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.

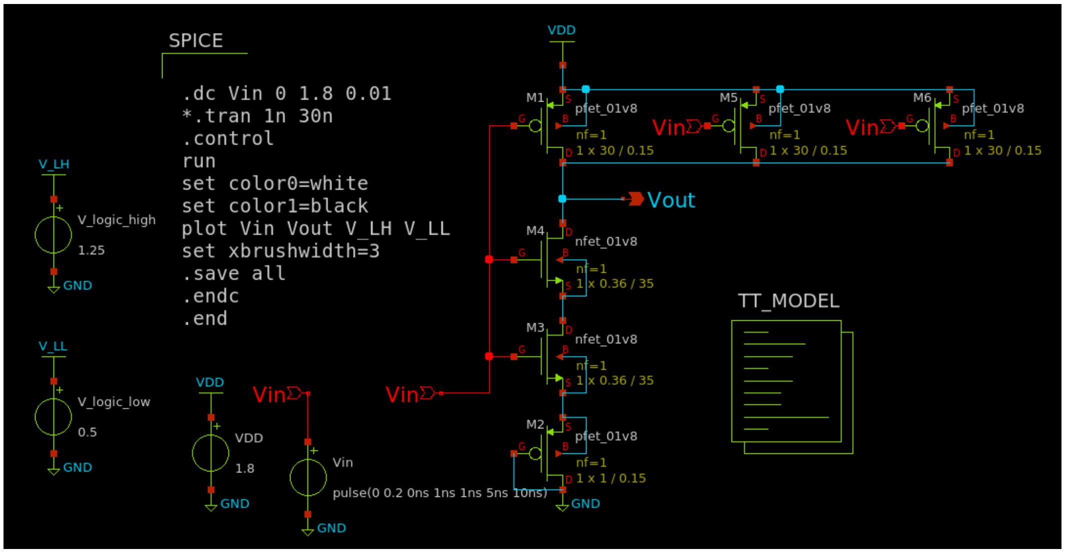
Steps:

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

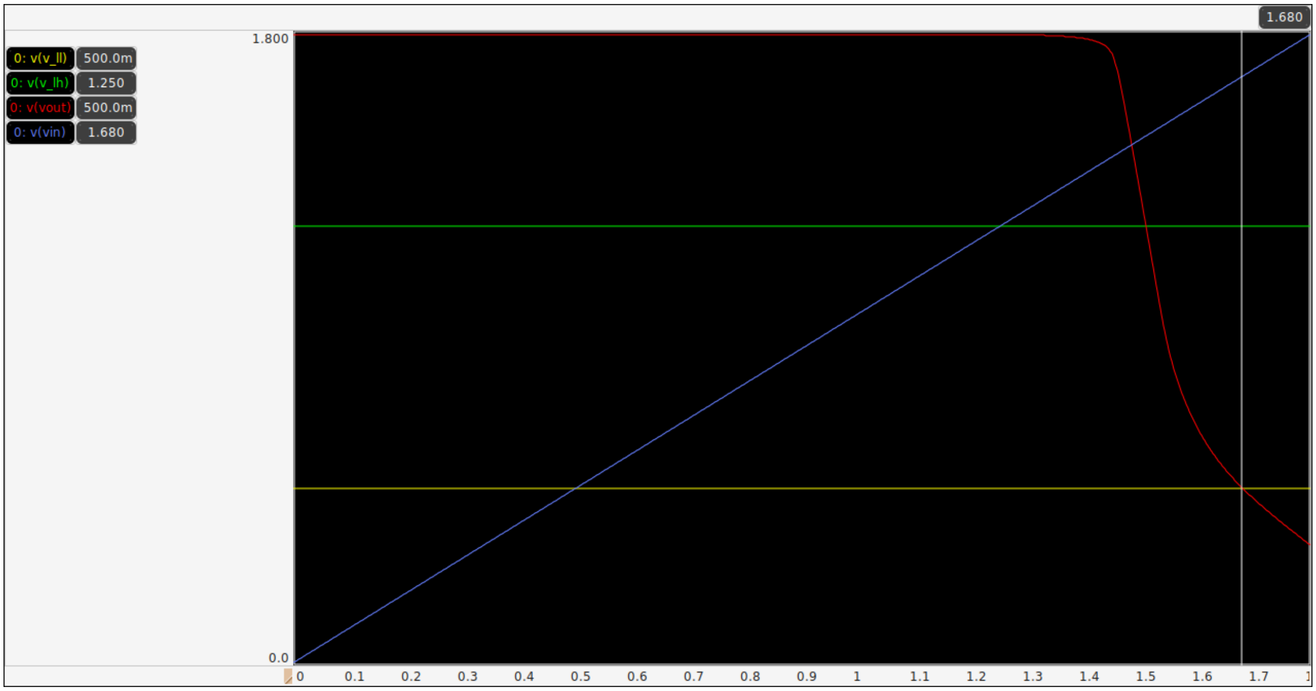
Technology	L Voltage	H Voltage	note
CMOS	0 to 30% of Vdd	70% of Vdd to Vdd	Vdd = Supply voltage
	≈0.5 Volt	≈1.25 Volts	Vdd = 1.8 Volts



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:



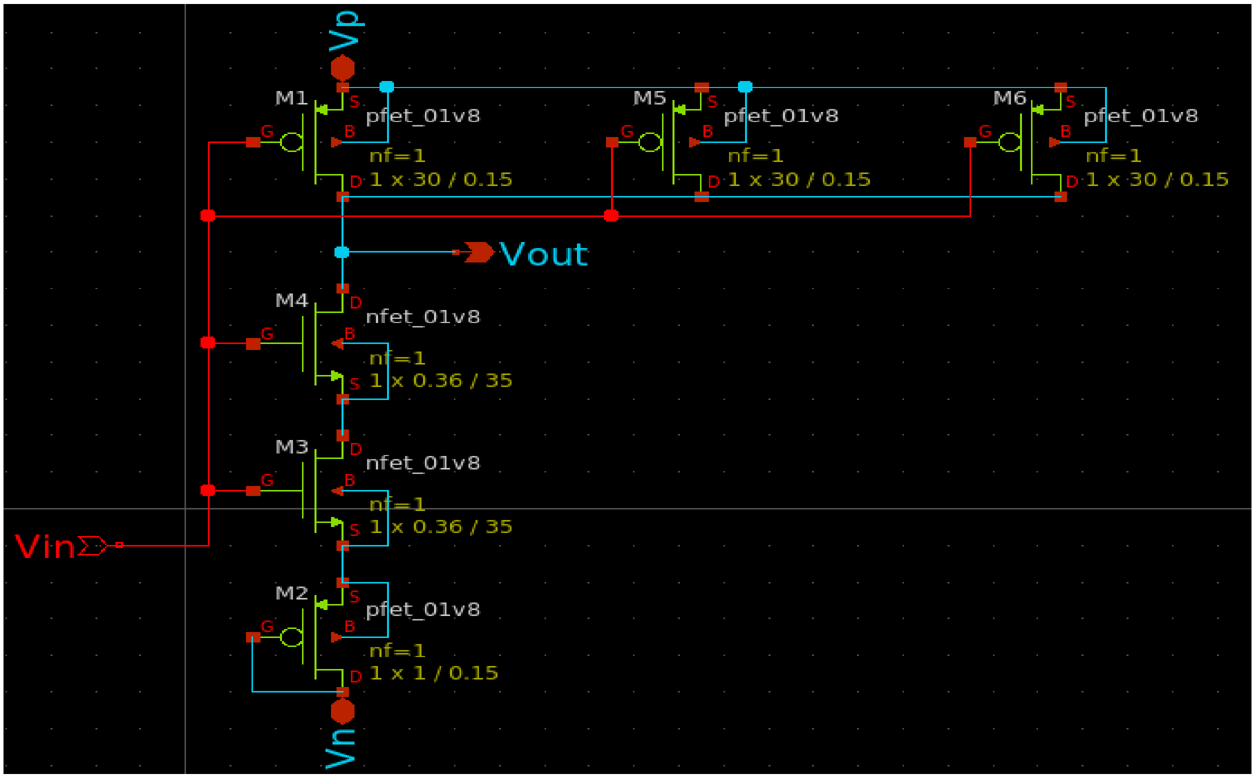
The inverter will switch from logic-high or undefined to logic-zero when input voltage exceeds 1.68 volts. So, for this inverter the threshold is 1.68 volts.

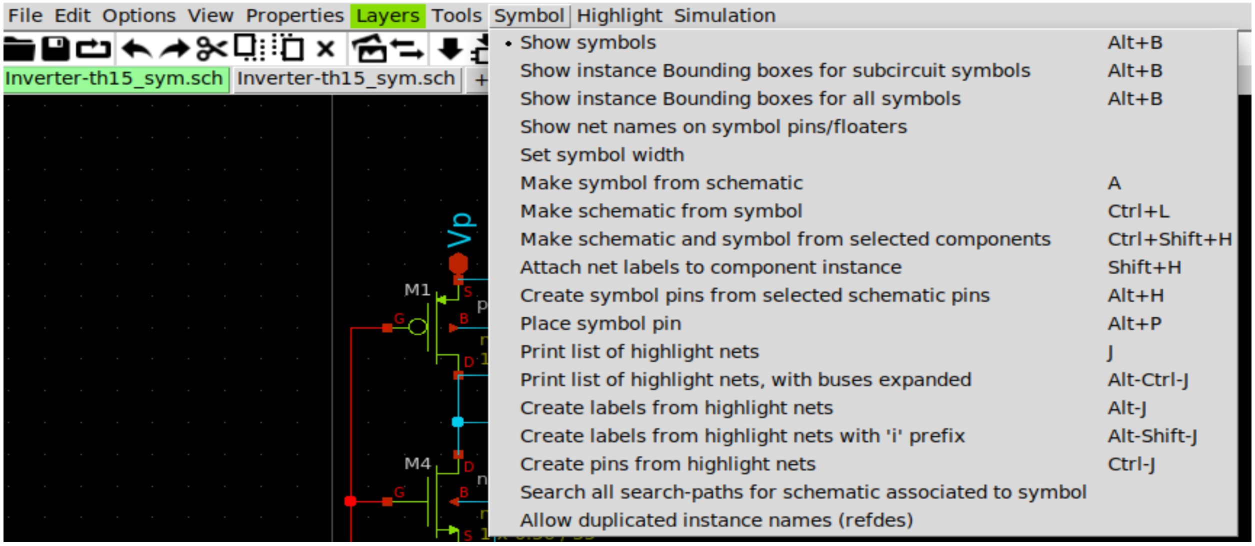
For a 4-bit ADC we need 16 voltage level including zero (0~1.68 volts). The quantization should be 1.68/16=0.112 volts.

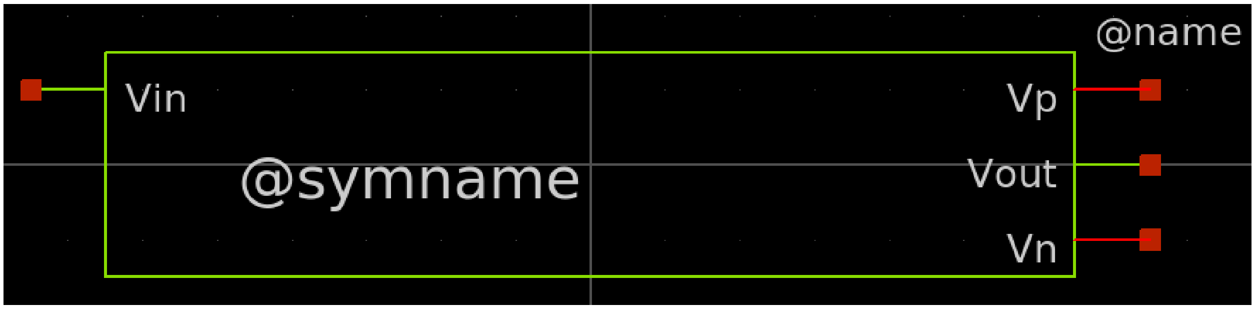
Vdd = 1.8 V

Th. 15: 1.680

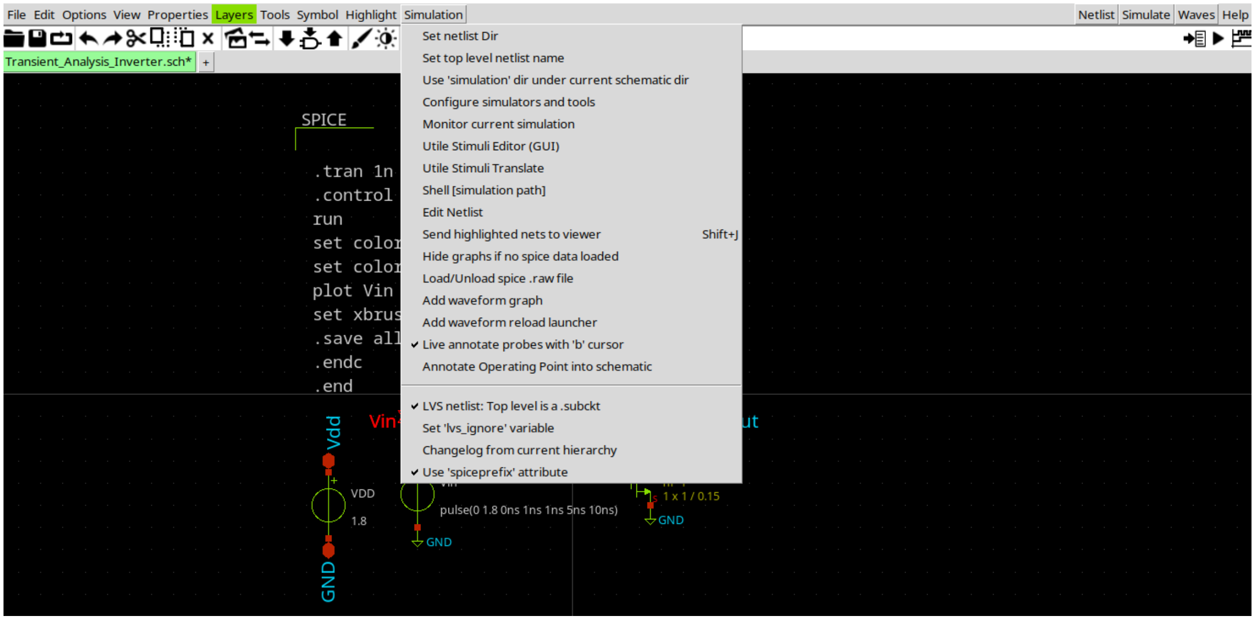
For calculation the schematic and the plot already mentioned earlier. Remove all source to make symbol from schematic. As the symbol will have to be connected to Vdd and GND, those pins are replaced with Vp and Vn accordingly.




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



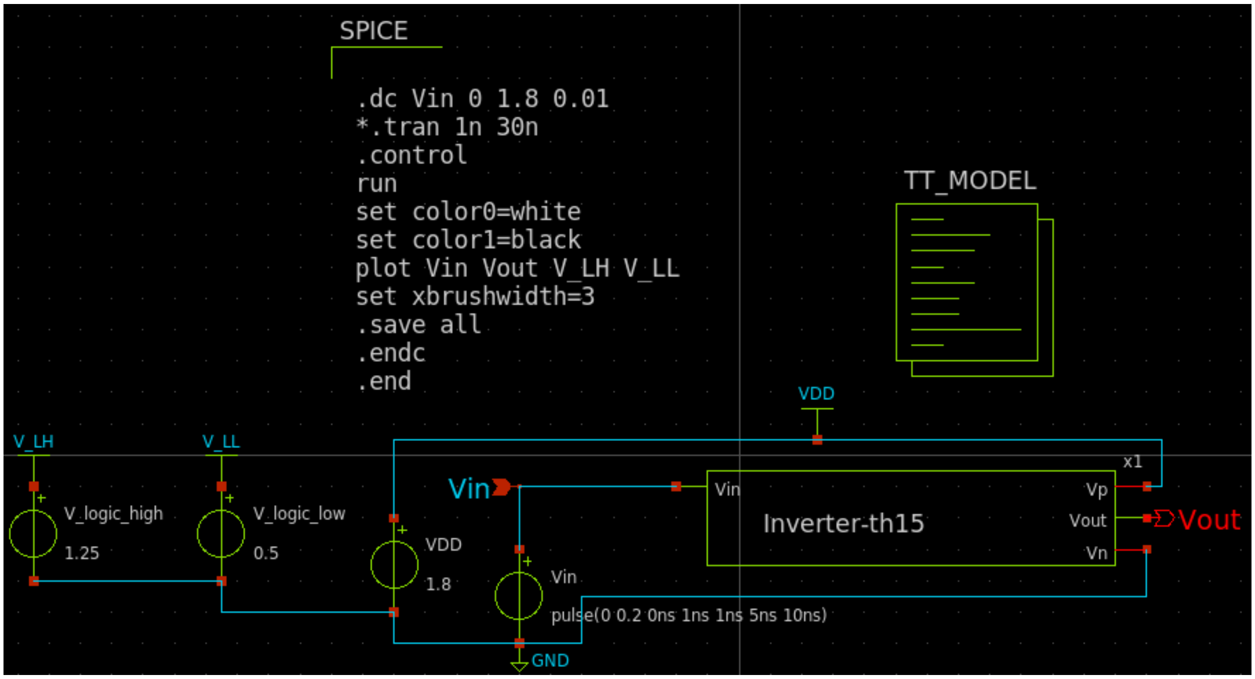
Magic=>before importing netlist to ngspice check LVS net


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
Vdd
Vin
VDD
1.8
GND
pulse(0 1.8 0ns 1ns 1ns 5ns 10ns)
1 x 1 / 0.15
GND
GND
ut

SPICE

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

The simplified schematic is as follows:

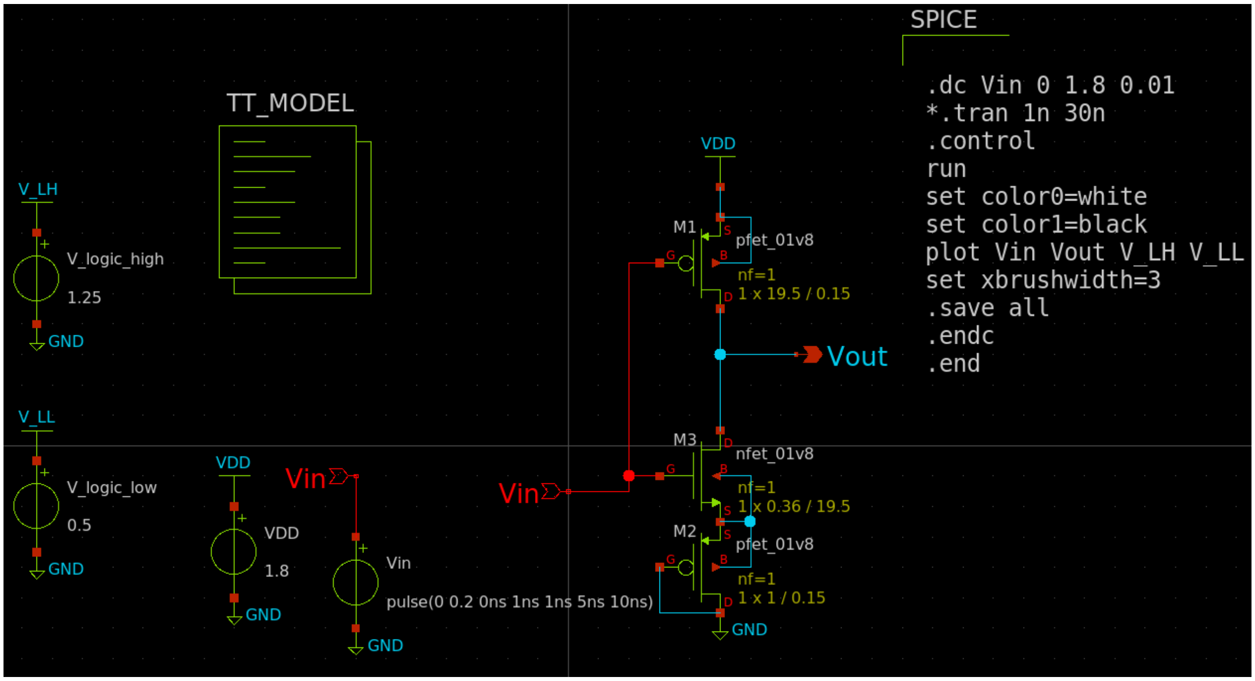


SPICE

```
.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
```

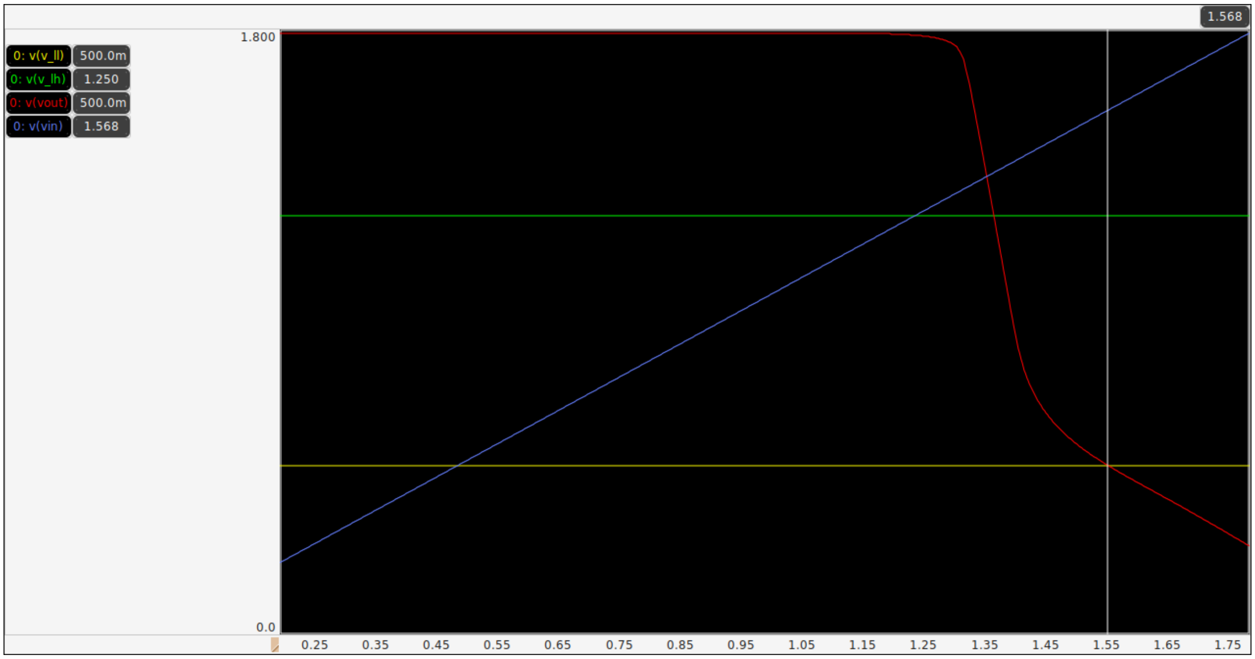
The plot is similar to the previous figure.

Th. 14: 1.568



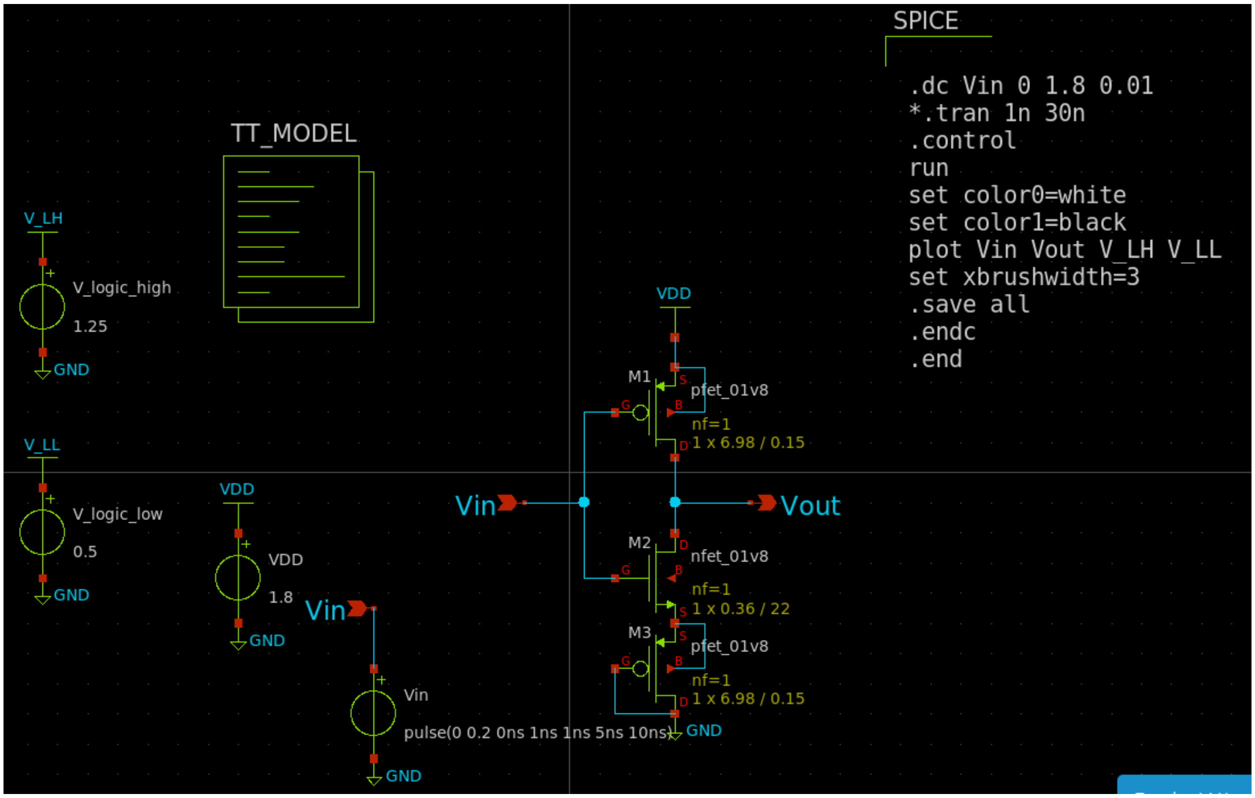
SPICE

```
.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
```

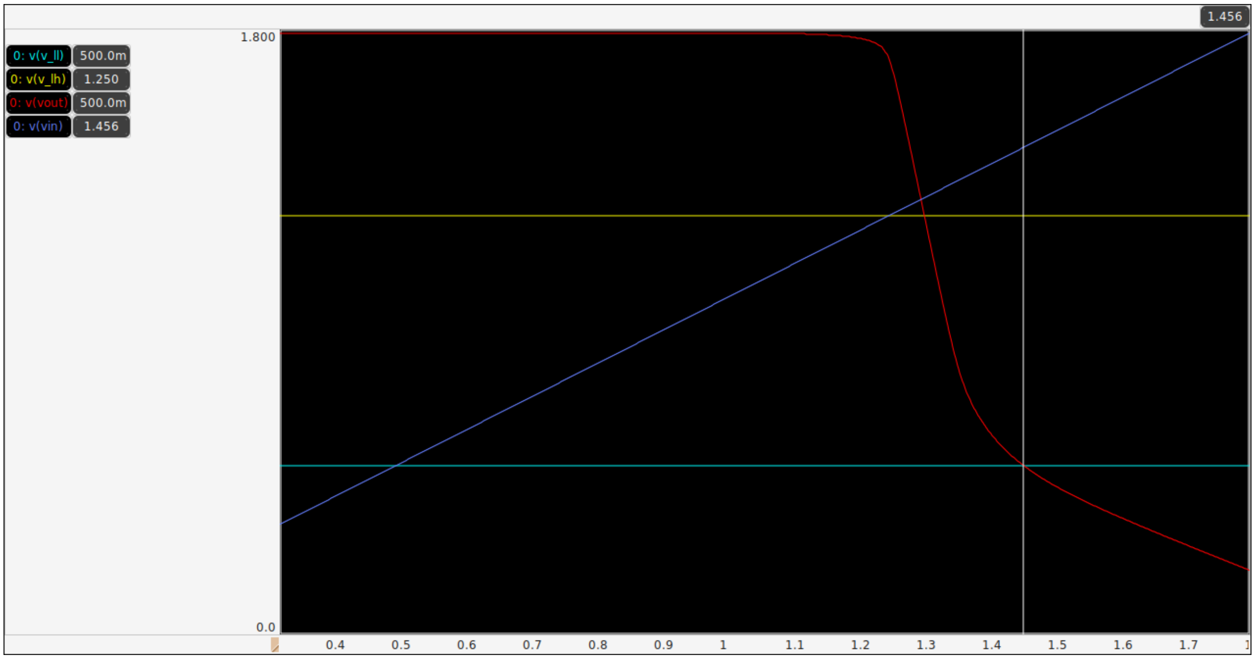


Rest of the figures are similar to the previous figures.

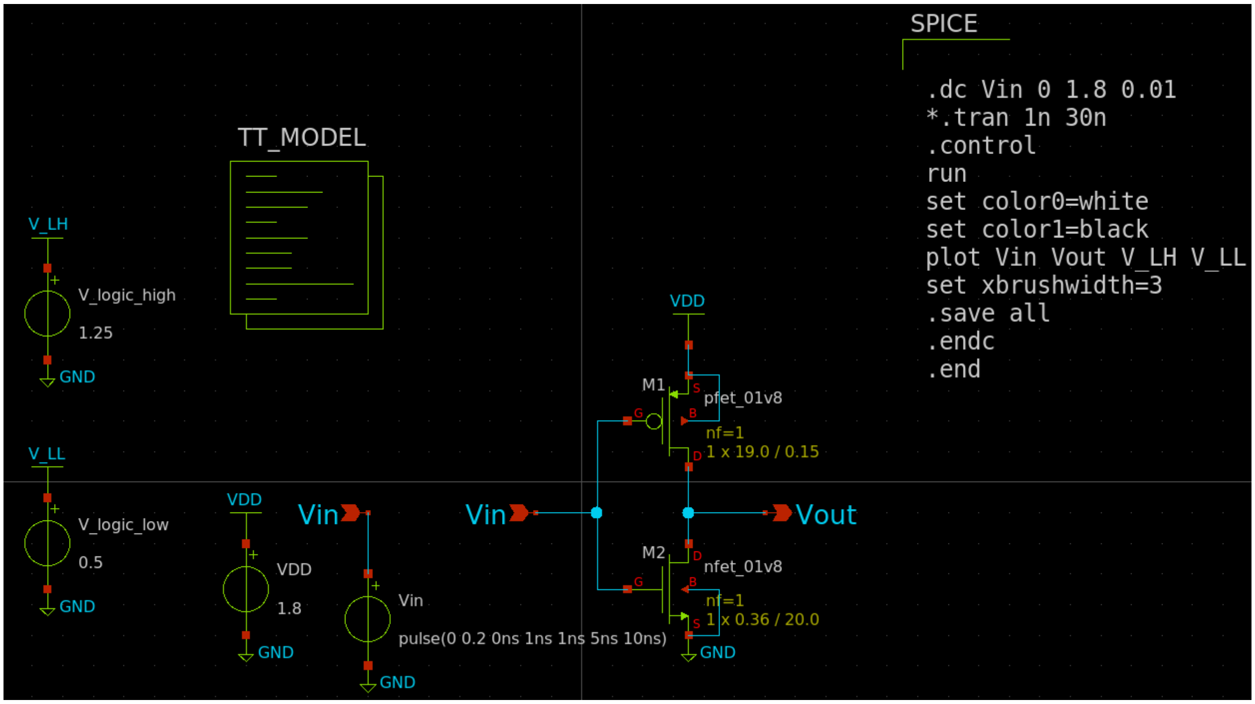
Th. 13: **1.456:**



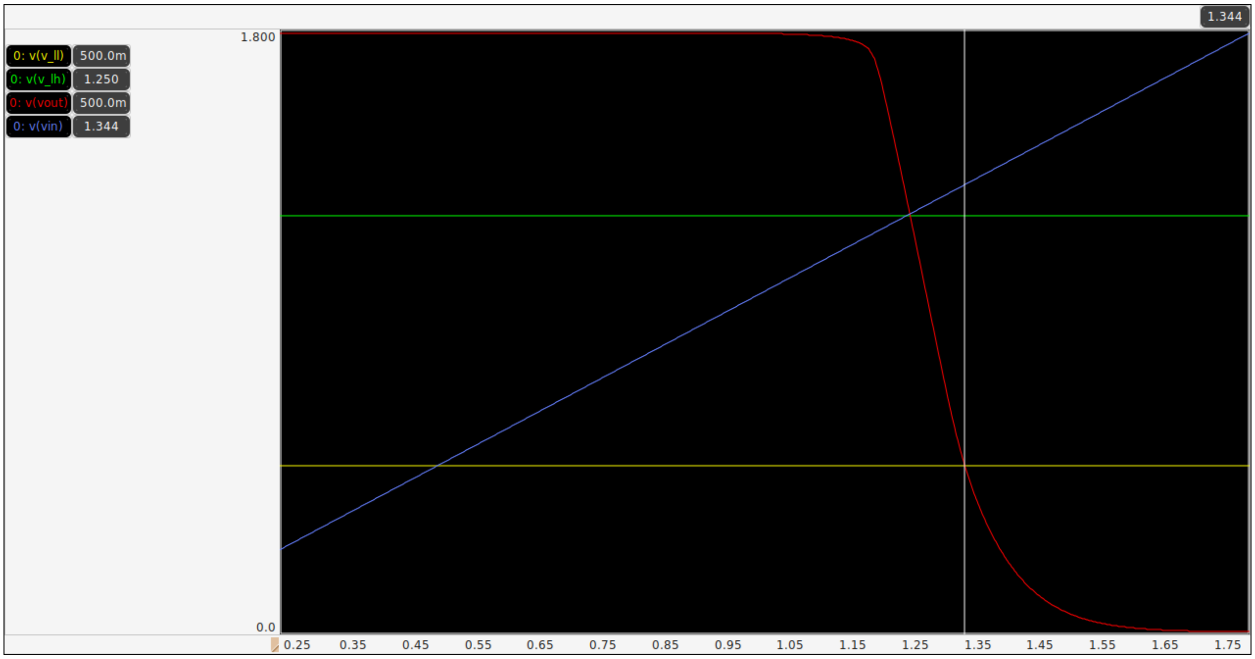
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



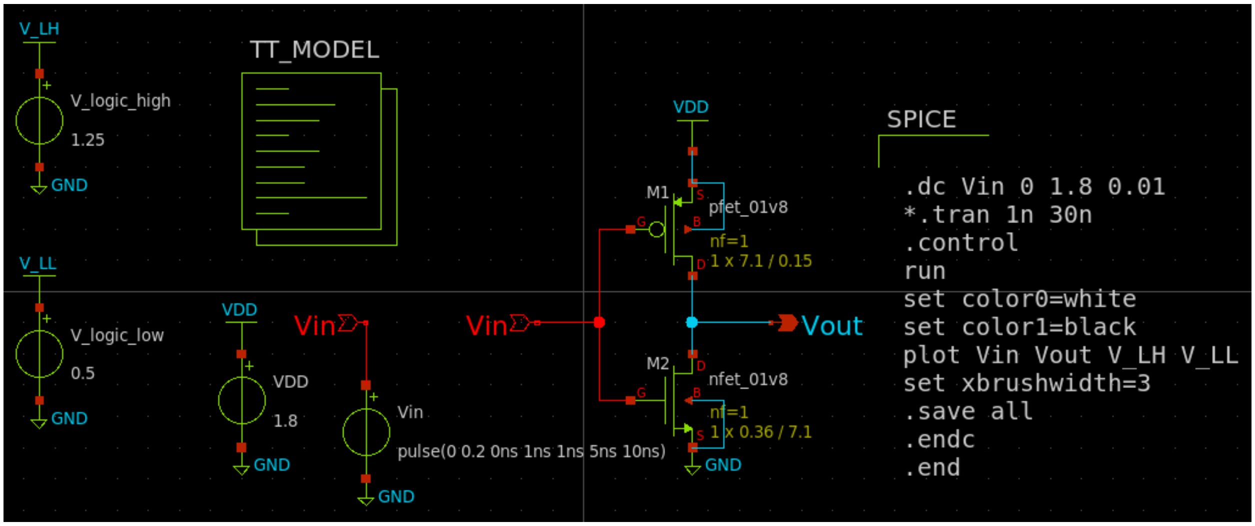
Th. 12: **1.344**



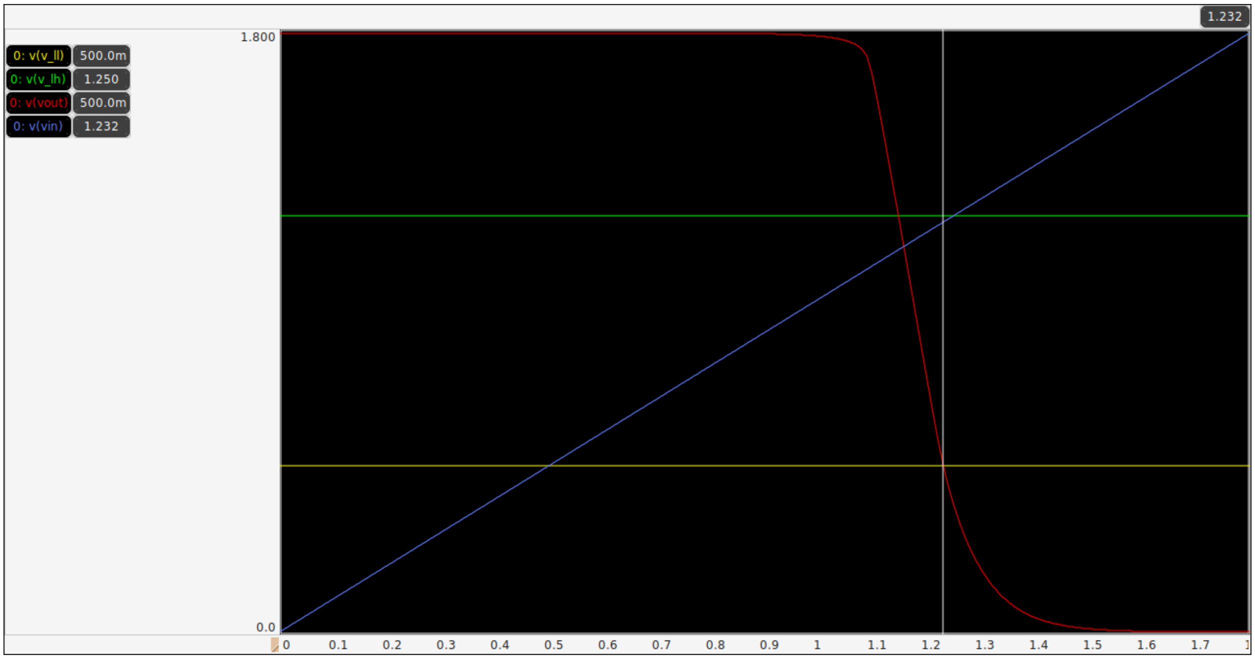
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



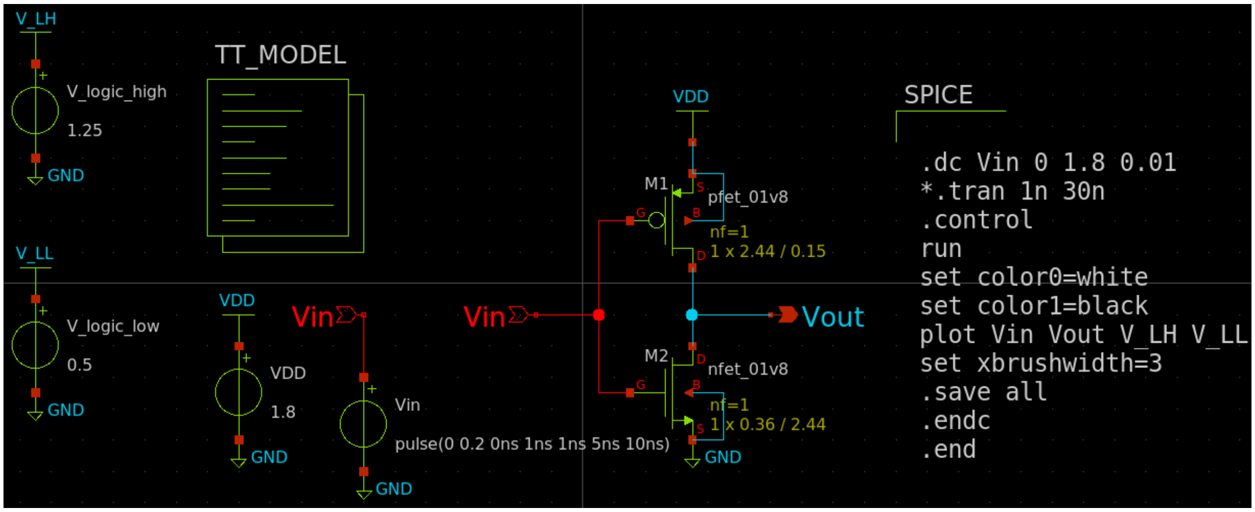
Th. 11: 1.232



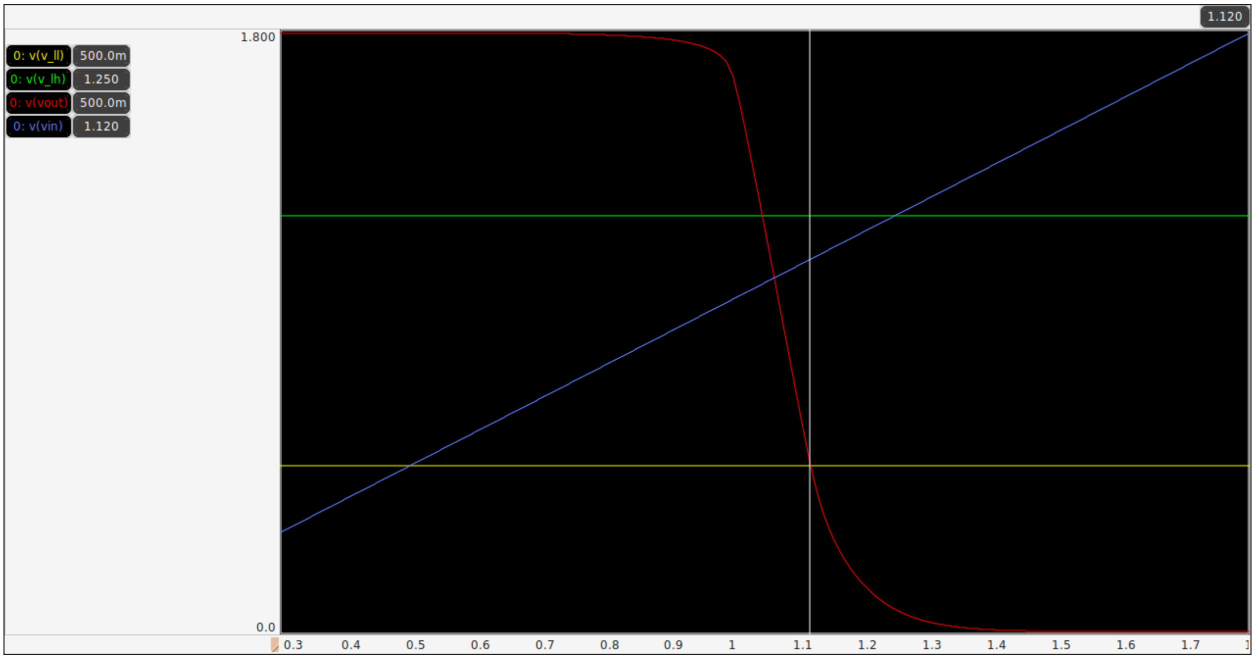
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



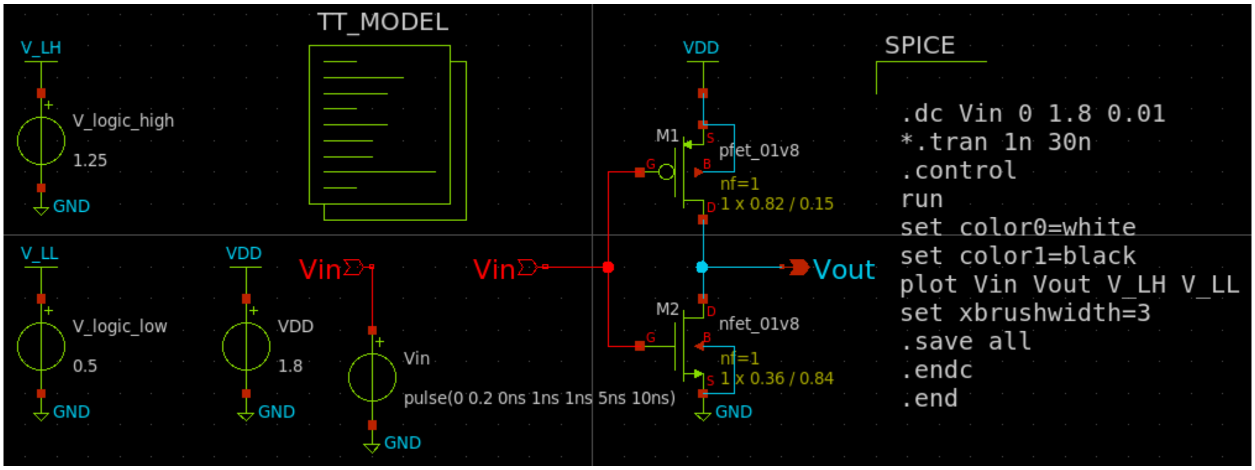
Th. 10: **1.12**



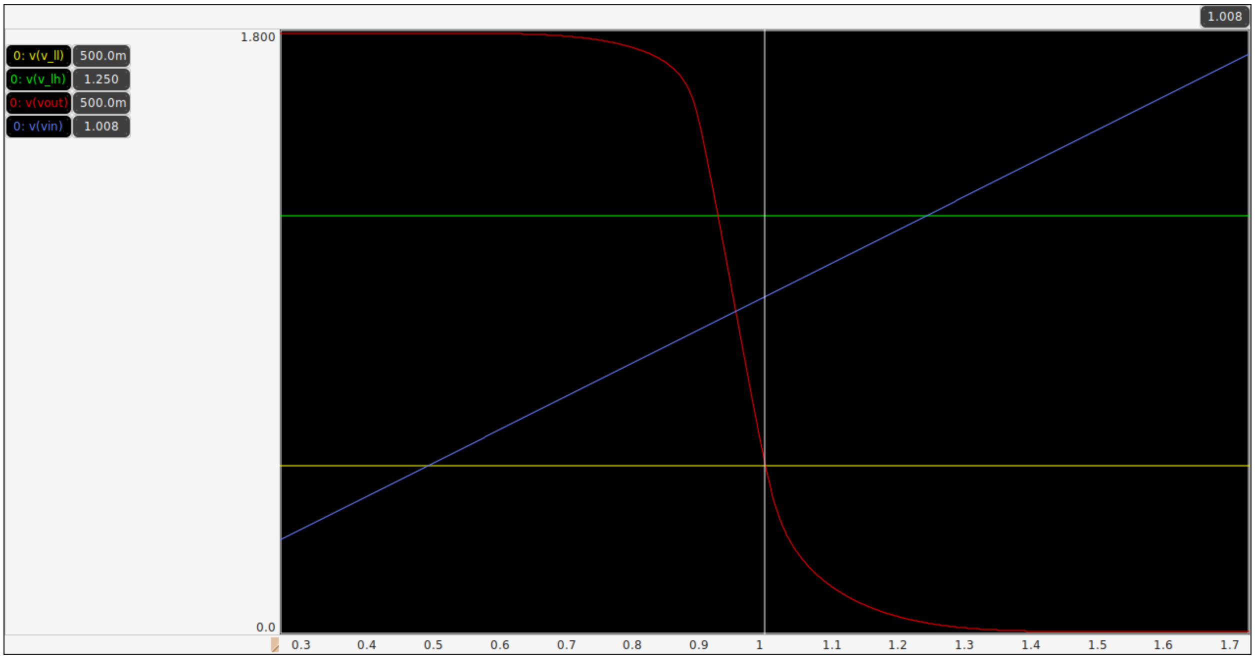
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



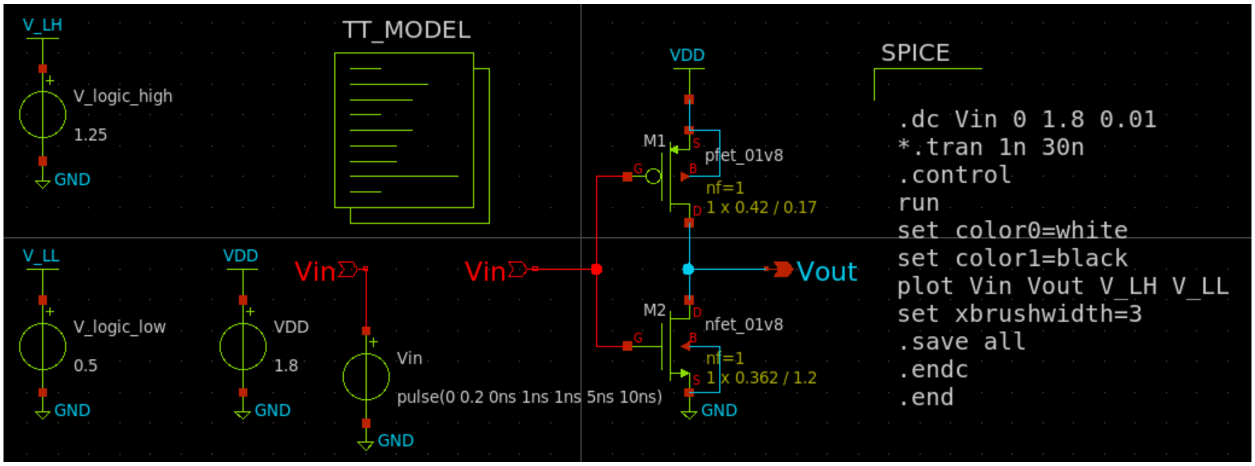
Th. 9: 1.008



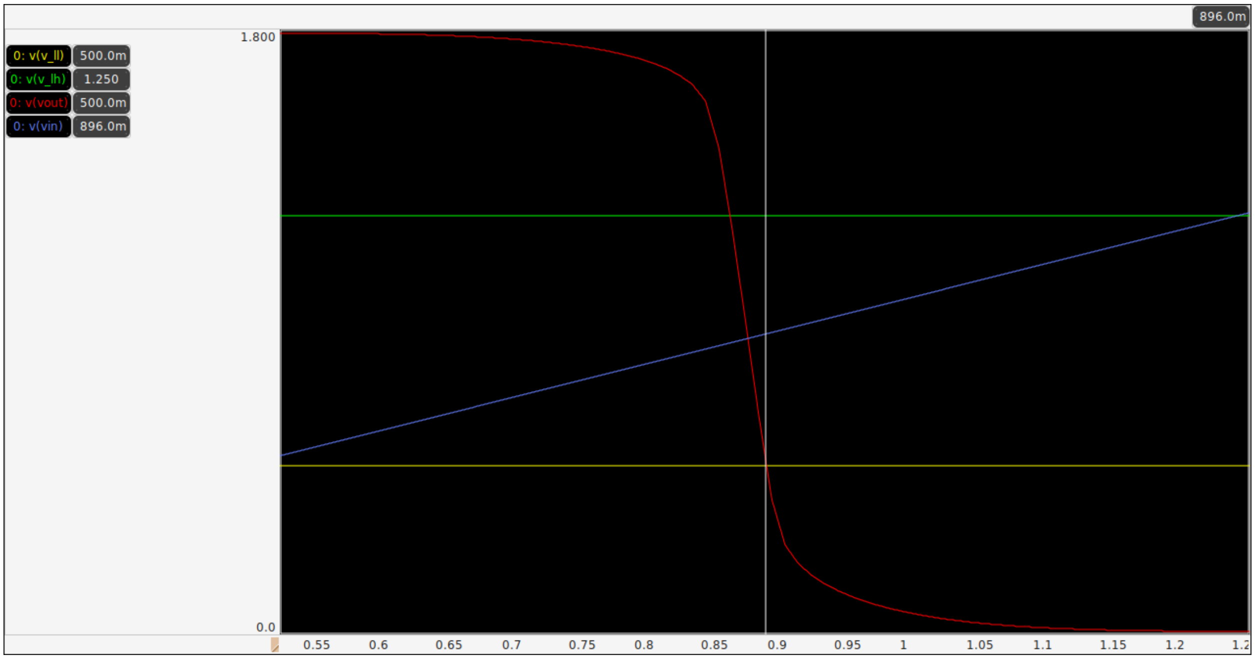
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



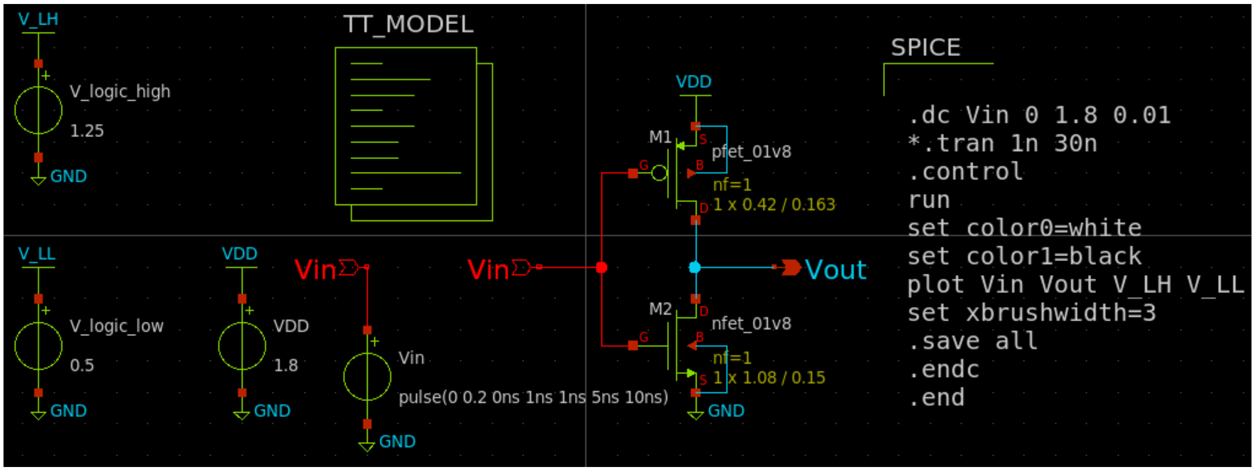
Th. 8: 0.896



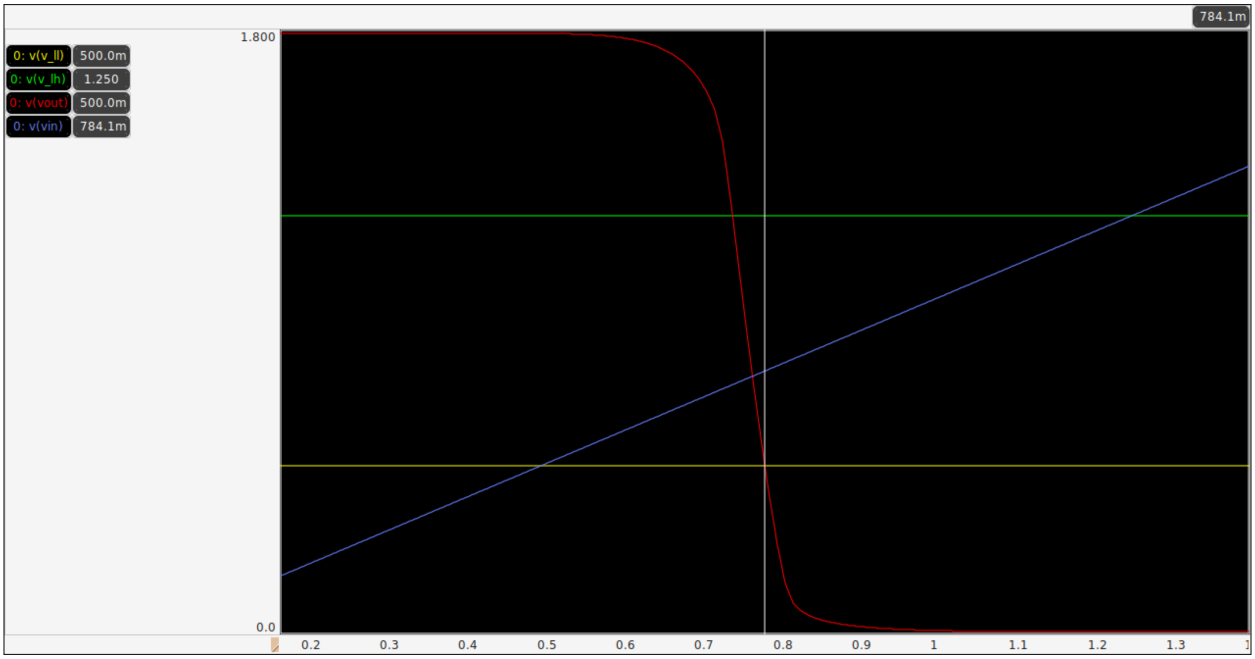
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



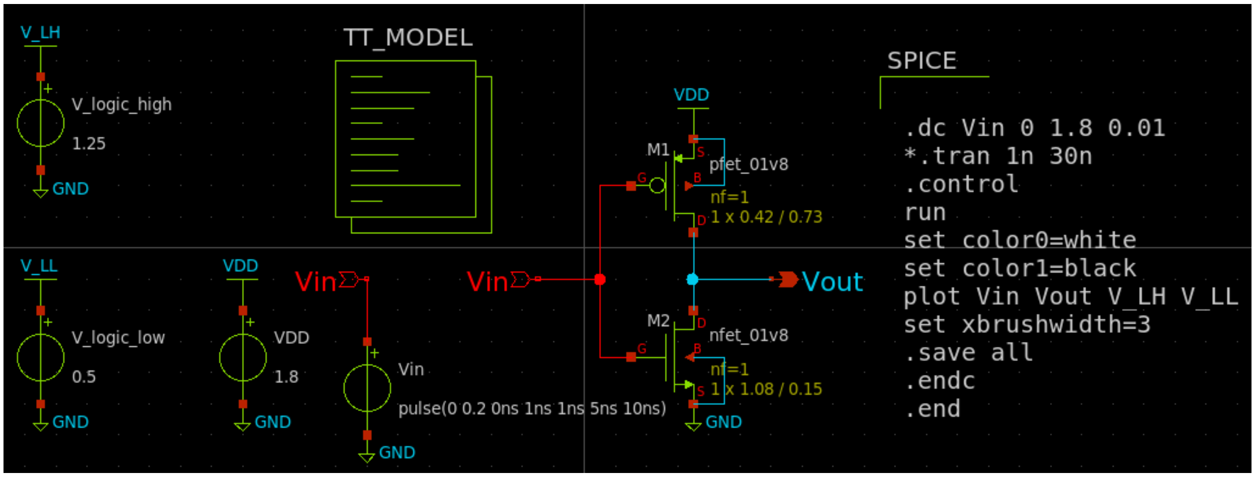
Th. 7: 0.784



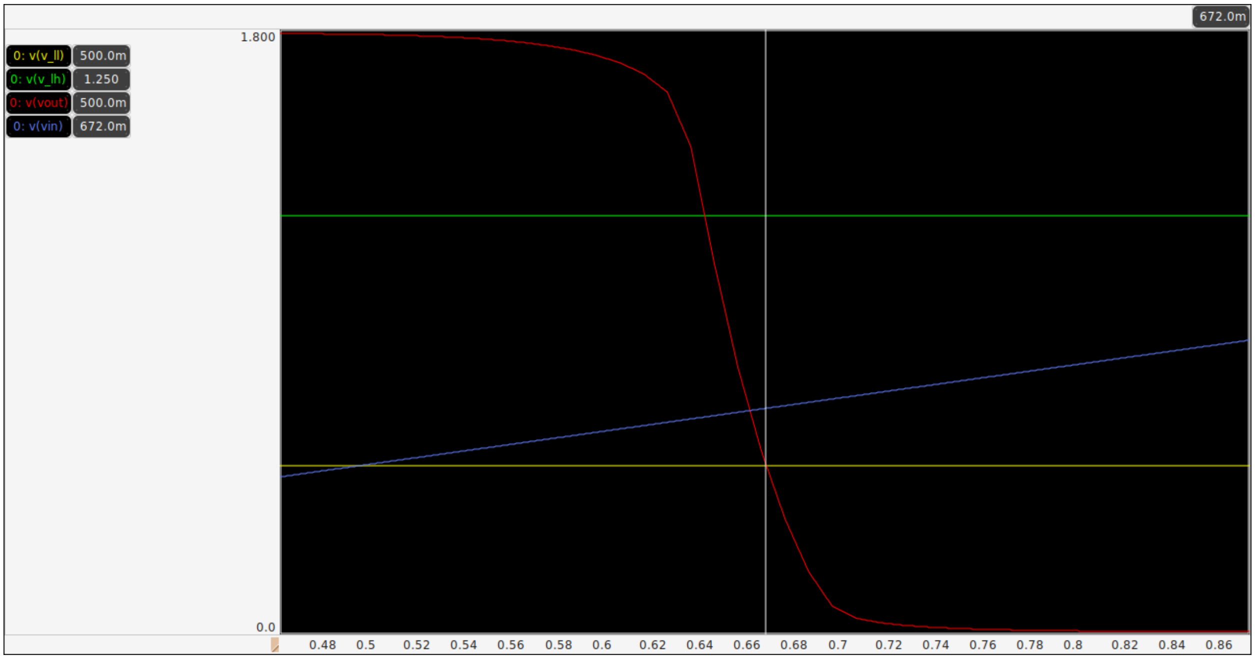
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



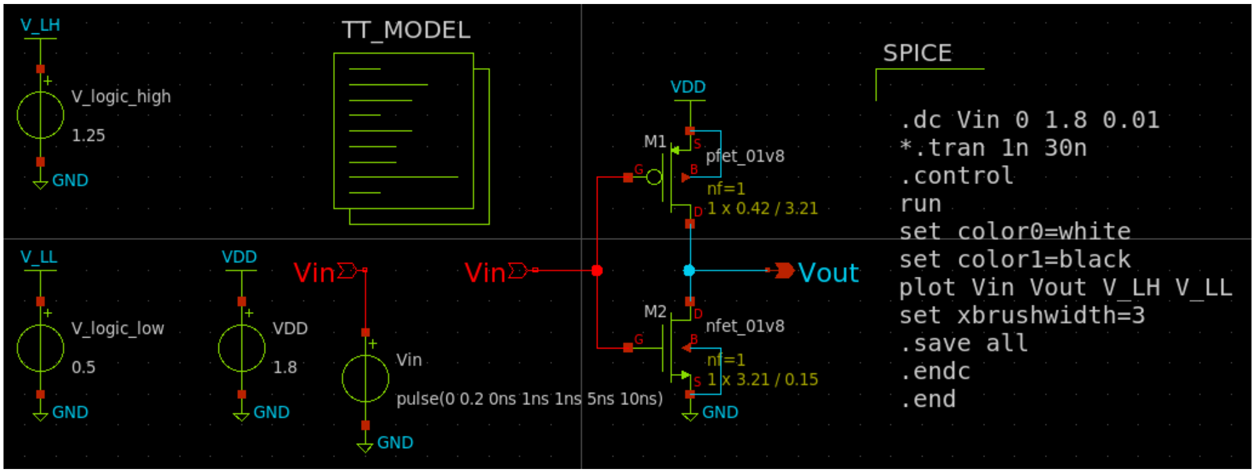
Th. 6: 0.672



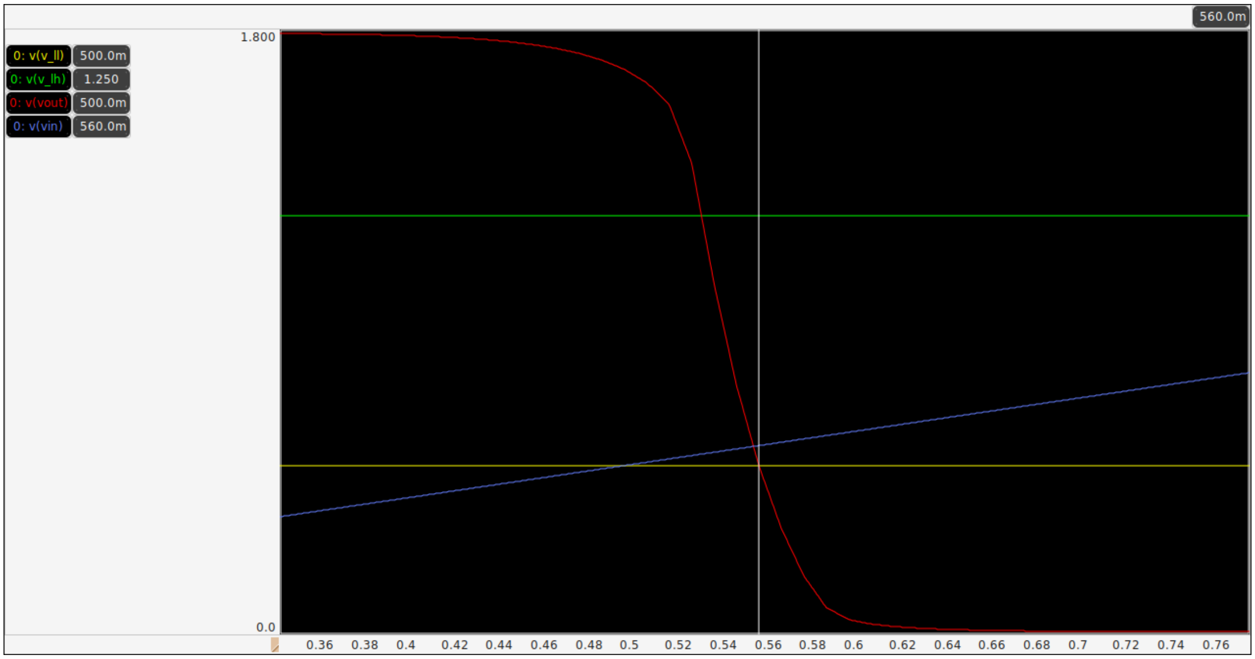
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



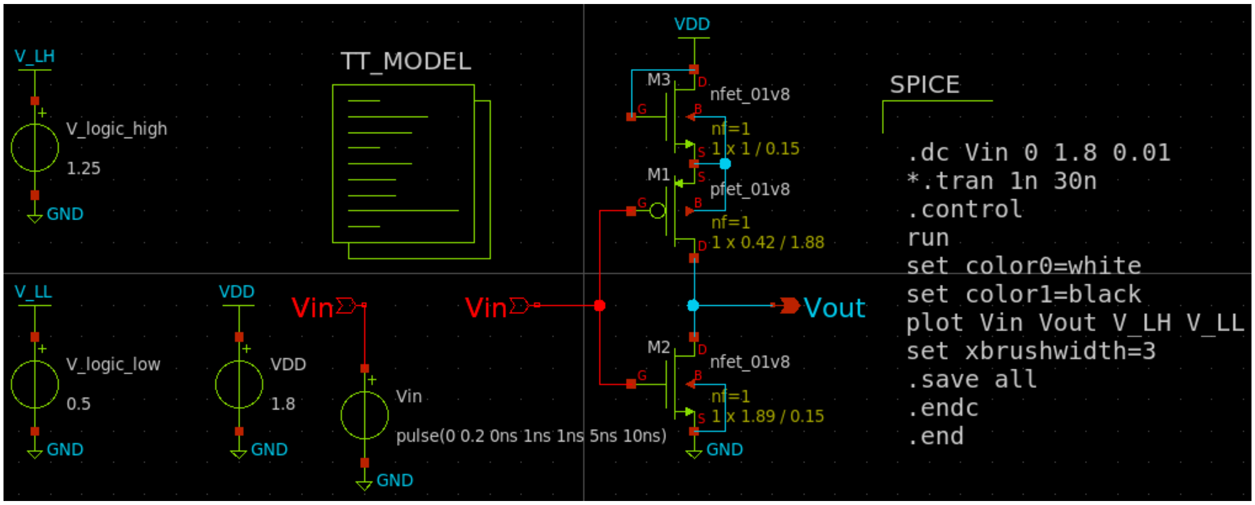
Th. 5: 0.56



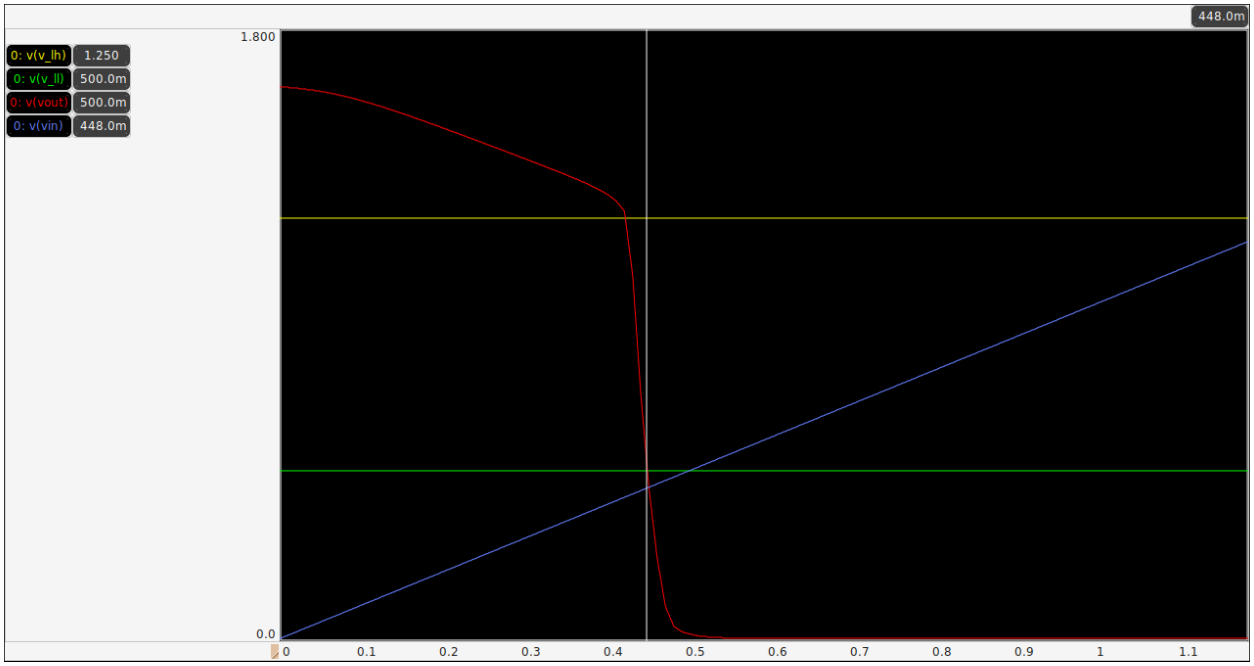
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



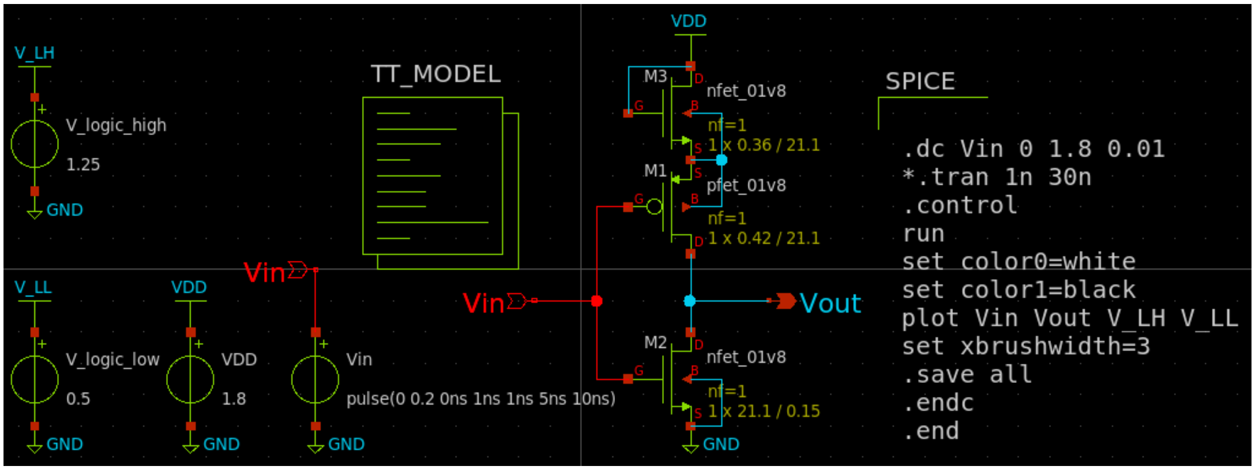
Th. 4: 0.448



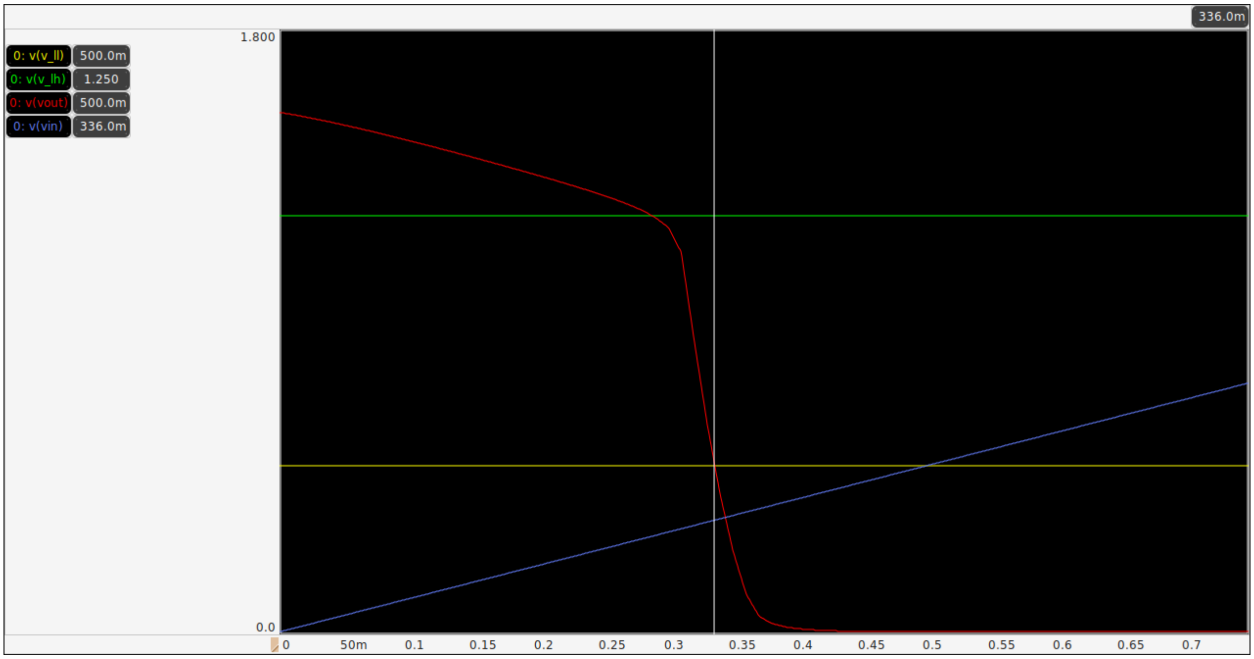
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



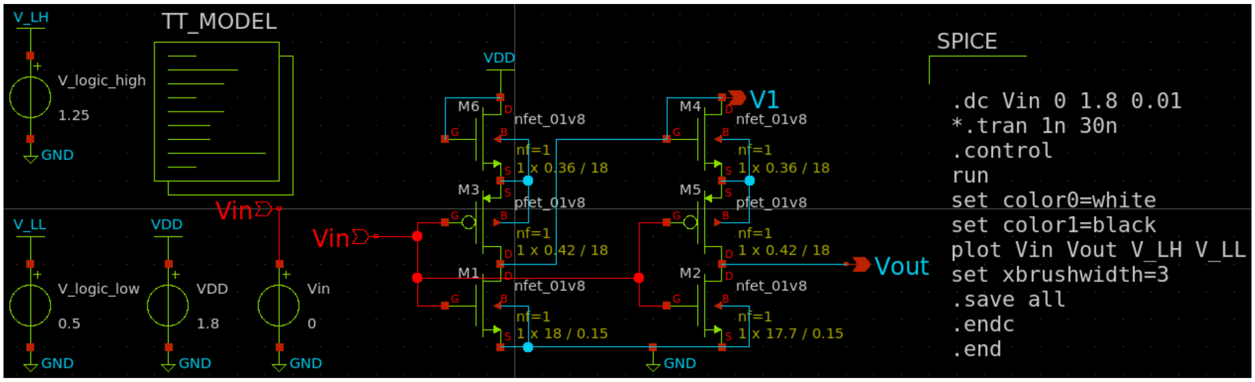
Th. 3: 0.336



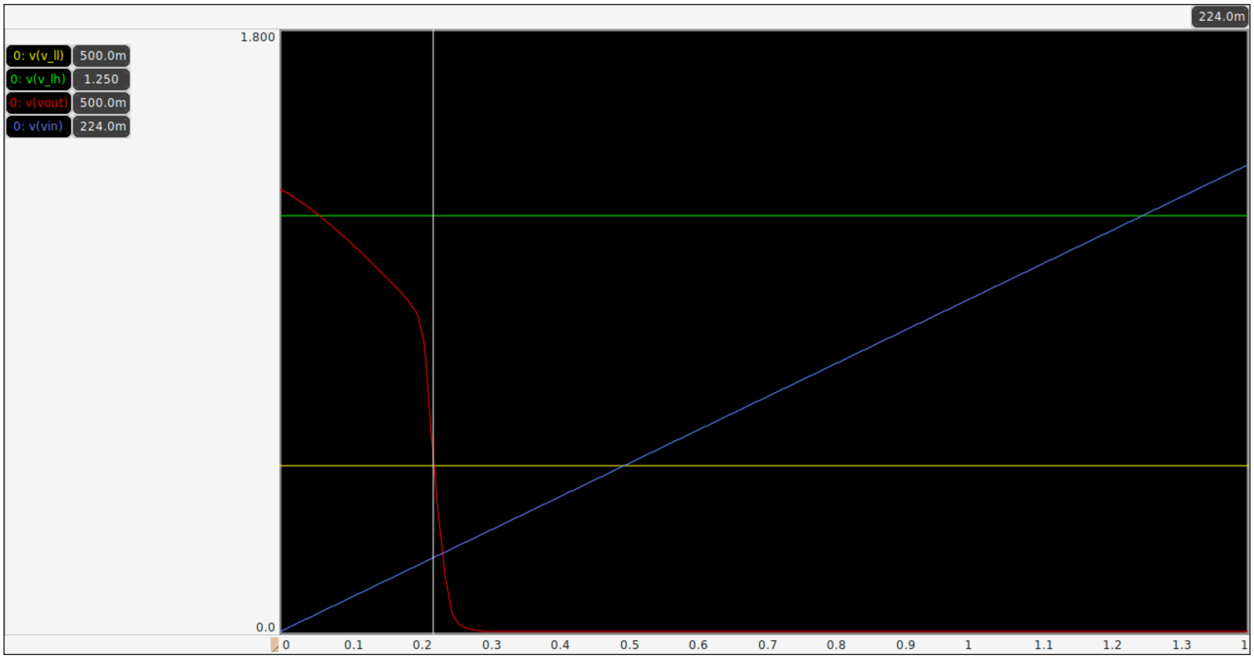
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



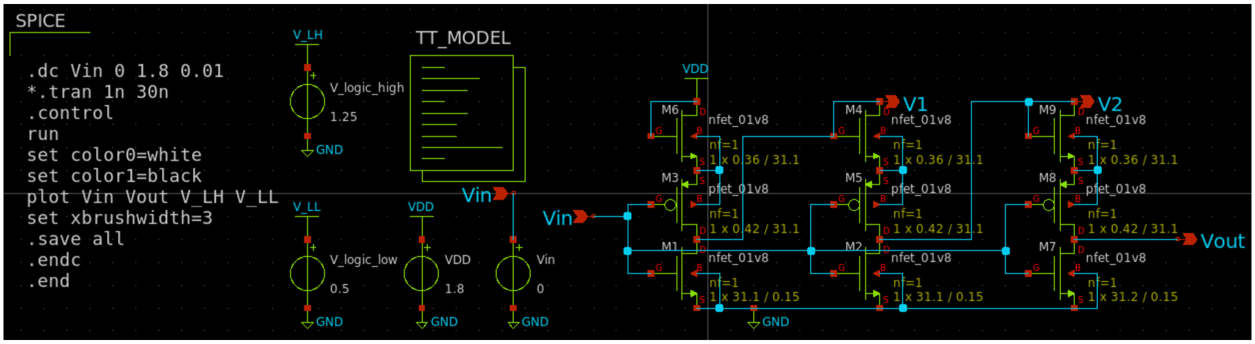
Th. 2: 0.224



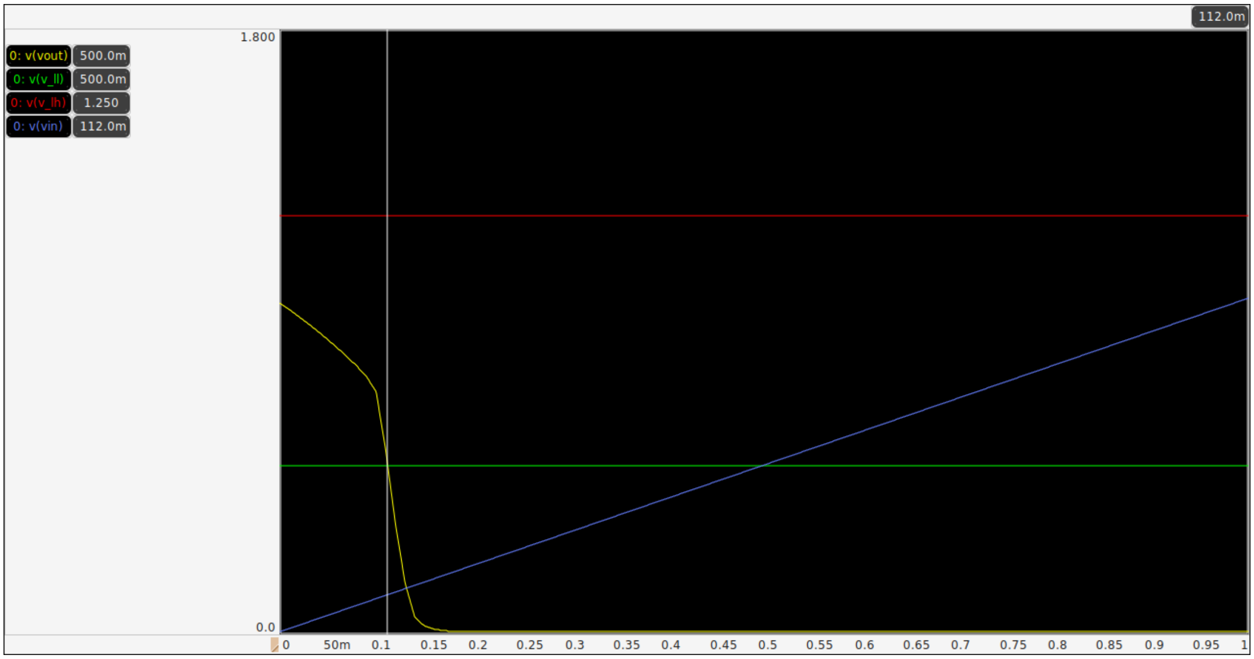
```
.dc Vin 0 1.8 0.01
*.tran 1n 30n
.control
run
set color0=white
set color1=black
plot Vin Vout V_LH V_LL
set xbrushwidth=3
.save all
.endc
.end
```



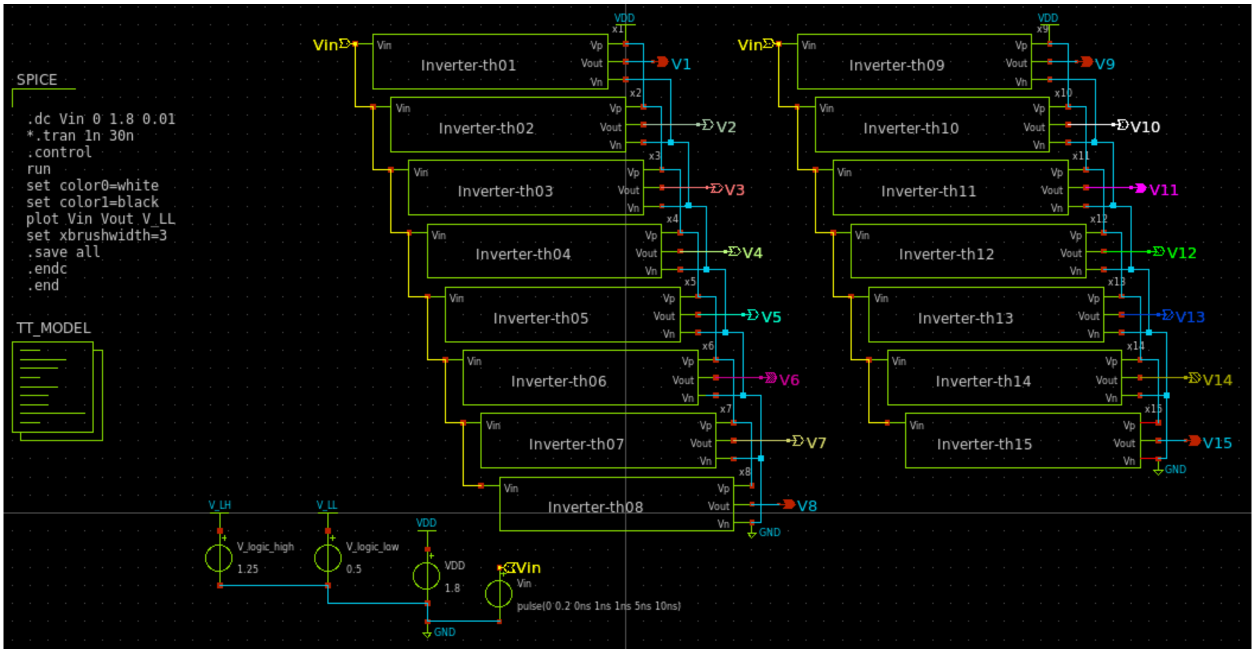
Th. 1: 0.112



```
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
```

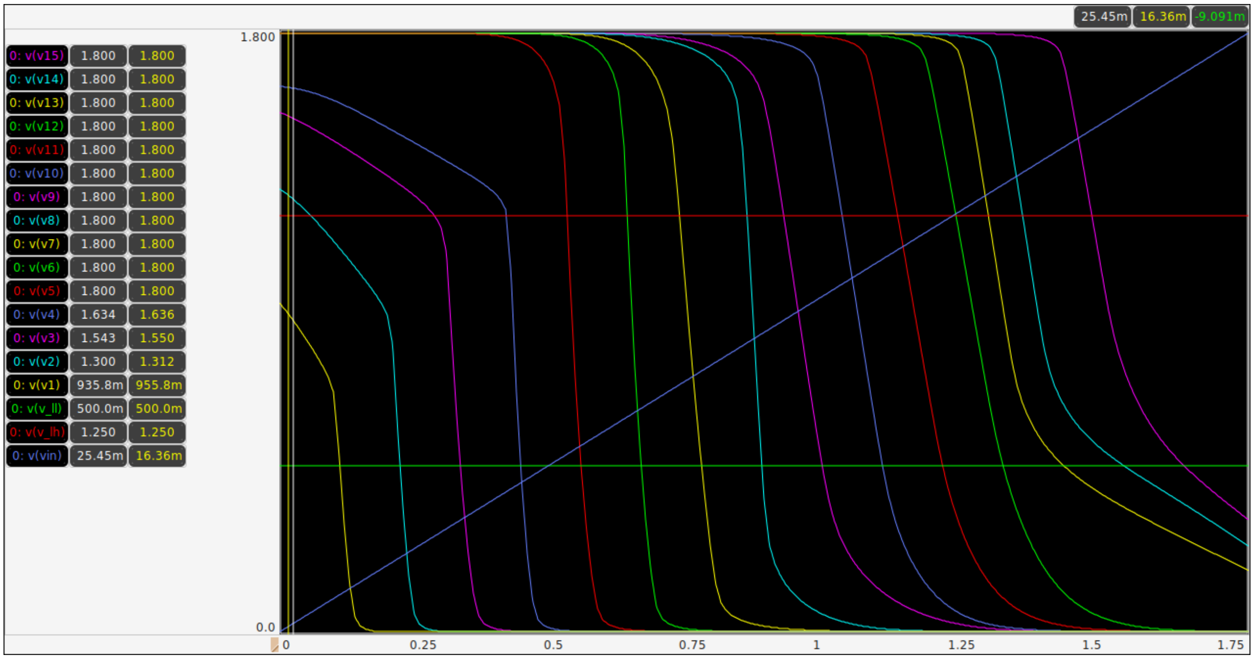


Combined Analog sch.



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://www.youtube.com/playlist?list=PLfGJEQLQIDBN0VsXQ68_FEYyqcym8CTDN

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

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
- simulate the complete (ultra small) ADC

References:

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

- What is Schematic Capture and why do we need it?
- What is a process?

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

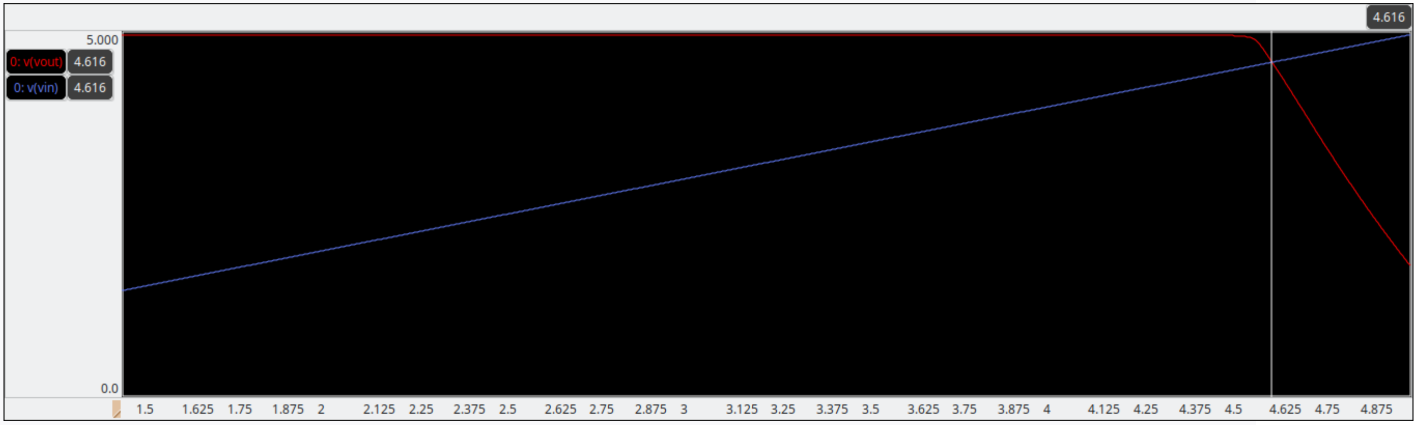
Run iverilo/icarus

```
iverilog file.v
```

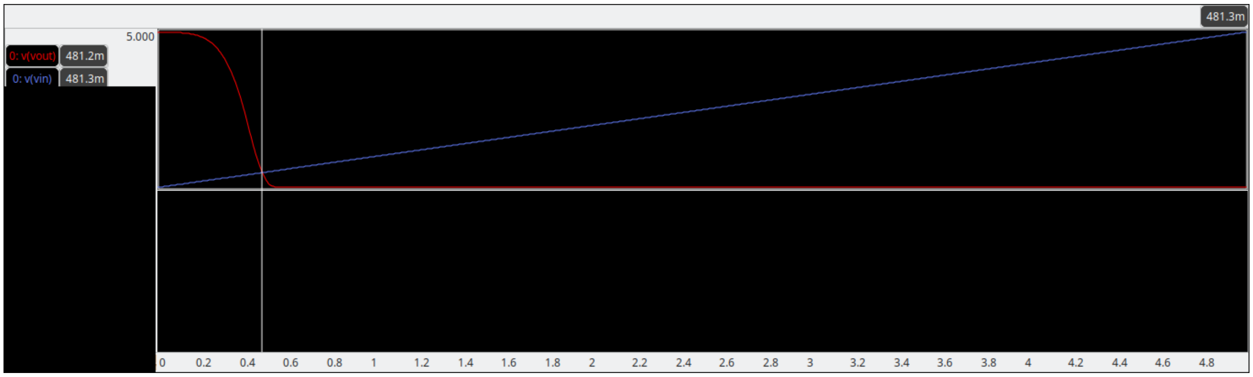
```
vvp a.out
```

Vdd = 5 Volt

Vsat_max = 4.616



Vsat_min = 481.3mV

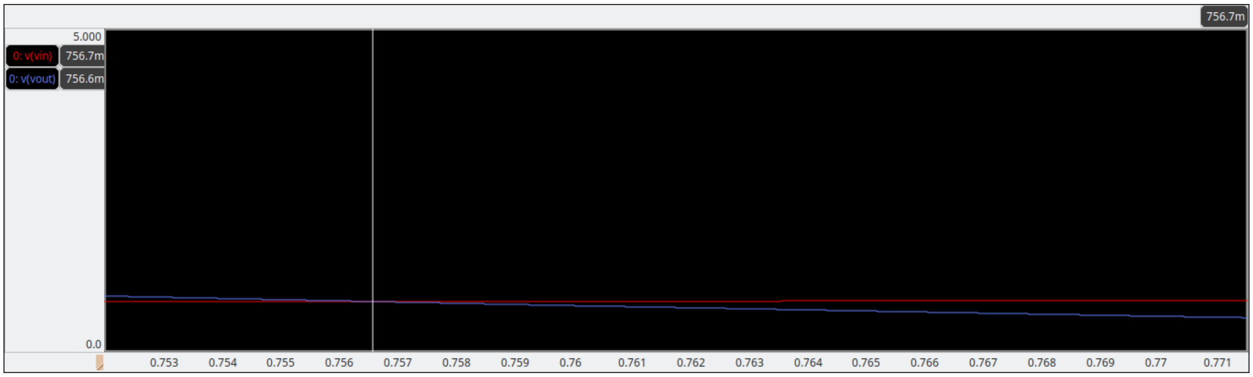


Voltage range = 4.616 – 0.4813 = 4.1347 V

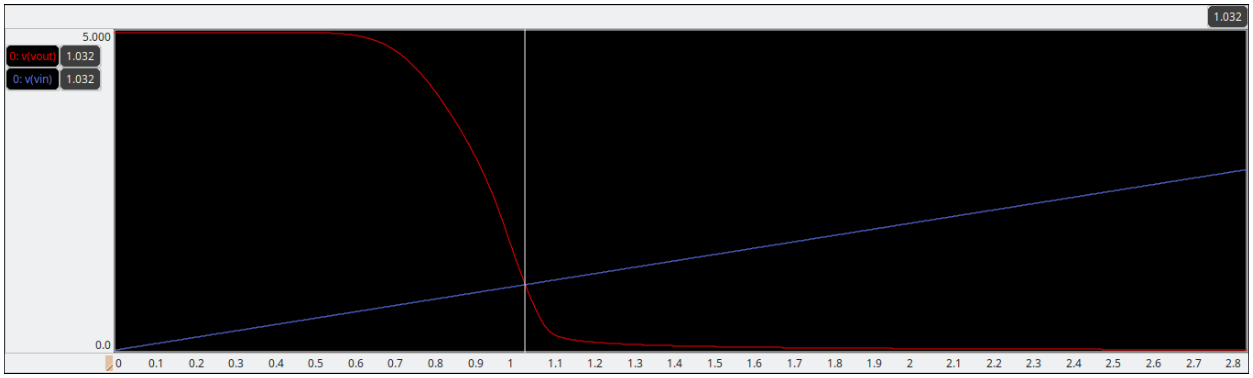
Step voltage = 4.1347/15 = 0.275647

Set. 1: 0.4813

Set. 2: 0.756947



Set. 3: 1.032594



Set. 4: 1.308241

Set. 5: 1.583888

Set. 6: 1.859535

Set. 7: 2.135182

Set. 8: 2.410829

Set. 9: 2.686476

Set. 10: 2.96212

Set. 11: 3.23777

Set. 12: 3.513417

Set. 13: 3.789064

Set. 14: 4.064711

Set. 15: 4.340358

Set. 16: 4.616005