Slide Set 4

Layout and Design Rules

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

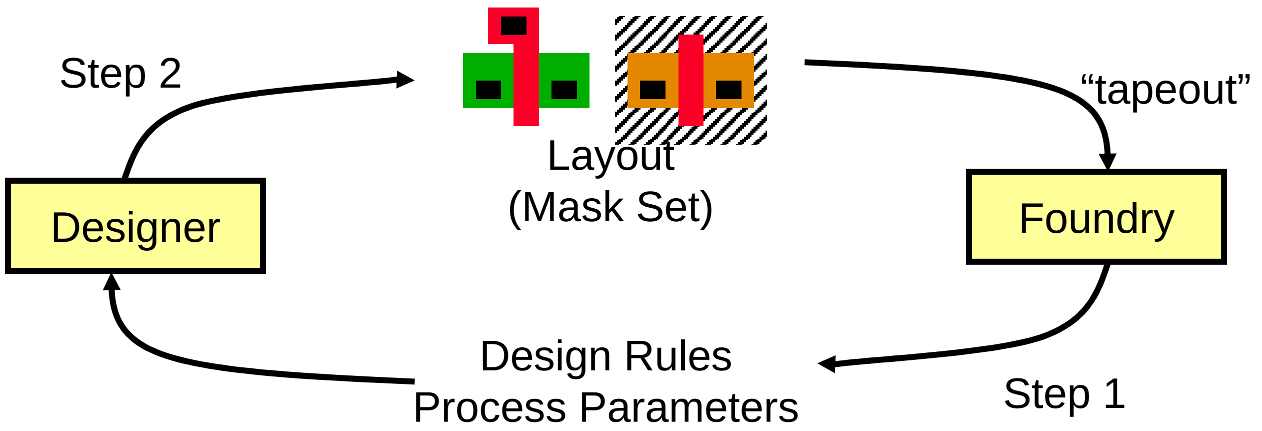
Overview

- Reading
  - Wolf 2.5, 2.6
- Introduction

  The fabrication discussion from last lecture motivates the design rules that need to be followed for layout. In this lecture, we describe a simple layout editor called Magic. This tool is relatively easy to learn and use, and serves as a good tool for layout courses such as EECE479. The different layers and contacts for CMOS designs are described and this leads to a discussion of the design rules needed when laying out integrated circuits. Finally, some methods of estimating the sizes of stick diagrams are presented as the rule of 8λ.

Specifying What to Build

- Now that we know what fabrication is trying to do, how do we tell them precisely what to build? We create a layout (GDS II data)



- We don't care about the real details of the fab, but we have to define the patterning of the layers (that meet the design rules) to specify our design.
- Sometimes knowing more about the fab details is useful when you need to debug a part.

MAGIC Layout Editor

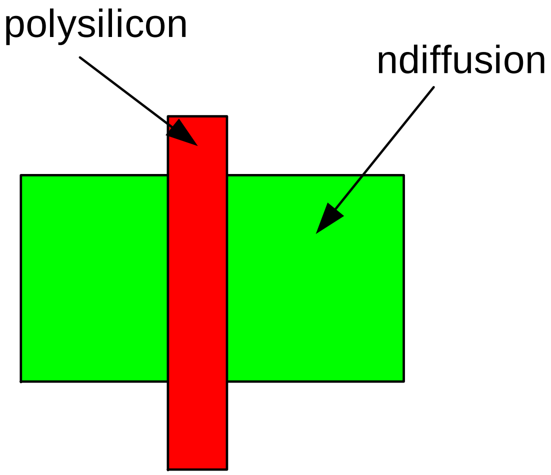
The different layers that a designer uses is generally set by the CAD tool.

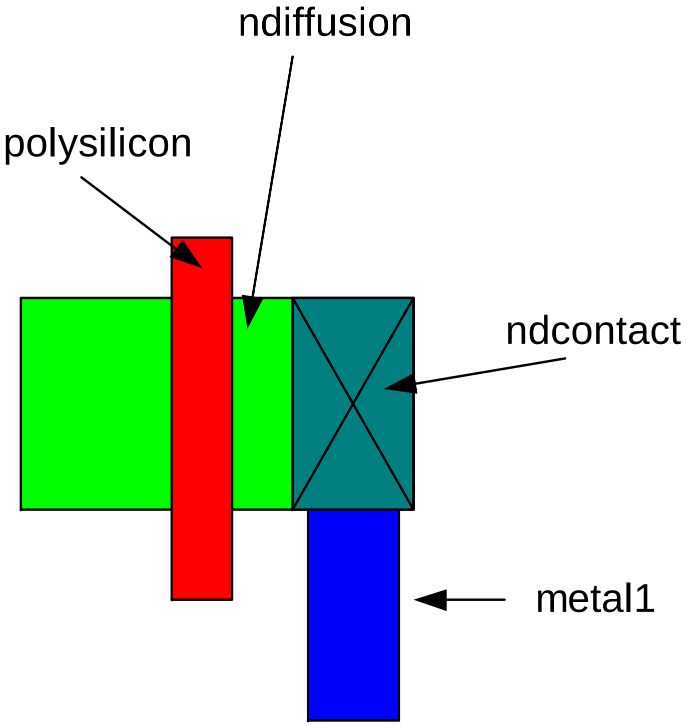
Our layout editor is ‘Magic’

- Magic is quick to learn and commonly available.
- Primarily Paint (Color)-based, not object-based.
- Highlighting of Electrically connected paint.
- Interactive DRC checking.

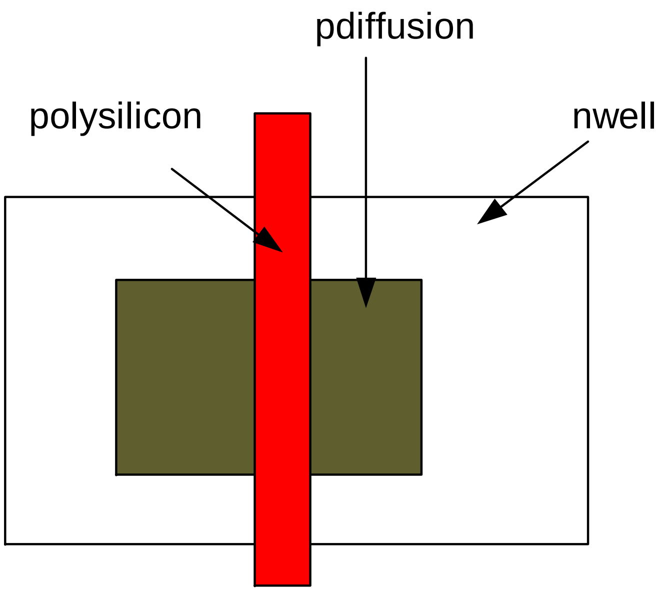
We will use the SCMOS design technology with Magic

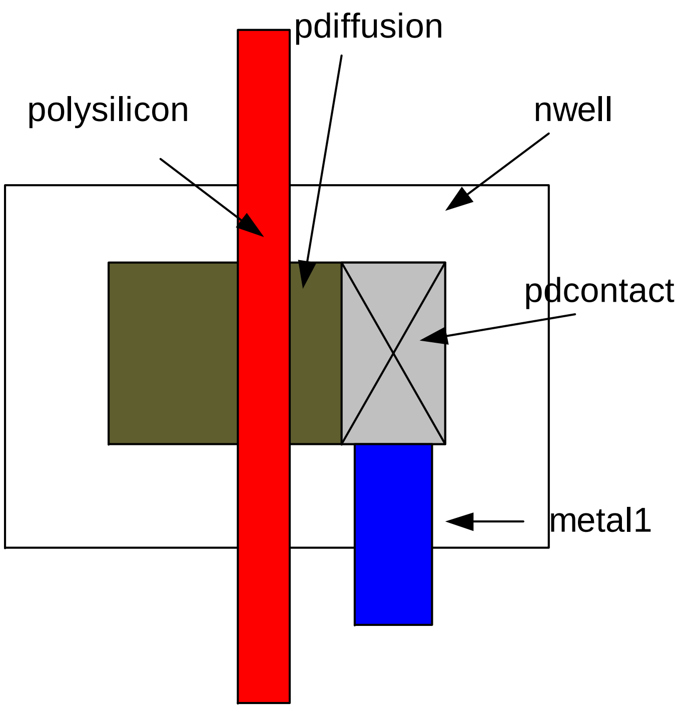
Laying out an NMOS



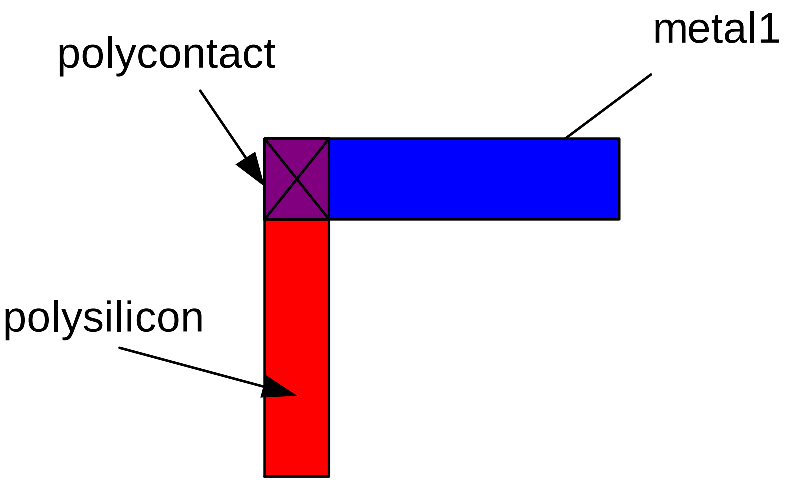


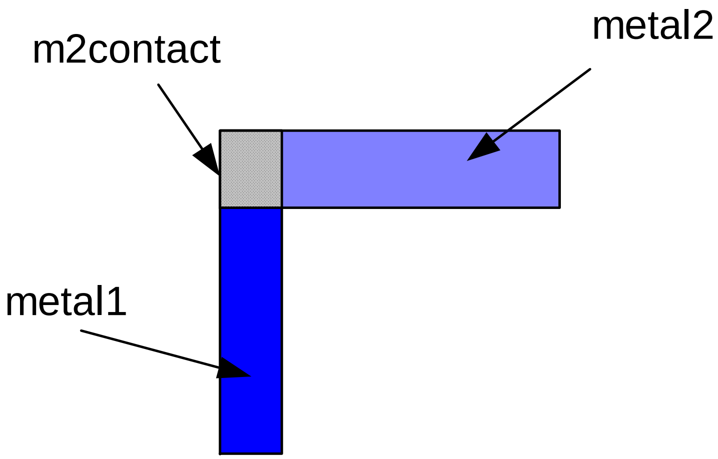
Laying out a PMOS





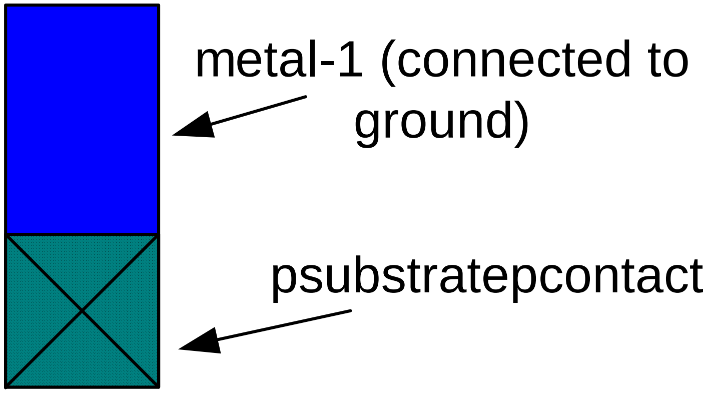
Connections and Vias





P-substrate contacts

Need to connect substrate to ground fairly often (at least once per cell). Do this with a psubstratepcontact

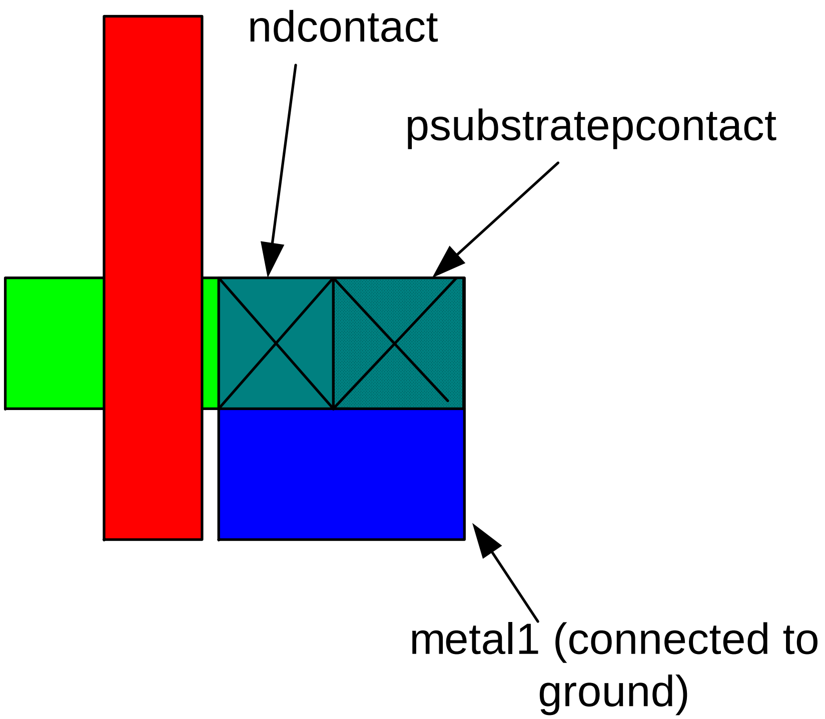


metal-1 (connected to ground)

psubstratepcontact

P-substrate Contacts

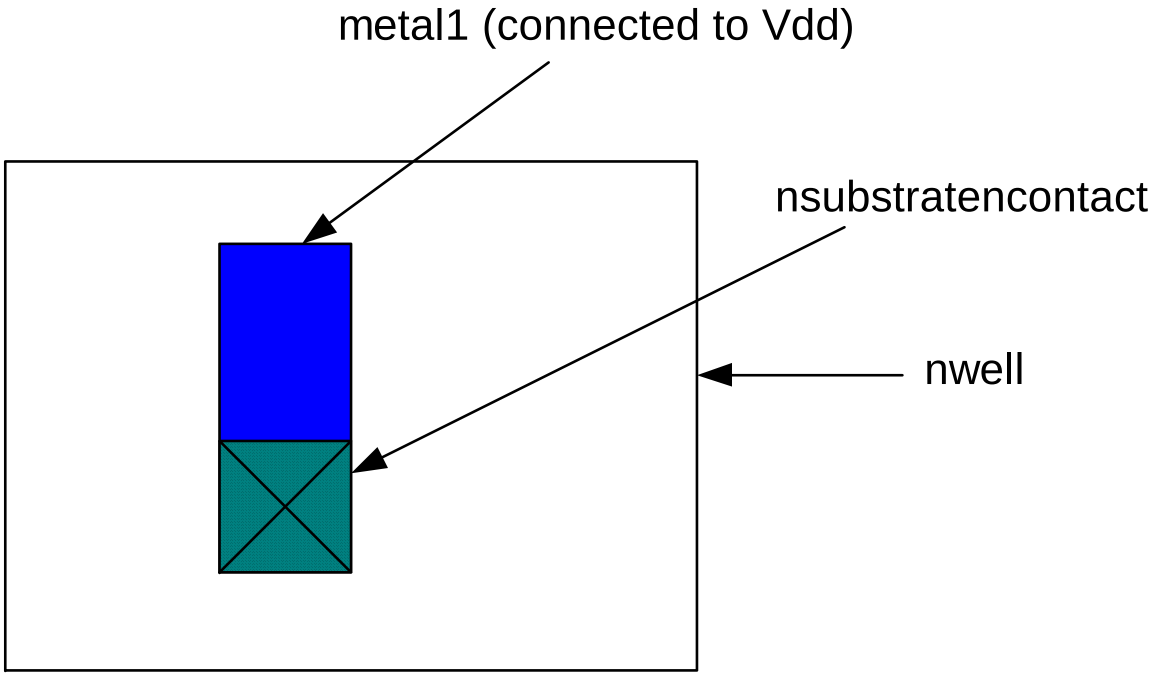
Sometimes, people combine their substrate contacts and NMOS transistors (likely, the source of many NMOS transistors are connected to ground).



Remember you only need to do this once per cell.

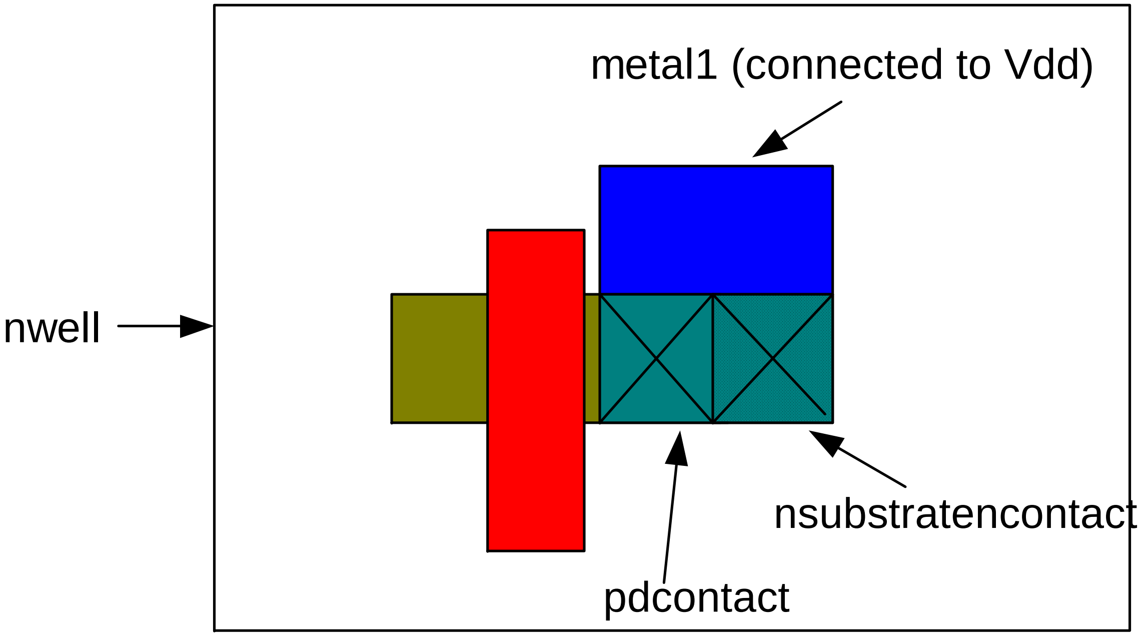
N-Well Contacts

Must also tie each Nwell to Vdd (about once per cell):



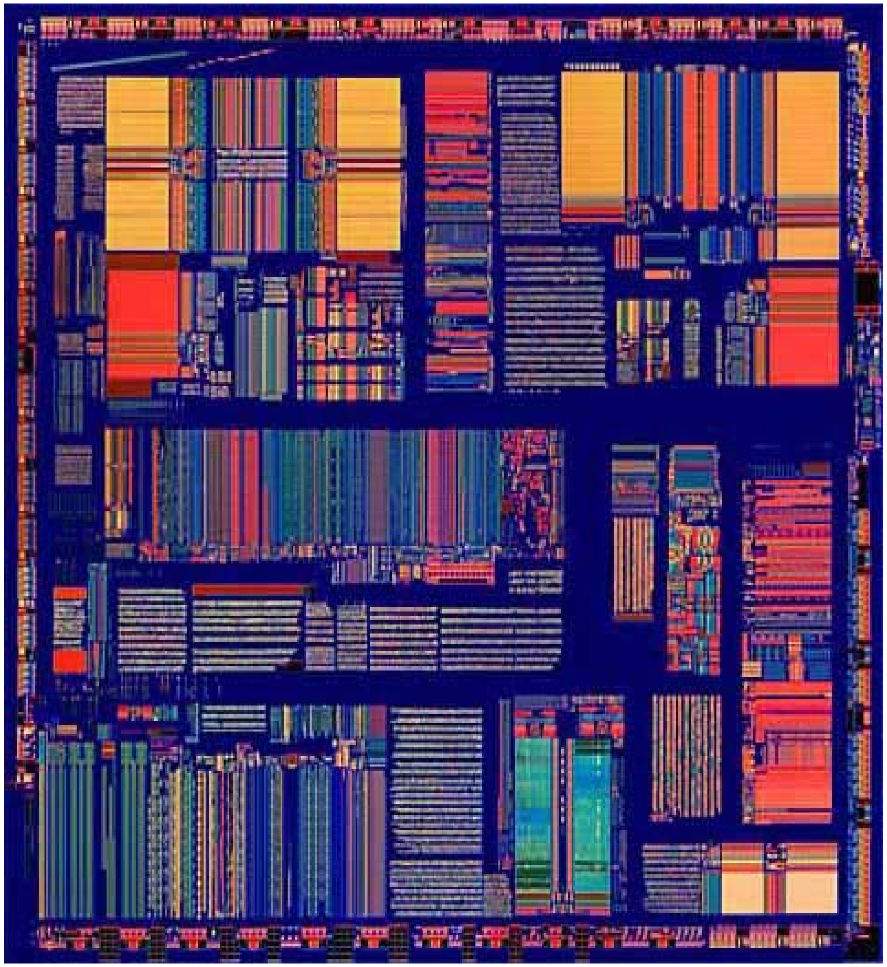
N-Well Contacts

Can also combine N-Well contacts with a PMOS transistor:



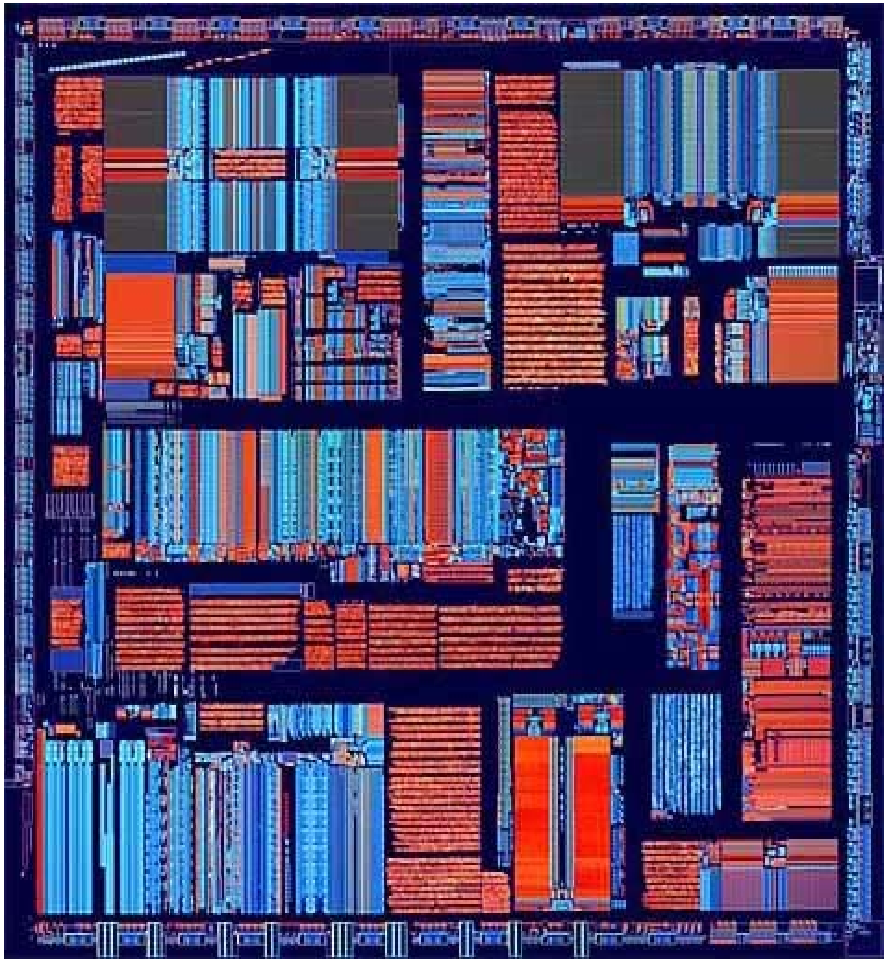
The best way to understand layout is to look at some existing layouts. In Assignment 2, you are given a cell called mystery.mag. Look at it very closely to understand how things are drawn and connected.

Pentium: First Poly Layer



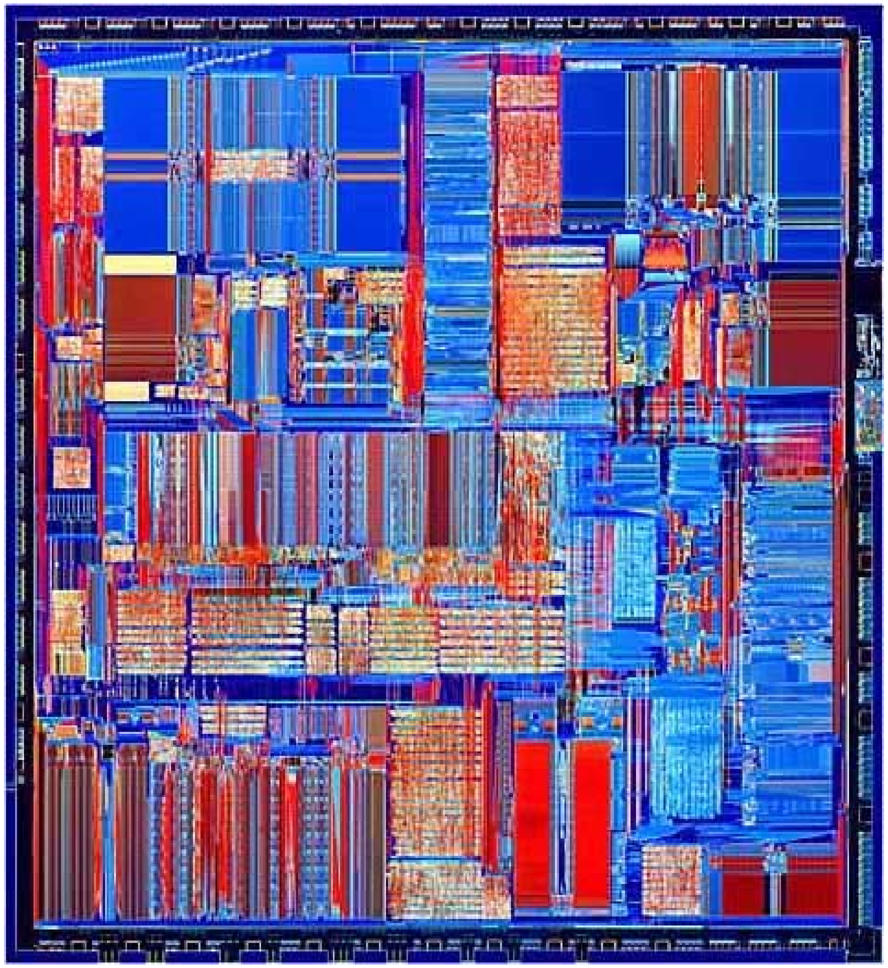
Source: M.W. Davidson, FSU © 1995-2001

Pentium: Second Poly Layer



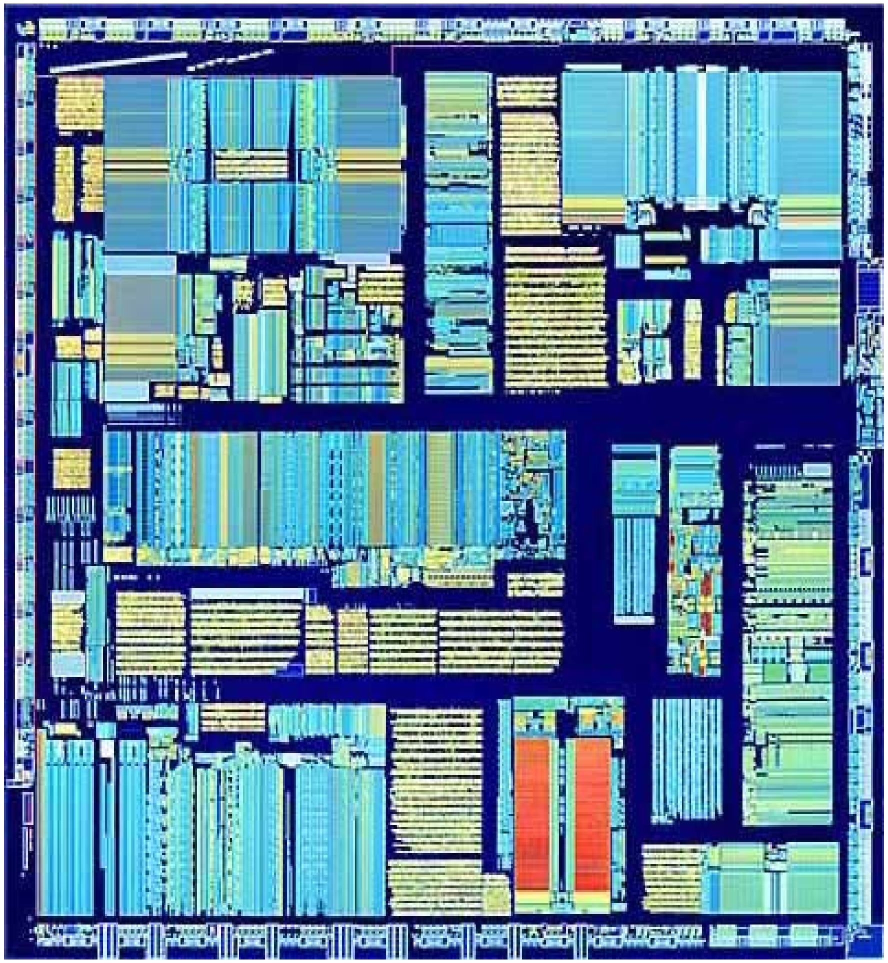
Source: M.W. Davidson, FSU © 1995-2001

Pentium: Metal Layer



Source: M.W. Davidson, FSU © 1995-2001

Pentium: Contact Layer



Source: M.W. Davidson, FSU © 1995-2001

Fabrication Constraints on Layout

Fabrication places many constraints on the layout

- In EECE479 we will worry about the two main types of constraints:
  - Resolution constraints (within one layer)
      - What is the smallest width feature than can be printed
      - What is the smallest spacing that will guarantee no shorts
    - Depends on lithography and processing steps that follow
    - Resolution often depends on the smoothness of the surface
      - Most modern processes are planarized, to keep surface flat
  - Alignment/overlap constraints (between two layers)
    - Like printing a color picture, need to align layers to each other
    - Mostly for contacts and poly overhang into diffusion

Design Rules

- Most processes have design rules that are expressed in absolute physical units
  - poly width 0.3μm
  - poly spacing 0.3μm
  - metal width 0.5μm
  - metal spacing 0.5μm
- Typically not multiples of one another
- Using process-specific design rules gives the densest layout but is difficult to master

- We will express our design rules in lambda (λ) units
  - λ is half the drawn gate length (poly width)
  - All other design rules are expressed in whole multiples of λ
    - poly width 2 λ, space 3 λ
    - metal width, space 3 λ
  - usually requires rounding up
  - rules are scaled to generate masks for a variety of processes

SCMOS Lambda (λ) Design Rules

We will be using the 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 0.030 μ in advanced fab lines

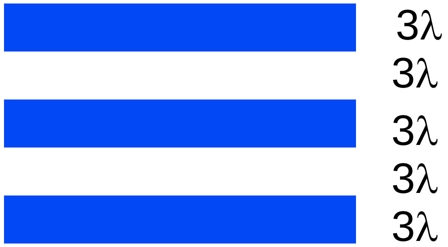
Ignores some of the ways to save area (so extra conservative)

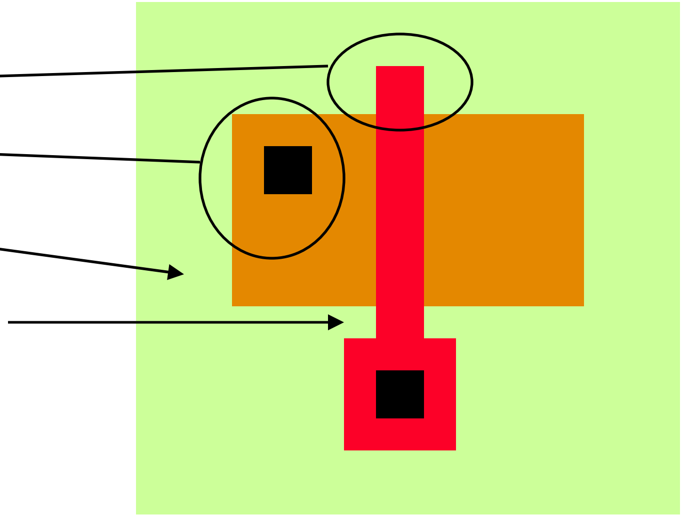
- Use only Manhattan Layouts
- Use only 90^o angles

Companies regularly do design scaling, even if they don't use the symbol λ

Geometric Design Rules

- Resolution
  - width and spacing of lines on one layer
- Alignment
  - to make sure interacting layers overlap (or don't)
- Examples
  - poly overlap of diff
  - contact surround
  - well surround of diff
  - contact spacing to unrelated geometry





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:

contact cut plus surround is 4λ

Using the Design Rules

While the Magic SCMOS design rules are simplified, there are still a number of rules to remember. A good way to start is to begin with a stick diagram of the cell you want to layout. Then you can use a subset of the rules to estimate what the layout will look like, and if it meets your standards you can begin the actual layout. While Magic makes layout easier, it always a good idea to have a plan on where things go before you start.

Warning:

While layout is often (sometimes) fun to do, it easily can be become an infinite time sink – one can always find a way to shrink the cell a few more microns. You should really have a plan BEFORE you start layout, and have a set constraints you are trying to achieve so you know when you are done.

Layout Guidelines

In CMOS there are two types of diffusion

- ndiff (green)
  - Poly crossing ndiff makes nMOS transistors
- pdiff (yellow / brown)
  - Poly crossing pdiff makes pMOS transistors

Be careful, ndiff and pdiff are different

- You can't directly connect ndiff to pdiff
  - Must connect ndiff to metal and then metal to pdiff
- Can't get ndiff too close to pdiff because of wells
  - Large spacing rule between ndiff and pdiff
  - Means you need to group nMOS devices together and pMOS devices together
- Can't connect poly directly to M2 -- must use M1 to reach M2

Basic Layout Planning

Here are a few more simple guidelines to CMOS layouts:

- You need to route power and ground (using metal)
  - often power and ground is run horizontally
- Try to keep nMOS devices near nMOS devices and pMOS devices near pMOS devices.
  - So nMOS usually are placed near Gnd, and pMOS near Vdd
- Run poly vertically and diffusion horizontally, with metal1 horizontal (or the reverse, just keep them orthogonal)

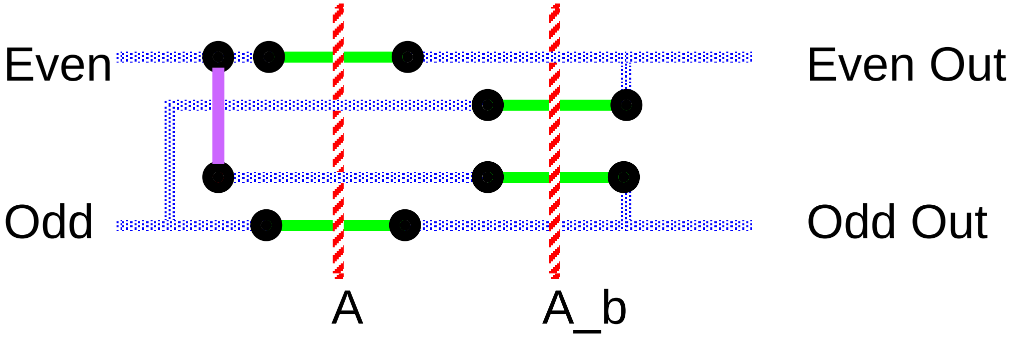
Basic Layout Planning

Here are a few more simple guidelines to CMOS layouts:

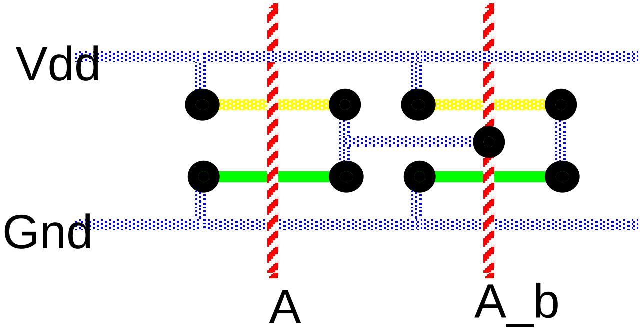
- Keep diffusion wires as short as possible (just connect to transistor)
- All long wires (wire that go outside a cell, for example) should be in either m1 or m2.
- Try to design/layout as little stuff as possible (use repetition/tools)

Typical Cell Layout Plan

Parity Circuit (from earlier).



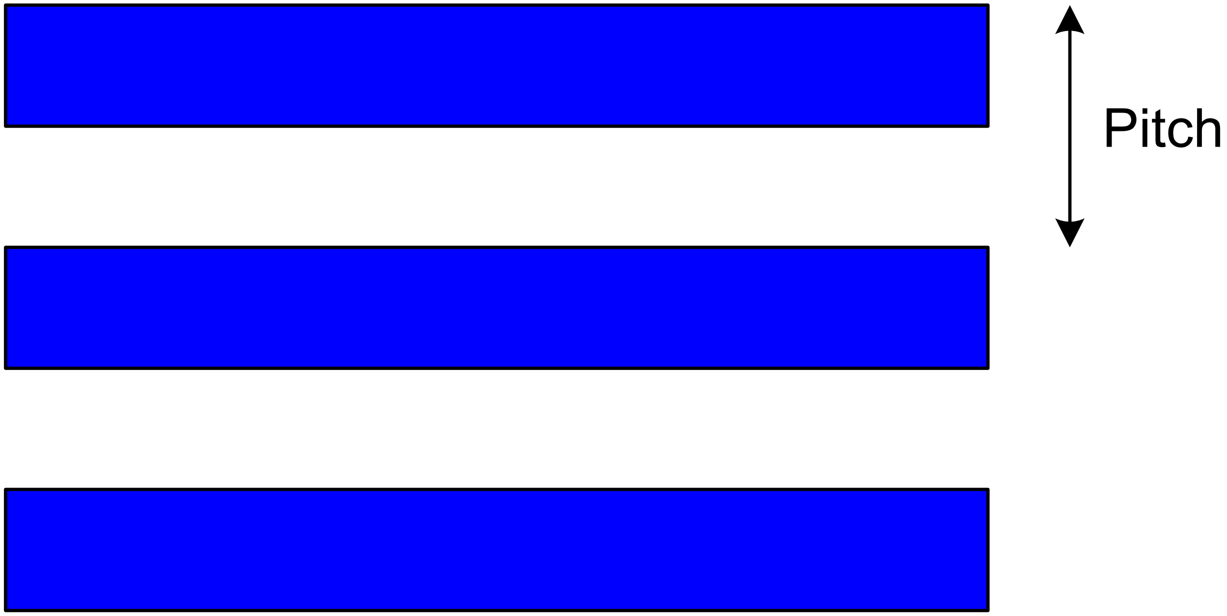
CMOS Inverter (this is actually two in series)



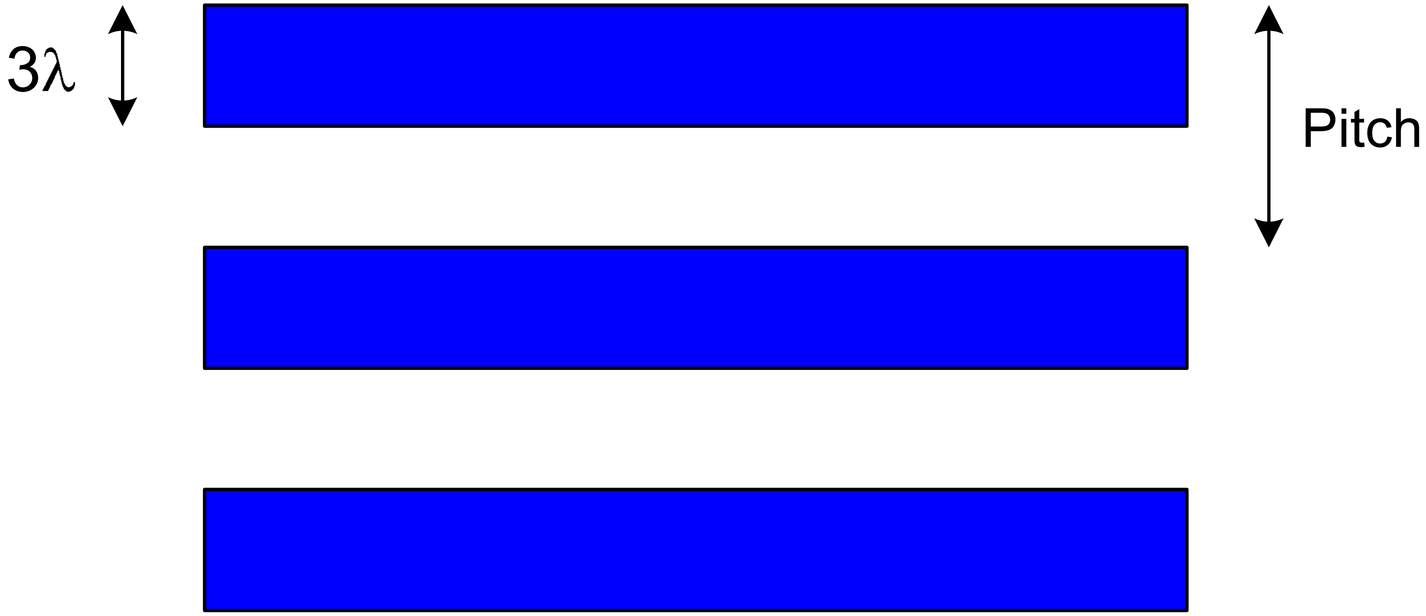
What I will show you next are some rules of thumb for estimating the size of a layout from a sticks diagram

You might want to do this if you are “floorplanning” your project (deciding how big each block will be and how they will fit together) before assigning each block to a group member

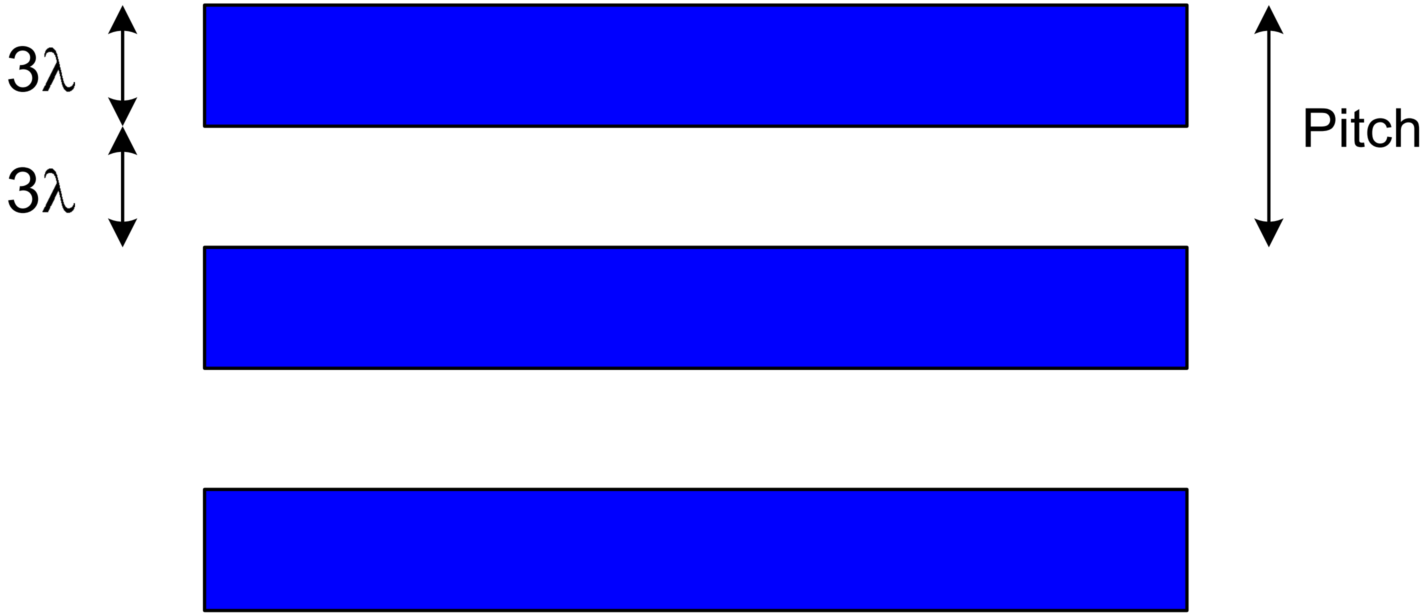
Definition: Pitch



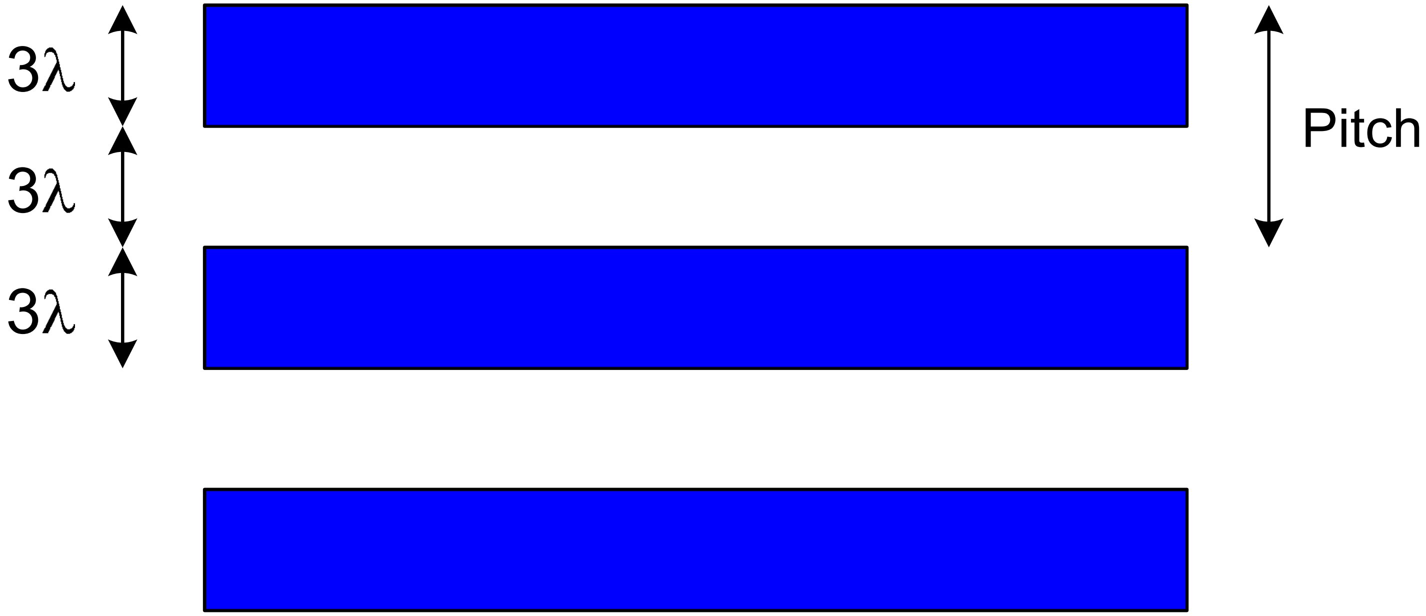
Definition: Pitch



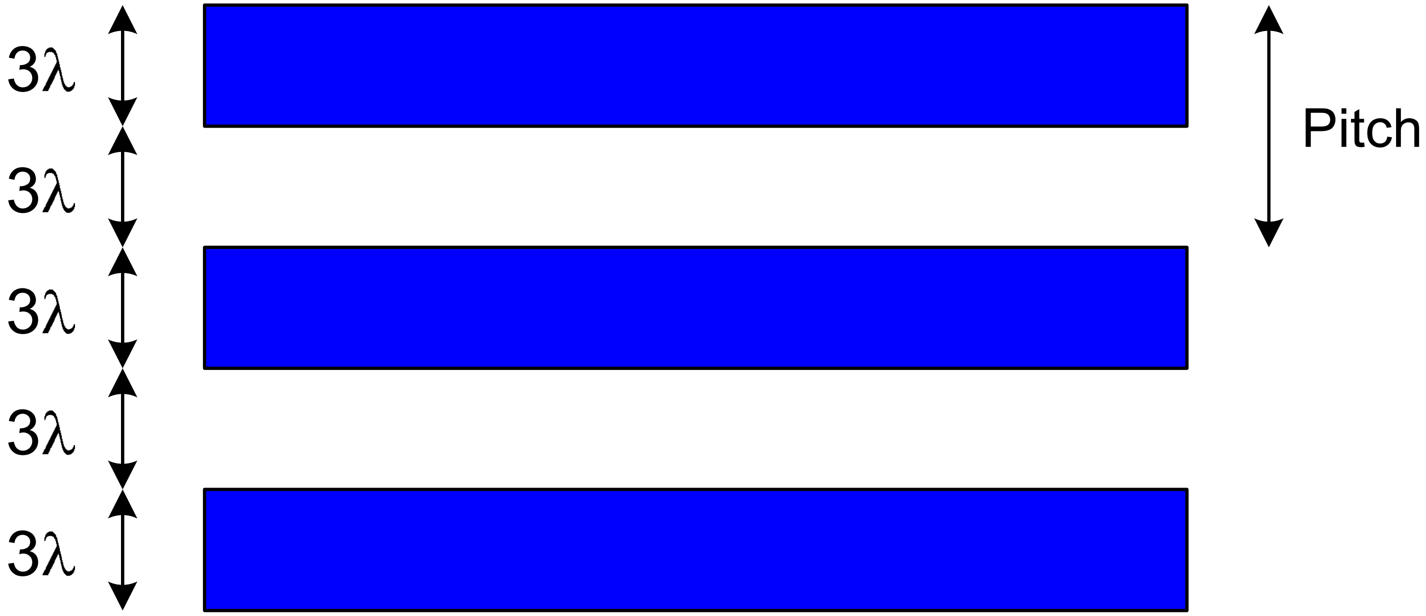
Definition: Pitch



Definition: Pitch



Definition: Pitch



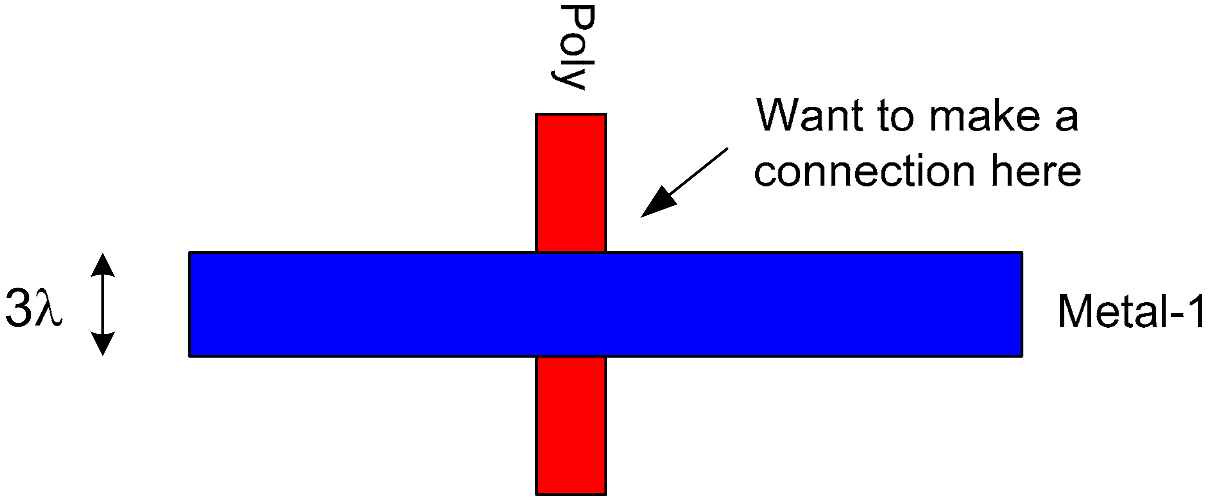
In this case, the “pitch” is equal to 6λ

If we have *n* metal lines, their height is approximately (*n**pitch)

Minor Complication

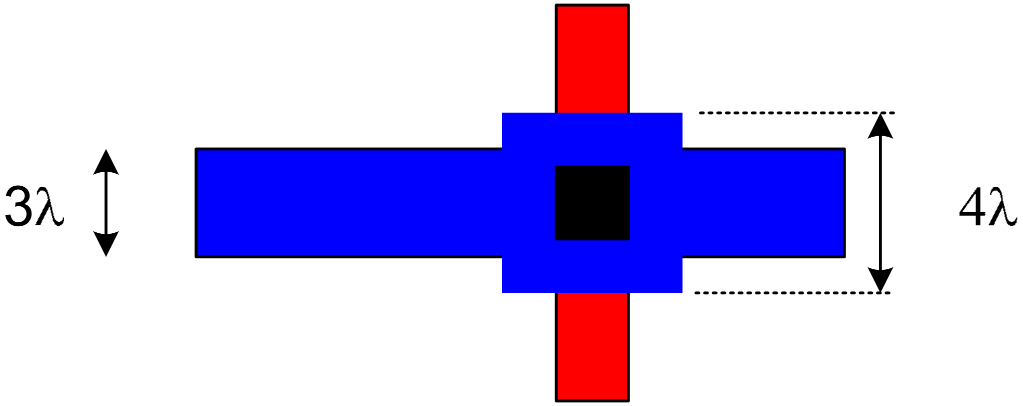
We rarely can put metal lines so close together.

The reason is: we often need to connect the metal lines to something (eg. poly) at some point.

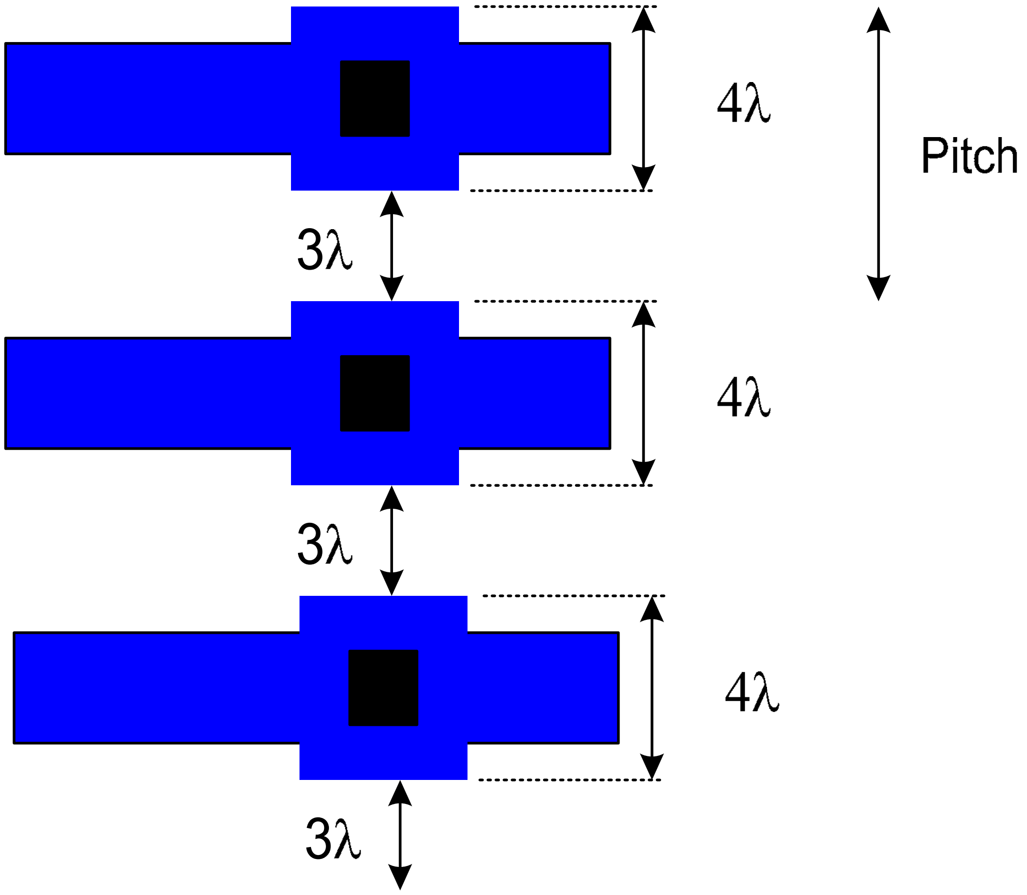


Minor Complication (continued…)

But, from our design rules, “contact cut plus surround” = 4 λ



Minor Complication (continued…)

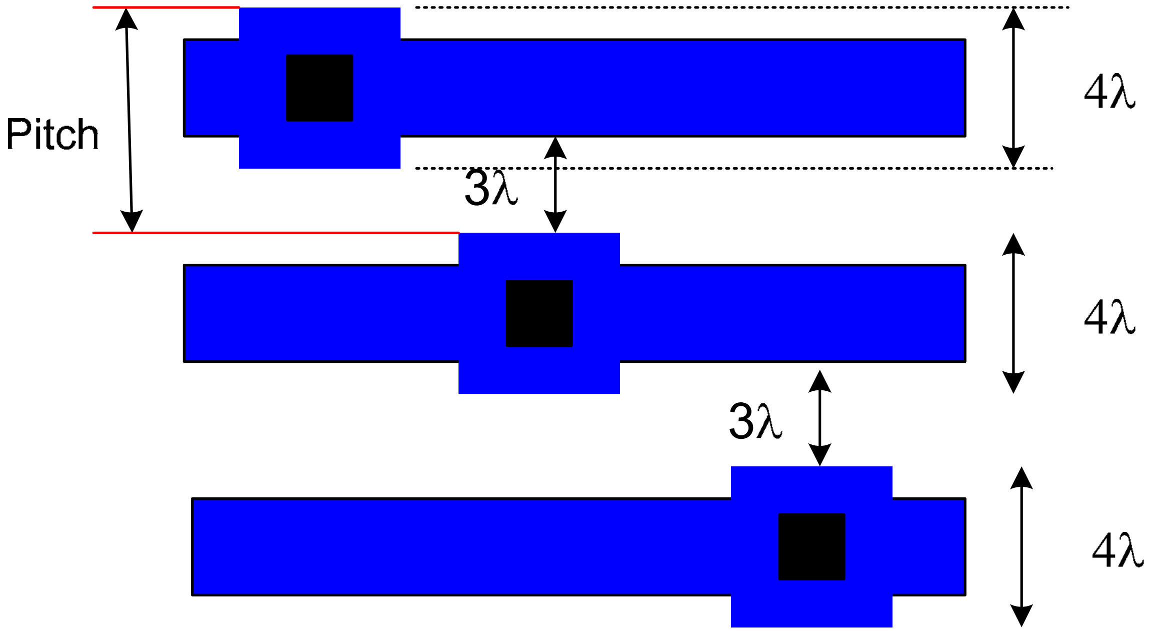


Worst case pitch is now 7 λ

This is called "fully contacted" pitch

Minor Complication (continued…)

Of course, a smart engineer (e.g. you) might choose to “stagger” the connections to compact things a bit more



Worst case pitch is now 6.5 λ

(can you see why this is?)

This is called “semi-contacted” pitch

Summary of metal pitch:

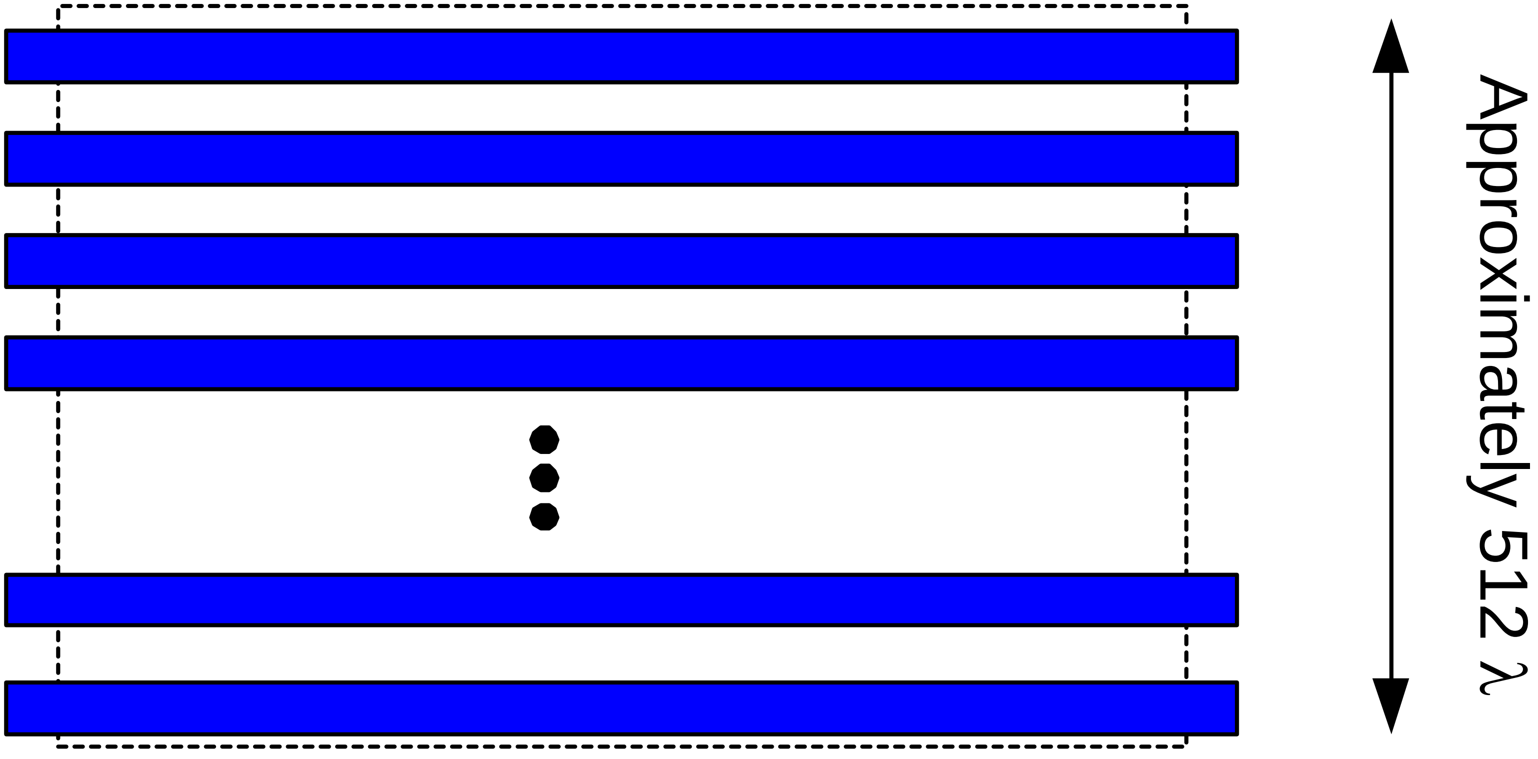
	Metal-1
No contacts:	6λ
Fully Contacted Pitch:	7λ
Semi-Contacted Pitch:	6.5λ

Summary of metal pitch:

	Metal-1	Metal-2
No contacts:	6λ	7λ
Fully Contacted Pitch:	7λ	8λ
Semi-Contacted Pitch:	6.5λ	7.5λ

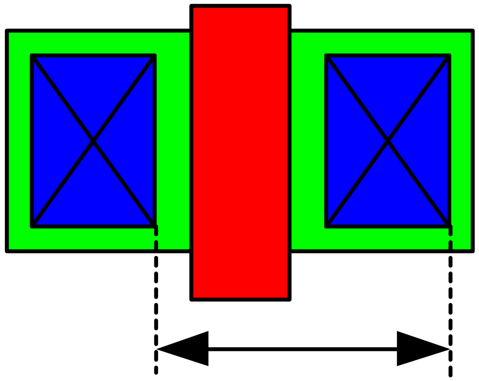
When we are estimating the size of a cell, let's use 8λ as a quick estimate of pitch

Example: if we have a cell that has 64 wires in parallel, the height of this cell would be approximately 8 λ * 64 = 512 λ



What about Transistors?

These also approximately fit on this 8 λ grid:

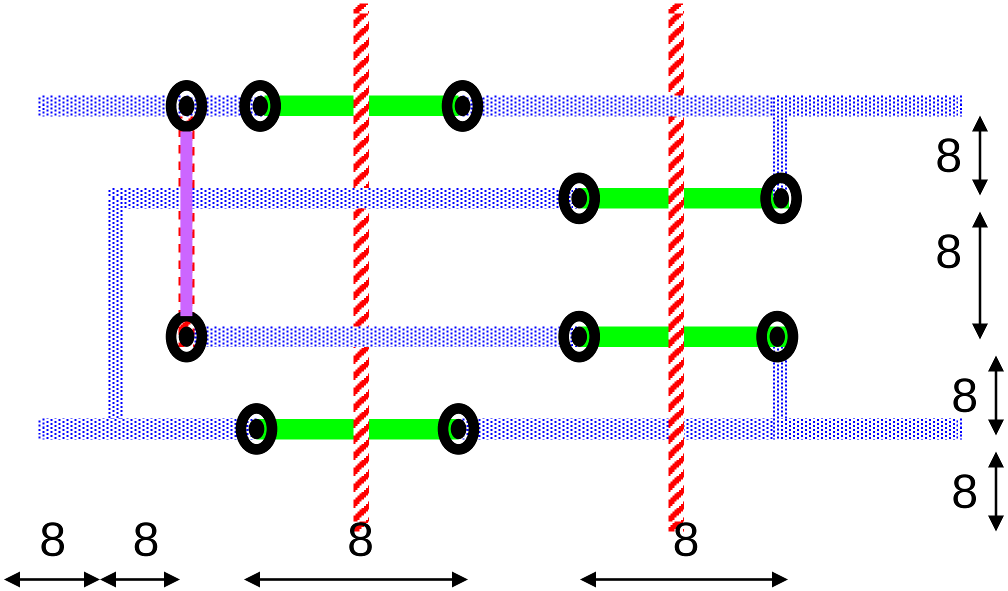


Approximately 8λ

The “Rule of 8’s”: When estimating size, everything is 8 λ

This obviously is not exactly correct, but it will give you a rough idea how big your cell is very quickly.

Example of Size Estimation: Parity Circuit



8 * (3 tall +1 for space to next cell) = 32λ tall

8 * (3 wide +1 for space to next cell) = 32λ wide

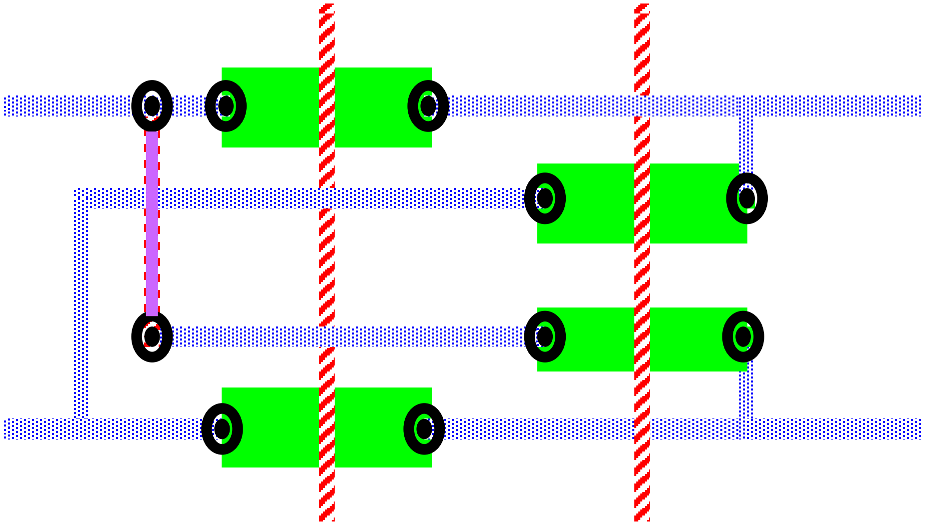
One more wrinkle…

This approximation assumes that all wires and transistors are of the minimum size.

For wires, this is usually the case (except for power/ground lines… but we'll talk about that later)

As we will see, however, transistors are often considerably wider than their minimum…. (example on next slide)

Example of Size Estimation: Parity Circuit



If we know the transistors are wider, we can add a little bit to the height of the cell. How much do we add?

-> Add *x*, where *x* is the difference between the actual width and minimum width

Size Estimation Summary

Don’t get too stressed out about getting your size estimations correct.

This is just a tool that might help you when are doing your initial layout planning.

As you get more experience, you will be able to come up with more accurate estimates faster. It just takes practice.