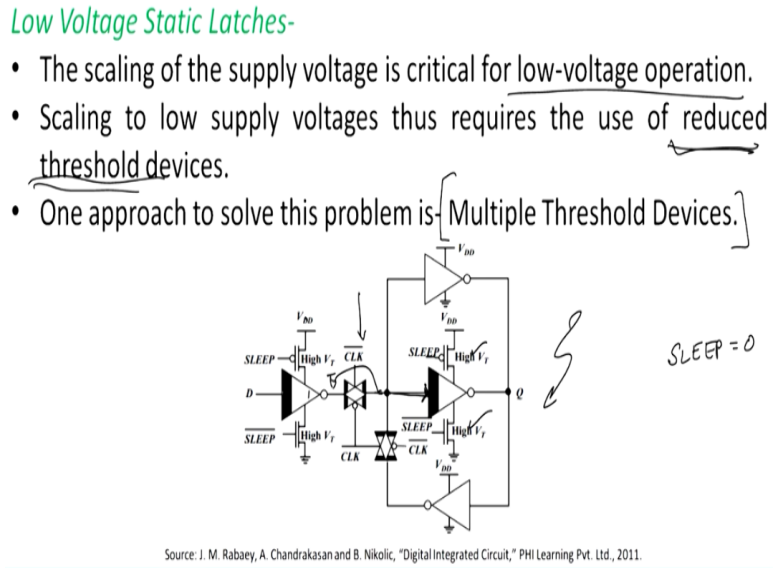
CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics & Communication Engineering
Indian Institute of Technology Roorkee

Module No # 06
Lecture No # 29
Sequential Logic Design – V

Hello everybody and welcome to the NPTEL online certification course on CMOS digital VLSI design. We will start with the sequential logic design 5 module number 5 and in our previous module we have seen how does a edge trigger based mugs based latch work and how we can convert from non-ratio to ratio logic in order to reduce the clock load. In order to reduce the clock load it is a good idea to make it ratioed to a non ratioed to ratioed logic that means make your feedback inverter slightly weaker as compared to your forward inverter and that makes your life easier in terms of having a problem.

But the problem I also which we looked was that you did have a problem of overlapping of clocks so if you have 1 and 1001 lap you might have condition where you have reverse conduction takes place also so when the mythology adopted is you generally design generally have the you can avoid it using two non-overlapping clocks as you see on this side of you slide.

(Refer Slide Time: 01:41)



This is basically the development of the two non-overlapping clocks which is basically a PHI1 and PHI2 so you have PHI1 which is coming here and you have PHI2 which is coming here. So

the idea here is that the clock and clock bar should be just in the opposite phase so clock = 1 clock bar = 0 so I therefore need not necessarily invert the clocks to get it but as long we are able to make them such that there is some phase difference between them in terms of delay or delay between them and this delay is always keeping constant is not changing with respect to time or distance then we can actually have a overlapping clock.

So if you see PHI1 and PHI2 so when PHI1 is high right is 1 your PHI2 is low and therefore this is low and therefore this PHI2 is low here so you see what will happen here is that when PHI1 is high the D is getting evaluated by this transistor by this inverter it comes here it goes to this loop since your PHI2 is 0 you will never have this loop corresponding on therefore a data will kept here at this particular point right if you convert this into a similarly this is PH is 0 your slave is disabled now if your PHI2 is driven to a PMOS right this will be on state whenever the PHI2 is 0 and therefore this so this will what happen is that this will go like this right.

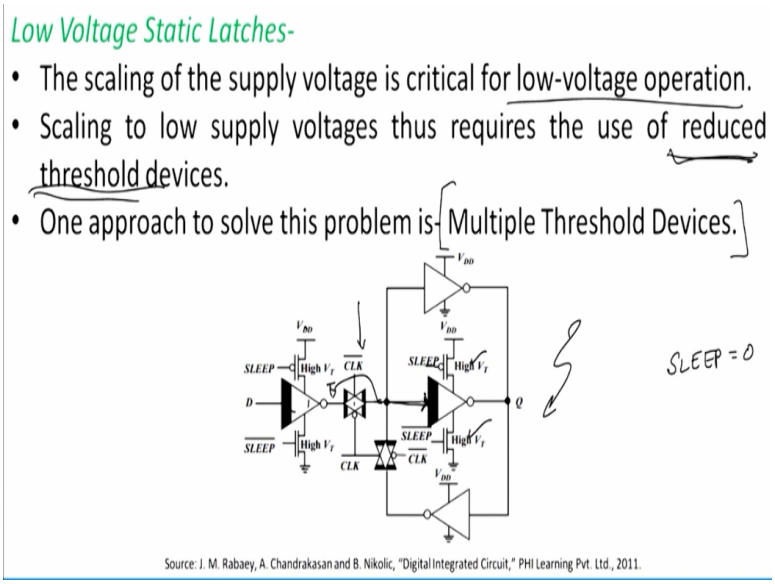
So it will be recapitulating in the next phase when the PHI2 is goes high and PHI1 goes to 0 this goes to 0 data this is cut off data is not sample PHI2 is 1 now and therefore whatever the last value of data was available here will be sampled by this particular point it comes here to Q. Similarly PHI1 is basically low now and therefore this feedback is cutoff and therefore the Q value is stored.

Now this again if PHI1 was driving a PMOS inverter here then PHI1 = 0 this feedback look will be available to you and therefore this Q value will have its own storage area. So this is he methodology which you talk about the idea here is that you are using a pass transistor logic right we are not using a tgd right we are not using a pass transistor and as you very well know pass transistor signal integrity is a big issue that means an NMOS will be able to pass a one which threshold voltage drop right.

And PMOS will be passing 0 with mode vtp rise now which means that if you go from tg2 and NMOS logic which is basically a pass transistor logic you might end up having a signal degradation at the input to the slave. So master will not primarily will have a big problem but your slave will might have a problem from that it is not able to get the actual value of voltage at the input side of the slave right and that makes the life difficult for this case into consideration.

But none the less for low frequency operations you generally use such type of non-overlapping clock based design no more so then offered. With this we have understood the basic idea of why do you require a non-overlapping clock and how can you generate a basically NMOS transmission pass transistor based logic mugs latch we can also have a for low pass for our low power requirements static latches wherein you use multiple threshold devices.

(Refer Slide Time: 05:13)



So you use basically multiple threshold or reduce threshold devices right wherein the critical path you use supply because see the idea was that you require to lower the power right because your drive it by clock. Clock is activity factor of 1 if you remember your power dissipation is in module activity factor of one will give you approximately the same power so which means that for every clock cycle is always a power consuming cycle right and therefore it is heavily power consuming.

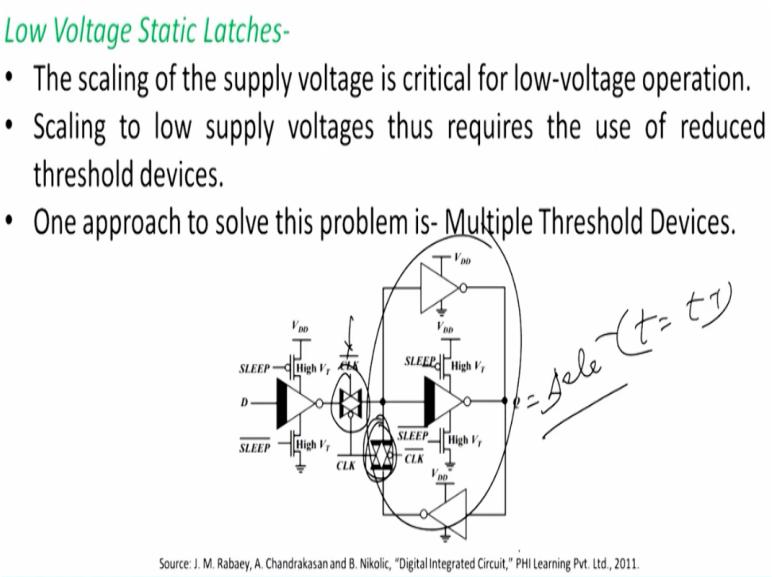
And therefore to offset the heavy power consumption we require to either go for low voltage operation which is this one right at the cost of some performance or we try to use a reduced threshold voltage devices at critical path now as you can see here if you look very closely the diagram which you see right and then so one problem is that you use a multiple threshold devices.

So in the critical path used low threshold devices in non-critical path you use high threshold devices right so that you do not allow a signal cross through non critical path and the power dissipations are low and through the critical path allow if you have threshold voltage device it will allow it to with low power now if you look at this point if sleep let us suppose sleep is signal which is given to you then if sleep is basically 0 then sleep bar will be 1 and therefore what will happen is that I will have inverter which is driven by VDD and 0 of course and this will switch on the device D right.

And so the device basically which is basically an inverter made up of low values will be connected to VDD and ground for sleep = 0 and sleep bar = 1 and D will come here D bar will be here if clock again = 0 and clock bar = 1 and clock bar = 0 this transmission gate become transparent and this D bar appears at this particular point. Similarly now what happening is that if sleep = 0 this is on this is on right and therefore this voltage this D appears at this particular point right at this particular point.

As it appear from this particular point it again inverts back and comes to Q right so this is the general flow of devices which you see in the critical path which is D this this and this try to use try to use low threshold voltage devices for the larger extent of time.

(Refer Slide Time: 07:48)



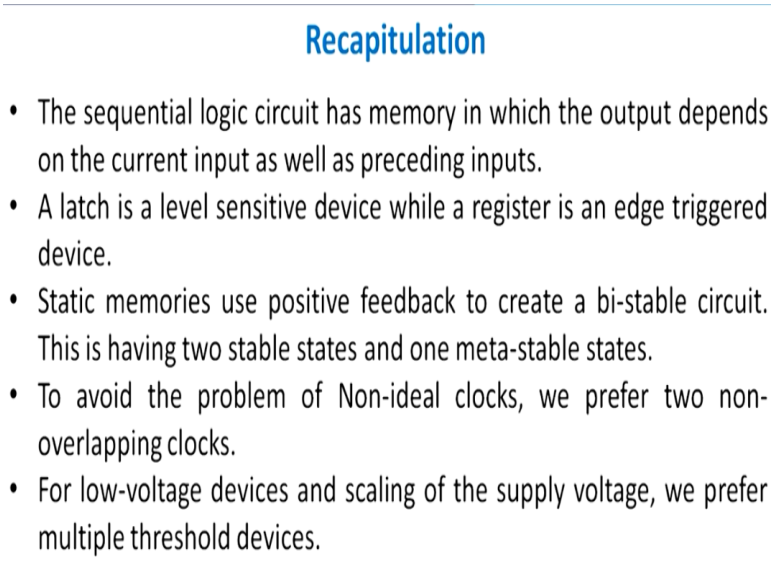
Now if you look very carefully from all these discussions you will see that I have used the tg here right I have also used tg here right. So when clock = 0 or clock bar = 0 this switch is on and

therefore it connects this to this right when clock = 1 then this is cut off therefore this is able to hold the data fine and therefore Q is fixed to a value of that equals to data we use data in previous data t – t = t – 1 data should be fixed at Q value.

So these are low voltage static latches is used very often in low power design issues where you use sleep transistors to do it but please understand sleep transistor are high Q transistors which primarily means that you can offered the high vt because they do not come into critical path they just allow your inverter to get switch on and off and therefore you are you can offered to have a high vt devices high threshold voltage devices in on critical path.

That is the reason you use multiple threshold devices across the whole path as far as this understanding right. So we have finished this with the low voltage devices let me therefore recapitulate what we did till now as far as static latches are concerned we will discuss only the static logic at this particular point of time. So we have understood the sequential logic in a is a memory whose output only depends on the current value of the input but also in the previous value of inputs right.

(Refer Slide Time: 09:13)



A latches and level sensitive device and a (refer time: 09:13) is a edge trigger device right static memory used feedback loop to create a bi-stable circuitry that we have seen and we have got 2 inverters connected back to back it is starts to behave like a latch and therefore any data which is

to be stored will be regenerated or re furnished and you will always get a fixed value of data available to you.

And you will as I discussed with you will have two stable points and one meter stable point or unstable point right the unstable is the middle of VTC voltage transfer characteristics and the stable point or the two edges of your of the transfer characteristics right. So why meta stable why it is known as unstable meta's stable because there if you give a small change in input your differential gain as so large that you will get a large change in output and this will be fed back again to the other inverter and it will go off and it will go on to a continuous fashion and you will get a large value in the value of the voltages in the output side right either 0 or 1.

Now we so we also understood what is basically a master slave condition edge trigger design what will the problem if you have a overlap available to you and can I remove it by using two non-overlapping clocks right PH11 PHI2 has been discussed here. For low voltage device and for scaling we require threshold voltages in non-critical path and low threshold voltage in critical path to reduce the delay.

So this gives a basic idea of the static latch now we come to the next phase of the next part which is basically the dynamic logic on the registers right and what is the dynamic latch therefore dynamic latch or a shift register is something like this if you remember the dynamic CMOS dynamic latch we did or earlier was that now static was independent of clock so basically if I noticed to showing one it will store node 1 for a long duration of time whereas the dynamic design is where in i can store a value of voltage at a particular point for sure but the storage will be finite in time domain or we will have a finite time domain unless right.

So you can store only for 3 milli second right so if you are able to extract the voltage in 3 milli seconds it is okay if not then that voltage value is vertically gone right. So we will discuss dynamic latches and registers her.

(Refer Slide Time: 11:34)

Outline

- Dynamic Latches and Registers
(a) Dynamic Transmission Gate Edge Triggered Registers
(b) C^2MOS Register
(c) True Single Phase Clocked Register (TSPCR)
- Alternative Register Styles
(a) Pulse Registers
(b) Sense Amplifiers Based Register

We will first of all take up dynamic transmission gate based triggered register then we come to C2MOS registers we will also discuss TSPCR which is basically true single phase clocked registers TSPCR then we look into registers styles and pulse registers in sense amplifier based register right. So just look at various styles of switch registers in this module and let us see how it is works with the design manipulation or design aspects ratio already done right and that is the most critical path of this module at least in this case.

(Refer Slide Time: 12:09)

Dynamic Latches and Registers

- When the registers use in computational structures that are constantly clocked, the requirement that the memory should hold state for extended periods of time, can be significantly relaxed.
- The temporary charge storage on the parasitic capacitor is used to represent the logic signal.
- The absence of charge represents 0 while the presence of the charge represents 1.
- To preserve the signal integrity, a periodic refresh of the value is necessary; hence the name dynamic storage.

Now you see the first point here so when the registers used in sorry used when the registers use in computational structures that a constantly clocked the requirement that the memory should hold state for extended period of time can be significantly relaxed right what does it mean? For

example as I discussed with you if you have a master and slave as long as you are delaying the slave to except data right the master is going to storing the data in the input side of the slave right.

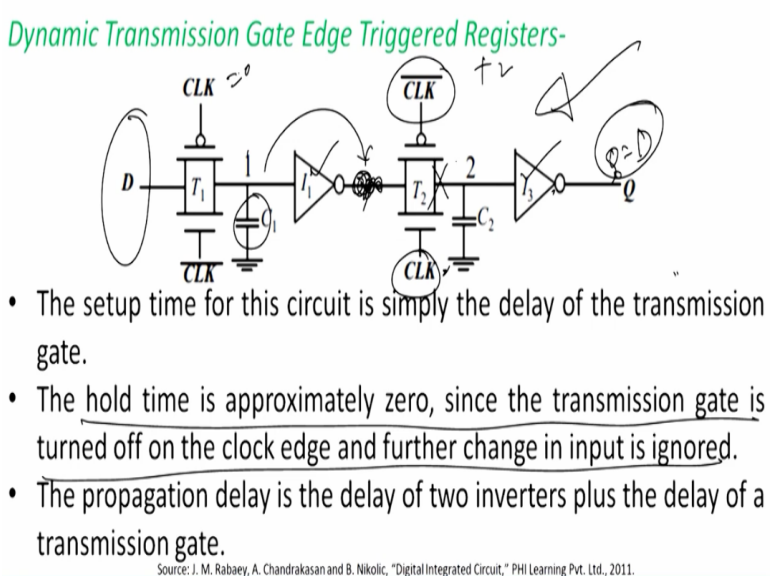
And therefore ahh I can simply extend the storage time at that particular load by extending the clock period so that is what it is written first type that when the registers used in computational structures are constantly clocked so you have clocked the requirement of memory should hold state of extended period it becomes means you do not therefore was simply changing the clock period I can allow the memory not to hold a particular bid of data for a longer period of time.

The temporary hold a particular bit of data for a longer period of time that temporary charge storage on the parasitic capacitances used to represent the parasitic capacitance which means that during the time for example the clock is low and your slave is disable then the output of master will temporary store the value of the data which is available to you in the input cycle right and that is the basic concept of the dynamic latch.

The absence of charge will be 0 and the presence will be the charge with equals to 1 right but the problem here is in that problem is that we were at dynamic logic that if you storing a charge and name it as 1 right then there is a finite probability that the charge which is stored there will start to leak right and when it starts to leak a voltage value will start to fall down so you require periodical refreshment of the voltage at the output load or even in the input node for the system to work in a proper manner right and that is the reason why we refer that this is the dynamic storage and therefore we require issues are concerned which we will be able to deal with very day.

Which means that we will be able to refresh all the logic in this case okay with this idea or concepts let us look at the dynamic transmission create edge trigger which is in front of you.

(Refer Slide Time: 14:39)



So this is the diagram of the dynamic TG based register if you look very carefully again as I discussed with you when clock = 0 and clock bar = 1 t1 = 1 D comes to 1 charges this capacitors C to a high value this high one inverts this to 0 let us suppose 1 to 0 and that instance of time this since clock = 0 this t2 is off and therefore this stores the data at particular point in the next half cycle clock bar goes to 0 and clock goes to 1 therefore what will happen is this t2 becomes on this appearing as Q = basically D bar so Q will be equal to D right is it okay.

As I discussed with you in the previous case here also whole time will be equals to 0 right this we have already discussed with you and we are discussing the basic logic that whole time will be 0 in this case because you are just sampling the data in the slave only at the positive rising edge of the clock positive assuming that the slave is being basically positive edge trigger design and master is the negative edge trigger design right.

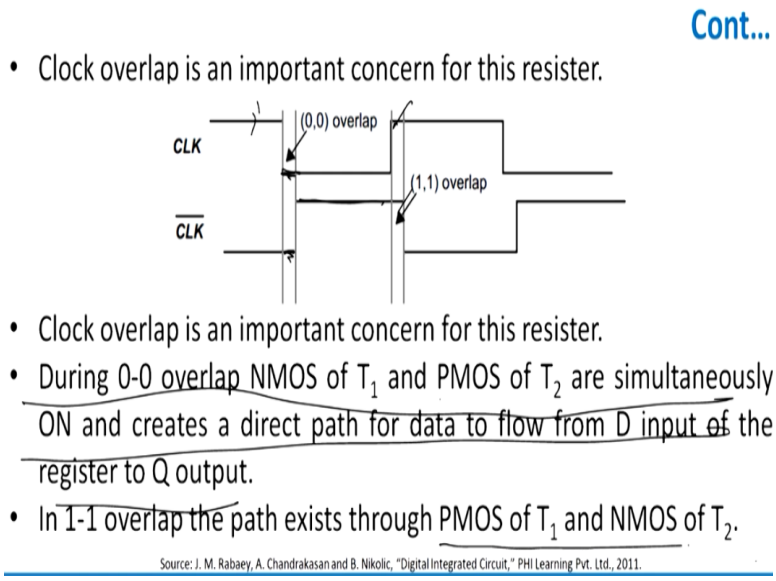
If you consider this to be true then you are actually sampling the data for the slave at the edge of the clock period when the clock is going high from low to high condition because it is positive edge trigger clock and therefore you see the whole is approximately equals to 0 as I discussed with you in the earlier case since the transmission gate is turned off on the clock edge and the further change in the input is ignored fine.

What is happening T2 is just cut off right and therefore anything happening in the data side which is this side which is basically not known to you. Similarly when you are evaluating it then

T 1 is cut off let us suppose then any change add D will not be reflected at the intermediate node and therefore this will give you a very static signal a very sort of very integrate signal available to you.

Now the propagation delay is basically the delay of two inverters + 1 transmission gate so this inverter + this inverter + this transmission gate if you 3 delay that is the delay which you get from here right please understand why did not you add T1 delay while doing it. So please understand why we did not do it right or you can have a look into any of the book standard book in why we are getting a change here right. We have seen one important point that we have discuss this earlier also that you do not require clock overlap right but let us suppose you have a clock overlap this overlap is known as 00 overlap where clock is 0 is overlapping with clock bar.

(Refer Slide Time: 17:29)



So clock bar is also 0 and clock is = 1 and therefore clock = 0 and clock bar = 1 so for a finite duration of time when you have 0 0 you clock = clock bar right and that is technically not is a long issue similarly for 11overlap you will have this equals to this and that is what is known as the clock overlap right okay.

Now let us suppose let us give you an idea right so 00 overlap NMOS of t1 which t1 NMOS of t1 right so let me say i do a I do a 00 overlap 00 overlap primarily means that my clock = 0 I = 0 fine which this condition of clock if clock goes to 0 and clock bar goes to sorry clock bar goes to 1 then clock goes to 0 means this is on clock bar = 1 means basically t1 is on. so t1 is basically

on in this case t1 is on so if you look clearly here so this is what happening between 00 overlap NMOS of T1 PMOS of t2 was simultaneously on and creates a direct path for order to flow from D of the register to the output Q right.

Let us see what it is mean it what does it mean effectively if you look back if you look back carefully here then when you have 11 overlap for even 00 overlap then let me tell you when you have 00 overlap then clock =0 and clock bar =0 right clock bar = 0 you will have this problem of so what is happening is basically your PMOS of T1 gets on right and PMOS of sorry so this all will be 0 here because this have to be 0 then PMOS of T1 and PMOS of T2 will be open and as result there might be a condition where D and Q are transparent to each other right.

So they are transparent to each other primarily meaning that any changes in D will be reflected on to Q provided that change is happening within the short period of delay whereas signal goes from input to the output of the system right and that is required critical this thing similarly the whole limit is approximately 0 since the transmission gate is turned off on the clock edge and further change in the input is ignored I think I have already del this point time and again that your whole time are always equals to 0 right and the whole time is 0 because and the reason being that once your clock edge has passed you cannot do anything else until and unless you wait for the next clock edge right.

So therefore that is the reason it is quite small so the propagation delay as I discussed with you some of the two inverters delays + the delay of the transmission gate is basically my propagation total probation delay from tcq right okay awe have seen what is happening here and we have seen the so we have seen so for 00 overlap NMOS tn NMOS of t2 are simultaneously on t1 and t2 means t1 and t2 is basically this one so t1 and t2.

So you have t1 here right NMMOS of t1 and PMOS of T2 gets on simultaneously and as a result what will happen is that the data flow available to you for input to output when one overlap is there then the path exist between PMOS of t1 and NMOs of T2 where you have 11 overlap 11 overlap PMOS of t1 and NMOS of T2 get on right and therefore you end up having a direct path between D and Q which want initially for your profile takes place. To remove all these basic fundamental principles of this we define C2MOS logic C2MOS logic so this is basically a clock

bar controlling mechanism choosey right so I think we will take up this one in the next discussion and explain to what is C2MOS logic and then we will what we can do in the sequential logic right okay thanks a lot