Analog IC Design – Xschem/Ngspice
Lab 02
Common Source Amplifier

Intended Learning Objectives

In this lab you will:

- Design and simulate a common-source amplifier.
- Learn how to generate and use design charts.
- Investigate gain non-linearity, the variation of the gain with input signal amplitude.
- Study the maximum gain attainable for a resistive-loaded CS amplifier and the effect of supply scaling on max gain.
- Learn how to use feedback to reduce non-linearity (gain linearization).

Part 1: Sizing Charts

1) We would like to design a resistive loaded CS amplifier that meets the specifications below. The design process involves selecting the sizing of the transistor (W and L), the bias point (V_{GS}), and the resistive load (R_D).

Spec	0.13um CMOS	0.18um CMOS
DC Gain	-5	-8
Supply	$1.2V$	$1.8V$
Current consumption	$100\mu A$	$100\mu A$

2) The first design decision is to choose L. Since there is no spec on bandwidth (speed), we may choose a relatively long L to provide large ro and avoid short channel effects. Note that r_o appears in parallel with R_D. Assume we will choose L = 2μm.
3) We can show that the gain is given by

$$|A_v| = g_m R_D = \frac{2I_D}{V_{ov}} * R_D = \frac{2V_{RD}}{V_{ov}}$$

Interestingly, the gain only depends on the voltage drop across V_{RD} and V_{ov}. However, to derive this expression we used $g_m = \frac{2I_D}{V_{ov}}$ which is based on the square-law. For a real MOSFET, if we compute V_{ov} and $\frac{2I_D}{g_m}$ they will not be equal. Let's define a new parameter called V-star (V^*) which is calculated from actual simulation data using the formula

$$V^* = \frac{2I_D}{g_m} \leftrightarrow g_m = \frac{2I_D}{V^*}$$

For a square-law device, V^*=V_{ov}, however, for a real MOSFET they are not equal. The actual gain is now given by

$$|A_v| \approx \frac{2V_{RD}}{V^*}$$

4) The choice of V_{RD} is constrained by the output signal swing. Since we usually want to provide large output swing, we choose the common-mode (CM) output level (DC output level) around $V_{DD}/2$. Thus, although increasing V_{RD} increases the gain, but the choice is limited by the supply voltage which is aggressively scaled down in modern technologies. That's one reason it is difficult to get high gain in modern technologies. Assuming CM output = V_{RD}=$V_{DD}/2$ and given the DC bias current, determine the value of R_D. Again, it is interesting to note that although the gain equals $g_m R_D$, it actually does not depend on R_D itself, but on the voltage drop across it, i.e., the product $I_D \times R_D$.
5) Given A_v and V_{RD}, calculate the required V^* (again note that $V^* \neq V_{ov}$ for a real MOSFET). Let's name this value V^*_Q.
6) The remaining variable in the design is to calculate W. Since the square-law is not accurate, we cannot use it to determine the sizing. Instead, we will use a sizing chart generated from ADT.

Using ADT Sizing Assistant, plot the following design charts vs Vgs for NMOS. Set V_{DS} = $V_{DD}/2$, L = 2u and W=10u. Sweep Vgs from 0 to $\approx V_{TH}+0.4V$

- V^* & V_{ov}(Vgs-Vth) overlaid
- I_D
- gm
- gds

7) You will notice that in the region of moderate inversion, V^* and V_{ov} are relatively close to each other (i.e., square-law is relatively valid). For strong inversion (large Vov) or weak inversion (near-threshold and subthreshold operation) the behavior is quite far from the square-law (although we are using L=2μm).
8) On the V^* and V_{ov} chart locate the point at which V^*= V^*_Q. Find the corresponding V_{ovQ} and V_{GSQ}.
9) Plot I_D, g_m, and g_{ds} vs V_{GS}. Find their values at V_{GSQ}. Let's name these values I_{DX}, g_{mX}, and g_{dsX}.
10) Now back to the assumption that we made that W=10μm. This is not the actual value that we will use for our design. But the good news is that ID is always proportional to W irrespective of the operating region and the model of the MOSFET (regardless square-law is valid or no). Thus, we can use ratio and proportion (cross-multiplication) to determine the correct width at which the current will be I_{DQ}=100μA as given in the specs. Calculate W as shown below

W	I_D
$10\mu m$	I_{DX} @V^*_Q (from the chart)
?	$I_{DQ} = 100\mu A$ (from the specs)

11) Now we are almost done with the design of the amplifier. Note that gm is also proportional to W as long as V_{ov} is constant. On the other hand, r_o = 1/g_{ds} is **inversely** proportional to W (I_D) as long as L is constant. Before leaving this part, calculate g_{mQ} and g_{dsQ} using ratio and proportion (cross-multiplication) and double check that A_v = $-g_m(R_D||r_o)$ meet the required gain spec.

Part 2: CS Amplifier

1. OP and AC Analysis

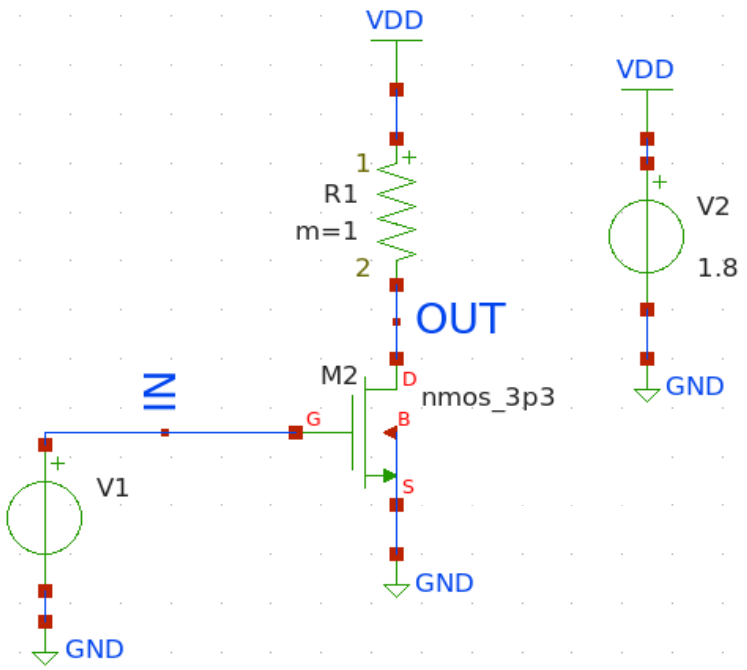
1) Create a new work directory.
2) In this lab we will deal with transistors from GF180 PDK. Copy the file "/home/tare/XschemForGF/xschemrc" to your work directory. This file initializes xschem for usage with GF180.
3) Open the terminal in your work directory. Run the commands below in the terminal:
4) export PDK_ROOT=/home/tare/pdk
5) export PDK=gf180mcuC
6) Open xschem and create a new schematic from the terminal (ignore the warning that this file doesn't exist)"
   xschem cs.sch
7) Include the model of Transistor as shown below. Write down this code inside "code shown.sym" block.

```
name=Models  only_toplevel=false
value="
.include /home/tare/pdk/gf180mcuC/libs.tech/ngspice/design.ngspice
.lib /home/tare/pdk/gf180mcuC/libs.tech/ngspice/sm141064.ngspice typical"
```

8) Inside the "waveform reload launcher". Place the following command

```
descr="load waves (press ctrl + left click)"
tclcommand="xschem raw_read $netlist_dir/[file tail [file rootname [xschem get current_name]]].raw"
```

9) Create a testbench for the resistive loaded CS amplifier using the V_{GSQ},R_D,L, and W that you got from the previous part.



10) To simulate the DC OP, we will add “code_shown.sym” block which contain the following commands

```
.option savecurrents
.control
 save all
 save i(@m.xm1.m0[id])
 save @m.xm1.m0[gm]
 save @m.xm1.m0[gds]
 save v(@m.xm1.m0[vgs])
 save v(@m.xm1.m0[vth])
 save v(@m.xm1.m0[vds])
 save v(@m.xm1.m0[vdsat])
 op
 let gm= @m.xm1.m0[gm]
 let ro = 1/@m.xm1.m0[gds]
 let Ao = gm * ro
 let Av = - gm * X
 remzerovec
 write lab2.raw
 show m : gm : gds : id : vgs : vth : vds : vdsat
.endc
```

Notes:

- **m.xm1.m0[id] : you should replace m1 with your own transistor name**
- **Replace X with the value of R_D obtained from Part1**
- **The commands of other “simulations” should be placed inside the “control” loop as shown above**

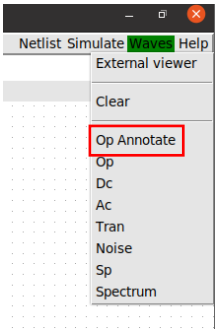
11) We will use “ngspice_get_value.sym” from devices to print the DC OP parameters which is defined as follows.

```
name=r4 node=i(@m.xm1.m0[id])
descr="Id="
```

“node” field should be replaced by the parameter you wish to print (you could get this syntax from the above code).
“descr” is just referring to display name.

12) Simulate the DC OP. Report a snapshot for the key operating point (OP) parameters. Compare the results with the results you obtained in Part1.

To Show the Annotation, you should load the file in which you write your OP from the “Op Annotate” option as shown below.



13) Compare r_o and R_D. Is the assumption of ignoring r_o justified in this case? Do you expect the error to remain the same if we use min L?
14) Calculate the intrinsic gain of the transistor.
15) Calculate the amplifier gain analytically. What is the relation (≪,<,≈,>,≫) between the amplifier gain and the intrinsic gain?

16) Create a new simulation configuration and run AC analysis (from 1Hz to 1GHz). Report the gain vs frequency. Annotate the DC gain and make sure it meets the spec.

```
name=V1 value="dc X ac 1"
savecurrent=true
```

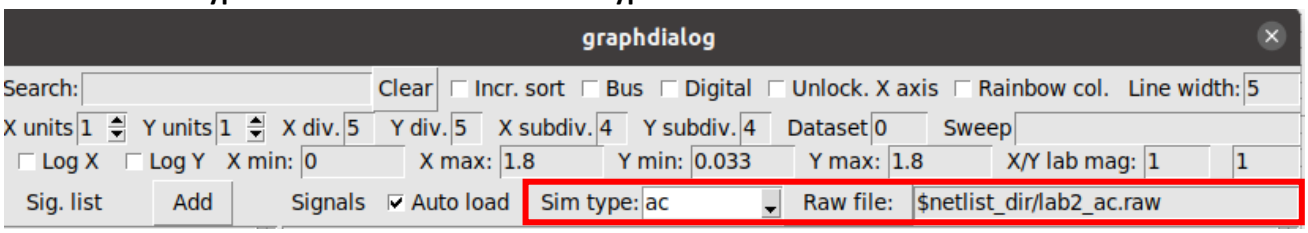
Source setup

```
ac dec 10 1 1G
remzerovec
write lab2_ac.raw
```

code to place inside code block

Notes:

- **Inside the "graphdialog", use "logX"**
- **Inside the "graphdialog" , Load the file in which you write your simulation results from "Raw file" and choose the type of the simulation from "Sim type".**



- **To create Vertical Cursor, use the hotkey "A"**

2. Gain Non-linearity

1) Create a new simulation configuration. Perform a DC sweep for the input voltage from 0 to V_{DD} with 2mV step.

```
dc V1 0 1.8 2m
save deriv(v(out))
let gain= deriv(v(out))
remzerovec
write lab2_dc_sweep.raw
```

2) Report VOUT vs VIN. Is the relation linear? Why?
3) Calculate the derivative of VOUT using calculator. Plot the derivative vs VIN. The derivative is itself the small signal gain. Is the gain linear (independent of the input)? Why?
4) Set the properties of the voltage source to apply a transient stimulus (sine wave of 1kHz frequency and 10mV amplitude superimposed on the DC input voltage).

```
name=V1 value="sin(X 10m 1k)"
savecurrent=true
```

Replace X with the value of V_{gs} obtained from Part1

5) Create a new simulation configuration. Run transient simulation for 2ms. Plot gm vs time. Does gm vary with the input signal? What does that mean?

```
tran 1u 2ms
remzerovec
write lab2_tran.raw
```

6) Is this amplifier linear? Comment.

Lab Summary

In Part 1 you learned:

- How to generate and use design charts for NMOS and PMOS transistors.
- How to design a resistive-loaded common-source amplifier.
- How the overdrive voltage of a MOS transistor deviates from the square law in different regions of operation.

In Part 2 you learned:

- How to do ac and DC simulations of a CS amplifier.
- How the gain of an amplifier changes with the input signal amplitude.
- How to get the maximum attainable gain of a CS amplifier by changing the load resistor.
- How to use feedback resistors to reduce gain non-linearity.

Acknowledgements

Thanks to all who contributed to these labs. Special thanks to Dr. Sameh A. Ibrahim for reviewing and editing the labs. If you find any errors or have suggestions concerning these labs, contact Hesham.omran@eng.asu.edu.eg.