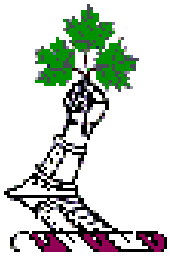

Royal Military College of Canada
Collège Militaire Royal du Canada
(Cadence University Alliance Program Member)

Department of Electrical and Computer Engineering
Département de Génie Electrique et Informatique

RMC Microelectronics Lab **Cadence Series**

Getting Started with Cadence

Tutorial One (Composer, Affirma, Virtuoso)

[Version 5.1 for Cadence.2005a - Dated 24 Aug 2005]

Authors (including revisions)

1. *Gord Allan, Version 3.1*
2. *Jean-Luc Derome, Version 4.3*
3. *Jean-Luc Derome, Version 5.1*

Approval Authority

1. *Dr. Dhamin Al-Khalili, Professor*

Table of Contents

List of Figures

1.0 Introduction

1.1 About this Manual

The *Getting Started with Cadence* manual provides the necessary information for the RMC users to get acquainted with the basic tools of Cadence. This guide is organized in six Sections: an Introduction, the Cadence Environment, Schematic Creation, Analogue Circuit Simulation, Analogue Characterization and a discussion on the Waveform Calculator. This guide is only meant as a starter document to get familiar with the basic Cadence tools.

1.2 The Big Picture

There is an overwhelming supply of CAD tools on the market for IC design and verification. Where then, does the Cadence suite fit in? As you are likely aware, semi-custom and custom IC design can be likened to puzzle building. For the most part, a designer will use available logical blocks (standard cells) and wire them together onto a single piece of silicon to solve a particular problem. Hence the term, application specific integrated circuits or ASICs.

Often a designer can not be content with only the standard blocks provided in a library. Even if the logic is available it may need re-design to meet the specific timing, power, or area constraints of a project. In academic institutions, these cells are made available through the Canadian Microelectronics Corporation (CMC), but in industry it may be cheaper to develop cells in-house, without licensing a technology library. Additionally, specialized components (Mixed-signal, optical, etc.) would not be included in any such standard library and will require their own development.

This is the area where the Cadence suite of tools is most appropriate. While tools exist to reasonably assemble thousands of cells together, the strength of Cadence is in creating, testing, and characterizing a custom function or standard cell.

To tackle such a project, the following design flow can be used:

a. Generate a schematic:
    i. using transistors and other ‘basic’ elements of a technology,
    ii. connect elements together and tune their parameters (W/L etc.) to suit the design, and
    iii. create a ‘Symbol’ of the cell (black-box abstraction);
b. Simulate and troubleshoot:
    i. using simulation and waveform viewers determine if the circuit meets design goals, and
    ii. troubleshoot parameters to get the circuit to perform properly;
c. Layout:
    i. given a circuit that does what you want in simulation you map that onto silicon, and
    ii. must follow design rules for minimum spacing etc.;
d. Extraction - LVS (Layout vs. Schematic):

   i. from the layout, determine what the schematic is, and
   ii. ideally what you started with, but you have to consider physical issues (e.g. parasitic capacitances); and

e. Post-Layout simulation and characterization
   i. simulate the resulting schematic after layout, and
   ii. has it met the design goals (power, speed, area, etc.)?

2.0 RMC Environment Setup and Start up

Before using the Cadence tools, you need to establish an X-Terminal session with the ECE CAD Compute Server and setup the Unix environment to run the Cadence application at RMC. This section will help the users establish the session and execute the required commands before attempting to run the tools.

2.1 Connecting to the ECE CAD Compute Server

The ECE CAD Compute server is accessed through an X terminal software that allows PC users to connect their Microsoft Windows systems to powerful X Window System applications residing on Unix-based servers. Xwin32 is the X Terminal software used at RMC. ECE CAD Compute server is called *sol4.rmc.ca*. It can be accessed from several locations:

a. SB3023 laboratory; and
b. ECE laboratories and Offices.

2.1.1 SB3023

There are 30 PC workstations that have access to the ECE CAD Compute server. An icon on your desktop (*Sol4*) should be present that allows you to connect to this server. If the icon is not present, the *Getting Started with XWin32* tutorial from the ECE Department can be used to configure the software. The tutorial can be found at "*http://intranet.rmc.ca/academic/elec/vlsitools/sw/documents/Xwin32.pdf*".

2.1.2 ECE Laboratories and Offices

All the laboratories in the ECE are configured to access the ECE CAD Compute Server. Follow the instructions in the *Getting Started with XWin32* tutorial from the ECE Department located at "*http://intranet.rmc.ca/academic/elec/vlsitools/sw/documents/Xwin32.pdf*".

In addition, most of the offices in the ECE Department are also configured to access the server. Should you require access, please send your request via E-mail to Jean-Luc Derome at "*elecadm@rmc.ca*".

2.2 Environment Setup

2.2.1 Environment

In order to run the Cadence tools, the system must be configured to run Cadence. The Cadence System Administrator can be contacted to have your machine setup in this environment if necessary (E-mail request to elecadm@rmc.ca).

You will be required to modify your shell environment by running a script design to point to the appropriate directories in order to run the Cadence tools (RMC maintained script). At RMC, the command "gocadence" has been written to perform this task. The user must type the command below at the Unix prompt and you should see the output message shown in *Figure 1*.

```
Setting Up for Cadence.2005a . . .
Done.
```

Figure 1: "gocadence" Message Output

a. **hostname$** . gocadence
   Note that there is a space between the "*dot*" and the "*gocadence*" command.

Note: Please note that **hostname$** is the Unix prompt and you ***do not need*** to type it at your command line prompt. The command line prompt is usually the hostname of the SUN workstation you are working on (ie: sol4 or labsun44).

2.3 Starting up Cadence

2.3.1 The First Run

The first time you start the Cadence tools, it is recommended that you create a directory related to the technology you will be using. This tutorial is based on the CMOSP18 technology from TSMC. Therefore, we recommend that the directory be called cadencep18. All files created during the utilization of the Cadence tools will be saved into this directory. Further breakdown of this directory is sometimes useful but it is left to the user for experimentation. The following commands is a possible list of commands to execute the **first time** you start up the tools:

a. *Copy Cadence Enviroment File*: This file indicates to Cadence the default commands to setup tools most often used. The main user interface can be configured in many ways. In particular, the cdsenv file recommended for this tutorial contains the simulator and waveform viewer defaults (spectre and AWD). The file is available from the link "*http://intranet.rmc.ca/academic/elec/vlsitools/sw/documents/cdsenv*". Save it to a known location on your UNIX account. Execute the following Unix command to copy the Cadence environment file:
   i. **hostname**$ cp cdsenv $HOME/.cdsenv

b. *Create a clean directory*: Please note that although it is possible to have more than one technology in the same directory, you can only access one at the time. Therefore, it may be wise to use the different technologies in separate directories. For example, The following Unix command will create a directory for CMOSP18:
   i. **hostname**$ mkdir cadencep18

c. *Start Cadence in cadencep18*: After initializing your UNIX environment, start Cadence by typing the command below and the Integrated Circuit Front and Back (icfb) window will be displayed to you (see *Figure 2*). Starting up the icfb window takes a few moments and several things will be displayed to you during this process. Be patient and wait for the whole process to complete.
   i. **hostname**$ cd cadencep18
   ii. **hostname**$ startCds -t cmosp18
   Note: You will be shown a window titled "*What's New in 5.10.41*". You can take the

time to read the content but it may be ignored at this time. Just close the window and continue with the instructions in the tutorial;

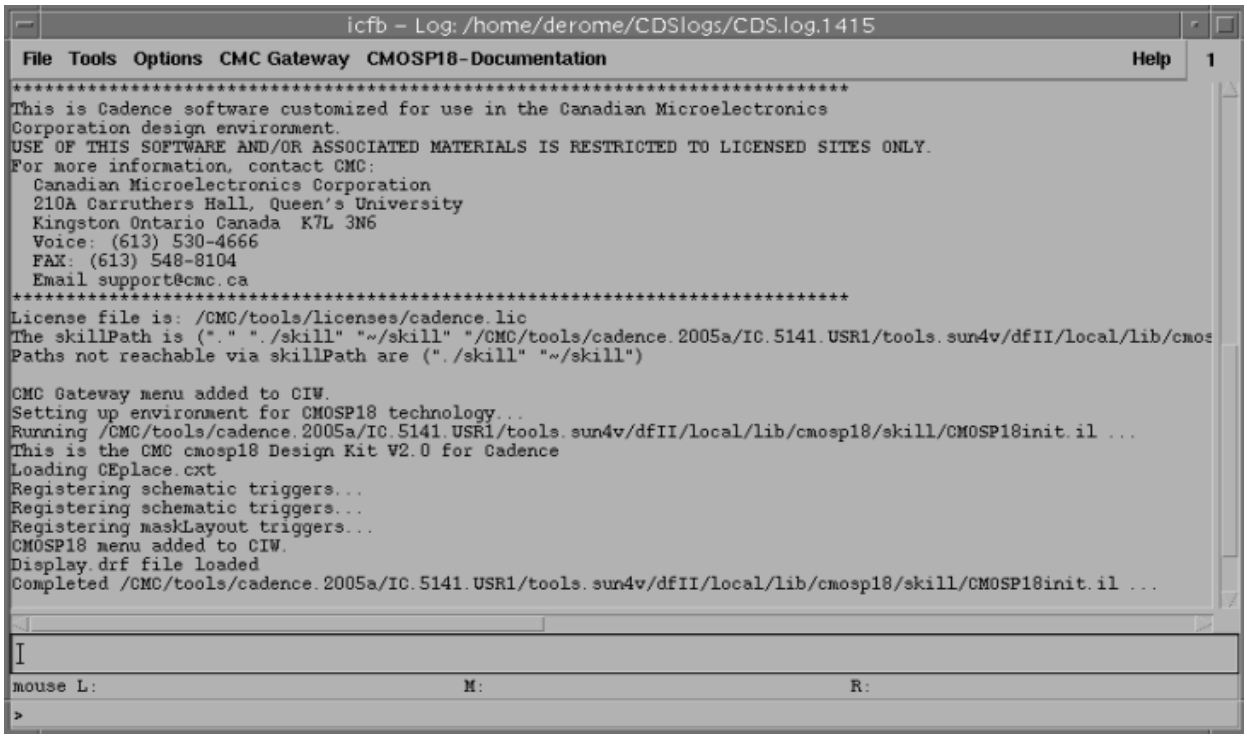

Figure 2: icfb window

d. *Library List*: You may notice a file called 'cds.lib' in your cadencep18 directory. This file is added to the "*current directory"* the first time Cadence starts. It contains a list of the libraries Cadence recognizes, and points to where they are located in the directory structure. When you add your own library soon, this list will be updated;

e. *Model Files*: Cadence uses the industry standard SPICE language for simulation and device specification. There are different levels of accuracy available depending on how well the devices are described. For your purposes, you will use the default models which reflect standard 0.18 μm characteristics. In the future it may be of use to develop one's own models and use them for simulation; and

f. *Note*: If at any time cadence hangs, the best way to recover is to remotely login to the workstation, list the running processes using the 'ps -a' command from a unix prompt, and then use the 'kill' command on the offending process ID. (i.e.: 'kill 1243' if 1243 is the PID)

2.4 Cadence Tools

To create, simulate, and layout a cell you use a few different tools. As indicated at *Section 2.3*, when cadence opens, it brings up the icfb window (*Figure 2*). This is the menu into the cadence tools and also allows us to perform other specific functions, such as library management. Browse through the menu and do not be intimidated by all the options and acronyms. Experiment with the tools to become more familiar with all the different options. Cadence is very robust, it will not let

you do any damage to the OS or the workstation. The links to the CMOSP18 documentation and CMC design-flow are useful. The following lists some of the tools provided by the Cadence suite:

a. Virtuoso Schematic Editor (aka. Composer) - to create schematics and symbols;
b. Affirma Analog Design Environment - for circuit simulation; and
c. Virtuoso Layout Editor - for layout.

For the time being, a good place to begin to understand the cadence environment is with the Library Manager, which can be opened selecting '**Tools | Library Manager**' (shown in *Figure 3*).

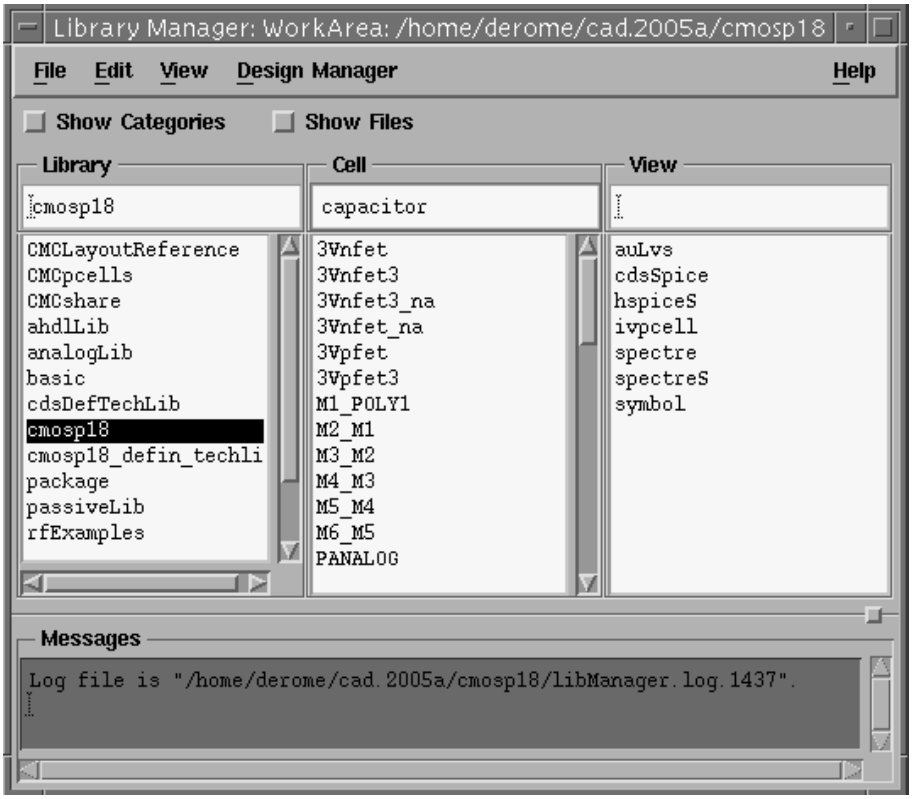

Figure 3: Library Manager

2.5 Libraries, Cells and Views

Throughout Cadence there is a hierarchy of Library | Cell | View. In *Figure 3*, the 'cmosp18' library is selected, which contains the logic gates (cells) for the 0.18µm process. The 'capacitor' cell is selected. As you go through this tutorial, you will create different 'views' or representations of a standard cell. In this example, you see that for the capacitor cell, the library has already provided some of these different views.

To appreciate what you will be creating, open and glance at the symbol, layout and extracted views of this (or any other) cell, look around the different libraries and open various views to get a feel for the different types. You will be advised that the views are read-only, which is fine since

you really do not want to re-design the OR gate. In the tutorial, you are going to create a RLC filter and a CMOS inverter.

In the next step of this tutorial, you will create your own library which will contain your user defined cells and their corresponding views. Some of the other libraries you may find:

a. **basic**: Contains various input sources and other basic components;
b. **cdsDefTechLib**: A default technology library that your design gets attached to if you didn't specify one when starting Cadence;
c. **cmosp18:** components for 0.18μm- FETs, BJTs, cap, resistors, etc.;
d. **package:** outline of packages (ceramic cases) to fit chips into;
e. **padsp35:** IO pads and drivers developed by CMC;
f. **tpz973g**: IO pads and drivers for CMOSP18; and
g. **vst_n18_sc_tsm_c4:** standard logic cells by TSMC; and

3.0 Generating the Schematics (RLC filter and Inverter)

3.1 Create a new library

To illustrate the process of both analog and pseudo-digital design, you will create and simulate two circuits, a RLC filter and a CMOS inverter. Before you begin though, you will create a new library to store these two cells and their views. The instructions to create a library are as follows:

a. From the icfb menu, select '**File | New | Library**';
b. Navigate to the cadencep18 directory (you should already be there);
c. Type in a name for your new library: ***rmc_tutorial***;
d. Select '***attach to an existing techfile***', click ***Ok***; and
e. ***Select the cmosp18_defin_techlib library*** when prompted and click ***Ok***. The techfile contains the design rules that apply when generating an actual layout. It is not necessary here, but is done to demonstrate the procedure.

3.2 Create a new cellview (Schematic)

Now it is time to create one of the schematics. From the icfb menu, select '**File | New | Cellview.**' You are prompted for the library to place the new cell in, and what type of view you are creating. Select ***'rmc_tutorial'*** and ensure '***Composer-Schematic***' is selected as the tool for your new cell. Note that 'schematic' is/becomes the default name of the view. Enter ***'rlc'*** as the cellname and click '***OK***' to open the new cell in composer.

The rest of this section describes the required steps to create the '*rlc*' circuit shown at *Figure 4*. You will then move on to create the inverter schematic.

3.3 Adding Components and wiring

Whether creating an analog, digital or mixed-signal schematic the fundamentals are the same. You place 'instances' of devices which are already defined in one of the referenced libraries, wire them together, and edit their properties (e.g. Resistance, capacitance, width, etc.) In the case of the RLC circuit, you will place a resistor, capacitor, inductor and ground. You wire the devices together using the wire tool, and then edit the properties of the devices to set R=75k, R=75, C=47n, and L=500m. Of course you then need to add pins for the filter's input and output signals.

You should note at this point that there are other methods of design entry other than actually drawing it via 'schematic capture.' Typically, HDL (hardware description languages) are used for large designs. Using an HDL language, you would obtain after compilation the specific components and connections between all the devices. The low level list of components and connections is commonly referred to in industry as the *net-list*.

The details of putting the schematic together follow, but **you are encouraged to try to figure it out yourself** through exploration. (One hint: The instantiate icon.)

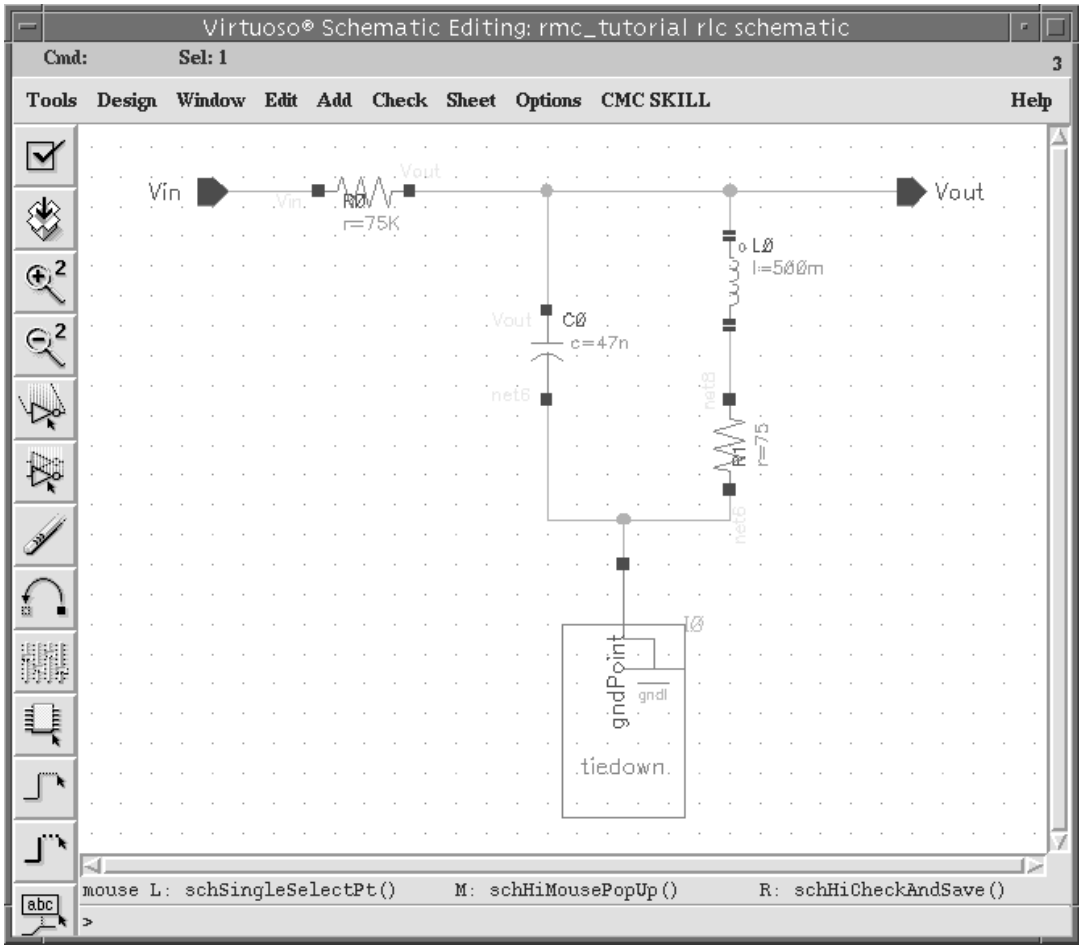

Figure 4: RLC Circuit

General Tip: You will find that Cadence likes to '**do what it did last**'. For example, if the last action you selected from the menu was 'delete', the left mouse button will now be assigned the '**delete**' command. This can lead to some unexpected (and scary) actions. Hit the '**ESC**' key to get Cadence out of this mode after selecting a command. Undo can come in handy here.

3.3.1 Instantiating a component

To place a resistor follow these steps from the main composer screen:

a. Click on the **Instance** Icon. The Add Instance window will be displayed with all fields empty. Click on the **Browse** button;

b. In the Library Browser window, select the **cmosp18** library, the **resistor** cell, and **symbol** view The Add Instance window should now look exactly like the one shown in *Figure 5* (the resistance field maybe different on your window);

c. Move the cursor to the Composer schematic window, the resistor symbol follows. Also, note that the Add Instance window has expanded to display other parameters. Before you click on the schematic window to place the resistor symbol, **edit the form, modifying the Resistance value to 75k** Ohms, as shown *Figure 5*. **IMPORTANT:** Do

not get concerned with all of the seemingly irrelevant parameters. They are used for more detailed simulations and other applications;

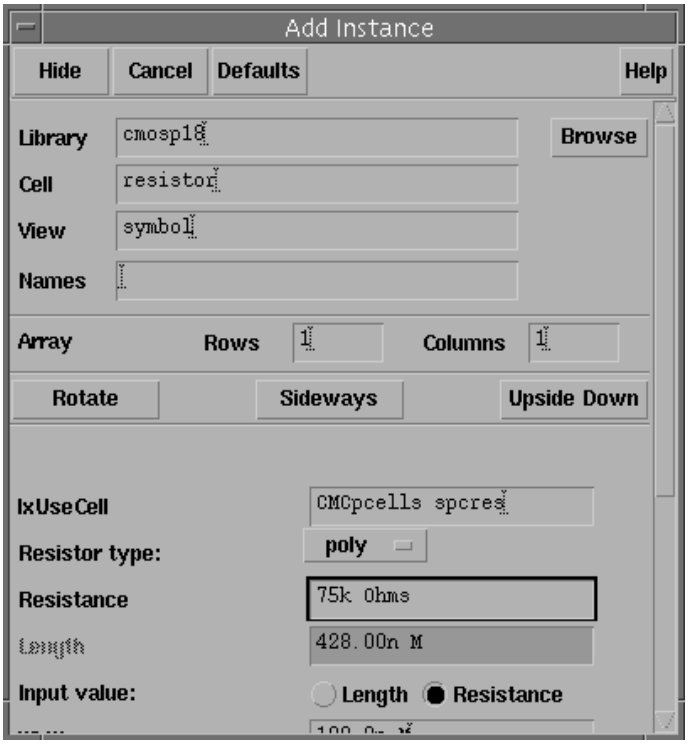

Figure 5: Add Instance Window - Resistor Cell

d. Now click in the composer window to place the resistor;

e. Another resistor symbol follows the cursor. Place it in the window then click on **Cancel** on the Add Instance window. The form disappears;

f. In the same way you added the resistor, add the other instances from the library, cell, view as indicated below. You Will change their values shortly to correspond to the schematic.

   i. C (cmosp18, capacitor, symbol),
   ii. L (analoglib, ind, auLvs), and
   iii. Ground/tiedown (cmosp18, tiedown, symbol).

   When you add the capacitor there are fields where you can fill in the model name and area. When manufactured, these determine the actual capacitance. For your purposes, you simply ‘hard-code’ the required capacitance as you will not intend to fabricate the circuit; and

g. To rotate the input resistor, click once on it to select (left-click with the mouse), then middle-click to open the auxiliary menu. Select **Rotate**.

3.3.2 Adding the I/O Pins

a. To add the input and output pins, click on the **Pin** icon in the lower left side of the Composer window. The Add Pin form appears as shown in *Figure 6*;

b. Under **Pin Names**, type **Vin Vout**. Note that **Direction** in the form reads **input**, also shown in *Figure 6*;

c. Click once on the schematic window. The first pin is placed. Note the other pin's symbol follows the cursor as you move across the window. The Add Pin form is still active, but with only **Vout** displaying in the **Pin Name** field; and

d. On the Add Pin form, change **Direction** to read **Output**. Place the **Vout** pin in the schematic window. You can close the Add Pin form when done.

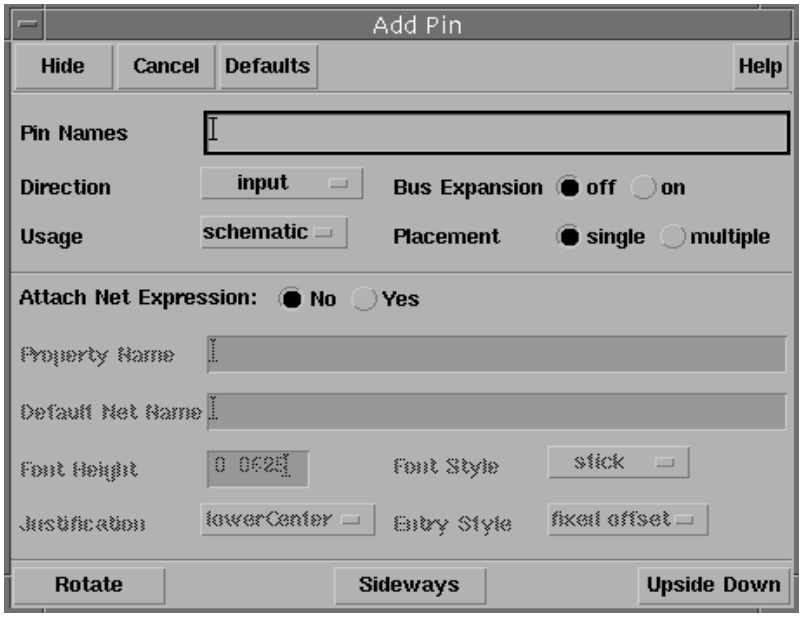

Figure 6: Add Pin

3.3.3 Connecting Wires

a. Before wiring, you may wish to move some of the components around on the screen to look more like the schematic shown in *Figure 4*. Use the 'ESC' key to ensure you are not stuck in a weird mode, and then you can drag the components around on the screen. To begin connecting the wires as per the schematic in *Figure 4*, click on the **Wire(narrow)** icon;

b. Start connecting your circuit by right-clicking at the components' terminal (diamond shape at the end of the components). Note that, when you move close to the component terminal, the diamond shape is displayed to you if you are in the Add Wire mode;

c. While you are in the Add Wire mode (the Add Wire window is displayed but not selected), click on the **s** key on your keyboard. This *snaps* the cursor to the closest diamond-shape terminal. You do not need to click on the terminal to connect the wire when you use the snap technique. The connection is made automatically and you simply move on to the next terminal to connect; and

d. Finish connecting the components together until your circuit resemble the one shown in *Figure 4*.

3.4 Modifying Instance Properties

A potential source of confusion are the properties associated with each device. Do not let the myriad of device options confuse you! In most cases you will use the default values.

To modify the device's properties, **click once to select the instance**, then click on the **properties icon**, or otherwise select it from the menu system (e.g. using the '***q***' bindkey).

Modify the Inductance (500m H), Capacitance (47n F) and 2nd resistance (75 ohm) values to reflect the schematic. By default, Cadence automatically writes the Instance Name.

Note: When you are entering values for various parameters, you do not need enter the units as they are pre-defined (ie: ohm, farad, etc.). You do, however, need to indicate the size via the standard SI prefixes as follows. For example, a capacitance of 20 femptoFarads would be indicated by entering the value *20f* in the Capacitance field of the components' properties (No space between the number and prefix). Here are Scale Factors (case-sensitive):

a. M= Mega (10^6);
b. k= kilo (10^3);
c. m= milli (10^-3);
d. u= micro (10^-6);
e. n= nano (10^-9);
f. p= pico (10^-12);
g. f= femto (10^-15); and
h. a= atto (10^-18).

Note (1): **No space** between the number and the scale factor (i.e. **1p**, for 1 pico Farad)

Note (2): **Do not include the units**, as they are predefined in the instance properties file.

3.4.1 Variable Parameters

In most applications you will want to simulate the behavior of a circuit with different loads/ capacitances/transistor widths etc. This is easily accomplished through the use of variables. Instead of assigning a constant (e.g. 75k Ω) to the component, you can use a variable name instead. When it comes time for simulation you simply assign a value to that variable, or instruct the simulator to run through with different values (called 'parametric analysis').

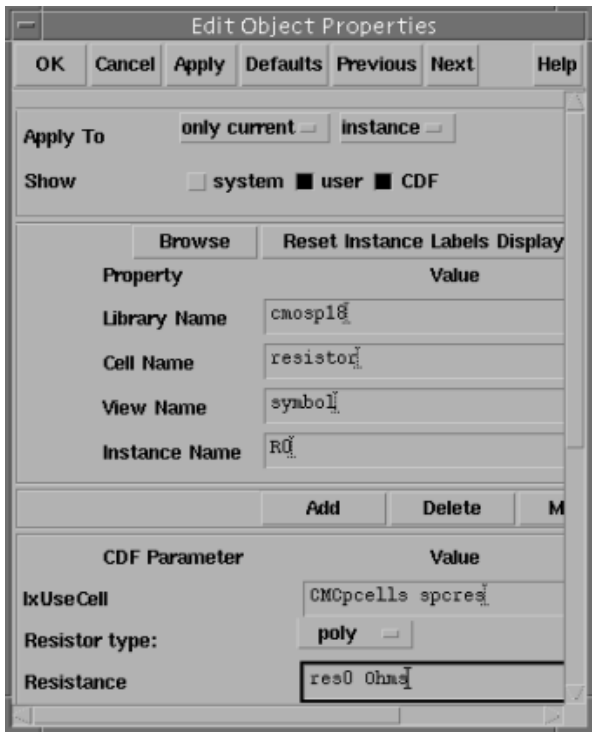

Figure 7: Edit Object Properties Window

As an example, for this rlc circuit the field Resistance (current value **'75k'**) for the series input resistor the value **'res0'** as shown in *Figure 7*. (To do this click on the resistance and use the properties icon). In the simulation you can assign 75k to 'res0' and everything else will look exactly the same. The advantage of using this technique is in the re-usability of the same schematic for different component values.

3.5 Checking and Saving

Now that you have the schematic drawn, you want to certify its correctness and save. Click on the **Check and Save** icon. The Cadence's icfb window will show that your schematic has been checked and saved as shown in *Figure 8*.

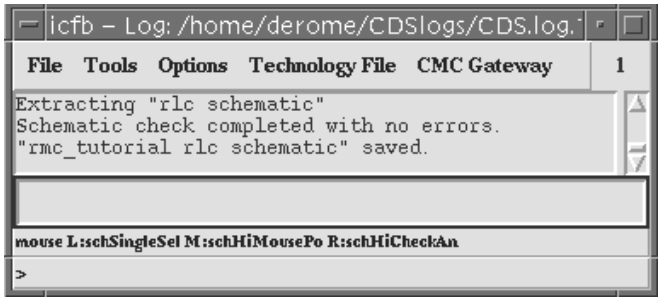

Figure 8: Check and Saved CIW Message

If you get Warnings/Errors, follow the instructions below to try to correct the problem:

a. Go back to your schematic and look for the "flashing" tiny squares;
b. Browse back in the icfb window and read the warning/error messages;
c. Fix the warnings/errors as necessary. Warnings are not as crucial as Errors;
d. For example "Dangling Wires" warnings should be checked; and
e. Click on the **Check & Save** icon, and repeat these steps until the design has no errors.

3.6 Create the Inverter Schematic

Using the exact same methods that went into creating the RLC circuit you will create an inverter circuit. **Close the rlc circuit** and **create a new cell** for the inverter (**call it inv**). Using the schematic shown in *Figure 9* as a guide, you will need to **instantiate the following components**:

a. PMOS transistor (cmosp18, pfet, symbol);
b. NMOS transistor (cmosp18, nfet, symbol);
c. Ground connection (cmosp18, tiedown, symbol);
d. Capacitor for your output load (cmosp18, capacitor, symbol);
e. Input and Output pins;
f. Input Source (cmosp18, vpulse, symbol), you will edit the timing properties at simulation; and
g. Power Source (cmosp18, vdc, symbol), edit properties to output DC @ 1.8 V.

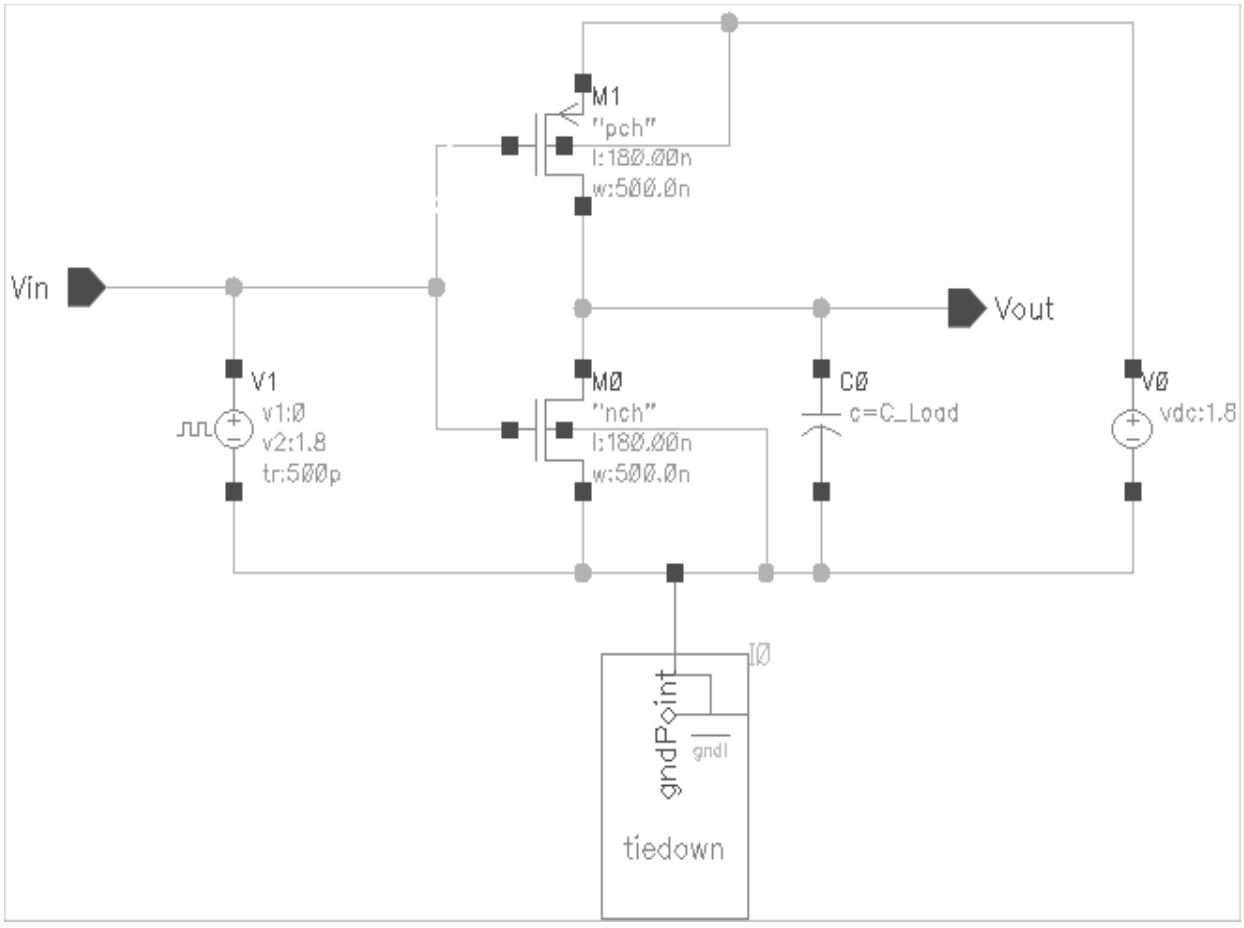

Figure 9: Inverter Schematic

When instantiating the transistors you can accept the default widths and lengths for the time-being. **Wire the devices** together as per the schematic shown in *Figure 9* using the wiring tool. You may want to make use of the 'flip' and 'rotate' commands from the edit menu. For the NMOS and PMOS transistors, at the properties pages, remove the area and periphery values and

change the model names to **nch** and **pch** respectively. For the output capacitance, set the *Capacitance* field to **C_Load**. In simulation, you will vary the load to see the effect on the output. **Check and save** the schematic before you move on.

Note that you are using devices from the cmosp18 library. Some of the same devices are duplicated in other libraries. For the most part it is irrelevant which one you choose.

An Aside about internal parameters: Often you may want to refer to the value of a parameter internal to a device. The most common example is to make the width of a transistor 2x the length. The iPar("[variable]") function is used to reference these internal values. The syntax under this example would be w=iPar("l")*2. You will set up the parameters later, prior to simulation.

4.0 Analog Circuit Simulation - RLC filter

Normally you would make symbol views of your 2 circuits before you perform a simulation. Then you would place those symbols into a new 'test-bench' schematic where the supply voltages and stimulus are added around the symbol. In this tutorial, you just want a simple non-hierarchical project, so you are not going to this level of abstraction.

In order to perform a simulation on your RLC circuit you need to provide some stimulus (an input voltage) and attach an output load. Open the RLC filter schematic (e.g. from the icfb, **File | Open**). To set up the circuit for testing, instantiate a *voltage source (cmosp18, vsin, symbol)* and a load *capacitor (cmosp18, capacitor, symbol)*. Wire them across the input and output pin as shown in *Figure 10* to get a fully testable circuit. Set up the voltage source's parameters such that *AC Magnitude=1.8* (the peak-peak used in AC sweep analysis), *AC Offset=900m*, *Amplitude=900m*, and *Frequency=6k*. This will give a 6 kHz sin wave from 0V -> 1.8V. Set the capacitive load for *C=1p*. Again, note that the units are implied and should not be user-specified. Do not worry if the net names are not the same (except for Vin and Vout of course). The net names are randomly assigned if not specified by the user. ***Check and Save*** once done.

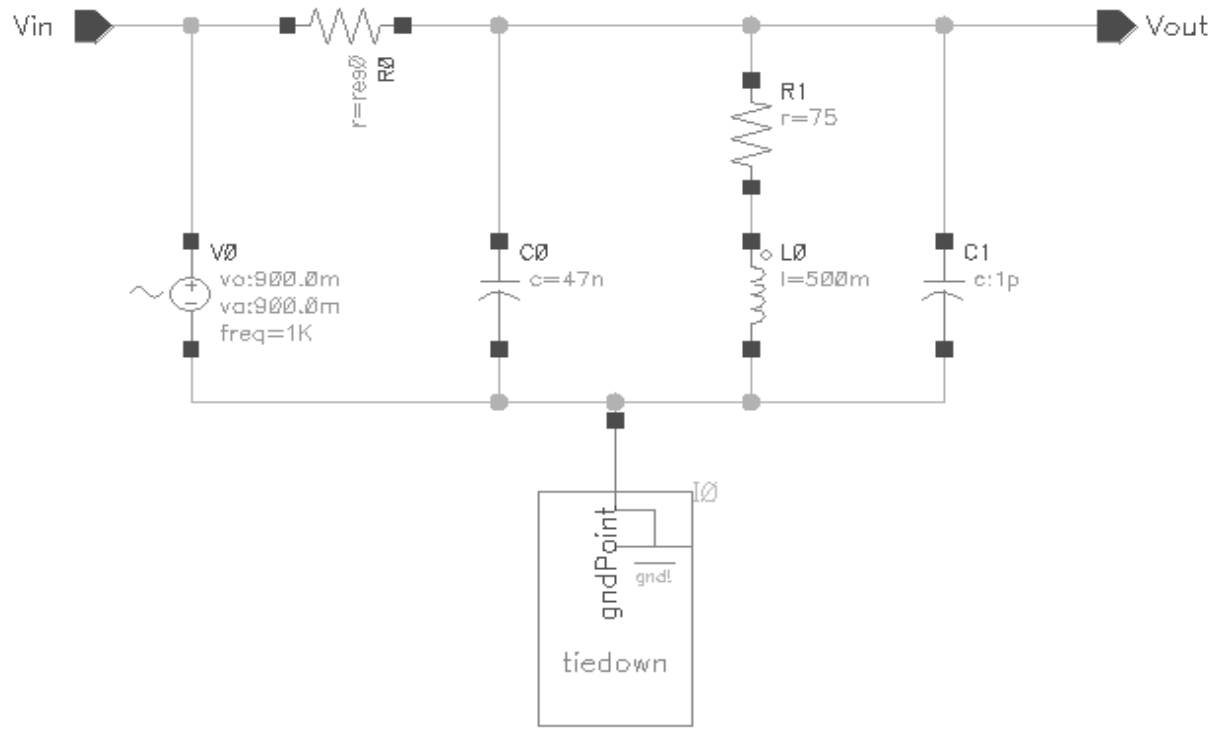

Figure 10: RLC Circuit for Simulation

Now that the schematic is ready for simulation, we start the simulation tools and setup the basic environment for analogue simulation. Select **Tools | Analog Environment**. *Figure 11* is displayed. First, select **Setup | Simulator/Directory/Host** and ensure that *spectre* is selected under Simulator and click ***OK***. From the Virtuoso Analog Environment, select **Session/Options**

and ensure that AWD waveform tool is selected and click ***OK***. Now, let's examine the Analogue Environment window in more details:

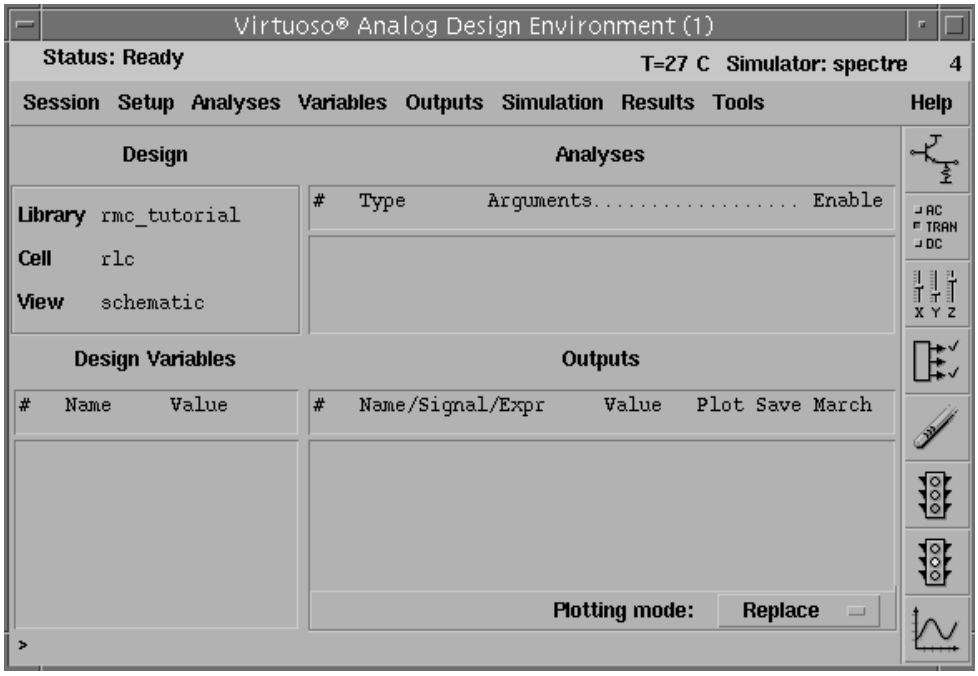

Figure 11: Analogue Circuit Design Environment

a. *Quick Command Area*: The icons on the right provide quick access to frequent commands/menus. Following the icons from the top to the bottom one guides you through the steps required to perform a complete simulation cycle;

b. *The Design Area*: Lists the Library, Cell, and CellView of the design being simulated;

c. *The Analyses Area*: Lists the types of analyses, any arguments (i.e. time interval), and whether the analysis is enabled in the current run;

d. *The Design Variables Area*: Will list components set-up as variables. Select ***Variables | Copy from Cellview*** and the *res0* variable will appear in this list. Edit the value of the resistance variable to equal 75k by using **Variables | Edit....** The Edit Design Variables window is displayed. Click on ***res0*** to select the desired variable. Enter ***75k*** in the Value (Expr) field and click ***Change***. Click ***Ok*** to exit the Editing Design Variables window and return to the Analog Simulation window; and

e. *The Outputs Area*: Lists names of nets/signals/expressions/ports to be plotted on the output waveform window.

4.1 Choosing the Analyses

In the Analog Design Envrionment window, click the **Choose Analysis** icon or select **Analysis | Choose** from the pull down menu. The form appears as shown in *Figure 12*.

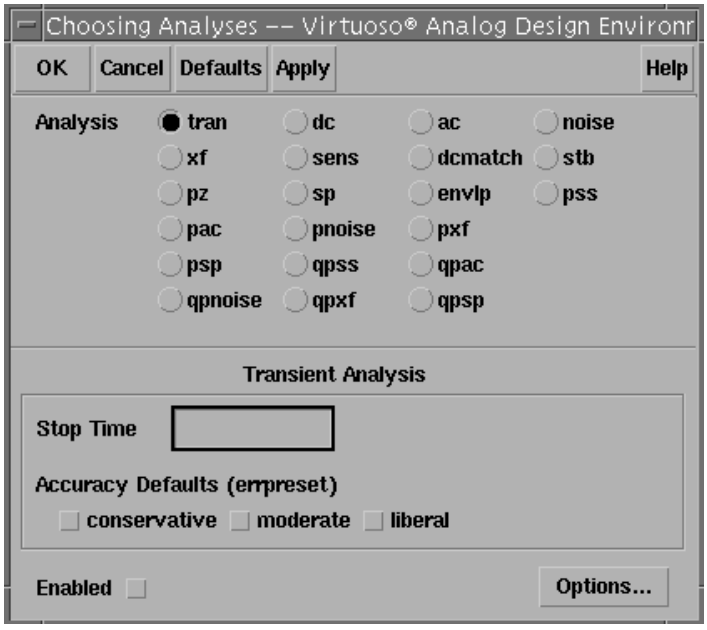

Figure 12: Choosing Analysis Window

You will simultaneously setup several analyses modes:

a. **Transient analysis:** This provides the transient output response of the circuit with respect to time. The user specifies the time period and the time variant input waveform while the simulator calculates the output response;

b. **AC analysis:** This simulates the AC performance of the circuit as a function of frequency, and is based upon the small-signal frequency response model;

c. **DC Operating Point:** This analysis simply determines the D.C. operating point of the circuit based on the parameters present on the schematic assuming all capacitors opened and all inductors shorted. It is the default mode and is automatically performed before any other analysis in order to determine the initial state of the circuit; and

d. **DC sweep mode:** This generates DC transfer characteristics for the circuit by varying a user specified independent source over a range of values.

4.1.1 Transient Analysis

You will perform a transient analysis. Follow the steps below to complete the analysis:

a. In the Analysis Section, select ***tran***;

b. Set the Stop Time field to ***2m***;

c. Select ***conservative*** and turn on the ***Enabled*** field; and

d. Click ***APPLY.*** **(do not click OK)**

Notice that in the Analog Artist Simulation Window, under the Analysis Section, a line is added to describe this analysis.

4.1.2 AC Analysis

You will perform an AC analysis. Follow the steps below to complete the analysis:

a. In the Analysis Section, select ***ac***;

b. Set the Sweep Variable to ***Frequency***;
c. Set the Sweep Range to ***Start-Stop***: Start: 0.01k, Stop: 10k;
d. Set the Sweep Type to ***Logarithmic,*** select ***Points per Decade*** and enter ***20***;
e. Turn on the ***Enabled*** field; and
f. Click on ***Apply***.

Again, notice that in the Analog Artist Simulation Window, under the Analysis Section, a line is also added to describe this analysis.

4.1.3 DC Sweep and DC Operating Point

You will now perform a DC sweep and a DC operating point analysis. Follow the steps below to complete the analysis:

a. In the Analysis Section, select ***dc***;
b. select ***Save DC Operating Point***;
c. In the Sweep Variable section, select ***Component Parameter***;
d. Click twice on ***Select Component***. This will allow you to select the instances from the schematic;
e. Click on the input source ***(Vsin component)*** from the Schematic window;
f. A form appears listing all the instances parameters. Select the ***dc*** parameter. Click ***OK***;
g. In the Sweep Range section, select ***Start-Stop***: ***Start: 0, Stop: 1.8***;
h. Turn on the ***Enabled*** field; and
i. Click ***OK*** this time.

This final analysis you need to setup. The analyses area should now list three different analysis.

4.2 Saving and Plotting Simulation Data

The simulation environment is configured to save all node voltages in the design by default. You can modify the default to save all terminal currents as well, or you can select specific set of nodes to save. You will select these nodes from the schematic window. From the Affirma Analog Circuit Design Environment window, perform the following tasks:

a. Select **Output => To be Plotted => Select on Schematic**.

Node voltages can be selected by clicking on the wire on the schematic window, and currents by clicking on the components' terminal. Unselecting can be performed either by clicking on the terminal/node again, or by selecting the corresponding line in the Outputs section of the Simulation window and clicking on the Delete icon.

a. **Select the input and output wires to the rlc circuit**. Observe the simulation window as the wires get added.

You now have the setup of the simulation completed for the RLC circuit. The completed Analog Analysis show now look like the one shown in *Figure 13*.

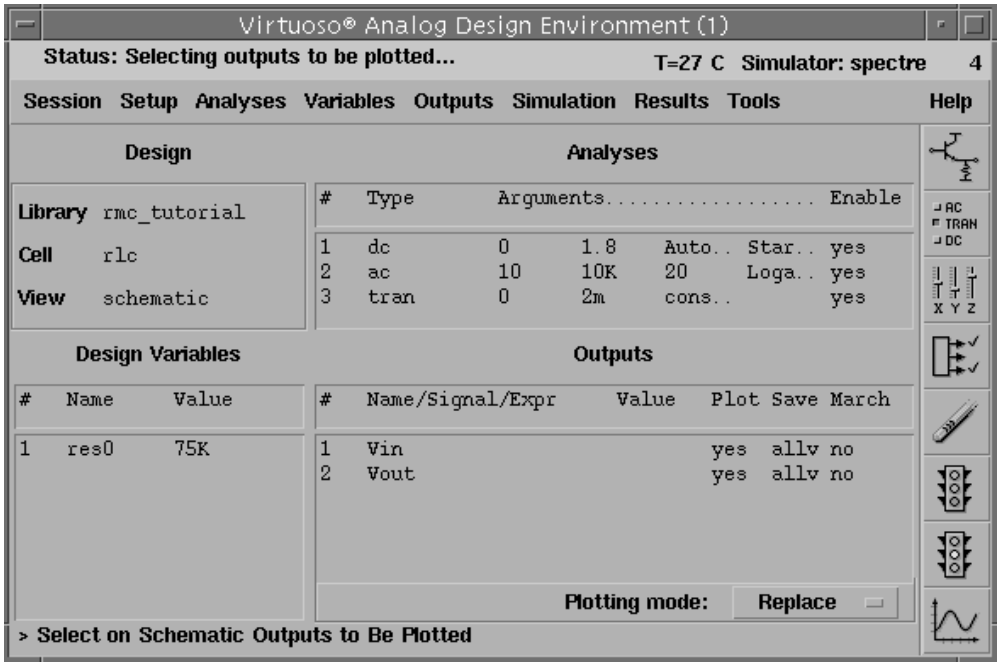

Figure 13: Analogue Circuit Design Environment - Setup Completed

4.3 Running the Simulation - The Waveform Window

You are now ready to run the simulation of the RLC circuit. Click on the ***Run Simulation*** icon (green light). Note that some information windows will be displayed to you. Simply click OK and the simulation will continue. When it completes, the resulting plots are shown automatically in the waveform window (shown in *Figure 14*). The three analysis responses are shown in separate plots. The DC response is not clear due to the voltage divider effect. The input and output waveforms are shown on the same scale. To separate them and otherwise change the appearance of the graphs, follow these steps:

a. ***Click*** anywhere in the DC Response area to work in this section. Notice the highlighted box reading "3" at the top-right corner of the section;

b. Select **Axes => To Strip**, or click on the ***Switch Axis Mode icon*** (shown in *Figure 15*) on the left. Observe the 2 separate plots displayed. This puts the input and output voltages on separate scales. You can repeat this step for all analysis;

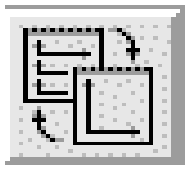

Figure 15: Axis Mode Icon

c. Use the Markers (A, B) (shown in *Figure 16*) to measure the output peak-to-peak amplitude of the Transient Response signal. Click on the ***Crosshair Marker A*** icon on the left. Click on the most ***negative peak*** of the output waveform. Repeat for Marker B, at the most positive peak;

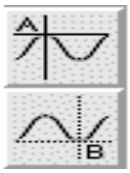

Figure 16: Marker A & B Icons

d. Read the markers data at the bottom of the screen. Look at the delta field (time, volts). You should observe a delta of approximately 25 mV; and

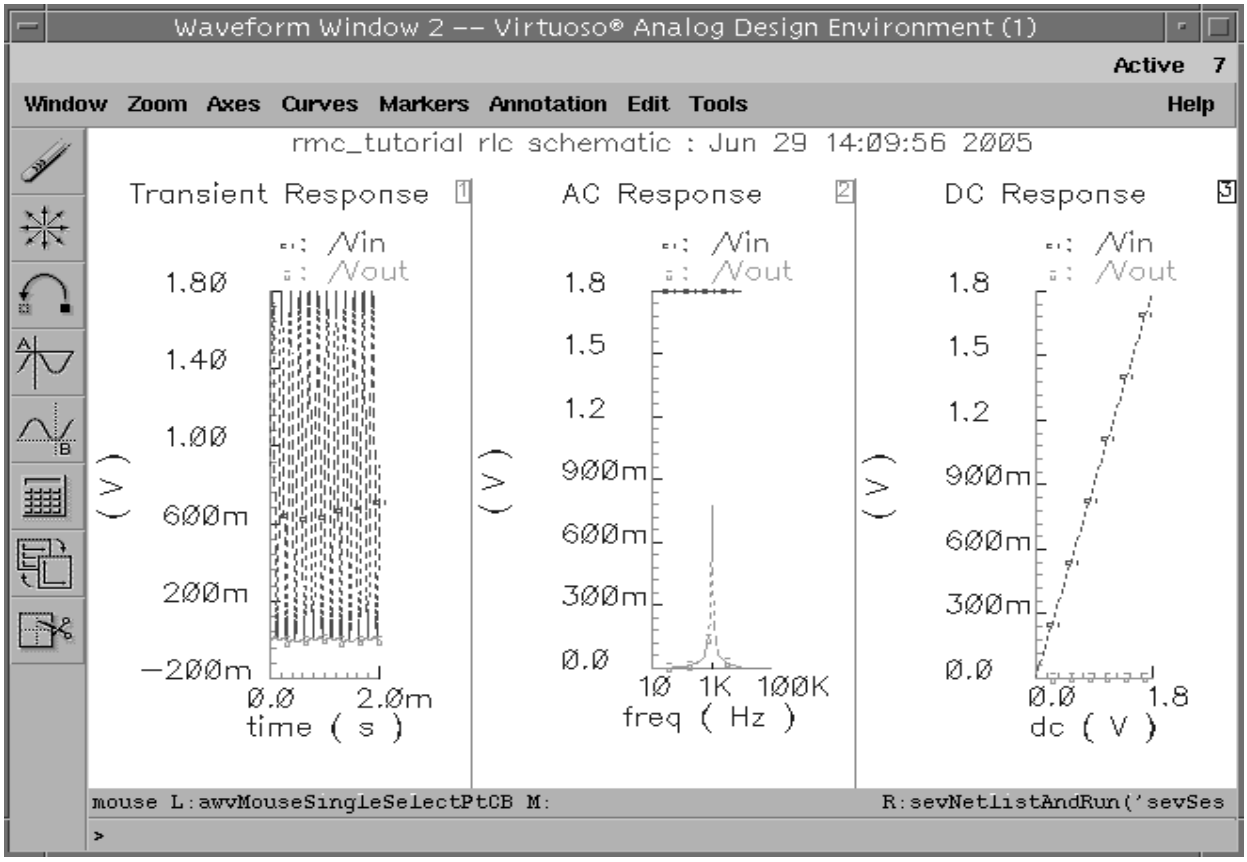

Figure 14: Waveform window

e. Select the AC Response section. This is the plot that really gives us the most information about the circuit. Note that at a frequency of 6 kHz, as you have simulated for the Vin/ Vout transient response (graph 1), the output is only 25 mV peak-peak as measured at steps *c* and *d*. Upon inspection of the output Vout vs Frequency, you can observe a peak at 1 kHz. This is a spiked notch filter centered at 1kHz. You are encouraged to re-adjust the frequency of the input source on your RLC schematic from 6k to 1k and re-run the simulation. Note that you will need to stop time for the transient analysis to see the full transient response of the circuit (circuit response before steady state is reached). A stop time of 8-10 ms should be sufficient;

The result of the transient analysis with a 1 KHz source is shown in *Figure 17*. Notice the larger gain and the initial damping response before settling into the steady-state. The Vout peak-peak voltage measured in *Figure 17* is ~ 850 mV (measure in the steady state area). In the frequency spectrum, the peak is observed at 1 Khz and corresponds to the operating frequency of our circuit as demonstrated in *Figure 17*.

4.4 Printing the Waveforms

Depending on the tool, you will access the print menu using two different interface:

a. Using the menu item ***Design | Plot | Submit***. This method is used from the schematic editor; or

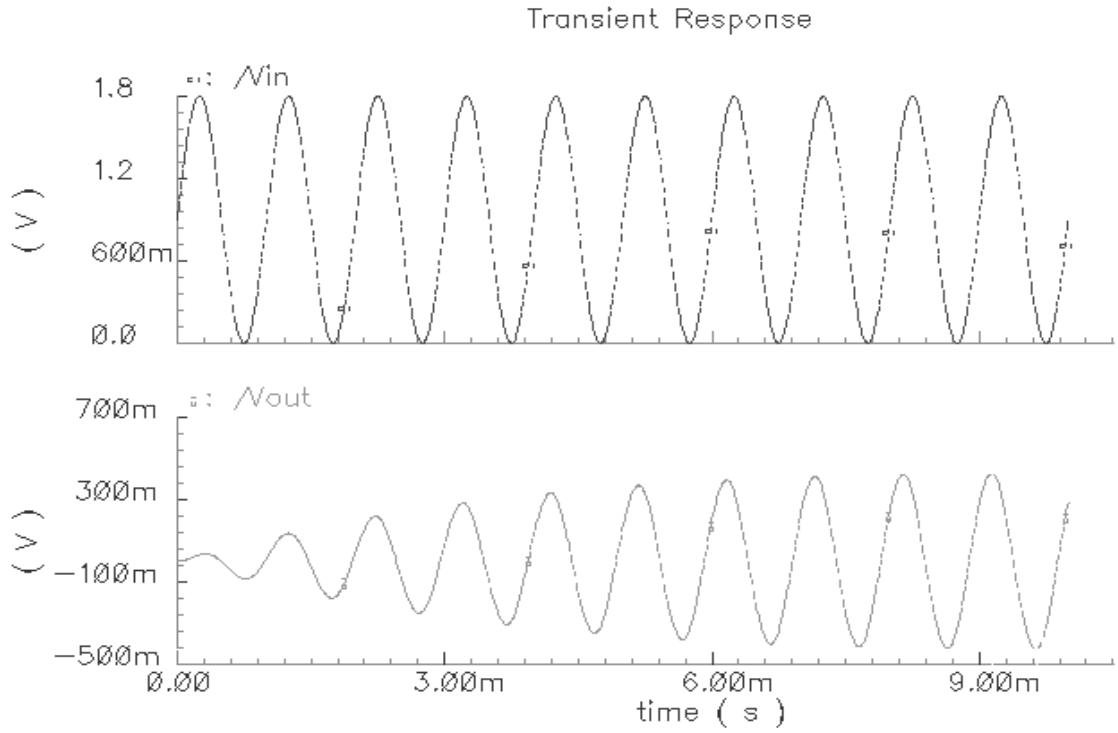

Figure 17: Transient Analysis Results

b. Using the menu item ***Window*** **|** ***Hardcopy***. This method is used from the waveform viewer.

You can submit your printing jobs to the printer directly or to a file. Most often, you will want print to a postscript file (use the file extension *.ps* for your filename - i.e.: *test.ps*). Using this method, the file can be used in different ways:

a. Within documents (for example: a word processor capable of handling images);
b. Manipulate the image using postscript file viewer; or
c. The file can be converted to a desired format, most often it is converted to the popular document reader format PDF.

If you submit your print job to the printer directly, RMC has the choice of two printers:

a. **sb3023**: located in Sawyer building SB3023. This room is the ECE/Mech Eng CAD Workstation laboratory. It contains 30 Workstations; or
b. **eevlsiptr**: located in Sawyer building SB3045. This room is the VLSI research laboratory and is restricted to researchers.

4.5 Saving the Simulation Session and Exiting

You have the options to save your simulation setup and/or the output waveforms. Note that the simulation results are saved deep in the directory structure for you, but you are not going to try to find them. To save the simulation configuration from the Analog Artist window, perform the following tasks:

a. Select **Session | Save State;** and
b. Choose a name to save the configuration under. This saves all your settings for Outputs, Analysis types, Variables, etc.

To save a waveform from the waveform window, perform the following tasks:

a. Select **Window | Save...** from the Waveform viewer; and
b. Choose a name to save the waveform you just created.

Please note that you need to remember the name of the file you used. It will be stored in the directory you started Cadence from (cadencep18). You may want to organized the waveform results to avoid the clutter of many files in the directory. For example, create a results directory and store the waveform in that directory. In order to load the saved waveform stored in the results directory, you will need to use the directory and the filename. To open a waveform file, perform the following tasks:

a. From the Analog Design Environment window (shown in *Figure 11*), select **Tools | Waveforms ....** The Waveform window is displayed;
b. From the Waveform window, select **Window | Load....** The Load window is displayed; and
c. If you used a directory to store waveforms, enter the directory/filename combination to load the waveform: ***wave_dir/wave_name***. If you did not use a directory, use the filename only: ***wave_name***.

5.0 Analog Characterization of an inverter

In order to perform the characterization of the inverter, there are a few steps you need to take care of. In particular, you will need to perform the following tasks:

a. Complete the schematic to adjust the component parameters necessary for the simulation: input source, load, transistor parameters, and model files;

b. Set up the simulation environment. This task is broken down in several pieces:
   i. Simulation and Model file setup,
   ii. Variable set up,
   iii. Output set up, and
   iv. Analysis set up;

5.1 Schematic Changes in Preparation of Simulation

The schematic build at *Section 3.6* needs to be adjusted/verify to ensure that the circuit is ready for simulation. In particular, you will need to setup the input source stimuli, ensure the transistor parameters are correct and change the capacitive load to a variable to simplify the analysis of the inverter. Perform the following changes to the inverter schematic created at *Section 3.6*:

a. You need to change the parameters for the input source pulse. There are many parameters for it and you will change the following parameters:
   i. In the schematic editor, select the input source and invoke the parameters as shown previously; and
   ii. Set the following parameters:
      1. **delay (td) = rise-time (tr)** = **fall-time (tf) = 500 picoseconds**,
      2. **pulse-width (pw) = 3n**,
      3. **period (per) = 8n**,
      4. Note that this gives us a clock frequency of 1/8ns = 125 Mhz**;**

b. Verify the transistor widths and lengths by selecting each one and checking its properties. Ensure the following parameters for both transistors are set to:
   i. **width=0.50**μm**=500nm**, and
   ii. **length=0.18**μm**=180nm**;

Ensure that the capacitive load (C's value) is a variable, called C_Load.

5.2 Simulating the Inverter

From the inverter schematic, follow the steps below to setup the simulation for the CMOS inverter:

a. Select **Tools->Analog Environment** to bring up Analog Artist with the inverter design pre-loaded;

b. From Analog Artist select **Setup | Model Libraries...**;
The **Model Library Setup** window is displayed.

c. In the Model Library File field, enter the filename "**/CMC/training/models/models18.scs**";
NOTE: Other spectre model files are available for different feature size: *models35.scs, models12,scs, models5.scs and models_simple.scs.*

d. In the **Section (opt.)** field, enter "**tt**";

e. Click **Add** in the **Model Library Setup** window;

f. Click **OK** in the **Model Library Setup** window;

g. In Analog Artist, choose a *transient analysis*. With a period of 8 ns, you need to set the stop time to 8n (long enough to see a full cycle of operation.) Your goal is to plot and understand the I/O characteristic. Hence you should plot Vin and Vout, and the currents through each transistor. Select **Outputs | To Be Plotted | Select on Schematic** and ***click on the Vin and Vout*** signal lines, and the ***plus terminal*** of the load capacitor. They will be added to the output list. Accept the changes. Select **Outputs | Save All...** and choose to save all currents and voltages (click the all button for siganls, power, device currents and AC terminal currents). Click ***OK***.

Finally, you will select a value for the load capacitor. Select **Variables | Edit...** The Name of the variable is, of course, **C_Load** and you will assign it a fixed value of **100f** for now. Later you will step through a couple different values using 'parametric analysis.' The setup is now done and the configuration screen should look like as shown in *Figure 18*.

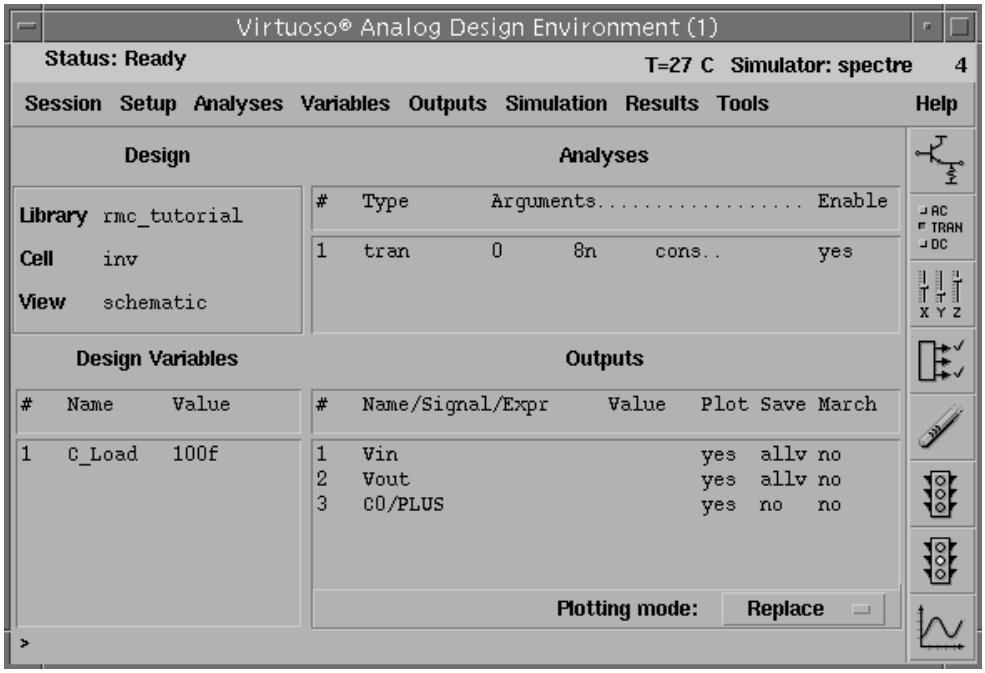

Figure 18: Analog Configuration Screen

Run the simulation. An information window may be shown to you regarding the models in Spectre. Just click OK to continue. Simulation progress will be shown in a separate window (spectre.out). Split the three results onto separate axis (Switch-Axis Mode shortcut button) and re-size. Your results should look like the graphs shown in *Figure 19.* As a trivial exercise, put the results back onto the same axis, and use the waveform cursors to measure the tphl and tplh. Remember that it is most commonly measured from the 50% mark of the input to the 50% mark of the output. You should measure approximately tphl=370ps and tplh=917ps. Save the waveform in the window using '**Window | Save**' and call it '**inv**'.

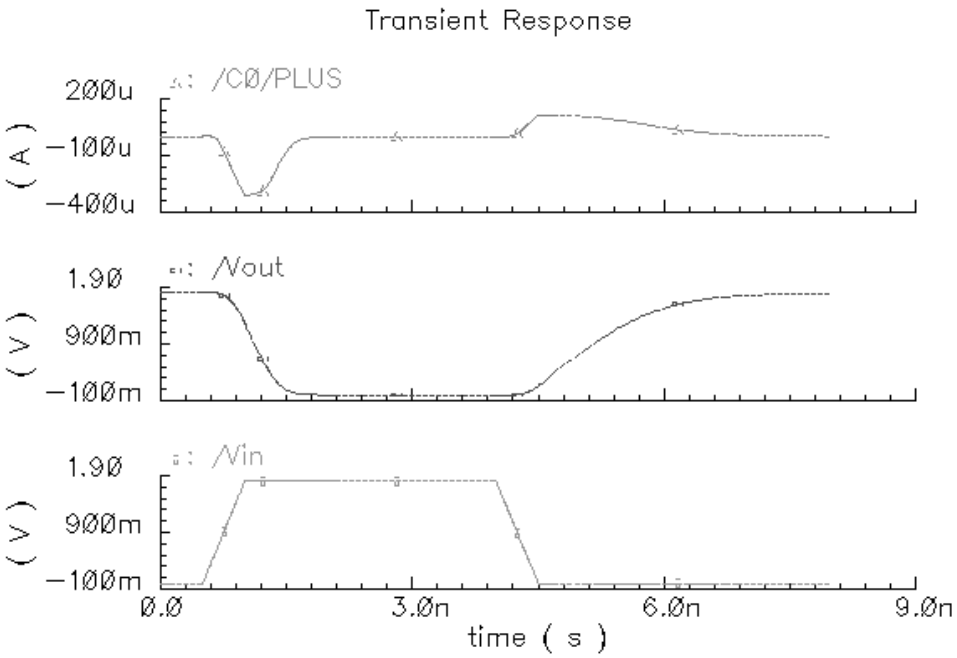

Figure 19: Simulation Results - Split Graphs

One of your final task will involve varying the capacitive load and monitoring the effect on the delay. First, in Analog Artist, remove the C0/PLUS current plot by selecting it and hitting the eraser icon. Select **Tools| Parametric Analysis...** from the Analog Design Environment window**.** *Figure 20* shows the Parametric Analysis window already filled with the correct analysis information. The variable name is **C_Load**, and you want to vary it from **100f to 1p in 5 steps**. From the **same** menu, select **Analysis | Start**. The status window should come alive and run the simulation 5 different times with different loads. Eventually the plots will all come up on the same graph (see plot in *Figure 21*). Of course this parametric analysis can be used to vary any parameter. You can see in *Figure 21* that as C_Load increases, so do rise and fall times.

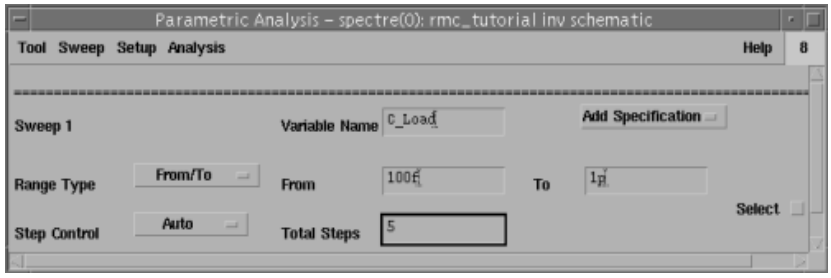

Figure 20: Parametric Analysis Window

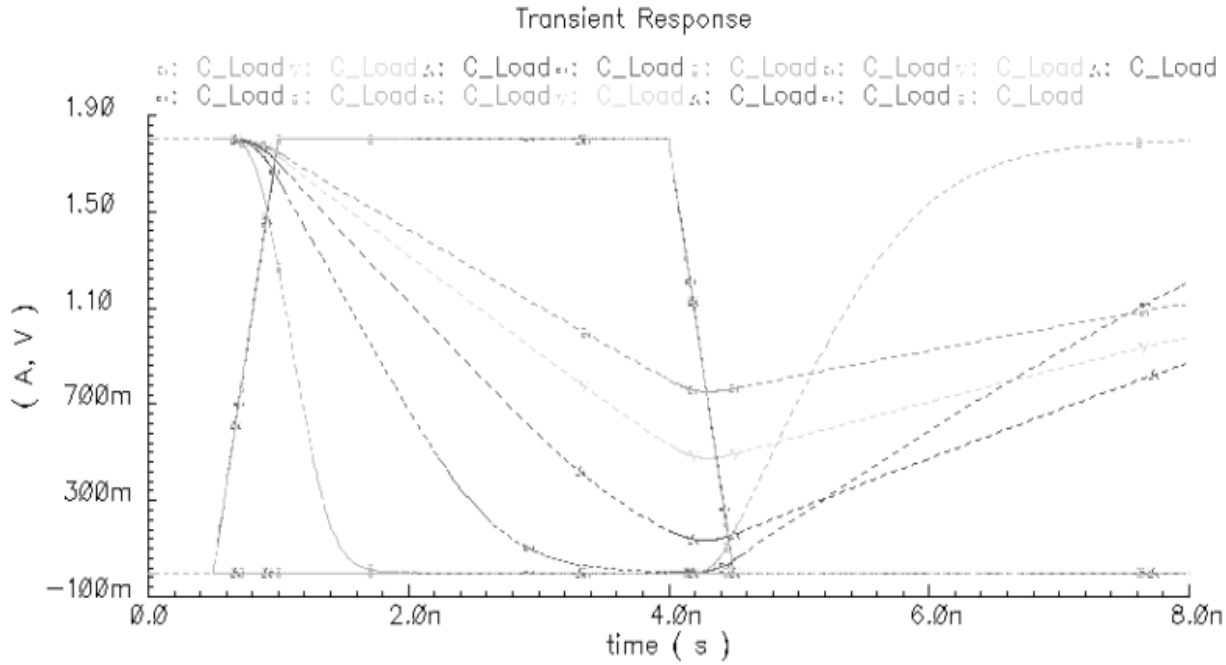

Figure 21: Parametric Analysis Results

6.0 The Waveform Calculator

6.1 Introduction

To analyze resulting waveforms you have an immensely powerful tool in the Waveform Calculator. From a waveform window, select **Tools | Calculator** (see *Figure 22*). As you might imagine, the most straightforward use of the waveform calculator is to add/subtract/multiply/ divide one waveform by another and plot the result. It will perform functions in either the time or frequency domain, operating in either Algebraic mode (Normal), or RPN (Reverse Polish Notation - a la HP calculators). To keep things simple you will use the Algebraic mode (**Options | Set Algebraic**), and only perform calculations based on transient (time-domain) simulations.

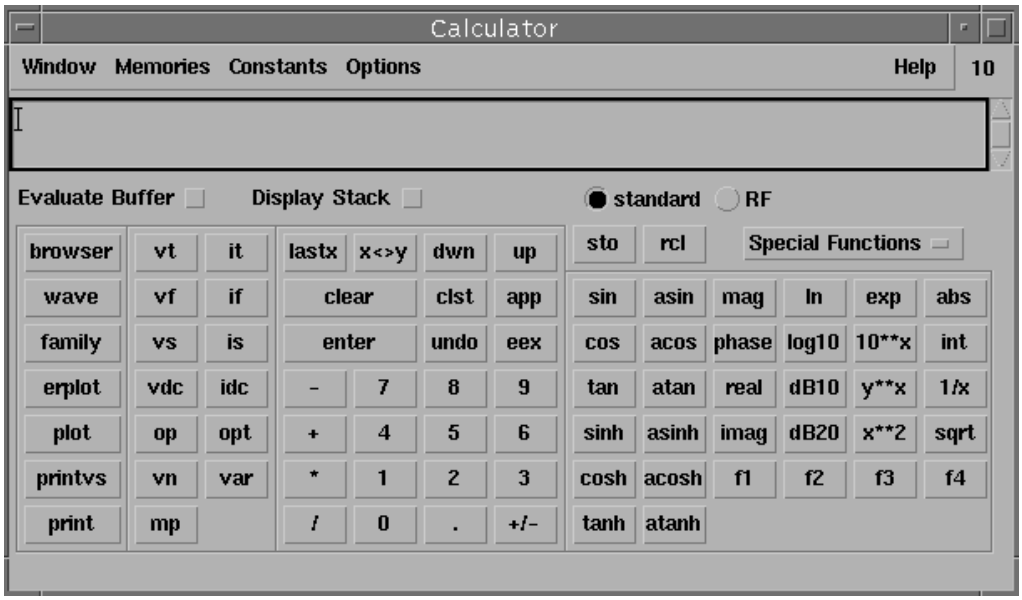

Figure 22: Calculator Window

6.2 Adding Voltages

There are two ways to enter results: from the schematic, or from the waveform plots. As an example, you will be adding the input and output voltages of the inverter. Open the waveform you saved at *Section 5.2* by selecting **Window | Load** and type **inv** and **select Ok**. You may have to redraw the display (run simulation again) if this operation fails. Perform the following operations:

a. On the calculator, **click 'wave'** and select the **Vin waveform** from the plot window;

b. **Click '+'** on the calculator;

c. **Click 'wave'** on the calculator and select the **Vout waveform** from the plot window;
You now have the whole expression to plot the addition of the two waveforms.

d. Click **erplot** from the calculator. It will reset the screen and plot the resulting expression (*Figure 23*).

Alternatively, instead of selecting the waveforms, you could have selected the ***vt*** (current-transient) button from the calculator, and selected the Vin and Vout nets from the schematic. If you wanted current expressions, the nodes of the device would be selected after clicking ***it*** button on the calculator.

Like any other calculator, you can store this result in memory. Select store (**sto** or **Memory | Store**), and save the expression as *voltage_add*. You can load the *voltage_add* waveform by using the ***rcl*** button on the calculator and selecting the *voltage_add* memory location. **Clear** the calculator.

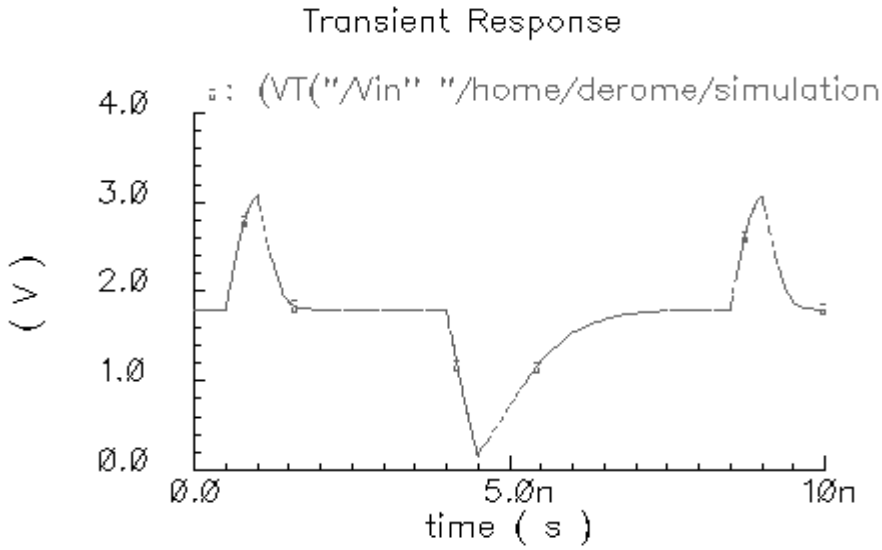

Figure 23: Resulting Voltage Waveform (Calculator)

6.3 Power Calculation

Finally, you want to determine the power consumption of the inverter. $PowerDissipation = \frac{1}{T}\int I_{DD}(t) \bullet V_{DD} = I_{DD}(avg) \bullet V_{DD}$. There are a few places in the circuit where you can measure the dissipation. For example, you could measure the power dissipation across each transistor and add them, or more easily, you can simply measure the current out of the source, and multiply that by the source voltage. You can build this power expression in the calculator or plot the source current and then use the calculator to plot the power dissipation (using the wave button on the calculator). You will be using the second option to find the power dissipation plot.

Building the expression is exactly like using a standard calculator. Remember that we saved all outputs back in *Section 5.0*. Well, you will be using the saved data to plot the source current. Select **it** on the calculator. In the schematic window, select the '**+ node**' of the vdc source. Note that you may need to use the "abs" function if the current from the source is not a positive value. Finally press the **erplot** button to plot the I_{DD} for one cycle.

We are now ready to calculate the power dissipation that we discussed earlier. This can be accomplished by simply multiplying the plot obtained from the source current by the output voltage of the source (~1.8V). Again, the calculator can be used to build the appropriate

expression but this time, we will use the **wave** button to select the absolute value of the source current. The resulting power dissipation plot should look like the one in *Figure 24***.**

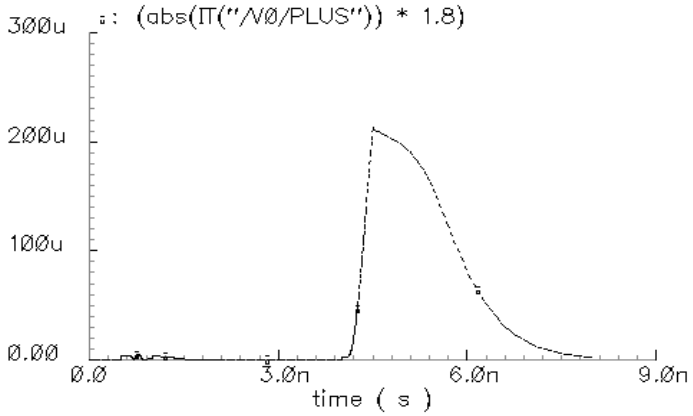

Figure 24: Resulting Power Dissipation Plot

Similar to any digital scope on a lab bench, the calculator can also do such things as calculate delay between two waveforms, find min, max, integrate a waveform, differentiate, calculate overshoot etc. More details about these '**Special Functions'** are available in the on-line help. You will use it to determine the average power. ***Clear*** any current expression in the calculator, select **Special Functions | average**. Click ***wave***, and select the ***instantaneous power waveform*** just created (shown in *Figure 24*). ***Close the bracket*** and ***click '='*** to get the final average power result. Your result should be around 42 uW.

The Getting Started with Cadence tutorial from RMC is completed. You are now ready to tackle basic work using the cadence tools.