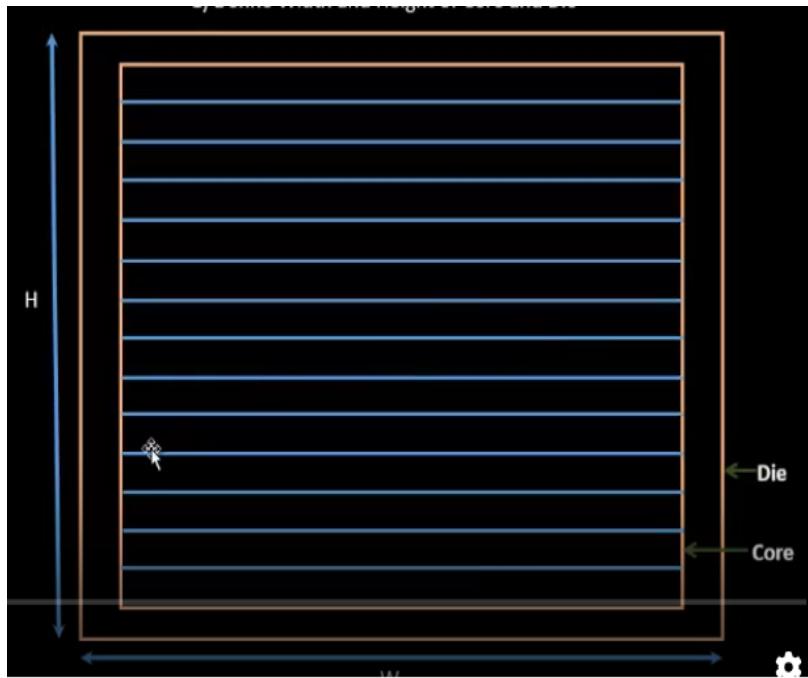


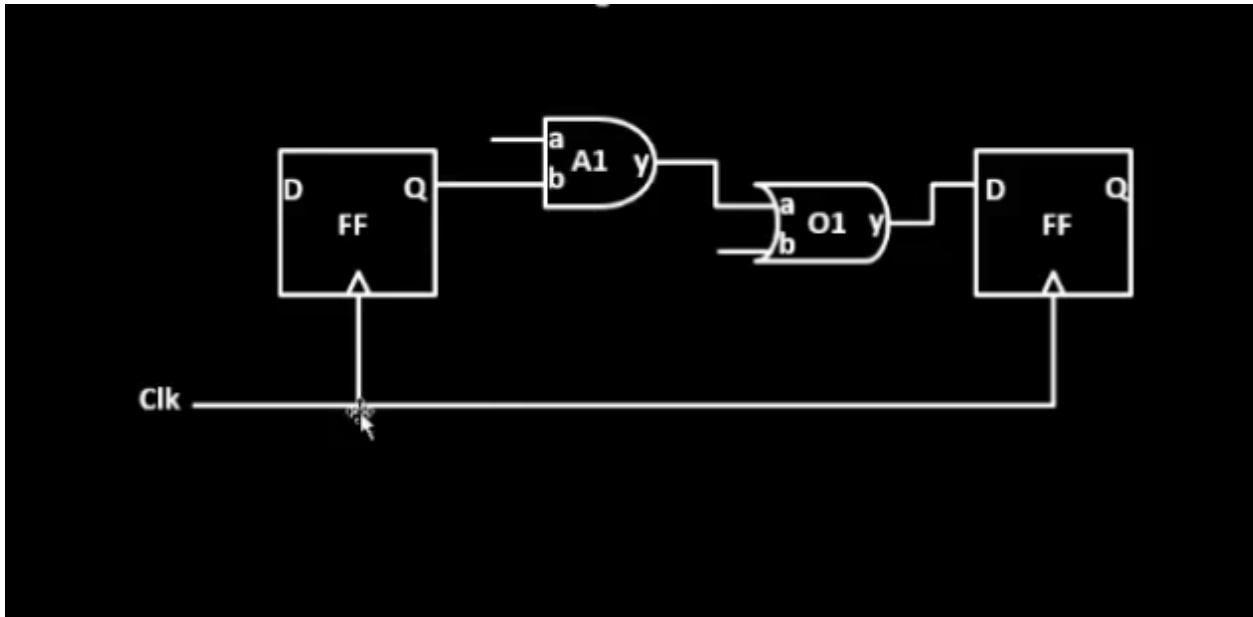
Day 2



This is the first step of defining width and height of core and die. This is the first step of the physical design flow

Values of W and H

1)Netlist



It is a basic net list.

A1 and O1 are “and gates”, “or gates”, “inverters”. They fall under the class of standard cells.

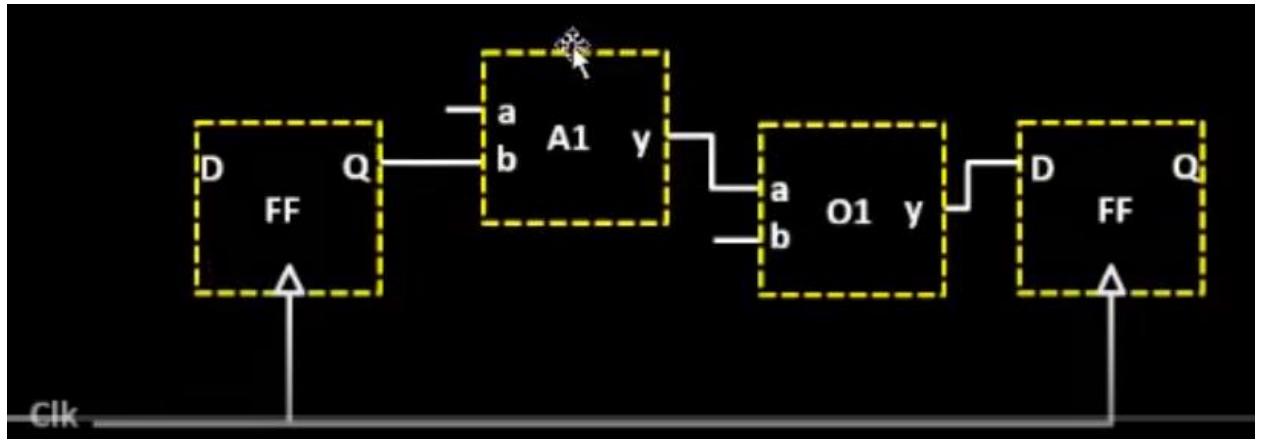
FF1 and FF2 fall under the class of registers or latches.

A net list defines a connectivity between all the components.

While defining the dimensions of the chip we are mostly dependent on the dimensions of the logic gates.

The next step is

Now we have to give a proper length and breadth to the particular gates.

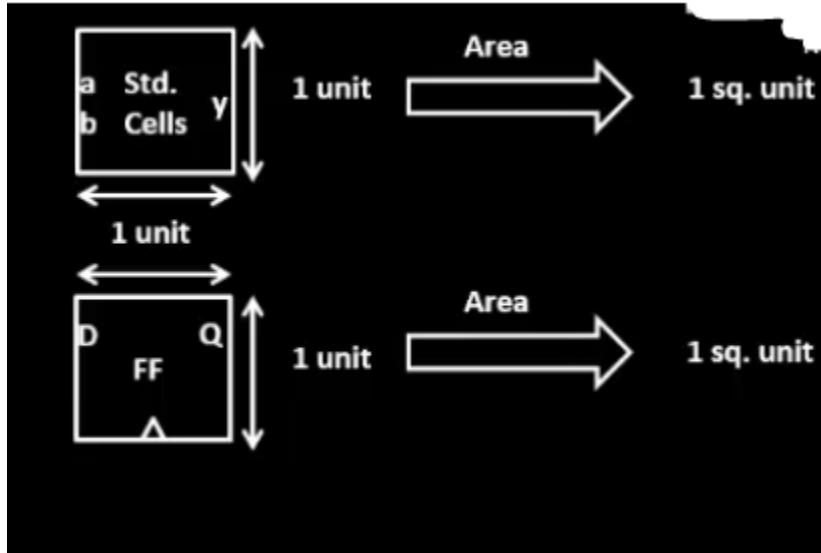


Next step.

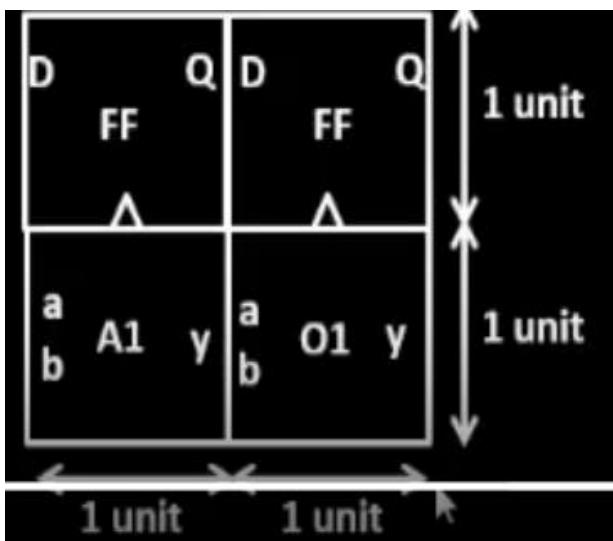
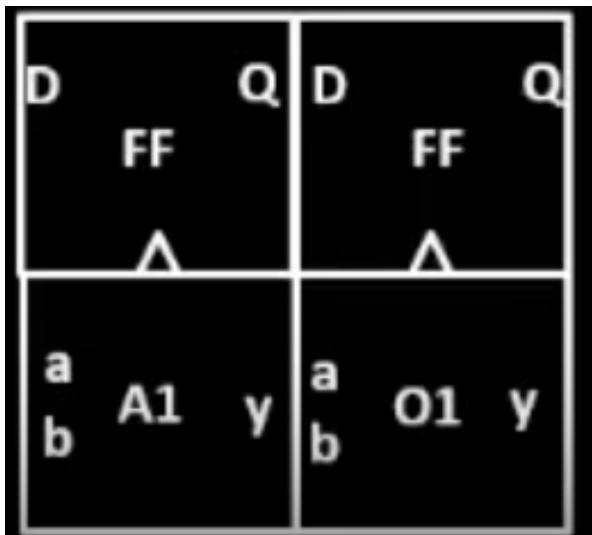
Now let's give a rough dimension for all of them.

Let's say the standard cells are given dimension of 1 unit by 1 unit. So the area of this standard cell will be 1sq unit.

Lets say that the flip flop has also same area(1sq unit).



Now we will remove all the wires and we will club all together in a single plate.



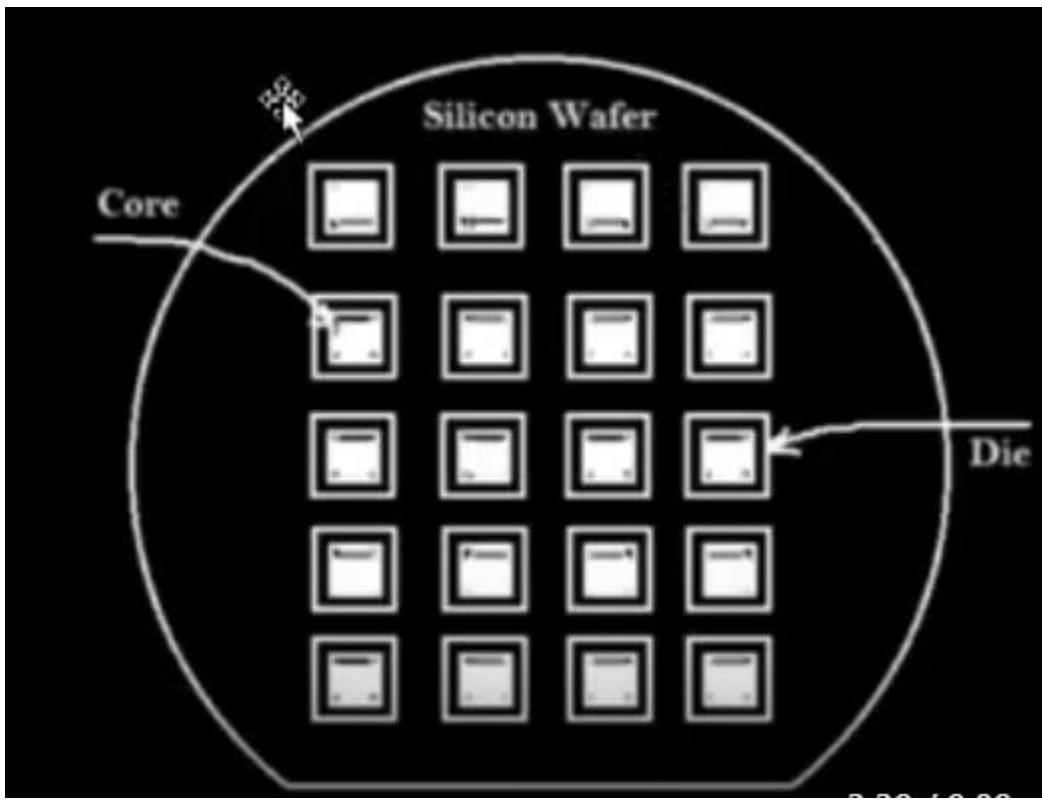
So the length will be 2 unit and breadth will be 2unit.

Area

$$= l * b = 2 * 2 = 4 \text{sq unit.}$$

So this is the minimum area occupied by the net list wherever it is placed.

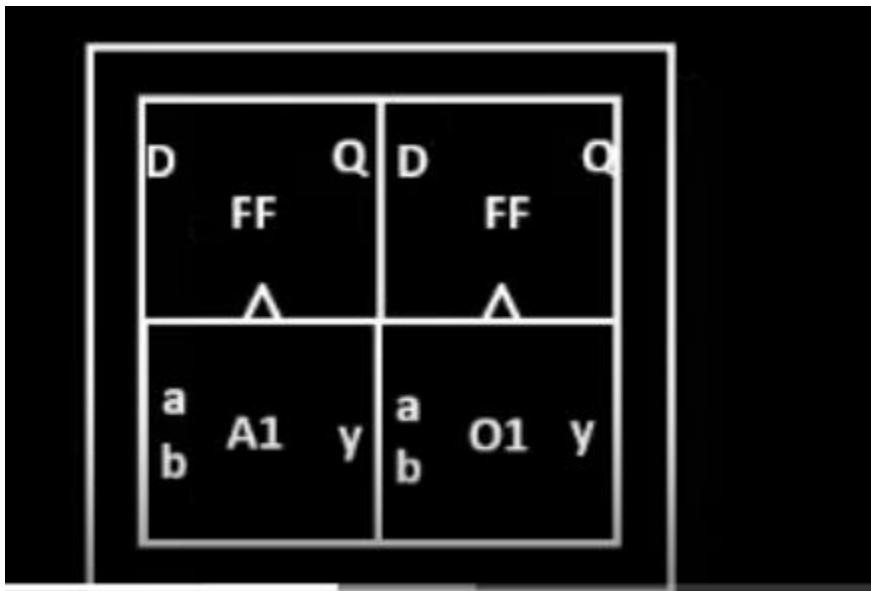
If we want to implement all of our logic in a silicon wafer, one section is referred to as the die. Inside the die we have the core.



A core is the section of the chip where all the fundamental logic design is placed.

A die, which consists of core, is a small semiconductor material specimen on which the fundamental circuit is fabricated.

Now we will keep a particular logic inside the core.



The net list which was 4sq units occupies the complete area inside the core. The complete core is occupied by the netlist. This is called as 100% utilization.

Utilization factor = Area occupied by the net list / Total area of the core.

The area occupied by the net list is 4sq units.

The area occupied by the core is the area of the particular core.

Utilization factor = $4 * 1 / 2 * 2$

$$=4/4$$

$$=1$$

When we say the utilization factor is 1 that means the core is fully occupied. If we want to add extra cells in the core that is not possible.

Aspect ratio = height / width.

In this particular example height=2, width =2 .So aspect ratio = $2/2 = 1$.

Whenever the aspect ratio is 1 the shape is square.

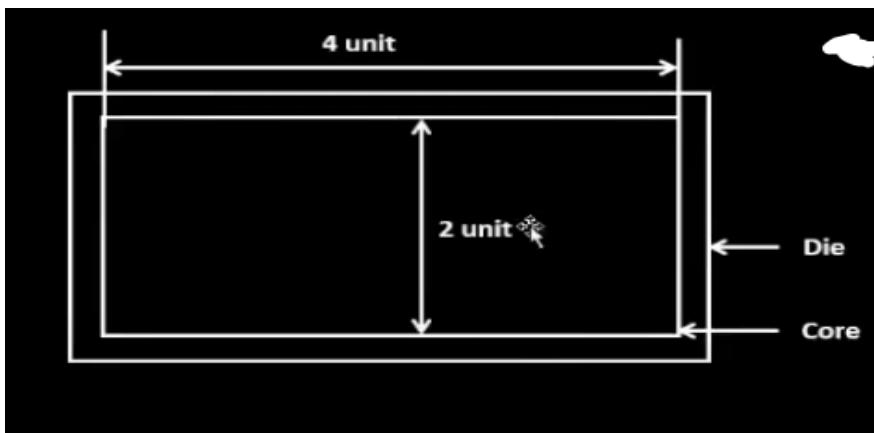
Whenever the aspect ratio is other than 1 something like 0.75,then it signifies the shape is rectangle.

Another example:

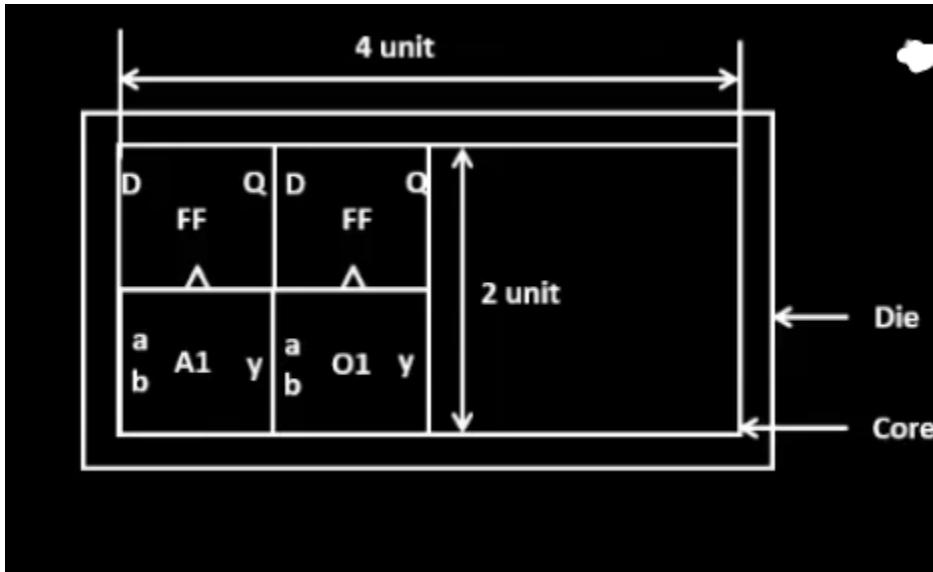
Dimensions of the net list = $2*2$ (4 sq units)

Height of the core = 2 unit.

Width of the core = 4 unit.



Now when we place the logic inside this it will look like this.



Utilization factor = Area occupied by the net list / Total area of the core.

$$\text{Utilization factor} = 4/8$$

$$= 1/2.$$

Any number less than 1 states there is still space for optimization.

This signifies a rectangle.

Aspect ratio = height / width.

$$= 2/4$$

Aspect ratio = $\frac{1}{2}$

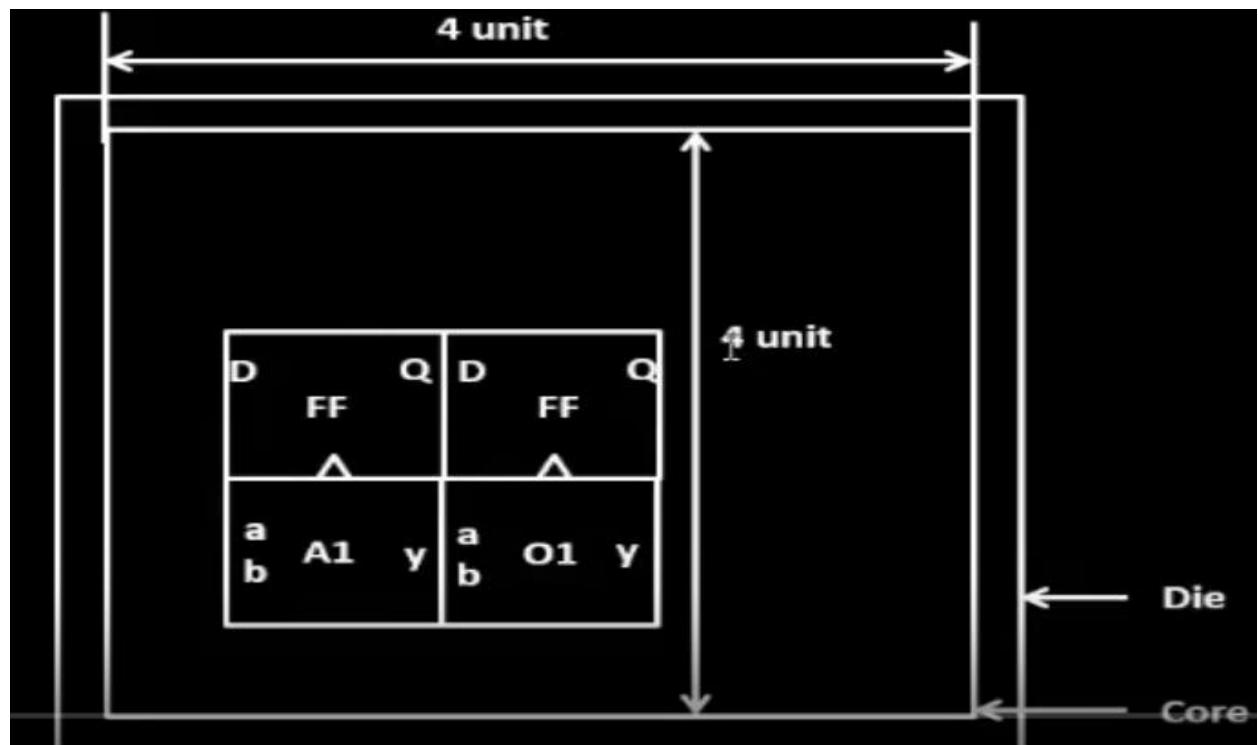
Another Example:

Height of the core = 4

Width of the core = 4

Dimensions of net list = 2×2 . (4sq units)

Now we place the logic inside the core. It will look like this.



Utilization factor = Area occupied by the net list / Total area of the core.

Utilization factor = $4/16$

$$=1/4(0.25\%)$$

25% of the core is occupied by the net list which is connected completely with ideal wires. We have 75% of the chip area available where we can place additional cells which can be used for routing

This signifies a square

Aspect ratio = height / width.

$$= 4/4$$

$$=1.$$

Now we have come up with the width and height of the core and the die.

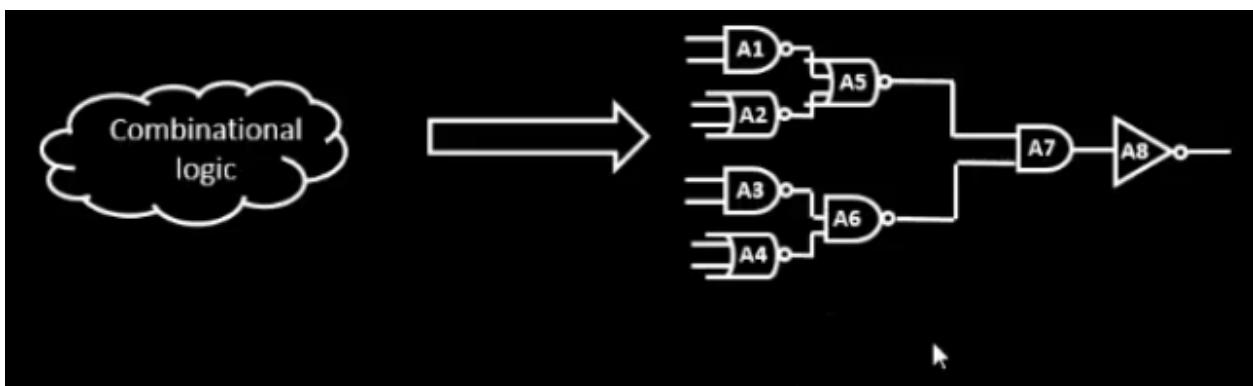
2)The next step is to define the locations of the predefined cells

Example:

Let's take a combinational logic

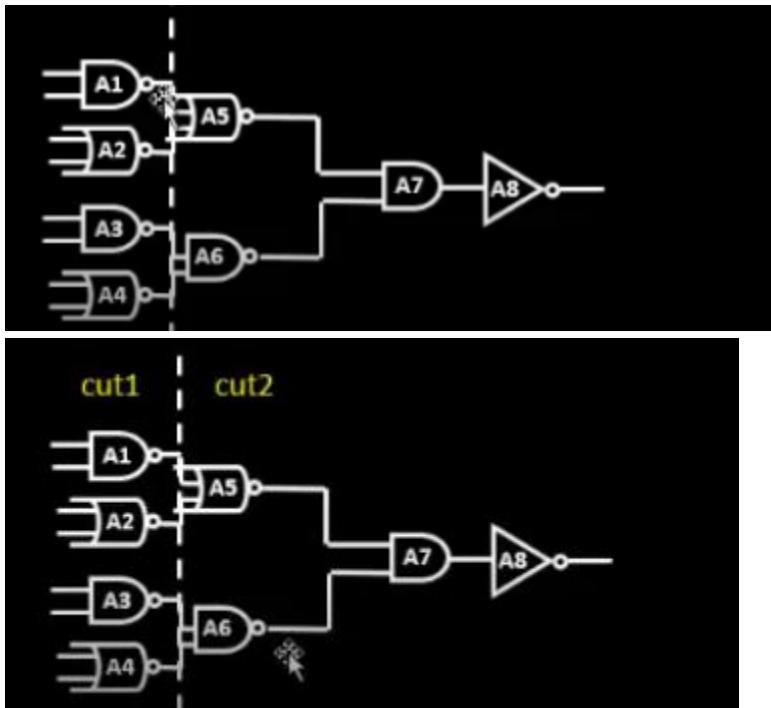
Let's assume that this logic does some amount of function. It may be a memory or a multiplier or complex mux or clock divider.

This logic performs such a big task that the output of the combinational logic is a huge circuit. It's a circuit of 50k gates or 100k circuits.

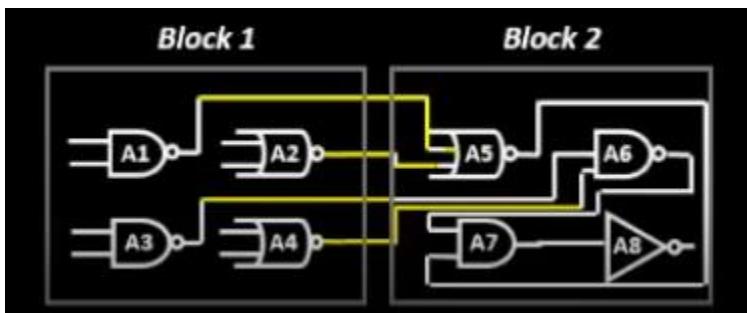


There is a way that we need not implement this circuitry every time as part of the main circuit. But we can take a piece of circuit out of the main circuit and implement that separately and even we can granularise the particular circuit itself. Let's say if it was 100K gates we can divide a circuit in such a way that it is 50K and 50K.

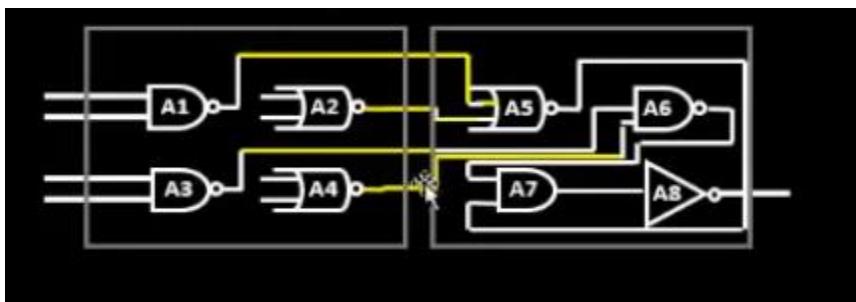
Lets take the circuit out and do some processing.

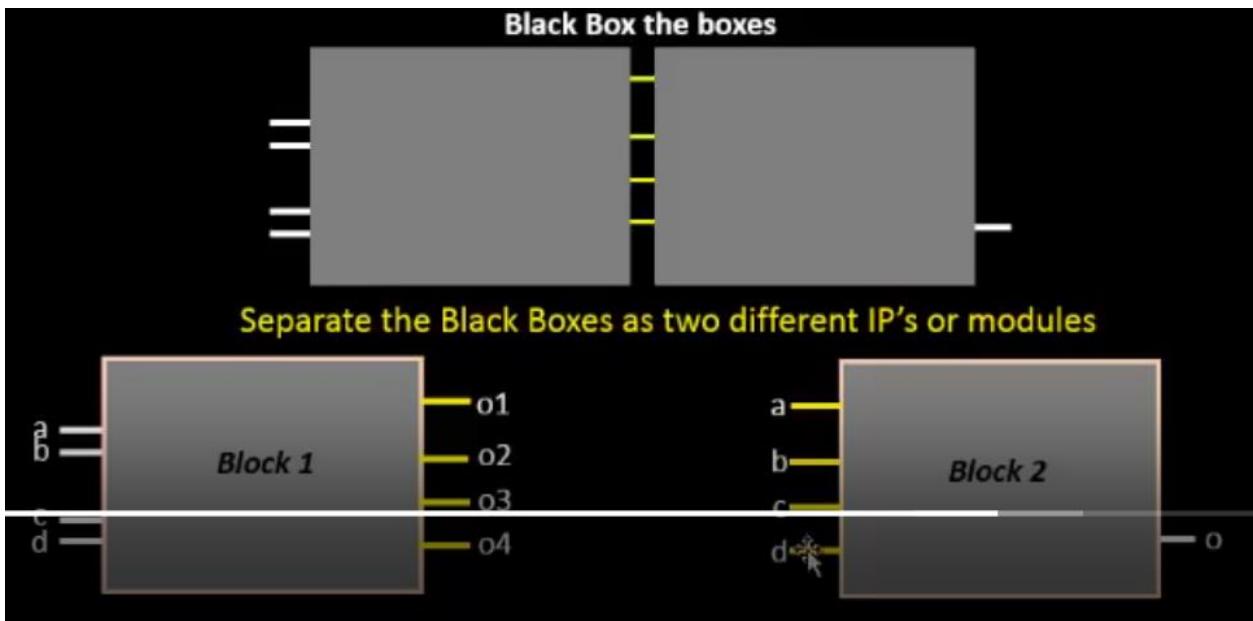


Lets separate them into blocks.



Extended IO pins.

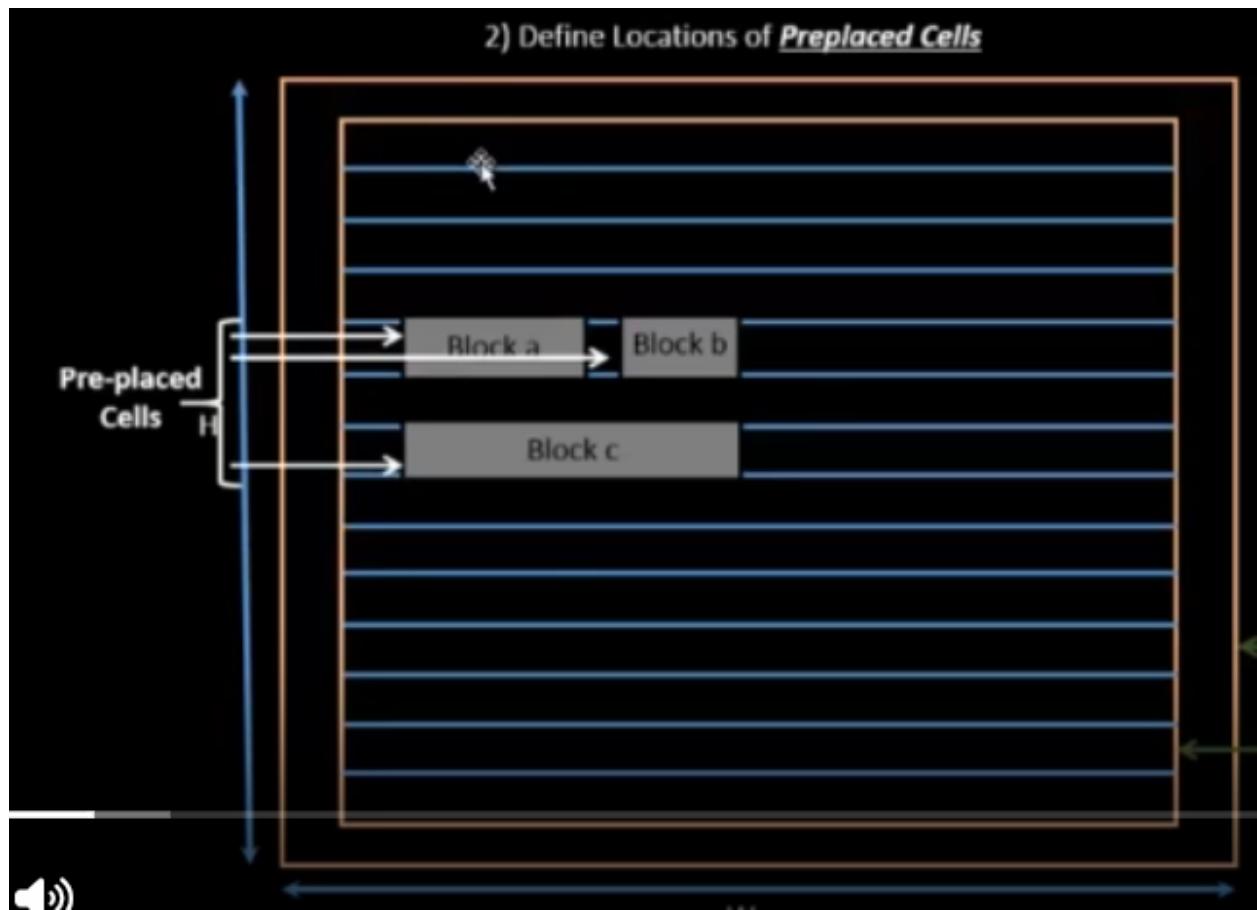




Arrangement of IP's in a chip is known as floorplanning.

Block a ,block b and block c are memories. We implement once and use multiple times. We will place block a block b and block c in input side. The pre placed cella are not touched,they cannot be moved. They have to be very well defined. We have to surround them with de-coupling

capacitors.

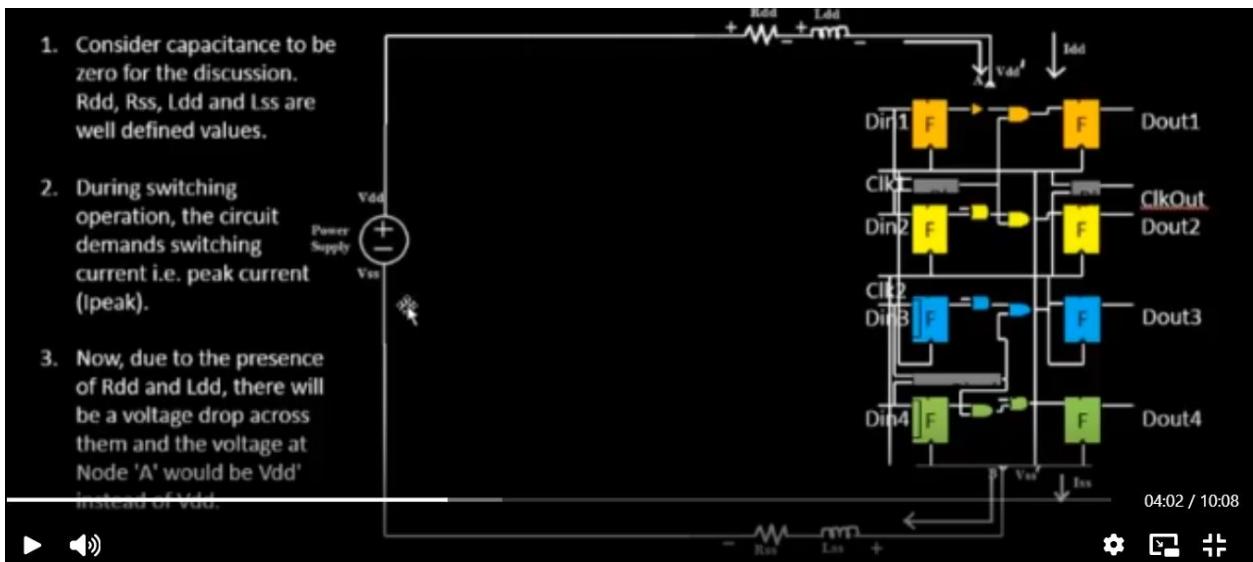


Lets say de-coupling capacitors is part of block a block b and block whenever there is switch from 0 to 1 it is the responsibility of the vss to take the charges. Resisance and induction are repeated multiple times. When vdd flows there is a drop (0.7 volts). 0.7 volts should be within the voice margin.

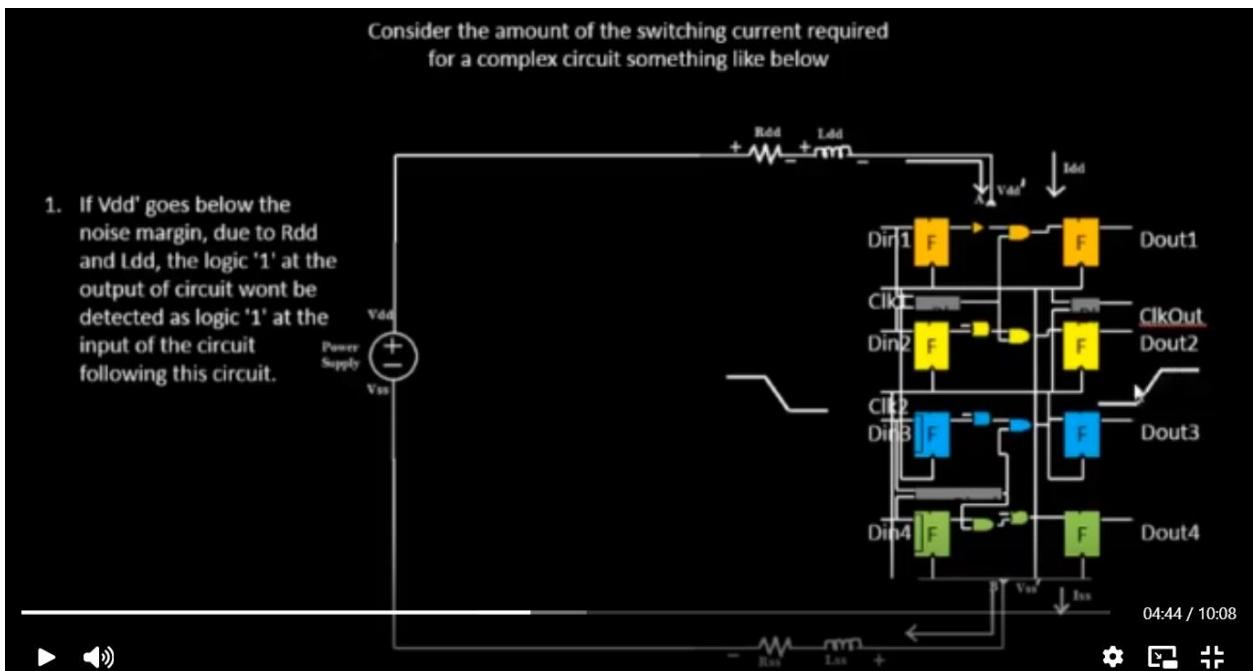
1. Consider capacitance to be zero for the discussion.
Rdd, RSS, Ldd and Lss are well defined values.

2. During switching operation, the circuit demands switching current i.e. peak current (I_{peak}).

3. Now, due to the presence of Rdd and Ldd, there will be a voltage drop across them and the voltage at Node 'A' would be Vdd'



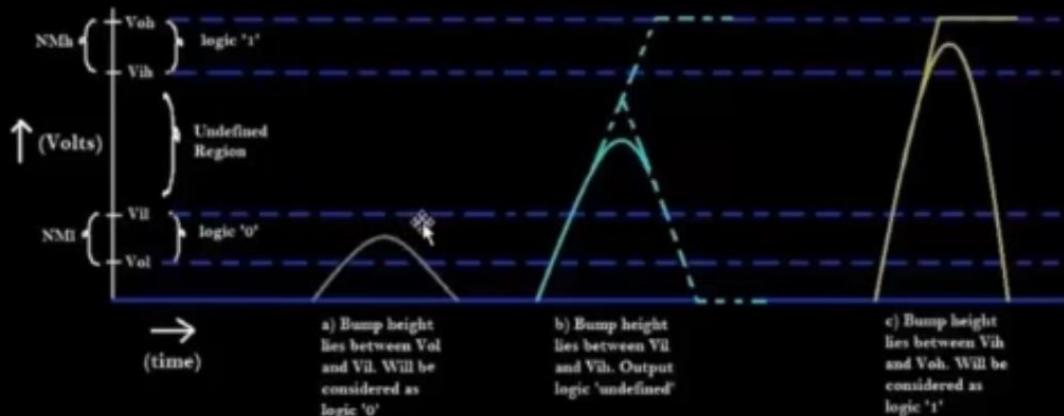
1. If Vdd' goes below the noise margin, due to Rdd and Ldd, the logic '1' at the output of circuit wont be detected as logic '1' at the input of the circuit following this circuit.



Noise margin diagram.

To exit full screen, press Esc

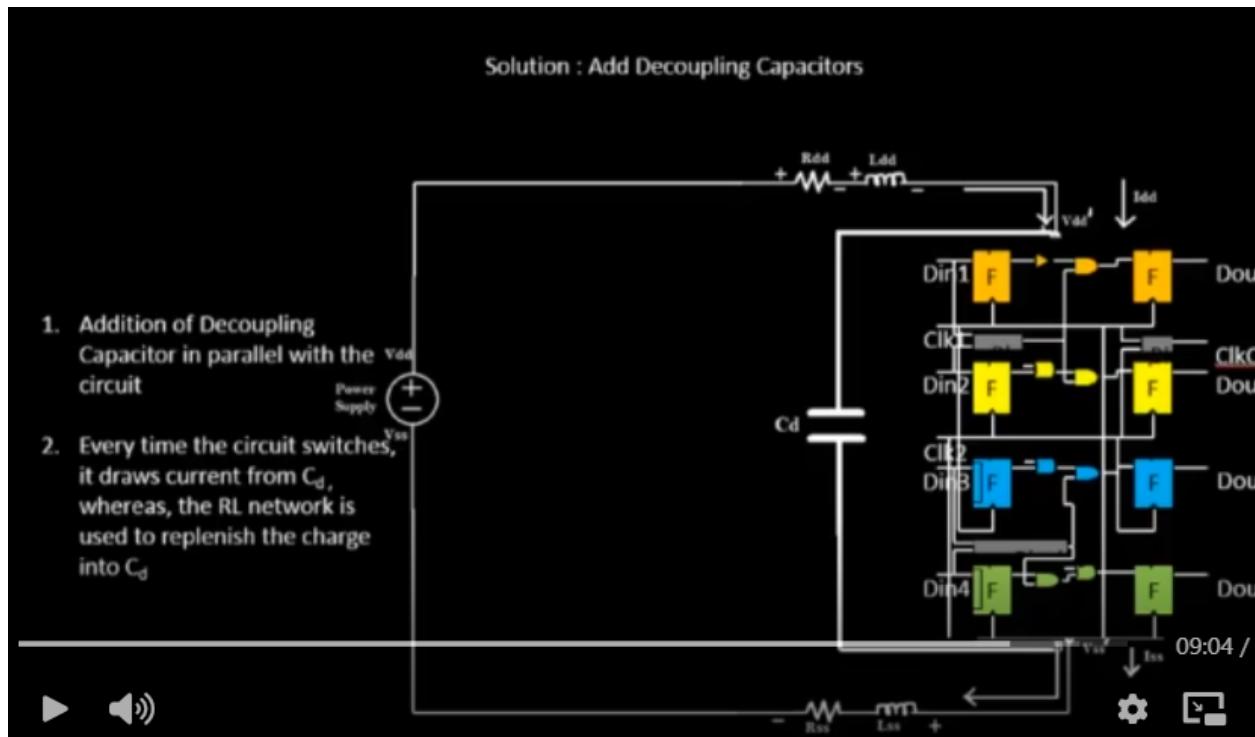
Noise Margin Summary



Noise induced bump characteristics at different noise margin levels

For any signal to be considered as logic '0' and logic '1', it should be in the NM_L and NM_H ranges, respectively

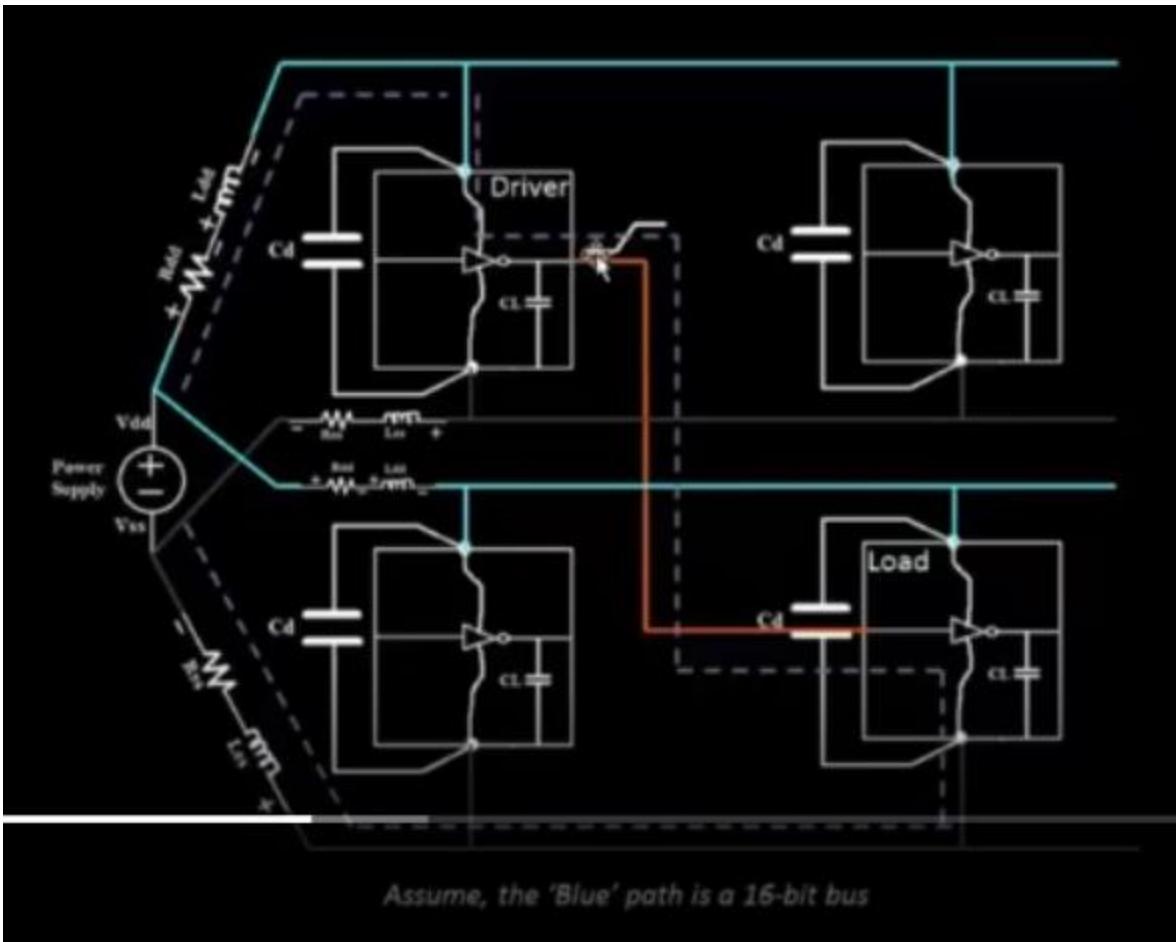
De coupling capacitors:

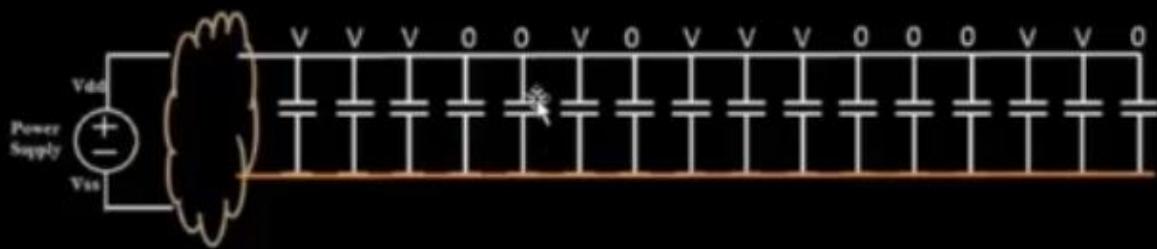


3) Surround pre-placed cells with Decoupling Capacitors



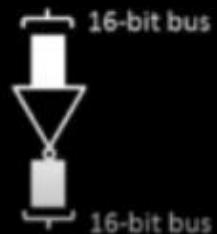
Global communication:



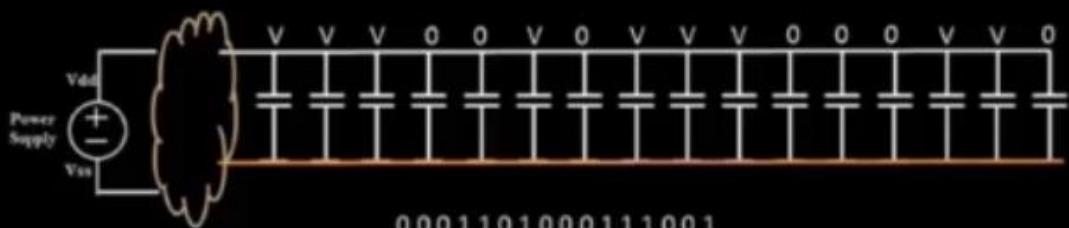


Now, Let's say the output of 16 - bit bus, is connected to an inverter

1110010111000110

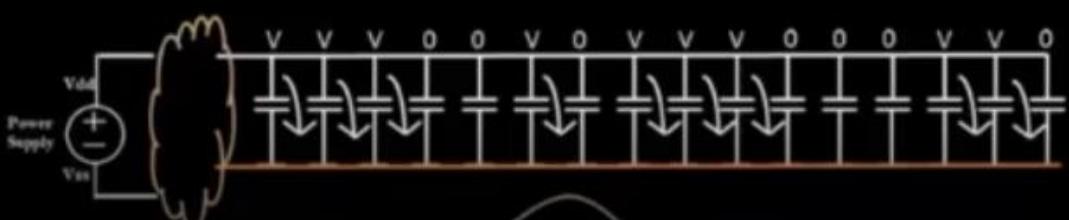


0001101000111001

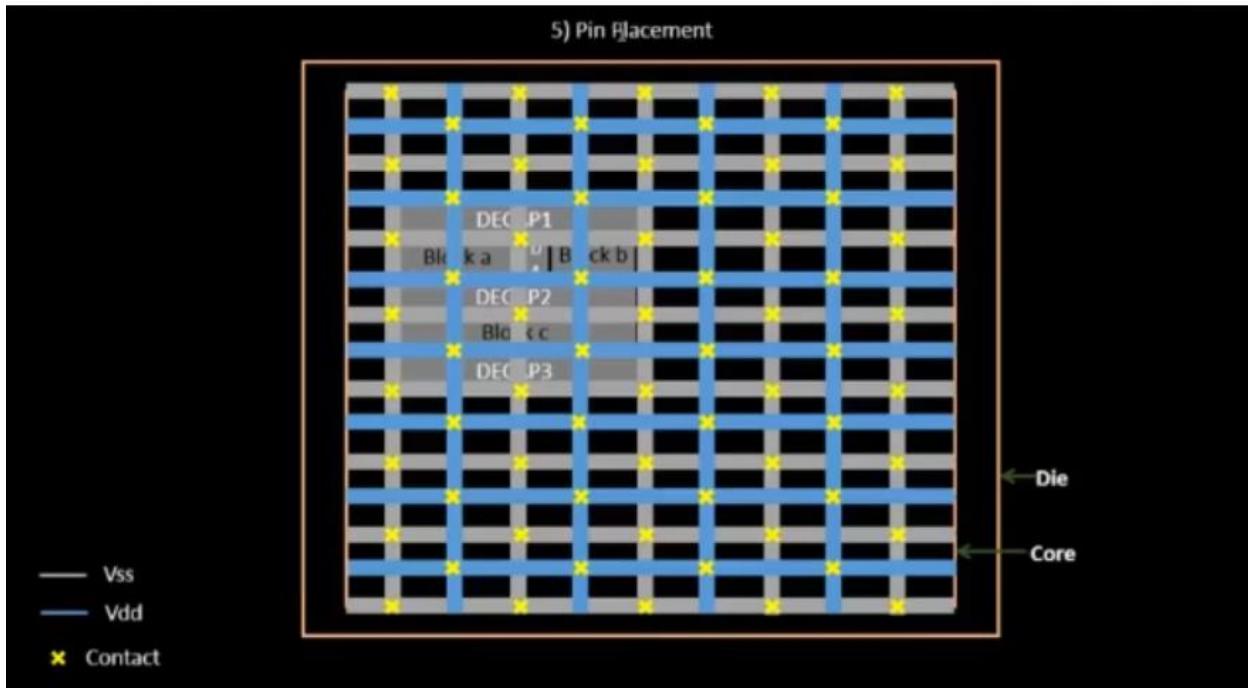
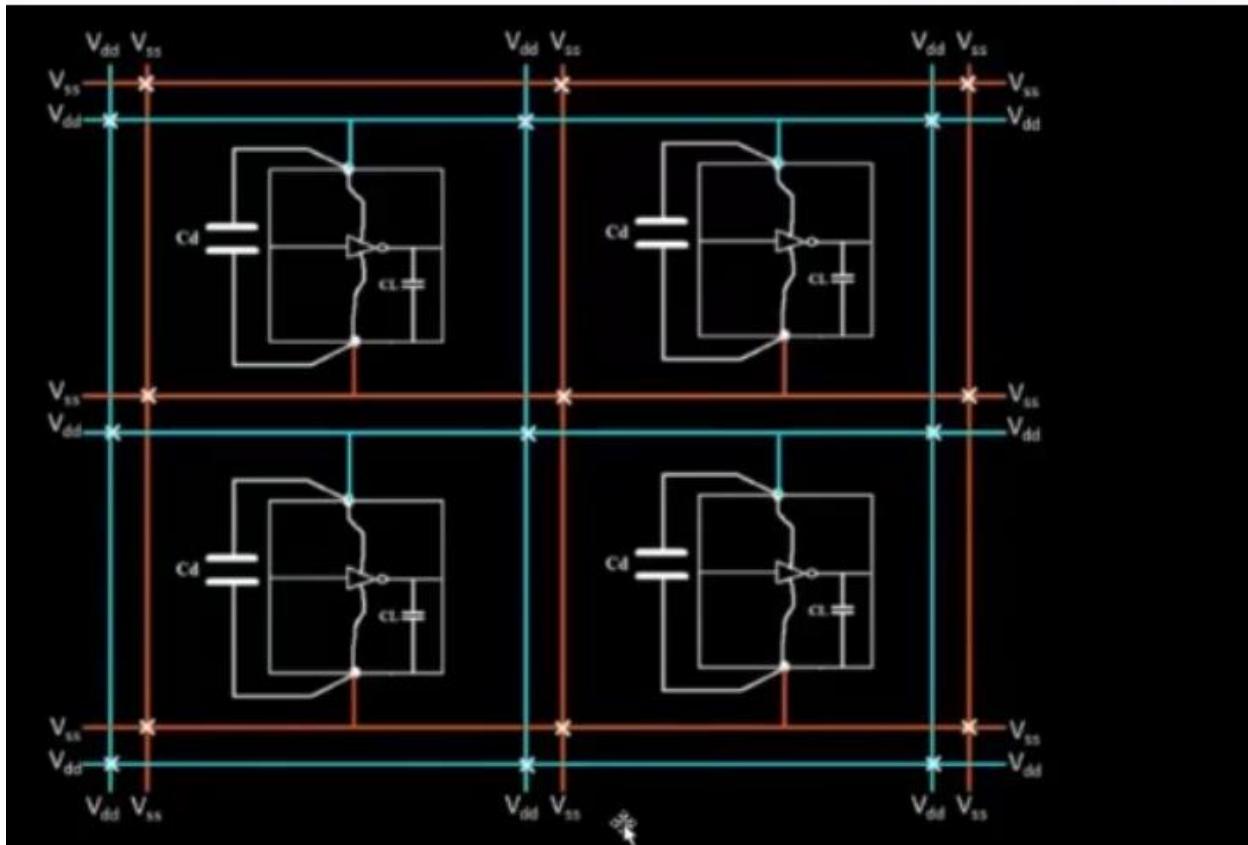


What does this mean?

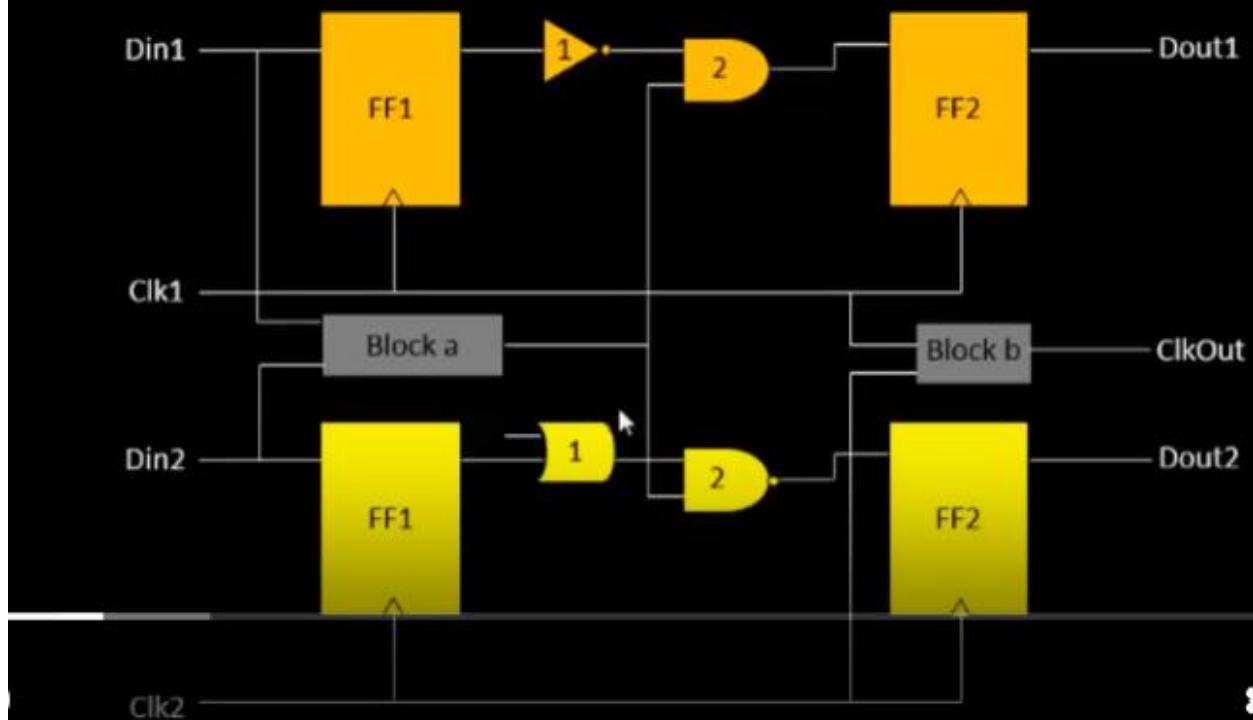
This means, all capacitors which were charged to 'V' volts will have to discharge to '0' volts through single 'Ground' tap point. This will cause a bump in 'Ground' tap point.



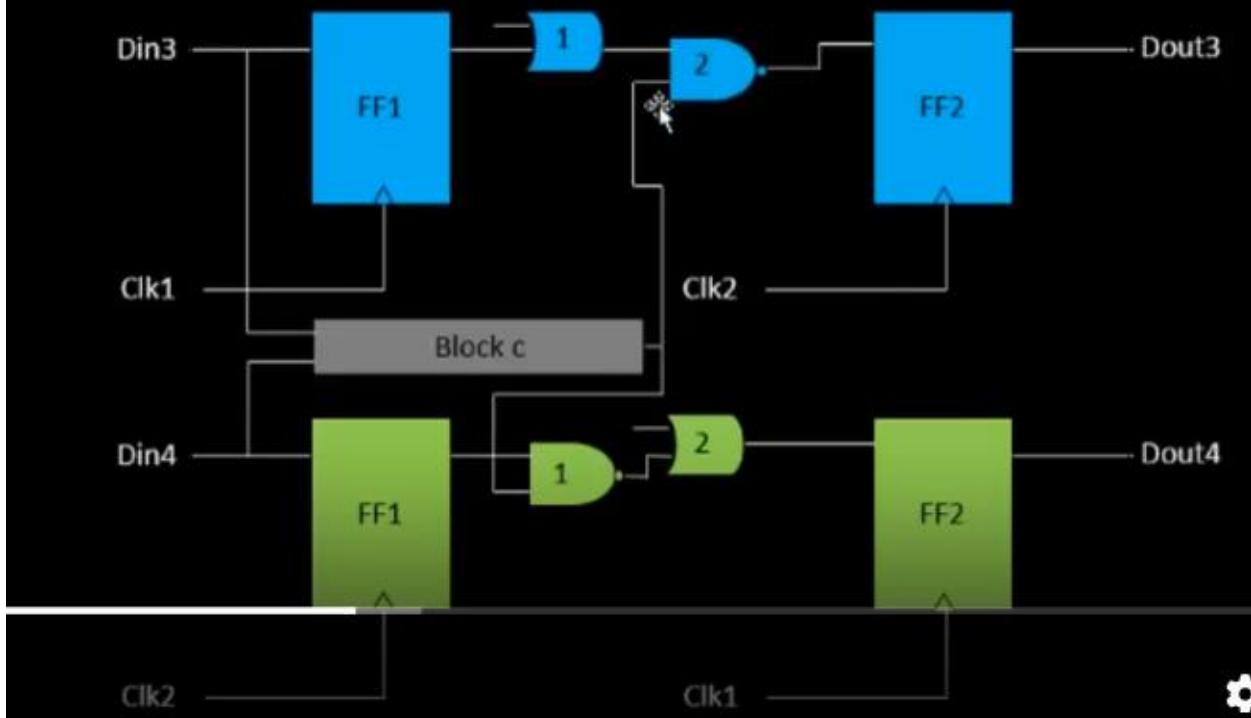
Ground Bounce

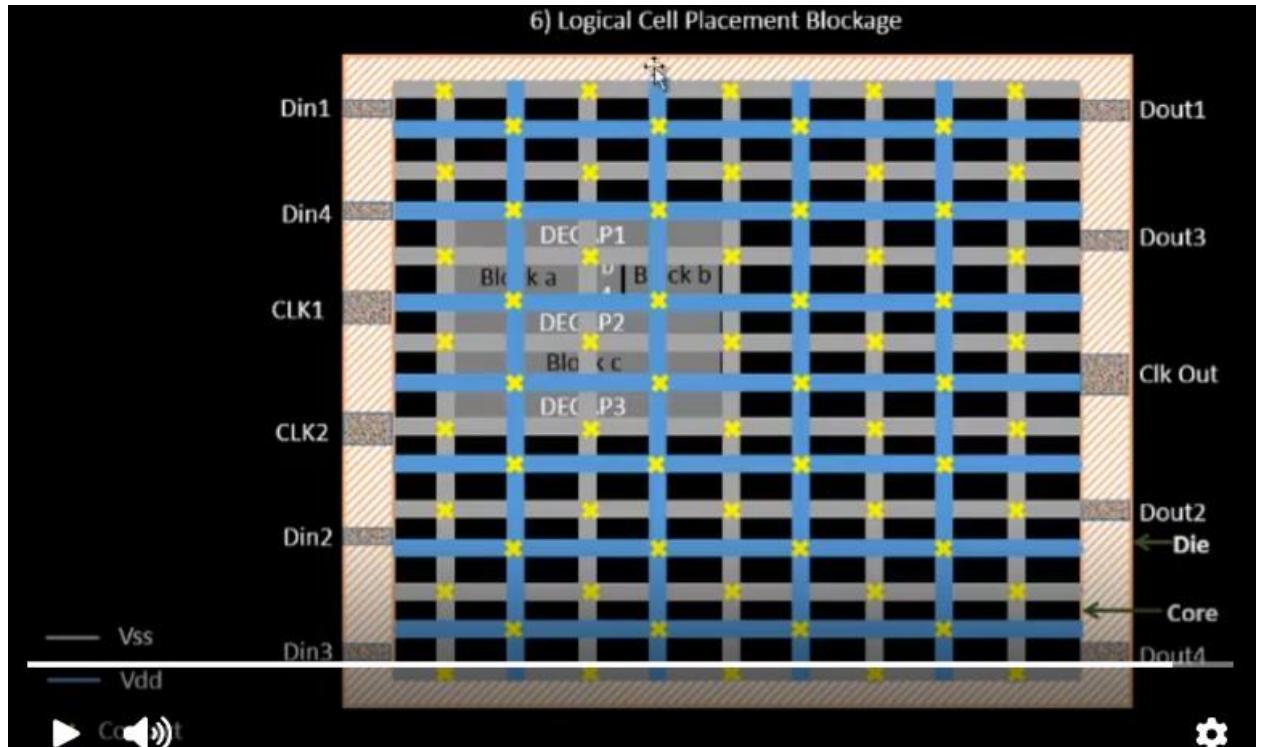
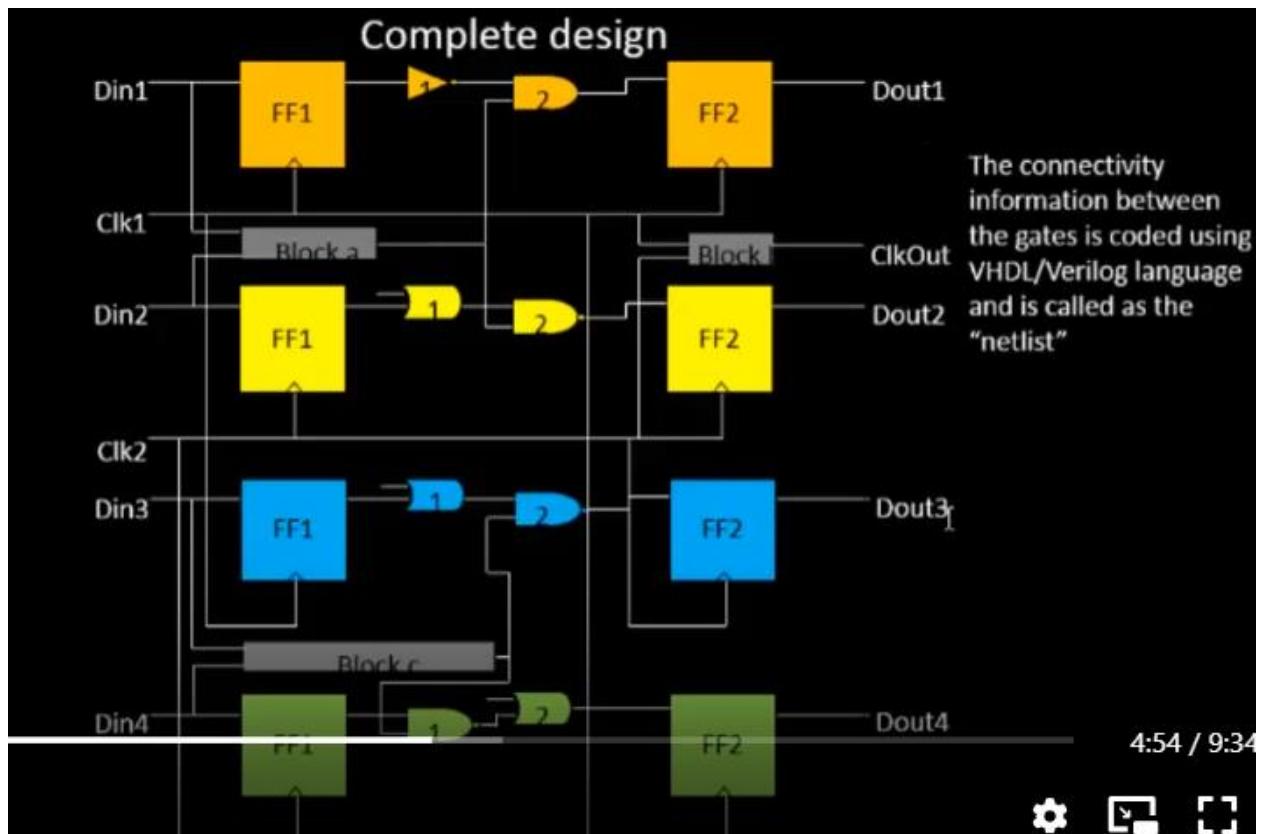


Lets take below design for eg. that needs to implemente



Lets take below design for eg. that needs to implemented





LAB 2:

```
File Edit View Search Terminal Help
WARNING PSM-0030] Vsrc location at (285.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 103.880um).
WARNING PSM-0030] Vsrc location at (425.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 103.880um).
WARNING PSM-0030] Vsrc location at (565.520um, 150.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 103.880um).
WARNING PSM-0030] Vsrc location at (565.520um, 290.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 257.060um).
WARNING PSM-0030] Vsrc location at (5.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (5.00um, 410.240um).
WARNING PSM-0030] Vsrc location at (145.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (145.800um, 410.240um).
WARNING PSM-0030] Vsrc location at (285.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 410.240um).
WARNING PSM-0030] Vsrc location at (425.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 410.240um).
WARNING PSM-0030] Vsrc location at (565.520um, 430.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 410.240um).
WARNING PSM-0030] Vsrc location at (285.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (286.200um, 563.420um).
WARNING PSM-0030] Vsrc location at (425.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (426.600um, 563.420um).
WARNING PSM-0030] Vsrc location at (565.520um, 570.880um) and size =10.000um, is not located on a power stripe. Moving to closest stripe at (567.000um, 563.420um).
INFO PSM-0031] Number of nodes on net VGND = 19223.
INFO PSM-0037] G matrix created sucessfully.
INFO PSM-0040] Connection between all PDN nodes established in net VGND.
INFO]: PDN generation was successful.
INFO]: Changing layout from /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/floorplan/picorv32a.floorplan.def to /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/floorplan/7-pdn.def
```



vsdworkshop25 [Running] - Oracle VirtualBox

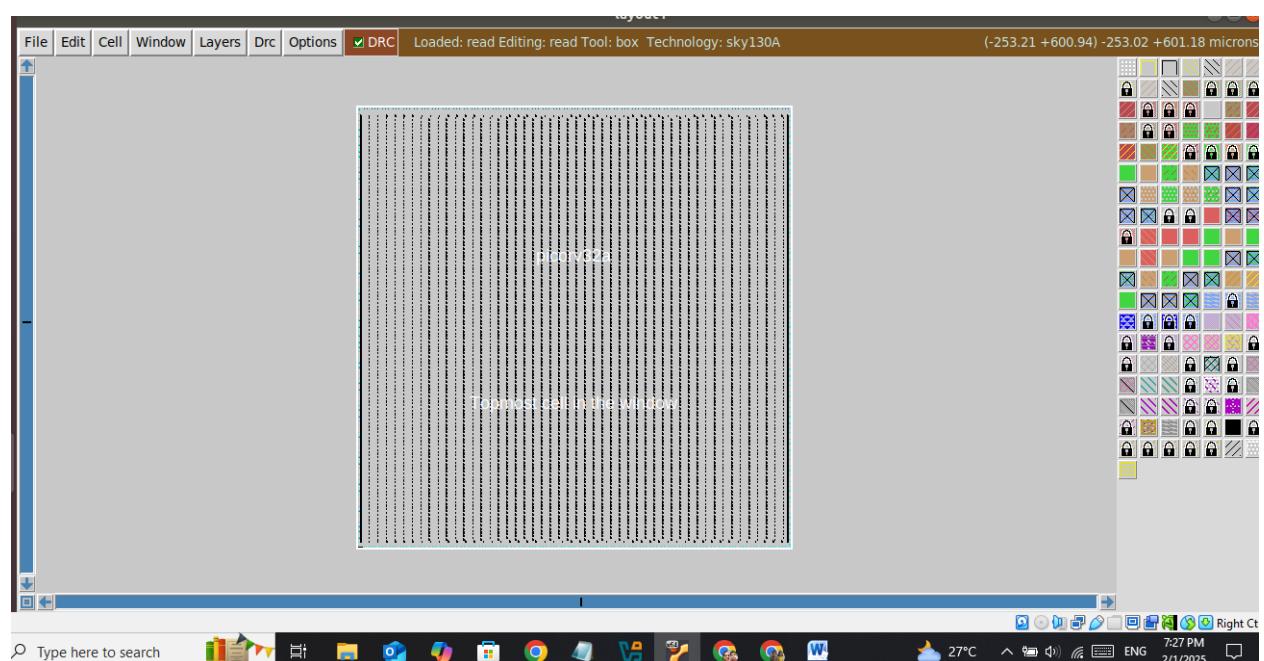
File Machine View Input Devices Help

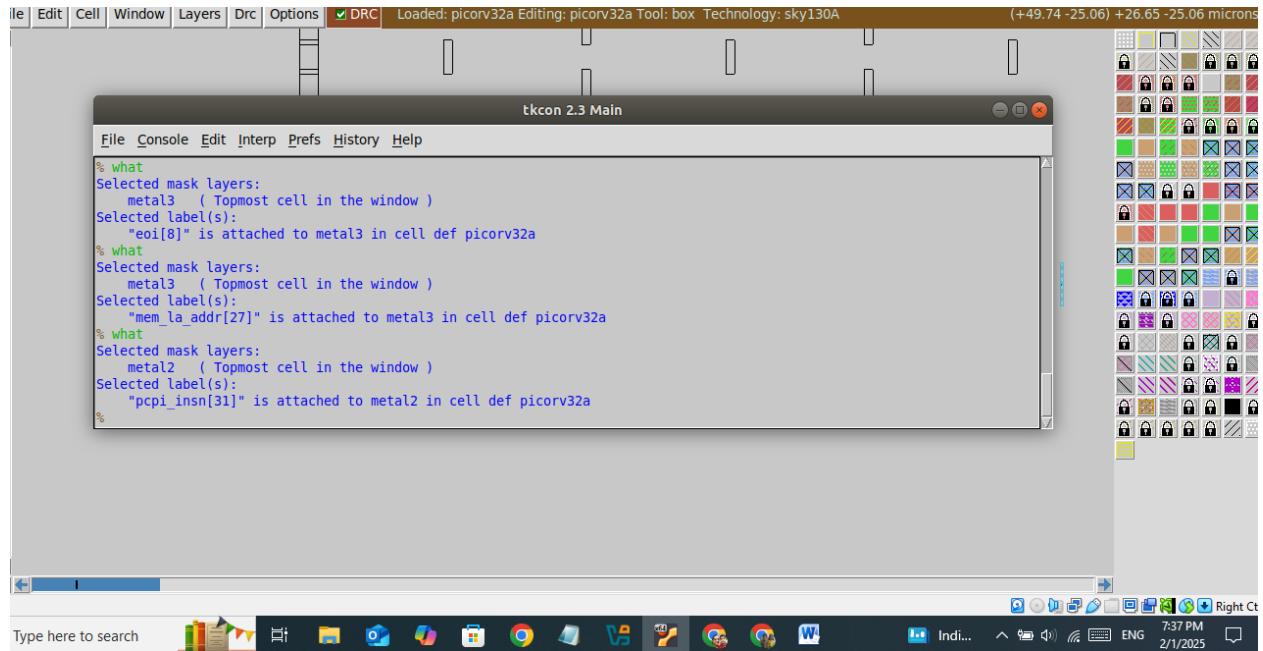
Sat 18:57

Activities Terminal

```
vsduser@vsdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/designs/picorv32a/runs/01-02_12-50/logs/floorplan$ ls -lrt
total 40
-rw-r--r-- 1 vsduser vsduser 1204 Feb 1 18:43 3-verilog2def.openroad.log
-rw-r--r-- 1 vsduser vsduser 12 Feb 1 18:43 3-verilog2def_openroad_runtime.txt
-rw-r--r-- 1 vsduser vsduser 1043 Feb 1 18:43 4-ioPlacer.log
-rw-r--r-- 1 vsduser vsduser 12 Feb 1 18:43 4-ioPlacer_runtime.txt
-rw-r--r-- 1 vsduser vsduser 1198 Feb 1 18:43 5-tapcell.log
-rw-r--r-- 1 vsduser vsduser 12 Feb 1 18:43 5-tapcell_runtime.txt
-rw-r--r-- 1 vsduser vsduser 8950 Feb 1 18:44 7-pdn.log
-rw-r--r-- 1 vsduser vsduser 12 Feb 1 18:44 7-pdn_runtime.txt
vsduser@vsdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/designs/picorv32a/runs/01-02_12-50/logs/floorplan$ vi 4-ioPlacer.log
vsduser@vsdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/designs/picorv32a/runs/01-02_12-50/logs/floorplan$ cat 4-ioPlacer.log
OpenROAD 0.9.0 1415572a73
This program is licensed under the BSD-3 license. See the LICENSE file for details.
Components of this program may be licensed under more restrictive licenses which must be honored.
Notice 0: Reading LEF file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/merged.lef
Notice 0:     Created 13 technology layers
Notice 0:     Created 25 technology vias
Notice 0:     Created 440 library cells
Notice 0: Finished LEF file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/merged.lef
Notice 0:
Reading DEF file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/floorplan/3-verilog2def_openroad.def
Notice 0: Design: picorv32a
Notice 0:     Created 409 pins.
Notice 0:     Created 14876 components and 115597 component-terminals.
Notice 0:     Created 14978 nets and 56051 connections.
Notice 0: Finished DEF file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/floorplan/3-verilog2def_openroad.def
#Macro blocks found: 0
Using 5u default boundaries offset
Random pin placement
vsduser@vsdsquadron:~/Desktop/work/tools/openlane_working_dir/openlane/designs/picorv32a/runs/01-02_12-50/logs/floorplan$
```

Type here to search                      

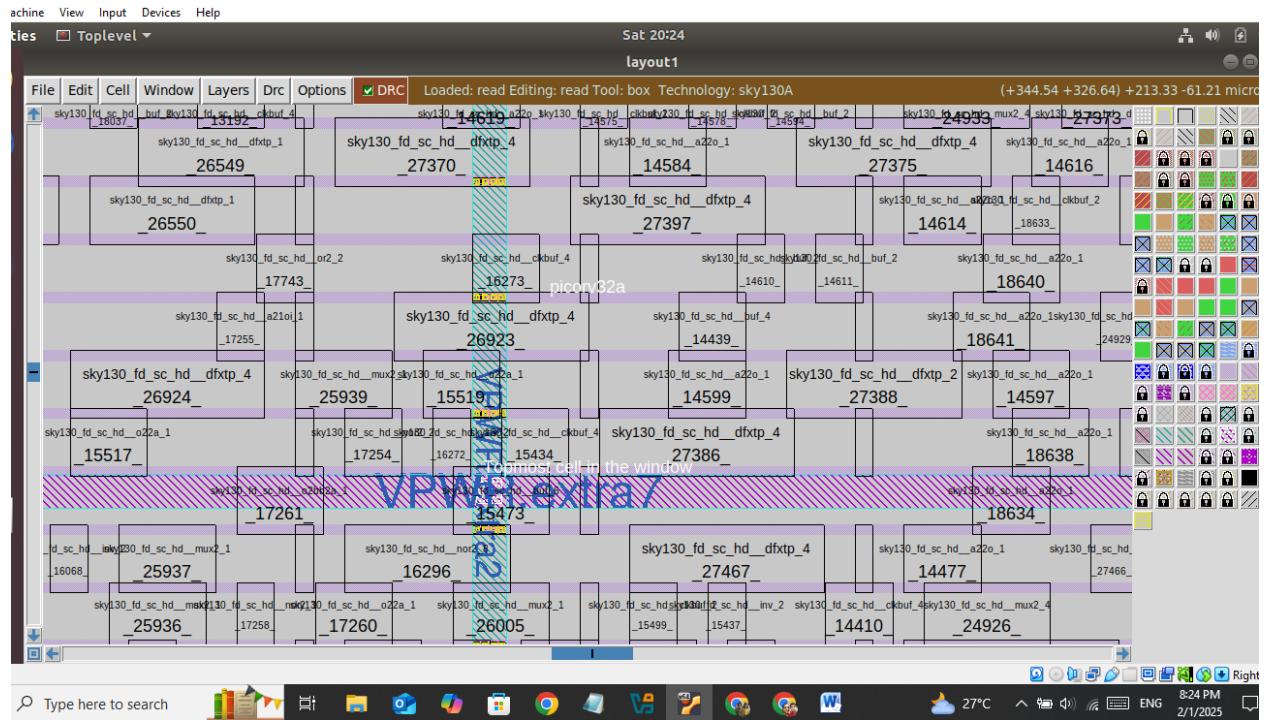




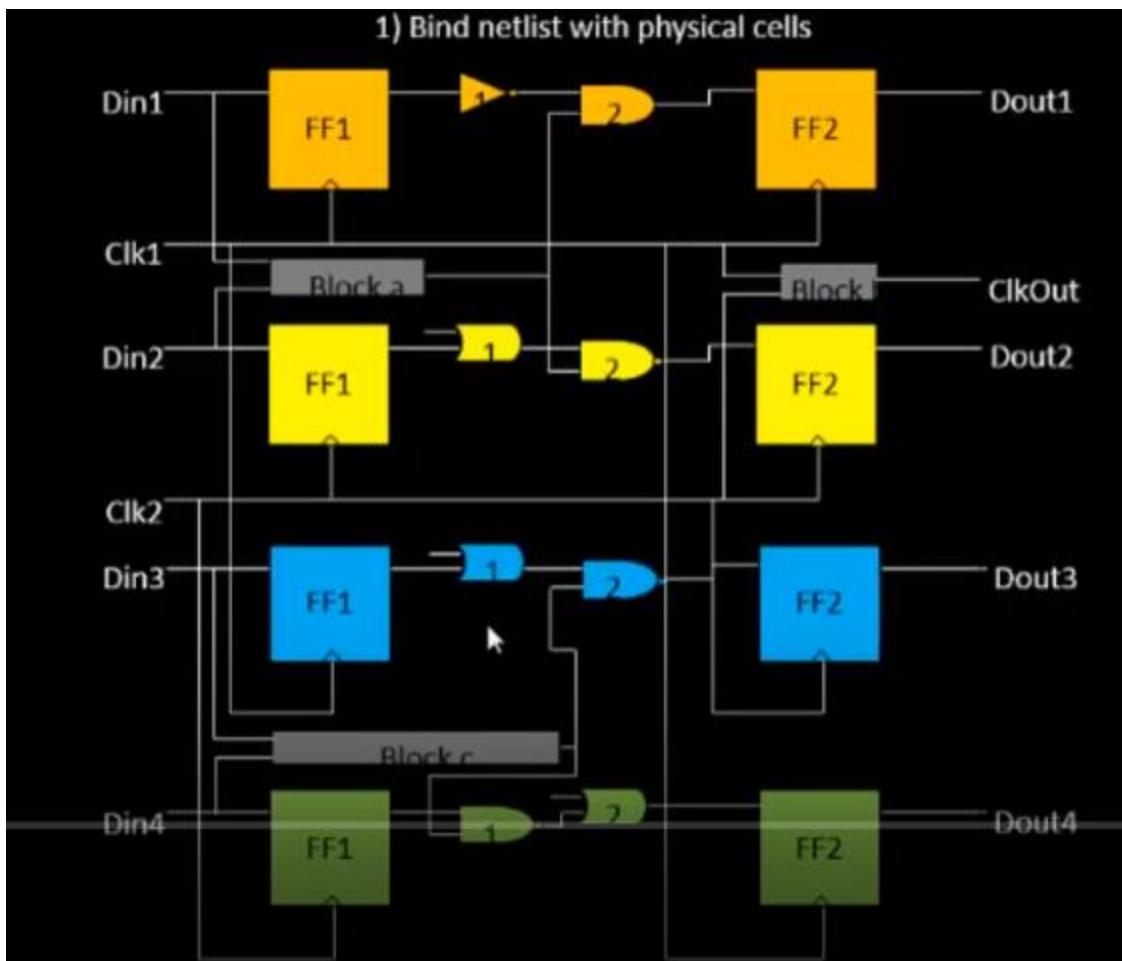
```
Sat 20:18
vsduser@vsdsquadron: ~/Desktop/work/tools/openlane_working_dir/openlane
File Edit View Search Terminal Help
utilization padded      55 %
rows                   238
row height             2.7 u

Placement Analysis
-----
total displacement      0.0 u
average displacement    0.0 u
max displacement        0.0 u
original HPWL          766080.0 u
legalized HPWL          779196.5 u
delta HPWL              2 %

[INFO DPL-0020] Mirrored 6193 instances
[INFO DPL-0021] HPWL before           779196.5 u
[INFO DPL-0022] HPWL after            766080.0 u
[INFO DPL-0023] HPWL delta           -1.7 %
[INFO]: Changing layout from /openLANE_flow/designs/picorv32a/runs/01-02_12-50/tmp/placement/8-resizer.def to /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def
[INFO]: Changing layout from /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def to /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def
[INFO]: Taking a Screenshot of the Layout Using Klayout...
[INFO]: current step index: 12
Using Techfile: /home/vsduser/Desktop/work/tools/openlane_working_dir/pdk/sky130A/libs.tech/klayout/sky130A.lyt
Using layout file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def
[INFO] Reading tech file: /home/vsduser/Desktop/work/tools/openlane_working_dir/pdk/sky130A/libs.tech/klayout/sky130A.lyt
[INFO] Reading Layout file: /openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def
[INFO] Writing out PNG screenshot '/openLANE_flow/designs/picorv32a/runs/01-02_12-50/results/placement/picorv32a.placement.def.png'
Done
[INFO]: Screenshot taken.
%
%
```

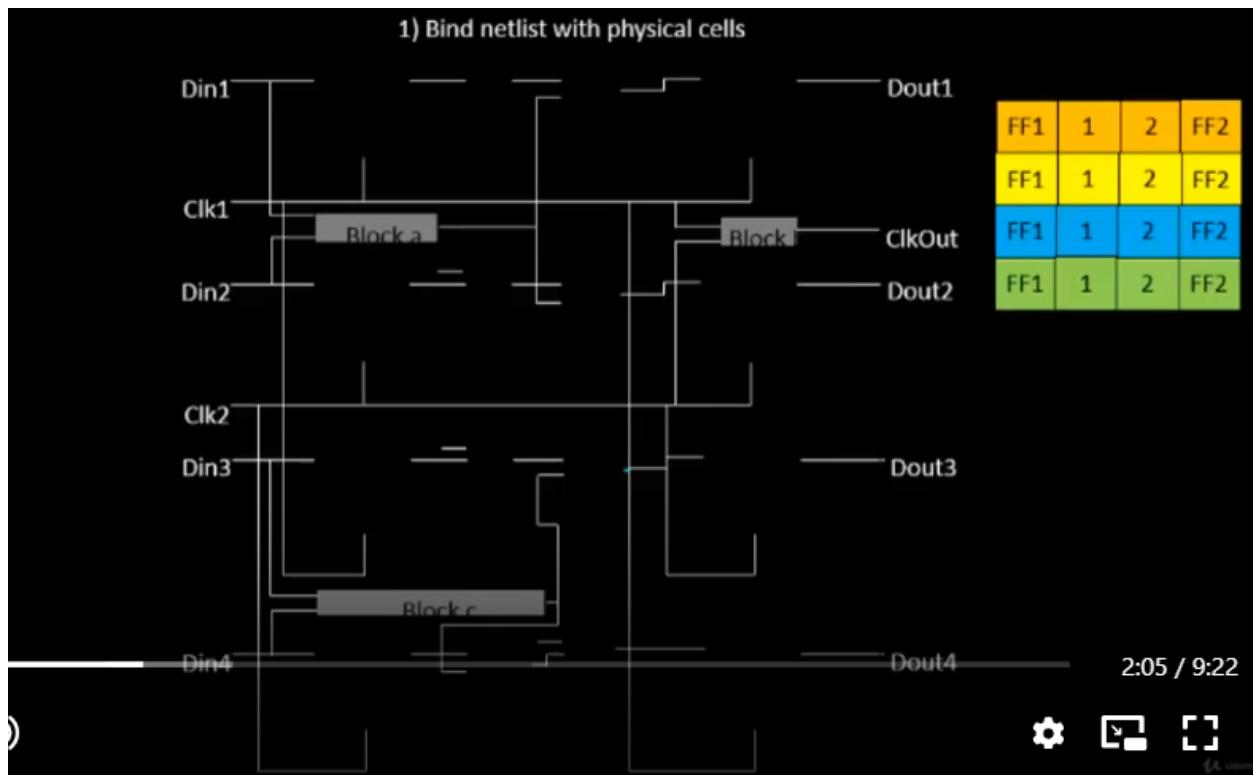


Placement and routing:

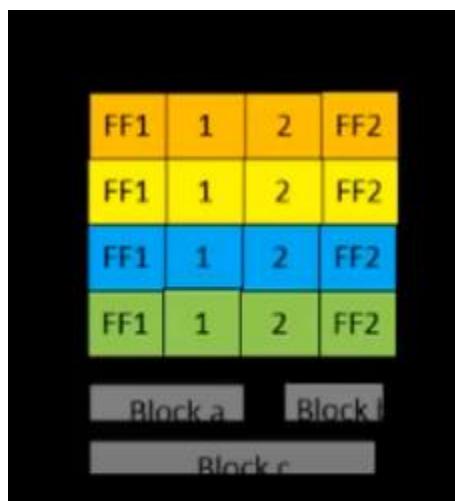


The shape of the gates represents the function of the gates. But in reality these don't have shapes. They just have a box.

The each component has been given the net list with proper shape of width and height.



Now lets remove the wires.



These things are present in a shelf called library.

A library is a place where we can find the timing information of the gate. Library types:

- Shapes and sizes
- Delay information

It will have width and height of each cell and required conditions for the flip flop to emit the output.

It also has various shapes of the various gates. There will be another option in the library.



The difference between first library and second library is that since this library has bigger and gates it will have least resistance it will be faster. Even we have bigger than this.

FF1	1	2	FF2
FF1	1	2	FF2
FF1	1	2	FF2
FF1	1	2	FF2

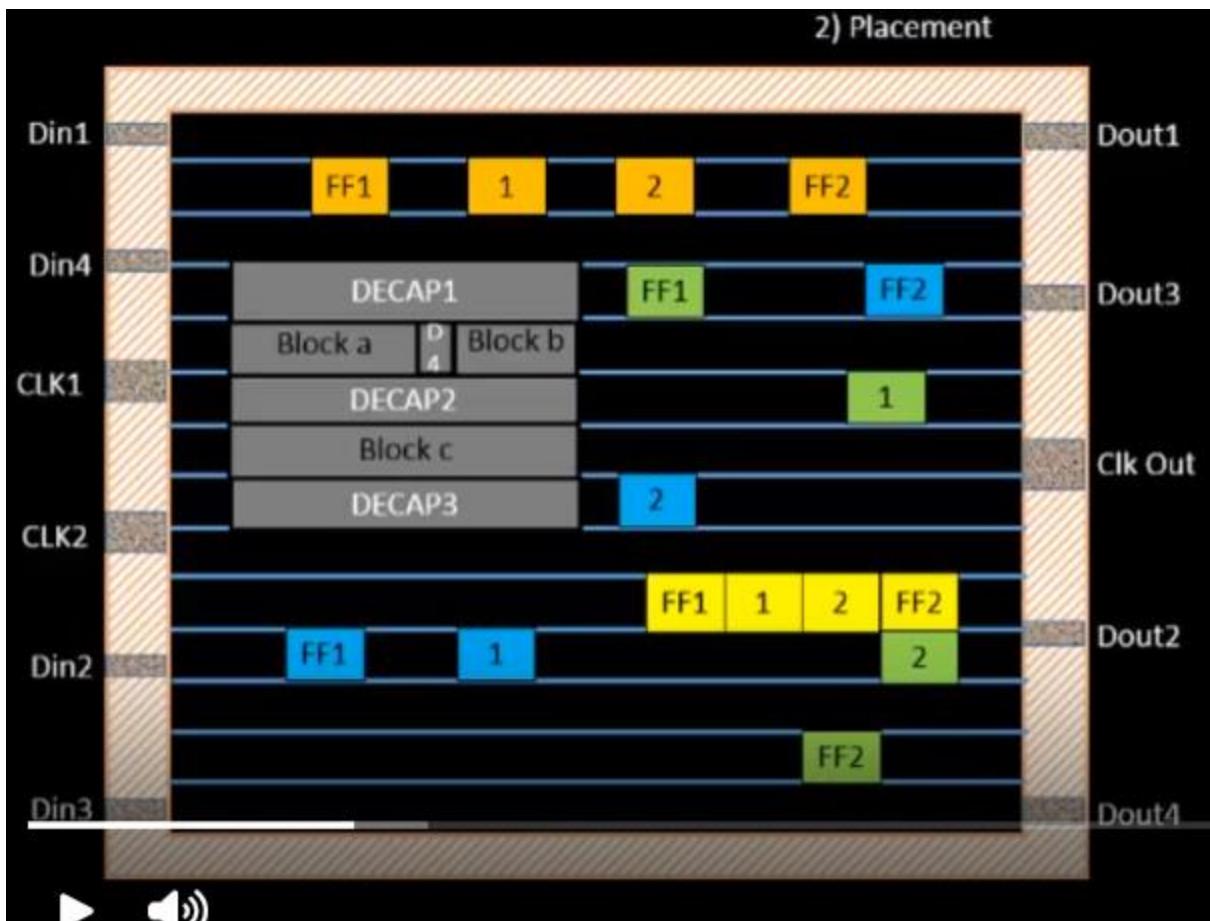
This is even faster. Basically the library has got different flavors of the cells. We can pick up what we want.

Library consists of cells ,delay information etc.

Once we have given proper shapes to the gates the next step is to take those particular shapes and place them in to the floorplan. We have a proper floorplan. We have a particular netlist. We have a shape and size given to the each component. The next step is to place the net list on the particular floorplan. Though we have to collect the connectivity information from the net list but net list does not come into consideration. But we have to take the physical view in the case of physical design.



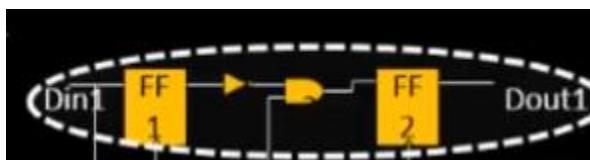
This is the floorplan and the pre-placed cells that we already had. Placement stage makes sure that these locations are not affected. It will take care that there will be no cells placed in that area. Now we will take the connectivity from the net list.

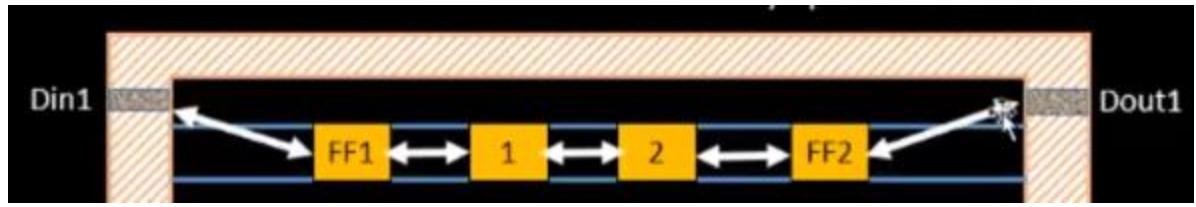


The next stage is the optimize placement.

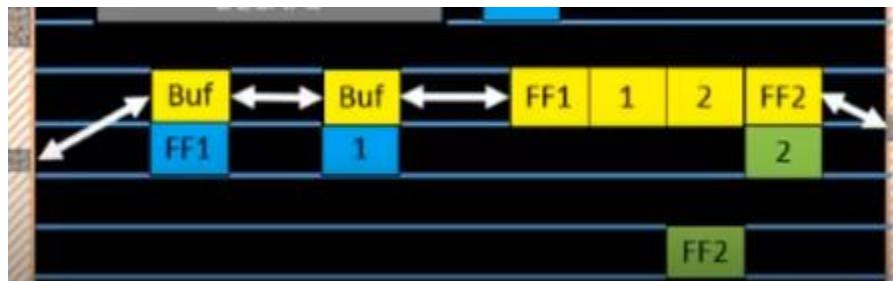
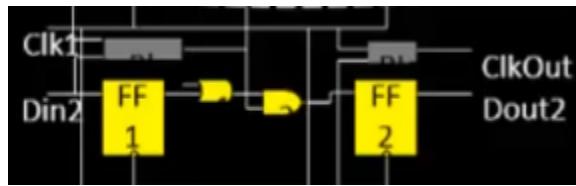
This is the stage where we estimate wire length and capacitance and based on that, insert repeaters.

Now first we will take this.

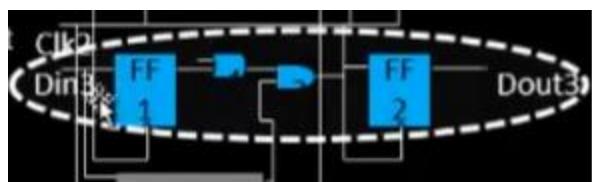


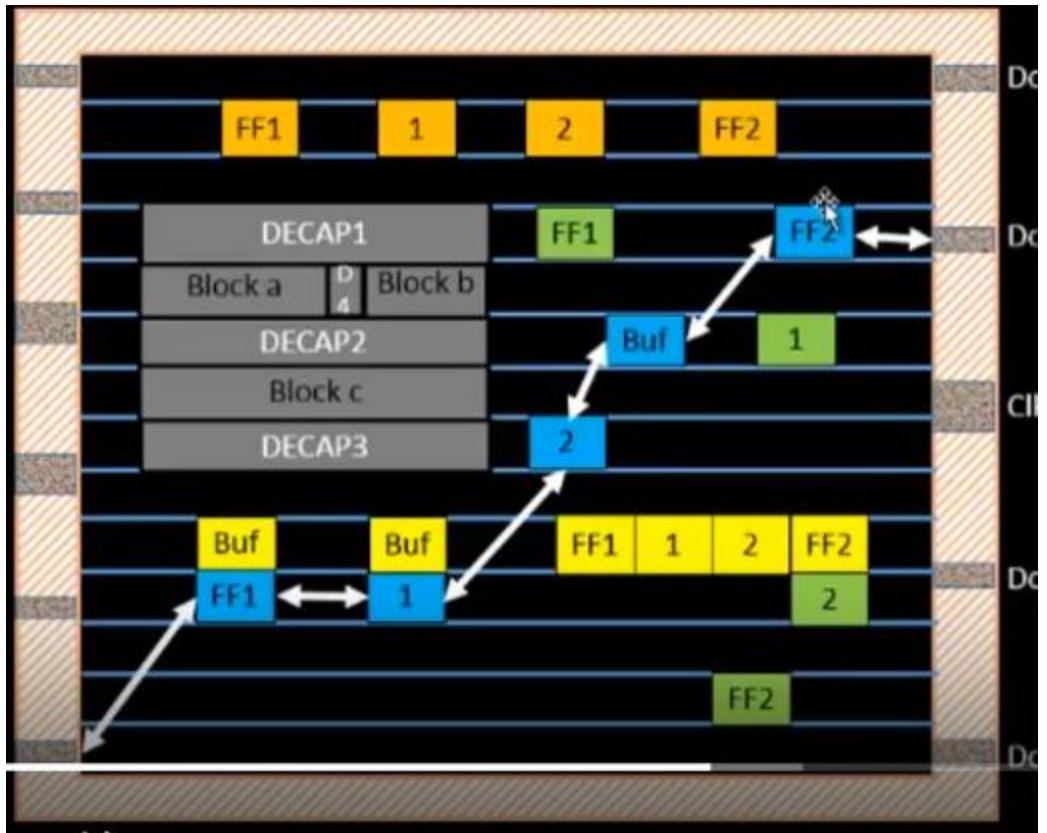


Now we will take this.

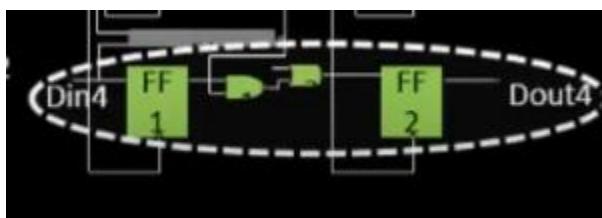


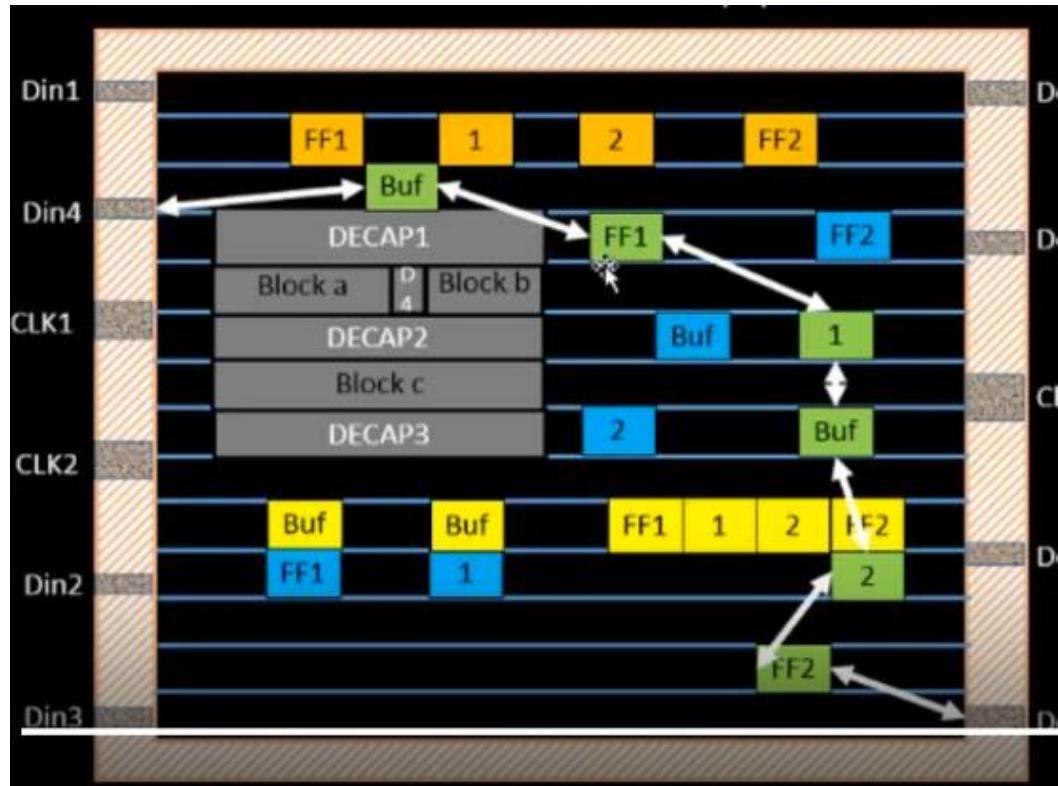
Now we will take this





Now we will take this:



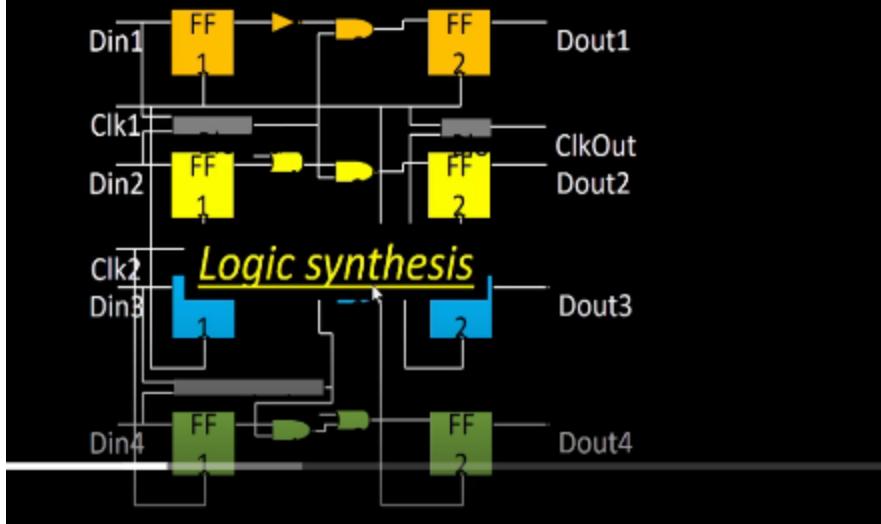


Need for characterization

First step of logic synthesis:

Library characterization

Part 1 – Concepts and Theory – NLDM characterization

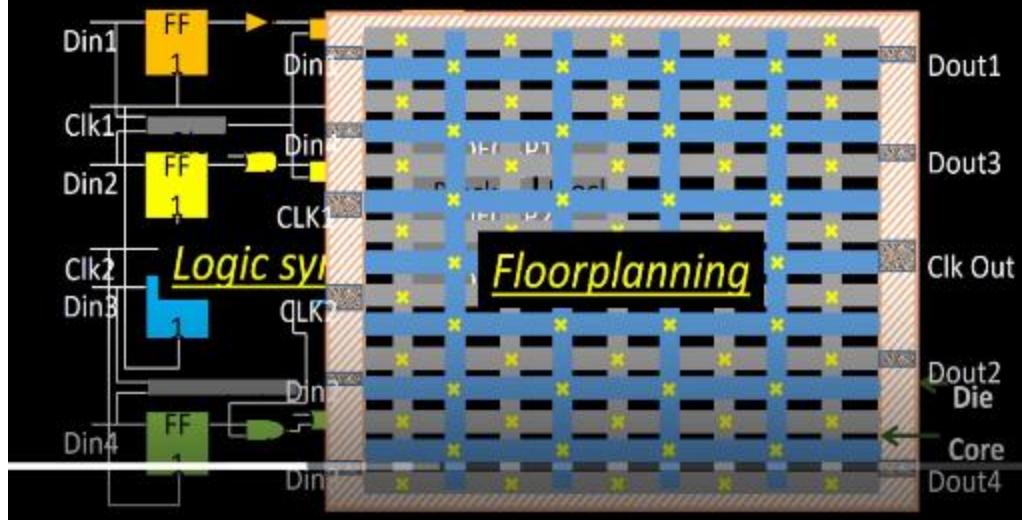


Second step:

Floor planning:

Library characterization and placement

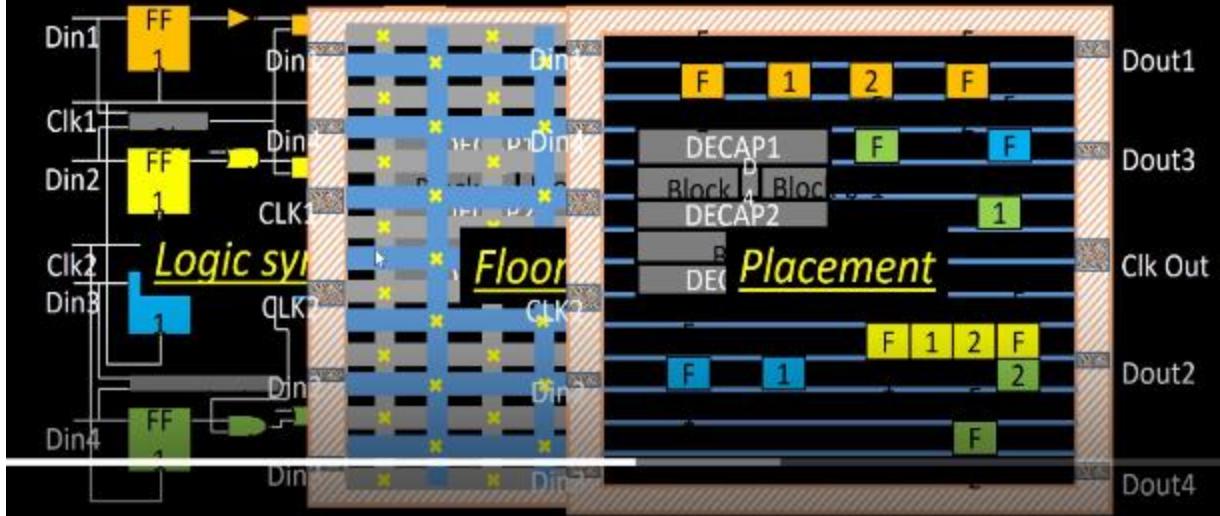
Part 1 – Concepts and Theory – NLDM, CCS timing, and library characterization



Third step:

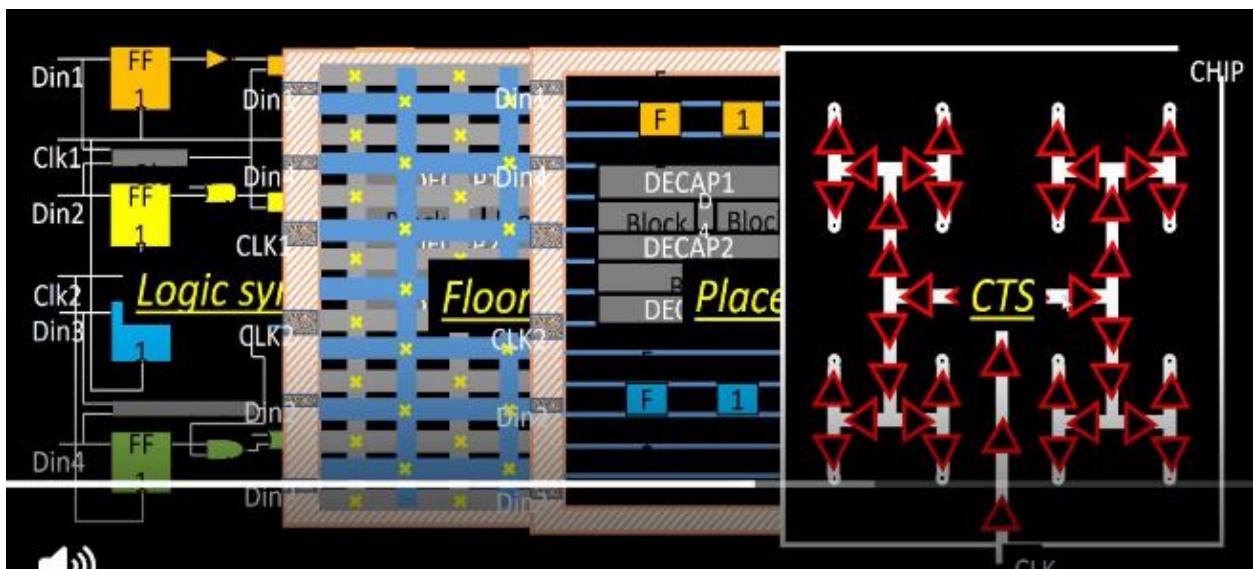
Placement:

Part 1 – Concepts and Theory – NLDM, CCS timing, power characterization



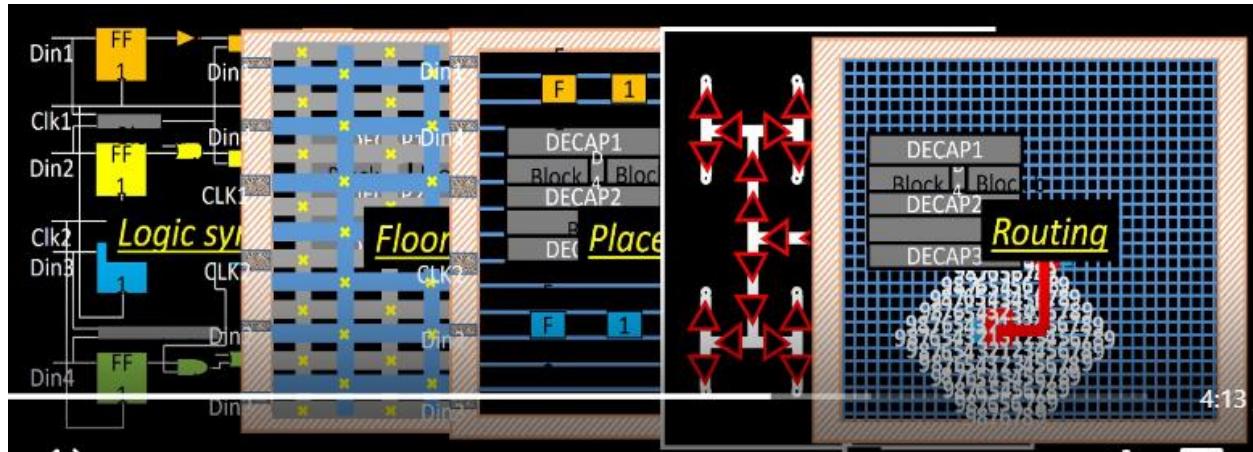
Fourth step:

Clock reset synthesis



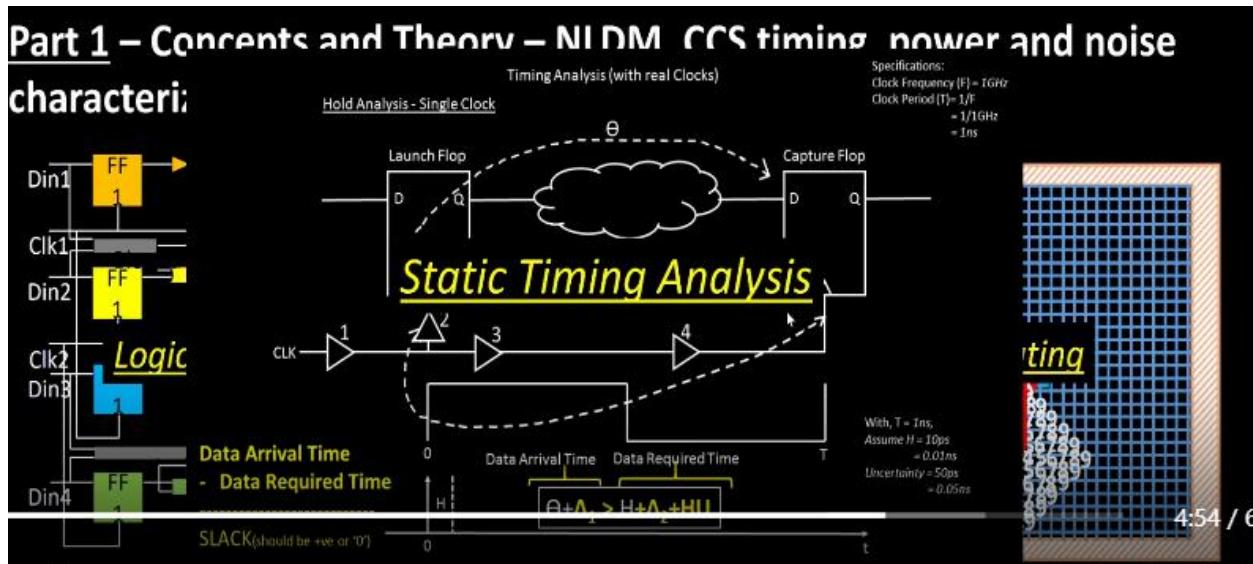
Fifth step:

Routing



Sixth step:

Static time analysis:



One common thing across all stages: “GATES or Cells”

Library characterization and

*One Common Thing across all
“GATES or Cells”*

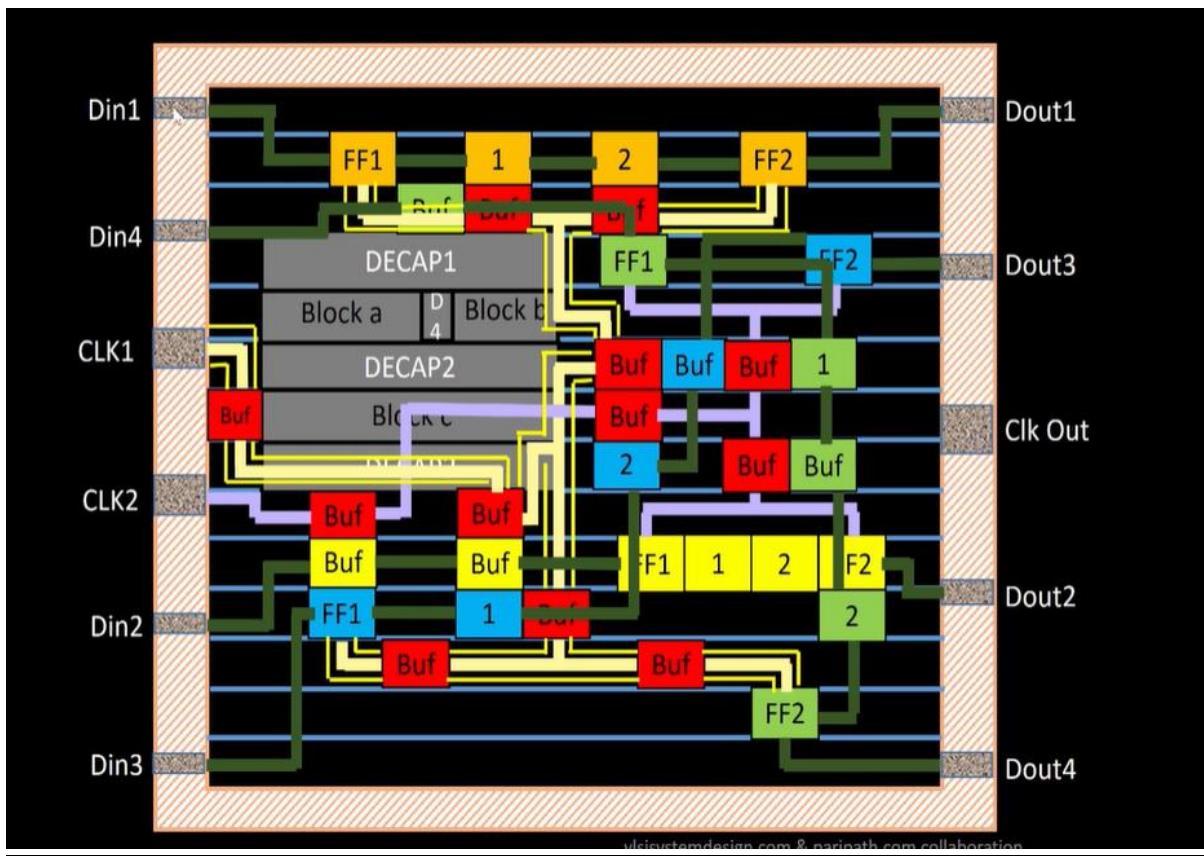
<u>Library</u>
• AND gate
• OR gate
• BUFFER
• INVERTER
• DFF
• LATCH
• ICG
•

visisystemdesign.com & panpath.com collaboration

How do we characterize, design or model these cells?

Cell design flow

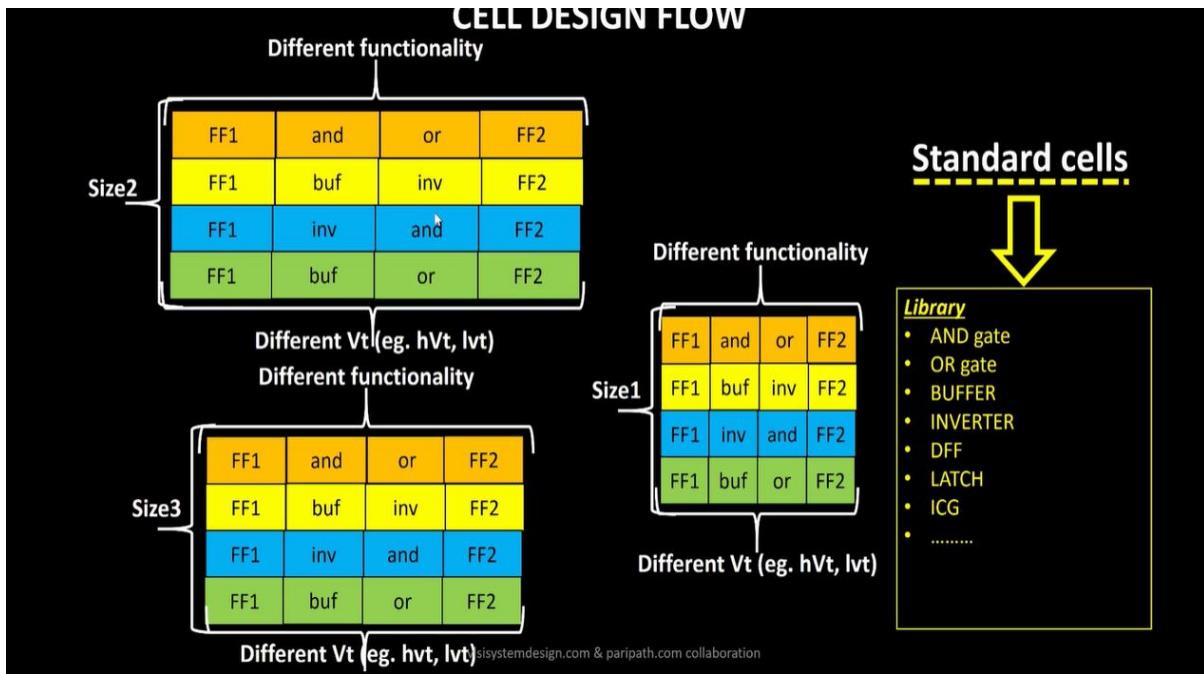
Routed chip

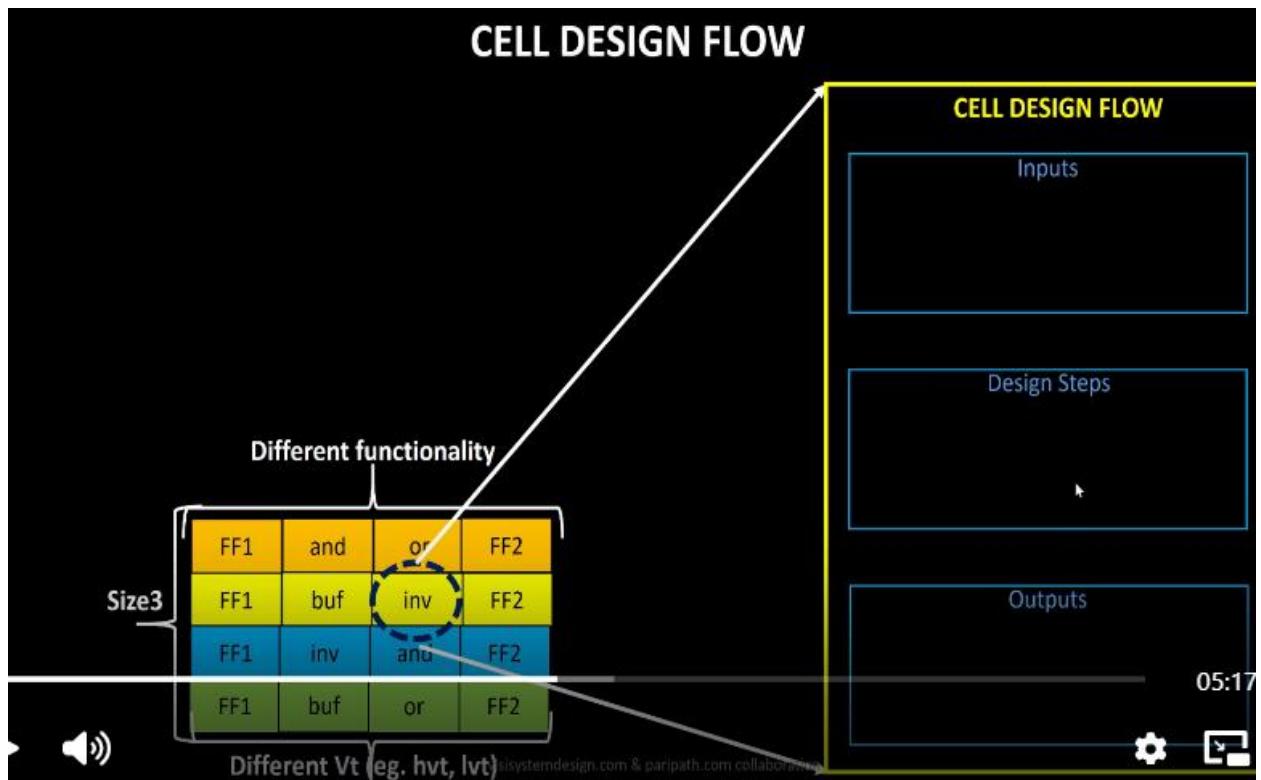


Standard cells.

Buffers, inverters and gates are called as standard cells. Standard cells are placed in libraries. IN an ic design flow all standard cells are present. Library is also a place where we keep all the decap cells, macros, IP's and so on. One of the section of the library is to keep the place the standard cells and gates. Library has got different

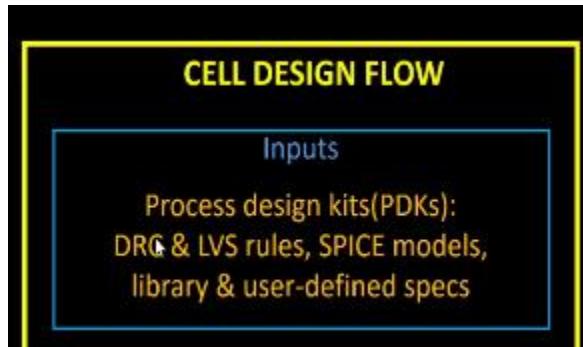
got with different function and cells with different sizes.



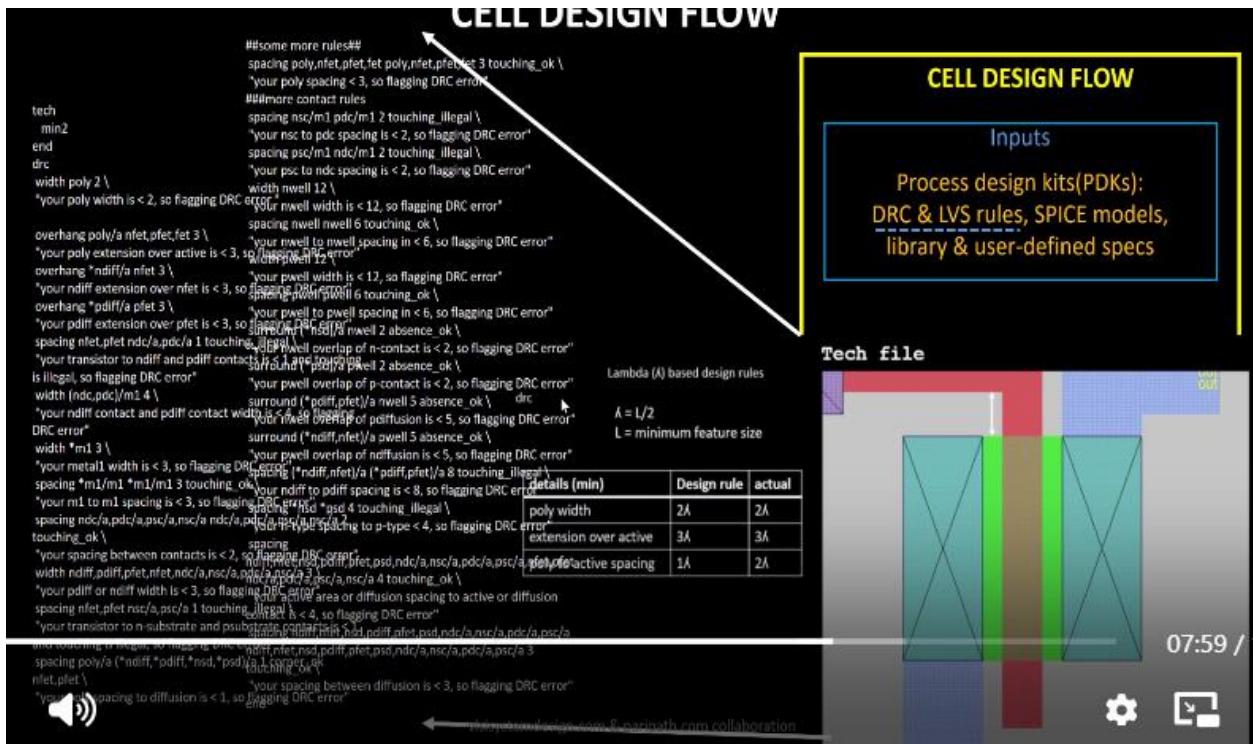


Inverter:

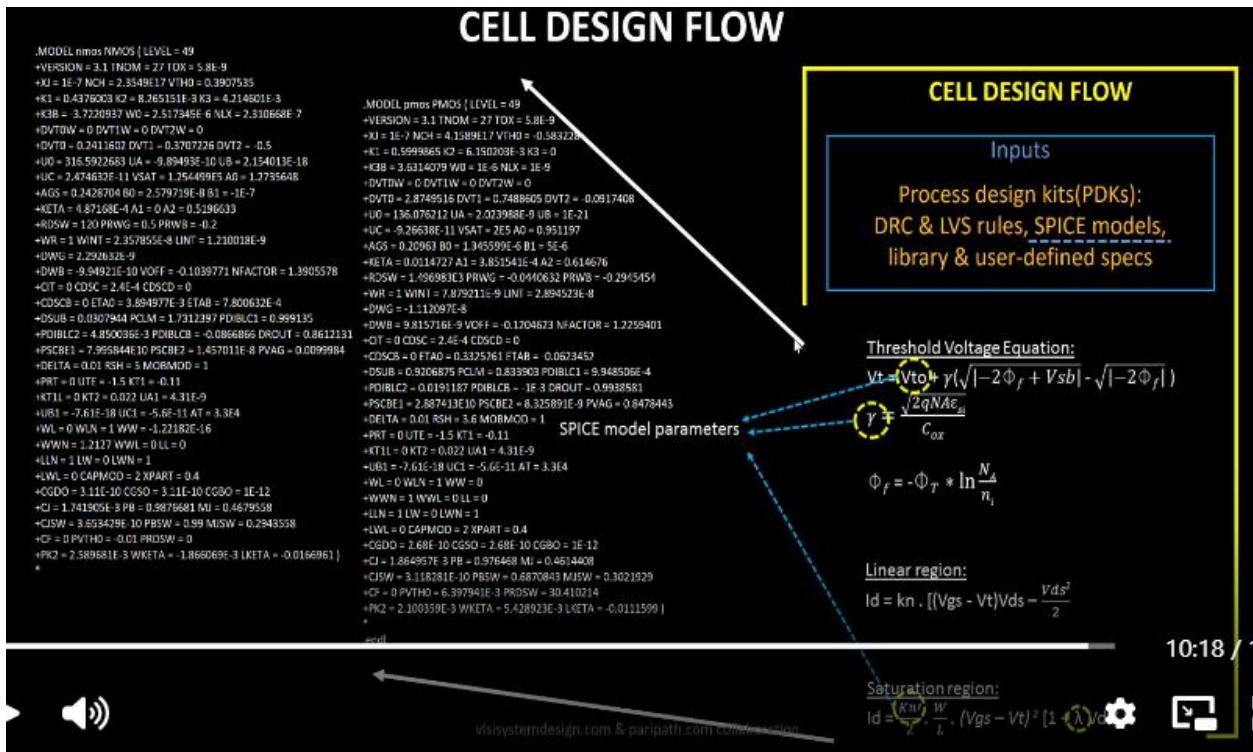
1) Input



a) DRC and LVS rules:

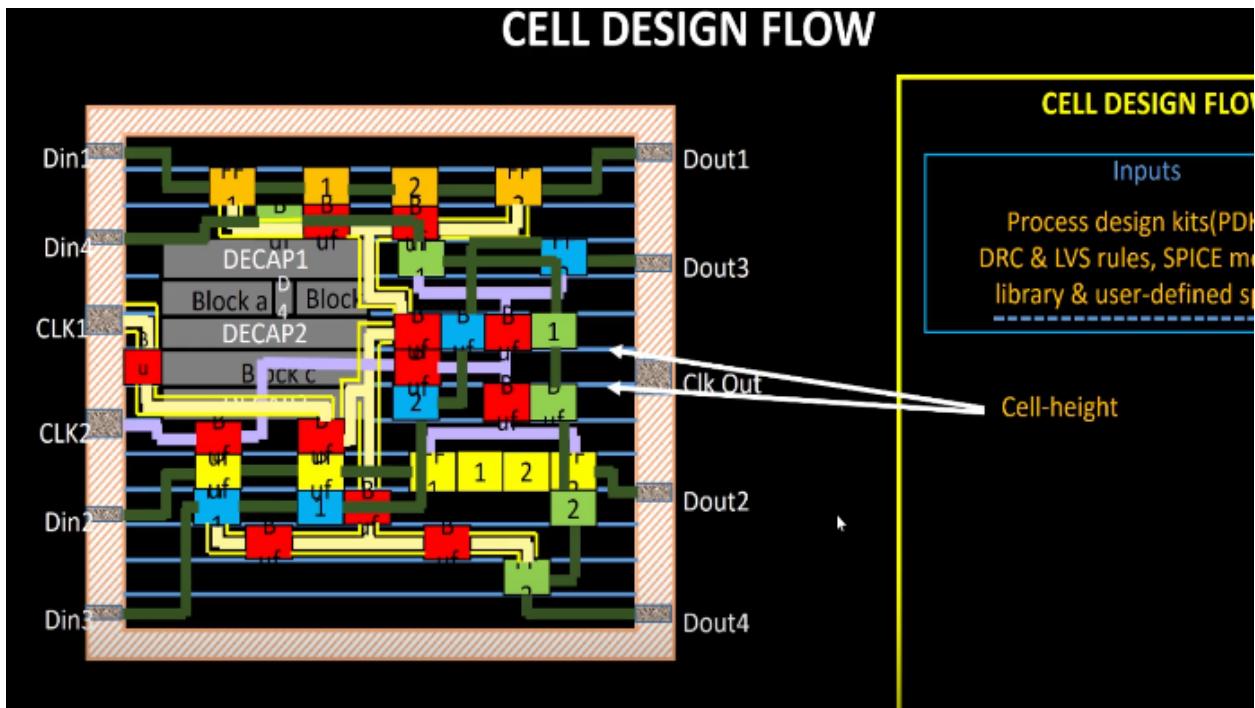


b) Spice models

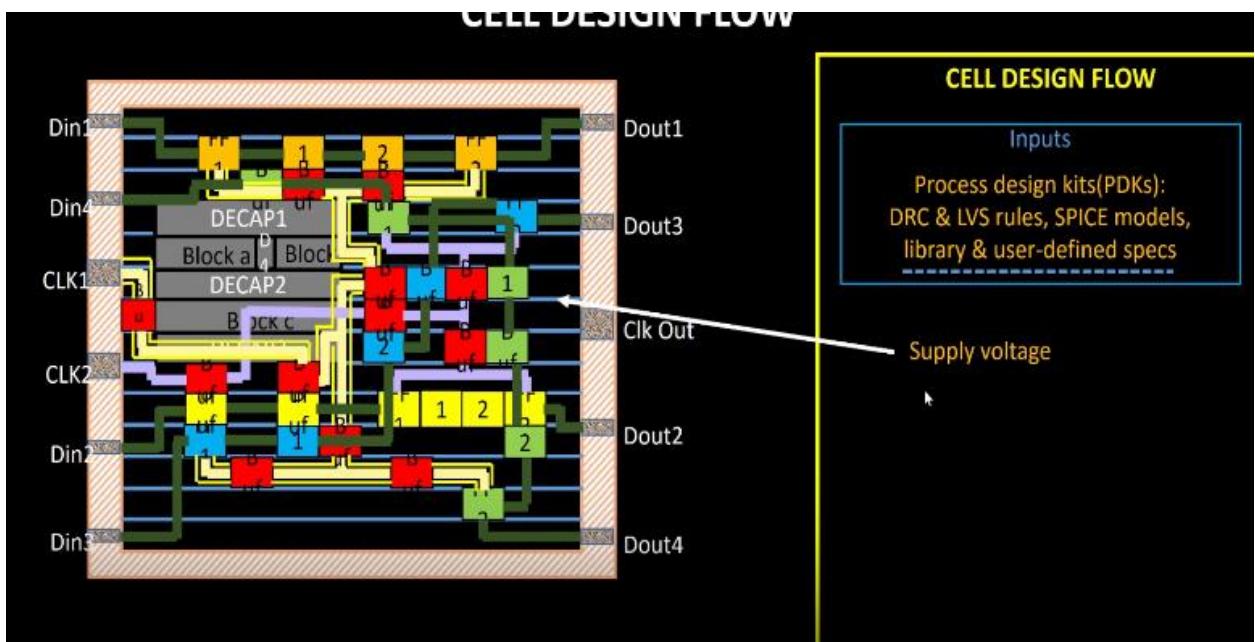


c) Library and user defined specs:

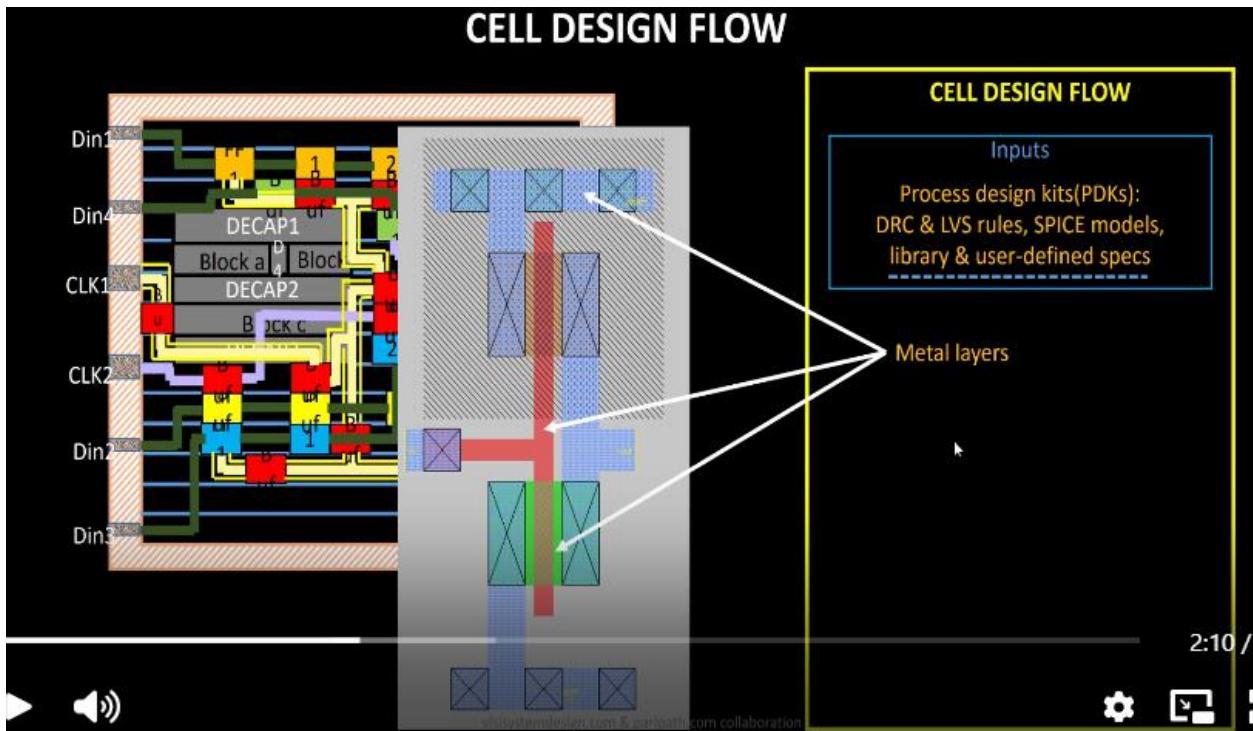
i) Cell height



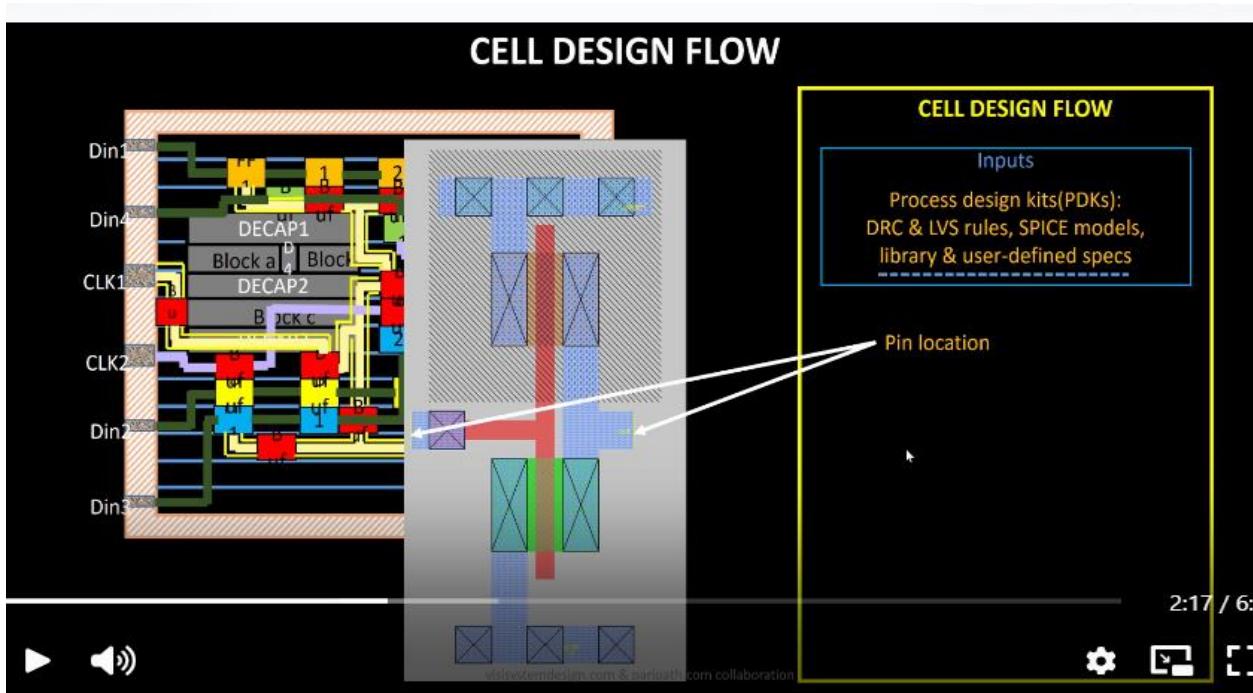
ii) Supply voltage



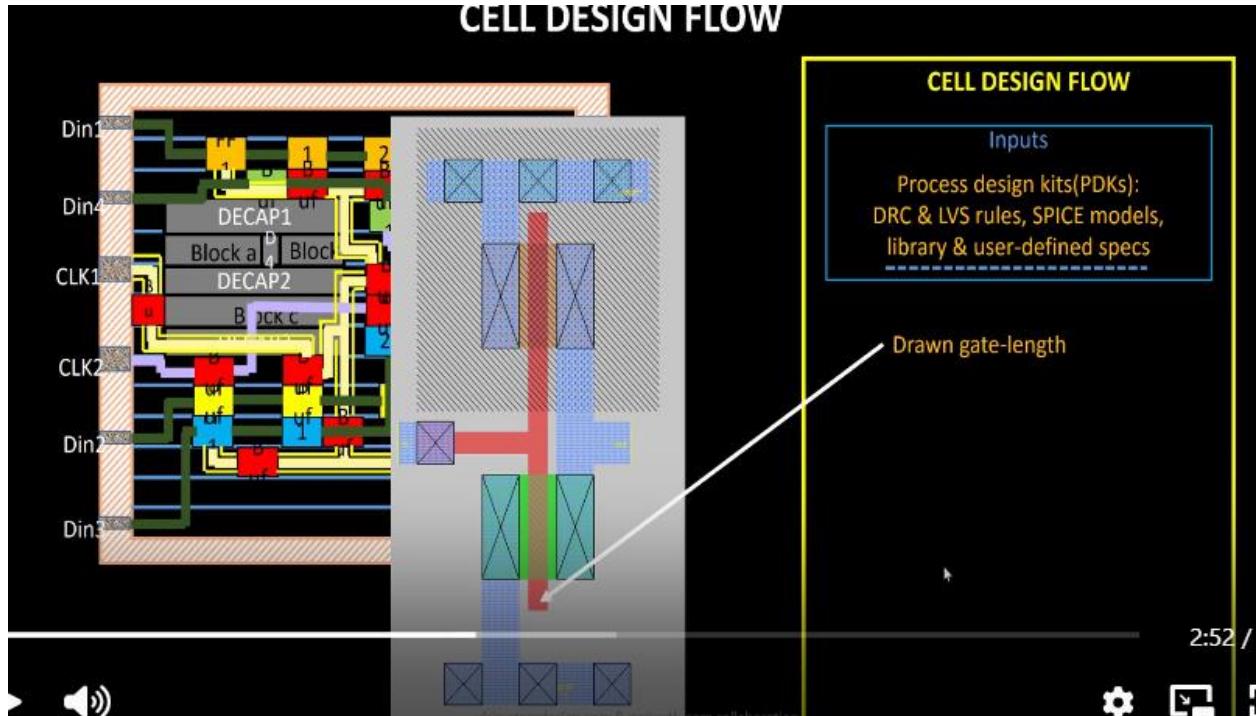
iii) Metal layers



iv) Pin Location



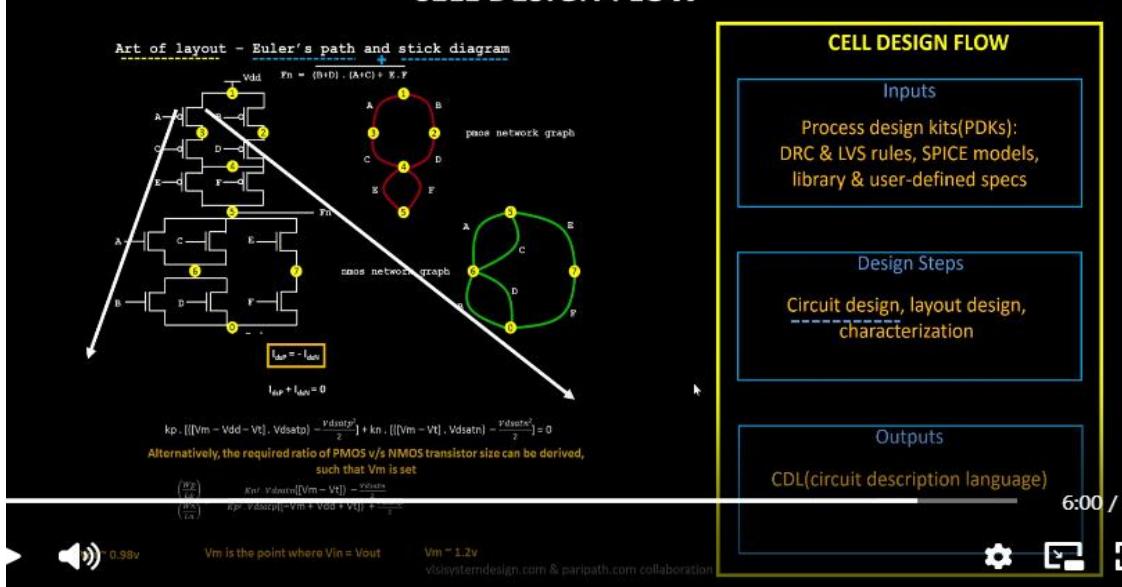
Drawn gate-length



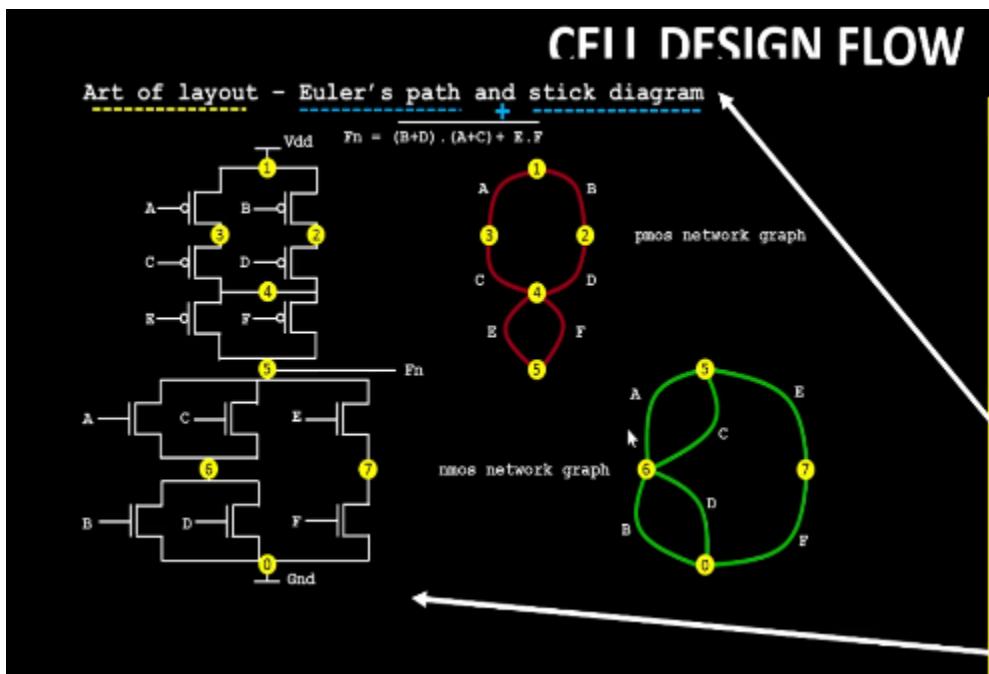
2) Design steps:

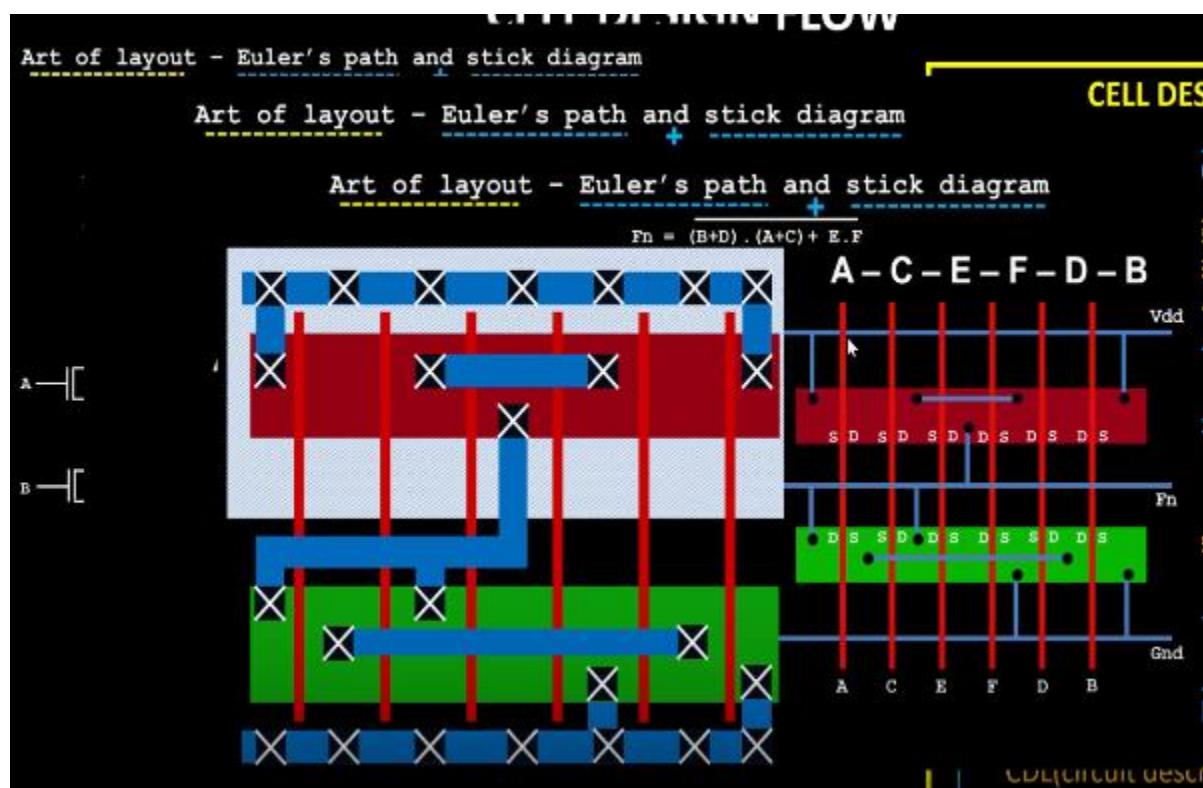
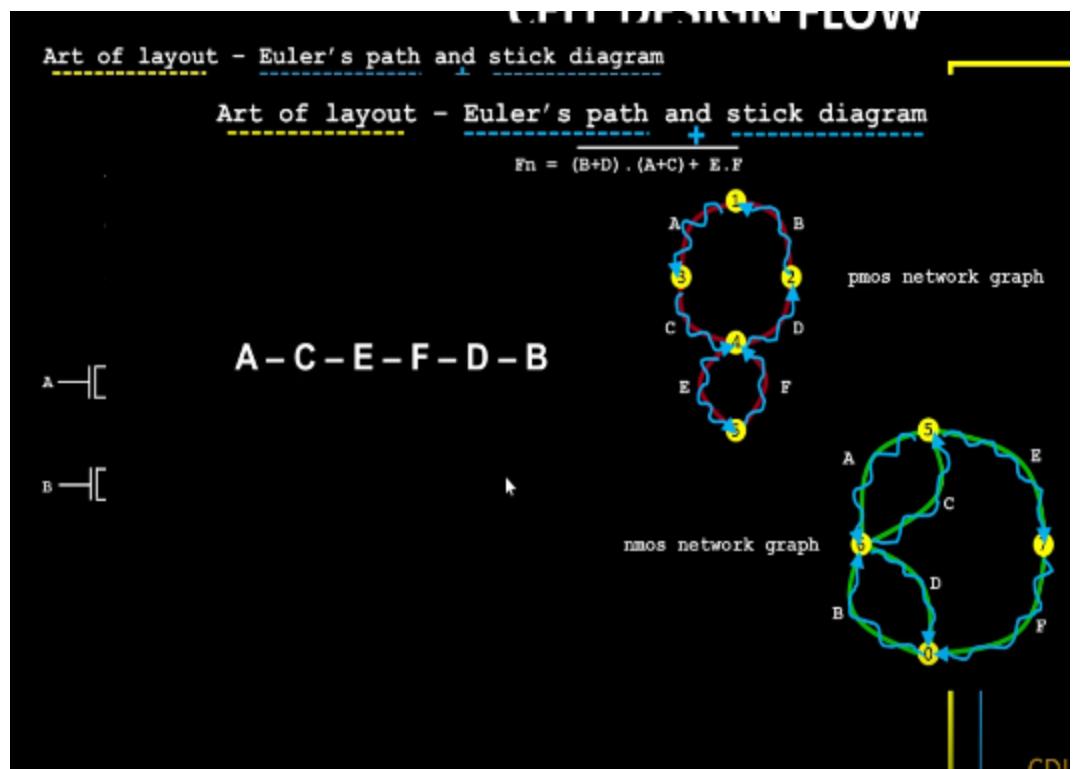
A) Circuit design:

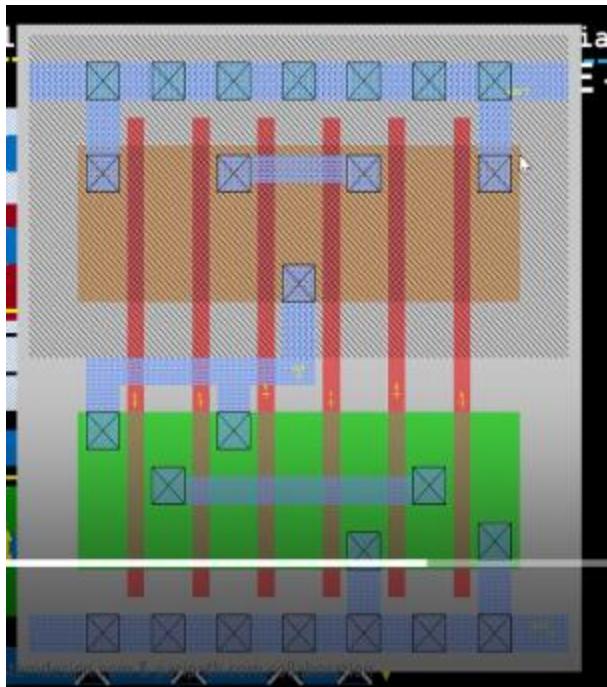
CELL DESIGN FLOW



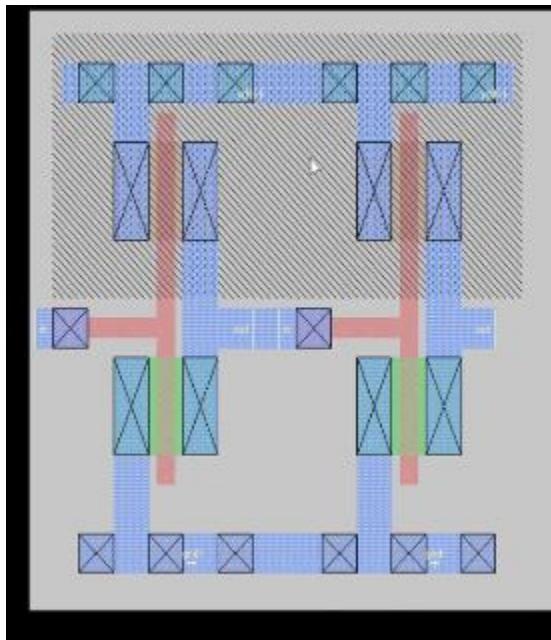
B) Layout design:

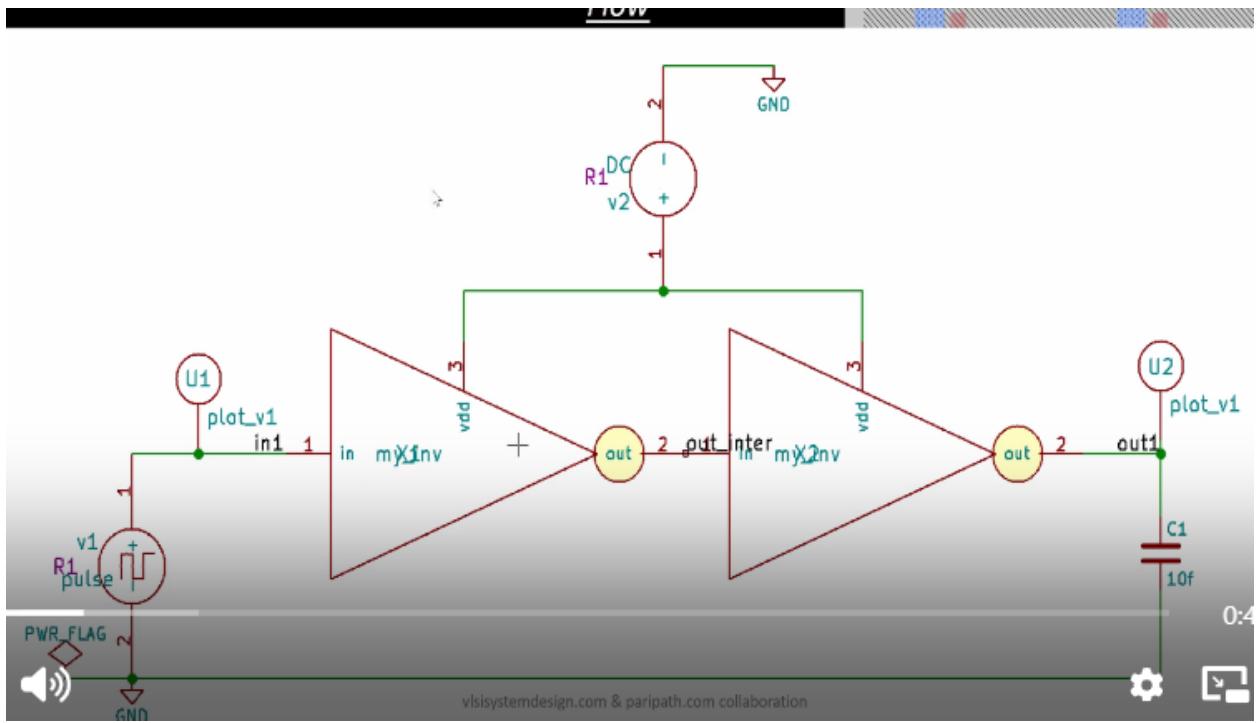




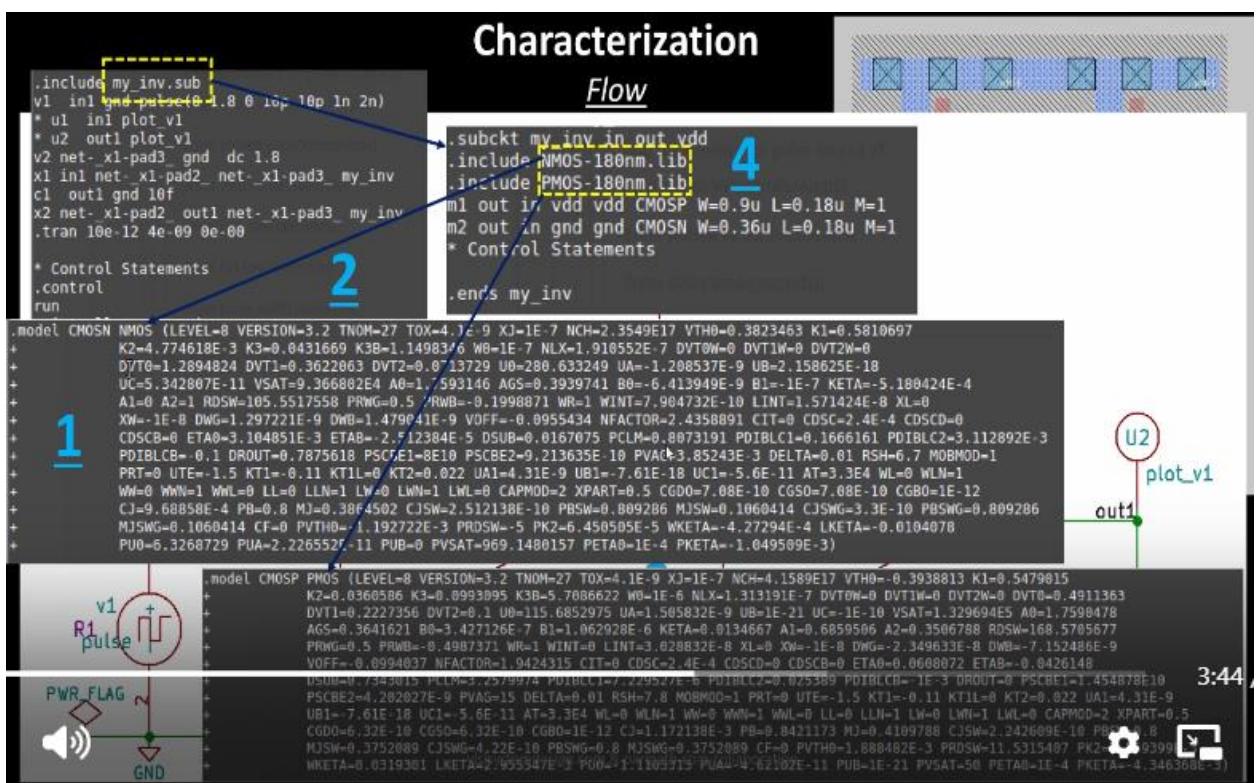


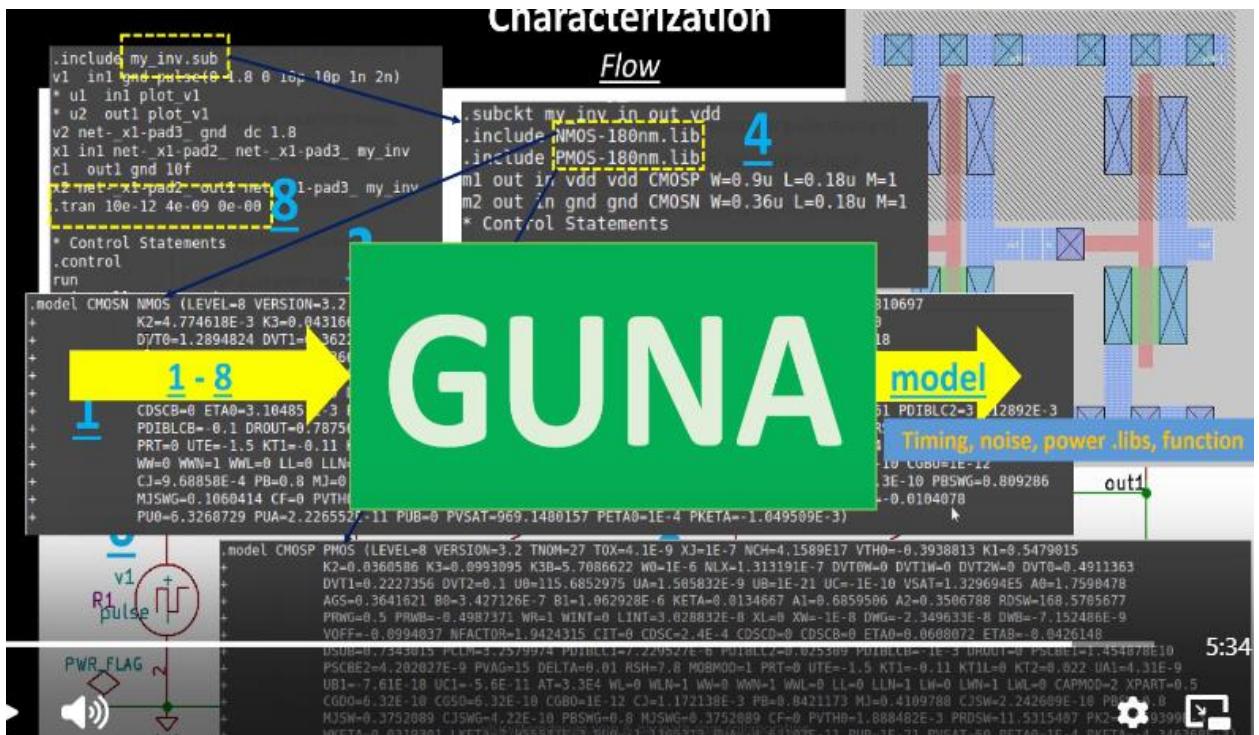
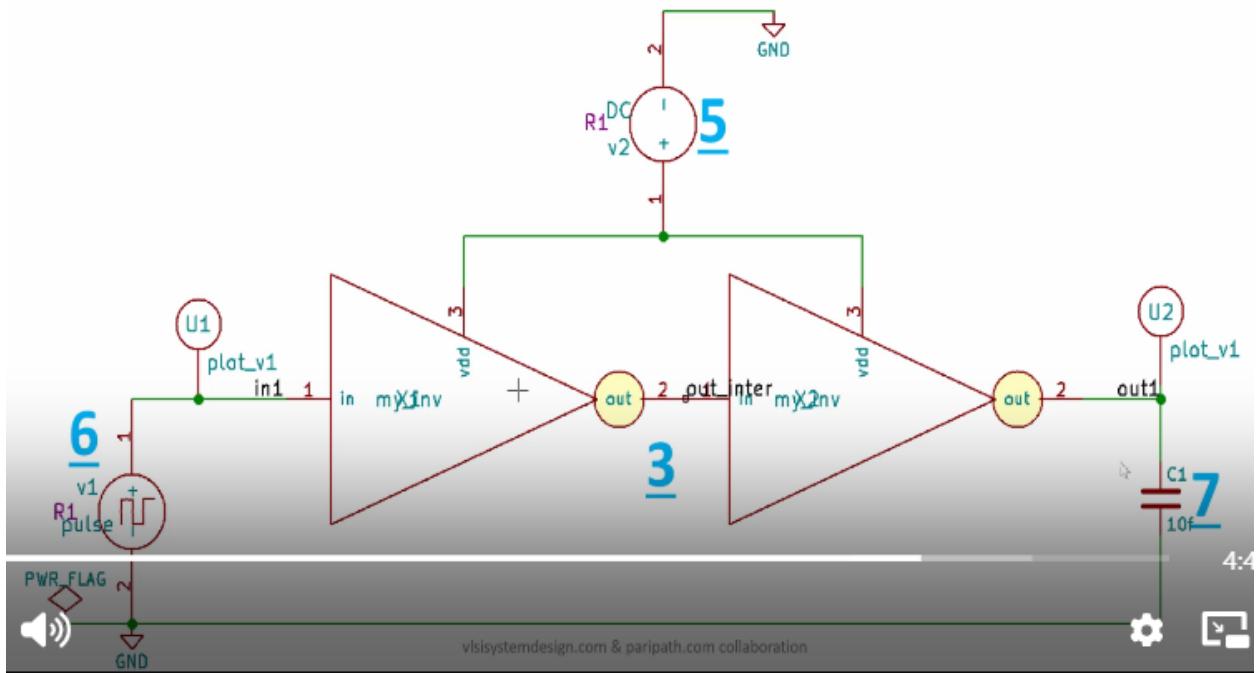
C) Characterization:





vislsystemdesign.com & paropath.com collaboration

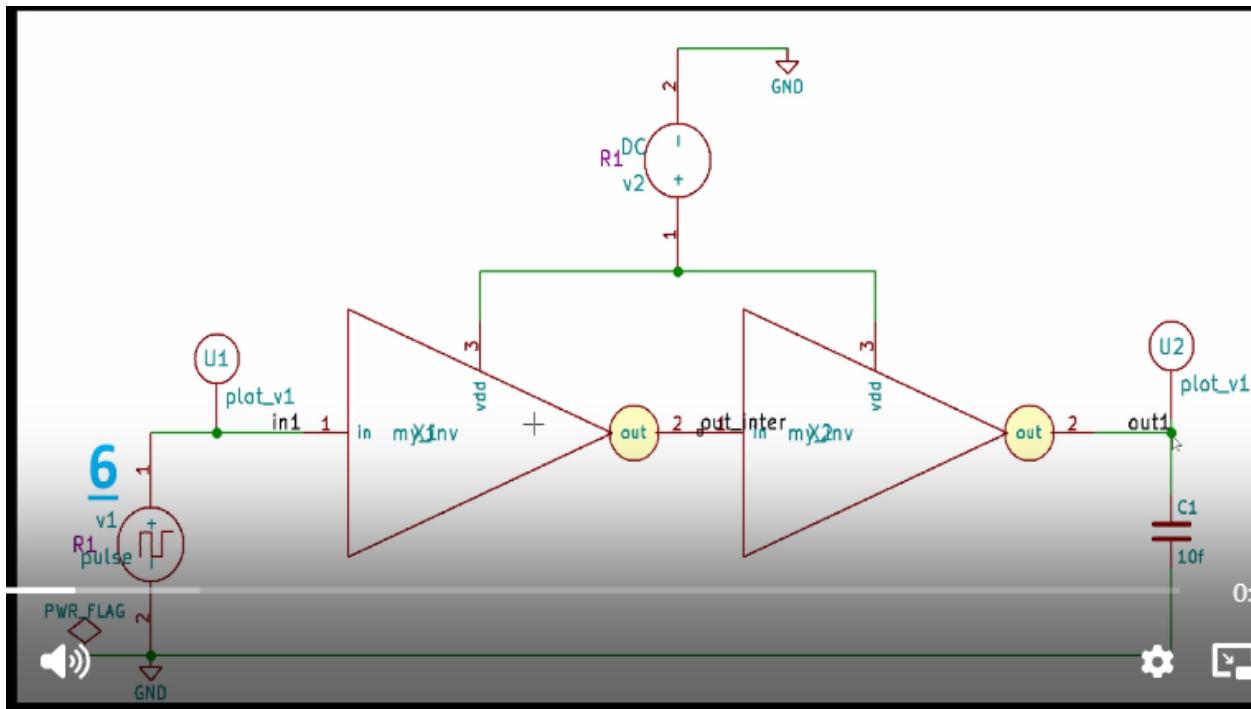


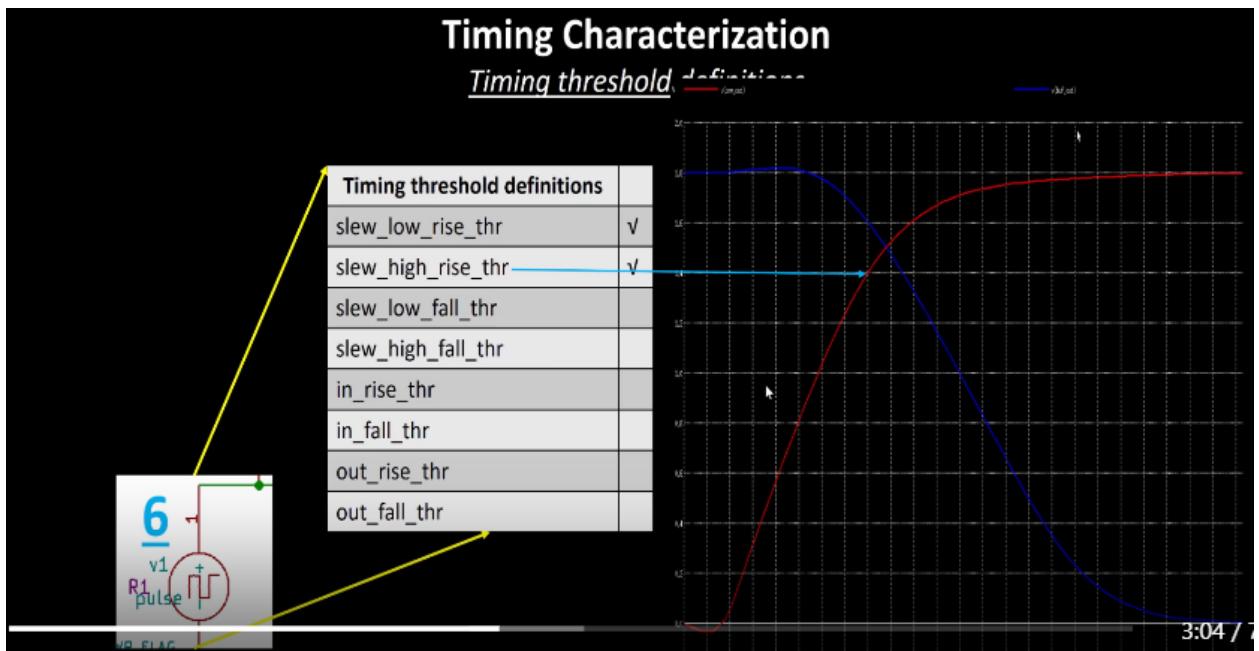
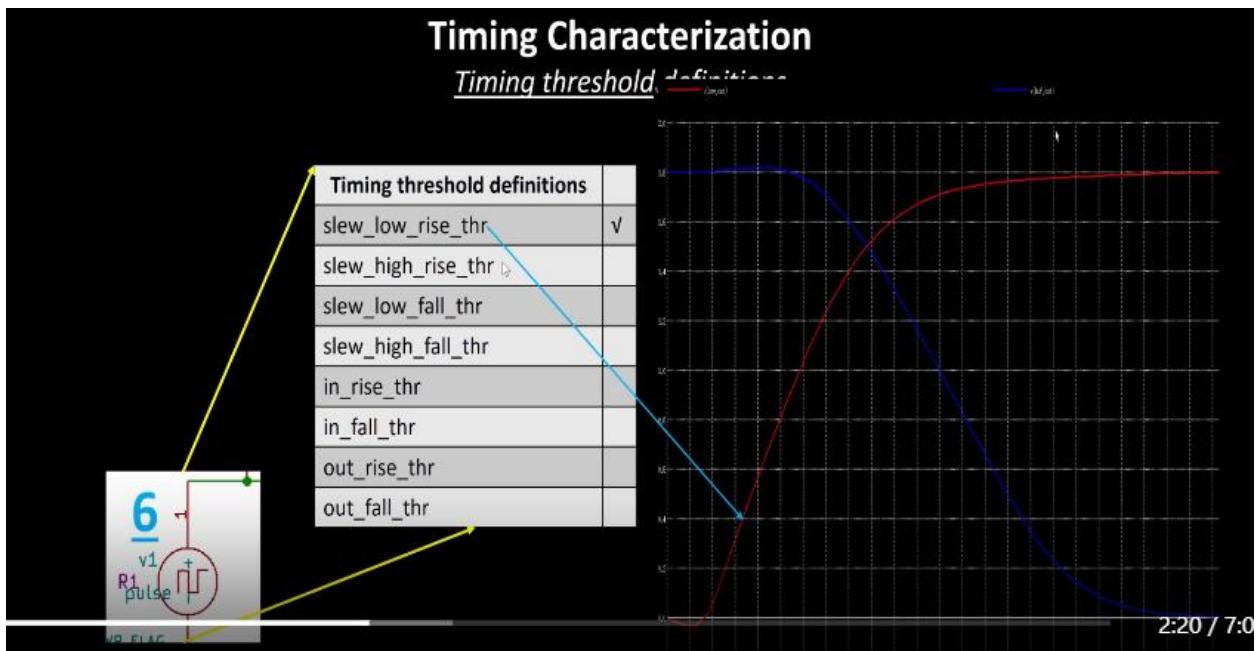




Timing Characterization

i) Timing characterization

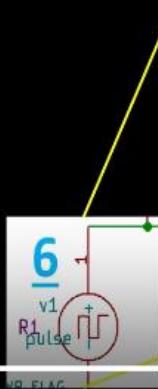




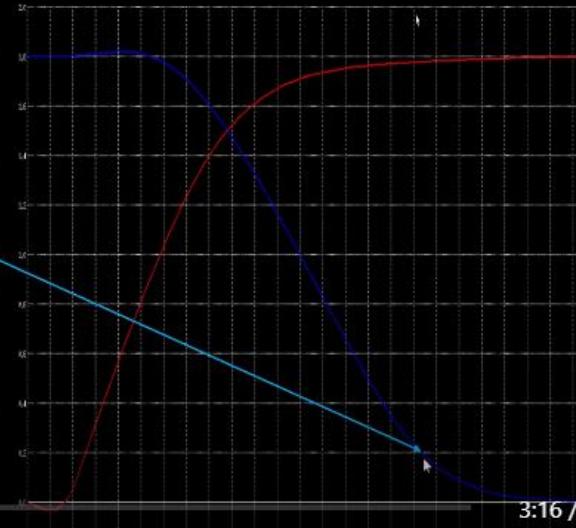
Timing Characterization

Timing threshold $\frac{dV}{dt} \text{ (mV/us)}$

$v_{dd,ext}$



Timing threshold definitions	
slew_low_rise_thr	✓
slew_high_rise_thr	✓
slew_low_fall_thr	✓
slew_high_fall_thr	
in_rise_thr	
in_fall_thr	
out_rise_thr	
out_fall_thr	

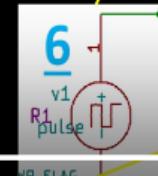


3:16 / 7:0

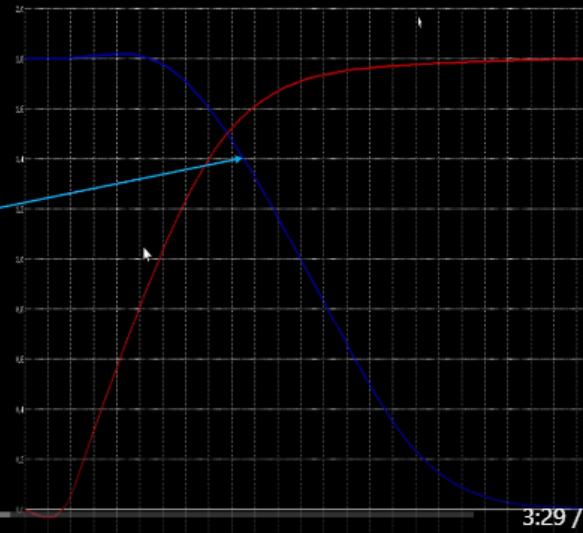
Timing Characterization

Timing threshold $\frac{dV}{dt} \text{ (mV/us)}$

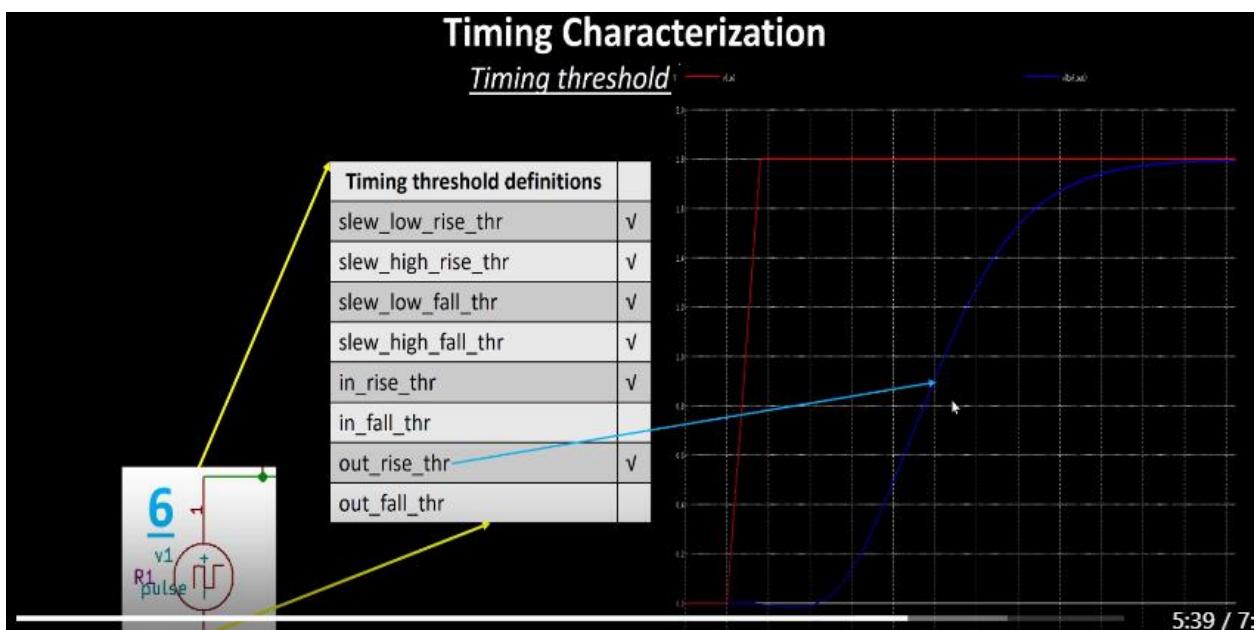
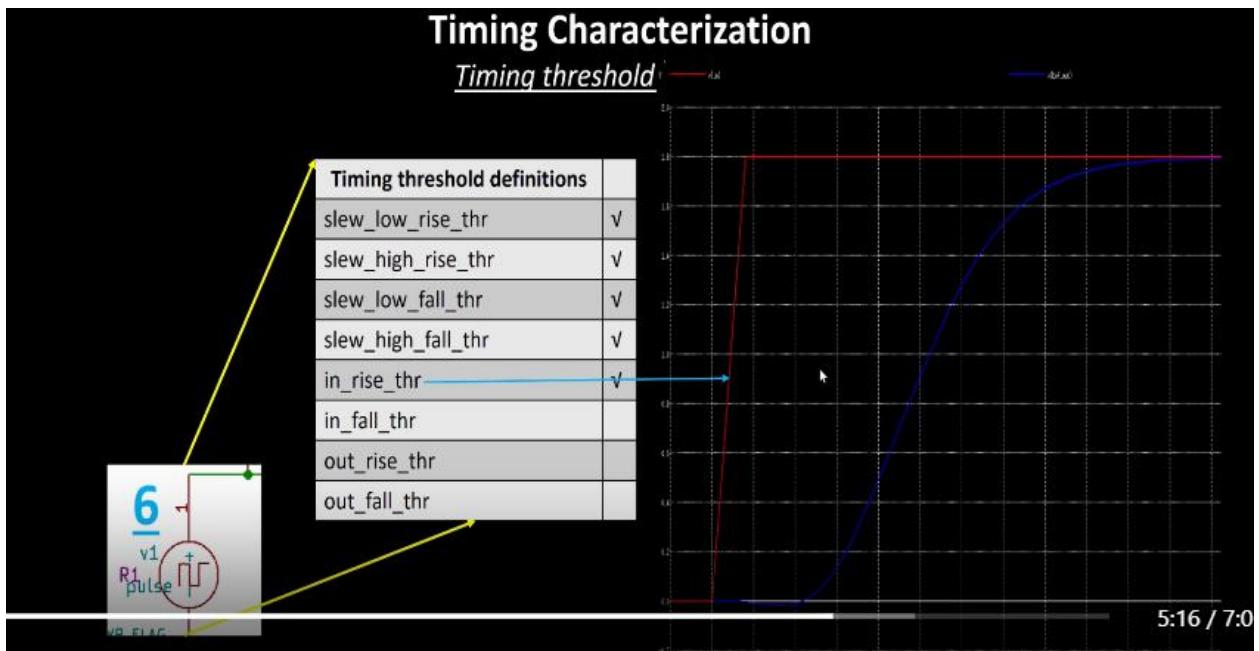
$v_{dd,ext}$

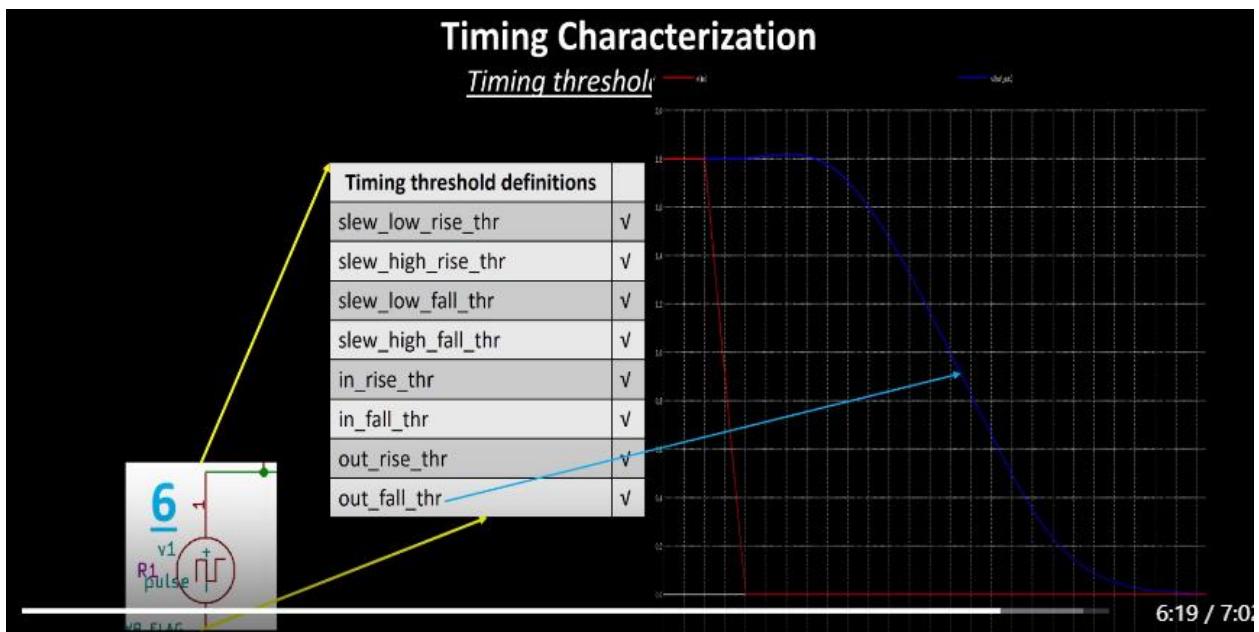
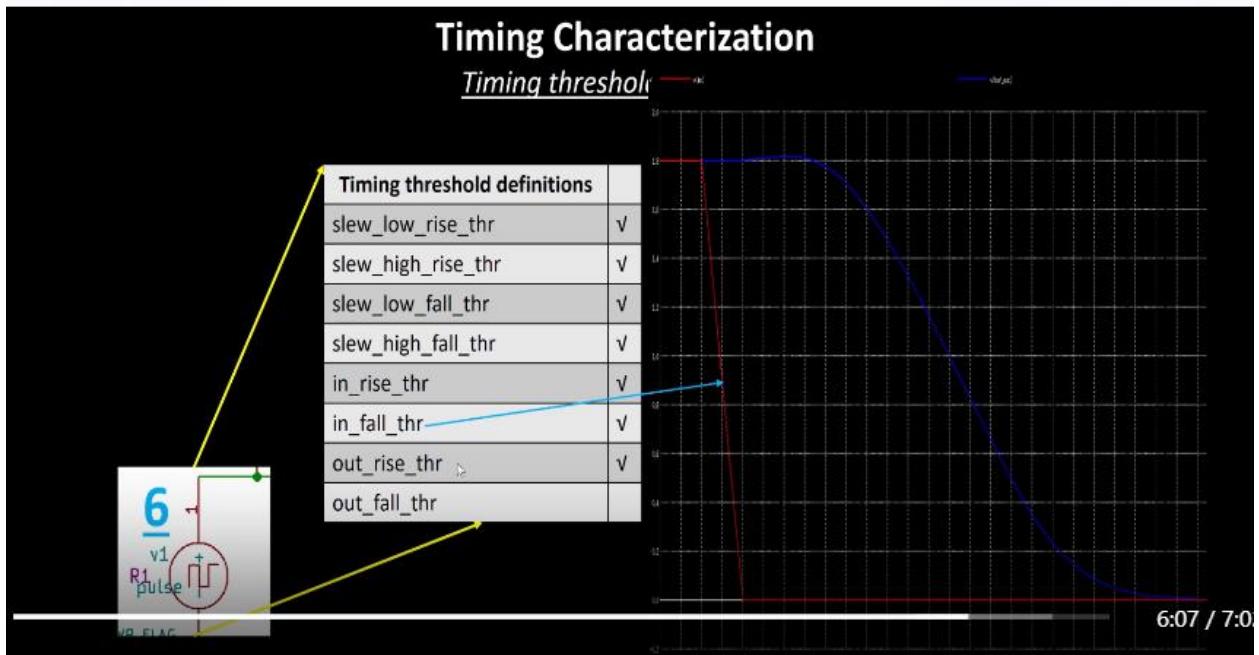


Timing threshold definitions	
slew_low_rise_thr	✓
slew_high_rise_thr	✓
slew_low_fall_thr	✓
slew_high_fall_thr	✓
in_rise_thr	
in_fall_thr	
out_rise_thr	
out_fall_thr	



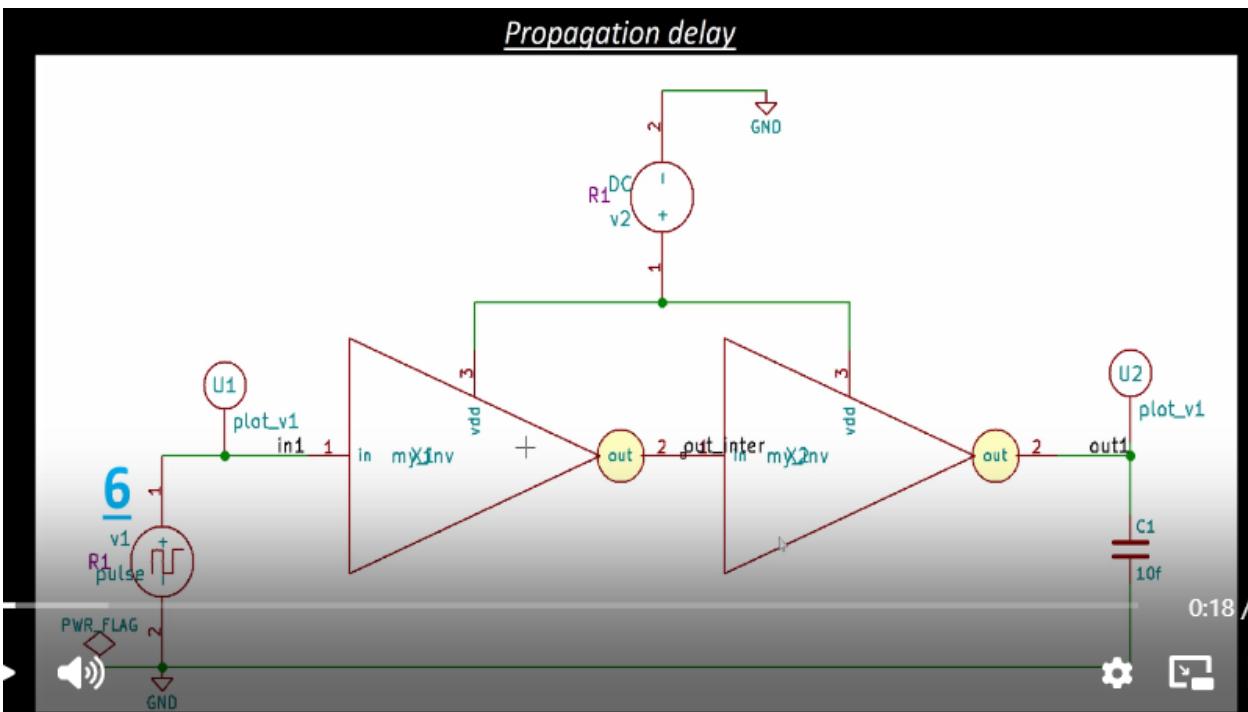
3:29 /





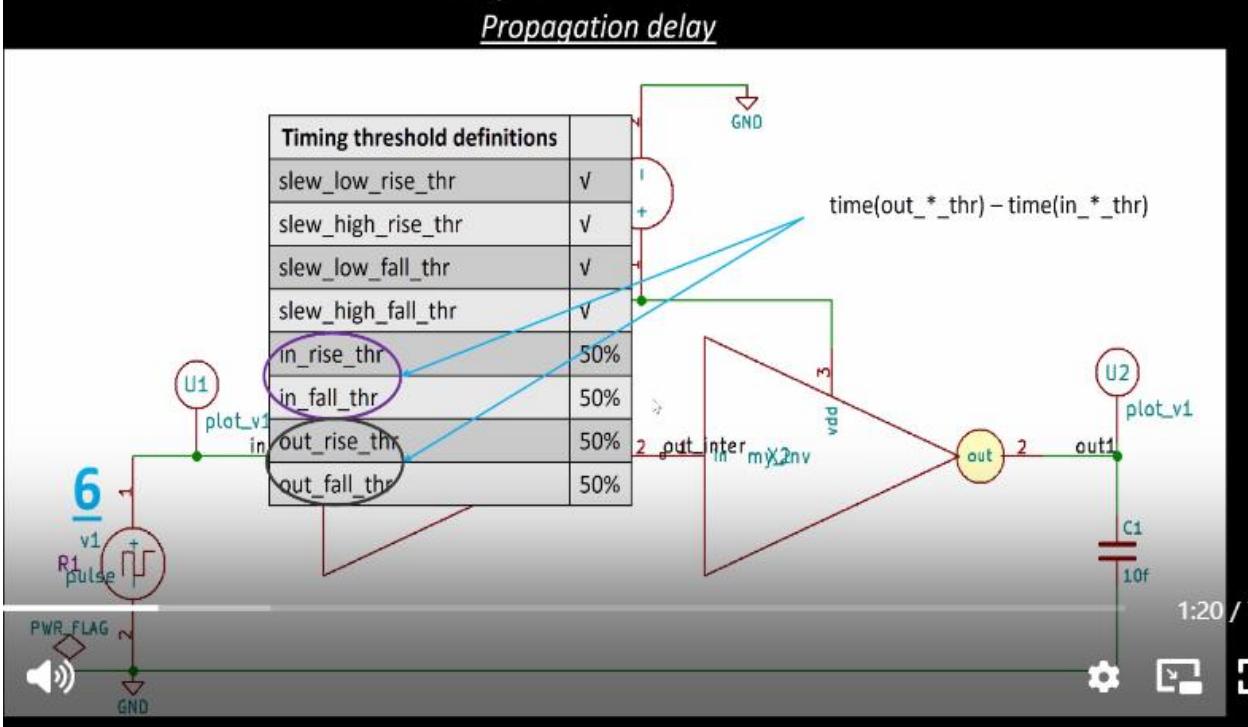
Propagation steps:

Propagation delay



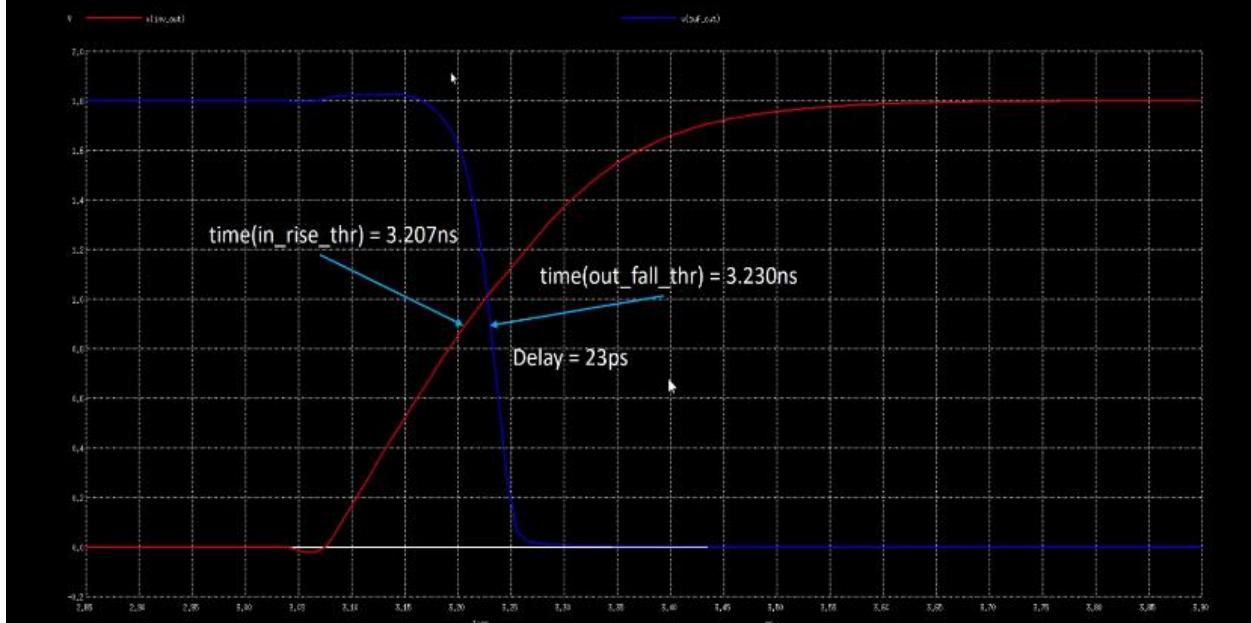
Timing characterization

Propagation delay



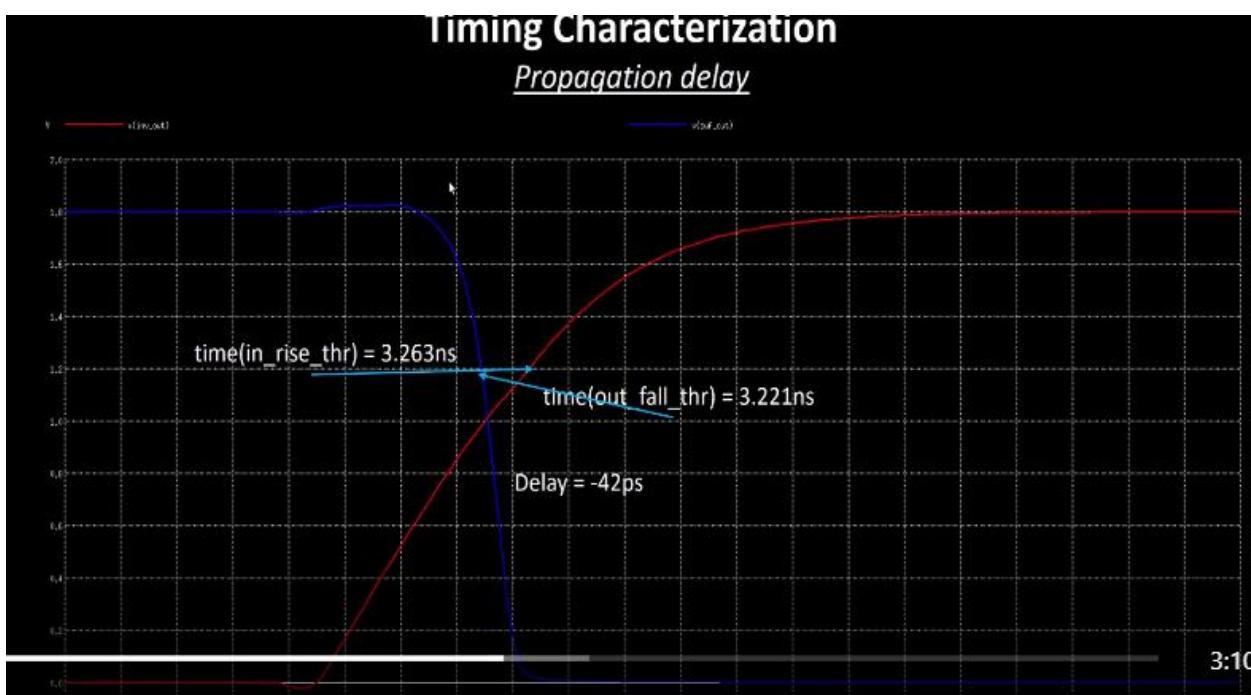
Timing Characterization

Propagation delay



Timing Characterization

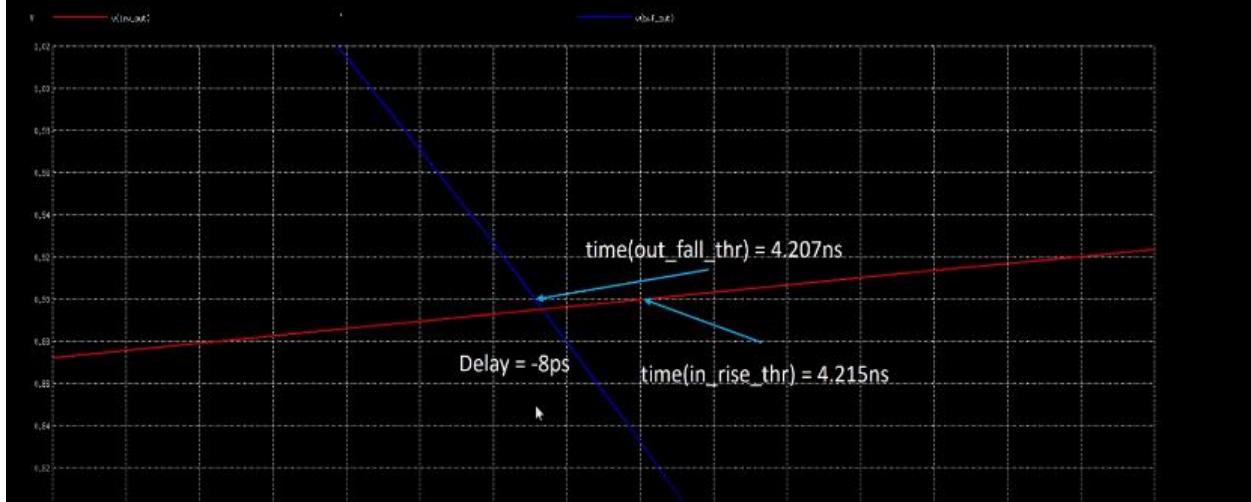
Propagation delay



3:10

Timing Characterization

Propagation delay



Transition time:

Timing threshold definitions	
slew_low_rise_thr	✓
slew_high_rise_thr	✓
slew_low_fall_thr	✓
slew_high_fall_thr	✓
in_rise_thr	50%
in_fall_thr	50%
out_rise_thr	50%
out_fall_thr	50%

Timing threshold definitions	
slew_low_rise_thr	v
slew_high_rise_thr	v
slew_low_fall_thr	v
slew_high_fall_thr	v
in_rise_thr	50%
in_fall_thr	50%
out_rise_thr	50%
out_fall_thr	50%

time(slew_high_fall_thr) – time(slew_low_fall_thr)

