CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology – Roorkee

Module No # 07
Lecture No # 33
Clocking Strategies for Sequential Design - I

Hello and welcome to NPTEL online certification course on CMOS digital VLSI design we are starting a new sort of the chapter in module today which is basically clocking strategies for sequential design part 1. So this is module number 1 for understanding the clocking strategies now as we have seen in a previous modules that all your design (()) (00:49) when you are doing a sequential design was all clock driven which means that your design as to be synchronize with the clock which means that when the.

For example if you are doing an H triggered design then with the rising age of the clock you should be able to sample the data and except the data for evaluation right. So it is very important and critical to note what are the various problems of clocking and how a clocking can be applied to a sequential logic for the best performance output to do that that is the whole idea between these two modules which will be handling over the next few lectures.

So the topic of the outline which we will be handling in this module are we will be looking into obviously introduce to you subject of clocking strategy we will look at the various the various nomenclature which people follow in sequential design then we will look into sequential your synchronize design techniques what are the various synchronize techniques? We will understand what do you mean by clock skew and clock jitter?

(Refer Slide Time: 02:03)

Outline

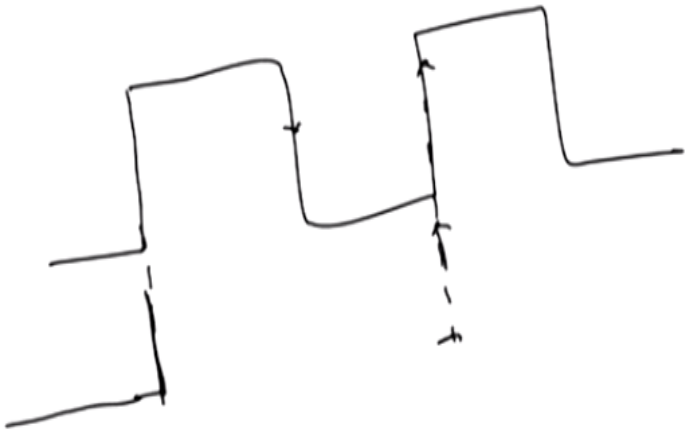
- Introduction
- Synchronous Design technique
- Clock Skew
- Clock jitter
- Impact of Clock Skew and Clock Jitter
- Sources of Skew and Jitter
- Design Techniques to Reduce of Skew and Jitter
- Clock Distribution
- Advantages of Synchronous Design
- Recapitulation

So this two understanding as to be there right so clocks Skew what is the difference between them what do you mean by them and how does it effect your sequential logic flow so therefore that is the fourth fifth part which tell me that if you have clocks and clock jitter how does it influence my performance of the sequential logic right. what are the sources therefore of skew and jitter and therefore if you know the sources you should find out the techniques for skew and jitter.

So that is also a part of the issue and as a part of it we therefore know how you need to distribute the clock across chip. So you should be able to do that in a much proper fashion then what are the advantages of synchronize design that we will be actually revising time and again as we move across in this system and there on this area and then finally we will be recapitulating the whole design flow that is the general trend which will be following for this module as far as clocking strategies is concerned .

If you remember when we were discussing clocking strategies we were actually telling to you we were actually discussing that or it was discussed to you.

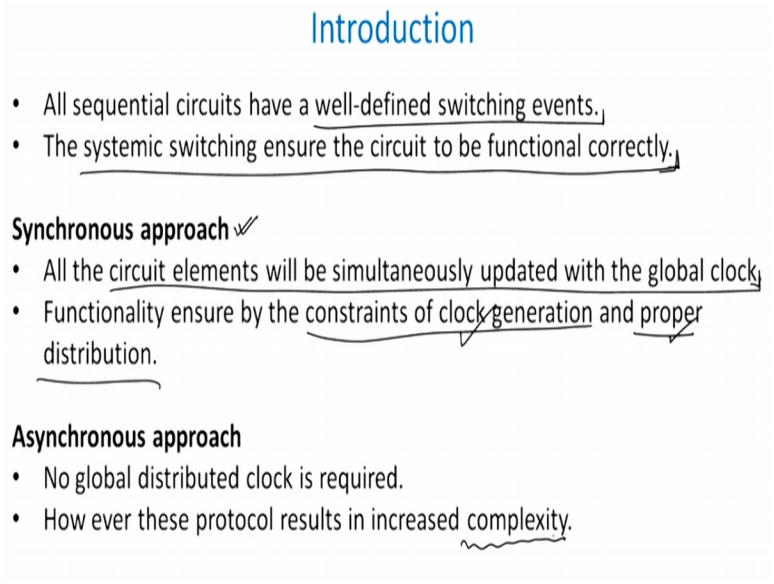
(Refer Slide Time: 03:18)



That during the rising edge so what I am trying to tell you is let us suppose I have a clock here right and every rising edge of the clock you are sampling the data. So I have a input here at this edge of the rising edge of the clock my data is sampled suppose then it falls here again edge is there the next sampling is only takes place at this point only right. So you have to be very cautious that since the data is always synchronize with respect to clock in a synchronize design of sequential logic.

You have to be very cautious that the data is available before the rising edge of the clock set up time as I discussed with you and should remain like that after the passing of the positive edge is basically the holding hold of the statement which is relative. So these as to be always available to you in all sets right and therefore you should not valid the set up time not the whole time that is that is(()) (04:16) you cannot do that right.

(Refer Slide Time: 04:21)



So with this what we will try to do is we will give you what is the meaning of synchronize design as far as sequential logic is concerned as you can see therefore all sequential circuits as well defined switching events right and we have also define what you mean by switching event. Switching event means that if an inverter switch is on and tries to pull an load to 0 or to ground then that even is known as switching event.

So let us discuss what is switching event? Switching event is defined as the event wherein you are able switch an output known either to 0 or to ground and you are able to sustain that for an longer duration of time right. Then you could also ensure that this switching is happening as to be at a very fast space because then only you will be able to allow the data to come inside the picture but then the advantage which come to you as compared to the previous case is like since your clock since it is clock driven therefore you have to wait till the next rising it in the clock.

Incase case of the single edge positive edge trigger or even negative edge of trigger you have wait till the next rising or the falling edge of the clock in order to evaluate the sample right that is the basic issue right. So you do not therefore speeden up unnecessarily accept an issues because if you want to speed up very well know but you will be consuming more power and then therefore the power is power dissipation will be typically very large in that case .

Now systematic switching as to be there to ensure the circuit is functionally correct right unlike in combination logic if you remember in sequential logic you also feedback paths right and in the

feedback path the time taken for the feedback to appear at the input side night be varying right it might even come after the clock age is past so you have to wait till next positive age of the clock to come for it to compare itself.

So you have to be very cautious how much amount of delay you should insert into the feedback path for exact matching between the data and the clock but then if you allow all these systematic switching you have to ensure that the functionality of the switch or the circuit is not compromise right. So functionality is still there which you should be there what is there are two approaches there are actually four approaches but we will discuss only two here the first one is the synchronize approach what does it mean that all the circuit elements will be simultaneously update with global clock.

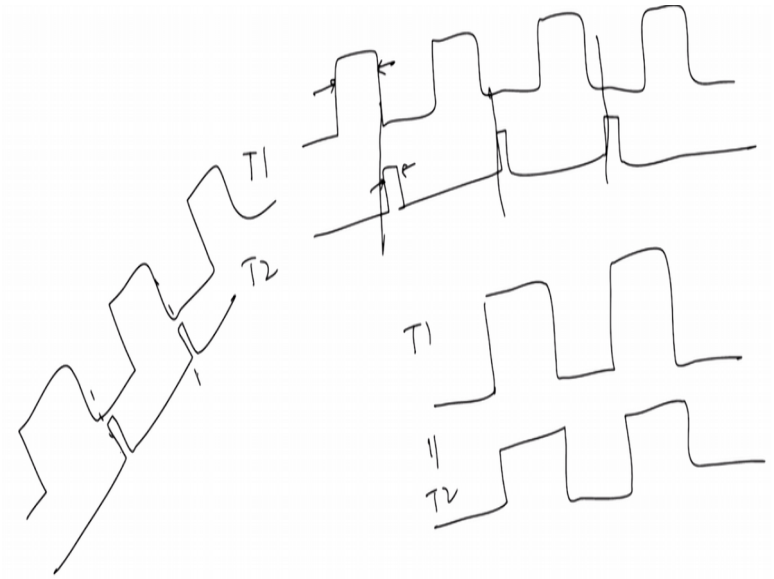
So you have a global clock, clocks are generally off chip because activity factor is one and therefore they are very power (()) (07:02) and therefore you cannot afford them to on chip if suppose practical purposes and therefore you rout through long interconnect from clock to the these sequential logics right. Now periodically the global clock will update the data flow through a system right and therefore you have to be very cautious that when I therefore generally for most practical purposes we try to extract the clock for the single global clock.

So maybe I have a single triple 5 a stable multi (()) (07:37) meter and from there the clock is been generated and try to extract cock from there itself right using long interconnect which is try to send it the data to drive this sequential logics. During the synchronize designs synchronize is basically meaning that your data is synchronize the clock the functionality as to ensure that the constraints of the clock generation and proper distribution that means the constraints of the set up time and whole time as well as how your distributing the clock in special domain.

These are the very two important parts to ensure that system or ship is working properly or not right and they ensure these two things ensure the clock generation or ensure the functionality of the chip is synchronizing of design wherein there is no global clock required. So you do not required any global clock however cost for pay for it that complexity is very large right. So for all practical purposes in this lecture or in this module at least we will be restricting ourselves to

synchronize design only right you can have synchronic design now you can have such kind of clocks I will give you an example.

(Refer Slide Time: 08:45)



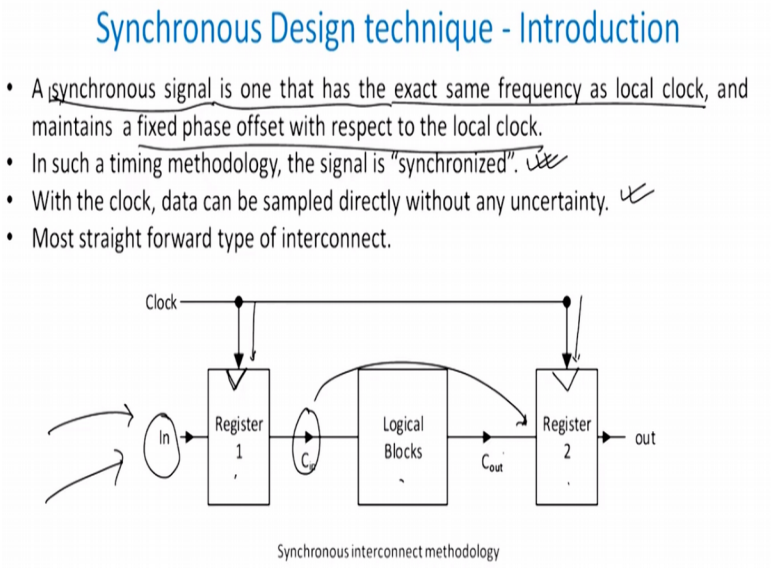
You can have a clock in which there are two clock let us suppose T1 and T2 they are perfectly like they are coming like this they are peak to peak is exactly the same but here might be the case there T1 an T2 might be face differed by certain values right it is clock so it is something like this you get which means that not only its (()) (09:04) is different but it phase is also different right and this that though their maintaining the constant phase different because there are this is also let us assume to be true then they are maintaining the exactly the same clock phase difference but they are not exactly the same clock which means that there is difference between this clock right.

So they are two exactly the same clock but if I have this is T1 exactly equals to T2 if I have a clock something like this and I have a clock which is something like this right something like this right then the it is differing in phase because the rising age are not properly matched and as a result what might happen is there is a phase difference between the two and then that might result in small changes in the logic design right the sequential logic design itself.

So we do have certain other count so the other methods of doing it will not going to detail of this one that most often used in in communication systems you use it right and so will not go in details of this one in any other further details. So you understood what is the synchronize design

we have understood what is a synchronize design and how they are different from each other and how is it that they will be utilized for all practical purposes.

(Refer Slide Time: 10:28)



Let us look at the synchronize design technique right as I discussed with you we will have the same frequency is the global clock a synchronize signal is the output signal which is giving of the data is the one which as the exact same frequency is the local clock so you have a local clock driver which will drive the system and therefore your synchronize signal will be one that will have same frequency as the local clock and will maintain a fixed phase offset with respect to the local clock.

Which means that your data train synchronize signal should be synchronize with the clock means they should maintain a fixed phase offset with synchronize clock right. And as long it is able to maintain that there is no problem in this functionality. So if you look at the second point in such a timing methodology signal is synchronized this is very important right the said signal which you are sending as a data has to be synchronized with respect to the clock.

So it cannot happen that you cannot go on increasing the clock frequency and your data is coming at a lower much frequency that would not work right we have to match the frequency as well as phase. You are allowed to do offset in the phase but in that offset of the phase is to be constant across the time domain or at across the frequency domain right that is pretty important that you should have the same offset available to you all the values of input signal.

So with the clock therefore the data can be sample directly as I discussed with you we have already seen it time and again that there is no required requirement for that. So you see here if you look at the diagram which you see the methodology this is the methodology which people use 90% of its time right and that is the standard methodology which people use for a longer duration of time right.

The methodology is something like this so you have register here register number 1 right you have register number 2 and have a logical block here which is basically try to just a minute I will just yes. So you have a logical block here so in most of the cases which will be studying in frequency logic design you will have to register one input register when output register and there will be some logical block here it may be combination logic also for example an gate or whatever we can have also a simple registered is build within the case this logical block as such.

And you have a clock so you see please see that I am driving a same clock and feeding it to shift register 1 and to 2 right so that I am what i am doing is as I am driving though from the same clock or two signals going to shift register 1 and 2 which effectively means that whenever there is a rising edge coming at shift register 1 rising edge will also appear at shit register 2 when there is a falling edge of register 1 there will be also a falling edge at shift register 2 fine.

But please understand that this logical block will give some amount of a delay right because whenever signal for example I am giving an input here right in the rising age of the clock let us assume that the positive edge trigger so let me say this is H trigger so I have a edge trigger here I have also edge trigger here and this is positive edge trigger so there is no circle here.

So I have an input and in the positive age of the cycle this input goes high and therefore this register able to put it into this Cin value here and it has and then therefore when the positive cycle of the clock as past the register is all in the non-transparent state and therefore the seen as to travel through logical block here read C out here right will reach C out here and has to be ready to be ready before the setup time of register two rising edge of the clock.

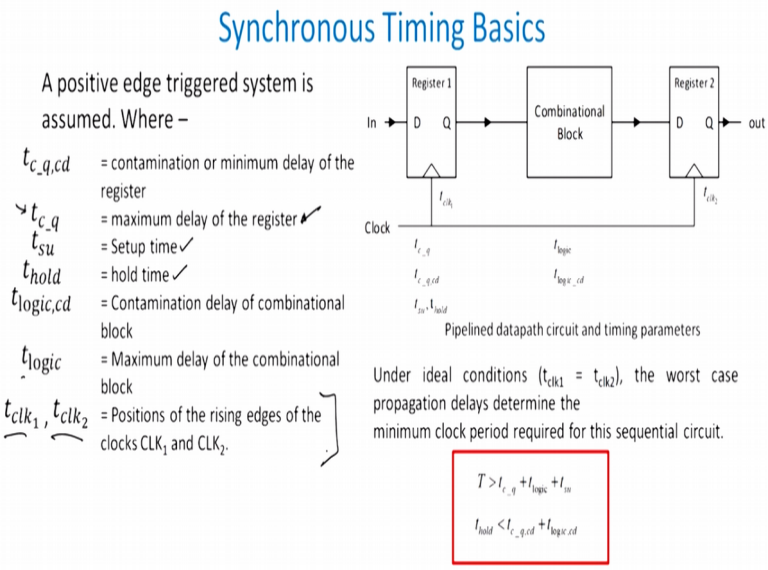
Second rising edge of the clock of shift register two and then it will be able to give the output as I discussed with in the previous term there is no direct connectivity as of now between input and output because these two registers are though working together but they will be non-transparent

during the phase when you are when the clock is not in the low to high or high to low assuming that they are edge trigger design which are available to you.

So this is what I wanted to discuss as far as the basic fundamental block of any sequential logical block is available to you right. So this is the basic sequential logical block which you see and this works fine it is ID or concept is without concept that this is with the concept that your clocks are perfectly okay and there is no clock movement between this point and this point which means that let us suppose this interconnect length is very large right and there is some problem with that clock moving that this point to this point then your sequential design is in problem.

Because then the rising age of this and this never synchronize with respect to each other for shift register number 1 and 2 the rising age are not synchronized as a result output will always be lagging behind and there will be chance that you might have some problem available to you right. So when you are trying to insert data between the clock and input and output register you have to ensure that that they are always available or available for evaluation at least one set of time before the clock register right. So this what we have learnt just now and we have explained to you what I am trying to say

(Refer Slide Time: 15:55)



We were discuss about synchronize timing issues basic timing issues here and will not go into details of this one if you remember we have already discussed with you what is known as set up time and whole time so set up time means the basic minimum time before the rising age of the

clock right when your data should remaining stable and whole time is basically the time after the rising edge of the clock when your data should be stable for proper evaluation of the date or the proper sampling of the data right.

So this much amount of time you have to give whatever you want to do it you can do it but this much the amount of time which you have to give it. This gives you a restriction on the total frequency of this sequential logic flow right let me give you the basics one first of all so tcq which you see the second one cq is basically the maximum delay of the register this I define as the maximum delay of the register.

We also define maximum delay means from input of the register to the output of the register the delay between those two is basically defined as the tcu delay right c2q delay tcq write down t logic is basically the combination logical block a delay because this will be composed of certain logical block cells and the delay associated with that will be basically defined as the tcq t logic delay right t logic delay will be defined.

So t logic is that maximum delay which is available to you please be very careful that in combination logic may be discuss with later on but there might be certain so how did you define a maximum delay you delay across the path where you have the critical path. So you need to first understand what is the critical path combination logical path and then across the critical path the total delay associate with that is basically defined as my t logic or maximum delay of combination logical block right.

We have therefore tc clock 1 an tc clock 2 these are the position of rising agent of clock 1 and clock 2 so I have two clocks clock 1 and clock 2 tc clock 1 and tc clock 2 are basically the rising areas of these two clocks right and they are they will be working on this areas to a larger extent so with this I will stop here today next turn we will take up this synchronize type basics and then we will look at the various strategies of sequential logic design thank you very much.