

# EE-429

# Fundamentals of VLSI Design

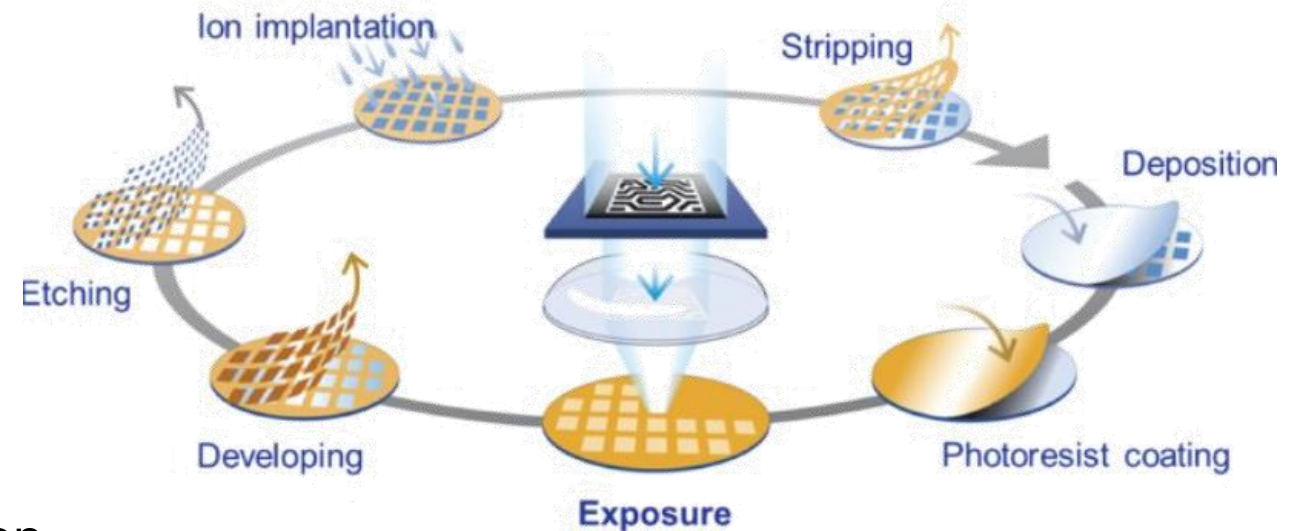
## Layout

Andreas Burg

# Manufacturing Process

- **CMOS is based on a photolithographic process and a sequence of**

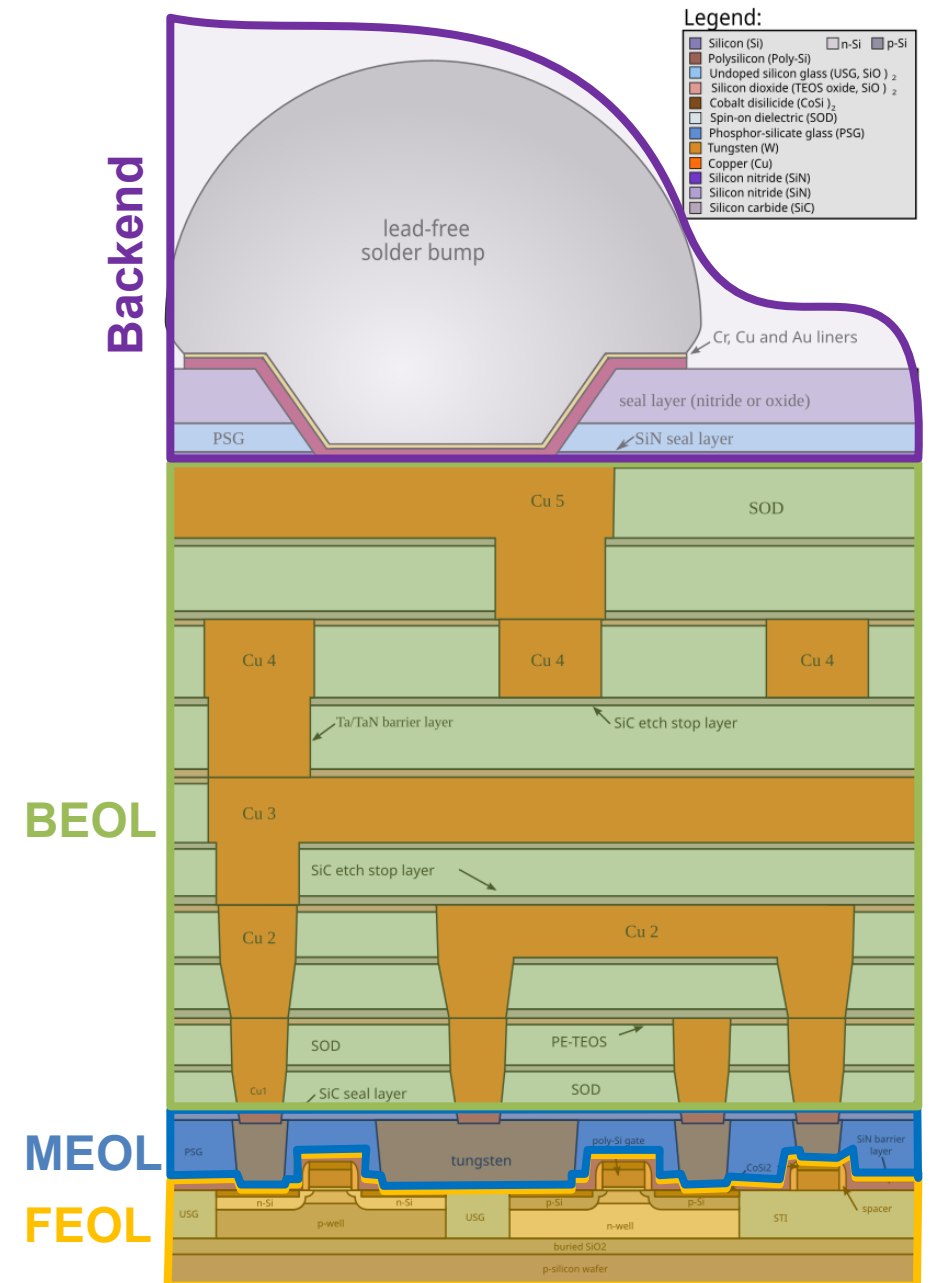
- Deposition or growth of some material
- Coating with a photoresist
- Exposure of the photoresist & developing
- Edging of the unprotected areas
- (Ion Implantation)
- Stripping of the photomask or mask layers
- There are many other small steps in between, adding up to >150 major steps for a modern process



- **All relevant information is contained in the lithography masks**

# CMOS Layer Stack

- **Manufacturing process results in layer stack**
  - A modern FinFET process has 30-50 masks
- **We distinguish between**
  - **Front end of line (FEOL):** mostly structures inside the wafer, comprising all active components
  - **Mid end of line (MEOL):** connections to the FEOL components (passive)
  - **Back end of line (BEOL):** main interconnect layers, starting from M1 or M2 and up
  - **Backend:** routing and contact layers for packaging
- **Complexity and density decreases from lower layers to higher layers**



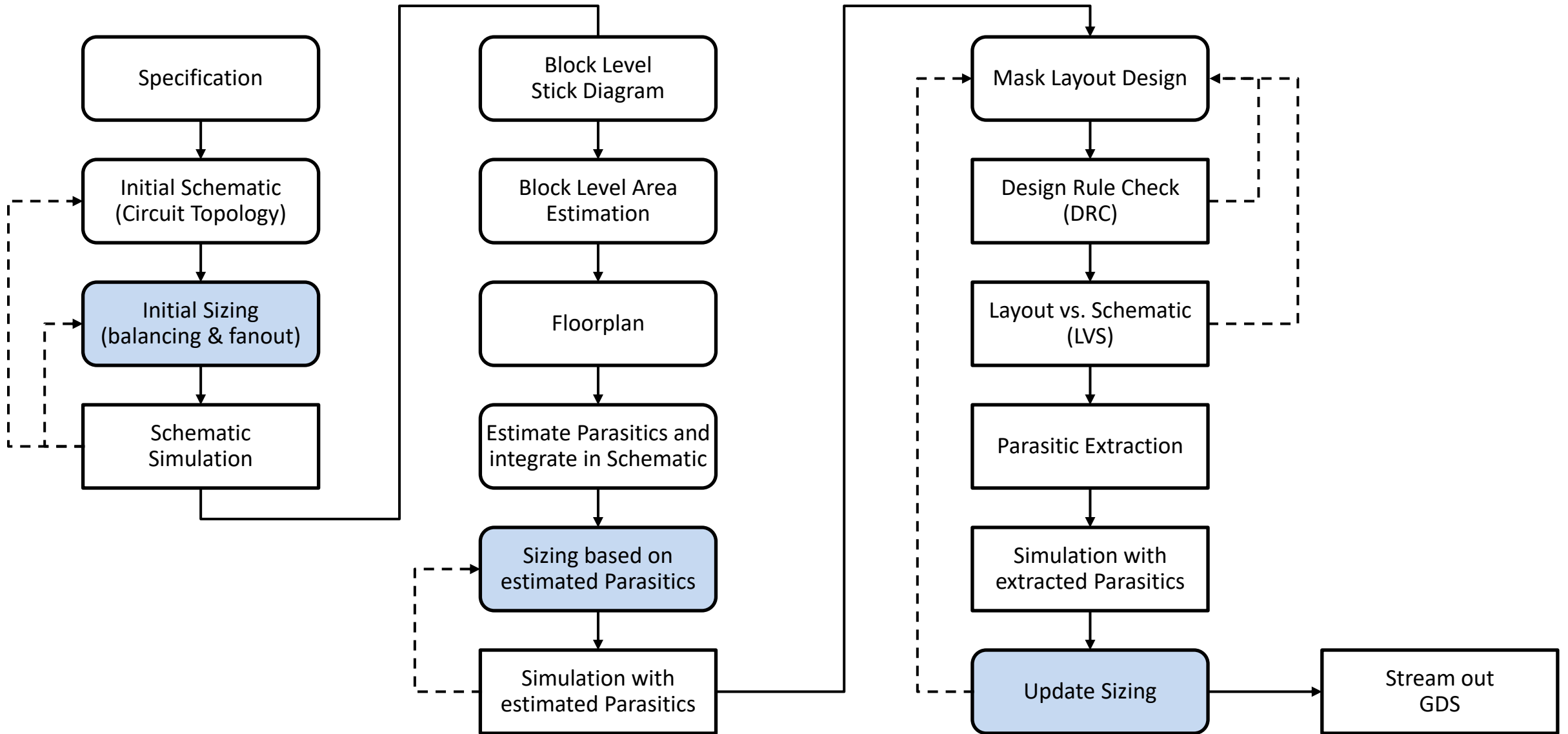
# Front-End and Back-End of Line (FEOL / BEOL)

- **Front end of line (FEOL): mostly structures inside or just on top of the wafer**
  - Forms the active devices based on doped silicon: transistors, diodes, MOS capacitors
  - Structures: wells, diffusion, gate oxide, gate structures, trench isolation, LOCOS
  - Process steps: oxidization, gate formation, ion implantation, annealing
  - Materials: Silicon, doping materials, silicon dioxide, high-k dielectrics, gate: polysilicon & metals
- **Mid end of line (MEOL): between FEOL active components and BEOL**
  - Forms substrate-, diffusion-, and gate-contacts and polysilicon routing, M0 for FinFETs, M1
  - Process steps: contact creation, silicidation (of polysilicon reduces resistance), barrier and liner deposition (protection of active silicon areas against diffusion)
  - Materials: Silicides, barrier materials, low-k dielectrics
- **Back end of line (BEOL): device, sub-circuit, and chip level interconnect**
  - Structures: Metal wires and vias, passive components (e.g., metal capacitors, inductors, metal resistors)
  - Process steps: metal and dielectric deposition, patterning, chemical mechanical planarization (CMP)
  - Materials: Copper, aluminium, low-k dielectrics, liners and barrier materials

# From Schematic to Layout

- **Physical design (layout) introduces parasitics that significantly change circuit behaviour compared to schematic only simulations**
  - Drive strength of circuit elements (and potentially even the topology) need to be adapted to properly deal with (drive) parasitic elements
- **Layout is tedious and time consuming and even small schematic updates require often significant layout updates**
  - Limited physical space: updating even a single transistor impacts many others
  - The tighter the layout the greater the effort for even very small updates
- **Plan ahead to avoid or minimize updates to the schematic after layout**
  - Anticipate layout parasitics early to avoid post-initial-layout surprises/changes
  - Build your initial layout to leave “room” for potential updates (you can always compact later)

# Full Custom Design from Schematic to Layout



# Basic Rules of Engineering

- **Engineers are lazy**  
If something takes too long, you are doing something wrong.
- **Divide and Conquer**  
Divide complex problems into a collection of smaller simpler problems, solve one by one.
- **Maximize Reuse**  
Try to avoid re-inventing the wheel when you can re-use existing pieces, even if the solution is slightly suboptimal
- **Simple and Regular**  
Try simple and regular structures, they are easier to design, debug, and to reuse / combine
- **Engineering is not a religion**  
Find the solution that best fits your problem

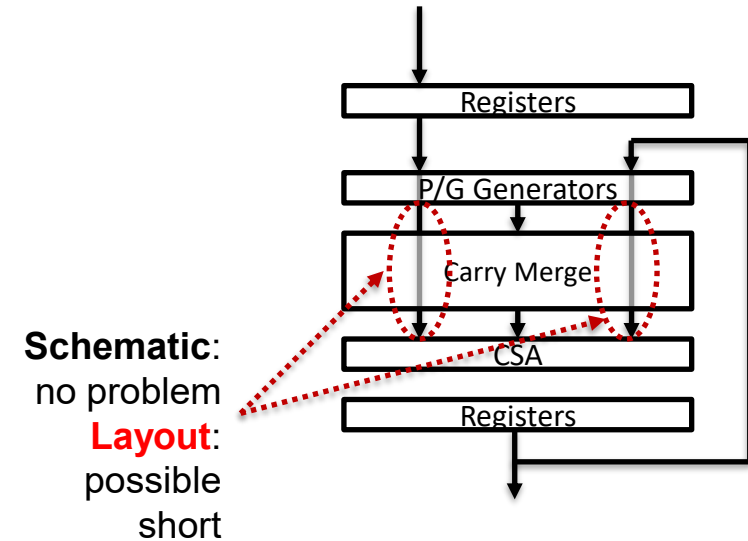
} Hierarchical Design !!!

# Before Starting a Layout

- **Plan ahead:** Full custom design is difficult. You should always plan your circuit with pen and paper before you start drawing
- **Connections determine your layout:** The external signal connections to your block influence your block greatly.
  - Before starting layout always think about your connections
  - What are the neighboring cells, how will you connect them
  - Assign signal directions
  - Assign layers to your signals
- **Each technology is different:** In each technology there are a couple of design rules that limit the performance. They are not always the same.

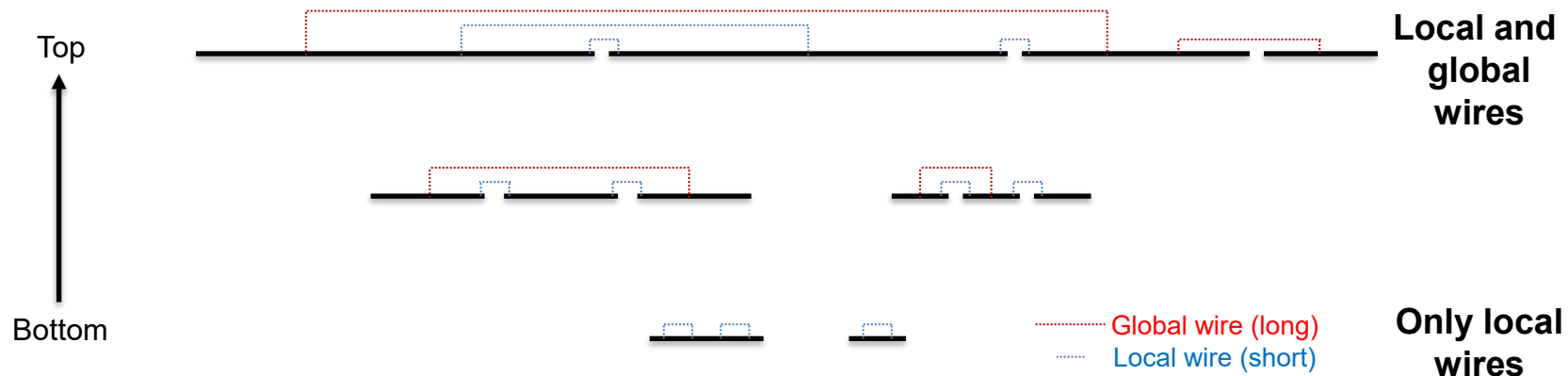
# Difference of Hierarchical Schematic and Layout

- **Schematic:** Levels are completely decoupled, i.e., occupy different resources for wiring and components (hierarchy= 3rd dimension)
- **Layout:** Different levels of hierarchy use the same physical resources (2D components and routing layers)
- **Impact on hierarchical layout:**
  - Bottom-up: always respect the layout of lower levels of hierarchy
  - Top-down: leave sufficient resources for lower levels of hierarchy during the design
  - Top-down planning with bottom-up design:
    - Top-level: floor-planning anticipates top-level design (floorplan) to estimate required resources;
    - Lower-levels: (bottom-up) are designed to leave sufficient resources for top-level structures



# Hierarchical Routing

- Higher levels of hierarchy are comprised of multiple blocks from lower levels of hierarchy
- Area of blocks grows as we proceed from bottom to top



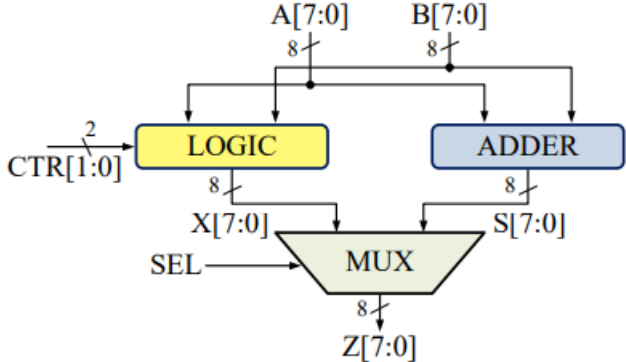
- Lower hierarchy levels: only short local wires
- Number of (long) global wires increases when moving up the hierarchy

# Floorplanning

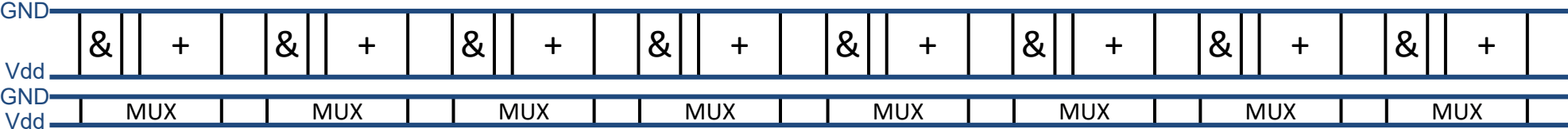
- **Floorplanning defines the layout plan for a hierarchical design**
  - Define the relative position of the individual instantiated building blocks, while taking into account
    - anticipated size
    - interconnect wire length
    - routing congestion
  - Reserve space for global wires that connect all blocks such as
    - Supply: VDD, GND
    - Critical signals: clocks
    - Long large busses
  - Pre-route critical and long global nets or reserve space for them
- **Floorplan helps to fix the outline (aspect ratio) of lower level hierarchical blocks that do not have a fixed layout yet to fit well into their future location in the floorplan**

# Floorplan Example

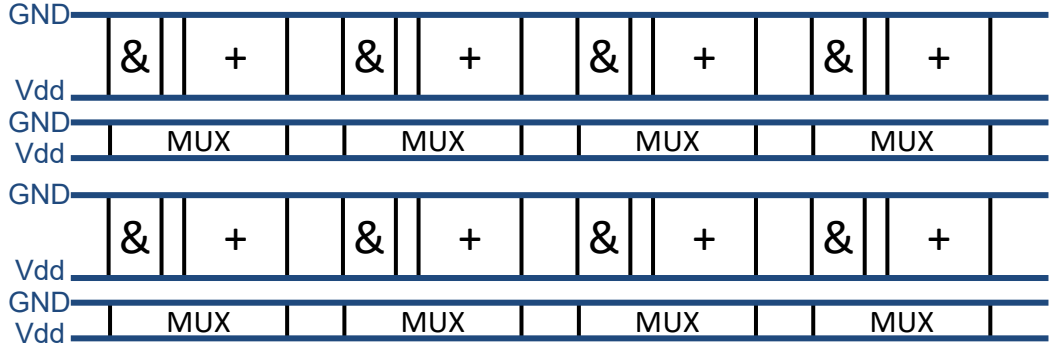
- Consider the example of an 8-bit ALU



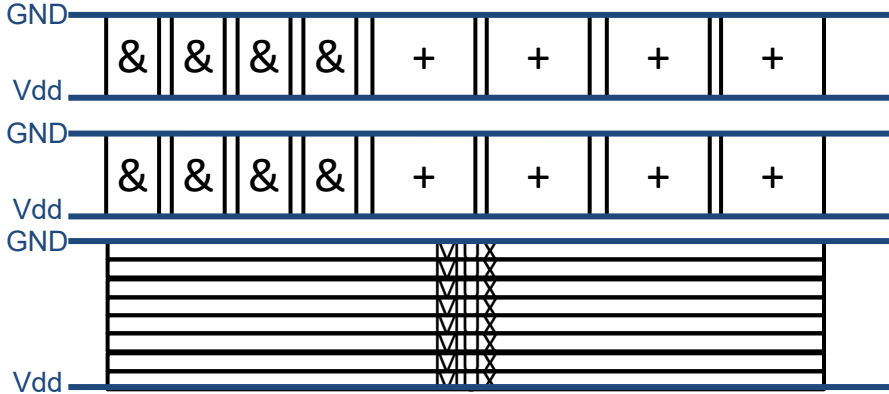
- Floorplan option 1:



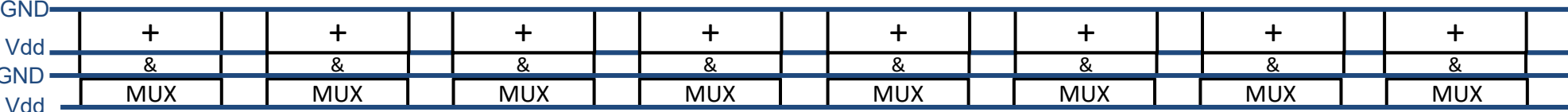
- Floorplan option 2:



### Floorplan option 3:

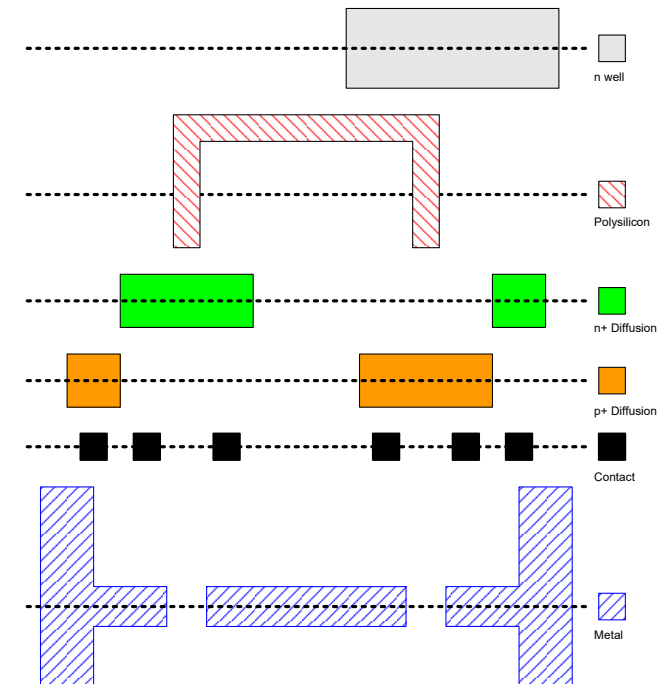
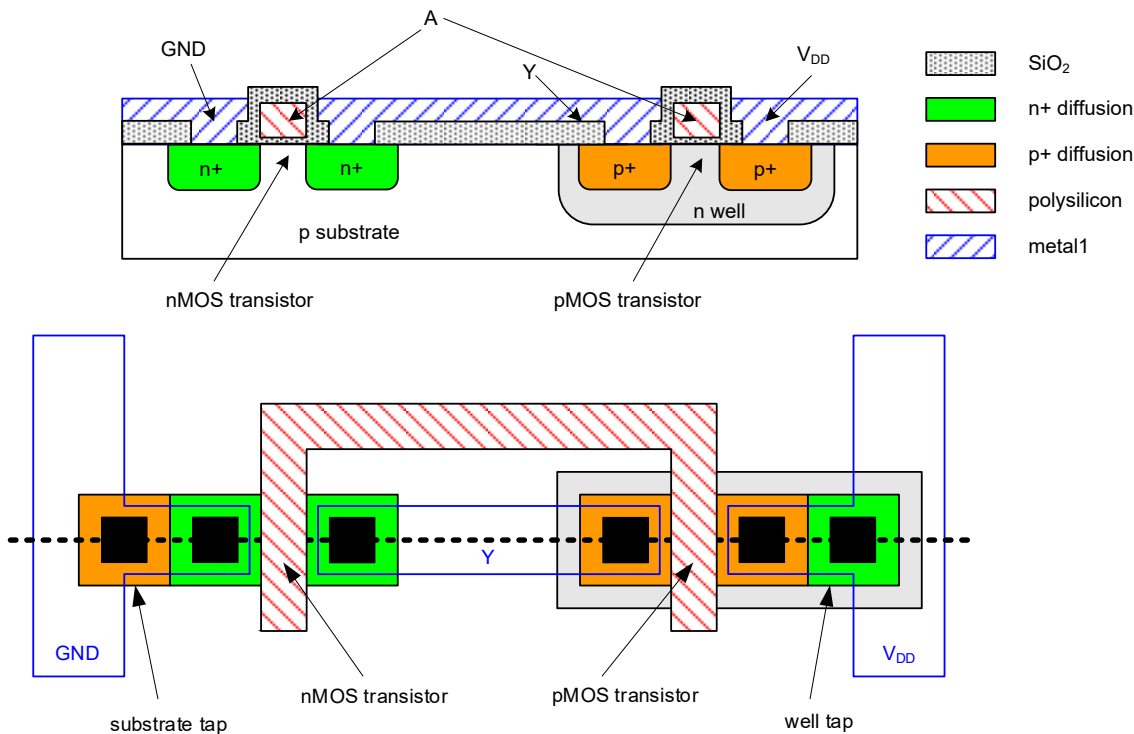


- Floorplan option 4:



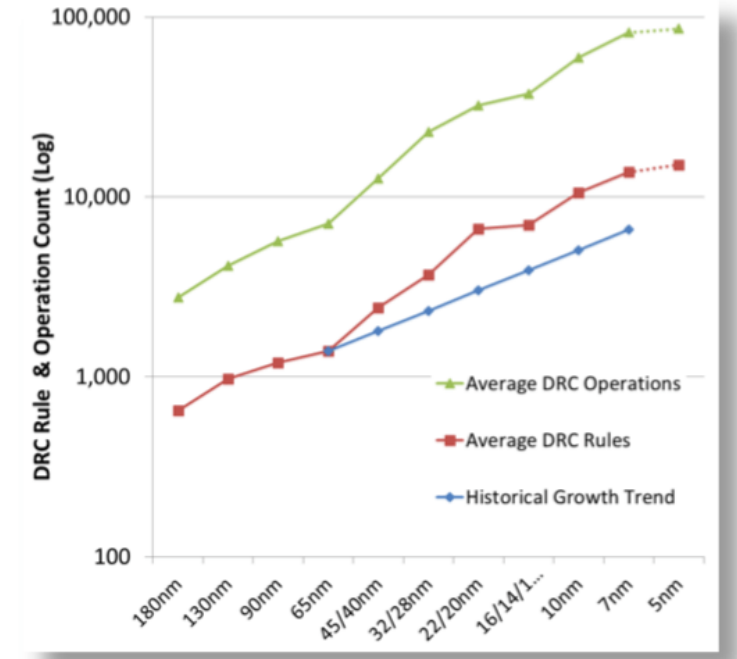
# Detailed Layout

- The layout defines the masks (directly or indirectly) and thereby defines
  - The shapes of **physical structures** created by depositing materials
  - **Regions with specific electrical properties** created by ion implantation
  - **Devices** formed by a combination of different shapes and regions



# Design Rules

- **Interface between designed and process engineers**
  - Simplified way to define what is allowed to ensure reliable manufacturing
  - Account for uncertainties in manufacturing and due to proximity effects with margins
  - Goal is to **ensure** consistent **high yield** and to **ensure that models match the silicon**
- **Specific to each process and to each manufacturer**
- **Modern technologies often have hundreds of design rules**
  - Complexity has grown dramatically
  - Rules are getting more restrictive:  
e.g., uni-directional routing, fixed transistor orientation, ...
- **Pushed rules: deviations from standard design rules**
  - Typically adopted for SRAM bitcells (1.5x area reduction)
  - Require “qualification” which is expensive and time consuming
  - Today, very hard to obtain access even for very large customers

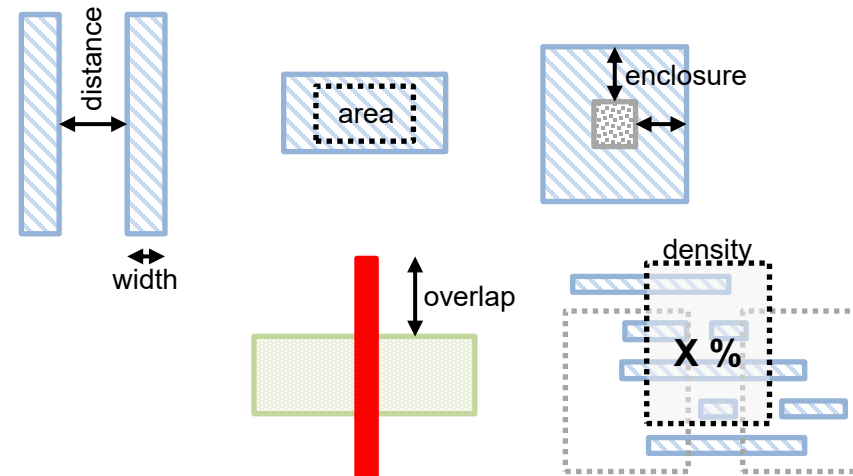


# Process Limitations

- **Reliable manufacturing imposes layout limitations**
  - Captures by the design rules described in the design rule manual (DRM) of the PDK
- **Design rules impose restrictions on structures on either the same layer (FEOL and BEOL), between FEOL layers, or between adjacent layers (BEOL)**

- **Typical limitations and corresponding design rules**

- Minimum distance/spacing
- Minimum width
- Minimum area
- Minimum enclosure
- Minimum overlap
- Minimum density



# Lambda and $\mu\text{m}$ (Micron) Design Rules

- **Early days of VLSI design: design rules defined by distances as multiples of a constant lambda ( $\lambda$ )**
  - Introduced by Carver Mead and Lynn Conway in the late 1970s
  - **Main idea: enable scalability** of layouts with technology scaling (no or little modifications)
  - **Good for educational purposes** and illustration
  - Too coarse, leading to **significant overhead**
  - **Not sufficiently generic** anyway to enable layout transfer between technology nodes
- **Modern design rules are always specified in  $\mu\text{m}$  (in the PDK)**
  - Technology dependent, but **allows to be precise, enabling dense layouts**
- **New approximate means to express dimensions based on pitches**
  - Contacted Poly Pitch (CPP) and metal pitch

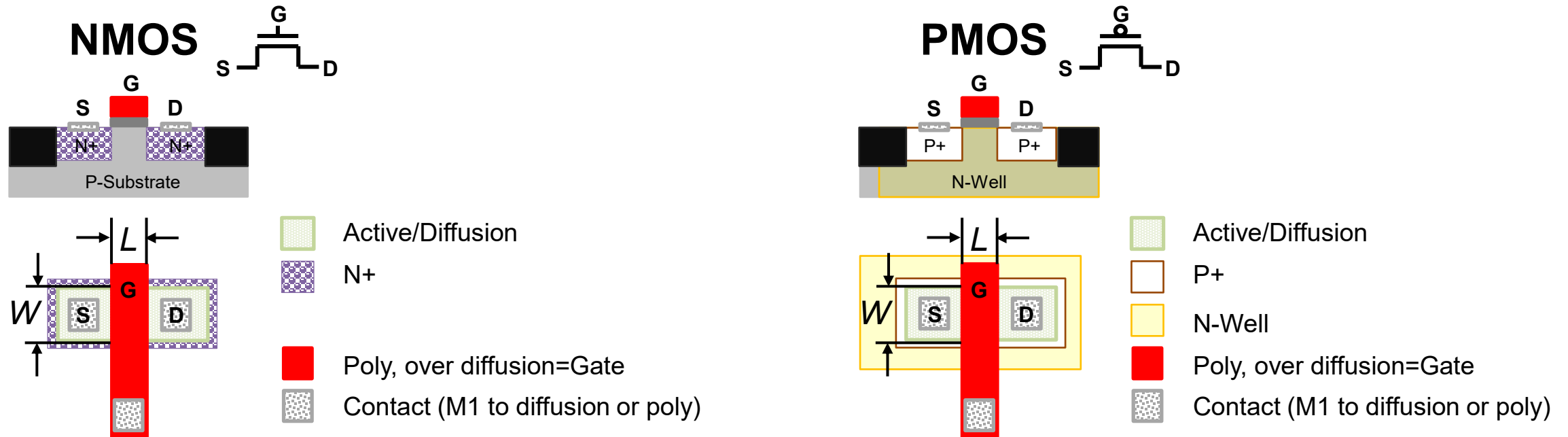
# Layers

- **Physical (manufacturing layers): translate into manufacturing masks**
  - Examples:
    - FEOL layers: Substrate, N-Well, P-Well, Deep N-Well, Diffusion/Active, N+, P+, VT selection, ...
    - MEOL: Contacts
    - BEOL layers: Metal (M1, M2, Mx),
  - Design rules described in the Design Rule Manual (DRM)
  - Included in the final tapeout GDS
- **Non-physical (CAD or annotation) layers: used exclusively for CAD tools**
  - Means to include additional (non-physical) information in the layout
  - Examples:
    - Pin layers: describe connection points and names for hierarchical design
    - Text layers (for each layer): label a physical net with a name
    - P&R boundary: indicate the outline/boundary of a hierarchical cell
  - Not included in the final tapeout GDS

# Basic Layout Structures: Transistors

- **NMOS & PMOS for a single-well process:**

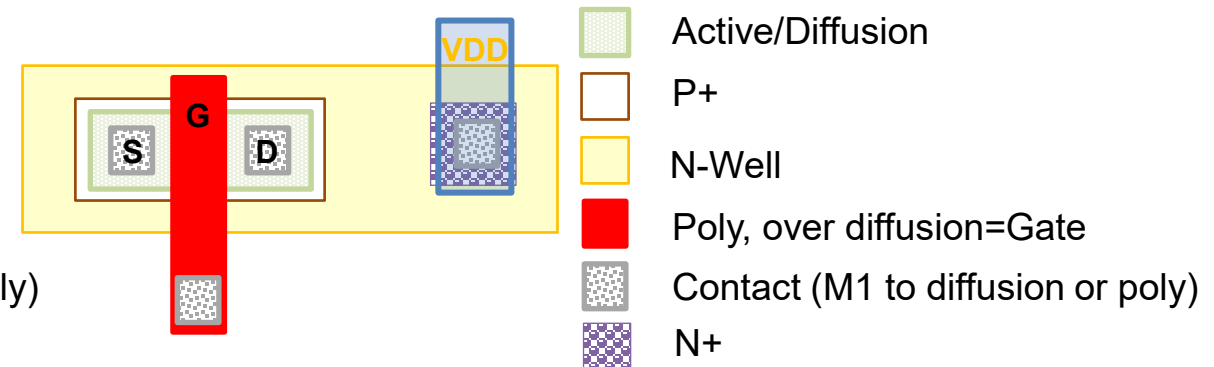
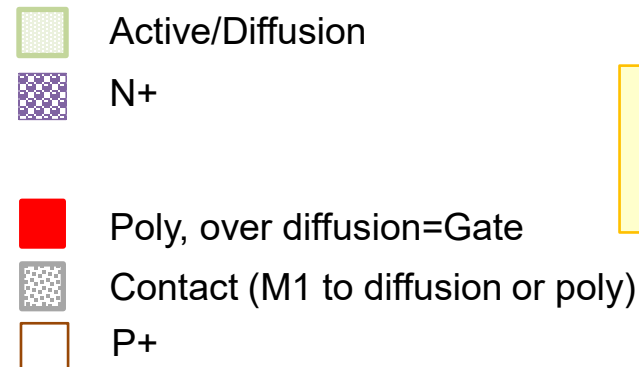
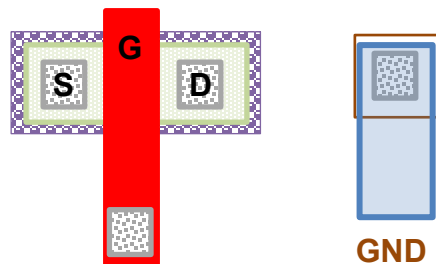
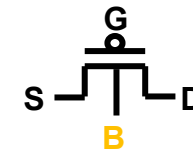
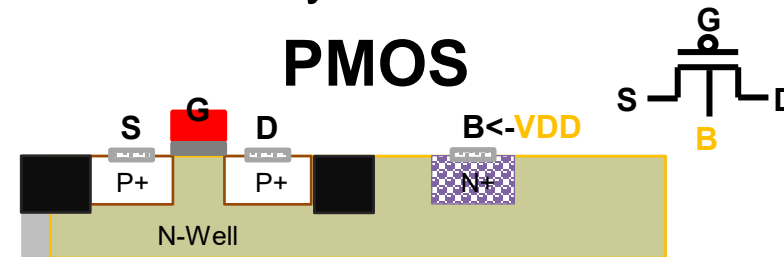
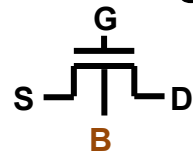
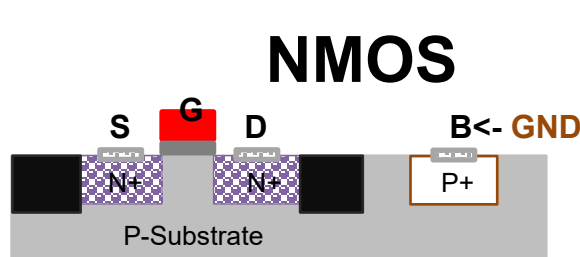
- only P-MOS sits in an N-well, while P-MOS sits directly in the substrate
- N+ and P+ can extend beyond the active/diffusion area for clarity
- Transistor defined by N+/P+ DIFFUSION which is the area in which P+/N+ overlap with DIFF



- **Substrate/Well contacts** (4th terminal **B**) **need to be added**, but can be shared (see next slide)

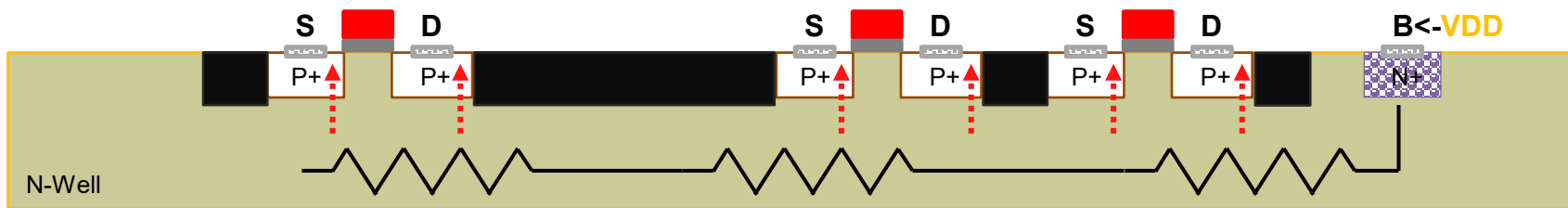
# Layout Structures: Substrate/Well Contacts

- For correct operation, the **P-Substrate** and the all **N-Well**s must be connected to **GND** and **VDD**
  - Substrate and well connections form the bulk (B) connections of P-MOS and N-MOS
  - Need many substrate/well connections across the chip for clean bulk potential (see next slide)
  - P-Substrate is connected through a P+ implant connected by a contact
  - N-Well is connected with through an N+ implant connected by a contact



# Layout Structures: Substrate/Well Contacts/Taps

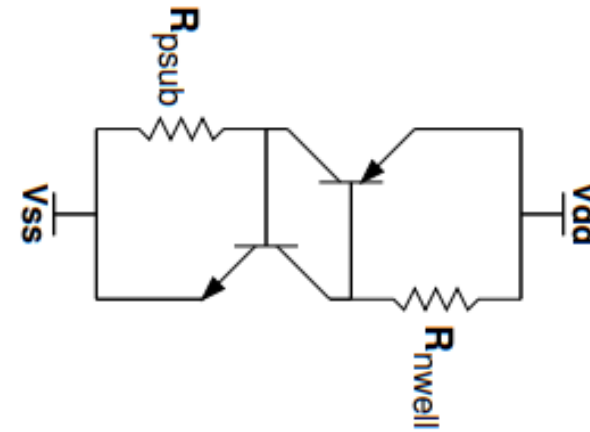
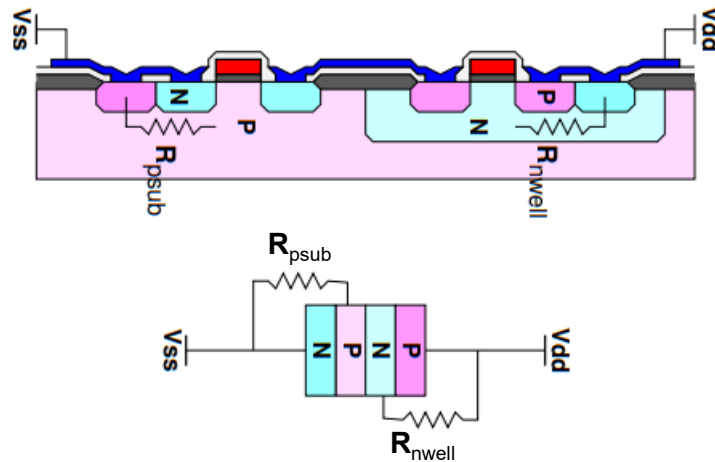
- **Substrate connections are important to**
  1. Correctly define the potential of the transistor bulk terminal impacts its characteristics)
  2. Avoid latch-up (see next slide)
- **A single (or few) connections are insufficient even when wells/substrate regions are connected since substrate/well resistance is high**
  - Substrate currents (e.g., from coupling or leakage) cause a local drop/rise in substrate potential
  - Every diffusion (transistor) leads to leakage currents into the substrate that must be absorbed



- **More substrate connections are always better, but minimum defined by**
  - The design rule manual **LATCH-UP RULES** (see next slide)
  - A rule of thumb: **5-10 transistors per connection**

# Latch-up

- **LATCH-UP** is a condition that arises **from too few substrate/well connections**
  - A parasitic short circuit causing failure (and sometimes destruction) of the chip
- **Latch-up is caused by**
  - Parasitic planar BJTs transistors formed by a) substrate, b) n-well, and c) diffusion regions
  - Potential differences in substrate and well leading to currents Emitter-Base and Base-Emitter currents that turn on the parasitic BJTs
  - Turned on BJTs creating a) a feedback that enforces this state and b) a VDD-GND short

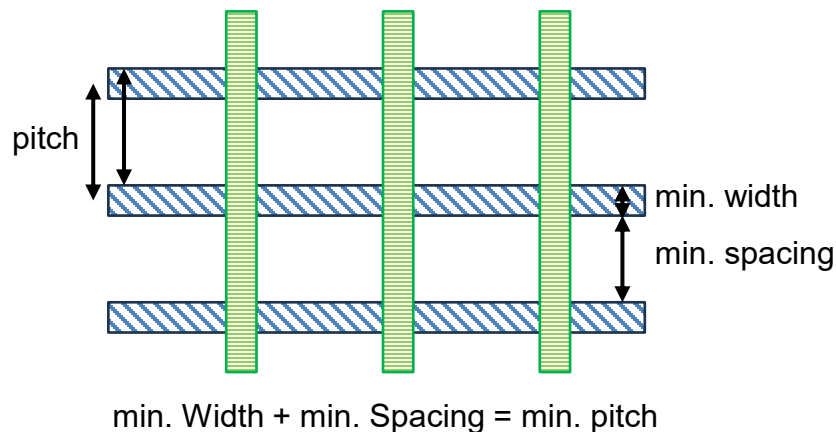
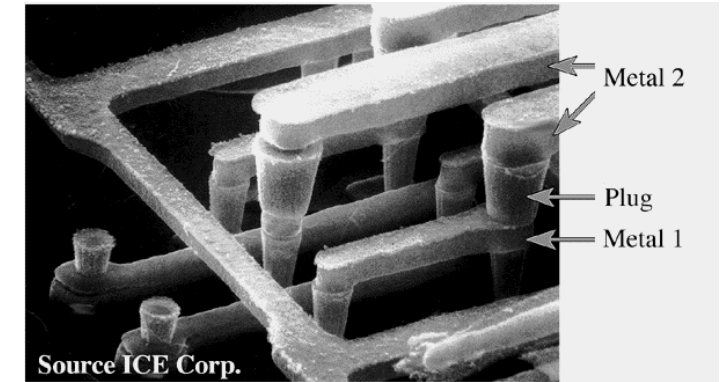


**LATCH-UP** rules for substrate/well contacts/taps are defined in the DRM

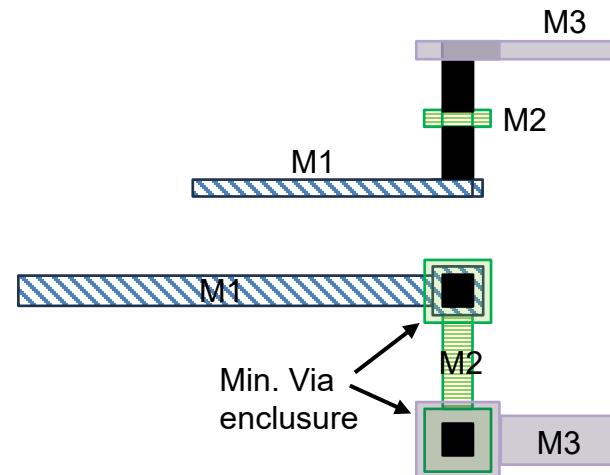
# Layout Structures: Interconnect & Vias (BEOL)

- Transistors are connected with BEOL interconnect/routing on multiple layers, connected by Vias

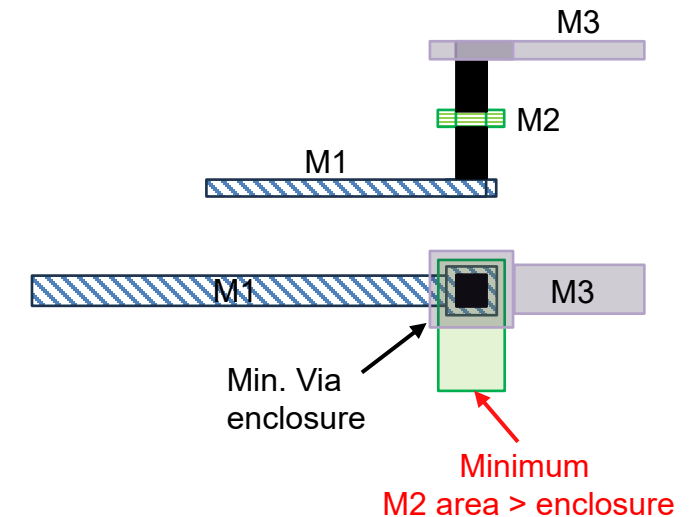
- Each layer has minimum spacing and minimum distance rules
- We often use Manhattan style routing: **each layer has a preferred direction** (horizontal or vertical), which change from layer to layer
- Vias must be enclosed by metal (min. enclosure rules)
- Some technologies allow to **stack Vias**



M1-M2-M3  
without stacked Vias



M1-M2-M3  
with stacked Vias



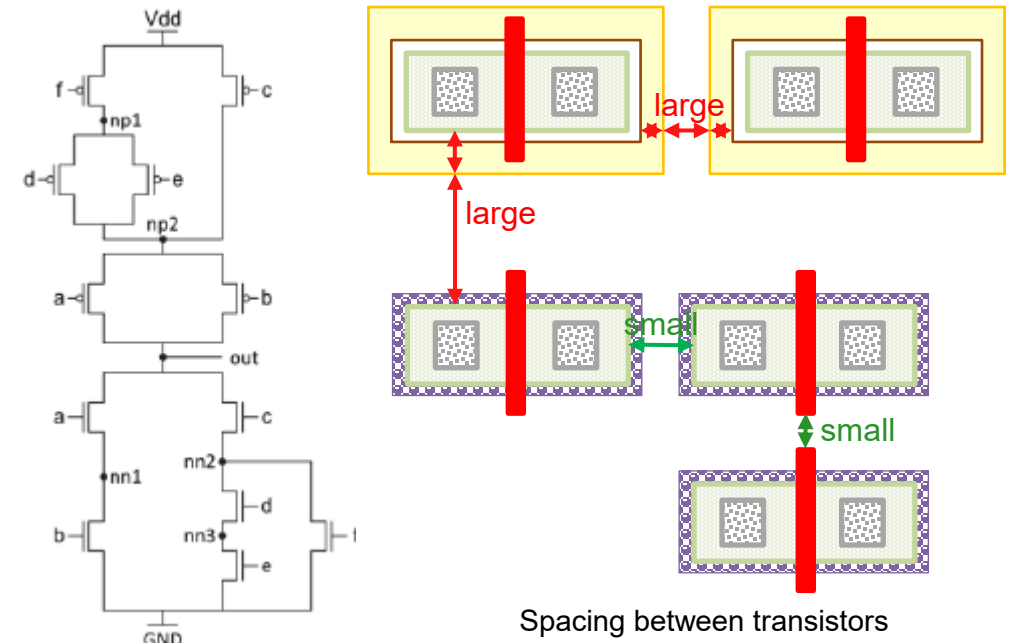
# Manufacturing Grid

- **Placing layout elements on arbitrary coordinates (even when meeting the design rules) will**
    - Make it difficult to work due to the absence of a snap-grid
    - Lead to issues due to manufacturing as coordinates ultimately must be rounded
  - **Definition of a “manufacturing grid” avoids these issues**
    - Defines a reasonable minimum grid size to which every element must snap
    - **Defined in the DRM and in the technology file**
    - **Must be set correctly in the settings of the layout editor** (often done automatically through the technology file)
    - The “manufacturing grid” is **much smaller than the minimum design rule** (typically  $\sim 0.005\mu\text{m}$ )
    - Changing the manufacturing grid in the layout editor makes drawing valid layouts impossible
- Never touch the manufacturing grid setting in the Layout Editor!!!**

# Arranging Transistors

- **Objective:** arrange transistors to **avoid unnecessary area penalties AND keep your layout structured and regular**
- **Few useful observations:** for now, just believe me, you will see they are true

- Active layers require relatively large separation
- N-well spacing is particularly large, but **N-Wells can be merged**
- PMOS/NMOS spacing is typically larger than spacing between transistors of the same type
- For CMOS:
  - Number of NMOS and PMOS is often identical or at least very similar
  - NMOS/PMOS transistors come in pairs: each input connects to the gate of one NMOS and one PMOS

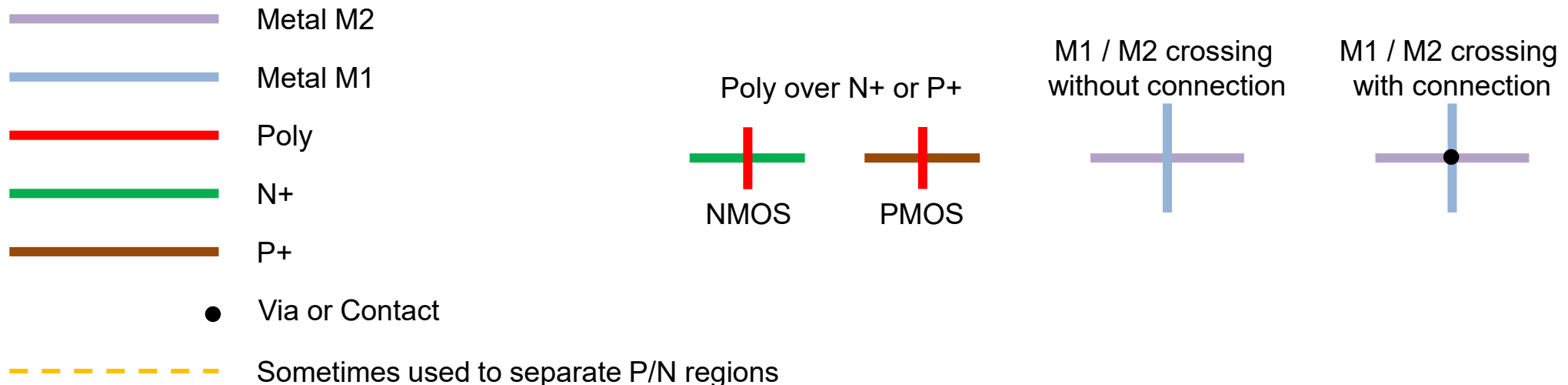


CMOS gate example:  
same number of PMOS/NMOS

- **Grouping all NMOS and all PMOS saves area** (avoids unnecessary spacing)

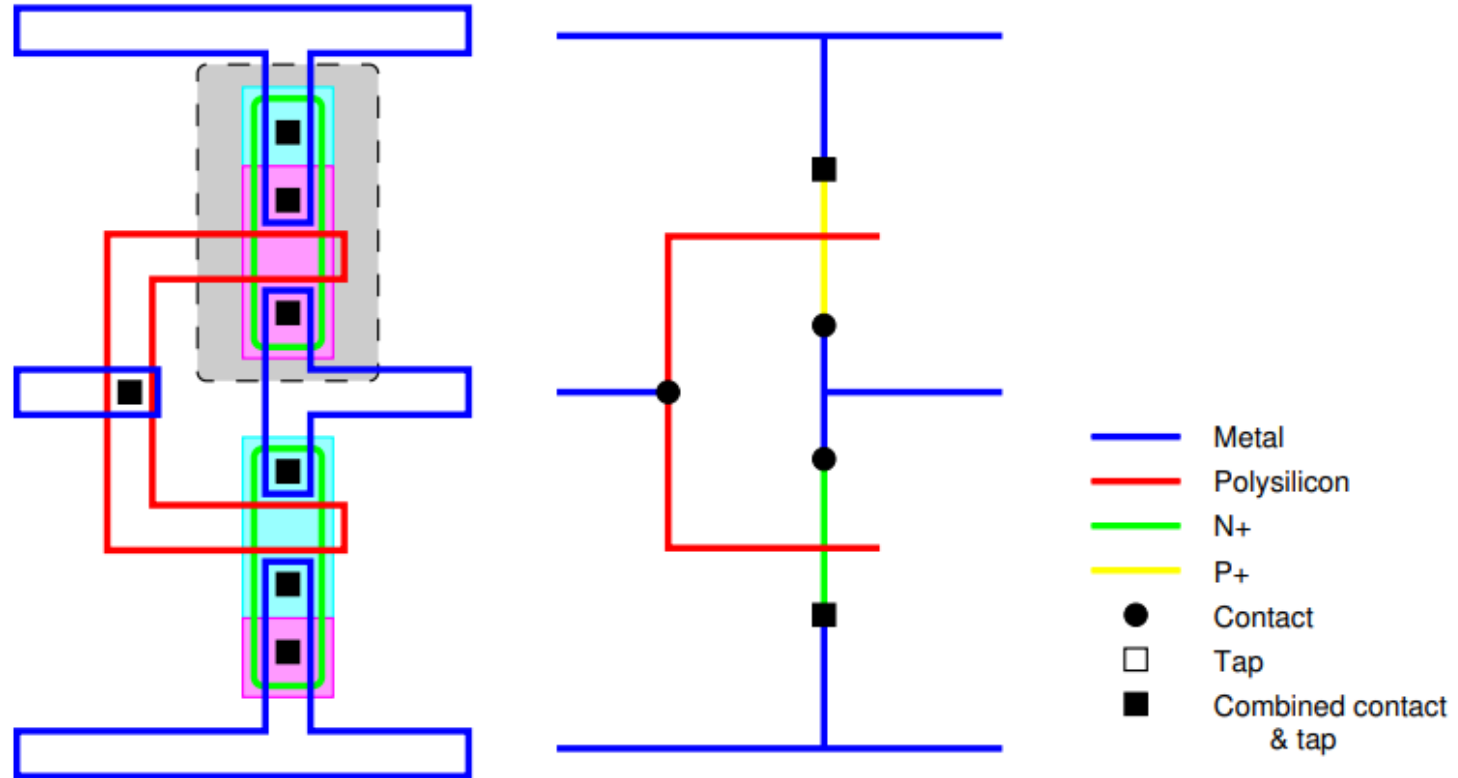
# Stick Diagrams

- **Simple way to quickly evaluate layout topologies**
  - Contains all transistors, their connections (contacts) and, interconnect (wires, vias),
  - Shows approximate relative placement and orientation of transistors and routing
  - Does not contain information on sizing (1D transistor model)
  - Ignores most design rules: no information on spacing, thickness, enclosure, ...
- **Basic elements of a stick diagram: color coded lines**



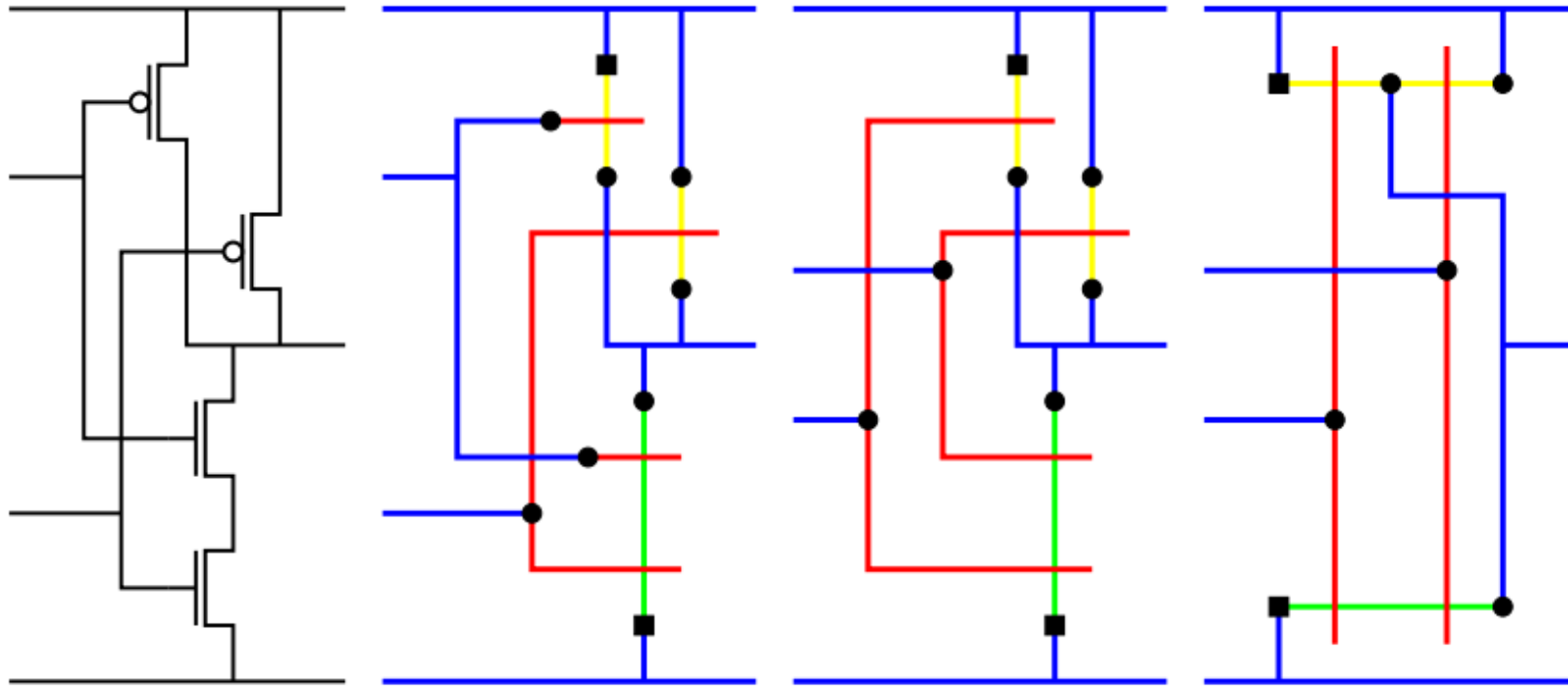
# Stick Diagram Examples

- Inverter:



# Stick Diagram Examples

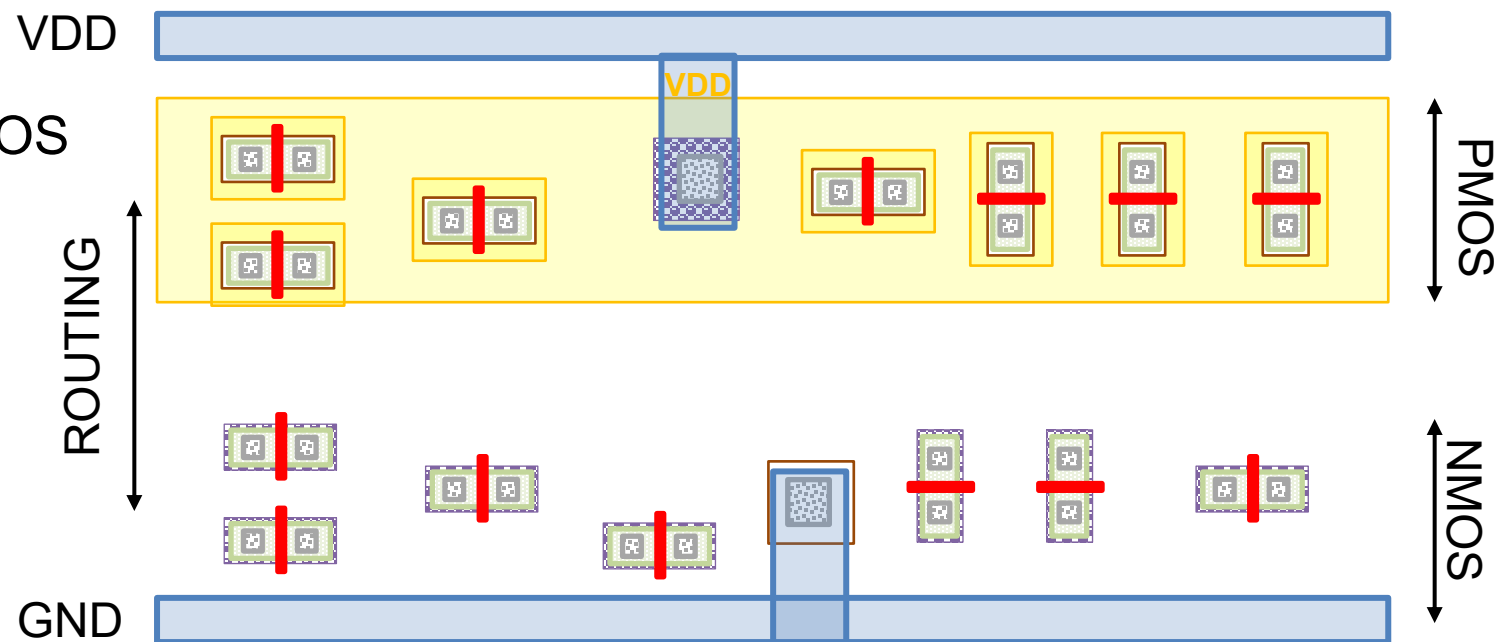
- NAND-2 gate:





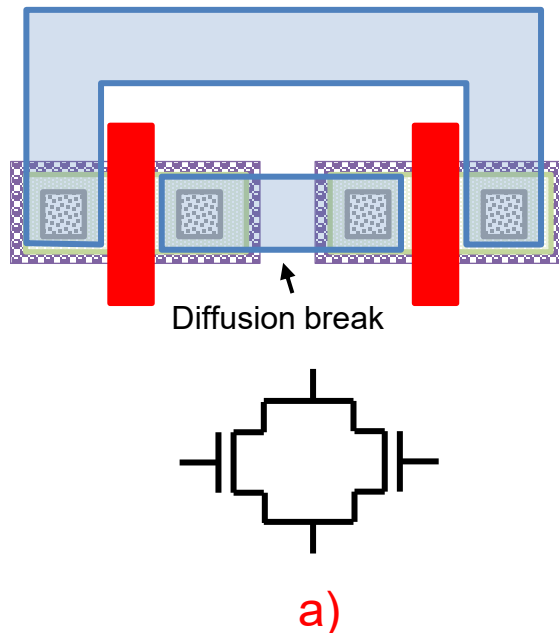
# Layout Template for Logic Cells

- **Separation of NMOS and PMOS combined with good scaling behavior with the number of transistors favours a layout in which**
  - NMOS and PMOS are separated, but have same orientation
- **NMOS are exclusively connected to GND, while PMOS are exclusively connected to VDD**
  - GND routed on NMOS side and VDD routed on PMOS other side
  - Easily connects to NMOS and PMOS source contacts
  - Easy connection to substrate and well contacts

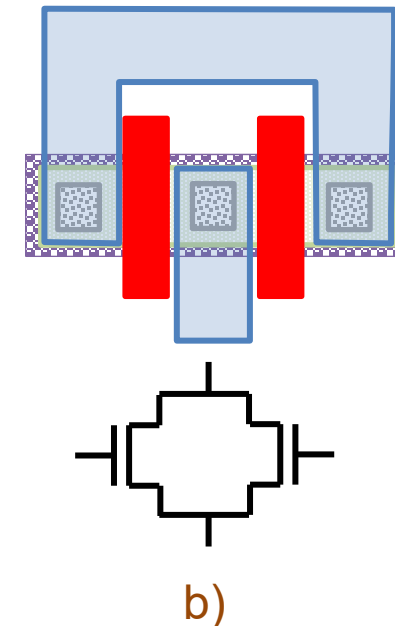


# Merging Diffusions for Parallel Transistors

- We often find transistors of the same type that are connected in parallel
- These can often be placed next to each other (avoid “diffusion breaks”)
- Three possible realization in the layout with very different size
  - a) Individual complete transistors: 2 separate diffusion regions + spacing
  - b) Contacted shared diffusion region: 1 diffusion region only, but still large due to contact

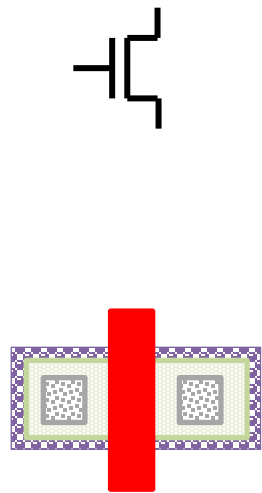


>

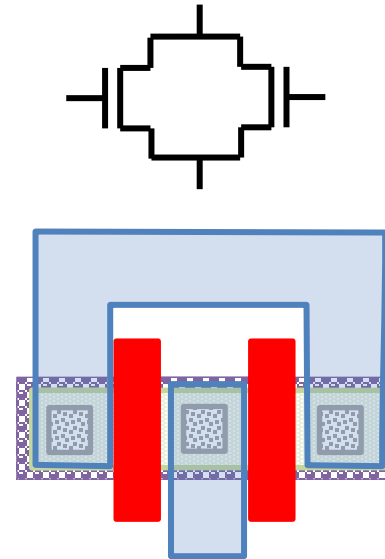
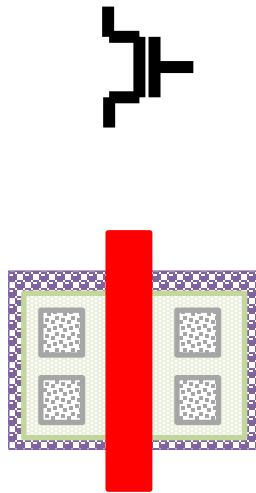


# Folding Large Transistors

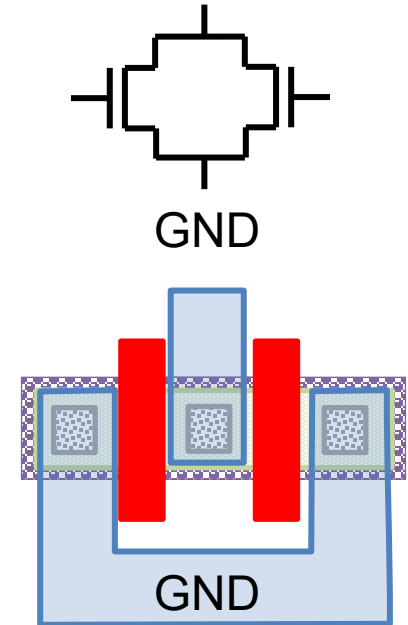
- **We often encounter transistors that are larger than min. size or than others**
  - Large transistors incur a large diffusion area, i.e. also large intrinsic capacitance
- **Larger transistors can be replaced by parallel transistors**



Two large diffusion areas  
( $C_s \sim 2W$  &  $C_d \sim 2W$ )



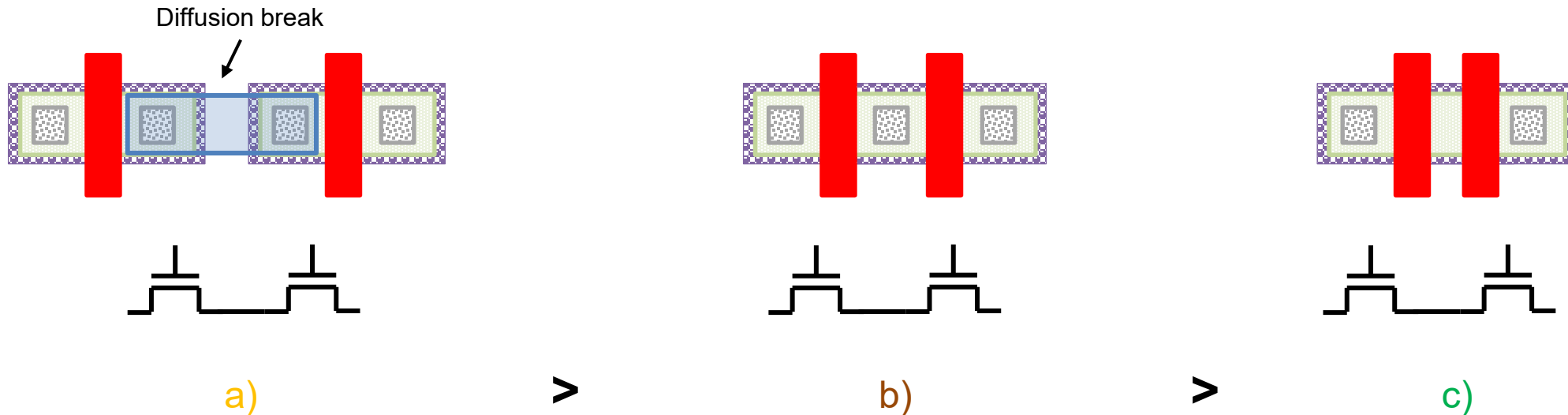
Two large diffusion areas  
( $C_s \sim 2W$ ,  $C_d \sim 1W$ )



Only signal diffusion matters  
( $C \sim 1W$ )

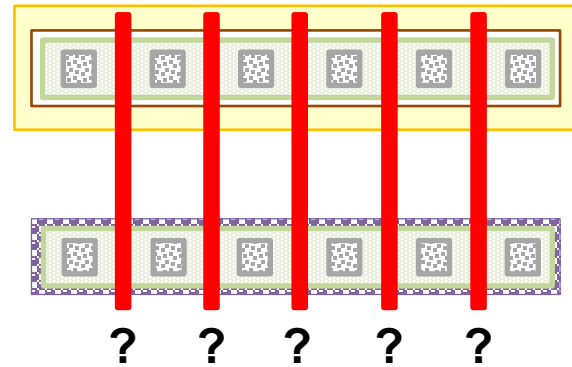
# Merging Diffusions for Series Transistors

- We often find transistors of the same type that are connected in series
- These can often be placed next to each other (avoid “diffusion breaks”)
- Three possible realization in the layout with very different size
  - a) Individual complete transistors: 2 separate diffusion regions + spacing
  - b) Contacted shared diffusion region: 1 diffusion region only, but still large due to contact
  - c) Shared (uncontacted) diffusion: separation only by 1 poly pitch



# Euler Path Algorithm: Objectives

- **Placing connected transistors next to each other is very efficient**
  - Allows sharing of source/drain areas and avoids costly “diffusion breaks”
  - Minimizes routing parasitics
- **Every input is connected to an NMOS and to a PMOS gate**
  - Placing corresponding PMOS and NMOS such that their gates align allows to use a single straight and short ploy to connect both gates



- **Need a method to order transistors to avoid diffusion breaks and to align corresponding NMOS/PMOS pairs**

# Euler Path Algorithm: Approach

- **Euler path: path through a graph that visits each edge only once**
- **Euler method:**
  - Consider the PMOS and NMOS networks of a gate as a graph
    - The transistors are the edges of this graph
  - Find all possible Euler paths for both PMOS and NMOS networks
    - Placing transistors in the order of any Euler path completely avoids any diffusion break
  - Among the path candidates for the PMOS and NMOS networks choose one that is identical (same order of gate inputs) for both networks
    - Same path (same order) allows to align corresponding PMOS and NMOS transistors
- **There is no guarantee that an Euler graph (or a matching Euler graph exists)**
  - If no proper matching Euler graph exists, break the graph into two or more sub-graphs

# Euler Path Algorithm: Examples

- There is no good algorithm to find the Euler path, but an exhaustive search often has manageable complexity

- **Example 1:**

- 4 valid options

A - C - B

B - A - C

B - C - A

C - A - B

A - B - C

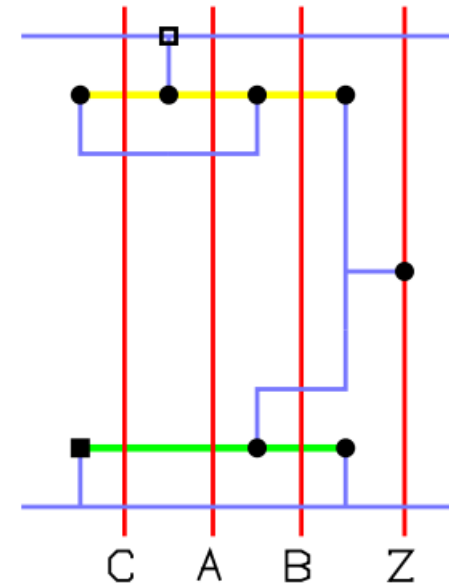
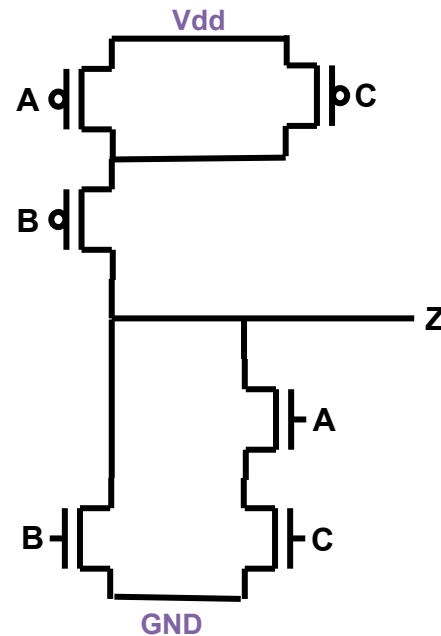
A - C - B

B - A - C

B - C - A

C - A - B

C - B - A

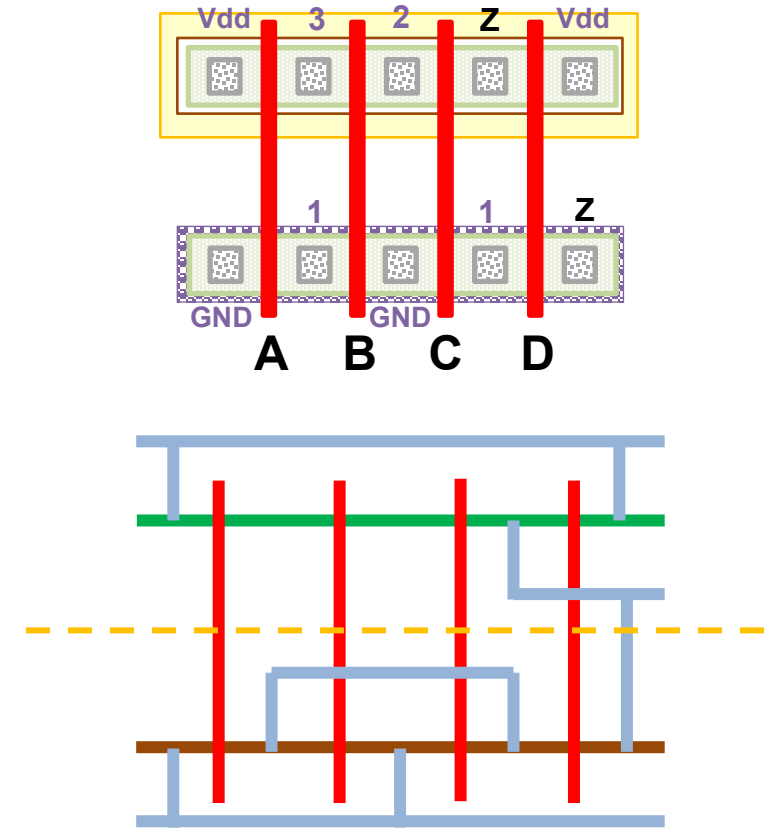
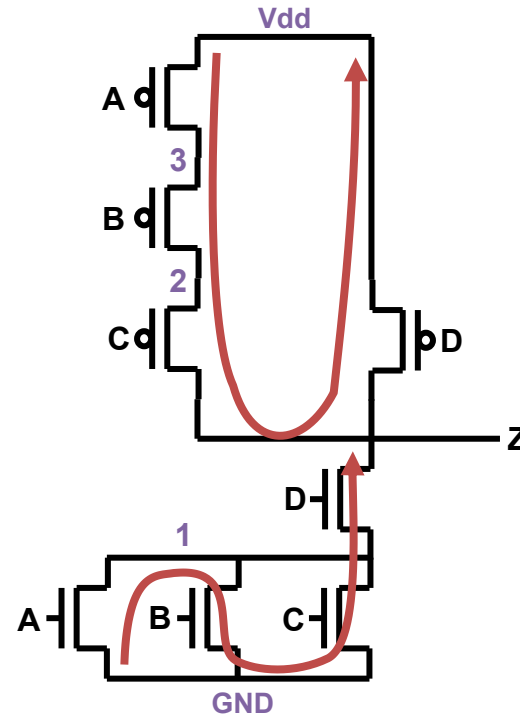


# Euler Path Algorithm: Examples

- Often, we don't need to search all paths, but we can directly look for a matching path of PMOS and NMOS since any matching path is fine

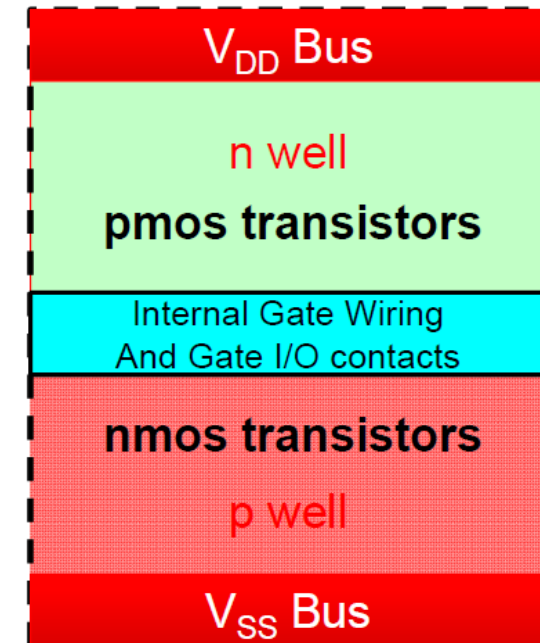
- **Example 2:**

- A – B – C – D works for both PMOS and NMOS



# A Scalable Layout Template for Digital Cells

- In digital designs we often need a set of logic cells that are later combined in various ways to realize more complex multi-level logic
- **Idea: define a logic cell layout template in a way that**
  - Physical cells can easily be combined in an almost arbitrary fashion
  - The template accommodates cells with very different complexities (e.g., inverters, basic logic gates, full adders, sequential elements)
- **The standard-cell layout template**
  - All cells have the same height
  - Only width varies depending on the complexity of the cell
  - Power and ground connections run horizontally with same height
  - Cells can abut without DRC violations
  - PMOS on one side, NMOS on other side
  - Use only few layers to leave other layers for cell-to-cell routing



# Layout Optimizations

- **Long Poly lines and especially chaining gate connections leads to very long RC delays**
  - Poly has both a high resistance AND a high substrate capacitance
  - Poly gates have an even higher capacitance

