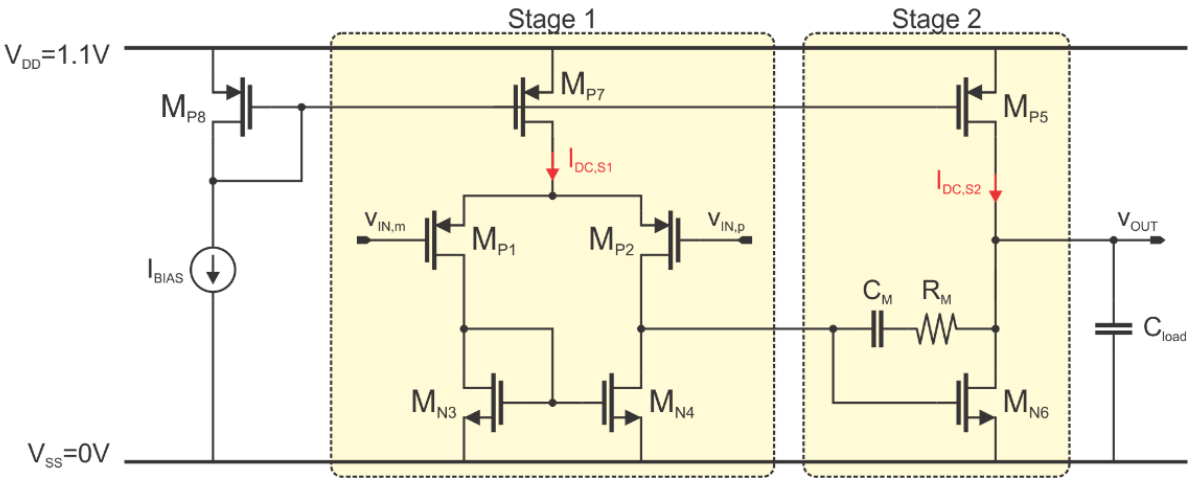
Analog Electronic Circuits – 2021-2022
Design Project Report

Group data

Group number	
Name – Student 1	Semilogo Ogungbure
Name – Student 2	

Goal:



Design of the 2stage OpAmp such that it passes the given specifications (insert your specs below):

$C_{load}[pF]$	3
DC gain [dB]	48
f_{GBW} [MHz]	50
Phase margin (PM) [deg]	> 70
Output swing [V]	> 0.7

Plan: Design of the 2-stage OpAmp on paper (10 points)

1) Calculate the V_{OV}, g_m, I_{DS} and L of each transistor (see exceptions under "*Remark*") and the required C_M and R_M in the OpAmp circuit. For that, insert all your calculations as well as your $\frac{g_m}{g_{ds}}$-, $\frac{g_m}{I_D}$-plots you used for your handcalculations below:

Hints:

- *First, plot $\frac{g_m}{g_{ds}}$ and $\frac{g_m}{I_D}$ across V_{OV} and gatelength L and do it again across V_{GS}, as presented in the 1st session. Think about whether to create the plots for a PMOS or for a NMOS device.*
- *Then, based on those plots, start your calculations.*
- *Furthermore, you can assume the following:*
  (1) the OpAmp is designed in triple-well-technology (i.e. $V_{SB} = 0$); (2) $V_{dsat} \approx V_{OV}$ and $V_{dsat} \geq 50mV$ across V_{OV}; (3) $C_M = \frac{1}{4} \cdot C_L$

Remarks:

- *For Mp5, Mp7 and Mp8 you are not required to calculate the g_m*

Plots used for the handcalculations:

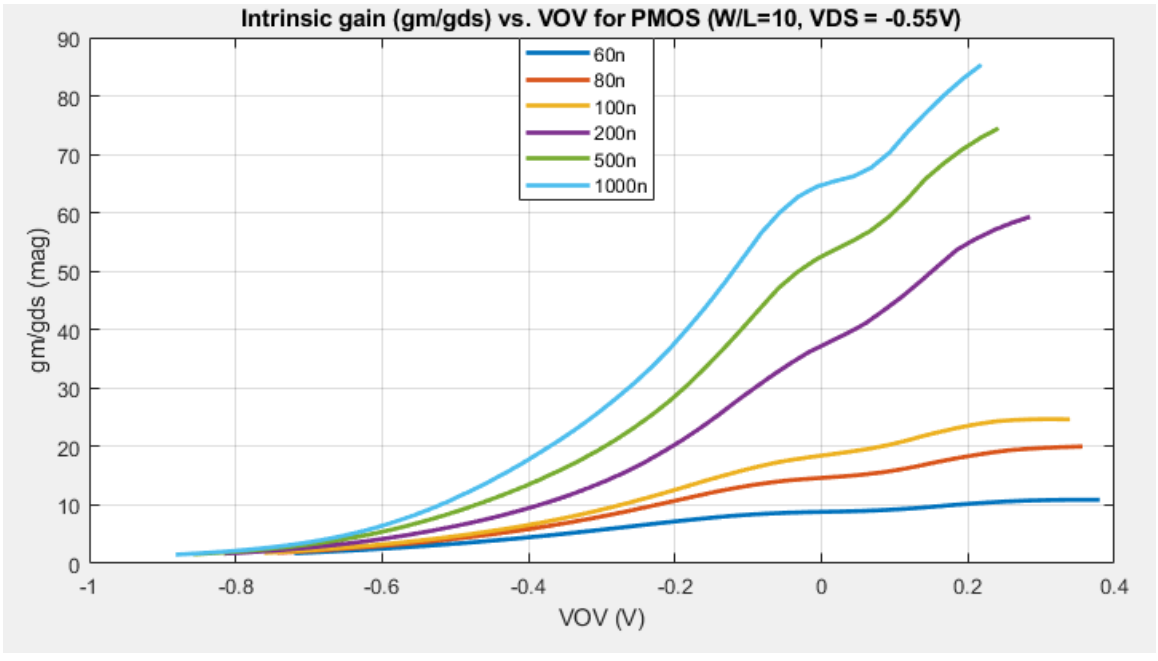


Fig.1

Figure 1: $\frac{g_m}{g_{ds}}$ vs V_{OV} across gatelength L

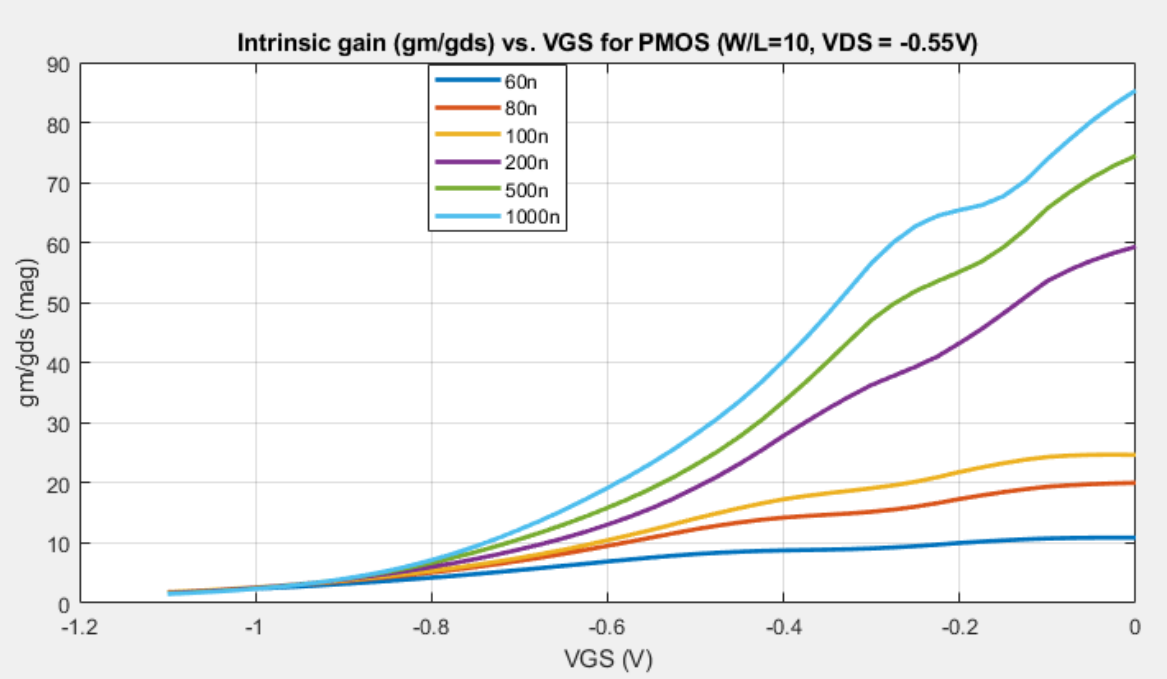


Fig.2

Figure 2: $\frac{g_m}{g_{ds}}$ vs V_{GS} across gatelength L

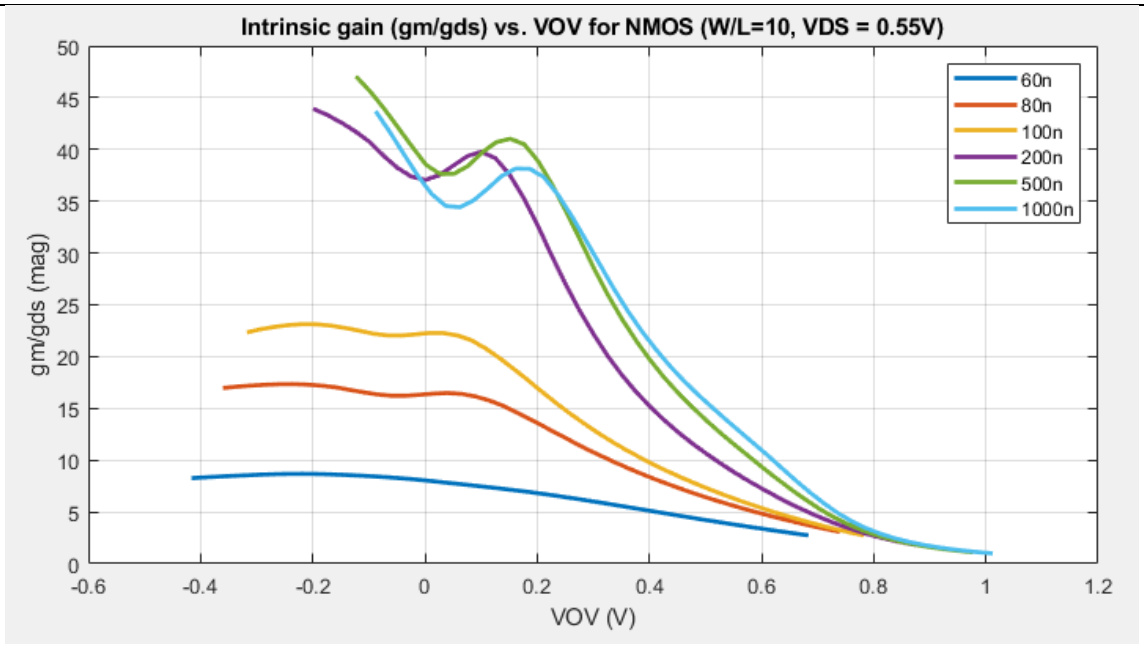


Fig. 3

Figure 3: $\frac{g_m}{g_{ds}}$ vs V_{OV} across gatelength L

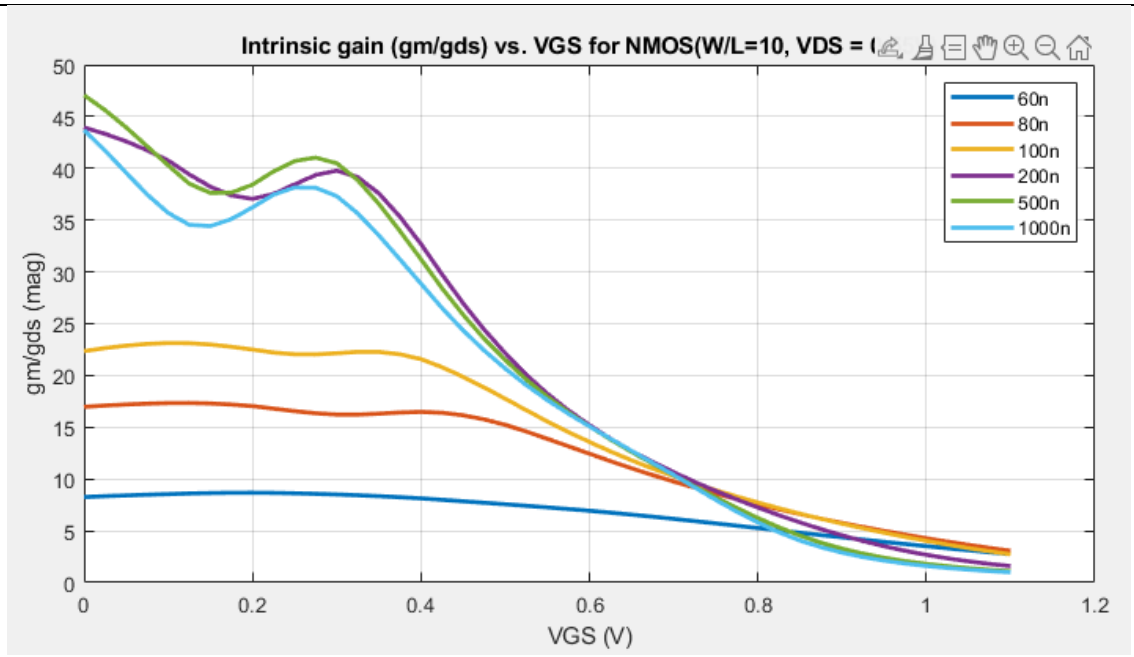


Fig. 4

Figure 4: $\frac{g_m}{g_{ds}}$ vs V_{GS} across gatelength L

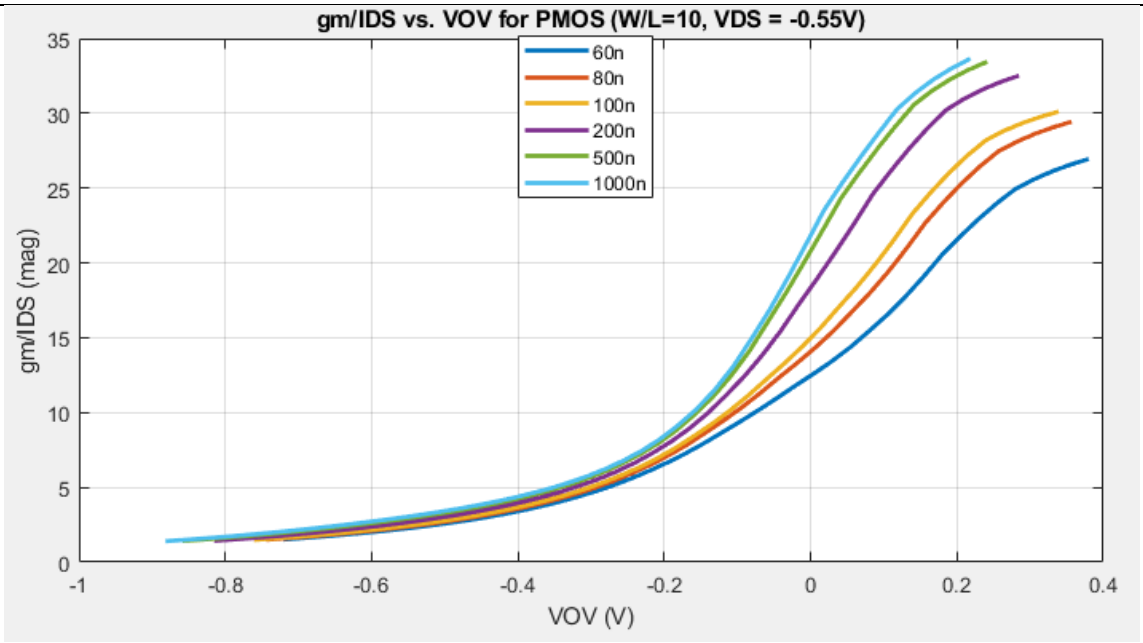


Fig. 5

Figure 5: $\frac{g_m}{I_D}$ vs V_{OV} across gatelength L

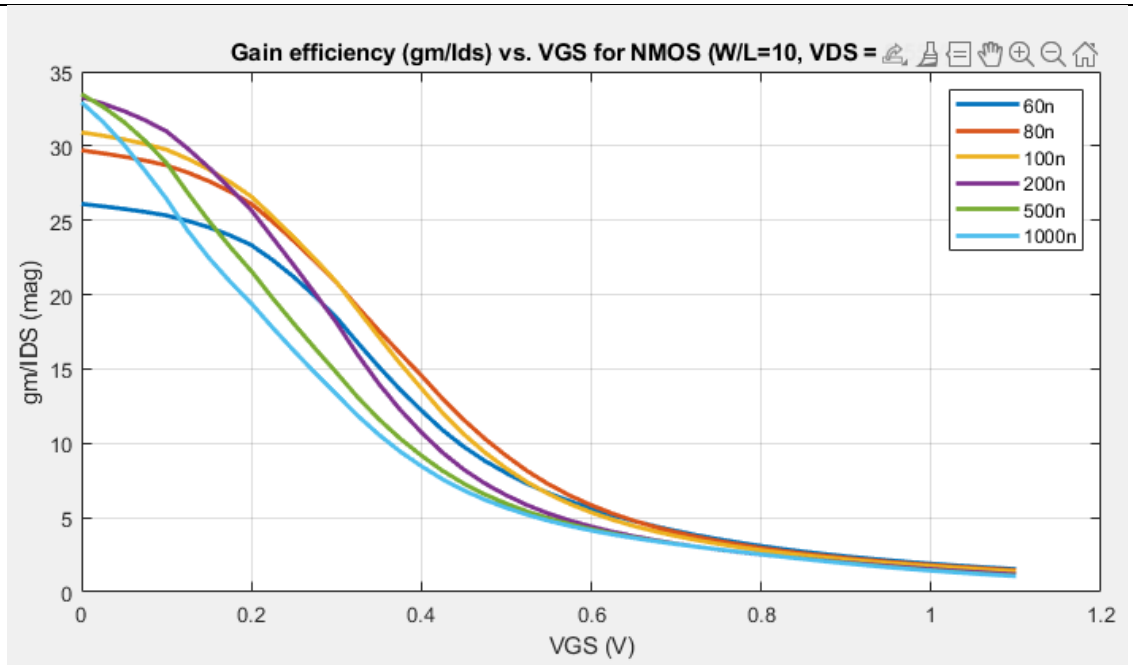

Gain efficiency (gm/Ids) vs. VGS for NMOS (W/L=10, VDS =
gm/IDS (mag)
VGS (V)
60n
80n
100n
200n
500n
1000n
0
5
10
15
20
25
30
35
0
0.2
0.4
0.6
0.8
1
1.2

Fig.6

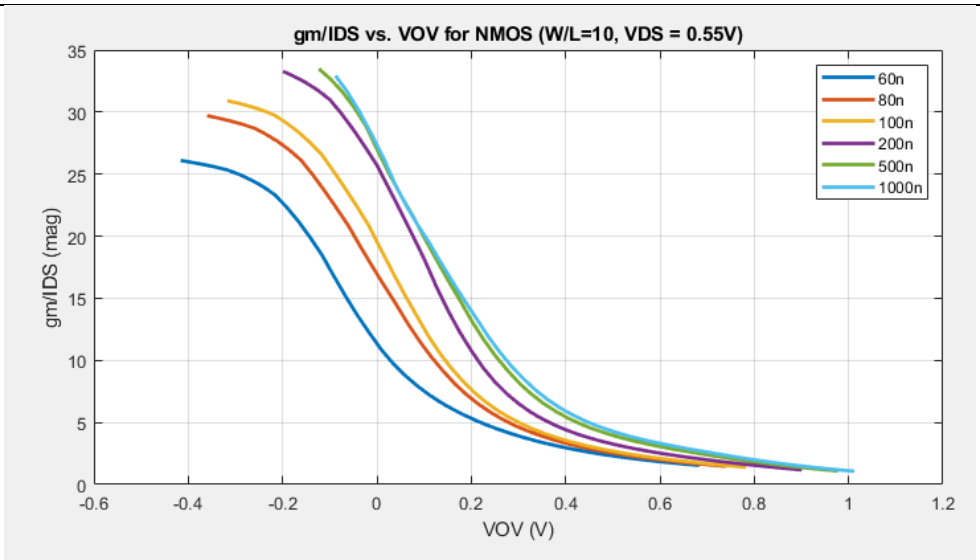

gm/IDS vs. VOV for NMOS (W/L=10, VDS = 0.55V)
gm/IDS (mag)
VOV (V)
60n
80n
100n
200n
500n
1000n
0
5
10
15
20
25
30
35
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2

Fig. 7

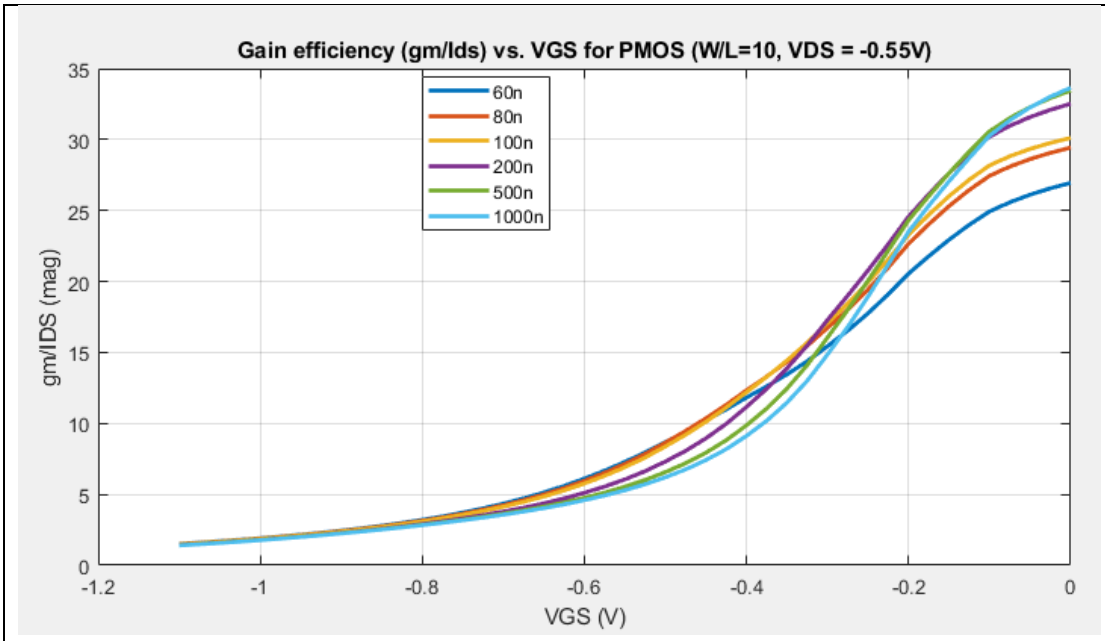


Fig. 8

Handcalculations:

Given Specifications:

$C_L = 3pF$

$f_{GBW} = 50MHz$

DC gain = 48dB

A necessary precaution is that zero compensation of the second pole and frequency gain bandwidth should be $f_{2nd\ pole} \geq f_{GBW}$

$C_M = \frac{1}{4} . C_L = \frac{3pF}{4} = 0.75pF$

Second pole approximation expression for a larger miller capacitance C_M and C_L

$f_{2nd\ pole} \approx gm6 / (2\pi * C_L)$

I take $gm6 = n * g_{m1}$ where n=slope and it is $1.5 \leq n \leq 5$, I choose n= 1.5

$$g_{m1} = f_{GBW} * 2\pi * C_M = 50e6 \times 2\pi \times 0.75e-12 = 2.3565e-4S = 0.23565m\ A/V$$

For stability: $-\frac{g_{m6}}{C_L} \approx n.f_{GBW}$. Hence, $gm6 = n.\frac{C_L}{C_M}.g_{m1} = 1.5 \times \frac{3\times10-12}{0.75\times10^{-12}} \times 2.3565e-4 = 1.4139 \times 10^{-3} = 1.4139mA/V$

To maximize the voltage swing while the input stage transistors are kept in saturation, $V_{DS6} = \frac{VDD}{2} = 0.55V = 550mV$

To find the values of $i_{ds,6}, g_{ds,6}$ and $v_{gs,6}$ look the graphs

DC Voltage gain= Stage1DC gain × Stage2 DC gain

$$= \frac{g_{m2}}{g_{ds2} + g_{ds4}} \times \frac{g_{m6}}{g_{ds6} + g_{ds5}} = A_{VDC1} \times A_{VDC2}$$

$$\approx \frac{1}{2} \frac{g_{m2}}{g_{ds2}} \times \frac{g_{m6}}{g_{ds6}} \ (g_{ds5} \ll g_{ds6}) \text{ and } g_{ds4} \approx g_{ds2}$$

Check the graph of gm1/gds1 at L1=200nm, Vov=0.1v, I obtained 45.6328, gds1=5.16404×10^{-6}

Note: $\frac{gm1}{gds1} = \frac{2.3565\times10^{-4}}{5.16404\times10^{-6}} = 33.18\text{dB}$ (Mp1 is symmetry with MP2)
For I_{DS1}, from fig.6, the graph of $g_m/I_{DS} = 26.1603V^{-1}$ at V_{OV}= 0.1V, $L_1 = 200\text{nm}$, g_{m1} is already known, $I_{DS1}=9.0079 \times 10^{-6} = 9\mu A$
The values are the same for transistor 2, M_{P2} by symmetry.

$$M_{n6}$$

Starting the sizing of the transistor from the output M_{n6} on the specification because I want a small intrinsic gain ($\frac{gm6}{gd6}$) I chose $L_6 = 80\text{nm}$ so that most of the gains will be made at the first stage of the circuit and avoid short channel modulation effects, the length should not be too short.
I choose a weak inversion level for economical circuit design $V_{ov_6} = 0V$, the lower the V_{ov} the better the cost.
For current mirror, $V_{GS4} = V_{DS4} = V_{GS3}$ and $V_{GS6} = V_{DS3}$
Fig. 7, gm6/ID6 at Vov=0V,with L_6=80nm I obtained 16.0516, therefore, IDS6=$8.808 \times -5A$.
From graph gm/gds at Vov,6=0V, I obtained 16.4227, the gds6=$8.609 \times 10^{-5}S$

$$Mn3\ and\ Mn4(Symmetry)$$

For g_{m3}, from the fig. 7, g_{m3}/I_{DS3} at $V_{OV} = 0.16$ (obtained from Matlab) with L_3=200nm, which is 14.004, $I_{DS3} = I_{DS1} = 9.0079 \times 10^{-6}A$. To obtain a high gain in the first stage I increase the channel length of the transistor and work in weak inversion to have a low g_{ds3}.
I obtained $g_{m3} = 1.261 \times 10^{-4}S$. According to fig. 3, g_{m3}/g_{ds3} =38.157, $g_{ds3}=3.3048 \times 10^{-6}S$.

$$M_{P5}, M_{P7}, M_{P8}$$

For g_{m5}, I use fig. 5, gm/I_{DS} = 8.19077 at V_{OV} = -0.2V, with L_5 =1000nm I_{DS} = I_{DS6} = $I_{DS5} = 8.808e-5A$ also $|V_{DS5}| = VDD - V_{DS6}$ =0.55V.
$gm5 => 7.2144e-04S$
$V_{DSAT6} \le V_{out} \le VDD - |V_{DSAT,5}|$ which means I need a small $|V_{DSAT5}|$ which is a weak inversion regime

To find $g_{ds,5}$, use fig.1, g_{m5}/g_{ds5}= 36.8236 at $V_{OV,5}$= -0.2, I obtained $g_{ds,5}$ =$1.9592e-05S$

For transistor 7, I choose $V_{DS7} = -0.25$, $V_{D7} = V_{S1} = 1.1 - 0.25 = 0.85V$. Increasing the channel length of M_{P7} will allow it to function as a good current source, so I choose $L_7 = 1000\text{nm}$ (good current source) where $V_{in} < V_{DD} + V_{ov,7} + V_{GS,1}$, to obtain a high input common mode range, the V_{ov7} is kept low using low inversion regime $V_{OV} = -0.2V$.
Because the gate and drain of M_{P8}, also M_{P7} and MP8 have $L_7 = L_8 = L_5$ to better control the current ratio between them.
$V_{GS,5} = V_{DS,8} = V_{GS7} = V_{DS,8}$, I found the V_{ov7}, V_{ov8} from fig.1.
This gives $I_{DS7} = 2 * I_{DS1} = 2(9.0079 \times 10^{-6}A) = 1.801 \times 10^{-5}A$

DC Voltage gain= Stage 1 gain × Stage 2 gain
$$= \frac{g_{m2}}{g_{ds2}+g_{ds4}} \times \frac{g_{m6}}{g_{ds6}+g_{ds5}} = A_{VDC1} \times A_{VDC2} =$$
$$\approx \frac{1}{2}\frac{g_{m2}}{g_{ds2}} \times \frac{g_{m6}}{g_{ds6}}\ (g_{ds5} \ll g_{ds6}) \text{ and } g_{ds4} \approx g_{ds2}$$

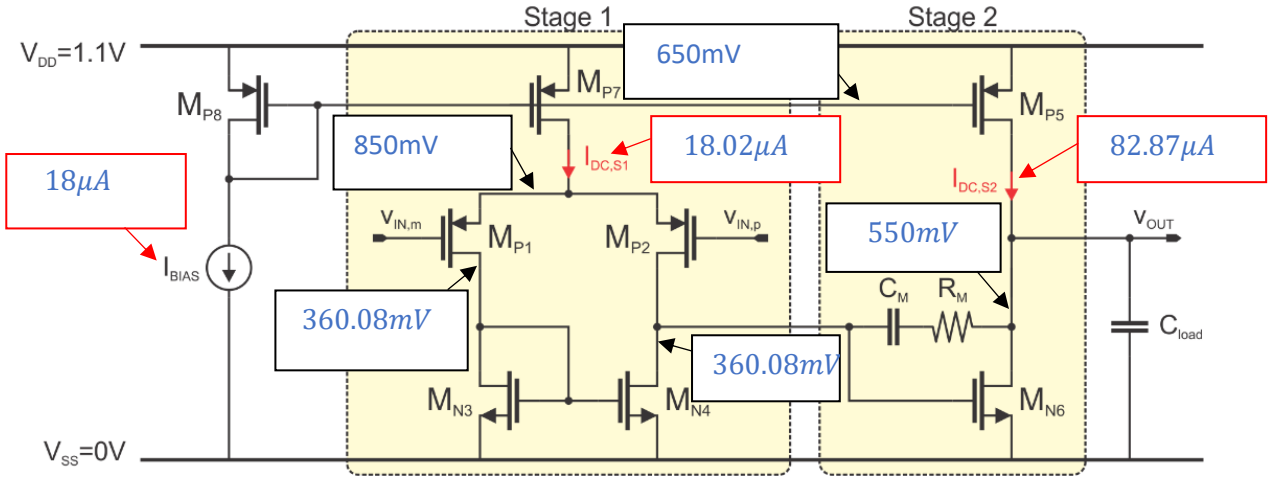
$g_{ds2} = g_{o2} = g_{ds1} = 5.164 \times 10^{-6}S$
$g_{ds4} = g_{o4} = g_{ds3} = 3.3048 \times 10^{-6}S$
$g_{ds5} = g_{o5} = 1.9592 \times 10^{-5}S$
$g_{ds6} = g_{o6} = 8.609 \times 10^{-5}S$
$g_{m2} = gm1 = 235.65\mu S$
$gm6 = 1413.9\mu S$
$A_{V1} = g_{m1}/g_{o4} + g_{02}$=27.83
$A_{V2} = g_{m6}/g_{05} + g_{o6}$=13.38

DC gain= $A_{V(TOTAL)} = A_{V1} \times A_{V2}$=372.33
$AV[dB] = 20\log(AV) = 51.4dB$
Node 1; $w_{p1} = \frac{g_{m3}}{C_{n1}}$; $w_{z1} = 2 * \frac{g_{m4}}{C_{n1}}$
Node 2: $w_{p2} = \frac{g_{ds1} + g_{ds3}}{C_M(1+A_{V2})}$; $w_{z2} = \frac{1}{C_M(\frac{1}{gm6} - R_M)}$
To compensate for non-dominant pole, I used the formula,
$R_M = \frac{C_L}{C_M * g_{m6}} + \frac{1}{g_{m6}} = \frac{C_L}{C_M * g_{m6}} + \frac{1}{1413\times10\text{^}-6S} = \frac{3pF}{0.75pF\times1.4139\times10^{-3}} + 707.2635 = 3.5363k\Omega$
R_M is such that the pole on node 3 is compensated by zero on node 2, which means $w_{p3} = w_{z2}$
The dominant pole frequency= $w_{P2} = P_{dom} = \frac{f_{GBW}}{A_{(VTOTAL)}} = \frac{50MHz}{51.4dB} = 134.4086\ kHz =$
Rather, $P_{dom} = \frac{f_{GBW}}{A_{(VTOTAL)}} = 2\pi \times 50 \times \frac{10^6}{372.33} = 0.84388MHz$

Total Phase Shift = $0° - \arctan\frac{f_{GBW}}{P_{dom}} = 0° - \arctan\left(\frac{50MHz}{134.33KHz}\right) = -89.84°$
Adding w_{z2} to PM I obtain, Phase Margin:
$PM = \text{Total Phase Shift} - (-180°) = -89.88° + 180° = 90.15°$
Node3: $w_{p3} = \frac{g_{m6}}{C_L} = \frac{1.4139\times-3S}{3pF}$

2) Place all the calculated voltages (in the blackbox) and all currents (in the red box) on the circuit depicted below and calculate also:

$V_{cm,in,min}$	$V_{gs3} - V_{dsat1} + V_{gs1}$= 115.4mV
$V_{cm,in,max}$	$V_{DD} - V_{gs1} - V_{ov} = 736.4$mV
$V_{out,min}$	$V_{dsat,6} = 62.91mV$
$V_{out,max}$	$V_{DD} + \lvert Vdsat, 5\rvert = 914.4mV$
P_{diss}	$V_{DD}\left(I_{ds} + I_{ds,7} + I_{ds,8}\right) = 0.1$mW



Design: Implementation of the OpAmp in MATLAB and LTspice (10 points):

3) Based on your handcalculations, create your OpAmp design in MATLAB. After completion, please insert your final and entire MATLAB code after the appendix (i.e. at the end of this report-document).

Remark:

- *Please make sure that all widths W are below 1mm*

4) Fill out both tables depicted below. All values are to be determined in MATLAB.

Device sizes and bias point parameters according to MATLAB

Device	W [µm]	L [nm]	I_{ds} [µA]	V_{OV} [V]	g_m [S]	g_{ds} [S]	g_m/g_{ds} [-]	$V_{ds,sat}$ [V]	V_{ds} [V]
M_{p1}	55.47	200	9.25	$100.00m$	235.62μ	5.28μ	44.625	$-44.56m$	$-550.00m$
M_{p2}	55.47	200	9.25	$100.00m$	235.62μ	5.28μ	44.625	$-44.56m$	$-550.00m$
M_{n3}	0.481	200	9.25	$162.32m$	125.40μ	4.50μ		$115.56m$	$360.08m$
M_{n4}	0.481	200	9.25	$162.32m$	125.40μ	4.50μ		$115.56m$	$360.08m$
M_{p5}	26.94	1000	82.87	$-200.00m$		18.50μ		$-175.21m$	$-550.00m$
M_{n6}	6.86	80	82.87	$-1.46u$	$1.410m$	86.46μ	16.308	$63.11m$	$550.00m$
M_{p7}	7.28	1000	18.47	$-203.25m$		33.17μ		$-177.77m$	$-189.92m$
M_{p8}	6.21	1000	18.49	$-201.23m$		5.61μ		$-176.18m$	$-418.19m$

Device	Units	Value
C_M	pF	0.750
R_M	Ω	3536.3
I_{BIAS}	A	18.02μ

5) Based on the parameters filled in the table above, design your OpAmp in LTspice. After completion, please insert your final and entire LTspice-netlist after the appendix (i.e. at the end of this report-document).

Experiment (10 points):

In this section you will simulate the following:

A. Frequency Response in MATLAB and LTspice
B. Noise contribution in LTspice
C. Linearity in LTspice

Note on how to present graphs in general:

- *When you make a graph, put labels on every axis to make clear what you are showing. Also, show the units!*
- *When you plot multiple curves on one graph, add a legend.*

A. Frequency Response in MATLAB and LTspice

Note on how to present graphs in this sub-section:

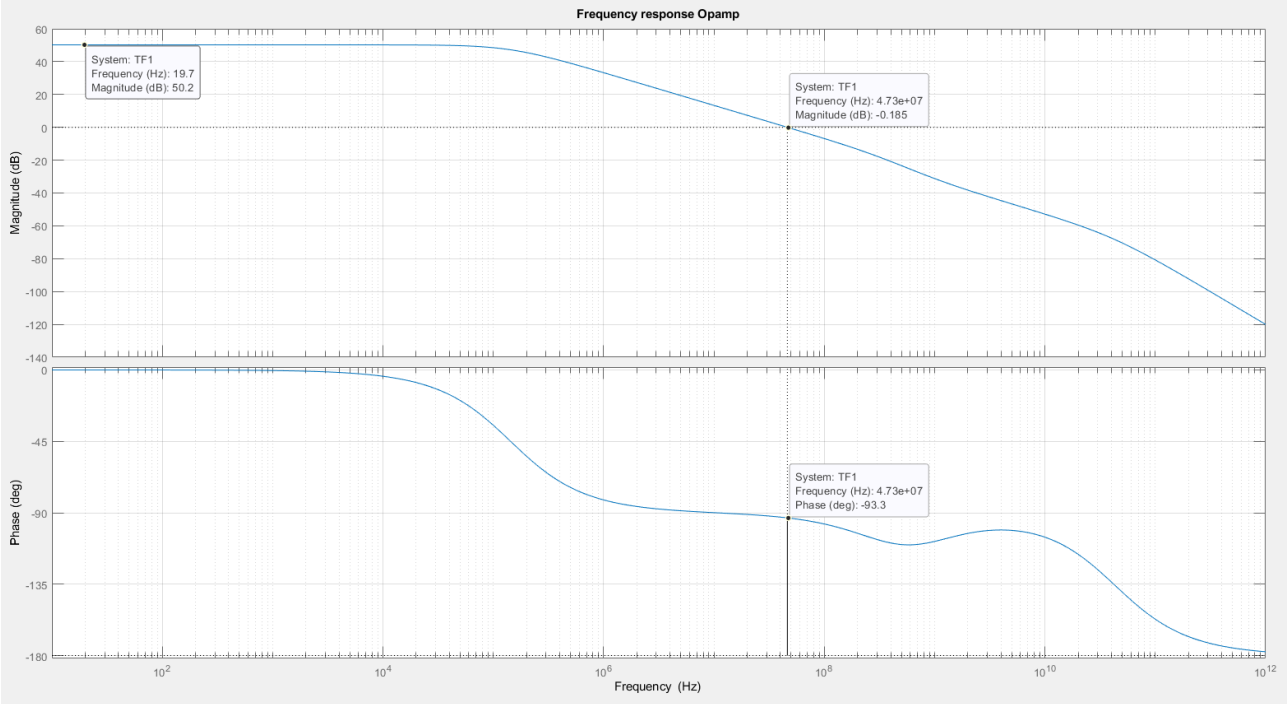
- *For magnitude plots use dB (linear scale) vs. Hz (logarithmic scale)*
- *for phase plots use ° (linear scale) vs. Hz (logarithmic scale).*

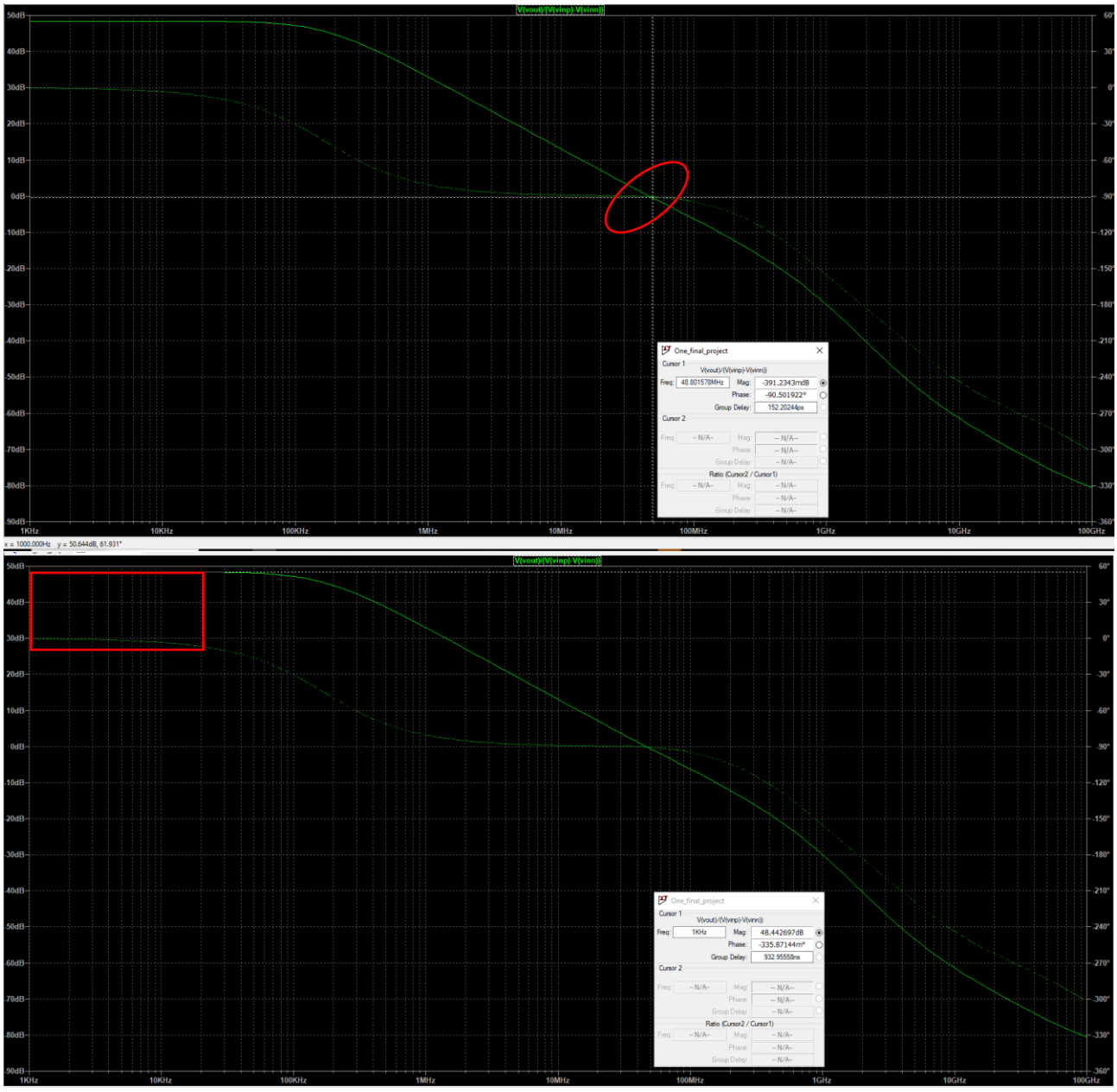
6) Simulate A_v (small voltage gain) with C_M and R_M in MATLAB and LTspice. Then, paste both A_v-curves in seperate plots below such that the difference between MATLAB and LTspice can be clearly seen. In addition to that, indicate the resulting phase margin (PM) in both plots.

Plots:

Simulator	PM (deg)
MATLAB	86.7
LTspice	89.5

7) Explain, analyse and interpret the results in "6)". Furthermore, if you observe significant differences between MATLAB and LTspice, explain why.





8) Simulate A_v in LTspice for the following cases:
   a. No compensation network ($C_M = R_M = 0$).
   b. With compensation capacitor but no compensation resistor ($R_M = 0$).
   c. With both the compensation capacitor and the compensation resistor.

   Then, show all 3 cases in seperate plots such that the differences are clearly visible. Furthermore, indicate on each plot the resulting PM.

Plot:

A. No compensation Network($C_M = R_M = 0$)

B. With Compensation Capacitor but no compensation Resistor ($R_M = 0$)

C. With both the compensation capacitor and the compensation resistor

There is a slight difference of $8.4°$ between the MATLAB PM ($86.7°$) and LT Spice ($89.5°$) because all the capacitances are not considered in MATLAB according to my hand calculations , the increase in PM in LTSpice is based on the frequency of the dominant pole which is smaller than that of the frequency of the dominant pole in MATLAB.

In MATLAB simulation, the zero is placed in the right hand half plane at the non-dominant pole frequency to compensate the non-dominant pole which helped in achieving the obtained PM, two poles and one zeros.

Therefore, the LT Spice and MATLAB use difference database, for example the difference in V_{th} and V_{ov} in the two softwares

Case	PM [deg]
No compensation network	180-178= 2
No compensation resistor	180-125= 54.7
With both the compensation	180-90.2=89.5

9) Explain, analyse and interpret the results in “8)”.

Case A. When there is no RM and CM, the PM decreases a lot, the GBW also becomes larger. The dominant and non-dominant poles both shifted to higher frequencies(299.5MHz), which means the circuit may face instability and be greatly perturbed by C_{load} adding its effects around the cross over

frequency which which reduces the PM because the two poles are also close to the cross over frequency.

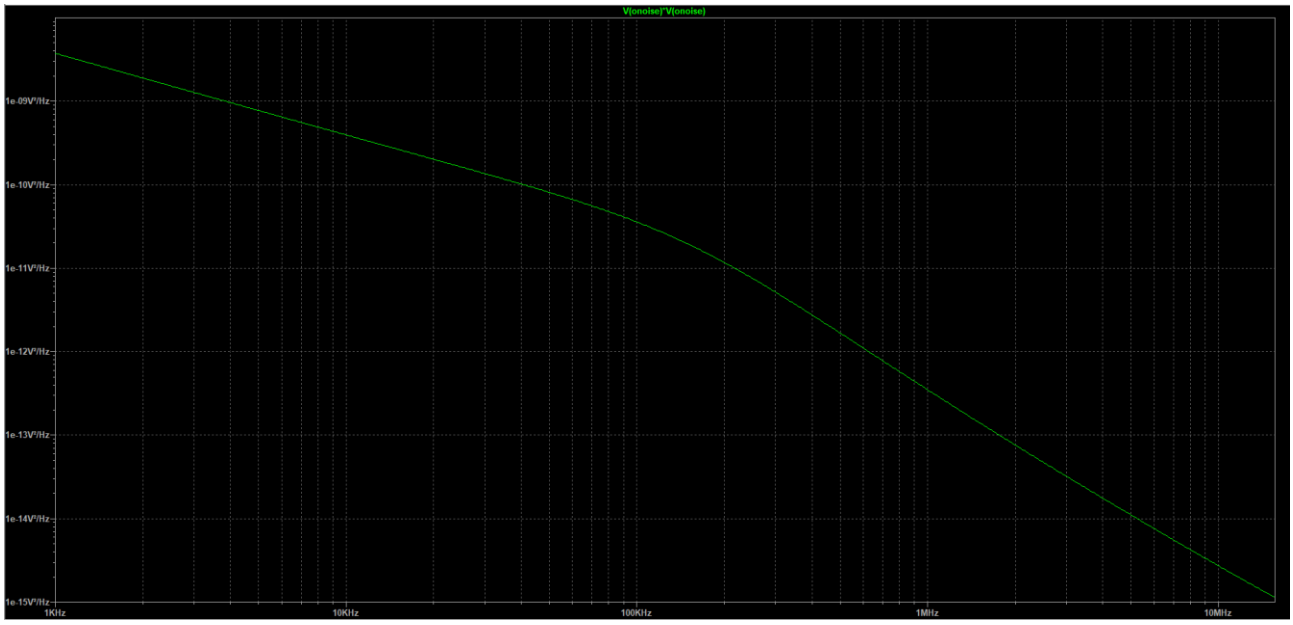
Case B. With Compensation Capacitor but no compensation Resistor ($R_M = 0$), the compensation capacitor ($C_M = 0.75$) explores the effect of miller capacitance by causing pole splitting, moves the pole of the C_{load} (non-dominant pole) away to higher frequency than the cross frequency. The non-dominant pole and the positive zero induce $90°$ phase shift each, the value of the zero is high to have effect on the PM. The dominant pole is moved to a lower frequency (41.04MHz), this improves the PM ($54.7°$).

Case C: With both the compensation capacitor and the compensation resistor, the R_M added allows the positive zero Z2 to be moved to lower frequency at the left half plane, the z2 helps in compensating the non dominant pole (P3) with absolute value close to the cross over frequency. This results in an increase PM where it is mostly needed. The unity gain frequency also increase because there is no more non-dominant pole.

B. Noise contribution in LTspice

10) Simulate the output-referred noise voltage power density $\overline{dv_{out,eq}^2}/\Delta f$ over an appropriate frequency range in LTspice. Then, insert the $\overline{dv_{out,eq}^2}/\Delta f$-plot below.

Plots:



11) Explain, analyse and interpret the results in "10)".

I plotted the graph until 10.fGBW which corresponds to the DC gain I obtained approximately, in the plot of output-referred noise voltage power density decreases exponentially until $0\ dB\ (v^2/Hz)$ in logarithmic scale, as the frequency increases towards ($10 \times 48.44MHz$). This is the characteristics of flicker noise which depends on 1/f. The nooise decreases stronger as the gain is attenuated which add further to the flicker noise. At around 23MHz, slope begins to change which corresponds to white noise (thermal noise) which is directly proportional to the frequency.

12) Simulate the input-referred noise spectral density by using the .NOISE option in LTspice. Also, by using LTspice, determine the total output integrated noise of the OpAmp. For that, take the integration bandwidth from 10 kHz to $10 \cdot f_{GBW}$. Furthermore, copy and paste the top 9 contributing elements using "View->SPICE Error Log" option and insert that list below.

out_totnvrms: INTEG(v(onoise))=0.00470396 FROM 1000 TO 8e+10

out_mp1nvrms: INTEG(v(mp1))=0.00165599 FROM 1000 TO 8e+10

out_mp2nvrms: INTEG(v(mp2))=0.00165923 FROM 1000 TO 8e+10

out_mp7nvrms: INTEG(v(mp7))=6.81114e-05 FROM 1000 TO 8e+10

out_mp8nvrms: INTEG(v(mp8))=3.02047e-05 FROM 1000 TO 8e+10

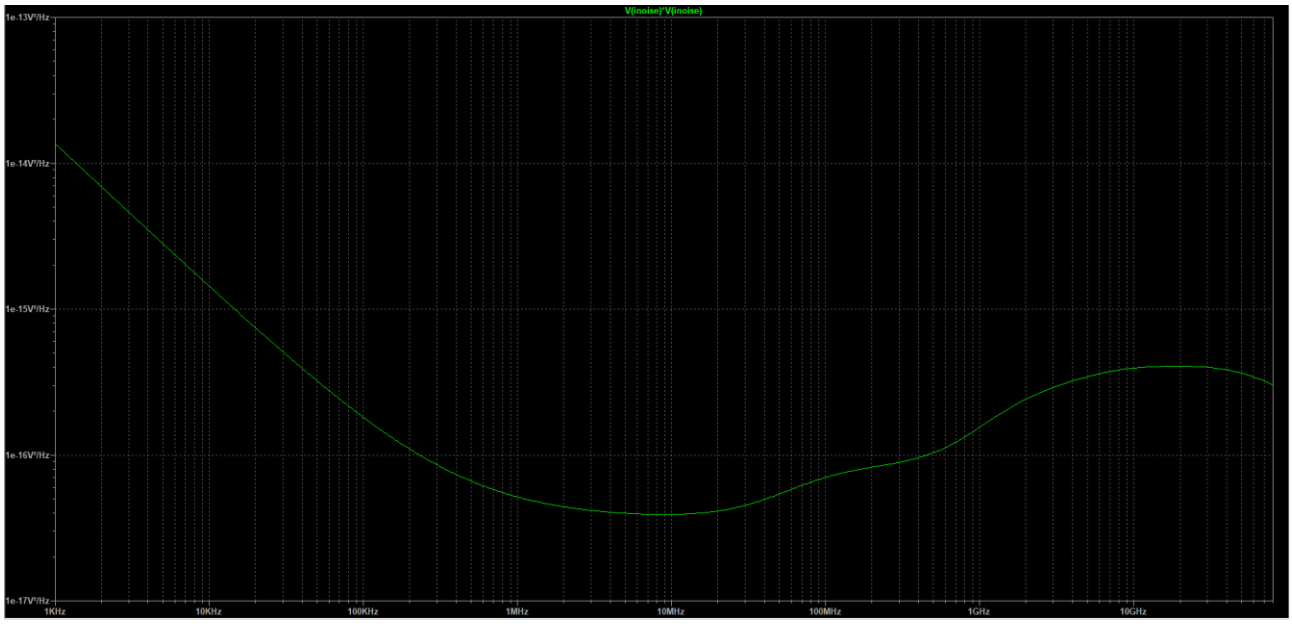
out_mp5nvrms: INTEG(v(mp5))=3.74476e-05 FROM 1000 TO 8e+10

out_mn6nvrms: INTEG(v(mn6))=9.59835e-05 FROM 1000 TO 8e+10

out_mn4nvrms: INTEG(v(mn4))=0.00299437 FROM 1000 TO 8e+10

out_mn3nvrms: INTEG(v(mn3))=0.00276476 FROM 1000 TO 8e+10

out_rmnvrms: INTEG(v(rm))=7.35347e-05 FROM 1000 TO 8e+10



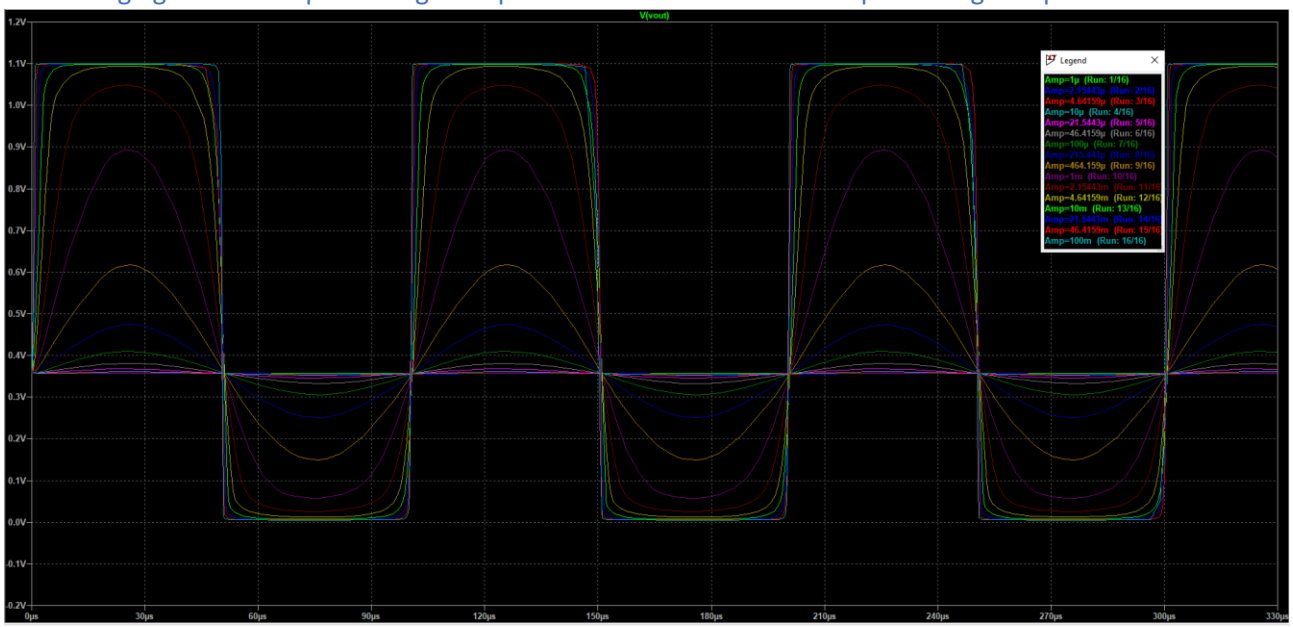
13) Explain, analyse and interpret the results in “12)”.

The OTA's total output integrated RMS noise is measured to be $\mathbf{0.00470396}\ V_{rms}$. At the first stage, the major contributors to the overall noise are Mp1, Mp2, mp3, and MP4. This noise is further amplified by the second stage. On the other hand, the noise contribution of MN6 and MP5 is reduced because they are not amplified by the second stage. MP7, MP8, and MP5, which function as current mirrors, exhibit significantly lower noise contributions since they are not amplified. They have longer gate lengths and are evenly distributed across the two branches of the OTA circuit, generating a common-mode contribution that is rejected by the OTA's common-mode rejection.

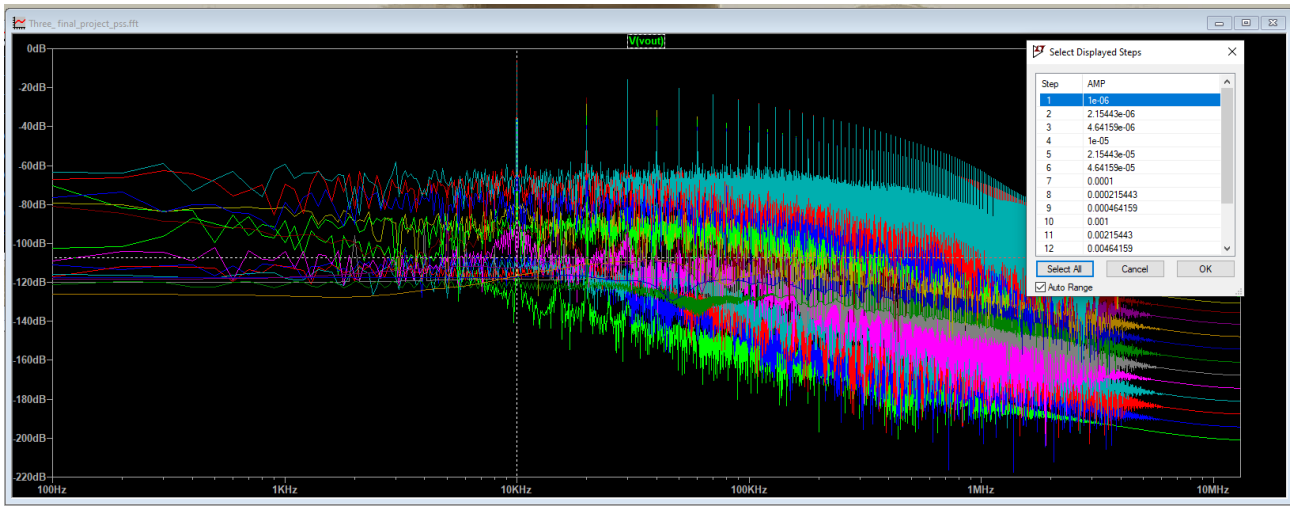
In the graph, there is an initial low-frequency region corresponding to flicker noise, followed by a flat region indicating white noise. The flicker noise, which exhibits a 1/f dependency, is inversely proportional to the

transistor area, as observed in the graph. As the frequency increases, there is a slight increase in noise due to the decreasing gain for input-referred RMS noise, causing the noise spectral density to be divided by a diminishing gain

The Voltage gain and Output Voltage Amplitude as a function of the Input Voltage amplitude



Frequency (Hz) response of the amplitude and Voltage gain as a function of the input voltage amplitude.

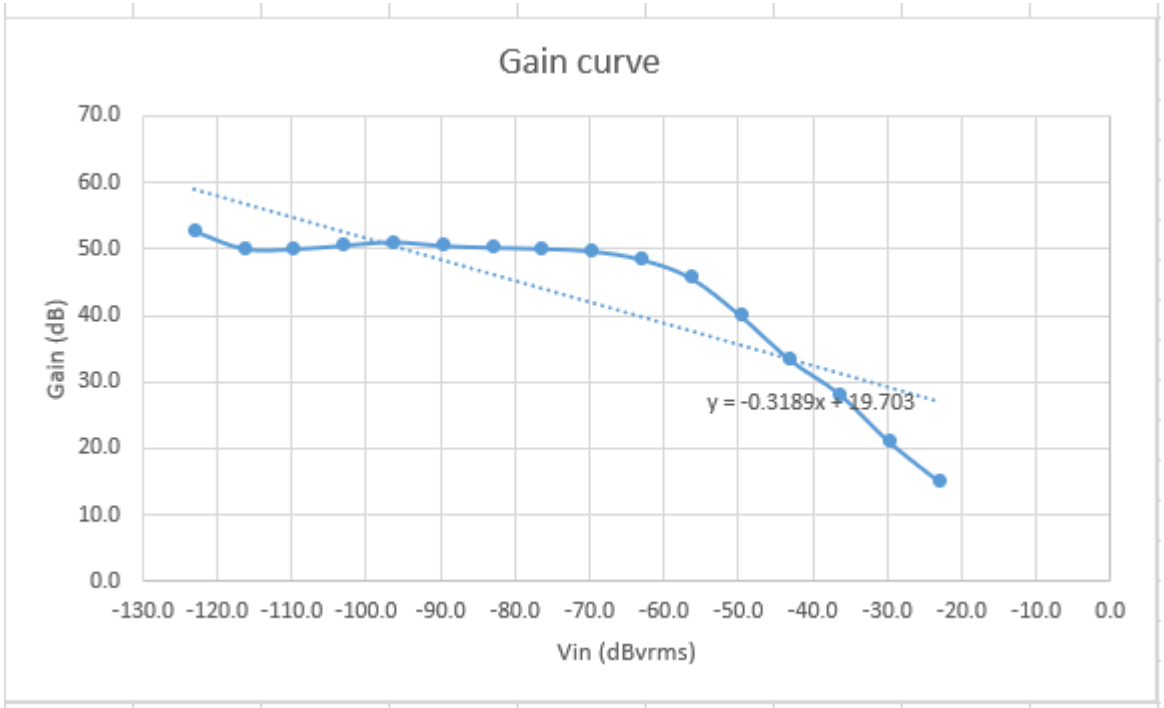


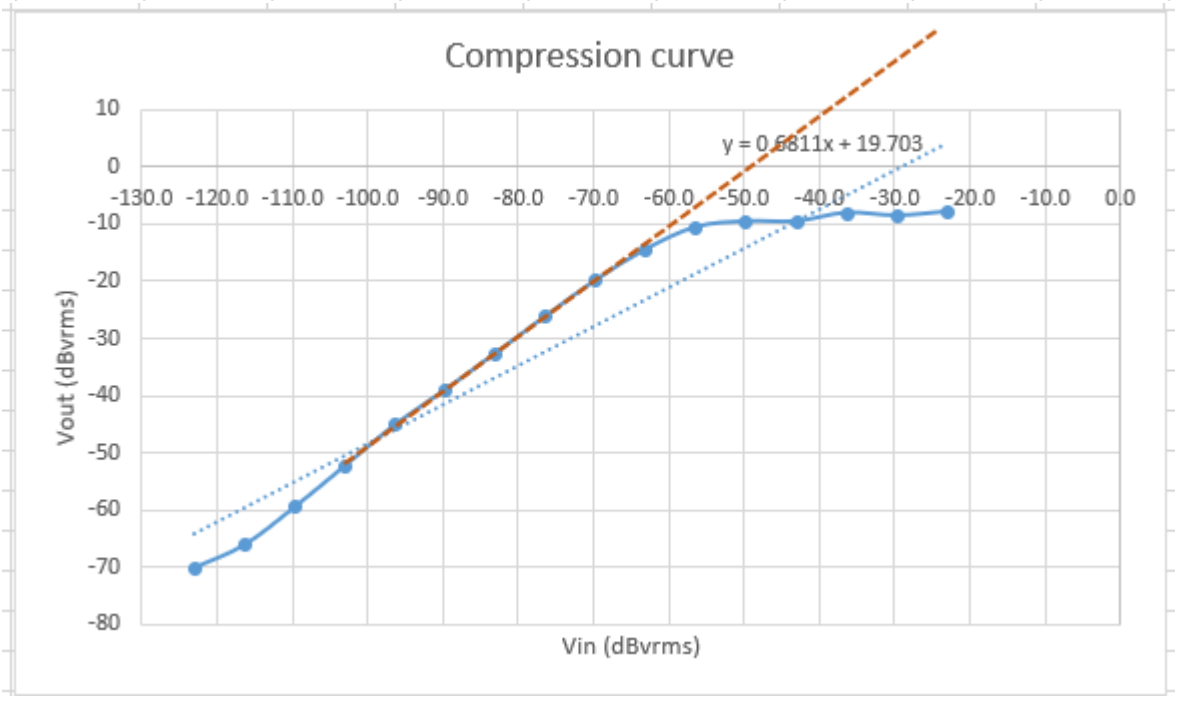
C. Linearity in LTspice

14) Simulate the output voltage amplitude and voltage gain as a function of the input voltage amplitude in LTspice. Then, insert the plots below and indicate the 1-dB compression point.

Plots:

Vin (V)	Vin (dBvrms)	Vout (dBvrms)	Gain (dB)
0.000001	-123.0	-70.208356	52.8
2.1544E-06	-116.3	-66.2	50.1
4.6415E-06	-109.7	-59.6	50.1
0.00001	-103.0	-52.4	50.6
2.1544E-05	-96.3	-45.2	51.1
4.6415E-05	-89.7	-39.1	50.6
0.0001	-83.0	-32.7	50.3
0.00021544	-76.3	-26.2	50.1
0.00046415	-69.7	-19.9	49.8
0.001	-63.0	-14.5	48.5
0.0021544	-56.3	-10.6	45.7
0.0046415	-49.7	-9.6	40.1
0.01	-43.0	-9.6	33.4
0.021544	-36.3	-8.1	28.2
0.046415	-29.7	-8.6	21.1
0.1	-23.0	-7.9	15.1





15) Explain, analyse and interpret the results in "14)".

There are two saturation points observed which are 0V and 1.1V (VDD) which creates non-linearity that causes the sine wave to be compressed as the voltage keeps increasing it turns the waveform to become a square waveform preventing further increase in amplitude and reaching the saturation of the output voltage, at this time the first order linear coefficient K1 has a lot of effect on the fundamental tone output according to the relationship fundamental tone Output $= K_1A_1 + 3/4K_3A_1^3$.

For graph B (the compression plot ($V_{out} \times V_{in}$), 1-dB compression point is observed at -89.7dB with the input Voltage Amplitude value of 4.6415×10^{-05} . The linearity of the OTA for input signals with varying amplitudes, show a linearity at low Vin and becomes saturated at higher Vin.

For graph A (Gain x Vin), the same behavior is noticeable with the gain that is very close to 48dB for low Vin while it begins to reduce for higher Vin

Conclusion (10 points):

16) Conclude your experiment by filling the editable fields in the performance table depicted below

Metric	Units	Specification	from hand-calculations	MATLAB	LTspice
DC gain	magnitude	251.2	295.1	231.7	264.2
DC gain	dB	48	49.4	47.3	48.4
Gain-Bandwidth frequency f_{GBW}	MHz	50	51	50.2	48.80
Dominant pole frequency f_{dom}	kHz	199	173	216.7	194.3
PM	°	> 70	90.1	86.7	89.5
$V_{in,cm,max} - V_{in,cm,min}$	V		0.61	0.622	0.318
$V_{out,max} - V_{out,min}$	V	> 0.7	0.85	0.862	0.856
P_{diss}	mW		0.13	0.1	0.14
$\overline{v_{out,eq}^2}$ (output-referred noise)	V_{rms}				0.000457165

17) Comment about the deviations between hand calculation, MATLAB and LTspice values found above. Conclude, what the causes of such deviations are.

There is a slight difference of $8.4°$ between the MATLAB PM (86.7 °) and LT Spice ($89.5°$) because all the capacitances are not considered in MATLAB according to my hand calculations , the increase in PM in LT Spice is based on the frequency of the dominant pole which is smaller than that of the frequency of the dominant pole in MATLAB.
In MATLAB simulation, the zero is placed in the right hand half plane at the non-dominant pole frequency to compensate the non-dominant pole which helped in achieving the obtained PM, two poles and one zeros. Therefore, the LT Spice and MATLAB use difference database, for example the difference in V_{th} and V_{ov} in the two softwares

The value of the f_{dom} which is $f_{dom} = f_{GBW} / (10^{(DCgain/20)})$ for handcalculation (173kHz) and Matlab (164kHz) are almost equal because their DC gain gain value are almost the same unlike LT spice (194kHz) that has a higher DC gain value.

18) If you have results which are not passing the specifications: conclude for each of those result what the cause is and what you need to do in order to make it pass, if you were repeating this experiment.

To conclude, my circuit has successfully passed all the requirements in MATLAB, hand calculation and LT Spice which includes, the width that is less than 1mm, all specifications error margin of 10% are all respected. The difference in the DC gain bandwidth could be due to the R_M and C_M values used which do not perfectly correlated .
The handcalculated Rm value worked both in LT spice and in the MATLAB.

Appendix:

Insert your MATLAB code here:

```
% Analog Electronics final project
clc;
close all;
clear;

addpath(genpath('circuitDesign'));
addpath(genpath('functions'));
addpath(genpath('models'));

load('UMC65_RVT.mat');

%% Initializations
designkitName           = 'umc65';
circuitTitle            = 'Analog Design - Project';
elementList.nmos        = {'Mn3','Mn4','Mn6'};
elementList.pmos        = {'Mp1','Mp2','Mp5','Mp7','Mp8'};

choice.maxFingerWidth = 10e-6;
choice.minFingerWidth = 200e-9;

simulator                       = 'spectre';
simulFile                       = 0;
simulSkelFile           = 0;
spec                            = [];
analog                  = cirInit('analog',circuitTitle,'top',elementList,spec,choice,...

        designkitName,NRVT,PRVT,simulator,simulFile,simulSkelFile);
analog                  = cirCheckInChoice(analog, choice);


%% Project: circuit
disp('                                  ');
disp('  VDD        VDD             VDD          ');
disp('   |          |               |           ');
disp('  Mp8-+---------Mp7--------------------Mp5          ');
disp('   |--+        |               |           ');
disp('   |       +--+--+       node 3->  +-----+---OUT  ');
disp('   |       |    |             |     |       ');
disp('   |    IN1--Mp1   Mp2--IN2          |     |       ');
disp('   |       |    |             |     |       ');
disp('   | node 1-> |    | <-node 2        |     Cl      ');
```

```
disp('   |        |--+  +------+-Cm---Rm----+     |       ');
disp(' Ibias       | | |     |    ↑       |     |       ');
disp('   |        Mn3-+-Mn4     |  node 4    |     |       ');
disp('   |        |    |     +-----------Mn6    |       ');
disp('   |        |    |               |     |       ');
disp('  GND       GND   GND             GND   GND      ');

%% AI: Implement your OpAmp according to your handcalculations

% specifications
spec.VDD = 1.1;
spec.fGBW = 50e6; %MHz
spec.Cl = 3e-12;  %pF
spec.GBW=spec.fGBW*(2*pi);
spec.gain = 48;   %DB
spec.Cm = 0.25 * spec.Cl;  %Cm=0.75pF
spec.Rm = 3.5363e+3;
I_bias =18e-6;

% Design Mn6
Mp1.gm = spec.fGBW * 2 * pi * spec.Cm;
n = 1.5;
Mn6.gm = n * spec.Cl * Mp1.gm / spec.Cm;

Mn6.vds = 550e-3;
Mn6.vov = 0;
Mn6.lg = 80e-9;
Mn6.vsb = 0.0;

Mn6.vth = tableValueWref('vth', NRVT, Mn6.lg, 0, Mn6.vds, Mn6.vsb); % vgs still unknown
Mn6.vgs = Mn6.vth + Mn6.vov;

Mn6.w = mosWidth('gm', Mn6.gm, Mn6);
Mn6 = mosNfingers(Mn6);
Mn6 = mosOpValues(Mn6);

% design Mp5
Mp5.ids = Mn6.ids;
Mp5.vsb = 0;
Mp5.lg = 1000e-9;
Mp5.vov = -0.2;
Mp5.vds = -(spec.VDD - Mn6.vds);

Mp5.vth = tableValueWref('vth', PRVT, Mp5.lg, 0, Mp5.vds, Mp5.vsb);
Mp5.vgs = Mp5.vth + Mp5.vov;
```

```
Mp5.w = mosWidth('ids', Mp5.ids, Mp5);
Mp5 = mosNfingers(Mp5);
Mp5 = mosOpValues(Mp5);

% design Mp1 and Mp2

Mp1.vds = -spec.VDD / 2;
Mp1.vsb = 0;
Mp1.vov = 0.1;
Mp1.lg = 200e-9;
Mp1.vth = tableValueWref('vth', PRVT, Mp1.lg, 0, Mp1.vds, Mp1.vsb);
Mp1.vgs = Mp1.vth + Mp1.vov;


% Mp2.ids = Mp1.ids;
Mp1.w = mosWidth('gm', Mp1.gm, Mp1);
Mp1 = mosNfingers(Mp1);
Mp1 = mosOpValues(Mp1);

Mp2 = cirElementCopy(Mp1,Mp2);

% design Mn3 and Mn4
Mn3.ids = Mp1.ids;
Mn3.vds = Mn6.vgs;
Mn3.vgs = Mn3.vds;
Mn3.vsb = 0;
Mn3.lg = 200e-9;

Mn3.vth = tableValueWref('vth', NRVT, Mn3.lg, Mn3.vgs, Mn3.vds, Mn3.vsb);
Mn3.vov = Mn3.vgs - Mn3.vth;

Mn3.w = mosWidth('ids', Mn3.ids, Mn3);
Mn3 = mosNfingers(Mn3);
Mn3 = mosOpValues(Mn3);

Mn4 = cirElementCopy(Mn3, Mn4);

% design Mp7

Mp7.vds = -(spec.VDD + Mp1.vds - Mn3.vds);
Mp7.lg = 1000e-9;
Mp7.ids = 2 * Mp1.ids;
Mp7.vsb = 0;
Mp7.vgs = Mp5.vgs;

Mp7.vth = tableValueWref('vth', PRVT, Mp7.lg, Mp7.vgs, Mp7.vds, Mp7.vsb);
```

```
Mp7.vov = Mp7.vgs - Mp7.vth;

Mp7.w = mosWidth('ids', Mp7.ids, Mp7);
Mp7 = mosNfingers(Mp7);
Mp7 = mosOpValues(Mp7);

% design Mp8

Mp8.vgs = Mp7.vgs;
Mp8.vds = Mp8.vgs;
Mp8.lg = Mp7.lg;
Mp8.ids = Mp7.ids; % = I_bias
Mp8.vsb = 0;

Mp8.vth = tableValueWref('vth', PRVT, Mp8.lg, Mp8.vgs, Mp8.vds, Mp8.vsb);
Mp8.vov = Mp8.vgs-Mp8.vth;

Mp8.w = mosWidth('ids', Mp8.ids, Mp8);
Mp8 = mosNfingers(Mp8);
Mp8 = mosOpValues(Mp8);

Av_TotalGain_dB = db(Mp1.gm * Mn6.gm / (Mp1.gds + Mn4.gds) / (Mn6.gds + Mp5.gds));

%% AI: Fill out the empty variables required to plot the transfer-function.
%  meaning of each variable see comment and
%  location of nodes see line 31

AvDC1 = Mp1.gm / (Mp1.gds + Mn4.gds);  % DC gain 1st stage
AvDC2 = Mn6.gm / (Mn6.gds + Mp5.gds);  % DC gain 2nd stage
AvDC1dB = 20*log10(AvDC1);          % DC gain 1st stage [dB]
AvDC2dB = 20*log10(AvDC2);          % DC gain 2nd stage [dB]
Av_Total=AvDC1*AvDC2;
C1 = Mp1.cgd + Mp1.cgs + Mn4.cgs; % Capacitance on node 1

G1 = Mn3.gm;                    % Admittance on node 1
C2 = spec.Cm*(1+AvDC2);           % Capacitance on node 2
G2 = Mp2.gds + Mn4.gds;           % Admittance on node 2
C3 = spec.Cl;                   % Spec.Cl + spec.Cm; % Capacitance on node 3
G3 = Mn6.gm;                    % Admittance on node 3
C4=Mn6.cgs + Mn6.cgd;
p3 = G3/C3;
G4    = 1/(spec.Rm) + Mn6.gm;
%% AI: Set-up Rm, Cc and Cl and calculate the zero required for the transfer-fct
z1 = 1/((1/Mn6.gm - spec.Rm) * spec.Cm);

%% AI: Fill out the empty variables required for the performance summary
```

```
Vin_cm_min  = Mn3.vdsat;
Vin_cm_max  = spec.VDD + Mp7.vdsat + Mp1.vgs;
Vout_cm_min = Mn6.vdsat;
Vout_cm_max = spec.VDD + Mp5.vdsat;
Pdiss       = spec.VDD * (Mp8.ids + Mp7.ids + Mp5.ids);

%% Sanity check (do not modify)

disp('=====================================');
disp('=     Transistors in saturation     =');
disp('=====================================');
if mosCheckSaturation(Mp1)
        fprintf('\nMp1:Success\n')
end
if mosCheckSaturation(Mp2)
        fprintf('Mp2:Success\n')
end
if mosCheckSaturation(Mn3)
        fprintf('Mn3:Success\n')
end
if mosCheckSaturation(Mn4)
        fprintf('Mn4:Success\n')
end
if mosCheckSaturation(Mp5)
        fprintf('Mp5:Success\n')
end
if mosCheckSaturation(Mn6)
        fprintf('Mn6:Success\n')
end
if mosCheckSaturation(Mp7)
        fprintf('Mp7:Success\n')
end
if mosCheckSaturation(Mp8)
        fprintf('Mp8:Success\n\n')
end


%% Summary of sizes and biasing points (do not modify)

disp('=====================================');
disp('=    Sizes and operating points     =');
disp('=====================================');
analog = cirElementsCheckOut(analog); % Update circuit file with

% transistor sizes
mosPrintSizesAndOpInfo(1,analog); % Print the sizes of the
```

```
% Mp8.ids=1.8016e-05;
% transistors in the circuit file
fprintf('I_bias\t= %6.2fmA\nRm\t= %6.2f Ohm\nCm\t= %6.2fpF\n\n',Mp8.ids/1e-3,spec.Rm,spec.Cm/1e-12);
% I_bias =18e-6;
spec.Cl = 3e-12;
omega_pdom = (spec.GBW)/Av_Total; % Desired dominant pole
Cm_Ideal = (Mp2.gds + Mn4.gds)/(omega_pdom * (1+AvDC2));
Rm_ideal = (1+ spec.Cl/Cm_Ideal)*(1/Mn6.gm);

poles = [G1/C1, G2/C2, G3/C3, G4/C4];
p1=G1/C1;
p2=G2/C1;
p3=G3/C3;
p4=G4/C4;

z2=2*p1;
zeros = [G1/C1/2, z1];
% if abs(p2) <abs(p3)
%    fGBW=Av_Total*p2/(2*pi);
% else
%    fGBW=Av_Total *p3/(2*pi);
% end

%% Performance summary (do not modify)

disp('=====================================');
disp('=       Performance               =');
disp('=====================================');

fprintf('\nmetrik       \t result\n');
fprintf('Vin,cm,min [mV] \t%.0f\n',Vin_cm_min/1e-3);
fprintf('Vin,cm,max [mV] \t%.0f\n',Vin_cm_max/1e-3);
fprintf('Vout,cm,min [mV] \t%.0f\n',Vout_cm_min/1e-3);
fprintf('Vout,cm,max [mV] \t%.0f\n',Vout_cm_max/1e-3);
fprintf('Pdiss [mW]      \t%.1f\n',Pdiss/1e-3);

%% Ploting transfer function (do not modify)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if control toolbox in Matlab is available
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s   = tf('s');
% transfer function
TF1   = AvDC1*AvDC2*((1+s*C1/(2*G1))*(1-s*(1/z1)))/ ...
```

```
        ((1+s*C1/G1)*(1+s*C2/G2)*(1+s*C3/G3)*(1+s*C4/G4));

freq = logspace(1,12,1e3);
figure(1)
margin(TF1,2*pi*freq); grid on;
h = gcr;
setoptions(h,'FreqUnits','Hz');
title('Frequency response Opamp');
hold all
```

Insert your LTspice netlist here:

```
 vcm N006 0 0.65

E1 vinn N006 N006 vinp 1

V1 vinp N006 SINE(0 100u 1k) AC 10m

Mp8 N002 N002 N001 N001 p_11_sprvt l=1u w=6.86u

Mp7 N003 N002 N001 N001 p_11_sprvt l=1u w=7.28u

Mp1 N005 vinn N003 N003 p_11_sprvt l=200n w=55.47u

Mp2 vinter vinp N003 N003 p_11_sprvt l=200n w=55.47u

Mn4 vinter N005 0 0 n_11_sprvt l=200n w=0.481u

Mn3 N005 N005 0 0 n_11_sprvt l=200n w=0.481u

Mn6 vout vinter 0 0 n_11_sprvt l=80n w=6.86u

Mp5 vout N002 N001 N001 p_11_sprvt l=1u w=26.94u

Rm vout N004 3536

Cm N004 vinter 0.75p

Ibias N002 0 18µ

Cload vout 0 3p

.model NMOS NMOS

.model PMOS PMOS

.lib C:\Users\Shola\AppData\Local\LTspice\lib\cmp\standard.mos
```

```
.lib BSIM4_UMC65.lib

.noise V(vout) V1 dec 20 1k 80G

.meas NOISE out_totnVrms INTEG V(onoise)

.meas NOISE out_mp1nVrms INTEG V(mp1)

.meas NOISE out_mp2nVrms INTEG V(mp2)

.meas NOISE out_mp7nVrms INTEG V(mp7)

.meas NOISE out_mp8nVrms INTEG V(mp8)

.meas NOISE out_mp5nVrms INTEG V(mp5)

.meas NOISE out_mn6nVrms INTEG V(mn6)

.meas NOISE out_mn4nVrms INTEG V(mn4)

.meas NOISE out_mn3nVrms INTEG V(mn3)

.meas NOISE out_rmnVrms INTEG V(rm)

* RMS total noise calculated at the output

* RMS noise contribution from each element seen at the output

.backanno

.end
```