Digital Electronic Circuits
Prof. Goutam Saha
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

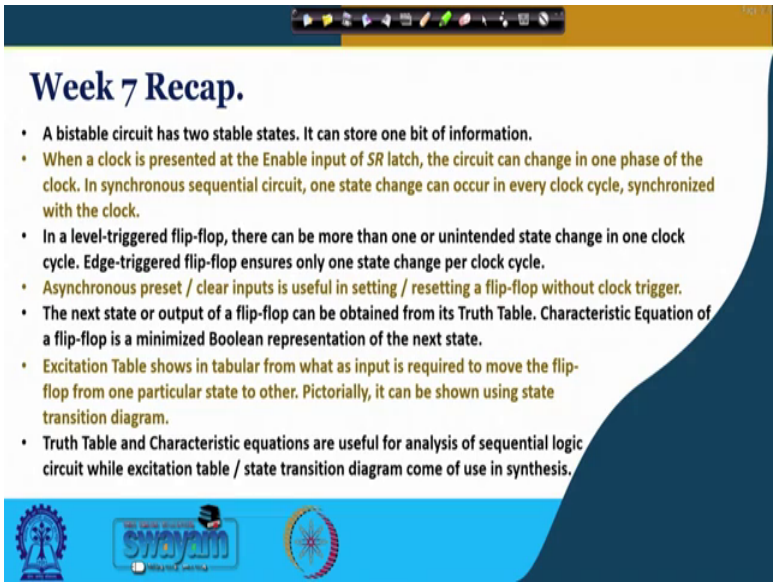
Lecture - 36
Register and Shift Register: PIPO to SISO

Hello everybody, we are in week 8 of this particular course and in this class, we shall start discussion on Register and Shift Register.

(Refer Slide Time: 00:29)

And we shall have a quick recap of what we had in the previous week and then we shall go in to the new topics.

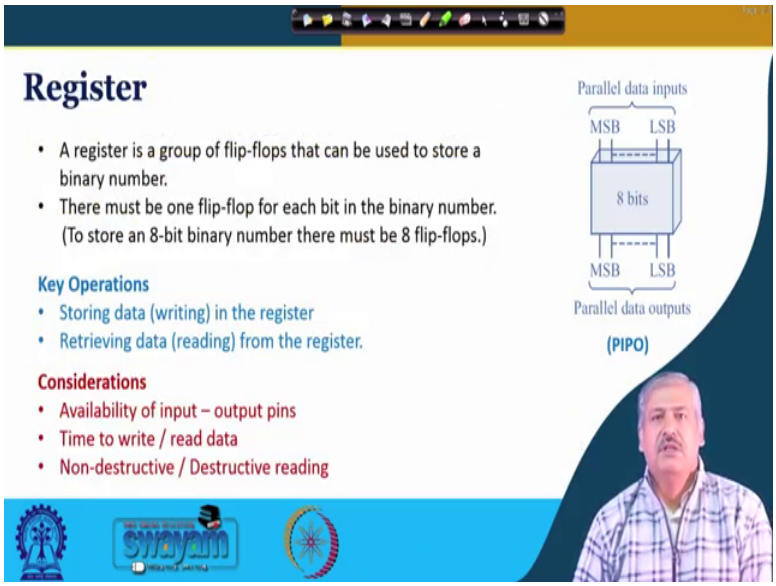
(Refer Slide Time: 00:37)



So, if you remember we discussed flip flops in the previous week, its various representations in the form of truth table, characteristic equation, excitation table, state transition diagram.

And then we understood the issues that could be there with level triggered flip flop for which edge triggered flip flop is advisable which ensures only one transition per clock cycle. And we also studied asynchronous preset and clear input, how it performs and how it changes the output or the state without requirement of a clock trigger and which is useful for initialization and various kind of you know resetting purposes. And we also looked at analysis of simple sequential logical circuit and synthesis also from the point of view point of conversion of flip flop from one type to another and mostly we do it on flip flop as an individual unit.

(Refer Slide Time: 01:55)



So, now we start discussion where we look at flip flops as a group ok. So, register is a group of flip flops that can be used to store binary data and if we think of say a binary numbers say 1001 if you want to store then we need of course, 4 flip flops ok. So, the larger the number, larger the information they will need as many number of flip flops and together we are say doing some manipulate manipulation. So, it will be a registered level manipulation that we are doing where all the flip flops within that particular group are addressed simultaneously are considered simultaneously ok.

So, for 8 bit flip flop, 8 bit register will be having 8 flip flops and the way we have seen them, if you want to write something that 8 bit number or 8 bit information and then read from it then the way we have done it before, we can think of you know providing a specific inputs to those flip flops and then reading it from the output ok.

So, directly loading it from the input side the way the flip flop truth table tells us if it is D flip flop then, it such flip flops 8 such D inputs will be there and the reading will be from the 8 q output and if you want to q bar, then q bar also can be made available. So, that is the way of you know writing and reading the way you have seen it before is called parallel data input and reading is parallel data output.

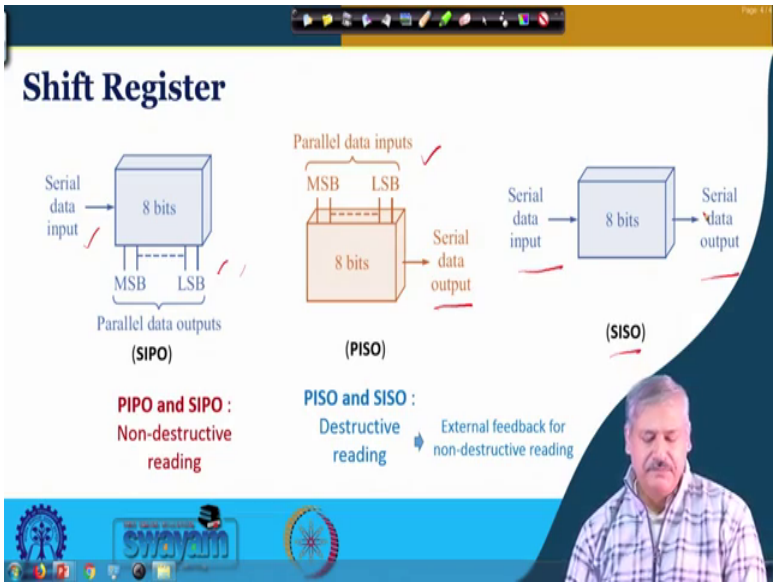
So, in short parallel input parallel output or PIPO so this we have already seen. And the data D writing is done through clock trigger of course, synchronous with D clocks. So, in one clock cycle the data is coming to the flip flops outputs and reading we can do as

many times as we want just by accessing the output values of the these flip flops these 8 flip flops is it clear. Now one issue that might come from storing large number of you know bits in the form of form of a you know group; the number of input output port or pins that will be required. So, if it is a 8 bit flip flop so you can imagine so 8 input means you will be required here then 8 output pins will be required here and if you want Q bar then of course, another 8. So, even if we do not consider Q bar so 8 plus 8, 16 then bcc ground, then clock, then usually there is asynchronous clear for initialization purpose.

So, you can consider you can think of how many you know pins will be required and standard packages you have seen 14 pin, 16 pin and all. So, this is going beyond that and so for larger number of bits where we want to store and then we retrieve these parallel in parallel output may be an issue ok. So, you have to think of alternative and when you think of those alternatives one idea that comes is to make it serially.

So, then comes the discussion or the topic of shift register ok, but then when you talk about conversion from parallel to serial, time to read or write becomes a issue and the other issue that comes into consideration is when you are you know reading it, weare pushing the data out of this you know shift register, so the data may be lost. So, all the things we shall discuss in detail in today's class and to get an idea about how to handle binary information which is of consists of larger number of bits clear ok; so, PIPO we have seen.

(Refer Slide Time: 06:29)



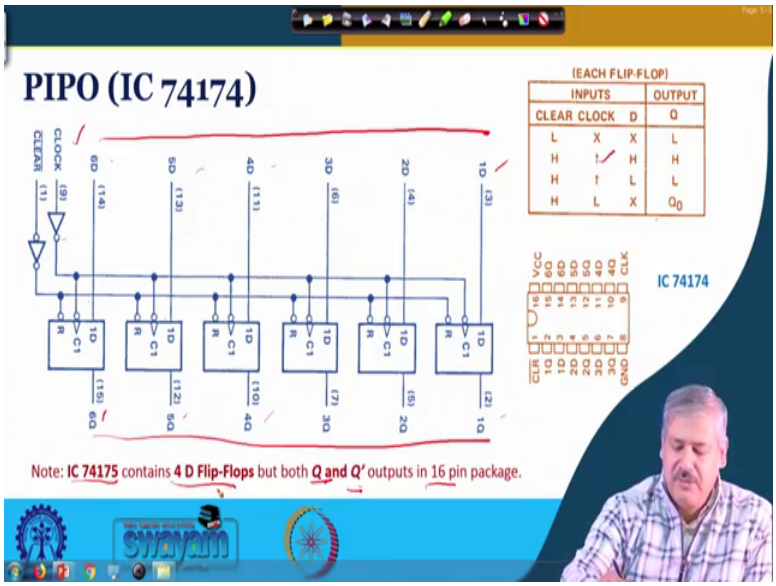
Now, let us look at three other types one is serial in parallel out ok. So, what is happening here? So, data is coming to the register serially ok. So, one bit at a time. So, to push 8 bit of information, you need 8 clock cycles ok. So, in each clock 1 data will be pushed. So, only 1 data input is sufficient serial data input is sufficient instead of those 8 that was required and reading is through parallel outputs ok.

So, then your saving on the data input size number of pins clear; so as many number of times you want to read you can read it no issue ok. So, this is one particular type ok. The other type that can be there that data is sent to the register parallel; see in one clock cycle all the data is getting loaded unlike the previous case and then the while the reading part of it the first data that is in the in most flip flop that is available and then you need another so if it is 8 bit so another 7 cycle to push data out of the flip flops when efficiency actual circuit and other things we shall discuss elaborately later with examples.

So, that will be required and at that time, since the data is being pushed, so these data as such in on its own will be lost; so that becomes a destructive reading. And the third option is the fourth option here other than PIPO, SIPO, PISO parallel in serial out here serial in parallel out there is serial in, serial out. So, that requires a list number of pins. So, data is you know introduced in the register serially with say 8 clock cycles for 8 bits and data is read also serially using the clock. So, data will be going out one after another ok.

So, to prevent this data being lost and all so you can have a feedback kind of mechanism that we shall see later ok.

(Refer Slide Time: 09:07)



So, we said that we shall look at the actual circuit that are used ok. So, the first thing that we see is a PIPO parallel in, parallel out which we are already familiar with in one single flip flop form on one unit of flip flop that you have seen how it works. So, this is IC 7417.

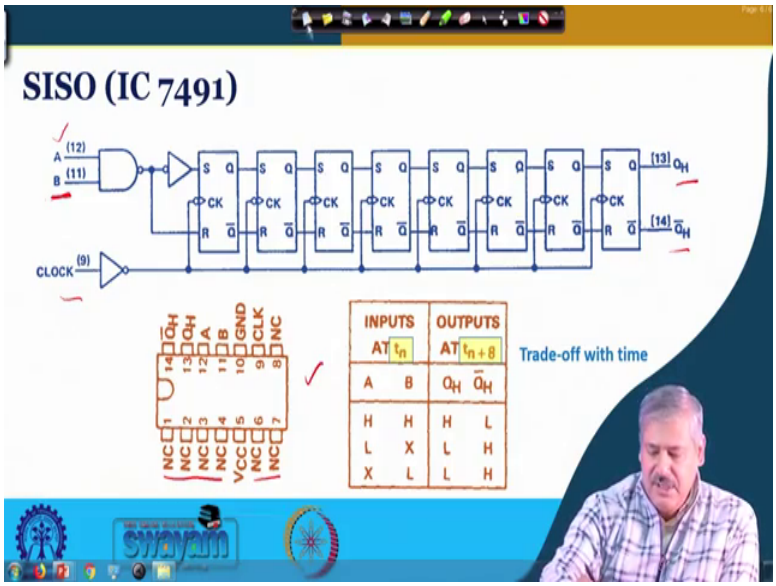
So, this is a 16 pin package you can see right and it has got 1, 2, 3, 4, 5, 6 right 6 flip flops, so 6 big data you can store here right and for that there are 6 outputs; 6 outputs and corresponding to that there are 6 inputs. So, whenever clock trigger comes, the clock is common right and if you look at this inverter here inverter here together then it is a positive s trigger. So, anywhere the clock trigger comes whatever is the data 6D here comes to 6Q directly. Similarly 5D comes to 5Q, 4D to 4Q and 1D to 1Q right.

And if you are looking for resetting it, clearing it then you can do all the flip flops together right; so, it is a common clear. So, it 2 bubbles here and then a bubble so basically it is active load ok. So, if you put a 0 right irrespective of the other values you will see that it is it will become reset is it ok. So, the number of pins as you see as many number of inputs and as many number of outputs and parallelly it is getting loaded in one clock cycle and you can do as many times of reading as you want without any issue ok.

Now, there is IC 74175 in which both Q and Q bar outputs are made available say in some application you require it and it is same 16 pin package which is similar to others 14, 16 packages that you are normally used ok. So, then since both Q and Q bar are

there, so you have to sacrifice the number of bits, number of flip flops that you can put inside the package ok; so it is a quad. So, 4 D flip flops are there in that particular IC ok. So, this is the issue with parallel in parallel out ok, but it is also useful and you can make use of it depending on your requirement.

(Refer Slide Time: 11:47)



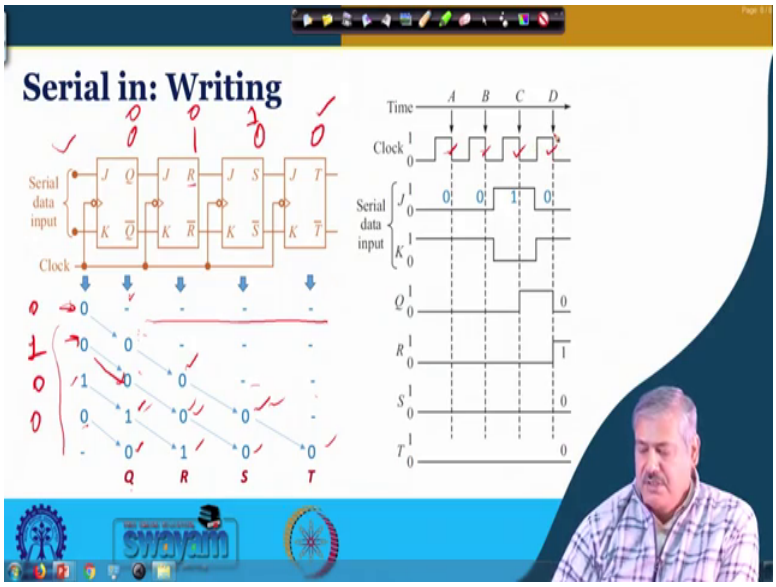
So, next is the other extend let us see we shall look at a SIPO and PISO also in subsequent classes ok. So, SISO this is again another practical circuit which is put into use so through IC 7491 that we discuss here. So, in this what you see is it is a 1, 2, 3, 4, 5, 6, 7, 8, 8 bit of information you can store it here, 8 bit of information. So, you added been parallel in parallel out you can imagine 8 plus 8 16 and so many other things that we discussed before.

But because it is a serial in serial out; so, a 14 pin package standard 14 pin package is sufficient and in that also you can see there are so many of no connection is not required NC stands for no connection ok. So, you have got a clock input common clock which get is triggering Q H and Q H bar because there are pins available so you take both the Q and Q bar as the serial out right.

And additionally you have got A and B where B you can consider as a control input. So, when B is 1, B is high then what about is the change in the A that goes to the flip flop I mean the flip flop is loaded accordingly, according to the variation in A and B is 0 then, it is all 0 that is coming over there is it fine right and you see that there is no

asynchronous reset search asynchronous clear. So, with B keep kept at 0 control input and if you feed the clock then after 8 clock cycle all of them will take the value of 0 is it fine, understood how this is organized.

(Refer Slide Time: 13:51)



Now, let us look at how this particular circuit can be used for I mean how such SISO thing can be used for writing ok. So, here we look at an example where I mean we can use any type of flip flop depending on our requirement. So, here we are using JK we shall see example with SR also, D we have just see ok.

So, in JK flip flop what we are doing we are just making sure that whatever is placed at J K will be just opposite of it ok. So, you can see in this timing diagram the serial input that we are having. So, if you have a J here 0 the corresponding K is 1 0 0 K is 1, whenever it has become 1 immediately it has become 0. So, that is ensured you can put an inverter and all at the between J and K and feed the data from J ok.

So, that is the idea. So, that is why you can do it is very simple right effectively it becomes a D flip flop right ok. Now; whatever is there in the serial data input in the beginning so we consider only the J part of it, K is just a you know inverted version of J. So, let us consider that there is a 0 present here ok. So, with one clock trigger; so these 0 goes to this place that is flip flop Q. So, we are naming it Q, R, S, T right. So, Q equates this value 0. So, that is how the arrow has been shown is it fine and there could be some value present earlier in this flip flops.

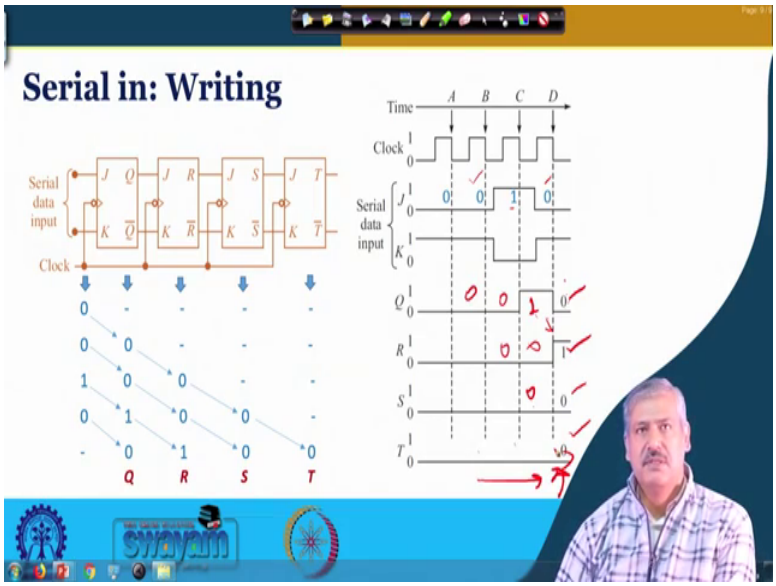
So, you do not need to you know bother about that because we have talked about writing a 4 bit data into this 4 bit shift register through serial input this is what we want to consider otherwise it could be 0 or 1 depending on whatever is there ok. So, when you complete the writing operation that data is lost of course, right. So, next value that we put is 0 at the input clock trigger comes what will happen? This 0 come to Q and this Q output is connected to input of R. So, this 0 now comes to 0 that was there over here in the next time point it comes over here is it right the other two you are here bothered.

Now, you present 1 at that time with next clock trigger, 1 comes here right just and this 0 comes over here and this 0 comes over here ok. So, S now has got 0, a R has got 0 and Q has got 1 and before the next clock trigger occurs 0 is presented at the input what will happen so this 0 will come to Q, this 1 will come to R, this 0 is coming to 0 and this 0 is coming to 0. So, four clock cycles have we lost and now Q, R and S, T are having value 0 1 0 0; 0 1 0 0 right. So, if you want to store so 0 0 1 0 so you can imagine that first you have to give 0 then instead of 0 1 0 0 so if it is 0 1 0 0 kind of thing next you have to give 1, then you have to give 0, then you to give 0 ok.

So, this is the way you have to plan your entry of serial data input depending on what you want to store here is it clear and the example is given through timing diagram right. So, what is of importance here is the clock edge right. So, it is a negative edge triggered that you can see this is the negative edge at which it triggers. So, at that time whatever is the value that is present at for J so here it is 0.

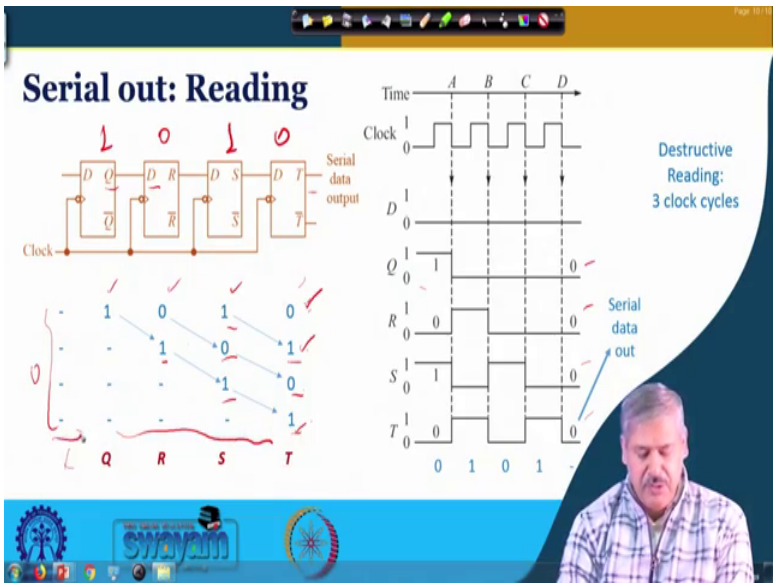
So, Q Q, R, S, T initially it has been considered it is 0. So, this 0 will be coming here ok. This is the one that were talking about then its value is 0 here before the clock trigger. So, this 0 is coming here and this 0 is coming there it is through the timing diagram same thing ok.

(Refer Slide Time: 18:37)



Now, before the next clock trigger you see that it has changed and when the clock trigger comes it is 1 ok. So, this 1 comes here and this 0 comes here and this 0 comes here in the desk clock in time period. So, this is way the time is you know changing right and after the before the next clock trigger 0 is there ok. So, this 0 comes here is already mentioned. So, this 1 comes here, this 1 ok, this 0 and this 0. So, ultimately 0 1 0 0 at this time point we shall see in this clock cycle after 1 2 3 4, 4 clock trigger that is how the writing is done is it fine.

(Refer Slide Time: 19:43)



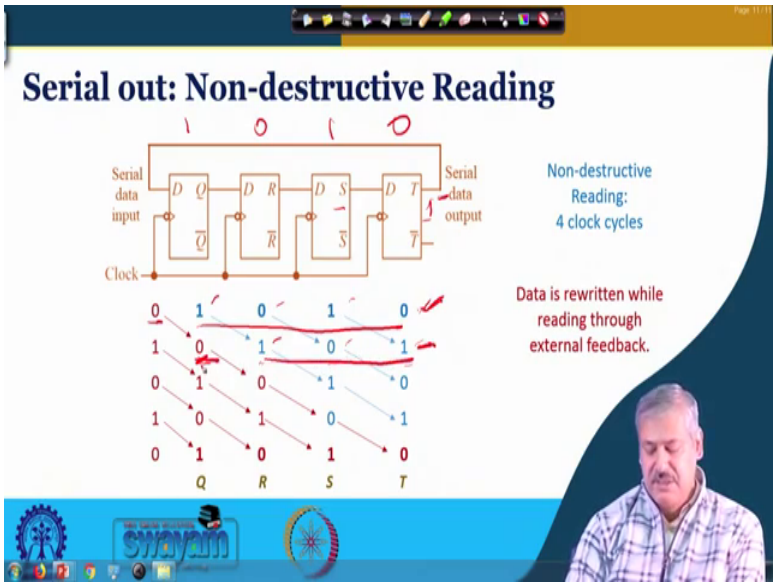
So, now we look at reading of it ok. So, you consider that initially it has been stored with 1 0 1 0 and this is the serial data out ok. So, 1 0 1 0 this is with which the flip flop is occupied with and then one clock trigger comes. So, at the inputs side it could be anything whatever is there that would be post if it is 0 or 1 it will be pushed ok, but we are just doing the reading operation and as we know that the way if we just read it this way something else will come here which is not the actual data so that is that is why it becomes a destructive reading ok. So, in the first clock pulse this 1 will be coming over here, this Q is the input to this flip flop ok. So, then now this is a D flip flop based shift register. So, this 1 comes here and this 0 will come over here and this 1 will come over there ok.

So, in the first before the application of the clock as it is you can read 0 and then after application of one clock trigger you can read 1 the information there then one more clock cycle this one will come here and this 0 will come here you can read 0 one here at this signal data out at the output of T and finally, one more clocks trigger; you can see 1 over here right. So, this way in this case since already one of the output was available accessible through serial data out ok. So, three more clock cycle you have got 0 1 0 1 all the bits read by this, but in this process now the flip flop except this one which is having the value of Q rest of the values are you can say garbage or you can say some something else, but not the one that your read ok.

So, this is by this you are actually having a destructive reading so that means, the data is lost and this is shown through the timing diagram ok. So, in this case this 1 0 1 0 you can see this is you and if you have if you are inputting you know 0 then at the input side then the finally, the values will become all 0, if this was all 0 the inputted over here then one more clock cycle all 0's will be coming and the serial data out that you read over here is like this is it 1010.

So, first you will read 0 here, then 1, then 0 and then 1 and if you continue to give 0 so 0 will come here.

(Refer Slide Time: 22:51)



Now, if you are interested in reading the data multiple times not that once you read and the data is lost ok. So, you may need in an application that you want to read the data multiple times it is not that you write ones that usually is the case write ones and by read many times then you need to see that for parallel in parallel out there was no issue isn't it.

So, parallel out you can read it has many times so now, no data data is getting lost, unless you are reloading it ok, but in serial out we have seen that while you are reading the data from the output terminal what is been shifted either 0 or 1 or any other value, so original data is getting lost ok.

So, one way to prevent that and affect multiple time reading is to use an external feedback ok; so, the one that you see as a connection like this ok. So, what happens in this particular case? What you are doing actually is every time you are reading something except for the first case, first case you know you are reading it just like first bit. So, whenever your reading the next bit say S you are writing this bit back over here, so reading and writing are happening simultaneously and now you need you know 4 clock cycles so that you are back to the data that you are originally started with ok.

So, if you had originally say 1010 you can read the way we have seen in the previous examples so 0101 because you are starting with 0 only ok. So, you can complete the reading what is there in three clock cycle, but then will be having 1 over here and you

want to get back the number in its original form then you need one more clock cycle let us see how it happens ok. So, this is the original one 1010 with which you start the reading process as well as simultaneous writing through a feedback and at the since this is 0, this 0 is feedback; this 0 is feedback here ok.

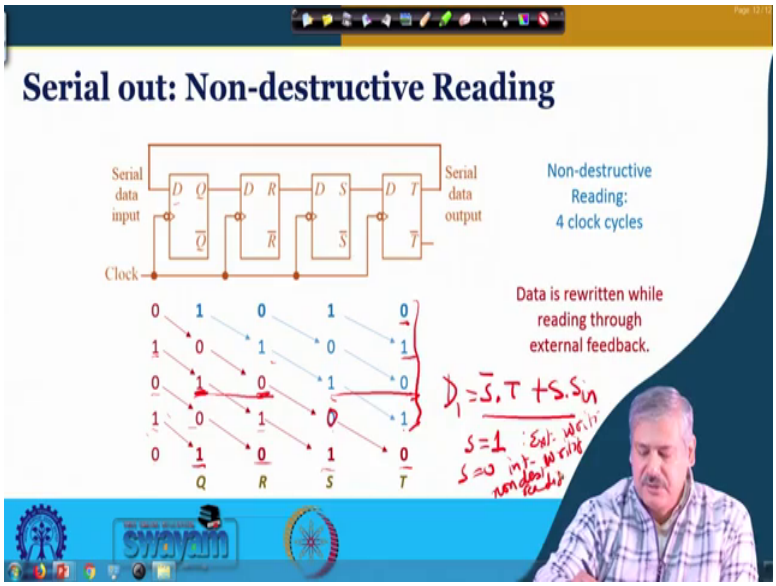
So, with next clock trigger what happens? This 1 comes over here, this 0 comes over here, this 1 comes over here and you can read this as 1 like the previous case ok. But at that time what you see because of the feedback this 0 information that was in T now is getting retained and it comes to Q ok. So, information of T 0 as 0 is not lost, it is coming back now you have got 1 0 1 and 0 here earlier this was 1010; now this is 1010 and this is it another clock at that time this 1 comes over here next clock.

So, this 0 will be post here ok, this 1 will be pushed here right and then this 1 comes to this place at Q and R gets the 0 ok. So, right now this is 1 0 of the previous one and this is 1 0 which is written back and this 0 is coming over here so again one more clock cycle this 1 will be coming and this is this will be in your red color so this is 0 coming here and this is 1 and this is 0 is it.

And this 1 is feedback here so earlier in D clock cycle this reading was complete 0's originally available at 1 0 1 rate in three more clock cycle. Now because we want to read it as many number of times and you want to store the information in 1010 form one more clock trigger, so this 1 comes over here and this 1 is coming back and 1010 you get here ok. So, reading and writing is happening simultaneously. So, this is your non destructive reading in a shift register where SISO option is only available ok.

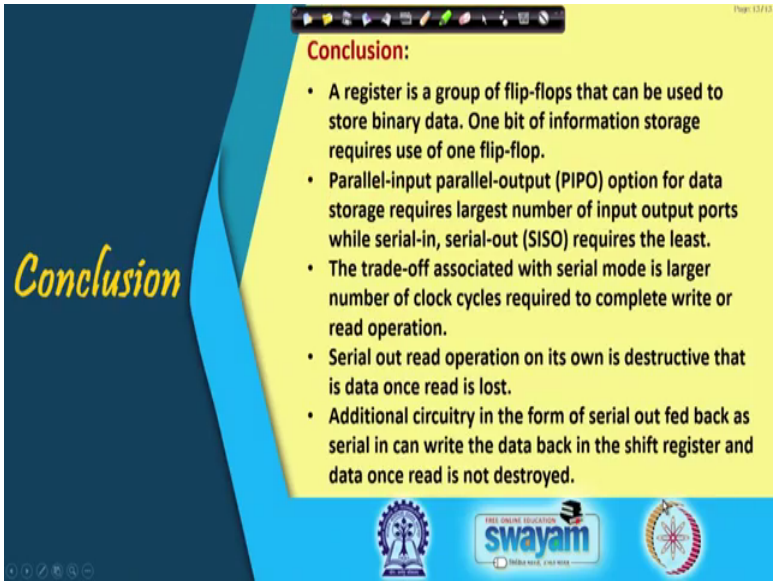
Now, if you want that you know you want to read it this way and also want to write through external inputs the way you have seen. So, would you like to disconnect or again connect this serial data input that is possible or you can think of a you know 2 2 1 multiplexer kind of mechanism by which this input D over here for the first flip flop ok.

(Refer Slide Time: 27:53)



So, D 1 is a select input. So, when it is 0 say your doing the reading so, that will be your this T plus when it is 1 you are doing the writing. So, that is the serial input ok. So, you just need to have a small circuit logic circuit over here by which if you put it in this manner then it will be external writing when S is equal to 1, so this is serial in and this is external writing and S is equal to 0 internal writing which is nothing but non destructive reading is it ok? Have you understood this part SISO, how it works and how we can make use of it and PIPO we have already seen before ok. So, with this we come to the conclusion of today's class.

(Refer Slide Time: 28:59)



So, with this we come to the conclusion of today's class. A register we have seen that is a group of flip flop that can be used to store binary data one bit of information storage requires use of one flip flop. Parallel input parallel output option for data storage requires largest number of input ports while serial in serial out requires the least of the four possible varieties that you have seen. The trade of associated with serial mode is I mean using serial in serial out is larger number of like clock cycles required to complete write and read operation ok.

So, there is the trade of wherever you go from parallel to serial. Serial out read operation on its own is destructive, but if you can put additional circuitry then using feedback which performs rewriting of the data back to the shift register then the non destructive reading can be implemented ok.

Thank you.