Project 1 Report

Link: https://www.notion.so/Project-1-Report-17ba50eabba94d05ba271b8cd0ba69ee?pvs=4

Team 13

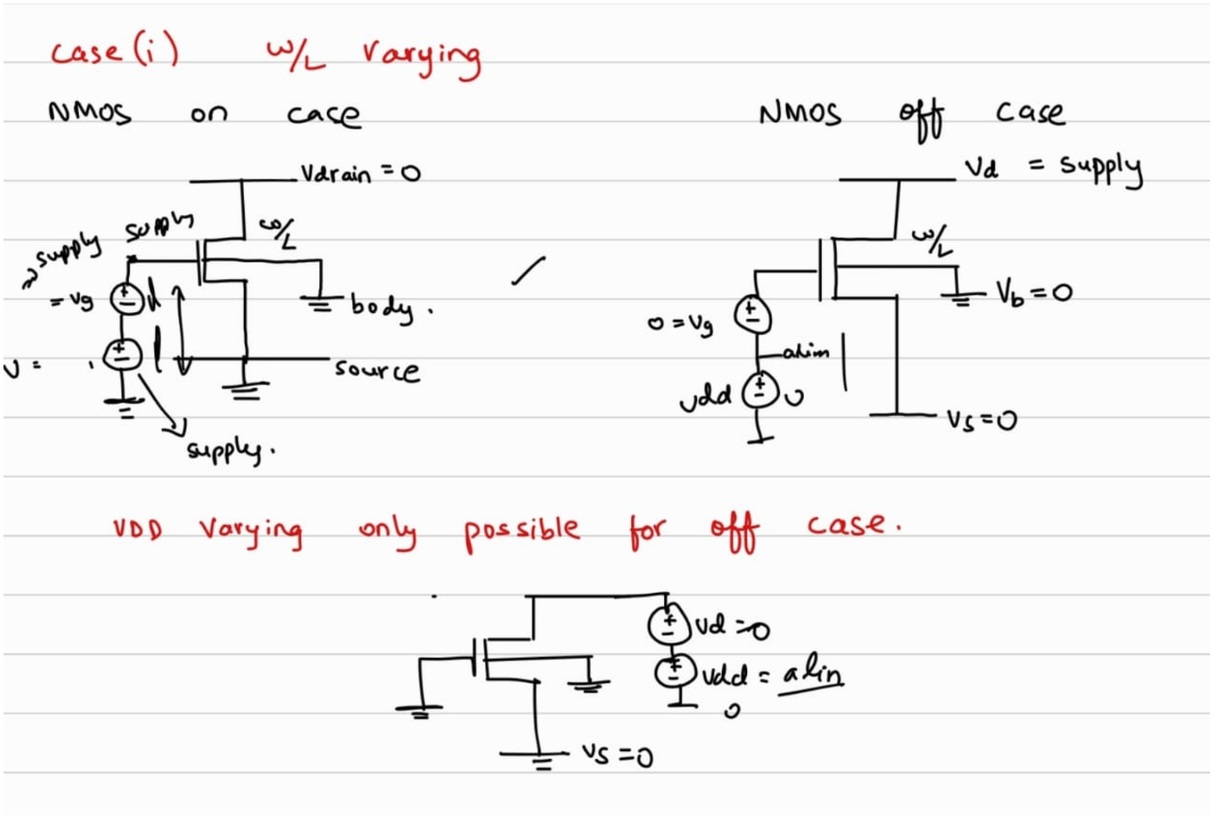
Name	Roll No.	Contribution
Kuncham Vishnu	2021102034	[Stage 1: Netlist][Stage 2:][Stage 2: NMOS Stack script and netlist][Stage 3: Total Circuit Leakage]
Jalluri Ram Gopal	2021102013	[Stage 1: Netlist][Stage 2:][Stage 2: PMOS Stack][Stage 3: CLA.ckt][NAND Leakage]
Pradhith VSS	2021102015	[Stage 1: VDD variation][Stage 2: PMOS Stack][Stage 3: CLA.ckt][NOR Leakages]
Koundinya Varma	2021112022	[Stage 1: W/L variation][Stage 2:][Stage 2: NMOS Stack][NAND][AND OR INVERTER Leakages]

Stage 1

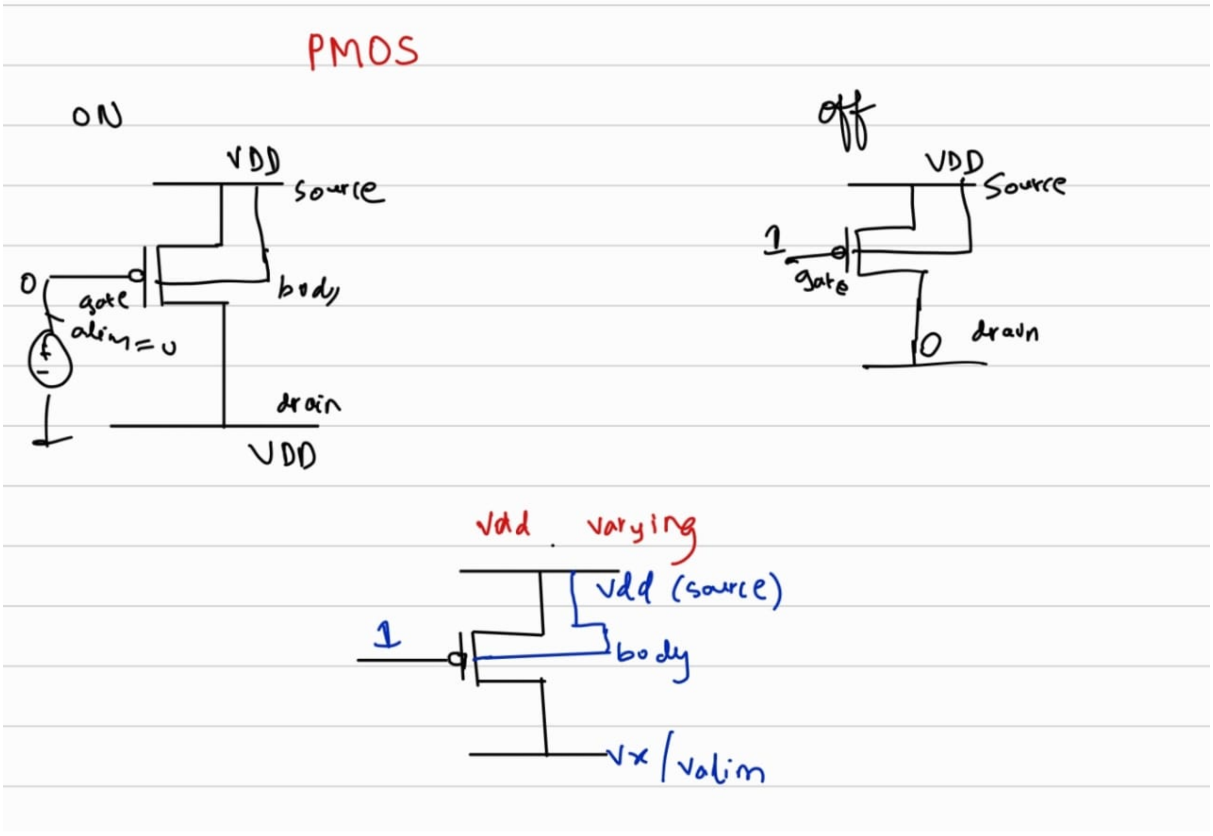
For this part, we must generate matrices for leakage current for single NMOS and PMOS for on-off conditions with varying widths and Drain Voltages.

```
i(vd):
[-1.81073e-11, -2.40044e-11, -2.94509e-11, -3.57397e-11, -4.32258e-11, -5.2200
3e-11, -6.29826e-11, -7.5947e-11, -9.15387e-11, -1.1029e-10, -1.32841e-10, -1.
59959e-10, -1.9257e-10, -2.31797e-10, -2.78997e-10, -3.35827e-10, -4.04309e-1
0, -4.86919e-10, -5.867e-10, -7.07399e-10, -8.53639e-10, -1.03114e-09]
[-4.14417e-11, -5.49278e-11, -6.73822e-11, -8.17655e-11, -9.8889e-11, -1.19418
e-10, -1.44083e-10, -1.73741e-10, -2.09409e-10, -2.52307e-10, -3.03897e-10, -
3.65936e-10, -4.40546e-10, -5.30289e-10, -6.38276e-10, -7.68296e-10, -9.24975e
-10, -1.11398e-09, -1.34227e-09, -1.61843e-09, -1.95302e-09, -2.35913e-09]

i(vg):
[5.730801e-15, 1.589759e-14, 3.186664e-14, 5.531605e-14, 8.829743e-14, 1.33308
5e-13, 1.933784e-13, 2.721685e-13, 3.740901e-13, 5.044436e-13, 6.695805e-13,
8.770943e-13, 1.136043e-12, 1.45721e-12, 1.853409e-12, 2.339839e-12, 2.934497e
-12, 3.658664e-12, 4.537467e-12, 5.600533e-12, 6.882749e-12, 8.425152e-12]
[1.309897e-14, 3.633735e-14, 7.283804e-14, 1.264367e-13, 2.018227e-13, 3.04705
1e-13, 4.420077e-13, 6.220994e-13, 8.550632e-13, 1.153014e-12, 1.53047e-12, 2.
004787e-12, 2.59667e-12, 3.330766e-12, 4.236364e-12, 5.348203e-12, 6.707421e-1
2, 8.36266e-12, 1.037135e-11, 1.280122e-11, 1.5732e-11, 1.925749e-11]

i(vs):
[1.810155e-11, 2.398851e-11, 2.941903e-11, 3.568438e-11, 4.313751e-11, 5.20669
4e-11, 6.278924e-11, 7.567486e-11, 9.116458e-11, 1.097858e-10, 1.321711e-10,
1.590815e-10, 1.914344e-10, 2.303396e-10, 2.771439e-10, 3.334875e-10, 4.013743
e-10, 4.832599e-10, 5.82162e-10, 7.017979e-10, 8.467557e-10, 1.022708e-09]
[4.14286e-11, 5.489149e-11, 6.730931e-11, 8.163903e-11, 9.868716e-11, 1.191133
e-10, 1.436414e-10, 1.731189e-10, 2.085544e-10, 2.511544e-10, 3.023665e-10, 3.
639316e-10, 4.37949e-10, 5.269578e-10, 6.3404e-10, 7.629481e-10, 9.182673e-10,
1.105616e-09, 1.3319e-09, 1.605624e-09, 1.937286e-09, 2.339864e-09]

i(vb):
[4.412519e-18, 5.051148e-18, 5.154377e-18, 5.18556e-18, 5.212142e-18, 5.24537e
-18, 5.2889e-18, 5.345838e-18, 5.419874e-18, 5.515605e-18, 5.63877e-18, 5.7965
38e-18, 5.997832e-18, 6.253751e-18, 6.578069e-18, 6.987855e-18, 7.504225e-18,
8.153263e-18, 1.63506e-17, 8.389667e-17, 4.836057e-16, 5.246463e-15]
[7.716151e-18, 8.83689e-18, 9.023884e-18, 9.087635e-18, 9.146794e-18, 9.222002
e-18, 9.320894e-18, 9.450472e-18, 9.619163e-18, 9.837477e-18, 1.011855e-17, 1.
047877e-17, 1.093857e-17, 1.152332e-17, 1.226455e-17, 1.320129e-17, 1.438188e-
17, 1.586598e-17, 3.447549e-17, 1.882214e-16, 1.09496e-15, 1.115201e-14]
```

The above is the data set obtained in matrices format where for varying VDD for 22 different values starting from 0.0 to 1.1V with a step size of 0.05V. We also varied the $\frac{W}{L}$ starting from $\frac{W}{L}$ to $8 \times \frac{W}{L}$. Firstly, we have an NGSpice script that outputs all the voltages and currents. Then, we call a script `trim.py` that takes the output of the NGSpice and gives us the above-generated matrix. Here, each matrix is of the size 8×22, where 8 represents different $\frac{W}{L}$s and 22 different values of Drain Voltages for both PMOS and NMOS.



Reference diagram for submission 1(NMOS case)

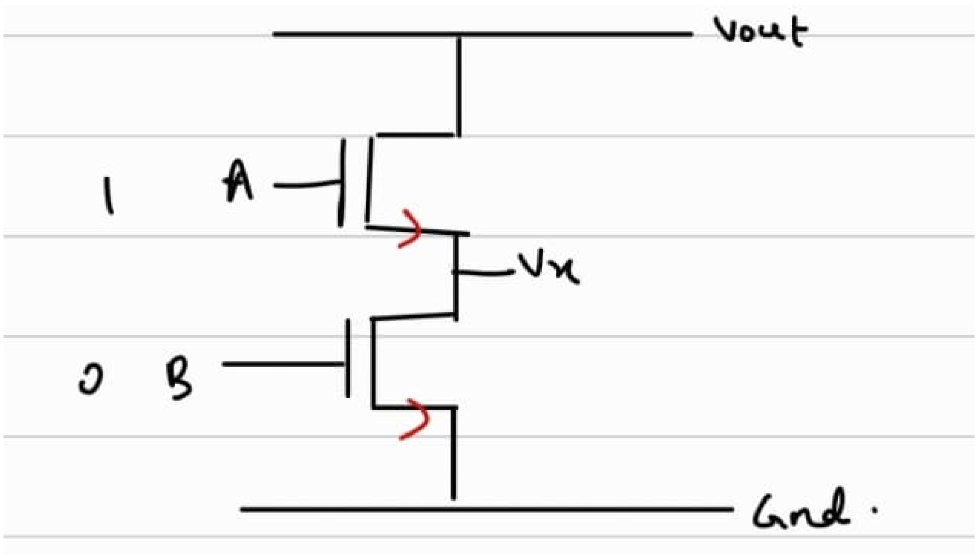


Reference diagram for PMOS Case

For on-off conditions of PMOS and NMOS, we use fixed configurations(i.e. we fix the drain, gate and source voltages). Then, by using a Python script, we vary the $\frac{W}{L}$s and store them in an `output.txt` file. We also vary the drain voltage of PMOS and NMOS in the OFF case and store the values of currents, which will then be used as the data set for the stacked case.

Stage 2

In this stage, we must generate the currents of a stacked MOSFET configuration. First, let us consider an NMOS stack with two inputs, A and B, at its gate, as shown in the figure below.



Let us look at how this stacked MOSFET circuit behaves for four different inputs.

Case 0:-

A	B	$V_{x_{theoritical}}$	V_{out}
0	0	Unknown	V_{DD}

Here, as both the MOSFETs are OFF and we are assuming the circuit to be a complimentary circuit, the V_{out} node charges to V_{DD}, and we can't comment on V_x exactly. To know the value of V_x, we use NGSpice.

Case 1:-

A	B	$V_{x_{theoritical}}$	V_{out}
0	1	≈ 0	V_{DD}

Here, as the lower MOSFET is in ON condition, V_x is driven to zero, and since the upper MOSFET is in OFF condition, the V_{out} is settled at V_{DD}.

Case 2:-

A	B	$V_{x_{theoritical}}$	V_{out}
1	0	$V_{DD} - V_{th}$	V_{DD}

Here, as the upper MOSFET is in ON condition, V_x can attain a maximum value of $V_{DD} - V_{th}$, and the V_{out} is settled at V_{DD}.

Case 3:-

A	B	$V_{x_{theoritical}}$	V_{out}
1	1	0	0

As both the MOSFETs are in ON condition, $V_x \& V_{out}$ are driven to zero.

These pre-recorded voltage values are now loaded into a script that takes the netlist of 2-stacked MOSFETs and stores them in a string. Based on our pre-recorded voltages, the values of A, B, and V_{out} are written into the `dst.ckt` and run using the subprocess module. Now, the `stdout` of the spice file is taken into an output array, and the voltage values of the drain gate and source of each MOSFET are taken from the output. Now, we have voltage values that can be mapped with the data set. Since we have drain, gate and source voltages, we made a new data set where the first three values are V_d, V_g, and V_s, followed by I_d, I_g, I_s, and I_b. This uses a netlist to vary a MOSFET's source and drain voltages.

The data set looks as follows;

```
Vd Vg Vs Id Ig Is Ib

0.0 0.0 0.0 -3.81513e-47 -1.75207e-47 -3.88586e-47 9.453056e-47
0.05 0.0 0.0 -1.81073e-11 5.730801e-15 1.810155e-11 4.412519e-18
0.1 0.0 0.0 -2.40044e-11 1.589759e-14 2.398851e-11 5.051148e-18
0.15 0.0 0.0 -2.94509e-11 3.186664e-14 2.941903e-11 5.154377e-18
0.2 0.0 0.0 -3.57397e-11 5.531605e-14 3.568438e-11 5.18556e-18
0.25 0.0 0.0 -4.32258e-11 8.829743e-14 4.313751e-11 5.212142e-18
```

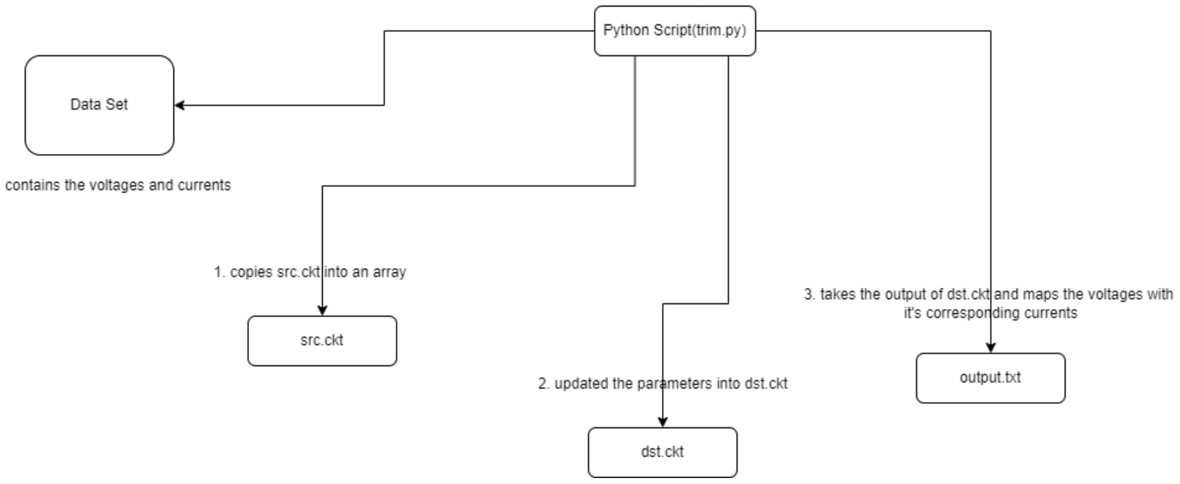
```
0.3 0.0 0.0 -5.22003e-11 1.333085e-13 5.206694e-11 5.24537e-18
...
```

Based on the values of drain, gate and source, we wrote a function called compute, which gives an index value that tells us the line to be picked in the provided data set.

Now, we will do this to all the MOSFETs in the circuit, and we'll be printing the values of leakages for each MOSFET for four different inputs, which is taken care of by the `trim.py` . The output of `trim.py` is as follows

```
input (AB)=  0.0

For mosfet1 the leakage currents are
1.1 0.0 0.1 -3.20181e-11 8.44105e-12 2.357422e-11 2.87724e-15

For mosfet2 the leakage currents are
0.1 0.0 0.0 -2.40044e-11 1.589759e-14 2.398851e-11 5.051148e-18

input (AB)=  1.0

For mosfet1 the leakage currents are
1.1 0.0 0.0 -1.03114e-09 8.425152e-12 1.022708e-09 5.246463e-15

For mosfet2 the leakage currents are
0.0 1.1 0.0 8.558463e-12 -3.37712e-11 8.558463e-12 1.665338e-11

input (AB)=  2.0

For mosfet1 the leakage currents are
1.1 1.1 0.85 -3.27079e-10 -1.06581e-14 3.270882e-10 1.412753e-15

For mosfet2 the leakage currents are
0.85 0.0 0.0 -4.04309e-10 2.934497e-12 4.013743e-10 7.504225e-18

input (AB)=  3.0

For mosfet1 the leakage currents are
0.0 1.1 0.0 8.558463e-12 -3.37712e-11 8.558463e-12 1.665338e-11

For mosfet2 the leakage currents are
0.0 1.1 0.0 8.558463e-12 -3.37712e-11 8.558463e-12 1.665338e-11
```

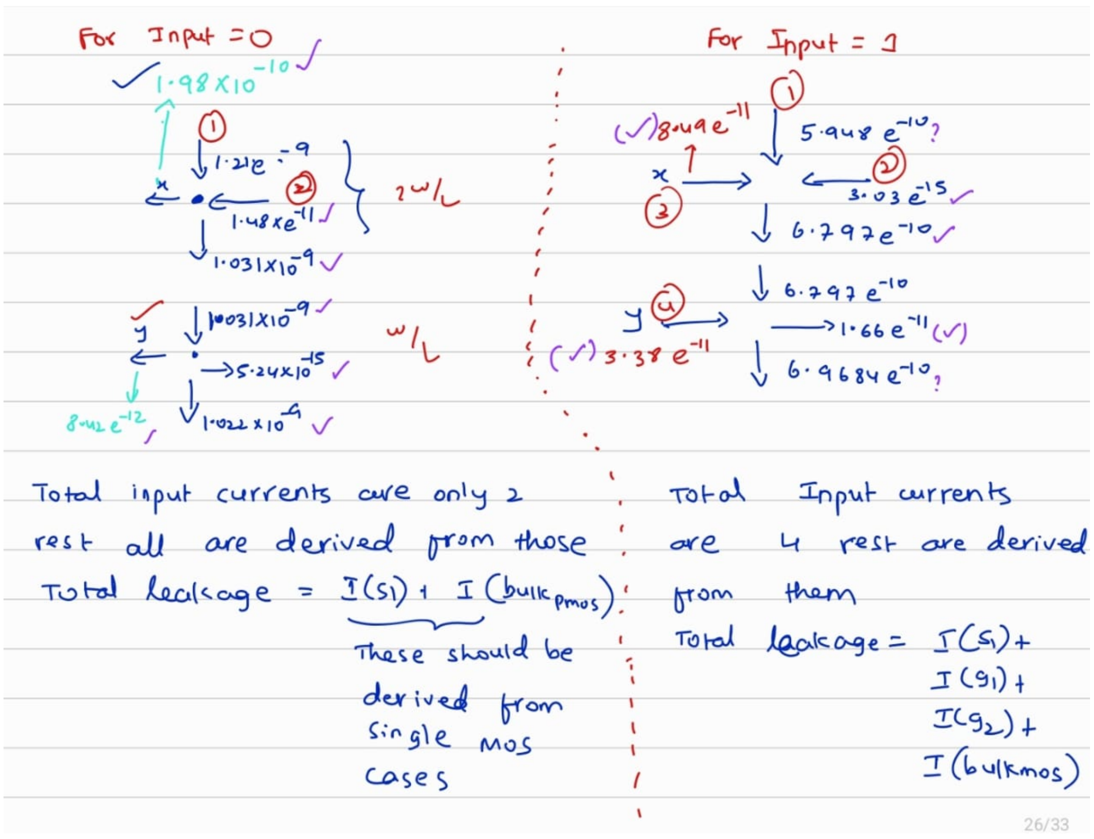
We also do the same analysis for PMOS following the procedures mentioned above, as seen in NMOS.


Python Script(trim.py)
Data Set
contains the voltages and currents
1. copies src.ckt into an array
src.ckt
2. updated the parameters into dst.ckt
dst.ckt
3. takes the output of dst.ckt and maps the voltages with it's corresponding currents
output.txt

Flow diagram for stage 2

Stage 3

1. Inverter



Total input currents are only 2 rest all are derived from those

Total leakage = I(S1) + I (bulk pmos)

These should be derived from single MOS cases

Total Input currents are 4 rest are derived from them

Total leakage = I(S1) + I(g1) + I(g2) + I(bulkmos)

We initially build an inverter netlist and print all the currents corresponding gate, drain, source and body. We can't use these data values directly, but we can use them to compare with single MOSFET data. As you can see in the above figure, the values that we wrote are obtained from the netlist, while the currents with a purple tick in the figure are the values that are closely matched with the single MOSFET cases.

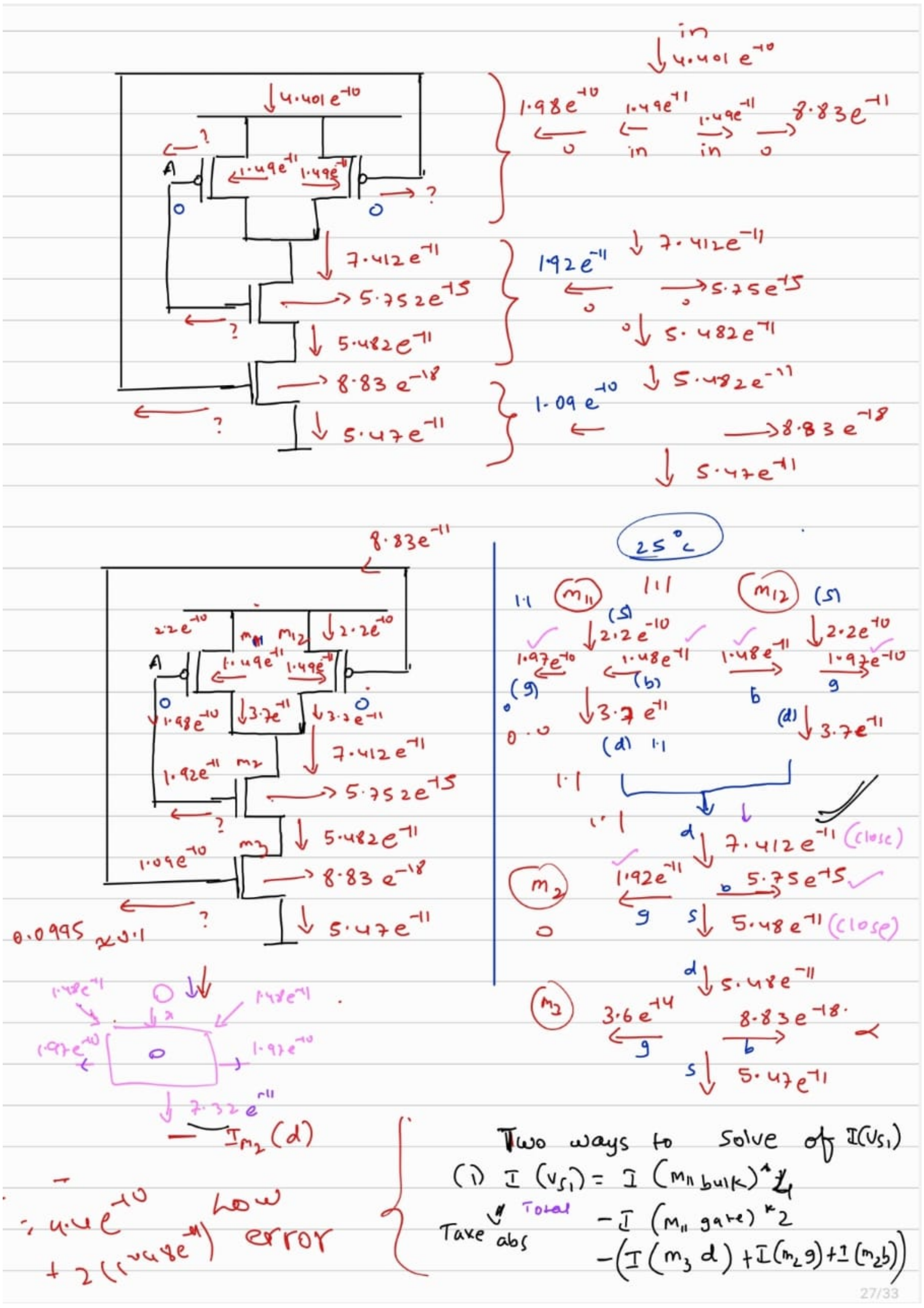
If we consider an inverter with input zero, the total leakage current is the current that is coming from the source and bulk of the top MOSFET, i.e. PMOS in the inverter. But when we matched them with the single MOSFET cases, the drain current did not match, so we needed to generate the drain current using KCL, and the KCL equation for the inverter case would be as follows.

This is the value of the source current. Now we need to add the bulk current of PMOS to get the total leakage current for input zero of the inverter, which can be obtained from the data set of stage 2. Similarly, we can also frame an equation for the inverter with input as 1. The obtained equation for an inverter with logic 1 as input is as follows

$$I_{\text{Total Leakage}} = I(s)_{PMOS} + I(g)_{PMOS} + I(bulk)_{PMOS} + I(g)_{NMOS}$$

where $I(s)_{PMOS}$ can be derived using the same KCL in the previous equation.

2. NAND



Similar to the inverter, we build a circuit for a NAND gate to calculate the leakage currents. Let us consider four different input cases as follows and their corresponding leakage current estimation.

Case: 0 0

$$I_{\text{Total Leakage}} = abs|2I_g(m11) + I_d(m2) - 2I_b(m11)| + 2I_b(m11)$$

Case: 0 1

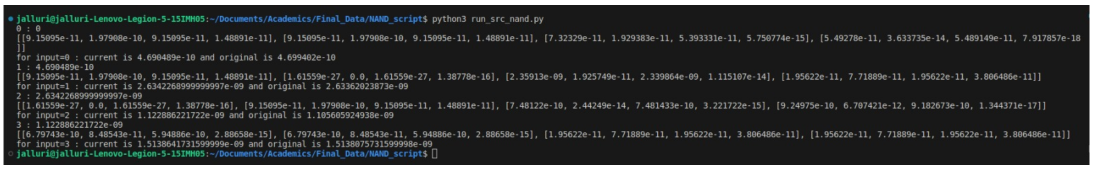
$$I_{\text{Total Leakage}} = abs|I_g(m11) + I_d(m2) - I_b(m11)| + I_g(m3) + I_b(m11)$$

Case: 1 0

$$I_{\text{Total Leakage}} = abs|abs| - I_g(m2) + I_d(m3) - I_b(m2)| + I_g(m12) - I_b(m12)| + I_b(m12) + I_g(m2)$$

Case: 1 1

$$I_{\text{Total Leakage}} = 2I_s(m11) + 2I_b(m11) + 2I_g(m11) + 2I_g(m2)$$

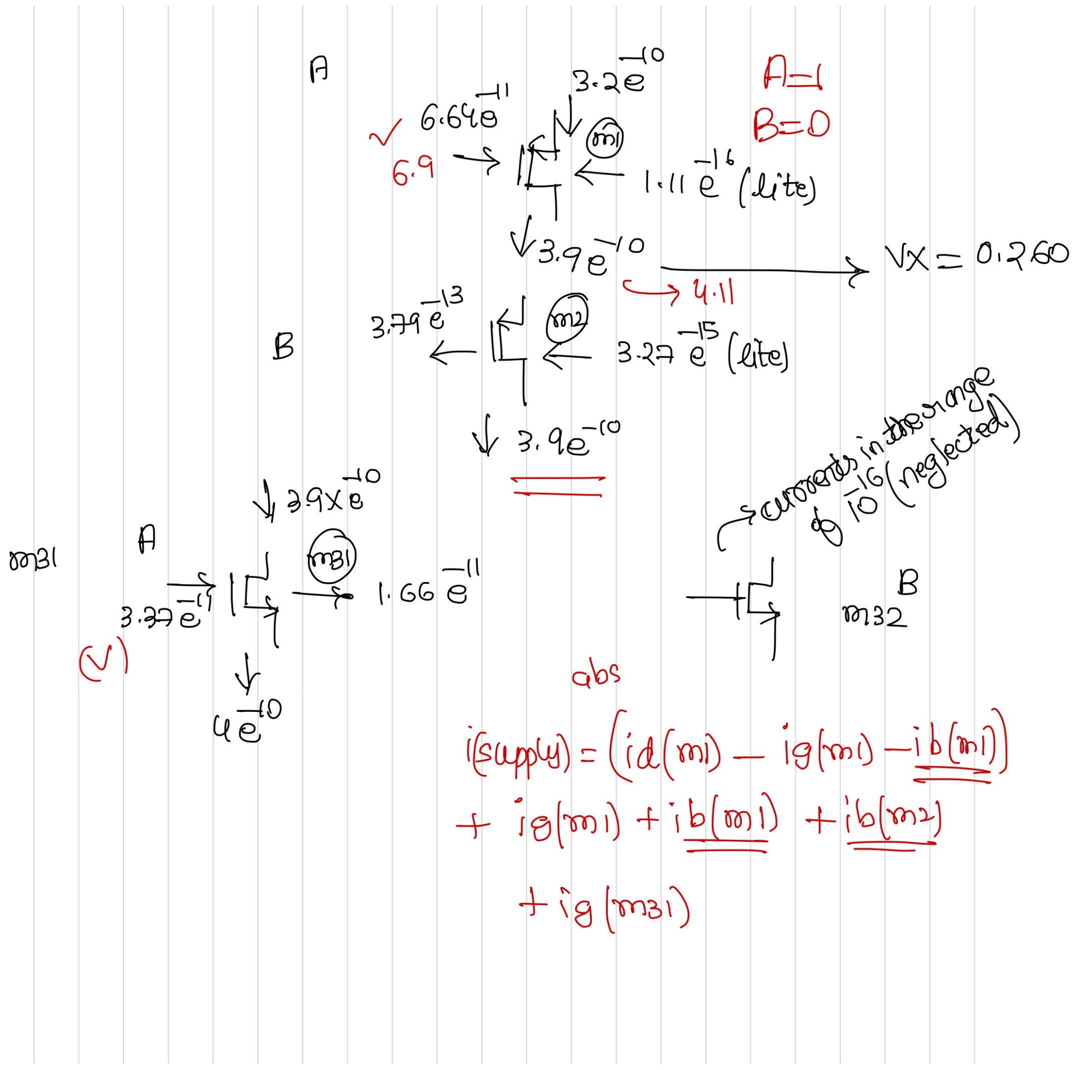


```
● jalluri@jalluri-Lenovo-Legion-5-15IMH05:~/Documents/Academics/Final_Data/NAND_script$ python3 run_src_nand.py
  0 : 0
  [[9.15095e-11, 1.97908e-10, 9.15095e-11, 1.48891e-11], [9.15095e-11, 1.97908e-10, 9.15095e-11, 1.48891e-11], [7.32329e-11, 1.929383e-11, 5.393331e-11, 5.750774e-15], [5.49278e-11, 3.633735e-14, 5.489149e-11, 7.917857e-18
  ]]
  for input=0 : current is 4.690489e-10 and original is 4.699402e-10
  1 : 4.690489e-10
  [[9.15095e-11, 1.97908e-10, 9.15095e-11, 1.48891e-11], [1.61559e-27, 0.0, 1.61559e-27, 1.38778e-16], [2.35913e-09, 1.925749e-11, 2.339864e-09, 1.115107e-14], [1.95622e-11, 7.71889e-11, 1.95622e-11, 3.806486e-11]]
  for input=1 : current is 2.6342268999999997e-09 and original is 2.63362023873e-09
  2 : 2.6342268999999997e-09
  [[1.61559e-27, 0.0, 1.61559e-27, 1.38778e-16], [9.15095e-11, 1.97908e-10, 9.15095e-11, 1.48891e-11], [7.48122e-10, 2.44249e-14, 7.481433e-10, 3.221722e-15], [9.24975e-10, 6.707421e-12, 9.182673e-10, 1.344371e-17]]
  for input=2 : current is 1.122886221722e-09 and original is 1.105605924938e-09
  3 : 1.122886221722e-09
  [[6.79743e-10, 8.48543e-11, 5.94886e-10, 2.88658e-15], [6.79743e-10, 8.48543e-11, 5.94886e-10, 2.88658e-15], [1.95622e-11, 7.71889e-11, 1.95622e-11, 3.806486e-11], [1.95622e-11, 7.71889e-11, 1.95622e-11, 3.806486e-11]]
  for input=3 : current is 1.5138641731599999e-09 and original is 1.5138075731599998e-09
○ jalluri@jalluri-Lenovo-Legion-5-15IMH05:~/Documents/Academics/Final_Data/NAND_script$ 
```

Leakage current matrix for each input case from data set and comparison with estimated and netlist leakage current `NAND.ckt`

The original current is the original leakage current extracted from `NAND.ckt` .

3. NOR



Reference image for input Case: 1 0

Considering the four different inputs in this case also,

Case: 0 0

$$I_{\text{Total Leakage}} = abs|2I_d(m31) + 2I_g(m1) - 2I_b(m1)| + 2I_b(m1)$$

Case: 0 1

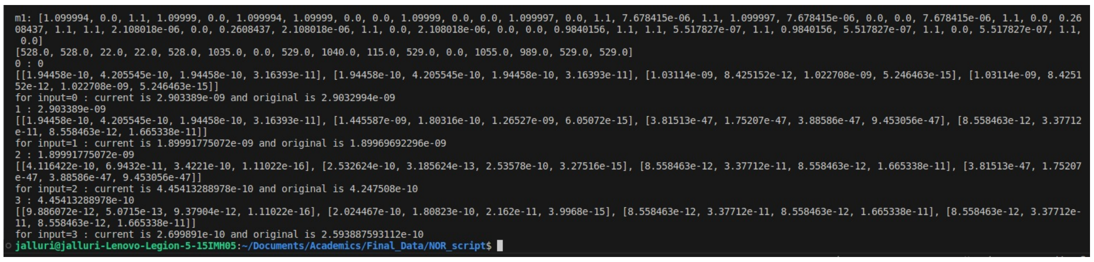
$$I_{\text{Total Leakage}} = abs|I_s(m2) + I_g(m1) - I_b(m1)| + I_b(m1) + I_b(m2) + I_g(m2) + I_g(m32)$$

Case: 1 0

$$I_{\text{Total Leakage}} = abs|I_d(m1) - I_g(m1) - I_b(m1)| + I_g(m1) + I_g(m31)$$

Case: 1 1

$$I_{\text{Total Leakage}} = abs|abs|I_d(m2) - I_g(m2)| - I_g(m1)| + I_g(m1) + I_g(m2) + I_g(m31) + I_g(m32)|$$



```
m1: [1.099994, 0.0, 1.1, 1.09999, 0.0, 1.099994, 1.09999, 0.0, 0.0, 1.09999, 0.0, 0.0, 1.099997, 0.0, 1.1, 7.678415e-06, 1.1, 1.099997, 7.678415e-06, 0.0, 0.0, 7.678415e-06, 1.1, 0.0, 0.26
08437, 1.1, 1.1, 2.108018e-06, 0.0, 0.2608437, 2.108018e-06, 1.1, 0.0, 2.108018e-06, 0.0, 0.0, 0.9840156, 1.1, 1.1, 5.517827e-07, 1.1, 0.9840156, 5.517827e-07, 1.1, 0.0, 5.517827e-07, 1.1,
 0.0]
[528.0, 528.0, 22.0, 22.0, 528.0, 1035.0, 0.0, 529.0, 1040.0, 115.0, 529.0, 0.0, 1055.0, 989.0, 529.0, 529.0]
0 : 0
[[1.94458e-10, 4.205545e-10, 1.94458e-10, 3.16393e-11], [1.94458e-10, 4.205545e-10, 1.94458e-10, 3.16393e-11], [1.03114e-09, 8.425152e-12, 1.022708e-09, 5.246463e-15], [1.03114e-09, 8.4251
52e-12, 1.022708e-09, 5.246463e-15]]
for input=0 : current is 2.903389e-09 and original is 2.9032994e-09
1 : 2.903389e-09
[[1.94458e-10, 4.205545e-10, 1.94458e-10, 3.16393e-11], [1.445587e-09, 1.80316e-10, 1.26527e-09, 6.05072e-15], [3.81513e-47, 1.75207e-47, 3.88586e-47, 9.453056e-47], [8.558463e-12, 3.37712
e-11, 8.558463e-12, 1.665338e-11]]
for input=1 : current is 1.89991775072e-09 and original is 1.89969692296e-09
2 : 1.89991775072e-09
[[4.116422e-10, 6.9432e-11, 3.4221e-10, 1.11022e-16], [2.532624e-10, 3.185624e-13, 2.53578e-10, 3.27516e-15], [8.558463e-12, 3.37712e-11, 8.558463e-12, 1.665338e-11], [3.81513e-47, 1.75207
e-47, 3.88586e-47, 9.453056e-47]]
for input=2 : current is 4.45413288978e-10 and original is 4.247508e-10
3 : 4.45413288978e-10
[[9.886072e-12, 5.0715e-13, 9.37904e-12, 1.11022e-16], [2.024467e-10, 1.80823e-10, 2.162e-11, 3.9968e-15], [8.558463e-12, 3.37712e-11, 8.558463e-12, 1.665338e-11], [8.558463e-12, 3.37712e-
11, 8.558463e-12, 1.665338e-11]]
for input=3 : current is 2.699891e-10 and original is 2.593887593112e-10
jalluri@jalluri-Lenovo-Legion-5-15IMH05:~/Documents/Academics/Final_Data/NOR_script$ 
```

For input 1 0, the intermediate node voltage is $V_x = 0.26$.

For input 1 1, the intermediate node voltage is $V_x = 0.98$.

 To generate the values for AND and OR, we used the leakage values of NAND, NOR,R and INVERTER. We build our whole circuit using AND, O, and NOT gates; wee use a script that matches our estimated current with the original current and calculates an error percentage. We have written one more script that calculates the error value for 512 different input cases.

After calculating all the error percentages, the results are as follows.

```
Average: 0.15775041561000316
Minimum value: 0.002819031378952852 at index 260
Maximum value: 0.2870387589847587 at index 206
```

The best-case error was for input 100000100

The worst-case error was for input 11001110