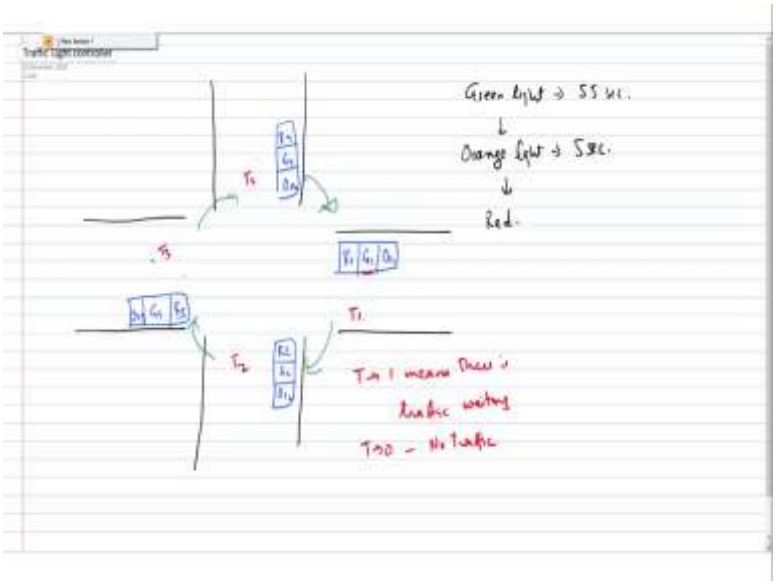
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology Ropar
Lecture 64
RTL Design – Traffic Light Controller

Hello everyone, today we are going to discuss another problem which is designing at a system level. Now, this is the last, also the last lecture of this particular module. In previous two lectures we have talked about algorithmic synthesis or algorithmic design where we have taken an C sort of application, C kind of application and then design the hardware logic step by step using that algorithm.

Now, today we are going to take a control system as an application. So, the differentiating factor in control based application is that the data path is almost not there or negligibly there. So, data path means your addition, subtraction, registers, those kinds of things will usually not be there and also in the control base system the state machine is also quite evident.

(Refer Slide Time: 1:24)



Now, let us look at this traffic light controller as an example today. So, one more characteristic of this controller based application or control based application is that the specification or the design statement is more important, what kind of functionality would be there in each case is important to notice. So, let us first think about this, let us try to understand about this traffic light controller.

So, what this traffic light controller is doing, this traffic light controller is, let us say is being installed on a road which is a four road junction. So, in this four road junction, what is being,

what usually happens that there is a traffic light at every in, traffic light on all the all the four roads. So, let us say there is a traffic light at all the places. So now, let us say that all be three sided.

So there are three lights, let us call this as R1, this as G1 and orange 1. So red, green, and orange would be there on all the three sides and we are numbering all the sides as 1 2 3. So let us say R2, G2 and orange 2; R3 G3 and orange 3. Similarly, this is the fourth side, there are also three lights here R4, G4 and orange 4.

So usually how this traffic light is working, so initially, let us say this side is green, then after some time, the second side will become green, and the third side will become green and then the fourth side will become queen. So it is a cyclic kind of a thing. So when this side is green, then all other three would be red, and when this is green, then all the rest of the three would be red.

So, now let us further consider that this is a smart traffic light controller. So, in a regular traffic light controller, it can be modal like a simple counter based state machine where you are saying that first you are in this state, then this state, then this state, then this state and then keep on rotating itself. So, but this is a smarter one, where along with having these red lights or basically these lights, there are also some sensors installed.

So, let us say the sensor here is T1, T2, T3 and T4. So, what these sensors are doing that it is trying to see whether there is a traffic in this direction have not. So T1 would be one l and say T1 means that there is a traffic waiting and if T is 0 means no traffic. Let us also take a practical assumption that, so whenever it is red, so whenever let us say this particular side is red or this particular light is red, so there would be traffic which is accumulating in this direction.

So, at that time this T1, T2, T3, so this T signal will give very accurate prediction, whether accurate information, whether is a traffic waiting or not. But let us say if traffic is running over it, then let us further assume that this T is kind of accumulating for a duration of time let us say one second. So, if for one second it does not see any traffic, any car or any bike, motorcycle, then the value will become 0. So, it will say there is not, there is no traffic.

So, in other words, what we are saying this T is an accumulated information for some fixed duration like one second, so during one second if it does not see any traffic then it will be 1. So, now, what these T1, T2, T3, T4 can do, so for example, if this particular side is green for

a moment, so now, if this T1 says that there is no traffic, so then it can move to the next direction.

So, then it can go in the next direction. So, let us say the I will put a direction arrows also that this will move from this side to second side, then it will move to third side, then it will move to fourth side and then it will again move to the first side. So, basically, if T1 is not there, if T1 is 0, then instead of G1 it will become orange and then it will become red, it will become green 2, then it will become orange and then it will become green 2.

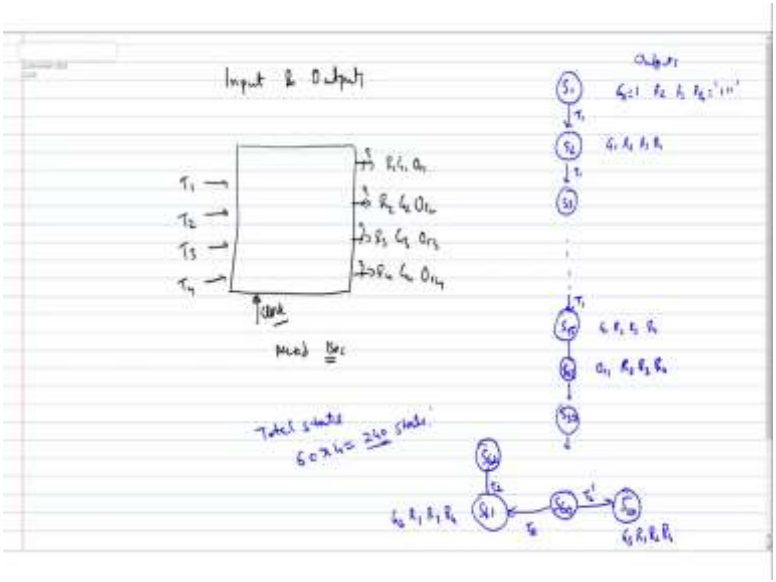
So, this is one application of this T traffic signal that it can reduce the duration of one particular side, traffic from one side. The other application could be, so let us say when T1 was green at the time T2 was red. So because T2 was red traffic would have been accumulated here. If there is no traffic which is accumulated, so that means T2 would be 0. If T2 is 0, then we can directly go from this particular side to this side.

So, this could help us in optimizing or reducing or maybe making this system as a fast forward system that if there is no traffic, we will move on to the next state. So, this is the first, this is the example of our how this T information would be utilized, the other information which we would like to say clearly here that how much time it would be there in a green state.

So, let us say green state we have to talk about in terms of seconds, so let us say green light should be high maximum for 55 seconds. And then after that from green it should become orange, orange light should be there for 5 seconds and after 5 seconds then it should become red.

If you will see, ideally, if something is green, then we can detect that how much time the red will take, so we will, we are not writing here how much time a red is taking, we are simply saying that how much time green is taking or orange is taking, red is would be determined based on the rest of the system. So this is the problem specification. Now, how should we design such a system? How should we design this?

(Refer Slide Time: 9:48)



So, let us first consider what would be the input, what would be the output. So, if we see what would be the input and output, then so what would be the input output? Input and output let us write clearly that what would be the input and outputs. It looks quite clear that our input are traffic 1, traffic 2, traffic 3 and traffic 4, while outputs are R1, G1 and orange 1; R2, G2 and orange 2; R3 G3 and orange 3; R4, G4 and orange 4. So, these are the 12 signals.

So, overall everything is 3 bit signals, so total of 12 outputs are there. What is the additional input? The additional input has to be a clock. Now, since we are talking about in seconds, let us further to make our life simpler we are saying that we should get a clock which is of period, clock period should be one second. If it is one second, then it will help us in designing. So, you can also ask question that if because you know in in hardware many a times clock is an external circuit and where you will not be able to get a precise clock.

So, here for example, we require one second clock, the input clock may be of the order of one megahertz. So, how do we get this one second clock? So, in that case, we have to use clock divider and using clock divider we get an clock which is approximately to one second. So, how to solve this problem; one, because it is clearly said that initially there would be G1 which is high and after that it would be orange of 1 and then after that it will become G2, which is high then orange 2 would be high.

So, one idea, one option could be that we can make it using a clear state machine which is a single state machine. So in that case, what we can do is, we can let us say, let us try to do that with a simple single state with one machine. So, if we do then let us say again to make things

simpler what we can do is we can design it using a Moore machine rather than Mealy machine.

So, the advantage of Moore machine is based on the state we can say that what would be the output. So, one option could be that we say each second is one state. So at S1, then so S1 means that our so the output, so these are the outputs I will say. So, here we can say it is G1 equal to 1 and rest all of them are 0. So, that means G1 is 1 and yeah, I have to say R2 equal to 1, R3 equal to 1 and R4 equal to 1; R2, R3 R4 equal to 1 and G1 equal to 1.

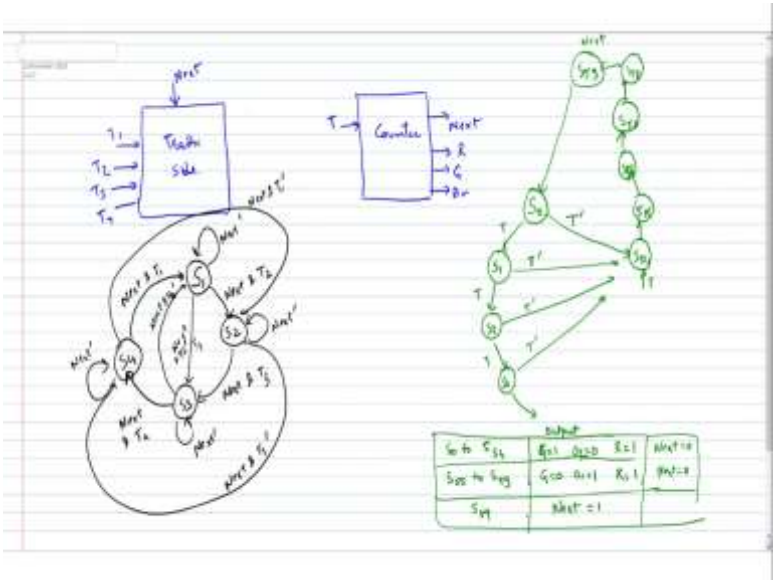
So, then if T1 is not there, then it will move to S2 in S2 also it will say G1 R2 R3 and R4 equal to 1 1 1 and then again it will go to T1 dash and then S3 so on, it will keep on doing till S55, because total up to 55 seconds, it is going to remain green. So, there would be green R2, R3 and R4 and all of these conditions would be when there is no traffic. So, when there is a traffic, so basically our T1 is 1. So, I have to rub this actually it is not correct.

So, basically it is we have to say that when traffic 1 is there, so that means there is a traffic on first side and so this is also input is traffic on the first side and then irrespective of if we have reached at S55 stage, then irrespective there is a traffic on first side or not, it has to go to S56 where orange 1 would be 1 and R2 R3 and R4 would be 1. S57, S58, so there these are all unconditionally it is going to all the states and then once it has least S60, from S60 it can go to S61.

The S61 would be that your R, green 2 would be there and red 1, red 3 and a red 4 would be there and then it depends on T2. So, then we can have S62, so this would be a flat state machine where the total number of states, here also we have to consider one more thing from S60 if T2 is there, then only it will go here, if T2 is not there, then it has to, if T2 is not there, if T2 dash is 1, so in that case, it has to go to some other state, let us say S I will write directly 120.

In that case, green 3 would be 1; red 1, a red 2 and a red 4 would be 1. So, one possible implementation would be to design it like a flat state machine where the total number of states are going to be around, total states would be 60 into 4 equal to 240. So, this is one possibility. The other possibility is that we try to partition them, we have a counter and along with the counter we can have one state which is trying to see that which particular side we are.

(Refer Slide Time: 17:04)



So, in that case, let us try to divide our state machines into two different ones. There are two state machines let us say this is first state machine, this state machine is taking input T1, T2, T3 and T4. This state machine is trying to decide that which particular side we are. So, which particular traffic side is working and what is the output to this, let us wait for a moment for that and the other state machine is a counter.

This counter is essentially ticking or it is counting how much time has already happened and when should we switch from this our traffic side states. So, basically, let us, it takes one consolidated input T, the other team is that, it will try to see that let us say the counter is working for side 1, if it is working for side 1, it will take T1 as input, when it is working on side 2 then it will take T2 as input so on so forth.

So, it will take T as input, other than that, it will also generate next as a output. So, this next is output will come here as a input. Along with that, let us also give the responsibility of generating red, green and orange to this counter state machine. So, you see, whenever we are designing any hardware or anyone doing any design, design exercise, there would be a lot of options that we can consider, here we are considering these options.

So, basically there could be there could be other way of implementing this also. So, what is the output traffic side? It looks like there is there is no output required. So, if there is no input required let us let us rub this part. So, there is no traffic. From the traffic side there is no output because red green and orange is taken as a output, again generated using counters. So, let us say that this traffic side is only telling that which side of the traffic is correct currently 1.

So, now let us see what would be the state machine for this traffic side. Now, let us consider that there is a, there is four states S1 state, now from the S1 state, S1 state means that the side is first side and if there is a next here, so the condition you have to remember that if it is a next and so first of all next has to be 1 and along with the next T1 is 1, sorry T1, yeah, along with that T2 has to be 1. If T2 along with next T2 is 1 then it will go to S2 state and similarly, if next is there and T2 is there, T3 is there then it will go to S2 state.

Similarly, here next has to be there and T4 is there then it will go to S4 side and from S4 if next is there and T1 is there, then it will go to S1 side. If next is not there all the state machines will keep on, will remain in their original state. So, one more possibility, so that if next is there and T2 is not there, then we can say next is there and T2 is not there then it will go to S3 state. We can also make it more complete by saying that next is there, T2 is not there, but T3 is there then it has to go to this state.

T2 is there; T2 is not there, but T3 is there, because highest priority for this and further if T2 is not there, T3 is also not there, then it can go to S4 state. So, similarly from S2 to S4 it can skip S2 to S4, it can skip S2 and can directly go to S4 if next is there and T3 is not there. So, that way S3 will get skipped and similarly, from S3 to S1 can also be there, if next is there and T4 is not there. If T4 is there then it will go to S4, if T4 is not there it will go to S1.

Similarly, there would be edge from S4 to, so similarly, there would be edge from S4 to S2 also when a next is there and T1 is not there. So, you can ask one question here that is the state machine complete, because it every time we are not taking into account what are the options of T2 T3 T4. So, for example here we are only considering these options that next is there and we are not talking what T2, we are not talking about T3 and T4, it is only T2.

So, first of all T1 has been taken care by the counter state machine, the T3 and T4 are not taken care because they are do not care there. They do not they does not matter whether they are 1 or 0. So, it is only the next at whenever we want to go to next state we would like to see whether T2 is there or not. So, that is one thing, the other thing is to consider all the possibilities, there should we jump from S1 to S2 also S2 to S3 or S1 to S4 also based on whether T3 is there, T4 is there or not.

So, let us to make things simpler we say that we can jump only on one particular side and can go to next side, we can skip only one side. So, then let us look at the counter state machine. So, for the counter state machine, let us have one more state machine where we are again using S1, S2, S3 notations, but these S1, S2, S3 are different for this. So, let me mark these

S1, S2, S3 with some different color let us say dark green. So, here, we will start with again S0 state or S1 be here, let us say start with 0 state and then we will move on to epsilon state.

So, when should we move from S0 to S1 state? When T is 1. So similarly, if T is 1, we will go from S1 to S2 state, S3 state, so on so forth, till we reach state number S54. That also we will say when T is 1. If T is not 1, then we will see that particular case, but from S54 to S55 and S56, these are all unconditional moves. So in these unconditional moves S56, 57, S58 and then say S59, so these are all unconditional moves. There are five states as S55, 56, 57, 58 and 59. They are all unconditional moves, they are going in this direction.

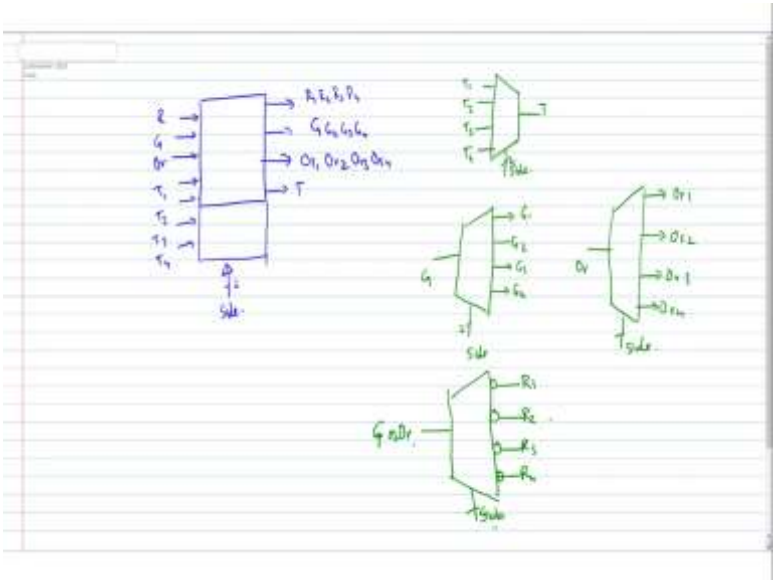
Now, after S59 we are supposed to go to the S0 state and also generate a signal next. Yes. So, let us write somewhere here. So, here next would be generated. Correct. So, what would happen if T is not there? If T is not there, which means that there is no traffic at that particular side, it could happen anywhere. So, whenever there is no traffic, what should be the solution? Should we jump to S0 state? Should we jump to S5? State? Which state should we jump?

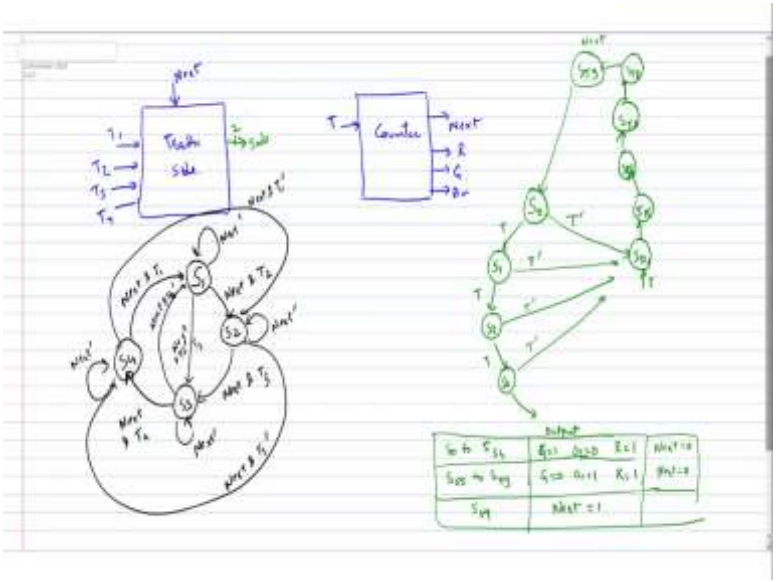
So, let us let us think in a realistic scenario, there is a there is a traffic which is flowing, after the flowing traffic, suddenly the traffic was not there. So that is why for one second there was no traffic so that is why T was 0. Now, if T is 0, if suddenly we will make, we will go to the next state, that is also not good. So, what should ideally happen that it should go to S54 state which means that there is no traffic and after that, it should, the light should remain orange for next five states.

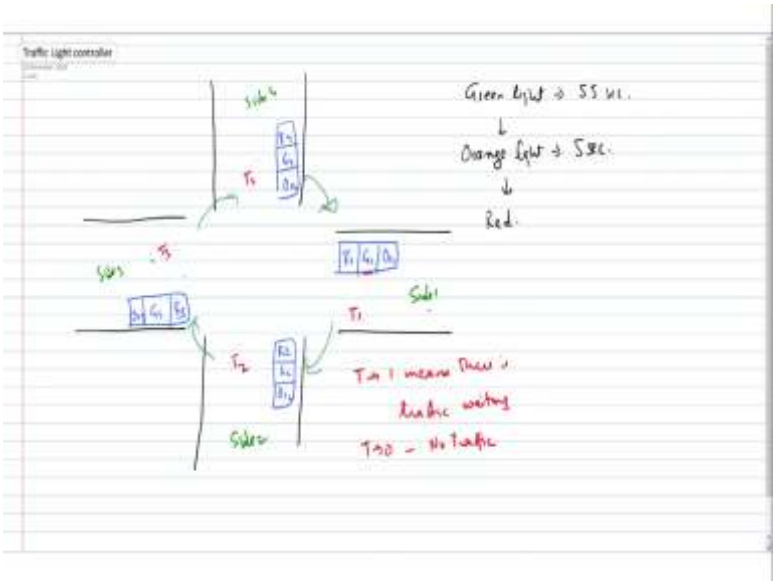
So, for all S0 S1 till S54, if there is no traffic, then we should lead to S54 status and what is the output in all of these states? So, output, so from S0 to S54 the output has to be, this is my output, so my output has to be R should be 1 and let us say, sorry, G should be 1 and orange is 0 and let us say R is also 1. So, we will see what could be the utilization of that and S55, to S59 you will see a green equal to 0 and orange equal to 1 and a red equal to 1.

So, these are the outputs for all the states, the specifically S59 will have another output which is next for all other ones the next is to be 0, here next equal to 1. Yes. So, now, there are a couple of questions which are still left; the questions is that how do we get this T information R and how this R will be given as R1, G1 as well as orange output? So, that means there has to be another module, there has to be another module, which will generate this.

(Refer Slide Time: 31:52)







So, let us design that module as, let us design another module which will take R, G and orange and T and is generating R1, R2, R3 R4. Similarly, G1, G2, G3 and 4G; orange 1, orange 2, orange 3 and orange 4. Similarly, this is generating T actually from T1, T2, T3 and T4. So, it is generating T from these four signals and from RGB, sorry, RG and orange it is generating R1, R2, R3; green 1, green 2, green 3, green 4 and similarly orange 1, orange 2, orange 3, orange 4.

So, based on what information it is going to generate, let us say, it takes 2 bit input, which is which side it is. So, which side it is, based on that it is going to generate this information. Now try to see carefully, let us look at this again. So, maybe let us look at this possibility sorry, so when this particular side, so let us mark this also as sides, so this is my side 1, this is side 2 and this is side 3, this is side 4.

So, when this side is 1, then green has to be 1 and R2, R3 and R4 has to be 0. So, those the counter state machine is clearly telling me that when G has to be 1, when orange has to be 1. So, if orange is 1 then also this R2, R3 and R4 has to be 1, so that means they will remain red. So, if I have this side information, this side information is actually captured by this state machine S1 is saying side 1, S2 is saying side 2, S3 saying side 3 and S4 is saying side 4.

So, I can say clearly here the output of this traffic side is a side which is a 2 bit output which is also essentially the state, it could also be state encoding. So the state encoding itself is the output in this case. Now based on this information, what we can Do we need to say that first of all, let us see when T1, T2, T3, T4. So, basically this is the input for us T1, T2, T3 and T4. So, these are the inputs on what basis we are going to generate T?

So, if I am in state S1, so that means, I am there inside S1, side 1. So, that means, I have to take T1 as a input for my counter. So, basically, if I am inside S1, if I am inside side 1, then based on T1 I will decide whether I have to keep on running the counter or how to go to state number S54. So, this multiplexer can clearly tell me that whether T1, T2, T3, T4 out of which, which particular T I have to select, and that selection can be done using side.

Now, the similar thing can happen with this R1, R2, R3, R4; G1, G2, G3, G4. So, basically you have two outputs there and then I can use this D mask, I can say that if G is the input here, then I can generate G1, G2, G3 and G4. So, here also I can use side. So, if G is 1 and if G is 1, based on the side G1 would be 1 and rest of them would be 0. If side is 2, then G2 would be 1, G1, G3 G4 would be 0; when side is 3, then G1 would be 0, G2 would be 0, G3 would be 1 and G4 would also be 0.

So, this way this G can help us in correcting and if G is 0, then all of them will also be 0, there would be certain time when G would be 0, so that means G1, G2, G3, G4 for all of them would be 0. And similar mask can help us, similar D mask can help us in getting the orange signals, orange, Or, orange 2, orange 3 and orange 4; again side can help us in distributing the signal.

So, if orange is 1 and side is 1, then orange 1 would be 1, otherwise all of them would be 0. So, if orange is 0, here orange is 0 then Or1, Or2, Or3, Or4, all of them would be 0. So, that you can see from here also, you see that orange would be 1 only for during this S55 to S59 states. So, only during that time, one of the orange will become 1, whether this side, this side, this side or the side. So, that help us in getting this orange signals as well as green signals What about red 1?

So, red can be done using this, so we can say if it is green or orange let us say we do not require an explicit red signal, we can say if it is green or orange then R1, R2, R3 and R4, so if side 1 is there, if this is side 1 that means R2, R3, R4 would be high and R1 is going to be 0. So, because this is 1, now they are all inverted. So, if any of them is 1 based on the side it will make R1 as 0 and R2, R3, R4 as 1.

Similarly, when side 2 is there, in case of side 2, R2 will become 1, R2 will become 0 if R or if green or orange any of them is 1 then R2 would become 1 and R1, R3 and R4 would become, sorry, I will repeat. So, when side 2 is high, so that means this side is high, so that means green or orange is there for side 2. So, that means, because side is sent as a control signal and these outputs are inverted. So, red 2 will become 0, but a red 1, red 3 and red 4 will become 1, so on so forth for all of them.

So, that also suggests us that this red output from the previous state machine could be redundant it could be generated from green and orange itself. Now let us consider one quick thing about the next also, the next state is high only in the S59th state of the counter, so otherwise next is always 0. When we are looking at the next, so basically we would like to go from S59 to S0, then at that time we are checking whether there is any traffic on the next lane or not.

If there is a traffic, then we are going to S2 state and whenever we are going to S2 state, my side encoding will become according to S2 and which will also determine that which particular T should be selected, which particular and which of course will tell us that which particular red would be there, which particular green and which particular orange has to be 1.

So, this is how we can design this automatic or basically traffic light controller which is quite efficient and which can help us in reducing the overall traffic delay.

So, what is the key message here? Breaking the state machines into, two different state machines helped us. The second thing is that we have to because it is a design exercise, I am again repeating because it is a design exercise, because of that we have to find out the interfaces of the internal circuits and we have to see that which input will go to which particular state machine. The state machine should be complete or we should be at least clear that what would happen if the other signals are there.

So, and also we have to look for all the scenarios, so that we make end to end which is complete in nature and which is doing all the functionalities which is specified by out input specification. So, with this we can close this particular lecture and this particular module also. And as a concluding remarks for this particular module, what we can say that given a problem, I will try to write that also.

(Refer Slide Time: 43:14)

So what is the concluding remarks? Conclusion is first thing that break the problem, break the problem into sub-problems and these sub-problems you have to solve independently or individually, then find the interfaces for all of them. So, finding the interface for each sub-problem is quite important, so that we clearly know what are the inputs for each sub-problem and what are the outputs, so we can generate those outputs and we can get those inputs.

So, these inputs, outputs could be internal to the system which are not reflected as the output, input output of the final systems. So, because it depends on us, so it is our responsibility to

see what could be the optimal solution, so as a design of exercise, there could be several options available to you at every moment and so you have to take, you may have to take decisions at your level and sometime this is also a possibility that specification, input specification may not be complete, so you have to think that what would be the practical way of looking at that and yes.

So, the third is that solve them, if we solve these sub-problems individually, then we would be able to complete the design exercise more efficiently. So, this is also you can say, the whole exercise is divide and conquer. So, dividing things into smaller chunks, even that these sub-problems can also be divided further and could be made even into sub-problems so that they can be solved independently. So, with this I would like to say thank you and you try to solve more problems from wherever you can find out. Thank you.