FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

521402S Telecommunications Circuit Design

Simulation exercise #01

Student	Aissa Azzaz
Student Number	2207335

22 April 2023

1 DC SWEEPS

In this part, Vgs and Vds are run, and the results are analyzed.

The Vgs is supplied by the V1 dc source which is swept from -0.1V to 1.1V with swept operating temperature from -50 ^{0}C to 100 ^{0}C with steps of 25^{0}C.

The modified "nmos45nm_iv.asc" schematic file is shown in Figure 1 below.

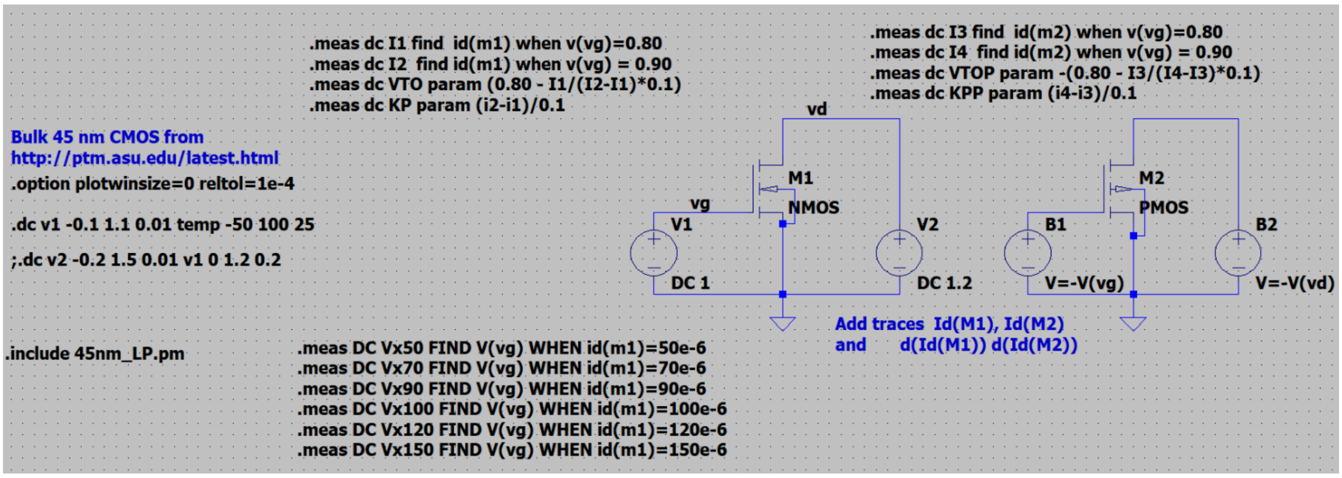

Figure 1. The modified "nmos45nm_iv.asc" schematic file

The resulting Id vs. Vgs and gm vs. Vgs curves for both devices are shown in Figure 2 and Figure 3 below.

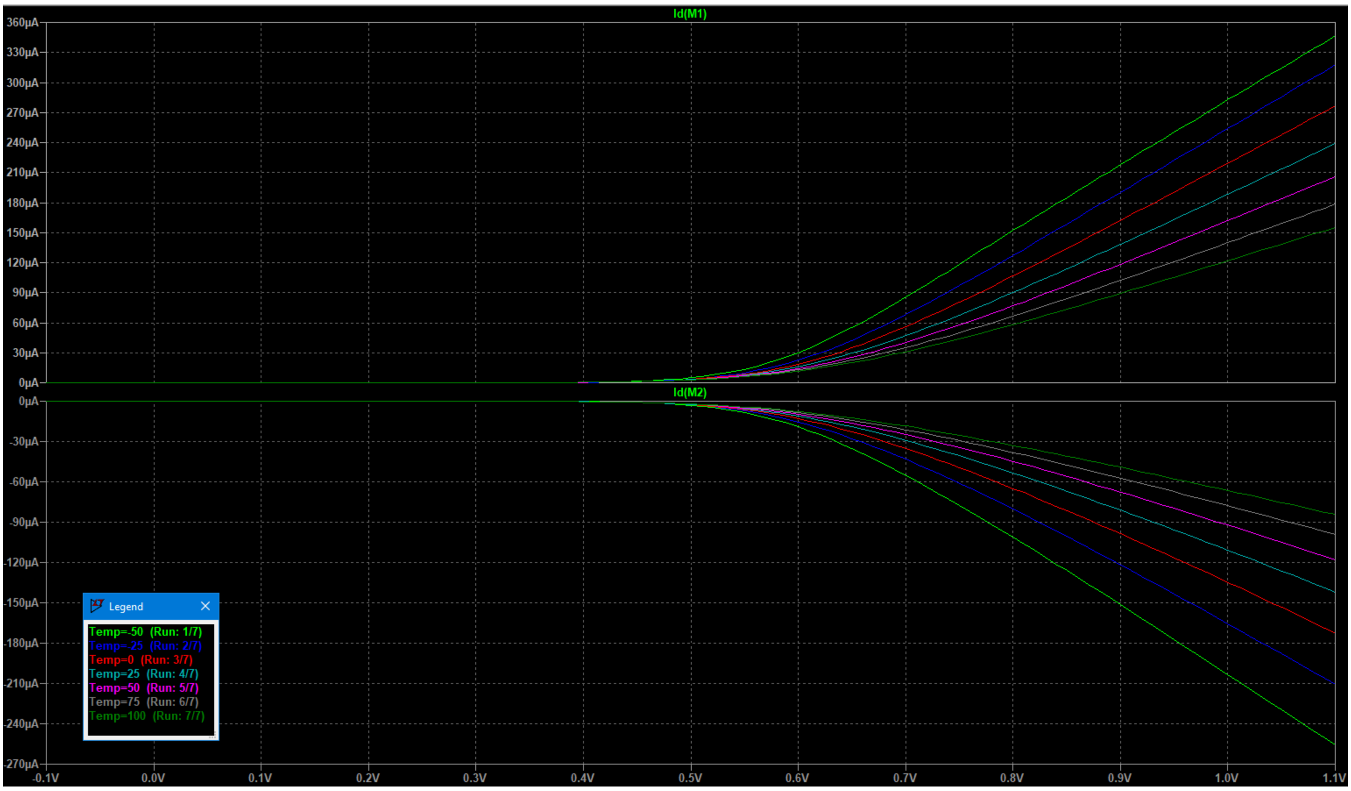

Figure 2. Id vs. Vgs curve for the NMOS and PMOS

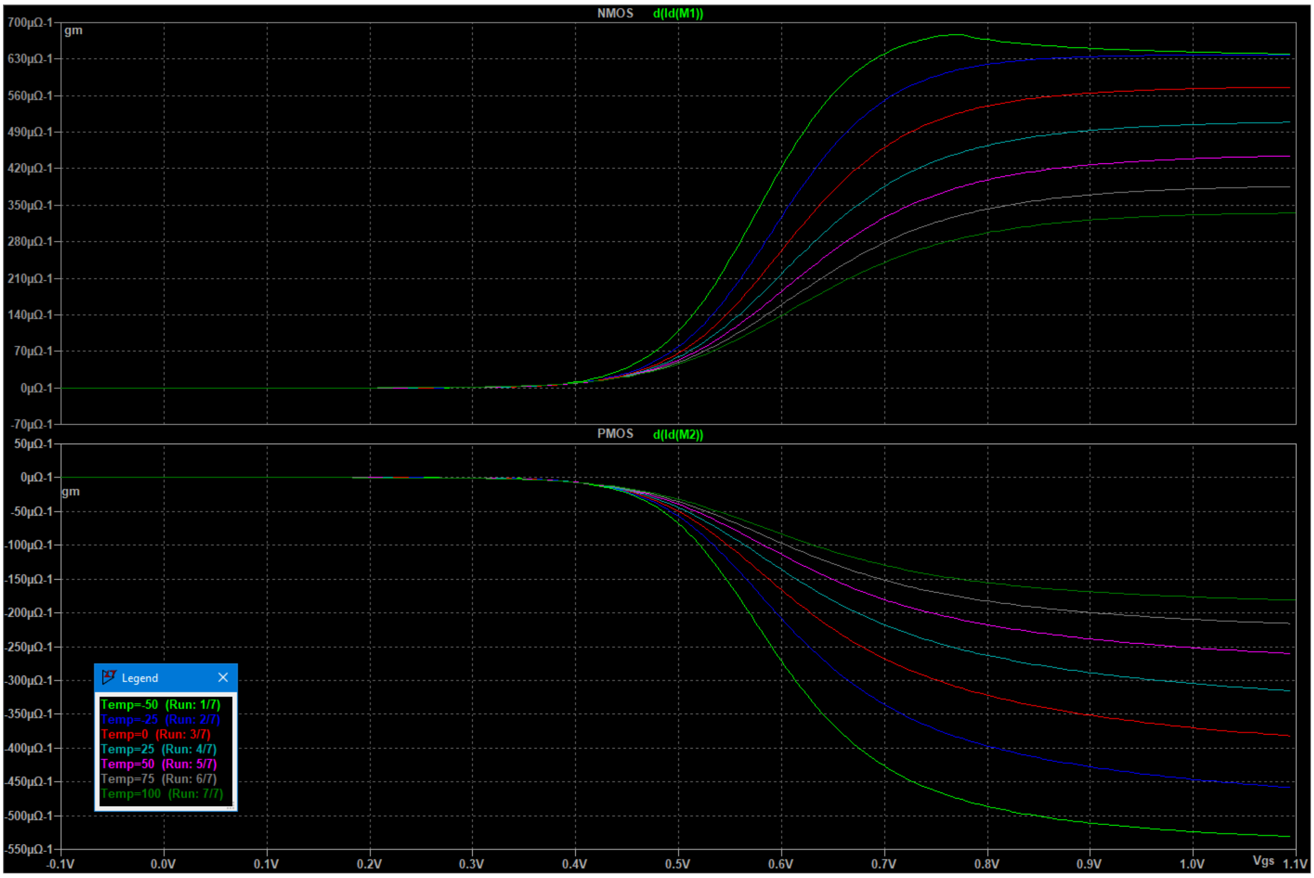

Figure 3. gm vs. Vgs curve for the NMOS and PMOS

The Vds is supplied by the V2 DC source which now swept from -0.2 V to 1.5 V with varying Vgs as seen in the commented spice directive. The resulting Id vs. Vds and go vs. Vds curves for both devices are shown in Figure 4 and Figure 5 below.

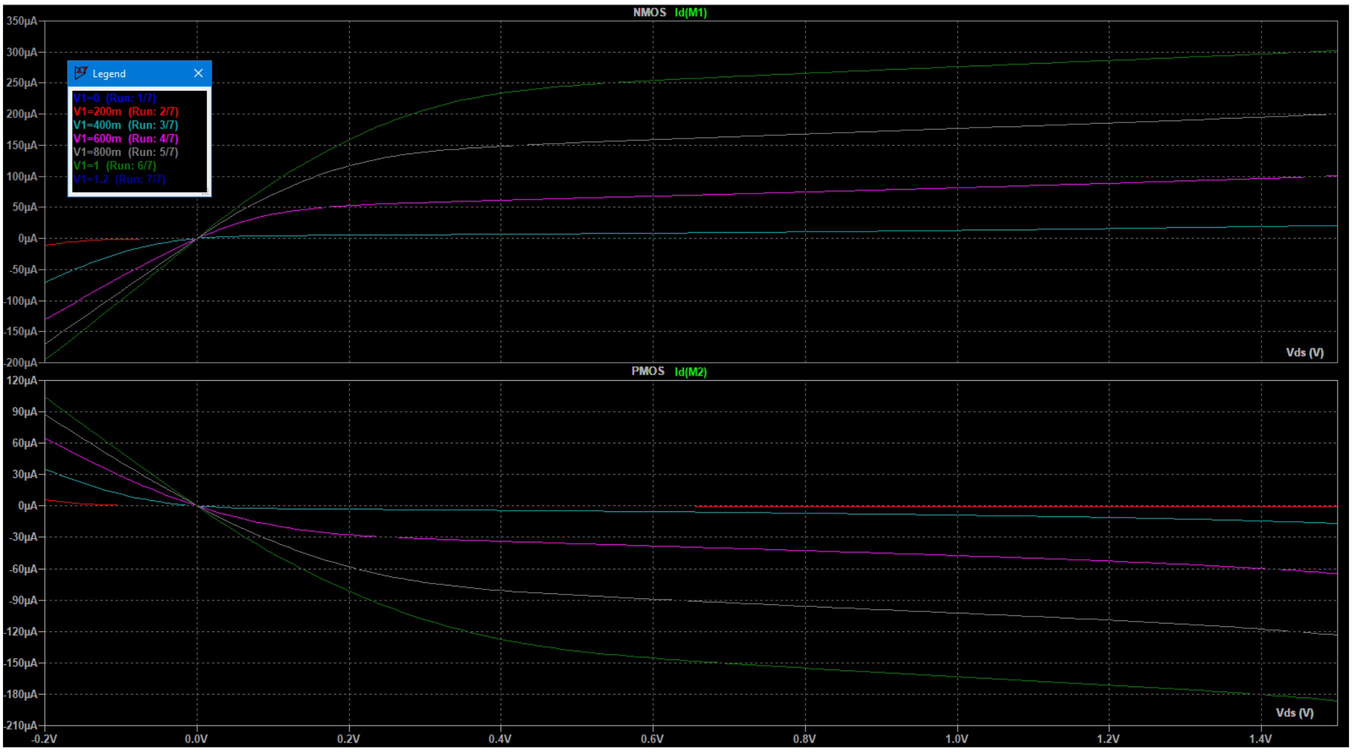

Figure 4. Id vs. Vds curve for the NMOS and PMOS

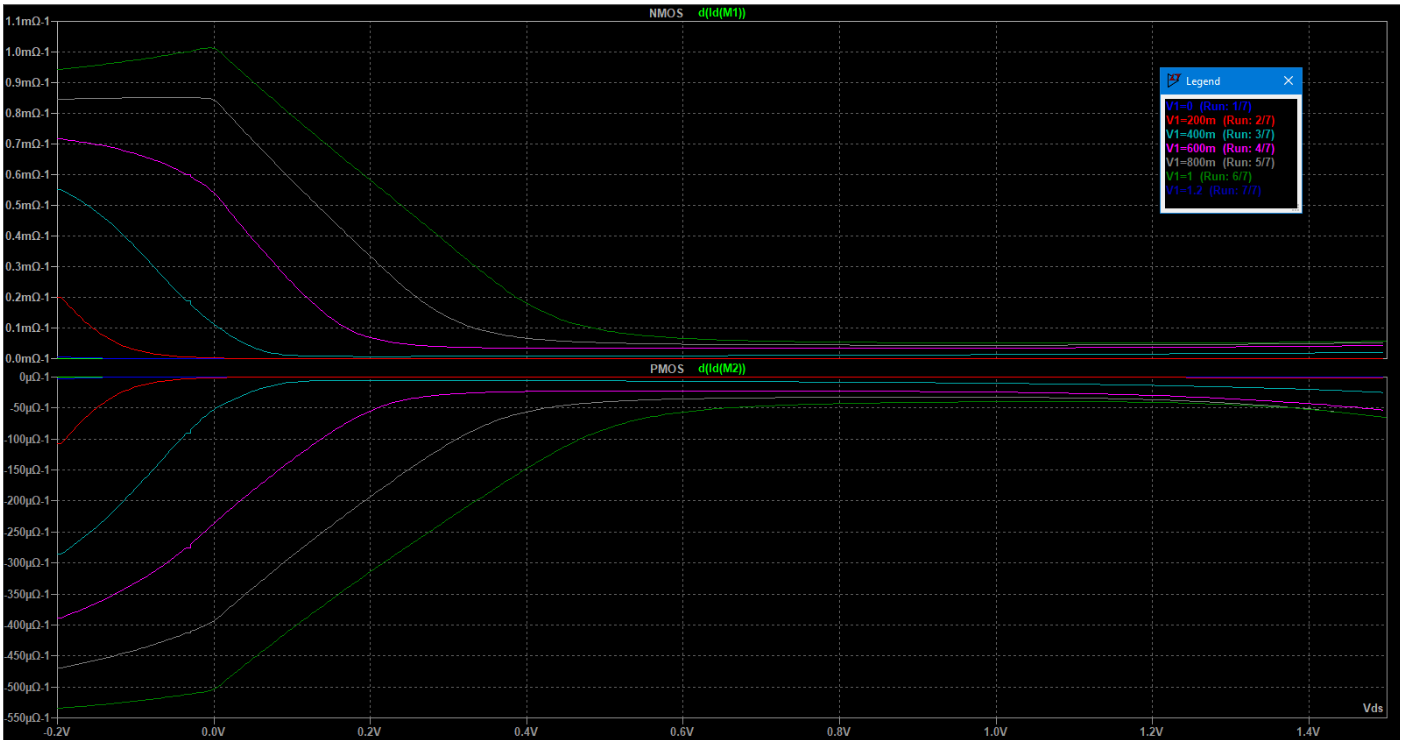

Figure 5. go vs. Vds curve for the NMOS and PMOS

From the previous figures, we notice that a proper bias point would be around Vgs = 0.8 V since the device are well enough in the saturation region and, for the different temperatures the gm is near the maximum achievable gm and increasing the Vgs doesn't result in better gm but rather it reduces the drain voltage swing.

Using provided (on the top left) and added (on the top right) .measure statements, the KP and VTO of both devices could be extracted against the temperature as shown in Figure 6 and 7 .

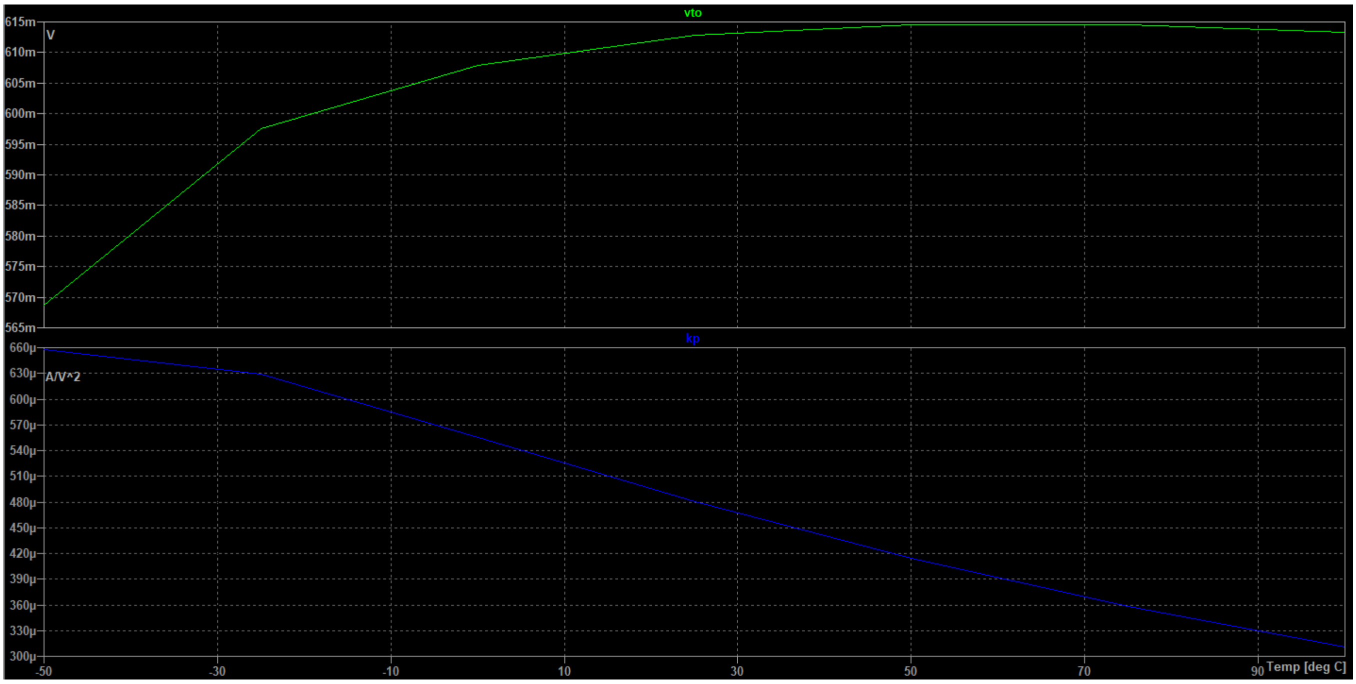

Figure 6. VTO and KP vs. temperature curves for the NMOS

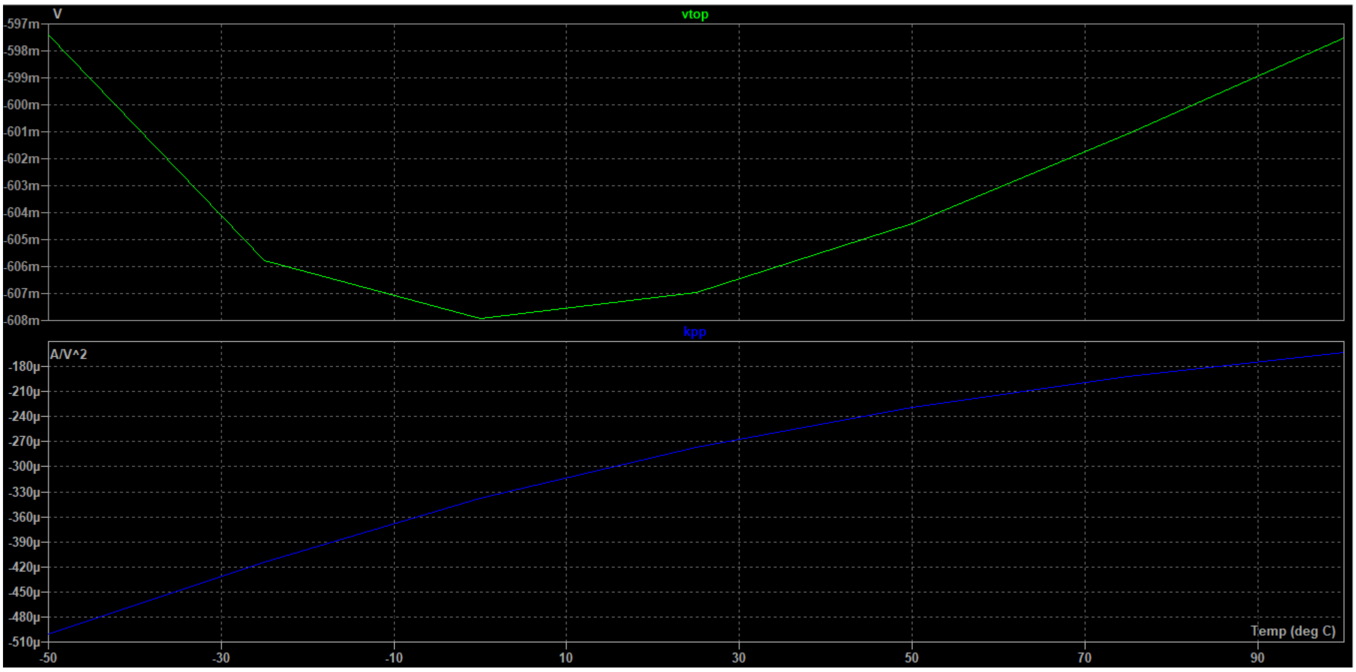

Figure 7. VTO and KP vs. temperature curves for the PMOS

For the NMOS device, the KP and VTO sensitivities for change in temperature are calculated using by derivation VTO and KP. At 30 ^{o}C, KP changes by -2.64 µA/(V^{2} ^{o}C) and VTO by 70 µV/^{o}C.

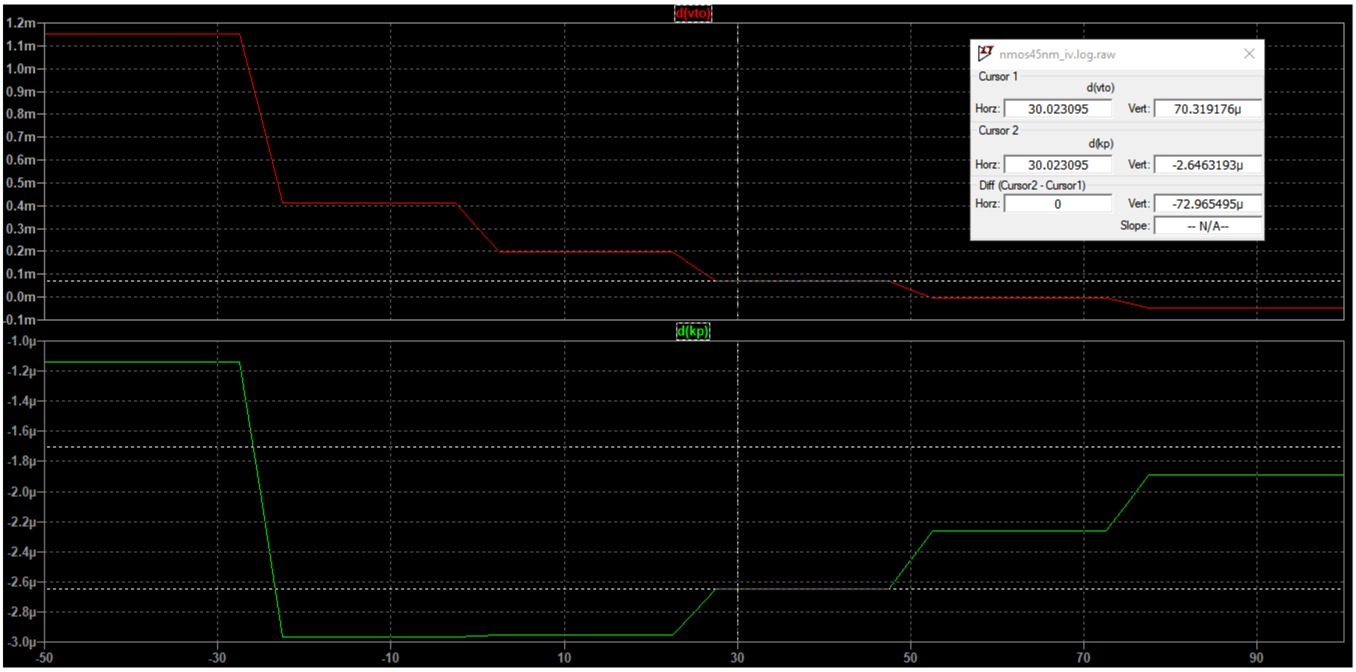

Figure 8. VTO and KP sensitivity to temperature changes for the NMOS

For the NMOS device, Figure 9 shows the required Vgs for constant Ids with varying temperature (check the .measure statements in the bottom). At 30 ^{o}C, and for a constant 100 μA of Ids the bias voltage needs to be corrected by around 1.4 mV/^{o}C.

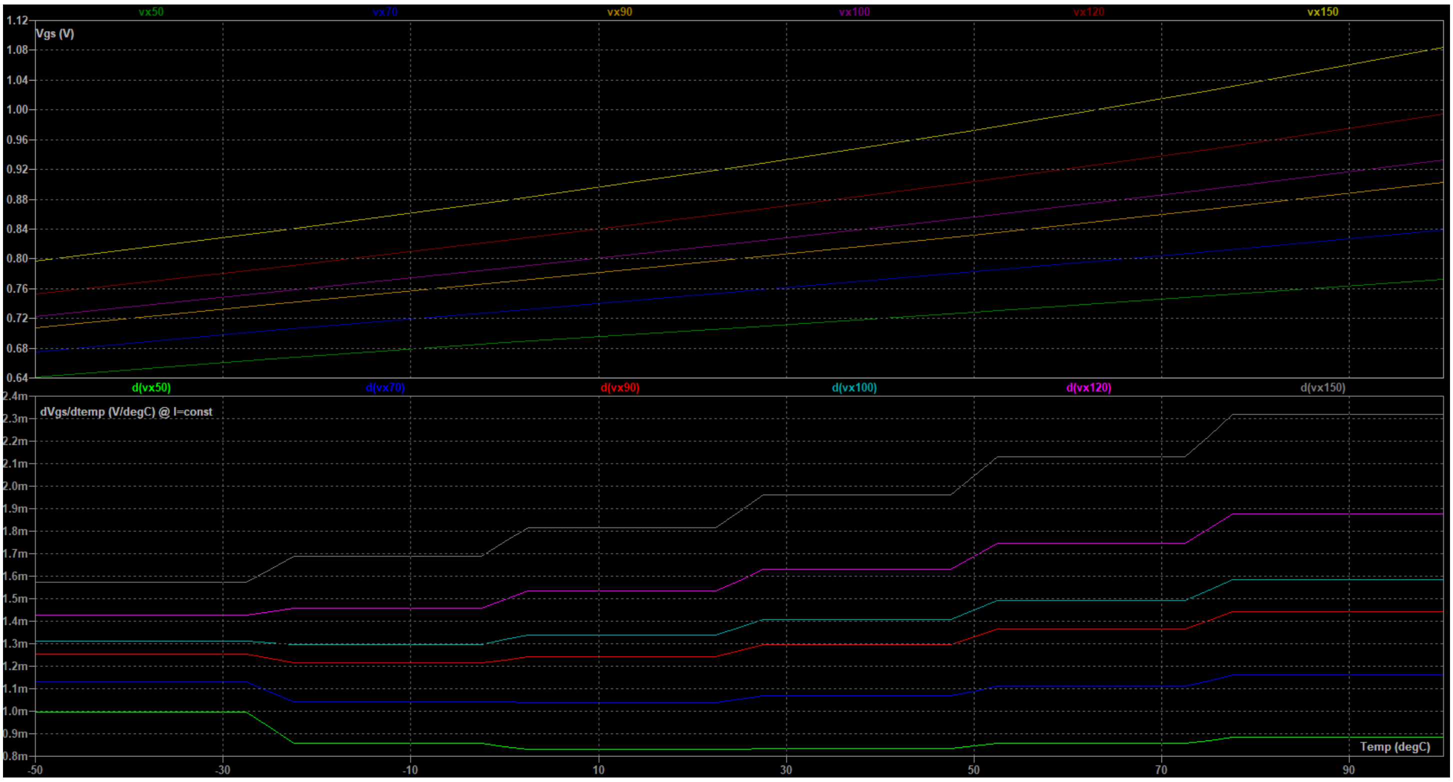

Figure 9. Vgs temperature compensation for the PMOS

2 AC ANLYSIS

In this part, the AC analysis is used and the results are analyzed.

First, Vgs was stepped and the schematic, Figure 10, with the added .measure expressions (the ones on the bottom left) extract the parameters of the NMOS shown in Figure 11.

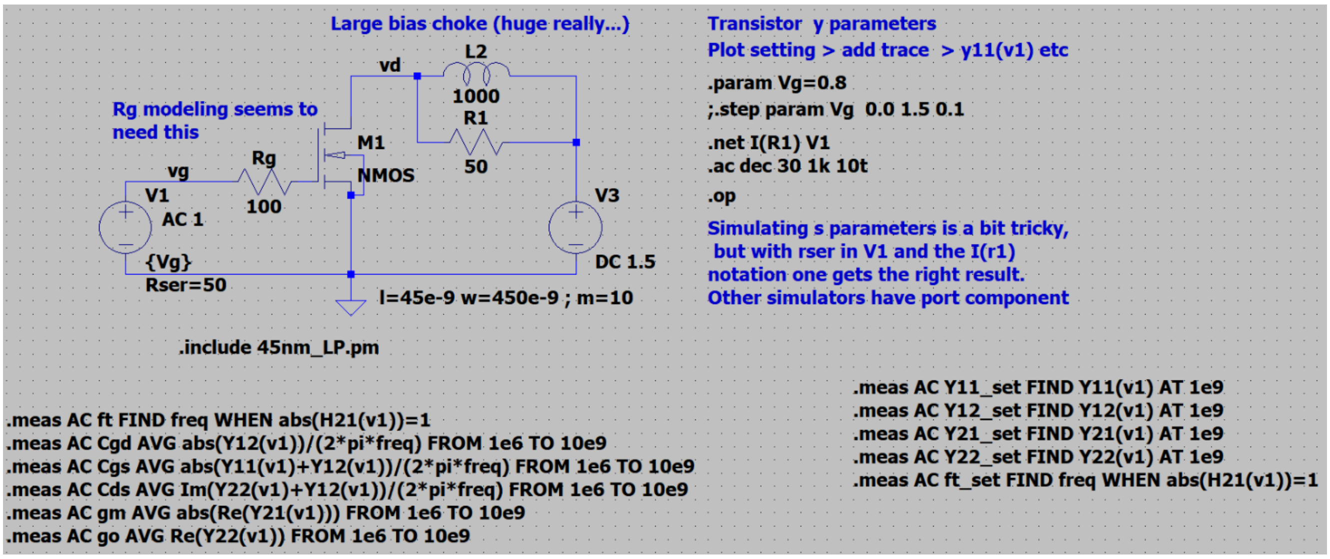

Figure 10. AC analysis schematic

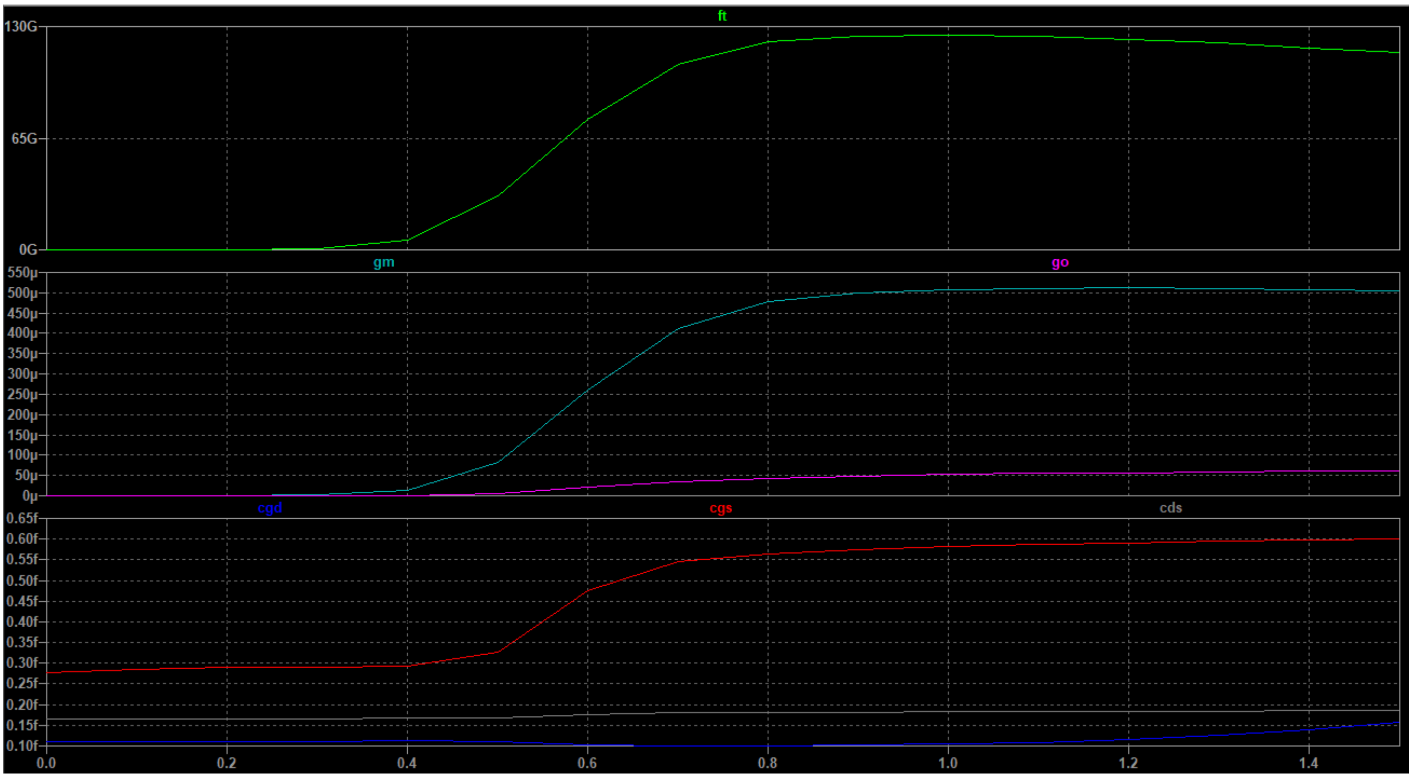

Figure 11. Results of AC analysis with Vgs sweep

For convince of the plots, the bais was fixed to Vgs=0.8 V and the different Y parameters were plotted with the H21 parameter as shown in Figure 12.

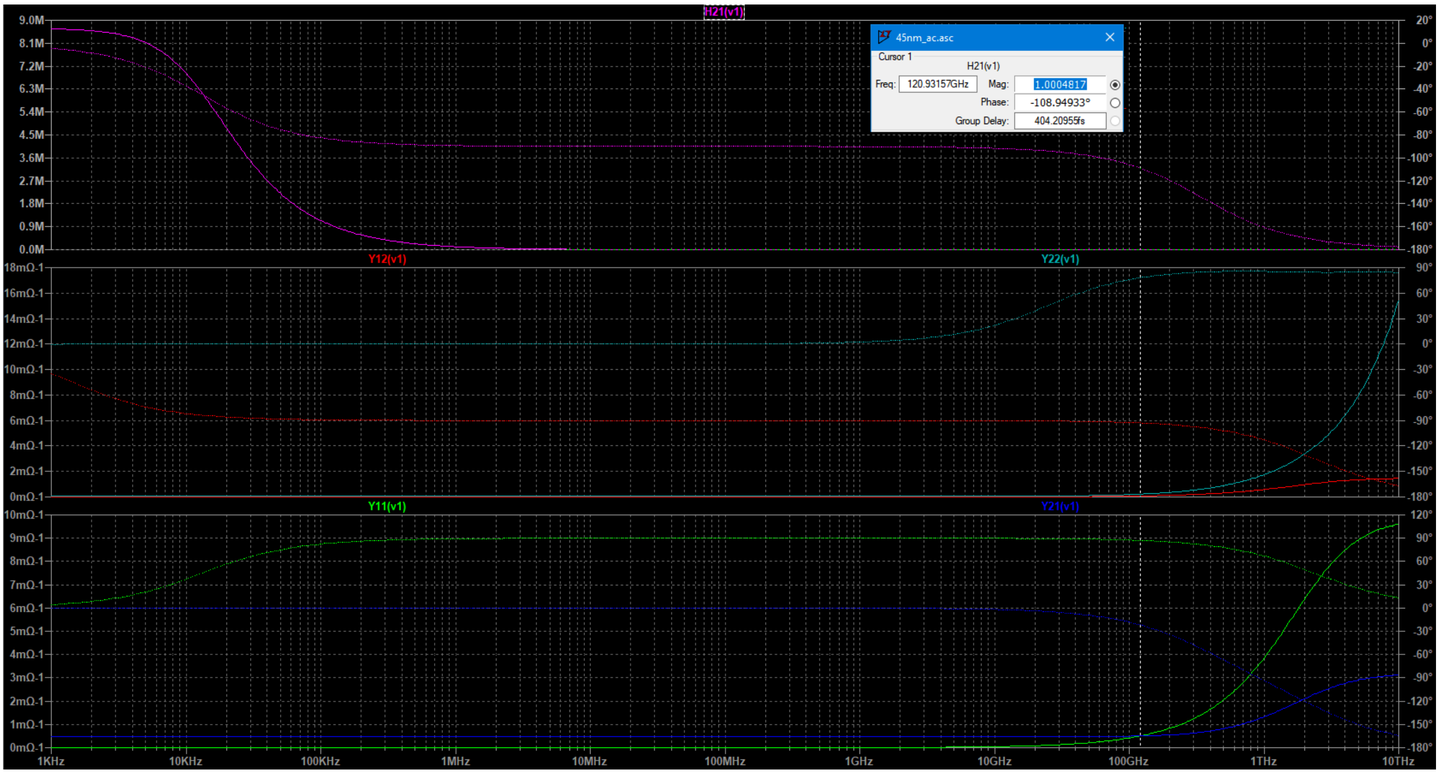

Figure 12. AC analysis resulting Y parameters with Vgs = 0.8 V

For the frequency range [1MHz-10GHz], the different parameters exhibit a linear change with respect to the frequency and their phase is almost constant, therefore this range was used in the .measure expressions (the ones on the bottom right).

The code below shows the extract from LTspice log file and some Matlab code that is used to calculate the different parameters and its results.

Using the NMOS model Y-parameters matrix and its mapping to the actual transistor parameters, the output conductance could be extracted from y22= (go+jw(Cds+Cgd) by taking only the real part of Y22.

The calculated ft, Cgd, go and gm matches the provided process curves.

```
%% SPICE Error log

% y11_set: y11(v1)=(-107.584dB,89.975°) at 1e+009
% y12_set: y12(v1)=(-123.986dB,-90.0241°) at 1e+009
% y21_set: y21(v1)=(-66.4256dB,-0.186952°) at 1e+009
% y22_set: y22(v1)=(-87.2439dB,2.33588°) at 1e+009
% ft_set: freq=(221.655dB,0°) at 1.20997e+011

%% Calculations
dB2cplx = @(dB,phase) (10^(dB/20))*exp(1i*deg2rad(phase)) ;

y11 = dB2cplx(-107.584,89.975) ;
y12 = dB2cplx(-123.986,-90.0241) ;
y21 = dB2cplx(-66.4256,-0.186952) ;
y22 = dB2cplx(-87.2439,2.33588) ;
h21 = dB2cplx(41.1586,-90.1619) ;

w = 2*pi*1e9 ;

Cgd = abs(y12)/w
Cgs = imag(y11+y12)/w
Cds = imag(y22+y12)/w
gm = real(y21)
go = real(y22)

ft_simulation = 1.20997e+011
ft_calculated = gm/(2*pi*(Cgs+Cgd)) %% from theory


%% Results

 % >> calcs
    % Cgd =
    %    100.5820e-018
    % Cgs =
    %    564.1095e-018
    % Cds =
    %    181.1478e-018
    % gm =
    %    477.2190e-006
    % go =
    %     43.3954e-006
    % ft_simulation =
    %    120.9970e+009
    % ft_calculated =
    %    114.2662e+009
```