Layout and LVS

Open-Source Analog Design Flow
Design Flow for Analog Integrated Circuits
Fabrication and PDK

Process Design Kit:

Set of specifications of the fabrication process, defined by the foundry. It includes:

- Available layers
- Physical and electrical properties
- Device primitives
- Simulation models
- Set of rules that ensure manufacturability of the design (Design Rule Check o DRC)
Since the vertical order of the layers is known, it is sufficient to represent the design seen from above, like a blueprint.

The shape of the polygons drawn in the different layers provide information of the physical geometry and distribution of the designed devices.
Layout
Before layout

During schematic design, once the simulations show the desired behaviour, the (preliminary) size of the devices are chosen.

In the following example, we consider the layout of a CMOS inverter, with lengths $L = 0.3 \text{ um}$ and widths $W_n = 1.5 \text{ um}$ and $W_p = 4.5 \text{ um}$.

In the example we use Magic and work with the iic-osic-tools Docker from JKU.
Magic: GUI

- Grid options
- DRC check
- Information: technology, layer
- Selection Box
- Layers: left click to make layer visible, wheel to paint the box and right click to make layer invisible

Console

www.opencircuitdesign.com/magic/
Magic: commands

Comandos:

- **box** `<x1> <y1> <x2> <y2>`: Choose box coordinates
- **paint** `<layer_name>`: Paint box
- **getcell** `<cell_name>`: Instantiate .mag file
- **sideways**: Mirror horizontally
- **upsidedown**: Mirror vertically
- **rotate** `[degrees]`: Rotate (90º by default)
- **label** `<name> <dir> <layer>`: Put label box
- **source** `<file_name>`: Run .tcl script
- **extract**: Extract devices
- **ext2spice**: Convert extraction into .spice
- **save** `<cell_name>`: Save .mag file
- **load** `<cell_name>`: Load .mag file

Note: all the actions in the GUI can be executed with commands, and written in .tcl scripts.
¿Which type is the substrate?

¿Where are the transistor terminals?

NMOS Layout

Execute magic and check technology in the upper right corner.

Draw the NMOS from the image, size W=150nm and L=30nm

<table>
<thead>
<tr>
<th>Rule</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>difftap.1</td>
<td>diffusion min width</td>
</tr>
<tr>
<td>difftap.3</td>
<td>diff. to diff. min spacing</td>
</tr>
<tr>
<td>poly.1a</td>
<td>polysilicon min width</td>
</tr>
<tr>
<td>poly.2</td>
<td>poly to poly min spacing</td>
</tr>
<tr>
<td>poly.7</td>
<td>ndiffusion overhang of nFET</td>
</tr>
<tr>
<td>difftap.2</td>
<td>Transistor min width</td>
</tr>
<tr>
<td>poly.8</td>
<td>min poly overhang of FET</td>
</tr>
</tbody>
</table>
On the window menu, select a 0.5um grid and enable snap to grid. Paint a 1.0um*1.5um rectangle width ndiffusion layer.

Change the grid size to 0.05 and paint a 0.3um*1.8um rectangle with polysilicon. This will generate a 0.3um*1.5um ntransistor.

Paint a 0.2um*1.2um rectangle with ndcontact on each side.

Paint a 0.3um*1.4um rectangle with locali on each side.

Check DRC
A PMOS transistor has a similar construction to the NMOS, except it is inside an nwell and diffusions are p+ type. The required layers are:
- nwell
- pdiffusion
- polysilicon
- pdcontact
- locali

For the example inverter we need a PMOS with $W=4.5\, \text{um}$ and $L=0.3\, \text{um}$. What about aspect ratio?
Device Generator

It is not necessary to do layout manually for each device.
Layout v/s Schematic
Layout v/s Schematic (LVS)

In Xschem:
- Generate .spice by pressing ‘Netlist’
- Open netlist with ‘Simulation> Edit Netlist’
- Save .spice in work directory

In Magic:
- Extract with: extract
- Configure with: ext2spice lvs
- Configure with: ext2spice subcircuit top off
- Get .spice: ext2spice

Once both netlists are in the same directory, execute netgen

netgen -batch lvs net1.spice net2.spice $PDK_PATH/tech/netgen/sky130A_setup.tcl lvs.log
LVS log

http://opencircuitdesign.com/netgen/tutorial/tutorial.html#Results

<p>| Subcircuit summary: |</p>
<table>
<thead>
<tr>
<th>Circuit 1: inverter_schem.spice</th>
<th>Circuit 2: inverter_layout.spice</th>
</tr>
</thead>
<tbody>
<tr>
<td>sky130_fd_pr__nfet_01v8 (1)</td>
<td>sky130_fd_pr__nfet_01v8 (1)</td>
</tr>
<tr>
<td>sky130_fd_pr__pfet_01v8 (1)</td>
<td>sky130_fd_pr__pfet_01v8 (3-&gt;1)</td>
</tr>
<tr>
<td>Number of devices: 2</td>
<td>Number of devices: 2</td>
</tr>
<tr>
<td>Number of nets: 4</td>
<td>Number of nets: 4</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Netlists match uniquely.</td>
<td></td>
</tr>
<tr>
<td>Cells have no pins; pin matching not needed.</td>
<td></td>
</tr>
<tr>
<td>Device classes inverter_schem.spice and inverter_layout.spice are equivalent.</td>
<td></td>
</tr>
<tr>
<td>Final result: Circuits match uniquely.</td>
<td></td>
</tr>
</tbody>
</table>
Post-Layout Simulation
Parasitic extraction

```
* NGSPICE file created from inverter.ext - technology: sky130A

.subckt inverter
X0 Vdd Vin Vout Vdd sky130_fd_pr_pfet_01v8 ad=0.3 pd=1.9 as=0.3 ps=1.9 w=1.5 l=0.3
X1 Vout Vin Gnd Gnd sky130_fd_pr_nfet_01v8 ad=0.525 pd=3.7 as=0.525 ps=3.7 w=1.5 l=0.3
X2 Vout Vin Vdd Vdd sky130_fd_pr_pfet_01v8 ad=0.3 pd=1.9 as=0.525 ps=3.7 w=1.5 l=0.3
X3 Vout Vin Vdd Vdd sky130_fd_pr_pfet_01v8 ad=0.525 pd=3.7 as=0.3 ps=1.9 w=1.5 l=0.3
.ends
```

Netlist with parasitics

```
X0 Vdd Vin Vout Vdd sky130_fd_pr_pfet_01v8 ad=0.3 pd=1.9 as=0.3 ps=1.9 w=1.5 l=0.3
X1 Vout Vin Gnd Gnd sky130_fd_pr_nfet_01v8 ad=0.525 pd=3.7 as=0.525 ps=3.7 w=1.5 l=0.3
X2 Vout Vin Vdd Vdd sky130_fd_pr_pfet_01v8 ad=0.3 pd=1.9 as=0.525 ps=3.7 w=1.5 l=0.3
X3 Vout Vin Vdd Vdd sky130_fd_pr_pfet_01v8 ad=0.525 pd=3.7 as=0.3 ps=1.9 w=1.5 l=0.3
C0 Vin Vdd 0.332f
C1 Vout Vdd 0.307f
C2 Vout Vin 0.225f
C3 Vout Gnd 0.301f
C4 Vin Gnd 0.513f
C5 Vdd Gnd 0.979f
.end
```