

Lab 1: Cadence[®] Custom IC design tools - Setup, Schematic capture and simulation

Objective

The objective of this lab is to familiarize you with the basics of the Cadence[®] Custom IC design tool, Virtuoso[®]. You will first set up your account to run the IC tool, learn how to manage your files with the Library Manager, understand the basics of the Schematics Editor, and simulate a simple circuit using the Analog Design Environment (ADE).

1. Setup



If you have never used Unix operating system before, first familiarize yourself with some of the basic Unix commands provided in Appendix A.

Once you have successfully logged into your account on a Linux machine, you need to take a few steps before you can start using the IC design tools.

A) Create a `~/.software` file (Only once)

The computer support group (CSG) has made a variety of software available for student use on the system. To enable the use of different software tools, every user needs to have a special file `.software` (the “.” is required!) in their home directory customized for their use. Using a text editor of your choice, create a new file by that name in your home directory and enter the following lines in that file:

```
IUS
ASSURA
IC
```

These three entries enable the use of the IC design tools with analog and digital design flow capability. You will need to log out and log in for the new file to take effect. If other tools are needed for a future lab, you may have to edit this file and add appropriate entries as directed by your TA.

B) Create a `~/cds.lib` file (only once)

The Library Manager (to be discussed in the next section) keeps information about system-wide libraries as well as your design libraries in a special file called `cds.lib` (the contents of `cds.lib` file are *case-sensitive*). For better organization purposes, you will create a `cds.lib` in your home directory. This file will include definitions of system wide libraries as well as any library that you would like to be available for *all* of your projects and classes. For starters, create the file with the following contents (remember to press enter at the end of second line):

```
INCLUDE $CDS_SITE/cdssetup/cds.lib
DEFINE analogLib $IC/tools/dfII/etc/cdslib/artist/analogLib
```

The first line includes a `cds.lib` file created by the system administrators from another location on the system. The second line defines a library “`analogLib`” supplied by Cadence which includes the basic circuit elements. Except for transistors, most of your schematic components will be instantiated from this library.



Cadence library names are case sensitive and spaces are not allowed. Avoid starting a library name with a number. Use underscores or mixed capitalization (e.g., `analogLib` is more readable than `analoglib`).

C) Create a project directory and a `cds.lib` file (for every project)

It is best to organize your work in multiple directories with each directory being dedicated to a specific class or to a research project. We will create a new directory where all of our libraries will be stored for this course.

```
mkdir ee434
```

Change into the new directory using the “`cd ee434`” command. Now create a `cds.lib` file in with the following one line as its contents (remember to press enter at the end of the line).

```
INCLUDE ~/cds.lib
```

This line will include the contents of our `~/cds.lib` file giving us access to the common libraries defined there. Whenever you launch Cadence IC tool from `~/ee434` and create any new libraries, this file will be automatically updated to include the definitions of the new libraries.

The basic setup for running the Cadence Custom IC design tools is now complete. Refer to this section whenever you start a new project and need to set up a new directory.

2. Starting Cadence Custom IC tool and creating libraries

After making sure that you are in the `ee434` directory you created earlier, type the command “`icfb`” as shown below.

```
icfb &
```

This will start the Cadence custom IC design tool in the background and, after a brief delay, you will see the Command Interpreter Window (CIW), as shown in Fig. 1. All errors and warnings are communicated to the user through the CIW, so make sure you always use it as your guide when troubleshooting.

In addition to the CIW, you will see two pop-up windows: the “What's New” and the “Welcome” windows. These windows provide you information about the Cadence tools’ new features and announcements from CSG, respectively. After reading, you can close them.



You can turn off the “What's New” window for future sessions by clicking on **Edit→Off at Startup...**

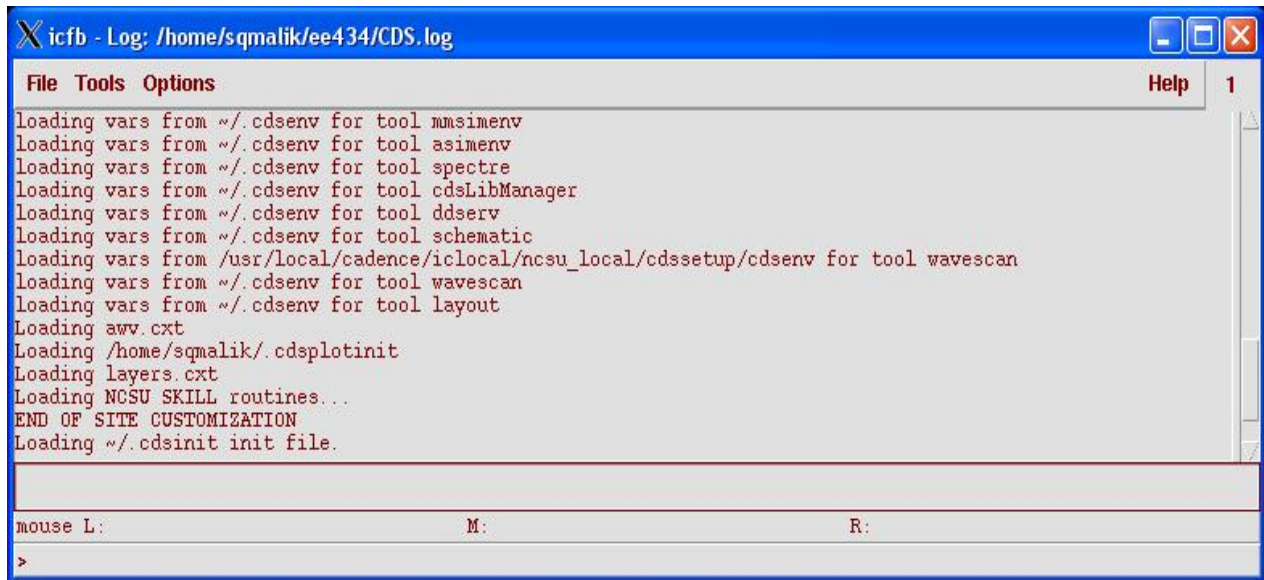


Figure 1: Command Interpreter Window (CIW)

At the startup, you will also see a Library Manager window, as shown in Fig. 2. The library manager is used to organize the data into libraries and cells. If you don't see the Library Manager window, go to CIW and click on **Tools**→**Library Manager**.

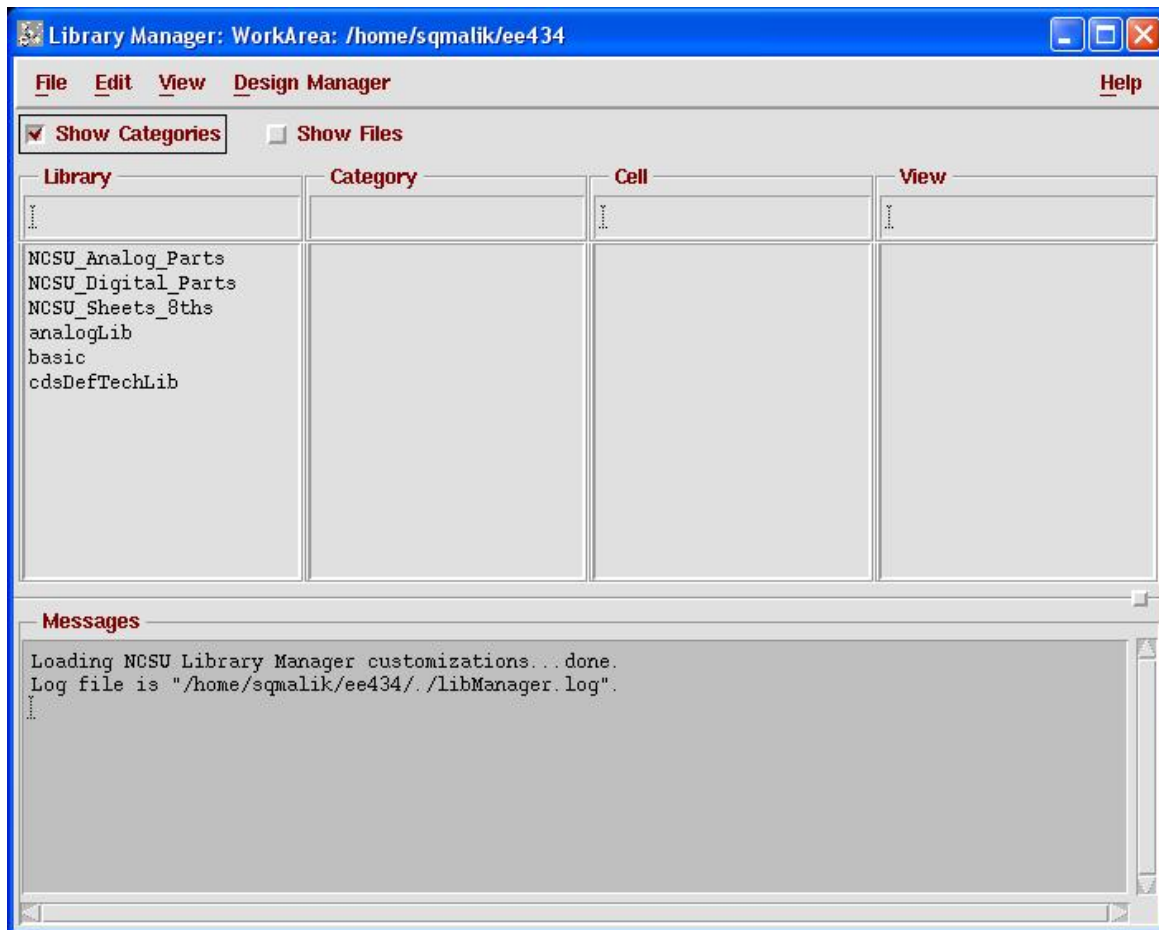


Figure 2: Library Manger

A) Creating a new library

We will group our designs in Libraries. Each library is essentially a folder to group relevant sub-folders together. Each Library can contain multiple “Cells” and each cell can have multiple “Views”. Let us first create a new library called “lablib” (you can use a different name if you prefer) and create cells for each lab in the library. To create a new library, go to the Library Manager and click on **File**→**New**→**Library**. In the form that pops up, type in the name of the library and click on OK. Leaving the path field empty creates the new library in the directory from which icfb was launched. A new library by the name you chose will appear in the library manager.

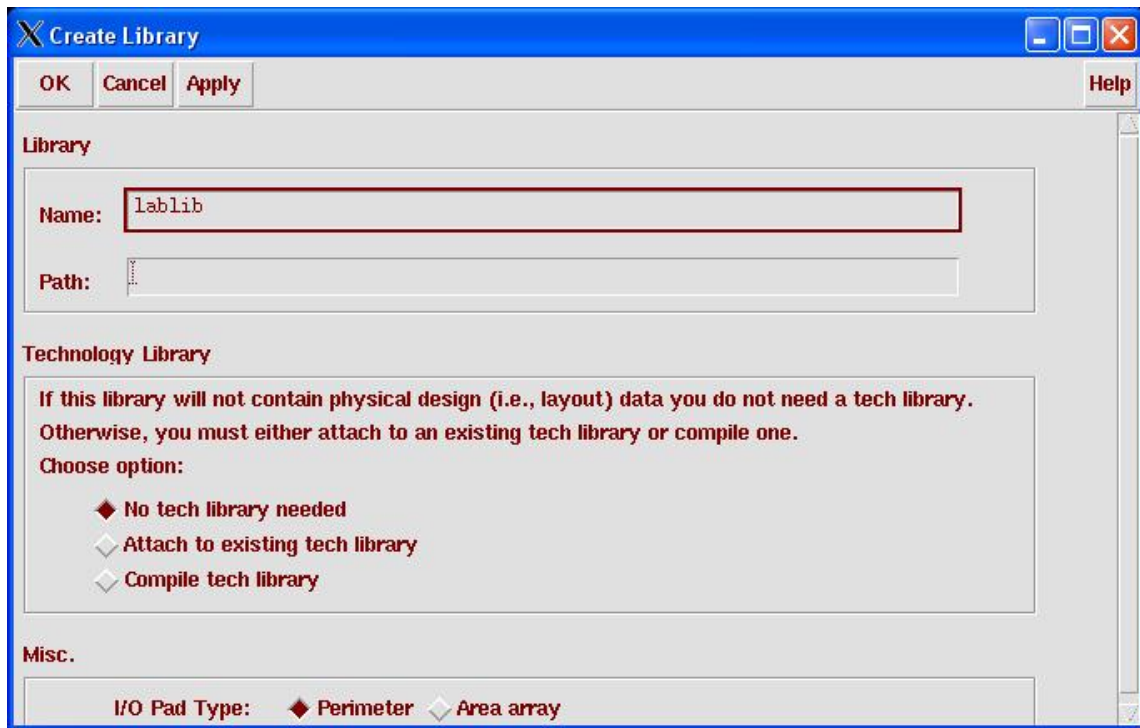


Figure 3 : Form to create a new library

B) Creating a new cell and a cell view

A cell in Cadence can represent any level of the hierarchy of a design. At the bottom level of the hierarchy, a cell may contain a single transistor or other basic circuit elements. At the top level of the hierarchy, a cell may contain interconnected cells to describe a complete system. In turn, each cell has at least one cell view. One way of looking at a cell is to think of it as a dice; The same dice shows a different number depending on how you **view** the dice. For our purposes, a cell called “inverter” could have a view called behavioral (describing the behavior of an inverter in a programming language), a schematic (describing the behavior of an inverter using transistors), a symbol (a view allowing the inverter to be re-used in other designs), etc. To create a new cell and one of its cell views simultaneously, first select the library in which you want the new cell to reside. Next, click on **File**→**New**→**Cell View...** in the Library manager, select type the name of the new cell and choose the desired view. For example, create a new cell called **lab1** with **schematic** view in the **lablib** library. The software will automatically launch the appropriate editor for the chosen view.

The first step in a new design is typically something you scribble on a notepad. Once you have a good idea of what is that you are trying to design and how that circuit should work, the next step is

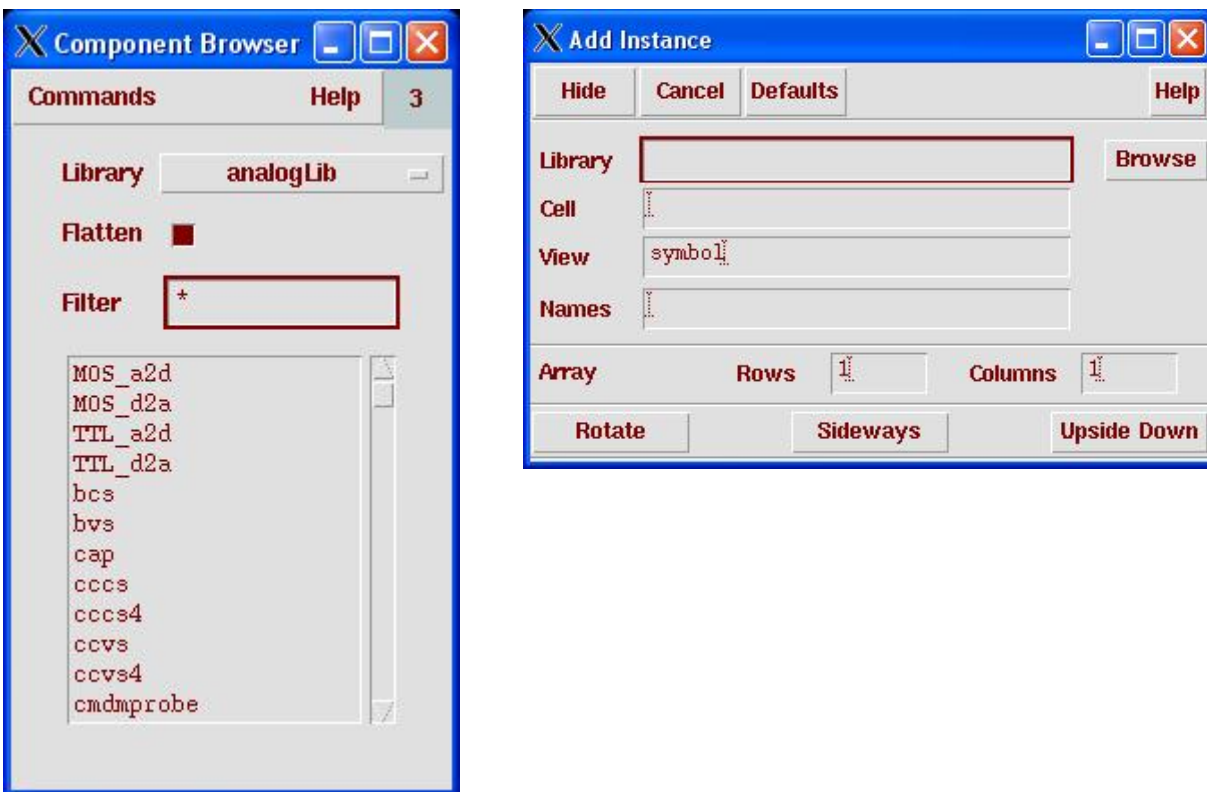
to verify your idea using a simulator. The next section describes the basics of the schematic capture tool in Virtuoso used in to prepare a circuit for simulation.

3. Schematic Capture

To practice schematic capture, we will enter the circuit shown in the Pre-lab section. You should already have the schematic editor window open editing the schematic view of the cell lab1. The first step is to instantiate all the required components.

A) Instantiating the components

To place all the components in the schematic window, click on menu item **Add→Instance**. You will see two windows popping up on the screen (shown below): “Component browser” and “Add Instance”.



In the “Component Browser”, change the library to “analogLib” and click on “Flatten” to see all the cells without categorization. Now find the resistor cell and click on it (The resistor cell is called res). Notice how the “Add Instance” form changes to reflect the selection of resistor with a default value of 1K. Change the value of the resistor to 10K. Do not click in the Schematic window yet! Just move your cursor over an empty space in the schematic window. You will see a yellow colored resistor “floating” with the cursor. If you click anywhere in the window, a copy of the resistor will be placed. Try it. If you need a component at a different orientation, click on “Rotate”, “Sideways”, or “Upside Down” in the “Add Instance” form.



You can use icons from the icon bar in the schematic window for common editing tasks. Let your mouse pointer hover over an icon to find that icon's function. As practice, locate the "instance" icon.

All the resistors that are placed in the schematic window will have resistance of 10K. What if you needed a different valued resistor? Just click on the "Add Instance" form, change the value, and then click in the window to get a component with the adjusted value.

Once you have enough resistors, instantiate a capacitor (cap), ground (gnd), and a voltage source (vsource) from analogLib. Press "Esc" key to end the instantiation mode.




You can use keyboard shortcuts for common tasks. These are extremely handy and you can work much faster if you get proficient with them. For example, while in schematic editor window, pressing the "i" key brings up the "Add Instance" form. Keyboard shortcuts are listed next to their menu entries.

B) Connecting the components

There are two ways to interconnect components. First, use the menu entry **Add→Wire (narrow)** to get into the wiring mode (You could also have pressed the shortcut "w" key or clicked on the Add-Wire (narrow) icon). When in wiring mode, click on the red terminal of a component to start one end of the wire (you do not need to keep the button down). Now click on the other component's terminal where you want the connection to terminate. If you make a mistake and want to quit the wiring mode, press the "Escape" key on the keyboard. You can undo the last action by pressing "u".

The second way to add wiring is to first bring the mouse pointer over the red terminal of a component's terminal. While clicking and holding the mouse button, drag your mouse and you will see a wire started. You can let go of the button now and the wiring will continue. Hit "Escape" anytime to cancel. Unlike the first method, the wiring mode automatically terminates with the completion of one connection.

Complete wiring the circuit. To make sure there are no floating components or connection problems,

we need to "Check and Save" the design. Click on the  icon or the "Check and Save" entry from the "Design" menu. Check the CIW window to make sure the design was saved without errors.

C) Adding labels

As you wire the components together, every net is automatically assigned a name such as net1, net2, etc. These names are fine for the software but are not easy for us to remember. Adding labels to wires helps us to interact with the tools more efficiently. To add a label to a wire, click on the menu item **Add→Wire Name** (or use shortcut "l"). In the form, type **vIn vMid vOut**. Bring your mouse pointer into the schematic window. If you do not see the "floating" vIn attached to your cursor, you first need to click on the title bar of the schematic window to make it active. Now you should see the vIn attached to the mouse pointer. Click your mouse pointer on the net you wish to call vIn. Note how the next label from the list you typed in the form appears once you have placed the first label. This is a convenient way of adding multiple labels without having to go back and forth between the form and the schematic window.

Labels provide functionality in addition to being visual aids: connectivity. If two unconnected nets have the same name, they are considered to be electrically connected. This feature can allow you to create schematics with less clutter but can also be a source of mistakes. Use the capability of labels to create connectivity judiciously or your schematics may actually become less readable.

D) Editing object properties

What if you made a typing mistake when creating labels or need to modify the value of a component? You can make the required changes by clicking on the object to select it and then clicking **Edit**→**Properties**→**Objects** (find the shortcut for this if you don't want to click three menu levels every time!).

For practice, let us modify the properties of the voltage source component, vsource. The vsource component is capable of generating a variety of stimuli.



Tip: It is possible, and in many situations, even preferable to use other sources from analogLib such as vdc (when generating only a dc voltage), vsin (for a sine wave), etc. We will use vsource for input stimuli and other independent sources for power supplies and clock signals.

Bring up the properties form for vsource. In the next section, we will run dc, ac, and transient simulations. For those simulations to work, we need to provide appropriate input stimuli.

- In the “DC voltage” box, type 1. This dc voltage is used to compute the quiescent operating point of the circuit. Click on apply (at the top of the form). This is the input for dc analysis.
- For ac analysis input, find the entry “Display small signal params” and turn the option on. The form will update to display new form fields. Add 1V in the “AC magnitude. This input value will be used for the AC analysis.
- For transient analysis, you need to choose the “Source type” field to the type you need. Enter all the applicable entries and click on OK.

The vsource component is now configured for simulations. Click on “Check and save” to save your design. Correct any errors or warnings before proceeding further.



Whenever you need to run simulations, always perform a “Check and save” first. During simulations, if you modify anything in the schematics, you must “check and save” for the changes to take effect.



Save often! Like any other software, Cadence tools are prone to crashing. Saving your work often will reduce the frustration caused by a crash. Using “Save” saves the design as-is without performing a “check”.

E) Adding comments

It is good practice to add comments to the schematics. Click on **Add**→**Note**→**Note text...** and fill out the form. The comments are useful reminders to yourself about the particular schematic and can also provide instructions for others if you share your schematics.

F) Printing

To print a schematic, click on **Design→Plot→Submit...** to bring up the printing form. If you would rather not print a header page (a page containing your username and the cell being printed), uncheck the “header” option on the form. Click on “Advanced options” button at the bottom of the form and uncheck the option of sending you an e-mail for every printout.

4. Simulating a circuit

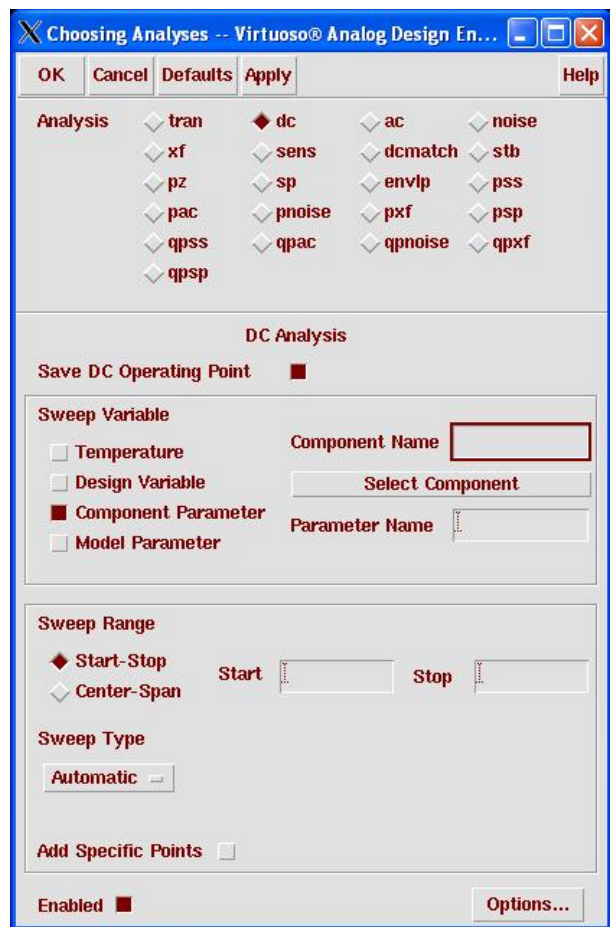
Once the schematic capture is complete, we can simulate the circuit to test if it performs according to the design. To start the simulation interface, click on **Tools→Analog Environment** in the schematics window to launch the Analog Design environment (ADE). The ADE window is used to select a simulator as well as set up all the options and simulations for the simulator. The default simulator is in ADE is a simulator called Spectre from Cadence. It has the same capabilities as Spice with enhancements introduced by Cadence. You can choose a different simulator by clicking on **Setup→Simulator/Directory/Host...** menu item. We will use Spectre for this lab and for most of our analog simulations.

A) Setting up the simulations

The ADE window is shown below. Familiarize yourself with the icons on the right side of the window. These icons represent the frequently used tasks in ADE. Click on the **Choose analyses...** icon to bring up the “Choosing Analysis” form. We will first run a dc analysis. Click on the button next to dc in the form. The form contents will update to show entries relevant to a dc analysis. We would like to save the dc operating point and sweep the input voltage from 0 to 5 volts. To sweep the input dc voltage, we first need to select the appropriate parameter. In the “sweep variable” box area, click on the “component parameter” button. The updated form is shown on the right.

To select the dc input voltage of the vsource, click on “Select component” and then click on vsource in the schematics window. In the new form that pops up, click on the first entry, dc, to choose the dc voltage of the vsource. You would follow a similar procedure to sweep other parameter of a source, if desired.

Next, enter the range of sweep in the start and stop boxes. Click on **Apply** for all these changes to take effect. This completes the dc analysis setup. Click OK to close the form and return to ADE.





You did not specify a step size for the dc analysis. The step size is automatically picked since the “Sweep type” is set to automatic. If you find the step size to be unacceptably large, you can change the “Sweep type” to linear and specify a step size.

B) Setting up the outputs

We need to tell the simulator which node voltages we are interested in plotting once the simulation is complete. To choose which signals to plot, click on the menu item **Outputs→To be Plotted→Select on Schematics**. The bottom of the ADE tells you to select the outputs you want plotted. Clicking on a net or a label selects voltage of the net to be plotted. Find the schematics window and click on the three labels we added earlier. Once done, press the escape key.



By default, only the voltages in the circuit are saved. To be able to save and plot waveform of a current, click on **Outputs→Save all...** in ADE and the turn on the option “selected” for “Save device currents (currents)”. To plot the current flowing into a device terminal click on the red terminal. If you see a circle around that terminal, the current will be plotted

C) Running the simulation and viewing the results

Now that we have set up which type of analysis we want to run and which outputs we are interested in, let us run the simulation. Click on the green traffic light icon to run the simulation. Once the simulation is complete, a waveform viewer window will come up with the plots of the outputs you selected in the previous step. Once again, familiarize yourself with the icons in the waveform viewer for common operations you need to perform on the waveforms.

Appendix A: Basic UNIX commands

This appendix introduces some basic commonly used **UNIX** commands. The following notation is used: words in the *fixed width font* are commands you need to type exactly, words in the *italic font* are to be substituted with appropriate words for your intended purpose, and words within square brackets [] are optional. Unix command as well as file names are *case-sensitive*.

`emacs filename`

Opens the file *filename* if it exists. If the file *filename* does not exist, a new file by that name is created. Instead of `emacs`, you can use any text editor of your choice. Some popular editors include `vi`, `pico`, etc.

`ls -l [directory]`

Lists the contents of the directory. If no directory name is supplied, contents of the current directory are listed.

`cp source_file target_file`

Copies the *source_file* to a new file named *target_file*.

`mv old_filename new_filename`

Moves/renames the *old_filename* to *new_filename*. If the target *new_filename* includes a path, the file is renamed and moved to that path. It can also be used to rename directory.

`rm trash_file`

Removes/deletes the unwanted *trash_file*. Note that file removed cannot be "undeleted", so make sure you know what you are going to delete.

`mkdir new_directory`

Makes a new subdirectory named *new_directory* within the current directory.

`cd [directory]`

Change current directory to the given directory. If no directory name is provided, the current directory will change to user's home directory.

`pwd`

Shows the present working directory

`rmdir trash_directory`

Remove/delete the unwanted *trash_directory*.

`man command`

Show the manual of the command. You can learn more detail about any given command using `man`, e.g. `man ls` shows all available options and functionalities for `ls` command.

`more filename`

Shows the contents of a file one screen at a time.

`ps`

Shows the currently running processes.

A couple of useful special symbols

~

The symbol `~` is used to denote the user's home directory. If the complete path of the home directory happens to be `/home/username/`, then using `~` represents that complete string. For example, if you want to look at the contents of a file in your home directory, the following commands produce the same result

```
more /home/username/myfile
```

```
more ~/myfile
```

&

When launching a program that opens in a new window, placing an `"&"` at the end of the command line allows the user to keep access to the command line, i.e., the `"&"` instructs the program to run in the background.

These are just a few shell commands you need to know to get started. You are encouraged and advised to learn other UNIX commands on your own, as they will increase your productivity, i.e., more you know, more efficient you will be.

Turn-in List:

1. Answers to the pre-lab.
2. Simulation results and their comparison with the pre-lab calculations

◆ This document is available at <http://class.ece.iastate.edu/ee434/labs/lab1.pdf>