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.

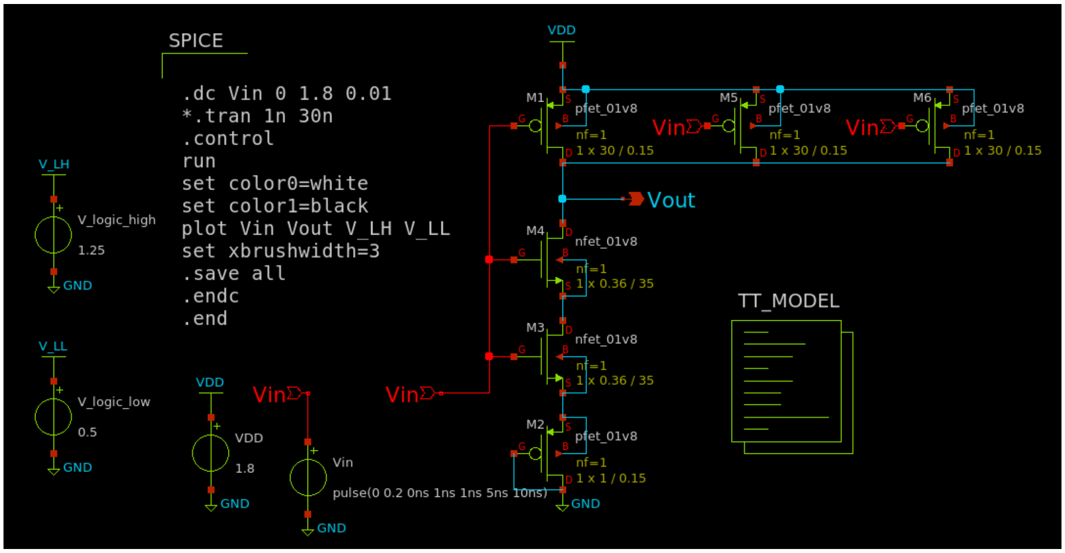
Steps:

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

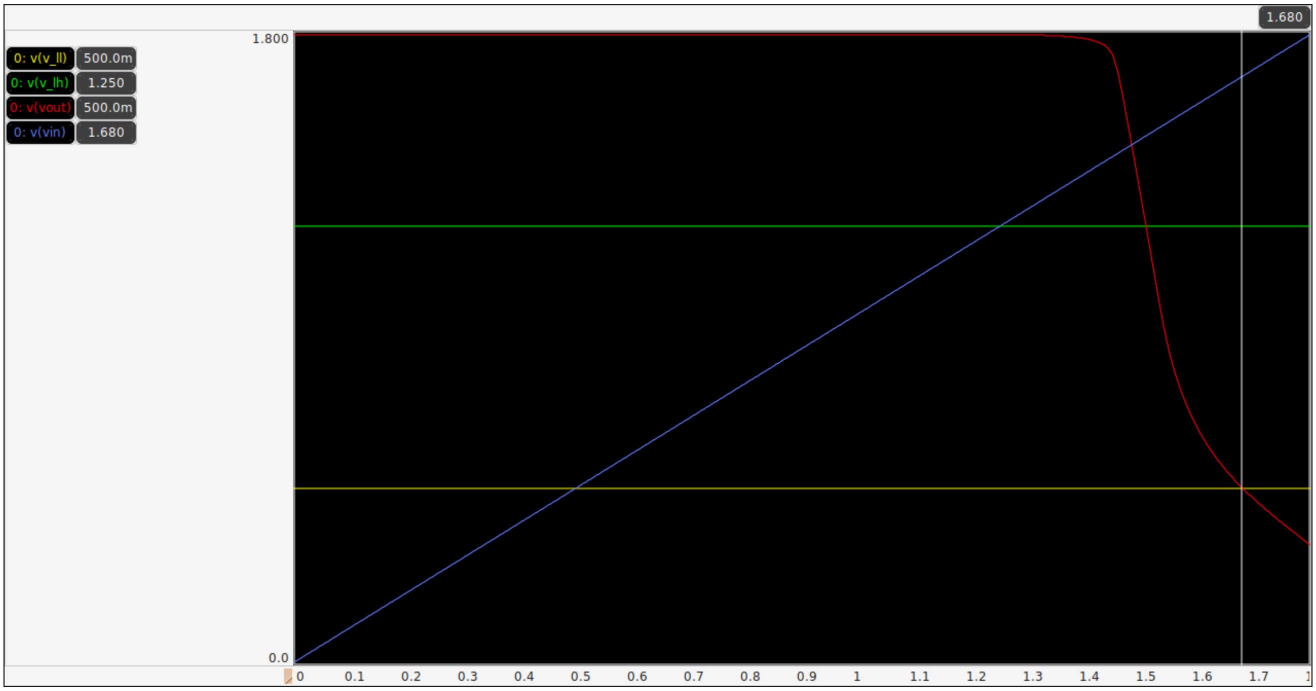
Technology	L Voltage	H Voltage	note
CMOS	0 to 30% of Vdd	70% of Vdd to Vdd	Vdd = Supply voltage
	≈0.5 Volt	≈1.25 Volts	Vdd = 1.8 Volts



SPICE

```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```

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



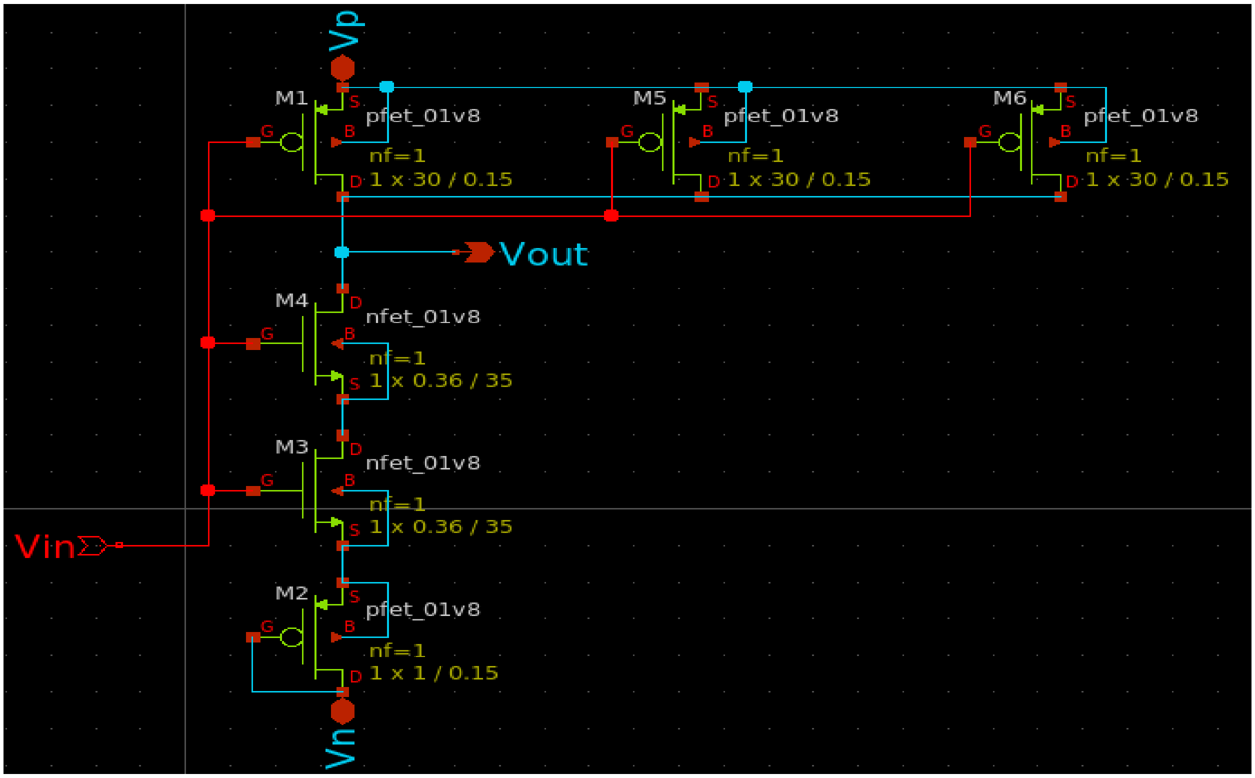
The inverter will switch from logic-high or undefined to logic-zero when input voltage exceeds 1.68 volts. So, for this inverter the threshold is 1.68 volts.

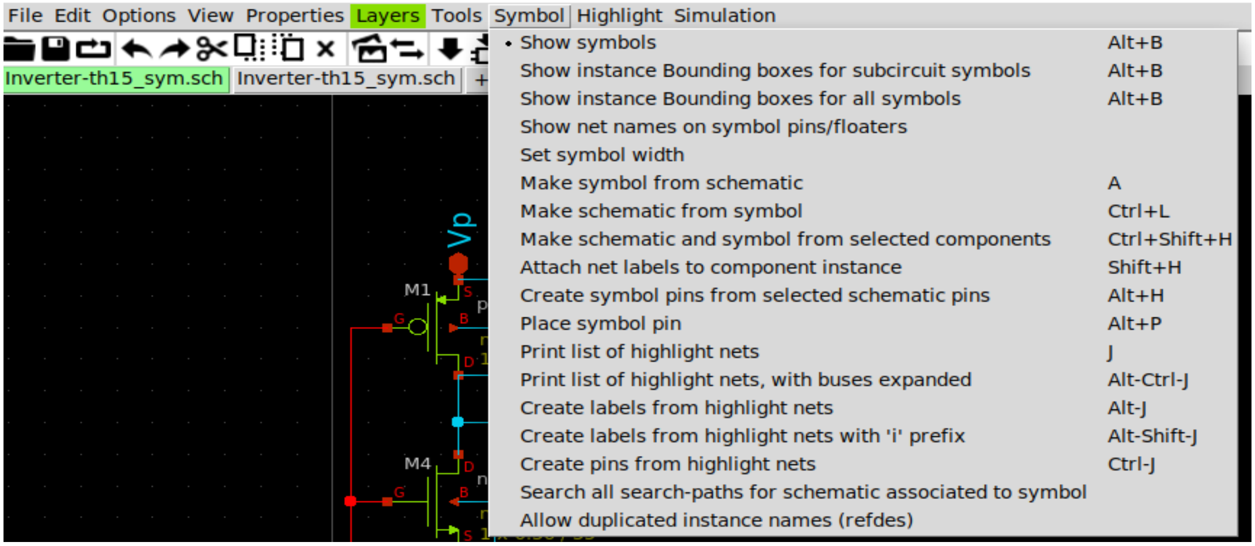
For a 4-bit ADC we need 16 voltage level including zero (0~1.68 volts). The quantization should be 1.68/16=0.112 volts.

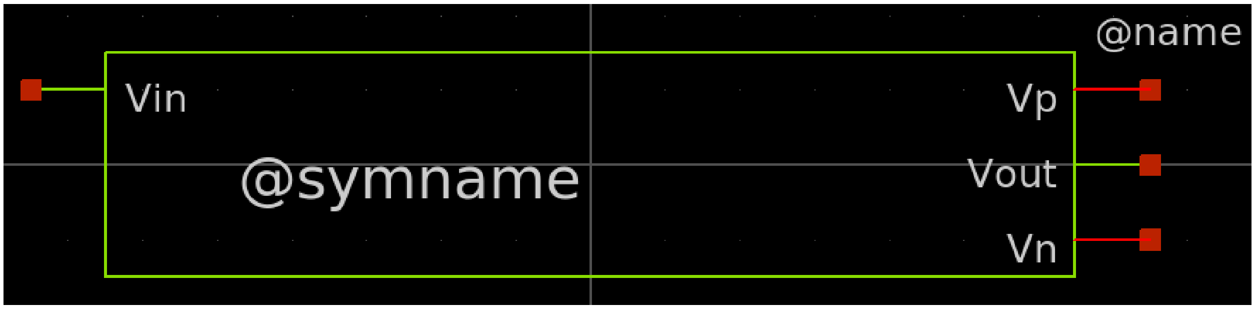
Vdd = 1.8 V

Th. 15: 1.680

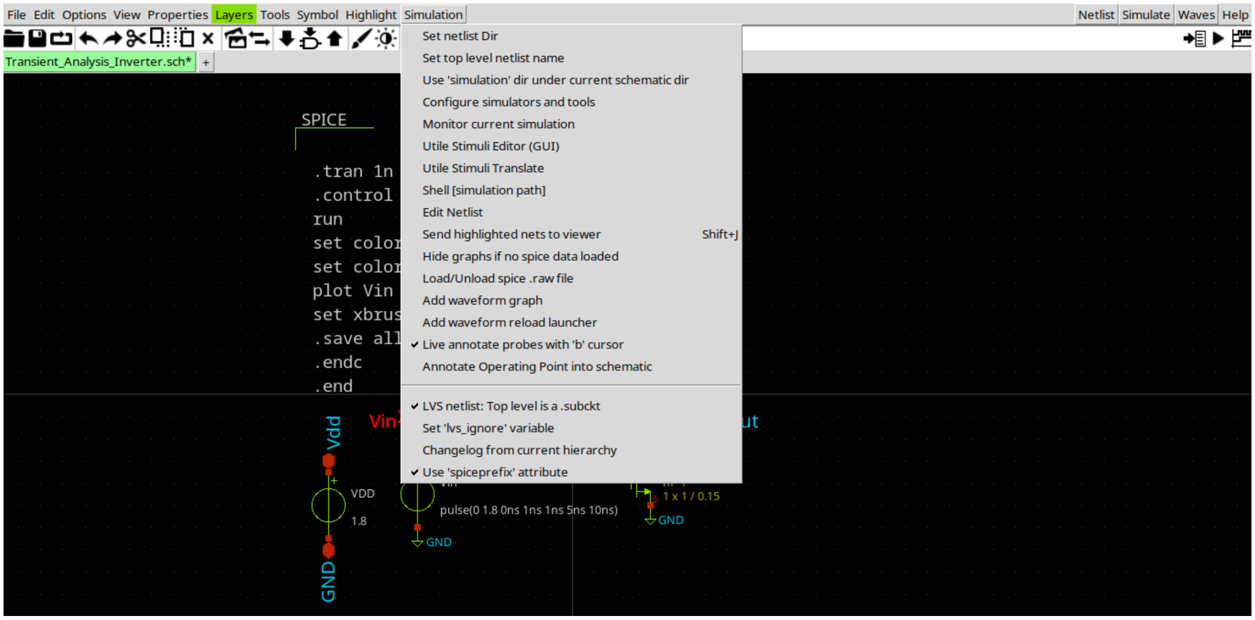
For calculation the schematic and the plot already mentioned earlier. Remove all source to make symbol from schematic. As the symbol will have to be connected to Vdd and GND, those pins are replaced with Vp and Vn accordingly.




File Edit Options View Properties Layers Tools Symbol Highlight Simulation
Inverter-th15_sym.sch
Inverter-th15_sym.sch
• Show symbols Alt+B
Show instance Bounding boxes for subcircuit symbols Alt+B
Show instance Bounding boxes for all symbols Alt+B
Show net names on symbol pins/floaters
Set symbol width
Make symbol from schematic A
Make schematic from symbol Ctrl+L
Make schematic and symbol from selected components Ctrl+Shift+H
Attach net labels to component instance Shift+H
Create symbol pins from selected schematic pins Alt+H
Place symbol pin Alt+P
Print list of highlight nets J
Print list of highlight nets, with buses expanded Alt-Ctrl-J
Create labels from highlight nets Alt-J
Create labels from highlight nets with 'i' prefix Alt-Shift-J
Create pins from highlight nets Ctrl-J
Search all search-paths for schematic associated to symbol
Allow duplicated instance names (refdes)
Vp
M1
M4

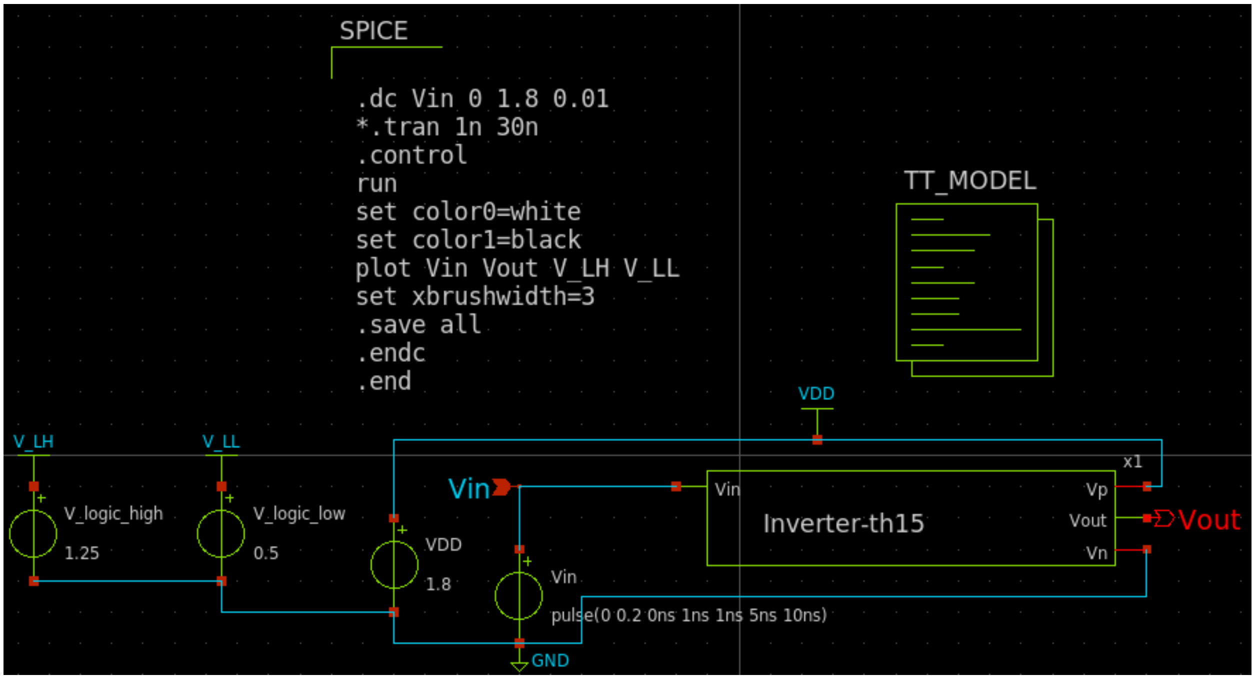


Magic=>before importing netlist to ngspice check LVS net


File Edit Options View Properties Layers Tools Symbol Highlight Simulation
Netlist Simulate Waves Help
Transient_Analysis_Inverter.sch*
Set netlist Dir
Set top level netlist name
Use 'simulation' dir under current schematic dir
Configure simulators and tools
Monitor current simulation
Utile Stimuli Editor (GUI)
Utile Stimuli Translate
Shell [simulation path]
Edit Netlist
Send highlighted nets to viewer Shift+J
Hide graphs if no spice data loaded
Load/Unload spice .raw file
Add waveform graph
Add waveform reload launcher
✓ Live annotate probes with 'b' cursor
Annotate Operating Point into schematic
✓ LVS netlist: Top level is a .subckt
Set 'lvs_ignore' variable
Changelog from current hierarchy
✓ Use 'spiceprefix' attribute
SPICE
Vdd
Vin
ut
VDD
1.8
GND
pulse(0 1.8 0ns 1ns 1ns 5ns 10ns)
GND
1 x 1 / 0.15
GND

```
.tran 1n
.control
run
set color
set color
plot Vin
set xbrus
.save all
.endc
.end
```

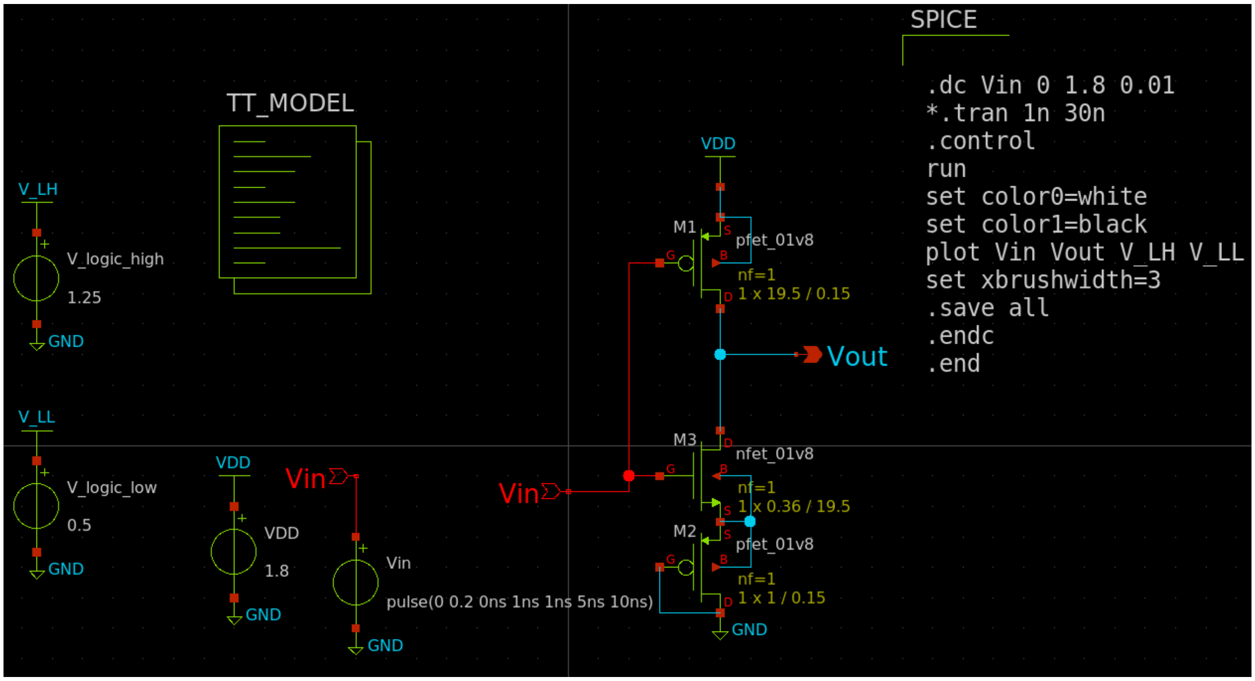
The simplified schematic is as follows:



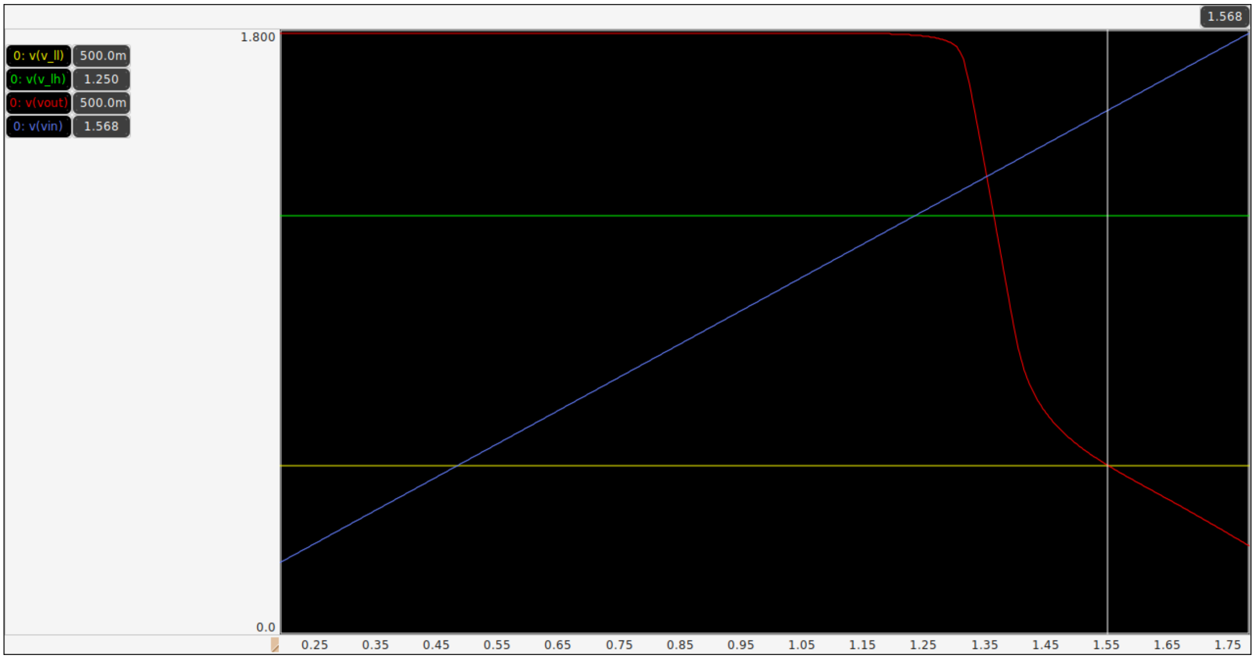
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```

The plot is similar to the previous figure.

Th. 14: **1.568**

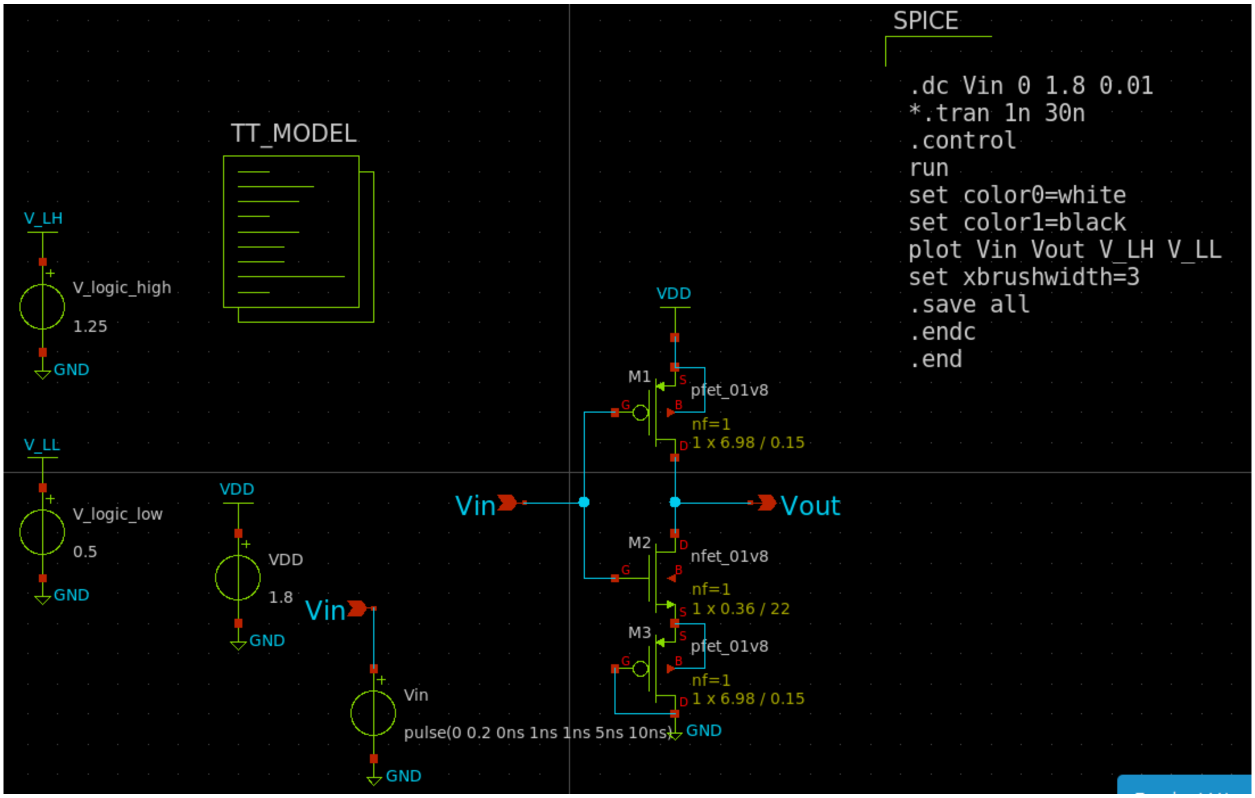


```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```

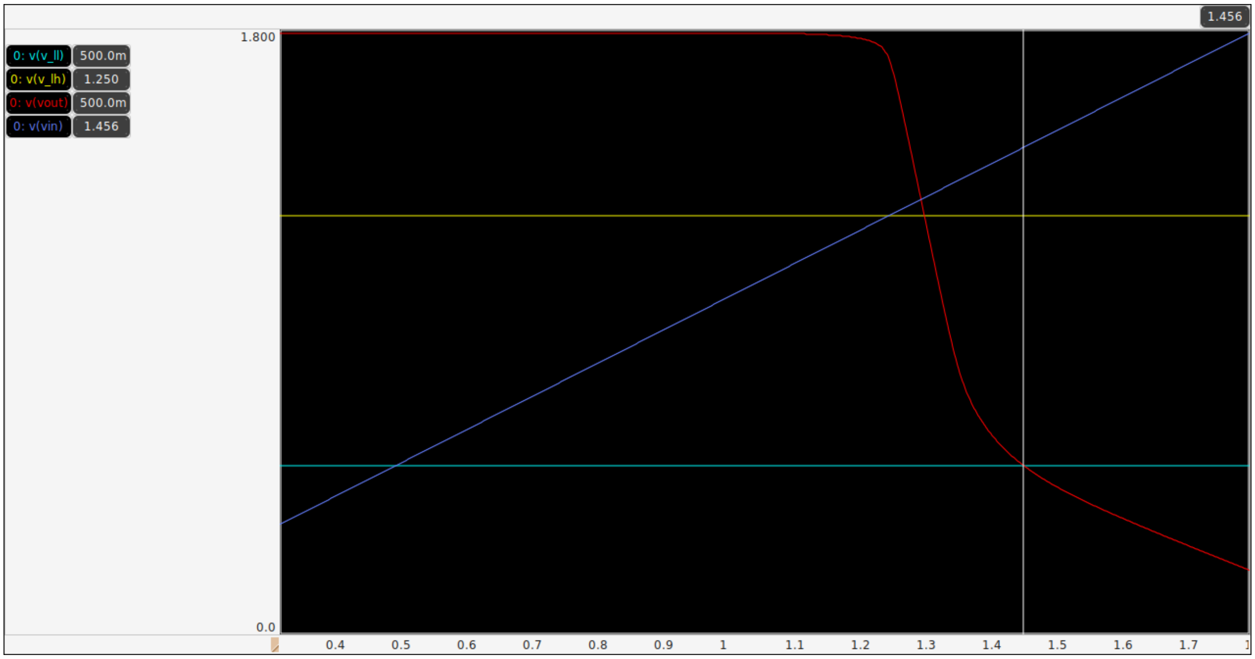


Rest of the figures are similar to the previous figures.

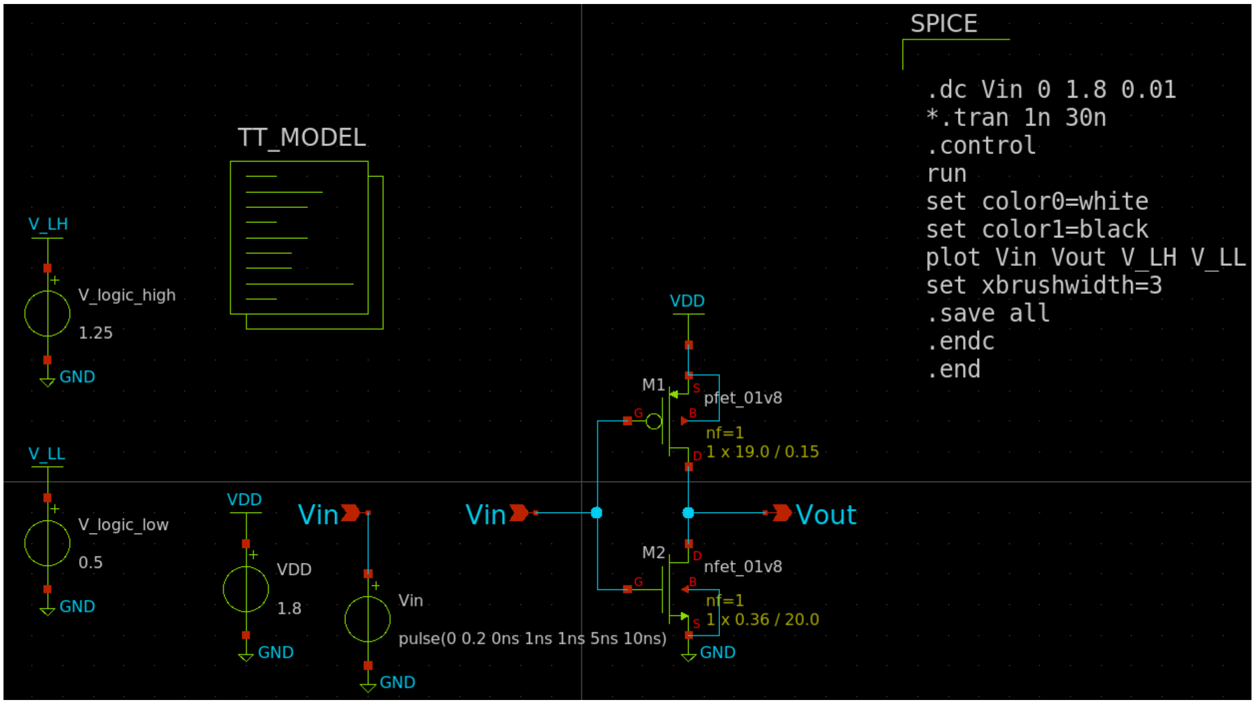
Th. 13: 1.456:



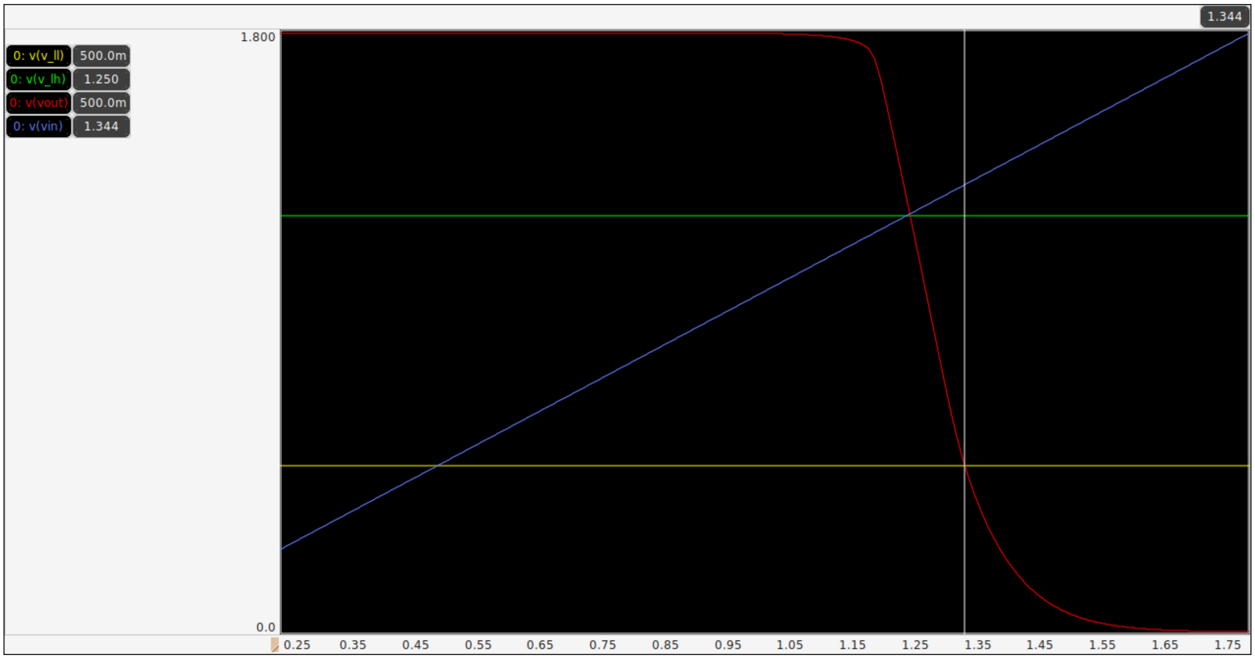
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



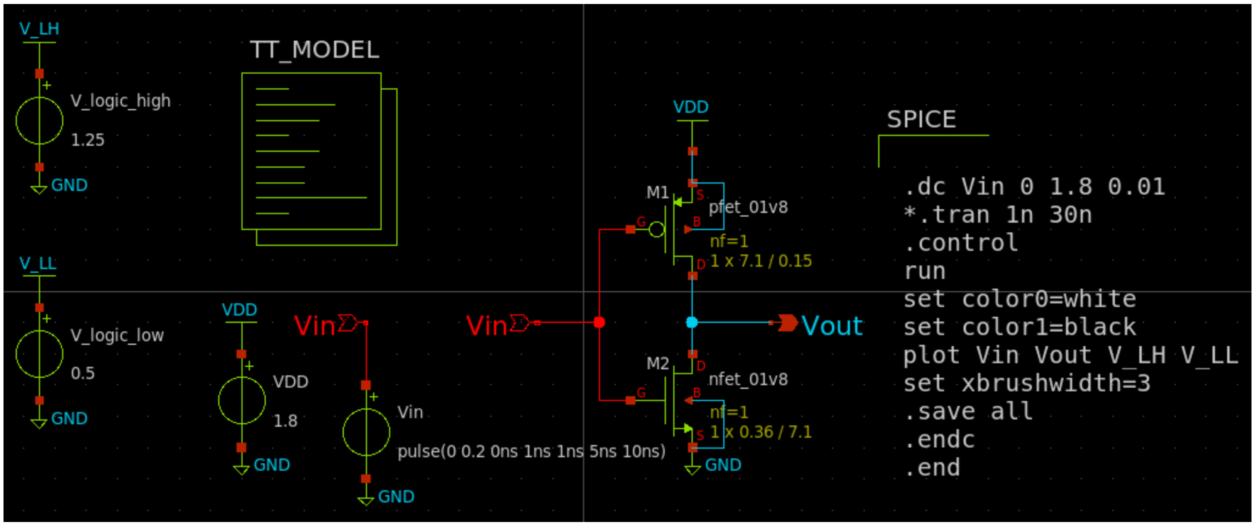
Th. 12: **1.344**



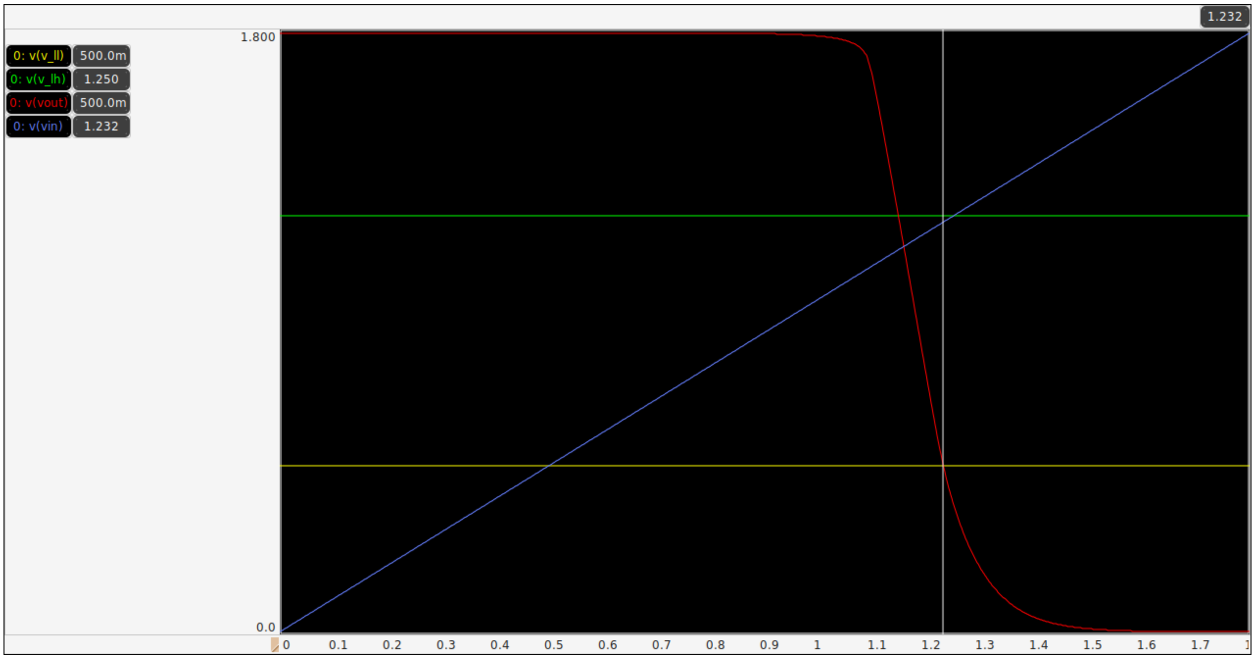
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



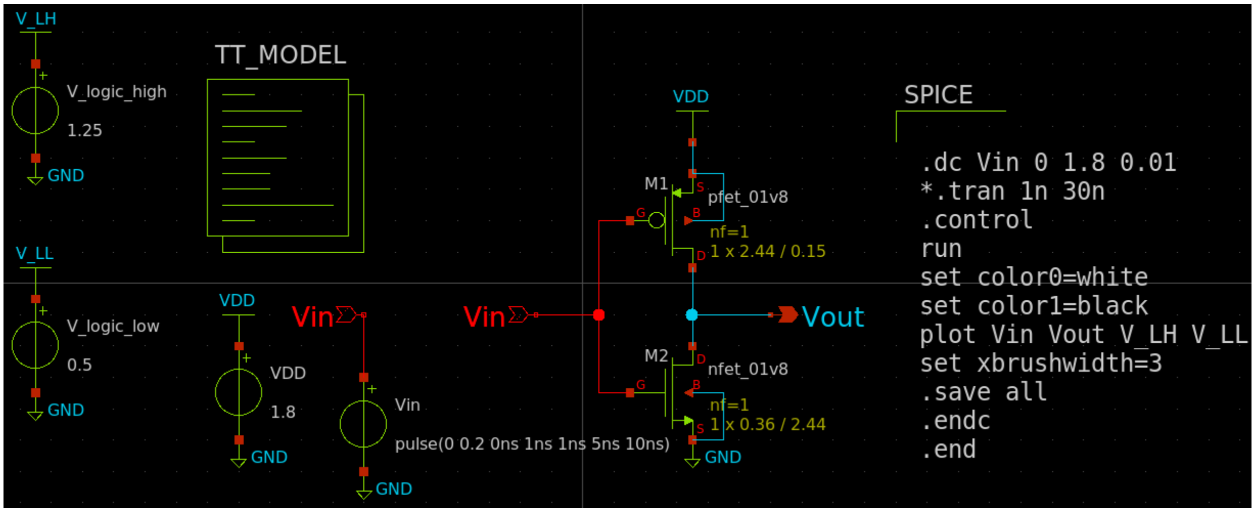
Th. 11: **1.232**



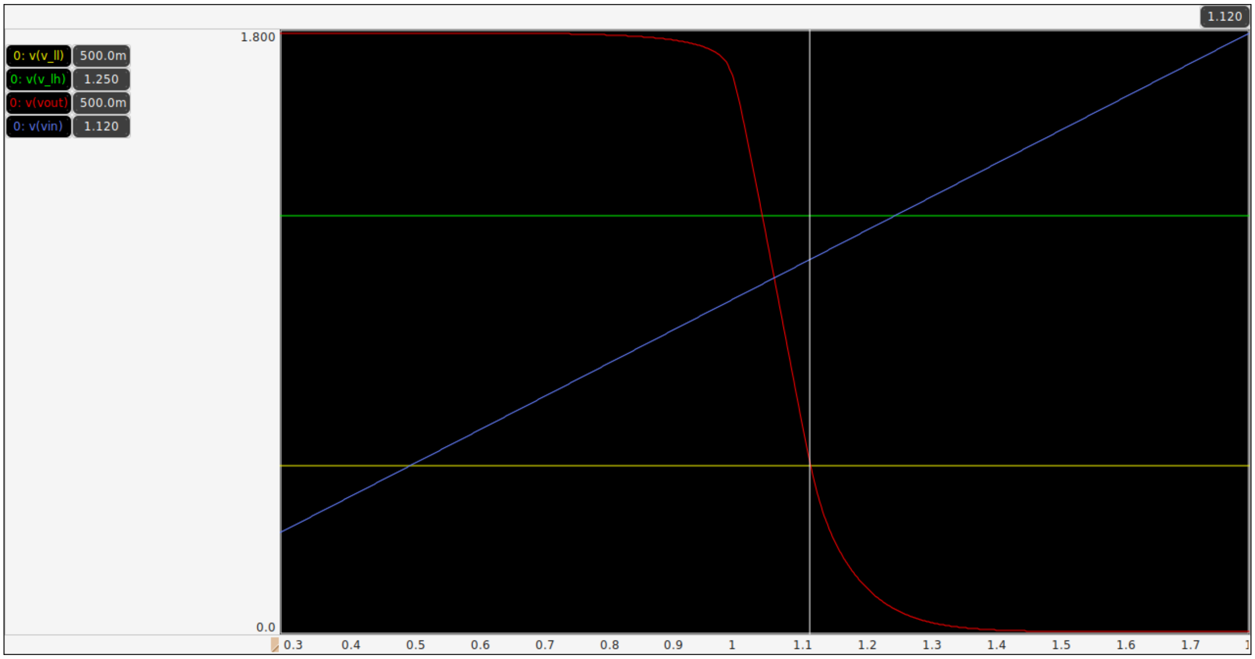
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



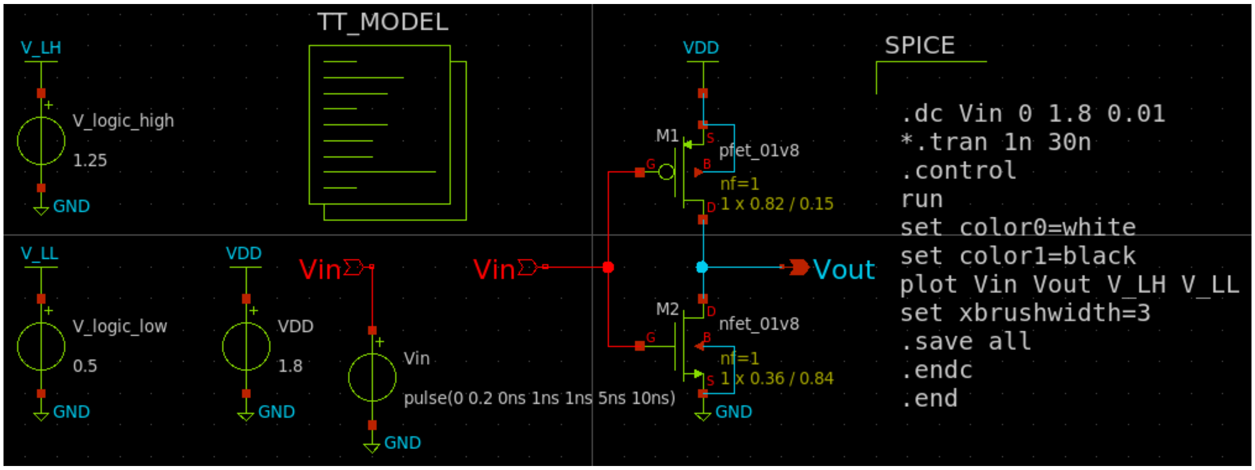
Th. 10: **1.12**



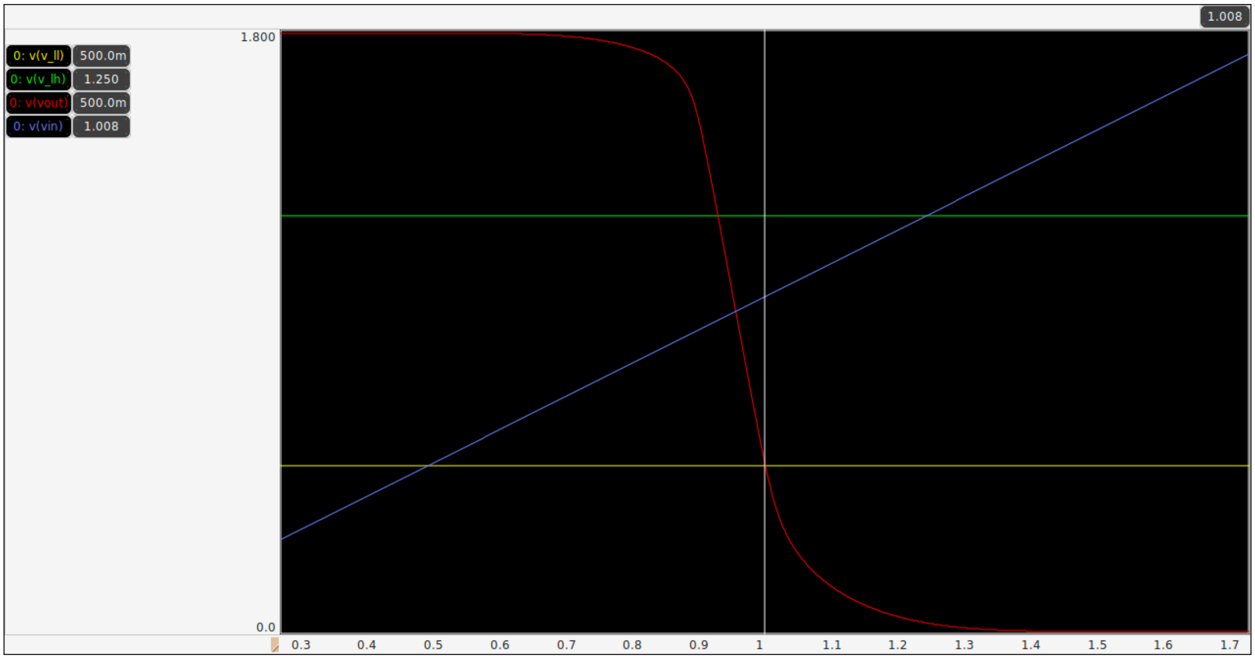
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



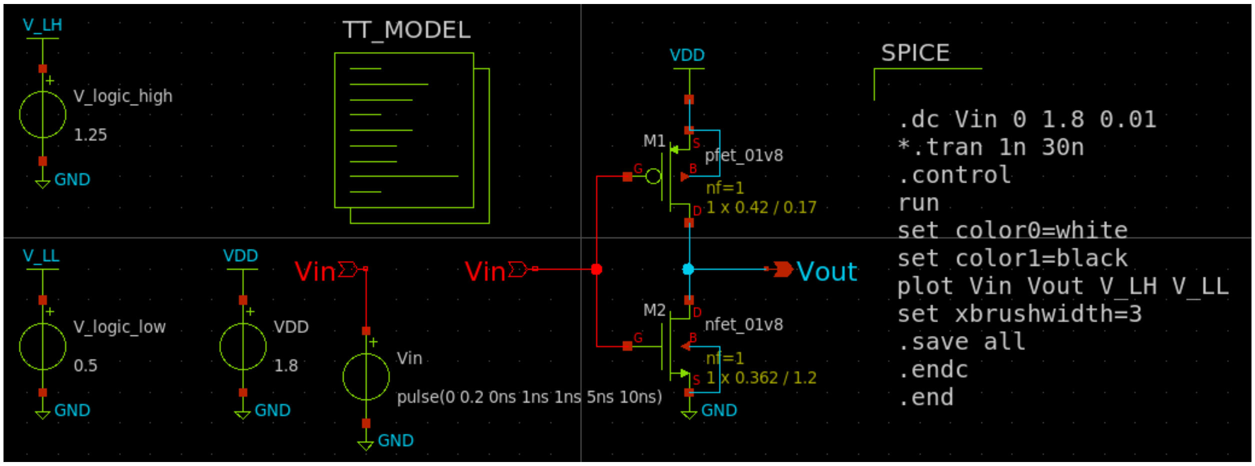
Th. 9: 1.008



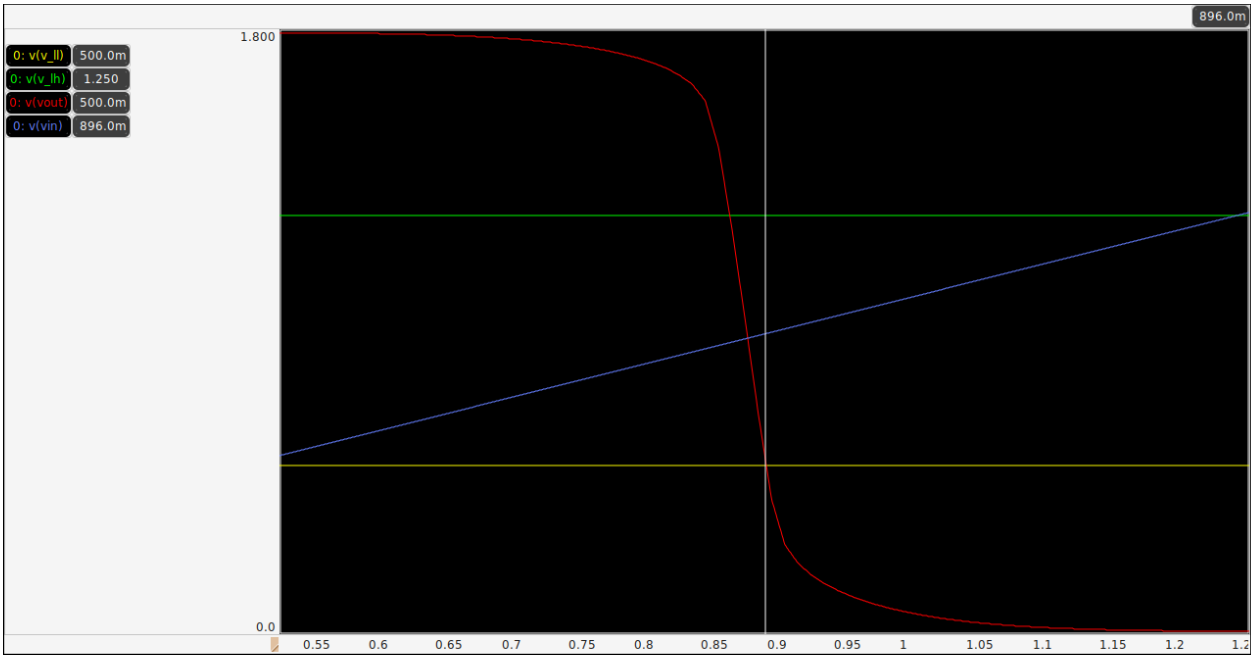
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



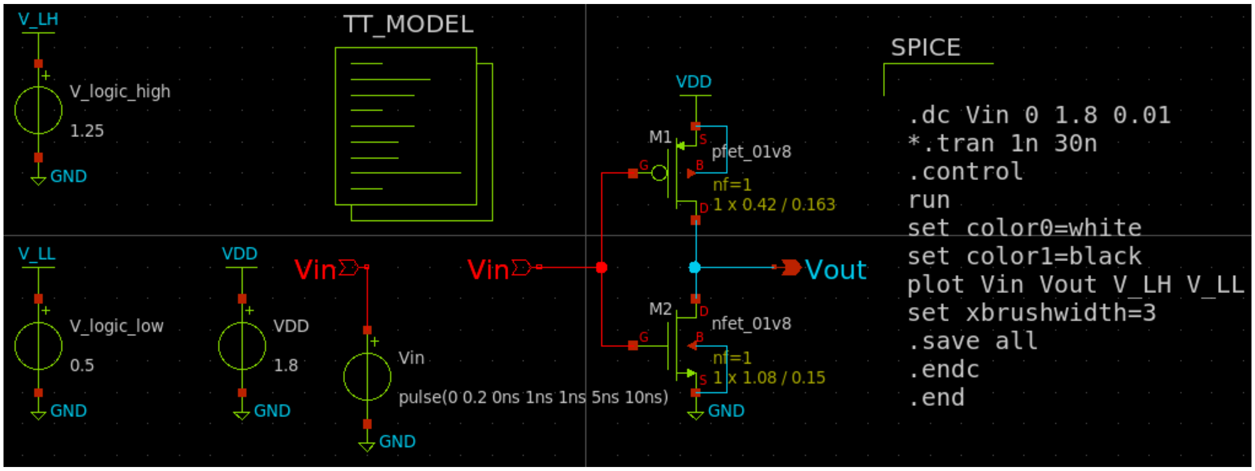
Th. 8: 0.896



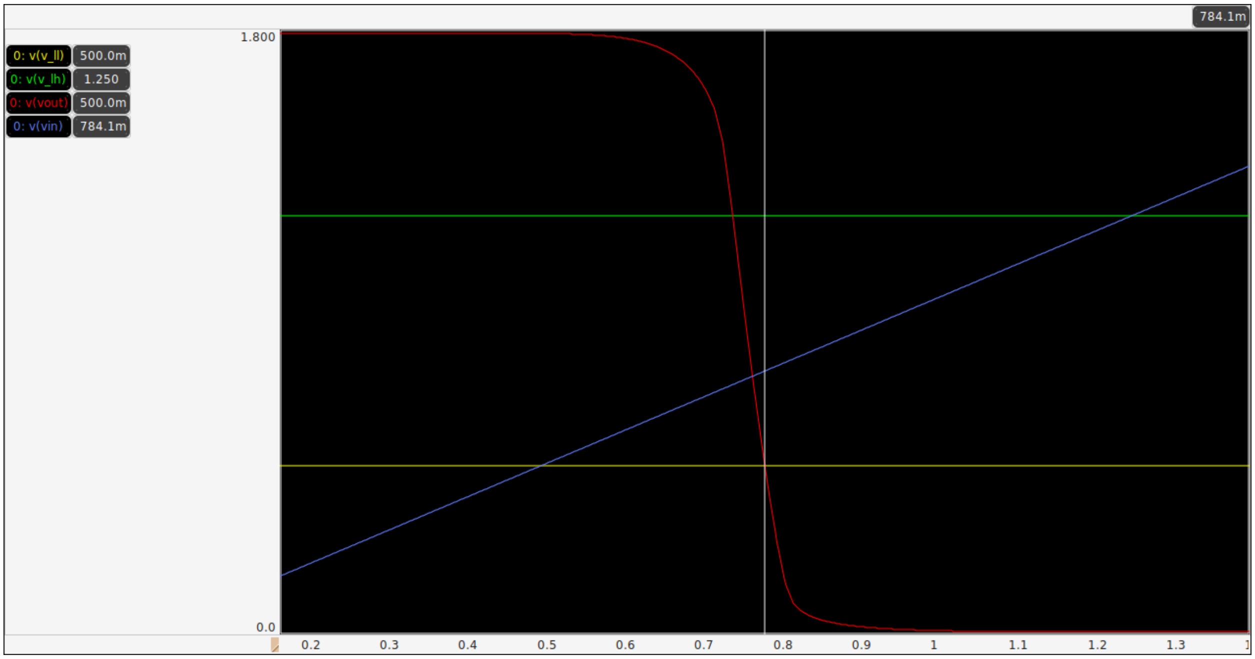
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



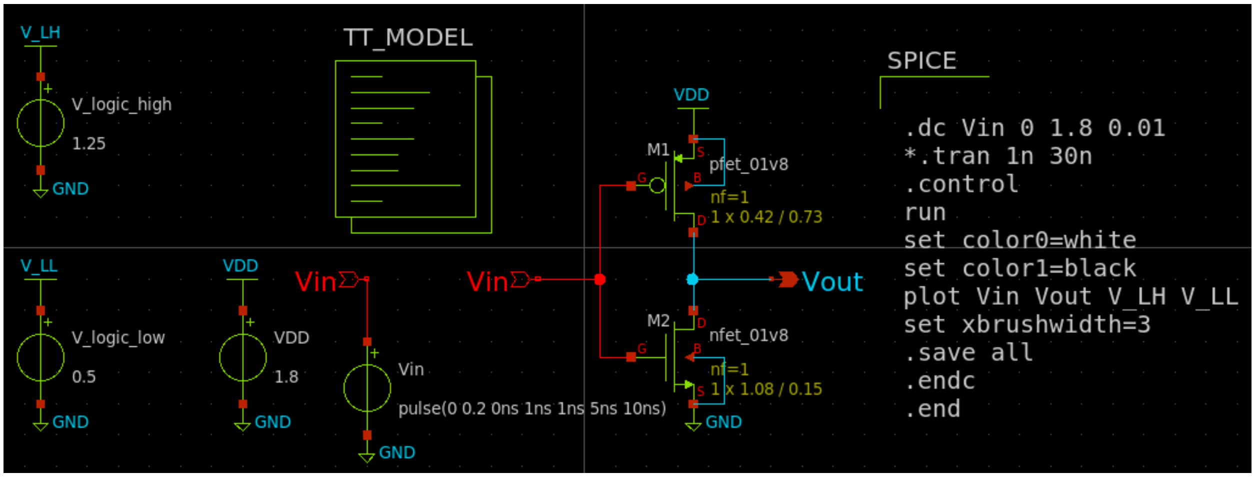
Th. 7: 0.784



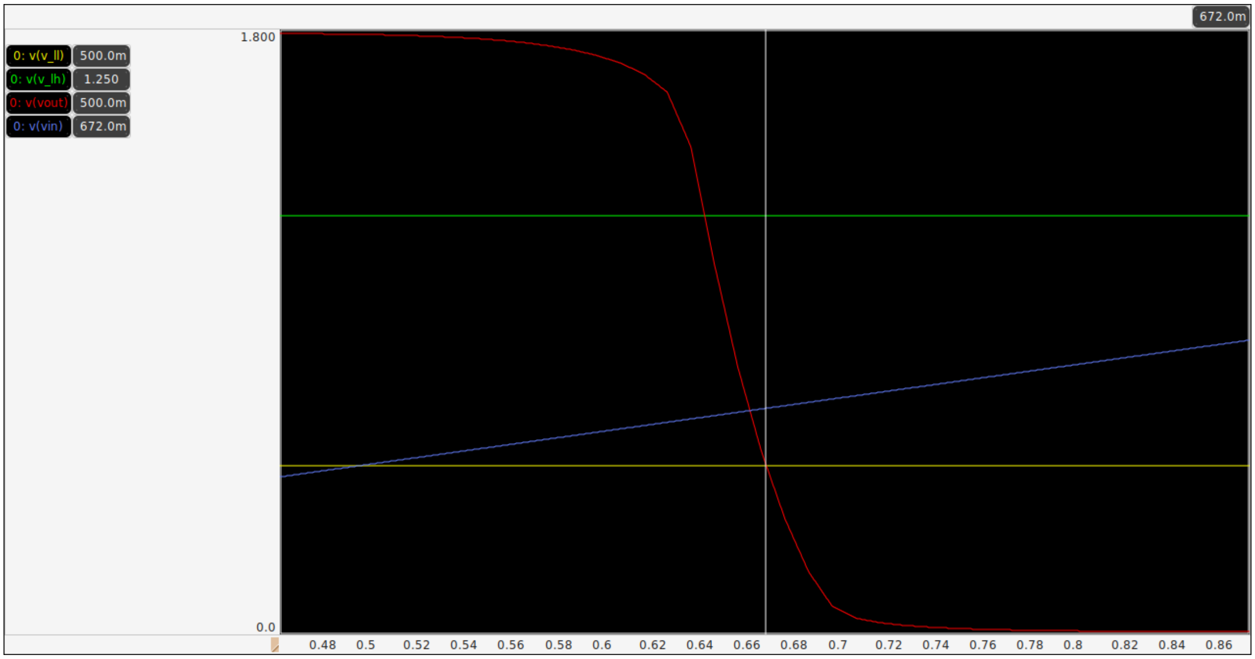
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



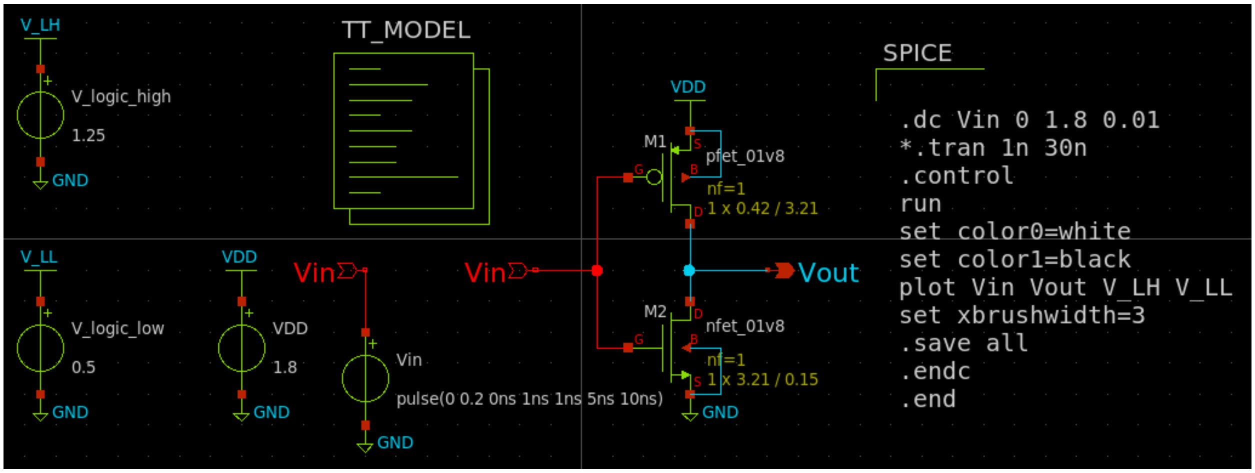
Th. 6: 0.672



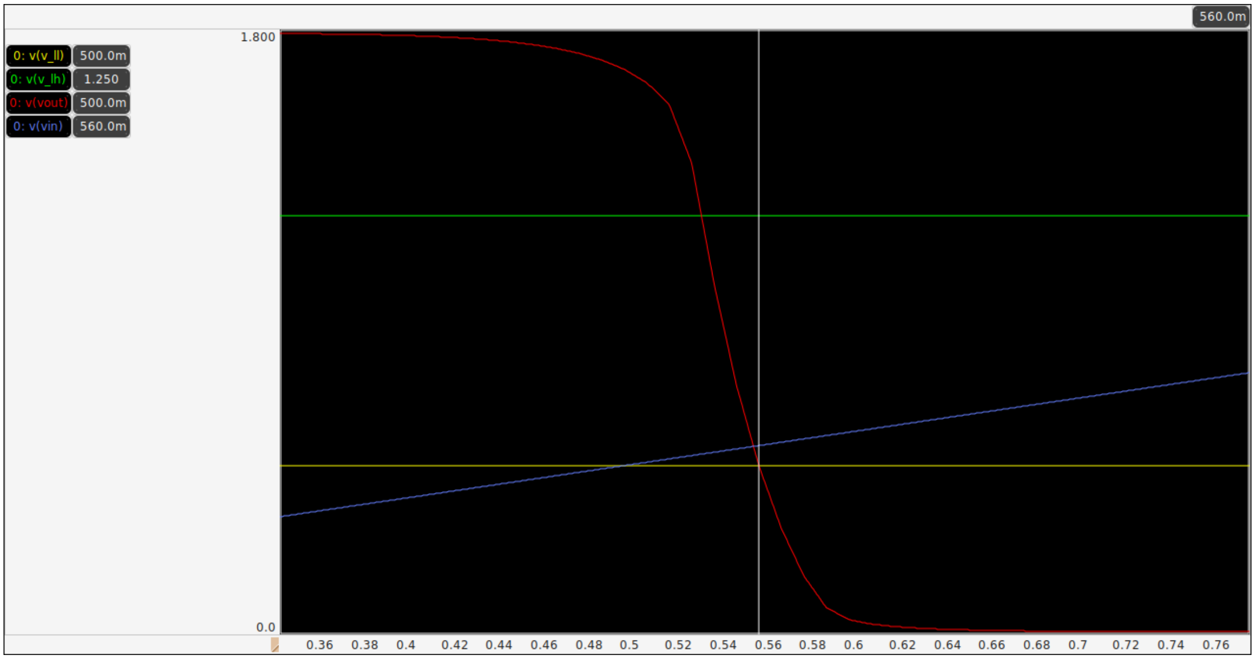
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



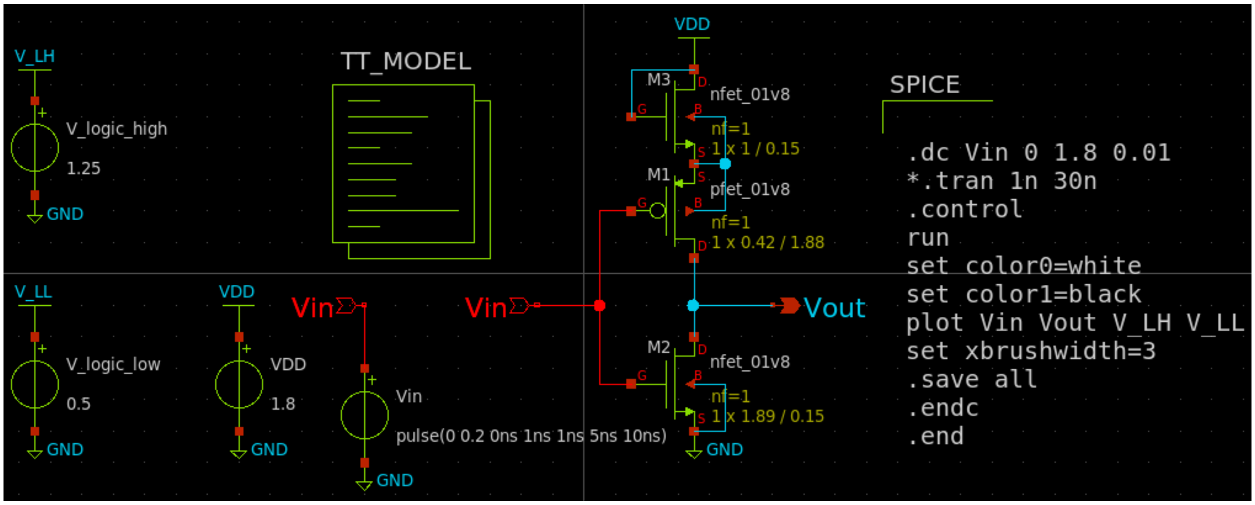
Th. 5: 0.56



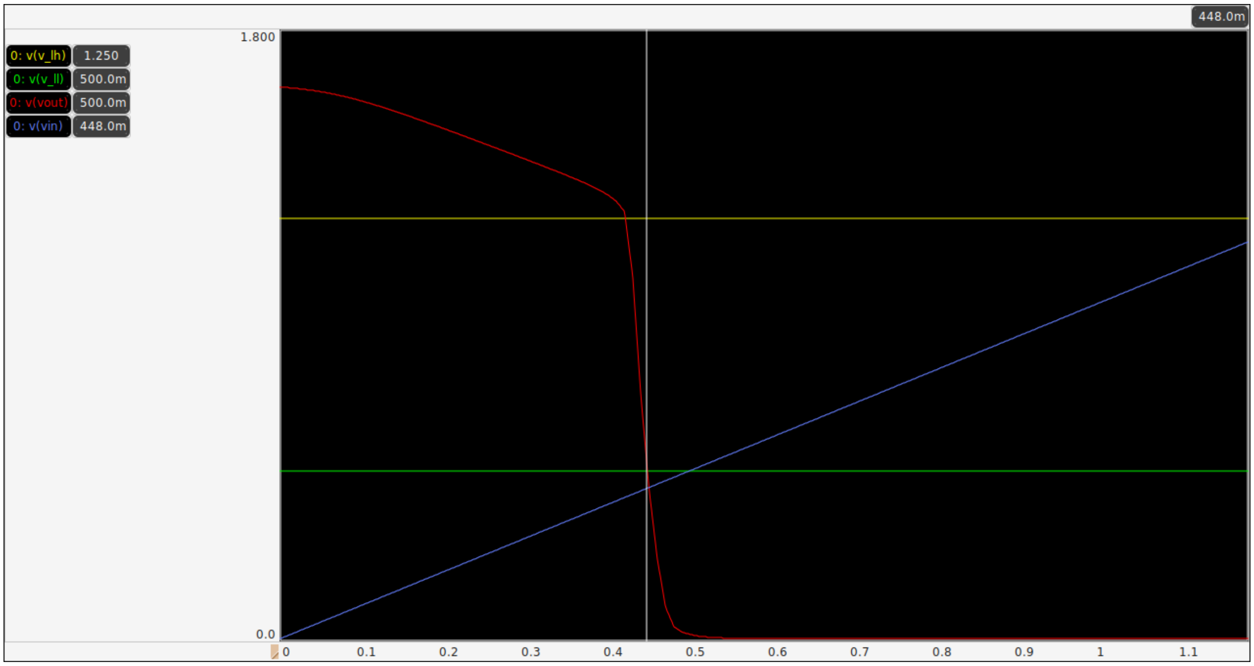
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



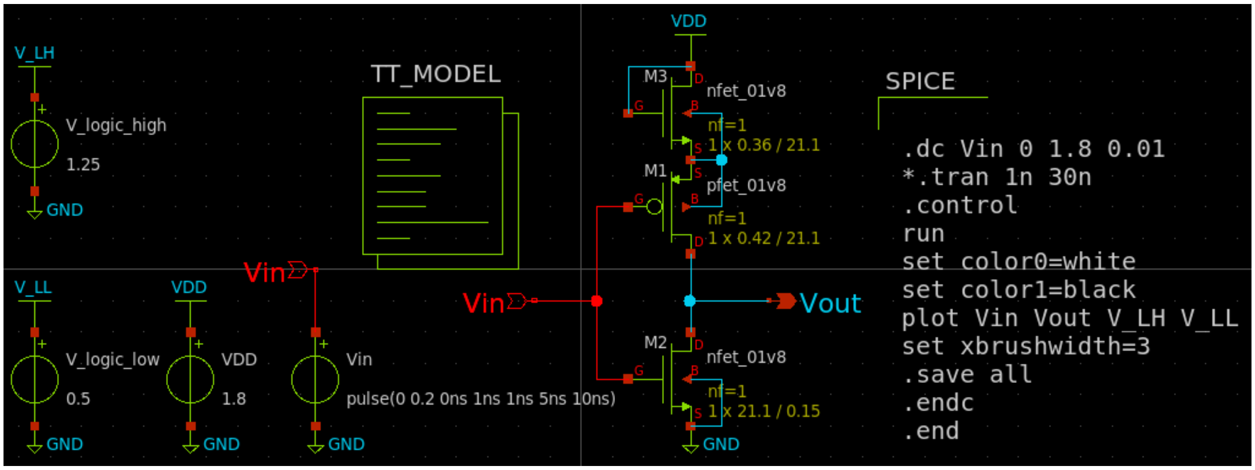
Th. 4: 0.448



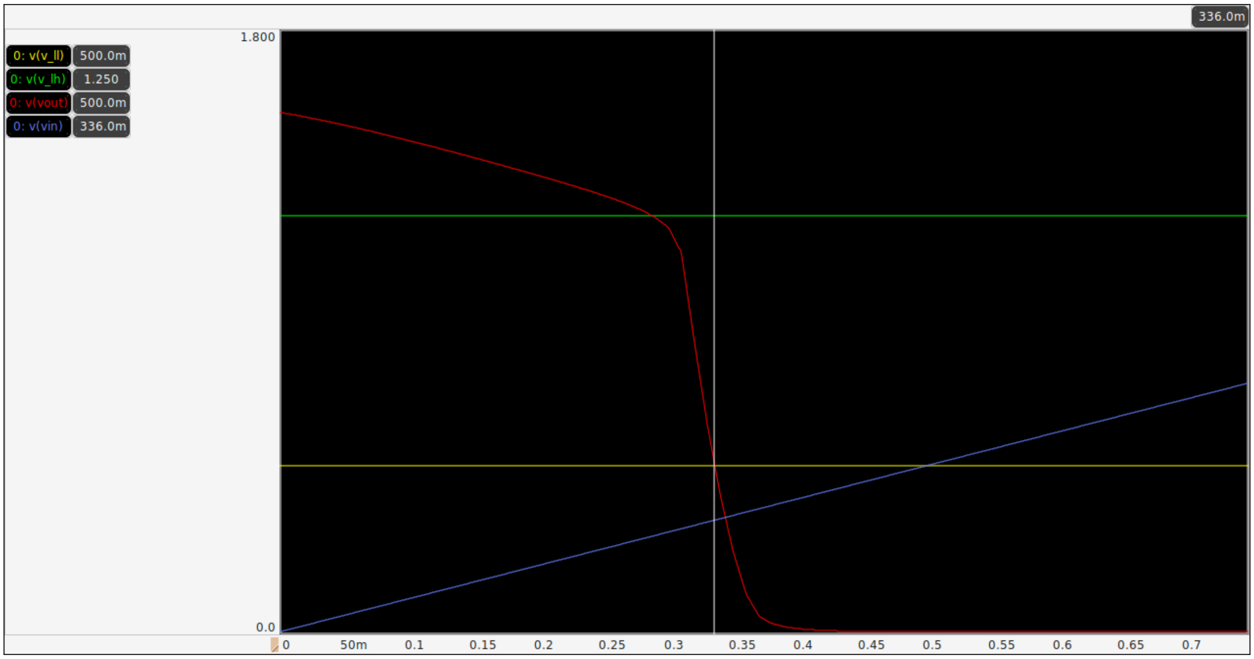
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



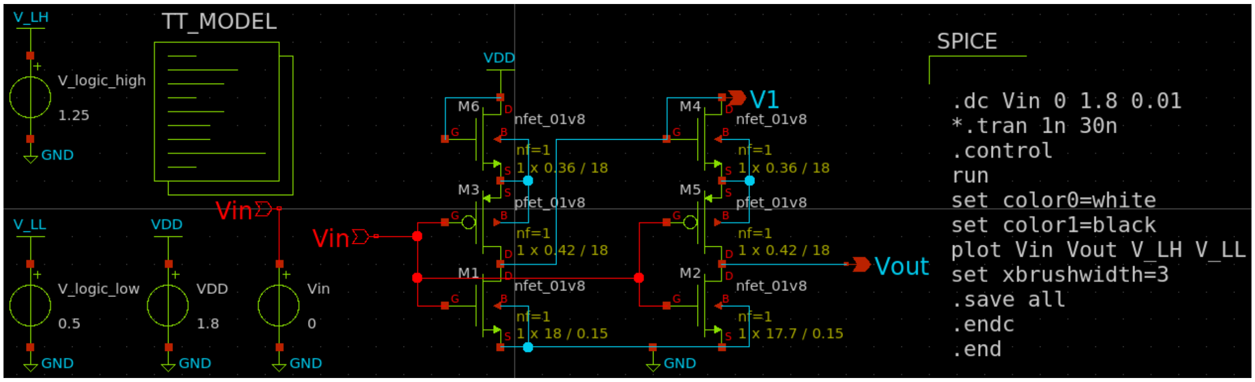
Th. 3: 0.336



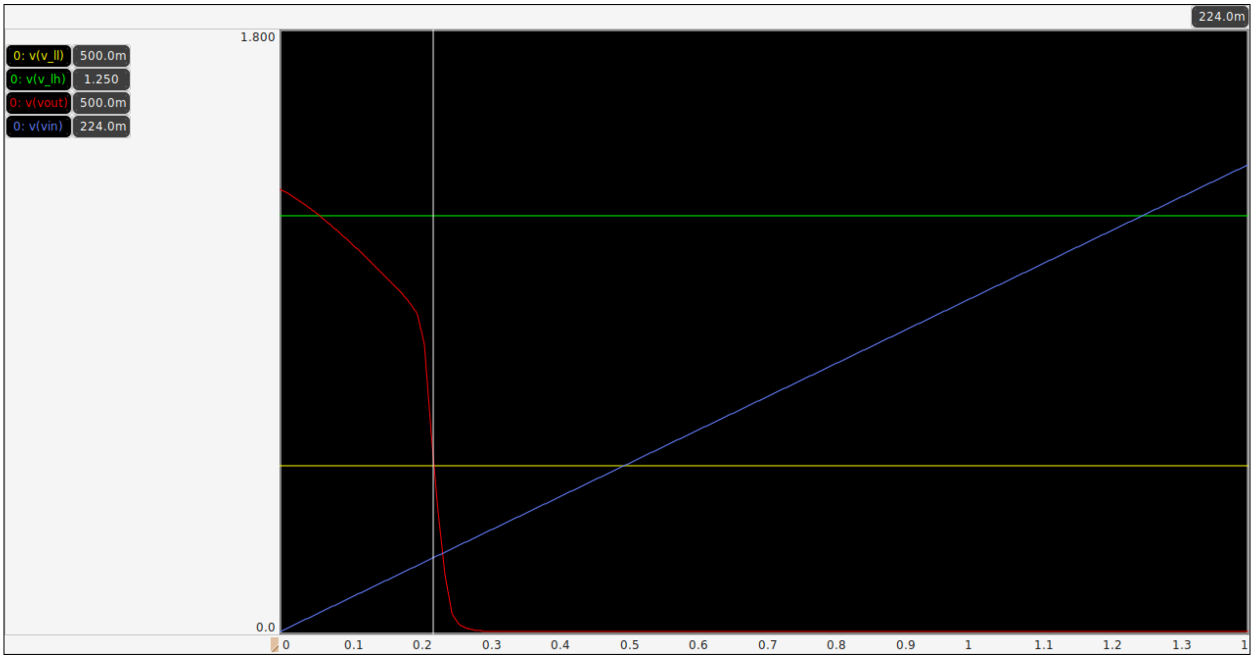
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



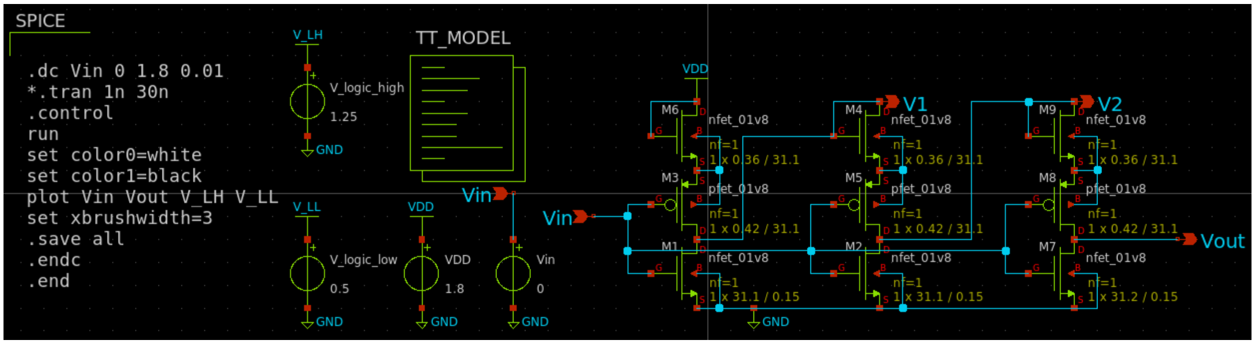
Th. 2: 0.224



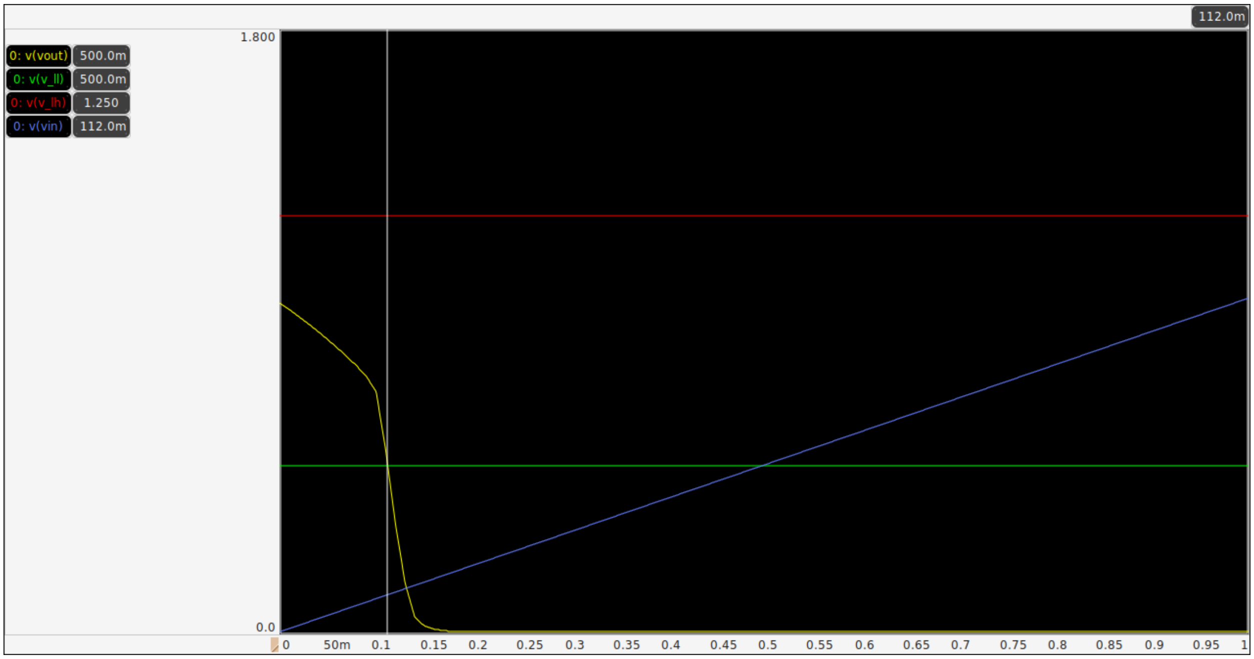
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



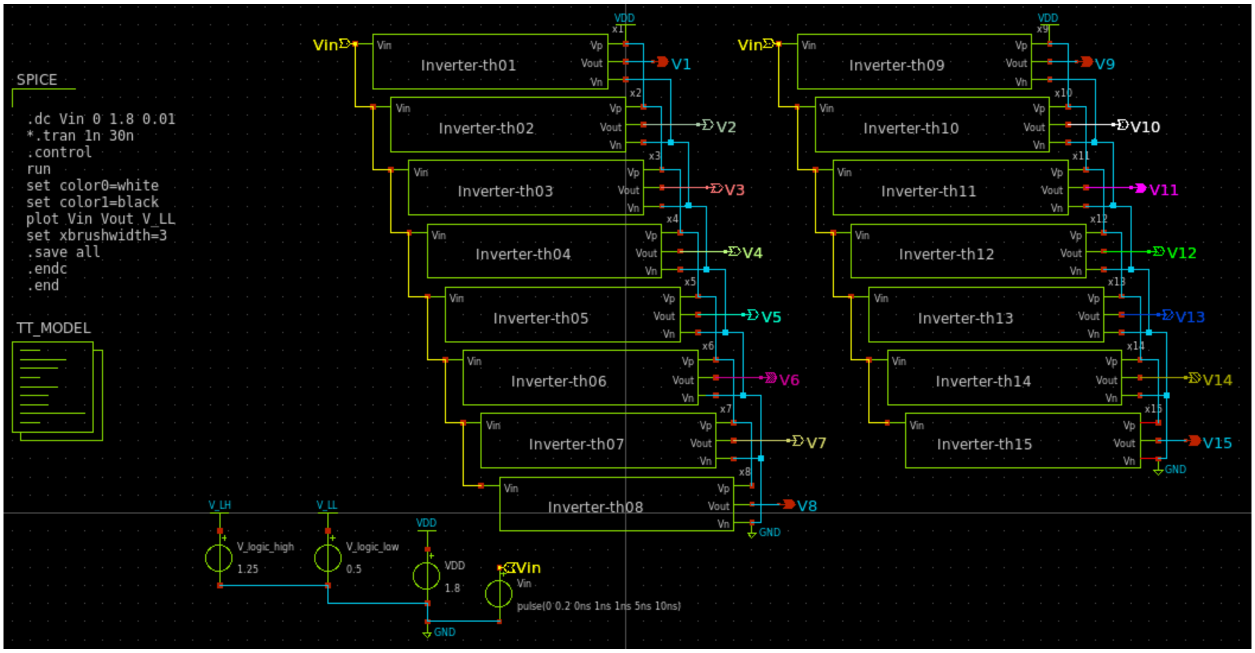
Th. 1: 0.112



```
SPICE
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```

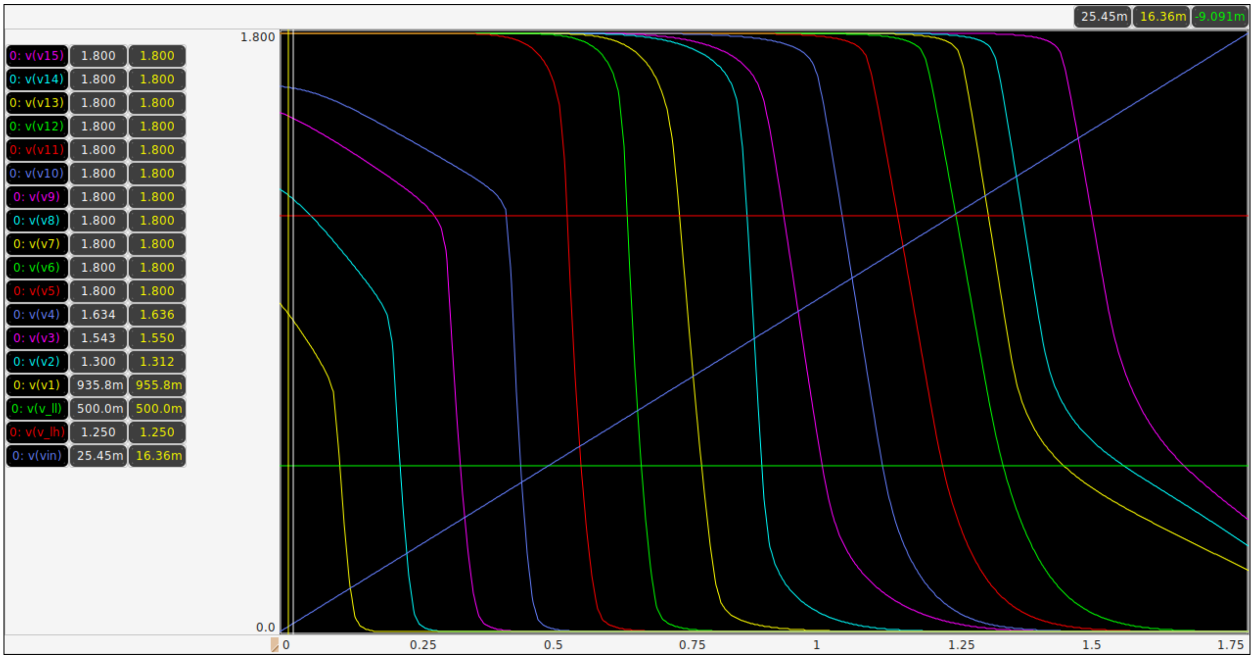


Combined Analog sch.



SPICE

```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LL
set xbrushwidth=3
.save all
.endc
.end
```



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

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://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/

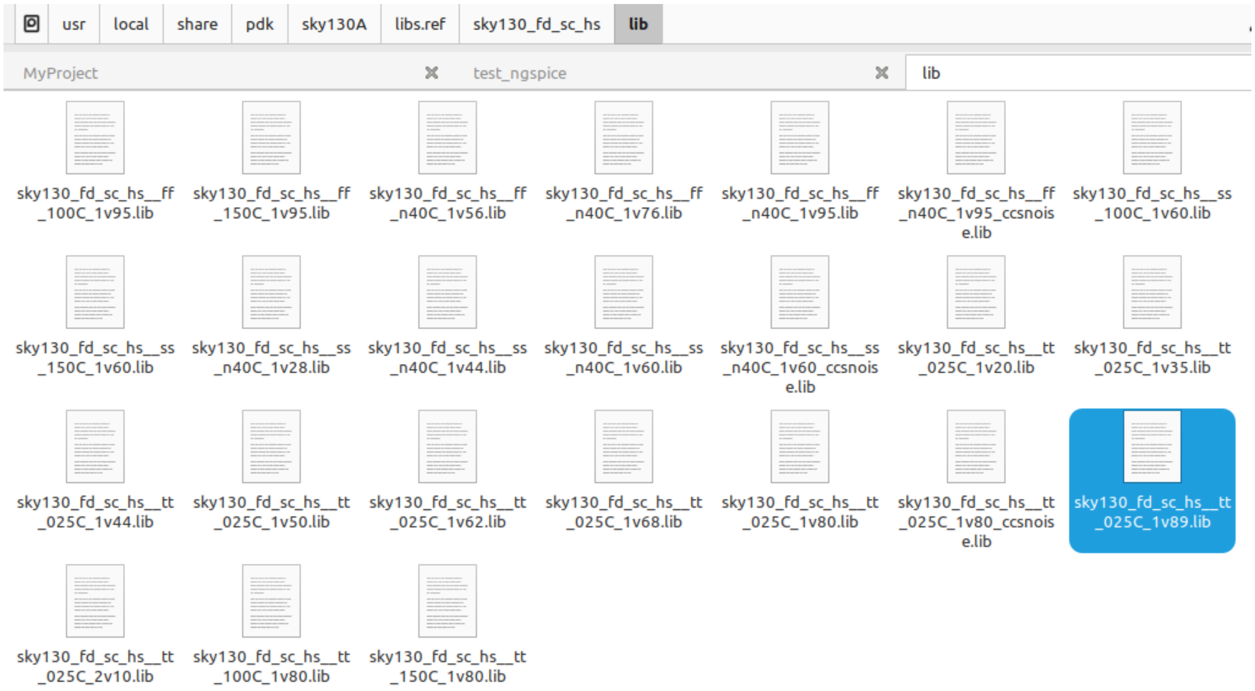
https://github.com/spdy1895/RTL_synthesis_using_sky130

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

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

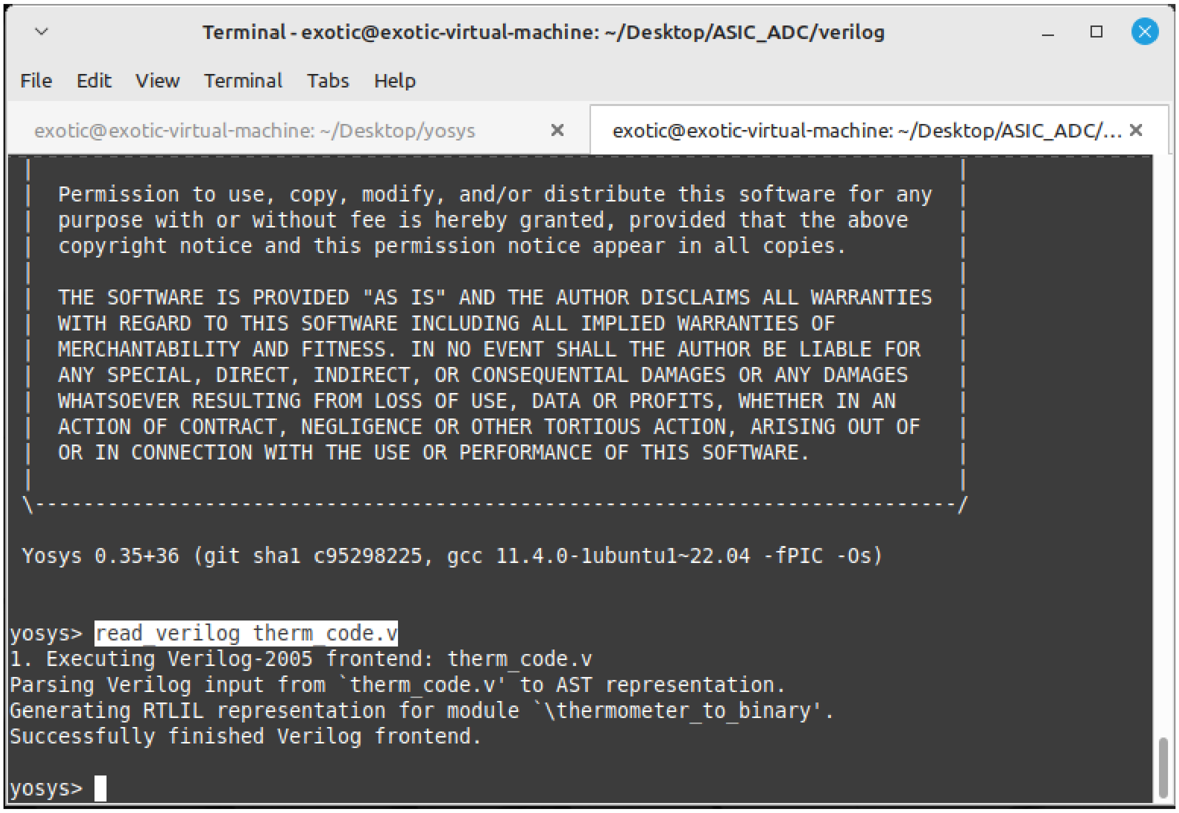
```
>> yosys            // open yosys console in terminal

>> read_liberty -lib
/usr/local/share/pdk/sky130A/libs.ref/sky130_fd_sc_hs/lib/sky130_fd_sc_hs__tt_025C_1v89.lib
```

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



sky130_fd_sc_hs__ff_100C_1v95.lib
sky130_fd_sc_hs__ff_150C_1v95.lib
sky130_fd_sc_hs__ff_n40C_1v56.lib
sky130_fd_sc_hs__ff_n40C_1v76.lib
sky130_fd_sc_hs__ff_n40C_1v95.lib
sky130_fd_sc_hs__ff_n40C_1v95_ccsnoise.lib
sky130_fd_sc_hs__ss_100C_1v60.lib
sky130_fd_sc_hs__ss_150C_1v60.lib
sky130_fd_sc_hs__ss_n40C_1v28.lib
sky130_fd_sc_hs__ss_n40C_1v44.lib
sky130_fd_sc_hs__ss_n40C_1v60.lib
sky130_fd_sc_hs__ss_n40C_1v60_ccsnoise.lib
sky130_fd_sc_hs__tt_025C_1v20.lib
sky130_fd_sc_hs__tt_025C_1v35.lib
sky130_fd_sc_hs__tt_025C_1v44.lib
sky130_fd_sc_hs__tt_025C_1v50.lib
sky130_fd_sc_hs__tt_025C_1v62.lib
sky130_fd_sc_hs__tt_025C_1v68.lib
sky130_fd_sc_hs__tt_025C_1v80.lib
sky130_fd_sc_hs__tt_025C_1v80_ccsnoise.lib
sky130_fd_sc_hs__tt_025C_1v89.lib
sky130_fd_sc_hs__tt_025C_2v10.lib
sky130_fd_sc_hs__tt_100C_1v80.lib
sky130_fd_sc_hs__tt_150C_1v80.lib

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

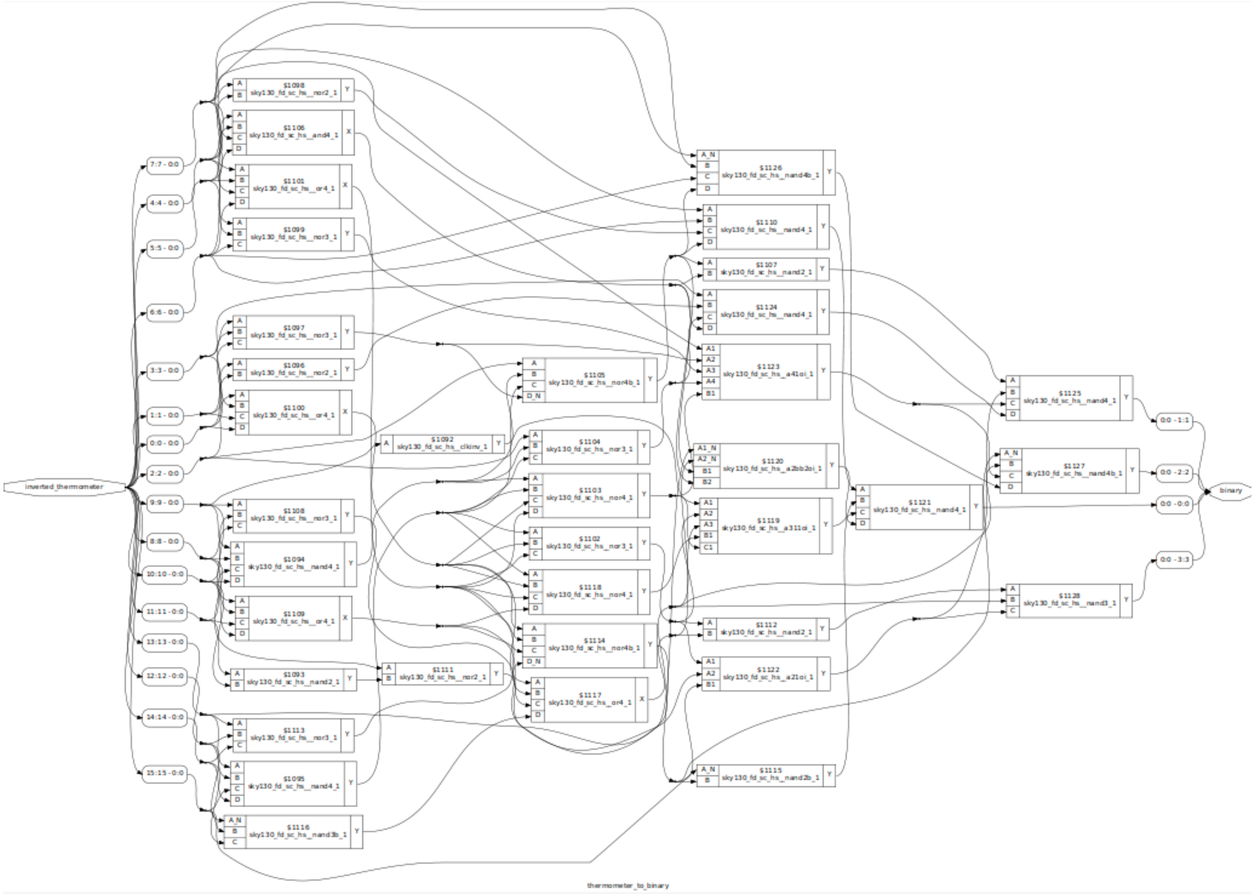

Terminal - exotic@exotic-virtual-machine: ~/Desktop/ASIC_ADC/verilog
File
Edit
View
Terminal
Tabs
Help
exotic@exotic-virtual-machine: ~/Desktop/yosys
exotic@exotic-virtual-machine: ~/Desktop/ASIC_ADC/...

```
 |                                                                            |
 |  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

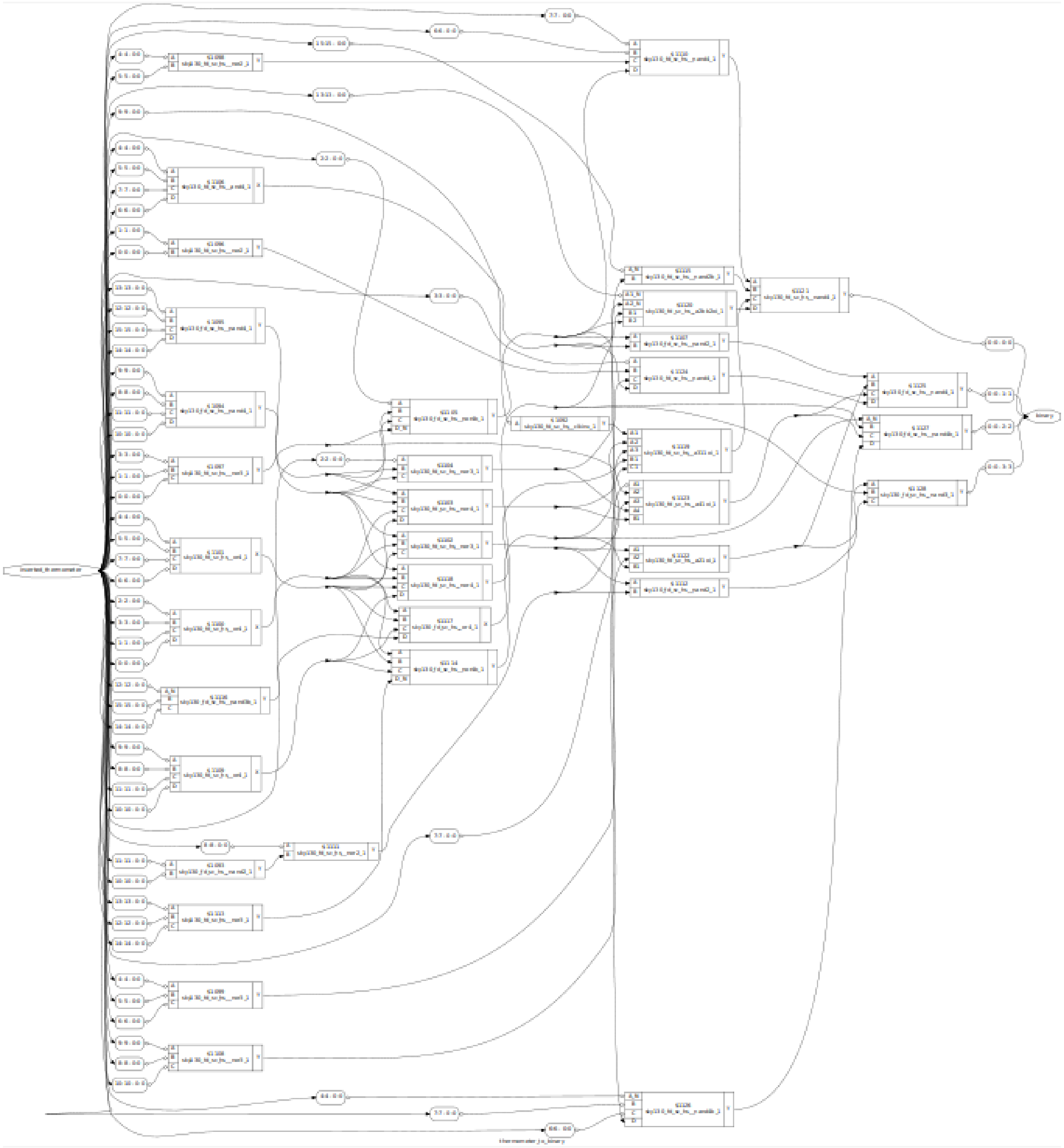
```
> > abc -liberty
/usr/local/share/pdk/sky130A/libs.ref/sky130_fd_sc_hs/lib/sky130_fd_sc_hs__tt_025C_1v89.lib
```

```
>> flatten      // // to invoke flat synthesis after netlist generation
// netlist generation
>> show // vew graphical
```



inverted_thermometer
7:7 - 0:0
4:4 - 0:0
5:5 - 0:0
6:6 - 0:0
3:3 - 0:0
1:1 - 0:0
0:0 - 0:0
2:2 - 0:0
9:9 - 0:0
8:8 - 0:0
10:10 - 0:0
11:11 - 0:0
13:13 - 0:0
12:12 - 0:0
14:14 - 0:0
15:15 - 0:0
$1098 sky130_fd_sc_hs__nor2_1
$1106 sky130_fd_sc_hs__and4_1
$1101 sky130_fd_sc_hs__or4_1
$1099 sky130_fd_sc_hs__nor3_1
$1097 sky130_fd_sc_hs__nor3_1
$1096 sky130_fd_sc_hs__nor2_1
$1100 sky130_fd_sc_hs__or4_1
$1092 sky130_fd_sc_hs__clkinv_1
$1108 sky130_fd_sc_hs__nor3_1
$1094 sky130_fd_sc_hs__nand4_1
$1109 sky130_fd_sc_hs__or4_1
$1093 sky130_fd_sc_hs__nand2_1
$1111 sky130_fd_sc_hs__nor2_1
$1113 sky130_fd_sc_hs__nor3_1
$1095 sky130_fd_sc_hs__nand4_1
$1116 sky130_fd_sc_hs__nand3b_1
$1105 sky130_fd_sc_hs__nor4b_1
$1104 sky130_fd_sc_hs__nor3_1
$1103 sky130_fd_sc_hs__nor4_1
$1102 sky130_fd_sc_hs__nor3_1
$1118 sky130_fd_sc_hs__nor4_1
$1114 sky130_fd_sc_hs__nor4b_1
$1117 sky130_fd_sc_hs__or4_1
$1126 sky130_fd_sc_hs__nand4b_1
$1110 sky130_fd_sc_hs__nand4_1
$1107 sky130_fd_sc_hs__nand2_1
$1124 sky130_fd_sc_hs__nand4_1
$1123 sky130_fd_sc_hs__a41oi_1
$1120 sky130_fd_sc_hs__a2bb2oi_1
$1119 sky130_fd_sc_hs__a311oi_1
$1112 sky130_fd_sc_hs__nand2_1
$1122 sky130_fd_sc_hs__a21oi_1
$1115 sky130_fd_sc_hs__nand2b_1
$1121 sky130_fd_sc_hs__nand4_1
$1125 sky130_fd_sc_hs__nand4_1
$1127 sky130_fd_sc_hs__nand4b_1
$1128 sky130_fd_sc_hs__nand3_1
0:0 - 1:1
0:0 - 2:2
0:0 - 0:0
0:0 - 3:3
binary
thermometer_to_binary

```
>> write_verilog therm.v // write the generated netlist to a new moule/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
```



```
>> abc -liberty
/usr/local/share/pdk/sky130A/libs.ref/sky130_fd_sc_hs/lib/sky130_fd_sc_hs__tt
_025C_1v89.lib
```

- simulate the complete (ultra small) ADC

References:

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

```
-----------------
```

Run iverilo/icarus

```
iverilog file.v

vvp a.out
```

```
---------------------------
```