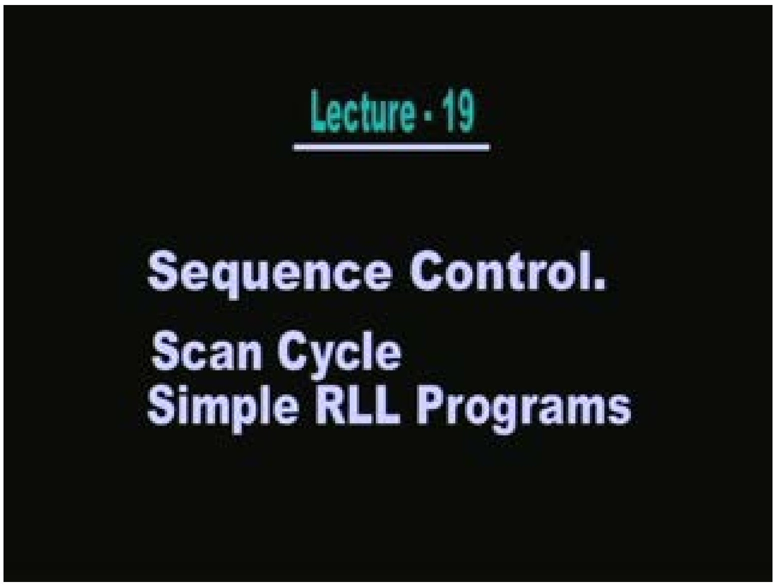
Industrial Automation and Control

Prof. S Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 19
Sequence Control Scan Cycle Simple RLL Programs

(Refer Slide Time: 00:44)



Lecture - 19

Sequence Control.

Scan Cycle
Simple RLL Programs

Good afternoon and welcome to lesson number nineteen, where we shall discuss we shall continue to discuss sequence control.

(Refer Slide Time: 01:03)

Indian Institute of Technology, Kharagpur

Lesson 19

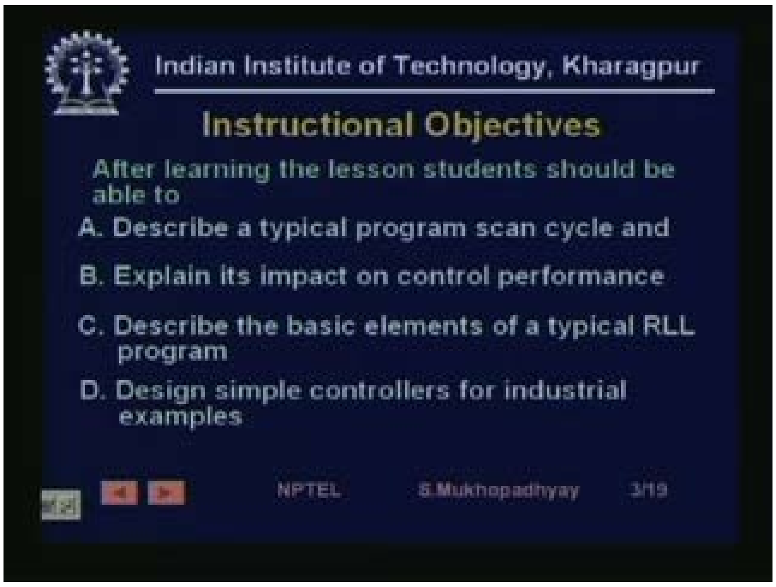
Sequence Control .

Scan Cycle
Simple RLL Programs

NPTEL S.Mukhopadhyay 2/19

Today, we will discuss a scan cycle that is scan is a program execution, so we will discuss how programs are cyclically executed. We will discuss the nature of it is impact on performance, and we will also discuss how to write some simple relay ladder logic programs.

(Refer Slide Time: 01:35)



Indian Institute of Technology, Kharagpur

Instructional Objectives

After learning the lesson students should be able to

A. Describe a typical program scan cycle and

B. Explain its impact on control performance

C. Describe the basic elements of a typical RLL program

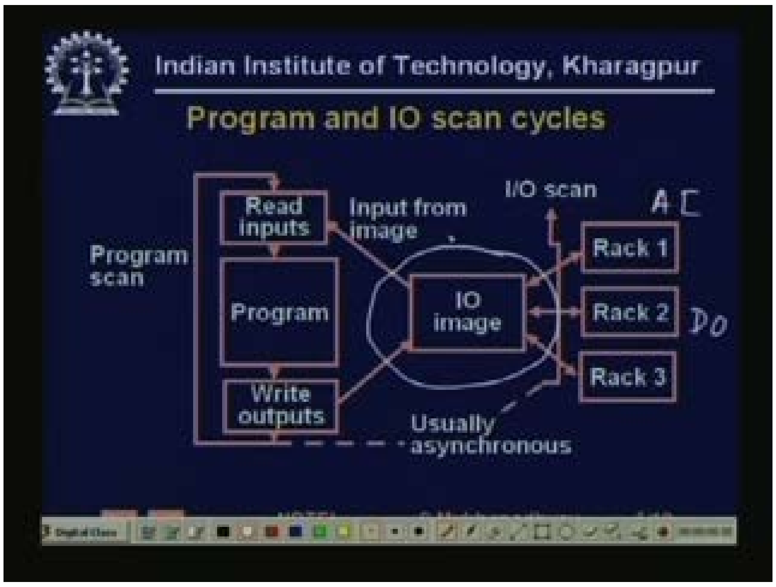
D. Design simple controllers for industrial examples

NPTEL S.Mukhopadhyay 3/19

So, as usual we give the instructional objectives first after today's lesson one should be able to describe a typical program scan cycle that is describe what takes place how the program works cyclically. We would be able to explain the impact of this kind of execution on the control performance typically on the fastness of response that is how fast the programmable logic controller responds to change is inputs with change are input. Then, we will go on to describe the basic elements of a typical relay ladder logic program, today will look at the very simple elements of the program.

In future lectures, we shall go on to see more advanced programming elements and finally, we will take two of the simplest example and see how exactly a relay ladder logic programs segment could look like. Also, we would learn to interpret a particular RLL program that is after we if you are given an RLL program then we have to know what it means and how the inputs and outputs will change with time, so we will do that, so that is the agenda today.

(Refer Slide Time: 03:06)

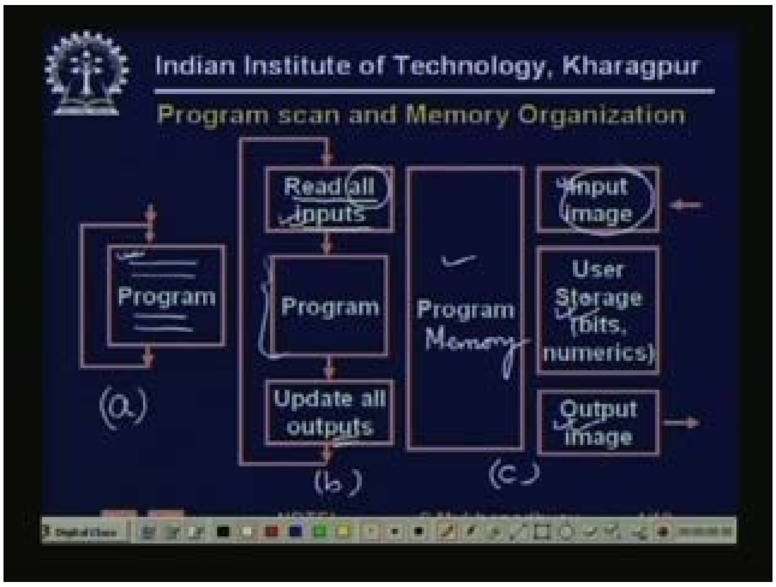


Indian Institute of Technology, Kharagpur

Program and IO scan cycles

Having said that, we are continuing on the same note from the last lessons if you recall in the last lesson, we had ended at a point when we had listed some of the hardware elements of a PLC system. They are namely back plane CPU memory the different IO modules wiring programmer man machine interface, etcetera. So, today we take a software focus and start looking at how the basic program execution goes in a PLC, so this one obviously, in like a most control programs a PLC program also works cyclically.

(Refer Slide Time: 04:11)



Indian Institute of Technology, Kharagpur

Program scan and Memory Organization

So, as it happens, basically this is one view say view a in view we are indicating that the program simply work cyclically. So, this program which is as which is a set of statements execute cyclically that is the execution begins here then next statement and so on till the last statement and then again the first statement will be executed. Now, what happens within this overall program box that is what the functions which are carried out in the program? So, as we as we all know that PLCs are real time embedded systems they are they are reactive in the sense that they accept inputs from the external world and produce outputs which go to the external world namely the machines.

So, obviously, the program has to be always aware of what changes are taking place in the external world namely the various variables that are being controlled in the machines. So, a typical programs cycle includes three steps first is this is an expanded view of a namely b, which shows that typically three activities take place during a program execution. So, for first one is read all inputs mark the stress on all which means that it is not that, so initially it means that initially all the inputs are read one after the other and there values are stored and then the then the execution of the logic begins.

So, this is a major difference with any standard program which you could write in a microprocessor in a microprocessor. You could write you could a write a program, where the various processing statements addition subtraction multiplication comparison whatever you have.

You could intersperse them with the input and the output statements, but, typically in PLC execution that does not take place. So, in PLC because of the fact that initially if the PLC programs you know there is a good there is always a stress on keeping think simple. So, those errors do not occur, so at the cost of some flexibility of programming we are intending to achieve some simplicity and that is that is a sound strategy provided performance is not affected. In this case it happens that you know typical microprocessors statement execution times run in micro seconds, while with PLCs the kind of things that we trying to control are either mechanical or thermal or chemical.

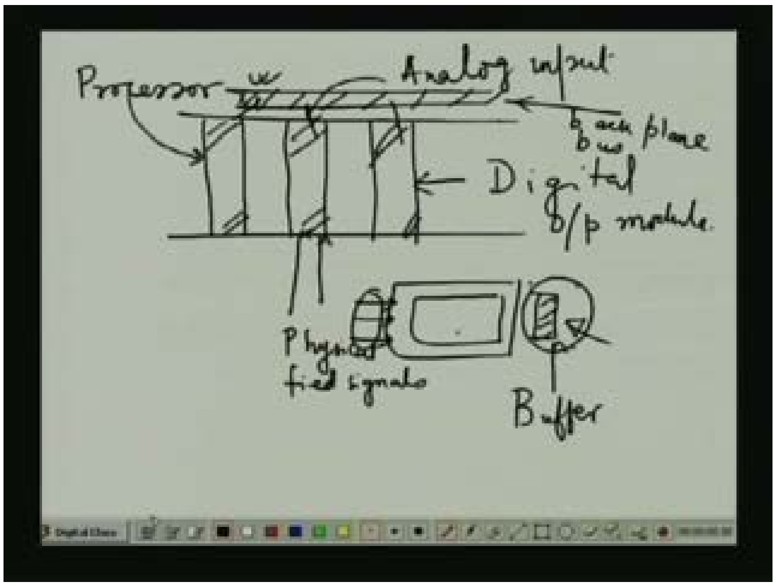
So, the time constants are usually, so large that this subtle difference between reading all the inputs at one time makes hardly any effect. So, therefore, typically for PLCs all inputs are read at one short at the beginning of the scan cycle one cycle of these three steps would be called as scan cycle after all inputs are read the program logic gets executed. Now, during the execution of the program logic various outputs are produced, so it happens as outputs are produced they are return to some temporary memory, but, but just like inputs are read at the same time. Similarly, it is not it is not that whenever some outputs are produced and some output values are stored in memory, they would be they would be going to the external world that does not happen.

So, after all the logic of the program is executed at the end all the outputs are output values, which go to the external world in the sense of you know various switches motor starters lamps indications. They are all updated at the end of the cycle and then the cycle repeats namely the next cycle of a inputs are read. So, this is the typical way that a PLC programs can takes place and so c is c is a kind of shows the kind of memories that are used in the PLC system. So, you have four kinds of memories first you have what is known as program memory where the logic program is stored that is the first one, then you have what is known as an input image.

So, when the inputs are read they are for the for the program logic evaluation, these input values will have to be there from time to time during the evaluation of the logic. So, therefore, they have to be necessarily stored in memory, so where the input values are stored is called an input image this is the second use of the memory. Similarly, as outputs are produced as the logic gets computed outputs are produced and they have to be stored till the time that they can be transferred to the external world.

So, for that output image memories used an as we know that for all computation we need we need we need certain in addition to the input and output variables we need certain temporary variables. So, for storing these things we need some other additional memory, so this shows how the overall memory available in a PLC system is used. Having understood that, we have to understand that this kind of a program execution we are in this in this figure we are what we trying to show is that many times it will happens? It will happen, if you if you let us look at the if you have understand this diagram before that you have to understand.

(Refer Slide Time: 11:26)



A typical you know PLC hardware organization, so what happens is that in a PLC the various units this we did not discuss in the earlier lesson. So, this various units are actually organized as modules, so you have what was known as you now racks. So, this is one module similarly, there could be another module by the side of it. So, this is one module this is another module similarly. So, this could be a this could be the main processor module, similarly, this could be an analog IO module input module, let us say similarly you can have digital input digital output various kinds are modules.

So, what happens often is that. So, now, you see this processor the external variables that is the various signals get interface to these modules and similarly, this could be a digital output module. So, what happens is that these external signals are these modules are again self independent they are they are they are as such independent circuit boards. So, they often have processors on them, so what happens is that the processor here the processor here actually what happens is that you now here is the bolt suppose we are taking. So, you have all these circuitry here in this side the physical signals are interfaced physical or field signals and on this side. So, these are physical signals and on this side symbolically may be in a buffer this is a buffer the values of the signals are stored.

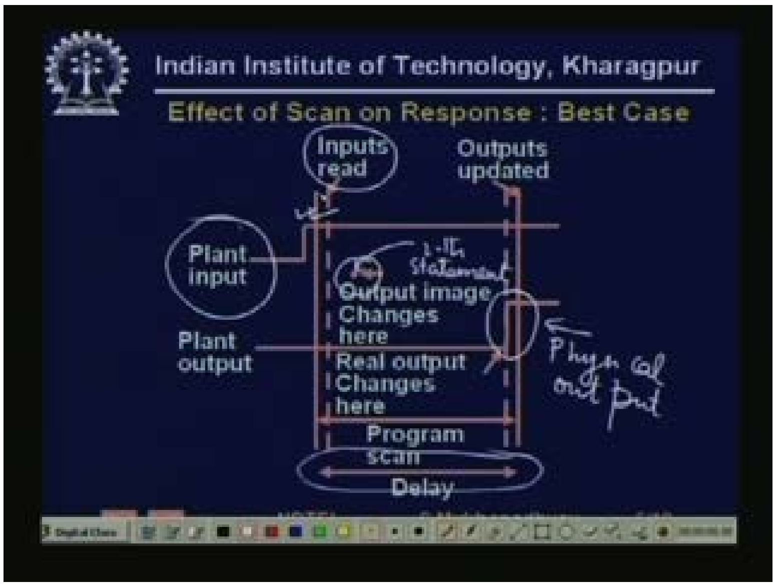
So, when the processor module reads inputs it is not that it reads this physical signals, it actually reads these buffer values which are stored, which are continuously kept in the buffers by the IO curves. So, this IO curves often work under the control of the processor it may work at the control of the processor or it may continuously or it may work independently cyclically itself. So, all these IO curves and actually there is there is a what is known as that is what we referred to in our earlier lecture as a back plane or a bus which is nothing but, a set of conductors to which all of these are connected.

So, when I say that have that that a processor reads an input it actually reads these buffer values over the bus. So, actually two activities keep on taking place, tele first one is that the physical signals keep getting transform from either to this buffers and this buffers could be physically situated either of the curve or it could even be situated in the processor memory. So, it may so happen those we have studied microprocessor will understand this maybe. So, happen that this curves every few units of time convert this values and then using some interrupt mechanism they will transfers these values to some part of the processor memory.

So, what I am trying to stress is that here there are two kinds of cycles going on one kind of cycle continuously cyclically samples the physical inputs and transform them to a part of the memory which we call the IO image and the processor. Here, it reads it actually reads from that image, so having understood this what is happening is that you see these are the various racks these are the various racks. So, these are the IO racks this could be an analog in this could be an analog input, similarly this could be a digital output various kinds of things.

So, there is a scanning mechanism which goes on which continuously updates either reads from the racks depending if it is an input rack or rights to a rack from a part of a memory this memory can be at this scanner can be at the curves at the various places. When the main PLC programs scanning actually reads these IO images from this IO image is actually stores reads the inputs and whenever it actually produces outputs. It also writes to this IO image and so it keeps writing to this IO image and then finally it get transferred to the racks. So, this is how the basic activities go on in a PLC, so now we have to see that what is the result of this activities who this kinds of scan scanning what kind of response what kind of impact it has on the on the various input and output updating. Typically, we are interested to know how much of delay we can except in responding to an input change.

(Refer Slide Time: 18:00)

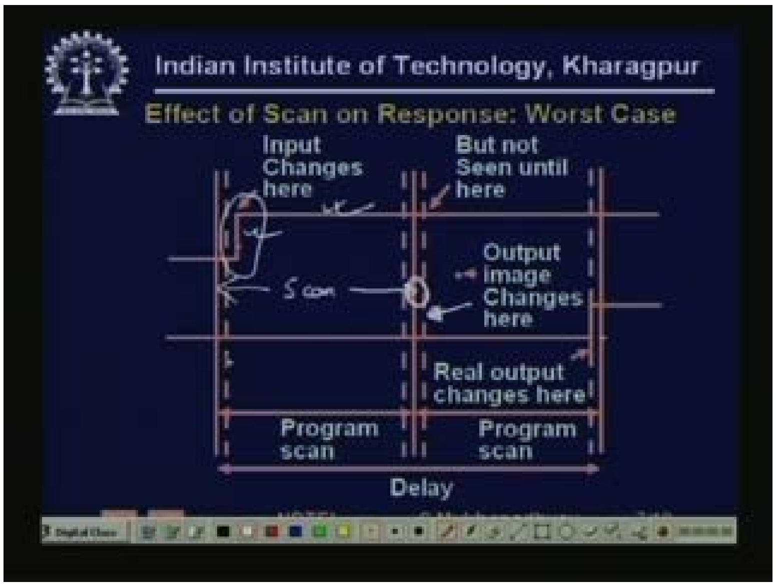


So, imagine that this is the there could be a best case and there could be a worst case, so we have a first looking at the best case. So, what could happen is that the plant input can change at arbitrary period of time we cannot control that right. So, suppose the plant input changes just before and in the inputs are read, so then immediately after it changes it will be read that is this change will be reflected and will come to the processor IO image very quickly. Then, then the while the program will be is the well while the program will be executed then in computing the output value the latest change of the input value will be taken care of will be accounted for.

So, possibly due to this input change at the end of the execution there will be an output change. Now, since it may actually happen suppose this is the statement this is the I th statement that is the effect of the of this input is used to compute some output at the I th logical statement. So, then what will happen is a during execution here is the this is the point where the I th statement gets executed. So, actually in the processor memory the output changes at this point of time, however it does not go to the physical world till the whole program execution has completed as we have said. So, therefore, the output a physical the physical output which goes to the process gets affected only at that time and there is a this amount of delay that is bound to occur.

So, there is the best case response time is the delay of one scan which depends on many factors like this speed of the PLC execution also the length of the RLL program. This is the best case it is called the best case because the input was because the input was read immediately after it changed. So, there was no delay in sensing the change in the input in the next case as we shall show that we shall see the worst case where this is just the opposite.

(Refer Slide Time: 21:04)

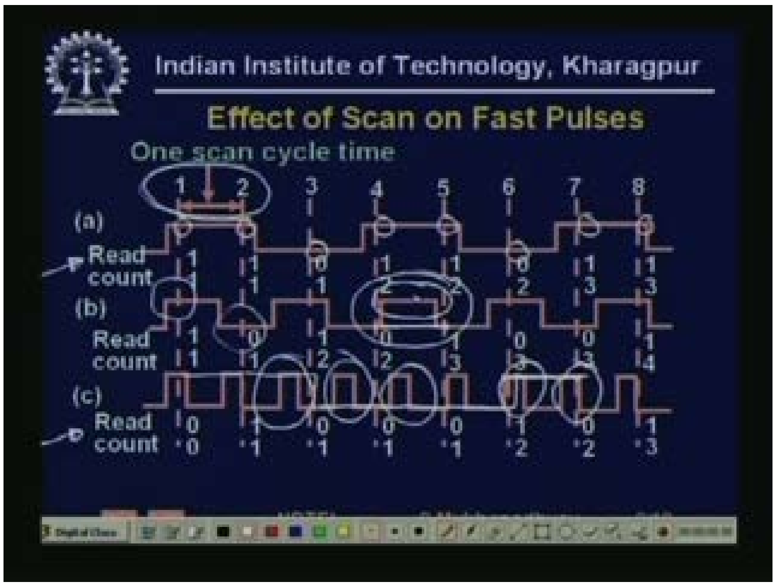

Indian Institute of Technology, Kharagpur
Effect of Scan on Response: Worst Case
Input Changes here
But not Seen until here
Scan
Output image Changes here
Real output changes here
Program scan
Program scan
Delay

So, this is the opposite this is the worst case, so what is the happening here unfortunately input has changed just after a reading cycle was completed. So, therefore, this change of the input was not sensed by the PLC in this scan cycle this scan cycle, it did not sense that the input has changed.

So, therefore it computed by this time it had computed and sent out the outputs once, but however, those outputs did not reflect the new value of input which occurred just after the input in the in this cycle to place. So, that is why it is the worst case, so it occurred, so there is a there is a maximum possible delay, however, at this point at this point in the second scan cycle inputs will be the read again. This time change is going to be sensed and is now after this after having sensed this there will be again we have one cycle of delay.

So, therefore, the worst case of delay is two scan cycles, so that is why be depending on the kind of response time we one has to decide based on an application whether such delay times are acceptable or not. So, we shall see that in certain cases these such delay times may not be acceptable in which case we have to take additional measure in the sense that you might put additional curves or additional hardware for taking care of them. So, this is the case where you get very fast pulses and one has to count those pulses.

(Refer Slide Time: 23:11)


Indian Institute of Technology, Kharagpur

Effect of Scan on Fast Pulses

So, in this case what we see is that suppose the scans the one scan time is this much if we have to count a the number of pulses which are arriving on pulse strain typically such pulses come from shaft angle and coders. If we have to count them using the PLC itself then if we can see that is the pulse strains are slow enough that is the pulse width that sufficient then for this width of the pulse strain all the level changes will be sensed.

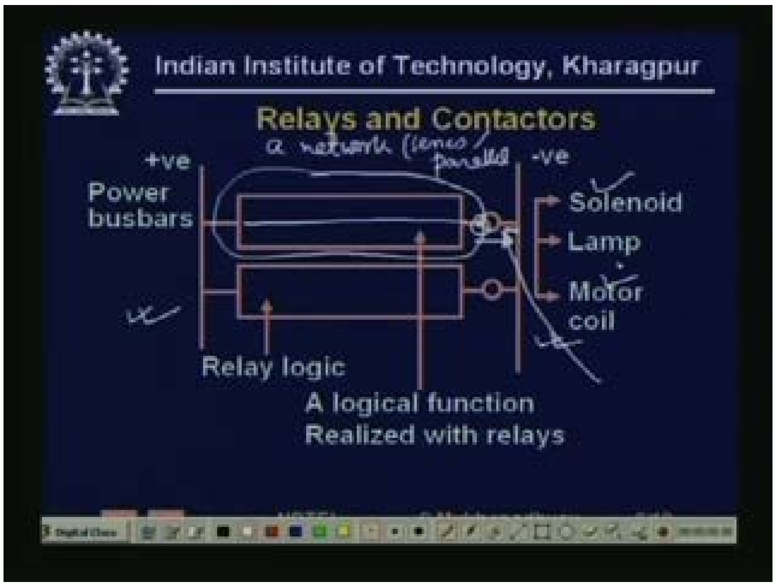
So, we are essentially going to count the number of pulses by counting the number level transitions. So, then our count is going to be correct and if we compute the speed best on that the value of speed we get is going to be correct, so this will be our control. On the other hand, if we increase the speed in which case the pulses will be of shorter duration and they will arrive faster. So, we will see that most of them are being still being sensed while suddenly one pulse is missed. So, you see that the program was scanned for this pulse the program when the program scanned the value was 0 and when the program scanned, next the value was again 0 and so the program assume that that it remain 0.

So, it went up to 1 in between and came down is not will be the program will be unaware of that and therefore there will be perhaps small delay in the computation of the speed. If you have faster pulses coming out still faster, then many such pulses will be based for example, this one. So, this will be more of a chance to see when for example, this pulse will be counted, this pulse will be counted. So, we will get you now you will get drastic for example, in this pulse actually what will be inferred when it was sensed here, it was 1, when it is sensed here it is also 1, so therefore it will be assumed that it continued at 1. So, the inferred pulse is going to be in this case the pulse count will be severely wrong.

So, it we assume that it has continued here and then it is continuing here and then it is continuing here and then it is continuing at this point. So, the count will be small fraction of the real speed and we are going to be totally out it is actually for this reason that for counting fast pulses you cannot use the PLC itself, but rather use a special curve. So, that curve does not have the purpose of that curve is solely to count the pulses coming from a such a fast shaft angle encoder it is no it is not loaded with any other task. So, therefore it can really sample that line very fast and therefore can compute the speed and the speed can be made available to the PLC in terms of a value.

So, rather than the PLC counting the pulses somebody else will count the pulse and will convert it to a value which the PLC will read from time to time. So, it is for such reasons that sometimes you now special functional modules like as I as we mention in the earlier class that is some sometime special you now high speed counter cards are needed it is for this purpose, so we go move on.

(Refer Slide Time: 27:20)


Indian Institute of Technology, Kharagpur
Relays and Contactors
a network (series parallel)
+ve
-ve
Power busbars
Solenoid
Lamp
Motor coil
Relay logic
A logical function Realized with relays

We look at the predecessor of a PLC actually you as we have said that PLCs just before the PLCs where invented or made out of microprocessors such sequence control problems use to typically tackle using control panels, which typically employed. You now think like relays contactors various kinds of switches and lamps various kinds of you now electromechanical timers and such things. So, they were actually physical devices which were hardware, so they were typically arranged for you, now it is of maintenance and an installation etcetera they were typically arrange modularly.

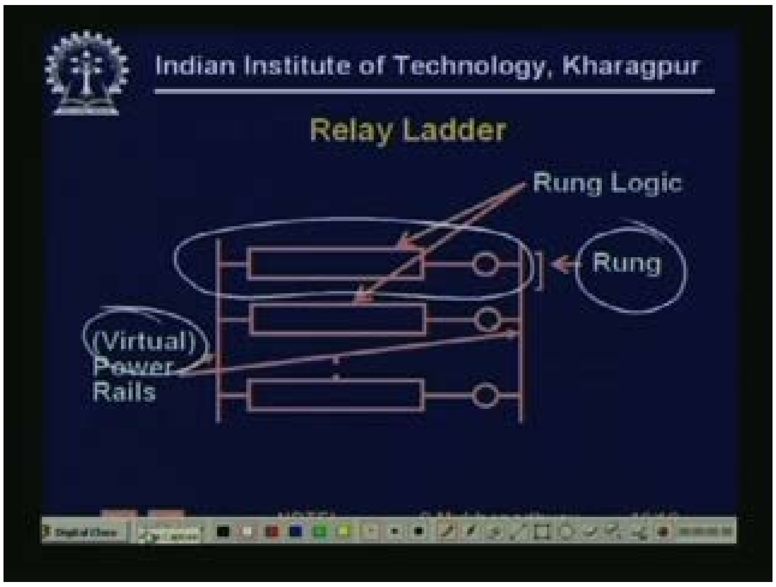
So, the typical arrangement would be that there will be a positive voltage busbar and there will be a negative voltage busbar and you will implement a particular logic. That is here you will implement basically a network which is a series or parallel network series or parallel of various kinds of switches. So, under only certain specific conditions of this switches this there will be a connection from this point to this point, otherwise under other conditions these two ends will not be connected. So, when they are not connected the voltage will not appear here and therefore, current will not flow.

So, whatever this output is this is sometimes is to be called an output coil because the where typically physical devices like solenoids or motor starter coils, so they have a typically coils. So, they are sometimes called output coils, it is a legacy, there they are still sometimes called output coils.

So, then the current will actually flow through this, so it is therefore by this switch network you can actually control you have control the situations under which this output is going to be excited. So, this is the wave things where made and when PLCs where made initially see you can always make a new device, but it is relatively more difficult to change the training and the mindset of the people who use them. So, just you now respecting the background of the practicing engineers the PLC programs the PLCs are nothing but microprocessor based system, so if you see essentially PLC programs are nothing but assembly language programs.

However, just so that people can write them and people can interpret them better without making mistakes. So, therefore, a graphical kind of programming language was evolved and it was used to so that the engineers could think in terms of relays and then using some tool such pictures such relay connection pictures could be transformed to an assembly language. So, that is why programmers have to be used to convert such graphical programming language which resemble relay a ladder logic physical relay ladder logic. They actually get kind of you now compile into an into an assembly language program or a machine language program and they none on the PLC.

(Refer Slide Time: 31:06)

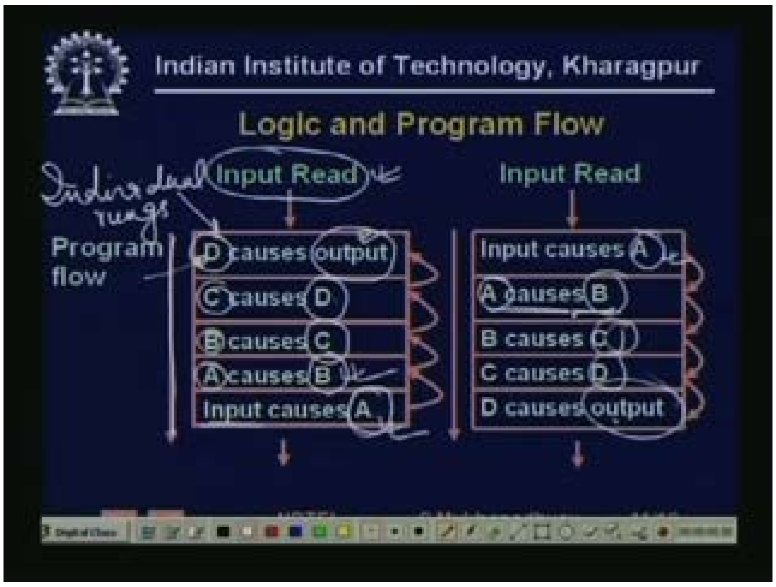

Indian Institute of Technology, Kharagpur

Relay Ladder

So, the kinds of PLC programs that will see they are also be organized like relay ladders while there is not real relay here.

So, this will draw this power rails which are virtual because this is entirely and obstruction and each one of those each one of these programs statements will be called a rung. So, actually these relay ladder logic programs and nothing but a series of rungs having between two rails which are virtual. So, the left part of the rung is a network of the various ladder logic elements like various kinds of contact timer’s etcetera. This is followed by an output coil which shows under what condition that output is going to be excited this is the way we are going to draw them, so coming back now we have to understand that.

(Refer Slide Time: 0:32:14)



So, you see that how the PLC program will be will be executed, so one after the other starting from the top after the inputs are read program execution is simply executing the logic of the rungs evaluating the output values from the top one after the other. So, typical program flow is from that from the top these are individual rungs individual rungs, now you have several logics and there all executed between one input read and output write output read. So, it may appear to you that in what order you are going to put these logics one after the other is actually immaterial makes no difference, but it does for example, imagine that you are. Now, sometimes what happens is that when there is a very complex logic, then you sometimes break of that logic in to several parts.

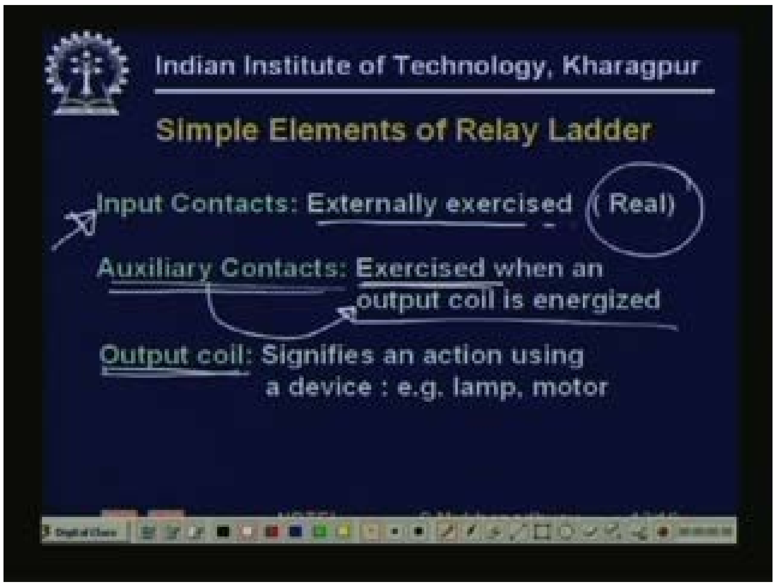
So, what happens is that initially you evaluate one some small part of the logic you compute some value. Then, you use that value and you put it in another expression and then evaluate something else, so it may happen that finally, we want to compute the final output. The final output can be computed based on some intermediate result D which can be computed in terms of some intermediate results C which can be computed in terms of B which can be computed in terms of A. Finally, the input, finally A can be computed in terms of the input, so you are essentially broken up a logic into these five different steps. Now, if we write it like this as we have shown that what will happen is that first, suppose the input is read which will cause a which is suppose to a cause a change in the output.

Now, now note that these are actually nothing but internal variables these DCBA these four are sub just internal variables there, they have no there are there are just you now memory variable. So, when you will evaluate output you will use the old value of D which was computed using the old input. So, therefore, output will not change in this cycle, however similarly C will not change because it is because and D also will not be will be updated because we will use the old value of C. Similarly, C will not be updated similarly, B will not be updated, however, A will be updated because now you got a new value of input in the next cycle.

This changed value of A will cause a change in B while C and D will still remains C D and output will still remain unchanged. So, in this way just because we have organized these programs in this fashion we will get a five scan cycle delays between an input read and an output change. This is completely unnecessary because if we had organized it in in a in a slightly different way then what would a happen suppose we just invert the order, then in the first cycle input will change A. So, A will get change in the first cycle, now when we evaluate the second one, we contained the new value of A, because we are taking it from the memory and the memory has been updated.

So, similarly, when B is updated C will be updated when C is updated D will updated and so in the first cycle itself output is going to be updated. So, this is something to be remembered that the program flow should represent the actual flow of logic that is what causes the other when one is developing a read a ladder programs, so now let us look at some of the elements which occur in a relay ladder program.

(Refer Slide Time: 36:25)



Indian Institute of Technology, Kharagpur

Simple Elements of Relay Ladder

Input Contacts: Externally exercised (Real)

Auxiliary Contacts: Exercised when an output coil is energized

Output coil: Signifies an action using a device : e.g. lamp, motor

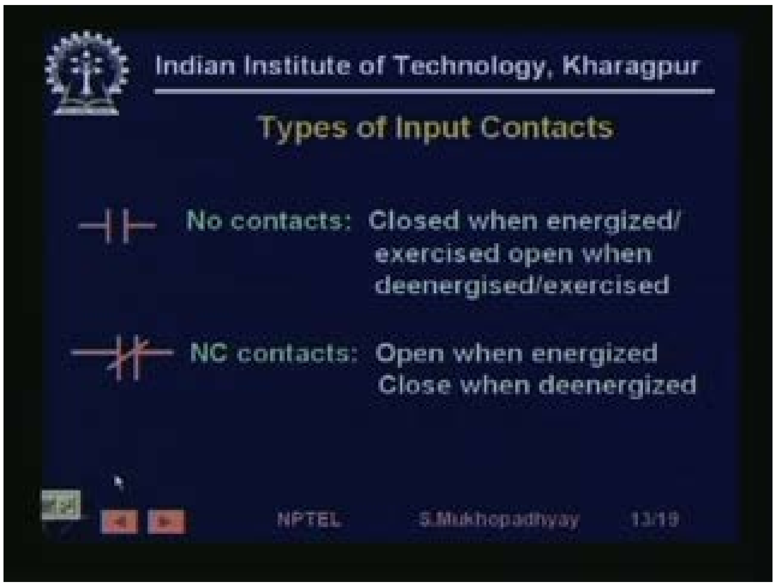
So, the simple elements of a relay ladder today we will look at there are various kinds of elements today we will look at three kinds of elements namely will look as we have already seen that there is a that there is an output coil at the end. We have also seen that it is like to rails then some logic and then finally, an output coil. So, this logic today we will study where the logics are just meet of some input and output contacts. In fact, they can be made of many things they can be actually this input and output contacts are those you have studied in detail electronics there are like combinational circuits. We could also put other elements in this within this logic to make this logic a sequential logic using timers and counters, then thinks like that.

So, that we will see on the in the in the next lesson today we are going to C circuits which are made simply of various kinds of switches. So, basically combination logic, so before we change this we must mention that there are now this switches are of two times some of them are called input contacts. So, they correspond to this contacts are physical you now they are obstructions of real physical inputs like some photo detector has detected a part. So, it has changed from 0, it has gone to 1, so that one of that, so an input contact will correspond to the physical photo detector device. Similarly, there may be a limit switch or there may be a pressure switch, so these are or there may be a push button which the user pushes, physically they may be operator pushes.

So, such contacts will be called input contact which are externally exercised by the machine; that means, external to the PLC and which are real that is there are corresponding to theses contacts there are real physical devices. On the other hands sometimes we will use the memory variables as contacts for example, as you have said that we can have we can use some value of the output coil suppose the output coil value of the I th rung we may use in the j th rung. So, how are we going to use that, so to use that value corresponding to the to an output coil, we will create a contact such contacts are called auxiliary contacts.

So, there they have no physical existent they are simply just memories and there used actually for logic evaluation and there exercised when an output coil is an energized. So, these auxiliary contacts correspond to output coils and finally, an output coil is also corresponds to a physical output in the real external world. So, these are the three elements which we first will construct our simplest PLC programs, there are we will we will use to two types of contacts.

(Refer Slide Time: 40:11)



Indian Institute of Technology, Kharagpur

Types of Input Contacts

No contacts: Closed when energized/ exercised open when deenergised/exercised

NC contacts: Open when energized Close when deenergized

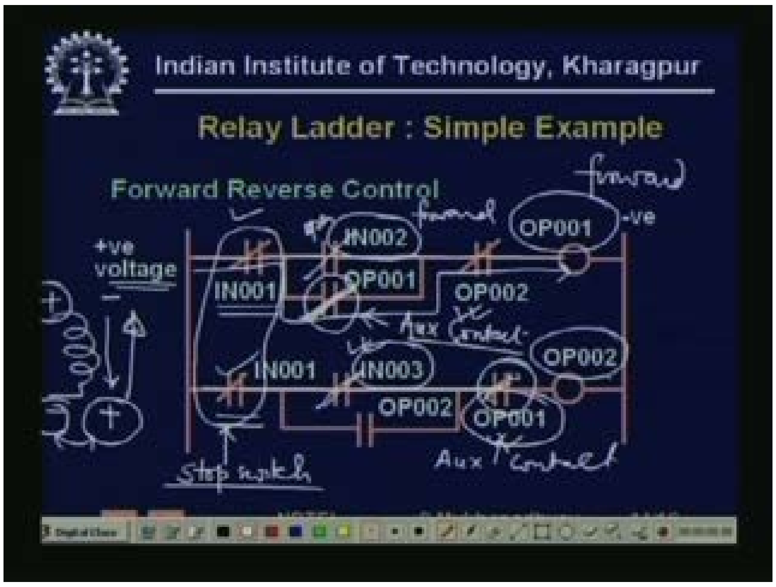
NPTEL S.Mukhopadhyay 13/19

The first kinds of contacts are called not no contacts they are NO contacts, NO means normally open similarly. So, this means that when the this contacts are not energized or there in the you know there are unexcited or D energized states then this contacts should be assume should be assumed to open because we always interpret PLC logic as if some switch is going to open.

So, we are always looking for closed parts on the RLL program, so when the when that contact will for example, when a when a push button is not pressed if it is represented by an NO contact. Then, when it is not pressed that contact in the RLL ladder logic is going to remain open. So, we must interpreted that similarly, we might have what is known as an NC contact where NC stands for normally closed. So, the same push button if we represented by an NC contact then if the push button is not pressed that is when it is not excited that contact will remain closed.

So, we must interpreted that way in the PLC logic when we try to see whether there is a continuous parts from the positive rail to the output coil positive N. So, it is open when energized and it is closed when de energized, now we will look at a very simple example.

(Refer Slide Time: 42:05)



This is an example which is it is simple version of an example which is used typically for let us say motor control. So, imagine that we have a motor and we want to there are there are two push buttons one says go forward. So, the motor will rotate in one way it could be a movement of a motor it could be a movement of a plunger in the forward direction. Similarly, there is another push button which says go in the reverse direction. So, how is this achieved this typically achieves by if current flows in, so may be maybe there is some solenoid somewhere. So, if this is positive this made positive this is negative current will flow in this way, then the motion will take place in one direction.

On the other hand, if this is made positive and this is made negative then current will flow in this way and then probably them motion will taking the other direction, this is the way it done. So, therefore sometime we have to connect the positive terminal to this point and sometimes you have to connect the positive terminal to this point now a potentially hazardous situation exists that is if by chance they are two different switches. If there are pressed together then what is going to happen is that the positive terminal and the negative terminal will get shorted this may cause an accident. So, we want to have a logic we do not feed the push buttons directly rather then we take it through a PLC.

So, that we ensure that even if they are pressed together no such problem will occur, so now we see what is going to happen. So, you see that these two push buttons corresponding to each push button we have two contacts. So, corresponding to the fast push button which we called IN 0 0 1, we have two corresponding to IN 0 0 1. Now, you see sorry this IN 0 0 1 is actually a master control C. So, nothing will happen see the motor will not rotate either this way or that way if unless this switch is if this switch is pressed

So, this is like an emergency stop you now this motor has three modes, so it has a mode move forward it has a mode move reversed and it has a mode stop. So, if this is actually a stop switch, so you see that and these are the two output coils which you can say symbolizes the arrangement of switching the power supply to either the positive and the negative. So, when this goes becomes one when OP is 0 0 1 becomes 1, then the motor get supply in one way and when OP 0 0 2 becomes 1 the motor get supply in the other way and when both are zero it does not get any supplies, so it is standing.

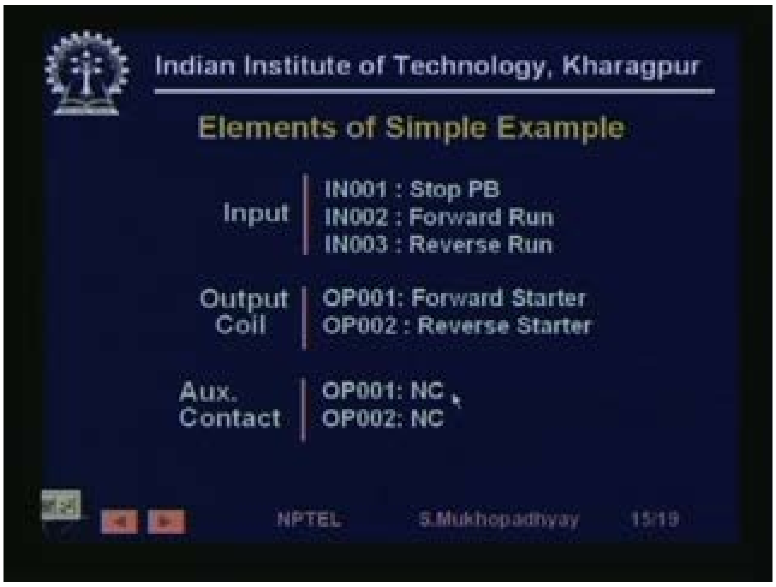
So, we see first thing we see that these are normally closed switches and when IN 0 0 1 is excited or the stop switch is pressed then this will become open because they are normally closed. So, therefore obviously there is no question of this positive rail there is getting at continuous path to the output coil. So, both will be off fine, so we have we have satisfied that condition there is that is the stop switch is pressed it is going to be 1. Next suppose the stop switch is not press, so it is off now if we pressed, suppose this is the push button this let us say forward. So, if this is pressed now we see initially both are off. So, therefore, this is also on and this is also on. So, when this is pressed this will become on and there will be a continuous path to the output coil.

So, this is the forward coil and the motor will start moving forward now note that this an auxiliary contact. So, the moment this gets supply or becomes one this is going to be excited. So, this will be closed even if you remove this switch it is a push button, so you have to you have to you have to press it and then you can release it you cannot keep holding it. So, you press it once and immediately even if you leave it the motor will keep running this that that is the arrangement that we have to make. So, that is made by this parallel path, so after this has become on even if this becomes again this becomes open there is still a parallel path and that parallel path is this one.

So, this forward coil continues to get supply and the motor continues to rotate now in this position while the forward coil is on suppose somebody presses the backward coil or the reverse coil what is going to happen. So, this will be come on, but interestingly nothing will happen that this coil will actually not get supply because of this one which is another auxiliary contact whenever this is on, this becomes excited and this become open. So, this will be open which means that even if this this is all ready open because this is off. So, even if this is becomes on that is no the continues path stops here because this is open.

So, therefore this cannot be excited while this is on if you have to excite it you have to first press the stop switch by which both will become off exit this time. You can press this one, then this will be come on and similarly, when this is on you cannot press you cannot make this one excited. So, this is the standard forward reverse interlock which ensures that simultaneously you cannot command the motor to move both ways to before the end, let us look at another example.

(Refer Slide Time: 48:53)



Indian Institute of Technology, Kharagpur

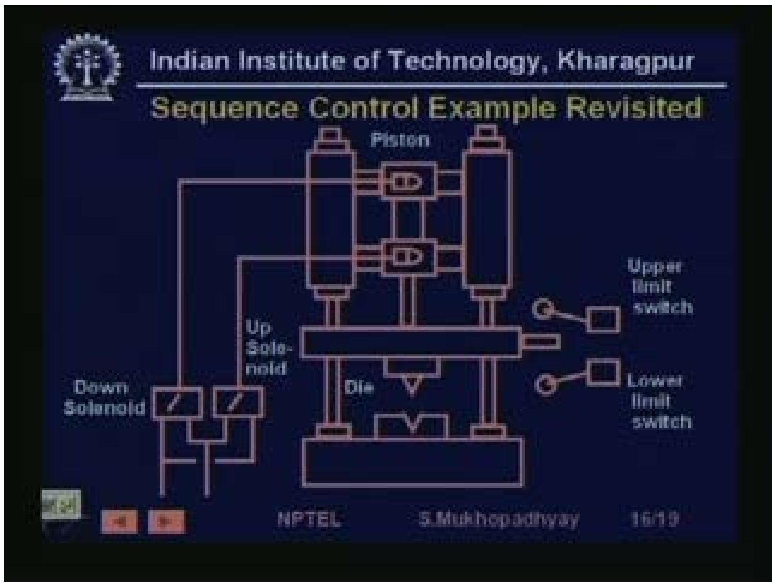
Elements of Simple Example

Input	IN001 : Stop PB IN002 : Forward Run IN003 : Reverse Run
Output Coil	OP001: Forward Starter OP002 : Reverse Starter
Aux. Contact	OP001: NC OP002: NC

NPTEL S.Mukhopadhyay 15/19

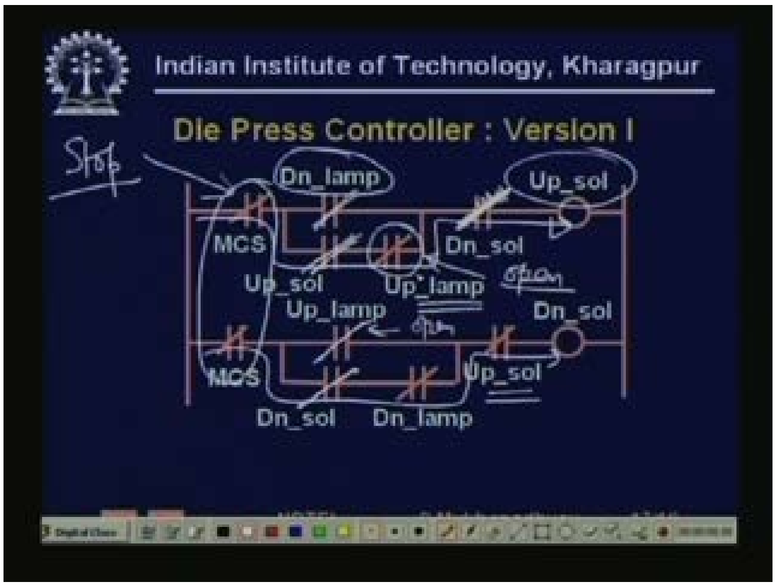
So, what are the elements, so elements of the example are the IN 0 0 1 which is a stop push button IN 0 0 2, which is a forward run push button IN 0 0 3, which is a reverse run push button. The output coils are forward starter and reverse starter and the auxiliary there are auxiliary contacts NC and NO also corresponding to these. So, there were NC there is a one NC and there is a 1, NO I am sorry anyway that we will check.

(Refer Slide Time: 49:23)



So, now let us a look at the second example this is our old you now die press, then we now it is behavior we have seen it in the in the earlier lesson. So, now we have to design a controller for it a control logic for it such that whenever a die is that is whenever some switch is master control switch is played pressed, it keeps on going through a cycle, so we will see it is simplest version.

(Refer Slide Time: 50:02)



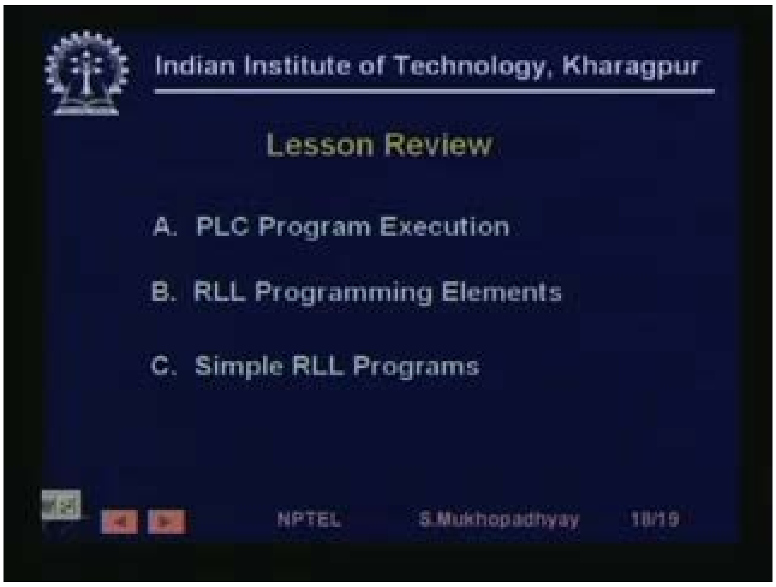
So, you know it is very similar to the forward reverse control because here also there is a forward reverse motion only thing is that, so you have master control switch again. So, if the master control switch is if the master control switch is if it is master control stop which kind of emergency switch. So, if it is flicked then this machine does not work, so neither up solenoid not down solenoid will get supply that is fine. Now, initially suppose the press is in the bottom position, so the down lamp that may be that may be the parking position of the shutdown configuration. We are not discussing how it will come to the shut down configuration we are just saying that from the shut down configuration if it starts I mean how does it start.

So, because the down lamp will be made, so this will become on the moment and so initially no initially down lamp is there. So, what will happen, it is down and because the down solenoid is this because the down solenoid is because the down solenoid is off. So, therefore, this actually is we should have this as an NO contact, so then this will become on it will become no it will become no all right, so it is an NC contact, this is this down solenoid is off, so therefore it is closed.

So, when the down lamp is close immediately the up solenoid will get immediately the up solenoid will get supply. So, when up solenoid get supply again this will be this is an auxiliary contact, so it becomes it becomes on. Now, there is a path through this way note that as it moves, so immediately when the up solenoid is on the press will start going up and when it is goes up the down limit switch and the down lamp will glow will open. So, even if this opens, but up solenoid will be on and it keep going up now the point is that it must stop somewhere. So, when it will reach the upper most position at that time the up lamp will glow when this will be glowing this will become open.

So, then the up solenoid will become, so now, the up solenoid become 0, so therefore, this is closed on the up lamp has glow. So, this is now closed, so immediately the down solenoid will now get supply and when the down solenoid get supply this is closed and the down lamp is anyway not excited. So, therefore it will start coming down and then the then the then the up lamp will open, but still there is a path in this way and the down solenoid will keep getting supply. So, the press will keep coming down under hydraulic pressure, so you see that using another contact we have we have just ensure that there is that there is alternative motion. So, these are the two examples of RLL that can be constructed with simple contacts to review the lesson, we have seen three major topics.

(Refer Slide Time: 53:38)


Indian Institute of Technology, Kharagpur

Lesson Review

A. PLC Program Execution

B. RLL Programming Elements

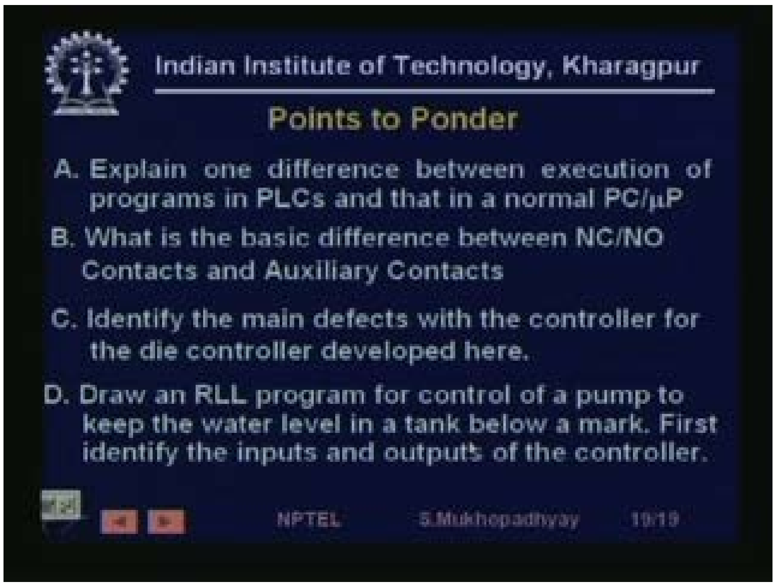
C. Simple RLL Programs

NPTEL S.Mukhopadhyay 18/19

One is a PLC program execution the second is seen the simple RLL programming elements of NC and NO contacts and input and auxiliary contacts as well as output coils.

Finally, we have seen two simple RLL programs which have shown that how do create interlocks and how do create alternating motion.

(Refer Slide Time: 54:02)



Indian Institute of Technology, Kharagpur

Points to Ponder

A. Explain one difference between execution of programs in PLCs and that in a normal PC/μP

B. What is the basic difference between NC/NO Contacts and Auxiliary Contacts

C. Identify the main defects with the controller for the die controller developed here.

D. Draw an RLL program for control of a pump to keep the water level in a tank below a mark. First identify the inputs and outputs of the controller.

NPTEL S.Mukhopadhyay 19/19

There are some points which you could test questions are points to ponder, for example what the difference between the executions of programs in PLCs and that we could write in a normal PC or a microprocessor. We have seen that the differences in the way in the input output is done, what is the basic difference between normally closed and normally open contacts and input and auxiliary contacts. It will be interesting to see that what could be possible defects we are the die press controller that we have given is a very simple controller what could be possible defects with it. Finally, it will be good if you can try and try to draw your own RLL program for the control of a pump to keep the water level in a tank below a mark.

So, imagine that in the house you have put some PLC control such that you never run out of water and it will sense the water level. So, a very important part of part before designing the RLL is to identify the inputs and the outputs of the controller what sensor you have going to use what are going to be outputs are PLC a etcetera then try to write the RLL program. So, that is all for today thank you very much will see for see the next lesson in the next class.

Thank you very much.

(Refer Slide Time: 55:36)

Preview of next lecture

(Refer Slide Time: 55:41)

Lecture - 20

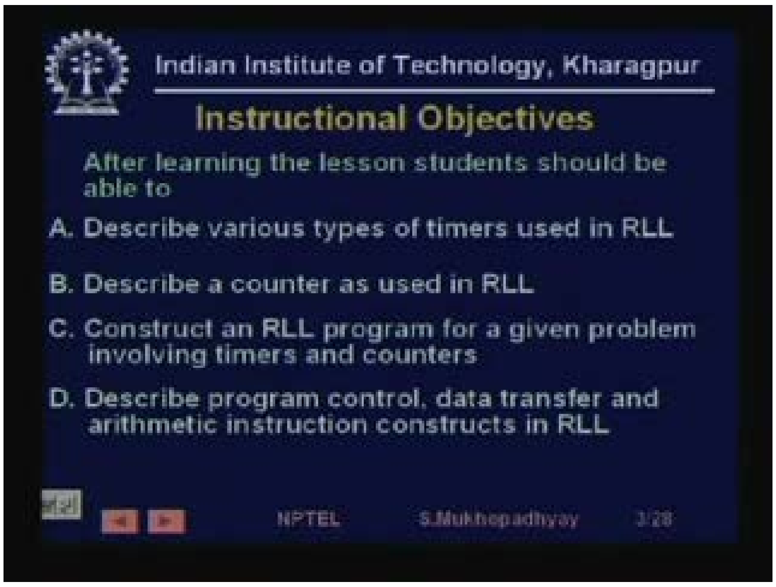
Sequence Control

More RLL Elements

RLL Syntax

Welcome to today’s lessons, which is lessons number twenty of the course on industrial automation and control. 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 there 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: 56:22)



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

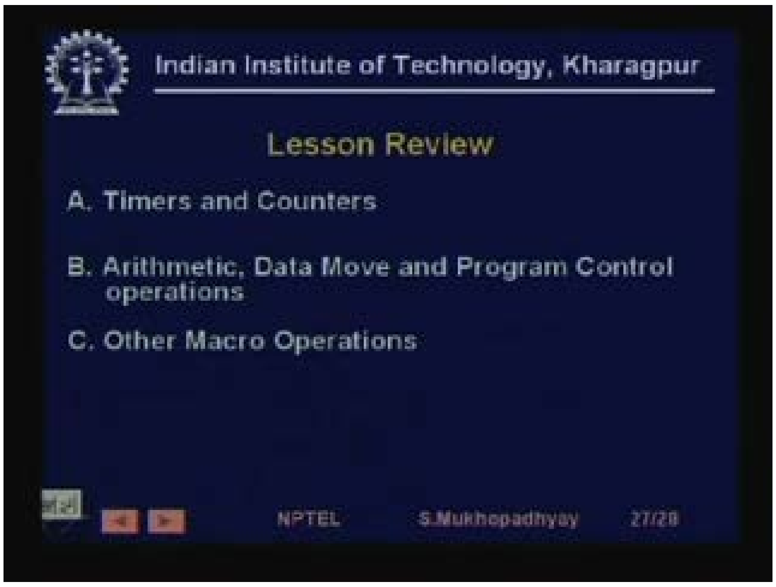
C. Construct an RLL program for a given problem involving timers and counters

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

NPTEL S.Mukhopadhyay 3/28

So, a student will be familiarized 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 a 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 there are 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 sometimes necessary, so we will get familiarized with some of them.

(Refer Slide Time: 57:17)



Indian Institute of Technology, Kharagpur

Lesson Review

A. Timers and Counters

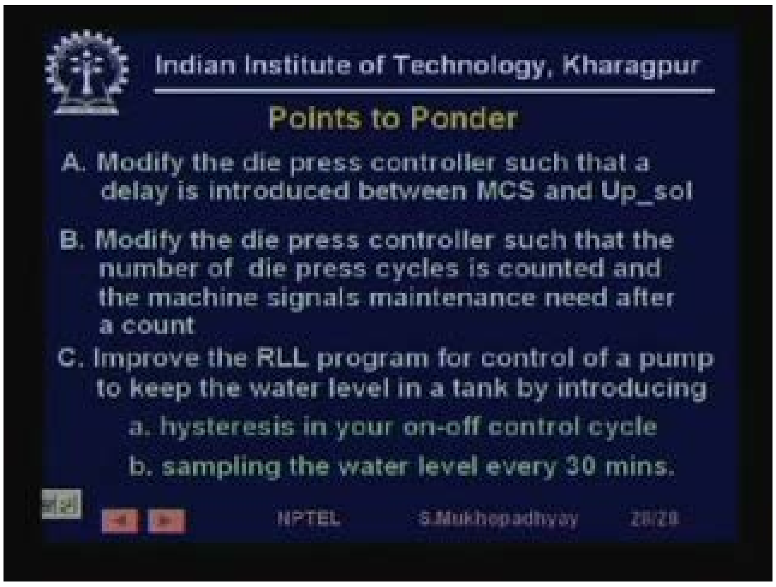
B. Arithmetic, Data Move and Program Control operations

C. Other Macro Operations

NPTEL S.Mukhopadhyay 27/28

So, we have come to the end of the lesson and in this lesson 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 coming to the end we have the usual points to ponder.

(Refer Slide Time: 57:59)



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

For example you could try to modify the die press controller such that a delay is introduced 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 modified the die press controller such that the number of die press cycles is actually counted. So, you say that after every 1000 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.

After it reached, it reaches a count you want to you want to stop the machine for maintaining. Similarly, you could improve 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 thirty minutes, so that is all for today.

Thank you very much.