# MCC092 Integrated Circuit Design: Lab 1

Schematic Design in Cadence Virtuoso

Version 2018-09-06

2012-2018 (65 nm CMOS): Kasyab Subramaniyan, Stavros Giannakopoulos
Per Larsson-Edefors and Lena Peterson
2008-2011 (130 nm CMOS): Simon Kristiansson and Lena Peterson
Chalmers University of Technology

## Contents

# 1 Lab Purpose

This is the first of four labs where you will have a chance to work with commercial design software for integrated circuit design. While the first lab gives a basic introduction to the Cadence Virtuoso design platform by way of schematic design of a CMOS inverter, the second lab will revolve around the hierarchical schematic design of an 8-bit ripple-carry circuit, whose constituent ripple-carry cell subsequently will be laid out in lab 3. In the final, fourth lab we will consider the design of a clock tree. Here, we will leave the safe haven of the window-based Virtuoso design platform and instead use the Linux command line and directly work with the netlist file, which is a very common work method in advanced research and development projects.

The main purpose of this first lab is that you should get familiar with Cadence Virtuoso and, at the same time, study important dynamic and static characteristics of the CMOS inverter. You will especially investigate if our crude hand-calculation models for the delay come close to the delays computed with the much more complex transistor models used in circuit simulation programs such as Spectre.

# 2 Pre-Lab Assignments

You will find the pre-lab assignment in a separate document available from the course home page in PingPong. You can find it most easily under the assignment called **Pre-lab preparation for lab 1**. This is also where you are to submit your solution to the pre-lab assignment. Note that even though you work in pairs in the lab you are to submit your own individual solution.

No later than at the beginning of the of the in-lab session you will get feedback from the teachers on your solution to the pre-lab assignment. Before you go ahead with the in-lab tasks, you should correct any mistakes you have made in the pre-lab assignment.

# 3 Lab Goal

This first lab is meant to be both a gentle introduction to Cadence Virtuoso and an exercise on the CMOS inverter. After completing this lab you should know how to start Cadence Virtuoso, how to enter the schematic for a CMOS gate and simulate it. You should also be able to find the fan-out-of-four (FO4) delay and noise margin for a CMOS inverter from circuit-simulation results.

In the lab you will carry out the instructions in this memo and show the results to the lab teaching assistants (TAs). If you do not finish on time, you will have to complete the in-lab tasks on your own and show the lab TAs your printed results at a later time (for example, at your next scheduled in-lab session). You should show the lab TAs these results during, or after, the in-lab session:

- The inverter circuit schematic.

- The schematic of the inverter test bench.
- The results from your circuit simulation using the inverter test bench:
  - A plot (or possibly more than one plot) from a transient analysis, from which you have determined the rise and fall time and fan-out-of-four (FO4) delays.
- The results from your DC analysis of the inverter:
  - One or two plots of the voltage transfer curve (VTC) and its derivative with respect to the input voltage, from which you have determined the switching voltage and the noise margins.

# 4 Introduction

In the lab part of this course, you will learn how to use the Cadence Virtuoso design platform (CV for short) for transistor-level electronic design to implement and characterize a standard cell. You will go through a complete design cycle consisting of, first, schematic capture and circuit simulation and, next, physical design ("layout") and verification ("design rule checking").

This first lab is meant both as a gentle introduction to CV and as an exercise on the CMOS inverter. As an important block of standard-cell design flows, i.e., as a buffer, the digital inverter will be extensively re-used in later labs.

The design software from Cadence is widely used in industry. The user interface tends to be a bit complex; be prepared to spend some time on figuring out how to handle CV. We have collected some tips into the Cadence Crib Sheet that is distributed together with the lab memos. In the Crib Sheet, we also explain some conventions used in the lab memos.

All actions that you are to perform are written in bullets. Since you should have read through each lab memo before the lab occasion, the in-lab tasks should not take too long time. However, for that to be true you will have to follow the lab memo closely. Since every step can be performed in many ways, the lab memo has the character of a carefully laid track; if you deviate too much from the main track, later actions may not produce the intended result.

If you have time at the end of the lab, we strongly encourage you to perform one or more of the extra tasks listed in Section 11. They will give you more insights in CMOS-inverter operation and also permit you to experiment with CV more freely. You can also do these tasks later, but then without the benefit of a lab TA to assist you.

# 5 Getting Started

Nowadays we use VNC to connect to a Linux server where all our EDA tools are housed and run. The detailed VNC instructions are available in a separate file, since they are the same regardless of what EDA tool you are to run.

- Sit at one of the lab computers. Follow the instructions on how to connect to the server `heffalump.ita.chalmers.se` using VNC, until you are logged in and have a desktop open.
- Start a terminal window. On `heffalump` this is done with the menu command: **Applications →System Tools→Terminal** in the desktop.

In these lab memos, we assume that the work files from all the labs are located in the same directory. This will be necessary for later labs to work as intended.

- Create a new directory for all these labs, for example with this command in the terminal window:

```
> mkdir -p $HOME/MCC092/cadence
```

Now you are ready to start the CV software and create your inverter schematic:

- Initiate the working directory and start CV:

```
> cd $HOME/MCC092/cadence
> /usr/local/cad/course/MCC092/Y2018/stm065
```

Here you first move to the directory you created above. Then the next line is a Linux command that sets up the required paths and variables needed to run CV and launches CV. This second command must be given each time you log in in order to start CV. It may be a good idea to add it to your `.bashrc` file as an alias so you do not have to remember it every time. After a couple of moments (or minutes depending on your machine and network performance) CV will start.

Now you should see several windows popping up. When all has settled you should find one window of interest left: **CIW** (Command Interpreter Window). In the current version of the tools this window is just called **Virtuoso 6.1.6-64b** at the top. (In addition there are a couple of windows that describe the current version of the program and design-kit features. These windows are of no interest for this lab and can be closed.) From the CIW, invoke the library manager:

- Choose **Tools→Library Manager...** in the **CIW**.

Both the CIW and the library-manager window are briefly discussed in Section 5 in the Crib Sheet.

# 6 Creating a Library and a Cell

To create a work library in CV you use the **Library Manager** which you just invoked.

- Choose **File→New→Library...** in the **Library Manager**. A new window called **New Library** appears.
- Type in a name for your new library, e.g., `labs`, and click on **OK**.

Now you will select a technology file for your library. A new window appears for this purpose. It may pop up behind your other windows, so you may have to look for it.

- In the **Technology File for New Library** window, select the option `'Attach to an existing technology library'`. Click on **OK**.
- Yet another window appears (**Attach Design Library to Technology File**). In the `'Technology Library'` field, scroll through the available technology files. Choose `cmos065` and click on **OK**.

Now that you have created your library you will continue to create your inverter cell. For this you have to return to the **Library Manager** window again.

- Select your lab library in the field `'Library'` in **Library Manager**.
- Choose **File→New→Cell View...** in **Library Manager**.
- Type in a cell name, e.g., `inverter`, into field `'Cell'`. See to it that field `'View'` is `schematic`. In the next step you will enter the schematic of your inverter.
- Click on **OK**.
- If prompted about trying a different license, click **Yes**.

The tool that now appears is called Virtuoso Schematic Editor; a short description is given in Section 7 in the Crib Sheet.

# 7 Drawing the Inverter Cell Schematic

You will implement an X4 CMOS inverter using transistors available in the library `cmos065`. The label X4 refers to drive strength and is a common parameter in cell libraries; X4 means that this inverter is sized up 4 times over the minimum width. The choice of `cmos065` means your design environment will be based on the 65-nm CMOS process technology supported by STMicroelectronics. The `cmos065` library contains symbols for the devices that the foundry offers in the process technology that we are using.

In these labs you will always use four-terminal transistor symbols with explicit connection to the bulk (nMOS transistor) and well (pMOS transistor). All bulk terminals must be connected to ground, and all well terminals to the supply voltage. All transistors are found in the category CMOS. You will always use the low-leakage type of transistors to keep the static power dissipation at a low level; these are called `nsvtlp` and `psvtlp`.

You will also use other components from the library `analogLib`. Here one can find "abstract" components that do not depend on the specific process technology used, e.g., voltage sources, ground symbols and so on.

## Transistors

Now, in the schematic capture phase, we will create our first circuit schematic. First we will place the two transistors needed for the inverter, by adding instances of the components in the library. As for several other commands, there are three ways to add component instances: (i) From the menu system Create→Instance, (ii) with the shortcut `[i]` or (iii) from one of the buttons to the left in the **Virtuoso Schematic Editor** window.

- Press `[i]`. The form **Add Instance** appears.

It is possible to specify the different component parameter values before adding a component to the schematic. One can place several components after each other, even with changes in the form between; for each component added, the current values in the form are used.

- Define your `nsvtlp` transistor first:

  ```
  'Library'  cmos065
  'Cell'     nsvtlp
  'View'     symbol
  'Width'    0.40
  'Length'   0.06
  ```

- The three fields of `'Library'`, `'Cell'` and `'View'` are best entered by using the `Browse` button to the right of the `'Library'` field. First identify your choice in the `'Library'` column, then select `'Cell'` and finally select `'View'`.
- The default unit for length and width is $\mu$m. You should only type in the value without any prefix or unit.
- After typing in the geometry values, press `Enter` and move the symbol to the desired position. You can rotate the symbol using `[r]` and flip it using `[shift-r]` while placing.
- Repeat the above steps for the pMOS transistor: Use the `psvtlp` transistor and a width of `0.8`.
- Finish adding instances by pressing `[ESC]` (with the mouse cursor in the schematic window).

Note that the **Add Instance** form may disappear when switching component type; then you have to re-invoke it using `[i]`.

If you later on want to change, e.g., the width of one transistor, you can select that transistor and press `[q]`. In the form **Edit Object Properties**, change `'Width'` and confirm by clicking on **OK**.

## Supply Voltages

The symbols for the supply voltage and ground are in the library `analogLib`. The cells are named `vdd` and `gnd`.

- Add `vdd` and `gnd` at appropriate places in your schematic.

Note that the symbol in the **Virtuoso Schematic Editing** window doesn't change until you have added the new component.

## Wire Connections

Next you must connect your transistors to supply and ground. You are to use the command **Create→Wire (narrow)** [w], which is explained further in Section 7.3 in the Crib Sheet.

- Connect the components according to your circuit sketch. Do not forget to connect the body terminals of the transistors to `vdd` or `gnd`.

## Signal Names

We can name the signals in the schematic (if we don't, the signals will be assigned more or less random names). There are two reasons for doing this: First, it is easier to interpret and understand the results of the circuit simulations if the signal names have meaning. Second, two signals with the same name are assumed to be electrically connected, and you can omit connecting these explicitly if the circuit schematic becomes more readable then. Read more in the Crib Sheet at page 11 about this and other practical naming conventions!

- Issue the command **Create→Wire Name...** [L]
- Add names to the two signals in your schematic. You can save yourself some mouse clicks by adding several names in the field `'Names'`, separated by spaces. The names will be placed in the schematic in the order they are written.

## Pins

Now when the schematic is done you are to add pins. These will serve as the interface of your inverter cell to the outer world. Pins are defined with the command **Create→Pin...**

- Press `[p]`. The form **Add Pin** is shown.

The pins are placed in the same manner as the transistors. As with the net names, several pins can be named in the field `'Pin Names'`, separated by spaces. The field `'Direction'` gives the direction of the signal in the pin; only pins with the same direction can be placed in the same line of pin names.

- Make sure that 'Attach Net Expression' is selected as No.
- Type in the pin name of the input node, select 'Direction' to be input and place the pin in the schematic.
- Open the form again, type in the pin name for the output, select 'Direction' to be output and place the pin in the schematic.
- Connect the pins to the rest of the schematic with wires.

## Saving Your Schematic

Now you have added all components to the schematic of your inverter cell and are ready to save your work (you should of course save your work more often than this). At the same time CV checks for errors in the schematic, e.g., unconnected wires, and warns you about them.

- Save the schematic with **File→Check and Save** or press [X].

Take a look in the CIW. If there are no errors, 'Schematic check completed with no errors' is printed here. Then you can move on to the next phase of this lab.

If you do have errors, the type of mistake—"warning" or "error"—and a short description will be shown in the CIW. On top of the error message, blinking yellow markers will appear in the schematic where the errors are located.

- Study the errors and correct them.
- Again, save the schematic with **File→Check and Save** or press [X].

## Creating a Symbol

It is very productive to create a hierarchy of cells, since you then may re-use one optimized cell elsewhere. In the event an optimized cell has to be modified late in the design flow, in a hierarchy this modification will propagate throughout the whole design, saving valuable design and verification time.

Since you plan to use your inverter cell as building block inside other schematics, you need to create a symbol that represents it. Here, in CV, the symbol is simply another view of the inverter cell. The easiest way to create the symbol is to generate it automatically from the schematic view.

- Choose **Create → Cellview→From Cellview...** in Virtuoso Schematic Editor.

The form **Create Cellview From Cellview** appears. Here you specify the type of cellview to create. Make sure that 'To View Name' is specified to symbol.

- Click on **OK** in the form **Cellview From Cellview**.

The form **Symbol Generation Options** appears next. Here you specify how the pins should be placed along the perimeter of the symbol. The symbol that is created will be rectangular with the pins placed as specified in this form. In this case the default, with the input to the left and the output to the right, will be appropriate.

- Click on **OK**.

Now a new window appears showing the created symbol. Although one can change the graphical representation of the symbol if one so desires, you are advised to leave it as it is and carry on with the lab. Close the window with the symbol, and the window with the schematic. It is just confusing to have too many open windows.

# 8 Creating a Test Bench for the Inverter

The next step is to create a so-called test bench to be used when investigating the properties of the inverter, namely to identify its FO4 delay. Thus, your test bench will contain five instances of the inverter and a few voltage sources for the signals and the power supplies.

First you create a new cell with the view `schematic`.

- Choose **File→New→Cell View...** in the **Library Manager**.
- Type in a name for the new cell, e.g., `inverter_tb`.
- Click on **OK**.

Now you add an instance of the inverter circuit you just implemented.

- Press `[i]` and select the inverter symbol from the `labs` library.

To give the circuits the proper supply voltage, we use a DC voltage source available in the library `analogLib`, i.e., `vdc`.

- Add one `vdc`, one `vdd` and one `gnd` component to an empty part of your schematic. These components won't be tied to any of the inverter symbols.
- Connect the `vdd` component and the `gnd` component to the `vdc` source's positive and negative terminal, respectively.

  The smart thing with this arrangement is that all instances of `vdd` and `gnd` that have been created in instantiated circuits will be connected to the voltage source `vdc`, since `vdd` and `gnd` are globally defined.

The input signal is best generated by a `vpulse` voltage source.

- Add a `vpulse` in the schematic.
- Connect the positive terminal of the `vpulse` source to the input of the inverter symbol.
- Connect the negative terminal to ground, i.e., to a `gnd` instance.

The output of the inverter, the delay of which you are to investigate, will be loaded by four similar inverters. Add four additional inverters to your the schematic, place them to enable a FO4 configuration, and connect their inputs to the output of the first inverter. You do not have to connect the outputs of the additional inverters to anything.

Make sure that you avoid using very short schematic wires; short can be defined as the height of the voltage source symbols `vdc` and `vpulse`. At this stage you may want to move the existing components apart somewhat to allow wires to extend. The reason we want the longer wires is that we later will select which signals will be presented after simulation and longer wires make it easier to make the selection. As we will see, to present the voltage of a signal node, we will have to select a wire segment. Conversely, to present the current through a node, we will have to select a node.

The connectivity schematic is now complete (we will also need to set some parameters, such as voltage levels, later).

It is always good to save a picture of the schematic for future reference, and while this can be done with any screen-capture tool, CV provides a better means to do so.

- In the schematic window, select **File→Export Image ...**.
- In the new window that appears type in the name that you want of the image and select the appropriate file type. It is strongly suggested to select `.png` as they have good quality and are easy to view in other operating systems (compared to `.eps`).
- Click the folder button and navigate to the directory where the image will be stored and press **Open**.
- It is also suggested you swap the background colors to make the image more clearly readable.
- Select any other options as appropriate and press **Save**.

We will now set the parameters of the different components in the schematic.

- Open the form **Edit Object Properties** by pressing `[q]`.

We begin with the `vpulse` component. We want to choose values for the input signal so that all combinations of the input are run through during the simulation, thereby making sure that our inverter implements the correct logic function. Furthermore, we want to estimate the delay from the input-voltage transition to the output-voltage transition.

- Select the `vpulse` component. Write the parameter values as shown below and click on **Apply**. Note that you should not type in any units, however the prefixes (n and p) are needed.

```
'Voltage 1'    0 V
'Voltage 2'    1.2 V
'Delay time'   1n s
'Rise time'    10p s
'Fall time'    10p s
'Pulse width'  1n s
'Period'       2n s
```

Now, set the values for the supply voltage.

- Select the `vdc` component, set `'DC voltage'` to `1.2 V` and click on **OK** to close the form.

We should now have completed the test-bench implementation. Let us save it and move on.

- Save by pressing `[shift-X]`. Warnings about floating nets can be ignored.

# 9 Simulating CMOS Inverter FO4 Delay

To investigate the performance of our inverter we will run a circuit simulation. We begin with opening the circuit simulation window.

- Choose **Launch→ADE-L** in the **Virtuoso Schematic Editor** window.

A new tool window, **Analog Design Environment** (below called ADE-L), appears. From this window all aspects of the circuit simulations are controlled. You choose which type of simulation you want to run, e.g., AC, DC, transient, etc., which signals you want to view, and what process corner and other simulation parameters you want to use.

You will begin by selecting the type of simulation to use and the parameters needed for the simulation. The types(s) of simulation that have been selected are shown in the field `'analyses'` in the **ADE-L**window.

- Issue **Analyses→Choose...** in the ADE-L window.

The form **Choosing Analyses** appears. Here you see all available simulation types; you should select `tran` which stands for transient, large-signal analysis suitable for digital circuits.

- Select the alternative `tran`.
- As `'Stop Time'`, type in `4n`

- Click on **OK**.

The stop time sets how long time the simulation will continue before it is stopped.

Now you must select which signals to view after the simulation has been run. In this case, the interesting signals are the input and output signals of the inverter, and the current drawn from the power supply. We select these in the circuit schematic.

- Issue **Outputs→To Be Plotted→Select On Schematic** in the **ADE-L**window.
- In the schematic window, select the wires corresponding to the voltages you want to view. The colors of the wires change when they are selected.
- Also specify that you want to view the current flowing from the voltage supply by clicking on the red connection at the top of the supply voltage source. The selection of current through a node will be indicated with a ring around the chosen node.
- Press `[ESC]` when you are done.

The selected signals should now be shown in the field `'Outputs'` in the **ADE-L**window. Check that they really are the ones you want to view.

Finally, you must select the corner to use. If this is not done, no simulation models are loaded since there is no default value and the simulation will abort.

- Issue **ArtistKit→*Setup Corners...** in the **ADE-L**window. A form called **Setup Corners** appears.

In this form do this:

- Under Scenario, select TT (the top option) for GLOBAL VARIATIONS.
- Click on **Save Model File** (make sure you don't mistake it for the **Save scenario** button).
- Close the window by selecting **Session→Close**

Now we are finally ready to simulate the circuit.

- Select **Simulation→Netlist and Run** or the short cut in ADE-L.
- Click on **Yes** as an answer to the question if you want to save the signals.

The simulation is now started. A separate window appears showing the progress of the simulation.

When the simulation has completed a waveform window appears with graphs of the signals you selected to view. To see the signals more clearly, you can plot each of them in a separate graph.

- Check the box next to **Graph→Split all strips** in the waveform window.
- Verify that the inverter functions correctly.

The Horizontal Cursor can help identifying the maximal or minimal value in a graph. Select **Trace→Horiz Cursor** or press `[h]`; while you move the cursor, notice the numerical values presented next to the cursor.

- Measure the largest current drawn by your circuit. This is most easily done with the signals in separate graphs. You may also want to use the zoom feature (`[shift+wheel]` or `[ctrl+wheel]`), after which you return to the full graph by pressing Fit `[f]`. Alternatively, you can use the zoom bar right above the graphs to adjust the exact zoom.

To measure distances in the graphs, markers can be used. To place the first marker press [a] to place it. For the second marker press [b]. The distance between the two markers will be presented, as well as the slope.

- Measure the rising and falling FO4 delays of your circuit. This is easiest with the signals in the same graph.
- Measure the rise and fall times of $V_{out}$. Are they different? Why?

Write down the results for the three tasks above. Compare the results for the FO4 delays with the ones from your pre-labs. Show your results to the lab TAs.

When you are done, close down the simulation windows and the schematics window.

# 10 CMOS Inverter VTC and Noise Margin

In this part of the lab you are to investigate the voltage transfer curve (VTC) of the inverter and specifically the switching voltage and the noise margin.

For this experiment you will use the same inverter as in the first experiment. You will follow steps similar to the ones above: creating a test bench, setting up a simulation, running it and plotting the results.

The first step is to make an additional test bench for `inverter`. You may want to consult previous parts of the lab memo to get all the details of these steps:

- In the library manager create a new cell view called `inverter-dc_tb` and open its schematic view.
- In this schematic place an instance of `inverter`.
- Add a DC voltage source `vdc` and connect its positive node to the `inverter` input. Add a `gnd` symbol and connect the negative terminal of the voltage source to ground.

- Add another DC voltage source `vdc` and `vdd` and `gnd` symbols, and connect the source between them and set its voltage to 1.2 V.
- Add a piece of wire to the output of the `inverter` instance.
- Name the input wire `in` and the output wire `out`.
- Check and save.

Now you are to simulate the DC transfer curve, also known as voltage transfer characteristic (VTC), of this inverter:

- Invoke the Analog Environment (ADE-L).
- Select **Analyses→Choose** so the **Choosing Analyses** window appears. Select `dc` to enable DC sweep simulation.
- In the same window, set `'Sweep Variable'` to `Component Parameter` and then press **Select Component**. Now, in the schematic, select the `vdc` source connected to the input of `inverter`.
- A window called **Select Component Parameter** appears. In this window select the top line, i.e., `''DC voltage''` and press **OK**.
- Now when you get back to the **Choosing Analyses** window, under `'Sweep Range'` set `Start` to `0` and `Stop` to `1.2`.
- Set `'Sweep Type'` to `Linear`. Select `Number of Steps` and type in `1000`.
  **Note:** Take great care to set the `'Sweep Type'` and `Number of Steps` correctly otherwise the simulation will most certainly give you faulty results.
- Press **OK**.

Now you should set up the corner to be used in the simulation to be TT, and the signal voltages to plot. Since this is a DC sweep, where the parameter you sweep is connected to the inverter input, the input voltage will be on the x-axis of the resulting graphs. We only need to select the inverter output. However, to easily find the switching voltage it is best to plot both the inverter input and its output.

Run the simulation. From your simulation what is the switching voltage? Does the value you find from the simulation correspond to the one you calculated in your pre-lab? You may have to zoom in to see it well. You can also use a smaller sweep range in the DC simulation to only get the part of the VTC that is of interest to you. To open the **Choosing Analyses** window, double click on the simulation line in the **ADE-L** window.

Next you are to find the noise margin for the inverter. For this purpose, it is helpful to derive the gain from your voltage curve, because you need to find the point where the gain equals -1. To derive the gain you have to take the derivative of the voltage curve with respect to the input voltage. You will use the **Calculator** for this purpose. But before you go on, remove the input voltage from the voltages being plotted as it will just confuse the following plots; then rerun the simulation.

- In the graph window click on the curve for the output voltage. It should then be highlighted.
- Right click and select **Send To→Calculator...**. Another window called **Virtuoso Visualization & Analysis XL Calculator** (**Calculator** from now on) appears.
- At the top part of the window you should be able to see a white area with `v("/out" ?result "dc")` written in it (the `out` part might differ depending on how you named your output wire).
- In the lower, left part of the **Calculator**, select **Function Panel**; a number of functions are presented. Click on the function `deriv`. The value at the top part should have changed to `deriv(v("/out" ?result "dc"))`.
- To evaluate, and get the associated function presented in the graph, press **Tools→Plot**.
- The derivative of the VTC should now show up in your graph window. If it appeared on a different sub-window you can drag and drop it on the same window as the VTC curve.
- If you are satisfied with the derivative curve you can set `ADE-L` to generate it every time you re-run the simulation. On the **Calculator** window press **Tools→Send Buffer To ADE L/XL Outputs**.

You now need to find the two points where the gain (the derivative) equals -1. Experiment with the markers to mark these two point in the graph window. The vertical markers (`[v]`) can be useful. Find the two corresponding points on the VTC. What are the four voltages you need to calculate the noise margins and what are the resulting noise margins? Do the values correspond well with what you calculated in the pre-lab?

Once you are satisfied with your simulation and your graphs, save your curves on an image file and show the lab TAs your result.

To save your graphs on a image file, do this:

- In the graph window, select **File→Save Image ...**.
- In the new window that appears navigate to the directory where the image will be stored, type in the name that you want of the image and select the appropriate file type. It is strongly suggested to select either .pdf or .png as they have good quality and easy to view in other operating systems.
- Select any other options as appropriate and press **Save**.

Close down all windows.

# 11 Extra for Those Who Have Time

Go back to the FO4 experiment in Section 9 and test running the same experiment using the corners SS and FF. What delays do you get for these corners? After you are done, be sure to set

the used corner in the model file back to TT (this is important because your inverter cell will be re-used in the coming labs).

Make the transistors in the `inverter` cell used in the noise-margin experiments in Section 10 ten times longer while keeping the width the same. Redo the noise-margin experiment. How did the noise margin change? How did the maximum gain magnitude change? Be sure to reset the transistor lengths after you are done.

You can also test changing the widths of the transistors, e.g., by making one of them four times wider than before, and check how the inverter switching voltage changes in the noise-margin experiment in Section 10. What change would you expect? Does it change as expected?

# 12 The End

In this lab exercise, you have learnt the rudiments of how to create a circuit schematic using components from different libraries. You have also investigated the dynamic and static properties of the CMOS inverter circuit using circuit simulations.

In preparation of the next lab, you will design a ripple-carry cell. In the lab, you will enter a hierarchical schematic, simulate it and ponder the implications of the simulation results.