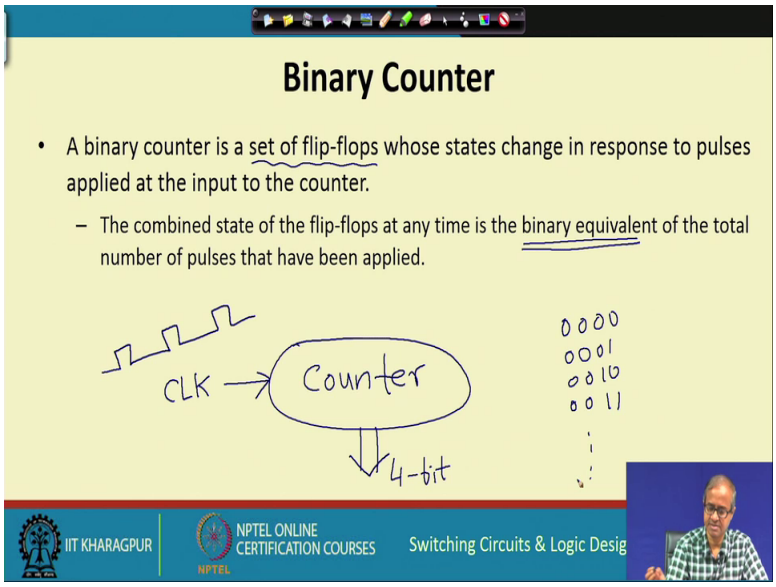
Switching Circuits and Logic Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 45
Design of Counters (Part – 1)

In this lecture we shall be discussing the Design of Counters, specifically a kind of counter which is called ripple counter, we shall be explaining the definition of ripple counter due course of time. So, the title of this lecture is design of counters part 1 ok. The kind of counters that we are considering here is the binary counter.

(Refer Slide Time: 00:47)

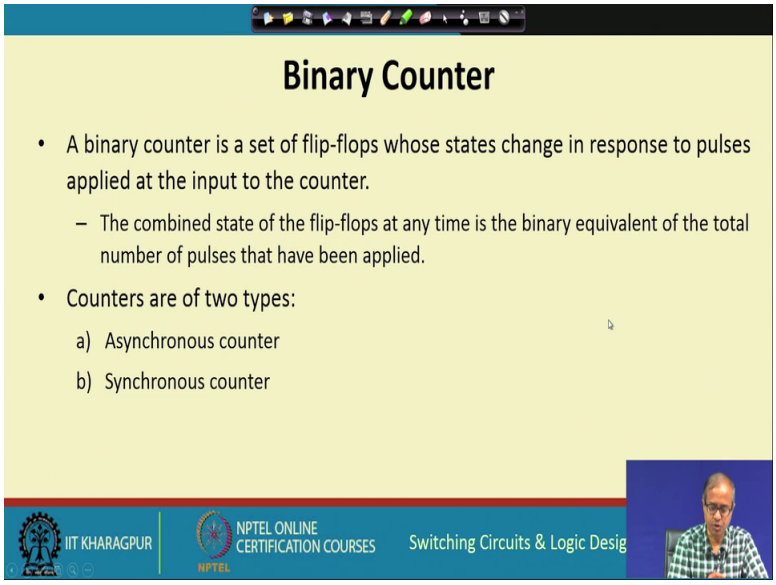


So, what do mean by binary counter? First thing is that the binary counter will be designed using a set of flip flops, whose states change in response to pulses applied at the input.

Like you say if I have a counter this is my counter, and I am applying some pulses in the input well we can call it clock, we can call it anything else some pulses are coming some pulses are coming in the input. And this counter is suppose to count how many pulses of come the combine state of the flip flop for example, if this is a 4-bit counter. So, in the output I will be having a 4-bit number coming out this I refer to as the combine state of the flip flops.

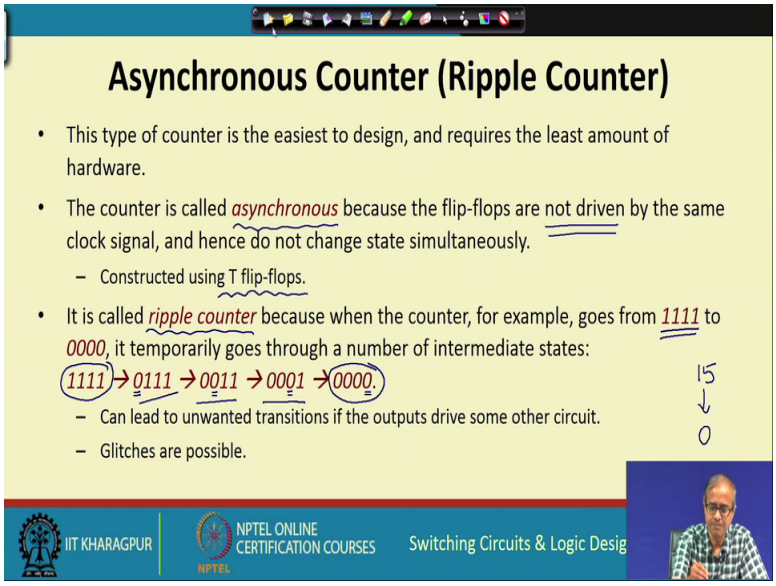
This will be the binary equivalent of the total number of pulses that have been applied; like if it is 4-bit it will start with 0, then 1, then 2 then 3 and so on this is what a binary counter essentially is right.

(Refer Slide Time: 02:10)



Talking about the types of counters broadly speaking counters can be either asynchronous or synchronous, let us try to see the differences between these two first.

(Refer Slide Time: 02:23)



First asynchronous counter; this kind of counter is sometimes also referred to as ripple counter now let us explain why we call it ripple counter and why is it asynchronous ok.

Because earlier if you recall we had the encountered this term ripple in the context of the design of adders, the ripple carry adder. In a ripple carry adder if you recall the carry is who are propagating from one full adder stage to the next, that is what we refer to as a rippling effect. Here also there is a similar rippling effect that will come in that we shall see how.

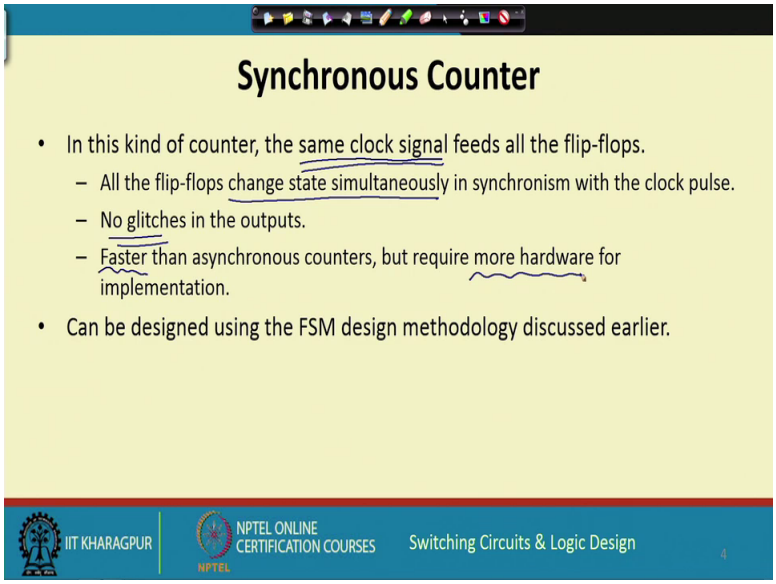
The first thing is that this kind of asynchronous or ripple counter are the easiest to design there this simplest, they required the minimum amount of hardware for implementation let us see how it we will this can be designed. Now let us try to explain these 2 terms asynchronous and ripple. So, we call this kind of counter asynchronous, because the flip flops it can be a four bit counter there will be flip flops. The flip flops are not driven by the same clock unlike a register. Now in a registered design if you recall all the flip flops where driven by the same clock signal, but in an synchronous counter we are not using the same clock to drive the flip flops, which means the flip flops are not changing the states at the same time there not synchronous. So, asynchronous means not synchronous and these kind of counters we shall see why are very conveniently designed using T flip flops.

Next let us see why they are called ripple counters. See here let us taken example that for a 4-bit counter the current state is 1 1 1 1, 1 1 1 1 means 15 right 1 1 1 1 means 15 now if another clock pulse comes like count is suppose to be 16, but in 4-bits you can only store a maximum of 15.

So, if another pulse comes it will again a reset to 0. So, a 15 you will again go back to 0. So, 1 1 1 1 to 0 0 0 0. Now in a ripple counter what happens is that this four flip flops will not change state simultaneously. So, what will happen? The first flip flop will change state first, then this second flip flop will change state, then the third will change state, then the last one will change state. So, this was the initial state, this was the final state and there are 3 intermediate or temporary states. So, when you apply a clock pulse or some signal in the input the flip flops will not change state simultaneously. There will be a rippling effect the first flip flop will change that will cause the second flip flop to will change, that will cause the third one to change which will cause the fourth one to change.

So, this is something similar to the rippling effect that we talked about this is why it is called a ripple counter. Now the problem in such ripple counter is that this as you can see that the transition from 1 1 1 1 to 0 0 0 0 is not smooth, there is some intermediate glitches or pulses that might come in between. So, if the output of the counter is driving some other circuit, this unwanted transition may cause some errors because of this glitches, but if such glitches are not really a problem, then you can use this kind of a counter because designing this kind of a counter is very simple right this is asynchronous counter.

(Refer Slide Time: 06:58)

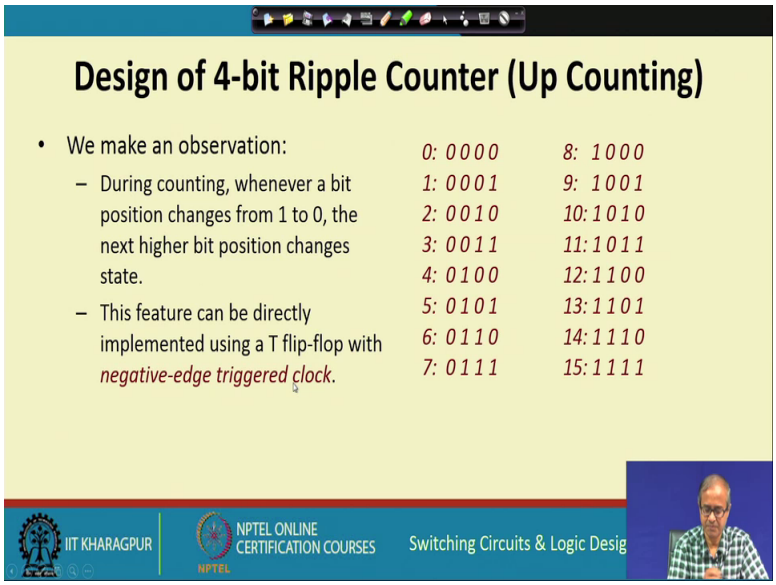


Now in contrast a synchronous counter is characterized by the fact that you have the same clock signal that is feeding all the flip flops; and because the same clock signal is feeding the flip flops whenever the active age of the clock comes, all the flip flips will change state simultaneously it was intermediate states are there was just now mentioned for ripple counter they will not come ok. So, all flip flops change state simultaneously which means no glitches in the outputs such counters are typically faster than asynchronous counter because there is no rippling effect, but the drawback is that it needs more hardware. Now we already know how to design a synchronous counter we have already seen it earlier.

When we talked about the formal design procedure for FSMs, we took an example of counter design also. So, when you want to design a synchronous counter that is the way

to go about it. You start with the state transition diagram or a state table, go through this steps and finally, arrive at the design using your chosen flip flop it can be T flip flop, SR or JK, D. So, as I said this kind of countess can be designed using a methodology which we have discussed earlier fine.

(Refer Slide Time: 08:35)



Now, here we are concerned about the design of ripple counters are asynchronous counter. Let us look at a 4-bit ripple counter which counts up means 0 1 2 3 4 upto 15 and then again back to 0 1 2 3. Now we make an interesting observation here which will help us in deciding how to design this kind of a counter the observation is as follows. During counting whenever bit position changes from 1 to 0, the next higher bit position will change state let us try to explain this with this sequence of counting.

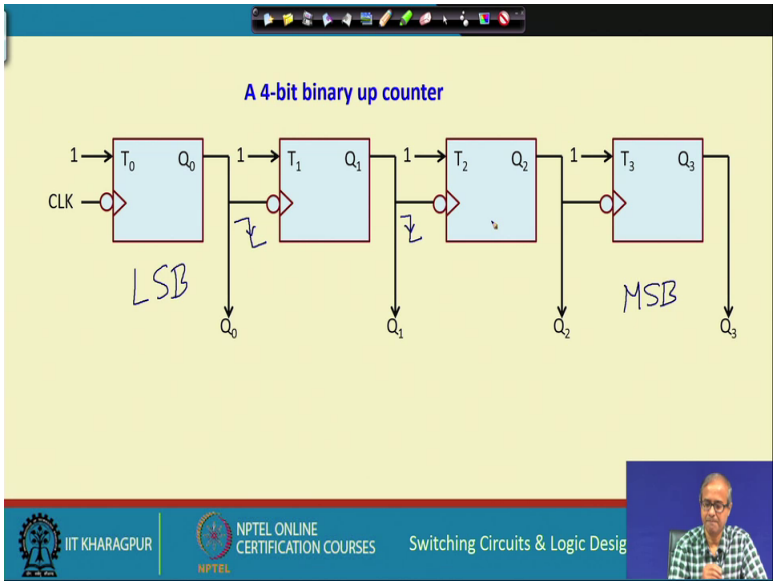
So, I am showing the decimal value 0 up to 15 and the corresponding binary values. Now I let me try to validate this whenever a bit position changes from 1 to 0, I look at the list significant thing see 1 to 0 there is there is a change here 1 to 0 there is another change here, 1 to 0 then again this 1 to 0 here 1 to 0 1 to 0 1 to 0.

Now, see whenever there is a 1 to 0 change the next higher bit also changes it was 0 it becomes 1 here it was the 1 it has become 0, it was 0 it has become 1 this is changing this is changing this is changing this is changing and so on right. So, whenever there is a change in one stage when they bit goes from 1 to 0 the next higher bit position changes state.

So, let us verify for the next position, in the next position 1 to 0 transition is coming here, next which is coming here, next which is coming here and the final one is coming here 1 to 0. You verify 1 to 0 so the next higher bit is change into 0 it is become 1. Then here this bit is changing it was 1 it has become 0 here, this bit is changing it was 0 it has become 1 and here also it was 1 it has become 0. So, this is true for all the bit positions and this is a very interesting characteristics of this counting process in binary.

Now, because of this we can have a very intuitive method of designing this kind of a counter now you think of a T flip flop. Suppose I have a T flip flop which is triggered by the negative edge of the clock. So, whenever there is a 1 to 0 transition in the clock the flip flop will change state. So, exactly the same behaviour is captured in this table right. So, whenever there is a change from 1 to 0, the next state is suppose to change. So, if the output of one stage I connect to the clock input of the next stage that will make my counter very simple. So, what you are saying is that this feature can be directly mapped into a T flip flop circuit with negative edge triggered clock, let us see how the counter will look like the counter will look like this.

(Refer Slide Time: 12:27)

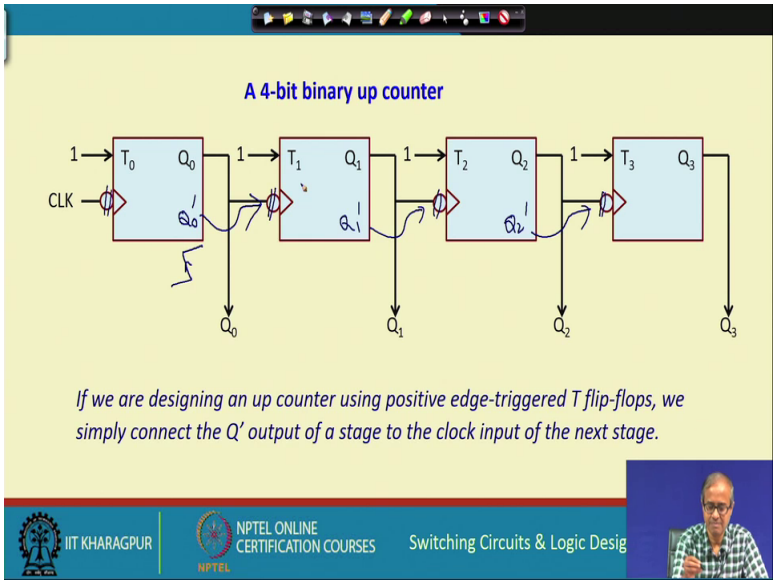


This is a 4-bit binary counter, I use 4 T flip flops, for the T inputs are permanently set to 1 which means whenever there is a clock the flip flop will toggle it will change state. So, the input clock is fed only to the first flip flop this is your least significant bit LSB and the rightmost one is your most significant bit. The output of this first flip flop is

connected to the clock of the second flip flop, output Q 1 is connected to the clock here and output Q 2 is connected to the clock here. So, that whenever there is a 1 to 0 transition here this flip flop will change state, whenever there is a 1 to 0 transition here this flip flop will change state and so on right this will continue to happen.

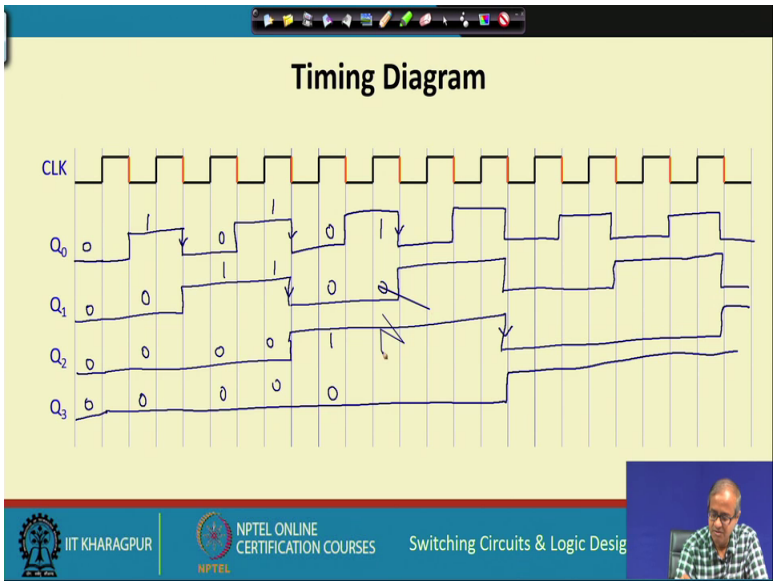
This is how was simple binary up counter can be designed, just connect a number of T flip flops in cascade which are triggered by the negative edge of the clock. Now one point to notice that suppose instead of negative edge triggered I have positive edge triggered T flip flop, then what do you have to do.

(Refer Slide Time: 13:50)



Which means these are not negative edge triggered these are positive edge triggered. So, what will do only changes that, I will be taking the Q 0 bar output and I will be connecting this here I will take the Q 1 bar output I will be connecting it here and so on Q 2 bar I will be connecting here. We simply connect the compliment output of the stage to the clock input of the next stage. Because if Q 0 change from 1 to 0 Q 0 bar will change from 0 to 1, and this is the positive edge triggered T flip flop this will also work right.

(Refer Slide Time: 14:38)

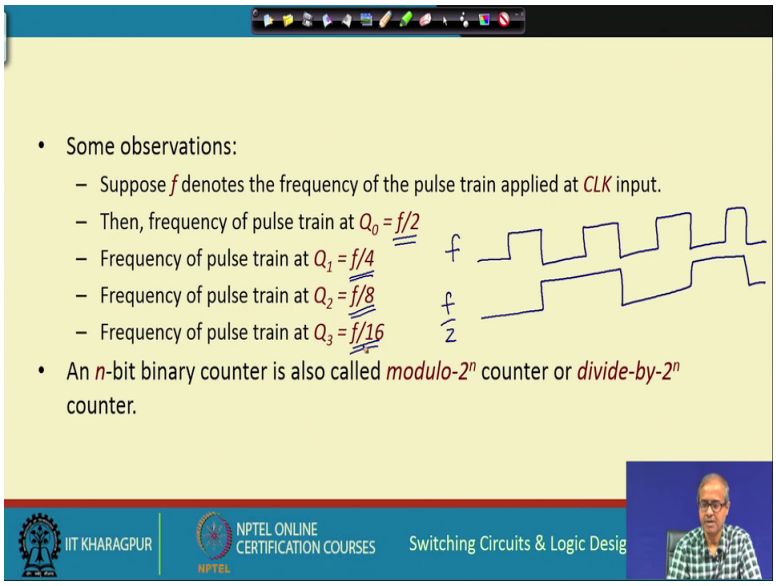


Now, let us work out the timing diagram for this example. Well here for the time being I am ignoring the delays. So, I am assuming that all changes are happening simultaneously. Let us assume initially Q 0 Q 1 Q 2 Q 3 there all in the 0 state 0 0 0 0 and clock is coming the activation of the clock is shown in red, the following edge. So, whenever there is a falling edge clock is connected to d 0. So, Q 0 will change state first let me draw Q 0. Again at the next falling edge Q 0 will again change state, again falling edge change state this will continue.

So, Q 0 will become like this, let us look at Q 1 this Q 0 output is fed to the d 1 is fed to the clock input of the next flip flop. So, whenever there is a falling edge in Q 0 this edges Q 1 will change state. So, Q 1 will change here again it will change here, again it will change here, again it will change here and again here similarly for Q 2, Q 2 will change here. So, whenever there is a falling edge in Q 1, next it will change here next it will change here and so on and finally, Q 3 we will change here.

So, whenever there is a falling edge. Just if you see it is 0 0 0 1 next it will be 1 0 0 0. So, it is exactly counting in binary, next it will be 0 1 0 0 then 1 1 0 0. So, look at it 0 0 0 0 is 0, 0 0 0 1 is 1, 0 0 1 0 2 3. So, like this it will go on, this will be 4, 0 1 0 0 this will be 5 right. So, in this way this will go on fine.

(Refer Slide Time: 17:23)

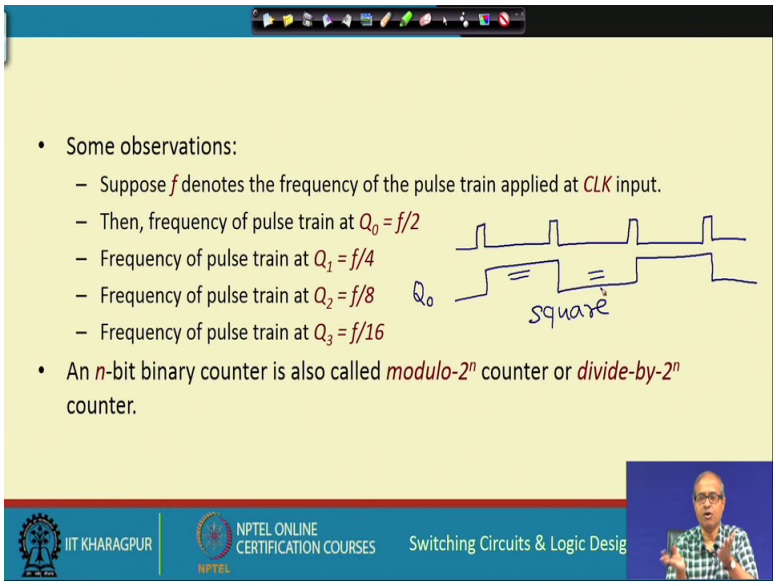


Now, there is one interesting observation, from the truth table or actually from the timing diagram that you have shown this if you recall the input clockwise coming, the output Q 0 was changing at the falling edge of the clock; that means, it was something like this the clock was like this coming and Q 0 was changing at the falling edge of the clock.

So, one thing you can say if the frequency of the input signal was f, the frequency of the Q 0 output will be f by 2 because for every 2 pulses of f you are getting 1 pulse of Q 0. So, at Q 0 the frequency will be getting divided by 2 stimulate Q 1 with respect to Q 0 again divided by 2. So, it will be f by 4 at Q 2 it will be f by 8 at Q 3 it will be f by 16. So, binary counter can be very nicely used for dividing the input frequency by some power of 2 this is one interesting characteristics.

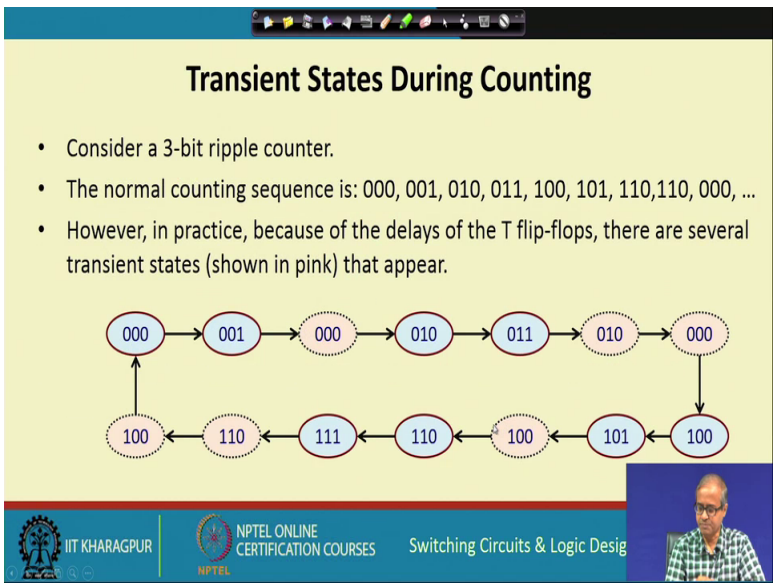
Now, another point also to note suppose by input clock is like this means it is not square it is something like this very nano pulses are coming in.

(Refer Slide Time: 18:59)



But if you look at the Q 0 output at every falling edge, it will be changing state. So, whatever you get it is a perfectly square wave, the on period and the off period will be exactly equal. So, from the output of the counter the f, the signals that you get, there will be perfect square waves on period and off periods are equal. So, whenever you require such a kind of a signal, you can use a flip flop or a counter to generate such square waves right these are 2 observations. So, an n bit binary counter in general this was a 4 bit counter you divided by 16. So, for n bit counter you can divided by 2 to the power n and such a counter is called the modulo 2 to the power n counter.

(Refer Slide Time: 19:52)

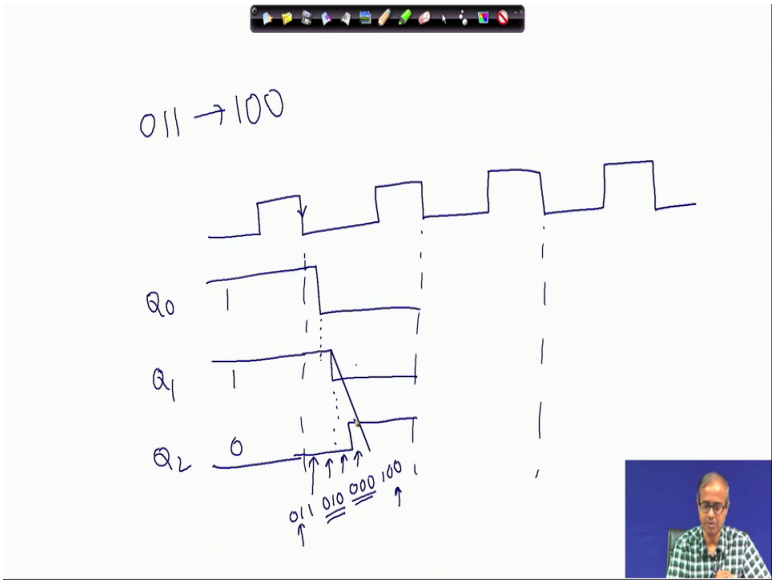


Now, one observation we have not discussed yet, I mentioned in the beginning that whenever a ripple counter is counting there will be some intermediate transient states because all the flip flops are not changing state simultaneously there will be some delay. So, here I am showing the real state transition diagram of a ripple counter, where the blue states are the permanent states and the pink marks states are the temporary states. Like whenever a single bit is changing like from 0 0 0 to 0 0 1 it is going directly, but 0 0 1 to 0 1 0 there are 2 bits which are changing.

So, here the first bit will be changing first then the second bit will be changing. 0 1 1 to 1 0 0 all the 3 bits are changing. So, first the LSB will change next LSB will change, then the most significant bit will change. Similarly 1 0 1 to 1 1 0, similarly 1 1 1 to 0 0 0 first bit will change, second bit will change third bit will change.

Now, let us see how it or may or may why it works, let us state illustrate it for this state 0 1 1 to 1 0 0 right.

(Refer Slide Time: 21:25)



Let us look at this 0 1 1 to 1 0 0 state. So, let me just show you the timing diagram to explain why this happens. This is the falling edge I just I am showing 1 clock only the first this is these are the 3 outputs Q 0 Q 1 and Q 2.

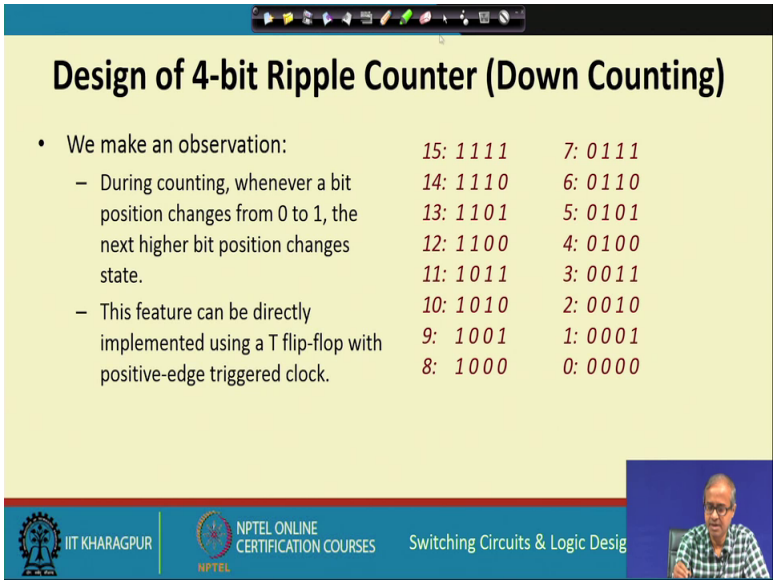
This state was 0 1 1. So, Q 2 was 0, Q 1 was 1, Q 0 was 1 0 1 1 now an active edge of the clock has come. Now what will happen you see each flip flop will have a delay. So, I am

showing the timing diagram with the delay here, because earlier I have ignore the delay in this timing diagram that had drawn.

So, Q 0 will change after a short delay like this, and this Q 0 will be fed to the T input to the into the clock input of the next flip flop. So, Q 1 will change now after again a little delay sorry. So, this will remain like this and Q 2 will change again after some delay. So, you see the intermediate states out here the state was 0, it was 0 1 1, 0 1 1 was initial of course, it was 0 1 1.

Then look at the state after this here, it was 0 1 and 0 it become 0 1 0 then look at the state here, it became 0 1 1 to 0 1 0 then it become 0 0 0. So, and finally, out here it became 1 0 0. So, these 0 1 0 and 0 0 0 these are intermediate states that are being generated 0 1 1 was the initial state. 1 0 0 0 was the final state and this intermediate states have been generated and these are happening because of the delays of the flip flops right. So, this is what you mean by there are transient states during the counting.

(Refer Slide Time: 24:25)

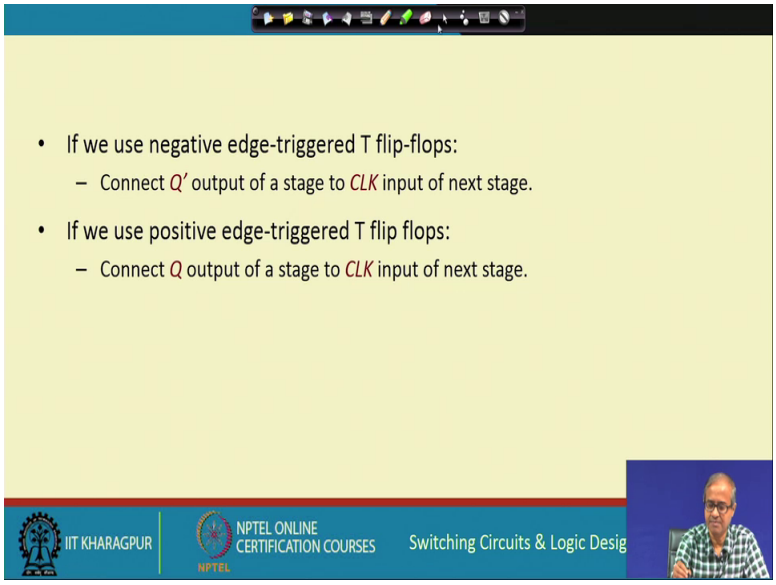


Now, if you want to design a down counter, the observation is very similar, but the direction of the changes are just reverse. So, whenever a bit position changes from 0 to 1, then the next higher position change the state if a accounting down. Let us say sometimes we need a counter to count down 5 4 3 2 1 like that, this is called a down counter. Suppose we want to design and down counter. So, it will start from 15 for a 4 bit counter, 14 13 down to 0 and again back to 15.

So, whenever there is a let us 1 to 0 transition I am showing some examples not to 1 to 0 0 to 1 here, 0 to 1 transition the next bit is changing it is a here also 0 to 1 translation next bit is changing. So, here also you see 0 to 1 transition 0 to 1 transition the next bit is changing it was 1 it was become 0.

So, just the polarity is reverse otherwise there is no basic change. So, to implement this you can use a T flip flop earlier we use negative edge triggered clock, but here you recall positive edge triggered clock. So, whenever there is a change from 0 to 1, then the flip flop should towel right this is the requirement.

(Refer Slide Time: 26:04)



So, to summarize, so if we use negative edge triggered flip flop, then we will have to connect Q bar output to the clock input of next stage, but if we use positive edge triggered flip flop, then you can directly connect the Q output to the clock input this will result in a down counter. So, with this way come to the end of this lecture.

Now, in the next lecture we shall be continuing with our discussion on counters, we shall be seeing some other issues with respect to counter design, and we shall be talking about general counter designs, not necessarily some counter which is counting up to some power of 2 and then going to 0, and this we should be discussing in the next lecture.

Thank you.