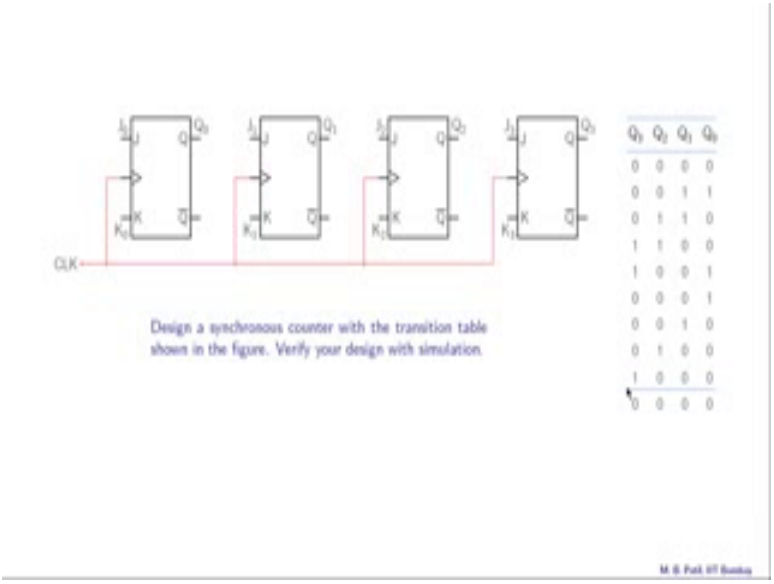
Basic Electronics
Prof. Mahesh Patil
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture - 68
Simulation of a synchronous counter

Welcome back to Basic Electronics. We have seen how to design a synchronous counter to satisfy the given state transition table. In this session we will follow that procedure to design a counter. We will also simulate the counter and verify that it does indeed follow the state transition table we started with. Let us get started.

(Refer Slide Time: 00:43)



Let us consider this design problem. Design in a synchronous counter with the transition table shown in the figure. And verify your design with simulation, so here is the transition table. We have Q 3 Q 2 Q 1 Q 0 equal to 0 0 0 0 in the first state, then 0 0 1 1 0 1 1 0 etcetera. And after 9 states we are back to 0 0 0 0 so it is a mod 9 counter. And this is how the counter can be implemented, we have 4 flip-flops here the outputs are Q 3 Q 2 Q 1 Q 0. Our first job is to design this counter; that means, we need to figure out what J 0 should be what K 0 should be what J 1 should be etcetera, in terms of Q 0 Q 1 Q 2 Q 3 so that the counter follows this transition table. And our next job is to simulate this counter and make sure that it does indeed follow this transition table.

(Refer Slide Time: 01:53)

So, let us begin with our design process. This is the transition table that the counter must follow. And here is a table that we have seen earlier which can be used in the design. If you know Q n and Q n plus 1 this table tells us what J and K should be. And note that we have the positive edge of the clock indicated here. Because we are going to use positive edge triggered J K flip-flops. What we need to do next is to obtain kernel maps for all of these J 0, K 0, J 1, K 1 and so on based on the transition table that the counter must follow all right.

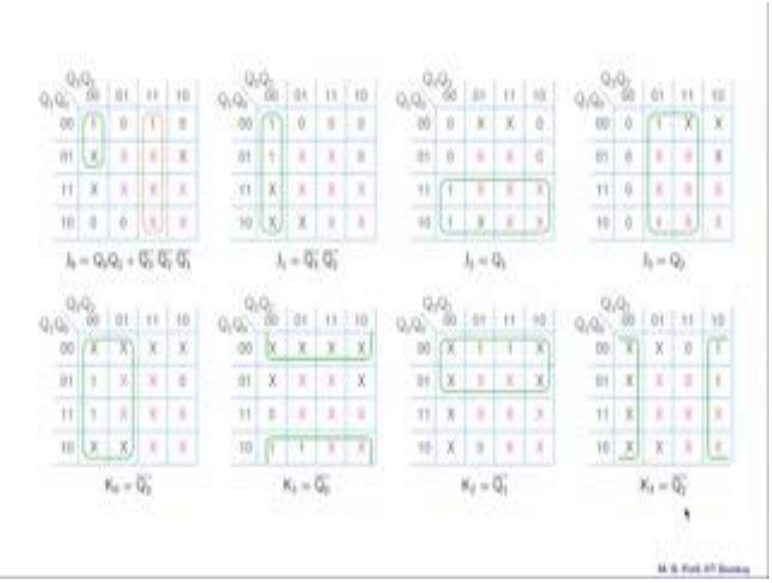
Let us begin with the do not care conditions; that mean the conditions which are not listed among these states. And those can be marked right away as Xs. For example, this state Q 3 Q 2 equal to 0 1 Q 1 Q 0 equal to 0 1 that is 0 1 0 1 is not listed in the transition table, and therefore, we do not care whether that is 0 or 1 so therefore, it is marked as X.

Now let us consider the first state; that means, 0 0 0 0. Now after the clock edge the active clock edge our Q 3 must go from 0 to 0 Q 2 must go from 0 to 0, Q 1 must go from 0 to 1 and Q 0 must also go from 0 to 1. So let us look up the required J and K values for this transition 0 to 0, that princes here. And that gives us J equal to 0 K equal to X. So for a J 3 and K 3 we must have J equal to 0, and K equal to X. And the same thing is true also about J 2 and K 2 because Q 2 also goes to the same transition.

What about Q 1? Q 1 goes from 0 to 1, so let us look up that entry 0 to 1 J must be 1, and K must be X and that is what we have over here J 1 is 1, K 1 is X and the same is true

also about Q 0. Q 0 also goes from 0 to 1, so therefore, we have J 0 equal to 1 and K 0 equal to X. And we can follow this procedure and complete the rest of the table and that is what we will get. And you are definitely encouraging to do this yourself and verify that your entries are the same as what we have shown over here.

(Refer Slide Time: 04:56)



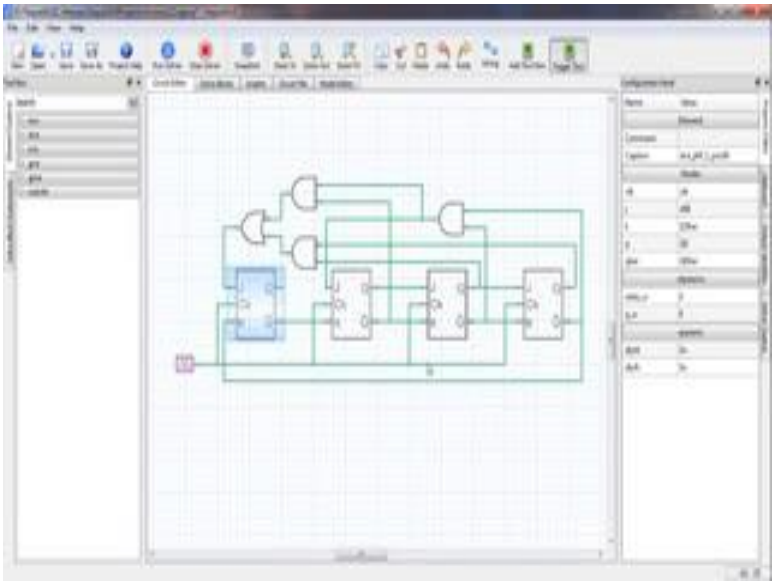
Our final step is to obtain expressions for J 0, K 0, J 1, K 1 and so on. And we can obtain minimal expressions using the K-maps that we constructed in the last slide which are reproduced over here. For example, J 0 is given by Q 3 Q 2 corresponding to this group here plus Q 3 bar Q 2 bar Q 1 bar corresponding to this group over here. Notice that we have 2 different types of Xs. Here the ones which are marked in the pink color came from states which are missing in the transition table, whereas, these others came from our table for J and K given Q n and Q n plus 1, but they are all do not care conditions and we can choose each one of them to be 0 or 1 so as to minimize the expression for J 0 or K 0 etcetera.

In the map for J 0, we have taken this X to be 0 0 0 0 0 0. We have taken this X to be 1 1 1 1. So that we get the largest possible groups containing all 1’s.

Now, you can verify that all these other expressions are correct and they correspond to the maps given over here. So we have got K 0 equal to Q 3 bar J 1 equal to Q 3 bar Q 2 bar K 1 equal to Q 0 bar J 2 equal to Q 1 K 2 equal to Q 1 bar J 3 equal to Q 2 and K 3 equal to Q 2 bar. We can now proceed to construct this counter with a circuit simulator

and then simulate to check whether the counter does indeed follow the transition table that we set out to achieve.

(Refer Slide Time: 06:58)



Let us now start constructing the schematic for the counter. First let us get the components and we will begin with the J K flip-flop. There is an element called J K FF 1 pos for positive edge triggered flip-flop that is the one. We then need to get and gates we can just get one right now, and then copy it later, or we also need a clock element that will serve as our clock.

Let us look at the properties of the J K flip-flop. It has got 2 real parameters. One is called delay high to low and the other one is called delay low to high. This parameter describes the delay involved when the output goes from high to low. And this 1 describes the delay involved when the output goes from low to high. Now these default values are 5 nanoseconds for each of them. And these are appropriate for our simulation because we are going to choose a clock period which is much larger than these values. So we will leave these properties at their default values. Similarly, the and or gate also have these parameters and they are both set to 5 nanoseconds. And we can leave those also at their default values. Let us look at the clock now.

Clock has got 2 parameters, t 1 and t 2, which are relevant in this simulation. T 1 is the interval for the first part of the clock, in which it is 0. We will have said that as 40 microseconds. And t 2 is the interval in which the clock is high; we will have said that as

10 microseconds apart from these elements we will also find the connector element convenient. So, let us bring that in. That is just a dummy element with no physical properties and it can be used for wiring all right. Our next step is to reproduce the flip-flop element, 3 more times so we do that by control c control v like that.

Let us now place the components and start our wiring, and let us get rid of this tool box. Since we are not going to require any more components and also the configuration panel, and let us copy this connector, so that we can connect the clock to all of these. Now let us bring the clock over here and do that wiring.

Since it is a synchronous counter, the clock must go to all these flip-flops like that. Next let us look at the connections which are required. These are the connections that we figured out earlier, and you notice that many of these connections do not require any gates. For example, K 1 is Q 0 bar now this is our Q 0 this is our Q 1 Q 2 Q 3. This is our K 1 and therefore, Q 0 bar can simply be connected to K directly like that. So it does not require or gate. The only gates you require are these. For J 0 and for J 1 so we require one and gate over there and some more gates over. There so let us figure that out now. Note that this Q 3 bar Q 2 bar also appears in the expression for J 0; that means, what we have here is J 1 ended with Q 1 bar. So we have one and gate there, one AND gate there, and one OR gate, and here we have one and gate. So we require 3 and gates and one or gate.

So, let us copy this like that. And our J 0 is going to be here, so maybe what we should do is to turn this around, because then this output can be directly connected to J 0. What we will do now is we will place these gates and then we will start the wiring. So this or gate gives us J 0, and this 1 can be 1 of these and operations. And this and gate can be used for Q 3 bar Q 2 bar so in fact, we can complete that wiring right away. This is our Q 2 bar and that is our Q 3 bar, like that. Let us use this gate to obtain Q 3 Q 2. Now this is our Q 2 that is our Q 3, so let us connect those 2. That is Q 2 and that is Q 3 like that. And the output of this gate should go to this or gate this 1 like that.

One question that comes to mind is whether there is a cross over here, or whether these wires are actually connected. The way to check that is to click on the wire, and that shows the complete path. So in this case this wire is simply crossing over this other one. It is a good practice to use this connector to indicate connections. So for example, if

there was actually a connection over there, then we could have placed the connector over there, and used that for our wiring.

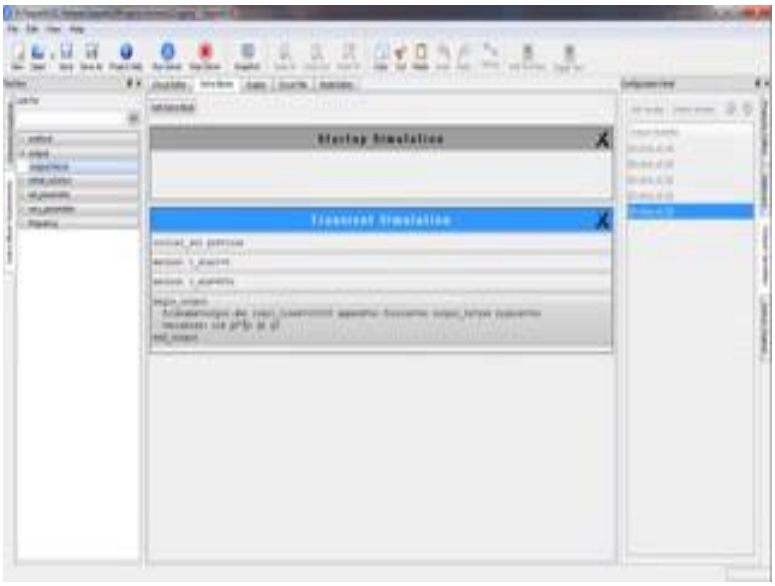
Let us complete the rest of the wiring now. The output of this gate must go to the or gate like that, and what is this gate giving us, right here, Q 3 bar Q 2 bar ended with Q 1 bar. Now this term is already here that is J 1. And we have already used one and gate to give us that. So let us make that connection and that needs to be ended with Q 1 bar. This is our Q 1 bar, so let us connect that to the other input like that. This is our J 0, so let us connect that here. What about K 0? K 0 is the same as Q 3 bar this is Q 3 bar so we will connect that to K 0 like that. What about J 1? J 1 is Q 3 bar Q 2 bar which is this output here of the AND gate. What about K 1? K 1 is the same as Q 0 bar that is our Q 0 bar. What about J 2? Same as Q 1 that is our Q 1 that is J 2. K 2 is Q 1 bar, J 3 is Q 2 that is our Q 2 and K 3 is Q 2 bar all right.

So, now our connections are complete. And we can move forward and name nodes and define output variables and so on all right. So let us make this full screen and bring back our tool box as well as the configuration panel like that. Let us name the nodes first starting with the clock; we can call this c l K. This is our Q 0, so the Q of this flip-flop is Q 0, and we will call this Q bar s Q 0 bar. Similarly, that is our Q 1 and that is Q 1 bar and so on.

Let us now specify our output variables. And since we are going to select more than one, we can click on add variable with the control key pressed. And that allows selection of multiple output variables. So we will select the clock Q 0 Q 1 Q 2 and Q 3. And we must remember to click on this add variable button once again. So that the add variable mode is terminated.

Let us change these default output variable names to some meaningful names all right. Now we have completed the circuit editor part of the project and now we will move on to the solve blocks. One important point and that is we want our counter to start in the state 0 0 0 0; that means, we want the output of each of the flip-flops to be 0, and that can be controlled using something called startup simulation. And whether a flip-flop output would be 0 or 1 is determined by the startup parameter called qsv. And by default it is 0 so therefore, if we perform startup simulation with the default settings our counter will automatically start in the state 0 0 0 0, all right.

(Refer Slide Time: 20:41)

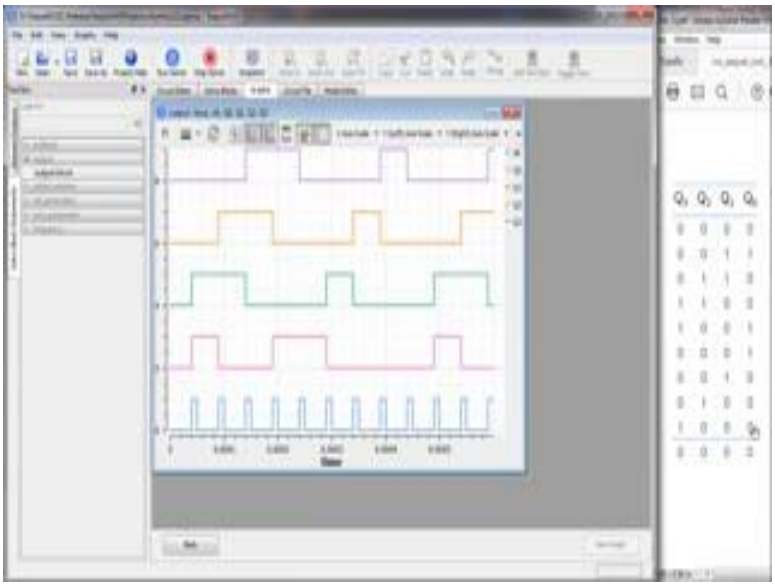


Let us now proceed with the solve blocks, let us add a solve block, and change it is type to startup that will set the initial conditions for the flip-flops. And now we will follow this up with a transient simulation solve block like that. Now in digital circuits this backward Euler method is not relevant we can remove that. Similarly, this delta t the simulation time step is not relevant when we have digital simulation, so let us remove that as well. And we need to specify the end of the simulation time. Now our counter is a mod 9 counter, and one clock period is 50 microseconds. So we should simulate for at least 9 clock cycles that is for 50 microseconds.

Let us say we simulate for little more than that 600 microseconds. We need to specify the initial solution or the method to obtain the initial solution. And that can be done using this statement here initial solve. And instead of initialize we should use the option previous here, because we want the output of this block to be used as the previous solution for the transient simulation block all right.

Next we want to add the output block. So let us add the output variables of interest namely clock Q 0, Q 1, Q 2, Q 3, all right. And we are now ready to run the simulation.

(Refer Slide Time: 22:39)



So, now the simulation has been completed and this output file has been created. Let us choose time as the X axis and let us look at all of these variables clock Q 0, Q 1, Q 2, Q 3. So that is the plot. This waveform here is the clock this one, is Q 0 so clock Q 0, Q 1, Q 2, Q 3. Let us now check whether our counter does follow the transition table which we intended to obtain. The first state is Q 3, Q 2, Q 1, Q 0 is 0 0 0 0, that is Q 3 so 0 0 0 0 the next state is 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 then we have 1 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0, and then back to 0 0 0 0. So our counter seems to work as desired and the design process is therefore, validated, all right.

Now, as a small extension of this particular exercise, what you can do is design a mod 10 counter by adding one more state after say this state, and that state could be 1 1 1 1 for example, so then you can go through this entire process of coming up with J 0 K 0 J 1 K 1 etcetera, and then simulate the circuit and check whether your design is correct.

In summary, we started with a given state transition table and designed a synchronous counter, which will satisfy the given sequence. We then made up the circuit schematic from scratch simulated the counter and verified that it follows the sequence we designed for. This is a simple circuit in terms of connections to be made and therefore, it is a good idea to try it out on a breadboard. That is all for now see you next time.