Cadence Tutorial A: Schematic Entry and Functional Simulation

Created for the MSU VLSI program by Professor A. Mason and the AMSaC lab group.

Revision Notes:

Jan. 2006	Updated for use with spectre simulator	C. Wallace
Aug. 2003	update and edit add intro/revision/contents sections standardize document format for all tutorials	A. Mason
Jan. 2003	modify simulation section	M. Parr
Aug. 2002	update figures for AMI06 process	J. Zhang
Jan. 2002	create original tutorial	K. Zhang & A. Mason

Document Contents

Introduction

This document is one of a three-part tutorial for using CADENCE Custom IC Design Tools (IC445) for a typical bottom-up digital circuit design flow with the AMIC5N process technology and NCSU design kit. This document, Tutorial A, covers setup of the Cadence environment on a UNIX platform, use of the Virtuoso schematic entry tool, and use of the Affirma analog simulation tool. Tutorial B and C cover other Cadence tools important for custom IC design.
Note: Your paths may be different depending on the class or project you are working on. Also note that you can find additional tutorials and notes on the web from courses at other universities. These may be helpful in learning Cadence, but because of differences in the environment setup, you probably will not be able to follow a different tutorial step by step.

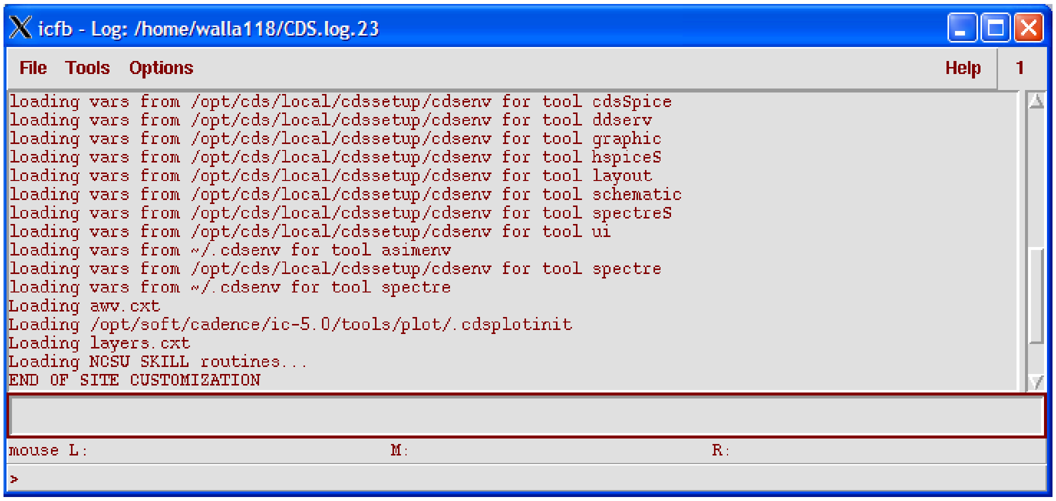
For more information about Cadence Virtuoso or the Affirma tool, see the manuals.

Environment Setup

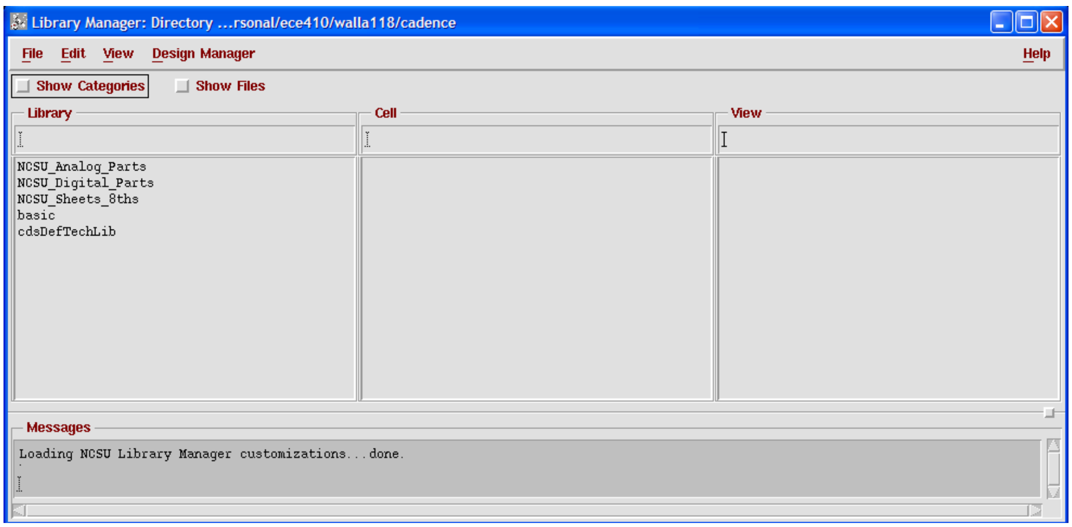
Before beginning this tutorial you must setup Cadence to work with your account. The steps for doing this may vary with each class/project, so be sure to follow any class-specific setup steps before proceeding with this tutorial.

If you have not already done so, launch Cadence now by going to your working directory and typing `icfb&` at the command prompt. A Command Interpreter Window (CIW) similar to the example below will appear. When all the configuration files have been read, the END OF SITE CUSTOMIZATION message will be displayed indicating the start up was successful. With each new session, Cadence starts a new CDS.log file in your home directory where all the messages that appear in the CIW will be stored.

Along with the CIW window, you should also see the Library Manager window that lists the libraries in your working directory. For now the NCSU-Analog-Parts library is the important one since it has basic circuit elements like transistors, current sources, voltage sources, ground, resistors, capacitors etc.



Command Interpreter Window



Library Manager Window

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 anytime during this tutorial you want to quit Cadence, make sure you save your work by selecting Design => Save and close the design windows by selecting Close from the menu. After you have closed all your working windows, select File => Exit and click Yes in the pop-up confirmation window to end the Cadence session.

Cadence File Organization

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

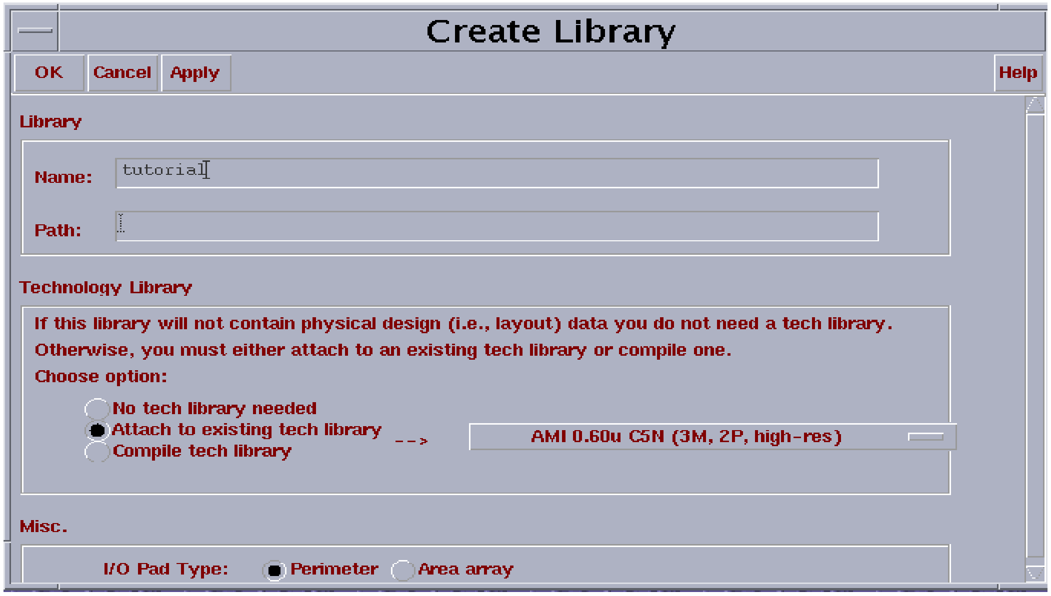
A library (which actually appears as a directory in UNIX) contains cells (subdirectories), which in turn contain views. Each library contains a catalog of all cells, viewed along with the actual UNIX paths to the data files. Each cell in a library uses the same mask layers, colors, design rules, symbolic devices, and parameter values (i.e. the information contained in the technology file). A *cell* is the basic design object. It forms an individual building block of a chip or system. 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

The Custom Design Process
For a full custom design (as opposed to a coded/synthesized design using, e.g., Verilog HDL), the design process begins by creating a schematic. The schematic is then simulated to verify operation and optimize performance. A layout of the circuit is generated and checked for design rule violations (DRC). The layout is then extracted and a layout vs. schematic (LVS) comparison is run to ensure the cell layout exactly matches the schematic. Finally the extracted layout is simulated to observe the effect of *parasitics* that will be present on the fabricated chip. These post layout simulation are closer to reality and will show if your design would work if fabricated. In this tutorial you will create a schematic for a basic digital logic gate, and inverter, and perform some basic simulations on the schematic to verify it is functioning properly.

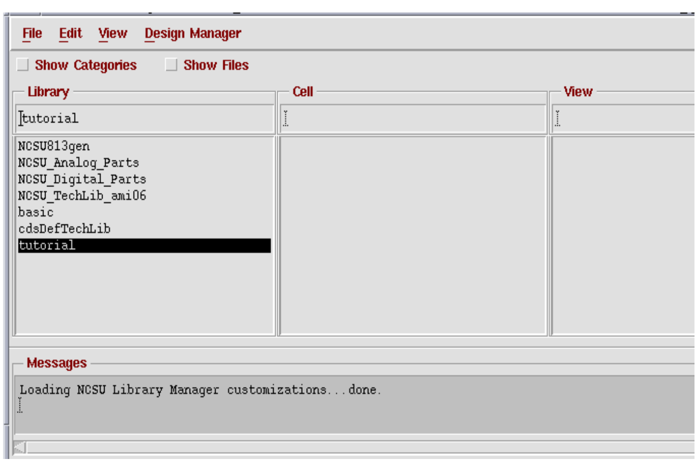
Creating a Design Library
STEP 1.
- In the Library Manager window, select File => New => Library to open the Create Library window shown below.
- Enter a Name for your library. The example shows the library name *tutorial*, but since you will probably be using the cells in this library (and adding more to it) choose a name like *lab*, *cmos*, or *digital*. Leave the Path blank.
- Click on the Attach to Existing Tech Library button and choose AMI 0.6u C5N (3M, 2P, high-res) as the technology file to be associated with your new library.
- Finally, click on the OK button and you now have an empty library to start adding cells to.

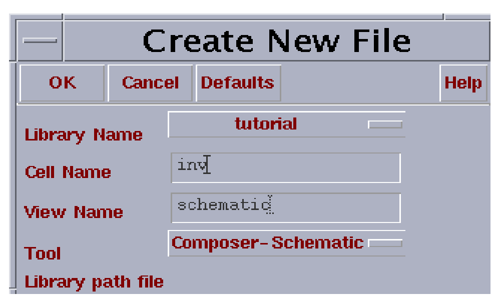


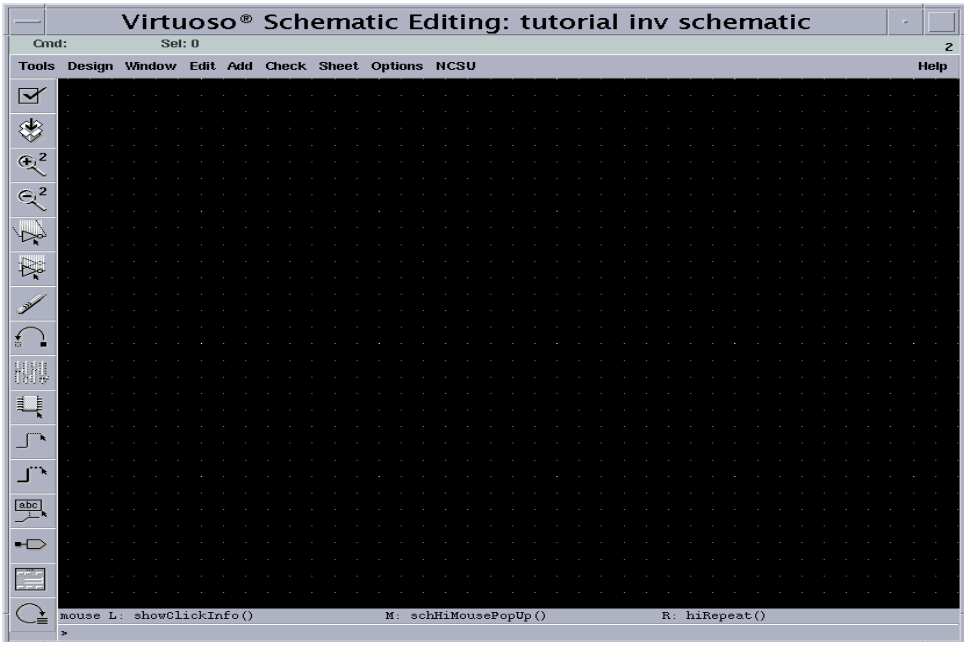
Creating a Schematic Cellview

STEP 2: Create a new schematic

- Go to the Library Manager window and click/select your library (for example "tutorial").
- Now select File => New => Cellview. Use the Create New File window that pops up to create the schematic view for an inverter cell.
- Enter the Cell Name "inv".
- Click on Tool drop-down list and select Composer-Schematic. 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.
- Click the OK button. The Virtuoso schematic editing tool will open with an empty Schematic Editing window as shown below.

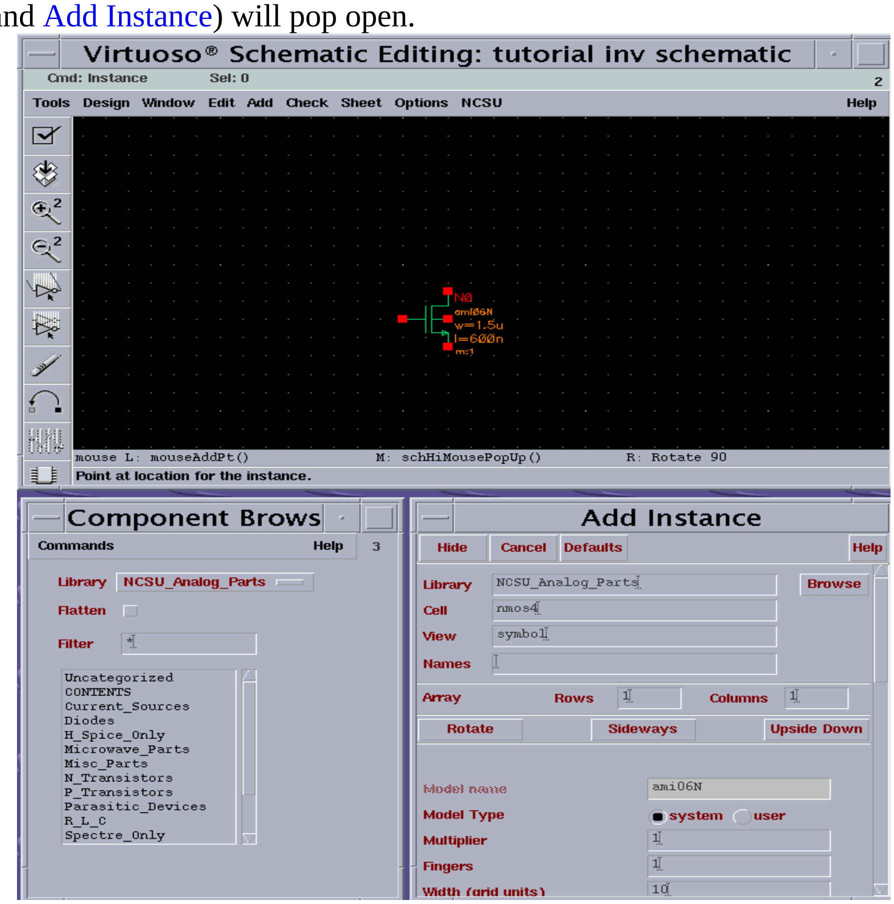






STEP 3: Add an nFET to the schematic

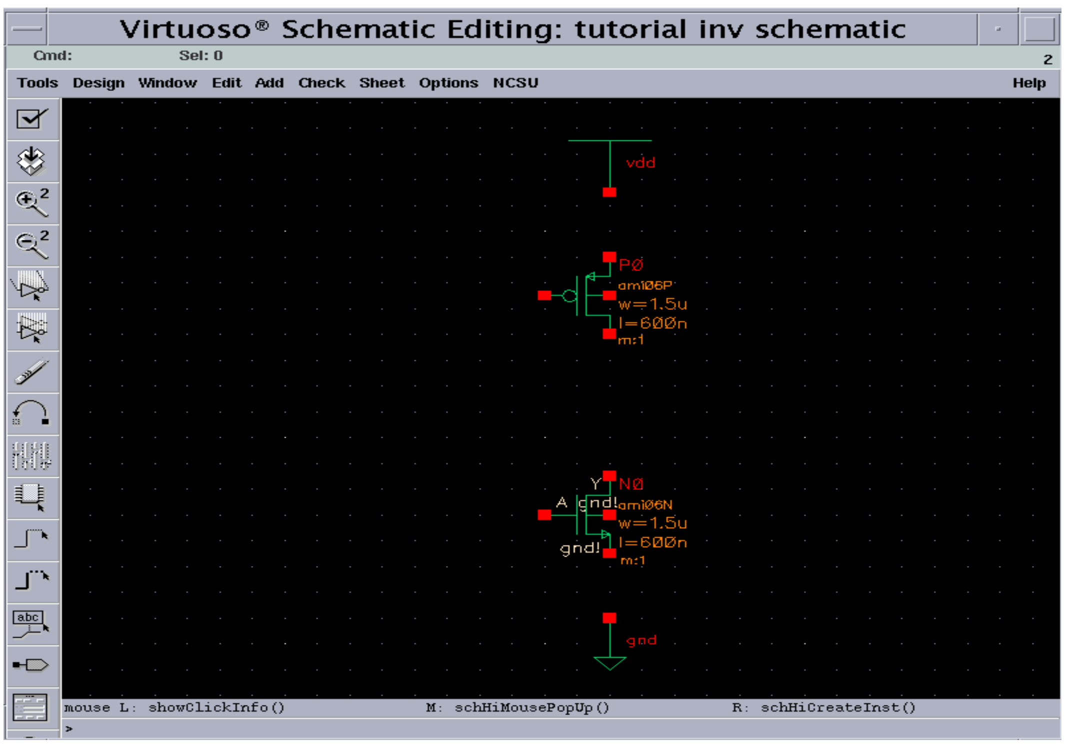
- In the Schematic Editing window Select Add => Instance to activate the Add Instance tool for adding components (transistors, sources, etc.) to your schematic. You can also invoke this tool by clicking on the *Instance* icon on the left-hand toolbar, or by typing the hot key ‘i’ with your mouse over the Schematic Editing window. Two windows (Component Browser window and Add Instance) will pop open.



- In the Component Browser window, under Library select NCSU_Analog_Parts. A list of parts will be displayed near the bottom of this window. From the parts list, click on N_transistors and a list of available nFET transistor elements will be displayed. Pick up the nmos4 element by clicking on it. This will attach the component to your mouse pointer.
- If necessary, click on the Rotate, Sideways, or Upside-down buttons in the Add Instance window to manipulate the component you are adding.
- Click on schematic area of the Schematic Editing window (main black area of the window) and the nmos4 component will appear in your schematic. Clicking on the schematic window again will add another copy of the component (don't do this). Pressing **ESC** on the keyboard will end the Add Instance function (but don't do this yet).

<u>STEP 4</u>: Add other components to the schematic

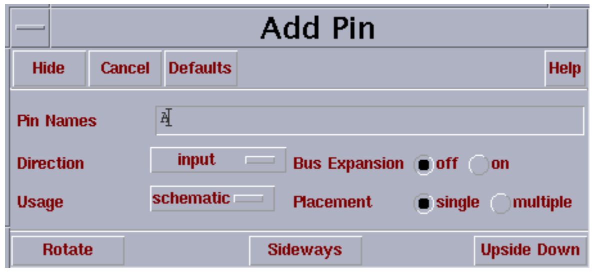
- In Component Browser window, go to the P-Transistors category in NCSU_Analog_Parts, and pick up pmos4 and add it to your schematic. Place it above the nFET as appropriate for a CMOS inverter. If you need to move a component, press '**ESC**' and left-click on the component and drag it to a new location. Pressing '**ESC**' will cancel the Add Instance command so you will need to restart it to add more components.
- In Component Browser window, go to the Supply-nets category and pick up vdd (supply voltage) and gnd (ground reference) and add them above and below, respectively, the MOSFET components. Now the Schematic Editing window look like the figure below.



<u>STEP 5</u>: Add I/O pins

- In the Schematic Editing window select Add => Pin (or use hot key 'p') to add an input *pin*.
- In the Add Pin window enter 'A' for the Pin Name and select input for the Direction.
- Click on the Schematic Editing window and drop the pin to the left of the transistors (between the gate inputs).

- Follow the same procedure to add output pin 'Y'. to the right of the transistors between the drain nodes. Be sure to use the correct pin direction. Press '**ESC**' when you are done.

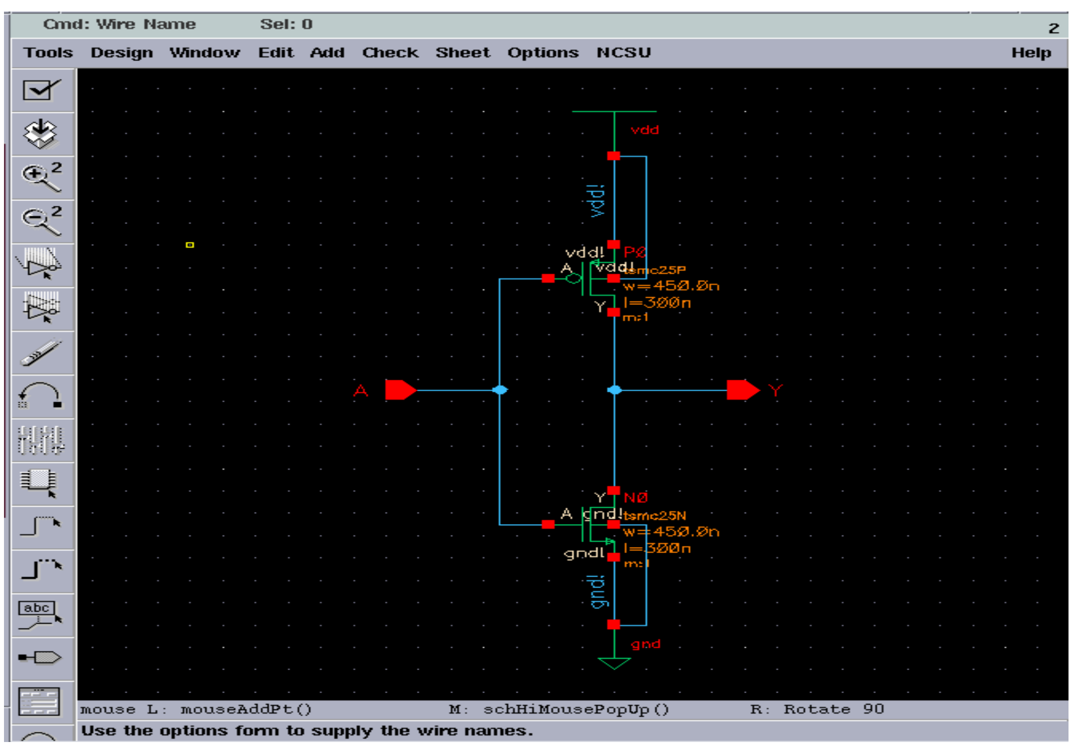


<u>STEP 6</u>: Add wire connections

- In the Schematic Editing window select Add => Wire (or use hot key 'w') to begin placing wires which will connect the terminals of the components and pins in the schematic.
- Wire the schematic as appropriate for a CMOS inverter. Be sure to connect the *body* terminal of each transistor (nMOS to ground, pMOS to VDD). Press **ESC** to end wiring.

<u>STEP 7</u>: Setting global labels

- In the Schematic Editing window type the letter 'l' to invoke the labeling tool. Since we will be using the same vdd and gnd supplies for all our cells, we want to make a ***global declaration*** for these labels. Cadence uses the "!" symbol after a label-name to indicate a global label, i.e., one that is common to all cells in a design.
- In the label pop-up window, enter the Name vdd!
- Move your mouse to the Schematic Editing window and drop the vdd! label on the wire that connects to the vdd pin.
- Repeat this procedure to label the global gnd! net. Press **ESC** when you are done.



Final schematic of an inverter

STEP 8: Check and Save the cellview

Now that you are familiar with the schematic editing tool, explore the menu commands in the Schematic Editing window to do the following:

- Check the cell for errors. If errors exist the error section will blink in the Schematic Editing window.
- Save the cellview.

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.

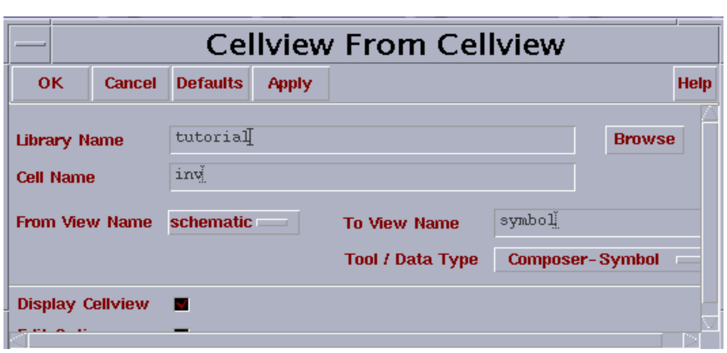
Undo Operations: When you make a mistake (accidentally delete a component, etc.), you can undo the action by click on the Undo icon in the toolbar.

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

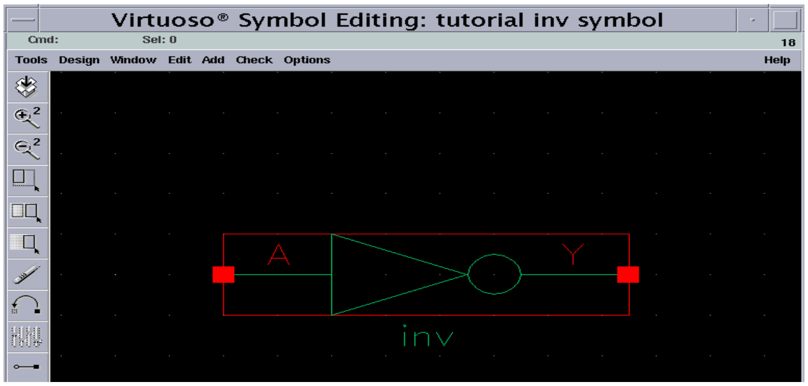
1). Press 'p' to add pins
2). Press 'q' on the device/instance to edit properties for the device
3). Press 'w' to add wires
4). Press 'f' fit the schematic in your schematic window
5). Press 'z' to zoom in the window
6). Press 'shift+z' to zoom out the window
7). Press 'l' to label a wire
8). Press 'Up' 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

STEP 9: Create a symbol

- In the Schematic Editing window, select Design => Create Cellview => From Cellview.
- In the Cellview From Cellview window that pops up, you should notice source (From View Name) set to *schematic* and the destination (To View Name) set to *symbol*, which is linked to the Composer-Symbol tool. You should not have to modify this window.
- Click on OK in the Cellview From Cellview window.



- The Symbol Editing window will pop up showing the default symbol, a rectangle with red square dots for input and out pins. You can keep this symbol, but it would be helpful if you modified it to a more meaningful symbol (such as a triangle for an inverter). Explore the options on this window and the tips below to define your symbol graphic.
- When the symbol is complete, save it. In the Symbol Editing window select Design => Save or click on the Save icon at the top of the toolbar.
- In the Symbol Editing window, select Window => Close to close the symbol.



Final inverter symbol

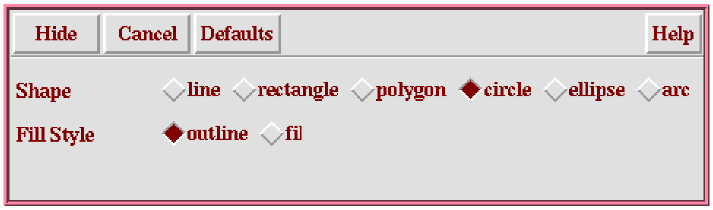
Symbol Editing Tips

General Notes:

- The red box around your symbol is called a selection box. When you place your symbol in a schematic, this box represents the area you can click in to select the symbol. You can change the size of selection box. It is a good practice to fit the symbol entirely within the red box; otherwise, you may not easily be able to select it when you instantiate it later on.
- The red square dots indicate the pin connections.
- [@InstanceName] and [@PartName] are display variables, which you may keep or delete. These are generally annoying for small cells/symbols but can be useful for higher-level circuits.

Drawing Lines: To draw a line, move your cursor to the Line icon (the icon with a descending line) and click on the left mouse button. Icons are to the left of the symbol window. Move your cursor to the location where you want to draw a line and click on the left mouse button. If you draw a closed object, all you have to do is to click on the points where you want to change the direction of the line. However, if you are drawing just a straight line, you need to double click the left mouse button on the end point of the line to signify the completion of drawing the line.

Drawing Shapes: To draw a circle (or any other shape), first click on the Line icon and then click on the options icon (the second to last icon). A pop-up menu will appear in which you can select different shapes and fill styles. Then move your cursor to the location where you want to draw the circle/shape and left-click. That is the location of the center of your circle. Move your mouse so that the circle is as big as you desire it to be and then click on the left mouse button again.

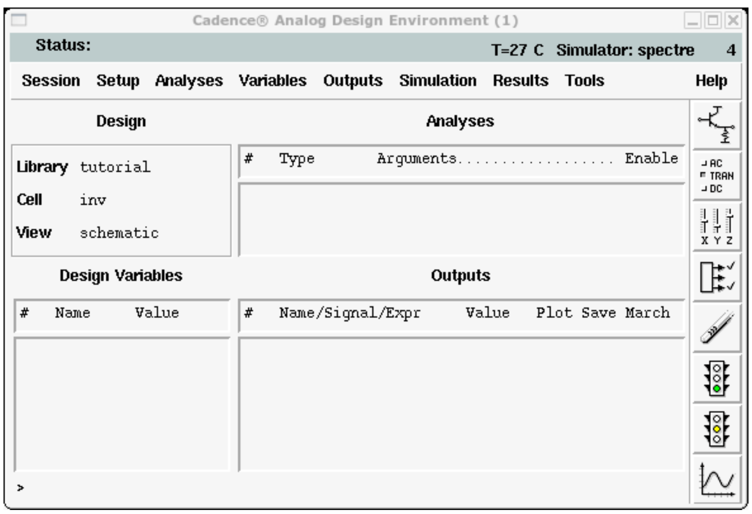


Simulation with Affirma Analog Circuit Design Environment

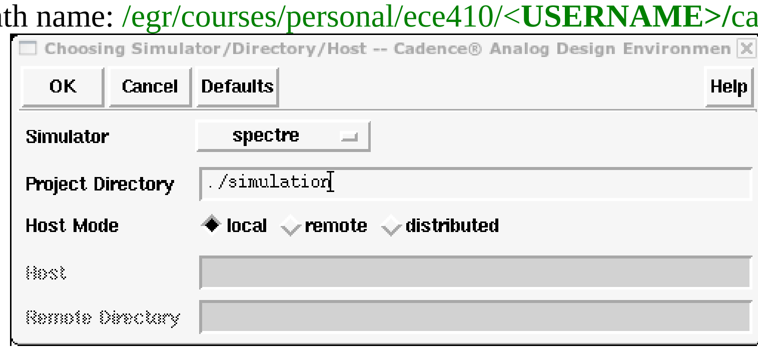
To verify a circuit is working and test the functionality of the schematic we must simulate the circuit. For this we will use the Cadence Affirma analog simulation tool. For now we will just run a simple transient analysis to confirm the circuit designed above is operating as an inverter should be. Additional information for running simulations with Cadence tools is provided in Tutorial C.

<u>STEP 10</u>: Launch the Affirma analog simulation tool

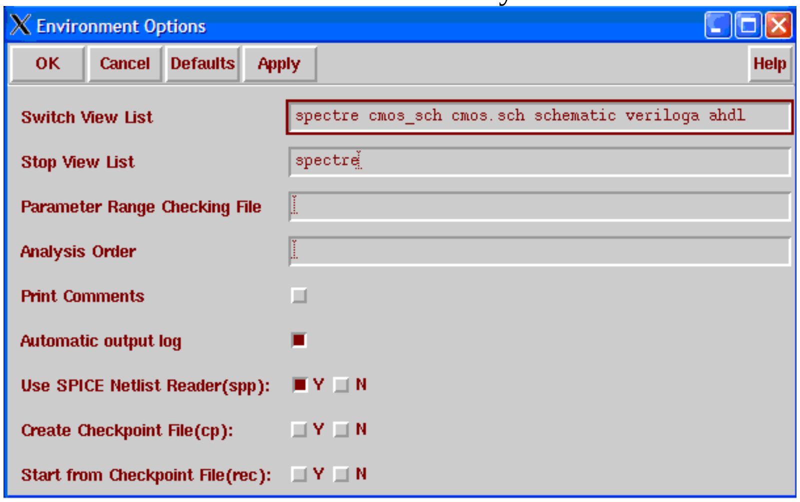
- With your "inv" schematic open, in the Schematic Editing window select Tools => Analog Environment, and the Affirma Analog Circuit Design Environment window will open.
  - o You can also launch this tool from the CIW by selecting Tools => Analog Environment => Simulation in the CIW. When the Affirma Analog Circuit Design Environment opens you have click on the Setup => Design to specify the library and cell, for example "tutorial" and "inv".



- In Affirma Analog Circuit Design Environment, click on Setup => Simulator/Directory/Host.
- Choose **spectre** as the Simulator (if not default).
- The default Project Directory should be "./simulation". This points to a directory named "simulation" inside of the directory you launched Cadence from. This is where your simulations files and results will be saved. You may set the Project Directory to any valid path, but you might find it useful to keep all simulations in one directory. To run multiple simulations on the same cell, you can use different paths for each simulation. Note that upon closing and reopening this screen, the project directory will automatically be transformed to the absolute path name: /egr/courses/personal/ece410/**<USERNAME>**/cadence/simulation/



- In Affirma Analog Circuit Design Environment, click on Setup => Environment.
- Next to "Use SPICE Netlist Reader(spp):", click the box "Y". This is necessary because the AMI06 transistor model libraries use SPICE netlist syntax. Click OK to close this window.



<u>STEP 11</u>: Set up 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 define 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 ECE410 Cadence directory. If you are new to UNIX, you may want to "NEdit" by doing the following:

- Open a terminal screen (or use any one that is already open).
- Make sure you are in your Cadence directory. If unsure, type
  `cd /egr/courses/personal/ece410/<USERNAME>/cadence`
- Type "`nedit &`" to create a new file.

- Once you have opened a text editor, type in the following three lines to define the simulator language (*spectre*), the DC supply voltage and a square wave pulse input voltage that will be useful for a transient analysis simulation to verify functionality of the inverter circuit. Note that the input waveform used may vary with the type of analysis needed, although the supply voltage source will generally remain constant.
  - `simulator lang=spectre`
  - `vdd (vdd! 0) vsource dc=3`
  - `V1 (A 0) vsource type=pulse val0=0 val1=3 delay=0 rise=0.05n fall=0.05n width=10n period=20n`
- Note: Each bulleted item represents one line in your stimulus file. The third bullet item (voltage pulse) is broken into two lines due to the document margins, but should only take up one 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.

- Save the text file. If you are using *Nedit*, this can be done by
  - Selecting Files => Save As
  - In the "New File Name:" box, the current path will be displayed. Click on the blinking cursor and add "stimulus.txt" to the path.
  - Click OK. This will save the file
- You can exit the text editor at this point (nedit: select File=>Exit). However, you may want to keep it open if you plan on making changes to the file during simulation.

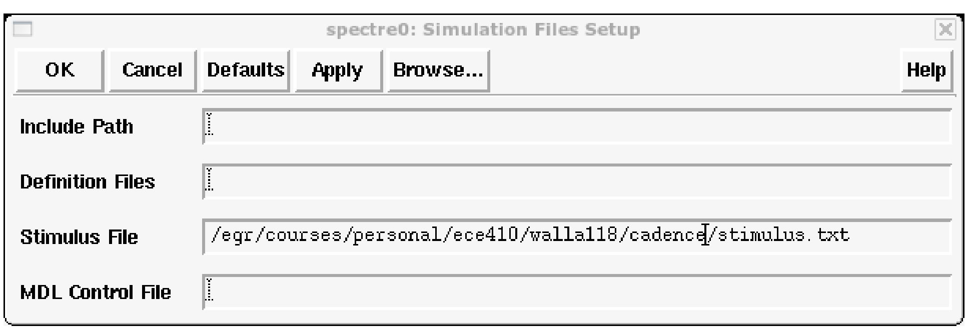
Stimulus File Tips

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=min_voltage val1=max_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 listed are good for the AMI-C5N technology and should not be modified unless you are sure you know what you are doing.

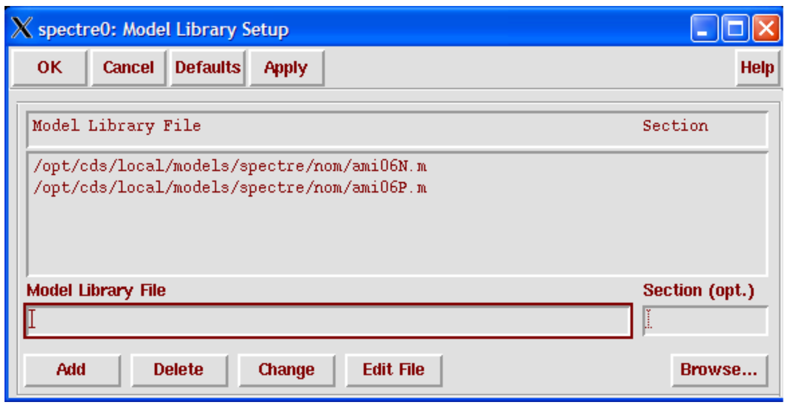
- Once you have written a stimulus file, open (refocus) the Affirma Analog Circuit Design Environment window. Click on Setup => Simulation Files.
- Click on the box that is labeled "Stimulus File" so that a blinking cursor appears in it.
- Click the Browse… button to open a file browser window.
- Navigate to where you have saved the file "stimulus.txt" – if you have followed the tutorial exactly, it should appear in the default directory opened by the file browser.
- Click on the name "stimulus.txt" and then click OK. The file browser should close, and the absolute pathname for your stimulus file should appear in the "Stimulus File" box.
- Click OK to apply changes and close this window.



STEP 12: Setup Model Files

- In Affirma Analog Circuit Design Environment, click on Setup => Model Libraries…
- If you have correctly set up your ECE410 Cadence environment, two Model Library Files should already appear, "ami06N.m" and "ami06P.m". If so, you may exit this dialog by pressing OK.

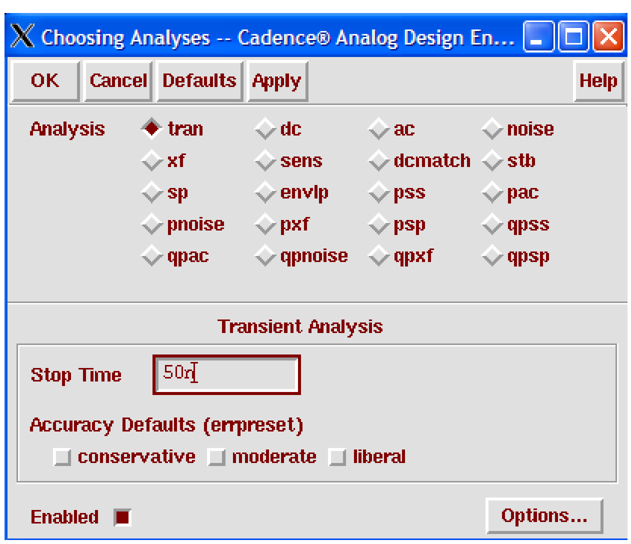
- You can add the necessary Model Library Files by
  - o Click the bottom "Model Library File" box
  - o Type "/opt/cds/local/models/spectre/nom/ami06N.m"
  - o Click the "Add" button. This should add the model to the Model Library File list and clear the Model Library File box
  - o Repeat the process or "/opt/cds/local/models/spectre/nom/ami06P.m"
  - o Your Model Library Setup dialog box should look like the one below. Once it does, you can click "OK" to close the dialog.



<u>STEP 13</u>: Setup analysis

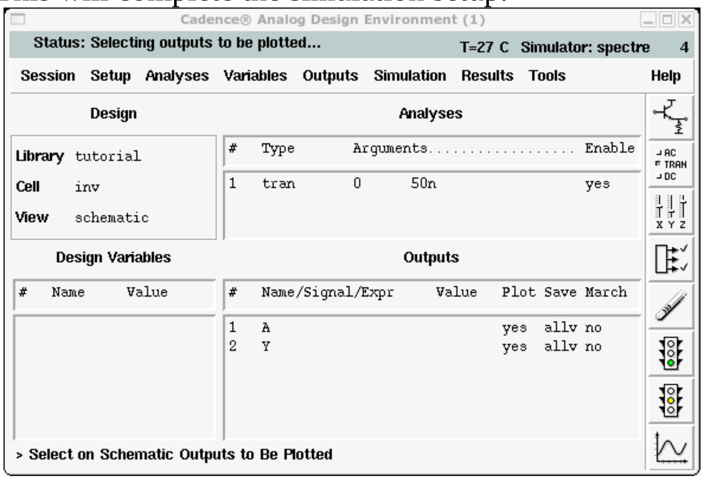
With the stimulus defined, we need to choose what type of analysis we wish to perform. Different analyses are useful for getting a variety of different data from the simulations. Here we wish to simulate the response of our inverter to a simple square wave input and verify that the output is indeed being inverted. For this we will setup a ***transient*** (over time) analysis. Alternative analysis types, including DC analysis, that are used to simulate different performance characteristics of a circuit are covered in Tutorial C.

- In Affirma Analog Circuit Design Environment window, select Analyses => Choose.
- In the window that pops up, select **tran** to choose a transient analysis.
- Enter the time limits for simulation by entering a *Stop Time* of "**50n**".
- Choose Enabled at the bottom of the screen and press OK.



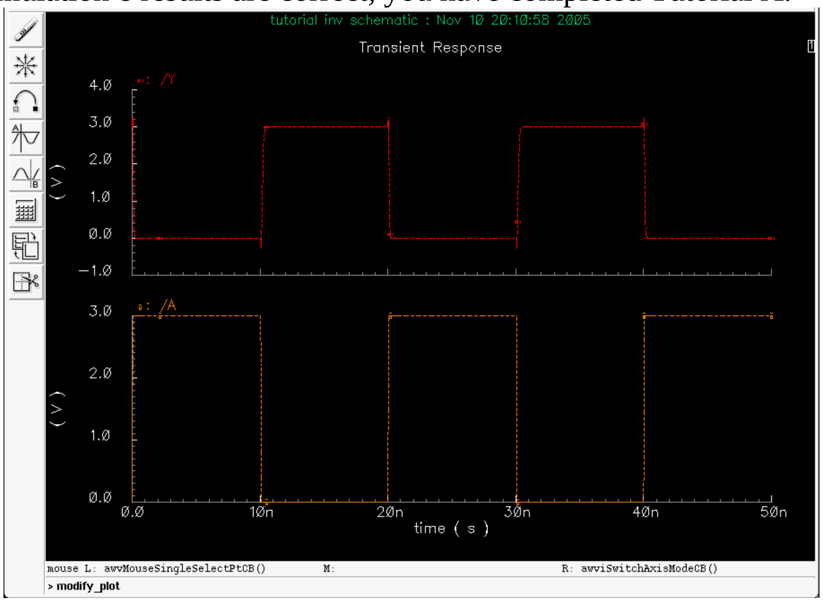
STEP 13: **Setup output traces**

- In Affirma Analog Circuit Design Environment window, select Outputs => to be plotted => Select on Schematic. This will activate the Schematic Window allowing you to pick which signals (nets/wires) you would like to have plotted during the simulation.
- In the Schematic Window click on the wire that is the input to your inverter and also click on the output wire. This will complete the simulation setup.



STEP 14: **Run simulation**

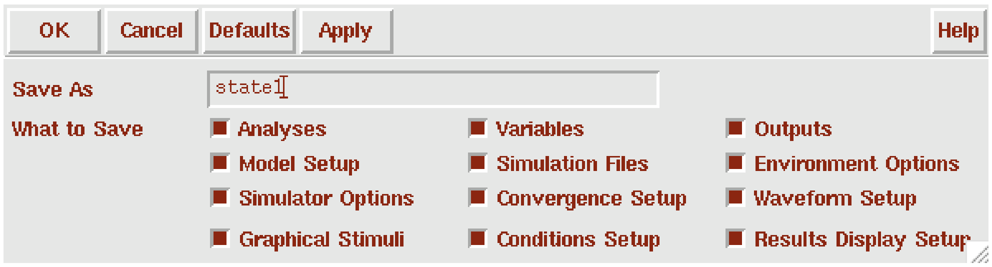
- In the Affirma Analog Circuit Design Environment window select Simulation => Netlist and Run.
- When the simulation is complete, the CIW should show "Reading Simulation Data ...... Successful". If the simulation was not successful, go to Simulation => Output Log in your Affirma Analog Circuit Design Environment to find out what the problem was.
- If the simulation results shows a typical inverter response (i.e. output high when input is low and visa versa) then the circuit is working properly. If not, you will need to go back to the schematic to track down the problem and rerun simulations until you get the proper response.
- To separate input and output signals, in the Waveform Window click on Axes => To Strip.
- When this simulation's results are correct, you have completed Tutorial A.



<u>Optional</u>: Saving and loading the simulation state

The Analog Design Environment state can be saved so that the settings do not need to be manually entered each time a design is simulated.

- To save the simulation state, in the Affirma Analog Circuit Design Environment window select Session => Save State

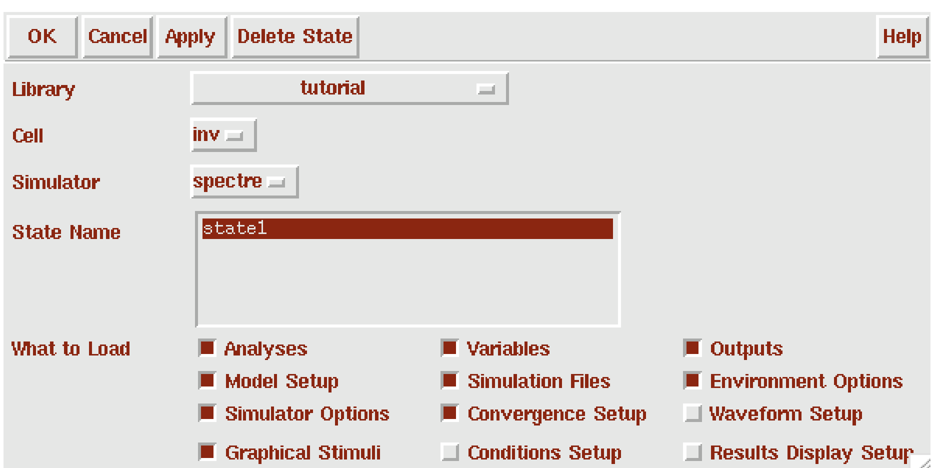


The "Save As" field does not have to be a unique name for all designs; the state is saved within the directory of the current cell view, so you can use the same Save As name for multiple cells without overwriting each other.. For example, cells named inv and nand2 can both independently have the state "state1" saved.

- Enter a Save As name, e.g., "state1" and click on "OK".

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 simulatoins. Saving outputs from complex cells with multiple plotted node waveforms can generate very large files.

- To load the saved state, after opening a specific cell, in the Affirma Analog Circuit Design Environment window select Session => Load State



- Select the desired State Name and click on "OK".

THE END