CHAPTER-2
STANDARD CELL TECHNIQUES

IC Mask Design

Faculty of Engineering
Alexandria University

Chapter Preview

- Why standardization is mandatory in digital layout
- Advantages of standardization techniques in analog layout
- Why we butt some cells together
- Tips if you have few metals
- Tips if you have lots of metals
- Digging channels for our wires
- When to run big power lines
- Getting signals in and out of tight spots
- How to guarantee a good fit between cells
- How to guarantee rule-perfect layout
- How to save time

Opening Thoughts on Standard Cell Techniques

- In order for automated layout tools to be able to place and connect components. Rules for the cells, rules for placement, and rules for connectivity should be defined.
- Standard cells also need to fit together in predictable patterns. A variety of standardization techniques are used to build a library of specially designed cells.
- These techniques are useful in analog mask design

Standardized Grids

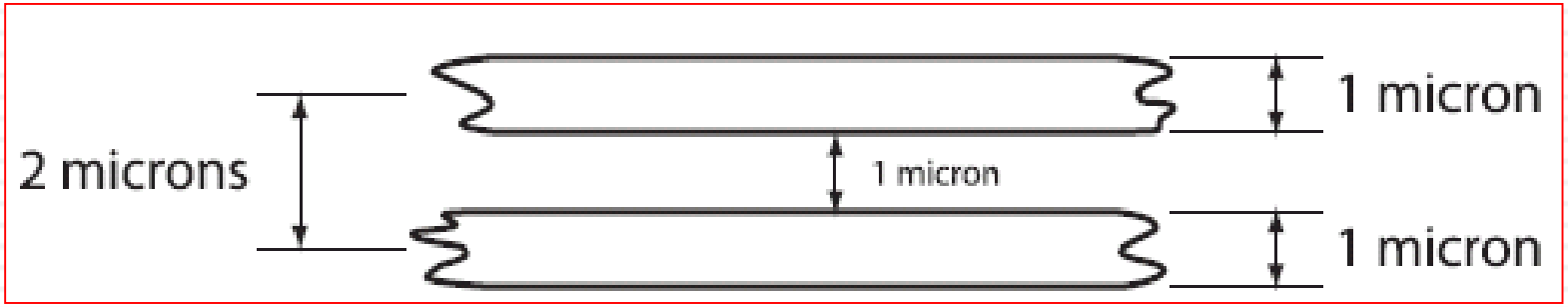
- This standardized layout system enables automated wiring and guarantees overall operable placement of standard cells, by aligning everything on a standard grid.
- Standard cells operate on this grid. The cells are designed uniformly, watching grid alignment and rules, an automated tool reign over placement and wiring. The circuit will be built correctly regardless of the software decisions.

Grid-Based Systems

- The classic router software is **grid-based.**
- A grid-based router has two constraints.
  - Wires can only be certain widths,
  - Wires can only be placed on a predefined coordinate grid.
- We cannot just freely design anything we want in a grid-based system. We must follow grid alignment rules.

Determining Grid Size

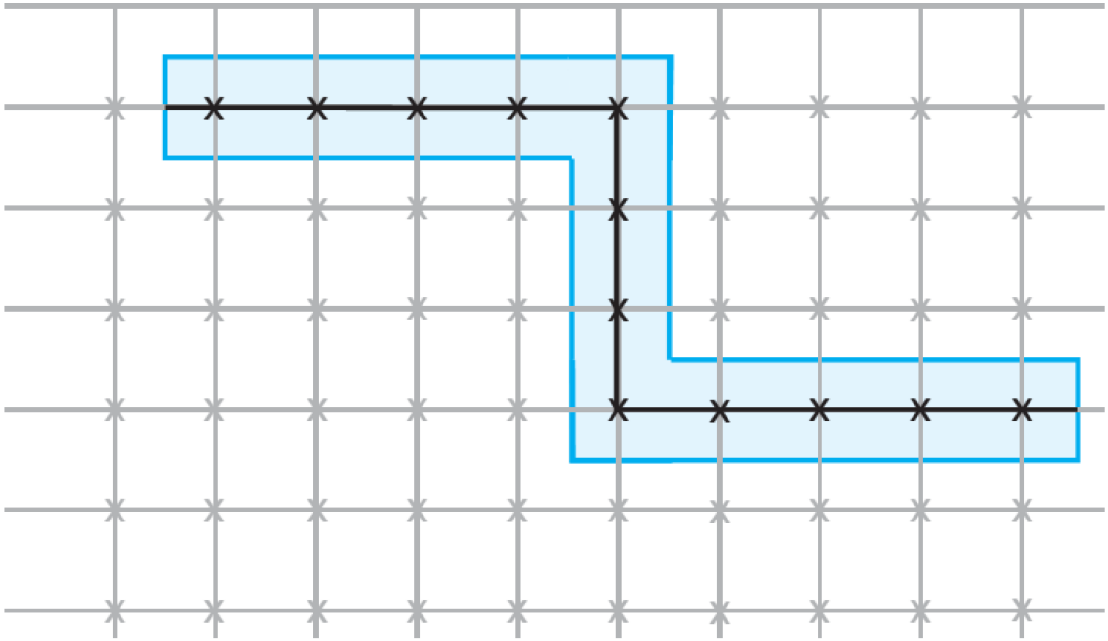
- Suppose the minimum Metal One wire width is 1 micron, and the minimum spacing is 1 micron.
- Therefore, our minimum distance for two parallel wires would require a distance of 3 microns. One micron for each wire, and 1 micron for the space between the wires.



- the distance between wire centerlines is half the top metal width, plus the minimum spacing between metals, plus half the bottom metal width. In this example, that's a total of 2 microns. The centerlines of the wires will be 2 microns apart. This makes a **2-micron grid**..

Design rules determine grid size

- Defined x (horizontal) and y (vertical) sets of grid lines across the whole of the chip.
- The grid-based router can only place wires along the grid lines.
- Following grid lines limits, the automated tools do the routing entirely based on these grids.



Different grid sizes on different layers in a grid-based router?

- Different sizes of grids can exist in each layer, but that could be awful trying to line up all our contacts, our vias, matching both horizontal and vertical grid intersections in each layer.

Rule-Based Routers

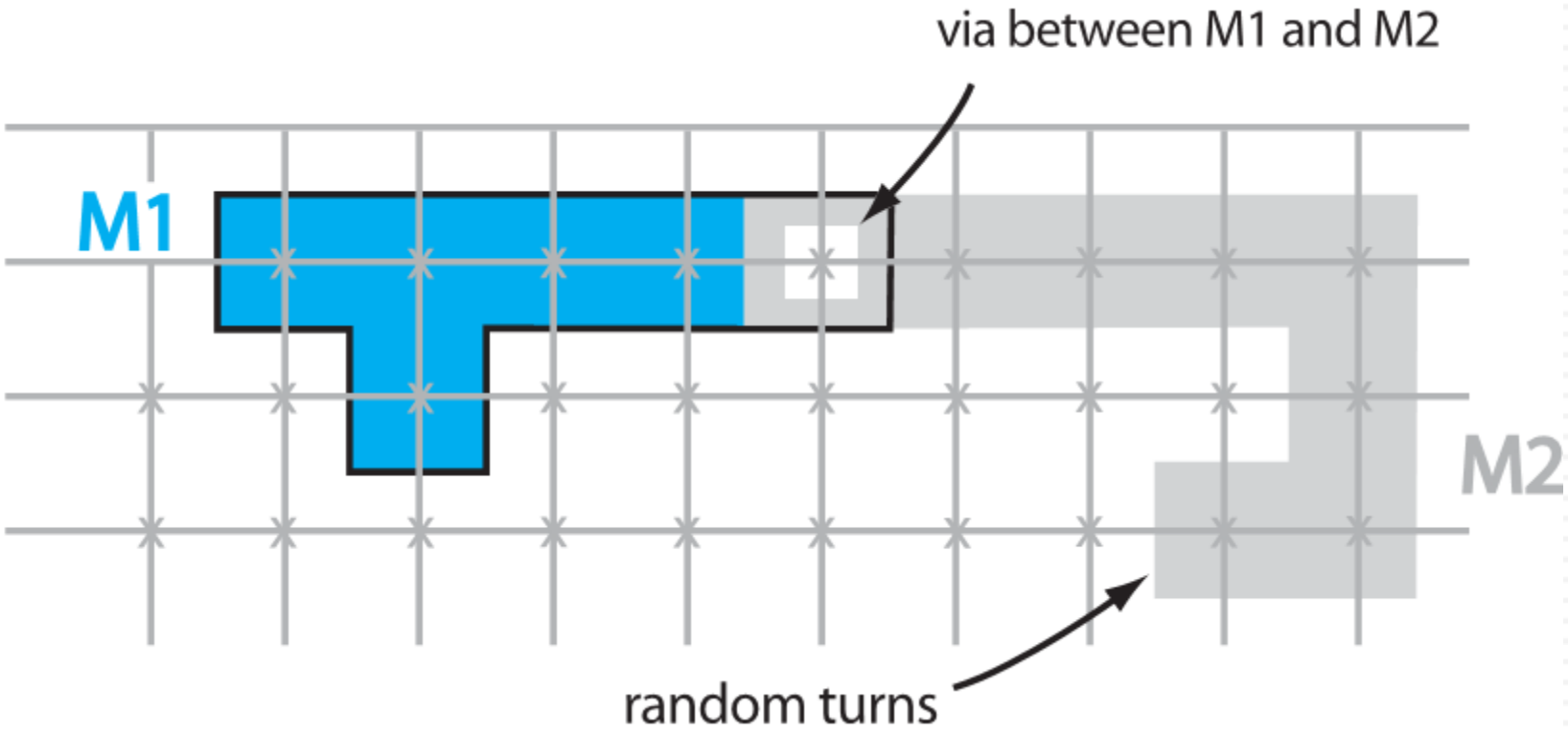
- In modern process, any two grid pitches in a process are typically not the same dimensions.
- Metals and spacing on different layers have different minimum widths.
- If we forced all grids to the same dimensions, we would have to universally use the largest requirements of any layer of the chip. This would waste space in the other layers.
- For example
  - if a layer can tolerate 1-micron spacing, why force it to use 2-micron spacing?

Rule-based router

- Grid-based router is called a **rule-based router.**
- For each layer containing wiring, instead of using a fixed grid, the computer actually uses the real design rules for that layer.
- Characteristics:
  - Most people use the grid-based approach since it is simpler to use. It makes the router's job easier.
  - It doesn't have to integrate an extra set of rules for each layer.
  - With the good old grid-based router, we just tell the software where the grid is, and it always places its wire on that grid. The wiring job is a lot easier, so the software is a lot simpler to write.
- Metal One width and spacing might not be the same as Metal Two width and spacing. Some wiring levels might be more squishable. They might have tighter grid spacing, so you save space in your layout.

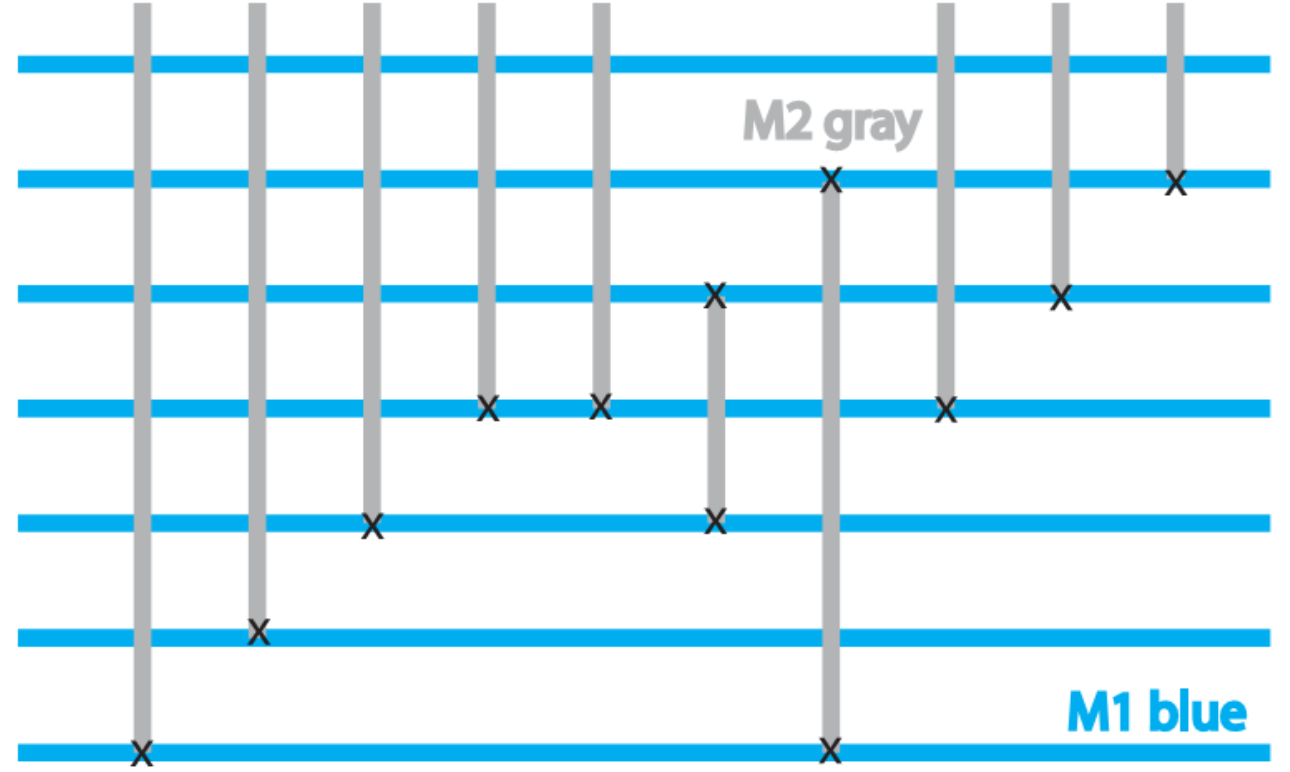
Directional Layer Technique

- Metal Two somewhere are needed to save the wiring from becoming trapped in Metal One.



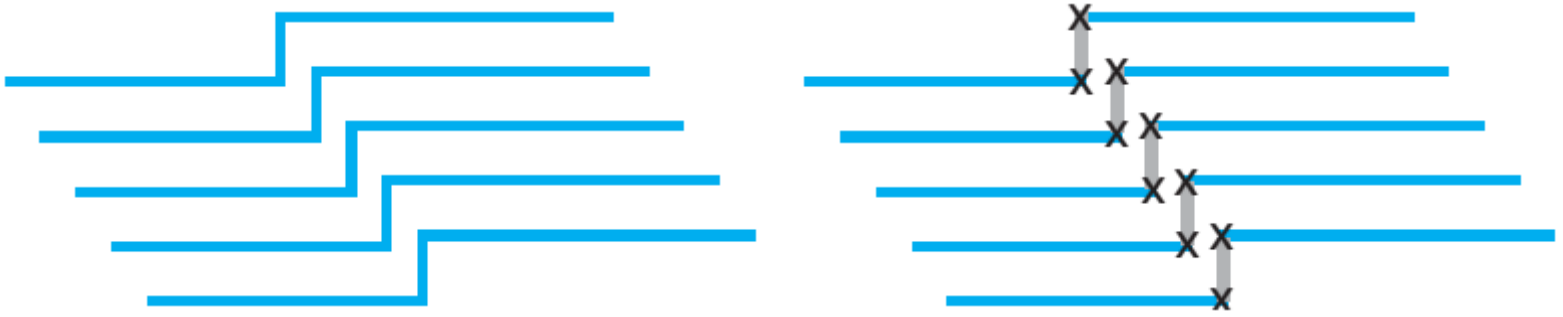
Directional Layer Technique

- Metal One horizontally and running all Metal Two vertically? Ingenious.



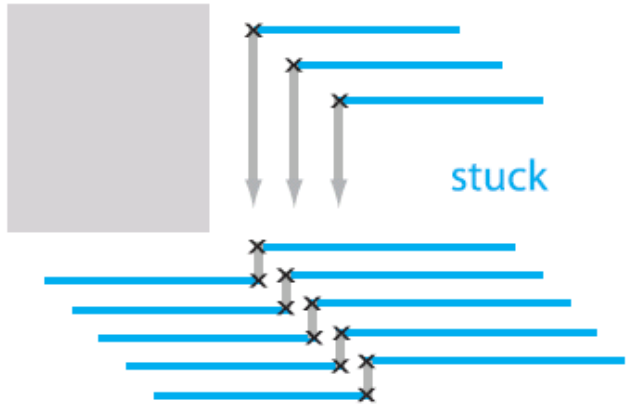
Change directions

- Do you think we still use Metal Two for our vertical run if we are only moving, say, one or two grids?

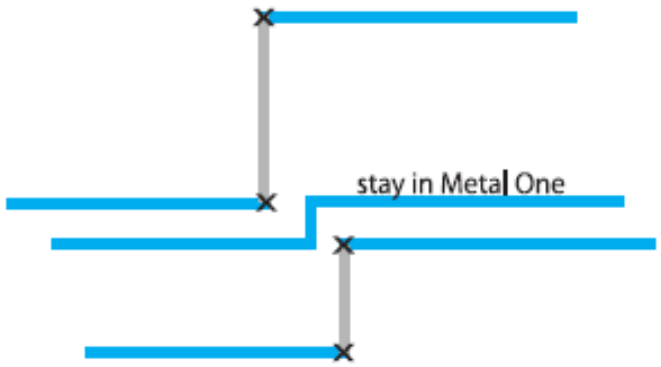


If the vertical jump is only one or two grids, you generally stay in the same metal.

Rule of Thumb



Metal Two is already used in this area.



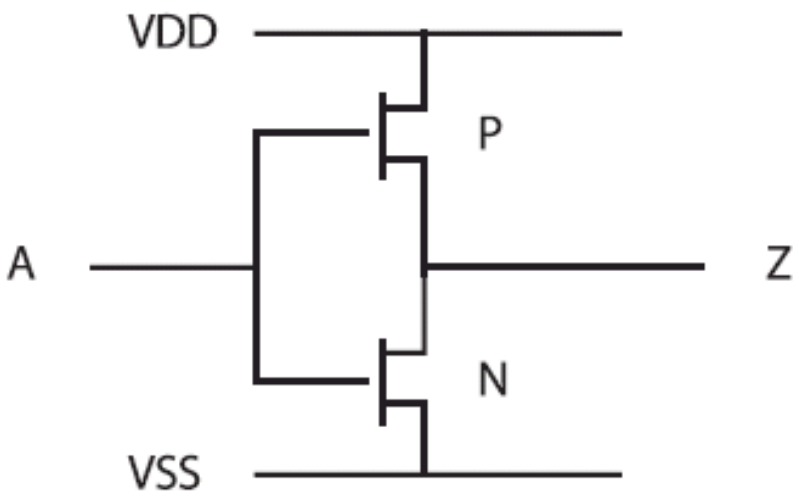
Stay in the same metal for small jumps.

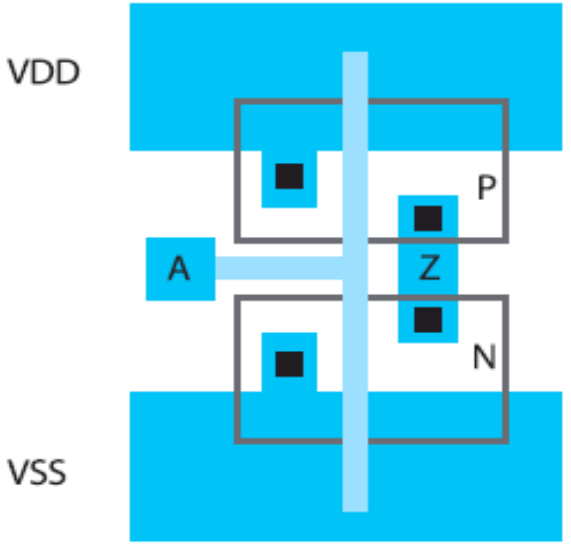
- **Don't bother changing metals for short jumps**
- The vias we introduce for such a small run can potentially introduce high resistances.
- Not only that, but vias can sometimes not etch properly. So, for small jumps of only one or two grids, it is not as reasonable to use your second layer of metal. Stay in Metal One.

Library Rules for Grid-Based Systems

- Devise a set of rules for everything dealing with our layout, when we use a grid-based router.
- Typically, we construct an entire standard cell library according to these rules. Every cell, every inverter, every NAND gate, absolutely everything conforms to the rules.

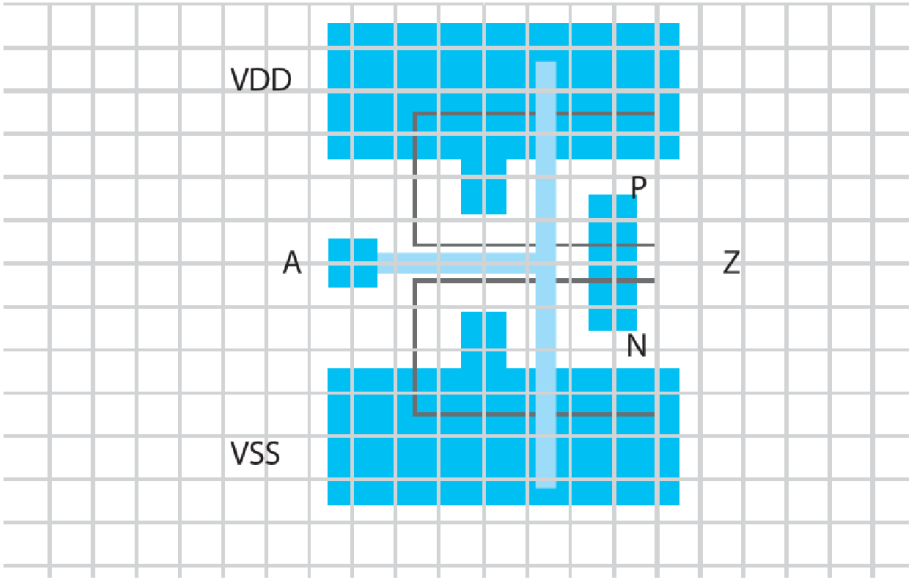
Input and Output Alignment





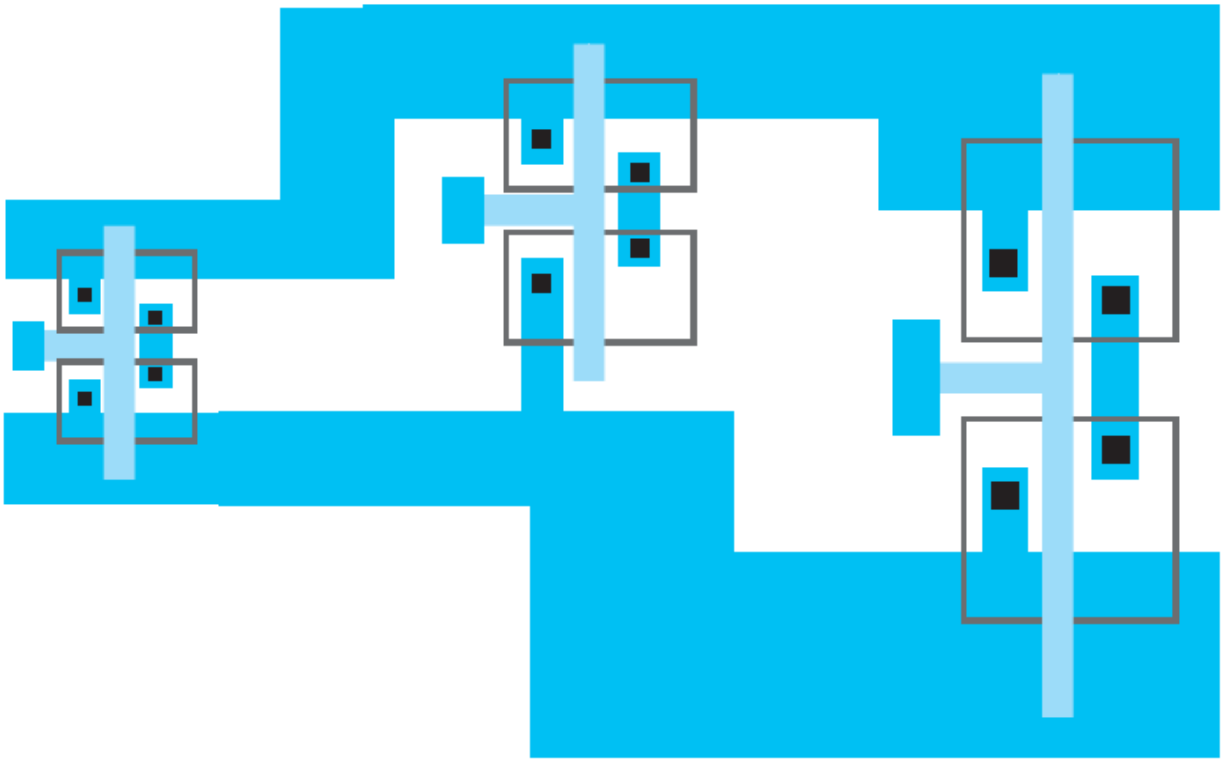
- Schematic reference for our standard inverter cell. We need input A and output Z to align perfectly with our grid wires, or they will miss the connections placed by the auto-router.
- the input and output, A and Z, located in the center of the cell.
- They must be located exactly on the same grid as all the wiring. How else would our wiring attach?

Grid-based Inverter



- All standard cell components have to be matched to the grid in this same way.
- The wires, the cells, the intersections—all layout entities need to obey the rules, such as alignment and isolation distances. Otherwise, we cannot guarantee that our automated system will give us a DRC and LVS clean job.

Fixed Height, Variable Width



Difficult to route power and ground

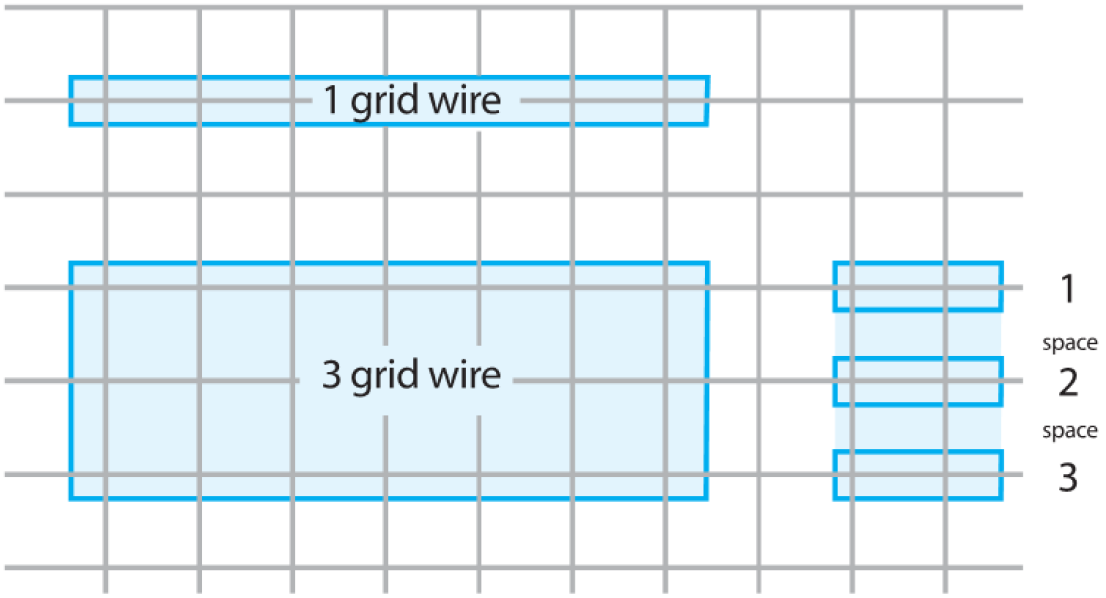
Digital libraries: fixed height, variable width.

- How to deal with large loads
  - If we need bigger logic gates with larger transistors to drive large loads, then we just make the cell wider and split the transistors to fit inside the rails. But, we still maintain the fixed library height.
- Height decision:
  - Typically, select a height that is a bit larger than the minimum, because you want to have a power rail that is larger than minimum.
- Fixed height, variable width is also a very useful technique to use in analog design.
  - If you have a very repetitive design, with lots of similarly sized cells

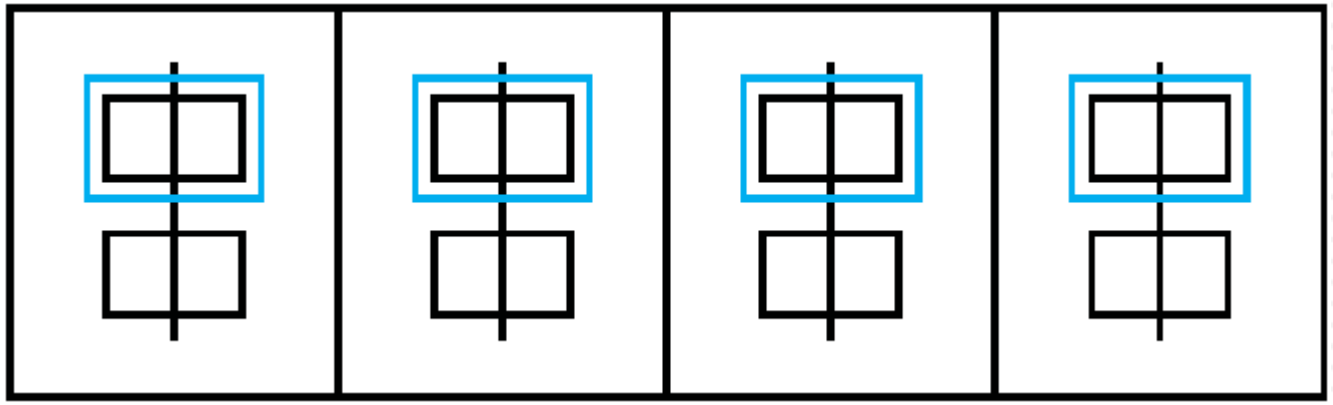
Determining Wire Gauge

- Our 1-grid wire sets the minimum distance between grid lines, as we saw earlier.
- One wire runs along one grid line. We can also make wires of larger gauges. The power rail in our inverter example is what we call a 3-grid wire. Typically, power rails are either 2- or 3-grid wire.

- To build a 3-grid wire, place three single grid wires on-grid, running side by side, and then fill the gaps between them with metal.

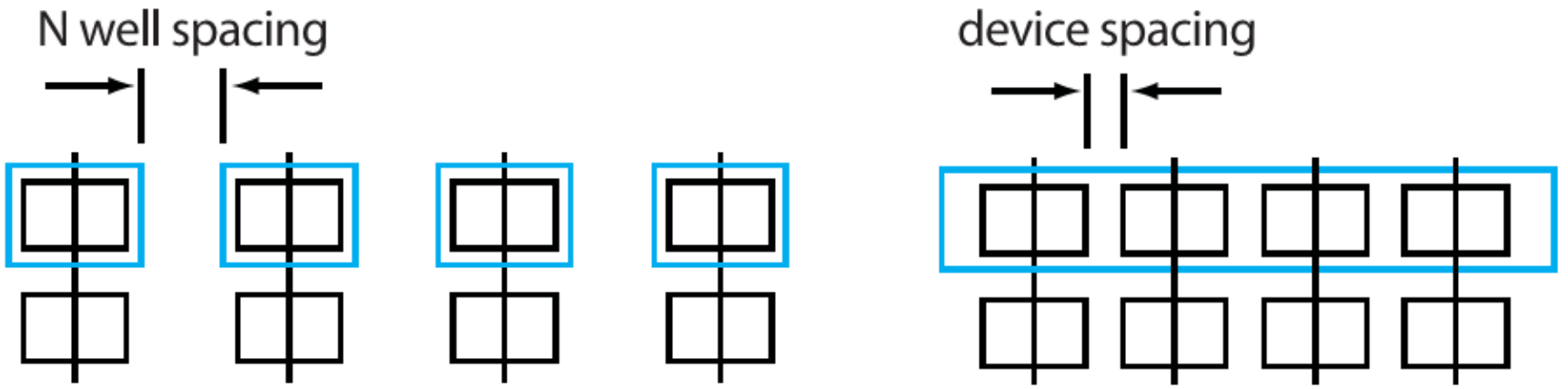


Common nWell



- Suppose we want to place four gates next door to each other. Typically, we want to place them as close as possible.
- The above design method wastes large amounts of space. Luckily, because most logic circuits have the PMOS devices connected to VDD along with the N well, we can create one large single N well and save space.

Common nWell



- one large N well now means that our limiting design rule is the transistor-to-transistor rule, which is much smaller. We can place our devices closer together by sharing the N well.

Common N Well

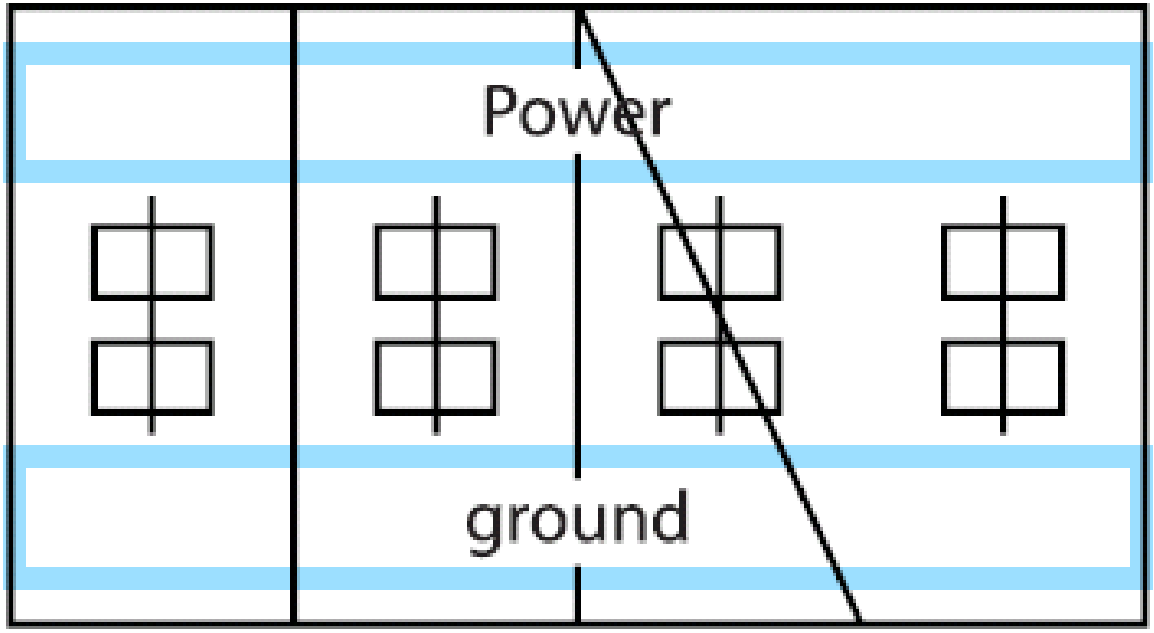
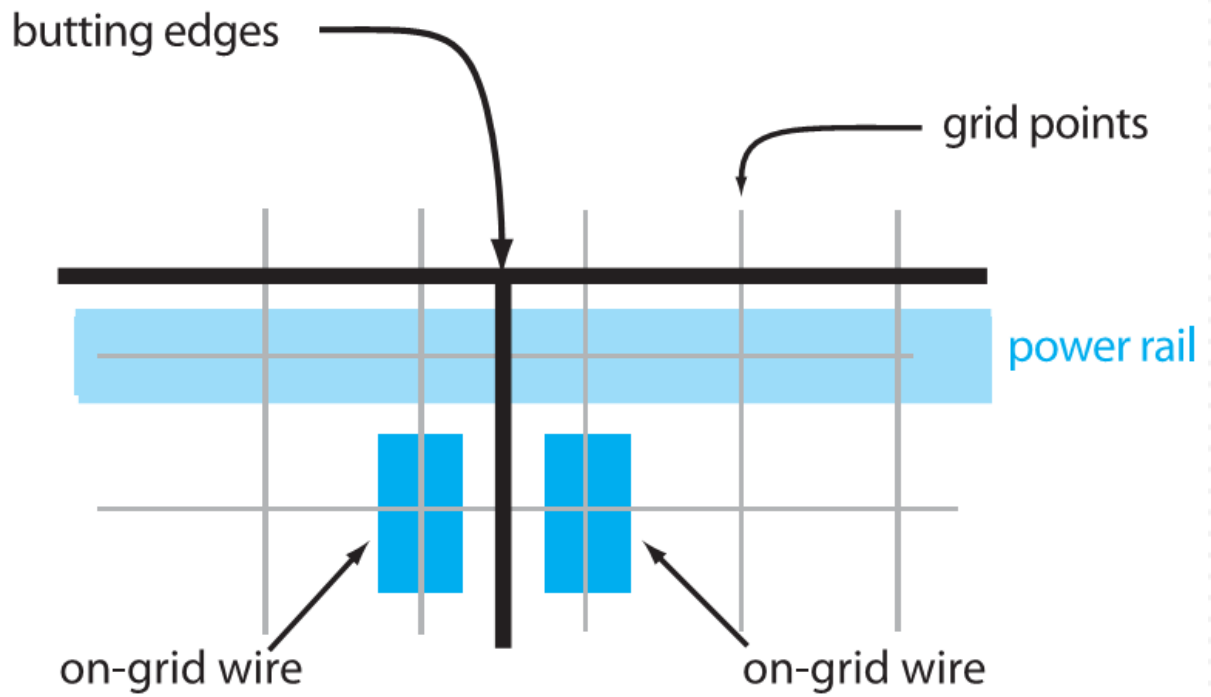


Figure 2–15. If the rail material extends to the edges of the cells, butting the cells together forms one long power rail.

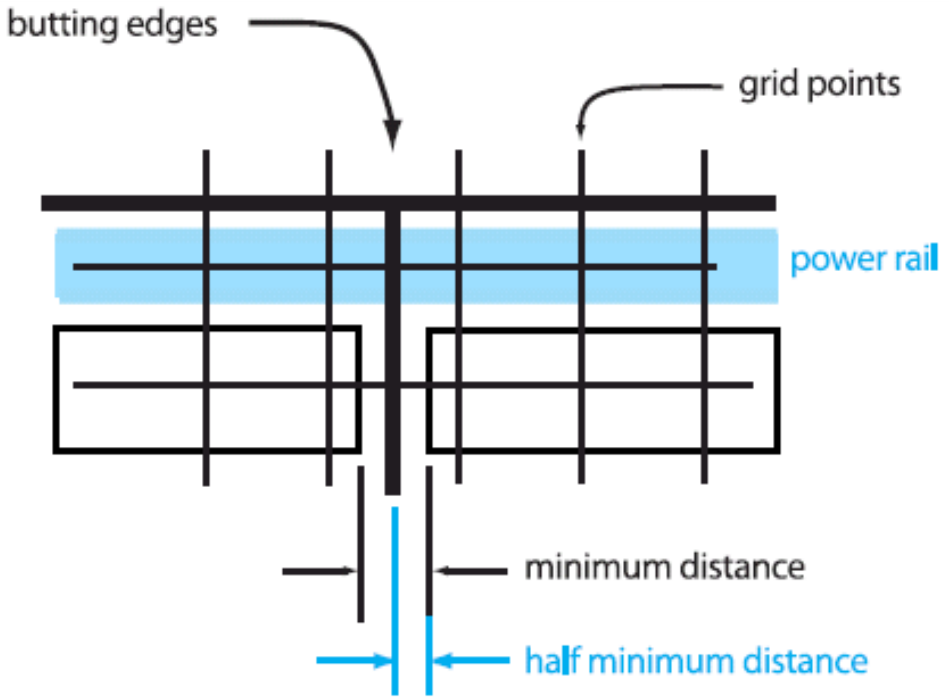
- The N well and the power rails butt against each other forming long, continuous N well and power rail strips.

Half-Grid Cell Sizing



- keep all internal wiring on-grid. The ends of the cells that butt against each other falling between gridlines, on the half-grid.
- so every edge of a cell—top, bottom, left, and right—needs to end on a half-grid. That keeps our internal components properly spaced on all sides.

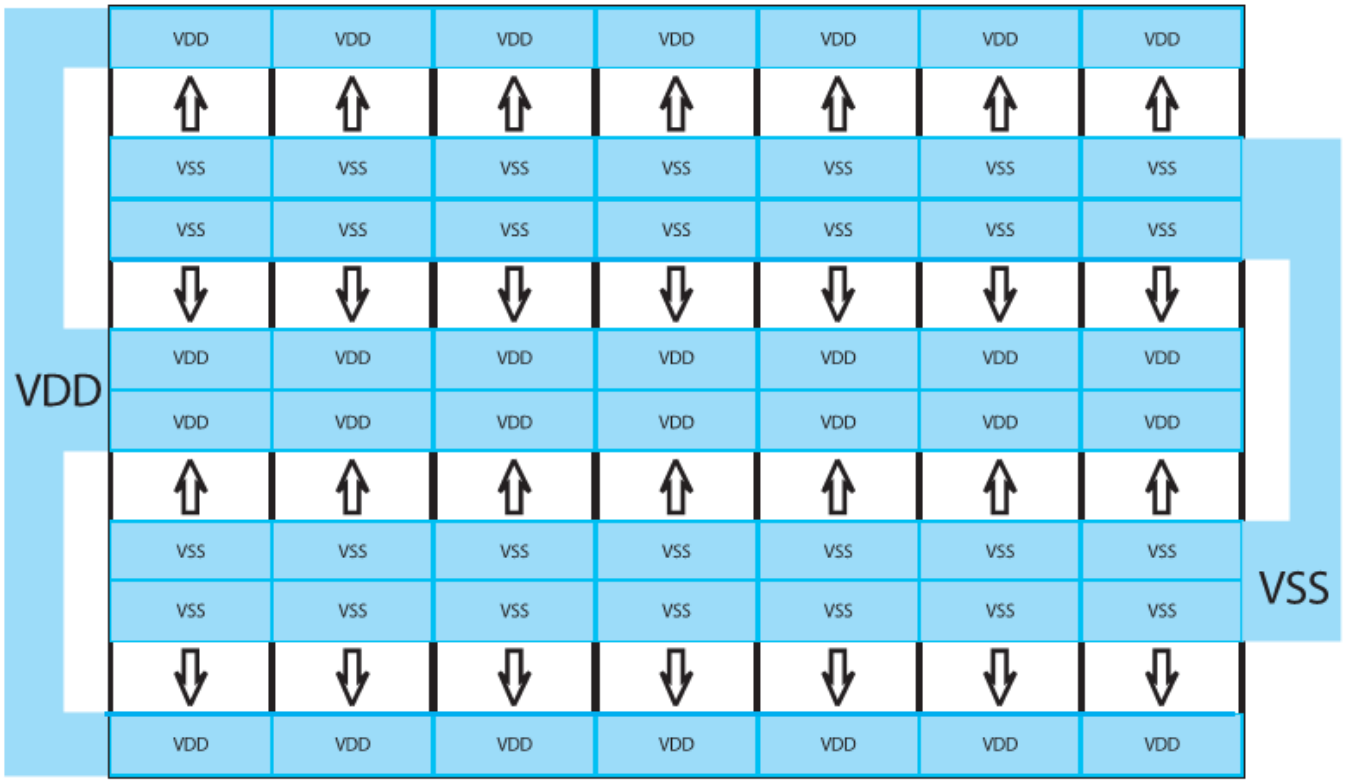
Half Design Rule



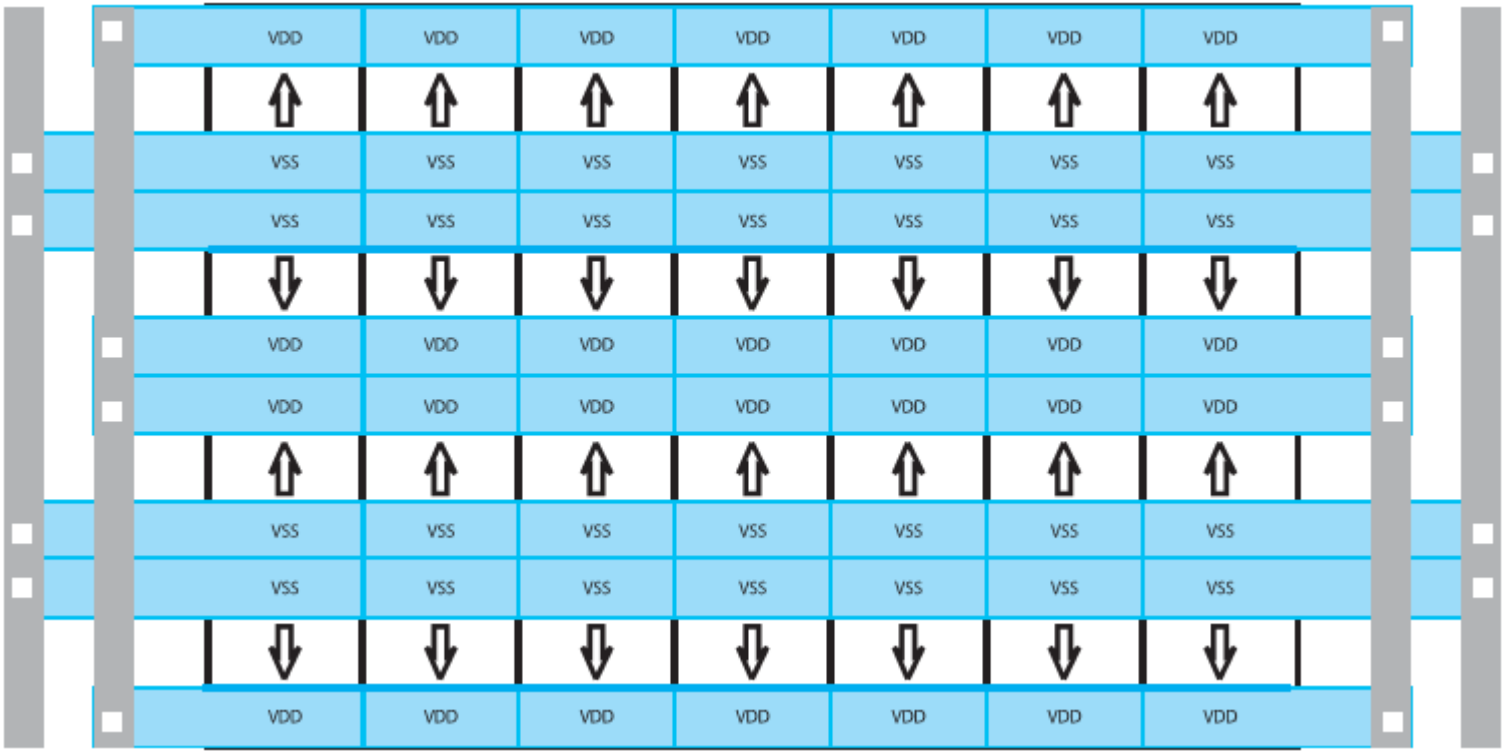
- Placing components one half minimum design rule distance from each edge puts one full design rule distance between components.

Routing Channels

- Since the centers of your cells are filled with components, we have no room to run additional wiring
- If we have very few metal layers to work with, this is a big problem

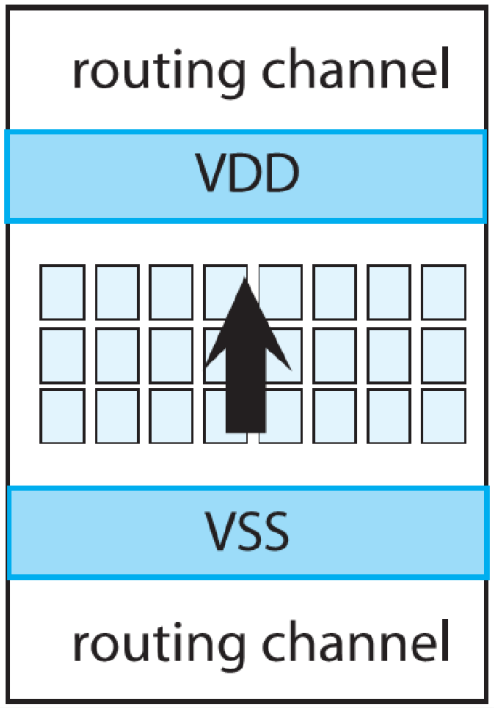


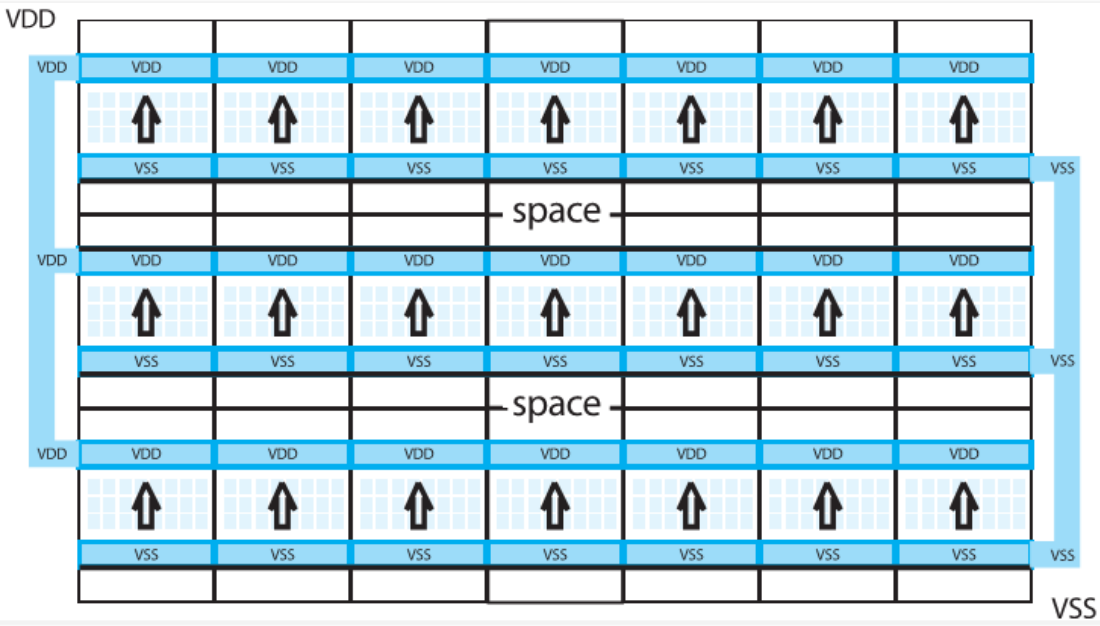
Power & Ground Connection



Typical power ring using Metal Two buses

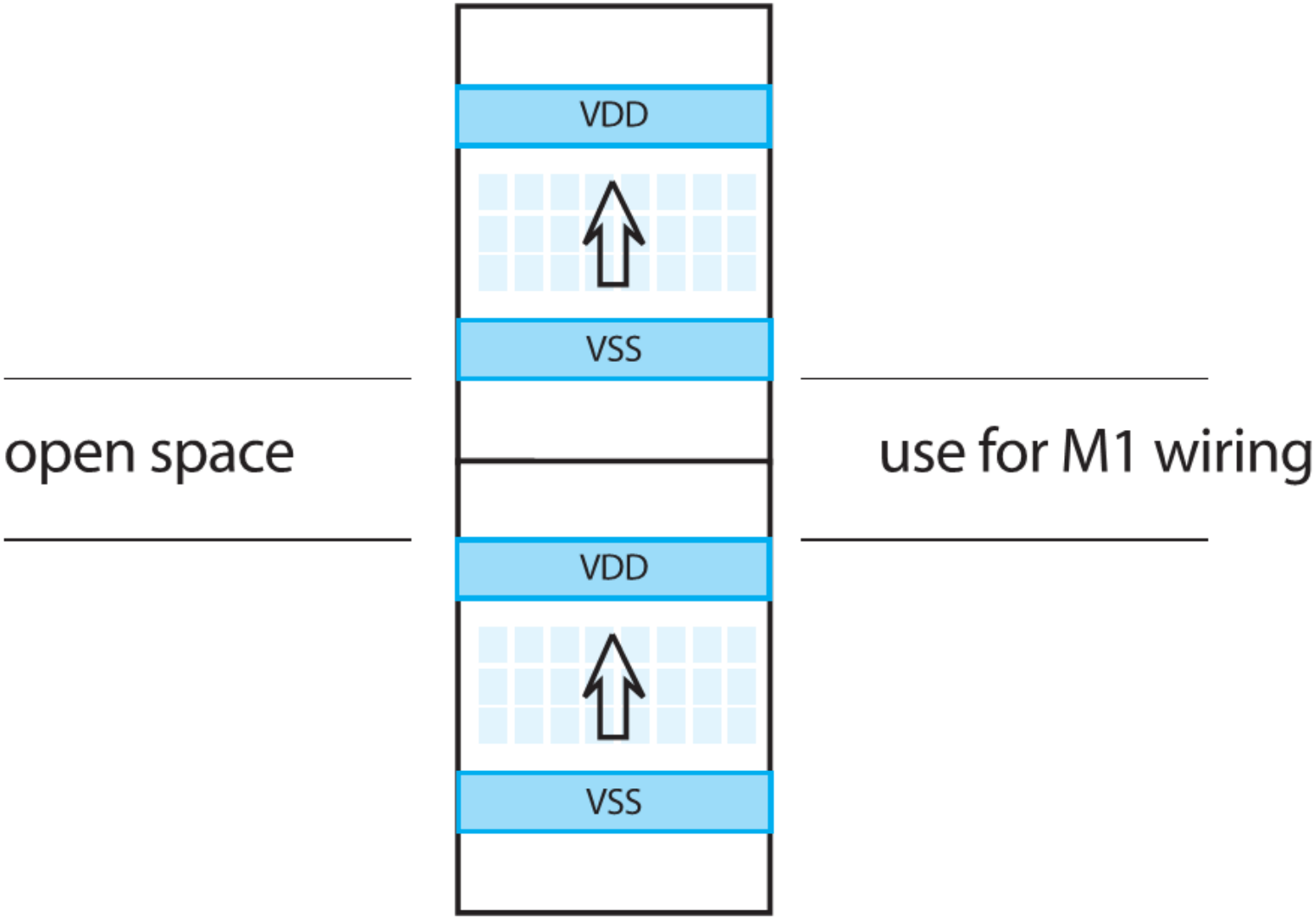
Routing Channel



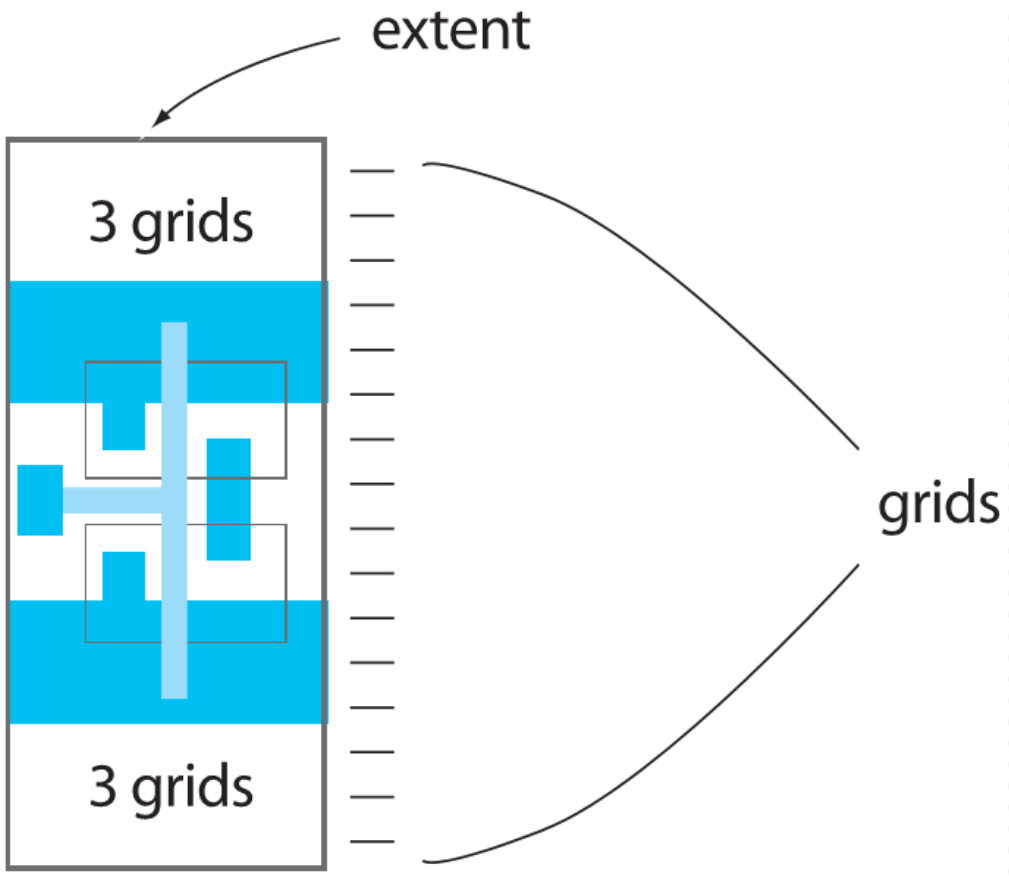


- Leave room at the extent for additional wiring.
- This arrangement requires no flipping of rows, and allows room to wire in lower metals.

Routing Channel



Routing Channel

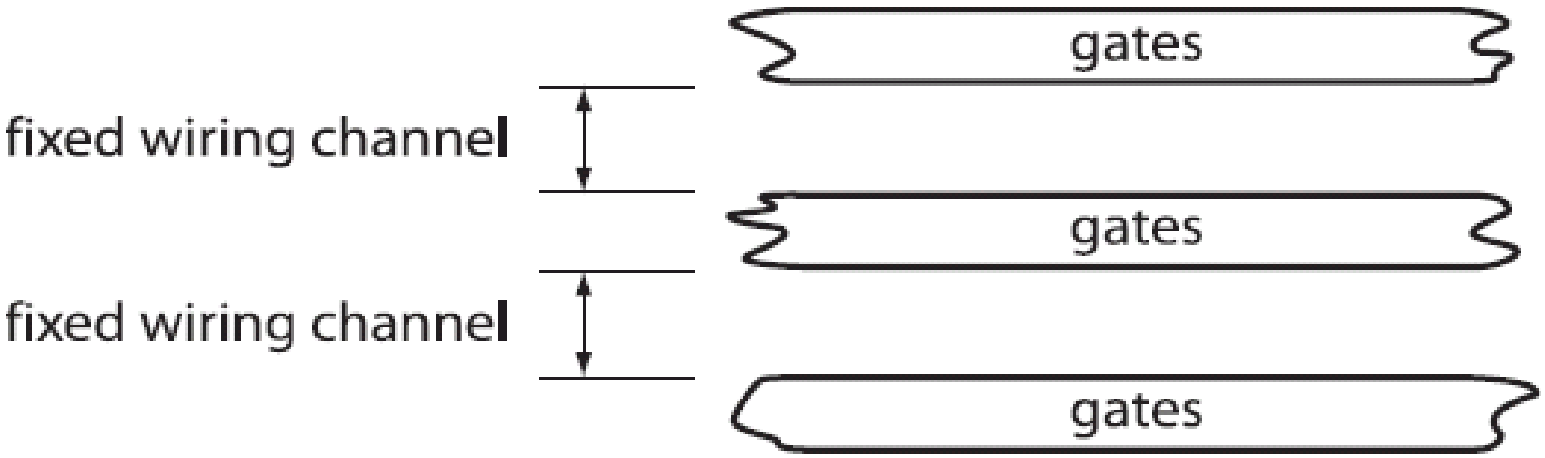


The routing channels, by the way, can be any height we want. We can make cells any dimensions to fit our needs.

Channel Routers

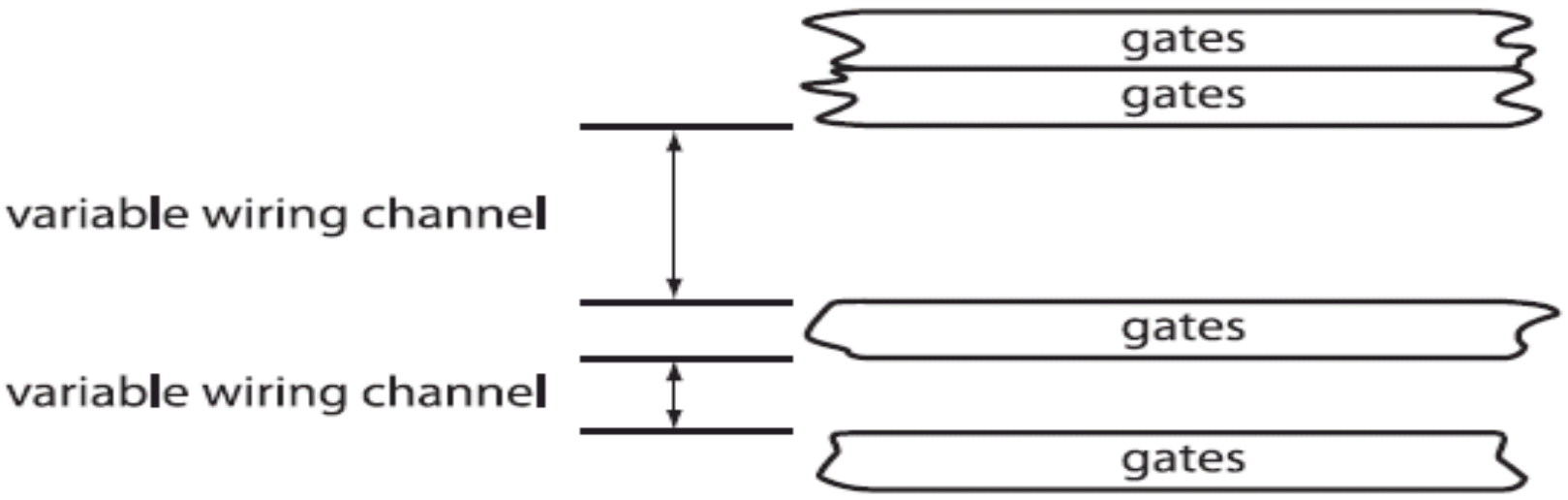
- Channel routers create channels between cells.
  - Architecture: A whole bunch of cell rows, then a big gap, then more cell rows.
  - Channel routers build in channels for wiring between rows of cells.
  - A **channel-based** approach leaves room for wires to be placed along designated channels.
  - Two architecture:
    - Fixed channel width
    - Variable channel width

Fixed Channel width



- The wires and their gaps are evenly spaced.
- Simpler to automate routing, but could be wasteful. Every channel does not necessarily need the same gap for wiring.

Variable Channel width

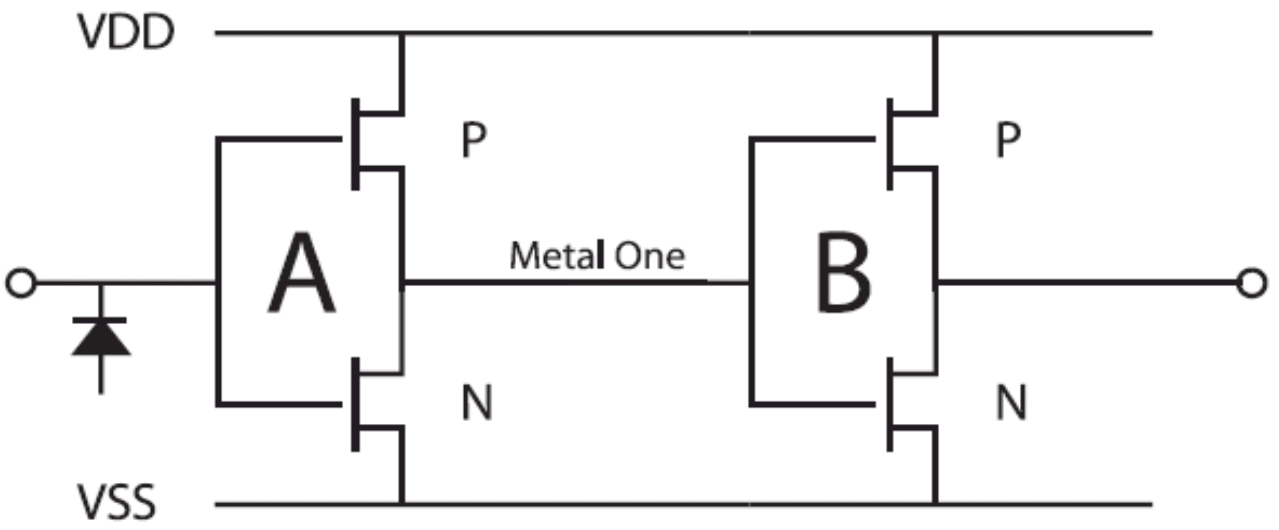


- Varying channel width according to need;
- Characteristic:
  - can give a much more compact die, a much smaller chip.
  - But, the software that drives the placement of cells has to be much more sophisticated, able to handle much more information. It announces higher demand to the Routing Tool;

Antenna Rule

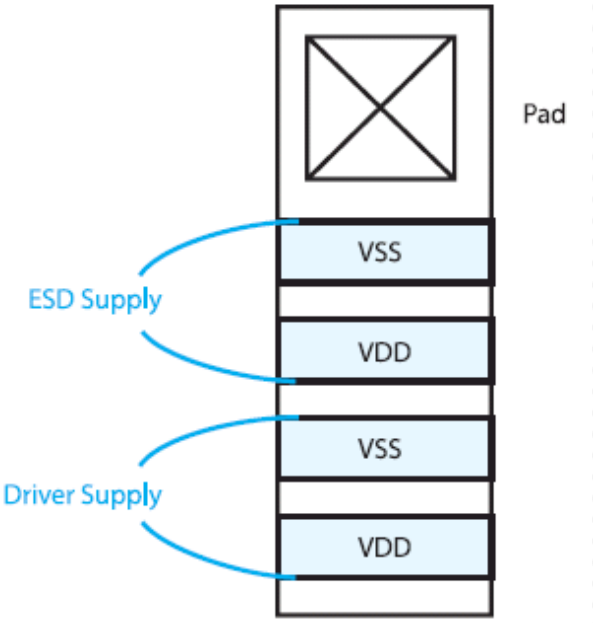
- is a design rule check that makes sure that any CMOS gate is tied to a diffusion before Metal One is processed.
- add a small reverse biased protection diode.
- Guarantee that any input will be tied down,
  - Usage: protected. Place these small protection diodes usually on the inputs to an FET gate.
- internal FET gates do not need any NAC
  - The output of a device can provide protection for the gate it is driving.
  - For instance: The internal FET gates of a big flip-flop are usually protected automatically by this connection. But for the inputs of an inverter, for instance, you have to build these protection diodes into your cells up front.

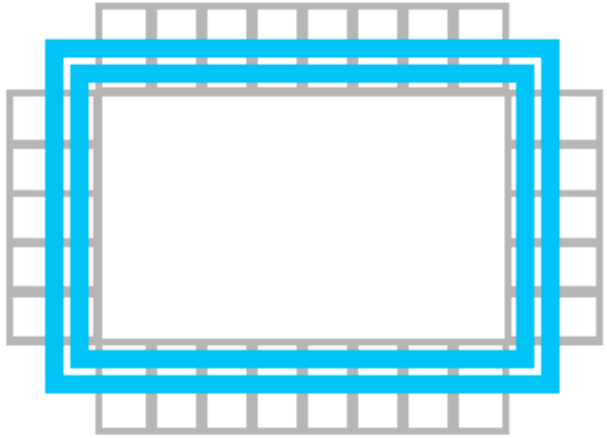
NAC diodes



- Have to build NAC diodes into all logic gates, particularly the inputs of standard cells.
- Cause: cannot guarantee standard cell will be driven directly from a diffusion in Metal One.
  - For instance: an input gate could be accessed in Metal Two.

Standardized Input and Output Cells





I/O pads surrounding the chip with their power rails.

- the cells that drive signals in and out of chips have to conform to grid-rules well.
- There are some rules, which are very similar to the rules for other components, mainly are used to protect proper alignment and design rule clearances.