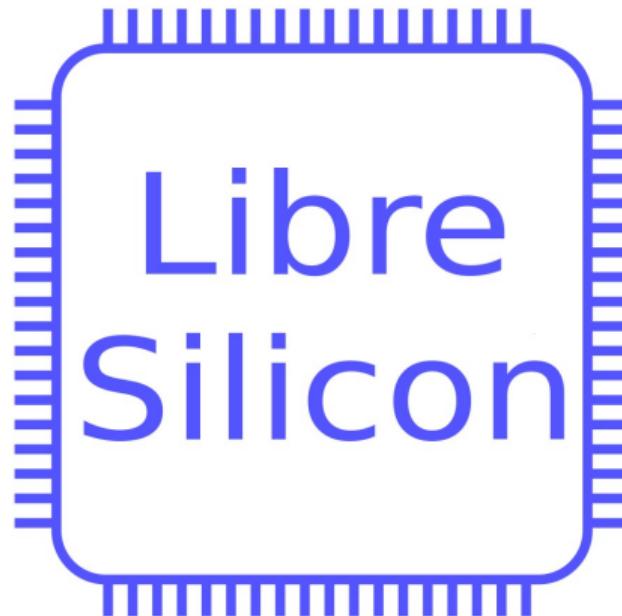
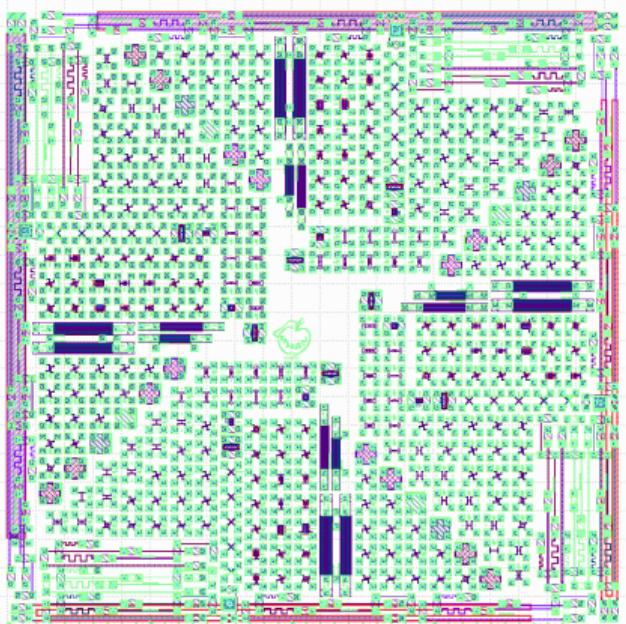


# LibreSilicon - Breaking the microchip monopoly



# The LibreSilicon project

- In 2017 leviathan already has found a clean room to rent at Hong Kong University of Science and Technology
- Last year at 34c3 he gave a Lightning Talk about LibreSilicon
- Since then we're meeting every week on Sunday 2100 HKT at Mumble
- Communicating, planing and working via mailing list and Mumble
- Held a tool chain hackathon end of May 2018
- Already two of us got qualification for clean room access at HKUST
- Processing our first test wafer for characterization (→ pics follow)

## Basic points

- Starting with  $1\mu m$  "feature size" because still well documented in text books
- Robust, at least 5 Volt tolerant, well suited for maker, tinkerer and hacker
- Twin-Well process for CMOS with 3 metal layer w/ very interesting additions
- Quite suitable for "low tec" in the basement
- For analog circuits, regarding their huge transistor sizes, small feature sizes do not matter

# Main (re-)construction areas

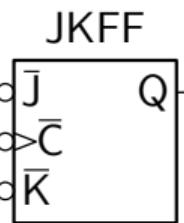
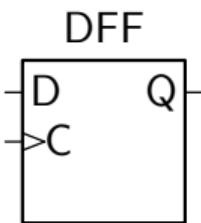
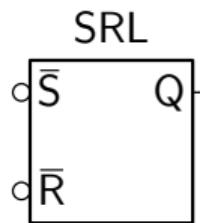
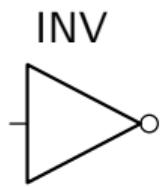
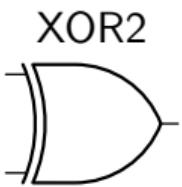
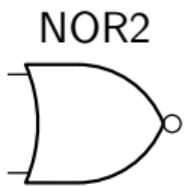
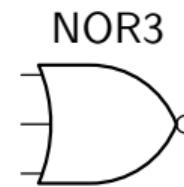
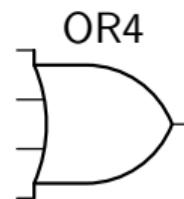
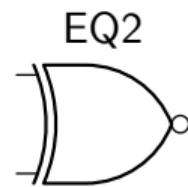
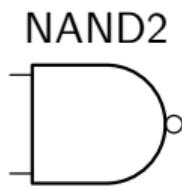
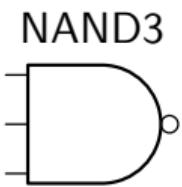
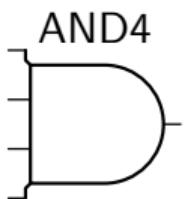
- ① Figure out / develop the process itself (→ almost done)
- ② Rebuild / modernize the tools / design flow (→ ongoing)
- ③ Compile / design a almost complete standard cell library (→ ongoing)

# Standard Cells

Standard cells are usually

- A collection of some dozens of combinatorial + sequential cells
- Instantiated many, many times in a netlist
- Used also as layout primitives

## Typical cells



... and much more

## Design goals

- Being almost complete (more cells → better netlist)
- Being low energy consuming (less Watts per square)
- Being as fast as possible (w/ small timing delay)
- Having a small footprint (small cell size)

Well, does not fit all together well.

## Cell representations

every single cell needs different representation for working smooth with different tools,  
e.g.

- For simulation (→ Verilog and Spice)
- For synthesis (→ Liberty file format)
- For timing (→ standard delay format)
- For layout (→ library exchange format)
- Or, even for dedicated tools (e.g. Magic)
- And documentation (schematics, truth tables, data sheets, ..)

# Cell representations

Regarding our goal of several hundreds cells (estimated 300+), generating **all** cell representations, becomes a huge task.

Nobody likes to do this manually.

We need a tool for that!

A cell generator.

# Cell generator

This cell generator, named "Popcorn", is still work-in-progress.

Starting from one source this tool

- Should generate all representation formats
- Already helps drawing the schematics
- Already generates a couple of data sheet like LaTeX file
- Was written in Tcl first, but
- Needs a rewrite in a more sophisticated / AI-ish language
- Now when test wafer characterization is done

# Standard cell library

- <https://www.github.com/chipforge/StdCellLib> (→ repository)
- [https://vcs.in-berlin.de/chipforge\\_stdcelllib/index](https://vcs.in-berlin.de/chipforge_stdcelllib/index) (→ wiki)

maintained by chipforge

# Open source tools

- yosys
- graywolf
- qrouter
- several FPGA routers

# graywolf

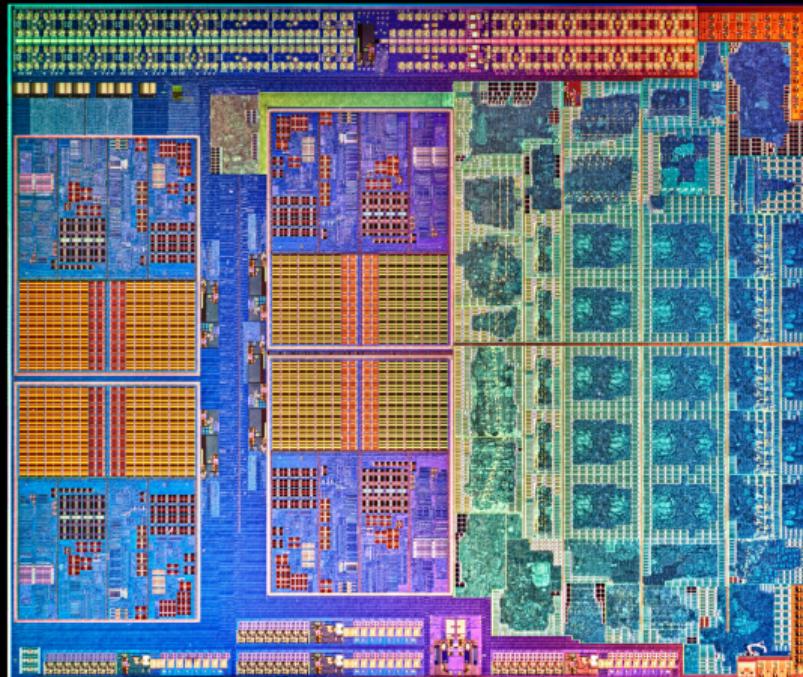
- Originates in academia: TimberWolf
- Simulated annealing
  - Meta heuristic that is useful not only for placement
- Inline syscalls
  - This is just a bad idea

# qrouter

- Started in 2011 by Tim Edwards
- Widely used for FPGA
  - Not ready for silicon
- Sequential routing
  - Parallelism not in scope
- Difficult to prove formal correctness
  - Prove that C implementation of rip-up and re-route is correct

# Productive tools

- Different tool sets like BonnRoute, Cadence suite, Alliance tools, etc.
- Similar capabilities with respect to silicon
- Just throw man-power at VLSI — what is automation?



## Routing: State of the art

- Place components for a large chip
- Route wires roughly along a chessboard for a large chip
- Route detailed tracks and vias for a large chip
- Formal correctness: Rip-up and Re-route
- Formal style: Sequential/Imperative code

## Routing: Proposed

- Decomposition for a large chip
- Place components and route for small chips in parallel
- Place abstract gates and route recursively
- Formal correctness: Reduction from SMT
- Formal style: Parallel/Declarative code

# Divide and Conquer

## Academia + Industry:

- Placement and Routing are different problems
- All components map to the same problem

## LibreSilicon:

- Placement and Routing are the same problem
- Different components map to different problems

# Routing Hierarchy

## Academia + Industry:

- Geographical partitioning of a wafer → *cut tree*
- Based on preceding placement steps

## LibreSilicon:

- Modular chip development → *subcell hierarchy*
- Subcells carry implicit and explicit subcells

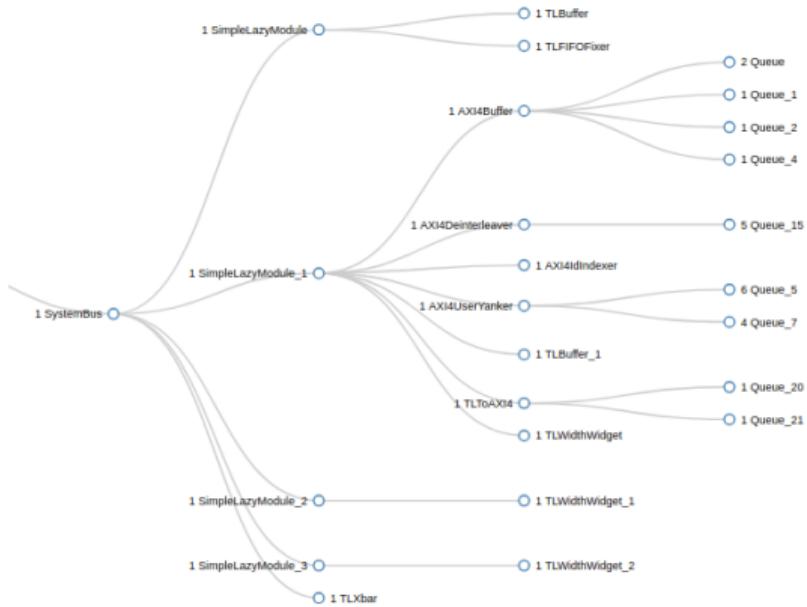
# Frontier: Parallelism

- BonnRoute: concurrency + shared memory model
- qrouter: none
- lsc: map + reduce

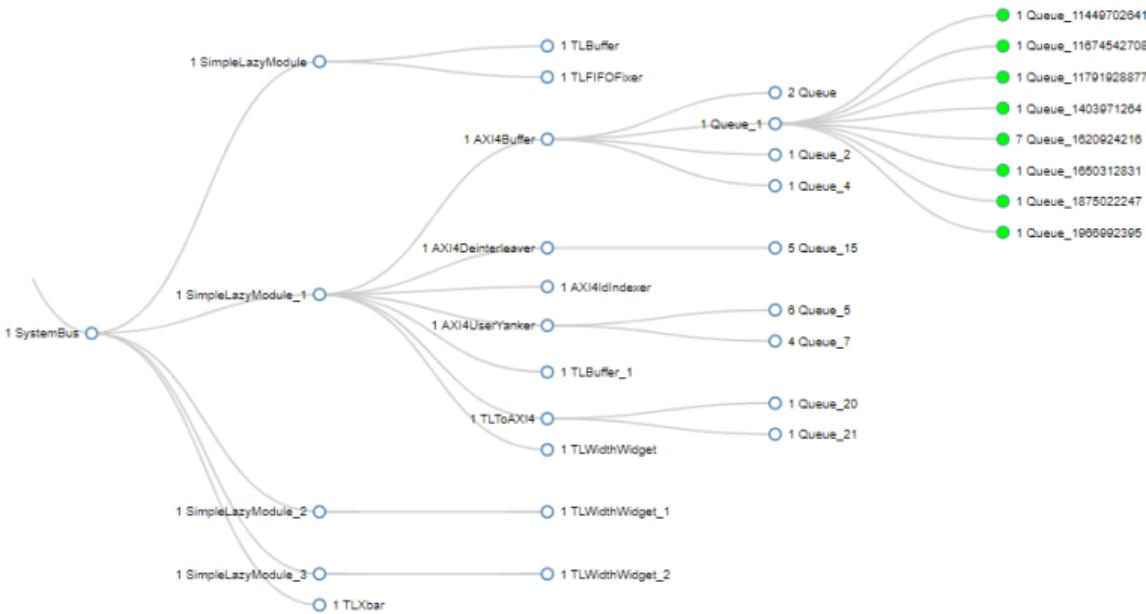
# Subcell hierarchies

- Explicit subcell hierarchies through high modularization
- Implicit subcell hierarchies through exlining
- Preserve hierarchy in compiler interfaces

# High modularization



# Exlining



# Unconstrained Small Unified Silicon Problem

- Components and nets → *rectilinear geometries*
- Components do not overlap
- Nets overlap with their pins on components

# Minimizing Goals

- Layout area
- Maximum wire length
- Via count
- Crossing number (computational)
- Wire jogs (minor)

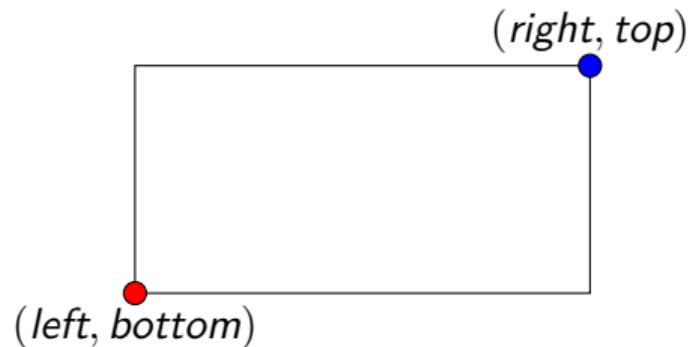
# Satisfiability Modulo Theories

- Optimization problems
- Abstraction from Boolean satisfiability
- Several solvers implement smtlib2
  - ABC from University of Berkeley
  - CVC4 from Stanford
  - Boolector from Johannes Kepler University
  - MathSAT from Fondazione Bruno Kessler and DISI-University of Trento
  - Yices from SRI
  - Z3 from Microsoft

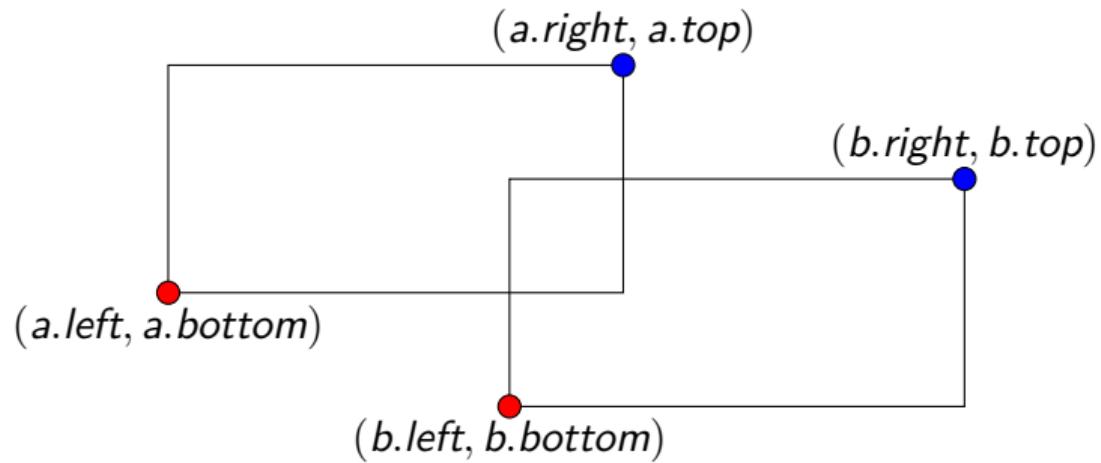
# Boolean Satisfiability

$$(\alpha_1 \vee \alpha_2 \vee \alpha_3) \wedge (\neg\alpha_4 \vee \alpha_5 \vee \alpha_6)$$

# Defining rectangular components



# Overlaps



# Reduction from SMT

$$a.right - a.left = dimension.x$$

$$b.right - b.left = dimension.x$$

$$a.top - a.bottom = dimension.y$$

$$b.top - b.bottom = dimension.y$$

$$b.left > a.right$$

$$\vee \quad a.left > b.right$$

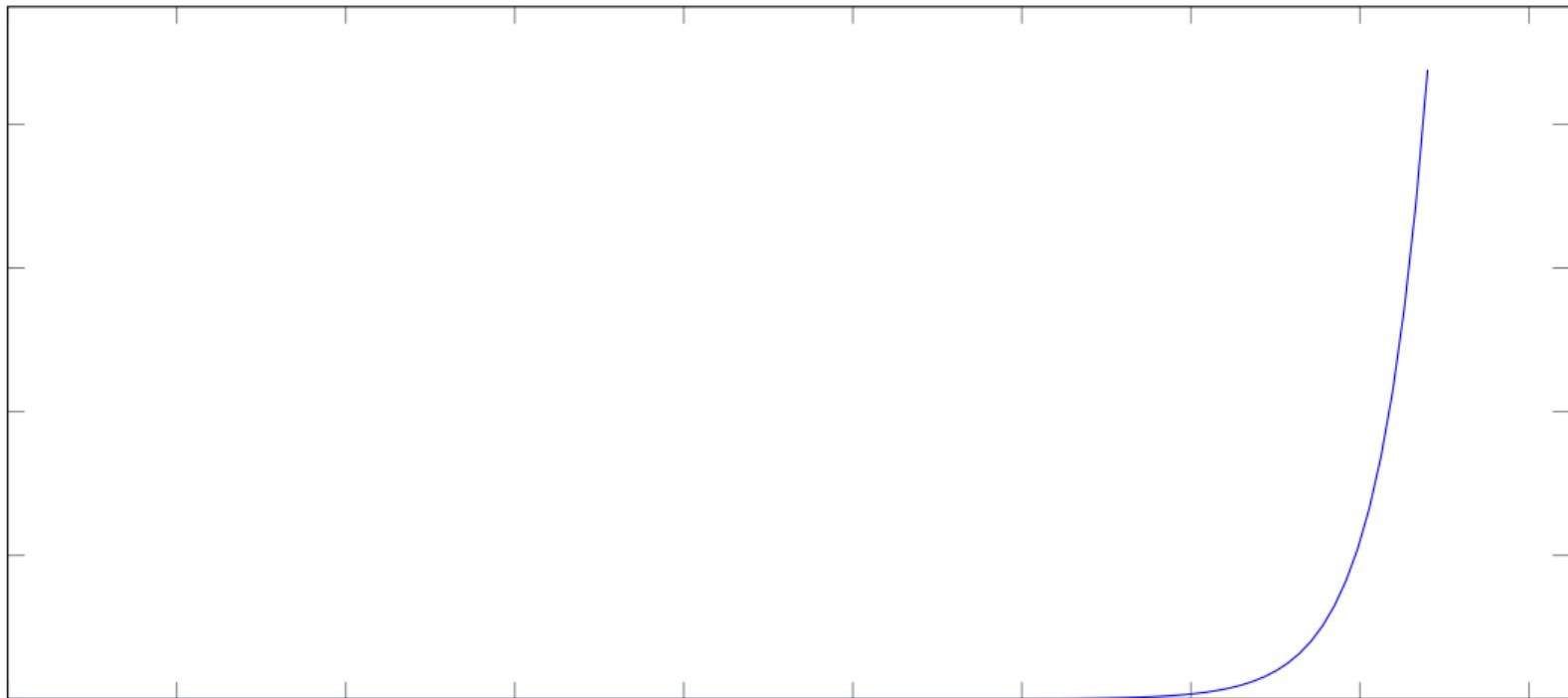
$$\vee a.bottom > b.top$$

$$\vee b.bottom > a.top$$

# Combining in the LSC Semigroup

overlaps + pin connect + arbitrary constraint

# Stay low



# Maximize yield

- Minimize area of a chip → *silicon compiler*
- Minimize physical errors → **silicon process**

# Libre Silicon Compiler

- <https://www.github.com/foshardware/lsc> (→ repository)

maintained by foshardware

# Why Hong Kong?

## History



# Why Hong Kong?



# Why Hong Kong?



RCL Semiconductors Limited  
興華半導體工業有限公司



# Why Hong Kong?

## Conclusion

- Advanced labs available for R&D
- Shenzhen and Hong Kong have fabs which are willing to introduce LibreSilicon
- Channels for easy export to Europe already established (One belt, one road)
- Climate is better
- Payment is better
- Food is better
- \* is better
- The sun shines brightest in the east ;-)

# Features

- MOSFETs
- LDMOSFETs (High voltage)
- BJTs
- Zener polysilicon diodes
- SONOS flash cells
- Polysilicon resistors
- Metal caps

# Process design considerations

- Should be portable
- Robust
- Low amount of layers
- KISS (Keep it simple and stupid)
- Avoid expensive machines
- Can be manufactured in a home lab

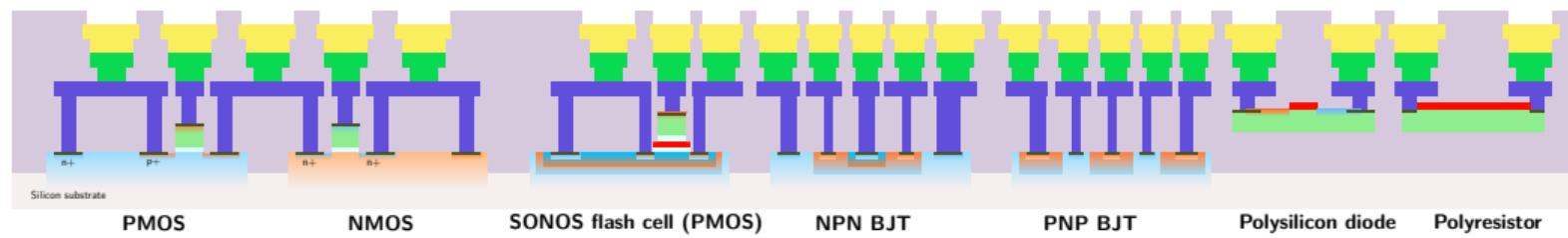
## Process design considerations



## Process design considerations



# Cross section



# PearlRiver (珠江芯片一号)

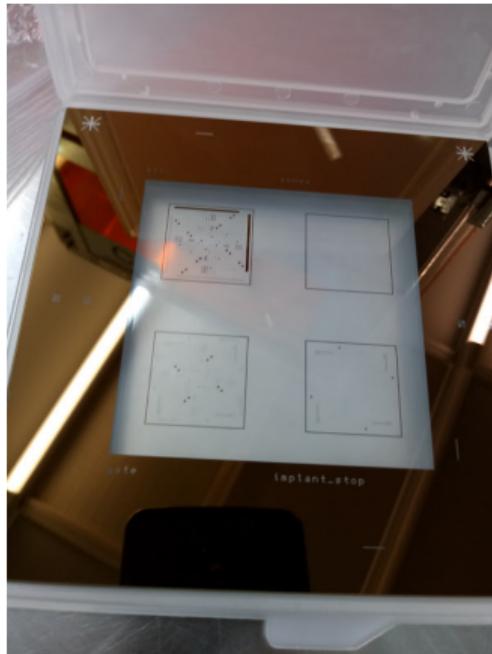


# PearlRiver (珠江芯片一号)

## Fulfills following functions:

- Debugging
- Calibration of new equipment to LibreSilicon
- Research of new features
- Syncing process features between fabs

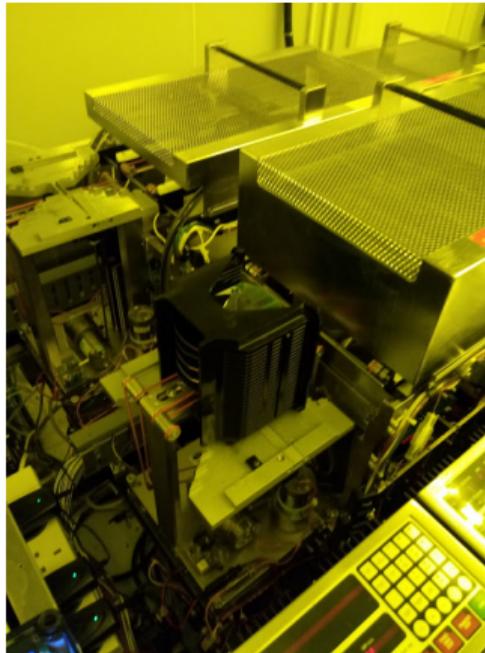
# Photomask



# Photomask

- Is stepper/aligner brand specific
- ASML stepper masks contain 4 layers each
- The NFF stepper has a reduction value of 5:1
- A 5 micron gate on the mask is 1 micron on the wafer

# Photo resist (HKUST)



# Photo resist (HKUST)

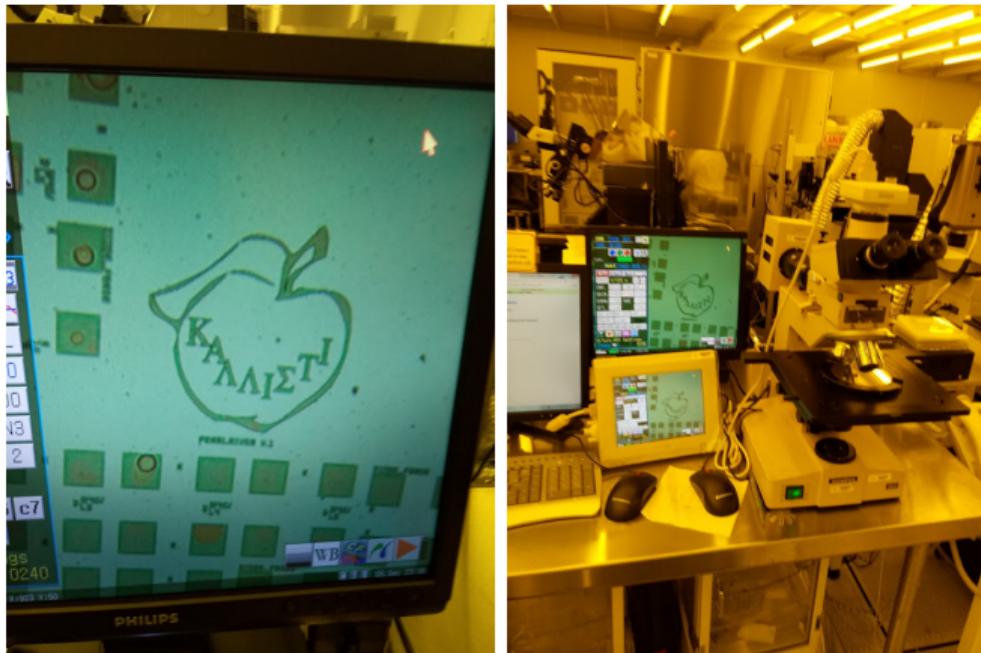
**Two types of photo resist:**

- FH 6400L (implantation)
- HPR 504 (normal etch)

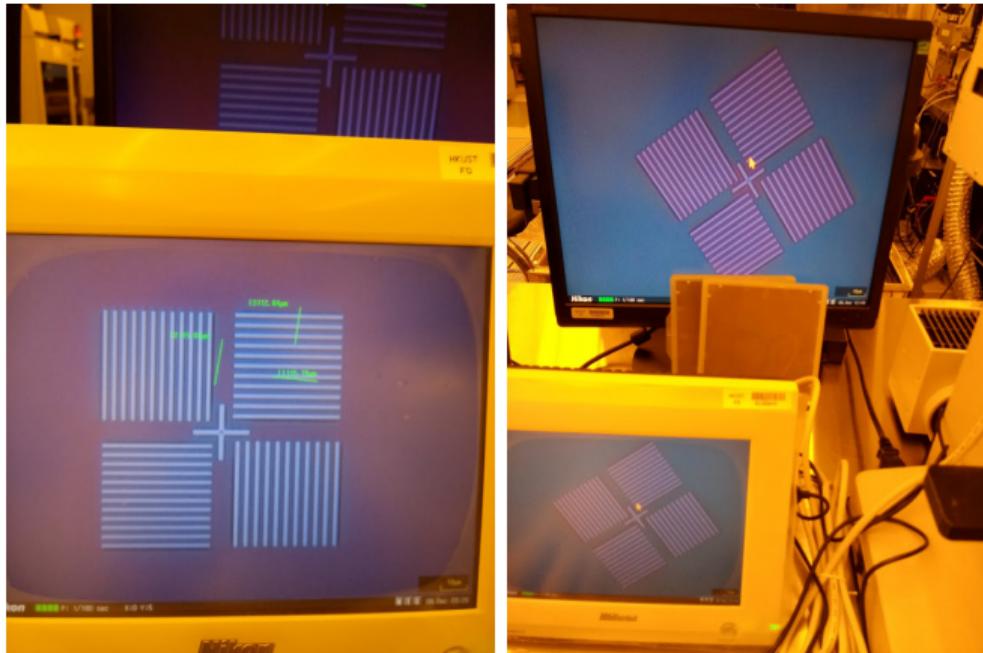
**Factors to consider:**

- Thickness of FH 6400L and implantation energy are interlinked
- Thickness of HPR 504 and etching time are interlinked (selectivity)

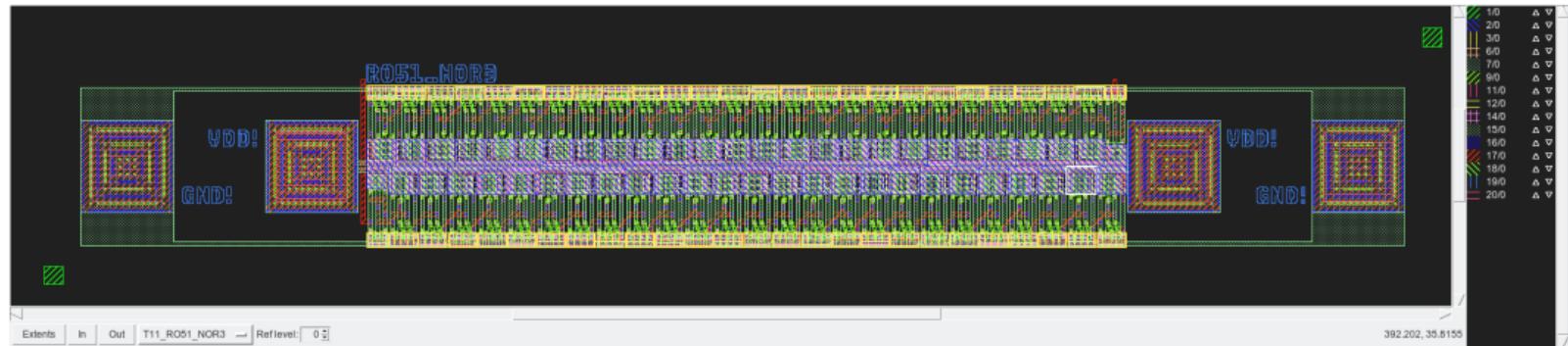
## After exposure



# Alignment

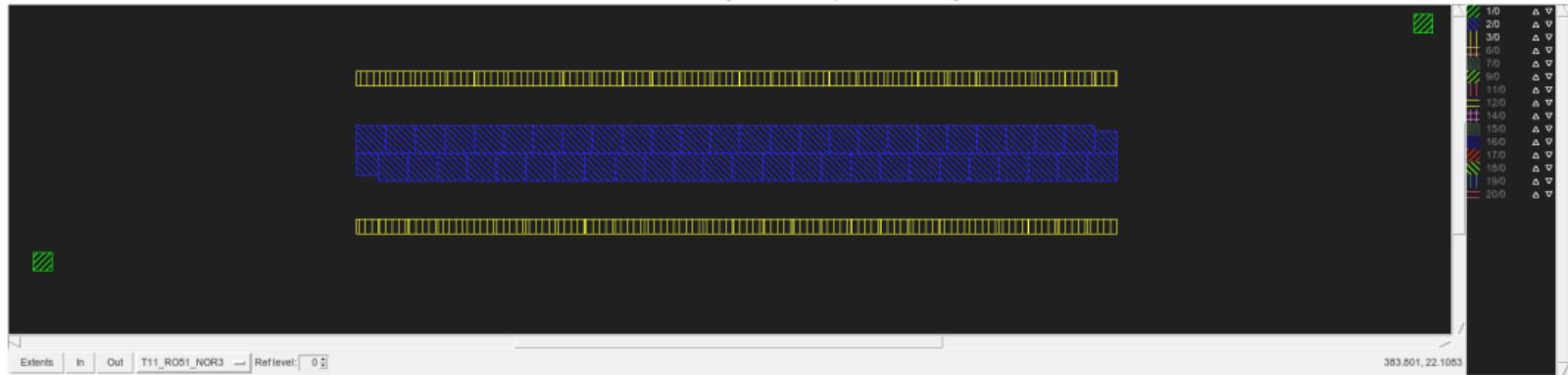


## Example: NOR3 ring oscillator



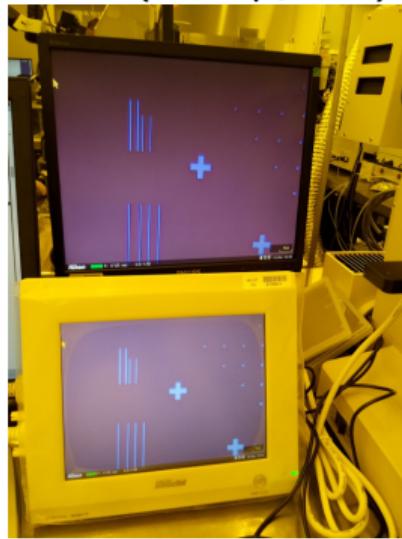
# Example: NOR3 ring oscillator

**Wells (nwell/pwell):**



## Example: NOR3 ring oscillator

Wells (nwell/pwell):



## Example: NOR3 ring oscillator

### **Recipe for nwell:**

- Coat with implant resist (soft bake 60 seconds, 110°C)
- Expose nwell-mask
- Puddle develop 69 seconds and hard bake 60 seconds at 120°C
- Implant Phosphorus,  $2.33 \times 10^{12} cm^{-2}$  @ 70keV
- Strip resist with plasma ash or 20 minutes in 120°C hot sulfuric acid

**Note:** Alternatively predisposition can be used

## Example: NOR3 ring oscillator

### Recipe for pwell:

- Coat with implant resist (soft bake 60 seconds, 110°C)
- Expose pwell-mask
- Puddle develop 69 seconds and hard bake 60 seconds at 120°C
- Implant Boron,  $1.93 \times 10^{12} cm^{-2}$  @ 40keV
- Strip resist with plasma ash or 20 minutes in 120°C hot sulfuric acid
- Diffuse both wells together for 4 hours at 1050°C in inert atmosphere ( $N_2$ )

**Note:** Alternatively predisposition can be used

# The Fick's equation

$$\frac{\partial N}{\partial t} = D \cdot \frac{\partial^2 N}{\partial x^2}$$

# The Fick's equation

$$x_I(t) = 2 \cdot \sqrt{D_e \cdot t} = 2 \cdot \sqrt{D_0 \cdot \exp\left(-\frac{E_a}{k \cdot T}\right) \cdot t}$$

| Element | $D_0 \left[ \frac{cm^2}{s} \right]$ | $E_a [eV]$ |
|---------|-------------------------------------|------------|
| P       | 10.50                               | 3.69       |
| As      | 0.32                                | 3.56       |
| Sb      | 5.60                                | 3.95       |
| B       | 10.50                               | 3.69       |
| Al      | 8.00                                | 3.47       |
| Ga      | 3.60                                | 3.51       |
| Cu      | 0.0025                              | 0.65       |

# The Fick's equation

$$N(x, t) = \frac{Q}{\sqrt{\pi \cdot D_e \cdot t}} \cdot \exp \left( \frac{-x^2}{4 \cdot D_e \cdot t} \right)$$

# The Fick's equation

For the concentration within the channel may  $x = 0$

This results in the concentration at the surface (around 100nm deep  $\ll 2$  microns)

$$N = \frac{Q}{\sqrt{\pi \cdot D_e \cdot t}}$$

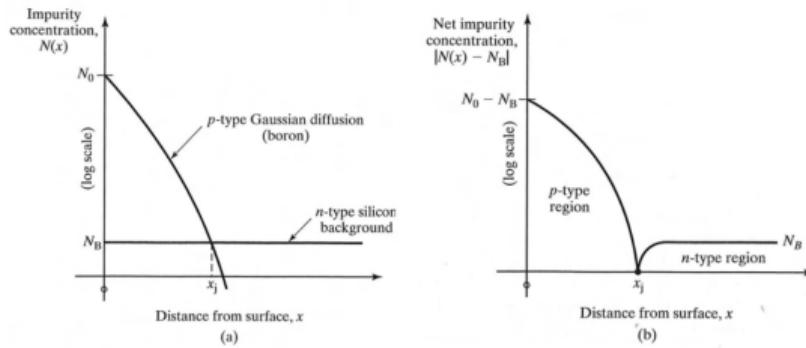
Or the implant dosage:

$$Q = N \cdot \sqrt{\pi \cdot D_e \cdot t}$$

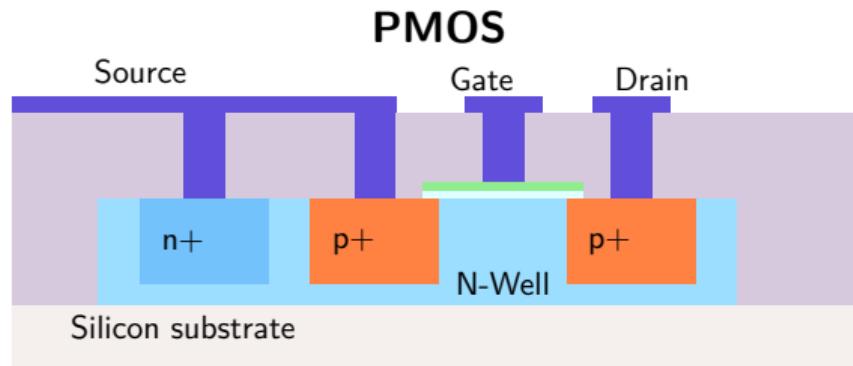
# The Fick's equation

Fixed source diffusion:

$$x_j = 2 \cdot \sqrt{D \cdot t \cdot \ln \left( \frac{N_0}{N_B} \right)}$$



# Threshold calculation



# Threshold calculation

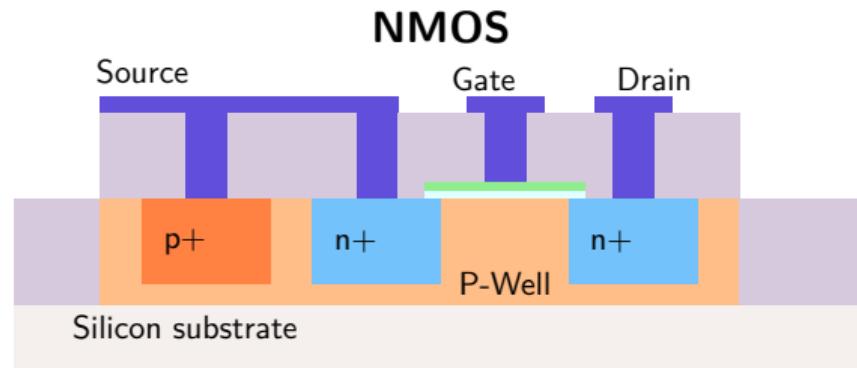
## PMOS

$$|\phi_F| = V_T \ln \frac{N_d}{n_i} \quad (1)$$

$$V_T = V_{FB} - 2 \cdot |\phi_F| - \frac{\sqrt{2 \cdot \epsilon_s \cdot q \cdot N_d \cdot (2 \cdot |\phi_F| - V_{SB})}}{C_{ox}} \quad (2)$$

$$V_{FB} = - \left( \frac{E_g}{2} - \phi_F \right) - \frac{Q_{SS}}{C_{ox}} \quad (3)$$

# Threshold calculation



# Threshold calculation

## NMOS

$$\phi_F = V_T \ln \frac{N_a}{n_i} \quad (4)$$

$$V_T = V_{FB} + 2 \cdot \phi_F + \frac{\sqrt{2 \cdot \epsilon_s \cdot q \cdot N_a \cdot (2 \cdot \phi_F + V_{SB})}}{C_{ox}} \quad (5)$$

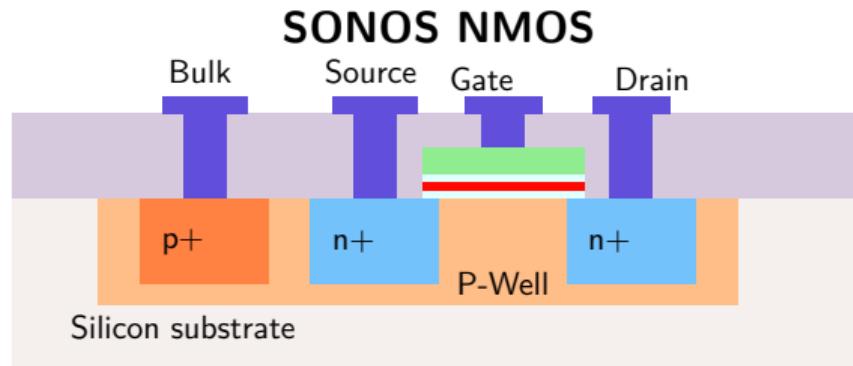
$$V_{FB} = - \left( \frac{E_g}{2} + \phi_F \right) - \frac{Q_{SS}}{C_{ox}} \quad (6)$$

# SONOS flash

## **Stands for**

**S**ilicon  
**O**xide  
**N**itride  
**O**xide  
**S**ilicon

# SONOS flash



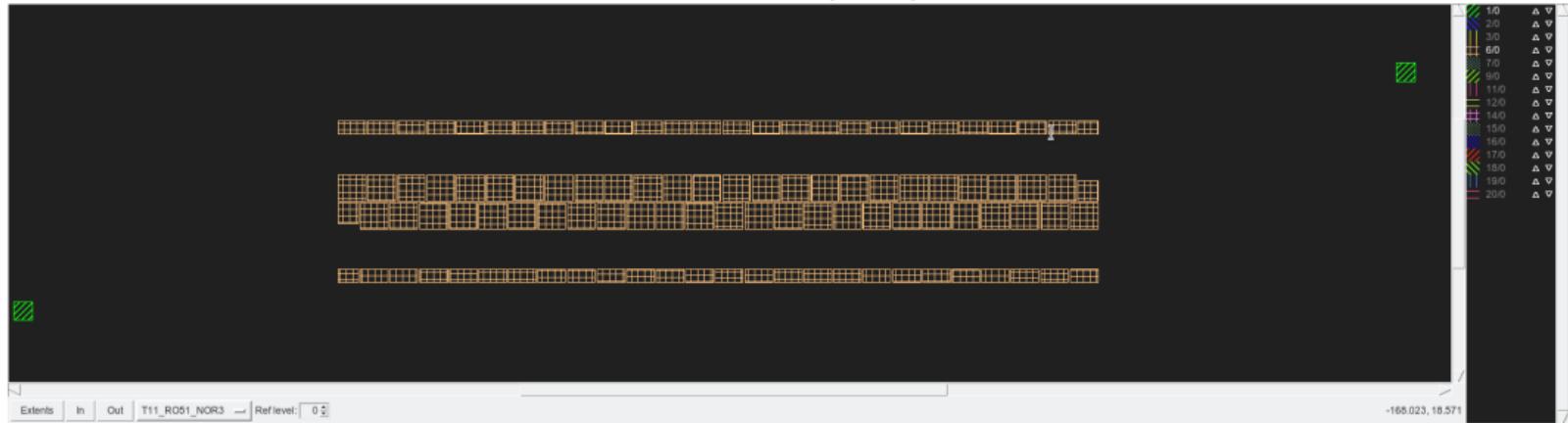
# SONOS flash

## SONOS NMOS

- Programming/Erasing happens by changing  $Q_{SS}$
- A variation of  $Q_{SS}$  shifts the threshold voltage
- $Q_{SS}$  can be changed by applying an enough high voltage between bulk and gate (approx. 20 V)
- High enough voltage tunnels electrons into the nitride
- Shifting the threshold voltage away from 0.8 V / -0.8 V makes it stay turned off when a "1" is applied

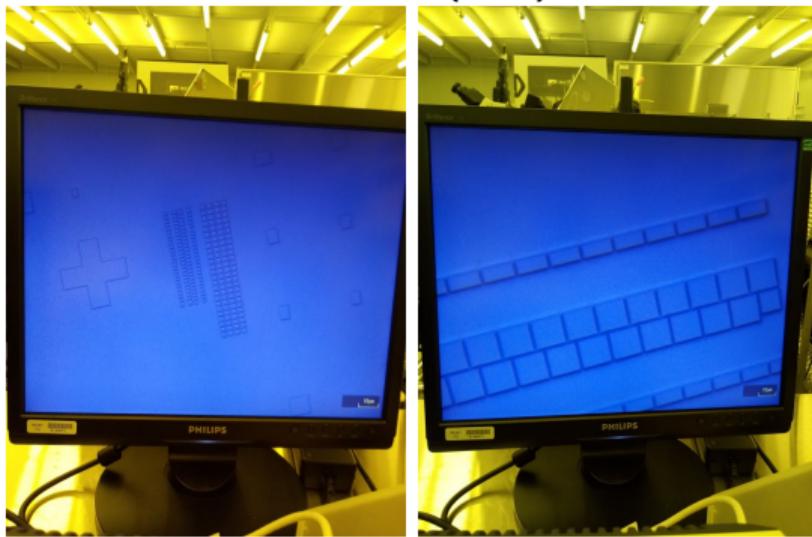
# Example: NOR3 ring oscillator

## Isolation (STI):



## Example: NOR3 ring oscillator

### Isolation (STI):



## Example: NOR3 ring oscillator

### Recipe for STI:

- **Dry**

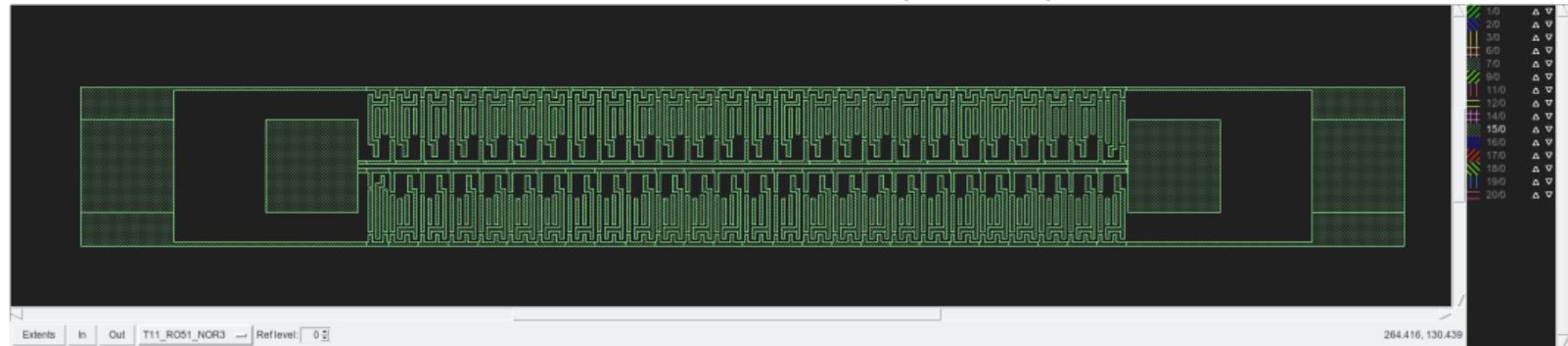
- Plasma etching recipes are machine specific
- Variate the cycles for your recipe to match 2 microns

- **Wet**

- Take TMAH:  $N(CH_3)_4^+ OH^-$  (Tetramethylammonium hydroxide)
- Dilute with deionized water with DI:TMAH (3:1)
- Heat TMAH (25%) to 80°C
- Dip wafer into the solution for around 6 minutes and 15 seconds (320nm/min, 2 microns)

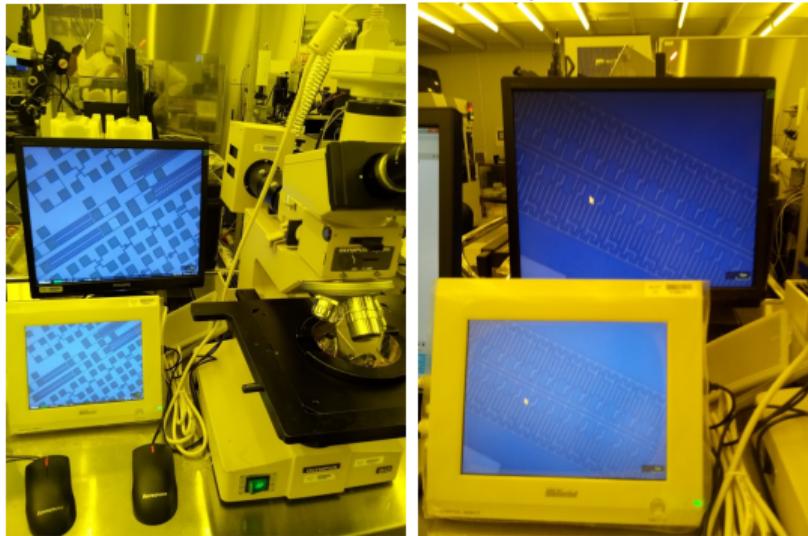
## Example: NOR3 ring oscillator

### Metal interconnect (metal1):



## Example: NOR3 ring oscillator

Metal interconnect (metal1):



## Example: NOR3 ring oscillator

### Recipe for metal interconnects:

- Make a vacuum (low pressure)
- Deposit 100nm Aluminum
- Deposit 30nm Titanium over the Aluminum
- Take out of vacuum
- Dip into HF:DI (1:10) water solution for a few seconds until Titanium is gone
- Dip into FeCl<sub>3</sub> or other suitable Aluminum etchant for around 30 seconds until Aluminum is gone

## Example: NOR3 ring oscillator

### **Passivation/Isolation materials**

- Low temperature oxide (LTO)
- Phosphosilicate glass (PSG)

**Can both be wet or dry etched**

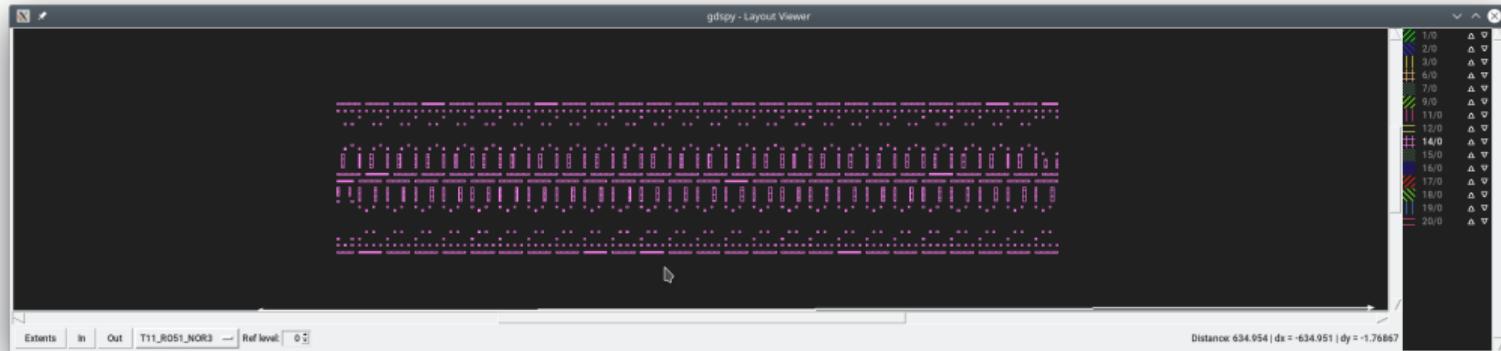
## Example: NOR3 ring oscillator

### Passivation/Isolation conceptional



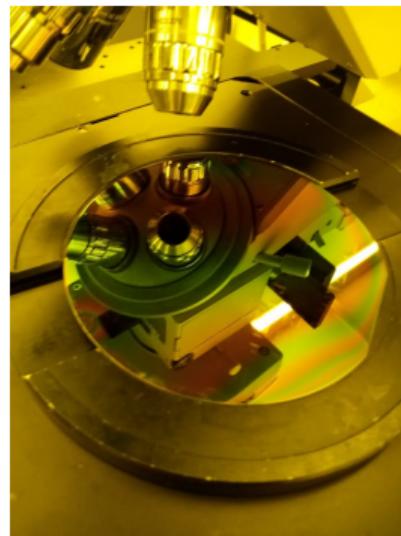
# Example: NOR3 ring oscillator

## Passivation/Isolation layout



# Example: NOR3 ring oscillator

## Passivation/Isolation in reality



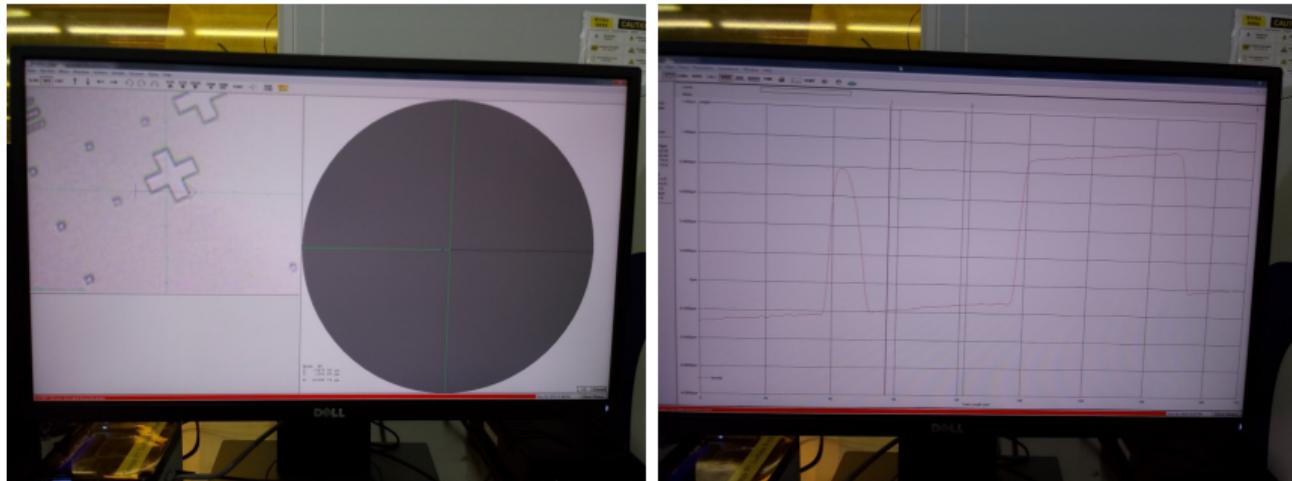
# Example: NOR3 ring oscillator

## Passivation/Isolation



# Example: NOR3 ring oscillator

## Passivation/Isolation



## Example: NOR3 ring oscillator

### Passivation/Isolation

- 1 micron is not enough
- 2 more microns need to be deposited
- Can be then be etched with BOE after exposure and development

# Process

- <https://www.github.com/libresilicon/process> (→ repository)

maintained by leviathanch

Next



**Victor and I**

Next

## We will

- ① Finish debugging all the features of PearlRiver (珠江芯片一号)
- ② Create preliminary Verilog and Spice models
- ③ Autogenerating standard cells with Popcorn scripts
- ④ Building ADCs/DACs and much more analog stuff
- ⑤ Build the North Point MCU (北角芯片)
- ⑥ Build the Sau Mau Ping SoC (秀茂坪芯片)

Next



**With lots of luck from the goddess**

# North Point (北角芯片)

## Features

- RISC-V core based on RV32EAC ISA
- Pin compatible to ATMega8
- Same features as ATMega8
- Tolerates up to 40V

# North Point (北角芯片)

## Survey

<https://survey.libresilicon.com>

# Future

- Equipment we are using is ready for going down to 500 nm ( $\rightarrow$  2 nodes shrinking)  
:-)

# Request for Chips

Which Free and Open Silicon do you like to see also?

- More Analog Stuff? (→ NE555?, uA741?)
- More Digital Stuff? (→ CD4000-series? LISP-CPU?)
- More Mixed-Signal Stuff? (→ SoC w/ Analog-Digital / Digital-Analog Converter?)
- ?

Keep on track, let us know your wish list!

**Mailing List:** <http://list.o2s.ch/mailman/listinfo/libre-silicon-devel>

Thanks!

非常感谢你们!  
**Thank you very much!**  
**Vielen herzlichen Dank!**

