INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

NPTEL
ONLINE CERTIFICATION COURSE

On Industrial Automation and Control

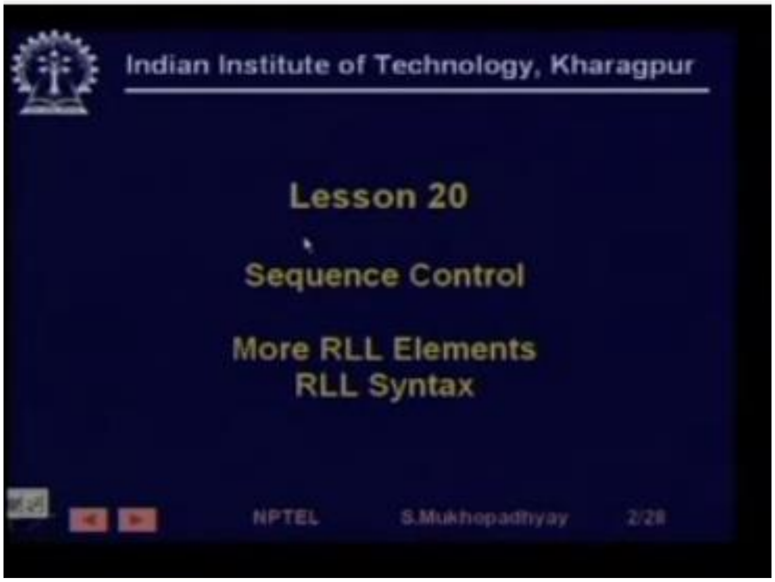
By Prof. S. Mukhopadhyay
Department of Electrical Engineering
IIT Kharagpur

Topic Lecture – 26
Sequence Control
More RLL Elements,
RLL Syntax

Welcome to today's lesson which is lesson number 20 of the course on industrial automation and control.

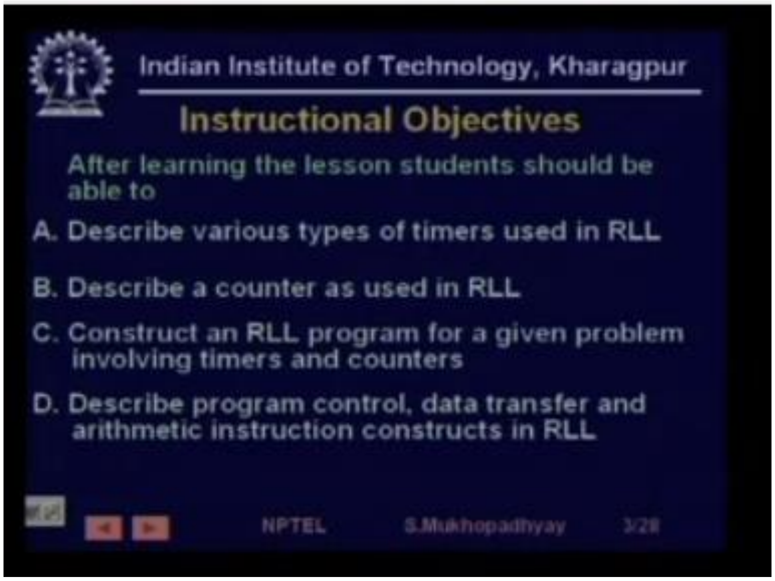
(Refer Slide Time: 00:29)

(Refer Slide Time: 00:31)



Today we are going to look at some new programming elements namely timers, counters etc which are required for RLL programming, we are going to understand their meanings and see their use in real simple but real typical industrial programs. So before we get on we take a look at the instructional objective.

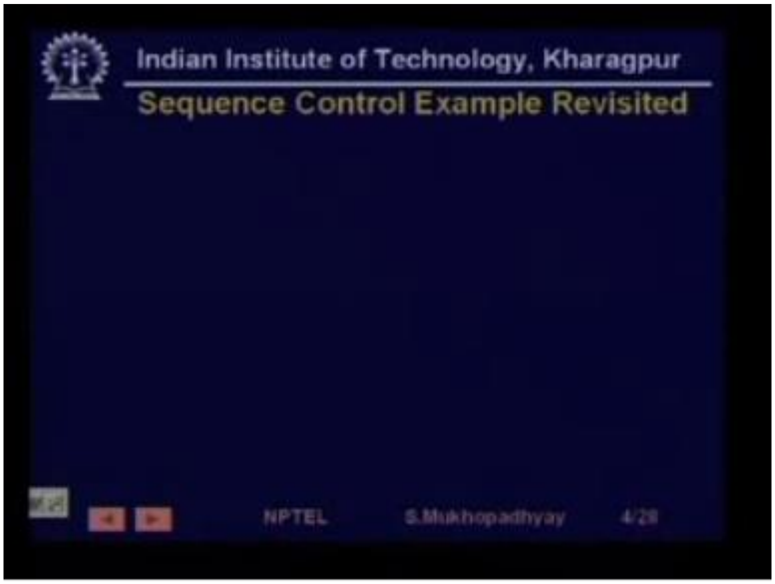
(Refer Slide Time: 00:58)



Which is to so that a student will be familiarized or he will be able to describe various types of timers used in delay ladder logic, he will be able to describe a counter, he will be able to construct RLL programs for simple problems involving these timers and counters and he will also be able to be familiar with some program control, data transfer and arithmetic instructions which are required in just like in any program you require if then else statements.

Which are program control statements, add statements you need statements for moving data from one location to the other, so here also you need such construct so for writing complete programs they are sometimes necessary so we will get familiarized with some of them.

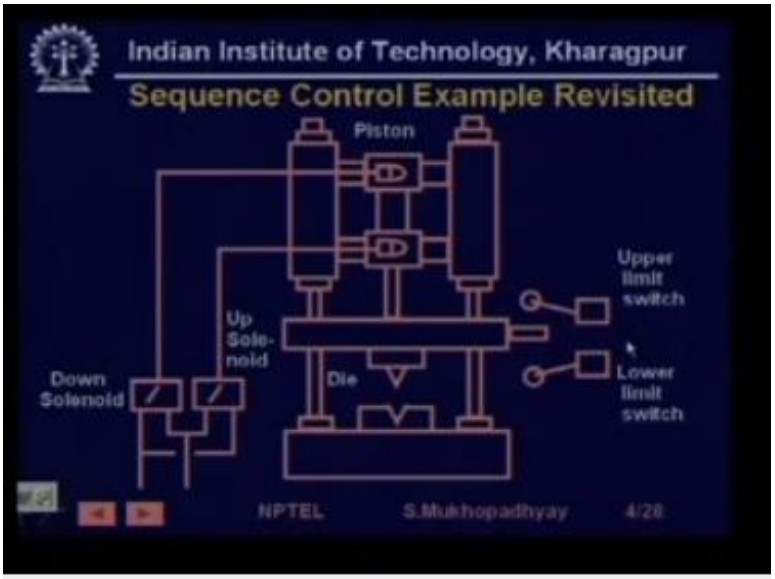
(Refer Slide Time: 01:56)



Of course we must remember in this context that there are real relay ladder logic programs are often somewhat non-standard so it is not that we are following a particular manufacturers constructs but it is very, very likely that most of these constructs will be found in each PLC manufacturers programming repair toil, so it is good to understand them in an abstract form and then if you are going to use a particular PLC.

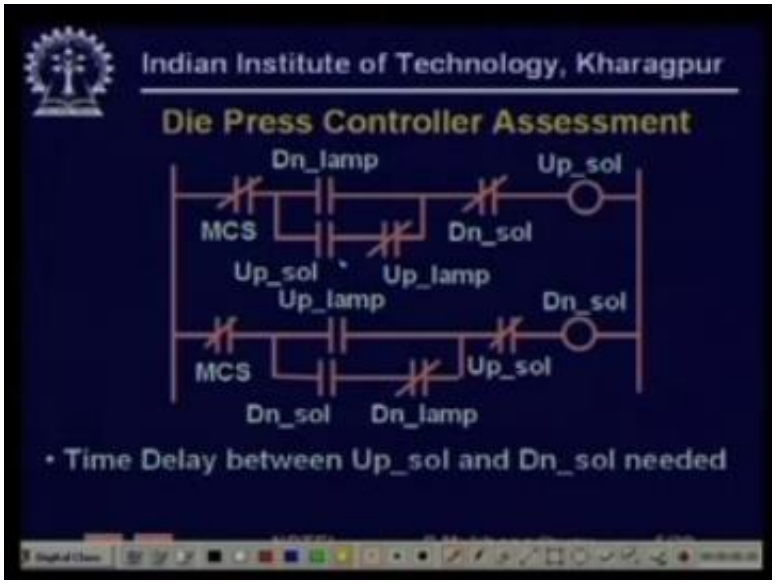
Then look up the particular manual for such constructs, so before we use timers and counters we want to motivate them, so we looked again take a second look at our previous example of the die press, so here is a die press.

(Refer Slide Time: 02:49)



Were basically there is a the piston moves the die up or down depending on whether the up solenoid or the down solenoid is activated this you know directs hydraulic power upward or downward to the piston so the die moves up or down and there are two sensors namely the upper limit switch and the lower limit switch which are which sends the positions of the end positions of the die.

(Refer Slide Time: 03:30)



So we take a look at the earlier solution which we proposed before in an earlier lesson so here what we did is here we propose a an RLL program which use only the normally on and the you know are they normally open and they normally closed contacts the real input contacts as well as some auxiliary contacts and some output coils, now what happens here is that let us say let us look at this. That initially suppose the suppose the down solenoid is on.

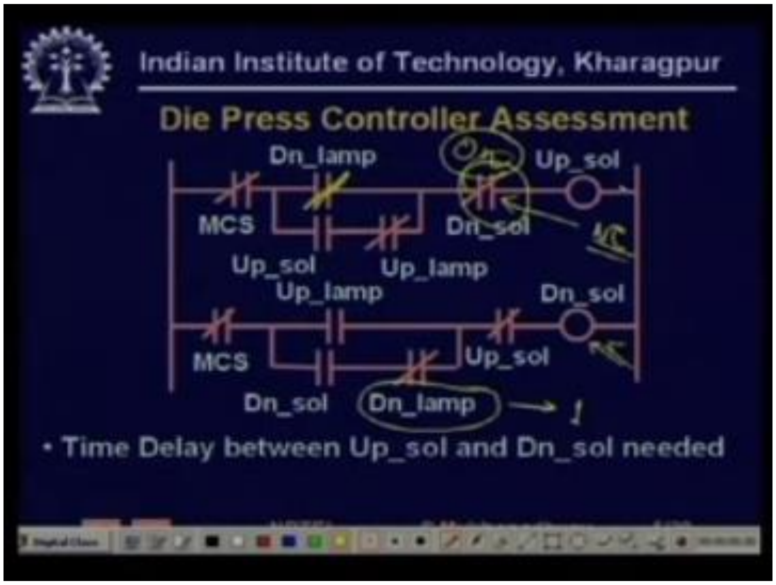
(Refer Slide Time: 04:29)

So the down solenoid is on means this is on so this is on when this is on obviously because you have an NC contact here so therefore up solenoid is off, so the die platform is coming down when it comes down it eventually makes the down limit switch.

(Refer Slide Time: 05:00)

And the down lamp will go on, so this goes to one immediately what will happen is that is that this previously the path was for that was being followed for connection was this, so this down solenoid will become off so now at this position the down solenoid is off and the up solenoid is off so the piston is slowing down there is there is no force forcing it down so it is flowing down.

(Refer Slide Time: 05:41)



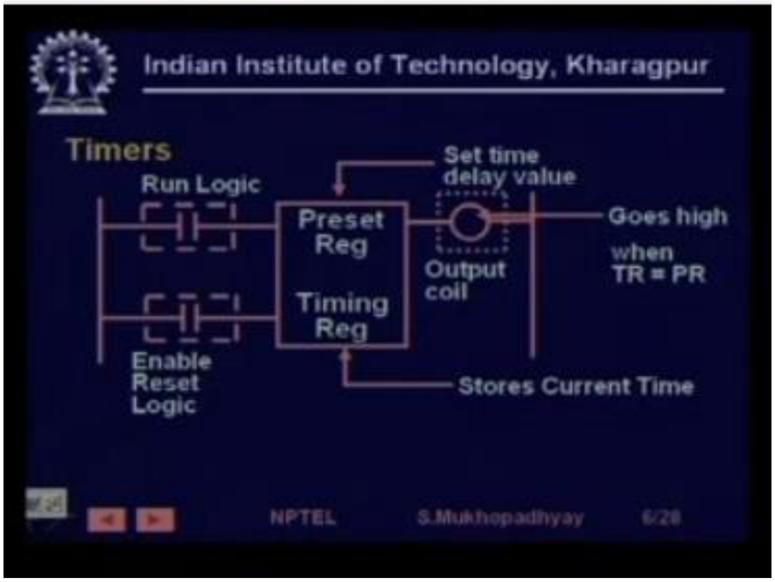
Now what happens in this program is that the moment the down lamp is so at in this position the down lamp is on and the down solenoid has become off, so now what happens here is that the down lamp is on so it becomes on while the down solenoid is now also on so this is also on because the down solenoid is off so and this is a normally closed contact because this is a normally closed contact.

So therefore when the down solenoid is off this is on so therefore immediately the up solenoid becomes on so you see that normally for a for a die praise the die has to come down on the suppose I it is a sheet metal on which you are trying to press into a particular form then you do not want that the moment it comes below immediately it goes up you want it to probably wait a little while and then go up for the next stamp so now it is a very common.

(Refer Slide Time: 06:54)

That a time delay between up solenoid and down solenoid is needed that is after the down lamp is on and by the time the up solenoid again becomes energized we want to delay, so what I wanted to say here is that such delays are very often needed in industrial operations and today we will see how to create this delays, okay. So we go on so now so as we shall see we will actually see the solution a little later. But first let us look at the timers which actually creates this delays.

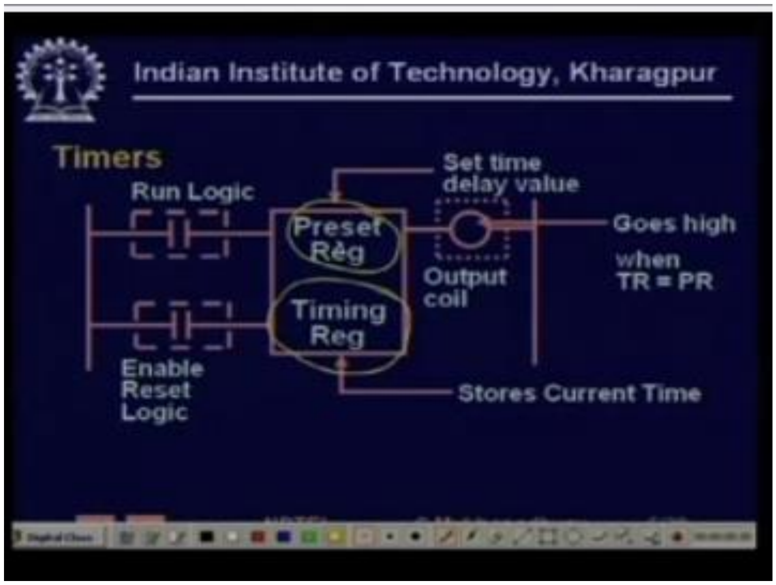
(Refer Slide Time: 07:32)



Now here I want to mention that these timer's the timers are sometimes you have we are all familiar with timers we are possibly our first interaction with introduction to timers was in the digital electronics codes, now those timers are actually hardware timers they you may you may actually use hardware timers also in a PLC in which case you have a you have separate you may have a separate timer card.

But in this case we are talking of the program so it is actually a see it is actually a piece of code which creates a delay in asserting some output, right. So the purpose of a timer is to create the delay how much delay that can be programmed number one and number two is that the timer actually you know it is the basic idea of a timer is that there are two registers.

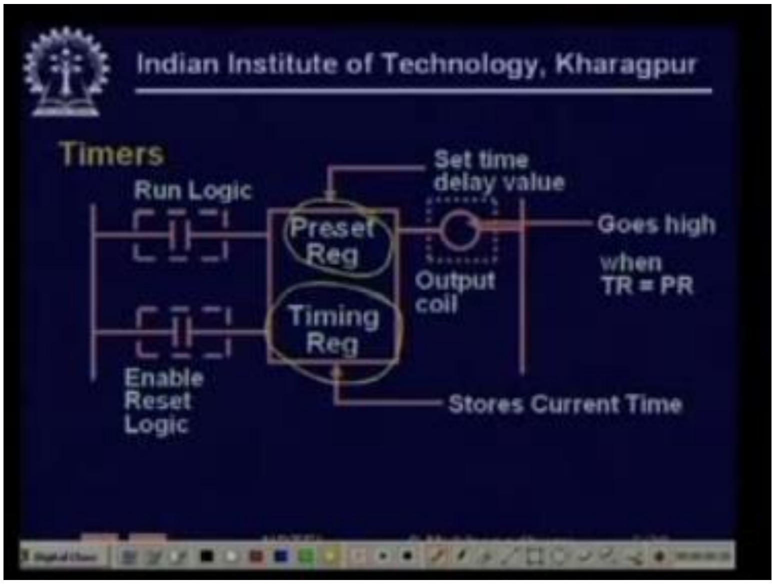
(Refer Slide Time: 08:27)



One is called the preset register and the other is called the timing register, so the preset register is actually set whenever you create a time and in the program you actually set the preset register gets loaded by a particular value and it stays fixed, while the timing register during the time that the timer is working is active the timing register keeps getting incremented using pulses from an internal clock of the PLC.

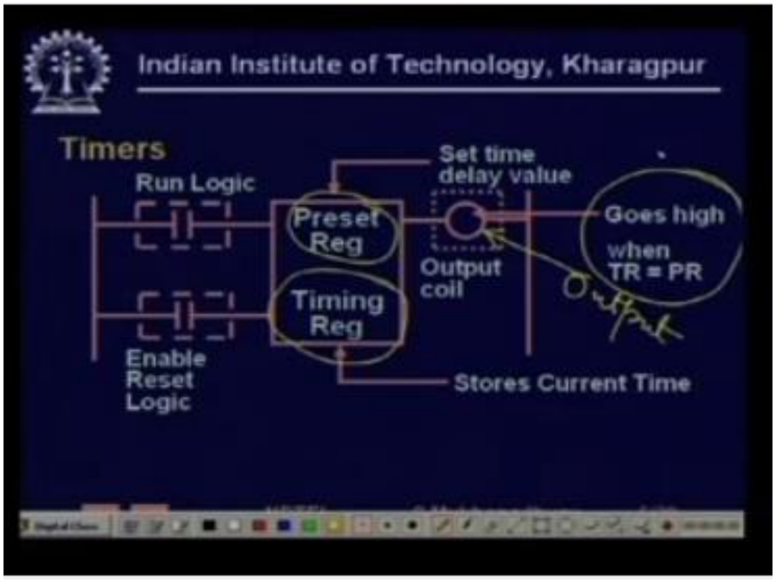
So it does not require any external clock sometimes if you may you may also for example as we shall see that counters are actually work on the extra external clock so because we want a particular timing so the general decide from the internal clock, so as the timing pulses on the clock are coming so the timing register keeps increasing.

(Refer Slide Time: 09:27)



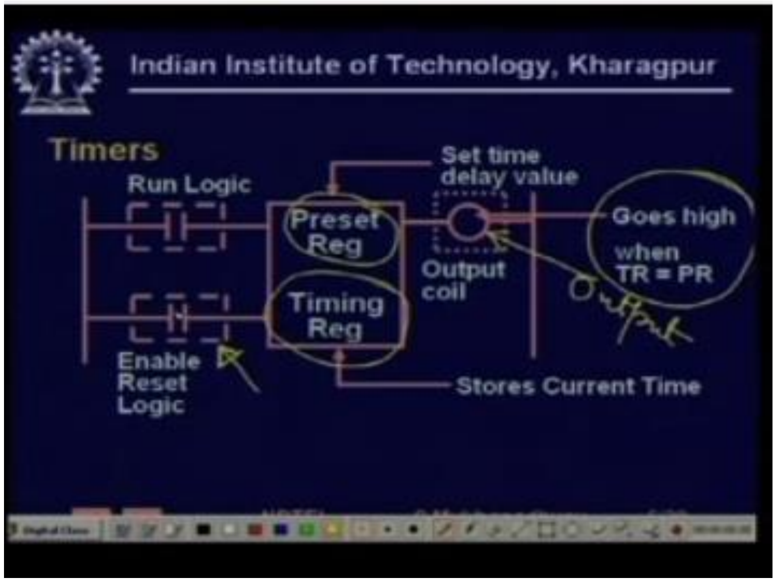
And when its value exceeds that every time there is a there is a comparison between the timing register the preset register, so after some time what will happen is that the timing register value will exceed the preset register value at which time the timer will stop timing further that is the timing register will stop incrementing and the output will be asserted.

(Refer Slide Time: 09:45)



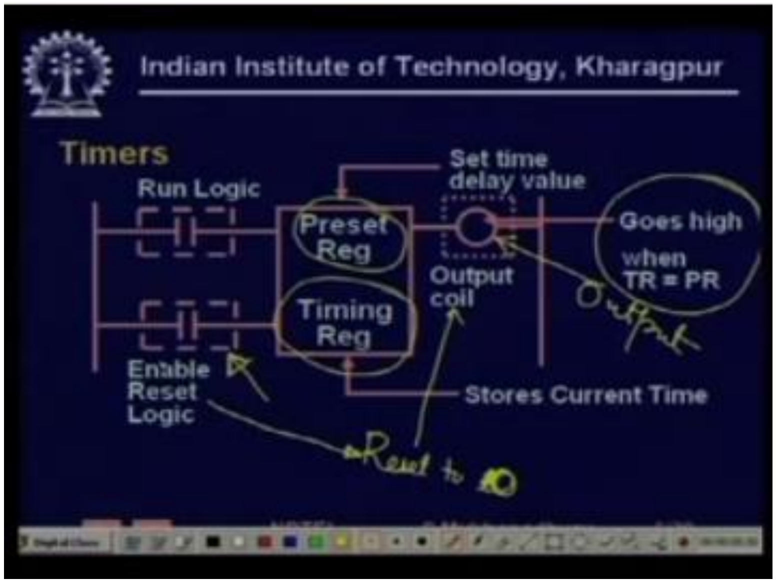
So the output will be asserted when TR = PR right. Now this thing happens when the timer is active so when is this timer active that is that can be again controlled by using two kinds of logic.

(Refer Slide Time: 10:07)



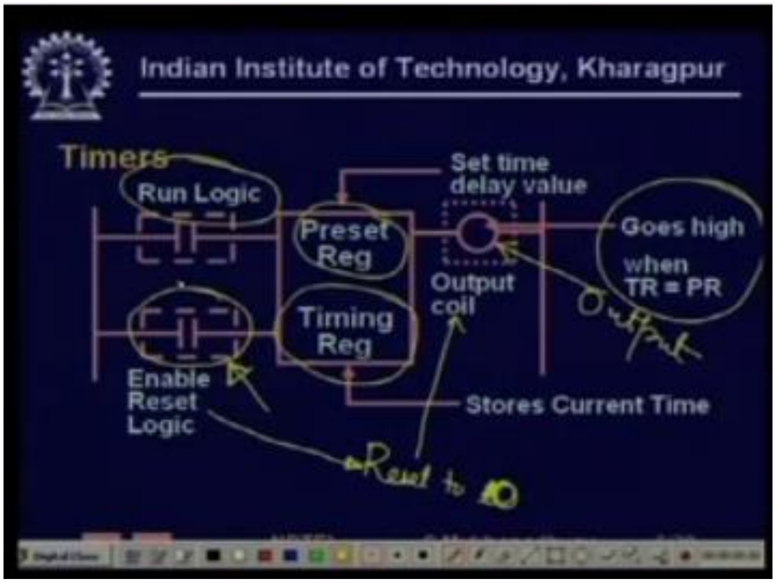
So first is called the enable or reset logic so when whenever this logic becomes zero the timer is not enabled it is inactive and the output coil is reset to1.

(Refer Slide Time: 10:22)



So this output coil reset to 1 when that is why it is called reset logic where reset to 0 I am sorry reset to 0, similarly now when it is enabled so at that time it is enabled to 2 time but whether it will actually time or not that can be again controlled by this run logic.

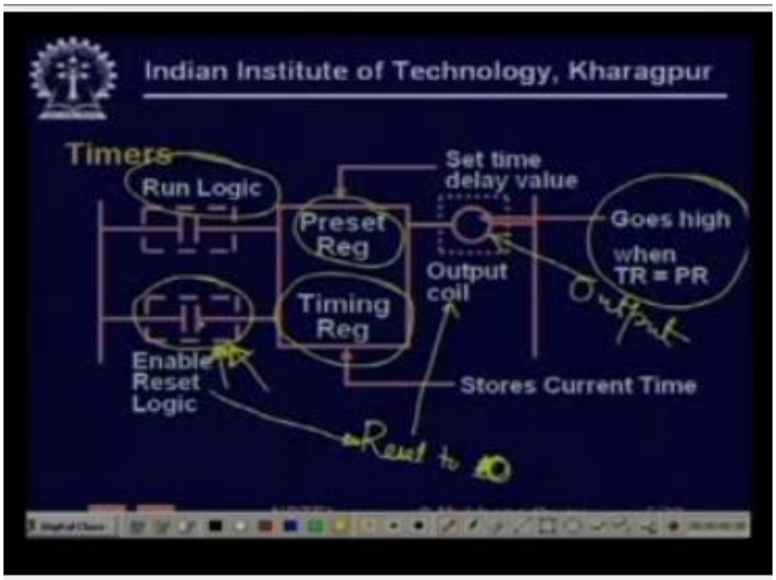
(Refer Slide Time: 10:54)



So when this run logic will be enabled at that time only the timing pulses come from the internal clock to the timing register and other times it is inhibited, okay. So I would also like to assert mention that while in this case we have mentioned we have we have actually put it by a single contact we could implement a complex logic based on which we want to enable or we want to assert the run logic.

For that we can put extra runs and the which will actually program the logic and then finally when the logic is satisfied or not that is that will be simplified by an that will be symbolized by an output coil that output coil contact we can put it here.

(Refer Slide Time: 11:40)

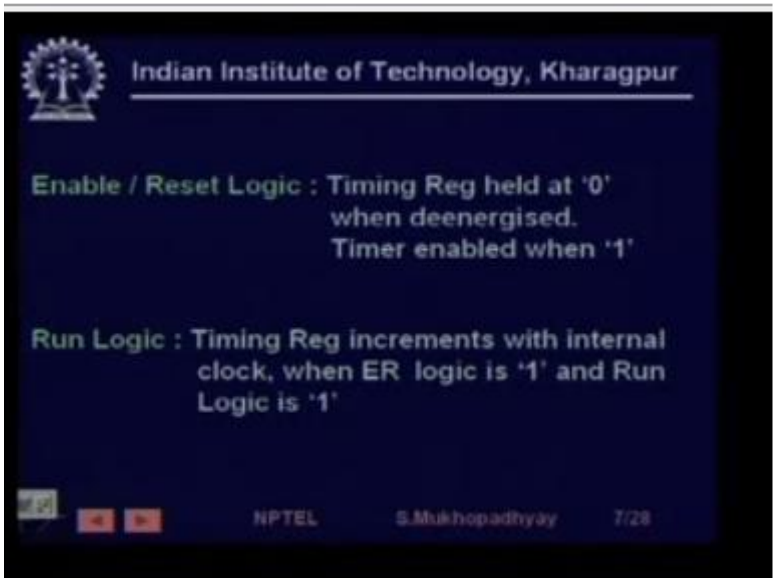


So it is it is not necessary that you have to always make it a very simple logic you can make it as complex logic as you want, so having done that so this is a basic timer. Now timers can be timers can be delay so we can have various types of delays and all of them can be realized by this by this basic timer module. So to recapitulate we have enable reset logic.

(Refer Slide Time: 12:13)

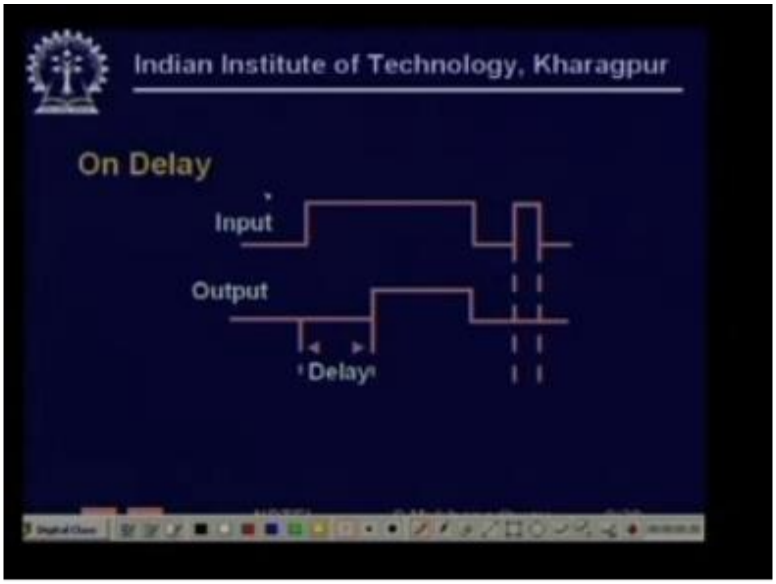
Where the timing register is held at zero when it is de-energized or zero and the timer is enabled when one, so at that time the timer is ready to receive the clock pulses and increment its values.

(Refer Slide Time: 12:28)



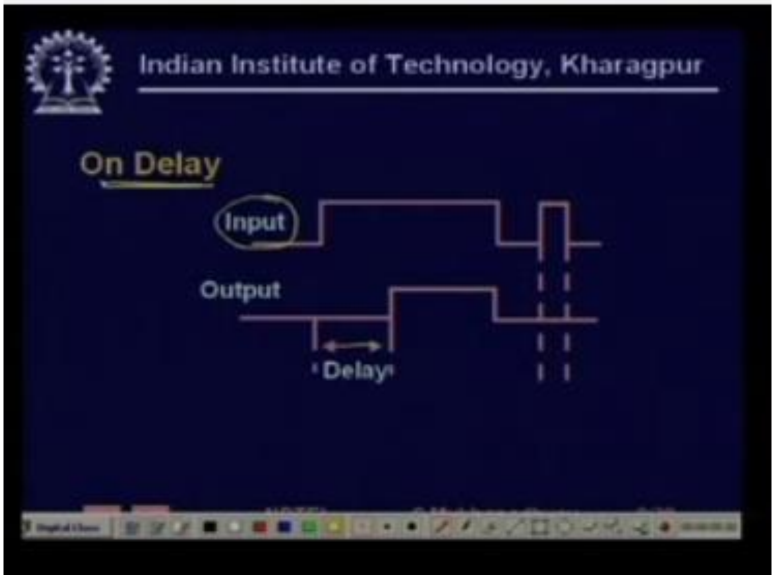
Similarly we have run logic where timing register increments with the internal clock when the enable reset logic is one that the timer is enabled and the run logic is also one just what I say, so now we take a look at the different kinds of timers first is the ON delay timer.

(Refer Slide Time: 12:44)



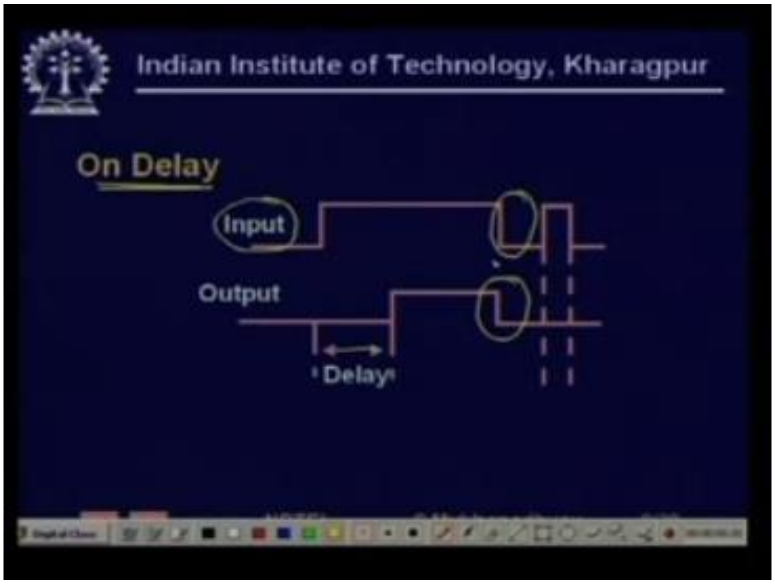
So here we are saying that if an input timer creates a delay between an input and an output.

(Refer Slide Time: 12:56)



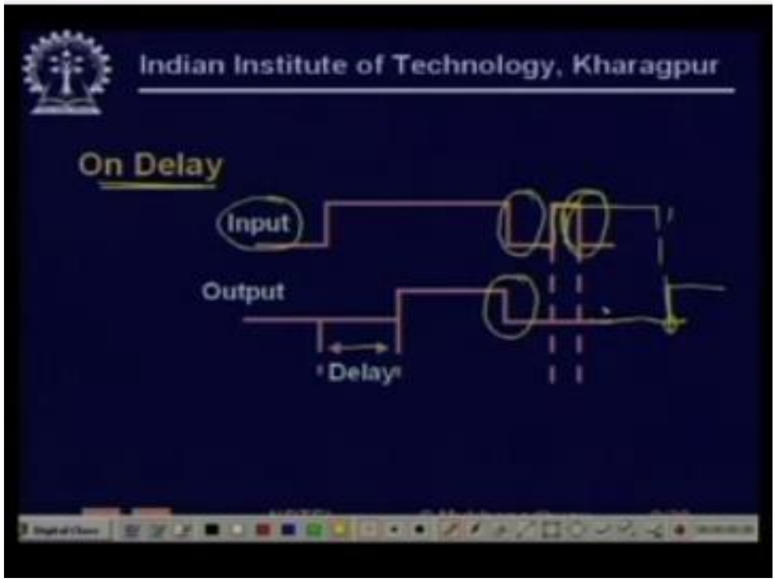
So here we say that if the input signal goes ON then the output signal will go ON after a little delay that delay that is why is called an ON delay, so while the input signal becomes ON the output signal becomes ON after a delay and but if the input signal goes OFF.

(Refer Slide Time: 13:18)



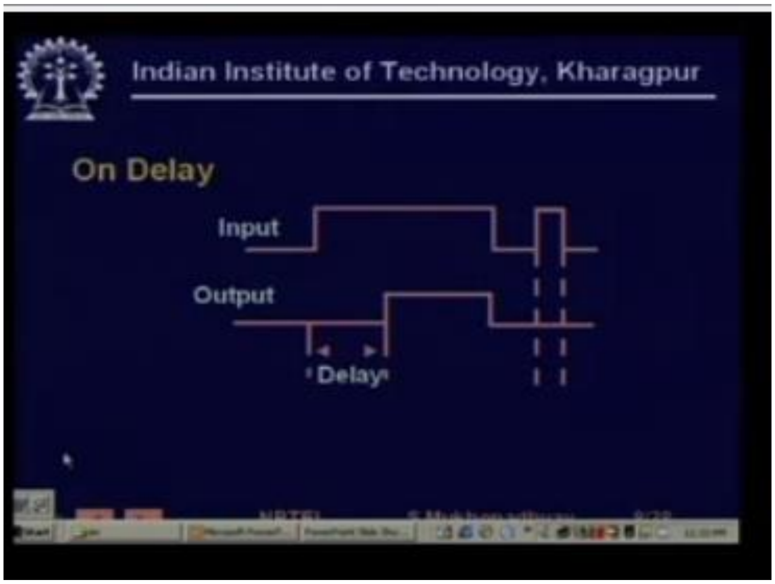
Then the output signal goes off immediately so there is no delay in getting off there is a delay in getting ON that is why it is called an ON delay timer.

(Refer Slide Time: 13:33)



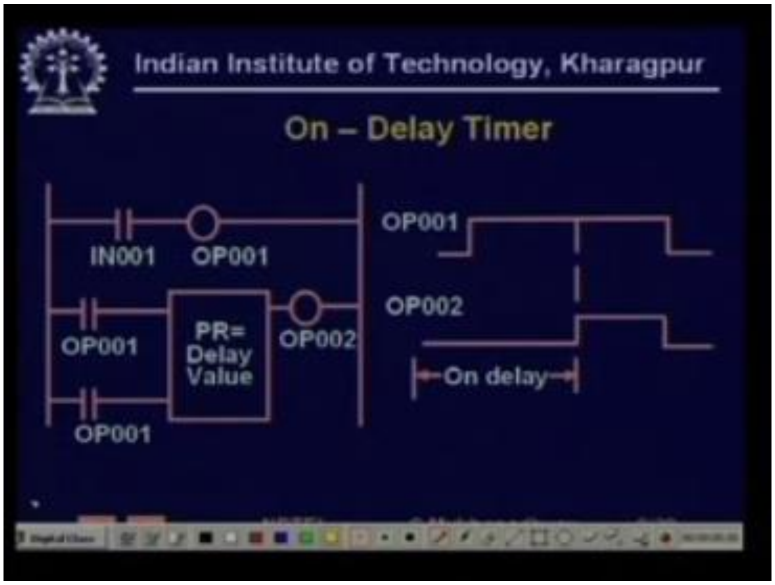
Similarly, note here that here again it becomes ON so here again it becomes ON now the timer if this ON would have persisted then the timer would have been on somewhere over here the timer would have been would have been ON because of this delay but unfortunately the input became OFF unfortunately or fortunately the input became OFF even before this delay interval could expire. So the timer also became off and that delay value got erased so the timer never went ON.

(Refer Slide Time: 14:29)



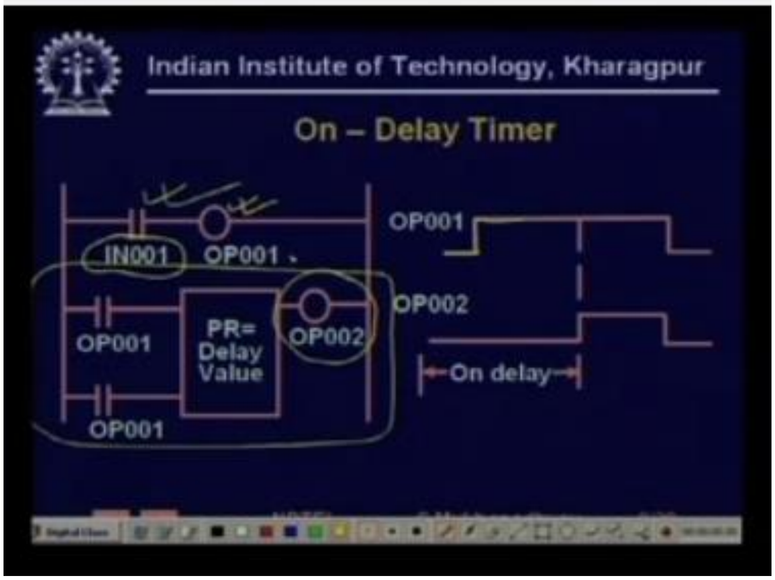
Similarly now we will see that we can easily realize this on delay timer using our old timer.

(Refer Slide Time: 14:43)



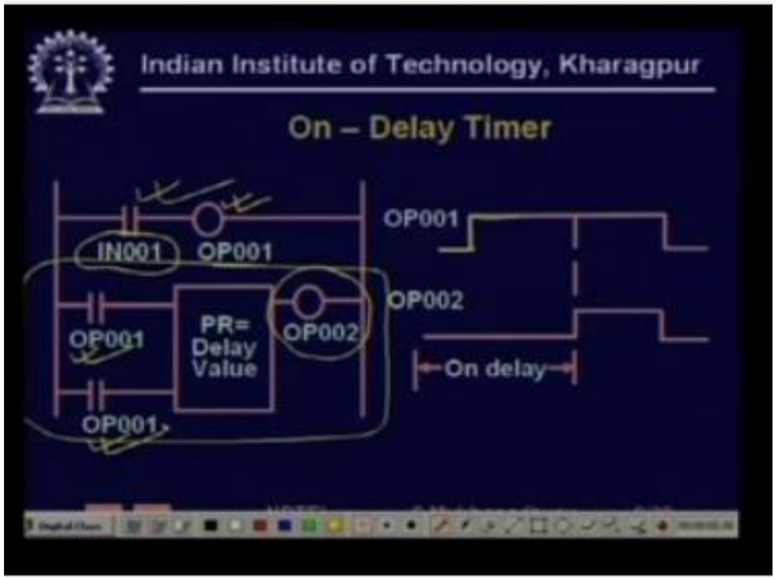
So now here is the circuit here is a RLL logic circuit or as number of runs which creates which takes a basic timer unit this is the basic timer unit that we have seen.

(Refer Slide Time: 14:58)



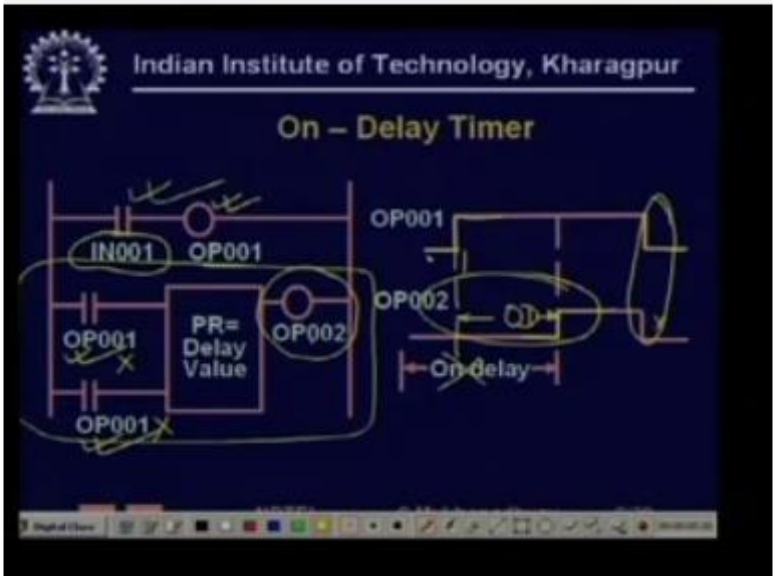
Takes it and couples it with another run and makes anon delay timer how, so suppose we pressed we asserted this input and we want that so immediately when you asserted this output goes high so immediately when we asserted this output goes high but we want that the other output hat is OP002 actually go high after a short delay, so what happens is that this way whenever these.

(Refer Slide Time: 15:41)



PO001 goes on so you see that the run logic is enabled and the enable reset logic is also enabled which means that the timer is typing is timing now.

(Refer Slide Time: 15:55)



So and when the delay when the so the timing instance is getting incremented with the internal clock pulses and when it will cross the preset register at that time the OP002 signal will go up, so you have this is wrong, so you have created a delay what is the ON delay now, so the on delay is this much. So this is the ON delay, on the other hand see the moment this becomes off so the moment this becomes off.

Immediately these two become off and immediately the OP002 also becomes off, so there is no delay in getting off while there is some delay in getting on that is why it is called an on delay timer. So let us see the other kinds of timers, now before doing that let us now so we want an ON delay let us use this on delay timer now in our die press example. See whether we can introduce a delay between the for the bit that is after the down lamp goes on. We want that the up solenoid will go on after an on delay, right. So we do that.

(Refer Slide Time: 17:31)

So here is our, you know our earlier the first two runs are very similar to our earlier solution except for this one.

(Refer Slide Time: 17:39)

I am sorry so except for this one, right. So now what is happening is that suppose the down solenoid is the down solenoid was on, so actually the connection was coming like this at this point the down lamp the down lamp suddenly became on the down lamp became on because the die hit the limit switch down limit switch.

(Refer Slide Time: 18:30)

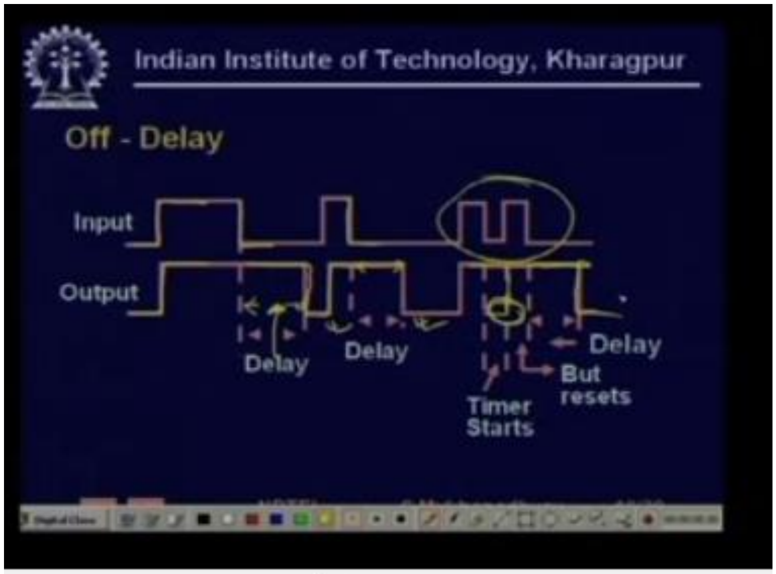
So the moment the down lamp becomes on so this level off and it and then the down solenoid now goes down, now when that now look at this so initially the OP002 is off so therefore this is off and MCS this is the master switch this also becomes off so initially supply was coming up to this, now the down lamp has become on. So immediately OP001 becomes on when OP001 becomes on the timer is enabled and its timing.

So after a delay this OP002 goes on becomes on when this becomes on now there is a direct path to this and up solenoid becomes on and when up solenoid becomes on so then that is then that then the usual operation starts so what we have demonstrated is that we have put a timer run and now there is a time delay.

(Refer Slide Time: 19:50)

Which can be set by the value of the preset register that we set here between the down lamp coming on and the up solenoid coming on, so that is what we have achieved. So now we look at the other different types of timers for example off delay.

(Refer Slide Time: 20:12)

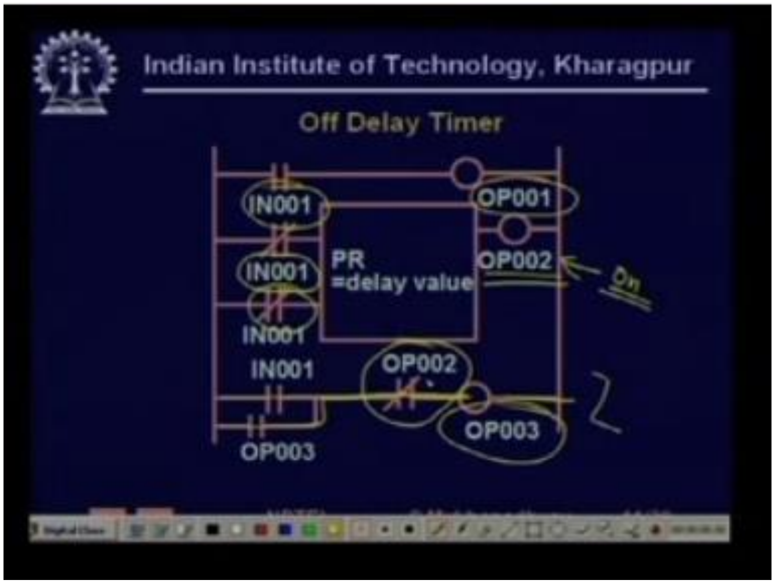


So the object timer is exactly like the ON delay timer except for the fact that now the delay is getting off so the moment the input becomes on the output also becomes on, no delay but when the input becomes off the output becomes off after a certain delay. So this is the delay this is the delay, okay. And the same phenomenon is actually observed that if this becomes on again for short pulse.

So this also becomes on then when it becomes off when it becomes off here this delay starts here but before the delay and then after this delay this becomes off but in this case before this it can this delay can expire there is another on so it will again become on, so this cannot fall because the delay is not expired so it continues and then at the end of this again after delay it becomes off.

So basically the same but just the Just a very similar operation with on delay but only applicable in this case when the input goes from on to off. So how do you realize this one, so let us simple to realize.

(Refer Slide Time: 21:54)

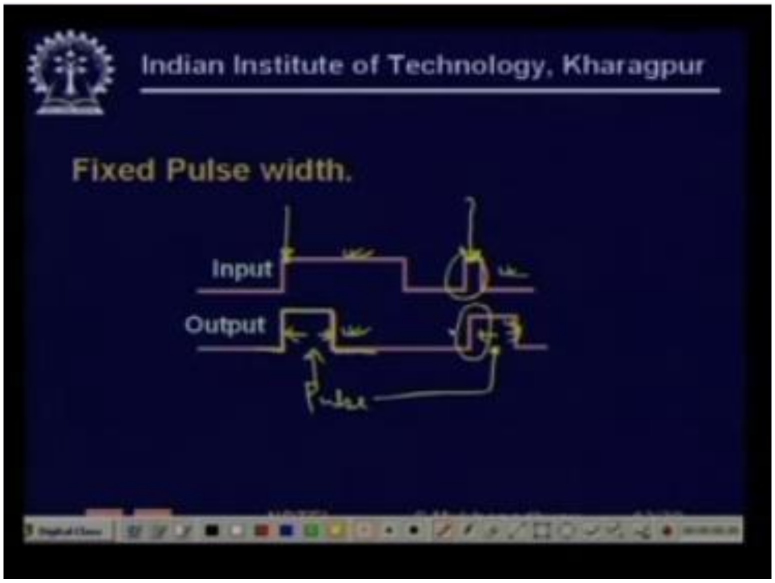


See that now we OFF delay, so again this is the input goes on so immediately OP001 goes on when OP001 goes on you see IN001 this is already off in the reset position so OP003 immediately goes on, this is my final output. So there is no delay in getting on now suppose and when OP003 goes off we can we can use this one so it latches, now imagine that IN001 goes off so when it goes off this OP001.

When this goes off now these are NC contacts so when all the time when this was on these this OP002 will held to 0. Now when it will be 0 then immediately the timer will start timing and after some time this will become go high become on, when this becomes on immediately when this becomes on this becomes off and OP003 falls. Now so there is a delay in getting OB 003 off. So this is a very simple realization of the off delay timer again using the basic timer construct that we have seen.

Similarly we could have various kinds of timers for example we could have a fixed pulse with timer where every time the input becomes on after there is respective of the delay a fixed pulse from the output.

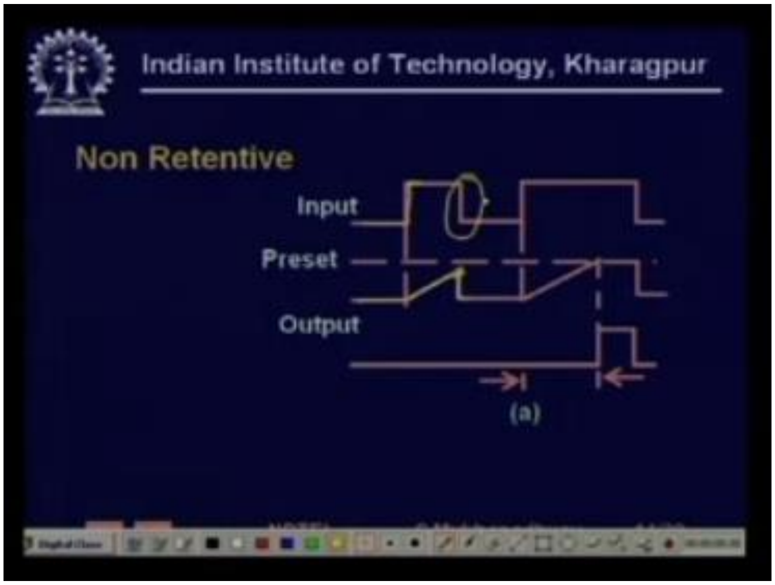
(Refer Slide Time: 23:45)



So here the this the this is just like an off delay timer yes no, not off delay timer this is that every time the when the input goes high the output immediately goes high and it is an respective of the input it stays for only for a fixed time and then comes down, so here also when this goes high this goes high and even if this comes down much earlier this keeps so here it does not come down but still this comes down.

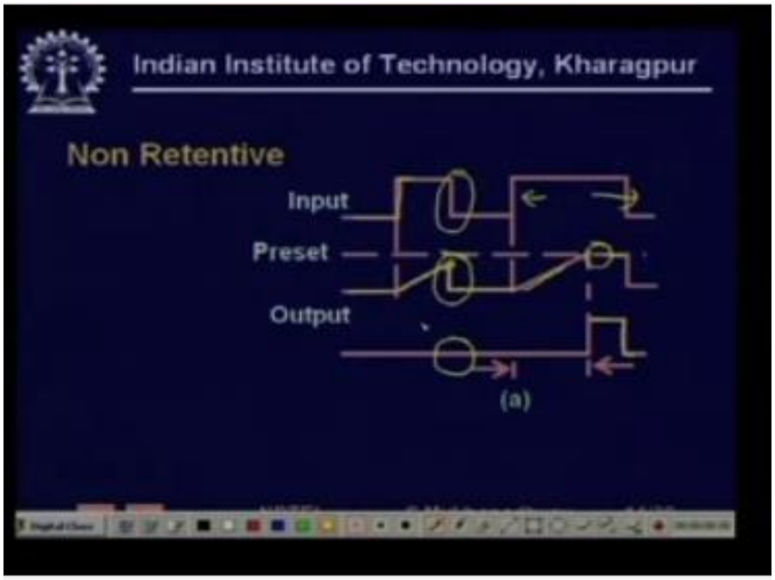
And here it comes down but still this does not come down so every time you get the same pulse width so every time this edge comes this ongoing edge at the input you get a fixed pulse at the output, this is the fixed pulse with timer and it will be an interesting exercise for you to see how this can be realized using a basic timer construct. So that is a point to ponder for you. Next is that we again can classify timers into two ways one is called retentive time another is called non retentive timer.

(Refer Slide Time: 25:08)



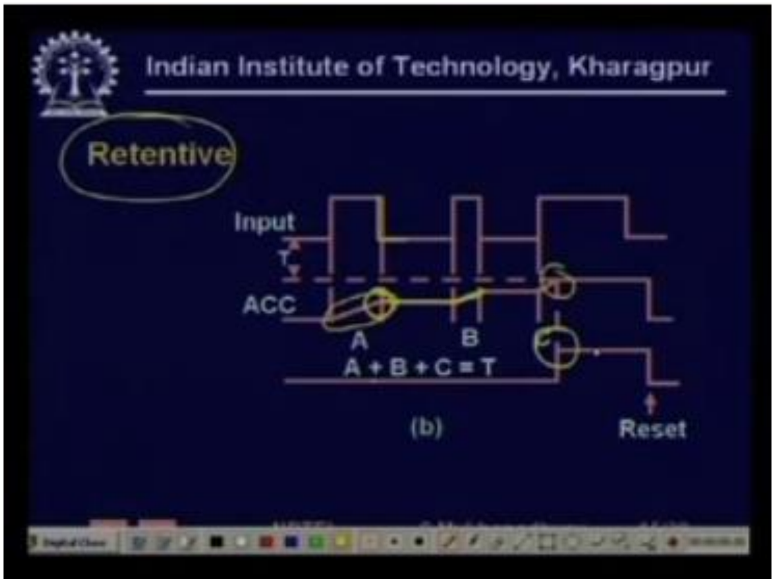
So for a non retentive timer what happens is that see the seethe input here goes up so the timing register starts increasing suppose a non delay timer, after some time what happens is that the input comes down, now the question is at this point so it has timed up to certain amount it has not reached his preset value. So now the question is that what will happen to the half but the input has come down. So what would happen to the timing resistor value at this point.

(Refer Slide Time: 25:50)



So in non retentive time words the timing register value is reset to zero, so every time you get another timing signal it against times and in this time because the input stage one for a much longer time so the preset values is reached and when it is reached the timer output is the timer output is asserted, see here the timer output is not asserted because the I mean the register did not exceed the preset register value. So this is called a non retentive timer contrasted to this there is a retentive timer which.

(Refer Slide Time: 26:34)



Now note the difference here, here also when the input goes one and then after sometimes comes down so the so the timing register value so this so the timing register value came up to this it actually did not cross the preset register so the output is maintained to 0. Now when this input falls down at this time also the timing register value is not lost but it is held so as long as the input is 0 it is held.

For the next pulse it counts from the previous value so that is why it retains its timing register value that is why it is called a retentive timer, finally as this increases at some point for the may be the next pulse it reaches the preset register value and at that point the timer goes high, okay. So this is the this is the semantics or meaning of the way the retentive timer works.