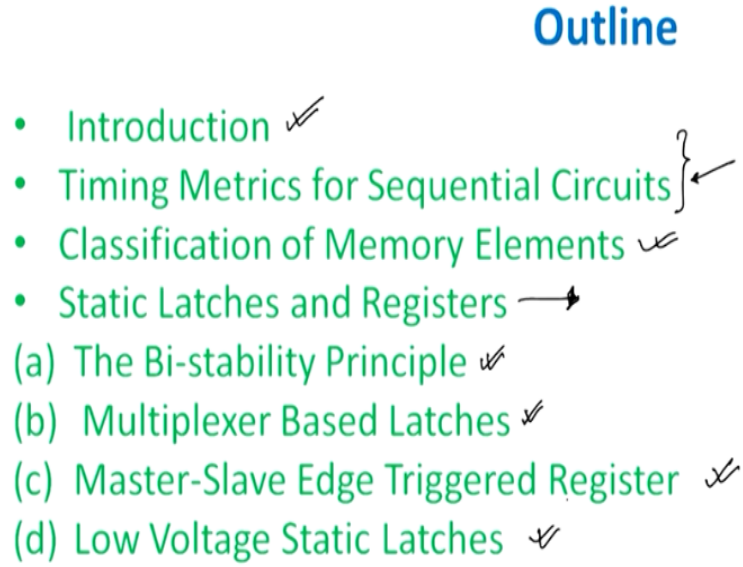
CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology – Roorkee

Module No # 05
Lecture No # 25
Sequential Logic Design– I

Hello everybody once and welcome to the NPTEL online certification course on CMOS digital VLSI design we have till now understood the basic the previous module we had understood logical effort and the way logical effort helps us to find out the delay in a combinational logical block. Now we will starting with sequential logic design it is a new session or new chapter you can say and we have finished with combinational logical block and we will be starting with differential logical block and sequential logical design as far as this module is concerned. So this module will be primarily focusing on the basic concepts of sequential logic design and within that within the next within this module we will be actually seeing the following things the outline of the whole module is something like this.

(Refer Slide Time: 01:21)



So we will introduce to you what is basically a sequential logic right and how it is different from the combinational logic and where we will be using sequential and where we will be using combinational logical blocks now unlike combinational in sequential circuits timing issues are very critical and one as to be quite cautious about the timing analysis or the timing of the data with respect to clock or with respect to each other.

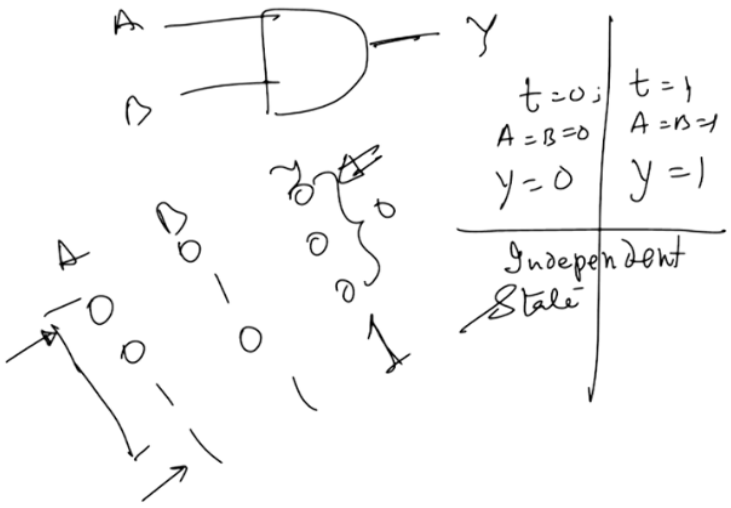
So we will be understanding various timing matrix of the sequential logic or the sequential circuits or so that these sequential circuits work properly and there are no violations as far the data is concerned that means you are able to properly feed the data process the data and take the data out in the output in the output part without compromising on the quality of the circuit itself.

Subsequently we will be looking into sequential circuits as memory elements I will explain to how does it work out and then we will go for static latches and registers. So what is the difference between shift registers (()) (02:35) registers and static latches within which we will be covering the Bi-stability principle we will also looking at Mugs based latches and then we will look at master slave edge triggered registers and low voltage static latches right.

So we will therefore see the different between latches and registers we will be also concentrating on the fact that how latches and registers are different from each other in terms of its operating conditions and so on and hence so forth before that we will actually for Bi-stability but means understand the basic concept of a memory location and what is the concept of Bi-stability as far as memory is concerned.

So let me start the basic fundamental principles of sequential logic and as we have seen that if you look very closely combinational logics circuits are just the functions of your inputs at that instant of time right. So which means that I have a combinational for example AND gate right if you look at it AND gate then you will appreciate that in AND gate you will.

(Refer Slide Time: 03:46)

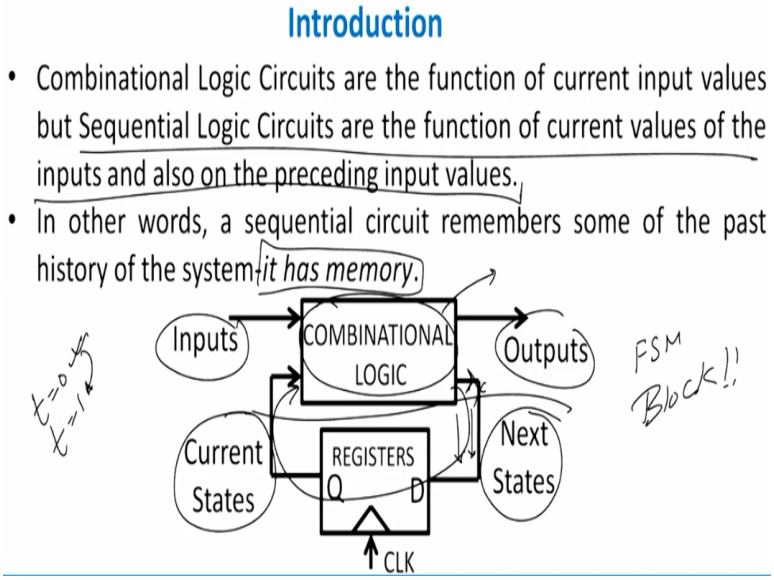


For example I have an AND gate right A and B are there so what is the truth table all of your aware of that why will be so A, B, C so if I draw the truth table i get right so this is the truth table as you get for the NAND gate which means that once your input pattern is fixed at A = B = 0 output as to be 0. Similarly if any on the 2 conditions 3 conditions are met output will be always equals to 0.

But when the input conditions are 1, 1 output will be also equals to 1 which means that at no point of time the value of the output depends on the inputs of the previous time right. So at t = 0 you had t = 0 there as t = 1 suppose at t = 0 suppose A and B where both equals to 0 and A = B both are equals to 1 so that t = 0 y will be equals to 0 at t = 1 t will be equals to 1.So what does it tell me that these two are totally different state available to you right so they are totally different states and the distinct states available to you.

At t = 0 and t = 1 and for calculating the value of y at t = 1 you actually do not require any information of the output voltage or current anything at t = 0. So they are totally they are independent states and not only they are independent states but they are actually in a sense they do not influence each other also right. So that I what I wanted to say for a combinational logic.

(Refer Slide Time: 05:36)



But for a sequential logic right for a sequential logic as I discussed here the output the current value of inputs also on the preceding value of the inputs so the output the for a that only depends upon the current sequence of inputs or current values on of inputs it also depends upon the previous values of inputs available to you right. So even if you do not understand any part of it as just as a layman you can understand that since the current value of voltage or the output depends on the previous value so you do have a concept of memory here because the system is able to memorize or at least stuck with value by inputs which are available to it in the previous cycle.

So anything like that is sort of the memory because it is storing something for some period of time to evaluate the result right. So all your sequential elements are memory basically part of memory only whereas combinational logical locks are not memory they are instantaneous in nature depending upon the value of input at the particular instant. So that is what I was saying in the second point that it has memory right it has memory.

Now if you look at the basic block if you forget about say you forget about this this part this part you forget about lower part right then you see input and input you get an output there is no feedback loop then we define this to be as the combinational logical block right this is been going on several example NAND gate, OR gate, NOR gate, XOR gate any of the standard gates which you use follow this type of logic where in you give an input you get an output here and this happens to be the core combinational logical block which is available to you.

But now what we are doing is sequential logic case we are feeding the previous state back to a register or some element back into the input right back as an input. So these are so these are basically the current states you do a evaluating of the combinational logical block I come to the next state and the next state enters the resistors this something happens there feeds it here and to this is the loop system available to you right.

So this is basically a FSM basic as a FSM block which you see so this is basically FSM block which you see finite state machine block right and they will generate you various FSM modules across the network right. Now as I discussed with your therefore unlike combinational therefore sequential will always have some amount of sort of memory element available to you why because if you look very carefully the data is basically making a loop across this network which means that at t = 0 suppose I had output then even at t = 1 this output of this will be t + input to the second stage right.

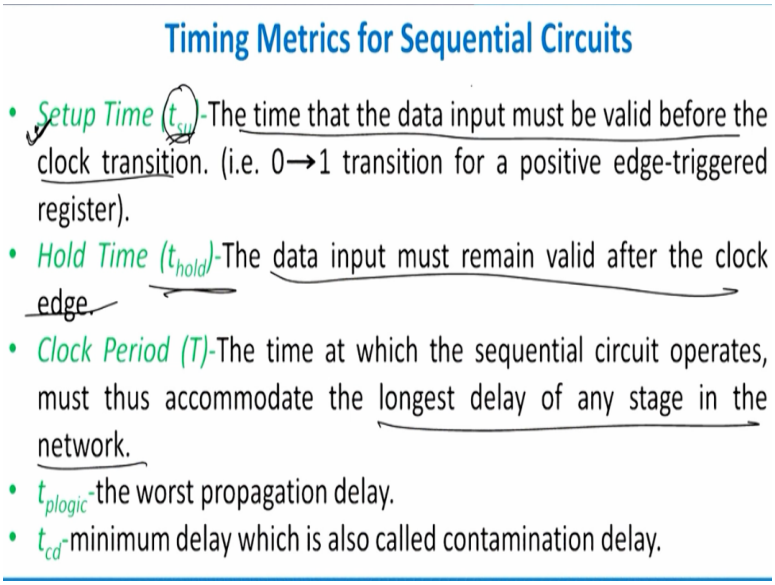
And therefore it will return certain values so therefore the output in the second stage will not only depend upon the input of the second stage but also upon the previous state output right. So that is quite interesting a way of looking at it so let us look at the timing matrix so with this knowledge where you have gained till now that the a combinational logical block is primarily a logical block where in your inputs are in total (()) (08:56) with the output whereas in the sequential logical block your outputs not only depend upon the input of the presence state but also upon the input of the previous state right and that is the reason you have to hold.

Now you see therefore why timing is very important sequential is that till a time so your output has to be held till a time that your next input is actually coming to the combinational logical block otherwise the combinational block will not understand from where input is coming is it coming from sequential is it coming from resistor or it is coming directly from the input right.

So what will happen to do as the way to wait till one loop of your output actually reaches the combinational block right and therefore your these are quite essential or quite important observations as sequential logic is concerned right and you have to very cautious in terms of observing why and how the timing diagram will effect overall thing. There are certain definitions of timing for sequential circuits which you should be aware of before we move forward in this

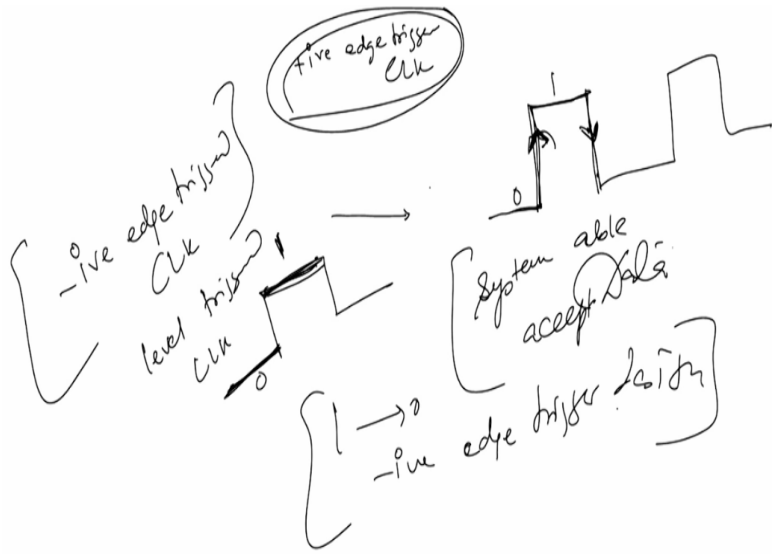
design and these are basically simple definitions which you should not only remember but understand why it is coming.

(Refer Slide Time: 10:19)



The first is the first timing is basically the set up time and also the effort to us t suffix su right. Now let us suppose I have I will define certain terms here which will be quite interesting that there are two types of clock which is available.

(Refer Slide Time: 10:41)



Suppose I have date coming right I defined a clock basically positive edge triggered design right let us suppose I define is known as the positive edge it basically means that when the clock is rising the positive edge which is this one right this is the falling edge again rising again falling so

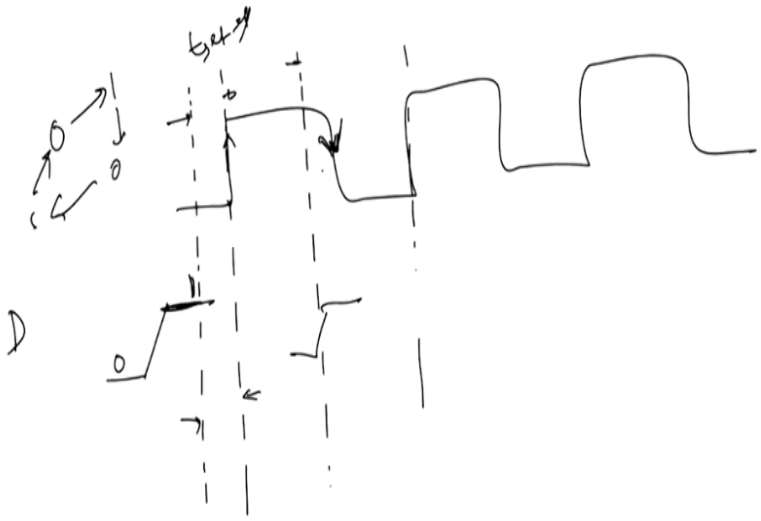
clock is going on during the rising edge of the clock the system is able to except data when the clock is rising.

Then we define that to be as positive edge triggered design or a positive edge triggered clock is it okay that means when my clock is rising you accepting the data from out from the input and you are doing something you can also have a negative edge triggered clock which again as the name suggest in the negative cycle when the clock is falling down at that point of time if the data is sampled we defined that to be as negative edge triggered clock.

So when I say positive edge triggered or a negative edge triggered I mean to say that when the clock is either rising or falling your data will be synchronize with the clock itself right you can also level triggered clock which what is it mean? Level triggered basically means that you will have so when the level is 0 or the level is one so when the level is one then it is sampling let us the data then we define it be as a positive level triggered level triggered design so I have a level triggered design level triggered clock right I get also when 0 is there it is actually excepting data or one is the resisting data.

So this is the positive level triggered this is negative level triggered this is 0 to 1 transition of the clock is doing I define this as to be positive edge trigger if it is 1 to 0 clock is doing hen define that to be as an negative edge trigger design fine 1 to 0 or 0 to 1 you will see right and therefore these are the very important issues which you should be carefully noted down. Now the setup time is defined as that time when the data input must be valid before the clock transition I will give you brief idea what i am trying to say.

(Refer Slide Time: 13:18)



Let us suppose I have a clock here right and I have a data input D here so D will go from 0 to 1 right this is 0 to 1 transition what it tells me is that just when we have a clock edge available to you your data should remain at 1 right at least a minimum few some units before your rising edge of the clock your data should means stable right which means that between this point and this point is basically known as t set up therefore between these two points that data cannot go from 0 to 1 and then from 1 to 0 and then again 0 to 1 and then 1 to 0 no.

Once it reaches this region that data should be stable at 1 or 0 whatever the value is right for positive edge triggered you got the point so for a positive edge triggered this is what you see for a negative edge trigger it will be something somewhere here right so for a negative edge triggered this is the point where you should trigger it but my data should be stable at least till point this point this point fine.

So for a positive edge trigger my set up time is basically the minimum time before the rising edge of the clock when my data needs to be stable and negative edge trigger clock is basically by setup time before the negative edge trigger when the clock is basically equals to 2 goes and then data is excepted so this is what define as the set up time issues.

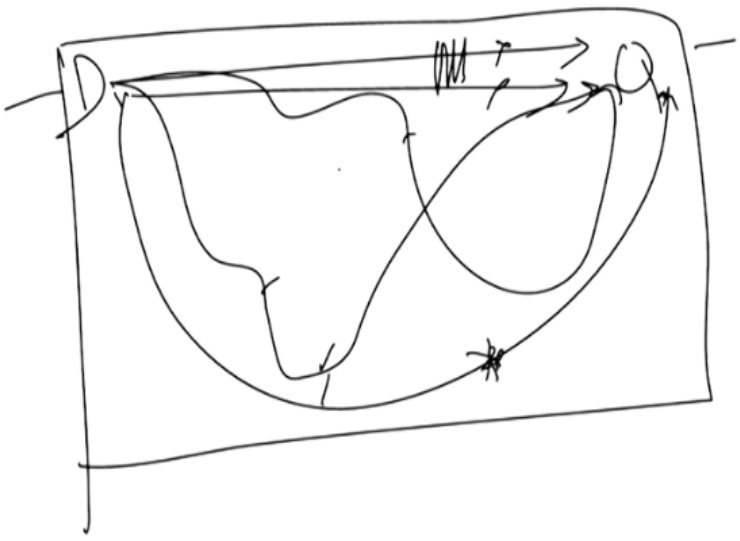
Now we can also have dual edge triggers I will not discuss that in details but dual edge trigger means your you are sampling like positive as well as negative if you want to increase the data rates you can make it dual edge trigger but then at each point you have to maintain certain things

which you should be very careful about right. So just to give you sorry so just to give you a brief idea about what I was talking about I discuss with you just now the set time right.

Now let me discuss with you what is known as whole time when my clock is going from 0 to 1 or the rising edge of the clock my data should remains stable to either 1 or 0 at least some time some minimum time before the rising edge of the clock that is defined as the set up time fine what is whole time just off reverse that means after the clock as passed you require to whole the data till some amount of time so that the sub sequent stage is able to accept the data that you are output is basically latched to particular value or fixed to a particular value right.

So we define whole time as the data input which must valid remain after the clock edge so setup time is just before the clock edge when your data should be stable and whole time is basically the data input after the clock edge when your data should be stable fine. So these are the two very important definitions as far as set up time the whole time issues are concerned in a (()) (16:26) design. Now we define the clock period as capital T which means that this is the time maximum time after sequential circuit or a (()) (16:39) operates such that it allows for the longest delay of any stage in the network which means that I will just show you a give you hint.

(Refer Slide Time: 16:50)



Let us suppose I have this network right and I have this network this is my D and this is my Q so D is the data input and Q is the output so what there is a single path which goes like this there are multiple path from here again go like this and then I have multiple paths which is coming like

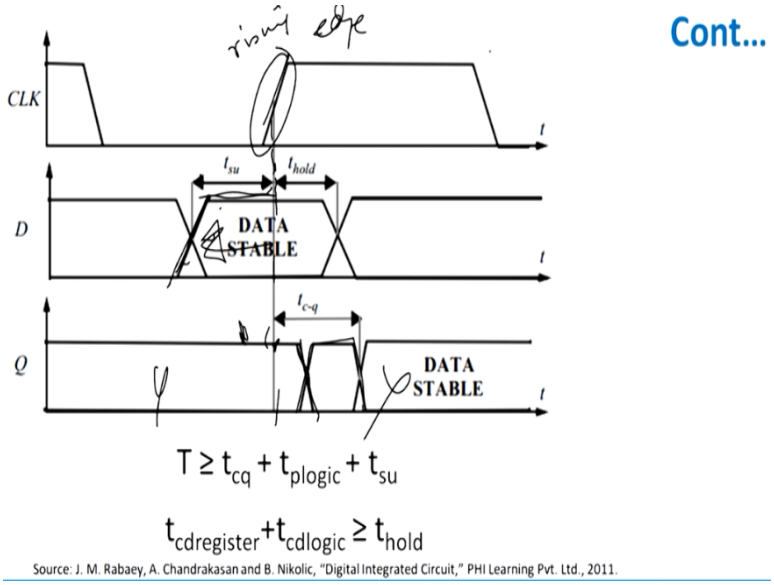
this and then it comes like this and so on hence so forth so obviously if anything not done this will have the maximum delay as compared to even this straight line paths.

But suppose it is clear for a simple logical block for a simple sequential logical block these two parts will have minimum delay as compared to these there paths available to you right therefore we define the time at which the sequential circuit operates must thus accommodate the longest delay of any stage which means that the longest delay between point A and point B should be part and parcel of the clock period right and that is very important tp logic the worst case propagation delay of the logic so for example if you have an adder or if you have a simple even in NAND gate CMOS based NAND gate you will have some intrinsic delay right.

So I am trying to tell you that you should have the worst in propagation delay available to you as far as designing is concerned there are certain delay known as contamination delay which is TCD and this is the minimum delay at which you will start getting the output available to you right. So TCD is the minimum delay available to you TP logic is the worst propagation delay which we feel between two points clock period is between two clocks the amount of time taken long to ensure the longest delay of any stage appears on the network.

Whole is the time after the clock (()) (18:29) past which the data should able to simple and hold the data what is set up time? Set time just before the rising edge of the clock when your require to set up the data and therefore you require certain amount of initial time to do that right. So these are the few important like definition which you should be able to aware of and you should be aware of therefore the various sequences are there and various issues which are there right.

(Refer Slide Time: 18:57)



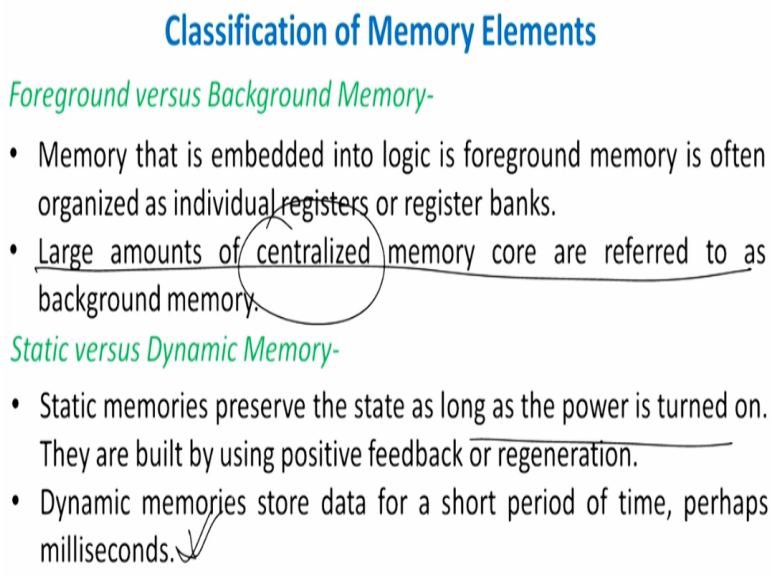
So that whatever I told you just now shown here pictorially and as you can see the clock is going so when the clock is high DNQ are high right when the clock goes low nothing happens because you are not anyway your are not anyway doing any latching it comes to this point. So this is the rising edge of next clock right so at that idea is that the one should be much a head of the rising edge of the clock and so what I am trying to tell you is that when the clock therefore goes high it remains the setup.

So this is the time till which the set up unit holds it value and then this is basically t setup this value is t setup fine. I think it is clear to all of you that t setup therefore is the minimum time right before the rising edge of the clock so that is the reason the clock is been taken here so this is the rising edge available here anything before then this which is on this side right and your gate is not able to understand what delay is basically known as t setup delay right after the gate as actually evaluated it then comes the t hold.

So t hold will also be coming into the picture right so t hold is basically the when the clock has past the amount of time taken for the circuit again to stabilize the basically for the full time circuitry fine and as you can see here you have the rising edge of the clock your data is stable here and therefore the data is equals to 1 which you see here so data equals to 1 even before that it holds one data here as it reaches the falling edge of the clock the output voltage starts to fall down here right it falls down here.

So I if do a rising I will get a rising value of the voltage at these points right so data is stable at this and at this data is quite stable right and that is quite interesting as far as this stable is concerned.

(Refer Slide Time: 21:00)



Now we have understood therefore what is the resistor and how does it work we will just check out may be one transmission gate based tg to explain to how it works in a transistor format. Memory is if you remember that is embedded into logic with logic in the for ground logic. So logic is in the fore ground and memory is in the back ground and is considered to be one of the most robust designs techniques for any digital system design however you remember you are doing a static memory and static design and dynamic design this is static design.

What was the static design you use simple CMOS transistors in the pull up and pull down with the capacitive loading and you told that well this is the combinational sequential logic meaning that this is basically logic which is sequential in nature i can solve it you can you did it possible in that domain and you will be able to do that well. But generally when you talk about design memory is always at the back side front end is obviously memory the back is basically memory which is there right.

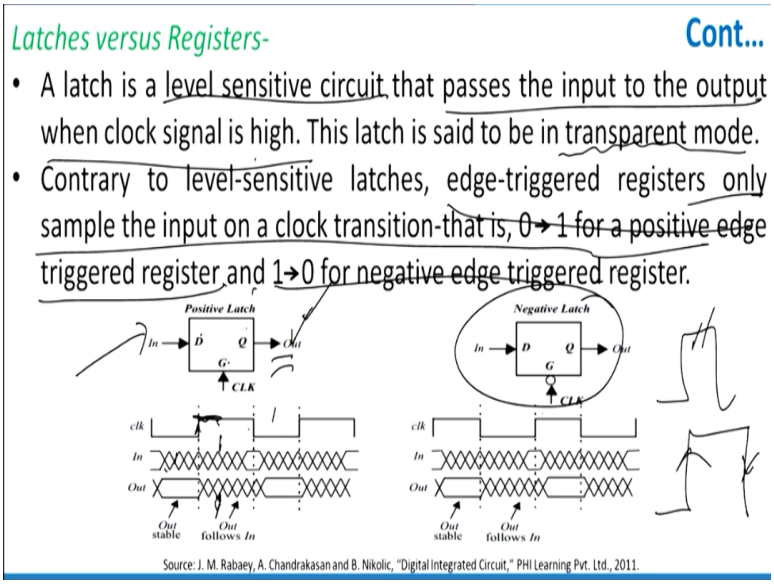
And therefore what was happening was a large amount of money was required for centralizing you memory but then if you centralize the memory you do not actually open up information and so what people are trying to do is that can we have a memory can we have a memory which as

computation but then I ensure that it is in such a state that I am not able to or any one of us is not able to steal the information or change it.

Therefore the memory is basically the background or fore ground is basically the overall system dynamics what is the (()) (22:45) system what is system static and dynamic memory will static memory preserves the state as long as the power is turned on and therefore when the power is turned off all the memories washed away in a static memory right they are generally built is using positive feedback or regeneration very straight forward and simple and but dynamic memory what is the problem is they stored the data but very short period of time right.

Perhaps milli second is what I gather but they will be smaller than that right so for a typical dynamic memory you will get a reduced value of your of your period of time then which we can evaluate the whole thing right. So this is the classification of memory right and then let me see what is the (()) (23:29) the resistance I have already discussed with you.

(Refer Slide Time: 23:32)



A latch is the level sensitive circuit so whenever level sensitive basically means this is level sensitive then edge means this is edge right so level sensitive circuit but passes the input to the output when the clock signal it is very straight forward. So when the clock is high it passes input to the ground input and output right and then in that state the latches said to be transparent mode right is known as transparent mode.

Contrary therefore level sensitive latches you have also edge trigger signals or resistors also which actually latch so if you look at this positive latch here when clock is high right so this was your input so input was varying whatever varying was taking place at g = clock so your g was given as clock and q was output here so you see just at the rising edge of the clock right out is stable right out is stable but then out has got no large amount of set time available to you sorry whole time available to you right.

And therefore you see this is follows out and this follows out in this case out follows in directly right and when the clock goes this is basically trigger so when the positive edge trigger of the clock something happened in the latch this is something sort this is a latch so when the it is the level trigger so whenever you have got output = this latch you should be able to total amount of logic here right so and hence so forth.

Let us look at negative latch is again same concept only thing is clock here is negative edge trigger and therefore in some falling edge of the clock you are able to sustain the output information available to you right whereas in this case you can sustain it the very first stage of the clock forgetting the value of the output. So as I discussed with you therefore edge trigger registers only same the input on the clock transition for 0 to 1 for positive edge triggered registers and to 0 for the negative register right.

So I am clear I hope we are trying to do is the basically both the things together on the same profile or the same net right. Next we compare a out static latches and registers right and we define to you the bi-stability principle which possibly I will discuss may be in the next turn when we come back to you and explain to you the bi-stability principle and register signals in this session okay thank you very much.