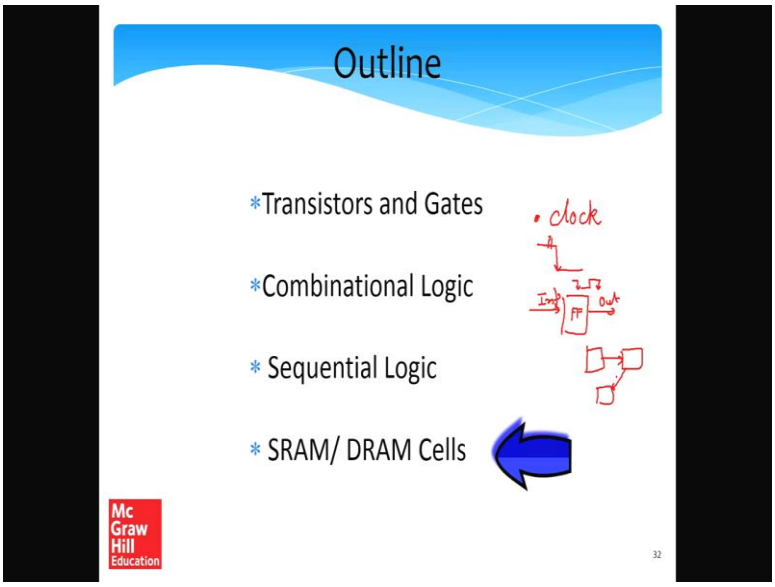
Computer Architecture
Prof. Smruti Ranjan Sarangi
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Lecture – 15
A Primer on Digital Logic Part-II

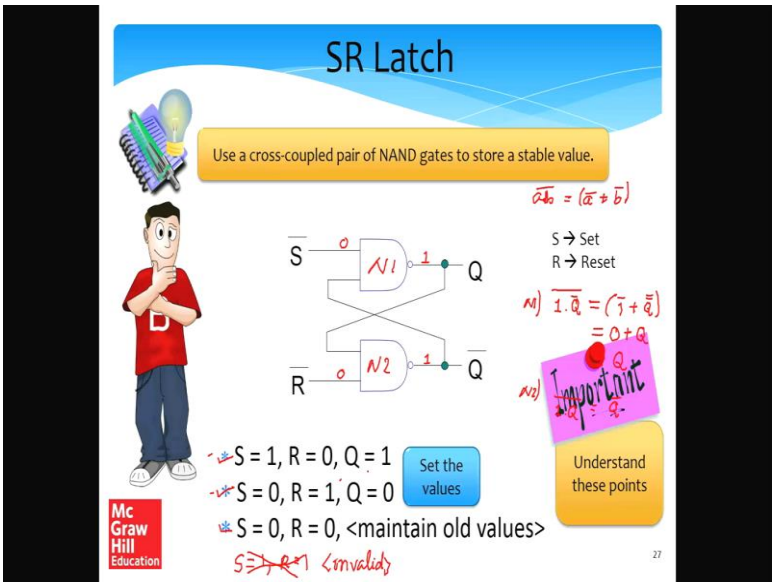
Let us now take a look at sequential logic. So, sequential logic is slightly different from combinational logic.

(Refer Slide Time: 00:26)



So, here the idea is that we save a value in a structure, and then we do some amount of competition on it. So, this called a logic cloud right we have a lot of gates and so on. And then again the output of this is again stored in another storage element.

(Refer Slide Time: 01:02)



So, the first question that we need to answer is that how we how do we store a single bit right, how do we ensure that we are able to store a single bit and the value sort of remains for as long as you want to keep it. So, the idea is that a store a bit we use a cross coupled pair of NAND gates and that will store a stable value. So, this particular structure over here is called an SR latch and so, S stands for set. So, what does set mean? A set means set the value of the output which in this case is Q to 1. And R means reset. Reset means set Q to 0. So, the SR latch has 2 inputs which is basically the complement of S and the complement of R. And the output and there are 2 outputs one output is Q and the other is the complement of Q.

So, in a sense it is a single output, but available and it is normal form as well as it is complemented form as simple as that. And why do we say it is cross coupled, the reason being that the output of the first NAND gate is an input to the second NAND gate right, let us maybe call this n 1 and let us call it n 2. So, the output of n 1 is an input to n 2 and the output n 2 is an input to n 1. So, that is the reason we call it a cross coupled NAND gate.

So, let us see how it works. The way that it works is something like this. So, let us assume that S is equal to 1. So, S trying to set the value of Q to 1, we are setting R to be equal to so, we have R is 0. So what happens? So, if S is 1. So, this particular value over here is 0 because it is the complement of S and since the R is 0 the value here is 1. So, in

any NAND gate, whenever one of the inputs is equal to 1 then the output, Sorry in a in any NAND gate when one of the inputs is equal to 0, then the output is equal to 1. Right because it is not possible that and condition will be true. So, as a result if one of the inputs is 0 we can happily write the output to be equal to 1.

So, since this is 1, 1 comes over here and the NAND of 1 and 1. So, and of 1 and 1 is 1. So, the NAND of 1 and 1 is 0. So, again 0 goes there and 0 and 0 it is NAND is 1. So, this stable system one more thing that we get to see is that in this case the value of Q is equal to 1 and Q bar or the complement of Q is the complement of one which is 0. So, the first value in a sense we have verified.

Now, let us take a look at the second combination, which is the value of S is 0, the value of R is 1. So, we are trying to reset it. So, we expect Q to be 0, but let us verify this is the case. So, let me just you know erase this, such that the reader does not get confused. Do I have an eraser? So, there is no reason to adopt a low tech solution we can adopt a high tech solution all right. So, now, what can be d1? So, in this case the value of S is 0. So, S complement is 1, and the value of R we are setting it to 1. So, the complement of R is 0. So, the same logic holds that in any NAND gate if one of the inputs is 0 then the output gets automatically set to 1 as a logic of a NAND gate fine.

So, in this case one travels all the way up here, 1 and 1 the NAND of that is 0. And 0 and 0 the NAND is 1. So, what we see is that in this case if the reset signal is 1 the value of Q is set to 0, and Q bar is it is complement which is 1 all right. So, these 2 values we are verified. So, what is the connotation again, the connotation here is that S stands for the set signal, and if S is equal to true then Q get set to 1. And R is equal to the reset signal. So, R is equal to 1 and S is equal to 0, then Q gets set equal to 0 all right.

So, now let us take a look at the rest of the 2 combinations. So, the rest of the 2 combinations are like this. Let me again erase the ink on the slide all right. So, in let us assume that S is 0 and R is 0. So, if S is 0 then 0 complement is equal to 1, and if R is 0 R complement is equal to 1 right. So, as a result we will not be sure about the output of the NAND gate. So, it depends on what we have. So, let us consider the first NAND gate. So, in this case the input is equal to a Q bar and in this case the input is Q.

So, let us essentially the question is what is 1 and Q bar right. And the entire thing complemented. So, this is a NAND of 1 and Q bar. So, one thing that we can see is that

in a NAND gate, or maybe you know, let us do one more thing using the De Morgan's laws that hopefully we all know. So, De Morgan's laws basically say that a, a and b NAND is a bar or b bar a complement or b complement. So, using this we open it up, this will be 1 complement or right. So, which is equal to so, one complement is 0 or. So, 0 or Q is equal to Q all right. So, basically the output of n 1 is equal to Q. And similarly what I can do over here is that I can for the cases. So, this is for n 1 for the case of n 2, what I can do is something very similar. So, this value here is supposed to be Q, and Some reason it is Q for which is not correct.

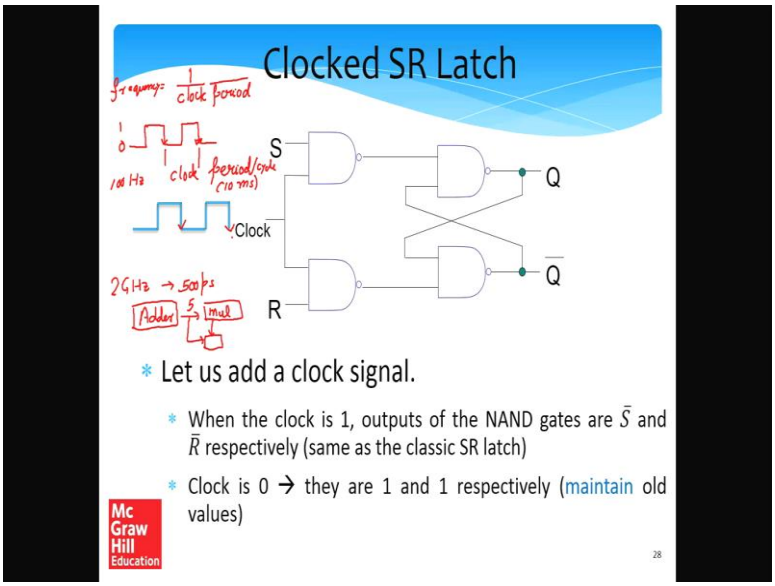
So, what we are doing here is you are computing 1 anded with Q entire thing complement. So, 1 and Q is Q. So, this is equal to Q bar right. So, basically what happens is that if you set S is 0 and R is 0, which means we are not trying to set the value neither are we trying to reset the value then the old values of Q and Q bar in a sense continue. This is a fantastic idea. So, I will tell you why. So, the way that this latch works is that we set we have a way of setting the value which means setting Q to 1, we have a way of resetting the value which means setting Q to 0 using these 2 combinations. And we have a way of maintaining the value which means whatever value is there in the latch it remains that can be done by setting S is 0 and R is 0.

So, both S and R are rounded. The latch will maintain it is value essentially forever which is exactly what we wanted to do. So, naturally readers will ask what happens, if he said S is 1 and R is 1 well. So, let us take a look at this this is somewhat a tricky case and this is a typical question that is asked in exams. So, if we set S is 1 and R is 1. So, S complement will be equal to 0 and R complement will be equal to 0. So, moment 0 is 1 of the inputs of the NAND gate the output is equal to 1. And so, both Q and complement of Q will be equal to 1 say in a sense this is an invalid input right. It sorts of producer something that we do not like, so it is invalid as an input, so you cannot set and reset the latch at the same time. So, we will not consider this input at all; will only consider the top 3 inputs.

So, essentially what are the top 3 inputs telling us, what the top 3 inputs are telling us is that we have a way of setting the latch resetting the latch and maintaining the value. So, mind you these are very important points. So, it is necessary to understand these points very nicely very clearly, and what I would advise the reader to do the listener of this video, is to basically look at all of this combination there is only 4 combinations, for SR

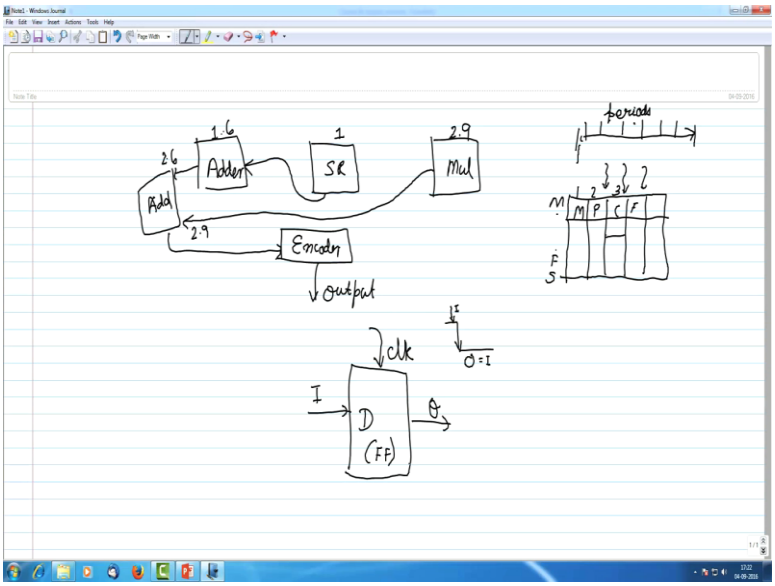
for S and R there are only 4 combinations 0 0 0 1 1 0 and 1 1 for all of these 4 combinations work out the results same way that I did and convince yourself that, if S is 1 then essentially we set the value if R is 1 and S is 0 we reset, otherwise we maintain all right.

(Refer Slide Time: 12:15)



Let us move forward. So, let us now consider the idea of having a clock. So what is that clock? So, basically consider maybe for explaining a clock the best would be to actually go out of this power point window and basically look at draw a slightly larger diagram. So, consider a digital system. So, typical digital system would have millions and not millions, but billions of components right in today's world.

(Refer Slide Time: 12:58)



So, there will be lots and lots of components right. So, I am drawing these as blocks.

So, we will have a lot of blocks right. Maybe you know this is an adder and this thing stores there are some SR latches here, there is a multiplier there is an encoded over here. So, let us assume that we want some data to come from the SR latch get you know from a series of set of SR latches get added. Similarly, we want then we want a multiplier to take that data. So we want some data coming out of the multiplier some from the adder we want to added once again right, and then we want some of these bits to choose 1. So, we then want to input these bits into the encoder and finally, we are interested the output of the encoder.

So, let us assume we have a circuit of these form. So, clearly there is a lot that is happening right. So, when designers will design it they need to ensure that the right input reaches the right circuit block at the same time. And with billions of transistors this is very difficult. Some degree of scheduling is actually required and if this amount of scheduling is not provided then essentially the entire thing will go everywhere. So let us because the latencies of these units will actually be very different. So, it is possible that the SR latch takes one nanosecond, the adder takes 1.6 nanoseconds the multiply takes 2.9 nanoseconds. So, actually if you see at this this adder the first input is ready at t plus 2.6 nanoseconds in the same way and the other one is ready at 2.9. And finally, you

know some data then again comes out. So, managing the timing of all of this would actually be very difficult.

So, what do we do? So, what do we do in school, what we do in school is that we have a timetable. And so, a typical timetable at least in Indian school the way it looks like is something like this. That we divide the entire school day into multiple blocks. You know they, so in India they are essentially called that is the part of India that I am from they are called periods. So, essentially in my school we used to divide the entire school day into 7 periods right. So, that the way that the weekly timetable was made is that. So, we actually had a 6-day school. So, from essentially Monday till Friday and Saturday, the way the time table was made us that we used to have a matrix and each matrix at this point was a period and so, then we may be the first period had maths the second had physics the third had chemistry then maybe we used to go out and play foot all right.

So, this is the way that the entire day was divided. So, this made scheduling very easy for example, if physics teacher could finish her second period in my class and then go and teach the third period. So, let us say this is the first second and third could go and teach the third period in another class. So, it made scheduling really easy. Then maybe go and teach the fifth period in another class. So, how where periods demarcated? So in my school people you know they whether good all days or the late 90s. So, not all the students you know used to have watches and even the watches were not in order sometimes.

So, what was done is that we used to have a big bell, and the bell used to sound at the end of every period. So, after that students used to themselves. So, teachers used to go to another class room. So, the entire that is the signalling was done on the basis of a bell that used to ring at the end of every period right. So, then something is used to happen the new classes to start again the bell again the bell and so on.

So, can we have a similar bell in a digital circuit, that is pretty much you know this tell me really useful if you have something of this nature I can do a bell. So, basically what happens in the digital circuit is that we have a signal called a clock, which reaches all parts of the chip. So, all the logic elements will get a clock. So, a clock is basically a signal which is a square wave. And so, basically this is 0 and 1, it is a square wave between 0 and 1 the duration of 0 and 1 is equal it is a wave of this kind. So, typically the

negative edge, we will discuss why. So, when is a transition from 1 to 0 is thought of as the bell. So, all the elements inside digital circuit coordinate themselves to the end of the clock period. So, the clock period is measured from one negative edge to one more negative edge. So, negative edge is when there is a transition from a 1 to a 0. So, the time between one negative edge to the other is called the clock period right.

So, this time you know can be 2 nanoseconds 5 nanoseconds and so on. And the number of clock periods one divided by the clock period, which is the number of clock periods that you have per second, this is known as the frequency right. So, let me call this the frequency all right. So, a number of clock periods per second. So, let us say number of clock periods per second right. So, this is called the frequency, for sorry that I will just change the definition of the frequency. So frequency is 1 divided by the clock periods not the number of clock periods I am sorry.

Let me just quickly change the definition over here. What I want to write is something sorry. So, the frequency is the number of clock periods that you see is per second, which is 1 divided by the clock period. For example, if the clock period is let us say 10 milliseconds. Then what is the frequency it is 1 divided by 10 milliseconds and that is 100; 100 times per second which is called 100 hertz. So, this is exactly the same frequency is a when we go and buy a laptop or a desktop, they say that we are having a 3 gigahertz processor, what this means is that 3 billion times a second you have a negative edge inside the chips. So, this is exactly what frequency means a frequency is the inverse of the clock period.

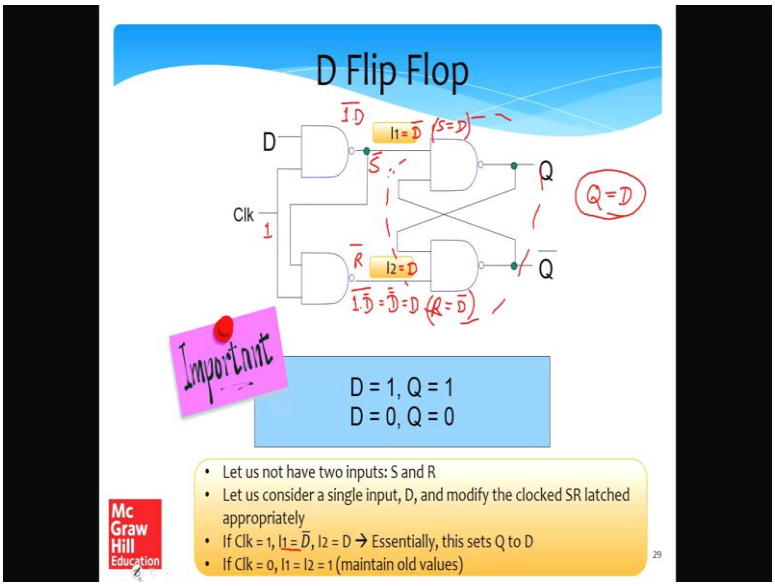
So, let us see if I buy a new machine or a laptop or a desktop and the frequency is let us say 2 gigahertzes. What this means is that every half nanosecond or 500 picoseconds we have a negative edge of the clock. And all digital elements inside a processor and including all the designers were designing them coordinate all their actions with respect to the negative edge of the clock. For example, let us see if there is an adder and the adder is providing it is output to the multiplier, the designer of the adder will tell the multiplier instead of you know specifying time in terms of nanoseconds which will just make life hard, they say that look the data the output of the adder will be given to the multiplier in 5 clock periods, or 5 clock cycles right. Clock period or clock cycle that is both are interchangeable.

So, in 5 clock cycles the data from the adder will be there with the multiplier. Similarly, if this data is going somewhere else and the data from the multiplier is also going to the same place again the coordination is done in terms of the clock periods. So, this makes the life of designing a digital circuit very simple. So, almost all digital circuits will have a clock.

So, let us take a look at what exactly happens if I have in SR latch with a clock. So, when the clock is equal to 1 right when a clock is equal to 1. So, in a NAND gate if any of the inputs is equal to 1, in this case what will happen? The output over here is 1 anded with S complement which is equal to S complement. Similarly, when the clock is 1 the output here is R bar. So, if I just go back to the previous slide this is exactly the inputs that we give to an SR latch. So, pretty much here the same inputs are being given. After that if you want to set the value we can set it S needs to be 1. So, Q will be 1, if you want to reset the value we can do that. So, in this case R needs to be 1 and S needs to be 0. So, Q will get set to 0 all right.

Now let us assume if the clock is 0. If the clock is 0 then what will happen. Let us take a look at that. If a clock is equal to 0 then the output of both the NAND gates are 1, and if the outputs of both the NAND gates are 1. So, let us go back to the previous slide and see what happened if the output of both the NAND gates are one it means that S is 0 and R is 0 which is this case and we maintain the old values right. So, basically what we can say for a clocked SR latch if the clock is equal to 0 we maintain the old values of Q and Q complement. If the clock is equal to 1 then this works as a regular SR latch. So, we can either set the value or we can reset the value all right.

(Refer Slide Time: 24:39)



So, let us now take a look at a different kind of a latch. So typically a latch with a clock; at least that is the convention that we would be adopting, that a latch with a clock will also call it a flip flop right. So, basically the previous one can also be called in SR flip flop. So, whenever there is clock we would like to call it a flip flop. So, let us look at take a D flip flop. So, what was the basic problem in an SR latch there were 2 inputs S and R. So, typically what we were doing is that we were setting S to 1 and R to 0 to set the value and S to 0 and R to 1 to reset the value. That was kind of complicated.

So, let us try to unify S and R. So, let us consider a single input D and modify the clocked SR latch appropriately all right. So let us take a look at the D flip flop is very important. So, that is the reason I have this important tag. The reason it is important is that it is very heavily used S the SR latch is a good latch to explain to a learner for the first time, but what is actually used much more frequently is a D flip flop. So, here is how it works. So, let us consider the case when a clock is 1 and a clock is 0.

So, let us first consider the case, the simpler case when the clock is 1. Well simple and not simple depends, but let us fine if you have always considered the clock to be 1 let us consider the clock to be 1.

Now, as well for the first for you know for starting to explain how the circuit works. So, the clock is 1. So, by this time you would have realise that we are computing 1 the output as 1 and D. So, this will be this is same as D bar, so I 1 is equal to D bar if a clock is

equal to 1 and if this is equal to D bar. So, D bar will come over here. So, what will compute here is 1 and which is equal to right. So, this is equal to I 2 is equal to D, all right. So, in the SR latch, then after that we have a typical SR latch. So, here we can say that the value of S is essentially equal to the complement of this which is D, and the value of R is equal to the complement of this which is D complement. So, I want to make import one important point here because a lot of students make a mistake.

So, let us look at the SR latch once again. The basic SR latch with a cross coupled pair of NAND gates. So, the inputs are not S and R. Let me repeat it once again, very important. So, let me circle this, the inputs are not S and R the inputs are S complement and R complement all right. Even, we are reasoning in terms of S and R the inputs are not S and R they are S complement and R complement. So, what is going in is S complement and R complement. Armed with this knowledge what is going in S complement is D complement. So, S is equal to D and R complement is equal to D. So, R is equal to D complement fine.

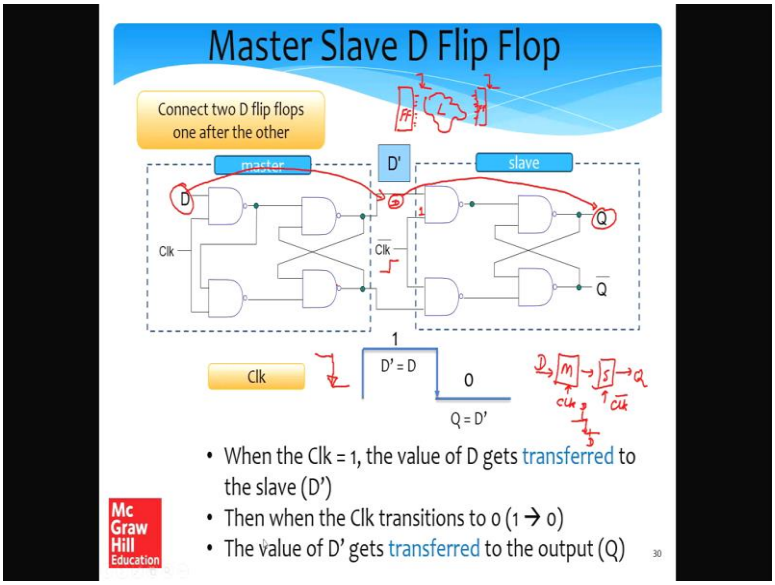
So, what will be the output well if D is 1 S will be 1? So, Q will be 1 if D is 0 then S is 0 and Q is 0. So, what we can essentially seem is that Q is being set to D all right. So, when the clock is 1 what we are doing is that we are transferring the input which is D to Q and with a delay of course, So, that is the reason is called D for delay D flip flop, and the delay is the signal passing through the logic gates, but finally, when the clock is 1 the flip flop is enabled. So, the input is getting reflected at the output and the most important point to remember is that the inputs of this SR latch at this point. So, this is the SR latch the inputs are pretty much S complement and R complement right.

So, now let me take a look at what happens if the clock is 0. Well if the clock is 0 then 0 goes to both the NAND gates. So, we have 1 and 1. So, essentially S bar is 1 and R bar is 1. So, the inputs are the set signal is 0 and the reset signal is 0. So, in this case if you go back to our basic SR latch if S is 0 and R is 0 we maintain the old values. So, we will do exactly the same I 1 and I 2 will be 1 and which means set and reset both are 0 and 0. So, we maintain the old values.

So, the working of this is exactly similar to the SR latch, is just that we have this structure over here to ensure that we can manage with a single input. So, basically what we are doing is that this is say storing 1 bit for us we have one input D, and we have 2

output one is Q and the other is Q complement. Essentially logically at least a single output and we are saving D. So, when a clock is 1 the value of D is getting stored. And it is reflecting here then at the time of the negative edge the value remains stable right. So, once it reaches 0, we will maintain the value of Q and Q bar that was set and as long as the clock is 0 the flip flop will maintain it is value all right.

(Refer Slide Time: 31:11)



So, now let us create a we still have not exploited the negative edge to the fullest extent, but let us do it by proposing a slightly more complicated structure called a master slave D flip flop. So, let us consider this circuit. It looks complicated it is not that complicated. So, let us break it down into 2 parts. So, let us consider the master. So, what is the master it is exactly the same circuit as this right, say exactly D flip flop is the master exactly the same and the slave is also exactly the same. So, you can consider 2 D flip flops with a D value goes in this is the first flip flop when this is the second 1. So, let us call the first one the master second one the slave, and then the output Q.

The catch is that the regular clock goes into the master and the complement of the clock goes into the slave that is all right. So, the diagram is very simple we have 2 D flip flops connected to each other. One the first D flip flop is called the master it takes the clock as the input. The second D flip flop is called the slave it takes the complement of the clock is input.

So, let us see what happens. So, when a clock is equal to 1 the value of D will get transferred to the output right. So, whatever is the value of D it will get transferred to this intermediate point because here the clock is 0. So, then we will have D dash here, D dash is equal to D right. So, the value of D moves from here to here right then when the clock turns 0 the value of D in moves from the intermediate point to the output which is Q. Because in the clock turns 0 clock bar actually turns 1 right. So, when the clock suddenly becomes 0 when there is a negative edge from one you move to 0 the clock the complement of the clock from moves in the other direction. So, since the clock here that is going in this signal becomes one at this particular point, the value here between the master and the slave gets to the output of the slave with some small delay of course. So, we have Q is equal to D dash right.

. So what is exactly happening? If I consider this complicated circuit and I just look in the vicinity of the negative edge of the clock, initially here the clock is 1 say the value here is let us say D, then just after the negative edge the value of Q which is the output of the master slave D flip flop will get set to D right. So, that is the only time in which the value will move from one side to the other side. And so, that is the significance of the negative edge that you can sort of thing the think of the negative edges as the bell that I had in my school. The moment the clock transitions from one to 0, essentially whatever is there at the input of a master slave structure gets reflected at the output all right.

So, that is the idea and then of course, if I disable the clock then what will happen is that this value that is there will be maintained. So, what is the advantage of having such a structure a master slave D flip flop. Well the advantages like this. So, it will be clearer actually in chapter nine, but I can give a little bit of a preview. So, the advantage is something like this that we can create a structure. So, typically what will happen while designing a large processor is that, we will have to save some values in D flip flops.

So, let me call this an FF, and then at the beginning of a clock cycle which means at a negative edge, the data will come at the outputs of the set of flip flops. We will then have a set of logic gates. So, it is typically represented as a cloud like this of logic gates, we will compute their value within the clock cycle and before the next negative edge the values will be ready, and they will then be stored in the flip flop right. In one more flip flop right because we will be ready just before the negative edge. So, then they will be

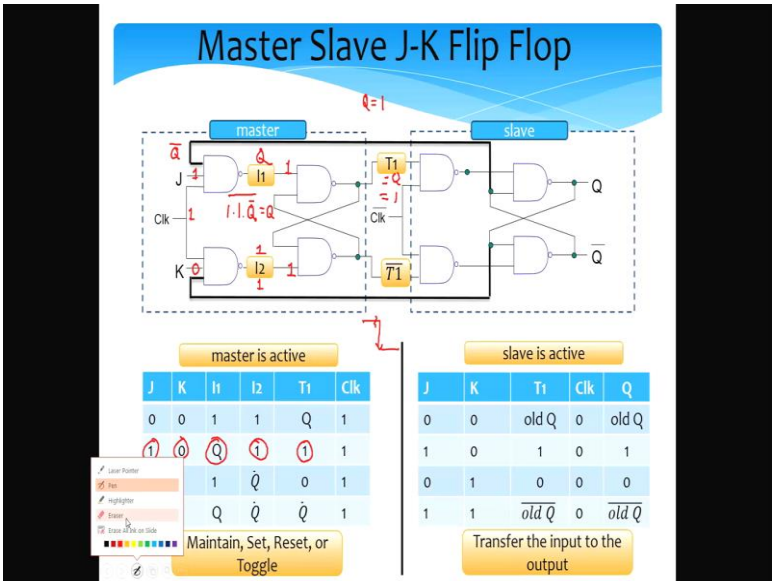
stored and in then in the subsequent clock cycle they will be available at the outputs of this flip flop.

So, this is a very important design strategy of digital circuits. It is called pipelining. And we will discuss all the advantages of pipelining in chapter 9, but the only thing that the reader or the person whose listening watching this video needs to appreciate is the followings. Let me again go back actually to my journal. So, the only thing that the reader needs to appreciate is that if this is a master slave D flip flop, and this is the input and this is the output, and this is pretty much the clock right. So, basically act the negative edge of the clock the output gets set equal to the input.

So, the input needs to sort of become stable slightly before the negative edge. So, let us assume that this point the input become stable, then clock transitions from one to 0 and that is when the output of the flip flop reflects the input. And the value in the flip flop can stay for as long as we want, we can disable the clocks in this case the value will remain. Or we can set no I think the only way is by disabling the clock otherwise the input will always reflect at the output right. For a D flip flop there is no other way, for SR flip flop we can set S and R to 0 and 0. So, that would maintain the value, but in the case of D flip flop we can disable the clock and the value will be maintained right.

So, we can store the value. That is the basic idea of what a D flip flop does, this is a very important logic element that will be using will be using in chapter 9 mostly, but at least we should take note of it now all right.

(Refer Slide Time: 38:36)



So, let us move let us now consider a similar circuit of the master slave D flip flop. It is called the master slave J K flip flop. It is slightly more complicated than that. So, let us first decompose the design into paths that we already know. So, as we see let us divide the design into 2 halves the master and the slave. So, the first part is a regular SR latch a clocked SR latch with a little bit extra, but the second part is exactly a clocked SR latch right there is no difference right. So, the second part is completely a clocked SR latch with the complement of the clock going in. And the first part is also a clocked SR latch, the only difference is that in this first and gate instead of 2 inputs there are 3 inputs that is the only difference, same as true of the second and gate over here. So, the interesting part is that this input is actually Q complement of Q which is coming in Q bar; this is what is coming in. And similarly here the complement of Q is coming in. Sorry the value of Q is coming in which is Q itself. So, that is the only difference. So, only difference is that this is the exactly the same as a master slave SR flip flop.

The only difference being that the first 2 NAND gates, actually take 3 inputs the first 2 NAND gates take 3 inputs the inputs are Q bar for the first NAND gate that takes an input J and so, the inputs here are called J and K not S and R, but J and K. So, the input where the J is coming the other input is Q bar and for K the other input is Q that is all. And the third input is the clock.

So, this is also a negative edge triggered system. So, there is a positive edge and then a negative. So, let us consider the case when a clock is 1. So, when the clock is 1 the master is active and the slave is not active the slave is maintaining it is value. So, we will not consider the slave. So, let us consider all the cases when the clock is 1, which means that we only need to consider the master. So, let us consider each of these cases, the cases are simple. So, the first case is when J is 0 and K is 0 right 0 and 0. So, it is a very simple case. So, if J is 0 and K is 0 then if one of the inputs of a NAND gate is 0, then the output is set to 1. So, with no difficulty we can set I 1 to 1 and we can set I 2 to 1 no difficulty at all.

So, if I 1 and I 2 are 1, what we have. What we will have is essentially this is S bar. So, set and reset both are 0. So, the values whatever values were there they will be maintained. So, what value was there the old value of T 1? So, T 1 is essentially we are using T 1 to label the value at this wire it is not a separate structure. So, what was the old value of T 1 the old value of T 1 was the same as Q right. So, basically whatever was in Q was in T 1 because so, let us just consider what happened before this clock cycle.

So, before this particular clock cycle they had a positive edge. So, at the time of the positive edge when this when the clock was 0, the value of T 1 was reflecting in Q and once a clock became 1 the slave got disabled. So, the old value of the T 1 that we have over here is essentially the same as the value of Q. So, as a result since we are maintaining the value of T 1 that we have is the same as the value of Q all right.

So, this is when the clock is equal to 1 just prior to the negative edge whatever, just prior to the negative edge. So, essentially we are considering let us say this instant of time. So, let me maybe draw a slightly bigger figure. So, we are essentially considering let us say this instant of time slightly before the negative edge. So, what we are saying is that if J and K both of them are 0, then the value at this point which we call T 1 will be the same as Q.

Why do we say that? We say that because a value of I 1 and I 2 are 1 each. So, the latch will maintain it is value. So, T 1 will be the same as the old value of T 1 what was the old value of T 1 let us consider this point of time. So, what was there in T 1 got reflected in Q. So, basically T 1 is the same as Q all right. So, we have considered this case which

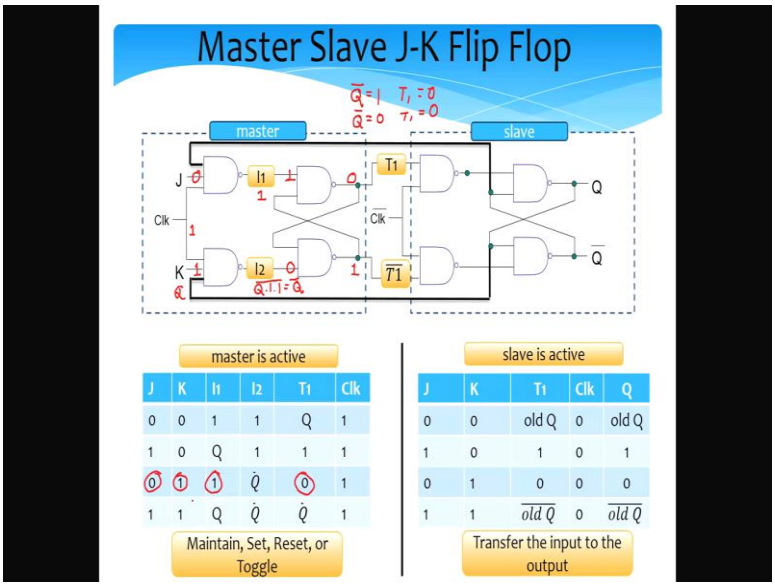
tells us that if J and K are 0 both of them are 0. Then the value of T 1 will be the value of Q and what point of time are you considering just before the negative edge.

Now, let us consider one more combination. Let us consider the next combination in which J is equal to 1 and K is equal to 0. So, if J is 1 and K is equal to 0, let us consider what we can do very quickly and the clock is 1 of course, what we can do very quickly is that we can set the output of this NAND gate to 1. Because the moment we have a 0 was an input the output of the NAND becomes 1. So, I 2 is equal to 1. So, what is happening over here? So, what is happening over here is that we are getting Q bar here right. So, the output I 1 is equal to 1 and 1 and Q bar entire thing complemented which is equal to Q right. So, I 1 here is equal to Q.

So, basically what we see is that I 1 is equal to Q, and I 2 is equal to 1. So, how do we solve this? The way that we solve this is that we consider 2 values we consider for both cases Q being 0 and Q being 1. So, let us considers the. So, this value over here is 1. So, let us consider the first case when the value of Q is 0 if the value of Q is 0 then the output here because since this is a NAND gate is equal to 1. So, T 1 is equal to 1 all right. So, when Q is 0 T 1 is equal to 1. So, we consider this case Q is 0.

Now, let us consider the next case which is Q is 1. So, let me just maybe erase, let us consider the next case where Q is equal to 1. So, here Q will be equal to 1. So, if you have 1 and 1. So, essentially both set and reset are 0. So, the value will be maintained. So, the value that will be maintained, what will we have is the same as this case which means T 1 is equal to the value of Q. And what is the point of time that I looking at just before the negative edge. So, whatever is the value of Q T 1 must have been equal to that and since the value of Q is equal to 1 T 1 is equal to 1. So, we see that in both cases T 1 is equal to 1 all right. So what is happening, in the J terminal if it is set to 1 and if K is set to 0, then irrespective of the value of Q in all cases the value at this point T 1 is being set to 1.

(Refer Slide Time: 47:34)



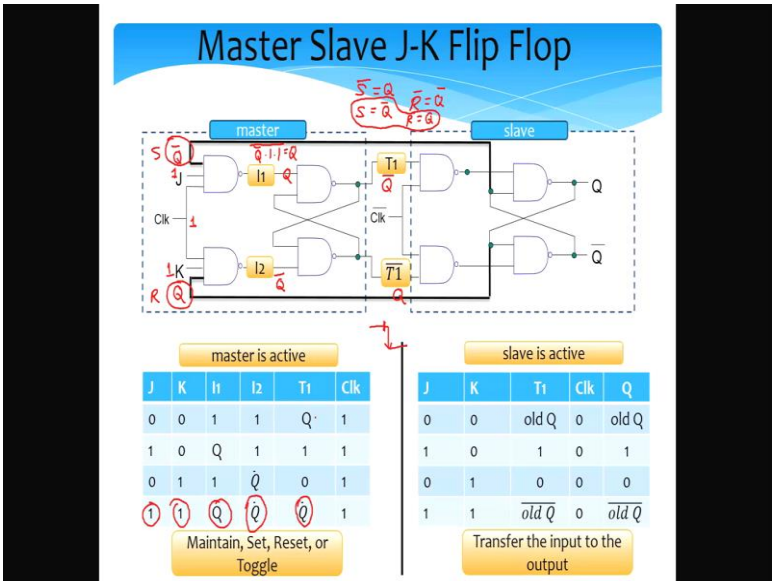
Now, let us consider the next combination. So, let us consider the next combination in which J is equal to 0 and K is equal to 1. So, if J is equal to 0 and K is equal to 1. So, what this allows is that we can nicely set the value of I 1 to be equal to 1 right. So, this cases covered. What is then not covered is basically this that what is the value of I 2. So, since the clock is equal to 1, and we are getting Q over here value of I 2 is Q and 1 and 1 entire thing complemented which is Q bar all right. So, the basically the value at this point is the complement of Q and the value at this point is 1.

So, let us do the same thing. Let us follow the same logic as we did before let us consider 2 cases. Let us consider the case where Q bar is equal to 1 and when Q bar is equal to 0. So, if Q bar is equal to 1. So, we will have 1 and 1 as both the inputs. So, the value will be maintained. So, whatever is the old value of T 1 and T 1 bar the same value will be maintained. So, what is the old value of T 1? It is the value of Q, since Q bar is equal to 1 Q is equal to 0. So, in this case when Q bar is equal to 1, T 1 is equal to 0 which is the value of Q.

So, let us now consider the other case. So, let me just erase some of the temporary values and let us consider the other case. When Q bar is equal to 0. So, if Q bar is equal to 0 then this particular point is 0 and so, nicely we can figure out the output of the and gate which is 1. So, 1 and 1 here is 0. So, we see that for both the cases irrespective of the value of Q the value of T 1 is 0 in this case. So, we write 0. So, we can think of J and K

as pretty much as S and R, if J is 1 and K is 0 the value of value of the middle between the master and the slave is set to 1. Similarly, if J is 0 and K is 1 the value at the point T 1 is set to 0. So, mind you I 1 I 2 T 1 and T 1 bar are essentially points. You know we are just referring to the values on certain wires.

(Refer Slide Time: 50:37)



So, let us now consider the last case. The last case is very interesting. So, in this case J is 1 and K is 1. So, if J is 1 and K is 1 what do we have, we have the clock to be 1, J is 1 and K is 1 all right. So, the value at this point is Q complement and the value at this point is Q. So, value of I 1 is essentially Q complement and 1 and 1 entire thing complemented which is equal to Q right. So, the value of I 1 is equal to Q and the value of I 2 is essentially Q complement which is Q bar. Given the fact that we have Q over here and we have Q bar over there.

So, essentially what is the set symbol? The set is, so mind you S bar is equal to Q. So, which essentially tells us that S is equal to Q bar and R bar is same as Q bar. So, R is equal to Q. So, taking both of these in our SR latch condition, what we can see is that the value of T 1 is a same value was the set is the same as S which is Q complement. And a value at this point is Q. So, what we have effectively it done is that we have this with Q complement at what point at the point which is just before the negative edge, we have the value of T 1 as Q complement. So, we can think of it as a toggle as like a bit flip right.

So, let me explain the logic once again. So, what happened is since all the values are 1. So, Q complement was coming here Q was coming here. So, basically this can if you think of the SR latch, this particular point is actually the S bit, and this particular signal here is the R bit. So, the S is being set to Q bar and R to Q. So, the outputs at this at the end of the SR latch will essentially be Q complement and Q. So, we are effectively toggling doing a bit flip of whatever is Q, we are setting T 1 to be the complement of Q, fair enough.

So, now let us again clean the screen. So, what is it that we have here? What we have over here is that we have looked at the point of time which is just before the negative edge, and at this particular point of time for the 4 combinations of J and K when J and K are 0 0 we have essentially we maintain the value, if J is 1 K is 0 we set the value if J is 0 K is 1 we reset the value. The interesting case which we have not seen up till now is when J is 1 and K is 1. In this case what we do is that, we essentially toggle the value of Q or we have we essentially flip the value of the bit. After this we have the negative edge. So, the clock become 0. So, the master becomes disabled and a slave becomes active. Once the slave becomes active we enter a new clock cycle right, because this is after the negative edge. So, we enter a new clock cycle.

So, in this case what we need. So, this is a regular SR latch no problems and since the clock is 0. So, basically we are considering clock bar. So, clock bar is equal to 1. So, the slave is enabled. So, we let us consider the value of T 1 which is T 1 and T 1 bar. So, essentially this is we are either trying to set or re reset the SR latch. So, in this case what did we have? In J and K were 0 the value of T 1 as Q, but that was in the previous clock cycles. So, let us refer to it in a current clock cycle as old Q. Which point of time are we considering? It is just after the negative edge right, just after the transition.

So, now we are considering the negative edges happened just after the transition and a new clock cycle. The value of T 1 is the old value of Q, the value of T 1 when J is 1 and K is 0 is 1. When J is 0 and K is 1 the value of T 1 is 0. And a interesting case when it is 1 and one it is the complement of old Q. So, what is the SR latch do in this case, well what it does is very simple it just transfers the values at it is input to the output because it is enabled. So, that that will take some time, but the time is minimal. So, all the values at the input will get transferred to the output. So, what is it that we get to see, what we get to see is that when J and K both are so, let maybe clean this screen.

So, what we get to see after the negative edge when a new clock cycle begins is that if J and K are 0, the J K flip flop maintenance it is value the value of Q is the same as old Q. So, what do you mean by old Q at the end of the negative edge the new cycle starts, at this point and this point. So, this is where we will have old Q and this is where we will have new Q. So, a new value of Q is same as the old value of Q. If J is 1 and K is 0 we are setting the value setting Q to 1. If J is 0 and K is 1 we are setting Q to 0. If J is 1 and K is 1.

So, this is the interesting case then we toggle it. So, the value of the output is the inverse is the bit complement of the previous output. So, we are essentially doing a toggle over here all right. So, this ends our discussion on flip flops. The important points that should be remembered are as follows. So, the important points are that we have the notion of a clock. Why do we need a clock well we need a clock the same reason because I had a bell in my school? It is essentially to synchronise different logical elements such that if all the inputs are ready by the time that we have the negative edge of the clock right. So, we are assuming a negative edge triggered system where at the time of the negative edge a new clock cycle starts. So, by this time all the inputs are ready.

The moment a new cycle starts, if you consider this to be a flip flop and this is the input inputs rather and this was the output. So, the moment we have a negative edge the inputs get transferred to the output. And we have a way of maintaining the value storing the value in the flip flop even across clock cycles. For example, in J K flip flop we set J and K to 0, then the value will get maintained across clock cycles. So, this is essentially a storage structure, where the inputs get reflected at the output for at the time that is a negative edge, and in a general condition the value is saved in the structure till the next negative edge.

So, with this we can create a lot of sequential logic we can create counters and we can create frequency divider we can create a host of digital circuits. So, I would like to ask the reader to look at any classic text on digital logic. So, they will find lots and lots of examples, with you know flip flops. You can also take a look at books on switching theory such as that you know you have a slightly deeper understanding. So, the book by Zechowy is one such reference where we discussed switching theory and a lot of applications and sequential circuits.

For doing you know a fairly complicated logic. So, for example, you want to make a traffic light controller. How would you do it you will essentially have a set of flip flops and you will have a clock and then pretty much will be a set of inputs and data moving between the flip flops and you can sort of realize complex logic with the help of that. So, I will not allow discuss this more. So, main aim in this chapter is to provide an overview. So, then we will move to the next part of the lecture which is memory cells.

So, what is a memory cells in the memory cell we can essentially store a lot more data you know you know you know fairly compact form, and so in a flip flop actually requires a lot of transistors a very nice and fast structure, but it requires a lot of transistors. So, as compared to that a memory design is far more compact. So, where the same area it allows us to save for more data. So, we will take a look at memory cells next. So we will do this in the next lecture.