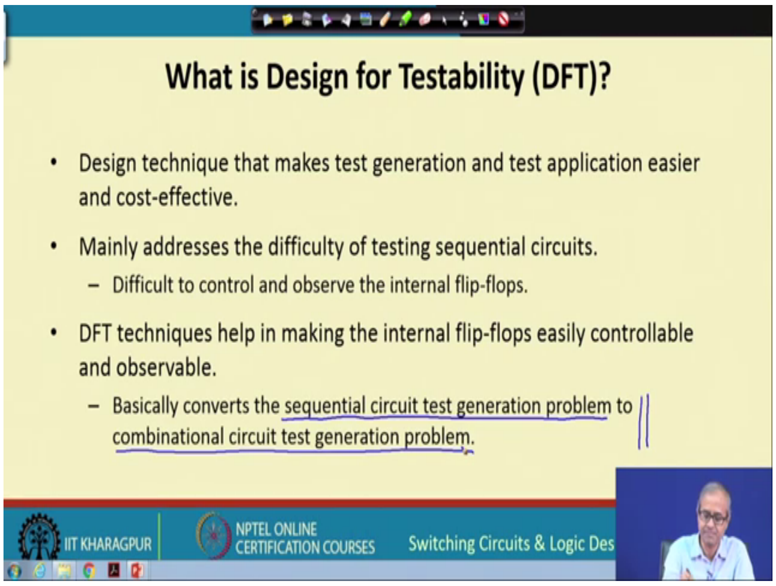
Switching Circuits and Logic Design

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 58
Design for Testability

We now talk about Design for Testability Techniques. Well, design for testability generally says some design principles of guidelines that makes the problem of testing easier. There are many such methods and techniques available but here in this lecture we shall be concentrating on one of the most widely used and popular techniques which is targeted towards the testing of sequential circuits which inherently is very difficult. So, the title of our lecture here is design for testability ok. Let us try to understand why we need this.

(Refer Slide Time: 00:58)



What is Design for Testability (DFT)?

- Design technique that makes test generation and test application easier and cost-effective.
- Mainly addresses the difficulty of testing sequential circuits.
  - Difficult to control and observe the internal flip-flops.
- DFT techniques help in making the internal flip-flops easily controllable and observable.
  - Basically converts the sequential circuit test generation problem to combinational circuit test generation problem.

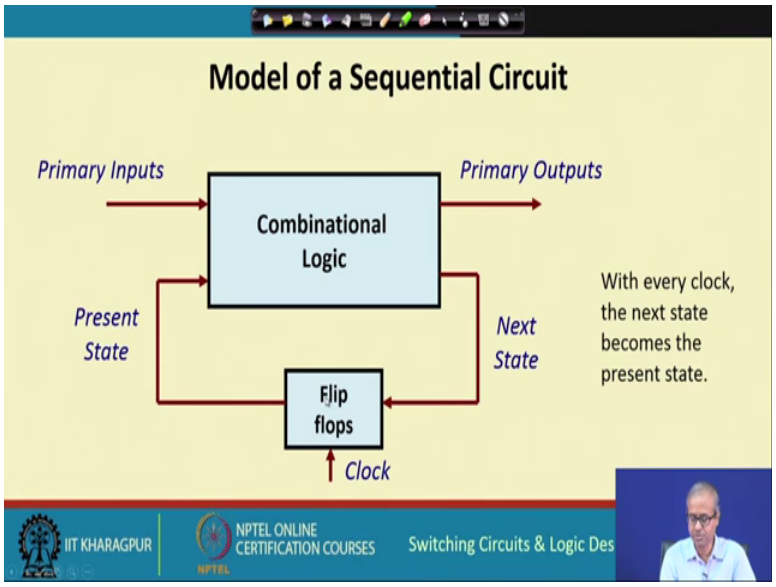
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

Design for testability or DFT in short as I said this constitutes of a set of design techniques or rules that makes both test generation and test application easier and also cost effective. So, as I said that here we primarily target the problem of testing sequential circuits because, I mentioned earlier that in a sequential circuit, because of the presence of the internal state variables the flip-flops, the number of possible input combinations can be even larger than combination circuits not only the primary inputs but also the state variables.

You have to consider all possible combinations of them there can be so many possible input combinations, that may come to the input of the combination circuit part ok. So, and because flip-flops are inside the circuit from outside it may not be so easy to initialise them to any known values we call it control and observe.

The DFT techniques particularly the one that we will be talking about will make the flip-flops easily controllable and observable. Controllable means you can initialise them to any value you want; observable means we can read out the value of the flip-flops whenever we want ok. So, the most important thing that we gain out of this is that is this last point, because we said that the sequential circuit test generation problem is very difficult. We want to simplify this problem into a combinational circuit test generation problem how we shall be seeing this?

(Refer Slide Time: 03:13)



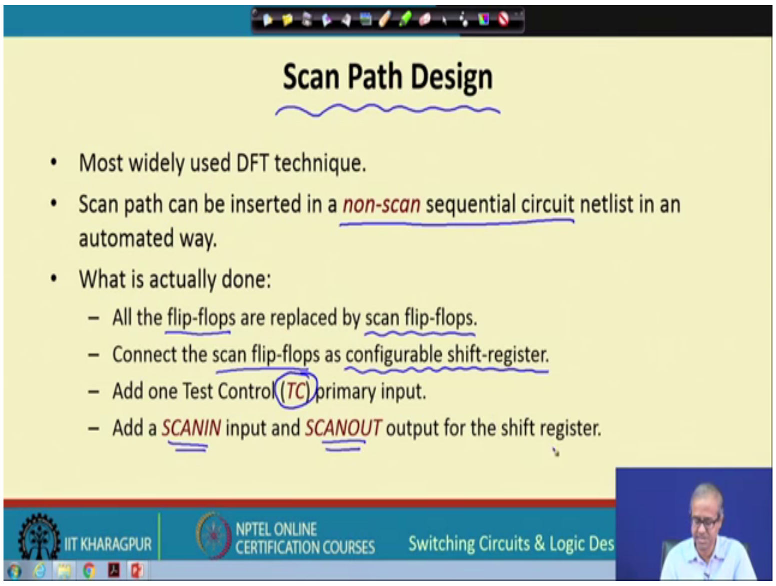
Model of a Sequential Circuit

With every clock, the next state becomes the present state.

So, we look at the standard model of a synchronous sequential circuit, where there is a combinational circuit part and the state variables in terms of the flip-flops, which are fed by a clock, present state and the primary inputs feed to the combinational logic which generates the primary output and the next state.

Now, as I said the main problem lies because since the flip-flops are inside the circuits which are not directly connected from the output from the inputs or outputs. Directly you cannot initialise this flip-flop, or directly you cannot observe the value of the flip-flops, this is the main problem.

(Refer Slide Time: 03:57)



Scan Path Design

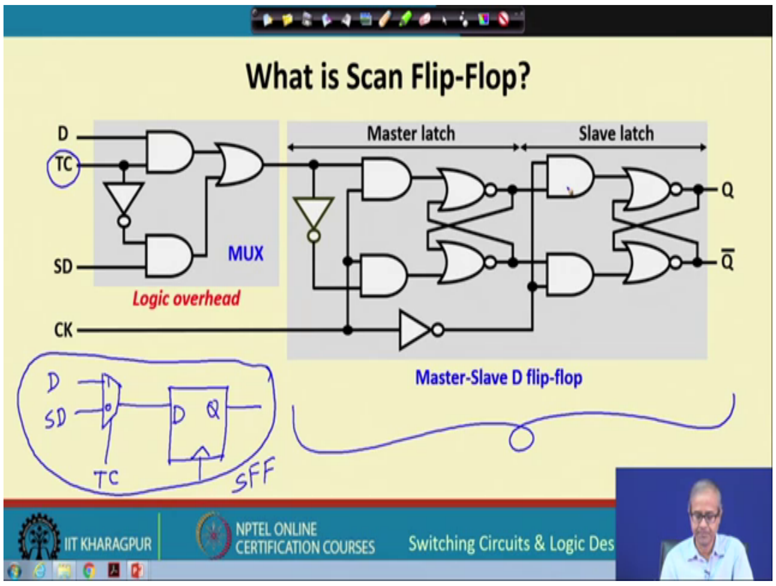
- Most widely used DFT technique.
- Scan path can be inserted in a *non-scan* sequential circuit netlist in an automated way.
- What is actually done:
  - All the flip-flops are replaced by scan flip-flops.
  - Connect the scan flip-flops as configurable shift-register.
  - Add one Test Control (*TC*) primary input.
  - Add a *SCANIN* input and *SCANOUT* output for the shift register.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

So, we will look at a very widely used and popular design for testability technique called scan path. Well, scan path I will explain what it means a non-scan sequential circuit is the conventional sequential circuit we know of, which start with a conventional synchronous sequential circuit and try to apply some rules to make it scan enabled you can say, scan path enabled. So, what are actually done there are a few steps, the steps are as follows.

First is that the flip-flops, they are modified into something called scan flip-flops. Then this scan flip-flops are connected in a certain way such that you can use them as a configurable shift register, we shall see how? One additional primary input pin is added and there are a couple of other pins scan in and scan out pins for the shift register that are also added. Let us see how it works?

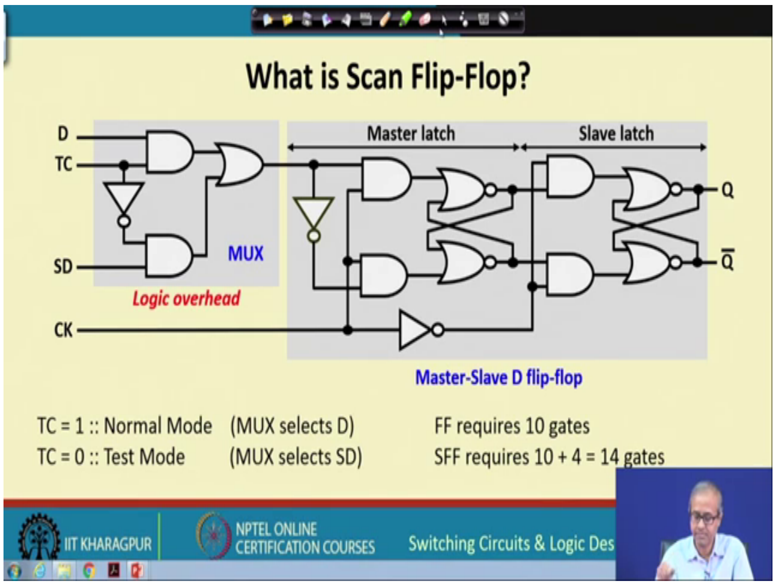
(Refer Slide Time: 05:21)


What is Scan Flip-Flop?

First let us see what is a scan flip-flop? This is a conventional master slave D flip-flop this part of it. So, you can say that I have a conventional master slave D flip-flop, D and a Q this I have with a clock. What I do at the input side I add a multiplexer to the input side and multiplexers select line is enabled by the this extra input TC, TC is stands for Test Control. During testing we need this and let us say this is 0 input, this is 1 input and the two inputs are called scan data SD and D.

So, a conventional flip-flop along with a multiplexer this is called a scan flip-flop, right. Scan flip-flop is nothing but I am adding a two to one multiplexer at the beginning of a normal master slave D flip-flop, right.

(Refer Slide Time: 06:50)



What is Scan Flip-Flop?

TC = 1 :: Normal Mode (MUX selects D)

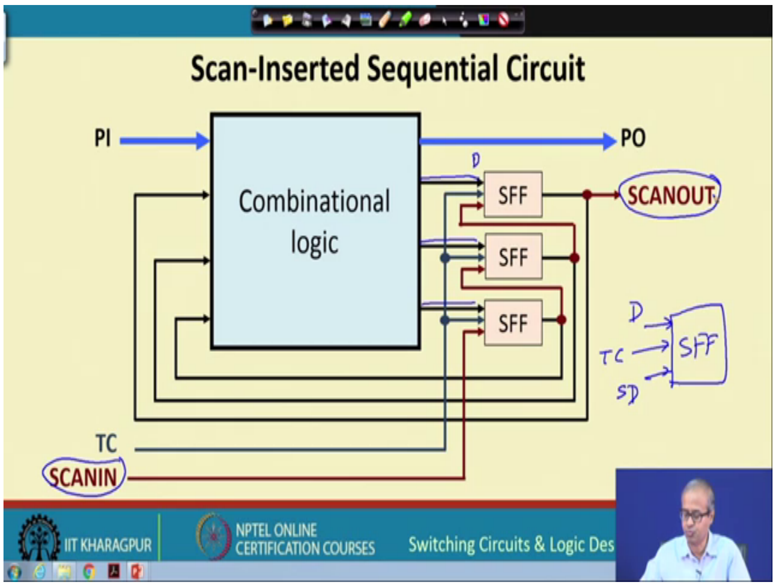
TC = 0 :: Test Mode (MUX selects SD)

FF requires 10 gates

SFF requires 10 + 4 = 14 gates

So, in the normal mode TC is set to 1, when D goes inside in the test mode when TC is 0 it is not the D input but SD goes inside, this is the only change, right. Earlier the normal master slave flip-flop required 10 gates but because of this additional thing we need 4 additional gates it becomes 14, just remember this.

(Refer Slide Time: 07:27)

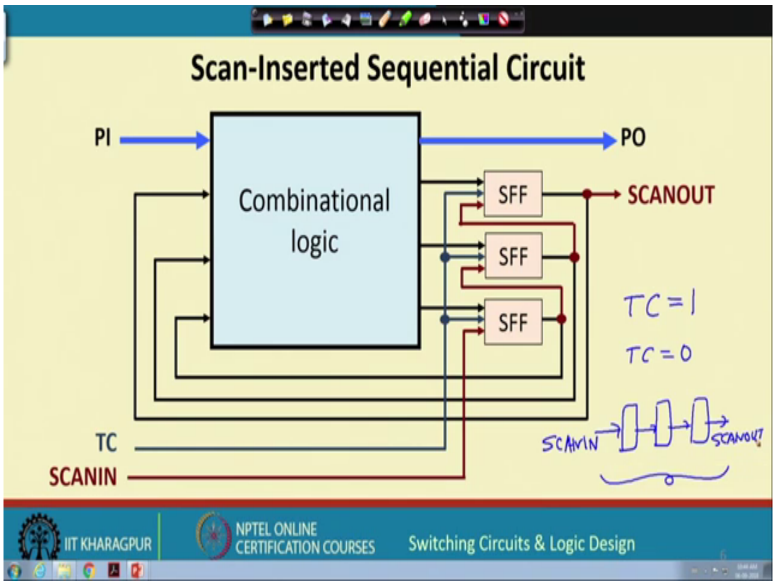


Scan-Inserted Sequential Circuit

Now, how do I modify our circuit? In the combination circuit the normal flip-flops are there, so we have replaced the flip-flops by scan flip-flops. You see the output of the combination logic was going to the input of the flip so, these lines remain as it is. They go to the D input of the scan flip-flop, for the scan flip-flop the first the upper input is D, the middle input is the test control, the middle input is TC and the lower input is the scan data right. And the output of the flip-flop we feed back to the input of the combination circuit as usual, they are coming.

But some additional circuitry we have added, how we have added? We have added another scan in input. This scan in goes to the SD of the first flip-flop, it goes to the SD of the first flip-flop and the output of the first flip-flop goes to the SD of the second flip-flop. This goes to the SD of the third, and the output of the last flip-flop also is brought out and it is called scan out one extra output pin.

(Refer Slide Time: 09:03)


Scan-Inserted Sequential Circuit

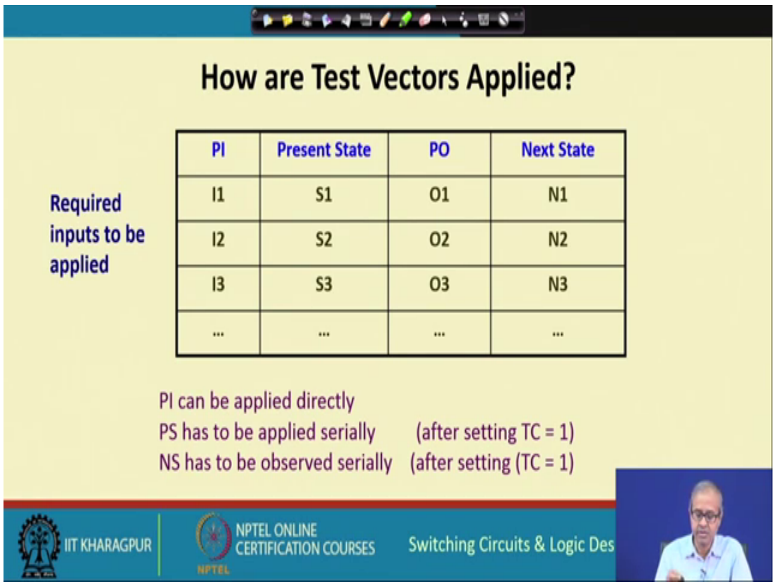
So, what we are actually doing here in this circuit? Just think, if TC equal to 1, this is the normal mode of operation. So, the combinational output goes in, D goes in and the output goes back the normal circuit operation. But, when TC equal to 0, the lower input of this scan flip-flop are selective, so it becomes like a shift register, you see it becomes a shift register.

So, once it becomes a shift register you have 3 stages of a shift register ok, they are connected like this. So, you have this scan in input connected here and you have this scan out connected here. So, whatever input you want to initialise in the flip-flop you can shift through the scan in input and whatever data is coming out you can observe through this scan out pin. This is the basic idea ok, just by adding this additional pins you can easily control the flip-flops you can easily observe the value of the flip-flops ok.

Now, another thing you see, now that we have a mechanism to completely control the flip-flops. This means not only the primary inputs I can apply any value I want here also because, I can initialise the flip-flop. Similarly, on the output side I can observe not only the primary outputs but also I can observe these, how? I can store them in the flip-flops then I can shift them out which means now my test pattern generation I can ignore the flip-flops.

I can look at the combinational circuit as if all the inputs I can apply directly all the outputs I can observe directly. So, I do not need a sequential circuit test generator, I need only a combinational circuit test generator where all the inputs I can apply all the outputs I can observe. This is what is the big advantage that we get here.

(Refer Slide Time: 11:15)



How are Test Vectors Applied?

Required inputs to be applied

PI	Present State	PO	Next State
I1	S1	O1	N1
I2	S2	O2	N2
I3	S3	O3	N3
...	...	...	...

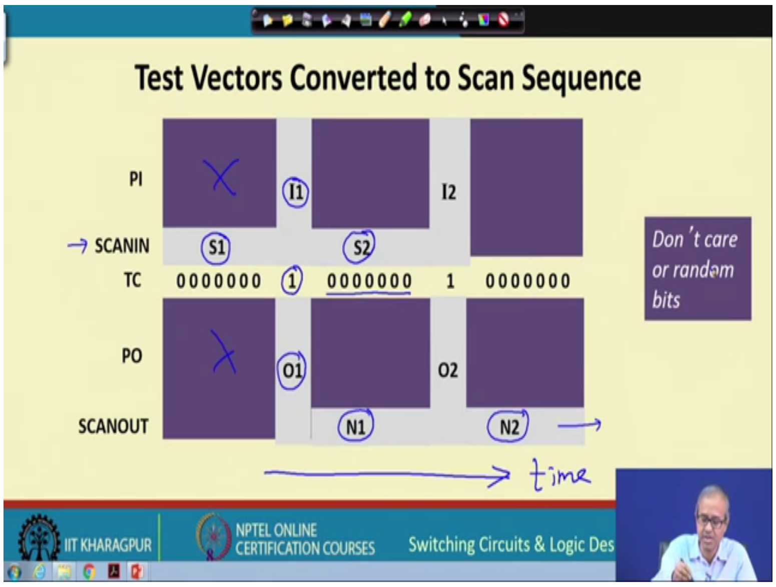
PI can be applied directly
PS has to be applied serially (after setting TC = 1)
NS has to be observed serially (after setting (TC = 1)

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

So, now the question is how was the test vectors applied? Now, the test generator you see just go back to the previous diagram once more, the PI and the present state, and PO and the next state. PI and the present state are the inputs PO and the next state are the outputs. So, the test generator will be generating the data where these will be the inputs of the combinational circuit and these will be the outputs of the combinational circuit.

So, let us say symbolically I am just given example. Suppose the test generators generated some inputs I1, S1, I2, S2, I3, S3 and the expected output values are O1, N1, O2, N2, O3, N3. Because, PI is available from outside it can be applied directly, but when you are applying this S1, S2, S3 you have to apply them serially through the shift register just after setting this equal to 0 not 1, this equal to 0. Similarly, the next state when you want to observe the output you have to observe them serially by considering them as shift register shifting them out right. And how are they shifted in and shifted out? I am showing pictorially in this diagram.

(Refer Slide Time: 12:43)



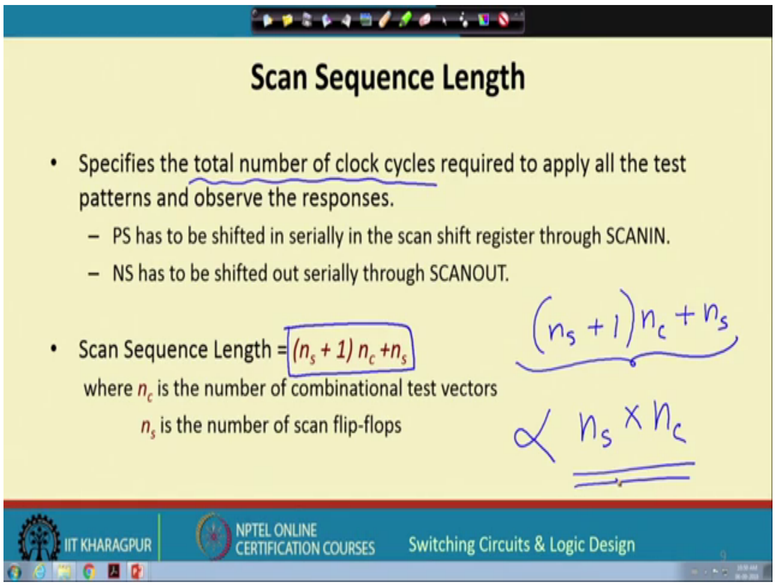
Test Vectors Converted to Scan Sequence

PI
SCANIN
TC
PO
SCANOUT
I1
I2
S1
S2
0000000
1
0000000
1
0000000
O1
O2
N1
N2
Don't care or random bits
time
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

As shown this is your axis of time, this is time. So, in the beginning you set TC equal to 0 and you shift in this S1 bit by bit through this scan in input. So, after S 1 has been shifted in suppose it is 7 bit, 7 bits are there then you apply I1 because PI, through PI you can apply I1 directly now, you are in the normal mode, TC equal to 1.

So, after doing this you can get through PO the outputs and whatever is the secondary output N1 the next state that through the scan out again you set TC to 0 you can shift them out. And here you can do some parallel operation, while N1 is being shifted out in parallel the next serial data this S2 the present state can be shifted in so, there can be parallel operations going on. Only in the last state there is no S only N2 is there this N2 has to be shifted out and this dark places are do not cares during the shifted in so PI are do not care you can apply anything you want. So, this is just to show you pictorially how the data input and output operations take place.

(Refer Slide Time: 14:26)



Scan Sequence Length

- Specifies the total number of clock cycles required to apply all the test patterns and observe the responses.
  - PS has to be shifted in serially in the scan shift register through SCANIN.
  - NS has to be shifted out serially through SCANOUT.
- Scan Sequence Length = $\boxed{(n_s + 1)\, n_c + n_s}$

  where n_c is the number of combinational test vectors

  n_s is the number of scan flip-flops

$$\underbrace{(n_s + 1) n_c + n_s}$$

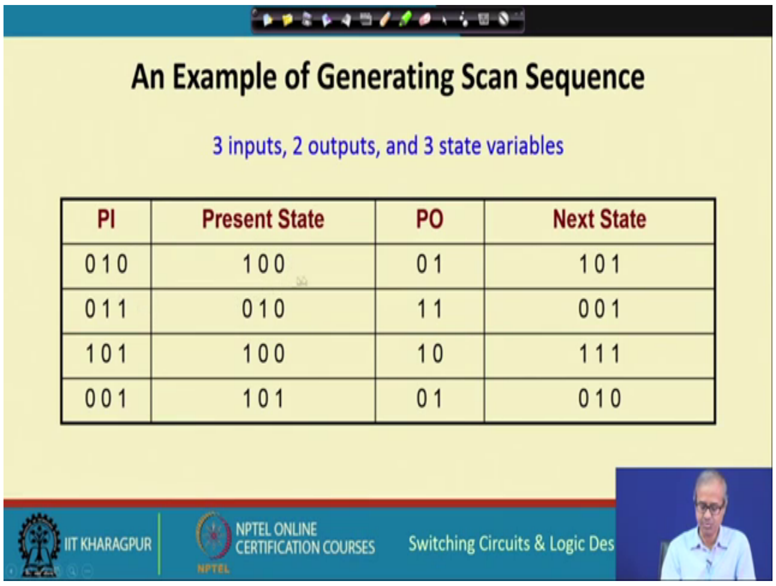
$$\propto \underline{\underline{n_s \times n_c}}$$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

But, the point to note is that that how much time it takes? So, you can make an simple calculation regarding the total number of clock cycles that are required, right. Now, the number of clock cycles can be determined like this n s plus 1 n c, because you see for every test vector n s this n s denotes the number of scan flip-flops. This many clock cycles will be required to shift in the data, then one cycle will be required to apply the PI and observe the output and you will have to repeat this for so many test vectors. There are n c number of test vectors and for the last test vectors you will have to shift out the next state. So, for that you will need another n s this is how this equation is coming right.

So, this is one drawback of this design for testability technique that the number of clock cycles required here will be roughly proportional to number of flip-flops multiplied by number of test vectors ok this is something you have to remember. That your testing time is increasing but you are able to get complete controllability and observability of the flip-flops that is the biggest advantage you gain out of this technique.

(Refer Slide Time: 16:02)



An Example of Generating Scan Sequence

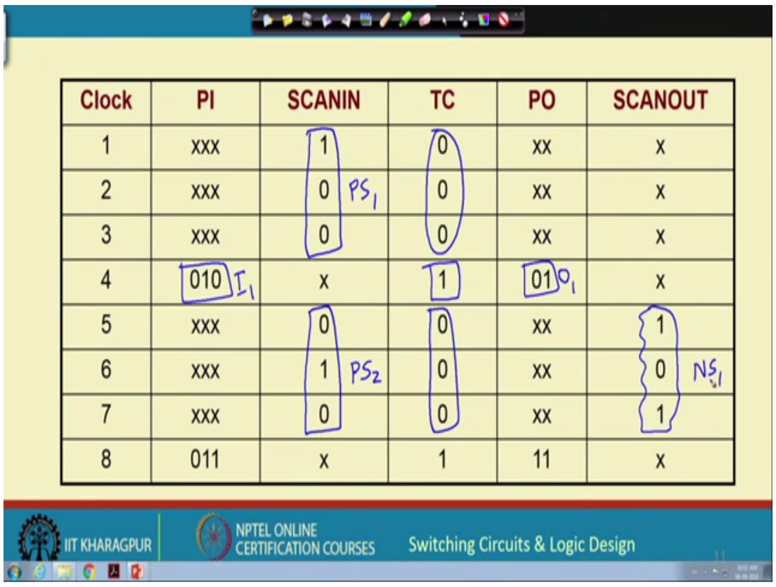
3 inputs, 2 outputs, and 3 state variables

PI	Present State	PO	Next State
010	100	01	101
011	010	11	001
101	100	10	111
001	101	01	010

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Switching Circuits & Logic Des

So, a very specific example 3 input 2, outputs and 3 state variables so, we can just apply this. Suppose, these are the tests which are generated by the test generator, these are the inputs and these are the expected outputs.

(Refer Slide Time: 16:22)

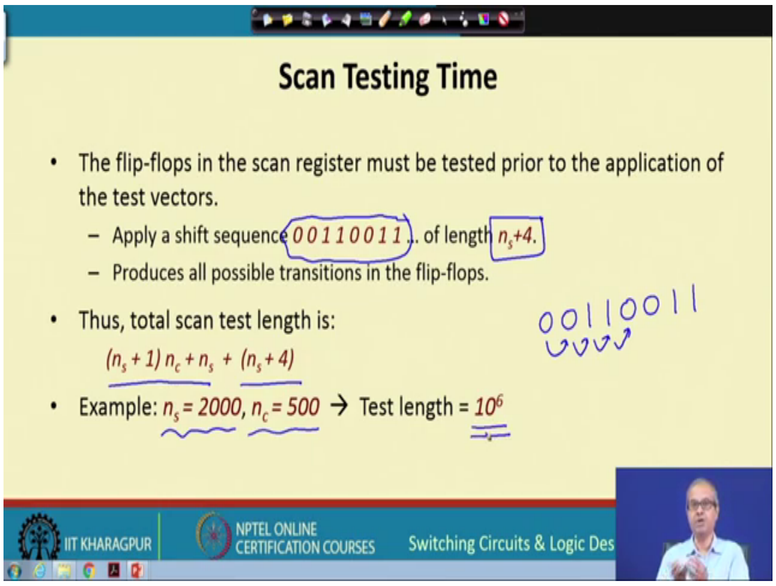


Clock	PI	SCANIN	TC	PO	SCANOUT
1	xxx	1	0	xx	x
2	xxx	0 PS_1	0	xx	x
3	xxx	0	0	xx	x
4	010 I_1	x	1	01 O_1	x
5	xxx	0	0	xx	1
6	xxx	1 PS_2	0	xx	0 NS_1
7	xxx	0	0	xx	1
8	011	x	1	11	x

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Switching Circuits & Logic Design

And clock cycle wise you can apply like this. So, first 3 clock cycles TC will be 0, you shift in the value of the present state, after it is done you set TC to 1, apply the primary input this is your I1 and observe the primary output. Then again set TC to 0, you observe the next state, in parallel you also feed in the present state of the next vector. This is your present state for the first vector, this is your primary input part of the first vector, this is your present state for this second factor, this is your output for the first vector, this is your TC, this will be your next state for the first vector. So, in this way it will go on ok so, this gives you roughly an idea how the thing works.

(Refer Slide Time: 17:42)



Scan Testing Time

- The flip-flops in the scan register must be tested prior to the application of the test vectors.
  - Apply a shift sequence $\boxed{00110011}$... of length $\boxed{n_s+4}$.
  - Produces all possible transitions in the flip-flops.
- Thus, total scan test length is:

  $\underline{(n_s+1)\,n_c+n_s} + \underline{(n_s+4)}$
- Example: $n_s = 2000$, $n_c = 500$ → Test length = $\underline{\underline{10^6}}$

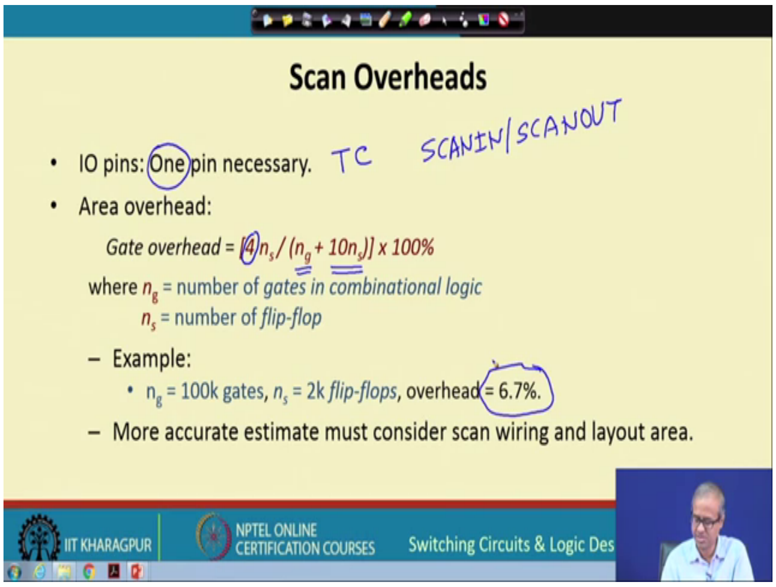
00110011
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

Now, just regarding the total scan testing time let me very briefly talk about another thing that well we have tested the combinational circuit part but what about the scan flip-flop part, there can be some faults there also. Now, there is a standard technique of testing a shift register. So, you apply a pattern like this 0 0 1 1 0 0 1 1 0 0 1 1 of total length n s plus 4. So, when we apply a pattern like this so, you can see that you are going through all possible transition 0 to 0, 0 to 1, 1 to 1 and also 1 to 0. So, you are verifying that all the flip-flops are able to carry out all possible transitions so, this is way you can test this scan flip-flop.

So, the total scan test length will be this is something which we have already seen earlier plus this. So, as an example if there are 2000 scan flip-flop and 500 test vectors if we multiply the test length becomes about a million. But, it is not really much, you see during testing if you apply a clock pulse let us say at 100 megahertz frequency then 10 to the power 6 is nothing, it will hardly take you how much I think 10 milliseconds or so ok. So, this is not a really big deal, because nowadays the clock frequency is very large.

So, even if we apply you have to apply a large number of test vectors or you can afford to do that ok.

(Refer Slide Time: 19:31)



Scan Overheads

- IO pins: One pin necessary. TC SCANIN/SCANOUT
- Area overhead:

  Gate overhead = $[4n_s / (n_g + 10n_s)] \times 100\%$

  where n_g = number of *gates in combinational logic*

  n_s = number of *flip-flop*

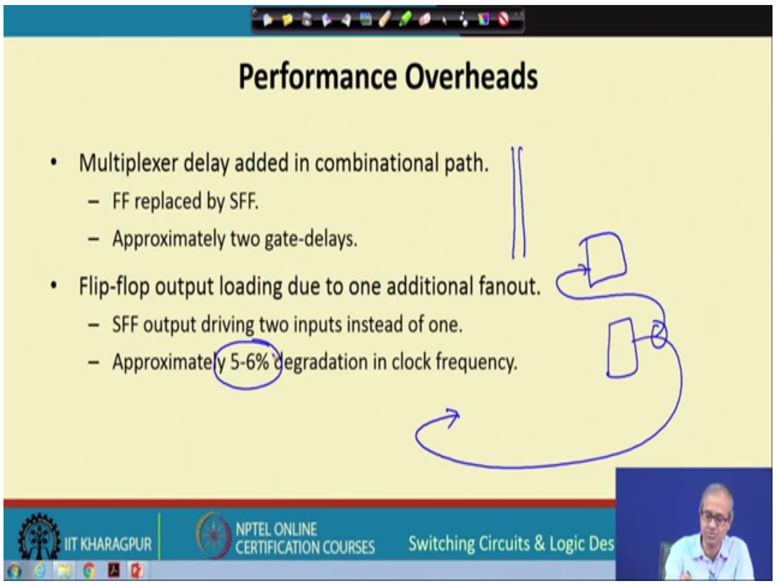
  - Example:
    - n_g = 100k gates, n_s = 2k *flip-flops*, overhead = 6.7%.
  - More accurate estimate must consider scan wiring and layout area.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

So, talking about this scan overheads well you need one mandatory extra input pin TC. But regarding this scan in and scan out, well if you can afford to have additional pins it is alright but, otherwise you can share them with your PI or PO pins also because PI and scan in you are never using them together. Similarly, PO and scan out you are never using them together you can share some of the pins if you want.

And talking about area overhead, for every scan flip-flop we said that 4 additional gates are required and if n g denotes the total number of gates in the combinational logic it will be n g plus so many flip-flops. So, originally the flip-flop contains 10 gates, but in this scan flip-flop 4 additional are required so, this is your gate overhead. So, a simple example numerical example 100k gates and 2k flip-flop, 2000 flip-flops, so overhead will be about 6.7 percent, these are rough estimate.

(Refer Slide Time: 20:50)



Performance Overheads

- Multiplexer delay added in combinational path.
  - FF replaced by SFF.
  - Approximately two gate-delays.
- Flip-flop output loading due to one additional fanout.
  - SFF output driving two inputs instead of one.
  - Approximately 5-6% degradation in clock frequency.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

And other thing is that there are some additional delays which are included in the path which you should also consider, because you have an additional multiplexer delay from the output of the combinational circuit because, the flip-flops have been replaced by scan flip-flops, so approximately two gate delays. And for the flip-flop the fanout is increasing, because earlier the flip-flop output was only feeding to the input of the combinational circuit but, now it is going to the input of the next flip-flop also. So, increase in fanout means extra delay.

So, roughly speaking this leads to 5 to 6 percent degradation in the clock frequency so, these are a few things you should keep in mind. So, we with this we come to the end of this lecture, where we have given you a brief overview about some of the design for testability techniques, DFT techniques that can be used to handle or tackle the problem of testing synchronous sequential circuits.

Thank you.