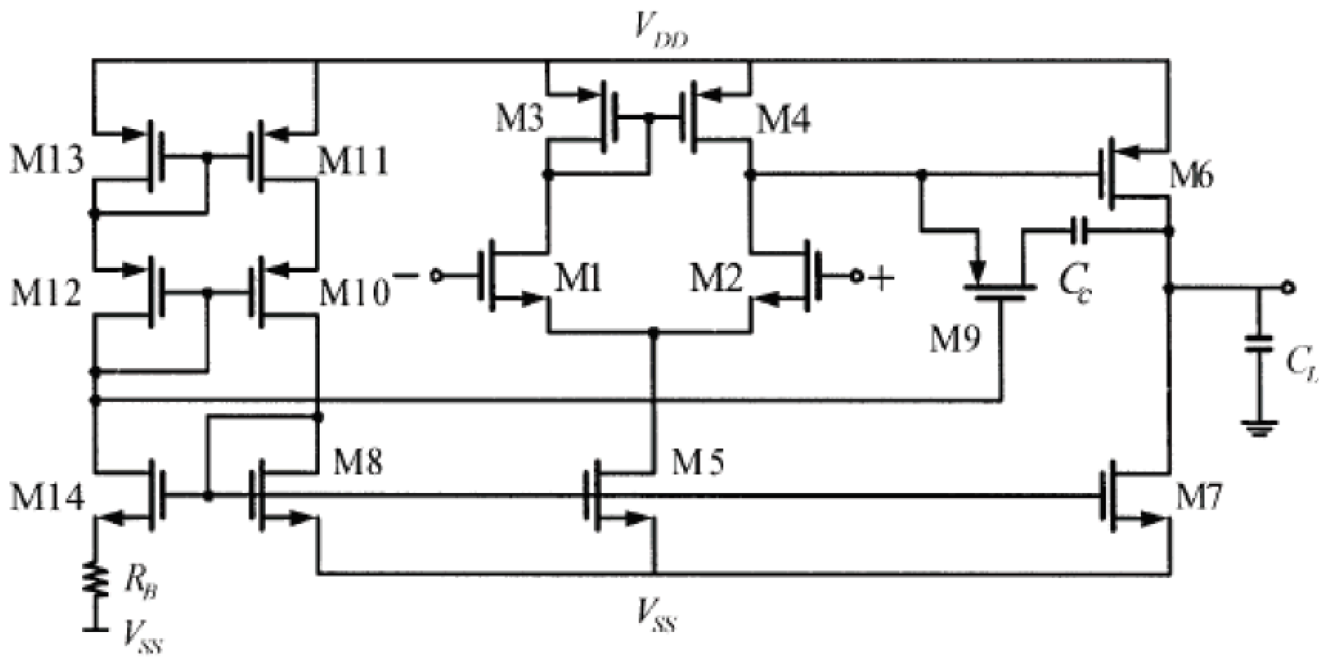
In [1]:
```
# Requirements - uncomment the following line to install required packages
# %pip install -q numpy pandas openpyxl
```

In [2]:
```
import numpy as np
import pandas as pd
import utils.functions_proj1 as fp1
```

Project 1: 2-Stage OpAmp



Design Phase

Project Specifications

In [3]:
```
VDD = 2.5                   # supply voltage
VSS = -2.5                  # voltage at the negative supply
SR = 5e6                    # slew rate +5/-5 V/µs, converted in V/s
CL = 5e-12                  # load capacitance in F
MAX_VIN = 2.1               # max common mode input voltage
MIN_VIN = -1.3              # min common mode input voltage
MAX_VOUT = 2.2              # max output voltage
MIN_VOUT = -2.2             # min output voltage
GBW = 5e6                   # gain bandwith of 5 MHz expressed in Hz
gbw_rad = GBW * 2 * np.pi   # gain bandwith in rad/s
DIFF_GAIN = 80              # differential gain in dB, has to be > 80 dB
PM = 60                     # phase margin in degrees, has to be > 60 degrees
NOISE = 30                  # sqrt(thermal noise) in nV/sqrt(Hz), has to be < 30 nV/sqrt(Hz)
```

Technological Parameters

nMOS

In [4]:
```
vto_nmos = 0.71         # Threshold voltage (V)
phi_nmos = 0.6          # Surface potential (V)
gamma_nmos = 0.01       # Body effect coefficient (V^0.5)
kp_nmos = 182e-6        # Transconductance parameter (A/V^2), beta prime parameter
lambda_nmos = 0.01      # Channel-length modulation parameter
tox_nmos = 9.6e-9       # Oxide thickness (m)
cj_nmos = 350e-6        # Junction capacitance per unit area (F/m^2)
cjsw_nmos = 120e-12     # Sidewall junction capacitance per unit length (F/m)
pb_nmos = 0.8           # Built-in potential (V)
mj_nmos = 0.33          # Junction grading coefficient
mjsw_nmos = 0.33        # Sidewall junction grading coefficient
cgdo_nmos = 0.046e-9    # Gate-drain overlap capacitance per unit width (F/m)
cgso_nmos = 0.046e-9    # Gate-source overlap capacitance per unit width (F/m)
```

pMOS

In [5]:
```python
vto_pmos = -0.901          # Threshold voltage (V)
phi_pmos = 0.6             # Surface potential (V)
lambda_pmos = 0.01         # Channel-length modulation parameter
tox_pmos = 9.0e-9          # Oxide thickness (m)
cj_pmos = 350e-6           # Junction capacitance per unit area (F/m^2)
pb_pmos = 0.8              # Built-in potential (V)
cgso_pmos = 0.046e-9       # Gate-source overlap capacitance per unit width (F/m)
gamma_pmos = 0.01          # Body effect coefficient (V^0.5)
kp_pmos = 41.5e-6          # Transconductance parameter (A/V^2), beta prime parameter
cjsw_pmos = 120e-12        # Sidewall junction capacitance per unit length (F/m)
mjsw_pmos = 0.33           # Sidewall junction grading coefficient
cgdo_pmos = 0.046e-9       # Gate-drain overlap capacitance per unit width (F/m)
```

Other technological parameters

In [6]:
```python
epsilon_ox = 3.45e-11                # Oxide permittivity (F/m)
Cox_pmos = epsilon_ox / tox_pmos     # Capacitance per unit area (F/m^2) of pmos
Cox_nmos = epsilon_ox / tox_nmos     # Capacitance per unit area (F/m^2) of nmos
mu_p = kp_pmos / Cox_pmos            # Mobility of holes (m^2/V·s) of pmos
mu_n = kp_nmos / Cox_nmos            # Mobility of electrons (m^2/V·s) of nmos
T_celsius = 25                       # Temperature in Celsius
T_kelvin = T_celsius + 273.15        # Temperature in Kelvin
```

Phase 1: Transistor Sizing

1. Calculate minimum Cc

We need to find the *minimum* **compensation capacitance** value in order to respect the given **noise** specification.

$$noise = \sqrt{\frac{v_n^2}{\Delta f}} \leq 30\frac{nV}{\sqrt{Hz}} \qquad V_{OV_3} = V_{DD} - V_{IN_{MAX}} - |V_{Tp}| + V_{Tn}$$

$$C_{C_{min}} = \frac{16}{3} \cdot \frac{K \cdot T}{GBW_{rad} \cdot \frac{v_n^2}{\Delta f}} \cdot \left(1 + \frac{SR}{GBW_{rad} \cdot V_{OV_3}}\right)$$

In [7]:
```
Vov3 = VDD - MAX_VIN - abs(vto_pmos) + vto_nmos  # overdrive voltage of M3
Cc_min = ((16 * kp_pmos * T_kelvin) / (3 * gbw_rad * NOISE**2)) * (1 + (SR / (gbw_rad * Vov3)))  # compensation capacitance

print(f"Vov3 = {Vov3:.3f} V\nCc = {Cc_min*1e12:.3f} pF")
```

```
Vov3 = 0.209 V
Cc = 4.111 pF
```

2. Current through M5, M6, M7

From the **slew rate** specific we can derive I_5 and $I_7 = I_6$:

$$C_{C_{charging}} \quad \Rightarrow \quad SR^+ = \frac{I_5}{C_C}$$

$$C_{L_{discharging}} \quad \Rightarrow \quad SR^- = \frac{I_7 - I_5}{C_L}$$

Imposing **charging = discharging**:

$$\frac{I_5}{C_C} = \frac{I_7 - I_5}{C_L} = SR \quad \Rightarrow \quad \begin{cases} I_5 = C_C \cdot SR \\ I_7 = I_6 = SR \cdot (C_C + C_L) \end{cases}$$

In [8]:
```
i5 = Cc_min * SR
i7 = i6 = SR * (Cc_min + CL)

print(f"i5 = {i5:.3e} A\ni6 = {i6:.3e} A\ni7 = {i7:.3e} A")
```

```
i5 = 2.056e-05 A
i6 = 4.556e-05 A
i7 = 4.556e-05 A
```

3. Length of M6

In the AC behaviour we have *three poles* and a *zero*:

$$\begin{cases} p_1 = -\frac{1}{C_{Miller}R_{out1}} = -\frac{1}{C_c g_{m6} R_{out2} R_{out1}} \\ p_2 = -\frac{g_{m6}}{C_L} \\ p_3 = -\frac{g_{m6}}{C_{GS6}} \cdot \frac{C_c}{C_c + C_L} \\ z = \left(1 + \frac{1}{g_{m6}R_c}\right) \cdot C_c \end{cases}$$

Knowing that:

$$C_{GS_6} = \frac{2}{3} C_{OX} \mu_p W_6 L_6 \qquad\qquad V_{OV_6} = V_{DD} - V_{OUT_{MAX}}$$

We can derive the length of the transistor M6 as follows:

$$L_6 = \sqrt{\frac{3}{2} \cdot \frac{\mu_p \cdot V_{OV_6} \cdot C_C}{GBW_{rad} \cdot (C_C + C_L) \cdot \tan(PM)}}$$

```
In [9]: Vov6 = VDD - MAX_VOUT   # overdrive voltage of M6
        L6 = np.sqrt((3/2) * ((mu_p * Vov6 * Cc_min) / (gbw_rad * (Cc_min+CL) * np.tan(np.radians(PM)))))  # max length of M6

        print(f"Vov6 = {Vov6:.3f} V\nL6 = {L6*1e6:.3f} µm")
```

```
Vov6 = 0.300 V
L6 = 6.356 µm
```

4. Form factor of M6

From the I_{DS} equation for a transistor in *saturation region* (such as M6), and knowing I_6, we can calculate the form factor of M6:

$$I_6 \simeq \frac{\beta'_p}{2} \cdot S_6 \cdot V_{OV_6}^2 \quad \Rightarrow \quad S_6 = \frac{2 \cdot I_6}{\beta'_p \cdot V_{OV_6}^2}$$

```
In [10]: S6 = 2 * i6 / (kp_pmos * Vov6**2)   # aspect ratio of M6
         W6 = S6 * L6                         # width of M6
```

```
print(f"W6 = {W6*1e6:.3f} µm\nS6 = {S6:.3f}")
```

```
W6 = 155.050 µm
S6 = 24.394
```

5. Currents I1, I2, I3, I4

M1, M2, M3, and M4 form a **differential pair** with an active load, and due to the **balancing** conditions we have:

$$I_1 = I_2 = I_3 = I_4 = \frac{I_5}{2}$$

In [11]:
```
i1 = i2 = i3 = i4 = i5/2    # current through M1, M2, M3, M4

print(f"i1 = i2 = i3 = i4 = {i1:.3e} A")
```

```
i1 = i2 = i3 = i4 = 1.028e-05 A
```

6. Form factor of M1, M2

Exploiting the **gain bandwidth** specific we can derive V_{OV_1}, needed to obtain $S_1 = S_2$:

$$\begin{cases} GBW = A_{d_0} \cdot |p_1| = \frac{g_{m_1}}{C_C} \\ g_{m_1} = \sqrt{2\beta_n' S_1 I_1} = \frac{2I_1}{V_{OV_1}} \\ I_1 = I_2 = \frac{I_5}{2} = \frac{SR \cdot C_C}{2} \end{cases} \Rightarrow \quad V_{OV_1} = \frac{SR}{GBW} \quad \Rightarrow \quad S_1 = S_2 = \frac{2 \cdot I_1}{\beta_n' \cdot V_{OV_1}^2}$$

In [12]:
```
Vov1 = SR / gbw_rad                     # overdrive voltage of M1
S1 = (2 * i1) / (kp_nmos * Vov1**2)     # aspect ratio of M1
S2 = S1                                 # aspect ratio of M2

print(f"Vov1 = {Vov1:.3f} V\nS1 = {S1:.3f}")
```

```
Vov1 = 0.159 V
S1 = 4.459
```

7. Form factor of M5

Looking at the circuit we can derive the formula for V_{OV_5}, needed to calculate the form factor of M5:

$$V_{OV_5} = V_{IN_{CM_{min}}} - V_{SS} - V_{OV_1} - V_{Tn} \quad \Rightarrow \quad S_5 = \frac{2 \cdot I_5}{\beta'_n \cdot V_{OV_5}^2}$$

```
In [13]: Vov5 = MIN_VIN - VSS - Vov1 - vto_nmos  # overdrive voltage of M5
         S5 = (2 * i5) / (kp_nmos * Vov5**2)     # aspect ratio of M5

         print(f"Vov5 = {Vov5:.3f} V\nS5 = {S5:.3f}")
```

```
Vov5 = 0.331 V
S5 = 2.064
```

8. Form factor of M7

Since M5 and M7 share the *same overdrive voltage*:

$$V_{OV_5} = V_{OV_7} \quad \Rightarrow \quad \frac{I_7}{I_5} = \frac{S_7}{S_5} \quad \Rightarrow \quad S_7 = \frac{C_C + C_L}{C_C} \cdot S_5$$

```
In [14]: Vov7 = Vov5                         # overdrive voltage of M7
         S7 = (Cc_min + CL) / Cc_min * S5    # aspect ratio of M7

         print(f"Vov7 = {Vov7:.3f} V\nS7 = {S7:.3f}")
```

```
Vov7 = 0.331 V
S7 = 4.574
```

9. Form factor of M3, M4

M3 build together with M4 a ***current mirror***. The ICMR analysis needs the OPAMP to be balanced, which means that the current flowing in one branch is equal to the current flowing in the other.

In order to achieve **perfect balancing** we need:

$$\begin{cases} V_{OV_4} = V_{OV_3} \\ I_4 = I_3 \end{cases} \quad \Rightarrow \quad S_4 = S_3 = \frac{S_6}{2S_7} \cdot S_5$$

In [15]:
```
S3 = (S6 / (2 * S7)) * S5   # aspect ratio of M3
S4 = S3                     # aspect ratio of M4

print(f"S3 = S4 = {S3:.3f}")
```

```
S3 = S4 = 5.504
```

10. Form factor of M9

Transistor M9 operates in **linear region**, because we want it to act as a ***resistive component*** to balance the zero effect. We extract the relation needed for the sizing of M9 by the equation derived for the *zero-nulling resistance* R_C (seen later):

$$\frac{1}{g_{m6}}\left(1 + \frac{C_L}{C_C}\right) = \frac{1}{\beta_p' S_9 V_{OV_{12}}}$$

Assuming $S_{12} = S_{13}$, $V_{OV_{12}} = V_{OV_{13}}$ and $V_{OV_{13}} = V_{OV_6}$:

$$S_9 = \frac{C_C}{C_C + C_L} \cdot \sqrt{\frac{S_{12} S_6 I_6}{I_{12}}} = \frac{C_C}{C_C + C_L} \cdot S_6$$

In [16]:
```
S9 = (Cc_min / (Cc_min + CL)) * S6  # aspect ratio of M9
S9_reduced = 0.9 * S9

print(f"S9 = {S9:.3f}")
print(f"S9_reduced = {S9_reduced:.3f}")
```

```
S9 = 11.007
S9_reduced = 9.907
```

In [17]:
```
i14 = i5
Vov14_max = 0.29 # considering Vs = -2.5 V and Vd = -2 V otherwise M10,M11 are in triode region

S14 = (2 * i14) / (kp_nmos * Vov14_max**2)  # aspect ratio of M14
S8 = 4 / S14
```

```
i8 = i14 * S8 / S14

print(f"i8 = {i8:.3e} A\nS8 = {S8:.3f}\ni14 = {i14:.3e} A\nS14 = {S14:.3f}")
```

```
i8 = 1.140e-05 A
S8 = 1.489
i14 = 2.056e-05 A
S14 = 2.686
```

In [18]:
```
i10 = i11 = i12 = i13 = i8
Vov10 = Vov14_max

S10 = S11 = S12 = S13 = (2 * i10) / (kp_nmos * Vov10**2)  # aspect ratio of M10

print(f"i10 = i11 = i12 = i13 = {i10:.3e} A")
print(f"S10 = S11 = S12 = S13 = {S10:.3f}")
```

```
i10 = i11 = i12 = i13 = 1.140e-05 A
S10 = S11 = S12 = S13 = 1.489
```

Biasing Resistor Rb

To define the value of the **biasing resistor** R_B we use the following relationships:

$$R_B = \frac{2}{g_{m_8}}\left(1 - \sqrt{\frac{S_8}{S_{14}}}\right)$$

In [19]:
```
gm8 = np.sqrt(2 * kp_nmos * i8 * S8)        # transconductance of M8
rb = 2 / (gm8 * (1 - np.sqrt(S8 / S14)))    # biasing resistor

print(f"Rb = {rb/1e3:.2f} kOhm")
```

```
Rb = 99.63 kOhm
```

Zero-nulling Resistor Rc

We use transistor M9 in **linear region** to emulate the behavior of a resistor R_C, in order to *balance the effect of the zero* by properly tuning the value of this resistance:

$$R_C = \frac{1}{\beta_p' \cdot S_9 \cdot V_{OV_9}}$$

$$R_C = \frac{C_L + C_C}{C_C} \cdot \frac{1}{\sqrt{2\beta_p' S_6 I_6}} = \frac{1}{g_{m_6}}\left(1 + \frac{C_L}{C_C}\right)$$

```
In [20]: gm6 = kp_pmos * S6 * Vov6       # transconductance of M6
         Rc = 1 / gm6 * (1 + (CL/Cc_min))    # zero nulling resistor

         print(f"gm6 = {gm6:.3e} S")
         print(f"Rc = {Rc/1e3:.2f} kOhm")
```

```
gm6 = 3.037e-04 S
Rc = 7.30 kOhm
```

DC Gain

$$A_d \simeq \frac{g_{m_1}}{g_{o_2} + g_{o_4}} \cdot \frac{g_{m_6}}{g_{o_6} + g_{o_7}}$$

```
In [21]: gm1 = np.sqrt(2 * kp_nmos * i1 * S1)    # transconductance of M1
         go2 = lambda_nmos * i2                  # output conductance of M2
         go4 = lambda_pmos * i4                  # output conductance of M4
         go6 = lambda_pmos * i6                  # output conductance of M6
         go7 = lambda_nmos * i7                  # output conductance of M7

         Ad = gm1 / (go2 + go4) * gm6 / (go6 + go7)  # DC gain
         Ad_dB = 20 * np.log10(Ad)                   # DC gain in dB

         print(f"DC gain ≃ {Ad_dB:.2f} dB")
```

```
DC gain ≃ 106.42 dB
```

Power Dissipation

We can calculate the **power dissipation** of our circuit knowing the *second stage current* I_7, the *differential pair current* I_5 and the *voltage drop across the circuit* $V_{DD} - V_{SS}$:

$$P_D = (I_7 + I_5)(V_{DD} - V_{SS}) = [SR(C_C + C_L) + SR \cdot C_C]\,(V_{DD} - V_{SS})$$

```
In [22]: pow_d = (SR * Cc_min + SR * (Cc_min + CL)) * (VDD - VSS)  # power dissipation

         print(f"Power dissipation = {pow_d*1e3:.3f} mW")
```

```
Power dissipation = 0.331 mW
```

Phase 2: AC Analysis

$$\begin{cases} p_1 = -\frac{1}{C_{Miller} R_{out1}} = -\frac{1}{C_c g_{m6} R_{out2} R_{out1}} \\ p_2 = -\frac{g_{m6}}{C_L} \\ p_3 = -\frac{g_{m6}}{C_{GS6}} \cdot \frac{C_c}{C_c + C_L} \\ z = \left(1 + \frac{1}{g_{m6} R_c}\right) \cdot C_c \end{cases}$$

$$C_{GS_6} = \frac{2}{3} \cdot C_{OX} \cdot \mu_p \cdot W_6 \cdot L_6$$

```
In [23]: rout1 = 1 / lambda_nmos * i2
         rout2 = 1 / lambda_nmos * i4
         cgs6 = 2 / 3 * (Cox_pmos * W6 * L6)

         p1 = - 1 / (Cc_min * gm6 * rout1 * rout2)
         # p1 = - GBW / Ad * 2 * np.pi
         # p1 = - GBW * 2 * np.pi
         p2 = - gm6 / CL
         p3 = - (gm6 / cgs6) * (Cc_min / (Cc_min + CL))
         z = (1 + 1/(gm6 * Rc)) * Cc_min

         poles = [p1, p2, p3]
         zeros = [z]
```

```
In [24]: pz_data = []
```

```
for i, p in enumerate(poles):
    pz_data.append(("Pole", f"p{i+1}", fp1.format_value(p), fp1.format_freq(p)))
for i, z in enumerate(zeros):
    pz_data.append(("Zero", f"z{i+1}", fp1.format_value(z), fp1.format_freq(z)))

df_pz_f = pd.DataFrame(pz_data, columns=["Type", "Index", "Value (rad/s)", "Freq (Hz)"])
df_pz_f
```

Out[24]:

	Type	Index	Value (rad/s)	Freq (Hz)
0	Pole	p1	-7.58e+20	1.21e+20
1	Pole	p2	-6.07e+07	9.67e+06
2	Pole	p3	-5.44e+07	8.66e+06
3	Zero	z1	5.97e-12	9.50e-13

Recap: Transistors Sizing

In [25]:

```
# Recap of transistor dimensions and currents
transistor_data = [
    ("M1", S1, 1, S1 / 1, i1),
    ("M2", S2, 1, S2 / 1, i2),
    ("M3", S3, 1, S3 / 1, i3),
    ("M4", S4, 1, S4 / 1, i4),
    ("M5", S5, 1, S5 / 1, i5),
    ("M6", W6 * 1e6, L6 * 1e6, S6, i6),
    ("M7", S7, 1, S7 / 1, i7),
    ("M8", S8, 1, S8 / 1, i8),
    ("M9", S9, 1, S9 / 1, 0),
    ("M10", S10, 1, S10 / 1, i10),
    ("M11", S11, 1, S11 / 1, i11),
    ("M12", S12, 1, S12 / 1, i12),
    ("M13", S13, 1, S13 / 1, i13),
    ("M14", S14, 1, S14 / 1, i14),
]
```

```
df_transistors_f = pd.DataFrame(transistor_data, columns=["Transistor", "W (µm)", "L (µm)", "W/L", "I (A)"]).assign(**{
        "W (µm)": lambda df: df["W (µm)"].map("{:.2f}".format),
        "L (µm)": lambda df: df["L (µm)"].map("{:.2f}".format),
        "W/L":    lambda df: df["W/L"].map("{:.2f}".format),
        "I (µA)": lambda df: (df["I (A)"] * 1e6).map("{:.2f}".format)
    }).drop(columns=["I (A)"])
df_transistors_f
```

Out[25]:

	Transistor	W (µm)	L (µm)	W/L	I (µA)
0	M1	4.46	1.00	4.46	10.28
1	M2	4.46	1.00	4.46	10.28
2	M3	5.50	1.00	5.50	10.28
3	M4	5.50	1.00	5.50	10.28
4	M5	2.06	1.00	2.06	20.56
5	M6	155.05	6.36	24.39	45.56
6	M7	4.57	1.00	4.57	45.56
7	M8	1.49	1.00	1.49	11.40
8	M9	11.01	1.00	11.01	0.00
9	M10	1.49	1.00	1.49	11.40
10	M11	1.49	1.00	1.49	11.40
11	M12	1.49	1.00	1.49	11.40
12	M13	1.49	1.00	1.49	11.40
13	M14	2.69	1.00	2.69	20.56

In [26]:

```
component_data = [
    ["Compensation Capacitance", "Cc", f"{Cc_min * 1e12:.2f} pF"],
    ["Zero-nulling Resistor", "Rc", f"{Rc / 1e3:.2f} kΩ"],
    ["Biasing Resistor", "Rb", f"{rb / 1e3:.2f} kΩ"],
    ["DC Gain", "Ad", f"{Ad_dB:.2f} dB"],
    ["Power Dissipation", "Pd", f"{pow_d * 1e3:.2f} mW"]
]

comp_headers = ["Parameter", "Symbol", "Value"]
df_components_f = pd.DataFrame(component_data, columns=comp_headers)
df_components_f
```

Out[26]:

	Parameter	Symbol	Value
0	Compensation Capacitance	Cc	4.11 pF
1	Zero-nulling Resistor	Rc	7.30 kΩ
2	Biasing Resistor	Rb	99.63 kΩ
3	DC Gain	Ad	106.42 dB
4	Power Dissipation	Pd	0.33 mW

In [27]:

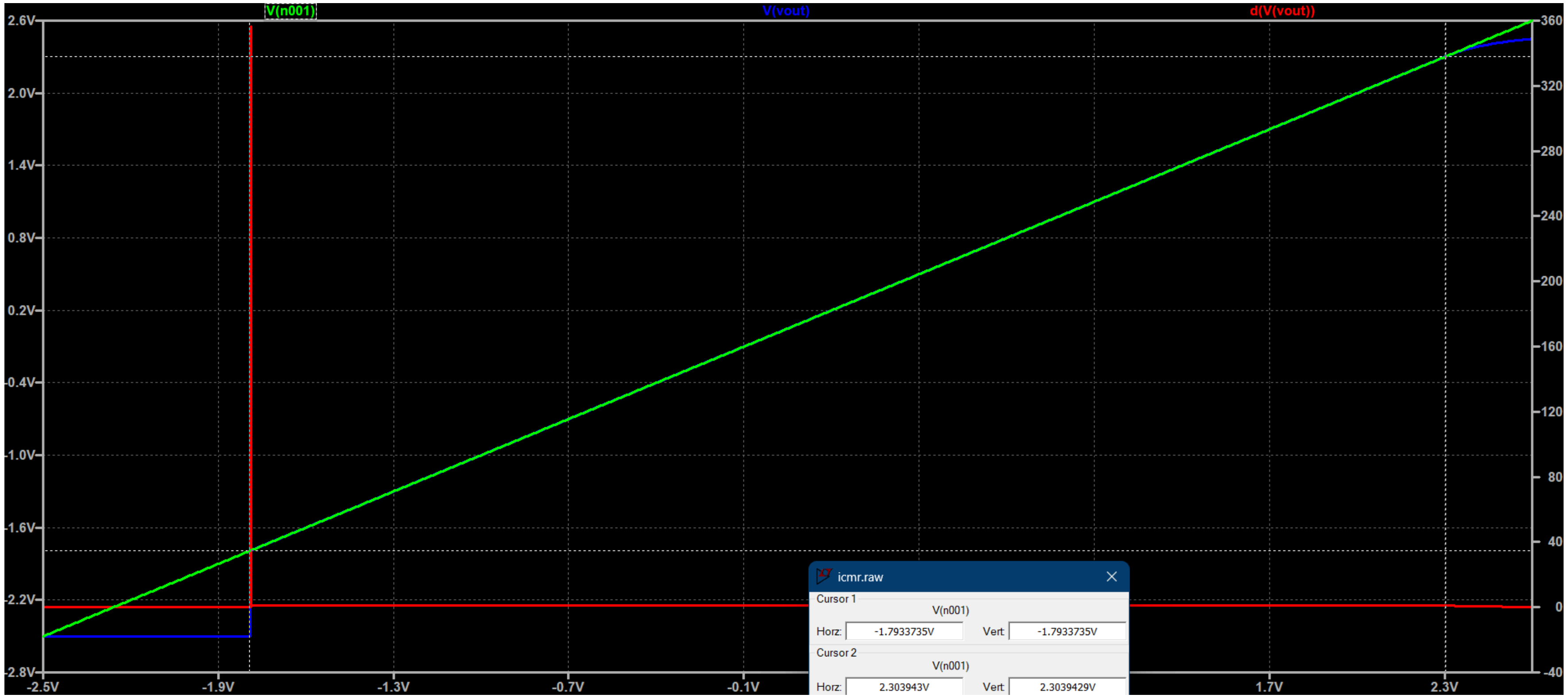
```
# All dataframes are saved in an external xslx file
dfs = {
    name: val for name, val in globals().items()
    if isinstance(val, pd.DataFrame) and name.endswith("_f")
}
fp1.save_df(dfs, "output_proj1_v2")
```

```
Output file 'output_proj1_v2.xlsx' saved in the current folder.
```

SPICE Plots Analysis

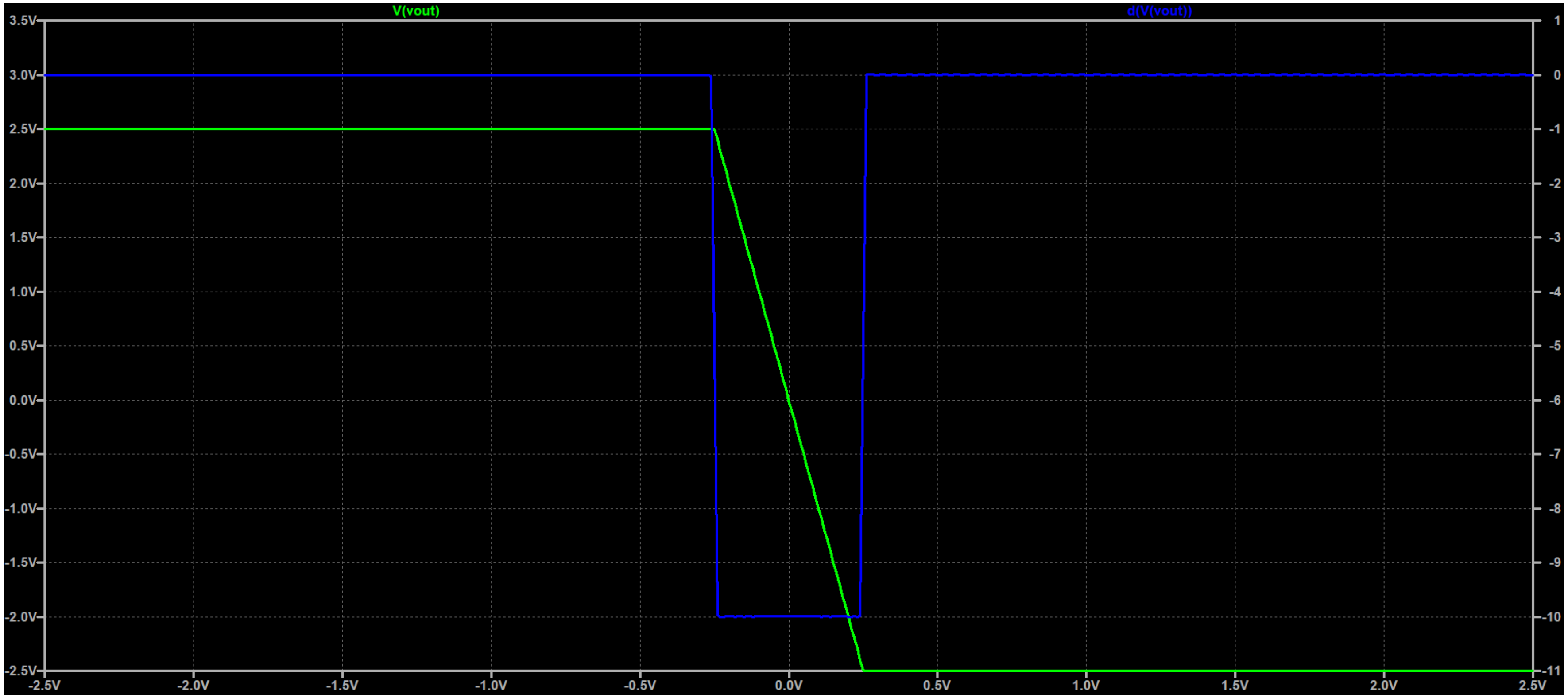
In this section, the results obtained from the circuit simulations will be analyzed, based on the given specifications and the calculated sizing values.

ICMR

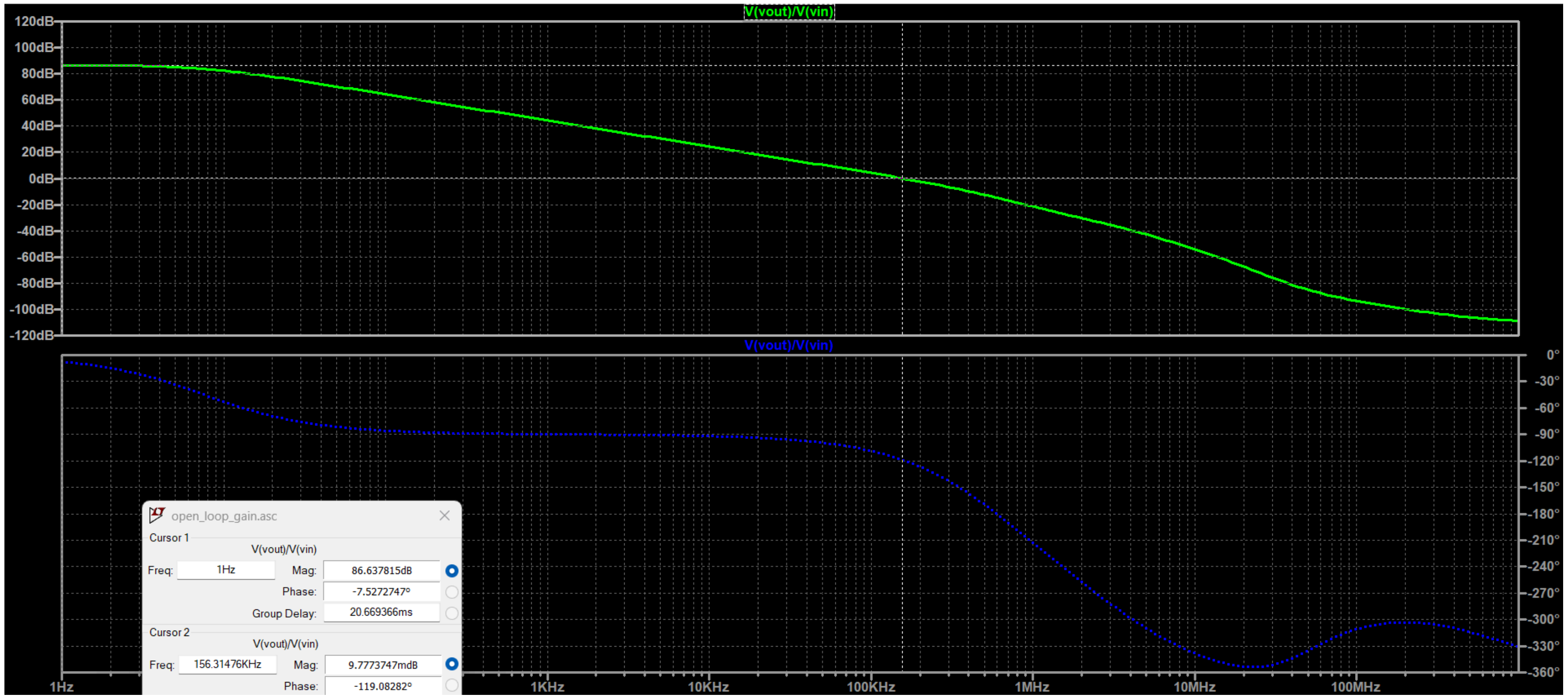


Cursor 1
V(n001)
Horz: -1.7933735V Vert: -1.7933735V
Cursor 2
V(n001)
Horz: 2.303943V Vert: 2.3039429V

Output Range



Frequency Analysis



open_loop_gain.asc

Cursor 1

V(vout)/V(vin)

Freq:	1Hz	Mag:	86.637815dB	●
		Phase:	-7.5272747°	○
		Group Delay:	20.669366ms	○

Cursor 2

V(vout)/V(vin)

Freq:	156.31476KHz	Mag:	9.7773747mdB	●
		Phase:	-119.08282°	○