should be specified as `divaExt.rul`. The “Extract Method” should be “flat,” "OK" the form. DIVA extraction will run, created an extracted view in your Library Manager. Using this extracted view, we can now run LVS. From your Virtuoso layout window, choose **Verify → LVS**. For your schematic netlist, select the cell `tutorial2.inverter.schematic`. For the extracted netlist, select the cell `tutorial2.inverter.extracted`. The “Rules Library” switch should be selected and the rules library specified as `parts`. The “Rules File” field should be specified as `divaLVS.rul`. Hit the “Run” button to kick off the job in background. You can monitor your job by hitting the monitor button. This opens a “Analysis Job Monitor” window. A job that has successful run will show a status of “succeeded.” This does not mean that the job was LVS clean. You need to look at the run log which you can get from the **Command → Show Run Log** option of the “Analysis Job Monitor” window. If your design is LVS clean, then you should get the remark “The net-lists match” in the run log.

Assuming that your design is LVS clean, we now want to extract the layout to get more accurate circuit simulation results. To do this, go back to the layout editor window for `inverter` and select **Verify → Extract**, as you did before. This time, however, we want to extract with capacitance parasitics for the wires. Hit “Set Switches”. A form will open up. Select “Cparasitics” and hit “OK.” This will tell the extractor that you want to extract parasitics. "OK" the “extractor” form.

Now we want to simulate the extracted results. Go back to your `test_inverter` schematic, which you used to simulate your inverter schematic. Set up the simulation exactly the way you did before, except from the Analog Artist Simulation window, choose **Setup → Environment**. On the “Switch View List”, add “extracted” to the front of the list. This will ensure that the netlist will run from your extracted netlist rather than from your schematic. Compare the results you get with the results from your schematic simulation. You might want to also look at the HSPICE netlist generated from extraction to see the parasitics generated.
contacts. Before drawing any wires, you need to select the correct drawing layer. You do this in the LSW (layout selection window) by selecting the appropriate layer. Each layer is also assigned a purpose, such as net or drawing. You will usually use purpose drawing, but this is not that important since all the purposes will be streamed out and treated equivalently when the chip is released. First choose the metal1-drawing layout from the LSW. By using **Create → Rectangle** or **Create → Path**, add in the wire. Do the same for the poly connection between the gates, this time using the poly-drawing layer. Notice that **Create → Path** automatically chooses the minimum width for the route. To change this, you need to hit F3 while the command is active and an option form will appear, in which you can change this width.

Using **Create Polygon**, create metal1 power and ground busses running horizontally at the top and bottom of your layout. You must also create substrate and nwell contacts. To create an nwell contact, you can use an instance of **NTAP** from library **parts**. You will have to use **Create → Rectangle** to create the appropriate nwell extension to accommodate the contact. Similarly, the substrate contact can be made with an instance of **PTAP** from the **parts** library.

Once your layout is completed, you will need to run design rule checking (DRC) using DIVA. To do this, choose **Verify → DRC**. The DRC form will open up. Ensure that you are using “Checking Mode” of “flat.” Also, the “Rule Library” option should be selected and rules library field filled in as **parts**. The “Rules File” field should read divaDRC.rul. This reads the DRC rules from our technology library (**parts**). The DRC rules file for our technology stored in this library is divaDRC.rul. "OK" the form to run DIVA DRC. The results of the run will be displayed in the CIV. If there are any errors, they will also be highlighted in your layout. Correct any DRC violations and re-run DIVA, repeating this until your layout is DRC clean.

Now that your layout is DRC clean, the next step in layout verification is to ensure that your layout matches your schematic, both in the connectivity of your transistors as well as in transistor widths and lengths. To do this, we need to run Layout- Versus- Schematic (LVS) checking. First, we must extract the layout. Our first extraction will only extract the devices; that is, we will not extract any of the parasitic capacitances associated with the wires. To do this, choose **Verify → Extract**. The “Rules Library” switch should be selected and the rules library specified as **parts**. The “Rules File” field
Figure 1: Final example layout for this tutorial.
Columbia University  
Department of Electrical Engineering  
EE E4321. Tutorial #2. Layout design, DRC, IVS, and extraction.

In this tutorial, we will layout an inverter, run design-rule checking (DRC) and layout-versus-schematic (IVS) checking, and extract the layout for simulation.

First create a library called tutorial2. In this library, create a cell called inverter, which contains an nmos and pmos wired up as an inverter. Both devices should have the layout type of “cc” with nf of 1. Let the width of the pnet be 6\( \mu \)m and the width of the nnet be 3\( \mu \)m. Create two pins, an input pin a and an output pin y.

Just to refresh your memory on how to simulate things in HSPICE (and also so that we can compare simulation results from schematics and from extracted layout), create another cellview test_inverter which contains an instance of your inverter cell and the appropriate sources to do a simulation of the rising and falling delay of the inverter with a load capacitance of 40\( \mu F \).

Now we will create the layout cellview for inverter. Choose File → New → Cellview. On the Create New File form, choose a library name of tutorial2, the cell name inverter, and the view name layout. “OK” the form. This bring up the Virtuoso layout editor.

There are two grid displays: minor grids and major grids. The minor grid spacing is set to 0.3\( \mu \)m (which is \( \lambda \) in our design rules), while the major grid spacing is set to 3.0\( \mu \)m. There are zoom-in and zoom-out options under Window. Choosing Window → Zoom To Grid zooms out to the smallest magnification at which the minor grid points are visible.

Now we are ready to draw objects in the layout window. An example of a final layout is shown in Figure 1 for you to use as a guide as you complete the tutorial. Select Create → Instance. Use the library parts and the cell nmos. Fill out the parameter fields with ltype of “cc” and nf of 1. Let the length be 0.6\( \mu \)m and the width be 3\( \mu \)m. Shift-F and Control-F let you display/not display the layout contents of the parameterized nmos layout cell that you have instantiated. Hitting the ESC key will stop the placing of repeated instances. Now, similarly place a pmos of width 6\( \mu \)m. For now just place it above the nmos.

Now that you have placed both transistors, you need to draw the connections to create an inverter. We also need to place substrate and nwell