VLSI Circuit Optimization

Thomas Jagielski and Navi Boyalakuntla

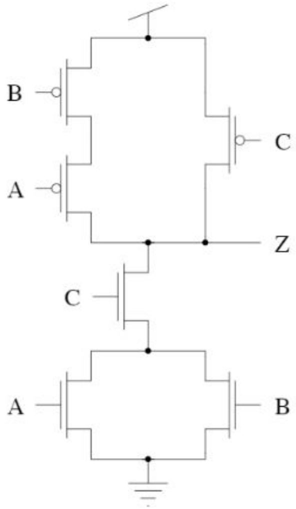
Resources Used

- Brad Minch Discrete Math Presentation
  - Unless otherwise specified, all visuals from this source
- Static CMOS Design Presentation - ZALAM
- Optimum Gate Ordering of CMOS Logic Gates Using Euler Path Approach Some Insights and Explanations - Kuntal Roy

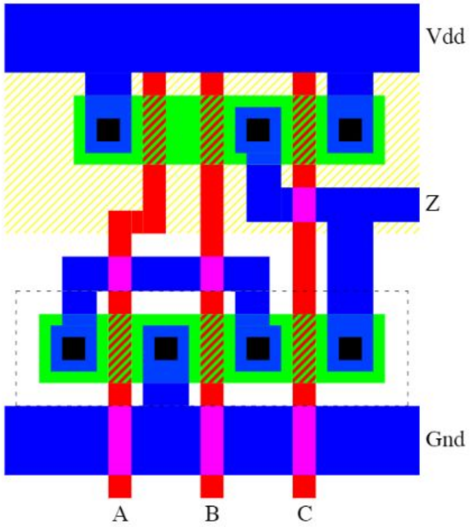
These resources and our paper can be found at the following link:

https://tinyurl.com/tgawukd

Overview/Goal



Schematic of
Z = A’B’+C’



Layout

Looking at logic graphs created from the schematic, we can order transistors in an optimized way.

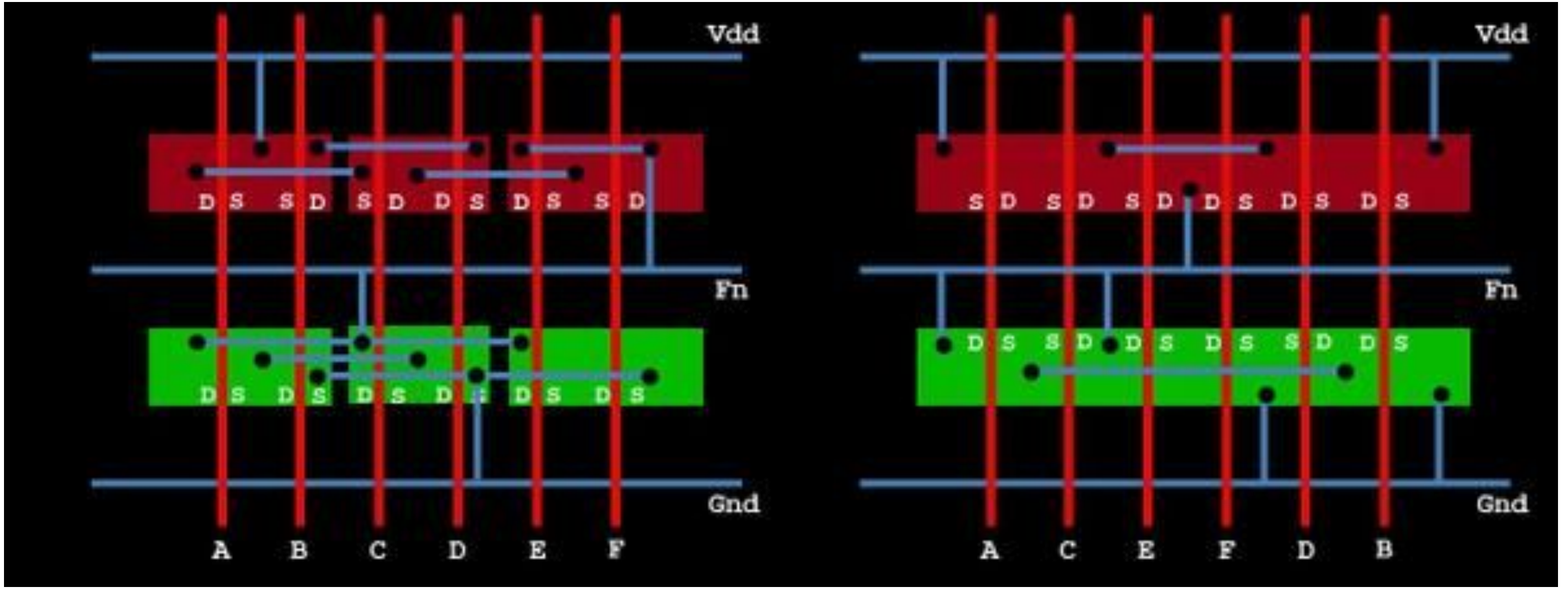
Why Do We Want to Optimize the Layout?

- Smaller package size
  - Less chip area
  - More circuits on a single silicon wafer
- Less chemicals being used
- Optimally shared diffusion strip
  - As the number of diffusion breaks being used increases, so does the layout area

Note: Diffusion strip refers to the doped semiconductive material (often silicon in CMOS applications). This is what creates a transistor.

Optimized vs Not Optimized

From: https://www.vlsisystemdesign.com/art-of-layout-eulers-path-and-stick-diagram-part-3/

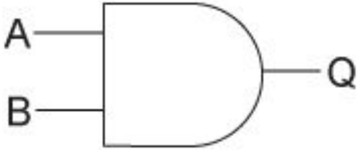


Not Optimized

Optimized

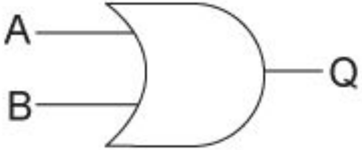
Logic Gates

And Gate



If A **AND** B are true,
then Q is true.

Or Gate

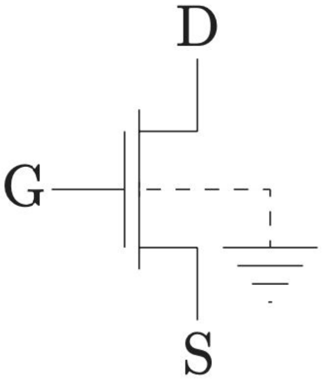


If A **OR** B are true,
then Q is true.

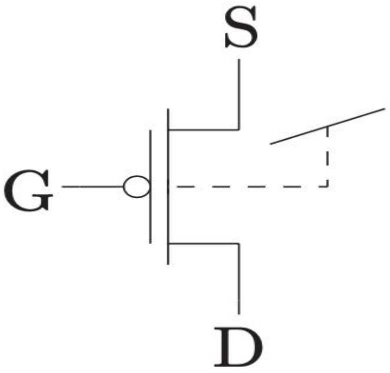
MOS Transistors

Transistor - Semiconductor based device which (for our application) is being used as an electrical signal based switch

NMOS Transistor

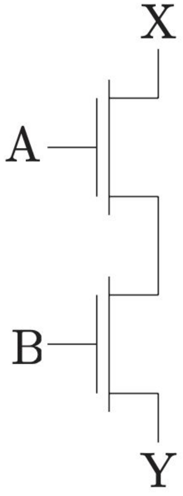


PMOS Transistor

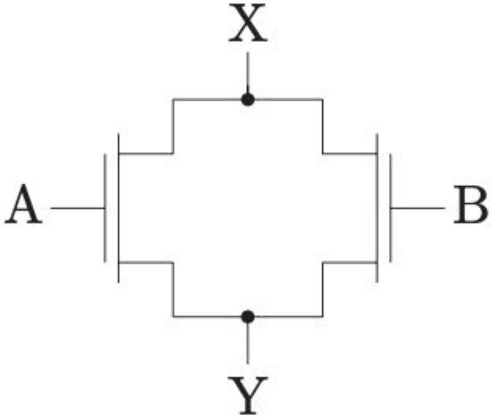


Representing Logic with Transistors

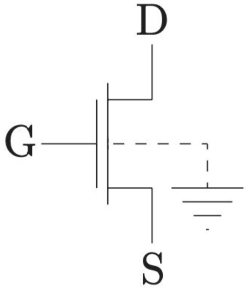
Transistor AND gate



Transistor OR gate

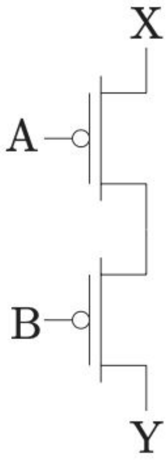


Remember: NMOS transistor

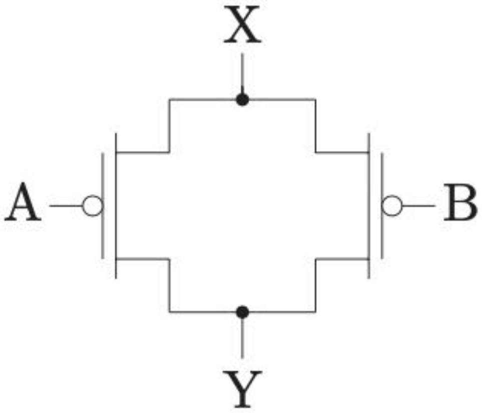


Representing Logic with Transistors

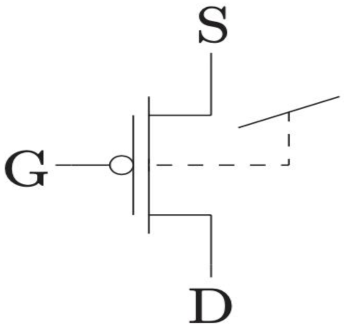
Transistor NAND gate



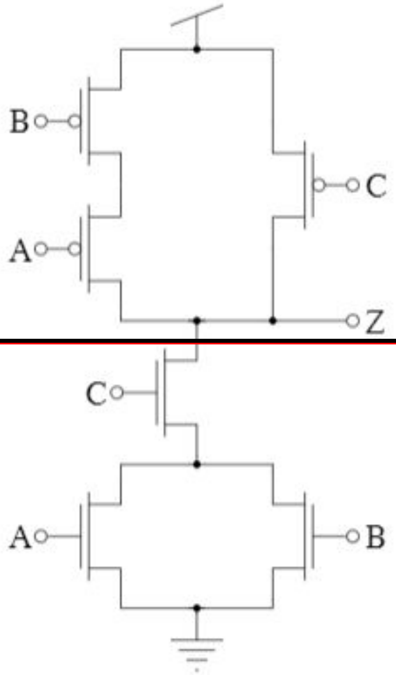
Transistor NOR gate



Remember: PMOS transistor



CMOS Gate Structure

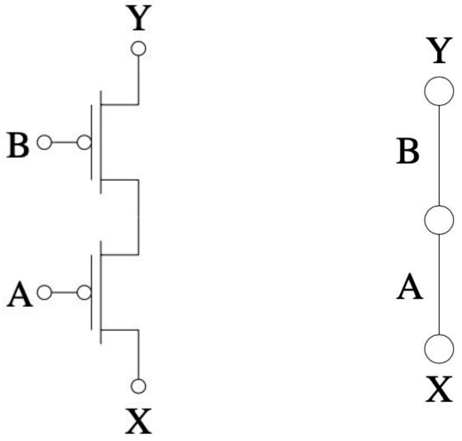


Pull Up Network - transistors allow current to flow from Vdd across them when the input signals (A,B,C) are LOW.

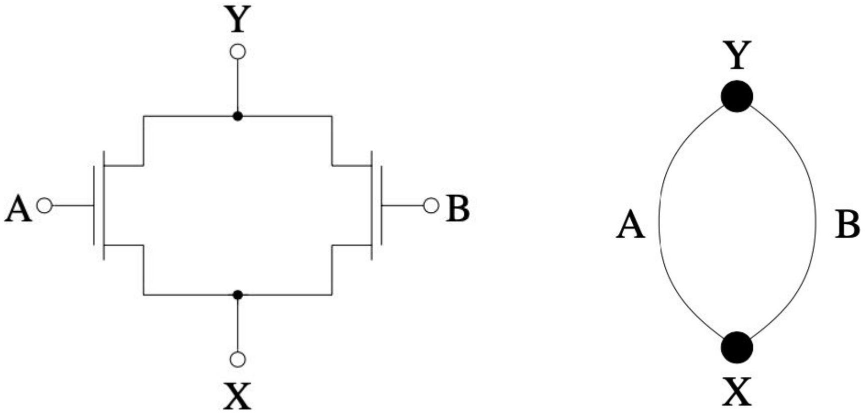
Pull Down Network - transistors are connected to GND when the input signals (A,B,C) are HIGH. This is Boolean complement to ensure that when the Pull-Up Network yields true, the Pull-Down Network yields false.

Circuits to Logic Graphs

Pull-Up Network Logic Graph (PUN)



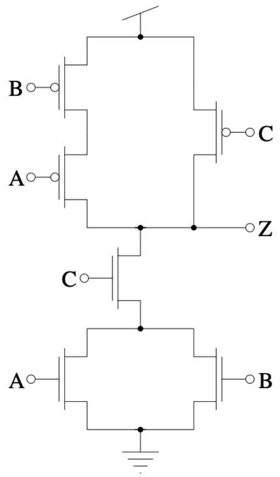
Pull-Down Network Logic Graph (PDN)



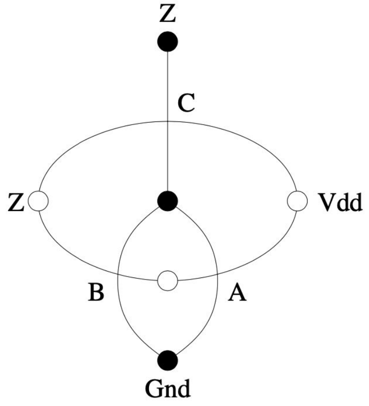
Transistors are edges and nodes (Vcc, Gnd, and output) are the vertices

Bringing it Together

Schematic of Z = A'B'+C'

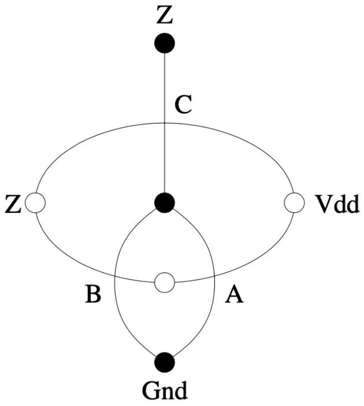


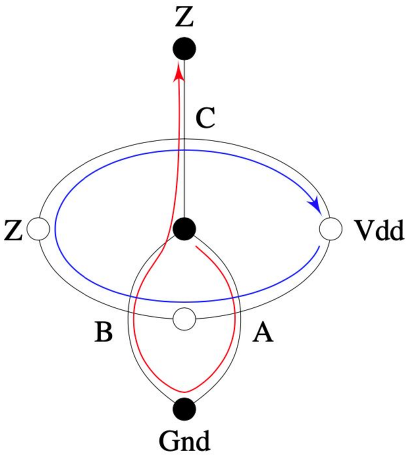
Combined Logic Graph - Edges are the same on both PUN and PDN graphs



Consistent Euler Path

An Euler path is consistent if it goes through the PUN and PDN with the same ordering of transistors (edges)

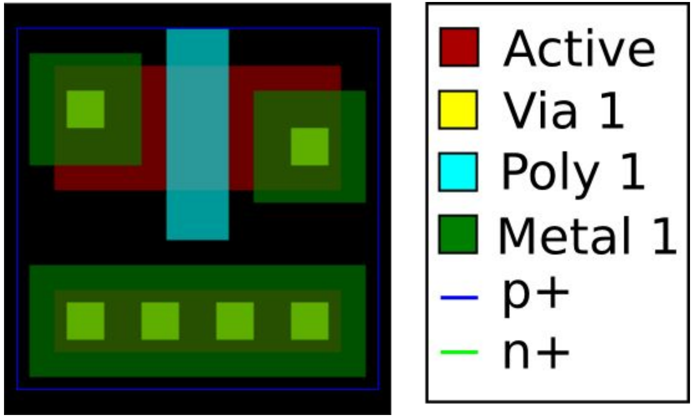




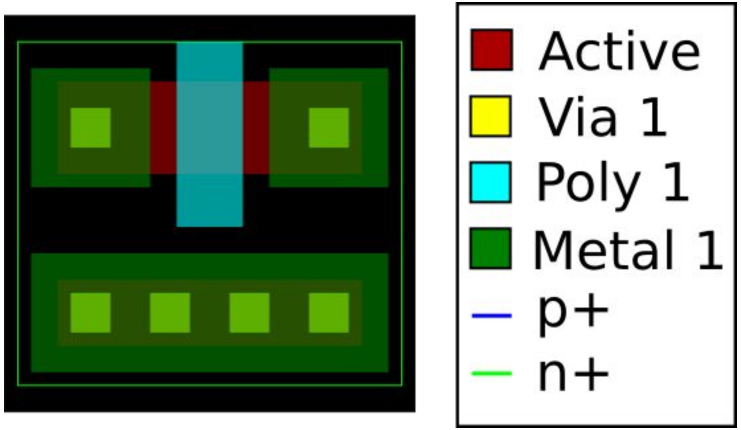
CMOS Transistor

Source:
http://www.globaltcad.com/en/showcase/example-gallery/cmos-logic/02-single-mos-transistor.html

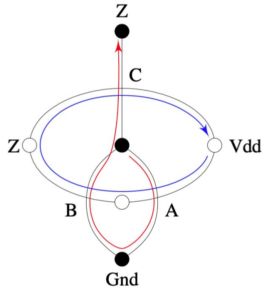
PMOS Transistor



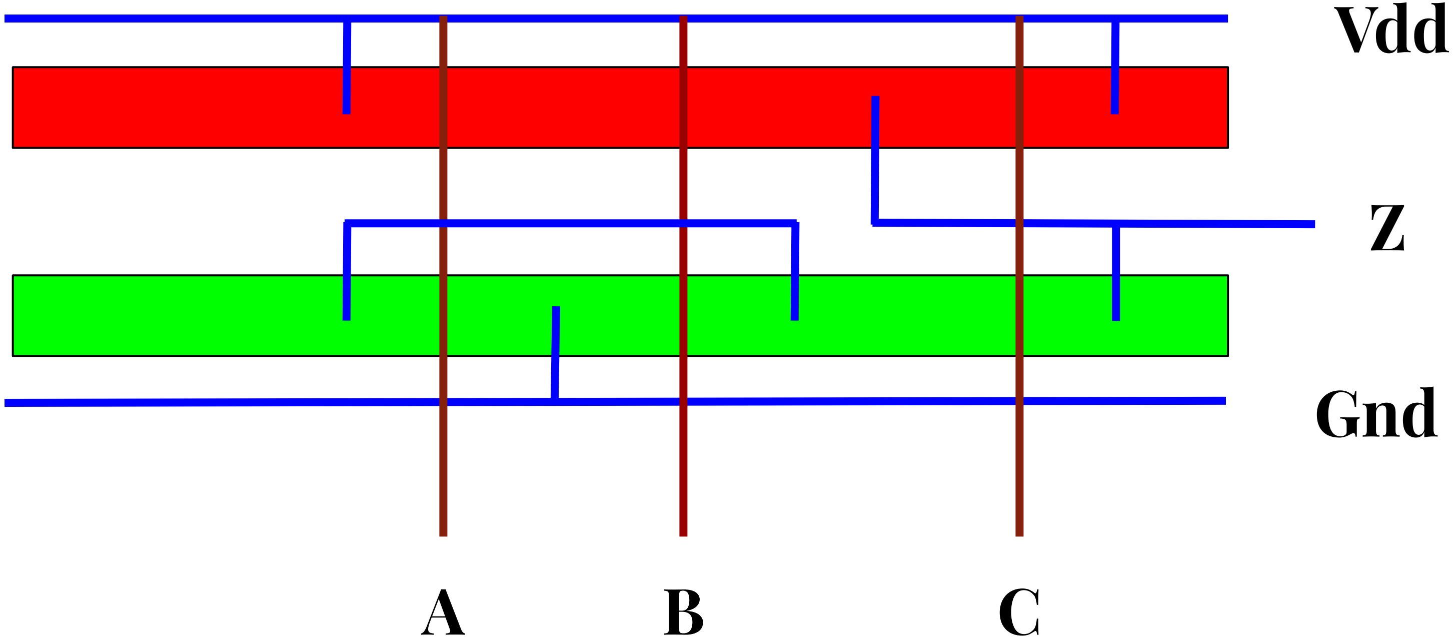
NMOS Transistor



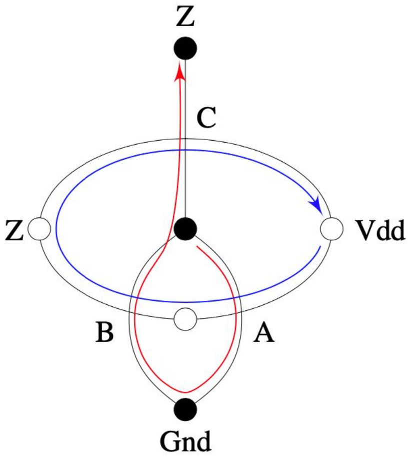
Stick Diagram of Z = A'B'+C'



A layout that does not take into account design rules



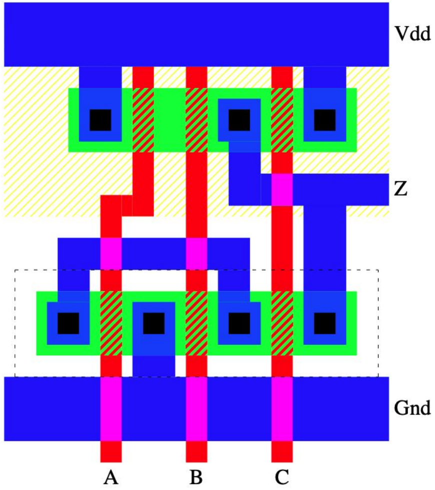
Euler Path to Layout



Consistent (same transistor ordering) Euler Paths through the PUN and PDN logic graphs.

The optimal ordering of inputs is the order of the the Euler path.

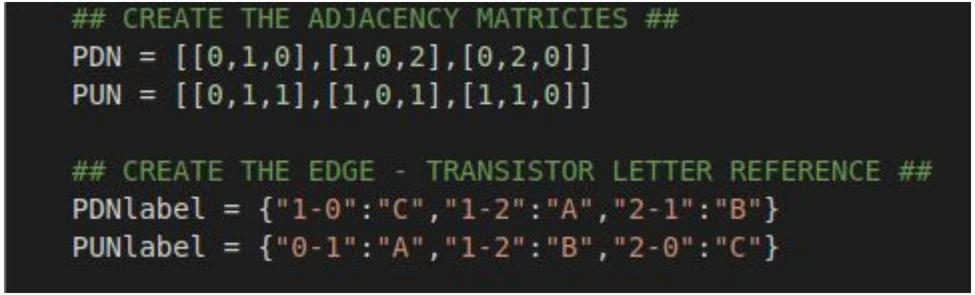
In this case, it is A-B-C.



Where Our Code Comes Into Play

We wrote a graph algorithm that goes through and finds all consistent Euler paths for an input adjacency matrix of the PUN and PDN.

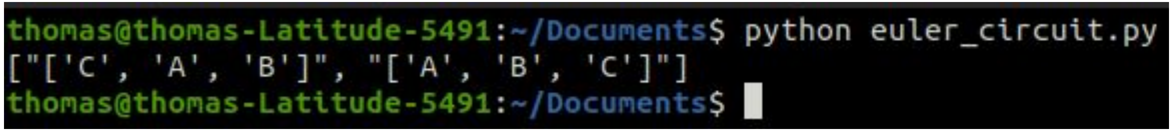
INPUT



```
## CREATE THE ADJACENCY MATRICIES ##
PDN = [[0,1,0],[1,0,2],[0,2,0]]
PUN = [[0,1,1],[1,0,1],[1,1,0]]

## CREATE THE EDGE - TRANSISTOR LETTER REFERENCE ##
PDNlabel = {"1-0":"C","1-2":"A","2-1":"B"}
PUNlabel = {"0-1":"A","1-2":"B","2-0":"C"}
```

OUTPUT



```
thomas@thomas-Latitude-5491:~/Documents$ python euler_circuit.py
["['C', 'A', 'B']", "['A', 'B', 'C']"]
thomas@thomas-Latitude-5491:~/Documents$
```

Conclusion

An interesting application of discrete math is using graphs to optimize transistor ordering such that the area it takes up is minimized. To do this, we:

1. Create PUN and PDN logic graphs from schematic
   a. Combine logic graphs to simplify the visual if desired
2. Find a consistent Euler path through logic graphs (using code)
3. Make stick diagram with the transistor ordering indicated from Euler path
4. Apply design rules on the stick diagram to make layout

Note:

1. Euler paths do not exist for all expressions.
2. Different forms of the same expression may affect whether or not an Euler path exists.
3. Euler paths are not unique.