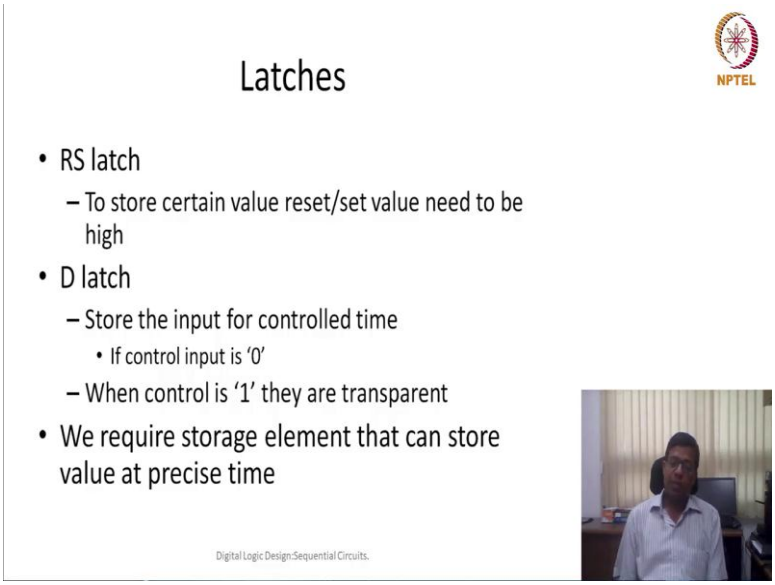
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology Ropar
Lecture 42
D-Flip-Flops

(Refer Slide Time: 0:21)



Hello everybody, today we would be talking about edge triggered D flip flops. In our previous lecture we have studied what is a basic memory element and we have seen that if we put a feedback loop in around any circuit, then that particular circuit can work like a memory element.

So, you started with the inverter and then we have seen that if we want to set or set the storage value to 0 or 1, then we can create a set reset or RS latch. So, in this RS latch, the duty was that if we have to set the stored value to 0, we have to make set as 1 and reset to 0. Similarly, if we want to store a value of 0, then we have to make reset as 1 and set as 0.

So, this particular combination gives us a good powerful mechanism, so that we can store either 0 or 1 inside a memory element. And this memory element is created from nothing else but our basic gates like AND, OR and NOT. Now, this set and reset although it is a good powerful feature, but if we want to set something, let us say I, I want to set something which is variable, so, I have I want to either set 0 or set 1, so this combination of set and reset what should be the value.

So, there has to be some additional circuitry or additional number of gates that need to be employed so that only either set or reset is 1. To avoid this kind of additional circuitry, so one

of the flip flop, one of the latch which we have studied in our previous lectures performs excellently. So, for this particular work, this latch is called D latch or delay latch D sometimes for delay also because whatever is the input that is given as the output here.

Now, in delay latch, whatever we would like to give, so let us say there is a signal x and this x can be given as an input to my D latch, then the output would be same as whatever is written in the D if D is 0, then output would be 0, if 1, if D is 1 input is 1, then output would also be 1, but we need the storage element.

So, the storage element can be created if we put a control or basically a control input. So, if this control input is 0, then whatever was stored in latch that will remain stored. If control input is 1 then it works like a transparent logic. So, whatever is the input that would be the output. So, essentially if I want to store anything inside a latch, particularly D latch, then I will make this control input as 0 and then whatever was stored, it will remain stored and whenever I want to store anything, I will make this control input as 1.

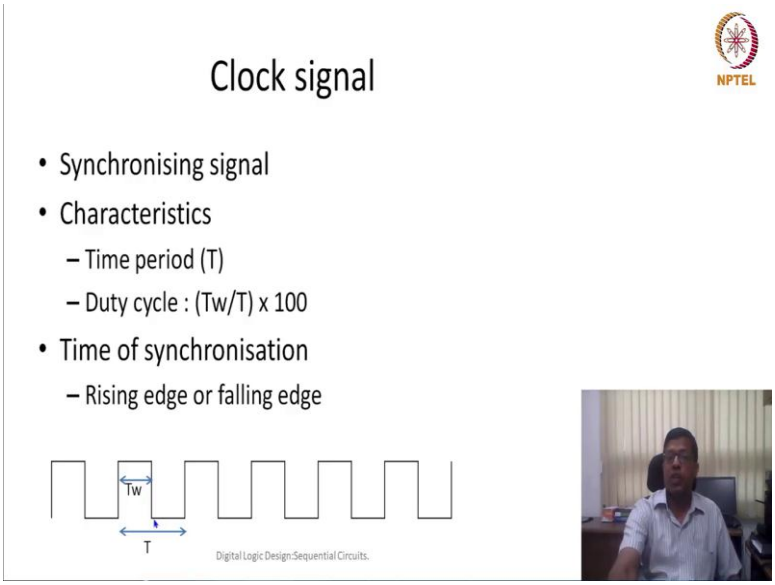
So, this gives us a very good mechanism that if I have any variable or any particular signal and I want to store whatever is the value inside that signal to I want to store in a memory element that I can use D latch. The only problem with the D latch is that we have to make sure that whatever we want to store or whenever this control input is 1, then the value whatever you want to store is there in at the input.

Because if D if the dual input is 1 whatever is at the input D it would be transparently reflected at output Q. Because it would be transparently reflected at output Q, so that means if this D is changing, my input is changing it is fluctuating my output will also fluctuate. Other thing I have to make sure that whatever I want to store is correct at this particular time. So, this is also that is why this particular latches are also called level triggered. So, whatever is the level if the level is high and it would trigger at this high level then whatever is the input that will get, will be stored.

Now, because this level or when my control input is 1 that gives significant amount of time to change my, to change in my input. That is why these latches may not be a very good, very good component to do synchronization or to store the value for definite amount of time. Yeah, value can be stored for definite amount of time because this 0 whenever control input is 0 that we can control. But the value when we cannot make sure that my D would be stable, my input would be stable during, during when it is 1 when control is 1. So, if I want to have

storage element, which can store the value at precise time, that is why we would require something which is additional.

(Refer Slide Time: 6:13)



So, what is this precise time? For this, I will take a small detour. And I will try to explain one of the very important synchronization signal that is called clock. In most of the digital circuits, this clock signal or a clock wire is a very one of the very important signals because it synchronizes, what do we mean by synchronizes.

So, particularly in synchronous sequential circuits, whenever the all the inputs would change at same time, all the outputs will also be reflected at the same time, so they would be received at same time. So, that means when every all the components, so you may have various combinational logic, adder, multiplier, multiplexers, all of them they would be synchronized with one signal that is called clock signal.

So, this synchronous signal, synchronizing signal clock is essentially the basis of synchronous sequential circuits. Now, why it is important, because many a times we would like to divide our work into smaller steps, each of this step if we make sure that each of these steps can be done in some definitive time, then things get easier for analysis, things get easier for designing.

So, that is why this clock signal is used in almost all kind of digital circuits. You can take an example also for example, when you go to market and try to buy a computer so, you will see what is the clock frequency of this processor, try to remember that what is the clock

frequency of your processor which is there in your laptop, so it must be somewhere around 2.4 gigahertz 2.8 gigahertz or maybe 3.2 gigahertz. What does that mean?

It means that your processor is driven by a clock which is running at the frequency of 3.2 gigahertz that means, in 1 second there will be 3.2 into 10 is to power 9 clock periods. So, what does this signify? This signify that the smallest step in or smallest work that can be done in your computer would be done in this 1 clock period.

So, essentially a clock signal is characterized by these two features one is called time period, another is duty cycle. So, time period is after which time, after how much time the clock signal will repeat itself. 1 by T is frequency and usually the frequency would be it could be in kilo hertz, megahertz, gigahertz.

So, in last around last 20, 30 years, 40 years this this clock frequency used to be one metrics of performance. For example, some 40 years back, 50 years, 40 45 years back when first processor came it used to run on 1 megahertz. Now, slowly these processors have started working on a couple of gigahertz.

So, this fast paced transition this is because now we are getting faster and that is why these clock frequencies are shrinking, clock period is shrinking, clock frequencies are increasing day by day. So, a clock period is one characteristic of your clock signal and now, clock signal would be, essentially it would be high for some time and then low for some time and then again high for some time and low for some time.

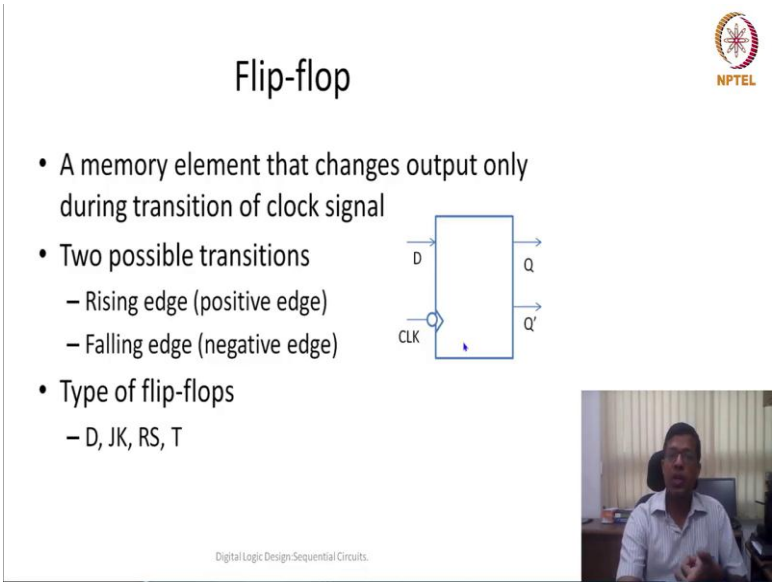
And one thing which would be there in most of the in the clock period of at least in computers that it would be same, so, clock period is going to be same and it always it would be high for same amount of time it would be low for same amount of time, because it is high for same amount of time. That is why we can also have another signifying characteristic of a clock signal, that is called duty cycle.

So, we can call it a pulse. So, what is whatever is the size of pulse width, the amount of time for this pulse or the clock signal is high divided by total time period we can say that is the duty cycle. So, ideally duty cycle could be 50 percent that means 50 percent it is high 50 percent it is low. But in different kind of circuits, it could be anything, it could be our duty cycle could be 10 percent, it could be 80 percent, could be 90 percent, 50 percent gives some ideal characteristics and it also be of some advantage, which we may talk sometime later during the course.

So, these are the two important characteristics. The other characteristic is at what time should we synchronize. So, there are two interesting times here one, we say a rising edge the other we say falling edge, rising edge means when clock will rise from 0 to 1. That is called the rising edge because clock was initially 0 then it is 1 and this edge is a rising edge.

You see that in the right direction we are increasing in the time, so initially, at some time, it was 0 then it has increased to 1. So, this is called rising edge which is also called positive edge. And then the other is negative edge when your clock when our clock will go from 1 to 0. So, with time from 1 to 0, this is the other transition. So essentially, there are two transitions. Each of the transition can act as a synchronization point, rising edge and falling edge. Now, let us get back to this kind of storage element which work on some precise synchronization points.

(Refer Slide Time: 13:08)



These storage elements memory elements, which can store the input at precise synchronization point or precise transition points are called flip flop. They are also called, so they will only change the output during the transition of clock period, they will not otherwise change their output. Now, output will change and we have seen that there are two transitions one is positive edge or rising edge.

So, let us say now, let us take an example of a flip flop which is we are saying it is like a D latch, but it is a D flip flop. Now, the difference between the symbol of a D flip flop and D latch is additionally you see this arrow. This additional arrow represents that this particular circuit is sensitive to transition.

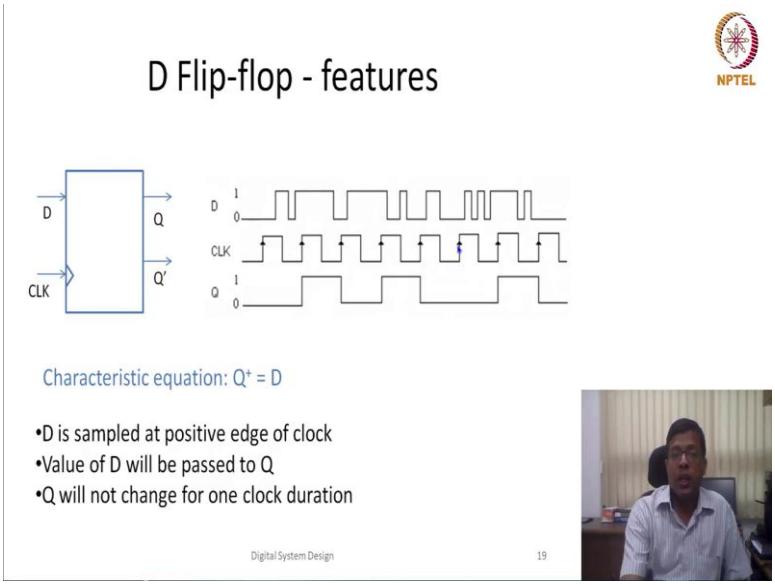
So, that means whenever this clock will transition from 0 to 1 that means there will be a rising edge or there is a positive edge then this output will change otherwise output will not change. So, my D my input can be anything in between, but when there would be a rising edge of the clock then only there is a possibility that Q will change otherwise Q will not change.

Similarly, there could be other transition which could be negative edge or falling edge. So, for negative edge or falling edge we put a circle on the clock signal. So, this arrow will still be there, this arrow signifies the edge triggered flip flop. This circle means negative edge. So, this if we have a circle here and we have an arrow like this, that means it is a negative edge triggered flip flop.

So, what is the meaning of this whole sentence negative edge triggered, so that means this flip flop will trigger only when the clock will have a negative edge. In other words, the output will change only if clock has a negative edge. So, at the negative edge, it will probably see the inputs and based on that the output will change.

There could be various type of flip flops, this D flip flop is one of the example there would we JK flip flop, RS flip flop. So, those flip flop we will study with in forthcoming lectures. So, today will focus on the basic property of a flip flop. And so, we take the simplest one D flip flop or a delay flip flop.

(Refer Slide Time: 16:13)



Now, a delay flip flop is represented by a characteristic equation that the next state is equal to D. Now, you see there is no control signal. So, at what time Q plus or next state would be

equal to D when there is a positive edge. So, you see an arrow here that means the clock transition, and because there is no circle here, so that essentially means the, whenever clock will be rising edge, whenever it would be positive edge of the clock, then my Q plus would be equal to D, otherwise it will remain whatever is the previous value of Q.

So, only during this positive transition, the input can change, output can change and the output would be equal to D. So, that is what we call characteristic equation of this particular flip flop. So, two important things, one, D would be passed to Q plus or the output only at the positive edge of the clock, the second thing, my output will not change during the rest of the clock period.

So, to understand this, let us take example. So, let us say this is my clock signal and sometime to represent that my clock signal, my this flip flop is positive edge triggered, so have I put an additional arrow. So, this additional arrow signifies that at this moment of time, at this moment of time, whatever is the D that will become Q.

So, because D was 0, Q has become 0 at the during this period, from this clock time to next rising edge my D had changed multiple times, but it does not matter, it does not affect my output at all. Now, when the next positive edge has come at that time, again, we sample D or we will see what is the input at D.

Now, because D was high at that time, so, my output would remain 1 till next positive edge and the next positive edge again we have sampled what is the input of D what is the value of D this was this was 0. So, it remained 0 for during whole 1 clock period. So, similarly for all other places, but here this also you see that, now, during this positive edge, my input was 0.

Within sorry, so here the because D was 0 the output became 0. Before the next positive edge came the output has again become, input has become 1 but it came 0 at the time of sample it was it was again 0. So, output did not change. Output will not change according to the transitions, but a transitions of the input only output will change whenever there is transition in the clock period. You can say or you can question what is the utility of such a circuit which is so constrained?

This constraint that my output can change only at a fixed time. Is putting some constraint on the input, but it is also making sure that my output will also not change for one clock period. What does it mean? It means that these outputs can also be given as an input to some other

signal, to some other circuit. So, let us take an example of an addition we implement 32-bit adder and we implemented using ripple carry adder.

So, and let us say the input which is coming to me ripple carry adder is coming from a flip flop. So, which will make sure that the input to my adder will not change for during that whole 1 clock period. Now, what should be the size of clock period, so this gives me an advantage because my input will not change for 1 clock period, then this will give me an advantage a clear advantage that my circuit will not go wrong because my inputs are stable.

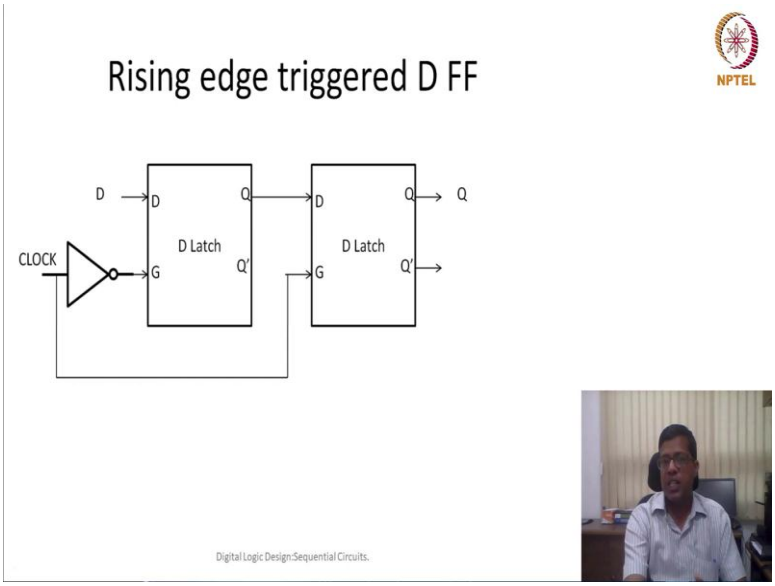
Because they are coming from a flip flop, there is no chance that my input will change before output has come. Now, let us say because of ripple carry adder and it is a 16-bit adder let us say the delay of 1 this whole 16-bit adder is 35 nanoseconds or 35 units. Now, what if I make sure that the clock period is more than 35 nanoseconds or 35 units that means, my input will not change for 35 units.

Now, let us consider further that the output of this adder is further used as a input to another D flip flop, which means that the output of this adder can keep on changing during this whole 35 units of time. And it keeps on changing we have seen. So, when we do, when we did this adder assignment, we did this Verilog code, we could see that the output keep on fluctuating only at the end of 35 units of time we make sure, it is sure that output will not further change.

So, that means this fluctuation can keep on happening, but whatever is the output of the adder at 35 units of time, we can sample it at correct time. So, this gives us some additional constraint to our circuit also that if my circuit has certain delay, I can set my clock period accordingly or in other words, if my clock period is fixed, then I can make my circuit accordingly that my circuit should stabilize its output before the change of clock period.

So, this gives us an additional a powerful mechanism to design very, very complex circuits. You can see your microprocessors can run very complex software's accurately, there is very less chance, there is almost negligible chance that their system will go wrong, it is precisely because it is defined that within 1 clock period what work can be done.

(Refer Slide Time: 23:57)



So, now how can we design how we can implement using our basic gates this kind of a D flip flop design can be done using 2 D latches. So, if we have 2 D latches one is one is gated with a with a clock signal which is inverted, the other which is which is gated with a clock signal, then we can design this rising edge triggered flip flop. So, what would happen because the clock is inverted.

Now, whenever clock is negative or clock is 0, when the clock signal is 0 at that time, whatever was a D value would be given as the output of first D latch. Now, the output has gone to first D latch when clock was 0. Then there is a clock transition at that time of clock transition. Here after that clock transition, this has become 1, this is the positive edge, from 0 it has become 1 and at the time of 1 whatever was taught in this output of first latch can be given as the output. So, the overall mechanism is such that my D value this this input value can go to final output only during the rising edge, otherwise, whatever is this D value will never be transmitted here will never be stored here or given as the output.

So, you can consider all other cases for example, whenever this is 1, this particular clock is 1 then this delay latch will not be active. So, this Q will not change similarly, when this is 1 this is 0 and then this particular delay latch would we ever would not be active. So, this Q will not go here.

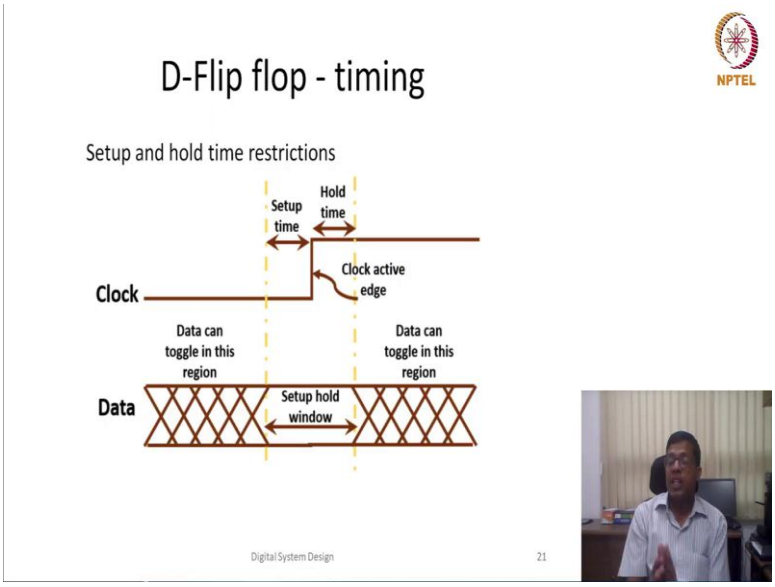
So, during the rest of the stable periods input will not go to output only during the rising edge the input D would be given as the or would be given to the output Q correct. So, similarly, we want to design a negative edge triggered D flip flop then the clock would be connected here and the negative clock, so the inverter would be connected to this particular gated input.

So, because it is this, I would also like to assert that this is one of the implementation, this is only a one implementation of D flip flop. Now, since, two these are fundamental circuits, they are used almost everywhere, almost in different combinations different places in digital circuits. So, they are designed, then there is a lot of focus on designing such kind of flip flops.

So, they are also designed directly using transistors, using CMOS transistor, PMOS and more, so different type of type of transistors and there is a optimization within. So, this is one of the simplest implementation which is using gates, but they can also be directly implemented using transistors or there could be other mechanisms also to implement edge triggered D flip flop.

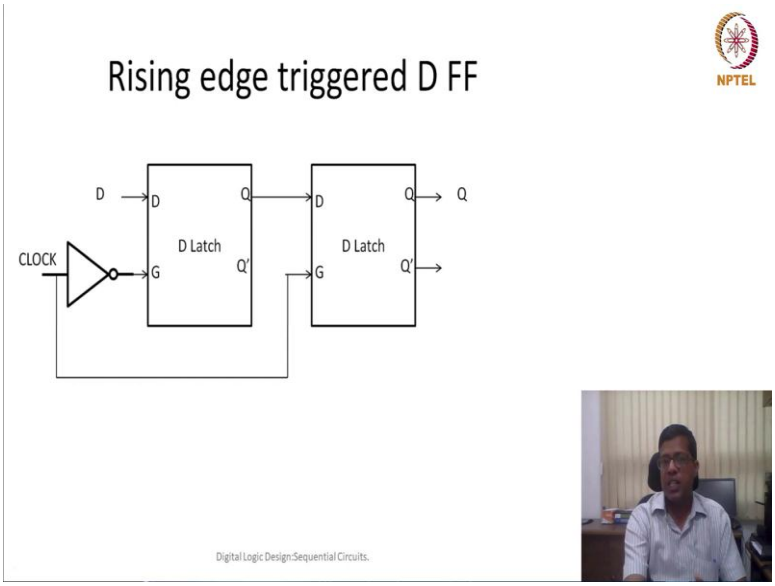
So, but one thing from this analysis or this kind of a circuit we understand that ideally, the behaviour of the D latch or D flip flop is that during the positive edge whatever is the value of D would go to Q but in a practical scenario, there is some delay of this D latch there is some delay of this D latch. So, there has to be some other timing constraint which we have to adhere.

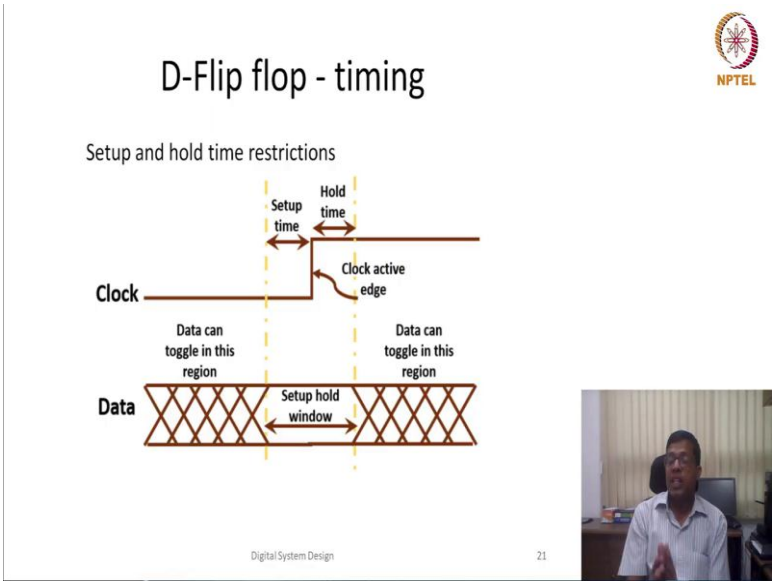
(Refer Slide Time: 28:06)



These timing constraints are represented using this diagram. Now, let us say this is my clock signal and this is my active edge or this is the edge at that time we would be taking the input and giving it to output. So, we have to make things work perfectly we have to make sure that D or the input will not change at least setup time before this rising edge and at least hold time before this rising edge. Now, for to make it towards the ideal arrangement the setup time has to be minimum whole time has to be minimum, but yes practically there would be non-zero values.

(Refer Slide Time: 29:05)





So, let us look at in this circuit whenever this circuit was, this clock was negative or clock was 0 that means this gated input was 1 so, that we whenever it was 1 at least sometime before that what should be, whatever should be the value of D, will go in the output of first latch. Similarly, because it was available at some time before the latch and whenever this particular gated input become 1, then only this can go and transfer to the Q.

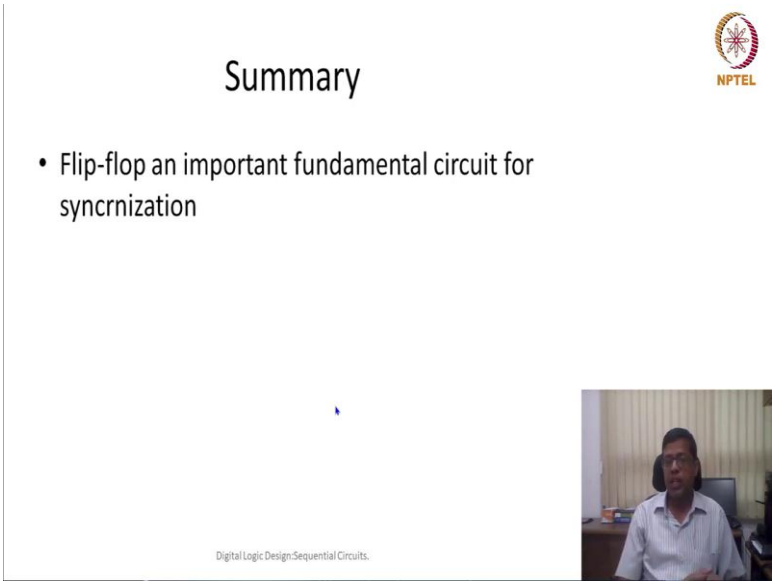
So, essentially there has to be some setup time involved and during this window my output my input should not change to make sure that the output is correct. The other information which is there so this is one restriction, which is on the setup time and whole time, there is one additional timing thing which we have to understand that whenever input will change, so whenever there is a positive edge. So, after this positive edge there would be certain time

after which the output would be reflected on to Q or at the output of my D flip flop. So, that would I can call latency of my flip flop.

So, the latency would be calculated from the rising edge or the active edge, but setup time again would be calculated with reference to my active edge. So, setup time would be the time when my input should be stable and hold time after this clock edge, active edge of the clock for how much time my input has to be stable.

So, my input can toggle anywhere here or anywhere here it does not matter, but during the setup and hold time window input should not change. That is a restriction which will make sure that my circuit is will function correctly. And these are also called design constraints. So, whenever we are designing any circuit, we keep this timing constraint in front of us.

(Refer Slide Time: 31:28)



So, with this we conclude today's lecture. So, in summary in today's lecture, we have seen how an edge triggered flip flop can be designed, how this fundamental circuit helps us in synchronizing different signals and how the synchronization is made, how it is internally implemented all of these things we have studied in today's lecture. And in our next lecture, we will see other implementation other different type of flip flops which are possible and their applications. Thank you very much.