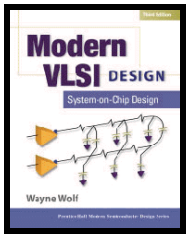
Slide Set 2

Introduction to Layout and the CMOS Gates

Steve Wilton
Dept. of ECE
University of British Columbia
stevew@ece.ubc.ca

Overview

- Reading
  - Wolf 2.6, 3.1-3.3



- Introduction

  This class concerns the design and layout of VLSI circuits. In this lecture, we will first examine the characteristics of an actual layout to give us an idea of what we are shooting for. Then, we will look at the basic operation of an NMOS transistor, and look at how it can be used to implement simple digital functions. Finally, we will look at a PMOS transistor and example how it can be used to create more efficient gates.

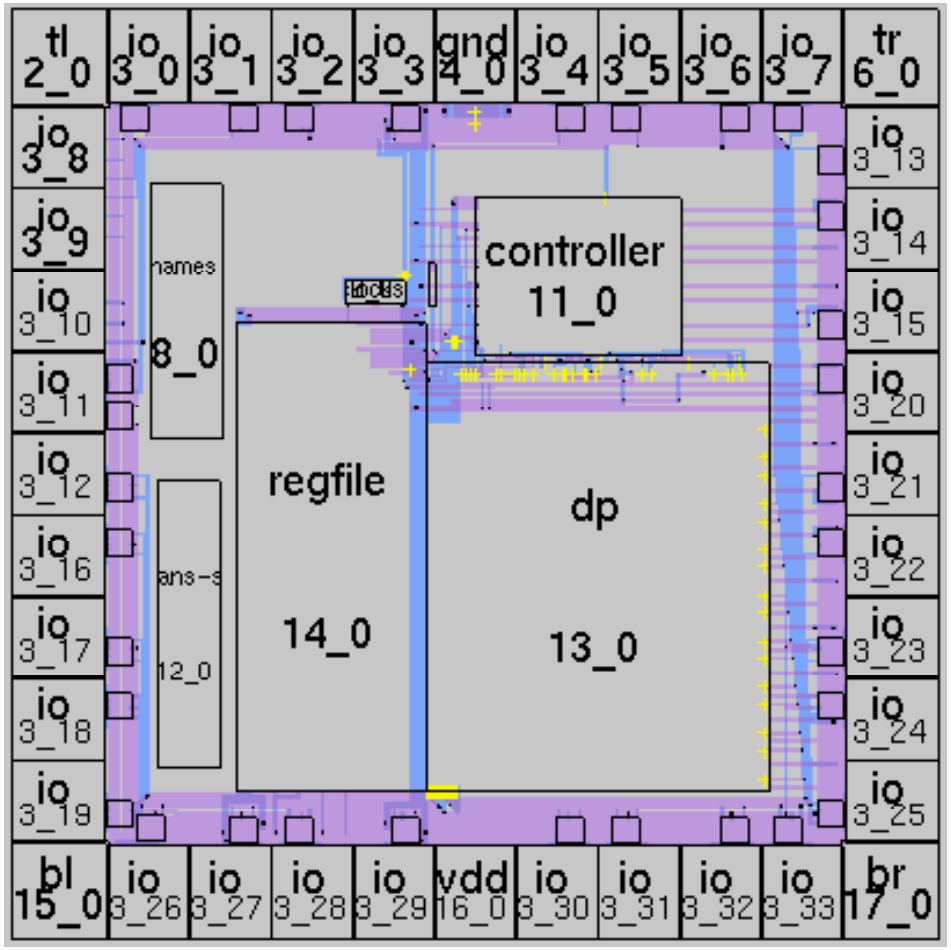
Layout: The Fabrication Specification

The end of the design process must create a set of drawings, one for each layer needed in the manufacturing process

- Layout drawing are complicated
  - There are many rules about the geometry to make sure the circuits can be reliably manufactured
    - Minimum width of wire, minimum spacing between wires, alignment rules
  - The layers represent transistors and wires, and need to create the correct function
  - Many rectangles for each transistor and wire, and there are millions of transistors and wires.
- Different layers are represented by different colors
  - People used to draw the layout on mylar (10s of transistors)
  - But not any more, now use CAD tools, and premade cells.

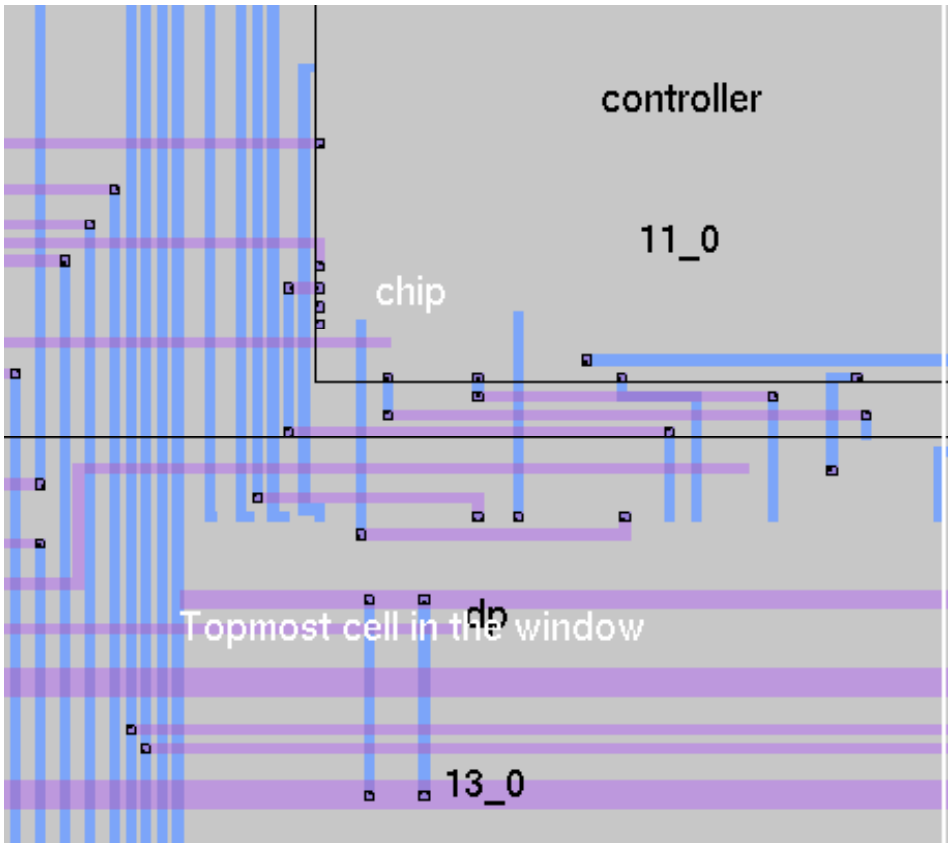
Layout Example

- Example from previous student project
- Use hierarchy to hide complexity
- Pads around chip
- Major blocks are shown
- Coloured regions are really many wires



Layout

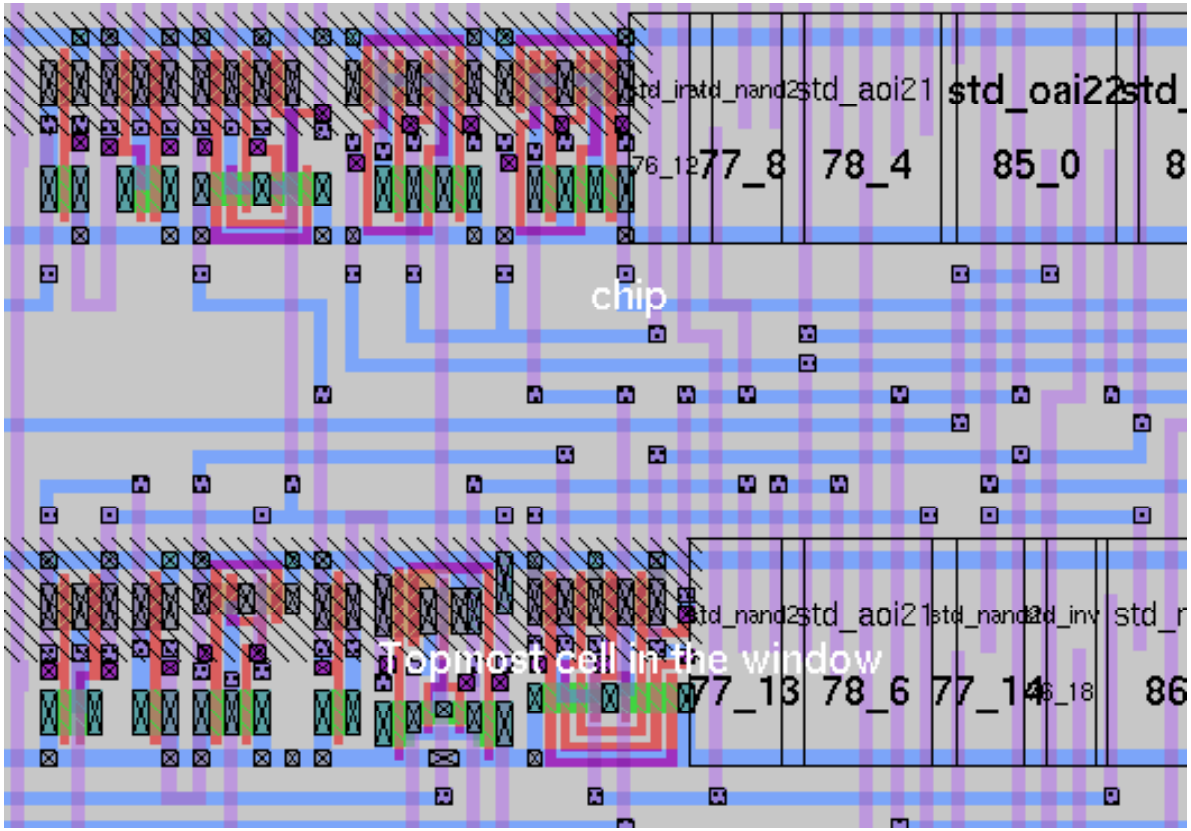
- This picture is an expanded view of a portion of the layout of the other page.
- The next two slides will look at the controller layout and some layout in the datapath



Controller Layout

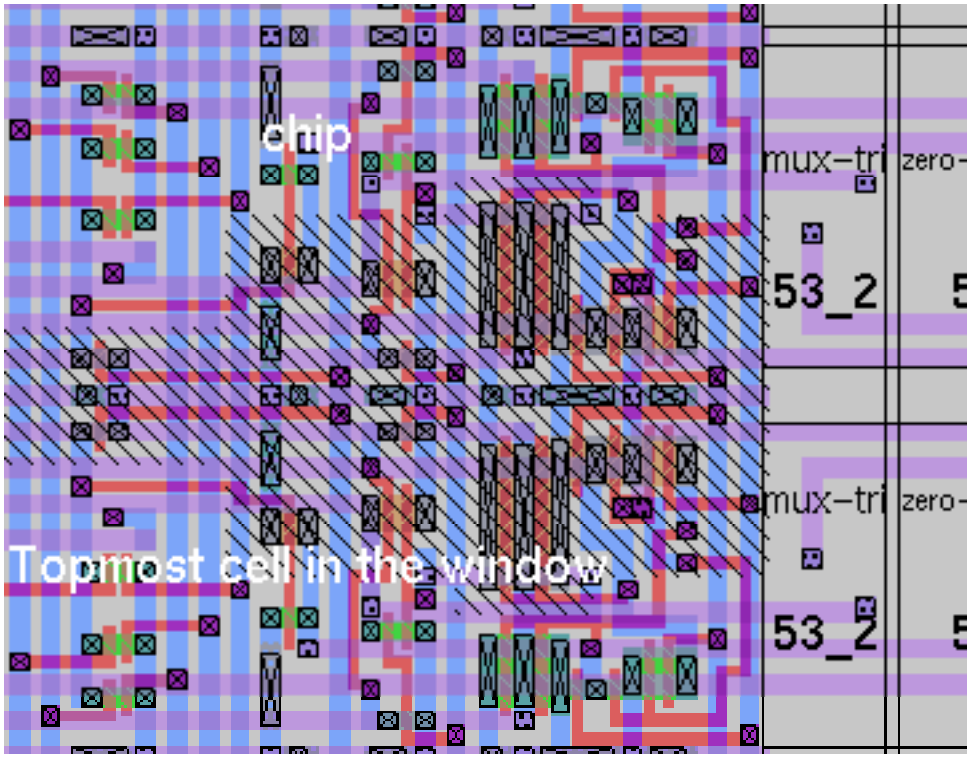
- Right half shows cells in the design
- Left half has the cells expanded to show the layout layers

- This design style has random wires



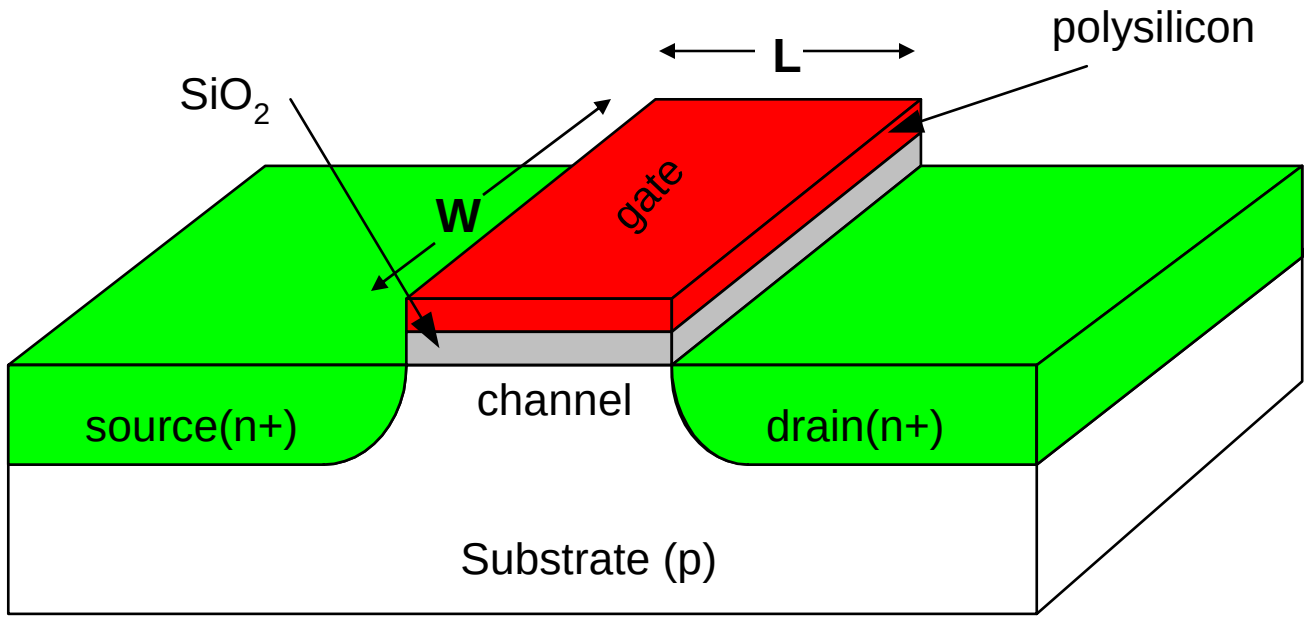
Datapath Layout

- Wires here are more regular
- Again
  - Cells on right
  - Expanded cells on left
- Transistor density is higher

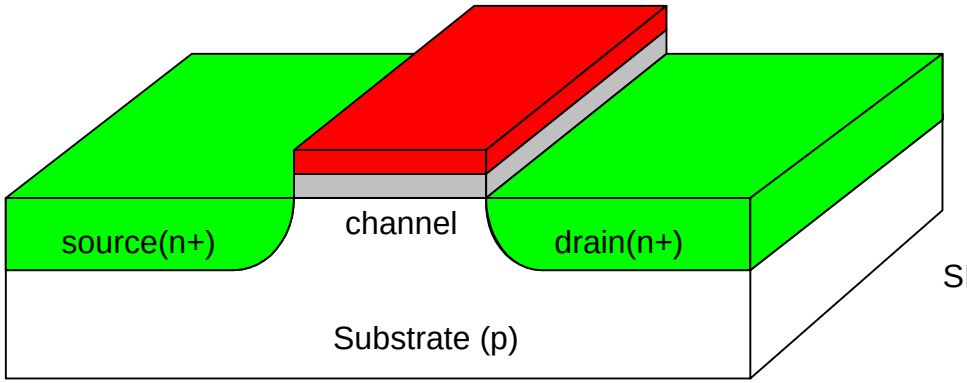


The NMOS Transistor

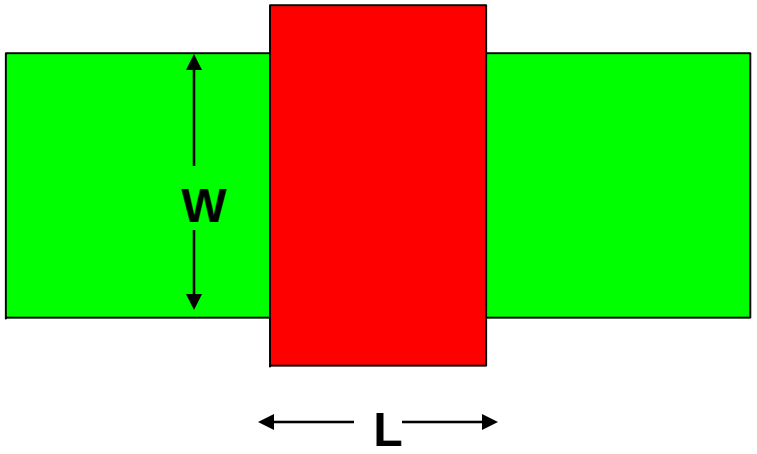
This is a basic NMOS transistor with a gate, source, and drain



The NMOS Transistor

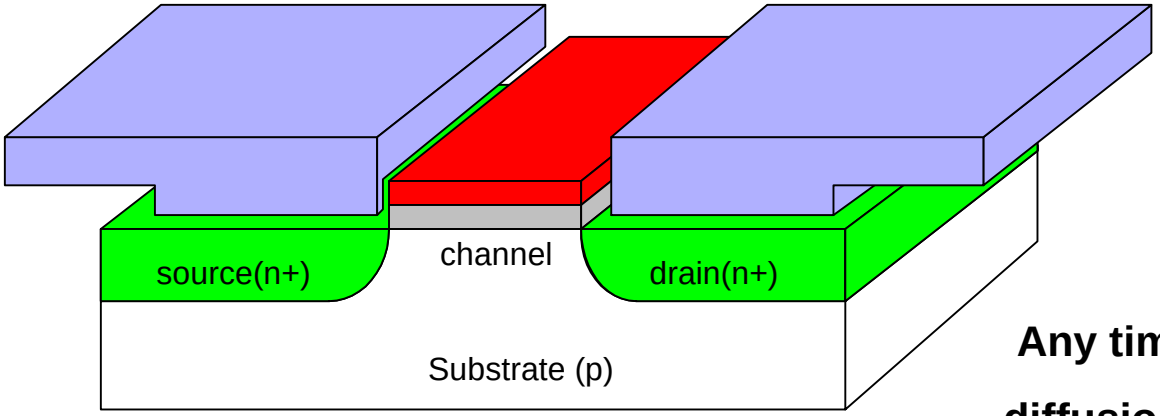


SIDE VIEW

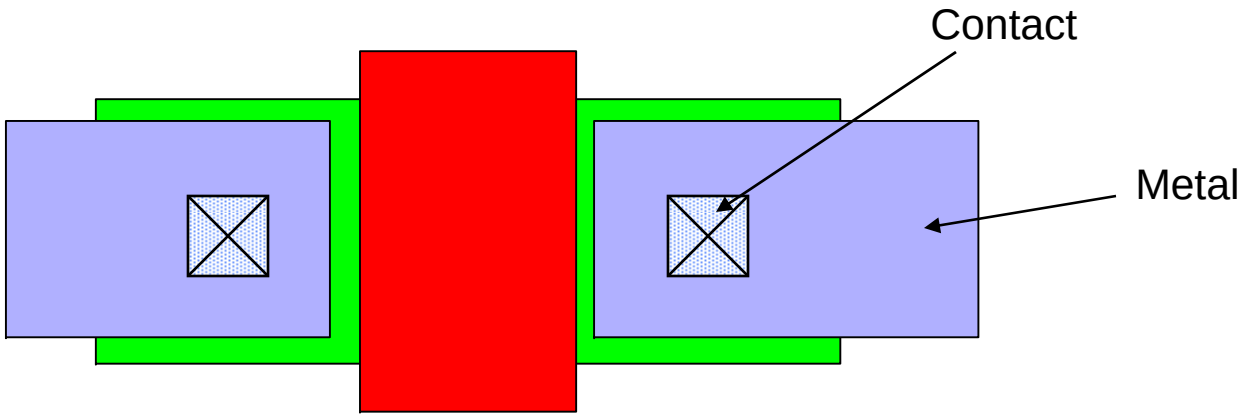


TOP VIEW

The NMOS Transistor

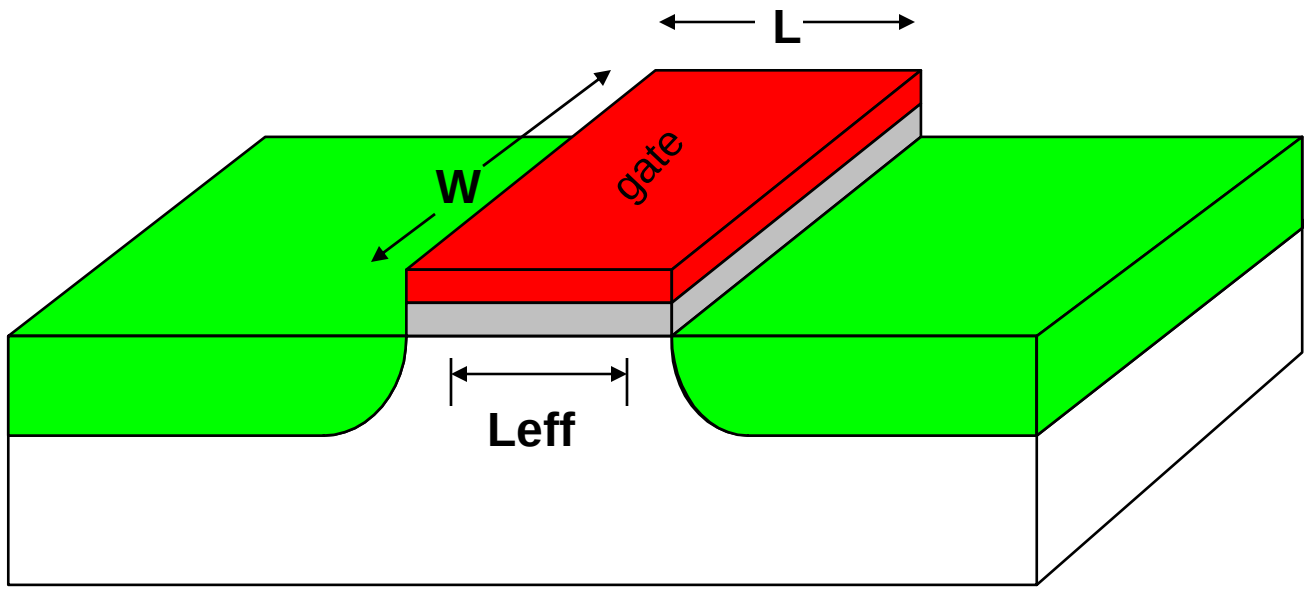


Any time polysilicon crosses diffusion, we have a transistor



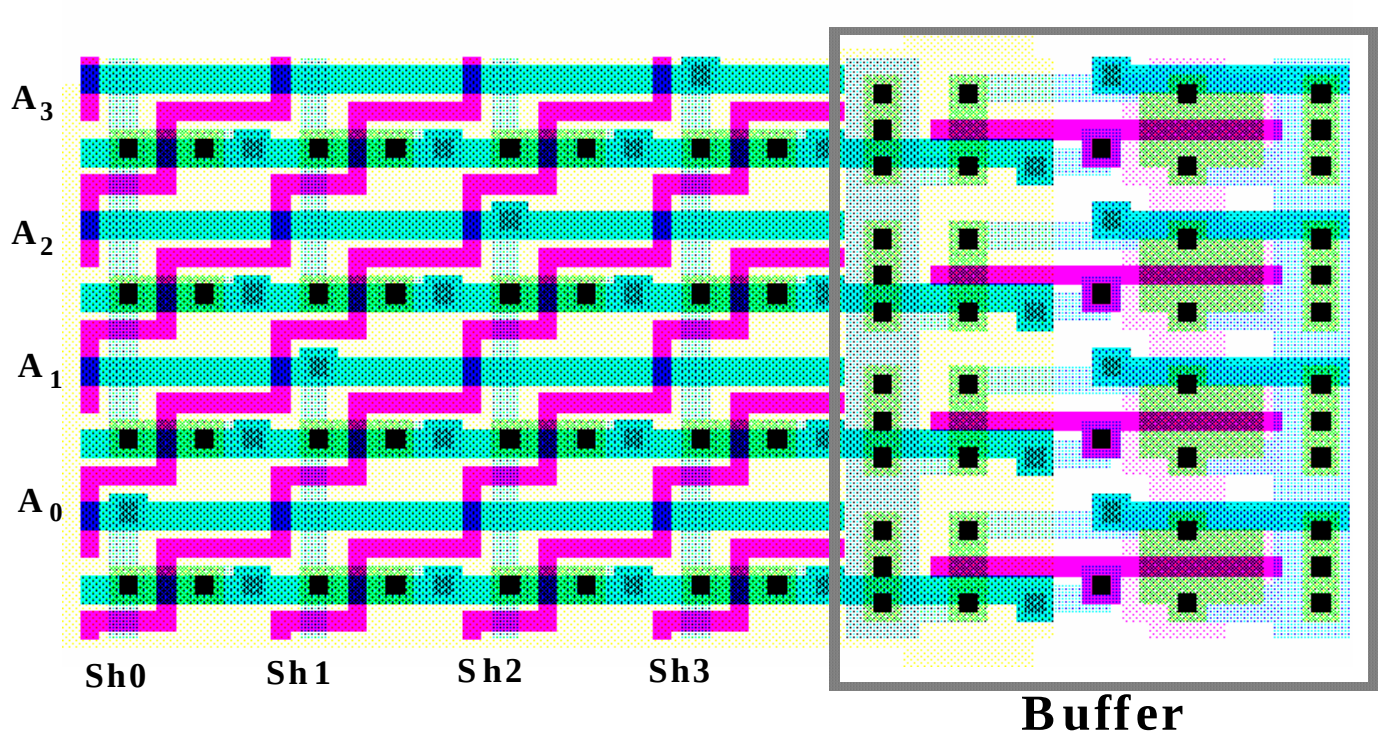
Leff vs. L

Due to fields around source and drain, the “effective” L is smaller than L



An Aside:

Soon, we will be designing complex layouts, like this:



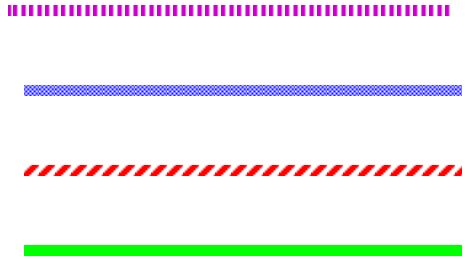
To make it easier to draw, we will use “sticks diagrams”

Stick Diagrams

- Stick diagrams are a simplified version of layout
  - Abstract the layout so wires are just lines
    - Don't worry about width or spacing
    - Just draw the center line of the wire
  - Spacing on different parts of the page need not be the same
    - Sneak another wire in when needed, without needing to redraw the whole layout
    - But try to keep spacing the same
      (since it will better estimate the real layout)
  - Good starting point before doing layout
    - But like most things, after you do some layout, you will have a better feeling for how to draw useful stick diagrams
- We will use stick diagrams often to demo stuff …

Wire Layers

- We represent the different wiring layers with different colors
  - metal2 - purple / orange
  - metal1 - blue
  - poly - red
  - diff - green / yellow

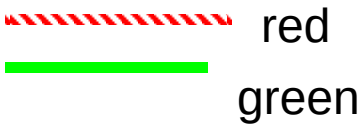


- Wires on the same layer that touch ALWAYS connect. There is no way to jumper a wire without changing layers.
- Wires on different layers can cross without connections. To form connections between different layers you need to explicitly draw a contact.

Transistors

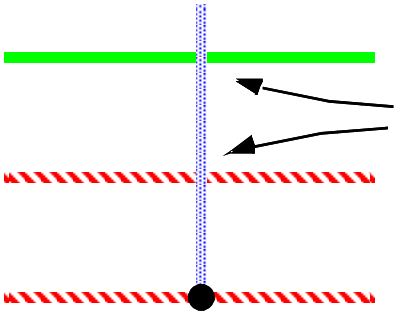
Are formed when poly (red) crosses diffusion (green or yellow).

(lots of fab steps to make it seem that simple)



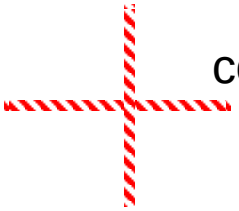
red

green

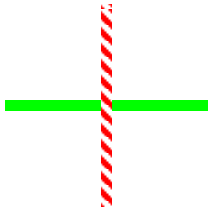


no connection

connection



connected

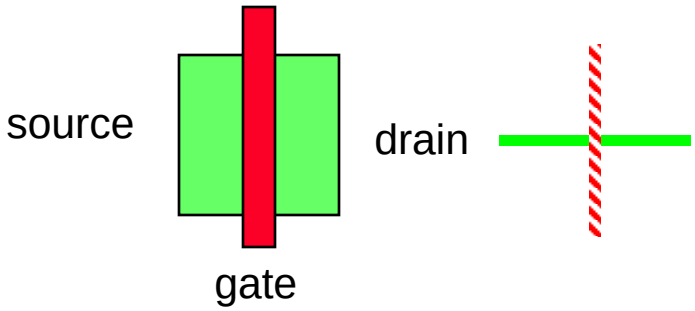


transistor

Simple Model of a MOSFET

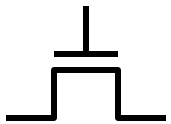
Three terminal device:

- source, drain
  - two ends of conductive path
- gate
  - controls conductive path



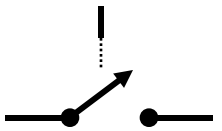
Simple Model of Operation

- conducts when gate is high
- open circuit when gate is low



Caveat

- passes 0s well,
- passes 1s poorly



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.

Gates

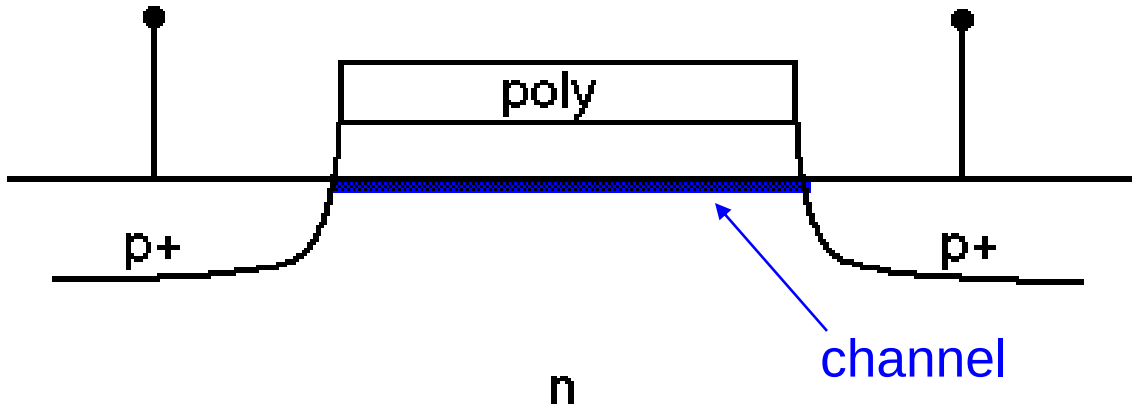
Board Notes: Simple Logic Gates using NMOS transistors

Problem with NMOS Gates: Static Power

Degraded Outputs

PMOS Transistors

Like an NMOS transistor, but polarities are reversed



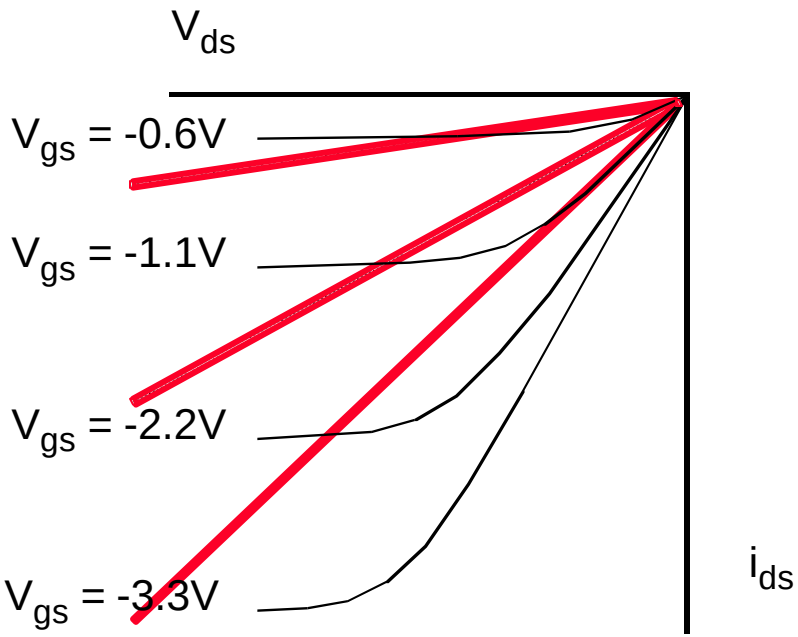
Channel carriers have + charge. Attracted by a negative voltage

Lowering the gate-to-source voltage below V_{th} attracts holes to form a thin p-region. This region allows holes to flow between the two p+ regions. When the channel is not formed, the p+ regions are again isolated by two back to back diodes. pMOS devices are placed in an n-substrate (or well)

Carriers will be flowing from source to drain

PMOS Transistors

Like NMOS but rotated (current and voltages are negative):



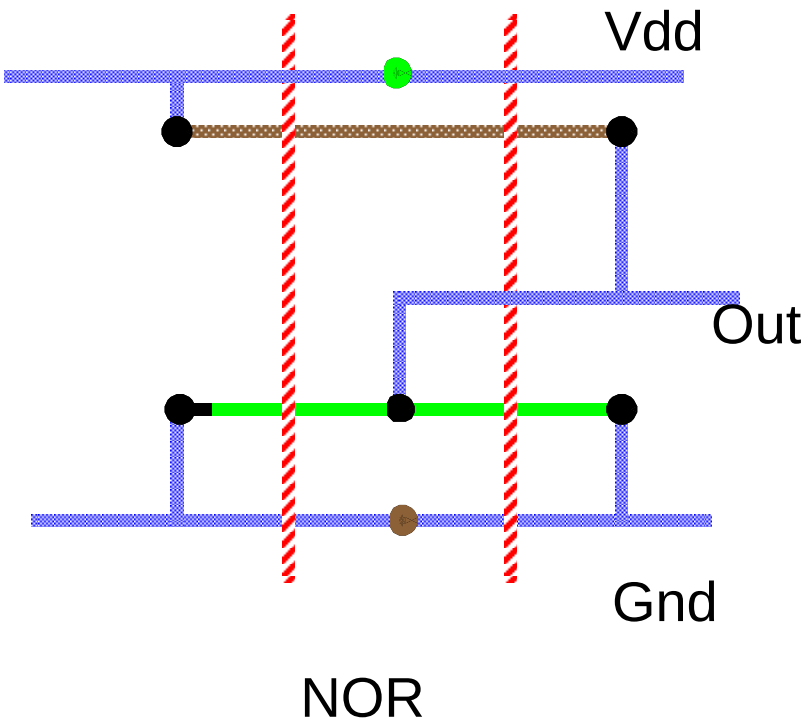
Again we will use a voltage variable resistor approximation for timing calc.

Current is about 1/2 NMOS current since the holes are slower than electrons (mobility is smaller)

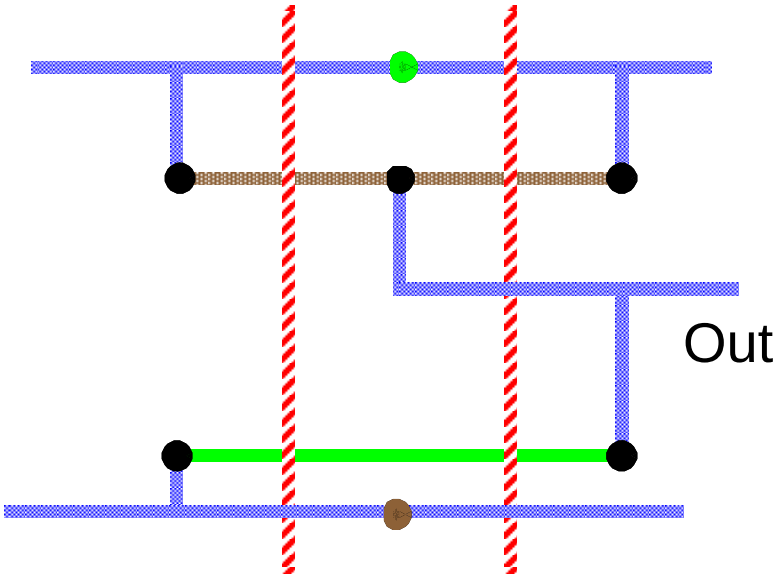
CMOS Gates

Board Notes: CMOS Inverter, NAND, NOR

NOR and NAND stick diagrams



NOR

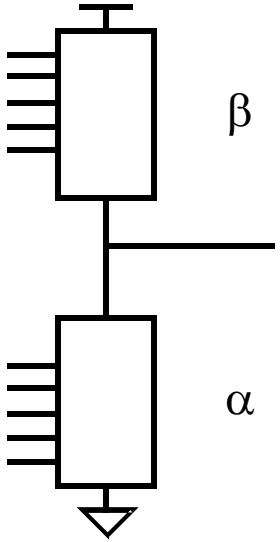


NAND

General CMOS Gates

To build a logic gate $f(x_1, \ldots, x_n)$, need to build two switch networks:

The pullup network connects the output to Vdd when f is false.



β PMOS only

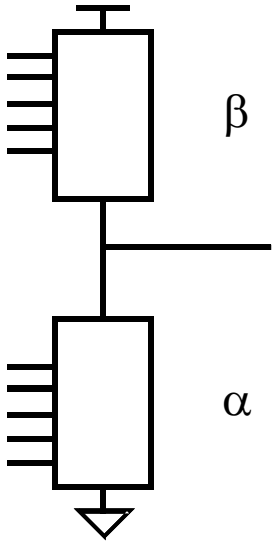
The pulldown network connects the output to Gnd when f is true.

α NMOS only

General CMOS Gates

Rule: The pull-up and the pull-down networks are DUALs

- If one is “on” the other is “off”, and vice-versa



PMOS only

NMOS only

This ensures that:

- Output is always driven with exactly one of: 0 or 1

Recipe for Designing A General CMOS Gate

1. Design a pull-down network that is “closed” when function evaluates to false

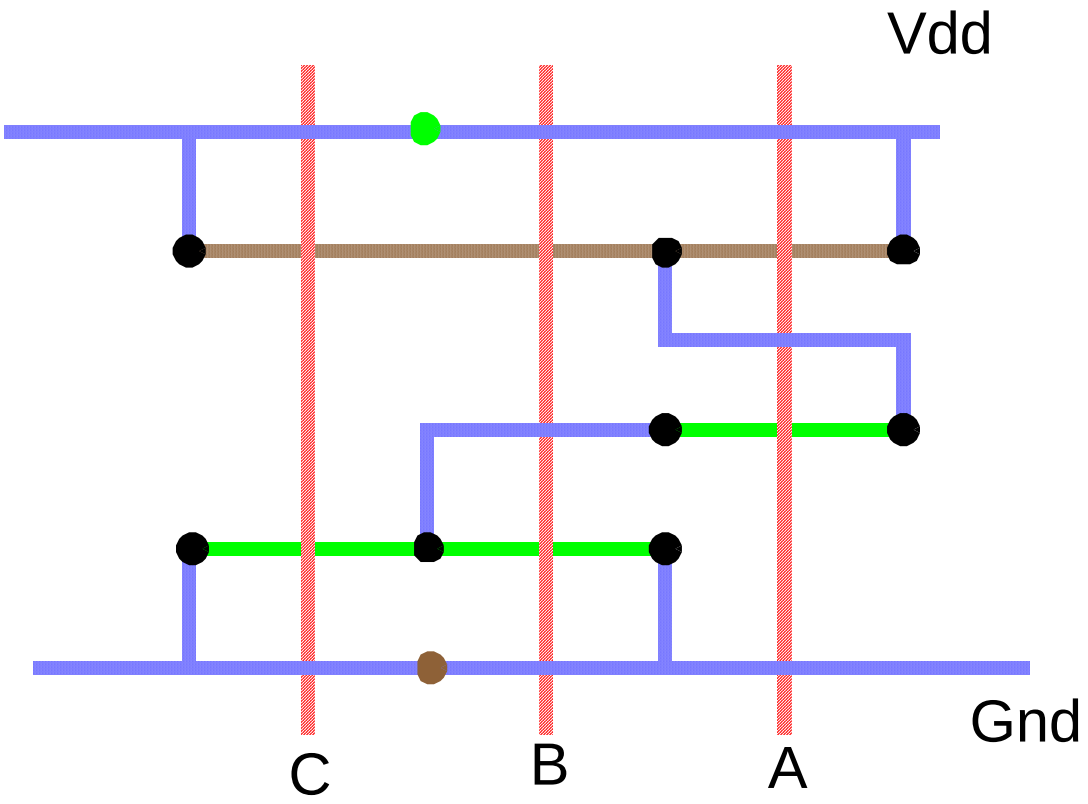
2. Design a pull-up network that is “closed” when function evaluates to true

How do you figure out when a function evaluates to false?
Use Demorgan’s Theorem

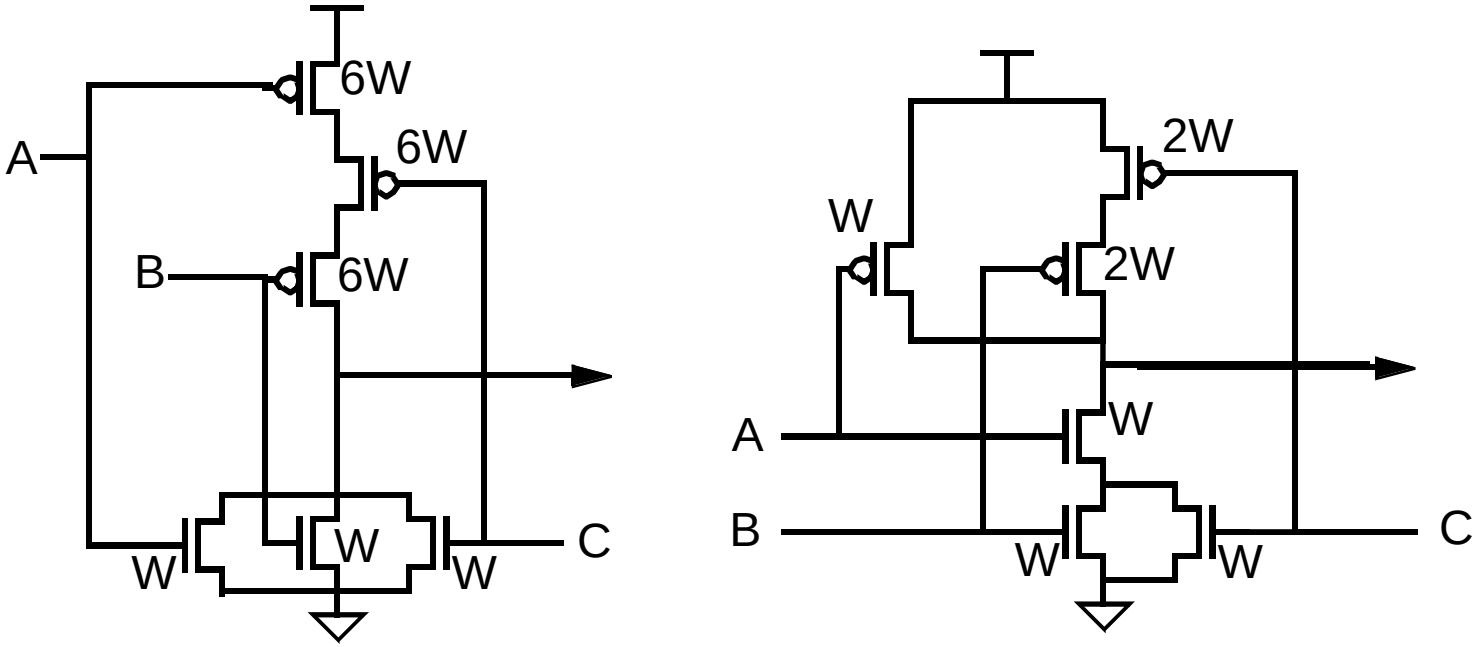
Board Notes: 1. Example for $F = \overline{A*(B+C)}$
2. Can you do F = A*B ?

Stick Layout

Stick layout for $F = \overline{A*(B+C)}$:



Does the size of each transistor matter?



Not for functionality, but transistor sizing will affect speed

Why? Coming soon...

Summary: Good things about CMOS Gates

Biggest Advantage: No Static Power!

- when the gate is not switching, there is *no* current between VDD and GND.
- this wasn't the case for the simple NMOS Gates we saw
- more about power later in the course

Another advantage: Functionality is independent of transistor sizes

- can optimize transistor sizes for speed
- more about this later in the course

For speed and area reasons, fanin of CMOS is limited to 3 or 4 series devices

This means that we can have 3 input NANDs and NORs and occasionally 4 input NANDs and NORs, but should avoid 5 inputs or more.