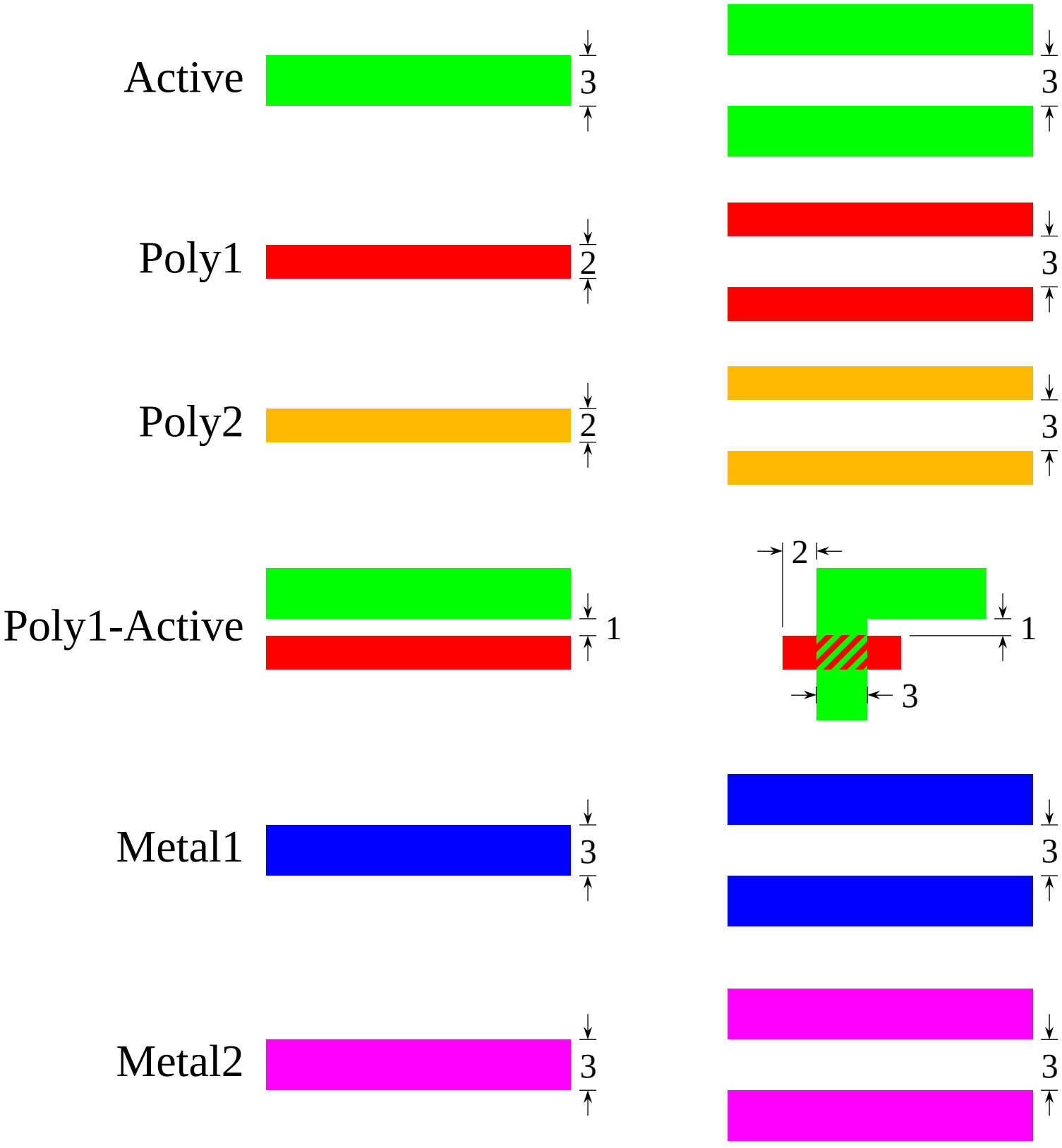
Layout for Digital Integrated Circuits

- Layout is the process of specifying the physical placement of and interconnections between all of the devices in a circuit.
- Layout is used to generate all of the mask layers used for chip fabrication.
- Layout for digital circuits:
  - Most transistors are minimum length
  - Transistor widths are sized to minimize delay

Scaleable CMOS Design Rules

- All device dimensions and placements are specified in terms of a scalable unit of distance called lambda (i.e., λ).
- λ is half of the minimum feature size specifyable in a given process (e.g., min. gate length).
- We will focus on a 0.5-μm *n*-well CMOS process with three metal layers. ($\lambda = 0.3\,\mu$m)

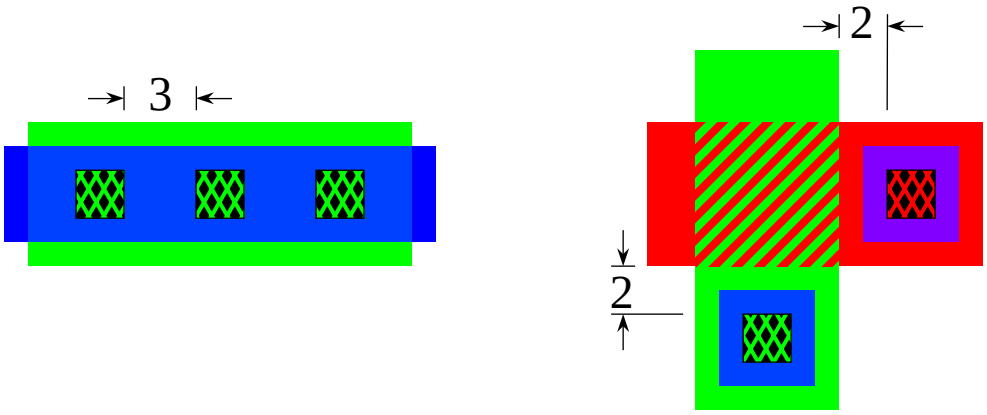
Common SCMOS Design Rules



Common SCMOS Design Rules

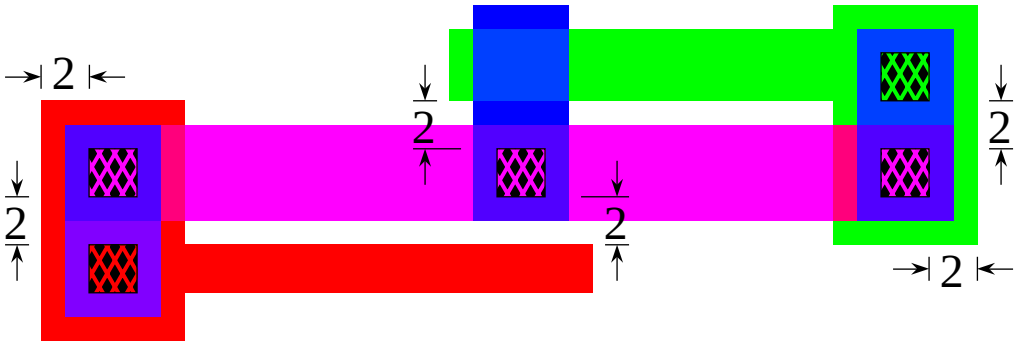
Contacts

Contact Spacing

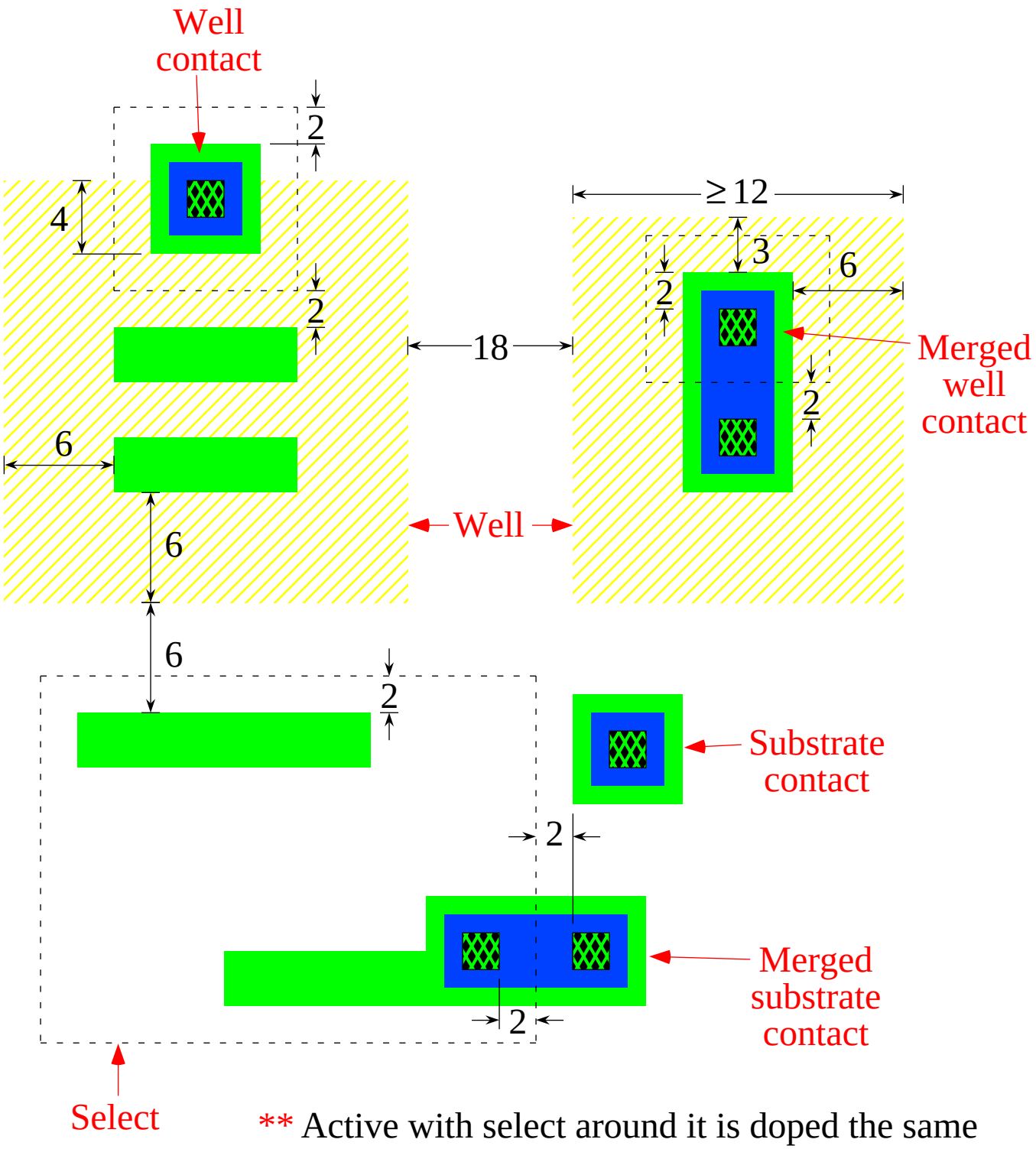


Vias

Via Spacing

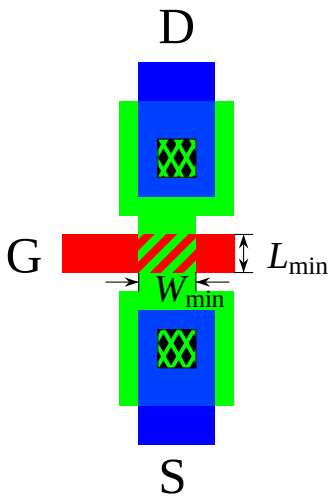


Common SCMOS Design Rules



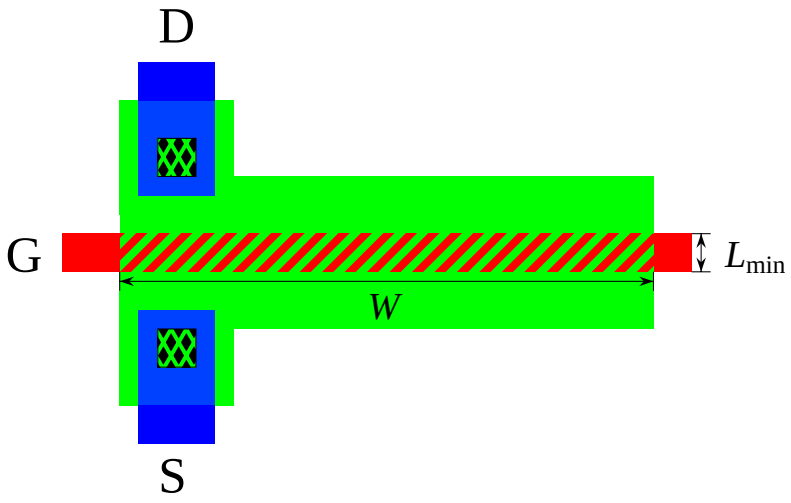
** Active with select around it is doped the same type as the well. Active without select is doped the same type as the substrate.

MOS Transistor Layout



Minimum size

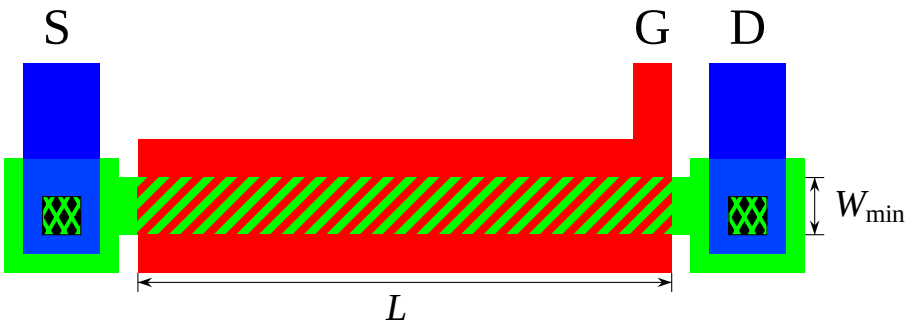
W_{min} and L_{min} set by design rules



Wide

Strong

Large W/L

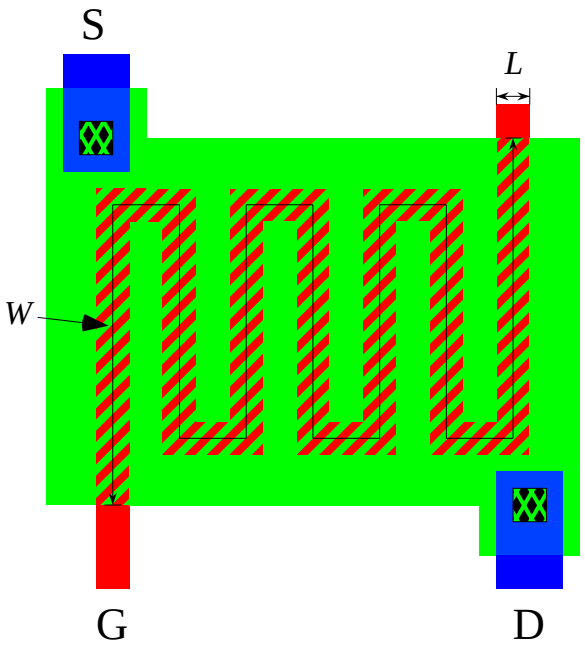


Long

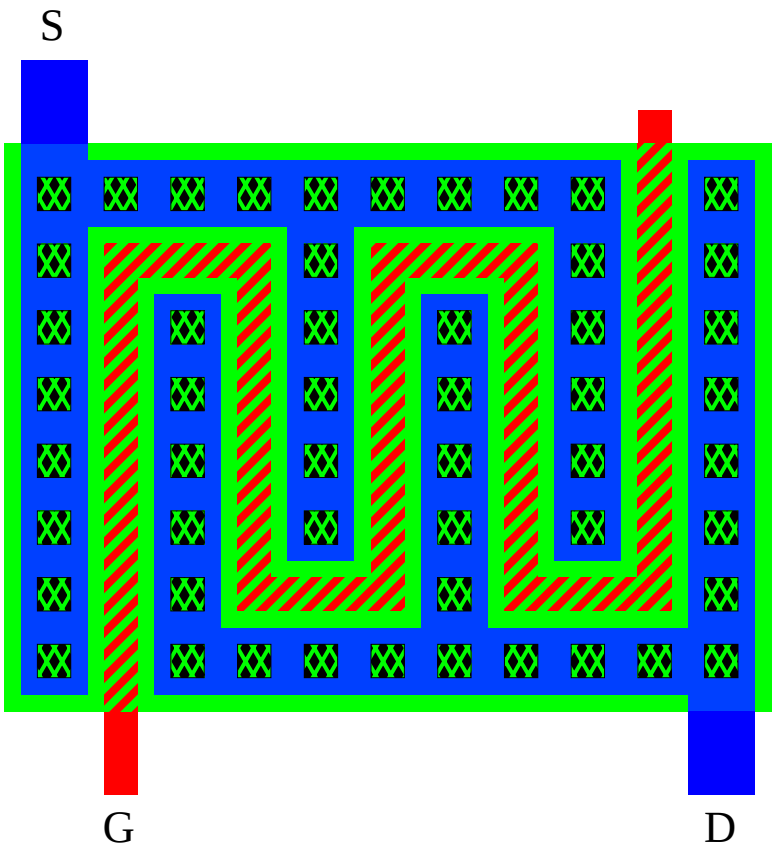
Weak

Small W/L

Very Wide MOS Transistors: Serpentine Transistors

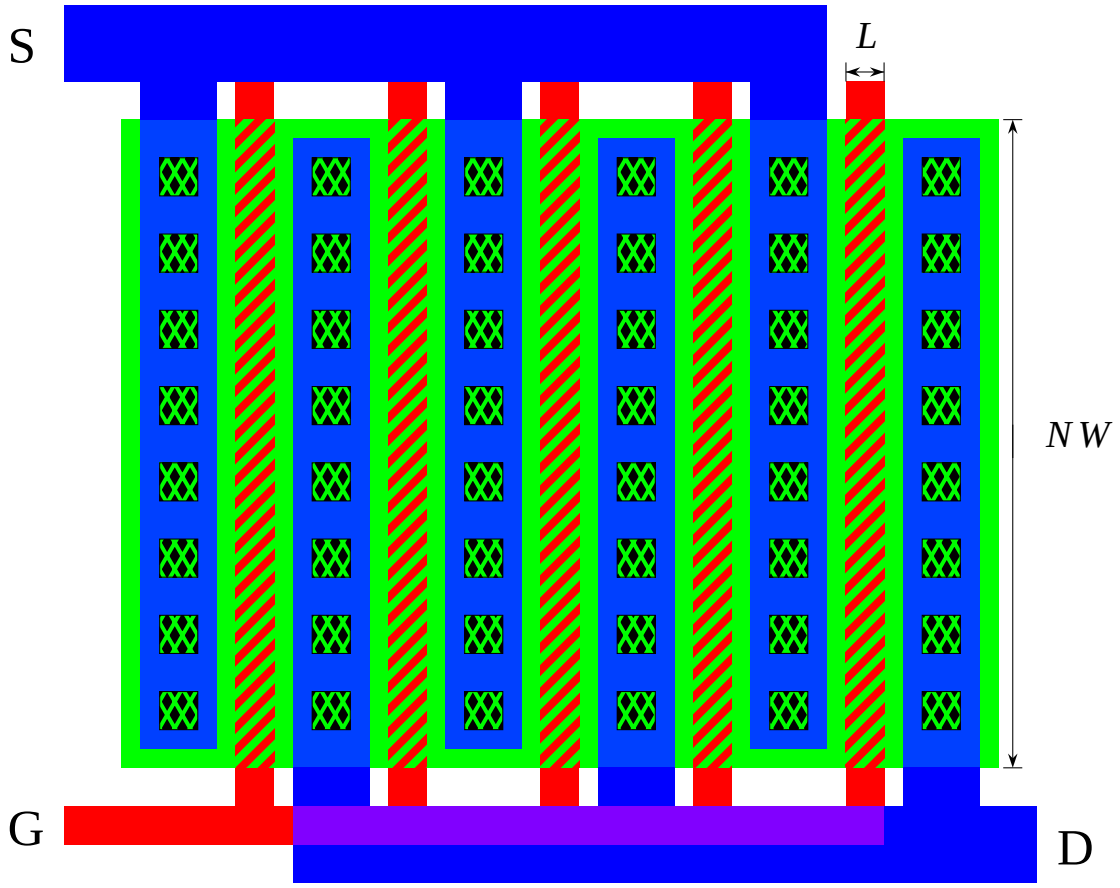


$W >> L$
Source and drain are interdigitated



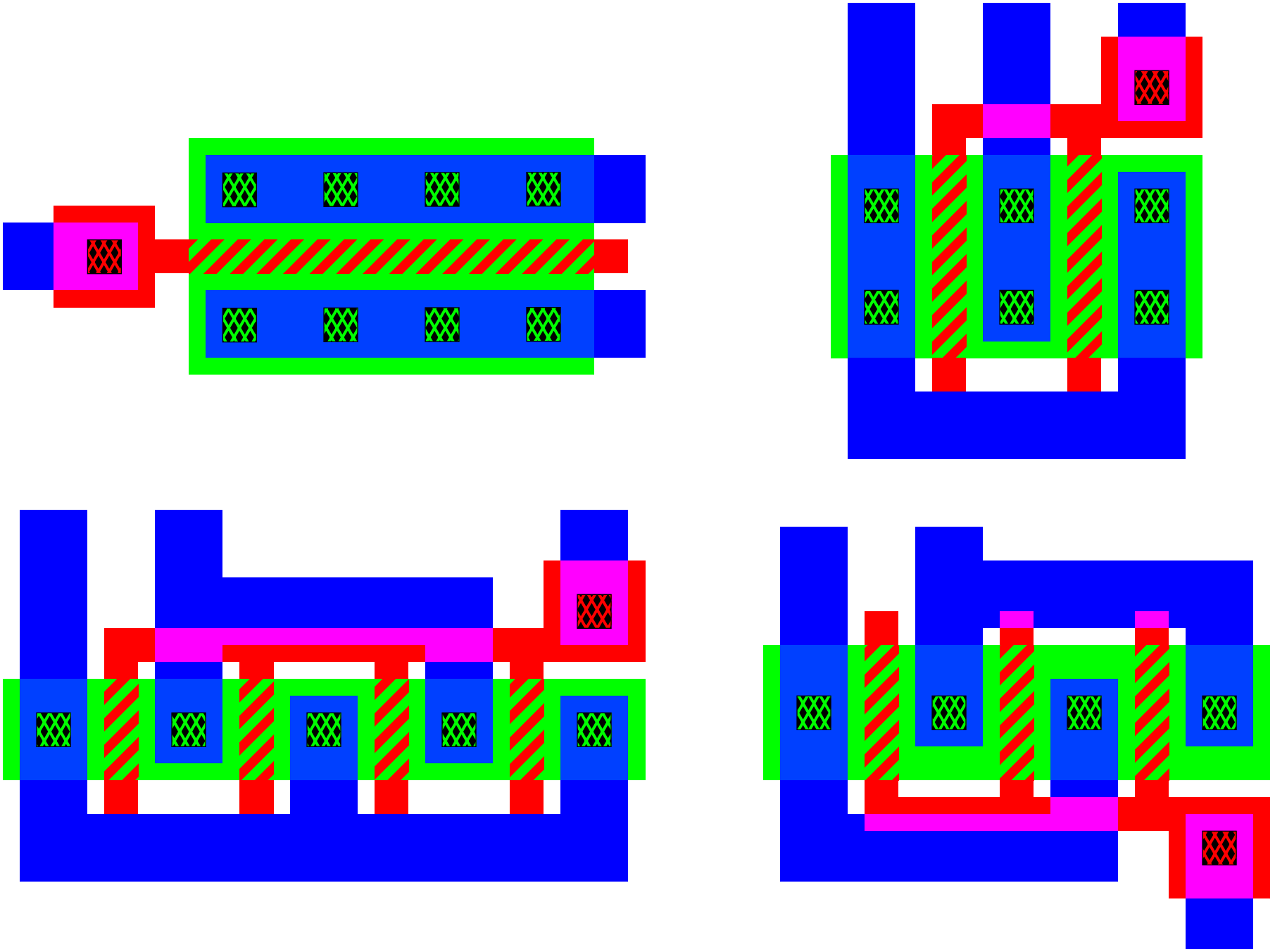
Minimize source and drain series resistance by using many contacts

Very Wide MOS Transistors: Stacked Transistors

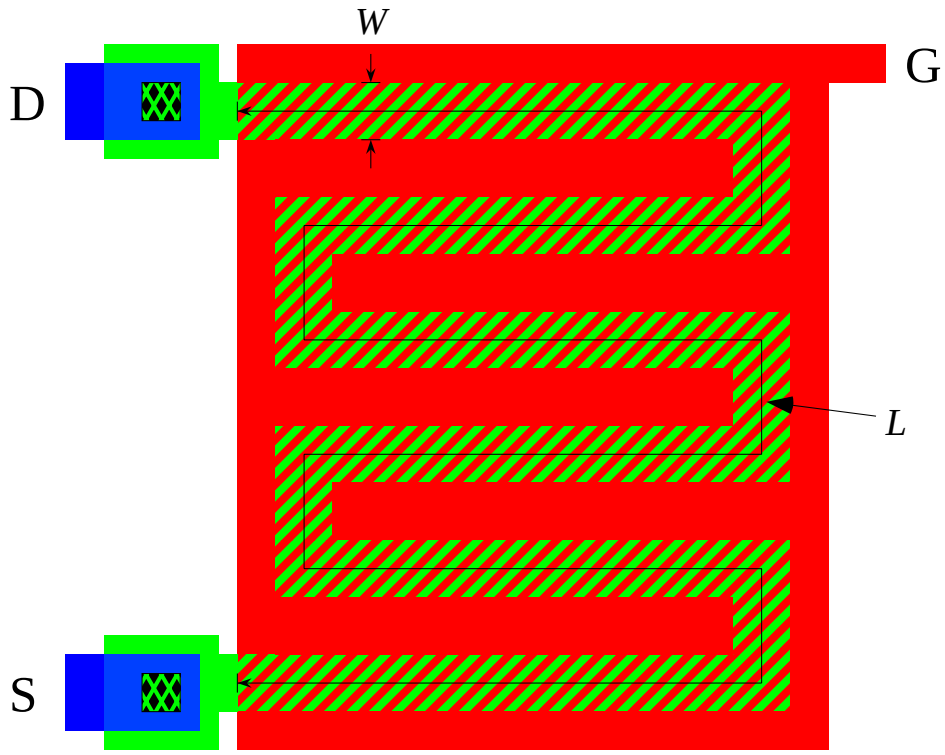


- Similar to serpentine transistors, but no bends.
- Source/drain regions shared $\Rightarrow$ reduced C_S and C_D.

Stacking Reduces Source/Drain Diffusion Area



Very Long MOS Transistors: Serpentine Transistors



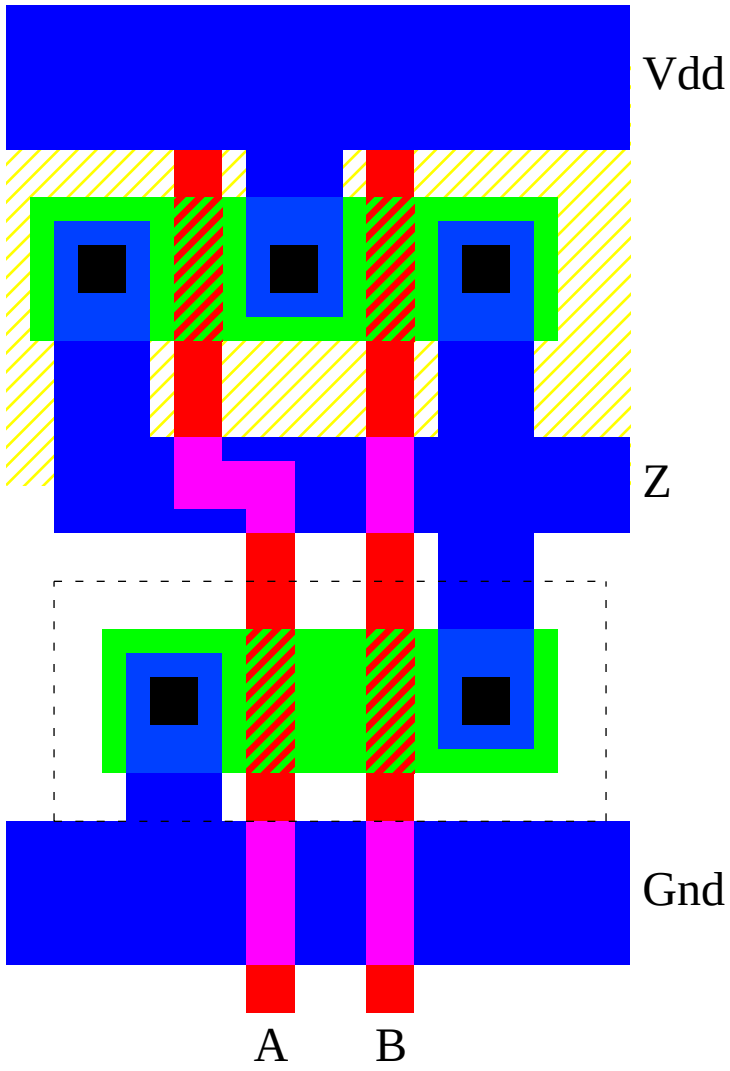
$W << L$

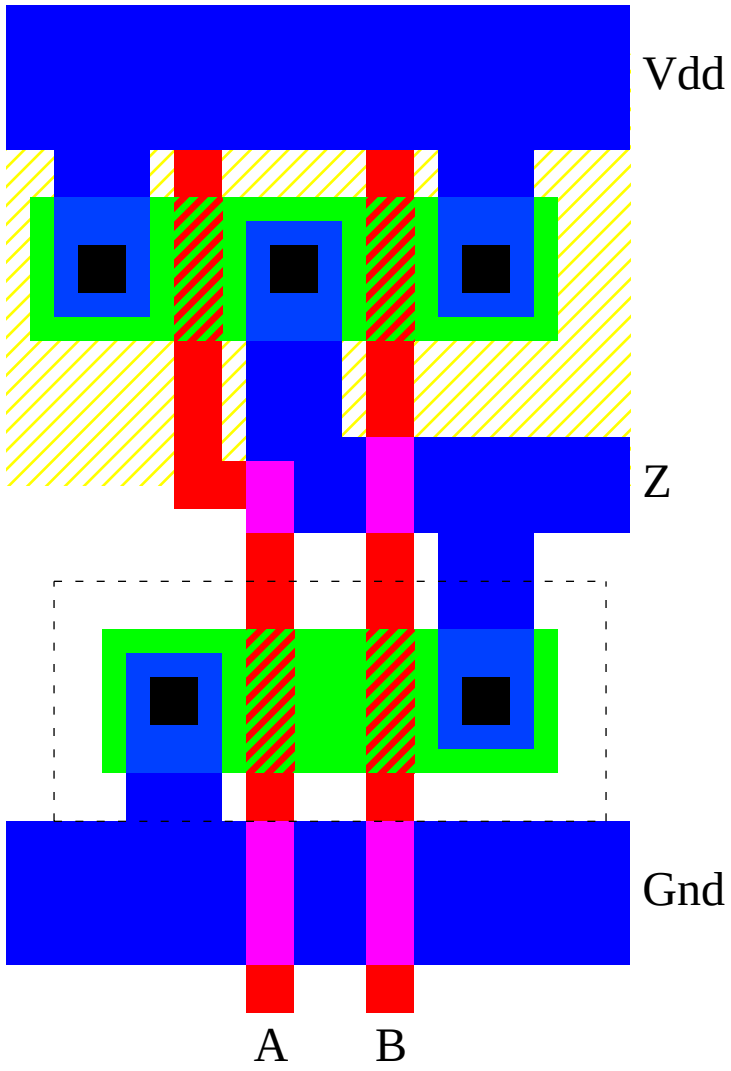
One large gate covers
the entire serpentine channel

Layout Guidelines for Complex CMOS Gates

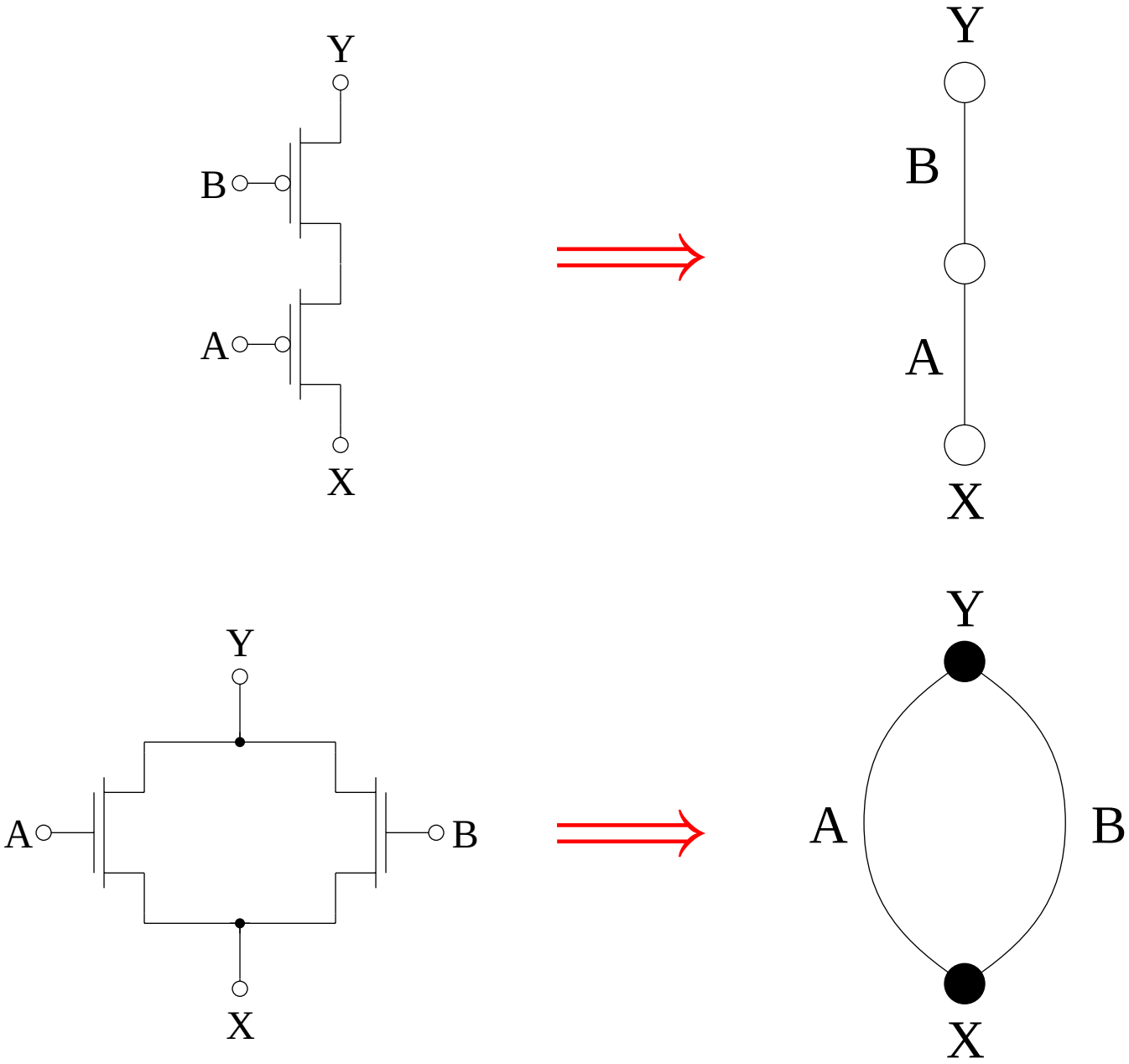
- Run power rails in metal parallel to strips of active.
- Run a strip of poly for each input orthogonal to the power rails.
- Order the input signals in order to maximize the number of shared source/drain diffusion regions and to minimize the number of broken poly lines.
- Finish connections in metal or poly (only use poly for short runs).
- Minimize internal node parasitics (both *C* and *R*) by
  - Keeping runs of active as short as possible
  - Sharing as many source/drain diffusions as possible
- Avoid using higher level metal layers for local interconnects.
- Consistently run horizontal connections in one metal layer (e.g., metal1) and vertical ones in another (e.g., metal2).

Which NAND Gate Layout is Better?





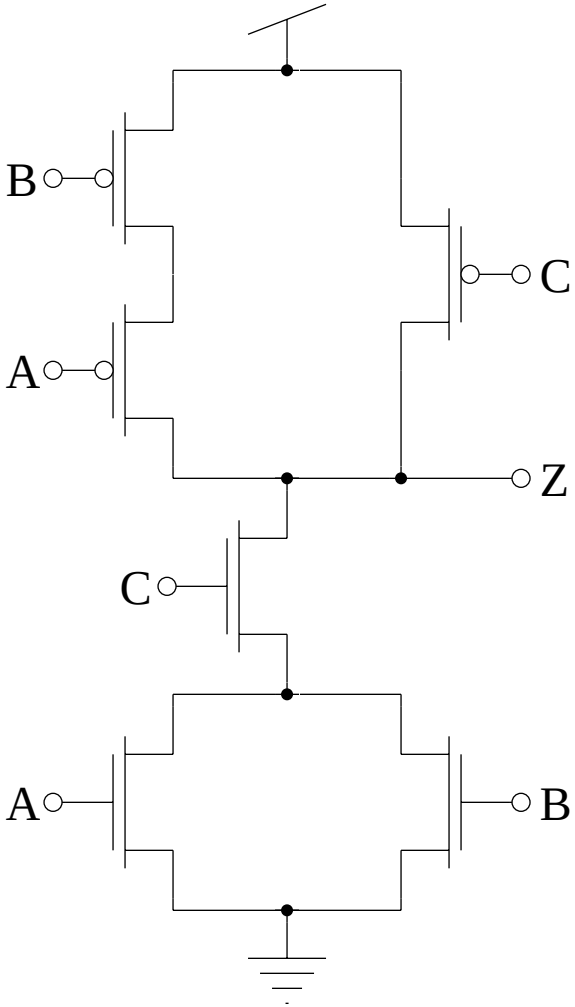
Logic Graphs for PDNs and PUNs

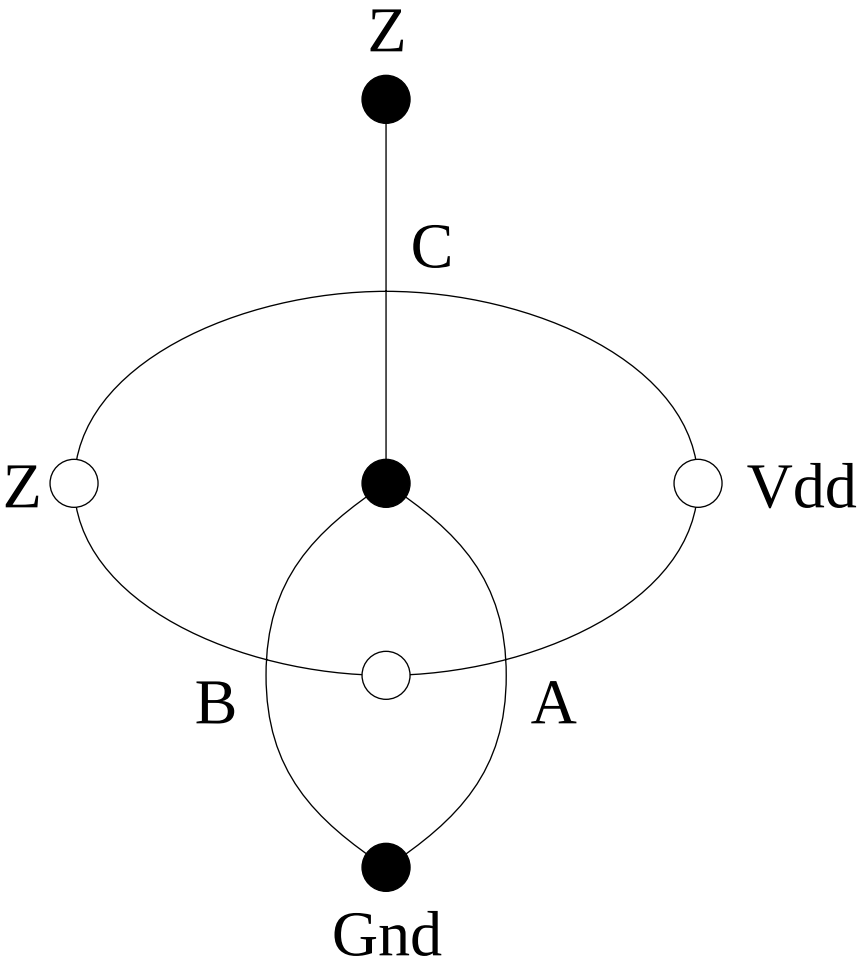


Euler Paths Through Logic Graphs

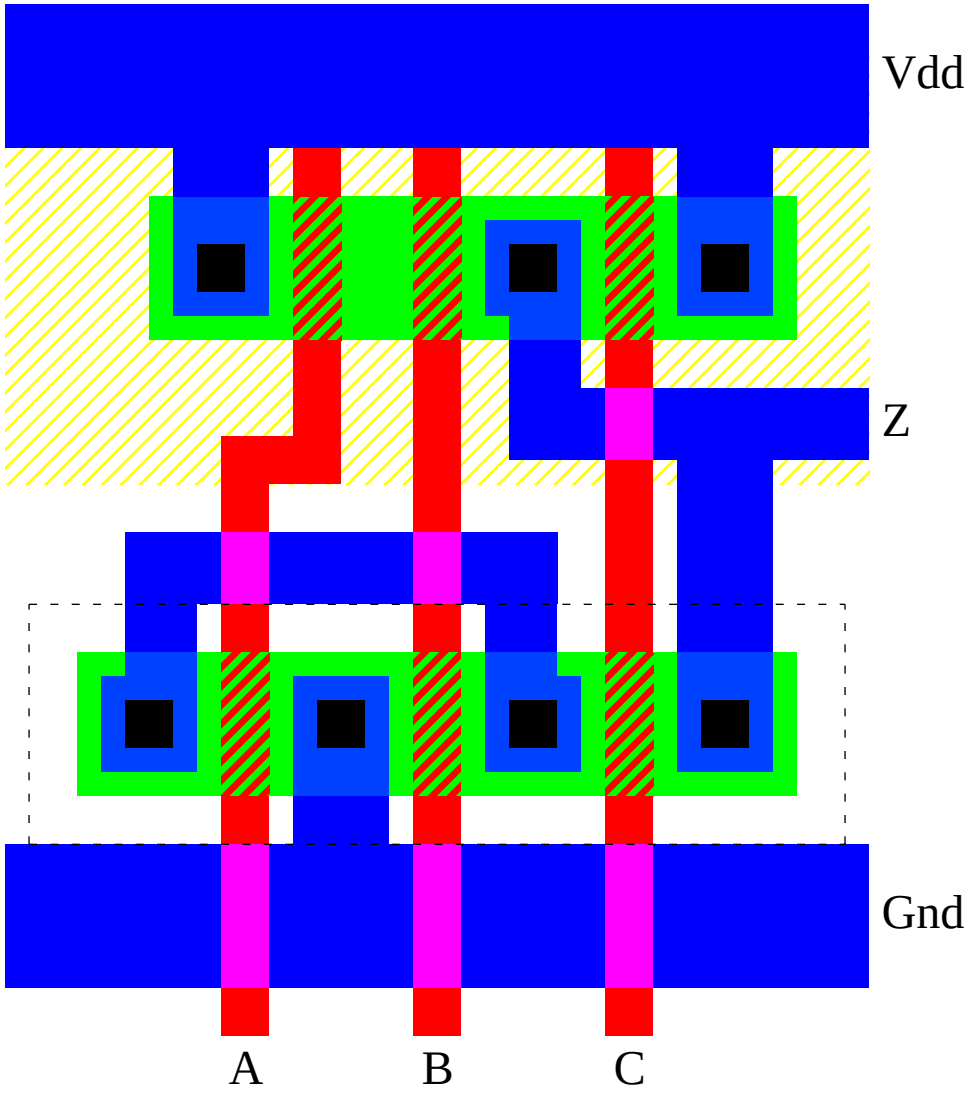
- Euler path: A path through all nodes in a graph such that each edge is traversed exactly once.
- If we can find *consistent* Euler paths through the PUN logic graph and the PDN logic graph, then we can order the inputs so that the poly strips are continuous and source/drain diffusions are shared maximally.
- The order of the inputs is the same as the order of the edges in our Euler paths.
- Some Euler path caveats:
  - Euler paths are not unique
  - Not all Boolean expressions have consistent Euler paths
  - One form of a Boolean expression may have consistent Euler paths while an equivalent form may not.

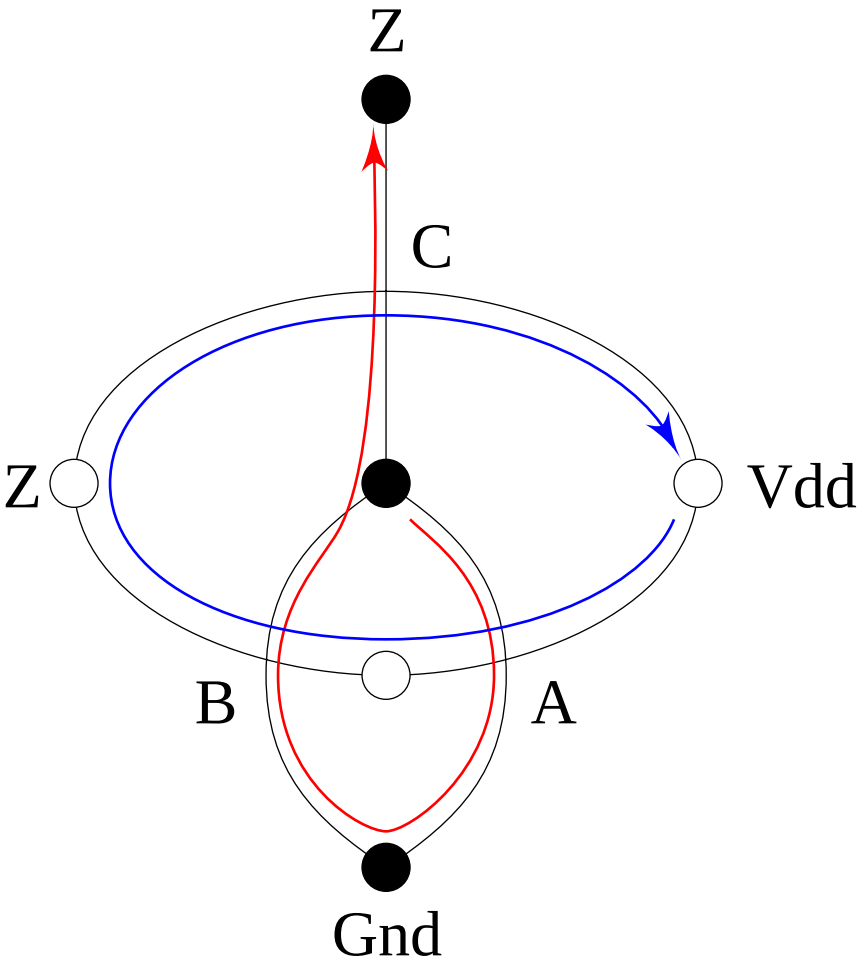
Euler Path Example: $Z = \overline{A}\,\overline{B} + \overline{C}$



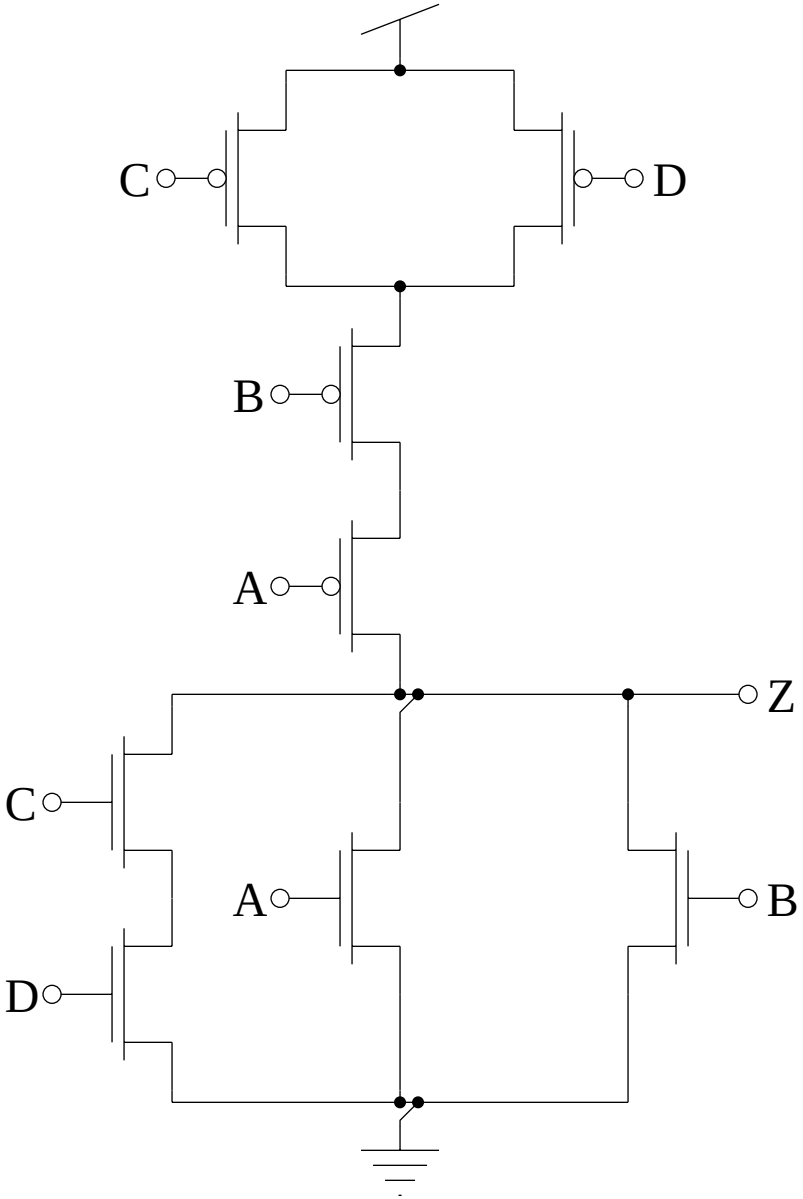


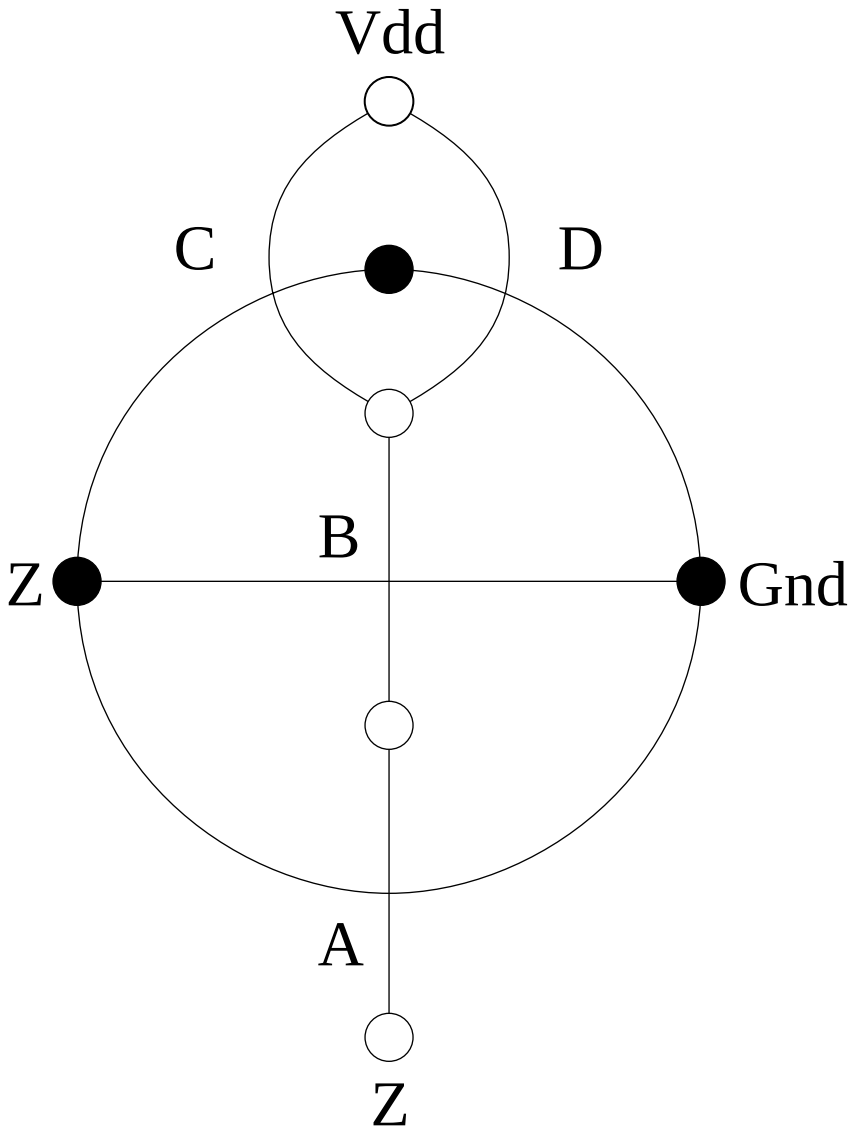
Euler Path Example: $Z = \overline{A}\,\overline{B} + \overline{C}$



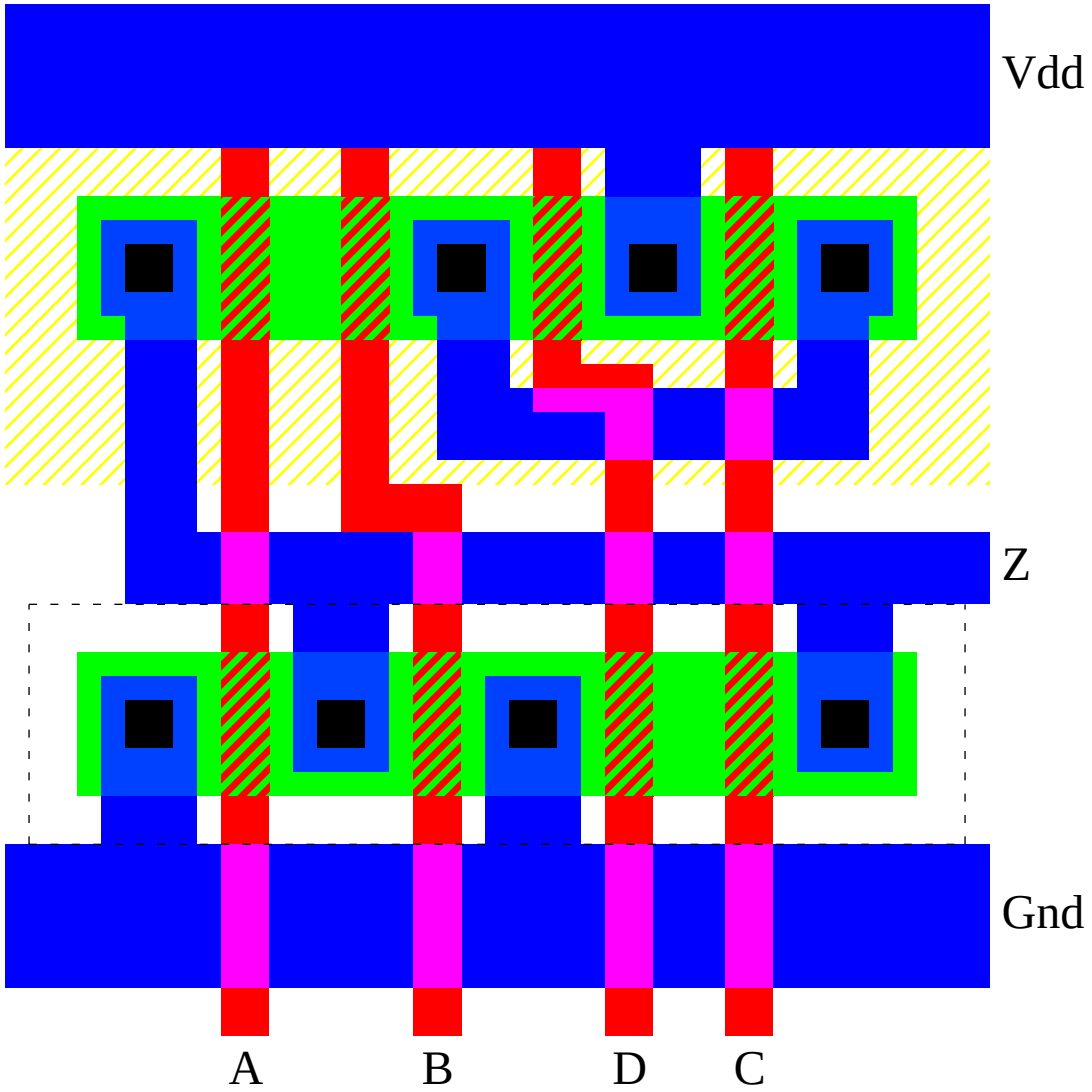


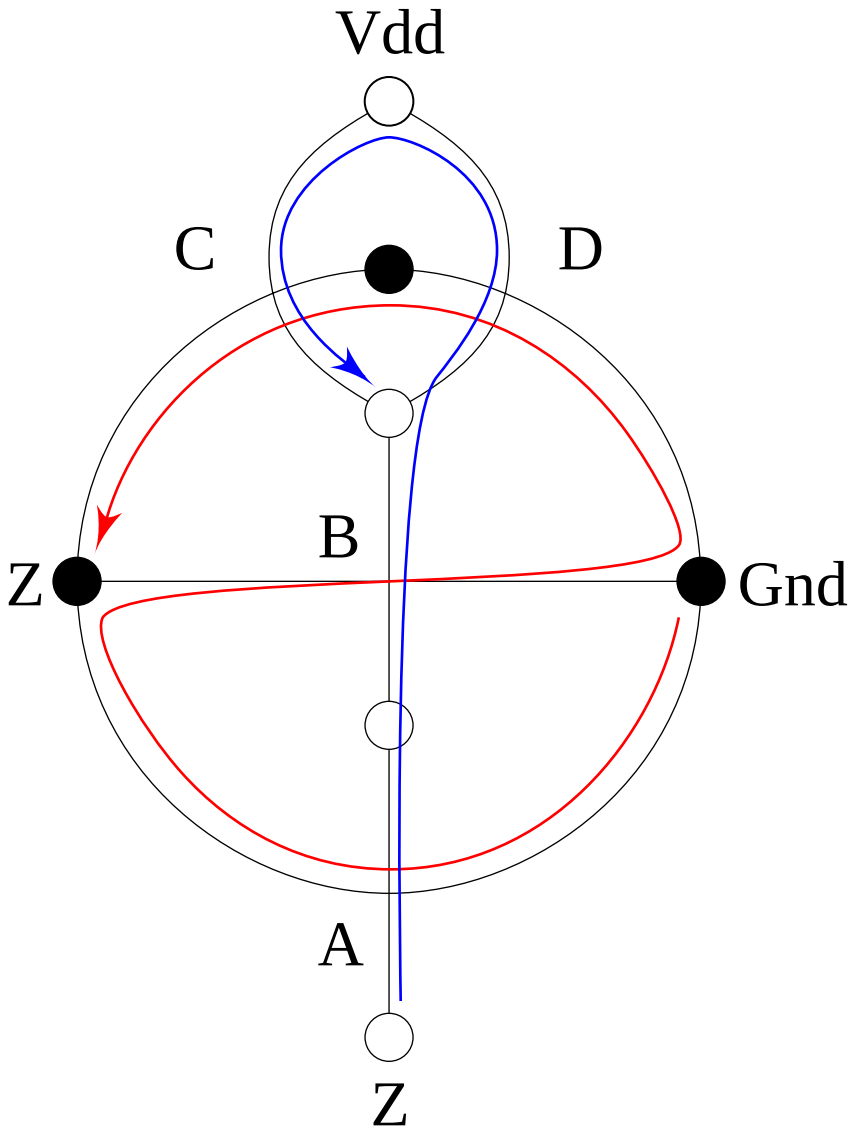
Euler Path Example: $Z = \overline{A}\,\overline{B}\,(\overline{C} + \overline{D})$



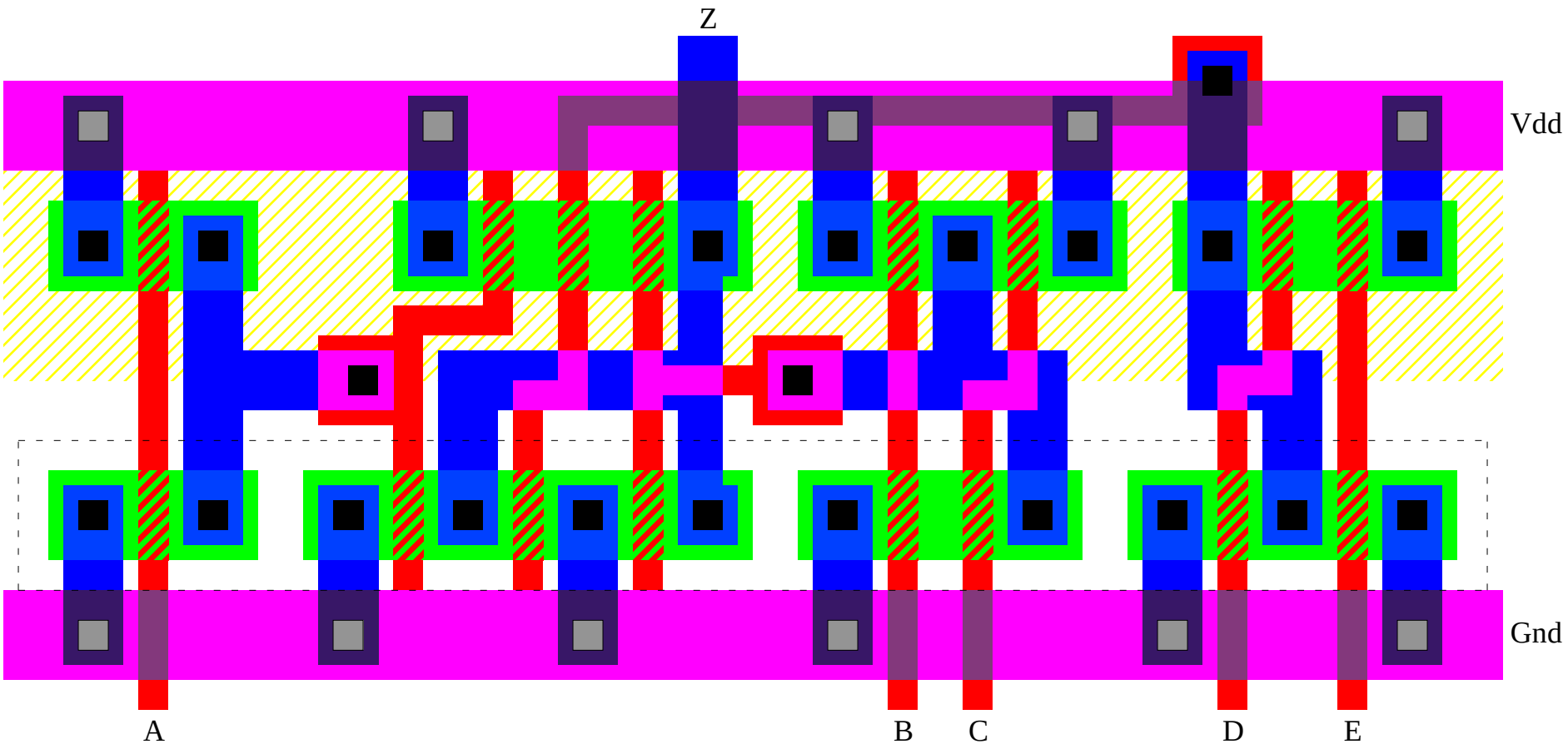


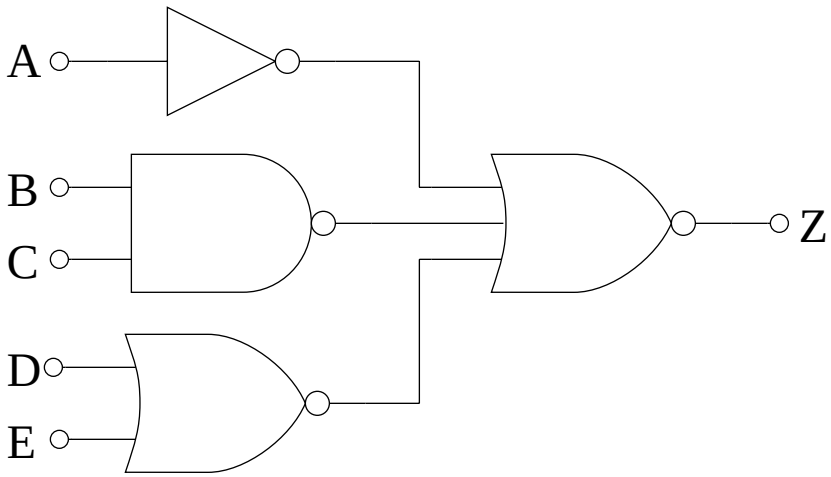
Euler Path Example: $Z = \overline{A}\,\overline{B}\,(\overline{C} + \overline{D})$



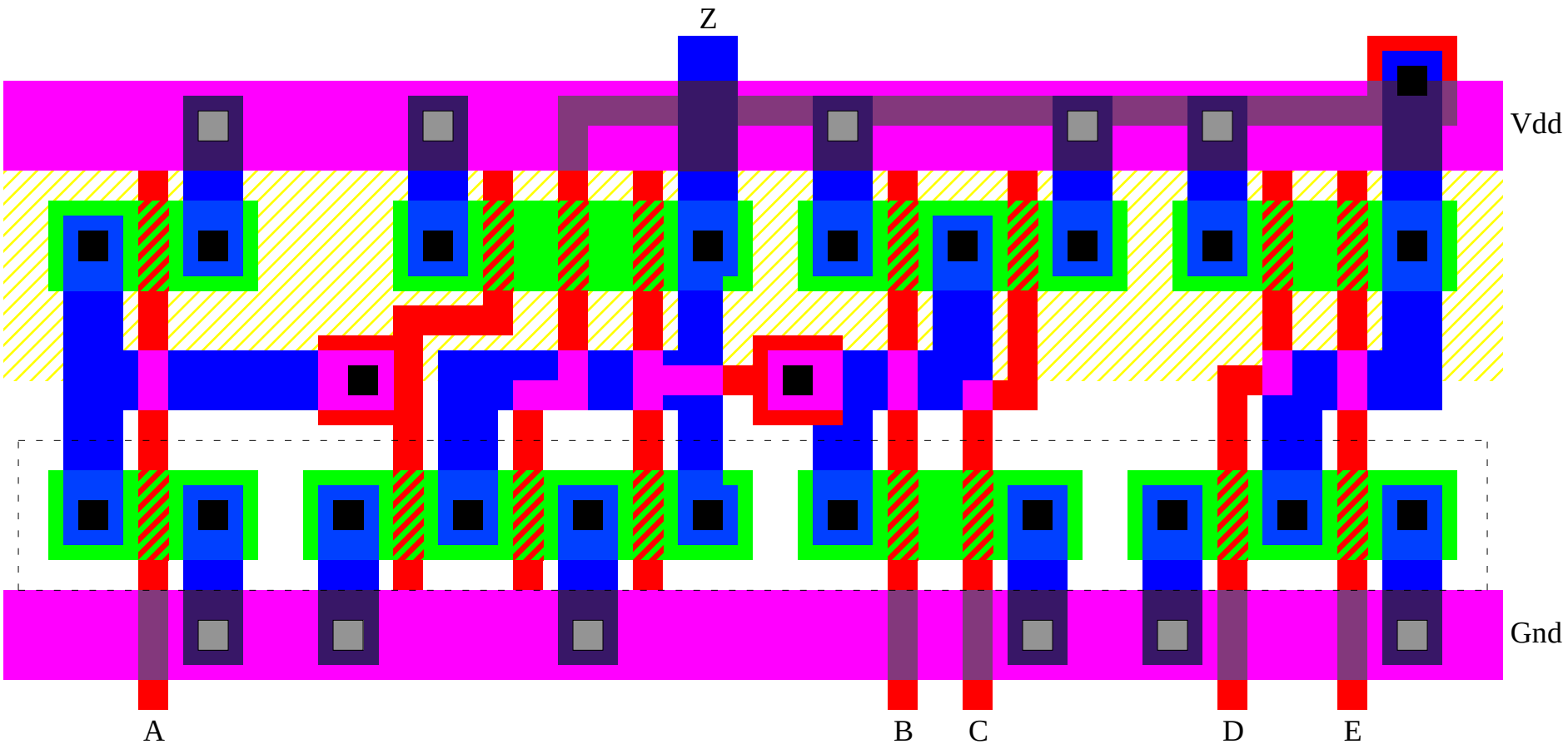


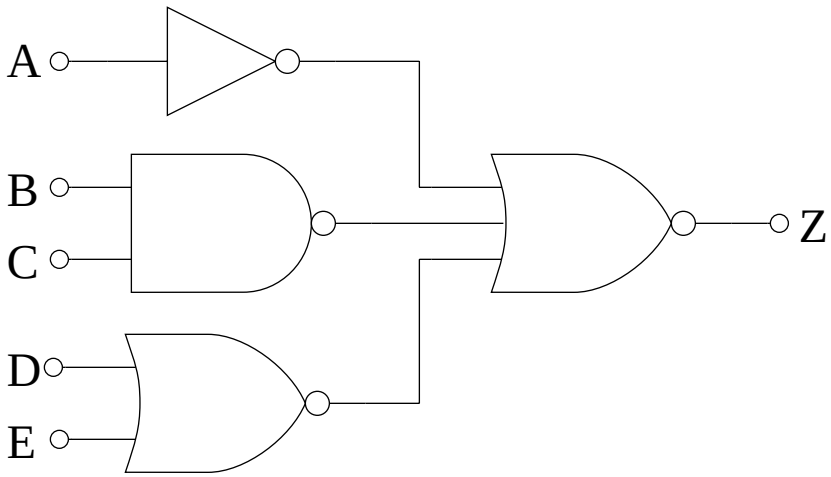
Compacting Layout of CMOS Gates





Compacting Layout of CMOS Gates





Compacting Layout of CMOS Gates