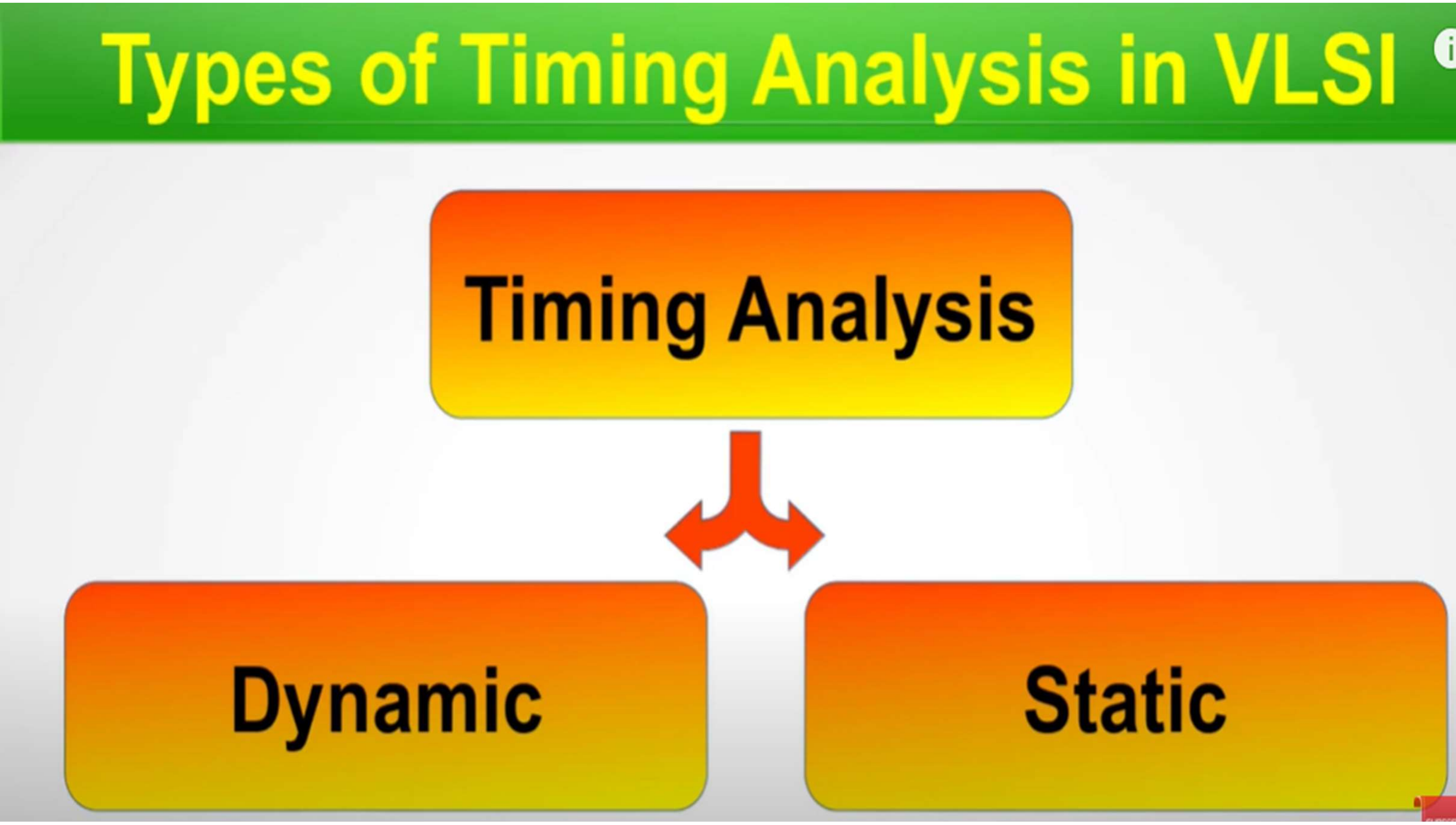
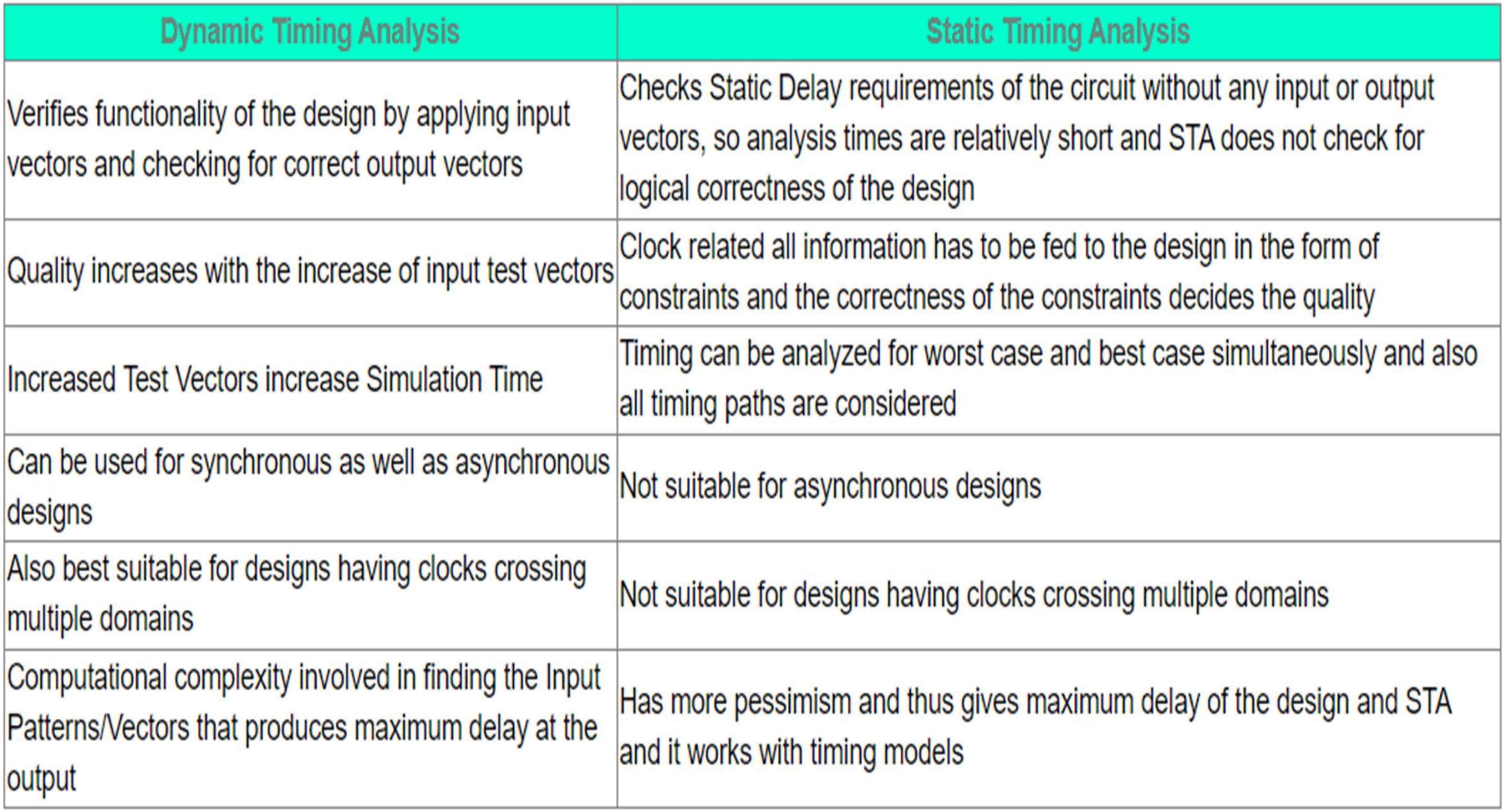
Timing Analysis

Unit -6

by
Dr. SAKTHIVEL.S.M



Types of Timing Analysis in VLSI



Dynamic Timing Analysis	Static Timing Analysis
Verifies functionality of the design by applying input vectors and checking for correct output vectors	Checks Static Delay requirements of the circuit without any input or output vectors, so analysis times are relatively short and STA does not check for logical correctness of the design
Quality increases with the increase of input test vectors	Clock related all information has to be fed to the design in the form of constraints and the correctness of the constraints decides the quality
Increased Test Vectors increase Simulation Time	Timing can be analyzed for worst case and best case simultaneously and also all timing paths are considered
Can be used for synchronous as well as asynchronous designs	Not suitable for asynchronous designs
Also best suitable for designs having clocks crossing multiple domains	Not suitable for designs having clocks crossing multiple domains
Computational complexity involved in finding the Input Patterns/Vectors that produces maximum delay at the output	Has more pessimism and thus gives maximum delay of the design and STA and it works with timing models

Static Timing Analysis

- Effective methodology for verifying the timing characteristics of a design without the use of test vectors
- Static Timing Analysis can be done only for Register-Transfer-Logic (RTL) designs
- Functionality of the design must be cleared before the design is subjected to STA
- STA approach typically takes a fraction of the time it takes to run logic simulation

STA is basically method of adding the net delays and cell delays to obtain path delays.
then STA tool analyzes all paths from each and every start point to each and every end point and compares it against the constraint(timing specification) that exists for that path.

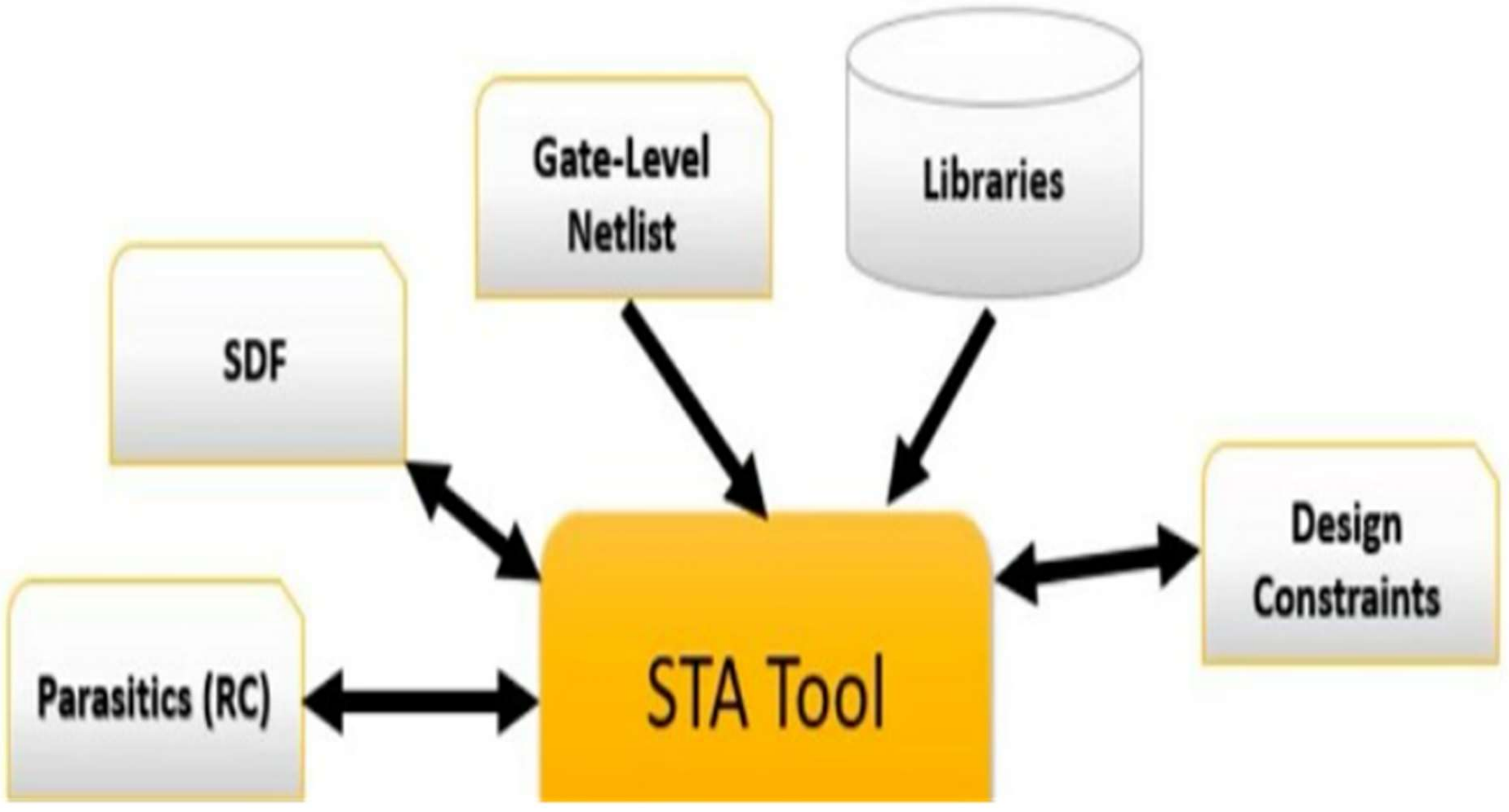
Purpose of Static Timing Analysis

- Fist, STA calculates the path delays for optimization tools. then based on the path delays, the optimization tool chooses cells from the timing library to create a circuit that meets your timing requirement.
- Second, STA analyzes the timing of a circuit to verify that the circuit works at the specified frequency.

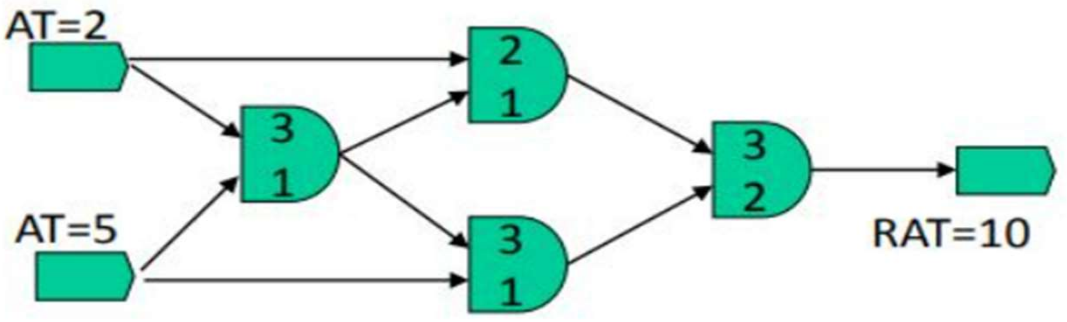
Main steps of STA

- Break the design into sets of timing paths
- Calculate the delay of each path
- Check all path delays to see if the given timing constraints are met

STA inputs and outputs



STA EXAMPLE



Path-based:

2+2+3 = 7 (OK)

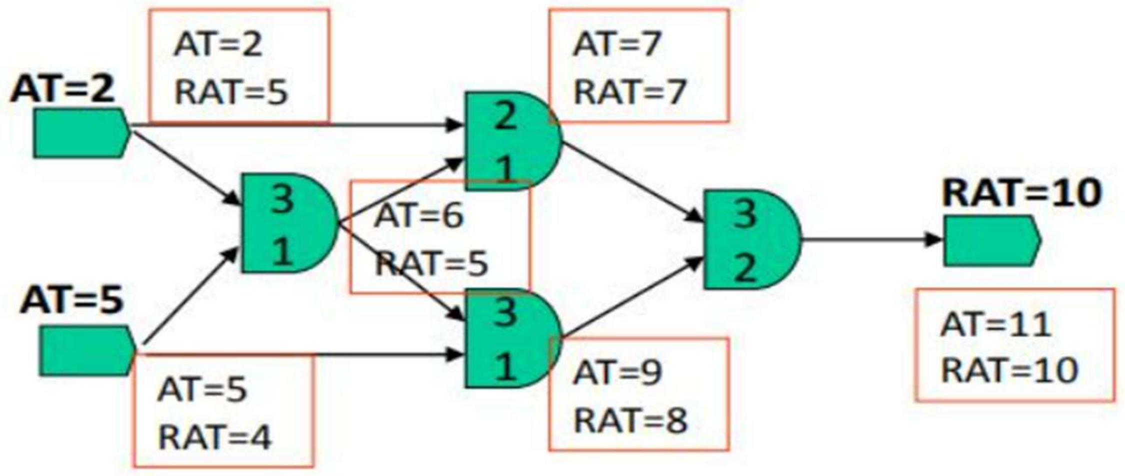
2+3+1+3 = 9 (OK)

2+3+3+2 = 10 (OK)

5+1+1+3 = 10 (OK)

5+1+3+2 = 11 (Problem!)

5+1+2 = 8 (OK)



Block-based:

Critical path is determined as collection of gates with the same, negative slack:

Slack = RT – AT

In our case, we see one Critical path with slack = -1

Types of Path under STA Scanner

Four Types of Path :

1.Input to Register

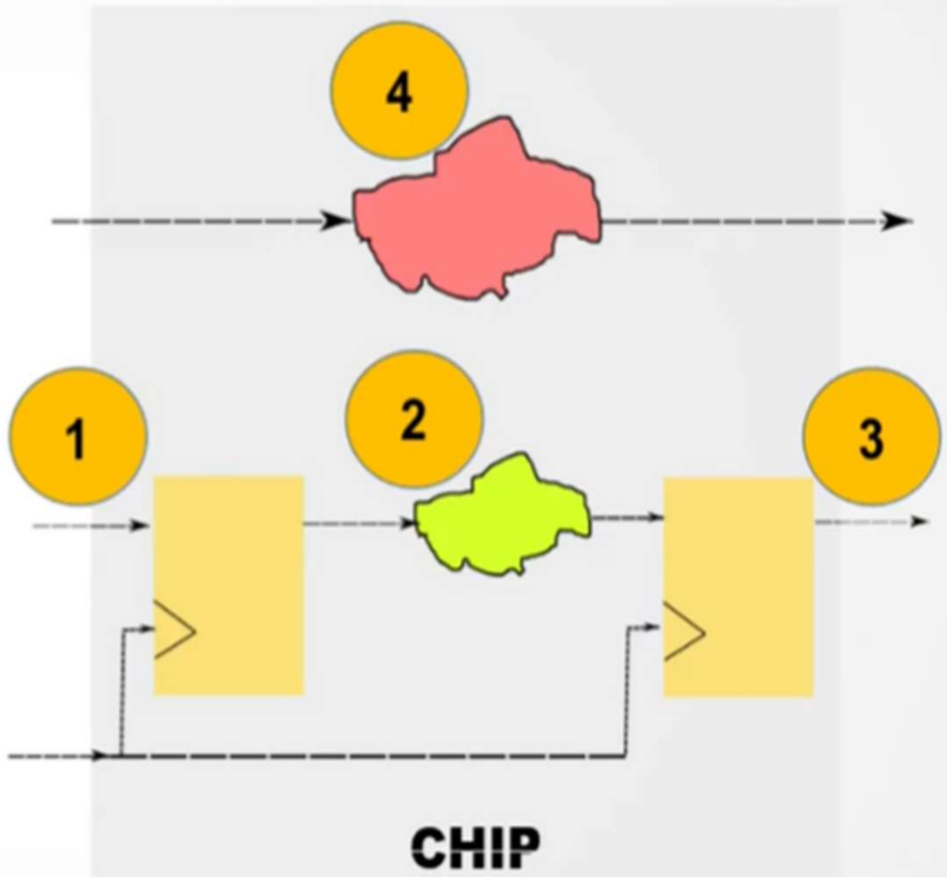
2.Register to Register

3.Register to Output

4.Input to Output

Each Path has :

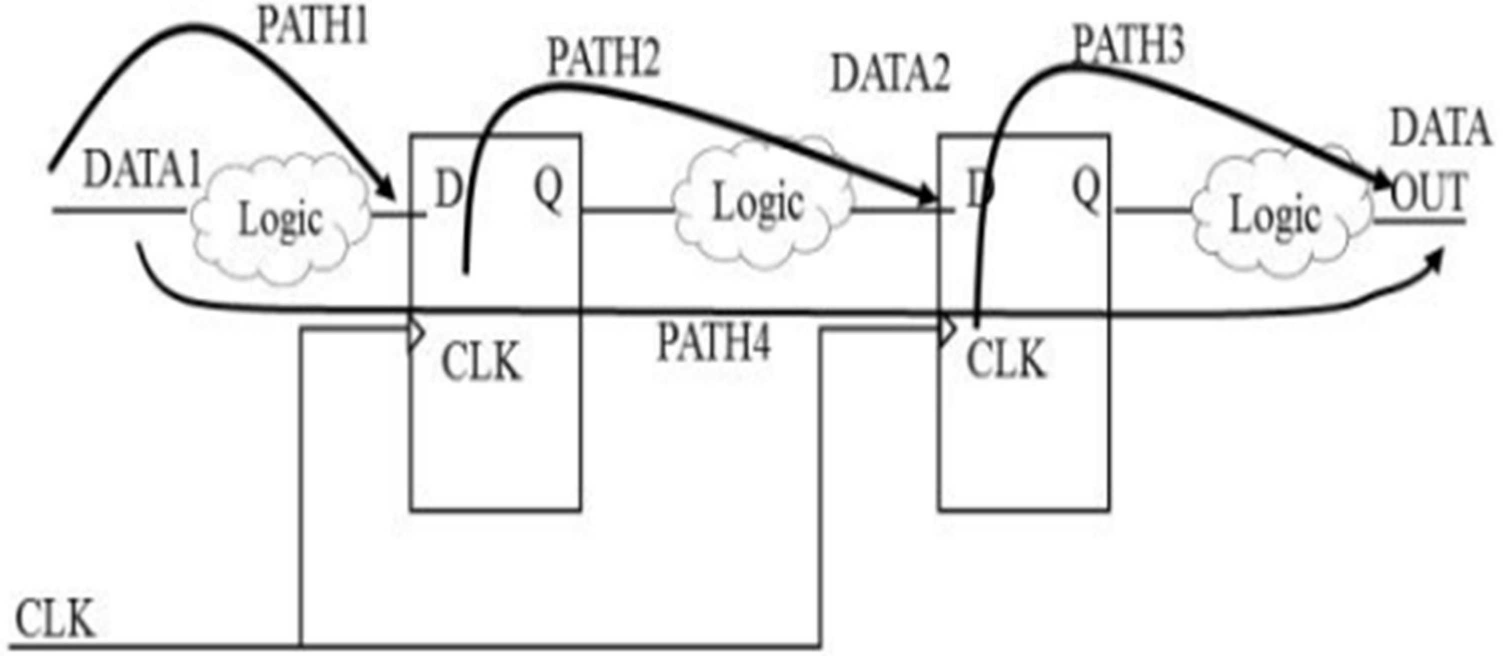
- Start point
- End Point



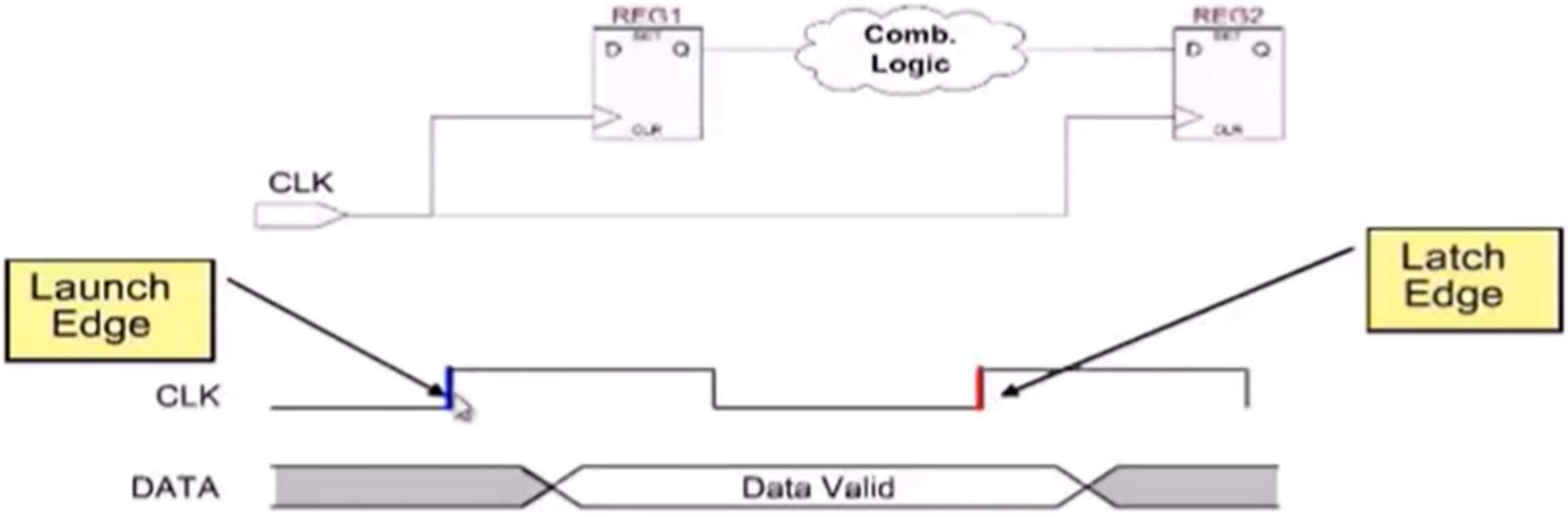
Timing Paths

A Timing Path is a point-to-point path in a design which can propagate data from one flip-flop to another

- Each path has a start point and an end point
- Start point: Input ports or Clock pins of flip-flops
- Endpoints: Output ports or Data input pins of flip-flops



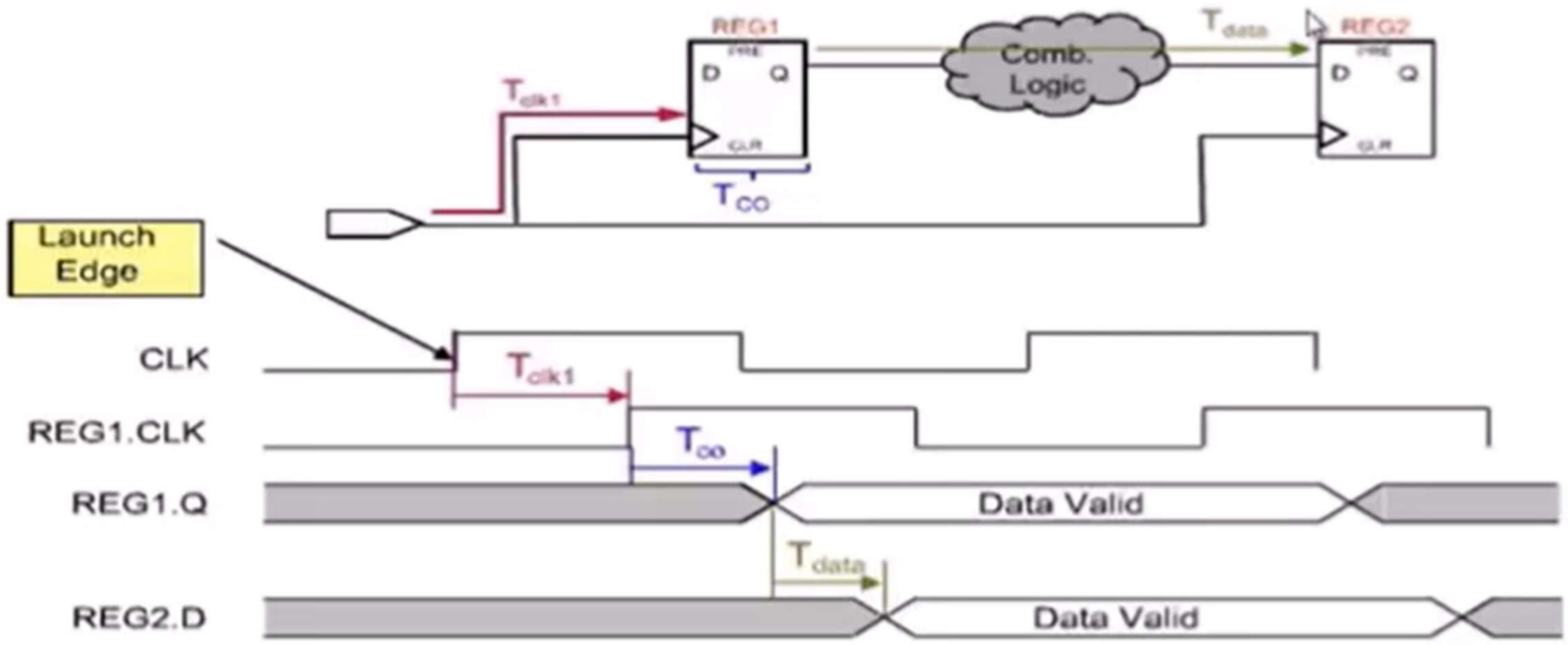
LAUNCH EDGE & LATCH EDGE



Launch Edge:	the edge which "launches" the data from source register
Latch Edge:	the edge which "latches" the data at destination register (with respect to the launch edge, typically 1 cycle)

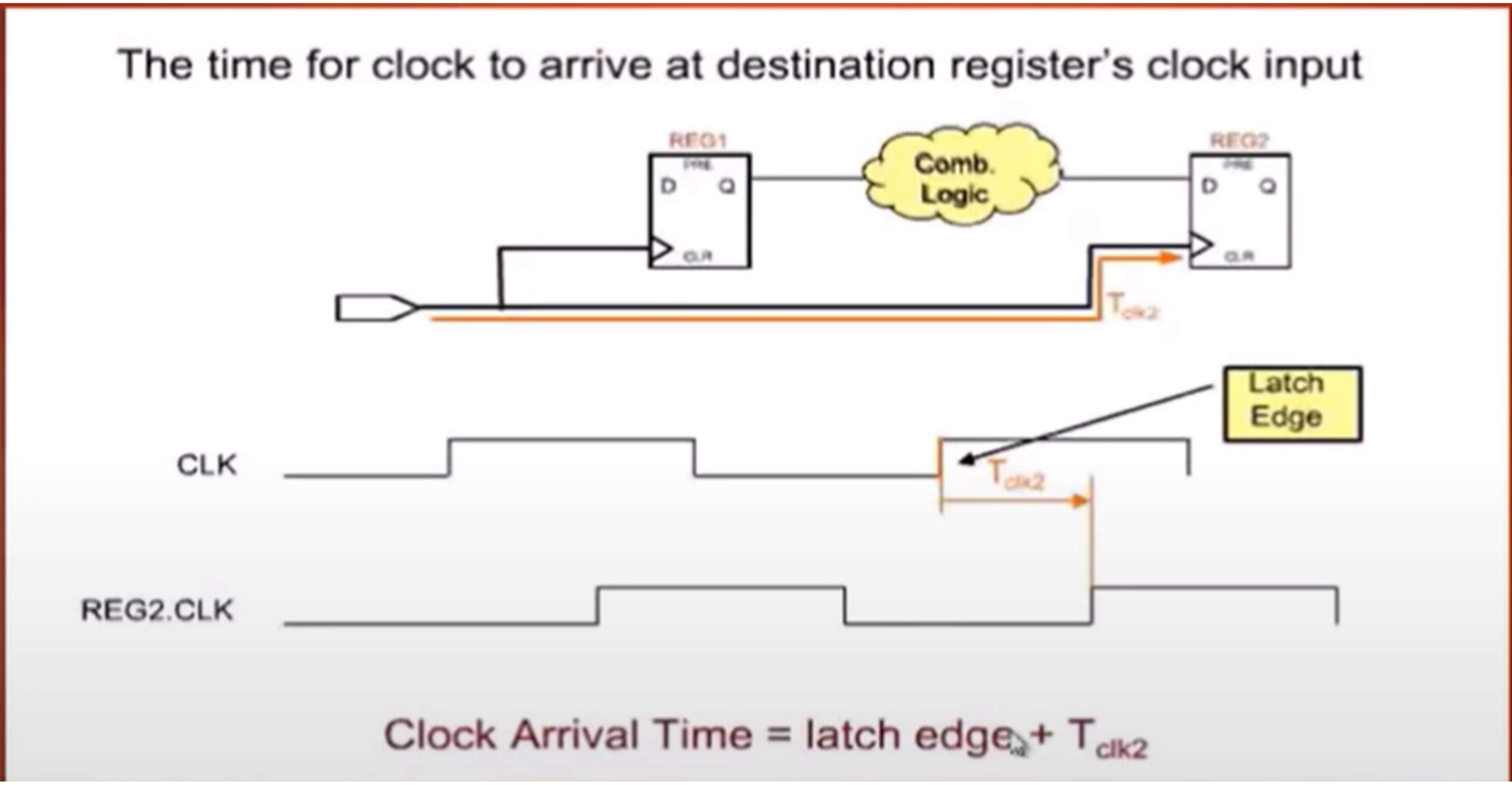
DATA ARRIVAL TIME

The time for data to arrive at destination register's D input



Data Arrival Time = launch edge + T_{clk1} + T_{co} + T_{data}

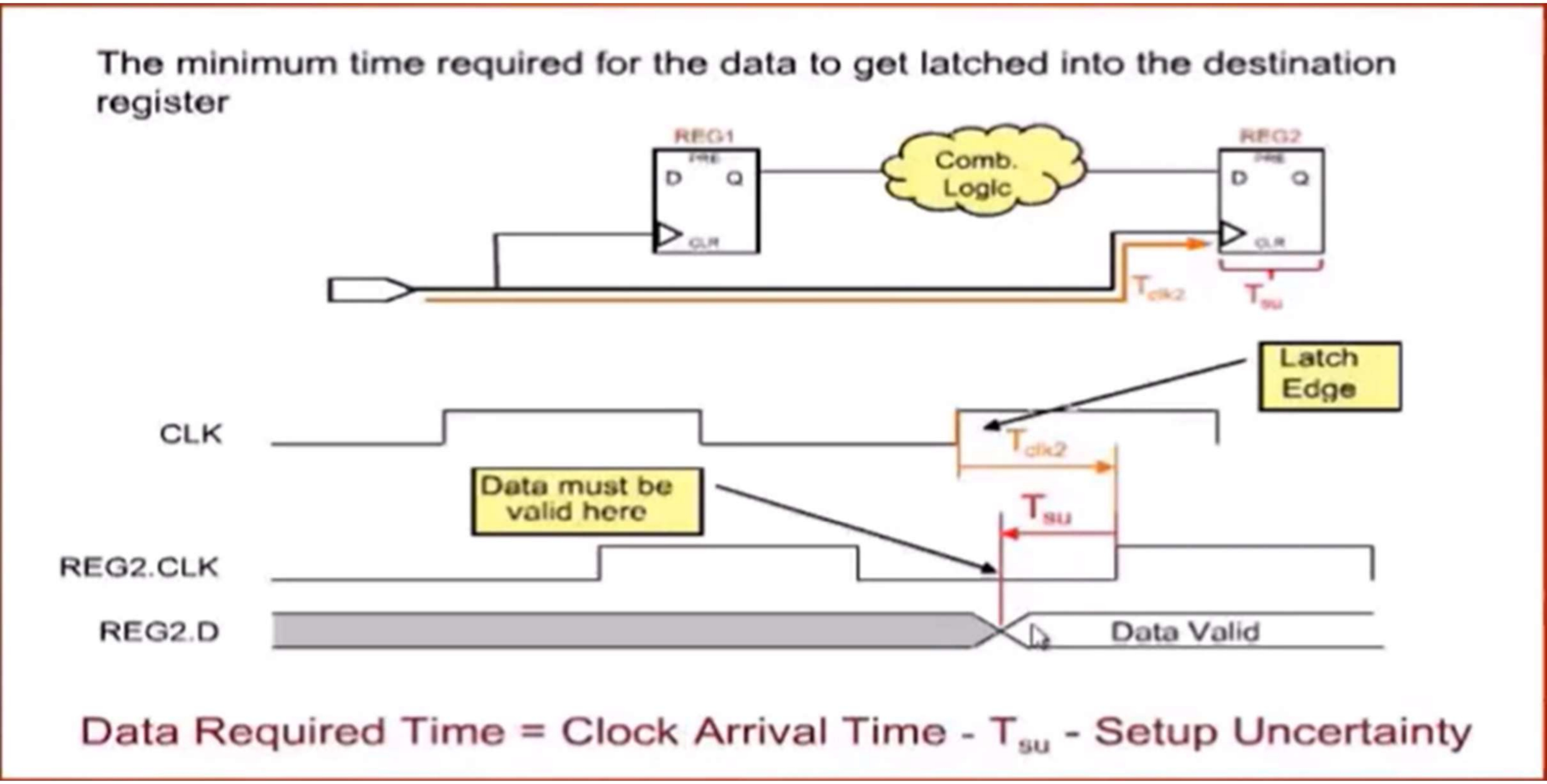
Clock Arrival Time



The time for clock to arrive at destination register's clock input

Clock Arrival Time = latch edge + T_{clk2}

DATA REQUIRED TIME - SETUP

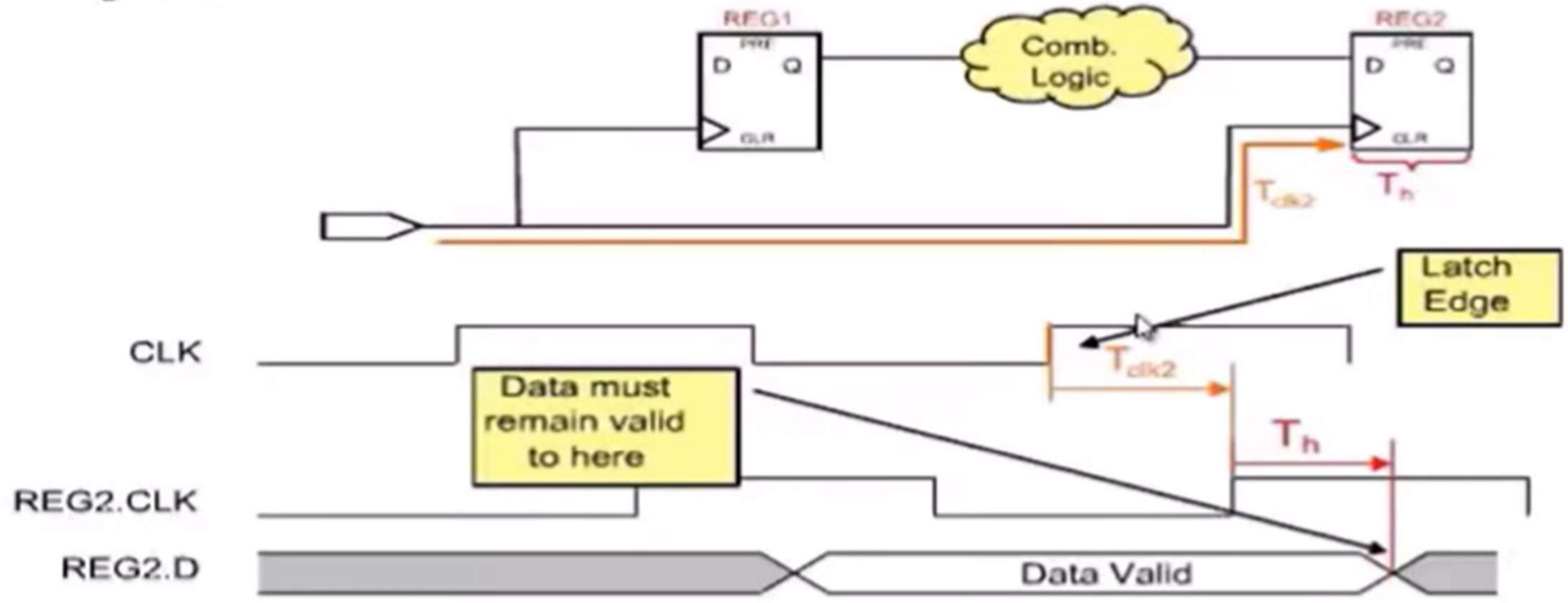


The minimum time required for the data to get latched into the destination register

Data Required Time = Clock Arrival Time - T_{su} - Setup Uncertainty

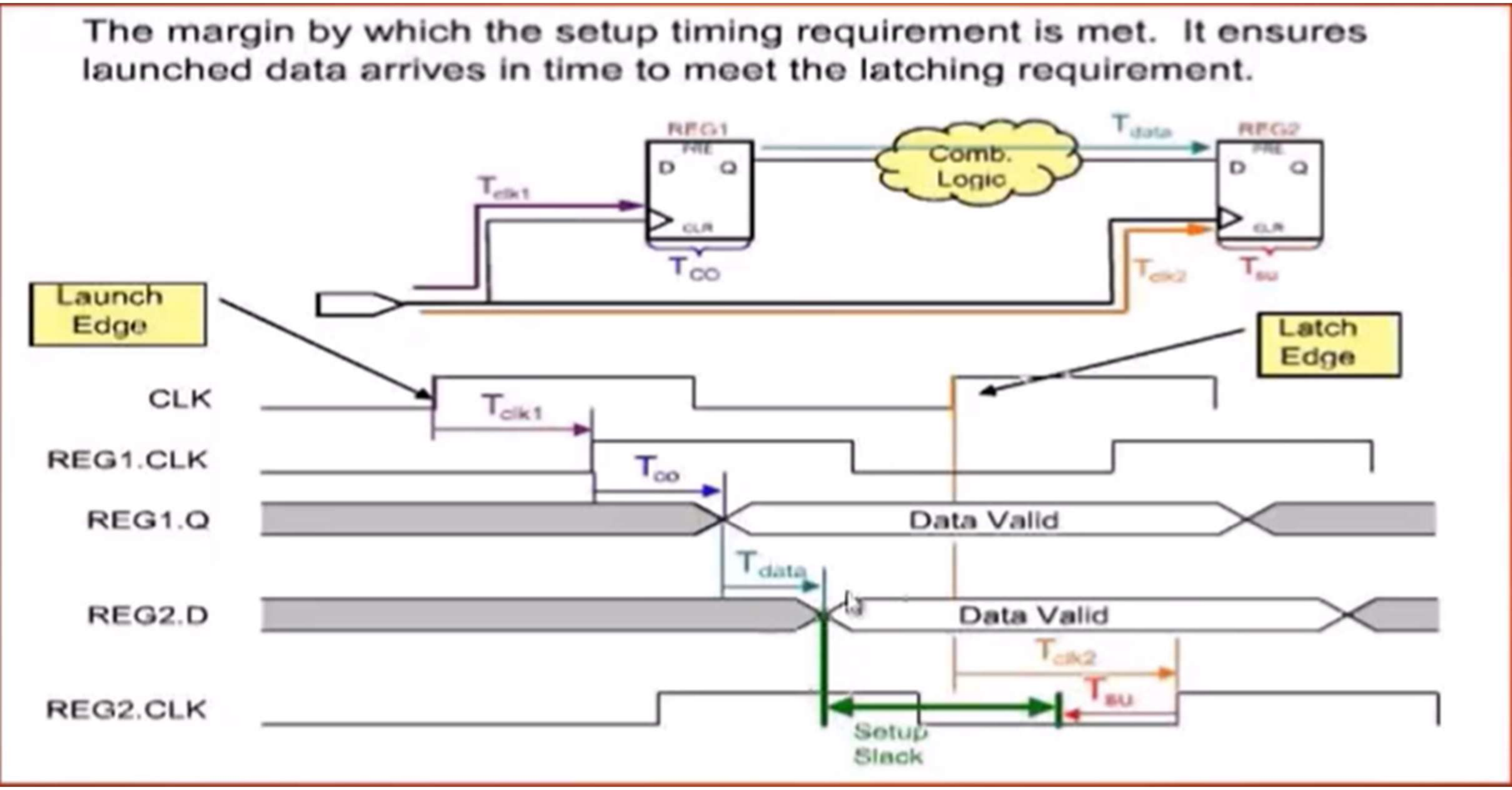
DATA REQUIRED TIME- HOLD

The minimum time required for the data to get latched into the destination register



Data Required Time = Clock Arrival Time + T_h + Hold Uncertainty

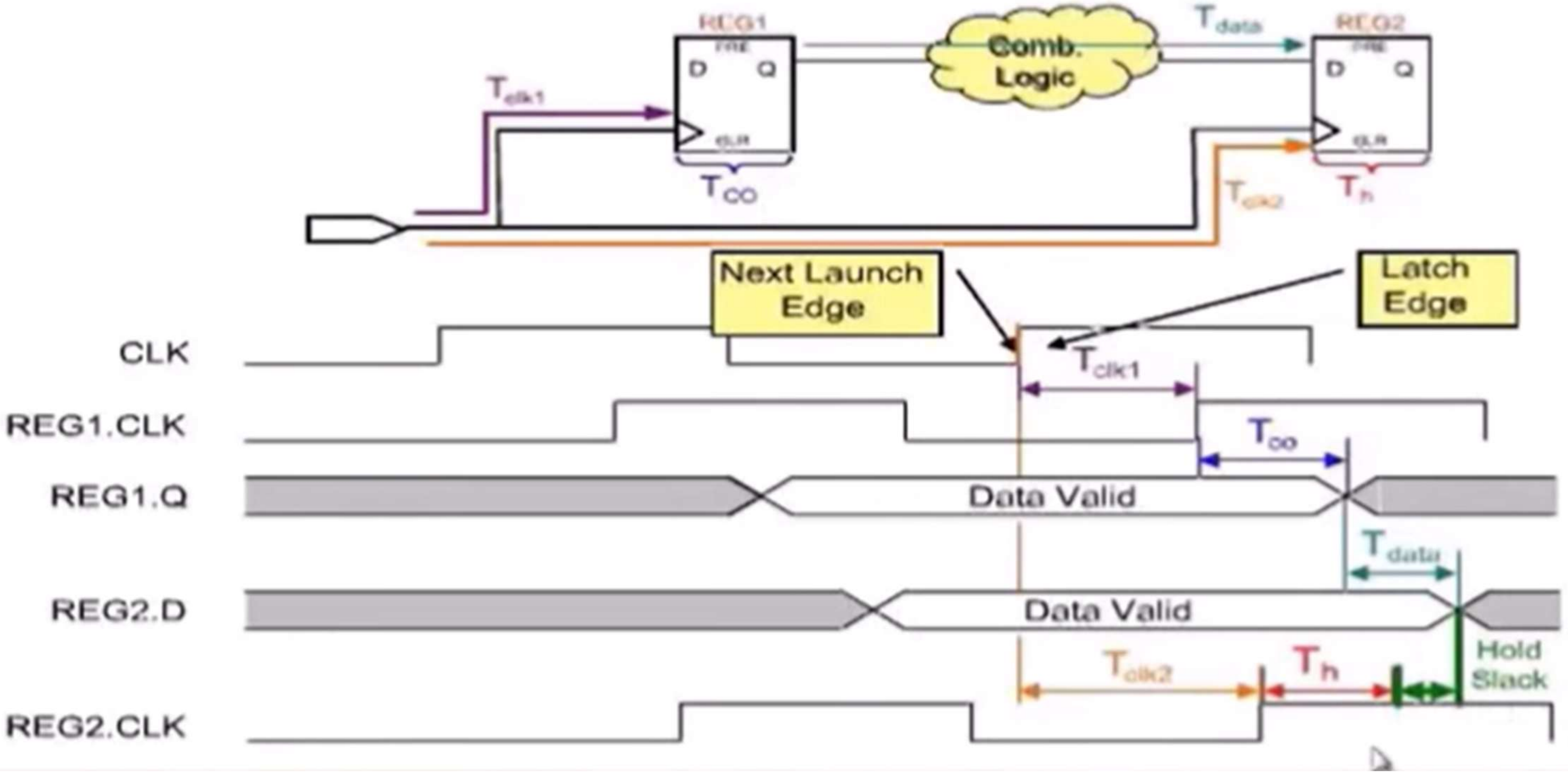
SETUP SLACK



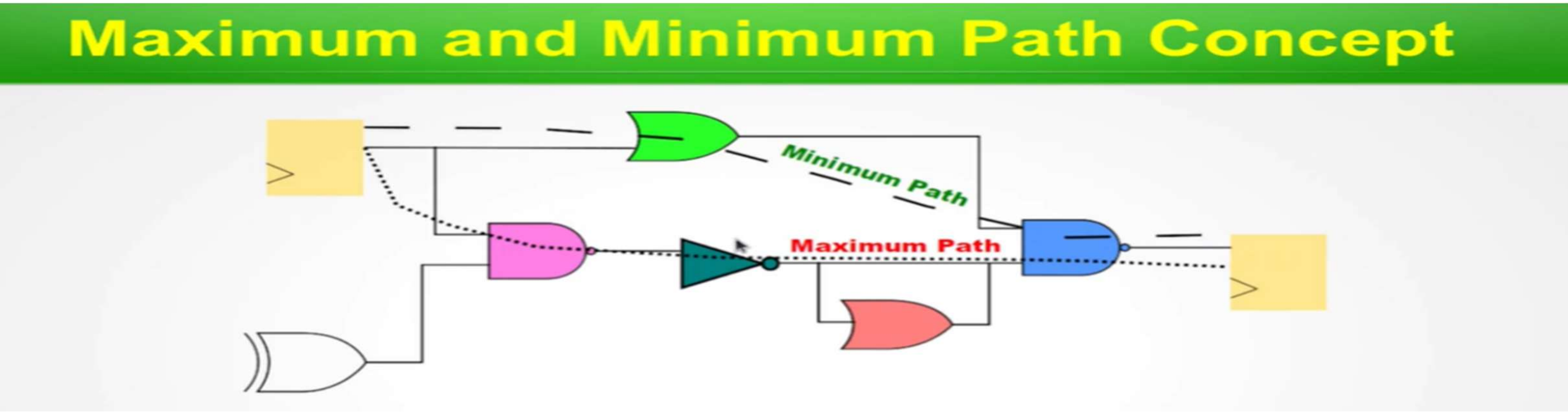
The margin by which the setup timing requirement is met. It ensures launched data arrives in time to meet the latching requirement.

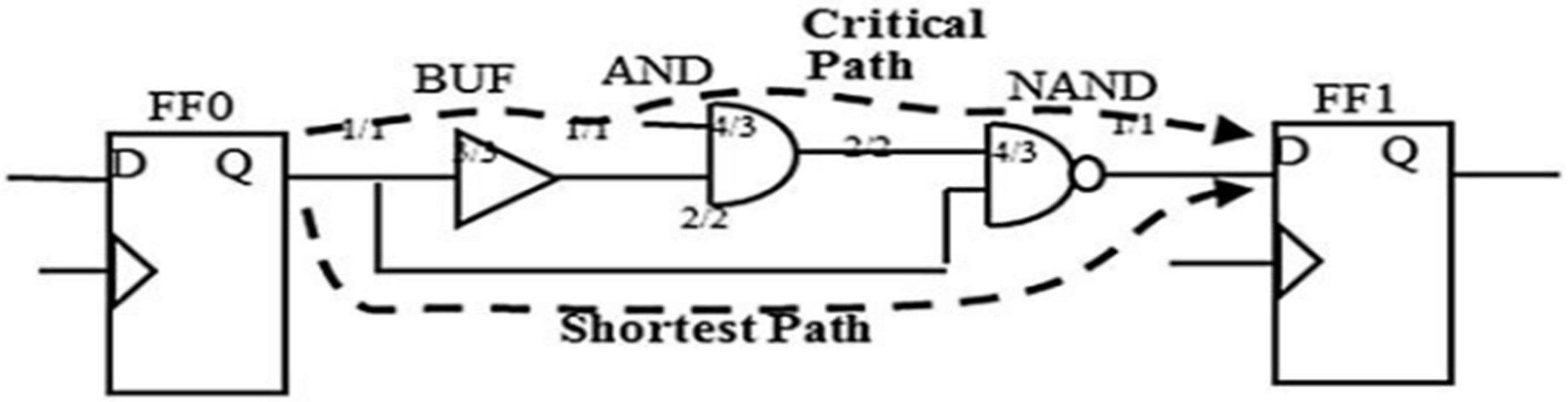
HOLD SLACK

The margin by which the hold timing requirement is met. It ensures latch data is not corrupted by data from another launch edge. It also prevents "double-clocking".

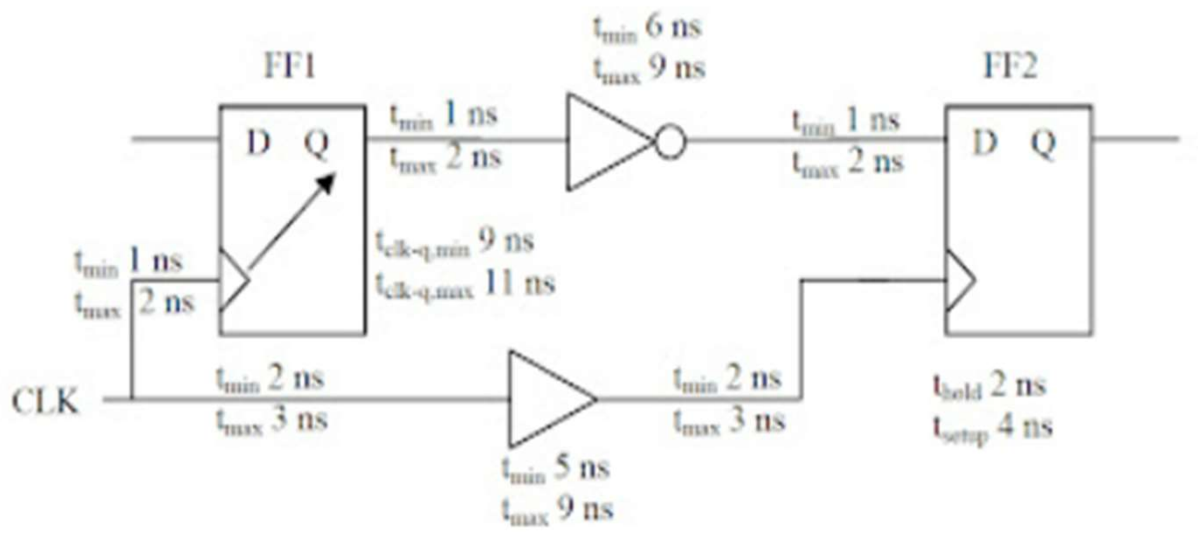


Critical & Shortest Path





Example of SETUP & HOLD VILOATIONS



Example of SETUP & HOLD VILOATIONS

Setup Analysis:

When a setup check is performed, we have to consider two things-

- Maximum Delay along the data path.
- Minimum Delay along the clock path.

If the difference between the clock path and the data path is negative, then a timing violation has occurred. (Note: there are few Exceptions for this- We will discuss that some other time)

Data path is: CLK->FF1/CLK ->FF1/Q ->Inverter ->FF2/D

Delay in Data path

= max(wire delay to the clock input of FF1) + max(Clk-to-Q delay of FF1) +max(cell delay of inverter) + max(2 wire delay- "Qof FF1-to-inverter" and "inverter-to-D of FF2")

=Td = 2+11+9+(2+2) = 26ns

*Note: The first part of the clock path delay (during setup calculation) is the clock period, which has been set to 15 ns. Hope You remember in last blog, I have mentioned very clearly that **Setup is checked at the next clock cycle.** That's the reason for clock path delay we have to include clock period also.*

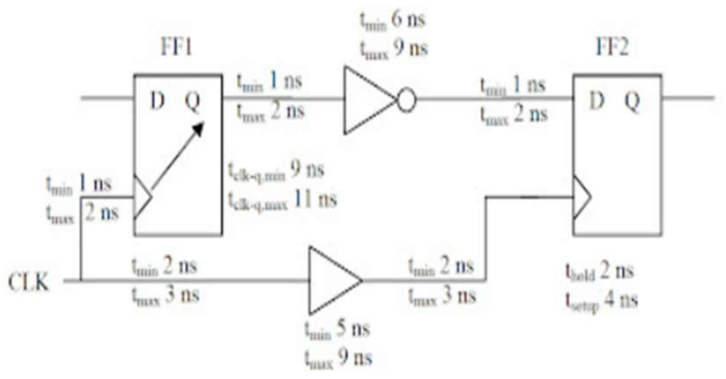
Clock path is: CLK-> buffer -> FF2/CLK

Clock path Delay

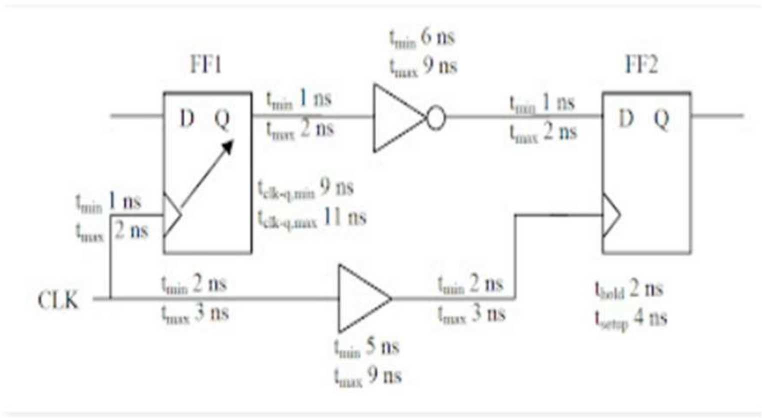
= (Clock period) + min(wire delay from CLK to Buffer input) + min(cell delay of Buffer) + min(wire delay from Buffer output to FF2/CLK pin) - (Setup time of FF2)

=Tclk = 15+2+5+2-4=20ns

Setup Slack = Tclk - Td = 20ns - 26ns = -6ns.
Since Setup Slack is negative -> Setup violation.



Example of SETUP & HOLD VILOATIONS



Hold Analysis:

When a hold check is performed, we have to consider two things-

- Minimum Delay along the data path.
- Maximum Delay along the clock path.

If the difference between the data path and the clock path is negative, then a timing violation has occurred. (Note: there are few Exceptions for this- We will discuss that some other time)

Data path is: CLK->FF1/CLK ->FF1/Q ->Inverter ->FF2/D

Delay in Data path

= min(wire delay to the clock input of FF1) + min(Clk-to-Q delay of FF1) +min(cell delay of inverter) + min(2 wire delay- "Qof FF1-to-inverter" and "inverter-to-D of FF2")

=Td = 1+9+6+(1+1)=18ns

Clock path is: CLK-> buffer -> FF2/CLK

Clock path Delay

= max(wire delay from CLK to Buffer input) + max(cell delay of Buffer) + max(wire delay from Buffer output to FF2/CLK pin) + (hold time of FF2)

=Tclk = 3+9+3+2 = 17 ns

Hold Slack = Td - Tclk = 18ns -17ns = 1ns

Since Hold Slack is positive-> No hold Violation.

<u>Note: A bigger clock period or a less maximum delay of the inverter solve this setup violations in the circuit.</u>

E.g

If Clock period is 22ns then

Tclk = 22+2+5+2-4=31-4=27ns AND Td = 26ns

Setup Slack = Tclk - Td = 27-26=1ns (No Violation)