**INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR**

**NPTEL
ONLINE CERTIFICATION COURSE**

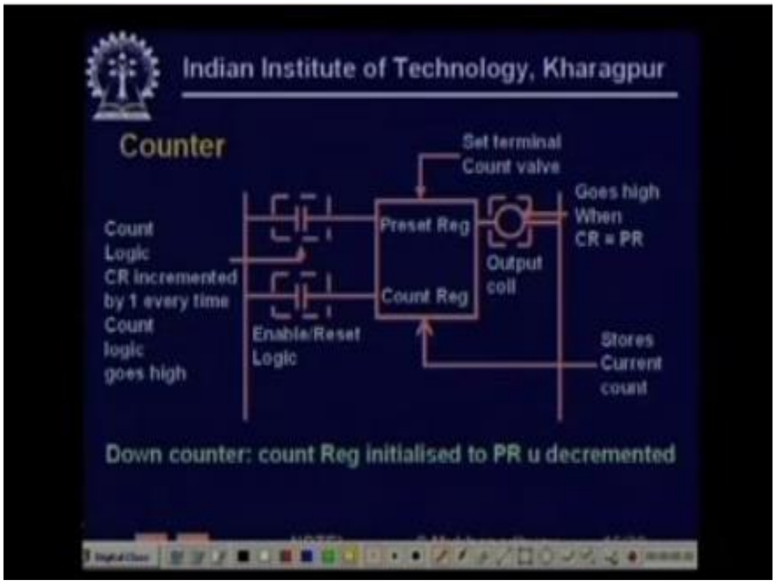
**On Industrial Automation and
Control**

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

**Topic Lecture – 27
Sequence Control,
More RLL Elements,
RLL Syntax (Contd.)**

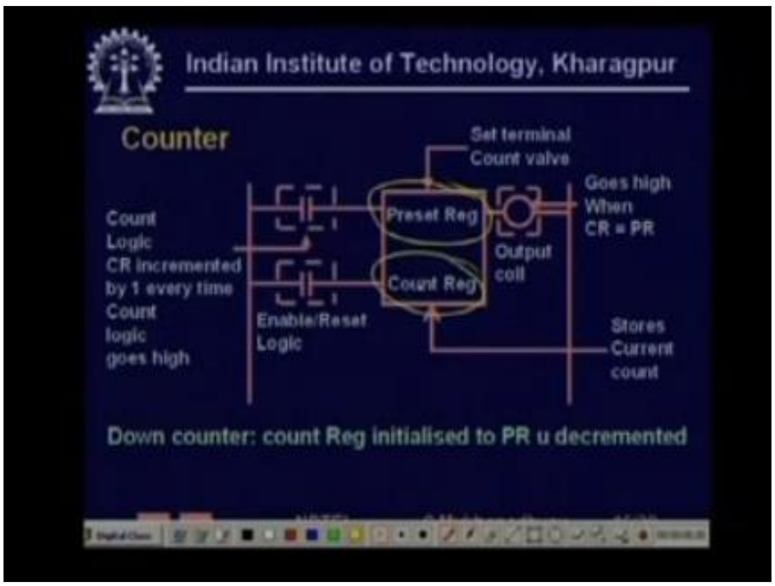
So now it so these are the roughly the various kinds of time are available and now we take a look at what is known as the counter.

(Refer Slide Time: 00:32)



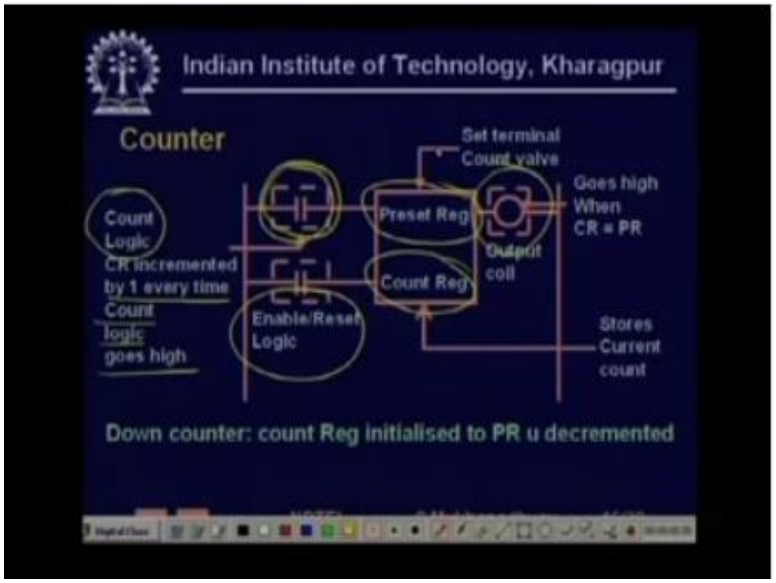
Okay so now the counter is nothing but frankly speaking the I mean the counter is nothing but a timer with an external pulse so now the in the in the counter also in the counter also there are two registers one is called a preset register and there is a count register but only thing is that in the timer the timing register is incremented by an internal clock at regular intervals of time so the timer is nothing but a counter which counts the timing pulses from an internal clock while for the count register.

(Refer Slide Time: 01:12)



For the counter discount you are trying to count something maybe the maybe the number of parts produced or the number of parts arrived or a conveyor or something like that so in which case with every event taking place there is a you have to generate some kind of a pulse using again contacts and sensors now those external pulses in this case will actually our augment the count resistor so that counter is in place of the internal clock pulses which were coming at the timer here we have external pulses which are triggered by the event that we want to count so again you have enable reset logic.

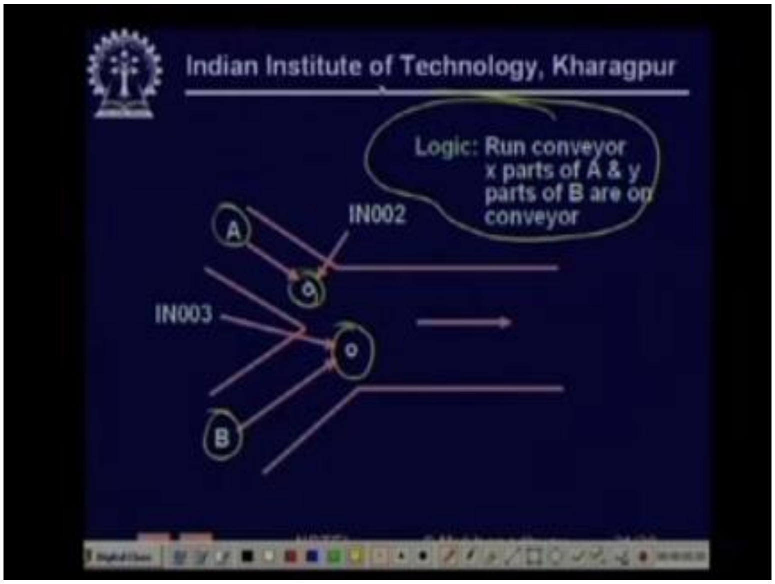
(Refer Slide Time: 01:54)



Same thing in this case we have what is known as a count logic rather than a timer run logic so the count register is incremented by one every time count logic goes high so every time this goes high from 0 the count register is incremented by one and just like the previous case when the count register value exceeds the preset register value at that time the terminal count is reached and the output coil goes high this is the meaning in some cases it may be a Down counter in which case the count register may be loaded with a preset register value.

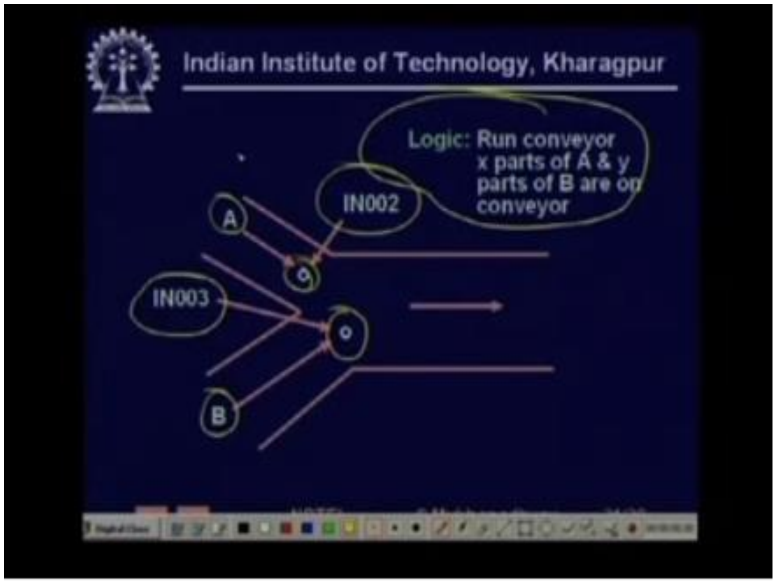
And it may count down to zero every time a pulse comes so it may be an up counter or it may be a Down counter it may have upward up and down count inputs by which it can have can be also an up-down counter some basic a very simple example.

(Refer Slide Time: 05:03)



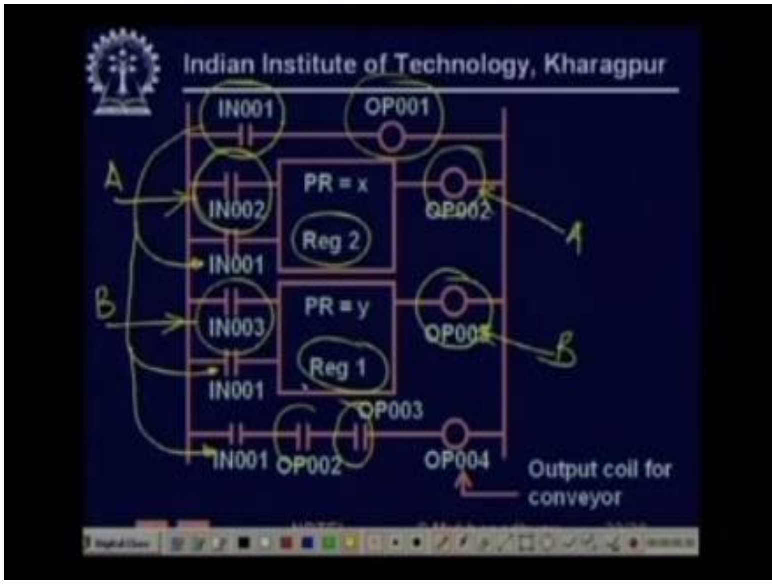
You have a conveyor into which parts are supplied so parts are coming from either this machine B or parts of type B are coming and parts of type A are coming so you want to implement this logic that run conveyor when parts x parts of e at least x parts of a and y parts of B around the conveyor at that time the conveyor will run this is what we want to program and the arrival of a part of type B.

(Refer Slide Time: 05:42)



Is asserted by this and arrival of a top part of type A is asserted by these two sensors so what is the logic so here is it so this is an example of using.

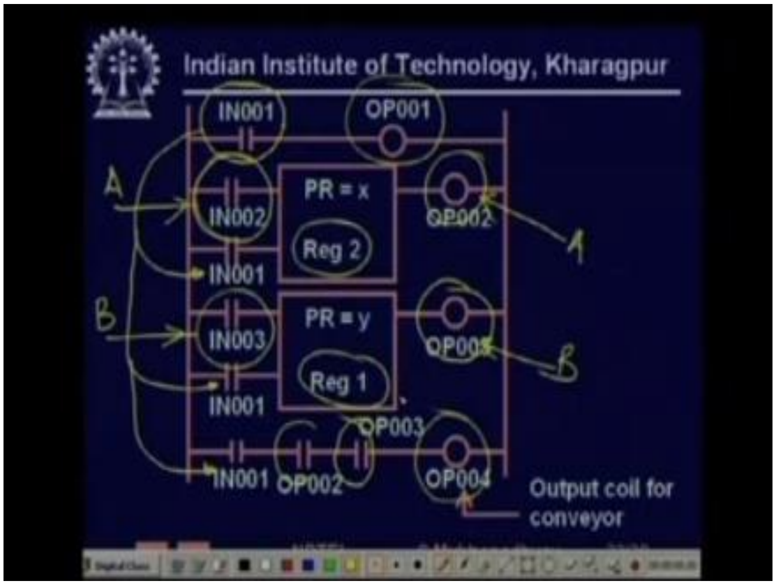
(Refer Slide Time: 05:58)



Counters so you see that we have two counters first of all this is the master switch the moment it goes on this output is asserted now this out now see that when IN001 is asserted this IN001also is also is asserted and this IN001is also asserted and this IN001 is also asserted so when this is 0 everything is 0 but when this goes one then all these rungs are enabled and both the timers are enabled now the arrival of part of type B is signified by this and arrival of the part of type A is signified by this.

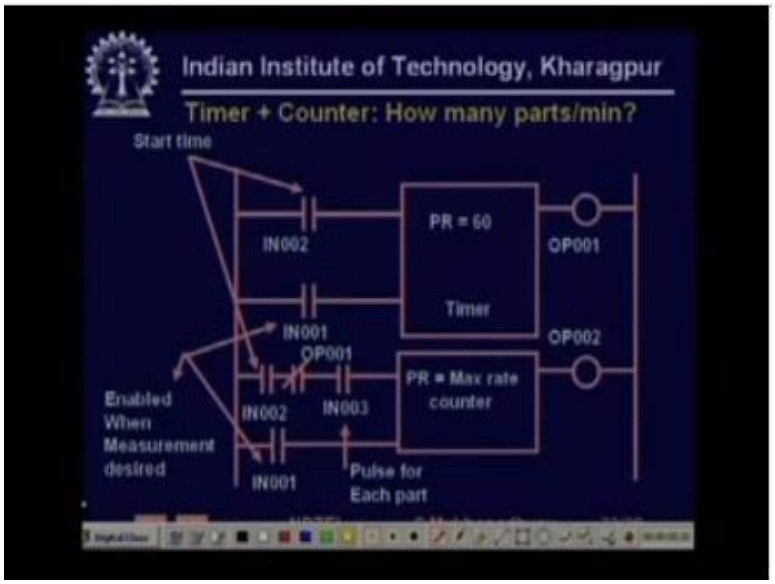
This is for A and this is for B so every time a part arrives there is a pulse here and the corresponding outputs are incremented and now when so the registers are incremented now the outputs the register that increment so when the registers when both of when is Easter crosses this output goes high so it is in series so OP00 so when you put them in series you want that both of them must reach their preset count values only then the conveyor.

(Refer Slide Time: 07:43)



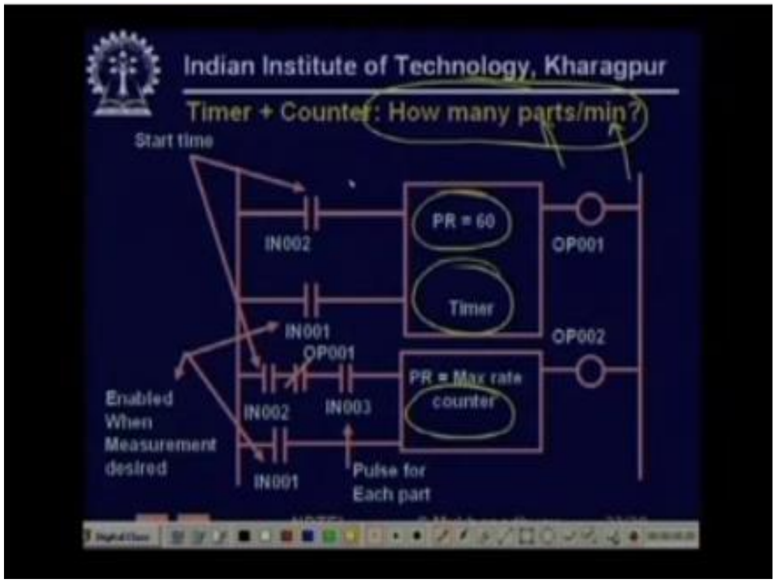
Will run if you put them in parallel it means that if any one of them reaches that preset value then they count then the conveyor will run so it is an example where you can use a counter for an industrial problem this is a nice example of counting.

(Refer Slide Time: 08:08)



How many parts per minute are going on the conveyor let us say which in a way indicates that weather what is the production rate so it may be an important management information to you know display what is the production rate so what so now is this therefore this is so we want to.

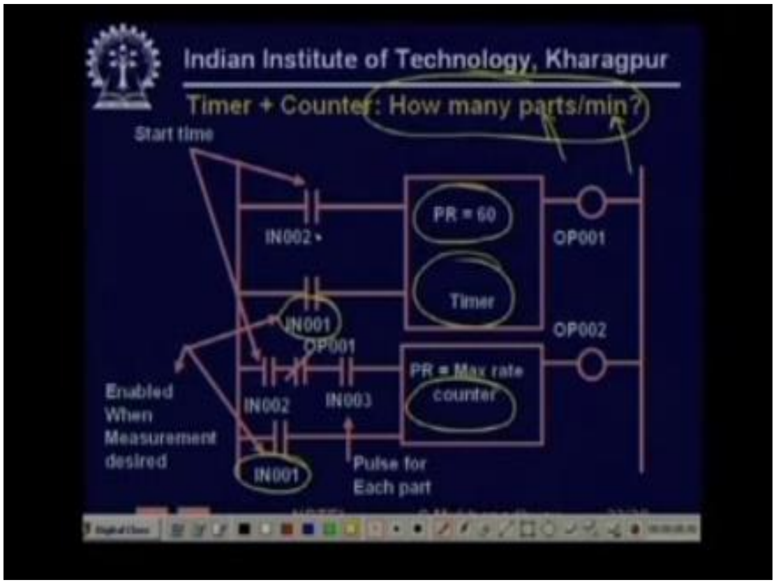
(Refer Slide Time: 08:27)



Count how many parts per minute so we want to not only keep count parts we want to also count it only over one minute of interval after we want the press start for this operation so obviously for creating this one minute interval we need a timer and for counting the part we need a counter therefore it is a mixed timer counterexample so here we have the timer and here we have the counter both have their here it is loaded to 60 because we are talking about minutes and we are assuming that the internal clock.

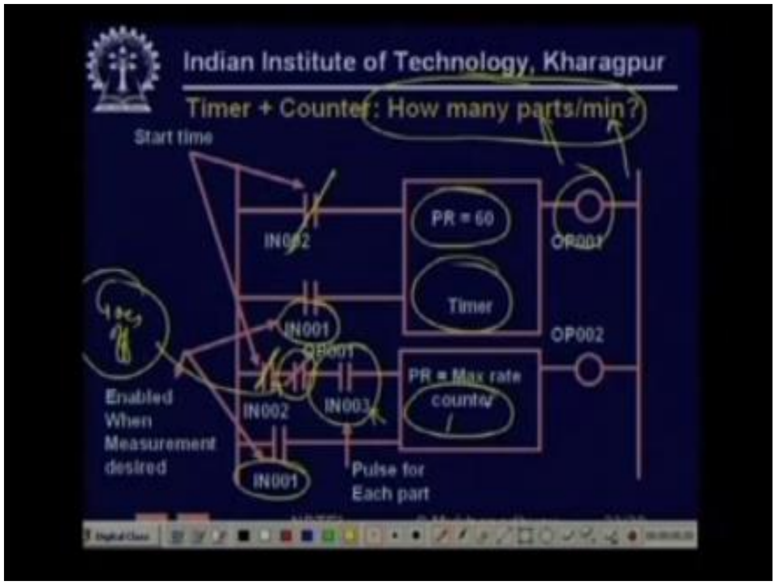
Pulses are available at every one second interval so now suppose IN001 is a switch which you which one can press when he wants a measurement that over the next one minute how many parts per fast so when these two are the one when this.

(Refer Slide Time: 09:25)



Main measurement desired contact is enabled at that time both this timer and the counter available now what happens is that so when you say start these are kind of you know master switch then when you say start time so you want to know of within this time of how many parts a pass so you start time so this goes high.

(Refer Slide Time: 09:55)

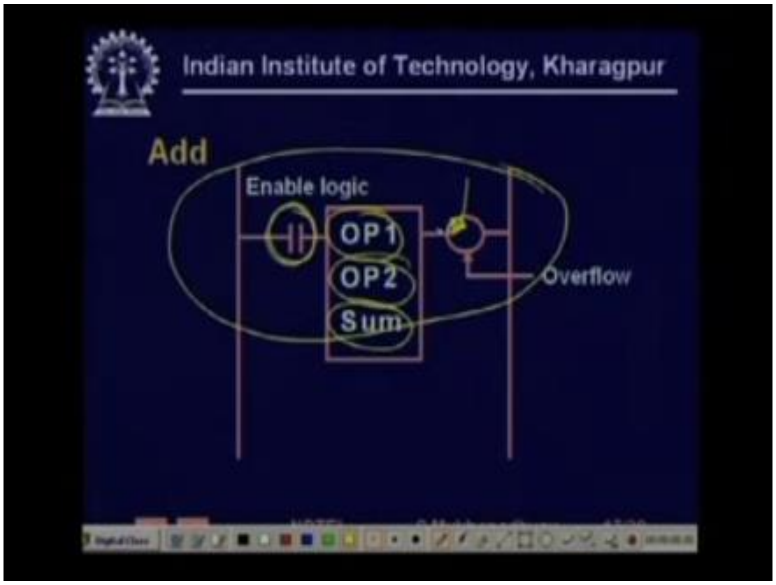


Now this does not go high so this is low so now and when this is on this is also on so now every time a part arrives you get a pulse so the counter goes up in the mean time after the time has expired OP001 goes open goes off so this goes to 0 and then further parts are not counted so this when this is going to 0 this becomes open whether this here you get a closed or not it does not matter this cannot be 0 because these are in series so therefore at that time you in the in the counter.

You have the value of how many parts have passed over the conveyor in the last minute these are so we have covered the timer and counter in some detail now we will take a look a very brief look at you know there are arithmetic instructions there may be logical instructions specifically instructions like compare instruction like doing and ignoring so all these instructions are available just like a low-level language and you can express them in various formats depending on the manufacturer as I said.

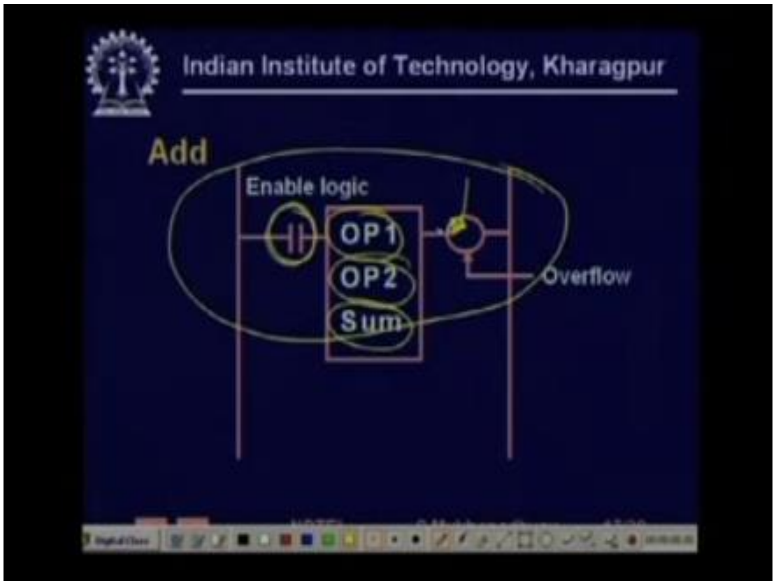
So in our in our format we are saying that by this diagram we are saying that this.

(Refer Slide Time: 11:45)



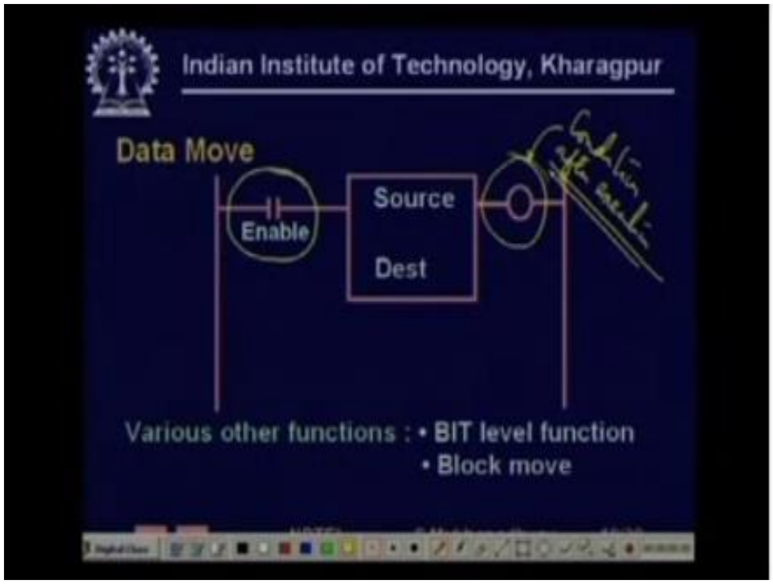
If you put this round it is wrong for doing the it will when this rung will be executed basically two operands operand one and operand two will be added and there some will be put at the location some provided this enable logic is on and this output coil may be used for various purposes one of them could be that if there is an overflow if there is an overflow that has occurred during the summation.

(Refer Slide Time: 12:20)



Then it could go one so it can indicate an error condition and then be used in the further logic so this is just an example of add similarly you could have sub you could have multiplied you could have various other things this is just one example of the of an arithmetic instruction next is data move so if you want to again you depict it as a rung because in the other everything is the run so in this case you have again that the data.

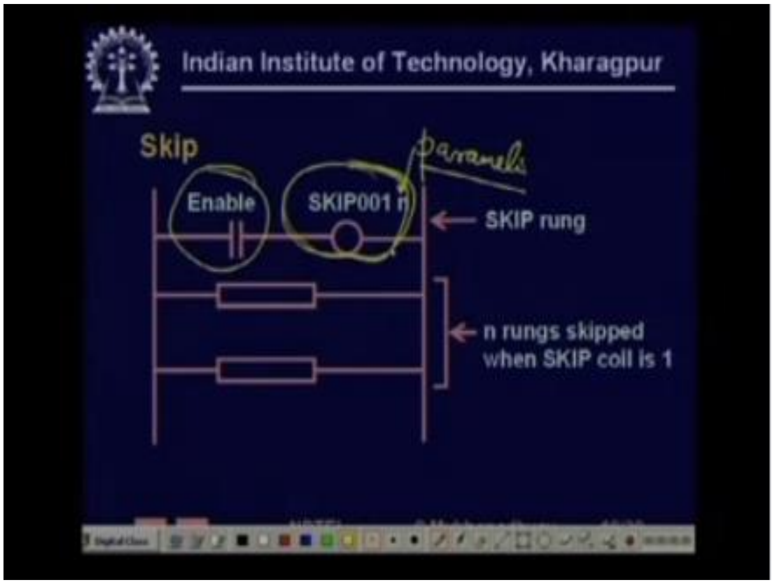
(Refer Slide Time: 12:55)



Move will take place depending on whether the enabling logic is satisfied or not and then data will move from source to destination and maybe if there is some address failure or something then or maybe after the data has moved actually the this can become one source so this can also be used for indicating condition after execution you can have so this is the data move you can have various other functions like you can have a bit manipulation functions you can have various kinds of block moves etc.

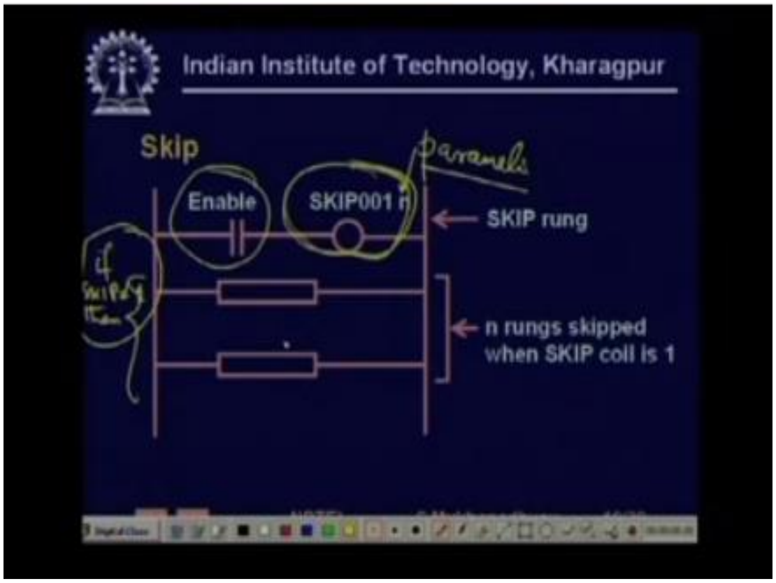
You also have various kinds of logical instructions so apart from a transfer that the last instructions we are very important that program control instructions by which we want to we want to control we do not want to execute all rungs of the of the RLL at all times but rather we want to control whether we can skip some of them or enforce some of their values so we have typically to give an example we have a skip skipping facility which so when.

(Refer Slide Time: 14:19)



Again when this enabled then this skip 001 this is that is a very special type of contact so it is enabled when it is enabled then what it means is that the next N run so you have skipped 0 1 and n is a parameter so it means that the next n rungs will not be evaluated they will when this is high this is a meaning so it is like you know it is like if skip not equal to 1 then so this is like if.

(Refer Slide Time: 15:01)

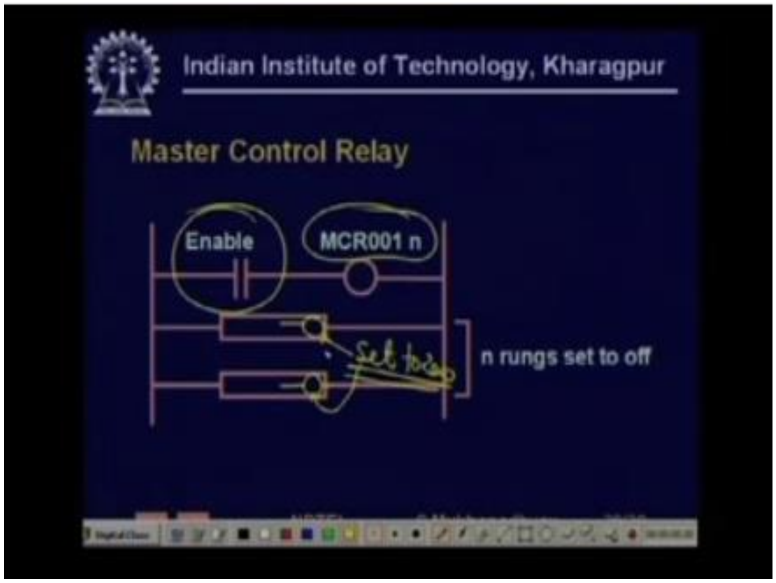


Skip not equal to1 then this will be executed if sleep is equal to 1 as long as stage one the be next n runs will not be evaluated and they will be maintained at though at the old value similarly you can have another as another facility which we call the master control relay.

(Refer Slide Time: 15:31)

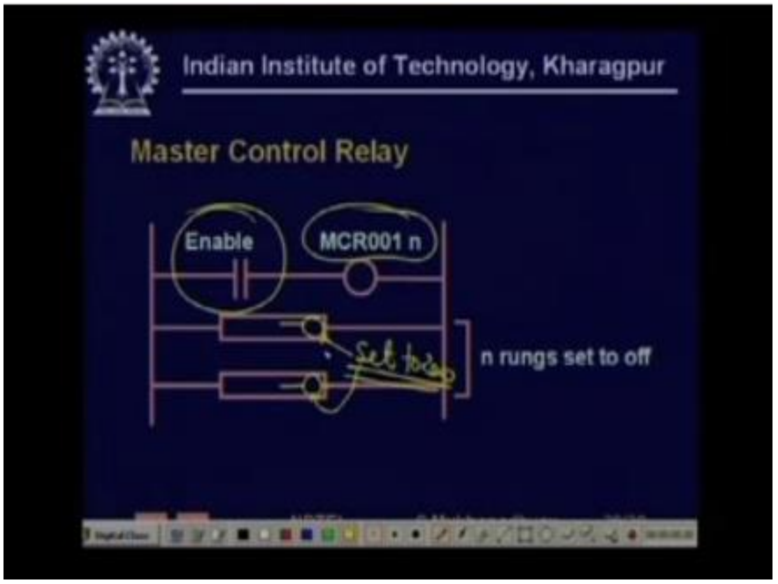
Which means that here I mean this is also a program control statement and it means that whenever the enable logic is satisfied then this MCR output coil which is a very special output coil is excited and which means that the next n ranks will.

(Refer Slide Time: 15:47)



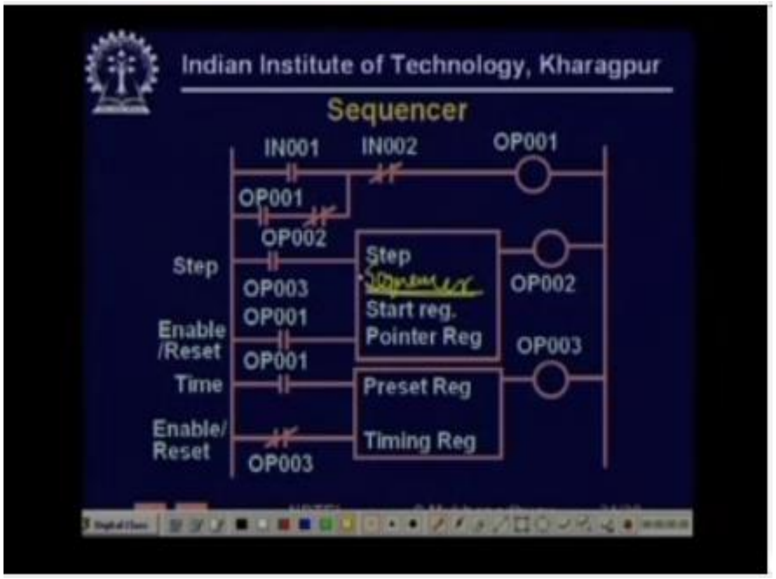
We set to their outputs each one of them has an output coil here which I am not shown so the next output coils these output coil values will all set to zero so without evaluating irrespective of the logic in this branch if this is excited.

(Refer Slide Time: 16:12)



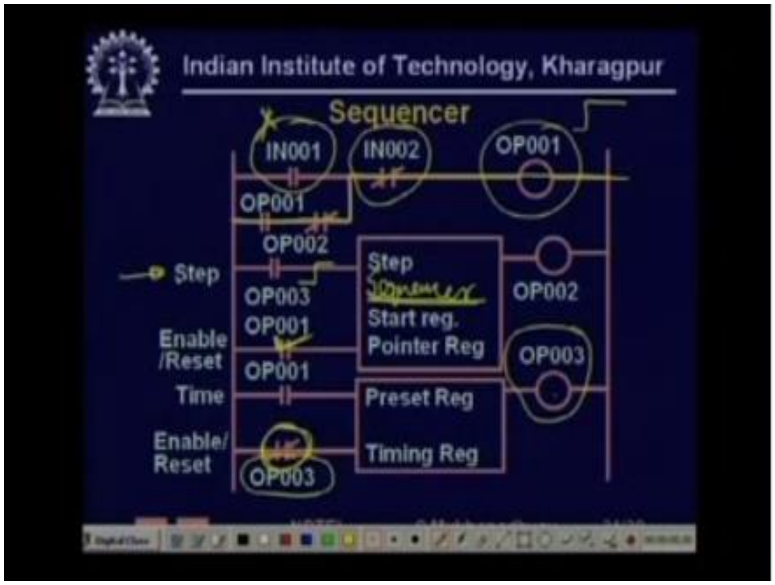
Then there will be all set to 0 but it will be but if this is not excited then they will be evaluated normal like normal runs according to normal PLC logic evaluation these are this there are some special instructions which also are there in a beaner PLC RLL program language for example this is a.

(Refer Slide Time: 16:41)



Sequencer so a sequencer this is actually the sequencer is a block which can be separately programmed and which nicely execute a sequence of steps every time it is excited so let us see what is happening here.

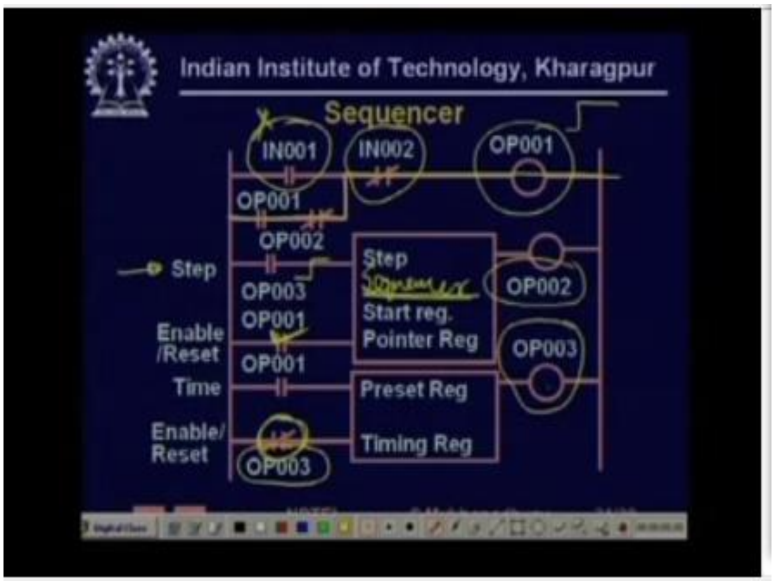
(Refer Slide Time: 17:03)



So first of all this is the master switch which will be come on let us assume that and suppose this is off so first of all what happens is that this goes high when this goes high and this is already low so therefore there is a path from this place to this place and even if this is taken off even if this is taken off this path maintains whenever there is a path here immediately this is enabled and this is this is enabled what happens is that when this is enabled and this is now reset now this is normally closed.

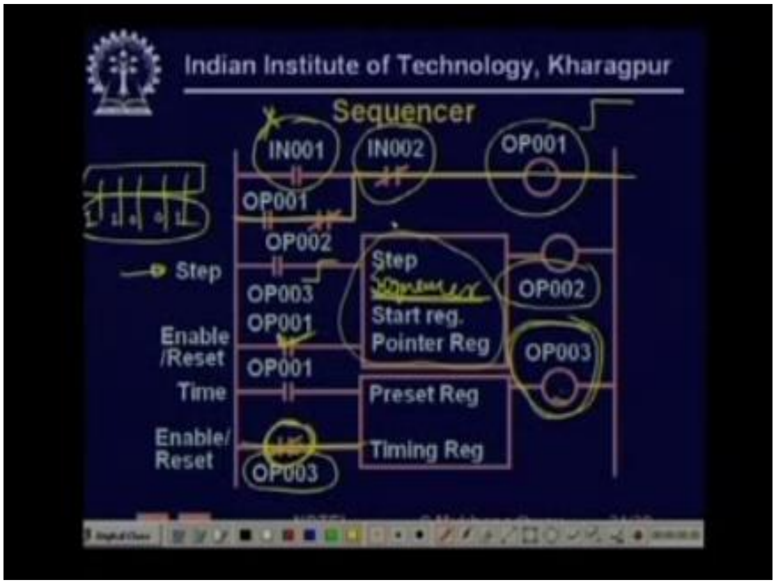
Oh no this is normally closed so when this is closed it is already enabled because this is off so therefore and then this OP001 goes on so when this OP001 goes on it starts timing so after a preset timing this OP003will go high now when this goes high immediately what happens is that you get a pulse here so then the sequencer every time it get up gets a pulse into its step input it executes a particular.

(Refer Slide Time: 18:56)



Set of outputs it will exercise okay so particular set of outputs which are you know bit outputs which are stored in some registers okay so there is a start register so there are sequence of registers every time a pulse comes a new set of register output so you have say.

(Refer Slide Time: 19:18)

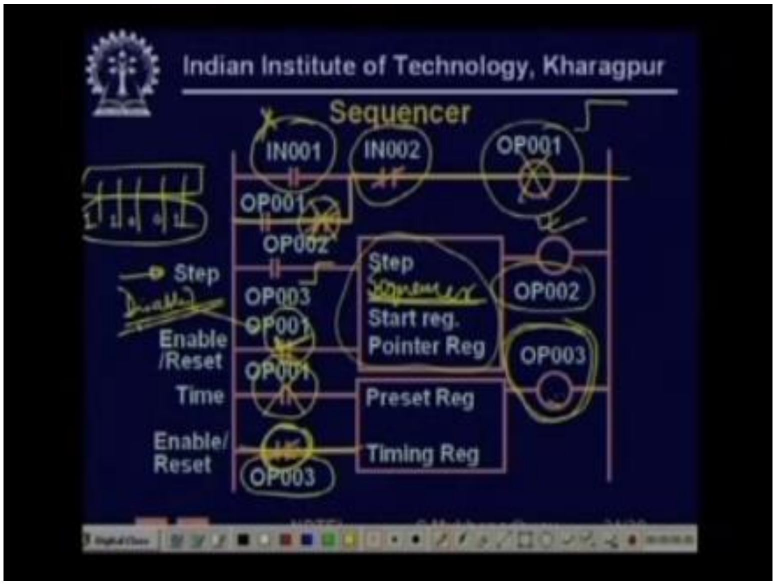


Register maybe these are you know three different or eight different valves so you have stored some values so maybe even in the first step this is 1this is 0 and this is one that is the way you have your programmed it so when step one is executed then these outputs will actually go to the field so the first pulse has come and the first step is executed after the first step is executed in the meantime you see that so the first step is executed and it is and is waiting there now this is when this has become one immediately this becomes 0 and it becomes 0 again.

So again it is enabled so now what happens is that an output 0 0 1 is already on so therefore again it time so basically what I happening is that it is with this arrangement it is continuously timing and then ever and the moment this OP003 goes high after the timing interval this gives a pulse here so a step is executed in the sequencer and then because it is latch like this so immediately when this goes high this is again becoming reset and then once it is becoming reset it is this one is enabled.

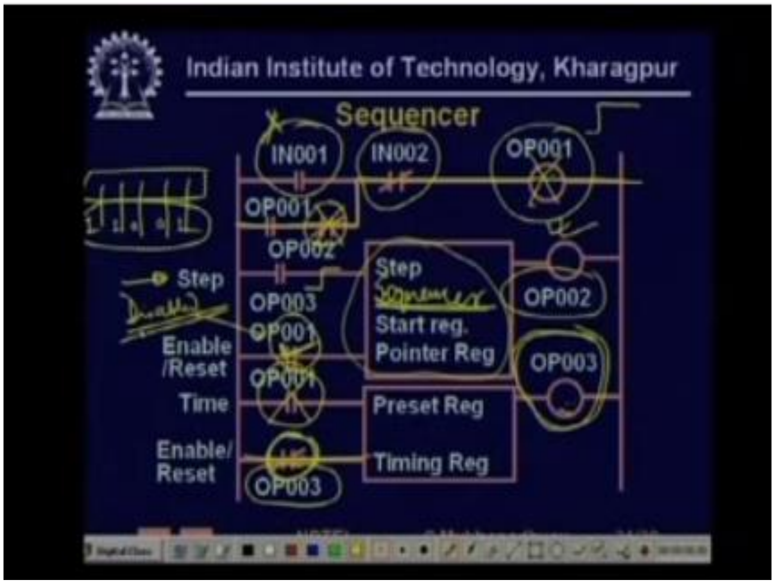
And therefore it is again timing so it is generating continuously generating some timing intervals and at the end of each interval you are getting a pulse which is executing the chart register the executing the sequence register of the sequencer at the end of the sequence the sequence has a fixed number of steps at the end of the sequence this will go high.

(Refer Slide Time: 21:10)

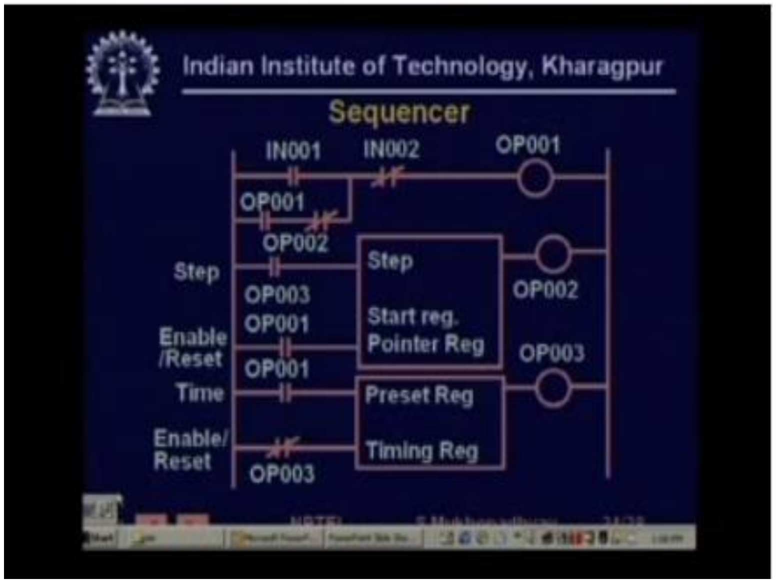


So once this goes high then what happens is that this one goes off when this one goes off and then there is no more timing here and this itself that is a sequencer itself is disabled so this is a way a sequencer works.

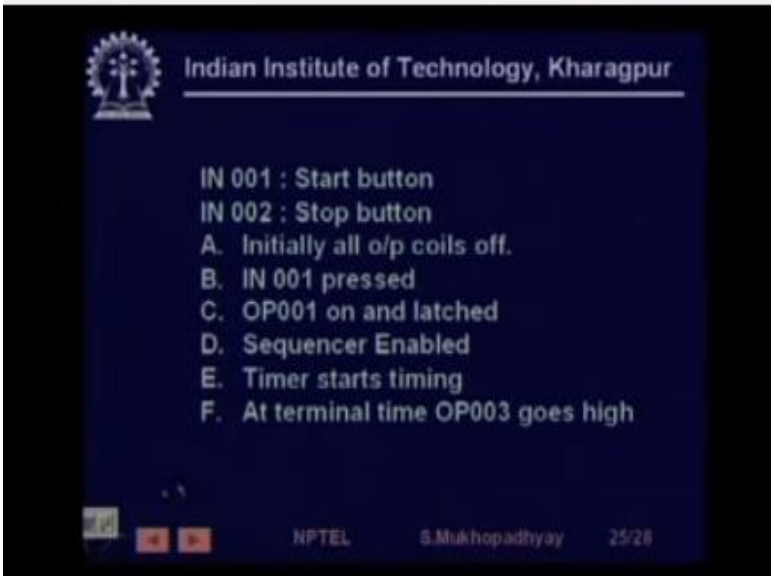
(Refer Slide Time: 21:46)



(Refer Slide Time: 21:48)

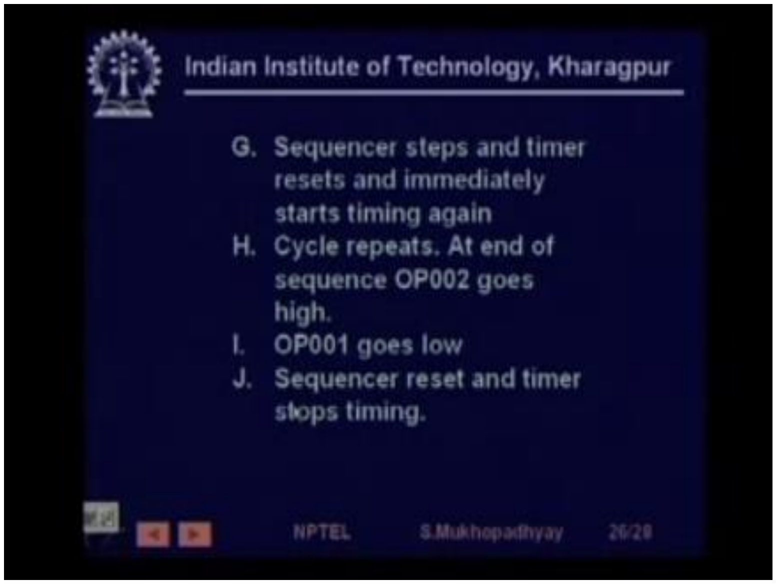


(Refer Slide Time: 21:50)



Now so this is this is what is expressed here so if you press Start button and 001is the start button IN002 is actually as top button you could make the sequence stop at any times so initially all output coils off then I am 01 pressed then output 001 on and latched then sequence are enabled the timer starts timing then a terminal time output three goes too high.

(Refer Slide Time: 22:20)

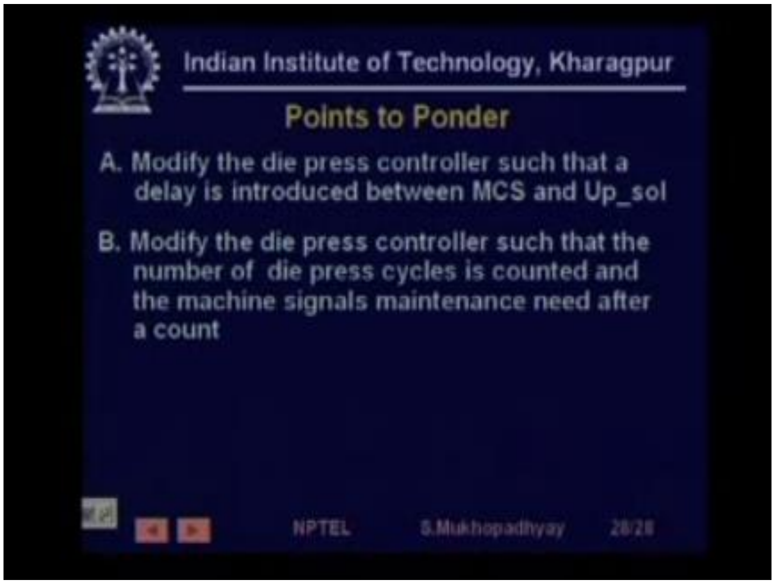


And immediately sequence of steps and timer resets and immediately start stymieing in that is what we explained and this cycle repeats at the end of the sequence OP002 goes high which means that the OP001 goes low and the sequencer resets and the timer stops time.

(Refer Slide Time: 22:40)

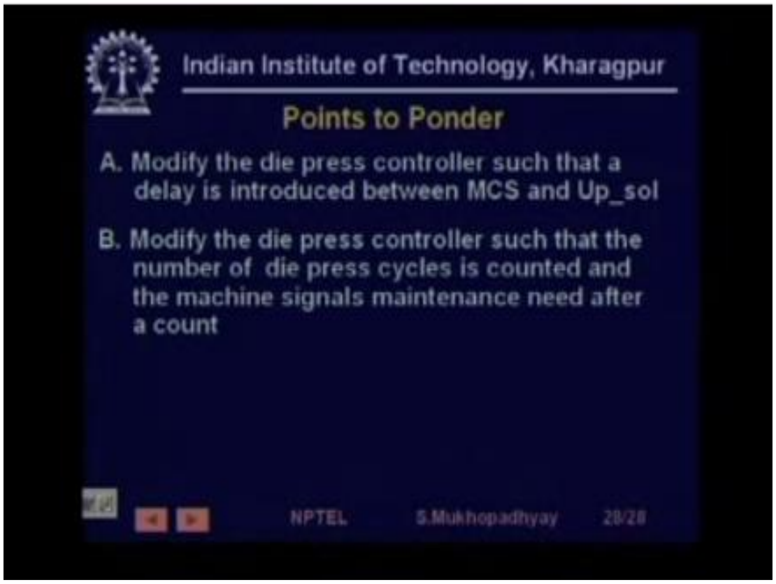
So this is how it works so we have come to the end of the lesson and in this lesson we have seen the various timers and counters and we have also seen some arithmetic that data move and program control operations and finally we had seen other you know macro operations like a sequencer there are sometimes even other some other continuous mode operations also like PID etc. Which we have not seen so far so coming to the end we have the usual.

(Refer Slide Time: 23:24)



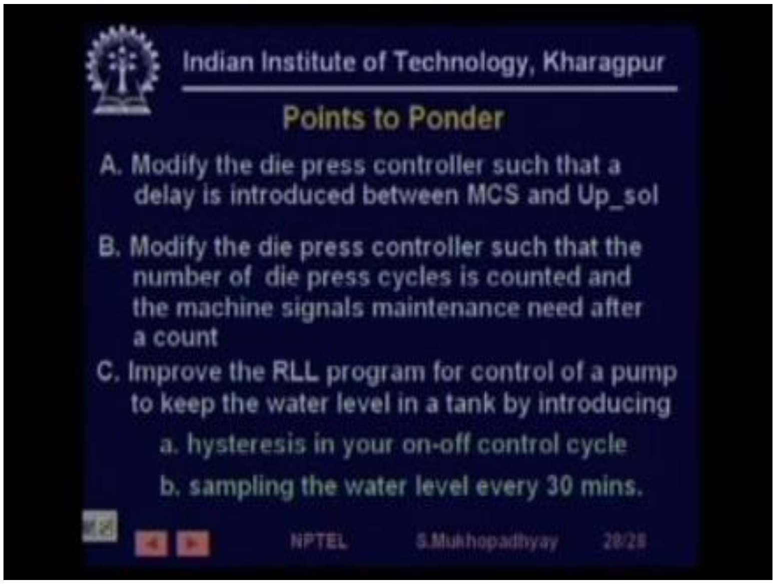
Points to ponder for example you could try to modify the die press controller such that a delay is introduced between that is after the master control switches put on and the up solenoid goes on there is a delay you can try to put that by modifying similarly you could also modify the.

(Refer Slide Time: 23:47)



Die press controller such that the number of die press cycles is actually counted so you say that after every thousand presses you want to stop the Machine and you want to maintain them so you want to count every time a complete cycle of die is one going up and one going down is completed you want to count them and after you the it reached a count you are do you want to stop the machine for maintaining similarly you could improve the RLL.

(Refer Slide Time: 24:18)



Program which is which we said in our earlier points to ponder earlier program for control of a pump to keep the water level in the tank by introducing a hysteresis in your own of control cycle and this an odd and also sampling the water level every 30 minutes that is not continuous sampling the order level every 30 minutes.

(Refer Slide Time: 24:42)

So that is all for today thank you very much.