INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

NPTEL
ONLINE CERTIFICATION COURSE

On Industrial Automation and Control

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

Topic Lecture – 28
A Structured Design Approach To Sequence Control

So welcome to lesson 21 of the course Industrial Automation and Control.

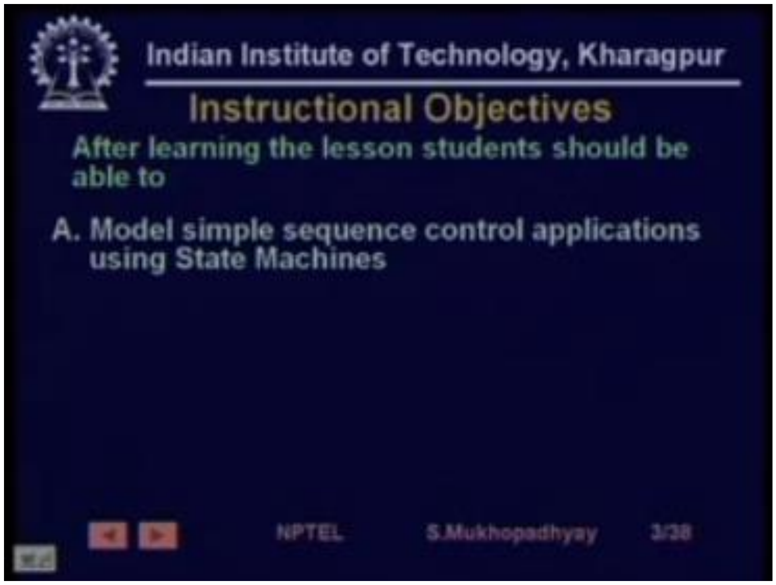
(Refer Slide Time: 00:26)

(Refer Slide Time: 00:28)

In this lesson we are going to learn a structure design approach to sequence control so far we have mainly seen the programming constructs have seen small, small program segments timers, counters. In this lesson for the first time we will see that given a practical problem how to study the problem, how to, what are the steps that you go through to finally arrive at an RLL program, So and this will be followed using a very systematic approach because as I have already told you that industrial control applications are very critical in the sense that they if you have programming errors in them they can be very expensive in terms of money or in terms of even can cost human lives extra.

So it is always good to have a very systematic design process by which you can decompose a problem and then finally arrive at a solution.

(Refer Slide Time: 01:25)



So we will look at the instructional objectives, the instruction objectives of this lesson are firstly to be able to model simple sequence control applications using state machines, state machine is actually a formal method and we advocate the use of 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 contain contradictions and are also easy to understand and develop. Then from these formal models we have to develop.

(Refer Slide Time: 02:00)

RLL programs for such applications, and for doing this there are certain apart from the RLL programs there are some modern programming constructs which are being made available one of them is the 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: 02:24)

So now let us go through the steps in basic broad steps in sequence control design. So first step is to study the step behavior this is a very critical step and most of the errors that happen in any programming exercise not only this kind of industrial automation programming any programming mainly arises from the fact that the programmer or the developer did not understand the system well, so this is a very important step and.

(Refer Slide Time: 02:52)

One must first of all identify inputs to this to the system that is the programmable controller program what inputs it will take, inputs can come from either from sensors in the field or it comes from operator interface which I call the MMI or the man-machine interface so somebody presses a pushbutton that is an operator input, right. On the other hand some limits which is made that is a sensor input.

(Refer Slide Time: 03:21)

Similarly identify the outputs, so switch on motor, so motor is an actuator that is a kind of output there is another kind of output for example, switch on some indicator or some lamb that would be an output which again goes to the MMI or the man-machine interface. So we have to first identify these.

(Refer Slide Time: 03:41)

Then study the sequence of actions and events under the various operational modes this is the main task you have to very carefully understand what is going to happen and what will happen after what, at what time intervals extra.

(Refer Slide Time: 04:00)

Then one thing that must be very clearly remembered is that when you are developing and an industrial automation program one not only has to remember, not only has to design for normal behavior but one must to some extent at least take into account the possible failures that can occur, otherwise a system that behaves well under normal behavior can behave in a very nasty manner if some simple element of the system like a sensor fields, right.

(Refer Slide Time: 04:42)

Then even apart from the automated behavior one has to examine the requirements that exist for number one manual control, manual control is very important because for finally if the automation equipment fails it should be able to operate the system using manual control, right maybe right on the field while the automated control may be actually working quite a distance away from the actual equipment it may be housed in some control room.

On the other hand the manual controls may be near the equipment at the field so the possibility of including such manual controls must be examined.

(Refer Slide Time: 05:23)

Whether some additional sensors are required some sensors may be there but to achieve a kind of functionality some other sensors may be needed.

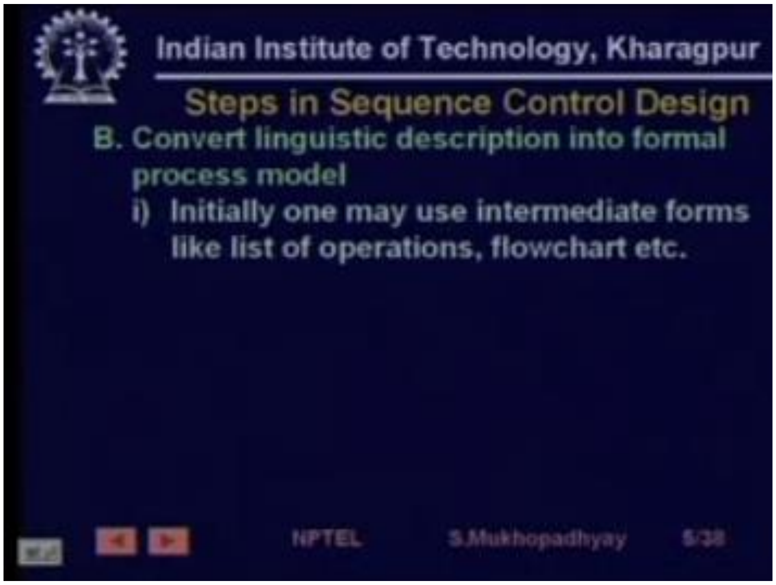
(Refer Slide Time: 05:31)

Indicators alarms and as well as operational efficiency or safety these are the factors which must be considered to finally arrive at the functionality, one must always remember that the customer may not always be able to express his or her needs and a good automation engineer should be able to supplement it with his own experience in such cases, so having done that the next step is to convert.

(Refer Slide Time: 06:04)

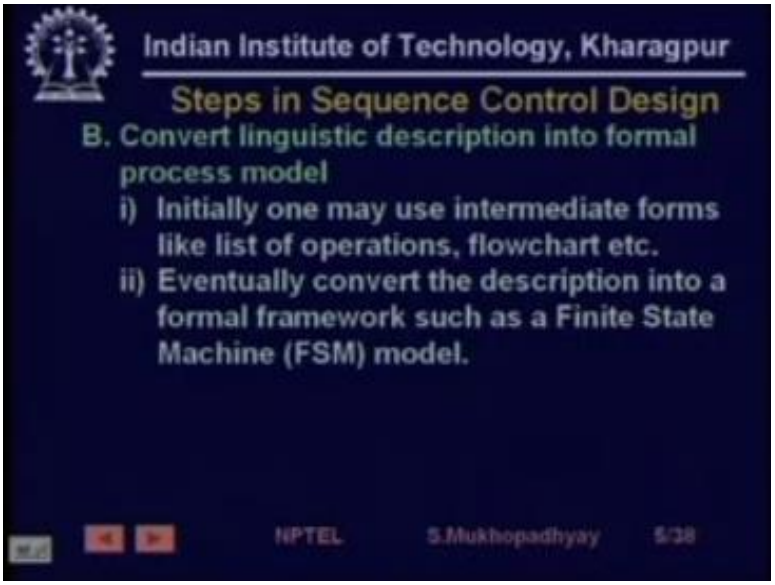
Generally these things are captured manually and using a linguistic description something like you know something like English statement so you talk to customer and talk to engineers on the field and get their requirements. But this is very dangerous to use for program development so we have to convert this linguists description into formal process models.

(Refer Slide Time: 06:27)



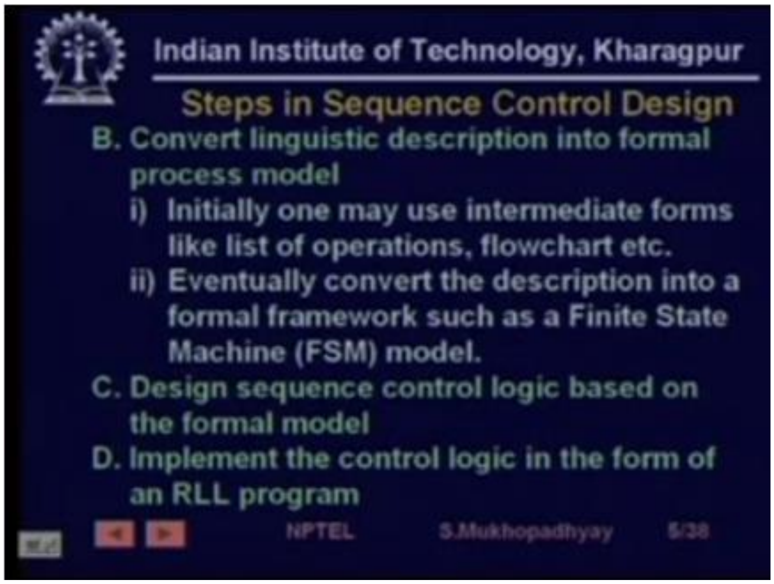
And in fact a lot of you know inconsistencies which are their ambiguities which are there in the linguistic description actually service at this time. Even for the during the process of transforming it into a formal process model one may initially use intermediate forms like you know like flow charts for example, okay.

(Refer Slide Time: 06:45)



Then finally, but finally it is prescribed that one should be able to convert it to a formal mathematical framework something like let us say a finite state machine which we will be using here, after up to this the operations are manual having done that then one has to go for design of the sequence control logic based on the formal model.

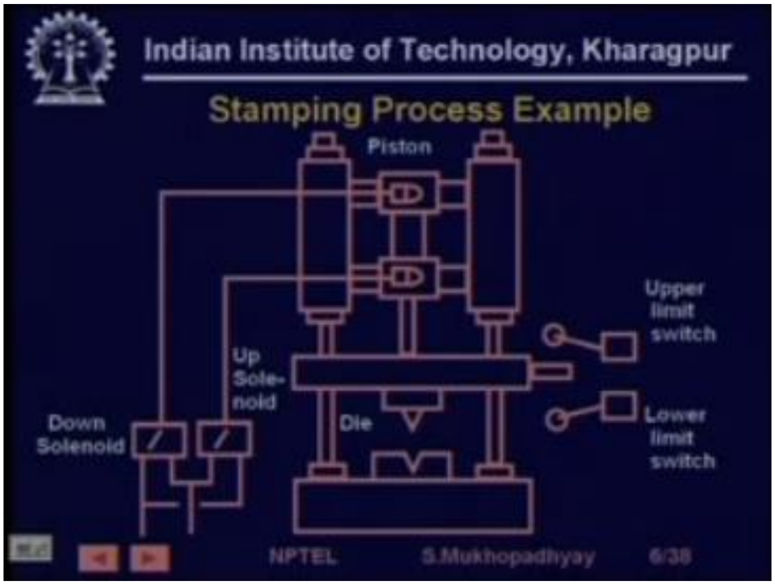
(Refer Slide Time: 07:11)



And then finally one has to implement the control logic in the form of an RLL program and it is preferable that these steps especially the step D is made as much as possible automatic, because this is a step which can be done in an automated manner once B and C have been carried out and for large programs it is always preferable to go for automatic programming because that will always lead to error-free programs provided your specifications were correct.

So we come back to our old stamping process example which we have seen in earlier lectures.

(Refer Slide Time: 07:50)



So here is a stamping process we know this process so we will we have made some addition to its functionality to be able to explain you know certain features of a system and make it more complete. So basic principle is the same that there is a piston which and there are two solenoids hydraulically driven piston which goes up and down and makes stamp things, okay. So if we now try to write its list of actions try to create a linguistic description of the process operation it will look something like this.

(Refer Slide Time: 08:31)

So in step A it says that if the auto push button is pressed so that is an operator input it turns powers and lights on so there is possibly a switch once one or more switches with we will consider it to be one, which will turn the power and the light on I mean the moment the auto button is pressed.

(Refer Slide Time: 08:53)

Where a part is detected the thing to be stamped when it is detected which one is placed at the proper place and detected, so there has to be a part sensor. The press ram the thing that will the heavy piece which will move and make a stamping it advances down.

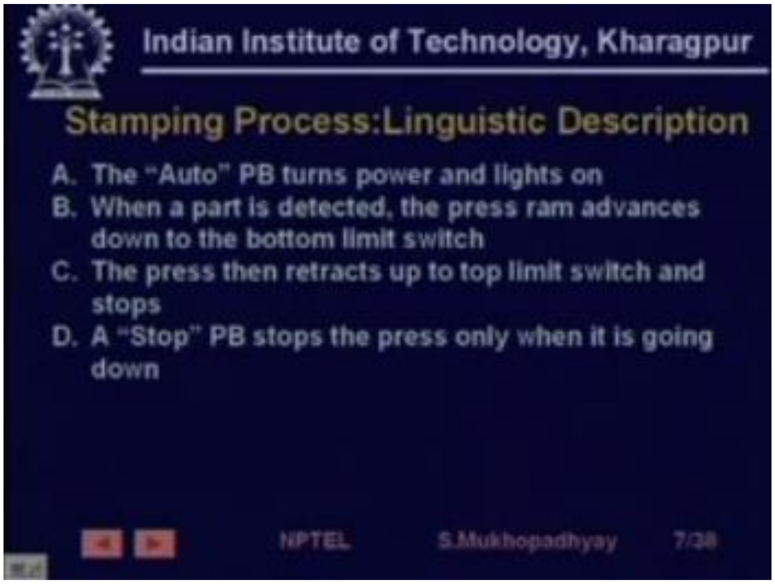
(Refer Slide Time: 09:12)

And it will stop once it makes the bottom limit switch so that is again a sensor and you must have actuators to make the RAM move down.

(Refer Slide Time: 09:25)

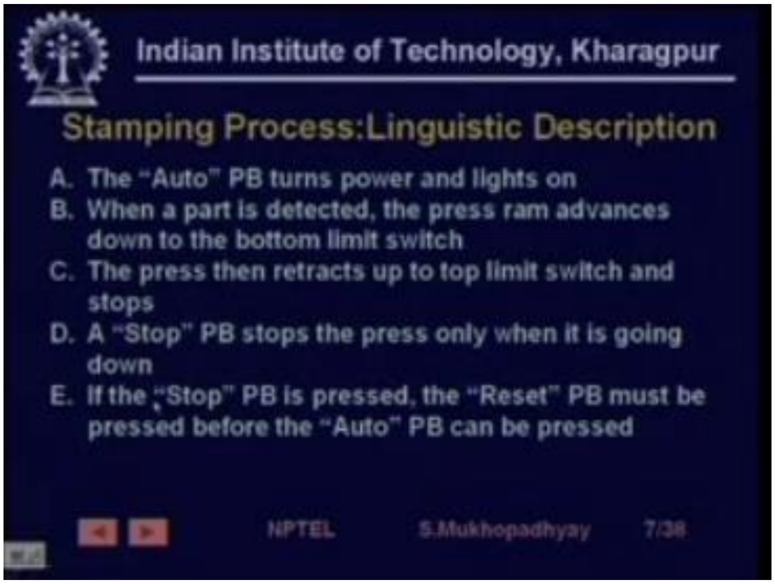
Then the press then retracts up to the top limit switch and stops, so it makes the stamping and stops all right.

(Refer Slide Time: 09:36)



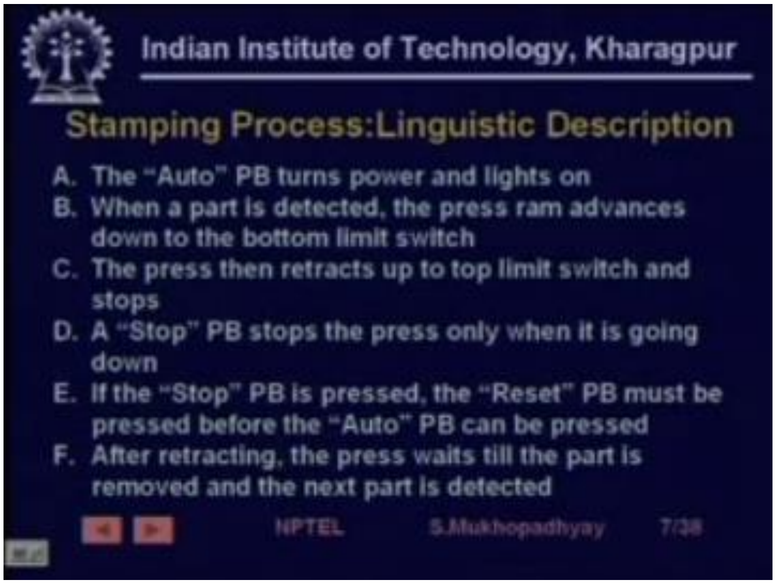
On the other hand due to some reason the operator may be may like to about a stamping operation, so there is a stop push button provided to the operator and a stop push button stops the press only where it is going down.

(Refer Slide Time: 09:51)



When it is going up it has no effect, because anyway that is not going to cause any problem. If the stop push button has been pressed it means that something abnormal might have happened so the reset push button must be pressed before the auto push button can be pressed for the next cycle of operation, so once you have press stop you have to press reset you know it is a kind of acknowledgement that the emergency has gone away and the automated operation can resume.

(Refer Slide Time: 10:22)



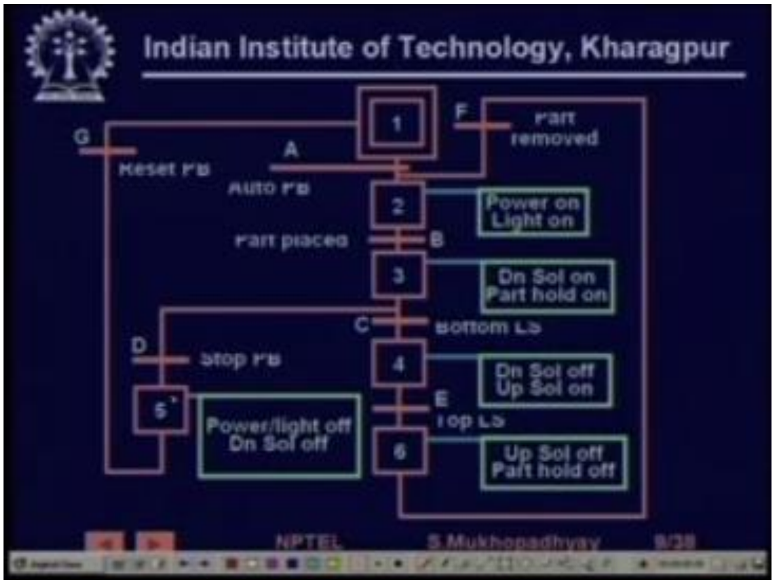
Finally after retracting and then going up the press waits till the part is removed and the next part is detected, so till the part will be removed and then after that when the next part will be detected again the RAM will start coming down. So this is the English behavior of the system, okay. So now let us try to convert it to an unambiguous mathematical description, so first step as I said is to get the process inputs and outputs.

(Refer Slide Time: 10:55)

So what are the process, so here are the process inputs and outputs, so as inputs we have part sensors which gives two kinds of we generates two kinds of events, one is part placed another this part removed then there are three kinds of push-button the auto push button, the stop push button and the reset push button these three are operator inputs, then there are the two limit switch sensors bottom limit switch and top limit switch.

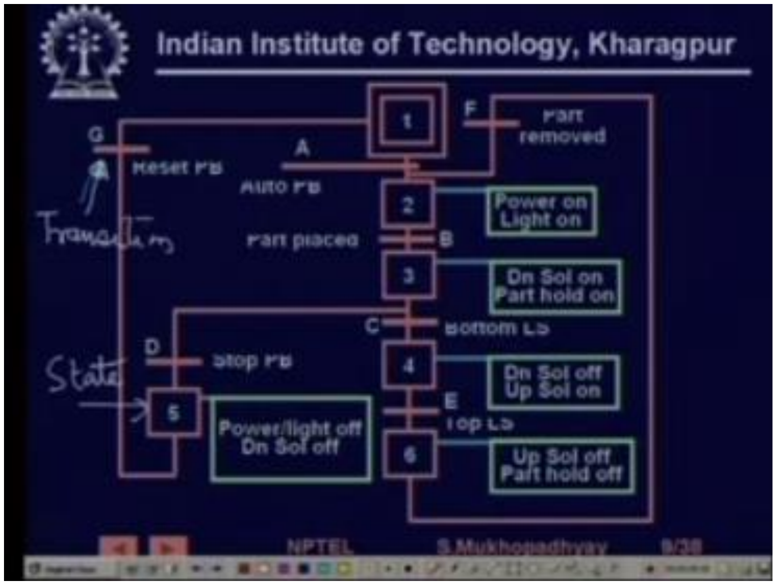
For outputs we have four outputs we have an up solenoid which moves the RAM up we have a down solenoid which moves it down, we have the power light switch and we have a part hold which holds the parts while it is being stamped, so these are our outputs. Now we develop a state machine so let us try to interpret this diagram.

(Refer Slide Time: 11:53)



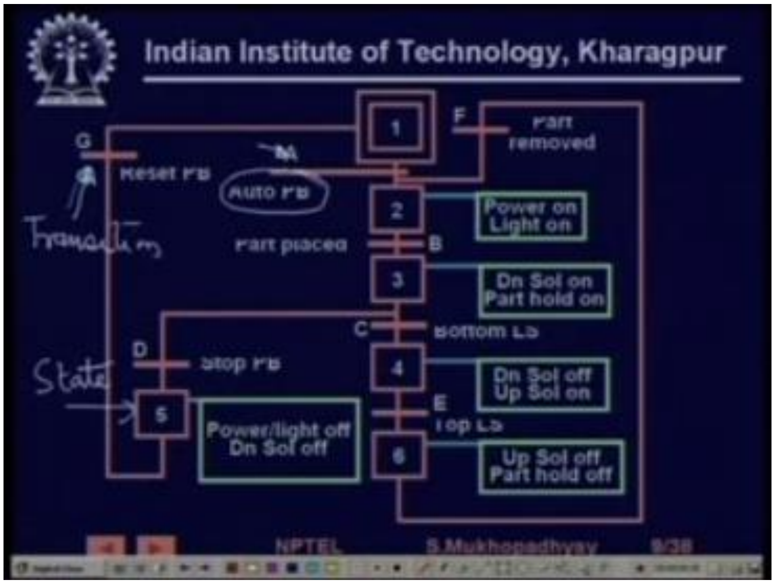
So what is happening in this diagram is that let me select my pen, so what is happening is that you see these squares are the states, we are possibly familiar with state machine so a state machine is like a graph which consists of a set of states these squares are the states.

(Refer Slide Time: 12:22)



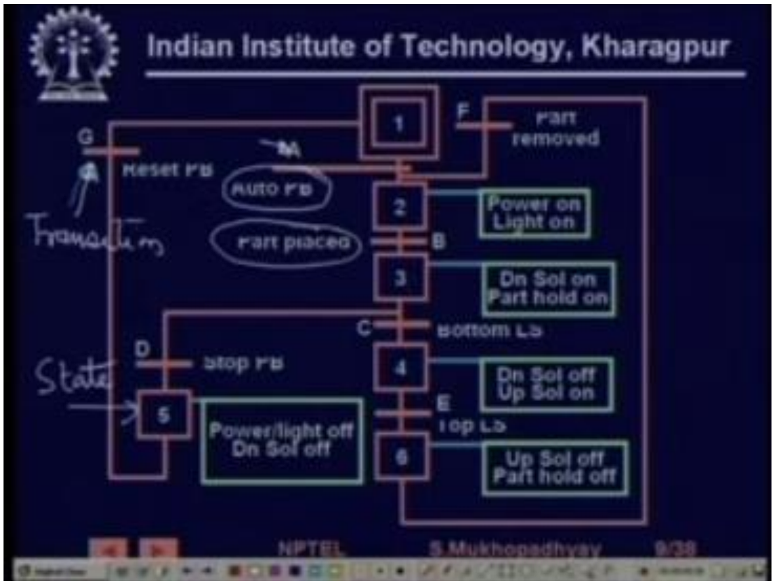
And a set of transitions for example, this is a transition this is not a good color go back to white so this is a transition and this is a state, so this is a transition and this is a state so what the system does is that system the system actually during its life cycle or during its activity the system actually moves from states to states through transitions, so it actually spends time most of the time in the states and transitions are generally assumed to be momentary that is, it is assumed that insignificant amount of time is required to change states. So you see that it says that.

(Refer Slide Time: 13:14)



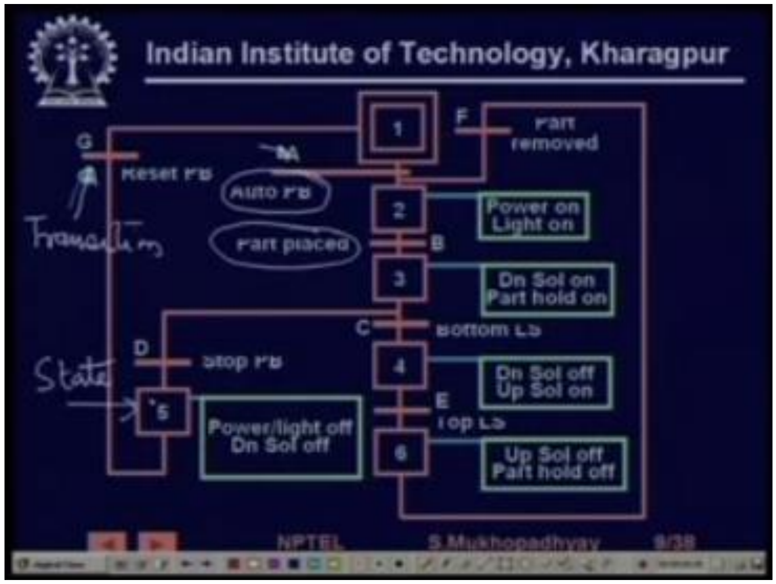
Initially the when you have double square it means that is the initial state, so if initially if the auto push button is pressed this is a transition A which gets activated which will take place and take the system from state 1 to state 2, if the auto push button is pressed so this is the transition condition. You can have much more complicated conditions in this case we have very simple conditions.

(Refer Slide Time: 13:42)



And then if this transition occurs then the system comes to state 2, in state 2 again if this transition B takes place whose condition is part placed it will come to state 3 so in this way depending on how the sensors are bringing in signals from the field the various transitions will be enabled and the system will hop from state to state that is the behavior of the system.

(Refer Slide Time: 14:09)



On the other hand these green rectangles indicate that at each state which are the outputs which are on, for example you can see that in state 1 nothing is on not none of the outputs are exercised while in state 2 the power and lights are on in state 3 the down solenoid is on actually this is the state when the solenoid is coming down, so you see that this is the initial state here the system is switching on the power and the light and possibly waiting for part place signal to come so it might spend some time here then it is coming down so takes time there, then from here it could either go this way or go this way and depending on which one of this have come.

So it may so happen that the bottom limit switch if the stop push button has not been pressed then eventually it will the bottom limit switch signal will come and then it will come to step 4 in which it will activate these outputs. On the other hand if before the bottom limit switch is pressed if the stop push button is pressed then it will come to this state where it will simply stop and put the power on light off.

So you see so this is the way using a graph of nodes and edges we can describe the behavior of the system unambiguously. So now so this is what we know, what we call the state transition diagram.

(Refer Slide Time: 15:39)

State No. / O/P	1	2	3	4	5	6
Power/Light	0	1	1	1	0	1
Part Hold	0	0	1	1	0	0
Up Sol	0	0	0	1	0	0
Dn Sol	0	0	1	0	0	0

And these are the outputs which are exercised at different and that is actually also encaptured in what is known as an output table, so the output table says that among the four outputs that we have namely power light, part hold, power lights which part hold up solenoid and down solenoid. Which are what are the status whether they are on or off in the various state so there are six states and there are four outputs so it says that the power light switch stays on in state 2, 3, 4 & 6 right.

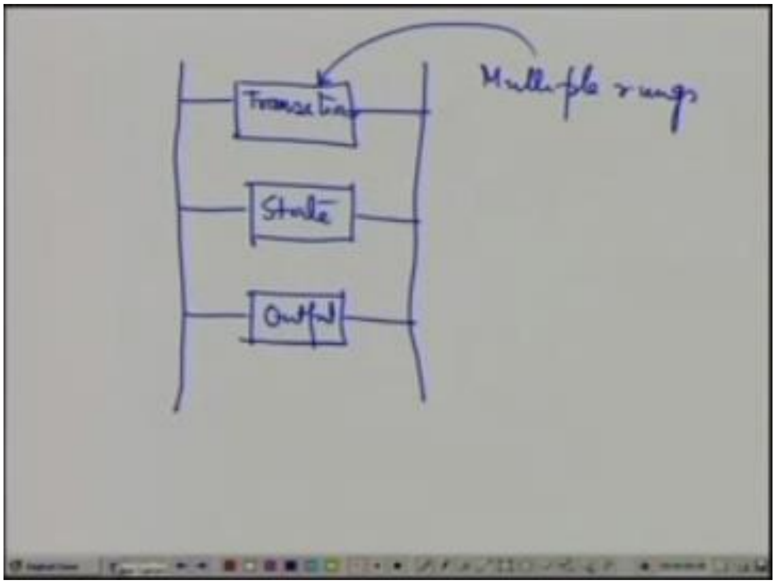
While the part hold stays on only during 3 and 4 up solenoid is 1 during 4 and down solenoid is 1 during 3.

(Refer Slide Time: 16:29)

Having done that we can start developing our, so you see we have seen that the as the system moves on the various state logics and transition logics are alternately computed. So first there is a state in which some state logic will be satisfied depending on that outputs will be exercised after that at some time some transition logic will get satisfied, so now the system will come to a different state so the previous state logic is going to be falsified and the new state logic will now become true and then based on that the corresponding outputs will get exercised.

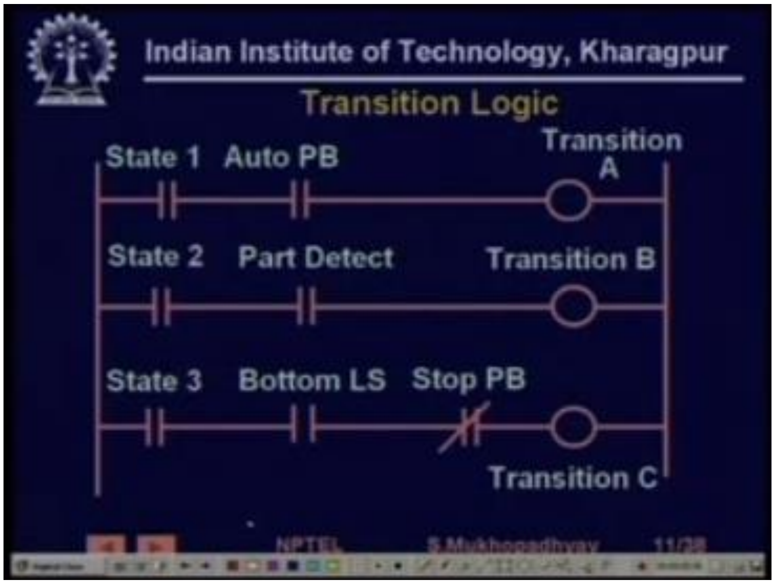
So this we have to now capture in a relay ladder logic program, right. So we will organize our program into three different blocks the first block the will contain, so maybe I will choose this pen now.

(Refer Slide Time: 17:45)



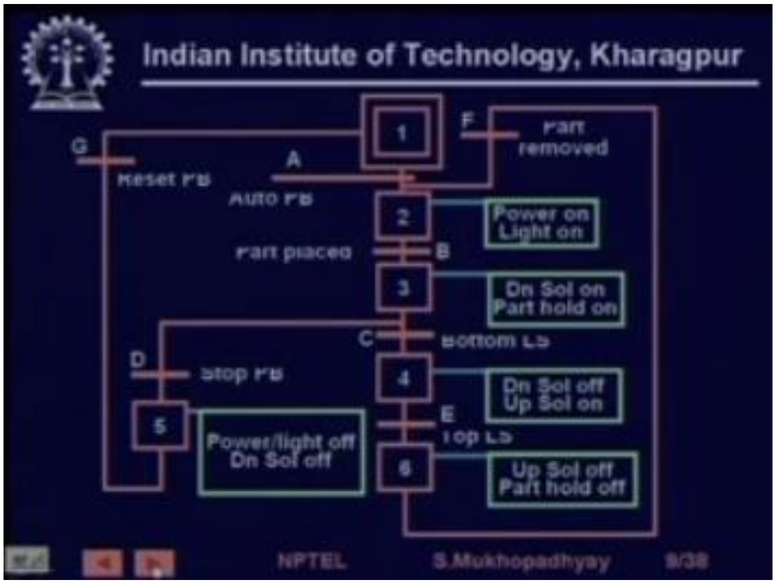
So the relay ladder logic will consist of three different blocks, okay. The first block will contain the transitions this is multiple runs then the state block and finally the output block, so we will now describe these three blocks in the case of this example.

(Refer Slide Time: 18:22)



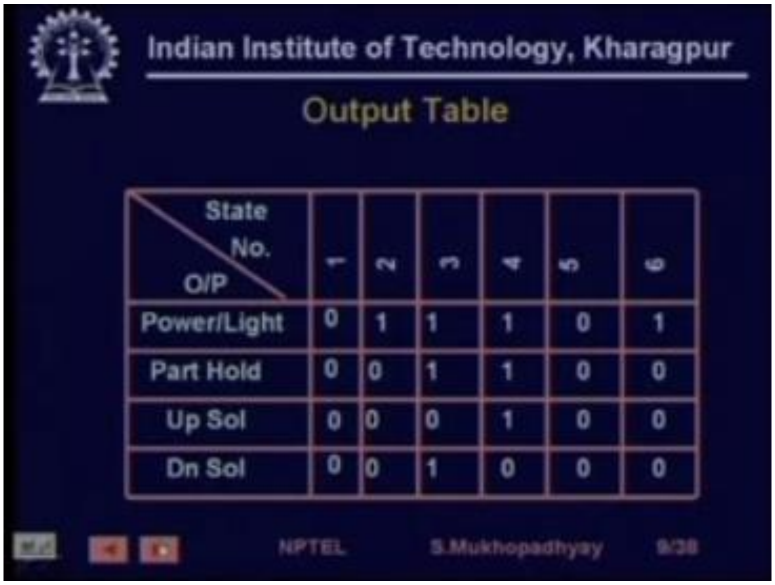
So let us first see the transition body for example, what does it say, it says that if when we will transition A logic will be satisfied. The transition A logic if you recall brings the system from state 1 to state 2, so if you wanted to see that.

(Refer Slide Time: 18:49)



We could go back just for once so you see here transition A takes the system from state 1 to state 2, transition B takes it from state 2 to state 3, transition C and D are in parallel it could take state 3 to either 4 or 5.

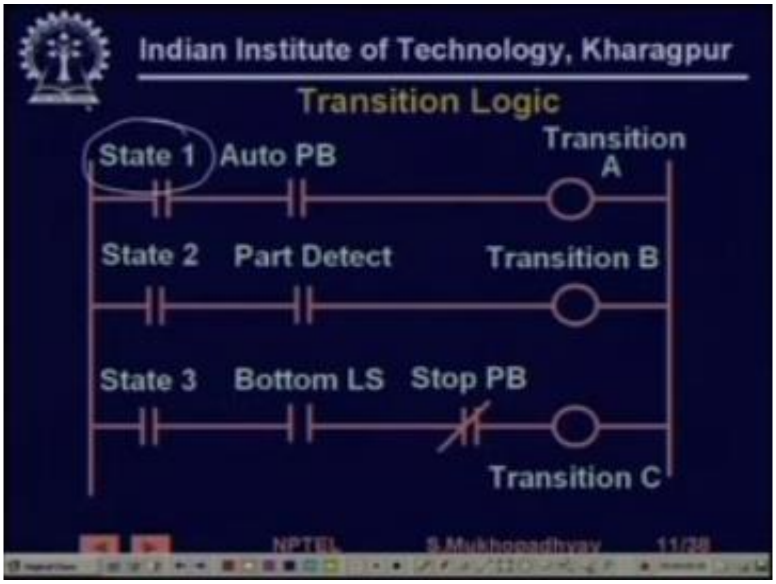
(Refer Slide Time: 19:15)



State No. / O/P	1	2	3	4	5	6
Power/Light	0	1	1	1	0	1
Part Hold	0	0	1	1	0	0
Up Sol	0	0	0	1	0	0
Dn Sol	0	0	1	0	0	0

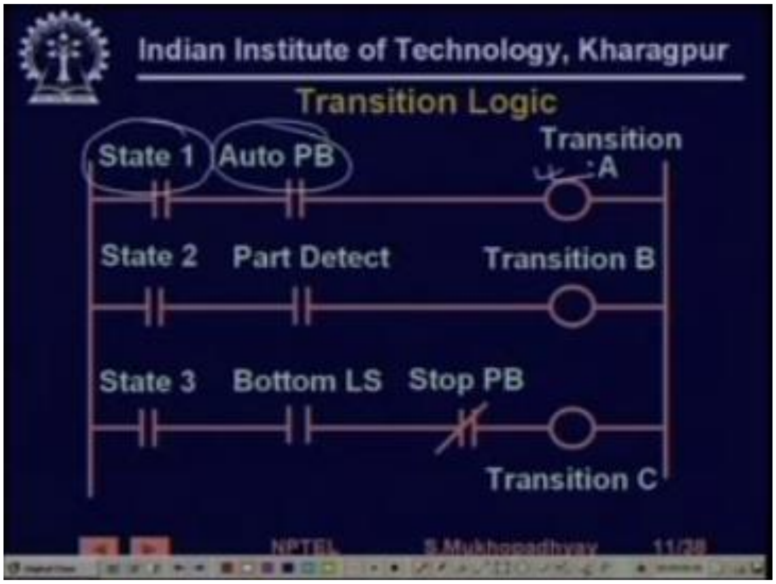
So let us remember this and then go ahead.

(Refer Slide Time: 19:17)



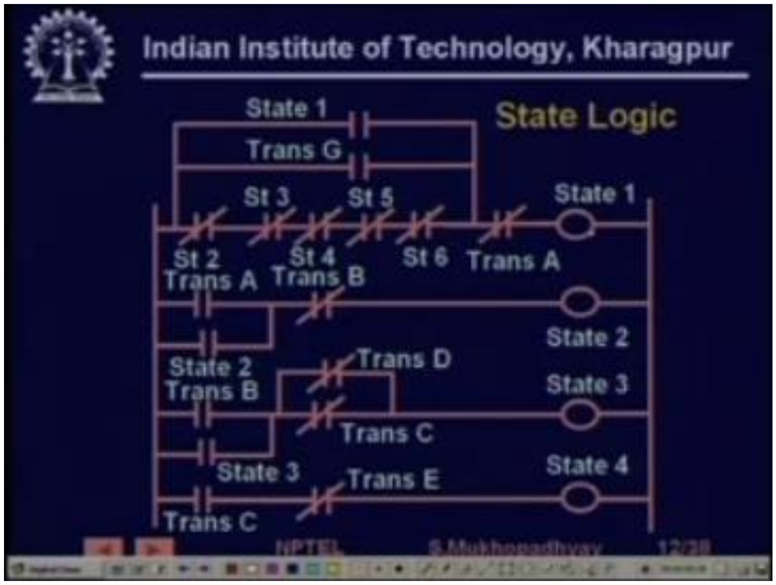
So it says that if the system is in state 1, if it is in state 1 that depends on the state logic then the corresponding to the state corresponding to every transition we have an output coil and corresponding to every state we have output coils. So this is actually an auxiliary contact corresponding to the output coil called state 1, so it is an abstract variable actually so it says that if it the system is in state 1 then this contact will be made.

(Refer Slide Time: 19:49)



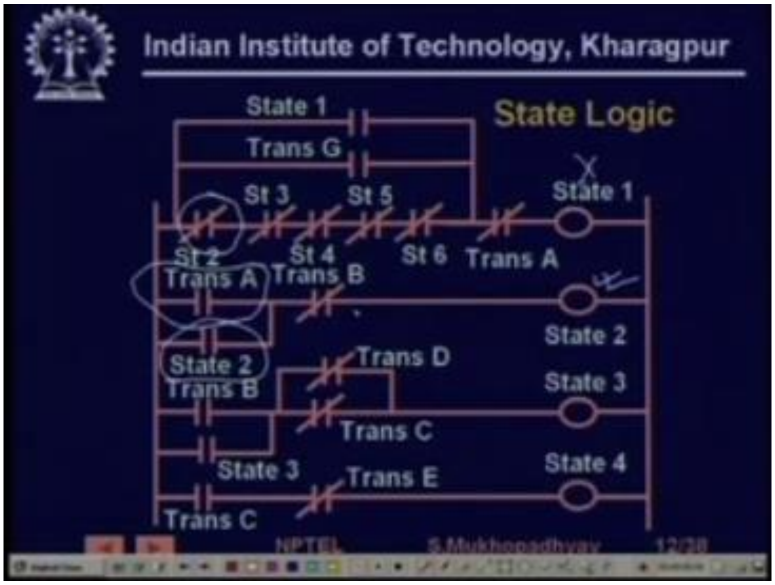
And at that point of time if the auto push-button signal comes then transition A will get enabled, so it will be on right, so if you have modeled if you have model or system well then at a time only one transition will get on right, if you have if we do not consider concurrency then at a time only one transition will get on. Now what will happen, now in the next stage so now transition A becomes on and state 1 was already on, so at this point of time we come to the state logic.

(Refer Slide Time: 20:29)



So now let us see what happens in the state logic, in the state logic see state 1 was on, right now because state 1 was on and because auto push button was pressed transition A became on, so the computation came from the transition logic to the state logic, so what happens is that it found state 1 is on.

(Refer Slide Time: 20:57)

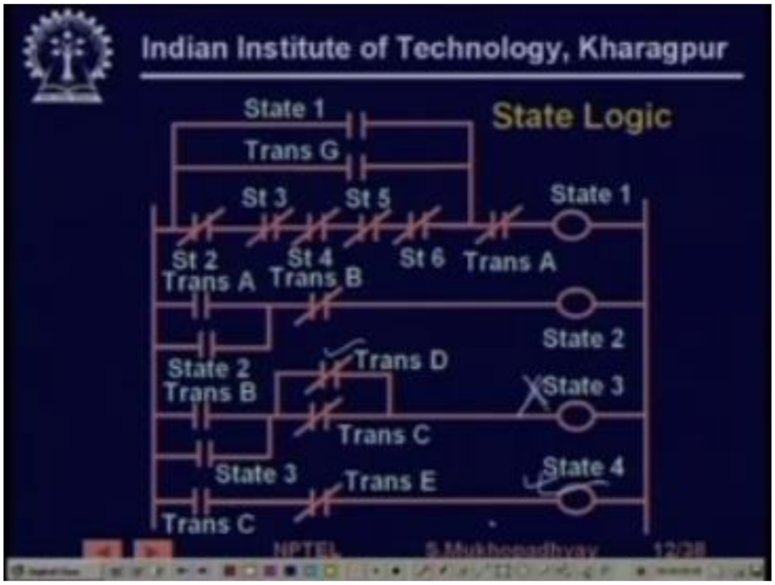


So what happened is that it found that the transition A is on at this point of time it found this transition on because transition logic has been already evaluated and it has been found to be true. So therefore, this auxiliary contact will be closed, so therefore now state 2 will be on. Now once state 2 is on two things happen, firstly you see in the next because state 2 is on this will be on this contact will be on and this contact will now be off.

So in the next cycle in the next scan cycle when this runs will be evaluated this will go off and because and transition A will because see this will go off and this will go off therefore because transition A will go off so therefore state 2 this can go off it does not matter because this is on so therefore this will stay on, so therefore it says that now the system is in state 2, right. Now when in this way again when state 2 is on at that time in the next cycle some other transition will become enabled depending on what sensors says one so what sensor signals are coming.

So similarly it will turn out for example in state 3 now you at some time transition B will take place transition B means that, transition B is let us see transition B so transition B is part placed correct.

(Refer Slide Time: 22:47)



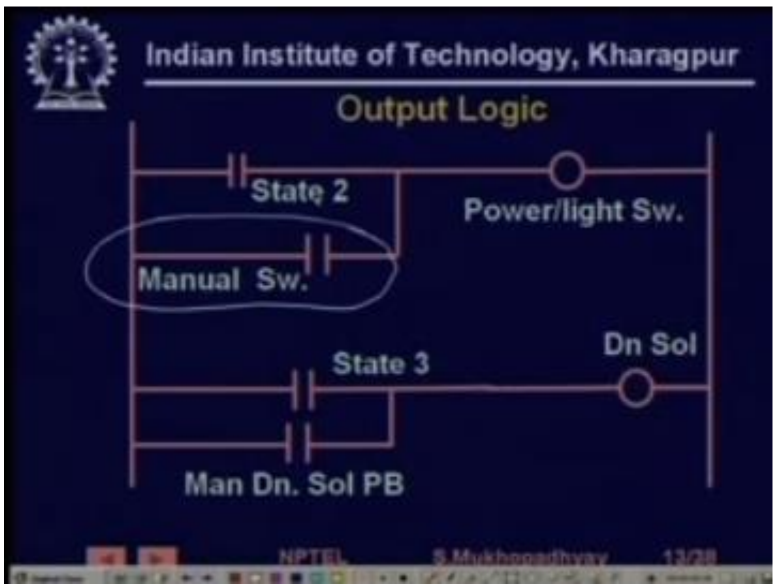
So when the part will be placed then if the if that part place signal comes then what will happen is the transition B will be on and these are not yet enabled so therefore state 3 will be on right. On the other hand while state 3 is on if either transition C or transition D occurred transition D is due to the stop push button being pressed and transition C is due to the bottom limit switch being made, if one of the any one of these occurs then it will no longer been state 3 but it will go to either transition state 4 if transition C occurs state 3 will be falsified and state 4 will become on.

On the other hand if transition D occurs then this will be falsified and then state 5 which is not shown here will be come on. So you see that mechanically once we have developed the state graph we can simply mechanically describe its behavior. So corresponding to every transition we are going to have one run, corresponding to every state we are going to have one run and as I have described we are going to put the enabling logics.

So we are going to say just from the graph that if when the system is that state 1 if auto push button is pressed it will go to state 2, simple this logic which is given from the graph will take every transition and will write the corresponding logic in the transition logic. Similarly we will say that if transition X has been enabled then it will reach this state, so we can do it from the graph mechanically just one by one this writing can be actually written by a program itself.

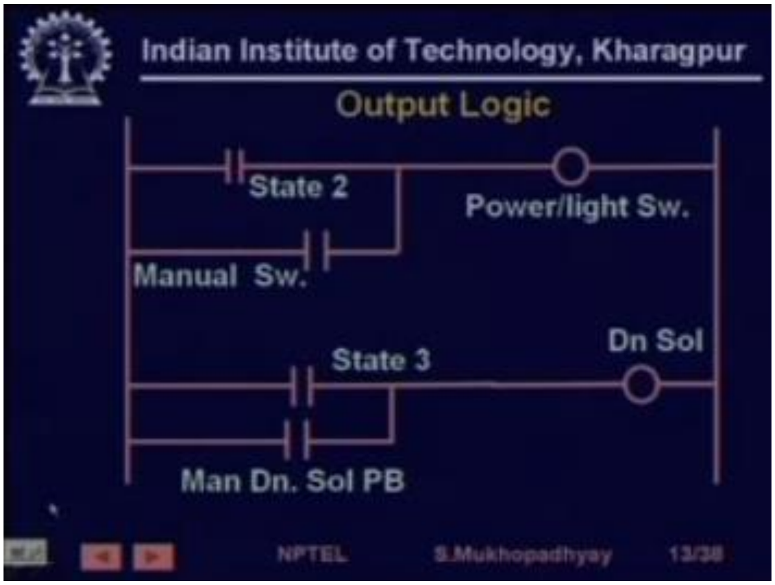
So one need not really think too much about the logic one should think about the logic while he is drawing the diagram after that the programming becomes automated this is very useful, okay. So now next we will have the output coil, output logic is very simple very, very simple especially in this case.

(Refer Slide Time: 24:55)



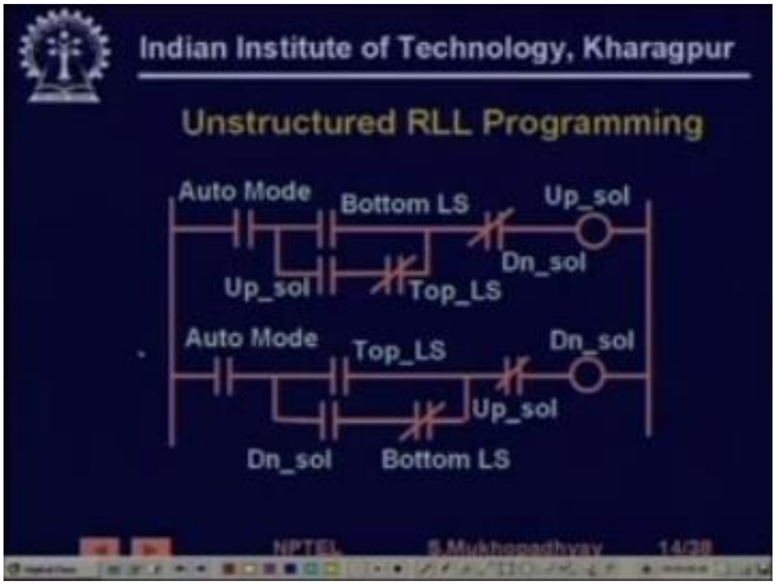
So the output logic says that if you are in state 2 then power lights which should be on as we have given in our output table. So only thing is that look here that we have also added some manual switch you know it can be sometimes we may need to check we may need to do things manually also, so the power 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 maybe made on. Similarly we can have a manual down push button.

(Refer Slide Time: 25:33)



So we can this is just to demonstrate that you can put additional logic to include manual operation of the system. So in this, so otherwise this program simply says that while you are in state 3 down solenoid will have to be activated very simple.

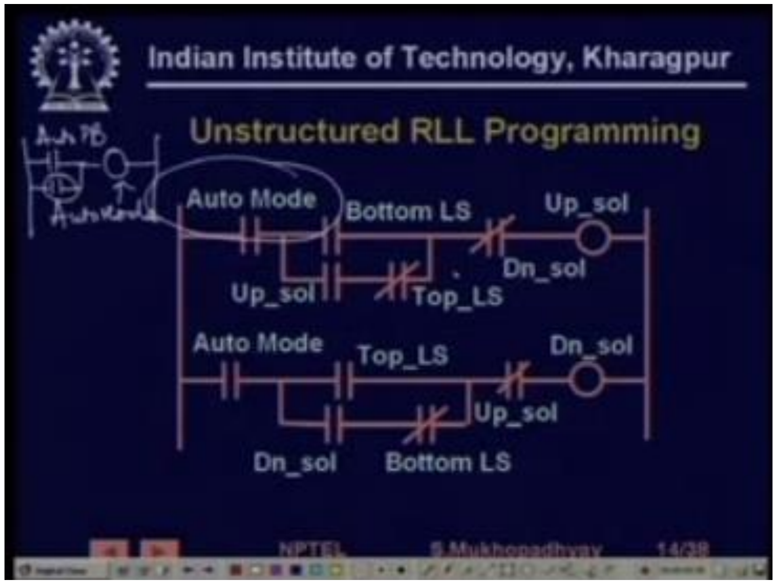
(Refer Slide Time: 25:56)



Compare this with the kind of programs that we had written earlier in fact or 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 just 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 kinds 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 using some temporary variables.

(Refer Slide Time: 26:37)



But the kind of here you see if you look at this program says it is very complicated logic and I am not even one 100% sure it is very difficult to be 100% sure whether this logic is full proof it says that if the auto mode by the way this auto mode is actually you can it is an auxiliary contact corresponding to some logical variable which you can set for it by a by a by a simple run that 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 a auto mode coil and this will be an auto mode auxiliary switch so that the PB can be released so this is the sort of you know persistent input, so if this auto mode is on so it says that this all the everything will work only if the auto mode is on, and then if the bottom limit switch is made and the down solenoid is not on, then the up solenoid can will be energized. Similarly and once the up solenoid is energized it will remain energized until the top limit switches looks okay, but one is never sure and especially when problems will have 200 states then will be impossible to write such direct programs, when a chances will be very high then if somebody wants to write it you will make mistakes.