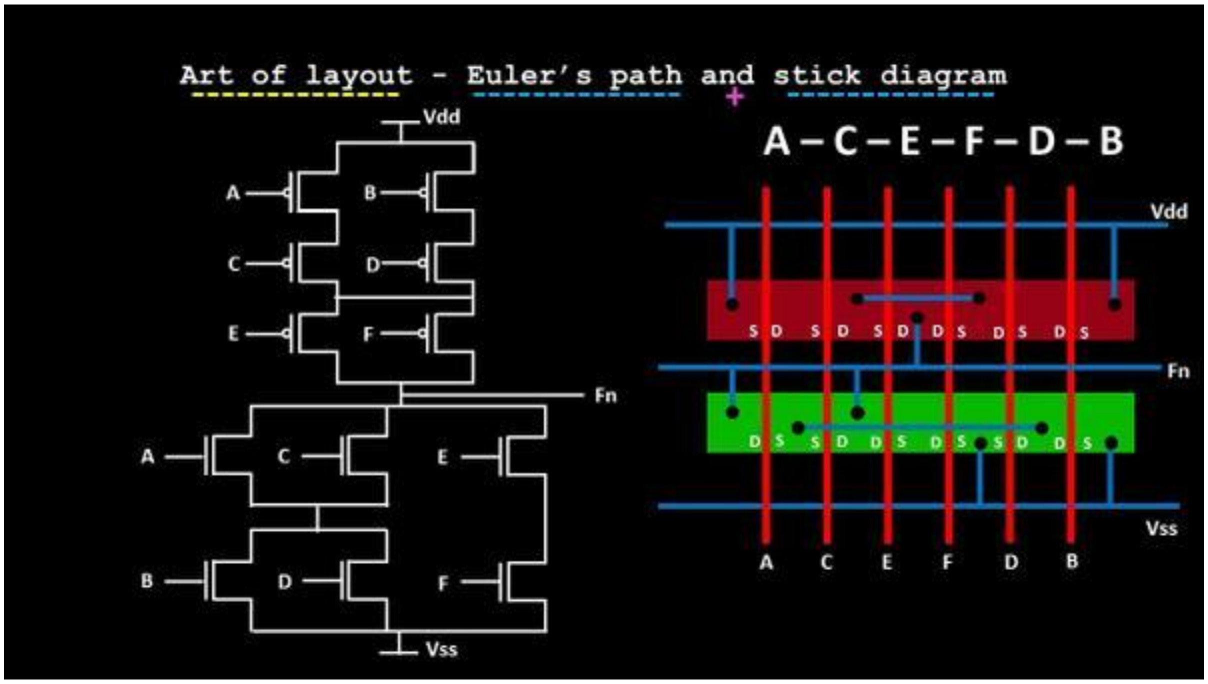
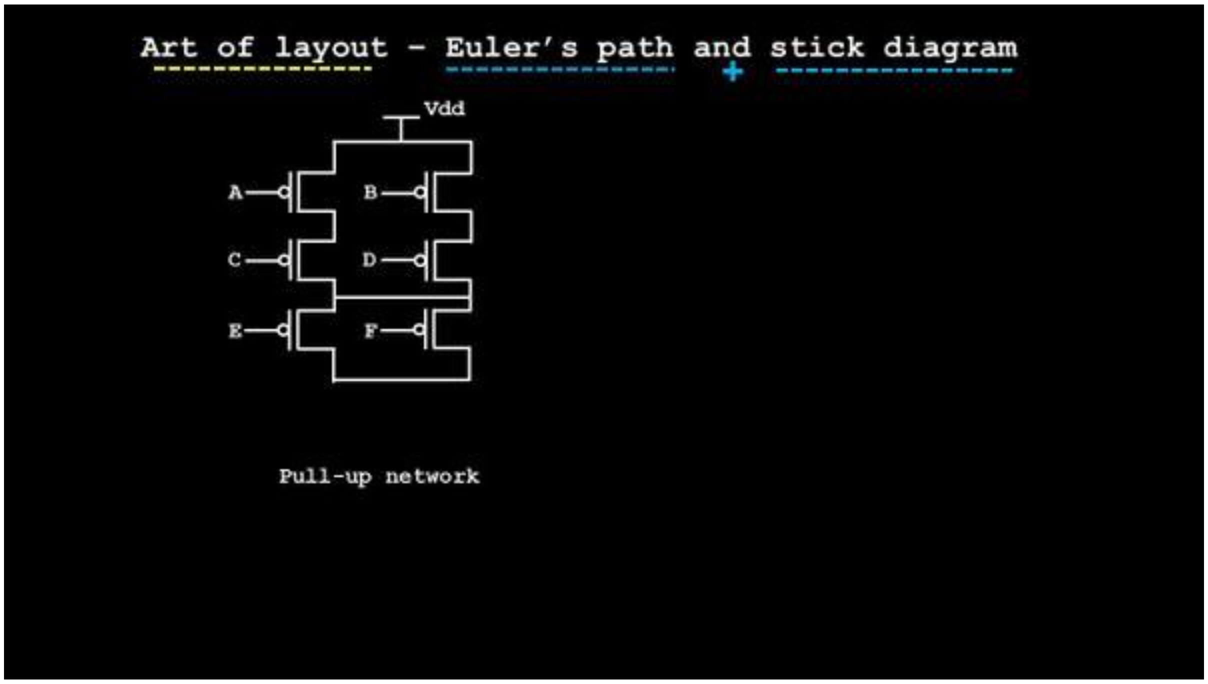
Art of Layout – Euler's path and stick diagram

Kunal Ghosh

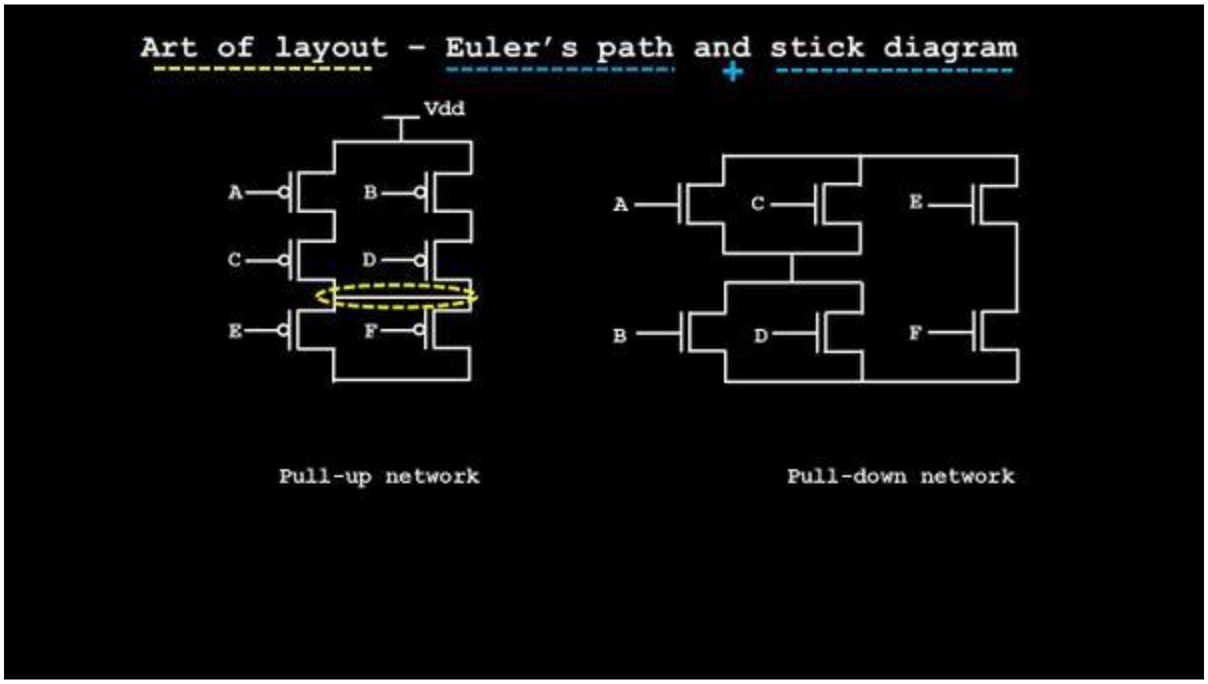


I wrote about Euler’s path and stick diagram in two different blogs, but now is the time to show you how are they connected.

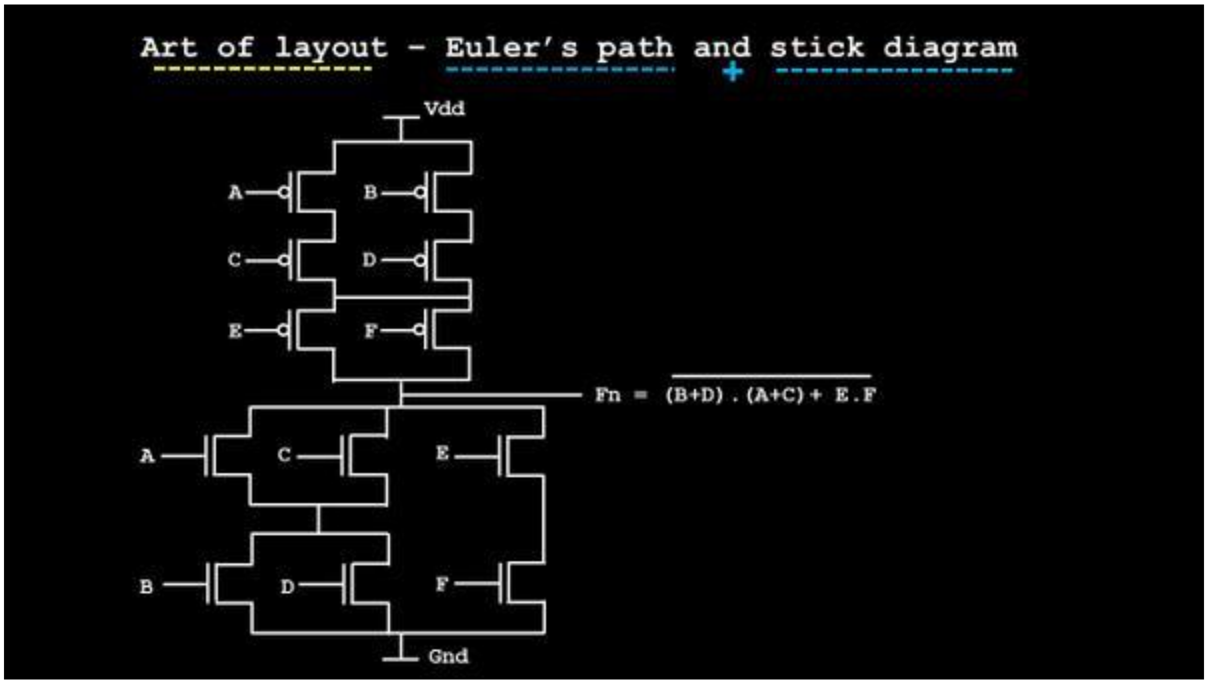
It’s simple and, seems, they can’t be separated out from each other. To prove that, let’s take a random logic using the below pull-up network and let’s use the CMOS duality theorem to build the equivalent pull-down network (I will talk more about duality theorem in my courses). For now, let’s argue about this: Using de-Morgan’s theorem, it can be shown that parallel connections of transistors in pull-up network corresponds to a series connection of pull-down networks and vice-versa i.e. dual networks



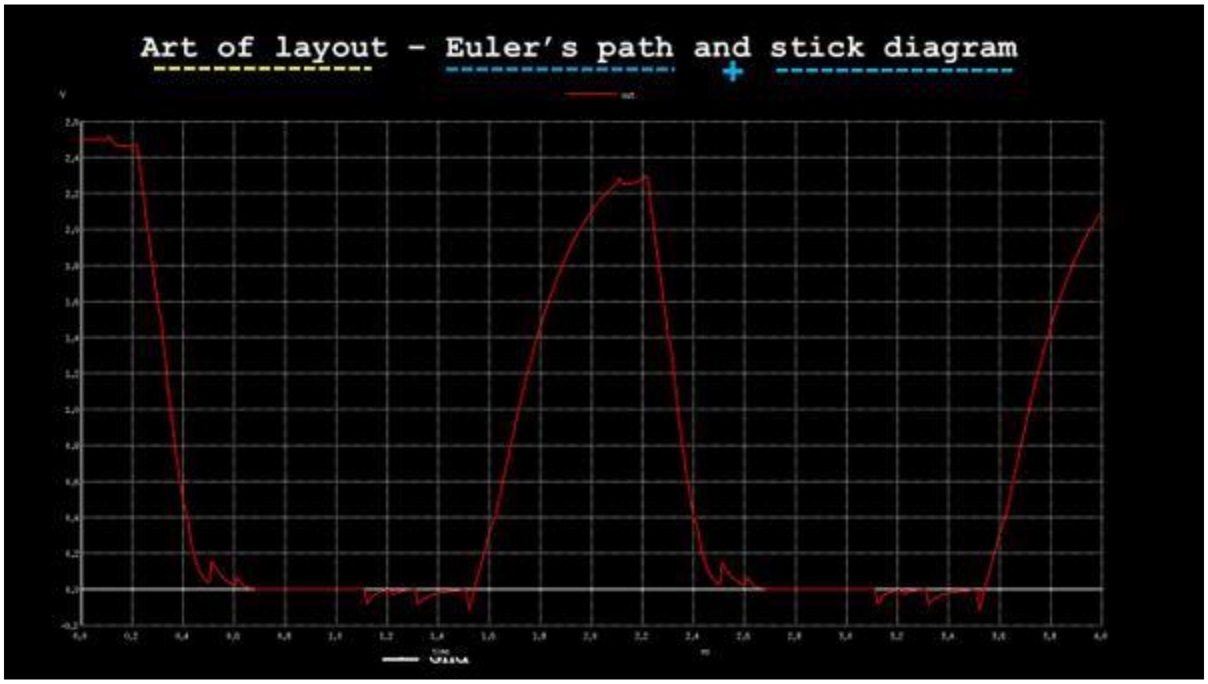
Based on that, below is the equivalent pull down network:



and here’s the whole circuit, with its respective function Fn:


Art of layout - Euler's path and stick diagram
Vdd
A
B
C
D
E
F
$Fn = \overline{(B+D) . (A+C) + E.F}$
A
C
E
B
D
F
Gnd

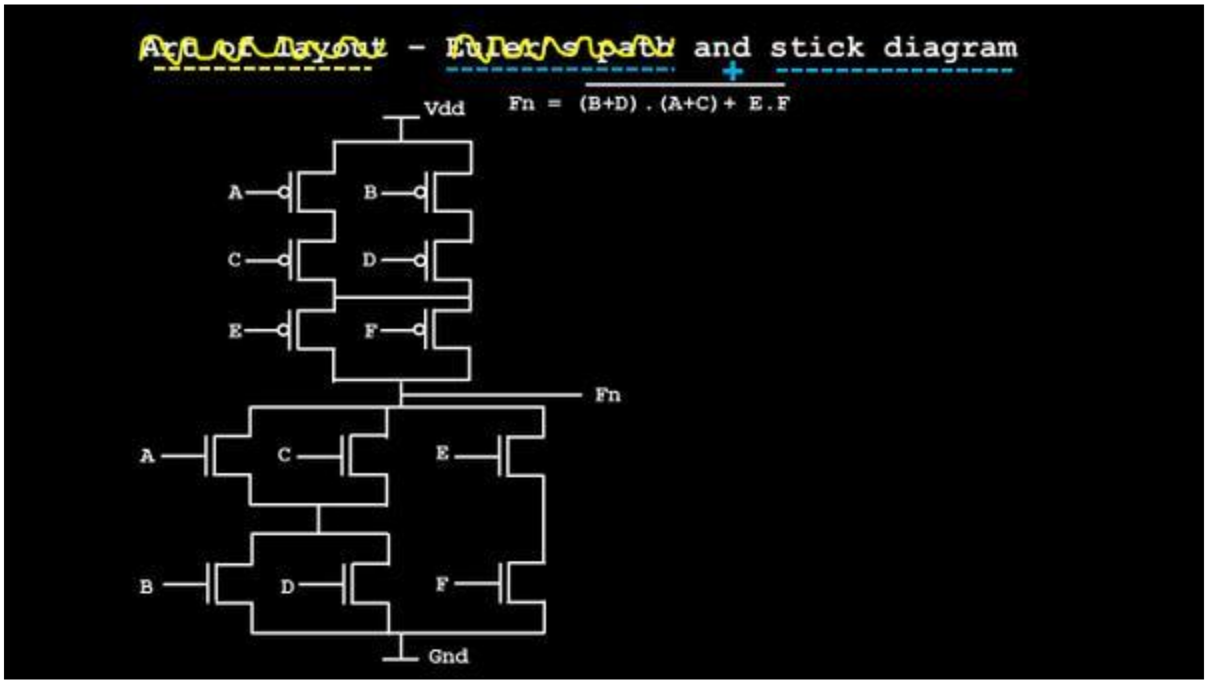
The first rule of any layout: Its characteristics in pre-layout phase should match with the characteristics of post-layout phase. Barring few parasitic delay, the shape and nature of waveform should exactly match. Considering that, I did a pre-layout SPICE simulation of the above network and below is what I get:



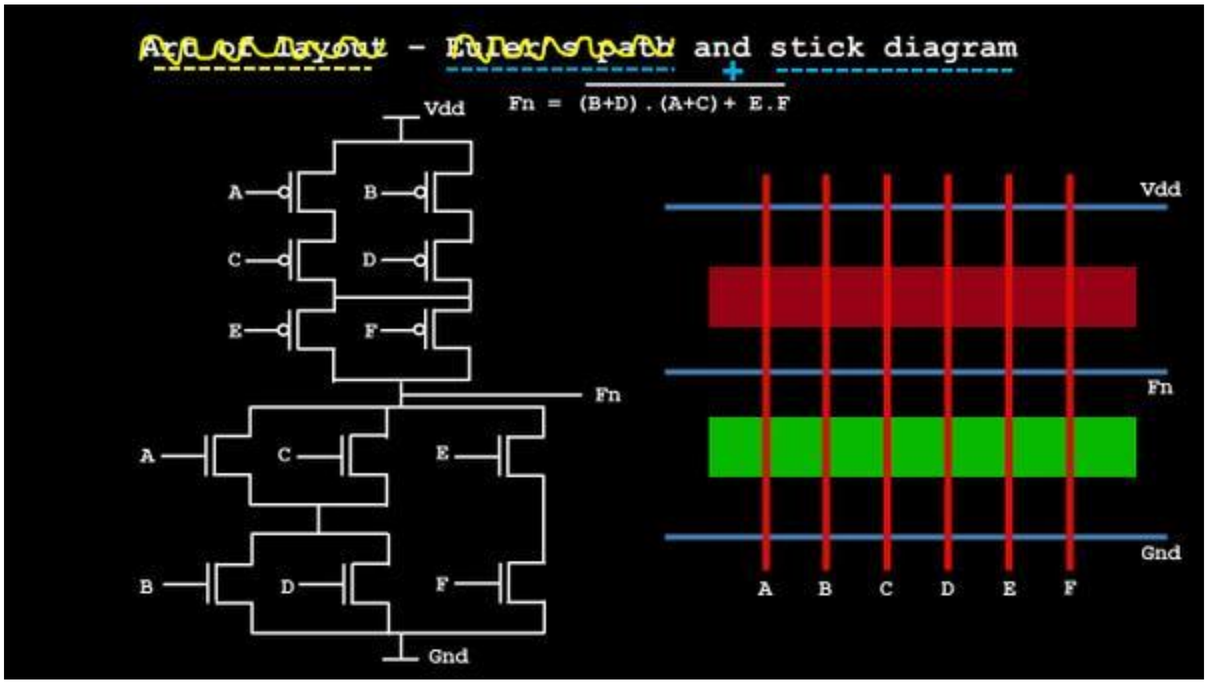
If, till this part, is clear and concise, the next job is to simply use the concepts of Euler's path and stick diagram, implement the layout and simulate it to get the above waveform…. easy…. right?

So, I have been bragging about that 'art of layout' is a combination of Euler's path and stick diagram. But you need a proof why am I saying that

Let's first find out what happens if we decide to go only by stick diagram and do not take Euler's path into account

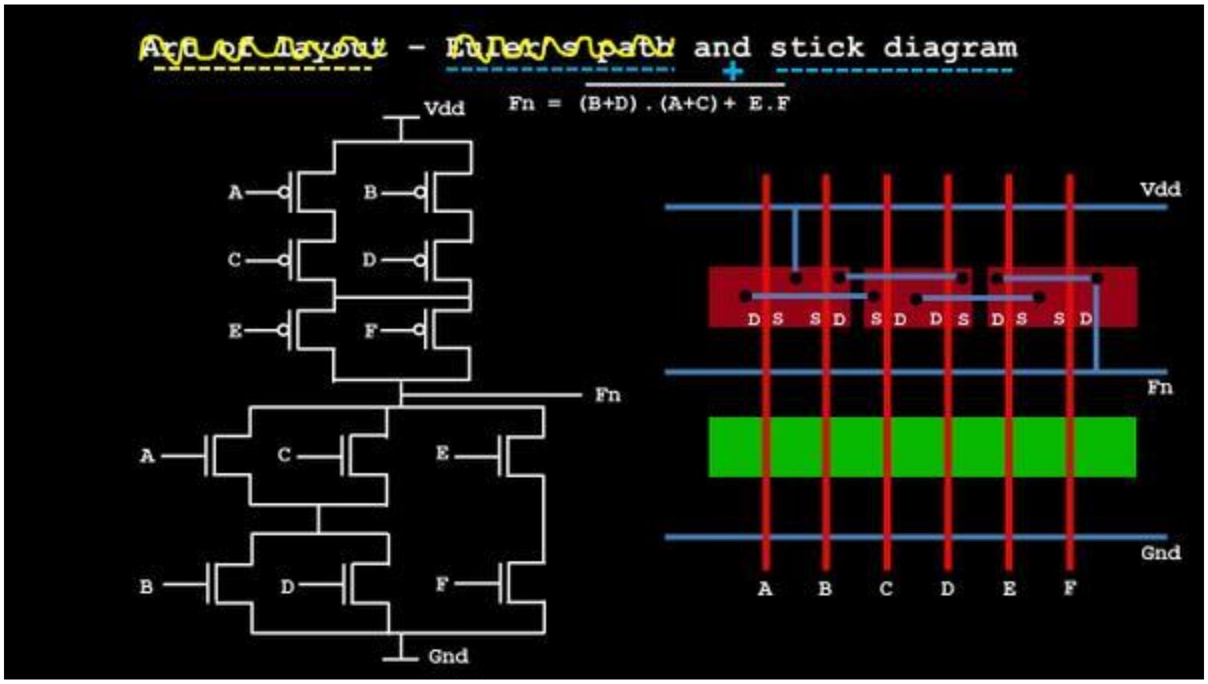


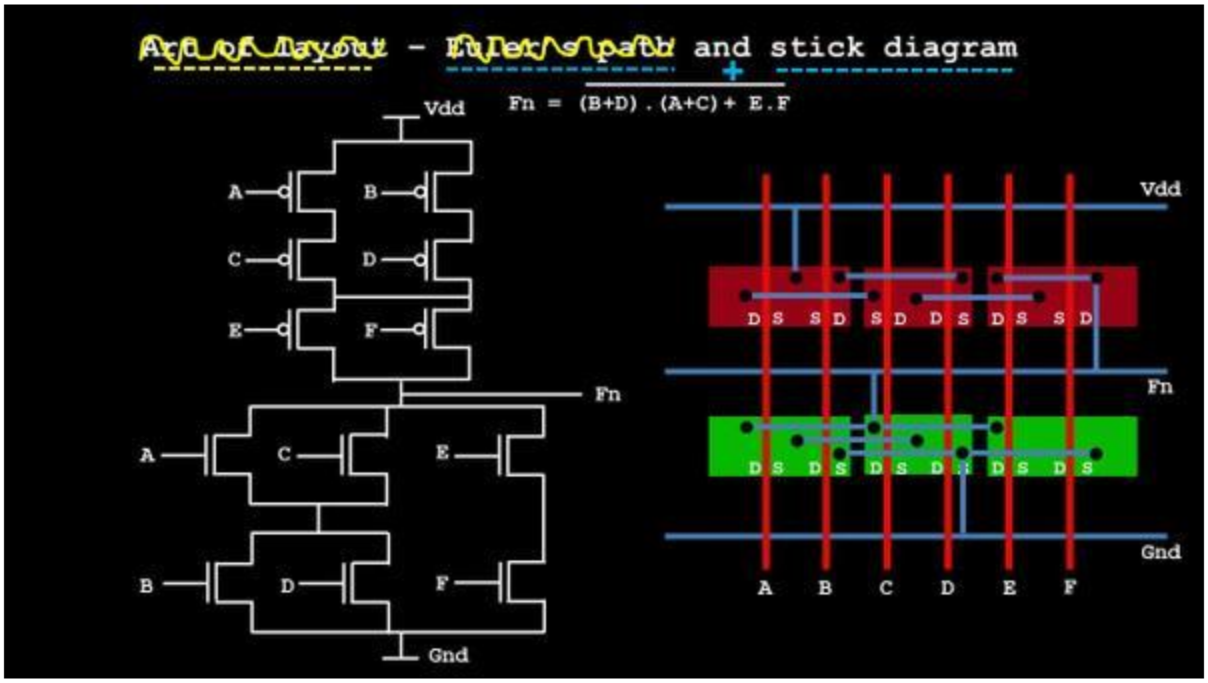
I will go by the rules of stick diagram… first draw supply lines, then output lines, then the pdiffusion, ndiffusion and finally the polysilicon strips in some random order A – B – C – D – E – F. Below is how it will look.



Art of Layout - Euler's path and stick diagram
Fn = $\overline{(B+D).(A+C)+E.F}$
Vdd
A
B
C
D
E
F
Fn
Gnd

Then, let's move further with some rules of stick diagram for connectivity…If you very carefully select the source 'S' and 'D' region around poly, then most likely, you will end up something like below for pull-up network and you will end up something like below for the pull-down network (you can verify the correctness of connectivity and ***in case you are looking for concepts on stick diagram, please take up custom layout course from here)***

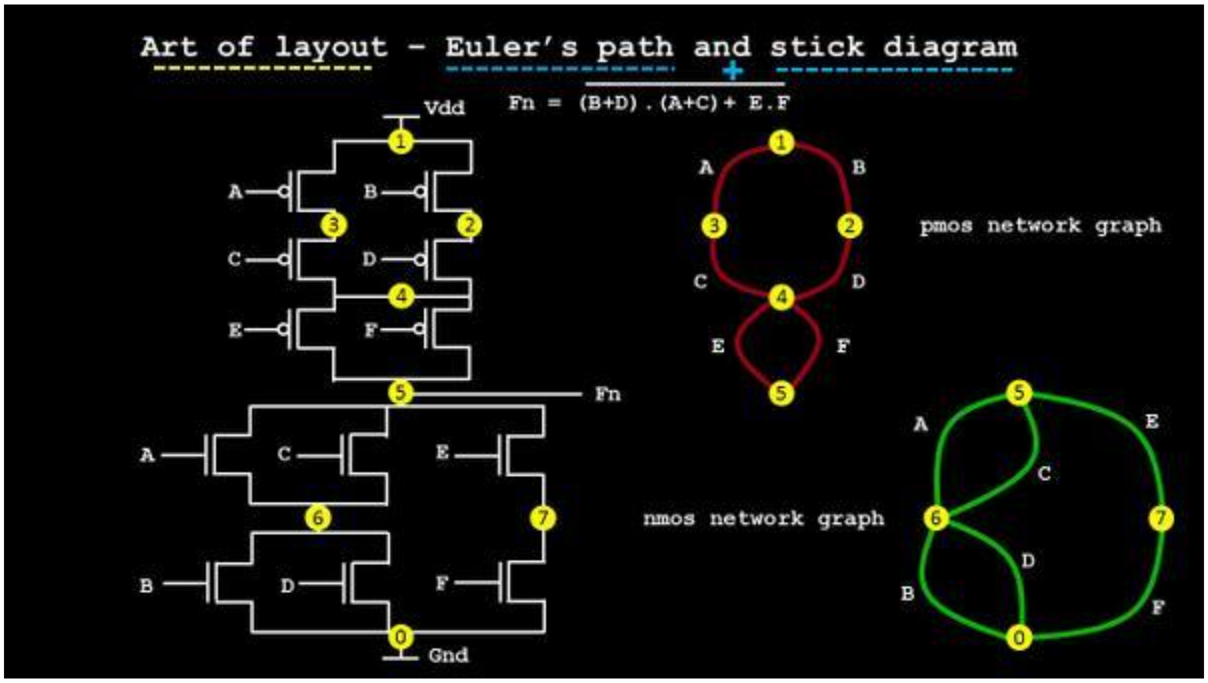




So, what do you see here …. the p-diffusion and n-diffusion continuity is miserable, too many breaks, clumsy metal lines, highly congested and complex to fabricate…. This is just a small logic function we are trying to build, imagine what will happen, if we select poly strips in randomness…. just think about the congestion for metal connections, it will be a nightmare to fix DRC's for such congested design….

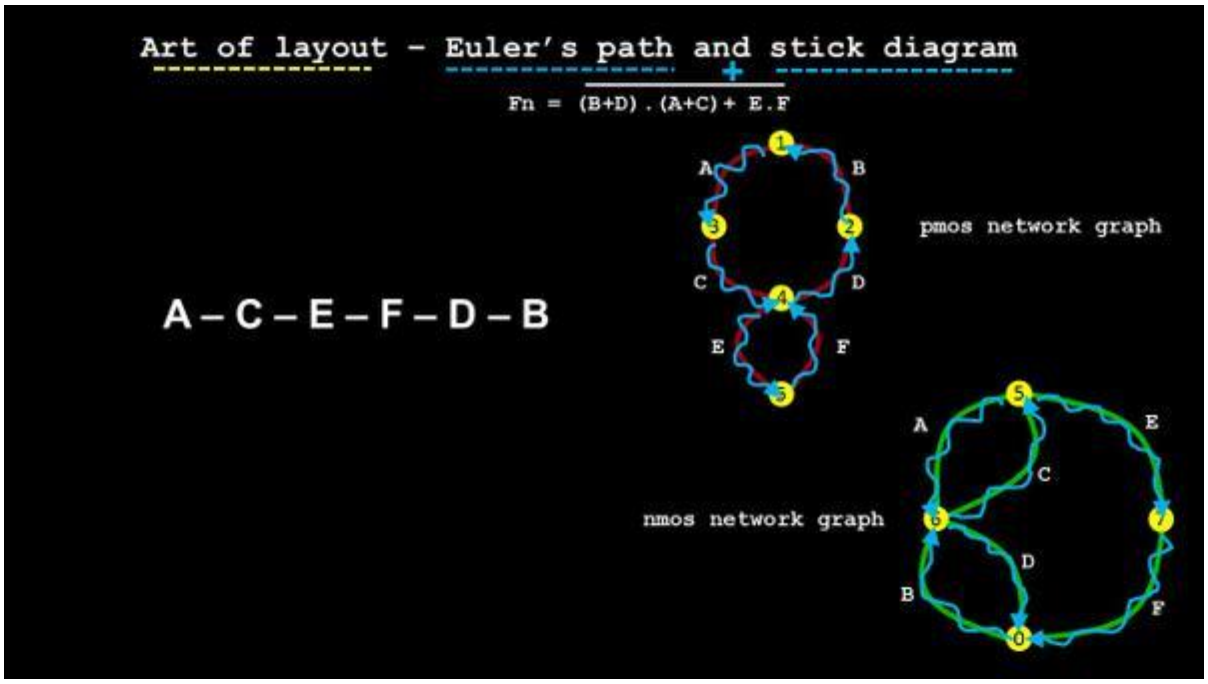
After the terrible layout, we saw in last 2 blogs, without considering Euler's path, it's now time to mend things and do it the right way, i.e. create an accurate gate input ordering using Euler's path, extracting stick diagram and finally drawing the layout.

Let's first create the below pmos and nmos network graph using transistors gate inputs as 'edges'. (to learn more about Euler's path, Euler's circuit and stick diagram, visit this link)

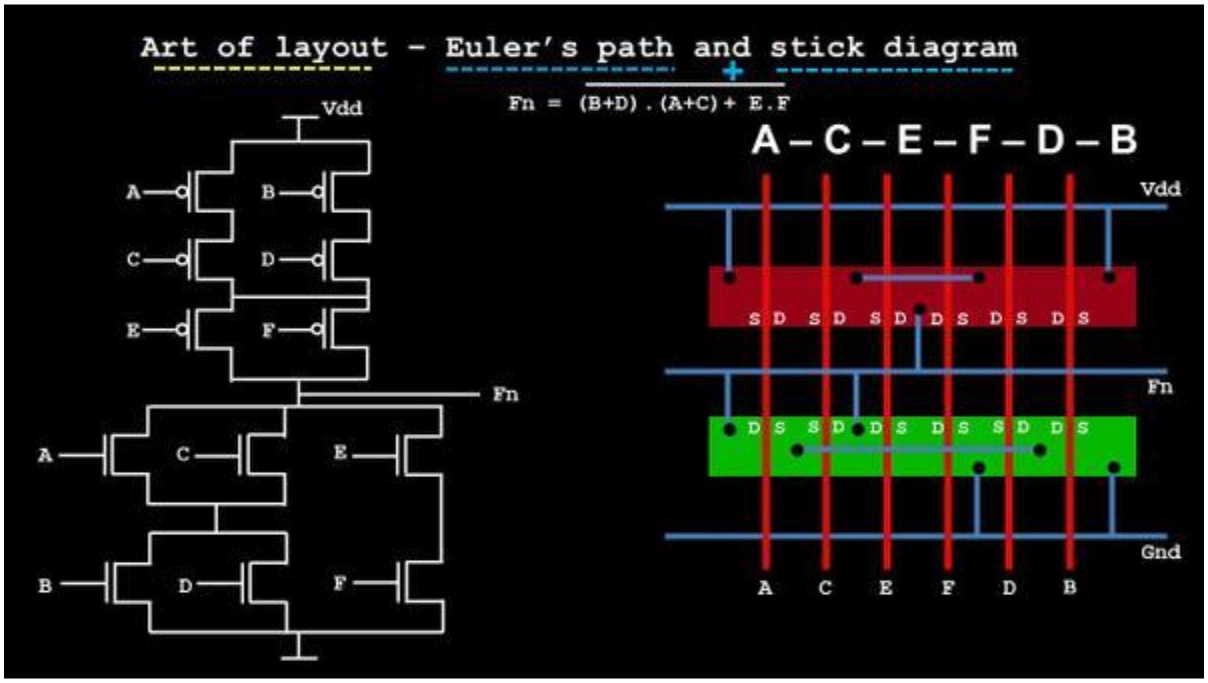


The node number 1, 2, 3, 4…etc. which you see encircled with yellow are called vertices and the gate inputs which labels the connections between the vertices 1, 2, 3, 4, etc. are called edges. Find out more about vertices and edges in this link

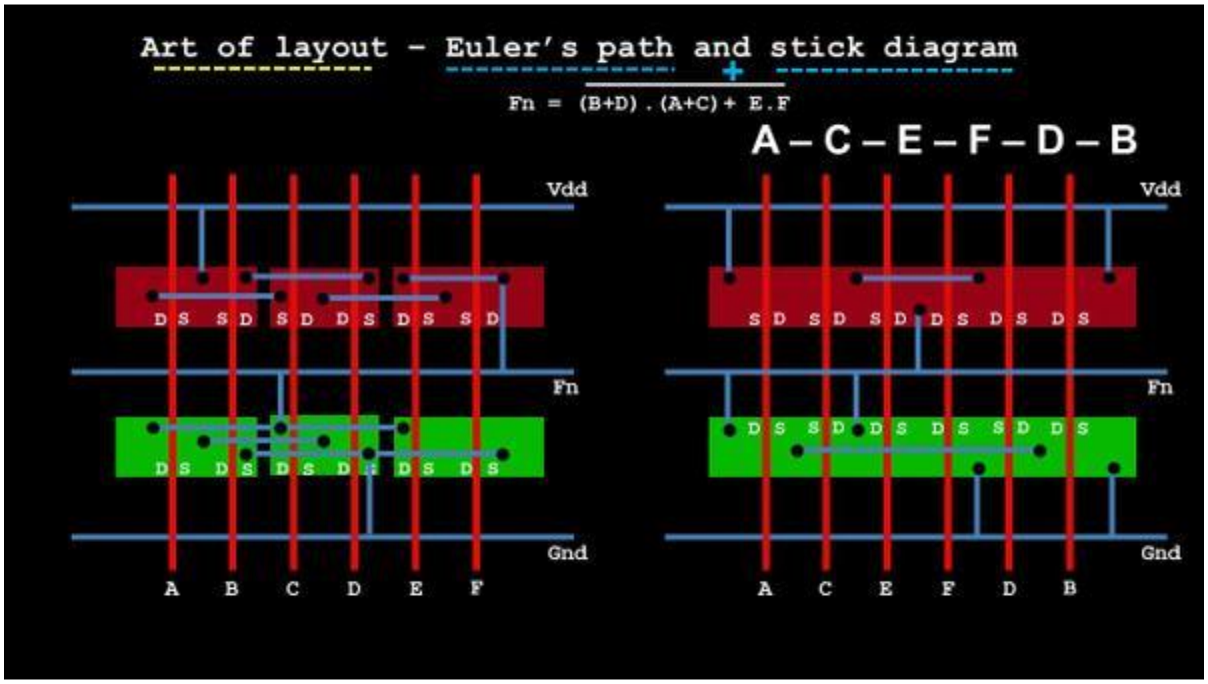
Next, we trace a path in both nmos and pmos network, in such a fashion that each edge is traversed only once. While we do that, we keep on registering the edge (or the gate input) we are travelling. I do that and I get the below Euler's path i.e. A-C-E-F-D-B



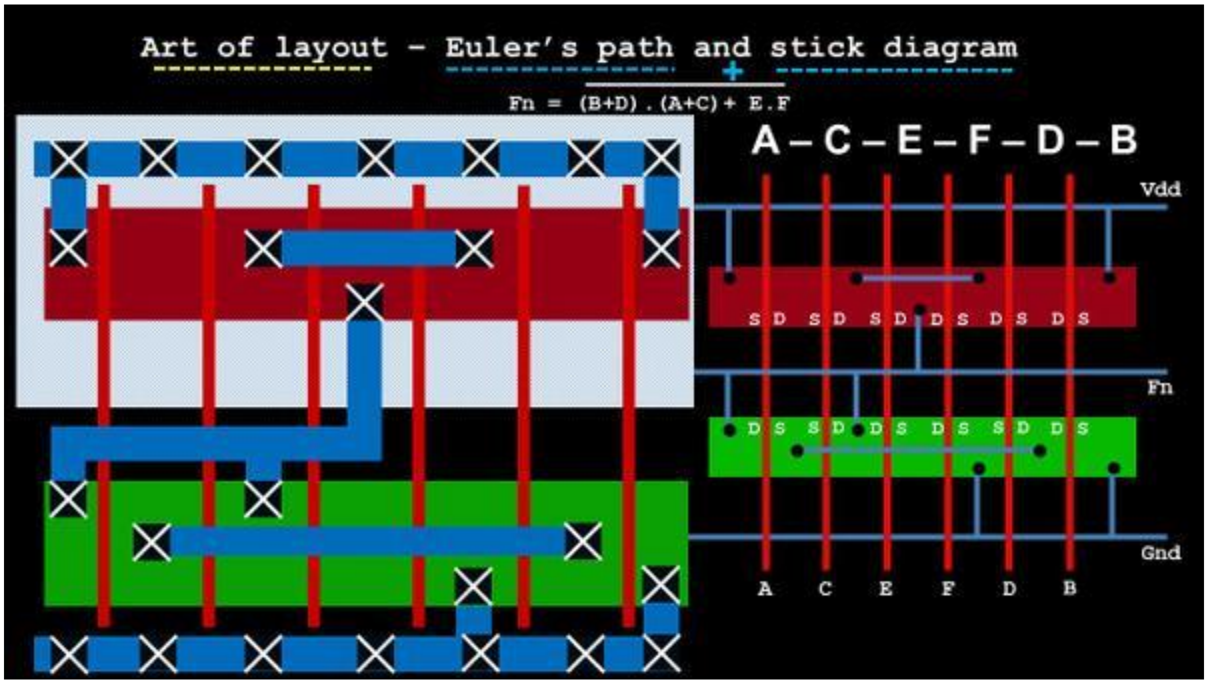
Do you want to see the power of Euler's path? See the below layout implementing the same circuit, but in a much simpler fashion as compared to one which we did in our previous post



Now compare this with previous layout below. Doesn't this look much cleaner, much organized and much simpler to implement in a layout editor? It does look…



Once done with stick diagram, the last step is the easiest one. i.e build the layout along with dimensions pertaining to design rules and below is what you get:



Wasn’t that a smartest way to implement layout? And I guarantee you, you take the toughest design, break into smaller logic, build each logic using Euler’s path and stick diagram, and connect each layout back…. you will get the most optimized layout in terms of area and power. Guess what you need to do to increase the performance…. just increase width of p-diff/n-diff and make it low resistance path to VDD and GND… This is what my STA friends call as ‘sizing’.