

Add to <https://github.com/pramchan/VSD-CMOS> after final edits and fixes after feedback from VSD.

# RP-VSD-CMOS-2025-Dec-Submission

<b>1.0 VSD-COMS Lab preparation</b>	<b>2</b>
<b>Lab 1- nfet M nfet Diode Channel length 2 and width 5 um</b>	<b>7</b>
<b>day1_nfet_idvds_L2_W5.spice</b>	<b>7</b>
<b>Lab 2.1 id-vds</b>	<b>10</b>
Lab2.2 id-vgv	13
<b>Lab 3.1 Trans</b>	<b>16</b>
Lab 3.2 inv vtc wp84 wn36 L15	17
Repeating lab 3.2 for delay calculations on vtc plot:	19
<b>Lab 4 inv-nm-wp1-wn36-L15</b>	<b>20</b>
<b>Lab 5.1 Device variation wp7-wn42 L15</b>	<b>21</b>
Lab 5.2 Supply variation	22

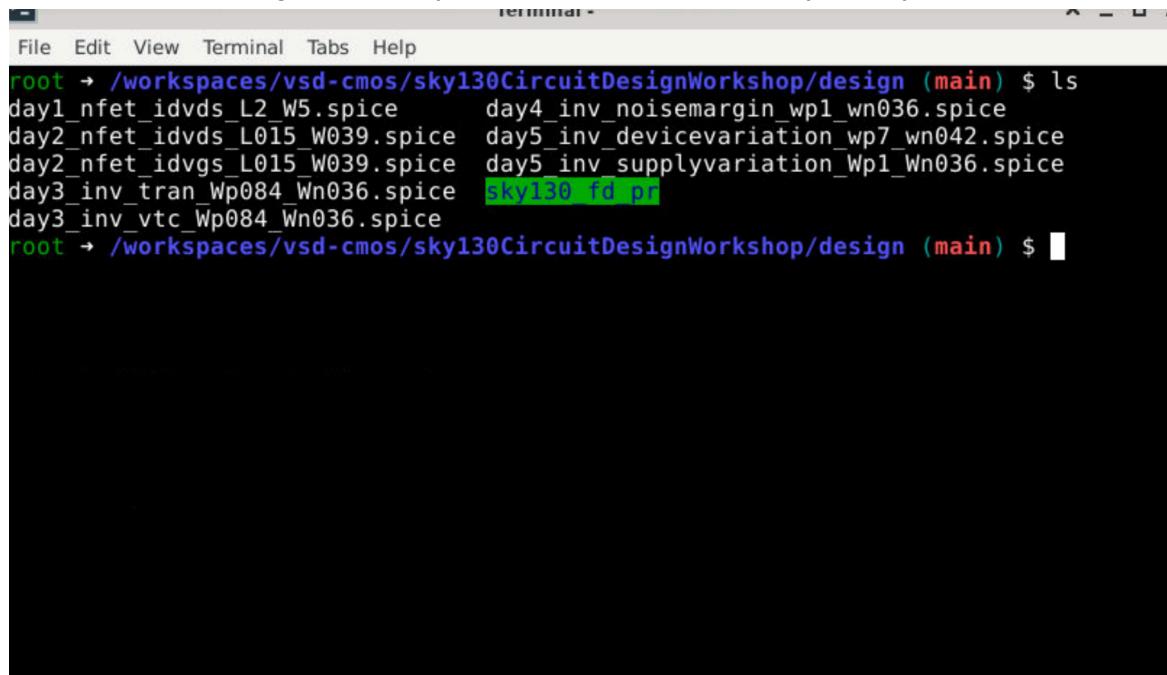
# 1.0 VSD-COMS Lab preparation

The Lab work needs to follow certain objectives and accordingly the learnings are as part of theory and practice to follow.

Theoretically the VLSI design requires knowledge of many things and that besides the revised tools like ngspice and system level use of Linux and better the latest Ubuntu 24.04.xx LTS.

Do not get distracted by VMs and Windows and waste your time. Please use github codespace directly.

Here is the main image of where you need to position, before you do your labs.



A screenshot of a terminal window titled "terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal prompt shows the user is in root mode at the path "/workspaces/vsd-cmos/sky130CircuitDesignWorkshop/design (main)". The user runs the command "ls" which lists several files: "day1\_nfet\_idvds\_L2\_W5.spice", "day2\_nfet\_idvds\_L015\_W039.spice", "day2\_nfet\_idvgs\_L015\_W039.spice", "day3\_inv\_tran\_Wp084\_Wn036.spice", "day4\_inv\_noisemargin\_wpl\_wn036.spice", "day5\_inv\_devicevariation\_wp7\_wn042.spice", and "day5\_inv\_supplyvariation\_Wp1\_Wn036.spice". The file "sky130\_fd\_pr" is highlighted in green. The terminal prompt ends with "\$".

**Figure 1. The base path to run your design labs**

To reach here please do the following.

- 1) Create your github account if you do not have one or want to use the vsdid offer of course <https://github.com/vsdip/vsd-cmos> Skip 1) and go to 2) to use your own or vsdid github account.

## Step-by-Step Guide

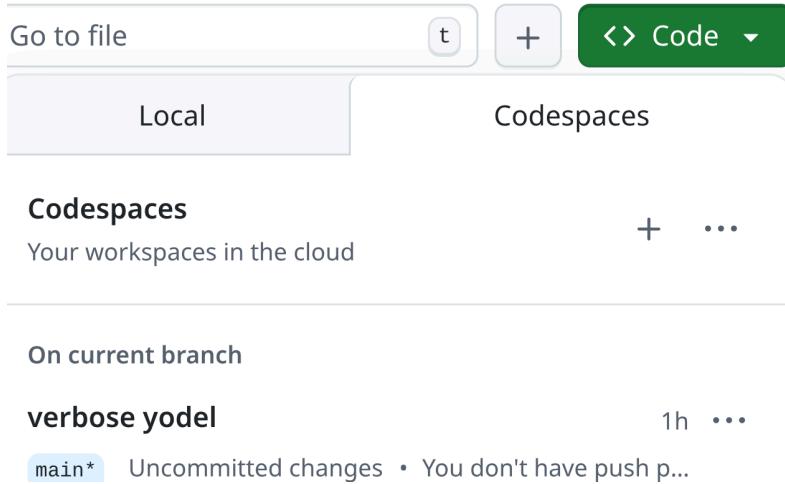
1. Go to the GitHub website: Open a web browser and navigate to the [GitHub sign-up page](#) at [github.com/signup](https://github.com/signup).

2. Enter your email address: Follow the prompts to enter your email address and then click Continue.
3. Create a strong password: Enter a secure password for your new account. GitHub requires your password to be at least 15 characters long, or at least 8 characters including a number and a lowercase letter. Click Continue.
4. Choose a username: Enter a unique username. This will be your identity on GitHub and part of your profile URL, so choose it carefully. Usernames can only contain alphanumeric characters or single hyphens and cannot begin or end with a hyphen. Click Continue.
5. Opt for email updates (optional): You can choose whether you want to receive occasional product updates and announcements via email. Type 'y' for yes or 'n' for no when prompted, and then click Continue.
6. Verify your account: Complete the brief verification puzzle to confirm you are a human.
7. Click "Create account": Once verification is complete, a Create account button will appear. Click it to proceed.
8. Enter the launch code: GitHub will send a launch code to the email address you provided. Check your inbox, copy the code, and paste it into the field on the GitHub page to verify your email address.
9. Personalize your experience (optional): You may be asked some questions about your experience level and how you plan to use GitHub (e.g., number of team members, if you are a student or teacher). You can answer these or skip this step to go straight to your dashboard.

Once these steps are completed, your free personal GitHub account is ready to use. It is highly recommended to configure two-factor authentication (2FA) for enhanced security.

2) Once on your own with a repository cloning [github.com/vsdid/vsd-cmos](https://github.com/vsdid/vsd-cmos) on your account follow the procedure as below. Note this procedure is valid for both your own vsdid who pays for use of it during the course.

- a) Go and click on the browser on + sign below <>CODE> button for creating a Codespace in the cloud.



**Figure 2.** The [github.com](#) starting point for creating codespaces

- b) It will create a new tab something like  
<https://legendary-giggle-pgwv77qg5xh6rj.github.dev/>
- c) Wait for 10-15 mts it will invoke a new tab with following reading  
Run **Sky130 CMOS design labs** on GitHub Codespaces — entirely in your browser, with **ngspice**, **noVNC GUI desktop**, and ready-to-use **SPICE simulation decks**. No local setup required — everything runs inside your browser tab.
- d) It will next create a new tab trying to start the desktop for your labs

# Directory listing for /

---

- [app/](#)
  - [core/](#)
  - [include/](#)
  - [utils/](#)
  - [vendor/](#)
  - [vnc.html](#)
  - [vnc auto.html@.](#)
  - [vnc lite.html](#)
- 

**Figure 3 : The browser screen that leads you to VNC (named noVNC a product name)**

Now click on the vnc.html link and you will get the desktop for your lab.

You will see in right tab of console something like this

Your application running on port 5900 is available. [See all forwarded ports](#)

[You will see a button called noVNC connect on the new tab which you click to get to the desktop on the same tab.](#)

Now focus on the desktop which has two applications icons, Ignore them.

[Applications](#)

[Run Program...](#)

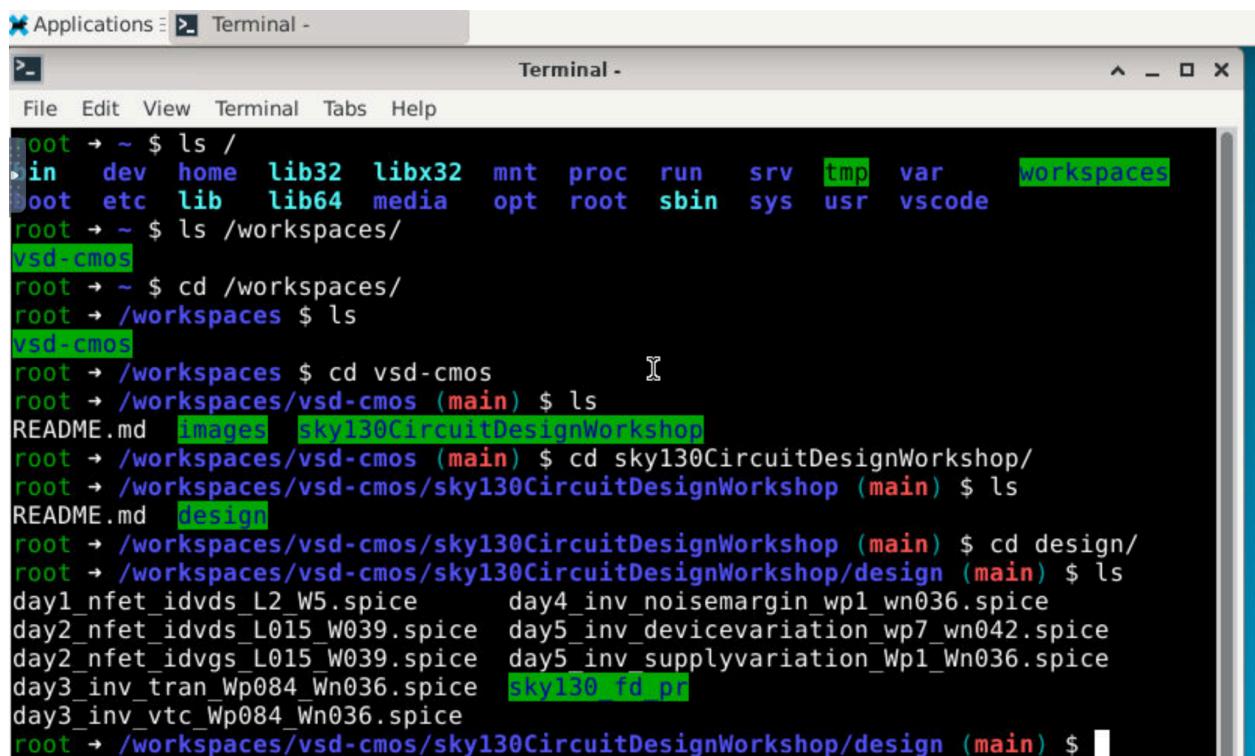
[Terminal Emulator](#)

[File Manager](#)

[Mail Reader](#)

Web Browser  
Settings  
Accessories  
Internet  
System  
About Xfce  
Log Out

Click on applications and invoke Terminal emulation which will create a unix terminal for you to run labs.



The screenshot shows a terminal window titled "Terminal -". The terminal is running as root and displays the following command-line session:

```
root ~ $ ls /
bin dev home lib32 libx32 mnt proc run srv tmp var workspaces
root ~ $ ls /workspaces/
vsd-cmos
root ~ $ cd /workspaces/
root ~ /workspaces $ ls
vsd-cmos
root ~ /workspaces $ cd vsd-cmos
root ~ /workspaces/vsd-cmos (main) $ ls
README.md images sky130CircuitDesignWorkshop
root ~ /workspaces/vsd-cmos (main) $ cd sky130CircuitDesignWorkshop/
root ~ /workspaces/vsd-cmos/sky130CircuitDesignWorkshop (main) $ ls
README.md design
root ~ /workspaces/vsd-cmos/sky130CircuitDesignWorkshop (main) $ cd design/
root ~ /workspaces/vsd-cmos/sky130CircuitDesignWorkshop/design (main) $ ls
day1_nfet_idvds_L2_W5.spice day4_inv_noisemargin_wp1_wn036.spice
day2_nfet_idvds_L015_W039.spice day5_inv_devicevariation_wp7_wn042.spice
day2_nfet_idvgs_L015_W039.spice day5_inv_supplyvariation_Wp1_Wn036.spice
day3_inv_tran_Wp084_Wn036.spice sky130_fd.pr
day3_inv_vtc_Wp084_Wn036.spice
root ~ /workspaces/vsd-cmos/sky130CircuitDesignWorkshop/design (main) $
```

**Figure 4. This is your lab start point**

Now two things to note.

Must read each lab code before running them using the “ngspice” tool to execute the code to view and record the reports per lab.

# Lab 1- nft M nfet Diode Channel length 2 and width 5 um

day1\_nfet\_idvds\_L2\_W5.spice

Eg, code for lab1 is seen using vim editor as follows

```
$vim day1_nfet_idvds_L2_W5.spice
```

Model Description

```
.param temp=27
```

\*Including sky130 library files

```
.lib "sky130_fd_pr/models/sky130.lib.spice" tt
```

\*Netlist Description

```
XM1 Vdd n1 0 0 sky130_fd_pr_nfet_01v8 w=5 l=2
```

```
R1 n1 in 55
```

Vdd vdd 0 1.8V

Vin in 0 1.8V

\*simulation commands

```
.op  
.dc Vdd 0 1.8 0.1 Vin 0 1.8 0.2
```

```
.control
```

```
run  
display  
setplot dc1  
.endc
```

```
.end
```

Understand all lab codes have common

Model description => the temperature for simulation is 27 degree Celcius  
 .param temp=27  
 \*Including sky130 library files (your base is on 'design' folder where you run codes & library of spice containing models describing in this case cmos is as shown in relative path/file under .lib with tt the top corner  
 .lib "sky130\_fd\_pr/models/sky130.lib.spice" tt

Notes:

Refer for Spice Parameters - <https://www.seas.upenn.edu/~jan/spice/spice.MOSparamlist.html>  
 Other details: <https://www.seas.upenn.edu/~jan/spice/spice.overview.html#MOSFETS>  
 Colorado chapter 7 not available - [Fundamentals of Power Electronics | Springer Nature Link \(formerly SpringerLink\)](#) but can read this -  
<https://ngspice.sourceforge.io/docs/ngspice-manual.pdf> and  
<https://www.scribd.com/document/475250670/Sect7-1-pdf>

\*Netlist Description here shows 4 nodes starting device XM1 the eXtended MOS and definition in .Lib as Vdd followed by nodes DGSS (drain, gate,source & substrate) in the circuit with XMOS device nfet of 01v8 (1.8v) and channel with 5 and channel length 2 um.  
 The second line shows specs for R1 between n1 and in nodes of 55 Ohms.  
 The third line of Vdd Voltage between vdd & 0 (Block/earth or substrate surface or earth) of 1.8V.  
 The fourth line Vin between nodes in and 0 (Block) as 1.8V

```
XM1 Vdd n1 0 0 sky130_fd_pr_nfet_01v8 w=5 l=2
R1 n1 in 55
Vdd vdd 0 1.8V
Vin in 0 1.8V
```

This Netlist describes the “ngspice” simulator about the circuit elements to fetch from library and their characteristics to run the simulation and next section to the operations and control to display the results of simulation of day1\_nfet\_idvds\_L2\_W5.spice

\*simulation commands

```
.op
.dc Vdd 0 1.8 0.1 Vin 0 1.8 0.2
.control
run
display
setplot dc1
.endc
```

```
.end
```

After running

```
$ ngspice day1_nfet_idvds_L2_W5.spice
```

We can plot using command

```
$plot - vdd#block
```

The result is a plot of the above in another tab as pop-up on the desktop tab. Therefore make sure your browser (chrome, mozilla wherever must allow pop-ups from [github.com](https://github.com) or if not for this application to run go to browser setup to enable specific or all for the duration of running the simulation codes and disable them later after the lab work)

Here is a sample report of lab1.

Lab1 id vs vds for `day1_nfet_idvds_L2_W5.spice` with sweeps for dc1 plot dc Vdd 0 1.8 0.1 Vin 0 1.8 0.2

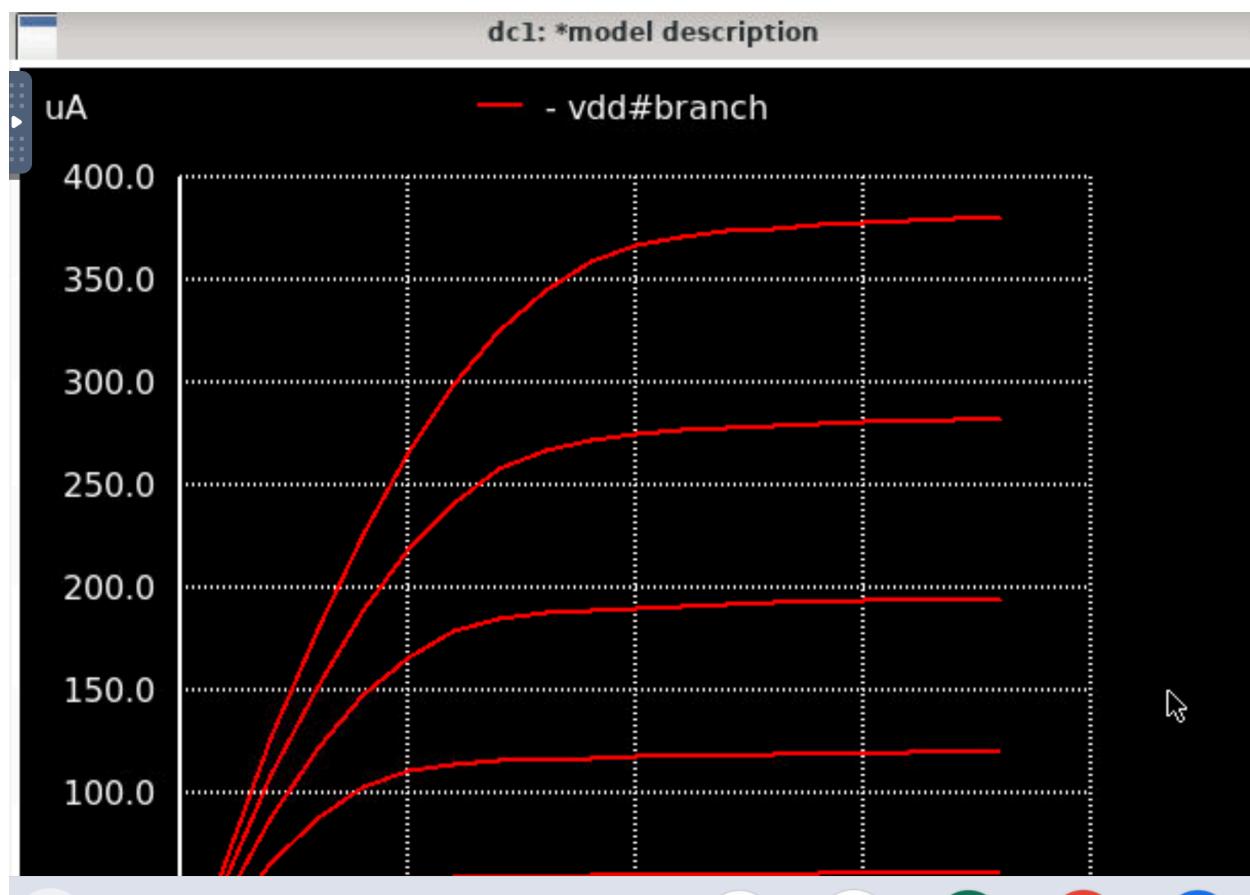


Figure 5: Plot Lab1 for nfet id-vds L2\_W5

## Lab 2.1 id-vds

day2\_nfet\_idvds\_L015\_W039.spice

```
XM1 Vdd n1 0 0 sky130_fd_pr_nfet_01v8 w=0.39 l=0.15
R1 n1 in 55
Vdd vdd 0 1.8V
Vin in 0 1.8V
```

```
.op
.dc Vdd 0 1.8 0.1 Vin 0 1.8 0.2
run
display
setplot dc1
```

General Theoretic Formulas for  $V_t$  (threshold voltage at  $V_{gs}$  for different  $I_d$  currents output vs  $V_{in}$  ( $V_{gs}$ ) sweep for various  $V_{dd}$  curves above and the nonlinear, linear and saturation regions with respective parameters.

Threshold Voltage Equation:

$$V_t = V_{to} + \gamma (\sqrt{|-2\Phi_f + V_{sb}|} - \sqrt{|-2\Phi_f|})$$
$$\gamma = \frac{\sqrt{2qNA\varepsilon_{si}}}{C_{ox}}$$

$$\Phi_f = -\Phi_T * \ln \frac{N_A}{n_i}$$

Linear region:

$$I_d = kn \cdot [(V_{gs} - V_t)V_{ds} - \frac{V_{ds}^2}{2}]$$

Saturation region:

$$I_d = \frac{Kn'}{2} \cdot \frac{W}{L} \cdot (V_{gs} - V_t)^2 [1 + \lambda V_{ds}]$$

Fig 6. Overall MOSFET device parameters and empirical formulas for emulation

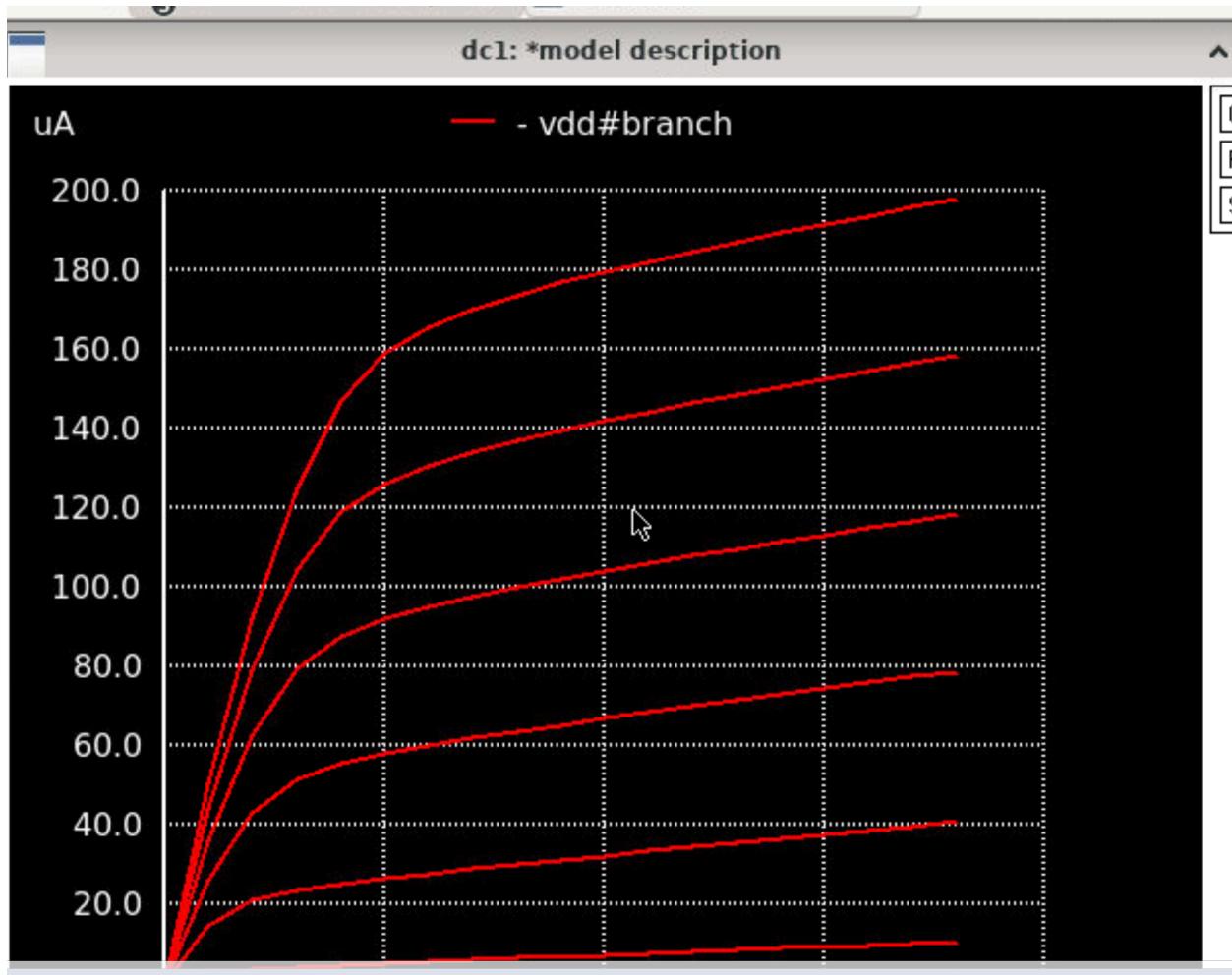


Figure 7. Id-vds L015 W039

## Lab2.2 id-vgv

day2\_nfet\_idvgs\_L015\_W039.spice

```
XM1 Vdd n1 0 0 sky130_fd_pr__nfet_01v8 w=0.39 l=0.15
```

```
R1 n1 in 55
```

```
Vdd vdd 0 1.8V
```

```
Vin in 0 1.8V
```

```
.op
```

```
.dc Vin 0 1.8 0.1
```

```
run
```

```
display
```

```
setplot dc1
```

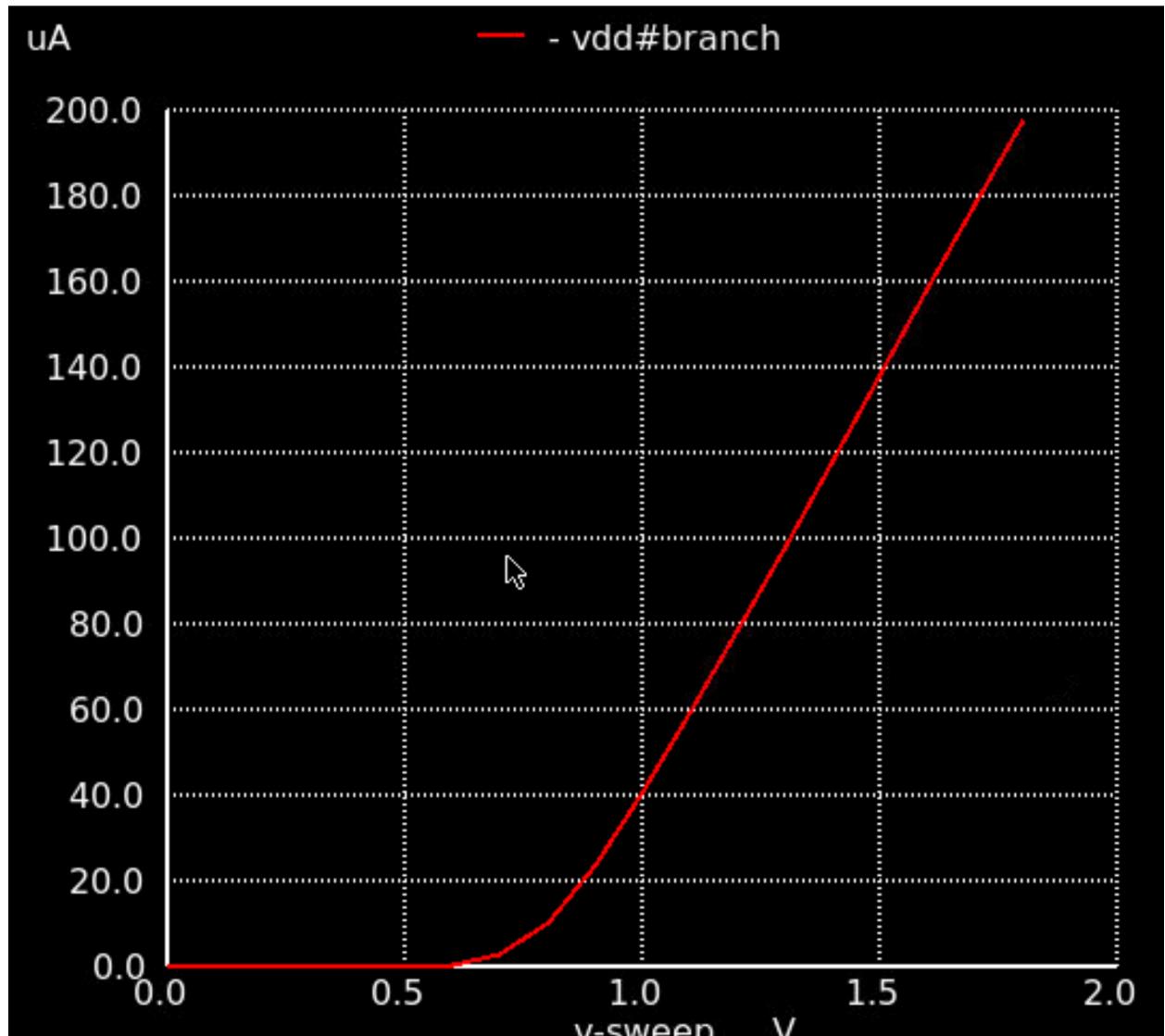


Figure 8: Id-vgs L015 W039

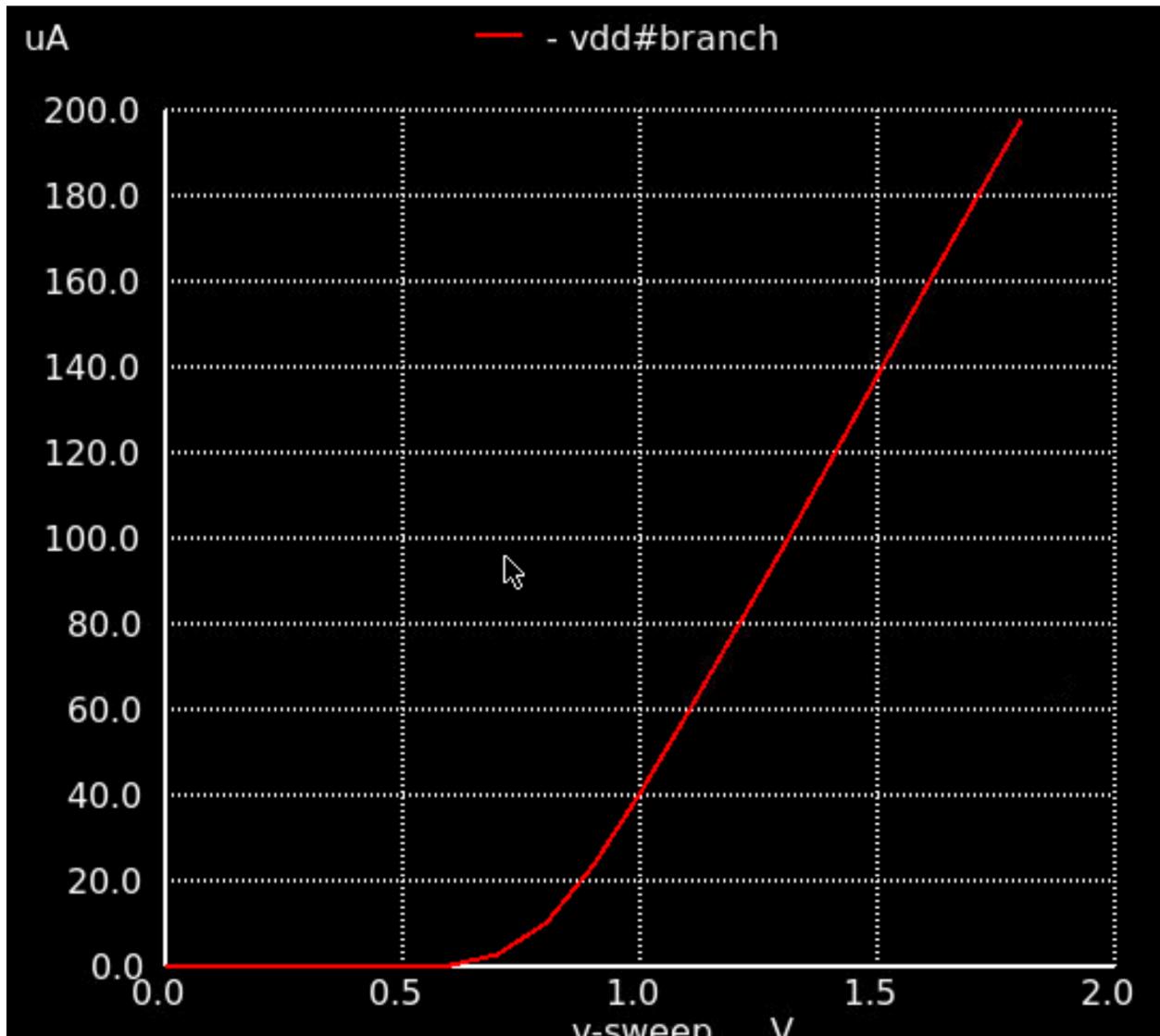


Fig 8 (backup) Id-vds L015 W039

## Lab 3.1 Trans

day3\_inv\_tran\_Wp084\_Wn036.spice

Complementary MOS

XM1 out in vdd vdd sky130\_fd\_pr\_pfet\_01v8 w=0.84 l=0.15

XM2 out in 0 0 sky130\_fd\_pr\_nfet\_01v8 w=0.36 l=0.15

Capacity load

Cload out 0 50fF

Pulse input

Vdd vdd 0 1.8V

Vin in 0 PULSE(0V 1.8V 0 0.1ns 0.1ns 2ns 4ns)

```
lib "sky130_fd_pr/models/sky130.lib.spice" tt

*Netlist Description

XM1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=0.84 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.36 l=0.15

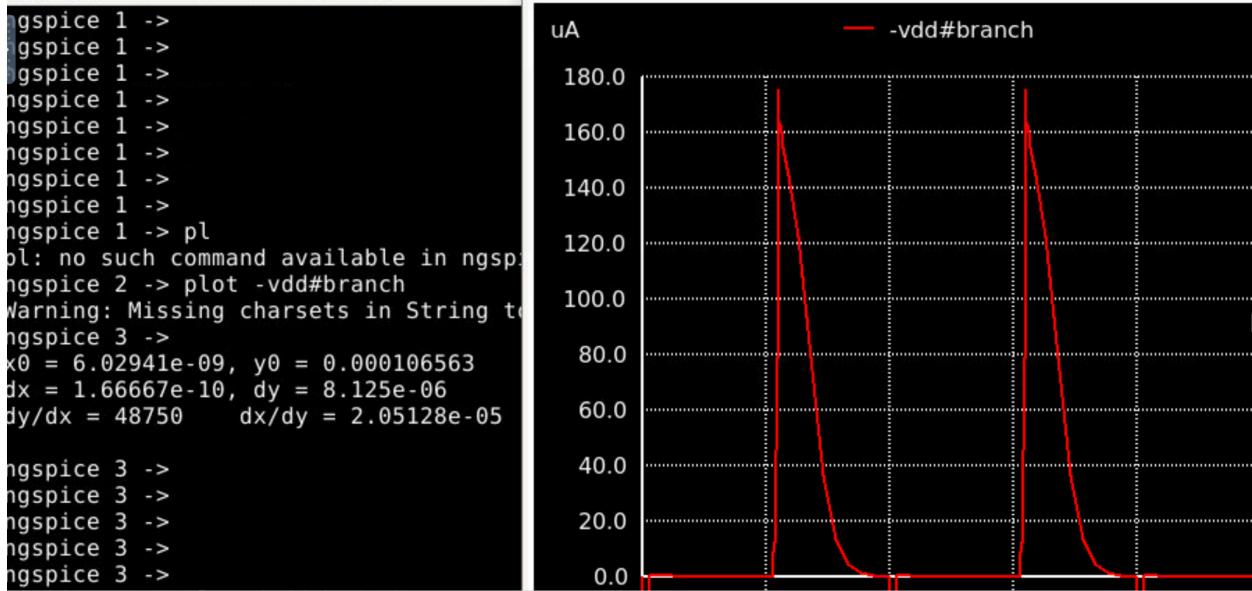
Cload out 0 50fF

Vdd vdd 0 1.8V
Vin in 0 PULSE(0V 1.8V 0 0.1ns 0.1ns 2ns 4ns)

*simulation commands

.tran 1n 10n

.control
run
.endc
```



**Figure 9 ( a above & b this ) : inv-tran Wp084\_Wn036 L15**

## Lab 3.2 inv vtc wp84 wn36 L15

day3\_inv\_vtc\_Wp084\_Wn036.spice

XM1 out in vdd vdd sky130\_fd\_pr\_\_pfet\_01v8 w=0.84 l=0.15  
 XM2 out in 0 0 sky130\_fd\_pr\_\_nfet\_01v8 w=0.36 l=0.15

Cload out 0 50fF

Vdd vdd 0 1.8V

Vin in 0 1.8V

\*simulation commands

.op

.dc Vin 0 1.8 0.01

```

Netlist Description

XM1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=0.84 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.36 l=0.15

Cload out 0 50fF

Vdd vdd 0 1.8V
Vin in 0 1.8V

*simulation commands

.op
.dc Vin 0 1.8 0.01
.control
run

```

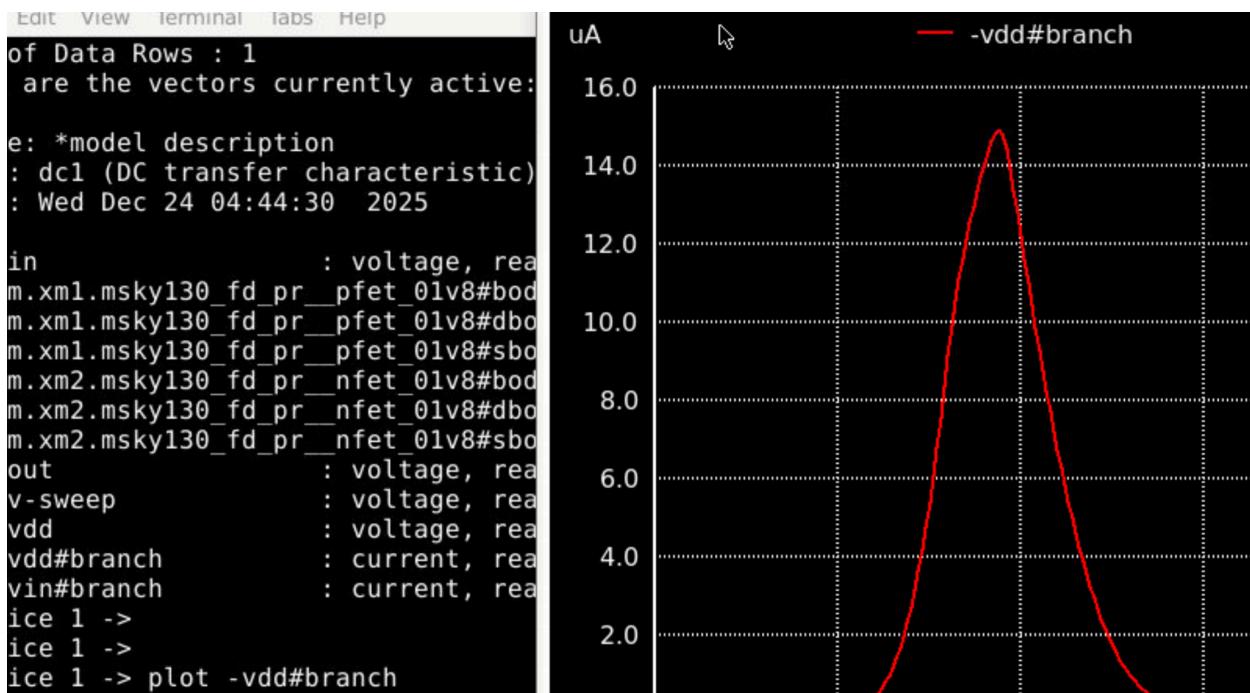
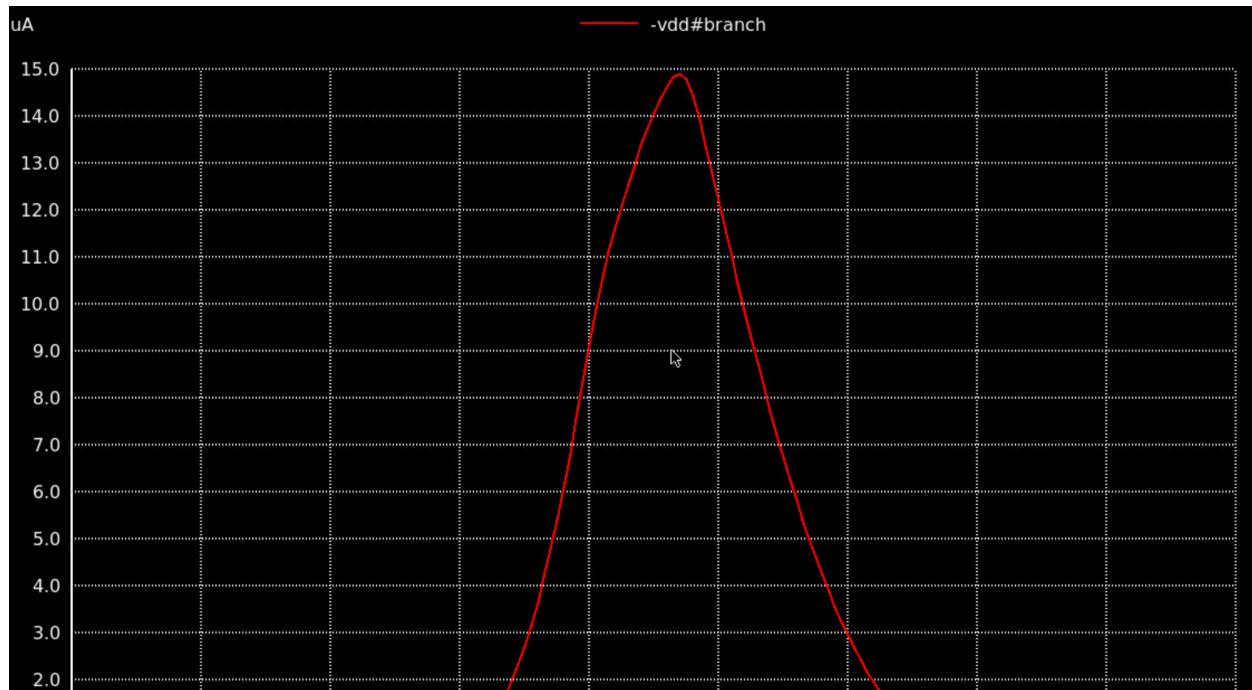


Figure 10: (a previous, b above) inv vtc Wp084\_Wn036 L15

## Repeating lab 3.2 for delay calculations on vtc plot:



```

ngspice 2 ->
x0 = 0.743103, y0 = 4.95238e-06      x1 = 0.744828, y1 = 5.02381e-06
dx = 0.00172414, dy = 7.14286e-08
dy/dx = 4.14286e-05      dx/dy = 24137.9

x0 = 0.868966, y0 = 1.30476e-05      x1 = 0.87069, y1 = 1.32381e-05
dx = 0.00172414, dy = 1.90476e-07
dy/dx = 0.000110476      dx/dy = 9051.72

x0 = 0.986207, y0 = 1.3e-05      x1 = 0.986207, y1 = 1.30238e-05
dx = 0, dy = 2.38095e-08

x0 = 1.13966, y0 = 4.97619e-06      x1 = 1.13621, y1 = 4.95238e-06
dx = -0.003444828, dy = -2.38095e-08
dy/dx = 6.90476e-06      dx/dy = 144828

```

For raising slope  $dy/dx = 4.143 * 10^{-5}$  raising delay =  $15 * dy/dx = 15 \times 6.215 \times 10^{-6} = 6.21 \mu s$

For falling slope  $dy/dx = 6.905 \times 10^{-6}$  falling delay =  $15 * dy/dx = 15 \times 6.905 \times 10^{-6} = 103.58 \mu s$

**Conclusion : Raising delay is smaller (steep raise) versus falling delay (slower fall)**

## Lab 4 inv-nm-wp1-wn36-L15

Day4\_inv\_noisemargin\_wp1\_wn036.spice

```
XM1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=1 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.36 l=0.15
Cload out 0 50fF
Vdd vdd 0 1.8V
Vin in 0 1.8V
*simulation commands
.op
.dc Vin 0 1.8 0.01
```

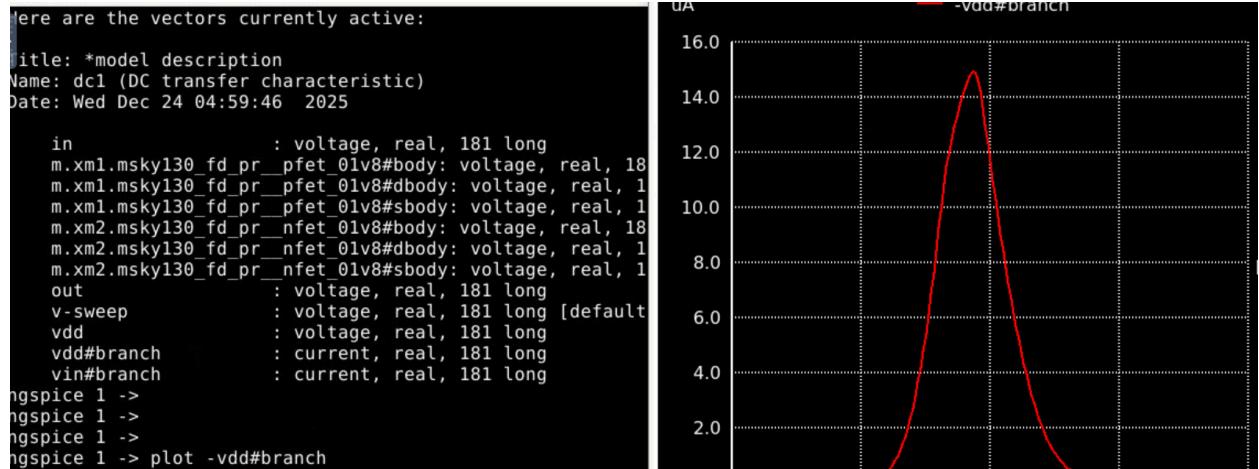
```
Netlist Description

XM1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=1 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.36 l=0.15

Cload out 0 50fF
Vdd vdd 0 1.8V
Vin in 0 1.8V

*simulation commands

.op
.dc Vin 0 1.8 0.01
.control
run
setplot dc1
display
```



**Figure 11: inv noisemargin wp1\_wn036**

## Lab 5.1 Device variation wp7-wn42 L15

Day5\_inv\_devicevariation\_wp7\_wn042.spice

XM1 out in vdd vdd sky130\_fd\_pr\_pfet\_01v8 w=7 l=0.15

XM2 out in 0 0 sky130\_fd\_pr\_nfet\_01v8 w=0.42 l=0.1

Cload out 0 50fF

Vdd vdd 0 1.8V

Vin in 0 1.8V

\*simulation commands

.op

.dc Vin 0 1.8 0.01

### Netlist Description

```
XM1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=7 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.42 l=0.15

Cload out 0 50fF

Vdd vdd 0 1.8V
Vin in 0 1.8V

*simulation commands

.op

.dc Vin 0 1.8 0.01

.control
run
setplot dcl
display
```

```
No. of Data Rows : 1
Here are the vectors currently active:

Title: *model description
Name: dcl (DC transfer characteristic)
Date: Wed Dec 24 05:05:39 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#body: voltage, real, 181 long
m.xm1.msky130_fd_pr_pfet_01v8#body: voltage, real, 181 long
m.xm2.msky130_fd_pr_nfet_01v8#body: voltage, real, 181 long
m.xm2.msky130_fd_pr_nfet_01v8#body: voltage, real, 181 long
m.xm2.msky130_fd_pr_nfet_01v8#body: 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 ->
ngspice 1 ->
ngspice 1 -> plot -vdd#branch
```

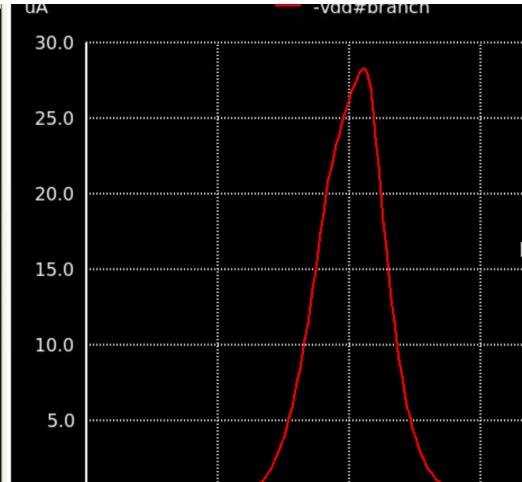


Figure 12: inv devicevariation wp7 L15\_wn042 L1

## Lab 5.2 Supply variation

day5\_inv\_supplyvariation\_Wp1\_Wn036.spice

XM1 out in vdd vdd sky130\_fd\_pr\_pfet\_01v8 w=1 l=0.15

XM2 out in 0 0 sky130\_fd\_pr\_nfet\_01v8 w=0.36 l=0.15

Cload out 0 50fF

Vdd vdd 0 1.8V

Vin in 0 1.8V

.control

```
let powersupply = 1.8
alter Vdd = powersupply
    let voltagesupplyvariation = 0
    dowhile voltagesupplyvariation < 6
        dc Vin 0 1.8 0.01
        let powersupply = powersupply - 0.2
        alter Vdd = powersupply
        let voltagesupplyvariation = voltagesupplyvariation + 1
    end
```

plot dc1.out vs in dc2.out vs in dc3.out vs in dc4.out vs in dc5.out vs in dc6.out vs in xlabel "input voltage(V)" ylabel "output voltage(V)" title "Inveter dc characteristics as a function of supply voltage"

The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. Below the menu, the terminal prompt is visible. The main area contains the following SPICE code:

```
M1 out in vdd vdd sky130_fd_pr_pfet_01v8 w=1 l=0.15
XM2 out in 0 0 sky130_fd_pr_nfet_01v8 w=0.36 l=0.15

Cload out 0 50fF

Vdd vdd 0 1.8V
Vin in 0 1.8V

.control

let powersupply = 1.8
alter Vdd = powersupply
    let voltagesupplyvariation = 0
    dowhile voltagesupplyvariation < 6
        dc Vin 0 1.8 0.01
        let powersupply = powersupply - 0.2
        alter Vdd = powersupply
        let voltagesupplyvariation = voltagesupplyvariation + 1
    end
```

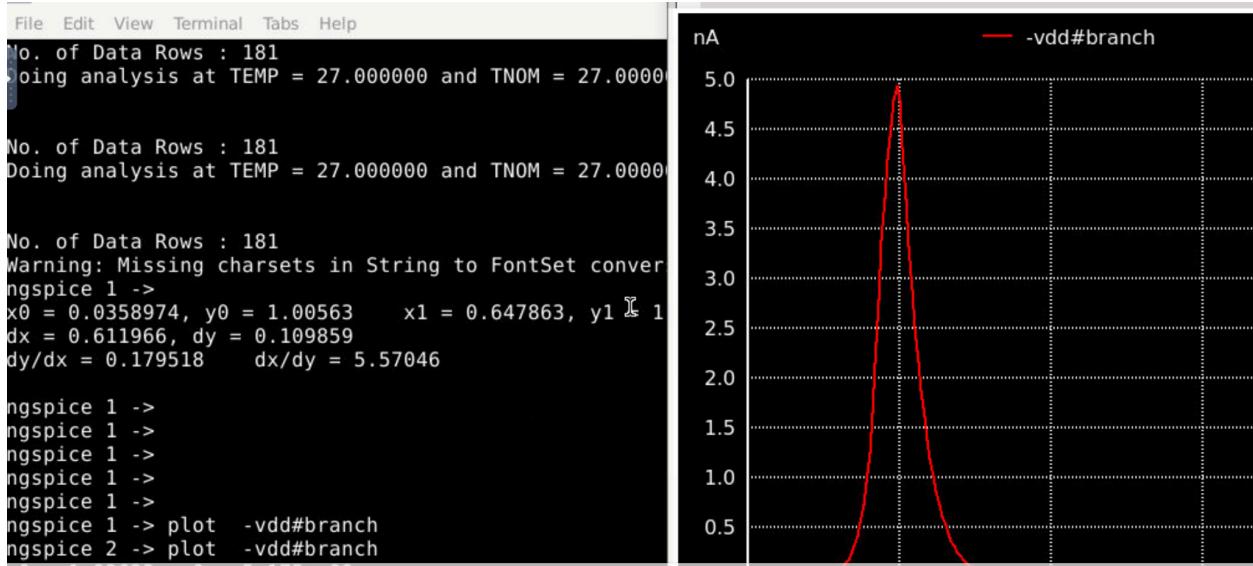


Figure 13 (a previous, b above) inv\_supplyvariation Wp1\_Wn036 L15

**Conclusion: VSD-CMOS is a great way to get to CMOS ngspice coding with proper theory behind the CMOS devices & circuit design (logical & physical)**