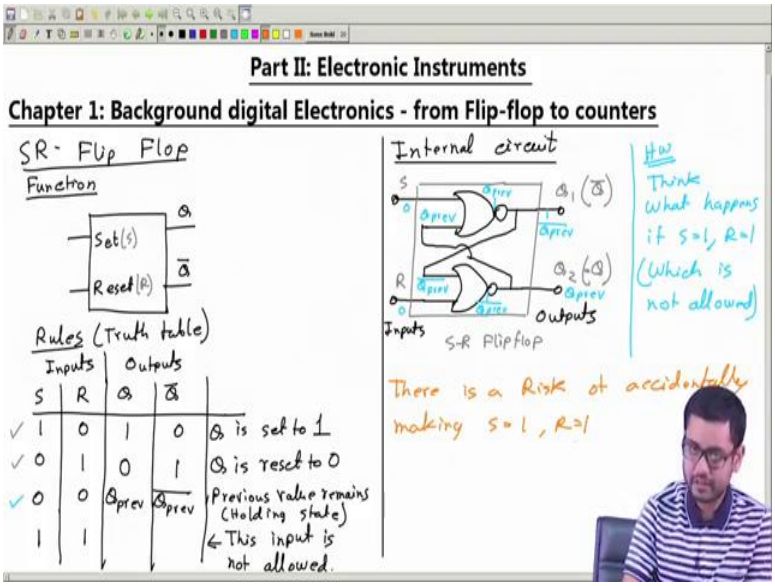
Electrical Measurement And Electronic Instruments
Prof. Avishek Chatterjee
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 47
Background: From Flip Flop to Counters - I

(Refer Slide Time: 00:33)



Welcome. Today we are going to start our second part in this course which is on Electronic Instruments. So, far we have studied classic mostly classical electromechanical instruments; all instruments that we have studied were electromechanical there were no electronic circuit at all. There were no digital electronics neither analog electronics nothing.

Today, we are going to start electronic instruments which will which can be either digital electronics base instrument or analog electronics based instruments. So, before we go in the core or the heart of this course, this part, we will spend some time in reviewing the required backgrounds, particularly some simple digital electronics and some analog electronics mainly op-amps.

So, we will spend significant amount of time in understanding basics of digital electronics and op amps and then, we will dive down into the heart of this course. So, this is our first chapter today and in this chapter, we will start our discussion from flip flops and we will go till the counters, we will develop counters from flip flops ok.

So, let start with what is called an RS flip flop or SR flip flop. What is it? So, this is a black box which has 2 inputs, we call them set, reset; S an R. S for set; R for reset and which has 1 or 2 outputs we can call them Q 1, Q 2 or we can call them Q and Q bar.

How does it function? So, the way it functions is that ok, let me write the rules ok. So, this is a digit this is a digital equipment, digitally the circuit therefore, the inputs can be only logical high or logical low; true or false. So, it cannot be a continuous voltage, it can be only a high or low it can accept only 2 values ok.

Now, the rule is. So, I that means, I can give four different choices of inputs; high and high, low and low, high-low, low-high so on or 0 1, 0 0, 1 0, 1 1 so on. So, I can write the rule as set, reset. So, this 2 are the inputs and outputs are Q and Q bar ok.

So, if I give say 1 and 0, the rule says Q will become 1. If I give 0 and 1, Q will become 0. So, if the way it behaves like if I give. So, we call it you can call that we are pressing the button set ok. So, if we press this button set, if set is pressed Q is set to 1.

So, Q is set to 1. So, here we write Q is set to value 1 and if we press the reset button ok; by making reset equal to 1 and set equal to 0, this will Q is reset. Reset means is gone back to 0 reset to 0. This is how we call it and then, then what else can we give? We can give 0 0 and we can also give 1 and 1. What will be the output in these 2 cases?.

This case is not allowed. This input is not allowed. What do I mean by that? If you are using this circuit, whatever this flip flop, you are never supposed to give input as 1 and 1. If you give by chance 1 and 1, what its output will be after that that is unpredictable ok. You if you give 1 and 1 you are not allowed to give, if you give 1 and 1, we cannot say what is going to happen. So, never do that 0 0.

So, if you give 0 0, what will be the value of Q output? We say that the value of Q will be same as Q previous that. What is Q previous? Q previous is the value of Q which was there before you gave this input 0 0 ok. So, before you gave this input, some value was there that could be 0, that could be 1 that value will remain ok. If the previous value was 0 after you give this input, it will remain 0 if the previous value was 1 it will remain 1 ok.

So, this we call we can say previous value remains or you call it holding; holding state. So, previous value is hold. So, this is how output Q behaves and what about Q prime, sorry

Q bar? Q bar this bar is the symbol for complement or negation. So, Q bar will be always opposite to Q. If Q is 1, it will be 0 if this is 0, this will be 1 ok. So, this will be Q previous bar and. So, these are this is the input output rules or we call it the truth table ok.

So, this is how this flip flop behaves. So, this is how it behaves from the input output perspective. We have not said yet what is there inside that is different question, but from input output behavior this is how it behaves. So, you first know this thing and then, we can talk about what is there inside internal circuit. Internal circuit is very simple, we basically have what is this a NOR gate.

This is another NOR gate. Two NOR gates; each of them will have 2 inputs and 1 output. The way we will make the circuit is that, we will connect the output of 1 to the input of other and similarly, the output of other goes to one of the inputs of the first one.

So then, there are 2 more out inputs left and then, this we will use as the inputs; where we can give some values. So, these are the final inputs of the circuit and this 2 are the outputs and then, I seal everything in a box. Then, this is become this is same as a SR flip flop ok. It has 2 inputs; it has 2 outputs ok.

So, let me call this outputs Q 1, Q 2 and let me call this say S and R. I could have just called this S and this R; no problem because the circuit is symmetric. So, I call either of them as set, the other are no problem. So, whichever you like you call set; the other one is reset.

Now, let us see how it behaves. So, if we give set a input 1 0; 1 and 0 here. Now, if this is 1, what will be the value here? This is one; no matter what the value here is ok. So, this part, this part is or this part is an OR gate. So, 1 org with anything is 1. So, here that means, we will at this point we will have 1 and this bubble is a negation complement. So, after this we will have 0 ok.

So, I have 0 here and then this 0 comes here. So, I have 0 0 0 or 0 or 0 is 0 and after this bubble, it is the opposite 1 right. So, I have 0 and 1 here. So, if I give 1 and 0, set 1 reset 0 output is Q 1 0 Q 2 1. Let me call this Q 2 as Q and let me call this as Q bar, then you see this rule come back to this rule; set 1 reset 0, 1 0 output is Q 1 Q bar 0. Here its 1 Q 1 Q bar 0 ok. We have got that. So, this is true. Now, let us check the other conditions. Let us make it opposite 0 and 1. So, what will be the output? ok.

So, now similarly this is 1; if this is 1, no matter what this value is 1 or 1 is of course, 1 if this is 1 this is definitely 0 and if this is 0, this is 0 comes here 0 or 0 is 0 here after this bubble 1. So, if I give 0 and 1 like this 0 and 1 here, then the output is Q = 0 like this; Q bar = 1 like this ok. So, this condition is also satisfied. Now, let us check say other situations. Let me give 0 and 0 and let us assume the value of Q was Q previous before I made this 0 and 0.

So, let us start with this. This equal to Q previous. This could be 0 or could be 1. Now, this Q previous comes here. So, here I have Q previous Q previous or with 0, something or with 0 something plus 0 is something that something. So, at this point we have Q previous and then, at this point after the bubble we will have Q previous bar. Now, this Q previous bar comes here.

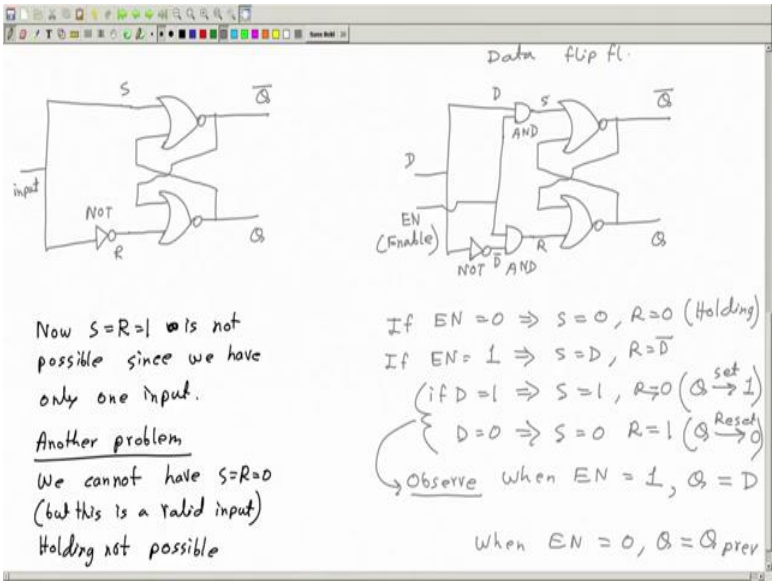
Now, this thing plus 0 is of course, this thing only. So, Q previous bar and after this bubble, this bar will be gone I will have Q previous. So, you see I started with the assumption that this is equal to Q previous and I landed up with the fact that this stays with at the same value Q previous. This value is not going to change, as long as you keep this 0 and 0 this value is not going to change. So, this is a Holding state.

This is also true for this circuit. Now, if you give 1 and 1, what will happen? I keep it as a homework, you think yourself. It will be fun ok, if you think something yourself this will be more fun than I myself telling you everything. So, homework; think what happens if S equal to 1 and R is equal to 1 which is not allowed, great. So, this is all about a SR flip flop

Now, so, what do we see? We see that there is a risk ok. So, there is a there is a risk that one may accidentally make both this set and reset equal to 1. There is a risk of accidentally say making set = 1 and reset = 1; that means, I say there is a chance of pressing set and reset simultaneously which is not allowed ok.

So, this circuit has some problem; I mean I mean when it when it comes to practical use this is a risk. So, therefore, we are going to see a modification. What is that modification? The modification is simple.

(Refer Slide Time: 16:52)



So, let me first draw the SR flip flop. Input goes to output; output goes to input ok. Now, what I will do? I will connect say NOT gate. So, this is a NOT gate or I I can also connect it, this here both will be same.

So, either here I connect a NOT gate or here I connect a NOT gate and then I sort this 2 and this is my input ok. So, this was S this was R in the SR flip flop this was Q this was Q bar in the previous diagram.

Now, I have made this modification; instead of giving allowing 2 inputs I allow only 1 input and this input goes both to set as well as reset, but I have a NOT gate in 1 of the paths. So, therefore, whatever this value is this value is of course always opposite. If I give a 1 here then I will have 1 here and 0 here, but if I give say a 0 here I will have 0 here and 1 here.

So, this ensures that set and reset can never be equal to 1 together ok. So, therefore, now S=1; S = R = 1 is not possible, since we have only 1 input connected in this way ok. So, that risk is gone.

But now there is a problem; another problem, what is that another problem? We cannot even have S = R = 0 because S equal to R is never possible in this circuit, they are always opposite. So, this S equal to R equal to 0 that is not also possible, but this is a valid state ok.

So, and but this is a valid input ok. So, we will also lose that possibility. So, that means, holding not possible; holding not possible. So, what can we do? What we do is this, we further modify this copy-paste.

Now, I will put 2 AND gates. So, this is a AND gate. It has 2 inputs. So, this is AND here also I put another AND gate and I connect this to ok. So, this has 2 inputs; this has 2 inputs; I connect one of their inputs together and I bring this out as another input. So, this 2 are now 2 inputs ok. So, let me call one as D, another as E or EN.

This EN now stands for Enable. What does it do? If Enable is 0, let us let us see if EN is equal to 0; then, what happens? This 0 goes here and here, now no matter what this value is and this value is; that means, no matter whatever the value of D is I will have 0 and 0 here ok. At set and reset; this is set, this is reset ok. So, I will have 0 and 0 at both set and reset. So, this is now the holding state.

If EN = 0 S=0 R=0

If EN =1 S =D R = $\overline{D}$

If D =1 S=1 R = 0

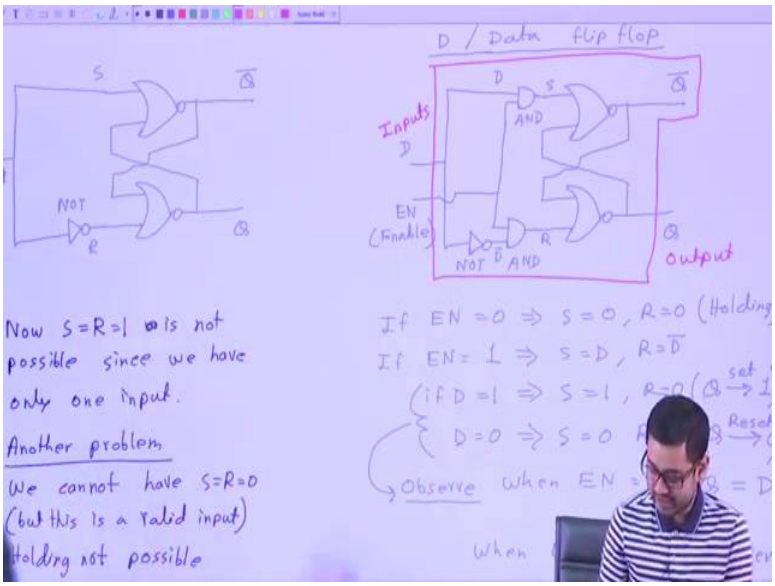
D = 0 S = 0 R =1

When EN =1 Q = D.

When EN = 0 Q = Q prev

So, this is the this is the rule of this new circuit ok. This new circuit has this rule and then, we give it a name we call it Data flip flop. Why do we call it data flip flop or D flip flop? D for D or data flip flop and that is why I actually call this input as D ok.

(Refer Slide Time: 26:17)

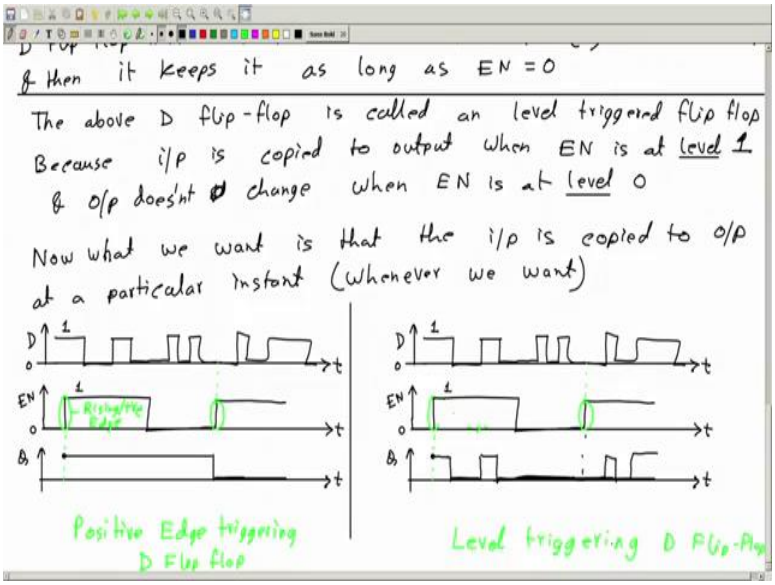


So, let me make a box. So, it has 2 inputs and it has 1 or 2 outputs. I do not need Q bar outside because Q bar is always opposite to Q. This is as redundant, if you want you can take it outside, no problem.

So, this has 2 inputs 1 output and the rule is that and the rule of this circuit let me write it is here; its already here. If you make enable 1, then output will become equal to input and if enable is 0, then no matter whatever the value of input is; data is output is not going to change. So, this is called a data flip flop because it can store 1 bit of data. How? By giving a number 1 or 0 here, we can get that number copied at the output. We also have to make enable 1.

So, if enable is 1, this data this input data is copied at the output and then if we make this enable 0, then this previous value of that data remains; it stays, it is never change. So, it memorizes that value; that is why we call that this is a memory element. This stores the data that comes here when the enable is on. So, let me just write a few words ok.

(Refer Slide Time: 28:03)



So, this D flip flop memorizes the value of the input; input D when enable is equal to 1 or enabled and then, it keeps it as long as enable is equal to 0 ok. So, if you keep the enable equal to 0; as long as you keep your enable equal to 0, if you change the value of D input output is not going to change ok. Good.

So, now next we are going to see another flip flop ok. So, before that, the above D flip flop or data flip flop is called an Level triggered or level enabled flip flop. Why? Because input is copied to output when enable is at level 1, level high and output does not change when enable is at level 0, no matter what the value of D is. So, this is sensitive to the level whether the level is 1 or 0 ok.

Now, what we want? So, this is next thing what we want. So, now, what we want is that the input is copied to output; that means, D is copied to Q at a instant whenever we want ok. So, this is like the situation is like this say we say this is let me draw some timing diagrams.

So, this is time t and say this is the value of D, it is changing. This is D which is changing with time and say we have time versus enable. So, this is EN and say this enabled is changing like this, this is changing. So, this is this is 1, this is 0; here this is also 1, this height is 0.

Now, what we want? We want the Q which is the output to change say only at the instant when enable goes from 0 to 1. So, this is a rising edge. This is a falling edge. Now, we want that the output, the input should get copied to output only during the rising edges of this enable signal. So, this is a small instant this is a very small instant of time. So, we want at this point, Q = D. Now, at this point D = 1. So, Q will become 1 here at this moment. So, Q will become 1 at this moment.

We do not know what the value of Q was before this; it could be 0 or 1 we do not know. But at this moment, it should become 1 because at this moment D = 1 and then, after that the enable signal is constant. So, we want the output not to change. Here there is a falling edge we want the output not to change even during the falling edge. Then, this is again constant. We want output not to change. So, that means, from here to here, the output will remain unchanged to this value of 1 and again, we have here arising edge.

So, at this moment once again the output can change and how will it change? It will become equal to the value of D at this one and at this moment the value of D = 0. So, now, the value of Q should go to 0 and then, it should remain unchanged as long as this is constant or as long as the next rising it does not come. This is what we want and this is what we call as edge triggering ok. Why edge triggering? Because this is triggered this change of the output is triggered only at the edge. To be more precise, we call it the positive edge triggering because this is triggered only during the positive or rising edge; it is not triggered during the negative or falling edge.

So, this is rising or positive edge. So, that is why we call this Positive edge triggering ok. Now in comparison to the previous situation ok. So, let me copy it. So, this is what you want. This is what we want now and now, let us see what we had before this is what we had before in this circuit and this is a level triggered D flip flop; what happens in this circuit, let us see that.

Let us take the same input D and same input enable. So, let us see how the output will change in this case. This is level triggered level triggering or level triggered D flip flop ok. D flip flop this is also D flip flop. So, in this case, what will happen is this. The portion where this enable is 1; that means, from here to here the output will be same as the input.

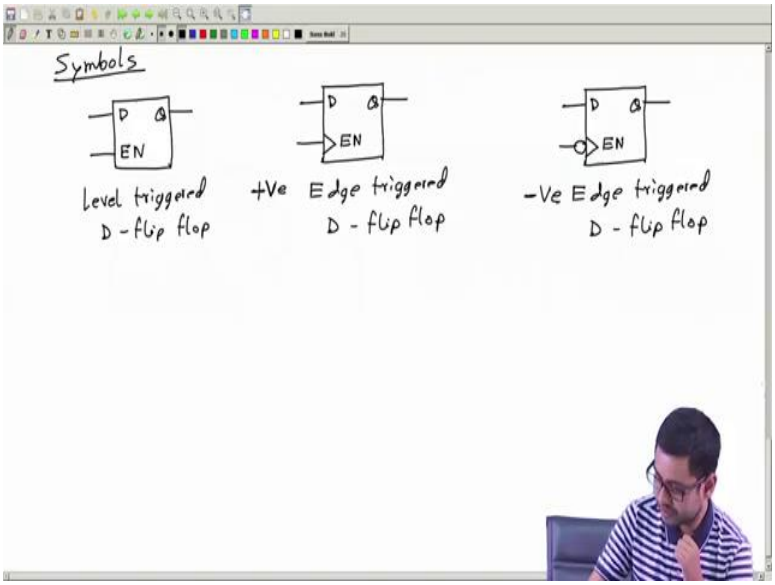
So, here the output will be 1 because input is 1; then, here the output will be 0 because input is 0 and then up to this point it will remain 0, then it will become 1 again and then it

stays 1 like this, then here it becomes 0 again; up to this its 0 fine and then the enable is off.

So, here to here no change will occur. So, this will remain like this up to this. Then, here again enable is 1. So, output can change. But it will not it will it can change, but D is also 0. So, up to this will remain 0 here it will become 1; here it goes to 0. So, now, it will copy the same value as long as enable is equal to 1.

So, this is level triggered D flip flop; this is what we have seen so far and this is what we now want. Next thing what we want is this. We have not seen this circuit yet, but this is what we want we are going to see this we are going to make this circuit.

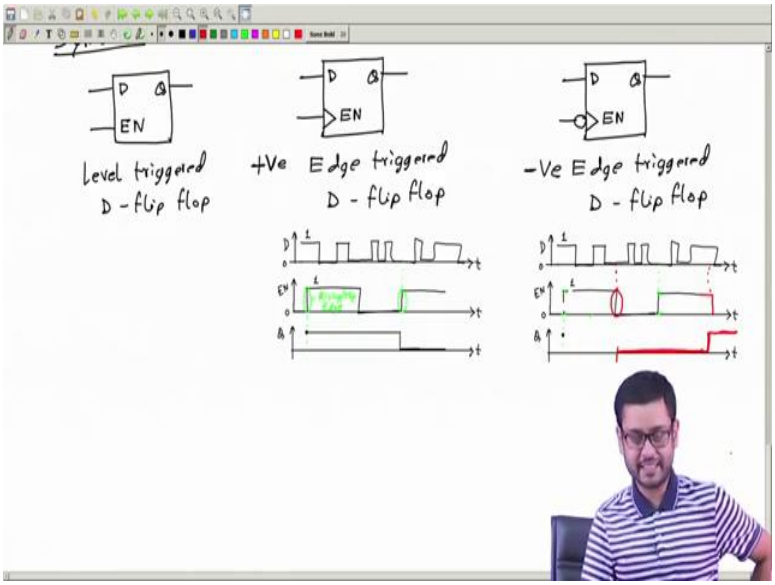
(Refer Slide Time: 38:14)



So, let us ok, before that let me tell you the symbols. So, if I draw a box with D; D means data input, Q output, enable this is also a input like this. This is this symbol is for level triggered D flip flop or data flip flop. But if I draw the same thing D Q as here, but if I put a triangle and I write enable, then this triangle symbol means this is a level triggered D flip flop sorry edge triggered D flip flop.

I can also sometimes draw it this way, where I will have a small bubble before this. This is also edge triggered D flip flop, but there is a difference. What is the difference? This is positive edge triggered flip flop and this is negative edge triggered D flip flop; that means, what does that mean?

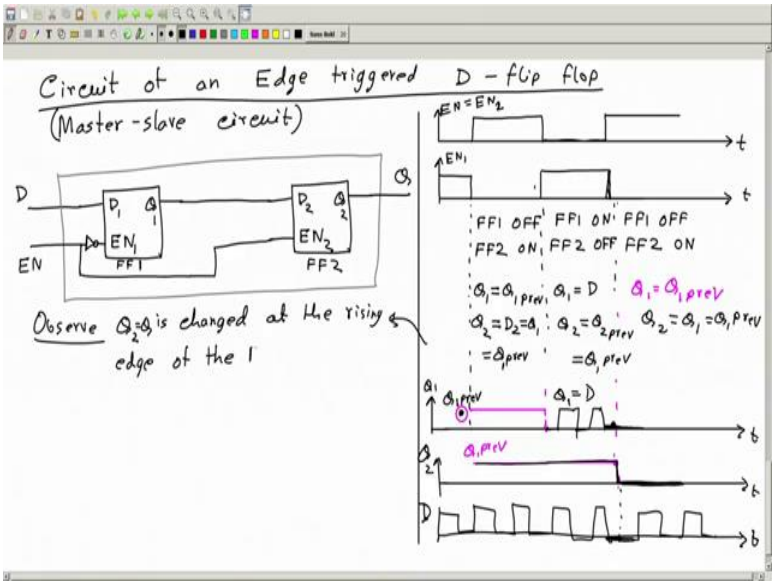
(Refer Slide Time: 40:22)



That means, this is positive edge triggered D flip flop, what we have already seen and what happens in case of a negative is triggered the only difference is that the changes occurred only during the negative edge not here ok. So, the changes occurred only during the negative edge. So, what will happen? Here Q will become 0 at this instant because data is 0.

So, Q will become 0 and it will remain 0, until and unless we get another negative edge maybe much later maybe here where this is 1. So, here it will become 1. Before this we do not know what the value is; before this whatever the value was it it was there we do not know. So, this is the symbol. Now, let us make the circuit.

(Refer Slide Time: 41:43)



So, circuit. So, the next thing we will study is the circuit of edge triggered D flip flop and this particular circuit is called a master slave circuit. This is the name. We will understand soon why it is called so. So, what we will do is this we will take 2 D flip flops; 2 level trigger D flip flops.

So, let me just take this 1 2. Now, what we will do is this. We will connect the output of the first one to the input of the second one and so, you can call this D 1 Q 1 D 2 Q 2 and then, this input will be the main input ok. So, I am going to make a level triggered circuit with this sorry I am going to make a edge triggered circuit with 2 level triggered D flip flop.

So, this 2 together will be my edge triggered circuit. So, the output of the first one goes to the input of the second one and this input is the main input, you can call it the D same as D 1 and this will be the final output or main output you call it Q which is same as Q 2 and this 2 enable what I will do is, I will connect them together via maybe a NOT gate.

So, I can put the NOT gate here and take the input like this I will call this enable EN, call it EN 1, call it EN 2 or I can also do with this way I connect it here and put the inverter here, the NOT gate here. So, this is a NOT gate; both will work. One of them will work as a positive edge triggered circuit, another will work as a negative edge triggered circuit. Let us see how this one works ok. So, what I will do I will draw once again some timing diagrams ok.

So, let me first draw enable say maybe this is a like this, then I can draw enable 1 and enable 2 ok; enable 1. So, I just write EN is equal to N 1 correct because sorry EN is equal to N 2 because enable goes directly to enable 2 ok. So, I do not have to draw it separately. Now, let me draw enable 1 enable 1 is nothing but the negative of enable 2 or enable. So, 1 here 0 here and like this.

Now, when enable is enable is say 0 let me ok; let me start from here and let me call this as flip flop 1 and flip flop 2 FF 1 and FF 2, here I can write FF 1 is off and FF 2 is on right because enable 2 is on. Here I write FF 1 on and FF 2 is off up to this instant ok.

Then, once again I can write FF 1 off FF 2 on. Now, if FF 2 if say if FF 1 is off first flip flop is off this flip flop is off, then whatever the value of D comes here will not go to this Q 1.

Q1 = Q1 prev Q1 = D

Q2 = D = Q1 Q2 = Q2 prev

.

Now, let us see what happens here? Here flip flop 1 is on; flip flop 1 is on means Q 1 is equal to D 1. D 1 is equal to D and what about Q 2? Q 2 Q 2 will stay at Q 2 previous because flip flop 2 is off. What is Q 2 previous? Q 2 previous is the value of Q 2 before this region; that means, here at this instant before this region and what was that value? Q 2 was so from here we see Q 2 previous was Q 1 previous. So, this is equal to Q 1 previous which was the value of Q 1 at this instant. So, Q 1 previous is the value of Q 1 at this instant.

So, let me draw Q the value of Q 1 and Q 2 as well t t. Here I draw Q 1 and Q 2 ok. Now, Q 1 I do not know what the value of Q 1 was, so before this instant. So, let me just before this instant call this value whatever that is, it can be 0 it can be 1 I do not know. I am just calling it Q 1 previous; this value. Just before this instant ok.

So, this value of Q 1, this can be 0, this can be 1; this value of Q 1 will remain here up constant and then, Q 1 can change depending on the value of D. But Q what about Q 2? Q 2 is Q 1 previous here. So, this is equal to Q 1 previous and here also this is equal to Q 1 previous up to this moment.

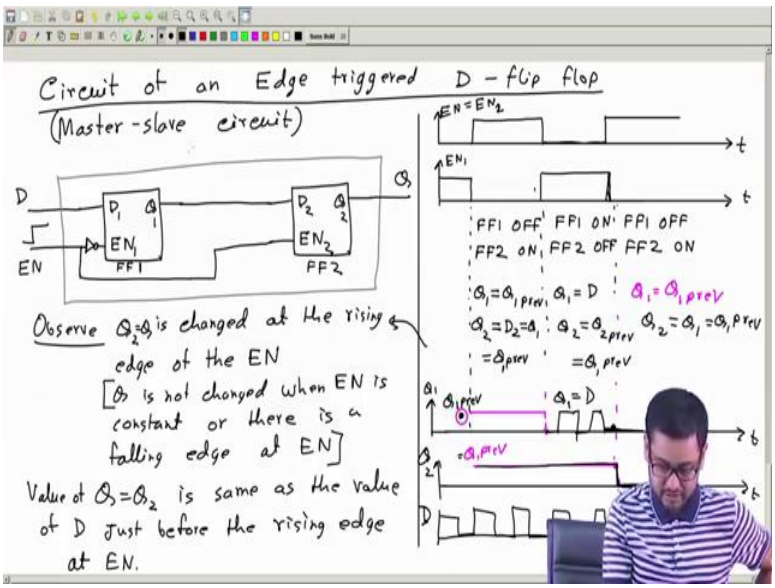
So, up to this moment, this is constant; it is not changing ok. Now, after this what happens ok? Once again, let us analyze this part ok. So, here I can safely write by copying this Q 1 is equal to Q 1 previous which now means is the value of Q 1 just before this instant which is here. What could be that value that will be depend on the value of D. So, I have to draw D as well. So, that this be the D and say the value is like this I do not know ok.

So, then at this instant D is 0. So, Q 1. So, here to here Q 1 is equal to D; here Q 1 is equal to D. So, let me just copy this pattern. So, 0, then 1, then 0, then this thing then 0 ok; so that means, here at this instant it is 0. So, then this value is the value of Q 1 previous ok. I I request you do this analysis yourself ok, because otherwise it may be a bit clumsy it is very difficult to make you understand; if you do not try it yourselves ok. So, this value I am saying is the Q 1 previous known.

Q2 = Q1 = Q1prev

So, with all this observe Q 2 is changed when enable goes from negative to positive at this instant ok. So, at this instant follow of the pointer ok. So, at let me show you this. So, at this instant ok. So, at this instant Q 2 is changed. Q 2 which is same as Q that is the output is changed at the rising edge of the enable ok.

(Refer Slide Time: 56:18)

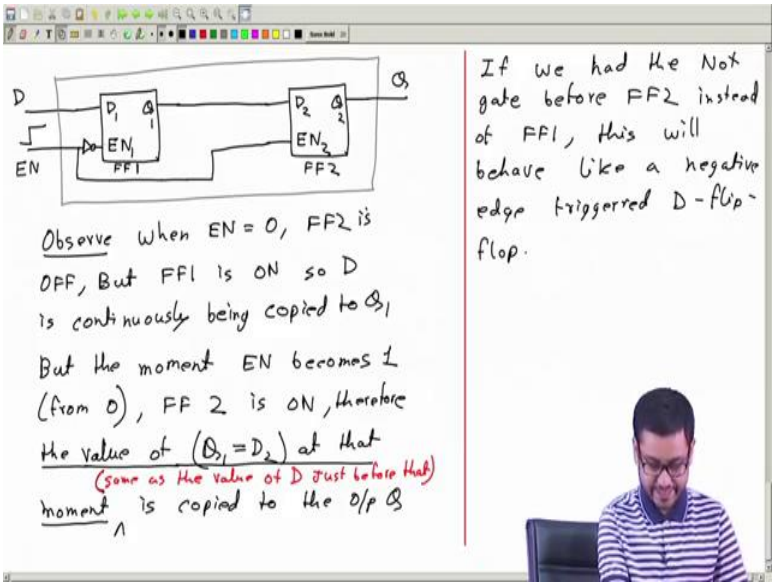


And cube is not changed. Q is that means, Q 2 is not changed when say enable is constant equal to 1 or equal to 0 or even when there is a falling edge ok. Q 2 is not changed when enable is constant or there is a falling edge negative edge at enable ok.

So, it changes only at the rising edge and it does not change at the falling edge and the value of Q, value of Q or Q 2 is same as same as what? It is the value of Q 1 previous; Q 1 previous is the value of D just before this instant this rising edge. So, this is the value of Q 2 is same as the value of D just before the rise rising edge at enable.

So, in summary, this value of D is copied to the Q from D 2 Q its copied only during the rising edge of this enable signal. So, therefore, this circuit behaves like a rising edge sensitive positive edge sensitive D flip flop ok. So, you can just to denote it you may you may just this is. So, this you can denote with this symbol. So, what we have made is a rising edge sensitive; a rising edge triggered D flip flop ok.

(Refer Slide Time: 58:36)



So, let me just tell the story in a different way. So, I told you the story with timing diagram ok. This is one way to verify that this is a edge sensitive edge triggered diagram, another way of understanding the story is this. So, observe that when enable is equal to 0 ok.

Then, flip flop 2 is off; that means, the non-transparent. But FF 1 is on. So, D data is continuously copied; continuously being copied; present continuous tense continuously being copied to Q 1. This is a temporary or intermediate states intermediate value.

But the moment enable becomes 1 from 0 flip flop 2 is on. Therefore, the value of Q 1; therefore, Q 1 which is same as D 2 is copied. Therefore, the value of Q 1 same as D 2, at that moment is copied to the output Q ok.

So, the moment this second flip flop is turned on, the value of Q 1 goes from here to here and this value give that value. So, the value at that moment is nothing but let me write it in a different color same as the value of D just before that moment ok.

Because D was being copied from here to here, this was on the D was coming from here to here and it was getting stored temporarily here, it was not the second one was not transparent. So, the second one was holding the previous value, but the moment the second one is stand on the last value of D goes through D 2 to Q 2 ok. This is how it works.

So, therefore, therefore, if we had the NOT gate before say FF 2 flip flop 2; that means, here instead of flip flop 1, this will behave oppositely like a negative edge triggered D flip flop ok. So, this is positive edge triggered, if we keep this connected directly and put the NOT gate here, then that will behave like a negative edge.

Thank you.