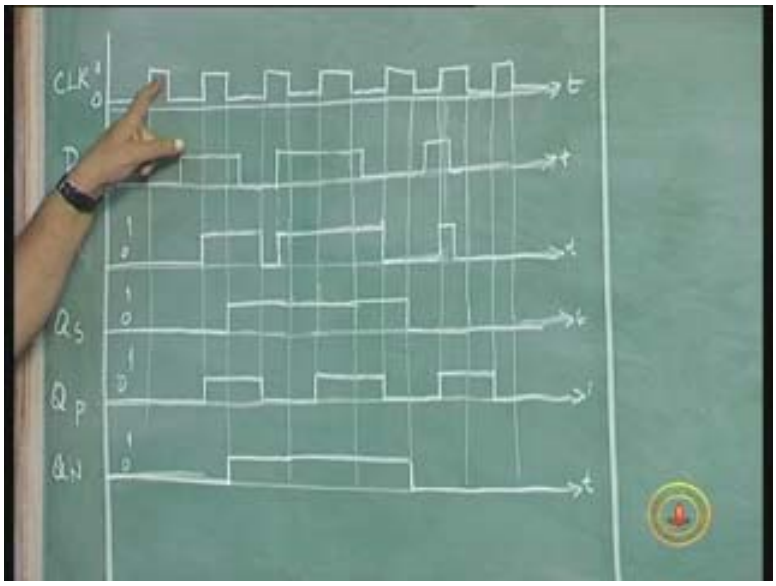
Digital Circuits and Systems
Prof. S. Srinivasan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Triggering Mechanisms of Flip Flops and Counters
Lecture # 19

(Refer Slide Time: 2:05)

We have been talking about different types of flip-flops namely the D flip-flop, SR flip-flop, JK flip-flop and T flip-flop. We also talked about level triggering and edge triggering, we talked about master slave flip-flops. So today we will look at the behavior of some of these flip-flops, different types of triggering for easy and the waveforms I am using a D flip-flop because D flip-flop has only one input but if it is SR or JK we need to have two inputs. So in order to determine the output we need to consider two inputs at the same time. So since I thought I would explain the difference of behavior of the output of different types of flip-flops in terms of triggering, master slave, positive edge triggered, negative edge triggered I am using D flip-flop as an example. The same concept may be extended to the other types of flip-flops JK or SR or T.

Now I am assuming a clock waveform, this (Refer Slide Time: 3:41) is the waveform of the clock 0 to 1 keeps going this is the clock period. Of course any flip-flop clock need not have, first of all they need not be the period of the clock from one positive edge to the next positive edge or from one negative edge to next negative edge it need not be uniform. Of course generally regular clock generation if you do using standard circuits you will always have uniform periods in general it doesn't have to be.

(Refer Slide Time 4:15)



Likewise it doesn't have to be equal, the level 0 and level 1 of the clocks need not be equal. Again, if you want to make them equal you can. You might have heard of this term duty cycle. A duty cycle is the time for which the clock period is on compared to the total period. The ratio of the on period of the clock or the high period for which the clock remains high to the total clock period is called duty cycle. Duty cycle can be half if you have positive and negative regions of the clock pulse equal in time interval otherwise it is the non fifty percent duty cycle. I have assumed an arbitrary clock but my only requirement it has to go from 0 to 1 back to 0 back to 1 back to 0 back to 1 so it is an arbitrary clock.

I am assuming that this clock is applied to different types of flip-flops, to D flip-flop of different triggering with different triggering mechanisms such as edge level triggered, edge triggered, positive edge triggered and negative edge triggered. So first I am applying it to the master slave flip-flop. In the master slave flip-flop first I would like to low, the master slave flip-flop is a level triggered flip-flop it is level sensitive, when the clock is high any change in the input D will change the output but when the clock is low of course changes in the input D will not affect the output. So if there are two flip-flops master and slave the master flip-flop is the simple level the sensitive flip-flop wherein when the clock is high this D is going to change the output. But since you are feeding it to the slave flip-flop whose clock is low and when the master slave clock is high the changes that happen will not be immediately reflected in the output of the slave that is the concept of master slave flip-flop.

So what is the change in the master output? Master output is the level sensitive output when the clock is high. This w is an arbitrary waveform of D (Refer Slide Time: 6:37) I said it is a good practice to change the data or any inputs of the flip-flop only when the clock is low but I have not adhered to that rule here I have given an arbitrary D which

changes at random. So D remains 0 to start with becomes 1 somewhere here in the middle when the clock is low again goes low when this clock is low and the second time it goes high in the middle of the clock when the clock is high it remains high for a while goes low again somewhere when the clock is low then again it goes high and this time it goes low when the clock is high. so you have all types of things changing when the clock is low both from 0 to 1 and 1 to 0 changing from 0 to 1 when the clock is high changing from 1 to 0 when the clock is high I have put all the combinations possible so that we can analyze behavior of flip-flop completely. So in a master slave flip-flop first I am drawing the waveform corresponding to the master output which is going to be fed to this slave. So master is a level sensitive level triggered D flip-flop whose output will follow the input when the clock is high.

When the clock is low nothing happens, and here even though D changes to 1 nothing happens at the output because the clock is low. At this point the clock becomes high and the D is 1 so output has to become 1. The output remains one through out the period of the clock on and since it continues to be 1 here the change from 1 to 0 during the clock negative or clock being 0 will not be immediately recognized will not immediately reflect the output and this change from 1 to 0 all the D input when the clock is low will be reflected at the output of the flip-flop only when the clock becomes 1 again and that is the behavior of any flip-flop. When the clock is low the entire flip-flop behaved as they are in a memory state. So this 0 to 1 transition shows here in the next clock edge and next time the clock becomes high but immediately thereafter when the clock continues to be high the input changes from 0 to 1 so naturally the output also has to be change from 0 to 1.

So the D flip-flop output in a level triggered mode is just you have to follow the input along with the clock high. If the clock is low you just stay put that is the easiest way to remember to do that. So this follows this continues to be 1, clock is low continues to be 1, clock is high continues to be 1, when the clock is low it goes to 0 but it not reflect at the output until the next clock happens, during the next clock period in the next clock on period so at this point in time this 0 to 1 transition will be reflected so this 0 to 1 transition is reflected here, this 0 to 1 transition is reflected here (Refer Slide Time: 10:05). And then in between it goes, this 1 to 0 transition reflects here, this 1 to 0 transition will reflect here when the clock becomes high again and since when the clock is high the input goes from 1 to 0 and the output also will become 1 to 0. This is the master output of a master slave D flip-flop or also the output of a level triggered D flip-flop. Both are same master and slave are both level triggered so we can call this either a Q output of a D flip-flop when the flip-flop works in the level triggered mode level sensitive. Level sensitive D flip-flop output or the master output of a master slave D flip-flop.

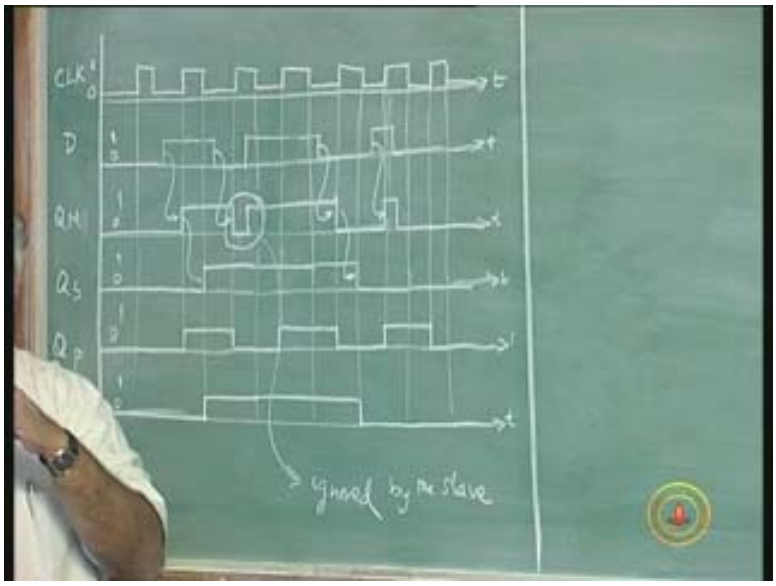
Now in the case of master slave flip-flop this output of the master is fed to the slave as the input D at the slave and the output will change according to the clock of that slave flip-flop which is the inverse of the original clock. This clock inverted is the slave clock. So now again originally it is 0 actually. We started with 1 to 0 again 0 output initially and this change in the input which reflects the change in the output of the master will go into

the slave only when the clock becomes 0. At this point in time the clock of the master becomes 0 and the clock of the slave becomes 1, the clock of the slave becomes 0 here and 1 here so this is a 0 to 1 transition of the clock for the slave so this change will come here.

Now again during this period this is on and once it is high s will this be seen now? Because now this is going from 1 to 0 of the output and can reflect the slave output only at this point only when the clock becomes 0. When the clock becomes 1 to 0 this change in the master output can reflect in the slave only at this point in time but before that what has happened is the master has become 1 again. So this change in the D in between is completely ignored. So at this point in time it was above to go to 0 but then it finds that it cannot go to 0 because the output of the master has already become 1 so it has to follow the master.

So the output continues to be 1 and this output becomes 1 to 0 which will be reflected at the slave output at the next negative edge or when the clock goes from 1 to 0. So this gets reflected here (Refer Slide Time: 14:14). This change is ignored by the slave. Likewise it became 1 here when the clock was low so when the master clock became 1 the output became 1 and continued to be 1 for a while and then it got 0 but then this one could have been reflected in the slave at this point in time before which the master output has become 0 so there is nothing to follow.

(Refer Slide Time: 15:26)



The slave follows the master output. The master was one at the beginning of the clock cycle had the 1 continued at the negative edge of the clock slave would have followed it. But before that master output has become 0 in between in the middle of the clock period so again this is ignored. Now this is interesting in one sense.

What we said was always change inputs when the clock is low but when the clock is high do not change inputs. Now automatically that is taking care of this. Only those inputs change which were made when the clock was low changed the output of the slave, this change happened when clock was high that is ignored by the slave, this change occurred when the clock was high ignored. That means this input change and this input change which are both of the value when the clock is 0 these are the ones which are reflected. So this is the slave output in which these temporary pulses these are called glitches, and momentarily the output goes from 0 to 1 back to 0 or 1 to 0 back to 1 these are called glitches which are undesirable but I want only flat waveform 0 1 0 1 like that. Any waveform to be of use should last for at least for one clock period.

If you want to use it, if you want to use any output I need to have it at least for one clock period so that it can activate whatever mechanism or output devices which needs to activate. This is a narrow pulse very small portion fraction of the clock period which again is a narrow pulse. These two narrow pulses or glitches would have been eliminated in the master slave flip-flops. So, master slave flip-flop serves the purpose of frequent changes in the input affecting the master and not reflected in the output of the slave. It only makes the change once in a clock period.

Originally we thought of master slave for avoiding racing. To avoid racing we thought of that because that is the same as this and in effect this is the same as this because there are two frequent changes within a clock period. Within the clock period being on there are too many changes in the output which we did not like so we avoided many changes in the output when the clock was high, and that is also the same thing here. So that design which was originally made to eliminate the changes in the output of the clock output of the flip-flop too frequently also helps us to ignore glitches. This is one operation, one option to have a master slave operation is to make sure that the output do not change frequently but they only change once in the clock period where any racing condition can be avoided. The other possibility is to use edge triggered flip-flop which is mentioned by me in the last lecture towards the end. I can have a mechanism or an extra hardware within my flip-flop which will make sure that once output transition takes place it will not happen again within the same clock period but it will only happen in the next clock period.

You can let the output change during the clock changing from 0 to 1 which is called a positive edge transition or when the clock changes from 1 to 0 this is called negative edge transition. So one is called positive triggered flip-flop and the other is called negative edge triggered flip-flop. In a positive edge triggered flip-flop all you have to do is to look at the input at the time when the clock goes from 0 to 1 and whatever will be the output based on that input will be the output for the entire clock period of that clock period. You don’t have to worry about what happened in between until the next clock edge comes. It is the same thing in the case of a negative edge triggered flip-flop.

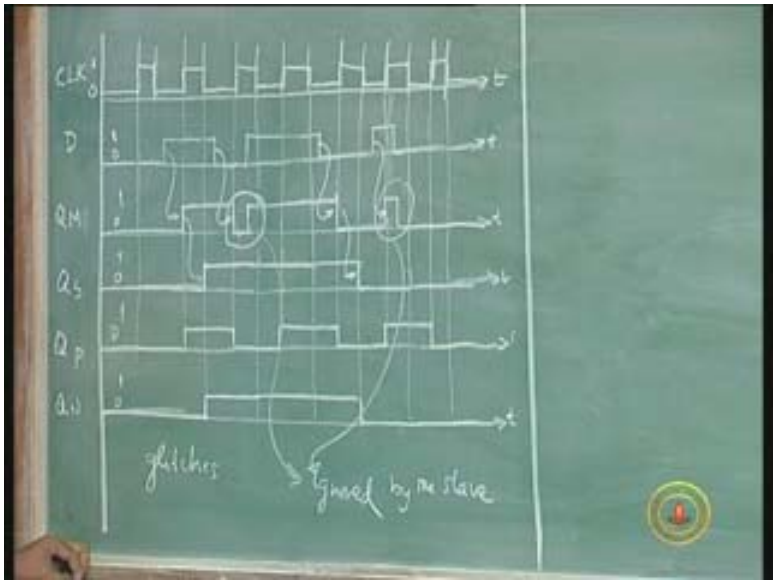
Once there is a change the flip-flop looks at the input at the time when the clock transits from 1 to 0 and that will decide the output and that output remain same until the next negative edge comes. In between you don’t about worry about the changes that took

place in the output. Therefore from the same clock period I have now drawn the positive edge triggered output. Had this been a positive edge triggered flip-flop instead of a master slave flip-flop, instead of a master slave flip-flop had it been a positive edge triggered flip-flop how will the output change? All you have do is to look at the positive edge, this is called positive edge (Refer Slide Time: 21:29) and find out the output at the positive edge and that input will be the output for the entire clock period. so at this point it is 0 so for one clock period it is 0, at this point it is 1 so for the next clock period it is 1, at this point it is 0 so this pulse is not going to be recognized so it has become 0, at this point it is 1 it is 1 here and at this point it is 0 so 0 and at this point it is 1 so 1 even though it doesn't fall back to 0 it continues to be 1. So once it becomes 1 it continues to be 1 until the next positive edge comes. At each positive edge you look at the data and decide the output and that output remains until the next clock edge comes and again at the next clock edge look at the data and decide the output.

Now it is not the same as this but now at least here the output remains stationary for the entire clock period and in between there is no output change. In the positive edge trigger if you apply this D flip-flop it happens only like this (Refer Slide Time: 22:55). It changes for one clock period, one clock period, one clock period but change in the input of the clock even though it occurred in the middle of the D pulse middle of the clock it is recognized here. So that way it is different from the behavior of this. Even in positive edge triggered flip-flop we have to be careful not to change the input....., even though output will change now only you do not want to have this change in the input when the clock is high, you should be careful not to give input changes or not to change inputs when clock is high because it will result in affecting the output.

Negative edge triggered flip-flop the same logic applies except that we now look at the negative edges and positive edges. What is a negative edge?
The negative edge is one which goes from 1 to 0 so this is a negative edge, this is the negative edge, negative edge, negative edge, negative edge and so on. In each of the negative edges you look at the input and decide the output. here it is 0 so the output is 0 for one clock period, (Refer Slide Time: 24:00) at this point it is 1 so 1 for one clock period and at this point it is 1 so 1 for one more clock period, at this point it is 1 so 1 for one more clock period, this point is 0 so 0 for one clock period, this point is 0 so 0 for one more clock period, this is 0 so 0 for one more clock period that is all.

(Refer Slide Time: 24:24)



Incuriously this is same as this. so a master slave flip-flop works like a negative edge triggered flip-flop so I have a choice now, I can have a master slave flip-flop, I can have a positive edge triggered flip-flop or I can have a negative edge triggered flip-flop, the waveforms are given. A waveform of the negative edge triggered flip-flop is identical to the slave waveform of a master slave flip-flop. If you want the waveform of a master slave flip-flop identical to the positive edge triggered I can do it by giving the negative of the clock to the master and regular clock to the slave.

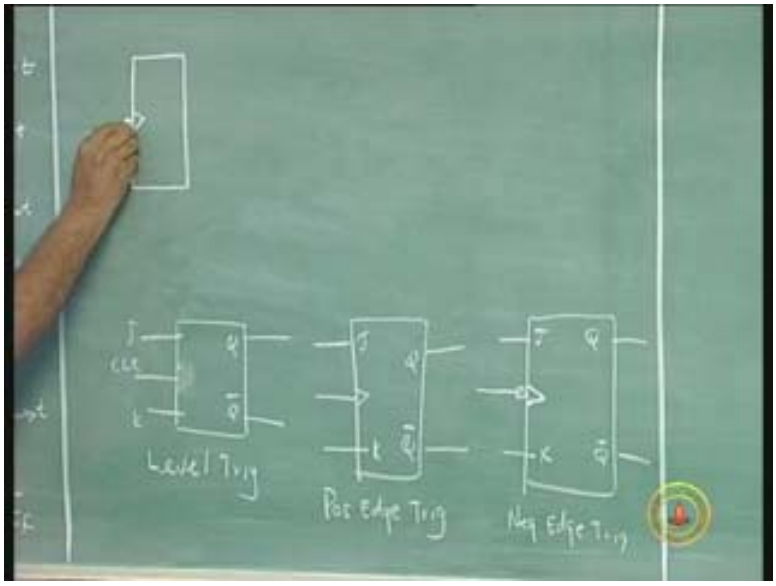
I now have a clock which I fed to the master, I inverted it and fed to the slave but instead I take the clock invert it and give to the master and then invert it again and give to the slave. That means the original clock is given to slave and inverted clock is given to the master then the behavior would be opposite. Therefore a master slave flip-flop will behave like a positive edge triggered flip-flop, so I can have any combination of the outputs or any combination of the behavior. Any behavior that you want can be achieved.

So having talked about latch being a storage element a bit storage 1 or 0 and making sure that we need some timing or a control pulse to decide when the data will be stored we introduce the clock and we call the clocked latch a flip-flop and then we get SR flip-flop and we wanted a single input so we made D flip-flop and said we wanted to remove S is equal to 1 R is equal to 1 condition made it a JK flip-flop and we worried about racing and removed the racing by introducing master slave concept or a edge triggered concept and also had a T flip-flop where only toggling will happen and I also explained about the behavior in terms of waveforms and everything namely circuitry, symbols, characteristic tables and so on.

The next question is what do we do with all these flip-flops?

D flip-flop as I said is simply used for data storage. So we have a data and store a bit of information you give it to D and clock it and it will store it, it is available in the output and I want to change it you put another value of D and then clock it again.

(Refer Slide Time 30:20)



JK I said in addition to data storage of course data storage of JK will be put to two inputs 0 1 for 0 storage and 1 0 for one storage in addition you can also use it as a toggling mode.

So first let us look at that feature because data storage is a simple thing, I put a data clock it so it has a storage data. SR and JK I need to do the same thing except that data has to be 0 1 1 0 combination. So the one in which the feature is sort of different is the toggling feature. Let us see whether we can use a toggling feature. So let me draw a JK flip-flop, here after I will only use blocks, use a block and it is all inside, the circuitry. So let us say it is an edge triggered flip-flop or a master slave flip-flop it doesn't matter, let me call it a negative edge triggered flip-flop or a master slave flip-flop because both are of the same behavior.

By the way when you write the symbol let us say JK and put clock here let us assume it is a level triggered clock level triggered flip-flop. If you think it is edge triggered you have put an arrow like this a knife edge, it is a point, meets only one point in time at one instant time this is the symbol for edge triggering. If you put like this it is a positive edge triggering (Refer Slide Time: 29:10) so this is level triggering you put like this it is edge triggering positive edge triggering and you put like this an arrow with a bubble where bubble is always inversion wherever you see bubble that is why in inverter you have a bubble, NAND gate is a bubble, NOR gate is a bubble so bubble stands for inversion it's a negative edge triggered flip-flop. These are symbols used. Of course you don't have a separate symbol for master slave you put like this for master slave also because master

slave behaves like a negative edge triggered flip-flop. So let me assume it is a master slave flip-flop or a negative edge triggered flip-flop and I put J is equal to 1 K is equal to 1 when J is equal to 1 K is equal to 1 what happens to the output? The output toggles, the present output is the complement of the past output. The output during this present clock cycle is the complement of the output during the previous clock cycle or the output of the next clock cycle is the complement of the output of the present clock cycle, this mode is called toggling mode.

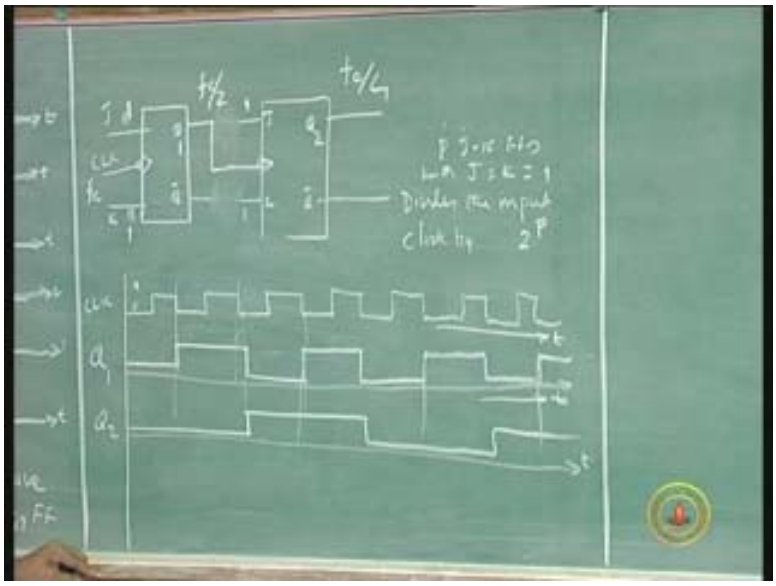
I am going to draw a clock waveform if it is a negative edge triggered or a master slave flip-flop it doesn't matter. So I am giving this clock here to this input I am making J is equal to 1 and K is equal to 1 what is the 1? 1 is 5V or high level voltage, it can be 5, 3.5 or whatever voltage you choose in your technology. Now let us assume initially Q was 0 and at the negative edge the output will change, it will change to the complement of what it was. So it was 0 to start with, this is time access (Refer Slide Time: 32:40) and at this point in time it will become 1 1 input and the output changes to its complement and I am giving this clock input and the clock negative edge because of the negative edge, the original Q was 0 and at this point the Q becomes 1 and at this point it becomes 0 and this point it becomes 1.

Now can you not see something in these two? As I mentioned earlier the period of a clock has the point from which you start from one point in the waveform and reach an identical point in the next cycle so here to here is the period, from here to here is the period, here to here is the period and in this case this is the period. For every two clock periods here this clock period is 1 so we have one period for every two clock period of the clock or it is positive for one whole clock period, but earlier the clock period was half positive half negative that is half 1 and half 0 now for the whole clock period it is 1, and for next whole period it is 0, and for next whole period it is 1 and so forth. So I am having an increase in the clock period by a factor of 2 or reduction in the frequency by factor of 2. So if you want to divide a clock frequency by a factor of 2 I can put it through the JK flip-flop and take the output of the Q. So basically a JK flip-flop is with J is equal to 1 K is equal to 1 or J is equal to 1 K is equal to 1 you want a special name for this then it is called toggled flip-flop. So instead of showing it as JK flip-flop you can say T is equal to 1 so when J is equal to 1 K is equal to 1 it is called T mode toggle mode. A toggle flip-flop is divided by two flip-flops.

Now if I take another flip-flop whose output is again J is equal to 1 K is equal to 1 and this clock period I am going to not give this clock but going to give this clock divide by two clock. If this clock is f_c and this should be f_c divided by 2. That f_c by 2 would be fed to the clock by the second flip-flop so the output would be f_c by 4 so we will call this Q_1, call this Q_2, Q_2 will remain 0, here it will go 1, 0, this is my Q_2. So every clock pulse period become half a period here so one clock period of this is equal to two clock period of this, now every clock period becomes half clock period here and I can go on dividing this further and further. So if I have a two flip-flops I get the output of f_c by 4 circuit so if I have n flip-flops let us say P JK flip-flops with J is equal to 1 K is equal to 1 then divide the input clock by 2 power P. so here you have straight away got a use for the flip-flop. I want to divide a frequency a frequency divider, I have a large frequency I want to divide

it for many reasons. Can you think of places where frequency has to be divided just it's a very simple general knowledge question, just come up with some answers. Where will you need a frequency divider? watches is a good example, minutes has to be divided into seconds and seconds has to be divided into minutes and minutes into hours and hours into days and days has to be divided into weeks and so forth.

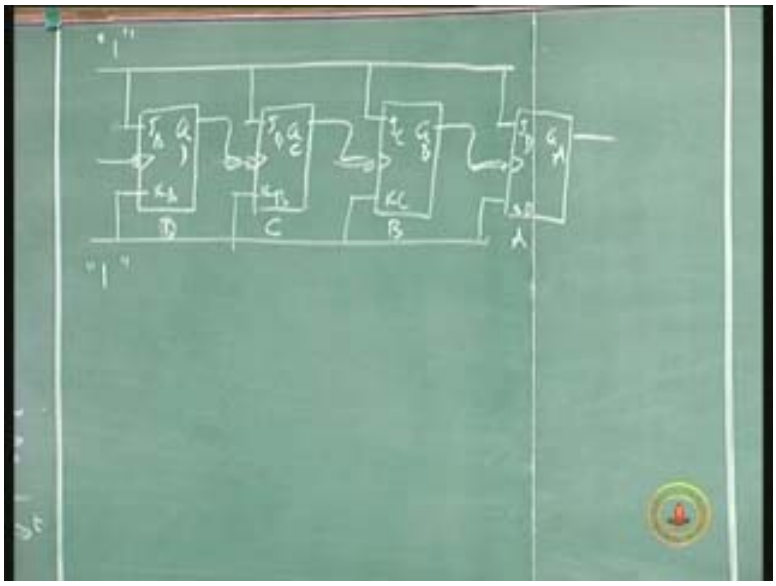
(Refer Slide Time: 39:40)



So one simple use of this flip-flop at least in the toggle mode is you keep on dividing the input clock original clock by a fraction of two every stage where p stages get divided by a factor of 2 power p. Now I don't have a space to draw this waveform. supposing I have four JK flip-flops or negative edge triggered flip-flops JK all Js tied to 1, all Ks tied to 1 this is Q, Q, Q, Q I will call this Q_A Q_B Q_C Q_D so this will be J_a K_a K_b J_b J_c K_c J_d Q_d (Refer Slide Time: 41:05) so this flip-flop is D flip-flop, C, B, and this is A. There are four flip-flops A B C D all negative edge triggered JK flip-flops and connecting the outputs like this.

Let us look at only the 0s. When the first clock becomes 0 the output of this is 0 0 0 0 that is four 0s then this becomes 1 and this continues to be this (Refer Slide Time: 41:49). After this this becomes 1 0 0 0 when this 1 becomes 0 so the first flip-flop immediately the Q_D flip-flop for every clock period it goes from 0 to 1 1 to 0, 0 1, 0 1, 0 1, 0 1 like that. Q_C goes like 0 0, 1 1, 0 0, 1 1, 0 0, 1 1, 0 0, 1 1 because every two clock pulses of the Q_D will only make this one change. Now this will be again repeated and this will remain 0 for four clocks then become 1 and remain for more four clock pulses, this will remain for eight clock pulses 0 and another eight clock pulses 1.

(Refer Slide Time 42:20)



So if you want to draw instead of drawing waveform of course since I don't have space to draw waveform for sixteen clock cycles I will do it in binary. So supposing this clock is Q_A Q_B Q_C Q_D this is 0 1 0 1 0 1 0 1 0 1 0 1 (Refer Slide Time: 43:20) and this will be 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1……

The value of QD during clock period let us call this clock period this is clock 0, clock 0 1 (Refer Slide Time: 44:10) we will all start with binary 0 1 2 3. So during clock period 0 all the outputs are 0, during clock period 1, during clock period 2 so during clock period 15 what it tells you? I have a circuitry to count the number of clock pulses. All I have to do is to find how many clock cycles have elapsed from 0. All I have to do is to look at the waveforms of the clock A B C D, if Q_A is 1, Q_B is 0, Q_C is 1, Q_D is 0 I know it is 1 0 1 0 which is the tenth clock pulse counting 0 as 1, starting from 0 actually it is the 11th clock pulse but we have to always number from 0 to 15 because there are only four bits. So this also can be used as a counter. they divide by two circuits, the divide by 2 power n circuit can be used as a 2 power n counter so this is 2 power 4 counter, using P flip-flops we can have 2 power P counter and when you say 2 power P counter we count from 0 to 2 power P minus 1. The 2 power P is 1 to 2 power P but we have to start with 0.

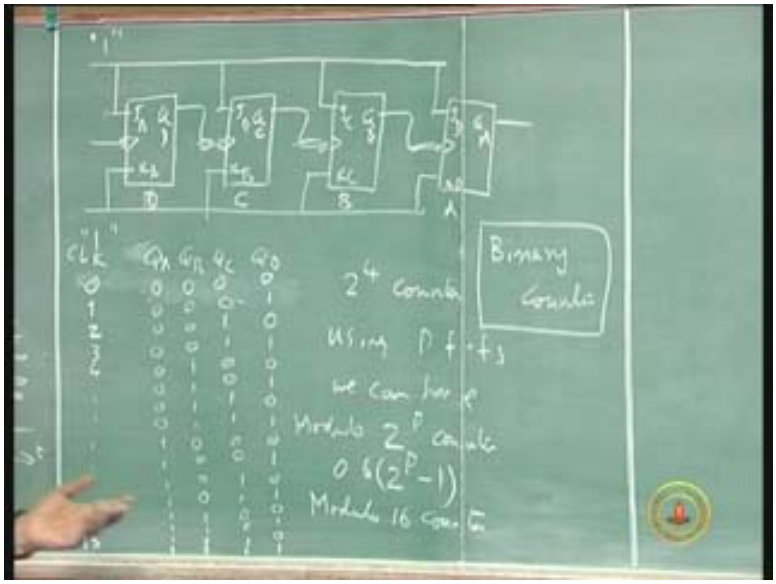
So looking at the waveforms of the four outputs A B C D you know that it goes from 0 1 0 1 0 1 0 1…… Any combination of outputs you can look at and see that it is the fifteenth clock cycle, this is thirteenth clock cycle that means including this starting from 0 fourteen clock cycles, this is the thirteenth clock cycle if 0 is counted. If we start counting from 0, start the count from 1 and after that what will happen is after fifteenth it will become 0 0 0 0 so we cannot count beyond fifteen, with four flip-flops I can only have a 2 power 4 counter. If I try to give the sixteenth clock pulse what will happen is this will become 0 because this is 0 1 0 1 0 1 0 1 so this will be 0, 0 0 1 1 0 0 1 1 so 0 0 1 1 so next will be 0 (Refer Slide Time: 46:45), four 0s, four 1s, four 0s, four 1s, so this

will be 0 and here it is eight 0s eight 1s so this will be 0s. So sixteenth clock pulse will revert it back to 0 0 0 0 again I start counting so it is called a Modulo Counter Modulo 16 Counter that means after 16 it is 0 or modulo 2 power n counter, modulo 2 power p counter. So if we have P flip-flops I can count any pulse starting from 0 to 2 power P minus 1 and if the next pulse comes 2 power Pth pulse comes all of the flip-flops will become 0 0 0 0 so I will have to start all over again. We have already seen two uses of these flip-flops. One is that you divide this 2 factor clock division like somebody said in watches you can use it and the other thing is use at counter.

Hence, suppose I want to count the number of people in this room all I have to do is to put this counter at the door and the clock will be the one that would be generated by the person coming in as I said the clock need not be at the same period of duration or same duty cycle. So any time a person comes in the room there is a momentary going from 0 to 1 back to 0 that is your clock and I use this into my counter sequence so at any time I look at the pattern of the counters the waveform of the counters based on the combination of 0s and 1s in this counter I can say how many people have come to this room so this is the counter, basically a set of flip-flops will be used as a counter.

It is a binary counter because we can only count binary values, you can make it decimal later on, we will see how to do that.

(Refer Slide Time: 49:00)



This is a two binary counter, we will also see some other uses of this. Now the problem is I can terminate only when the number is 1 1 1 1. Suppose I want to count up to 10 and go back to 0 in a decimal counter that is I would like to count from 0 to 9 and back to 0 how do I convert this into a counter which can count from 0 to 9 and back to 0 or any other arbitrary value, suppose I want to count up to 12 I want to count dozens, I want to pack something by dozens, I have a product to be packaged I want to count 12 so how do I do it? After 12 it becomes 0 0 0 0 is it possible? So how do you make it count non 2 power n

values and if you want to count up it is all right. But when people leave this room I want to know how many people have left this room so counting down also should be possible. So as people leave the room the count has to reduce by 1 so can we convert into an up counter and down counter, can we convert into a counter which is in count non 2 power n values etc we will see in the next lecture.