# Digital VLSI Design
# Project I: CMOS Leakage Current Estimation

Rupak Antani
2022102045
rupak.antani@students.iiit.ac.in

Prakhar Raj
2022102066
prakhar.raj@students.iiit.ac.in

Om Mehta
2022102046
om.mehta@students.iiit.ac.in

Himanshu Gupta
2022102002
himanshu.gupta@students.iiit.ac.in

## I. Introduction

The aim of this project is to develop a model to estimate the total leakage current of any CMOS circuit, with a specific focus on carry look-ahead adder circuit and verify the accuracy of the estimated value as compared to the actual value of leakage current.

The project consists of three stages:

- **Stage 1:** Finding actual leakage currents for an individual MOSFET for both on and off cases and creating a matrix from the values obtained.
- **Stage 2:** Finding the actual as well as estimated total leakage current for stacked MOSFETs (two stacked PMOS or NMOS transistors in series) and creating a matrix from the obtained values.
- **Stage 3:**
  - Developing a model for the estimation of the total leakage current of ISCAS 74182 circuit, using the matrices generated in the earlier two stages.
  - Comparing the estimated and actual value of leakage currents.

## II. Leakage Currents in MOSFETs

Leakage currents can occur in several forms in MOSFET devices, and each form has different causes and dependencies on the device's parameters. Understanding these leakage mechanisms is essential for minimizing unwanted power loss and accurately estimating the total leakage current in circuits.

### A. *Subthreshold Leakage*

Subthreshold leakage current ($I_{sub}$) occurs when the transistor operates below the threshold voltage, in the weak inversion region. It occurs due to the formation of an npn region by the drain, depletion region and the source, similar to a BJT. It is primarily a function of the drain-source voltage ($V_{DS}$) and the gate-source voltage ($V_{GS}$) and is given by the equation:

$$I_{sub} = I_0 \exp\left(\frac{V_{GS} - V_{th}}{nV_T}\right)\left(1 - \exp\left(\frac{-V_{DS}}{V_T}\right)\right) \tag{1}$$

where:

- $V_{th}$ is the threshold voltage,
- $V_T = kT/q$ is the thermal voltage,
- $I_0$ is a pre-exponential factor dependent on device characteristics.

### B. *Gate Leakage*

Gate leakage ($I_{gate}$) is caused by quantum tunneling of carriers through the gate oxide. This leakage becomes significant when the oxide thickness decreases in advanced technology nodes. The gate current can be expressed as:

$$I_{gate} = A\left(\frac{V_{gs}}{t_{ox}}\right)^{\alpha} \exp\left(-\frac{Bt_{ox}}{V_{gs}}\right) \tag{2}$$

Here, $t_{ox}$ is the oxide thickness, and $A$ and $B$ are fitting parameters derived from the material properties of the gate dielectric.

### C. *Body Leakage*

Body leakage current ($I_{body}$) results from reverse-biased p-n junctions between the source/drain and the substrate. The magnitude of body leakage depends on the body bias voltage ($V_{sb}$) and is generally expressed as:

$$I_{body} = I_s\left(e^{\frac{V_{sb}}{nV_T}} - 1\right) \tag{3}$$

Where $I_s$ is the reverse saturation current, and $n$ is the ideality factor.

### D. *Junction Leakage*

Junction leakage ($I_{junction}$) arises from reverse-biased p-n junctions, primarily between the source/drain and the bulk in MOSFETs. It is caused by minority carrier diffusion or thermally generated carriers in the depletion region. The leakage is influenced by junction area, temperature, and doping concentrations, and is given by:

$$I_{junction} = I_s\left(e^{\frac{V_{ds}}{nV_T}} - 1\right) \tag{4}$$

Where $I_s$ is the reverse saturation current, $V_{ds}$ the drain-source voltage, $V_T$ the thermal voltage, and $n$ the ideality factor (1 to 2). Junction leakage increases with temperature and can be significant in highly doped, scaled technologies.

## III. Simulation Methodology

This project on CMOS leakage currents estimation is simulated using NGSpice.

### A. *Technology and Parameters*

The simulations were performed using the **32nm High-Performance (HP)** technology node with the following parameters:

- **Supply voltage**: 1.1V
- **Threshold Voltage**: 0.5V
- **Temperature**: 25°C
- **Channel width values** ($W$): {32nm, 64nm, 96nm, 128nm, 160nm, 192nm}

### B. *Automation and Data Collection*

- A Python script was employed which facilitated the automated sweeping of the channel width ( W ) , capturing the leakage currents under different operating conditions and generating corresponding **matrices** *(.csv files)*.

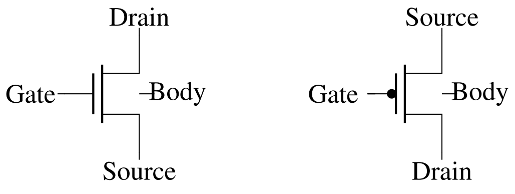
### C. *Voltage and Width Sweeps*

- To capture the variations in leakage currents, we performed simulations by sweeping the Drain & Source voltages ($V_{ds}$) from 0V and 1.1V with a step size of 0.01V .
- For each voltage combination, the channel width was also varied across six values, ranging from 32nm to 192nm, to study the impact of width scaling on leakage currents.

The individual transistor results were used as a baseline for leakage estimation in more complex circuits such as a carry look-ahead adder, which is analyzed in the later stages of this project.

## IV. Stage 1: Individual NMOS and PMOS Transistors

In Stage 1, we measured leakage currents for NMOS and PMOS devices for both on and off casess. Leakage currents were recorded for varying gate and drain voltages, and the results were stored in a matrix for further analysis.



NMOS Transistor PMOS Transistor

### A. *Observations:*

We have plotted graphs for trends of leakage currents with respect to varying voltages.

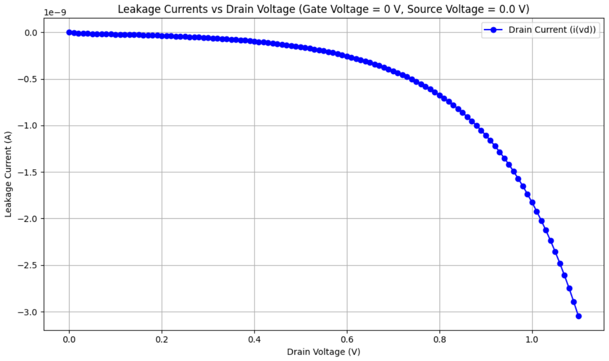


Fig. 1: Drain Leakage Current *Vs* Drain voltage in OFF state

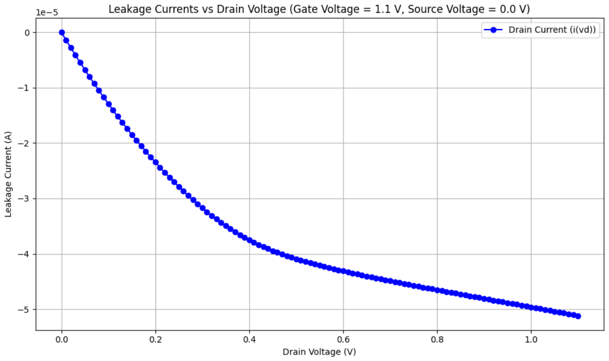


Fig. 2: Drain Leakage Current *Vs* Drain voltage in ON state

We can clearly observe that in OFF state, the drain leakage current is in the order of $1e^{-9}$ and in ON state, the drain leakage current is in the order of $1e^{-5}$. Also, for both cases we can observe that the drain, source, body and gate currents increase on increasing the drain voltage.

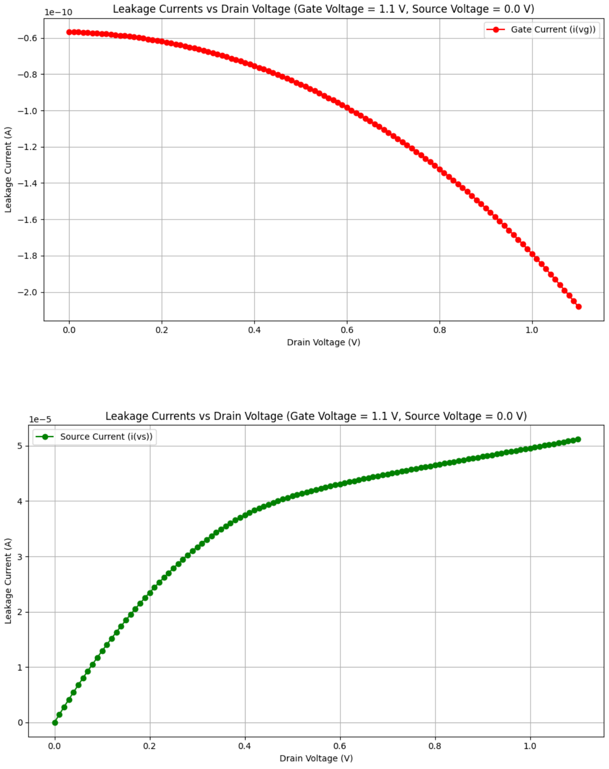


Fig. 3: Source Leakage Current *Vs* Drain voltage in OFF state

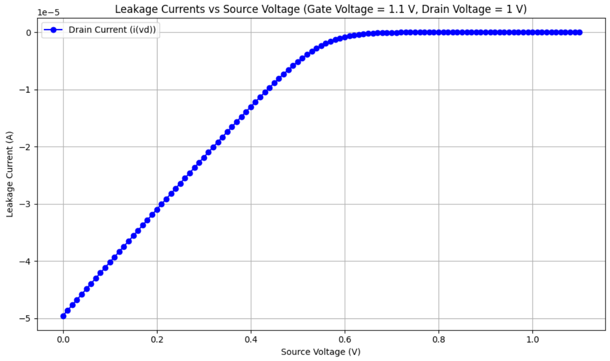


Fig. 4: Drain Leakage Current *Vs* Source voltage in ON state

Here we observe that the drain, source and body currents decrease with increase in **Source** Voltages.

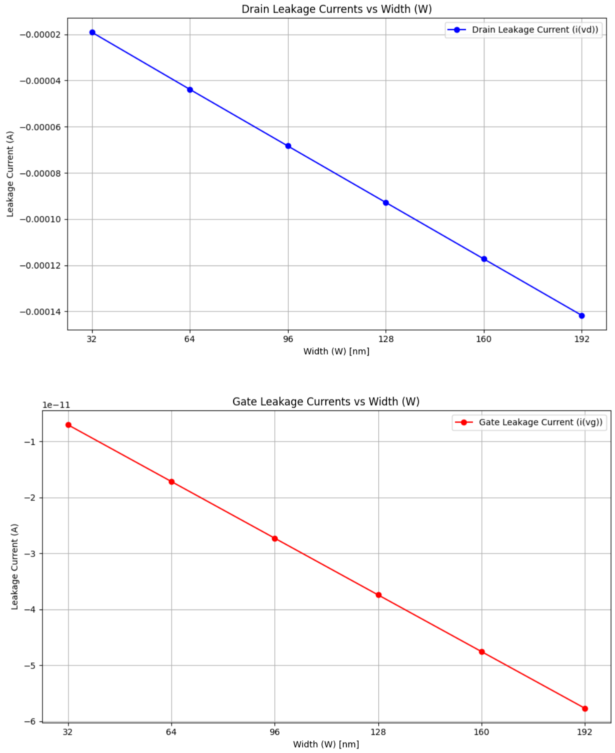


Fig. 5: With increase in W(width), the Drain and Gate leakage currents also increases.

In **Fig. 5.**, we have plotted the trend of leakage current with increasing channel width $W$. We took constant Gate, Drain & Source voltages and found the corresponding currents from which all the matrices were generated (for $W = \{32, 64, 96, 128, 160, 192\}$).

The results show that the leakage current increases with higher channel width and higher $V_{gs}$, particularly in the subthreshold regime.

## V. Stage II: Stacked NMOS & PMOS Devices

In this stage, we focused on finding the actual and estimated total leakage currents for stacked MOSFETs (two stacked PMOS or NMOS transistors in series) and generating the corresponding matrices (.csv file) for each value of channel width $W$.

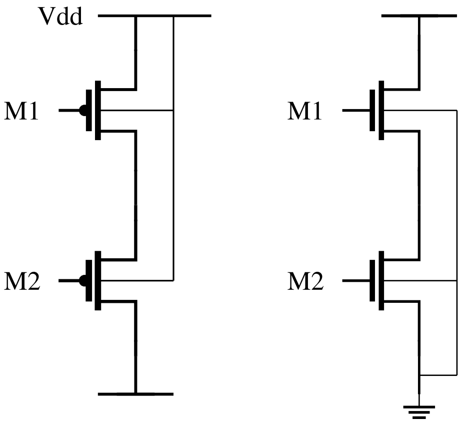


Fig 1: stacked NMOS  PMOS

We are considering the following cases:

- For stacked NMOS:

TABLE I: Stacked NMOS Logic States

| Input Logic | State |
|---|---|
| 0 0 | OFF |
| 0 1 | OFF |
| 1 0 | OFF |
| 1 1 | ON |

- For stacked PMOS:

TABLE II: Stacked PMOS Logic States

| Input Logic | State |
|---|---|
| 0 0 | ON |
| 0 1 | OFF |
| 1 0 | OFF |
| 1 1 | OFF |

For NMOS:

- When there is low input logic (0), *Drain Leakage dominates*.
- When there is high input logic (1.1), *Gate Leakage dominates*.

For PMOS:

- When there is low input logic (0), *Gate Leakage dominates*.
- When there is high input logic (1.1), *Drain Leakage dominates*.

For stacked NMOS, when both the transitors are ON , the drain of the top NMOS is connected to GND. and for the rest of the cases it is connected to the supply. Similarly, for stacked PMOS, when both the transitors are ON , the drain of the top PMOS is connected to supply voltage and for the rest of the cases it is connected to GND. This is done because we are imagining a pull up circuit on top of the NMOS stack and pull down circuit below the PMOS stack to simulate the behaviour of a CMOS circuit.

Thus, to calculate the total leakage currents for stacked NMOS transistors, suppose of we take the input logic of 01, we add the drain leakage current of the first NMOS (off state) and the gate leakage current of the second NMOS (on state). Similarly, for stacked PMOS transistors under the same input logic 01, we sum the gate leakage current of the first PMOS (on state) and the drain leakage current of the second PMOS (off state).

After calculating the actual leakage currents using NGSpice simulations and determinin the total leakage currents as discussed above, we also estimated the leakage currents using the matrices (.csv file) generated in ***Stage I***. This estimation yielded an accuracy of approximately 99%

## VI. Stage III: Leakage Current Estimation for ISCAS 74182 Circuit

Stage III involved using the results and matrices obtained from the first two stages to estimate the total leakage current for ISCA 74182 circuit. Leakage currents for individual transistors in the circuit were summed, and the estimates were compared with NGSpice simulation results. It involved the following steps:

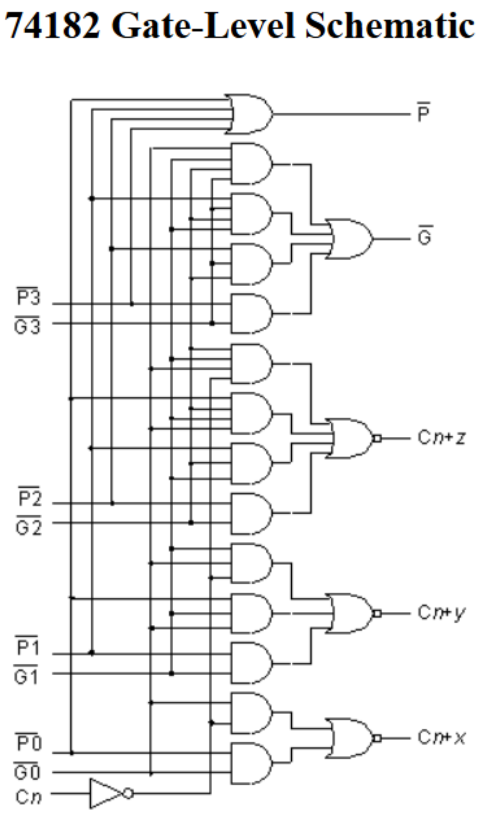
### *A. Building the circuit:*

In this section, we constructed two-input NAND, 2-input NOR and inverter sub-circuits at the transistor level using PMOS and NMOS transistors in NGSpice ensuring that the transistors are appropriately sized so that the pull-up and pull-down delays are equal. Subsequently, we used these two-input gates to build three-input and four-input NAND and NOR gates. After establishing these fundamental sub-circuits, we proceeded to make the adder sub-circuit that incorporates all the previously created sub-circuits.

### *B. Finding Actual Leakage Current Using Simulation:*

The actual total leakage current of the circuit made in NGspice was calculated by adding the current flowing from the supply voltage and currents through each of the inputs that are high.

The total actual leakage current for the ISCAS 74182 as obtained from the NGspice simulation varies based on the input values and is in the range of 330 - 410 nA.



### *C. Building a Model for Estimation of Leakage Current:*

The estimation of leakage current of a CMOS circuit is done by using the matrices obtained from ***Stage I & Stage II*** through python scripting. First, the total leakage current for 2-input NAND, 2-input NOR and inverter gates for different set of inputs are estimated by adding the values of leakage currents of the individual as well as stacked transistors present in the gate. Next, using the leakage currents of these gates, we calculate the overall leakage current of the higher input gates such as 4-input NAND, 3-input NAND, 4-input NOR, etc for all possible set of inputs. Finally, the Python script reads the circuit's structure, including the gates and their respective inputs, from the NGSpice file. The leakage current for each of these gates is then taken and then added together to get the total estimated leakage current.

### *D. Comparing the Accuracy :*

Lastly, we compare the actual and estimated leakage current of the circuit to ensure that the model is correct and can be used for estimation. Though the accuracy of the model we have built varies for each set of inputs, it is more that 93% for most of the inputs cases, with the maximum accuracy being 98.3%.

## VII. Conclusion

Estimating leakage currents through models and Python scripts offers several benefits compared to direct NGspice simulations:

- **Faster Analysis:** Estimation models save time by avoiding lengthy NGspice simulations, especially for large circuits with many transistors.
- **Scalability:** Once leakage is calculated for basic gates, it can be quickly scaled to more complex circuits, making it easier to handle larger designs.
- **Automation:** Python scripts can automate leakage estimation, streamlining the process and reducing manual effort or errors.
- **Quick Verification:** Estimating allows for rapid verification of design changes, providing quicker feedback compared to rerunning full simulations.

## VIII. CONTRIBUTIONS

- **Rupak Antani and Himanshu Gupta:** Worked on the NGspice part of all the three stages of the project as well as report making.
- **Om Mehta and Prakhar Raj:** Python scripting of all three stages of the project as well as report making.