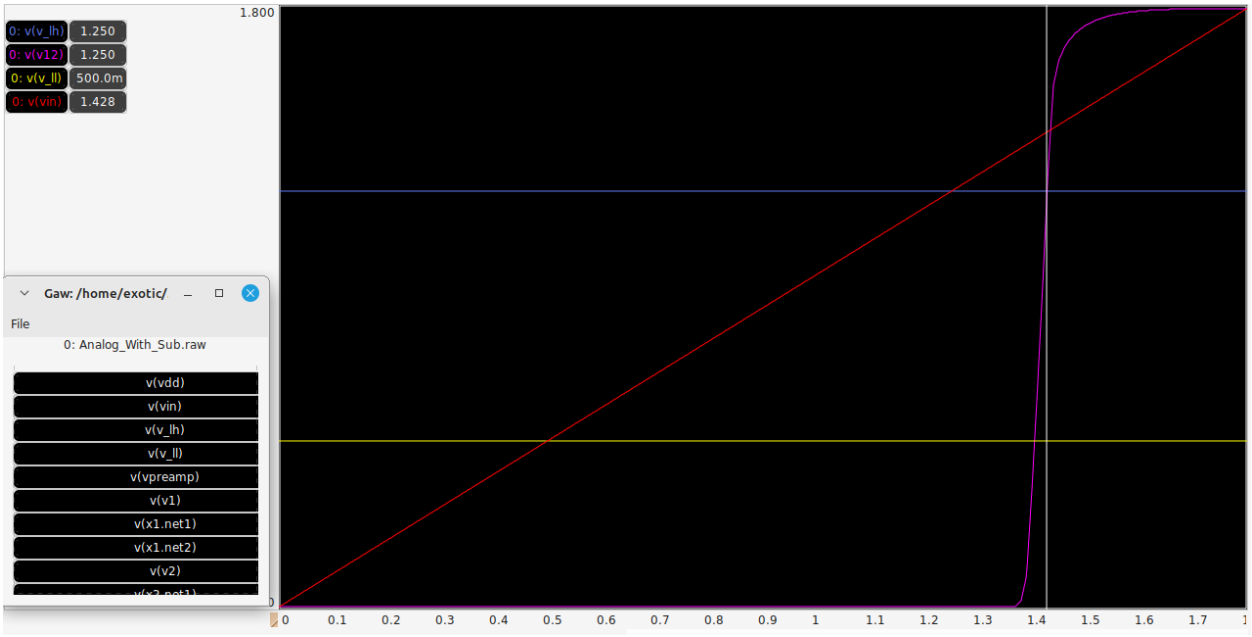

Vp
M2
pfet_01v8
G
B
S
D
nf=1
1 x 2.4 / 0.15
M3
pfet_01v8
nf=1
1 x 0.42 / 2
V13
M1
nfet_01v8
Vin
nf=1
1 x 0.36 / 2.5
M4
nfet_01v8
nf=1
1 x 2 / 0.15
M0
pfet_01v8
nf=1
1 x 1 / 0.15
Vn


0: v(v13) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.547
1.800
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(x10.net2)
v(v11)
v(x11.net1)
v(x11.net2)
v(v12)
v(x12.net1)
v(x12.net2)
v(v13)
v(x13.net1)
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

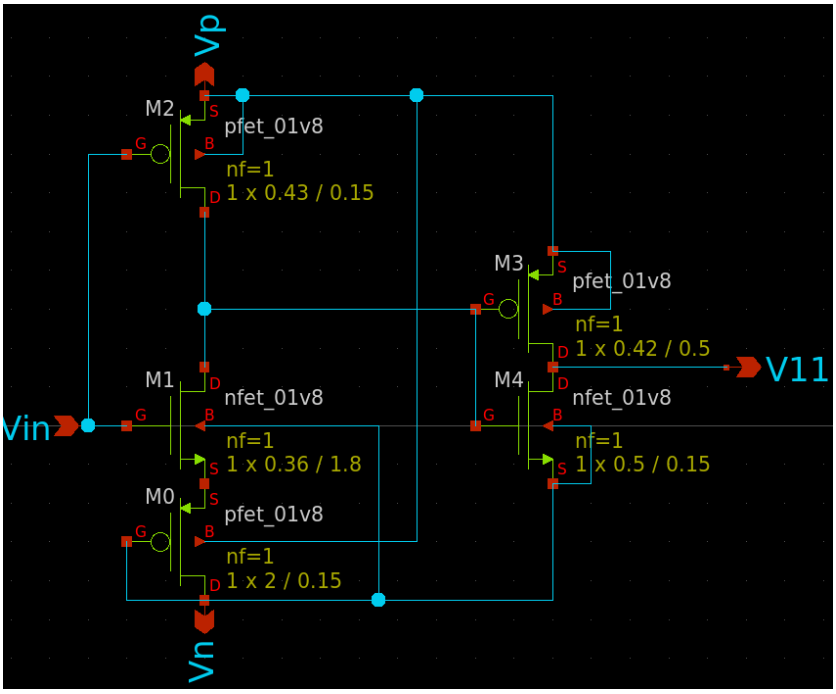


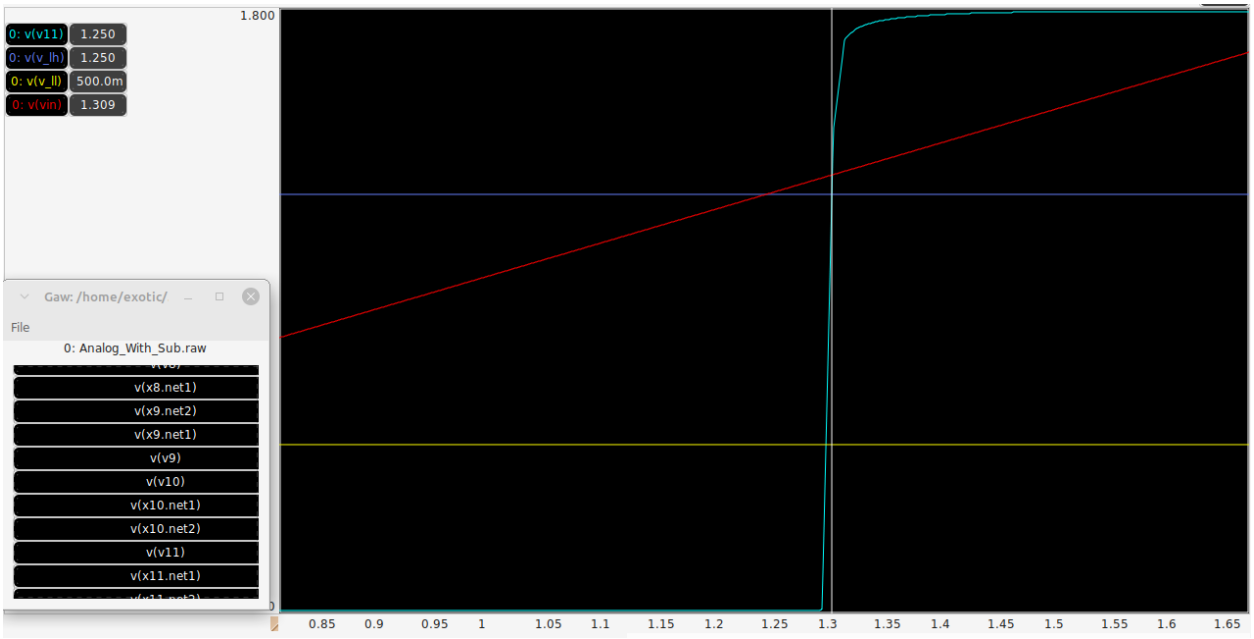
Th. 12: **1.428**


Vp
M2
pfet_01v8
nf=1
1 x 1.0 / 0.15
M3
pfet_01v8
nf=1
1 x 0.42 / 1.0
V12
M1
nfet_01v8
Vin
nf=1
1 x 0.36 / 2.90
M4
nfet_01v8
nf=1
1 x 1.0 / 0.15
M0
pfet_01v8
nf=1
1 x 1.349 / 0.151
Vn

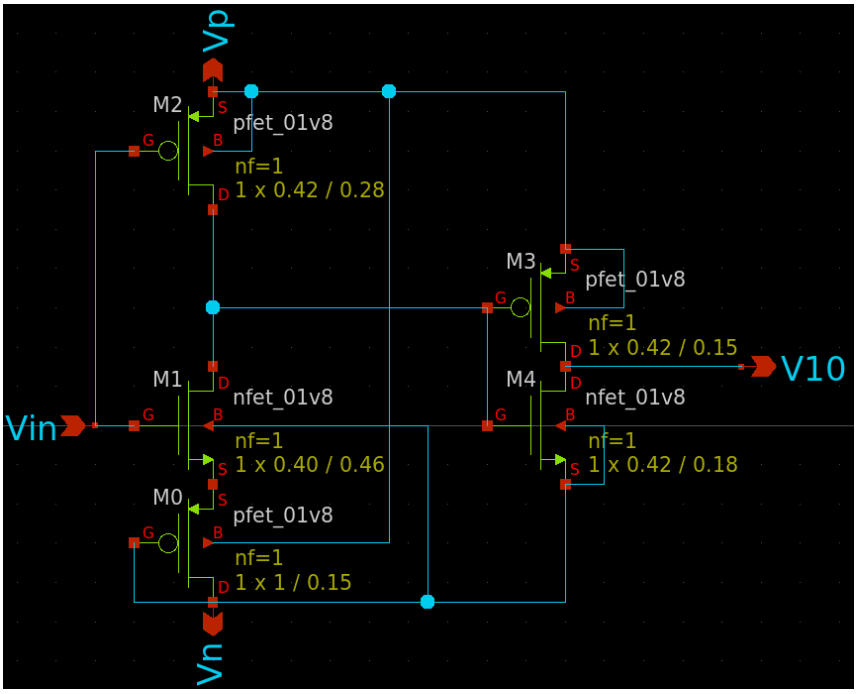

0: v(v_lh) 1.250
0: v(v12) 1.250
0: v(v_ll) 500.0m
0: v(vin) 1.428
1.800
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(vdd)
v(vin)
v(v_lh)
v(v_ll)
v(vpreamp)
v(v1)
v(x1.net1)
v(x1.net2)
v(v2)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

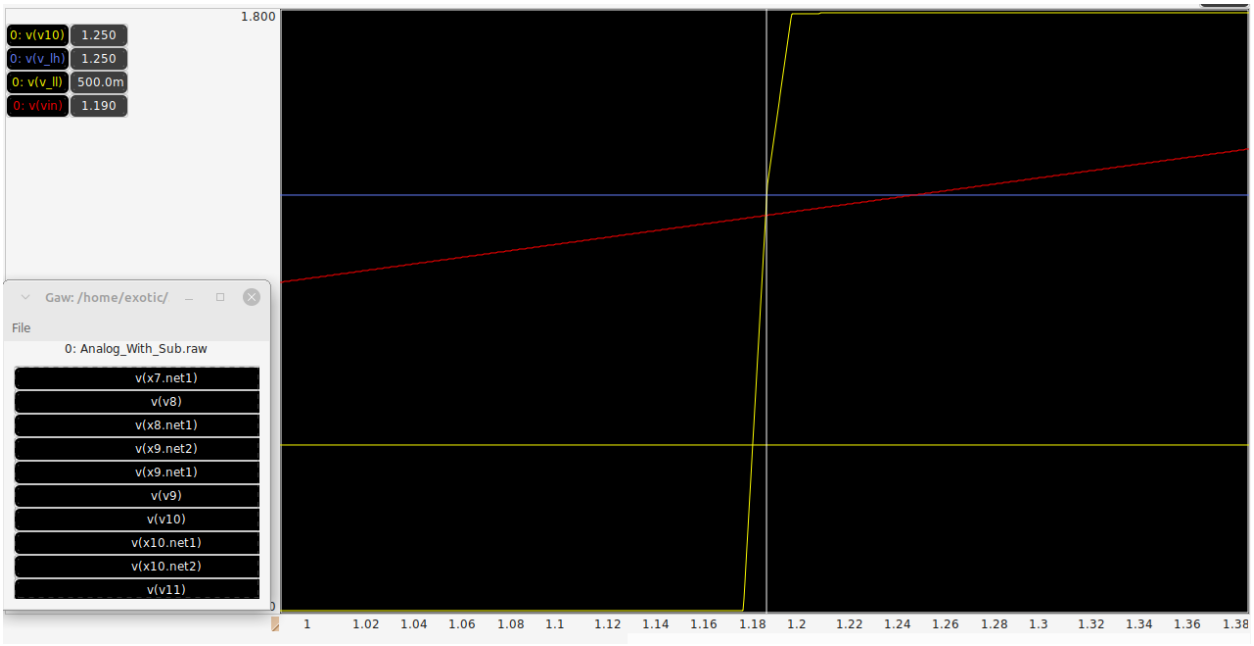
Th. 11: **1.309**

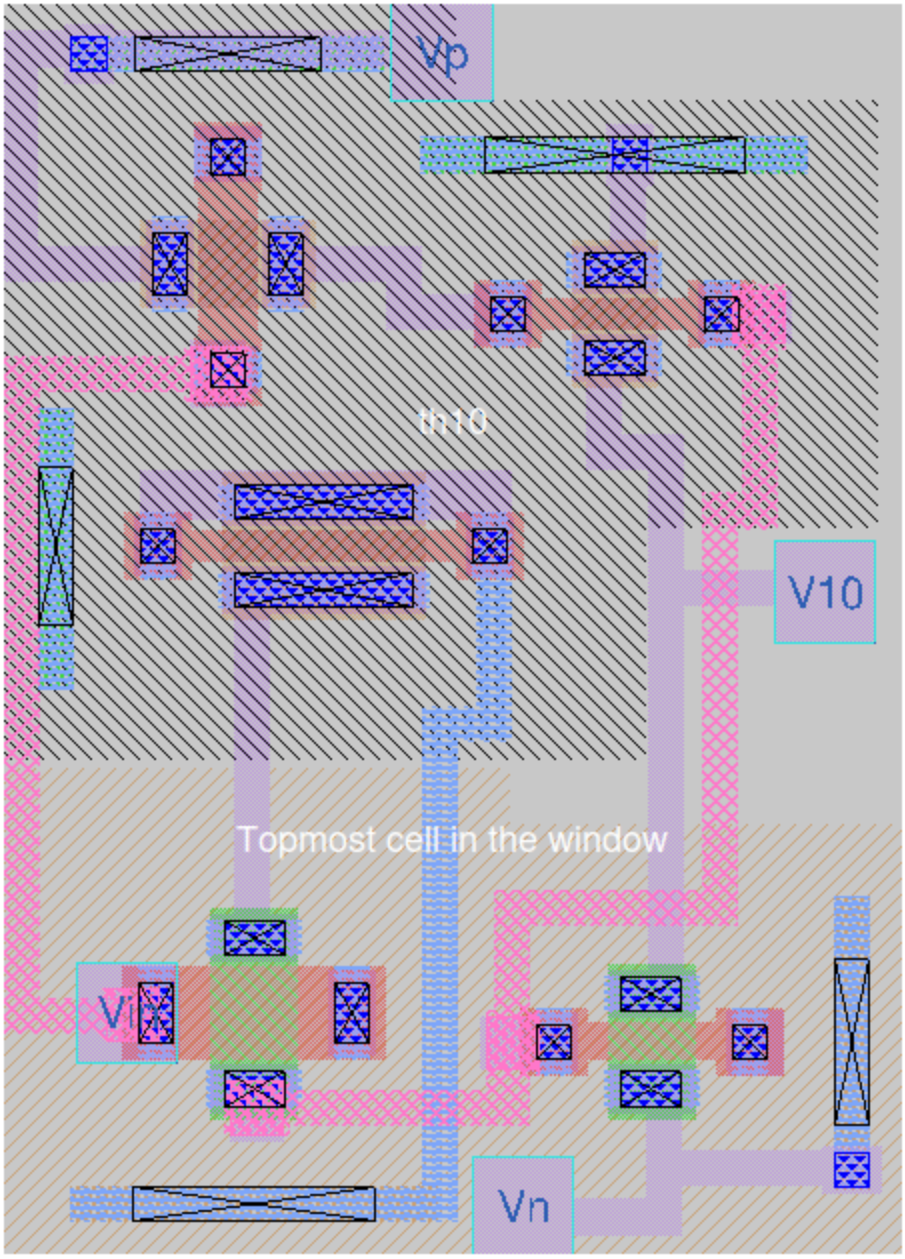

Vp
M2
pfet_01v8
nf=1
1 x 0.43 / 0.15
M3
pfet_01v8
nf=1
1 x 0.42 / 0.5
V11
M1
nfet_01v8
nf=1
1 x 0.36 / 1.8
M4
nfet_01v8
nf=1
1 x 0.5 / 0.15
Vin
M0
pfet_01v8
nf=1
1 x 2 / 0.15
Vn


1.800
0: v(v11) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 1.309
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(x8.net1)
v(x9.net2)
v(x9.net1)
v(v9)
v(v10)
v(x10.net1)
v(x10.net2)
v(v11)
v(x11.net1)
0.85 0.9 0.95 1 1.05 1.1 1.15 1.2 1.25 1.3 1.35 1.4 1.45 1.5 1.55 1.6 1.65

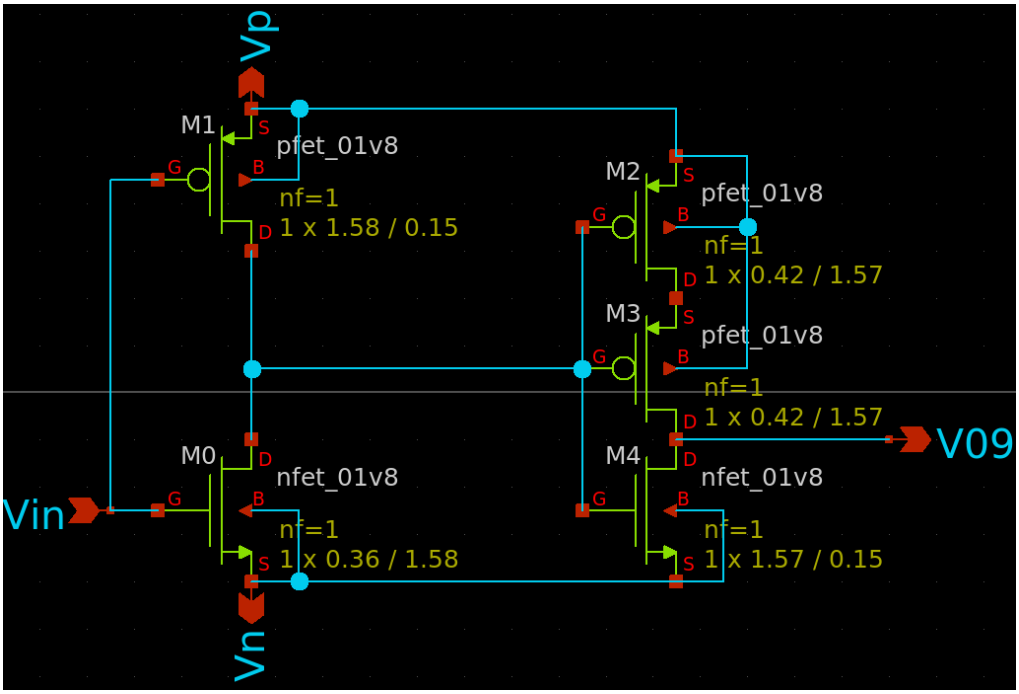
Th. 10: 1.19

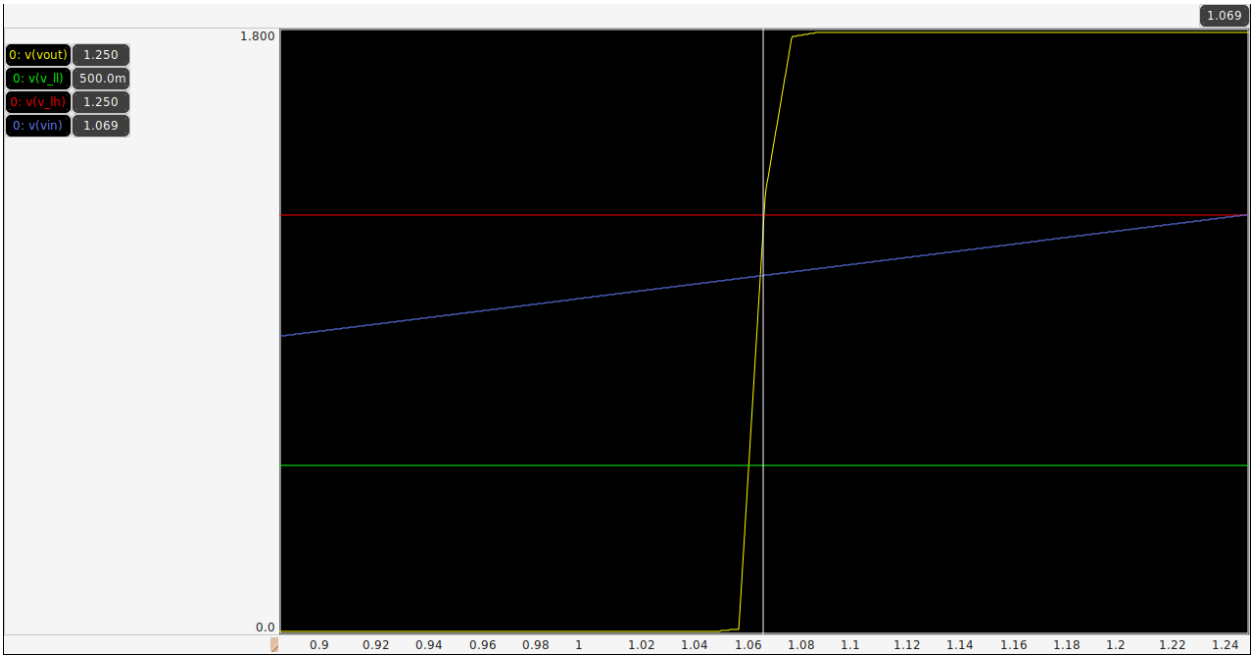

Vp
M2
pfet_01v8
nf=1
1 x 0.42 / 0.28
M3
pfet_01v8
nf=1
1 x 0.42 / 0.15
V10
M1
nfet_01v8
nf=1
1 x 0.40 / 0.46
M4
nfet_01v8
nf=1
1 x 0.42 / 0.18
Vin
M0
pfet_01v8
nf=1
1 x 1 / 0.15
Vn

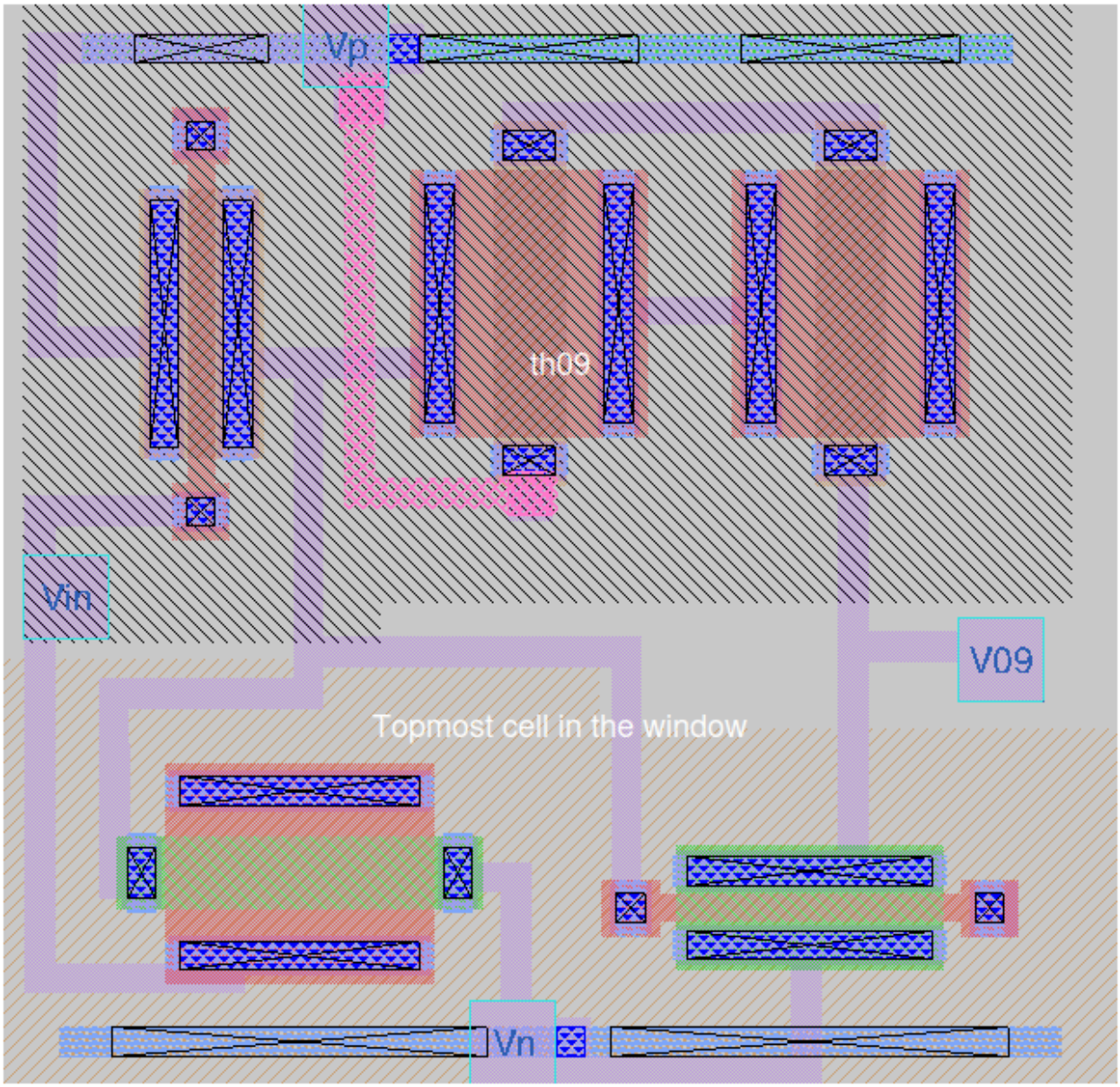

1.800
0: v(v10) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 1.190
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(x7.net1)
v(v8)
v(x8.net1)
v(x9.net2)
v(x9.net1)
v(v9)
v(v10)
v(x10.net1)
v(x10.net2)
v(v11)
1
1.02
1.04
1.06
1.08
1.1
1.12
1.14
1.16
1.18
1.2
1.22
1.24
1.26
1.28
1.3
1.32
1.34
1.36
1.38



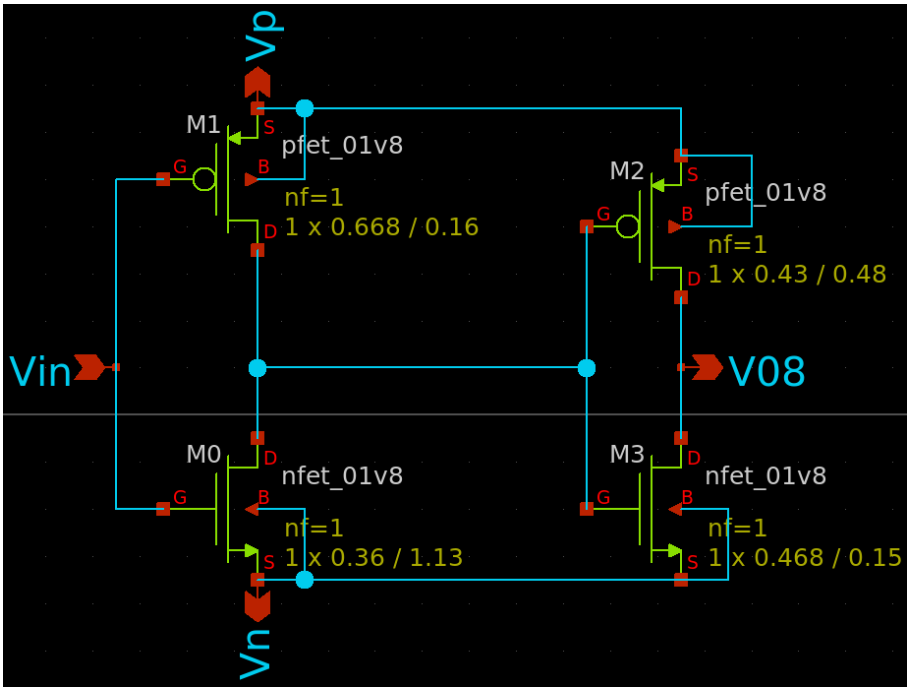
Th. 9: 1.071

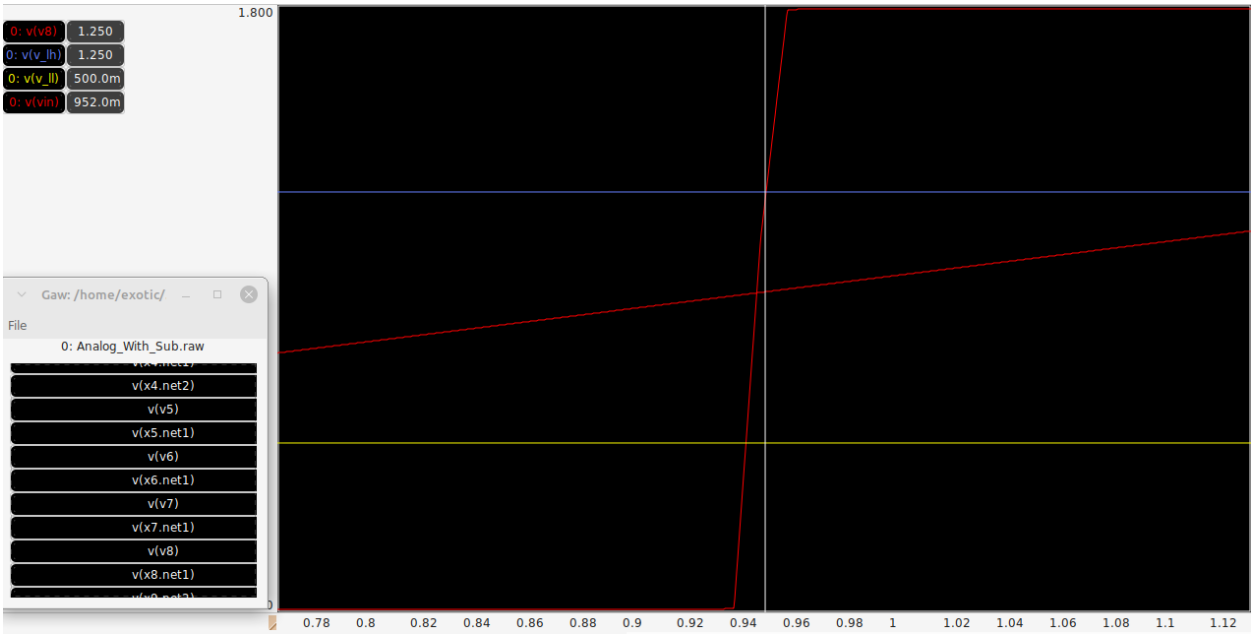

Vp
M1
pfet_01v8
G
B
S
D
nf=1
1 x 1.58 / 0.15
M2
pfet_01v8
nf=1
1 x 0.42 / 1.57
M3
pfet_01v8
nf=1
1 x 0.42 / 1.57
V09
M0
nfet_01v8
nf=1
1 x 0.36 / 1.58
M4
nfet_01v8
nf=1
1 x 1.57 / 0.15
Vin
Vn

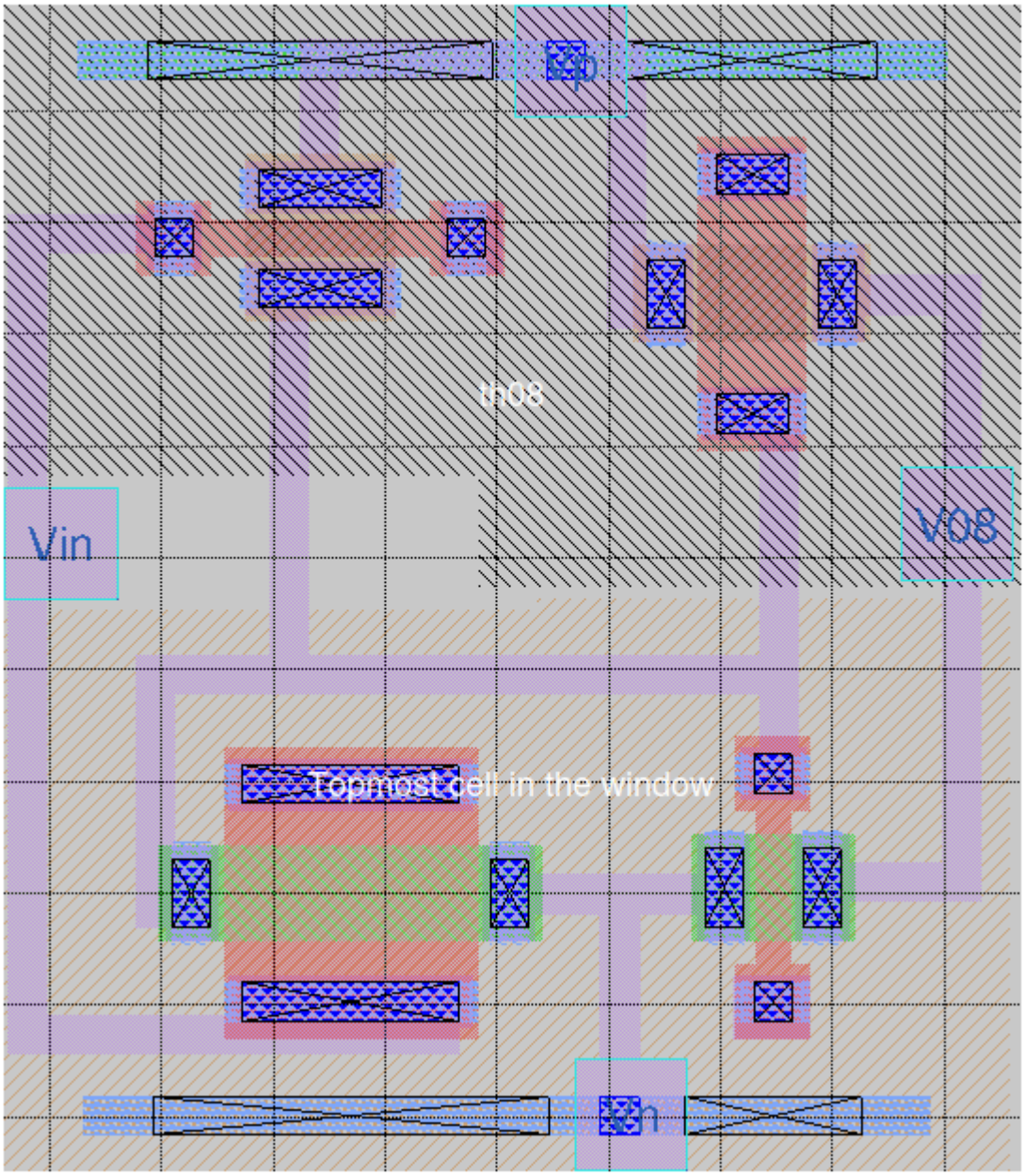

0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.069
1.069
1.800
0.0
0.9
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
1.1
1.12
1.14
1.16
1.18
1.2
1.22
1.24



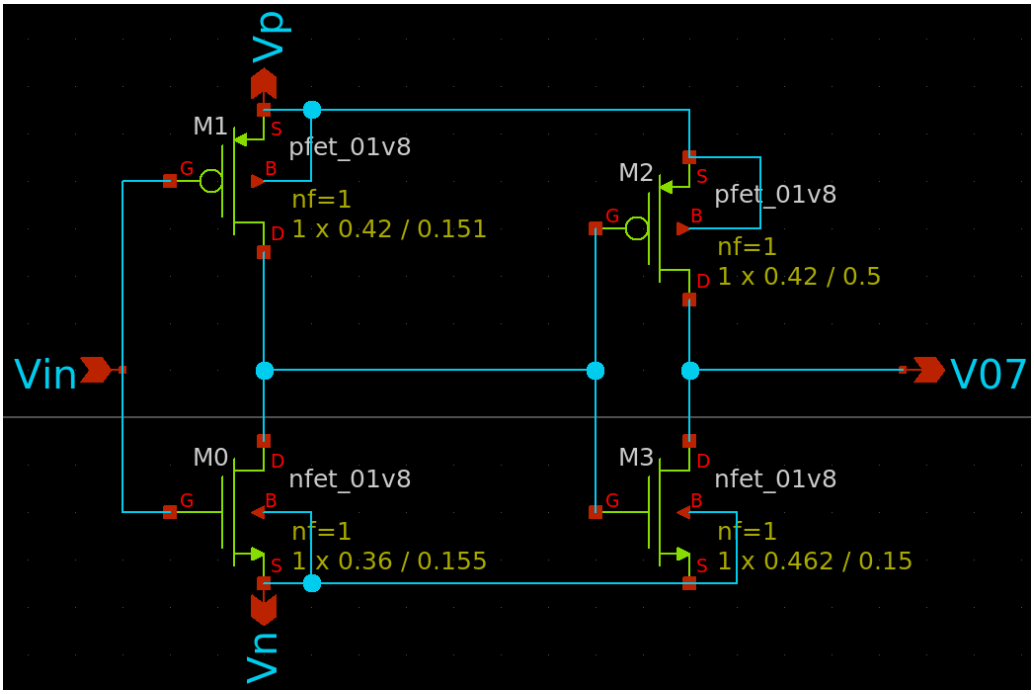
Th. 8: 0.9504

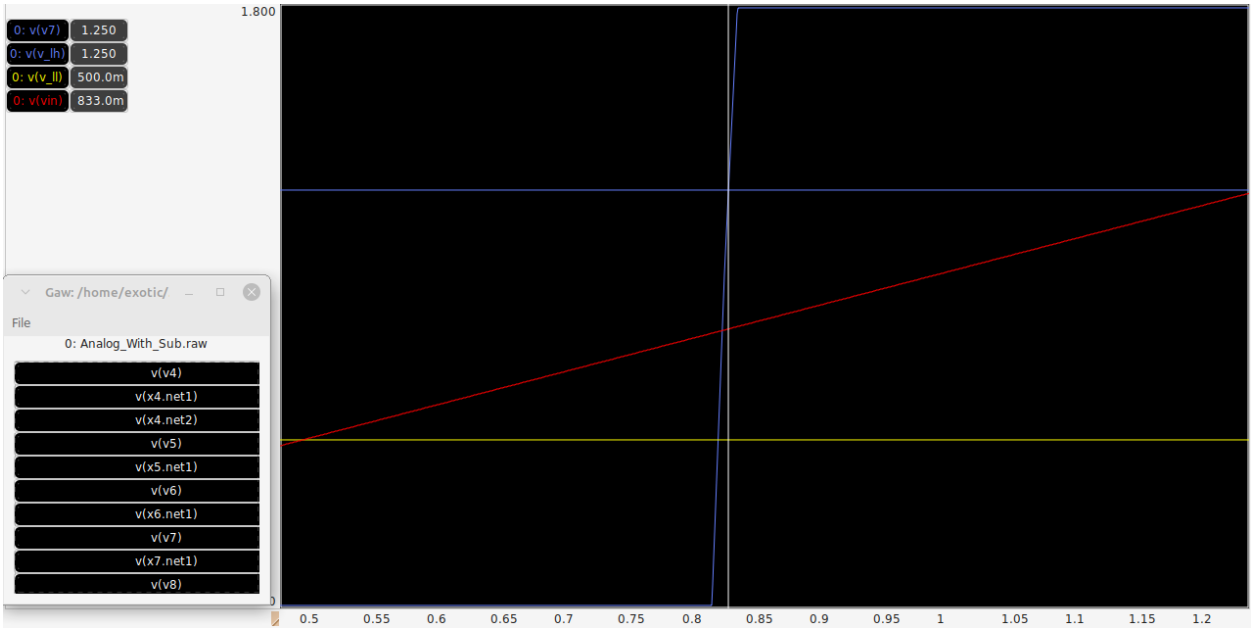

Vp
M1
pfet_01v8
G
B
S
D
nf=1
1 x 0.668 / 0.16
M2
pfet_01v8
nf=1
1 x 0.43 / 0.48
Vin
V08
M0
nfet_01v8
nf=1
1 x 0.36 / 1.13
M3
nfet_01v8
nf=1
1 x 0.468 / 0.15
Vn

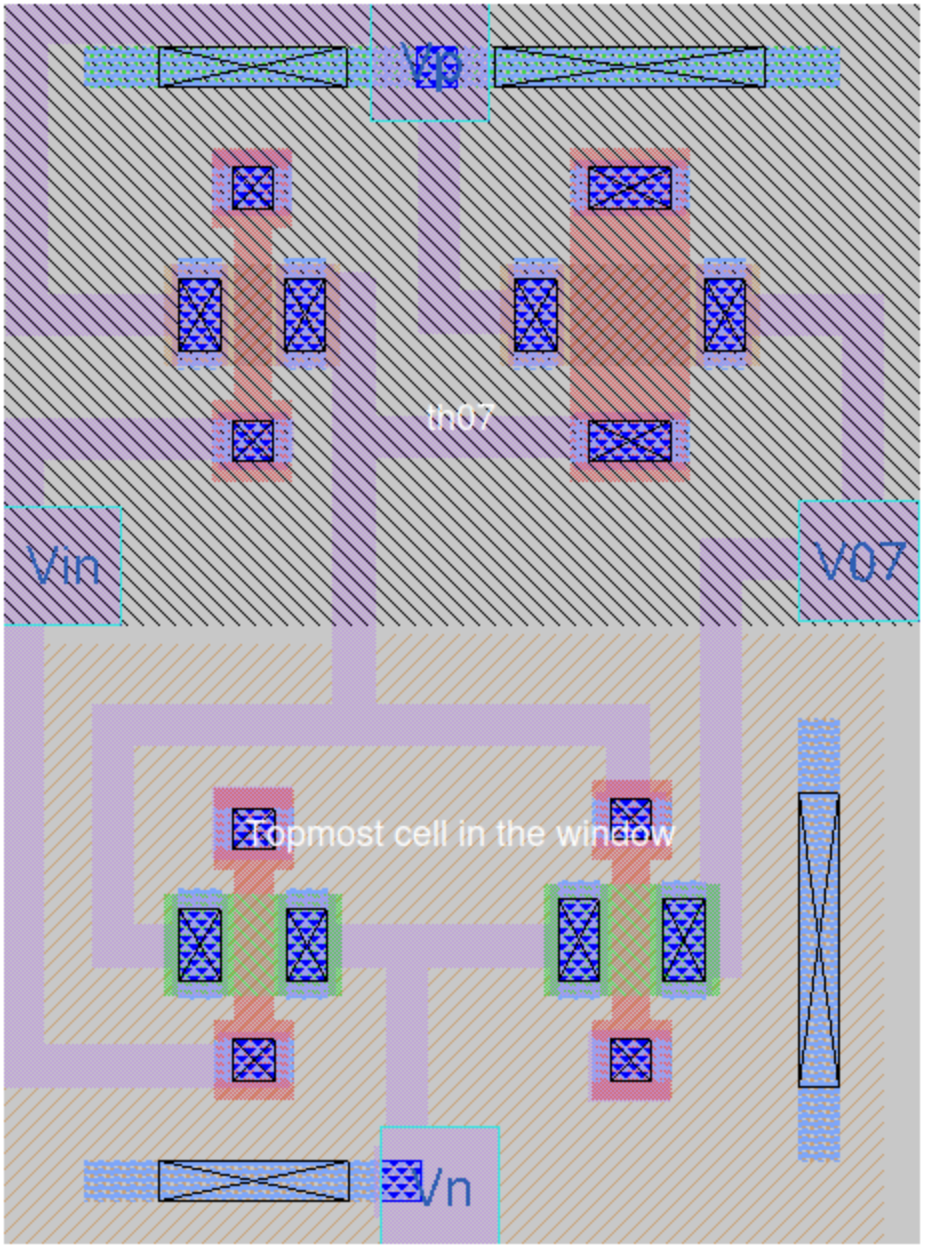

1.800
0: v(v8) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 952.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(x4.net2)
v(v5)
v(x5.net1)
v(v6)
v(x6.net1)
v(v7)
v(x7.net1)
v(v8)
v(x8.net1)
0.78
0.8
0.82
0.84
0.86
0.88
0.9
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
1.1
1.12



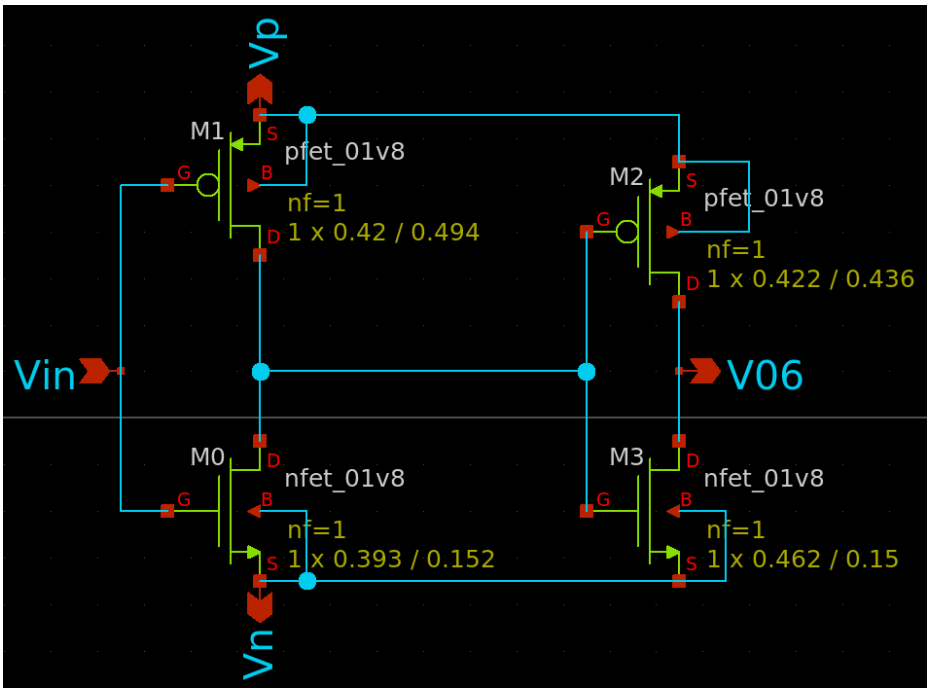
Th. 7: 0.8316


Vp
M1
pfet_01v8
G
B
S
D
nf=1
1 x 0.42 / 0.151
M2
pfet_01v8
nf=1
1 x 0.42 / 0.5
Vin
V07
M0
nfet_01v8
nf=1
1 x 0.36 / 0.155
M3
nfet_01v8
nf=1
1 x 0.462 / 0.15
Vn

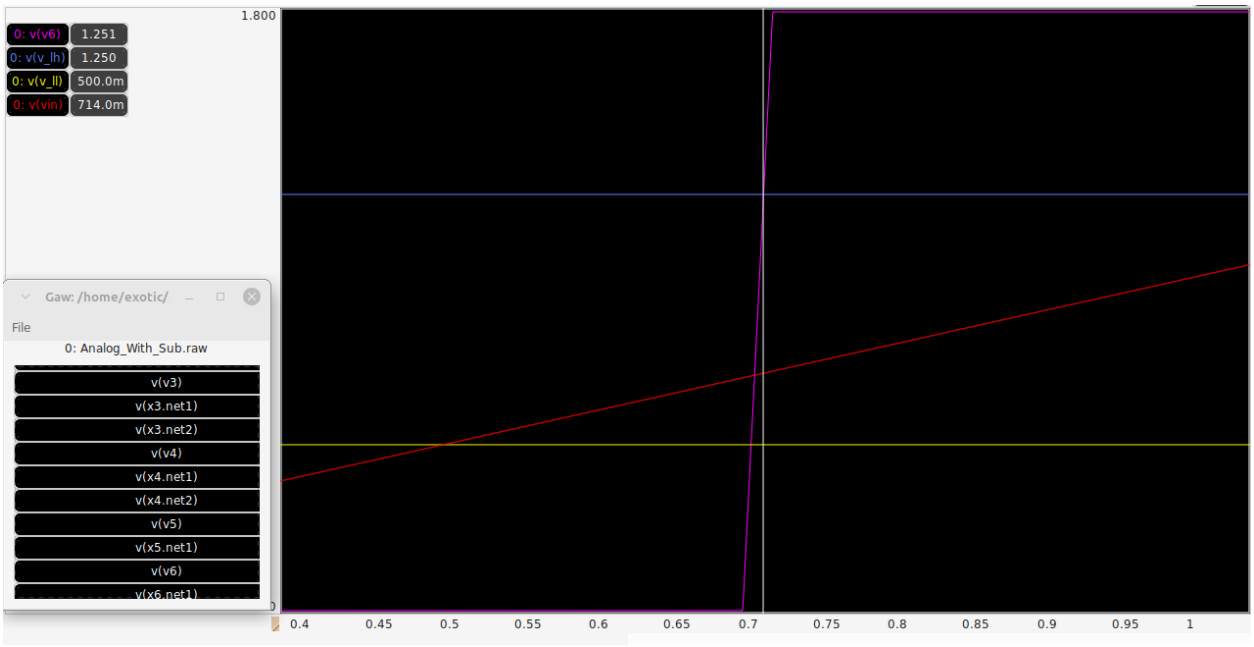

1.800
0: v(v7) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 833.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(v4)
v(x4.net1)
v(x4.net2)
v(v5)
v(x5.net1)
v(v6)
v(x6.net1)
v(v7)
v(x7.net1)
v(v8)
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
1.05
1.1
1.15
1.2



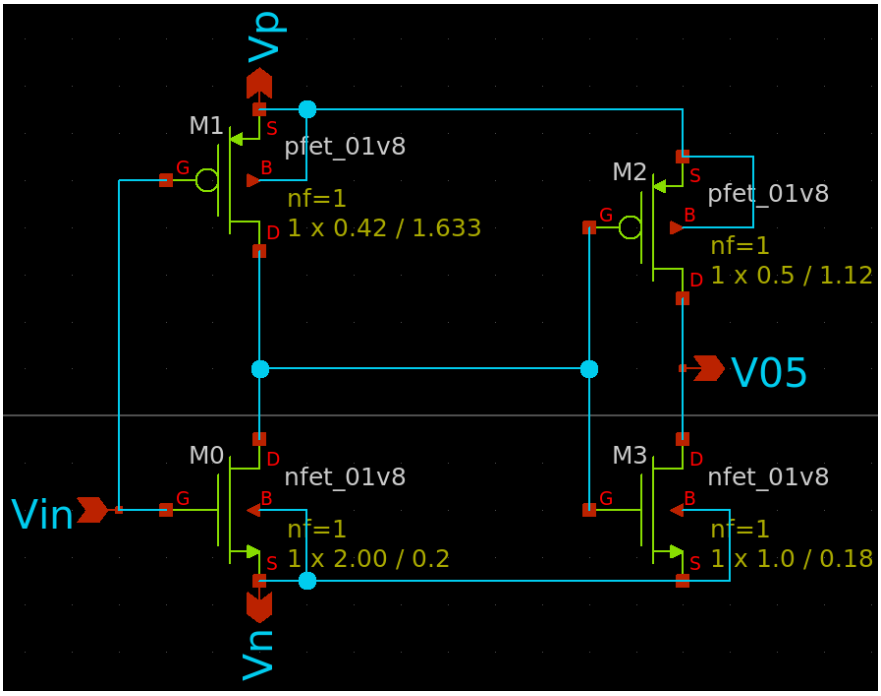
Th. 6: 0.7140

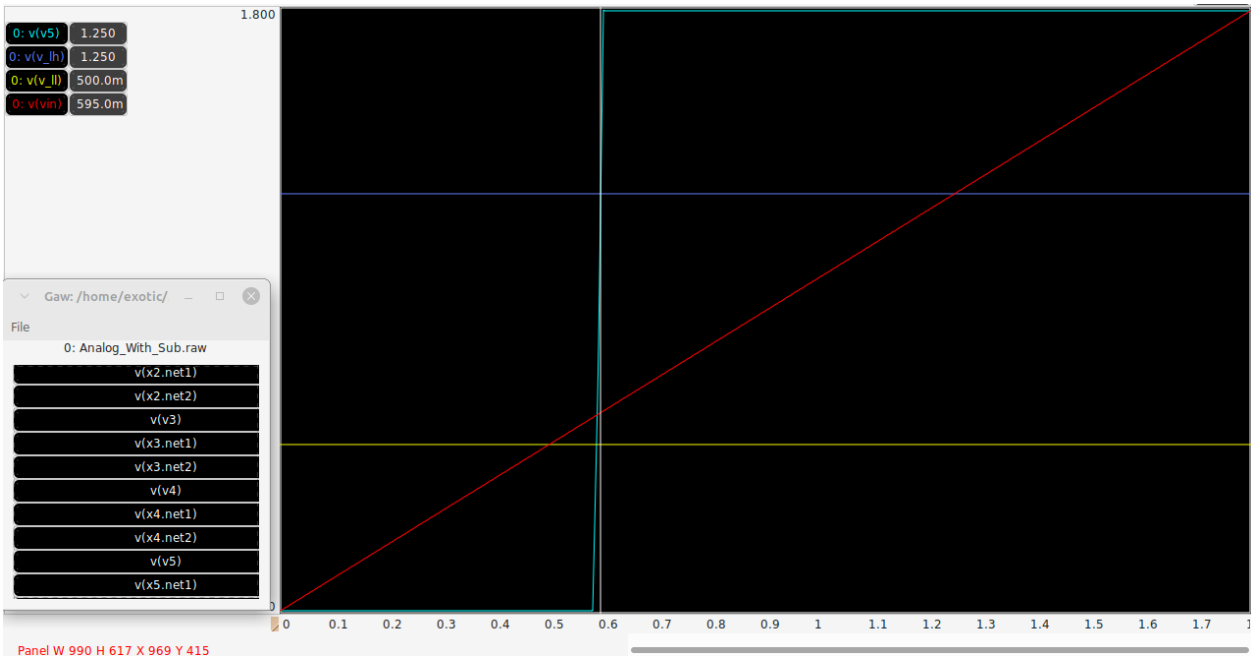

Vp
M1
S
pfet_01v8
G
B
nf=1
D
1 x 0.42 / 0.494
M2
S
pfet_01v8
G
B
nf=1
D
1 x 0.422 / 0.436
Vin
V06
M0
D
nfet_01v8
G
B
nf=1
S
1 x 0.393 / 0.152
M3
D
nfet_01v8
G
B
nf=1
S
1 x 0.462 / 0.15
Vn

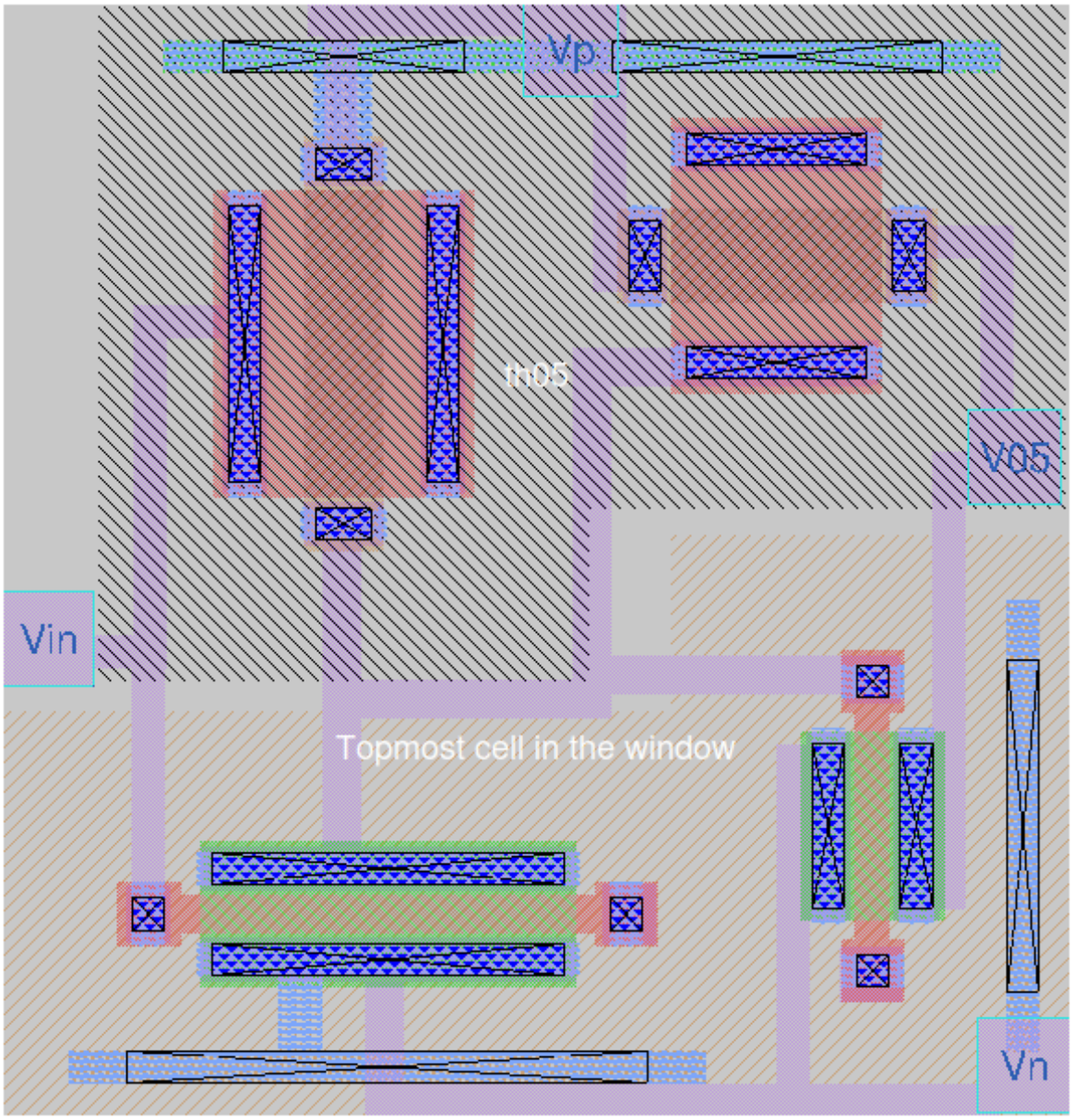
S


1.800
0: v(v6) 1.251
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 714.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(v3)
v(x3.net1)
v(x3.net2)
v(v4)
v(x4.net1)
v(x4.net2)
v(v5)
v(x5.net1)
v(v6)
v(x6.net1)
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1

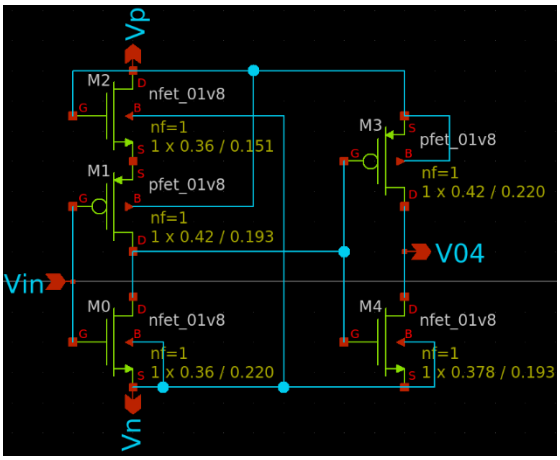
Th. 5: 0.595

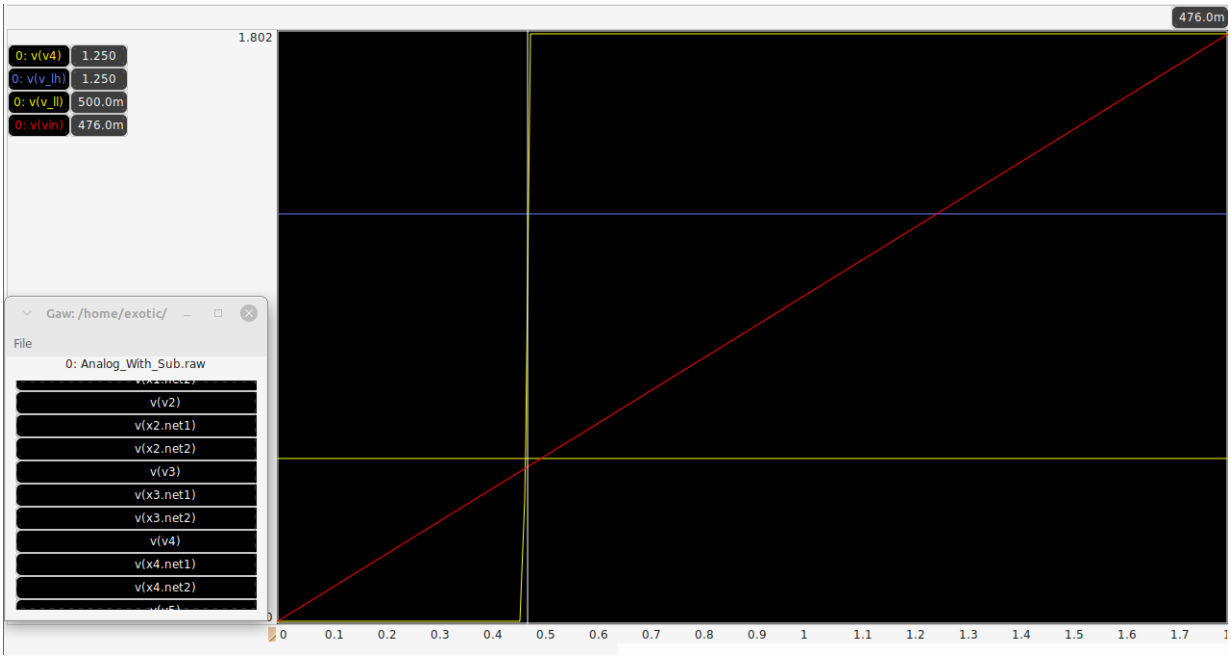

Vp
M1
pfet_01v8
G
B
S
D
nf=1
1 x 0.42 / 1.633
M2
pfet_01v8
nf=1
1 x 0.5 / 1.12
V05
M0
nfet_01v8
nf=1
1 x 2.00 / 0.2
Vin
M3
nfet_01v8
nf=1
1 x 1.0 / 0.18
Vn

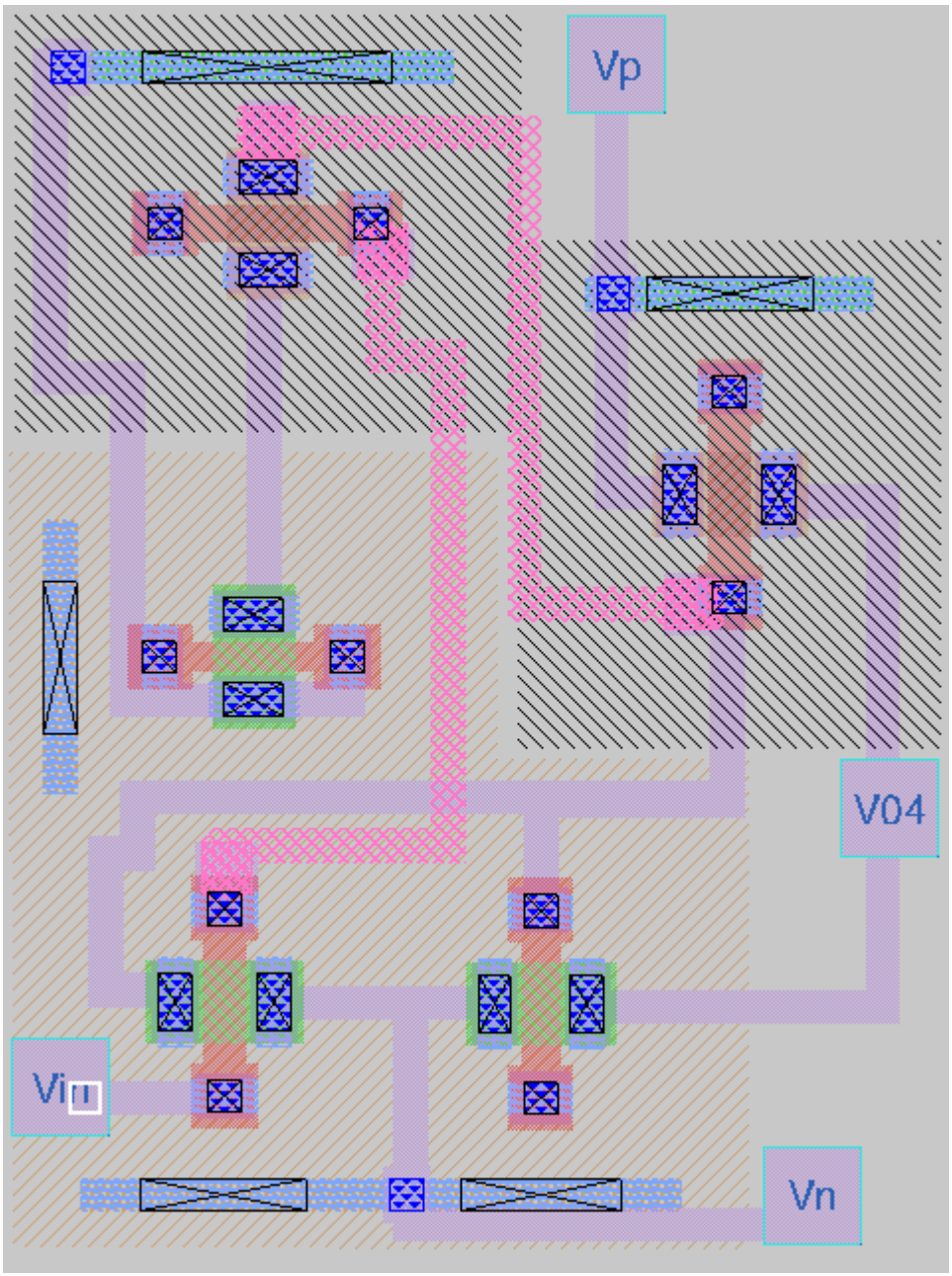

1.800
0: v(v5) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 595.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(x2.net1)
v(x2.net2)
v(v3)
v(x3.net1)
v(x3.net2)
v(v4)
v(x4.net1)
v(x4.net2)
v(v5)
v(x5.net1)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
Panel W 990 H 617 X 969 Y 415



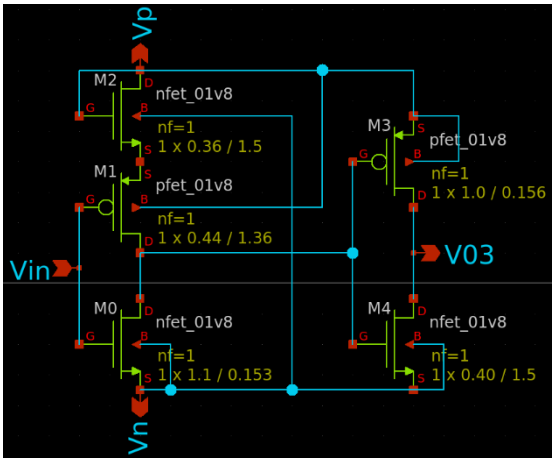
Th. 4: 0.4760

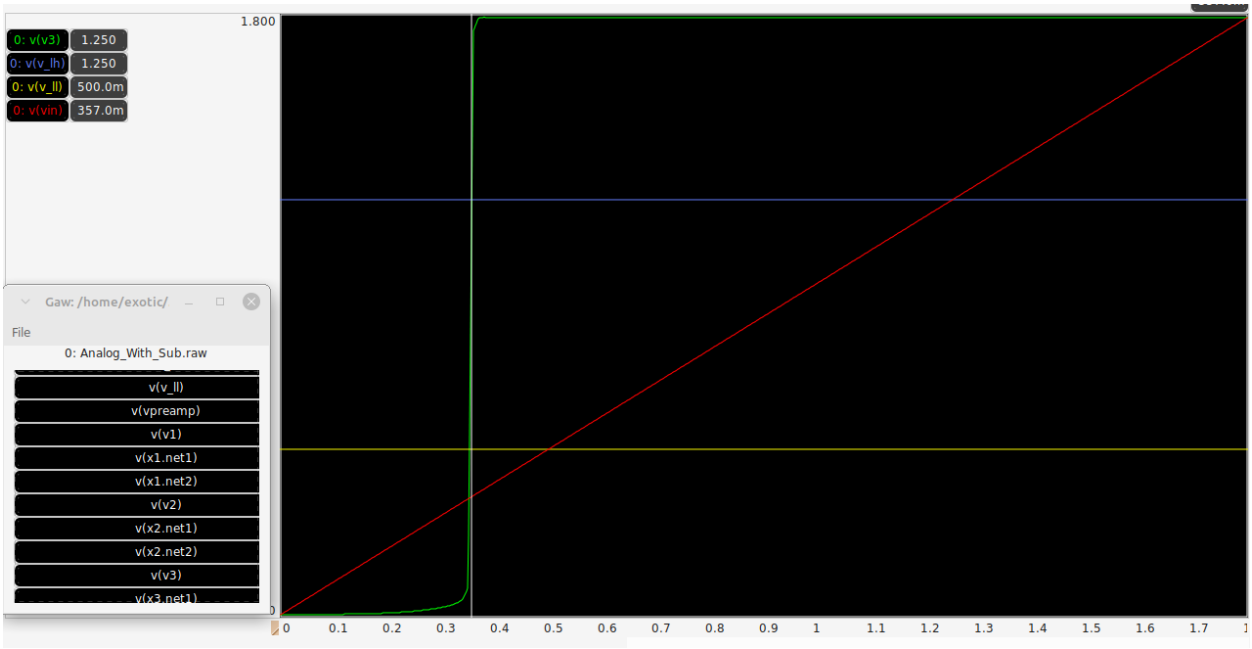

Vp
M2
nfet_01v8
nf=1
1 x 0.36 / 0.151
M1
pfet_01v8
nf=1
1 x 0.42 / 0.193
M3
pfet_01v8
nf=1
1 x 0.42 / 0.220
V04
Vin
M0
nfet_01v8
nf=1
1 x 0.36 / 0.220
M4
nfet_01v8
nf=1
1 x 0.378 / 0.193
Vn

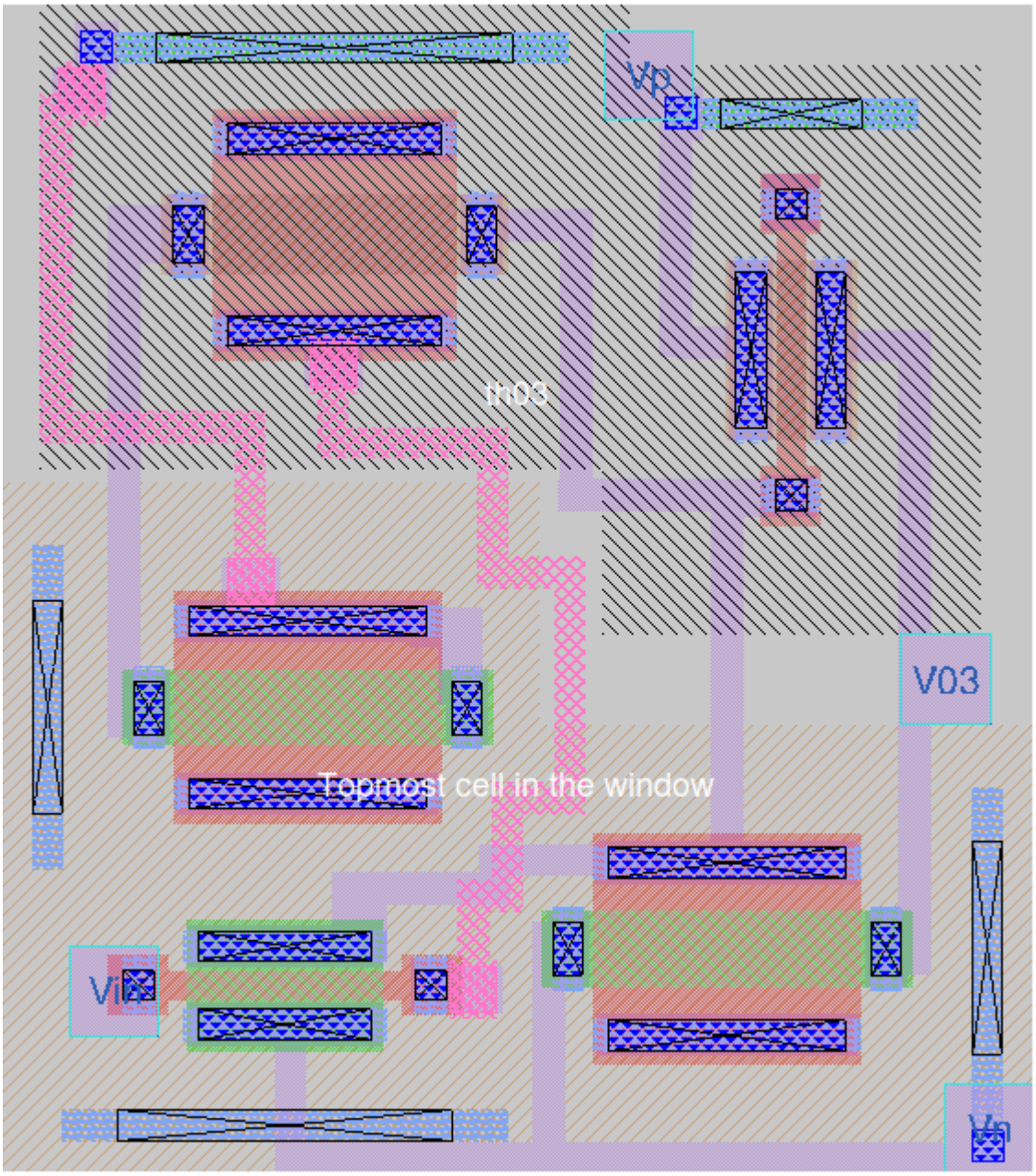

476.0m
1.802
0: v(v4) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 476.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(v2)
v(x2.net1)
v(x2.net2)
v(v3)
v(x3.net1)
v(x3.net2)
v(v4)
v(x4.net1)
v(x4.net2)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7



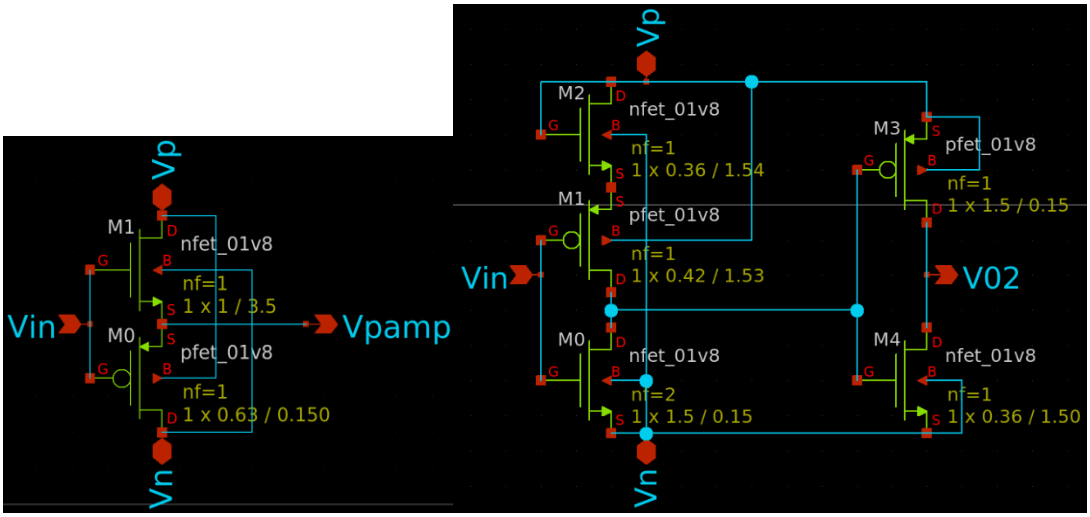
Th. 3: 0.3570

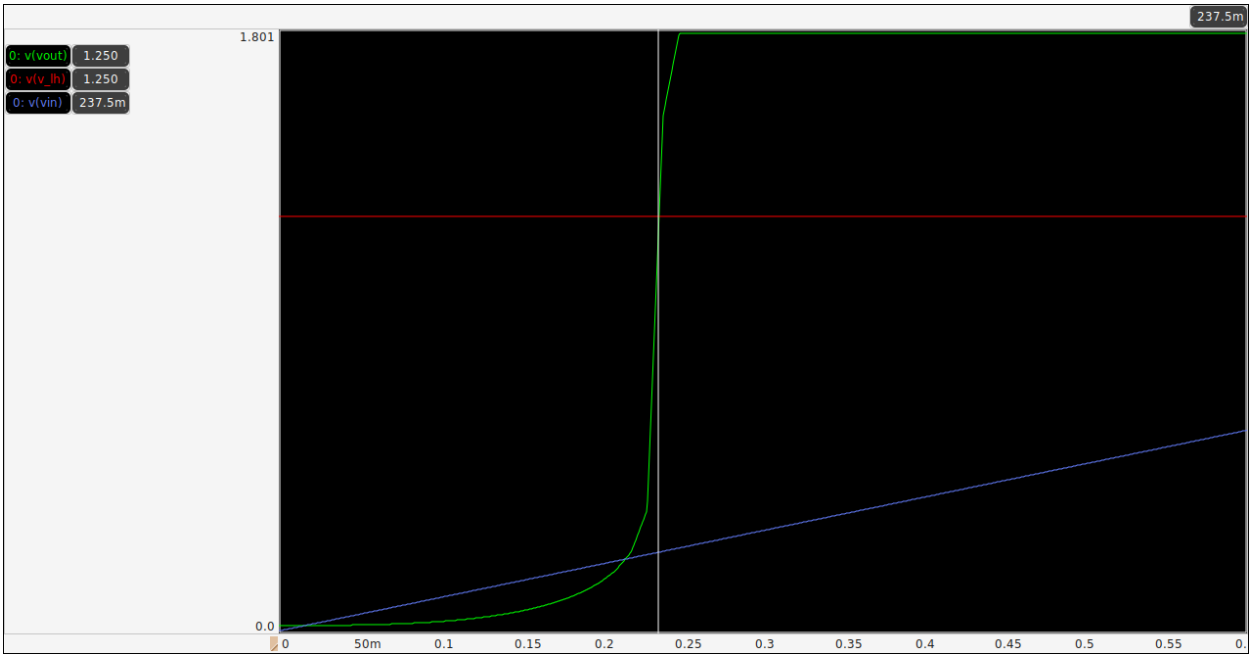

Vp
M2
nfet_01v8
nf=1
1 x 0.36 / 1.5
M3
pfet_01v8
nf=1
1 x 1.0 / 0.156
M1
pfet_01v8
nf=1
1 x 0.44 / 1.36
Vin
V03
M0
nfet_01v8
nf=1
1 x 1.1 / 0.153
M4
nfet_01v8
nf=1
1 x 0.40 / 1.5
Vn

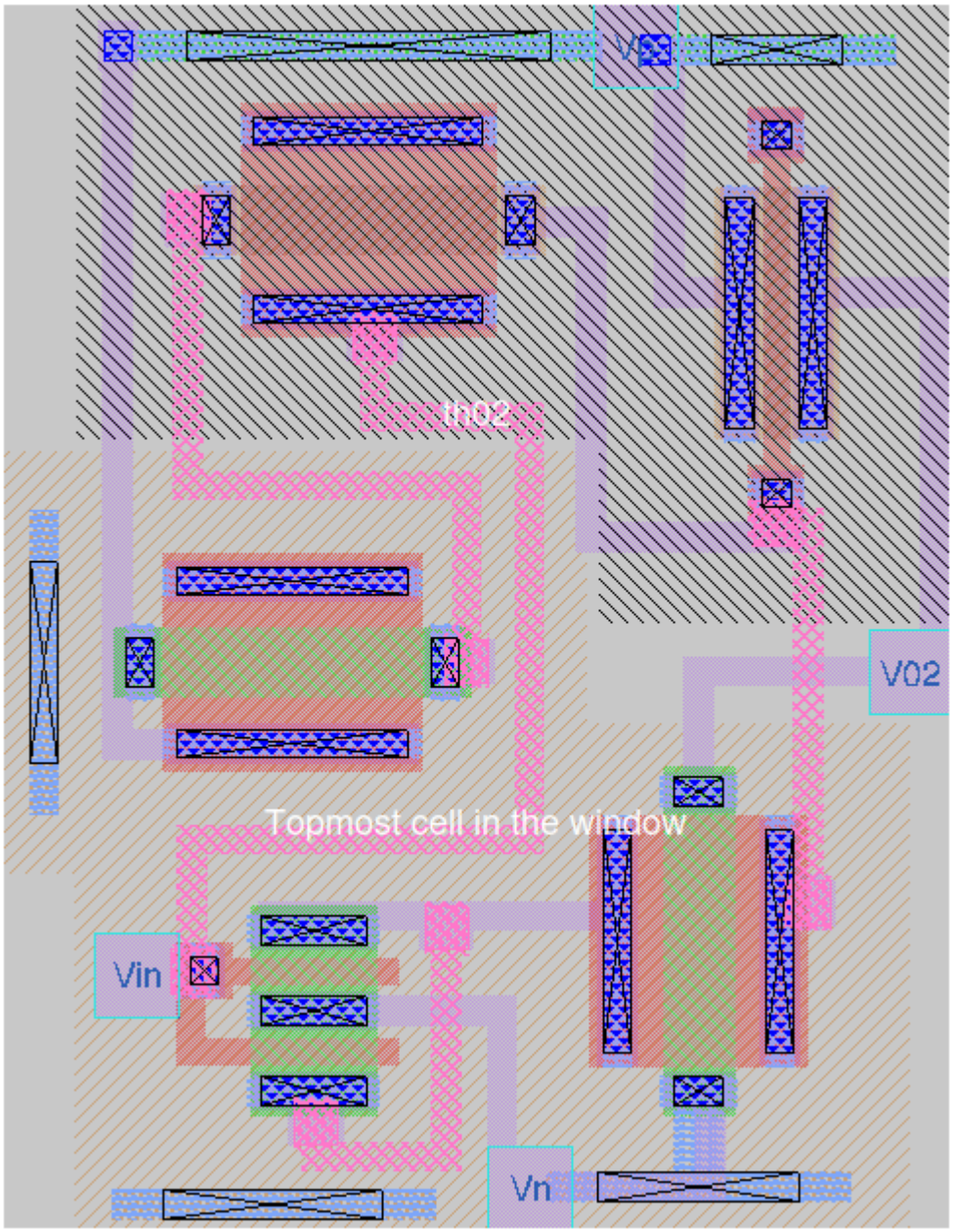

1.800
0: v(v3) 1.250
0: v(v_lh) 1.250
0: v(v_ll) 500.0m
0: v(vin) 357.0m
Gaw: /home/exotic/
File
0: Analog_With_Sub.raw
v(v_ll)
v(vpreamp)
v(v1)
v(x1.net1)
v(x1.net2)
v(v2)
v(x2.net1)
v(x2.net2)
v(v3)
v(x3.net1)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7



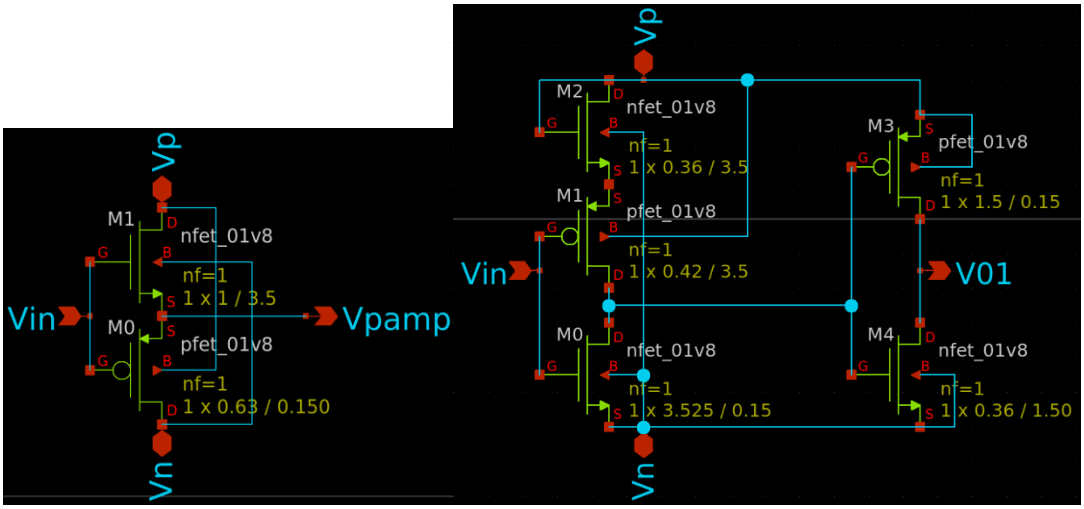
Th. 2: 0.2380


Vp
M1
nfet_01v8
nf=1
1 x 1 / 3.5
Vin
M0
Vpamp
pfet_01v8
nf=1
1 x 0.63 / 0.150
Vn
Vp
M2
nfet_01v8
nf=1
1 x 0.36 / 1.54
M3
pfet_01v8
nf=1
1 x 1.5 / 0.15
M1
pfet_01v8
nf=1
1 x 0.42 / 1.53
Vin
V02
M0
nfet_01v8
nf=2
1 x 1.5 / 0.15
M4
nfet_01v8
nf=1
1 x 0.36 / 1.50
Vn

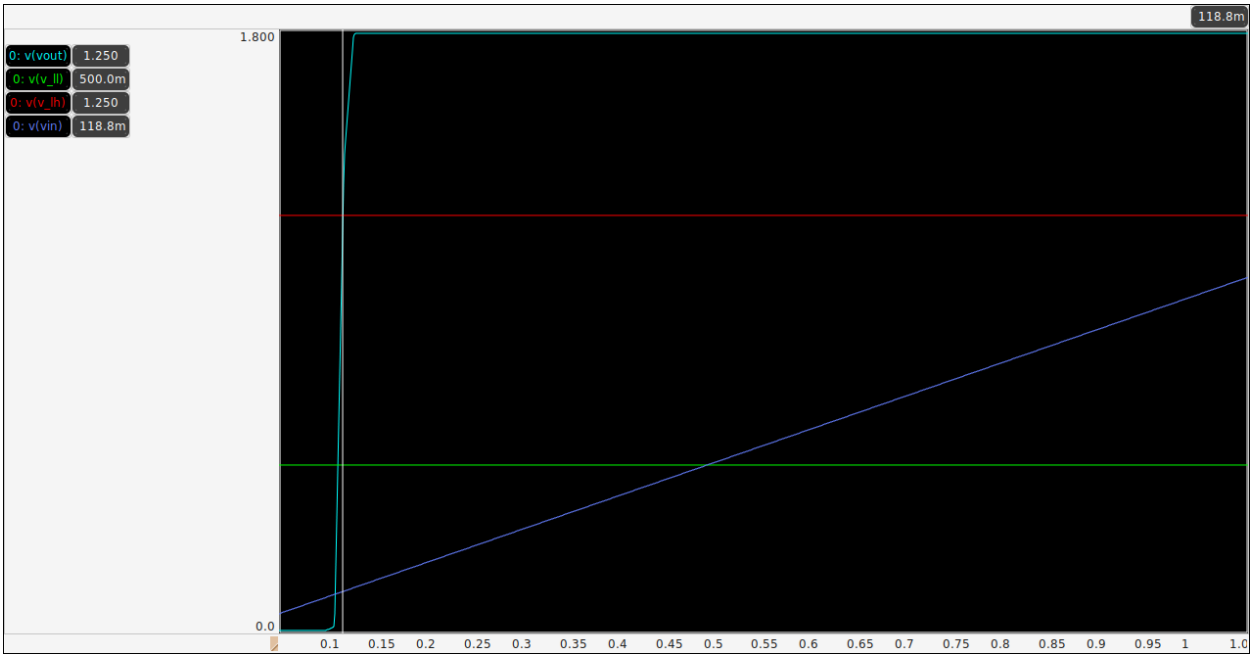

237.5m
0: v(vout) 1.250
0: v(v_lh) 1.250
0: v(vin) 237.5m
1.801
0.0
0
50m
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

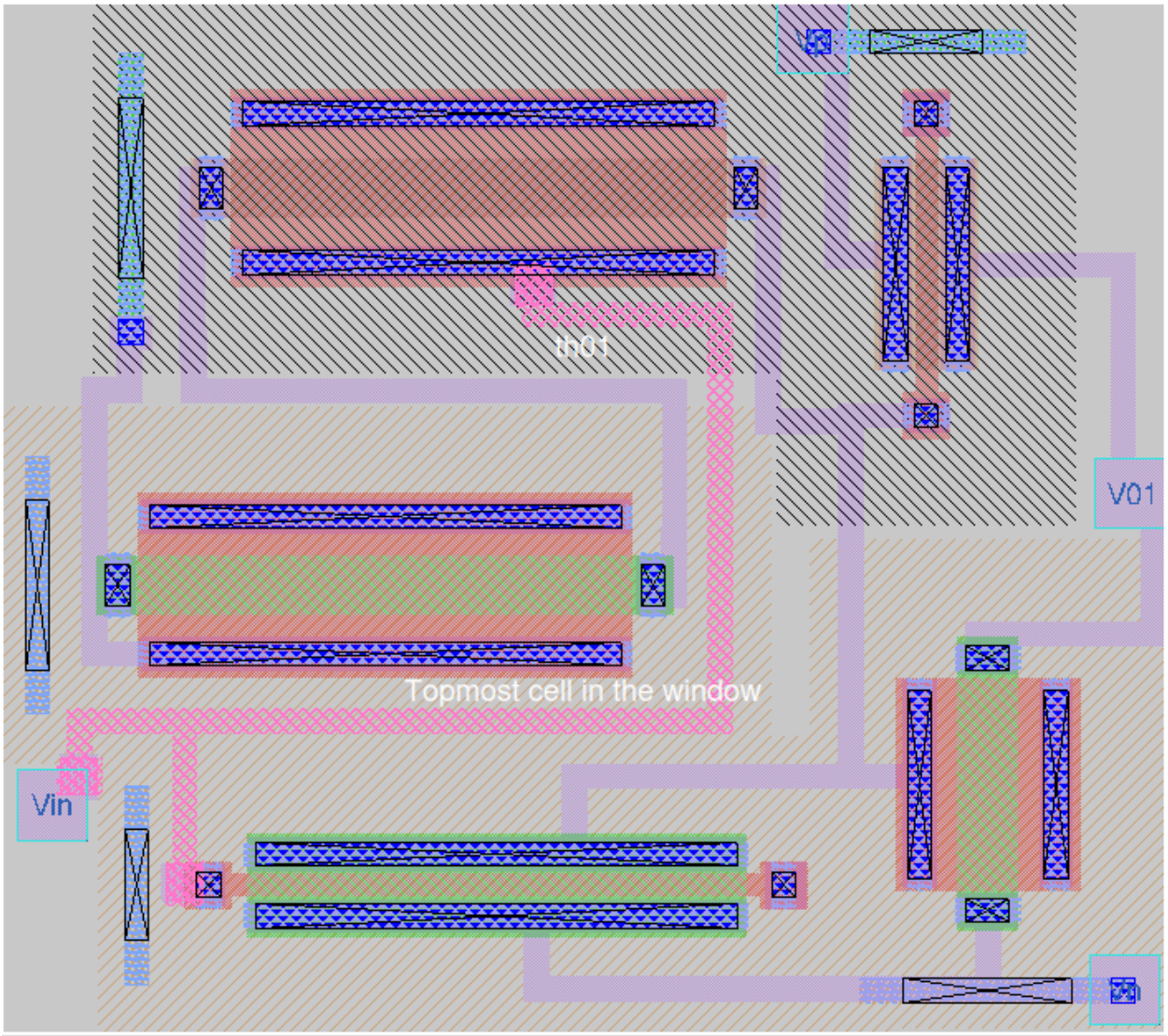


Th. 1: 0.119s

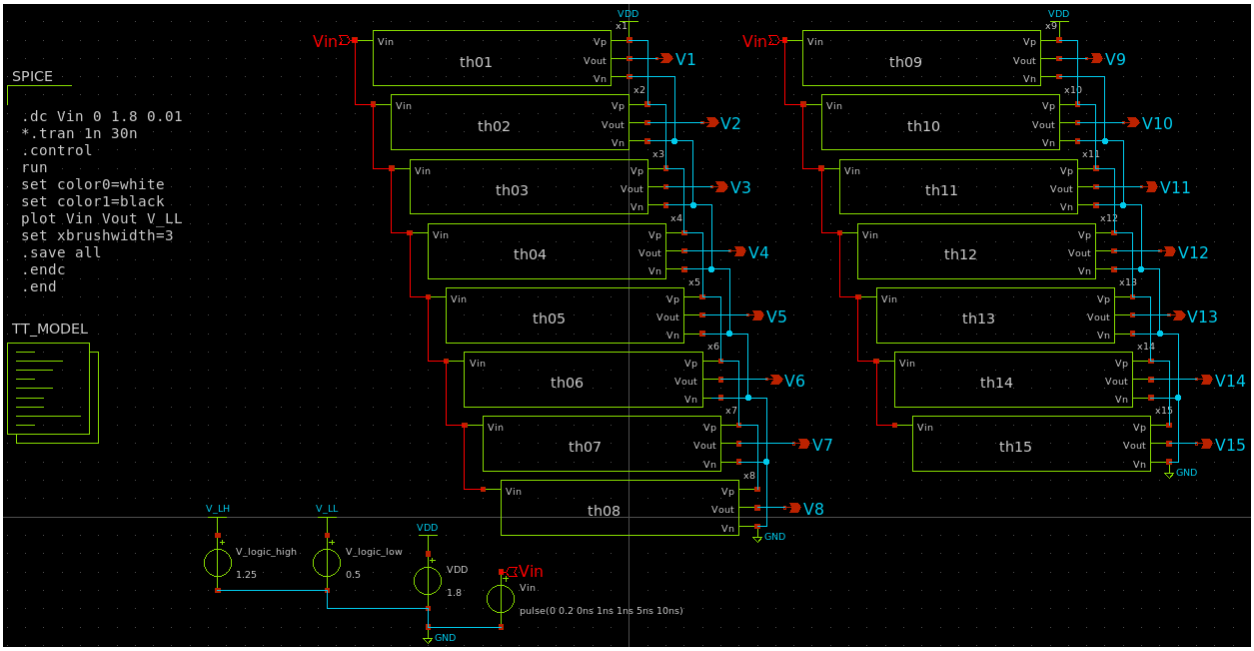


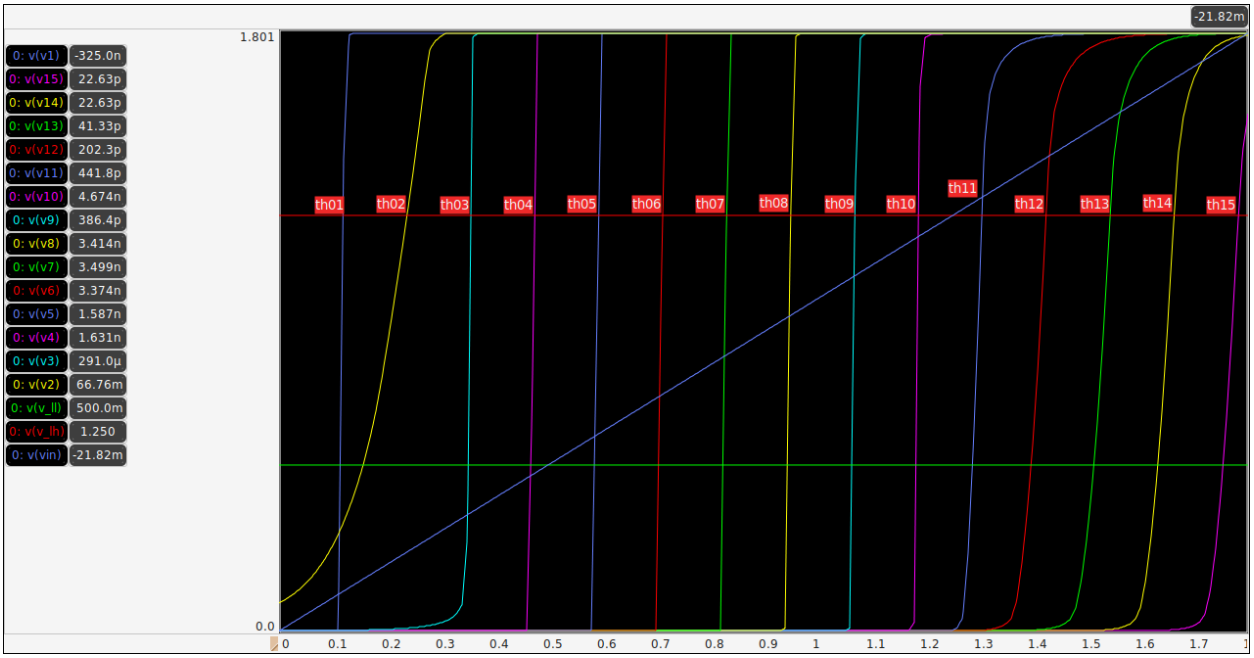
With combination of all thresholds the length and witdh need to be further adjusted.





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.githubusercontent.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'b000000000000000: binary = 4'b0000;   // ~0.0000
      16'b000000000000001: binary = 4'b0001;   // ~0.1188
      16'b000000000000011: binary = 4'b0010;   // ~0.2376
      16'b000000000000111: binary = 4'b0011;   // ~0.3564
      16'b000000000001111: binary = 4'b0100;   // ~0.4752
      16'b000000000011111: binary = 4'b0101;   // ~0.5940
      16'b000000000111111: binary = 4'b0110;   // ~0.7128
      16'b000000001111111: binary = 4'b0111;   // ~0.8316
      16'b000000011111111: binary = 4'b1000;   // ~0.9504
      16'b000000111111111: binary = 4'b1001;   // ~1.0692
      16'b000001111111111: 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=PLfGJEQLQIDBN0VsXQ68_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/BIqLk23hE90?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

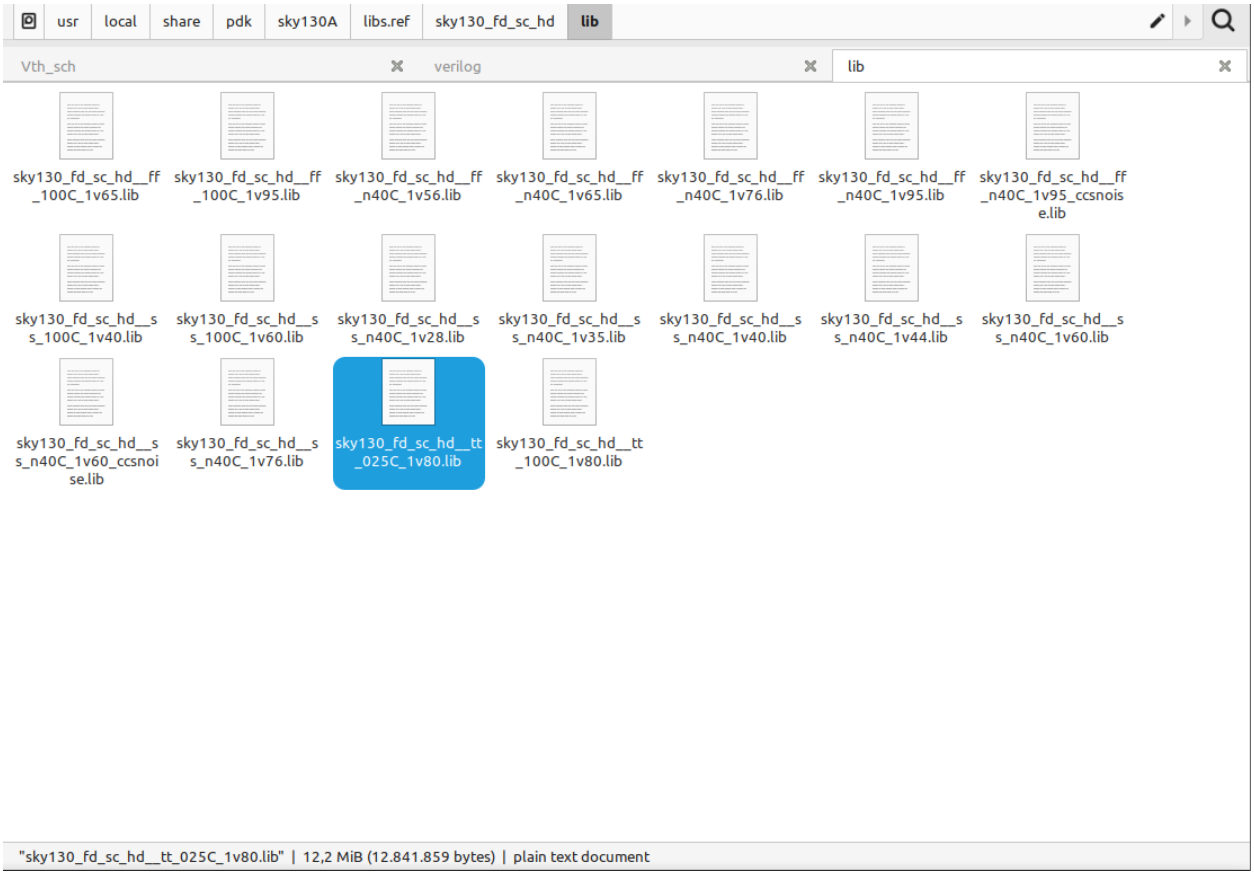
https://github.com/Imellal/RTL-workshop-Sky130-PDK

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

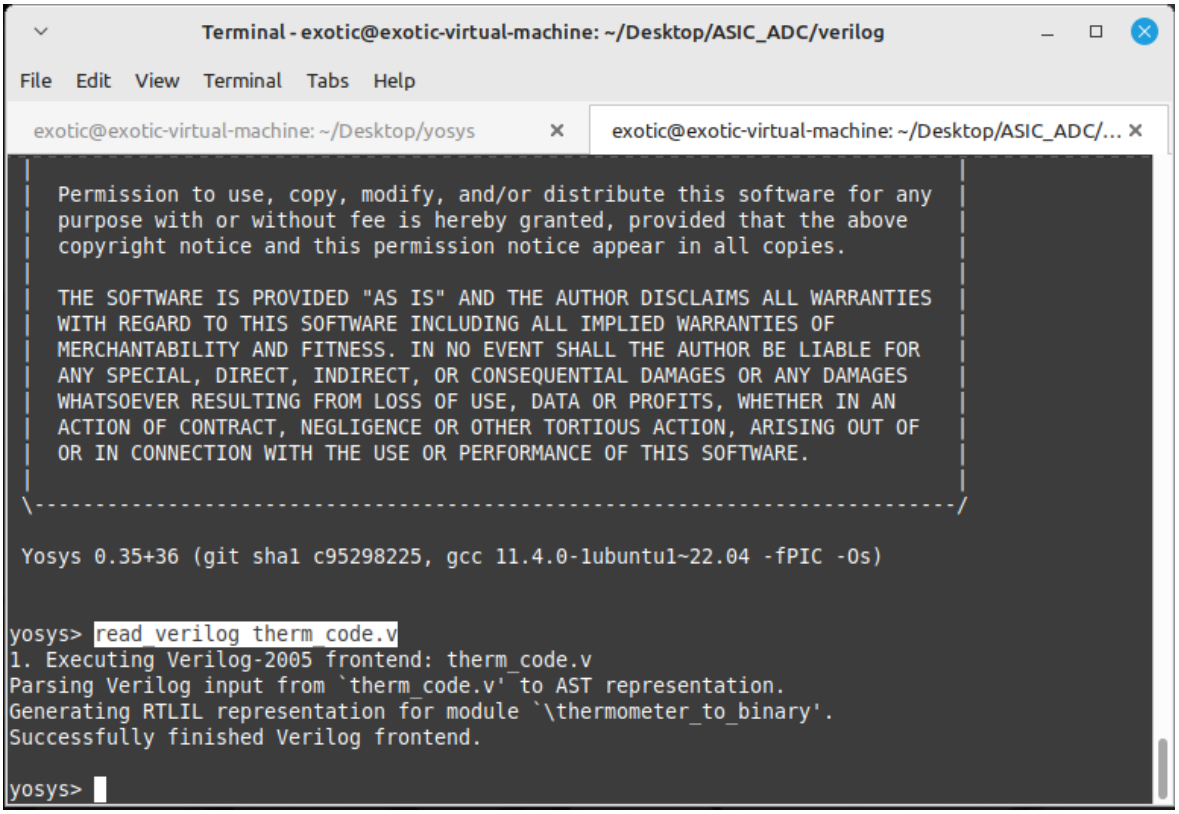
>> yosys // open yosys console in terminal

>> 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)



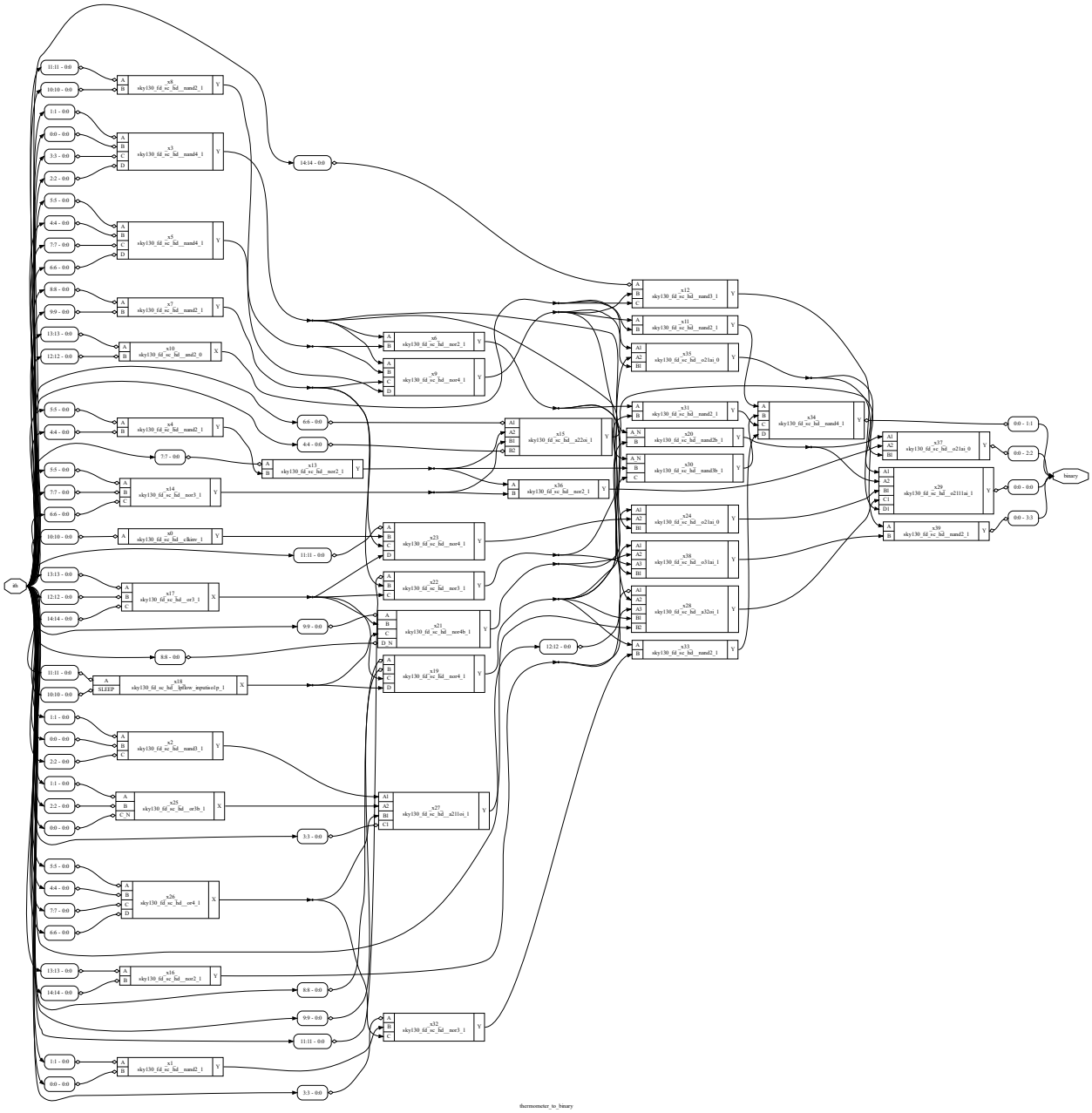
>> **read_verilog therm_code.v** // read the verilog file



>> **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 // vew 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
```



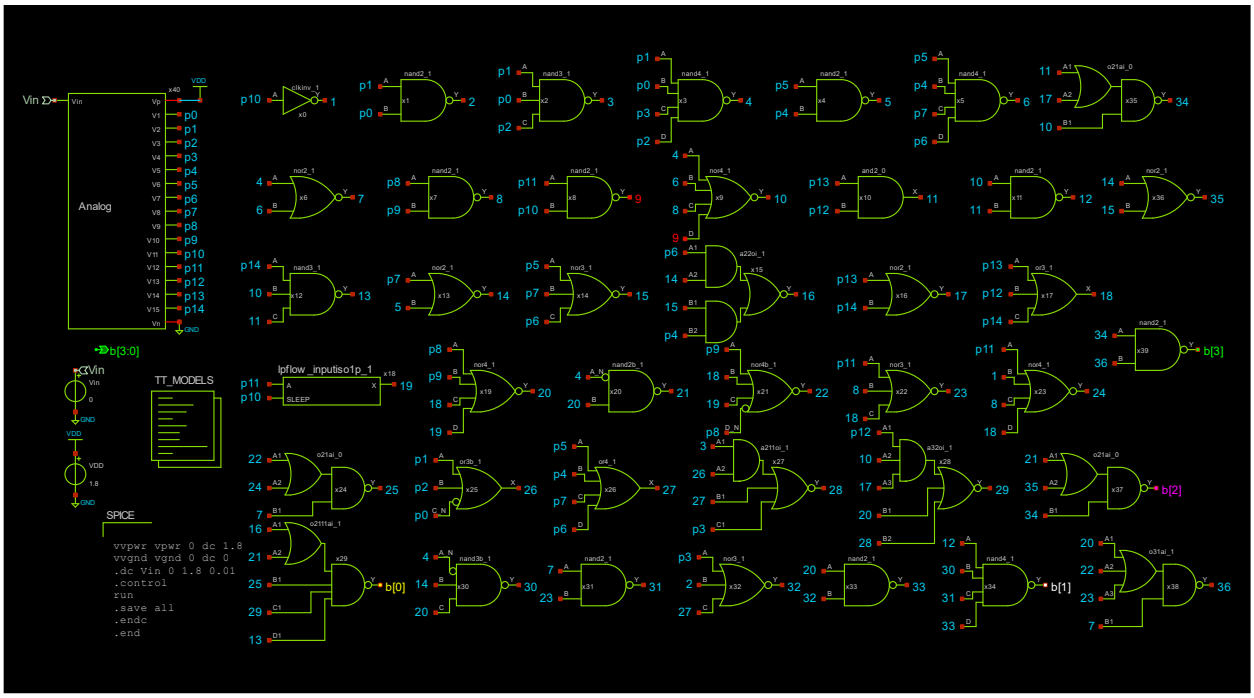
Netlist to schematic conversion:
https://github.com/aidangoettsch/asg

install NetlistViewer:
https://github.com/f18m/netlist-viewer/blob/master/NetlistViewer/build/linux/README.md

doteditor:
https://sourceforge.net/projects/netlistviewer/

Iverilog with gtkwave:
https://youtu.be/Ryt1ms8Tido

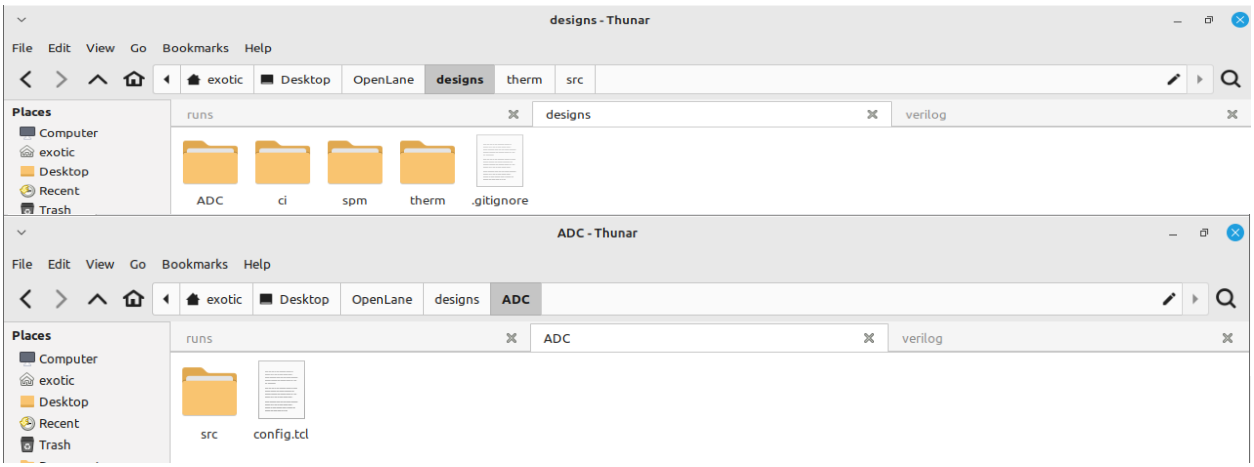
Digital flow with qflow:
http://opencircuitdesign.com/qflow/tutorial_nogui.html
Layout editor
https://www.youtube.com/watch?v=K-_6-GMwmc4
https://www.layouteditor.org/layout/basics/installation



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.ctl is preferred the following command is to be entered:
./flow.tcl -design ADC -init_design_config -add_to_designs -config_file config.tcl



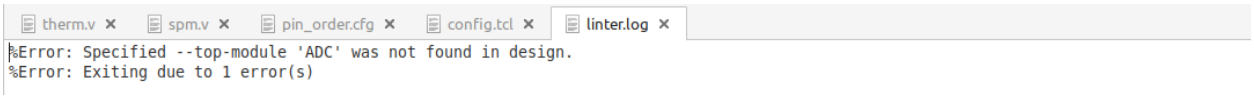
Now the config.ctl has to be configured accordingly (resolving the errors) and the verilog file need to be copied to src folder, the verilog file name should be same as the design name, otherwise specified in the command.
At first let's try run the processes with default config:

```
./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



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

```
[ERROR PDN-0175] Pitch 5.9800 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
```

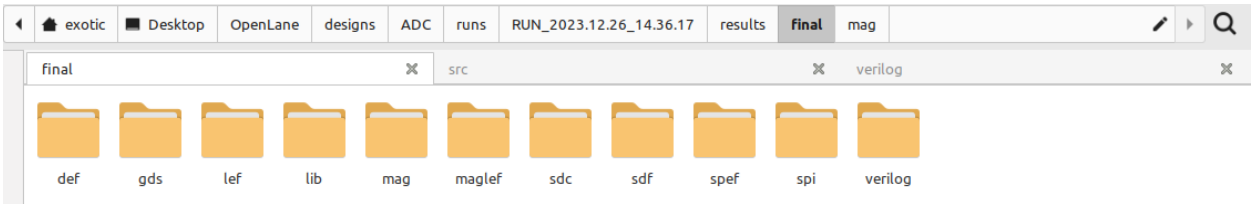
after adding above lines the error is gone but new error appeared.

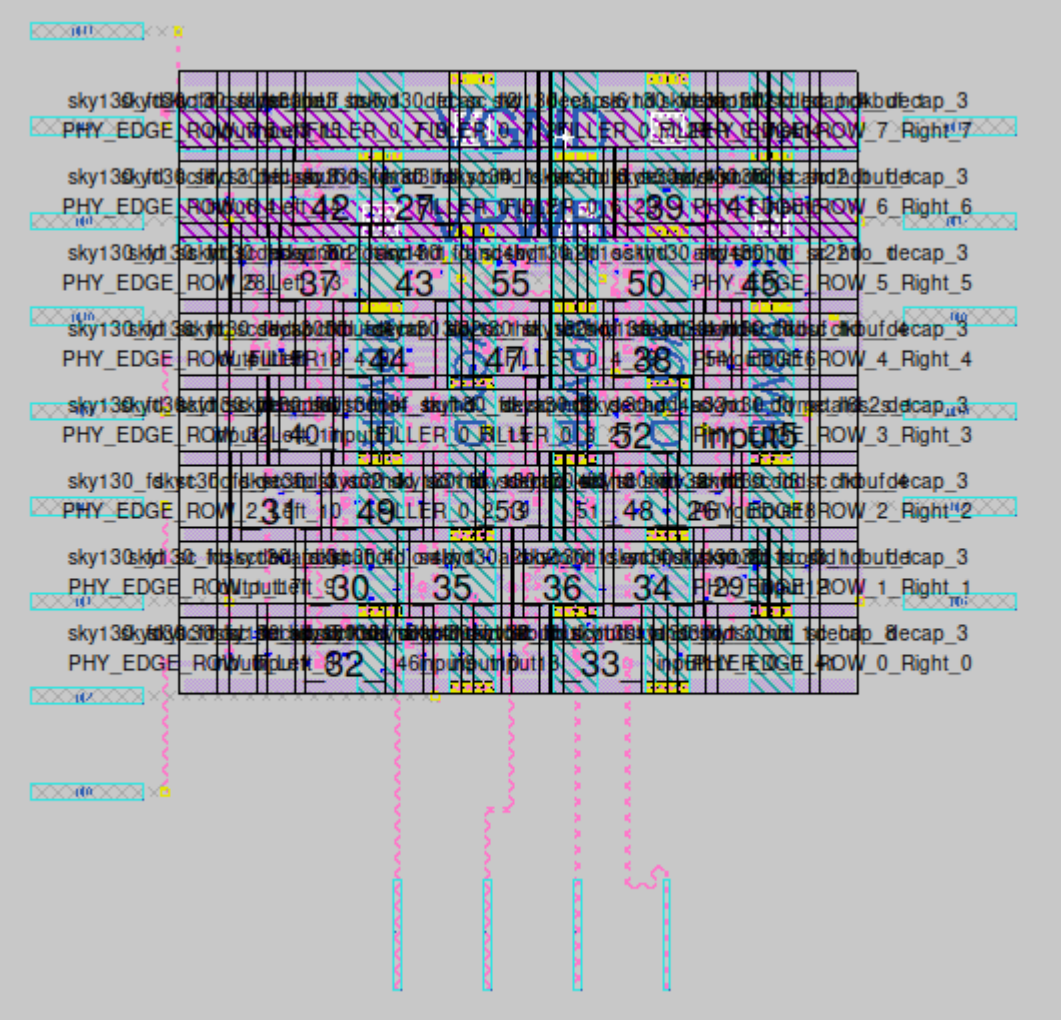
```
[ERROR]: during executing openroad script /openlane/scripts/openroad/gpl.tcl
[ERROR]: Log: designs/ADC/runs/RUN_2023.12.26_14.25.51/logs/placement/6-global_skip_io.log
[ERROR]: Last 10 lines:
[INFO GPL-0017] NonPlaceInstsArea: 66313600
[INFO GPL-0018] PlaceInstsArea: 294032000
[INFO GPL-0019] Util(%): 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-puhttps://github.com/syedimaduddin/msvsd4bituc/tree/main/Week-2blic

Run ALIGN:

```
python3 -m venv general
source general/bin/activate
schematic2layout.py ../ADC/Analog -p ../pdks/SKY130_PDK/
```

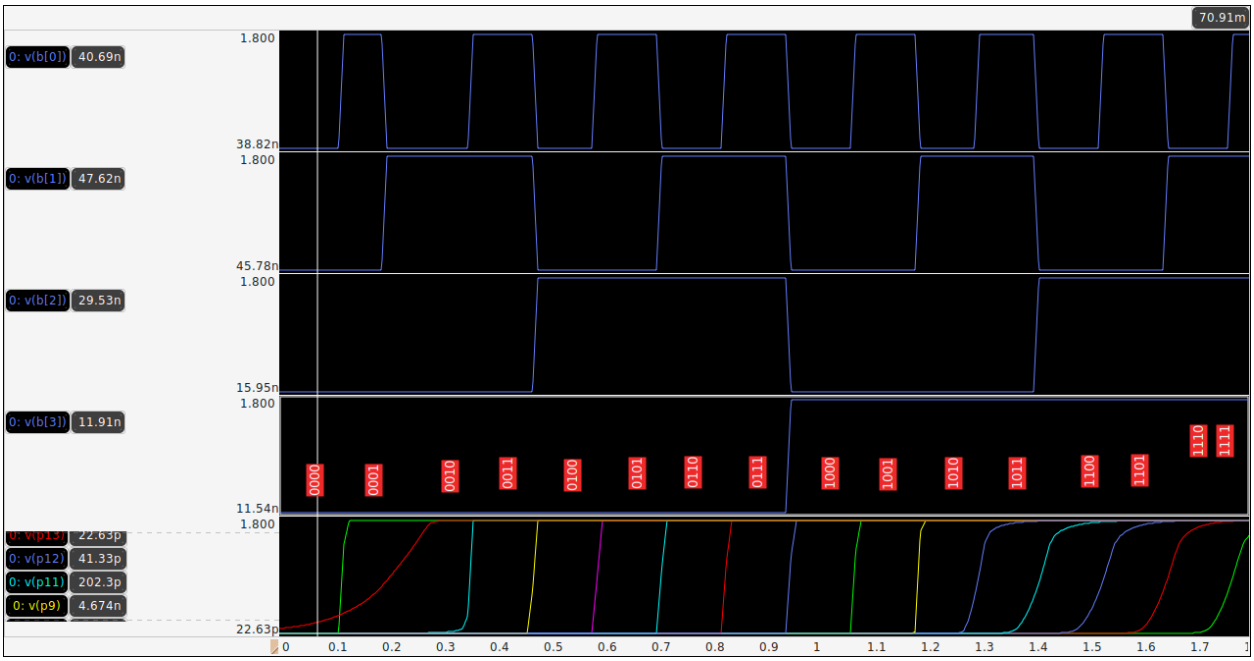
- simulate the complete (ultra small) ADC

Prelayout simulation:

To run these standard cells the “TT_MODELS” content should be:

‘’’

name=TT_MODELS
only_toplevel=true
format="tclevel(@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

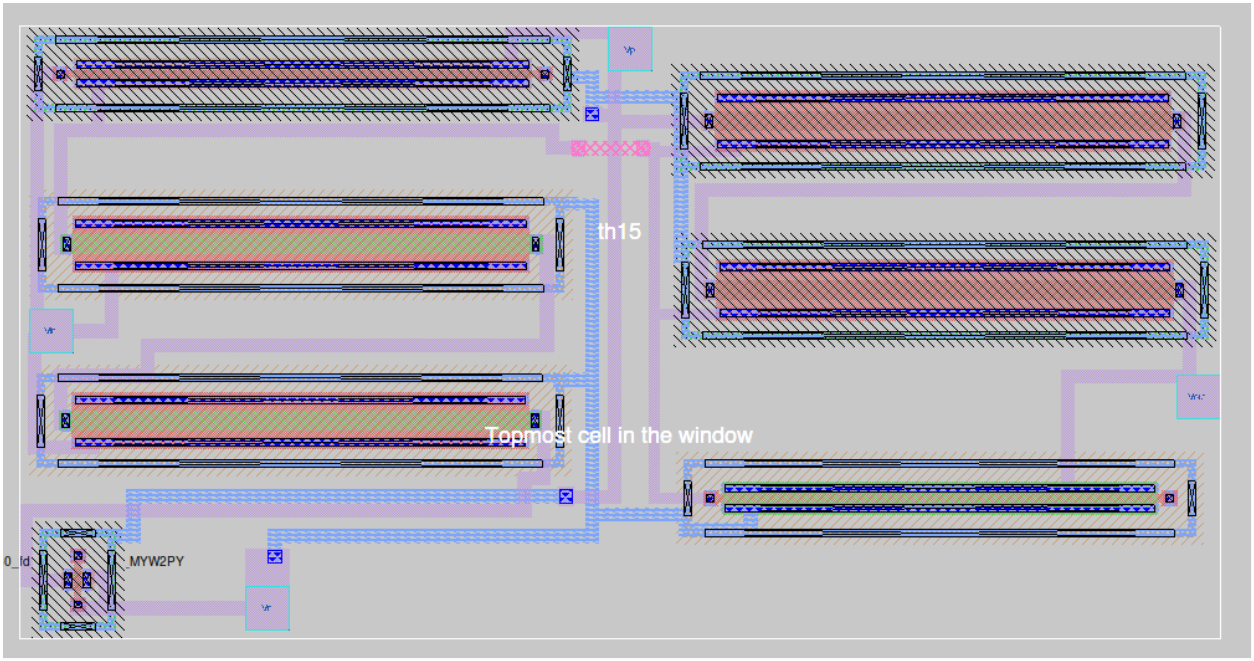
:specialopen netlist

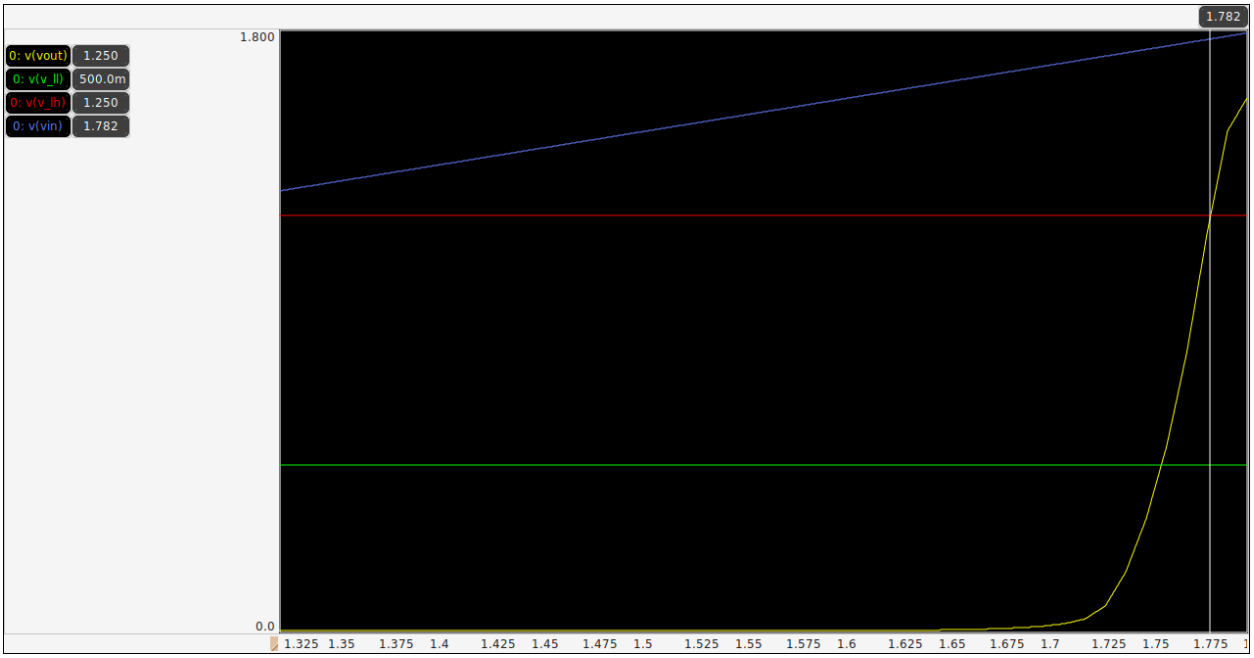
Save netlist (save and autowrite)

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

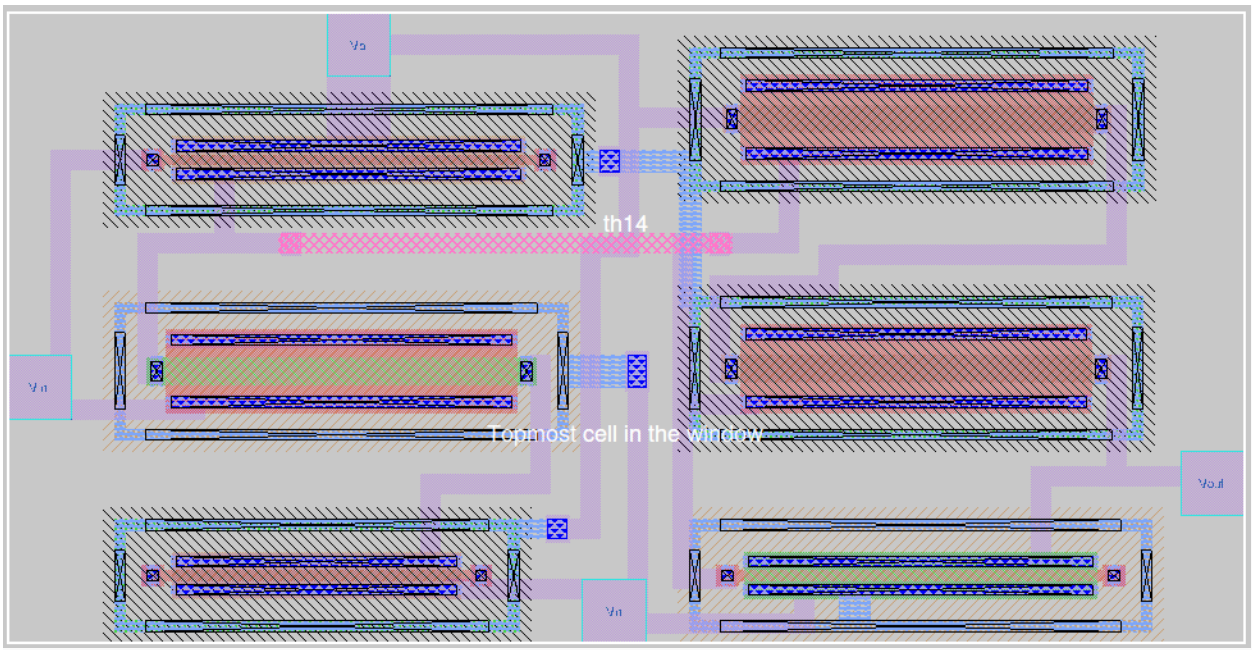
Analog Design:

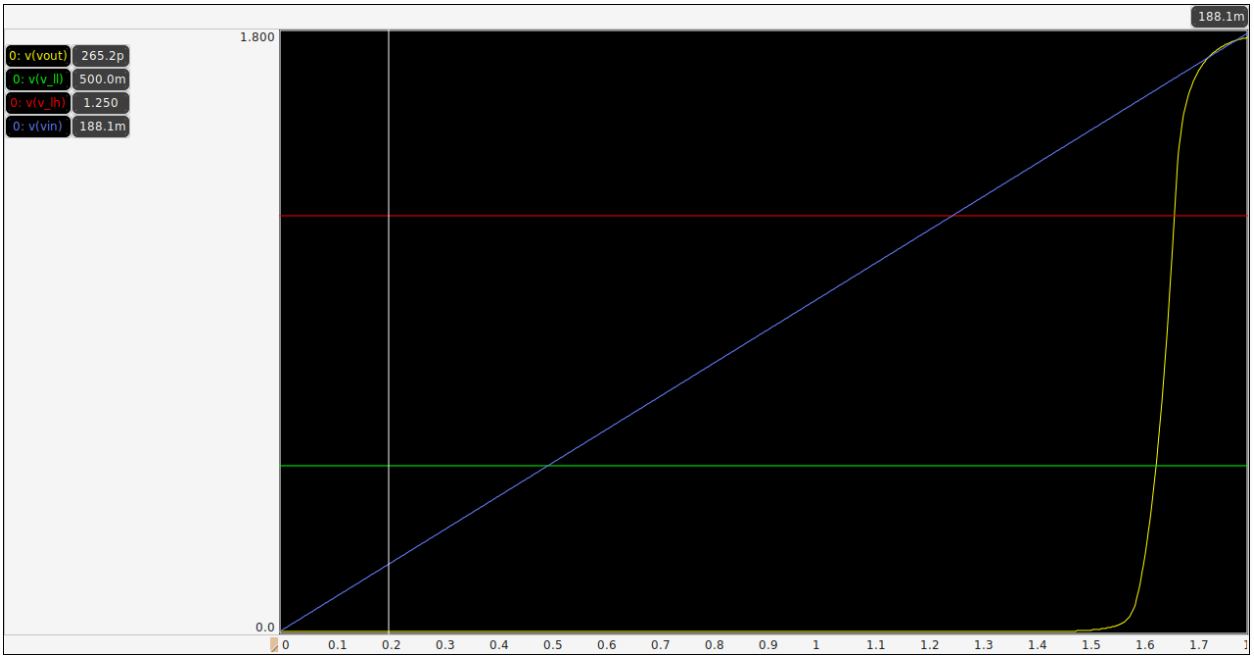
Th. 15 (1.782)



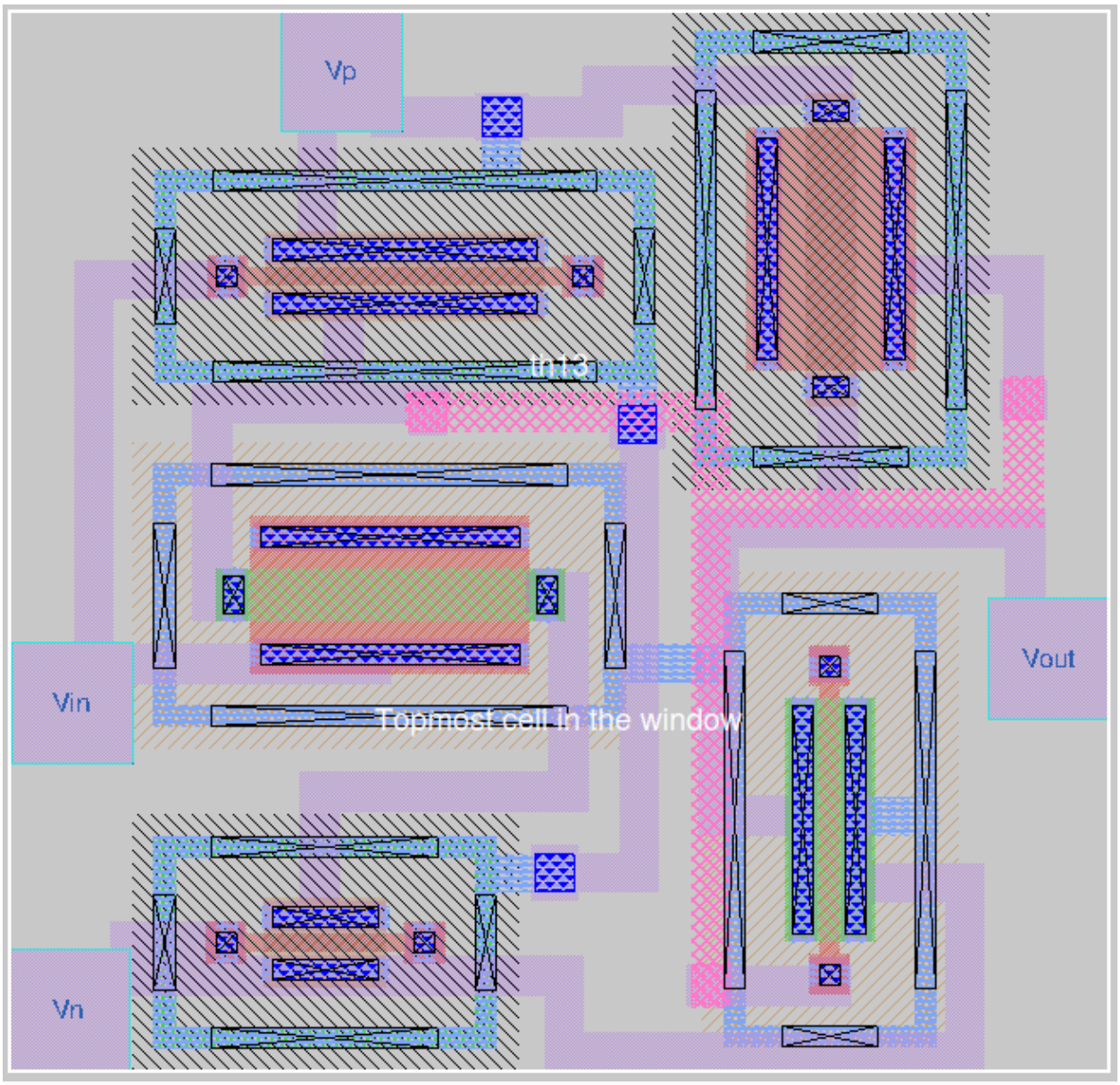


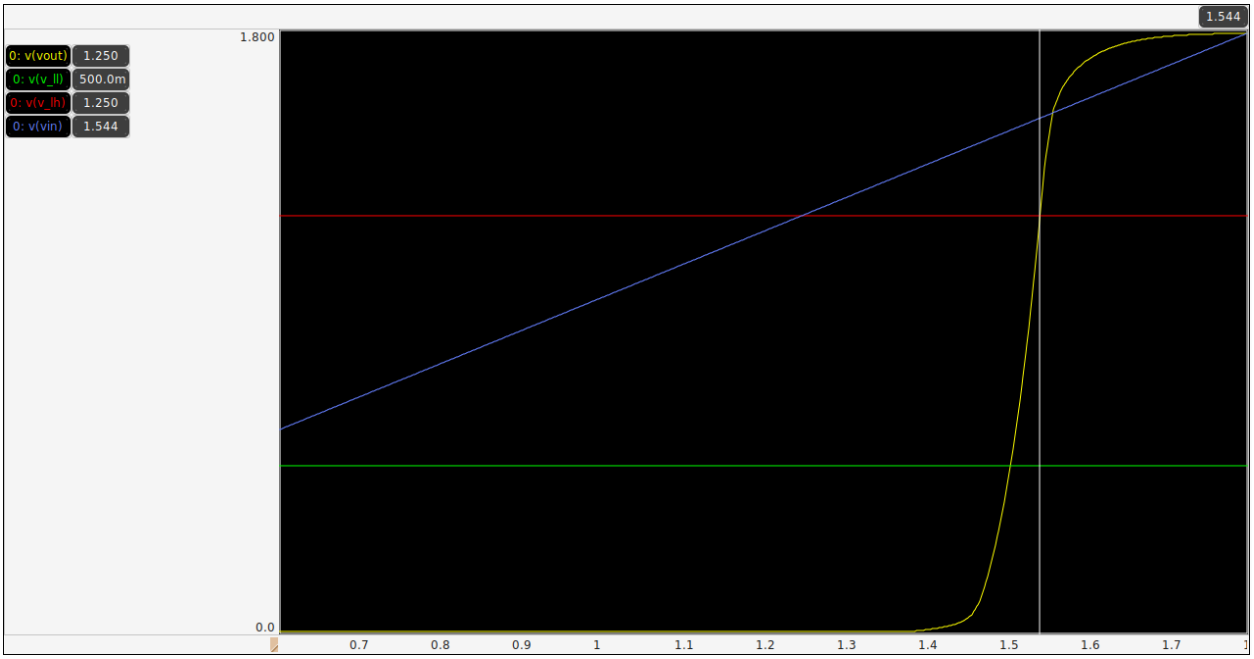
Th. 14 (1.663)


th14
Topmost cell in the window

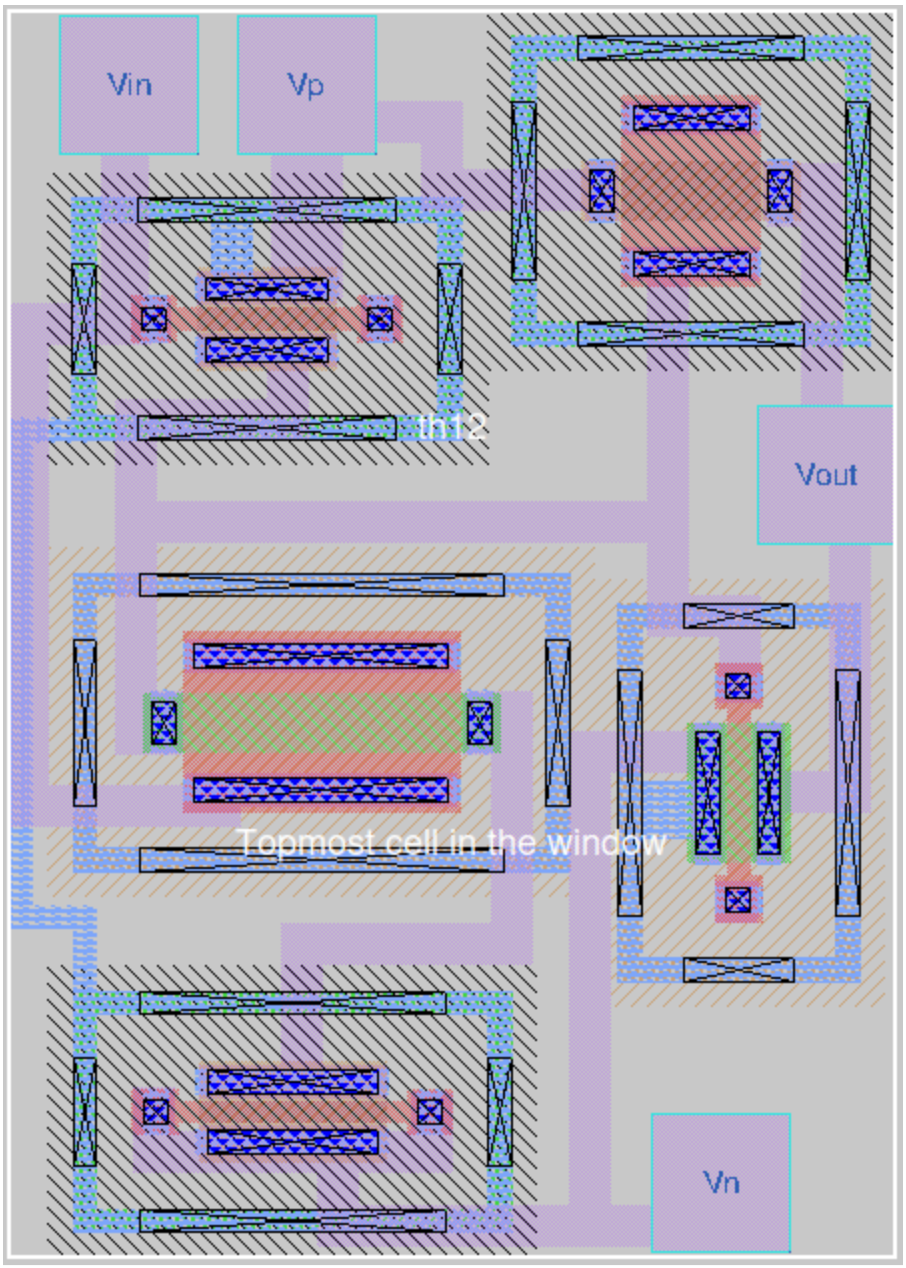

188.1m
0: v(vout) 265.2p
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 188.1m
1.800
0.0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7

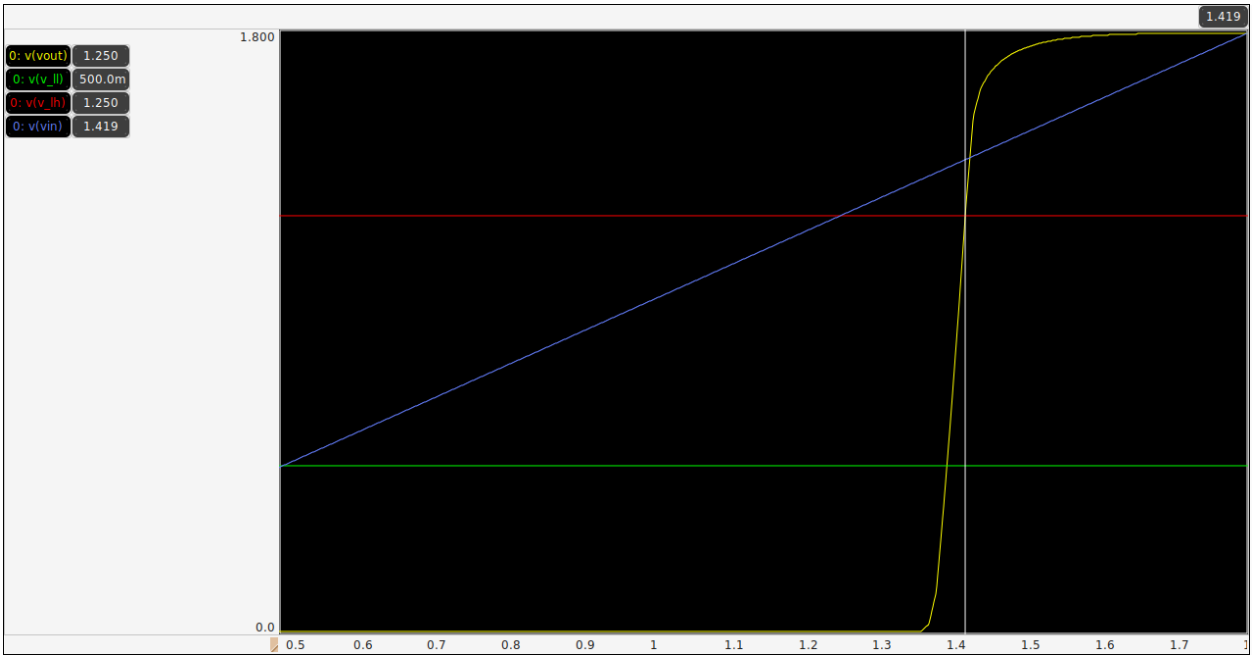
Th. 13 (1.544)


Vp
Vin
Vn
Vout
Topmost cell in the window

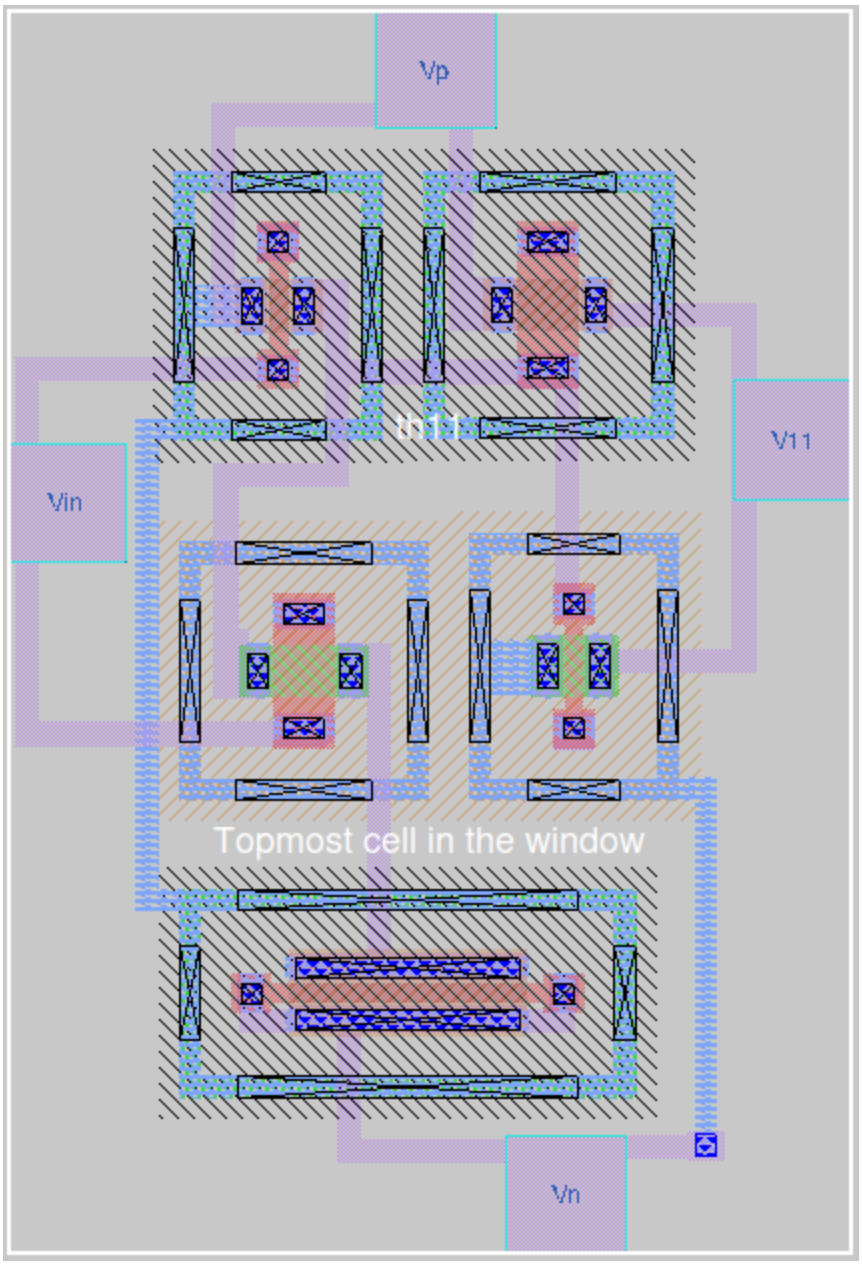

1.544
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.544
1.800
0.0
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

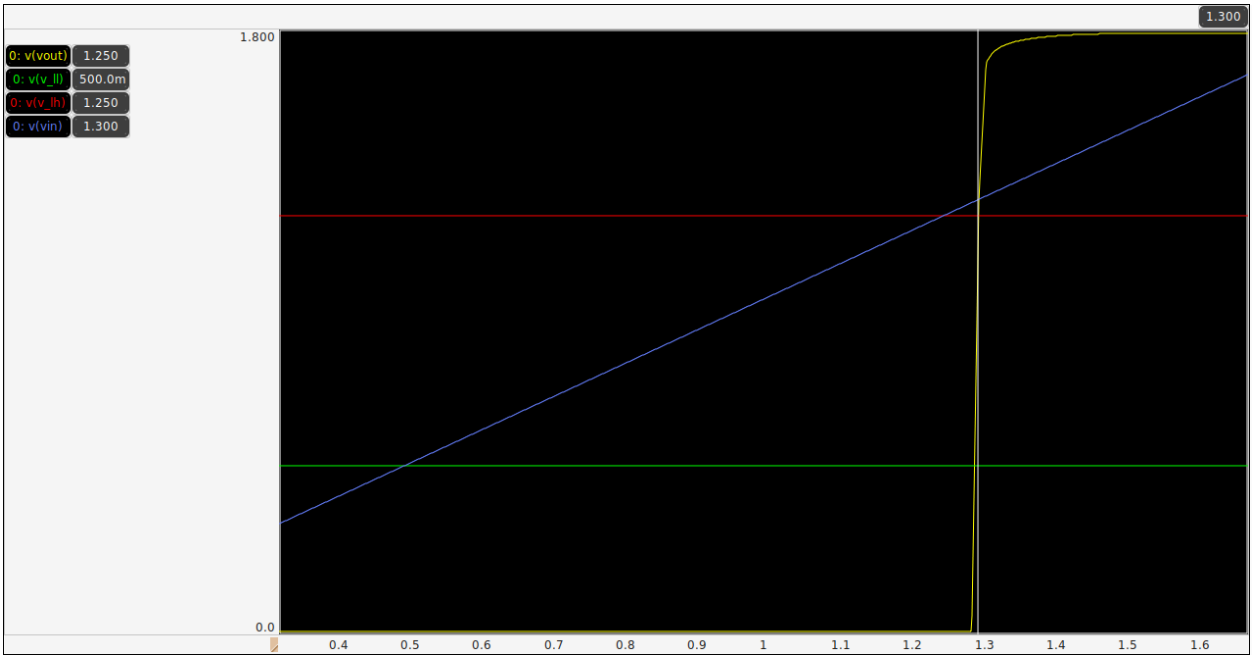
Th. 12 (1.419)


Vin
Vp
th12
Vout
Topmost cell in the window
Vn

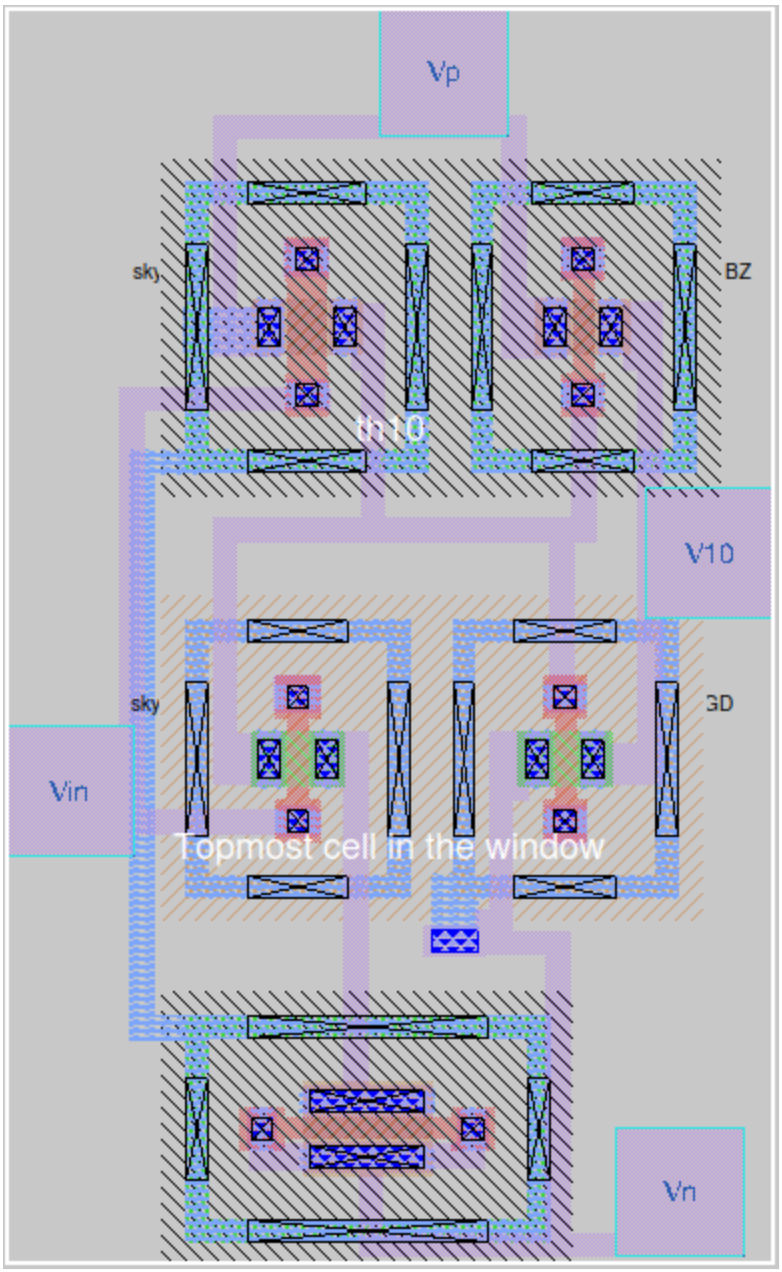

1.419
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.419
1.800
0.0
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

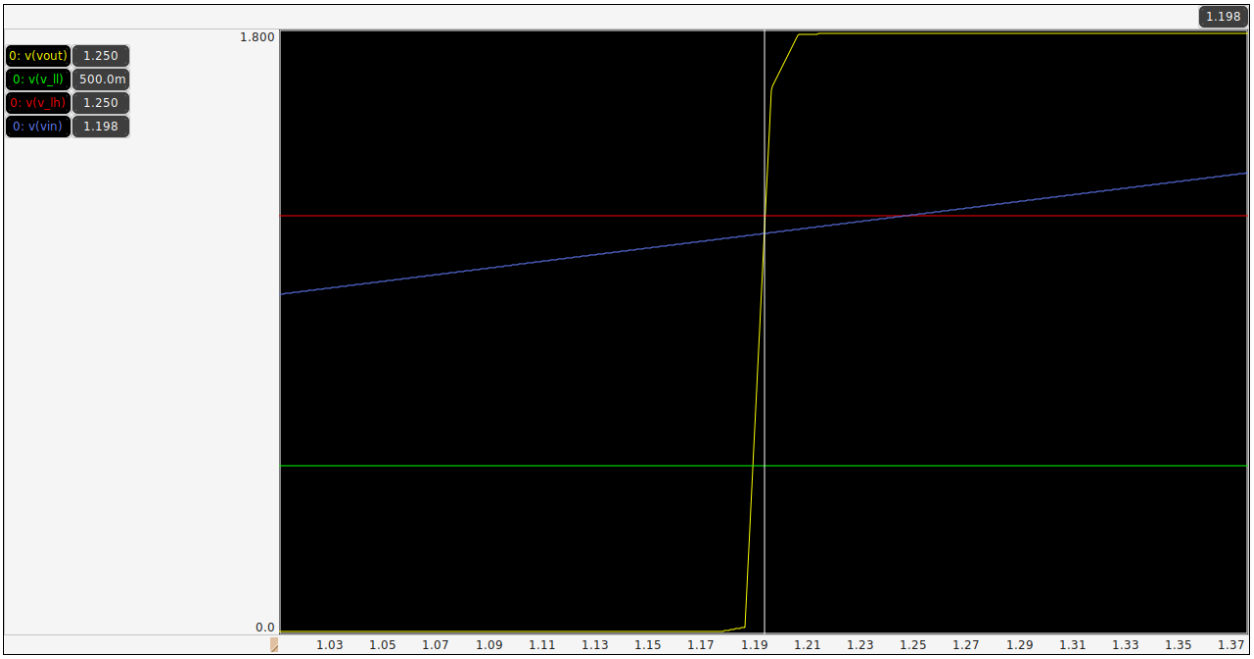
Th. 11 (1.30)


Vp
Vin
V11
Topmost cell in the window
Vn

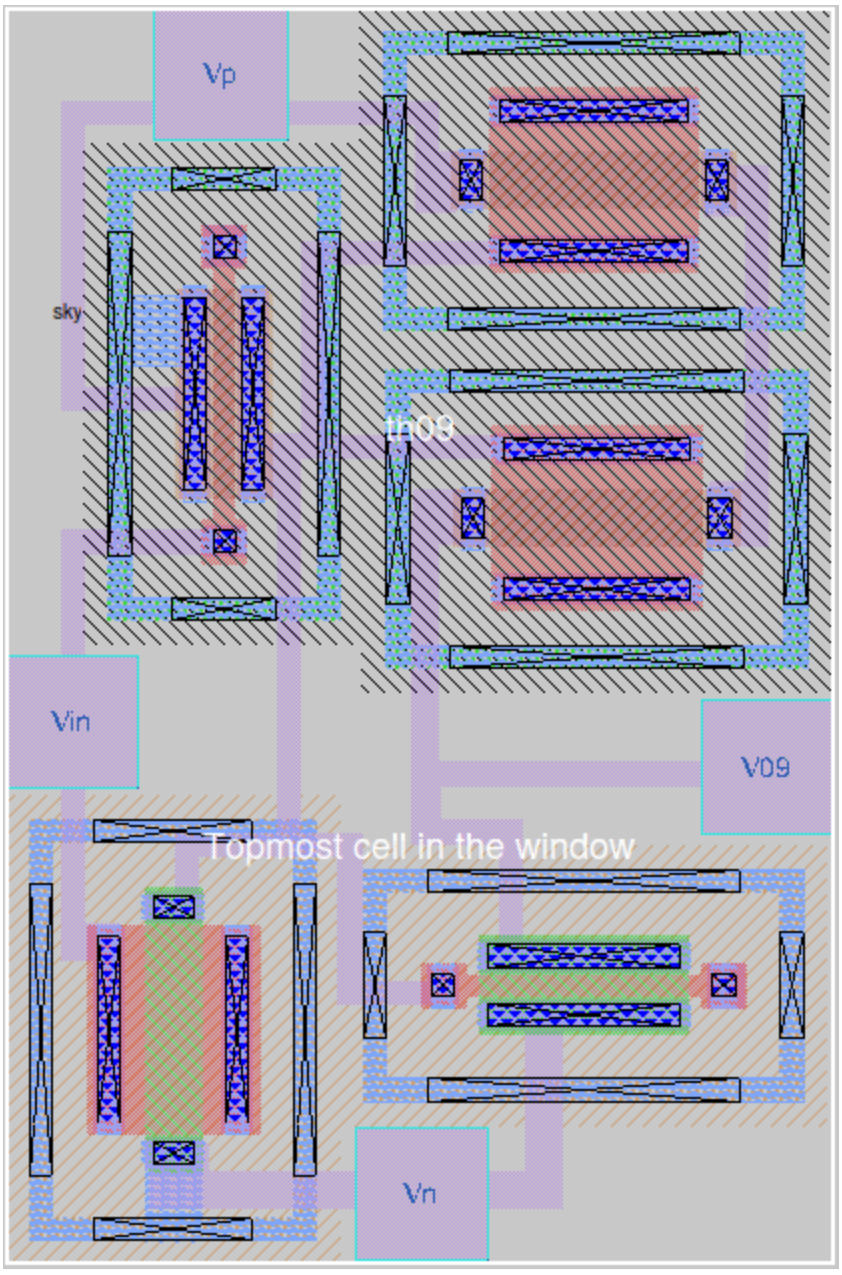

1.300
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.300
1.800
0.0
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6

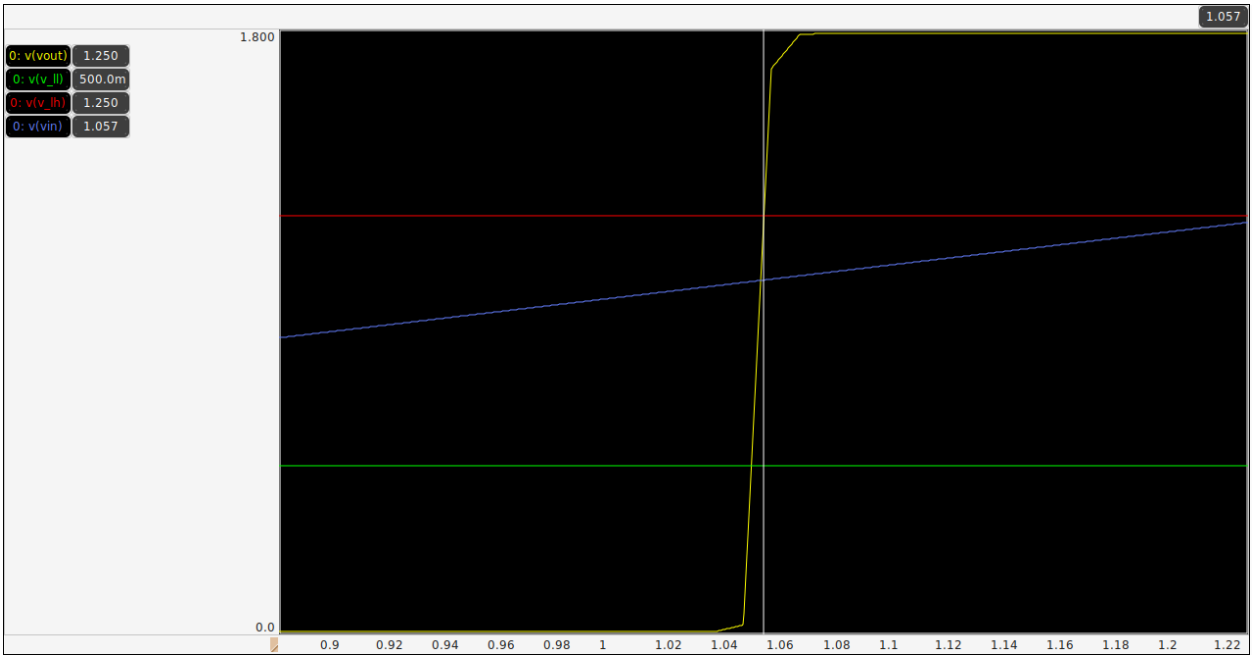
Th. 10 (1.198)


Vp
sky
BZ
th10
V10
sky
3D
Vin
Topmost cell in the window
Vn

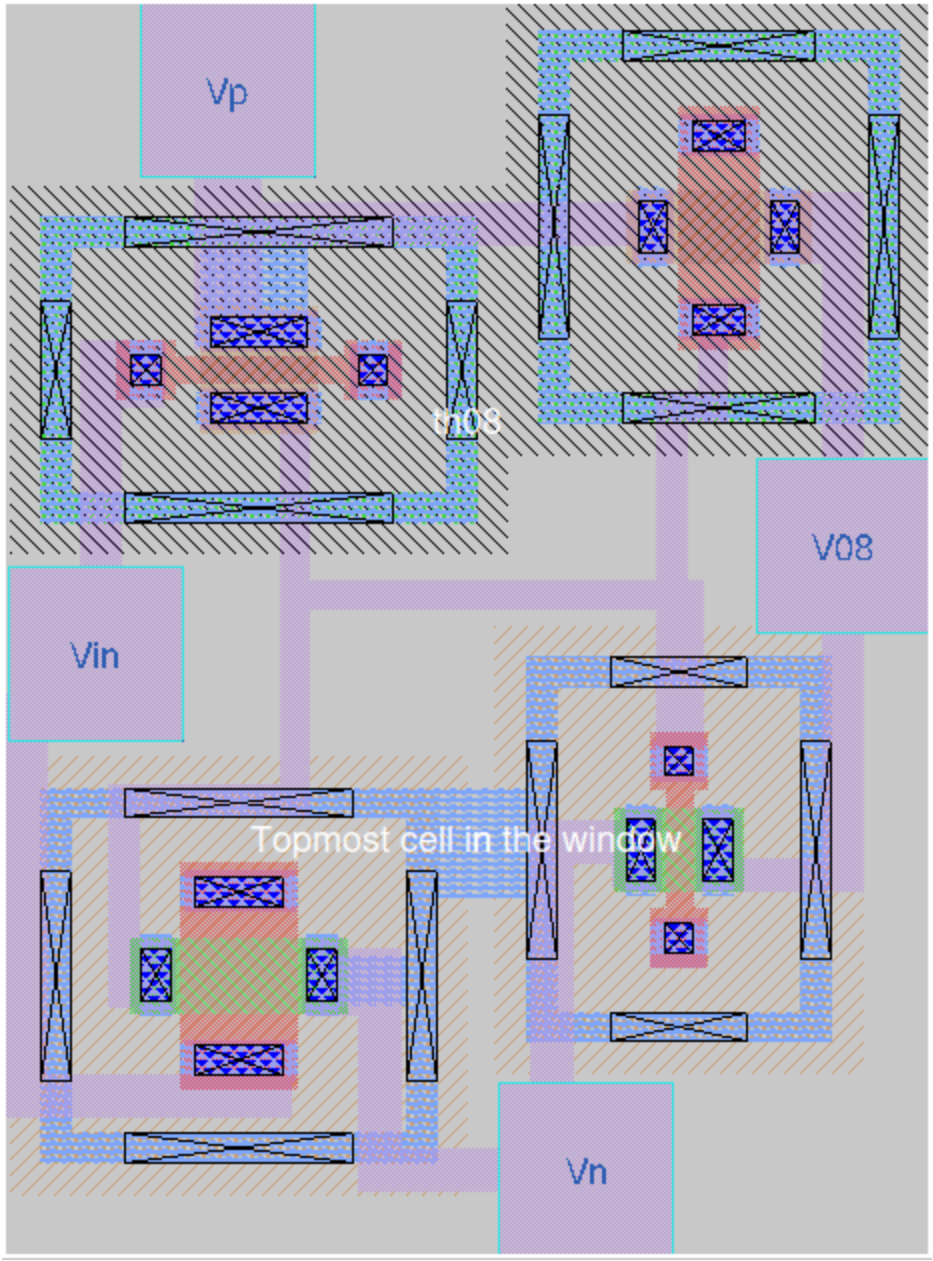

1.198
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.198
1.800
0.0
1.03 1.05 1.07 1.09 1.11 1.13 1.15 1.17 1.19 1.21 1.23 1.25 1.27 1.29 1.31 1.33 1.35 1.37

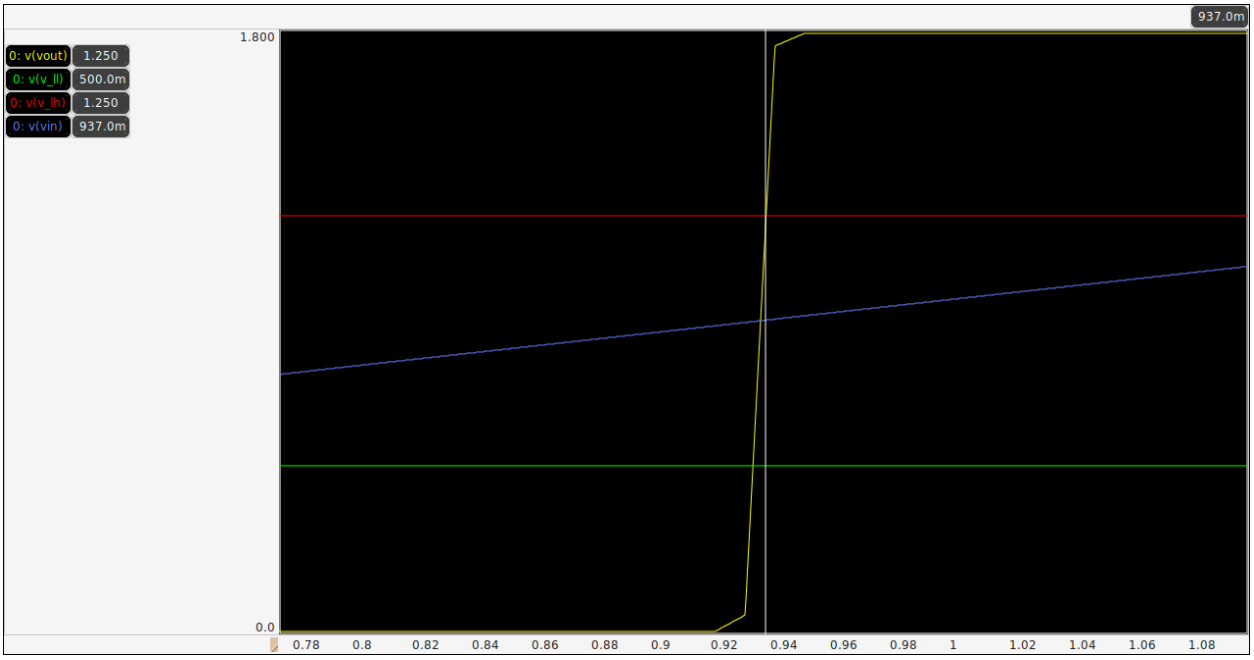
Th. 09 (1.057)


Vp
sky
th09
Vin
V09
Topmost cell in the window
Vn

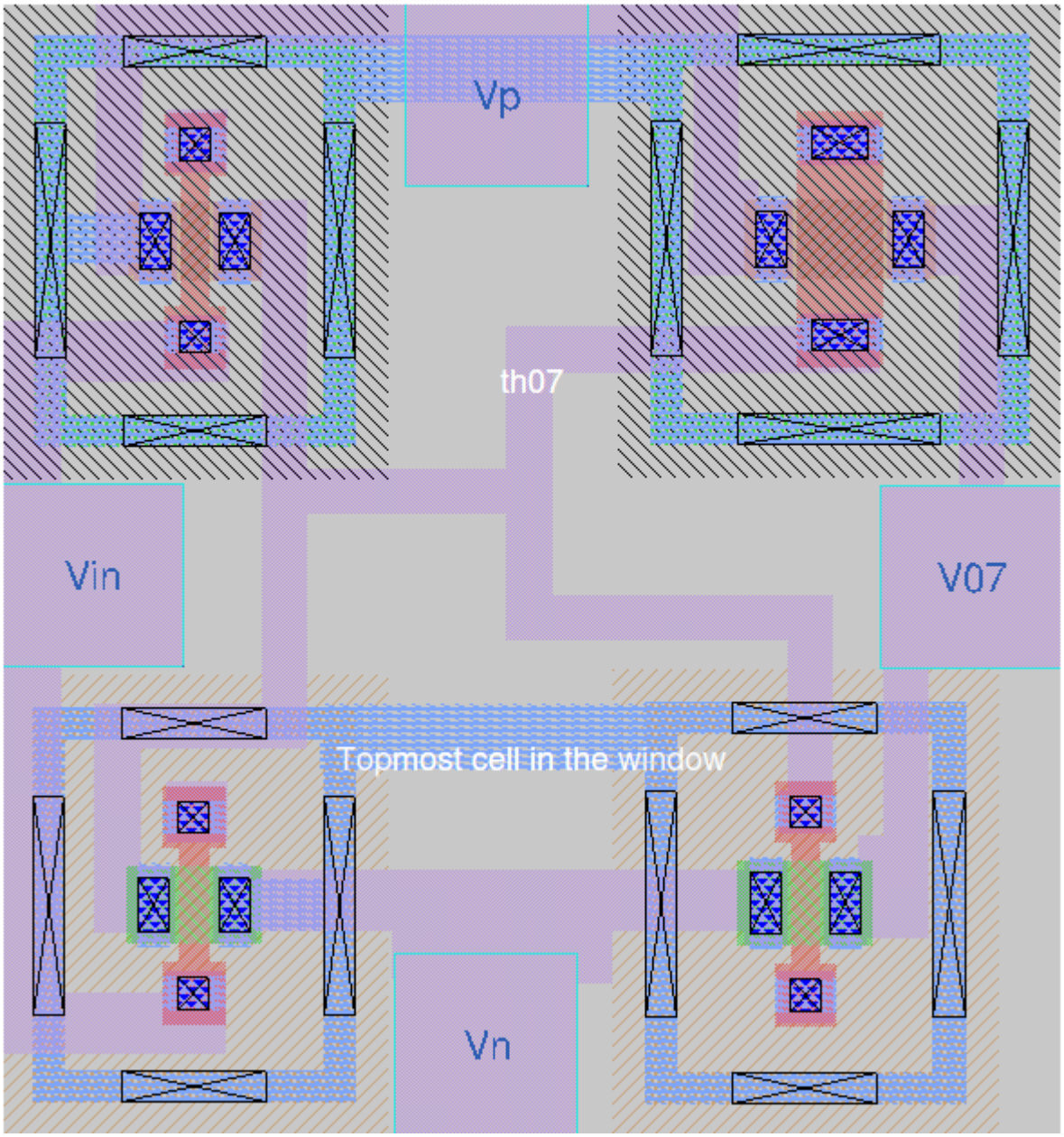

1.057
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 1.057
1.800
0.0
0.9 0.92 0.94 0.96 0.98 1 1.02 1.04 1.06 1.08 1.1 1.12 1.14 1.16 1.18 1.2 1.22

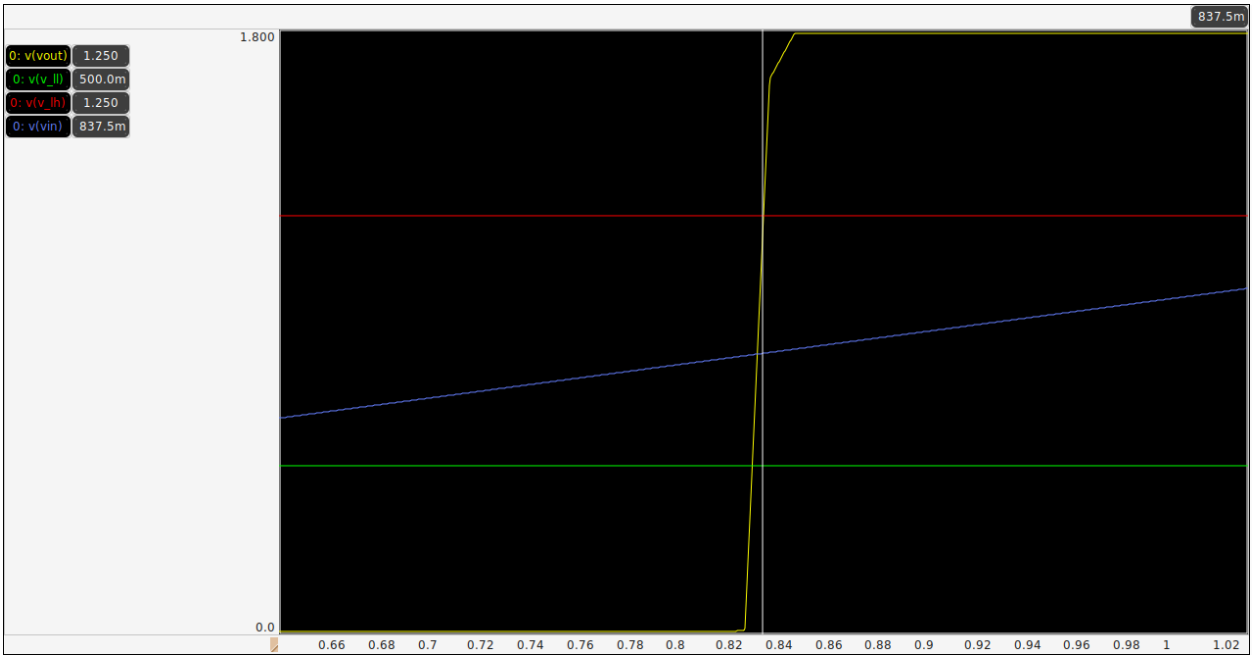
Th. 08 (0.937)


Vp
V08
Vin
Topmost cell in the window
Vn

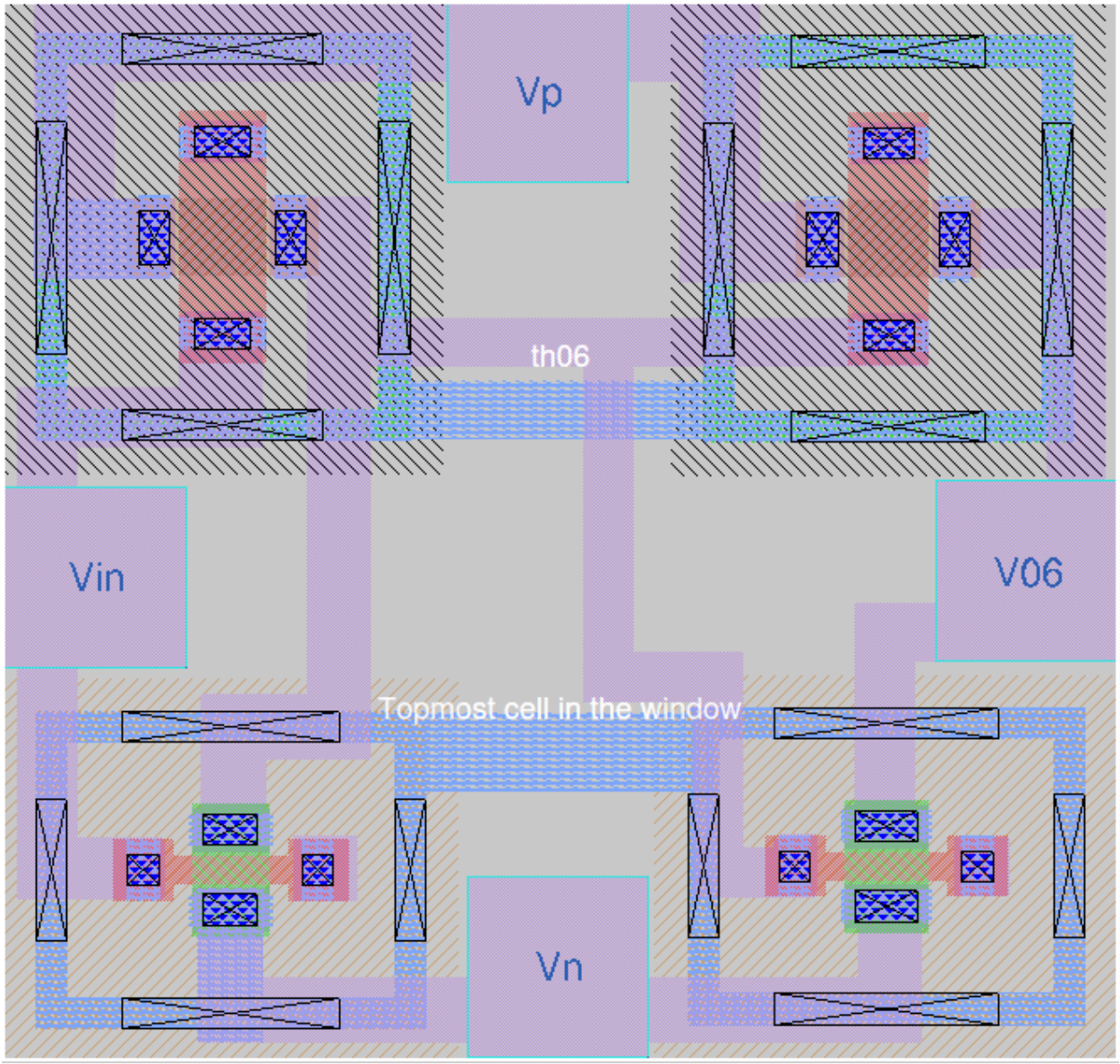

937.0m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 937.0m
1.800
0.0
0.78 0.8 0.82 0.84 0.86 0.88 0.9 0.92 0.94 0.96 0.98 1 1.02 1.04 1.06 1.08

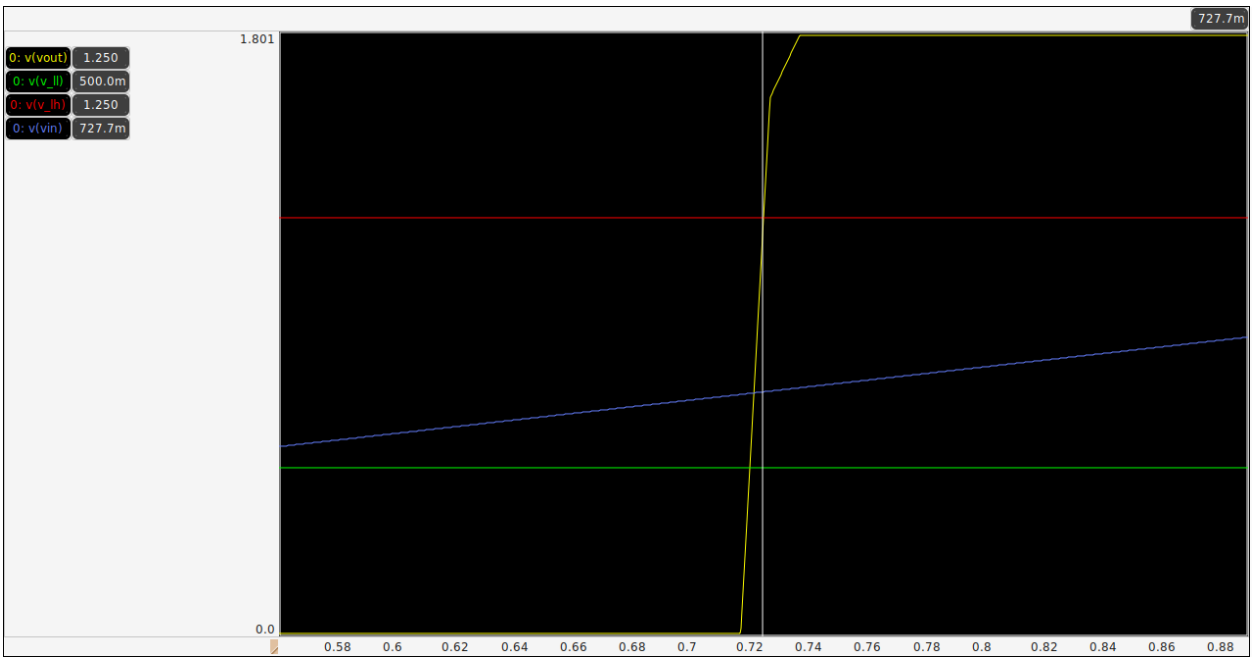
Th. 07 (0.8375)


Vp
th07
Vin
V07
Topmost cell in the window
Vn

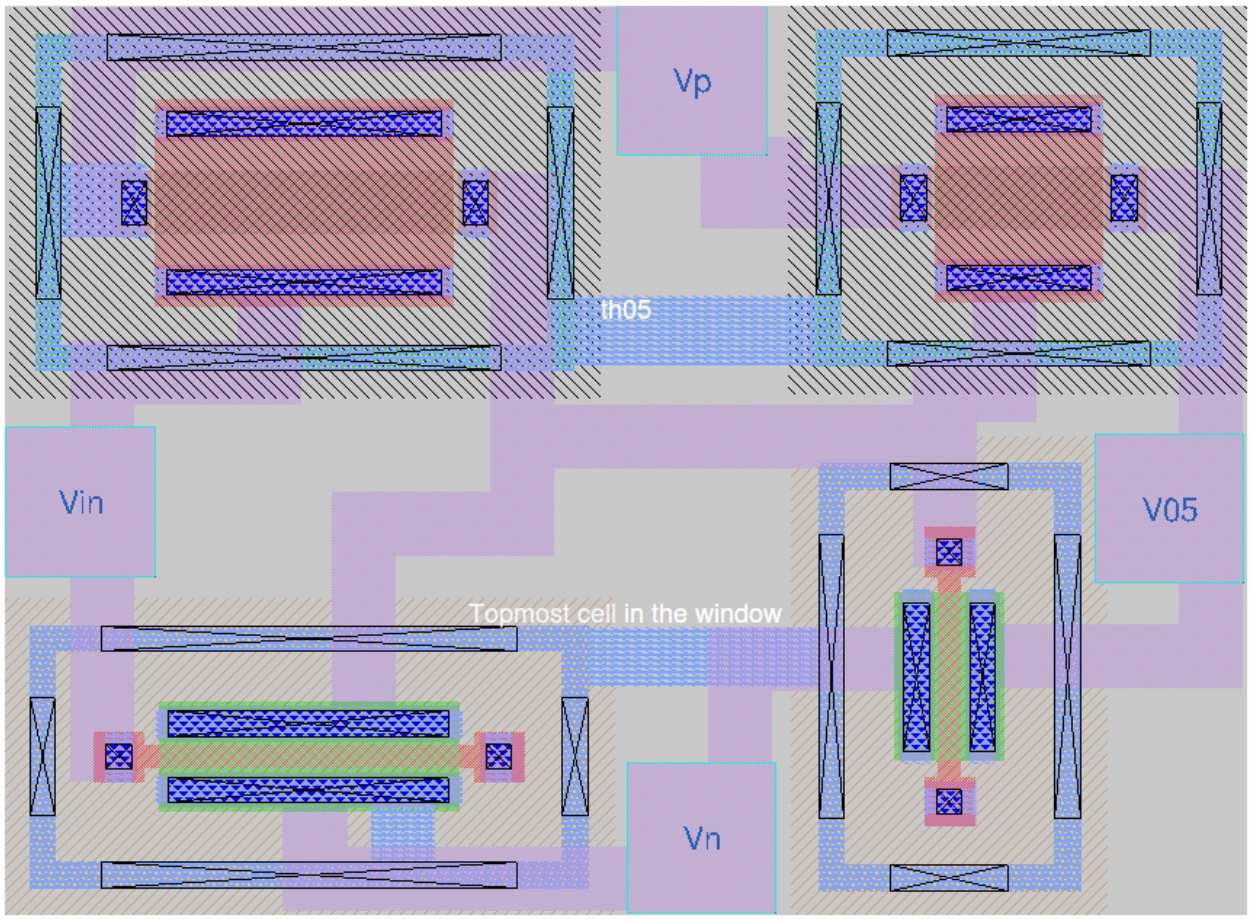

837.5m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 837.5m
1.800
0.0
0.66 0.68 0.7 0.72 0.74 0.76 0.78 0.8 0.82 0.84 0.86 0.88 0.9 0.92 0.94 0.96 0.98 1 1.02

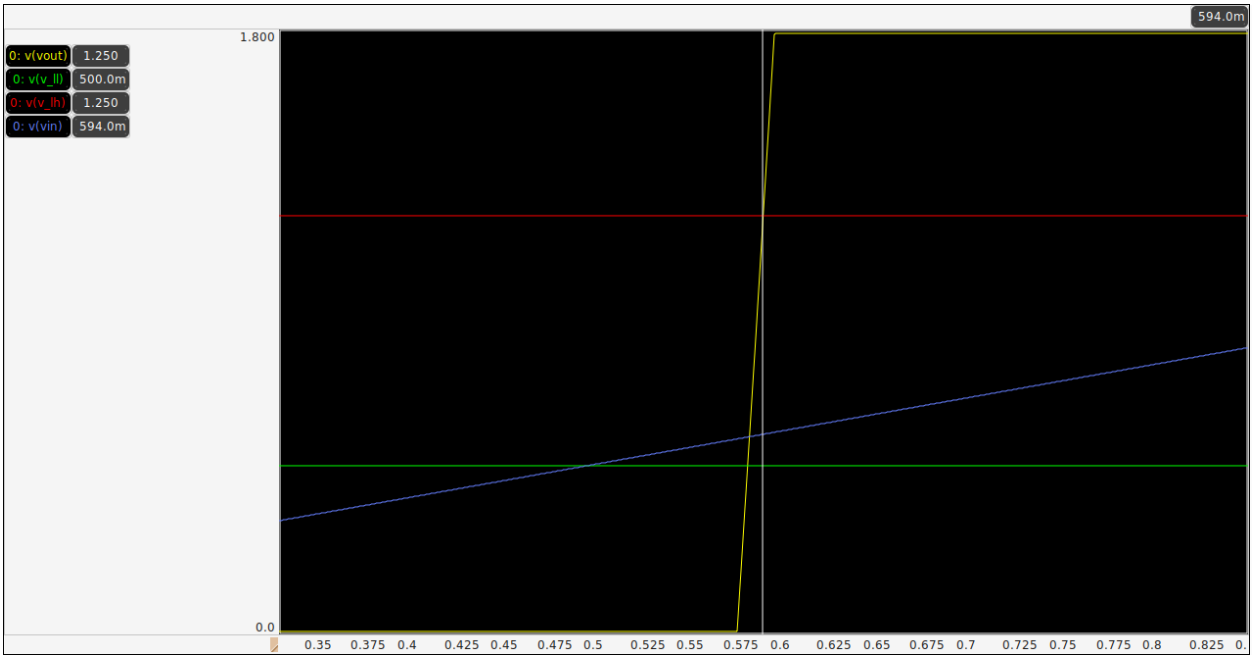
Th. 06 (0.7278)


Vp
th06
Vin
V06
Topmost cell in the window
Vn

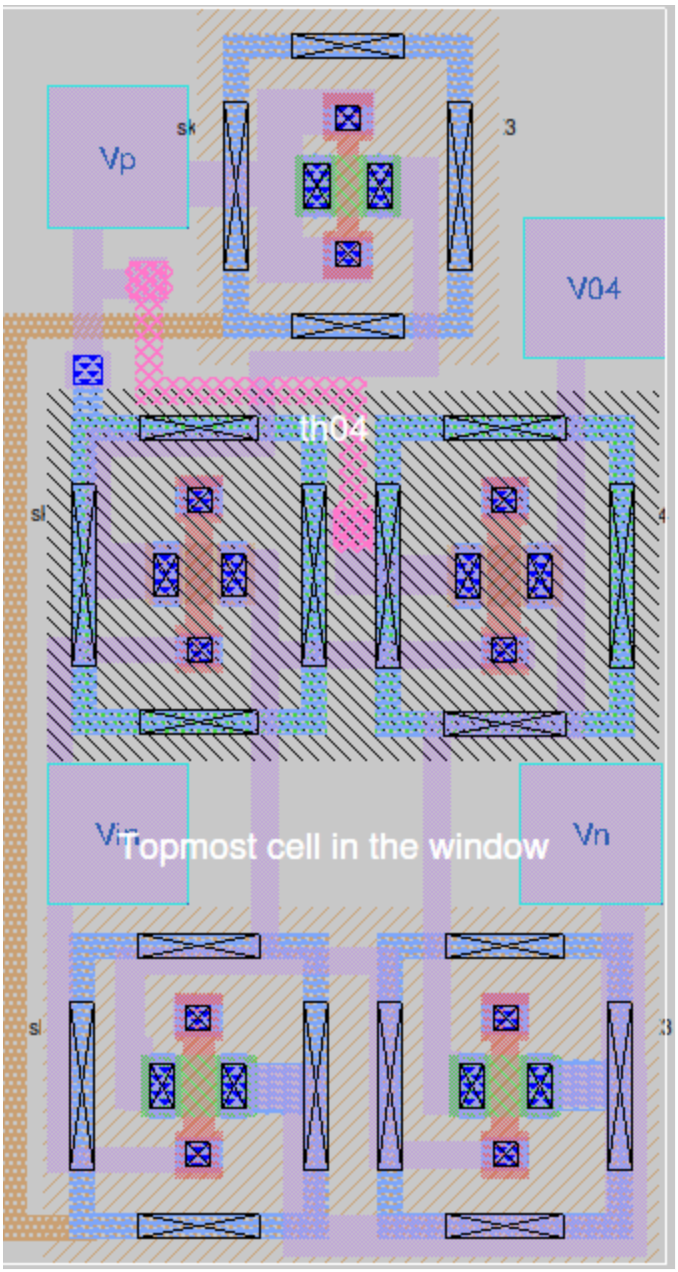

727.7m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 727.7m
1.801
0.0
0.58 0.6 0.62 0.64 0.66 0.68 0.7 0.72 0.74 0.76 0.78 0.8 0.82 0.84 0.86 0.88

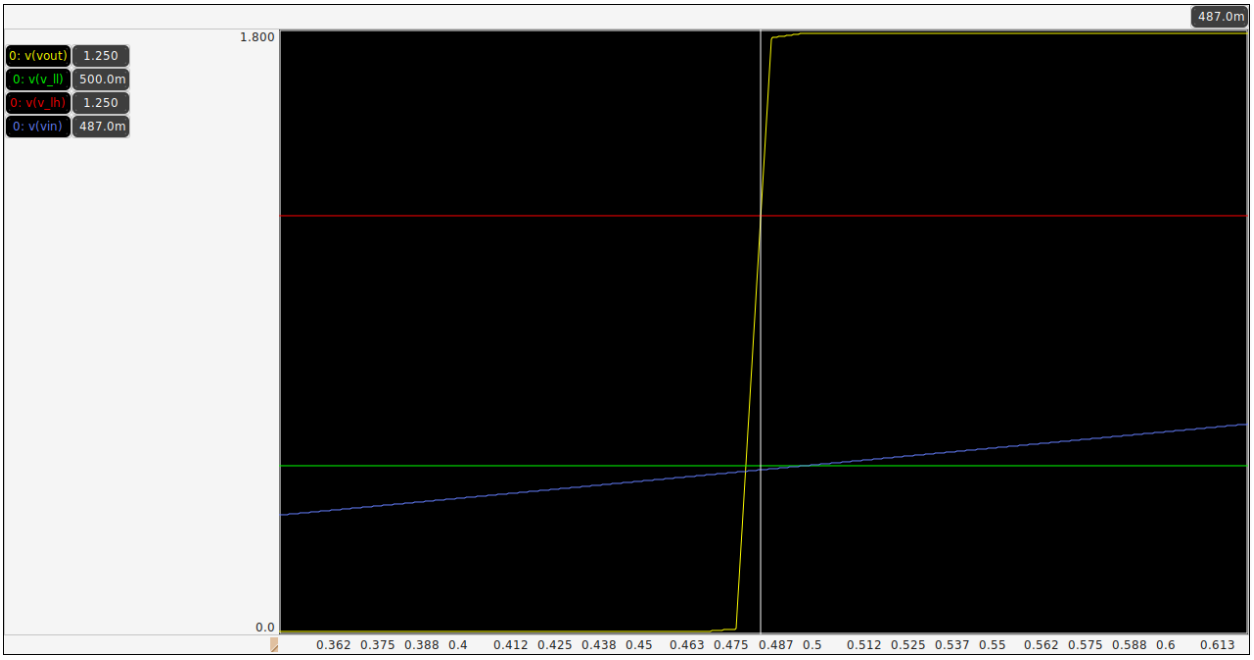
Th. 05


Vp
th05
Vin
V05
Topmost cell in the window
Vn

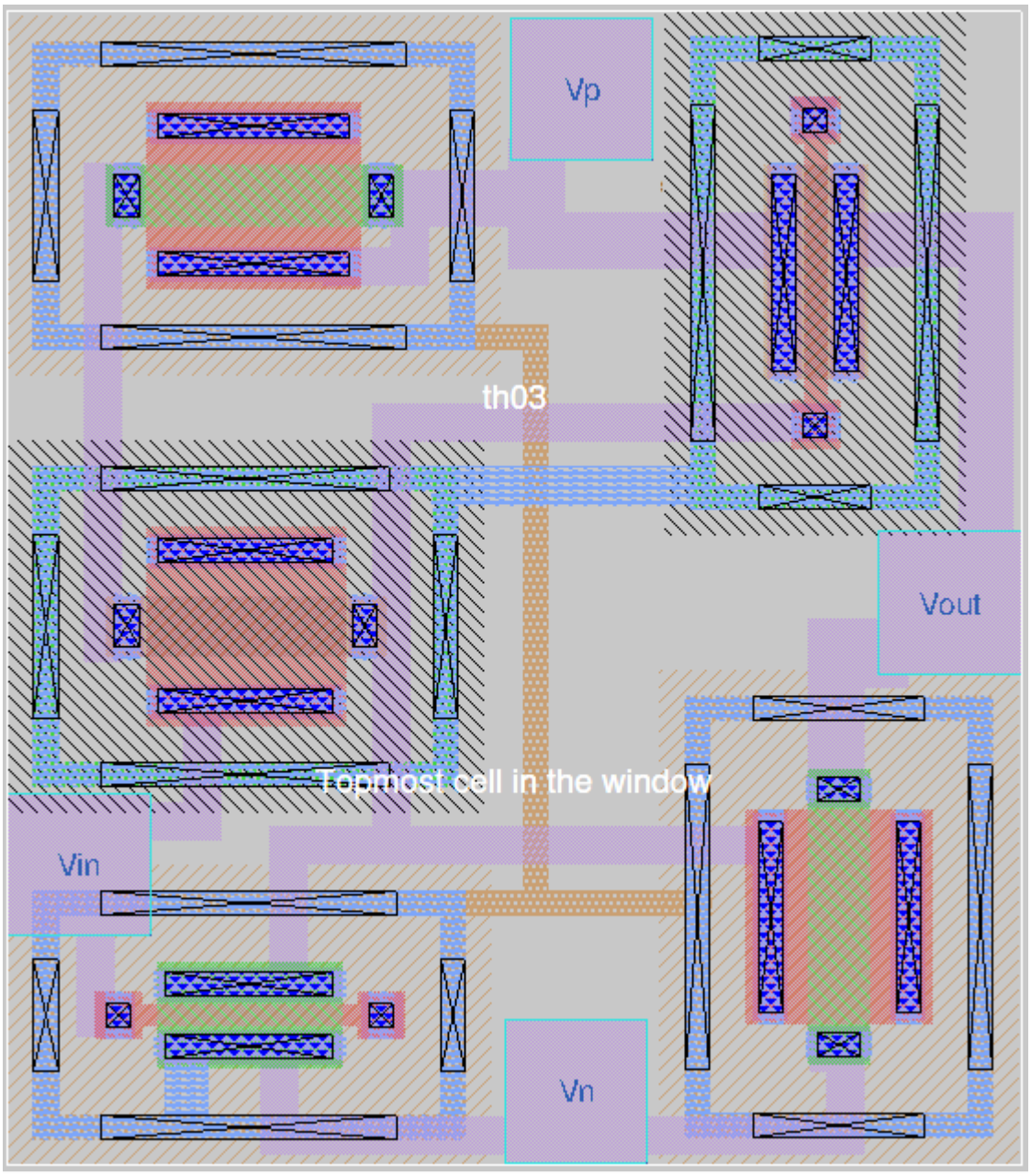

594.0m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 594.0m
1.800
0.0
0.35 0.375 0.4 0.425 0.45 0.475 0.5 0.525 0.55 0.575 0.6 0.625 0.65 0.675 0.7 0.725 0.75 0.775 0.8 0.825

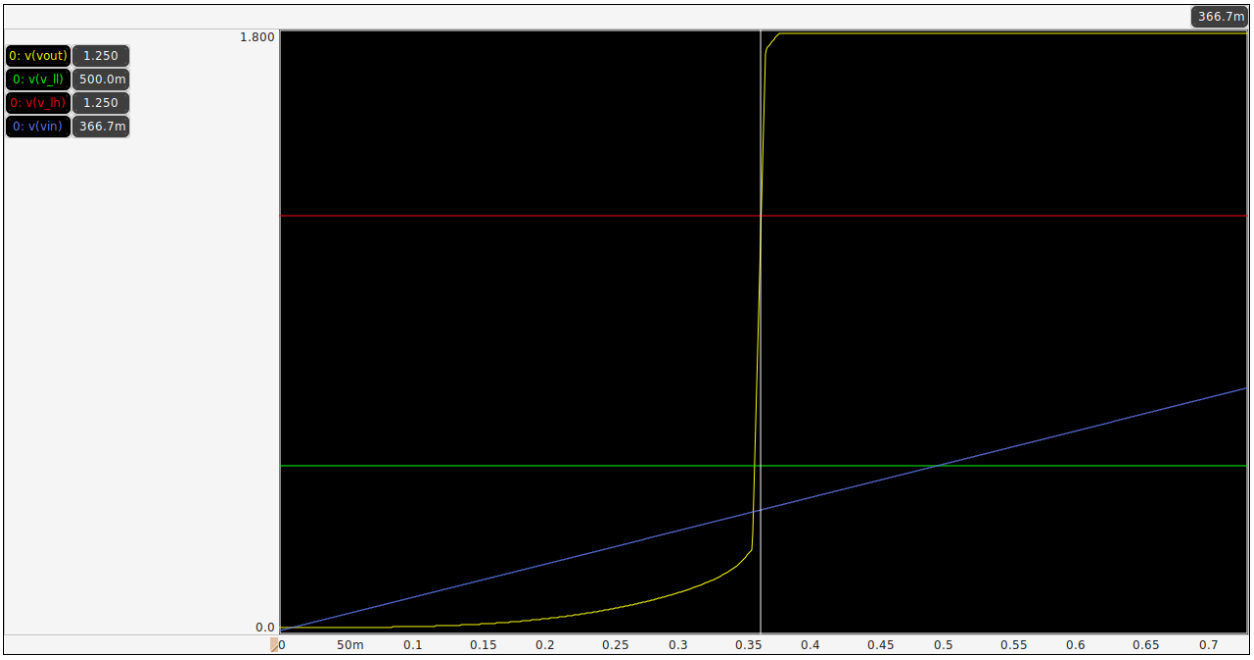
Th. 04 (0.487)


Vp
V04
th04
Vn
Topmost cell in the window

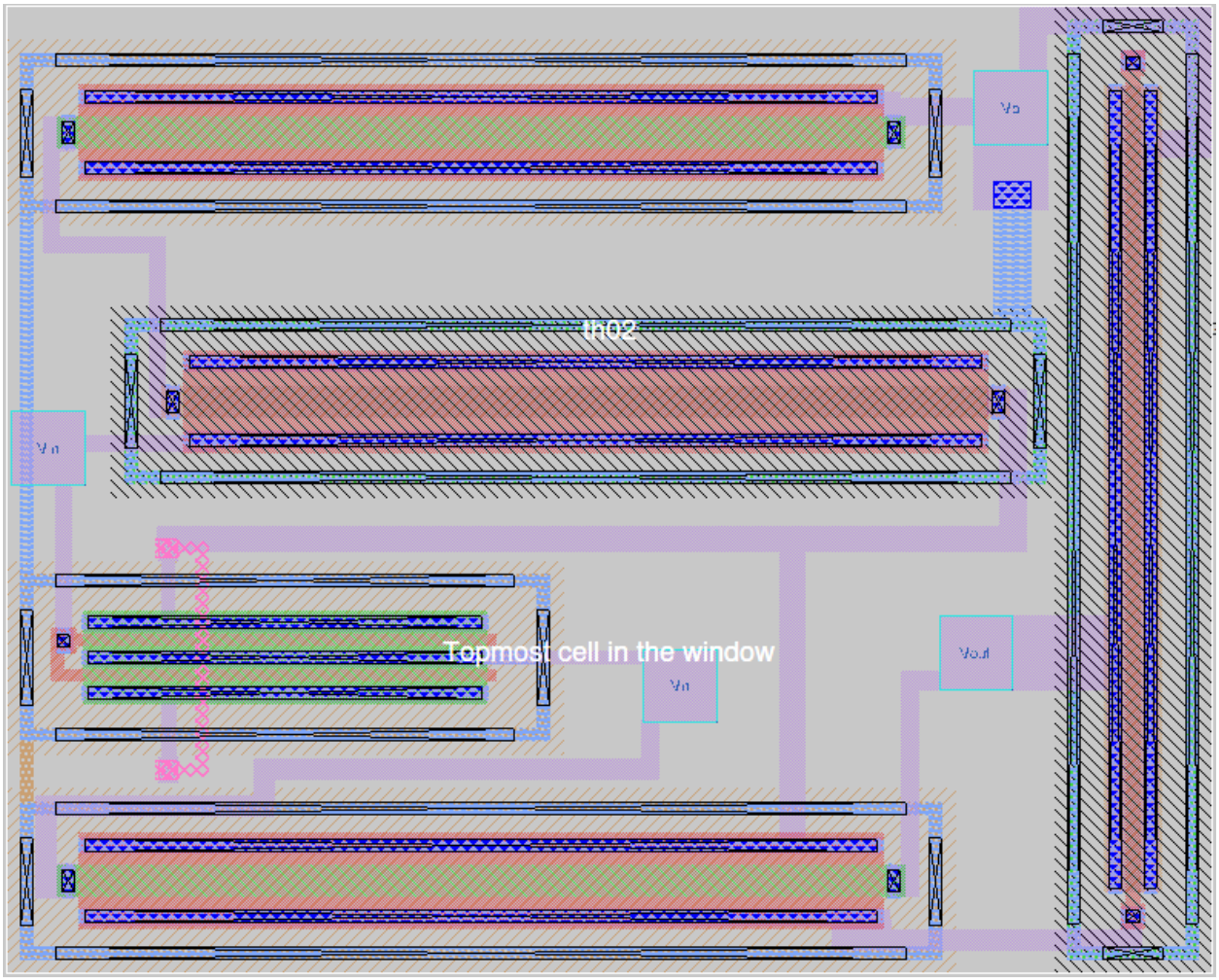

487.0m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 487.0m
1.800
0.0
0.362 0.375 0.388 0.4 0.412 0.425 0.438 0.45 0.463 0.475 0.487 0.5 0.512 0.525 0.537 0.55 0.562 0.575 0.588 0.6 0.613

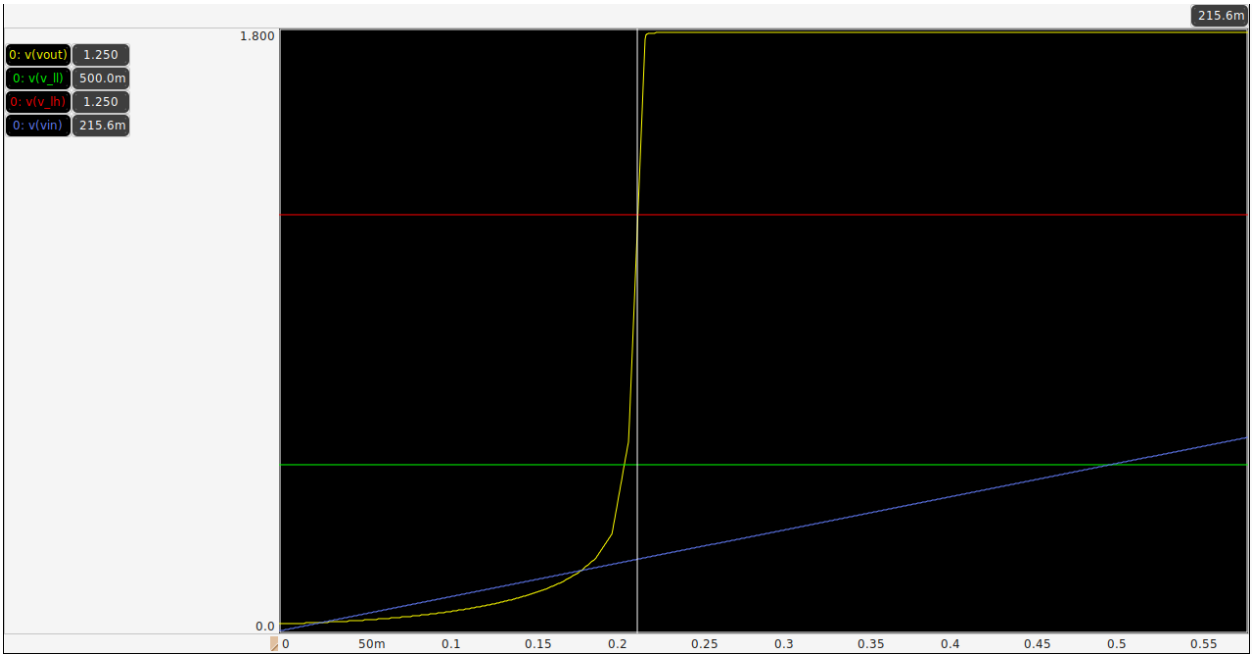
Th. 03 (0.3667)


Vp
th03
Vout
Topmost cell in the window
Vin
Vn

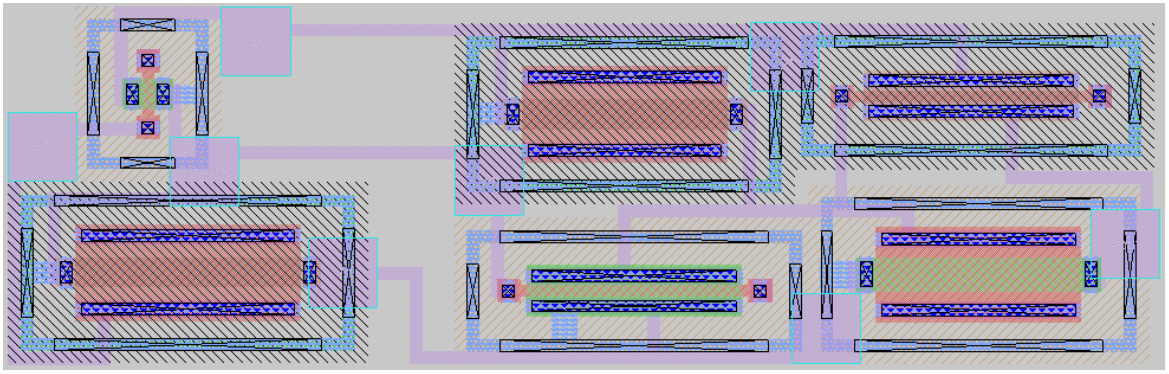

366.7m
0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 366.7m
1.800
0.0
0
50m
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7

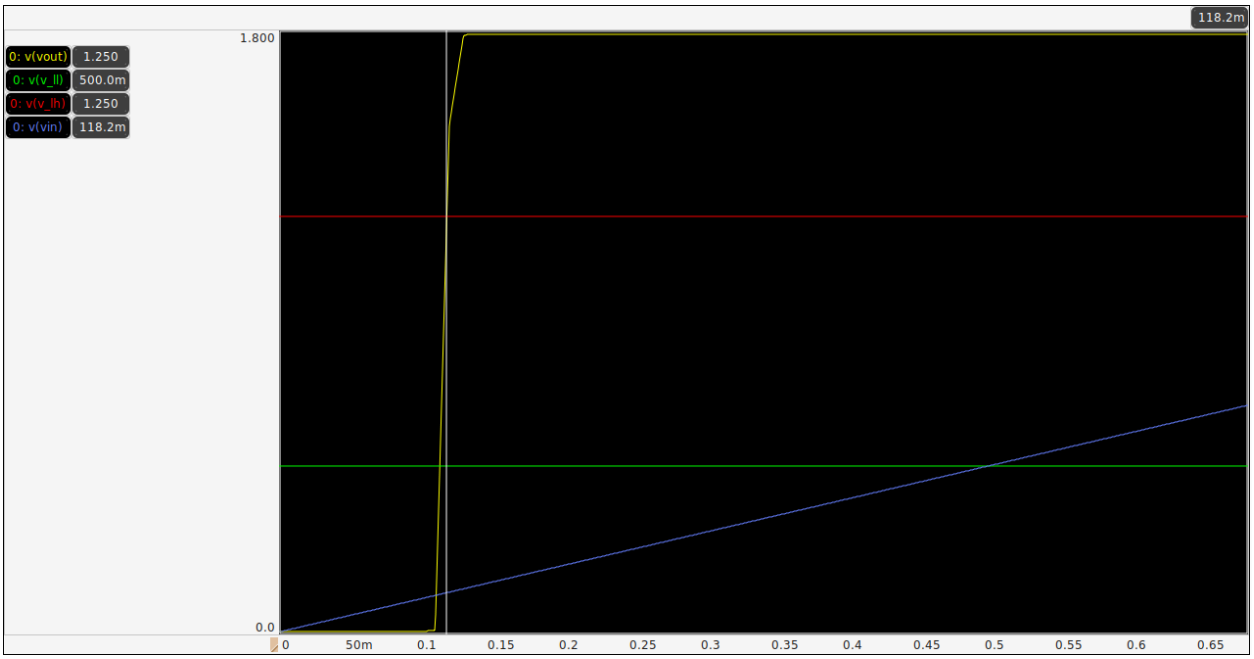
Th. 02 (0.2156)


Vo
In02
Vin
Topmost cell in the window
Vout
Vin

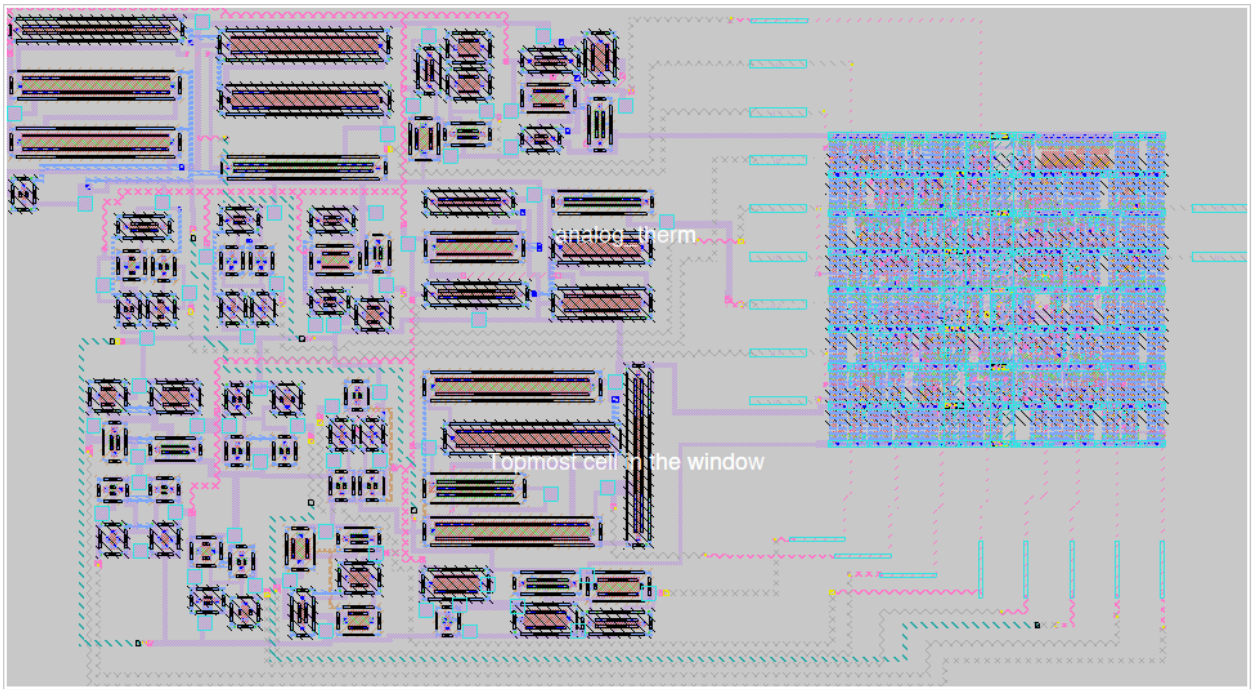

0: v(vout) 1.250
0: v(v_ll) 500.0m
0: v(v_lh) 1.250
0: v(vin) 215.6m
215.6m
1.800
0.0
0
50m
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55

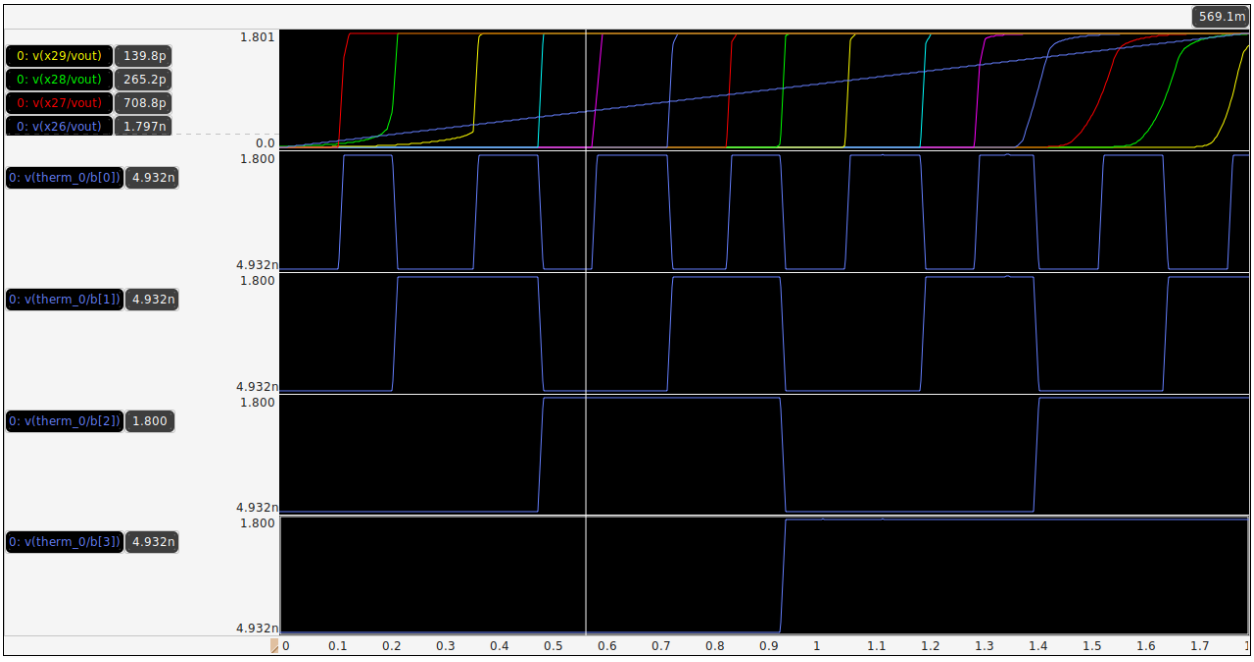
Th. 01 (0.1182)





Combined Layout:





Run iverilo/icarus

iverilog file.v

vvp a.out