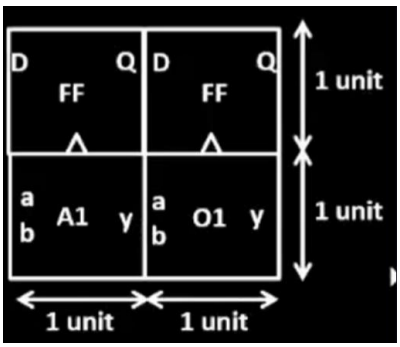
DAY-2 GOOD FLOORPLAN VS BAD FLOORPLAN AND INTRODUCTION TO LIBRARY CELLS.

1. CHIP FLOOR PLANNING CONSIDERATIONS

1: Utilization factor and aspect ratio

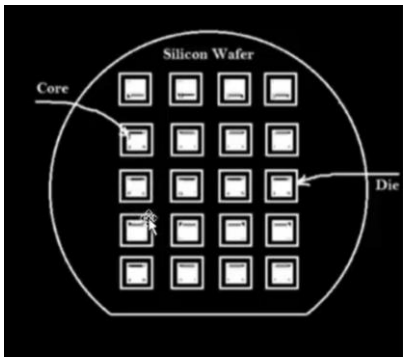
- **Determining the core and Die:** The width and height of the Core and Die, which is a crucial first step in the physical design flow. To begin, consider a simple netlist composed of two flip-flops connected by basic combinational logic. A netlist essentially defines how different electronic components are interconnected.

  The std cell dimension 1x1 unit



  Height=Width = 1+1=2units

- **Block diagram of core and a die**



- **Utilization factor:** Area occupied by netlist / Total area of the core
  - Say the given area occupied is 4units and area of core is 2 x 2
  - We have utilization factor=4x1/2x2=1

- If utilization factor is 1 its completely occupied there is no extra spacing for components

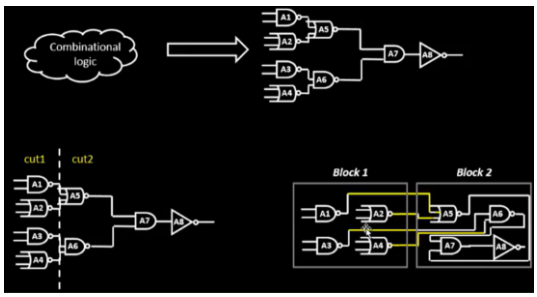
- **Aspect Ratio:** height/width = 2/2 = 1 unit

  - If the aspect ratio is 1 the chip is square shaped anything other than one is rectangle shaped

2: Concept of pre-placed cells

- **Breaking Down a Large Combinational Logic**

  Imagine we have a large combinational circuit performing a specific function, composed of N logic gates. Instead of tackling the entire circuit as one block, we divide it into smaller sections. The entire logic is split into two parts, and each part is grouped into its own separate block. These two blocks will then be designed and implemented independently.



- **Black Boxing the Blocks**

  Once we've defined the blocks, we extend their input and output pins to make connections easier. At this point, we black box each block — meaning we hide the internal details and treat them as standalone units. From the perspective of the top-level netlist, the internal logic of these blocks becomes invisible — only the interface (inputs and outputs) is visible. These blocks are now treated as independent IPs (Intellectual Property blocks) or modules.

- **Benefits of Modular Design with Ips**

  The biggest advantage of this approach is reusability. Once an IP block is designed and verified, it can be reused multiple times in different parts of the chip or even in other projects.
  Other examples of such reusable IPs include:
  - Memory blocks
  - Clock-gating cells

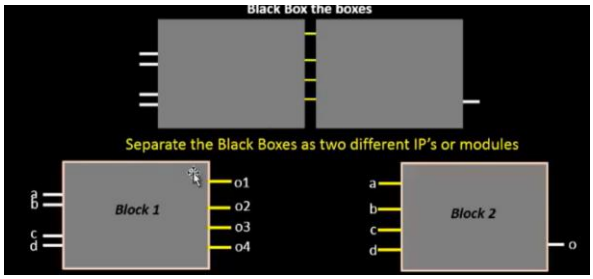
- Comparators
- Multiplexers (MUX)

All these components are part of the top-level netlist. They perform specific functions based on input signals and return outputs, but their internal implementation is done only once.

- **Floor planning and Pre-Placed Cells**

  The process of arranging these IPs on a chip is known as floor planning. During this stage, designers choose specific locations on the chip for each IP based on performance and connectivity needs.

  These IPs have predefined positions set by the designer — that's why they're known as pre-placed cells. These cells are strategically positioned before automated placement and routing begins. As a result, automated tools are restricted from modifying their location during the later stages of design.

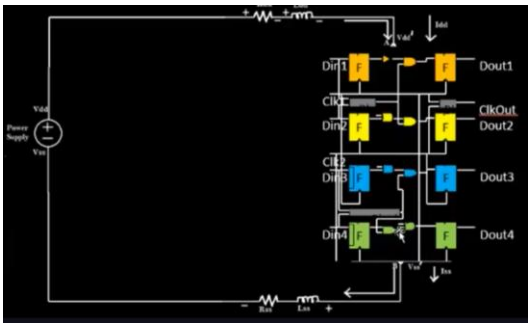


3: Decoupling capacitors

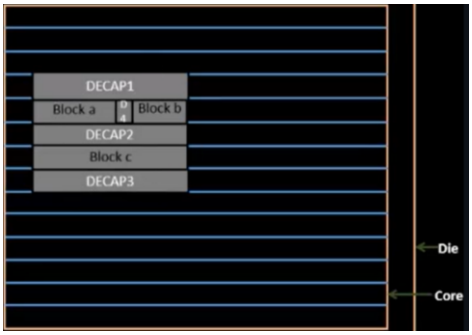
- **Surrounding Pre-Placed Cells with Decoupling Capacitors**

  - Let's take a moment to consider a section of a circuit that forms part of a previously discussed logic block. Suppose a gate — say, an AND gate — transitions its output from logic level 0 to 1, or from 1 to 0. During such transitions, the gate draws a switching current, which is influenced by the small intrinsic capacitance present at its output.
  - To properly represent a logic '1', this tiny capacitance needs to be fully charged. Similarly, to represent a logic '0', it must be completely discharged.
  - Now, if we assume this capacitance to be negligible (zero) for simplicity, the demand for switching current still remains valid. During a switching event, the circuit experiences a brief but intense current surge, often referred to as peak current.
  - In this scenario, certain electrical elements like Rdd (resistive component), Ldd (inductive component), and Lss (source-side inductance) come into play — all of which have defined and constant values.

- When this peak switching current flows through the circuit, the resistance and inductance (Rdd and Ldd) cause a voltage drop. As a result, the voltage at Node A is slightly reduced and registers as Vdd' instead of the expected Vdd.
- To address this voltage drop and stabilize the power delivery, decoupling capacitors are strategically placed around pre-placed cells. These capacitors act as local energy reservoirs, providing the necessary charge during switching events and helping maintain the voltage level close to ideal — effectively minimizing the impact of transient fluctuations.

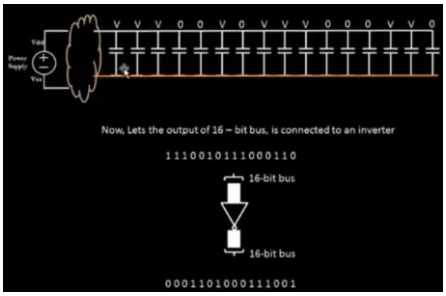


- After the placement of decap around preplace cells it looks something like the below



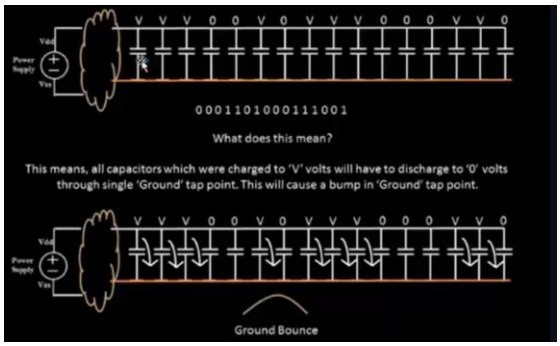
4: Power planning

- **Assuming a 16 bit bus:**
  - There is only one power supply here which could be a problem for capactiors for charging and precharging

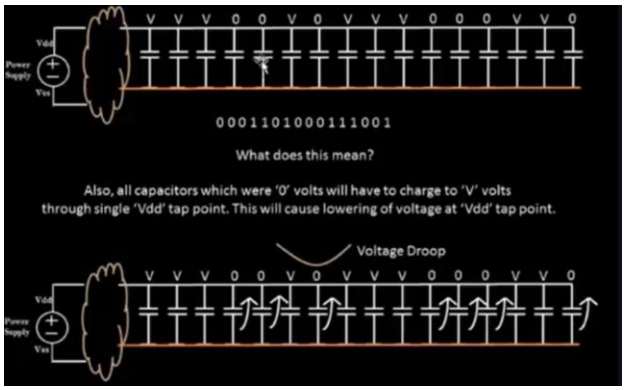


- **Ground bounce:** Say the inverted value which makes all 1's to 0's and vice versa at the stage all the 1's to 0's gets discharged at the same time

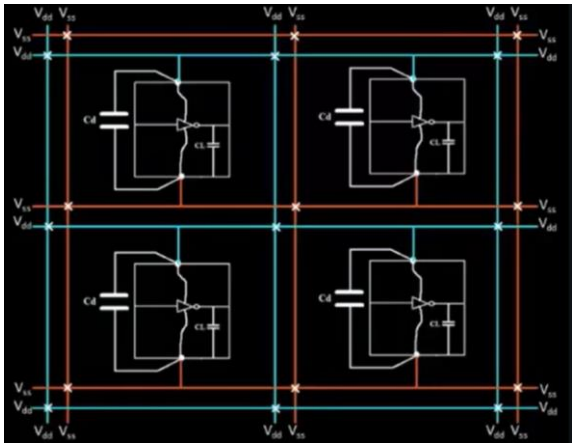
causing ground bounce ie., it will go slightly above the noice margin or will fall in between the range which is uncertain.



- **Voltage drops:** Say the inverted value which makes all 1's to 0's and vice versa at the stage all the 0's to 1's gets charged at the same time causing Voltage drop ie., it will go slightly below the noice margin or will fall in between the range which is uncertain.



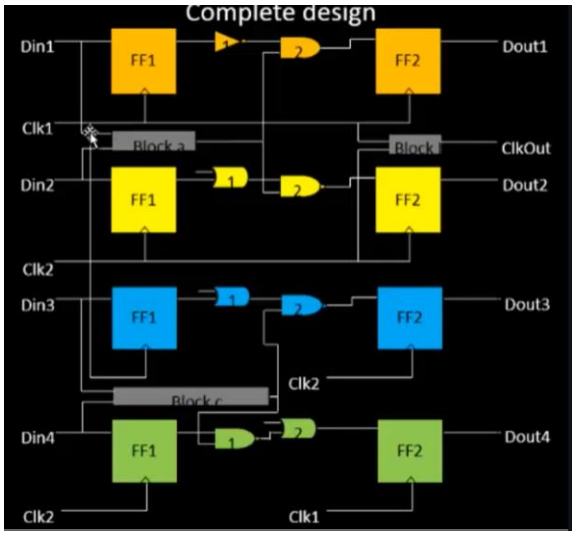
- To ensure this doesn't happen we make sure the voltage and ground line placed properly such that every capacitor can access it nearest source which could avoid voltage drop or ground bounce



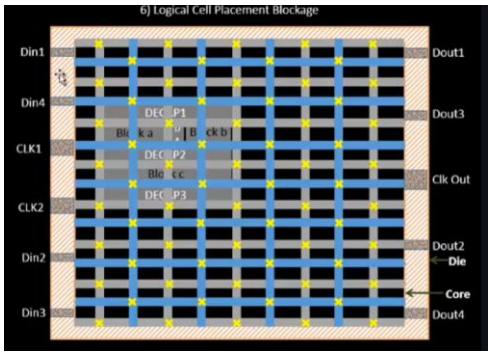
5: Pin placement and logical cell placement blockage

- **Design and pin placement:**
  - The pins are placed logically according to our design logically

- The design



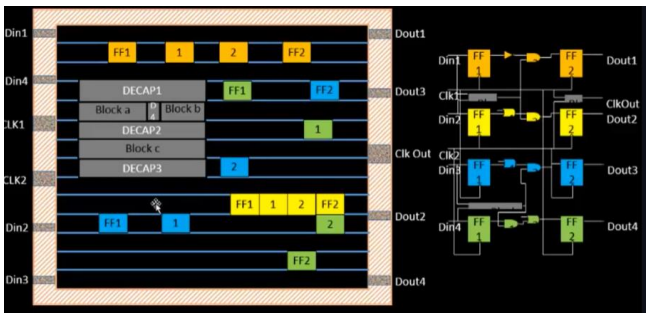
- Pin placement with pre placed cells , decap cells, power planning making it ready for flooplanning



2. LIBRARY BINDING AND PLACEMENT

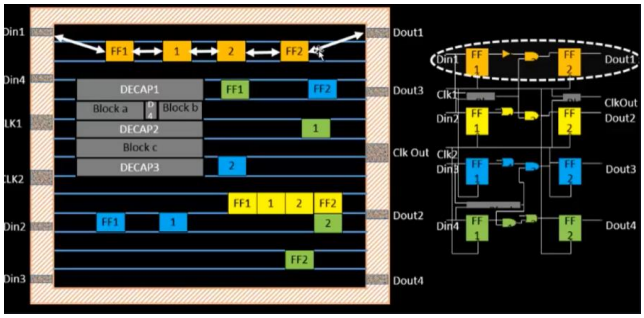
1: Netlist binding and initial place design

- **According to the design place the cells at each step near to the pins, atleast if not very near make sure the components are placed in a way such that a buffer can connect them**



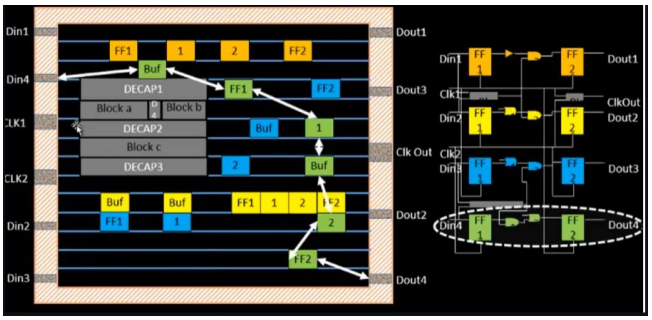
2: Optimize placement using estimated wire-length and capacitance

- Connect the logic as in check if they can be connected by making sure the wire length.
- We should not have more wire length because they could cause an uncertainty in the output.
- Hence, we maintain the min and max wire length during routing stage



3: Final placement optimization

- Place buffers where ever the distance between the two components to be connected is more such that it would connect maintain the logic.



4: Need for libraries and characterization

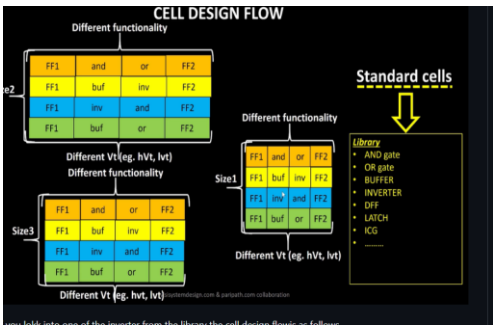
- Every **Integrated Circuit (IC) design flow** follows a structured sequence of essential steps to transform a functional description into a working chip. The first of these steps is **Logic Synthesis**.
- Let's assume we start with a design described in **RTL (Register Transfer Level)** — a high-level representation of the circuit's functionality. **Logic synthesis** is the process of translating this RTL code into a valid hardware implementation. In simpler terms, it converts the functional description into a **network of logic gates** that performs the same operation.
- Once logic synthesis is complete, the next stage is **Floorplanning**. In this step, we import the synthesized gate-level netlist and begin defining the physical layout — specifically, the dimensions of the **Core** and **Die**. This step provides a rough blueprint for how the chip will be organized spatially.

- Following floorplanning, we move to the **Placement** phase. Here, each logic cell is positioned on the chip in a way that improves initial **timing and performance**. Good placement helps optimize signal paths and prepare for later stages.
- After placement comes **Clock Tree Synthesis (CTS)**. This step ensures that the **clock signal** reaches all parts of the chip uniformly. The objective is to maintain **equal rise and fall times** and deliver the clock signal to every sequential element (like flip-flops) with minimal skew.
- Next is the **Routing** phase. During routing, we establish the actual physical connections (wires) between the cells. This step follows a defined flow that often depends on the **characteristics of the flip-flops** used in the design.
- Next is the **Routing** phase. During routing, we establish the actual physical connections (wires) between the cells. This step follows a defined flow that often depends on the **characteristics of the flip-flops** used in the design.
- Throughout all of these steps — from synthesis to STA — the one common element is the use of **logic gates or standard cells**. These components are the building blocks of every stage in the digital design process.

3. CELL DESIGN AND CHARACTERIZATION FLOWS

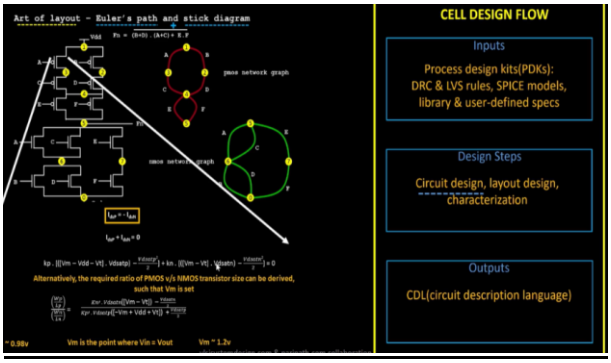
1: Inputs for cell design flow

- The cell design is categorised into 3 parts inputs, design steps and outputs



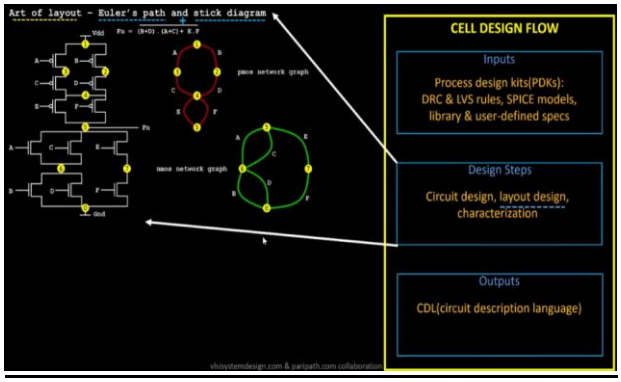
2: Cricuit design step

- Inputs, design steps and outputs are all connected are major steps for a cell design flow

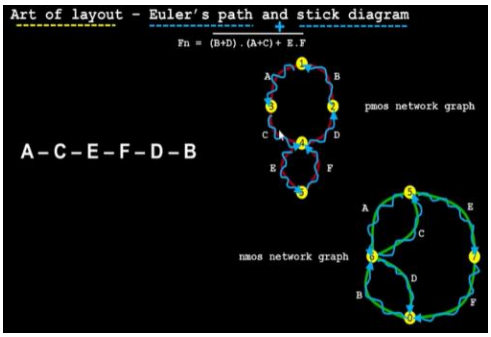


3: Layout design step

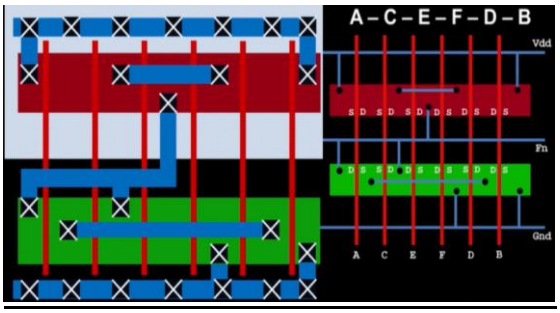
- To obtain the function implemented through the MOS transistor through a set of PMOS and NMOS transistor and the second step is to get the PMOS network graph and the NMOS network graph out of the design that has been implemented.



- Using Eulers path determine the path

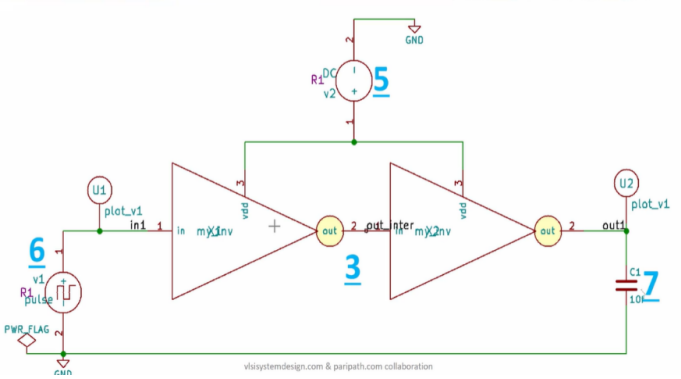


- Get the stick diagram



4: Typical characterization flow

- Load the Model: The first step is to import the required device model into the simulation environment.
- Read the Extracted SPICE Netlist: Next, we load the SPICE netlist that describes the circuit's extracted layout and connections.
- Define Buffer Behaviour: In this step, we define or identify the functional behaviour of the buffer used in the circuit.
- Include Inverter Subcircuits: Now, we read in the definitions for the inverter subcircuits that will be part of the simulation.
- Connect Power Supplies: After loading the circuit elements, we attach the appropriate power supplies (e.g., VDD, GND) required for the components to operate.
- Apply Input Stimulus: Once the circuit is powered, we apply the desired input signals or stimulus to test its behaviour.
- Set Output Load Capacitance: To simulate realistic conditions, we define the necessary output capacitance at the circuit's output node(s).
- Run the Simulation: Finally, we issue the appropriate simulation command — for example, use the .tran command for transient simulation, or the .dc command for DC analysis.



- All the inputs from 1-8 feed it to GUNA model which is used for Timing, Power and Noise characterization.

4. GENERAL TIMING CHARACTERIZATION PARAMETERS

1: Timing threshold definitions

- These eight parameters determine characterization flow. Determined in lab steps