Analog Electronics (Project) 4, Assignment 1

Thomas Harley
s1810956

October 21, 2021

The University of Edinburgh
Sanderson Building
Robert Stevenson Road
The King's Buildings
Edinburgh
EH9 3FB

E1 Transistor Parameter Extraction

Plot I_{DS}-V_{DS} plots of NMOS with W=20μm, L=1μm, bulk connected to ground. Do a parametric simulation of the I_D-V_{DS} plot with the NMOS transistor using V_{GS} from 1 to 5 (steps of 1V).

The first step of this exercise is to create a schematic for a basic NMOS circuit we can use to perform this simulation. We import an nmosm4 device from the PRIMLIB library. We connect the Gate and Drain to separate voltage sources, and the Source and Bulk to GND. The schematic is shown in Figure 1.

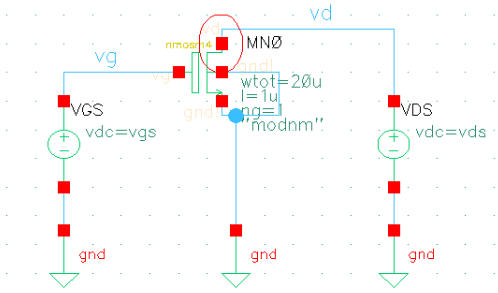

Figure 1: Schematic Diagram for Experiment 1

Now we set up a basic simulation. Launch ADE-L, configure a basic DC simulation, sweep the value of our voltage source V_{DS} from $0V$ to $5V$ in steps of $1V$. This should be a large enough range to observe the saturation region of the NMOS we are simulating. Back in the main ADE-L window we click Outputs > To Be Plotted > Select on Design. Then click on the Drain Node of the NMOS (circled on schematic). This will plot I_{DS}. The values of the two voltage sources have been given variable names 'vds' and 'vgs'. We initially set these to $2V$. Bear in mind that the value of 'vds' is unimportant as we are sweeping V_{DS} from $0V$ to $5V$.

We run the simulation. This will generate a plot of I_{DS} against V_{DS} for $V_{GS} = 2V$ (configured earlier). We can now perform parametric analysis, sweeping V_{GS} from $1V$ to $5V$ in steps of $1V$ generating 5 different traces on the plot as per the question. The result of this parametric simulation is shown below in Figure 2. The figure was generated using MATLAB (source code available in Appendix A)

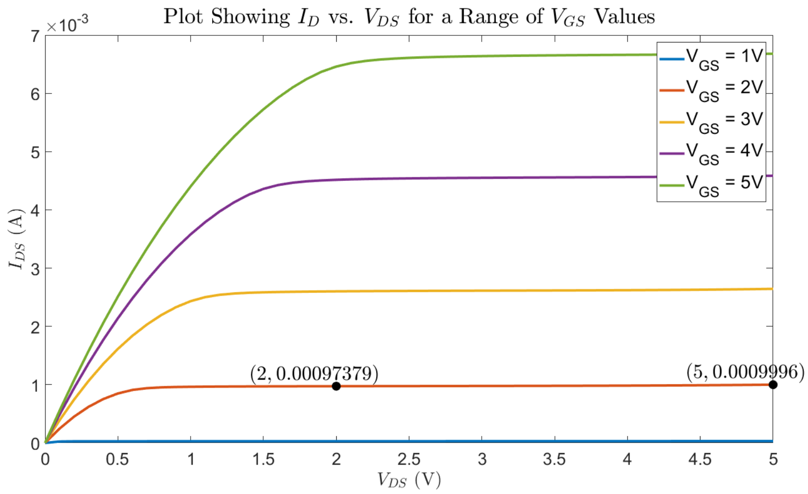

Figure 2: Parametric Simulation of I_{DS} vs. V_{DS} for Various Values of V_{GS}

We can now extract a lot of information from this graph. The first order equation for a transistor in saturation is given (in a simplified form) by:

$$I_{DS} = \beta(V_{GS} - V_t)^2(1 + \lambda V_{DS}) \tag{1}$$

Our goal is to calculate the parameters β and λ. During our DC simulation we found the value of V_t to be $0.8459V$ by annotating the DC operating points on our schematic. Now, considering the NMOS curve where $V_{GS} = 2V$. We can pick two points in the saturation region $(2, 9.7379 \times 10^{-4})$ and $(5, 9.996 \times 10^{-4})$. These points are marked on Figure 1. Subbing the values of these points into Equation 1, along with our value for V_t we can create two simultaneous equations:

$$9.7379 \times 10^{-4} = 1.332\beta + 2(1.332)\beta\lambda \tag{2}$$

$$9.996 \times 10^{-4} = 1.332\beta + 5(1.332)\beta\lambda \tag{3}$$

Multiplying equation 2 by -1 and combining equation 2 with equation 3 we end up with:

$$3.996\beta\lambda = 2.581 \times 10^{-5} \tag{4}$$

Dividing by 3.996 and subbing $\beta\lambda$ back into equation 2 we can calculate that $\beta = 7.1816 \times 10^{-4}$ and (by extension) $\lambda = 8.99521 \times 10^{-3}$.

What is the r_0 of your transistor if you assume $\lambda V_{DS} << 1$. Consider an average I_{DS} for the range you have selected.

We know that r_0 is approximately given by the following equation:

$$r_0 \simeq \frac{1 + \lambda V_{DS}}{\lambda I_{DS}} \tag{5}$$

Therefore, if $\lambda V_{DS} << 1$ then $r_0 \simeq \frac{1}{\lambda I_{DS}}$. We calculate the average value of I_{DS} for the selected range (taking the mean of all the values between $2V$ and $5V$ on our plot). We can use MATLAB to quickly calculate this. We find that the average $I_{DS} = 9.825987 \times 10^{-4}$. Subbing this into our equation we find:

$$r_0 \simeq \frac{1}{\lambda I_{DS}} \simeq \frac{1}{(8.99521 \times 10^{-3})(9.825987 \times 10^{-4})} \simeq 113.139k\Omega \simeq 113k\Omega \tag{6}$$

We should note that this value is not particularly accurate as we are using a heavily simplified model.

Plot I_{DS}-V_{GS} plots of NMOS with W=20μm, L=1μm, bulk connected to ground.

This task requires just a small modification of our original simulation. In our DC Simulation settings we sweep V_{GS} from $0V$ to $5V$ instead of V_{DS}. We set V_{DS} to an arbitrary value of $2V$. We now run the simulation again with these changed settings (no need for parametric analysis as the plots would look almost exactly the same). The plotted result is shown below in Figure 3. The y axis has been changed to a logarithmic scale:

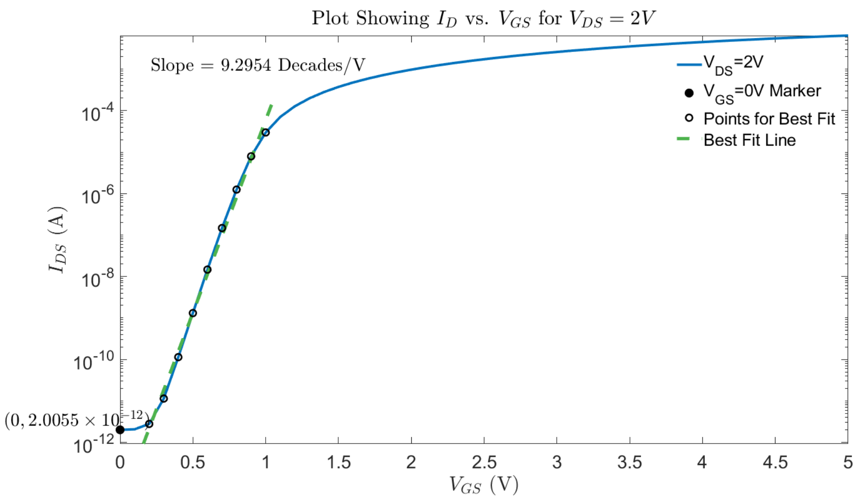

Figure 3: Simulation of I_{DS} vs. V_{GS} for $V_{DS} = 2V$

From this plot, determine the sub-threshold slope (when $V_{GS} < V_t$) of the NMOS transistor and also the OFF current (when $V_{GS} = 0$). What does a non-zero off current mean?

Examining Figure 2, we have plotted a line of best fit in the sub-threshold region using a particular MATLAB Library to fit a straight line to the logarithmic axis. Appendix A contains the source code. This gives us a sub-threshold slope of 9.2954 Decades$/V$. The OFF current is also marked on the graph and is equal to $2.0055 \times 10^{-12}A$. A non-zero off current means that there is some (negligible) **leakage current** coming from the NMOS.

E2 Common Source Amplifier

Plot V_{in} and V_{out} in the time domain. What is the transient gain you achieved in either case? Does it match to the max DC gain, if not why?

For this task we must create a new schematic of a common source amplifier. The same transistor is used as in the previous experiment with W = $20\mu m$ and L = $1\mu m$. $R_D = 100k\Omega$ and we will use a supply voltage V_{DD} of $5V$. We also use an AC input configured with an initial DC_{bias} of $0.9V$, an amplitude of $0.1mV$ and a frequency of $1kHz$. We keep the amplitude small to reduce the chance of distortion on V_{out}. Now we must set up our amplifier for maximum gain. We find the appropriate DC bias value. Set up a basic DC simulation and plot V_{out} against V_{in} sweeping from $0V$ to $5V$. We use the calculator to plot the absolute value of the derivative of V_{out} to find the point at which the gain is maximised. We find that it is approximately $1.04V$. The maximum DC gain we observe when we examine the derivative of V_{out} equals 34.75. We are ready to perform transient analysis. The stop time was set to $6ms$ (6 cycles). The transient plots of V_{in} and V_{out} are shown in Figure 4

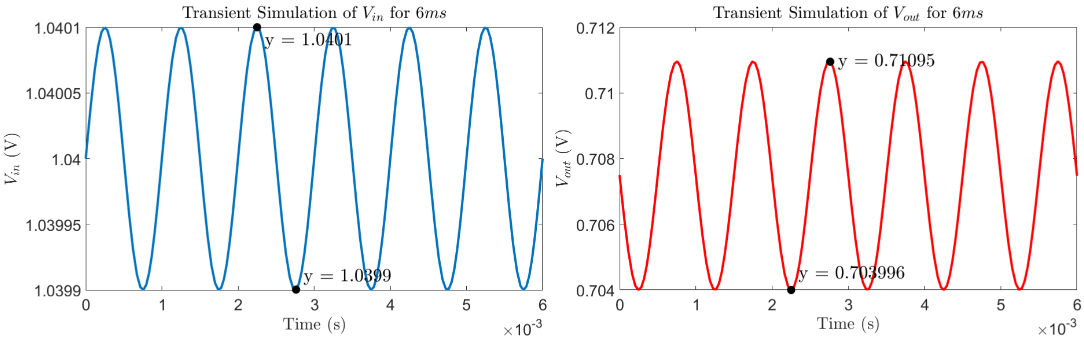

Figure 4: Transient Simulation of V_{out} and V_{in} for Frequency of $1kHz$

As you can see I have plotted the peak voltages for both plots. From this we can calculate peak to peak V_{in} equals $1.0401 - 1.0399 = 0.0002V = 0.2mV$ (as expected since our amplitude is $0.1mV$). Peak to peak V_{out} equals $0.71095 - 0.703996 = 0.006954V = 6.954mV$. Calculating gain:

$$gain(A_V) = \frac{V_{out}}{V_{in}} = \frac{6.954}{0.2} = 34.77 \tag{7}$$

This almost exactly matches the expected max DC gain which is 34.75. This is to be expected as we picked a DC bias point to maximise gain. The small discrepancy that does exist is likely due to measurement error.

If you already have an estimate of r_0 for the transistor, will increasing R_D increase gain further?

Yes, increasing R_D further will increase the gain. The gain of our common source amplifier can be given by the simplified equation: $gain(A_V) = -g_m \frac{R_D r_0}{R_D + r_0}$. Therefore, increasing R_D will always increase the gain. In fact the maximum or "intrinsic gain" of our common source amplifier is given when $R_D = \infty$ and the gain becomes equal to $-g_m r_0$. I partly tested this by running some quick DC simulations with R_D set to $200k\Omega$ and $300k\Omega$ which achieved gains of 45.633 and 51.88 respectfully.

Plot the AC simulation result and indicate the bandwidth. Use DC operating points to measure the power consumption of the circuit.

We must now perform an AC simulation to assess the bandwidth. We make the AC magnitude of our voltage input $1V$ so we can read the magnitude output directly as gain. Now configure the AC simulation in ADE-L. We set the simulation to start at $10Hz$ and stop at $10GHz$. The frequency has already been set to $1kHz$. As per the assignment we set the Points Per Decade to 5. We now run this, simulating the magnitude of V_{out} against frequency (left hand plot of Figure 5). We measure our peak magnitude (and thus gain) to be 34.824. Any discrepancy between this and previous gain values is likely due to measurement errors.

We can use the "bandwidth" function in the Cadence calculator to work out the $3dB$ bandwidth which is $79.76MHz$. The last thing to do is convert this magnitude plot to a bode plot. We can use the Cadence calculator "$db20$" function. This gives the bode plot shown on the right hand side of Figure 5 below. The bandwidth has also been indicated by a vertical line:

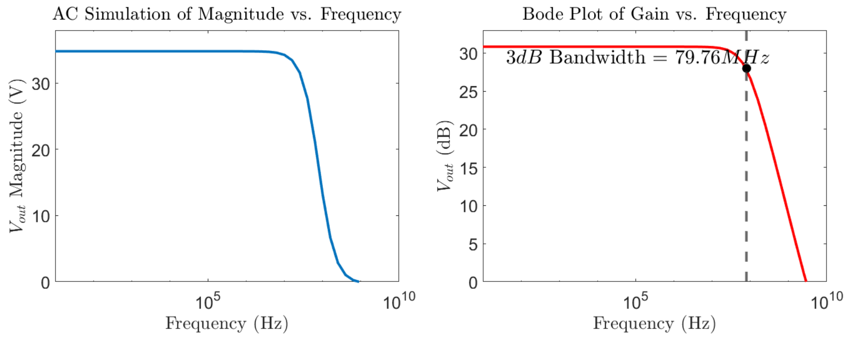

Figure 5: AC Simulation Plots showing V_{out} as a Magnitude and in dB vs. Frequency

The maximum gain in terms of dB is $\simeq 30.8375dB$. The final part of this question is to calculate power consumption. The Results > Print > DC Operating Points tool shows useful data from our circuit including power consumption for the different components. The table below illustrates this. The voltage source V_{in} draws a negligible amount of power. The voltage source V_{DD} draws the necessary power for our resistor (R_D) and transistor (nmosm4). The Current and Voltage is shown as well (as $P = IV$). Note that the power drawn by V_{DD} is equal to the power consumed by $R_D + nmosm4$.

Component	V_{DD}	V_{in}	R_D	nmosm4
Power Drawn/ Consumed	$-213.7\mu W$ $(-42.74\mu A \times 5V)$	$-1.082aW$	$183.5\mu W$	$30.24\mu W$

Table 1: Power Consumption of Common Source Amplifier

Take two frequencies, one within and one outside the bandwidth of the amplifier and perform a transient gain simulation. Indicate whether the transient gain and the AC gain value matches.

The frequency taken within the bandwidth is $0.1MHz$. The frequency taken outside the bandwidth is $100MHz$. The transient plots for both are shown below. The peak values are marked:

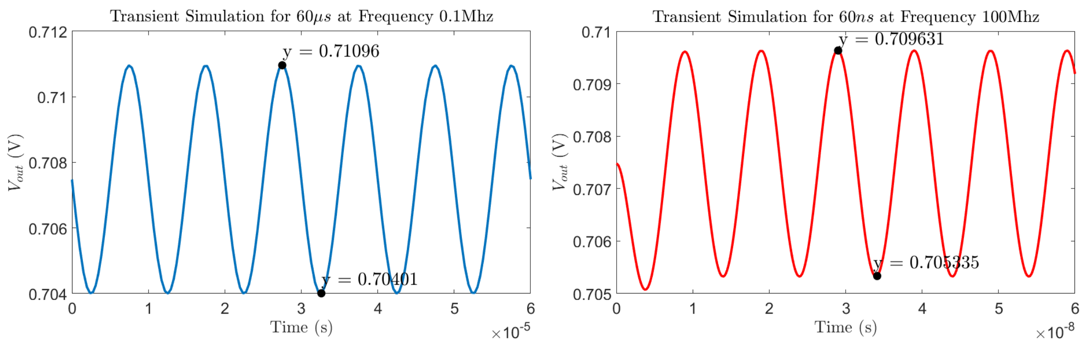

Figure 6: Transient Simulation for Frequencies of $0.1MHz$ and $100MHz$

The peak to peak voltage for our transient simulation at frequency $0.1MHz$ is $0.71096 - 0.70401 = 6.95mV$. Since we are using the same amplitude $V_{in} = 0.2mV$. Doing a quick gain calculation by applying equation 7: $gain = \frac{6.95}{0.2} = 34.75$. This matches the AC gain value from earlier. This is to be expected as $0.1MHz$ lies within the bandwidth. However, the peak to peak voltage at frequency $100MHz$ is $0.709631 - 0.705335 = 4.296mV$. Applying equation 7 again, $gain = \frac{4.296}{0.2} = 21.48$. This does not match the AC gain we calculated earlier which is also to be expected as $100MHz$ lies outside the bandwidth of our Common Source Amplifier.

If you were to measure a real chip with just this circuit on it, do you think the bandwidth of the amplifier would match what you simulated? Why?

No, the bandwidth would not match what was simulated. Real world factors that we cannot simulate will influence the bandwidth if this circuit was printed onto a real chip. Noise from the environment or variations in the manufacturing/ printing process are two potential factors which could effect the real circuit performance.

E3 Current Mirror

Is I_{OUT} matching the expected current, why not? In the simulation, vary I_{REF} $(0-1mA)$ and plot I_{OUT}. If you remove connection between gate and drain of $M1$, would the mirror still work? Why?

For this experiment we create a simple current mirror using two NMOS devices (M1 and M2) with W = $40\mu m$ and L = $1\mu m$. An ideal current source on the M1 branch is set to $100\mu A$ (I_{REF}). V_{DD} is set to an arbitrary value ($5V$) and tied to M1 and M2. We measure $I_{OUT} \simeq 109\mu A$ so it is not matching the expected current. This is because, for this circuit to properly mirror the current, **all** the voltage terminals for both M1 and M2 **must** equal each other. We know that $V_{GS1} = V_{GS2}$ as they are connected together, but V_{DS1} and V_{DS2} do not. Using the DC operating points function in cadence we can see that $V_{DS1} = 1.058V$. Meanwhile $V_{DS2} = 5V$ (from V_{DD}). This difference in drain-source voltages explains why the current is not mirroring properly. The drain source voltage for M1 (V_{DS1}) is influenced by an effect known as "channel length modulation" which is therefore contributing to the discrepancy between V_{DS1} and V_{DS2}. We can actually minimise this effect by lengthening the transistors.

I_{OUT} vs. I_{REF} has been plotted on the left side of Figure 7 below. As you can see they do not quite mirror each other. Removing the connection between the gate and the drain of M1 will break the current mirror. We know for a current mirror to work both transistor devices must be in saturation ($V_{DS} >= V_{GS} - V_t$). While V_{DS1} is connected to V_{GS1} this inequality will hold true and the device will remain saturated, but if we remove this connection the NMOS may fall out of saturation.

Plot V_D vs.I_{OUT} for Figure 2(B). What's the V_{DS} value for which $I_{REF} = I_{OUT}$? Why this value?

We now disconnect the drain of M2 from V_{DD} and give it its own separate voltage source. We sweep this voltage source from $0V - 5V$. The plot of this is shown below on the right hand side of Figure 7. The V_{DS} value for which $I_{REF} = I_{OUT}$ is equal to $1.058V$. This makes sense as this is the drain source voltage for M1 (V_{DS1}). So, when V_{DS2} is set to this voltage they equal each other and the current mirror operates properly.

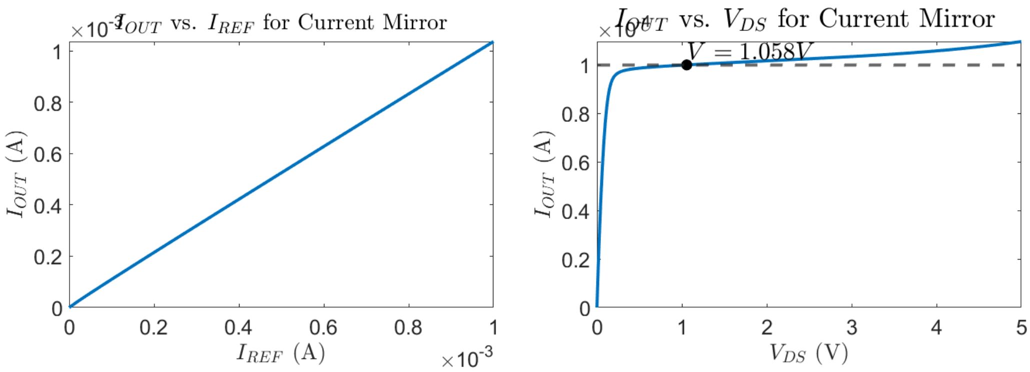

Figure 7: Current Mirror Simulations of I_{OUT} vs. I_{REF} and I_{OUT} vs. V_{DS}

Plot the DC transfer curve and determine the bias point for maximum gain.

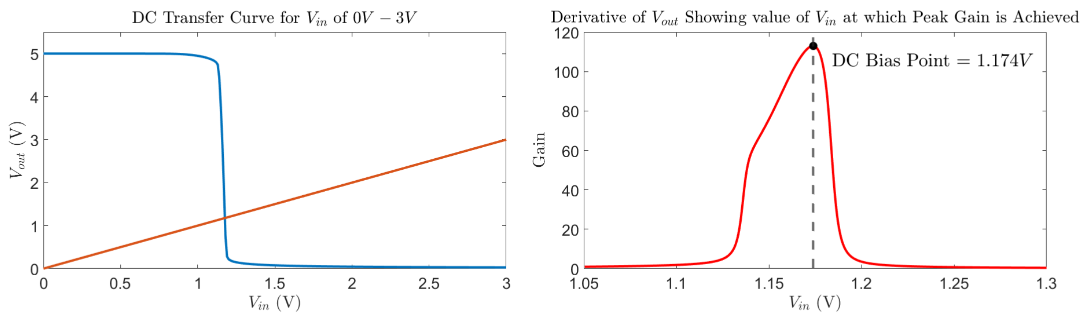

Figure 8: Simulation showing DC Transfer Curve of Circuit and Absolute Derivative of the Curve

In this next part we model a common source amplifier with an active load consisting of a PMOS current mirror. AC input has amplitude = $0.1mV$ and frequency = $1kHz$. Now we find the bias point for maximum gain. We plot the DC transfer curve and then plot the derivative of that to find the point at which maximum gain is achieved. Both plots are shown above in Figure 8. The bias point which achieves maximum gain works out at about $1.174V$ with a maximum gain of $\simeq 113$. We take the absolute value of this derivative to remove any negative caused by 180 degree phase shift.

Plot V_{in} and V_{out} in the time domain. What is the transient gain you achieved and does it match to the max DC gain (gain obtained from DC analysis)?

Now we set the DC bias to $1.174V$. Our transient simulation produces the following plots in Figure 9. Our peak to peak V_{in} is unchanged from E2 and remains $0.2mV$. Our peak to peak value for V_{out} is equal to $1.41942 - 1.397 = 0.02242V = 22.42mV$. Plugging these values into equation 7: $gain = \frac{22.42}{0.2} = 112.1$. The small discrepancy between this value and the measured max DC gain is likely due to measurement error when picking the transient peak points.

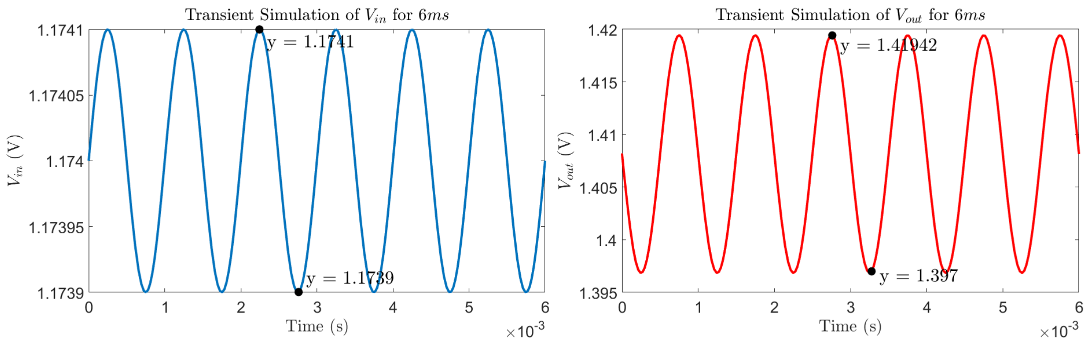

Figure 9: Transient Simulation of V_{out} and V_{in} for frequency $1kHz$

Simulate the circuit to generate a Bode plot from AC simulation. Show that the gain matches with the ones from transient simulation.

We configure an AC simulation with a magnitude of $1V$, start and stop points of $10Hz$ and $10GHz$. Points per Decade set to 10. Running this we simulate the magnitude of V_{out} which corresponds to our gain. We use the cadence calculator "$db20$" function to convert magnitude to dB creating a bode plot at well. Both plots are shown below in Figure 10, with the max gain annotated on the left hand side plot. The max gain according to the AC simulation is $112.998 \simeq 113$. This corresponds to the gain from our transient simulation (tiny discrepancy).

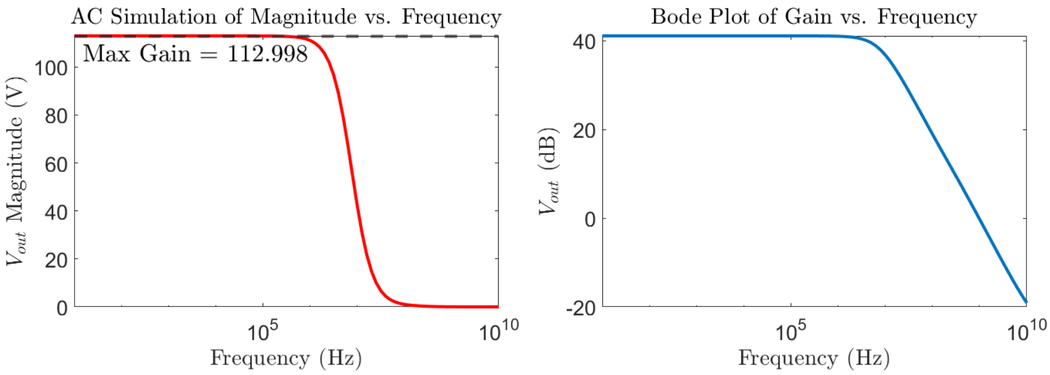

Figure 10: AC Simulation Plots showing V_{out} as a Magnitude and in dB vs. Frequency

Increase the current bias by a factor of 2 and measure the maximum gain from DC analysis. Similarly, halve the current and perform the simulation again. Is the gain changing, why?

Yes the gain does change. When we increase the current bias by a factor of 2 the gain is reduced ($200\mu A$ results in a gain of $\simeq 85$) and when we halve the current bias the gain is increased ($50\mu A$ results in a gain of $\simeq 140$). This is because, according to equation 6 as I_{DS} increases, r_0 of the transistor will go down. Since (as stated previously) $gain = -g_m \frac{R_D r_0}{R_D + r_0}$ as r_0 goes down so will gain. Therefore I_{DS} is inversely proportional to gain. Since I_{REF} sets I_{DS}, I_{REF} is inversely proportional to gain. This explains the behaviour of our simulation.

A MATLAB Source Code

A.1 Figure 2 Source Code

```
set(0,'defaulttextinterpreter','latex')
x = IDVDS.VDS;
y1 = IDVDS.VGS1; y2 = IDVDS.VGS2; y3 = IDVDS.VGS3; y4 = IDVDS.VGS4; y5 = IDVDS.VGS5;

plot(x, y1, 'LineWidth',3.0, 'DisplayName','V_{GS} = 1V')
xlabel('$V_{DS}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
ylabel('$I_{DS}$ (A)', 'FontSize',22)
set(gca,'FontSize',20)
set(gcf,'position',[0,0,1500,1000])
title('Plot Showing $I_{D}$ vs. $V_{DS}$ for a Range of $V_{GS}$ Values', 'FontSize',26)

hold on
plot(x, y2, 'LineWidth',3.0, 'DisplayName','V_{GS} = 2V')
plot(x, y3, 'LineWidth',3.0, 'DisplayName','V_{GS} = 3V')
plot(x, y4, 'LineWidth',3.0, 'DisplayName','V_{GS} = 4V')
plot(x, y5, 'LineWidth',3.0, 'DisplayName','V_{GS} = 5V')
plot(x(52),y2(52),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
plot(x(22),y2(22),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
text(x(22)-(x(22)*0.3), y2(22)+(y2(22)*0.2), '($2, 0.00097379$)', 'FontSize', 24)
text(x(52)-(x(52)*0.12), y2(52)+(y2(52)*0.2), '($5, 0.0009996$)', 'FontSize', 24)
legend('V_{GS} = 1V', 'V_{GS} = 2V', 'V_{GS} = 3V', 'V_{GS} = 4V', 'V_{GS} = 5V', 'FontSize', ...
    22, 'AutoUpdate', 'off')
hold off
```

A.2 Figure 3 Source Code

```
set(0,'defaulttextinterpreter','latex')
x = IDVGS.VGS;
y = IDVGS.VDS2;

semilogy(x, y, 'LineWidth',3.0, 'DisplayName','V_{DS} = 2V')
xlabel('$V_{GS}$ (V)', 'FontSize',20)
set(gca,'FontSize',18)
ylabel('$I_{DS}$ (A)', 'FontSize',20)
set(gca,'FontSize',18)
set(gcf,'position',[0,0,1500,1000])
title('Plot Showing $I_{D}$ vs. $V_{GS}$ for $V_{DS} = 2V$', 'FontSize',24)
index = (x ≥ 0.2) & (x ≤ 1);

hold on
plot(x(2),y(2),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
text(x(2)-(x(3)*8), y(2)+(y(2)*0.6), '($0, 2.0055\times10^{-12}$)', 'FontSize', 22)
text(x(5)-(x(5)*0.3), y(22)+(y(22)*0.2), 'Slope = 9.2954 Decades/V', 'FontSize', 22)
[slope, intercept] = logfit(x(index), y(index), 'logy', 'LineWidth', 4, 'MarkerSize', 8, ...
    'fontsize', 22)
yApprox = (intercept)+(slope)*log10(x(index));
legend('V_{DS}=2V', 'V_{GS}=0V Marker', 'Points for Best Fit', 'Best Fit Line', 'FontSize', 20)
hold off
```

A.3 Figure 4 Source Code

```
set(0,'defaulttextinterpreter','latex')
xVin = VOUTVIN.Vin_X
yVin = VOUTVIN.Vin_Y
xVout = VOUTVIN.Vout_X
yVout = VOUTVIN.Vout_Y
tiledlayout(1,2)
set(gcf,'position',[0,0,2000,600])
% Left plot
nexttile
plot(xVin, yVin, 'LineWidth',3.0)
title('Transient Simulation of $V_{in}$ for $6ms$', 'FontSize',26)
```

```
ylabel('$V_{in}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(xVin(80),yVin(80),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
     'MarkerFaceColor','black')
plot(xVin(68),yVin(68),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
     'MarkerFaceColor','black')
text((xVin(80)+0.0001), (yVin(80)+0.00001), 'y = 1.0399', 'FontSize', 24)
text((xVin(68)+0.0001), (yVin(68)-0.00001), 'y = 1.0401', 'FontSize', 24)
hold off
% Right plot
nexttile
plot(xVout, yVout, 'LineWidth',3.0, 'color', 'red')
title('Transient Simulation of $V_{out}$ for $6ms$', 'FontSize',26)
ylabel('$V_{out}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(xVout(80),yVout(80),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
     'MarkerFaceColor','black')
plot(xVout(68),yVout(68),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
     'MarkerFaceColor','black')
text((xVout(80)+0.0001), (yVout(80)+0.00001), 'y = 0.71095', 'FontSize', 24)
text((xVout(68)+0.0001), (yVout(68)+0.0005), 'y = 0.703996', 'FontSize', 24)
hold off
```

A.4 Figure 5 Source Code

```
set(0,'defaulttextinterpreter','latex')
x1 = ACBODE.freq
y1 = ACBODE.gain
x2 = ACMAG.freq
y2 = ACMAG.gain
tiledlayout(1,2)
set(gcf,'position',[0,0,1500,700])
% Left Plot
nexttile
semilogx(x1,-y1, 'LineWidth', 3.0)
ylim([0 38])
xlim([0 10^10])
title('AC Simulation of Magnitude vs. Frequency', 'FontSize',26)
ylabel('$V_{out}$ Magnitude (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Frequency (Hz)', 'FontSize',22)
set(gca,'FontSize',20)
% Right Plot
nexttile
semilogx(x2,y2, 'LineWidth', 3.0, 'color', 'red')
ylim([0 33])
xlim([0 10^10])
title('Bode Plot of Gain vs. Frequency', 'FontSize',26)
ylabel('$V_{out}$ (dB)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Frequency (Hz)', 'FontSize',22)
set(gca,'FontSize',20)
xline(79760000, 'LineWidth',3.0, 'LineStyle','--')
hold on
text((x2(5)+0.0001), (y2(5)-1.5), '$3dB$ Bandwidth = $79.76MHz$', 'FontSize', 24)
plot(79790000,28,'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
hold off
```

A.5 Figure 6 Source Code

```
set(0,'defaulttextinterpreter','latex')
xVout1 = TRANS1.Vout_X
yVout1 = TRANS1.Vout_Y
```

```
xVout2 = TRANS2.Vout_X
yVout2 = TRANS2.Vout_Y
tiledlayout(1,2)
set(gcf,'position',[0,0,2000,600])
% Left plot
nexttile
plot(xVout1, yVout1, 'LineWidth',3.0)
title('Transient Simulation for $60\mu s$ at Frequency 0.1Mhz', 'FontSize',26)
ylabel('$V_{out}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
xlim([0 6*10^(-5)])
hold on
plot(xVout1(85),yVout1(85),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
plot(xVout1(73),yVout1(73),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
text((xVout1(85)), (yVout1(85)+0.0004), 'y = 0.70401', 'FontSize', 24)
text((xVout1(73)), (yVout1(73)+0.0004), 'y = 0.71096', 'FontSize', 24)
hold off
% Right plot
nexttile
plot(xVout2, yVout2, 'LineWidth',3.0, 'color', 'red')
title('Transient Simulation for $60n s$ at Frequency 100Mhz', 'FontSize',26)
ylabel('$V_{out}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(xVout2(310),yVout2(310),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
plot(xVout2(362),yVout2(362),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
text((xVout2(310)), (yVout2(310)+0.0002), 'y = 0.709631', 'FontSize', 24)
text((xVout2(360)), (yVout2(360)+0.0002), 'y = 0.705335', 'FontSize', 24)
hold off
```

A.6 Figure 7 Source Code

```
set(0,'defaulttextinterpreter','latex')
x = E3MIRROR.V_DS
y = E3MIRROR.I_OUT
x2 = E3IOUT.x
y2 = E3IOUT.y
tiledlayout(1,2)
set(gcf,'position',[0,0,2000,550])
% Left plot
nexttile
plot(x2, y2, 'LineWidth',3.0)
title('$I_{OUT}$ vs. $I_{REF}$', 'FontSize',26)
ylabel('$I_{OUT}$ (A)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('$I_{REF}$ (A)', 'FontSize',22)
set(gca,'FontSize',20)
% Right plot
nexttile
plot(x, y, 'LineWidth',3.0)
xlabel('$V_{DS}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
ylabel('$I_{OUT}$ (A)', 'FontSize',22)
set(gca,'FontSize',20)
set(gcf,'position',[0,0,1500,500])
title('$I_{OUT}$ vs. $V_{DS}$ for a Current Mirror ', 'FontSize',26)
yline(0.0001, 'LineWidth',3.0, 'LineStyle','--')
hold on
text(1.058,(0.0001+0.000005), '$V = 1.058V$', 'FontSize', 24)
plot(1.058,0.0001,'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
hold off
```

A.7 Figure 8 Source Code

```
set(0,'defaulttextinterpreter','latex')
xVout = E3DC.Vout_X; yVout = E3DC.Vout_Y;
xVin = E3DC.Vin_X; yVin = E3DC.Vin_Y
x = E3DCDERIV.X; y = E3DCDERIV.Y
tiledlayout(1,2)
set(gcf,'position',[0,0,2000,800])
% Left plot
nexttile
plot(xVout, yVout, 'LineWidth',3.0)
title('DC Transfer Curve for $V_{in}$ of $0V - 3V$', 'FontSize',26)
ylabel('$V_{out}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('$V_{in}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
ylim([0, 5.5])
hold on
plot(xVin, yVin, 'LineWidth',3.0)
hold off
% Right plot
nexttile
plot(x, y, 'LineWidth',3.0, 'color', 'red')
title('Derivative of $V_{out}$ Showing value of $V_{in}$ at which Peak Gain is Achieved', ...
    'FontSize',26)
ylabel('Gain', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('$V_{in}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(1.174,113,'o', 'MarkerSize',10, 'MarkerEdgeColor','black', 'MarkerFaceColor','black')
text((1.174+0.01),105, 'DC Bias Point = $1.174V$', 'FontSize', 24)
xline(1.174, 'LineWidth',3.0, 'LineStyle','--')
hold off
```

A.8 Figure 9 Source Code

```
set(0,'defaulttextinterpreter','latex')
xVin = E3TRANS.Vin_X; yVin = E3TRANS.Vin_Y;
xVout = E3TRANS.Vout_X; yVout = E3TRANS.Vout_X;
tiledlayout(1,2)
set(gcf,'position',[0,0,2000,600])
% Left plot
nexttile
plot(xVin, yVin, 'LineWidth',3.0)
title('Transient Simulation of $V_{in}$ for $6ms$', 'FontSize',26)
ylabel('$V_{in}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(xVin(80),yVin(80),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
plot(xVin(95),yVin(95),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
text((xVin(80)+0.0001), (yVin(80)-0.00001), 'y = 1.1741', 'FontSize', 24)
text((xVin(95)+0.0001), (yVin(95)+0.00001), 'y = 1.1739', 'FontSize', 24)
hold off
% Right plot
nexttile
plot(xVout, yVout, 'LineWidth',3.0, 'color', 'red')
title('Transient Simulation of $V_{out}$ for $6ms$', 'FontSize',26)
ylabel('$V_{out}$ (V)', 'FontSize',22)
set(gca,'FontSize',20)
xlabel('Time (s)', 'FontSize',22)
set(gca,'FontSize',20)
hold on
plot(xVout(95),yVout(95),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
plot(xVout(110),yVout(110),'o', 'MarkerSize',10, 'MarkerEdgeColor','black', ...
    'MarkerFaceColor','black')
text((xVout(95)+0.0001), (yVout(95)-0.001), 'y = 1.41942', 'FontSize', 24)
text((xVout(110)+0.0001), (yVout(110)-0.0005), 'y = 1.397', 'FontSize', 24)
hold off
```

A.9 Figure 10 Source Code

```
set(0,'defaulttextinterpreter','latex')
x1 = E3ACBODE.freq; y1 = E3ACBODE.gain;
x2 = E3ACMAG.freq; y2 = E3ACMAG.gain;
tiledlayout(1,2)
set(gcf,'position',[0,0,1500,700])
% Left Plot
nexttile
semilogx(x2,-y2, 'LineWidth', 3.0, 'color', 'red')
title('AC Simulation of Magnitude vs. Frequency', 'FontSize',26)
ylabel('$V_{out}$ Magnitude (V)', 'FontSize',22)
xlim([0 10^10])
set(gca,'FontSize',20)
xlabel('Frequency (Hz)', 'FontSize',22)
set(gca,'FontSize',20)
yline(112.998, 'LineWidth',3.0, 'LineStyle','--')
hold on
text(15,105, 'Max Gain = 112.998', 'FontSize', 24)
hold off
% Right Plot
nexttile
semilogx(x1,y1, 'LineWidth', 3.0)
title('Bode Plot of Gain vs. Frequency', 'FontSize',26)
ylabel('$V_{out}$ (dB)', 'FontSize',22)
xlim([0 10^10])
set(gca,'FontSize',20)
xlabel('Frequency (Hz)', 'FontSize',22)
set(gca,'FontSize',20)
```