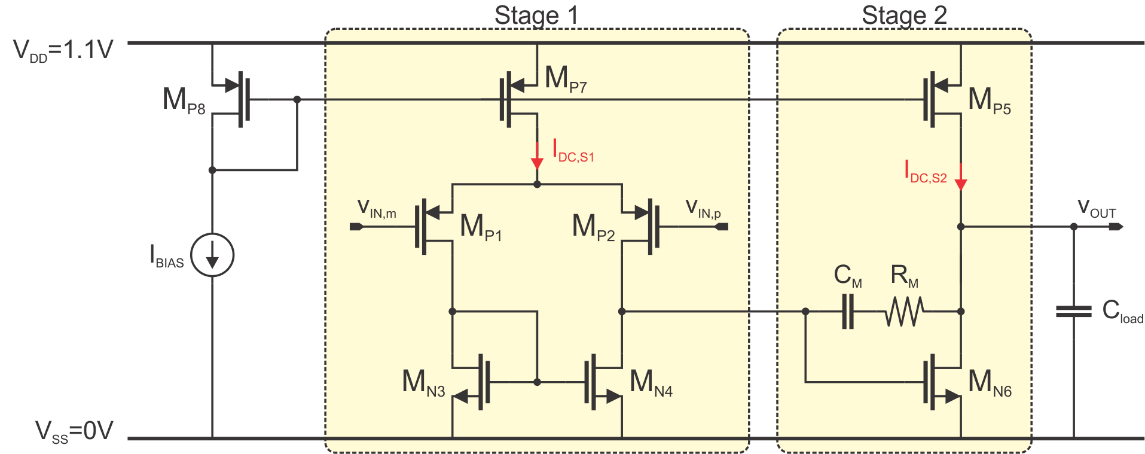
**Analog Electronic Circuits – 202**2**-202**3

**Design Project Report**

# Group data

|  |  |
| --- | --- |
| Group number | 21 |
| Name – Student 1 | Rogier De Nys |
| Name – Student 2 | Warre De Winne |

# Goal:



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

|  |  |
| --- | --- |
|  | 5 |
| DC gain [dB] | 47 dB => 223.87 |
| [MHz] | 17 |
| 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:

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

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

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

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

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

|  |
| --- |
|  |
|  |

Handcalculations:

We have Av0 = Av1 \* Av2, where Av1 ≈ gm1/go1 and Av2 ≈ gm6/go6 (if we assume go3 << go1 and go5 << go6).

We can calculate gm1 directly from the GBW and Av0 (since the dominant pole is at node 2, the output of the first stage): fGBW = gm1/(2\*π\*Cm) ⬄ gm1 = 133.5 µS.

To ensure a large enough phase margin, we take fp4 = 3\*fGBW; since p4 ≈ -gm6/Cl, gm6 = 3\*fGBW\*Cl\*2\*π = 1.60 mS.

Using the gm vs. VOV curve for nmos, we decide VOV6 = 0 V, L6 = 80 nm and W6 = 100\*L6 = 8 µm. This gives gm6 = 1.65 mS (what we want), go6 = 100 µS, IDS6 = 96.5 µA. This means Av2 = gm6/go6 = 16.5. VGS6 = 0.36 V (= VDS4 = VDS3 = VGS3.)

From this, it follows that Av1 = Av0/Av2 = 13.6 = gm1/go1. Since gm1 = 133.5 µS, go1 must be 9.82 µS. We now need to find values for L1, W1, VOV1 and VDS1 such that these conditions are met. This is the case for L1 = 200 nm, W1 = 12\*L1 = 2.4 µm, VOV1 = -0.078 V, VDS1 = -0.2375 V. Then we have go1 = 9.4 µS, gm1 = 130.5 µS and gm1/go1 = 13.9. IDS1 = 9.7 µA.

We still need to size Mn3. IDS3 = IDS1 = 9.7 µA, VDS3 = VGS3 = 0.36 V, and we want go3 << go1 (= 9.4 µS).

In practice, it wasn't possible to find a W3 & L3 such that go3 << go1, and IDS3 = 9.7 µS.

For this reason, we retry finding Mn1, but this time without assuming go3 << go1, so Av1 = gm1/(go1+go3), or go1+go3 = 9.82 µS.

When we take L1 = 500 nm, W1 = 25\*L1 = 12.5 µm, VOV1 = -0.021 V, VDS1 = -0.3 V, we have gm1 = 134.9 µS and go1 = 3.97 µS and IDS1 = 6.86 µA, while VGS1 = -0.26 V. Now we need to find the dimensions for Mn3 so that IDS3 = 6.86 µA and go3 = 9.82 µS - 3.97 µS = 5.85 µS. VDS3 = VGS3 = 0.36 V still.

This turns out to be L3 = 106 nm, W3 = 4.5\*L3 = 477 nm. Then we have go3 = 5.86 µS, IDS3 = 6.86 µA.

Because of symmetry, these are also the dimensions and characteristics of Mp2 and Mn4.

The only transistors left to design are Mn8, Mn7 and Mn5. These act as a current sources, so they need a low gds. For this reason we choose L8 = L7 = L5 = 2 µm. The widths are defined by the currents flowing through them.

For Mp5, we have IDS5 = IDS6 = 97 µA and VDS5 = -(1.1 V - 0.55 V) = -0.55 V, while for Mp7, IDS7 = IDS1 + IDS2 = 2\*IDS1 = 13.72 µA and VDS7 = -(1.1 V - 0.36 V - 0.3 V) = -0.44 V. We need to find values for W5, W7, W8 and Ibias such that VGS5 = VGS7 = VGS8 = VDS8.

As it turns out, it isn't possible to find a transistor Mp5 that allows a current of 97 µA, while keeping gds5 << gds6 = 100 µS. For now, we set W5 = 45 µm, VOV5 = -0.246 or VGS5 = -0.444 V, which gives IDS5 = 97 µA and gds5 = 17 µS. This was the lowest gds we could find.

We use this value of VGS5 to calculate W7, knowing that VGS7 = VGS5 = -0.444 V, IDS7 = 13.72 µA, VDS7 = -0.44 V and L7 = 2 µm. This gives W7 = 6.5 µm. gds7 is then 3.3 µS (which is not ideal, since it is comparable in value to gds1 and gds3) and VOV7 = -0.25 V.

The final values of W8 and Ibias we can calculate using VDS8 = VGS8 = -0.444V. We have W8 = 9.19 µm and Ibias = 19.4 µA (and VOV8 = -0.25 V). Since these last transistors act as a current mirror, we should have IDS7 = Ibias\*W7/W8 (this is the case), and IDS5 = Ibias\*W5/W8. The last value is 95 µA, which is close enough to IDS5 = 96.5 µA. The difference is probably due to the large value of gds5 and the difference between VDS5 and VDS8.

We can try and find new values for W6 and L6 such that VGS6 is still 0.36 V, gm6 = 1.60 mS and go6 = 100 µS - 17 µS = 83 µS. This way, we can make sure that Av2 = gm6/(go5+go6) will still be exactly 16.5, and we don't need to update the values of the other transistors. IDS6 will probably change, but we'll then need to find an appropriate Mp5 so that IDS5 = IDS6 and go5 = 100 µS - go6.

Luckily, for L6 = 90 nm and W6 = 80\*L6 = 7.2 µm (and VGS6 = 0.36 V, VDS6 = 0.55 V), we have gm6 = 1.62 mS, IDS6 = 96.5 µA and go6 = 85 µS. This matches the requirements almost perfectly, and IDS6 stays the same, so we won't even have to adapt Mp5.

------

Second stage:

Av2 = 22.4, gm6 = 157 µS (fp4 ≈ 20 MHz) ------> go5+go6 = 7.0 µS

L6 = 1 µm, VDS6 = 0.55 V, VGS6 = 0.2735 V ------>

W6 = 3.6 µm; IDS6 = 10.7 µA, gm6 = 158 µS, go6 = 4.15 µS (-> go5 = 2.85 µS), VOV6 = 0.19 V

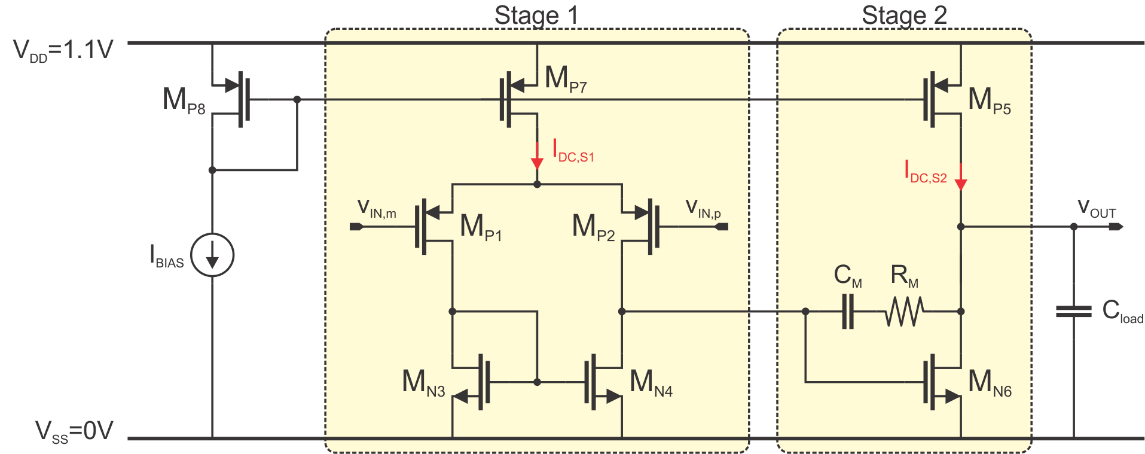
Mp5:

VDS5 = -0.55 V, VGS5 = -0.54 V, L5 = 1 µm, IDS5 = IDS6 = 10.7 µA; VOV5 preferably > -0.2V

-> W5 = 1.5 µm gives IDS5 = 10.5 µA, gds5 = 2.36 µS, VOV5 = -0.32 V

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

|  |  |
| --- | --- |
|  | Mp2.vth + Mn4.vth + Mn4.vov = 184 mV |
|  | Mp7.vov + Mp2.vov + VDD = 783 mV |
|  | Mn6.vdsat = 61 mV |
|  | VDD - abs(Mp5.vdsat) = 896 mV |
|  | VDD\*(Mp8.ids + Mp7.ids + Mp5.ids) = 0.1 mW |



550 mV

2.18 uA

46 uA

400 mV

0.550

400 mV

20 uA

700 mV

# 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 | 10.08 | 1e3 | 1.0 | 0.31355 | 9.28e-6 | 388e-9 |  | 0.19677 | 0.4 |
| Mp2 | 10.08 | 1e3 | 1.09 | 0.31355 | 9.28e-6 | 388e-9 |  | 0.19677 | 0.4 |
| Mn3 | 91.5e-3 | 1e3 | 1.09 | 0.31355 | 9.28 |  |  |  |  |
| Mn4 |  |  |  |  |  |  |  |  |  |
| Mp5 |  |  |  |  |  |  |  |  |  |
| Mn6 |  |  |  |  |  |  |  |  |  |
| Mp7 |  |  |  |  |  |  |  |  |  |
| Mp8 |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Device** | **Units** | **Value** |
| CM | pF |  |
| RM | Ω |  |
| IBIAS | A |  |

### 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:**

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

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

### 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:**

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

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

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:**

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

### 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.

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

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:**

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

# 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 |  |  |  |  |
| DC gain | dB |  |  |  |  |
| Gain-Bandwidth frequency | MHz |  |  |  |  |
| Dominant pole frequency | kHz |  |  |  |  |
| PM | ° | > 70 |  |  |  |
|  | V |  |  |  |  |
|  | V | > 0.7 |  |  |  |
|  | mW |  |  |  |  |
| (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.

### 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:

Insert your LTspice netlist here: