Example: Post-layout Simulation (CMOS Inverter)

Step 1: Extracting from the Layout

The mask layout only contains physical data. In fact it just contains coordinates of rectangles drawn in different colors (layers). The extraction process identifies the devices and generates a netlist associated with the layout.

Make sure you have a layout window with a finished design ready. Make sure that the design does not contain any DRC errors.

1. From the Verify menu select the option Extract
   (verify --> Extract)

A new window with extraction options will appear. The default options will only extract ideal devices. This ideal case would result in a list much similar to the schematic. For a more accurate representation, however, we will have to take the parasitic effects into account. To enable the extraction of parasitic devices, a selection parameter called a switch has to be specified. You can type the switch into the designated box, or you can select it from a menu using the Set Switches option.

The switch specified in the example (above) to enable extracting the parasitic capacitances is called Extract_parasitic_caps.
Check the Command Interpreter Window (the main window when you start Cadence) for errors after extraction.

Following a successful extraction you will see a new cell view called *extracted* for your cell in the library manager. See the following section for accessing the extracted view.
Example: Post-layout Simulation (CMOS Inverter)

Step 2: The Extracted Cell View

Following the extraction step a new cellview is generated in your library. This cell view is called **extracted view**.

Try loading the cellview. It will open up a layout that looks almost identical to the layout you have extracted. You will notice that only the I/O pins appear as solid blocks and all other shapes appear as outlines.

The red rectangles indicate that there are a number of instances within this hierarchy. Try pressing **Shift-F** to see all of the hierarchy.
This will reveal a number of symbols. If you zoom in you will be able to identify individual elements, such as transistors and capacitors. You will notice that the parameters (e.g., channel dimensions) of these devices represent the values they were drawn in the layout view.

Apart from your actual devices you will notice a number of elements, mainly capacitors in your extracted cell view. These are not actual devices. They are parasitic capacitances, side effects formed by different layers you used for your layout.

The next step will be to correspond the extracted netlist to that of the schematic. This is called the Layout Versus Schematic checking. This will ensure that the schematic that we have drawn and the layout are identical.
Example : Post-layout Simulation (CMOS Inverter)

Step 3 : Layout Versus Schematic

In this step we are going to compare the schematic and the extracted layout to see if they are identical.

1. From the Verify menu select the option LVS.

If you had previously run a LVS check, this would pop-up a small warning box. Make sure that the option Form Contents is selected in this box.

The top half of the LVS options window is split into two parts. The part on the left corresponds to the schematic cell view and the right part corresponds to the extracted cell view that is to be compared. Make sure that the entries in these boxes represent the values for your circuit.

Although there are a number of options for LVS, the default options will be enough for basic operations, select Run to start the comparison.
The comparison algorithm will run in the background, the result of the LVS run will be displayed in a message box. Be patient, even for a very small design the LVS run can take some time (minutes).

![Job succeeded message box]

The succeeded message in the above message box, indicates that the LVS program has finished comparing the netlists, **NOT THAT THE CIRCUITS MATCH**. It might be the case that the LVS was successful in comparing the netlists and came up with the result that both circuits were different.

To see the actual result of an LVS run you have to examine the **output** of the LVS run. The *Output* option is right next to the *Run* command.

![LVS output example]

You can take a look at the complete LVS result [here](#). The most important part of the report can be found in the figure above. It states that the netlists did indeed match. If you discover that there is a mismatch, you must go back to the layout view and correct the error(s).

Most of the other options on the LVS form, are for finding mismatches between two netlists and to generate netlists that include only parasitic effects relevant to one part of the circuit.
Step 4: Summary of the Cell Views.

So far you have created a number of cell views corresponding to the same circuit. In this section we want to review all of these cellviews and discuss why they are used.

1. Schematic view

For any design, the schematic should be the first cell view to be created. The schematic will be the basic reference of your circuit.

2. Symbol view

After you are done with the schematic, you will need to simulate your design. The proper way of doing this is to create a separate test schematic and include your circuit as a block. Therefore you will need to create a symbol.
3. Layout view

This is the actual layout mask data that will be fabricated. It can be generated by automated tools or manually.

4. Extracted view

After the layout has been finalized, it is extracted, devices and parasitic elements are identified and a netlist is formed.
5. Test Schematic

A separate cell is used to as a test bench. This test bench includes sources, loads and the circuit to be tested. The test cell usually consists of a single schematic only.
After a successful LVS you will have two main cell views for the same circuit. The first one is the schematic, which is your initial (ideal) design, the second is the extracted, which is based on the layout and in addition to the basic circuit includes all the layout associated parasitic effects. Since both of these views refer to the same circuit they can be interchanged.

In this example we are going to re-run the simulation example, but we will make the simulator to use the extracted cell view instead of the schematic cell view.

Make sure that you are in the test schematic, that you used to simulate your design earlier.

1. Start Analog Artist using **Tools --> Analog Artist**

The Analog Artist window will pop up.

2. From the **Setup** menu choose the **Environment** option.
A new dialog box controlling various parameters of Analog Artist will pop up.

The line that we will have to alter is called the *Switch View List*. This entry is an ordered list of cell views that contain information that can be simulated. The simulator (in fact the netlister) will search until it finds one of these cellviews. The default entry does not contain an extracted cellview. We will simply add an entry for extracted cellview in front of the schematic cellview.

As a result of this modification, the simulator will use the extracted cell view of the cell, if one is available.

3. Choose analyses

In the simulation example a transient analysis was used, this time we will use a DC simulation. In a DC simulation the value of any voltage or current source is varied over a specified range. It is used to obtain input/output characteristics of circuits.
The basic options of the DC analysis are not very straightforward. The first step is to determine what parameter will be swept.

Choose *Component Parameter* as the Sweep Variable.

You can select the parameter from the schematic window after you click on *Select Component*. 
As each component has a number of parameters, you will be given a list of parameters associated with the component you select.

In the example given above we have selected the DC voltage of the voltage source as the sweep variable.

After we have selected the variable we can decide, the range where the variable will change.

This example changes the DC voltage source connected to the input from 0 Volts to 3.3 Volts.
The last parameter determines how the sweep will be performed. A linear sweep will increment the value of the sweep variable by a fixed amount. The example below uses a step size of 10 mill volts.

From this point on the simulation will continue just as it has been described in the Simulation Tutorial, except for the fact that the results will now include parasitic effects from the actual layout.