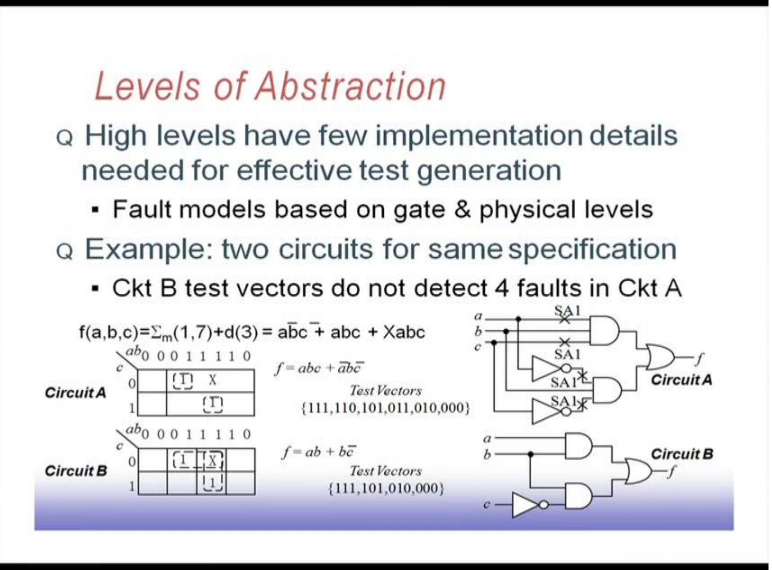
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
Introduction (Contd.)

Now levels of abstraction like a system can be viewed at different levels as we have already seen that we can see that gate level, we can see that transistor level or it maybe even at a much higher level at a functional level you can see.

(Refer Slide Time: 00:22)



So, as we go higher and higher levels of implementation. So, we have got lesser detail about the internal implementation. So, implementation details are not available, so as a result for test generation. So, it will be more at the input output situation input output points only. So, it will not go into the details of the implementations.

So, fault models based on gate and physical level. So, you when we are considering the gate level, so we have got some fault models. So, you may go to physical level in terms of transistors and wires, we have got a more detailed view of the system. So, as a result the types of faults that can be detected by a lower level fault model. So, that may not be possible with a higher level fault model, but of course, if when we are going for this lower level fault model, the number of faults and number that are that are possible becomes huge. So, as a result it may be difficult to generate is better for all those faults.

So, that way a higher level model can help us in generating test patterns which are much more compact in nature.

So, as a typical example consider 2 circuits here. So, both of them are corresponding to say this particular Boolean function; so a b bar, c bar, plus a b c plus a b c. So, this is sorry this is a b bar c bar plus a bar b c this bar should move here, a bar b c plus x a b c. So, this is this term is do not care. So, now, naturally I can have different type of grouping. So, in one realization it may be like this a b c plus a b bar b c bar, in another case I can have this realizations. So, a b plus b c bar. So, that is also possible.

Now, you see that when you are looking into this circuit. So, this is the circuit implementation, and for circuit b this is the circuit implementation. Now if we use a test vector generation for this circuit. So, these are the test vectors that can detect different faults off on or all these lines in the circuit, where as this type of test vectors it will not be able to detect these faults. So, the fault, so these line stack at one or these lines stack at one or these line stack at one. So, this faults cannot be detected by test vectors that I have generated considering the second implementation of the circuit.

So, if we have. So, implementation dependent is generation. So, that is important. So, we have to have the proper implementation details for generating the tests properly.

(Refer Slide Time: 03:22)

Overview of VLSI Test Technology

Q Automatic Test Equipment (ATE) consists of

- Computer – for central control and flexible test & measurement for different products
- Pin electronics & fixtures – to apply test patterns to pins & sample responses
- Test program – controls timing of test patterns & compares response to known good responses

Now how do it test a system? So, what are the components that are needed when we are trying to test chip or a system? So, one very important part in this testing process is the automatic test equipment or ATE. So, this will be coming to us again and again. So, this is some sort of computer. So, that will store that is patterns and responses in it and there will be some facility by which this test patterns can be applied to the circuit and the test through the channels of the ATE, and the responses will be collected and then you can collect you can compare these responses with the fault free response is stored in the computer's memory.

So, this ATE it will consists of a computer for central control and flexible test and measurement for different products. So, depending upon the type of tests so that can be done with this ATE, we should have facility for applying those patterns or measure the values or if it is allowing you to do analogue testing. So, I ready queue testing. So, you have to be able to measure the current and all that.

Second thing is that pin electronics and fixture; how do you apply this patterns to the pins and sample responses. As we know that this chips that we are testing they can have a different number of pins in them, similarly this ATE that I have. So, the ATE also will have a number of channels. So, this ATE channel width is also has to be some fixed value.

Now, the problem is that suppose we have an ATE which has got a channel width of say 40 bits. Now I have a chip that has got say 100 pins in it, now how do I test this 100 pin chip with a 40 an ATE with the channel width of the 40 bits. So, naturally we have to do some sort of multiplexing. So, how can we do all these things? So, that is actually a part of the ATE. So, ATE can tell like you can apply it is patterns in this fashion by doing the multiplexing. So, this is pin electronics and fixture. So, this will be talking about what are the ATE channels we have, and how to use them for applying it is patterns.

And the third thing is that test program. So, that will control the timing of test pattern compare responses to known good responses. So, this when that is patterns are to be applied, in what sequence and what is and it will collect the response at what time it should be collected, and then the comparison part. So, these are actually part of ATE and since the circuit complexities are going up. So, this ATE will need to have large amount of memory to hold a test patterns and the responses.

So, that way the cost of the ATE is also becoming in normal. So, if it depends on two factors, the channel width and also the memory capacity that it has. So, there are challenges like the as I C technology is increasing significantly. So, some chips are becoming more and more complex, but ATE that is the one that is most of the company fabrication house or the test testing company they will be having one ATE or say certain capacity. So, it is not advisable that with every new chief that we design. So, we have to have an ATE that can run at a higher speed. So, every new version of chip running at higher speed at ATE speed should also go out. So, there must be some technique by which we can use this older ATE's before the faster chips as well. So, that is also a part of this ATE design.

(Refer Slide Time: 07:15)

Overview of VLSI Test Technology

- Automatic Test Pattern Generation (ATPG)
  - Algorithms generating sequence of test vectors for a given circuit based on specific fault models
- Fault simulation
  - Emulates fault models in CUT and applies test vectors to determine fault coverage
  - Simulation time (significant due to large number of faults to emulate) can be reduced by
    - Parallel, deductive, and concurrent fault simulation

So, ATE stores the test patterns; now where from where from do we get those test patterns. So, they are actually coming from some automatic test pattern generation technique called ATPG. So, there are very nice algorithms by which we can generate test vectors for a given circuit on specific fault models. So, you say that so this circuit has got say 10 points at which there maybe faults. So, each of those is if you are considering this stuck at fault, then for every point we need to generate the test patterns. So, there are certain algorithms by which we can do this test pattern generation. So, they are known as automatic test pattern generation algorithms.

Second important another piece of software; so ATPG is a piece of software which is a generating the test patterns; another important piece of software that are needed by testing community is the fault simulation techniques. So, what happens is that I am given set of test patterns. So, how do I know how many of the faults are actually covered by this test pattern set? So, for that purpose I need to inject those faults into the circuit, apply the test pattern set, and see whether it is detecting the faults or not.

Now, naturally we cannot do this before the circuit comes into existence. So, and at a particular the physically their fault has occurred in the circuit. So, but that will they that way we cannot wait for the faults to occurs. So, what is done? In software itself; so we can have a piece of simulator, which will be injecting those faults into the circuit description and it will try to see like what is simulate the circuit in the presence of those faults by applying the test vectors and see whether it is a giving us some responses which is different from the fault free responses. So, if it is so then we can say that this test pattern is detecting the has the capability to detect the fault.

So, this fault simulation technique so the emulates fault models in the circuit and apply a test vectors and determine the fault coverage. So, naturally the question is the simulation type a time, how much time it needs to do the simulation. So, if I have got say 100 test patterns and say 100 faults, and if I am simulating each the system for each fault and each test pattern; that means, the entire circuit has to be simulated 100 into 100 times. So, that is the huge number. So, can we reduce this time? So, there are you may go to fault simulation technique, so we will see there are several techniques by which we can do this fault simulation, say parallel simulation and deductive simulation, concurrent simulation. So, there are several techniques by which we can do this over and above the very simple serial simulation.

(Refer Slide Time: 10:14)

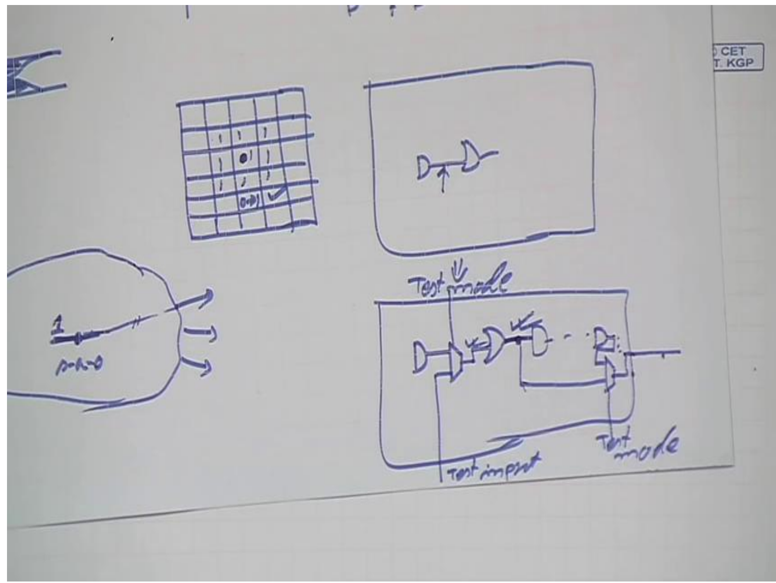
Overview of VLSI Test Technology

- Design for Testability (DFT)
  - Generally incorporated in design
  - Goal: improve controllability and/or observability of internal nodes of a chip or PCB
- Three basic approaches
  - Ad-hoc techniques
  - Scan design
    - Boundary Scan
  - Built-In Self-Test (BIST)

Now, for helping in the testing process; so there are certain techniques that have been incorporated into the design, they are known as design for test ability. So, design for test ability means. So, if a circuit is designed by the designer and the test engineer is asked to do the testing. So, it may so happened that it is difficult very difficult to prove certain points inside the circuit, because these I C chips. So, I do not have access to the internal points of the chips. So, I have to do it through the primary input output lines only. So, it may be difficult to access those points from the primary inputs and outputs. So, if it is difficult to access, so it may be difficult to test the faults that might have occurred at those points.

So, what is designed for testability tells is that we incorporate something extra into our design, so that this testing process is applied. So, it improves controllability, controllability in the sense that I can put any desired point in the circuit to a desired a logic value. So, that way it is controllability. Similarly observability means I have got I can observe the value of some point in the circuit by some technique at the primary output. So, what I essentially mean is that, if this is the full circuit that I have full design that I have now. So, it may be that I want. So, you know I have got the primary inputs to the system; here I have got the primary outputs.

(Refer Slide Time: 11:41)

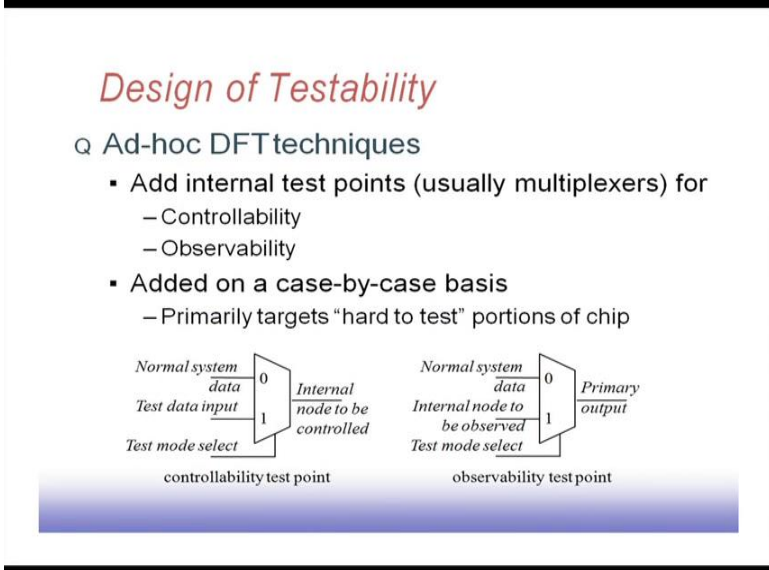


Now, if I want to check the status of this particular line. So, whether this line is say stuck at 0 or not. So, if I want to test this one, then somehow I have to apply some patterns, so that this line becomes in a fault free circuit, this line becomes one. So, that has to be done. So, somehow I have to apply the test patterns, so that it becomes one.

Similarly, due to this effect of this fault, it should get propagated to some output, so that I can see that this a fault has occurred. So, one thing is that to control this point to a desired value and the second thing is to observe the value at some primary output. Now if the circuit is quite complex, it may not be very simple to have a to do this things. So, what is done? We have to have some additional hardware incorporated into the system, so that this a checking becomes easy.

Now, so this DFT, so this is generally incorporated in to the design and the goal is to improve controllability and our observability of internal nodes of the chip or it may be for PCB also, that PCB also we have got a number of chips mounted on the PCB. So, that individual chips inputs and outputs may not be traceable. So, we have to have a look at that from the system input output point of view. So, that way we can have to you may need to have a PCB also some feature has to be incorporated so that it becomes testable.

(Refer Slide Time: 13:48)



So, there are 3 main approaches for this design for testability insurance; one is know the ad hoc technique. So, second class is a scan design or boundary scan, the third one is built in self test. So, we will look into each of them in a nut shell. So, adopt technique it means that we add additional internal point. So, test point. So by multiplexers, maybe that this point we need to control. So, what we do? We put a multiplexer before that, and there is a controllability it is this multiplexer control input is there, so if this control input is 0, then this normal system data will pass through this so that the circuit will operate at it is normal way, and if you want to see if you want to put some particular value at this at this point. So, we have got some test data input. So, test data input will be passed through this multiplexer at this point.

So, this way we can control the value that we have at this point. So, that is known as. So, controllability is enhanced by this means. On the other hand the observability; so if you want to improve the observability, so what happens is that? So, this point we want to observe. So, we add another multiplexer and we put it the, this is the normal system output can go to primary output. So, it pass it to a multiplexer, so if this point is selected then for normal operation of the system though this normal system data will go to primary output, where as when we are trying to observe the value of some point internal point. So, we put it like this.

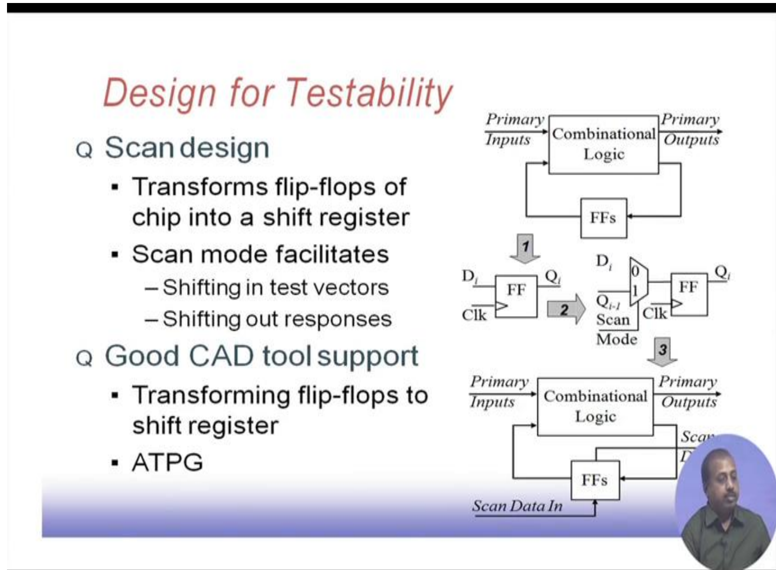
So, what we essentially do is that if this is a big circuit that we have, and you want to say there is one and gate and this output is going to some or gate etcetera, and we want to control the value at this point. We want to control the value at this point, so this circuit will be modified like this. So, this and gate is there. So, it will be passing through a multiplexer and this multiplexers output will go to the OR gate. So, this line is equivalent to this line and there will be another line. So, this is that circuit boundary there will be another line which will be taking out so which we call the test input. So, this is our test input and this multiplexer has to be controlled. So, this multiplexer has to be controlled; that is controlled by that test mode. So, this is selecting the test mode selecting the test mode. So, if the test mode select this test input then the test input can be put at this point otherwise a normal operation it will go.

Similarly, if you want to observe the value at that we have in this point. So, it may be difficult because it may be feeding some other gates etcetera. So, it is difficult to see the output. So, what you do is that we select one primary output, we select one primary output want to which we want to visualize this point see this point. So, if this primary output is say finally, coming from son gate, so it is coming from say this get it is connected to this primary output. So, what we do? We modify it so that there is another multiplexer again. So, the connection established is like this and this is that test mode.

So, in test mode; so this line value will be available at this primary output. So, that way observability of this point is increased and similarly controllability of this point is increased by putting these extra components into the system. So, that way we can have this or we can add internal tests points for controllability and observability enhancement of course, which is quite ad hoc because wherever we feel that it is difficult to observe a line or control a line, we are putting this type of multiplexers. So, primary target hard to test portions of the chip; that means, part of the chip for which you find it is very difficult put a particular signal line to some desired value. So, there we do this thing and similarly for the output also where it is very difficult to observe. So, there we actually put this one.

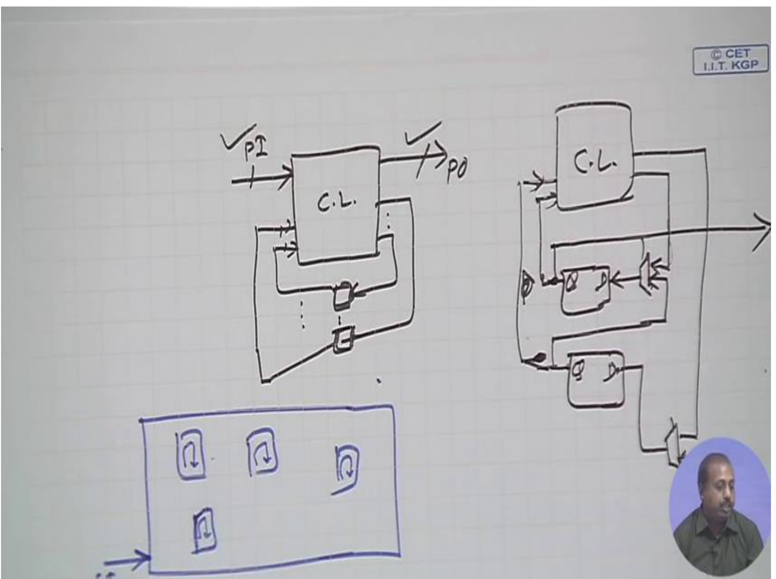
So, this will be added on case to case basis and whatever you do from one circuit may not be can true for the other circuit. So, it has to be vary from circuit to circuit, and it is hard to test regions of the chip that is targeted in this fashion.

(Refer Slide Time: 18:39)



Other way out is to have this scan design. So, this is the typical system that we have. So, any system will have this combinational logic and some flip flops in it. So, we transform this flip flops, so that they form a shift register ok.

(Refer Slide Time: 19:07)



The basic difficulty that we have is that, if I have a combinational logic. So, it has got some primary inputs and there are some flip flops which at the nested flip flops.

Now, you see that these being primary input and primary outputs. So, these lines they can be controlled very easily and also observed very easily, but it is difficult to control

say values that I need to put at this point. Because starting with the initial state of the finite state machine, so, these flip flops they will have some value depending upon the state transition that are affected into the system. So, as a result it is difficult to come up with a proper the state transition, so that we have the desired values at this flip flops. So, this scan design is a transformation that will help us in doing these operations.

What we do is that say this is the original flip flop that I have. So, this say D i is the d input Q i is the Q output and clock. So, it is transformed into something like this where you put a multiplexer. So, in this multiplexer we have got this D i coming connected to the 0 line, and Q i minus 1 that is the Q output of the previous flip flop in the chain. So, that is connected here and rest of the thing remains unaltered. So, in a design what happens if this flip flops they form a change? So, what we are doing is that we have got one flip two flip flop say for example, in a design. So, we put some multiplexer here, we put some multiplexer here, here I have got the combinational logic. So, this data input comes here where as from this multiplexer, from this D flip flop the Q output it is put here.

So, similarly for this flip flop we have got the input coming from here. So, that will again go to a multiplexer and this output goes here. So, this way this output is all these outputs they also simultaneously go to the circuit for normal operation. So, for normal operation these multiplexers are selected in such a fashion, so that these combinational logic outputs are fed into the multiple in to the flip flops. So, that they operate in the normal operate normal mode. But in the test mode what happens is that this we put some pattern here say suppose we put a pattern value S here. So, these value S will be coming into the d flip flop, and the next cycle. So, in the next clock so that S value will go to the other the next flip flop and whatever new value we put in this line. So, that will come to this flip flop.

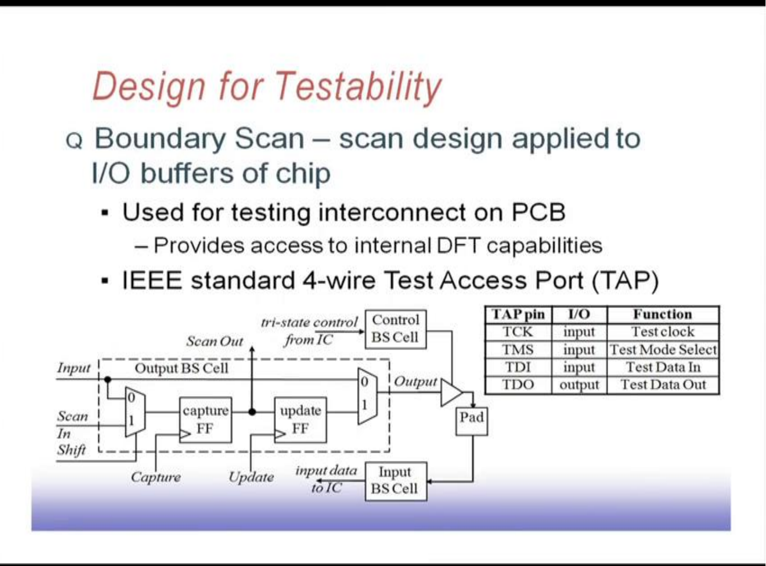
So, this way it will form a chain of flip flops, now once we have got a chain of flip flop that is forming a some sort of shift register for this pattern, for these out of these a flip flops that we have in the circuit; so that that they have converted into a shift register. So, for applying test vectors the primary input part we apply directly, for the state bits. So, we have configured this state flip flops into a shift register. So, we shift this test vectors into the shift register.

Then the circuit is made to operate in the normal mode. So, when it operates in the normal mode, so these patterns are pattern is applied to the circuit and the then the responses are collected again into this flip flops. Now you see that when the when this is operating the test mode. So, what we can do? We can take out this point and say that is our scan out point S O and this point is scan in this is S I.

So, what we can do? We can we can scan in the test vector similarly once response has been captured into this flip flops, so we can scan out the responses through this s o line serially. So, this scan in and scan out operations can be used for putting that test vectors into these flip flops, so that my test application process becomes simpler and of course, there can be overlaps like when we are shifting out the response of previous vector. So, we can shift in the next test vector.

So, this is one of the very standard designs like we though the flip flops that we have in the circuit. So, they are converted into a scan flip flops and they are connected over shift register organization and whatever value we want to have in those flip flops, we can shift them serially into those flip flops and accordingly we can apply the some desired test pattern there. So, the cad tools that we have. So, they have got facility by which we can transform all the flip flops into this shift register, so that can be done; and these ATPG tools they are also they have which take help of this scan based circuits, so that the test pattern generation becomes simple; compared to sequential circuits where we have got flip flops this test pattern generation is a complex problem, but once we have got this scan incorporated into the system. So, this entire system can be treated as a combinational circuit only. So, that way the combinational ATPGS can be used for generating the test patterns.

(Refer Slide Time: 25:14)



So, in design for testability we have got this another important issue with the PCBs. So, this is basically in PCB what we have is, we have got a number of chips mounted. So, they are mounted in some fashion and each of these chips, so if this is the PCB that we have. So, on this PCB we can have several search chips mounted; several search chips mounted now this has got. So, if you want that these flip flops they should be initialized to some value. So, they have got their own scan chains inside, now how to take help of those scan chains and if you want to shift in some serial patterns through this into this scan chains how can it be done. So, this is done by means of this boundary scan structure.

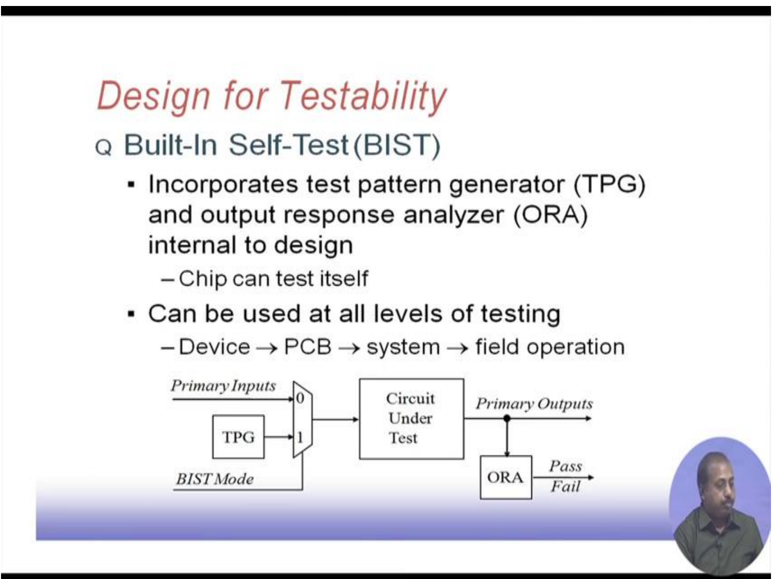
So, what happens is that. So, one boundary scan cells. So, it is structured like this. So, this normal input will come on this input line and this scan input will come on this line and this shift control will control whether this input will go or this scan input will go. So, accordingly this will come to this capture flip flop. So, this when this capture flip flop comes then the pattern gets, so the response gets captured on to this.

So, it can go to the scan out of this cell. So, that it can be connected to another scan cell entry; see and this also you have got update. So, this update will be updating the output of this cell, and then it is going to this boundaries scan cell control and that can go to this pad. So, this is the input output pin. So, from the input output pin we can have this input output cell and that connect to the I C or from it can go to I C or if it is an output. So, we

can see the output at this pad. So, from this pad that value can come to as an input to the I C or it can go we can observe the output from this point.

So, these are used for this PCB manufacturing, PCB testing and all that. So, there is an IEEE standard which is called test access port, where we have got this line T is a test clock test mode select line test data input line and test data output line. So, any chip that follows this particular TAP standard this IEEE standard. So, they can be connected over boundary scan architecture.

(Refer Slide Time: 28:01)



Also there are built in self test structure. So, where this is a where what happens is that the circuit on the test we do not apply test pattern from outside. So, in the circuit itself we have got a test pattern generator, and a response analyzer integrated. So, from outside it just tells that it should start testing now. So, according to this pattern generator it will go on generating the test patterns, they will be applied to the circuit; the responses that we get. So, they will be compressed at these output response analyzer and then this at the final when all the patterns have been applied this output response analyzer. So, it will declare whether this circuit passes the test or fails.

So, this is very much useful because from outside I do not have to apply any test vector. So, there is no question of ATE from automated test equipment transferring the patterns and all that. So, that way it is very much useful. So, it can be used the device level, PCB level, system level, or field operation; it is particularly important for field operation

because there are I do not have any ATE available, so there it is the circuit itself is generating test patterns, and comparing responses to detect it is help.

(Refer Slide Time: 29:17)

Concluding Remarks

- Many new testing challenges presented by
  - Increasing size and complexity of VLSI devices
  - Decreasing feature size
- Low power VLSI testing
- Temperature aware testing

So, we have seen that many new testing challenges have come due to the increasing size and complexity of VLSI devices; feature size and transition size is decreasing, low power VLSI testing. So, the power reduction is an important factor, because it is normally seen that this test power is much higher compared to normal system op normal operation power, we will see why this thing happens later, but this is one fact and with this increasing power consumption the temperature is also increasing significant. So, that also needs to be considered. So, initially we will be concentrating on these testing techniques ATPG, faults simulation and all these things and towards the end of the course we will look into these advanced topics like low power testing temperature aware testing etcetera.

Thank you.