SCHOOL OF ELECTRONIC ENGINEERING (SENSE)

Lab Report

ECE5002: Analog VLSI Design

Master of Technology In

Very Large-Scale Integration (VLSI) Design

Submitted to

Dr. Gurumurthy Komanapalli

By

Syed Haasin Sameer **(25MVD7048)**

Index of Lab Manual

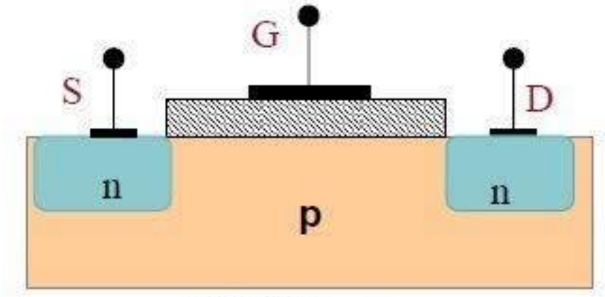
1. Transfer and Drain characteristics of NMOS

Aim: To design and simulate NMOS and Verify its Characteristics

Software & Technology Used: CADENCE VIRTUOSO, 180nm

Theory: MOSFETs

The metal-oxide semiconductor field-effect transistor (MOSFET) is a transistor used for amplifying or switching electronic signals. In MOSFETs, a voltage on the oxide-insulated gate electrode can induce a conducting channel between the two other contacts called source and drain. The channel can be of n-type or p-type and is accordingly called an nMOSFET or a pMOSFET (also commonly nMOS, PMOS). It is by far the most common transistor in both digital and analog circuits.



MOSFET Structure

Threshold voltage

To determine the threshold voltage we have to plot the ID vs. VGS curve and determine the x-axis intercept of the curve while maintaining VGS = VDS as shown in figure.

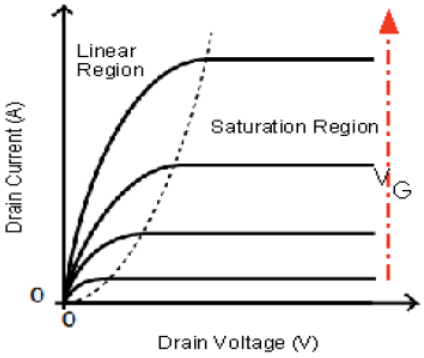


Figure 2: MOS IV characteristics

Procedure: -

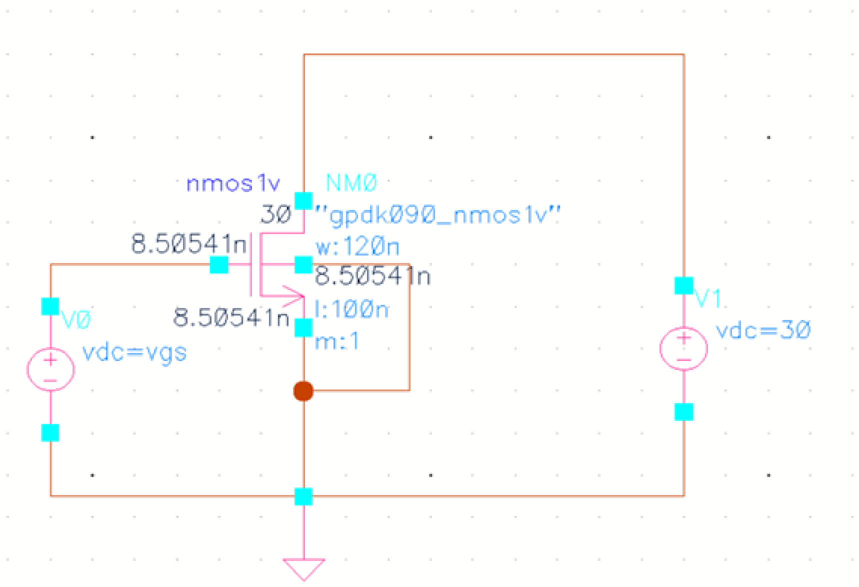
1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - csh
   - cd cd_work
   - virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select " Library".
4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.
5. Again click on " new" , now choose the Cell view

   5.1 Check Library, create a cell with cell name as "nm_c".
6. Now You got a schematic Editor.
7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.
8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

Schematic Diagram: -



Analysis: -

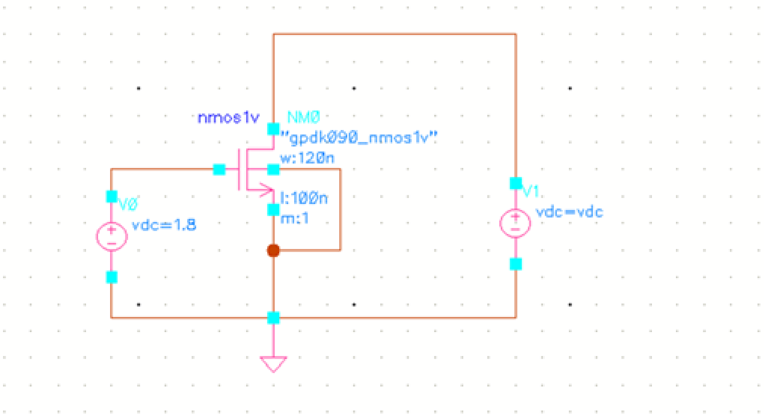
1. Transfer Characteristics (ID vs VGS)

Objective: Observe how ID changes with VGS (VDS fixed).

Steps:

1. Set VDS = 1.8 V (constant).
2. Open ADE_L → Analysis → DC.
3. Select Sweep Variable = VGS (e.g., 0 V to 1.8 V).
4. Go to Outputs → Select on Schematic → choose ID.
5. Run the simulation to get ID vs VGS plot.
6. From the graph, note:
    - Threshold Voltage (VTH)
    - Current rising region (saturation region)

Schematic Diagram: -



2. Output Characteristics (ID vs VDS)

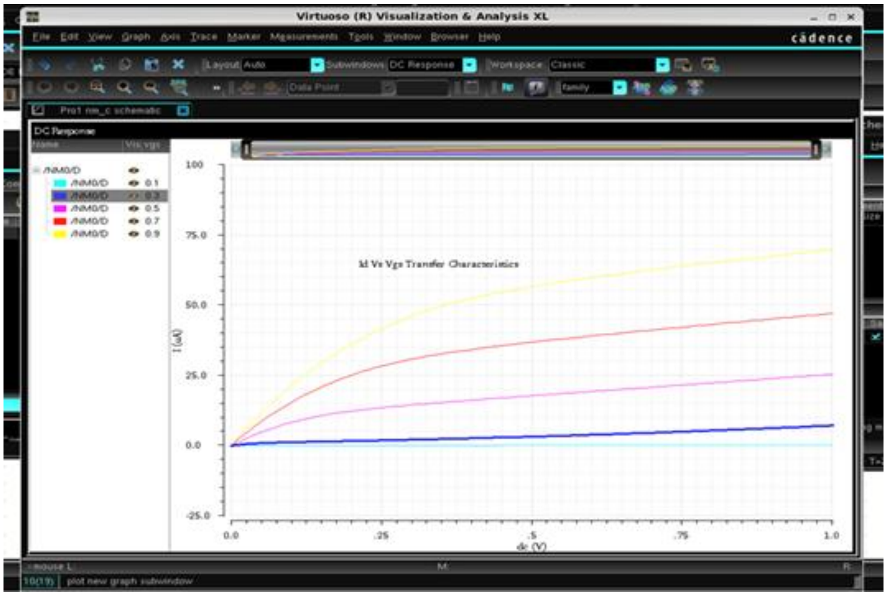
Objective: Observe how ID changes with VDS for different VGS.

Steps:

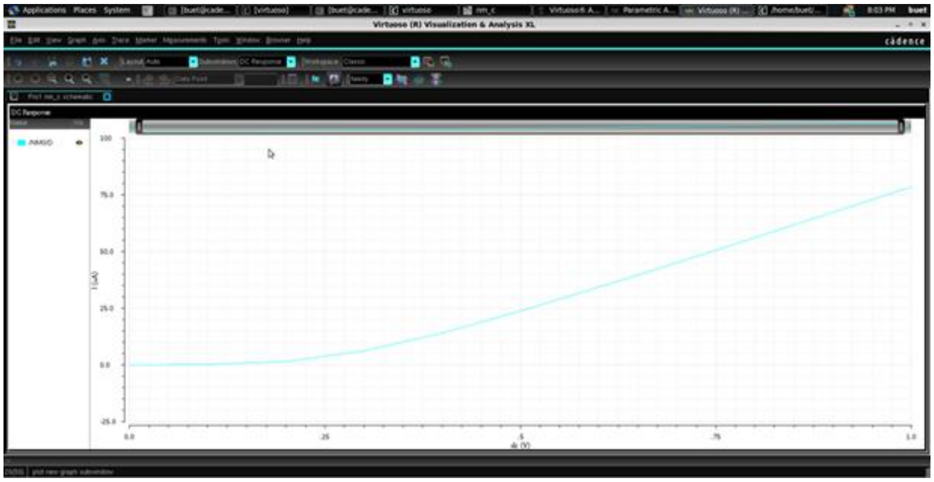
1. Set VGS = constant values (e.g., 0.9 V, 1.2 V, 1.5 V).
2. Open ADE_L → Analysis → DC.
3. Select Sweep Variable = VDS (e.g., 0–1.8 V).
4. Go to Outputs → Select on Schematic → choose ID.
5. Run the simulation to get ID vs VDS graph.
6. Observe:

- Linear region (small VDS)
- Saturation region (VDS > VGS − VTH)

Graphs & Observation: -

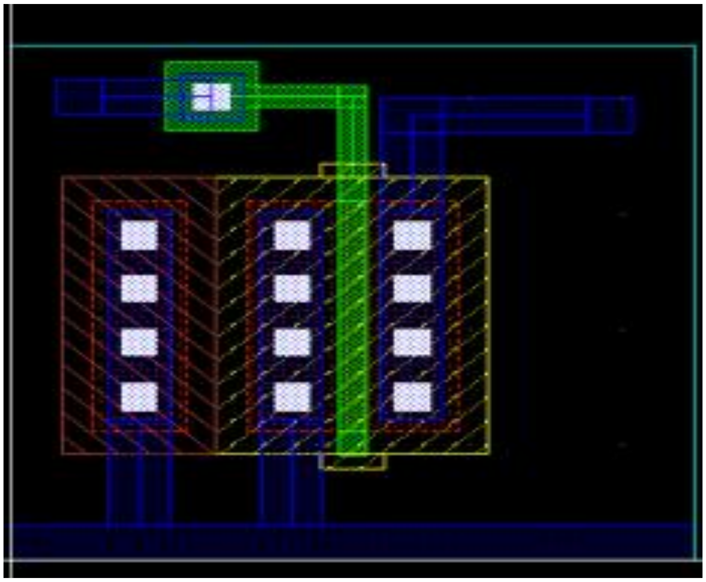


- ID vs VGS (Transfer): current ≈ 0 until VGS ≈ VTH; beyond VTH, ID rises (nearly quadratic in ideal model) — use intercept to estimate VTH.



- In the **ID vs VDS** graph, observe the **linear region** and **saturation region** of the NMOS transistor.
- The **ID vs VGS** curve shows the **threshold voltage (VTH)** where conduction starts.
- Drain current increases with higher VGS due to stronger channel formation.

Layout : -



Final Outcome

- The **Transfer** and **Drain** characteristics of NMOS were successfully simulated using **Cadence Virtuoso**.
- The experiment demonstrates:
  - Relationship between **ID**, **VGS**, and **VDS**.
  - **Saturation** and **Ohmic regions** of NMOS operation.
  - Extraction of **Threshold Voltage (VTH)** from the ID-VGS curve.

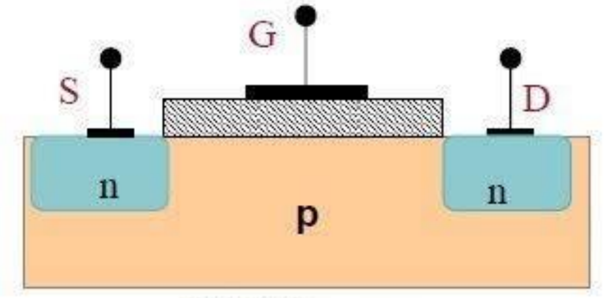
2. Transfer and Drain characteristics of PMOS

Aim: To design and simulate PMOS and Verify its Characteristics

Software & Technology Used: CADENCE VIRTUOSO, 180nm

Theory: MOSFETs

The metal-oxide semiconductor field-effect transistor (MOSFET) is a transistor used for amplifying or switching electronic signals. In MOSFETs, a voltage on the oxide-insulated gate electrode can induce a conducting channel between the two other contacts called source and drain. The channel can be of n-type or p-type and is accordingly called an nMOSFET or a pMOSFET (also commonly nMOS, PMOS). It is by far the most common transistor in both digital and analog circuits.



MOSFET Structure

Threshold voltage

To determine the threshold voltage we have to plot the ID vs. VGS curve and determine the x-axis intercept of the curve while maintaining VGS = VDS as shown in figure.

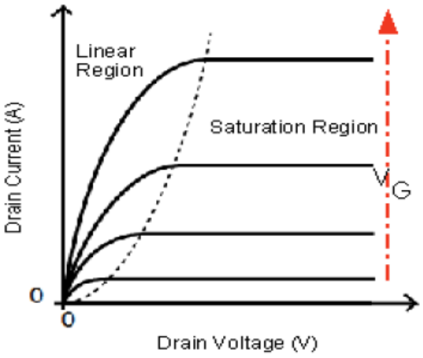


Figure 2: MOS IV characteristics

Procedure: -

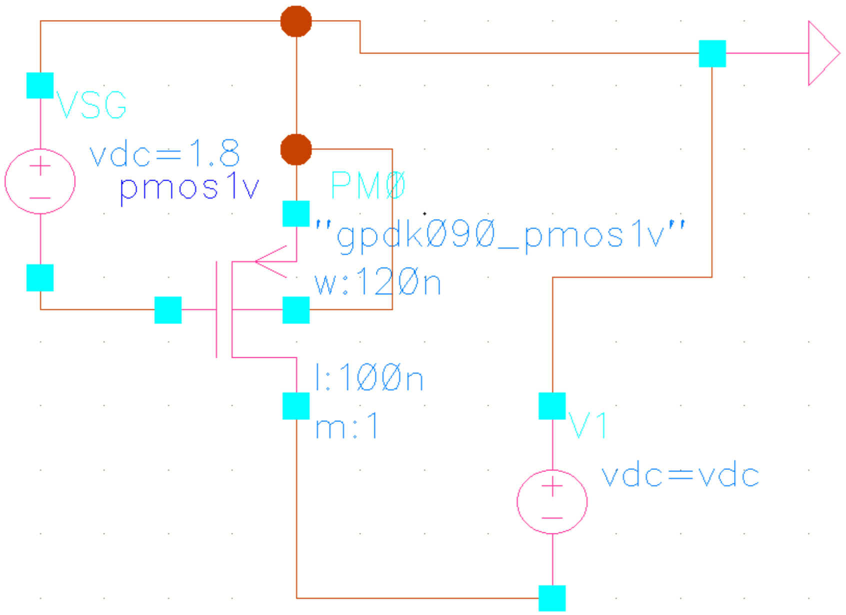
1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - ➢ csh
   - ➢ cd cd_work
   - ➢ virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select " Library".
4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.
5. Again click on " new" , now choose the Cell view

   5.1 Check Library, create a cell with cell name as "nm_c".
6. Now You got a schematic Editor.
7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.
8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

Schematic Diagram: -



Analysis: -

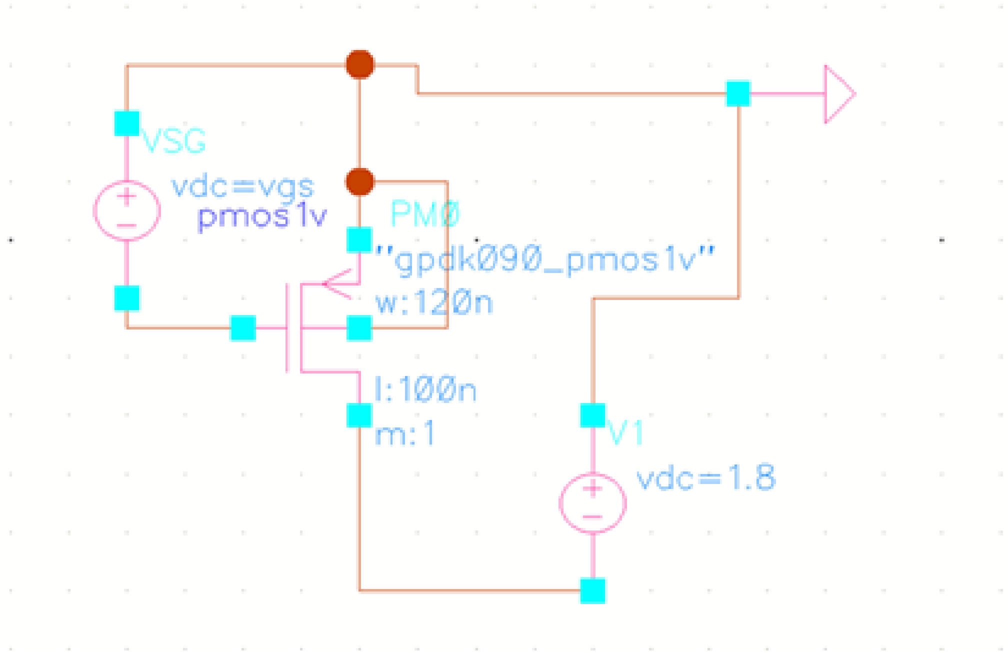
1. Transfer Characteristics (ID vs VGS)

Objective: See how ID changes with gate voltage.

Steps:

1. Keep VDS = constant (e.g., –1.8 V).
2. Open ADE_L → Analysis → DC.
3. Set sweep variable = VGS (range: 0 → –1.8 V).
4. Go to Outputs → Select on Schematic → choose ID.
5. Run simulation.
6. Observe the threshold voltage and how ID increases as VGS becomes more negative.

Schematic Diagram: -



2. Output Characteristics (ID vs VDS)

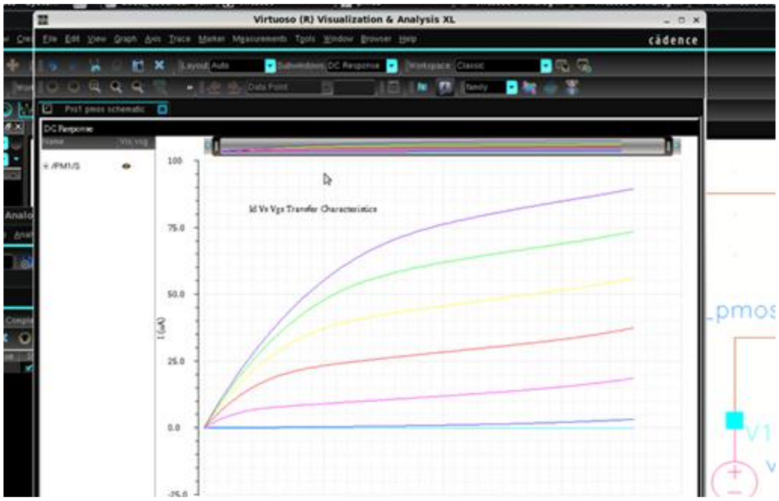
Objective: Observe how ID changes with VDS for different VGS.

Steps:

7. Set VGS = constant values (e.g., 0.9 V, 1.2 V, 1.5 V).
8. Open ADE_L → Analysis → DC.
9. Select Sweep Variable = VDS (e.g., 0–1.8 V).

10. Go to Outputs → Select on Schematic → choose ID.
11. Run the simulation to get ID vs VDS graph.
12. Observe:
    - Linear region (small VDS)
    - Saturation region (VDS > VGS − VTH)

Graphs & Observation: -


Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
Data Point
Pro1 pmos schematic
DC Response
/PM1/S
Id Vs Vgs Transfer Characteristics
100
75.0
50.0
25.0
0.0
-25.0
I (uA)
pmos
V1

Layout Auto
Subwindows DC Response
Workspace Classic
family

1. ID vs VGS (Transfer Curve)

ID ≈ 0 until VGS reaches threshold (negative value).

After VTH, ID increases as VGS becomes more negative.

The curve helps estimate threshold voltage of PMOS.

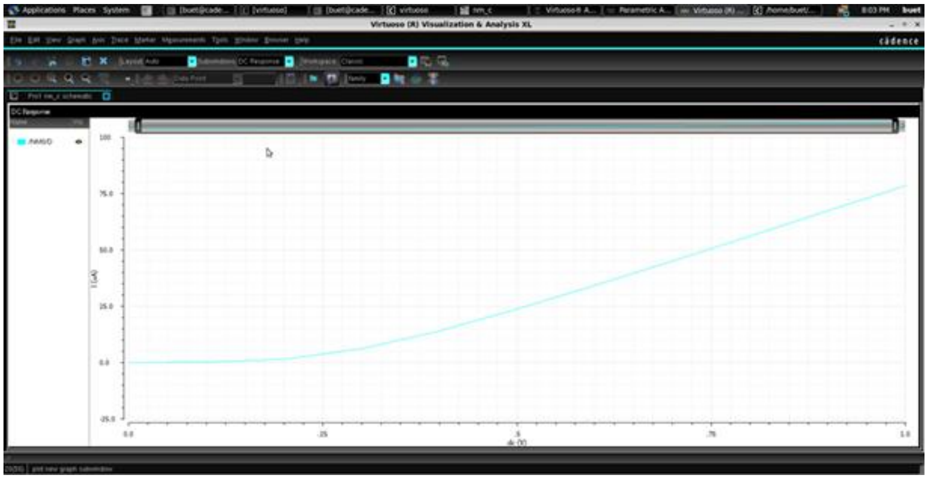
2. ID vs VDS (Output Curve)

For each VGS value:

At small |VDS| → Linear region (ID rises with VDS).

At larger |VDS| → Saturation region (ID almost constant).

Larger negative VGS gives higher ID and earlier saturation.



Final Outcome

- PMOS transfer (ID–VGS) and output (ID–VDS) characteristics were successfully simulated.
- The experiment confirms the relationship between ID, VGS, and VDS.
- Both ohmic and saturation regions of PMOS operation were clearly observed.

3. Common Source Amplifier with Common source using NMOS

Aim

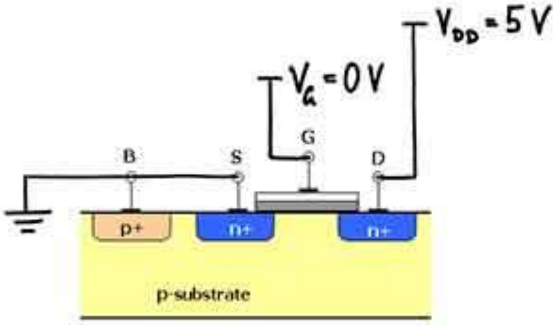
To design and simulate a **Common Source (CS) Amplifier with a Common Source (CS) Load** using **Cadence Virtuoso**, and to analyze its transfer characteristics, voltage gain, and frequency response.

Software Used

- **Cadence Virtuoso (ADE L Environment)**
- **GPDK 90 nm / 180 nm CMOS Technology Library**
- **VMware Workstation** (Linux-based platform)

Theory (NMOS Common Source Amplifier)

The **Common Source (CS) Amplifier** is a fundamental single-stage MOSFET amplifier that provides **voltage amplification** with **phase inversion**.
In this experiment, the CS amplifier uses a **CS load** (an active load transistor) instead of a resistive load, which increases the effective load resistance and hence the overall voltage gain.



1. Operation Principle

- The **input** signal is applied at the **gate** of the NMOS transistor.
- The **output** is taken from the **drain**.

- The **source** terminal is common for both input and output, typically connected to ground.
- A **second transistor** acts as an active load (CS load) replacing the drain resistor, enhancing gain and linearity.

2. Current and Voltage Relations

When the NMOS operates in **saturation region**, the drain current $I_D$is given by:

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{GS} - V_{TH})^2(1 + \lambda V_{DS})$$

where,

- μ_n= Electron mobility
- C_{ox}= Gate oxide capacitance per unit area
- W/L= Width-to-length ratio of the transistor
- V_{TH}= Threshold voltage
- λ= Channel length modulation parameter

3. Voltage Gain

For small-signal operation, the **voltage gain** of the CS amplifier is:

$$A_v = -g_m(r_{o1} \parallel r_{o2})$$

where,
$g_m = \frac{\partial I_D}{\partial V_{GS}} = \mu_n C_{ox}\frac{W}{L}(V_{GS} - V_{TH})$is the transconductance,
and $r_{o1}, r_{o2}$are the output resistances of the amplifier and load transistors, respectively.

The **negative sign** indicates a **180° phase shift** between input and output.

4. Significance of CS Load

Using a **MOSFET as a load** provides much higher dynamic resistance compared to a resistor, resulting in:

- **Higher gain**
- **Better linearity**
- **Reduced power dissipation**

Procedure

Step 1: Launch Cadence Virtuoso

1. Open **VMware** and start the **Cadence Virtuoso** environment.
2. In the terminal, type:
3. csh
4. cd cd_work

5. virtuoso &
6. The **Cadence Log Window** will appear — expand it to continue.

step 2: Create a New Library

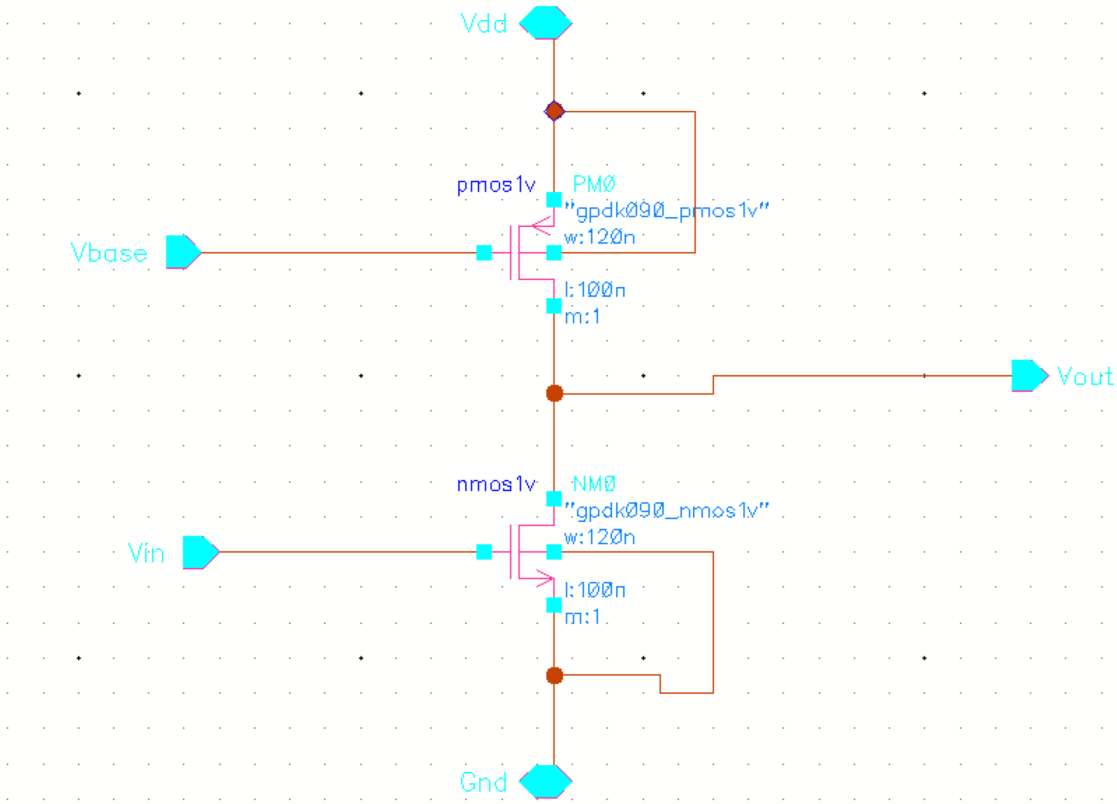
1. Go to **File → New → Library**.
2. Provide a name (e.g., Pro_1).
3. Choose **Attach to an Existing Technology Library** and click **OK**.
4. In the pop-up, select **gpdk90** or **gpdk180** as the technology library.

Step 3: Create a New Cell View

1. Again, go to **File → New → Cell View**.
2. Set:
    - **Library:** Pro_1
    - **Cell Name:** CS_CSLoad
    - **View:** Schematic
3. Click **OK** to open the **Schematic Editor**.

Step 4: Build the Circuit

1. Go to **Create → Instance**.
2. From **gpdk90** library, select:
    - nmos1v (Amplifying transistor)
    - pmos1v (Load transistor)
3. From **analogLib**, select:
    - vdc (for DC sources)
    - gnd (for ground)
4. Place and connect all the circuit elements as per the **Common Source Amplifier with CS Load** schematic.



Step 5: Assign Source Values

- **V<sub>base</sub> (Bias voltage):** Set DC bias to ensure the transistor operates in saturation.
- **V<sub>in</sub>:** Small AC signal (e.g., 1 V peak at 1 kHz).
- **V<sub>2</sub> (VDD):** Power supply voltage (e.g., 1.8 V).
- **Ground voltage:** Set to **−1 V** as reference.

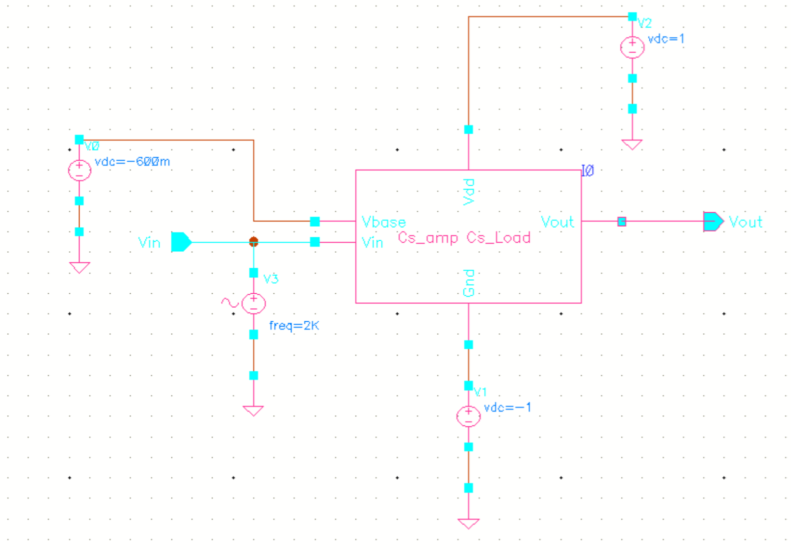
Step 6: Setup Analysis in ADE L

1. Open **ADE L (Analog Design Environment)**.
2. Go to **Analysis → Choose**, then set up:
   - **Transient (tran):**
     - Stop Time = 10 µs
     - Accuracy = Moderate
   - **DC Analysis (dc):**
     - For finding operating points.

    - **AC Analysis (ac):**
        - Frequency Range = 1 Hz to 1 MHz

3. Click **Outputs → To Be Plotted → Select on Schematic**, and select **Vin** and **Vout** nodes.
4. Finally, click **Netlist and Run** to start the simulation.

Schematic Diagram



Analysis:

1. The **output waveform (Vout)** is **inverted** with respect to the **input (Vin)**, confirming **phase inversion**.
2. The **output amplitude** is greater than the input, showing **amplification**.
3. **Voltage Gain:**

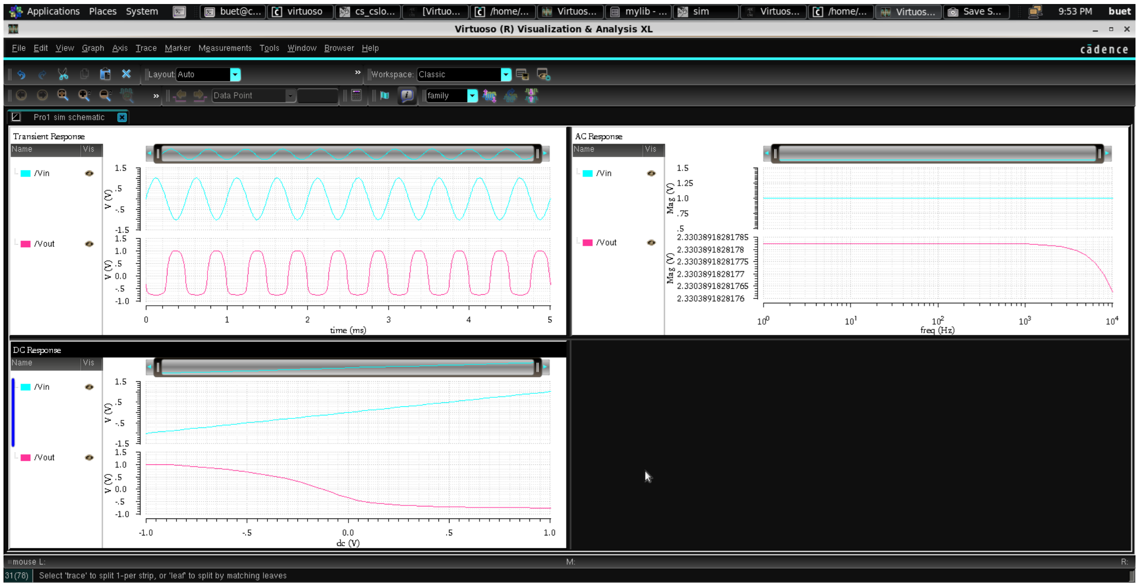
$$A_v = \frac{V_{out}}{V_{in}}$$

4. Using a **CS load** instead of a resistor increases the effective resistance, improving the **voltage gain**.
5. Both transistors must remain in **saturation** for linear amplification.

Graph & Observation :

Parameter	**Description**	**Typical Value**
VDD	Supply Voltage	1.8 V
Vin	Input Signal	1 V peak

Vout	Output Signal	Inverted & Amplified
Gain (A<sub>v</sub>)	Voltage Gain	10–20 (approx.)
Phase Shift	Input–Output	180°

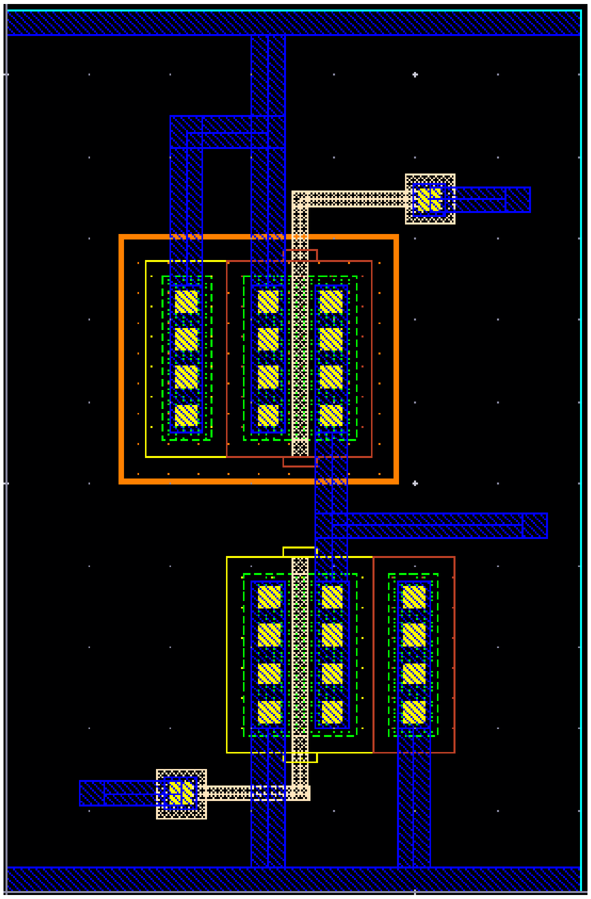


Select 'trace' to split 1-per strip, or 'leaf' to split by matching leaves

Graphs to be plotted:

1. **Vin vs Vout (Transient Response):** To verify phase inversion and amplification.
2. **Gain vs Frequency (AC Response):** To study frequency response characteristics.

Layout: -



Final Outcome

- Successfully designed and simulated a **Common Source Amplifier with Common Source Load** using **Cadence Virtuoso**.
- Verified that **Vout** is **180° out of phase** with **Vin**.
- Observed that **using a CS load increases voltage gain** due to higher output resistance.
- Demonstrated the **principle of voltage amplification** in a MOSFET-based analog circuit.

4. Analysis for CS Amplifier with Load Resistance

Aim: Design and simulate a Common-Source amplifier with an active PMOS load. Measure DC bias, small-signal gain, and output swing.

Software & Technology Used: CADENCE VIRTUOSO, 180nm

Theory: CS amplifier with load resistance (concise)

- Single transistor (M1) common-source stage with resistor RD at drain. Small-signal voltage gain:

$$A_v \approx -g_m \cdot R_{out}$$

where $R_{out}$is load seen by M1's drain (≈ RD ∥ r_{o1} for ideal resistor load).

- **Biasing:** choose VGS to set a quiescent ID so M1 is in saturation.
- **Bandwidth / pole:** input capacitances and load set high-frequency pole; coupling capacitors set low-frequency roll-off.
- Replace RD by a PMOS configured as a current source (active load). Active load produces a **much larger output resistance** → much higher gain.
- Small-signal approximate gain:

$$A_v \approx -g_{m1} \cdot (r_{o1} \parallel r_{o2})$$

Suggested nominal device sizes (adjust to technology):

- M1 (NMOS): W/L = 10/1
- M2 (PMOS load): W/L = 20/1
- VDD = +1.8 V (match gpdk library)
- Vbias ~ 0.9–1.1 V (tune to get desired ID)

<u>Procedure: -</u>

1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - csh
   - cd cd_work
   - virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select " Library".

4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.

5. Again click on “ new” , now choose the Cell view

5.1 Check Library, create a cell with cell name as “nm_c”.

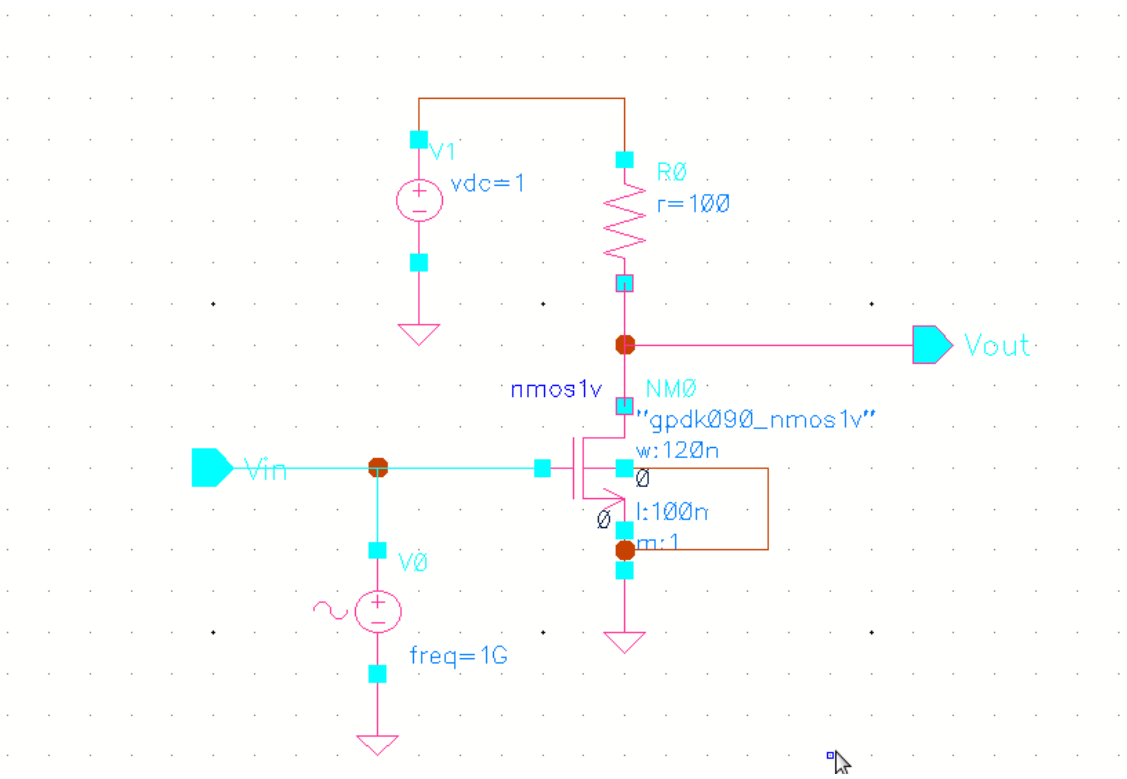
6. Now You got a schematic Editor.

7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.

8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

Schematic Diagram: -



Assign Component Values

- **VDC = 1.0 V**
- **Vin (AC)** = amplitude 1 V peak, frequency 1 G Hz
- **RD** = choose value as 100

Analysis: -

1. Transient Analysis

1. ADE-L → Analysis → Transient

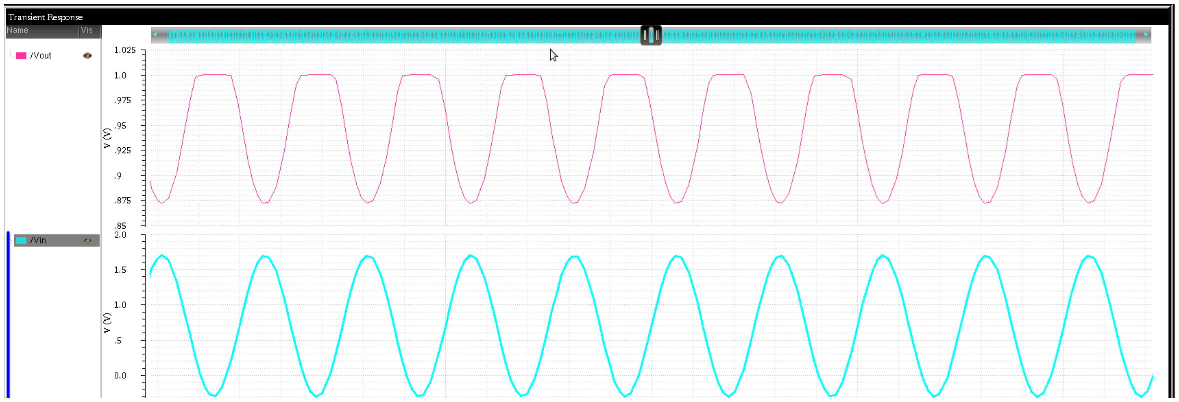
2. Stop time = 10 µs
3. Accuracy: Moderate
4. Outputs → “Select on Schematic”
   - Select Vin
   - Select Vout

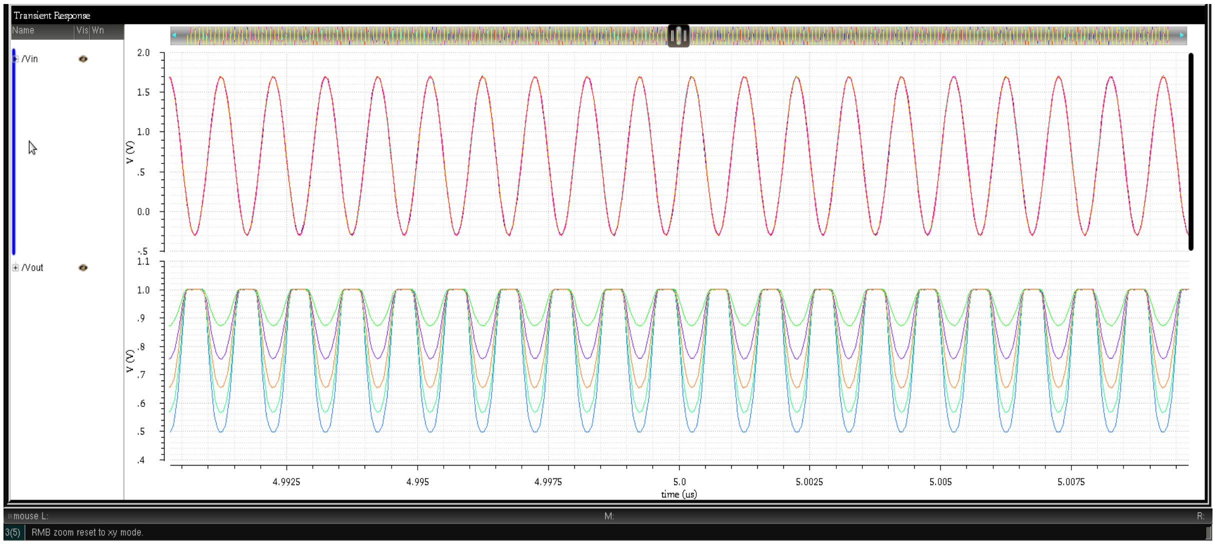
2. Parametric Analysis

To observe gain vs load resistance:

- Set RD as parameter (e.g., 10 kΩ, 20 kΩ, 50 kΩ…)
- Use Tools → Parametric Analysis

Graphs & Observation: -





3(5) RMB zoom reset to xy mode.

Analysis :

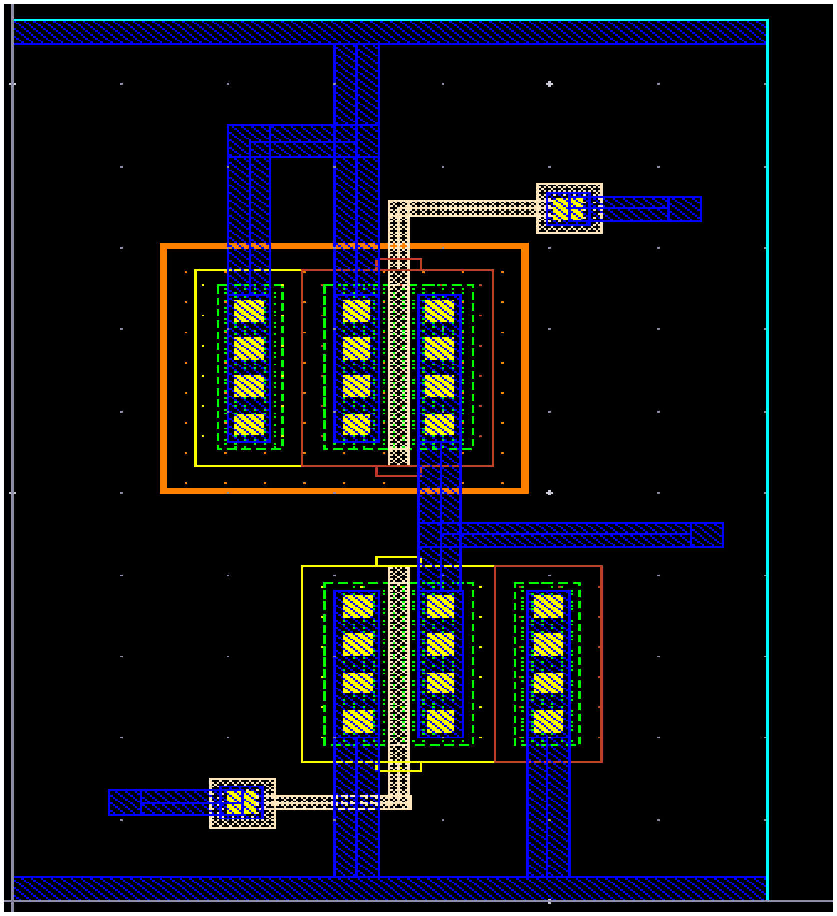
- **Vin vs Vout (Transient)**

  To verify inversion & amplification

- **Vout amplitude vs RD (Parametric)**

  Shows how load resistance affects gain

- **Layout : -**



Final Outcome

- Successfully designed and simulated a Common Source Amplifier with Load Resistance using Cadence Virtuoso.
- Verified that Vout is 180° out of phase with Vin.
- Observed that using a Resistance Value increases voltage gain due to higher output resistance.
- Demonstrated the principle of voltage amplification in a MOSFET-based analog circuit.

5. Common Gate

1. AIM

To design, simulate, and analyze a **Common Gate (CG) Amplifier** using Cadence Virtuoso and study its **DC**, **Transient**, and **AC** characteristics.

2. TECHNOLOGY USED & TOOLS / SOFTWARE

Technology

- **gpdk90** or **gpdk180** CMOS technology

Tools / Software

- **Cadence Virtuoso**
- **ADE L (Analog Design Environment)**
- Linux Terminal environment

3. THEORY – INTRODUCTION

A **Common Gate (CG)** amplifier is one of the fundamental MOSFET amplifier topologies, along with the Common Source (CS) and Common Drain (CD) configurations. In the CG amplifier:

- The **gate** is held at a fixed DC bias (AC-grounded).
- The **input signal** is applied to the **source** terminal.
- The **output** is taken from the **drain** terminal.

Because the gate is common to both input and output (in AC sense), the circuit is called **Common Gate**.

Key Features

- **Voltage Gain > 1** (medium gain)
- **No phase inversion** (0° phase shift between input & output)
- **Very low input impedance**
- **High output impedance**
- **Excellent high-frequency response** (no Miller effect)

These properties make the Common Gate amplifier suitable for:

- **Wideband amplifiers**
- **RF applications**
- **Low-input-impedance interfaces**

- **Current buffer stages**

Working Principle

A small change in input voltage at the source modifies the MOSFET’s VGS, causing a corresponding change in the drain current. This produces an amplified voltage across the drain load, resulting in an output signal that follows the input without inversion.

4. PROCEDURE

Step 1 – Open Cadence Environment

Open Linux Terminal and run:

```
csh

cd cd_work

virtuoso &
```

Step 2 – Create Project Library

- Go to **File → New → Library**
- Enter library name (e.g., *Pro_1*)
- Attach to technology: **gpdk90/gpdk180**

Step 3 – Create Cell for Schematic

- Go to **File → New → Cell View**
- Cell name: **Common_Gate**
- View: **schematic**

Step 4 – Build the Circuit

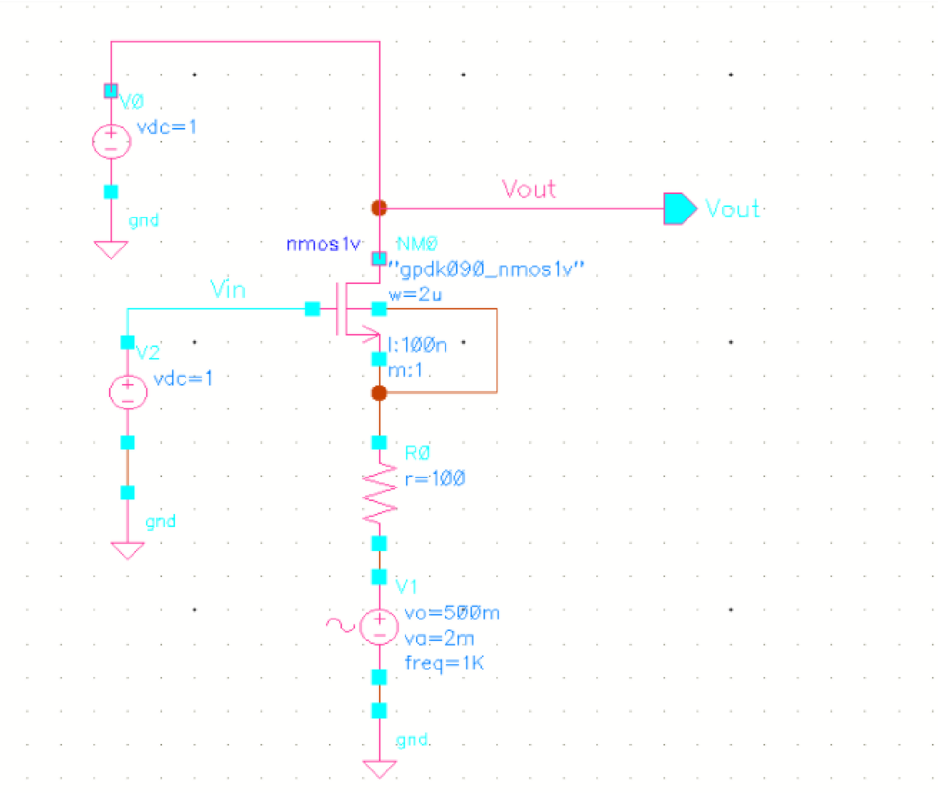
Place components using **Create → Instance**:

- **NMOS (nmos1v)**
- **VDC** for gate bias
- **Vsin** for small-signal AC input
- **VDD** for drain supply
- **Load resistor (if used)**
- Ground connections

Connect the circuit in **Common Gate configuration**:

- Gate → fixed DC bias
- Source → AC input signal
- Drain → output node (Vout)

5. SCHEMATIC DIAGRAM



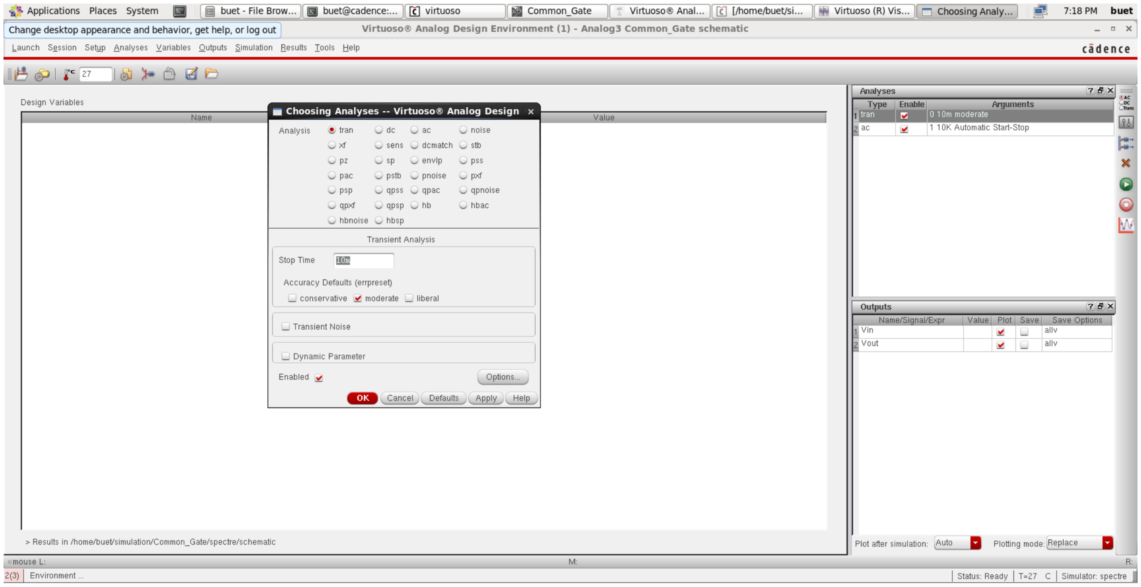
6. TYPES OF ANALYSIS

A) Transient Analysis

Objective:
Observe time-domain response, gain, and waveform tracking.

Steps:

- ADE L → **Analyses** → **tran**
- Stop time: (e.g., **5 ms**)
- Use small-signal input: Vsin (e.g., 2 mV, 1 kHz)
- Run simulation
- Plot input and output waveforms



Virtuoso® Analog Design Environment (1) - Analog3 Common_Gate schematic

Choosing Analyses -- Virtuoso® Analog Design

Analysis: ● tran

Transient Analysis

Stop Time: 10m

Accuracy Defaults (errpreset): □ conservative ☑ moderate □ liberal

□ Transient Noise

□ Dynamic Parameter

Enabled ☑

Analyses

	Type	Enable	Arguments
1	tran	☑	0 10m moderate
2	ac	☑	1 10K Automatic Start-Stop

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	Vin		☑	□	allv
2	Vout		☑	□	allv

> Results in /home/buet/simulation/Common_Gate/spectre/schematic

Plot after simulation: Auto Plotting mode: Replace

Status: Ready | T=27 C | Simulator: spectre

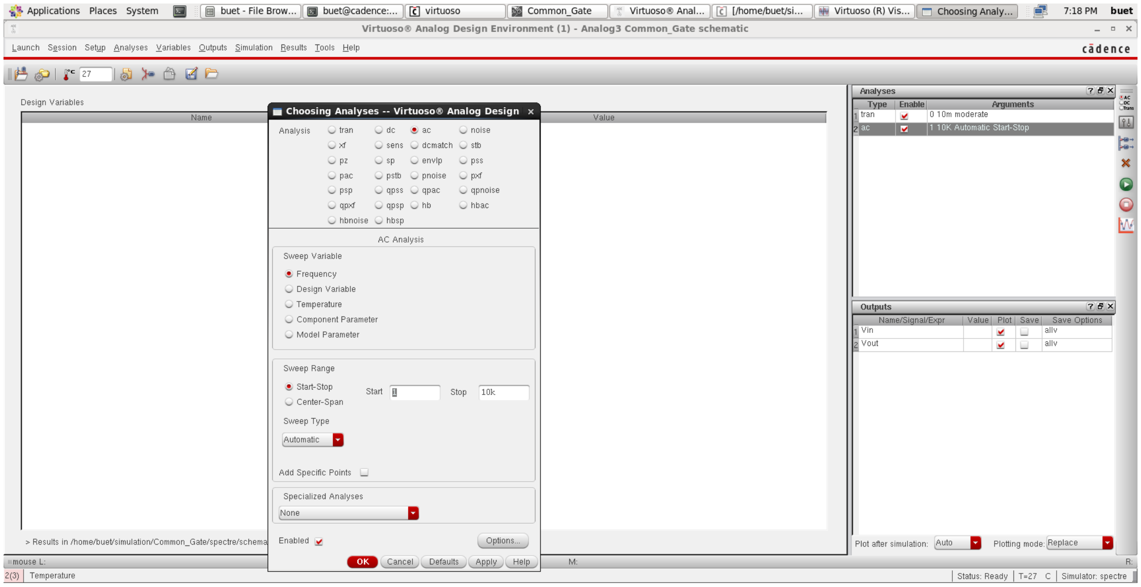
D) AC Analysis

Objective:

Determine small-signal voltage gain and frequency response (bandwidth).

Steps:

- ADE L → **Analyses → ac**
- Frequency sweep: **100 Hz to 150 MHz**
- AC magnitude = 1 V for easy gain calculation
- Run simulation
- Plot gain (Vout/Vin)



Virtuoso® Analog Design Environment (1) - Analog3 Common_Gate schematic

Choosing Analyses -- Virtuoso® Analog Design

Analysis: ● ac

AC Analysis

Sweep Variable: ● Frequency ○ Design Variable ○ Temperature ○ Component Parameter ○ Model Parameter

Sweep Range: ● Start-Stop ○ Center-Span Start: 1 Stop: 10k

Sweep Type: Automatic

Add Specific Points □

Specialized Analyses: None

Enabled ☑

Analyses

	Type	Enable	Arguments
1	tran	☑	0 10m moderate
2	ac	☑	1 10K Automatic Start-Stop

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	Vin		☑	□	allv
2	Vout		☑	□	allv

> Results in /home/buet/simulation/Common_Gate/spectre/schema

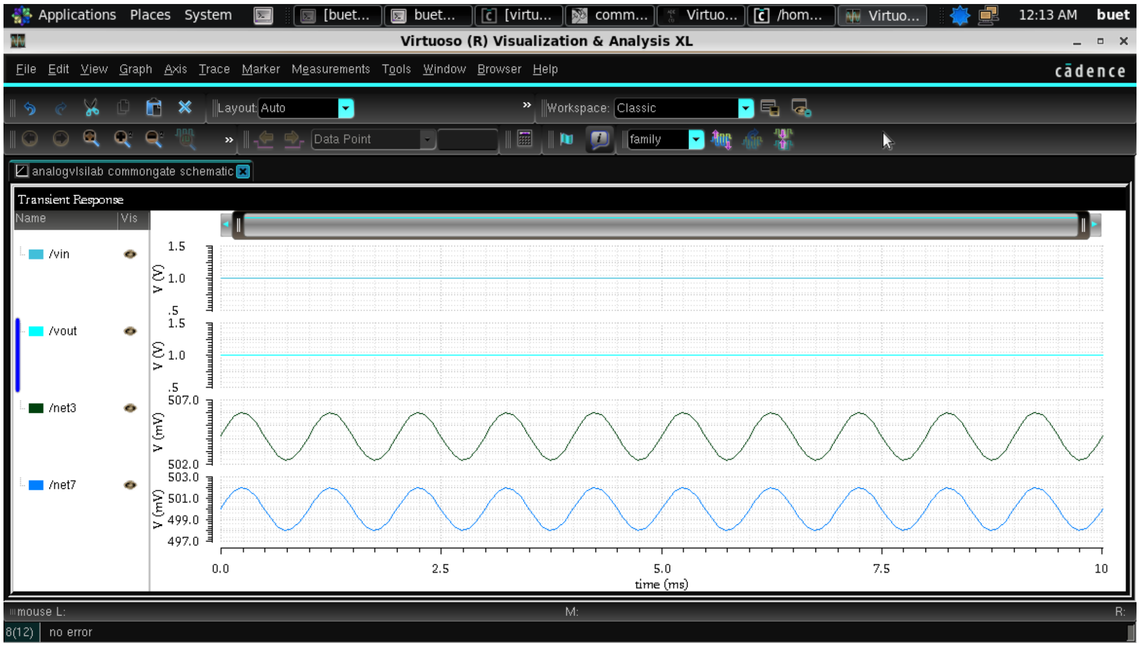
Plot after simulation: Auto Plotting mode: Replace

Status: Ready | T=27 C | Simulator: spectre

7. GRAPH OBSERVATIONS

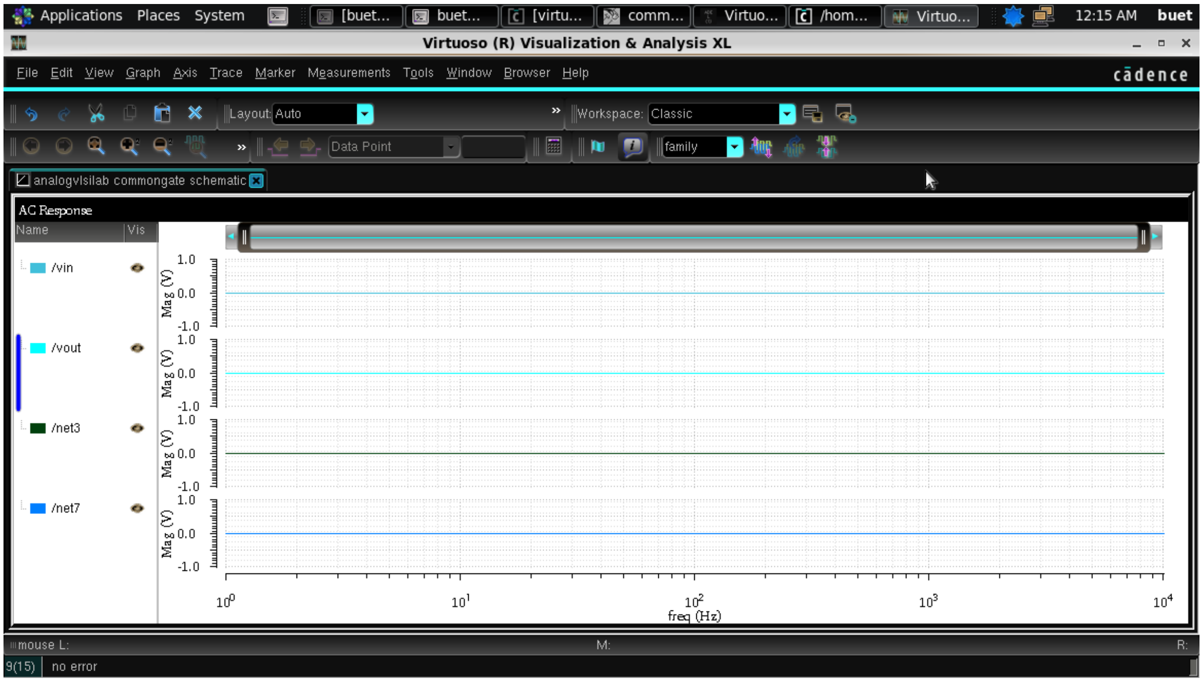
1. Transient Graph Observation

- Output waveform closely follows input waveform. No 180° phase shift → output is **in phase** with input. Slight gain above unity (depends on gm and RD).

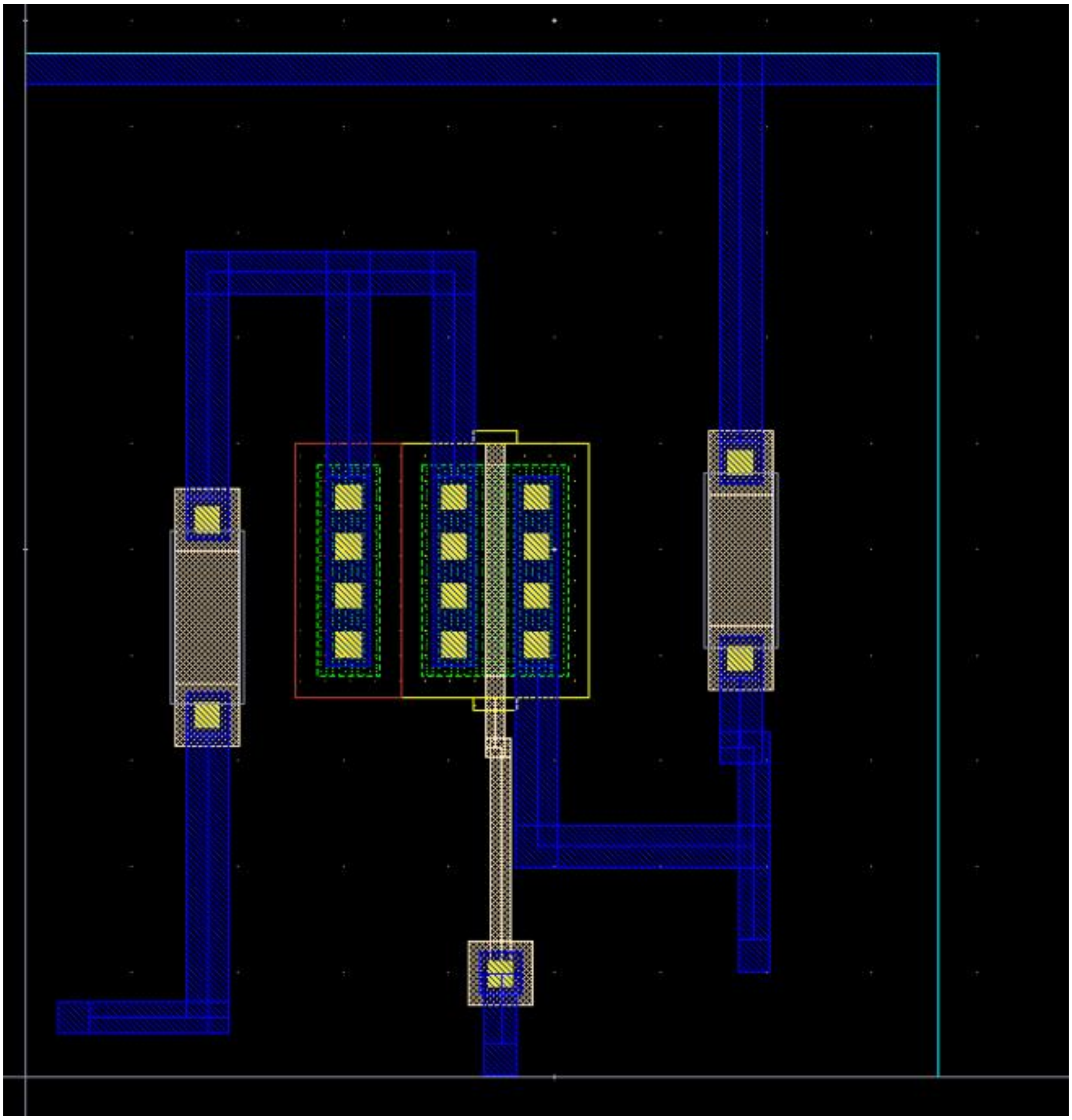


2. AC Graph Observation

- Low-frequency gain > 1 (typically gm × RD).



8. Layout:-



9. FINAL OUTCOME / RESULT

- The **Common Gate amplifier** was successfully designed and simulated in **Cadence Virtuoso**.
- The amplifier showed:
  ✓ **Medium voltage gain (>1)**
  ✓ **No phase inversion**

✓ **Low input impedance**

✓ **High output impedance**

✓ **Wide frequency bandwidth**

- Transient, and AC analyses confirm correct operation of the CG amplifier.

6. Common Drain

AIM : - To design, simulate, and analyze a **Common Drain (Source Follower)** amplifier using Cadence Virtuoso and study its **DC**, **Transient**, and **AC** characteristics.

TECHNOLOGY & TOOLS USED :-

- **Technology:** gpdk90 or gpdk180 CMOS
- **Tools:** Cadence Virtuoso, ADE L
- **Devices Used:** nmos1v (NMOS transistor)
- **Sources:** VDC (bias), Vsin (input AC), VDD, VSS, Ground

Theory: -

A **Common Drain Amplifier**, also known as a **Source Follower**, is one of the most important MOSFET amplifier configurations. In this circuit:

- **Input** is applied at the **gate**
- **Output** is taken from the **source**
- **Drain** is connected to the supply (VDD)

The name "source follower" comes from the fact that the **source voltage follows the input voltage**, only reduced by the gate–source voltage drop (VGS).

Key Characteristics :-

- **Voltage Gain ≈ 1** (slightly less than unity)
- **No phase inversion** (0° phase shift)
- **Very high input impedance**
- **Low output impedance**
- **Used as a buffer** to drive heavy loads

Basic Principle :-

When input voltage increases, the NMOS source rises to maintain constant VGS, hence:

$$V_{out} \approx V_{in} - V_{GS}$$

Because of its high input impedance and low output impedance, the common drain amplifier is widely used for **buffering** and **impedance matching** in analog ICs.

Procedure: -

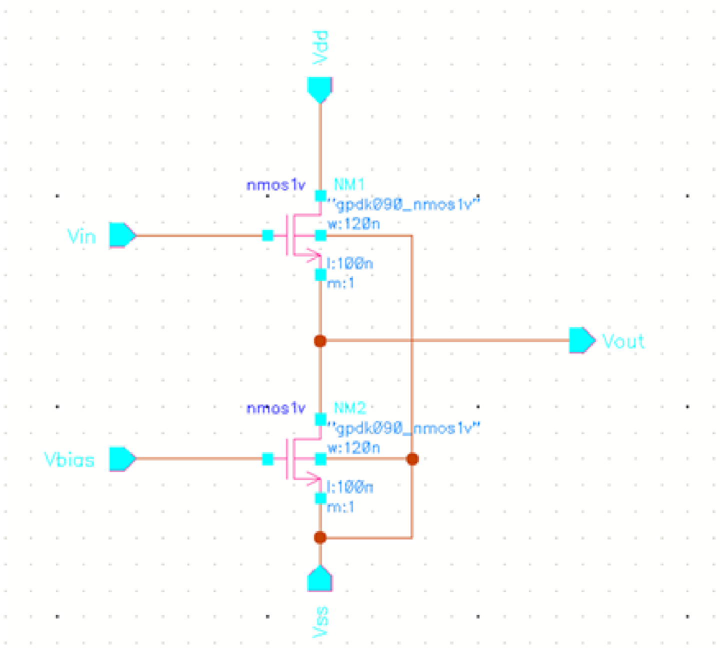
Step 1: Open Cadence

```
csh

cd cd_work

virtuoso &
```

Step 2: Create a Library & Cell

- File → New → Library → Attach gpdk90
- File → New → Cell View → Name: **nm_c**

Step 3: Build the Circuit:



- **NMOS (nmos1v)**
- **Vsin (Vin source)**
  - AC Magnitude: **1 V**
  - Amplitude: **5 mV**
  - Frequency: **1 kHz**
- **Vbias = 1.1 V**
- **VSS = –5 V**

- **VDD = +5 V**
- **Vout pin (output)**

Connect NMOS in **source follower** configuration:

- Gate → Vin
- Drain → VDD
- Source → Vout

Symbol creation is optional.

ANALYSIS SETUP (RUN ANALYSIS)

1) Transient Analysis

- ADE L → Analyses → tran
- Stop Time: **5 ms**
- Accuracy: Moderate
  Purpose: To observe the actual time-domain output waveform.

2) DC Analysis

- Analyses → dc
- Sweep variable → Vout
- Sweep range: **–5 V to +5 V**
  Purpose: To check DC relationship between Vin and Vout.

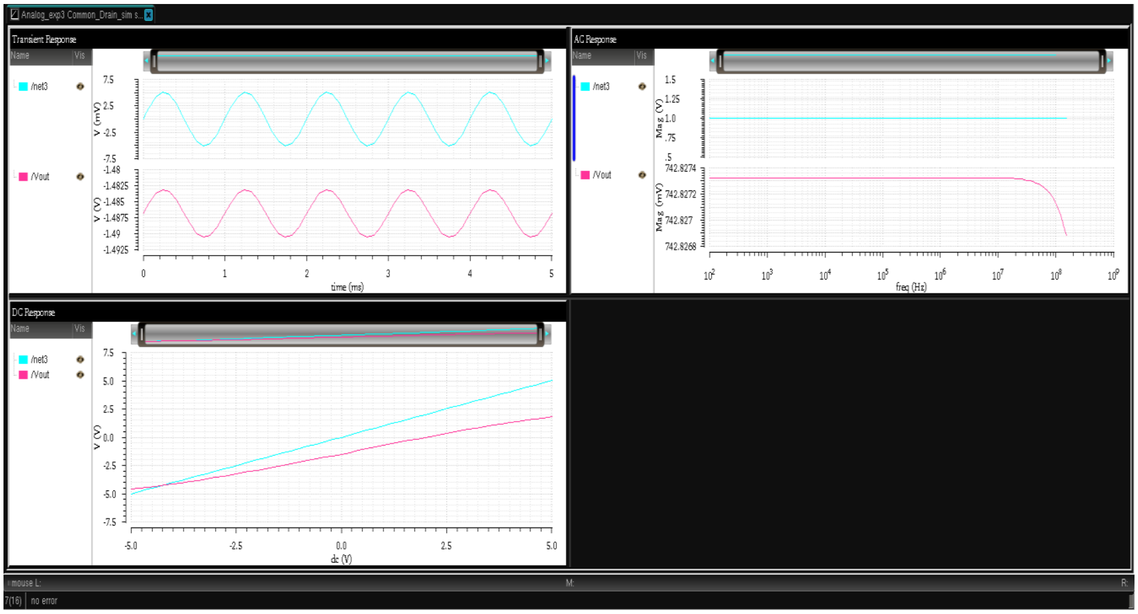
3) AC Analysis

- Analyses → ac
- Sweep Range → **100 Hz to 150 MHz**
- Sweep Type → Log
  Purpose: To observe voltage gain and bandwidth.

4) Select Outputs

- Outputs → To Be Plotted → Select Vin & Vout
- Run Simulation

GRAPH OBSERVATIONS



1. DC Graph Observation

- Vout increases as Vin increases.
- Vout ≈ Vin – VGS (about 0.7–0.9 V lower).
- No inversion is observed.

2. Transient Graph Observation

- Output follows the input sine wave closely.
- No phase change.
- Output amplitude slightly lower than input (~0.9 gain).

3. AC Graph Observation

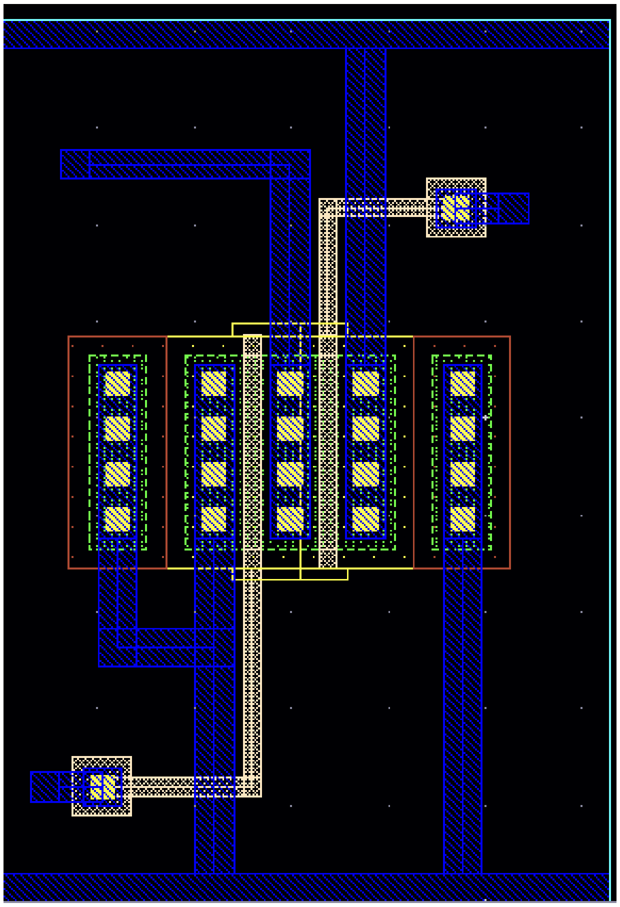
- Low-frequency gain ≈ **0.9–1** (close to unity).
- High bandwidth, gain falls at high frequencies.

RESULT

- The Common Drain amplifier was successfully designed using gpdk90.

- The voltage gain was observed to be **~0.9 to 1**.
- **No phase inversion** was observed.
- The amplifier showed **high input impedance, low output impedance**, and **good bandwidth**.

Layout:-



CONCLUSION

The **Common Drain (Source Follower)** amplifier works as a **buffer**. The output voltage closely follows the input with only a small loss due to VGS. The amplifier provides:

- High input impedance
- Low output impedance
- Unity voltage gain
- Wide bandwidth

7. Cascode Amplifier with Resistive Load

Aim: -

To design a Cascode Amplifier with R-Load and perform the AC, DC and Transient characteristics also Draw the layout and simulation using Cadence Virtuoso.

Technology Used & Tools: -

Technology :

- gpdk90 / gpdk180 CMOS Technology Library
- MOSFET devices: NMOS (nmos1v) and PMOS (pmos1v)

Tools / Software :

- Cadence Virtuoso Schematic Editor
- Analog Design Environment (ADE)
- VMware Linux Environment

Theory:-

A Common Source amplifier with a current-source load replaces the drain resistor with an active load, usually a MOSFET biased to work as a current source (constant-current device). The source is grounded, the input is applied at the gate, and the output is taken from the drain.

Because a current source has a very high output resistance, the small-signal load resistance seen at the drain becomes very large. This significantly increases the overall voltage gain of the amplifier. The gain is approximately:

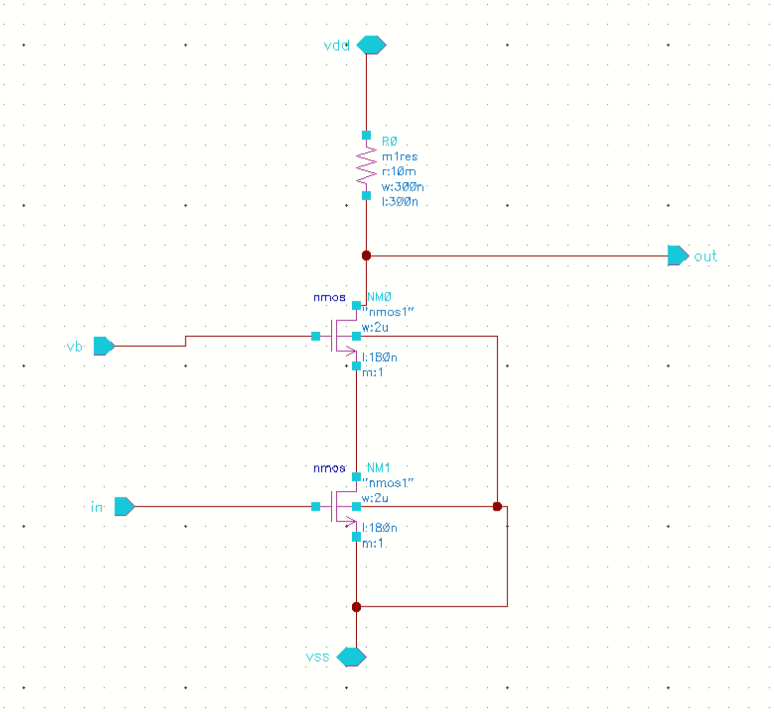
Since the load MOSFET has a large (r_{o,load}), the effective load resistance becomes much larger than a simple resistor, giving higher gain compared to a resistive-load CS amplifier. The amplifier still shows 180° phase inversion between input and output.

This topology is widely used in integrated circuits, where large resistors are impractical. The CS amplifier with current-source load offers high gain, good linearity, and low power consumption, making it suitable for analog front-end stages.

Procedure:-

1. Draw the schematic of the circuit using the required components in Cadence Virtuoso Schematic Editor.
2. Assign all the necessary pins such as input, output, supply, and ground using the Create → Pin option.
3. Create the symbol for the schematic by selecting Create → Cellview → From Cellview.
4. Create a testbench schematic and instantiate the generated symbol using Instantiate → Component.
5. Add and connect the required sources (DC, AC, pulse, supply) and load elements in the testbench.
6. Set up the simulation by launching ADE, selecting the appropriate analysis , and entering the required parameters.
7. Run the simulation and observe or plot the necessary output waveforms from the results window.
8. For the Layot go to layot XL window and draw the layot.
9. Perform DRC check and LVS check .
10. Obtain the EXTRACTED VIEW Of the LAYOT by Clicking Quantus Assura.
11. Create Config Window and Perform PRE-LAYOT and POST-LAYOT simulation.

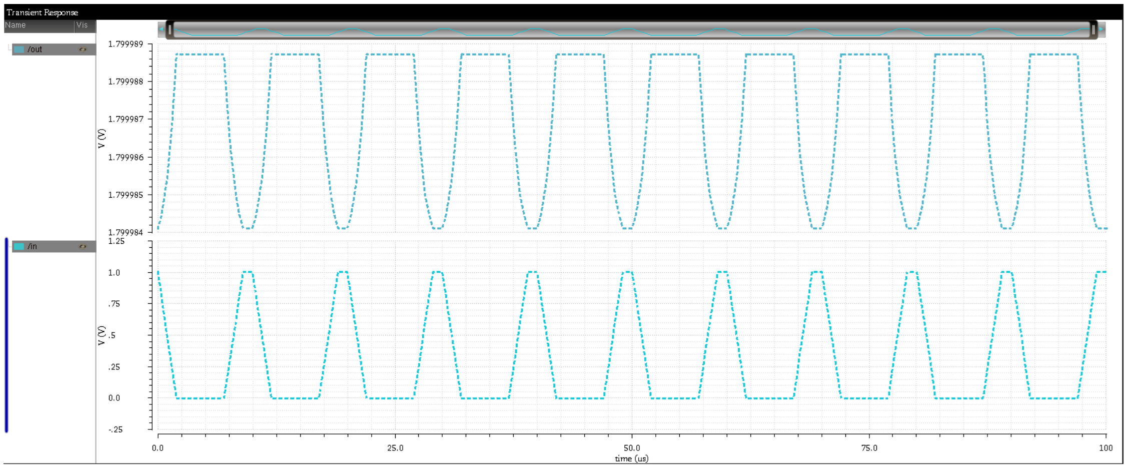
Schematic Diagram :-



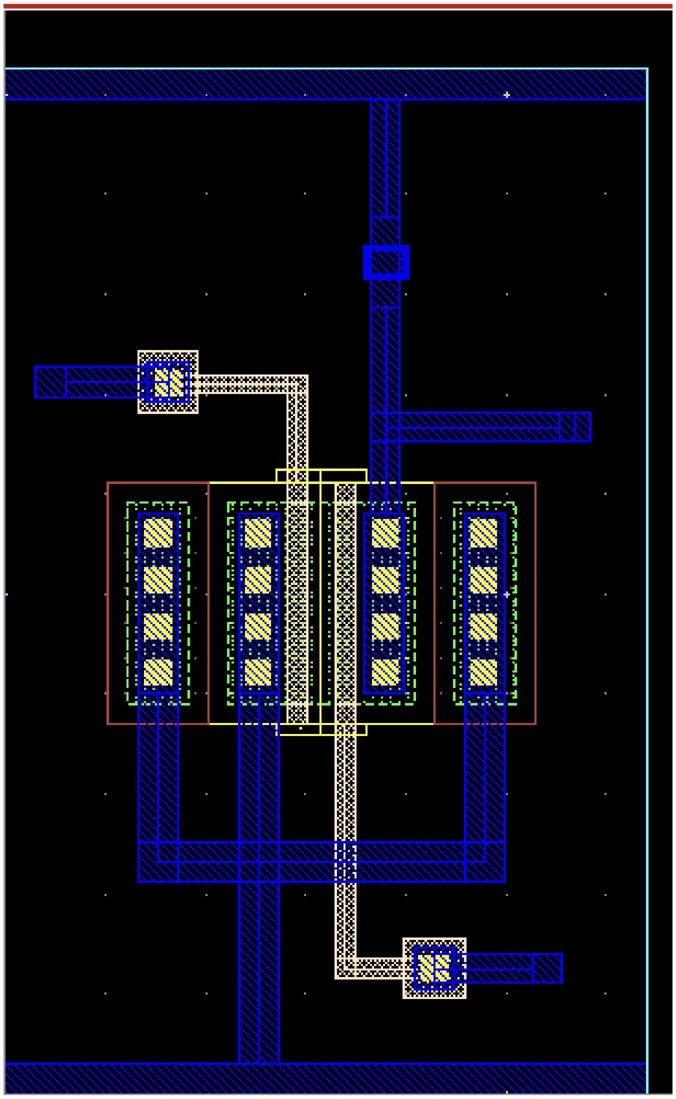
Transient Analysis: -

- ADE L → Analyses → tran
- Stop Time: **5 ms**
- Accuracy: Moderate

Purpose: To observe the actual time-domain output waveform.



Layout:-



Conclusion :-

Designed a Cascode Amplifier with R-load and observed its transient Analysis ,Drew the LAYOUT and Compared PRE-LAYOT and POST-LAYOUT simulation

8. Current Mirror

Aim: -

To design and simulate a **MOS Current Mirror** using Cadence Virtuoso and analyze its **DC characteristics** as well as **AC characteristics** and the **current copying accuracy**, and **output compliance**.

Technology Used & Tools: -

Technology :

- gpdk90 / gpdk180 CMOS Technology Library
- MOSFET devices: NMOS (nmos1v) and PMOS (pmos1v)

Tools / Software :

- Cadence Virtuoso Schematic Editor
- Analog Design Environment (ADE)
- VMware Linux Environment

Theory:-

A **current mirror** is an analog building block that copies (mirrors) a reference current from one MOSFET to another.
Two identical transistors are used:

- **M1 (Reference transistor)**: A known current is forced through this device.
- **M2 (Output transistor)**: Mirrors (copies) the same drain current as M1.

Key Concepts:

- Both MOSFETs must be **matched** (same W/L).
- Both must operate in **saturation region**.
- The gate and drain of M1 are shorted (diode-connected).
- The gate of M1 drives the gate of M2 → forces equal VGS.
- Since $I_D \propto (W/L)(V_{GS} - V_{TH})^2$, equal VGS gives **equal currents**.

Procedure: -

1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.

- ➢ csh
- ➢ cd cd_work
- ➢ virtuoso &

2. Expand the Log Space you will get a cadence Log Window .

3. Go to file --- > Choose New ------ > Select " Library".

4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.

5. Again click on " new" , now choose the Cell view

5.1 Check Library, create a cell with cell name as "nm_c".

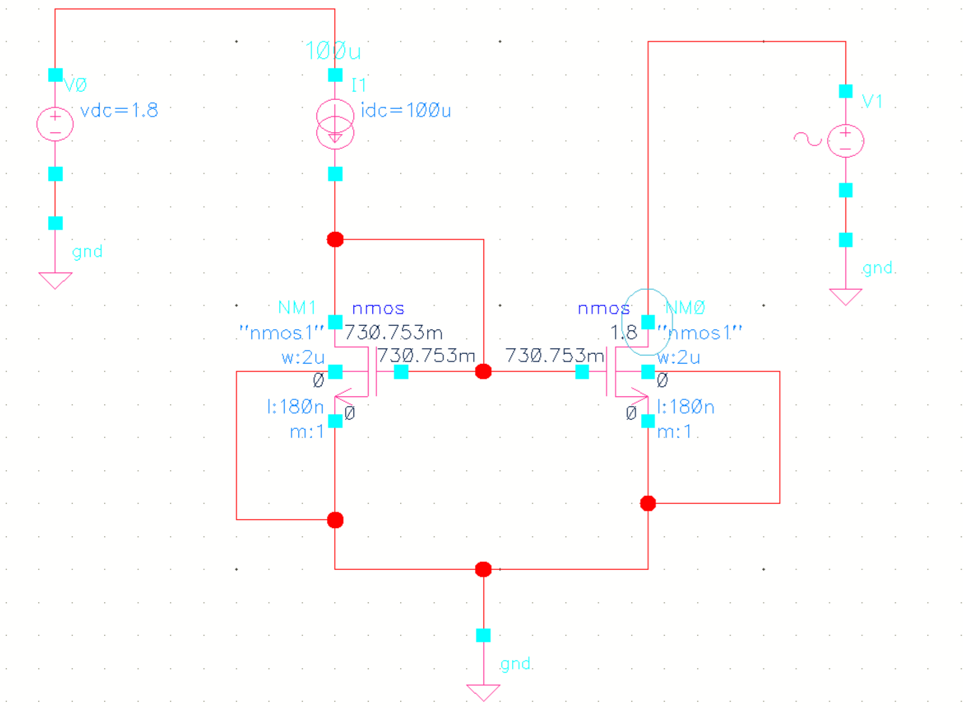
6. Now You got a schematic Editor.

7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.

8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

Schematic Diagram: -



1. DC Analysis – Current Copy Accuracy

Steps:

1. ADE L → Analyses → Choose
2. Select:
   - **dc**
   - Sweep variable: **Component Parameter**
   - Sweep the **VDD** of M2's drain (Output node)
   - Start = **0 V**, Stop = **1.2 V**, Step = **0.01 V**
3. Outputs → Select on Schematic
   - Select **Id(M1)** and **Id(M2)**

2. AC Analysis (Optional)

1 Go to **Analyses → Choose**

2 In the popup window:

- **Analysis Type:** ac
- **Sweep Type:** log
- **Number of Points per Decade:** 10 (or 20 for better resolution)
- **Start Frequency:** 10 Hz
- **Stop Frequency:** 10 MHz

3 Click **OK**

Graph Observations :-

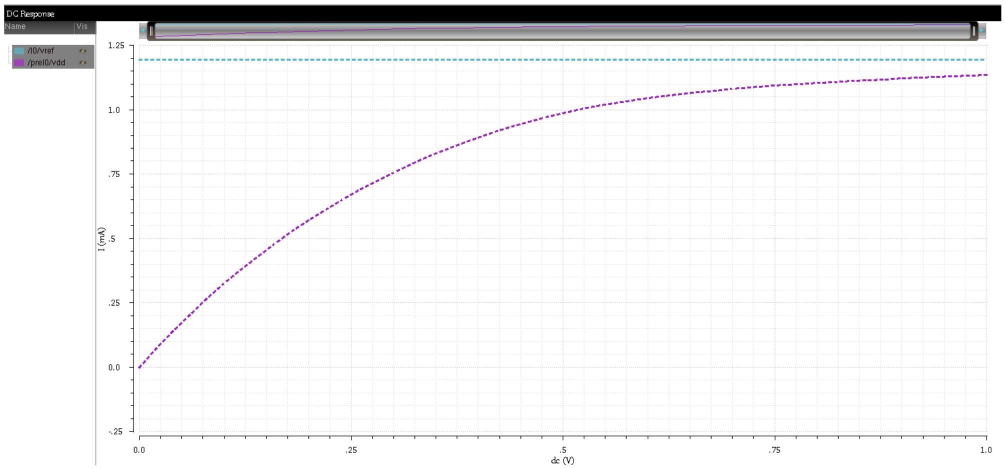
1) DC Sweep Graph Observation

Graph: Plot of **Iout** vs **Vout** (sweeping Vout from 0 to 1.2 V)

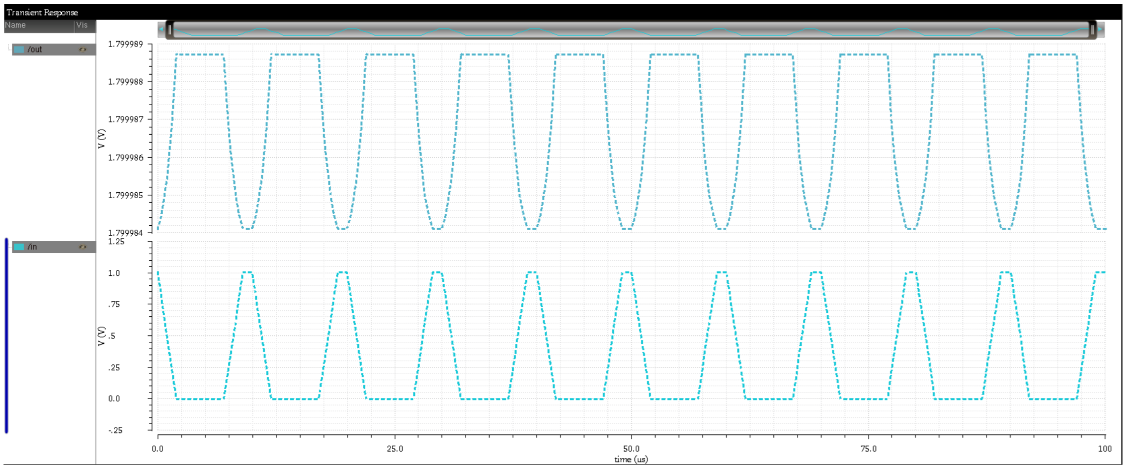
Observation:

- When **Vout is small (< 0.3 V)**, the output transistor is not in saturation → **Iout is very low**.
- After **Vout ≈ 0.3 V** (compliance voltage), **Iout becomes almost constant**, matching the reference current **≈ 50 µA**.

- At higher Vout (0.6 V to 1.2 V), **Iout stays flat**, showing **proper current mirroring**.



- **Similarly do it for AC analysis and Transient Analysis.**



Conclusion from DC Graph:

- **Compliance voltage ≈ 0.3 V**
- **Iout ≈ Iref = 50 µA**
- Mirror works correctly after entering saturation.

2) AC Analysis Graph Observation

Graph: |Iout| (or gain) vs frequency (10 Hz to 10 MHz)

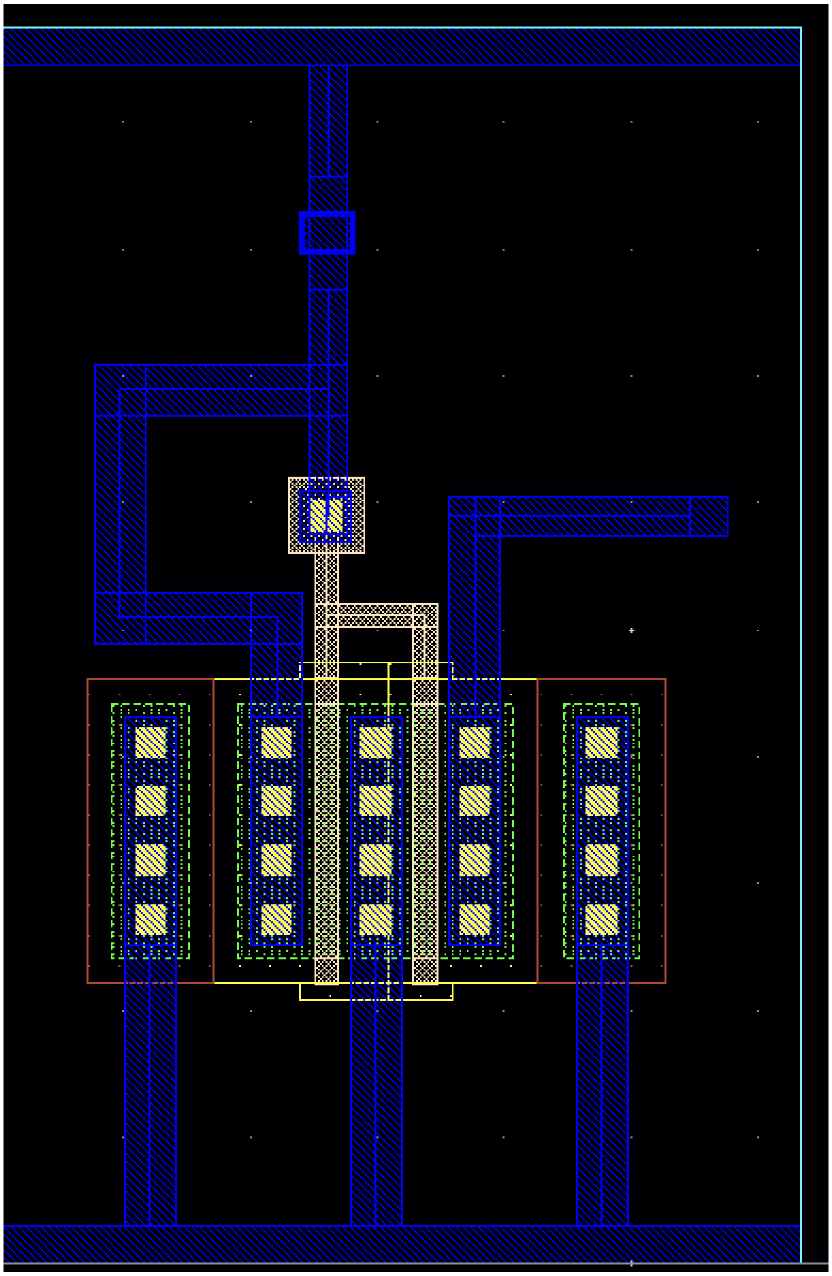
Observation:

- At low frequencies (10 Hz – 10 kHz), **gain ≈ 1 A/A** (0 dB) → perfect mirroring.
- Beyond hundreds of kHz, gain starts to reduce due to MOS capacitances.
- At very high frequencies (above a few MHz), the gain drops.

Conclusion from AC Graph:

- **Low-frequency gain = 1 (ideal mirroring)**
- **High-frequency roll-off** due to device parasitics
- Mirror works best in **low and mid frequencies**.

Layout :



Conclusion

- The current mirror works correctly once the output transistor enters saturation.
- Output current remains constant and equal to the reference current.
- Mirror is accurate at low and mid frequencies and is widely used in analog IC design.

8: Differential Amplifier

Aim: -

To design and simulate a **Differential Amplifier** using Cadence Virtuoso and analyze its **DC characteristics** , AC Characteristics as well as Transient Analysis.

Technology Used & Tools: -

Technology :

- gpdk90 / gpdk180 CMOS Technology Library
- MOSFET devices: NMOS (nmos1v) and PMOS (pmos1v)

Tools / Software :

- Cadence Virtuoso Schematic Editor
- Analog Design Environment (ADE)
- VMware Linux Environment

Theory:-

The differential amplifier is a voltage subtractor circuit which produces an output voltage proportional to the voltage difference of two input signals applied to the inputs of the inverting and non-inverting terminals of an operational amplifier.

This means that a differential amplifier naturally eliminates noise or interference that is present in both input signals.

Key Equations (Small-Signal Analysis):

- Differential Gain (Ad):

$$A_d = \frac{vout}{vin1 - vin2} = -gm\ (R_D \parallel ro)$$

- Common-Mode Gain (A_{cm}):

$$A_{cm}=\frac{RD}{2(RSS\,||ro|)}$$

(where R_{SS} is the output resistance of the tail current source)

Procedure: -

1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - csh
   - cd cd_work
   - virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select “ Library”.
4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on “ OK “ .

   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.
5. Again click on “ new” , now choose the Cell view

   5.1 Check Library, create a cell with cell name as “nm_c”.
6. Now You got a schematic Editor.
7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.
8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

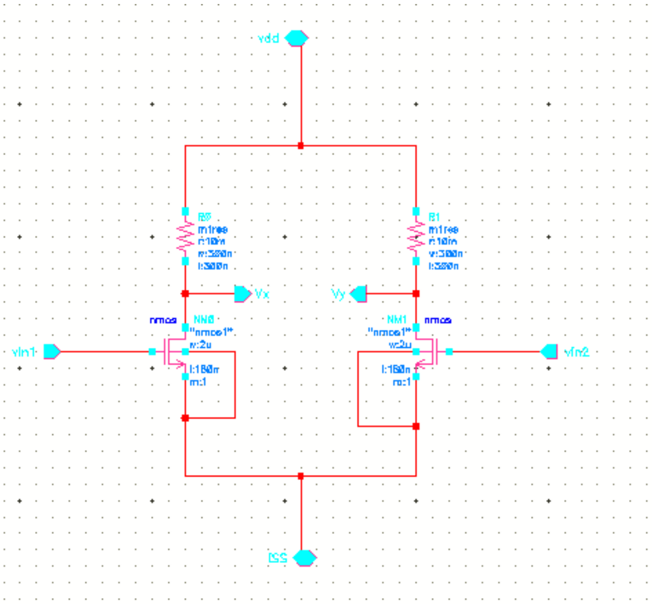
1. DC Analysis – Current Copy Accuracy

Steps:

1. ADE L → Analyses → Choose
2. Select:
   - **dc**
   - Sweep variable: **Component Parameter**
   - Sweep the **VDD** of M2’s drain (Output node)
   - Start = **0 V**, Stop = **1.2 V**, Step = **0.01 V**

3. Outputs → Select on Schematic

    - Select **Id(M1)** and **Id(M2)**

- 

2. AC Analysis (Optional)

1 Go to **Analyses → Choose**

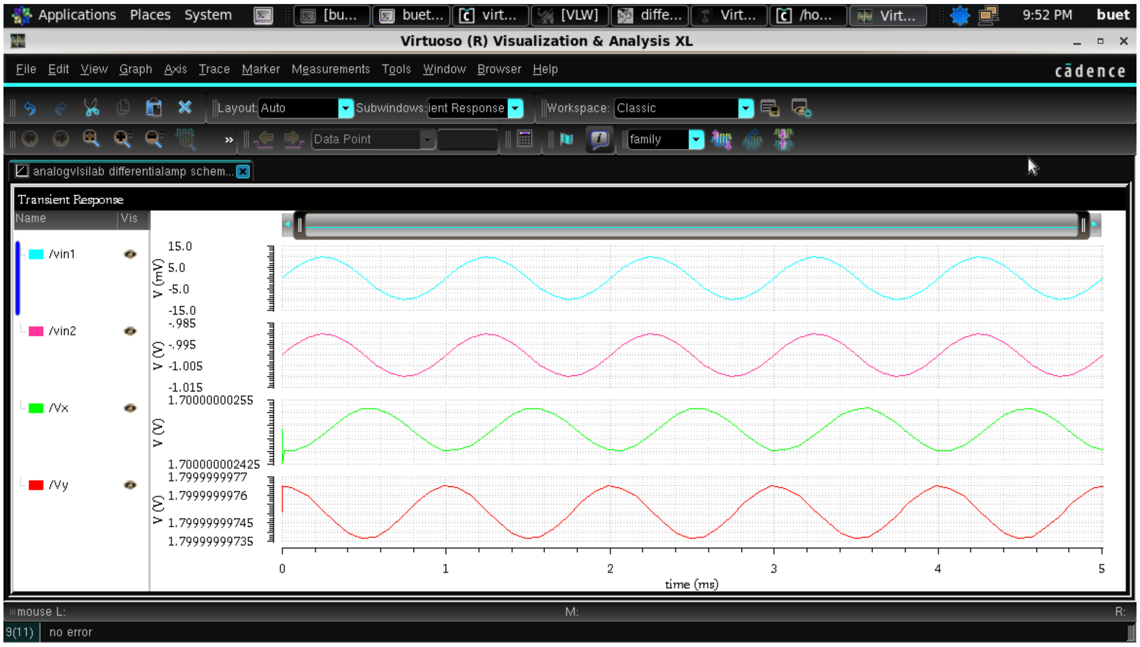
2 In the popup window:

- **Analysis Type:** ac
- **Sweep Type:** log
- **Number of Points per Decade:** 10 (or 20 for better resolution)
- **Start Frequency:** 10 Hz
- **Stop Frequency:** 10 MHz

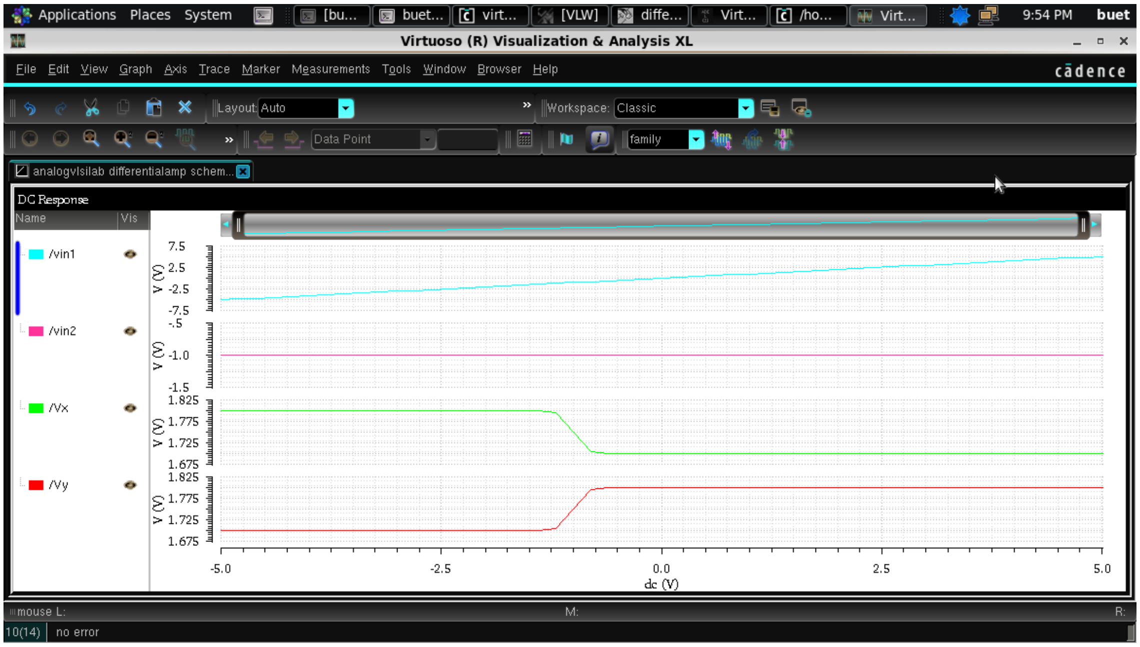
3 Click **OK**

Graph :-

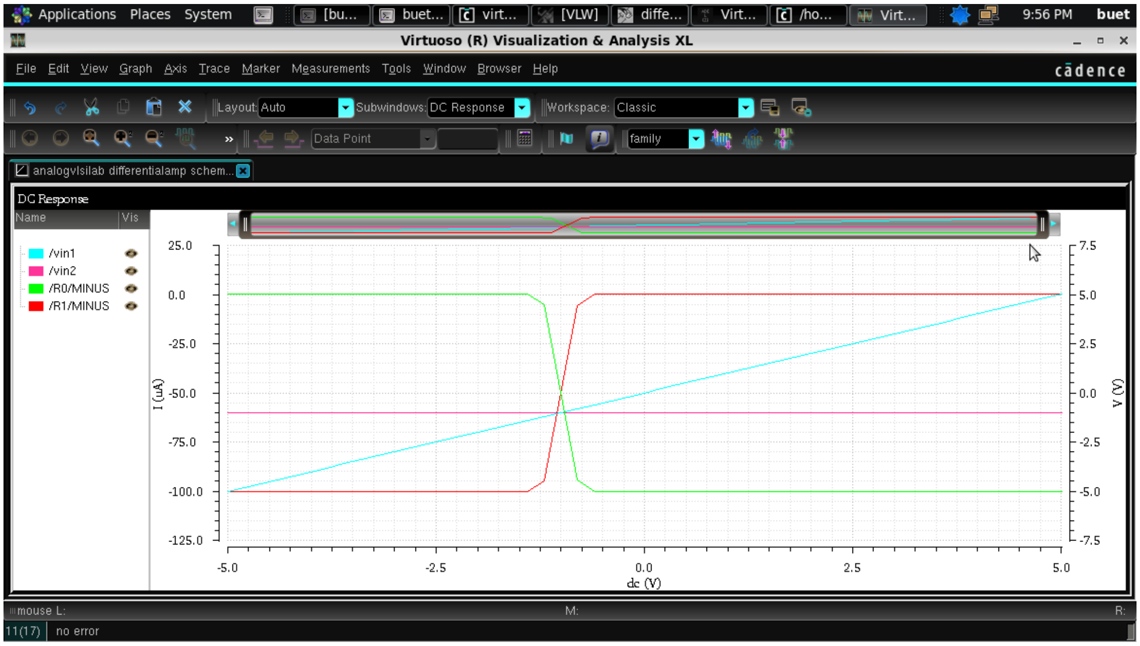
Transient Analysis:-



DC Analysis (Voltage Vs Voltage) :-


Applications Places System
[bu... buet... virt... [VLW] diffe... Virt... /ho... Virt...
9:54 PM buet
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
Layout: Auto
Workspace: Classic
Data Point
family
analogvlsilab differentialamp schem...
DC Response
Name
Vis
/vin1
/vin2
/Vx
/Vy
7.5
2.5
-2.5
-7.5
V (V)
-.5
-1.0
-1.5
V (V)
1.825
1.775
1.725
1.675
V (V)
1.825
1.775
1.725
1.675
V (V)
-5.0
-2.5
0.0
2.5
5.0
dc (V)
mouse L:
M:
R:
10(14)
no error

DC Analysis (Voltage Vs Current) :-



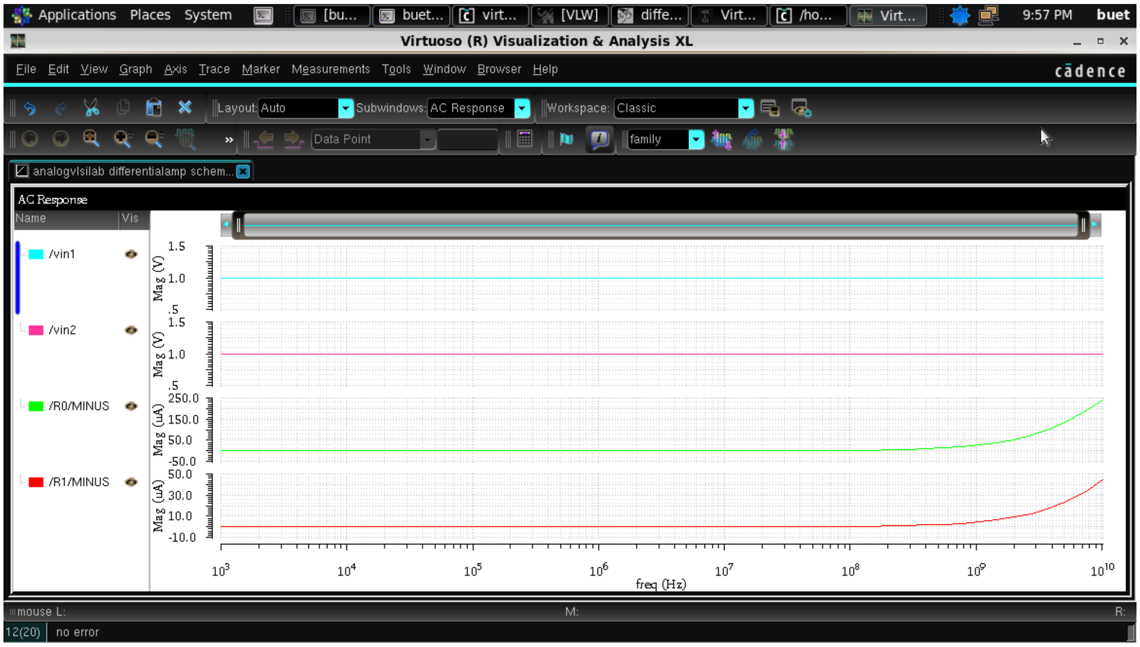
Layout: Auto

Subwindows: DC Response

Workspace: Classic

11(17) no error

AC Analysis (Voltage VS Current) :-



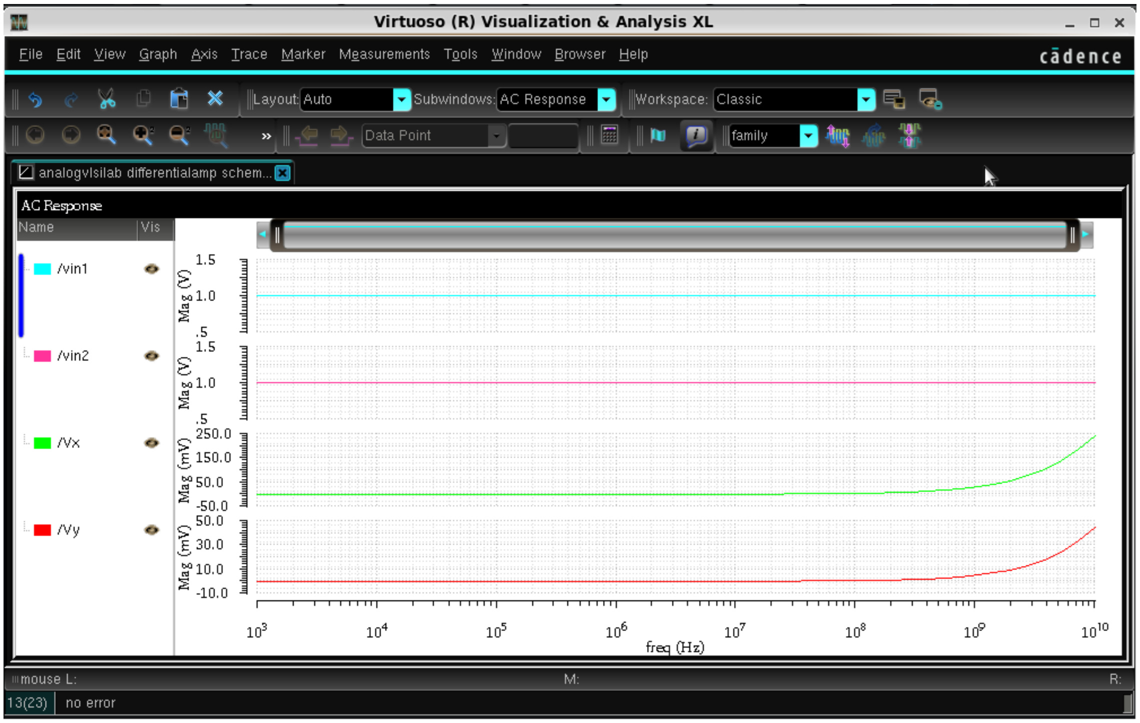
Layout: Auto

Subwindows: AC Response

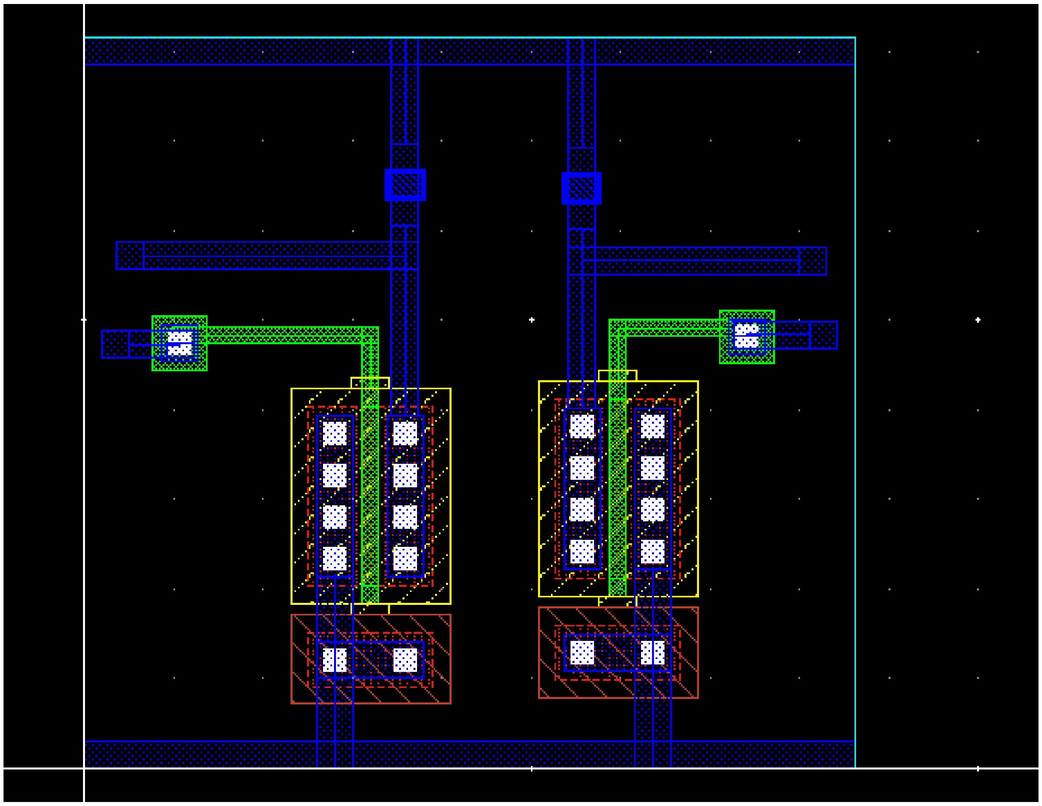
Workspace: Classic

12(20) no error

AC Analysis (Voltage Vs Voltage) :-



Layout :-



Observations:

- When one input increases, the current through that transistor increases while the current through the other decreases.
- The output voltage changes according to the difference between the two input signals.
- Common-mode input changes produce very small output variation, showing good common-mode rejection.
- The differential amplifier provides a high differential gain and low common-mode gain.

Conclusion:

The MOS differential amplifier amplifies the difference between two input signals and rejects common-mode signals. Performance is strongly dependent on:

- Matching of MOS transistors
- Tail current stability
- Load resistance / PMOS load characteristics.

9. Single Stage Operational Amplifier

Aim: -

To design and simulate a **Single Stage Operational Amplifier** using Cadence Virtuoso and analyze its **DC characteristics** , AC Characteristics as well as Transient Analysis.

Technology Used & Tools: -

Technology :

- gpdk90 / gpdk180 CMOS Technology Library
- MOSFET devices: NMOS (nmos1v) and PMOS (pmos1v)

Tools / Software :

- Cadence Virtuoso Schematic Editor
- Analog Design Environment (ADE)
- VMware Linux Environment

Theory :

An Operational Amplifier (Op-Amp) is a high-gain, direct-coupled amplifier widely used in analog electronic circuits for signal conditioning, filtering, and performing mathematical operations such as addition, subtraction, integration, and differentiation. It is typically designed using differential amplifiers, gain stages, and an output buffer to provide high input impedance and low output impedance.

Basic Structure

An Op-Amp generally consists of three main stages:

1. Input Differential Stage
    - Compares two input signals (inverting and non-inverting)
    - Provides high input impedance
    - Ensures excellent Common-Mode Rejection Ratio (CMRR)
2. Gain Stage

   - Provides very high open-loop voltage gain
   - Amplifies the differential signal from the input stage
   - Includes internal compensation for stability
3. Output Stage
   - Delivers high current drive capability
   - Maintains low output impedance for maximum signal transfer

Ideal Characteristics of an Op-Amp

- Infinite open-loop gain (A → ∞)
- Infinite input impedance (Z_in → ∞)
- Zero output impedance (Z_out = 0)
- Infinite bandwidth
- Zero input offset voltage
- Infinite CMRR and slew rate

Although practical Op-Amps cannot fully achieve these ideal parameters, modern CMOS-based designs closely approach them for accurate and stable amplification.

Modes of Operation

An Op-Amp operates mainly in two configurations:

1. Open-Loop Mode
   - Very high gain, used for comparators
   - Not suitable for linear amplification due to instability
2. Closed-Loop Mode
   - Uses feedback (negative feedback is most common)
   - Controls gain, improves stability, bandwidth, linearity, and reduces distortion

Key Performance Parameters

- Open-Loop Gain (A_OL): Determines amplification capability
- Bandwidth (BW): Frequency range over which gain is stable
- Slew Rate (SR): Maximum rate of change of output voltage
- Input Offset Voltage: Small differential voltage required to make output zero

- CMRR: Ability to reject common-mode noise
- Power Supply Rejection Ratio (PSRR): Immunity to supply variations
- Output Swing: Maximum positive and negative output limits

Procedure: -

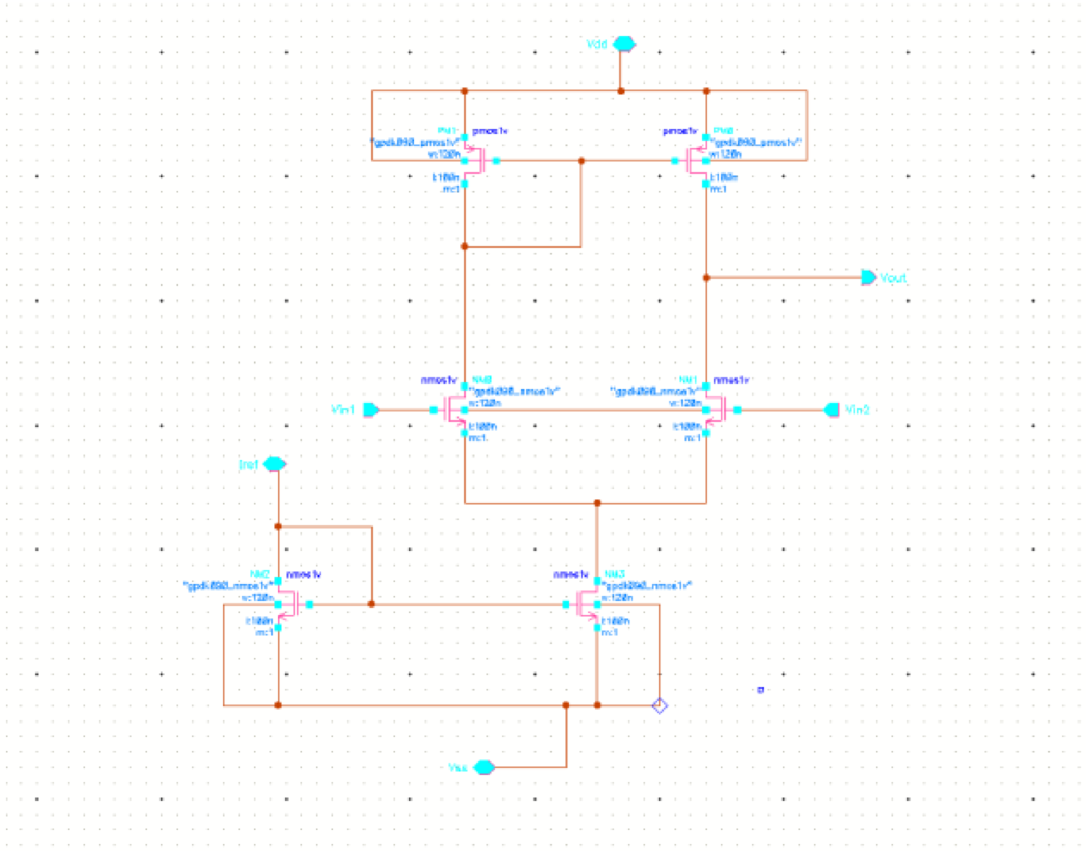
1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - csh
   - cd cd_work
   - virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select " Library".
4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.
5. Again click on " new" , now choose the Cell view

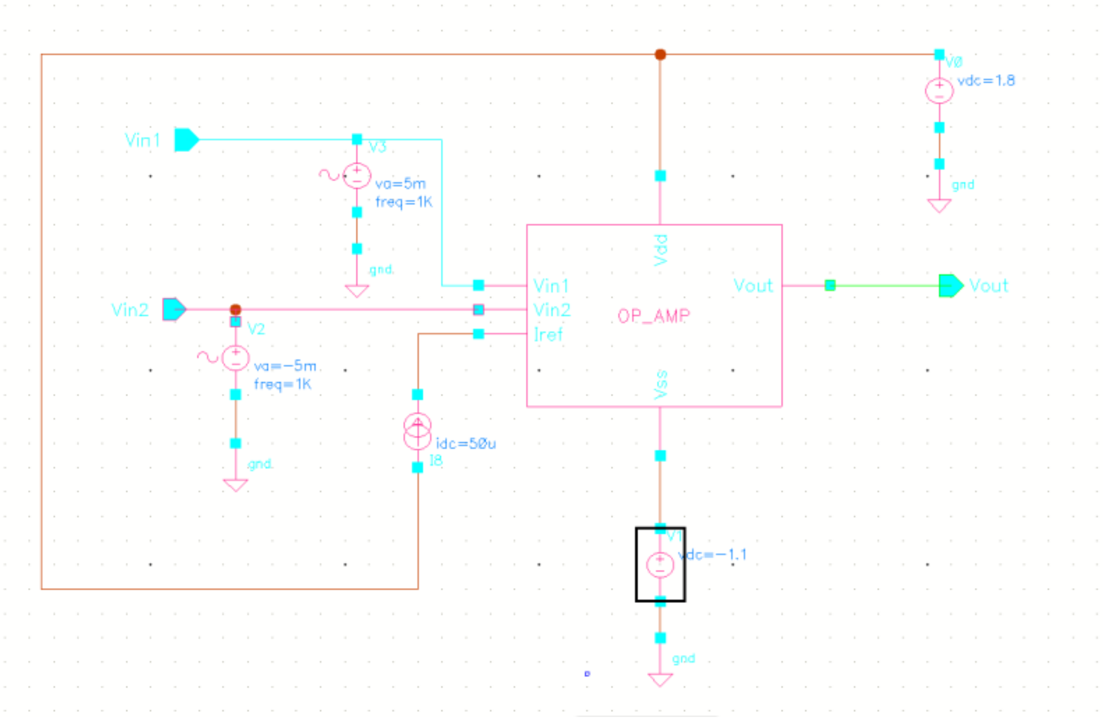
   5.1 Check Library, create a cell with cell name as "nm_c".
6. Now You got a schematic Editor.
7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.
8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.

Circuit Diagram:-



The Symbol:



1. DC Analysis

Steps:

1. ADE L → Analyses → Choose
2. Select:
   - **dc**
   - Sweep variable: **Component Parameter**
   - Choose the **V3 Component**
   - Set the sweep range from Start = **-5 V**, Stop = **5 V**

2. AC Analysis

Steps:

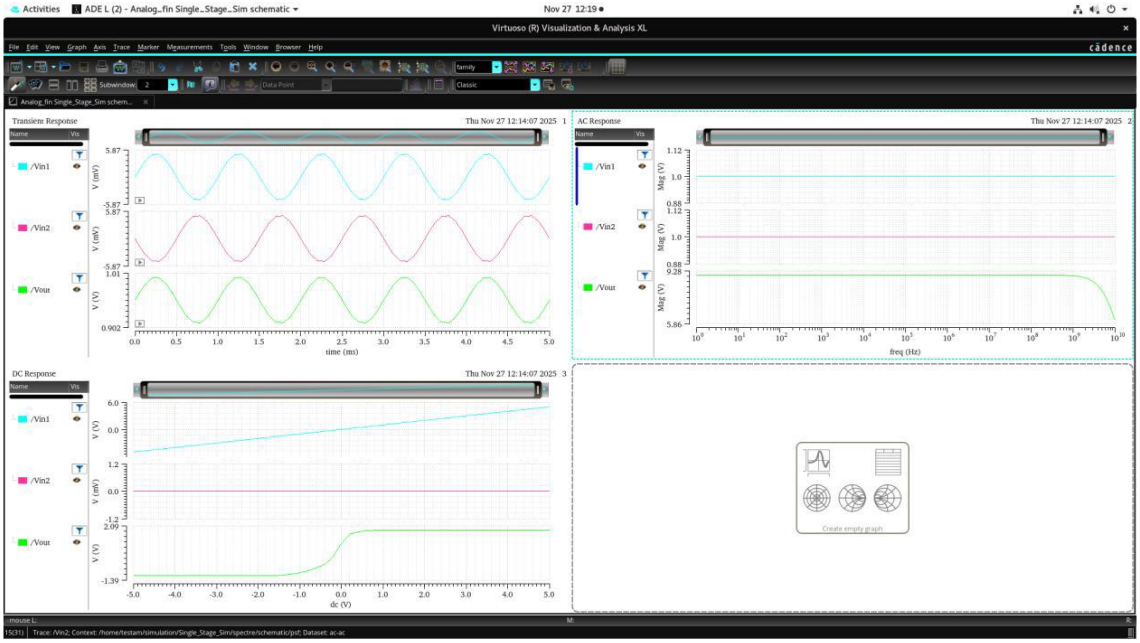
1. ADE L → Analyses → Choose
2. Select:
   - **AC**
   - Sweep variable: **Frequency**
   - Set the sweep range from Start = **1 V**, Stop = **10G**
   - Set Sweep type as Logarithmic

3.Transient Analysis: -

- ADE L → Analyses → tran
- Stop Time: **5 m**

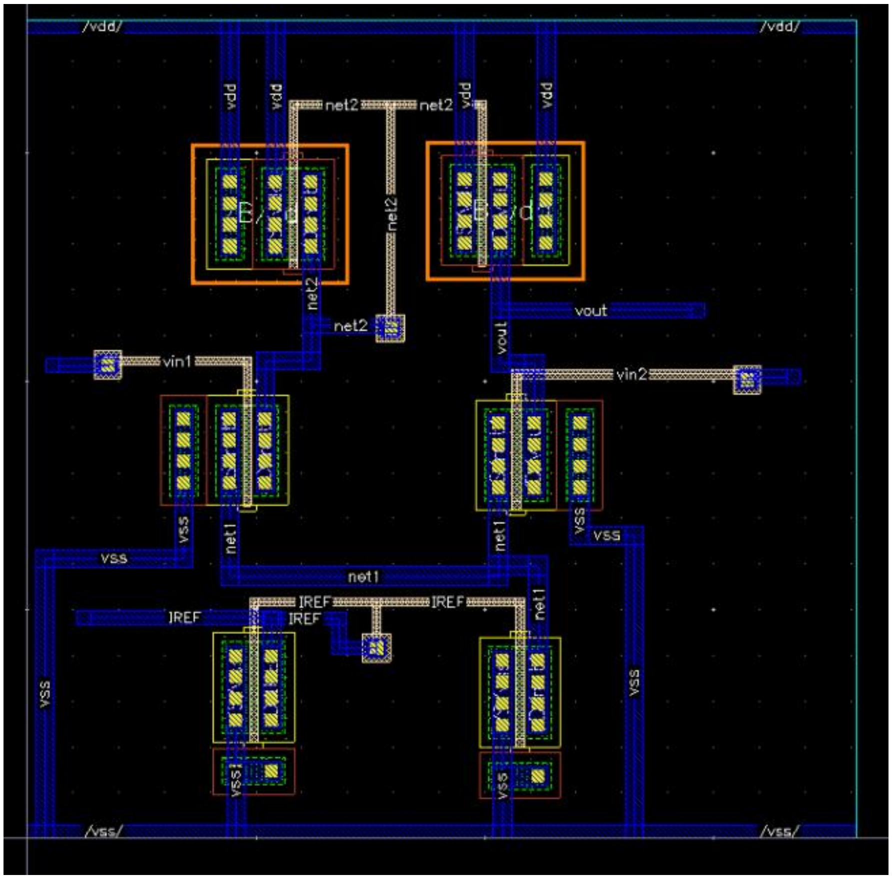
Accuracy: Moderate

Graph :-



15(31) | Trace: /Vin2; Context: /home/testam/simulation/Single_Stage_Sim/spectre/schematic/psf; Dataset: ac-ac

Layout :-



Final Outcome: -

- An Single Stage Operational Amplifier was designed and simulated using Cadence Virtuoso with gpdk90/gpdk180 CMOS technology.

- The full design flow included schematic creation, symbol generation, and circuit connection.

- DC, AC, and Transient simulations were executed using the ADE environment.

- DC analysis confirmed correct transistor biasing and input-output characteristics.

- AC analysis helped evaluate gain, bandwidth, and overall frequency response.

- Transient analysis verified the dynamic behavior and stability of the Op-Amp with time-varying signals.

- A layout was designed to ensure physical implementation while satisfying design rules.

- The experiment provided improved understanding of analog IC design techniques and Op-Amp performance analysis in Cadence.

10. Two Stage Operational Amplifier

Aim: -

To design and simulate a **Two Stage Operational Amplifier** using Cadence Virtuoso and analyze its **DC characteristics**, AC Characteristics as well as Transient Analysis.

Technology Used & Tools: -

Technology :

- gpdk90 / gpdk180 CMOS Technology Library
- MOSFET devices: NMOS (nmos1v) and PMOS (pmos1v)

Tools / Software :

- Cadence Virtuoso Schematic Editor
- Analog Design Environment (ADE)
- VMware Linux Environment

Theory :

An Operational Amplifier (Op-Amp) is a high-gain, direct-coupled amplifier widely used in analog electronic circuits for signal conditioning, filtering, and performing mathematical operations such as addition, subtraction, integration, and differentiation. It is typically designed using differential amplifiers, gain stages, and an output buffer to provide high input impedance and low output impedance.

Basic Structure

An Op-Amp generally consists of three main stages:

4. Input Differential Stage
   - Compares two input signals (inverting and non-inverting)
   - Provides high input impedance
   - Ensures excellent Common-Mode Rejection Ratio (CMRR)
5. Gain Stage

    - Provides very high open-loop voltage gain
    - Amplifies the differential signal from the input stage
    - Includes internal compensation for stability
6. Output Stage
    - Delivers high current drive capability
    - Maintains low output impedance for maximum signal transfer

Ideal Characteristics of an Op-Amp

- Infinite open-loop gain (A → ∞)
- Infinite input impedance (Z_in → ∞)
- Zero output impedance (Z_out = 0)
- Infinite bandwidth
- Zero input offset voltage
- Infinite CMRR and slew rate

Although practical Op-Amps cannot fully achieve these ideal parameters, modern CMOS-based designs closely approach them for accurate and stable amplification.

Modes of Operation

An Op-Amp operates mainly in two configurations:

3. Open-Loop Mode
    - Very high gain, used for comparators
    - Not suitable for linear amplification due to instability
4. Closed-Loop Mode
    - Uses feedback (negative feedback is most common)
    - Controls gain, improves stability, bandwidth, linearity, and reduces distortion

Key Performance Parameters

- Open-Loop Gain (A_OL): Determines amplification capability
- Bandwidth (BW): Frequency range over which gain is stable
- Slew Rate (SR): Maximum rate of change of output voltage
- Input Offset Voltage: Small differential voltage required to make output zero

- CMRR: Ability to reject common-mode noise
- Power Supply Rejection Ratio (PSRR): Immunity to supply variations
- Output Swing: Maximum positive and negative output limits

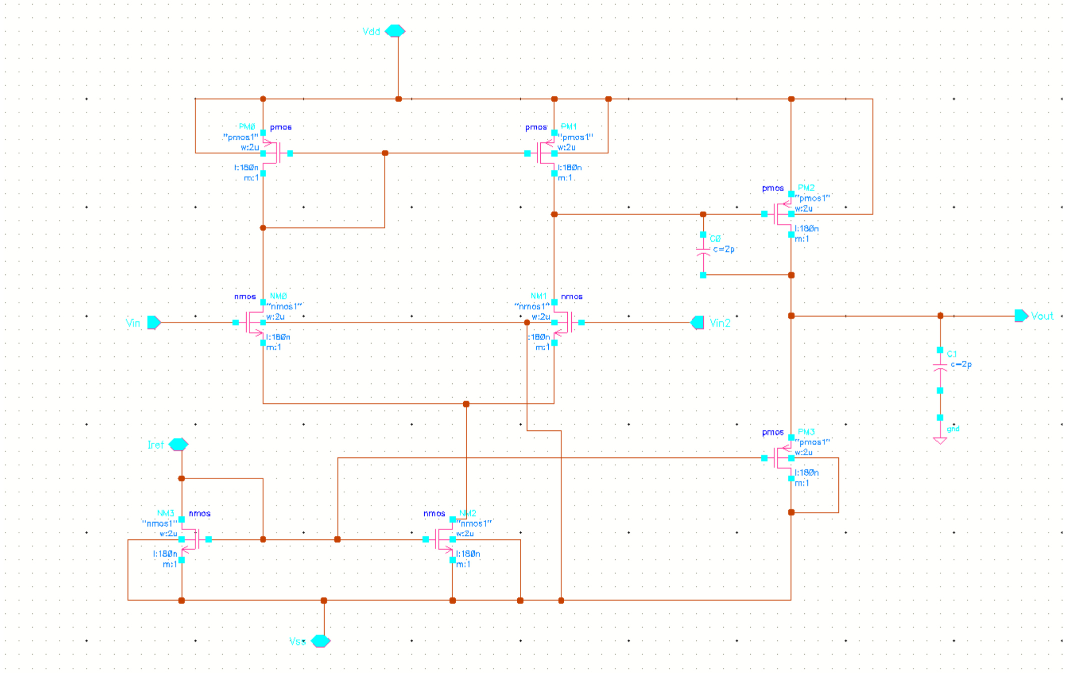
Procedure: -

1. Go to Linux OS and click on the terminal and open the Cadence Log window by using below commands.
   - ➢ csh
   - ➢ cd cd_work
   - ➢ virtuoso &
2. Expand the Log Space you will get a cadence Log Window .
3. Go to file --- > Choose New ------ > Select " Library".
4. 4.1 Provide a name : Pro_1 (Attach to an Existing technological Lib) -> Click on " OK " .

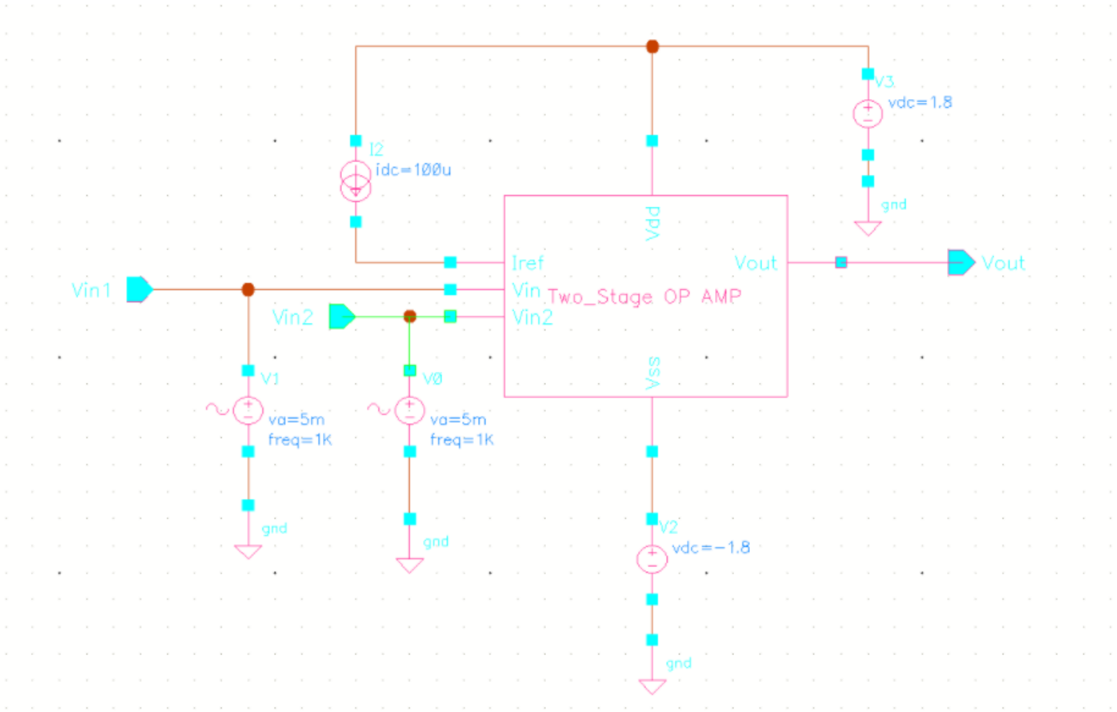
   4.2 A pop Up will Open about the Technology Library -> Click on gpdk90 / gpdk180.
5. Again click on " new" , now choose the Cell view

   5.1 Check Library, create a cell with cell name as "nm_c".
6. Now You got a schematic Editor.
7. Go to Create ->Instance ->Browse lib : gpdk90 / gpdk180 , cell : nmos1v , view: Symbol.
8. Again go to Create -> Instance -> Browse lib : analoglib , cell : vdc , view: Symbol

Like wise Create the circuit elements and connect the circuit as your requirement.



Convert this into a Symbol and add the sources required.



- Provide the Vdd value by using VDC as 1.8 V.
- Provide the IRef value by using IDC as 100 micro-Amp.
- Provide the amplitude of V1 as 5 v and Frequency as 1K and same value for V2 but with 180^0 Phase shift value.
- Provide the Vss value by using VDC as -1.8 V.
- Add a Vout Pin to check the output.

Analysis:-

Open ADE (Analog Design Environment)

Load your schematic and go to:

Launch → ADE L / ADE XL

Select Analysis Type

Go to:

Analyses → Choose…

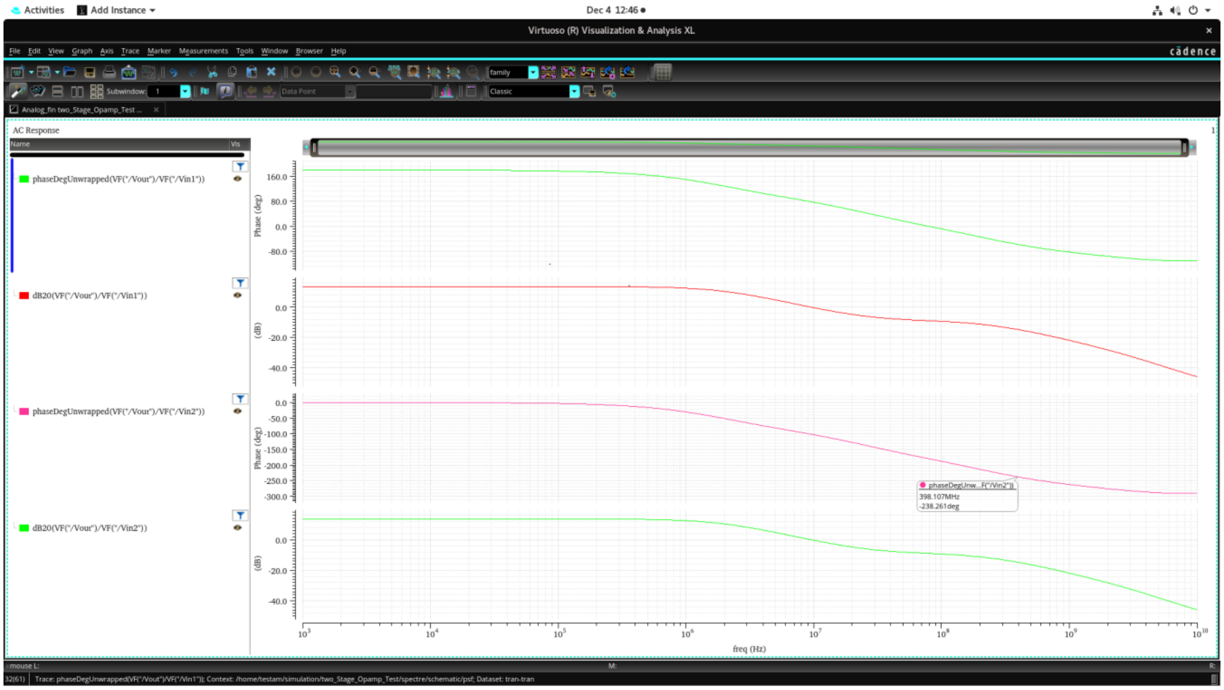
Select **AC** from the list.

Set AC Analysis Parameters

- **Sweep Variable** → Frequency (default)

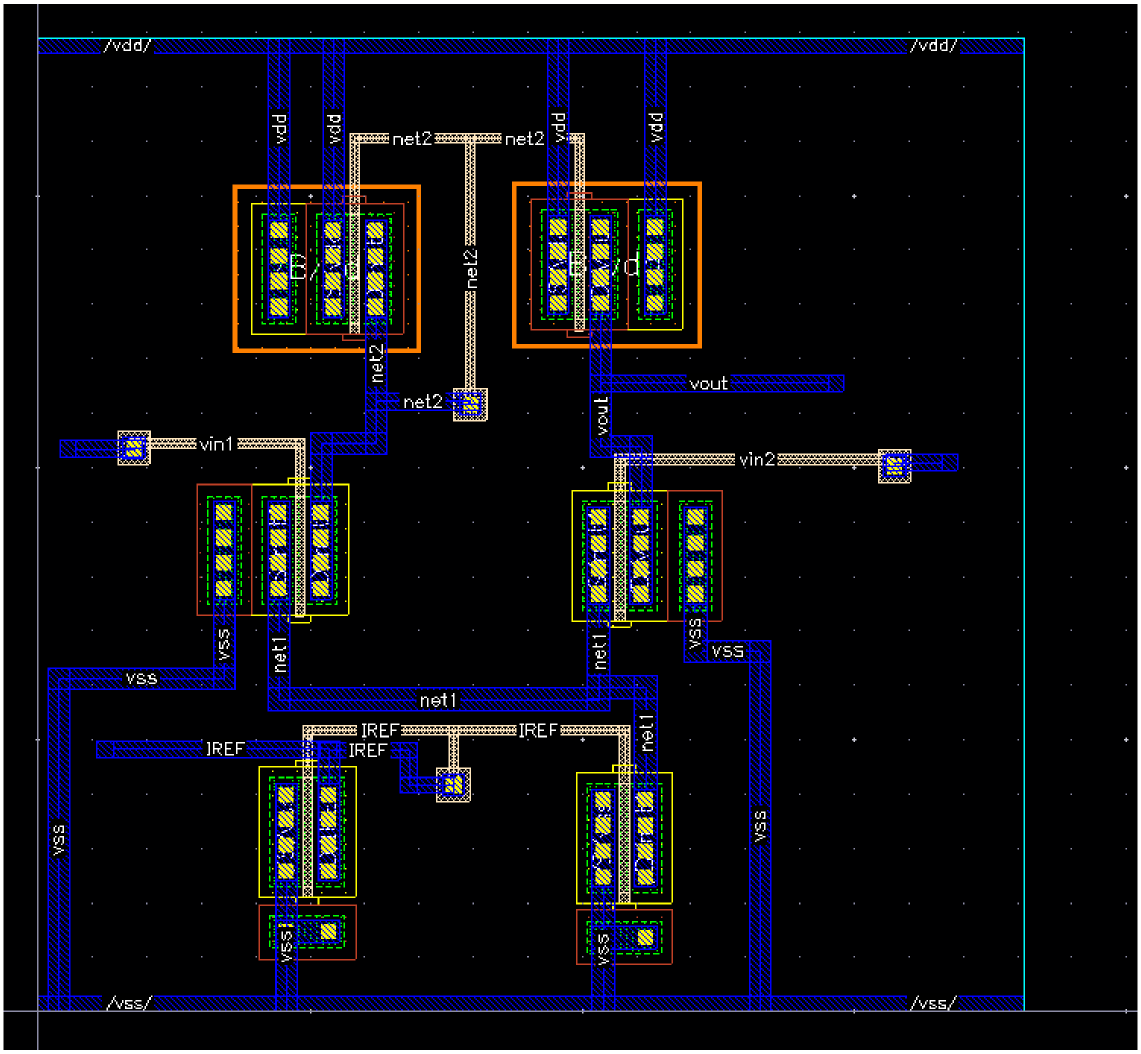
- **Sweep Type** → Linear / Logarithmic
  (Commonly **Logarithmic** for wide range frequency response)
- **Start Frequency** → 1K Hz
- **Stop Frequency** → e.g., 100 M Hz or 1G Hz *(for Op-Amp up to high frequency limits)*
- **Choose Automatic Sweep Type and apply.**
- **Run the values** Then press **OK**.
- **Here our Aim is to check the "Gain vs Frequency and Phase vs Frequency characteristics of the Op-Amp" .**
- For that step go to Results -> Choose the Direct Plot -> AC Gain Phase, To check the output we have to run the AC Analysis and choose the Vin 1 and V-out as well as Vin2 and V-out.
- To check Vin1 Vs Vout and Vin2 Vs Vout Gain vs Frequency and Phase vs Frequency characteristics.

Plot / Graph: -


Activities
Add Instance
Dec 4 12:46
Virtuoso (R) Visualization & Analysis XL
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
family
Subwindow: 1
Data Point
Classic
Analog_fin two_Stage_Opamp_Test ...
AC Response
Name
Vis
🟩 phaseDegUnwrapped(VF("/Vout")/VF("/Vin1"))
🟥 dB20(VF("/Vout")/VF("/Vin1"))
🟪 phaseDegUnwrapped(VF("/Vout")/VF("/Vin2"))
🟩 dB20(VF("/Vout")/VF("/Vin2"))
Phase (deg)
160.0
80.0
0.0
-80.0
(dB)
0.0
-20.0
-40.0
Phase (deg)
0.0
-50.0
-100.0
-150.0
-200.0
-250.0
-300.0
(dB)
0.0
-20.0
-40.0
phaseDegUnw....F("/Vin2"))
398.107MHz
-238.261deg
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}
freq (Hz)
mouse L:
M:
R:

32(61) | Trace: phaseDegUnwrapped(VF("/Vout")/VF("/Vin1")); Context: /home/testam/simulation/two_Stage_Opamp_Test/spectre/schematic/psf; Dataset: tran-tran

Layout: -


/vdd/
/vdd/
vdd
vdd
net2
net2
vdd
vdd
net2
net2
net2
vout
vin1
vout
vin2
vss
net1
net1
vss
vss
vss
net1
IREF
IREF
IREF
IREF
net1
vss
vss
vss
vss
vss
/vss/
/vss/

Final Outcome: -

- A Two Stage Operational Amplifier was designed and simulated using Cadence Virtuoso with gpdk90/gpdk180 CMOS technology.

- The full design flow included schematic creation, symbol generation, and circuit connection.

- AC analysis was executed using the ADE environment.

- The **Gain vs Frequency** and **Phase vs Frequency** characteristics of the Op-Amp

- A layout was designed to ensure physical implementation while satisfying design rules.

- The experiment provided improved understanding of analog IC design techniques and Two Stage Op-Amp performance analysis in Cadence.