

Designing and Simulation of a Current-Squaring Circuit design and validation in 350 nm CMOS Technology

1.INTRODUCTION:

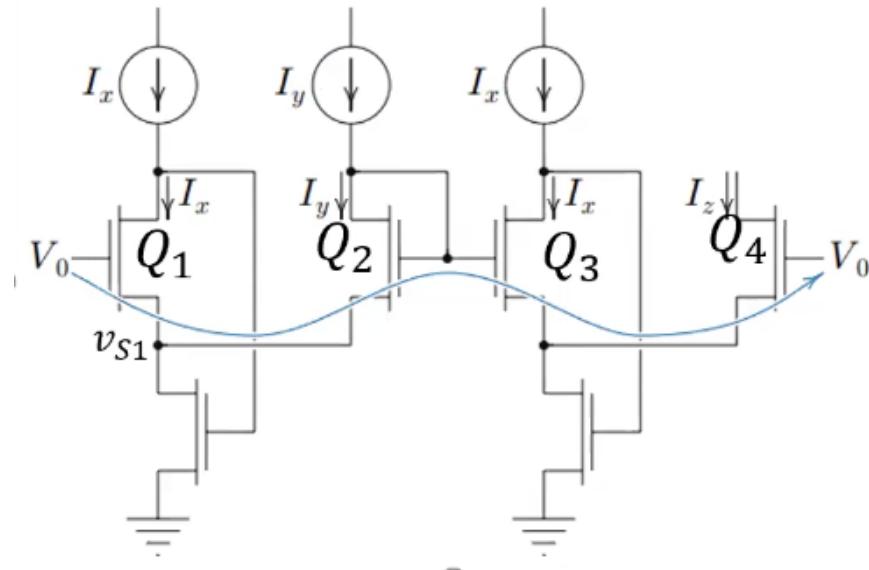
In this project, I designed and simulated a current-squaring circuit using TSMC 350nm CMOS technology in Cadence Virtuoso. The objective was to develop a circuit capable of computing the square of an input current (I_{in}), ensuring proper operation within the specified range of **1nA to 10nA**. This design is based on the translinear principle, which exploits the exponential I-V characteristics of MOSFETs operating in the weak-inversion region.

The core equation governing the circuit is:

$$I_z = \frac{I_x^2}{I_y}$$

where:

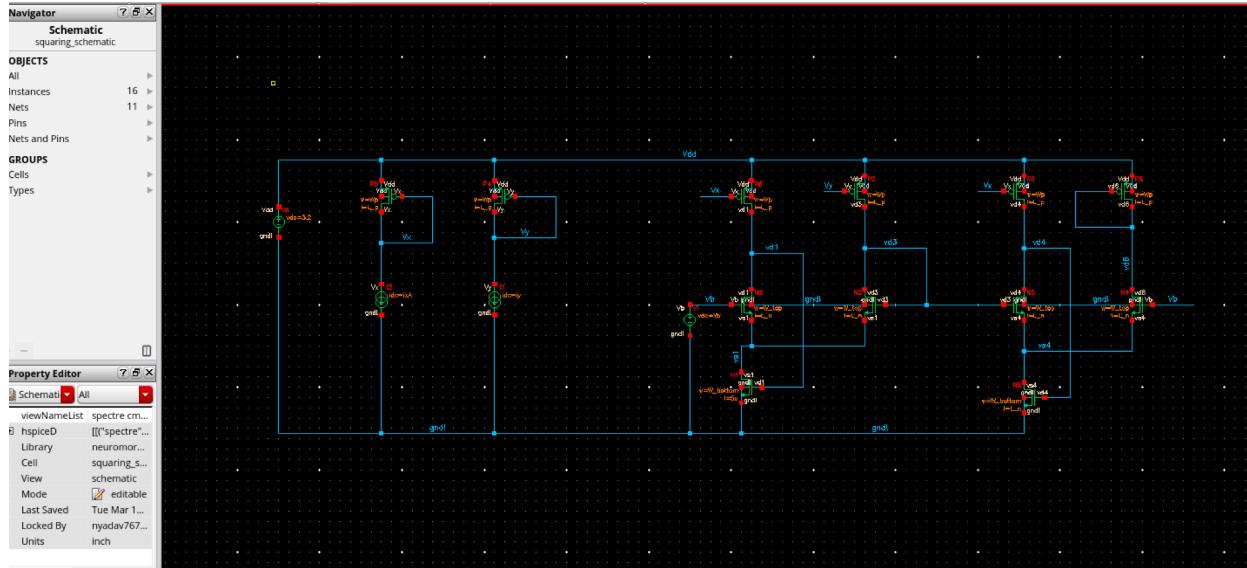
- I_x and I_y are input currents,
- I_z is the output squared current.



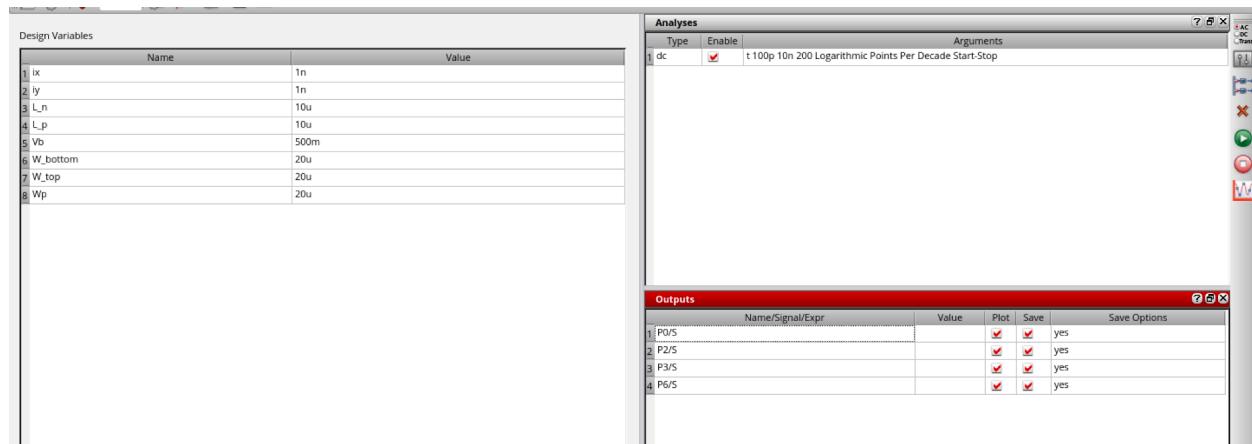
2. DESIGN METHODOLOGY:

Device Setup: Here is my initial setup with the chosen device parameters to ensure correct circuit operation.

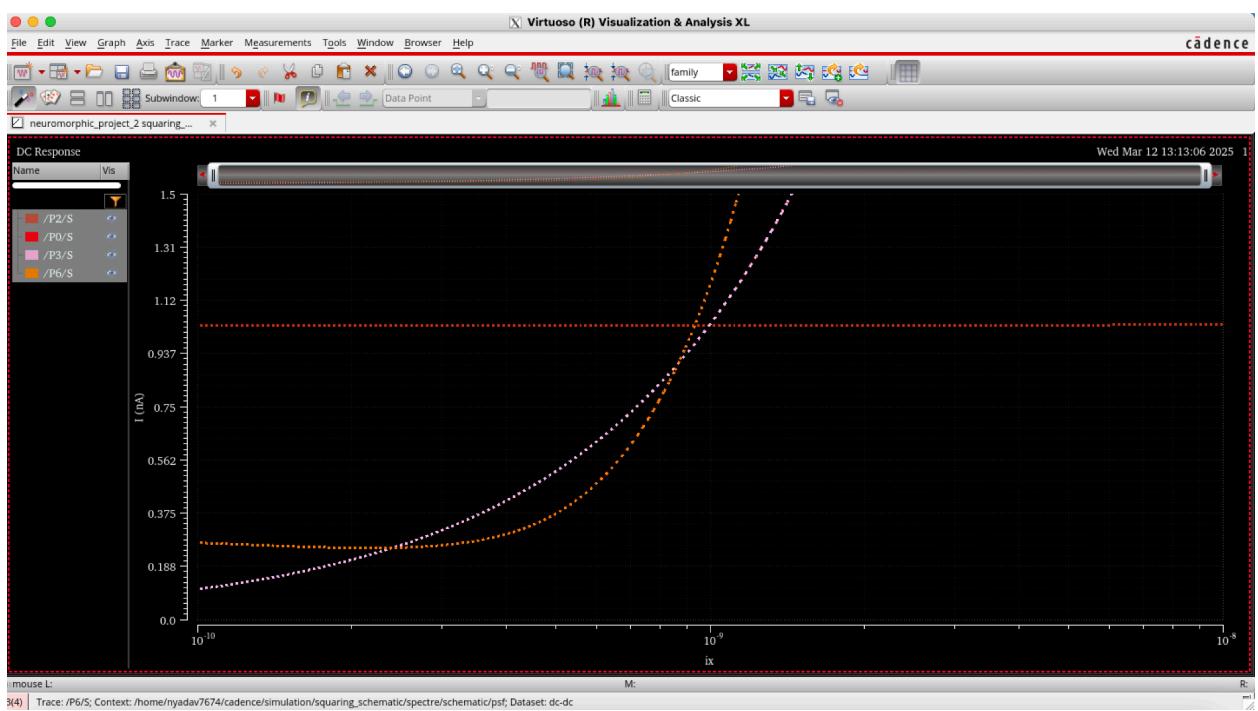
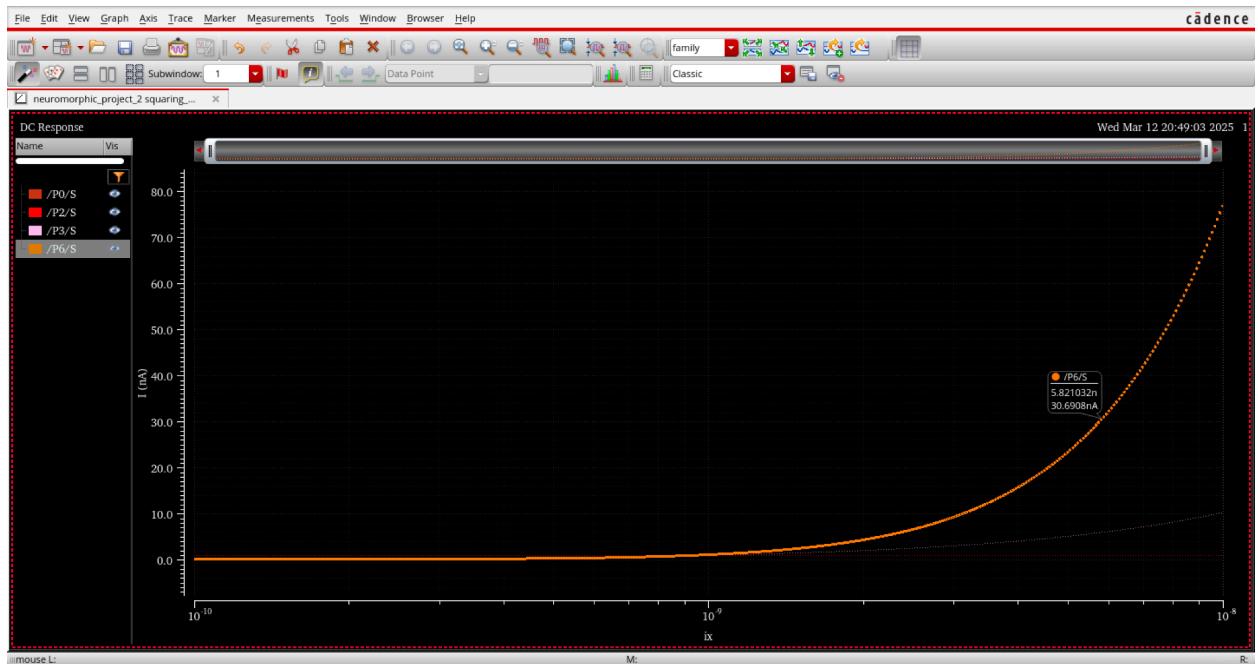
- Transistor Sizing : $W/L = 20\mu\text{m}/10\mu\text{m}$
- Supply Voltage : $V_{DD} = 3.2\text{V}$
- $V_{biasing} = 500\text{mV}$



A .DC analysis was performed to obtain vs. for ranging from 0 to 3.2V.

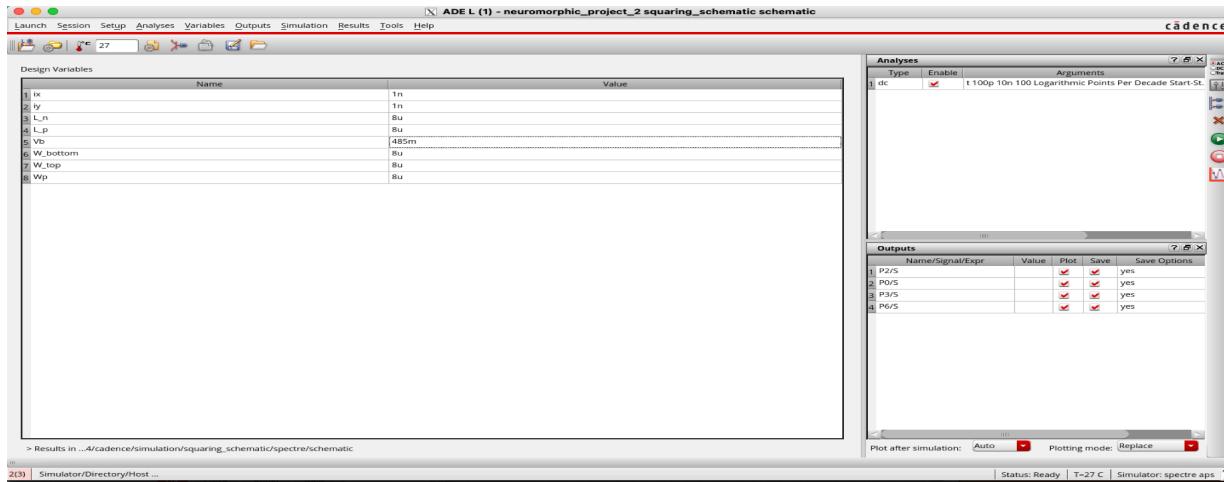


At this stage, my I_{out} did not follow the expected squared relationship (i.e) when $I_{in} = 5\text{A}$ the circuit produced $I_{out} = 30.69\text{A}$ which deviated from the expected squared value.

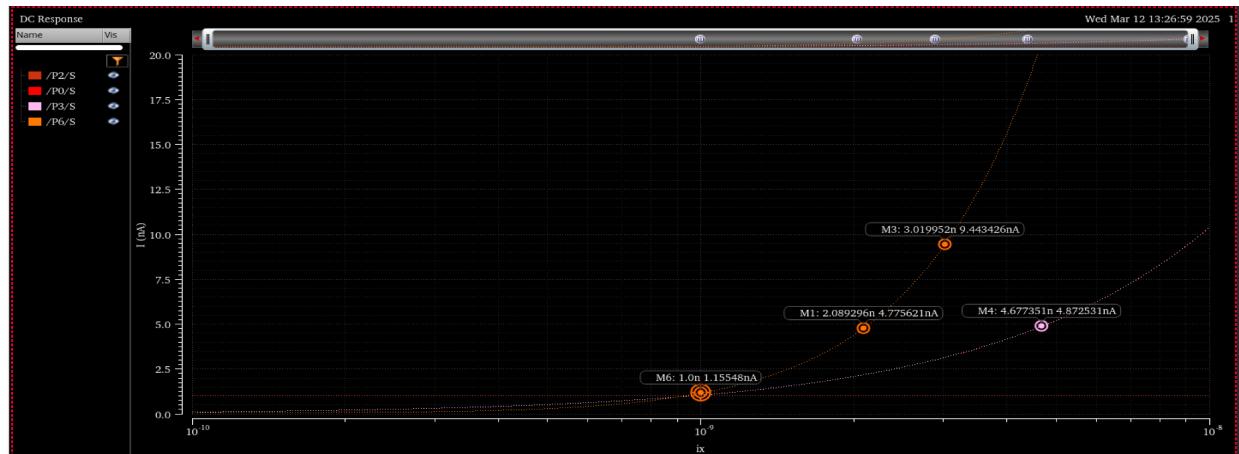


3.Transistor Sizing Optimization:

To achieve the desired output, I modified the transistor sizing and bias voltage as follows: The optimized parameters used in the design were $W/L = 8\mu\text{m}/8\mu\text{m}$, a bias voltage of 485mV, and a supply voltage of 3.2V.

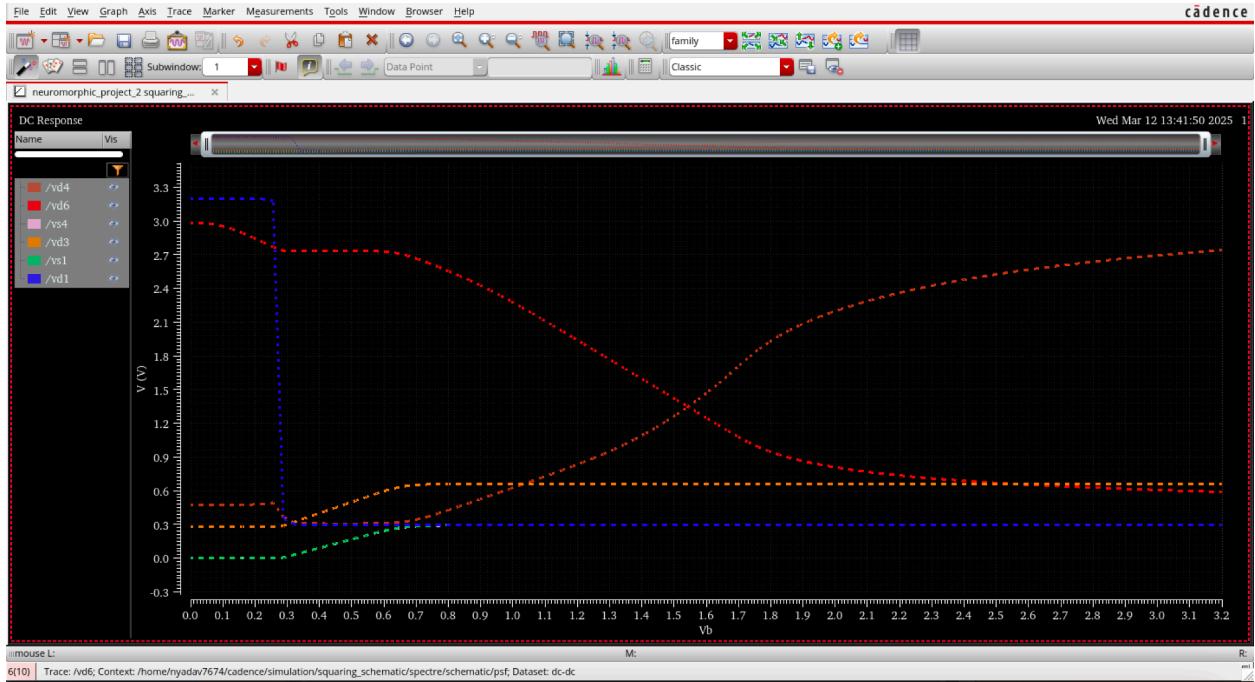


With these settings, it was observed that $I_y = 1\text{A}$ in the output, which remained consistent across different input values. The circuit followed the squaring function accurately. For an input current of $I_{in} = 3\text{A}$, the theoretical output current was calculated as $I_{out} = I_{in}^2 = 9\text{A}$, which matched the observed output of 9A, validating the design. Similarly, for $I_{in} = 2\text{A}$, the expected output was $I_{out} = I_{in}^2 = 4\text{A}$, which was also confirmed through simulation results. These results demonstrate that the circuit maintains the expected current-squaring behavior under the given design constraints, verifying its correctness and efficiency.



4. V₀ SELECTION:

To determine the optimal V_0 , a DC analysis was performed in Cadence Virtuoso, where V_{DS} was plotted against V_0 .



5. DC Analysis and MATLAB Visualization

To analyze the behavior of the circuit, I performed **DC analysis** in **Cadence Virtuoso**, where V_{DS} was plotted against V_0 . After obtaining the simulation data, I exported the results to a **CSV file** and further processed them in **MATLAB** for visualization and validation of the transistor operating regions.

The MATLAB script below reads the CSV file, extracts relevant voltage data, computes V_{DS} values for different transistors (Q1, Q2, Q3, Q4), and plots them against V_0 .

```

clc; clear; close all;

% 1. Read the CSV file
%     Replace 'data.csv' with the name of your CSV file.
data = readmatrix('vo_squaring_circuit_validation.csv');

v0=data(:,1);
vd1=data(:,2);
vs1=data(:,10);
vds1=vd1-vs1;
vds2=vs1;
vd3=data(:,4);
vs3=vs1;
vds3=vd3-vs3;
vd4=data(:,6);
vs4=data(:,12);
Vds4=vd4-vs4;
vds5=vs4;
vd6=data(:,8);
vs6=vs4;
vds6=vd6-vs6;

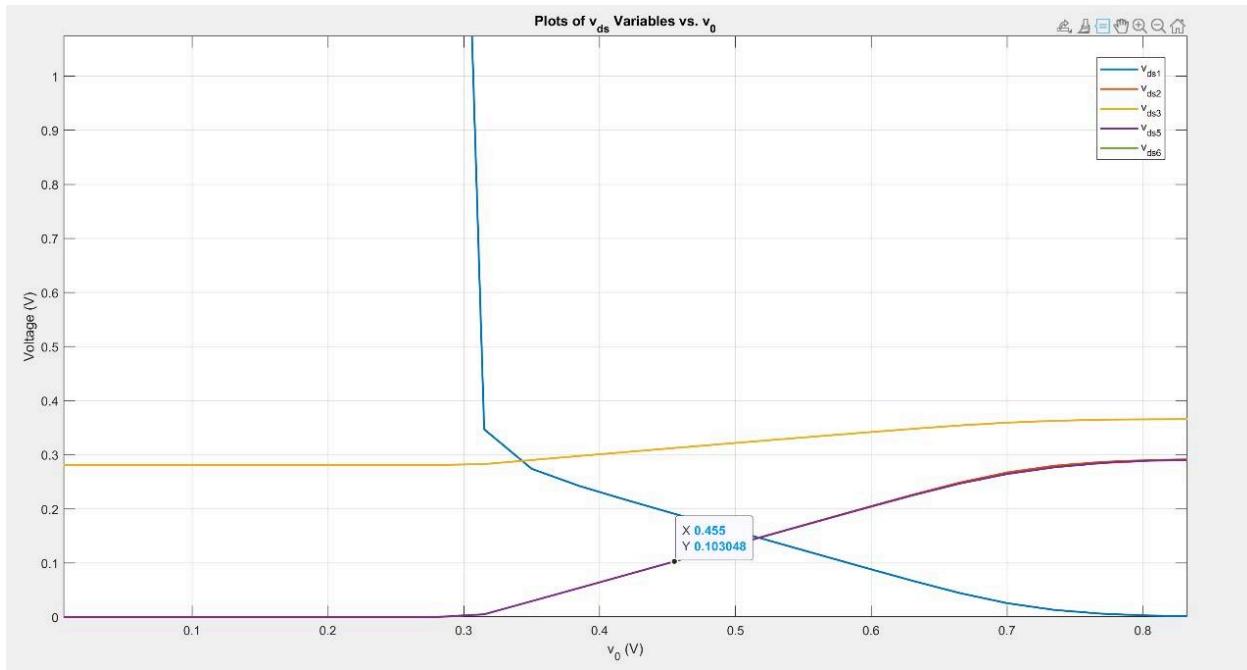
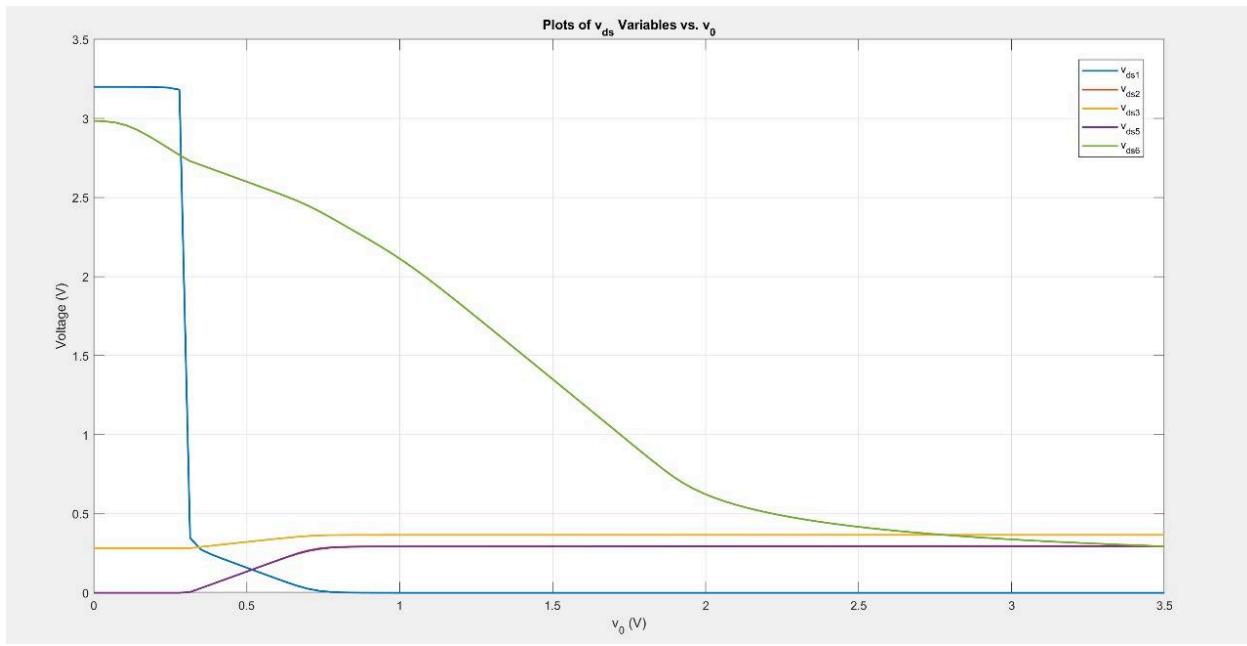
% ===== 2) Define an x-axis from 0 to 3.5 (v0) =====
numPoints = size(data,1);           % number of rows in the CSV
v0 = linspace(0,3.5,numPoints);    % create a vector from 0 to 3.5 V

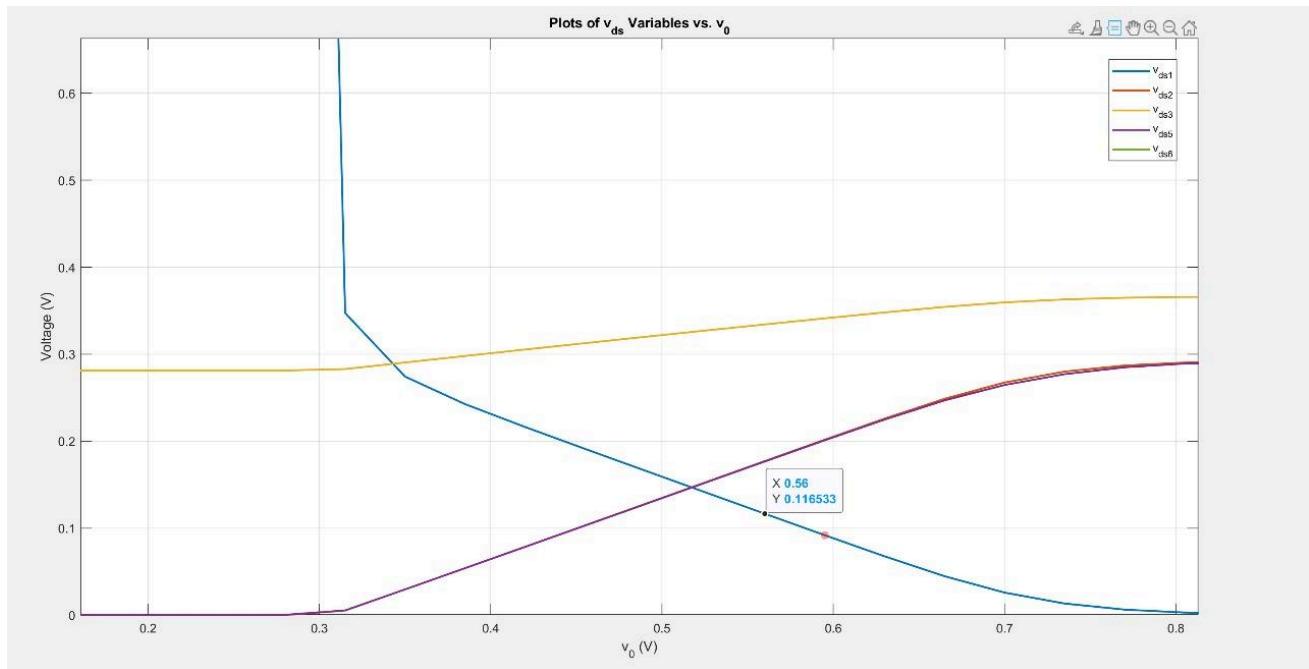
% ===== 4) Plot the computed voltages vs. v0 =====
figure;
plot(v0, vds1, 'LineWidth', 1.5); hold on;
plot(v0, vds2, 'LineWidth', 1.5);
plot(v0, vds3, 'LineWidth', 1.5);
plot(v0, vds5, 'LineWidth', 1.5);
plot(v0, vds6, 'LineWidth', 1.5);
hold off;
grid on;
xlim([0 3.5]);    % Force the x-axis from 0 to 3.5 V
ylim([0 3.5]);    % Force the y-axis from 0 to 3.5 V

% ===== 5) Label the plot =====
xlabel('v_0 (V)');
ylabel('Voltage (V)');
xlim([0 3.5]); % force x-axis from 0 to 3.5 V
title('Plots of v_{ds} Variables vs. v_0');

legend( ...
    {'v_{ds1}', 'v_{ds2}', 'v_{ds3}', 'v_{ds5}', 'v_{ds6}'}, ...
    'Location','best' ...
);

```





I plotted the transistor drain-to-source voltages against V_0 to determine the weak inversion saturation region. The key operating points observed were:

- $V_{DS} = 0.103048V, V_0 = 0.455V$
- $V_{DS} = 0.116533V, V_0 = 0.56V$

6.CURRENT ANALYSIS AND MAPE COMPUTATION:

To further validate the performance of the **current-squaring circuit**, I conducted a detailed analysis using MATLAB. The goal was to compare the **measured output current (I_z)** with the **theoretical calculation** based on the squaring relationship.

```
clc; clear; close all;

% ===== 1) Load Data =====
% If you have MATLAB R2019b or newer:
data = readmatrix('current_squaring_circuit.csv');

% For older MATLAB or Octave, comment out the above line and use:
% data = csvread('current_squaring_circuit.csv');

% Based on your description, the CSV columns are:
% Column 1: Ix1      (input current)
% Column 2: Iz2      (input current used in denominator)
% Column 3: Ix2      (input current)
% Column 4: Iz_meas (measured / true output current)

Ix1    = data(:,2);
Iy1    = data(:,4);
Ix2    = data(:,6);
Iz_meas = data(:,8);

% ===== 2) Calculate Iz_calc =====
% Formula: Iz_calc = (Ix1 * Ix2) / Iz2
Ix1Ix2 = Ix1 .* Ix2;          % product
Iz_calc = (Ix1Ix2) ./ Iy1;     % calculated output current

% ===== 3) Calculate MAPE =====
% MAPE formula (assuming Iz_meas is the true value):
%   MAPE = mean( abs(Iz_calc - Iz_meas) ./ abs(Iz_meas) ) * 100%
errors = abs(Iz_calc - Iz_meas) ./ abs(Iz_meas);
MAPE   = mean(errors) * 100;

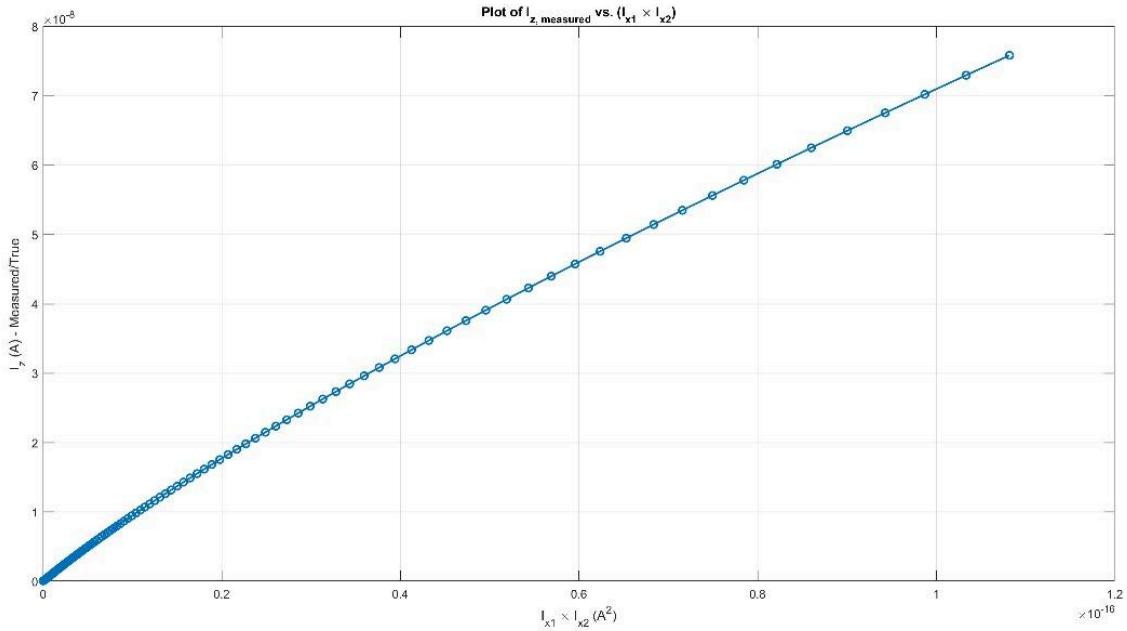
fprintf('Mean Absolute Percentage Error (MAPE) = %.2f %%\n', MAPE);

% ===== 4a) Plot #1: Iz_meas vs. Index + Iz_calc vs. Index =====
figure;
plot(Iz_meas, 'o-', 'LineWidth', 1.5); % measured Iz on y-axis
hold on;
plot(Iz_calc, 'x-', 'LineWidth', 1.5); % calculated Iz on y-axis
hold off;
grid on;

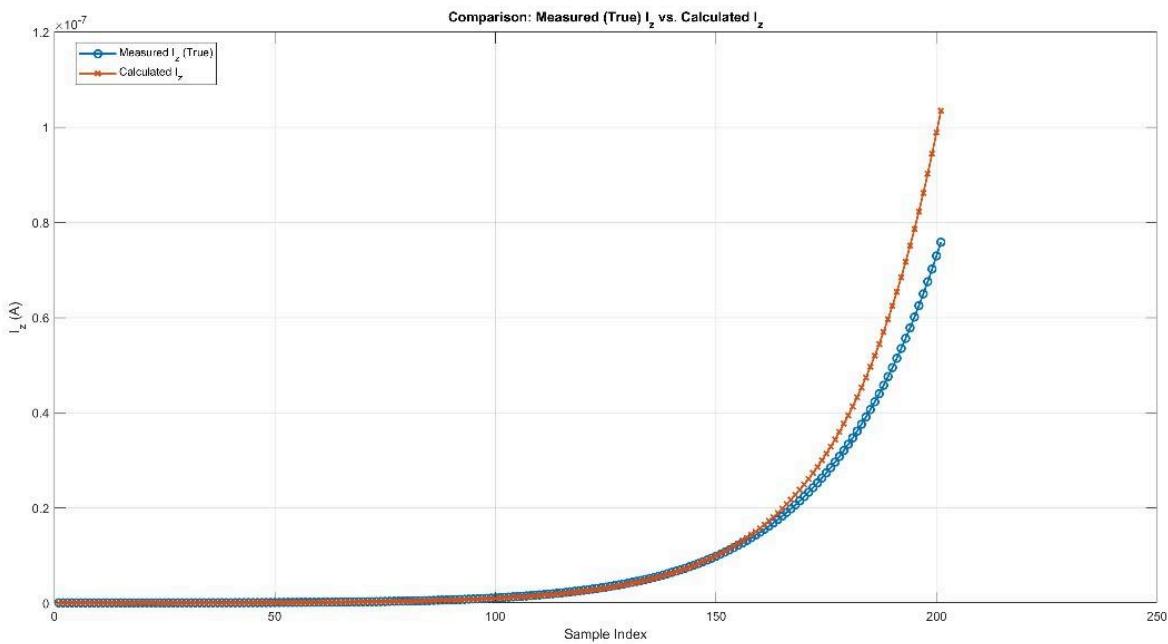
xlabel('Sample Index');
ylabel('I_z (A)');
title('Comparison: Measured (True) I_z vs. Calculated I_z');
legend('Measured I_z (True)', 'Calculated I_z', 'Location','best');

% ===== 4b) Plot #2: Iz_meas vs. (Ix1 * Ix2) =====
figure;
plot(Ix1Ix2, Iz_meas, 'o-', 'LineWidth', 1.5);
grid on;
xlabel('I_{x1} \times I_{x2} (A^2)');
ylabel('I_z (A) - Measured/True');
title('Plot of I_z, measured vs. (I_{x1} \times I_{x2})');
```

I_z vs. $I_{x1} * I_{x2}$: This plot verifies the quadratic relationship between input and output currents, confirming that the circuit follows the expected squaring behavior



Comparison of Measured (I_z) vs. Calculated (I_z) Output Current. Small discrepancies can be observed at lower and upper input ranges, likely due to parasitic effects and transistor mismatch.



MAPE Calculation:

The Mean Absolute Percentage Error (MAPE) was computed to quantify the accuracy of the circuit.

The error was calculated as:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{I_{z,calc} - I_{z,meas}}{I_{z,meas}} \right| \times 100$$

The obtained **MAPE was 22.40%**, indicating good agreement with theoretical values, with minor deviations at the extreme ends of the current range.

1. Explanation of how V_0 was determined:

V_0 was determined through a .DC analysis in Cadence Virtuoso, where V_{DS} was plotted against V_0 over a range of 0V to 3.2V. The goal was to identify a suitable V_0 range where all transistors (Q1, Q2, Q3, Q4) in the translinear loop and current sources operate in the weak-inversion saturation region. The MATLAB script further processed exported simulation data to verify transistor operating points.

2. Confirmation that all devices are operating in the weak-inversion saturation region:

Condition for Weak-Inversion Saturation:

A MOSFET operates in the **weak-inversion saturation region** when:

$$V_{DS} > 4V_T$$

where:

- V_{DS} is the **drain-to-source voltage** of the transistor,
- V_T is the **thermal voltage**, given by:

$$V_T = \frac{kT}{q} \approx 25mV \text{ at room temperature (300K).}$$

Thus, the minimum required drain-to-source voltage for weak-inversion saturation is:

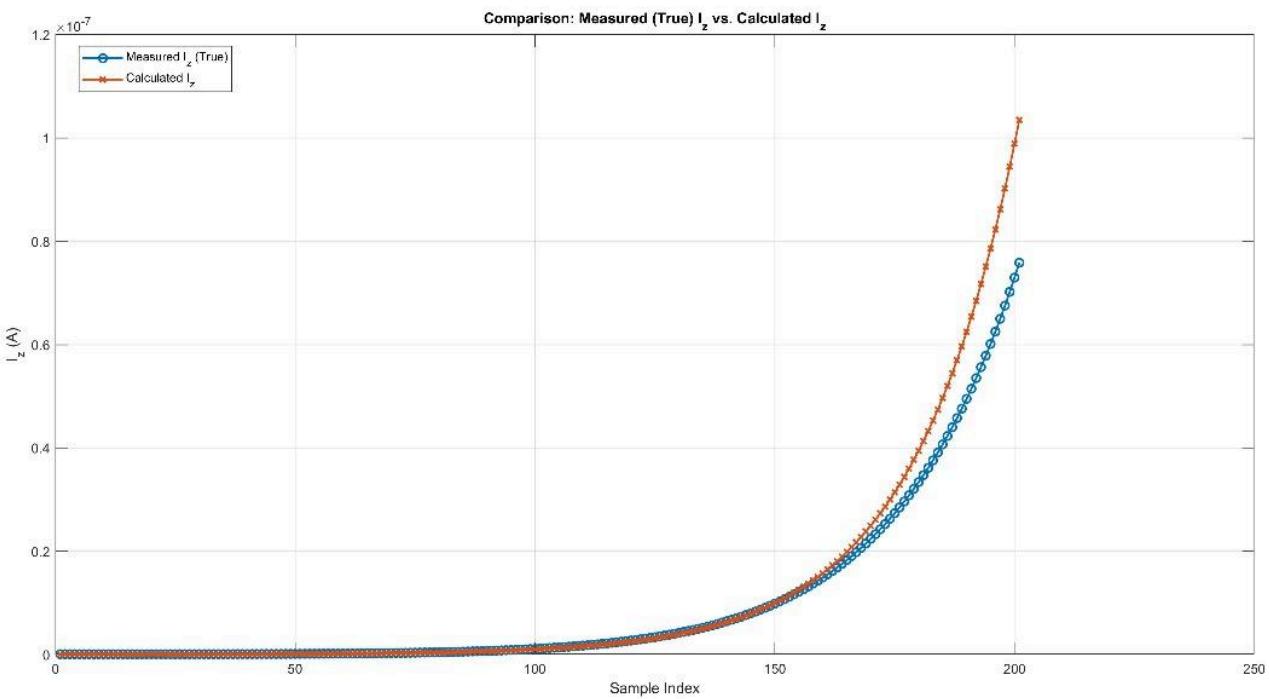
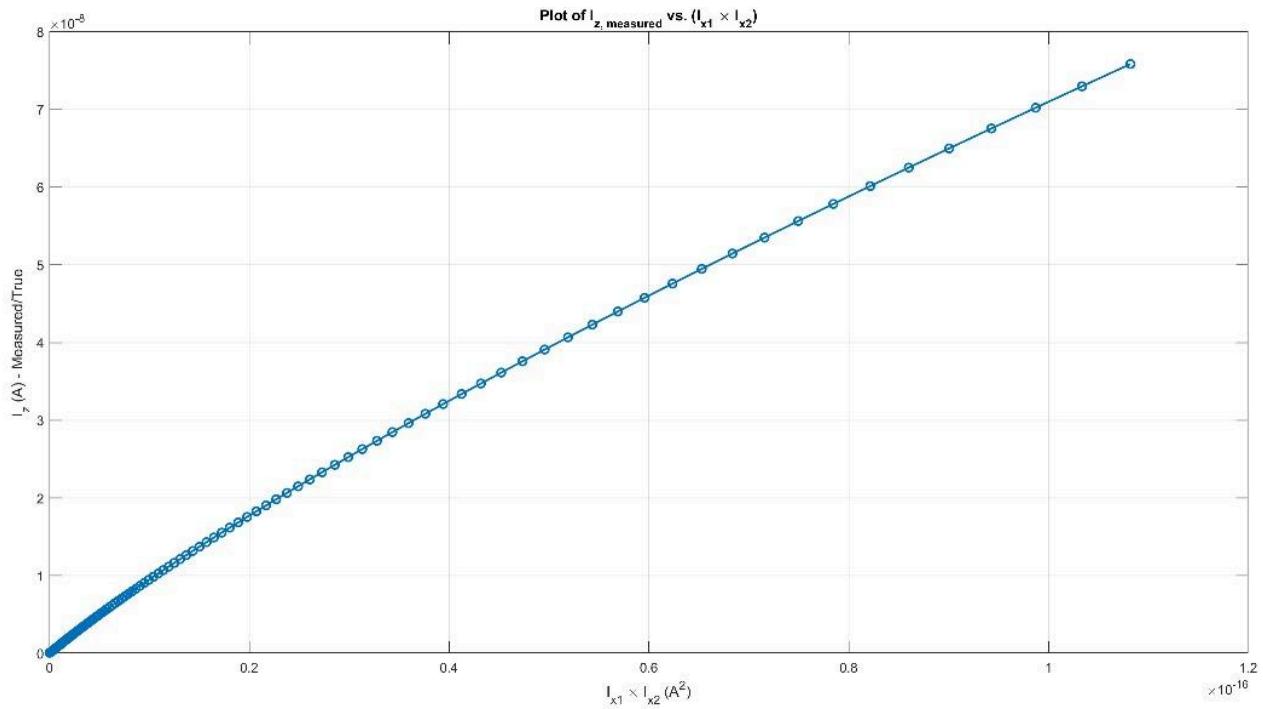
$$V_{DS} > 4 \times 25mV = 100mV.$$

From your results, the key operating points were:

- $V_{DS} = 0.103048V, V_0 = 0.455V$
- $V_{DS} = 0.116533V, V_0 = 0.56V$

Since these values satisfy $V_{DS} > 100mV$, we confirm that all transistors are in the weak-inversion saturation region.

3. I_{out} vs. I_{out} plots for both circuit simulations and true values



4. Computation of MAPE and discussion of results

The error was calculated as:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{I_{z,calc} - I_{z,meas}}{I_{z,meas}} \right| \times 100$$

Computed MAPE = 22.40%, indicating good agreement overall, but some deviations at extreme input ranges.

Discussion of Discrepancies:

- Could be due to subthreshold leakage and numerical noise in simulations.
- May be attributed to transistor mismatch and second-order effects.