

Introduction to the SkyWater PDK

The New Age of Open Source Silicon



Tim Edwards
SVP Analog & Platform



efabless
efabless.com



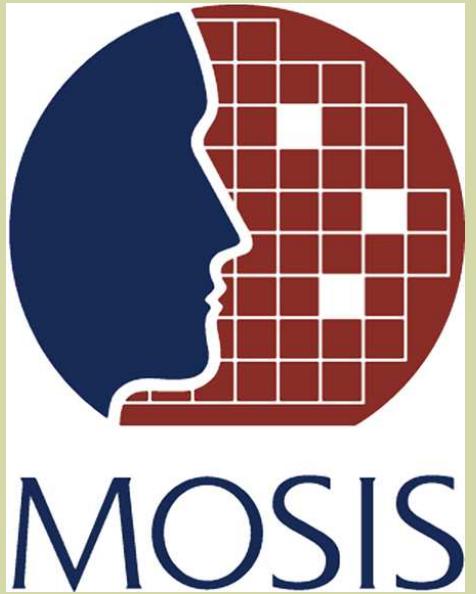
Open Circuit Design
opencircuitdesign.com

October 8, 2021

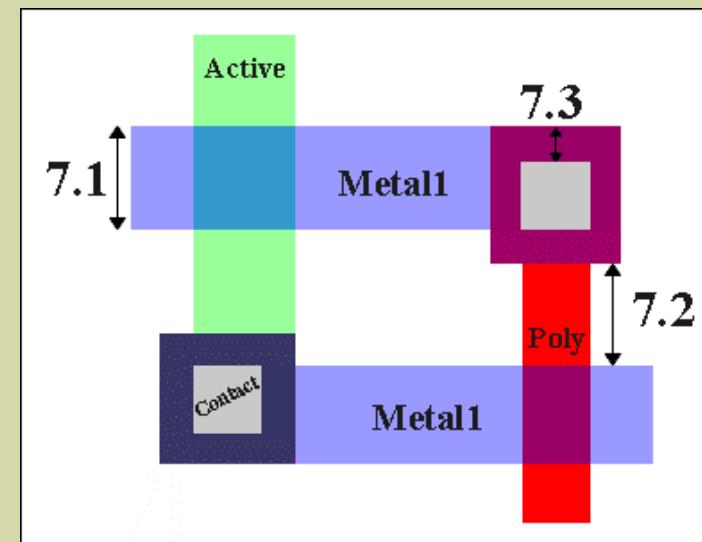
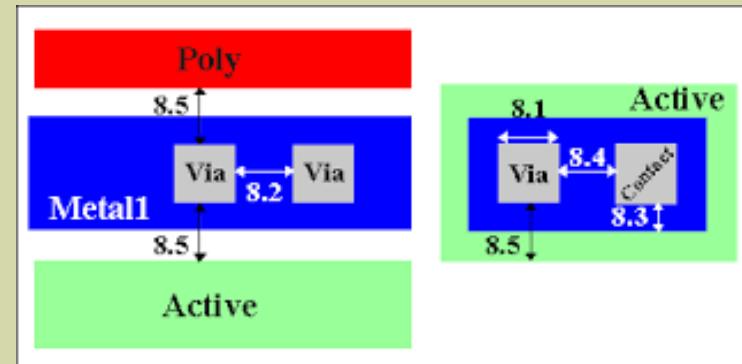
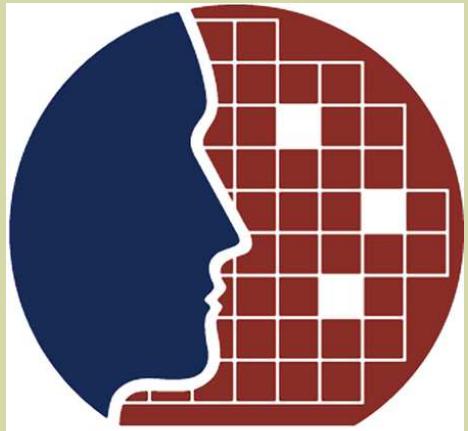
Introduction to the SkyWater PDK



Introduction to the SkyWater PDK



Introduction to the SkyWater PDK



Introduction to the SkyWater PDK



130nm SCMOS

Introduction to the SkyWater PDK



130nm SCMOS

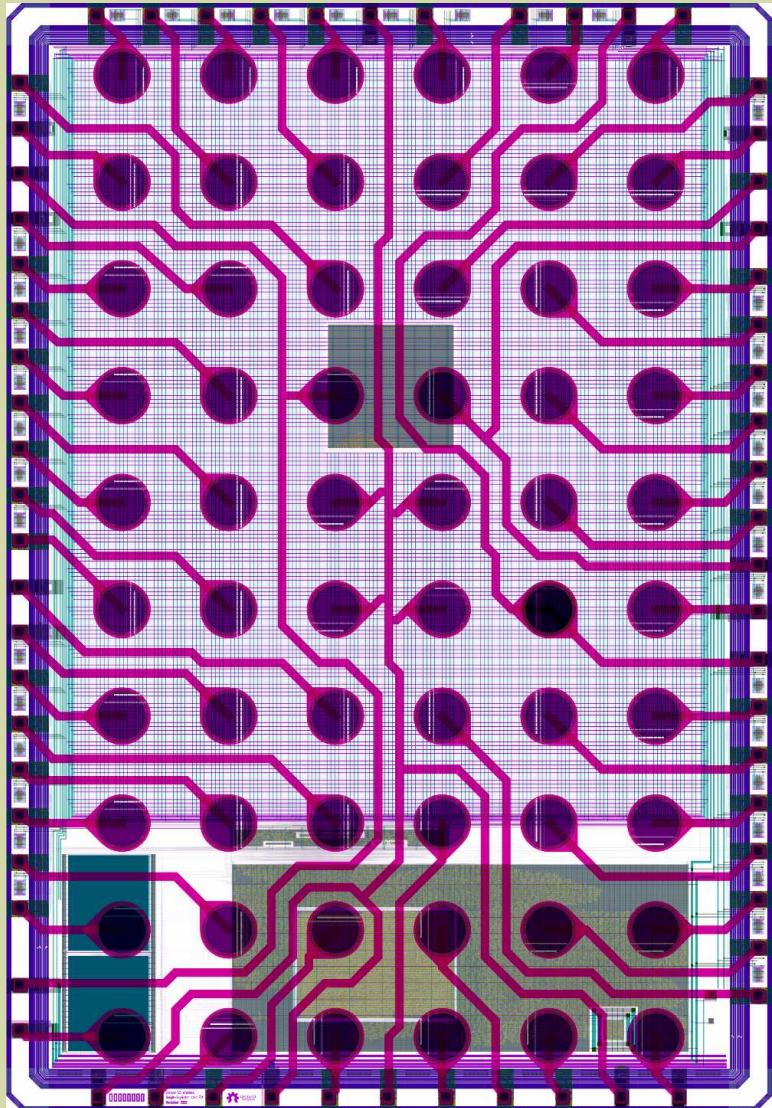
<https://github.com/google/skywater-pdk>

Introduction to the SkyWater PDK



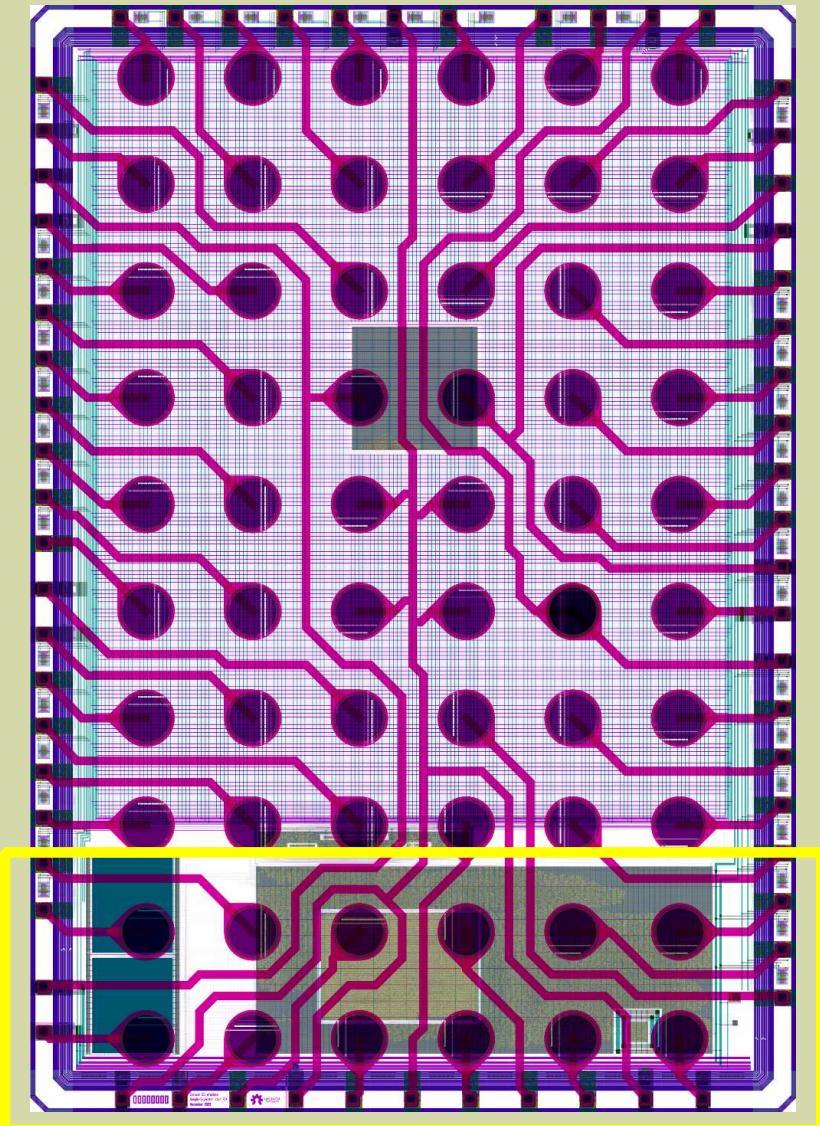
Introduction to the SkyWater PDK

efabless.com



The "Caravel"
harness chip

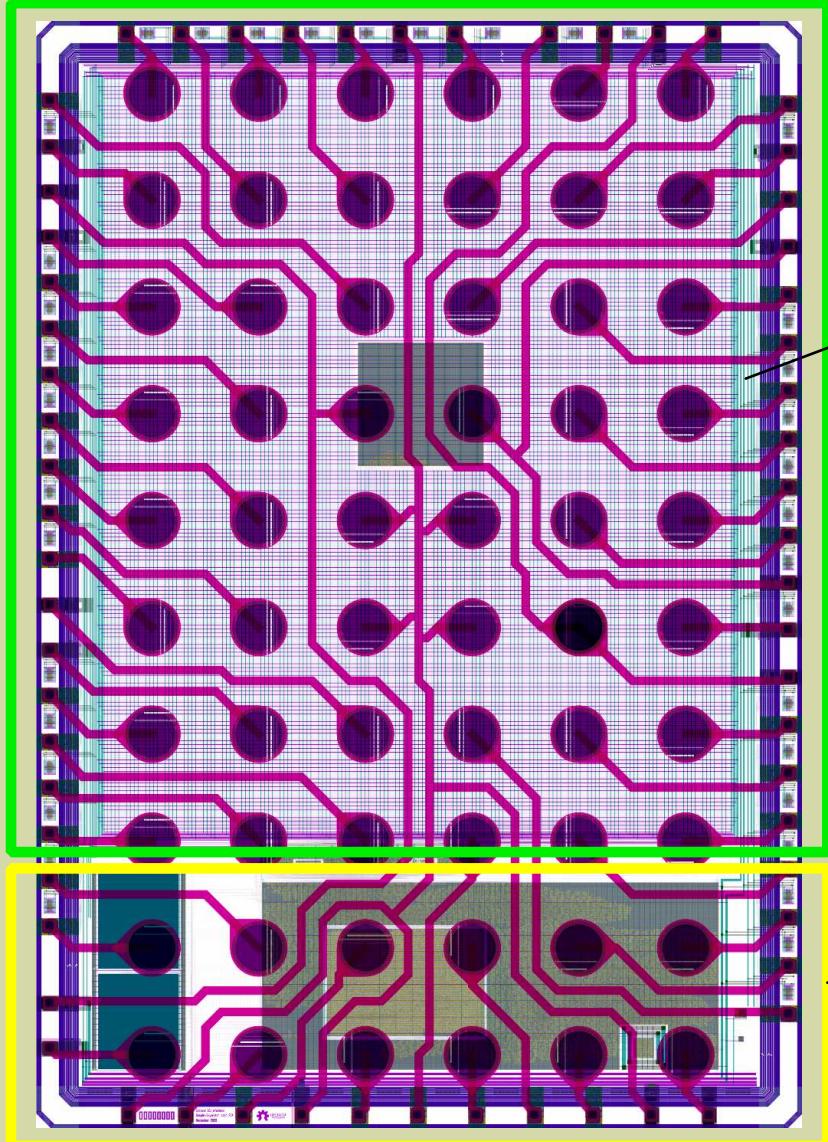
Introduction to the SkyWater PDK



The "Caravel"
harness chip

RISC-V
processor

Introduction to the SkyWater PDK

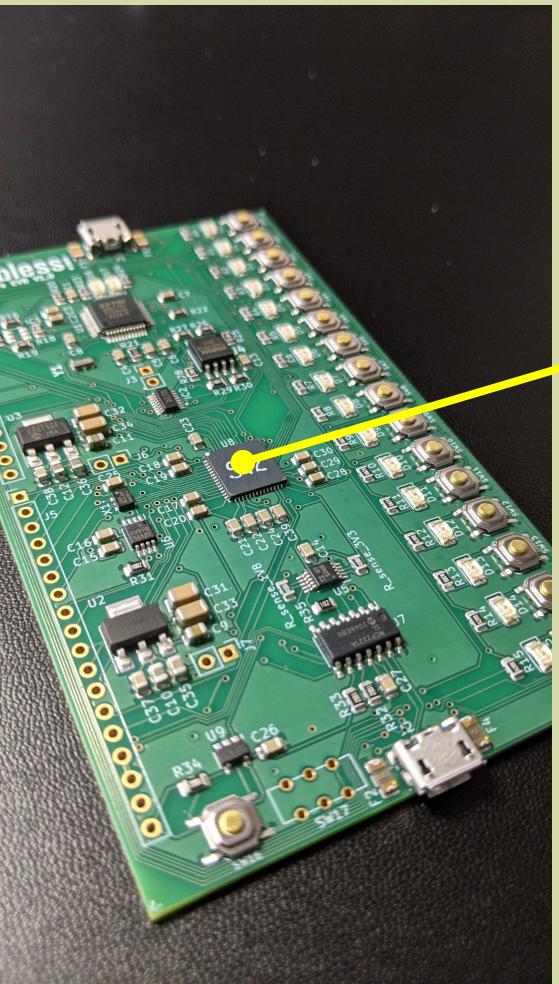


user project area

The "Caravel"
harness chip

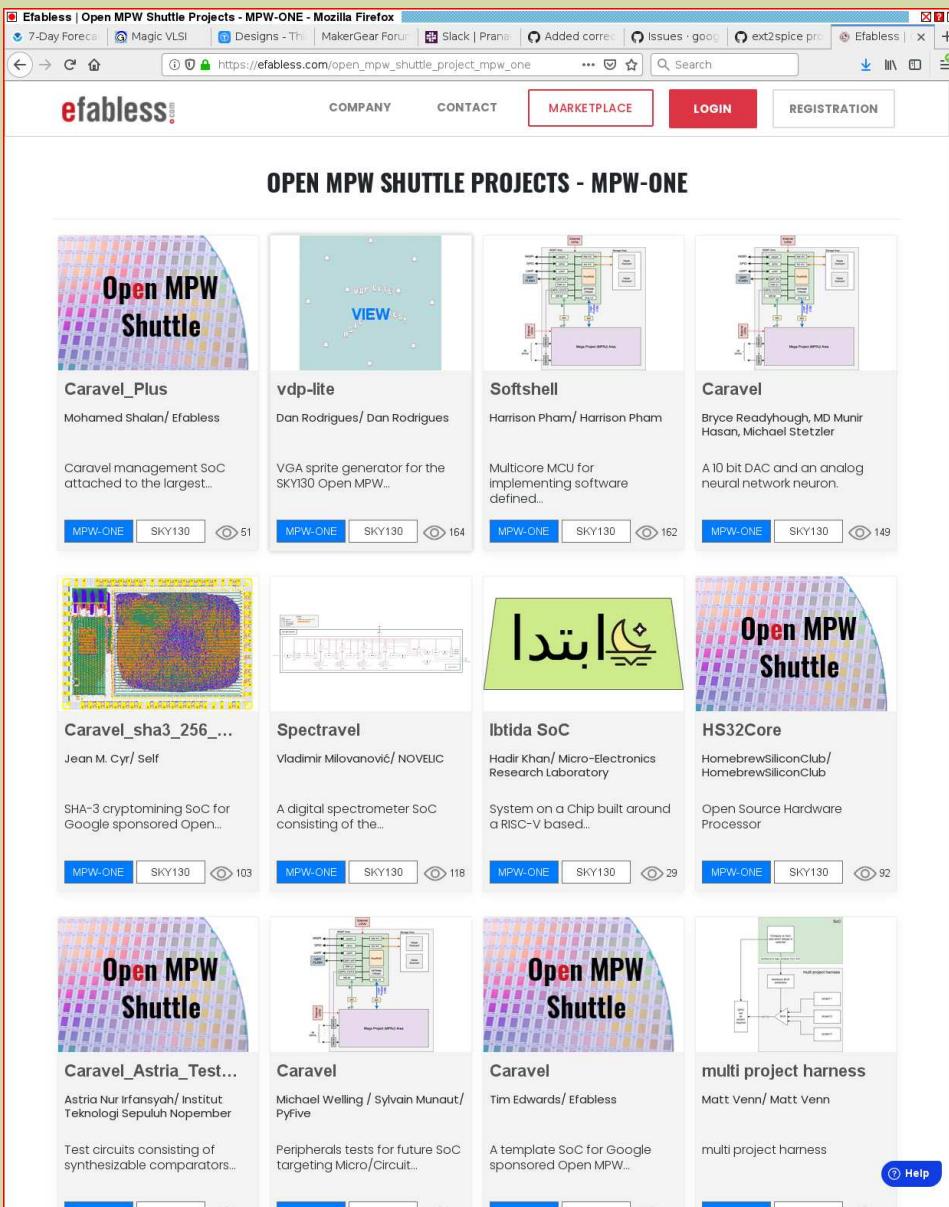
RISC-V
processor

Introduction to the SkyWater PDK



Your project
here!

Introduction to the SkyWater PDK



Projects on Google/SkyWater MPW-One

https://efabless.com/open_mpw_shuttle_project_mpw_one

Introduction to the SkyWater PDK

The screenshot shows a grid of project cards on the Efabless website. Each card displays a thumbnail, the project name, the creator, a brief description, and two buttons for 'MPW-ONE' and 'SKY130'. A red border highlights the 'Open MPW Shuttle' project card.

| Project | Creator | Description | MPW-ONE | SKY130 |
|------------------------|--|---|---------|--------|
| Caravel_Plus | Mohamed Shalan / Efabless | Caravel management SoC attached to the largest... | MPW-ONE | SKY130 |
| vdp-lite | Dan Rodrigues / Dan Rodrigues | VGA sprite generator for the SKY130 Open MPW... | MPW-ONE | SKY130 |
| Softshell | Harrison Pham / Harrison Pham | Multicore MCU for implementing software defined... | MPW-ONE | SKY130 |
| Caravel | Bryce Readyhough, MD Munir Hasan, Michael Stetzler | A 10 bit DAC and an analog neural network neuron... | MPW-ONE | SKY130 |
| Caravel_sha3_256... | Jean M. Cyr / Self | SHA-3 cryptomining SoC for Google sponsored Open... | MPW-ONE | SKY130 |
| Spectravel | Vladimir Milovanović / NOVELIC | A digital spectrometer SoC consisting of the... | MPW-ONE | SKY130 |
| Ibtida SoC | Hadir Khan / Micro-Electronics Research Laboratory | System on a Chip built around a RISC-V based... | MPW-ONE | SKY130 |
| HS32Core | HomebrewSiliconClub / HomebrewSiliconClub | Open Source Hardware Processor | MPW-ONE | SKY130 |
| Caravel_Astria_Test... | Astria Nur Irfansyah / Institut Teknologi Sepuluh Nopember | Test circuits consisting of synthesizable comparators... | MPW-ONE | SKY130 |
| Caravel | Michael Welling / Sylvain Munaut / PyFive | Peripherals tests for future SoC targeting Micro/Circuit... | MPW-ONE | SKY130 |
| Caravel | Tim Edwards / Efabless | A template SoC for Google sponsored Open MPW... | MPW-ONE | SKY130 |
| multi project harness | Matt Venn / Matt Venn | multi project harness | MPW-ONE | SKY130 |

Your project
here!

The SkyWater Open PDK

PDK = "Process Design Kit"

The SkyWater Open PDK

SKY130

The SkyWater Open PDK

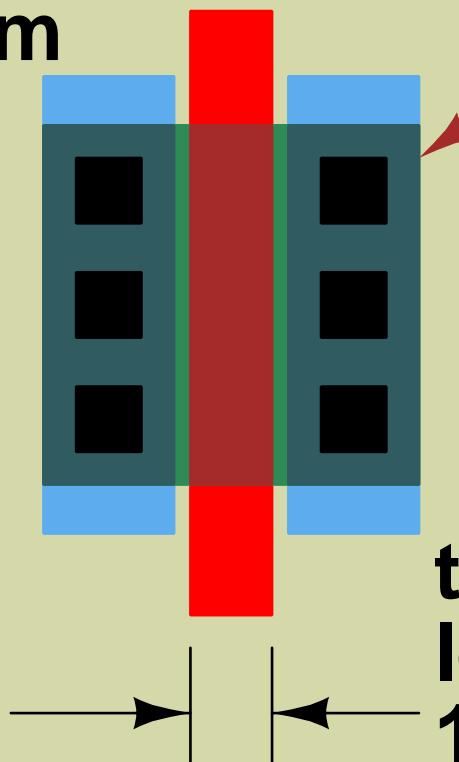
SKY130

130 nm

The SkyWater Open PDK

SKY130

130 nm



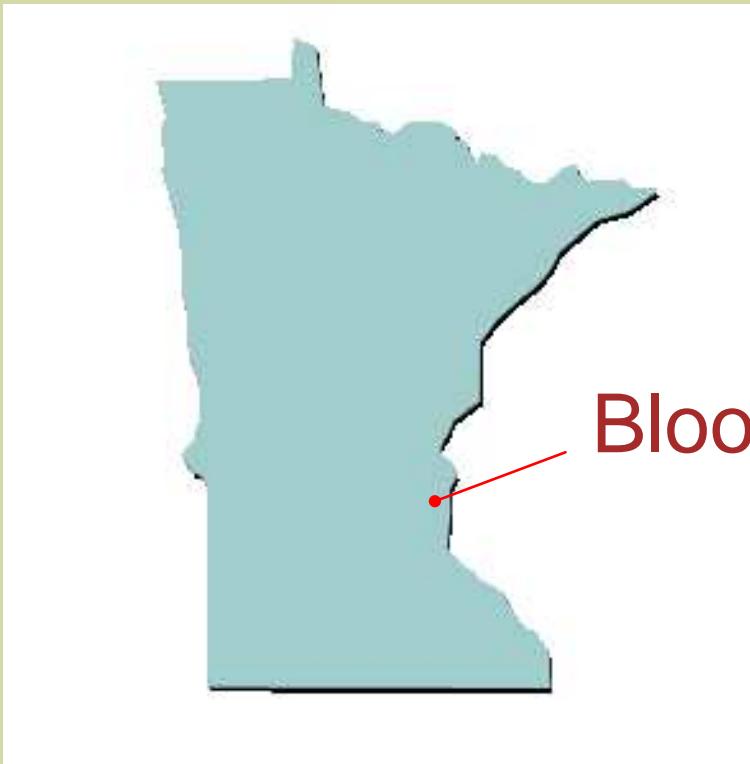
minimum-length transistor^{*}

**transistor
length =
130 nm**

"feature size"

*caveat: for obscure reasons, the minimum size device in the SKY130 process is actually 150 nm. . .

The SkyWater Open PDK



Minnesota

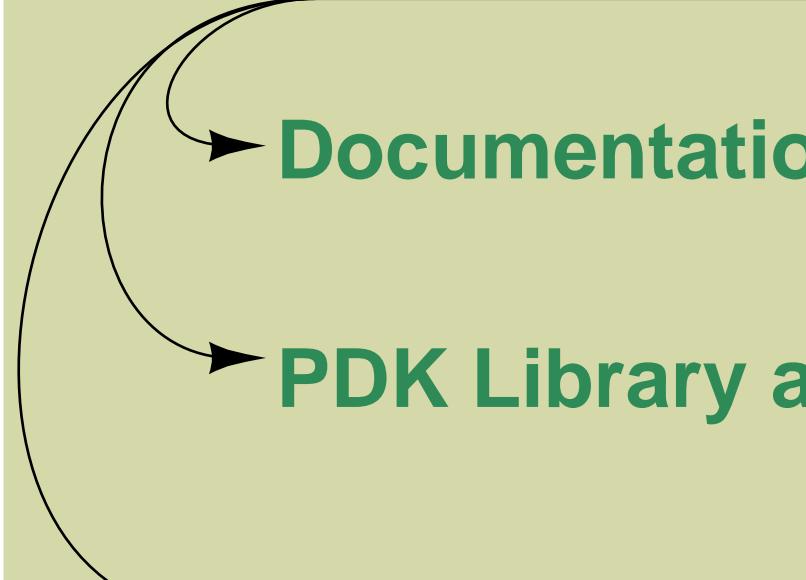
The SkyWater Open PDK

Public repository

- 
- Documentation
 - PDK Library and files

The SkyWater Open PDK

Public repository



Documentation

PDK Library and files

Community

The SkyWater Open PDK

Public repository

- **Documentation**

<https://skywater-pdk--136.org.readthedocs.build>

- **PDK Library and files**

- **Community**

The SkyWater Open PDK

Public repository

- **Documentation**

<https://skywater-pdk--136.org.readthedocs.build>

- **PDK Library and files**

<https://github.com/google/skywater-pdk>

- **Community**

The SkyWater Open PDK

Public repository

- **Documentation**

<https://skywater-pdk--136.org.readthedocs.build>

- **PDK Library and files**

<https://github.com/google/skywater-pdk>

- **Community**



<https://join.skywater.tools>

Open-Source EDA Tools

Open-Source EDA Tools

open_pdks

http://opencircuitdesign.com/open_pdks

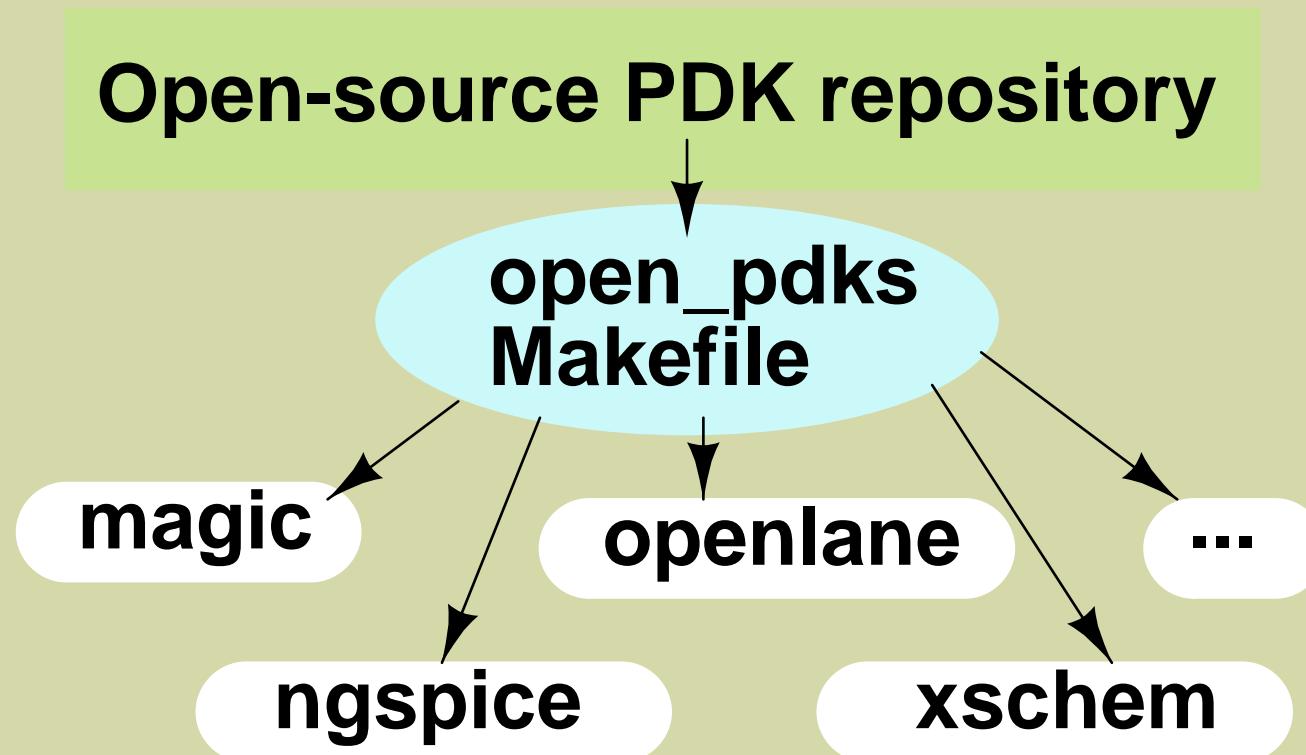
https://github.com/RTimothyEdwards/open_pdks

Open-Source EDA Tools

open_pdks

http://opencircuitdesign.com/open_pdks

https://github.com/RTimothyEdwards/open_pdks



Open-Source EDA Tools

open_pdks

Steps to installing the SKY130 PDK

1. Clone the repository

"**git clone https://github.com/RTimothyEdwards/open_pdks**"

2. Run "**cd open_pdks**"

3. Run "**configure --enable-sky130-pdk**"

4. Run "**make**"

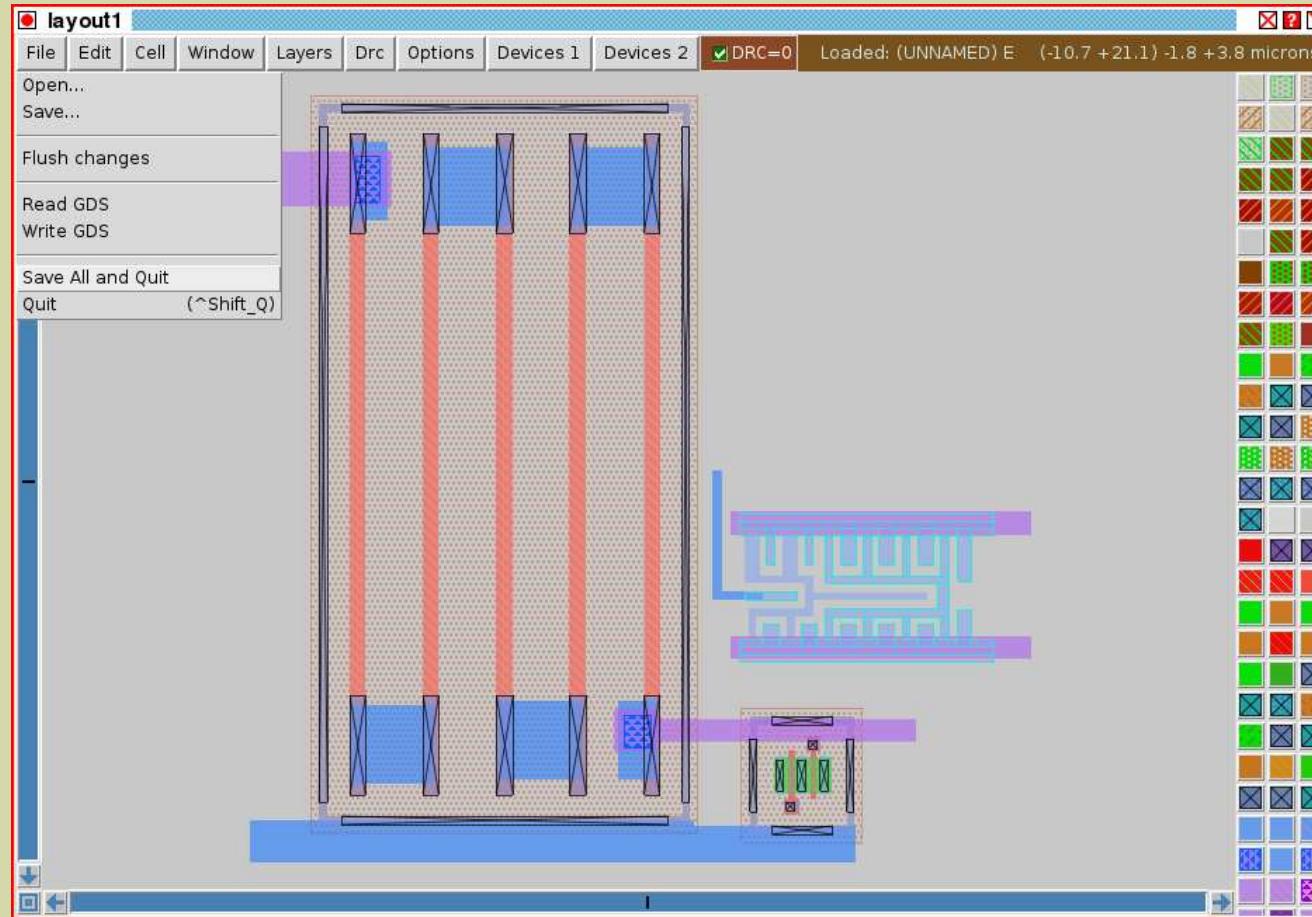
5. Run "**sudo make install**"

Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Magic

<http://opencircuitdesign.com/magic>



Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Klayout

<https://www.klayout.de>

```
# The PCell declaration for the circle
class StarPCell < PCellDeclarationHelper

    include RNA

    def initialize
        # Important: initialize the super class
        super

        # declare the parameters
        param(l, TypeLayer, "layer", default == LayerData::new(0))
        param(r1, TypeDouble, "Inner radius", default == 1.0e-10)
        param(r2, TypeDouble, "Outer radius", default == 5.0e-10)
        param(n, TypeInt, "Number of rays", default == 3)
        param(d, TypeInt, "Ray angle", default == 3, unit == deg)

    end

    def display_text_impl
        # Provide a descriptive text for the cell
        "#starPCell(L#{l.to_s},R1#{r1.to_f},R2#{r2.to_f},N#{n.to_s},D#{d.to_s})"
    end

    def produce_impl
        # This is the main part of the implementation: create the layout
        # compute the ray parts and produce the polygons
        d = Math::PI * d / 180.0
        a = 0.0
        n.times do |i|
            dpts = [
                DPoint::new(r1 * Math.cos(a - d), r1 * Math.sin(a - d)),
                DPoint::new(r1 * Math.cos(a - d), r1 * Math.sin(a - d)),
                DPoint::new(r2 * Math.cos(a - d), r2 * Math.sin(a - d)),
                DPoint::new(r2 * Math.cos(a - d), r2 * Math.sin(a - d))
            ]
            cell.shapes(l_layer).insert(DPolygon.new(dpts))
            a += Math::PI * 2 / n
        end
    end
end
```

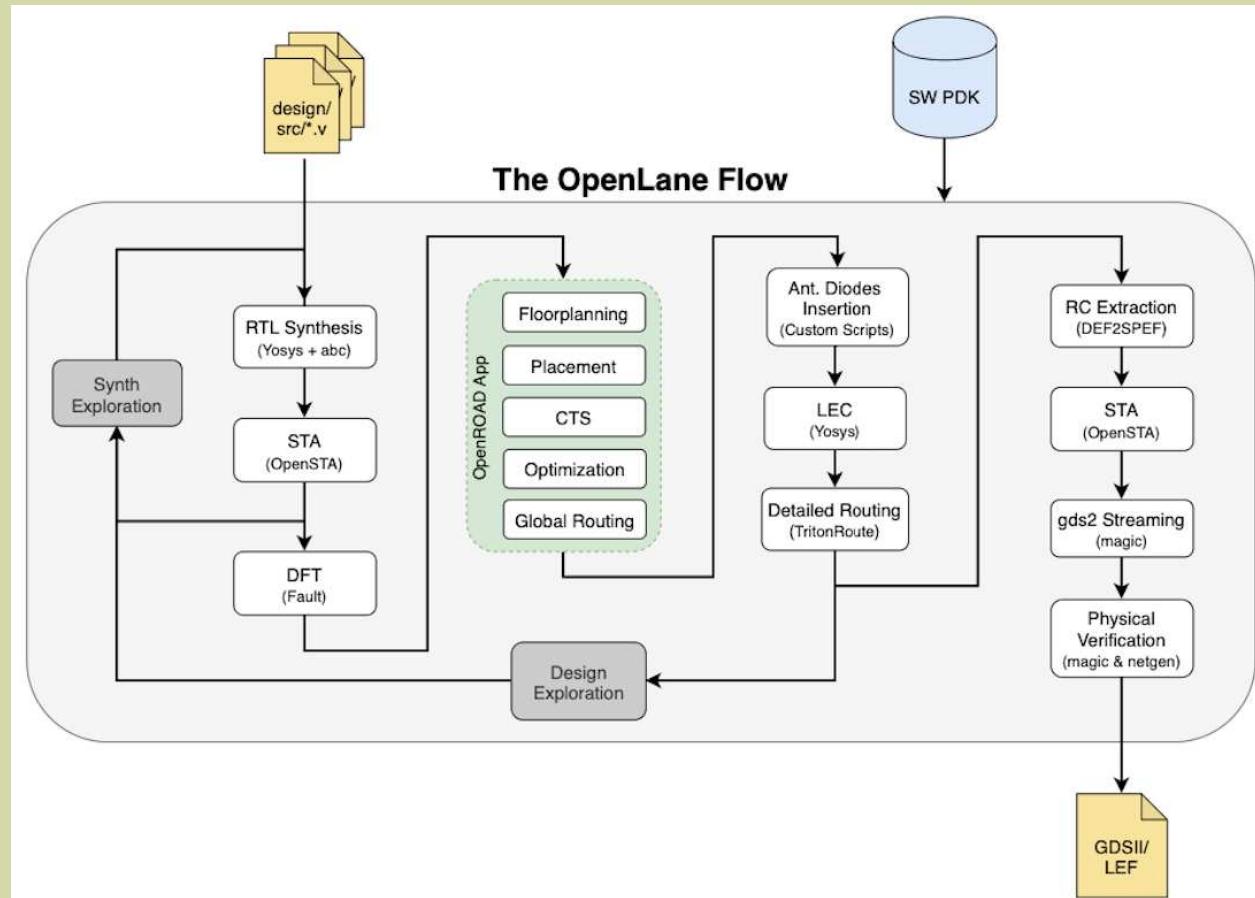


Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Openlane

<https://github.com/The-OpenROAD-Project/OpenLane>

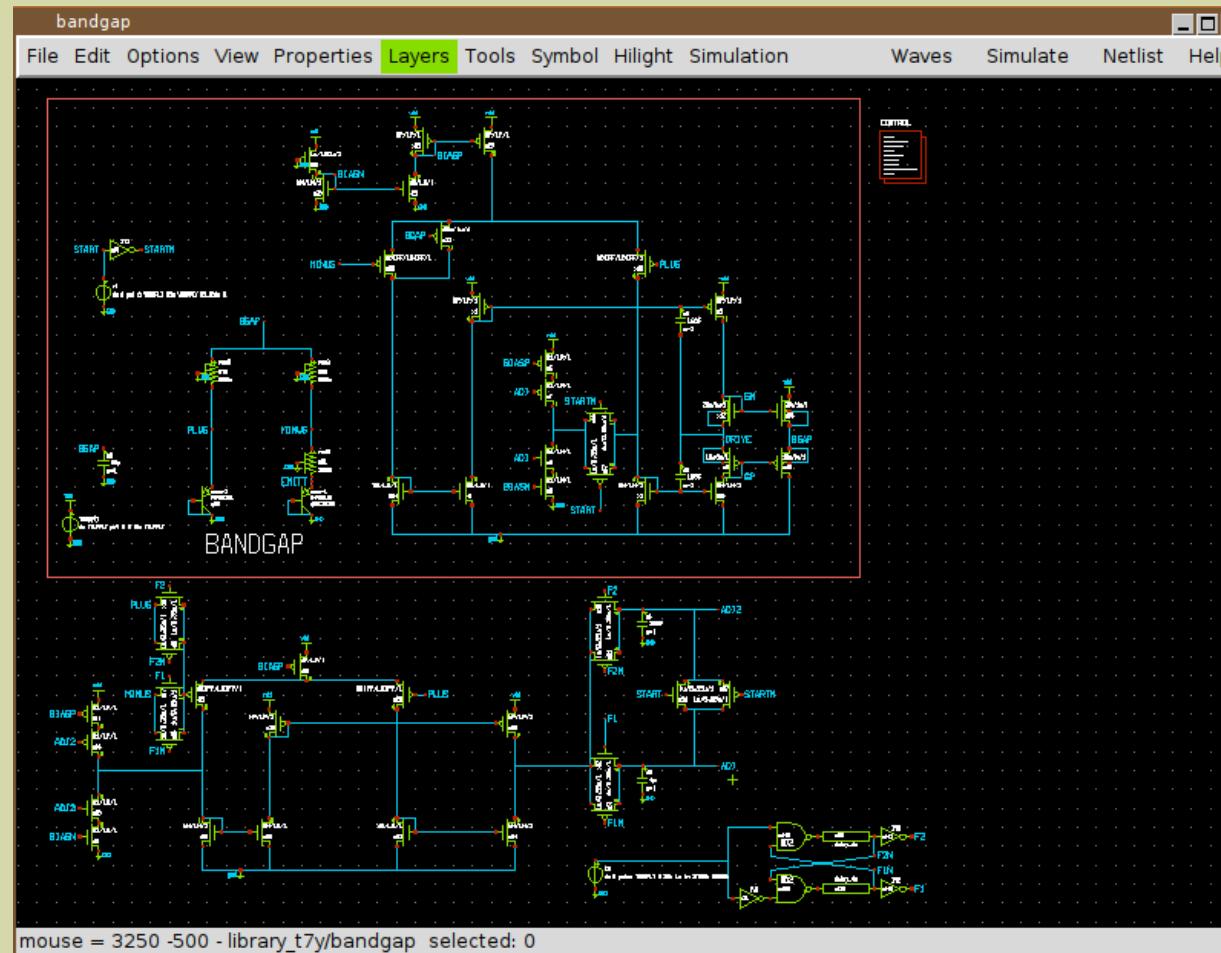


Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Xschem

<https://github.com/StefanSchippers/xschem>

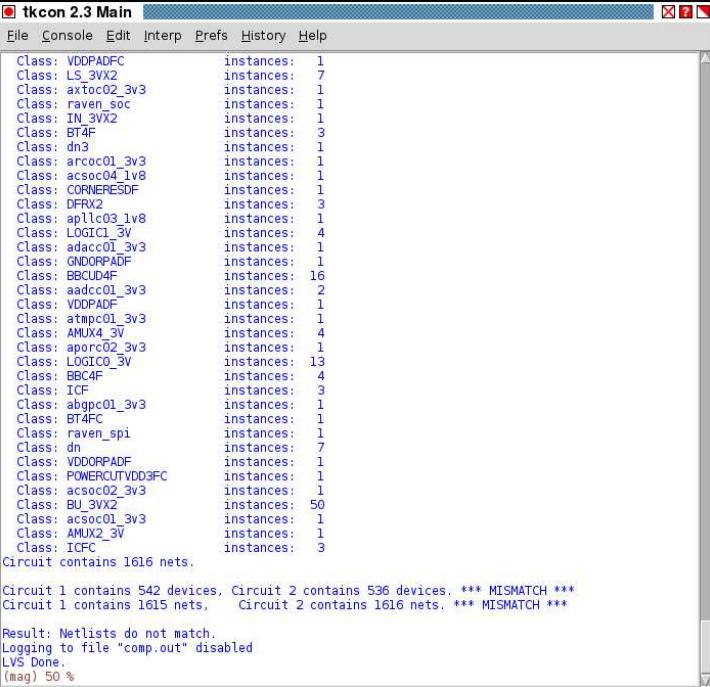


Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Netgen

<http://opencircuitdesign.com/netgen>



tkcon 2.3 Main

```
File Console Edit Interp Prefs History Help
Class: VDDPADFC instances: 1
Class: LS_3VX2 instances: 7
Class: axtoc02_3v3 instances: 1
Class: raven_soc instances: 1
Class: IN_3VX2 instances: 1
Class: BT4F instances: 3
Class: dn3 instances: 1
Class: arcoc01_3v3 instances: 1
Class: acsoc04_1v8 instances: 1
Class: CORNERESDF instances: 1
Class: DFRX2 instances: 3
Class: apilc03_1v8 instances: 1
Class: LOGIC1_3V instances: 4
Class: adacc01_3v3 instances: 1
Class: GNDORPADF instances: 1
Class: BBCUD4F instances: 16
Class: aadcc01_3v3 instances: 2
Class: VDDPADF instances: 1
Class: atmpc01_3v3 instances: 1
Class: AMUX4_3V instances: 4
Class: aporc02_3v3 instances: 1
Class: LOGIC0_3V instances: 13
Class: BBC4F instances: 4
Class: ICF instances: 3
Class: abgpc01_3v3 instances: 1
Class: BT4FC instances: 1
Class: raven_spi instances: 1
Class: dn instances: 7
Class: VDDORPADF instances: 1
Class: POWERCUTVDD3FC instances: 1
Class: acsoc02_3v3 instances: 1
Class: BU_3VX2 instances: 50
Class: acsoc01_3v3 instances: 1
Class: AMUX2_3V instances: 1
Class: ICFC instances: 3
Circuit contains 1616 nets.
Circuit 1 contains 542 devices, Circuit 2 contains 536 devices. *** MISMATCH ***
Circuit 1 contains 1615 nets, Circuit 2 contains 1616 nets. *** MISMATCH ***
Result: Netlists do not match.
Logging to file "comp.out" disabled
LVS Done.
(mag) 50 %
```

mrxvt

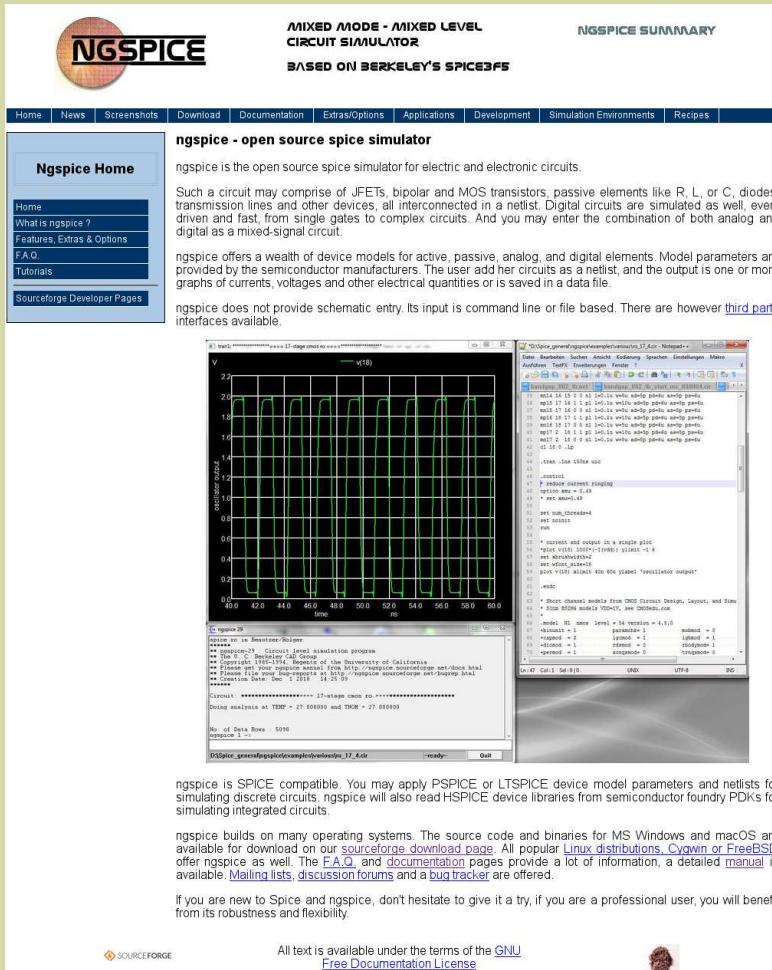
```
BT4FC (1)
aadcc01_3v3 (2)
aopac01_3v3 (1)
acsoc02_3v3 (1)
LOGIC1_3V (4)
XSPRAMBLP_1024X32_M8P (1)
adacc01_3v3 (1)
raven_soc (1)
LOGIC0_3V (13)
DFRX2 (3)
IN_3VX2 (1)
abgpc01_3v3 (1)
dn3 (1)
acsoc01_3v3 (1)
aporc02_3v3 (1)
AMUX4_3V (4)
acmpc01_3v3 (1)
arcoc01_3v3 (1)
Number of devices: 542 **Mismatch**
Number of nets: 1615 **Mismatch**
NET mismatches: Class fragments follow (with fanout counts):
Circuit 1: raven
Net: VDD3V3
raven_spi/vdd3 = 1
cmm5t/1 = 160
LS_3VX2/VDD3V3 = 6
|BT4FC (1)
|aadcc01_3v3 (2)
|aopac01_3v3 (1)
|acsoc02_3v3 (1)
|LOGIC1_3V (4)
|XSPRAMBLP_1024X32_M8P (1)
|adacc01_3v3 (1)
|raven_soc (1)
|LOGIC0_3V (13)
|DFRX2 (3)
|IN_3VX2 (1)
|abgpc01_3v3 (1)
|dn3 (1)
|acsoc01_3v3 (1)
|aporc02_3v3 (1)
|AMUX4_3V (4)
|acmpc01_3v3 (1)
|arcoc01_3v3 (1)
Number of devices: 536 **Mismatch**
Number of nets: 1616 **Mismatch**
Circuit 2: raven
Net: overtemp_ena
|raven_soc/overtemp_ena = 1
|LS_3VX2/A = 1
```

Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

Ngspice

<https://ngspice.sourceforge.net>



Open-Source EDA Tools

Tools Currently Supported by Open_PDKs

iverilog

<https://iverilog.icarus.com>

qflow

<http://opencircuitdesign.com/qflow>

IRSIM

<http://opencircuitdesign.com/irsim>

xcircuit

<http://opencircuitdesign.com/xcircuit>

xyce

<https://xyce.sandia.gov>

fun fact: These slides were drawn with xcircuit!

Open-Source EDA Tools

SkyWater SKY130 Libraries

1. Digital standard cells

`sky130_fd_sc_hd` `sky130_fd_sc_hdll`
`sky130_fd_sc_hs` `sky130_fd_sc_ms` `sky130_fd_sc_ls`
`sky130_fd_sc_lp` `sky130_fd_sc_hvl`

2. Primitive devices / analog

`sky130_fd_pr`

3. I/O cells

`sky130_fd_io`

4. 3rd-party libraries

`sky130_ml_xx_hd`
`sky130_sram_macros`

Open-Source EDA Tools

PDK (e.g., SKY130) Installed Filesystem Structure

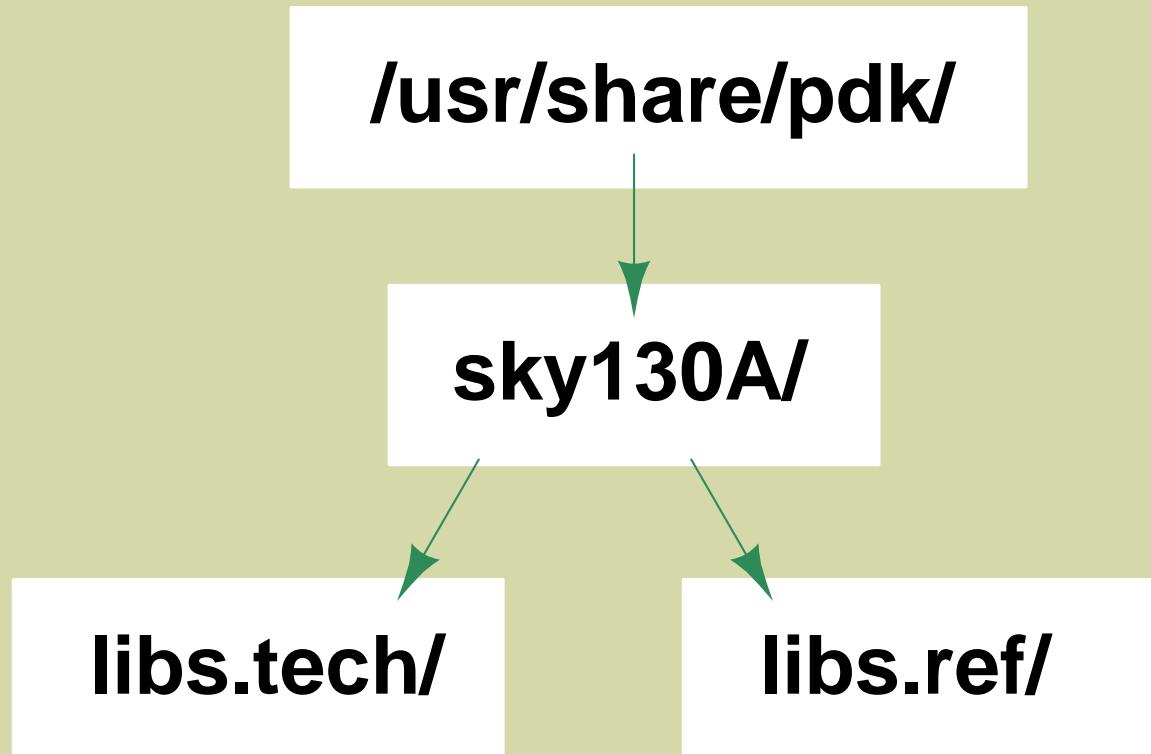
`/usr/share/pdk/`



`sky130A/`

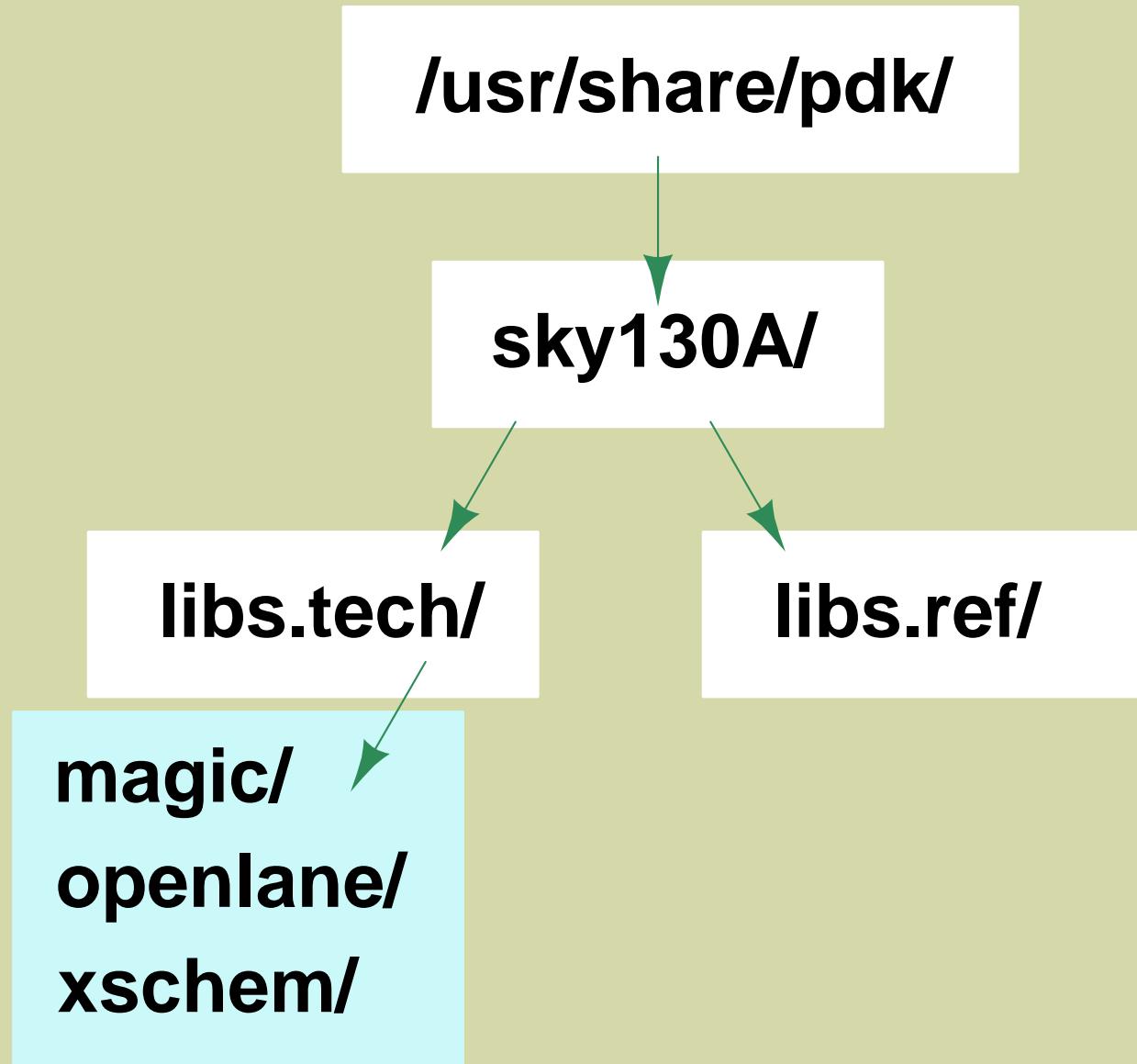
Open-Source EDA Tools

SkyWater SKY130 Installed Filesystem Structure



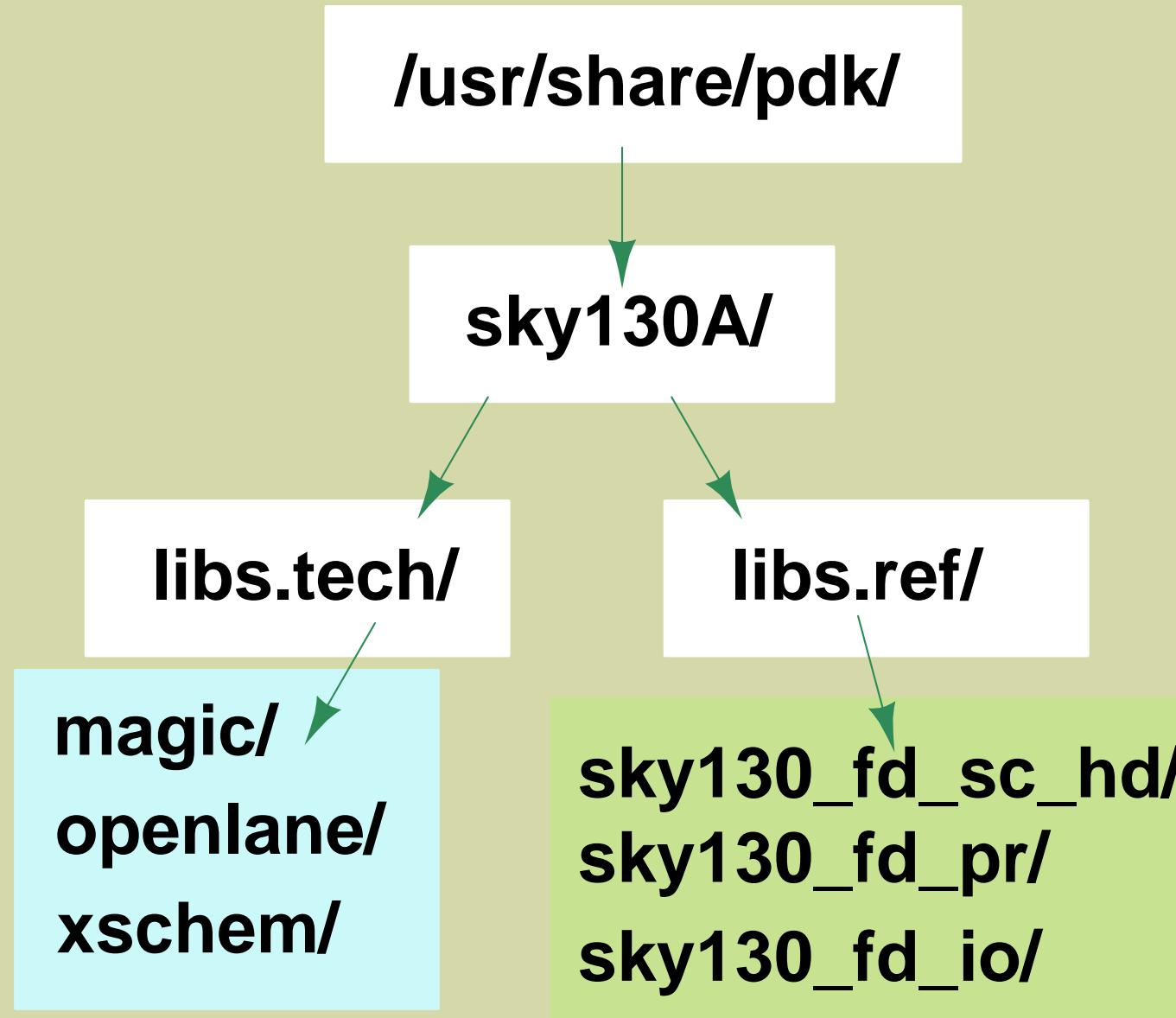
Open-Source EDA Tools

SkyWater SKY130 Installed Filesystem Structure



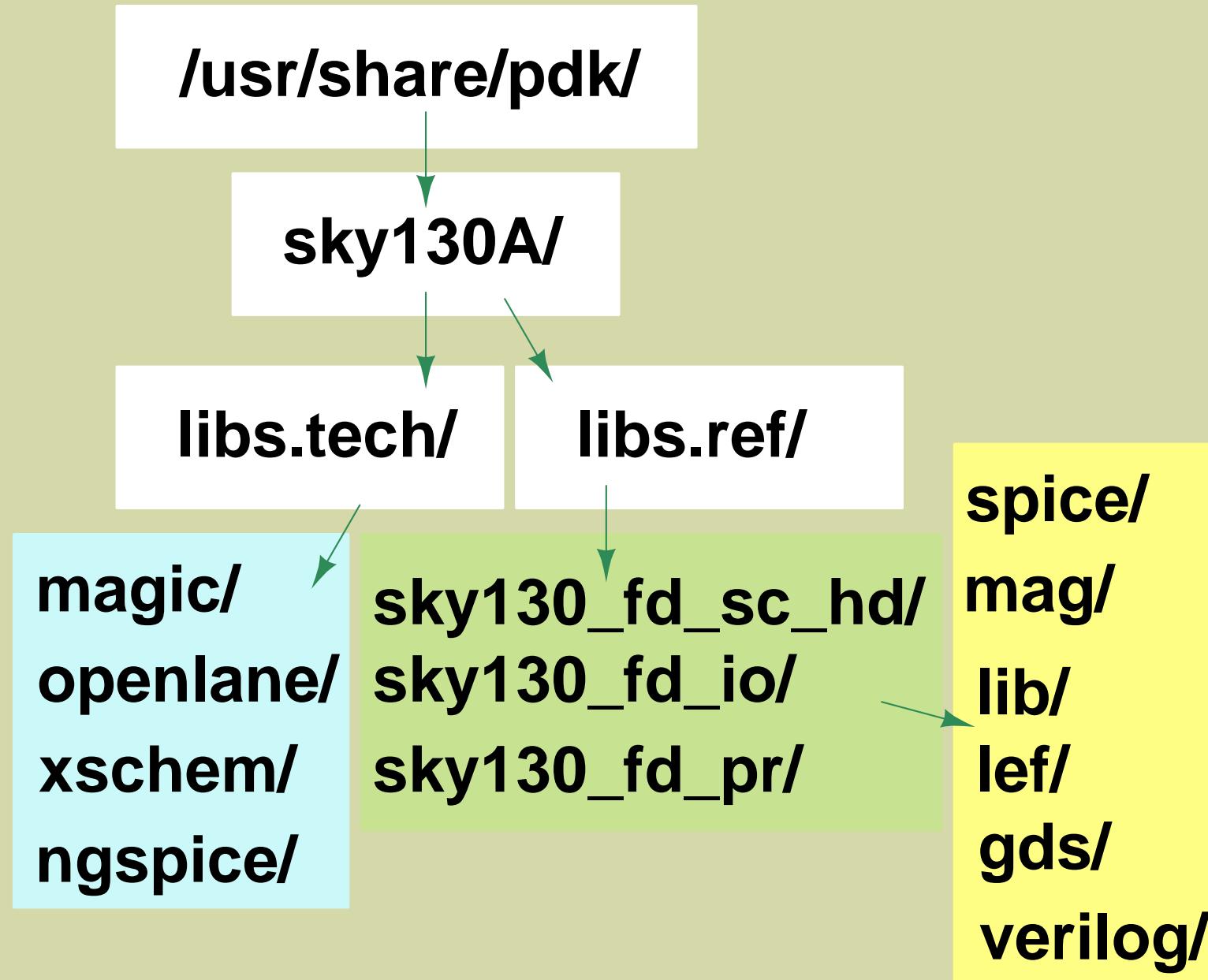
Open-Source EDA Tools

SkyWater SKY130 Installed Filesystem Structure



Open-Source EDA Tools

SkyWater SKY130 Installed Filesystem Structure



Open-Source EDA Tools

Open PDKs Project Filesystem Structure

project_root/

e.g., "my_sky130_project"

Open-Source EDA Tools

Open PDKs Project Filesystem Structure

project_root/

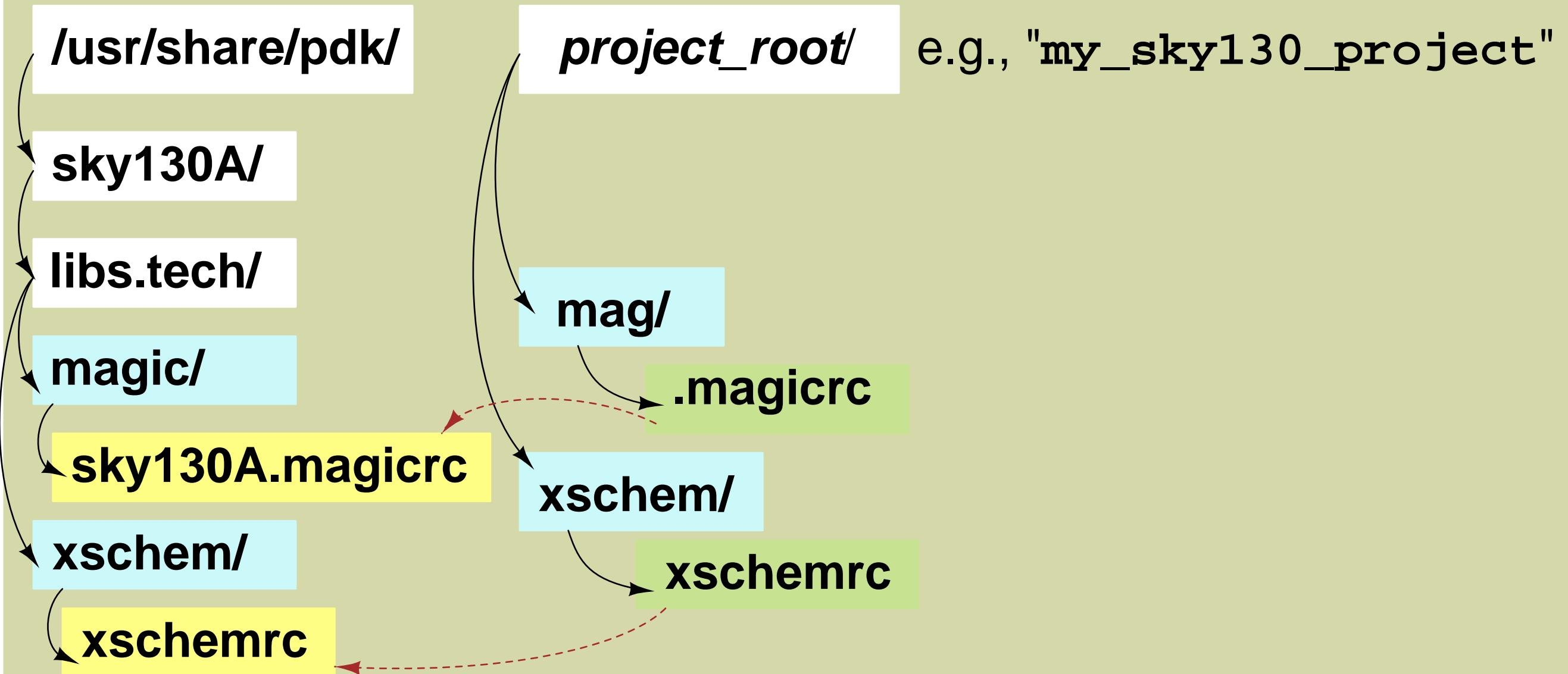
e.g., "my_sky130_project"



**xschem/
spice/
mag/
openlane/
verilog/**

Open-Source EDA Tools

Open PDKs Project Filesystem Structure



Open-Source EDA Tools

Open PDKs Project Filesystem Structure

Project Management

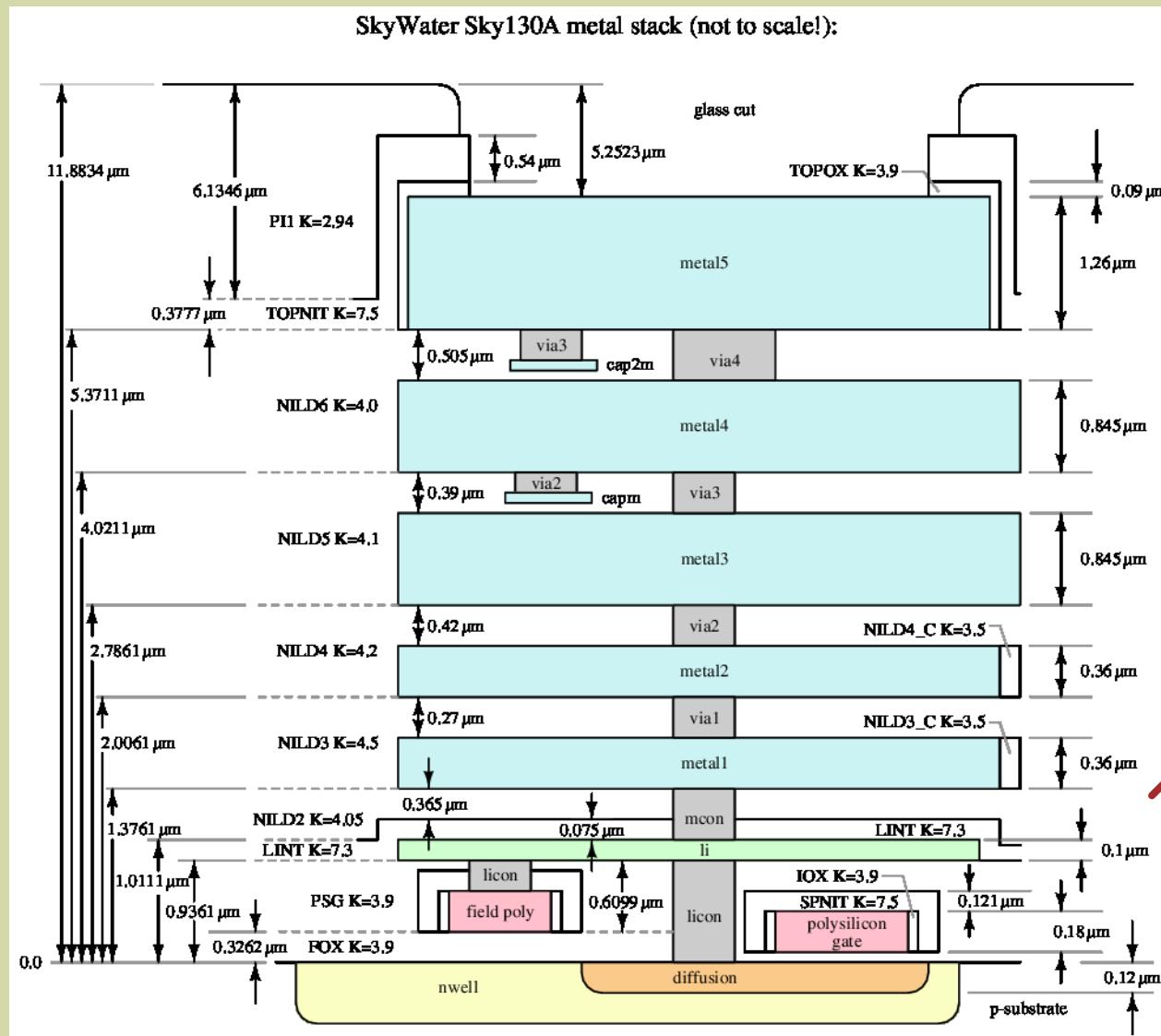
/usr/share/pdk/scripts/project_manager.py

(work in progress)

Understanding the SkyWater PDK Layers

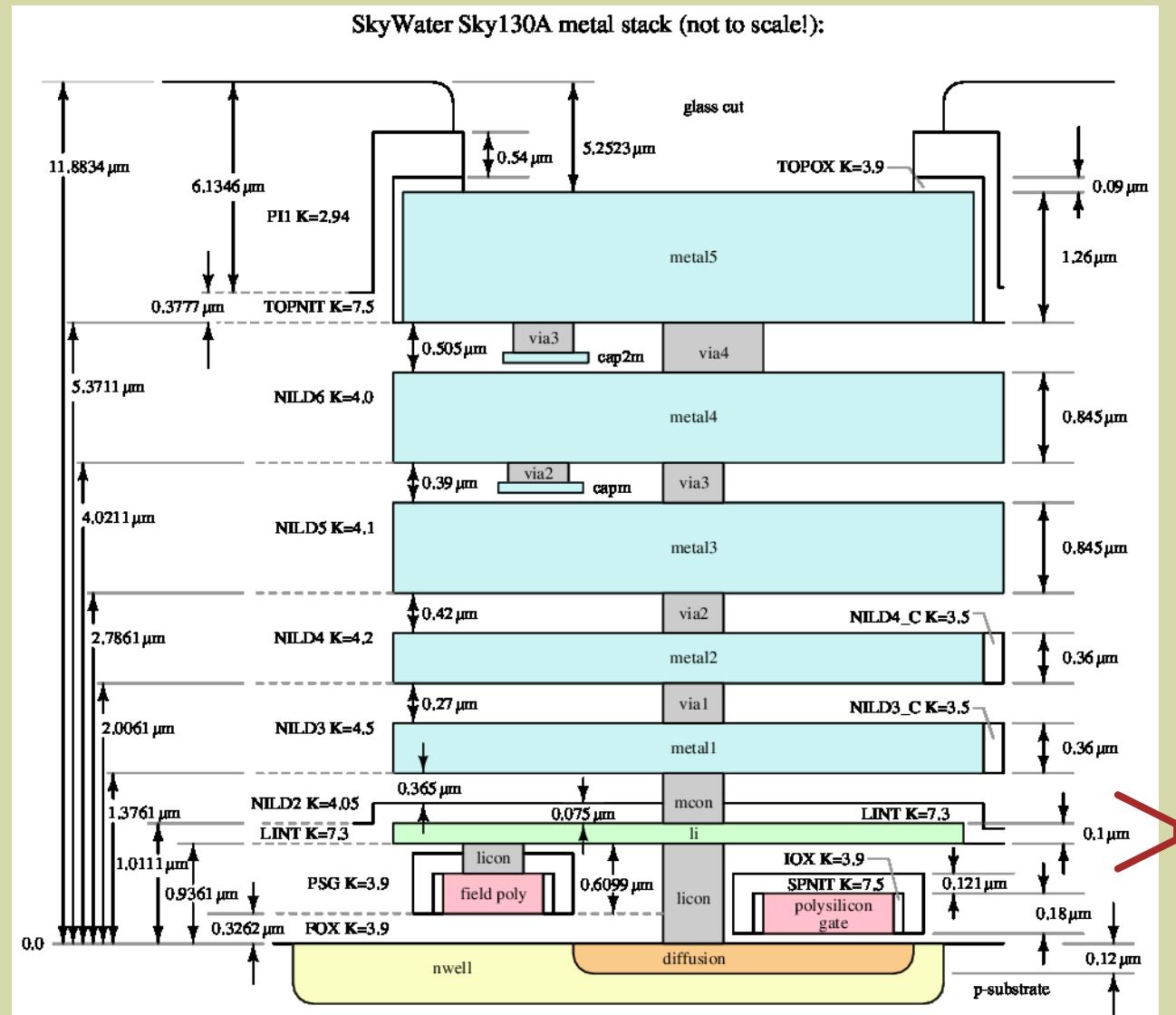
SKY130

Understanding the SkyWater PDK Layers



5 layers of aluminum metal

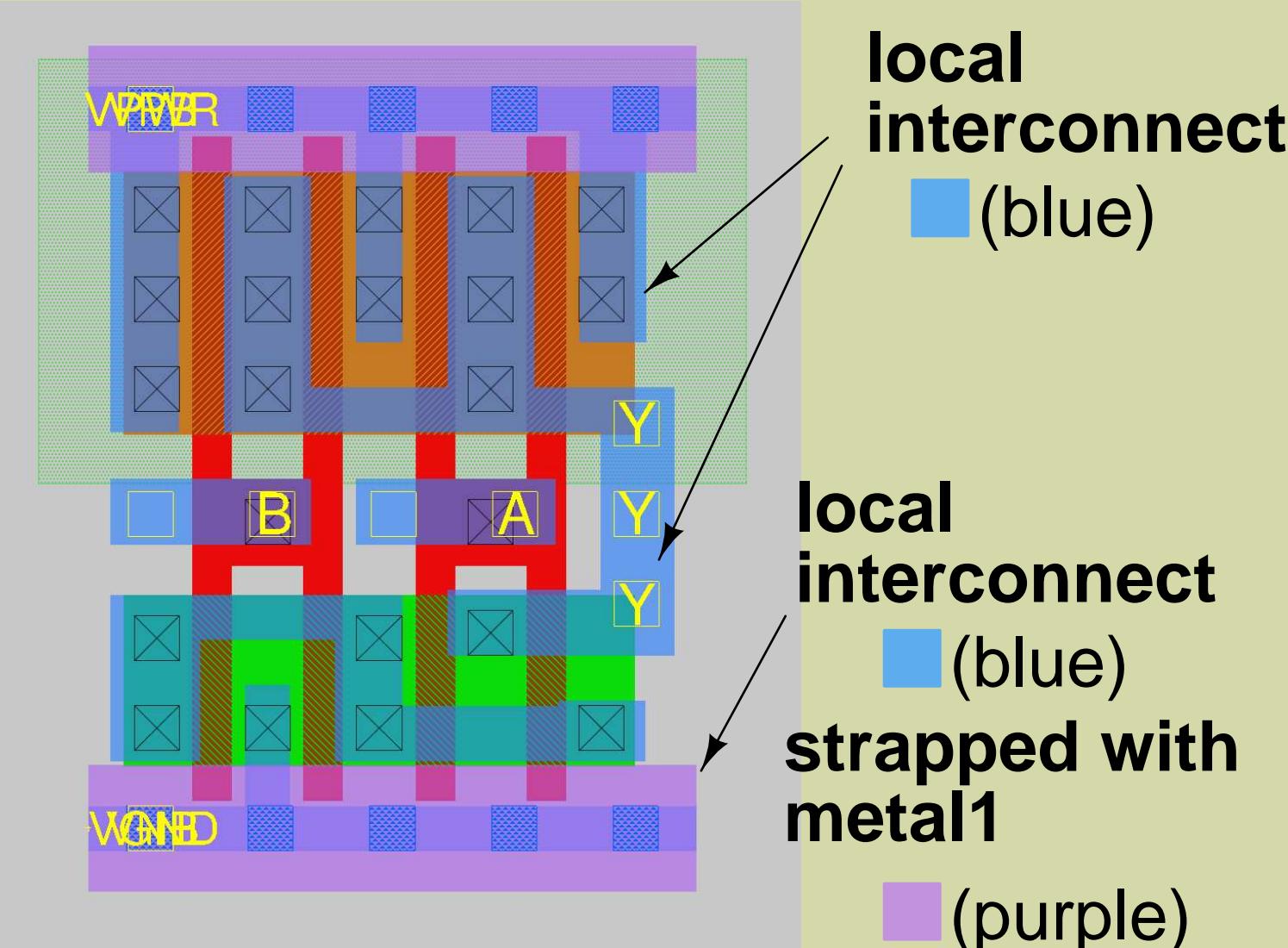
Understanding the SkyWater PDK Layers



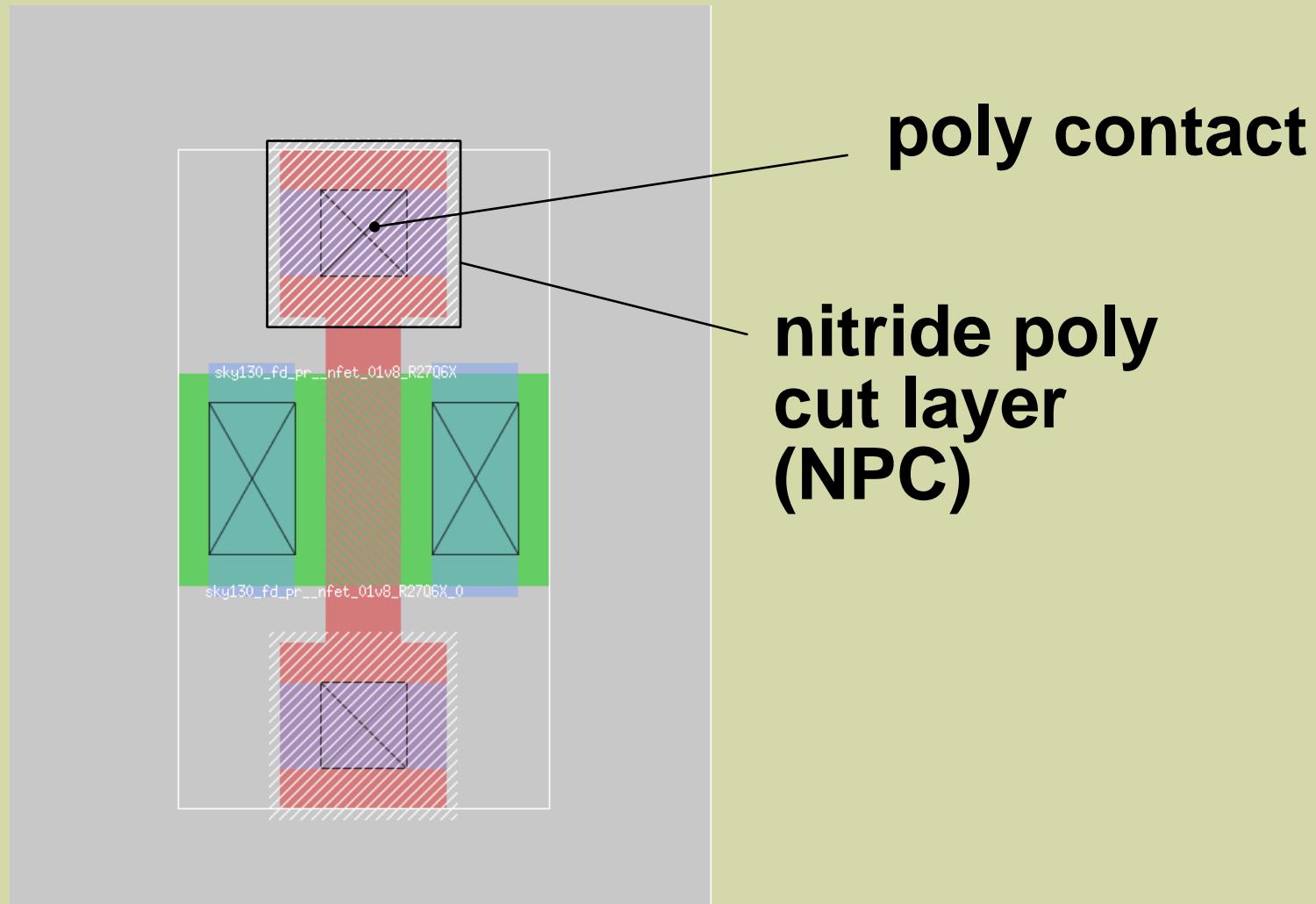
Titanium Nitride (TiN)
aka "Local interconnect"

Understanding the SkyWater PDK

sky130_fd_sc_hd__nand2_2

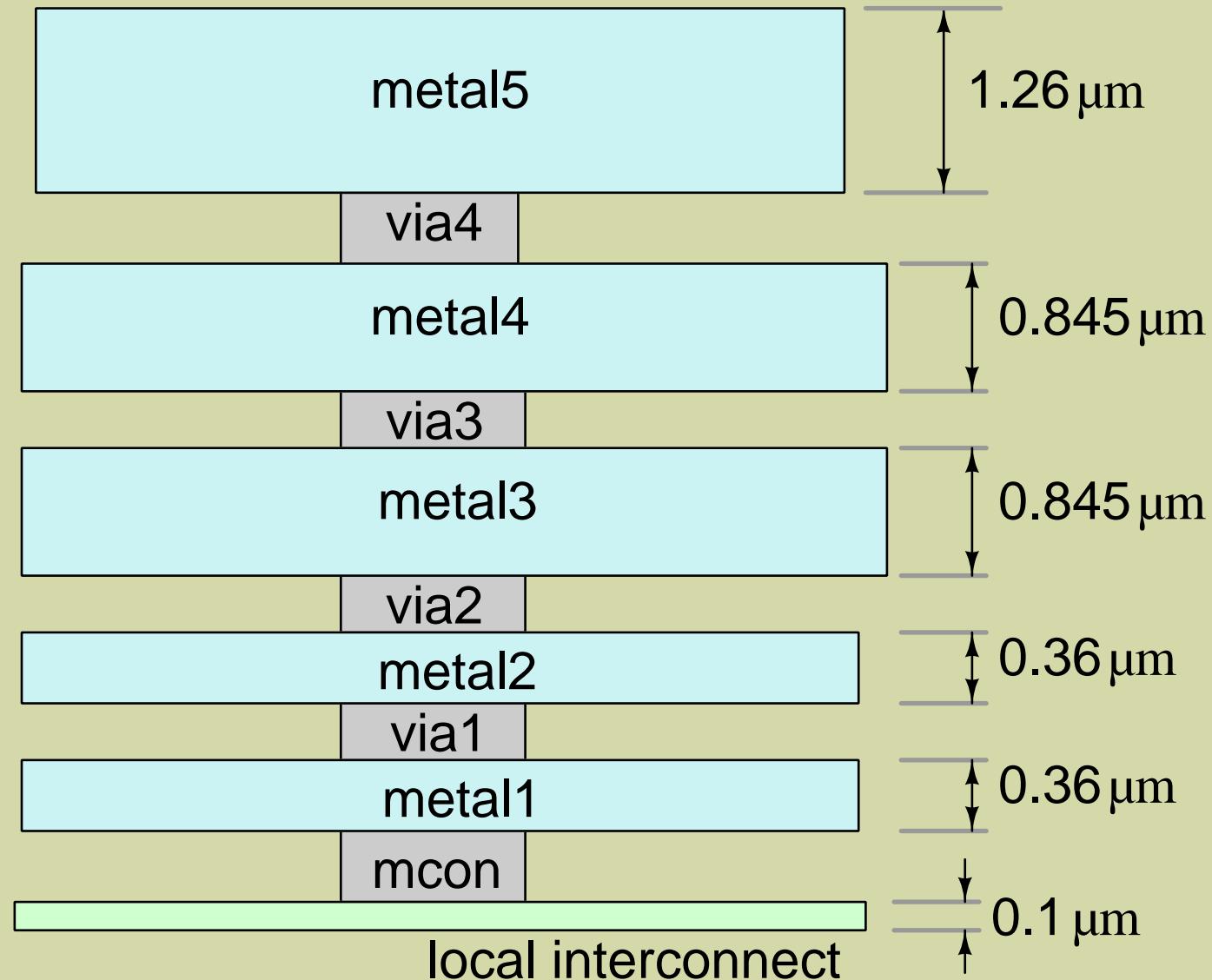


Understanding the SkyWater PDK Layers



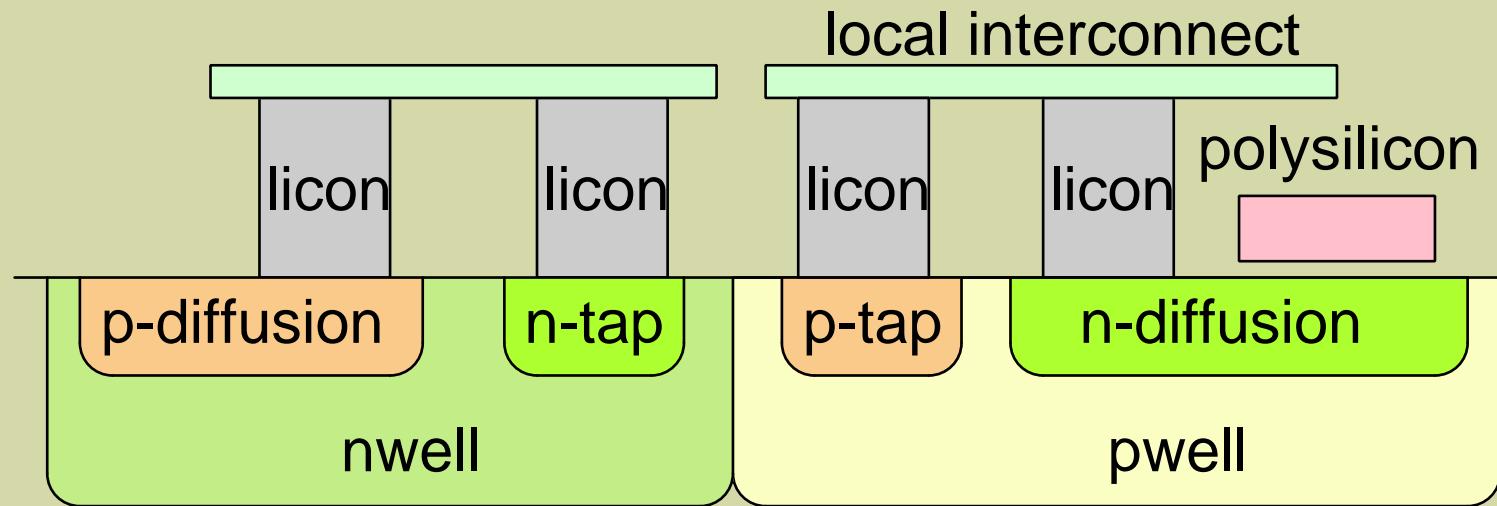
Understanding the SkyWater PDK

back-end metal stack



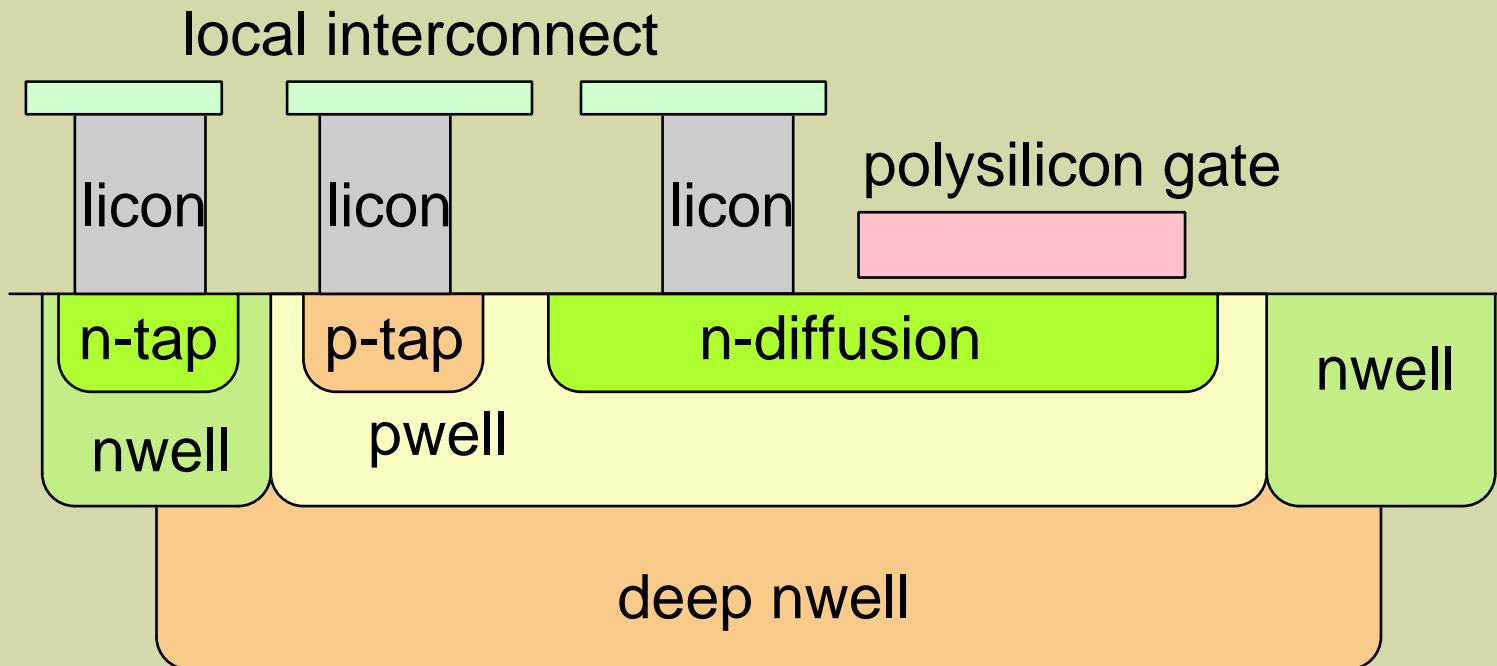
Understanding the SkyWater PDK

front-end layers



Understanding the SkyWater PDK

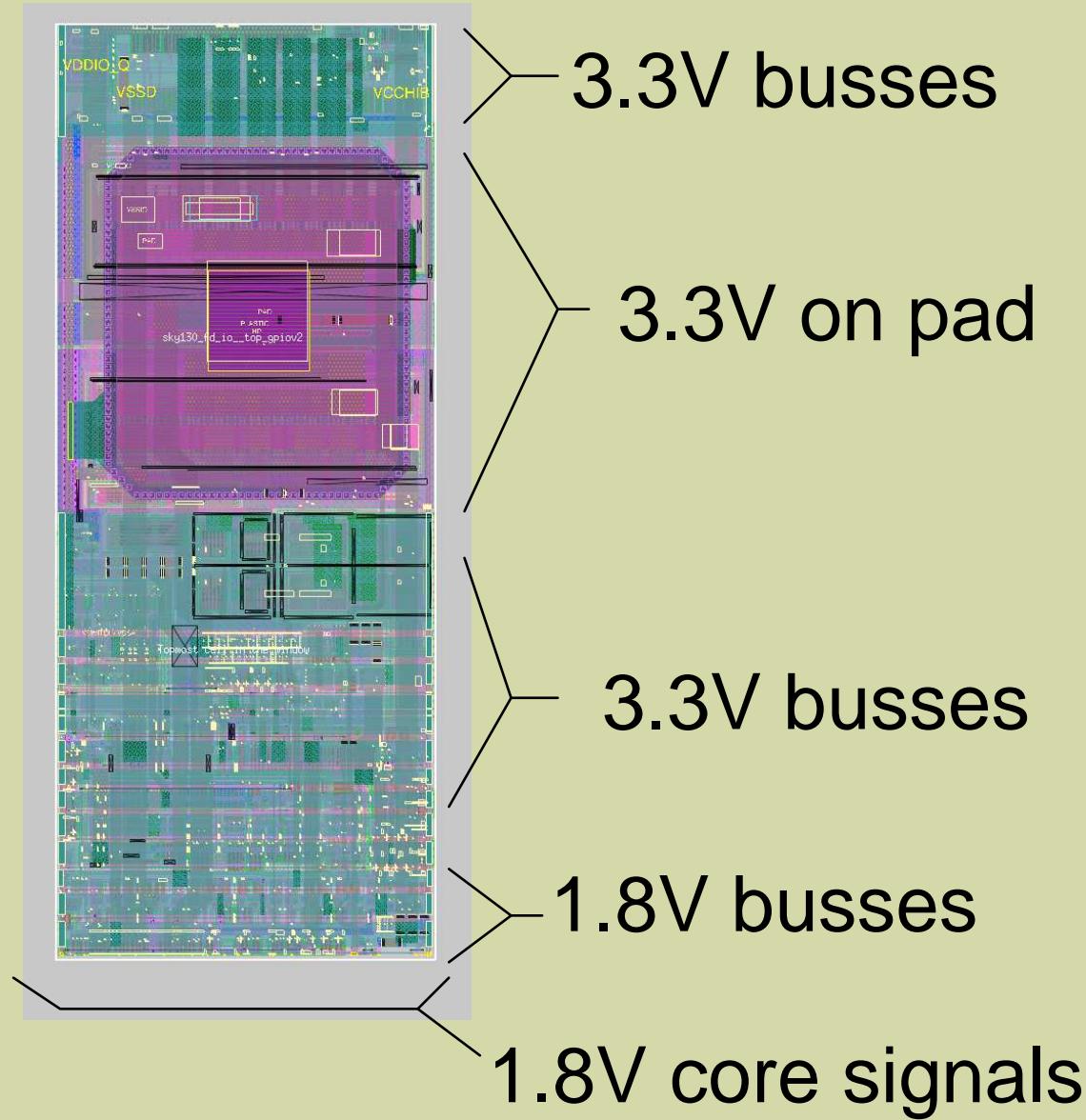
front-end layers



Understanding the SkyWater PDK

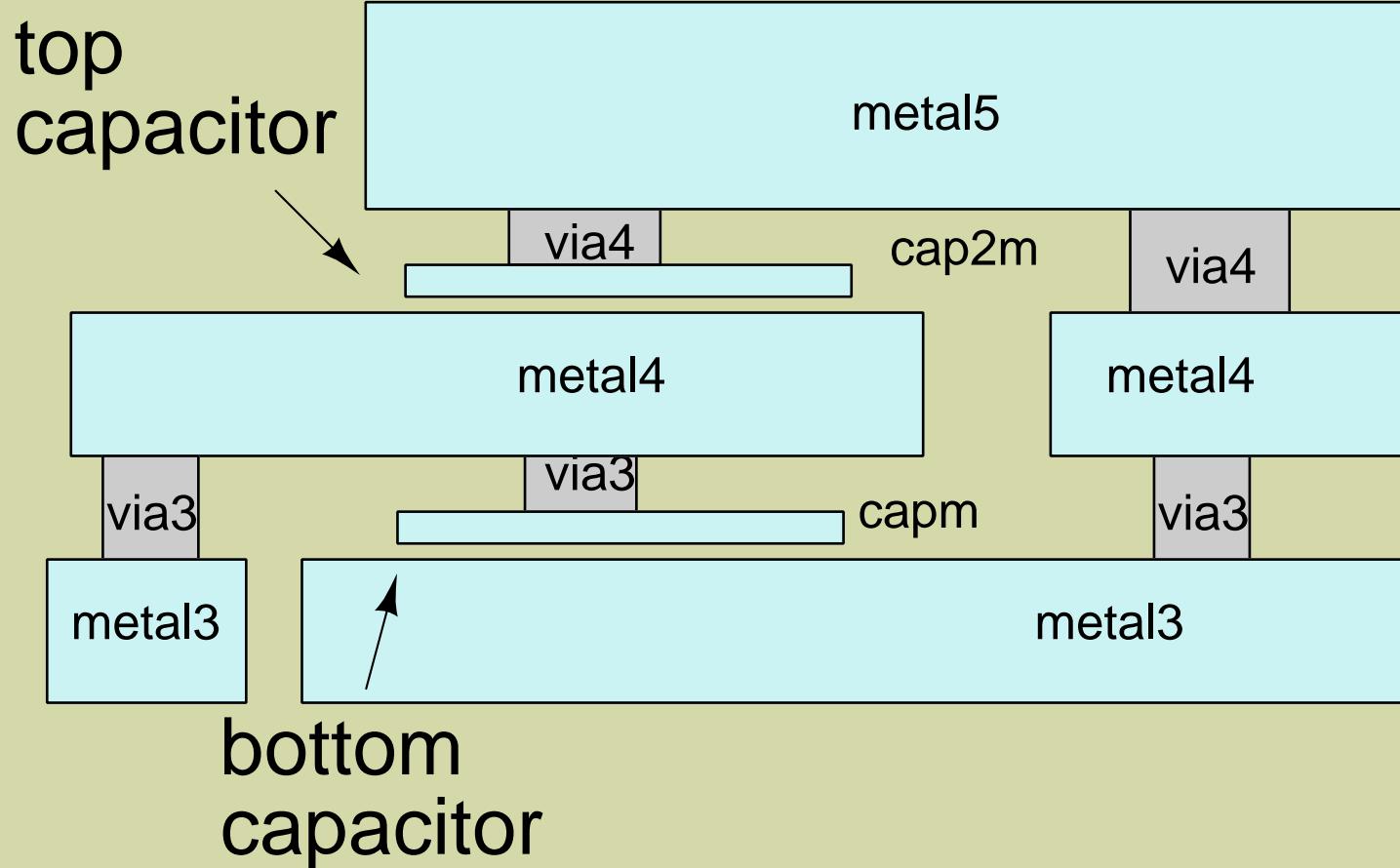
high voltage layer (HVI)

GPIO pad



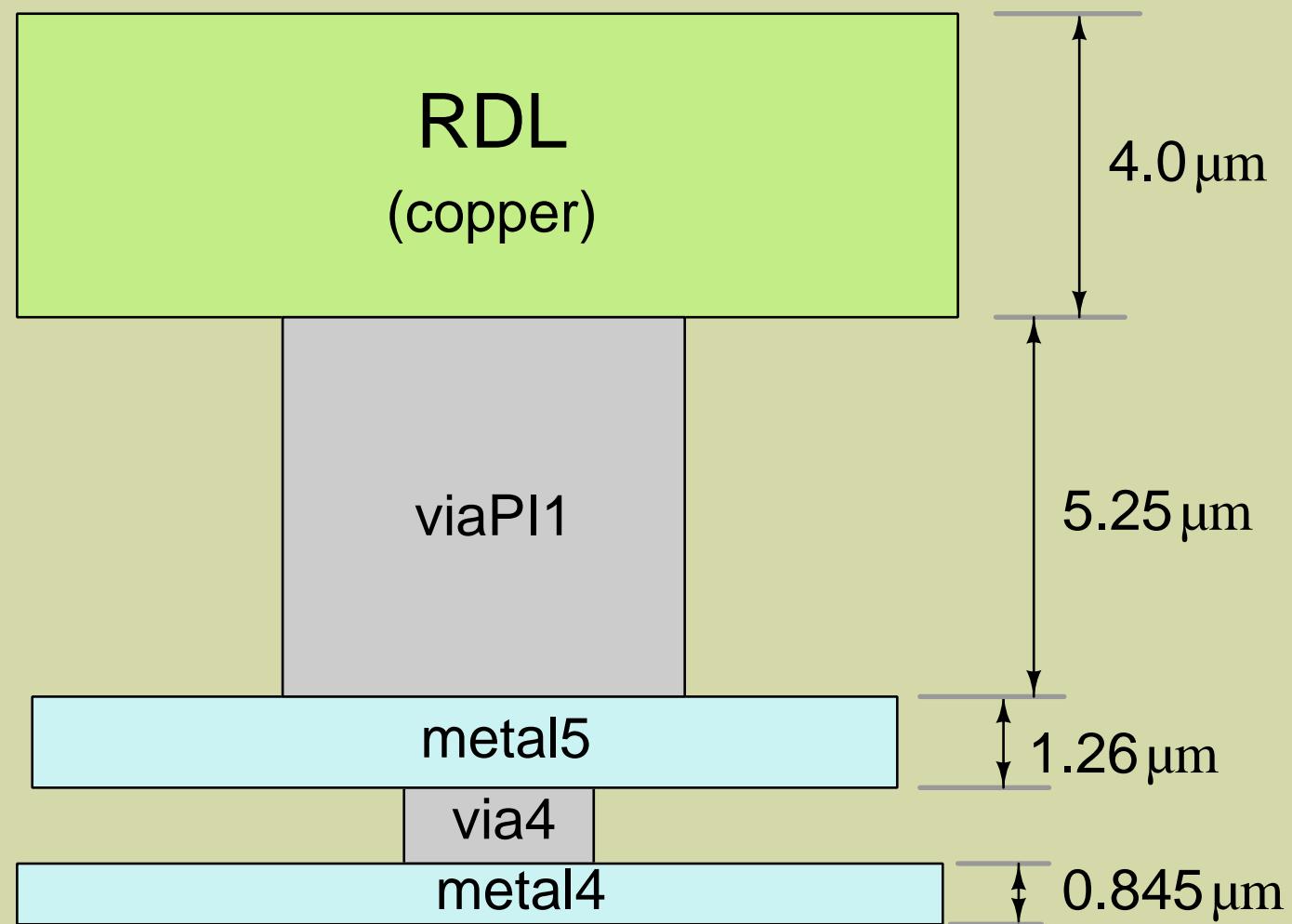
Understanding the SkyWater PDK

MiM cap layers



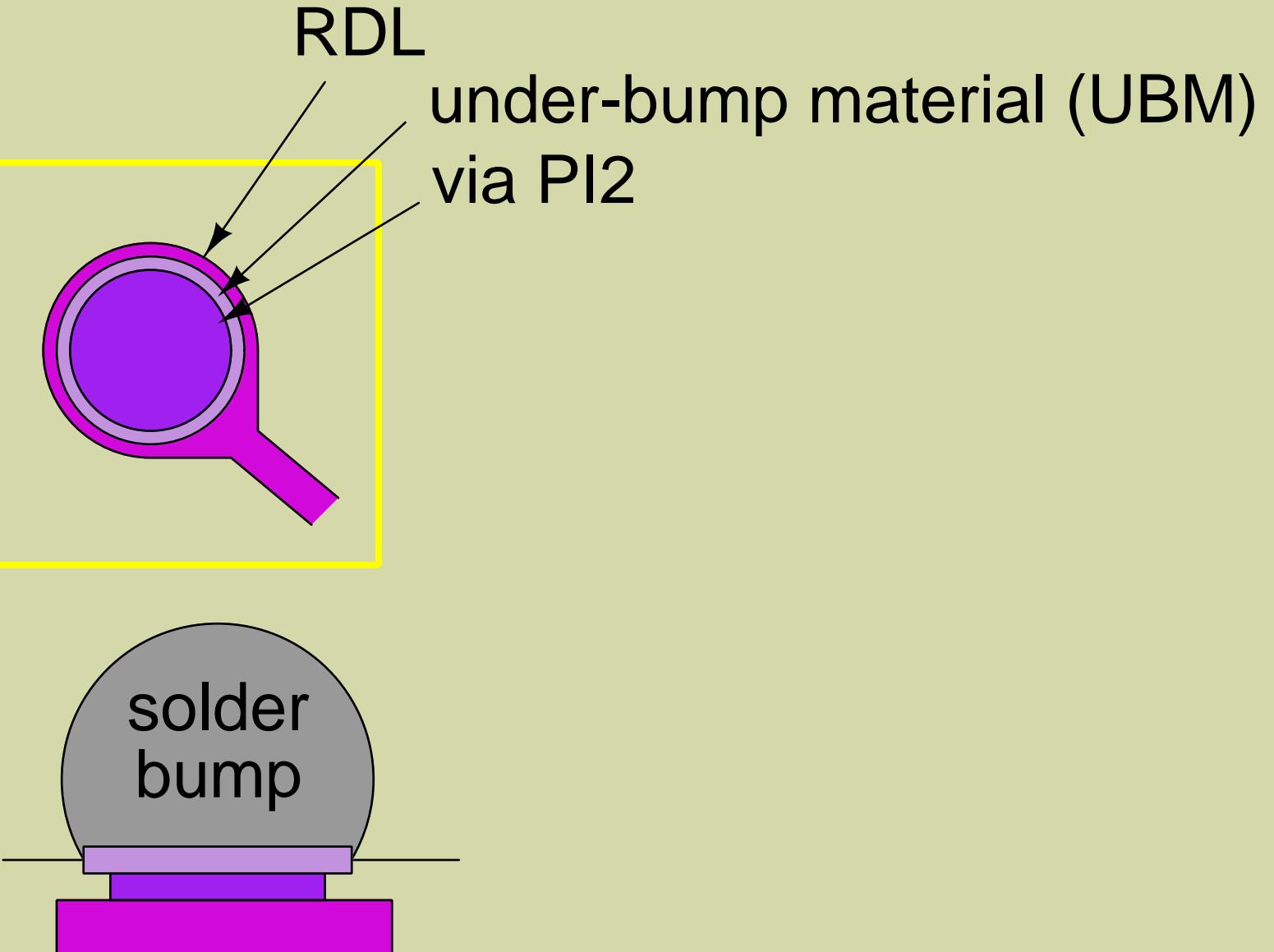
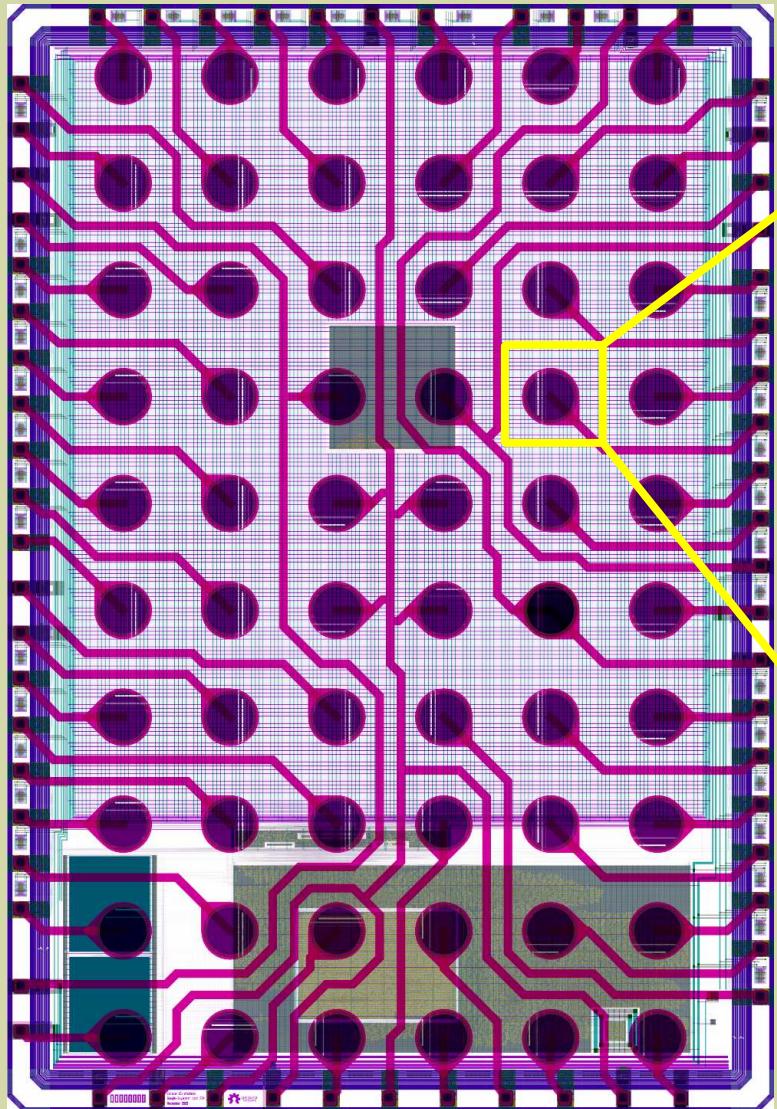
Understanding the SkyWater PDK

Redistribution layer



Understanding the SkyWater PDK

Redistribution layer

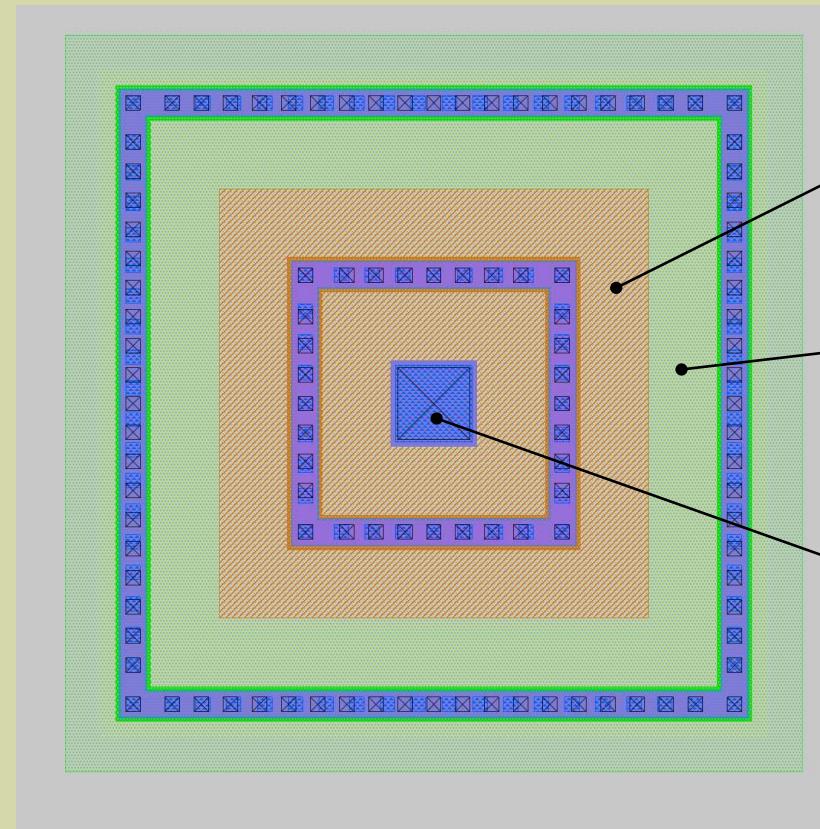


Understanding the SkyWater PDK

Devices

Understanding the SkyWater PDK Devices

bipolar NPN



base
pwell in
deep nwell

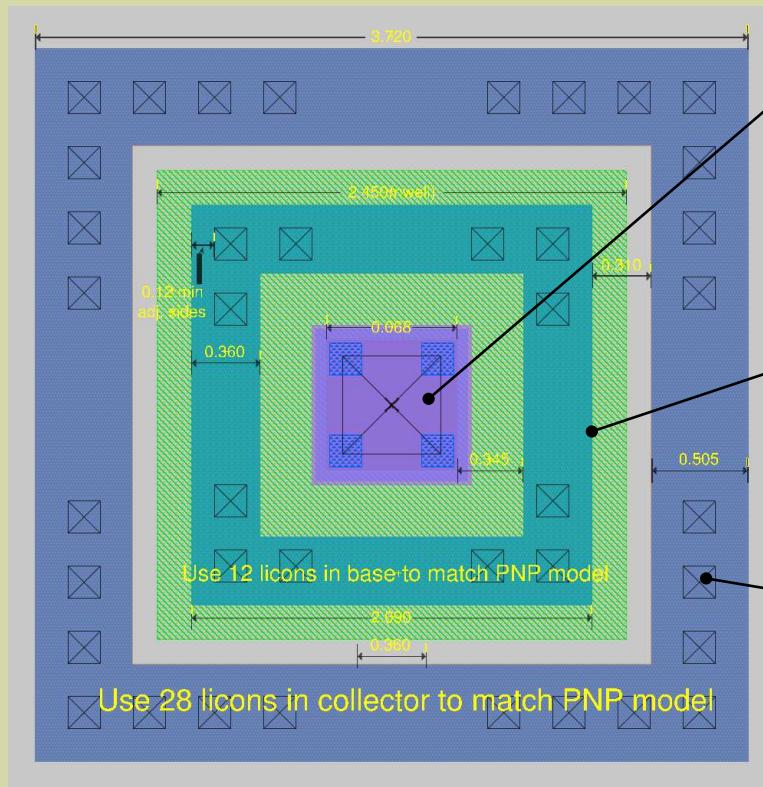
collector
n well

emitter
n diffusion

Understanding the SkyWater PDK

Devices

bipolar PNP



emitter

p diffusion

base

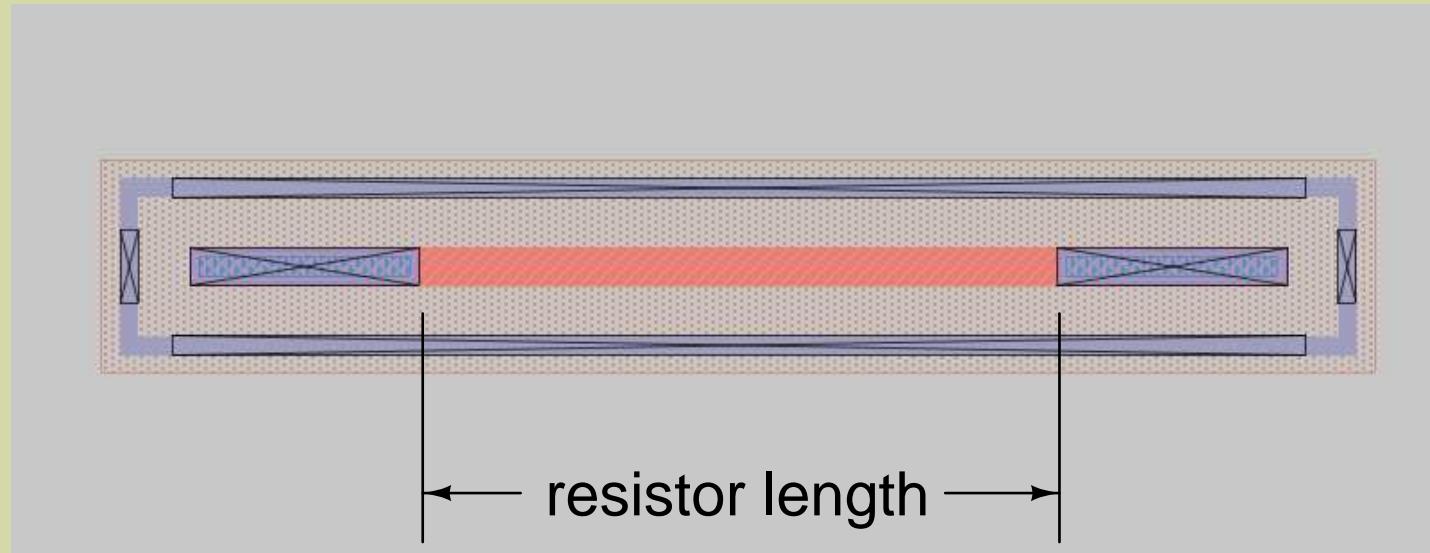
n well

collector

p substrate

Understanding the SkyWater PDK Devices

polysilicon resistors

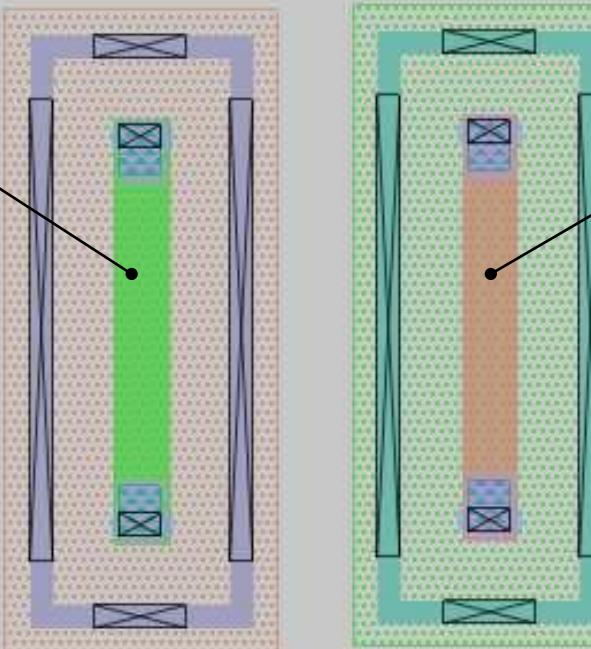


Understanding the SkyWater PDK

Devices

diffusion resistors

n diffusion
resistor on
substrate

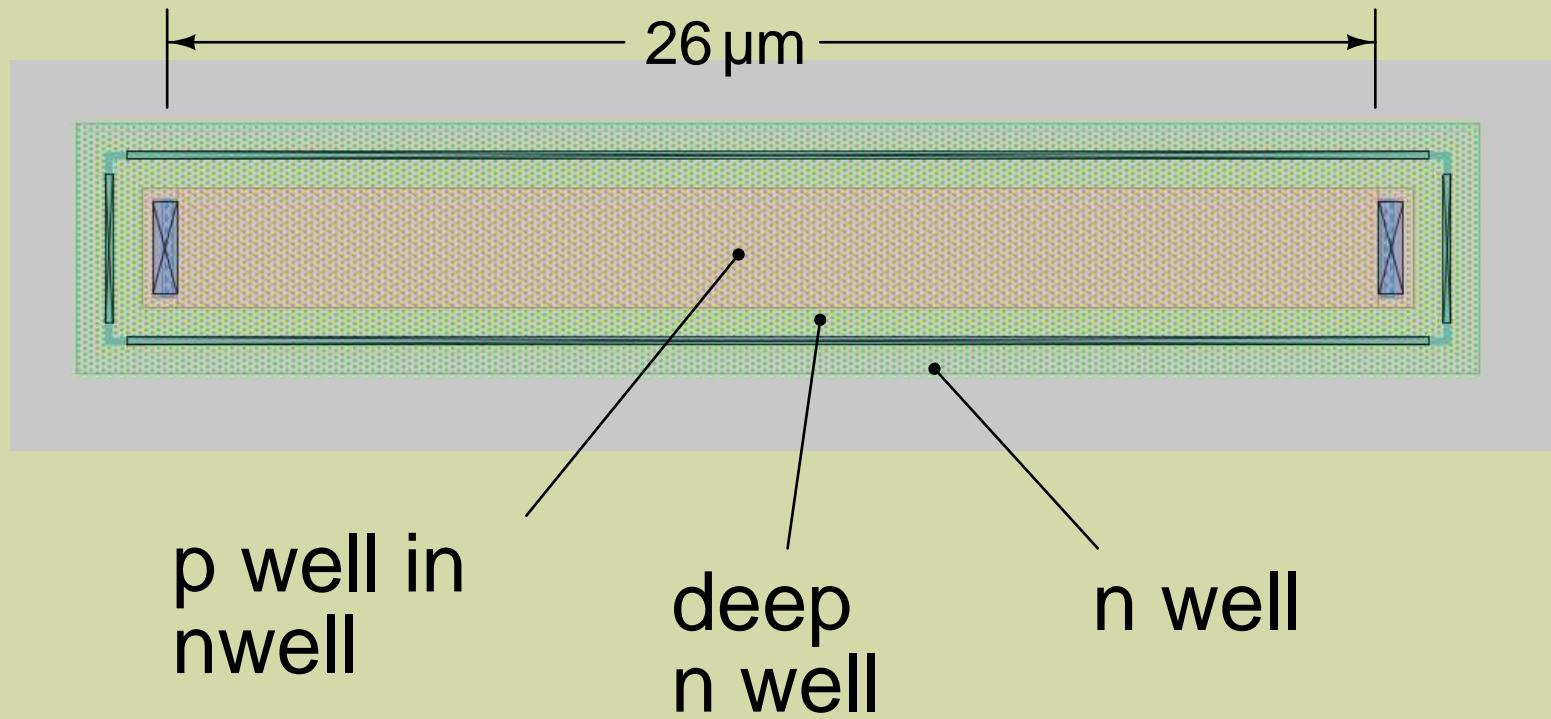


p diffusion
resistor in
nwell

Understanding the SkyWater PDK

Devices

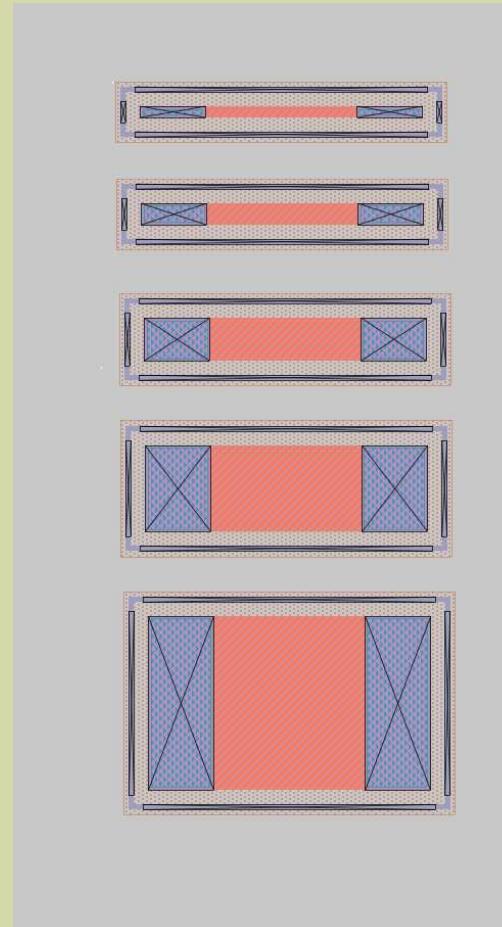
pwell resistor



Understanding the SkyWater PDK

Devices

Discrete widths



width = 0.35 μm

width = 0.69 μm

width = 1.41 μm

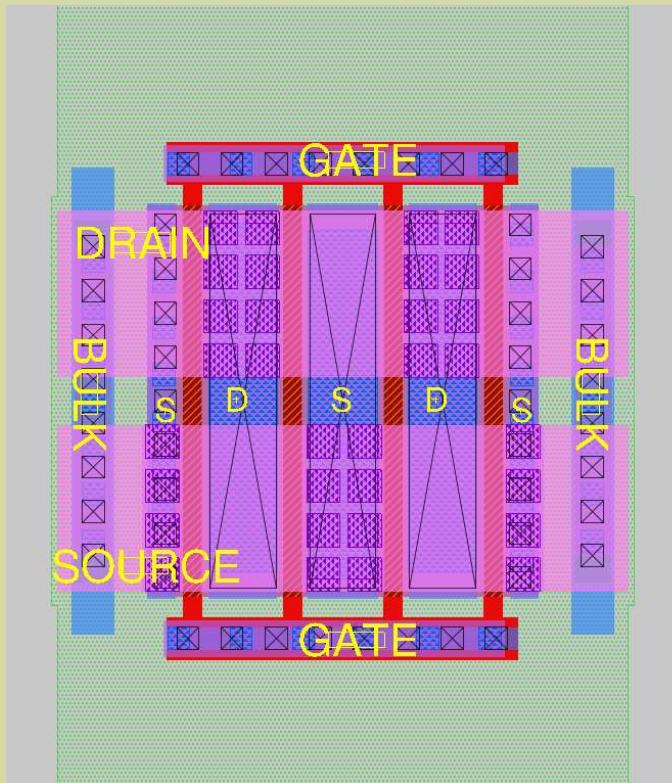
width = 2.85 μm

width = 5.73 μm

Understanding the SkyWater PDK

Devices

Reference layouts



RF 0.18V pFET

4 fingers

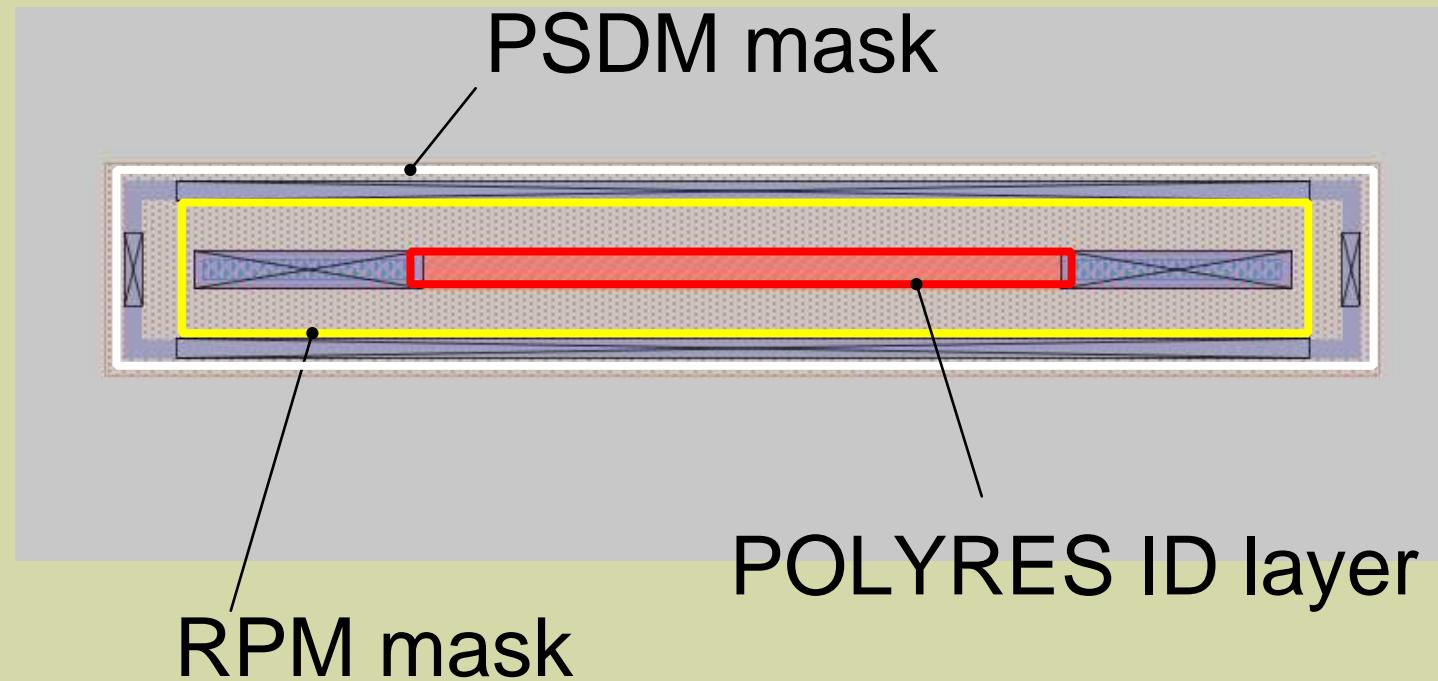
width = $3.0 \mu\text{m}$

length = $0.15 \mu\text{m}$

Understanding the SkyWater PDK

Devices

Hidden mask layers



Understanding the SkyWater PDK

Libraries

Understanding the SkyWater PDK Libraries

Digital standard cells

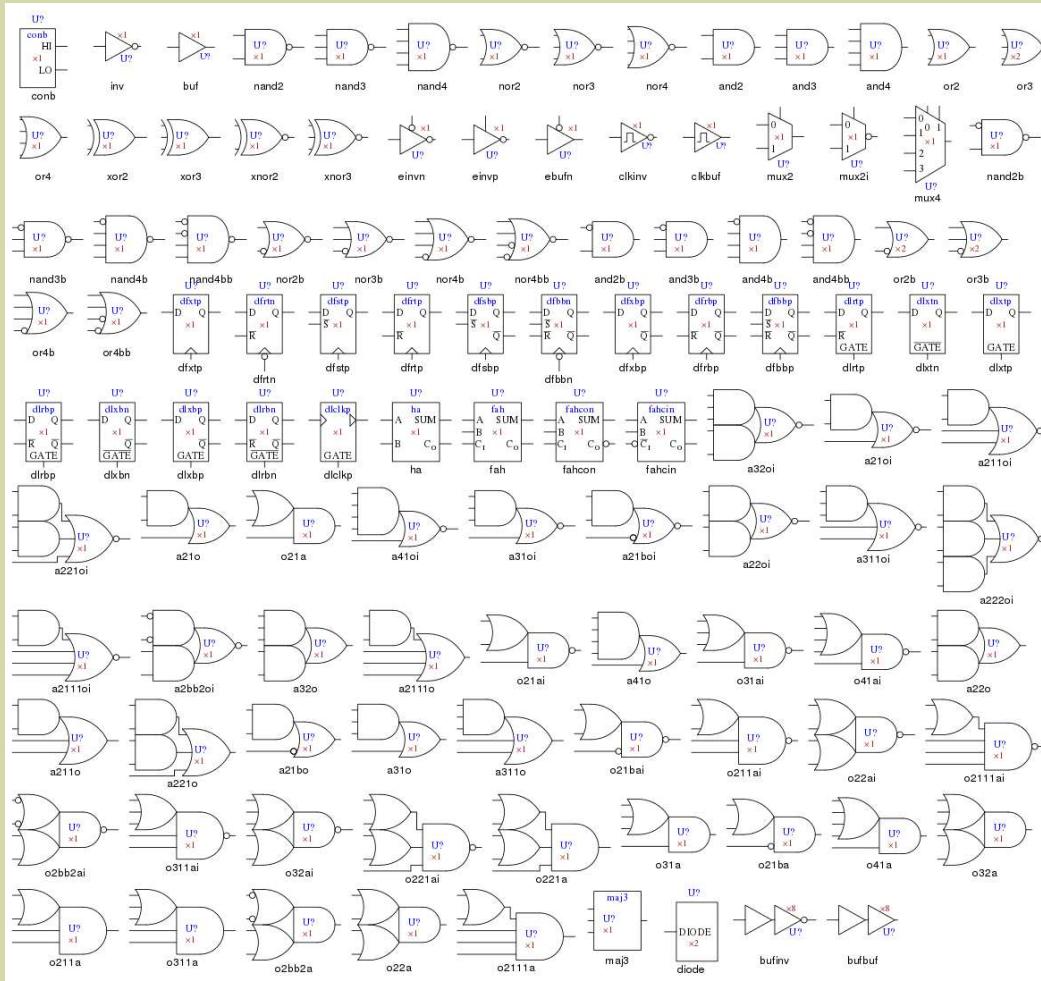
I/O cells

Primitive devices and models

Understanding the SkyWater PDK

Libraries

Digital standard cells



Understanding the SkyWater PDK Libraries

Digital standard cells

Naming convention:

sky130_vendor_library-type[_name]

Understanding the SkyWater PDK Libraries

Digital standard cells

Naming convention:

```
sky130_vendor_library-type[_name]
```

sky130_fd_sc_hd

foundry

standard cells

high density

Understanding the SkyWater PDK

Libraries

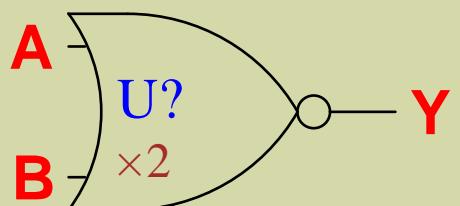
Digital standard cells

Naming convention:

sky130_vendor_library-type[_name]

library: **sky130_fd_sc_hd**

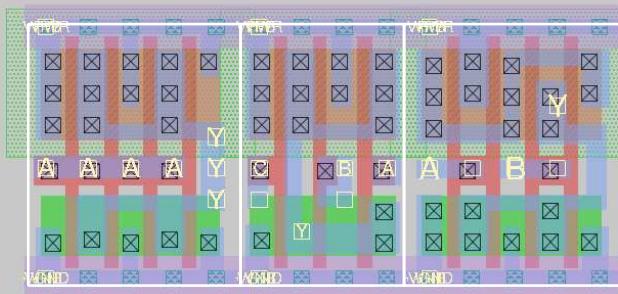
cellname: **sky130_fd_sc_hd__nor2_2**



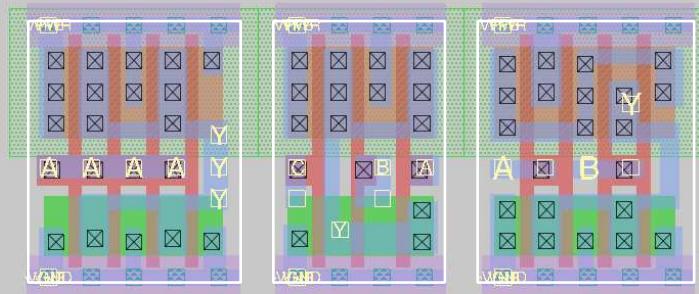
Understanding the SkyWater PDK

Libraries

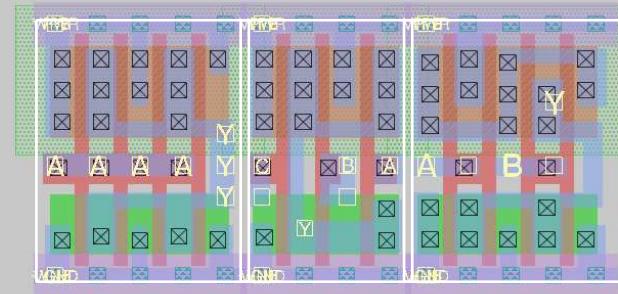
Digital standard cell abutment (when placed manually)



correct: abutment
boxes touching



too far apart

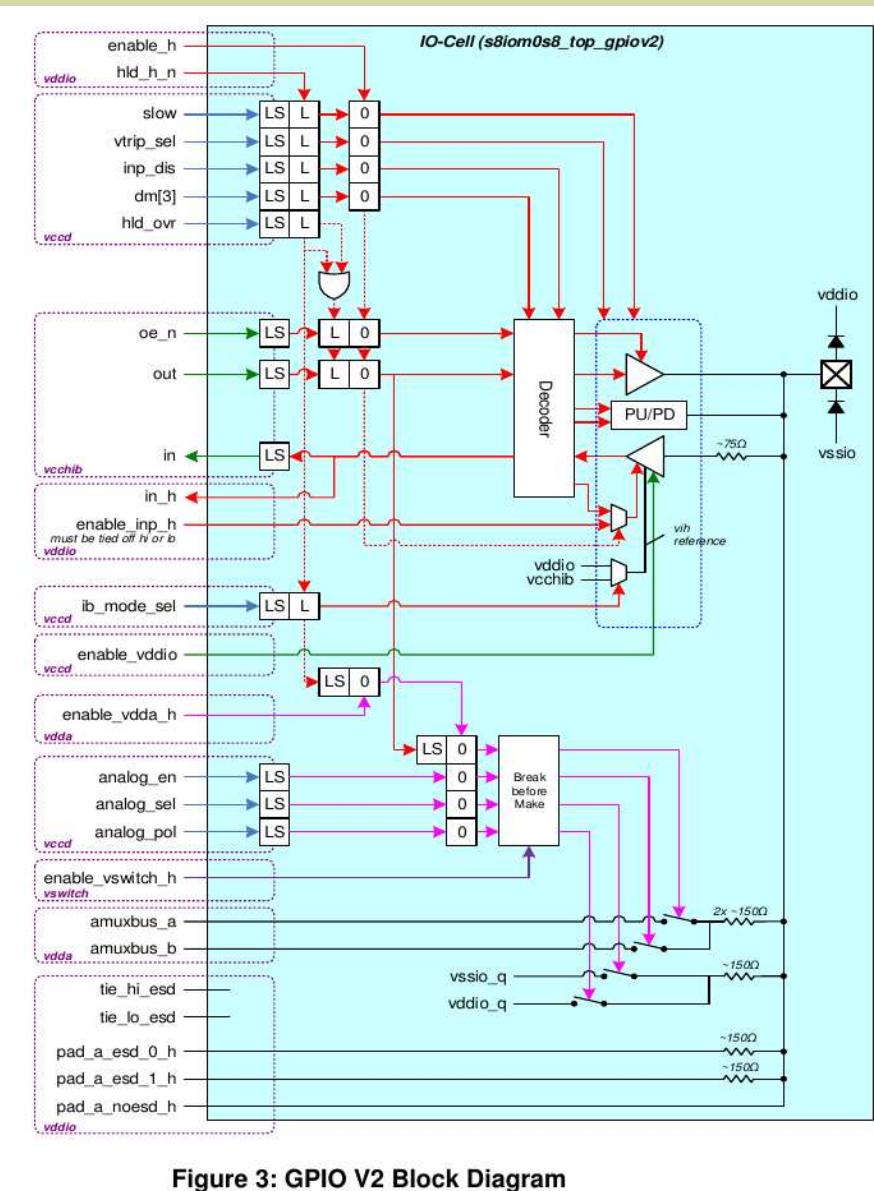


too close together

Understanding the SkyWater PDK Libraries

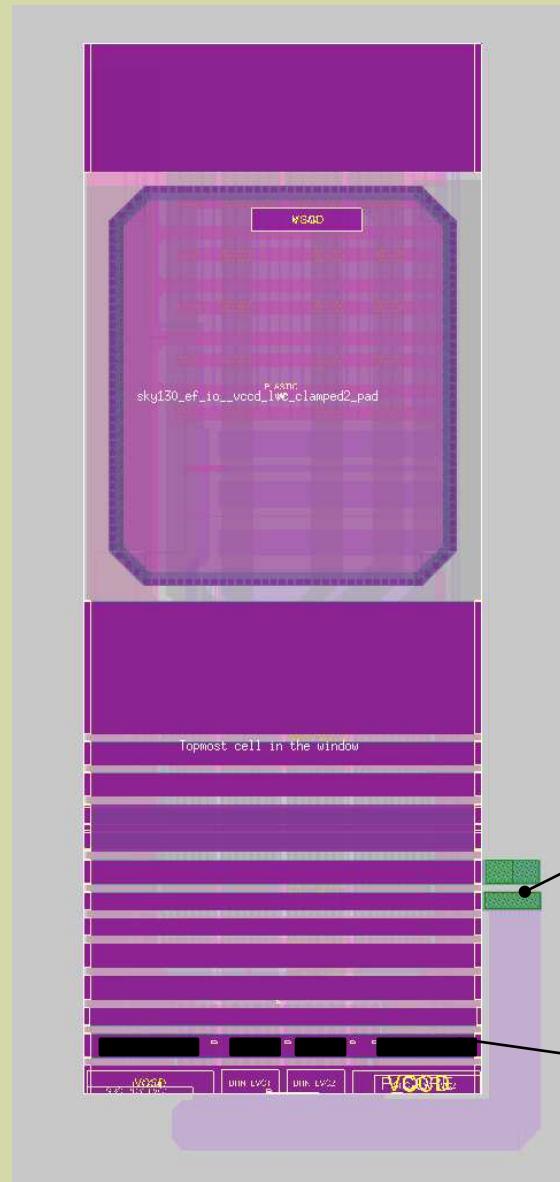
**Digital standard cells
Documentation??**

Understanding the SkyWater PDK



Libraries
I/O cells
input/output
sky130_fd_io
foundry

Understanding the SkyWater PDK



Libraries

I/O cells

sky130_ef_io

efabless

overlay connects
clamps to power
rail

overlay connects
pad to power rail

sky130_ef_io_vccd_lvc_clamped2_pad

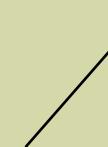
Understanding the SkyWater PDK

Libraries

primitive devices and models

`sky130_fd_pr` • primitives

foundry

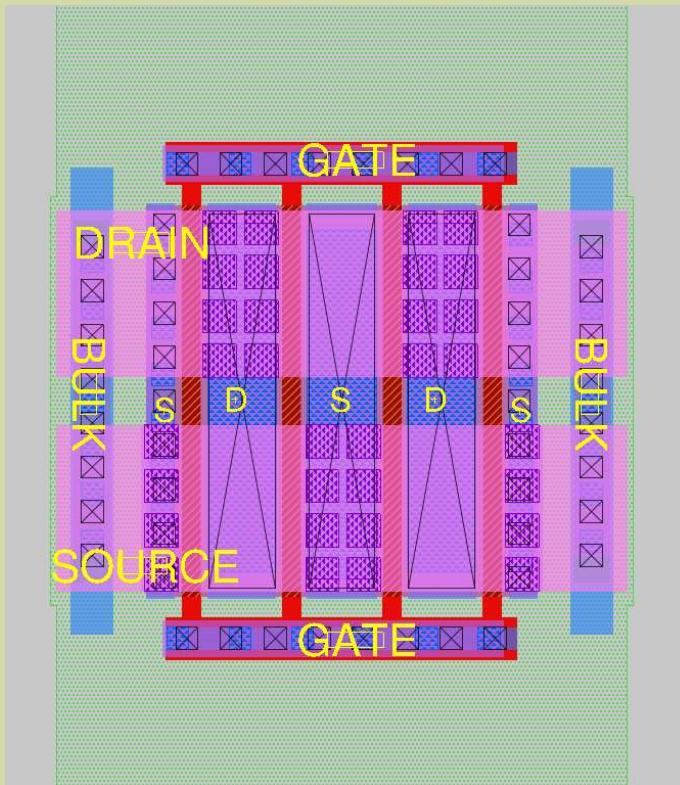


Understanding the SkyWater PDK

Libraries

primitive devices and models

sky130_fd_pr



RF frequency
reference layout

RF 0.18V pFET

4 fingers

width = 3.0 μm

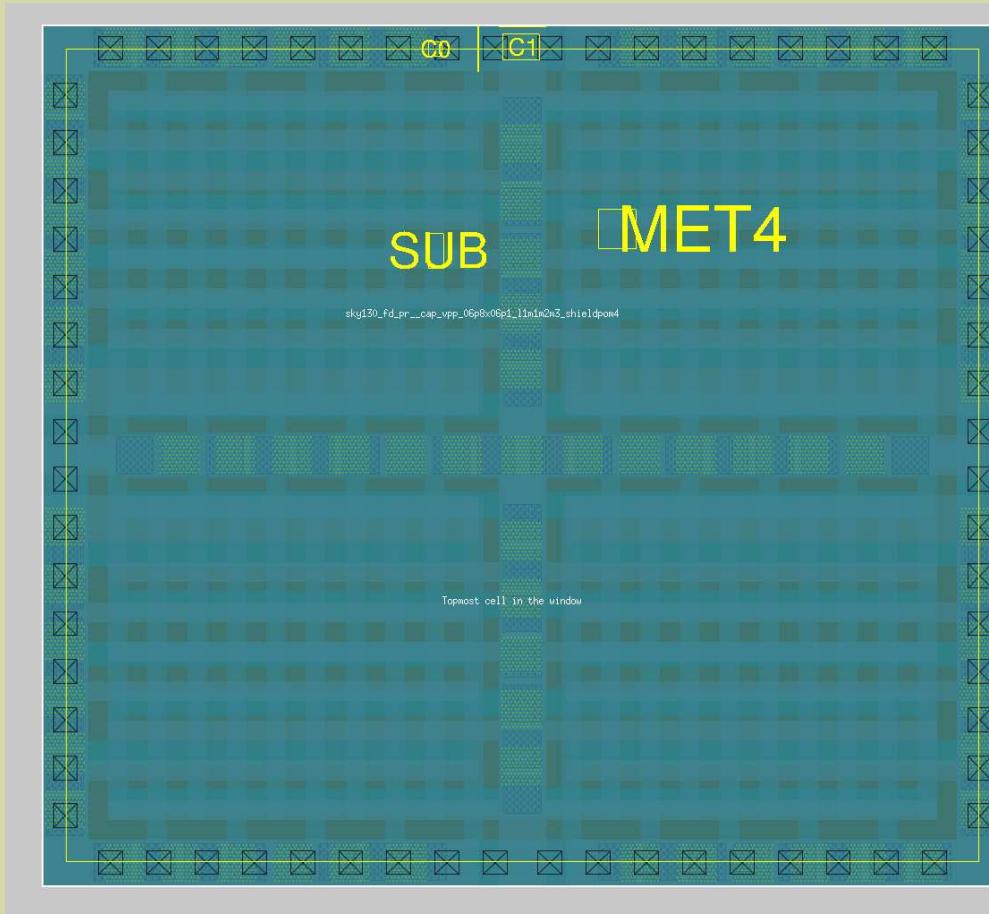
length = 0.15 μm

Understanding the SkyWater PDK

Libraries

primitive devices and models

sky130_fd_pr



**vertical
parallel
plate
capacitor
(vpp cap)**

`cap_vpp_06p8x06p1_l1m1m2m3_shieldpom4`

Understanding the SkyWater PDK

Libraries

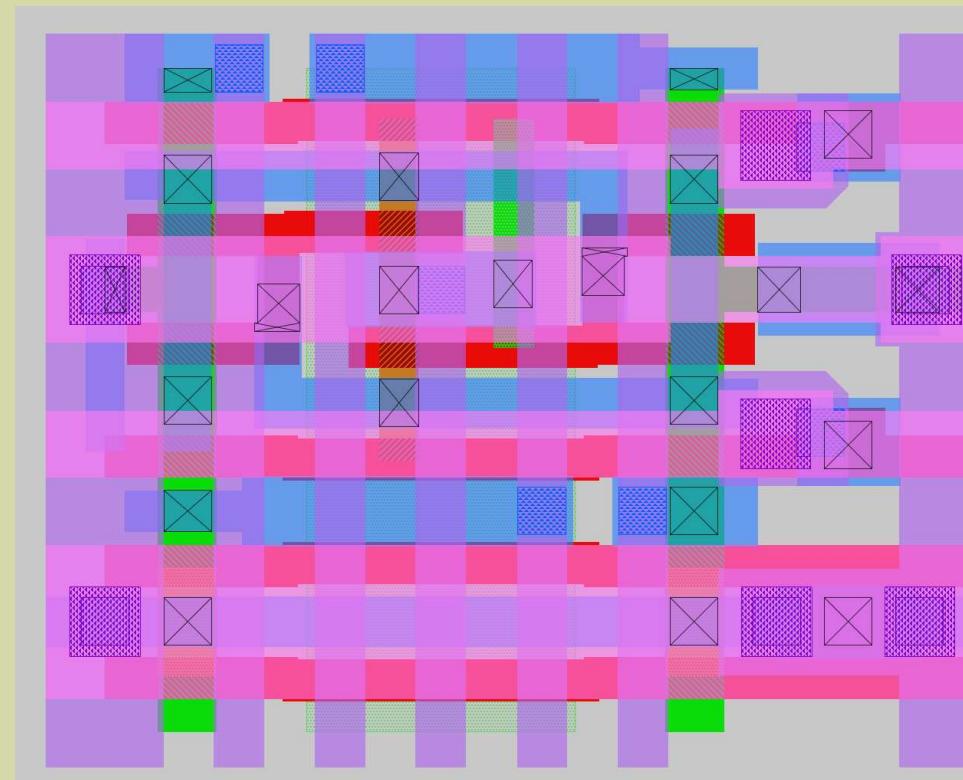
SRAM

sky130_sram_macros

sky130_fd_bd_sram

openRAM

foundry (not yet available)

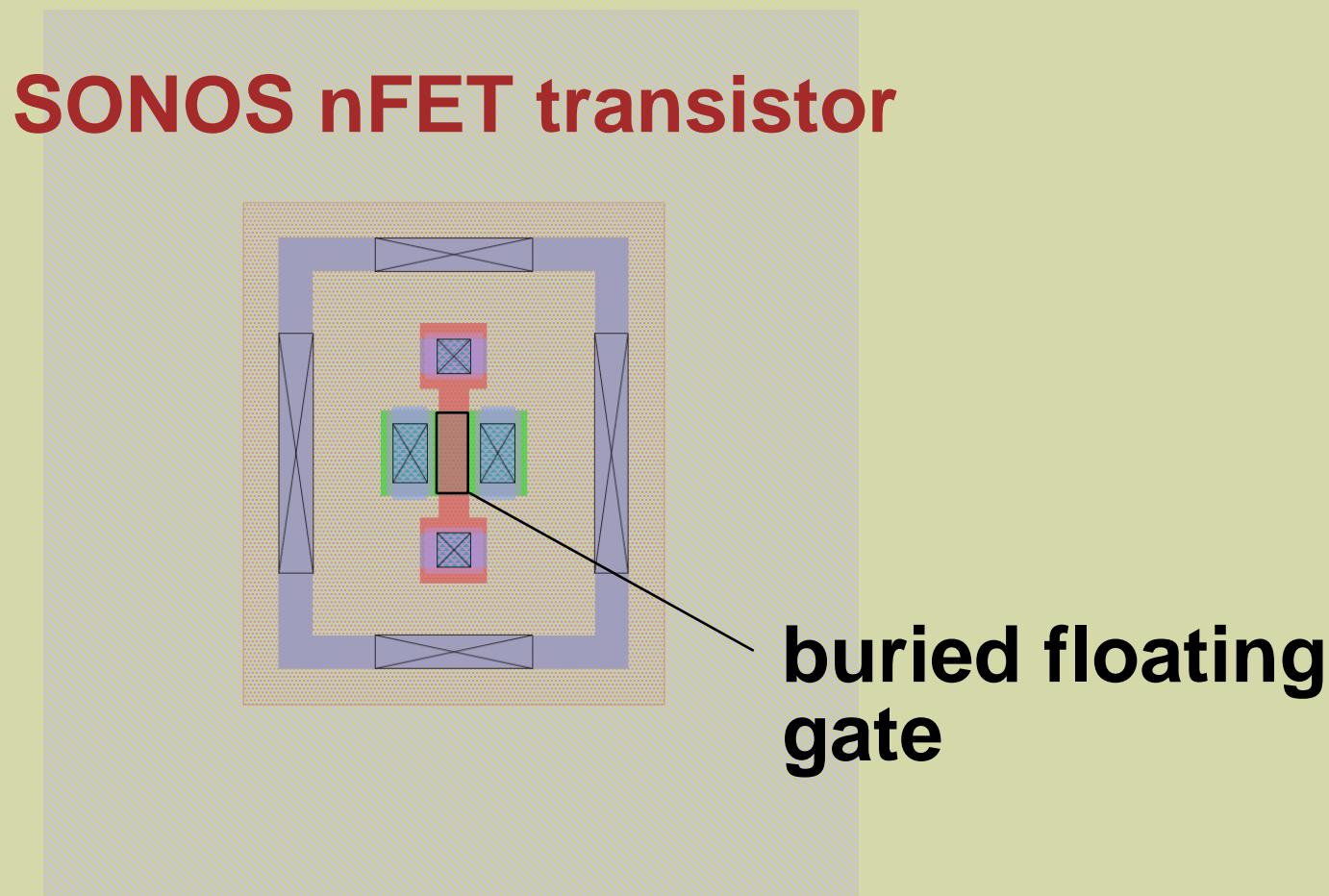


dual-port
SRAM
bit cell

Understanding the SkyWater PDK

Libraries

NVRAM



Open Source Tools and Flows

or... How can you design something?

Open Source Tools and Flows

A simple manual design flow

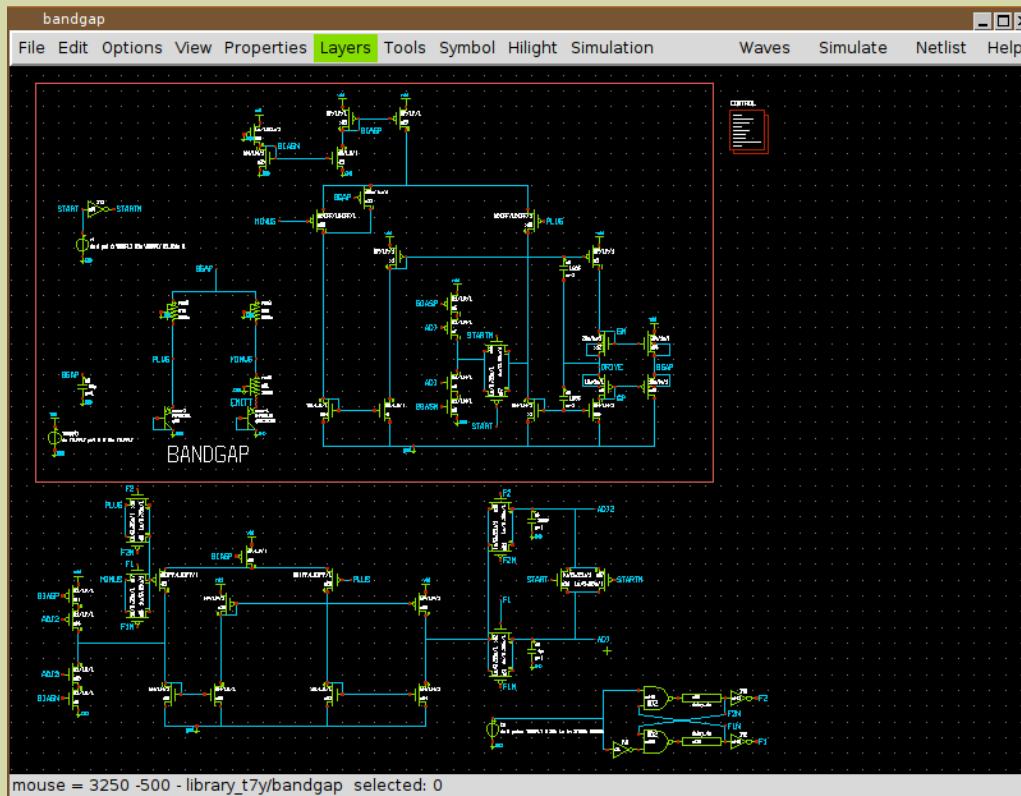
1. Schematic

Open Source Tools and Flows

A simple manual design flow

1. Schematic

xschem



Open Source Tools and Flows

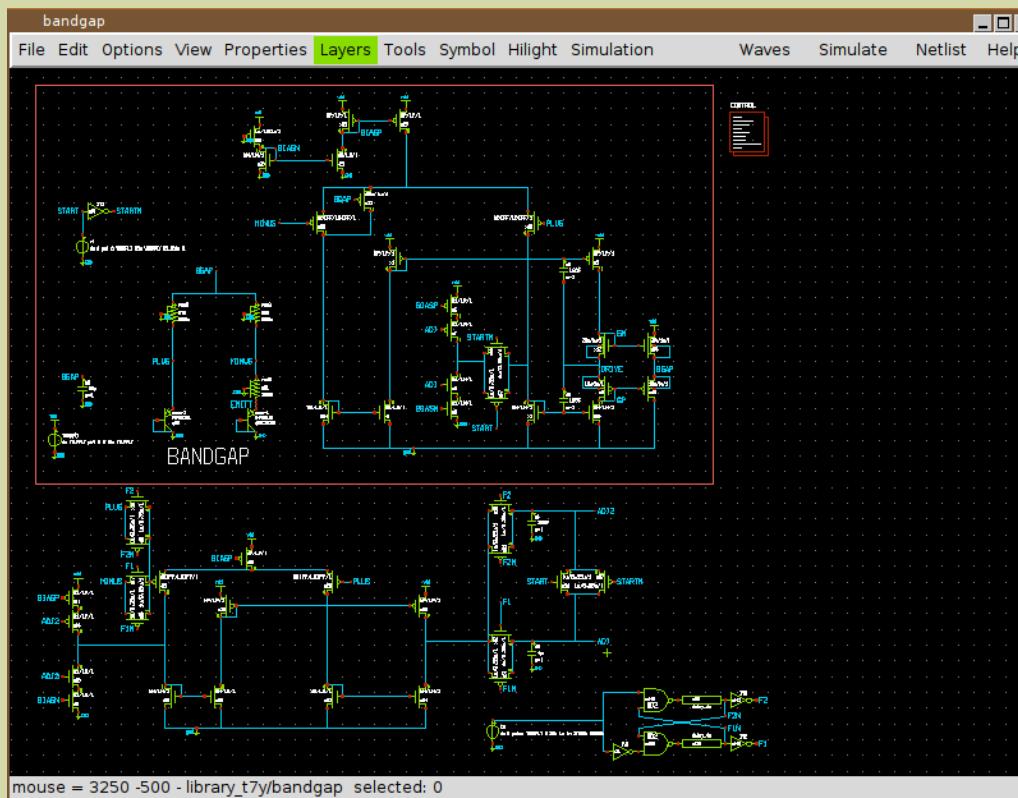
A simple manual design flow

1. Schematic

ngspice (analog simulation)

xschem

gaw (waveform viewing)



Open Source Tools and Flows

A simple manual design flow

1. Schematic



2. Layout

Open Source Tools and Flows

A simple manual design flow

1. Schematic

export →



simulate (ngspice)

2. Layout

Open Source Tools and Flows

A simple manual design flow

1. Schematic

export →



simulate (ngspice)

2. Layout

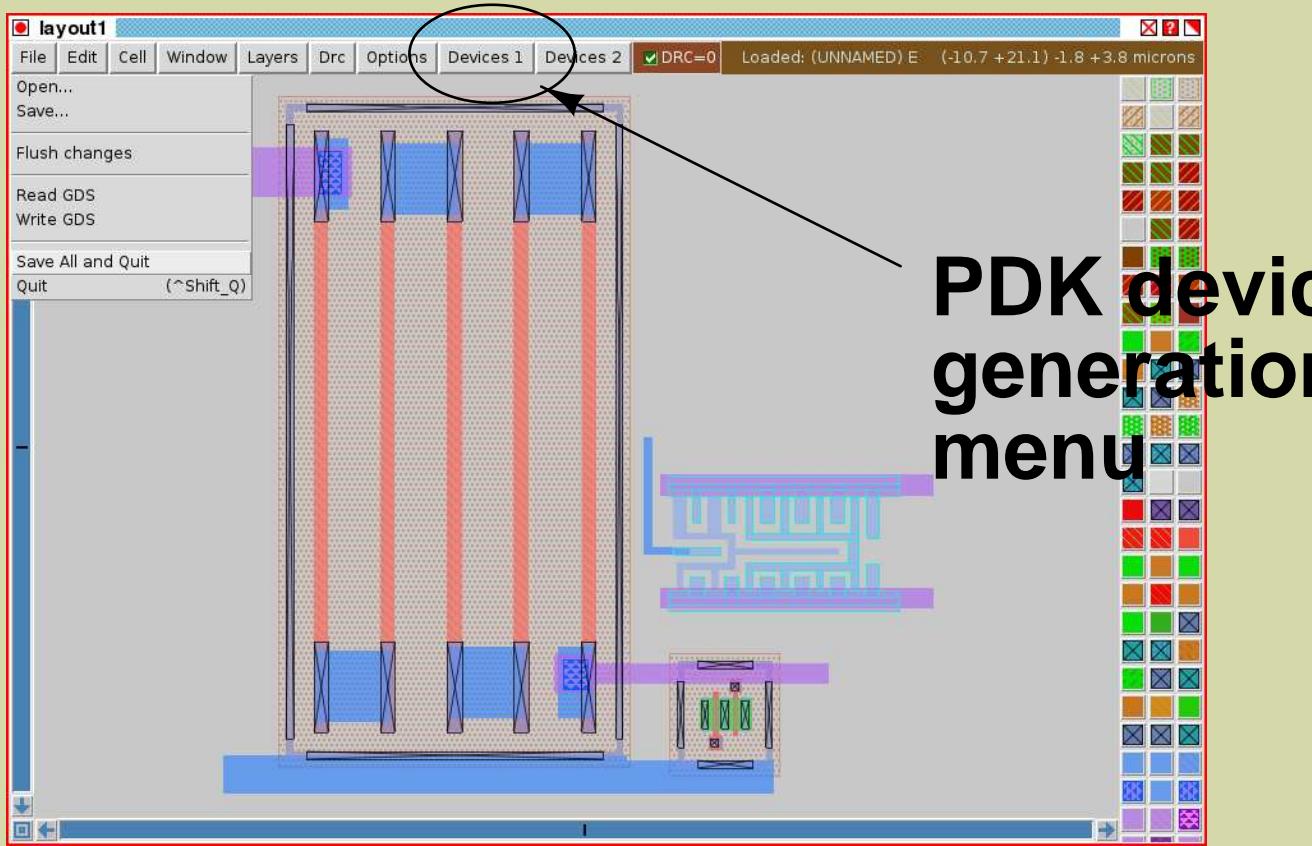


generate (magic)

Open Source Tools and Flows

A simple manual design flow

Parameterized devices in magic

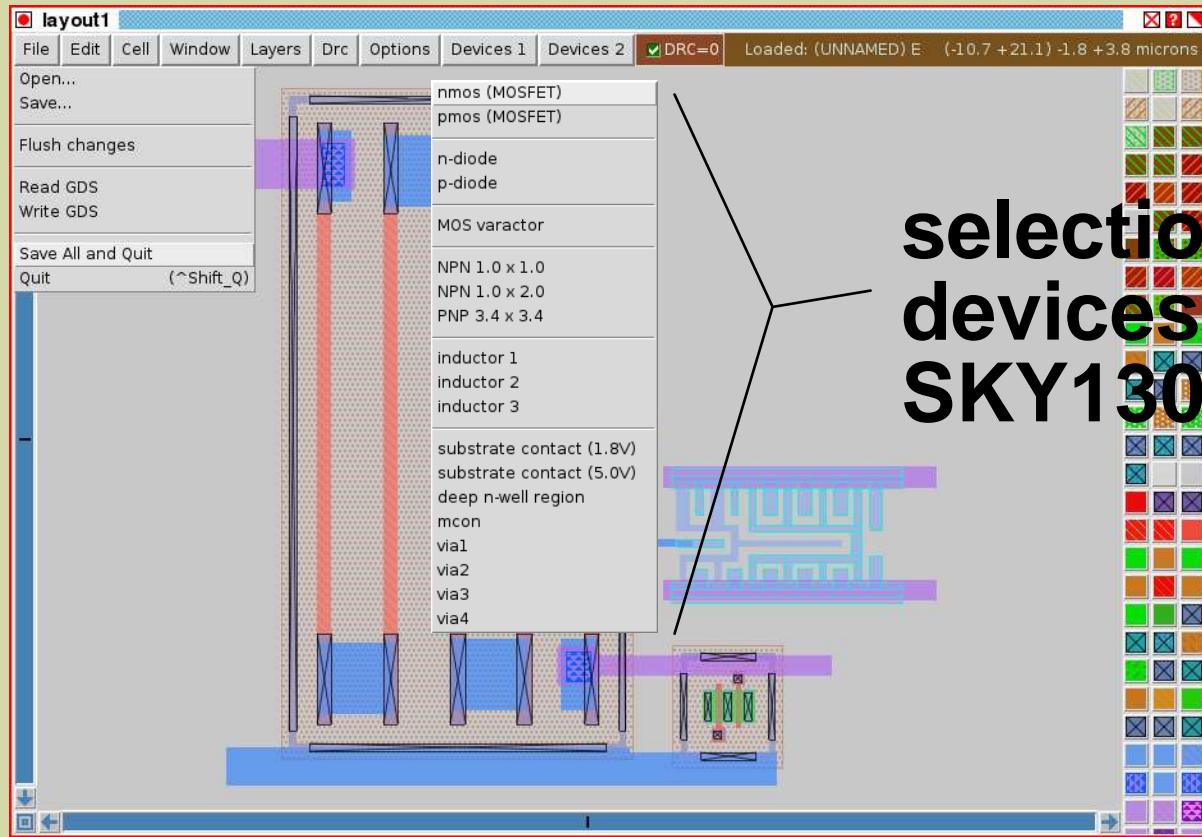


PDK device
generation
menu

Open Source Tools and Flows

A simple manual design flow

Parameterized devices in magic



Open Source Tools and Flows

A simple manual design flow

Parameterized devices in magic



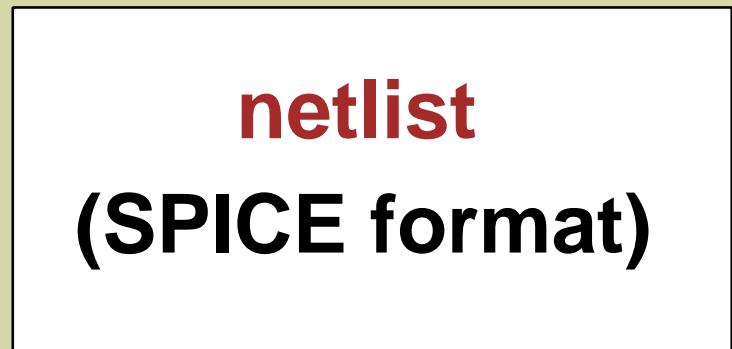
parameter selection window

Open Source Tools and Flows

A simple manual design flow

1. Schematic

export →

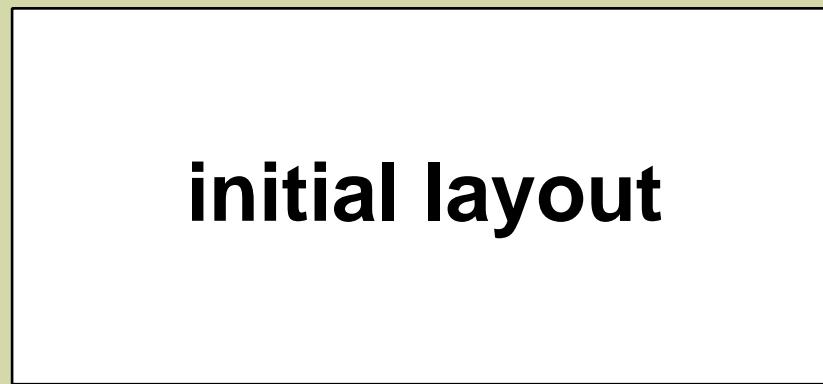


simulate (ngspice)

2. Layout

↓ generate (magic)

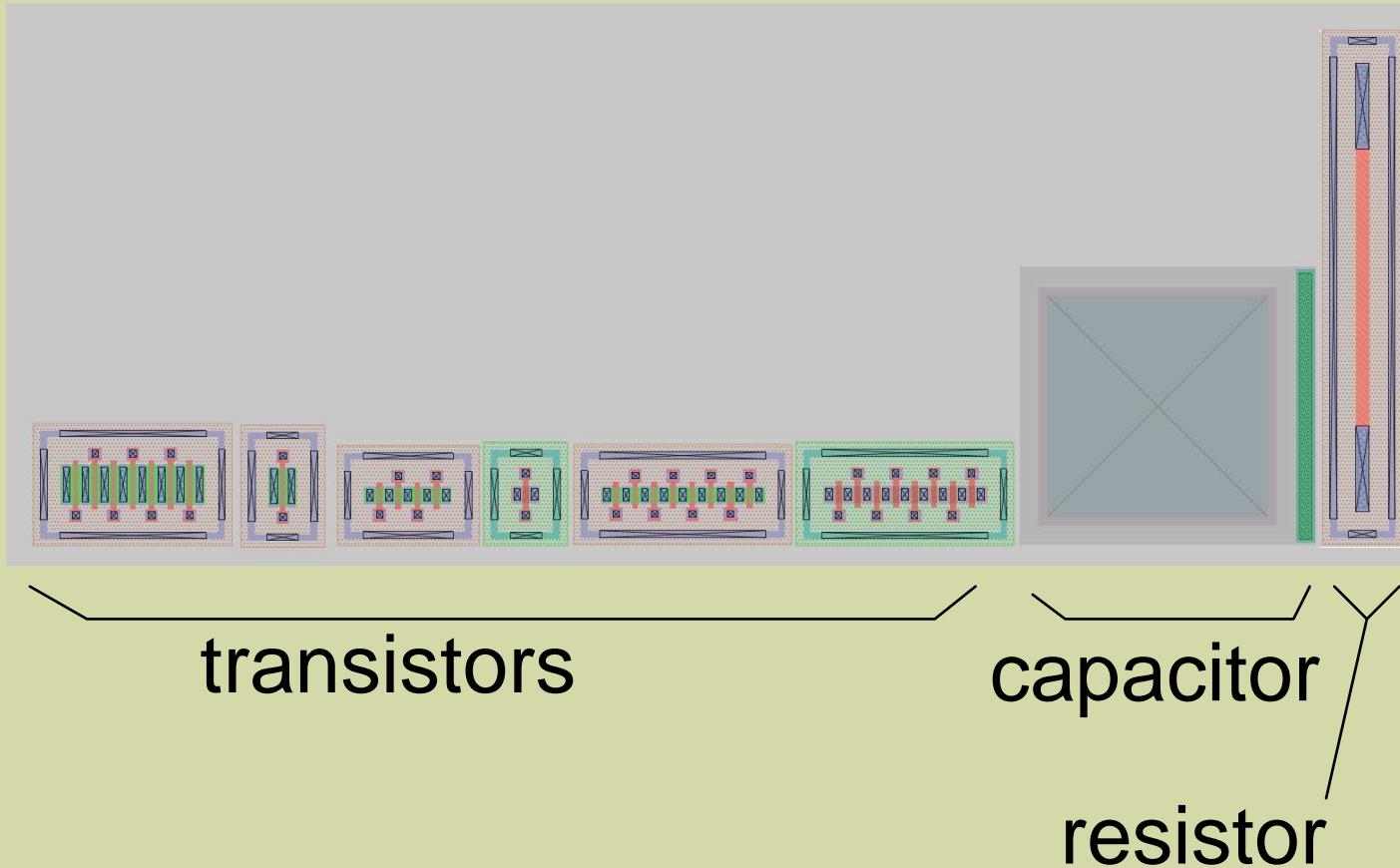
import →



Open Source Tools and Flows

A simple manual design flow

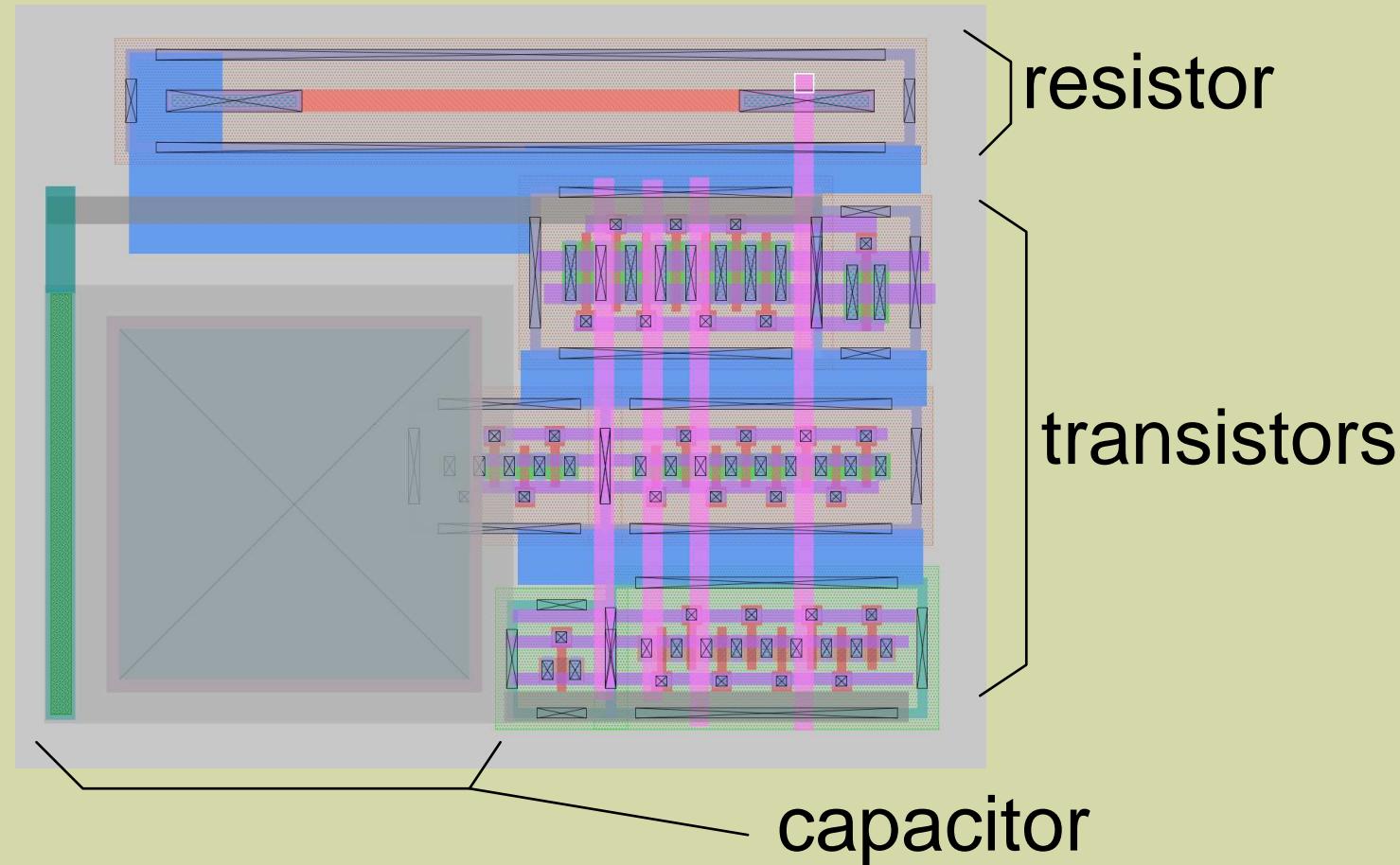
Example initial layout of devices in magic,
imported from schematic



Open Source Tools and Flows

A simple manual design flow

Analog example layout
work in progress in magic



Open Source Tools and Flows

A simple manual design flow

1. Schematic

export →

(schematic
capture)

netlist

(SPICE format)

2. Layout

export →

(extraction)

netlist

(SPICE format)

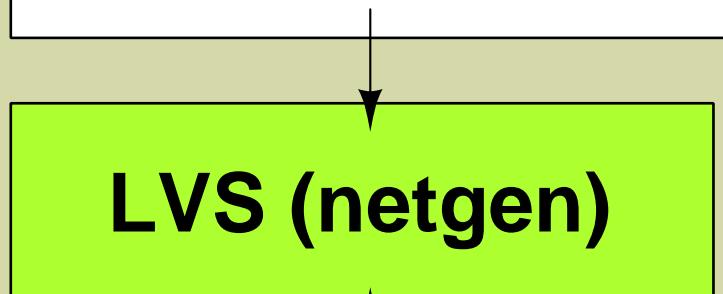
Open Source Tools and Flows

A simple manual design flow

1. Schematic
export →



2. Layout
export →



Backend validation DIY

Running a simple example through the entire validation flow

- 1. xschem Create a schematic**
- 2. ngspice Simulation validation**
- 3. magic Layout, DRC validation**
- 4. netgen LVS validation**