

Tutorial:Analog Artist with HSPICE

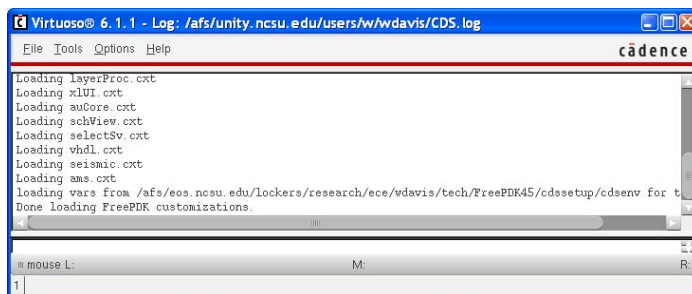
The original document link is “http://www.eda.ncsu.edu/wiki/Tutorial:Analog_Artist_with_HSPICE”

This tutorial will introduce you to the *Cadence* Environment: specifically *Composer*, *Analog Artist* and the *Results Browser*. It will also show you how to use the simulator *HSPICE* in stand-alone mode to make certain parts of your design exploration easier.

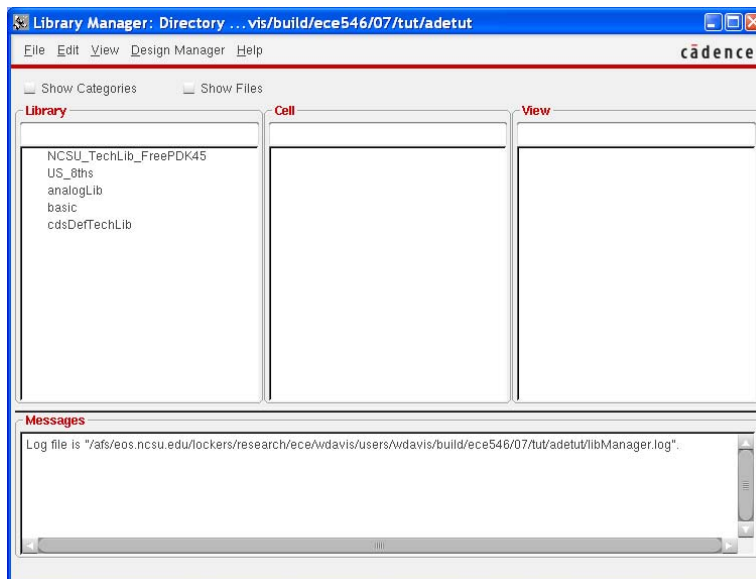
Important Note: The order in which you add cadence2008 and freepdk45 to your environment is important. Unfortunately, the scripts aren't advanced enough to be used in any order. Many students' difficulties have been solved simply by sourcing the setup scripts in the right order.

```
%tsh
% mkdir adetut
% cd adetut
% source cadence_setup.cshrc
% source setup_hspice_new.csh
% icfb &
```

The first window that appears is called the *CIW* (Command Interpreter Window).

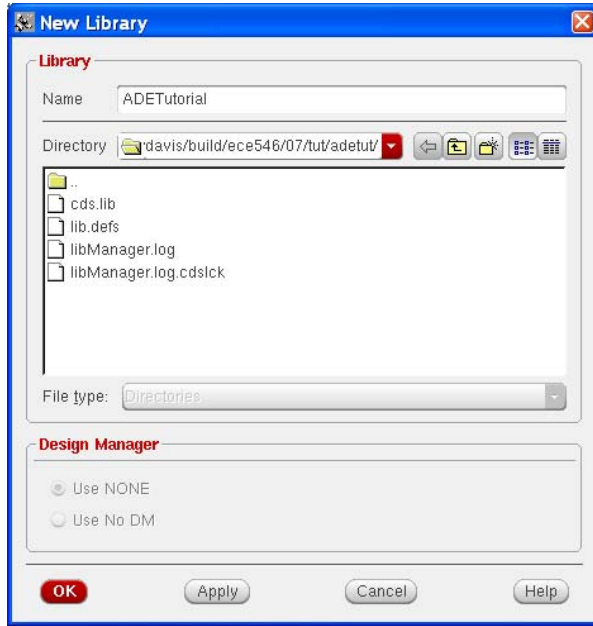


Another window that is very handy is the *Library Manager*, which allows you to browse the available libraries and create your own. To display this window, choose **Tools -> Library Manager...** from the CIW Menu.



Create the myInverter Schematic

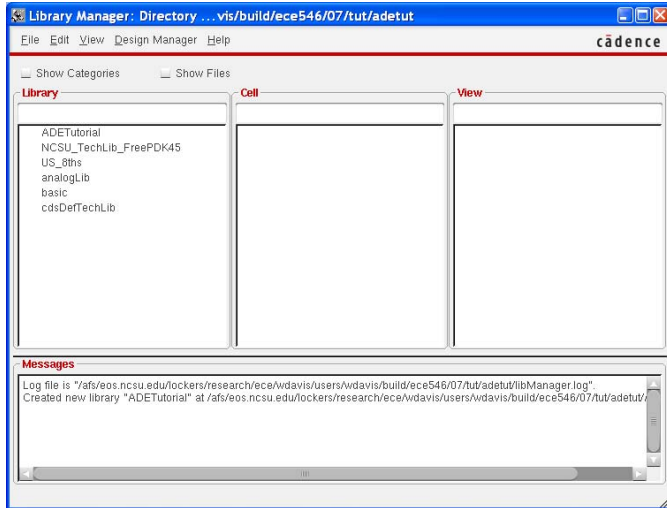
In the Library Manager, create new library called ADETutorial. Select **File->New->Library**. This will open new dialog window, in which you need to enter the name and directory for your library. By default, the library will be created in the current directory. After you fill out the form, it should look something like this:



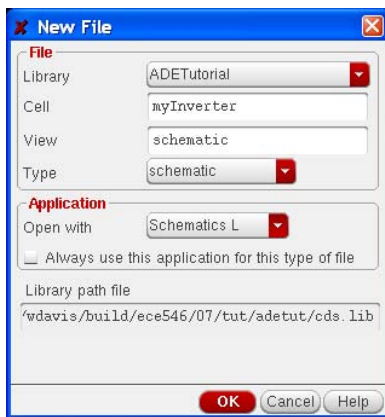
Click OK. Next, you will see a window asking you what technology you would like to attach to this library. Select "Attach to an existing technology library" and click OK. In the next window, select "NCSU_TechLib_FreePDK45".



You should see the library "ADETutorial" appear in the Library Manager.

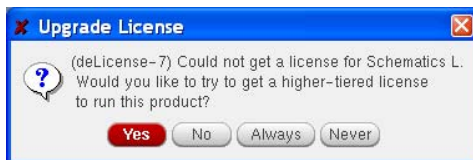


Next, select the library you just created in the Library Manager and select **File->New->Cell View....** We will create a schematic view of an inverter cell. Simply type in "myInverter" under cell-name and "schematic" under view. Click OK or hit "Enter". Note that the "Application" is automatically set to "Schematics L", the schematic editor.

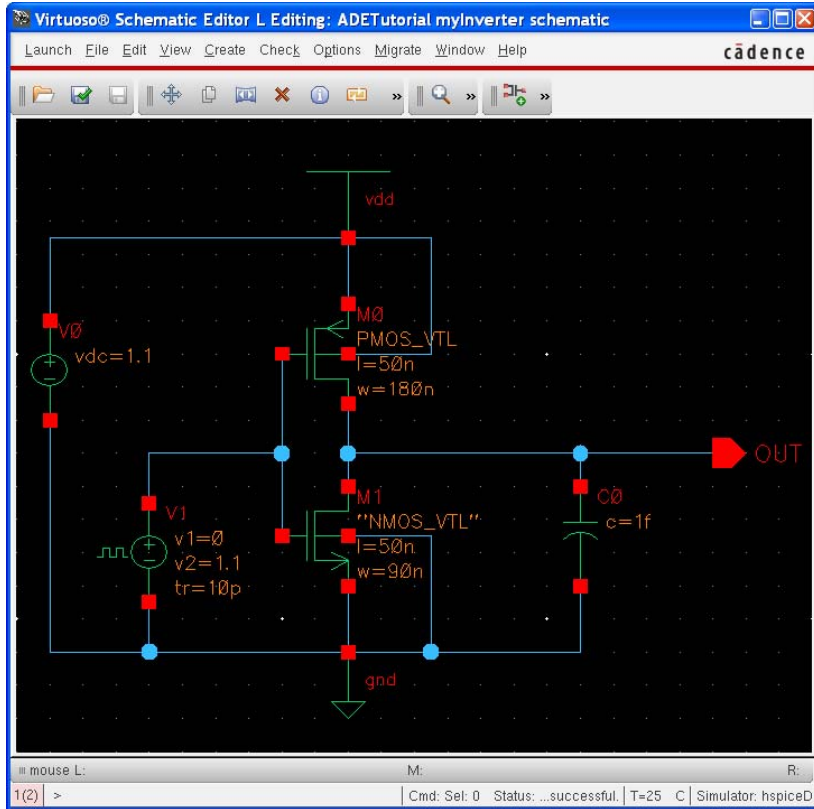


Alternatively, you can select the "Schematics L" tool, instead of typing out the view name. This will automatically set the view name to "schematic".

Click Ok. You may see the following window. Simply click Ok to ignore this warning.

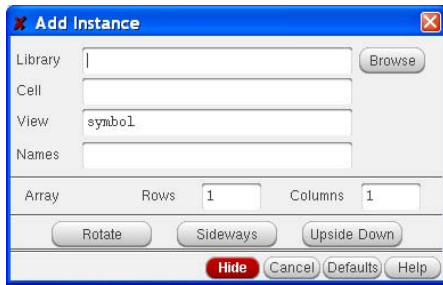


After you hit "OK", the blank Composer screen will appear. The image below shows the final schematic that we will make in this tutorial.



To generate a schematic like this, you will need to go through the following steps:

From the **Schematic Window**, choose **Create->Instance....** The **Add Instance**, dialog will appear.



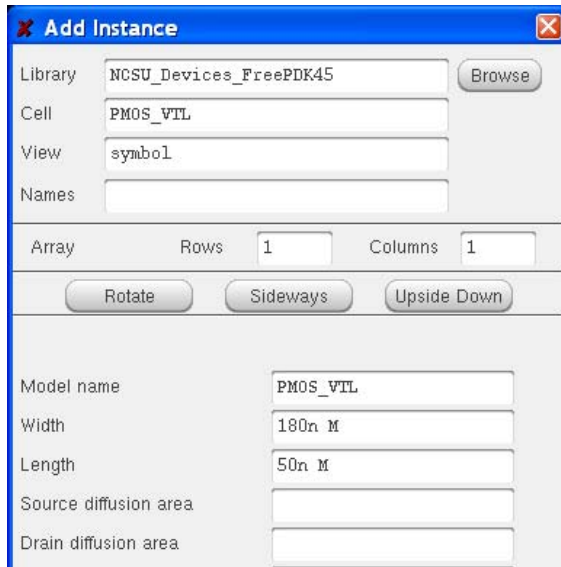
In the **Library** field, select **NCSU_Analog_Parts**. We will place the following instances in the **Schematic Window** from the **NCSU_Analog_Parts** library as instructed below:

Description	Library	Cell	View
NMOS Transistor	NCSU_Devices_FreePDK45	NMOS_VTL	symbol
PMOS Transistor	NCSU_Devices_FreePDK45	PMOS_VTL	symbol
Supply Nets	analogLib	vdd, gnd	symbol
Voltage_Sources	analogLib	vdc, vpulse	symbol
Passive Elements	analogLib	cap	symbol

Note: pay special attention to the parameters specified in *vdc*, *vpulse*, and *cap*. These parameters are very important in simulation.

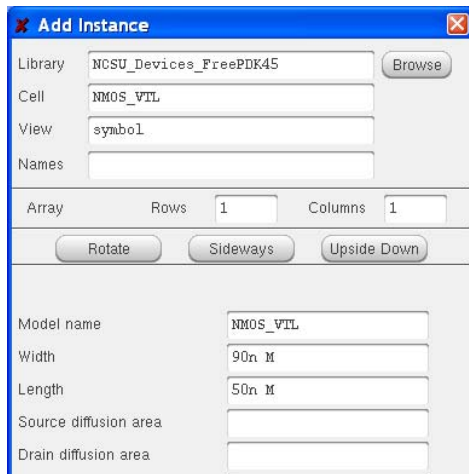
Place pmos instance

In the **Add Instance** window, select the *PMOS_VTL* cell from the library *NCSU_Devices_FreePDK45*. You may either type the values in or click *Browse* and find them in the Library Manager. After you select the cell, the "Add Instance" dialog will change to show the options for this cell. Set the *Width* to "90n" and the *Length* to "50n". Note that "M" will be filled in automatically. Place the PMOS cell in the **Schematic Window**



Place nmos instance

Next, in the **Add Instance** window, select the *NMOS_VTL* cell from the library *NCSU_Devices_FreePDK45*. Set the *Width* to "90n" and the *Length* to "50n". Place the cell in the **Schematic Window**

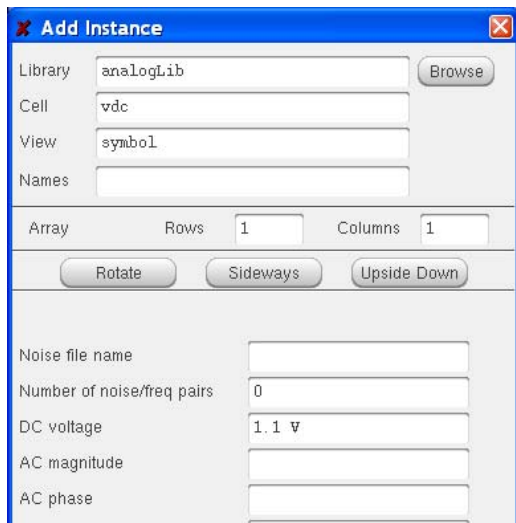


Place gnd and vdd instances

Next, from the *analogLib* library, place instances of *gnd* and *vdd* in the *Schematic Window*.

Place vdc instance

Next, from the *analogLib* library, select *vdc symbol*. In the DC voltage field, enter *1.1*. Note that the "V" will be inserted automatically. Place it in the *Schematic Window*.



Place vpulse instance

Next, from the *analogLib* library, select *vpulse symbol*. Enter the following values in the form:

Voltage 1:	0 V
Voltage 2:	1.1 V
Delay Time:	0 s
Rise Time:	10p s
Fall Time:	10p s
Pulse Width:	40p s
Period:	100p s

Place it in the *Schematic Window*.

Add Instance

Library: analogLib

Cell: vpulse

View: symbol

Names:

Array: Rows: 1 Columns: 1

Frequency name for 1/period:

Noise file name:

Number of noise/freq pairs: 0

DC voltage:

AC magnitude:

AC phase:

XF magnitude:

PAC magnitude:

PAC phase:

Voltage 1: 0 V

Voltage 2: 1.1 V

Period: 100p s

Delay time: 0 s

Rise time: 10p s

Fall time: 10p s

Pulse width: 40p s

Temperature coefficient 1:

Temperature coefficient 2:

Nominal temperature:

Type of rising & falling edge:

Place cap instance

Next, from the *analogLib* library, select *cap* symbol. Enter "1f" in the *Capacitance* field. Note that the "F" will be filled in automatically. Place the instance in the *Schematic Window*.

Add Instance

Library: analogLib

Cell: cap

View: symbol

Names:

Array: Rows: 1 Columns: 1

Model name:

Capacitance: 1f F

Width:

Length:

Multiplier:

Place OUT pin

From the **Schematic Window** menu, select **Create -> Pin...**

In the Pin Names field , enter *OUT*

In the Direction field, select *output*

Place it in the *Schematic Window*



Place wires

In the Schematic Window menu, select **Create -> Wire (narrow)**

Place the wire to connect all the instances

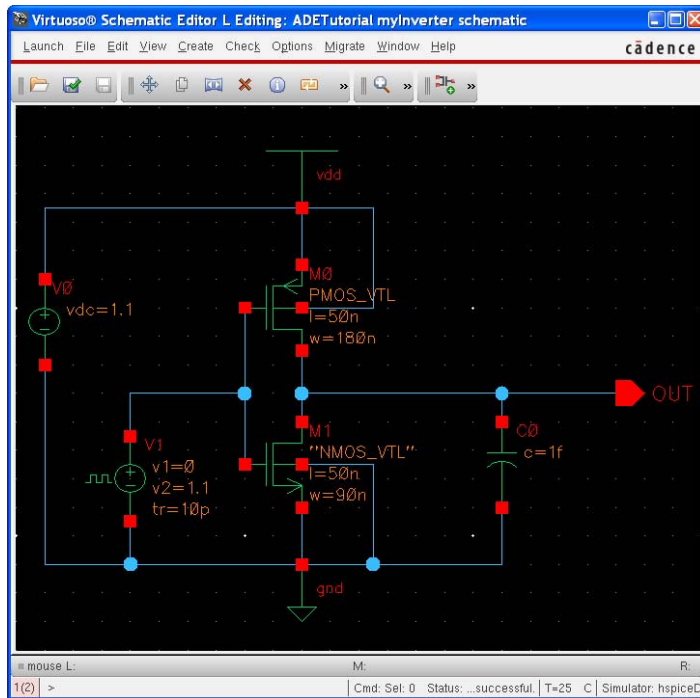
Select **File -> Check and Save**.

Look at the CIW. You should see a message that says:

```
Extracting "myInverter schematic"  
Schematic check completed with no errors.  
"ADETutorial myInverter schematic" saved
```

If you do have some errors or warnings, the CIW will give a short explanation of what those errors are. Errors will also be marked on the schematic with a yellow or white box. Errors must be fixed for your circuit to simulate properly. When you find a warning, it is up to you to decide if you should fix it or not. The most common warnings occur when there is a floating node or when there are wires that cross but are not connected. Just be sure that you know what effect each of these warning will have on your circuit when you simulate.

Your schematic should look like the one shown below.



For more Information about Virtuoso

If you would like to learn more about the schematic editor, you can work through chapters 1-5 of the *Virtuoso Schematic Editor Tutorial* that comes with the Cadence documentation. Start the documentation browser by typing

`cdsdoc &`

at the command prompt, make sure that *IC6.1.1* is selected in the *Active Library* pull-down box at the top, and then select **Virtuoso Schematic Editor->Virtuoso Schematic Editor Tutorial** in the browser window that appears. This should start an HTML browser that displays the table of contents for the tutorial. You may also find the *Virtuoso Schematic Editor L User Guide* very helpful to describe some of the further commands available in the schematic editor.

If you find that you cannot view the figures correctly in the web browser, you can browse to the documentation directory in...

`/ECEnet/Apps1/cadence08/ic610/doc`

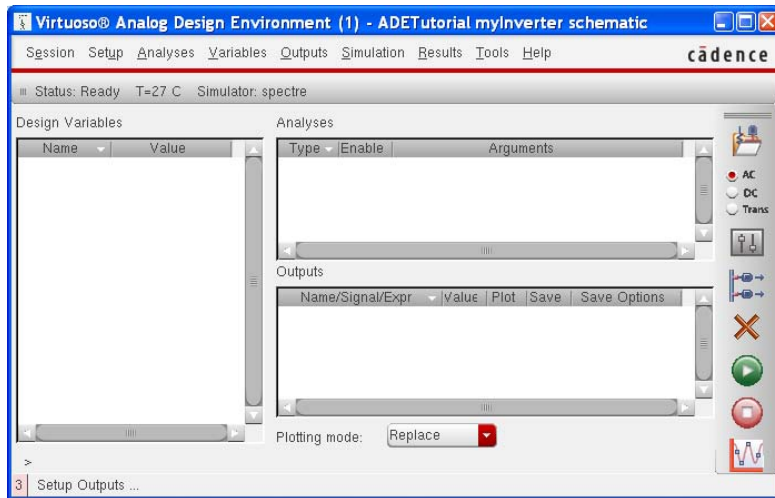
...where you will find PDF files for all of these documents. The *cdsdoc* documentation browser offers many more links for you to learn about the Cadence Design Framework.

Simulate the Schematic with HSPICE within Analog Artist

Set up the Simulation Environment

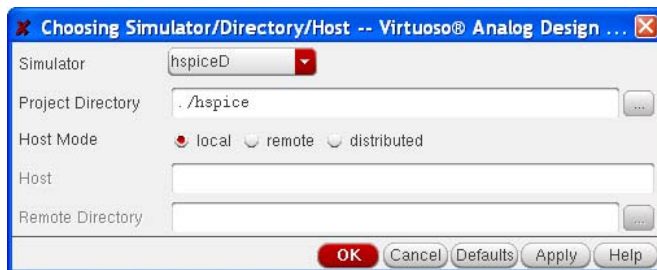
You are now prepared to simulate your circuit.

From the Schematic Window menu, select **Launch -> ADE L**. You may get another message saying that the license need to be upgraded. Simply click Ok to proceed. A window will pop-up. This window is the Analog Design Environment Window.



Choose a Simulator

From the Analog Artist menu, select **Setup -> Simulator/Directory/Host**. Enter the fields as shown below. Choose hspiceD as your simulator. Your simulation will run in the specified Project Directory. You may choose any valid pathname and filename that you like.



Generate the HSPICE Netlist

Next, generate the netlist with the following commands:

From the ADE window menu, select **Simulation -> Netlist -> Create**. This will cause a window to open that displays the text of your netlist.

In the netlist text window, choose **File -> Save As**. Enter the name "inv.sp". The path is relative to the directory where you started Cadence.

Simulate the Schematic with HSPICE in standalone mode

We'll now use the HSPICE simulator to analyze this circuit. This analysis can be done within Cadence, . However, it's sometimes easier to do certain kinds of analysis manually in HSPICE than it is to figure out which Cadence menu option will give you the desired result.

We'll start by adding the analysis and output options to the file **inv.sp** that you created in *Virtuoso*. Open up the file in your favorite text editor (a popular one is nedit). It should look something like the following:

```
** Generated for: hspiceD
** Generated on: Aug 23 12:03:21 2007
** Design library name: ADETutorial
** Design cell name: myInverter
** Design view name: schematic
.GLOBAL vdd!

.TEMP 25
.OPTION
+   ARTIST=2
+   INGOLD=2
+   PARHIER=LOCAL
+   PSF=2

** Library name: ADETutorial
** Cell name: myInverter
** View name: schematic
m0 out net12 vdd! vdd! PMOS_VTL L=50e-9 W=180e-9
m1 out net12 0 0 NMOS_VTL L=50e-9 W=90e-9
v0 vdd! 0 DC=1.1
v1 net12 0 PULSE 0 1.1 0 10e-12 10e-12 40e-12 100e-12
c0 out 0 1e-15
.END
```

This file looks a lot more complicated than it should be. The act of using the **Analog Environment** to create the netlist placed many comments in the file that make it look more cluttered. There are several important things to note about this netlist:

1. Hspice code will always interpret the first line of the file as a comment. Make sure that the first line of your code IS a comment, because if the first line is part of the code for your circuit, it will not be read correctly.
2. Note the ".END" statement at the end of the file. Your file will not finish execution if you do not have this ".END" statement at the end of the code.

Modify the Netlist for a Transient Simulation

We will make a couple of changes to this file to perform a transient simulation.

1. Add the POST option to the ".OPTION" statement. This will ensure that all output variables are saved. This can lead to large files that exceed your disk-quota later in the semester, but it makes things easier for small circuits. (NOTE: Be sure to put the "POST" option at the end of the list. Putting it at the beginning gives an "unknown file error" when opening the output in the waveform viewer, for some unknown reason.) You may add it to the end of the line that reads "PSF=2", or create a new line that begins with "+". Note that the "+" character at the beginning of a line denotes a line continuation.
2. Add a line that reads ".tran 1p 200p". This will create a transient analysis that has a maximum time-step of 1ps and lasts 200ps.
3. Add the following line to access the FreePDK45 transistor models:

```
.include '$PDK_DIR/ncsu_basekit/models/hspice/hspice_nom.include'
```

Running an Hspice simulation

In the same directory as your "inv.sp" file you must add Hspice to your environment. Unfortunately, the HSPICE environment is not compatible with the Cadence environment. In order to run HSPICE, you will have to log in with a new UNIX shell. Then, change to the directory where you want to run HSPICE and type the following commands:

```
% hspice inv.sp >! inv.lis
```

This will cause some output text to be sent back to the inv.lis file. In the event that you have an error with your simulation you will need to read through that output to find out what the error is. If your simulation ran without errors then your output should say:

```
>info: ***** hspice job concluded
```

If your simulation had errors it will look like this:

```
>error ***** hspice job aborted
```

For more information on how to use HSPICE, see the HSPICE Tips and documentation links on the ECE 546 course web-page.

Viewing your waveform

Once your simulation runs without errors, we can view the waveform that was generated. Hspice doesn't have any built-in waveform viewer, so we will have to use the *CosmosScope* software package for that. In the same directory as your "inv.sp" file, start *CosmosScope*.

```
% scope &
```

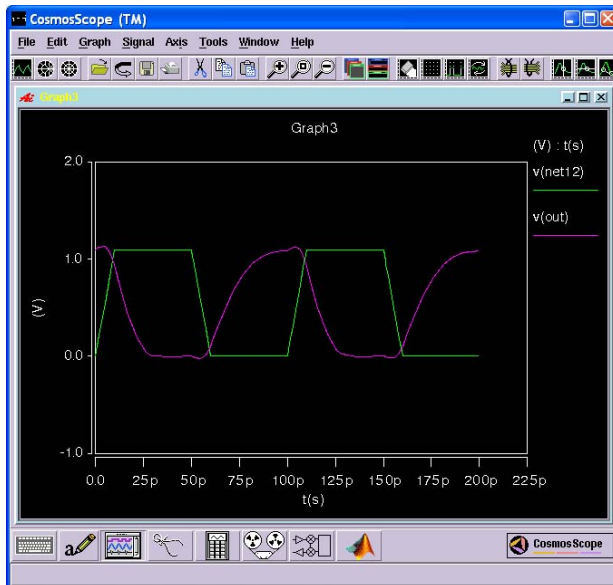
1. Within cscope, select **File -> Open -> Plotfiles...**, and open the "inv.tr0" file. The output of transient analyses will have the extensions .tr0, .tr1, etc., depending on how many .alter statements you have in your file. We have no .alter statements, and so there is only one transient output.

This should cause a window to appear labeled "(1) inv.tr0", in which you can browse the available signals in this file.

2. Double click on the "v(net12)" entry in the signal window. Note that *net12* is connected to the gates of the NMOS and PMOS transistors. It may be named differently in your simulation.

This should cause a graph window to open, plotting the waveform of the input.

3. Left-click on the "v(out)" entry in the signal window to select it. Then right-click in the graph window and choose "Plot". This will plot both signals on the same graph, as shown below.



For more information on how to use CosmosScope, see the documentation links on the ECE 546 course web-page.

Modify the Netlist for a DC Simulation

We will make a couple of changes to this file to perform a DC simulation.

1. Add a line that reads ".DC V1 lin 100 0 1.1". This will create a DC analysis that sweeps voltage source V1 linearly from 0V to 1.1V with 100 points.
2. Only one sweep is allowed, so we need to remove the .TRAN line, if it exists.
3. Edit the "V1" line so that it reads simply "V1 net12 0 0" (where *net12* is simply the name of the net connected to the output of V1... the name may be different for your design). The *vpulse* data on the line is unnecessary for a DC analysis.
4. Remove the "C0" line. The capacitor is meaningless in a DC simulation.

After you make the changes, re-run the simulation.

Measure the Slope of the VTC

The DC simulation will produce an output file with a ".sw0" extension (for "sweep"). Open this file now in cscope and plot the **v(out)** signal. Because we swept V_{in} from 0 to 1.1 V, you should see a plot of V_{out} vs. V_{in} , which is the voltage transfer characteristic (VTC) of the inverter.

From here, you can select **Tools->Measurement Tool** to get a window called *Measurement*. In the *Measurement* window, choose **Measurement->General->Slope** and set the active graph and signal to the VTC that you just plotted. Finally, select *Apply measurement to: Entire waveform* and click *Apply*. You should see a cursor appear on the curve. Click and drag this cursor to pull it along the VTC and plot the slope value as it moves. When it gets close to -1, leave the cursor where it is and click *Apply* again. This will give you another cursor to find the second point where the slope is -1. You may also click and drag the measurement value to make a more attractive plot.

You will also need to find the value of X at this point. To do that, select **Measurement->General->At X** in the *Measurement* window. Click *Apply* to get a cursor, and place the cursor directly over one of the two cursors you just placed. This second measurement give you the X & Y coordinates of the point. Click *Apply* a second time to get the X & Y coordinates of the second point. You can browse and delete measurements from the *Measure Results* window, which can be displayed by selecting **Signal->Measure Results...** in the main CosmosScope window.

To print out the waveform, the easiest thing to do is export the graph as a PostScript file and copy it to Windows for printing (with either Ghostscript or Microsoft Word). But first, you'll need to change the color-map of the window. Otherwise, you'll get a black background in your post-script file, which will use far too much toner when printing it out. To do this, right-click in the graph window and choose **Colormap->Mono**. If you want to change the line-style for different lines in order to differentiate them, right-click on the line in the legend and choose **Style**. Once the graph looks the way you like, choose **File->Export Image...** in the main CosmosScope window and set the file-type.