EE236: Experiment 7
MOSFET I/V Characteristics and Device Parameters

Param Rathour, 190070049
Spring Semester, 2021-22

Contents

1 Overview of the experiment

This report contains my approach to the experiment, the circuit's design with the relevant circuit designs and output plots & values.

1.1 Aim of the experiment

- To plot I/V characteristics of Zener Diode and determine its break-down voltage.
- To plot and understand I_D/V_{DS} characteristics of NMOS at different V_{GS} and calculate r_0 (resistance saturation region) and V_A (early voltage) considering channel length modulation.
- To plot and understand I_D/V_{GS} characteristics of NMOS in saturation region and obtain the parameter V_T, g_m and subthreshold slope
- To measure small signal transconductance g_m

1.2 Methods

1.2.1 Zener Diode

First, I/V characteristics of Zener Diode is plotted and then break-down voltage was calculated.

1.2.2 NMOS

First, I_D/V_{GS} characteristics of NMOS at $V_{DS} = 200mV$ was plotted and the parameters were calculated.
Then, I_D/V_{DS} characteristics of NMOS at different V_{GS} was plotted and the parameters were calculated & compared.
Then, I_D/V_{GS} characteristics of NMOS at saturation was plotted and the parameters were calculated.
Lastly, Small Signal Transconductance and Voltage Gain Ratio was calculated using V_{out} and V_{in}
All figures & parameters were calculated using hands-on experiment and exporting the results to python

2 Design

The circuits designed for all parts are shown below with their physical counterparts

2.1 Zener Diode

The power supply was varied from $-7V$ to $2V$ and the voltage & current across Zener Diode was measured (Circuit 1).

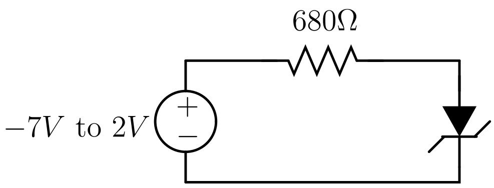


Figure 1: I/V measurement of Zener

2.2 NMOS

2.2.1 I_D/V_{GS} characteristics of NMOS at $V_{DS} = 200mV$

V_{DS} was kept as $200mV$ and V_{GS} was varied from 0 to $5V$ (Circuit 2a). Values of I_D and V_{GS} were noted. V_T is the $x-$intercept

$$x - \text{intercept} = \frac{-(y - \text{intercept})}{slope}$$

Both Linear Fit and Poly Fit was tried, Poly Fit gave best results.
Subthreshold Slope $= \left(\frac{\partial \log I_D}{\partial V_{GS}}\right)^{-1}$ in mV/decade

2.2.2 I_D/V_{DS} characteristics of NMOS at different V_{GS}

V_{GS} was kept as $2.5V$ and V_{DS} was varied from 0 to $5V$ (Circuit 2b). Values of I_D and V_{DS} were noted.
In saturation region slope I_D/V_{DS} is minimum $\rightarrow r_0 = \frac{1}{\text{minimum slope}}$
Early voltage can be calculated by taking the mean of the $x-$intercepts of the extrapolated lines from saturation regions.

$$x - \text{intercept} = \frac{-(y - \text{intercept})}{slope}$$

and slope, $y-$intercepts were calculated using Python's `Scipy` package This graph is a straight line in some range of I, V. Also, $I_L = 0$ for Dark characteristics.
We can calculate its slope to get η and y-intercept by interpolating that straight line.

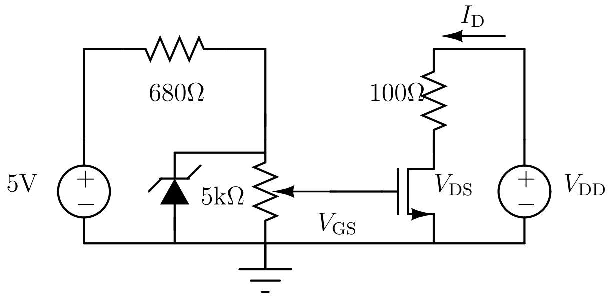
2.2.3 I_D/V_{GS} characteristics of NMOS at saturation

V_{GS} varied by varying V_{DD} from 0 to $5V$ (Circuit 2b). Values of I_D and V_{GS} were noted.

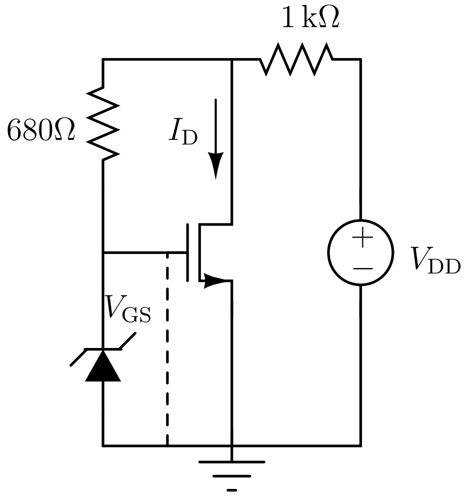
2.2.4 Small Signal Transconductance NMOS

$V_{GS} = 3V$ & $V_{DS} = 5V$, now a sine wave generated usinf AFG with amplitude $100mV_{pp}$ and frequency $1kHz$ was applied at the input to find output voltage.

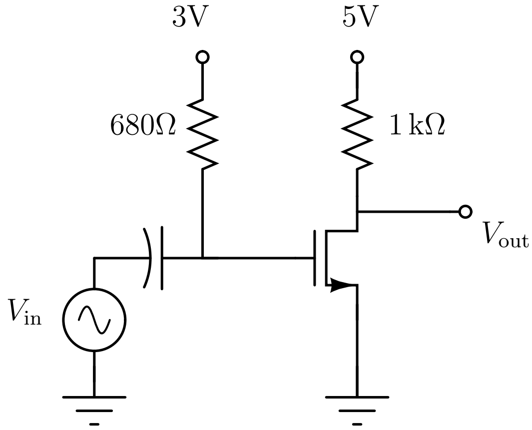
$$|A_v = g_m \cdot R_D \quad \text{where } R_D = 1000\Omega$$



(a) I_D/V_{DS} measurement of NMOS

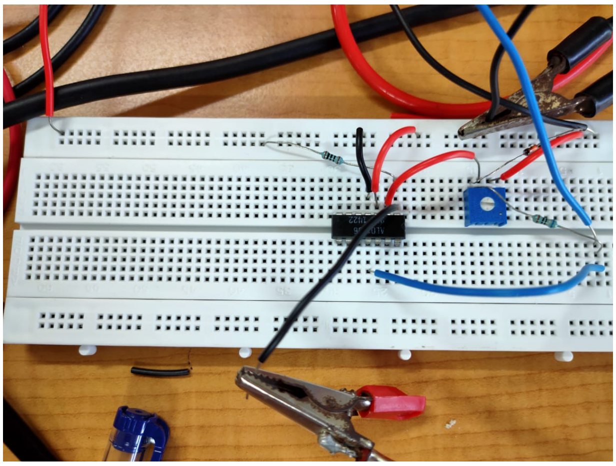


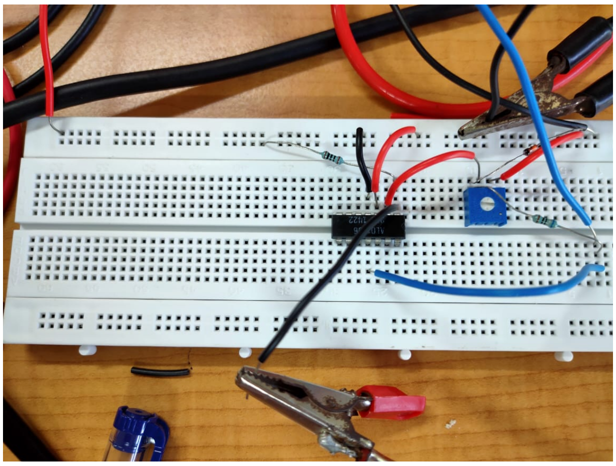
(b) I_D/V_{GS} measurement of NMOS



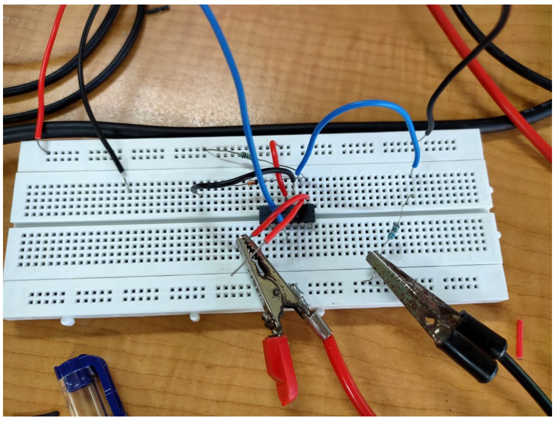
(c) Small Signal Transconductance

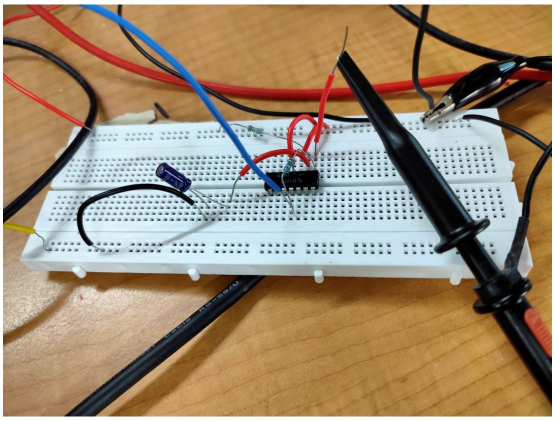
Figure 2: Design circuits



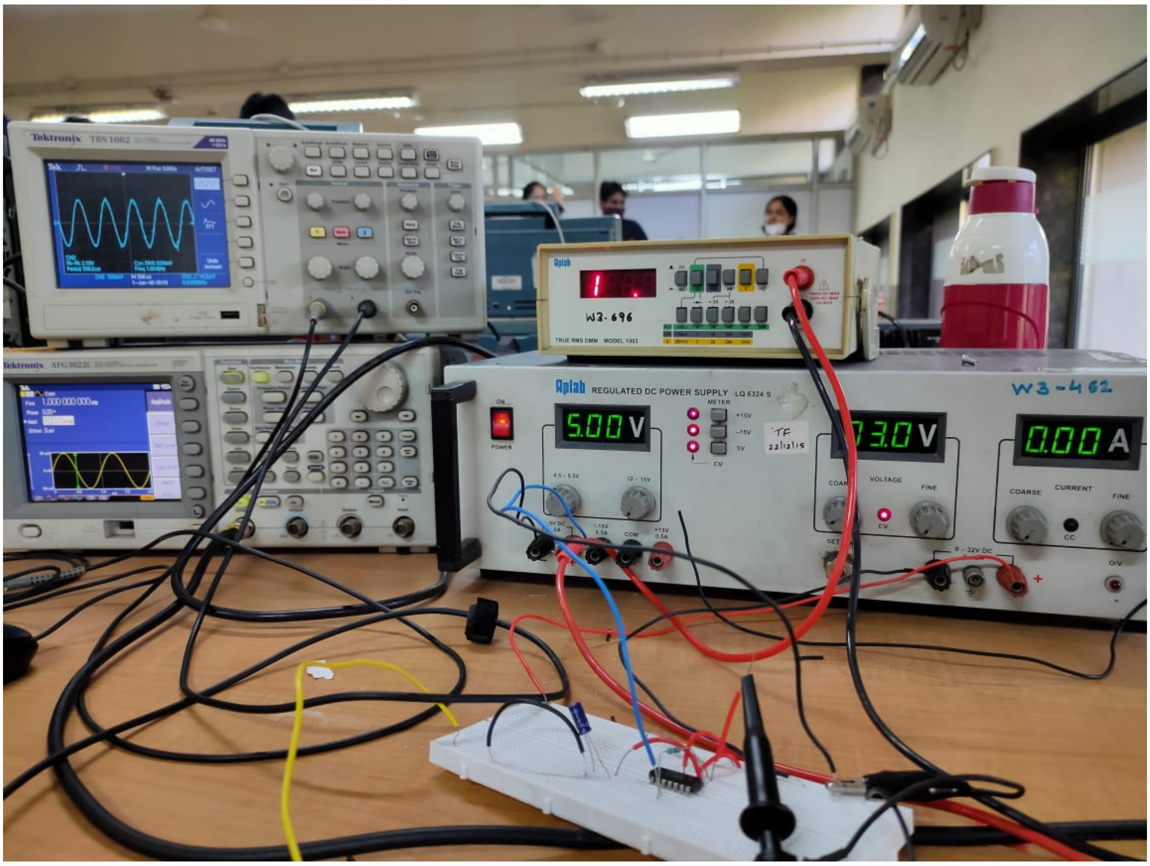


(a) I_D/V_{GS} characteristics of NMOS at $V_{DS} = 200mV$ (b) I_D/V_{DS} measurement of NMOS at different V_{GS}





(c) I_D/V_{GS} characteristics of NMOS at saturation (d) Small Signal Transconductance



(e) Small Signal Transconductance

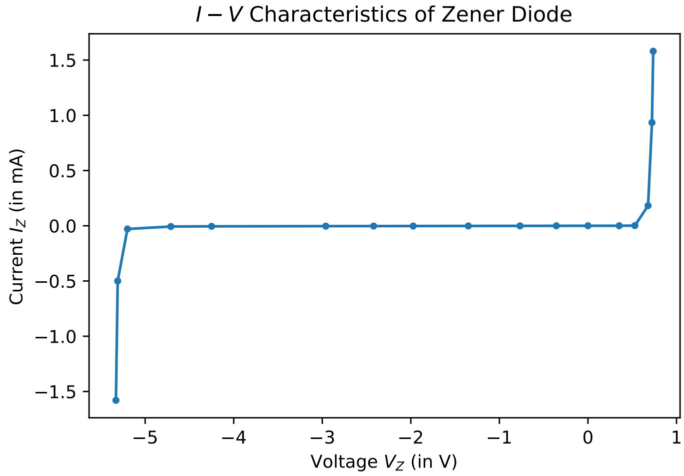
Figure 3: Design circuits

3 Experimental Results

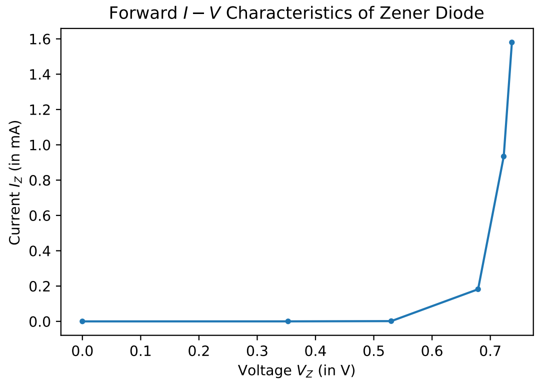
Please check my code section for exact measured values.

3.1 Plots

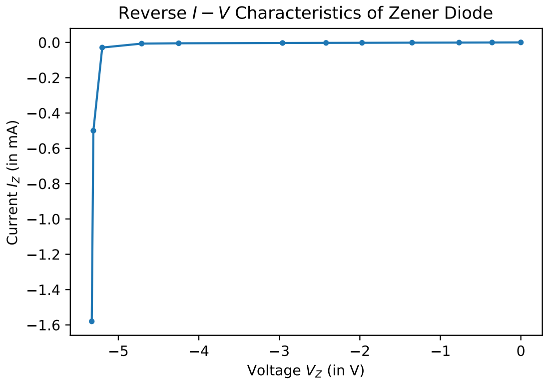
3.1.1 I/V characteristics of Zener Diode



(a) I/V



(b) Forward I/V

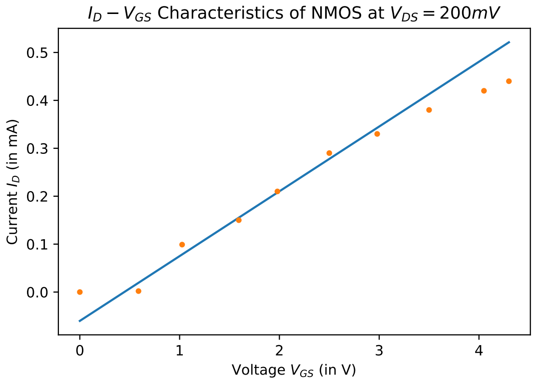


(c) Reverse I/V

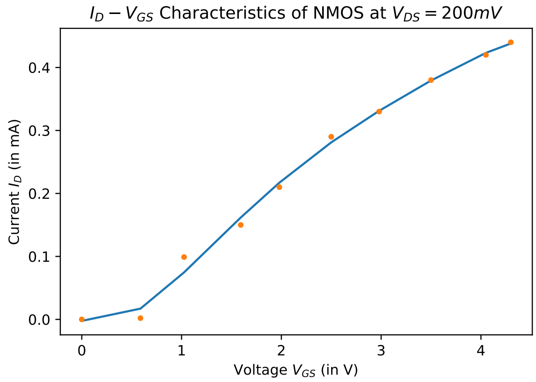
Figure 4: I/V characteristics of Zener

Break-down Voltage $= 5.31V$

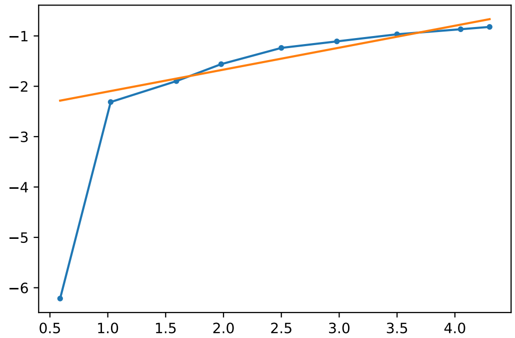
3.1.2 I_D/V_{GS} characteristics of NMOS at $V_{DS} = 200mV$



(a) $V_{DS} = 200mV$ straight line fit



(b) $V_{DS} = 200mV$ polynomial (degree = 5) line fit

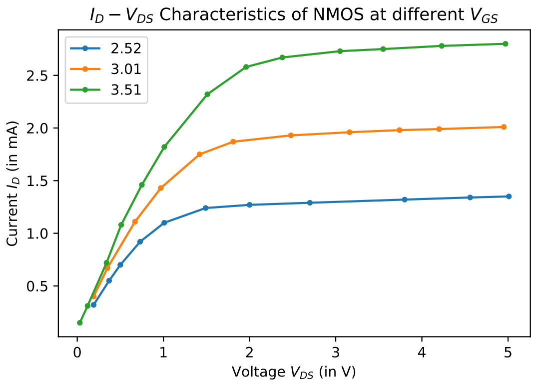


(c) $\log I_D/V_{GS}$ characteristics

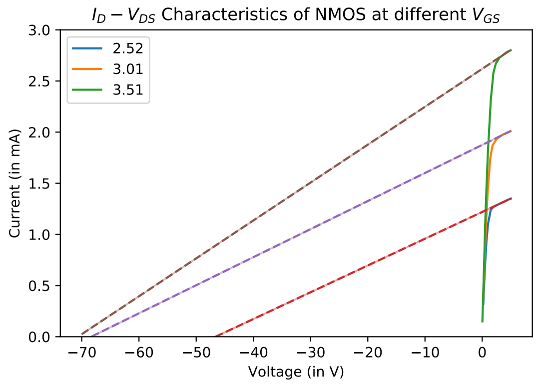
Figure 5: I_D/V_{GS} characteristics of NMOS

Using Polyfit, $V_T = 0.4452396758207092V$ g_m max at $1.593V = 0.1501103670847114m\mho$ Subthreshold Slope $= 111.99463374192149mV/\text{decade}$

3.1.3 I_D/V_{DS} characteristics of NMOS at different V_{GS}



(a)



(b) with extrapolated lines

Figure 6: I_D/V_{DS} characteristics of NMOS at Different V_{GS}

As magnitude of V_{GS} increases, magnitude of I_D increases.
$r_o = [45, 37.5, 37]k\Omega$ for $V_G S = [2.52, 3.01, 3.51]V$ $V_A = [-46.60981597 - 68.34507439 - 70.66927649]V$ for $V_G S = [2.52, 3.01, 3.51]V$
All extrapolated lines meet at $V_{A_{\text{mean}}} = 61.87472228159754V$

3.1.4 I_D/V_{GS} characteristics of NMOS at saturation

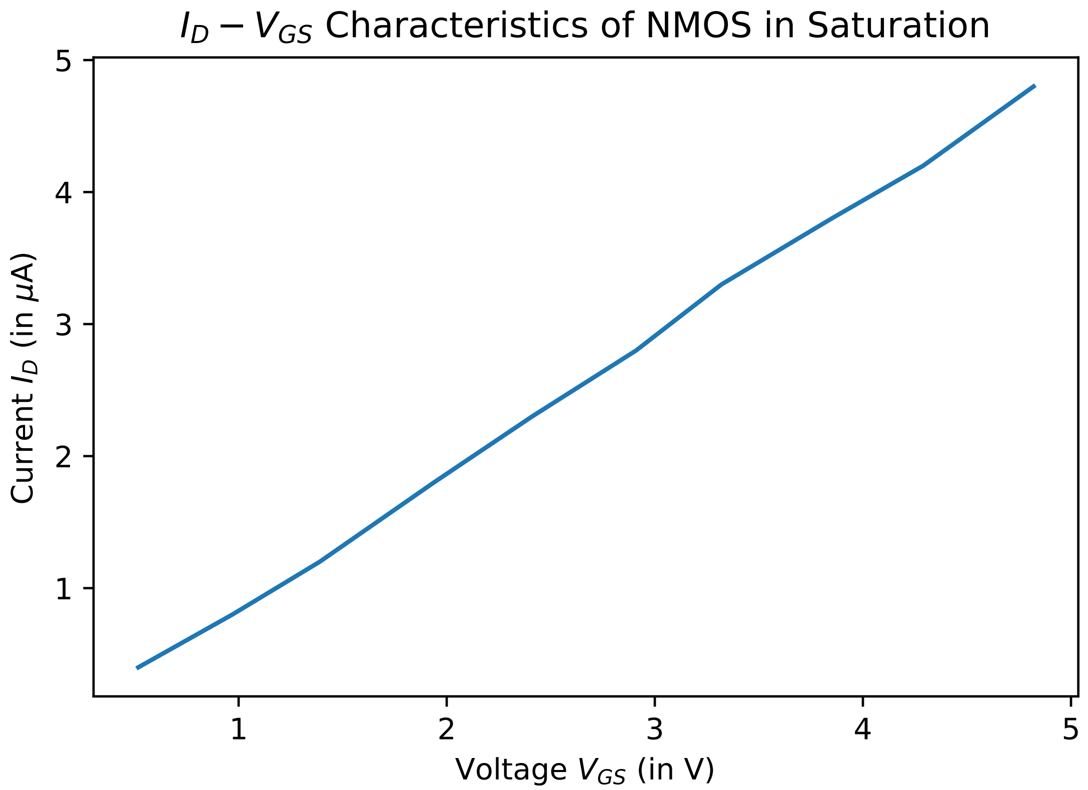
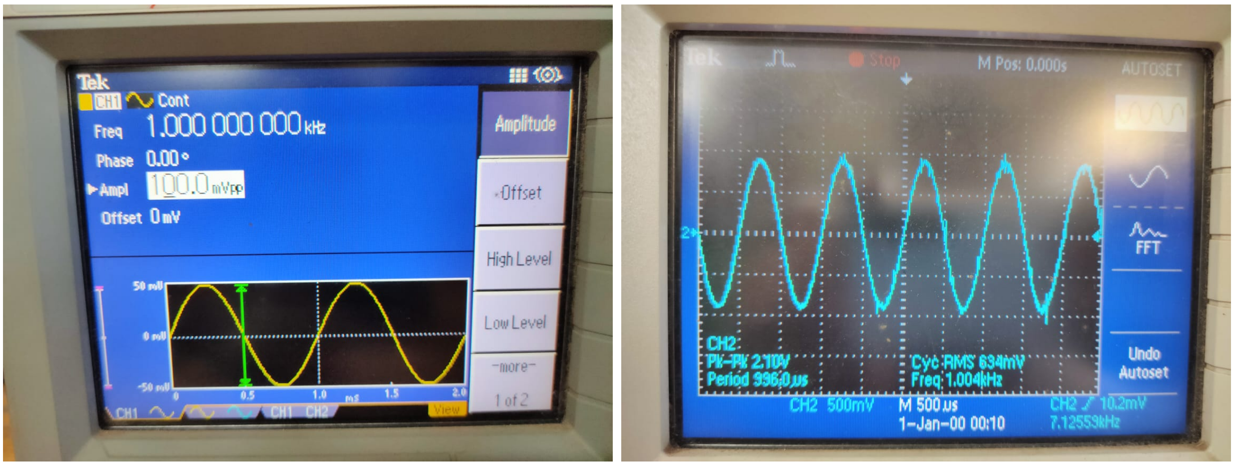


Figure 7: I_D/V_{GS} characteristics of NMOS at saturation

As magnitude of V_{GS} increases (neglecting the small V_{GS} values as NMOS is saturated), magnitude of I_D increases linearly

3.1.5 Small Signal Transconductance NMOS



(a) V_{in} (b) V_{out}

Figure 8: Small Signal Transconductance

$g_m = 21.0m\mho$, as $A_v = 21$

3.2 Python Code for Plots and Calculations

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import math
import pandas as pd
import bisect
```

3.2.1 I/V characteristics of Zener Diode

```
V_S = [-6.8, -6, -5.5, -4.9, -4.4, -3.1, -2.5, -2.1, -1.4, -0.7, -0.3,
       0, 0.3, 0.6, 1, 1.5, 2]
V_Z = [-5.33, -5.31, -5.2, -4.71, -4.25, -2.96, -2.42, -1.973,
       -1.351, -0.767, -0.357, 0, 0.353, 0.530, 0.679, 0.723, 0.737]
I_Z = [-1.58, -0.5, -29.1e-3, -6.8e-3, -5e-3, -3.3e-3, -2.7e-3, -2.3e-3,
       -1.5e-3, -1e-3, -0.5e-3, 0, 0.1e-3, 1.5e-3, 182e-3, 0.934, 1.58]

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{Z}$ (in V)')
ax1.set_ylabel('Current $I_Z$ (in mA)')
ax1.set_title('$I-V$ Characteristics of Zener Diode')
ax1.plot(V_Z, I_Z, '-o', markersize = 3)
fig1.set_dpi(150)
fig1.savefig('01.pdf')

fig2, ax2 = plt.subplots()
ax2.set_xlabel('Voltage $V_{Z}$ (in V)')
ax2.set_ylabel('Current $I_Z$ (in mA)')
ax2.set_title('Forward $I-V$ Characteristics of Zener Diode')
ax2.plot(V_Z[-6:], I_Z[-6:], '-o', markersize = 3)
fig2.set_dpi(150)
fig2.savefig('02.pdf')
```

```python
fig3, ax3 = plt.subplots()
ax3.set_xlabel('Voltage $V_{Z}$ (in V)')
ax3.set_ylabel('Current $I_Z$ (in mA)')
ax3.set_title('Reverse $I-V$ Characteristics of Zener Diode')
ax3.plot(V_Z[:-5], I_Z[:-5], '-o', markersize = 3)
fig3.set_dpi(150)
fig3.savefig('03.pdf')
```

3.2.2 I_D/V_{GS} characteristics of NMOS at $V_{DS} = 200mV$

3.2.2.1 Linear Fit

```python
V_GS = np.array([0, 0.588, 1.025, 1.593, 1.98, 2.5, 2.98, 3.5, 4.05, 4.3])
I_D = np.array([0, 2e-3, 99e-3, 0.15, 0.21, 0.29, 0.33, 0.38, 0.42, 0.44])

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of NMOS at $V_{DS}=200mV$')
fig1.set_dpi(150)
slope, y_intercept, r_value, p_value, std_err = stats.linregress(V_GS[1:-3],
                                                                 I_D[1:-3])
ax1.plot(V_GS, slope*V_GS + y_intercept, '-o', markersize=0.01)
ax1.plot(V_GS, I_D, 'o', markersize=3)
# print(y_intercept)
V_T = -y_intercept/slope
print("V_T =", V_T)
g_m = slope
print("g_m =", g_m)  # in m mhos
fig1.savefig('111.pdf')
```

3.2.2.2 Polynomial Fit

```python
V_GS = np.array([0, 0.588, 1.025, 1.593, 1.98, 2.5, 2.98, 3.5, 4.05, 4.3])
I_D = np.array([0, 2e-3, 99e-3, 0.15, 0.21, 0.29, 0.33, 0.38, 0.42, 0.44])

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of NMOS at $V_{DS}=200mV$')
fig1.set_dpi(150)

p = np.poly1d(np.polyfit(V_GS, I_D, deg = 5))
ax1.plot(V_GS, [p(i) for i in V_GS], '-o', markersize=0.01)
ax1.plot(V_GS, I_D, 'o', markersize=3)
V_T = -y_intercept/slope
print("V_T =", V_T)
g_m = slope
print("g_m =", g_m)  # in m mhos
fig1.savefig('112.pdf')

d = np.poly1d.deriv(p)
max_g_m = -np.inf
max_V_GS = 0
for i in V_GS:
```

```
    s = d(i)
    if s > max_g_m:
        max_g_m = s
        max_V_GS = i

print(max_V_GS, max_g_m)

fig2, ax2 = plt.subplots()
ax2.plot(V_GS[1:], np.log(I_D[1:]), '-o', markersize=3)
fig2.set_dpi(150)

slope, y_intercept, r_value, p_value, std_err = stats.linregress(V_GS[2:],
                                                                 np.log(I_D[2:]))
ax2.plot(V_GS[1:], slope*V_GS[1:] + y_intercept, '-o', markersize=0.01)
ax2.set_xlabel('Voltage $V_{GS}$ (in V)')
ax2.set_ylabel('Current $\log{I_D}$')
ax2.set_title('$\log{I_D}-V_{GS}$ Characteristics of NMOS at $V_{DS}=200mV$')
fig2.savefig('12.pdf')
SS = 1/((np.log(I_D[2])-np.log(I_D[1]))/(1000*(V_GS[2]-V_GS[1])))
print(SS)
```

3.2.3 I_D/V_{DS} characteristics of NMOS at different V_{GS}

```
V_GS = [2.52, 3.01, 3.51]
V_DS = [[0.19, 0.37, 0.5, 0.73, 1.01, 1.49, 2, 2.7, 3.8, 4.56, 5.01],
        [0.19, 0.35, 0.67, 0.97, 1.42, 1.81, 2.48, 3.16, 3.74, 4.2, 4.95],
        [0.03, 0.12, 0.34, 0.51, 0.75, 1.01, 1.51, 1.96, 2.38, 3.05, 3.55, 4.23, 4.97]]
I_D = [[0.32, 0.55, 0.7, 0.92, 1.1, 1.24, 1.27, 1.29, 1.32, 1.34, 1.35],
       [0.4, 0.67, 1.11, 1.43, 1.75, 1.87, 1.93, 1.96, 1.98, 1.99, 2.01],
       [0.15, 0.31, 0.72, 1.08, 1.46, 1.82, 2.32, 2.58, 2.67, 2.73, 2.75, 2.78, 2.8]]
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{DS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{DS}$ Characteristics of NMOS at different $V_{GS}$')

for i in range(len(V_GS)):
    ax1.plot(V_DS[i], I_D[i], '-o', markersize=3)
ax1.legend(V_GS)
fig1.set_dpi(150)
fig1.savefig('21.pdf')

slope = []
for i in range(len(V_GS)):
    slope.append((I_D[i][-1]-I_D[i][-2])/(V_DS[i][-1]-V_DS[i][-2]))
r_0= [1/i for i in slope]
print("r_0", r_0)

y_intercepts = np.zeros(len(V_GS))
slope_fit = np.zeros(len(V_GS))
for i in range(len(V_GS)):
    slope, intercept, r_value, p_value, std_err = stats.linregress(
                                                        V_DS[i][-4:],I_D[i][-4:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)
```

```
V_A = np.zeros(len(V_GS))
R_0 = np.zeros(len(V_GS))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([0,3])
ax1.set_title('$I_D-V_{DS}$ Characteristics of NMOS at different $V_{GS}$')
fig1.set_dpi(150)
intersect = -70
x = np.arange(intersect,5,0.01)
for i in range(len(V_GS)):
    ax1.plot(V_DS[i], I_D[i], '-o', markersize=0.01)
for i in range(len(V_GS)):
    ax1.plot(x, (slope_fit[i]*x+y_intercepts[i]), '--o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
    R_0[i] = 1/slope_fit[i]
print(V_A)
print(np.mean(V_A))
print(R_0)
ax1.legend(V_GS)
fig1.savefig('22.pdf')
```

3.2.4 I_D/V_{GS} characteristics of NMOS at saturation

```
V_GS = np.array([0.518, 0.97, 1.39, 1.938, 2.41, 2.91, 3.32, 3.85, 4.29, 4.82])
I_D = np.array([0.4, 0.8, 1.2, 1.8, 2.3, 2.8, 3.3, 3.8, 4.2, 4.8])

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in $\mu$A)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of NMOS in Saturation')
# ax1.legend(V_GS)
fig1.set_dpi(150)
slope, y_intercept, r_value, p_value, std_err = stats.linregress(
                                                     V_GS[-2:], I_D[-2:])
# ax1.plot(V_GS, slope*V_GS + y_intercept, '-o', markersize=0.01)
ax1.plot(V_GS, I_D, '-o', markersize=0.01)
print(y_intercept)
V_T = -y_intercept/slope
print("V_T =", V_T)
g_m = slope
print("g_m =", g_m)  # in m mhos
fig1.savefig('31.pdf')
```

3.2.5 Small Signal Transconductance NMOS

```
A_v = 2.1/0.1
R_D = 1000
g_m = A_v/R_D*1000 # in m mhos
print(A_v, R_D, "\ng_m = ", g_m)
```

4 Experiment completion status

Completed everything in Lab successfully.

5 Questions for reflection

Please, also check by observations section for this.

- As Positive Body voltage, V_{SB} decreases so, V_T increases
  As Negative Body voltage, V_{SB} increases so, V_T decreases
- The subthreshold slope is important as a steep subthreshold slope means a faster switch between ON and OFF for a switching device
- An accurate method to calculate V_T, is to approximate $I - V_G$ using polynomials (see code section)
- To find V_T, we can use $\sqrt{I_D} - V_G$ characteristics.

Overall, the lab was good. Thanks!