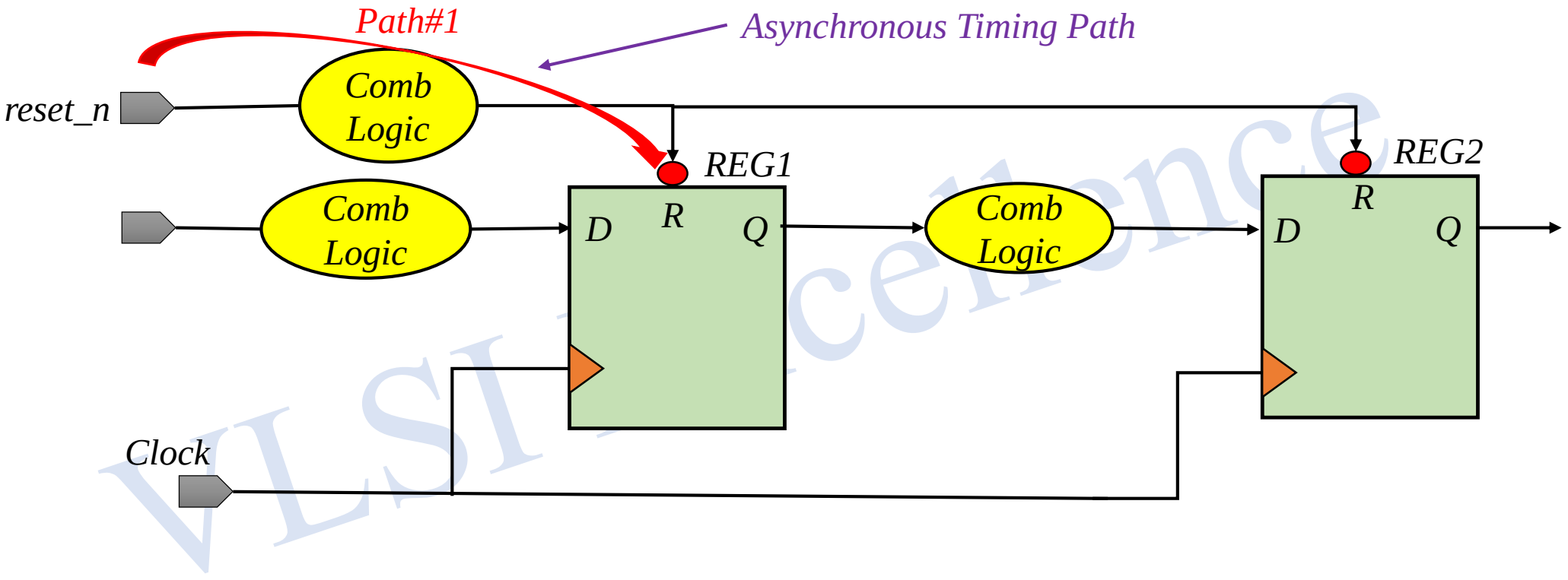
Static Timing Analysis (STA)

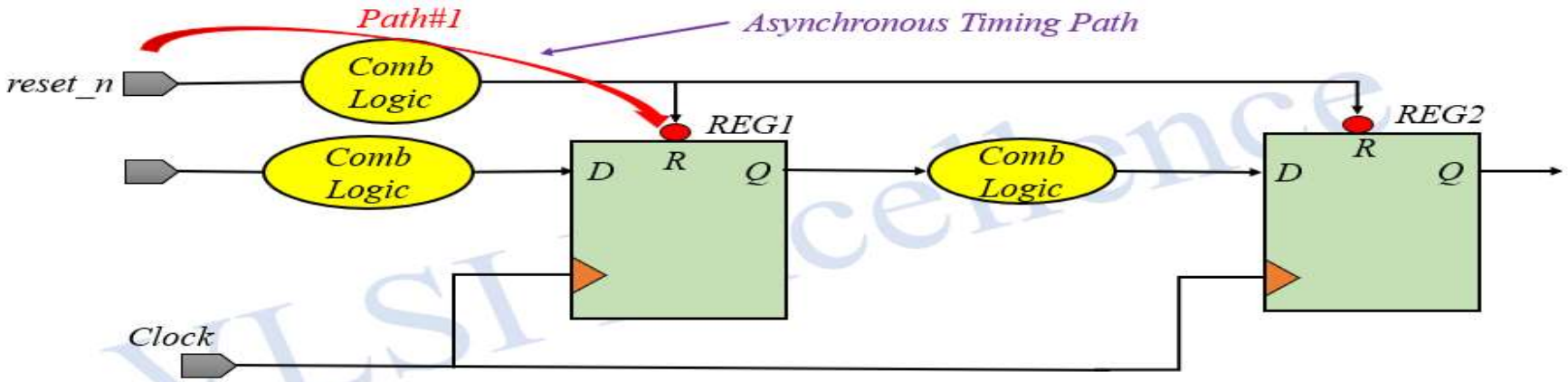
Lecture #15: Asynchronous Timing Checks

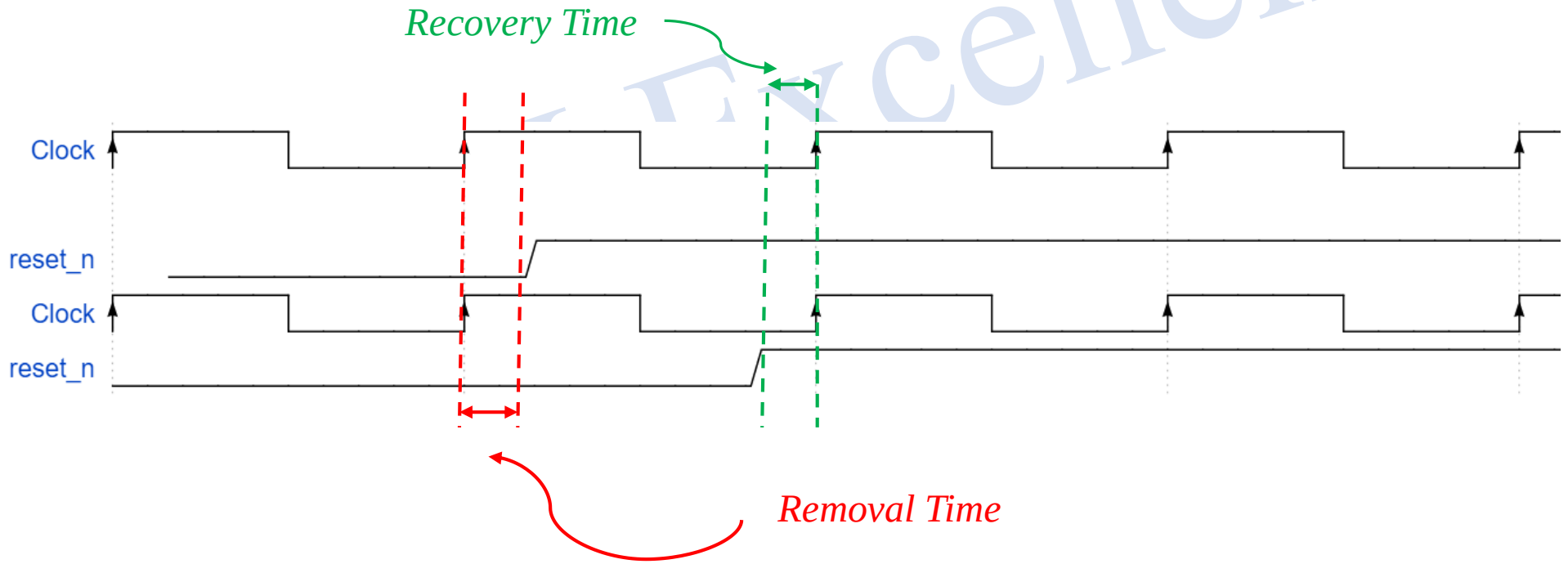
Video Lecture Link

Static Timing Analysis (STA) – Asynchronous Timing Checks



Static Timing Analysis (STA) – Asynchronous Timing Checks

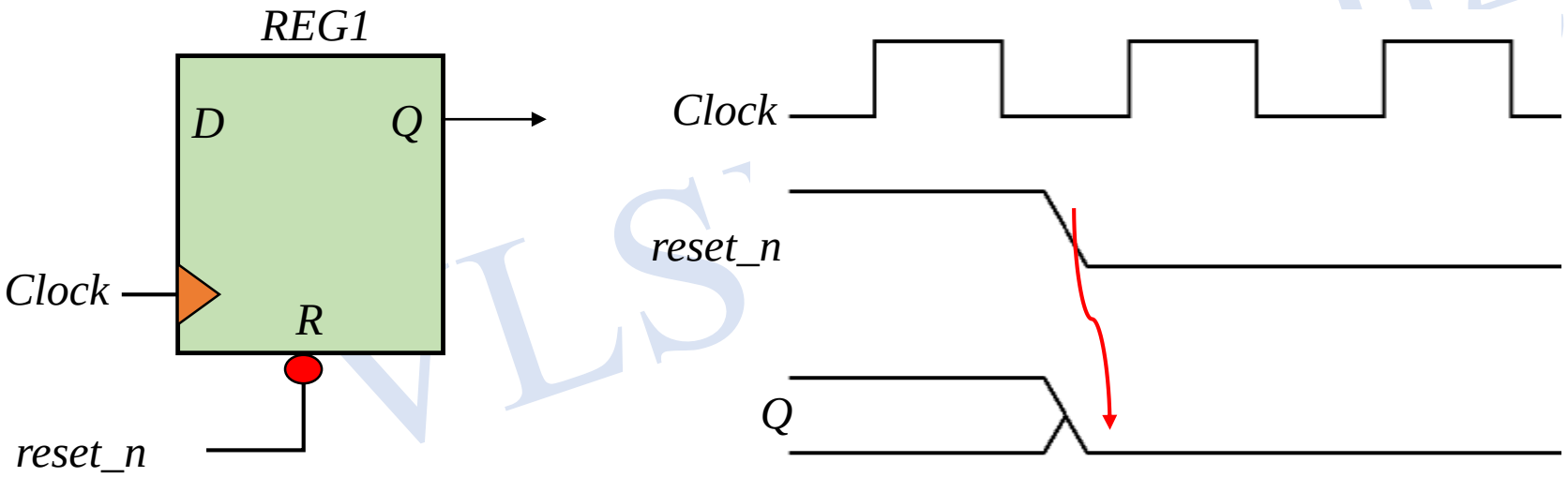




Static Timing Analysis (STA) – Asynchronous Timing Checks

Asynchronous Reset Assertion :

In a flip-flop, assertion of reset causes the output to go to its reset value (which is normally "0"). The assertion of reset is an asynchronous event and is not impacted by the state of clock. As it can be seen below, Output asynchronously goes to "0" as an effect of reset going to its active state “0“ (Active Low Reset).

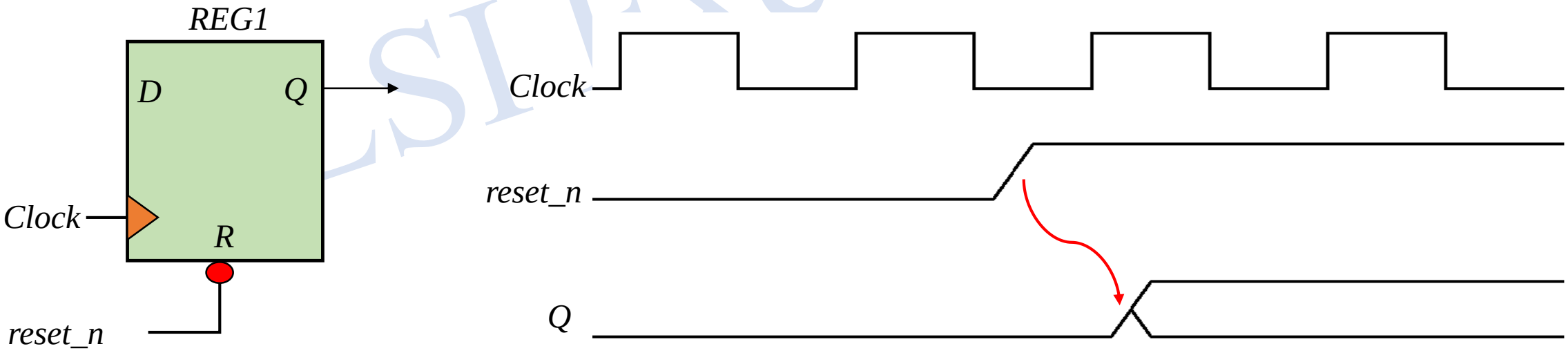



```
always @(posedge clock
                or negedge reset_n)
    if(!reset_n)
        q <= 1'b0;
    else
        q <= d;
```

Static Timing Analysis (STA) – Asynchronous Timing Checks

Asynchronous Reset De-assertion :

The de-assertion of asynchronous reset causes the output to get out of the impact of reset and behave like a normal flip-flop. When the reset gets de-asserted, its output remains to be "0" until the clock edge. When the clock edge arrives, the value at the input of the flop propagates to the output. However, the position of reset de-assertion with respect to clock edge matters and if the reset toggles in the vicinity of clock edge, the flip-flop may go metastable. This is avoided by defining recovery and removal checks for reset de-assertion. For the sake of simplicity, we can say that recovery and removal checks are setup and hold checks for reset de-assertion.



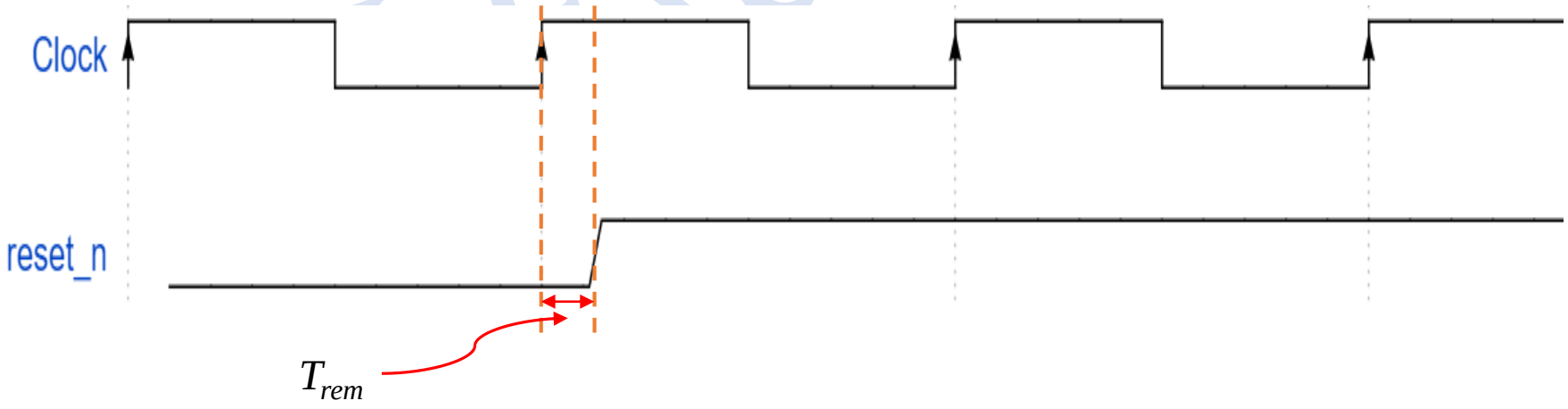
Static Timing Analysis (STA) – Asynchronous Timing Checks

Reset Removal Check :

During the de-assertion of a reset, the control to the output of a flip-flop transfers from the reset line to the clock signal, like a regular D flip-flop. To avoid the register entering metastable state, we must ensure that the reset is not de-asserted in certain time frames of the active clock edge.

The Removal Time, $\mathbf{T_{rem}}$, refers to the minimum time, after the active clock edge, that the reset must be stable before being de-asserted.

The reset Removal Check ensures that the de-asserted reset signal is not captured by the same clock edge that launches the reset.



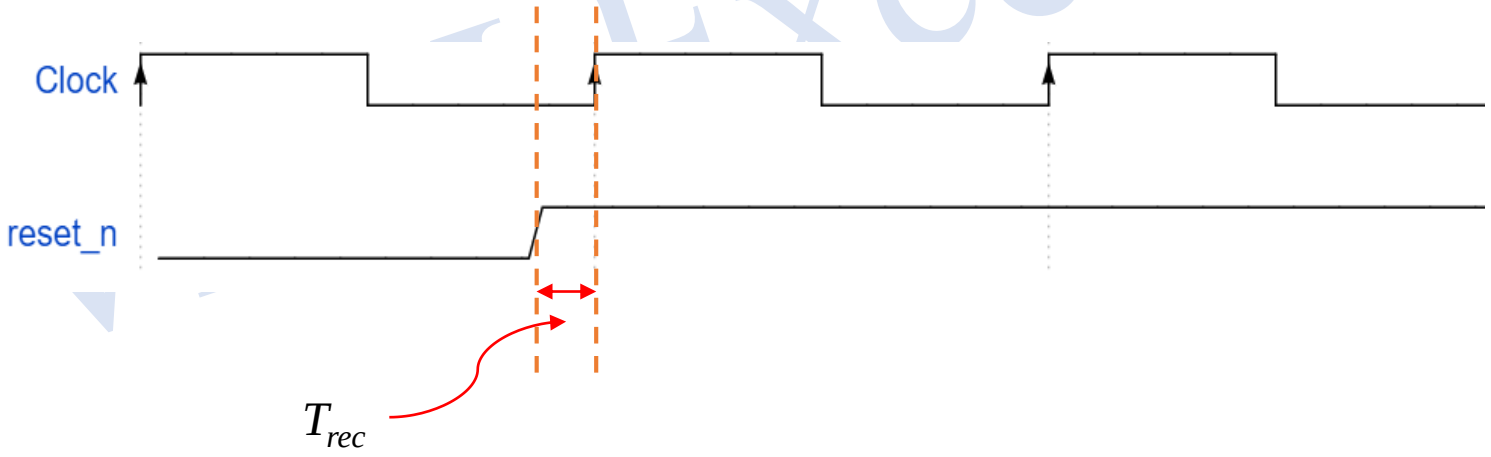
Static Timing Analysis (STA) – Asynchronous Timing Checks

Reset Recovery Check :

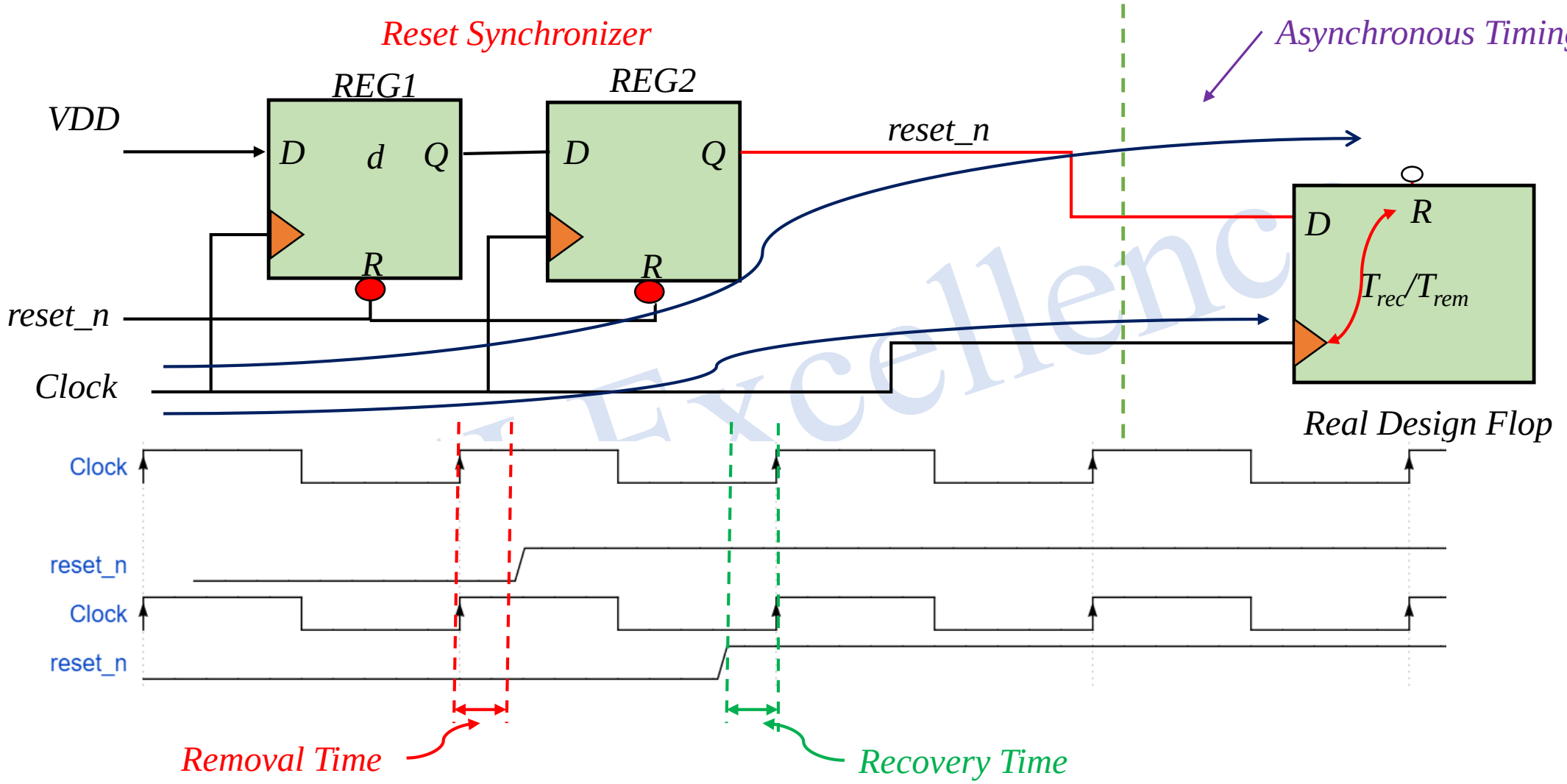
Recovery check ensures that the de-asserted reset signal allows the clock signal to take control of the output at the desired clock edge. For this, reset signal must be stable at least "recovery time" before the active clock edge.

Reset Recovery Time, T_{rec}, is the minimum time between the de-assertion of a reset and the clock signal being high again.

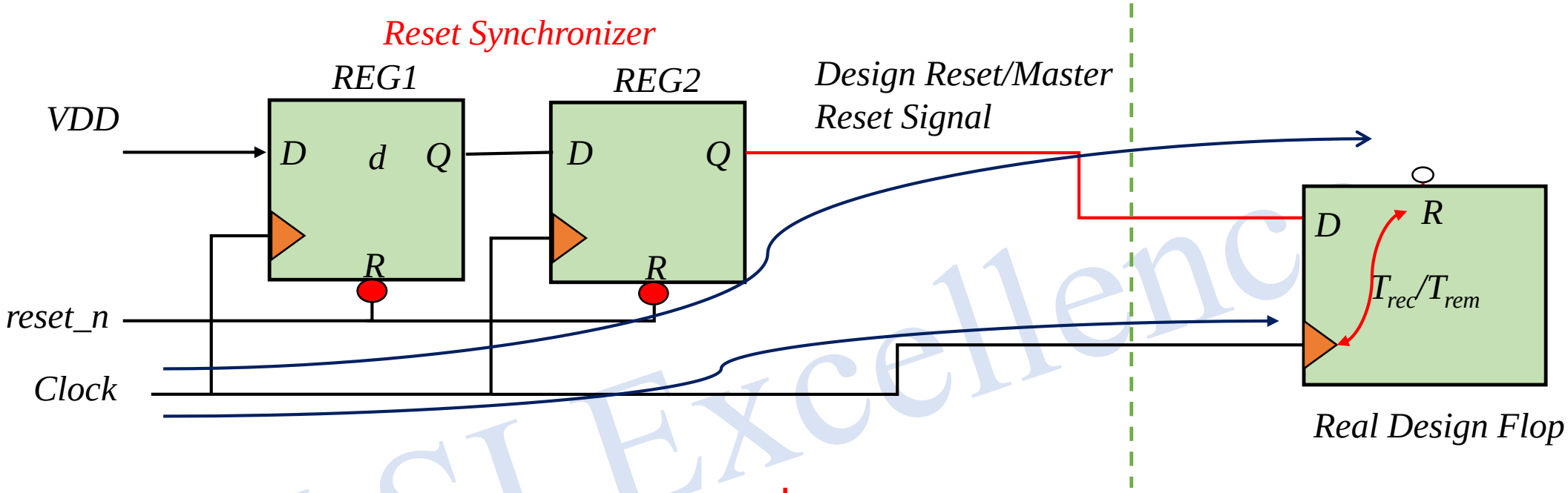
The reset Recovery Check ensures that the reset signal is stable for a minimum time after de-assertion, before the next active clock edge.



Static Timing Analysis (STA) – Asynchronous Timing Checks



Static Timing Analysis (STA) – Asynchronous Timing Checks



Recovery Slack = Required Time – Arrival Time

Arrival Time = $T_{clk\_to\_q}$[REG2]

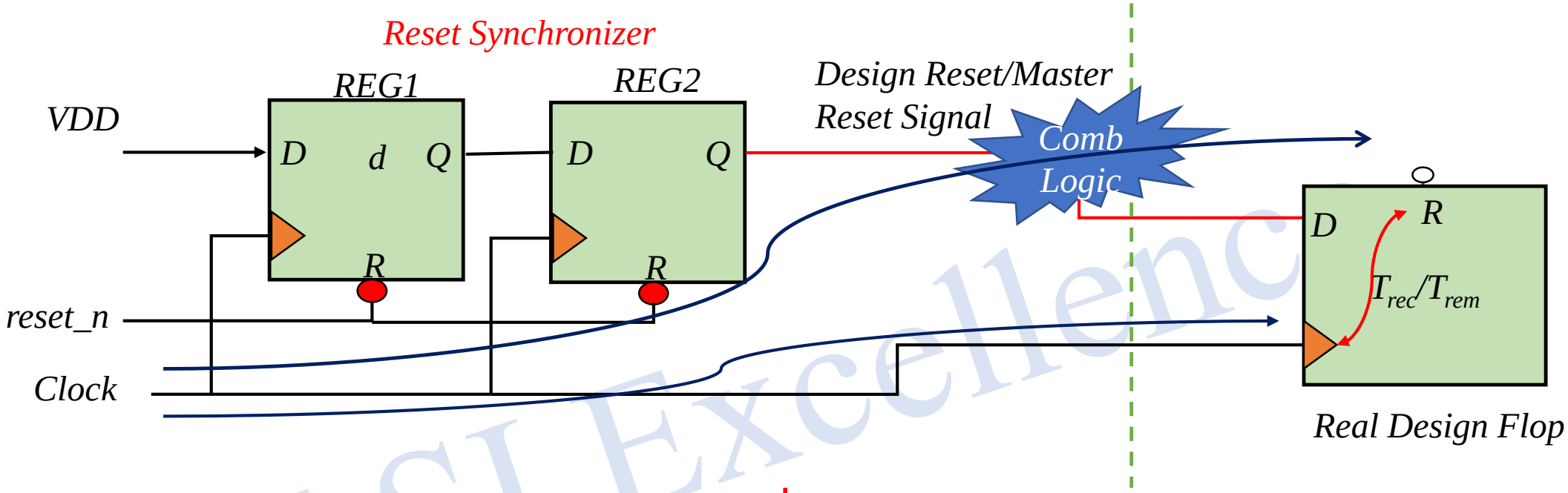
Required Time = T - T_{rec}

Removal Slack = Arrival Time – Required Time

Arrival Time = $T_{clk\_to\_q}$[REG2]

Required Time = T_{rem}

Static Timing Analysis (STA) – Asynchronous Timing Checks



Recovery Slack = Required Time – Arrival Time

Arrival Time = $T_{clk\_to\_q}$[REG2] + T_{comb}

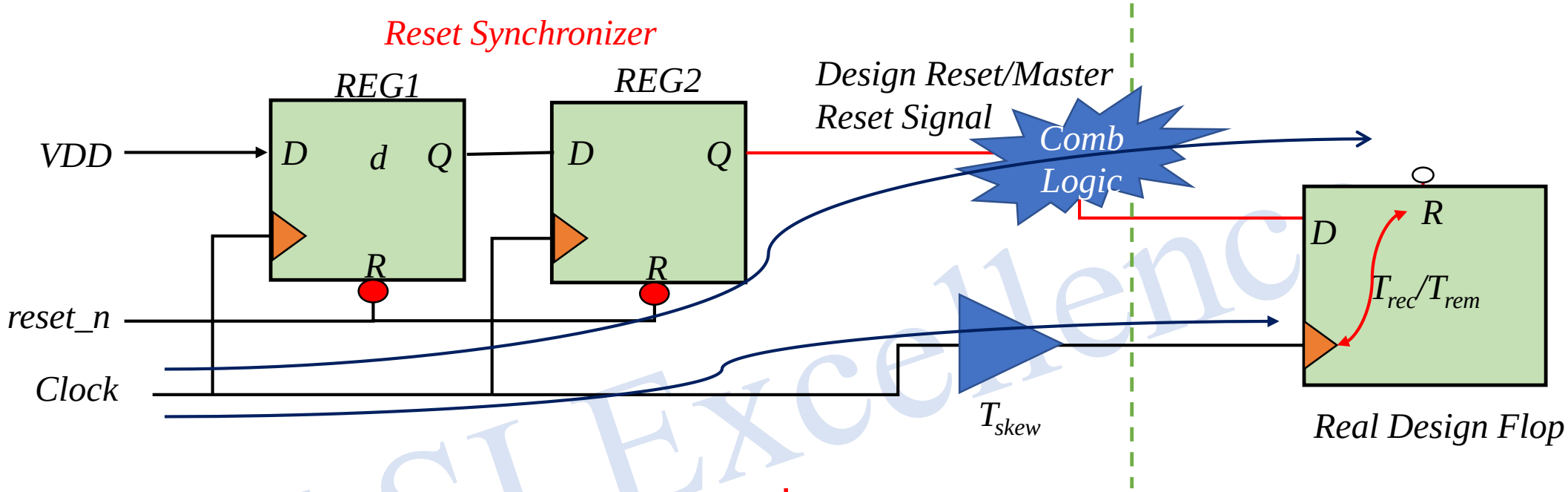
Required Time = T - T_{rec}

Removal Slack = Arrival Time – Required Time

Arrival Time = $T_{clk\_to\_q}$[REG2] + T_{comb}

Required Time = T_{rem}

Static Timing Analysis (STA) – Asynchronous Timing Checks



Recovery Slack = Required Time – Arrival Time

Arrival Time = $T_{clk\_to\_q}$[REG2] + T_{comb}

Required Time = T – T_{rec} + T_{skew}

Removal Slack = Arrival Time – Required Time

Arrival Time = $T_{clk\_to\_q}$[REG2] + T_{comb}

Required Time = T_{rem} + T_{skew}

Static Timing Analysis (STA) – Asynchronous Timing Checks – Imp Points

1) REG1/D is tied to 1 and hence, we do not need to consider any timing requirements for this (Static Signal)
2) Reset de-assertion timing is required for REG1/Q -> REG2/D at the clock frequency at which reset De-assertion is happening
3) Recovery-Removal Checks are required for REG2/Q -> Real Design Flop/R Path
4) Timing at REG1/R pin is not required, since, it is put there to absorb metastability and come out of metastability before next clock edge
5) Timing at REG21/R pin is not required, since, when R gets de-asserted, REG2/D and REG2/Q are both at value "0".
6) Both REG1/R & REG2/R need at least a certain pulse width in order to detect the reset.
This requirement is generally given in the timing model of flip-flop

Static Timing Analysis (STA) – Asynchronous Timing Checks – Imp Points

7) Need to make sure all the flip-flops in design (in same clock domain) gets reset in same clock cycle else defeating purpose of reset synchronization
8) There can be multiple reset synchronizers per clock domain in a single design
9) Reset Synchronizer basically manipulates the originally asynchronous reset to have synchronous de-assertion

Best Free VLSI Content

1. Verilog HDL Crash Course – Link
2. Static Timing Analysis (STA) – Theory Concepts – Link
3. Static Timing Analysis (STA) – Practice/Interview Questions – Link
4. Low Power VLSI Design – Theory Concepts – Link
5. Low Power VLSI Design (LPVLSI) – Practice/Interview Questions – Link
6. Digital ASIC Design Verilog Projects – Link

Please Like, Comment, Share & Subscribe My Channel in Order to Reach Out the Content to a Larger Audience.

Thanks !!