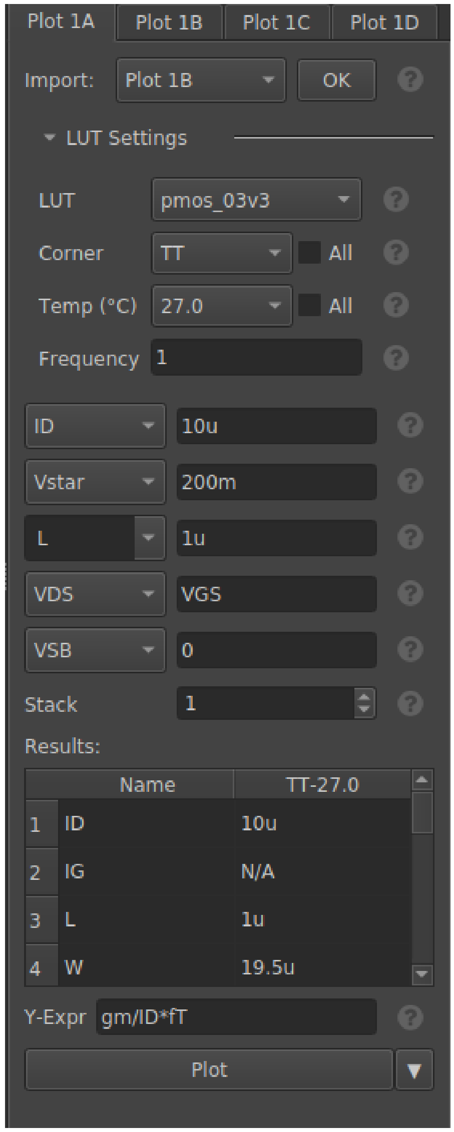
# ITI

# LAB4

# Common Drain Frequency Response

## Contents

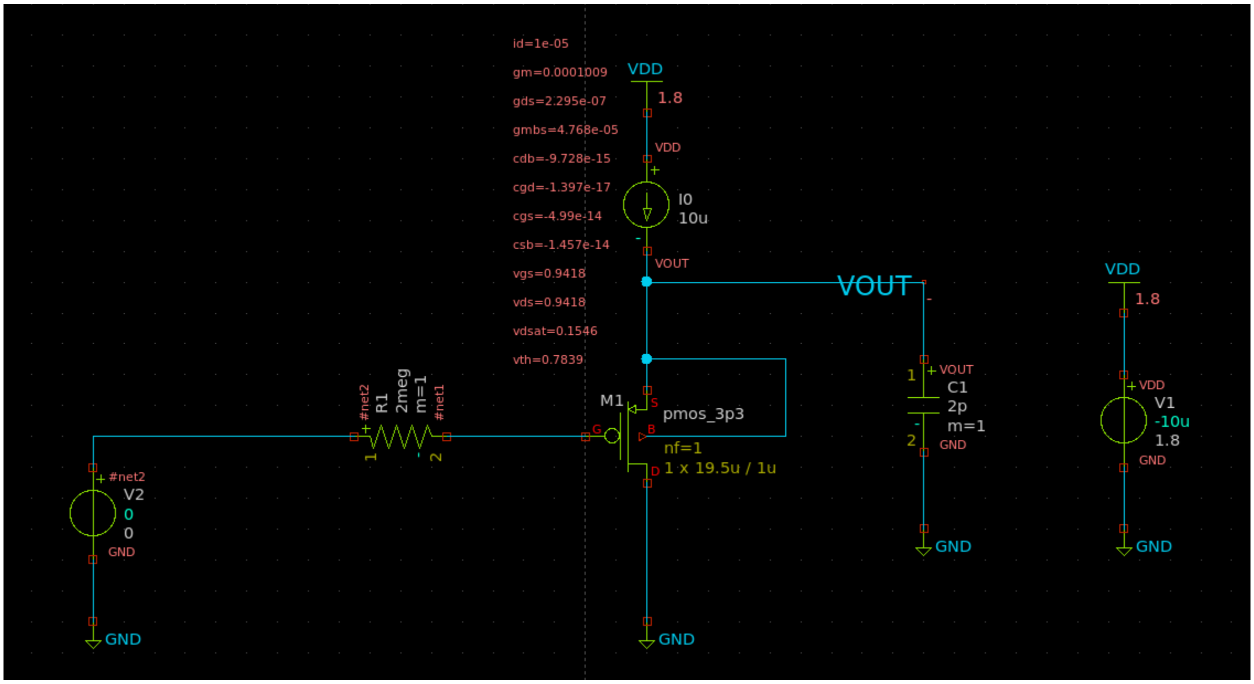
# Part 1: Device Sizing Using SA




W=19.5 $\mu m$

# Part 2: CD Amplifier

## OP (Operating Point) Analysis



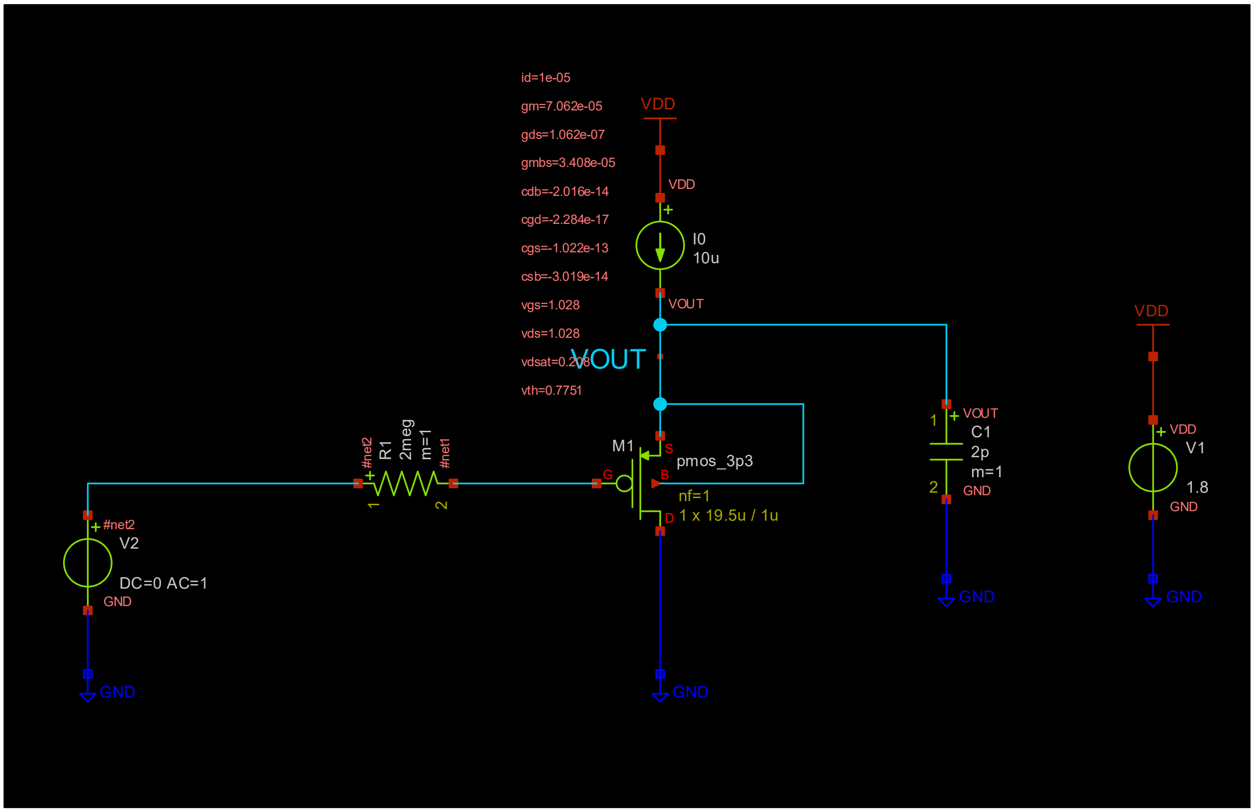

Since that $V_{dsat} = 0.1546V$ , $V_{ds} = 0.9418V$

$V_{ds} > \ V_{dsat}$

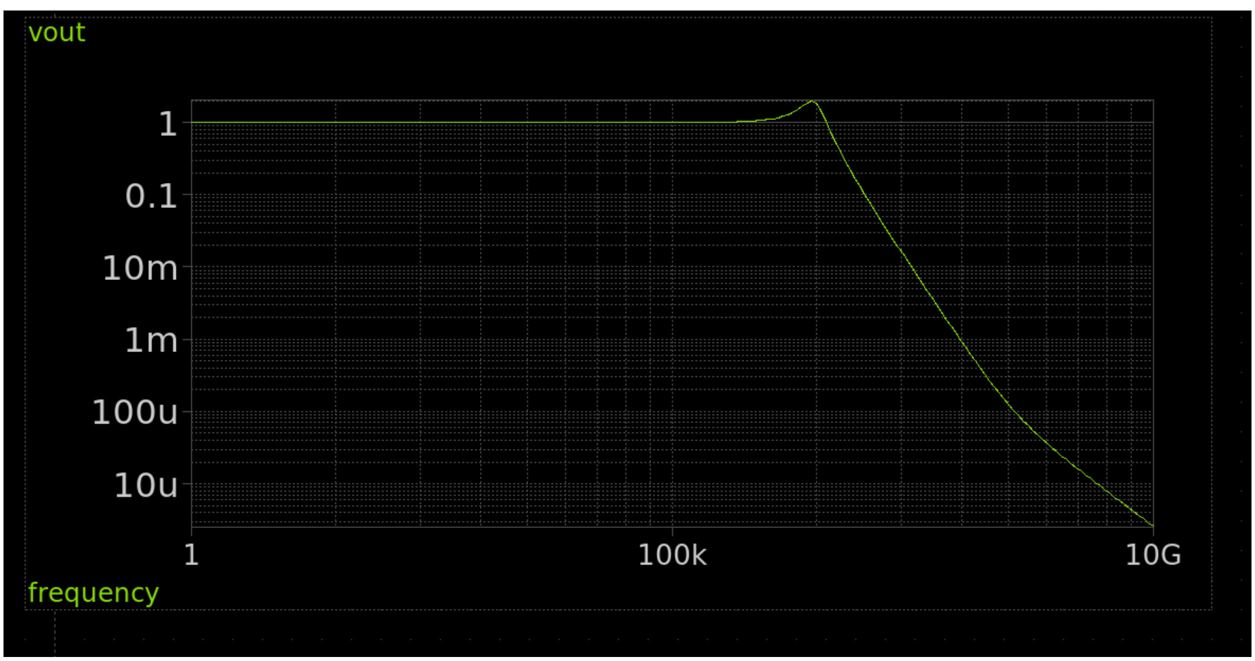
Meaning that the transistor is in the saturation region

# AC Analysis

## Schematic




## Bode plot (Magnitude)



Peaking

```
No. of Data Rows : 201
max_vout_db          =  6.018124e+00 at=  2.818383e+06
max_vout_db = 6.018124e+00
binary raw file "lab_04_CD_ac.raw"
ngspice 1 ->
```

Quality factor

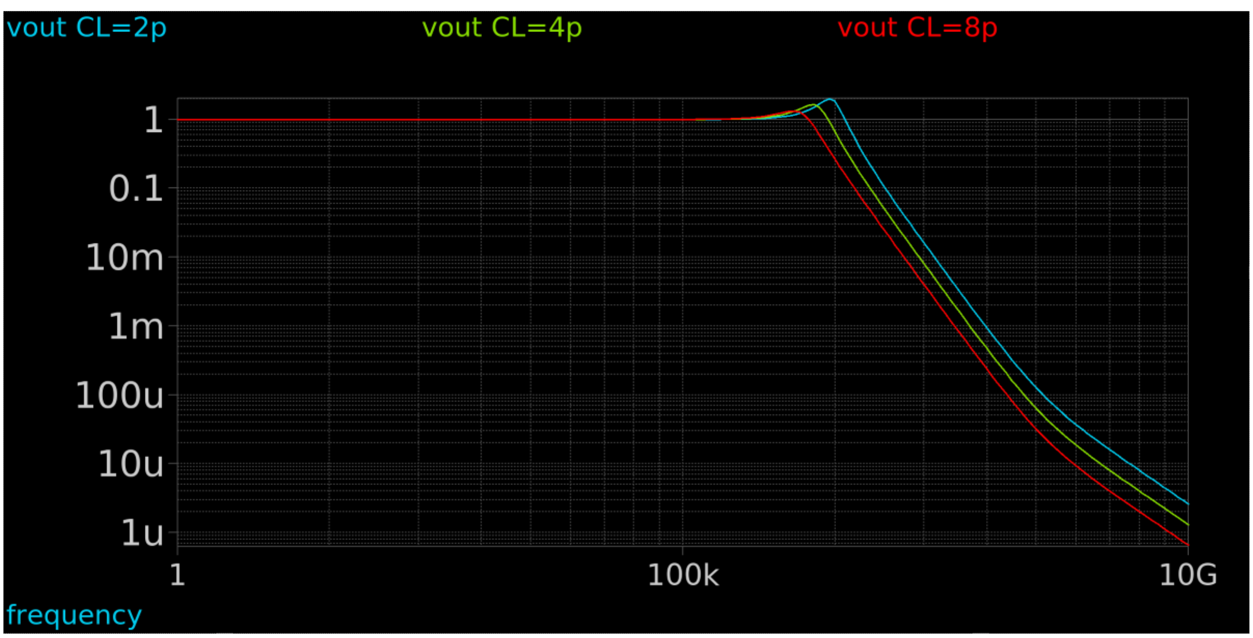
$$Q = \sqrt{\frac{g_m(C_{gs} + C_{gd})R_{Sig}}{C_L}}$$

$$Q = \sqrt{\frac{7.062 \times 10^{-5}(1.022 \times 10^{-13} + 2.284 \times 10^{-17})2 \times 10^{6}}{2 \times 10^{-12}}}$$

$Q = 2.686$

parametric sweep: CL = 2p, 4p, 8p.

*Bode plot magnitude overlaid on same plot*



*The peaking vs $C_L$*

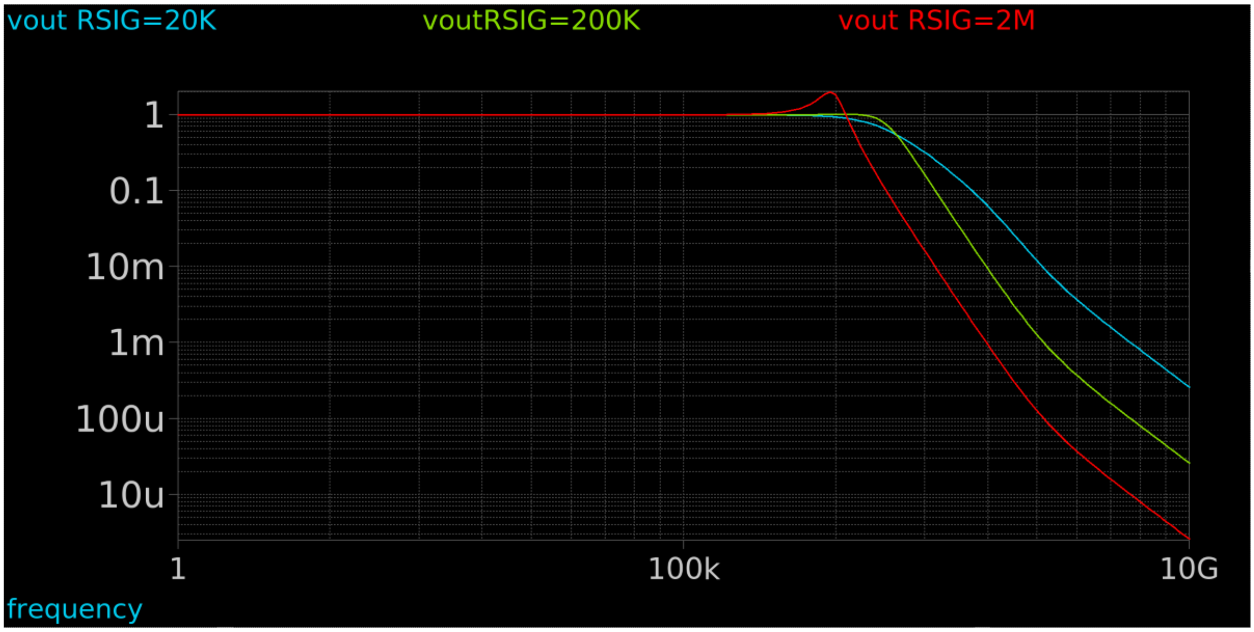
```
No. of Data Rows : 201
peak_db             =  6.018124e+00 at=  2.818383e+06
at C=2E-12 the value of peak in dB is 6.01812
binary raw file "lab_04_CD_ac_CLsweep.raw"
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000


No. of Data Rows : 201
peak_db             =  4.358113e+00 at=  1.995262e+06
at C=4E-12 the value of peak in dB is 4.35811
binary raw file "lab_04_CD_ac_CLsweep.raw"
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000


No. of Data Rows : 201
peak_db             =  2.414030e+00 at=  1.258925e+06
at C=8E-12 the value of peak in dB is 2.41403
```

Comment:

As the load capacitance $C_L$ increases, the peaking in the frequency response decreases. This occurs because a larger $C_L$ lowers the overall bandwidth and increases the dominant pole's effect, effectively reducing the gain at higher frequencies. Additionally, the larger capacitance dampens the resonant behavior of the circuit, leading to a flatter frequency response and less pronounced peaking near the cutoff frequency.

parametric sweep: $R_{Sig}$ = 20k, 200k, 2M

*Bode plot magnitude overlaid on same plot.*



*The peaking vs $R_{Sig}$*

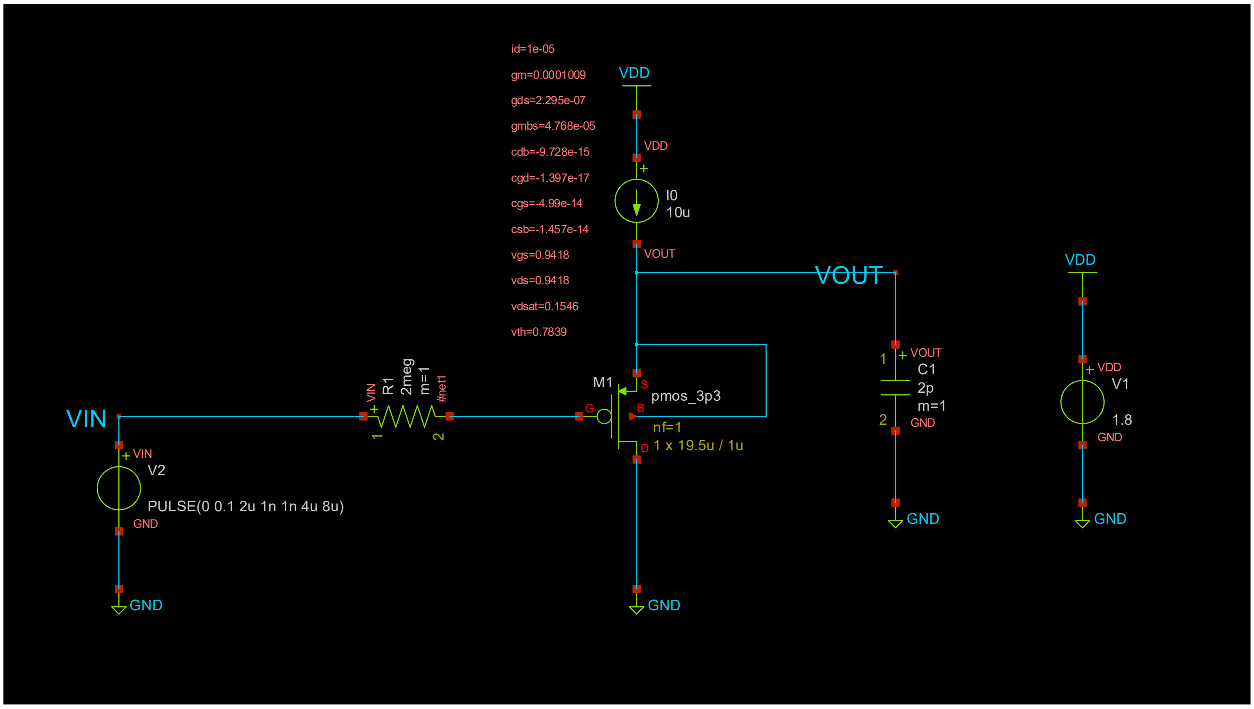
```
No. of Data Rows : 201
peak_db              =  -1.973027e-02 at=  1.000000e+00
at R=20000 the value of peak in dB is -0.0197303
binary raw file "lab_04_CD_ac_RLsweep.raw"
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000


No. of Data Rows : 201
peak_db              =  1.326869e-01 at=  3.981072e+06
at R=200000 the value of peak in dB is 0.132687
binary raw file "lab_04_CD_ac_RLsweep.raw"
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000


No. of Data Rows : 201
peak_db              =  6.018124e+00 at=  2.818383e+06
at R=2E+06 the value of peak in dB is 6.01812
binary raw file "lab_04_CD_ac_RLsweep.raw"
ngspice 1 ->
```

Comment:

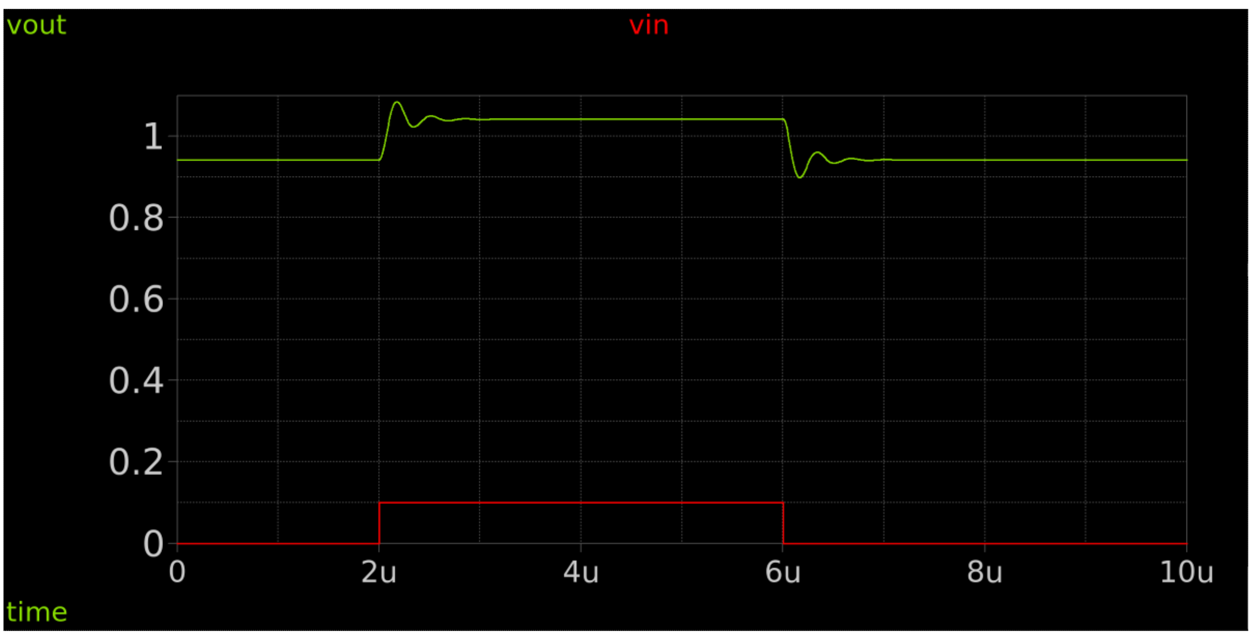
As $R_{sig}$ increases, the peaking in the frequency response becomes more pronounced. This is because a higher resistance reduces the damping in the system, increasing the quality factor Q. A higher Q corresponds to a sharper and more pronounced peak near the resonant frequency. In other words, less damping allows more energy to build up at that frequency, resulting in greater peaking

# Transient Analysis

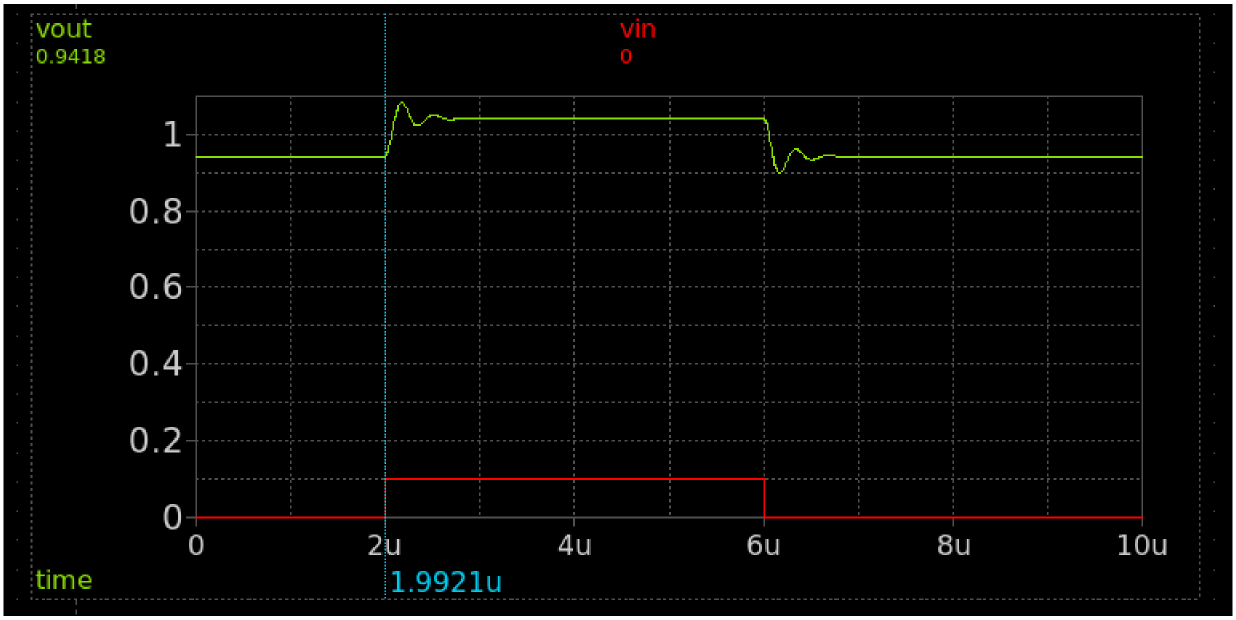
## Schematic




## Vin and Vout overlaid vs time.



The DC voltage difference (DC shift) between $V_{\text{in}}$and $V_{\text{out}}$




From the graph the DC shift =0.9418V and from the DC OP the $V_{gs} = 0.9418V$

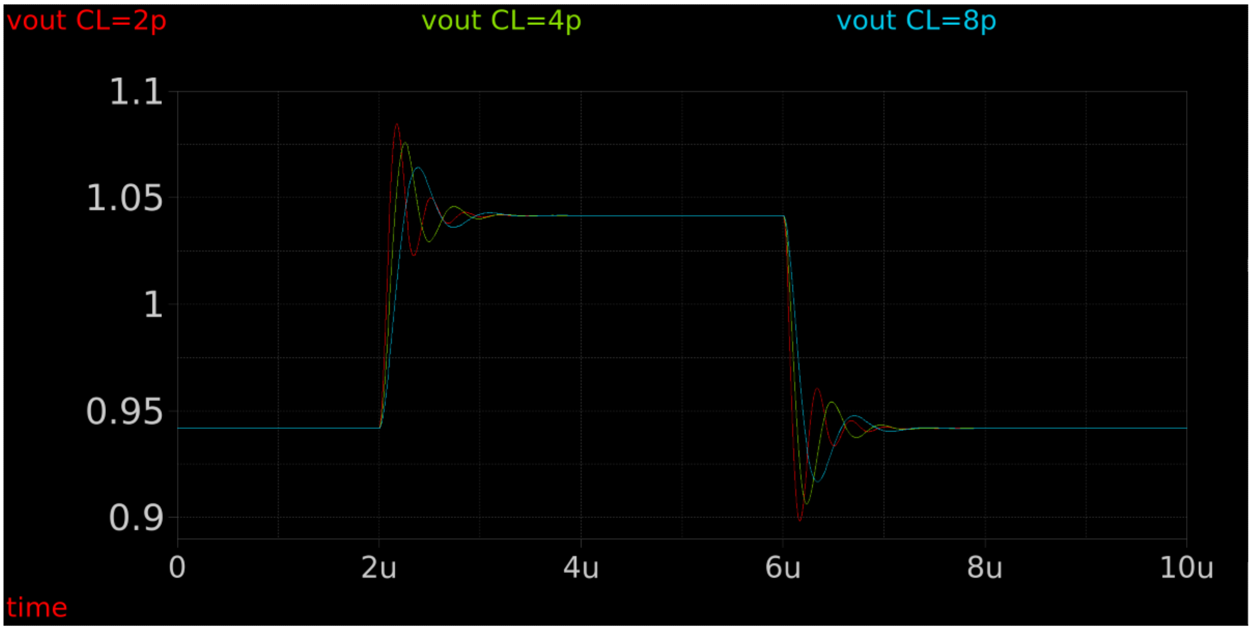
Which means that the DC shift between $V_{\text{in}}$and $V_{\text{out}}$ equals exactly $V_{\text{GS}}$

Comment:

The transient response of $V_{\text{out}}$ shows clear ringing behavior, characterized by oscillations following a step change before the signal settles. This indicates that the system is underdamped, likely due to a low damping ratio. The ringing is a result of energy exchanging between reactive elements and its presence confirms that the circuit has a relatively high quality factor.

Parametric sweep: CL = 2p, 4p, 8p.

$V_{out}$ vs time overlaid on same plot.



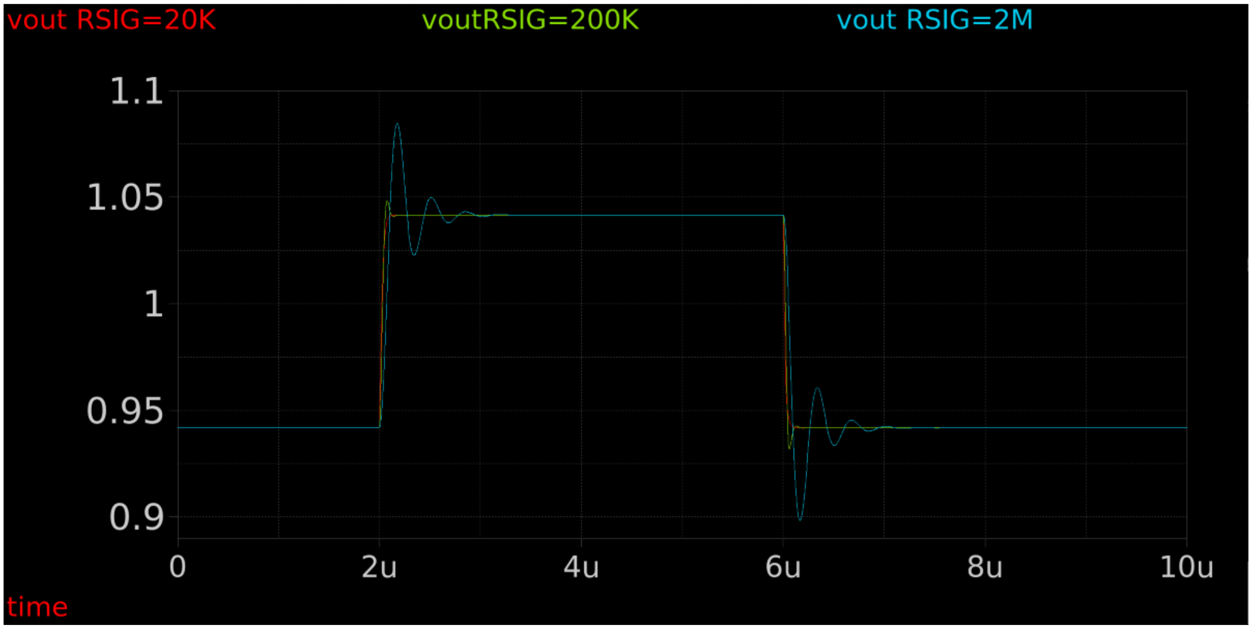

Overshot

```
1 const.cl_val = 2.000000e-12
2 tran1.os = 4.310105e+01
3 const.cl_val = 4.000000e-12
4 tran2.os = 3.424372e+01
5 const.cl_val = 8.000000e-12
6 tran3.os = 2.259230e+01
```

Comment: As the load capacitance $C_L$ increases, the overshoot in the time-domain response decreases. This is because a larger $C_L$ increases the effective damping in the system. The capacitor acts as an energy storage element that slows down the rate of voltage change, thereby reducing the tendency of the circuit to overshoot its final value. Additionally, increasing $C_L$ lowers the system's natural frequency and reduces the quality factor Q, both of which contribute to a more damped response with reduced overshoot.

Parametric sweep: Rsig = 20k, 200k, 2M.

$V_{out}$ vs time overlaid on same plot.

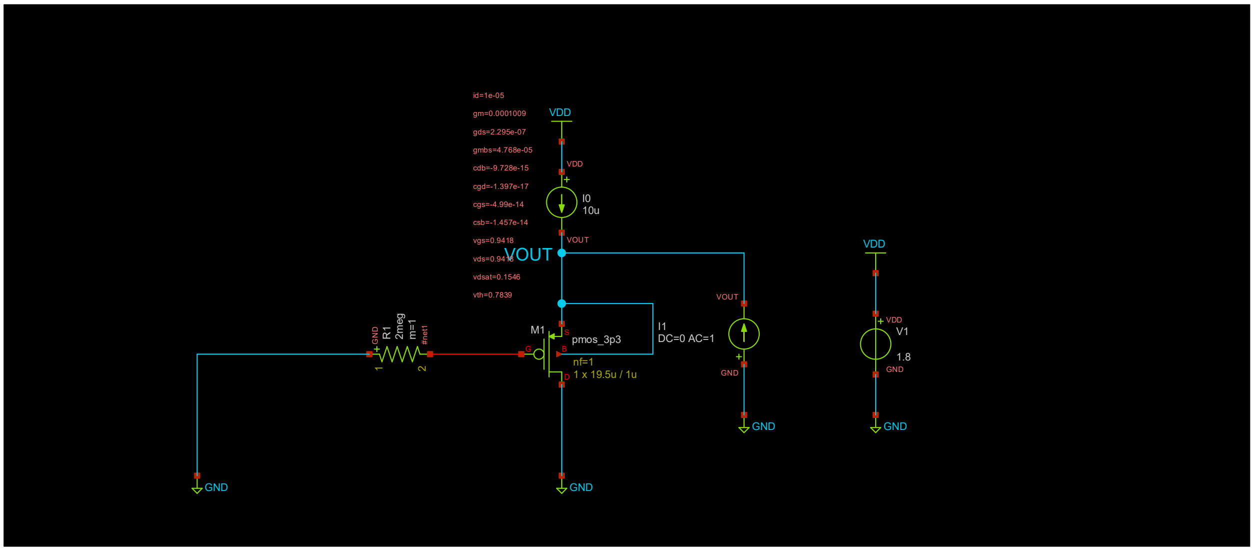



Overshot

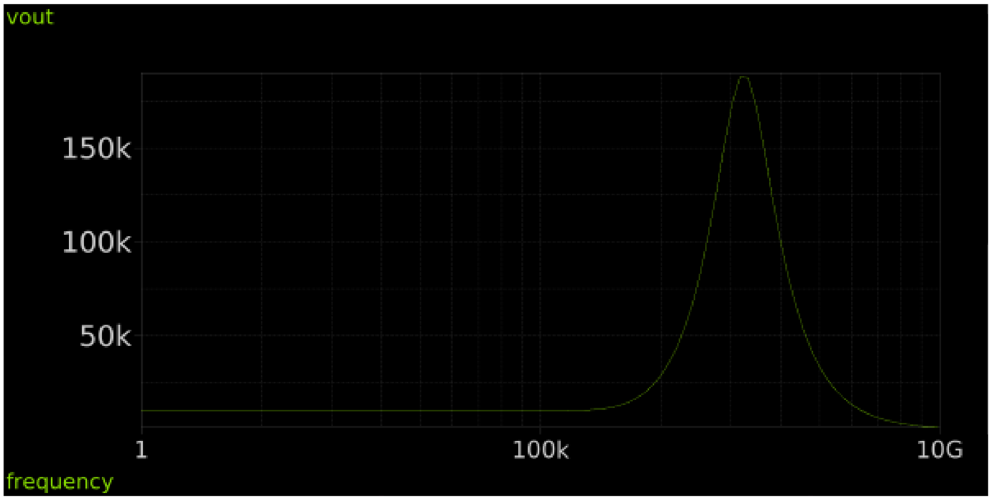
```
const.r_val = 2.000000e+04
tran1.os = 0.000000e+00
const.r_val = 2.000000e+05
tran2.os = 6.725675e+00
const.r_val = 2.000000e+06
tran3.os = 4.310105e+01
```

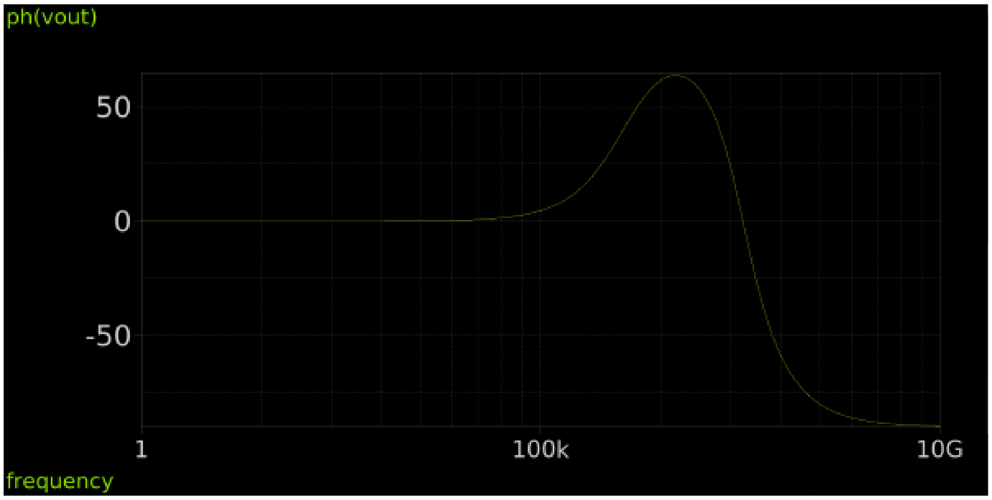
Comment: As $R_{sig}$ increases, the overshoot decreases. This is because higher resistance introduces more damping into the system, which helps dissipate energy more effectively. As a result, the circuit becomes less underdamped, and the output settles more smoothly with reduced oscillations and overshoot. This behavior is consistent with the increase in damping ratio and decrease in quality factor Q.

## $Z_{out}$ (Inductive Rise)




The output impedance (magnitude and phase) vs frequency.



- Yes, $Z_{out}$ tends to fall at high frequency.
  This is because the $C_{gd}$ capacitor introduces a frequency-dependent path that effectively lowers the output impedance as frequency increases.

## Zeros and poles

```
ngspice 1 -> print all
pole(1) = -3.99832e+14,0.000000e+00
pole(2) = -1.74115e+08,1.369887e+08
pole(3) = -1.74115e+08,-1.36989e+08
zero(1) = -9.68700e+14,0.000000e+00
zero(2) = -4.21570e+14,0.000000e+00
zero(3) = -7.35400e+06,0.000000e+00
ngspice 2 ->
```

$$H(S) = \frac{\left(1 + SR_{sig}\left(C_{gs} + C_{gd}\right)\right) \times r_0}{1 + SR_{sig}\left(C_{gs} + C_{gd}\right) + r_0 \times \left(1 + SR_{sig} \times C_{gd}\right)\left(SC_{gs} + g_m\right)}$$

$$Z_{out} = \frac{\left(1 + SR_{sig}\left(C_{gs} + C_{gd}\right)\right) \times r_o}{S^2\left[R_{sig}\left(C_{gs} + C_{gd}\right)C_{db}r_o + R_{sig}C_{gs}C_{gd}r_o\right] + S\left[C_{db}r_o + R_{sig}\left(C_{gs} + C_{gd}\right) + C_{gs}r_o + g_m r_o C_{gd}R_{sig}\right] + g_m r_o + 1}$$

Using matlab to calculate the poles and zeros of $Z_{out}$

```
Code: %%%%%%%%%%%%%%%%%%%%%%%%%%%%

function analyze_Zout()
    R_sig = 2e6;          % Signal resistance in ohms
    C_gs  = 52.5e-15;     % Gate-source capacitance in farads
    C_gd  = 3.073e-15;    % Gate-drain capacitance in farads
    C_db  = 18.57e-15;    % Drain-to-bulk capacitance in farads
    gm    = 99.15e-6;     % Transconductance in siemens
    r0    = 4.357e6;      % Output resistance in ohms (assumed value for
completeness)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Symbolic Transfer Function Z_out(s)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    syms s
```

```
Csum = C_gs + C_gd;
Numerator = (1 + s*R_sig*Csum) * r0;

% Coefficient of s^2
A2 = R_sig * Csum * C_db * r0 + R_sig * C_gs * C_gd * r0;

% Coefficient of s
A1 = C_db * r0 + R_sig * Csum + C_gs * r0 + gm * r0 * C_gd * R_sig;

% Constant term
A0 = gm * r0 + 1;

Denominator = A2*s^2 + A1*s + A0;

Zs = simplify(Numerator / Denominator);

fprintf('\n=============================\n');
fprintf(' Symbolic Z_out(s)\n');
fprintf('=============================\n');
pretty(Zs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convert to Numeric Polynomials
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Num_exp = expand(Numerator);
Den_exp = expand(Denominator);

num_coeffs = sym2poly(Num_exp);
den_coeffs = sym2poly(Den_exp);

% Display Coefficients
fprintf('\nNumerator Coefficients:\n');
disp(num_coeffs);
fprintf('Denominator Coefficients:\n');
disp(den_coeffs);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute Zeros and Poles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
zeros_Z = roots(num_coeffs);
poles_Z = roots(den_coeffs);

% Display Results
fprintf('\n=============================\n');
fprintf(' Zeros of Z_out(s) in rad/s\n');
fprintf('=============================\n');
```

```
    for i = 1:length(zeros_Z)
        fprintf('Zero %d: %.6e + j%.6e\n', i, real(zeros_Z(i)),
imag(zeros_Z(i)));
    end

    fprintf('\n============================\n');
    fprintf(' Poles of Z_out(s) in rad/s\n');
    fprintf('============================\n');
    for i = 1:length(poles_Z)
        fprintf('Pole %d: %.6e + j%.6e\n', i, real(poles_Z(i)),
imag(poles_Z(i)));
    end

end
```

```
==============================
 Zeros of Z_out(s) in rad/s
==============================
Zero 1: -8.997175e+06 + j0.000000e+00

==============================
 Poles of Z_out(s) in rad/s
==============================
Pole 1: -1.478970e+08 + j1.405927e+08
Pole 2: -1.478970e+08 + j-1.405927e+08
```