Cadence Virtuoso Logic Gates Tutorial

A step-by-step description of designing and testing an AND logic gate using Cadence Virtuoso

Document Contents

Introduction
Starting Cadence Virtuoso
Creating a Design Library
Creating a Schematic Cellview
Creating a Symbol
Setting Up Simulation with Analog Design Environment (ADE)
Running Functional Simulations (transient analysis)
Appendix A: Saving Image Files
Appendix B: Saving Simulation States

Introduction

This document describes how to perform gate-level design and simulation of logic circuits using Cadence Virtuoso with the NCSU design kit. For a full custom design (as opposed to a coded/synthesized design using, e.g., Verilog/VHDL), the design process begins by creating a schematic. The schematic is then simulated to verify operation and optimize performance. To implement an actually microelectronics chip, this would be followed by creating a layout of the physical circuit, checking for design rule violations, and several other steps to ensure the design is correct. For ECE331, we will only perform schematic design and simulation of logic gates. In this tutorial you will create a schematic for a basic digital logic gate, and AND gate, and perform some basic simulations on the schematic to verify it is functioning properly.

Starting Cadence Virtuoso

Before beginning this tutorial you should have setup your account to work with Cadence Virtuoso. The steps for doing this may vary with each class/project, so be sure to follow any class-specific setup steps before proceeding with this tutorial. ECE331 students should have completed the *Cadence Virtuoso Setup Guide* before continuing.

Following instructions in the *Cadence Virtuoso Setup Guide*, start Virtuoso

1. With Xming and PuTTY running, move into your Virtuoso working directory (e.g., ECE331/virtuoso) and then enter the following at the PuTTY command prompt to enable Cadence commands.

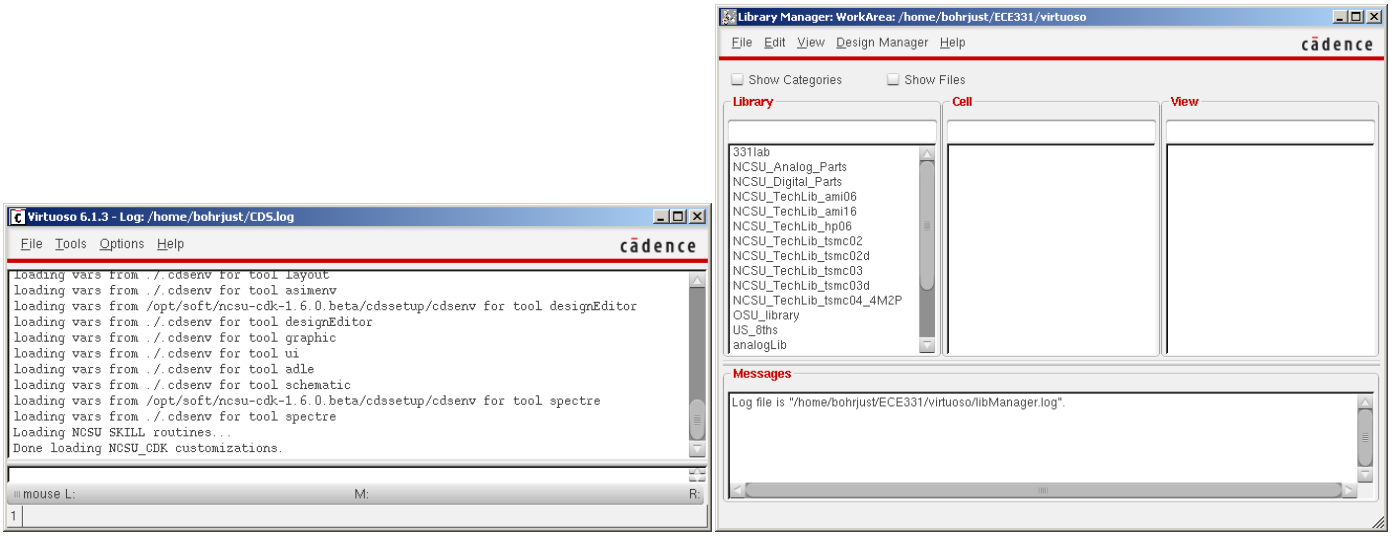
   source $SOFT/cadence

2. To start *Virtuoso* (while in ECE331/virtuoso), enter the command

   virtuoso &

   You should now see two new windows open on your desktop as shown below. The Virtuoso 6.1.3 Command/Log (VCL) window will display a log of completed operations and allow you to enter command-line instruction at the bottom command prompt. When all the configuration files have been read, you should see a "*Done loading NCSU_CDK customizations*" message in the VCL window indicating the start up was successful. With each new session, Cadence starts a new CDS.log file in your working directory where all the messages that appear in the VCL window will be stored.

   The second window that opens is the Library Manager window, which lists the libraries in your working directory. For ECE331 students, the only library that you will use is the ECE331 library and project libraries you create. You can ignore all other libraries; they are for other classes or research purposes.



Main Virtuoso windows after launching Virtuoso

In this tutorial, a simplified convention will be used to show the sequence of steps for the pull down menu. For example, File => Exit will indicate that you open the pull down menu for File and then click on Exit. Another example could be Tools => Analog Artist => Simulation, which will indicate that you pull down the Tools menu, then click on the Analog Artist button and finally click on the Simulation button.

Note: If at any time during this tutorial you want to quit *Virtuoso*, make sure you save your work by selecting File => Save on all open schematics and then close the schematic windows by selecting File => Close All. After you have closed all your schematic windows, select File => Exit in the VCL window to end the Virtuoso session.

Creating a Design Library

Cadence File Organization

To start a design in Cadence, you must first create a project library where you can store your design cells. Every library is associated with a technology file that supplies the design rules, extraction parameters, etc. required to view, design, and simulate your circuit. Cadence stores its files in libraries, cells, and cellviews.

A library (which actually appears as a directory in UNIX) contains cells (subdirectories), which in turn contain views. Each library contains a catalog of all cells, viewed along with the actual UNIX paths to the data files. Each cell in a given library uses the same technology file parameters. A *cell* is the basic design object. It forms an individual building block of a circuit. It is a logic, rather than a physical, design object. Each cell has one or more *views*, which are files that store specific data for each cell. A *cellview* is the virtual data file created to store information in Cadence. A cell may have many cellviews, signifying different ways to represent the same data represented by the cell (for example, a layout, schematic, etc).

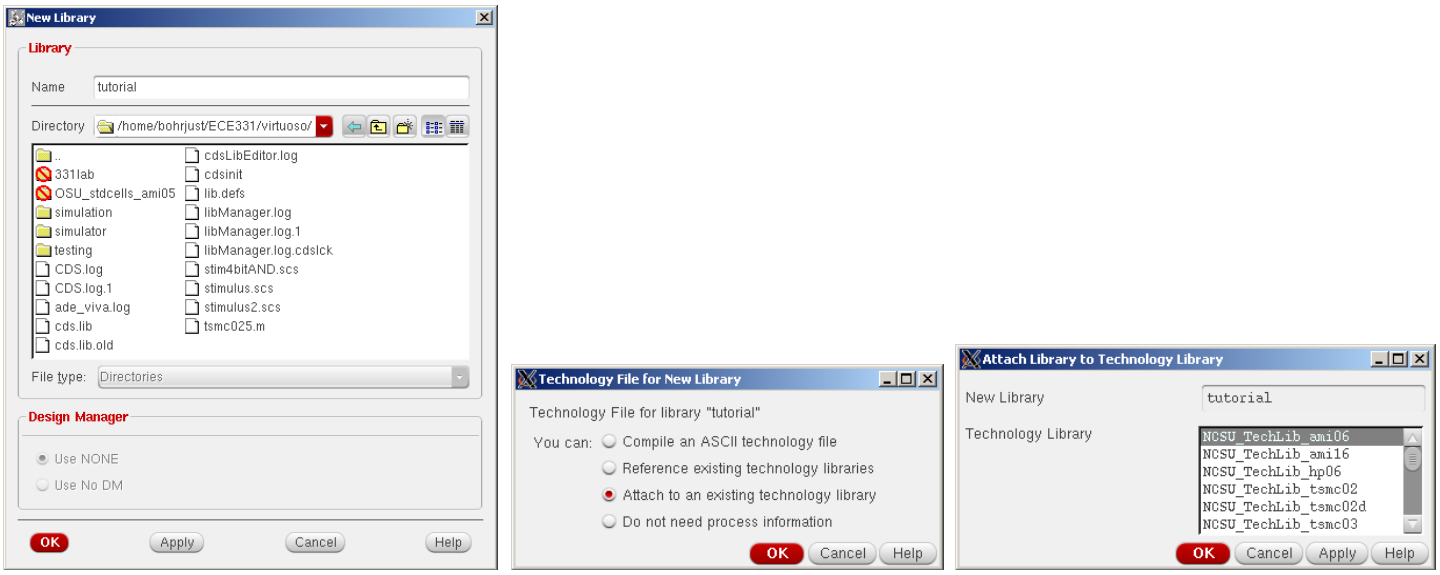
Example Organization:

Library: logic_gates
- Cell: inv
  - View: schematic
  - View: symbol
- Cell: nand2
  - View: schematic
  - View: symbol
  - View: layout
  - View: extracted

Library: ripple_carry_adder
- Cell: 1bit_adder
  - View: schematic
- Cell: 2bit_adder
  - View: schematic

Complete the following steps to create your project library:

1. In the Library Manager window, select File => New => Library to open the New Library window shown below.
2. Enter a Name for your library. The example shows the library name *tutorial*, but since you will be using the cells in your project library (and adding more to it) choose a name like *331lab*, or *digital*. Do not use ECE331 as there is already a parts library with that name. Leave the Path as it defaults. Click OK.
3. In the window that opens, click on the Attach to Existing Tech Library button and click OK. Then choose NCSU_TechLib_ami06 as the technology file to be associated with your new library. Click OK again.
4. Having returned to the New Library window again, click on OK and you will now have an empty project library you can start adding cells to.

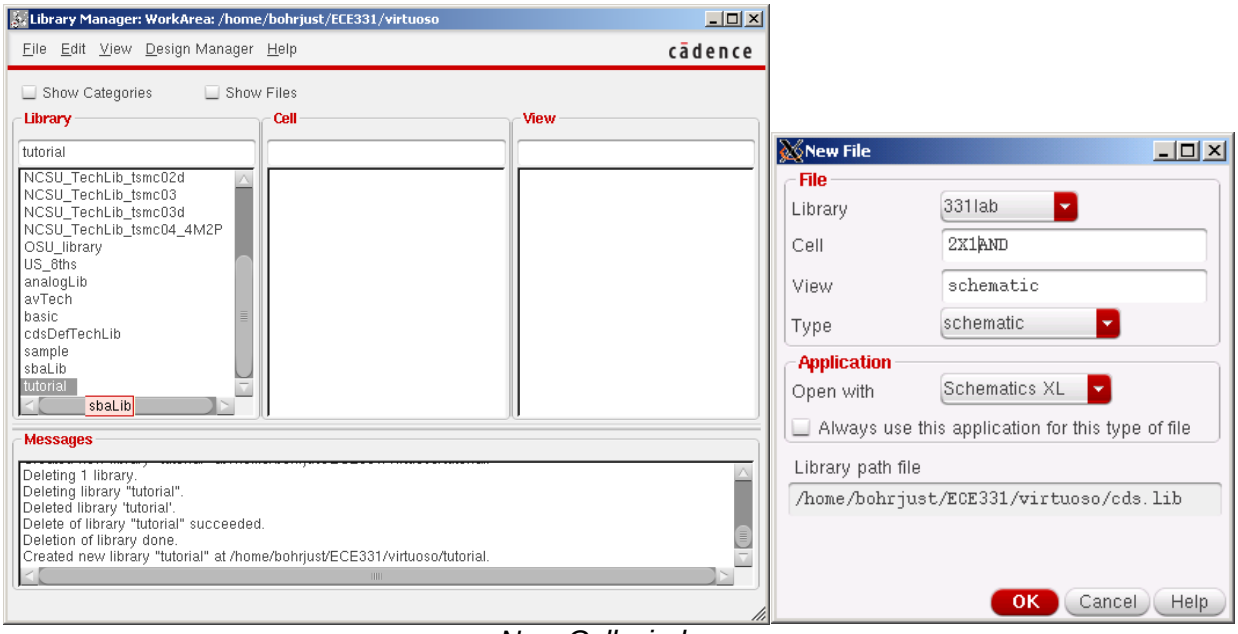


New Library windows

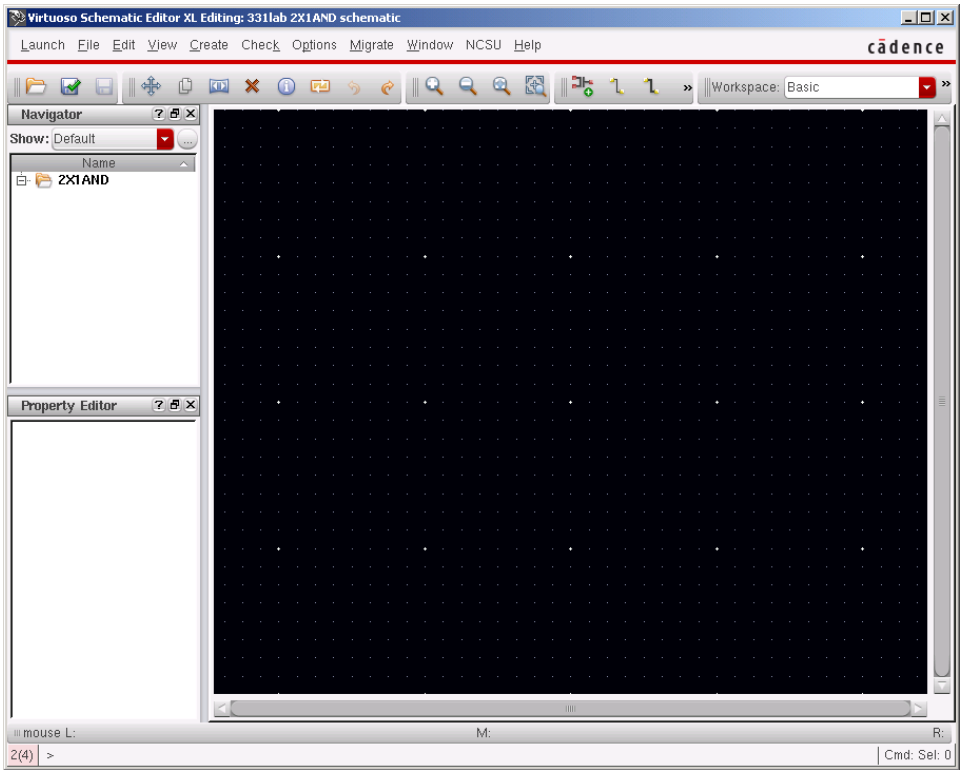
Creating a Schematic Cellview

Complete the following steps to create the schematic of an AND gate within your new project library.

1. Go to the Library Manager window and click/select your library (for example "tutorial").
2. Now select File => New => Cellview. Use the New File window that pops up to create the schematic view for a 2-input AND gate.
3. Make sure your project library is in the Library field. If not, *Cancel* and repeat step 1. Next, enter the Cell Name "2x1AND" for your two-input (and one output) AND gate.
4. Select schematic from Type field and Schematic XL from Open with field. This is where you choose which Cadence tool you want to use and the appropriate *View* name for each tool will be filled in automatically. Here we will be creating the schematic view.
5. Click the OK button. The Virtuoso Schematic Editing tool will open with an empty schematic window as shown below.



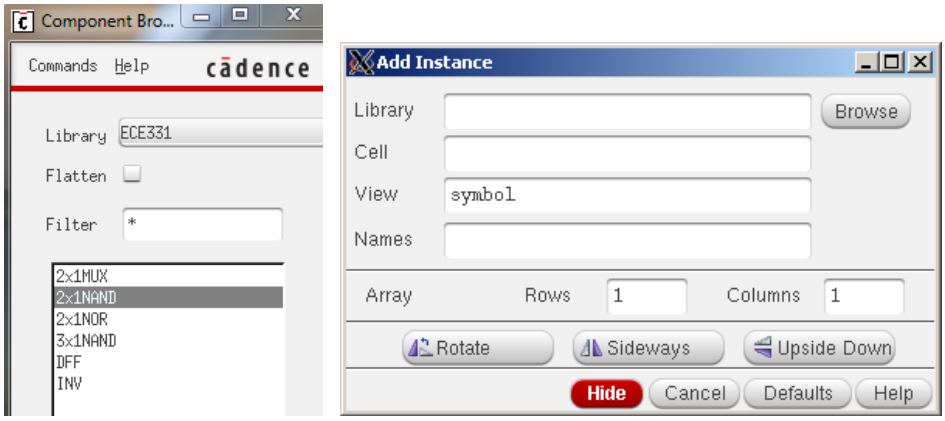
New Cell windows



Virtuoso Schematic Editing window

Add Components: With the 2x1AND cell schematic generated, you can now begin to design the AND gate using components in the ECE331 library.

6. In the Schematic Editing window, select Create => Instance to activate the Add Instance tool for adding components (gates, resistors, etc.) to your schematic. You can also invoke this tool by clicking on the *Instance* icon on the top toolbar (see image below), or by typing the hot key 'i' with your mouse over the Schematic Editing window. Two windows (Component Browser window and Add Instance) will pop open.



Add Instance windows

7. If Component Browser window does not appear automatically, select Browse in the Add Instance window. In the Component Browser window, under Library select ECE331 (it may be the default). A list of parts will be displayed near the bottom of this window. From the parts list, click on 2x1NAND. This will attach the component to your mouse pointer. You can also type the cell name into the Add Instance window (if you know the exact name). If desired, click on the Rotate, Sideways, or Upside-down buttons in the Add Instance window to manipulate orientation of the component you are adding.

8. With the NAND gate selected, click on schematic area (main black area) of the Schematic Editing window and the 2x1NAND component will appear in your schematic. Clicking on the schematic window again will add another copy of the component, but don't do this. Pressing **ESC** on the keyboard will end the Add Instance function, but don't do this yet either!

9. To construct an AND gate out of the gates in the ECE331 library, you will need one NAND gate and one INV gate. Return to the Component Browser window, select the INV cell, and click on the Schematic Editing window to add the INV to your schematic. Place the INV to the right of, but not attached to, the NAND gate since this is where it should go to form an AND function.

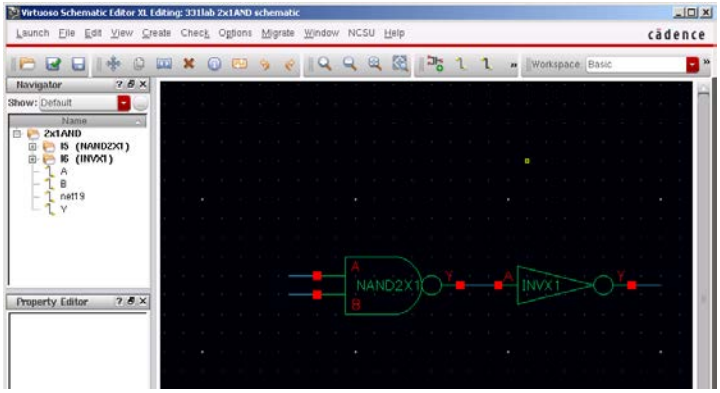
10. Press **ESC** to end adding instances.

Add Wire Connections: With the gates placed, you can now connect the components to complete the AND.

11. In the Schematic Editing window select Create => Wire (narrow) (or use hot key 'w') to begin placing wires that will connect the terminals of the components and pins in the schematic. Add a wire between the NAND output and the INV input as shown below. Also, add wires on the NAND inputs and INV output so that we can place pins on the other end of the wires.

    When adding wires, click once to start a wire or place a node without ending the wire. To end a wire, double-click or single-clicking on a component terminal (e.g., gate input/output, pin) . Once a wire is ended, clicking on the schematic background screen again will start a new wire.

12. Press **ESC** to end wiring. You schematic should now look similar to the schematic below.



AND schematic after adding instances and wires.

Add Pins: All that remains are input and output pins!

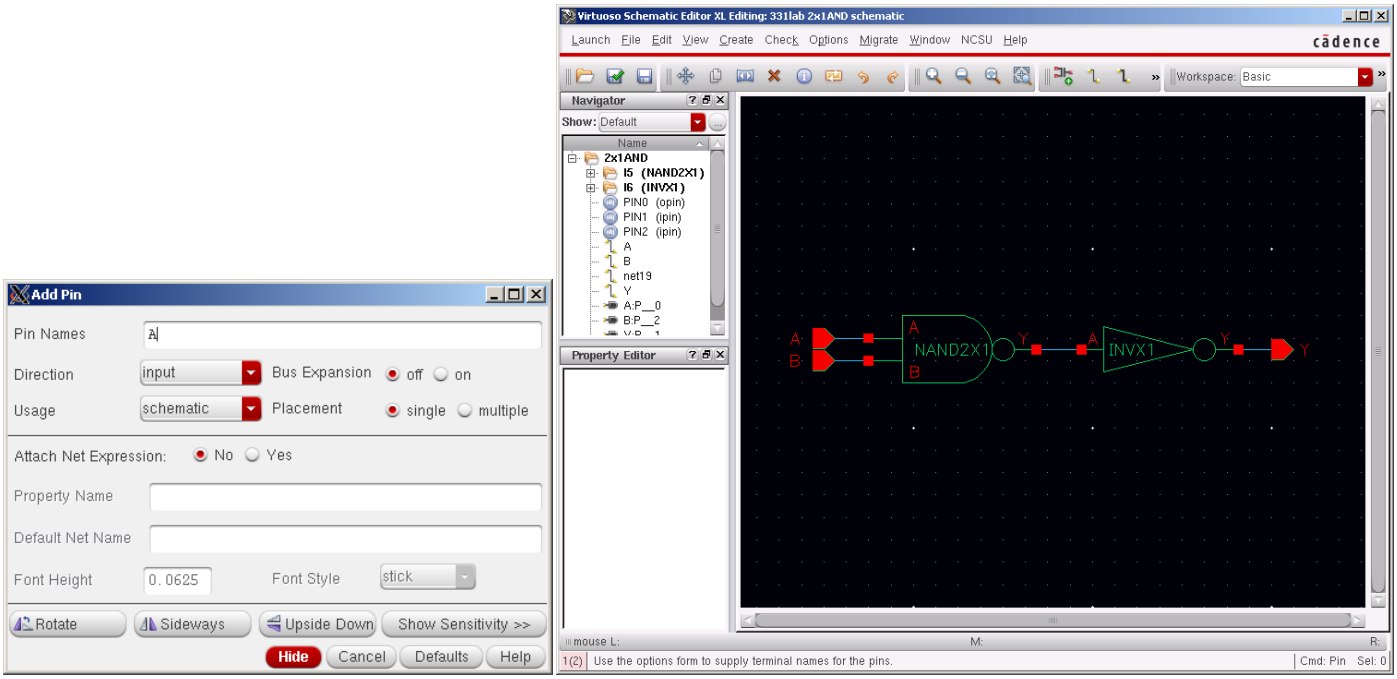
13. In the Schematic Editing window, select Create => Pin (or use hot key 'p') to add an input *pin*.

14. In the Add Pin window that opens, enter 'A' for the Pin Name and select input for the Direction.

15. Click on the Schematic Editing window and drop the pin on the wire to the left of the NAND gate. Click right on the wire and the pin will be attached to the wire without need to edit the wire or add a new wire.

16. Follow the same procedure to an input pin 'B' to the other NAND input, and add an output pin 'Y' to the output of the INV gate. Be sure to select output in the Direction field for the output pin. Note: you can click on the Rotate, Sideways, or Upside-down buttons in the Add Instance window to manipulate orientation of the component you are adding (not needed here/now).

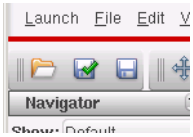
17. Press '**ESC**' when you a done adding pins.



Add Pin window and completed AND schematic.

Check and Save the cell view: After completing any schematic, it should be checked for errors and saved.

18. To check for errors and save the schematic cell view, click on the Check and Save icon (blue floppy disk with green check mark). You will be notified if warning or errors were detected. If so, return to your schematic and look for errors. The most common errors are due to improper wiring or forgetting inputs/output pins (unattached terminals).



When everything checks and saves correctly, you will see a message in the VCL window stating "*Schematic check completed with no errors*" and "*<cell name> saved*".

Creating a Symbol

Complete the following steps to create a *symbol* for your AND gate. A symbol is a simplified representation of a circuit schematic showing only the inputs and outputs. Symbols can be added to other schematics using the Add Instance tool, just like you added the NAND and INV into your AND schematic. Symbols allow complex circuits to be represented by simple boxes (or other shapes, like the INV triangle) and I/O pins.

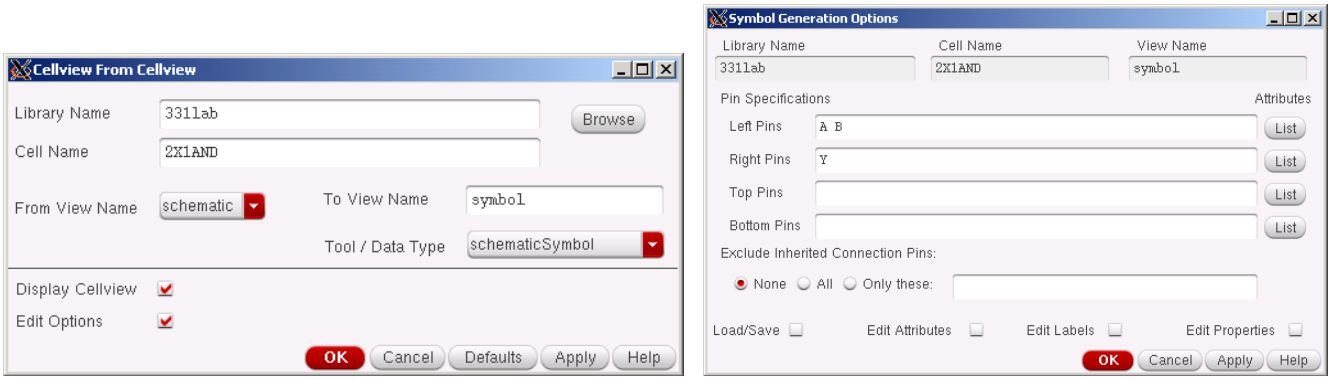
1. In the 2x1AND Schematic Editing window select Create => Cellview=>From Cellview.

2. Two symbol generation configuration windows will pop up sequentially. Leave all the settings to default values and select OK in each window to launch the *Symbol Editor* tool.

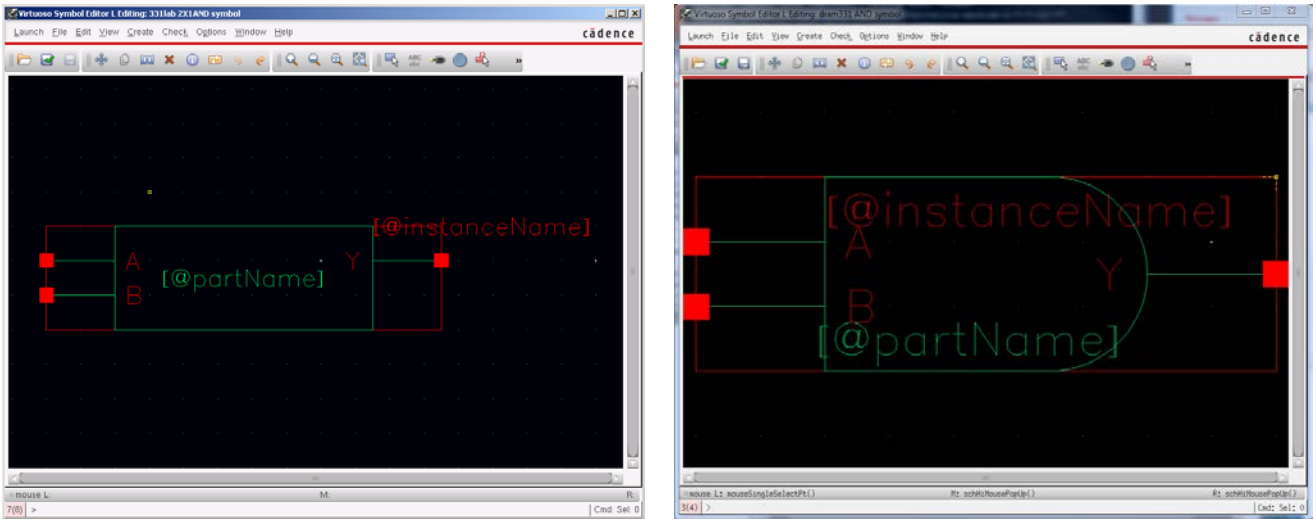
3. In the Virtuoso Symbol Editing window, notice the default symbol shape is a simple rectangle. While this is functionally fine, it would be better if our AND gate looked like a typical AND symbol. Select the

default green rectangle and delete it. Then, use the drawing tools in the top right menu to draw a symbol closer to an AND gate. Try the Circle and Polygon tools. When you realize these won't work well, try the Arc and Line functions.

4. Next, use the cursor to move the output (Y) pin to a proper location for you symbol shape. Notice that the pin (red box linked to text 'Y') is separate from the wire connecting the pin to the symbol body. You can move the part and instance identification parameters anywhere that is convenient.



Symbol generation windows



Default (left) and edited (right) AND gate symbol

5. When you are happy with your symbol, click the Check and Save icon to, you guessed it, check and save your symbol. Errors or warning will be noted in the VCL window.

6. When your symbol is error free, exit the symbol editor by selecting File => Close All.

You have now completed the schematic and symbol design portions of this tutorial. Below are some general editing tips useful for working with Virtuoso and then the tutorial proceeds with simulations, the last (but lengthy) topic.

General Editing Tips

Mouse Buttons: In most Cadence tools, the left mouse button is used to select components, wires, etc., and the middle mouse button can be used to change object properties, e.g., the width and length of a transistor.

Moving Objects: If you want to move any object, just move the cursor on top of the object and type 'm'. The object will then move with the cursor. Or you can select the objects to be moved by left-click and drag to draw a box around the objects. After highlighting the objects to be moved, type 'm' and the highlighted objects will move with your cursor.

Deleting Objects: If you want to delete an object, move the cursor on top of the object and hit the 'DEL' key. You can also highlight an object or a group of objects by drawing a box around it, as described above, and pressing the 'DEL' key.

Undo Operations: When you make a mistake (accidentally delete a component, etc.), you can undo the action by click on the Undo icon in the toolbar.

BindingKey: The following "hot keys" are available for the schematic editing tool.

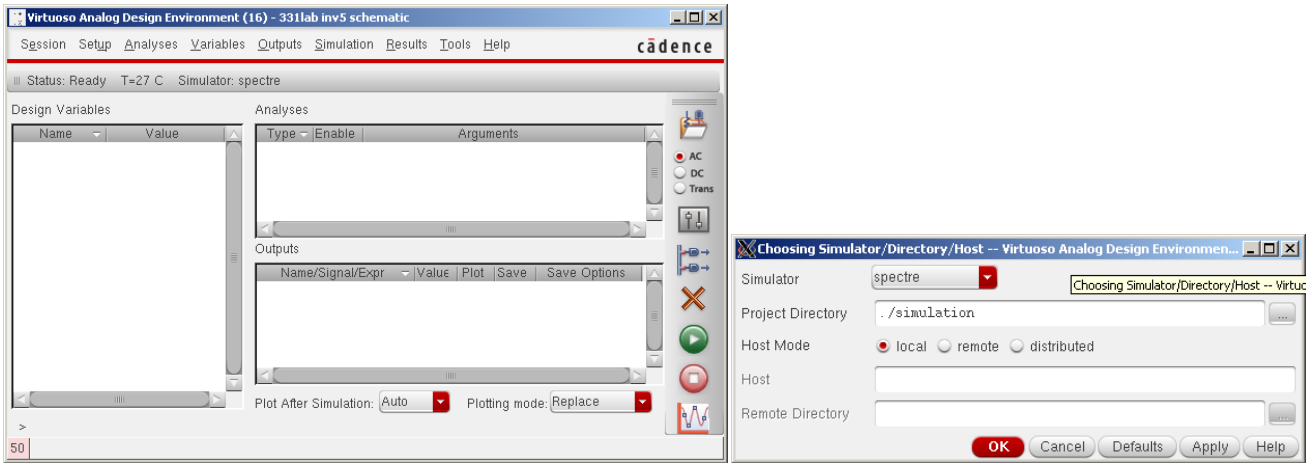
1). Press 'p' to add pins
2). Press 'q' on the device/instance to edit properties for the device
3). Press 'w' to add wires
4). Press 'f' fit the schematic in your schematic window
5). Press 'z' to zoom in the window
6). Press 'shift+z' to zoom out the window
7). Press 'l' to label a wire
8). Press 'Up' and 'Down' arrows to move up and down within a schematic window
9). Press 'ESC' to terminate an operation in the schematic window
10). Press 'u' to undo an operation in the schematic window

Setting Up Simulation with Analog Design Environment (ADE)

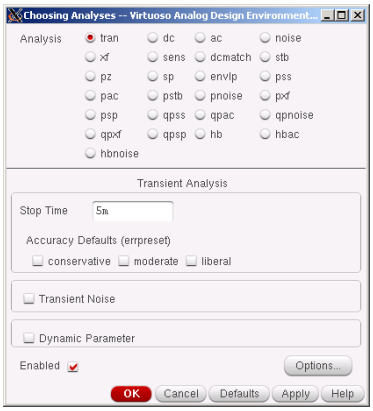
To test the functionality of a schematic we must simulate the circuit. For this task, Cadence provides analog simulation tools within the Virtuoso Analog Design Environment (ADE). For now we will just run a simple transient analysis to confirm the circuit designed above is operating as an AND gate should. However, the Virtuoso ADE is capable of running any simulation SPICE can.

Setup Analog Simulation:

1. With your 2x1AND schematic opened, in the Schematic Editing window select Launch => ADE L to open the Virtuoso Analog Design Environment (ADE) window. You can also launch this tool from the VCL by selecting Tools => ADE L => Simulation in the VCL menu.
2. In the ADE window that opens, choose the cell to be simulated by selecting Setup => Design and specifying the Library Name and Cell Name for your AND gate. Click OK when done.
3. Back in the ADE window, click Setup => Simulator/Directory/Host. Ensure that spectre is selected as the Simulator (should be the default). Leave all other settings to default and click OK to exit this setup.



Virtuoso ADE window (left) and Setup Simulator window (right).



4. In the ADE window, go to Analysis => Choose. In the window that opens, make sure the Analysis is set to tran. Enter 10m in the Stop Time input box and check Enabled at the bottom. The click OK to save the settings.

Setup Stimulus: The schematic defines the components within the cell but does not define the control signals (typically voltage sources) necessary to test the operation of the circuit, such as the power supply voltage and an input voltage signal. These signals are referred to as the *stimulus*, and here we will use a *spectre* stimulus text file to define these signals.

You will need to use a text editor in order to create a stimulus text file. You can use any editor you are familiar with. You can name the stimulus file any name, and put it wherever you wish. In this tutorial, we will call the file "stimulus.txt" and place it in your *ECE331/virtuoso* directory.

5. Start the text editor of your choice. If you are new to UNIX, you may want to use "gedit" by doing the following:
   - Open a terminal window (or use any one that is already open). This is a PuTTY window with a command prompt. If you used the virtuoso & command to launch Virtuoso, you may have to hit ENTER to get a command prompt to return.
   - Make sure you are in your *virtuoso* directory. If unsure, type cd ~/ece331/virtuoso.
   - Type gedit & to create a new file.

6. Once you have opened a text editor, type in the following lines to define the simulator language (*spectre*), the DC supply voltages, and a two square wave pulse input voltage sources that will allow you to perform a transient analysis simulation to verify functionality of the AND circuit. Note that the input waveform used may vary with the type of analysis needed, although the supply voltage source will generally remain the same.

```
simulator lang=spectre
global gnd!
vdd (vdd! 0) vsource dc=3
Gnd (gnd! 0) vsource dc=0
V1 (A 0)  vsource  type=pulse  val0=0  val1=3  delay=0  rise=0.05n  fall=0.05n
width=1m  period=2m
V2 (B 0)  vsource  type=pulse  val0=0  val1=3  delay=0  rise=0.05n  fall=0.05n
width=1m  period=3m
```

   Note: Each bulleted item represents one line in your stimulus file. The 5th and 6th lines (V1 and V2 pulse voltages) are broken into two lines due to the document margins, but they must be on a single line in your stimulus file.

   Make sure that you always end the last line by pressing Enter. This adds a new line character to the line, and informs the simulator that the voltage definition is complete. If you do not do this, you will likely get a netlist read error when you try to Netlist and Run the simulation.

7. Save the text file. If you are using *gedit*, this can be done by selecting File => Save As , entering a 'Name' like *stimulus.txt*, choosing the path to /ECE331/virtuoso in the 'Save in folder' area, and clicking Save at the bottom.

8. You can exit the text editor. For *gedit*, use File => Quit. However, you may want to keep the editor open to work on other simulation stimulus files later.

If you have any trouble creating your own stimulus file, you can download a file with the same values as above from the class website. Choose Lab1A.txt and save it to your virtuoso directory. Note, the tutorial will assume the file is saved as "stimulus.txt".

Regarding the Stimulus File

The voltage sources above are defined using *spectre* syntax. The DC supply voltage named "vdd" is between nodes vdd! and 0 (ground) with value of 3V. The input square wave is defined using the *spectre pulse* syntax

```
<vname> (node+ node-) vsource type=pulse val0=not_pulse_voltage val1=pulse_voltage
delay=delay_length rise=rise_time fall=fall_time width=pulse_width period=period_length
```

where the default units are volts and seconds. The 'n' on the time values sets them to nanoseconds (i.e. 'n' is a 10^{-9} multiplier). Although you can vary the timing values as necessary to meet simulation goals, the rise and fall times shown in Step 6 are good for the chosen CMOS technology and should not be modified unless you are sure you know what you are doing. The illustration below helps to define the pulse waveform parameters.

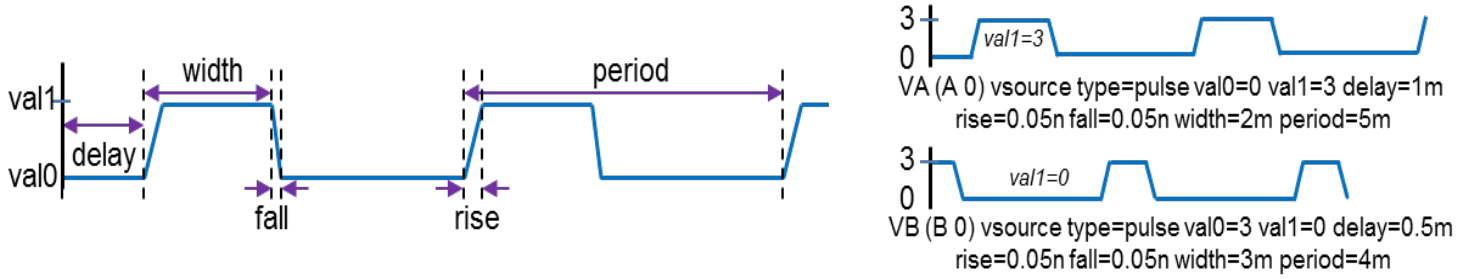
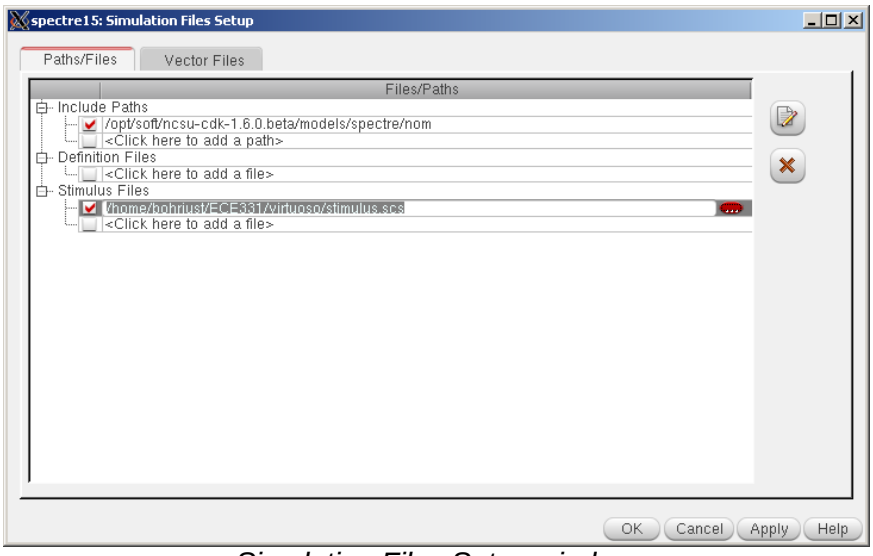


Illustration of pulse waveform parameters (left) and two example pulse definitions (right)

Linking Stimulus File:

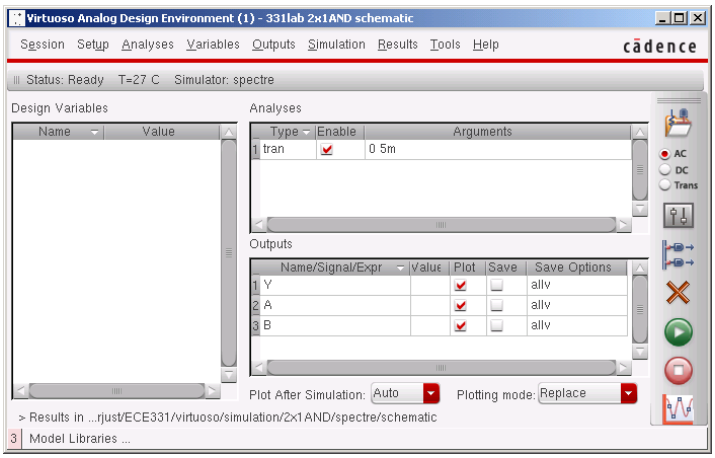
9. Once you have saved the stimulus file, return to the ADE window and link your stimulus file to the simulation program. Click on Setup => Simulation Files.

10. In the setup window that opens, you need to locate your stimulus file. Below the line “Stimulus Files”, click on the “<Click here to add a file>” line and a Browse button should appear on the right hand side. Click the Browse button, select your stimulus file from the window that opens, and click Open to return to the setup window. You should now see the absolute pathname for your stimulus file listed under “Stimulus Files”. If so, click OK to close the setup window without changing anything else.



Simulation Files Setup window

Setup Output Traces: The only step that remains before running the simulation is to select what signals you would like to view after the simulation is complete.



11. In the ADE window, select Outputs => To Be Plotted => Select On Schematic. This will activate the Schematic Editor window and allow you to pick the signals (nets/wires) you would like to have plotted during the simulation simply by clicking on them.

12. In the Schematic Editor window, click on the wire that is connected to the output (Y) of the AND gate. Then, select (click on) the wires connected to each of the inputs (A and B). As you click, you should see the signals being added to the ADE window.

This completes your simulation setup. Your ADE window should now look like the image on the right.

Running Functional Simulations (transient analysis)

Now you will verify that the circuit you designed is indeed operating like and AND gate.

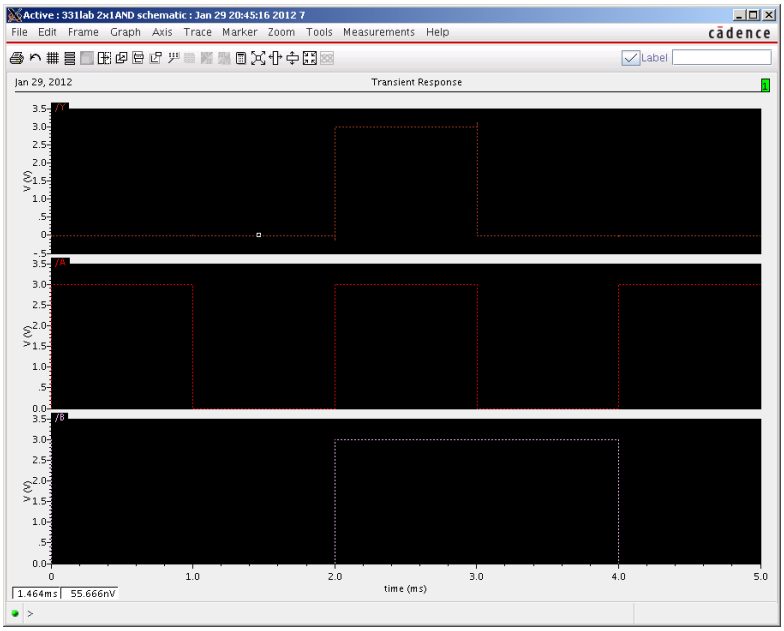
Running the Simulation:

1. In the ADE window, select Simulation => Netlist and Run to start the simulation using all of the parameters you just setup. When the simulation is complete, the VCL log should show "reading simulation data ..... successful" (although there may be lines after this too). If the simulation was not successful, go to Simulation => Output Log in the ADE window to find out what the problem was.

2. The simulation results will be plotted in a waveform window. Initially, all plotted signals will appear on top of each other. To separate the signals, select Axis => Strips in the waveform window.

3. Under Part 1 of the Lab 1 Check-off Sheet, enter the expected output values (0 or 1) in the **Theoretical** column for each input combination of an AND gate. Hint: only one of the rows should be '1'! :D

4. Now, view the plotted signals in the waveform window. Observe the output value (0 or 1) for each A,B input combination (0, 0), etc. Note: the inputs will not be in the save order as the truth table. Verify the simulation results match the expected/theoretical values. If not, look back at your circuit schematic and try to determine what is wrong. Re-run simulations until you get the proper response. Consult the TA if you need help.



Simulation waveform window (with slightly different stimulus pattern than defined above)

5. When your AND gate is simulating correctly, show the TA your schematic, symbol, and simulation results. Ask the TA to sign off on Part 1 of your Check-off Sheet.

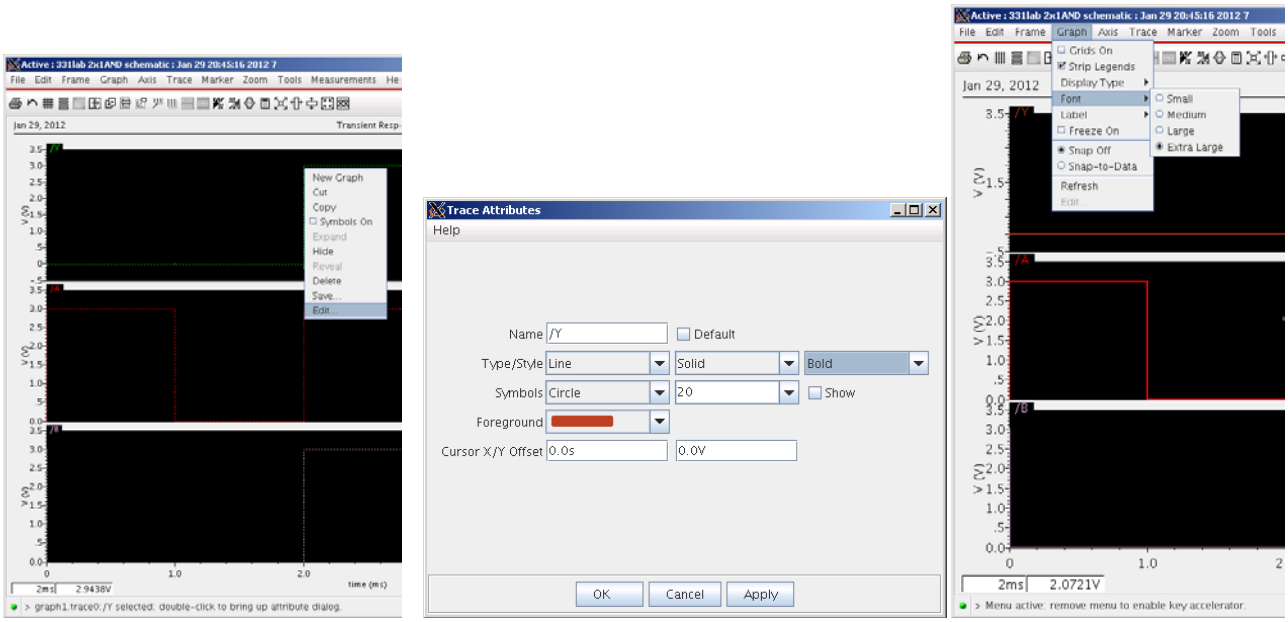
6. To close all Virtuoso windows, it is best to close them individually to ensure everything has been saved. Close the ADE by selecting Session => Quit. Close the Schematic Editing window by clicking File => Close (or Close All). To complete the rest the lab assignment you may not want to close the main VCL window yet, but when you do simply select File => Exit. You will then need to type "exit" into the PuTTY command prompt for each terminal you have logged into (until the PuTTY window closes).

You have now completed the tutorial and can continue with the next step of your lab assignment. Please note, the following appendix provides valuable information regarding saving images (of schematics, simulation plots, etc.). You will need to perform this task later. You can either try it now for practice or refer to this tutorial in later labs when these steps are required. A second appendix provides instruction on saving and loading simulation states that might save you some time in the future. Just remember that this information is in this tutorial.

Appendix A: Saving Image Files

In order to more clearly see plot results, the graph text can be made larger and plot lines can be made thicker.

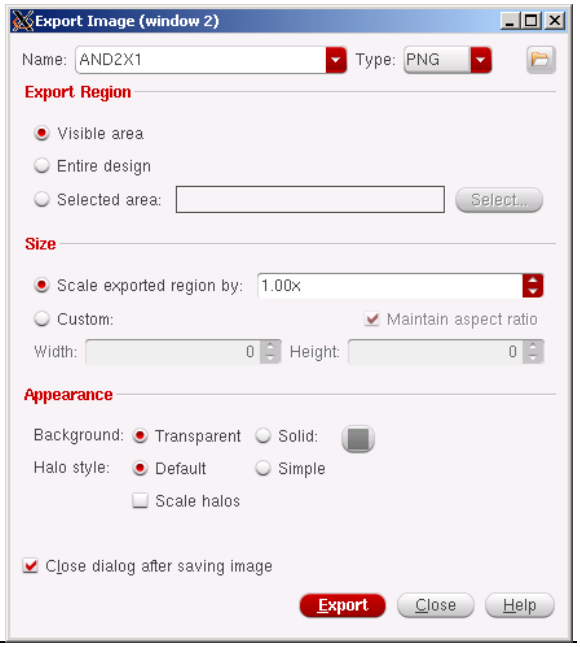
- To make the plot line thicker, in the Waveform Window , select each signal trace (individually) then right click on the selected line and choosing Edit. The options in the Trace Attributes window are shown below and are self explanatory. Try changing all traces to Solid and Bold style.
- To make the text on the graph larger, in the Waveform Window select on (for example) Graph => Font => Extra Large.



Adjusting waveform display parameters

In order to save a graph or schematic as an image file, for example to include within a report, follow the steps below.

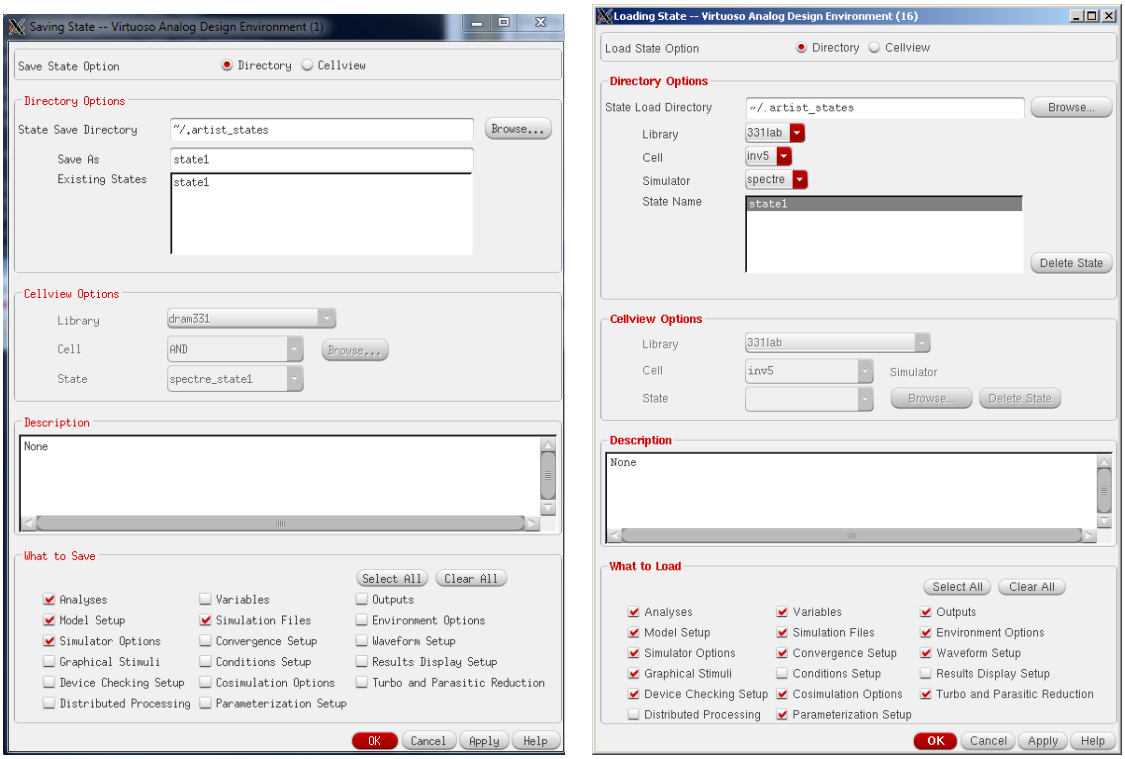
- In the Waveform Window click on File => Save as Image. In the configuration window that opens, you can set the image type (PNG, TIFF, or BMP), choose to convert the black background to white, and choose a filename to save to. The image file will be saved in your current working directory (from which you initially launched Virtuoso) unless you browse to a new location.
- To save a schematic as an image file, in the Schematic Editing window select File => Export Image. The configuration window to the right will open. Enter in the name at the top of the window, select the image type (PNG by default) and select the Background to be Transparent (generally preferred over a black background). Click Export and the image will be saved in your virtuoso working directory.



Appendix B: Saving Simulation States

Analog Design Environment states can be saved so that the settings do not need to be manually entered each time a design is simulated. However, the default settings can consume a lot of disc space so be sure you study your options a bit and monitor your disc usage if you choose to use this feature.

- To save the simulation state, in the Analog Design Environment window select Session => Save State. The “Save As” field does not have to be a unique name for all designs; the state is saved within the directory of the current cellview. Thus, you can use the same *Save As* name for multiple cells without overwriting each other. For example, cells named inv and nand2 can both independently have the state “state1” saved.
- Enter a *Save As* name, e.g., “state1” and click on OK. Note, by default the states will be saved at your root directory in a folder titled “.artist_states”.
- Saving all simulation data can take a lot of memory so you may find it useful to alter the *What to Save* parameters to save only the items you need to run future simulations. Saving outputs from complex cells with multiple plotted node waveforms can generate very large (GB!) files. A good starting point would be to save only the parameters that were set during this tutorial, such as Analyses, Model Setup, Simulator Options, and Stimulation Files.
- To load the saved state, after opening a specific cell, in the Analog Design Environment window select Session => Load State. Select the desired State Name and click on OK.



Save (left) and load (right) simulation states windows

THE END