Lecture 2

The Building Blocks

Slides adapted from: Mark Horowitz
with contributions from Subhasish Mitra and Don Stark

Overview

Reading

W&H 1.4(Device model); 1.5 Fabrication

http://www.intel.com/pressroom/kits/chipmaking/

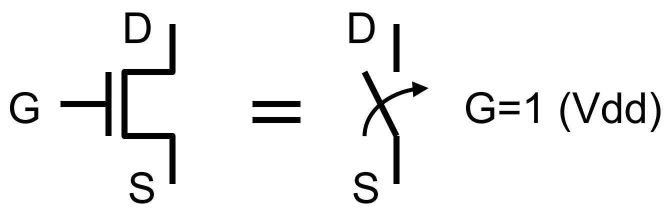
http://www.youtube.com/watch?v=duzO0YX4WnA

+ W&H Ch. 3 Fabrication in more detail

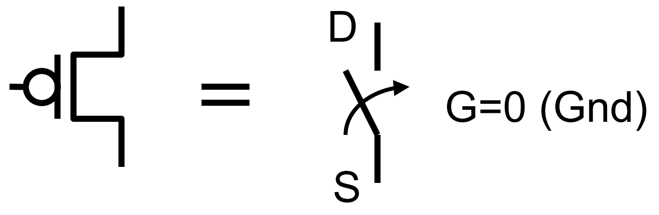
Introduction

In this lecture we look at the building blocks of CMOS logic, transistors and wires. While a transistor is a complicated non-linear element, for most of the time we can use a simple abstraction for it, modeling it as a switched resistor. Logic gates can be formed by connecting the output (through switch networks) to either Vdd or Gnd. Next we look at how these transistors are built, and how to create the layout needed for their fabrication.

Simplest Model for MOS transistors



NMOS transistors



PMOS transistors

- Let Logic value 1 be Vdd, value 0 be Gnd

Life is not this simple but let's assume that now. More details soon.

- NMOS devices are switches
  - G is 1 -> the drain D and source S are connected
  - G is 0 -> the drain D and source S are not connected
- PMOS devices are switches
  - G is 0 -> the drain D and source S are connected
  - G is 1 -> the drain D and source S are not connected

Terminology

Note that the source and drain terminals are really the same, but by convention the source terminal is the one with the lower voltage on it. Thus, the maximum voltage between the gate and the {source, drain} is the voltage between the source and the gate. (This fact will be important later.)

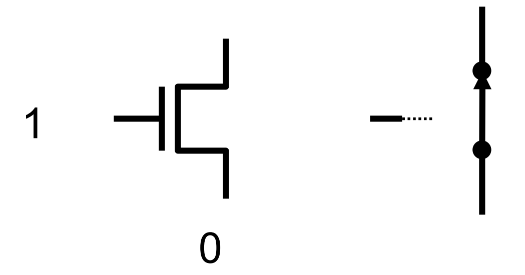
The voltage on the gate controls the connection between the source and the drain. When the gate is high, the source and drain are connected together. When it is low, the terminals are disconnected.

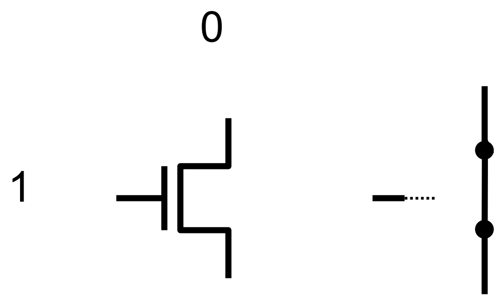
CAUTION: do NOT use the words "open" and "closed" to describe switches. Is open an open electrical circuit (no flow), or an open fluid valve (flow)? You get opposite results, depending on which analogy you use.

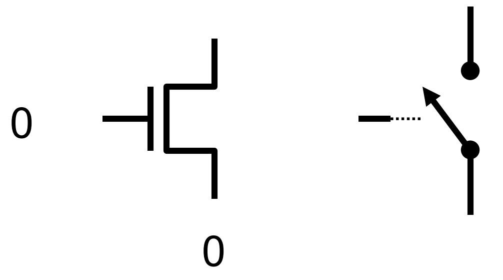
This description is for nMOS transistors. For pMOS everything is reversed. The source is the higher voltage terminal, and the transistor is on when the gate is much lower than the source. More on pMOS later

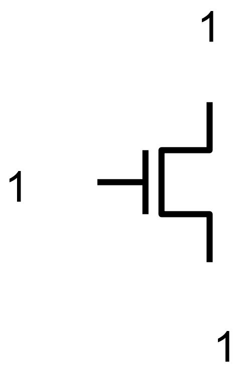
- Transistor Examples

The state of these transistors:





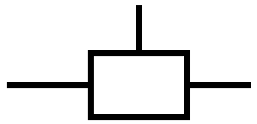




Complicated, it is really off. More on this later. Assume it is on for now

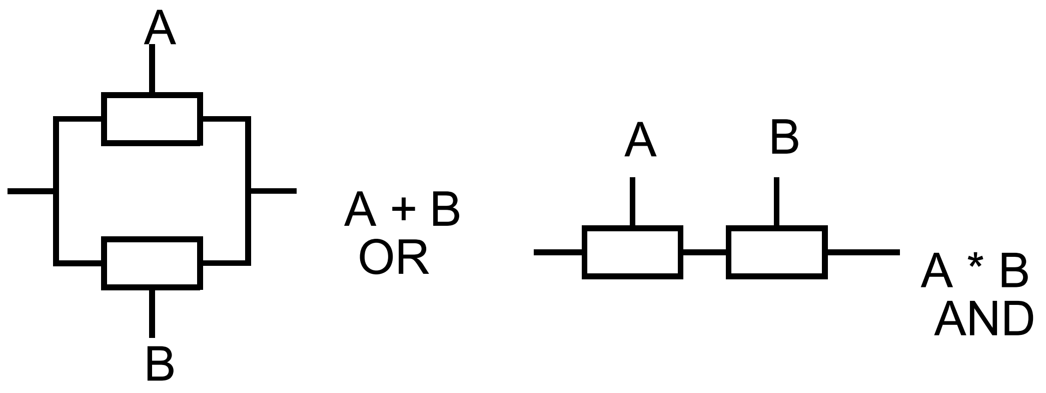
Switch Networks

- Since transistors can be modeled as switches
  - Look at what we can make out of switches
  - Draw an abstract switch as
    - Control (gate) terminal is on top



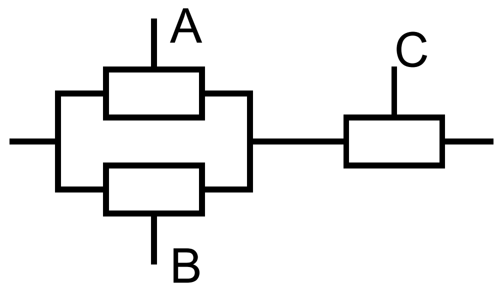
- Can build switch networks
  - Are not logic gates themselves!!!
  - Are a collection of switches that still have two non-control terminals
    - Define function of a switch network as the inputs conditions that connect the two non-control terminals of the network
- Structure of switch network sets its logic functions:
    - 'OR' functions are constructed by parallel switches
    - 'AND' function are constructed by series switches

- Switch Networks

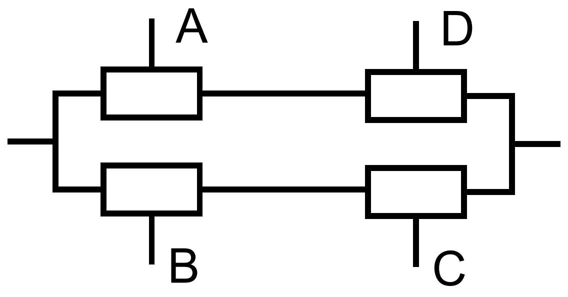


- The function of a switch network is true when the two terminals of the network are connected together. Since for parallel switches the terminals are connected if either switch is on, the function is OR. For series switches the network is conducting only if both switches are on, hence an AND.

-General Switch Networks



(A + B) C



(AD) + (BC)

- More complex connections are possible
- Composition rules are simple. Use a recursive definition:
  - Parallel combination of switch networks yields an OR of the component switch networks' functions
  - Series combination of switch networks yields on AND of the component switch networks' functions.

But Switch Networks are not Gates

- What to be a logical abstraction
  - Inputs are read, and produce output
  - Implies unidirectional

- While switches “read” the control inputs
  - They connect their other two terminals (or not)
  - How to make them completely unidirectional?

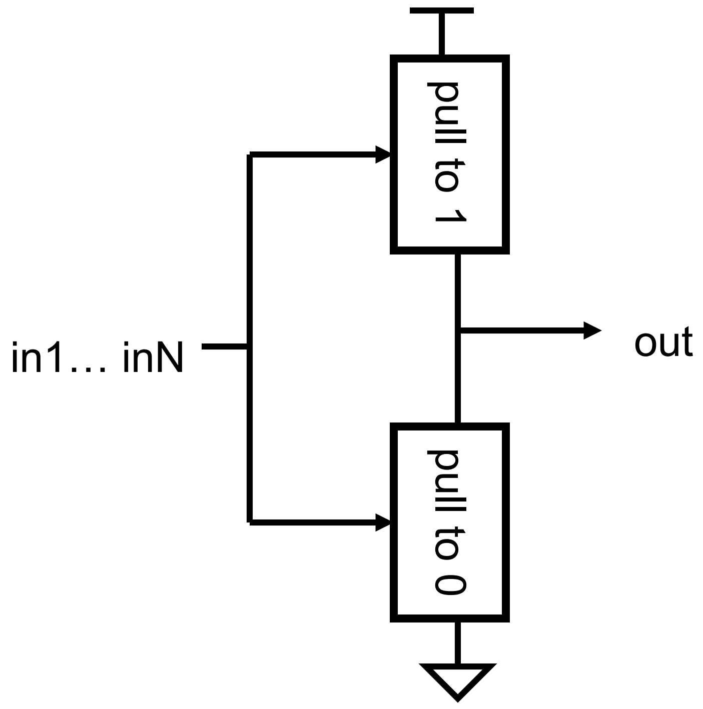
- Only allow the output to connect to constants
  - Vdd (1) and Gnd (0)
  - If you don’t do that, your circuit might still work
    - And there are many designs that use “transmission gates”
    - But they are not unidirectional signals

Logic Functions From Switches

- Not a difficult task if you follow two simple rules
  - Connect switches so output is always driven to either Vdd, Gnd
  - GND and Vdd are never connected to one another
    - Don’t want to short the power supply out.

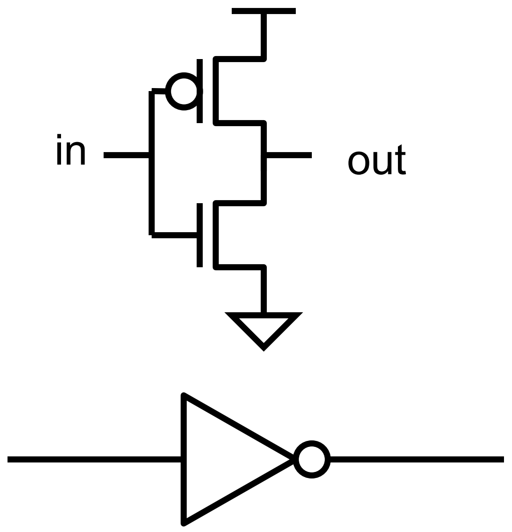
- What does this mean that all gates look like?

Two Switch Networks: Pull-to-1 and Pull-to-0



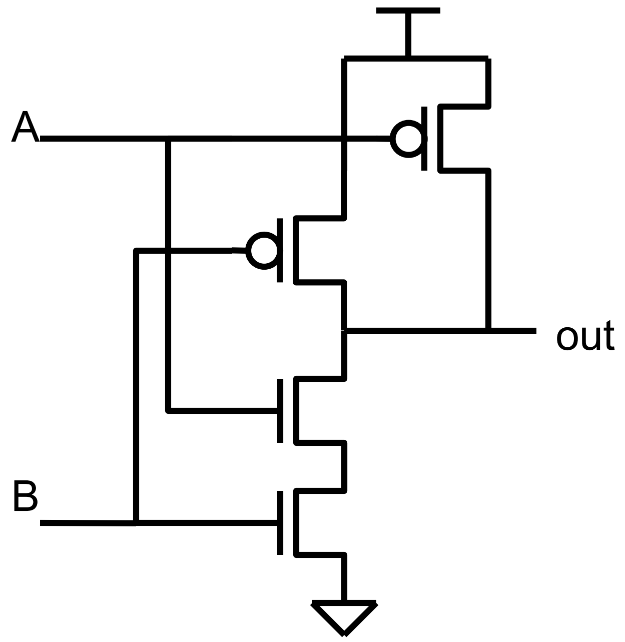
- To satisfy our two rules:
  - Make the switch functions complementary
  - F + F’ = 1 F * F’ = 0 F’ = not(F)

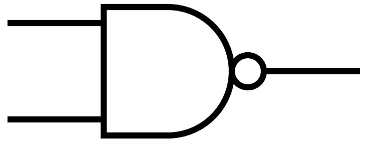
Degenerate Case: Inverter



- Meets the rules
  - Out is always driven (Either pMOS or nMOS is always active)
  - Vdd and Gnd are never shorted
    - At least with valid inputs

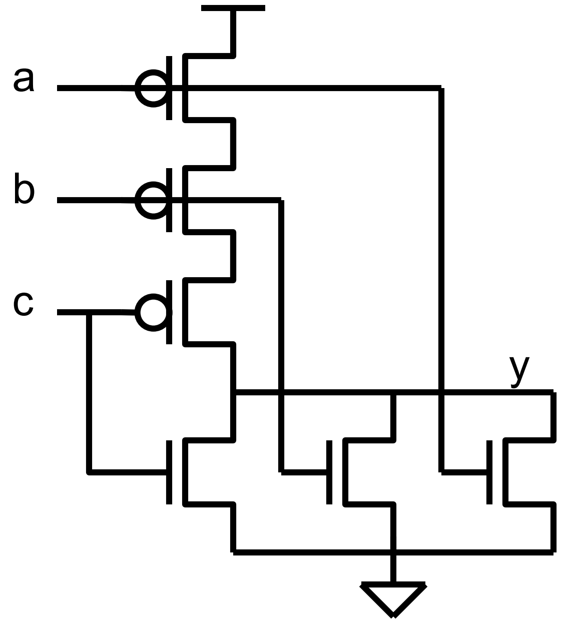
NAND Gates

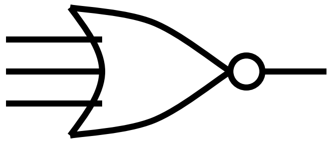




- nMOS in series pulling to Gnd
- pMOS in parallel pulling to Vdd
- Out always driven by pMOS's or nMOS's but never both
- Number of inputs ("fan-in") can be increased

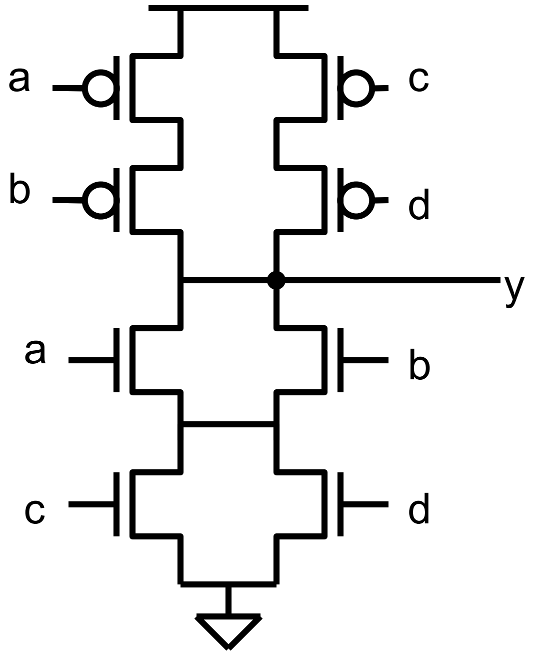
NOR Gates

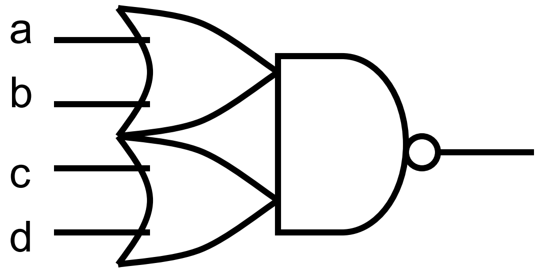




- pMOS in series
- nMOS in parallel
- Either pMOS or nMOS drives but not both
- Dual of NAND gate

More Complex Functions Possible





- Large number of possibilities
- Typical standard cell library will have many permutations

Demorgan Law is Your Friend

- Remember Demorgan's Law?
  - (a + b)' = a' b'
  - (a b)' = a' + b'
  - More generally the complement of a function
    - Is the dual of the function with inverted inputs.
  - Dual = exchanging the AND and OR operations
- Can apply to arbitrarily complex expressions.

a' = NOT (a)

- Now think about CMOS gates
  - Pullup switch function is complement of pulldown function
    - And pMOS gates invert all the inputs
    - Implies the pullup network is the dual topology of pulldown

Building Transistors and Wires

- Shown that gates can be built from transistors
  - Latches and flops can be built as well

- So all we need to be able to do is build transistors and wires
  - Transistors
    - Need nMOS and pMOS
    - Will need to have something to separate nMOS from pMOS
  - Wires
    - Need multiple levels of metal wires (need lots of wires)
    - Need insulation around wires
    - Need contacts/via, holes in the insulation
      - To connect wires on different levels

The Magic of Semiconductors

- I assume you are not that interested in semiconductor physics
  - Have a brief introduction in the next few slides
  - Book has a longer description (W&H 1.3)

- So let’s just say that semiconductors are magical
  - If you put ndiff in them you can make nMOS transistor
  - If you put pdiff in them you cam make pMOS transistors
  - But nMOS and pMOS don’t like each other
    - They can’t share the same space
    - The space around nMOS must be Gnd (0)
    - The space around pMOS must be Vdd (1)
  - So pMOS need to be surrounded their “own” space (nwell)
    - And nMOS needs to be spaced away from the nwell

Fabrication

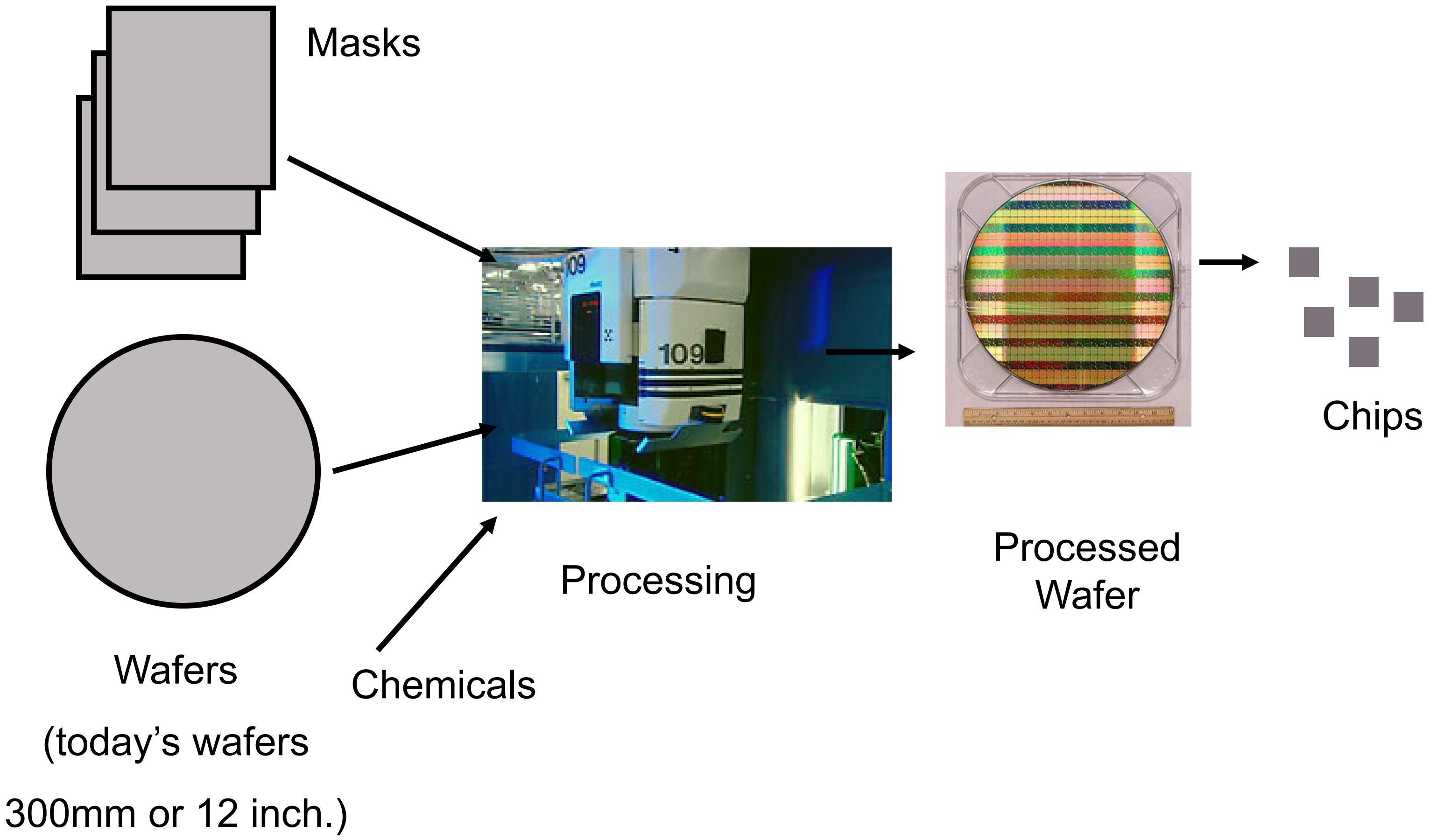
The process used that creates these devices/wires.

- Look at how to create:
  - Working transistors
    - ndiff, pdiff, wells, poly, transistors, threshold adjust implants
  - Wires
    - contacts, metal1, via, metal2

Fabrication is pretty complex.

- There are whole classes devoted to it.
- Give a brief overview of the process, for background.
- Want to understand origin of layout rules / process parameters
  - The abstractions of the process for the designer (us).

A High Level View: Fabricating Chips



Photolithography – The Key

- Take an image on a plate of glass
  - Project on to a wafer
  - Like photo-enlarger
- EXTREMELY HIGH Resolution
  - 30nm out of 300mm
  - 1/10,000,000
- If printed the US at this resolution
  - Approx 4,000 km
  - Resolution would be .4m!

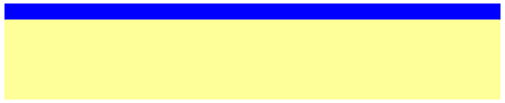
- Basic Fabrication Steps

- Transfer an image of the design to the wafer
- Do something to imaged parts of wafer
  - Implant – add impurities to change electrical properties
  - Deposit – deposit metal or insulator from chemical wafer
  - Grow – place silicon in oxidizing ambient
  - Etch – Cut into surface of topmost layer(s)
- Strip off image and sometimes polish wafer to make it flat
- Go to the next step

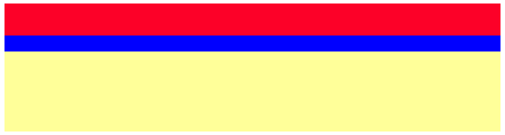
- Key points:
  - Entire die image transferred at once
  - Entire wafer surface processed at once
  - Economical because everything done in parallel

Basic Processing

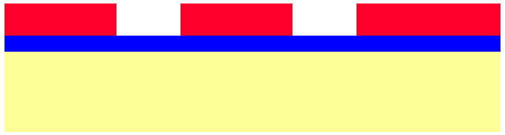
Start with wafer at current step



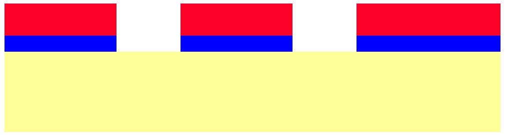
Spin on a photoresist



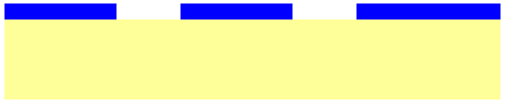
Pattern photoresist with mask



Step specific processing
etch, implant, etc...

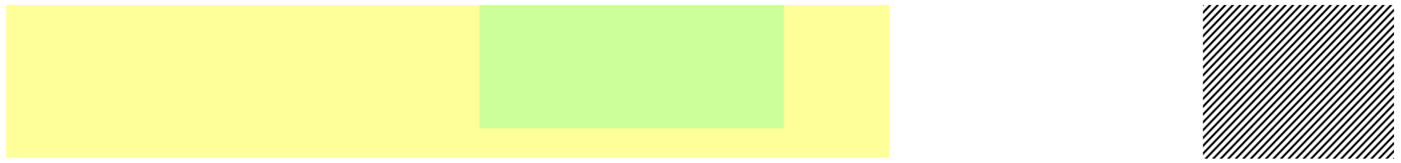


Wash off resist

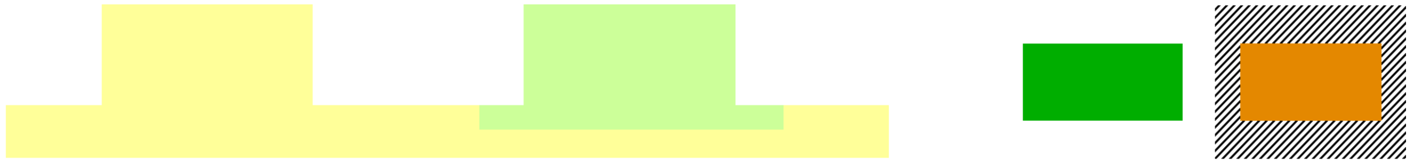


Making Transistors - 1

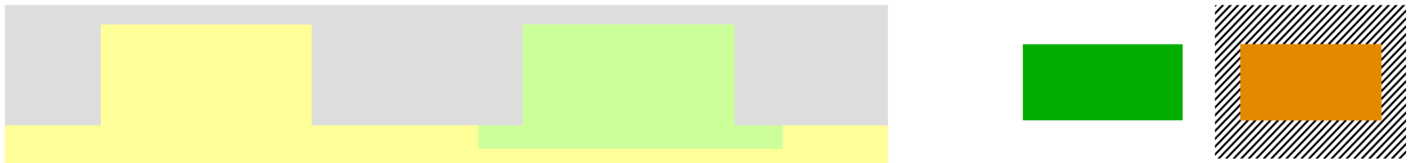
Implant N-Well



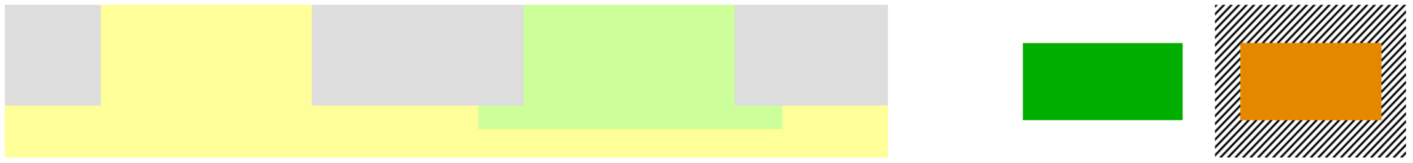
Etch away silicon where there won't be transistors



Deposit oxide in trenches

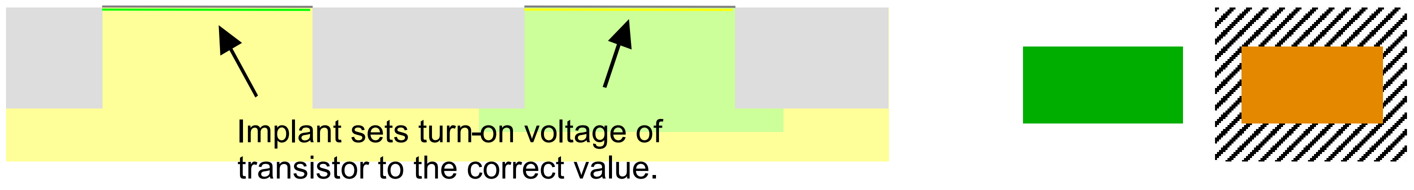


Polish top of wafer flat and expose silicon

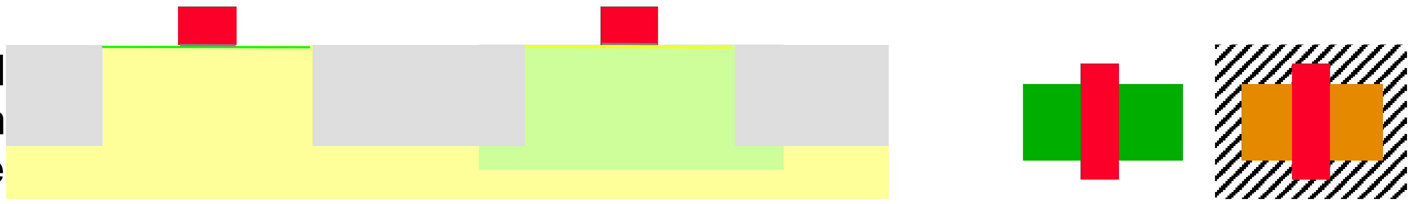


Making Transistors - 2

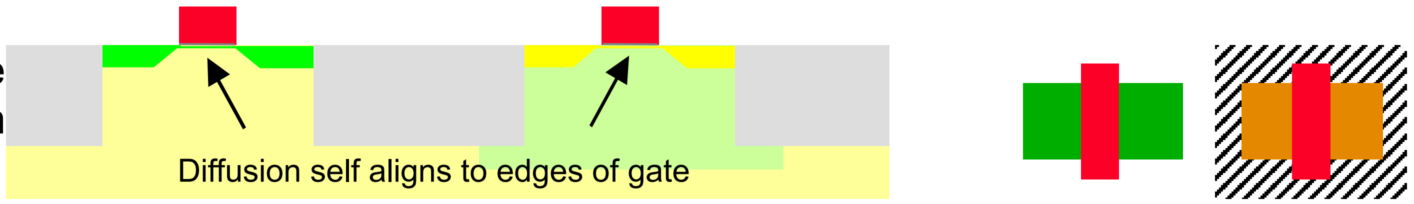
Gate Implant
and grow oxide


Implant sets turn-on voltage of
transistor to the correct value.

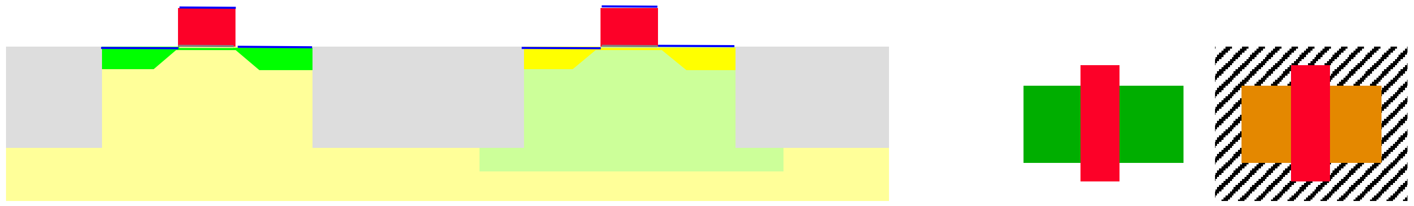
Deposit and
Etch Polysilicon
for gate



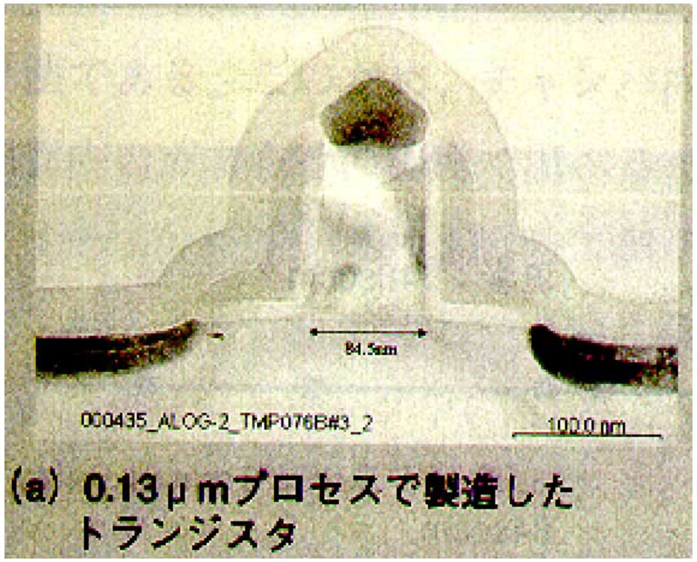
Implant source
and drain



Coat Poly and
Silicon with
metal to reduce
Resistance
(Cosi)



Actual Transistors

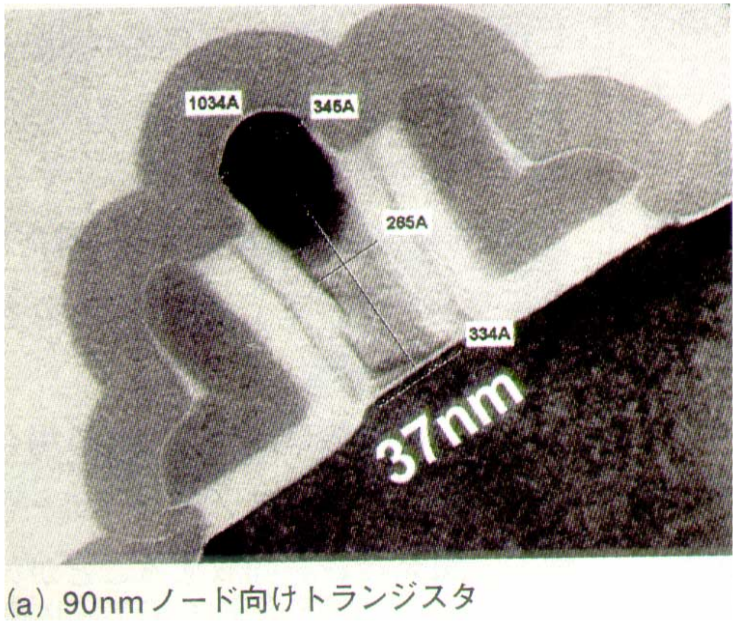


TSMC
0.13μ

Source: Nikkei Microdevices 11/00

TI
0.09μ

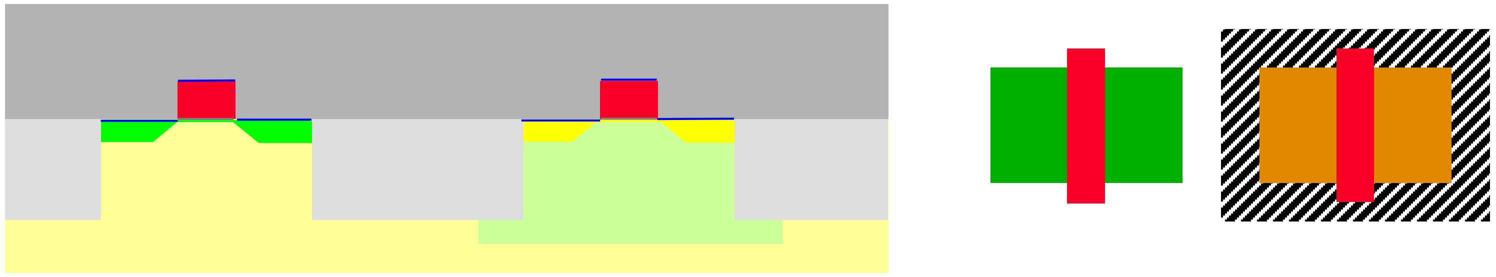
Source: Nikkei Microdevices 9/02



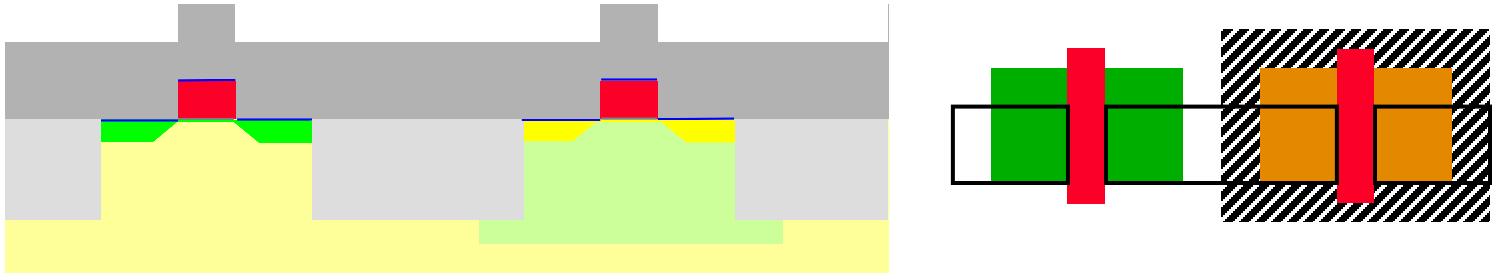
- Much more complex structure generated from drawn polysilicon gate

Making Wires - 1

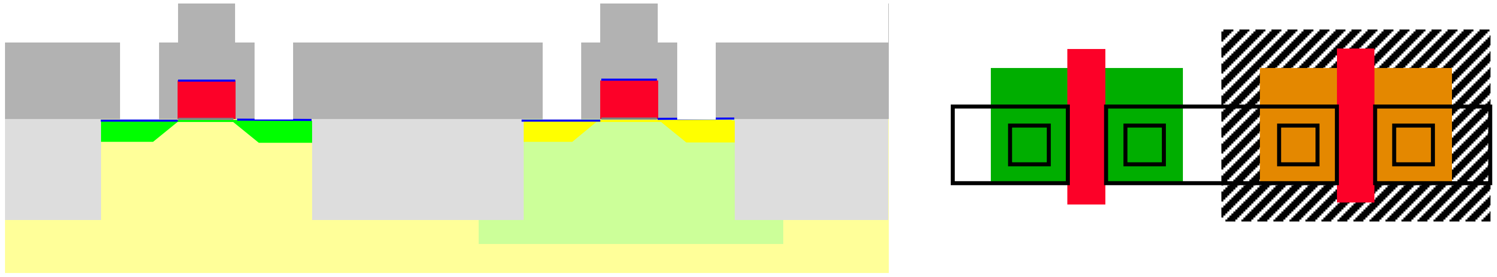
Deposit Dielectric



Etch metal trenches

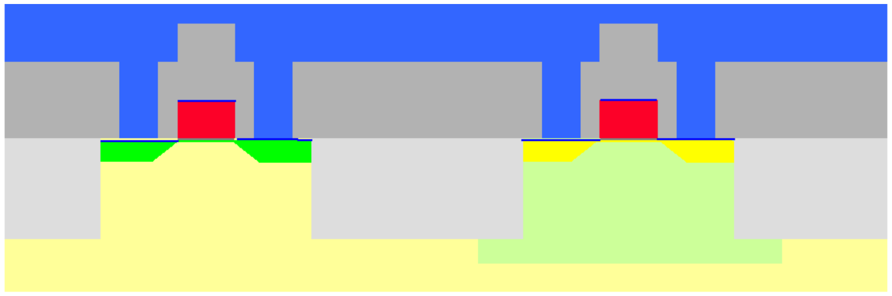


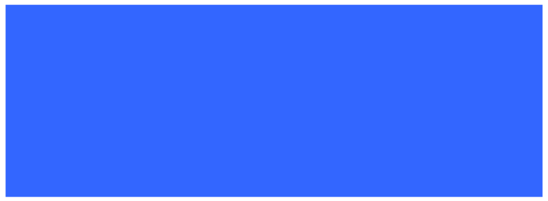
Etch via trench



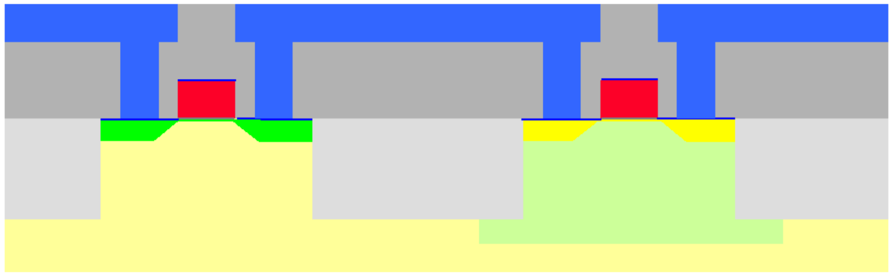
Making Wires - 2

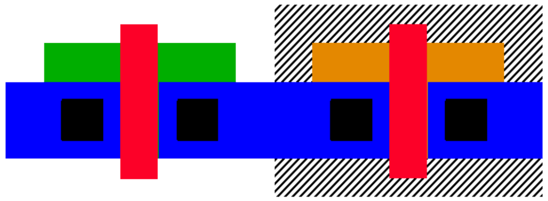
Deposit metal





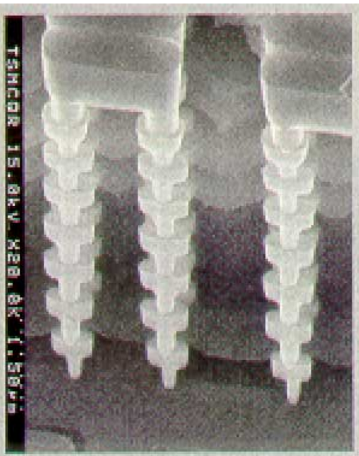
Polish surface back





- Same steps can be repeated to add additional metal layers

Actual Wires



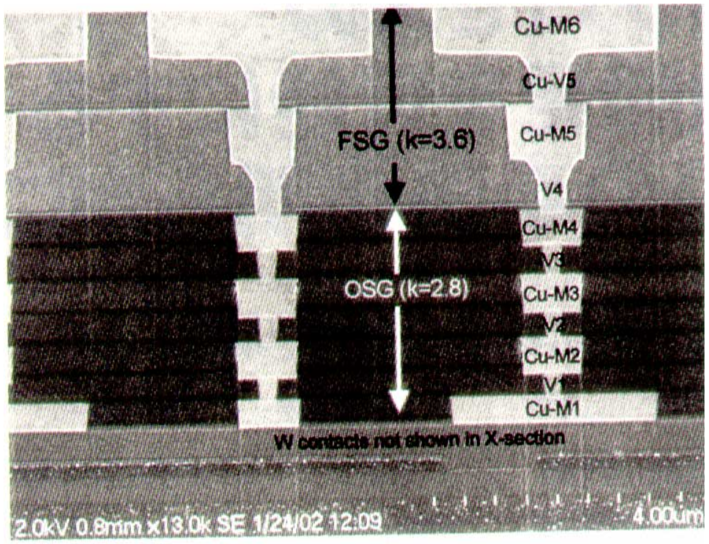
(b) 0.13μmプロセスで製造した8層Cu配線

TSMC
0.13μ
8 layers
Cu

Source: Nikkei Microdevices 11/00

TI
0.09μ
6 layers
Cu

Source: Nikkei Microdevices 9/02



(b) 90nmノード向け多層配線

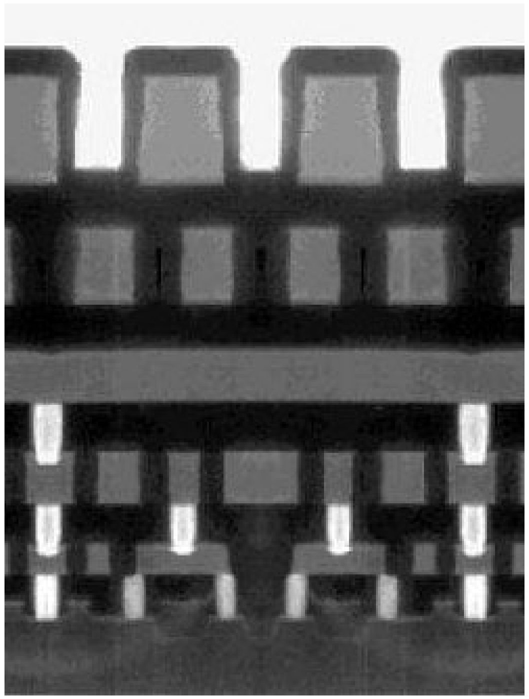
- 6-10 layers of metal
- Vias and wires manufactured at same time (dual damascene)
- Top levels are thicker for power distribution
- Interlayer dielectrics are not all SIO_2 ($\varepsilon_r < 3.9$)

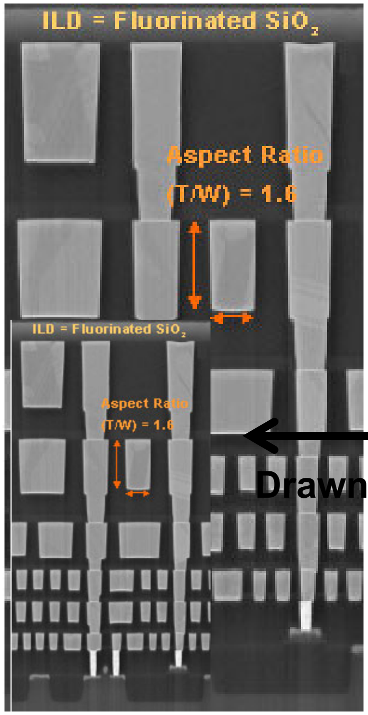
Wire Scaling

As chips scale, wires have increased in number and complexity

- Cross-sectional areas and spacings have decreased

Intel's 0.25µm process, 1997



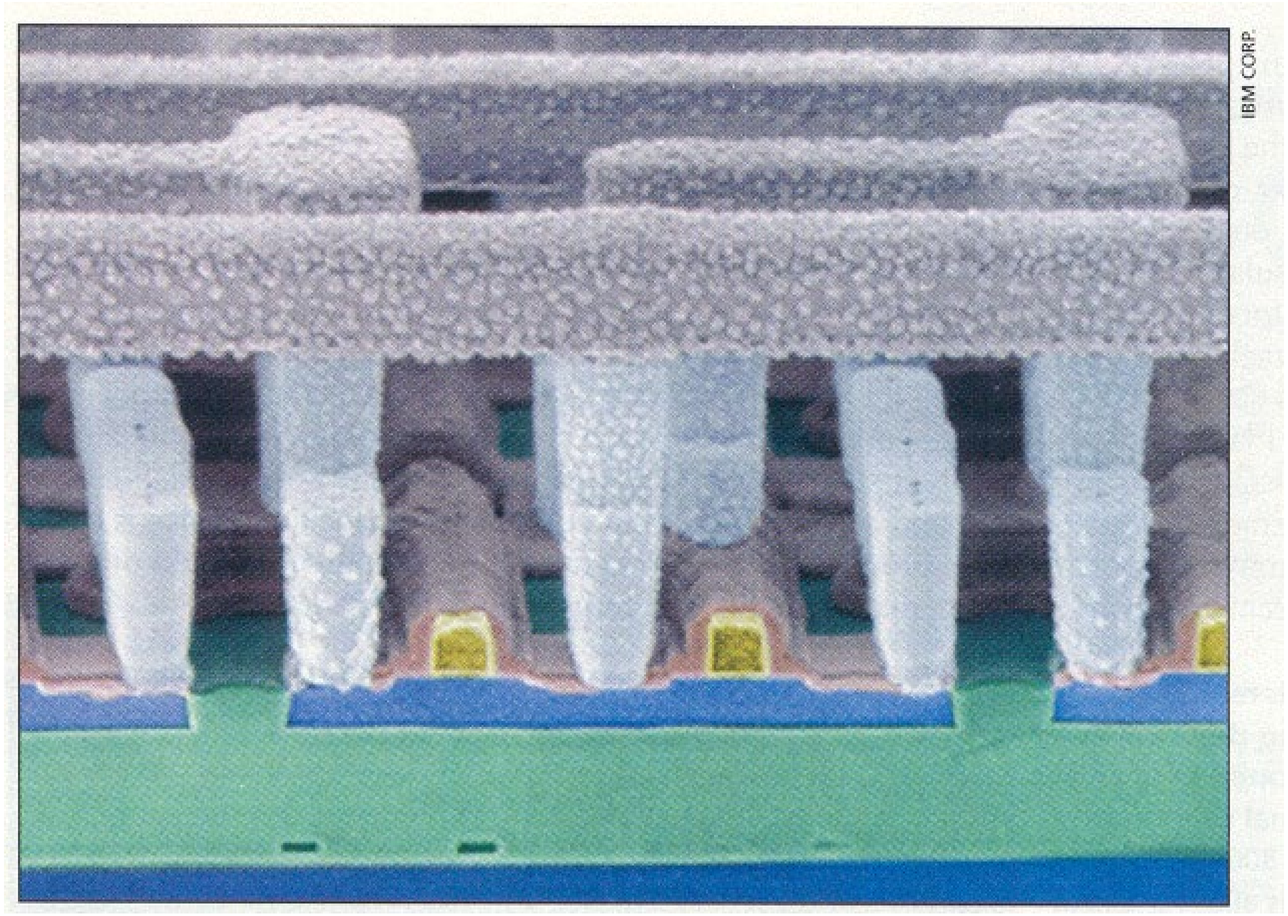


Intel's 0.13µm process, 2002

Drawn to scale

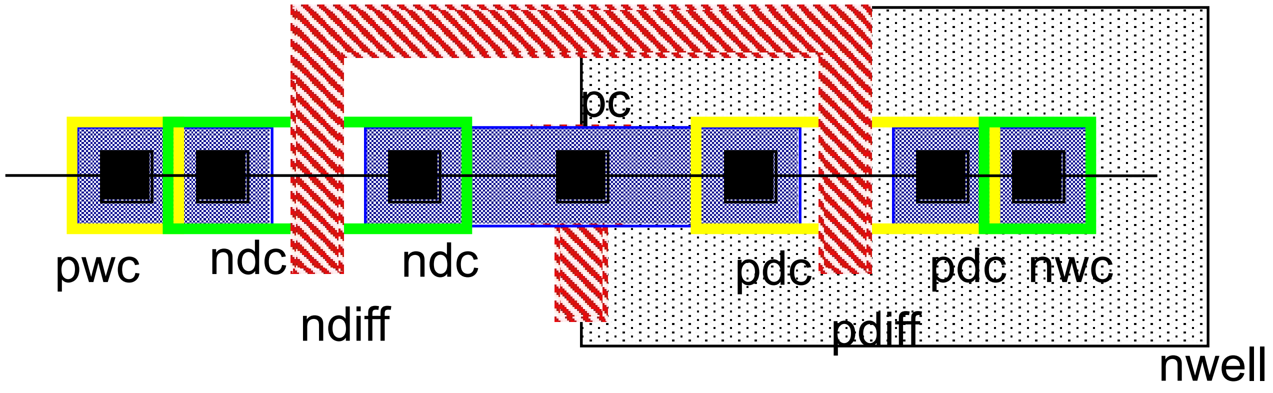
Source: www.intel.com

Actual Chip

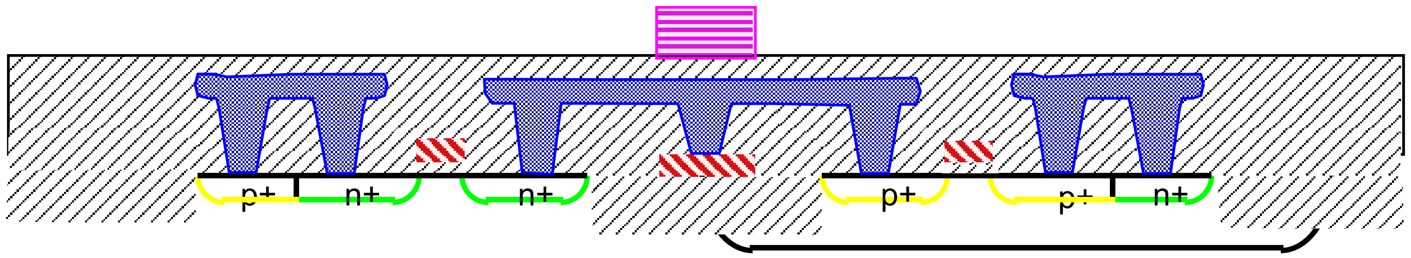


Your Job

- Create the artwork that looks like this:

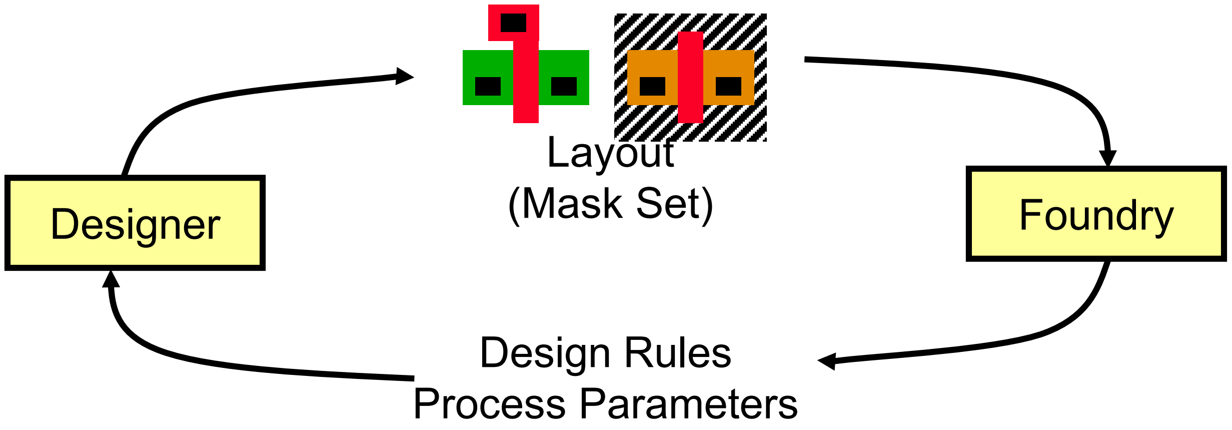


- So the fab can create something like this (in cross-section).



Abstraction and Fabrication

- Don't know (or want to know) all the details of fabrication
- Abstract way unnecessary detail and provide simplified user interface
  - Drawing layers: Devise simplest set that provides required information
  - Design rules: Requirements on relationships between shapes
  - Process parameters: transistor performance, conductor thickness and resistivity, dielectric thickness and permittivity



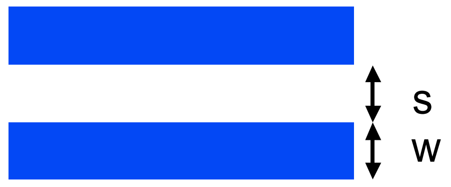
Fabrication Constraints on Layout

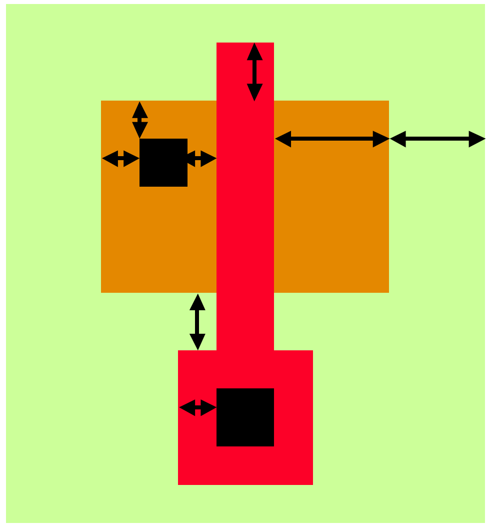
Fabrication places many constraints on the layout

- The most fundamental are:
  - Resolution constraints
    - What is the smallest width feature than can be printed
    - What is the smallest spacing that will guarantee no shorts
  - Alignment/overlap constraints
    - Like printing a color picture, need to align layers to each other
    - Generally need a little tolerance to be manufacturable
- In modern technologies there are many other rules too
  - Density rules
  - Proximity rules
  - Antenna rules

Geometric Design Rules

- Resolution
  - width and spacing of lines
  - smaller than λ of light
    - Makes everything hard
- Alignment
  - to make sure interacting layers overlap (or don't)
  - contact surround
  - poly overlap of diff
  - well surround of diff
  - contact spacing to unrelated geometry





SCMOS Lambda (λ) Design Rules

- MOSIS SCMOS design rules
  - They are a simplified set of rules
  - Allow you to send your designs to a number of fab lines
  - Rules are based on λ, a type of scaling constant
  - λ was initially 1.5 μ, and now it is < 0.022 μ

- Ignores some of the ways you used to save area
  - Use only Manhattan Layouts
  - Use only 90^o angles
  - But today's technology is more restrictive

SCMOS Design Rule Highlights

Resolution rules:

LAYER	WIDTH	SPACE
poly	2	3
diff	3	3
metal1	3	3
metal2	3	4
nwell	10	9
cut	2	2
via	2	3

Alignment rules:

cut/via surround	1
poly overlap diff	2
poly space to diff	1

Notes:

Cut plus surround is 4
causes layout to fall on an 8λ grid

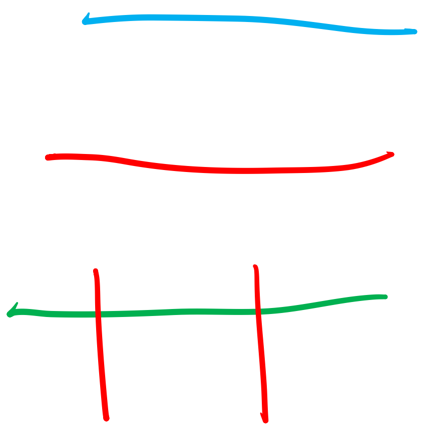
Layout is Fun, But Time Consuming

- Problem with layout is everything has to be consistent
  - If you missed a wire, you need to make room for it

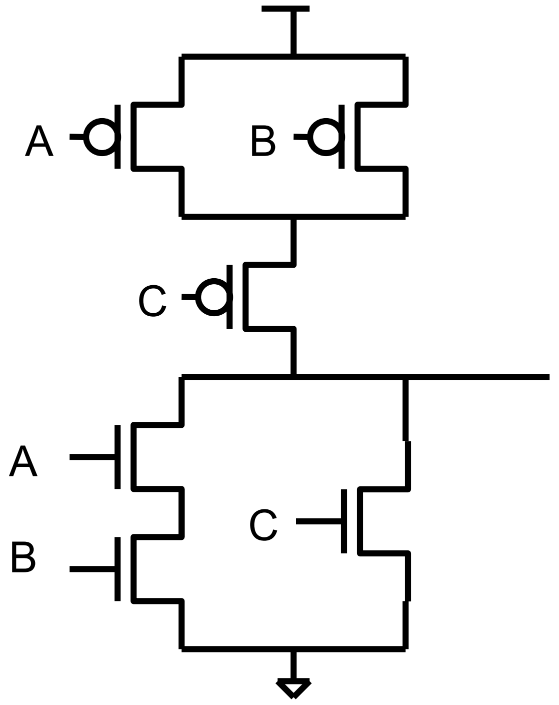
- It is good to do some fast planning first
  - Usually technique is called stick diagrams

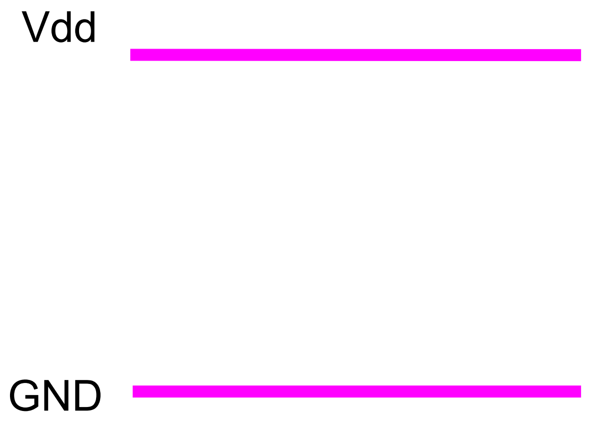
Stick Diagrams

- Abstract version of layout
- Work out topology
  - lines (wires) drawn in color to denote a particular layer
  - lines on same layer cannot cross
  - lines drawn with no thickness
  - approximate relative spacing
- Use colored pencils



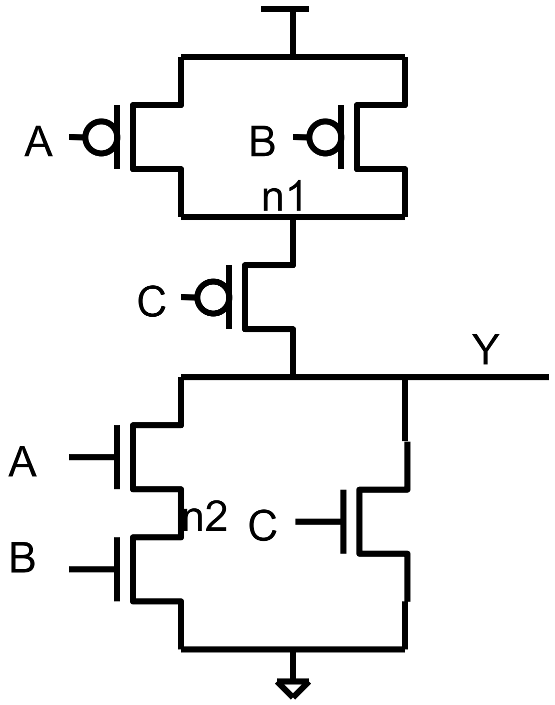
An Example – Step 1

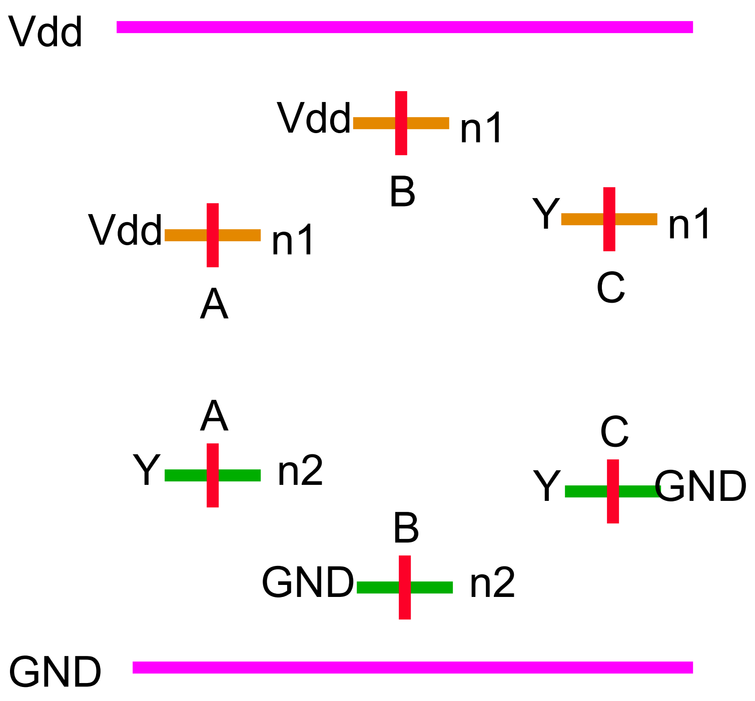




- Choose global directions for routing layers
  - Position power lines in top layer of metal

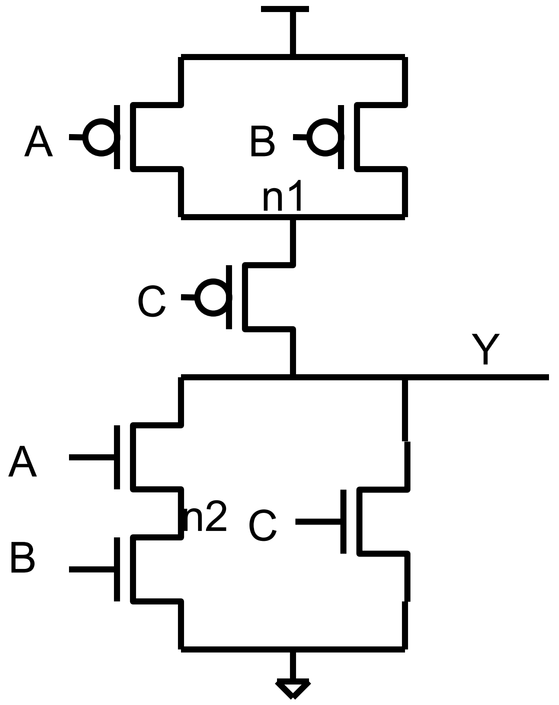
An Example – Step 2

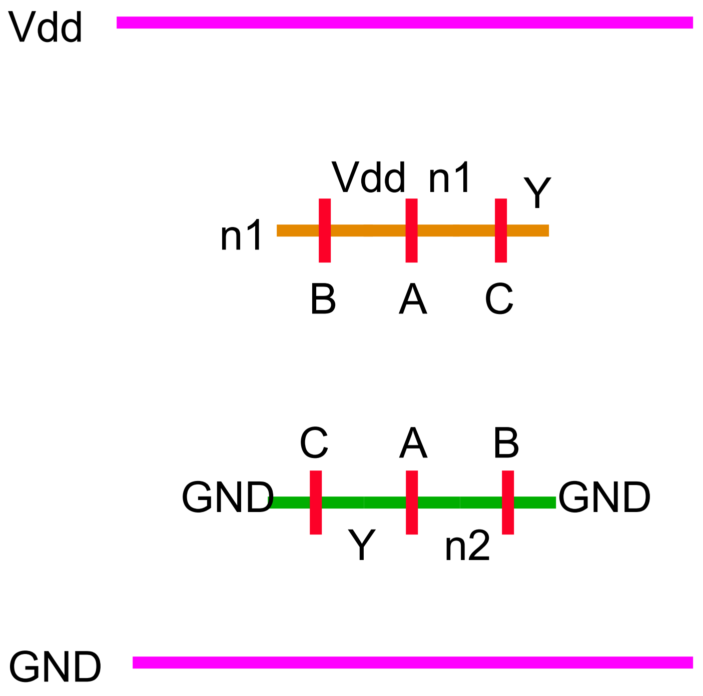




- Cluster together NMOS with NMOS and PMOS with PMOS
- Generally keep gate orientation the same

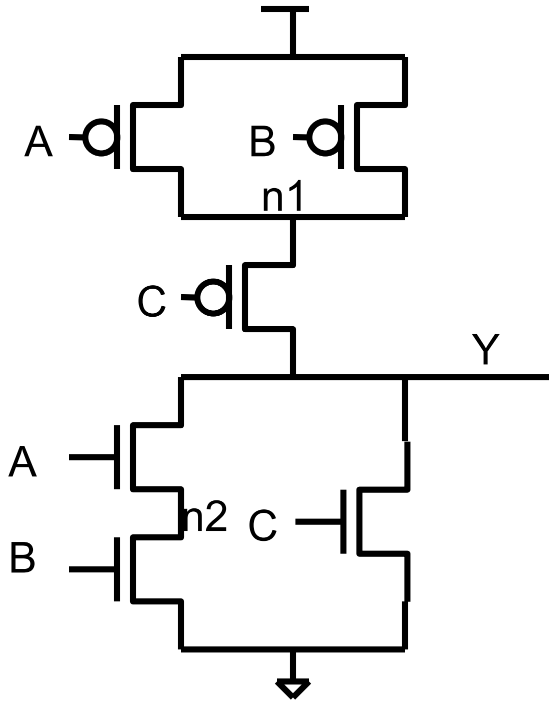
An Example – Step 3

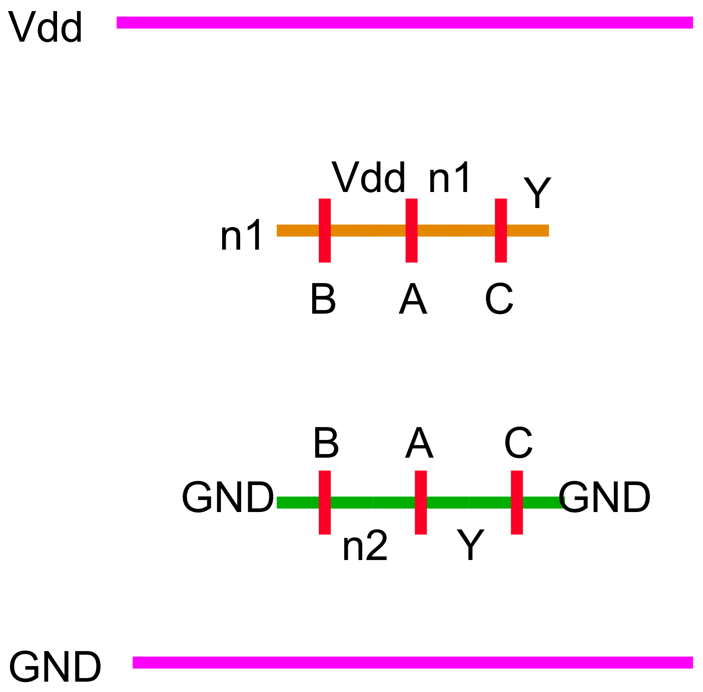




- Arrange transistors so that common sources/drains can be shared
  - Give precedence to shared signals over shared vdd/ground

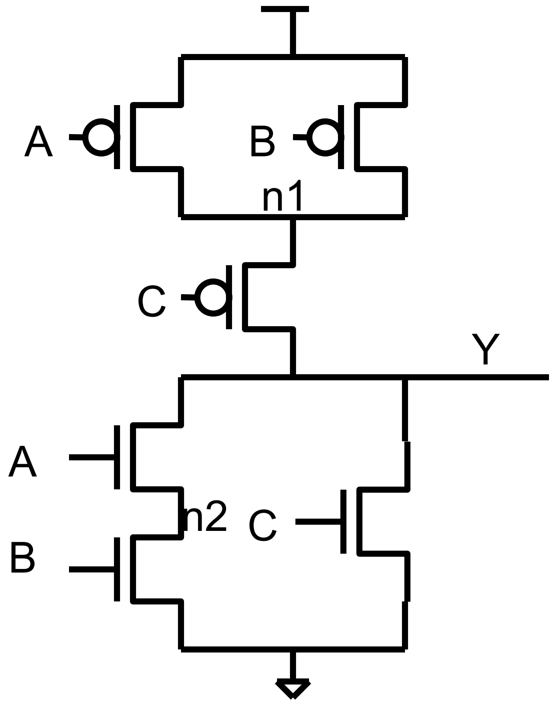
An Example – Step 4

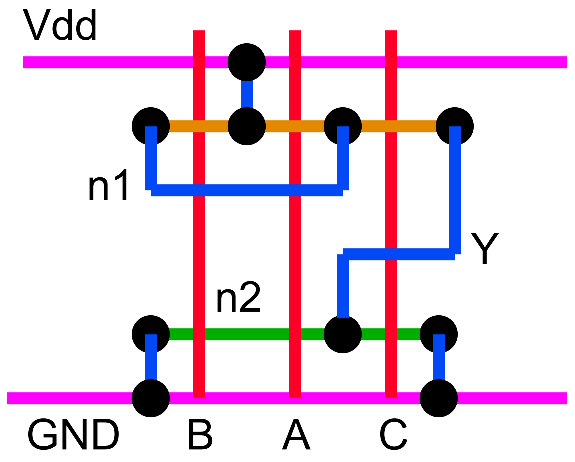




- Arrange transistors so that common gates line up

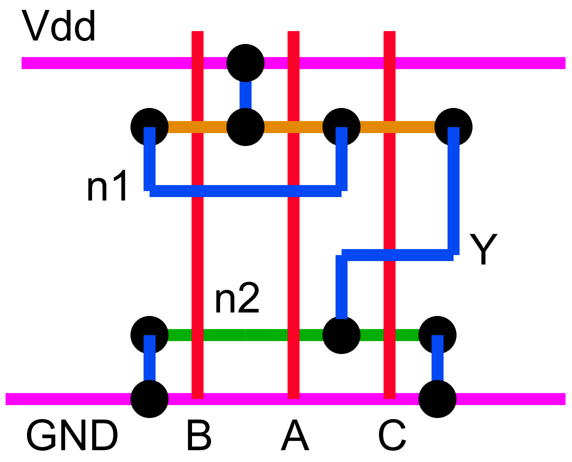
An Example – Step 5

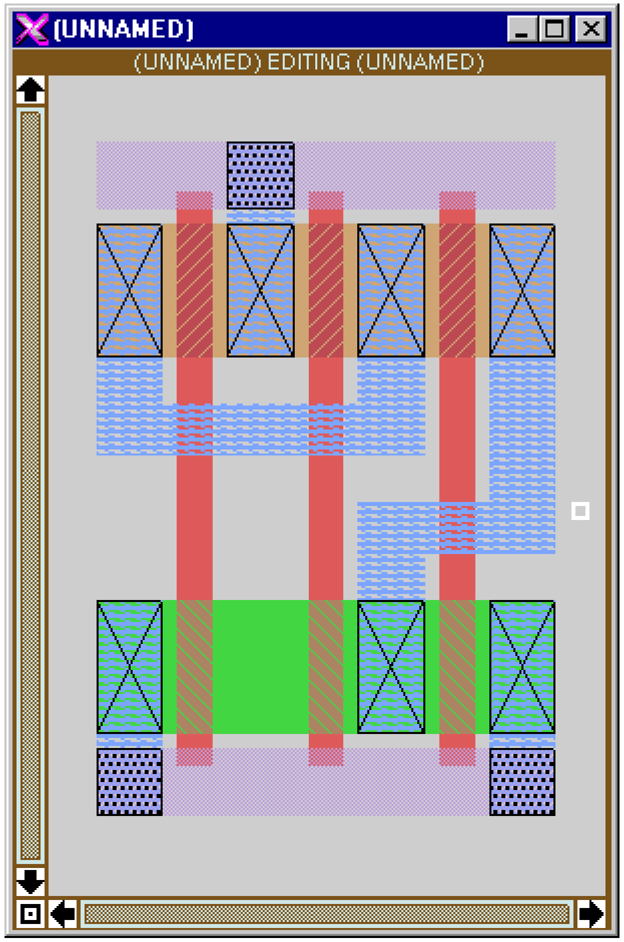




- Connect everything up

Convert to Real Layout



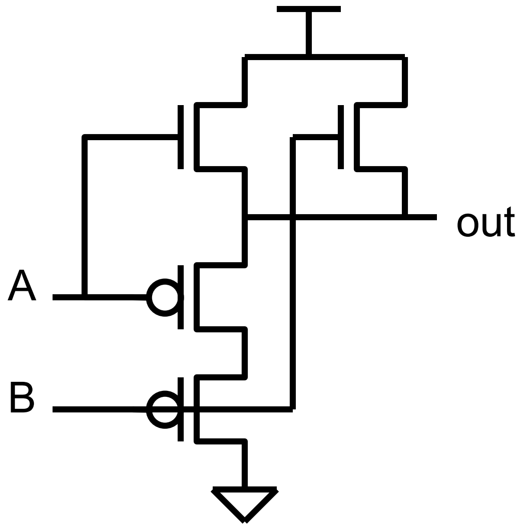


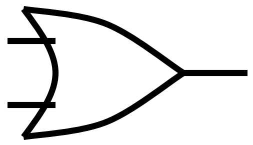
Basic Layout Planning

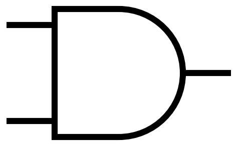
- Choose global directions for routing layers
  - Adjacent levels should route perpendicular
  - Example: m2 horizontal, m1 vertical
- Position power lines first in top layer of metal
- Cluster together NMOS with NMOS and PMOS with PMOS
  - Generally keep gate orientation the same
- Arrange transistors so that common sources/drains are shared
- Arrange transistors so that common gates line up
- Limit lengths of diffusion and poly routing – use metal

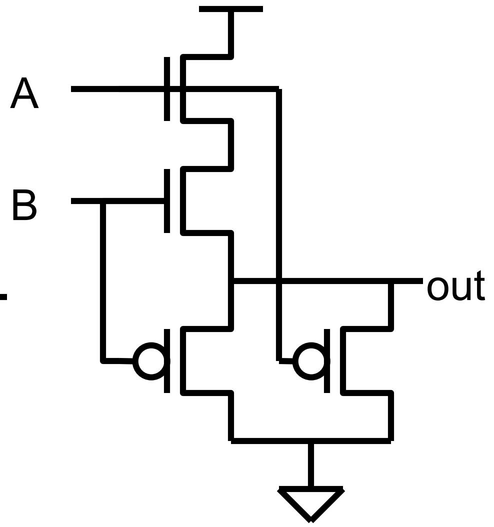
- Use repetition/tools (try to design as little as possible)
  - Critical issue

For Next Time: How About AND & OR?









- Seems to obey the rules
  - Output always driven by pMOS or nMOS network
  - Vdd and GND never shorted together

- Doesn’t work in reality
  - Transistors aren’t really switches