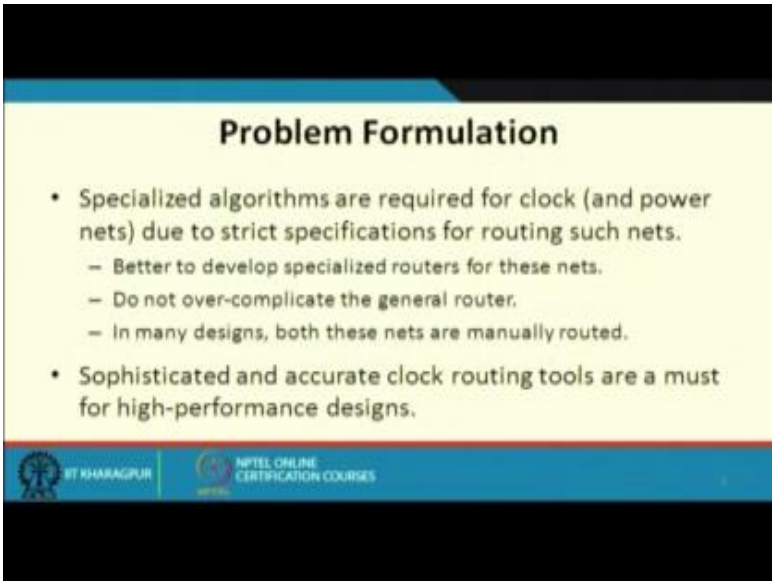
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 27
Clock Network Synthesis (Part 1)

Hello welcome back. So, if you recall we had discussed some clocking issues during the last week. So, where we talked about some clock related parameters like delay, skew, and when a clock feeds a flip flop sometimes like the hold time and the setup times. So, we saw that how this times can be incorporated in a design, in order to calculate the affective clock period or the clock frequency with which a circuit can be operated, without any problem taking into account these variabilities.

So, we continue with a discussion, today we shall be considering first the problem of the so called clock network synthesis. So, given a clock source how we distribute it actually to the terminal points where we require the clock signal to be connected to and finally, we shall be looking at some issues of the routing of the power net; that means, the v d d n ground. So, we start with clock network synthesis.

(Refer Slide Time: 01:42)



So, the problem formulation goes as follows, like you see for the clock and the power nets, there are some very specific requirements. See earlier the routing algorithms that we looked at they considered signal nets, two terminal or multi terminal nets. This signal

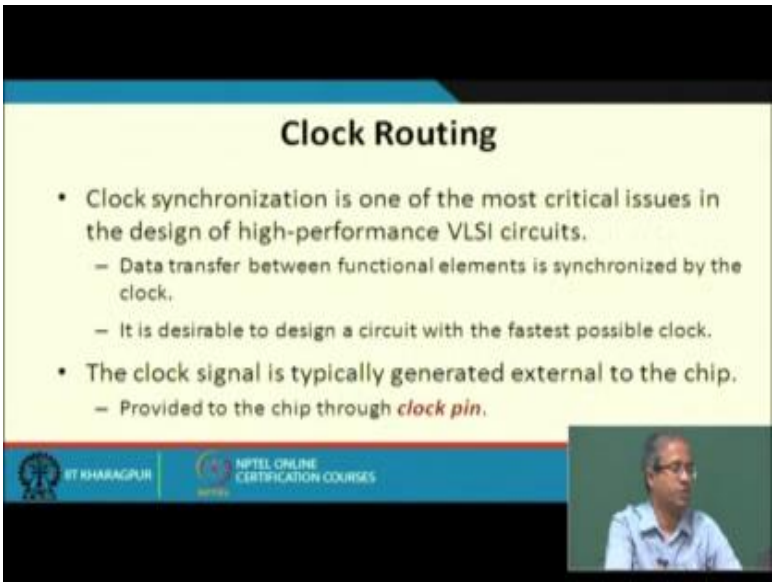
nets basically carried some digital signals 0 or 1 and one characteristic of this signal nets was that the dimension of this net; that means, the number of points this multi terminal nets were connecting were not too high, because you know a gate if it has a large number of fan outs, then due to the capacity loads the rise time and the fall times will go up and the performance will be degrading.

So, the issue for this kind of nets was different; our sole objective was to minimize the number of tracks for example, in channel routing and we did not consider the sizing of the wires, which means we assume that we can use this smallest possible width for the wires, because fan out is limited and the current flowing through these wires will not cross certain limit. So, that was our assumption.

But when we talk about clock and power it is not like that, the clock and the power you can say the nets they are typically very large in size, in the sense that they connect a large number of points in a chip. Say in a typical chip there can be thousands of points where the clock signals should be fed to, and again for every block we need the power supply and ground lines to be routed. So, these are examples of nets, where the number of terminal points can run into thousands or even more. So, we need some very different kinds of routing, well again for clock we need to take care of the skews and other issues jitter. So, it is looks quite complicated. So, we need a whole set of different strategies to handle and route these kinds of nets.

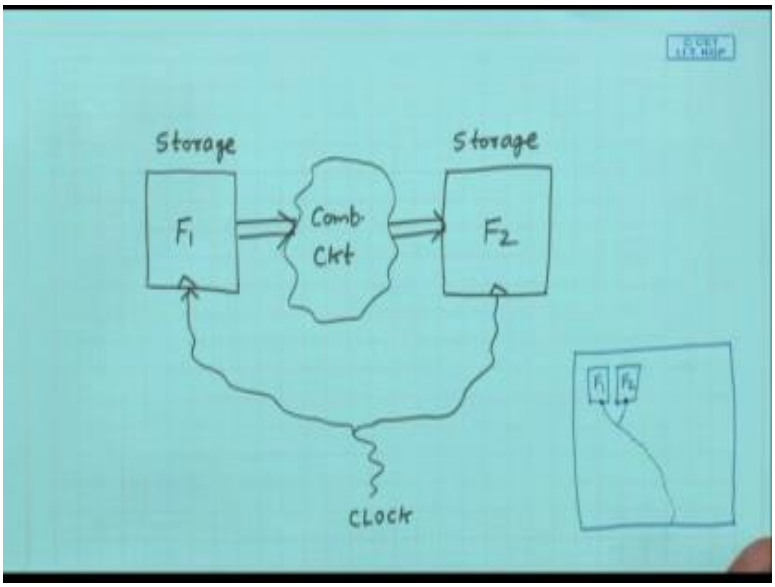
So, the motivation is as follows, because of their very specific requirements, so we want to develop or formulate specialized routing algorithms. So, we do not want that a single router should be there that can handle the normal signal nets and also the clock and power nets, because their requirements are very clearly different. And there are systems were at least part of these nets clock and power they are routed manually. So, for high performance design particularly for the clock net, we need some sophisticated analysis and very accurate routing tools, so as to control the skew and jitter parameters of the clock.

(Refer Slide Time: 05:06)



So, talking about clock routing, now clock synchronization is of course, one of the most important issues in digital systems, which or most parts are synchronous in nature means they are driven by a clock.

(Refer Slide Time: 05:31)



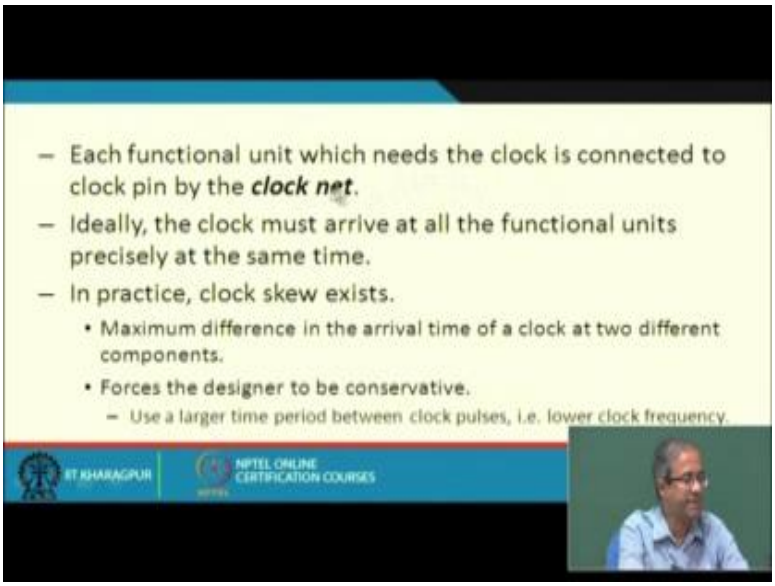
So, when I say that driven by a clock it means that you have some functional blocks; let us say F 1, there is another functional block F 2. So, let us consider only 2. So, there can be some combinational circuits between them, this is let us say these are the combinational circuits, some combination circuit. And this functional block F 1 and F 2

that we are talking about these are actually storage, which means these consists of some flip flops or latches. So, the output of this F 1 will go to the combinational logic combinational circuit, output of this will be coming here and there will be a clock source somewhere from where I need to connect a clock signal here as well as I need to connect a clock signal here.

Now, here in this case you try to understand the requirement this F 1 and F 2; so if inside a chip if you consider this as the layout of the chip, if this F 1 and F 2 are very close together, let us say F 1 is here and F 2 is very close together then wherever the clock source is the time taken for the clock to reach F 1 and F 2 will; obviously, be approximately equal, if we are able to keep this clock terminal points as close as possible, but in general this is not possible. This F 1 and F 2 can be located at (Refer Time: 07:17) arbitrary locations in this chip, which means the clock signal when it reaches this F 1 and F 2 the delays will be different which means there will be clock skew, there can also be clock jitter (Refer Time: 07:32) jitter we are not considering right now skew is the one which we are considering, jitter will come automatically along with it. So, the idea is that when we generate a clock from some point, it has to be sent or routed to the different points in a chip. So, it must be routed in such a way that the delays should be approximately equal this is one of the main requirements, right.

So, for general clock routing the consideration is that. So, all operations are typically synchronous with respect to a clock signal, and if we can minimize skew so we can use the fastest possible clock. Because if you recall earlier I mentioned that because of the skew and the other such delays, our clock time period increases which means our effective frequency decreases. So, we are trying to run a circuit as the fastest possible clock. So, we will try to minimize this skew this king of delays or uncertainties in the clock. Now some terminologies; so the clock signal which is fed to a chip, typically it is generated external to a chip for most modern system, and this particular point through which the clock signal enters the chip. So, we refer to as a clock pin through which the clock signal enters the chip, so we refer to as a clock pin.

(Refer Slide Time: 09:08)

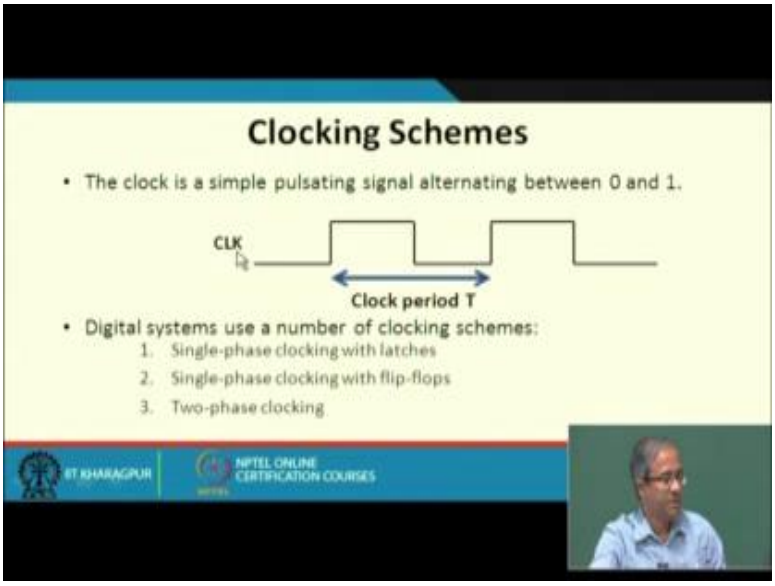


Now, from this clock pin we have to send out connections to every point where the clock signal is required, this is referred to as the clock net. So, clock net is that routing that wire from the clock pin which is coming at one end, I have to send the connections to all the points where clock signal is required. The whole net is called a clock net that clock net routing is our main point of discussion here. So, ideally speaking what we would like to have here is that the functional units that I have talked about. So, this F 1 and F 2 those two functions I have talked about, there can be many such this functional units can be distributed all throughout the chip, but we want ideally speaking that the clock signal must reach all of these units at the same time and if it is achieved then we can say that there will be no clock skew.

But unfortunately as you can understand we cannot have this ideal situation in practice so we will have to live with clock skew. Just to recall what is clock skew? Clock skew is the clock signal arrives at two different clock points and there will be a relative delay between them, either the first clock this first clock arrives the little later that the second clock or the vice versa. Because of the means unequal length of the wires that are used to connect the two pins and also variability in the parasitic resistances and capacitances drivers various parameters are there, the delays can be different and because of that you can have this clock skew right.

So, to handle clock skew what obvious thing one obvious approach can be to enforce some kind of a conservative design, which means I am playing safe. I am saying that let clock skew be there let me assume some maximum (Refer Time: 11:41) let us say 10 percent or 15 percent of the clock period will be kept for the skew. So, with that calculation I can lower the clock frequency or I can increase the time period this is the so called conservative design. But for high performance systems you do not want yourself to be so conservative; you would like to reduce this skew as much as possible so that you can increase the frequency. So, even 10 percent increases in frequency, we will lead to a very fantastic improve in performance, all right.

(Refer Slide Time: 12:17)

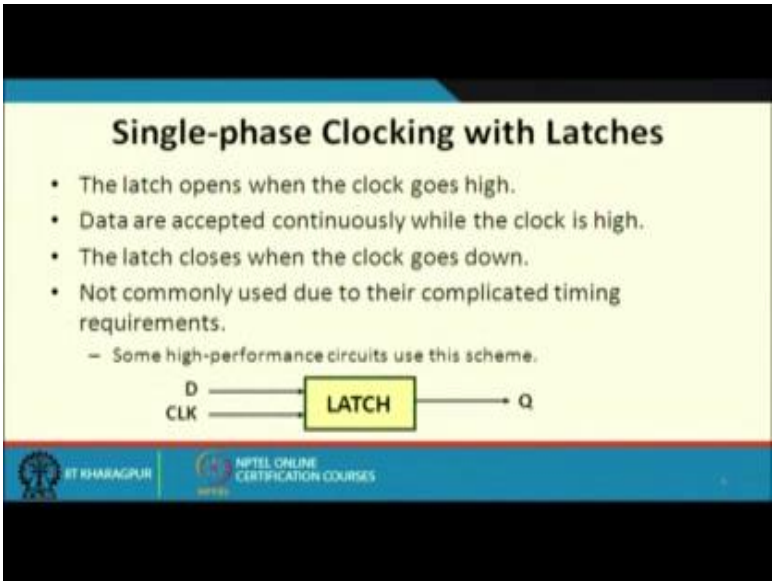


So, some very simple concepts in clocking, so as you know a clock signal looks like this, it is a periodic pulsating signals goes from 0 to 1 and 1 to 0. Depending on the system you are using so you can have various different types of clocking schemes, you can have single phase clocking or you can have multi phase clocking particularly two phase is quite popular and when you talk about clocking, you can have latches as the storage elements, you can have flip flops as the storage elements. So, what is a difference between a latch and a flip flop just to recall? A latch is a storage device where the data gets stored as long as the clock remains high, which means a latch is a level triggered device as long as the level of the clock signal is high, which is considered as the active state the latch is open and whatever changes occur in the input during this time will get stored accordingly.

But in a clock typically we implement a clock triggered a flip flop triggered by a clock with respect to some of the edges. So, we design the flip flop like a positive edge triggered or a negative edge triggered. So, exactly at the point where the edge occurs whatever is the input value that will be taken and the corresponding value will get stored. So, after the clock edge appears, if there are some changes in the inputs that will no longer be reflected. So, in case of flip flops we are talking about précised times, exactly at this time where the clock edge occurs whatever is the input I will take that. Well of course, the setup and hold time as I told about that has to be taken into account, but for the time being I am ignoring the setup and hold times I am saying that whenever the clock edge comes, exactly the input will get stored in the flip flop.

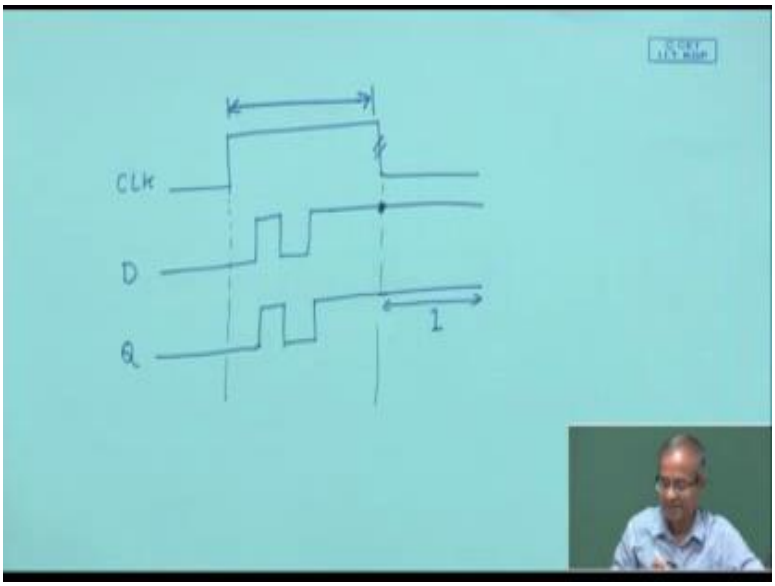
Well in contrast in a latch as I said as long as clock is high, the storage is open and the input I mean even if it changes during this time the final value will get stored finally, all right.

(Refer Slide Time: 14:47)



So, let us look at these one by one. So, single phase clocking with latches; single phase means here I am assuming that my schematic looks like this, it is a D type latch D input Q output, and I have a single clock. Single clock means I have a single phase clock and here as I said the latch is activated, when the clock goes high and as long as the clock remains high whatever is at D that will be accepted and will get stored in the latch. So, when the clock becomes 0 like what I am saying is that.

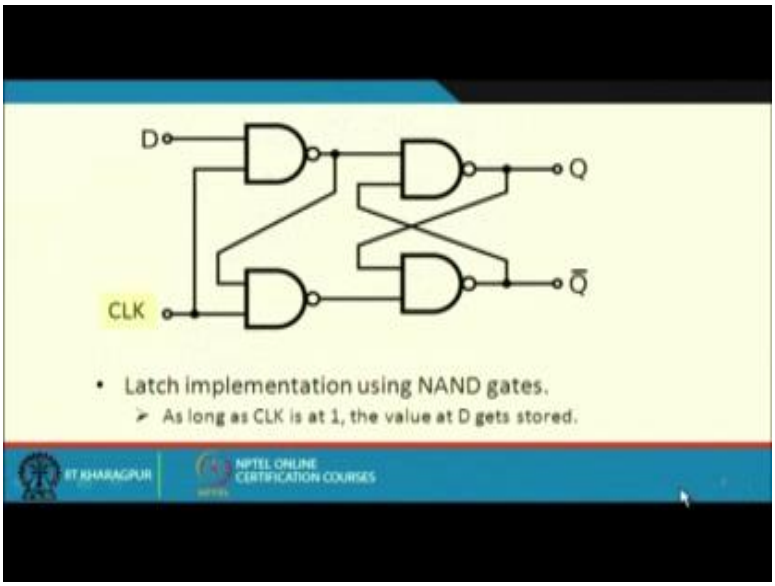
(Refer Slide Time: 15:30)



Suppose my clock signal is like this and let us say my D signal changes, because of some reason it changes and it remains stable like that. So, if you look at the Q output this is the time where clock is becoming high. So, whatever changes will be there in D there will be a small delay of course, So, this Q will be following d, but at the time the clock becomes 0 again your latch is closed. So, whatever was the last (Refer Time: 16:12) of D this one this one will remain. So, the final value of Q will be 1 right this is what is. So, the latch remains active or open during the entire duration the clock is high, right.

So, normally we do not want to use latch because of their complicated timing requirement like as I said as long as it is high, if the input changes the output also changes; that should not disturb any circuit which it is driving, but if you are careful enough then latch based design can give you better performance faster circuits we shall see some examples later. So, some high performance circuits typically use latches.

(Refer Slide Time: 17:03)

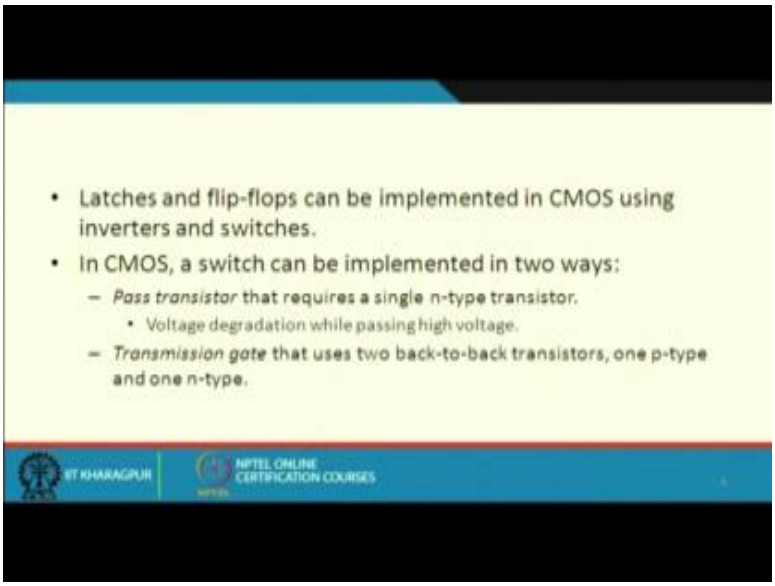


Now, a simple latch that uses gates is shown here. So, here we are showing a simple D type latch consists of 4 to input NAND gates. So, you can see here clearly if clock is high, let us say clock input is 1 then whatever this clock input is 1. So, whatever is applied at D so at the output here we will get D bar, this D bar will come here clock is 1 and we will get D here. So, if we apply D this line will be D, this line will be D bar and this is a cross coupled latch, if this is D bar then finally, the value of D will get stored here and if it is D the value of D bar will get stored here. So, the value of D will ultimately be stored and will be available at Q, and D bar will be available at Q bar.

But if clock is 0 let us say then both this inputs of the NAND gate are 0 and 0. So, both these outputs are 1 and 1 this is 1, this is also 1. So, whatever is here Q, Q and 1 will be Q bar and Q bar and 1 will be Q. So, whatever is stored will remain right. So, as long as clock is 1 the value of D will get stored, but as soon as clock becomes 0 whatever are stored that will be retained. So, this is how a simple latch works.

Fine; so now, let us see these latches how they can be implemented using MOS technology because in VLSI chips, we normally do not implement flip flops like this using NAND gates. So, there are more compacts of designing latches and storage cycles let us see.

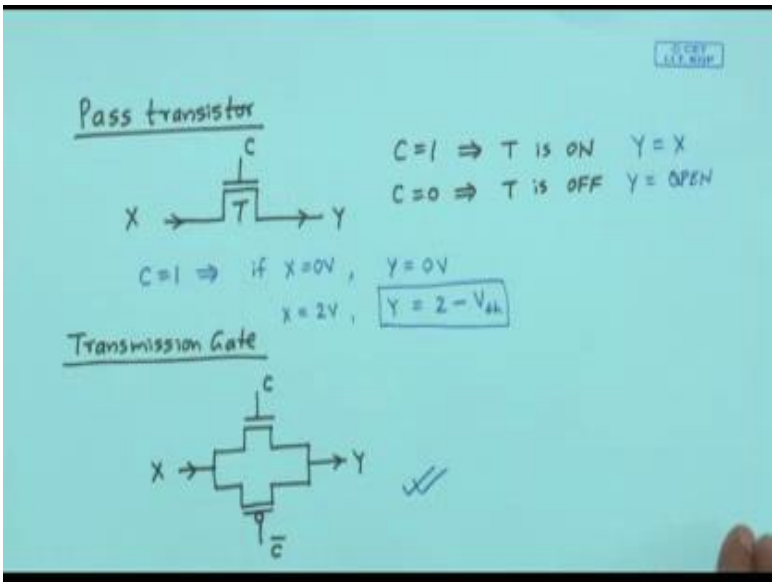
(Refer Slide Time: 19:06)



So, in CMOS technology, latches and also flip-flops, they are typically implemented by using CMOS inverters; that means node gates and some CMOS switches. So, what is a CMOS switch? So, a CMOS switch is just like an electrical switch, so if the switch is on it some applied voltage at the input will go to the output, if the switch is off that voltage will not go. So, it is either a on or off switch. So, either a voltage is transmitted or it is not transmitted, we will see how these switches look like.

So, the switches can be of 2 types: one is called a pass transistor, that requires a single transistor for it is implementation and you can have a transmission gate which consists of two transistors one p-type and one n-type. One drawback of the pass transistor is that it can incur some voltage degradation I shall show how let us take an example.

(Refer Slide Time: 20:19)



So, you first look at pass transistor. So, a pass transistor looks like this, I have a simple n type transistor. Suppose this is my input let us say X, this is my output let us say Y and this is my control input C. Now for an n type inverter if you recall. So, for n type MOS transistor for example, sorry for a n type transistor if C equal to 1, which means the transistor T is on which means it conducts, if C is 0 it means the transistor is off, it does not conducts.

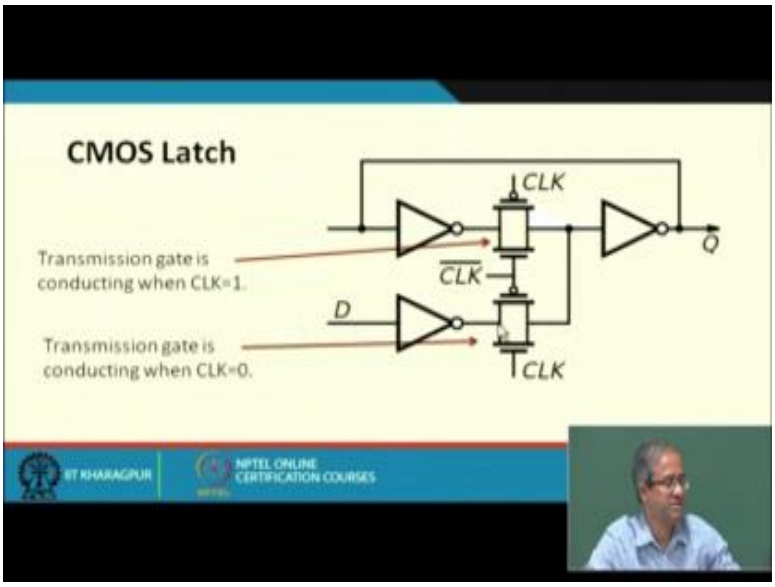
So, when C is 0 it means that I have a open switch. So, if C is 0 means I have Y equal to open let us write open. So, this X is not coming to Y, but if C is 1 then X is coming to Y, but one small problem here is that because of the property of the n type transistor, suppose I have control input 1, this switch is supposed to be on. So, for this case if X equal to 0 volts very low voltage, then Y will also be 0 volts without any voltage drop. But if X equals to high voltage let us say this transistor is working at 2 volts let us say I apply 2 volts then the output will be 2 minus V threshold. V threshold is a small voltage and there will be a voltage drop. So, you see that these kind of a switch when I am applying a high voltage, let us say that may indicate logic one at the input. So, there will be a voltage degradation in the output right this is a drawback.

So, to remove this drawback, I mean we can have something called as transmission gate. So, what is a transmission gate? You see just as usual I have a this kind of a n type transistor, and I use another back to back p type transistor symbolically shown like this,

and I connect these 2 ends, I connect these 2 ends. I apply X here and I take Y from here. So, if is I apply control C here I will be applying the reverse C bar here; you see just one property of these n type and p type transistors as I have said, this n type transistor can transmit 0 very well, but for logic one there is a voltage degradation, but for p type transistor it is reverse, it can transmit logic one very well, but for logic 0 there will be a degradation.

So, if I connect an n type and p type in parallel, then both 0 and one will be transmitted without any degradation that is the advantage. So, this transmission gate can transmit both logic 0 and logic 1 to the output, without any degradation; that is why typically for many design we do not use pass transistor, but rather we use a transmission gate. So, we show examples of latches that use both pass transistors and also transmission gates.

(Refer Slide Time: 24:16)

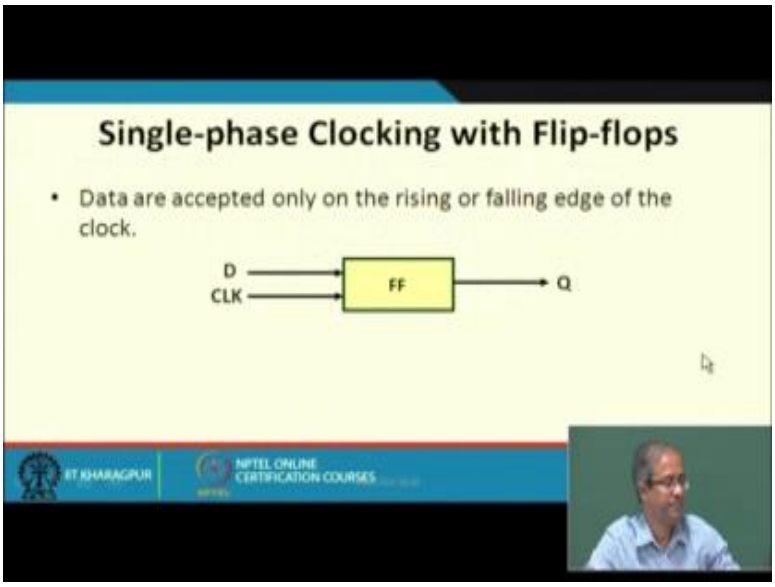


Here I show a simple CMOS latch, how it looks like? You see here there are two back to back invertors, a node gate a CMOS node gate a CMOS node gate I am showing it symbolically, and here I have a transmission gate, I have clock I mean this clock is applied to the p type transistor, clock bar is applied to the n type transistor. So, what does this mean? When clock will be 0 then this gate will be conducting, which means when clock is 0 I have a cross couple inverter with feedback that will constitute a stable storage system.

So, whenever we have two inverters connected in a feedback loop, that will constitute a storage system or a latch the data will be stored in definitely. So, here if clock is 0 we will be having such a system, but if clock is 1 you see what will happen? This will open, but this switch is closed here you connect a reverse Y clock bar is in; p type clock is in n type. So, if clock is 1 then this transistor is on, this gate is on and whatever I have applied to D that will get stored in Q. So, this D will go to Q and afterwards when clock goes back to 0, again this transistor will be off this will be on and whatever last value at Q that will remain right. So, this is how this CMOS latch works.

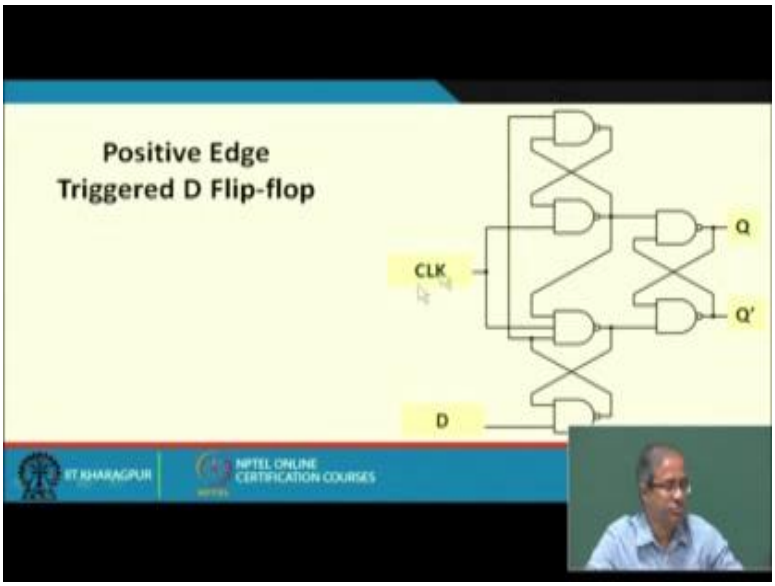
So, when clock equal to 1 this will actually be clock equal to 0 this will be conducting when clock equals to 0, these two are reversed I can read it as 0 read it as 1 and this will conduct if it is 1.

(Refer Slide Time: 26:15)



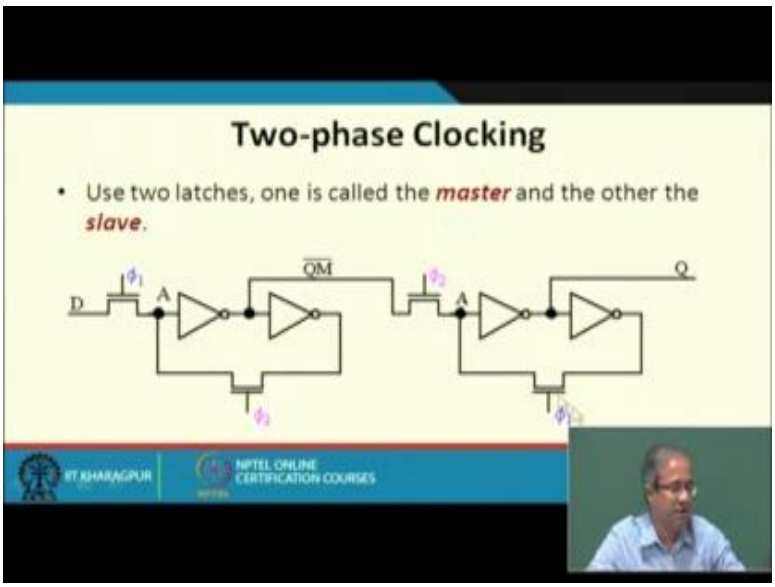
Later on if you move on the flip flopping as I said flip flops are typically activated by the edge of the clock, either the rising or the falling edge.

(Refer Slide Time: 26:31)



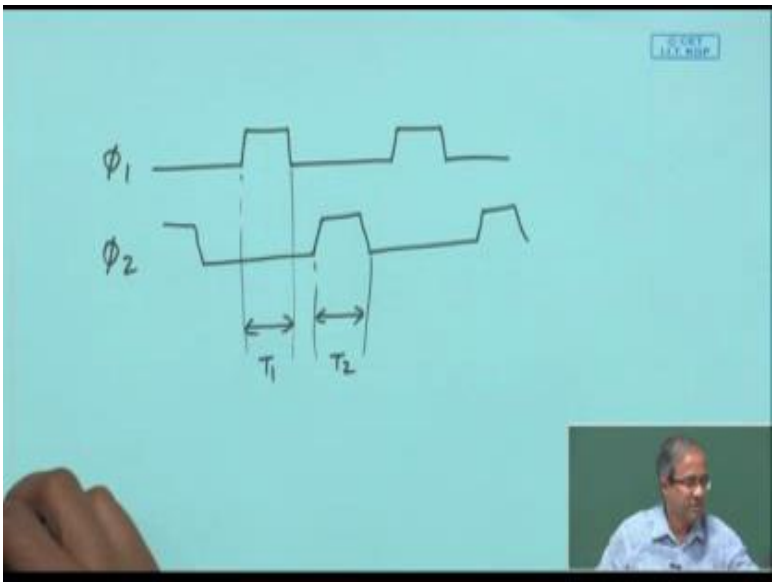
So, without going into the detail this is a typical diagram of a positive edge triggered D flip flop constructed using NAND gates. The basic idea is you have the latch here which stores the data, the circuit which you show here in the beginning this is actually used to store and capture the clock edge and store and capture the value of D. So, both taken together you will be feeding it to the next latch stage and it will get stored. So, basically a positive edge triggered D flip flop will require 6 such NAND gates, this is one of the designs other designs are also possible.

(Refer Slide Time: 27:16)



Now, here we show two phase clocking. For simplicity of the diagrams I am showing transmission gates, I am showing simple pass transistors and not transmission gates. So, here I have a latch, here I have a latch. So, I am using the clocks like I am not showing this phi and phi bar, I am showing them a phi 1 and phi 2, phi 1 and phi 2.

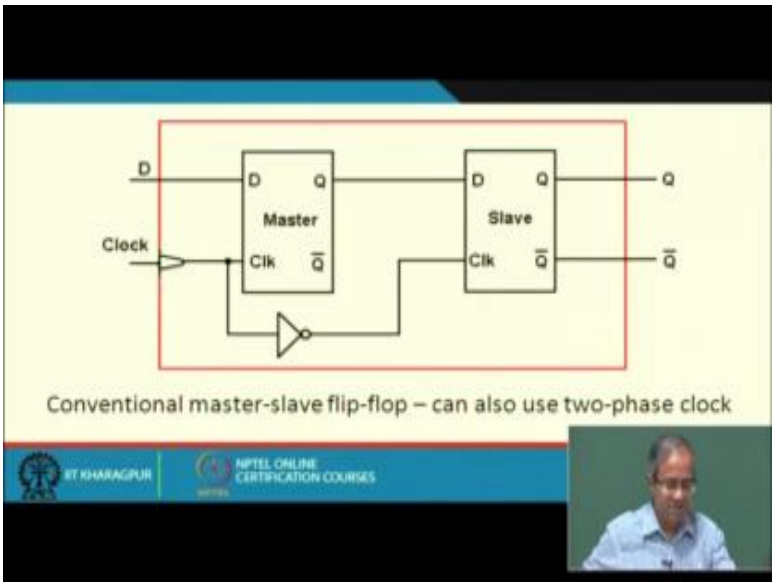
(Refer Slide Time: 27:48)



Now, this phi 1 and phi 2 they are said to be two phase clocks; the clocking will be like this I am just showing the relationship. So, you see phi 1 is active for certain duration, phi 2 is certain for some other duration, let us call them T 1 and T 2 and these two durations are mutually exclusive they do not overlap in time. This is the point instead of using a clock and it is compliments I am using two phases whose means on period mutually are disjoint they do not overlap.

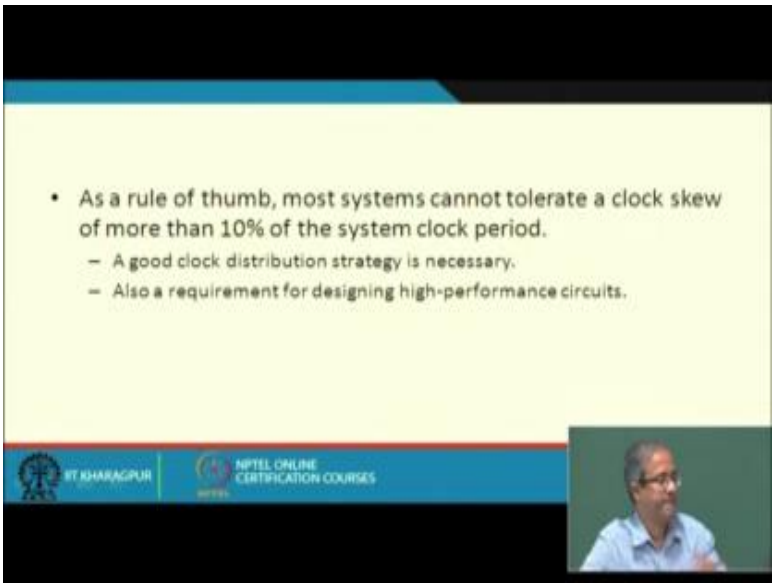
So, in this diagram you see when phi 1 is 1 then this transistor is off, this is cut and whatever is applied at D will come here and get stored here and when phi 1 is 1 this loop is on, and whatever is shored here is stored and when phi 2 is on and phi 1 is of the reverse will happen. So, whatever is stored here that will remain here and whatever is here will get transferred here. So, this is a typical master slave kind of a flip flop constructed using two latches, this is a first latch this is the second latch.

(Refer Slide Time: 29:20)



So, this is the more conventional diagram of a master slave latch, consisting of D type storage, and another D type storage. So, either you can use a clock and it is complements or you can use as I said two phase clock phi 1 and phi 2 both will work.

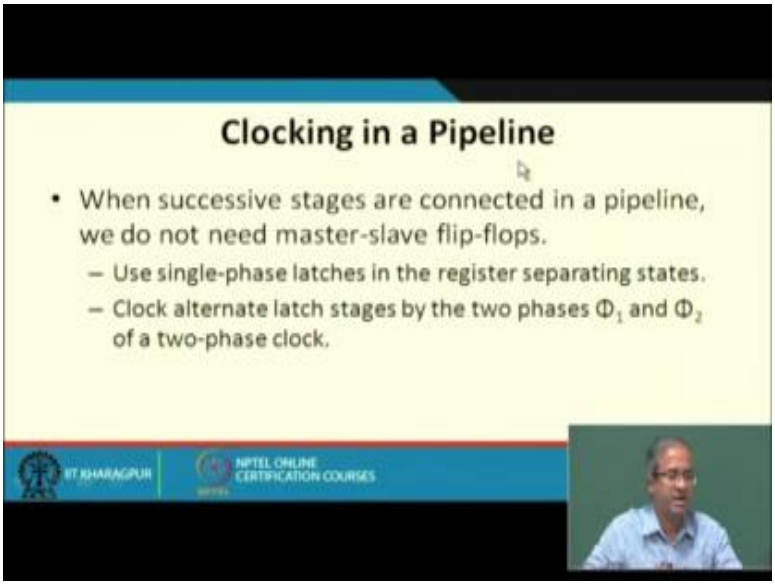
(Refer Slide Time: 29:41)



So, as a rule of thumb most of the systems we design, we have to limit clock skew to within 10 percent. So, although we can have some clock skew, but maximum allowable clock skew is typically as a rule of thumb 10 percent. So, you need a good clock distribution strategy, which we shall be looking at next and as I have said this is a

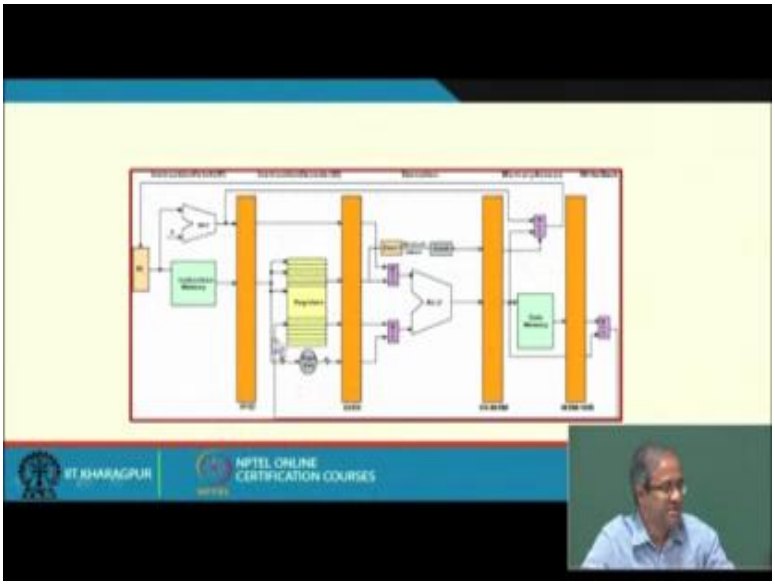
requirement for designing high performance circuit, because we can use clocks of higher frequencies.

(Refer Slide Time: 30:16)



Well when you talk about a pipeline I talked about earlier. So, you have successive stages which are clocked and there is a combinational circuit in between. So, I am showing a typical pipeline diagram.

(Refer Slide Time: 30:33)

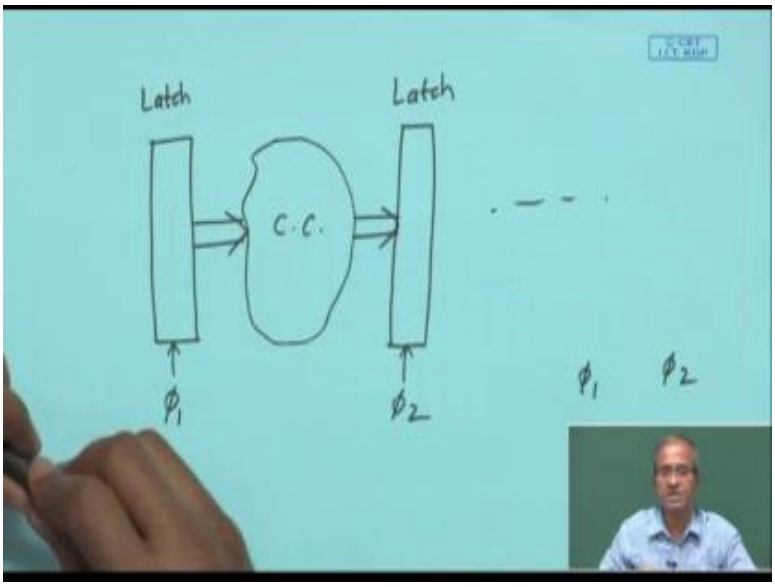


So, you need not have to read what is there inside I am just showing there are some combinational circuits between, and these orange rectangular boxes are the registers. So,

what I said is that there are there is a clock which is spread to the registers and at every clock data shifts from one register to the next through the intermediate combinational circuit.

Now, each register you know to avoid some problems rays and other things, they are designed using master slave technology master slave flip flop; that means, you need master latch and a slave latch just like I mentioned earlier.

(Refer Slide Time: 31:23)



But you see if you do something like this, I am showing one stage of the pipeline; these are two registers there are more and there is some combinational circuit in between this register is feeding this combinational circuit, this is feeding here.

So, normally the same clock is fed, what I am saying is that now let me feed phi 1 here and phi 2 here, and these two latches I am not using them as master slave flip flop, but a simple single stage latches, single stage latch. So, phi 1 and phi 2 will alternate, the next one will be phi one next 1 will be phi 2. So, they will be moving from one latch to another, as if this is master as if this is slave, but it is going through the intermediate combinational circuit.

So, if you do this kind of a design, what it helps you in that is that your latches becomes simpler. So, instead of using the master slave flip flops, we can use simple single stage latches right. So, this saves the area. So, this is what is mentioned here. So, you can use

single phase latches in the registers and the pipeline and alternate stages can be clocked by phi 1 or phi 2 like phi 1 phi 2, phi 1 phi 2 like that.

So, with this we come to the end of this lecture. In the next lecture we continue with our discussion, we shall see more about clock distribution and clock networks.

Thank you.