# *Using Magic VLSI to layout and simulate a Skywater 130nm CMOS ring oscillator*

## 1. Layout

To layout and simulate the ring oscillator we will use the inverter cell magic_inv.mag.

```
magic magic_ring.mag
```

Set the grid to 50nm × 50nm and set the cursor to snap to the grid

```
% grid 50nm
% snap user
```

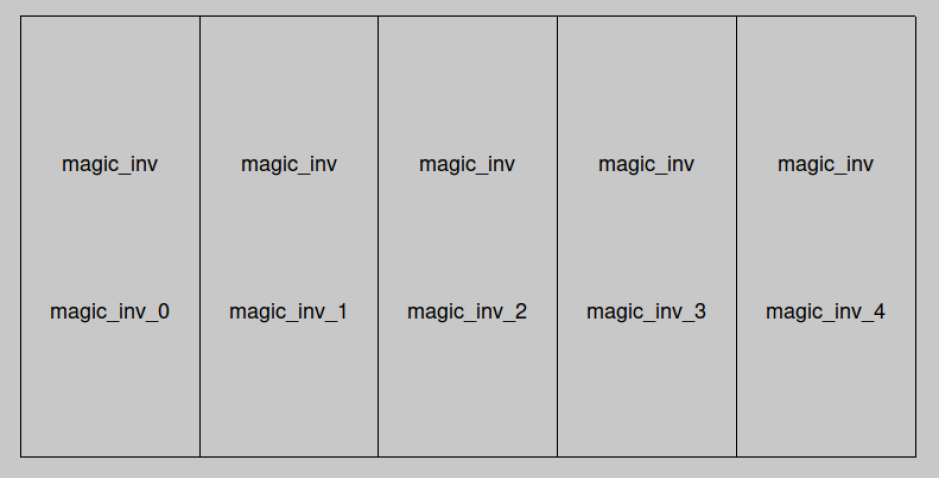
Select the coordinates were to insert the first inverter cell:

```
% set box 0 0
% set position 0 0
```

Get the cell using the command:

```
% getcell magic_inv
```

Select the cell and copy it 4 times, so that there are in total 5 (odd number) instances of it.




Select and expand one of the cells

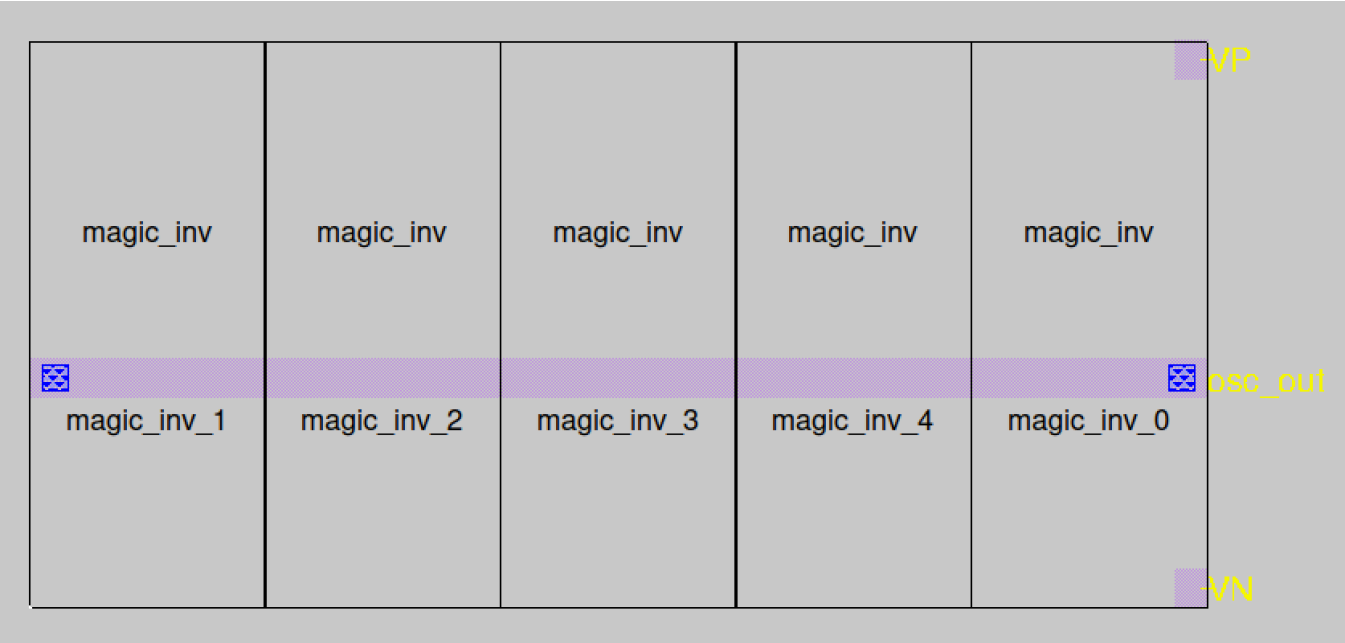
```
% expand
```

Selecting and copying the cell by hand 4 times is not a big deal, however if the number of copies is large one can use the array command:

```
% getcell magic_inv
% array 5 1
```

When using the array's command the cells' instance name follow a different convention.


To join the input of the first inverter to the output of the last inverter, use a layer of metal1 (m1) and connect it with `mcon` (a.k.a. `viali`) to the underlaying local interconnect layer (`li`). Next, paint some metal1 on top of the power rail and the ground rail, so we can add all the labels at the top hierarchy level. To add the labels for the power rail, the ground rail and the output signal of the ring oscillator use the `Edit > Text` menu in the layout window.



2. **Layout vs. Schematic**

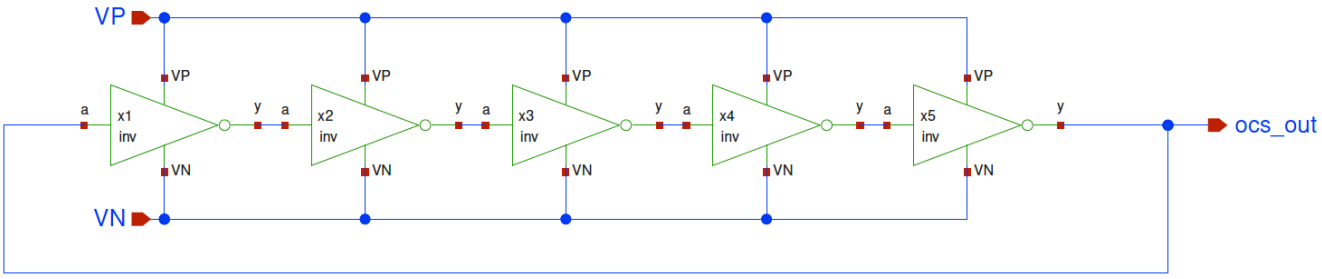
From within magic extract the LVS spice netlist of the layout:

```
% extract all
% ext2spice lvs
% ext2spice -d
```

The spice netlist generated is `magic_ring.spice`:

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

.subckt magic_inv A Y VP VN
X0 Y A VN VN sky130_fd_pr__nfet_01v8 ad=4.5e+11p pd=2.9e+06u as=4.5e+11p ps=2.9e+06u w=1e+06u l=150000u
X1 Y A VP VP sky130_fd_pr__pfet_01v8 ad=9e+11p pd=4.9e+06u as=9e+11p ps=4.9e+06u w=2e+06u l=150000u
.ends

.subckt magic_ring
Xmagic_inv_0[0] osc_out magic_inv_0[1]/A VP VN magic_inv
Xmagic_inv_0[1] magic_inv_0[1]/A magic_inv_0[2]/A VP VN magic_inv
Xmagic_inv_0[2] magic_inv_0[2]/A magic_inv_0[3]/A VP VN magic_inv
Xmagic_inv_0[3] magic_inv_0[3]/A magic_inv_0[4]/A VP VN magic_inv
Xmagic_inv_0[4] magic_inv_0[4]/A osc_out VP VN magic_inv
.ends
```

Create the schematic (`ring_xschem.sch`) of the ring oscillator using xschem:




The netlist associated to the schematic is `ring_xschem.spice`:

```
**.subckt ring_xschem VP VN ocs_out
*.ipin VP
*.ipin VN
*.opin ocs_out
x1 ocs_out net4 VP VN inv
x2 net4 net1 VP VN inv
x3 net1 net2 VP VN inv
x4 net2 net3 VP VN inv
x5 net3 ocs_out VP VN inv
**.ends

* expanding   symbol:  inv.sym # of pins=4
* sym_path: /home/talarico/ihome/ngs406/layout/tutorial_sky130/inv.sym
* sch_path: /home/talarico/ihome/ngs406/layout/tutorial_sky130/inv.sch
.subckt inv  a  y  VP  VN
```

```
*.opin y
*.ipin a
*.ipin VP
*.ipin VN
XM1 y a VN VN sky130_fd_pr__nfet_01v8 L=0.15 W=1 nf=1 ad='int((nf+1)/2) * W/nf * 0.29' as='int((nf+2)/2) * W/nf * 0.29'
+ pd='2*int((nf+1)/2) * (W/nf + 0.29)' ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W'
+ sa=0 sb=0 sd=0 mult=1 m=1
XM2 y a VP VP sky130_fd_pr__pfet_01v8 L=0.15 W=2 nf=1 ad='int((nf+1)/2) * W/nf * 0.29' as='int((nf+2)/2) * W/nf * 0.29'
+ pd='2*int((nf+1)/2) * (W/nf + 0.29)' ps='2*int((nf+2)/2) * (W/nf + 0.29)' nrd='0.29 / W' nrs='0.29 / W'
+ sa=0 sb=0 sd=0 mult=1 m=1
.ends
```

To run the LVS comparison we use the following executable python script LVS.

```
#!/usr/bin/env python3

import os, sys

if len(sys.argv) != 3:
    print(' LVS: you must specify two netlist filenames to compare!')
    sys.exit(1)

os.system('netgen -batch lvs {} {} \
        ~/share/pdk/sky130A/libs.tech/netgen/sky130A_setup.tcl'.format(sys.argv[1],
sys.argv[2]))

sys.exit(0);
```

Before running LVS we need to prepare the generated netlist `magic_ring.spice` so that the top level circuit is not a subcircuit wrapper. This can be done either by hand:

```
* NGSPICE file created from magic_ring.ext - technology: sky130A
* ... and modified by hand as follows:
* - comment out .subckt for the top level
* - comment out .ends for the top level
* - add .end

* filename: magic_ring.cir

.subckt magic_inv A Y VP VN
X0 Y A VN VN sky130_fd_pr__nfet_01v8 ad=4.5e+11p pd=2.9e+06u as=4.5e+11p ps=2.9e+06u w=1e+06u l=150000u
X1 Y A VP VP sky130_fd_pr__pfet_01v8 ad=9e+11p pd=4.9e+06u as=9e+11p ps=4.9e+06u w=2e+06u l=150000u
.ends

* .subckt magic_ring
Xmagic_inv_0[0] osc_out magic_inv_0[1]/A VP VN magic_inv
Xmagic_inv_0[1] magic_inv_0[1]/A magic_inv_0[2]/A VP VN magic_inv
Xmagic_inv_0[2] magic_inv_0[2]/A magic_inv_0[3]/A VP VN magic_inv
Xmagic_inv_0[3] magic_inv_0[3]/A magic_inv_0[4]/A VP VN magic_inv
Xmagic_inv_0[4] magic_inv_0[4]/A osc_out VP VN magic_inv
* .ends

.end
```

```
LVS magic_ring.cir ring_xschem.spice

...
Circuit 1 contains 10 devices, Circuit 2 contains 10 devices.
Circuit 1 contains 7 nets,     Circuit 2 contains 7 nets.
...
Circuits match correctly.
Result: Circuits match uniquely.
...
```

or it can be done automatically by setting up `ext2spice` with the option `subcircuit top off`

```
% extract all
% ext2spice lvs
% ext2spice subcircuit top off
% ext2spice
```

This way `magic_ring.spice` is already in the desired format to run LVS.

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

.subckt magic_inv A Y VP VN
X0 Y A VN VN sky130_fd_pr__nfet_01v8 ad=4.5e+11p pd=2.9e+06u as=4.5e+11p ps=2.9e+06u w=1e+06u l=150000u
X1 Y A VP VP sky130_fd_pr__pfet_01v8 ad=9e+11p pd=4.9e+06u as=9e+11p ps=4.9e+06u w=2e+06u l=150000u
.ends


* Top level circuit magic_ring

Xmagic_inv_0[0] osc_out magic_inv_0[1]/A VP VN magic_inv
Xmagic_inv_0[1] magic_inv_0[1]/A magic_inv_0[2]/A VP VN magic_inv
Xmagic_inv_0[2] magic_inv_0[2]/A magic_inv_0[3]/A VP VN magic_inv
Xmagic_inv_0[3] magic_inv_0[3]/A magic_inv_0[4]/A VP VN magic_inv
Xmagic_inv_0[4] magic_inv_0[4]/A osc_out VP VN magic_inv
.end
```

```
LVS magic_ring.spice ring_xschem.spice

...
Circuit 1 contains 10 devices, Circuit 2 contains 10 devices.
Circuit 1 contains 7 nets,     Circuit 2 contains 7 nets.
...
Circuits match correctly.
Result: Circuits match uniquely.
...
```

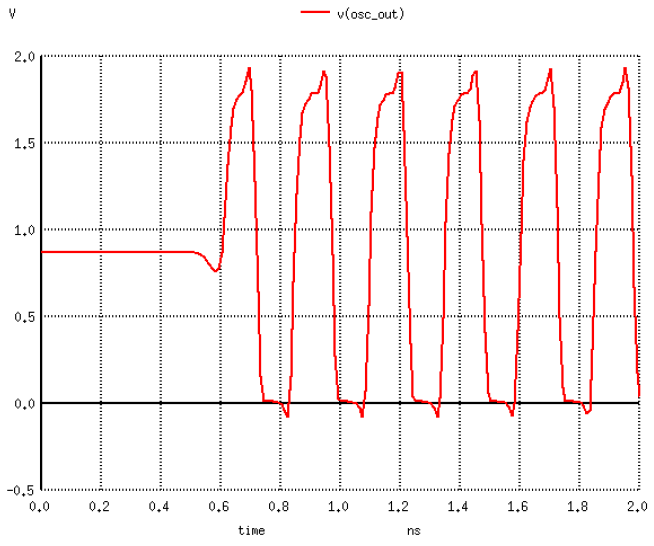
### 3. Simulation of the magic netlist (w/o parasitic extraction)

Create a testbench `tb_magic_ring.sp`:

```
* tb_magic_ring.sp

.lib ~/share/pdk/sky130A/libs.tech/ngspice/sky130.lib.spice tt


.subckt magic_inv A Y VP VN
X0 Y A VN VN sky130_fd_pr__nfet_01v8 ad=4.5e+11p pd=2.9e+06u as=4.5e+11p ps=2.9e+06u w=1e+06u l=150000u
X1 Y A VP VP sky130_fd_pr__pfet_01v8 ad=9e+11p pd=4.9e+06u as=9e+11p ps=4.9e+06u w=2e+06u l=150000u
.ends


* Top level circuit magic_ring

Xmagic_inv_0[0] osc_out magic_inv_0[1]/A VP VN magic_inv
Xmagic_inv_0[1] magic_inv_0[1]/A magic_inv_0[2]/A VP VN magic_inv
Xmagic_inv_0[2] magic_inv_0[2]/A magic_inv_0[3]/A VP VN magic_inv
Xmagic_inv_0[3] magic_inv_0[3]/A magic_inv_0[4]/A VP VN magic_inv
Xmagic_inv_0[4] magic_inv_0[4]/A osc_out VP VN magic_inv

VDD VP 0 DC 1.8
VGND VN 0 DC 0

.tran 10p 2n

.control
run
plot v(osc_out)
.endc
```

Run the simulation:

```
ngspice tb_magic_ring.sp
```



### 4. Useful commands in box tool mode

| command | macro | Description |
|---|---|---|
| `getcell` *cellname* | | import *cellname* as a subcell of the current cell edit |
| `expand` | `x` | expand the view of subcells under the box |
| `unexpand` | `X` | unexpand the view of subcells under the box |
| `expand toggle` | `^X` | toggle between expand and unexpand |
| `select cell` | `i` | select cell under cursor |
| `select more cell` | `I` | select more cell |
| `select less cell` | `^I` | deselect cell under cursor |
| `select clear` | `,` | clear selection |
| `array xsize ysize` | | *xsize* indicates how many elements the array should have in the x-direction, and *ysize* indicates how many elements it should have in the y-direction. The spacing between elements is controlled by the box's width (for the x-direction) and height (for the y-direction) |
| `edit` | `e` | use selected cell as new edit cell |
| `redraw` | `^L` | redraw the display |
| `pushstack` | `>` | load the selected cell |
| `popstack` | `<` | return from editing a cell that was loaded using **pushstack** |
| `dump` *cell* | | Copy contents of the indicated *cell* into the current edit cell |
| `tool` *name* | | Change the layout tool to *name* (where *name* may be one of **box**, **wiring**, **netlist**, or **pick)**. |

### 5. Final Ruminations

Interconnections between layers can also be done using the wiring tool.

Using the box tool, select the box from which you want to start the wire and paint it of the desired material:

```
paint metal1
```

Switch from the box tool to the wiring tool.

```
% tool wiring
```

In wiring tool mode the cursor is an arrow rather than a crosshair as it is in box tool mode.

To select the starting material, left click on it. To draw the wire, drag the mouse. To change the direction of the wiring left click. To switch the wiring layer, use shift-left-click to go up one layer or shift-right-click to go down one layer. Shift-left click and shift-right click automatically add the correct contact between the two layers under the cursor. Once you are done wiring to get out of the dragging mode right-click.

## 6. Useful commands in wiring tool mode

| Mouse button | Description |
|---|---|
| `left` | Pick wire material and size from under the cursor and begin interactive wire placement |
| `right` | Cancel interactive wire placement |
| `middle` | Place a wire at the position shown by the interactive wire tool highlight box, and continue interactive wire placement. |
| `Shift-left` | Change the type of wire to the next plane (e.g., metal1 to metal2) |
| `Shift-right` | Change the type of wire to the previous plane (e.g., metal2 to metal1) |
| `Shift-middle` | Place a contact at the current location and switch to the wire type on the next plane. |
| `Scrollwheel up` | Increase the wire size by one unit |
| `Scrollwheel down` | Decrease the wire size by one unit |

References:


[1] Magic VLSI user's guide (Tim Edwards)
http://opencircuitdesign.com/magic/userguide.html
[2] CMOS Inverter VTC and Transient Simulation Tutorial Using Xschem and Ngspice (Bradley A. Minch, @bminch)
https://www.youtube.com/watch?v=bm3l21ExLOY
[3] Creating a Hierarchical Schematic in Xschem (Bradley A. Minch)
https://www.youtube.com/watch?v=BpPP2hE_eK8
[4] Creating a Hierarchical Layout in Magic Using the SKY130 PDK (Bradley A. Minch)
https://www.youtube.com/watch?v=RPppaGdjbj0
[5] Layout Versus Schematic Tutorial Using Netgen – Part 1 (Bradley A. Minch)
https://www.youtube.com/watch?v=NCaNF4EunYU&t=134s
[6] Layout Versus Schematic Tutorial Using Netgen – Part 2 (Bradley A. Minch)
https://www.youtube.com/watch?v=_xsZbaTBEEA&t=41s
[7] Magic Tutorial #1 – Highlight of Magic Features (e-fabless)
https://www.youtube.com/watch?v=ORw5OaY33A4&t=285s
[8] Magic Tutorial #2 - Using cells, copy, move and wiring (e-fabless)
https://www.youtube.com/watch?v=NUahmUtY814