Digital Circuits and Systems

Prof. S. Srinivasan

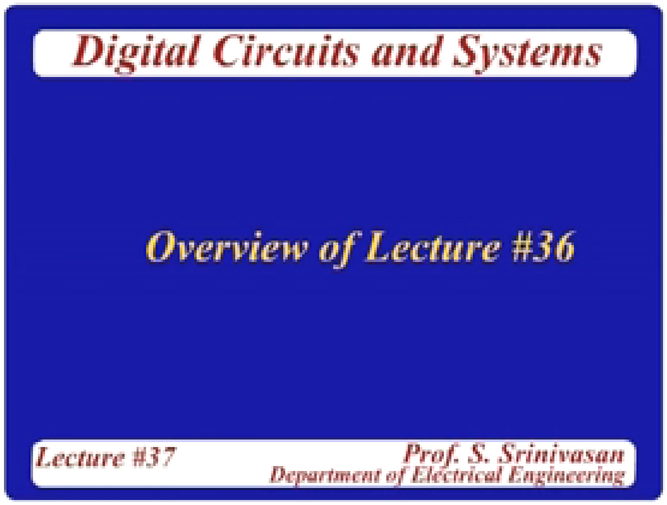
Department of Electronics and Communication Engineering

Indian Institute of Technology, Madras

Lecture - 37

System Design Example

(Refer Slide Time: 1:06)

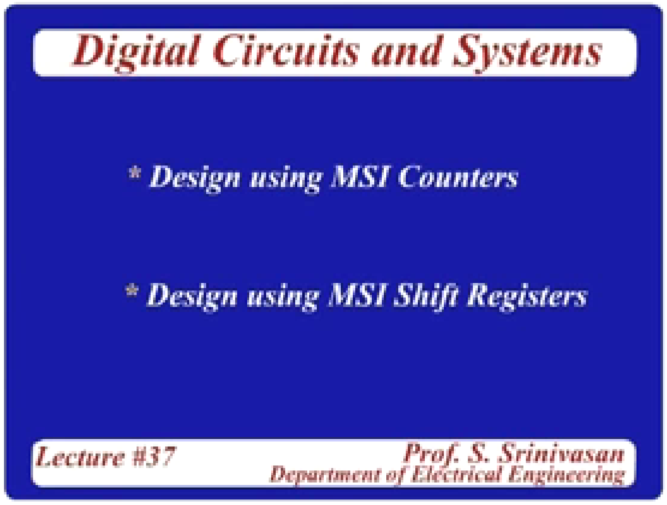


Digital Circuits and Systems

Overview of Lecture #36

Lecture #37 Prof. S. Srinivasan Department of Electrical Engineering

(Refer Slide Time: 1:17)



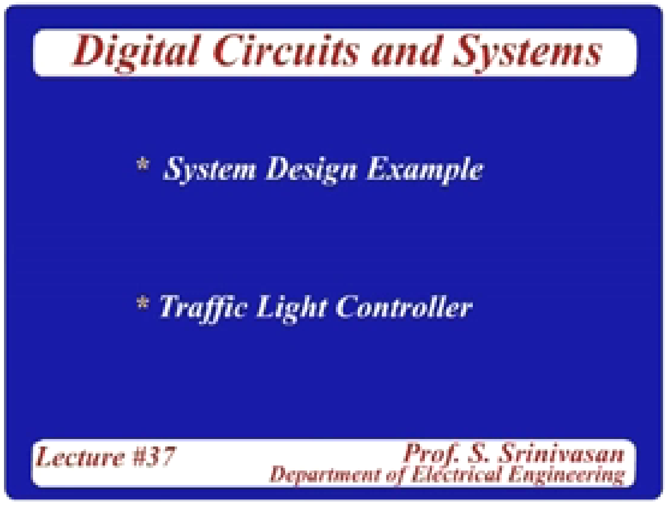
Digital Circuits and Systems

* Design using MSI Counters

* Design using MSI Shift Registers

Lecture #37 Prof. S. Srinivasan Department of Electrical Engineering

(Refer Slide Time: 1:39)



Digital Circuits and Systems

* *System Design Example*

* *Traffic Light Controller*

Lecture #37 Prof. S. Srinivasan Department of Electrical Engineering

So far we have seen design of circuits, sometimes even sub-systems using different types of ICs, small scale integrated circuits, medium scale integrated circuits and large scale integrated circuits. In the case of combinational logic apart from the gate implementation we have also seen how to implement circuits using multiplexers and programmable devices. In the case of sequential circuits we talked about flip-flops, counters and registers. We have taken some examples sometimes examples such as code converter or pattern generation, pattern recognition looking for a particular pattern etc.

These were simply examples to illustrate various design techniques using combinational logic, building blocks like gates and sequential logic building blocks like flip-flops and also to implement or to illustrate the concept of using MSIs and LSIs for implementing these circuits. Some of them you call them as sub-systems because of their complexity.

But in a digital system design, most of these applications today in electronics are digital systems be it a traffic light controller or a controller to control the moment of a lift or elevator or to control the flow of fluids in a chemical plant or controlling the level of various fluids, it could be automotive electronics, it could be a simple home appliance like washing machine or remote control of the TV or it can be a little more complex device such as flight control in aircraft or for flight simulation or for rocket launch or it could be a simple computer or a calculator, whatever be the system the complexity appears to be formidable.

When you are asked to design, let us say, a huge system, a digital system, you will be overwhelmed with the complexity of the system. But if you think for a while all the tools that you have developed in this course so far, basic implementation of combinational building blocks using SSI, LSI, MSI components, basic implementation of sequential building circuits, sequential circuits using building blocks such as flip-flops and registers, the same techniques you will be using again and again.

We talked about minimization using Karnaugh Maps for gate reduction, then we talked about multiplexer based design, we also combined the combinational and sequential designs together in a single IC like Programmable Array Logic device where there is scope for putting up a combinational part and a sequential part in the same circuit. So really when a system is given to you, if you start analyzing the requirements, if you identify clearly the specifications, if you identify clearly the input signals available and the output signals required we must be able to come up with a state graph and once we have a state graph you know how to implement it. Of course it may become a very large problem with large number of states and then we have to go for some techniques like CPLDs and SPGS. But conceptually we have most of the tools, most of the techniques available to us now. To illustrate this point we are going to look at some of these examples.

How it is important to understand the design requirements? How it is possible to understand fully this system we are going to design; the input requirements and output requirements, and how this can be met easily and how to build them using these components. So you have to identify the circuit, the system requirements very clearly before we can choose the components. Once we choose the components we should know how to use them properly then the design becomes straightforward. This approach is called top down design approach.

Take an example of a computer design, a computer looks formidable when I ask you to build a computer or a calculator but on the other hand if you start thinking of computers as a collection of building blocks such as arithmetic logic unit, memory unit, input unit, output unit then functionally understand each one of those and look at the arithmetic logic unit as an example, after all it contains arithmetic and logic functions and we know how to implement arithmetic functions, it can be gate oriented, it can be multiplexer oriented, it can be a MSI for adder or MSI for adder subtractor.

Likewise logic functions we know how to implement and we should know how to combine in the same block an arithmetic and logic function. So when you go top down and down and down you will able to break a system into sub-system, a sub-system into functional units, functional units can be either available, it can be an available component which is available readily as SSI, MSI or LSI or we build them using SSI and MSI and then of course comes the interconnection and controlling of them and that is where the state graph comes into picture, how do we control, what is the sequence of events that takes place and how the signals from one unit feed into the next unit and how the output of the second unit affect the third unit and so forth. So it is clearly the understanding of the specification of the problem and drawing a proper state graph. Once you have that rest of the design falls in place and we will soon realize that we have the power to design a whole range of digital systems.

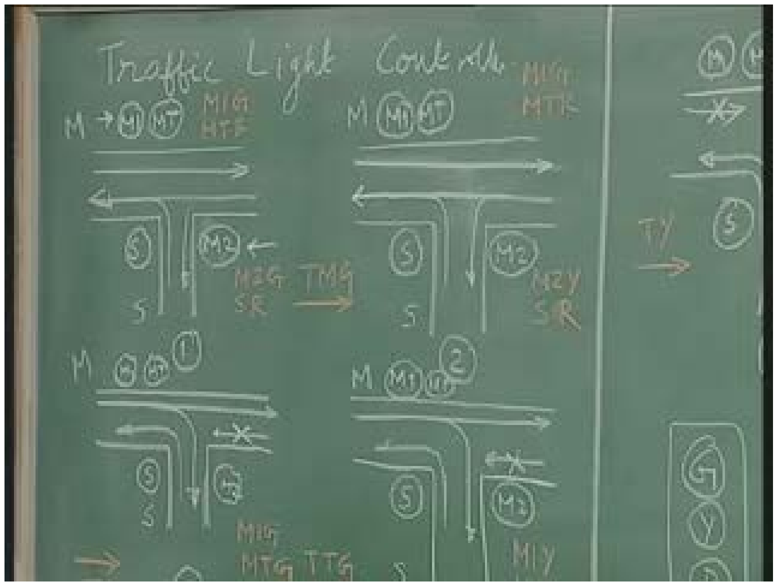
We are going to illustrate this concept by means of two examples; one is a simple example and the second one is a slightly more involved. I will tell you why it is slightly more involved not because it is bigger in size but because it is a slightly different concept we are going to talk about over there. In the first example we will consider a traffic light controller. We know that traffic lights are installed in road junctions and they come with a variety of features; it could be a simple two-way intersection where one street is green for a while and the other street is red then after sometime they switch. Then there can be a street with turn signals. So, in addition to main street and side street signals we need to have a turning signal, when you have a turning signal you have to make sure some traffic is allowed in a certain direction and not allowed in certain other direction because it will collide with the traffic coming through the opposite direction. We can add to that some pedestrian crossings.

There are several ways in which you can control this timing of these signals, you can have a set timing. For example, the main straight signal will be on for a while let us say two minutes or one minute and side street signal will be on for thirty seconds then it will repeat or you can have a timing based on the requirement. The main street will be on, green, and side street will be red all the time until there is a traffic detected in the side street, that can be done by means of sensors or switches.

We can put some light switches or sensors in the side street, whenever there is a need the side street can go green and the rest of the times it can stay red, then we have to make sure that this does not happen very frequently because you do not want to interrupt the main road traffic very frequently so we may have to put a condition that if traffic occurs in side street and if the main street is on at least for a given amount of time then and only then it can be recognized and the lights will change. So you can go on inserting conditions making it more and more complex.

Then you can have pedestrian switches as I said. Sometimes the traffic is so fast and the pedestrians are left with no option to cross the street safely. So there you can install some pedestrian switches and once in a while but again not too frequently, once in a while you can detect the pedestrian switch and if it is on the traffic can be halted to let the pedestrians cross. Therefore, some of these examples or some of these features can be built-in and you can decide on the timing, sequence etc. Now this will be illustrated by a simple example we will take.

(Refer Slide Time: 11:47)

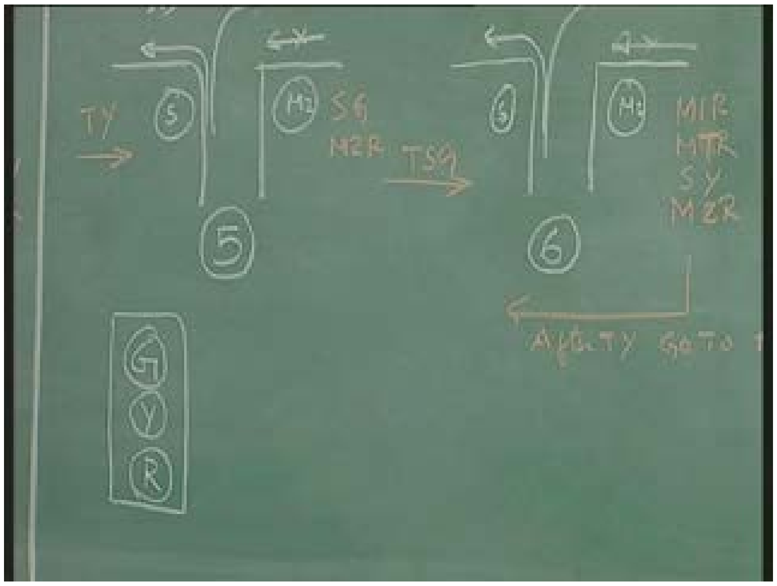


So in this class and may be in the subsequent one or two classes we will talk about system design and the first system we are going to design is a traffic light controller. To make it simple, as I said, I am going to assume an intersection of a main street and a side street. This street I am calling a main street (Refer Slide Time: 12:12) where traffic is allowed in both directions and this street is a side street. Normally main street traffic is allowed that means there is a green in both directions and the side street can take a left turn here from main to side or side to main, left turn. That means even when the traffic is on here one can make a turn from this street to this street or from this street to this street of course safely making sure that you don't collide with the oncoming traffic from this direction. So this I will call, this is one state (Refer Slide Time: 12:57) there are several states in which lights cannot exist. The first state in which it is found is green on this side, green on this side, this traffic is not allowed, the turning signal is red and the side to this, and for this also you see a turning can happen from main to side or from side to main, both signals will be red normally. So, to identify this I am going to call these lights……. M1 is the light for traffic from left to right on the main street, M2 is the light for traffic from right to left on the main street, M1 and M2 are the normal main street lights and both are normally on green.

MT is the turn signal for turning from the main street into the side street, MT, M turn the main turn that means turn from main. And S is the side street signal which is used to turn from side into main. So normally M1 is green, M2 is green, M2 is red, S is red that's why I wrote here M1 green, MT red, MT is the turn signal, M2 Main 2, the second signal to the main, green, SR side is red this (Refer Slide Time: 14:27) is the first state which is the usually available state.

As I said, at this point I can decide to wait for the traffic condition to be detected or I can go for a timing. So let the traffic be on for a while. To make this problem simpler, I am going to make it as a time dependent, change in lights rather than a sensor dependent. That means after a while the time for which the main is green is called TMG I am calling it TMG. After the time TMG has elapsed that means time T for which the main is green, after that interval of time we have to prepare to let the main street traffic go into the side street. That means we should prepare to take MT to green. But I can't abruptly do that because in order to do that I need to make M2 red. I can't make M2 red from green suddenly because that will create problems for people who are traveling there at a fast rate or for high speed vehicles so you go through an intermediate light called amber or red or yellow some people call it yellow, some people call it amber. So this is the disposition of the lights. You have a green light, yellow light and a red light. Each of these lights, when I say M1 M2 MTS each of these is a fitting with three independent lights; green, yellow, red so before you make this turn I need to make M2 yellow.

(Refer Slide Time: 15:55)

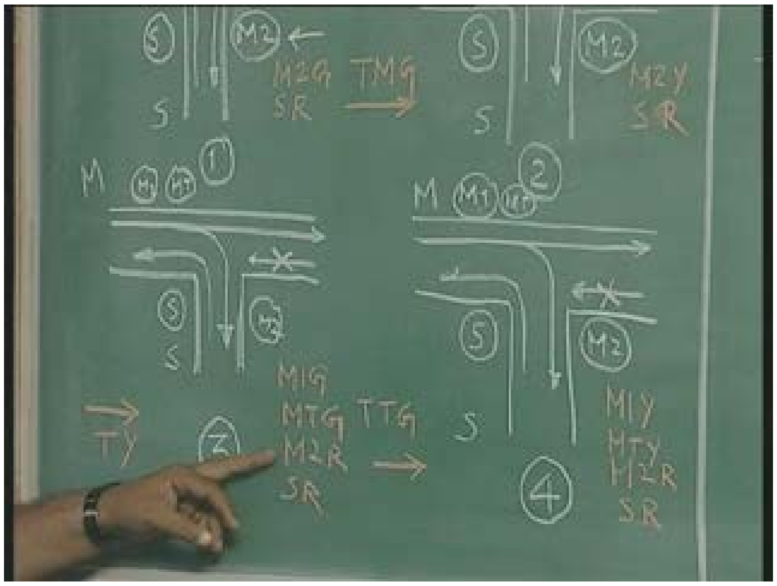


M1 will continue to be green, MT will continue to be red, side will continue to be red but M2 will change from green to yellow that is my second state. My second state will have M1 green, MT red, M2 yellow and side red. That means I am preparing for the turning from the main street to the side street by switching on the turn signal as green not immediately but after a while.

Again the yellow light, how long should I keep it, that dependents on the traffic flow conditions and the speed of the vehicles etc. Normally the main street yellow has to be longer than the side street yellow because main street traffic is faster compared to side street traffic. A vehicle which is coming in a faster rate needs more time to stop so yellow for the main streets should be longer than yellow for the side street. We will not go to all those nitty-gritty, we can always incorporate all these features, changes later on.

Therefore to make the problem as I said, to keep its specification simple I am going to call it TY. TY is the yellow, the time for which any light is going to be yellow, whether it's a main light, side light or turn light. Whichever light is going to be yellow the time for which it will be yellow before it becomes red we will call it TY. So, after TY from here state S 2 after the elapse of TY we go to the third state. In the third state I have a main that continues to flow, there is no problem but M2 is now red because from left to right traffic is no problem, from right to left I cannot allow traffic because I have to allow traffic from main to side that means I need to allow for this turn. If you want to allow for this turn I need to block this traffic that means M2 should become red. M1 will continue to be green, MT will now become green from red, red will become green so that the traffic will be allowed, side of course remains…… this light (Refer Slide Time: 18:25) again we cannot allow the traffic from side to the main that is in this direction so it is continuously red.

(Refer Slide Time: 18:40)



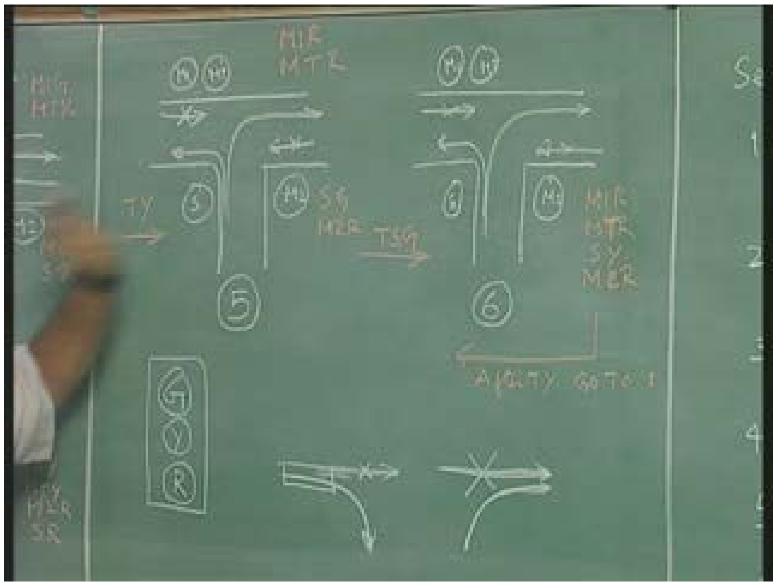
Therefore, side is red, M2 is red, MT green and M1 green. This (Refer Slide Time: 18:44) is the third state in which the circuit will be found, the traffic light controller will be found in this third state. Now, how long will you want this turn signal to be on? Let us call this TTG, TG stands for turning green, turning signal green and T is the time. The time for which the turning signal is green is called TTG so, for the period TTG these conditions will prevail. At the elapse of TTG, at the end of TTG what will happen is I should stop this traffic from main to side and let the other side street where this side street traffic in the back will remain.

So originally I allowed from main to side now I should allow from side to main. So TTG has to be turned to off made green and S should be made….. this should be made red and this should be made green. But I can't make it red directly from green, we have to go through yellow. So this state, state 4 is main turn, the turning signal in the main road will be yellow. M1 is yellow. Why am I making M1 yellow is because unlike in the case where the traffic from main to side is organized because I am assuming that the main street is going like this (Refer Slide Time: 20:26) this is the side lane, it is two lanes of traffic and those who want to turn are already standing here in a separate lane. I need not stop this for letting this on.

Now, on the other hand, when I am turning from here to here I cannot let this happen here because we don't know how well these lanes are in this street. Supposing if the lanes are not well-defined there is a possibility that this traffic coming from the side road into the main road may collide with the traffic which is going directly from main road from left to right. In order to avoid this while I am allowing this traffic to continue when this is allowed I am not allowing this traffic (Refer Slide Time: 21:26) to continue when this is allowed. So from main to side I am allowing main street traffic from left to right whereas from side to main I am not allowing the main street traffic from left to right. That means I don't want this to happen which means I should not only make this MT red but I should also make M1 red. The main should be red, T should be red but both cannot be made red immediately, they have to go through yellow so this state makes M1 yellow, M2 yellow, this continues to be red of course and this continues to be red. This is the condition which is a warning to the main road users whether they are directly going or turning, now the light is going to be switched off to red so be careful either stop, stop or prepare to stop. So that condition is called state condition 4 or state 4 which will again last for the period of T yellow.

I am not again distinguishing between yellow timing and the main street and the side street or the turn signal so T yellow is the time period I am allowing for this condition to elapse, TY is the time for which I am allowing this condition to last so after the end of this TY main one will become red so that the main street traffic from left to right will be halted, MT will be stopped, the traffic from left to right which is also going to turn that also will be stopped so M1 is red, MT is red, M2 is already red so no problem, so this has stopped (Refer Slide Time: 23:10) and this has stopped allowing traffic from side to red. So side to red is allowed that means side becomes green.

(Refer Slide Time: 23:20)



All the while I was allowing this free traffic (Refer Slide Time: 23:24) from side to main on the left hand side there is no problem, side to left to the main side, this is not being hindered by anybody, it is always on. So what is called free left in our traffic parlance and in our system, in our rows when you say free left it means you can always turn without having to wait for any signal. Of course you have to make sure that there is safety, it is safe to turn, make sure that there is no oncoming traffic and that you will not collide with anybody or any vehicle, so as long as it is safe you can turn without having to wait for a signal. Therefore, such a condition is somehow called free turn or free left in our parlance; I find this peculiar coinage in our traffic, in our cities in India.

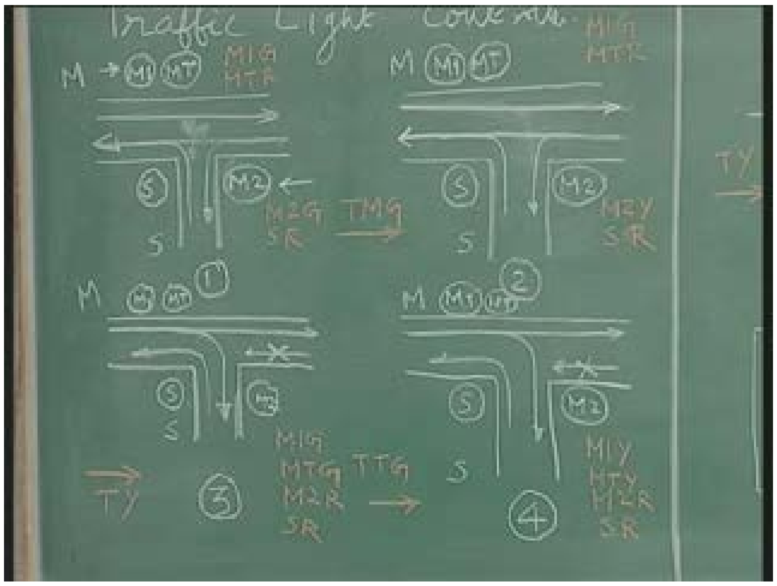
Therefore, I have a free left here, free left, free left, free left, free left, and free left (Refer Slide Time: 24:25). Here this is free left but not here. This is not a free left anymore because this red is…… M2 is controlling not only direct traffic but also the side traffic. Main to side from right to left, right to left main and right to left side both are controlled by M2 so it is not free left. When this is on, this can also go and when this is on this can also go, when I am stopping this…., this also cannot happen because I do not want this to go on and collide with this. Similarly, when I am stopping this, this cannot go.

Now, at the end of TY as I said the side is green, this is the only traffic allowed apart from the free left here, this traffic cannot go further, cannot turn, this traffic cannot go further and this cannot turn. This is state 5 (Refer Slide Time: 25:24). And this will last for a while where we want to prepare…., it depends on the time we want the side green to be active, so decide the period for which you want the side green to be active, so decide the period for which the side green is going to be on, so when SG is on this will be like this and at the end of the period SG we call it TSG the time for which this side is green, TSG.

We will go back to this condition where we will have to prepare to make this red that means this side to main will have to be halted and you will have to let this traffic flow freely in both the directions because that is the condition which we want most of the time. So, for that what I need to do is to make side red but the side cannot be made abruptly it has to be made yellow first then red. So, this condition, this continues to be red, this continues to be red but side becomes yellow. Once this happens we know that this side row traffic has to stop.

Once this is yellow the traffic here will have to prepare to stop so at the end of the period TY, after TY we go to 1, 1 is what? 1 is when left to right traffic is allowed to the main street, right to left traffic is allowed in the main street, no turning is allowed from main to side, no turning is allowed from side to main but this free left at this left (Refer Slide Time: 27:00) is allowed at this corresponding…. at exactly this state. This […27:04] state is nothing but main left to right, main right to left, this traffic is not allowed and this traffic is not allowed, this is allowed as I said, always and this is allowed as long as this is…. this turn is allowed as long as this traffic is allowed.

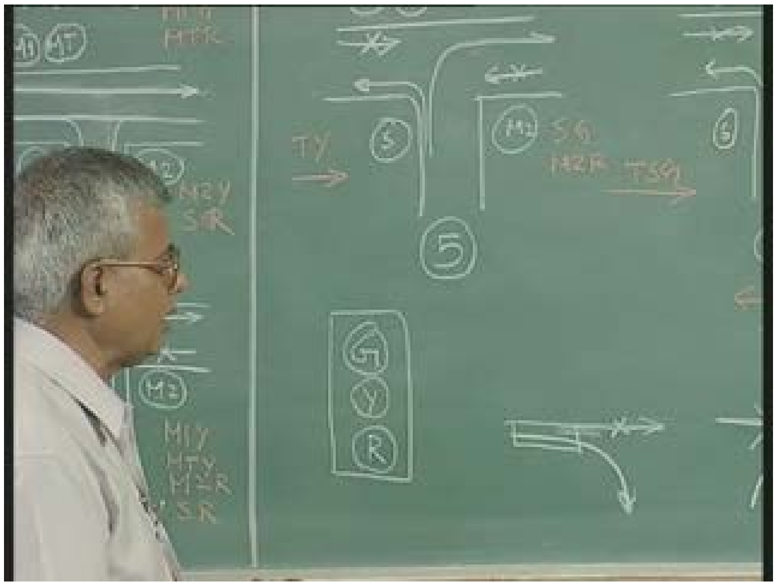
(Refer Slide Time: 27:41)



So 6 states under which in which this circuit can excess, this problem is a example of how a specification has to be translated in a state machine, a state machine means a state diagram and for that I need a very clear understanding of the problem. These figures give a very good understanding of the requirement. What are the various inputs and outputs? Inputs are TY the yellow time, TMG main green time, TTG turn green time, TSG side green time. So TY, TMG, TTG and TSG are the four inputs, systems. What are the outputs? Each light M1, M2 the main lights; MT turn light and S side light, each of them can have a green, yellow, red so these are the outputs. So there are 4 lights each with 3 combinations and 4 outputs, these are the inputs and outputs and the circuit can exist in various conditions and move from state to state depending on the conditions that becomes a state graph. And once we have a state graph you know how to implement it.

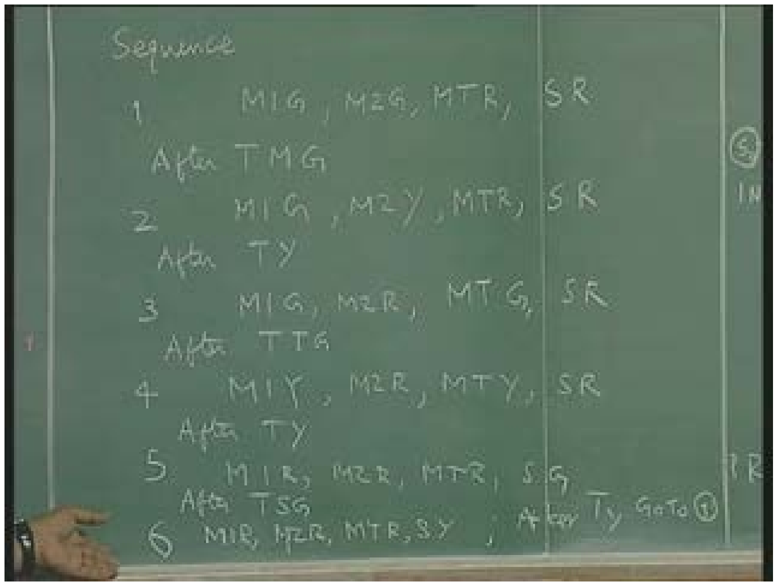
So the problem translation, the translation is a problem from specification word description as they call it. From a clear description of the problem in words, English sentences, to drawing of the state graph is the most critical thing, once that is done the rest is easy because we know the procedure; how to do the implementation using standard techniques, you can use gates, you can use multiplexers, you can use ROM, PROM, you can have PLAs PALs, you can use flip-flops for state variables or you can use registers, you can use shift registers, you can use a counter or whatever.

(Refer Slide Time: 29:59)



So this is the algorithm they call it, algorithm is the step by step operation, the sequential operation. These diagrams are only an illustration to help us to understand the problem. The word problem is translated into this (Refer Slide Time: 30:13) this is the sequence of events. This sequence of events is a description in an algorithmic sense.

(Refer Slide Time: 30:20)



Sequence

1 M1G, M2G, MTR, SR

After TMG

2 M1G, M2Y, MTR, SR

After TY

3 M1G, M2R, MTG, SR

After TTG

4 M1Y, M2R, MTY, SR

After TY

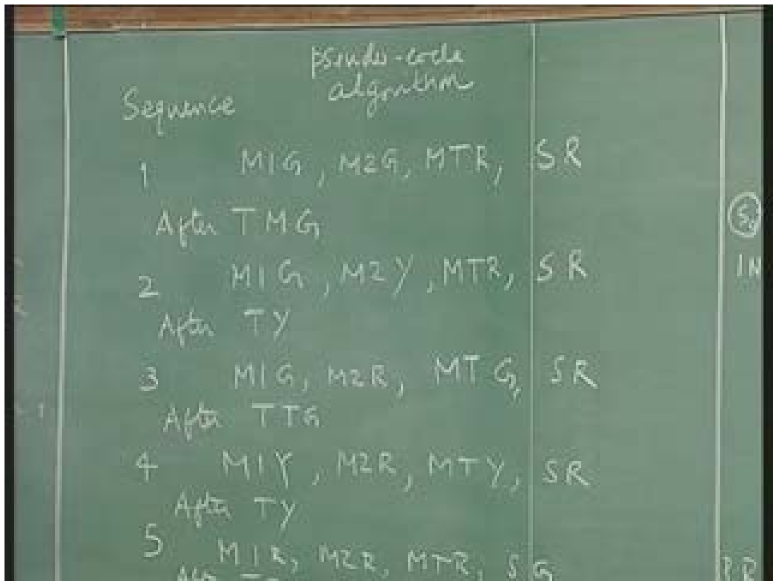
5 M1R, M2R, MTR, SG

After TSG

6 M1R, M2R, MTR, SY ; After TY GoTo ①

Those of you who are familiar with programming, computer programming, you will call it a code, a program, it is not a program in the sense that use syntax of a particular language like C language or anything, it looks like a program but it is not a program, you can call it a pseudo-code if you want. Pseudo-code is a program but not using syntax of a particular language, it is an English description or you can also call it algorithm.

(Refer Slide Time: 31:02)



pseudo-code algorithm

Sequence

1 M1G, M2G, MTR, SR

After TMG

2 M1G, M2Y, MTR, SR

After TY

3 M1G, M2R, MTG, SR

After TTG

4 M1Y, M2R, MTY, SR

After TY

5 M1R, M2R, MTR, SG

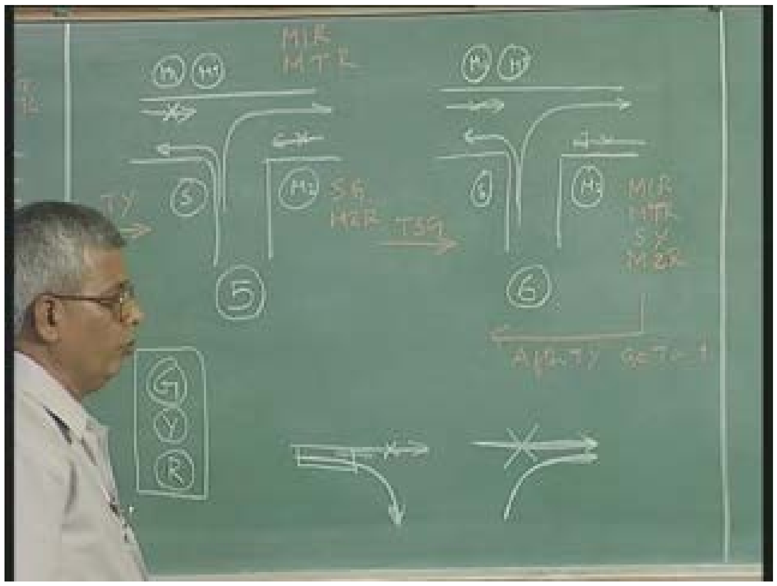
What is an algorithm? Algorithm is a step by step behavior of the circuit based on various conditions of input and what conditions of output are possible. So what are these steps again to go through? Normally the circuit is with; main 1 green, main 2 green, main turn red and side red.

This condition is allowed for a period call TMG main green time. After the end of main green time this M1 continues to be green and M2 turns yellow to start stopping the traffic from right to left on the main street. For MTR the turn signal is still right, still red, still red, side signal is still red. Then we allow this condition for a period we call yellow T yellow which is uniform for all the yellow lights even though main yellow is normally more than the side yellow but we are keeping it equal to make it simple.

After the period of T yellow, main 1 continues to be green that means left to right traffic is allowed but right to left traffic is stopped M2 becomes red. MTG turn light is allowed, turning traffic from main to side is allowed by a green light, side red that means turning traffic from side to main is not allowed. This condition will last for a period you call TTG, TG stands for turn green, the period for which it is green is called TTG. After the elapse of TTG I have main 1 yellow, main turn yellow. I am going to stop the traffic from left to right on the main street and also the turn traffic from main to side.

To allow the side to main traffic, in order to do that I need not go through the yellow sequence. So M1 yellow, MT yellow, M2 continues to be red because the traffic from right to left is still not allowed because once there is a side to main traffic I cannot allow this traffic, I cannot allow this traffic from right to left, it continues to be red, side traffic is still red, of course we have not cleared it. This condition (Refer Slide Time: 33:35) prevails for a time TY the time for which the yellow signal is defined. At the end of TY both the main 1 and the turn signal from main both becomes red. That means we are cutting of traffic from left to right as well as cutting of traffic from maid to side. M1 red, MT red, M2 is anyway red so side is allowed that is green. That means it is the first time and the only time the side traffic is allowed.

(Refer Slide Time: 34:15)

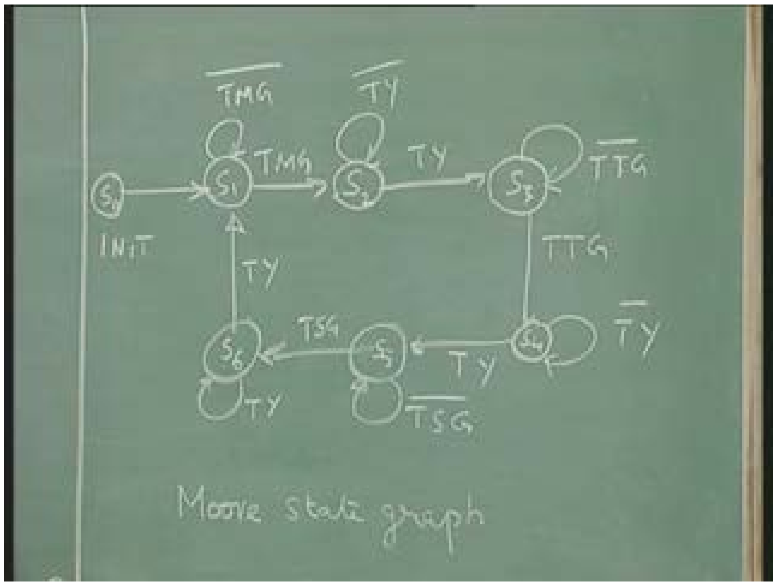


This condition will exist for a period of TSG, SG stands for side green and T stands for time. TSG is time for which the side is green, at that period side green will have to become yellow, become eventually red. So, after this period side becomes yellow so that becomes red later on but since it is still yellow traffic is not completely halted so I cannot allow any other light to change so M1 has to continue to be red, M2 has to continue to be red, MT has to continue to remain red and side will become yellow. And this condition will last for a period of TY, TY stands for the yellow light period in this case from side to main. After TY we go back to 1 which is our primary interest. Main 1 green and main 2 green, traffic in both directions and no turning on this side and no turning on this side.

This is the algorithm (Refer Slide Time: 35:41) we can call it algorithm. Algorithm is a step by step procedure of the events or the behavior of the circuit. Pseudo-code, people use it because pseudo-code will look like a computer program, that is not a program in any sense because program requires some syntax for a particular language, C language, C plus plus but it is not like that. But really it is a sequence of events that takes place. And each of these sequences correspond to one state of the machine. So we have the state machine in which the circuit has to exist in each of these states and they go from each one of these states to another state based on certain input conditions defined by the timing: TY, TTG, TMG, TSG. We can put additional conditions like pedestrian switches, traffic flow conditions by sensors and all that but we are not doing all that now in this case to make it simple.

How is this state graph going to look like? First of all let us see, the state graph is, first of all we have to see whether the state graph is a Moore graph or a Mealy graph. This is a Moore graph. Why is it a Moore graph? Moore graph defines outputs for states; Mealy graph defines outputs for state and a particular input combination. In this particular case states define the outputs. The circuit can exist in all these 5 states or 6 states 1 2 3 4 5 6 and for each state the outputs are frozen, these are the outputs. So, when the outputs are given fixed corresponding to states that is the Moore state graph. So we already said that there are 6 states S 1 S 2 S 3 S 4 S 5 S 6, in S 1 the circuit remains, there is a timer which goes, this time is called MG, TMG, as long as TMG has not elapsed, is not over the circuit remains in this state, TMG can be several clock cycles. The clock cycles are different from the state cycles because once…… this circuit need not be very high speed circuit. When you try to decide what clock, at what frequency this circuit should operate it need not be very high, it is not like a Pentium processor or anything, we are talking of traffic lights which change slowly, we talk of at least one minute for the main street green and about 30 seconds for the side street green and yellow may be 10 seconds, 15 seconds, 20 seconds depending on the conditions.

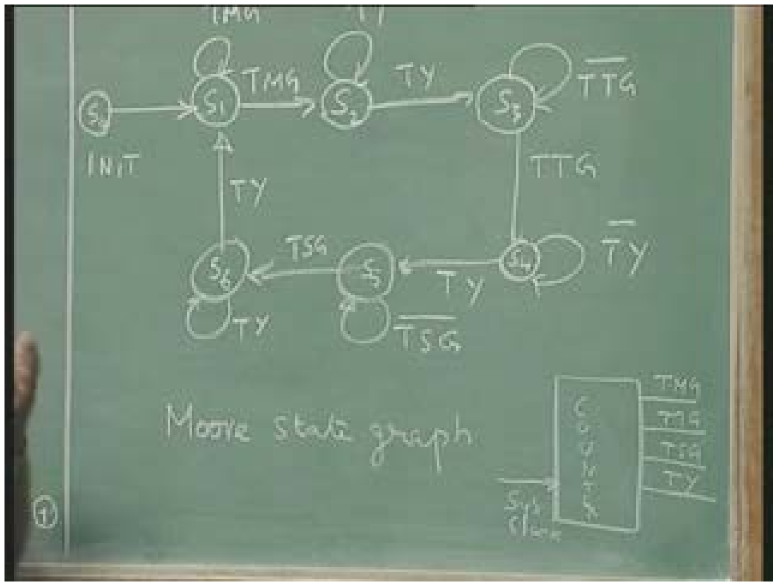
(Refer Slide Time: 37:57)



So, when we are talking about that the frequency at which the clocks have to be fed for these circuits need not be very very high. But still it will be several clock cycles in each of these timing intervals because we will have the timing interval which will be several clock periods of the clock that you feed as the system clock. So you have a system clock which is counted and when a particular count is reached we will call it TMG. So we define, for this given time, depending on the time TMG let us say it is 1 minute, 60 seconds. Assuming you have a clock period of 1 second which is too large I am just saying for an example, sixty such clock cycles exist for TMG. So you keep counting and when sixty counts are reached to this state (Refer Slide Time: 39:53) that's what I mean. So there will be a system clock and a counter. The counter counts the system clock and comes up with its values; TMG TTG TSG TY, this are the signals which are used as these signals. This can be several clock cycles of the system clock which can be itself a very very small clock, a very very low frequency clock, you need not worry.

So we have S 1, the circuit remains in S 1 until TMG is reached, when TMG is reached it goes to S 2, S 2 is the yellow state so it will remain in S 2 until TY becomes high, until that time when TY is not there it remains in this state so this is the main green, this is main yellow then when it is over we go for the turn signal so this becomes turn green. After TY the turn signal becomes green so TTG the period for which it is going to be green so until TTG elapses the circuit remains in the S 3 state, after TTG it goes to TY goes to S 4 (Refer Slide Time: 41:30) S 4 is again the yellow state that corresponds to that light, the turn signal, turn signal light has to be yellow, that time is called TY. So, in S 4 it remains until TY is reached, until TY is reached it remains in S 4 TY bar and then it goes to TSG side green. So side green will ask for a period called TSG so in S 5 the circuit will remain as long as TSG is not reached. When TSG is reached it goes to S 6 which is yellow state for this side light.

(Refer Slide Time: 42:16)



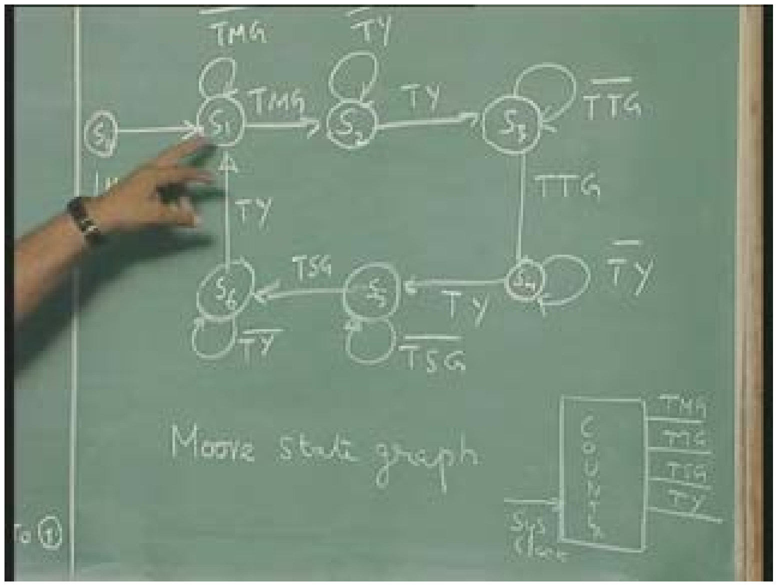
So, if that condition is TY, as long as TY is not reached in S 6 in will continue to be side yellow and at the end of TY it goes back to S 1 so this will repeat. So this is my state diagram with inputs as a said: TMG TTG TSG and TY; outputs are already defined, each of these output states; S 1 S 2 S 3 S 4 S 5 S 6 I already defined the outputs here. For example, for S 1 the outputs are; M1G MTR M2G SR, in S 2 the outputs are; M1G MTR M2Y SR. So outputs are not drawn normally in a Moore circuit, I should write the outputs in the states because the states circuits are too small and I have already used this to write the state label. I don't have the space to write the list of all the outputs in each of these states but we know we have already listed them there. So this will do.

This extra state here, you don't worry about it, this will do (Refer Slide Time: 43:38), it goes on repeating in 6 states assuming this goes forever. But normally in any condition, any practical design problem you don't design for something to go on forever, there is always an interruption, a power supply failure or the power is switched of in the night or some resetting condition exists because you want to stop the traffic light for some reason. So what we can do is, there are couple of things.

Suppose in the night you don't want the supply to be on, you can switch of all the lights in which case there is a master switch where all the lights are off. But there is nothing like off here you know because you have only three lights; green, yellow, red so if all of them should be off then the power should be off so you can switch off the power. Or you can define a state called the initial state in which you can have a blinking red, something like that. You might have seen in busy streets, at the end of the day or night, late in the night you will see the main street this yellow will blink that means proceed cautiously, and in side streets red will flash saying stop and go. When you see a flashing red light and when you are trying to enter from the side street to the main street the flashed red indicates that you should stop, look for the traffic in both directions and then only proceed. If you see a flashing yellow in the main street you know that a side street is approaching there may be traffic there, be watchful so proceed cautiously, that is the blinking yellow. So we want to put all those conditions.

We can introduce an extra state called INIT state. This could be the state in which the circuit is switched on for the first time because we are going to finally have flip-flops to implement so flip-flops can be 0 or 1 or whatever, a combination. We will find them in any of these combinations so make sure that as soon as you put a reset state from that state it goes to S 1 so this initial state is an optional state, you do not have to have it. In the initial state you can have some blinking conditions or whatever condition then it goes here (Refer Slide Time: 45:50).

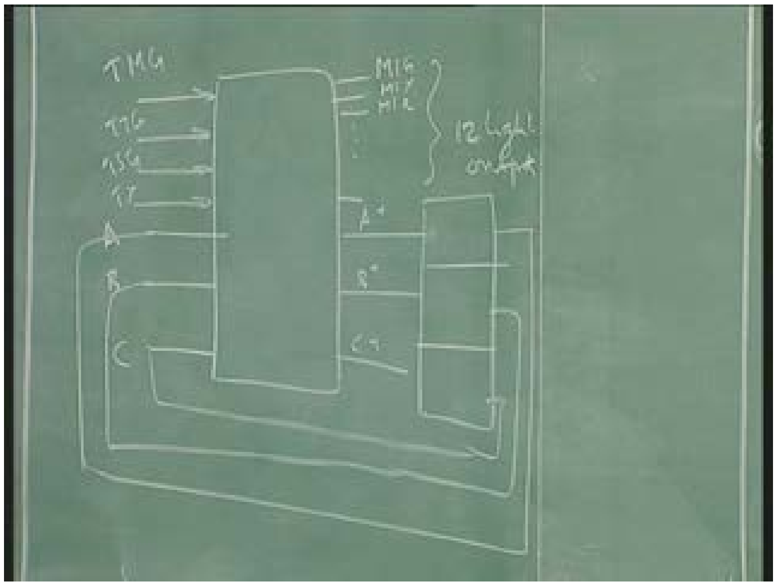
(Refer Slide Time: 45:51)



So this is the state graph with 3 variables because there are 6 states leaving out the INIT. Without this there are 6 states that means 3 variables, 4 inputs in addition to states inputs and 12 outputs, four lights each and three colors. But actually in these 12 outputs some of them are redundant. For example, some lights are complement of other lights. So we can make some simplification on these twelve lights we can remove but then finally lights are there, twelve lights are there. The logic of one light may be the complement of the other there is no need to have sometimes separate lights sometimes we can have same signal going from for two different lights if one is a complement of the other. But I am not taking all that into account right now.

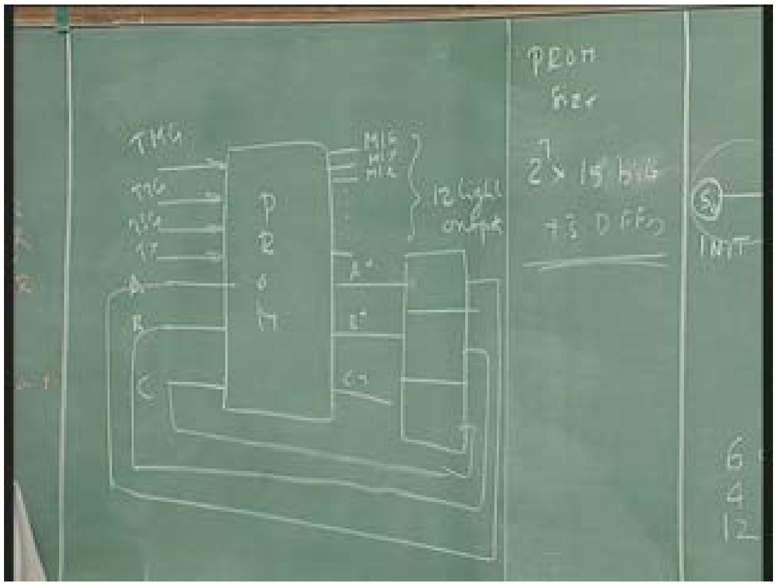
Right now I am looking into the bare state graph, same, there are 4 inputs, 12 outputs, 6 states, 6 states requires 3 variables that means 4 input variable plus 3 state variables. So can you have combination of, my combinational logic will be A B C are 3 state variables, 4 input variables; TMG TTG TSG TY and then 3 output variables then these 12 outputs; M1 green, M1 red, yellow M1 etc so 12 light outputs.

(Refer Slide Time: 48:28)



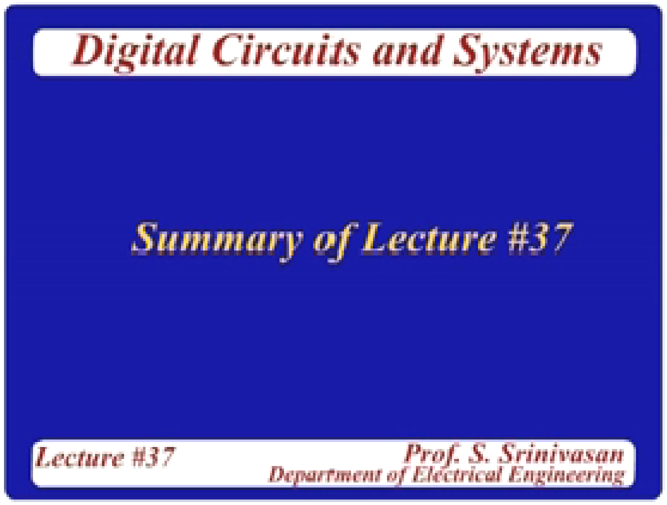
You know I can implement it using combinational logic, flip-flops and combinational logic. This combinational logic can be gates, multiplexers, PROMs or you can put the whole thing in a PAL Programmable Array Logic. So if it is a PROM solution the size would be….. it is a PROM 2 power 7, 12 plus 3. I am assuming no reduction in these signals. If I can combine some of these outputs I can reduce the size but I don't want to really do it.

(Refer Slide Time: 49:01)



Without doing any compression, compaction of the outputs I have 12 outputs for this and 3 from here so 15 outputs, 4 plus 3 is 7 inputs so 2 power 7 15 bits is the size of the PROM plus 3D flip-flops can do the job. We can look at some of the other implementations in the next lecture.

(Refer Slide Time: 49:28)



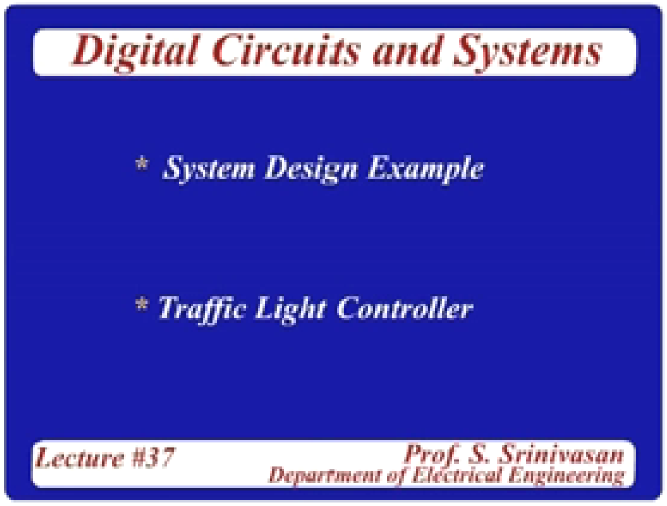
Digital Circuits and Systems

Summary of Lecture #37

Lecture #37

Prof. S. Srinivasan
Department of Electrical Engineering

(Refer Slide Time: 00:49:47)



Digital Circuits and Systems

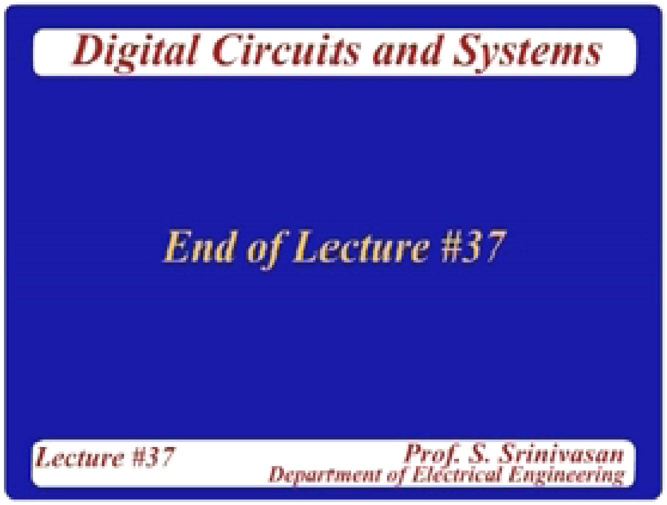
* System Design Example

* Traffic Light Controller

Lecture #37

Prof. S. Srinivasan
Department of Electrical Engineering

(Refer Slide Time: 00:49:53)

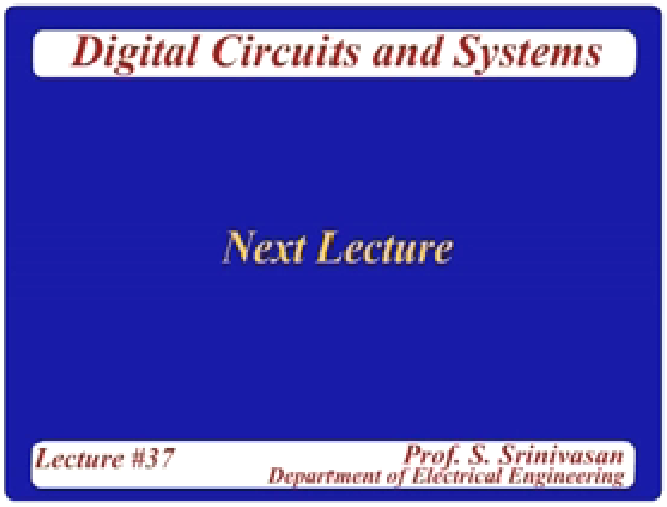


Digital Circuits and Systems

End of Lecture #37

Lecture #37 Prof. S. Srinivasan Department of Electrical Engineering

(Refer Slide Time: 00:49:59)

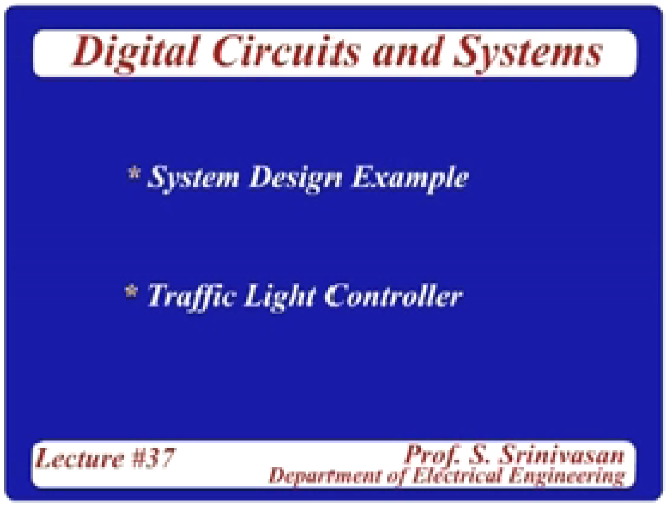


Digital Circuits and Systems

Next Lecture

Lecture #37 Prof. S. Srinivasan Department of Electrical Engineering

(Refer Slide Time: 00:50:17)



Digital Circuits and Systems

* System Design Example

* Traffic Light Controller

Lecture #37

Prof. S. Srinivasan
Department of Electrical Engineering