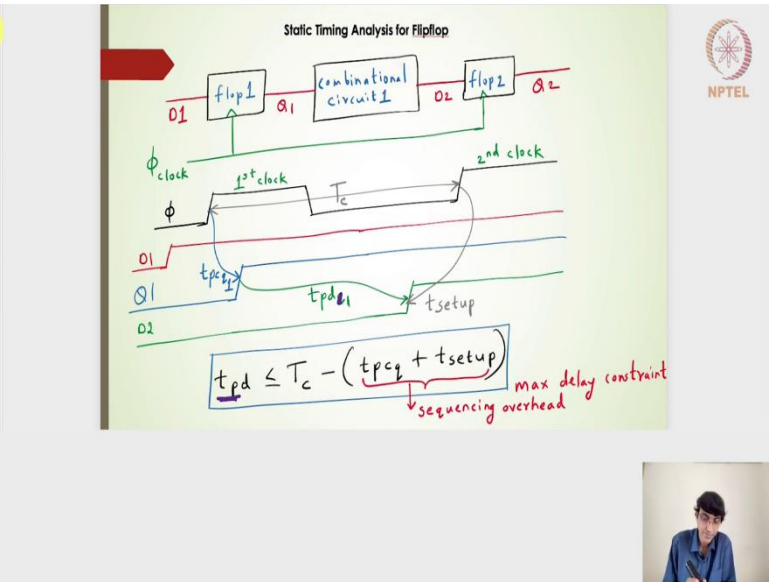
Design and Analysis of VLSI Subsystems
Dr. Madhav Rao
Department of Electronics and Communication Engineering
International Institute of Information Technology, Bangalore

Lecture - 76
Static Timing Analysis - Continued

Hello students, welcome to this lecture on Static Timing Analysis. In this particular lecture we will take a look at some of the examples of the combinatorial circuits within the two, it is there in between the two flops, that it is a timed circuit output. Few some parameters for the 65nm technology node, utilize those set up timing parameters and then the combinatorial circuit propagation delay parameters and then try to design a clock.

That it does not violate the setup time and in this lecture we will also look into the hold time failures and proceed further for the flip flop designs. It is going to be a purely a flip flop designs take use some of this parameters for the flipflop designs and then try to identify what should be the clock time period and if it is does not match properly, then what are the failures we will we should analyze alright.

(Refer Slide Time: 01:18)



Moving ahead, let us say this is one of the designs which we have seen in the last lecture, where we had the flop 1 and flop 2 which is driven by the same clock signal. Let me also pick up my pointer and in between we have the combinatorial circuit.

It will have its own propagation delay and the flop is characterized by the setup time and then the propagation clock to q. Similarly, flop 2 will be characterized by the setup time as well as the propagation clock to q as well as the whole time for both the a flops will be characterized.

Let me take this particular clock signal, and then the 1st clock and then the 2nd clock and then the D1 is arrived much earlier than the D setup time for the flop 1. Let us say that the Q1 the output of the flop 1 is evaluated or it has reached or a steady state of Q1 has been attained after the propagation clock to Q1 of the flop 1 design and post that it is given the input to the combinatorial circuit 1.

It will be the propagation delay one of the combinatorial circuit. The output D which goes to the D2 of the flop 2 input has reached after the propagation delay of the combinatorial circuit 1. Let us say that this has arrived much much earlier than the t_{setup} time of the for this 2nd rising edge of this flop 2.

Then we have this particular expression which is t_{setup},

$$t_{pd} \leq T_c - (t_{pcq1} + t_{setup})$$

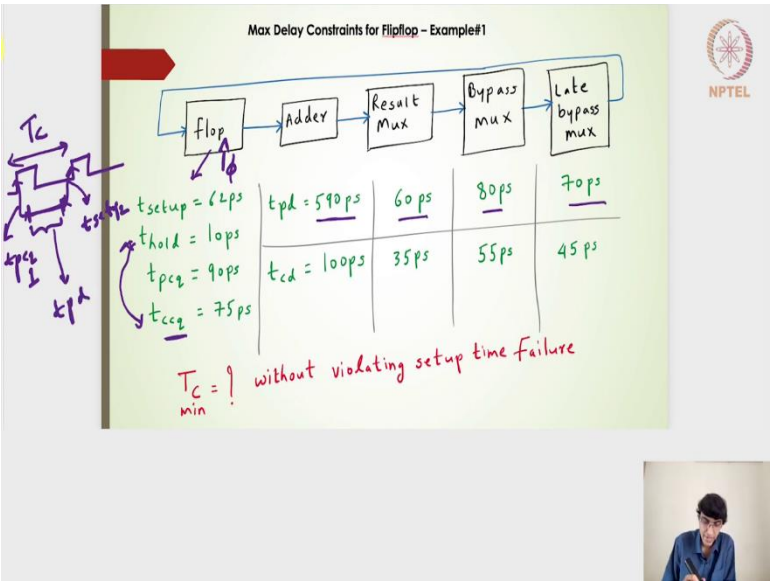
Where this is called as a maximum delay constraints. What it really means is the combinatorial circuit design number 1 this has to be designed in a way that this t_{pd} value or whatever $T_c - (t_{pcq} + t_{setup})$ is the maximum propagation delay, it has to be designed with.

That is the maximum delay constraints for the combinatorial circuit number 1. $t_{pcq} + t_{setup}$ it is a characterized parameters which is taken from the standard cell libraries. It is basically it is nothing but the sequencing overhead parameters. It is basically an overhead parameter of the flop 2 and then overhead parameter flop 1 to t_{pcq} and t_{setup} and it is called as the overhead, because it takes its own time flop 1 will take its own time.

That we will get the output of the flop 1 and then flop 2 for the flop 2, the output of the combinatorial circuit 1 has to reach before the t setup time. This becomes kind of an overhead for the combinatorial circuit design, that is why it is called as a sequencing overhead, but we cannot avoid it because we need the sequencing to be generated, because

we need the output of the combinatorial circuit to be timed. In that sense it is a sequencing overhead, hope it is clear.

(Refer Slide Time: 04:23)



Moving further let us take a typical example for a digital blocks, where we have this series of the combinatorial circuits blocks like the adder, muxes, bypass muxes, late bypass muxes and which is given to the same flops. We do not have flop 1 and flop 2 and the flop 3 as such what we have is the same flops design and the output of this combinatorial circuits is kind of it has to be evaluated at every clock edges or at different clock edges. If I consider at the flop 1 on the 1st clock edges, it will have its own overhead like the t_{pcq}.

The output will be generated after the t_{pcq1} and then its series of combinatorial circuits will give the output here and it should reach to the 2nd clock edge just before the t setup time. It should give the output here, after the t_{pcq} of the 2nd clock edge.

Similarly, it will keep on pumping the output back to the input of the flop designs and then once again the combinatorial computational output will be provided in the 3rd clock edge 4th clock edge and so on. Let us say that the flop is kind of characterized with this parameter set up time of 62ps hold of 10ps, t_{pcq} propagation clock to q of 90ps and contamination clock to q of 75ps.

We will come back to this t hold and contamination delays little bit later, but we will try using this setup and t_{pcq} because, we have anyway seen the maximum delay constraints.

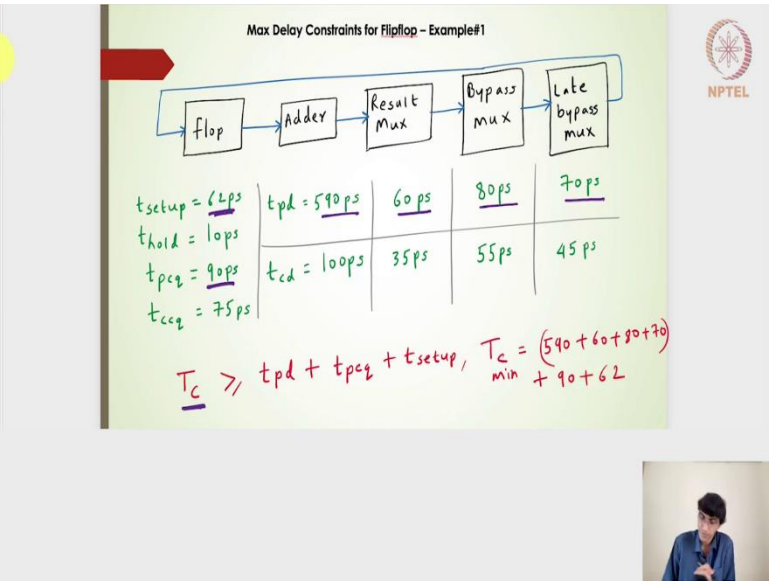
The propagation delay of this adder block, result muxes, bypass muxes, and a late bypass muxes are given in here 590, 60, 80 and 70ps respect to. Its contamination delay parameter is also given, but I think we will have a look at it in a later stage.

If I just go by the design requirement or the specifications, we can easily see here that the t_{cd} value is actually less than the t_{pd} value. The 100ps is less than 590, 35ps is less than 60, 55ps is less than 80ps and 45ps is less than 70ps and intuitively that makes sense, because a contamination delay is always less than that of the propagation delay.

The question here is what should be the clock that we need to design and what should be its minimum duration, that we do not violate the setup time failure. If I remember the clock we have the clock and if I consider the positive edge clock, we have to ensure that within the sequencing overhead the combinational circuit the propagation delay of the combinatorial circuit blocks here, should be reached before the t_{setup} time.

This t_{pcq} is there will be there for the flop to throw the output. This will determine my clock duration. If I look into the clock duration here it will be nothing but the minimum clock duration, that we do not get into the setup time failure, will be nothing that the addition of all these 3 components. The 3 components are this will be t_{pcq} of the 1st flip flop, this will be the t setup of the 2nd flip flop here both the flip flops have are the same and then this will be the combinatorial circuit propagation delay value.

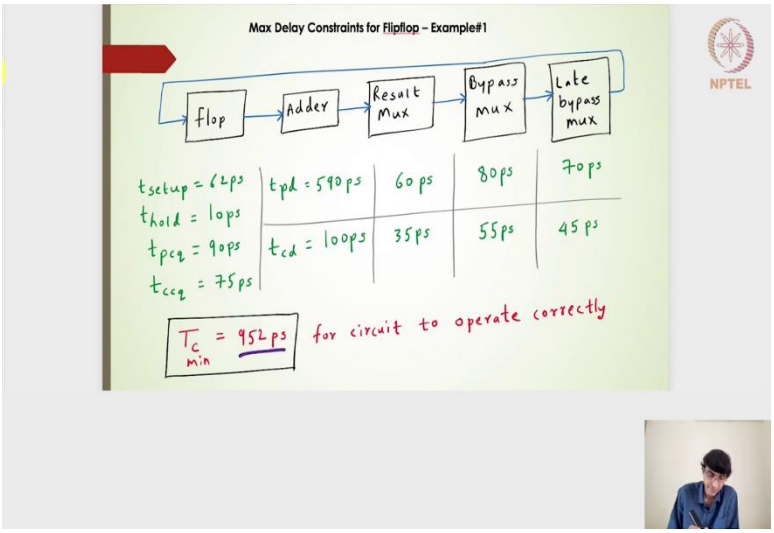
(Refer Slide Time: 08:08)



$$T_c \geq t_{pd} + t_{pcq} + t_{setup}$$

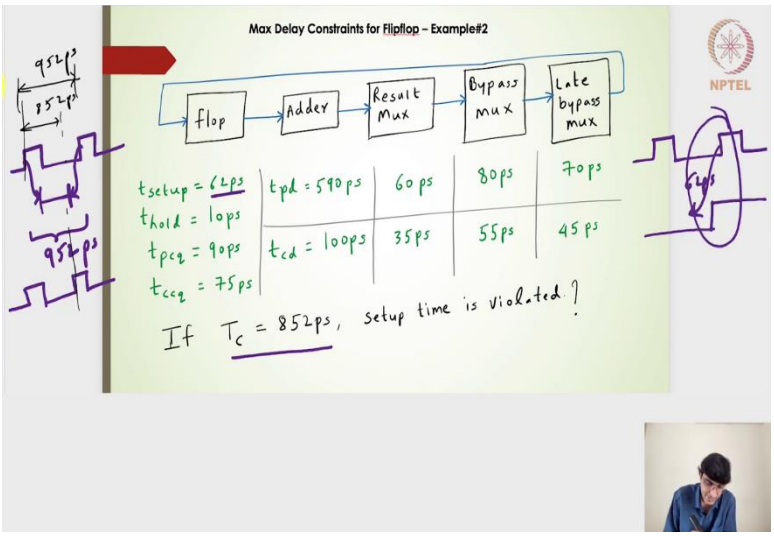
$$T_{c_{min}} = (590 + 60 + 80 + 70) + 90 + 62 = 952ps$$

(Refer Slide Time: 08:41)



This is what we will get 952ps should be the minimum clock time period, that we do not get into the set of time violation. For the circuit to operate correctly, it should be greater than that. If I choose 100ps that is very well good, that will be better, but it should not go below 952ps.

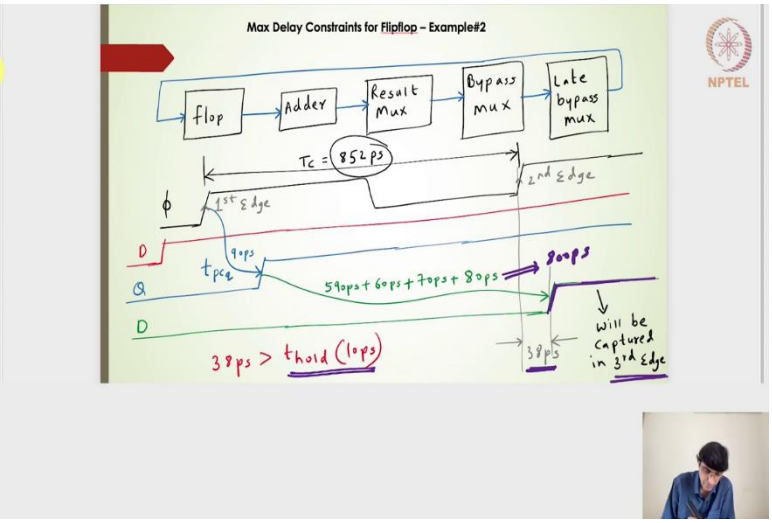
(Refer Slide Time: 09:04)



Moving ahead, let us say that that clock time period is 852ps, then what is the problem. Remember that if it is 852ps and as we had seen earlier let me draw that again. If I have a clock time period, this t_{pcq} and propagation delay of all this 3 combinatorial boxes, let me try to erase this and this is this t_{setup} time.

All this three parameters requires 952ps that is what we have decided in the previous slide, but now what if the clock itself is shot? what if the clock itself is a shot here, 852ps and what happens? if I take this clock and then this clock instead of 952, it is 852ps. We were supposed to have 952ps, but whatever we have 852ps. My output, in fact, I need to determine where this combinatorial circuit output will reach and whether it will violate the set up time failure.

(Refer Slide Time: 10:31)

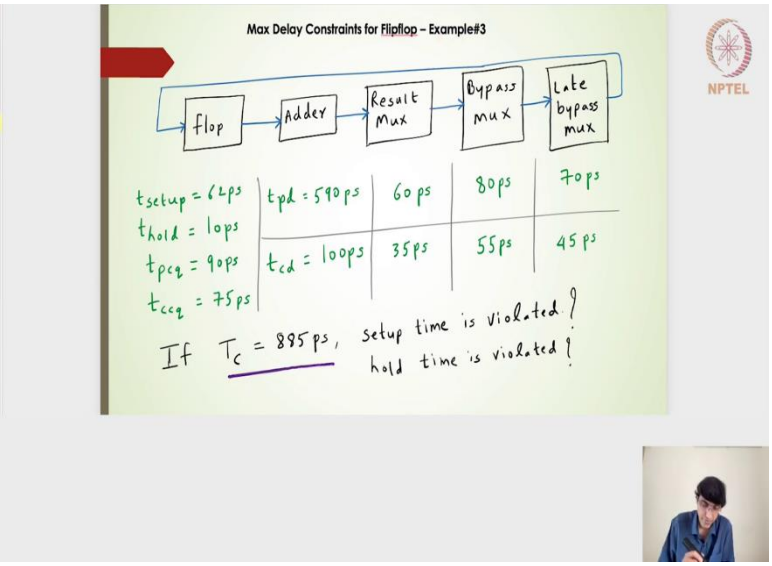


Moving ahead, this is what my timing diagram will look now, I can now for 852ps. For an 852ps the 1st edge and the 2nd edge I have written and then from the 1st edge I will get t_{pcq} of the flop for this one and then it proceeds to the combinatorial circuits 4 blocks and then the output, which will be fed as an input to the flop. That is what I have written here D, it takes 590 + 60 + 70 + 80, it is 90 here t_{pcq} from the flop. Then the combinatorial circuits addition of the propagation delay which will be 590 + 60, which will be 650 + 70 + 80 which will be 800ps. This the total will be 800ps. The 800 + 90, it will be 895 and 852 is what the clock duration is. It is likely to pass the output of late bypass mux is going to surpass the 2nd clock edge.

In fact, it is going to surpass the 2nd clock edge by $890 - 852 = 38$ps. What is likely to happen is 38ps does not fall in the t hold the time duration. If I consider the t hold time duration which is the t hold which is 10ps after the rising clock edge, it does not it surpasses that 10ps also, it is 38ps. This particular output of the combinatorial circuit, which is given to the flop which will be captured now, not in the second edge, but in the third clock edge, it will not have any setup time failure effects. Setup time failure effect happens only if my output actually reaches somewhere in between the setup time duration and then the clock edge.

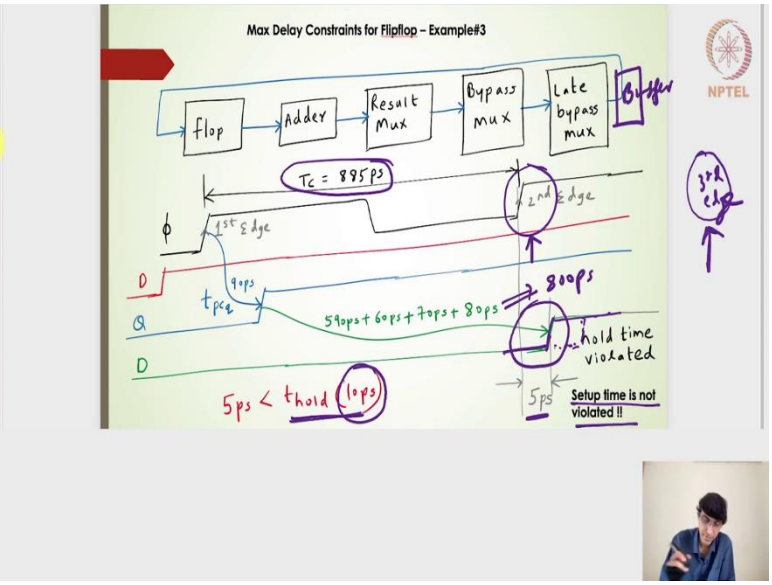
If the setup time duration in the previous case, let us say that the setup time is 62ps. If it reaches somewhere between from the clock edge to the 62ps, if I have a clock edge here and this is my 62ps. But, my output actually reaches somewhere in between this, then we have this setup time failure but in this case it is actually not doing that and in fact, it is reaching much much earlier with respect to the 3rd clock edge and it is reaching much much later than the 2nd clock edge. In that sense it is not having the setup time failure alright.

(Refer Slide Time: 13:12)



The other question is if let us say if the T_c = 885ps, whether it is going to have the set up time failure and then the hold time violation. Again the 885ps I need to draw the clock diagram and then find out my output of the late bypass mux, whether its reaching you know near the second clock edge or what.

(Refer Slide Time: 13:38)



If I do that, this is my clock of 885ps of the duration. The 2nd clock edge is going to arrive after 885ps of the 1st clock edge and if I consider this 90ps of t_{pcq} and then the remaining this will be our 800ps.

The 885 and 800 + 90 = 890, it is going to arrive after 5ps after the clock edge the 2nd clock edge. It is not arriving 5ps before the 2nd clock edge, then it will be a setup time violation. Now, it is arriving after, it is not the setup time violation anymore now, but it is having a hold time violation, the reason is the hold time is of 10ps.

It is not arriving after 10ps, but rather it is arriving before the 10ps. It is arriving after the 5ps to the clock edge. It falls in between that hold time and then the data can actually get sneaked in to the output. The setup is not violated, but the hold time is violated, because what we are assuming is the because the setup is not violated.

This particular clock edge will now be at captured at the 3rd clock edge, but rather because the contamination delay or whatever the output it can pass here, this gets sneaked in within this the 5ps or the 10ps window of the t_{hold} time.

Some of the output can get captured into the second clock edge, but whereas the setup time is not violated, what it means is the output should be captured in the 3rd clock edge. But here because of the hold time failure some of the outputs can get captured in the second clock edge which is not correct. It will be very very inconsistent in the sense some of the

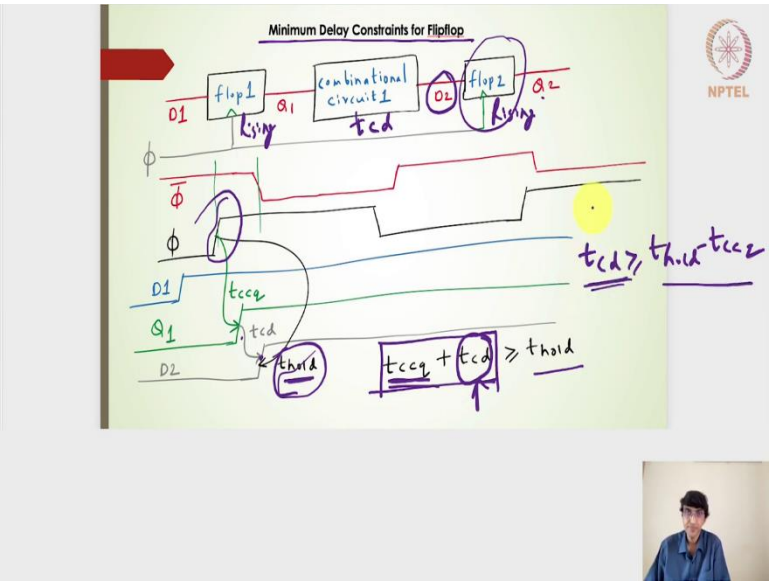
output will be captured in the 3rd clock edge and some of them will be captured in the 2nd clock edge, that is not right appropriate design. We need to ensure that this particular output is changing only after the 2nd clock edge and its hold time.

We need to have to add a buffer here, which will give some more delay and that delay is going to have and it is going to avoid the hold time failure. In that sense I will not have a hold time failure I will not have a set up time failure and then we can consistency or confirmly or confidently say that the output will be captured in the 3rd clock edge.

The output which is captured in the 1st clock edge will be captured in the 3rd clock edge and then it will be captured in the 5th clock edge and then so on. What we really need to do is we need to design a circuit, we will have to make the confirmations that both the hold time failure as well as the setup time failure are avoided.

If one of them is violating and if other is not violating then also we will have a problem, if both of them are violating then also we will have a problem. What we need to ensure that it is both of them both the static timing analysis or the constraints like the setup time violation as well as the hold time violation does not occur in our design, hope this is clear.

(Refer Slide Time: 16:58)



I will just introduce you to the minimum delay constraints for the flip flops right. What we have see is a maximum delay constraints, now for the combinatorial circuit what is a

maximum delay constraints that we know, that is actually a function of the clock time period and then its parameters propagation clock to q and then the set up time.

The minimum delay constraints is about what should be the minimum delay, a what should be the minimum delay for the combinatorial circuit. So, that it avoids the hold time failure. If I look into this particular design, it is nothing but the clock the same clock is given to the flop 1 and flop 2, D1 signal is given to the flop 1. So, that I will get the Q1 output which is given to the combinatorial circuit 1 and then the output of the combinatorial circuit 1 is fed to the flop 2 design. If I draw the clock here and then remember that this flops are for the rising edges. These are for the rising edges of the clock rising edges of the clock.

My clock is like this. My $\overline{Q}$ or the $\overline{clock}$ signal, because there will be some delay for the $\overline{clock}$ to be generated from the clock. That is when the clock hold time failure is likely to occur or the hold time itself is actually defined by the difference between the $\overline{clock}$ and the clock signal or whatever the propagation delay for the $\overline{clock}$ from the clock source. That propagation delay is nothing but the hold time.

Let us say that the D1 has arrived earlier, that it is quite stable and it has arrived before the setup time of the flop 1 design and then Q1. The Q1 coming from the flop 1 and it is coming really fast, it could be due to the contamination output contamination clock to q output.

Here I am not taken propagation clock to q, it is the contamination clock to q. If it is a contaminated output here, which occurs and this contaminated output goes to the propagates to the combinatorial circuit, through the contaminated output here and let us say that the combinatorial circuit has 10 inputs, but not all 10 inputs are changing only one input is changing in that sense I will get a fastest output here.

That fastest output we can say it is a contaminated output, because this is a contaminated input coming to the combinatorial circuit. Then we will have a contaminated output or it can be specified or characterized by the contamination delay of the combinatorial circuits. We will have contaminated output here, which if at all it reaches before the t_{hold} of the flop 2, if it reaches below before the t hold of the flop 2 then we will have a problem. Let

me try to explain this again, flop 1 we are expecting the input to be captured the output should be capturing the input in the 1st clock edge.

Flop 2 what we are ideally expecting is D2 will arrive after the hold time of the clock 1. That it will be able to capture consistently in the clock 2 signal. But, if the D2 signal arrives whatever it is in the contaminated outputs or the propagated output propagation delay output or the contaminated delay output.

If it reaches before the hold time of the 1st clock edge, then the flop 2 is likely to capture it might actually gets sneaked in into the Q2 signal. The Q2 will be appearing in the clock in the 1st clock itself rather than what we are expecting is that it will come into the second clock edge.

But, because of the contaminated output coming from the flop 1 the contamination clock to q or delay output and then the contamination delay of this combinatorial circuit. There is a likelihood, if both of them if it appears if it is really small and if it is smaller than the t hold time of this particular flop 2's which we have used, then my Q2 can get actually sneaked in the 1st clock edge itself which is not good.

What we are expecting is from the setup time, we are expecting that it will arrive in the 2nd clock edge, but the contaminated output can actually sneak in into the first clock edge itself, which is kind of violating, it is violating the hold time failure. What we really need is the combinatorial circuits contamination delay and the flop 1's t_{ccq} both this one and this one addition of this should be greater than the t hold time which we have.

$$t_{ccq} + t_{cd} \geq t_{hold}$$

If that is the case $t_{cd} < t_{hold}$. Then we are rest assured that Q2 although is D2, we get a D2 signal which is very very fast, because of the contamination signal coming from flop 1 as well as the t_{cd} of the combinatorial circuit, but still it passes it comes after a t_{hold} time.

It cannot sneak into the output, sneaking in is possible only in the 2nd clock edge. It will pass only in the 2nd clock edge. My t hold whatever it is it is kind of a characterized from the flop designs standard cell libraries. We have to ensure that $t_{ccq} + t_{cd} \geq t_{hold}$. t_{ccq} is from the flop designs is again a characterized parameter.

We do not have much of an allowance to design, what we can have a different design is the t_{cd}, t_{cd} is a design parameter. We have to design the combinatorial circuit such that the $t_{cd} \geq t_{hold} - t_{ccq}$. We will have to also have to come up with a combinatorial circuit design such that t_{pd}, the t_{pd} progression delay should not violate the setup time failure.

The combinatorial circuit designs now have 2 constraints one is coming from the setup time failures, avoiding the setup time failures and then another constraint is coming from the hold time failures. The setup time failures will give us the maximum delay constraints and then the hold time failures here gives us the minimum delay constraints. If I write t_{cd}= $t_{hold} - t_{ccq}$.

The minimum delay that it should have the combination circuit should have. So, as to avoid the hold time failure is $t_{hold} - t_{ccq}$, that should be the minimum delay. I hope a you are able to understand this.