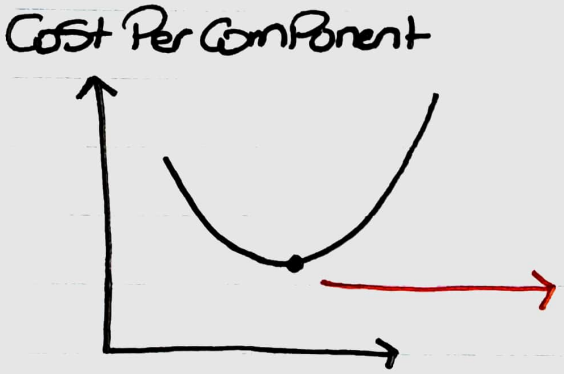
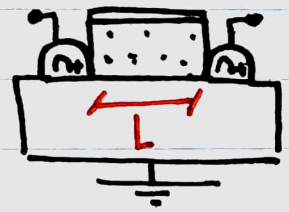
# VLSI
## Lecture 3

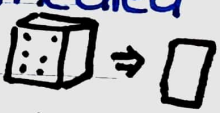



Starting at a certain point, cramming more components in the same IC does not help. ①

① (by making the channel length smaller beyond limits we get more defects)



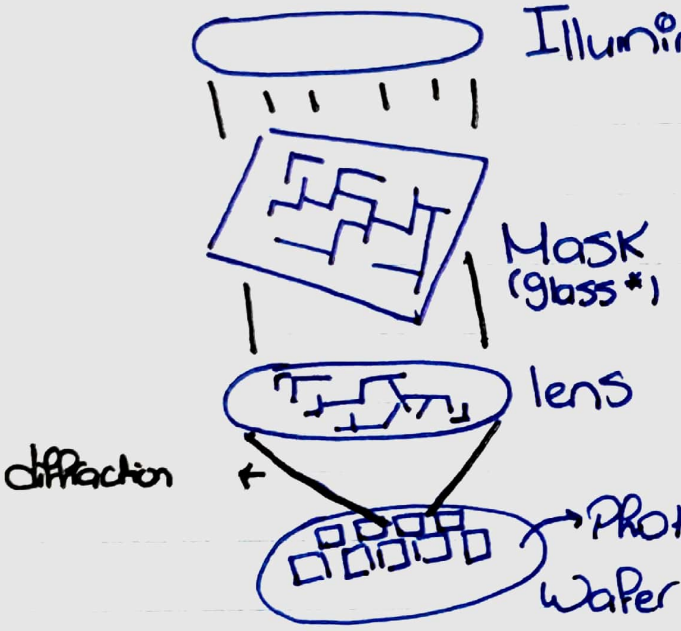

The length is the critical dimension (State-of-the-art tech uses 7nm and big corps are competing to bring it down to 3nm)

→ The driver for making the length smaller is economics since we can package more transistors in the same area at the same price. // Device Scaling 

- A serious bottleneck is the fabrication technology.

Any specific semiconductor manufacturing process is reffered to as a node.

# Lithography Process

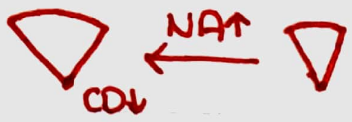



① ← GDSII File ← Design

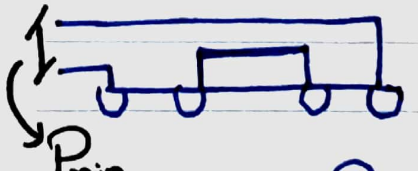
- mask is not just one layer.
→ $2M to make then $1.5K per wafer.

## Critical dimension

$CD = P_{min}/2 = K_1 \lambda / NA$ → Numerical aperture (lens design)

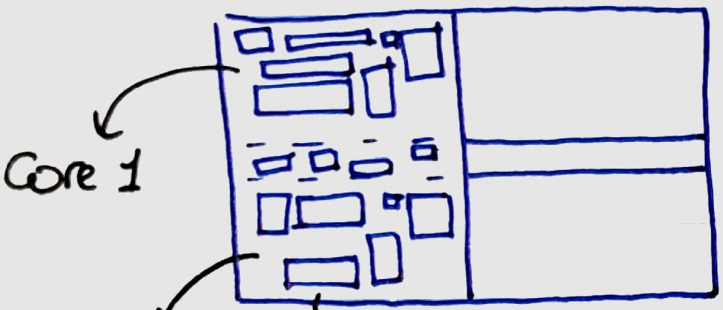



- Minimum Pitch (min distance between metals)
- Wavelength of light used (Physics/Optics) } light → U.V. → Extreme U.V. → X-ray
  - trying to get there but it's hard.
- Process Factor (due to Physics/Chemistry)
  → e.g. using a different material from air between the layers




① → The Process isn't very flexible & is a bottleneck to CD.

Intel Pentium II (duo*)




↳ looks like a grid (it's a cache (memory) this is why the Pattern is fixed)

//Pseudo colors } Feel random

//Slide 53 Skipped.

Moore's law over 10 years (design tools)

1970 manual design (by hand*)

1980 a library of known gates was made
(hardware that can be cut & Pasted by hand*)
//Std. Cells • Productivity: x → 100x

1990 design through register transfer level (RTL)
(Formulates the design in terms of registers)
e.g. design through VHDL.
• IP reuse of RTL designs (code) made sense

2000 IP based design & High level Synthesis

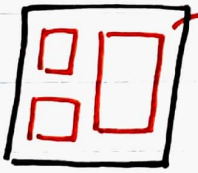
writing code that only expresses the behaviour (e.g. in C/SystemC)

→ Still exists but not so mature

→ Mentor Graphics CatoPult

2010 Platform based design & System level Synthesis

→ like FPGA but way more advanced



→ Reconfigurable building blocks

// Can build an entire System

- Comes with tools to help go from require-ments to design. // C Code*

* Every 10 years Productivity was multiplied by 100

Architectures // besides other info in the slides

SSI/MSI → Controller → Algorithm → System → Networks

Design Element

Sub-System ← IP ← Arithm.

Polygons → St.d Cells → Reg.

On a chip

// that weren't mentioned in the lec.

Due to how much the mask costs, verification is so important (which is another bottleneck)

70% Design

30% Verification (Simulation)

- Too many scenarios to test! (Dynamic verification)

Can also be emulation (on an FPGA) // limited (big design → Partitioning, Frequency limits, ...)

a 3rd option

Formal Verification

- Confirm mathematically that requirements ≡ design

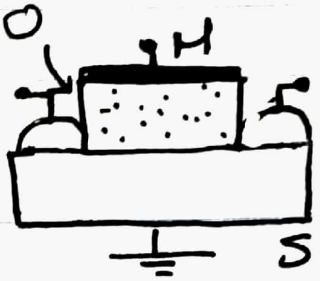
// Trend in industry uses logic, State machines, ...

↳ req. definition is a Problem (given in human language when needed in a Programming language) (math*)

# CMOS Inverter

Complementary (of the gate)
- PMOS
- NMOS

Metal Oxide Semiconductor (of the transistor)
- Nowadays Poly Silicon




# De facto Standard in tech. (CMOS rules)

why?

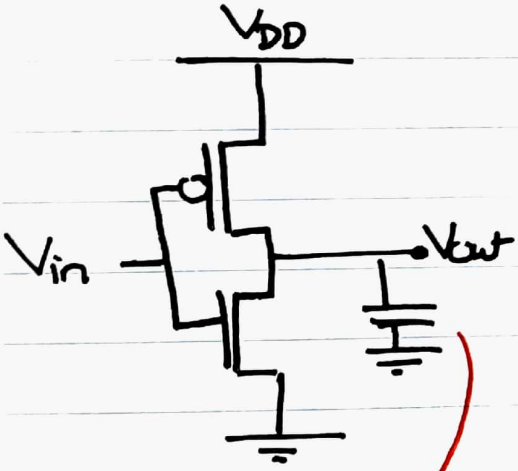
- Ratioless
- Full rail-to-rail swing

$V_{in} = 0 \quad \therefore V_{out} = V_{DD}$

$V_{in} = V_{DD} \quad \therefore V_{out} = 0$

→ helps with noise immunity (high noise margin)

→ tech.s with more limited swings suffer when cascading (0→4.7→0.5 3←4.2← ?) //drive heavy lo

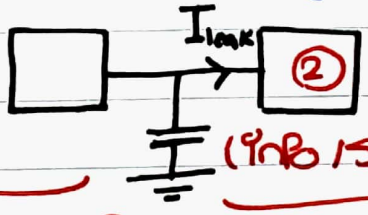



Parasitic CaP. (due to driven load & CaP.s in NMOS/PMOS)

- The CaP. is what stores 0 or 1.

→ $R_{on}$
- Zout is low → CaP charges/discharges quickly (low delay) //eq

Zin is high




high Zin @ ②
- Fast charging
- No leakage

low Zin @ ②
- Slow charging
- leakage (Charge high $C_{in_2}$)

*gates drive each other with current

- No Static Power dissipation (In steady state (0 or 1) either PMOS or NMOS is OFF → once output settles $V_{ds} = 0$ and no power is consumed)

// Other families do suffer from $P_{static}$.

- Delay depends on output impedance and load capacitance.
  - ↳ $R_{on}$

Notice that with scaling → Area ↓, Cap. ↓ } Delay ↓

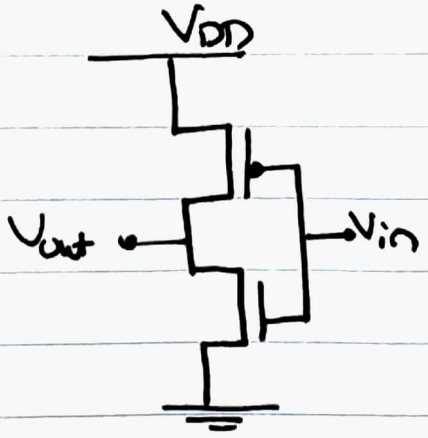
→ Metals are closer, Cap ↑ } Delay ↑

// trade-off

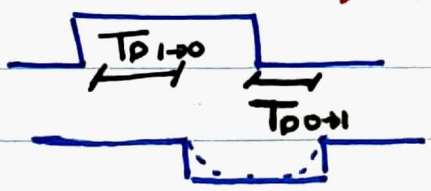
Any CMOS gate is "intrinsically an inverter"

as PMOS on ↔ low input ↔ Pull up output
NMOS on ↔ high input ↔ Pulldown output

- Design buffer? At least two ▷o s
  AND? Meant ▷o AND ?!

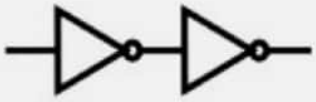


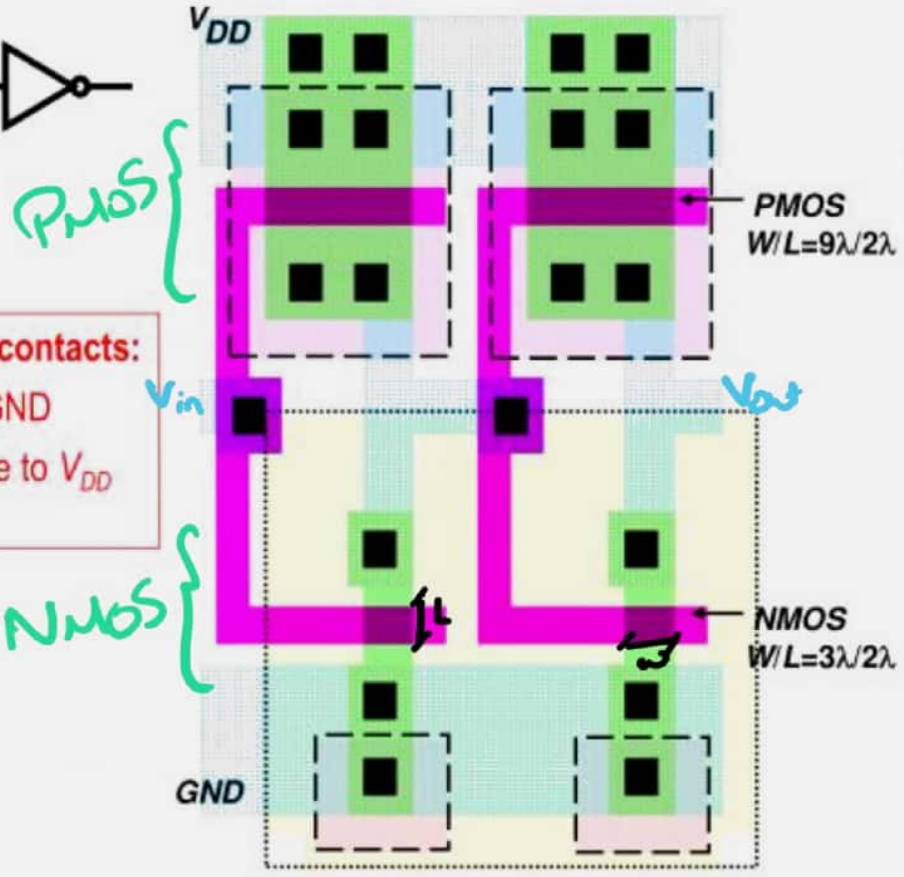

← To balance the circuit & thus the delay
PMOS has 2x width ($M_n \simeq 2 M_p$)
↳ like 9P two are in //.




$T_{P_{1\to0}} = T_{P_{0\to1}} = T_{P_{worst\ case}}$

# CMOS Inverter Layout (simplified)






**Note body contacts:**
- p-well to GND
- n-substrate to $V_{DD}$

**P-well mask** (dark field)

**Active** (clear field) ①

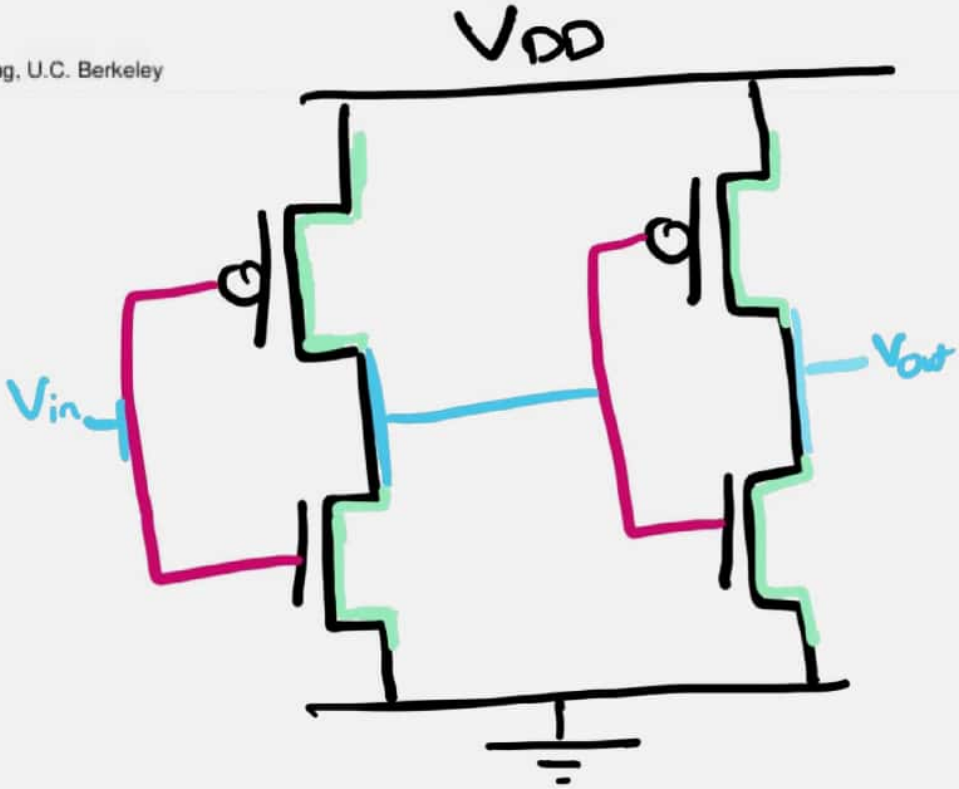
**Gate** (clear field)

**Select mask** (dark field & clear field) ②

**Contact** (dark field) ③

**Metal** (clear field) ④

rofessor N Cheung, U.C. Berkeley



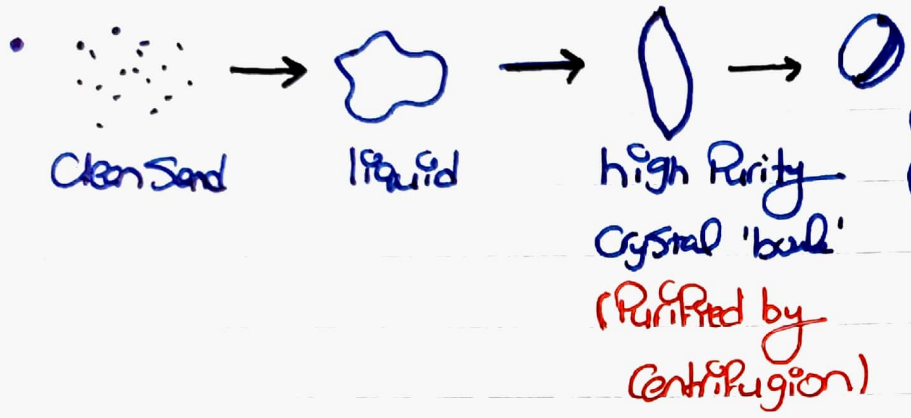

① ion diffusion layer (n+ or P+)

② decides whether its P+ or n+

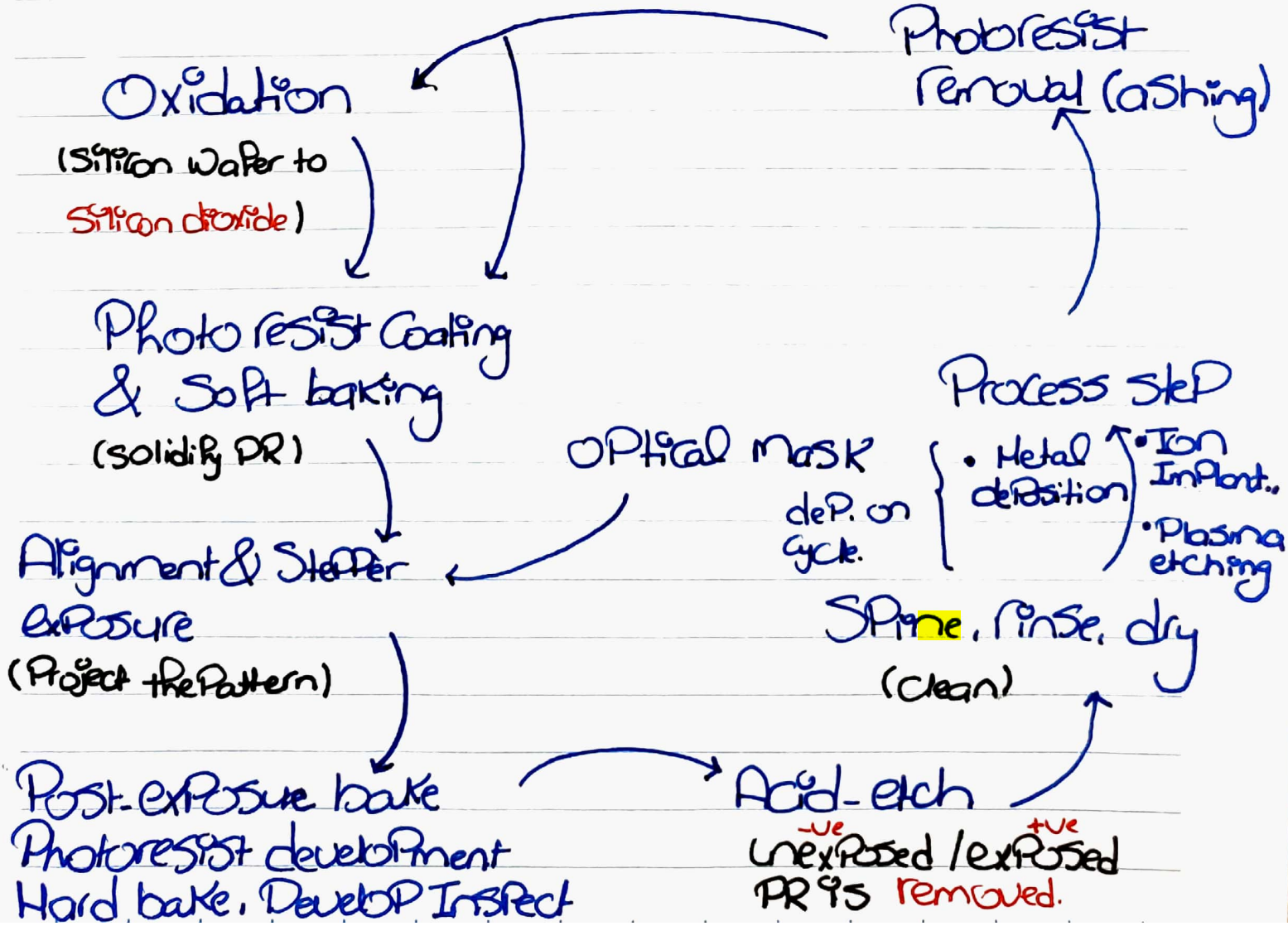
③ Contact/via help connecting diff. layers.

④ metal layer for connections.

# Growing the Silicon Ignot




# Photolithographic Process



Slide 64)


Slide 65)

→ This time its negative etching

* Doping: Diffusion and ion implantation

→ Heat/Pressure + Boron gas

→ Accelerate Boron via an electric field

boron penetrates diffusion occurs within the semiconductor

- Etching helps create chip features by selectively removing material added during deposition.

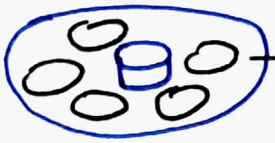
Deposition
- Chemical vapor deposition
- Chemical deposition
- Sputtering AL

} e.g. to create the Polysilicon layer

↳ بخ

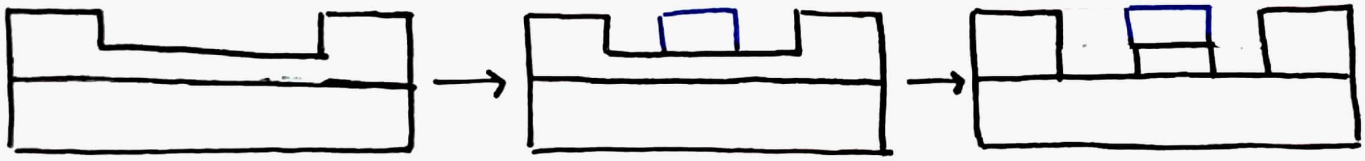
Etching
- wet etching
- dry (Plasma) etching

Polishing



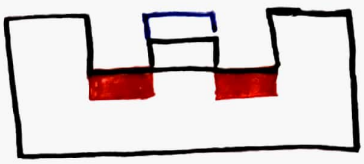
→ helps attain a super-flat wafer.

Doping Source & Drain? who needs a mask!

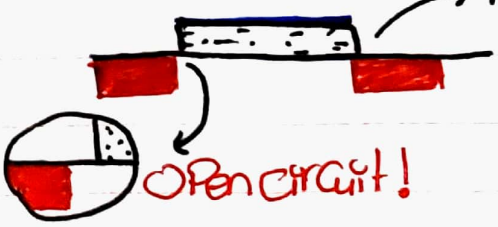


- Create thin oxide in the active region (source, channel drain) elsewhere it can be thick.
- Deposit Polysilicon (gate)
- etch thin oxide from active regions → Polysilicon (gate) acts as our mask!

- Now dope the source & drain

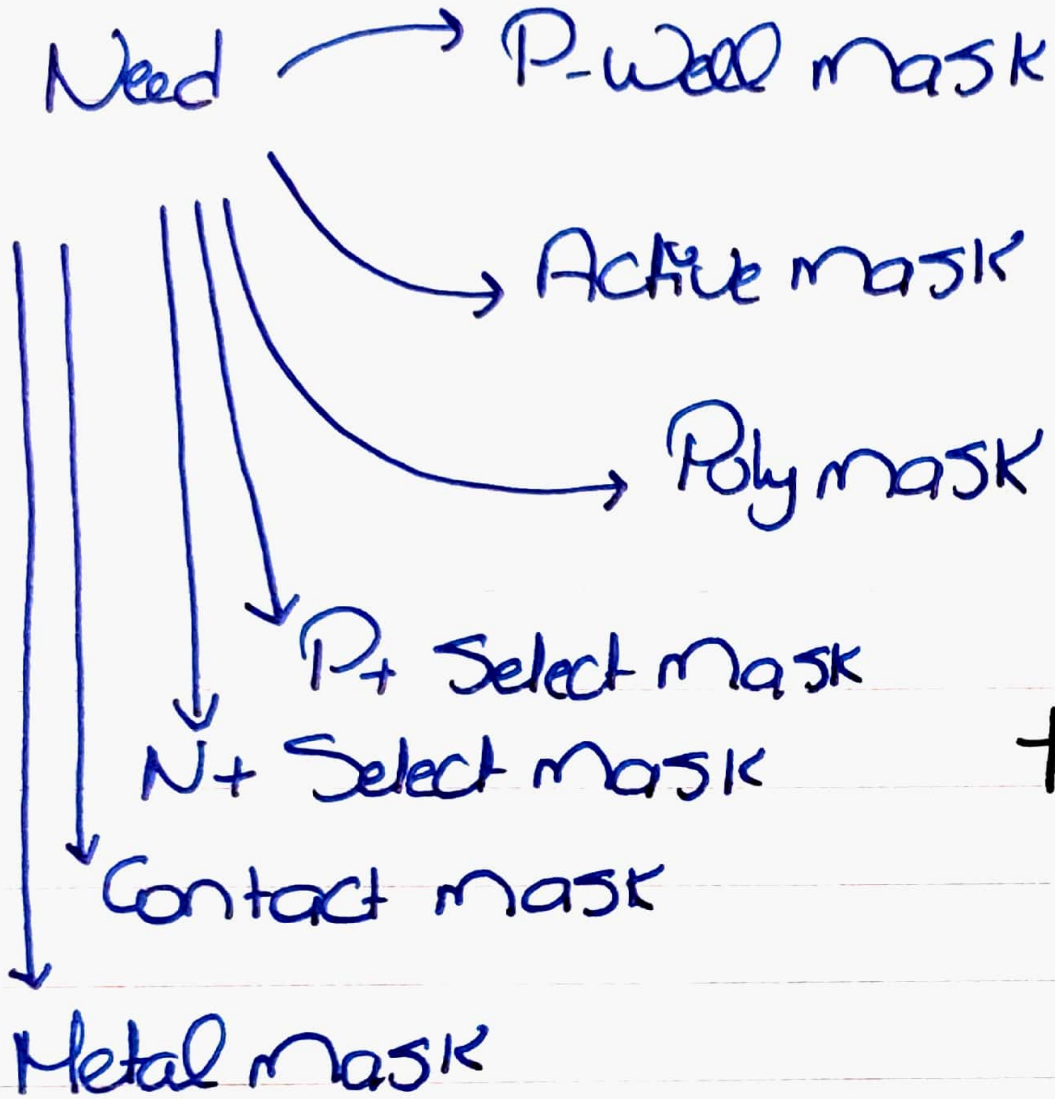


If we start with doping (diffusion first)




} Consequences of imperfect alignment.

# CMOS Inverter Process




*each is illustrated on the slides.

. Remember the CMOS inverter layout few pages ago?

thank you <3.