Basic Physical Design interview questions

1. What are the input files required for PnR?

- Gate level Netlist (.v)
- Library
  - Std cell timing, logical & power information (.lib) or (.db)
  - Std cell Physical information (.lef)
  - Std cell Technology information (.tlef)
  - Synopsys Design Constraints (.sdc)
- FP DEF and scan DEF
- Power intent
  - Unified Power Formate (UPF)
  - Common Power Formate (CPF)
- RC co-efficient files (capTable / Table look up(TLU))

2. How many types of LEF files are there? State the differences between the two. Which LEF do we read into the PnR tool first?

There are 2 type of Library Exchange Format(LEF) in PD
1. Technology LEF
2. The Physical Std. Cell LEF

1. Technology LEF:

Technology LEF part contains the information regarding all the metal interconnects, via information and related design rules.

2. The Physical Std. Cell LEF:

The Cell LEF file provides geometric and physical information about the standard cells used in the IC design.

Cell Name, Shape, Size, Orientation & Class
Port/Pin Name, Direction, locations, layer information.

In PnR tool we read technology LEF first.

3. Name Physical Cells that are used in PnR and explain why we need them.

What is Physical Cell?

These cells are not present in the design netlist. These cells don't have any logic pins and use only to meet some DRC rules and for design protection and they don't have any signal connectivity, only Vdd and Vss. they do not appear on timing paths reports. They are typically invented for finishing the chip.

Tap cell – Well tap cells or Tap cells are used to prevent the latch-up issue in the CMOS design. Well tap cells connect the nwell to VDD and p-substrate to VSS in order to prevent the latch-up issue.

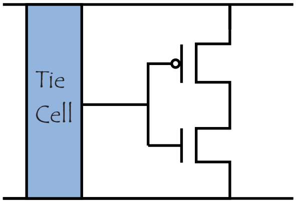
Well tap cells have no logical functions, it has only two connections.

- nwell to VDD
- p-substrate to VSS

TAP CELL will reduce substrate & well resistance and that will help to avoid LATCHUP.

Tie Cell – Tie cell is a std cell. It designed to provide the high or low signal in the input gate terminal.

In design there should not be any floating input pin. they must connect with VDD or VSS. If a std cell input pin connect direct to the VDD or VSS it might damage the cell, so to prevent this issue we use tie cell.



Endcap Cell (Boundary Cell) – There are high chances to get damaged the gate of standard cells placed at the boundary during the manufacturing of chip.

https://www.linkedin.com/in/ajmain-ar-rafi/

To prevent such damages at the boundary we have a special kind of cell in the standard cell library is called end cap cell or boundary cell. This cell not only protects the gate damage at the boundary, but it also used to meet DRC rules related to well & implant layer case and alignment of different voltage domains.

Where to Place ENDCAP Cell :

- We place these cell before placement so it is called preplaced cell .
- Place these cell after site row creation and macro placement
- Place these cell at the end of row ..

Decap Cell – Decap cells are basically a charge storing device made of the capacitors and used to support the current requirement in the power supply network.

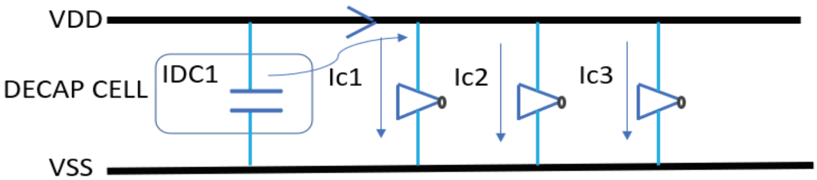
There are various reasons for the large current requirement in the circuit and if there are no extra source to handle this requirement, power droop or ground bounce may occur.

These power droop or ground bounce will affect the constant power supply and it lead to the delay of standard cells.

Why we use DECAP CELL :

As we know transition delay is inversely proposal to voltage, so decreasing in voltage will increase delay of circuit, if drop or bounce is significant then that lead to malfunction also in design.

To overcome this issue we need to do IR analysis and identify location where we need support of decap cell and then we add DECAP on those area near the std cell that provide current to the circuit when it is required, as shown below:



Where to place DECAP CELL :

Decap cells are placed generally after the power planning and before the standard cell placement

Disadvantage of DECAP CELL:

- It adds leakage current .
- It increase area

Filler Cell – Filler cells primarily are non-functional cells used to continue the VDD and VSS rails.

Filler cells are used to establish the continuity of the N- well and the implant layers on the standard cell rows. The use of Filler Cells is they reduce the DRC Violations created by the base (N-Well, Pim & Nim) layers. and also help maintain the Power Rail, VDD/VSS connection continuity.

4. What are the different types of timing paths that are analyzed during timing analysis? Which timing path is the most critical path and needs to be resolved before any other timing paths?

There are 4 types of timing paths:

- Data Path.
- Clock Path.
- Clock Gating Path
- Asynchronous Path

Critical Path:

In short, I can say that the path which creates Longest delay is the critical path.

Critical paths are timing-sensitive functional paths. because of the timing of these paths is critical, no additional gates are allowed to be added to the path, to prevent increasing the delay of the critical path.

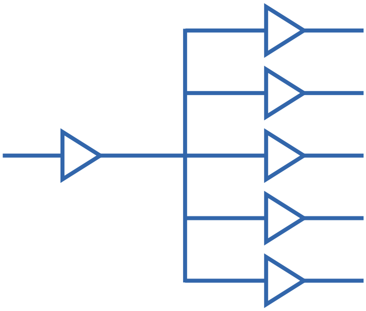
Timing critical path are those path that do not meet your timing. What normally happens is that after synthesis the tool will give you a number of path which have a negative slag. The first thing you would do is to make sure those path are not false or multicycle since it that case you can just ignore them.

Taking a typical example (in a very simpler way), the STA tool will add the delay contributed from all the logic connecting the Q output of one flop to the D input of the next (including the CLK->Q of the first flop), and then compare it against the defined clock period of the CLK pins (assuming both flops are on the same clock, and taking into account the setup time of the second flop and the clock skew). This should be strictly less than the clock period defined for that clock. If the delay is less than the clock period, then the "path meets timing". If it is greater, than the "path fails timing" The "critical path" is the path out of all the possible paths that either exceeds its constraint by the largest amount, or, if all paths pass, then the one that comes closest to failing

5. Explain the following terms in detail. You may use diagrams to elaborate.

(i) Fanout Violation (ii) Transition Violation (iii) Capacitance Violation (iv) Max Length Violation (v) Noise Violation.

(i) Fanout Violation: The maximum number of input pin that an output pin can drive is called fanout, If the number of input drive pins exceeds the limit then it leads to the fanout violation.



Let assume in Design Rule Fanout the max fanout number is three.

But, here we can see the number of driving cell is five, so the number of input drive pins exceed the limit. so, here fanout volation is two.

https://www.linkedin.com/in/ajmain-ar-rafi/

(ii) Transition Violation: When a signal takes too long to transit from one logic level to another. It is occurs when the transmition time of a signal is too slow. This can happend when driving cell can't produce enough current for the loads.

It occurs in various way

- High output load
- Long routing delays
- Low driver strength
- Nonlinear effects

Transition violations can be fixed many ways:

- Upsizing the driver cell/Increasing the driver strength
- Reducing the output load by adding buffers

(iii) Capacitance Violation: A capacitance violation occurs when the capacitive load of a net exceeds the maximum capacitance that a driving cell can drive.

This can happen for a number of reasons, including:

- Increased fan-out
- Increased interconnect capacitance
- Increased cell capacitance

Capacitance violations can lead to a number of problems such as _

- Signal integrity problems
- Power consumption
- Timing violations

Capacitance violations can be fixed by a number of methods, including:

- **Adding buffers:** Add Buffer in a net means adding a buffer cell between the driving cell and the net. The buffer cell will help to increase the drive strength of the net and reduce the capacitive load.

- **Downsize of the cell:** Resizing a cell means changing the size of the cell. Resizing the cell can help to reduce the cell capacitance and the capacitive load of the nets that drive the cell.

(iv) Max Length Violation: When the length of a wire exceeds the maximum allowed length specified by the design rules.

This can happen for a number of reasons, including:

- The wire was routed too long
- The wire was routed around obstacles
- The wire was routed through a high-density area.

It can create a number of problems, including:

- **Timing violations:** If the wire is too long, it can take too long for the signal to travel from one end to the other
- **Power consumption:** Longer wires have more resistance, which can lead to increased power consumption.
- **Signal integrity:** Longer wires are more susceptible to noise, which can degrade the signal quality.

To fix this problem, we can do:

- **Rerouting the wire:** This may involve changing the routing path or the width of the wire.
- **Buffering the wire:** This involves adding a buffer to the wire, which can help to boost the signal strength and reduce the propagation delay.
- **Shortening the wire:** This may involve moving the components closer together or using a different routing technique.

(v) Noise Violation: Noise is any unwanted signal that can interface with the designed signal. It occurs when the noise level on a signal crosses a certain threshold. This can happen if the signal is routed near a noise source or the signal is amplified at any time.

This can happen for a number of reasons, including

- **Crosstalk:** Crosstalk is the coupling of noise from one signal to another. This can happen when two signals are routed close together, or when they share a common ground plane.
- **Parasitic capacitance:** Parasitic capacitance is the capacitance that exists between different parts of an integrated circuit (IC). This capacitance can store charge, which can then be released as noise.
- **Ground bounce:** Ground bounce is a phenomenon that occurs when the voltage on the ground plane of an IC fluctuates. This can be caused by a number of factors, including switching noise from neighboring signals, or by voltage drops in the power supply.
- **Power supply noise:** Power supply noise is noise that is present on the power supply lines of an IC. This can be caused by a number of factors, including switching noise from neighboring signals, or by voltage drops in the power supply.

Noise violations can lead to a number of problems, including:

- **Functional errors:** Noise can cause a signal to change value, which can lead to incorrect operation of the circuit.
- **Timing violations:** Noise can cause a signal to arrive late or early, which can lead to timing errors.
- **Power consumption:** Noise can increase the power consumption of an IC.

There are a number of techniques that can be used to prevent or mitigate noise violations, including:

- **Routing signals carefully:** Signals that are susceptible to noise should be routed carefully to minimize the amount of crosstalk.
- **Using guardbands:** Guardbands are areas of metal that are placed

6. What is the Antenna Effect? How do we prevent it?

The antenna effect is a phenomenon that can cause damage to the gate oxide of MOS transistors during the fabrication process, particularly when using plasma etching.

The antenna effect occurs when a large metal is directly connected to the gate of a transistor. The metal interconnect acts as an antenna, and it can collect charges from the plasma during etching. These charges might take rapid discharge through the gate oxide, and damage the gate oxide.

There are three basic techniques to prevent antenna effect

1. Metal hoping
2. Antenna Diode (Reverse diode)
3. Place physical cell

7. Explain the latch-up issue in brief. How do we prevent the latch-up issue?

Latch-up is a condition where a low impedance path is created between a supply and ground. When std cell triggered the low impedance path created even after the trigger is no longer present.

To prevent the latch-up issue we use Well tap cells or Tap cells are in the design. Well tap cells connect the nwell to VDD and p-substrate to VSS.

TAP CELL will reduce substrate & well resistance and that will help to avoid LATCHUP.

8. What is Clock Tree Synthesis? How many types of clock tree structures are there? Describe the PnR stages of a design that has no clock.

Clock tree synthesis is a automatic insertion of buffers/inverters along the clock paths of the ASIC design to balance the clock delay to all clock input.

CTS perform in order to balance the skew and minimize insertion delay.

There are four types of clock tree structures:

1. H-Tree
2. X-Tree
3. Binary-Tree
4. Fish-bone Tree

PnR flow of a design where has no clock

- Sanity Check
- Intialization
- Floorplan
- Placement
- Route

9. Explain why Clock Routing takes place before Signal Routing.

In design, clock path is the most critical, and CTS is done before signal routing to avoid congestion to the clock nets so we can achive a good skew for timing.

If routing is done first , it will have more freedom of tracks and it will use more routing resources and it lead to timing failure and thus to fix it, clock buffres may be added or upsizing/downsizing many happen which will result in higher cell utilization. this will further be a problem during ECOs.

10. What is Global Route and Detail Route.

Global route -

- Identifying routable path for the nets driving/driven pins in a shortest distance
- Does not consider DRC rules, which gives an overall view of routing and congested nets
- Assign layers to the nets
- It interconnect by breaking the routing portion of the design into rectangles called global routing cells (gcells)
- Avoid congested areas and also long detours
- Avoid routing over blockages

Nano/Details Routing -

- Detailed routing follows up with the track routed net segments and performs the complete DRC aware and timing driven routing
- It is the final routing for the design built after the CTS and the timing is freeze
- Filler Cells are adding before Detailed Routing
- Detail Routing is done after analyze the cause for congestion in the design, add density screen or change floopian etc.

11. Difference between Global Route and Detail Route. Why do we perform Special Routing?

Global route plans interconnect by breaking the routing portion of the design into rectangles called global routing cells (gcells) and assigning the signal nets to the gcells, whereas detail routing is to follow the global routing and perform the actual physical interconnections of ASIC design.

Special routing is used for balance power distribution for standard cells and macro by connection of VDD and VSS.

12. Decribe the main steps that take place during Placement stage. What do you mean by legalization?

Standard cell placement: The std cells must be placed in the assigned region (i.e. row) so that the ASIC can be routed efficiently and the overall timing requirements can be satisfied. The tool determines the location of each std cell instantiations on the die using algorithms. The main objective is to minimize the interconnect wire length as well meet the design rule constrains.

Legalization - All the std cells are must be placed in std cell rows. If there is overlap of any std cell or placed out of std cell rows or placed out the core area is called legalization violation.

13. What is a SPEF file and what information does it contain? Can we perform timing analysis without a SPEF file? Explain your answer.

SPEF is a Standard Parasitic Exchange Formula file, it contains the net capacitances and resistances of a metal. Without a SPEF file timing analysis is not possible because in timing analysis we need metal perasitic information.

14. How do we make sure that our design meets the rules that are required to be meet for the design to be fabricated?

In the stage of physical sign-off there are two types

1. DRC
2.LVS

DRC - When we check DRC violation we need input as routed GDS & rule file and in the output we will find our result if any violation show that's mean there are some place where drc rule didn't meet. And we have to clear DRC violation before fabrication.

We faced DRC violation in metal speaching, width, or layer

LVS - When we check LVS violation we need input as routed GDS layout fle & scematic file and in the output we will find our result if any violation show that's mean there are some place where LVS rule didn't meet. And we have to clear LVS violation before fabrication.

We faced LVS violation due to metal connectivity open, short.

15. In which physical verification method will the violation be reported if a design has opens and shorts?

In LVS verification method, if metal has opens and shorts it will violate and show the report.

16. State different types of variables that are used mostly in shell scripting.

There are two types of variables that used in shall scripting

1. Scalar Variables
2. Array Variables

Scalar variables can hold only one value at a time on the other hand Array variables can hold multiple values.

17. Explain ways to create shortcuts in linux? How do you check the type of shortcut that has been created?

Soft link: To create a soft shortcut link of the file or directory we have to use ln command

Command is: ln -s <path to file or directory> <path to shortcut>

Ex: To create a shortcut of the file on the home directory /home/user/myfile

Command will be:

```
ln -s /home/user/myfile  ~/myfile.desktop
```

By using ls -l we will find the shortcut that has been created the file or director show the orginal file link beside it

18. What is the shebang line in shell sciprting?

#!/bin/bash is a sequence of characters.

(#!) is called shebang and it is used to tell the linux kernel which interpreter to use when executing the script. This line, also known as a hashbang.

19. How do you execute a shell script?

1. Create a file with .sh extension using a text editor such as vi or vim or nano in linux.

    -> vi hello.sh

2. Write the script in the file using an editor.

```
#!/bin/bash
echo "Hello World"
```

3. Set the script executable with command chmod in linux

```
chmod u+x hello.sh
```

4. Execute the shell script in linux by following way
    - ./ hello.sh
    - sh hello.sh
    - source hello.sh

20. How many types of conditional commands are there in Shell Scripting? Write down their syntax.

There are 4 types of conditional commands we are use in Shell Scripting.

The syntax are:

1. if statement
2. if-else statement
3. elif statement
4. Nested if

1. if statement:

```
if [condition];
        then
        commands
fi
```

```
if [ "$name" = "John Doe" ];
        then
        echo "Hello, John Doe!"
fi
```

2. if-else statement

```
if [condition];
        then
        commands
        else
   commands
fi
```

```
if [ "$name" = "John Doe" ];
        then
        echo "Hello, John Doe!"
        else
   echo "I don't know who you are."
fi
```

https://www.linkedin.com/in/ajmain-ar-rafi/

3. elif statement

```
if [condition];
        then
                commands
elif [condition];
        then
                commands
fi
```

4. Nested if

```
if [condition1];
        then
        if [condition2];
            then    # statements to
                        be executed if both
                        conditions are true
        fi
        else        # statements to be
                        executed if condition1
                        is false
fi
```

```
read -p "Enter your name = " name

if [ -n "$name" ];
    then
        if ["$name" == "John Doe"];
        then
        echo "Hello, John Doe!"
        fi
    else
    echo "Please set the `name`
            variable."
fi
```

21. State the importance of sed command and awk command in Shell Scripting.

The sed and awk commands are two of the most powerful and versatile tools available for text processing in Linux.

The sed command is an essential tool for shell scripting, as it allows you to perform powerful text manipulation operations on files. Some of the most common uses of sed in shell scripting include:

1. Searching and replacing text

2. Inserting and deleting lines
3. Splitting and joining lines
4. Formatting text
5. Regular expressions

Here are some additional examples of how sed can be used in shell scripting:

To search for all lines in a file that contain the word "hello", so we can use the following command:

```
sed -n '/hello/p' file.txt
```

To replace all "hello" word with the word "goodbye" in a file, we can use the following command:

```
sed 's/hello/goodbye/g' file.txt
```

To insert the line "This is a new line" at the end of a file, we can use the following command:

```
sed '$ a This is a new line' file.txt
```

To delete all lines in a file that start with the word "comment", we can use the following command:

```
sed '/^comment/d' file.txt
```

On the other hand , The awk command is an important and versatile tool that can be used for many different tasks related to text processing. It is a powerful tool that can be used to search for patterns in text, extract data from text, and perform calculations on text. Awk is a powerful tool that can be used to automate many tasks related to text processing.

Here are some of the importance of awk command in shell scripting:

- Awk can be used to search for patterns in text.
- Awk can be used to extract data from text.
- Awk can be used to perform calculations on text.

We can use awk to create a report that summarizes the data in a file.
We can use awk to filter a file to only include lines that match a certain criteria.

23. What do you mean by the sensitivity list? Which level of abstraction in verilog coding uses the sensitivity list?

The sensitivity list is a compact way of specifying the set of signals, events on which may resume a process.

```
always @(posedge clk or negedge reset) begin
    if (reset) begin
      // Reset the circuit
    end
      else begin
       // Update the circuit's state based on the current inputs
      end
    end
```

In this example, the sensitivity list are clk & reset

The sensitivity list is only used in the behavioral level of abstraction.

24. Difference between $monitor and $display in testbench?

$display is a system tasks used for displaying values of variables or strings during simulation in a testbench

Whereas

$monitor is a system task that continuously monitors expressions and displays their values whenever they changes.

25. Write down the verilog code that will swap contents of two registers with and without a temporary register.

Verilog code for swap content

With temporary register

```
always @ (posedege clock)
begain
temp = b;
b = a;
a = temp;
end
```

This process known as Blocking behavior

Without temporary register

```
always @ (posedege clock)
begain
a <= b;
b <= a
end
```

This process known as Non-Blocking behavior

26. What does the line "timescale 1ns/1ps" signify in a testbench?

1ns – This specifies the base time unit, which is 1 nanosecond. This means that all time-related values and delays in the simulation will be interpreted in nanoseconds unless otherwise specified.

1ps – This specifies the time precision, which is 1 picosecond. The time precision determines the smallest time increment. In this case, it's 1 picosecond, which is a very small time interval.

So, the line "timescale 1ns/1ps" indicates that the simulation is operating with a time unit of 1 nanosecond and a time precision of 1 picosecond.

27. Write down the verilog code for a D-Flip Flop.

D Flip-Flop with Synchronous Reset Low Level:

```
module DFF_sync(d,clk,rstn,q);
input D;
input clk;
input rstn;
output reg q;
always @(posedge clk)
begin
    if (rstn==1'b1)
        q <= 1'b0;
    else
        q <= d;
end
endmodule
```

D Flip-Flop with Asynchronous Reset Low Level:

```
module DFF_async(d,clk,rstn,q);
input D;
input clk;
input rstn;
output reg q;
always @(posedge clk or negedge rstn)
begin
    if(rstn==1'b0)
        q <= 1'b0;
    else
        q <= d;
end
endmodule
```

28. Write down the verilog code for a D-Latch.

A D-latch is a digital circuit that can be used to store one bit of data. It is a type of latch, which is a circuit that can store data temporarily. The D latch gets its name from the "data" input, which is used to store the data that the latch will output.

```
module d_latch (d, clk, rst, q)
  input wire d,
  input wire clk,
  input wire rst,
  output reg q;
```

```
always @(posedge clk or posedge rst)
          begin
          if (rst) begin
                   q <= 0;
          end else begin
                   q <= d;
          end
  end
endmodule
```

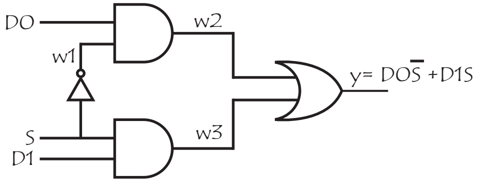
29. Explain the term "DUT" with respect to testbenches.

DUT stands for Device Under Test. It is the hardware design that is being tested by the testbench. The testbench applies inputs to the DUT and verifies that the DUT produces the correct outputs.

The DUT is typically implemented in a hardware description language (HDL) such as Verilog or VHDL. The testbench is also implemented in HDL, and it communicates with the DUT through a set of interfaces. And there are two main types of interfaces:

- Drivers: Drivers generate the input signals for the DUT.
- Monitors: Monitors capture the output signals from the DUT.

30. Write down the verilog code for a mutiplexer using ONLY dataflow modelling.



```
module MUX_2*1 (y,DO,D1,S);
input DO, D1, S;
output y;
wire w1, w2, w3;

assign y = (DO & ~S) | (D1 & S);
endmodule
```

31. Describe Setup and Hold Violations in detail. Use waveforms to explain your answer.

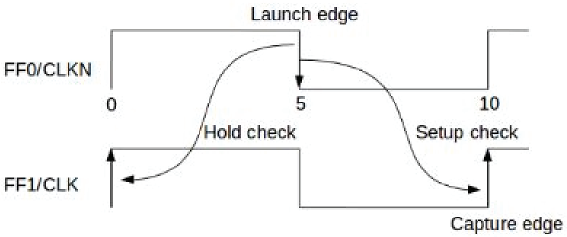
Setup time – Setup time are the minimum durations that a data signal must be stable before the clock edge, for a sequential element (such as a flip-flop or a latch) to capture it correctly.

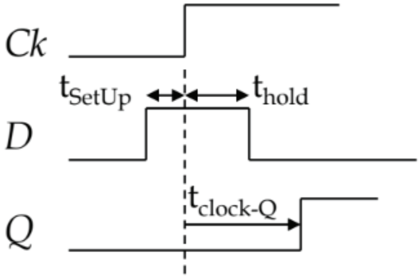
◇ Setup violation occurs when data-path is slow compared to the clock captured at capture flop.

◇ Setup violations can happen as a result of slow conditions (slow process, high temperature) leading to signals arriving too late in the clock period.

Setup slack time = Required time – Arrival time

T(setup/slack) = Tr –Ta





Hold time – Hold time are the maximum durations that a data signal must be stable after the clock edge for a sequential element (such as a flip-flop or a latch) to capture it correctly.

◇ Hold violation occurs when data-path is fast compared to the clock captured at capture flop.

◇ Hold violations can happen as a result of fast conditions (fast process, low temperature) leading to signals arriving too early in the clock period.

Hold slack time = Arrival time – Required time

T(hold/slack) = Ts –Tr

32. What are the placement goals?

Goals of placement

1. Area and timing optimization
2. Minimal cell and pin density
3. Routable design
4. Minimize the wire length
5. Minimize the power consumption
6. Minimize congstion

33. What are the macro placement qulities?

A good macro placement have following qualities:

1. Provide better utilization
2. Provide a compact layout
3. Allow us to make robust power planning for macros
4. Reduce routing congstion
5. Make timing closure easy

34. What is Skew & Slack?

Skew - Defference between clock arrival time in two different sequential element (such as a flip-flop or a latch)

Slack - Defference between data required time and data arrival time.

35. How to avoid setup and hold time violation?

Setup violations - It can happen as a result of slow conditions (slow process, high temperature) leading to signals arriving too late in the clock period.

To avoid setup violation we must follow the Strategy to Reduce Delay.

1. Increase Wire Thickness (reduces resistance) and increase the spacing between wires (decreases coupling capacitance).

2. Use lower Threshold Voltages (LVT) cells: Cells with lower threshold voltage have lesser transition times which makes the propagation of logic faster.

3. Better drive strength cell: A cell with better drive strength can charge the load capacitance quickly, resulting in lesser propagation delay. So make sure cell has better drive strength. if it is not we need to upsize the cell so that circuit gain a better driving streangth.

Hold Violation - It can happen as a result of fast conditions (fast process, low temperature) leading to signals arriving too early in the clock period.

To avoid setup violation we must follow the Strategy to Reduce Delay.

1. Insert buffers. The timing path where hold violation is occurring, if the delay is increased due to these buffers, then it shall ultimately lead to positive slack thereby improving the chances of hold time being met.

2. Use higher threshold voltages (HVT) cells: If you have multiple varieties of cells with variable threshold voltages, then the cells with higher threshold voltage will have higher transition times. This reduces the chances of the hold time being violated.

36. How do I fix setup or hold violation during placement?

Drive strength is the capacity of a cell to drive a value to the cell connected to its output.

Different sizes of std cells have different capacitance, smaller cells have small capacitance and vice versa. Its easier to drive a small cell than a large one. The load connected at the output of a cell is the combined capacitance of all the cells connected at the output.

So larger the load, larger is the drive required to "force" the values at the output.

Hence the fanout of a cell is usually kept low to avoid driving problems.

At placement stage, one should only focus on fixing setup violations as the clock tree is still not built at that phase in the design flow.

Setup violations at placement stage can be fixed using the following ways:

- Optimise data path with less depth as much as possible.
- Have better drive strength cells in the data path.
- Set the timing optimisation effort level to high (assuming congestion is under control and the run time trade off is reasonable).
- Have regions defined to place specific timing critical logic in a specified area to reduce the wire length.