



LAYOUT AND MOS TRANSISTORS INTRO

James E. Stine, Jr.

Edward Joullian Endowed Chair in Engineering

Oklahoma State University

Electrical and Computer Engineering Department

Stillwater, OK 74078 USA

james.stine@okstate.edu

Errors

- It is no problem to help you with problems.
 - However, if I solve your problem, you will not learn anything.
- Please beware that its very difficult to debug over Email.
 - If you are remote or distance learning, call me if there is a specific problem.
 - If you feel uncomfortable calling me, then document your problem on slack step by step.
 - Even if you call, its sometimes easier to also document your problem and then call me.
 - For on-campus students, please stop by and show me the problem – I will do my best to help.
- Don't give up – learn what the problem is, so you don't repeat it.
 - Do not be eager to just have your problem solved!
- Please do not forget to submit homework!
- **Suggestion:** sometimes taking a 5–10-minute break on something can make night and day for any problem.

Simplified View of MOSFET

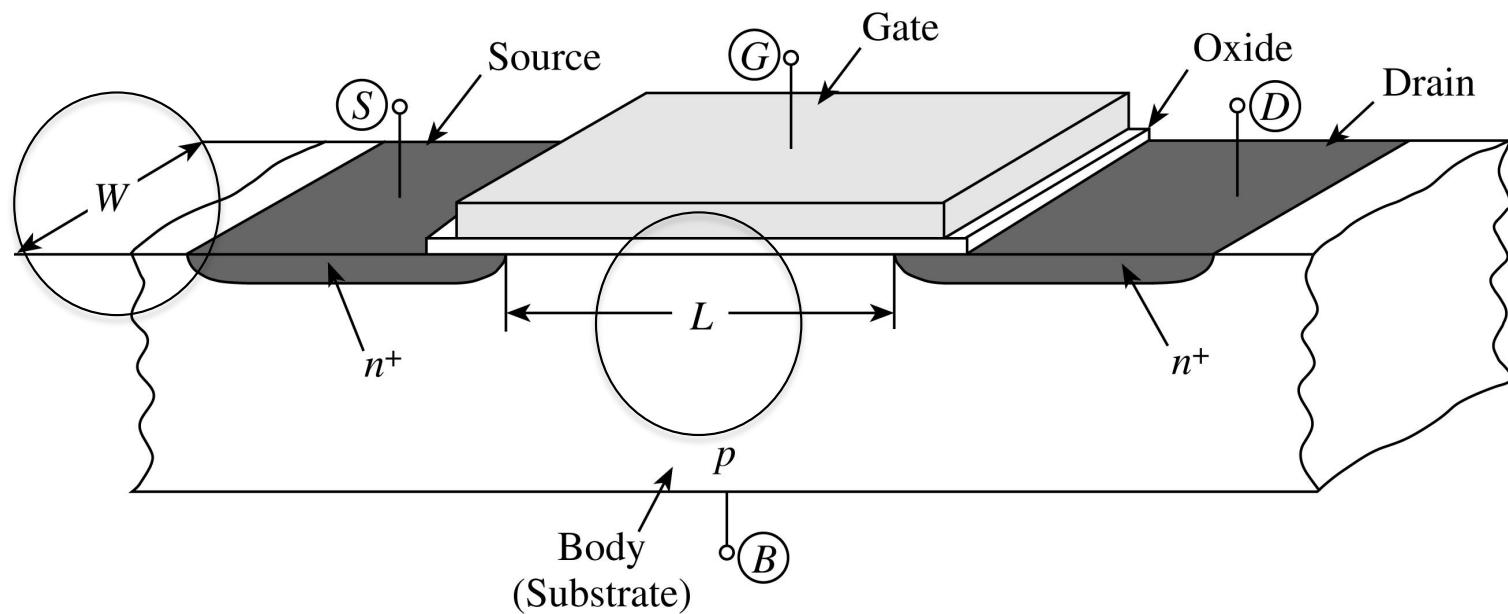
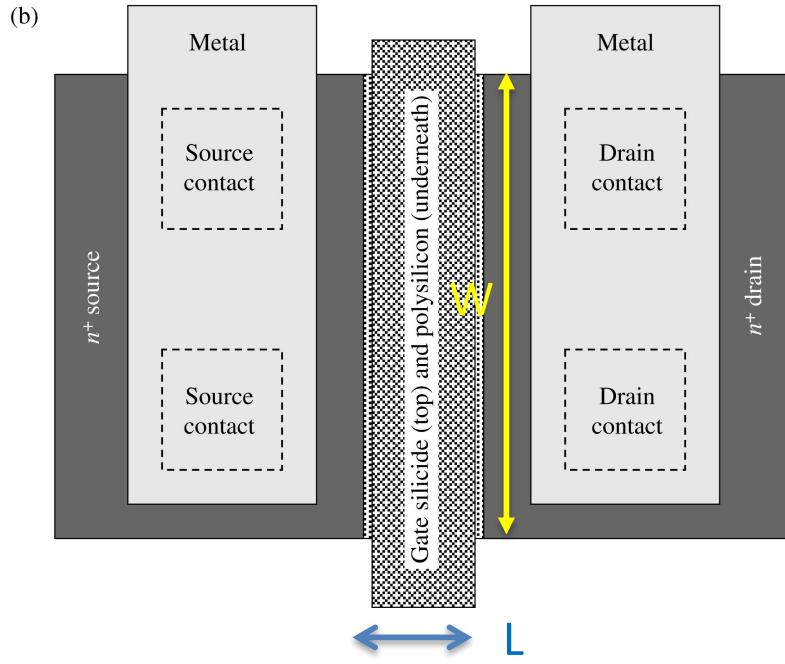


FIGURE 1.30 Simplified structure of an n-channel MOS transistor.

Two key design parameters : W and L

Layout



- Layout is a two dimensional structure to represent the actual layout representation.
- Each color represents a layer in 3 dimensions.
- Its up to you to visualize the layout to organize it correctly.
- However, it was the job of the Computer Engineer to program the designs into actual representable steps done by a computer.
 - Why?

Feature size or Length is quality factor on actual transistor design, since it quantifies the process in terms of a single element!

<http://smithsonianchips.si.edu/>

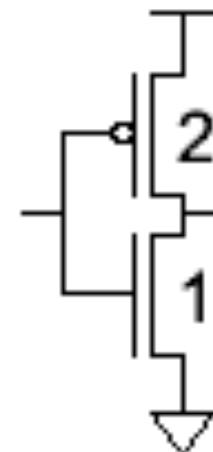
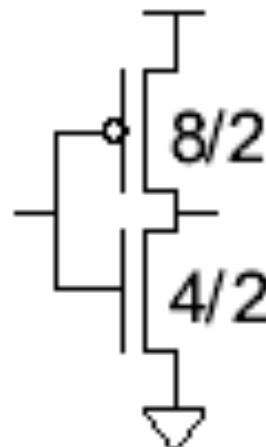
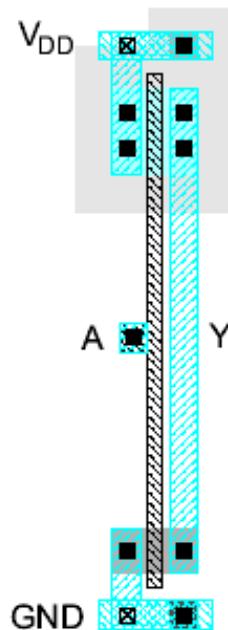
September 12, 2018



What does this mean?

Inverter Layout

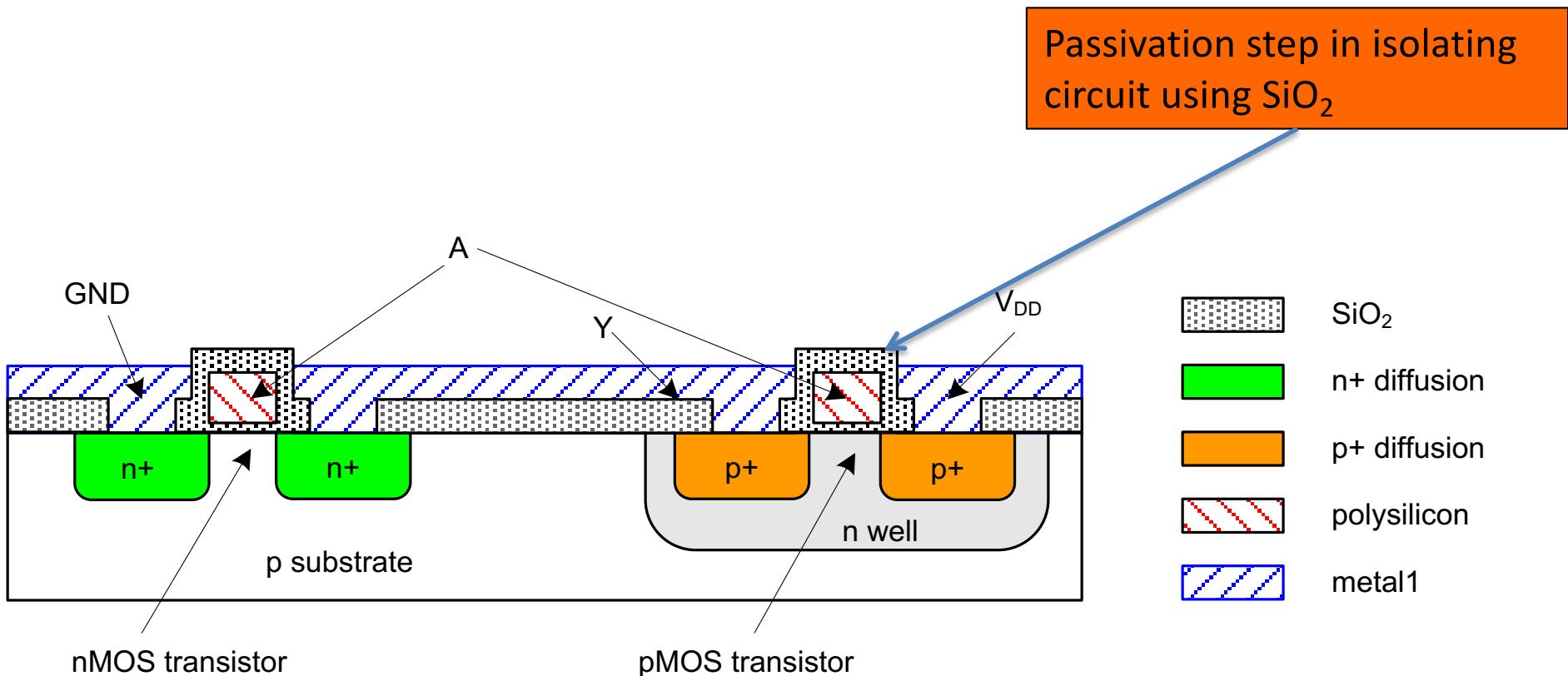
- Transistor dimensions specified as Width / Length
 - Minimum size is $4\lambda / 2\lambda$, sometimes called 1 unit
 - In $f = 0.6 \mu\text{m}$ process, this is $1.2 \mu\text{m}$ wide, $0.6 \mu\text{m}$ long



$\lambda = 0.3\mu\text{m}$

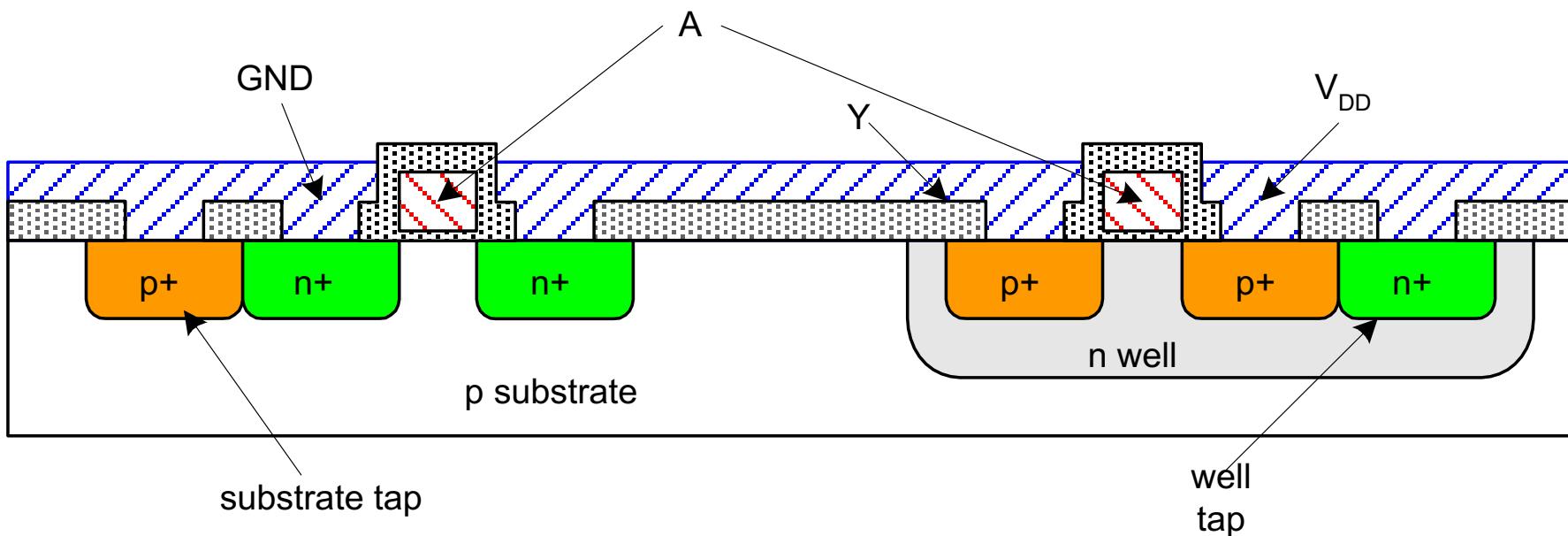
Inverter Cross-section

- Typically use p-type substrate for nMOS transistors
- Requires n-well for body of pMOS transistors



Well and Substrate Taps

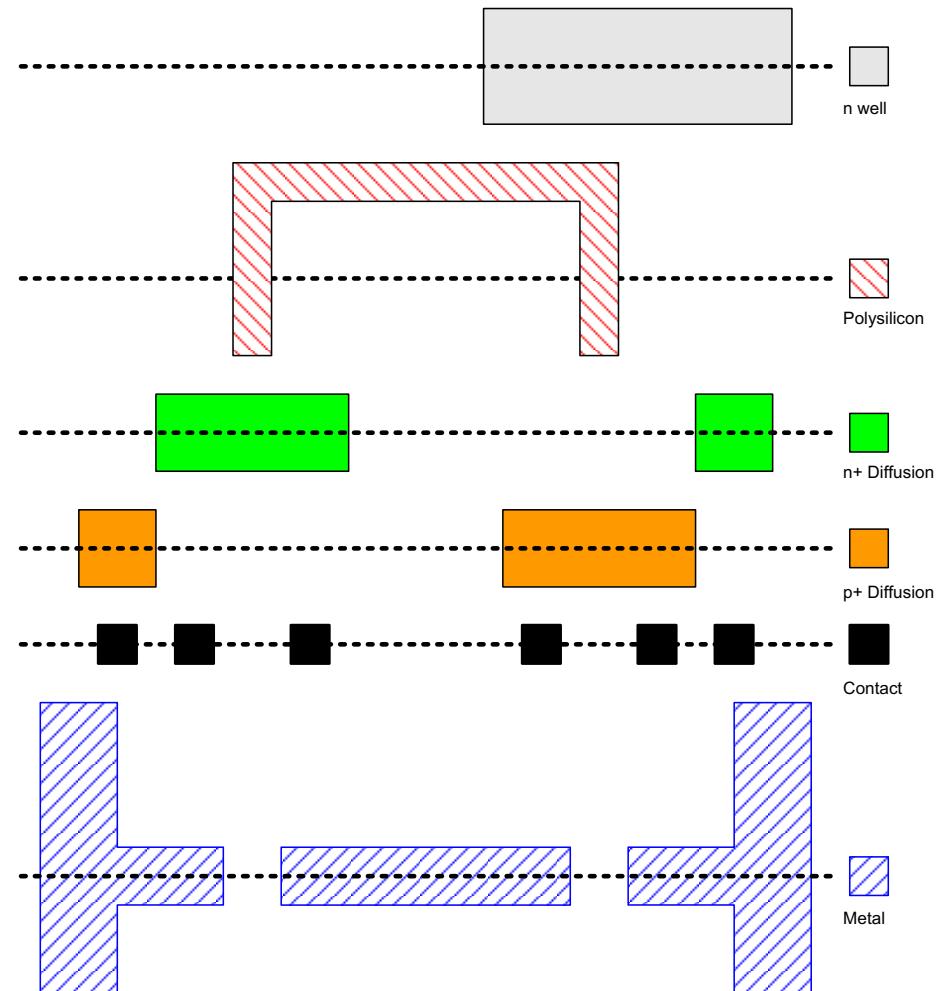
- Substrate must be tied to GND and n-well to V_{DD}
- Metal to lightly-doped semiconductor forms poor connection called Schottky Diode
- Use heavily doped well and substrate contacts / taps (also called tubs)



Might be different for different technologies (e.g., twin-tub designs)

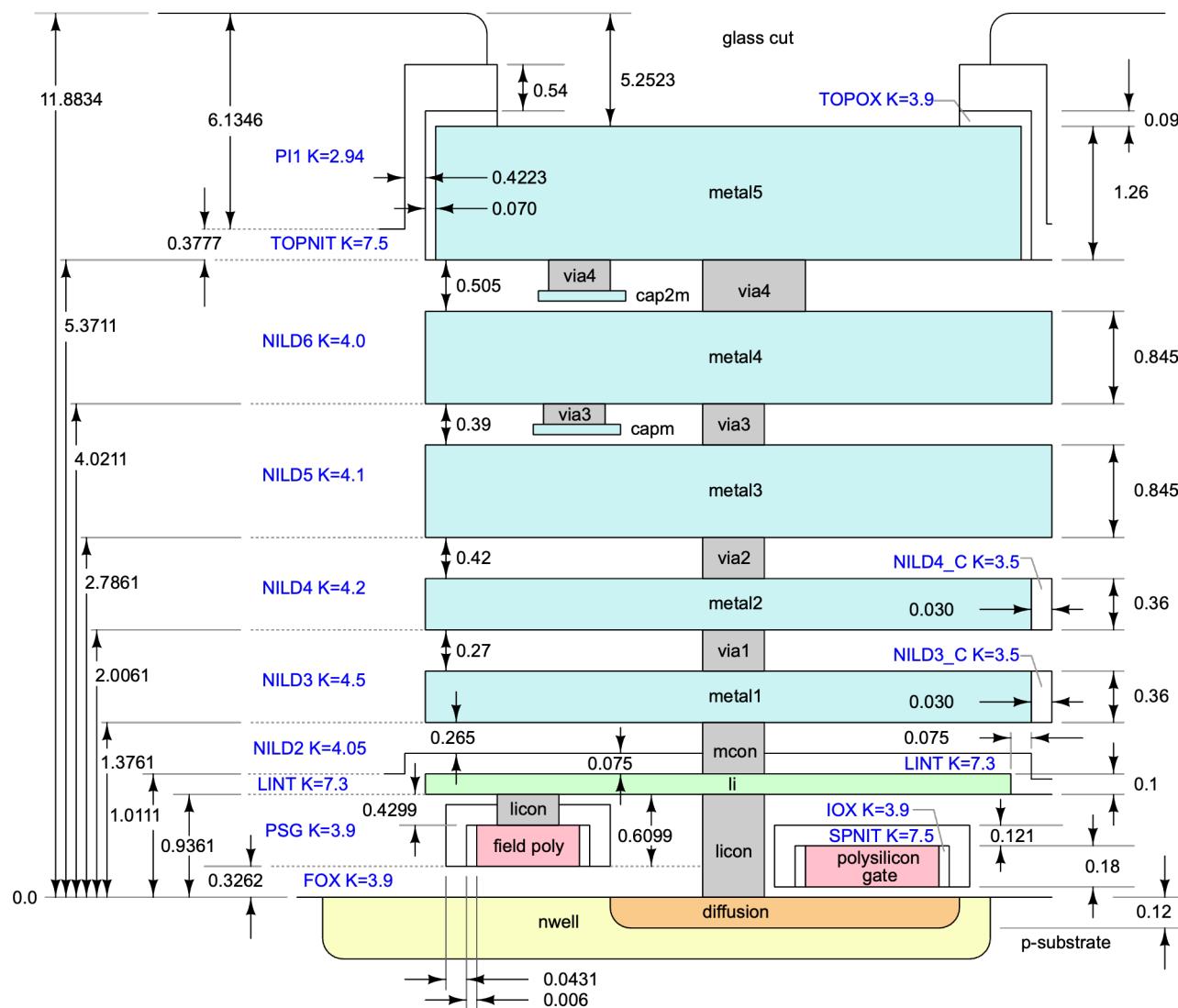
Detailed Mask Views

- Six masks
 - n-well
 - Polysilicon
 - n+ diffusion
 - p+ diffusion
 - Contact
 - Metal



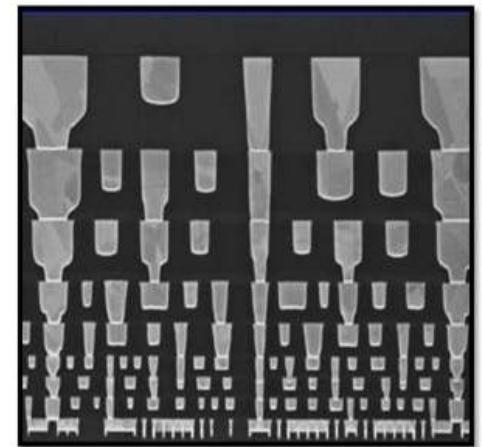
SKY130 Layers

(Diagram not to scale!)



Metal Stack

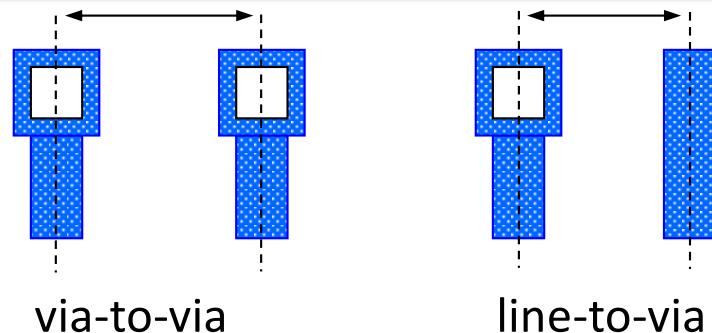
- For SKY130, we have five metal layers in our “stack” (not including poly and locali)
 - Metal 1 : m1
 - Metal 2 : m2
 - Metal 3 : m3
 - Metal 4 : m4
 - Metal 5 : m5
- Polysilicon or poly can be used for routing, but has higher resistance, capacitance, so should be limited in its use.
 - poly is on a lower level than m1
- Local interconnect (li) can be used too but its highly resistive.
- Smaller technologies have many layers of metal.
 - Companies may charge different rates for different stacks to help customers afford designs.
- The art of layout usually involves efficiently using metal or interconnect and this is why *pitch* is important! (**STICK DIAGRAMS!!!**)



Source : Intel

Track pitch and direction

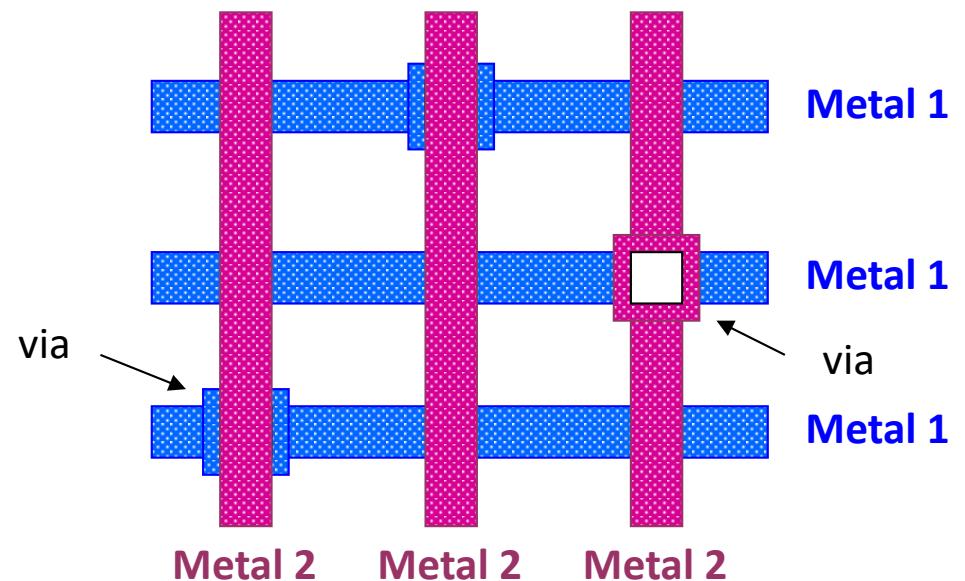
- The pitch at which wires are spaced defines the size of a routing track.
- Can be defined either via-to-via or line-to-via.
- To simplify P&R, metal layers are used in alternating directions.
 - Odd layers for horizontal routing and even layers for vertical routing (HVH).
 - On more advanced tech nodes, it is common to see both of the first two metal layers run horizontally.



via-to-via

line-to-via

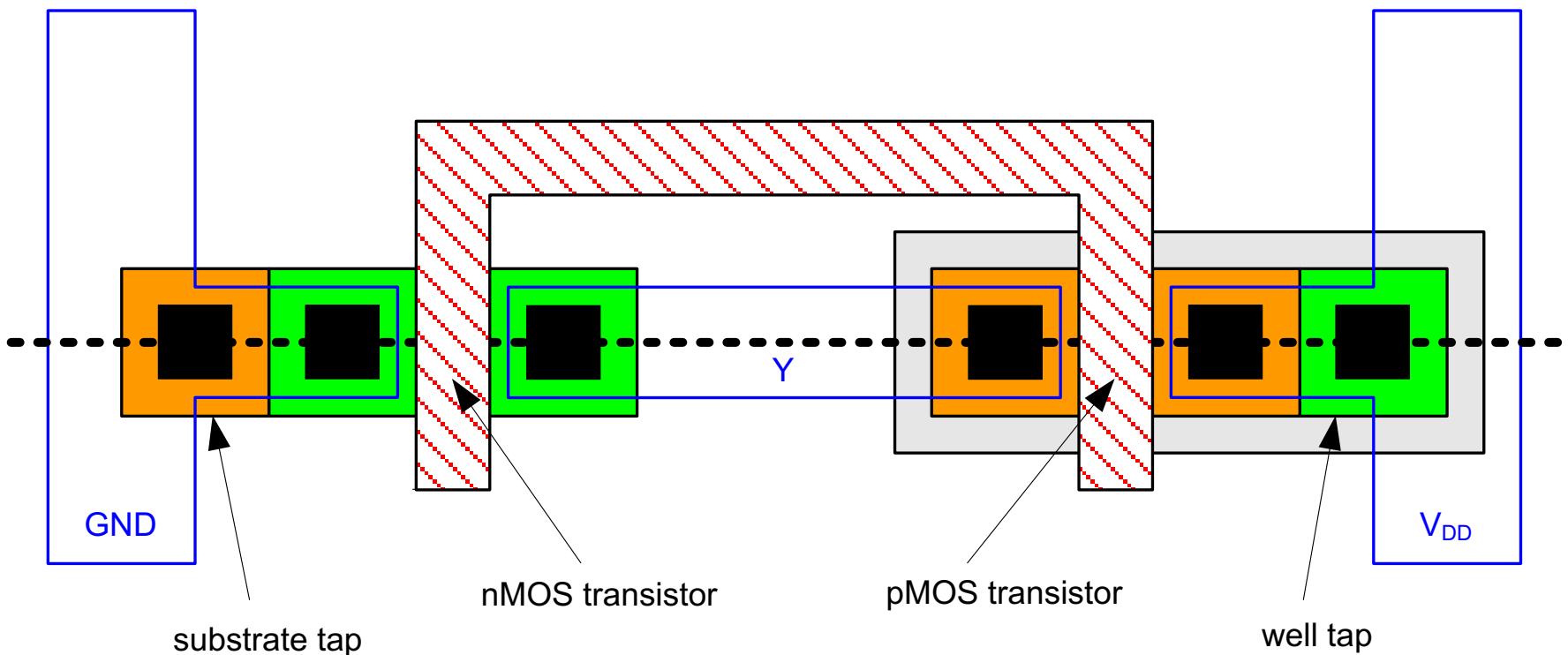
Line-to-via because it is the minimum pitch in which you can place a via next to a route track



Inverter Mask Set

- Transistors and wires are defined by *masks*
- Cross-section taken along dashed line

2D structure



Important Tips!

- Since wires are the contributing factor for most designs, it's important you handle the designs well even at 0.5um (we use 0.6u since this is the drawn transistor length, whereas, the 0.5um is the actual transistor length).
- Always assume your design is going to reuse design (**hierarchy**) as much as possible.
 - Object-oriented design.
- Wires are critical, therefore, make sure you use wires as “thinly” as possible at the lowest gate level layouts.
 - Try not use M2 for “leaf-cell” gates! (sometimes, you have no choice).
 - Leaf cells are the lower-level design structures for most systems and tend to be the items most designs scrutinize to optimize.
- As you lay down your wires, the following are useful paradigms.
 - Alternate your M1 and M2 layers in terms of methodologies even at leaf-level designs.
 - Use pitch as a qualifying distance for all designs! (establish design-wide parameter)

[https://en.wikipedia.org/wiki/Magic_\(software\)](https://en.wikipedia.org/wiki/Magic_(software))

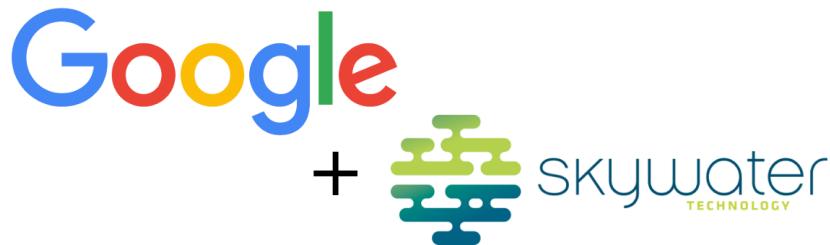
Interconnect

- Different layers of metal are typically called *interconnect*, because the interconnect parts together.
- However, there is different definitions on how these interconnects can connect together.
- Terminology
 - Contact : the connection of either poly or diffusion to metal1
 - polycontact
 - ndc/pdc
 - Vias: the connections of metal to metal
 - localli
 - metal1
 - metal2
 - metal3
- Avoid other methods to contact wires, such as poly and diffusion – the resistance and capacitance is much higher on these layers than metal layers (they have high amounts of connectivity).
 - Use metals!!!!!!



FOSS 130nm PDK

- Cooperation between Google, Skywater Technologies, including OSU.
- Goal is to create an entirely open-source Process Development Kit and flow.
 - Anyone should be able to jump on and create an ASIC without having to pay
 - We are providing them with a standard cell kit, and our general expertise
- Google has already contracted a fabrication facility.
 - They are producing chips and badly need more designs to fabricate
 - It is not economical to fabricate just a few chips at a time.
 - Google has indicated that there is room to tape out anything we at OSU want!



FOSS 130nm Production PDK
github.com/google/skywater-pdk

SkyWater Technologies : our Fab

SkyWater Technologies
Address: 2401 E 86th
St, Bloomington, MN
55425



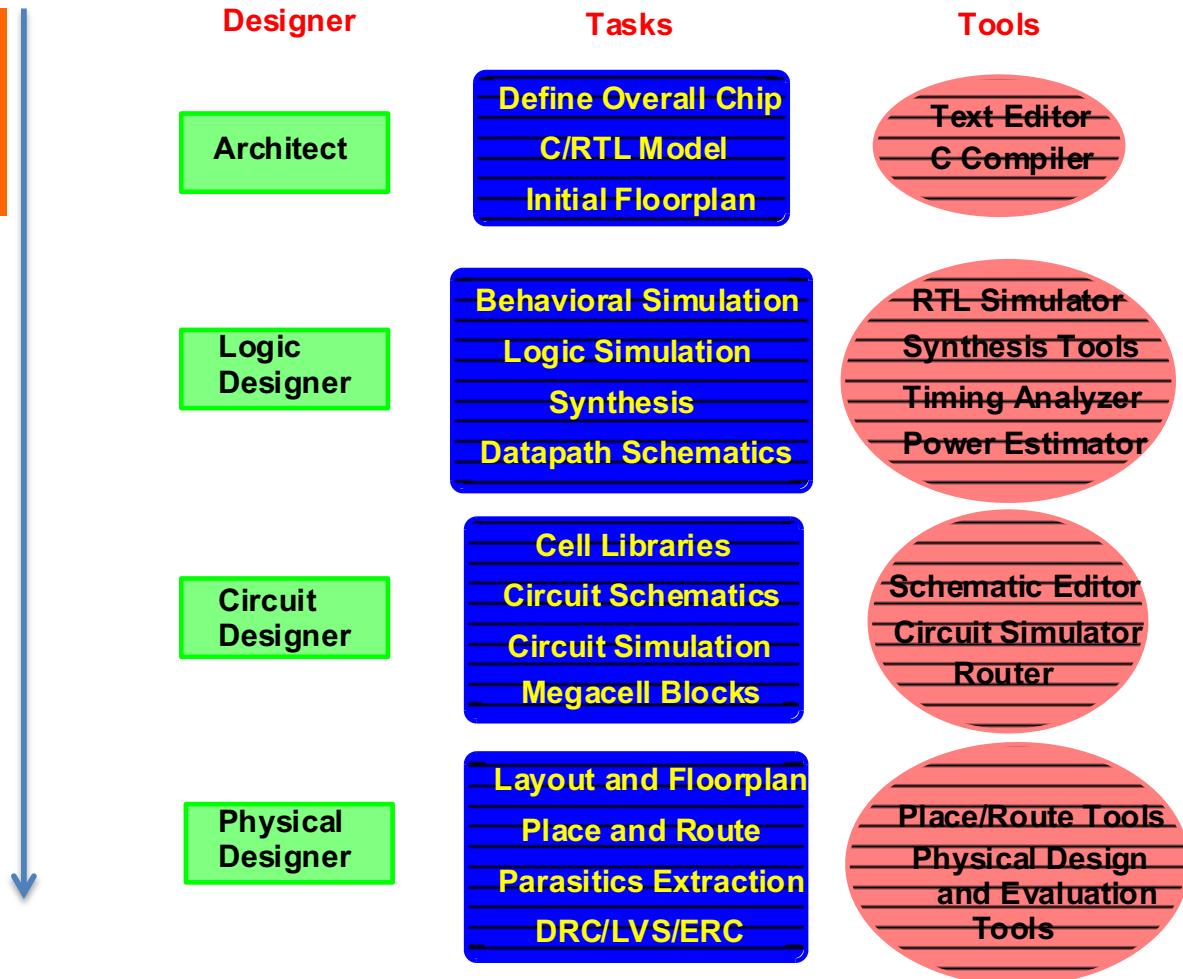
<https://www.skywatertechnology.com>

Steps in Design

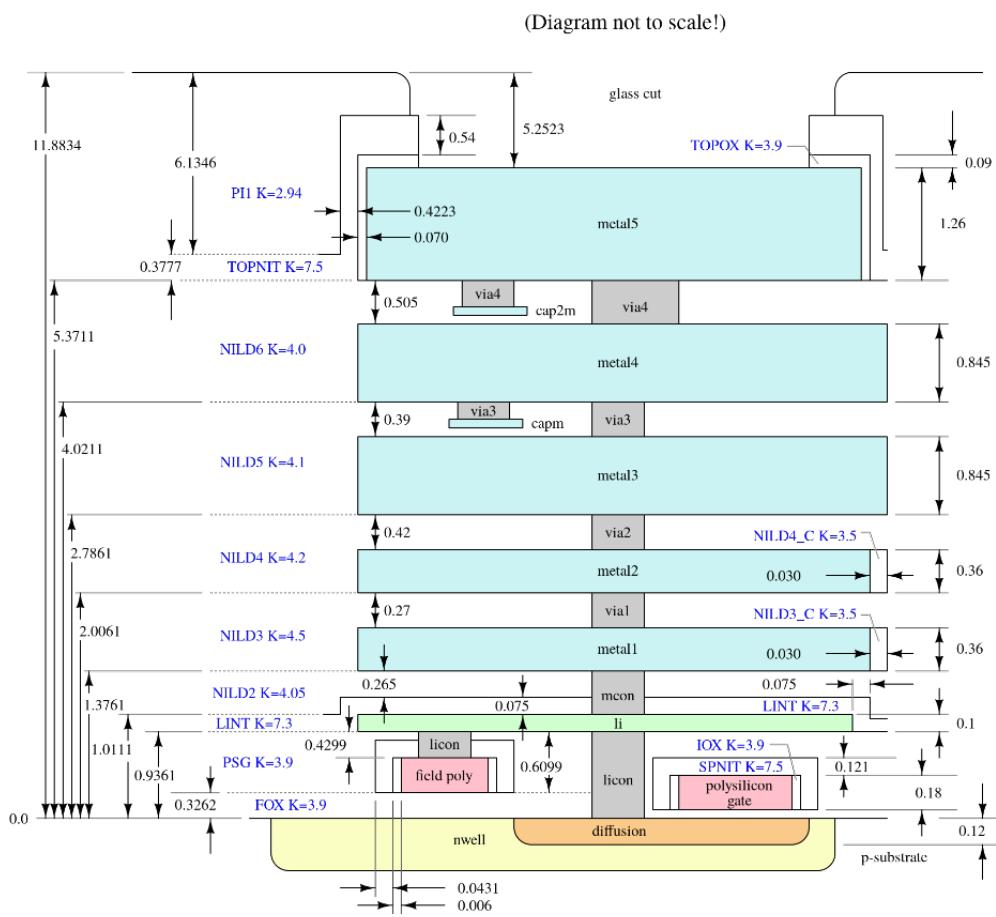
Synthesis of Integrated Circuits – forward motion in design!

Design Flow

What is it all about?



Metal Stack



- The 130nm process has 5 aluminum metal layers, as well as a high-resistance TiN “local interconnect” layer.
- There are 5+1 total metal layers available.
 - Try to not go beyond metal2 in your designs.
- li1 has a resistivity about 1/3rd that of poly, but around 100x that of m1.
 - In the metal stack diagram on the left, li1 comes between poly and m1.
 - You have to pass through it on your way up the stack.
- Your design will be better if you use it sparingly.
- Be creative!

Layer Cheat

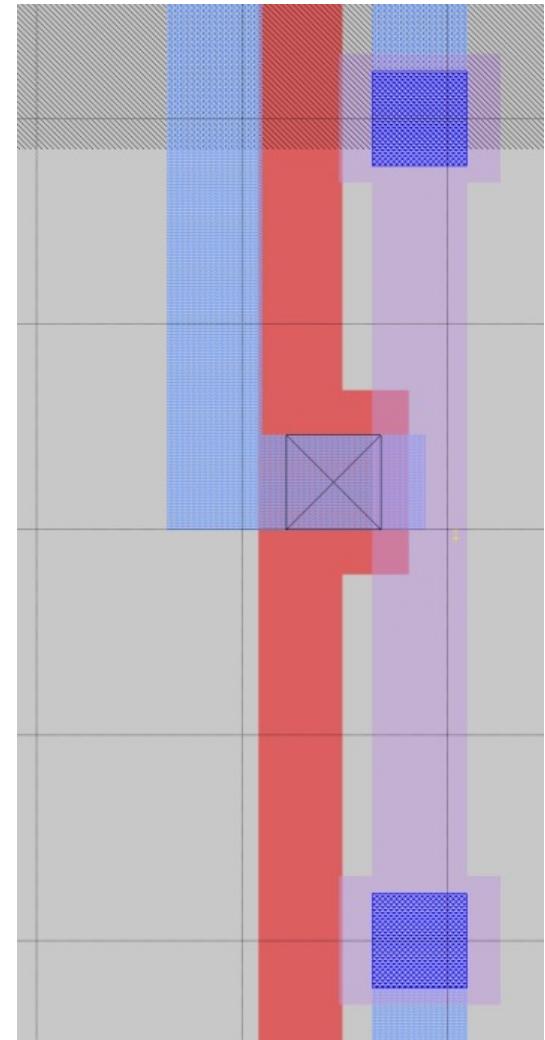
Layer	Name
nwell	N-type well
ndiff	N-type diffusion
pdiff	P-type diffusion
ndc	N-type diffusion contact
pdc	P-type diffusion contact
nscl	N-type substrate contact
nsd	N+ substrate
poly	Polysilicon
pc	Contact from poly up to li1
locali or li1	Local interconnect (TiN)
metall1	Metal 1 layer
m1c	Contact from li1 up to m1

Drawing

- Everything goes according to rules otherwise you have a DRC.
- Ask questions if you get something on slack that you cannot fix.
- 1 grid unit = $5\text{nm} = 5 \times 10^{-9}\text{m}$
- We no longer use $2 \cdot \lambda$ for length → its drawn to the size you need.
- Our length is $0.15 \mu\text{m} = 150\text{nm}$ even though we use a 130nm process
(remember the L_{eff})
 - That means for a $150\text{nm} = 30$ grids
- Use the :grid to help you, if needed (remember you can redefine the grid to whatever you want – e.g., :grid 30)

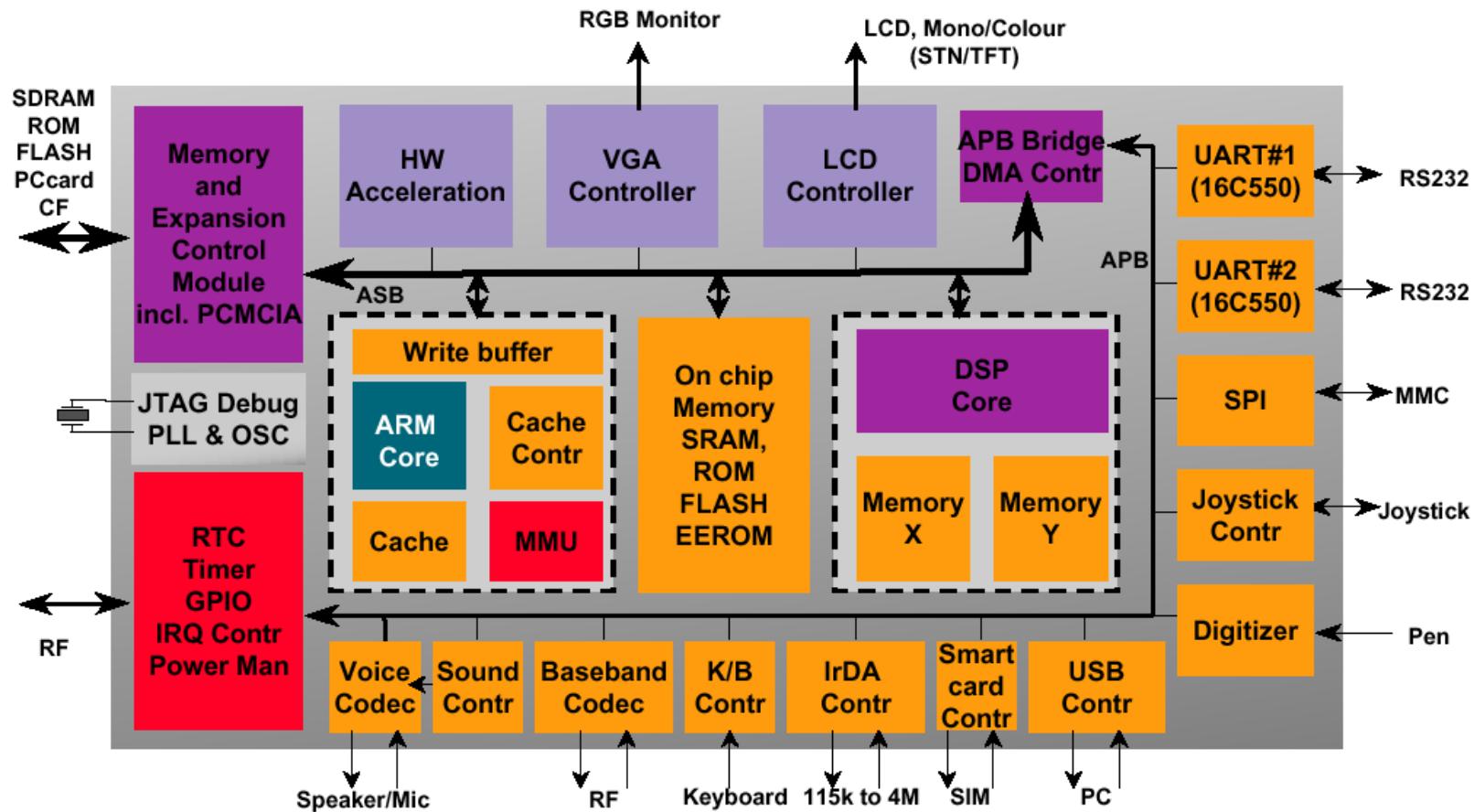
Design specifications

- In this 130nm process, as is the norm in more advanced processes, vias require overhang.
 - Examples can be seen on the right.
- The spice models for this technology are currently being updated.
 - As of right now, only a discrete list of transistors can be correctly modeled.
 - Lengths should be 0.15um (the minimum)
 - Widths should be either 0.42um, 0.55um, 0.64um, 0.84um, 1um, 1.26um, 1.68um, 2um, or 3um
- Restrictions encourage creativity 😊



System on a Chip (SoC)

Source: ARM



This is not any different than putting together several devices on one breadboard!

Design Flow

FOSS 130nm Production PDK
github.com/google/skywater-pdk

- Design flow is important, because it makes a viable path towards a mask.
 - SKY130
 - <https://skywater-pdk.readthedocs.io/en/main/>
- A mask is the format which is uploaded to the machines that represents layers in three dimensions.
 - gds2 (CALMA) – most popular format as it's a binary format and can easily be secured cryptographically.
 - Represented by a number from 0 to 255 (8-bits).
 - cif (Caltech Interchange Format) – the format we will use in this class (text-based and output of Magic by typing :cif).
 - Each layer is defined by three letter designator for each and every layer.
 - Extra layers not utilized for mask machines are removed.
 - e.g., CPG = polysilicon

Scripts

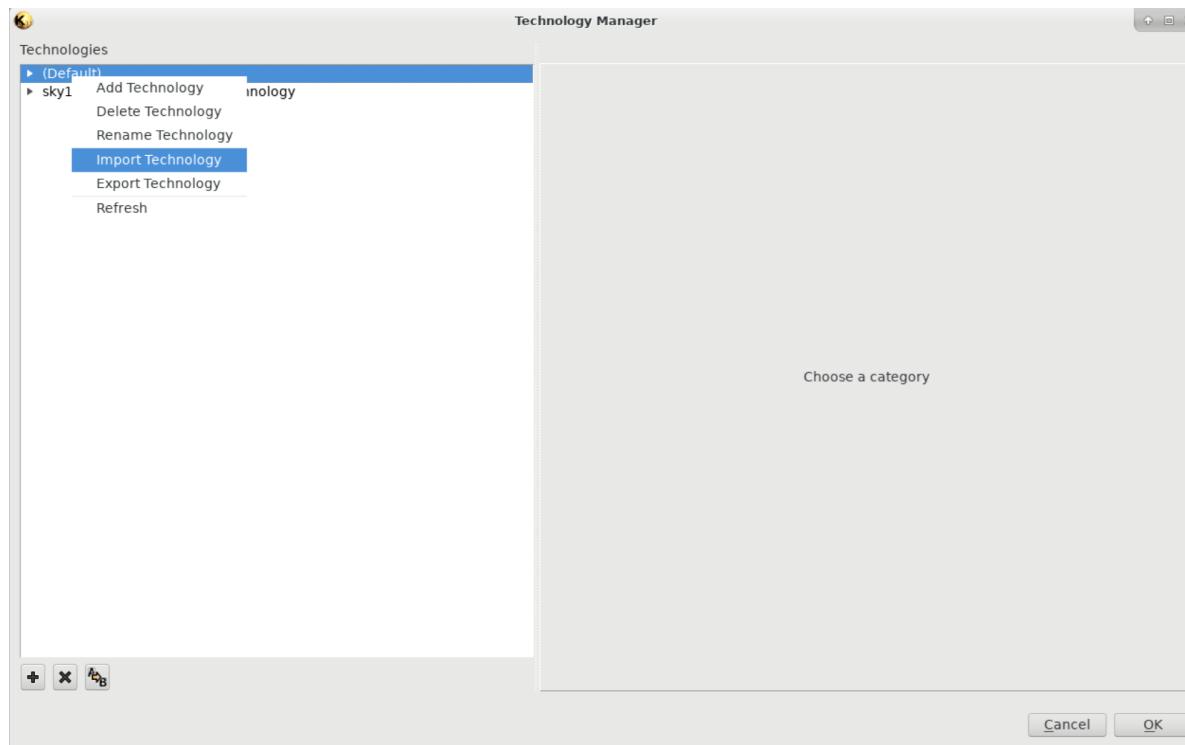
- Scripts simplify life pure and simple.
 - A good script is easy to use, well documented, and useful.
- We use one I created that students enjoy using over the last couple of semesters.
- `ext4mag file.mag`
 - This script will take a magic file and extract it to give you its .gds, .sim and optionally its SPICE deck (more on SPICE later).
 - For now, you can respond “N” when asked if you want to generate SPICE.
- Feel free to modify – you can find its location by typing:
 - `which ext4mag`

Layout

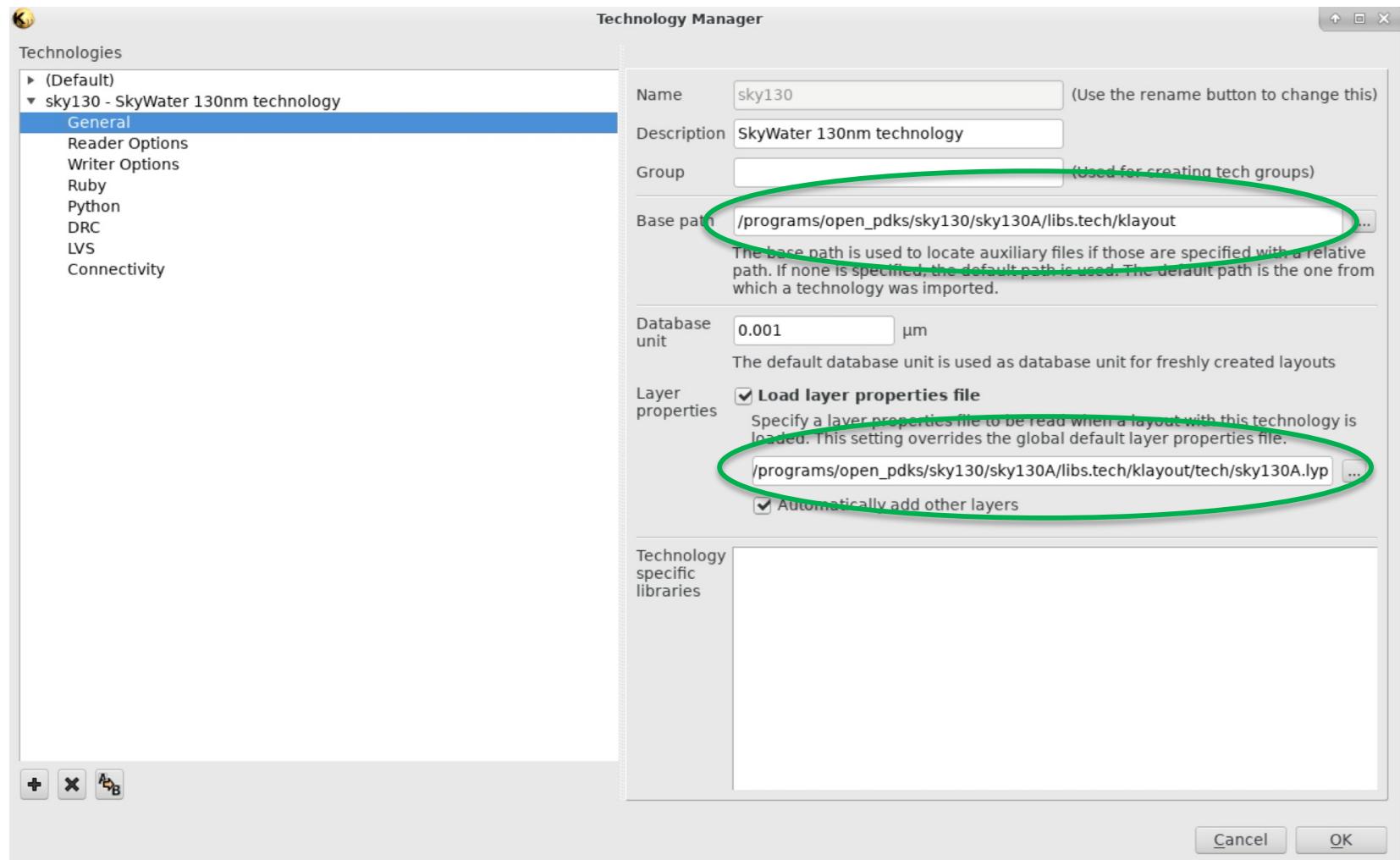
- Magic
 - Output to CIF/GDS
 - Use your stick diagrams!
- Printing of Magic – always klayout (please do not use screen shots!)
 - klayout translates gds layers into postscript output that can be inserted into documents or printed to PDF.
 - <http://stineje.github.io>
 - ps2pdf ./file.ps
- klayout should be installed with eda-tools
 - klayout inv.gds
 - You may need to set some of the layer information (see next slide).

Add sky130 to klayout

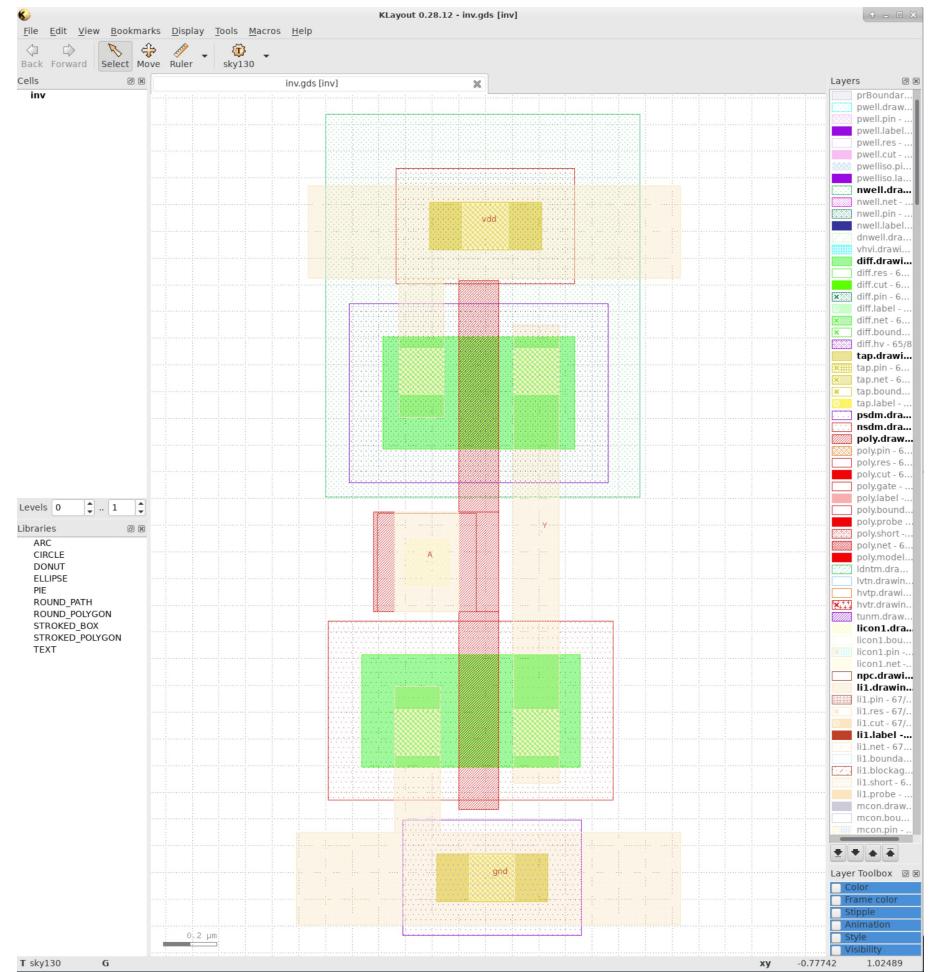
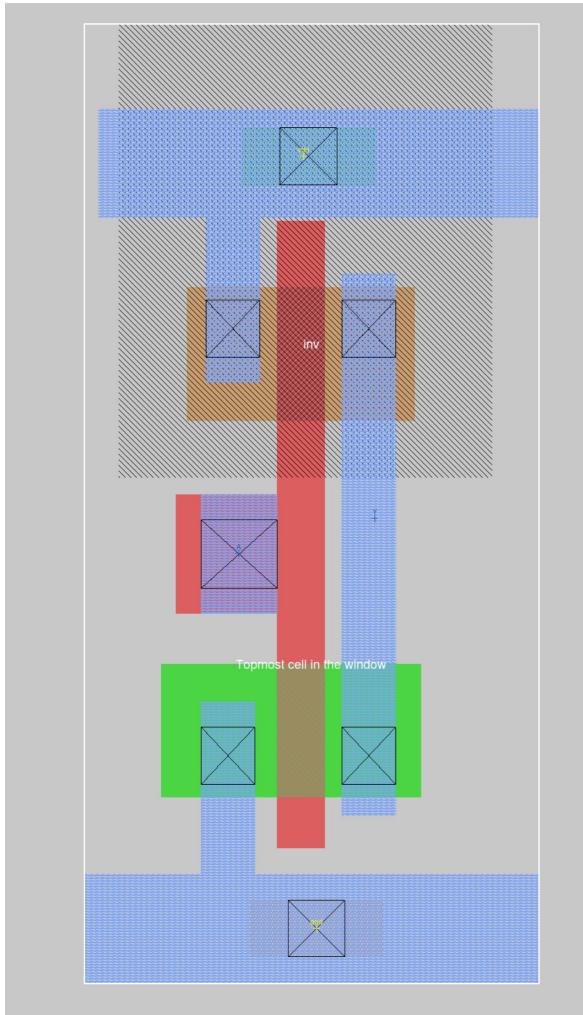
- Tools → Manage Technologies
- Set location of files on the screen : right mouse on Default and Import Technology.



Settings for klayout



Layout vs. Plot



Verification

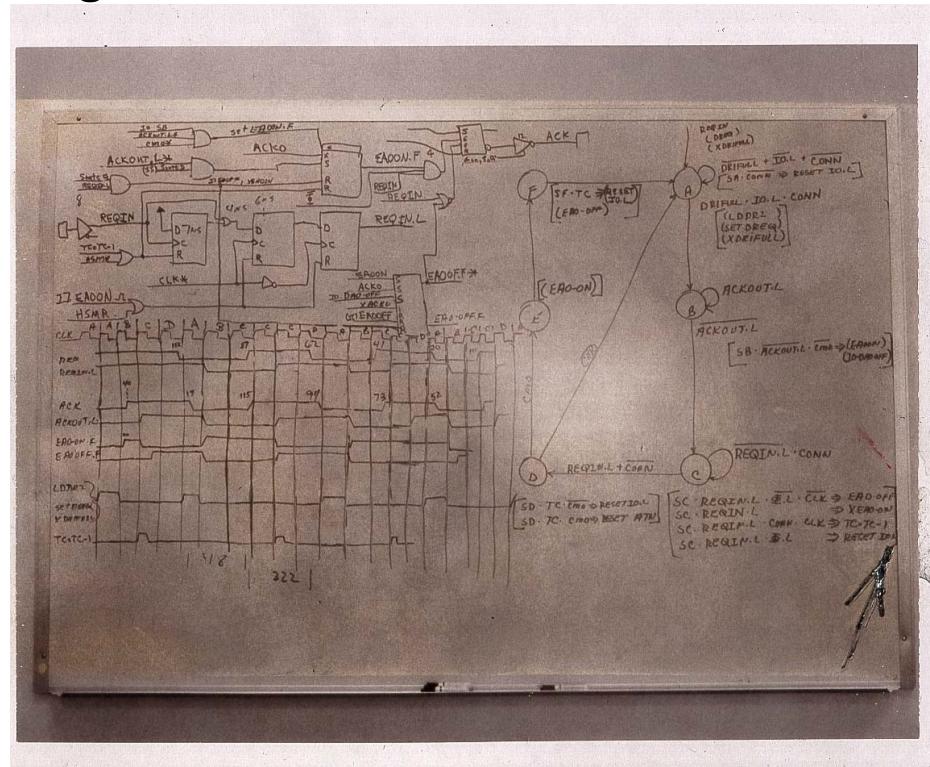
- Verification is the process of going back from a stage in the design flow to actually figure out what is happening and whether you are on the right track.
- All tools use netlist-to-netlist design stages!
- Schematic entry is utilized to guarantee designs are actually what they are designed for.
 - Design for Manufacturing (DFM)
 - DRC – Design Rules Check – guaranteeing layout obeys rules.
 - LVS – Layout vs. Schematic – guaranteeing schematic is actually what its supposed to be.
 - ERC – Electric Rules Check – guaranteeing actual design meets electrical reliability rules (e.g., Wells are attached properly).

Never forget LVS!

- Never forget LVS – it guarantees you are on the right track.
- Certain companies have “grounds for dismissal” in their contracts for missing LVS!
- LVS should be used after every cell passes DRC.
- Use it as a debugging feature as designs are attached one-by-one.
 - It also is a check on whether you are doing things correctly.
- Starts with the schematic – this is usually done first at most companies.
 - Schematics can be textual
 - Hardware Descriptive Languages (HDL)
 - SPICE files
 - MultiSim from National Instruments

Simulation

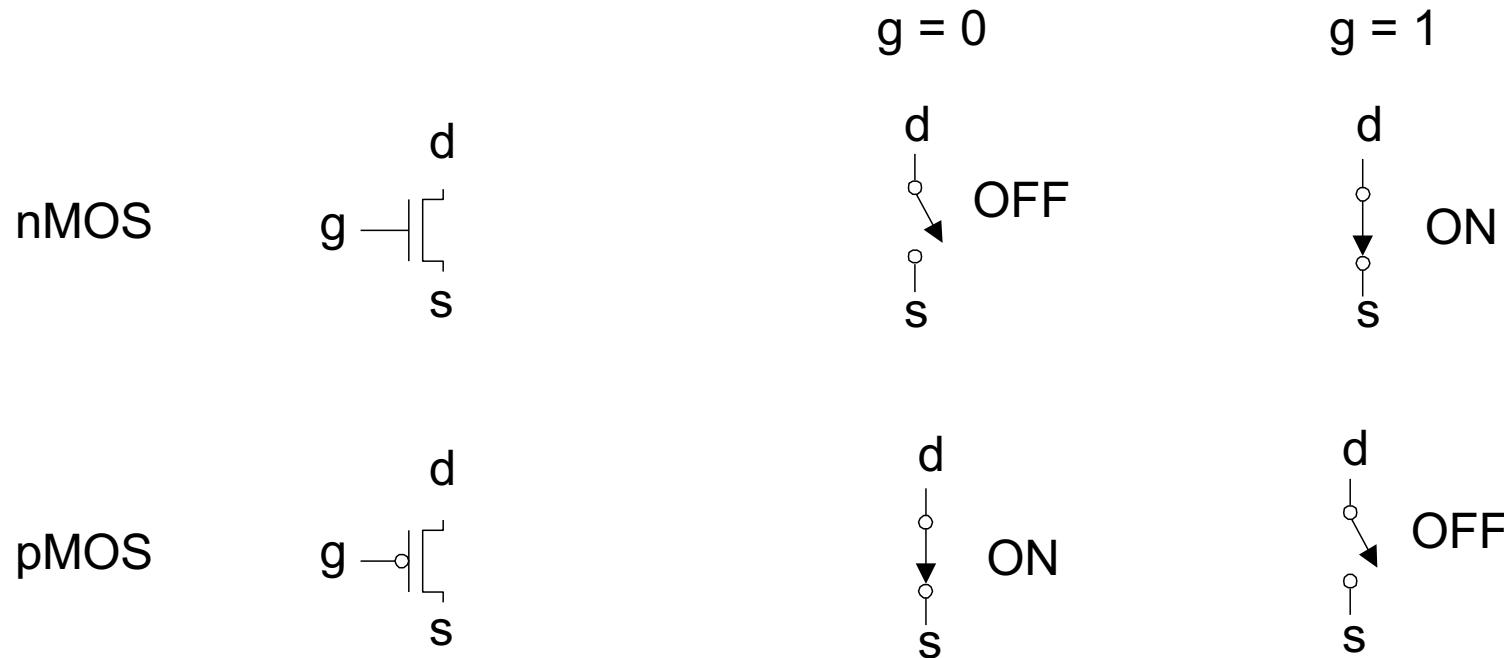
- Simulation is important, because it validates design.
- Designers will sometimes look to emulate a certain design sequence.
- Therefore, having the ability to simulate your design is important for validating your design!



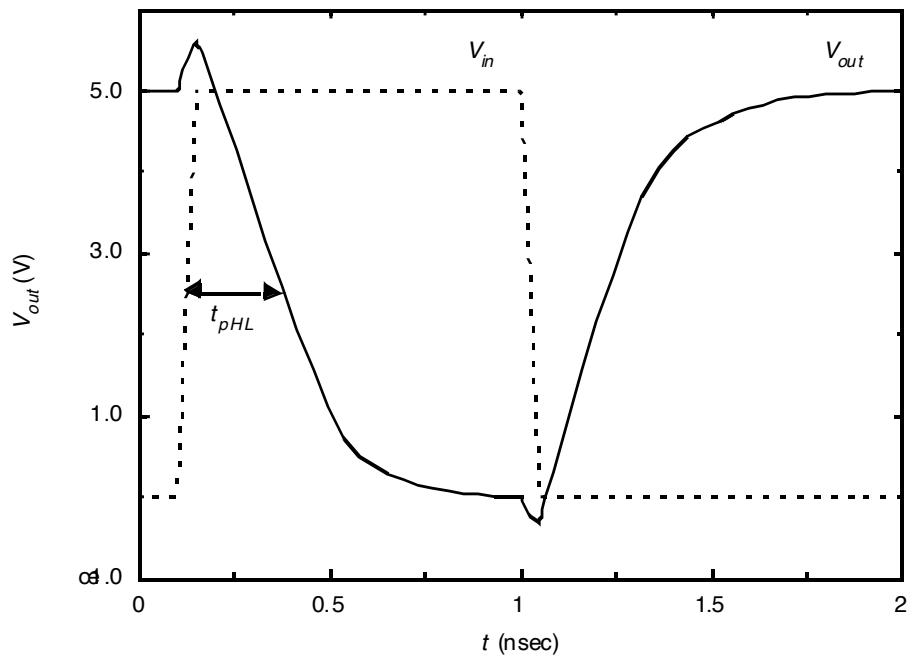
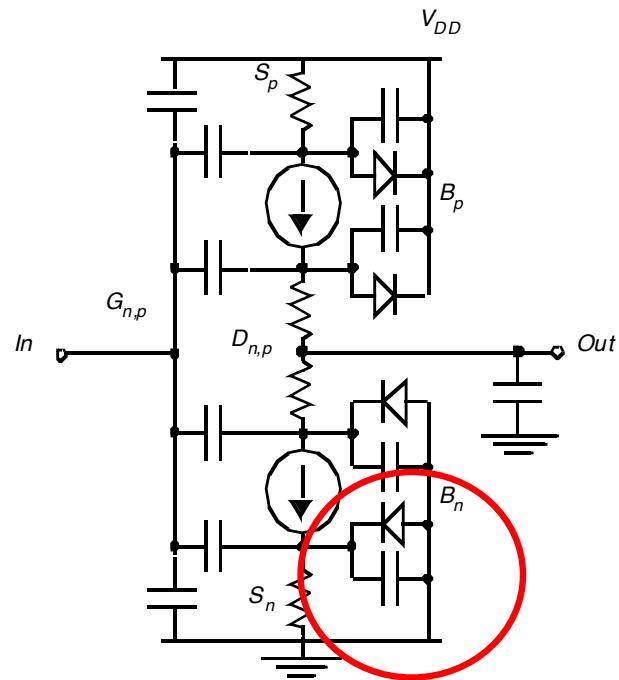
Actual photo of whiteboard used to develop NCR SCSI chip – [Smithsonian] 32

Switch-Level Simulation

- It is important that we use abstraction to make design function!
- That is, we use binary to help us understand digital logic function even though Voltages are actually manipulated.
- We can model our designs as switches!
 - Thus, we can model them in simulation software!



Digital Data as Analog Signals



Circuit Simulation

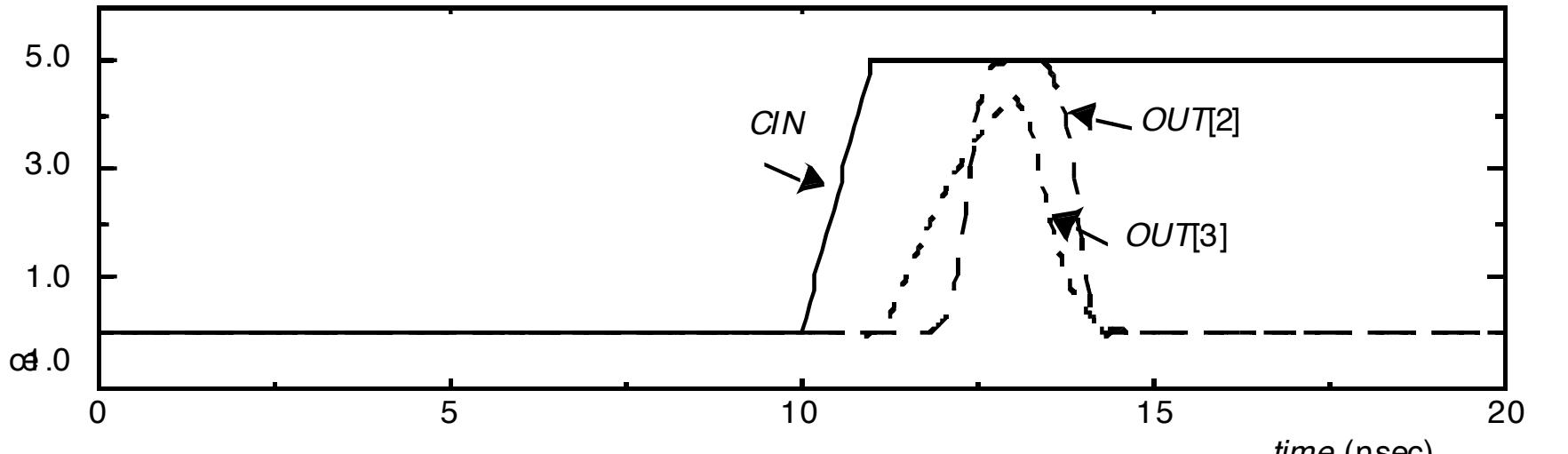
Both Time and Data treated as Analog Quantities
Also complicated by presence of non-linear elements
(relaxed in timing simulation). Impractical for large circuits

Discretizing Time

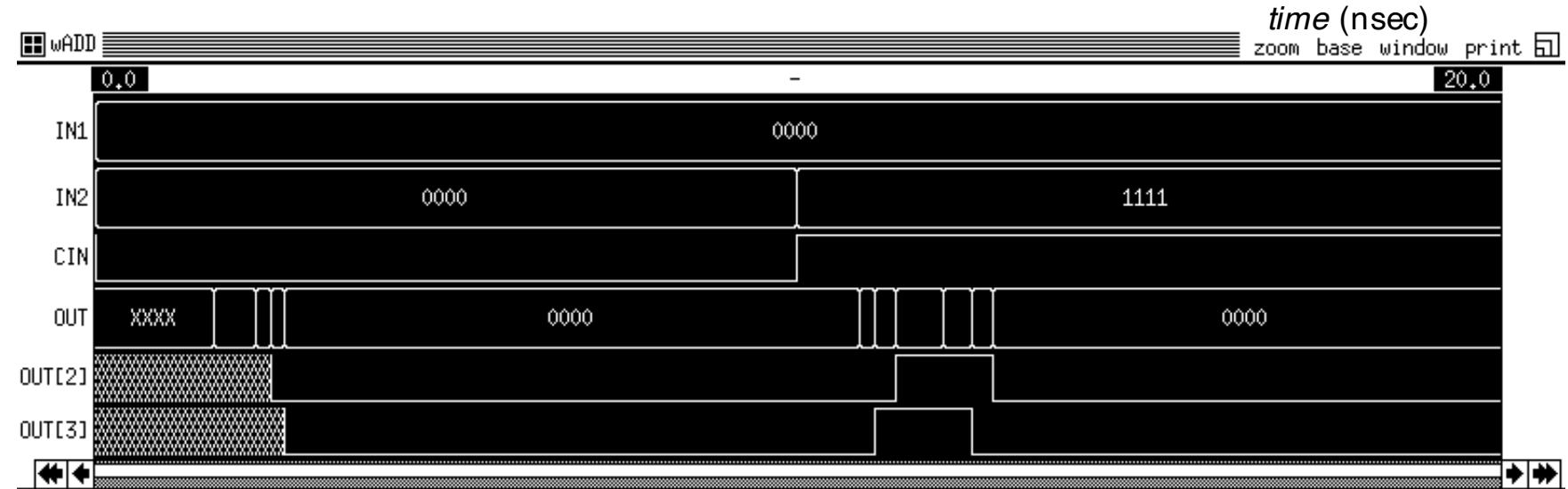
- Evaluate circuits only at “interesting” times
- Event-driven simulation
 - evaluate gates only at a future time of interest
 - current time + gate delay
 - for more accuracy gate delay = function of load
 - still, events can happen at any time
- IRSIM is a switch-level simulation tool we will be using for our digital circuits to get an idea of the design flow aspect of design.

Circuit vs. Switch Level Simulation

Circuit



Switch



[Adapted from <http://infopad.eecs.berkeley.edu/~icdesign/>. Copyright 1996 UCB]

CMD Files

- IRSIM allows command files to automate verification (sort of like a batch file).
 - You should use known “good” vectors to always test your design.
 - What constitutes “good vectors?
 - A collection of “good” vectors that designers use for a design is called a golden file.
- CMD files are easy to create
 - See wiki
 - Let’s try an example...

Design Flow

1. Create Schematic
2. IRSIM : switch-level simulation (schematic)
3. SPICE : circuit-level simulation (for sizings)
4. Create Stick Diagram
5. Design Layout
 - DRC
6. LVS (always!) ←
7. IRSIM : switch-level simulation (layout)
8. SPICE : circuit-level simulation of extracted layout (for timing/power)
9. Plot of layout (klayout)



Note: Hardware Descriptive Languages (HDLs) have overtaken simulation of operation (and initial timing) as they easy to use and repeat!

Summary

- We have a modified version of the flow that we have used in the past.
- We will explore this new addition to hopefully help you get better ideas about how to create layout with better EDA software.
- The new tool is called Mentor Graphics (now Siemens) Calibre
 - It is an industry-standard EDA tool and seems to work very well for what we need.
- Use scripts and command files to simulate things easily and repeatedly.
 - Good use of scripts can save groups several hours of work!



Let's be creative
and have fun!!!!