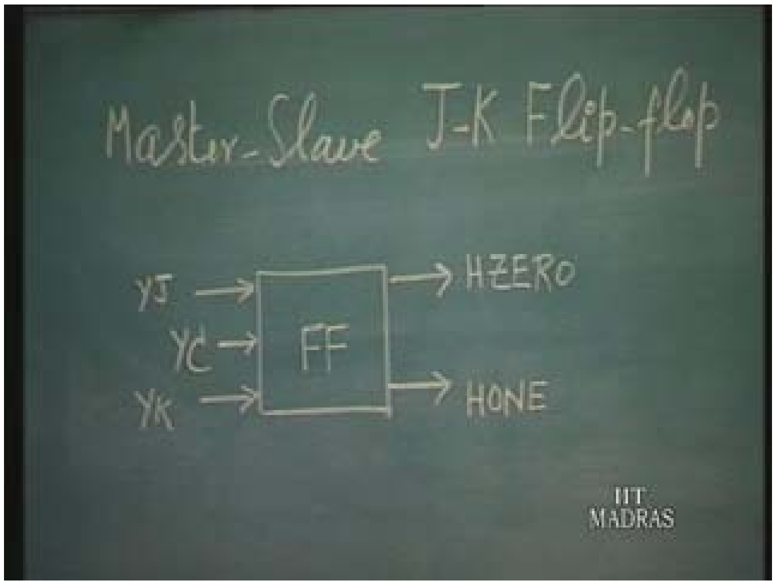
Computer Organization
Part – I
Prof. S. Raman
Department of Computer Science & Engineering
Indian Institute of Technology
Lecture – 8
State Machine Design

Towards the end of the previous lecture we considered the design of JK flip-flop, specifically master–slave JK flip-flop; we just started that. Why did we take up this particular design? Mainly because this particular design will involve a few states – a couple of inputs and a couple of outputs to deal with, and hence only a few states to deal with, and the whole design can be explained both from the hardwired control point of view as well as from the micro-program control point of view. Otherwise, if you just start discussing the CPU design, there are too many states and too many issues, too many inputs and outputs to be considered. So as an example, we are considering this. So bear this in mind.

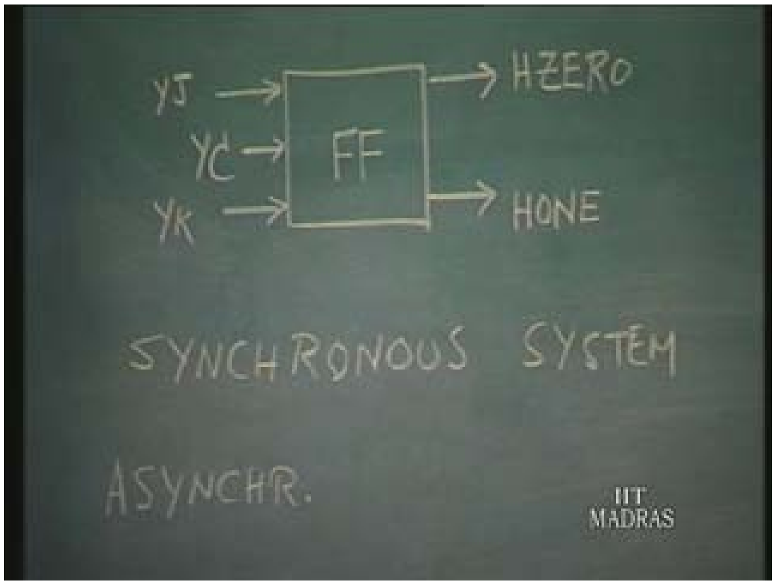
(Refer Slide Time 00:02:07)



In fact we had worked out a table, called a truth table or excitation table, specifically with reference to flip-flop, with J and K as the inputs, and the present state and the next state. In any state we know that a system develops an output. But though we talk about states, bear in mind that essentially in any system design, we are concerned with the inputs to be handled and the outputs to be generated. We may call the state the system goes through as a necessary evil. So state is only a means to achieve precisely the ends we want. We are not really bothered about the state, but the digital system goes through states in accepting the inputs and in generating the outputs. So really speaking, the input and the output – the relationship between these – is what precisely we are interested in and states come in-between.

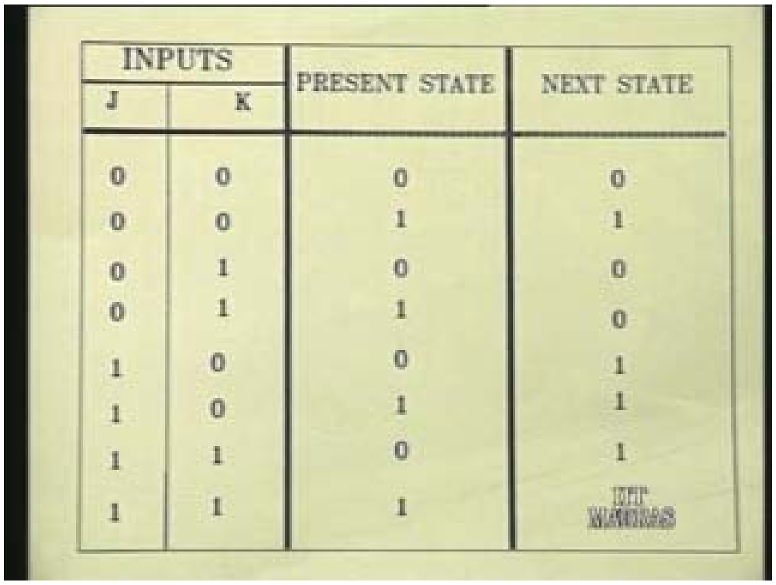
Now we said in the previous lecture itself that the inputs can be called YJ, YK and YC; that is, J and K as the input and C is the clock input, because in any system if you have a clock, we can synchronize the generation of the output with reference to that clock; that is why we have a clock. And that is one reason a system having a clock will be called a synchronous system. It is also possible, instead of a synchronous flip-flop, to have the other one, called asynchronous flip-flop. Now in a system, though what we are talking about is just this component flip-flop, one can also have an asynchronous one. In that, if you have that asynchronous system, the clock will be absent. What it means is that as soon as the inputs change, the output will change in asynchronous system.

(Refer Slide Time 00:04:22)



In the asynchronous system, as soon as the inputs change the output will change; internally there may be state changes, whereas in the synchronous system just because the inputs change, the output may not change. It will be synchronized with reference to the clock. While discussing this flip-flop and its behavior we had not considered the clock yesterday. Now let us go to the chart and review that particular excitation table.

(Refer Slide Time 00:05:03)



INPUTS		PRESENT STATE	NEXT STATE
J	K		
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	

You must be familiar with this JK flip-flop and specifically I am going to talk a little more for those who have not yet been initiated into switching theory and logic circuits. For those who have done a course on switching theory and logic circuits, this may be a repetition. You can see these are pairs 00; that is one pair; then you have a 01 pair; then a 10 pair and a 11 pair. Now because we are considering these for a given present state, what is the next state? So far, in one the state may be 0 and we are considering the second case state as 1; that is why we are having the pairs of inputs here.

That is, for a given input J and K, let us say 0 and 0 are here; for the present state 0, what is the next state, and for the present state 1, what is the next state? That is what we are saying here. As long as K is 1, that is, in this particular pair 01, J is 0; this is as long as K is 1, we find that whatever may be the present state, the next state is 0. That is why I said K can be considered as a reset input, which means it causes the output to go 0; that is, the next state to go 0. Similarly, look at this pair – J is 1 in both these. Now whatever may be the present state, the next state is 1; so we say J is equivalent to set input. Now what about this pair? In this J and K both are 1 and we have already said that whenever both J and K are 1, it does not mean it sets as well as resets; basically it means it toggles. Whatever be the present state, it just goes to the other state. If it were 0, the next state will be 1. If the present state is 1, the next state will be 0. So this in fact is a toggle input.

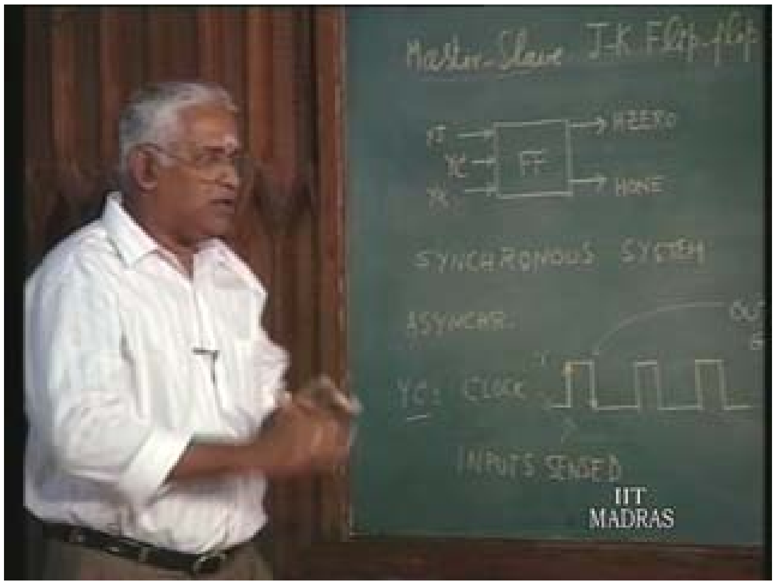
So now you can see that if J is 0, when K is 1, the next state will be 0 if K is 0; when J is 1 the next state will be 1, and if both J and K are 1, then the next state will be a complement of the present state. If it is 0, the next state will be 1; and if it is 1, the next state will be 0. You may notice one more thing here – the K input – look at this K input here – may be I will explain this further as we proceed along. This excitation table actually gives the relationship between the present state and the next state for a given input. In other words, what we say is how this flip-flop gets excited by its inputs to assume the new next state; that is why it is called excitation table.

But in our case, we not only have these two inputs J and K, but we also have the clock. We have not yet considered that. We have to consider this clock also. Now if I include the clock in the particular excitation table, then the table will become unwieldy. So I will just explain the reason for that. The two inputs, J and K, and the present state – these three variables are considered; that is why we had eight states in that particular chart – take a look at the chart – there are eight states. That is because J and K and the present state – these three variables are considered; so 2^3 is 8. Now if you include the YC clock also, it will become 2^4 or 16. So the size of the table will double; and it is not really necessary to take a look at it. There is one other thing; we have at least talked about the JK part, but we have not yet talked about the master–slave part; that particular thing is linked with the clock. What is that? It is like this – the clock is the one in the processor which will be defining the state and so, really speaking, a clock is a state defining input, whereas you can consider in this case J and K as possibly data inputs. The clock is in fact the state defining input. Now in a master–slave flip-flop, this is what is happening. In the rising edge, that is, this particular one the rising edge of the clock the inputs, namely J and K will be sensed and internally the flip-flop will go through some state change. It will not immediately affect the output and in the falling edge of that clock, the output will be generated. In other words, at the rising edge the inputs are sensed, and on the falling edge of the clock, output is generated – more than one output is generated. This in fact is the principle of the master–slave.

What is the idea behind this? It basically says whatever may be the state of input after this edge it is not going to affect the output and subsequently the behavior of the flip-flop. This is going to be useful especially in the case of some noisy situation, where there may be some noise on the inputs J and K. By the time the falling edge comes, the state of the input may have changed. It can affect the behavior of the system. We will not go too much into it; it is in fact the electrical engineering conservation outputting. From computer science point of view, it is the logic we will take a look at.

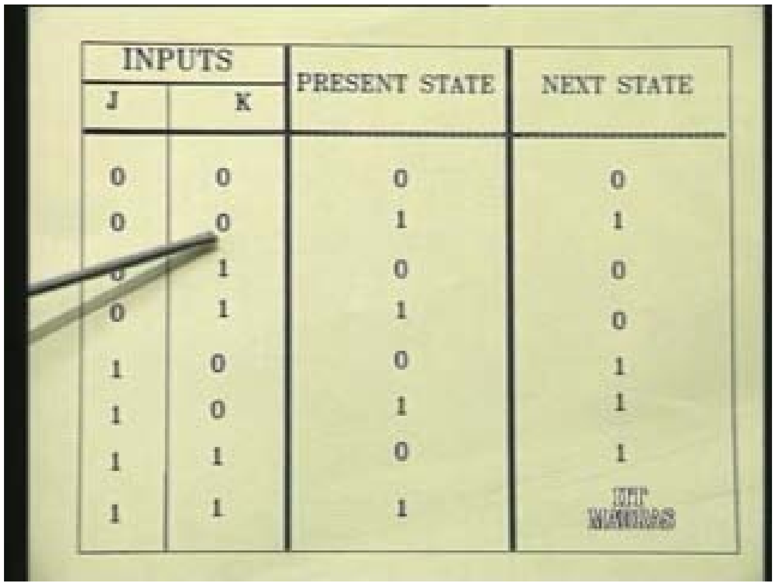
So the inputs are sensed during the rising edge of the clock. What I mean is that when the clock YC becomes 1 – that is let us see it this way logic 0 and this is logic 1 – what is the input? That is sensed and when the clock falls back to 0, okay the clock falls back to 0; that is the 1 level this is the 0 level. The two logic levels are here, whatever the voltage or current levels may be. So the outputs will be generated when the falling edge inputs are sensed during the rising edge. It can be the other way also, meaning the inputs can be sensed during falling edge; it doesn't matter what it is, in which case we will be considering this as the clock duration that is solved. Instead of this, we will be considering the other one – it hardly matters what it is. What exactly is the principle of master–slave? [14:13] It says during the rising edge of the clock, sense the inputs and prepare the system's outputs. It only says prepare the system's outputs; it does not change the output immediately, and actually affects that change in the output during the falling edge – this is the master–slave principle. We will keep referring to the chart as and when required. Whatever I have said about J and K and about the clock, which is YC, is the same as this.

(Refer Slide Time 00:14:54)



Instead of drawing another table with 16 entries, let us straightaway create the ASM chart for this particular flip-flop, because ASM chart is going to give me the whole behavior of the system. What is the system here? It is master–slave JK flip-flop. We know that the system goes through states. I will arbitrarily talk of starting state, let us call that state a, and it is quite arbitrary. In state a, I will just say I assume that output generated is H ZERO. Remember we were always talking about the state box and we were recording the output being generated in that box. In state a, output generated is 0; now we shall take a look at the chart again and see what we are talking about here.

(Refer Slide Time 00:19:21)

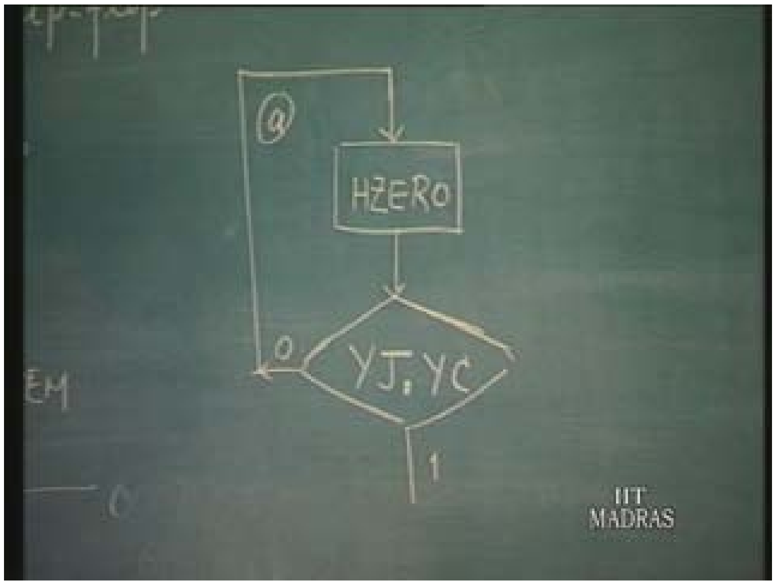


INPUTS		PRESENT STATE	NEXT STATE
J	K		
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	

Now let us say you are somewhere here; that is, the present state is 0; the output generated is 0. In the present state 0, the output generated is H ZERO. Now when the present state is 0, these are all the various possible things. Take a look – if J and K both are 0, the next state is going to be 0; there is going to be no change. Take the next case – when the present state is 0, if you have J as 0 K as 1, what is the next state? Anyway, K is a reset input; so the next state continues to be 0; there is no change. In other words, as long as J is 0 – we are talking about these two cases – as long as J is 0, the state of K does not matter; it does not do anything. Now coming to this, that is when the present state is 0, what is the state for that? It is 1, and for this 1, the present state is 0 and the next state is 1. Take a look at the J and K levels for these two present states 0. When J is 1, as long as J is 1, that is, here and here, irrespective of whether K is 0 or 1, the present state changes to 1. So actually, it does not matter what the value of K is; it only depends on J. Let us combine these; let us get back to the ASM chart and combine that with the information I have given and include it as the input condition.

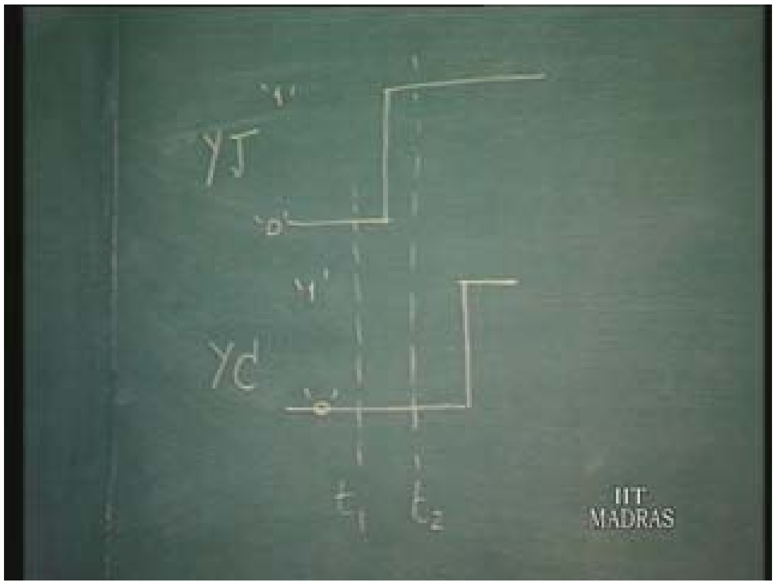
In this state a, in which 0 is the output, I am assuming H ZERO is the output, which is generated in present state 0, whatever I had given in the chart earlier. So in this, it only matters what the value of J is; K need not be checked and now combine that with whatever I had said earlier about the clock. If the clock were 0, there is nothing, so as long as J is 0 and the clock is also 0 – that is the symbol for it – it will continue to be in state 0, that is, generate output H ZERO, because K does not matter, it only depends on J. As long as J is 0, YJ and YC will also be 0; now we will also take YC be 0. When will this combination be 0? It means YJ and YC 0 really means either J or C or both are 0. For any one of them being 0, this will be 0. So as long as the clock is in 0 levels or as long as YJ is 0, both will remain 0: this is the condition.

(Refer Slide Time 00:20:59)



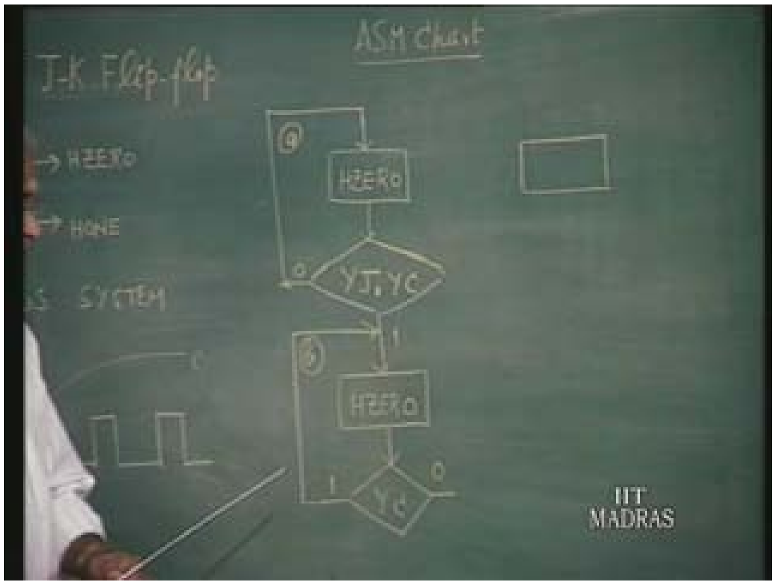
For the present state a, in which I have arbitrarily assumed the output generated is H ZERO, which is equivalent to my assuming the present state of the flip-flop being 0 as per the chart, as long as YJ is 0, it is fine; it will remain. What happens when YJ and YC become 1? Actually we have not talked about this kind of a combination earlier. This is a logical and combination of two inputs. This means both YJ and YC are 1. So suppose this is the 0 level and this is the 1 level, what we are talking about is this. See here – this is 0 level this is the 1 level; now you just take a specific instant here YJ and YC both are 0. So that corresponds to the ASM chart to this now; take this instant. In this instant, let me call these t_1 and t_2.

(Refer Slide Time 00:22:07)



During t_1, both YC and YJ are 0; during t_2, YJ is 1 and YC is still 0. Then again this holds good. Now take this instant, t_3. I am saying YJ and YC both are 1. So for instance this is 1; YJ and YC both are 1. What did we say earlier? When it is 1, the input alone is sensed; the output should not change. The output will change only for the falling edge. So there is an internal change in the state, but output will have to be continuously be generated as H ZERO itself; that is, there is a state, let us say state change from a, and let us call this as b. There is a state change from a to b, but the output continues to be H ZERO. That is mainly because of what we had said earlier. Now the inputs are sensed only in the raising edge. So now YJ input need not be checked; YC is all that has to be checked. Next, mainly because as long as YC is 1 – we do not know about YJ, we will come to that later and fill in – this will continue to be in the same state, because the output should change only when YC falls to 0.

(Refer Slide Time 00:24:08)

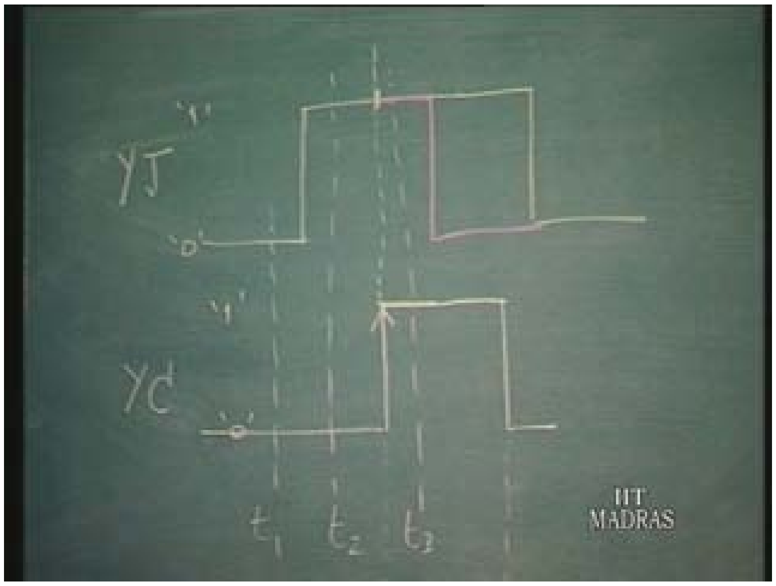


When YC falls to 0, that is, when YC becomes 0, then we say there is a state change and now there will be an output change also, mainly because we had said when YJ becomes 1, the present state 0 should become the next state 1. Let us go back to the chart and confirm that again. See the present state is 0, and the next state is 0 in this; present state is 0 and the next state is 0 as in this case. The present state is 1, the next state is 1 only for J being 1; similarly here too present state is 0 and the next state is 1 only for J being 1. We have considered the case of J being 1; that particular one is YJ. So we had come from this state to this state mainly because YJ was 1; so you should cause a change in the output but not until the clock falls from 1 to 0; that is why there is a delay. Now when it is 0, that is, when the clock is 0, there is a change in state to c; from b to c and now the output that is generated is H ONE, which corresponds to – with reference to the chart – the next state being 1. Just see here – J is 1; the next state is 1. Actually, we are constantly from present state 0 to 1. So all the 0 to 1 transition goes to this; that means basically when the clock falls, then at this instant, we are saying at this instant, the output that was earlier 0, now becomes 1.

Suppose I have another waveform for indicating the H ZERO and H ONE – it will look somewhat like this. That is, until this time, H ZERO was generated at this instant; H ZERO goes and H ONE is generated. Pay some attention to this because we are talking about H ONE, H ZERO and we are also talking about level 1 and level 0. Basically it is nothing but a complement of the other one: 0 as complement of 1; and here also we find the waveforms as complement. You can work it out. I said during this transition the input was sensed; so internally, at this instant itself, there was a change in the state. But the output, as far as the output is concerned, there was no change. It continued to be generated as H ZERO. Now you may even say that this duration corresponds to state a; up to this from here to here it corresponds to state b.

What about YJ? Suppose it continues and falls, suppose after this instant, when actually inputs would have been sensed, it had gone but fallen here, now there are two situations – YJ continues for some more time, whereas here, YJ falls.

(Refer Slide Time 00:29:07)



Now the change in the input really does not affect; that in fact is the principle of the master–slave component; that is the main thing. So the inputs are sensed at a particular instant and outputs are generated at a different instant. But of course, the instant is related by the clock; it is not arbitrarily different.

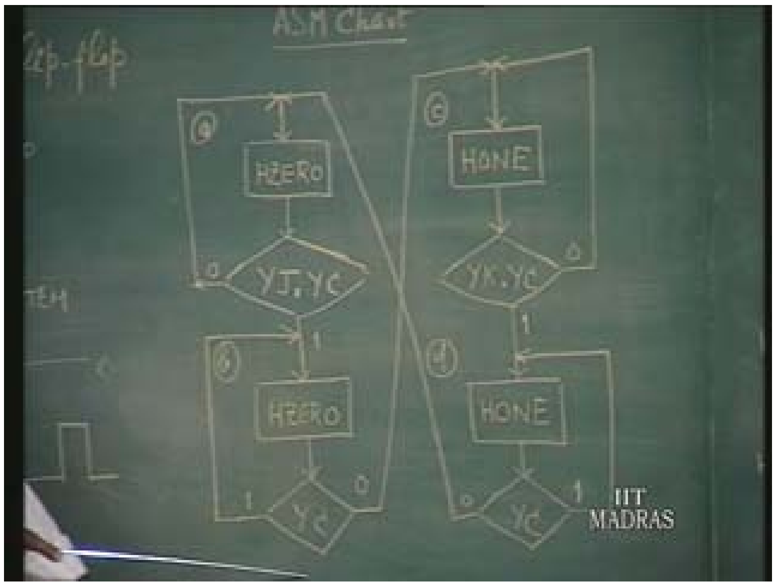
So whether J had changed as shown here or as shown here, does not affect the output. Now we are in a state, the present state, state c, in which the output 1 is generated. This corresponds in the chart for the present state being 1. Like we worked out for present state 0, the present state is 1. The next state is 1; in this present state is 1 and the next state is 0. You can see that there is a transition from present state 1 to next state 1, which means the state continues really. Now there is a transition from 1 to 0; you can see K is 1. Because of K being 1, there is a transition from 0 to 1. When K was 1 – as in this case – there was no transition. We know that K is reset input. Now what about this case? Again 1 to 1; so K is 0. Now compare this particular one with this.

Here we have 1 to 1 transition and here also we have 1 to 1 transition; now in both, K is 0; J is 0 in one case and 1 in the other case. So whether J is 0 or 1, that is, irrespective of J when K is 0 it continues as 1 to 1; it does not depend on J. Now see this 1 to 0 transition: K is 1, J is 0. Here again, there is 1 to 0 transition. In both cases, you can see that there is 1 to 0 transition – 1 to 0 transition here and 1 to 0 transition – in this 1 to 0 transition also K is 1. For this transition, J is 0 whereas for this transition J is 1. That means it does not depend on J. So in a 1 to 0 transition, that is, in a state when we have the output H ONE when it has to go to another state in which it has to generate the output 0, it is not going to depend on J like what we had earlier.

It is something like what we may call symmetrical. Now I can complete this ASM chart by hiding this condition, saying that YK is the only one that must be sensed and now, for master–slave principle, I also add this signal YC. As long as it is 0, the present state in which 1 is generated continues when it becomes 1, that is, when K becomes 1. Shall we check the chart again what is happening when K becomes 1? When in the present state 1, K is 1, this is 1. Now we see that it may transit from 1 to 0.

In this present state 1, K is 0; the output is still 1. On the other hand, in this present state 1, there is transition to 0 because K is 1. So 1 to 0 transitions take place again; let us see the chart – 1 to 0 transitions takes place when K is 1. So when K is 1, the state transition must take place, but as per the master–slave principle, only internal change may take place; the output should not change. Again, similar to the previous one, we have a state change from c to d, but the output will continue to be one H ONE. As before, check for the state of c, because the input K was sensed in the raising edge; now the corresponding output change will take place when it falls to 0. As long as the clock is 1, it continues in the same state generating that output. When the clock falls to 0, we have a state transition from state d to a, when the output also changes.

(Refer Slide Time 00:35:39)



Now you can see that the ASM chart describes the behavior of the entire machine. In the present state 0, in which the output is 0, J becomes 1. Ultimately it will lead to a state transition so that the output generated will be 1. But that depends on the condition of the clock; when the flip-flop is in state, it generates an output 1; depending on K, when it becomes 1, there will be change in the output to 0; again that depends on the clock. So actually combining this chart information in which only the J and K had been given, plus the clock which I had given you, we have worked out the ASM chart of this master–slave flip-flop.

Everything you find in this excitation table can be verified; you can verify that that particular thing is satisfied when you follow the various paths in the ASM chart. Instantly when you go from one state to another state, that particular thing is called a path; it is also called a link path. For instance let us work it out. For the present state a, there are two link paths – what are they? For the present state a, you have one link path, which takes it back to a, another path which takes the system to state b. In other words, we say that for the present state a, there is one state transition that is possible, which is a to a itself. That is one link path; the other link path is state transition a to b. There are two paths here; a to a is one possible transition and a to b is another transition. So there are two possible transitions in this. Whichever path you follow, it so happens that in this the output does not change; that does not matter. That is because of the master–slave principle we had introduced.

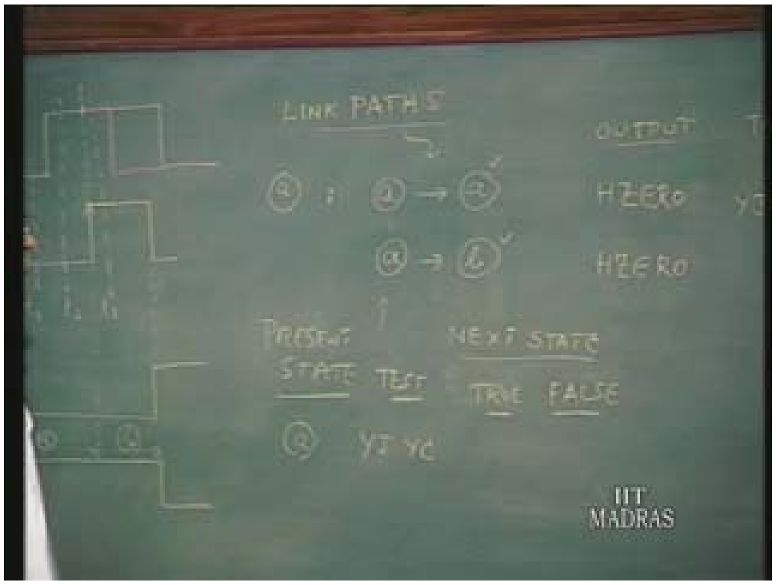
Similarly, in present state a, let us describe the behavior of the system. This is an example we took. I will completely explain now. In present state a, generate output H ZERO and look for the inputs YJ and YC; if YJ and YC are 0, there is no state transition. If YJ and YC were 1, there is a state transition from a to b. So for this present state a, there are two possible next states: in one case it is a, and in another case, it is b. The output generated in both the cases happens to be H ZERO. So here you have these link paths, which in fact actually talk about the next state. This is your present state we are talking about; from this present state, we are talking about the next state.

(Refer Slide Time 00:40:23)

As I was telling you in the previous lecture, we have a false next state and a true next state. We have arranged it that there are only two paths. It so happens that the false next state is a and the true next state is b, and that is as far as the next state is concerned. As far as the output is concerned, it is nothing much. You can see that in state a or in state b, the output that is generated is H ZERO. That is what we had written there.

There is one other thing – that is, the input or the conditions which must be tested. Given the present state a, the inputs that must be tested are the conditions of YJ and YC; these are the two inputs. Even though the system has three inputs in state a, two of the three inputs must be tested – that is what it says. K does not matter because that is how the behavior of the system is. Now I will create one micro-instruction word because that is what it is basically. That is, for a given present state a, check the conditions of YJ and YC; now because there are two next states – that is, one is a true next state and the other is a false next state – I have to record that here. So the true next state is this; false next state is this.

(Refer Slide Time 00:42:45)



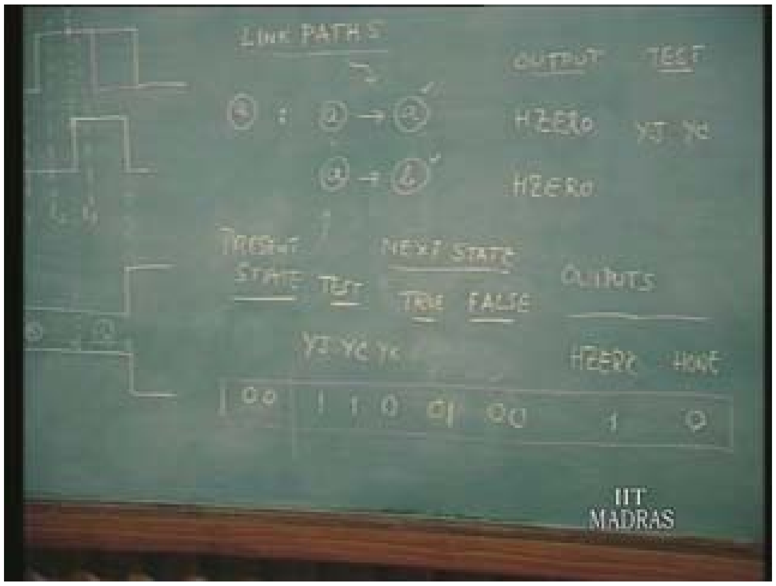
So the true next state for the present state a is b; and a false next state is a and the output that must be generated must be H ZERO. I can put the same thing in a slightly different form. That is, I can say the system has three inputs: YJ, YC and YK. I will do it for the present state a. For the present state a, the system has YJ, YC and YK, of which YJ and YC must be tested in present state a; YK need not be tested. The next state, the true next state of a is b – I am just writing the same thing down – and the false next state of the present state is a. Among the outputs that must be generated, we have two: H ZERO and H ONE, of which in present state a, H ZERO must be generated and H ONE is not to be generated. Now what did we start with? We started with a flip-flop and started saying that it has two states essentially and then, depending on the state in which it is, it generates a 0 or a 1 output.

For generating a 0 or a 1, essentially you need a two state machine: just two states. What is it that we are having? We are having a four state machine. Why? It is mainly because we have introduced the master–slave principle; because of that we have four states. Now here we have a four state machine and we have three inputs to be checked and two outputs to be generated. So with this, we will try to see what a micro-program control is and what the hardware control is because it is easy for us to discuss and deal with.

Because there are four states, I can now implement the memory element in the state machine. Remember we are talking about the state memory – go back and recall the generalized state machine model. We had a state memory, which is the one which holds all the states and here we have four states so that particular memory element must be either in a or b or in c or d. At any time, the machine is only in one of the states. So for representing four states, you need at least two memory elements, two memory cells.

If I call these memory elements a and b, these two memory elements can be either in 00 or 01or 10 or 11 states. So there are four possibilities – any specific memory element will be either in 0 or 1. So with two memory elements, we can represent the whole thing. Since this is a four state machine, we need two memory elements and I will, instead of calling them a, b and c, d, that is, using the lower case, now I will be having two memory elements and call these state variables. These are two variables, A and B, which represent the states of the machine. Arbitrarily, I will assume 00 as the code for state a and 01 for the state b; I said arbitrary – 11 for state c; and 10 for state d. So I will just arbitrarily assume there are four possibilities: 00, 01, 11, and 10. Though I said arbitrary, there is some reason for this that we will not go into at this stage. So now here I can rewrite; I can go back to this: instead of a I can write the code 00, that is, my present state. Instead of a, I am writing 00; instead of b, I will write 01. Now here we have the entire information for the present state a, the code of which is 00.

(Refer Slide Time 00:49:04)



So what it says is that in state 00, look for inputs YJ and YC – that is what this one says. Look for the inputs YJ and YC, and depending on the input test result being true, the next state is 01. If it is false, the next state is 00; and in the present state 00, generates the output H ZERO. So this word, the entire thing, is giving me information about all the link paths – there are only two link paths for the present state.

Now extend, go to state b, work out similarly, go to state c, work out similarly, go to state d, work out similarly, and we have talked about the entire system. So with four such words, we have the whole behavior of this master–slave JK flip-flop system completely defined. And when that information is given to the controller, we are actually implementing the flip-flop itself, that is, the particular element itself. So we will continue further on with this in the next lecture.