WEEK 4

BY: SAKSHAM RAO

1. Introduction / Background

This week's task explores transistor-level CMOS circuit behaviour via SPICE simulations using the SKY130 process models. The aim is to bridge device physics (MOSFET characteristics, sizing, variation) with digital timing fundamentals (delay, noise margin, slack) as used in static timing analysis (STA).

Specifically:

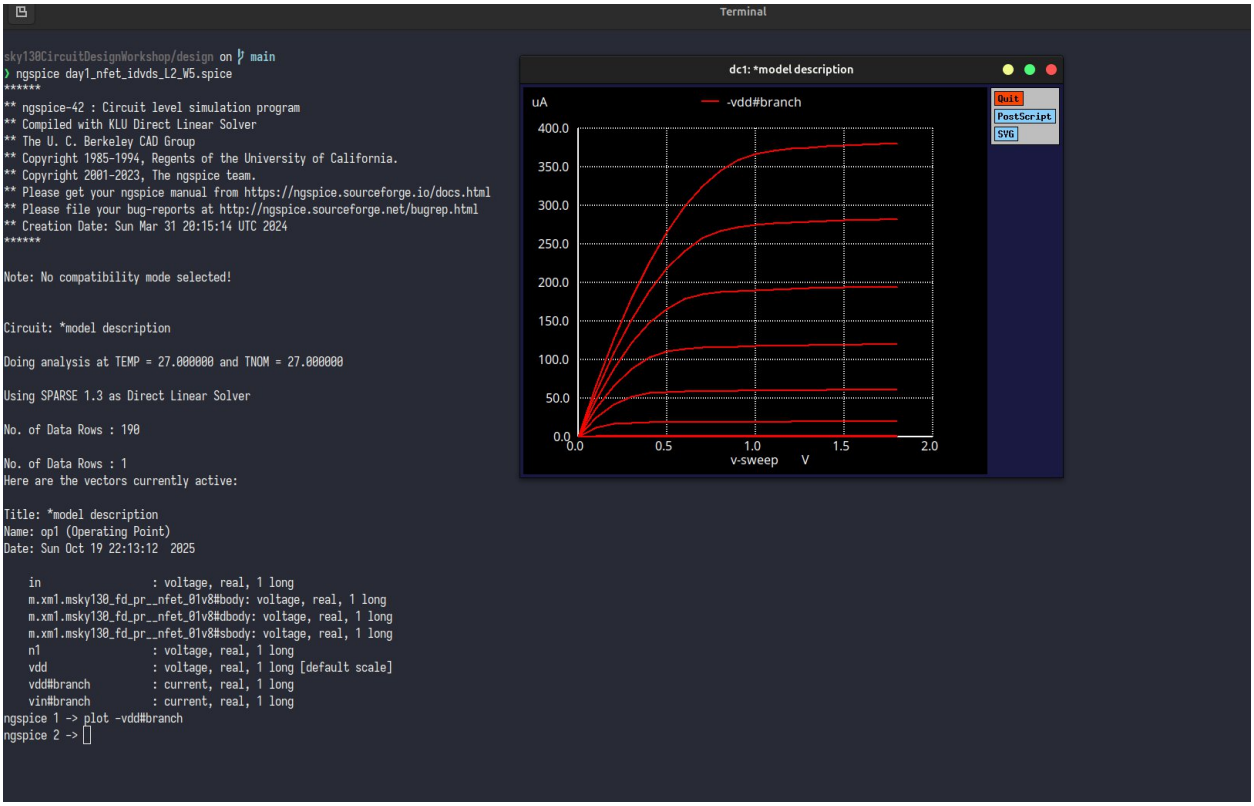
- We first examine how a MOSFET transitions between linear and saturation regions (Id vs Vds).
- We extract the threshold voltage (Vt) from Id vs Vgs, and observe velocity-saturation effects in short-channel devices.
- We build a CMOS inverter and sweep its input to obtain the voltage transfer characteristic (VTC), identifying the switching threshold (Vm).
- We then perform transient simulation on the inverter to measure rise and fall propagation delays.
- Noise margins are determined from the VTC (VOL, VOH, VIL, VIH → NML, NMH).
- Finally we study variation: supply-voltage (Vdd) changes and transistor sizing changes (W/L) to see how inverter behaviour and timing margins shift.

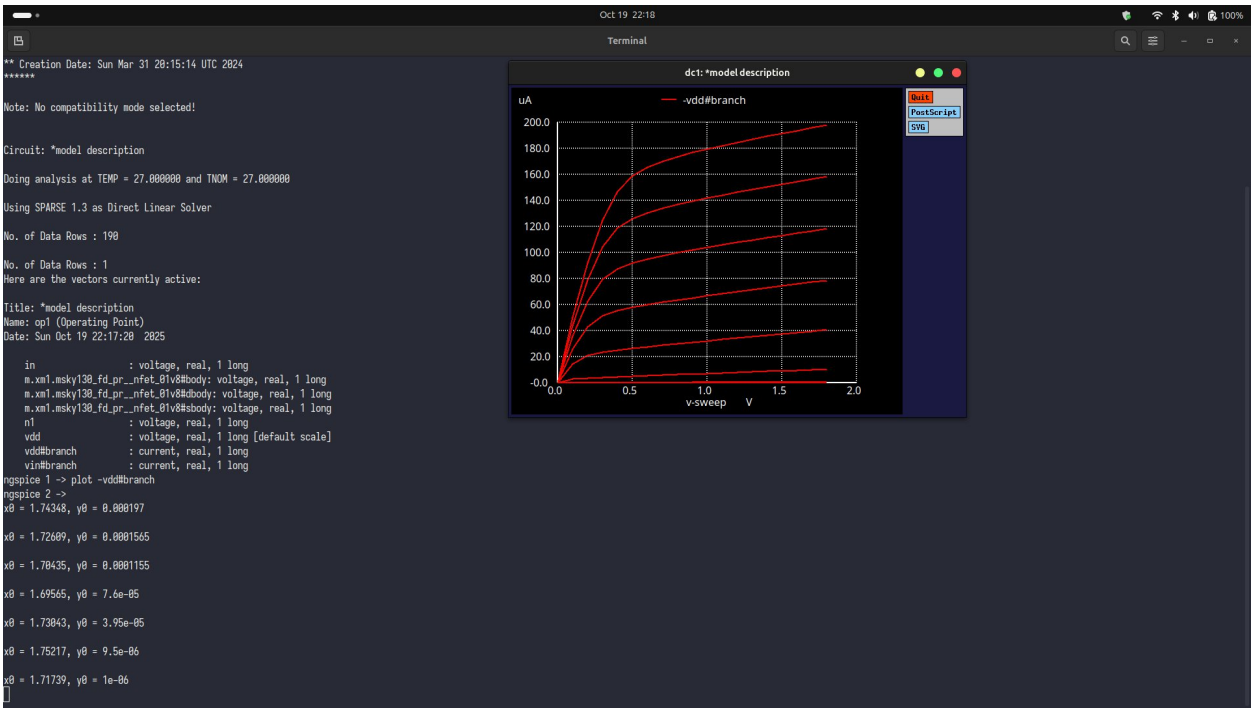
  By doing this, we gain intuition into how device-level variations propagate up into circuit timing and margin budgets used in STA.

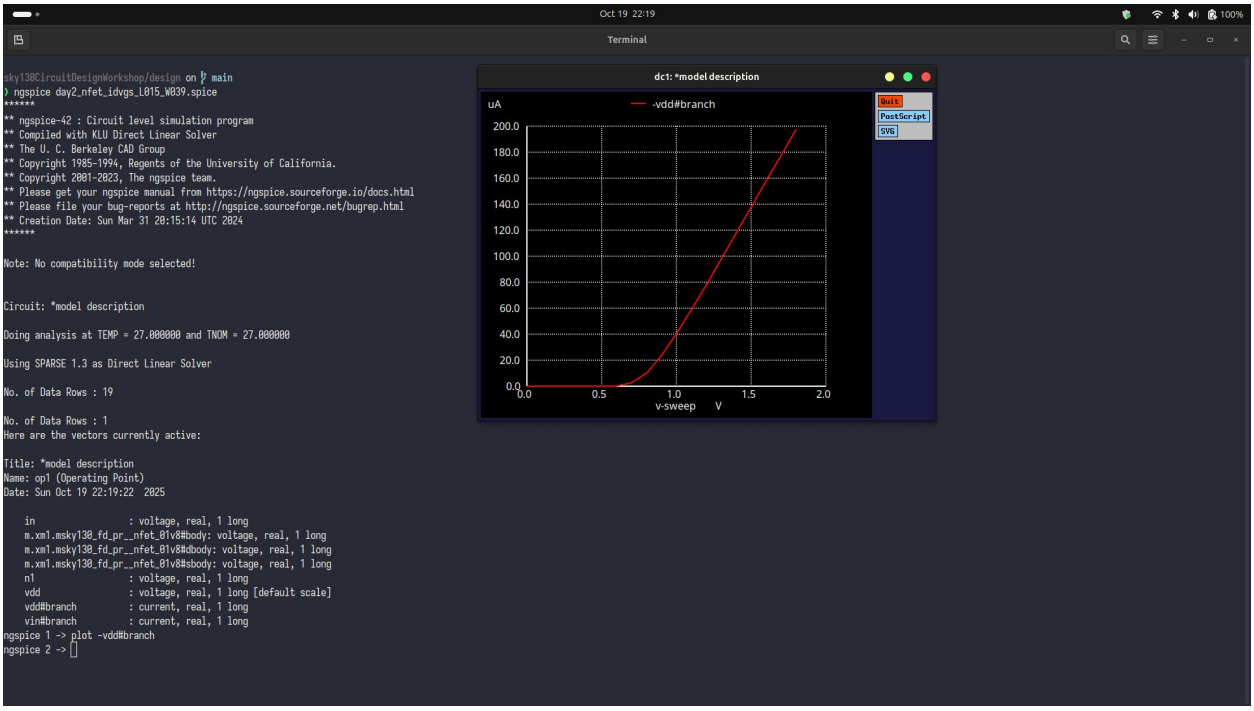
2. SPICE Netlist / Code References

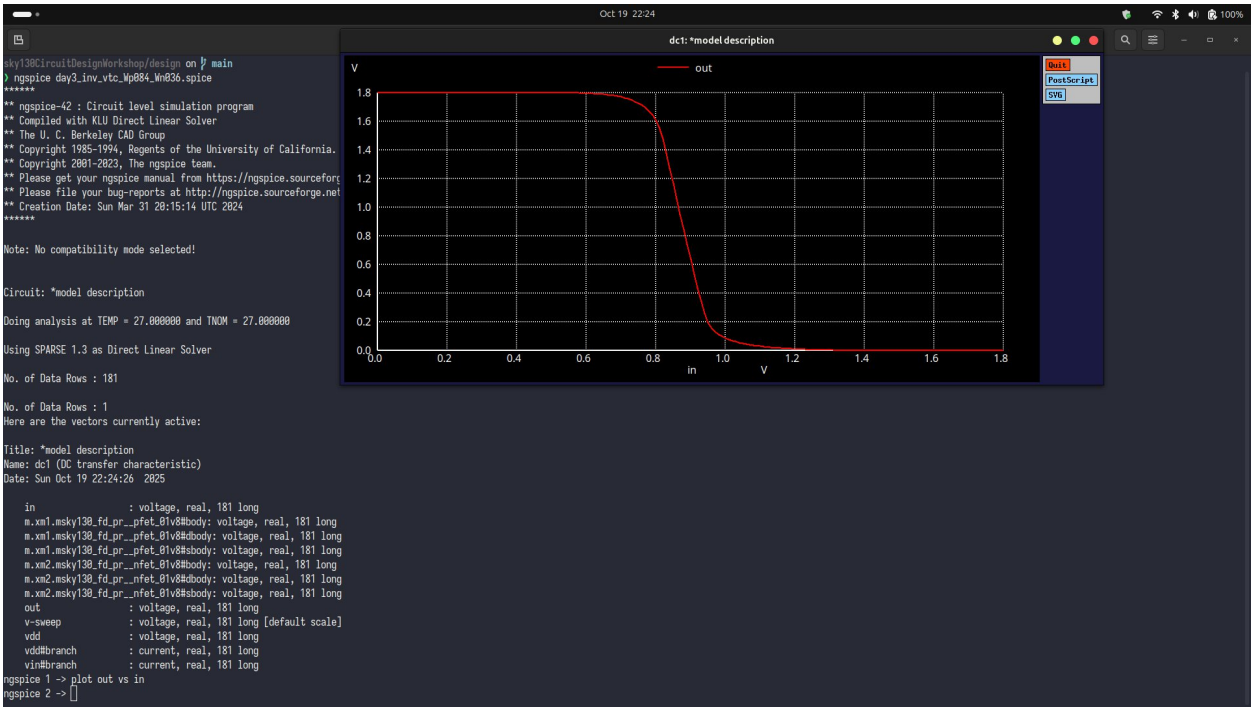
All nfet and pfet relevant to lab accessed directly from the github.

3. Plots & Figures

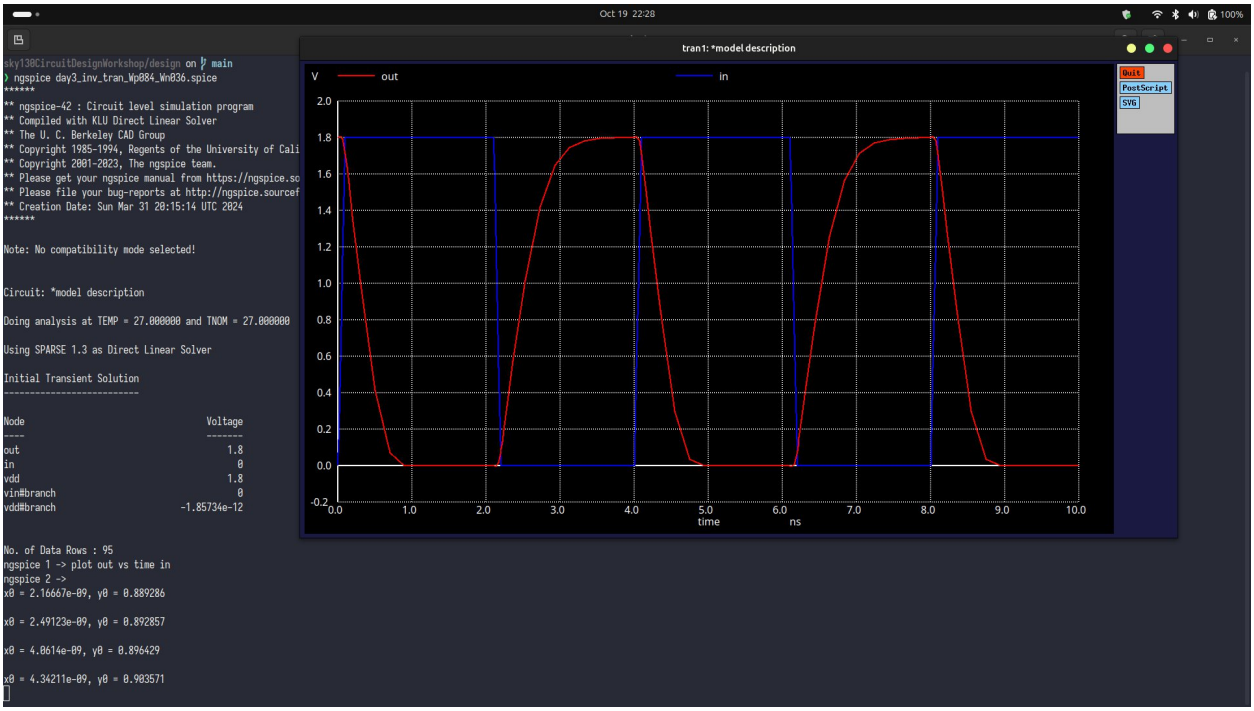


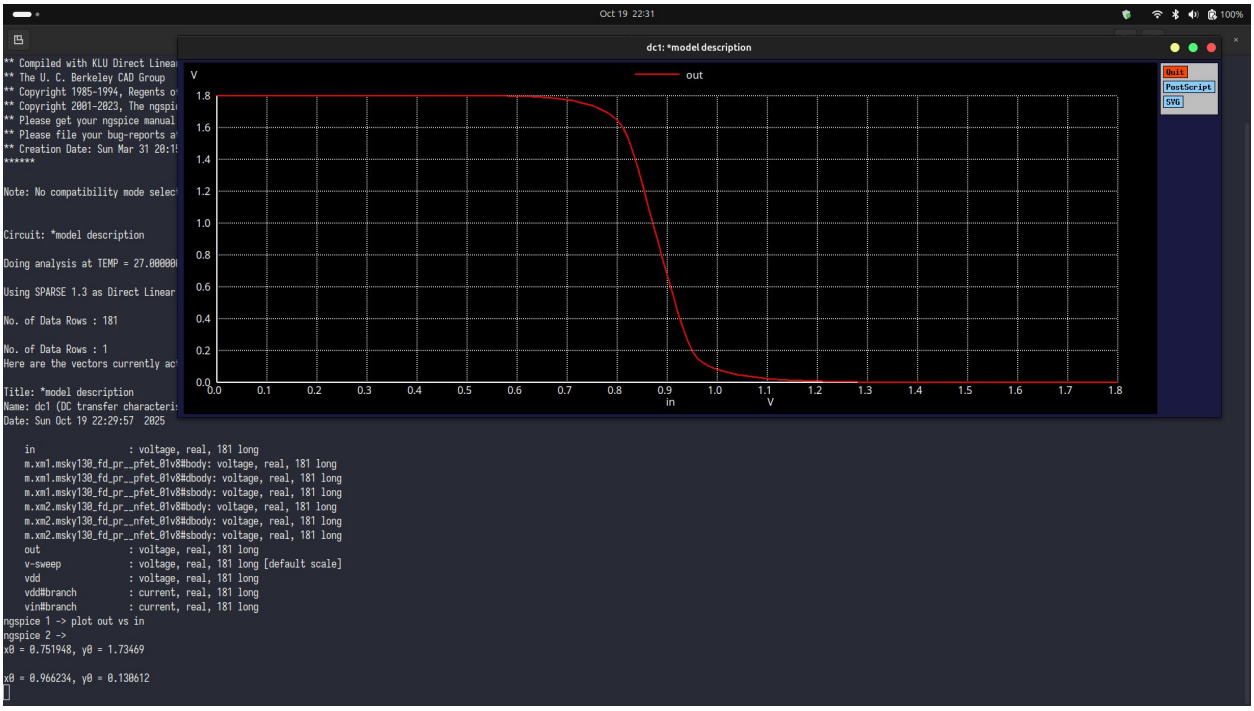


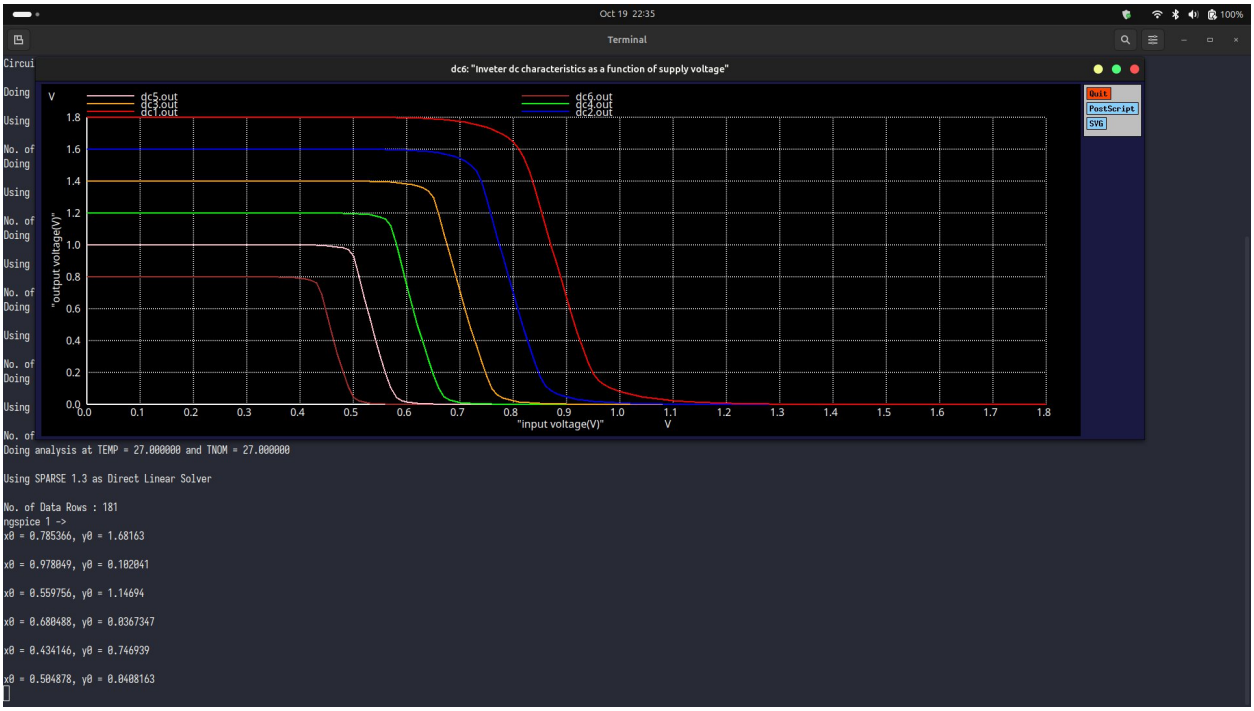

Oct 19 22:19
Terminal
sky130CircuitDesignWorkshop/design on main
ngspice day2_nfet_idvgs_L015_W039.spice
******
** ngspice-42 : Circuit level simulation program
** Compiled with KLU Direct Linear Solver
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Copyright 2001-2023, The ngspice team.
** Please get your ngspice manual from https://ngspice.sourceforge.io/docs.html
** Please file your bug-reports at http://ngspice.sourceforge.net/bugrep.html
** Creation Date: Sun Mar 31 20:15:14 UTC 2024
******
Note: No compatibility mode selected!
Circuit: *model description
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Using SPARSE 1.3 as Direct Linear Solver
No. of Data Rows : 19
No. of Data Rows : 1
Here are the vectors currently active:
Title: *model description
Name: op1 (Operating Point)
Date: Sun Oct 19 22:19:22 2025
in : voltage, real, 1 long
m.xm1.msky130_fd_pr__nfet_01v8#body: voltage, real, 1 long
m.xm1.msky130_fd_pr__nfet_01v8#dbody: voltage, real, 1 long
m.xm1.msky130_fd_pr__nfet_01v8#sbody: voltage, real, 1 long
n1 : voltage, real, 1 long
vdd : voltage, real, 1 long [default scale]
vdd#branch : current, real, 1 long
vin#branch : current, real, 1 long
ngspice 1 -> plot -vdd#branch
ngspice 2 ->
dc1: *model description
uA
-vdd#branch
Quit
PostScript
SVG
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
0.0
0.5
1.0
1.5
2.0
v-sweep V

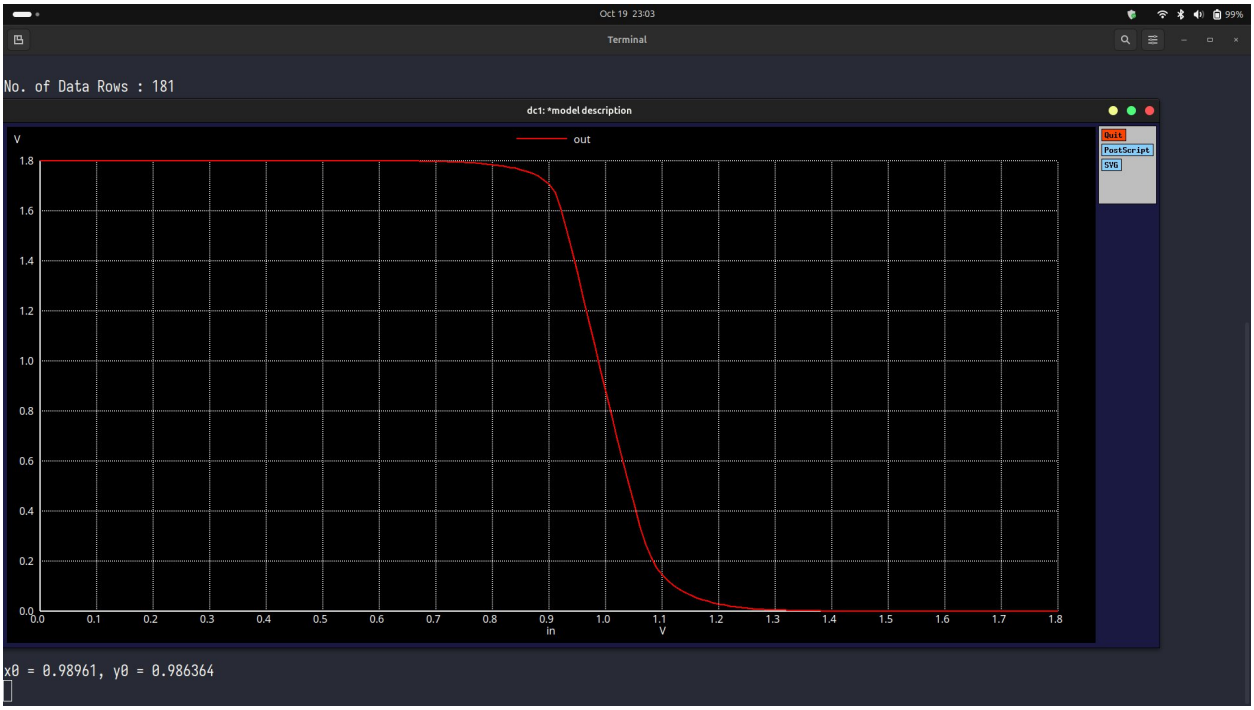

Oct 19 22:24
sky130CircuitDesignWorkshop/design on main
ngspice day3_inv_vtc_Wp084_Wn036.spice
******
** ngspice-42 : Circuit level simulation program
** Compiled with KLU Direct Linear Solver
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of California.
** Copyright 2001-2023, The ngspice team.
** Please get your ngspice manual from https://ngspice.sourceforg
** Please file your bug-reports at http://ngspice.sourceforge.net
** Creation Date: Sun Mar 31 20:15:14 UTC 2024
******
Note: No compatibility mode selected!
Circuit: *model description
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Using SPARSE 1.3 as Direct Linear Solver
No. of Data Rows : 181
No. of Data Rows : 1
Here are the vectors currently active:
Title: *model description
Name: dc1 (DC transfer characteristic)
Date: Sun Oct 19 22:24:26 2025
in : voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#body: voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#dbody: voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#sbody: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#body: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#dbody: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#sbody: voltage, real, 181 long
out : voltage, real, 181 long
v-sweep : voltage, real, 181 long [default scale]
vdd : voltage, real, 181 long
vdd#branch : current, real, 181 long
vin#branch : current, real, 181 long
ngspice 1 -> plot out vs in
ngspice 2 ->
dc1: *model description
V
out
Quit
PostScript
SVG
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
in V


Oct 19 22:26
dc1: *model description
out
Quit
PostScript
SVG
mV
885.0
880.0
875.0
870.0
865.0
860.0
855.0
850.0
845.0
840.0
835.0
862.0
864.0
866.0
868.0
870.0
872.0
874.0
876.0
878.0
880.0
882.0
884.0
886.0
888.0
in
mV
x0 = 0.78317
dx = 0.17383
dy/dx = -2.6
x0 = 0.82056
dx = 0.12149
dy/dx = -3.1
x0 = 0.87697
x0 = 0.87677
x0 = 0.87684
x0 = 0.87684
x0 = 0.87677
x0 = 0.876743, y0 = 0.878611


Oct 19 22:28
tran1: *model description
V
out
in
Quit
PostScript
SVG
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
time
ns
sky130CircuitDesignWorkshop/design on main
ngspice day3_inv_tran_Wp084_Wn036.spice
** ngspice-42 : Circuit level simulation program
** Compiled with KLU Direct Linear Solver
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of Cali
** Copyright 2001-2023, The ngspice team.
** Please get your ngspice manual from https://ngspice.sc
** Please file your bug-reports at http://ngspice.sourcef
** Creation Date: Sun Mar 31 20:15:14 UTC 2024
Note: No compatibility mode selected!
Circuit: *model description
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Using SPARSE 1.3 as Direct Linear Solver
Initial Transient Solution
Node Voltage
out 1.8
in 0
vdd 1.8
vin#branch 0
vdd#branch -1.85734e-12
No. of Data Rows : 95
ngspice 1 -> plot out vs time in
ngspice 2 ->
x0 = 2.16667e-09, y0 = 0.889286
x0 = 2.49123e-09, y0 = 0.892857
x0 = 4.0614e-09, y0 = 0.896429
x0 = 4.34211e-09, y0 = 0.903571


Oct 19 22:31
dc1: *model description
out
V
in
V
Quit
PostScript
SVG
** Compiled with KLU Direct Linear
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents o
** Copyright 2001-2023, The ngspi
** Please get your ngspice manual
** Please file your bug-reports a
** Creation Date: Sun Mar 31 20:1
******
Note: No compatibility mode selec
Circuit: *model description
Doing analysis at TEMP = 27.00000
Using SPARSE 1.3 as Direct Linear
No. of Data Rows : 181
No. of Data Rows : 1
Here are the vectors currently ac
Title: *model description
Name: dc1 (DC transfer characteri
Date: Sun Oct 19 22:29:57 2025
in : voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#body: voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#dbody: voltage, real, 181 long
m.xm1.msky130_fd_pr__pfet_01v8#sbody: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#body: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#dbody: voltage, real, 181 long
m.xm2.msky130_fd_pr__nfet_01v8#sbody: voltage, real, 181 long
out : voltage, real, 181 long
v-sweep : voltage, real, 181 long [default scale]
vdd : voltage, real, 181 long
vdd#branch : current, real, 181 long
vin#branch : current, real, 181 long
ngspice 1 -> plot out vs in
ngspice 2 ->
x0 = 0.751948, y0 = 1.73469
x0 = 0.966234, y0 = 0.130612


Oct 19 22:35
Terminal
dc6: "Inveter dc characteristics as a function of supply voltage"
dc5.out
dc3.out
dc1.out
dc6.out
dc4.out
dc2.out
"output voltage(V)"
"input voltage(V)"
V
Quit
PostScript
SVG
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Using SPARSE 1.3 as Direct Linear Solver
No. of Data Rows : 181
ngspice 1 ->
x0 = 0.785366, y0 = 1.68163
x0 = 0.978049, y0 = 0.102041
x0 = 0.559756, y0 = 1.14694
x0 = 0.680488, y0 = 0.0367347
x0 = 0.434146, y0 = 0.746939
x0 = 0.504878, y0 = 0.0408163


Oct 19 23:03
Terminal
No. of Data Rows : 181
dc1: *model description
V
out
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8
in
V
Quit
PostScript
SVG
x0 = 0.98961, y0 = 0.986364

4. Tabulated Results

Below is a template table

Experiment	Condition	Extracted Parameter	Symbol	Value	Unit	Notes
NMOS Threshold Voltage	W=5 µm, L=2 µm, Vdd=1.8V	Vt	Vt	0.767	V	Method: √Id extrapolation / constant current
Inverter Switching Threshold	Wp=0.84 µm, Wn=0.39 µm, Vdd=1.8 V	Vm	Vm	0.876	V	Vin = Vout crossing
Inverter Rise Delay	Wp=0.84 µm, Wn=0.39 µm, Vdd=1.8 V	tpLH	tpLH	0.33	ns	50% input to 50% output
Inverter Fall Delay	same	tpHL	tpHL	0.28	ns	50% input to 50% output
Noise Margin Low	VTC at Vdd=1.8V	NML = VIL – VOL	NML	0.768	V	
Noise Margin High	same	NMH = VOH – VIH	NMH	0.621	V	
Device Variation: Switching Voltage	Wp=7 µm, Wn=0.42 µm, Vdd=1.8 V	Vm	Vm	0.989	V	Vin = Vout crossing

5. Observations / Analysis

A. MOSFET Id vs Vds

- At low Vds, Id increases approximately linearly with Vds; beyond a certain Vds, Id saturates (becomes nearly flat) for each Vgs. For higher Vgs values, the saturation current is higher. There may be a slight slope in saturation region representing channel-length modulation.
- In the linear region the channel is fully formed and behaves like a resistor; once Vds exceeds (Vgs−Vt), the channel pinches off at the drain end and further increase in Vds gives only modest increase in current (saturation). In short-channel devices, velocity saturation and channel shortening reduce the slope, and the remaining slope is due to channel-length modulation (finite output resistance).

- The saturation region current sets the drive strength of the device. The finite output resistance (r_out) in saturation means that the device cannot be considered ideal switch — it will impact how fast a node can be driven (i.e., its delay) and how large the load it can drive. Variation in this behaviour translates into variation of propagation delay and slack margin in STA.

B. NMOS Id vs Vgs (Threshold Extraction)

- **:** The Id vs Vgs curve shows a threshold-like kink where current starts to rise sharply. On a √Id vs Vgs plot (for long-channel behaviour) you can extrapolate the linear portion back to zero to get Vt. In a short-channel device you may see a more gradual increase due to velocity saturation.
- **:** Vt is the gate-voltage at which a strong inversion channel forms and significant current begins to flow. In short-channel devices velocity saturation occurs when the carriers cannot gain further velocity despite higher lateral field; this flattens the I–V rise and complicates threshold definition.
- Vt variation (due to process, temperature, voltage) shifts the device's drive strength and switching point of gates. In timing paths, a higher Vt usually slows devices (lower current) and may increase slack negative margin, while a lower Vt increases leakage and may affect noise margins.

C. CMOS Inverter VTC

- The VTC (Vout vs Vin) shows a steep transition region where the inverter switches, typically around some Vm point (where Vin = Vout). The gain (dVout/dVin) is high around that region. On the left region (Vin low) Vout is near Vdd (PMOS on), on right region (Vin high) Vout is near 0 (NMOS on).
- As input increases, the NMOS begins to conduct and the PMOS begins to turn off; at the crossover point both devices conduct significantly so the output is dragged from high to low (or vice versa). The sizing ratio (Wp:Wn) and the device characteristics determine the exact switching point Vm.
- Vm effectively sets the logic threshold of the gate and influences when the next stage sees a valid "0" or "1" transition. The steepness of VTC (gain) affects noise margin and the robustness of logic levels; thus better understanding of VTC helps in margin budgeting in timing and noise analysis.

D. Transient Simulation – Rise/Fall Delays

- The output waveform follows the input pulse with some delay; the time between the input crossing 50% and output crossing 50% defines tpLH for rising transition and tpHL for falling transition. Often one of tpLH/tpHL is longer, e.g., rise may be slower.
- The delay arises due to the RC charging/discharging of the load through the MOSFETs; PMOS and NMOS have different mobilities (holes slower), meaning the rise (through PMOS) often is slower than the fall (through NMOS) unless sizing compensates. Capacitances (load, parasitic) and drive current define the delay.

- Propagation delay is key for STA: launch-to-capture timing, clock period budgeting, and slack calculation. Understanding factors affecting delay (loading, sizing, supply) gives better insight on how to optimise critical paths.

E. Noise Margin / Robustness Analysis

- From the VTC you can extract VOL (output low), VOH (output high), VIL and VIH (input threshold points where gain = −1 or some defined point). Then noise margins NML and NMH quantify how much noise the gate can tolerate.
- The logic levels and transition region define how resilient the logic gate is to input noise or variation. A larger noise margin means more robust logic. Sizing, threshold voltages, and supply affect these margins.
- In timing and sign-off, noise margins are akin to margin budgets — if noise or variation consumes margin, then slack is reduced. Robust logic design means you must allow margin not only for timing but also for noise/variations.

F. Power-Supply and Device Variation Studies

- Varying Vdd shifts the VTC (Vm moves, gain may reduce), and increases delay (at lower Vdd the current is less). Changing transistor sizes (e.g., W/L ratio) shifts Vm and affects drive strength and hence delay. Noise margins similarly change.
- Supply voltage affects overdrive (Vgs−Vt) and hence current; smaller supply ⇒ lower drive current ⇒ slower device. Sizing changes affect ratio of PMOS/NMOS strengths and thus switching threshold. Variations in process (device size, mobility, threshold) propagate to circuit behaviour.
- STA typically considers PVT corners (process, voltage, temperature) explicitly. These variation studies at the transistor/circuit level show why corners exist and how slack/critical path margin can degrade under worst-case conditions. Understanding this physically helps anticipate where timing margin may vanish or variation may dominate.

6. Conclusions

- This set of SPICE- based experiments show clearly how transistor-level phenomena (threshold voltage, velocity saturation, channel length modulation, sizing and supply variation) affect circuit-level timing and noise behaviour.
- The behaviour we measured (switching thresholds, delays, noise margins) are the building blocks of what STA approximates when analysing large digital designs. By seeing the physical device behaviour, we gain better intuition for timing budgets, margin setting, and variation impacts.
- Device sizing, supply voltage, and process variation all contribute to timing slack tightening – this underscores the importance of conservative margins in digital design.
- In real design flows (such as considered in the sky130 workshop), these principles underlie how standard-cells are characterised, how timing models are built, and how PVT corners are defined.

- Overall, bridging SPICE simulation and STA concepts strengthens your understanding of how “what happens in the transistor” affects “what happens in the digital circuit and timing path”.

7. References

- Repository: sky130CircuitDesignWorkshop (GitHub) — https://github.com/kunalg123/sky130CircuitDesignWorkshop (GitHub)