Indian Institute of Technology Kanpur

National Programme on Technology Enhanced Learning (NPTEL)

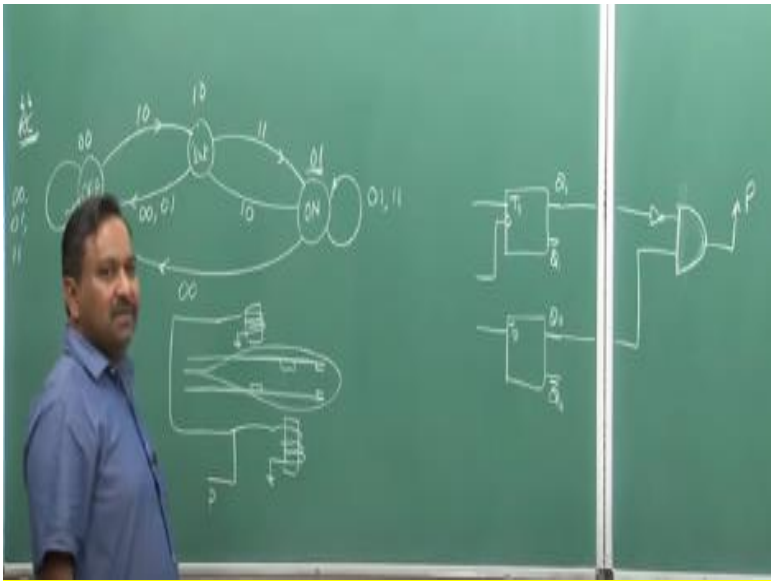
Course Title
Digital Switching

Lecture – 04

by
Prof. Y. N. Singh
Dept. of Electrical Engineering
IIT Kanpur

In the previous lecture you were looking at the crossbar and how the cross points are operated. So if you remember then I had actually drawn a state diagram there by which we can build up a control circuitry for a cross point. So let us look at that and see how we can design a cross point circuitry afford is a very trivial exercise, but we will still do it.

(Refer Slide Time: 00:42)



So the state diagram was something like this, so there was an OFF state when the cross point should not be operated or in fact it will always remain in cross point is OFF and ON state I am using another state called intermediate state only to ensure that R goes first and then the row

actually goes first and then the column is activated. So that it goes into ON state if the column goes first and row goes second, then it will not go into ON state.

So that is the reason why we actually have also an intermediate state and we also have an ON state and you can actually map or compare this figure what I had drawn in the earlier one. So in fact this is RC the row and column control which is will lead to this state is 00, 01 and 11 it will go into the intermediate state if the row is activated column is not then will it can go into intermediate state and it can only go to ON state only by intermediate one okay.

And of course if when it is in the intermediate state if R goes back to 0 so you have 00 or 10 directly moves to 01 in that case it will come back to the OFF state. It can only go to ON state when after this the column also goes up. And of course, after this even if row actually remains becomes 0 it will remain in the same state the column has to be maintained at one. And of course, if you have 10 basically the row is activated you will come back to again the intermediate one and for all other combinations.

So 01 and 11 you will retain here and if you have a 00 you will come back to this particular state okay. So with this actually now we can build up a logic circuit I am going to do it with a toggle flip-flop you can actually use anything else. So since there are three states I require at least two flip-flops so that I can each flip-flop flip can possibly represent two states, so I need to have for at least two represent these three ones the fourth state I will not be using okay.

So I will be using two flip-flops I will draw the diagram picture here itself, I call it T1, I call it T0, Q0 and Q1 these are the outputs in fact there will be inverted output also which will be coming this can be used by us. And of course, I will take these state outputs and based on that I will decide whether I will be, I am going to actually activate the cross point or not. So cross point is structure something like this I had drawn it earlier also.

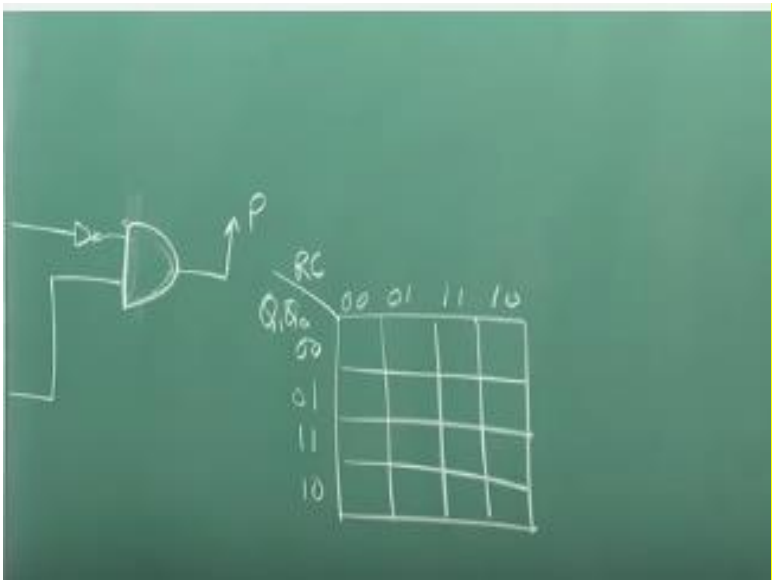
This is one, so I will, I am actually going to take and short these two and this is my activation I call it signal p, whenever P is high this cross point is active, when P is low it is not going to be active. So I am going to generate a p from here and the way I can represent it is that this

intermediate state is represented as 10 ON is 01 you can do it other way around then the whole computation actually will change.

And I can put it as 00 which is the OFF state there is no 11 state of course in this case so that will never be happening in this system okay. So I have to generate this p, so p should be only activated when you are in ON state which actually means your Q1 has to be 0 and Q0 has to be 1. So only in that case this is what is going to be the controlling the P so this will only be one when Q1 is 0 and Q0 is 1 rest all the time it will be 0.

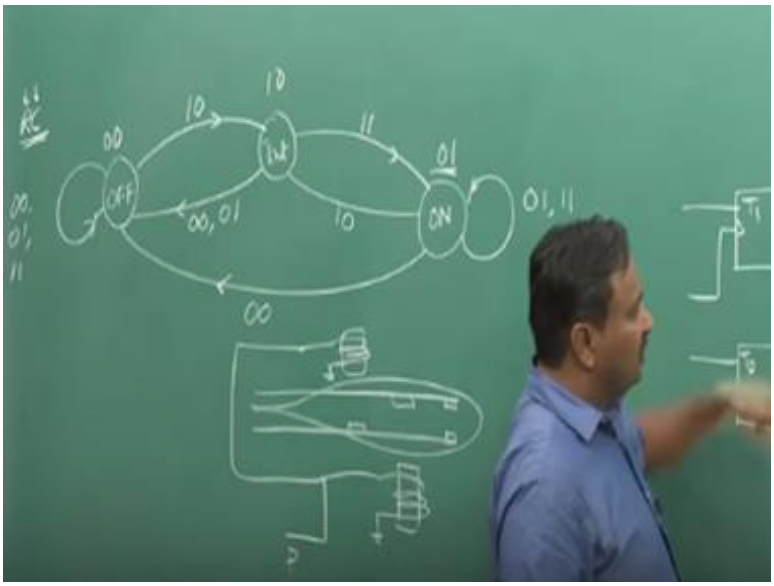
And that is what I want here, so it is if you are in ON state this will remain ON otherwise this cross point will not be ON if you are not in the ON state. So all other states are immaterial, so where 001011 does not matter of course 11 will never be happening. And of course, now I can actually build up I need to ask just estimate what will be toggle values depending on what is the current state of these flip-flops and what is my input.

(Refer Slide Time: 06:19)



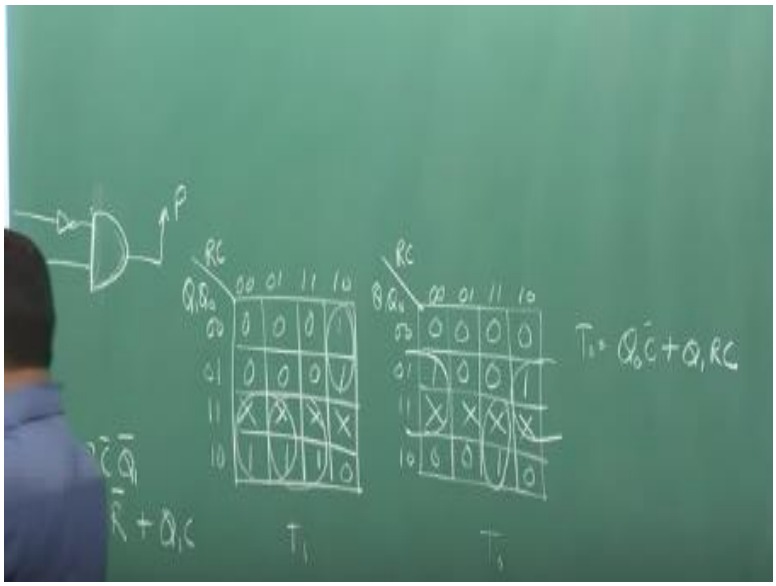
So I can actually draw a karnaugh map for this it is actually elementary, but let us do it. So I am assuming that you know that how the karnaugh map operates.

(Refer Slide Time: 06:44)



So let us see if it is you are in 00 a state so Q1 and Q0 both are 00 that 0 state means you are in OFF. So if your RC is 00 you come back to OFF state okay, from OFF you come back to the OFF state, so what does it mean, that from OFF you are coming back to OFF state so you are in 00 state you are in 00 state so there should not be any toggling.

(Refer Slide Time: 07:20)



So all toggle T1 and T0 both have to be 00. So I require two of them, so one is for P1 other one is P0, so I have to put a 0 here a 0 because there is no toggling required okay, when it is 01 then also no toggling is required. So here I have to put a 0, when it is 11 no toggling is required. So I have to put a 0 here okay, and when it is 10 then it should go from OFF to intermediate state okay, so Q1 has to be toggled. So it is 00 RC is 10.

So it has to go from 00 to 10 state which means this has to be toggled and this should not be toggled okay. So once you are this basically means I have taken care of when you were in 00 state, so what will be the next state that I have decided. There is no 11 state so that becomes a do not care condition. So I can use it for optimizing by basically logic circuit here, the community or logic which I will be using here as a input to generate inputs to these flip-flops.

So if you are in 01 state in that case sorry, you are in 10 state which is intermediate one. So if you are in 10 state you get a 00 then what is going to happen. So I am talking about this particular row, so 10 with a 00 you have to go to OFF state which actually means only the Q1 has to be toggled. So this will be toggled and this will not be toggled okay. And if you are going

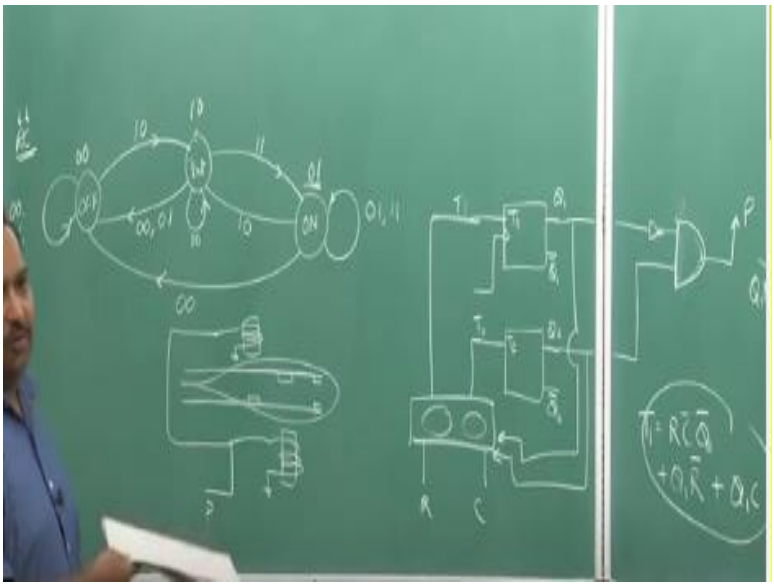
to have a 01 as a input in 10 state then also the same result so I am going to put a 1 here and a 0 here okay.

Now if you put a 10 is being inputted then what is going to happen, so if you are in the same state put 10 you go back to the same one, so no toggling will be required so it has to be 00. And if it is 11 you go from 10 to 01 so both of them have to be toggled. So I will put a 1 here and a 1 here. Now if you are in ON state what is going to happen, you are in state 01 if your RC is going to be 00 okay, basically these so it goes from 01 state to 00.

So Q1 is not toggled, but Q0 is toggled, so Q0 is toggled Q1 is not toggled. Then if you are in get a 01 you remain in same state so no toggling, so both of them will remain 00 when you get a 10 then you go to 10 state, so both of them have to be toggled. So this is what is going to happen. And when you have 11 you remain in the same state so no toggling. So this is what will be your karnaugh map.

And of course, now you can do the optimization. So in this case you will have this will do three terms and of course, if you solve for this you should get P1 = RCQ bar plus Q1R bar this actually term corresponds to this value. And then you will have Q1C, so there are one two and three terms so T1 will be given by this and similarly, from here I can find out what will be T0. So this requires only two terms the one is combination of this another one is this combination. So this should be Q0C bar plus Q1RC.

(Refer Slide Time: 12:31)



So essentially you will take Q1 Q0 you will take R and C row and column controls, you take these as the inputs and you will take this as a input and you will implement the logics which are here and will generate T1 and you are also generate T0. So these the two logics will be implemented these ones will be implemented in this box. And you get your controller for a cross point okay, and that is how your crossbar is going to be implemented.

Acknowledgement

Ministry of Human Resources & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL, IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapobrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

Sanjay Mishra

Shubham Rawat

Shikha Gupta

K. K. Mishra

Aradhana Singh

Sweta

Ashutosh Gairola

Dilip Katiyar

Sharwan

Hari Ram

Bhadra Rao

Puneet Kumar Bajpai

Lalty Dutta

Ajay Kanaujia

Shivendra Kumar Tiwari

an IIT Kanpur Production