EE 618: CMOS Analog VLSI Design

Assignment 1

Student Name: Bhuvansh Goyal
Roll Number: 22B3908
Instructor: Prof. Rajesh Zele
Submission Date: 20 Aug 2025

Contents

1 Characterizing the MOS devices

1.1 I_{sd} vs V_{sg} for PMOS

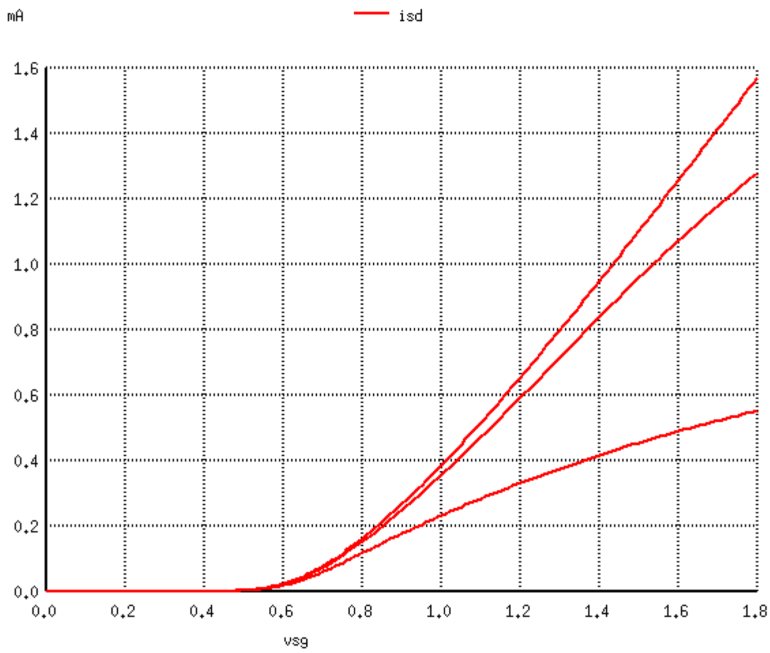

Figure 1: I_{sd} vs V_{sg} for $V_{sd} = 0.2$, 0.6, 1.0 V

1.2 I_{sd} vs V_{sd} for PMOS

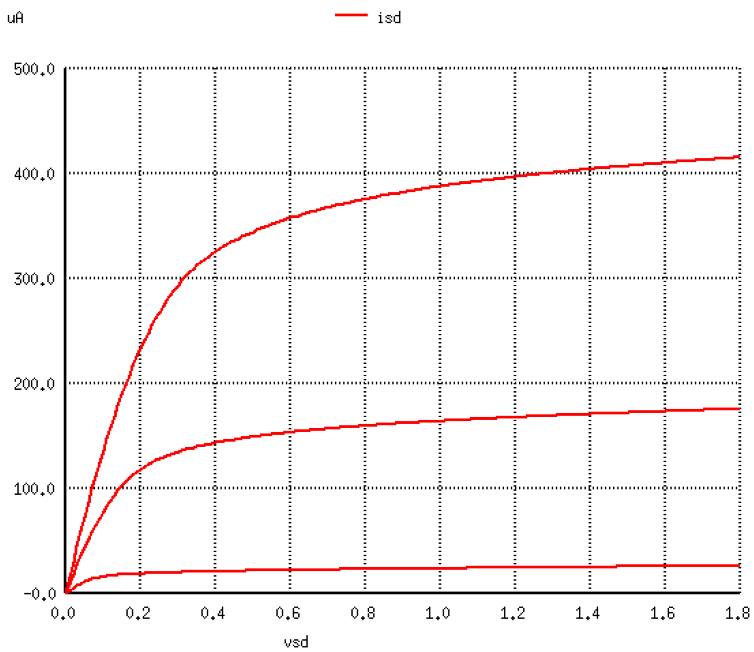

Figure 2: I_{sd} vs V_{sd} for $V_{sg} = 0.6$, 0.8, 1.0 V

1.3 I_{ds} vs V_{gs} for NMOS

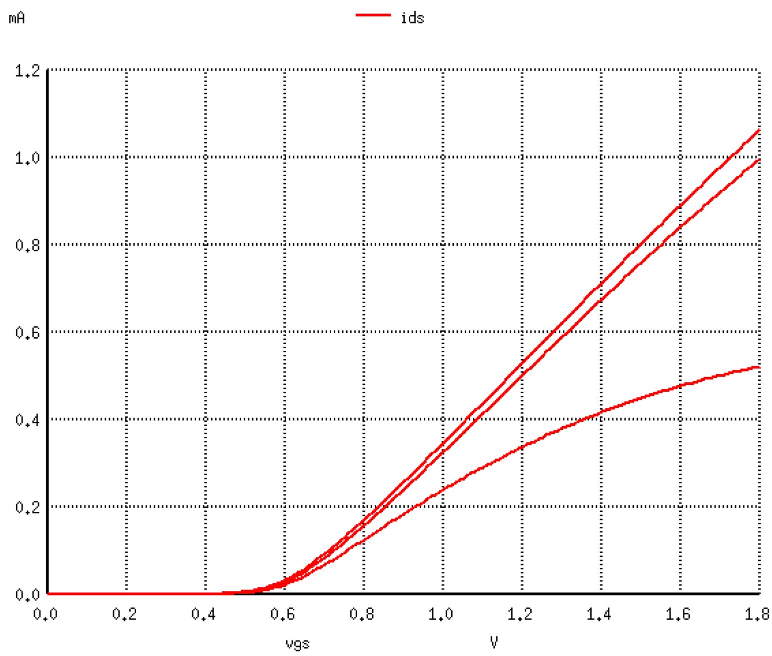

Figure 3: I_{ds} vs V_{gs} for V_{ds} = 0.2, 0.6, 1.0 V

1.4 I_{ds} vs V_{ds} for NMOS

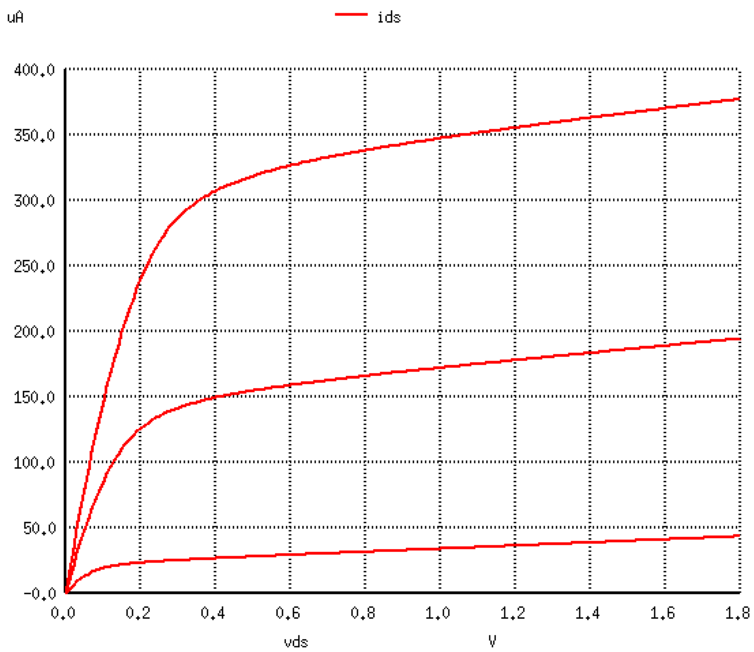

Figure 4: I_{ds} vs V_{ds} for V_{gs} = 0.6, 0.8, 1.0 V

1.5 g_{mp} Calculation

g_{mp} is calculated at $V_{sg} = 0.9\,\text{V}$ and $V_{sd} = 1.0\,\text{V}$. By definition,

$$g_{mp} = \left.\frac{\partial I_{sd}}{\partial V_{sg}}\right|_{V_{sd}\,\text{constant}}$$

The derivative is taken from the I_{sd}–V_{sg} curve at $V_{sd} = 1.0\,\text{V}$, using the `deriv` function in ngspice. The result obtained is:

$$g_{mp} = 1.125\,\text{mS}$$

1.6 g_{mn} Calculation

g_{mn} is calculated at $V_{gs} = 0.9\,\text{V}$ and $V_{ds} = 1.0\,\text{V}$. By definition,

$$g_{mn} = \left.\frac{\partial I_{ds}}{\partial V_{gs}}\right|_{V_{ds}\,\text{constant}}$$

The derivative is taken from the I_{ds}–V_{gs} curve at $V_{ds} = 1.0\,\text{V}$, using the `deriv` function in ngspice. The result obtained is:

$$g_{mn} = 0.880\,\text{mS}$$

1.7 r_{outp} Calculation

r_{outp} is calculated at $I_{sd} = 400\,\mu\text{A}$. By definition,

$$r_{outp} = \frac{1}{\frac{\partial I_{sd}}{\partial V_{sd}}}$$

The derivative is taken from the I_{sd}–V_{sd} curve at $V_{sg} = 1.0\,\text{V}$ and $I_{sd} = 400\,\mu\text{A}$, using the `deriv` function in ngspice. The result obtained is:

$$r_{outp} = 27.132\,k\Omega$$

1.8 r_{outn} Calculation

r_{outn} is calculated at $I_{ds} = 350\,\mu\text{A}$. By definition,

$$r_{outn} = \frac{1}{\frac{\partial I_{ds}}{\partial V_{ds}}}$$

The derivative is taken from the I_{ds}–V_{ds} curve at $V_{gs} = 1.0\,\text{V}$ and $I_{ds} = 350\,\mu\text{A}$, using the `deriv` function in ngspice. The result obtained is:

$$r_{outn} = 24.348\,k\Omega$$

1.9 V_{thp} Calculation

V_{thp} is obtained by extrapolating the slope of the $\sqrt{I_{sd}}$–V_{sg} curve at $V_{sg} = 0.7\,\text{V}$, where $V_{sd} = 1.0\,\text{V}$. After this point, the curve bends down as the device enters the linear/triode region. The point where the extrapolated line meets the V_{sg} axis is taken as V_{thp}.

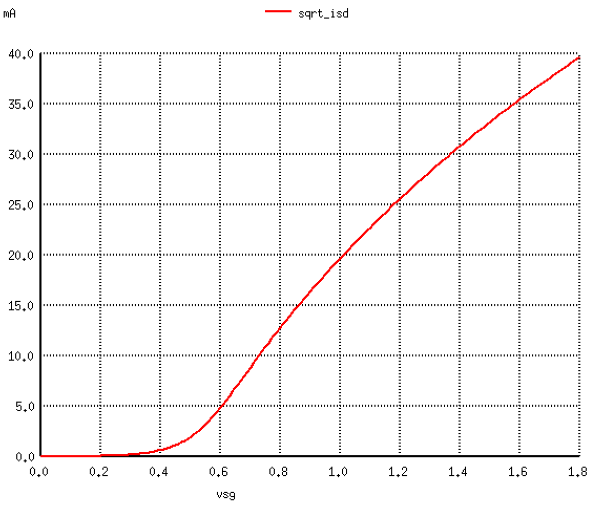

Figure 5: $\sqrt{I_{sd}}$ vs V_{sg} for $V_{sd} = 1.0\,\text{V}$

The value obtained is:

$$V_{thp} = 0.482\,\text{V}$$

1.10 V_{thn} Calculation

V_{thn} is obtained by extrapolating the slope of the $\sqrt{I_{ds}}$–V_{gs} curve at $V_{gs} = 0.7\,\text{V}$, where $V_{ds} = 1.0\,\text{V}$. After this point, the curve bends down as the device enters the linear/triode region. The point where the extrapolated line meets the V_{sg} axis is taken as V_{thn}.

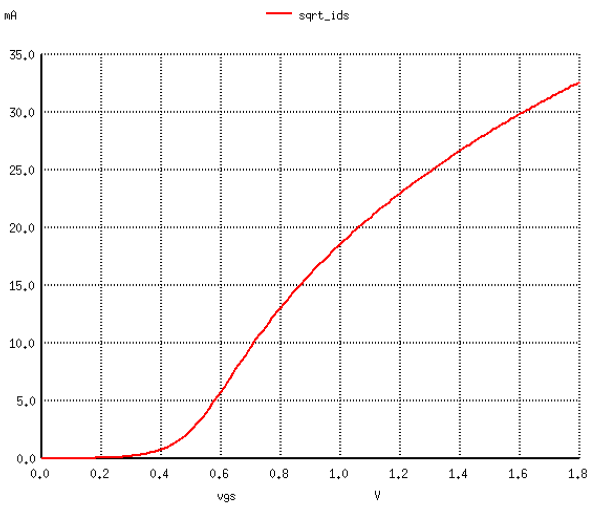

Figure 6: $\sqrt{I_{ds}}$ vs V_{gs} for $V_{ds} = 1.0\,\text{V}$

The value obtained is:

$$V_{thn} = 0.438\,\text{V}$$

1.11 λ_p Calculation

λ_p is calculated using the values of r_{outp}, I_{sd}, and V_{sd} at the same operating point where r_{outp} was obtained. The formula used is:

$$\lambda_p = \frac{1}{(I_{sd} \cdot r_{outp}) - V_{sd}}$$

Substituting the values from the r_{outp} calculation, the result obtained is:

$$\lambda_p = 0.104\,\mathrm{V}^{-1}$$

1.12 λ_n Calculation

λ_n is calculated using the values of r_{outn}, I_{ds}, and V_{ds} at the same operating point where r_{outn} was obtained. The formula used is:

$$\lambda_n = \frac{1}{(I_{ds} \cdot r_{outn}) - V_{ds}}$$

Substituting the values from the r_{outn} calculation, the result obtained is:

$$\lambda_n = 0.134\,\mathrm{V}^{-1}$$

1.13 $\mu_p C_{ox}$ Calculation

$\mu_p C_{ox}$ is calculated under the same conditions as g_{mp}, i.e., at $V_{sg} = 0.9\,\mathrm{V}$ and $V_{sd} = 1.0\,\mathrm{V}$. The formula used is:

$$\mu_p C_{ox} = \frac{g_{mp} \cdot L}{W \cdot (V_{sg} - V_{thp}) \cdot (1 + \lambda_p V_{sd})}.$$

After substituting the values, the result obtained is:

$$\mu_p C_{ox} = 62.712\ \mu\mathrm{A/V}^2$$

1.14 $\mu_n C_{ox}$ Calculation

$\mu_n C_{ox}$ is calculated under the same conditions as g_{mn}, i.e., at $V_{gs} = 0.9\,\mathrm{V}$ and $V_{ds} = 1.0\,\mathrm{V}$. The formula used is:

$$\mu_n C_{ox} = \frac{g_{mn} \cdot L}{W \cdot (V_{gs} - V_{thn}) \cdot (1 + \lambda_n V_{ds})}.$$

After substituting the values, the result obtained is:

$$\mu_n C_{ox} = 151.292\ \mu\mathrm{A/V}^2$$

2 DC Analysis

2.1 DC Transfer Curve of Inverter

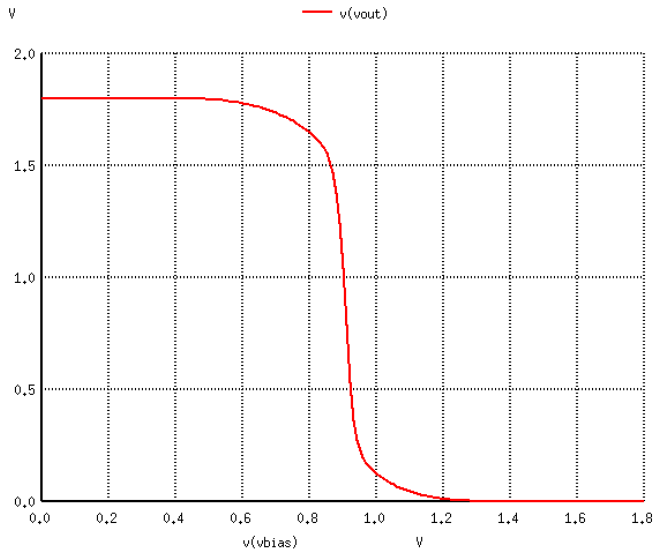

Figure 7: V_{out} vs V_{in} for inverter

2.2 Gain vs V_{in} Plot of Inverter

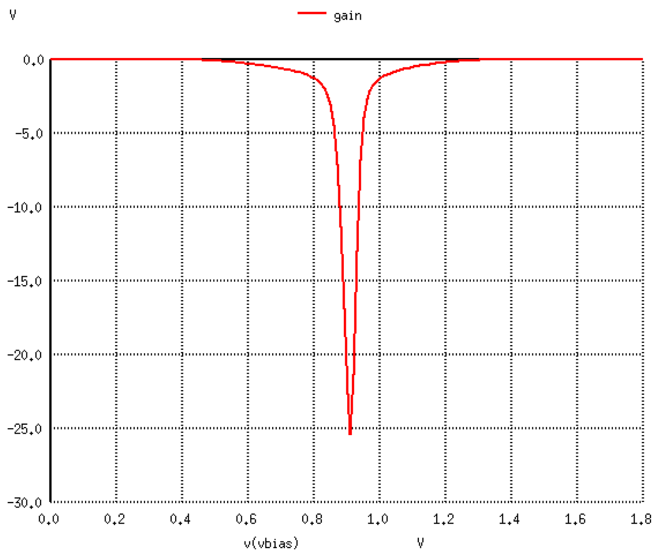

Figure 8: Gain vs V_{in} plot for inverter

From the above plot, here is the gain at $V_{in} = 0.9\,V$:

$$\text{Gain} = \frac{dV_{out}}{dV_{in}} = -22.636$$

3 Transient Analysis

3.1 Output and Input Waveforms

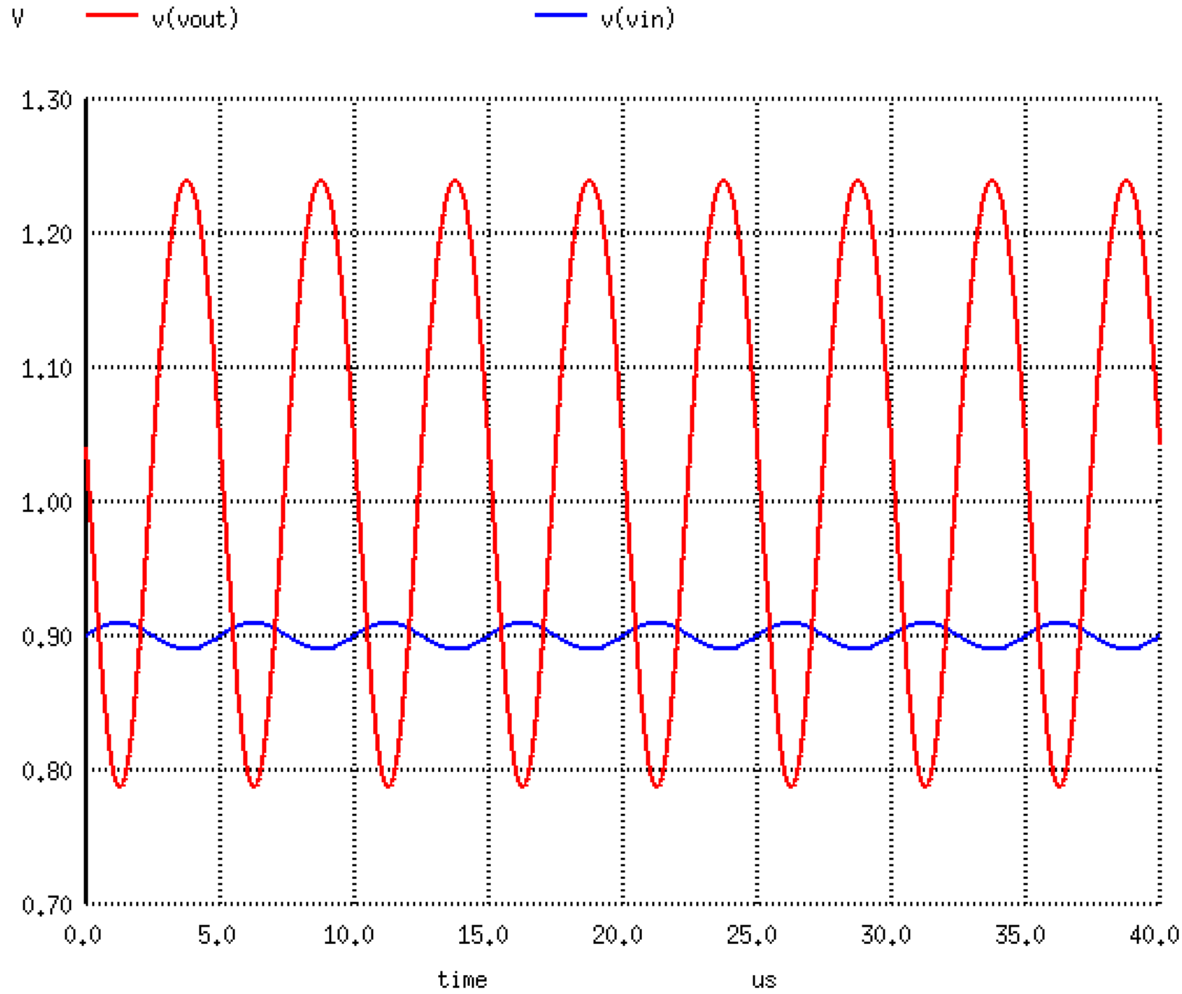

Figure 9: Transient waveforms of inverter showing V_{out} and V_{in}

The measured peak-to-peak output voltage is:

$$V_{out,pp} = 0.453\,V$$

The measured peak-to-peak input voltage is:

$$V_{in,pp} = 0.020\,V$$

Therefore, the gain observed from the transient analysis is:

$$\text{Gain} = \frac{V_{out,pp}}{V_{in,pp}} = -22.647$$

4 Code

4.1 NMOS

Listing 1: Ngspice netlist for NMOS characterization

```
.title NMOS Characterization

.include models-180nm

vds d 0 dc 0
vgs g 0 dc 0

mn d g 0 0 CMOSN W=2u L=0.18u

.control

run

set color0 = white
set color1 = black
set color2 = red

let gm_probe_vgs = 0.9
let gout_probe_ids = 350u
let vth_probe_vgs = 0.7

let k = 0
let w = 2u
let l = 0.18u

* ------------------ Ids vs Vgs @ Vds = 0.2V ------------------ *

alter vds dc 0.2
dc vgs 0 1.8 0.01

let ids = -i(vds)
let vgs = v(g)

plot ids vs vgs

* ------------------ Ids vs Vgs @ Vds = 0.6V ------------------ *

alter vds dc 0.6
dc vgs 0 1.8 0.01

let ids = -i(vds)
let vgs = v(g)

plot ids vs vgs

* ------------------ Ids vs Vgs @ Vds = 1V ------------------ *

alter vds dc 1
dc vgs 0 1.8 0.01

let ids = -i(vds)
let vgs = v(g)
```

```

let sqrt_ids = sqrt(ids)
let sqrt_slope = deriv(sqrt_ids)

plot sqrt_ids vs vgs

meas dc vth_probe_sqrt_slope find sqrt_slope when vgs = vth_probe_vgs
meas dc vth_probe_sqrt_ids find sqrt_ids when vgs = vth_probe_vgs

let vth = vth_probe_vgs - (vth_probe_sqrt_ids / vth_probe_sqrt_slope)
print vth

plot ids vs vgs

let slope = deriv(ids)
meas dc gm find slope when vgs = gm_probe_vgs

let k = (gm * l) / (w * (gm_probe_vgs - vth))

* ------------------ Ids vs Vds @ Vgs = 0.6V ------------------ *

alter vgs dc 0.6
dc vds 0 1.8 0.01

let ids = -i(vds)
let vds = v(d)

plot ids vs vds

* ------------------ Ids vs Vds @ Vgs = 0.8V ------------------ *

alter vgs dc 0.8
dc vds 0 1.8 0.01

let ids = -i(vds)
let vds = v(d)

plot ids vs vds

* ------------------ Ids vs Vds @ Vgs = 1V ------------------ *

alter vgs dc 1
dc vds 0 1.8 0.01

let ids = -i(vds)
let vds = v(d)

plot ids vs vds

let slope = deriv(ids)

meas dc gout find slope when ids = gout_probe_ids
meas dc gout_probe_vds find vds when ids = gout_probe_ids

let rout = 1 / gout
print rout

let lambda = 1 / ((rout * gout_probe_ids) - gout_probe_vds)
```

```
print lambda

let k = k / (1 + lambda)
print k

* ------------ Ids vs Vgs @ Vds = 0.2V, 0.6V, 1V ------------ *

dc vgs 0 1.8 0.01 vds 0.2 1 0.4

let ids = -i(vds)
let vgs = v(g)

plot ids vs vgs

* ------------ Ids vs Vds @ Vgs = 0.6V, 0.8V, 1V ------------ *

dc vds 0 1.8 0.01 vgs 0.6 1 0.2

let ids = -i(vds)
let vds = v(d)

plot ids vs vds

.endc

.end
```

4.2 PMOS

Listing 2: Ngspice netlist for PMOS characterization

```
.title PMOS Characterization

.include models-180nm

vdd vdd 0 dc 1.8
vsd vdd d dc 0.2
vsg vdd g dc 0

mp d g vdd vdd CMOSP W=7u L=0.18u

.control

run

set color0 = white
set color1 = black
set color2 = red

let gm_probe_vsg = 0.9
let gout_probe_isd = 400u
let vth_probe_vsg = 0.7

let k = 0
let w = 7u
let l = 0.18u

```

```
* ------------------ Isd vs Vsg @ Vsd = 0.2V ------------------ *

alter vsd dc 0.2
dc vsg 0 1.8 0.01

let isd = -i(vsd)
let vsg = v(vdd) - v(g)

plot isd vs vsg

* ------------------ Isd vs Vsg @ Vsd = 0.6V ------------------ *

alter vsd dc 0.6
dc vsg 0 1.8 0.01

let isd = -i(vsd)
let vsg = v(vdd) - v(g)

plot isd vs vsg

* ------------------ Isd vs Vsg @ Vsd = 1V ------------------ *

alter vsd dc 1
dc vsg 0 1.8 0.01

let isd = -i(vsd)
let vsg = v(vdd) - v(g)

let sqrt_isd = sqrt(isd)
let sqrt_slope = deriv(sqrt_isd)

plot sqrt_isd vs vsg

meas dc vth_probe_sqrt_slope find sqrt_slope when vsg = vth_probe_vsg
meas dc vth_probe_sqrt_isd find sqrt_isd when vsg = vth_probe_vsg

let vth = vth_probe_vsg - (vth_probe_sqrt_isd / vth_probe_sqrt_slope)
print vth

plot isd vs vsg

let slope = deriv(isd)
meas dc gm find slope when vsg = gm_probe_vsg

let k = (gm * l) / (w * (gm_probe_vsg - vth))

* ------------------ Isd vs Vsd @ Vsg = 0.6V ------------------ *

alter vsg dc 0.6
dc vsd 0 1.8 0.01

let isd = -i(vsd)
let vsd = v(vdd) - v(d)

plot isd vs vsd

* ------------------ Isd vs Vsd @ Vsg = 0.8V ------------------ *

```

```
alter vsg dc 0.8
dc vsd 0 1.8 0.01

let isd = -i(vsd)
let vsd = v(vdd) - v(d)

plot isd vs vsd

* ------------------ Isd vs Vsd @ Vsg = 1V ------------------ *

alter vsg dc 1
dc vsd 0 1.8 0.01

let isd = -i(vsd)
let vsd = v(vdd) - v(d)

plot isd vs vsd

let slope = deriv(isd)

meas dc gout find slope when isd = gout_probe_isd
meas dc gout_probe_vsd find vsd when isd = gout_probe_isd

let rout = 1 / gout
print rout

let lambda = 1 / ((rout * gout_probe_isd) - gout_probe_vsd)
print lambda

let k = k / (1 + lambda)
print k

* ------------- Isd vs Vsg @ Vsd = 0.2V, 0.6V, 1V ------------- *

dc vsg 0 1.8 0.01 vsd 0.2 1 0.4

let isd = -i(vsd)
let vsg = v(vdd) - v(g)

plot isd vs vsg

* ------------- Isd vs Vsd @ Vsg = 0.6V, 0.8V, 1V ------------- *

dc vsd 0 1.8 0.01 vsg 0.6 1 0.2

let isd = -i(vsd)
let vsd = v(vdd) - v(d)

plot isd vs vsd

.endc

.end
```

4.3 Inverter

Listing 3: Ngspice netlist for Inverter analysis

```
.title Inverter
.include models-180nm

vdd vdd 0 dc 1.8
vsin vin vbias sin(0 10m 200k)
vbias vbias 0 dc 0.9

mp vout vin vdd vdd CMOSP W=7u L=0.18u
mn vout vin 0   0   CMOSN W=2u L=0.18u

cl vout 0 250f

.control
run
set color0 = white
set color1 = black
set color2 = red

* -------- Transient Analysis -------- *

tran 10ns 40us

plot v(vout) v(vin)

meas tran vout_max max v(vout)
meas tran vout_min min v(vout)

let vpp_out = vout_max - vout_min
print vpp_out

meas tran vin_max max v(vin)
meas tran vin_min min v(vin)

let vpp_in = vin_max - vin_min
print vpp_in

let gain_transient = vpp_out / vpp_in
print gain_transient

* -------- DC Analysis -------- *

alter vsin dc 0
dc vbias 0 1.8 0.01

plot v(vout) vs v(vbias)

let gain = deriv(v(vout))
plot gain vs v(vbias)

meas dc gain_at_vbias_0v9 find gain when v(vbias) = 0.9

.endc

.end
```