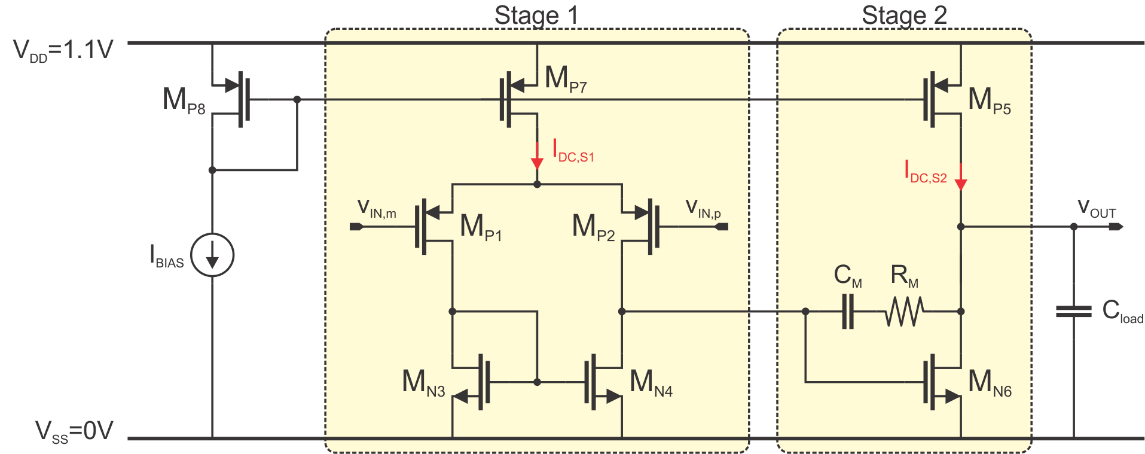
**Analog Electronic Circuits – 202**1**-202**2

**Design Project Report**

# Group data

|  |  |
| --- | --- |
| Group number |  |
| Name – Student 1 |  |
| Name – Student 2 |  |

# Goal:



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

|  |  |
| --- | --- |
|  | 2 |
| DC gain [dB] | 47 |
| [MHz] | 43 |
| Phase margin (PM) [deg] | > 70 |
| Output swing [V] | > 0.7 |

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

### Calculate the and L of each transistor (see exceptions under “Remark”) and the required and in the OpAmp circuit. For that, insert all your calculations as well as your -, -plots you used for your handcalculations below:

Hints:

* First, plot and across and gatelength L and do it again across , as presented in the 1stsession. 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. ; (2) and across ; (3)

Remarks:

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

Plots used for the handcalculations:

|  |
| --- |
|  |
| **Figure 1:** vs across gatelength |

|  |
| --- |
|  |
| **Figure 2:** vs across gatelength |

|  |
| --- |
|  |
| **Figure 3:** vs across gatelength |

|  |
| --- |
|  |
| **Figure 4:** vs across gatelength |

|  |
| --- |
|  |
| **Figure 5:** vs across gatelength |

|  |
| --- |
|  |
|  |

Handcalculations:

SPECIFICATIONS :

VVD = 1.1V ; > 0.7V ;

Since the transistors forming the current mirrors (Mp8, Mp7, Mp5) are current sources, we choose the largest length to lower the CLM 🡪 .

We also assume: .

The total DC gain = 47dB = 223.9 V/V. We set

FOR THE SECOND STAGE DESIGN :

We know that :

= - = 0.7

= - = - = 1.1 - 0.2 = 0.9V

= - = 0.9 - 0.7 = 0.2V

So we get for = = = 0.55V

And we know from the circuit that .

Second stage gain: By looking at the figure X, we can say that such that we can write : which corresponds to a Vov = 0.145V and a length 125nm.

From the figure X, Y, Z respecively, we find :

From and , 46.5

Since Mn6 is in series with Mp5 :

We know that from plot X

We also have

FOR THE FIRST STAGE DESIGN :

To find the , we are going to use the transistor 7 and 6 :

Second stage gain: By looking at the figure X, we can say that such that we can write : which corresponds to a Vov = -0.135 and a length of 150 nm.

From the figure X, Y, Z respecively, we find :

From and , 20.44

Since Mp2 forms a differential pair with Mp1 and the same between Mp3 and Mp4, these transistors have the same parameters.

For the transistors 4 et 3, we know that :

= 0.42

We then get a

To have a negligible CLM, we choose .

From the figure 3, we get

From the figure 3, we get

Now that the DC gain has been satisfied with the previous computations, we need to satisfy the frequency gain bandwidth and the phase margin conditions. To do so, we are going to compute the Miller devices that are going to maximize the phase margin and get the frequency bandwidth required.

Since the gain of the stage a bigger than the gain of the second stage, we know that the poles are splitting apart, there is an important Miller efffect. Thus, we know that there is a dominant pole between the three and that the dominant pole is the pole at node 2 :

We also know that the

So we get :

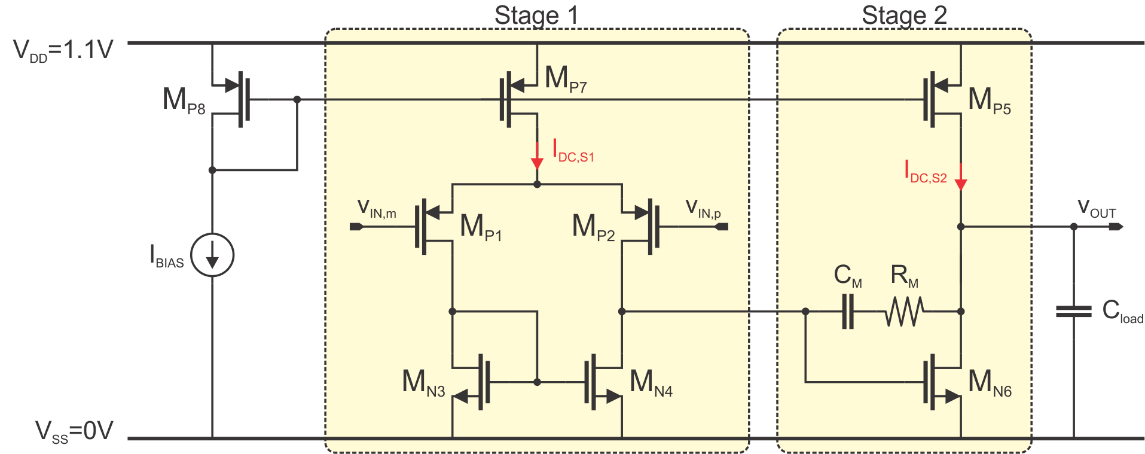
We can then isolate the miller capacitance which gives us :

= =

We need to compute the miller resistance :

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

|  |  |
| --- | --- |
|  | Mn3.vgs - Mp1.vdsat + Mp1.vgs = 0.42 + 0.145 - 0.43 = 0.14 V |
|  | Mp7.vdsat + Mp1.vgs +spec.VDD = -0.2 -0.43 + 1.1 = 0.47 V |
|  | Mn6.vdsat = 0.145 V |
|  | spec.VDD + Mp5.vdsat = 1.1 - 0.2 = 0.9V |
|  | spec.VDD\*(Mp7.ids + Mp8.ids + Mn6.ids) |



0.68 V

40.88 µA

46.5

0.42 v

0.55 V

0.42 V

40 µA

0.2 V

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

### 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*

### 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]** | **Ids**  **[μA]** | **VOV**  **[V]** | **gm**  **[S]** | **gds**  **[S]** | **gm/gds**  **[-]** | **Vds,sat**  **[V]** | **Vds**  **[V]** |
| Mp1 | 1.53 | 150 | 20.40 | -0.00135 | 202,23e-6 | 9.66e-6 | 20.91 | -145.04e-3 | -508.53e-3 |
| Mp2 | 1.53 | 150 | 20.40 | -0.00135 | 202.23e-6 | 9.66e-6 | 20.91 | -145.04e-3 | -508.53e-3 |
| Mn3 | 1.52 | 1000 | 20.40 | 0.0032698 | 163e-6 | 7.05e-6 |  | 208.14e-3 | 413.55e-3 |
| Mn4 | 1.52 | 1000 | 20.40 | 0.0032698 | 163e-6 | 7.05e-6 |  | 208.14e-3 | 413.55e-3 |
| Mp5 | 14.98 | 1000 | 46.50 | -0.00200 |  | 9.94e-6 |  | -175.19e-3 | 550e-3 |
| Mn6 | 1.25 | 125 | 46.50 | 0.00145 | 554e-6 | 20.40e-6 | 27.15 | 118.41e-3 | 550.00e-3 |
| Mp7 | 15.69 | 1000 | 40.80 | -0.0020345 |  | 62.21e-6 |  | -177.91e-3 | -200e-3 |
| Mp8 | 13.41 | 1000 | 40.00 | -0.0020151 |  | 12.13e-6 |  | -176.38e-6 | -418e-3 |

|  |  |  |
| --- | --- | --- |
| **Device** | **Units** | **Value** |
| CM | pF | 0.71 |
| RM | Ω | 6891 |
| IBIAS | A | 4\*10^-6 |

### 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:

1. Frequency Response in MATLAB and LTspice
2. Noise contribution in LTspice
3. 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.

1. 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).

### Simulate (small voltage gain) with and in MATLAB and LTspice. Then, paste both -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:**

Frequency response Opamp in Matlab

Chart

Description automatically generated

Frequency response Opamp in LTspice

Graphical user interface

Description automatically generated

|  |  |
| --- | --- |
| **Simulator** | **PM (deg)** |
| MATLAB | 88.3 |
| LTspice | 90.5 |

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

It can be seen that the DC gain is equal to 47dB in both Matlab and LTspice. However, the phase margin and GBW are a bit different between the two even though from the matlab perspective, it respects the specification. From the LTspice perspective, the GBW is a bit less than excpected. It can be seen from the DC parameter analysis that some DC values are not the same, close but still different. And a little change between Vov's or Vth's can have a huge impact on the frequency analysis since those are key parameters.

To still have a decent GBW and PM in LTspice, we overshoot the GBW in matlab such that the Cm and RM changed and have a bigger GBW .

### Simulate in LTspice for the following cases:

### No compensation network ().

### With compensation capacitor but no compensation resistor ().

### 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:**

No compensation network:

Graphical user interface

Description automatically generated

No compensation resistor:

Graphical user interface

Description automatically generated With both the compensation:

Graphical user interface

Description automatically generated

|  |  |
| --- | --- |
| **Case** | **PM [deg]** |
| No compensation network | 46 |
| No compensation resistor | 41 |
| With both the compensation | 90.5 |

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

Without compensation network, one can notice that the GBW is way bigger than with a compensation network. While the phase margin is way lower.

To handle that, the dominant pole has been moved to the left by adding the miller capacitance such that the GBW has been reduced.

To increase the phase margin, the miller resistor has been put after the Cm. The miller resistance moves the zero to the left such that it compensates the pole 3. Both of the zero and the pole were inducing the fall of the phase. This compensation can be seen on the plot with the PM not falling at high frequency.

1. Noise contribution in LTspice

### Simulate the output-referred noise voltage power density over an appropriate frequency range in LTspice. Then, insert the -plot below.

**Plots:**

Graphical user interface, text

Description automatically generated

### Explain, analyse and interpret the results in “10)”.

We can see on the two plots above that the output noise voltage is decreasing with frequency. Indeed, the output signal noise can be expressed as : =

Knowing that can be approximated to : We can conclude by this relation and the plots than the noise is inversely proportional to the frequency and it becomes constant at a certain frequency. The first part is call a flicker noise, the noise falls by a decade when a frequency decrease by a decade. The constant noise is a white noise that becomes stronger than the flicker noise.

### 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 kHz to . Furthermore, copy and paste the top 9 contributing elements using "View->SPICE Error Log" option and insert that list below.

Input-referred noise spectral density:

Graphical user interface

Description automatically generated

out\_totnvrms: INTEG(v(onoise))=0.015487 FROM 1000 TO 1e+009

out\_mp1nvrms: INTEG(v(mp1))=0.0107594 FROM 1000 TO 1e+009

out\_mp2nvrms: INTEG(v(mp2))=0.0109016 FROM 1000 TO 1e+009

out\_mp7nvrms: INTEG(v(mp7))=0.00010321 FROM 1000 TO 1e+009

out\_mp8nvrms: INTEG(v(mp8))=2.74387e-005 FROM 1000 TO 1e+009

out\_mp5nvrms: INTEG(v(mp5))=9.46637e-005 FROM 1000 TO 1e+009

out\_mn6nvrms: INTEG(v(mn6))=0.000349515 FROM 1000 TO 1e+009

out\_mn4nvrms: INTEG(v(mn4))=0.00167099 FROM 1000 TO 1e+009

out\_mn3nvrms: INTEG(v(mn3))=0.00151606 FROM 1000 TO 1e+009

out\_rmnvrms: INTEG(v(rm))=8.21082e-005 FROM 1000 TO 1e+009

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

The flicker noise can be observed like in the ouptout plot but is niot affected by the gain since it is the input. The gain attenuation has thus no impact on the noise.

The biggest noise output voltages are the ones from the active load pair, Mp1, Mp2, Mn3 and Mn4. Since these noise voltages are amplify with the gain of the stage 2 and the stage 1 aswell and the transistor 5 and 6 are only amplify by the gain of the stage 1, it is normal to have bigger noise voltages from the transistor of active load pair. Indeed, from this spice err log, we can see that Mp1 and Mp2 create the most noise output voltage by a factor of 10 at minimum and a bigger noise than the Mn3 and Mn4 . It can be explained by the fact that the flicker noise is proportional to and the Mp1 and Mp2 are the transistor that have the smallest length in the active load pair.

1. Linearity in LTspice

### 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:**

Graphical user interface, chart

Description automatically generated with medium confidence

Chart, histogram

Description automatically generated

Chart, line chart

Description automatically generated

Chart

Description automatically generated

### Explain, analyse and interpret the results in “14)”.

the 47 dB gain is achieved for low amplitudes, but it starts falling for higher input voltages.

We can see that the 47 dB gain is achieved for low amplitudes of Vin (After 0.5e-6), but it starts falling for higher input voltages.

There are linearities around the operating point and if the small signal is large enough, we leave the operating point. The non-linearity is negligible.

At higher amplitudes it is compressed. The linearity remains acceptable below 1uV (1-dB compression point).

# Conclusion (10 points):

### 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 | 223.9 | 223.9 | 221 | 216 |
| DC gain | dB | 47 | 47 | 46.9 | 46.77 |
| Gain-Bandwidth frequency | MHz | 43 | 43 | 43 | 42 |
| Dominant pole frequency | kHz | 191 | 191 | 194 | 216 |
| PM | ° | > 70 |  | 88.3 | 90.5 |
|  | V |  | 0.33 | 0.363 | 0.375 |
|  | V | > 0.7 | 0.755 | 0.807 | 0.8030 |
|  | mW |  | 0.14 | 0.14 | 0.14 |
| (output-referred noise) | Vrms |  |  |  |  |

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

One can see that the values on the table above differ a little between the handcalculations and matlab, and between matlab and LTspice.

Between the HC and Matlab, the mismatchs can be explained by the fact that we have made some approximations, for example, we only used graphs for a W/L = 10 but this ratio sometimes changes in Matlab.

The mismatchs between matlab and LTspice are caused by the own calculation and database of LTspice. Some DC parameters like Vth and Vov are close but not the exactly the same and the impact of those two parameters are huge.

### 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.

/

# 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

%% EX: Specs

spec.VDD = 1.1;

spec.fGBW = 43e6; % [Hz] GBW frequency %Overshoot frequency for LTspice

spec.GBW = spec.fGBW \* 2 \* pi; % radians/s

spec.gain =10^(47/20); % [] voltage gain

spec.gaindB = 47; %

spec.CL = 2e-12; % [F], Load cap

spec.RL = 0; % [Ohm]

Vout = 0.55;

Mp7.vdsat = -0.2;

Mp5.vdsat = -0.2;

Mp8.vdsat = -0.2;

Mp5.ids = 46.5\*1e-6;

Mp8.ids = 40\*1e-6;

Mp7.ids = 40.8\*1e-6;

%% Mp5: Second stage NMOS: Design choices and implementation

Mp5.lg = 1000e-9;

Mp5.vsb = 0;

Mp5.vov = -0.2;

Mp5.vds = Vout - spec.VDD;

Mp5.vth = tableValueWref('vth', PRVT, Mp5.lg, 0, Mp5.vds, Mp5.vsb);

Mp5.vgs = Mp5.vov + Mp5.vth;

Mp5.w = mosWidth('ids', Mp5.ids, Mp5);

Mp5 = mosNfingers(Mp5);

Mp5 = mosOpValues(Mp5);

%% Mn6: Second stage NMOS: Design choices and implementation

Mn6.vov = 0.145;

Mn6.lg = 125e-9; % [m], channel length

Mn6.vsb = 0;

Mn6.vds = Vout;

Mn6.vth = tableValueWref('vth', NRVT, Mn6.lg, 0, Mn6.vds, Mn6.vsb);

%Mn6.vgs = 0.42;

Mn6.vgs = Mn6.vov + Mn6.vth ;

Mn6.ids = Mp5.ids;

Mn6.w = mosWidth('ids', Mn6.ids, Mn6);

Mn6 = mosNfingers(Mn6);

Mn6 = mosOpValues(Mn6);

%% Mn7: Second stage NMOS: Design choices and implementation

Mp7.lg = 1e-6;

Mp7.vsb = 0; % triple-well-technology

Mp7.vov = -0.2;

Mp7.vds = Mp7.vov;

Mp7.vgs = Mp5.vgs;

Mp7.vth = tableValueWref('vth',PRVT,Mp7.lg, 0,Mp7.vds,Mp7.vsb);

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

Mp7 = mosNfingers(Mp7);

Mp7 = mosOpValues(Mp7);

%% Mp8: Second stage NMOS: Design choices and implementation

Mp8.lg = 1e-6;

Mp8.vov = -0.2;

Mp8.vsb = 0; % triple-well-technology

Mp8.vgs = Mp7.vgs;

Mp8.vds = Mp7.vgs;

Mp8.vth = tableValueWref('vth',PRVT,Mp8.lg, 0,Mp8.vds,Mp8.vsb);

Mp8.vov = -Mp8.vth +Mp8.vgs ;

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

Mp8 = mosNfingers(Mp8);

Mp8 = mosOpValues(Mp8);

%% Mn1/2: Second stage NMOS: Design choices and implementation

Mp2.lg = 150e-9; %changer de 65 à 100

Mp2.vov = -0.135;

Mp2.vsb = 0; % triple-well-technology

Mp2.vds = Mn6.vgs - (Mp7.vdsat) - spec.VDD;

Mp2.vth = tableValueWref('vth', PRVT,Mp2.lg, 0, Mp2.vds,Mp2.vsb);

Mp2.vgs = Mp2.vov + Mp2.vth;

Mp2.ids = 0.5\*Mp7.ids;

Mp2.w = mosWidth('ids',Mp2.ids, Mp2);

Mp2 = mosNfingers(Mp2);

Mp2 = mosOpValues(Mp2);

Mp1 = cirElementCopy(Mp2, Mp1);

%% Mn4/3: Second stage NMOS: Design choices and implementation

Mn4.lg = 1000e-9; % [m], channel length CHANGER LG DE MN4 POUR AUGMENETER FDANS LE GRAPHE

Mn4.vsb = 0.0;

Mn4.ids = Mp2.ids;

Mn4.vds = Mn6.vgs;

Mn4.vgs = Mn6.vgs;

Mn4.vth = tableValueWref('vth', NRVT, Mn4.lg, 0, Mn4.vds, Mn4.vsb);

Mn4.vov = Mn4.vgs - Mn4.vth ;

Mn4.w = mosWidth('ids', Mn4.ids, Mn4);

Mn4 = mosNfingers(Mn4);

Mn4 = mosOpValues(Mn4);

Mn3 = cirElementCopy(Mn4, Mn3);

% %% 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

%

spec.Cm = 0.709e-12; % [F] miller cap %From the handcalculation

AvDC1 = Mp1.gm / (Mp1.gds + Mn3.gds); % DC gain 1st stage

AvDC2 = Mn6.gm / (Mn6.gds + Mp5.gds); % DC gain 2nd stage

AvTot = AvDC1 \* AvDC2; % Total DC gain

C1 = Mp1.cgd + Mn3.cgs + Mn4.cgs; % Capacitance on node 1

G1 = Mn3.gm; % Admittance on node 1

C2 = spec.Cm\*(1 + Mn6.gm/(Mn6.gds + Mp5.gds)); % Capacitance on node 2

G2 = Mp2.gds + Mn4.gds; % Admittance on node 2

C3 = spec.CL; % Capacitance on node 3

G3 = Mn6.gm; % Admittance on node 3

% %% AI: Set-up Rm, Cc and CL and calculate the zero required for the transfer-fct

%

spec.Cm = (Mp1.gds + Mn3.gds)\*AvTot/(2\*pi\*spec.fGBW \* (1 + Mn6.gm / (Mn6.gds + Mp5.gds)));

spec.Rm = (1/Mn6.gm)\*(1+spec.CL/spec.Cm);

z1 = 1/((1/Mn6.gm - spec.Rm)\*(spec.Cm + Mn6.cgd));

C4 = spec.CL; % Capacitance on node 4

G4 = spec.Rm; % Admittance on node 4 (hint: what happens with CL at very high

%

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

Vin\_cm\_min = Mn3.vgs - Mp1.vdsat + Mp1.vgs;

Vin\_cm\_max = spec.VDD + Mp7.vdsat + Mp1.vgs;

Vout\_cm\_min = Mn6.vdsat;

Vout\_cm\_max = spec.VDD + Mp5.vdsat;

Pdiss = spec.VDD \* (Mp5.ids + Mp7.ids + Mp8.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

% transistors in the circuit file

fprintf('IBIAS\t= %6.2fmA\nRm\t= %6.2f Ohm\nCm\t= %6.2fpF\n\n',Mp8.ids/1e-3,spec.Rm,spec.Cm/1e-12);

%% 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)

bode(TF1,2\*pi\*freq); grid on;

h = gcr;

setoptions(h,'FreqUnits','Hz');

title('Frequency response Opamp');

hold all

Insert your LTspice netlist here:

vdd N001 0 1.1

vcm N006 0 0.478

E1 vinn N006 N006 vinp 1

V1 vinp N006 SINE(0 {amp} 10k) AC 10m

Mp8 N002 N002 N001 N001 p\_11\_sprvt l=1u w=13.41u

Mp7 N003 N002 N001 N001 p\_11\_sprvt l=1u w=15.69u

Mp1 N005 vinn N003 N003 p\_11\_sprvt l=150n w=1.53u

Mp2 vinter vinp N003 N003 p\_11\_sprvt l=150n w=1.53u

Mn4 vinter N005 0 0 n\_11\_sprvt l=1u w=1.52u

Mn3 N005 N005 0 0 n\_11\_sprvt l=1u w=1.52u

Mn6 vout vinter 0 0 n\_11\_sprvt l=125n w=1.25u

Mp5 vout N002 N001 N001 p\_11\_sprvt w=14.98u l=1u

Rm vout N004 7365

Cm N004 vinter 0.66p

Ibias N002 0 40µ

Cload vout 0 2p

.model NMOS NMOS

.model PMOS PMOS

.lib C:\Users\leand\OneDrive\Documents\LTspiceXVII\lib\cmp\standard.mos

.include BSIM4\_UMC65.lib

.op

.tran 10m

.step dec param amp 1u 100m 3

.param amp=1n

.backanno

.end