

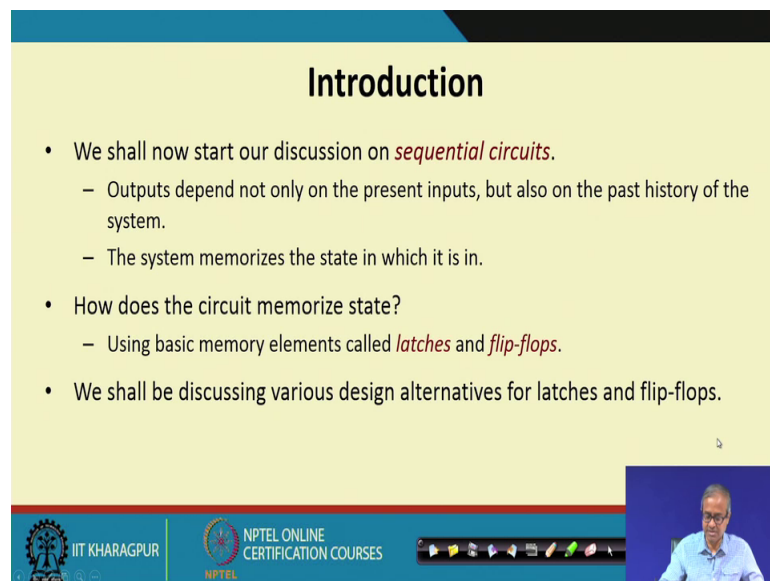
Switching Circuits and Logic Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Latches and Flip-Flops (Part I)

We now start our discussion on sequential circuits. Let me try to explain what we are going to talk about in the next few lectures. So, far the kind of logic circuits, the design, the optimization that you had talked about they were basically concerned with combinational circuits. If you recall, in a combinational circuit the outputs depend only on the inputs that we apply at that particular point in time.

In contrast, when you talk about a sequential circuit there the output also depends on some kind of previous history what kind of inputs were applied earlier, we should be talking about examples later. So, for this kind of a sequential circuit we need to store or memorize some information ok. So, the topic of our discussion in this lecture is Latches and Flip-Flops; the first part of it. These latches and flip flops essentially constitute the basic building block using which we can store some information; well information in this case we are talking about bits.

(Refer Slide Time: 01:45)



Introduction

- We shall now start our discussion on *sequential circuits*.
 - Outputs depend not only on the present inputs, but also on the past history of the system.
 - The system memorizes the state in which it is in.
- How does the circuit memorize state?
 - Using basic memory elements called *latches* and *flip-flops*.
- We shall be discussing various design alternatives for latches and flip-flops.

The slide includes logos for IIT Kharagpur and NPTEL Online Certification Courses, along with a small video inset of the professor in the bottom right corner.

We want to store the 0s and 1s ok; so, let us see. So, as I have just now mentioned that for a sequential circuit; the outputs of the circuit will depend not only on the inputs that

we have applied, but also on some kind of past history of the system; which in other word means that the system has to memorize which state it is in. Let me take a very simple example to illustrate what we are trying to talk about.

When you imagine that you have build a system; an automated system whereby whenever someone enters the room, the light turns on automatically. Well, it is not a very big thing you can have a sensor which will give a pulse or one and whenever there is an interruption in some light when a person enters. And there will be a circuit which will be turning on the switch of the lamp whenever that kind of signal comes.

Now, what you want is whenever that person comes out of the room; the lamp should automatically be turned off. So, there can be multiple persons in the room right; so, the system should remember or memorize how many persons have entered the room so far; suppose there are 10 persons who have entered. So, unless and until all the 10 persons come out of the room exit the room; the lamp will not be switched off.

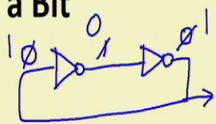
So, for the circuit like this; this circuit or this system has to remember how many persons are currently in the room and that constitutes the state of the system. This is the basic idea behind this design of sequential circuits; well there are 2 types of sequential circuits you can talk about later synchronous and asynchronous ok; mostly we shall be discussing about synchronous sequential circuits ok

So, in order that the circuit can memorize the states we have to have some kind of basic memory elements called latches and flip flops. So, these latches and flip will be our basic memory elements; I mean using which we can store the state of the system and this latches and flip flops there are various type, we shall be discussing the various types and how they are designed in the subsequent discussions.

(Refer Slide Time: 04:37)

Basic Idea Behind Storing a Bit

- Consider a cascade of two inverters with feedback.
 - Constitutes a stable state of the system.
 - The circuit can memorize the output values as long as power is on.
- In practice, we need something more.
 - We should be able to set the output values to 0 or 1 as per our requirement.
 - Need some additional circuitry.
 - The exact functionality distinguishes between different types of latches and flip-flops.



The slide features a blue header with the title 'Basic Idea Behind Storing a Bit'. Below the title is a list of bullet points. The first bullet point is underlined. To the right of the text is a circuit diagram showing two inverters connected in a feedback loop. The output of the first inverter is connected to the input of the second, and the output of the second is connected back to the input of the first. Handwritten labels '0' and '1' are placed near the inputs and outputs. The slide footer includes the IIT Kharagpur logo, NPTEL Online Certification Courses logo, and a Windows taskbar.

Now, what we are trying to say is how to design a circuit in which we can store some information, store the value of a bit whether it is 0 or a 1. So, storing a bit is the most basic form of information storage; whenever you have multiple bits we can store a word ok. So, let us see what is the basic requirement for storing a bit electronically; what do you mean by storing, where do we store ok?

You see when you talk about some device like a magnetic disc; there we have an idea well magnetically you are trying to store something; we are trying to create tiny magnets on the surface of the disc; which side is the north pole, which side is the south pole that will determine whether you are storing a 0 or a 1. But in an electronic circuit; how do we store? Let us try to understand that.

Well here what we are saying is that if we have a cascade of 2 NOT gates; NOT gate means inverters with feedback that will constitute the most basic form of storage system, this will constitute a stable state of the system. What do you mean by that? Let us try to see. Suppose we have an inverter like this, there is another inverter like this 2 NOT gates; they are connected in cascade with feedback.

Now in such a circuit let us say suppose assume that somehow this input is at logic 0; well if this input is a logic 0 the output of the NOT gate will be at logic 1, output of this NOT gate will again be at logic 0 and that same 0 is being fed back here; you see this is a stable state of the system.

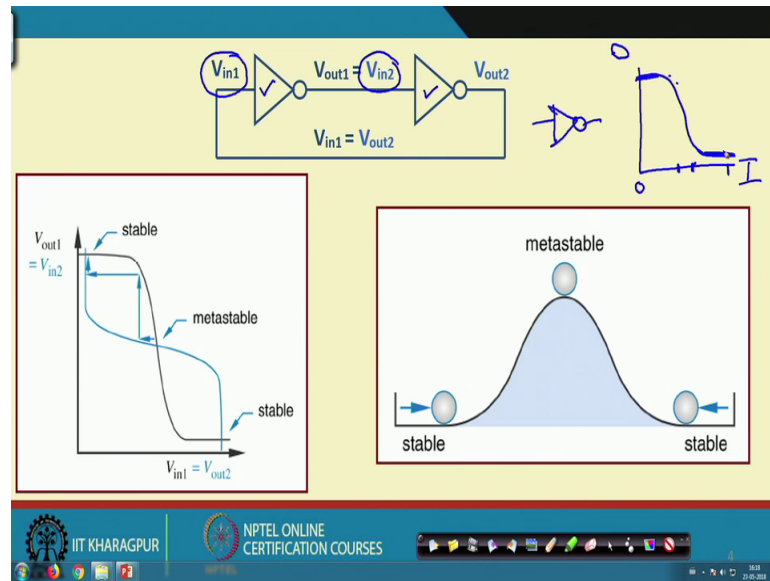
Whenever we set this system in this state; the values of those 0s and 1s will never change, it will memorize the state as long as power is on; the circuit is powered on it will remember the state. Let us say if we take the output from this point; let us say from this point if we take the output this point is at state 0. Now somehow if you are able to change it; suppose if you are able to make this 1, which will make this 0 which will again make this 1 and the same 1 is fed back here then we say that the output is 1.

So, this configuration; a pair of NOT gates or inverters connected in a chain like this is the basic memory element and constitutes stable state. And as you can understand whenever a circuit is in this kind of a value; this 1 will remain a 1 because no other circuit is trying to make it 0 because this NOT gate output is also 1; the same 1 is again applied as an input here.

So, it constitutes a stable system this is never deleted or modified unless we change something externally right. But in practice we need something more you see what we saw? We saw that we have a stable system that can be designed using 2 cross coupled NOT gates, but what we have not talked about so far is how to set the output values to 0 and 1 depending upon our wish, depending upon our requirement.

So, we also need to have that kind of a circuit that from outside we can set the circuit state to either 0 or to a 1 as per whatever we need ok. So, for that reason; for that purpose we need some additional circuitry and depending on what kind of additional circuitry you are using and what is the exact functionality of your circuit; we can distinguish between various kinds of latches and flip flops accordingly and this we shall be discussing in our subsequent slides.

(Refer Slide Time: 09:17)



Let us look into this cross coupled inverter once more in a little more detail. So, in the diagram one top we are showing this 2 NOT gates which are connected in the chain like this. The input is V_{in1} , the output of this gate is V_{out1} which is applied as the input to the next gate which are calling V_{in2} . And the output of this gate V_{out2} is again fed back here and is applied as V_{in1} . Now you see one thing whenever we have a single inverter like this; a single NOT gate.

So, if we try to plot the input versus output voltages; the input voltage and this is the output voltage; let us say O ; the curve typically shows a behavior like this. So, when the input this is 0 volts and this is the maximum. So, when the input is 0 the output is at a high level and there is a point where the; the outputs slowly starts to change and beyond the point, the output is more or less at lower level this is how inverter works.

Now, let us see this diagram in this diagram we have plotted V_{out1} on this side and V_{in1} on this side; this is the first inverter. And the solid black line shows the characteristic of the slide I showed for this first inverter. Now for the second inverter this V_{in} and V_{out} are reversed right? This V_{out} of the first gate will be V_{in} of the next gate and V_{out} of the next gate will be V_{in} of the first gate.

So, to show the characteristic curve of the second gate; we just reverse the axis. So, along the y axis we show the input V_{in2} and along the x axis, we show the output V_{out2} . And you have a similar curve like this; this is the blue curve shown like this. Now we

are showing both the plots on the same curve because the output of one gate is actually the input of the other gate fine.

So, what we say here is that there are 2 regions of the system one out here and one out here where the system is stable. Stable means whatever is applied in V in 2 that is regarded as logic 0 and the output will be logical high and this V out 2 will also be logic 0. So, V out 1 will be at logic high; V in is 0, V out is high. So, there are 2 region; if it is V in is high, it will be low output will low.

If V in is low the output will be high, but in the middle region it is not a very stable thing because if the input changes; the output changes very sharply. If you look at this curve in the middle part the slope is very high. So, if the input changes a little bit the output changes pretty sharply; that is considered as a meta stable or an unstable state. So, if the circuit is in a metal stable or unstable state, it will try to move into one of the stable states as quickly as it can ok.

So, in terms of a pictorial representation if this is my input voltage range. So, on the 2 sides; we have 2 stable states and whatever is in between this is unstable or metastable states. So, it is like a ball if it is in a metastable state; it will always try to roll into one of the stable states right let us move on.

(Refer Slide Time: 13:34)

Design of Latches

- A latch is a temporary storage device that has two stable states, 0 and 1.
 - Level sensitive storage element (also called bistable multivibrator).
 - A flip-flops is a special kind of latch where a clock signal triggers the change in the stored value.

Input

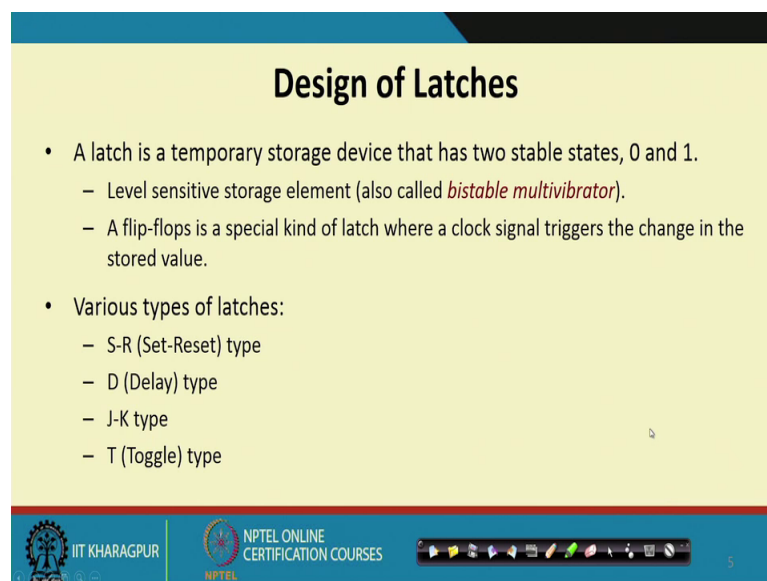
The slide features a hand-drawn diagram of a step function labeled 'Input'. The signal starts at a low level, remains constant for a period, then transitions sharply to a high level and remains constant thereafter. The slide also includes a small video inset of a speaker in the bottom right corner and a footer with logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

Now, let us come to what we are actually trying to discuss here; so, how to design the storage elements? So, we first talk about the design of storage elements called latches. A latch is nothing, but a storage element or a storage device that can store a single bit of information ok; it can store either a 0 or a 1 which means it has 2 stable states 0 or 1.

Because there are 2 states 2 stable states and it can switch from one to the other; it is sometimes also called a bistable device. And it since it switches between 2 states it is it is also called bistable multivibrator; it vibrates between 2 states something like that. So, this is also a name which is given bistable multivibrator.

And latches are level sensitive means suppose I am applying some input. So, when I am applying input let us I have applied a 0 or a 1. So, when I am applying a 0 and applying a continuous 0; when I am applying a 1 I am applying a continuous 1. So, this circuit will react to the voltage level ; that means, how long I am applying a 0, how long I am applying a 1; it will not look at exactly when I am changing it from 0 to 1 or 2 so.

(Refer Slide Time: 15:28)



Design of Latches

- A latch is a temporary storage device that has two stable states, 0 and 1.
 - Level sensitive storage element (also called *bistable multivibrator*).
 - A flip-flops is a special kind of latch where a clock signal triggers the change in the stored value.
- Various types of latches:
 - S-R (Set-Reset) type
 - D (Delay) type
 - J-K type
 - T (Toggle) type

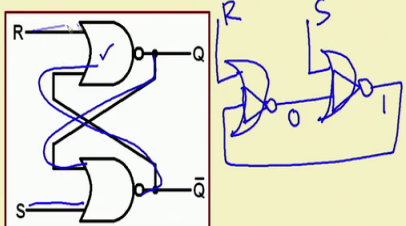
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Level sensitive means this circuit operation will depend only on the voltage levels; level of the voltage that I am applying in the inputs alright. So, the different kind of latches and also flip flops that we shall be talking about are Set Reset or S-R type Delay; D type, J-K this is also the short from Jack and King, but it is not very commonly referred to; so, we simply call it J-K and T means Toggle; these are the 4 most widely used flip flops and latches we shall be discussing them


(Refer Slide Time: 15:56)

The Set-Reset (S-R) Latch

- Consists of a pair of cross-coupled NOR or NAND gates.
 - Two inputs (S and R) and two outputs (Q and Q').
 - The output can be set to 0 or 1 by applying suitable values on S and R inputs.



S	R	Q	Q'	
0	0	No change	No change	
0	1	0	1	
1	0	1	0	
1	1	?	?	Invalid



We start with the most basic kind of a latch; this set reset latch or an S-R latch; this is the most fundamental of the storage element. And let us try to understand how this works; well a set reset or a S-R latch consists of a pair of cross coupled NOR or NAND gates.

Now, in this diagram I am showing cross coupled NOR gates; so, why I call it cross coupled? Because you see there are 2 NOR gates, the output of one is fed to the input of the other. Now let me talk about one thing; I mentioned earlier that 2 inverters connected in cascade with the feedback constitute a stable storage element. Now suppose this inverters; I modify into a 2 into gate, let us say I make this as NOR gate, I make this also a NOR gate and I apply another input here and I apply another input here.

So, one of them I call it S another one I call it R; now the purpose of R and S are to forcibly set these lines to 0 or 1 whatever you want from outside which for a simple NOR gate chain it is not possible to do because there is no external input. So, S-R flip flop is just an extension this latch is just an extension of that inverter pair feedback with feedback that kind of a chain but we have replaced the inverter by a 2 input gate in this case NOR gate.

So, you see these 2 circuits are actually identified. So, here we have drawn in a slightly different way the NOR gate with the R input this is the one. The output of which is going to the input of the other NOR gate, output of which is going to the input of the other

NOR gate and the other input is coming S; other input is coming S. The output of this coming to the input the output of this again is again coming to the input here R is applied ok. So, these 2 circuits are same ok.

Now, let us try to look at the functional behavior. So, how does a S R flip flop behave? Well this I shall be explaining in more detail later, but let us look at it just at phase value; whatever I am saying here. Suppose I apply S equal to 0 or R equal to 0 which means in this circuit S is 0 and R is both are 0 and 0. So, the output Q and Q bar one is the complement of the other ok; Q and Q are the complement Q and output and its complement. So, it will be storing some value.

Suppose Q is storing 0 and Q bar is the complement of that. So, if we apply R as 0 0; you see this 0 is coming here 0 and 0 this output will be 1 and 1; 1 and 0 output of the NOR gate will be 0; so, there is no change. Similarly if you see this is 1 and this is 0 then also no change; this one will remain 1 because 0 0 is 1 and 1 0 is 0.

Therefore if you apply 0 0; there is no change in the outputs right; this is the first one. Now suppose I apply 0 1; S equal to 0 and R equal to 1, now for NOR gate if we apply a 1 to one of the inputs; the output is forced to become 0 right and this is 0 and 0 this will become 1. So, you see for this case Q is 0; Q bar is 1 ; so, whenever I have to set the output Q to 0, I will have to apply S equal to 0 and R equal to 1.

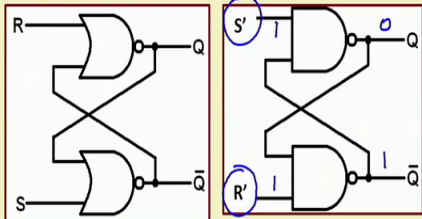
Similarly, if I apply the reverse S equal to 1 and R equal to 0; then this 1 is the input of this NOR, this gate will be it will become 0 and 0 0 and this will become 1; so the output is 1. So, whenever I want to make the output 1 I have to apply 1 0. Now the last row I shall be explaining a little later it says that if I apply 1 1; this will be an invalid combination, well I shall be explaining this a little later.

You see means our basic purpose is fulfilled; we have been able to set the output to 0, we have been able to set the output to 1. But if you do not want to change we also have an input is in which you can keep it in this same state as it was. So, we have designed a very basic storage element which can store either a 0 or a 1 or it can also remember whatever it was if I apply a 0 0 in S and R; there will be no change in the outputs fine.

(Refer Slide Time: 21:30)

The Set-Reset (S-R) Latch

- Consists of a pair of cross-coupled NOR or NAND gates.
 - Two inputs (S and R) and two outputs (Q and Q').
 - The output can be set to 0 or 1 by applying suitable values on S and R inputs.



S	R	Q	Q'	
0	0	No change	No change	
0	1	0	1	
1	0	1	0	
1	1	?	?	Invalid

State Table

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, this is the circuit diagram using NOR gates; now just using NAND gates also you can design a very similar circuit. The only change with respect to this stable well just one thing; let me tell you also as I mentioned for a combinational circuit when we talked about the input output behaviour in the form of a table we called it a truth table, but for a sequential circuit such an input behavior we referred to as a state table.

Because it also captures the state of the system not only the input output behavior ok; so, this table that was showing here this actual is a state table right. Well here for the NAND one; the only difference is we are applying S bar here and you are applying R bar here; the not of S and R . Now you can verify the operation; suppose S is 0, R is 0 the first row which means S bar is 1 and R bar is 1. For an NAND gate let us say the output was 0 this was 1; so, 0 and 1 for a NAND gate output will remain as 1 and 1 and 1; output will remain as 0; so, same state no change.

Similarly, for 1 and 0 see if I apply S 0, R 0 there will be no change right. So, if I apply S equal to 0, R equal to 1; S equal to 0 means S bar is 1, R equal to 1 means R bar is 0. NAND gate any input 0 will force the output to be 1; 1 and 1 this will be 0 ; so, you see the output is 0. Similarly if you reply here let us if we apply the reverse S equal to 1 and R equal to 0; S equal to 1, R equal to 0. So, this 0 will make it 1, this 1 we make it 0; so the output is 1 right. So, now let us see that why the fourth combination is considered to be invalid.

(Refer Slide Time: 23:54)

The slide features a title "Why is $S = R = 1$ an invalid input?". On the left is the logic symbol for an S-R latch with inputs S and R, and outputs Q and \bar{Q} . In the center is a NAND-level diagram with two NAND gates. The top gate has inputs \bar{S} and Q, and the bottom gate has inputs \bar{R} and \bar{Q} . The outputs are Q and \bar{Q} . Handwritten blue notes on the right state: $S=0, R=0 \Rightarrow NC$ and $S=1, R=1 \Rightarrow ?$. The bottom of the slide includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a small video inset of a speaker.

So, let us try to just explain this with respect to this NAND level diagram. Now what you have seen so far is for S equal to 0 and R equal to 0 combination, the behavior of the circuit is there is no change right. And we are saying that S equal to 1 and R equal to 1; this is an invalid combination; you should not apply right. Now let us see that what will happen; if I apply S equal to 1, R equal to 1 here; let us see. S equal to 1 means S bar is 0, R equal to 1 means R R bar is 0.

So, because both the gate inputs are 0 ; so, both the output should be forced to become 1. So well; so I mean you may argue one way of arguing means well I am calling them Q and Q bar. Now because both of them are 1 and 1; so, it is wrong that is why this is invalid well, but I can argue in a different way; suppose I have designed the circuit in such a way that only Q is available to you, Q bar is not available.

Then I can say that if I apply a S is equal to 1, R equal to 1; Q will always be 1 as the circuit shows. So, there is no ambiguity here; so, where is that ambiguity is coming in or why you are calling it invalid? Well we are calling it invalid because you see this S equal to 0, R equal to 0 combination is supposed to be a combination which will not cause any change. So, what is the meaning of S equal to 0, R equal to 0; which means I have applied the 1 here and I have applied a 1 here.

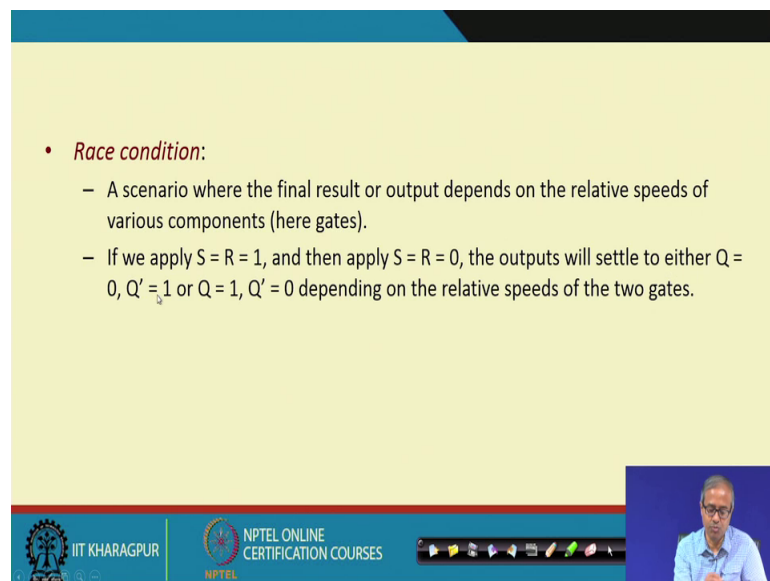
So, my circuit state was this 1 1 it was; now if I apply 1 1 you see this 1 and 1; this will become 0, this 1 and 1; this will also become 0. Now if both these 2 gates are exactly of

the same speed then this one will be changing to 0 at the same time. Then again 0 and 1 will be 1; 0 and 1 will be 1; then again change back to 1 at the same time.

But in reality what happens 2 gates are never of the exactly same speed, one will be slightly slower than the other because of fabrication differences and differences and characteristics. Suppose this gate is faster and this gate is slower; so what if this is gate is faster than this 1 1 was there; let us say 1 1 was there earlier. This 1 and 1 will cause this to become 0 first and this 0 will now be fed back to make it 1.

So, now we have a stable state 0 and 1, but if this gate was faster than the reverse would have happened we will get 0 here and 1 here. So, the confusion is after applying S 1, R 1; if we apply S 0, R 0 you cannot definitely say whether the output will become 0 or 1; it will depend on the relative speeds of the 2 gates this is a phenomenon which is called race condition ok.

(Refer Slide Time: 27:31)



The slide features a yellow background with a blue header and footer. The main content is a bulleted list defining a race condition. At the bottom right, there is a small video inset of a man in a blue shirt. The footer contains logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a navigation bar.

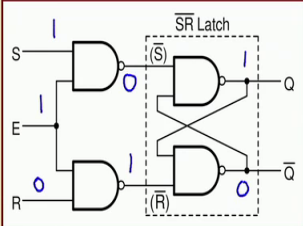
- **Race condition:**
 - A scenario where the final result or output depends on the relative speeds of various components (here gates).
 - If we apply $S = R = 1$, and then apply $S = R = 0$, the outputs will settle to either $Q = 0, Q' = 1$ or $Q = 1, Q' = 0$ depending on the relative speeds of the two gates.

So, this is exactly what I explained; race condition is a scenario with a final result depends on the relative speeds of the gates. So, I said if I apply S-R 1 and then apply S R 0; so you cannot definitely predict what the output will be, it can be either 0 Q bar 1 or Q 1, Q bar 0 depending on the speeds.


(Refer Slide Time: 27:54)

Gated S-R Latch

- A gated latch requires an *enable* input (E).
 - When $E = 1$, the latch is *active*.
 - When $E = 0$, the latch is *de-active* and the outputs do not change.



E	S	R	Q	Q'
0	X	X	NC	NC
1	0	0	NC	NC
1	0	1	0	1
1	1	0	1	0
1	1	1	?	?

 NPTEL ONLINE CERTIFICATION COURSES

Now, this one thing if you want to extend the design of an S-R latch to a gated latch. So, you see here we have made a very small change; we have added an enable input.

We have added the third input and have added 2 NAND gates here and this NAND implementation was already there; the basic S-R latch; S bar R bar we are just added this and we call this as S we call this as R. We say when E equal to 1; the latch is active when E equal to 0 the latch is not active. So, now this state table will look like this; when E equal to 0 then irrespective of what you are applying in S and R; there will be no change NC means No Change.

Next 4 rows indicate when the latch is active E is 1; if an S is 0, R is R is 0 just like previous since no change; 0 1 will cause the output to become 0, 1 0 will cause the output to become 1 and 1 1 is an invalid combination. Now let us workout one combination let us say enable is 1; let us say 1 0 be fourth row 1 0; S is 1, R is 0. This is an NAND gate 1 and 1 will make it 0; 1 0 will make it 1, this 0 will make this output 1; 1 and 1 this gate will make the output 0. So, the output Q will be 1; 1 circuit right.

So, this circuit works perfectly right and just there will be using an enable input you can either activate it or deactivate it by setting E to 0 or 1.

(Refer Slide Time: 29:48)

Gated S-R Latch :: Alternate Design

E	S	R	Q	Q'
0	X	X	NC	NC
1	0	0	NC	NC
1	0	1	0	1
1	1	0	1	0
1	1	1	?	?

Now, you can have an alternate design instead of the NAND latch, you can have been NOR latch only difference is instead of NOR gate you need to have and gates instead of NAND gates which are earlier, here you will have only and gates.

So, I leave it an exercise for you to verify that this circuit also works as an S-R flip flop and the same state table is satisfied here ok; these are just 2 alternate designs.

(Refer Slide Time: 30:22)

Gated D Latch

- A *D-latch* has a single input *D*.
 - When the latch is enabled, the value at *D* gets stored in *Q*.
- A *gated D-latch*:

E	D	Q	Q'
0	X	NC	NC
1	0	0	1
1	1	1	0

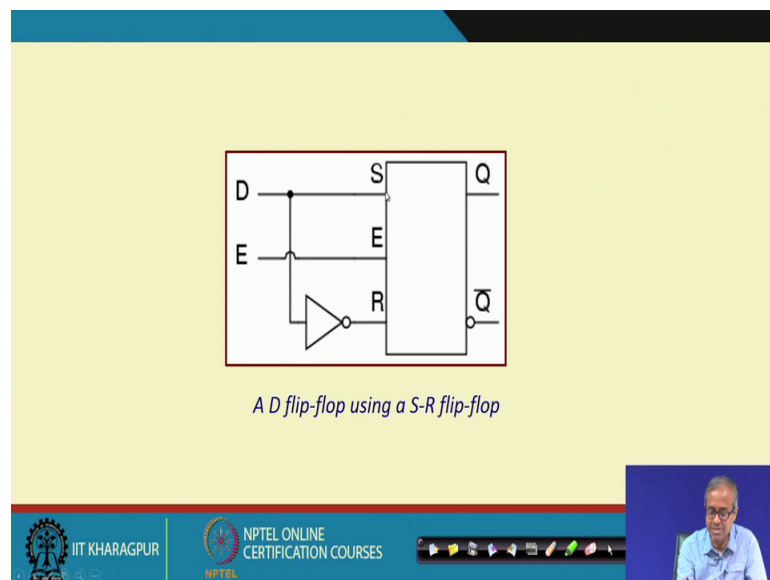
Now, come to another kind of a latch which is much simpler type which is called a D-latch. Now what is the D-latch just as compared to an S-R latch which has 2 inputs S and

R a D-latch has a single input called D. And basically when you enable the latch the value you are applying a D will get stored inside the latch very simple. This is how you can design a D-latch; you see the second design is easier.

The first part which you see here is just an S-R latch where this is your S input and this is your R input. You take an inverter D you directly connect to S and D bar you directly connect to R; you get a D latch. Well if you start with the NAND based design then your design can be like this you can save one inverted here; it means the one gate less is required D enable. And the output of this when enable is one D will become d bar this bar can be fed here and you can directly connected here.

So, these are 2 alternate designs of a D flip flop; D-latch and for D-latch how it behaves? If the enable is 0 irrespective of D; there is no change in the output. If you are enabling the latch if D is 0, output will become 0. If d is 1; output will become 1, whatever is the input that same value will get stored in the output this is; however, the gated D-latch works.

(Refer Slide Time: 32:14)



Just as I mentioned in the diagram that if you have an S R latch available with, you can construct a D-latch just by using a NOT gate. D you apply directly to S and with a NOT you can apply to R. So, we say of course we can see later that how any arbitrary kind of flip flop or latch can be converted into other type of flip flop.

But in this lecture we stop here and in our next lecture, we will we shall be looking at the other kind of designs of latches and flip flops, what are their characteristics and how they can be converted from one form to the other.

Thank you.