Schematic Capture and Spice Simulation Tutorial

Introduction:

Welcome to C2S2's Analog Sub-Team! The intent of this tutorial is for you to familiarize yourself with the *xschem* schematic editor as well as *Ngspice*, a Spice circuit simulator - both of which are open-source tools. By the end of this tutorial, you will have successfully captured the schematic of an inverter and performed transient analysis.

Before you dive into this tutorial, please familiarize yourself with how to navigate a Linux operating system. Your work on this team will primarily be done on our team's *ecelinux* server. If you need a refresher on Linux commands or a crash course on Linux, please refer to the excellent tutorial developed for ECE 2400 and ECE 4750 students:

https://cornell-ece4750.github.io/ece4750-sec01-linux/

Once you have done so, please SSH into our server:

c2s2-dev.ece.cornell.edu

Now that you have logged into our server, enter the following command on the command line to source the set-up script:

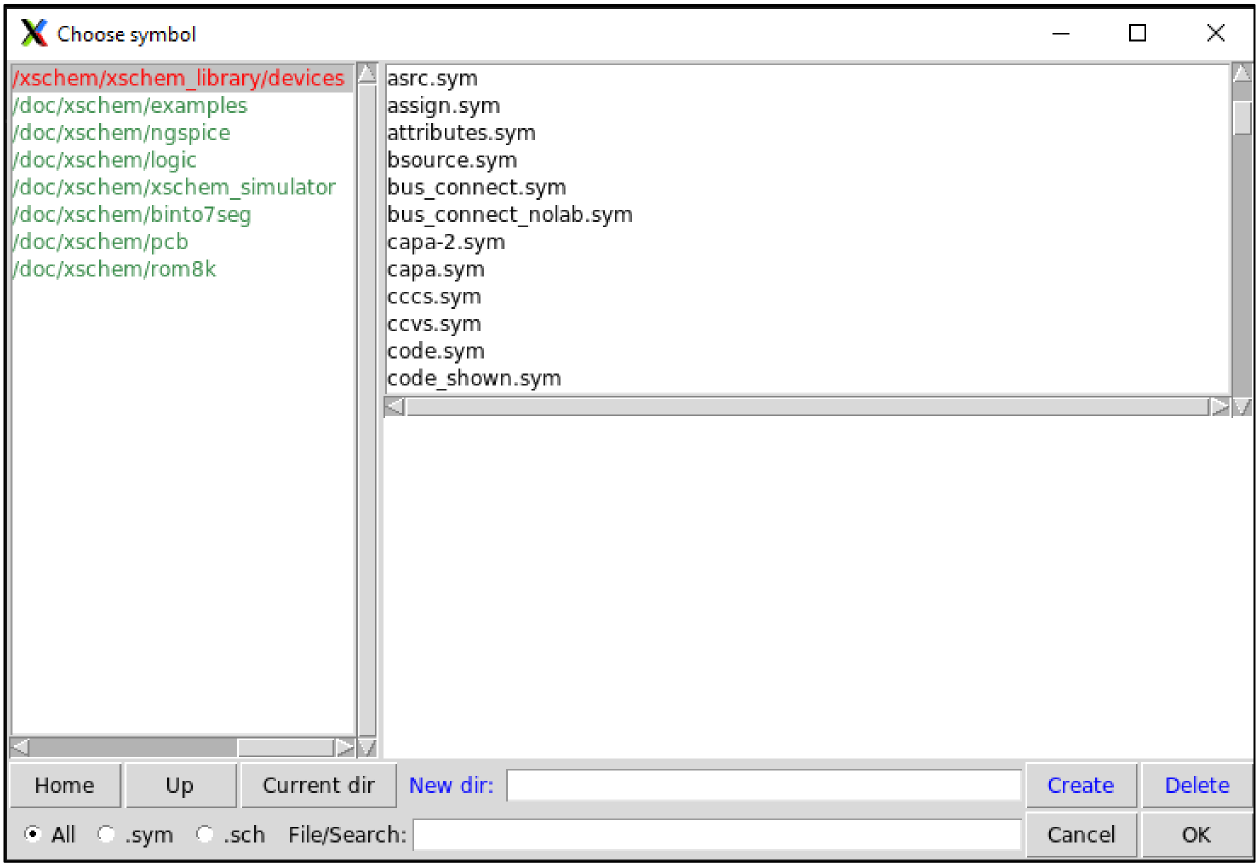
```
source setup-c2s2.sh
```

Please note that you should source this set-up script each time you log into the C2S2 server. For the sake of convenience, you may want to consider placing the above command within your *.bashrc* file. Furthermore, you should remove any pre-existing source set-up commands from your *.bashrc* (upon removal, exit the server, log back in, and source the C2S2 set-up script).

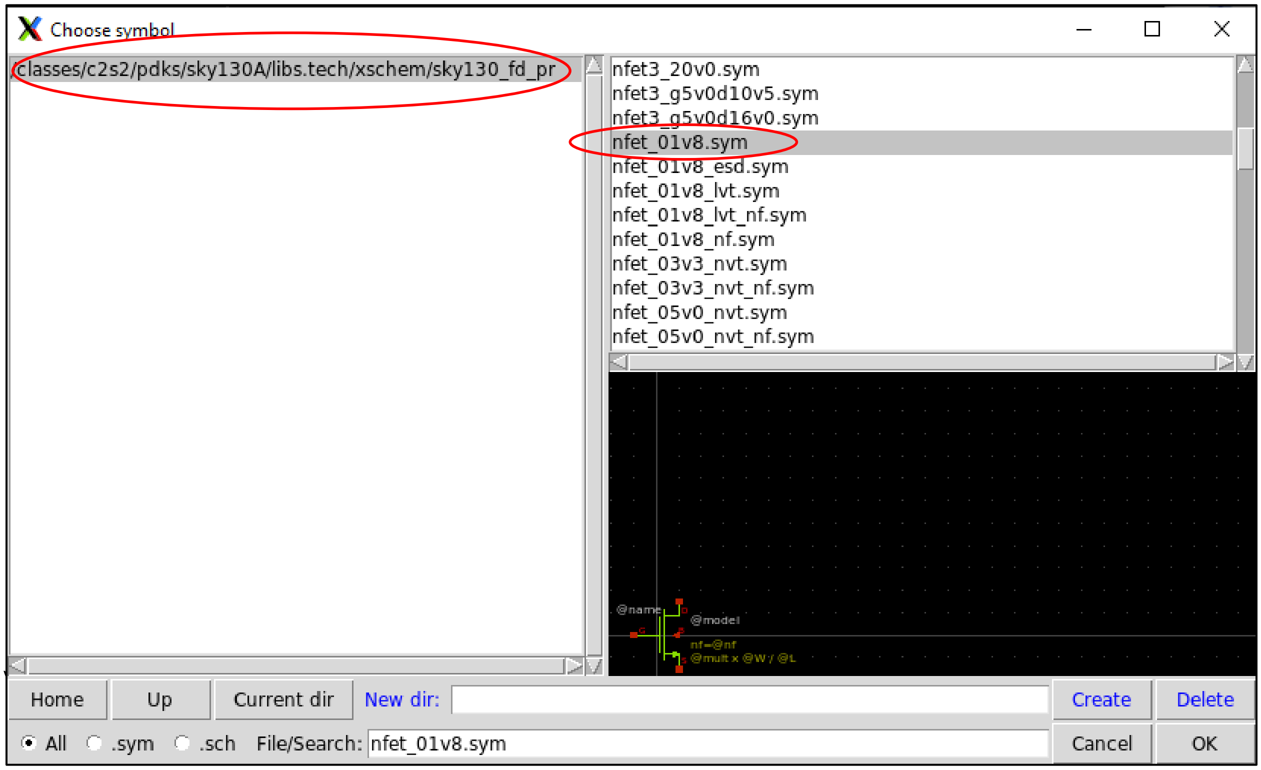
Now you can simply enter the following command to launch *xschem*: xschem

Once you have launched *xschem*, navigate to the top-lefthand corner of the window and click *file*, then *New Schematic*. You should save this file (click *file*, then *Save as)* within a personal directory of your choice; for now, make a directory within your user-specific folder and save the schematic within this directory. I highly suggest frequently saving your work in the case the server or the program itself crashes.

Now, we will populate the schematic with the components necessary for building and simulating the converter. First, press *Shift+i* on your keyboard to pull up a window that shows you your selection of available components (you can also click *Tools* then *Insert symbol*). The window should look like this:

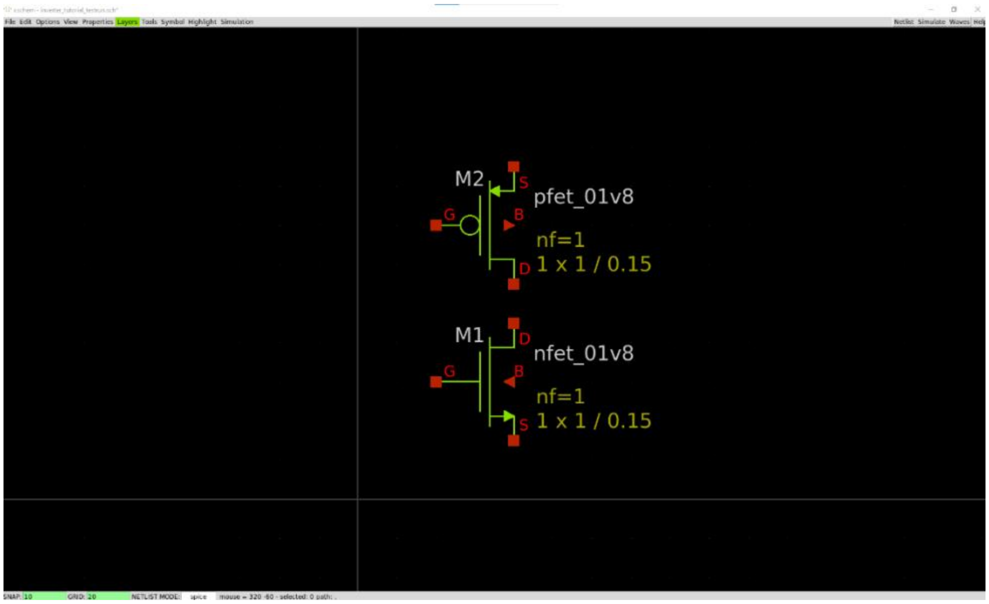


Once this window has appeared, navigate to the following directory, and select *nfet_01v8*, which is a 1.8V NMOS device:

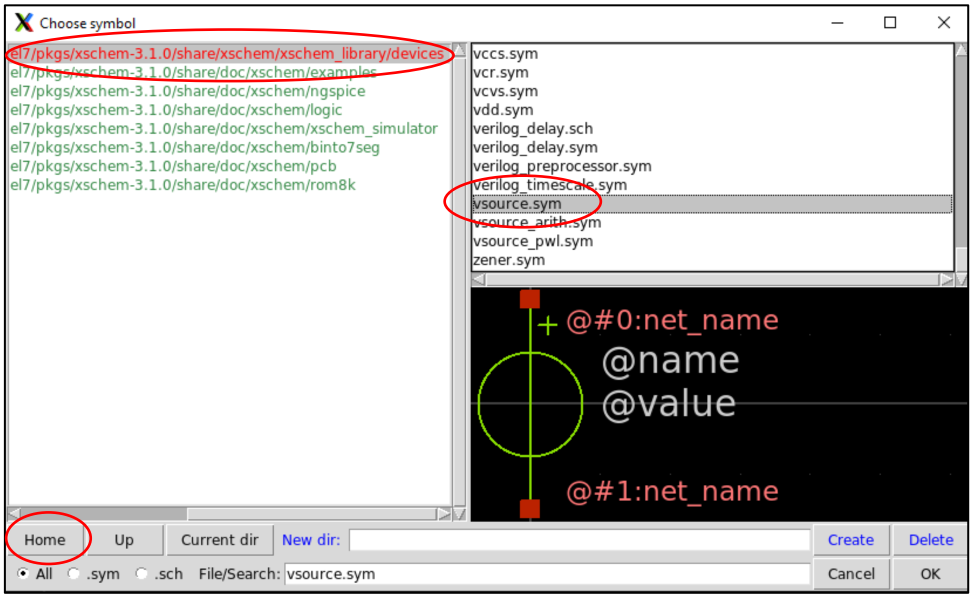


File/Search: nfet_01v8.sym

Once you have done so, press OK and place the component on your schematic. Repeat this process and place a pfet_01v8 on the schematic as well (a 1.8V PMOS device from the same library). If you want to move components around, click on the component to select it and then press M on your keyboard; finally, left click on the location where you wish to place the component. You can select multiple components at once by pressing down on your mouse's left-click button whilst dragging your cursor across the region of the screen that contains the components of interest. Your schematic should now look something like this:


File Edit Options View Properties Layers Tools Symbol Highlight Simulation
Netlist Simulate Waves Help
M2
S
G
B
D
pfet_01v8
nf=1
1 x 1 / 0.15
M1
D
G
B
S
nfet_01v8
nf=1
1 x 1 / 0.15
SNAP: 10
GRID: 20
NETLIST MODE: spice
mouse = 320 -60 - selected: 0 path: .

Now you will populate the schematic with a DC voltage source as well as a square-wave voltage source (one will act as VDD for your inverter and the other will act as the input to your circuit): *vsource.sym* and *sqwsource.sym* respectively. To find these components, click on *Home* and then the first path available (the text color should be red). This library contains ideal sources and symbols that will hold your Spice directives, which are necessary for simulation:

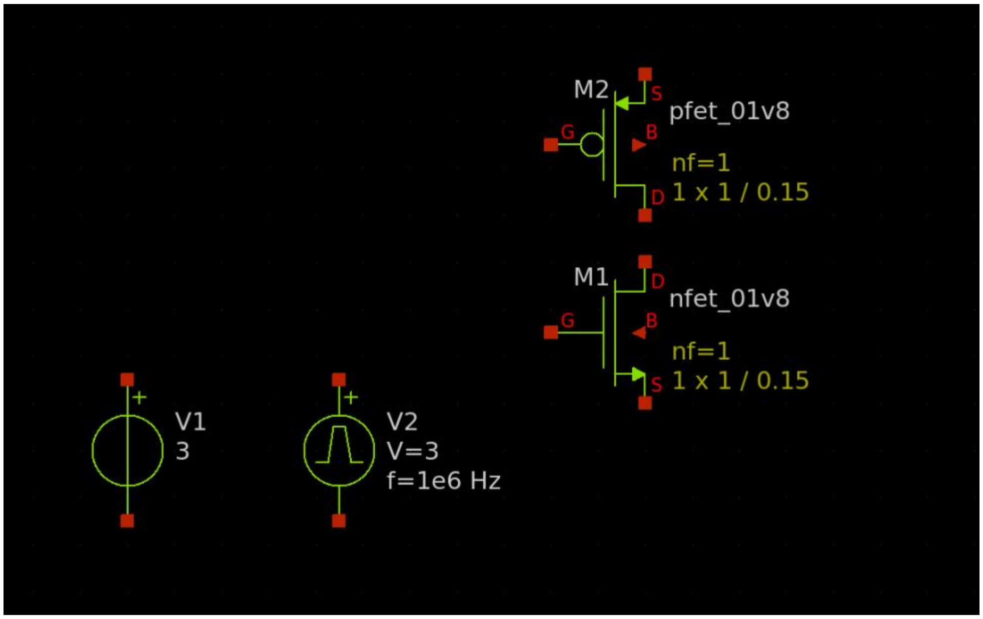

Choose symbol

el7/pkgs/xschem-3.1.0/share/xschem/xschem_library/devices
el7/pkgs/xschem-3.1.0/share/doc/xschem/examples
el7/pkgs/xschem-3.1.0/share/doc/xschem/ngspice
el7/pkgs/xschem-3.1.0/share/doc/xschem/logic
el7/pkgs/xschem-3.1.0/share/doc/xschem/xschem_simulator
el7/pkgs/xschem-3.1.0/share/doc/xschem/binto7seg
el7/pkgs/xschem-3.1.0/share/doc/xschem/pcb
el7/pkgs/xschem-3.1.0/share/doc/xschem/rom8k

vccs.sym
vcr.sym
vcvs.sym
vdd.sym
verilog_delay.sch
verilog_delay.sym
verilog_preprocessor.sym
verilog_timescale.sym
vsource.sym
vsource_arith.sym
vsource_pwl.sym
zener.sym

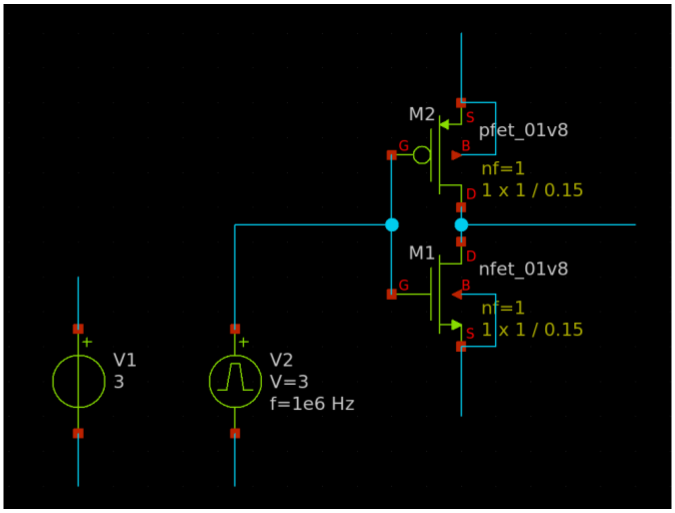
● All ○ .sym ○ .sch File/Search: vsource.sym

Your schematic should now look something like this:

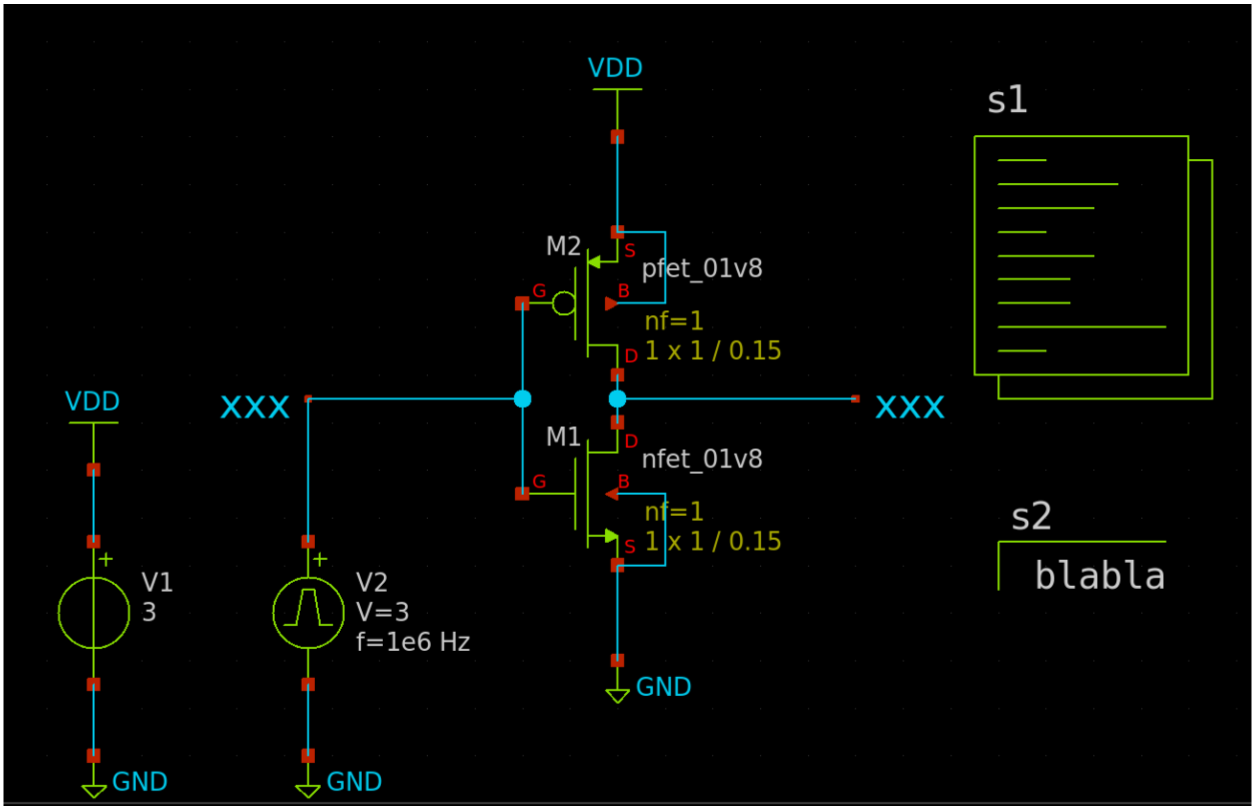


Next, we will connect the devices using wires. Specifically, we will tie the gates of the transistors together as well as their drains. We will also create wire stubs that will eventually be labelled with net labels. We will also connect the gates of both devices to the square-wave voltage source. Lastly, we will tie the bulk terminals of the transistors to their respective source terminals. The bulk terminals can be tied to other locations to change the threshold voltage of the MOSFETs, but for now, we will stick the aforementioned instructions.

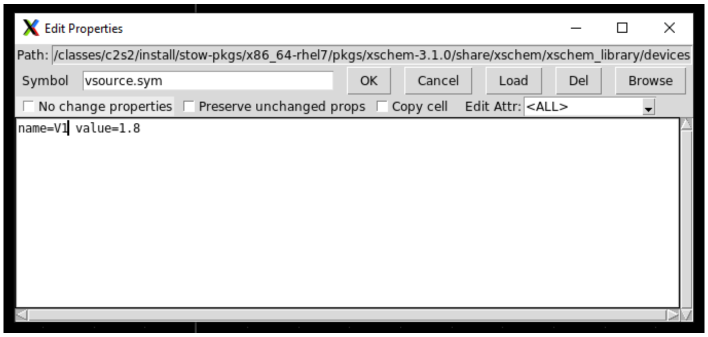
To create a wire, press W where you wish to begin drawing your wire and left click at the location where you wish to end the wire; if you want your wire to change directions, continue to press W at the locations where you want your wire to turn and only left click where you want to end the wire. If you mess up and wish to delete a wire or a component, simply select the object of interest and press the delete key. Your schematic should look like this now:



Now, we will add *ground* connections, *VDD* symbols (which will act as labels for your *VDD* supply voltage), code block symbols, and net labels to the circuit (for the inverter's *Vin* and *Vout*). Search for the respective symbols in the same library as before: *gnd.sym*, *vdd.sym*, *code.sym*, *codeshown.sym*, and *lab_pin.sym*. To copy a component, click the component and press C; finally, click where you wish to place the copied component. To associate a wire with a net label, click on the wire of interest when placing your *lab_pin* symbol. To flip a symbol, press *Alt+F*. To rotate a symbol, press *Alt+R*. Your schematic should now look something like this:



We are only a few steps away from being able to simulate our circuit! Now, we will turn our attention to configuring the voltage sources and code block properly. To edit the properties of a component, double click on the component or click it once, and press Q (just like in Cadence Virtuoso). First, let us set the DC voltage of the VDD supply. Double click on the V1 voltage source and edit the value to be 1.8 (which corresponds to 1.8 volts).



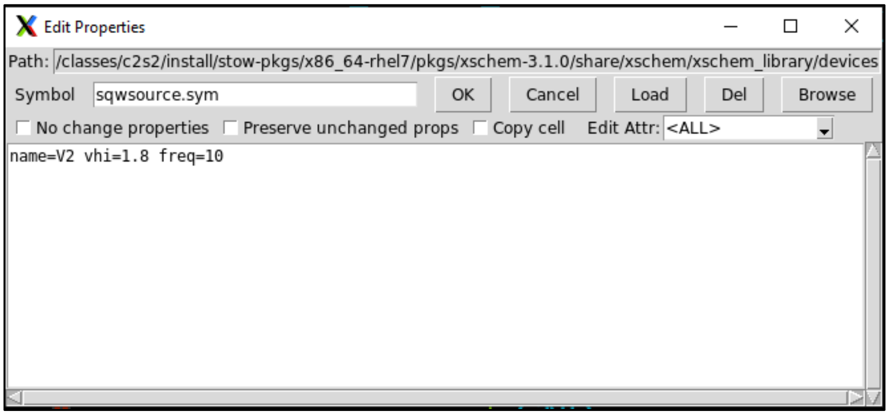
Path: /classes/c2s2/install/stow-pkgs/x86_64-rhel7/pkgs/xschem-3.1.0/share/xschem/xschem_library/devices

Symbol vsource.sym

☐ No change properties ☐ Preserve unchanged props ☐ Copy cell Edit Attr: <ALL>

```
name=V1 value=1.8
```

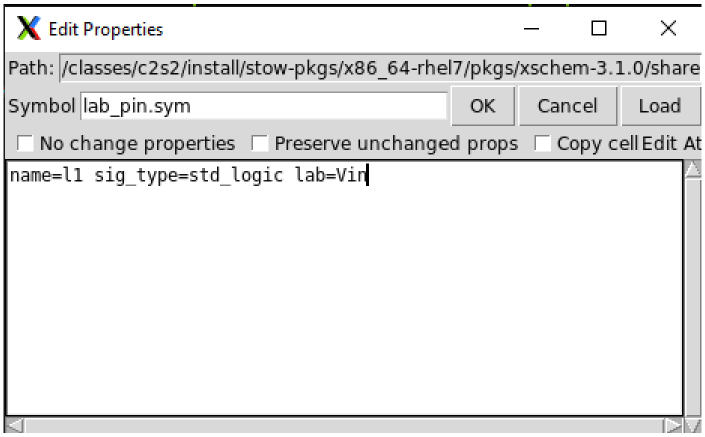
Next, we will configure the square input voltage source that will be fed into the inverter. Set vhi (which stands for voltage high) to 1.8. This means that the square wave will oscillate between 0 and 1.8V. Next, set the frequency to 10 (this corresponds to 10Hz).


Edit Properties
Path: /classes/c2s2/install/stow-pkgs/x86_64-rhel7/pkgs/xschem-3.1.0/share/xschem/xschem_library/devices
Symbol
OK
Cancel
Load
Del
Browse
□ No change properties
□ Preserve unchanged props
□ Copy cell
Edit Attr: <ALL>

Symbol: sqwsource.sym

```
name=V2 vhi=1.8 freq=10
```

For the net labels, label the input to the inverter *Vin* and the output of the inverter *Vout*.

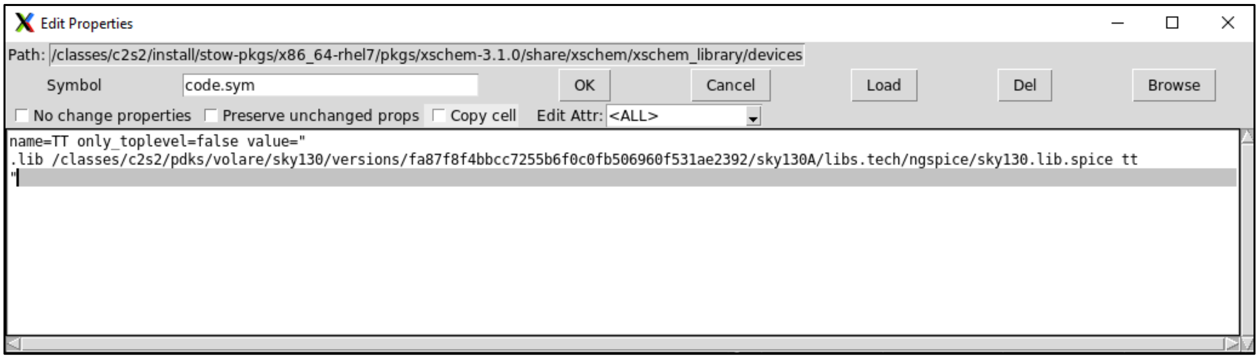

Edit Properties
Path: /classes/c2s2/install/stow-pkgs/x86_64-rhel7/pkgs/xschem-3.1.0/share
Symbol
OK
Cancel
Load
□ No change properties
□ Preserve unchanged props
□ Copy cell
Edit At

Symbol: lab_pin.sym

```
name=l1 sig_type=std_logic lab=Vin
```

Now, double click the *s1* code block and modify its parameters to match what is shown in the following image. Essentially what the code is doing is providing a path to the PDK's Spice library. The *tt* at the end of the path simply means that the devices in the schematic will operate nominally. We can adjust *tt* in order to use other models; this will prove helpful when we will wish to the test the circuit under different corner conditions (for example, process variation and temperature variation). The path has been provided below for your convenience:

/classes/c2s2/pdks/volare/sky130/versions/fa87f8f4bbcc7255b6f0c0fb506960f531ae2392/sky130A/libs.tech/ngspice/sky130.lib.spice



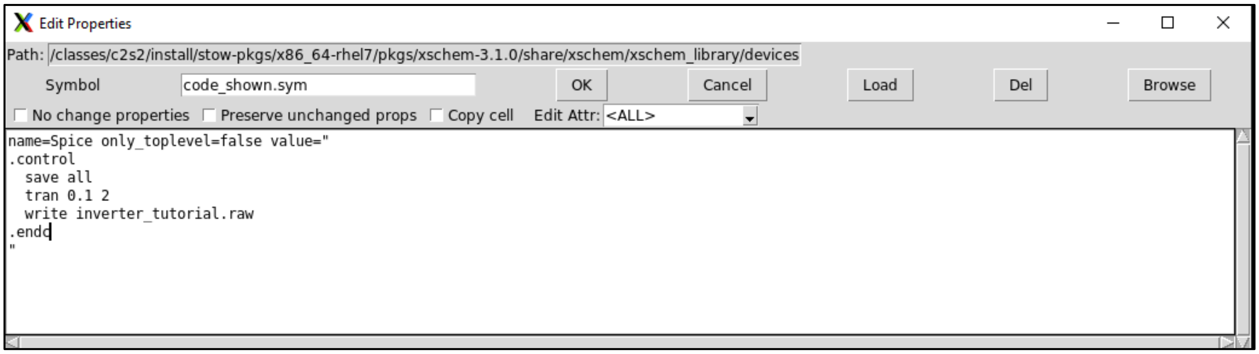
Path: /classes/c2s2/install/stow-pkgs/x86_64-rhel7/pkgs/xschem-3.1.0/share/xschem/xschem_library/devices

Symbol code.sym

□ No change properties □ Preserve unchanged props □ Copy cell Edit Attr: <ALL>

```
name=TT only_toplevel=false value="
.lib /classes/c2s2/pdks/volare/sky130/versions/fa87f8f4bbcc7255b6f0c0fb506960f531ae2392/sky130A/libs.tech/ngspice/sky130.lib.spice tt
"
```

Lastly, we need to enter the Spice directives into the *s2* code block to perform the transient analysis as well as to capture the raw data so that we can display waveforms. Modify the parameters of the code block to match what is shown in the following image. Essentially, the *.control* and *.endc* demarcate the beginning and end of your Spice directives. The *save all* saves the data from the simulation. The *tran 0.1 2* means that a transient simulation will be performed for 2 second in 0.1 second increments. Lastly, the final write command places the data into a raw file that can be used to generate a waveform. Make sure the name of the raw file matches the name of your schematic.



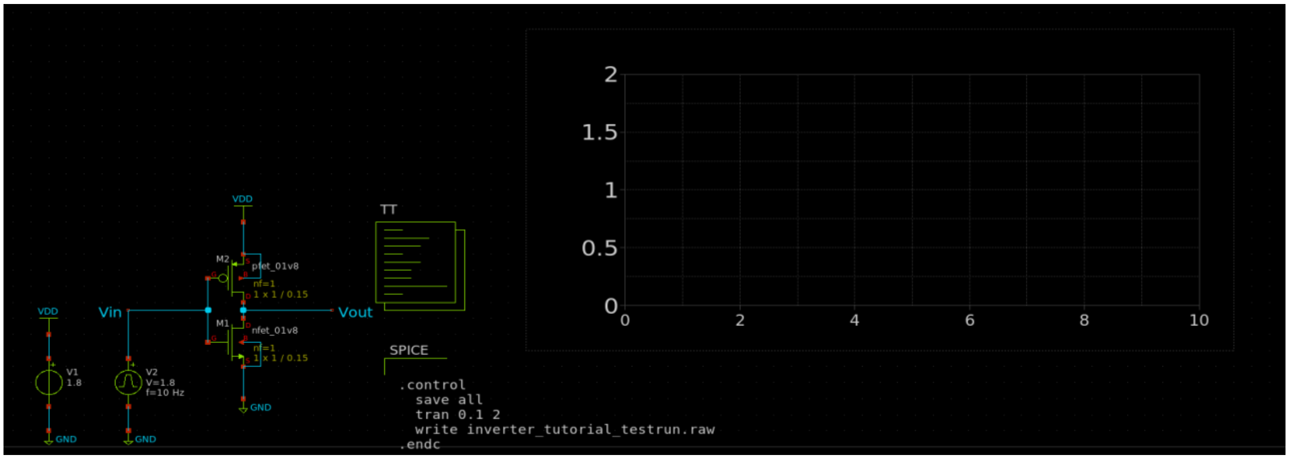
Path: /classes/c2s2/install/stow-pkgs/x86_64-rhel7/pkgs/xschem-3.1.0/share/xschem/xschem_library/devices

Symbol code_shown.sym

□ No change properties □ Preserve unchanged props □ Copy cell Edit Attr: <ALL>

```
name=Spice only_toplevel=false value="
.control
  save all
  tran 0.1 2
  write inverter_tutorial.raw
.endc
"
```

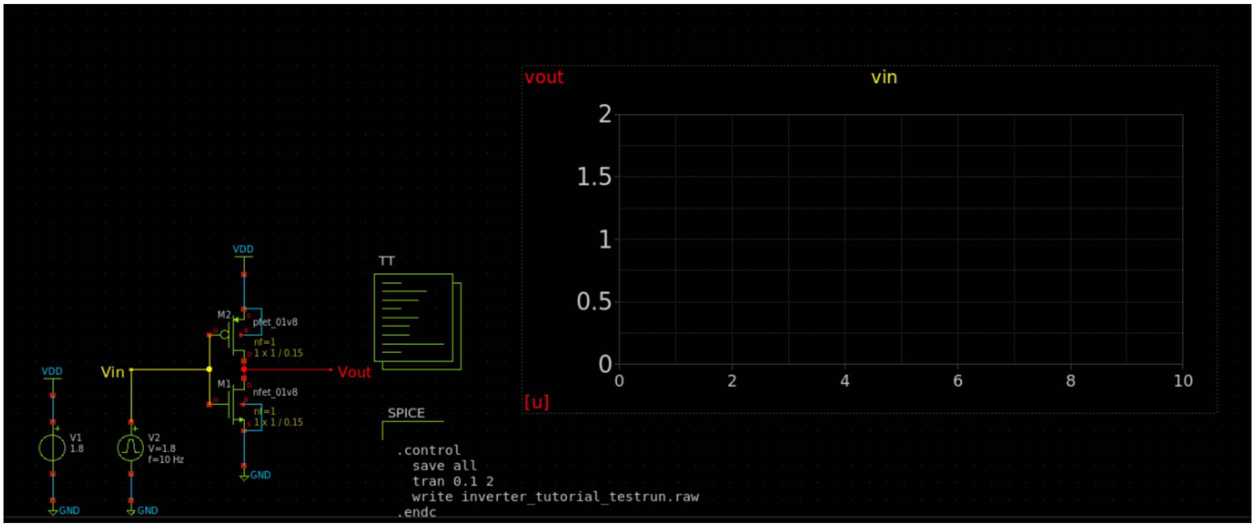
Next, go to the toolbar at the top of your screen and click Add waveform graph. Place the graph in an appropriate location that does not obscure or cover your schematic. Your schematic should look like this now:



```
.control
  save all
  tran 0.1 2
  write inverter_tutorial_testrun.raw
.endc
```

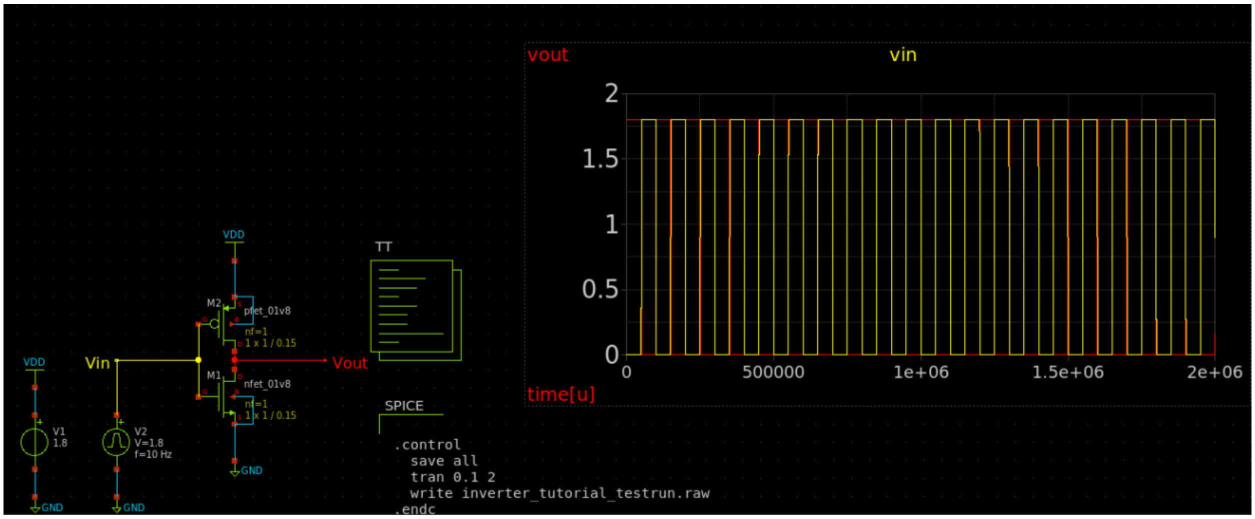
We are now in the home stretch! Click *Netlist* in the top-right corner to generate a netlist that will be used by *Ngspice* to run your simulation. If you want to view the netlist that has been generated, click *Simulation* (in your toolbar) and then *Edit Netlist*. Finally, click *Simulate*!

To view your waveforms, click on your graph to select it. Afterwards, click on the wire of interest and press *Alt+G*. You can repeat this process multiple times on the same graph if you wish to superimpose waveforms or you can simply add more graphs. Your screen should look something like this:



```
.control
  save all
  tran 0.1 2
  write inverter_tutorial_testrun.raw
.endc
```

Next, click *Simulation* and then *Load/Unload ngspice .raw file*. This will load the data from the simulation (which we stored in a raw file) so that we can view it as a waveform. Now click on the graph itself and press F to fit the waveforms on the screen. You should see the input signal and its inverted counterpart!



```
.control
  save all
  tran 0.1 2
  write inverter_tutorial_testrun.raw
.endc
```