Industrial Automation and Control
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 20
Sequence Control More RLL Elements RLL Syntax

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

(Refer Slide Time: 00:54)

Indian Institute of Technology, Kharagpur

Lesson 20

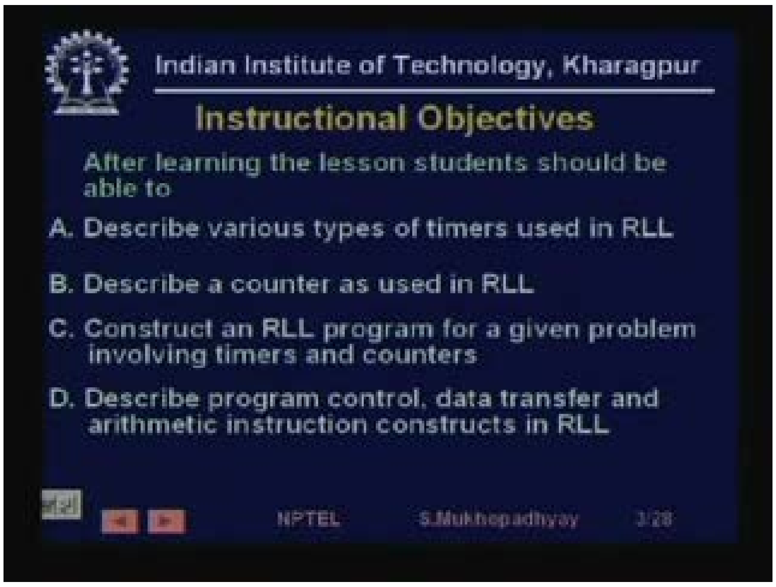
Sequence Control

More RLL Elements
RLL Syntax

NPTEL S.Mukhopadhyay 2/28

Today, we are going to look at some new programming elements namely timers counters etcetera 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.

(Refer Slide Time: 01:20)


Indian Institute of Technology, Kharagpur

Instructional Objectives

After learning the lesson students should be able to

A. Describe various types of timers used in RLL

B. Describe a counter as used in RLL

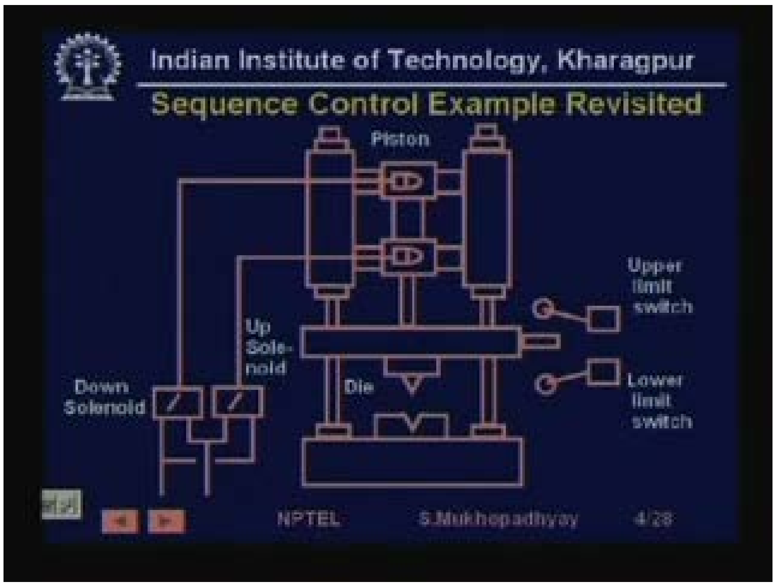
C. Construct an RLL program for a given problem involving timers and counters

D. Describe program control, data transfer and arithmetic instruction constructs in RLL

NPTEL S.Mukhopadhyay 3/28

So, before we get on we take a look at the instructional objective which is to, so that a student will be familiarize or he will be able to describe various types of timers used in relay 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. 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 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 constructs, so for writing complete program they are sometime necessary, so we will get familiarize with some of them.

(Refer Slide Time: 02:19)

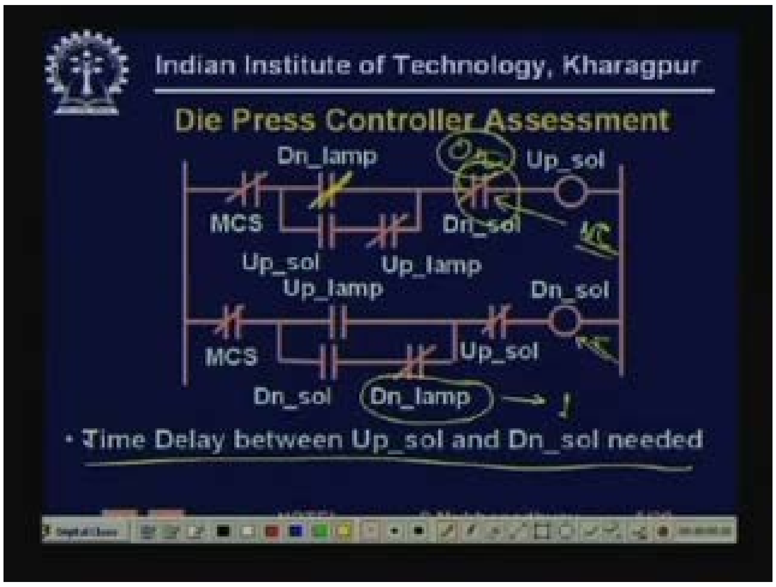

Indian Institute of Technology, Kharagpur

Sequence Control Example Revisited

Of course, you must remember in this context that there are relay ladder logic programs are often a somewhat nonstandard. So, it is not that we are following a particular manufacturers constructs, but it is very likely that most of these constructs will be found in each PLC manufacturers programming repertoire. 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 look take a second loop at our previous example of the die press.

So, here is a die press where a 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 sense the positions of the end positions of the die, so we take a look at the earlier solution, which we proposed before in an earlier lesson.

(Refer Slide Time: 03:53)


Indian Institute of Technology, Kharagpur

Die Press Controller Assessment

- Time Delay between Up_sol and Dn_sol needed

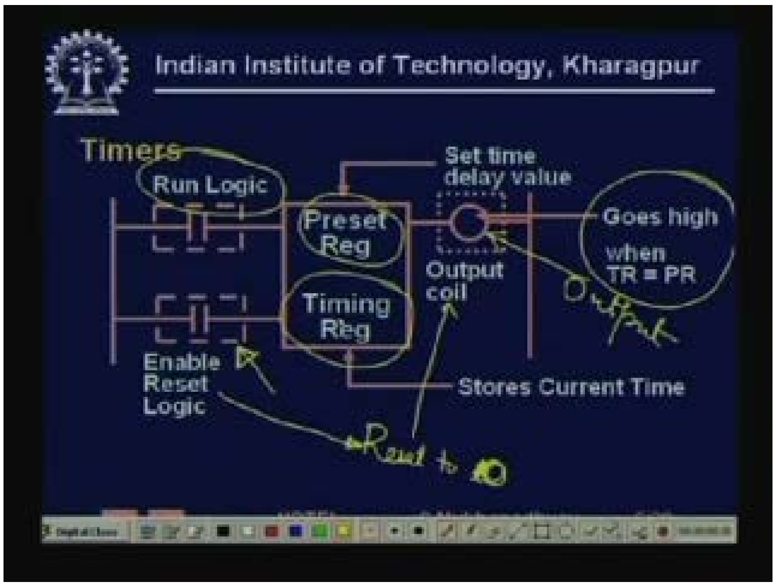
So, here what we did is here we proposed a an RLL program which use only the normally on and the no and that is that is normally open and the normally closed contacts, the real input contacts as well as some auxiliary contacts and some output coils. Now, what happens here is that let us look at this that initially suppose the down solenoid is ON. 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 and the down lamp will go on.

This goes to 1 immediately what will happen is that is that this previously the path was path that was in 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 slowing down. Now, what happens in this program is that the movement the down lamp is 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, this is also on because the down solenoid is off.

So, 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 press the die has to come down on the suppose it is a sheet medal 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 it may be very common that a time delay between up solenoid and down solenoid is needed that is after the down lamp is on. By the time the up solenoid again becomes energized, we want to delay, so what I want to say here is that such delays are very often needed in industrial operations and today we will see how to create these delays. So, as we shall see, the solution a little later, but first let us look at the timers, which actually create these delays.

(Refer Slide Time: 07:56)



Now, here I want to is mention that these timers the timers are sometimes you have we are all familiar with timers, we are our possibly our first interaction with introduction to timers was in the digital electronics course. Now, those timers are actually hardware timers they in you may you may actually use hardware turn time is also in a PLC in which case you have you have a separate you may have a separate timer card, but in this case we are talking of the program.

So, it is actually it is actually a piece of code which creates a delay in asserting some output, so the purpose of the timer is to create the delay how much delays that can be programmed number one and number two is that the timer.

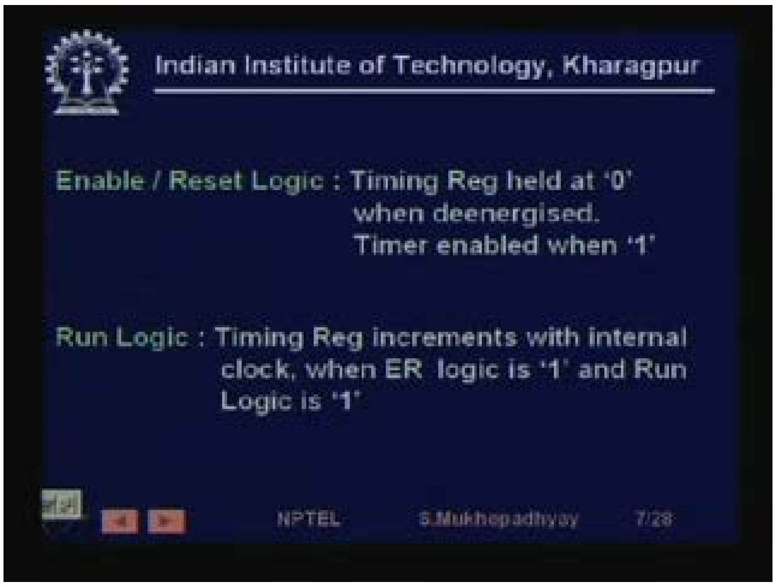
Actually, you know it is the basic idea of a timer is that there are two resistors one is called the preset register and the other is called the timing register. So, the preset register is actually set wherever you create timer in the program you actually set the preset register gets loaded by a particular value and it stays fixed. While the timing register during the time 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 it, you may you may also for example as we shall see that counters are actually work on the extra external clock.

So, because you want a particular timing, so generally it is said from the semi internal clock. So, as the timing pulses on the clock are coming, so the timing register keeps increasing and when its value exceeds this every time there is a there is a comparison between the timing register and the preset register. So, after sometime what will happen is that the timing register value will exceed the present register value at which time the timer will stop timing. Further, the timing resistor will stop incrementing and the output will be asserted, so the output will be asserted when TR equal to PR. Now, this thing happens when the timer is active, so when this timer active, that can be again controlled by using two kinds of logics.

So, first is called the enable or reset logic, so whenever this logic becomes 0, the timer is not enabled it inactive and the output coil is reset to 1. So, this output coil reset to one when that is called reset logic reset to 0, I am sorry reset to 0. Similarly, now when it is enabled, so at that time it is enabled two time, but a whether it will actually time or not that can be again controlled by the run logic. 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. So, I would also like to assert mention that in this case we have mentioned, we have actually put it by a single contact we could easily 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. So, it is is not necessary that you will have to always make it a very simple logic, you can make it as complex the logic as you want. So, having done that, this is a basic timer, now timers can be timers create delay, so we can have various types of delays and all of them can be realized by this basic timer module.

(Refer Slide Time: 12:36)



Indian Institute of Technology, Kharagpur

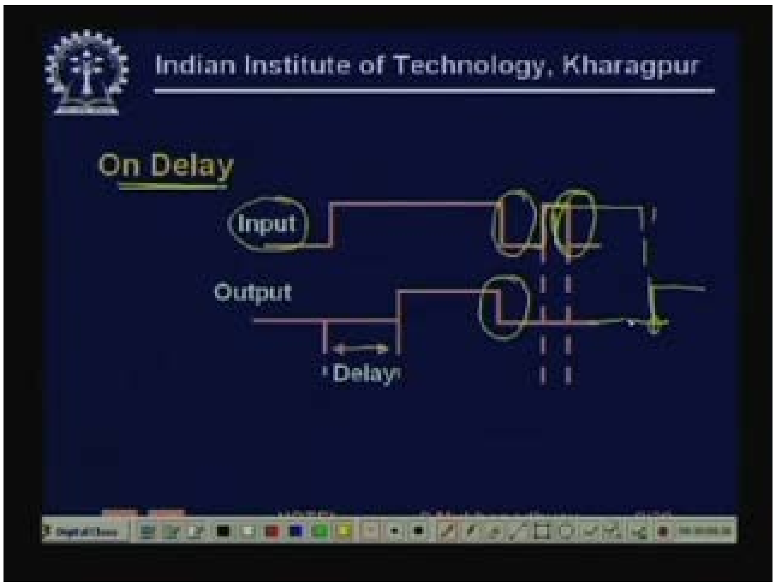
Enable / Reset Logic : Timing Reg held at '0' when deenergised. Timer enabled when '1'

Run Logic : Timing Reg increments with internal clock, when ER logic is '1' and Run Logic is '1'

NPTEL S.Mukhopadhyay 7/28

So, to recapitulate we have enable reset logic where the timing register is held at 0 when it is de-energized at 0, the timer is e is enabled when 1, so at that time the timer is ready to receive the clock pulses and increments its values. Similarly, we have run logic where timing register increments with the internal clock when the enabled reset logic is one that is the timer is enabled and the run logic is also 1, that is what I say, so we take a look at the different kinds of timers, first is the on delay timer.

(Refer Slide Time: 13:07)

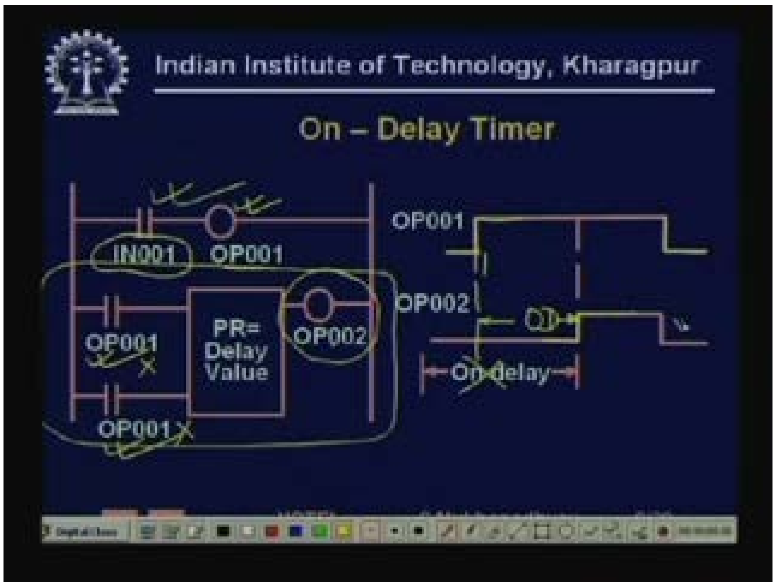

Indian Institute of Technology, Kharagpur

On Delay

So, here we are saying that if an input timer creates a delay between an input and an output. So, here we say that if the input signal goes on then the output signal will go on after a little delay, which is why it is called on delay. So, while the input signal becomes 1, the output signal becomes on after a delay, but if the input signal goes off then the output signal goes off immediate. 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. Similarly, note here that here again it becomes on, so here again it becomes on now the timer if this on would have 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 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, similarly now we will see that we can easily realize this on delay timer using our old timer.

(Refer Slide Time: 15:06)


Indian Institute of Technology, Kharagpur

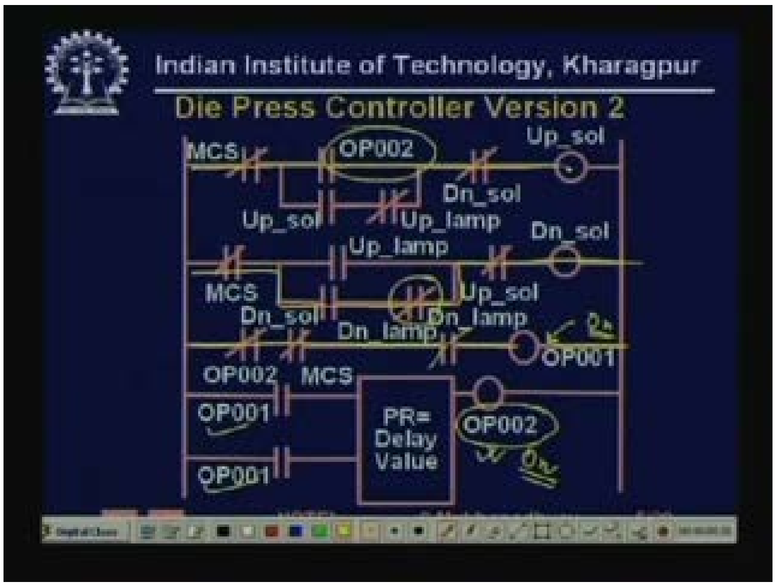
On – Delay Timer

So, now here is a circuit here is a RLL logic circuit or as number of runs which creates which takes the basic timer unit this is the basic timer you need that we are seen takes it and couples it with another run and makes an on delay timer. 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 that is OP 0 0 2 actually go high after a short delay. So, what happens is that this whenever these are OP 0 0 1 goes on, so you see that the run logic is enabled and the enabled reset logic is also enabled which means that the timer is typing is timing now.

So, the timing register is getting incremented with the internal top pulses and when it will cross the preset register at that time the OP 0 0 2 signal will go up. So, you have this is this is 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 movement this becomes off, so the movement this becomes off immediately these two become off and immediately the OP 0 0 2 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 on delay timer, so let us see the other kinds of timers. Now, before doing that we want an on delay let us use this on delay timer now in our die press example to see whether we can introduce a delay.

Between that is after the down lamp goes on we want that the up solenoid will go on after an on delay. So, we do that, so here is our you know our earlier the first two runs are very similar to our earlier solution except for this one, I am sorry.

(Refer Slide Time: 18:02)

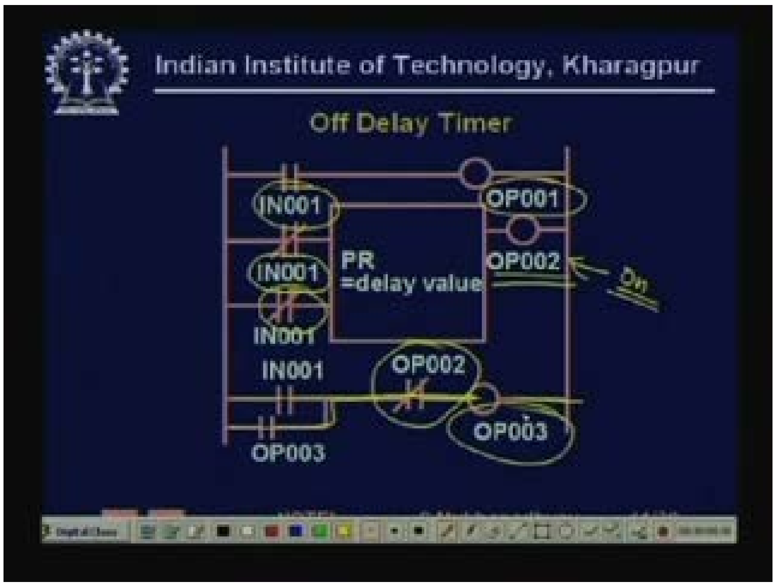

Indian Institute of Technology, Kharagpur

Die Press Controller Version 2

So, except for this one, 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. So, the movement the down lamp becomes on, so this will off and did and the down solenoid now go down. Now, when the now looks at this, so initially the OP 0 0 2 is off, so therefore, this is off and MCS this is the this is the masters switch this also becomes off. So, initially supply was coming up to this now the down lamp has become on, so immediately OP 0 0 1 becomes on when OP 0 0 1 becomes on the timer is enabled and its timing.

So, after a delay this OP 0 0 2 goes on becomes on, when this becomes on, now there is a direct paths to this and up solenoid becomes on and where up solenoid becomes on. So, then the usual operation starts, so what you have demonstrated is that we have put a timer run. Now, there is a time delay 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 timer for example, off delay.

(Refer Slide Time: 20:34)

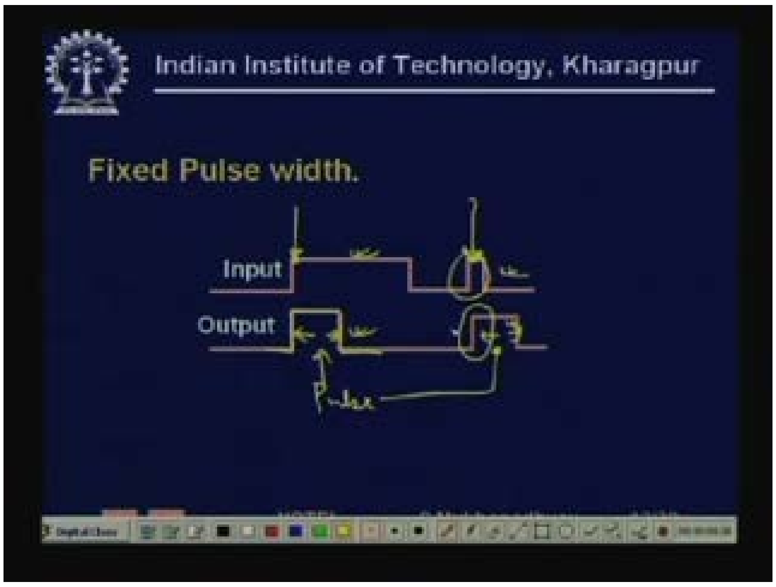


So, the off delay timer is exactly like the on delay timer except for the fact that now the delay is getting off. So, the moment the input becomes on the output also becomes on no delay, but when the input becomes off the output becomes off after the certain delay. So, this is the delay this is the delay and the same phenomenon is actually observe that if this becomes on again for a short pulse, so this also becomes on. Then, where it becomes off where it becomes off here this delay starts here, but before the delay and then after is 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 has not expired, so it continues and then at the end of this again after a delay, it becomes off.

So, basically is 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 we realize this one, so that is simple to realize, so you see that now we are having off delay, so again this is the input goes on. So, immediately OP 0 0 1 goes on when OP 0 0 1 goes on you see IN 0 0 1, this is already off in the reset position, so OP 0 0 3 immediately goes on, this is my final output. So, there is no delay in getting on now suppose and when OP 0 0 3 goes off we can we can use this one say it latches. Now, imagine that IN 0 0 1 goes off, so when it goes off this OP 0 0 1 when this goes off, now these are NC contacts, so when all the time when this is was on these this OP 0 0 2 was held to 0.

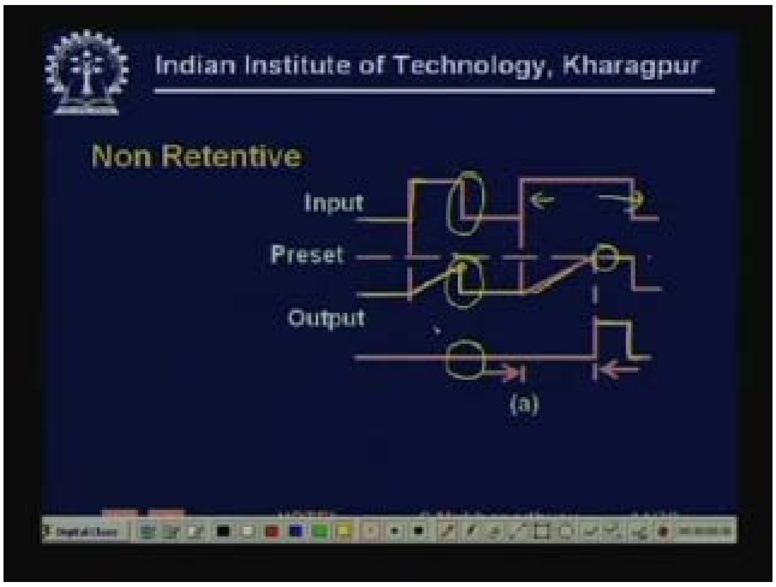
Now, when it will be 0, then immediately the timer will start timing and after sometime this will become go high come on. When this becomes on immediately when this becomes on this becomes off and OP 0 0 3 falls down, so there is a delay in getting OP 0 0 3 off. So, this is very simple realization of 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 there is a irrespective of the delay, you will get a fixed pulse from the output, so here this is just like an off delay timer.

(Refer Slide Time: 24:08)



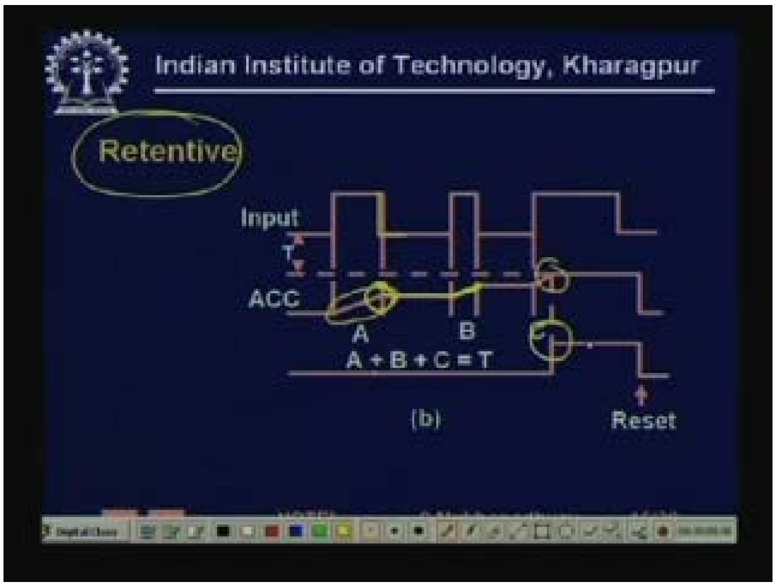
Not output to the timer, this is that every time the when the input goes high the output immediately goes high and it is an irrespective of the input it stays for only for a fixed time and then comes down. So, here also when this goes high this goes high and even if this comes down much earlier this 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, 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 realizes using our basic timer construct. So, that is point to ponder for you, next is that we again can classify timers into two ways one is called retentive timer another is called non retentive timer, so for a non retentive timer what happens is that see the see the input here goes up.

(Refer Slide Time: 25:31)



So, the timing register starts increasing, suppose it is on 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 a amount it has not reached is preset value. So, now the question is that what will happen to the off, but the input has come down, so what would happen to the timing register value at this point. So, in non retentive timer the timing register value is reset to 0, so every time you get timing signal it against time and in this time because the input stays one for a much longer. So, the present value is reached and when it is reached the timer output is the timer output is asserted. Here, the timer output is not asserted because the timing 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 notes the difference here.

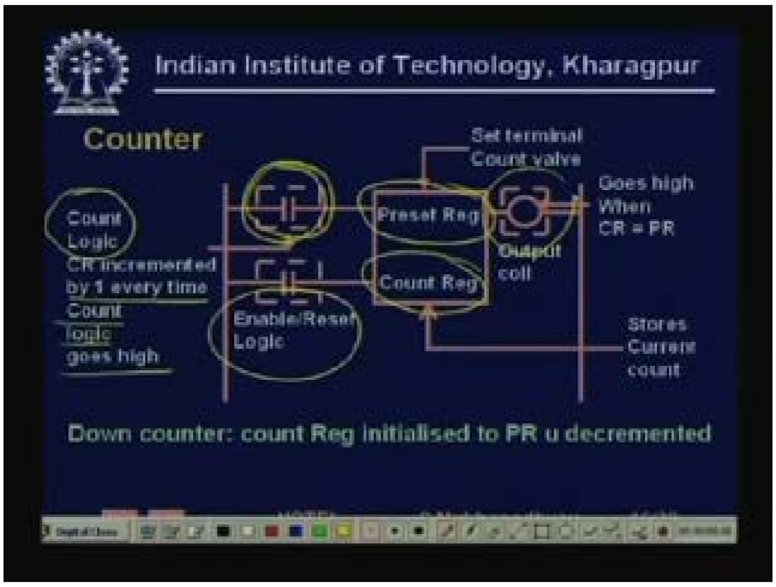
(Refer Slide Time: 26:56)



Here also, when the input goes one and then after sometimes comes down, so the timing register value, 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 count from the previous value, so that is why it retains its timing register value that is why it is called as retentive timer.

Finally, as this increases at some point for the may be for the next pulse it reaches the preset register value and at that point the timer goes high. So, this is the semantics or meaning of the way the retentive timer works, so these are the roughly the various kinds of timers available and now we take a look at what is known as the counter.

(Refer Slide Time: 28:24)


Indian Institute of Technology, Kharagpur

Counter

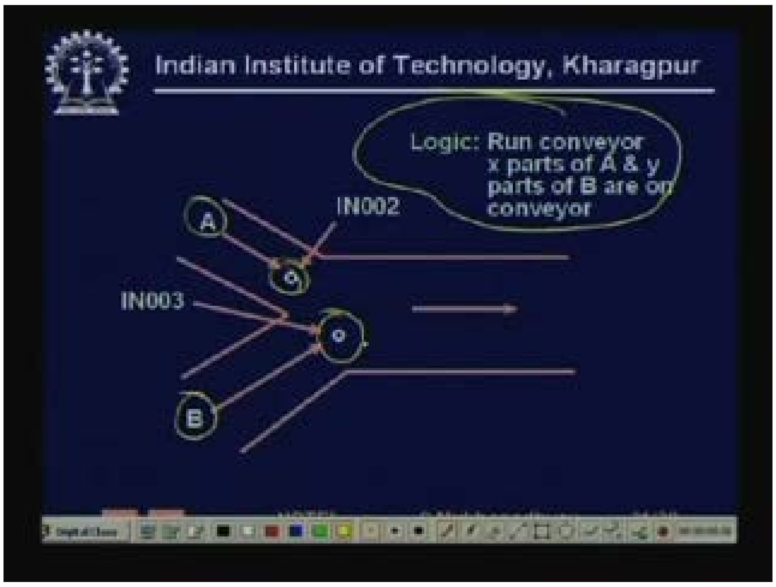
Down counter: count Reg initialised to PR u decremented

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 the preset register another 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 for the counter. This count you are trying to count something may be the number of parts produced or the number of parts arrived or something like that.

So, in which case with every event taking place there is you have to generate some kind of a pulse using again contacts and sensors. Now, those external pulses in this case will actually augment the counter history, so that counter is in place or the internal clock pulses which were coming at the timer here we have external pulses which are triggered by the events that we want to count. So, again you have enabled reset logic 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 1 every time count logic goes high, so every time this goes high from 0, the count register is incremented by 1. 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, so this is the meaning.

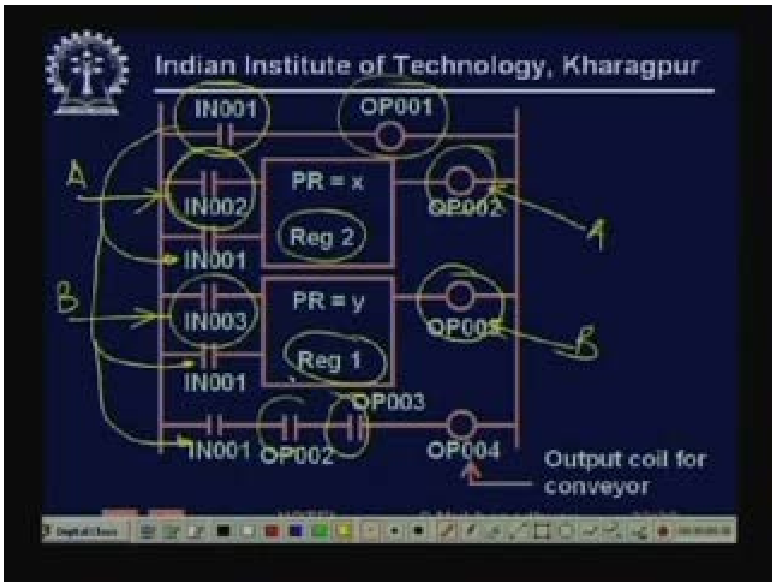
In some cases, it may be a down counter in which case the count register may be loaded with a preset register value and it may count down to 0 every time a pulse comes. So, it may be an up counter or it may be a down counter or it may have up or up and down count inputs by which it can have it can be also have up down counter, so excuse me some basic a very simple example.

(Refer Slide Time: 31:07)


Indian Institute of Technology, Kharagpur
Logic: Run conveyor x parts of A & y parts of B are on conveyor
IN002
IN003
A
B

You have a conveyor into which parts are supplied, so parts are coming from either this machine B for parts of type B are coming and parts of type A are coming. So, you want to implement this logic that to run conveyor when parts x parts of A at least x parts of A and y parts of B are on the conveyor. At that time, the conveyor will raw this is what we want to program and the arrival of a part of type B is asserted by this an arrival of a top part of type A is asserted by these two sensors, so what is the logic, so here this is an example of using counters.

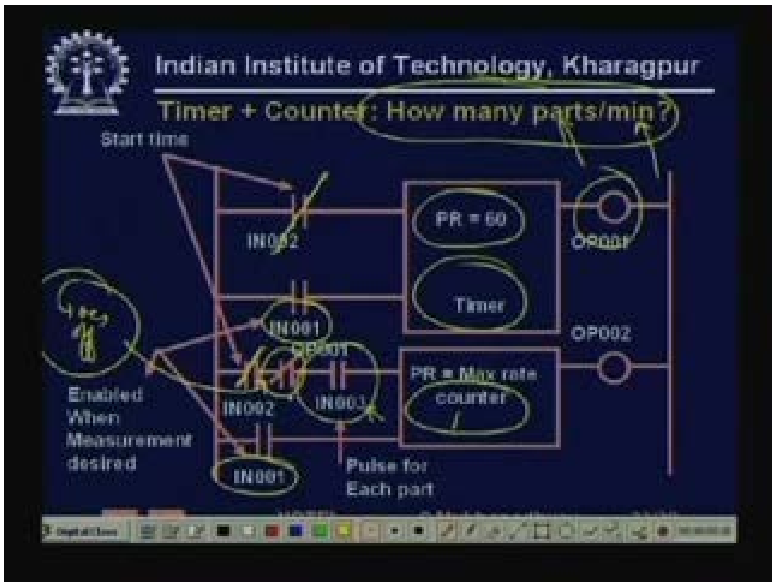
(Refer Slide Time: 32:02)



So, you see that we have two counters first of all this is the master switch the movement it goes on this output is asserted. Now, see that when IN 0 0 1 is asserted this IN 0 0 1 also is also is asserted and this IN 0 0 1 is also asserted and this IN 0 0 1 is also asserted. So, when this is 0, everything is 0, but when they goes one then the all these runs are enabled and both the timers are enabled. Now, the arrival of part of type B is signified by this and arrival of type part of type A is signified by this, so 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 the registers are incremented or the outputs the registers are incremented.

So, when the registers when both of when these register crosses this output goes high, so it is in series, so OP 0 0 1. So, when you put them in series you want that both of them must reach their preset count values only then the conveyer will run if you put them in parallel it means that if any one of them reaches that preset value, then the conveyer will learn. So, this is an example where you can use a counter for an industrial problem, this is a nice example of counting how many parts per minute are going on the conveyer, let us say.

(Refer Slide Time: 34:13)


Indian Institute of Technology, Kharagpur
Timer + Counter: How many parts/min?
Start time
IN002
PR = 60
Timer
OP001
IN001
OP001
OP002
PR = Max rate
counter
IN002
IN003
Enabled When Measurement desired
IN001
Pulse for Each part

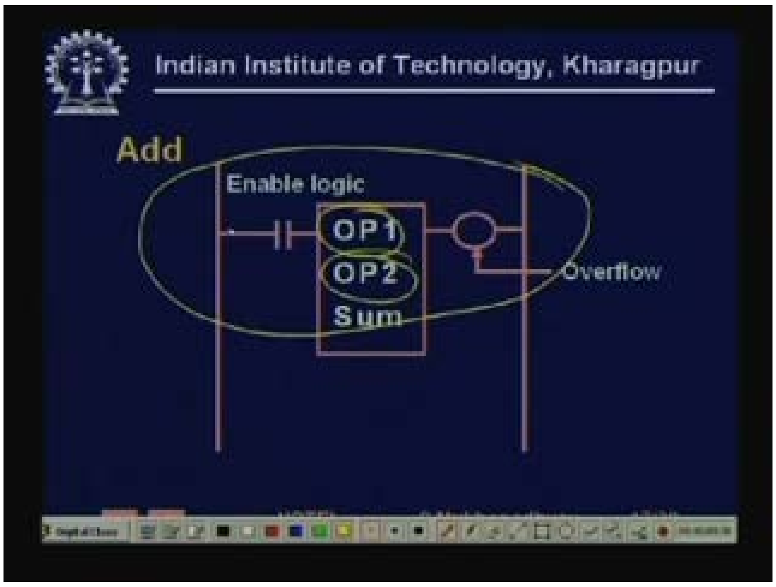
In a way, this indicates that whether what is the production rate, so it may be important management information to you know display what is the production rate, so now is this, therefore we want to count how many parts per minute. So, we want to not only keep count parts we want to also count it only over one minute of interval after we want the we have press start for this operation. So, obviously for the creating this one minute interval we need a timer and for counting the parts we need a counter therefore, it is a mixed timer counter example. So, here we have the timer and here we have the counter both have there 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 IN 0 0 1 is a switch, which you which one can press when he wants a measurement that over the next one minute how many parts pass. So, when these two are the when this main measurement desired contact is enabled at that time the both these timer and the counter are enabled. Now, what happens is that when you say start this is a 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 have passed, so you start time. So, this goes high and this goes high this does not go high, so this is low, so now 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 OP 0 0 1 goes open goes off. So, they this goes to 0 and then further parts are not counted, so when this is going to 0 this becomes open with this.

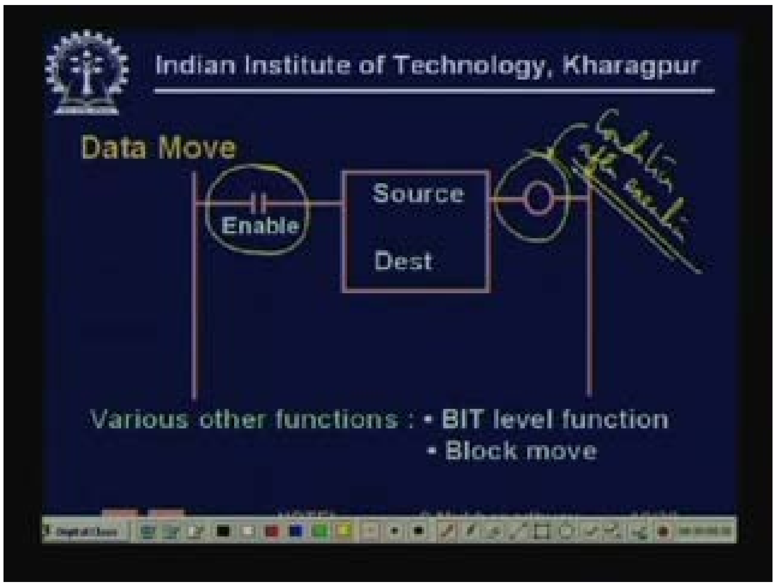
Here, you get a close or not it does not matter this cannot be zero because these are in series. So, therefore, at that time you in the in the counter you have the value you have the value of how many parts have passed over the conveyer in the last minute. So, we have covered the timer and counter in 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 ing OR ing. So, all these instructions are available just like a low level language and you can express them in various formats depending on the manufactories as I said. So, in our in our format, we are we are saying that by this diagram we are saying that this if you put this run.

(Refer Slide Time: 37:50)



It is a run for doing it, when this run will be executed basically two operands operant 1 and operant 2 will be added and their sum will be put at the location sum provided this enable logic is on. This output coil may be used for various purposes one of them could be that if there is an over flow if there is an over flow that has occurred during the summation, then it could go 1. 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 multiply you could have various other things. This is just one example of the arithmetic instruction, next is data move, so if you want to again you depicted as a run because in the RLL everything is the run.

(Refer Slide Time: 38:58)



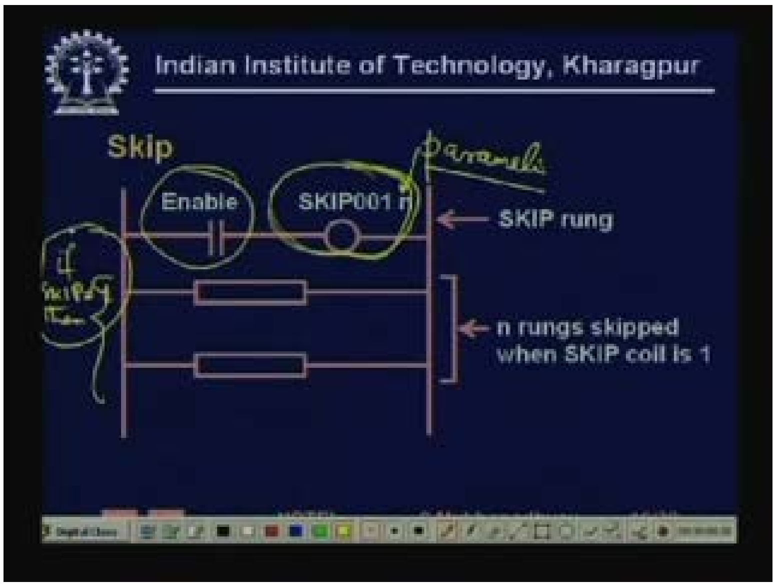
Data Move

Various other functions : • BIT level function
• Block move

So, in this case you have again the data move will take place depending on whether enabling logic is satisfied or not. Then, data will move from source to destination and may be if there is some address failure or something then or may be after the data has moved actually this can become one. So, this can also be used for indicating condition after execution, you can have, so this is a data move you can have various other functions. You can have bit manipulation functions you can have various kinds of block moves etcetera, you also have various kinds of logical instructions.

So, apart from data transfer, let the last instructions we have very important are program control instructions by which we want to we want to control we do not want to execute all runs of the of the RLL at all times. Rather, we want to control whether we can skip some of them or enforced some of their values, so we have typically to give an example we have a skip skipping facility, so again when this is enabled.

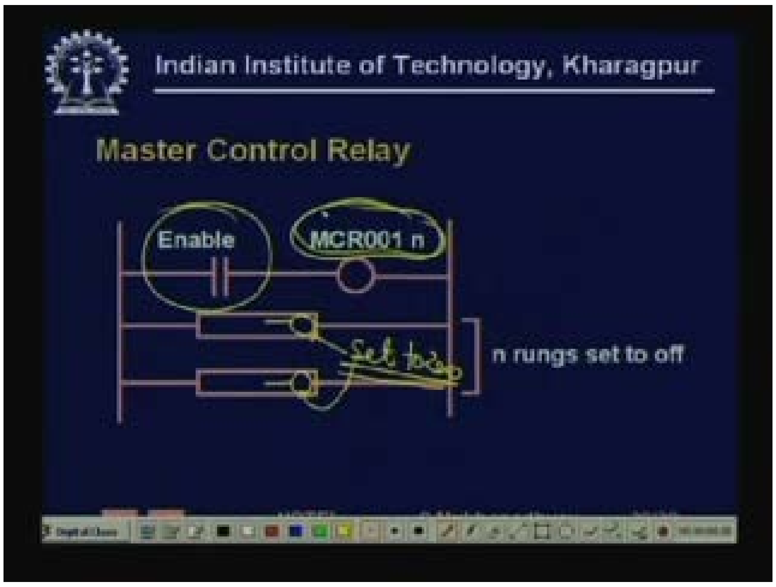
(Refer Slide Time: 40:25)


Indian Institute of Technology, Kharagpur

Skip

Then the skip 0 0 1, this is the this is the very special type of contact, so it is enabled when it is enabled, then what it means is that the next n rungs. So, you have skip 0 0 1 and n is a parameter, so it means that the next n rungs will not be evaluated, they will when the when the when this is high, so this is the meaning. So, it is like you know it is like if skip not equal to 1, then so this is like if skip not equal to 1, then this will be executed if skip is equal to 1 as long as stage one the next n rungs will not be evaluated. They will be maintained at the old value, similarly you can have another facility which you called the master control relay.

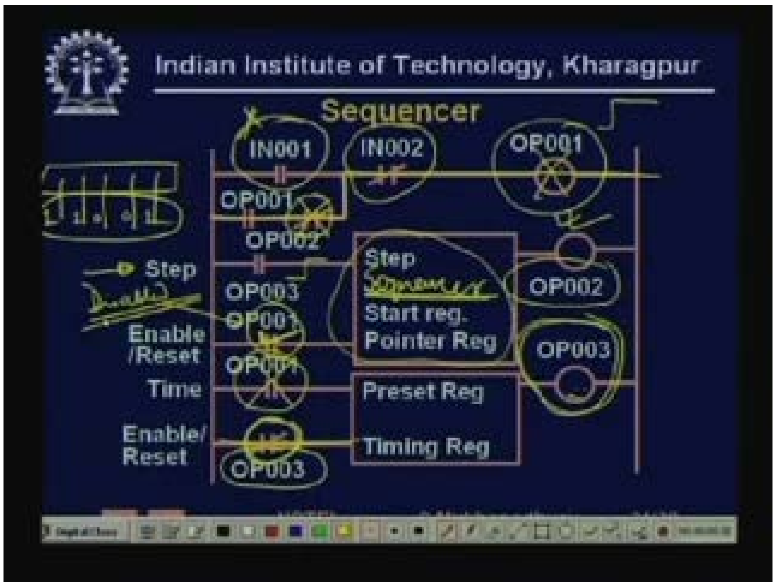
(Refer Slide Time: 41:36)


Indian Institute of Technology, Kharagpur

Master Control Relay

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 rungs will be set to there 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 0. So, without evaluating irrespective of the logic in this branch if this is excited then they will be all set to 0, but if this is not excited, then they will be evaluated normal like normal rungs according to according to normal PLC logic evaluation. This is a this there was some special instructions which also are there in a in a in a PLC RLL program language, for example this is sequencer.

(Refer Slide Time: 42:46)



So, a sequencer this is actually the sequencer this is the sequencer, so the sequencer is a block which can be separately programmed and which nicely executes a sequence of steps every time it is excited, so let us see what is happening here. So first of all this is a master switch which will become on let us assume that and suppose this is off. So, first of all what happens is that this goes high goes high, when this goes high and this 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 remain dense. Whenever there is a path here immediately and this is enabled. So, when this is enabled what happens is that this is enabled and this is this is now reset, no this is normally closed, this is normally closed.

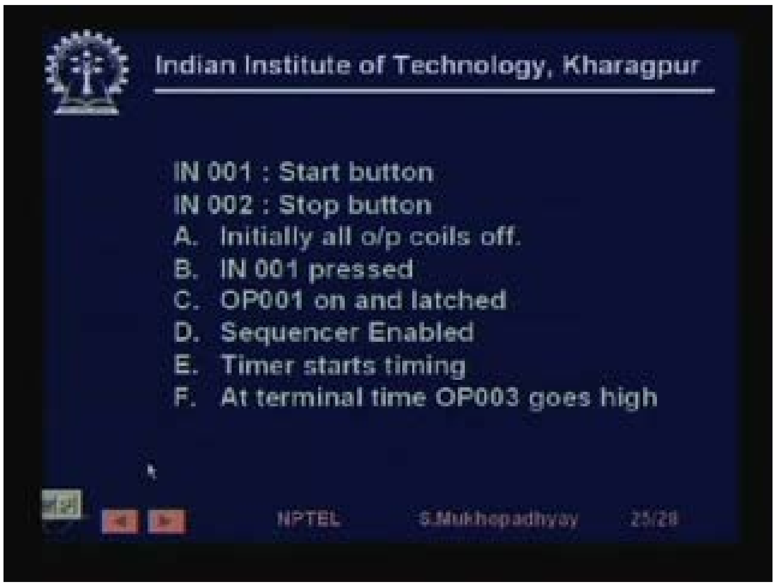
So, when this is closed, it is already enabled because this is off, so therefore this OP 0 0 1 goes on, so when this OP 0 0 1 goes on, it starts timing. So, after a preset timing this OP 0 0 3 will go high, now when this goes high immediately what happens is that you get a pulse here. So, then this sequencer the sequencer every time it get a gets a pulse into in to its step input, it executes a particular set of outputs it will exercise. So, particular set of outputs which are you know bit outputs which are stored in some register. So, there is a start register, so there is sequence of registers every time a pulse comes a new set of register outputs.

So, you have say m register may be these are know three different or eight different valves. So, you have stored some value, so maybe even the e in the first step this is 1, this is one this is 0, this 0 and this is one that is, so where you have you your program data. So, when step 1 is executed, then these outputs will actually go to the feed, so the first pulse has come and the first step is executed after the first step is executed in the mean time you see that. So, the first step is executed and it is and it is waiting there, now when this has become 1, immediately this became 0, I need become 0 again, so again it is enabled, so now what happens is the n output 0 0 1 is already on.

So, therefore again it times, so basically what happening is that it is with this arrangement, it is continuously timing and then and the movement this OP 0 0 3 goes high after the timing interval this gives a pulse here. So, a step is executed in the sequencer and then because its latch like this, so immediately when this goes high, this is again becoming reset and then once it is becoming is reset e, it is this one is enabled and therefore its again timing. So, it is continuously generating some timing intervals and at the end of each interval you are getting a pulse which is executing the start register executing the sequence register of the sequencer.

At the end of the sequence, the sequence has a fix number of steps, so at the end of the sequence this will go high. So, once this goes high then what happens is that this one goes off, when this one goes off. Then, this one goes off and when this one goes off then there is no more timing here and this itself that is a sequencer itself is disabled, so this is the way a sequencer works, now this is this is what is expressed here.

(Refer Slide Time: 47:56)



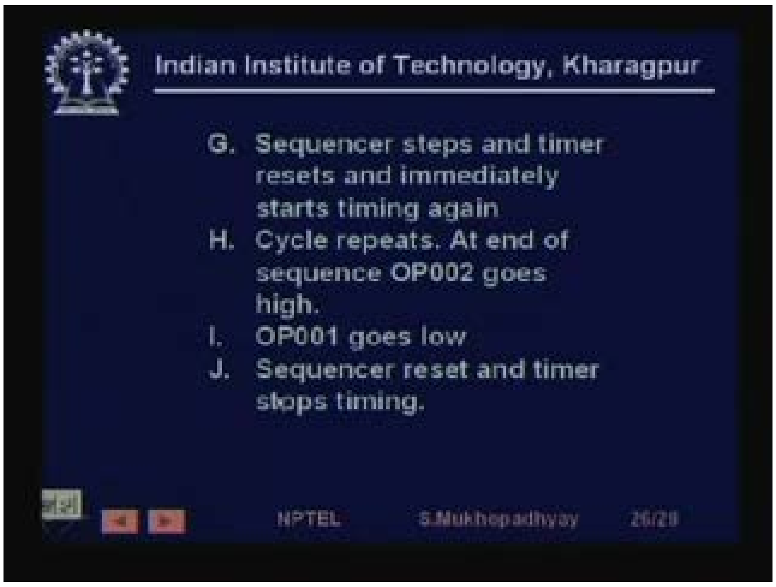
Indian Institute of Technology, Kharagpur

IN 001 : Start button
IN 002 : Stop button
A. Initially all o/p coils off.
B. IN 001 pressed
C. OP001 on and latched
D. Sequencer Enabled
E. Timer starts timing
F. At terminal time OP003 goes high

NPTEL S.Mukhopadhyay 25/29

So, if you press start button IN 0 0 1 is the start button IN 0 0 2 is actually a stop button you could make this sequence stop at any times. So, initially all output coils off then IN 0 0 1 is pressed, then output 0 0 1 on and latched, then sequencer enabled then timer starts timing, then a terminal time output 3 goes to it is high.

(Refer Slide Time: 48:23)



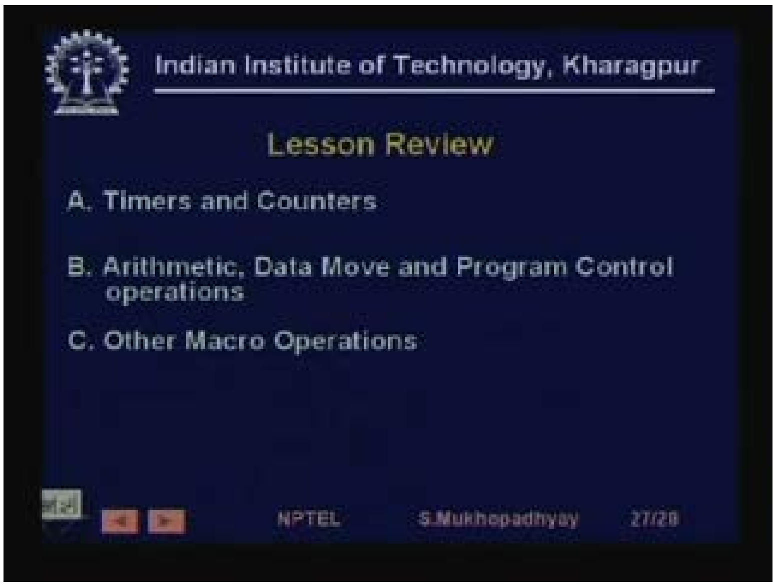
Indian Institute of Technology, Kharagpur

G. Sequencer steps and timer resets and immediately starts timing again
H. Cycle repeats. At end of sequence OP002 goes high.
I. OP001 goes low
J. Sequencer reset and timer stops timing.

NPTEL S.Mukhopadhyay 26/29

Immediately, sequencer steps and timer resets and immediately starts timing again, we explained, the cycle repeats at end of sequence OP 0 0 2 goes high, which means that OP 0 0 1 goes low and the sequencer resets the timer stops timing, so this is how it works.

(Refer Slide Time: 48:45)



Indian Institute of Technology, Kharagpur

Lesson Review

A. Timers and Counters

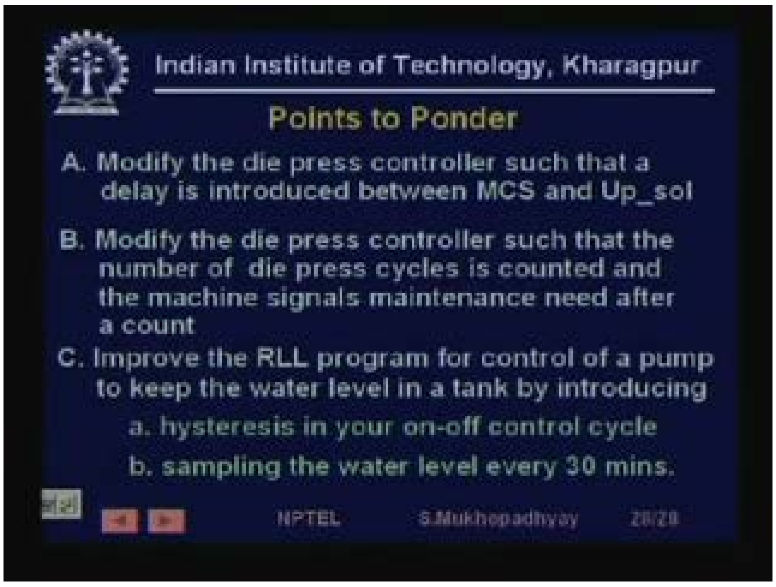
B. Arithmetic, Data Move and Program Control operations

C. Other Macro Operations

NPTEL S.Mukhopadhyay 27/28

So, we have come to the end of the lesson and in this lessons we have seen the various timers and counters and we have also seen some arithmetic that data move and program control operations. 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 etcetera which we have not seen, so for coming to the end, we have the usual points to ponder for example.

(Refer Slide Time: 49:28)



Indian Institute of Technology, Kharagpur

Points to Ponder

A. Modify the die press controller such that a delay is introduced between MCS and Up_sol

B. Modify the die press controller such that the number of die press cycles is counted and the machine signals maintenance need after a count

C. Improve the RLL program for control of a pump to keep the water level in a tank by introducing

a. hysteresis in your on-off control cycle

b. sampling the water level every 30 mins.

NPTEL S.Mukhopadhyay 28/28

You could try to modify the die press controller such that a delay is introduce between that is after the master control switch is 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 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 the machine. So, you want to count every time a complete cycle of dieses one going up and one going down is completed you want to count them and after it reached, it reaches a count you want, you want to stop the machine for maintaining.

Similarly, you could improves the RLL program, which is which we said in our earlier points to ponder RLL program for control of a pump to keep the water level in a tank by introducing a hysteresis in your on off control cycle. Also, sampling the water level every thirty minutes that is not continuous sampling, the water level every 30 minutes, so that is all for today, thank you very much, in the next lecture, we will we will continue with PLC s.

(Refer Slide Time: 50:58)

Preview of next Lecture

(Refer Slide Time: 51:02)

Lecture - 21

A Structured Design Approach
to
Sequence Control

So, welcome to lesson twenty-one of the course industrial automation and control

(Refer Slide Time: 51:19)

Indian Institute of Technology, Kharagpur

Lesson 21

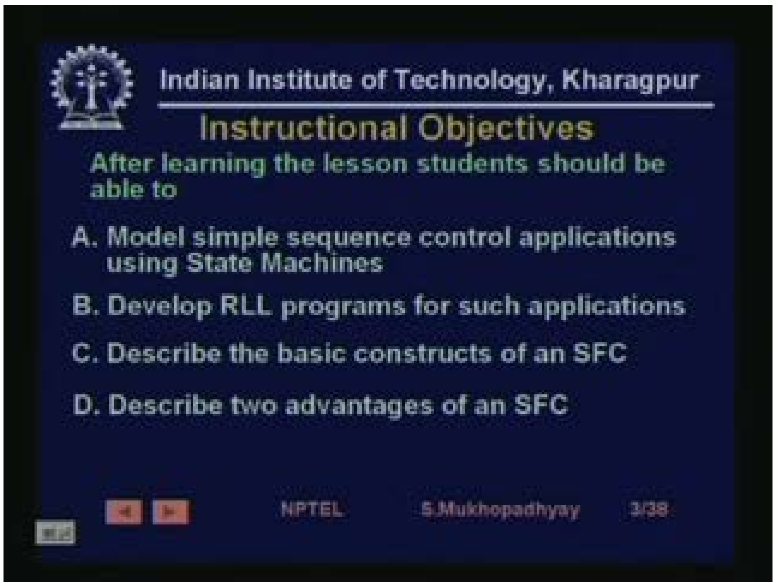
A Structured Design Approach
to
Sequence Control

NPTEL S.Mukhopadhyay 2/38

In this lesson, we are going to learn a structure design approach to sequence control, so for we have mainly seen the programming constructs have seen small program segments timer counters. In this lesson, for the first time we will see that given a practical problem how to how to study the problem how to what are the steps that you go through to finally arrive at an RLL program.

So, this will be followed using a very systematic approach because that I have already told you that industrial control applications are very critical in the sense that if you have programming errors in them. They can be very expensive in terms of money or in terms of even get cost human lives etcetera. So, it is always good to have a very systematic design process by which you can decompose a problem and finally arrive at the solution, so we will look at the instructional objectives.

(Refer Slide Time: 52:18)



Indian Institute of Technology, Kharagpur

Instructional Objectives

After learning the lesson students should be able to

A. Model simple sequence control applications using State Machines

B. Develop RLL programs for such applications

C. Describe the basic constructs of an SFC

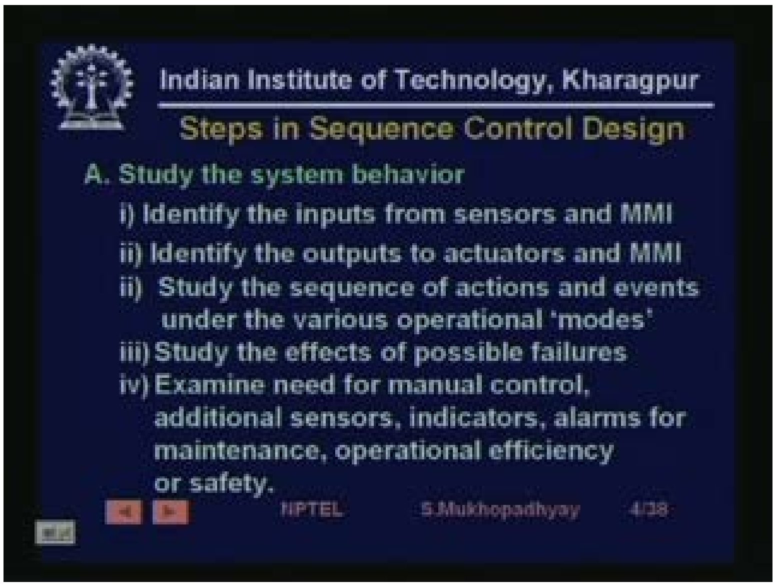
D. Describe two advantages of an SFC

NPTEL S.Mukhopadhyay 3/38

The instructional objectives of this lesson is are first let you able to model simple sequence control applications using state machines state machine is a is actually formal method. We advocate the use a formal methods because English can be very ambiguous sometimes contradictory also. So, we have to model it using methods which have which are unambiguous consistent do not content contradictions and are also easy to understand and develop.

Then, from these formal models we have to develop RLL programs for such applications and for doing this there are certain apart from the RLL programs, there are some modern programming construct which are being made available. One of them is a SFC or the sequential function chart, so we will take a look at that and also understand some of its advantages, so that is the these are the instructional objectives of this lesson.

(Refer Slide Time: 53:16)



Indian Institute of Technology, Kharagpur

Steps in Sequence Control Design

A. Study the system behavior

i) Identify the inputs from sensors and MMI

ii) Identify the outputs to actuators and MMI

ii) Study the sequence of actions and events under the various operational 'modes'

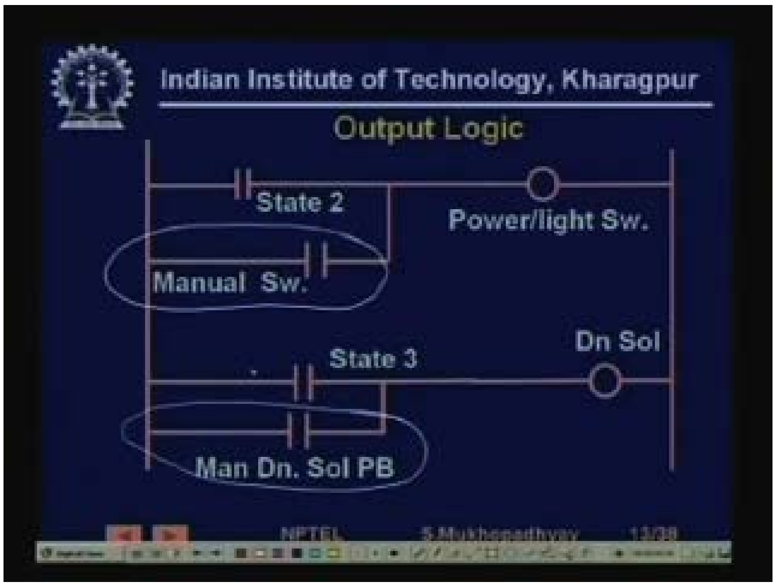
iii) Study the effects of possible failures

iv) Examine need for manual control, additional sensors, indicators, alarms for maintenance, operational efficiency or safety.

NPTEL S.Mukhopadhyay 4/38

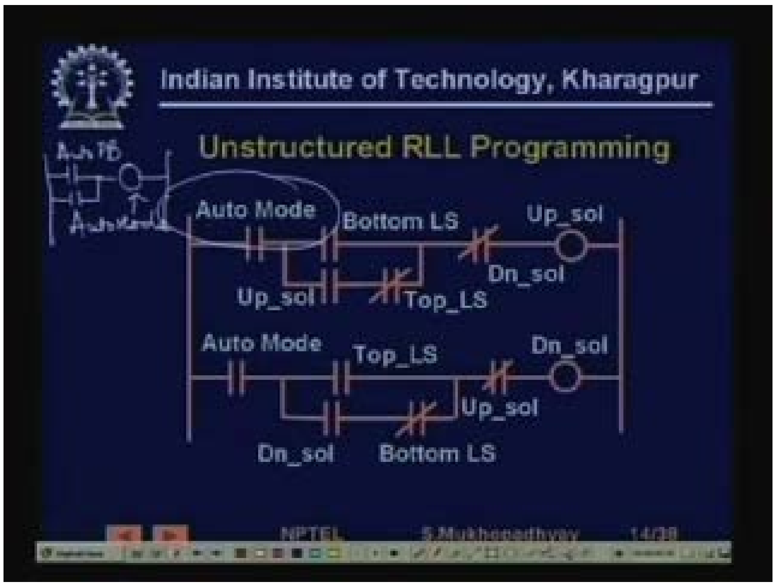
So, now let us go through the steps in basic broad steps in sequence control design, so first is to study the system behavior this is a very critical step and most of the errors that happen in any programming exercise. Not only has this kind of industrial automation programming any programming mainly arisen from the fact that the programmer or the developer did not understand the system well. So, this is a very important step, so one need not really think too much about the logic, one should think about the logic while he is drawing the diagram after that the programming becomes automatic this is very useful. So, now next we will have the output coil output logic output logic is very simple very simple especially in this case.

(Refer Slide Time: 54:04)



So, the output logic says that if you are in state two then power light switch should be on as we have given in our output table. So, only thing is that look here that we have we have also added some manual switch you know it can be sometimes we may need to we may need to check we may need do things manually also. So, the power light switch will be on here we have put a manual switch, so if the PLC is running then if you press the manual switch, then also power light switch may be made on. Similarly, we can have a manual down push button, so we can this is just to demonstrate that you can put additional logic to include manual operation of the system. So, in this otherwise this program simply says that while you are in state three down solenoid will have to be activating very simple.

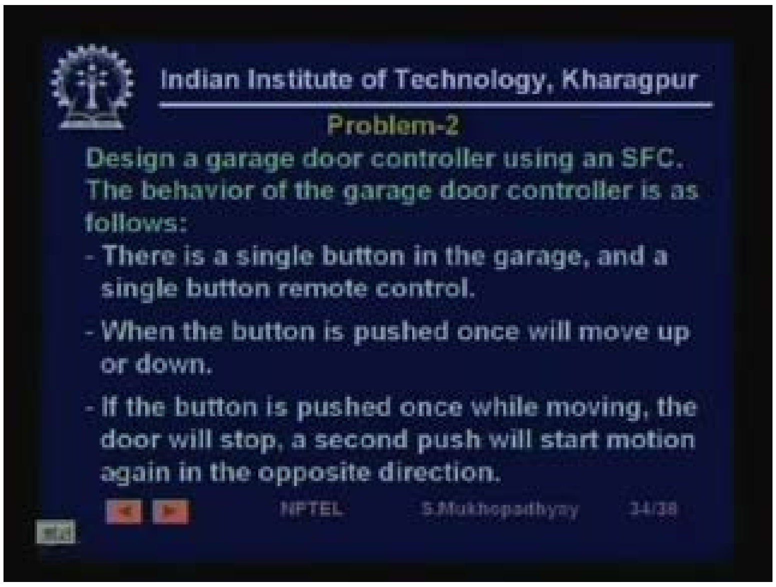
(Refer Slide Time: 55:05)



Compare this with the kind of programs that we are written earlier, in fact for this process itself we had written some program. So, there we did not have any concept of states and transitions, we were directly trying to write outputs in the form of inputs. Now, the problem with this kind of problems is that they are get here systems generally have memory that is why you need the need the concept states it is not that if you get a certain kind of inputs you will have to produce certain kind of outputs. It depends on which state the system is in, so the concept of state is very important and well you can you can bring it down in bring.

It possibly in certain cases during some temporary variables, but the kind of here you see if you look at this program this program says it is very complicated logic and I am not even 100 percent sure. It is very difficult to 100 percent sure whether this logic is full proof; it says that if the auto mode by the way this auto mode is actually, it is an auxiliary contact corresponding to some logical variable which you can set where by a simple rung. If it is auto PB and then you have an auto mode coil and then you have this is auto PB and here you can have auto mode. So, you can have an auto mode coil and this will be an auto mode auxiliary switch, so the PB can be released, so this is a sort of you know persistent input.

(Refer Slide Time: 56:44)



Indian Institute of Technology, Kharagpur

Problem-2

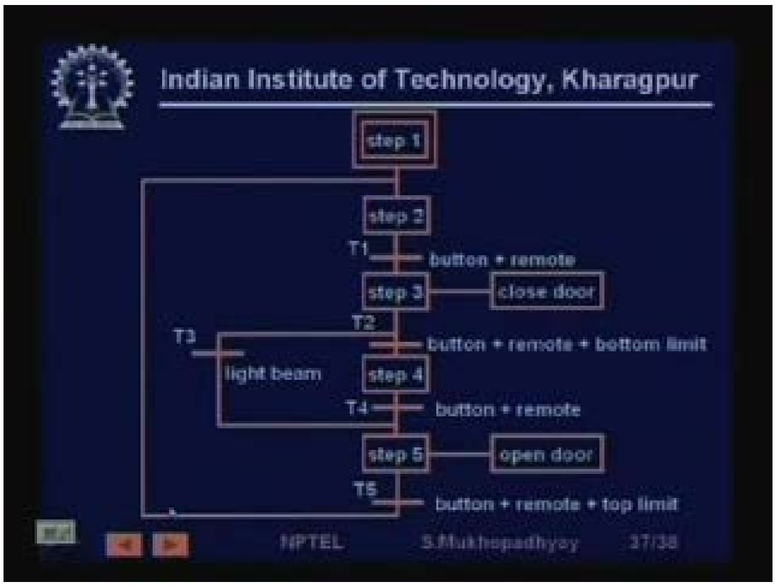
Design a garage door controller using an SFC. The behavior of the garage door controller is as follows:

- There is a single button in the garage, and a single button remote control.
- When the button is pushed once will move up or down.
- If the button is pushed once while moving, the door will stop, a second push will start motion again in the opposite direction.

NPTEL S.Mukhopadhyay 34/38

So, there is a single button in the garage and a single button remote control, so you can have either a button pressed in the garage or you can have a button pressed in button pressed from the remote, so there are two kinds of buttons.

(Refer Slide Time: 57:00)



So, you have either a button pressed from the garage or from the remote, then will go to step three and will start closing the door this is the this is the output. On the other hand, if button has been a button been pressed, either some local or some remote, so if we press a button what will happen, then the door will stop.

On the other hand, if the limit switch is made what will happen the door will stop, so if either a button has been pressed or the bottom limit switch is reached then immediately the door will stop. If the light beam is interrupted then it will not only stop it will actually reverse, so immediately it is going to reverse. Here, what happens is that if you pressed a button and to stop it, then if you press it again, then it will go to reverse. So, this is a you see I we have we have captured the behavior of the garage door in the in this form using SFC. So, you can do you can do a similar thing also for the traffic light that and an exercise, so that brings us to the end of this lesson.

Thank you very much and see you again for the next lesson, bye.