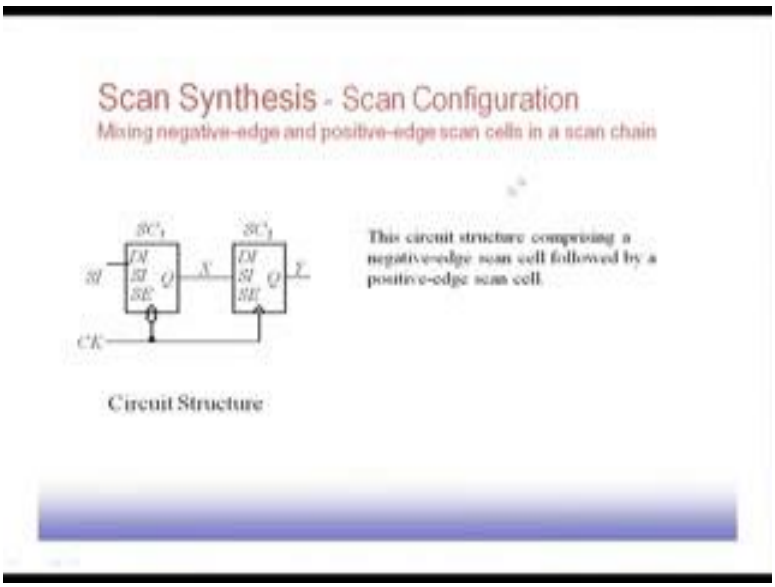
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

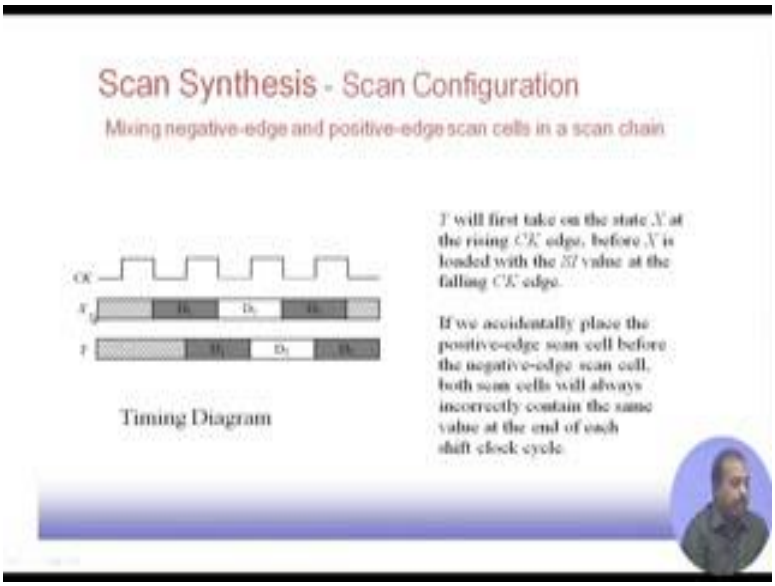
Lecture - 10
DFT (Contd.)

(Refer Slide Time: 00:23)



Scan synthesis; so in this scan synthesis the one stage of it is scan configuration. So, in that process, while mixing this negative edge and positive edge signals; positive edge scan cells in a scan chain because the circuit, so it is having a negative edge scan cell followed by a positive edge scan cell. So, this is the circuit structure. So, it has to be connected like this because after this negative edge has come, this value will be set to some proper value and then in the positive edge, the value will can be sense by the next scan cell. So, they need to be group like this.

(Refer Slide Time: 00:58)



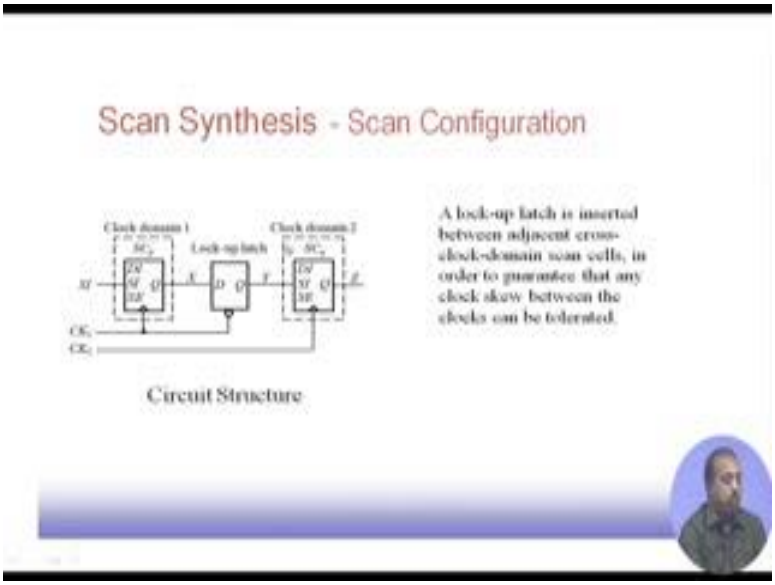
The way it operates is that this is the scan clock that is running and this is the X input, this is the SI and this is the X and then we have got Y. So, suppose when this clock is going low at the negative edge, the value is put on to this X line then when this clock was going high, So, this D 1 value comes to this one, similarly this X gets the value D 2 when this line is; this clock is going low at this point and when this clock is going high, the next flip flop gets the value of D 2.

Y will take the take on the first X at the rising clock edge before X is loaded with the SI value at the falling clock edge. So, that will happen, but if we do just the reverse, if you put that positive edge scan cell before the negative edge scan cell, in that case both scan cells will always in correctly contain the same value at the end of each shift clock cycle. So, what you say is that suppose we just do the reverse that is we have got this one is positive edge triggered and this is negative edge triggered. So, what will happen? When this SI value is given when the positive edge comes the value will be loaded here and in the negative edge that value will be loaded here, but this flip flop will continue; still continue with the previous value.

For some point of time both the flip flops will have the same value. So, what is happening is that whatever value you put on to this D flip flop, so in on to the SI line. So, it will get loaded onto both the flip flop. So, whenever I have got 2 types of flip flops getting mixed, negative edge triggered and positive edge triggered; the design synthesis

should always do like this that the negative edge triggered flip flop should be put fast and then the positive edge triggered flip flop, otherwise there will be problem.

(Refer Slide Time: 02:57)



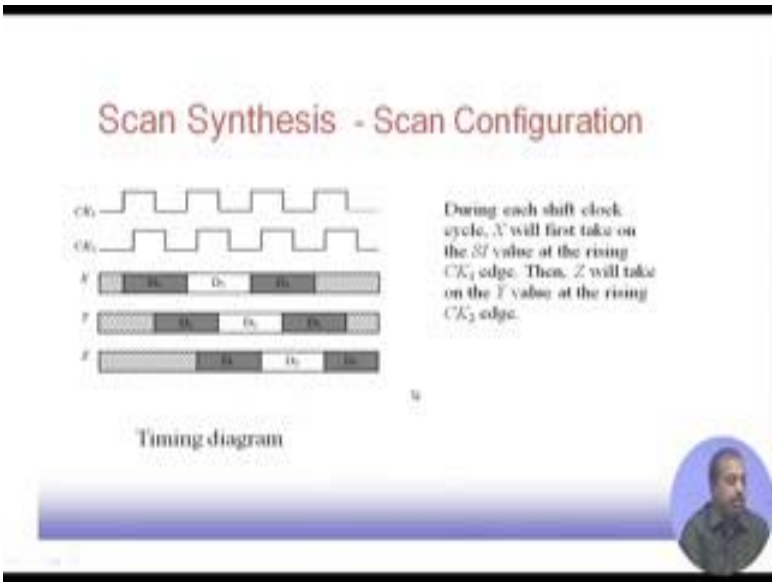
Then comes then if there are 2 different clock domains like so this is the domain 1 and this is domain 2. So, in between we need to put a lock up latch. So, it is a lock up latch will be inserted between adjacent cross clock domain scan cells in order to guarantee that any clocks skew between the clocks can be tolerated. So, if there is a say clock 1 is driving this clock domain and clock 2 is driving this clock domain, it may so happen that there is a skew. So, over the time the difference between clock 1 and clock 2 the time difference; it changes. So, if it change then what we can do? So this, the value maybe before it has been sample by this clock domain to flip flop. So, it is this clock domain one is going to modify it.

What is done? When this clock is low, this lock up latch will hold the value and it when this clock becomes high. So, then this will remember the previous value. So, this lock up latch will be inserted in between whenever we have got this type of cross clock domain structure. So, in the some earlier example, we have seen that there may be data transfer between clock domains, whenever we have that thing, this scan synthesis process; it must introduce this type of lock up latches between the domains.

Here is that lock up latch sort of thing like say clock 1 is this one. So, clock 2 is this one. So, during each shift clock cycle, X will first take on the value of SI whatever is there at

the rising edge of rising CK 1 edge then Z will take on Y at the rising CK 2 edge. So, that will that is going to happen and Y in between; this Y signal in between, it will hold the value for Z to sample. So, this is you see that X has changed to the value D 2, but this if this Y was not there if this Y was not there then this Z will get a wrong value if Y was not there this latch was not there Z will get a wrong value because it has the because of this clock edge skew.

(Refer Slide Time: 05:17)



(Refer Slide Time: 05:23)

That has to be taken care off then at the RTL level at the register transfer level also. So, we can do something to enhance this testability. So, why RTL? The RTL designs are needed like, actually what is happening is that if you are talking at the gate level then the number of gates in the design is increasing significantly as the designs are becoming more and more complex. So, number of gates are becoming increasingly high. So, there is a triplicate tremendous growth in the number of devices then the time timing becomes a problem. So, we have we do not have much time to put the product in to markets. So, in the introductory classes we have seen that the time to market is an important parameter. So, that has to be controlled. So, if we are going to test at always at the gate level then we have to we have to do much more testing compared to a higher higher level testing like at the RTL level.

And there is a potential of yield loss potential of yield loss like if I do not do RTL level testing then we will have something like this that some tests may get bypassed and at the device level when you are doing. So, some functionality is might not have been tested thoroughly. So, that way there may be some potential yield loss.

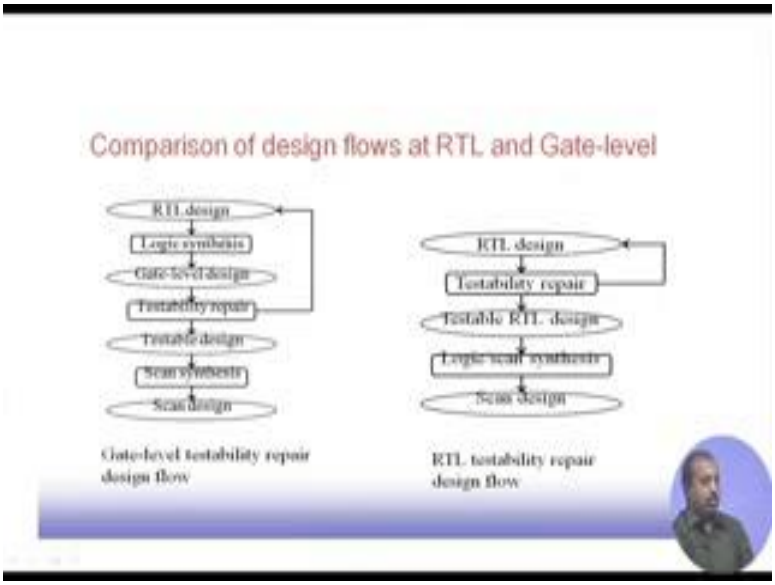
Then low power issues like at the RTL level. So, we know that we can, if the circuit is very big and we try to test the whole thing together. So, entire part of the circuit will be getting power as a result, there will be a lot of power consumption, but for a particular test pattern it may be targeted to test a fault in a particular region of the circuits. So, it may be a sufficient to excite only that region. So, the power only 2 that region, but this type of demarcation may be difficult to do when we have got a gate level circuit. So, when we have got a RTL level circuit. So, since we know that you have got this adder multiplier multiplexer this is some decoder like that. So, we can target each of them separately as a result we may be having better power reduction.

Then row core reusability. So, this is another very interesting issue because now most of the systems that we are designing. So, they are based on this reused principle. So, core are basically some pre designed pre verified module. So, that are going to be integrated into the system. So, if you have got this cores then we know that for apply for testing a particular core, the core provider will give us the test vectors directly. So, what is needed is just to apply those test vectors to that core and get the responses and check them, the because of this increased reusability. So, we do not need to do test generation for these individual cores as we are using this core based design. So, our effort may be less from

this point of sight because we are just have to apply the given test pattern. So, test generation is not necessary of course, application may become a problem we will see that later, but generation part at least is a speared.

Then time to market pressure. So, the overall time that we have in our hand is less. So, if you test at the RTL level. So, maybe we are testing in terms of functionalities. So, if you are testing in terms of functionalities, functions which are very much important. So, maybe we do we test those functions thoroughly functions which are not that much important maybe we just we do not do a hundred percent testing of that. So, as a result the time to market will reduce and that will help us in getting better be returned from the product.

(Refer Slide Time: 09:14)

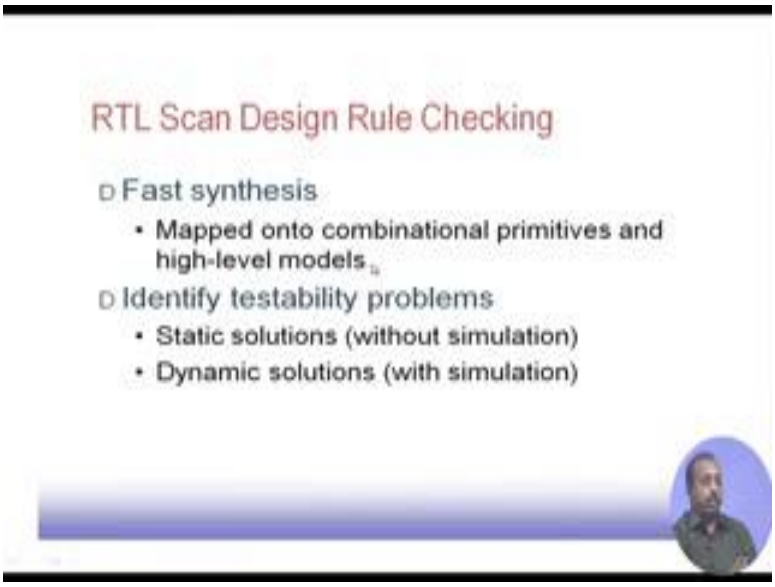


Design flow at RTL and gate level. So, the gate level testability repair design flow will be like this. So, it will take the RTL design it will go through a logic synthesis stage after logic synthesis will get the gate level design on that gate level design. So, there will be testability repairs that all those scan rules are there. So, they all those repairs will take place. So, this process will go on. So, once we are true we know that my design is tested it has been all the rules are all the rules are being followed then we get a testable design.

On this testable design, we do a logic scan synthesis that will be inside the scan chains and all that and then thus we get the scan design. On the other hand at the RTL level if you do so, start at the RTL design then we do a testability repairs. So, we see whether

there is any feedback loop and all that whether it is a gated clock at all that. So, all those things but those tests are the RTL level. So, we get the testable RTL design after we have got this testable RTL design we do the logic and scan synthesis. So, that is done and that way we get the scan design. So, testability modules are inserted at the RTL level itself.

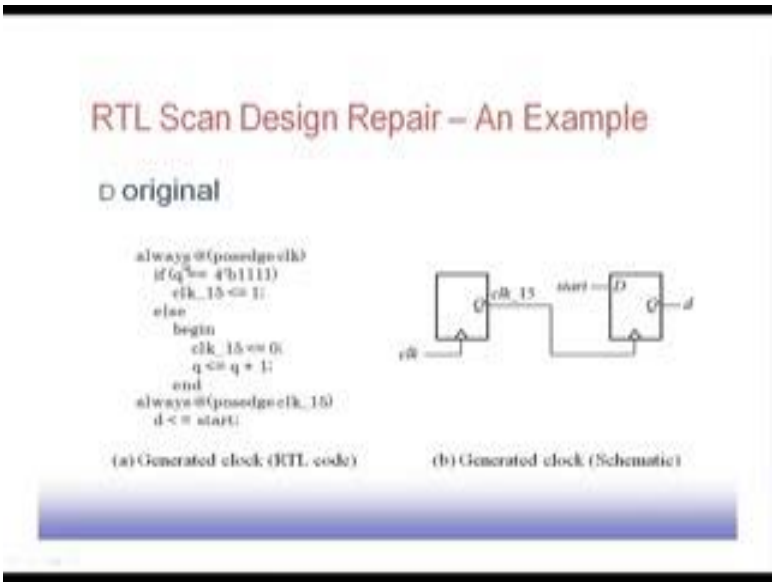
(Refer Slide Time: 10:36)



RTL, the scan design rule checking, for fast synthesis, they are mapped onto combinational primitives and high level models. So, we can since we want to do the synthesis process fast. So, we have got pre design module. So, we just map the functionalities on to those high level model and also combinational primitives that if previously designed modules are there we just put it on to that.

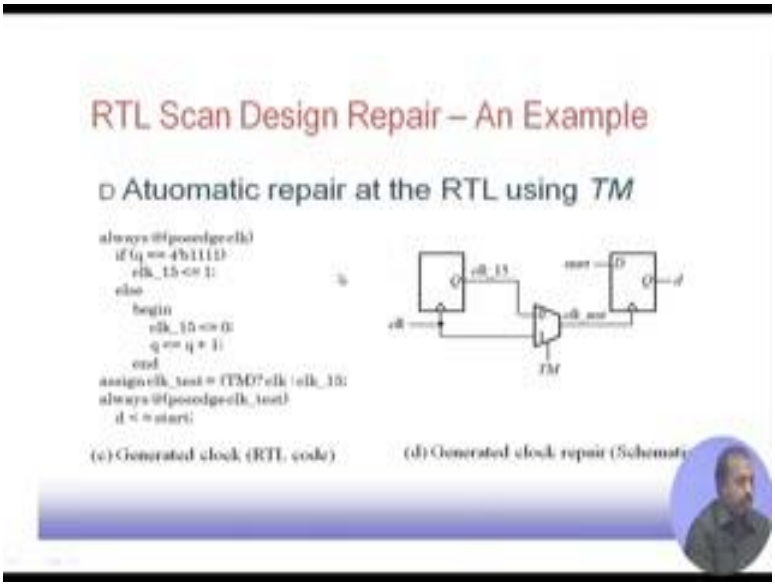
But there may be testability problems. So, I had to identify the testability problem we may we may have some static solutions that is we just analyse the behaviour we just check the connection between the modules and all that or there maybe dynamic solutions. So, there is simulate the system and try to see whether it is creating any problem or not.

(Refer Slide Time: 11:27)



This is one RTL level design. So, it says that at so at this is some set of very log code. So, at the positive edge of clock if Q happens to be all one then clock 15 equal to 1 else clock 15 equal to 0 and Q equal to Q plus 1. So, what is happening is that and at the edge of. So, this way it is actually counting the clock 15 values. So, after 15 clock pulses have been obtained. So, there will be this D will get the start value. So, this is exactly what is happening. So, this clock comes and this clock after the after this is this has become this clock 15 line has become high. So, it will be acting as clocks for this D flip flop and accordingly the start value will go to D.

(Refer Slide Time: 12:26)

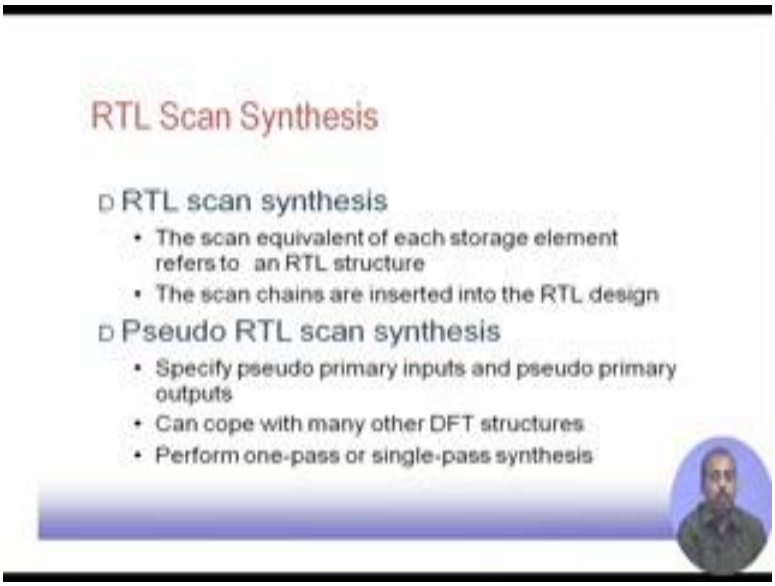


Now, the problem is that there is a asynchronous that that there will be some scan problem in this because now this in the test mode. What we do is that this if this clock; clock 15 line is coming here in the test mode. So, it will be selecting the test mode is 0. So, it will be selecting this clock 15 line from here. So, it will be going into this D flip flop.

On the other hand, if this mode is one then this clock goes directly to this one. So, in this case what is happening is that you see that these flip flop when this clock 15 is low clock 15 is not high. So, it is not getting any clock. So, we cannot put it a put it into a scan put it on to a scan chain because this will not be this is will this clock is different from this clock. So, we what we are doing we are trying to modify it. So, that under in a test mode the clock signal goes to both the flip flops, but in the normal operation mode of course, this clock 15 will be driving the clo.ck of this flip flop.

So, this is modified. So, what is happening at the RTL? Level itself the design is modified. So, we had this as the previously r; previous RTL code and now we have got this as the new RTL code. So, it says that assign clock test equal to. So, there is a condition check if TM signal is low in that case. So, it is if TM is high then it will be assigning clock. So, if TM is high then clock test is getting clock if TM is low then clock test is getting clock 15, this correction that we want to be there in the circuit. So, that is that in the behaviour in the in the RTL level code we have this line can be inserted. So, that is actually the RTL level corrections that is done by the RTL repair tool

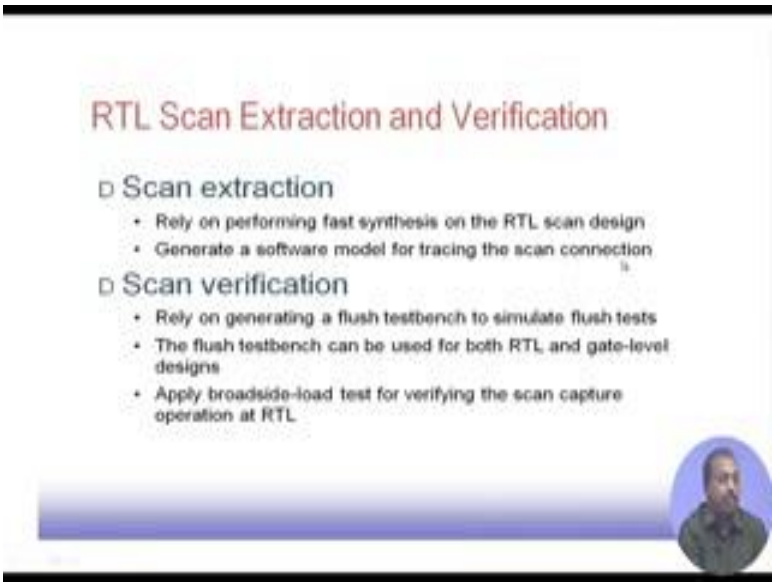
(Refer Slide Time: 14:32)



Then this RTL scan synthesis the scan equivalent of each storage element refers to an RTL structure the scan chains are inserted into the RTL design. So, for every storage RTL, structure will also have storage element. So, all those storage elements will be converted into the scan equivalents all the flip flops that are there in the RTL structure. So, there will be converted into scan flip flop all the latches will be converted in to scan latches. So, like that we will this RTL scan synthesis will convert all the storage elements into the RT into the scan structure.

On the other hand, this scan chains will get created via this storage elements. So, this is RTL scan synthesis and there is a pseudo RTL scan synthesis. So, this will specify the pseudo primary input and pseudo primary output and it can cope with many other DFT structures perform one per single pass synthesis. So, you will see that. So, what is pseudo you can specify some of the inputs to be pseudo primary inputs or pseudo primary outputs and only for those flip flops or those inputs outputs it will try to put it put them on to a scan. So, it is actually going towards that partial scan.

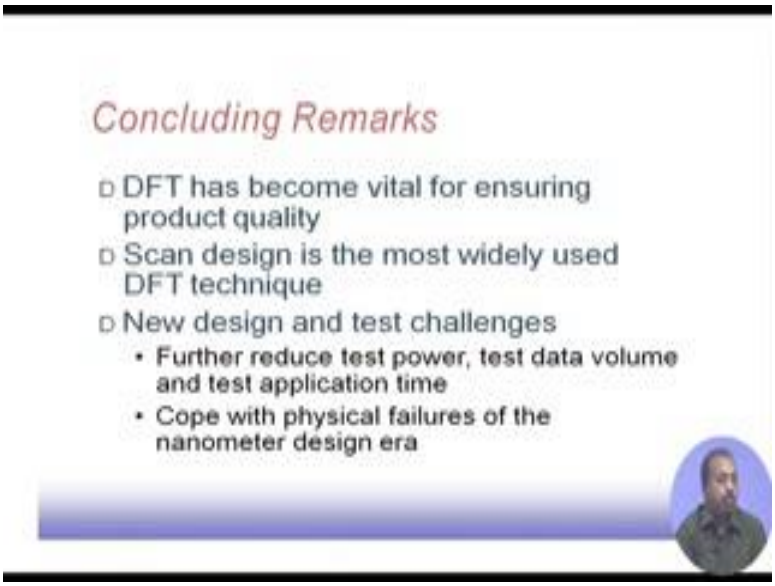
(Refer Slide Time: 15:49)



Scan extraction and verification. So, scan extraction it will rely on performing a fast synthesis on the RTL scan design and generate a software model for tracing the scan connection. So, again the same thing that is it is trying to at the RTL description level itself. So, it will try to see where at the scan lines where at the scan chains inserted. So, it is actually the lines that have been inserted into the description corresponding to that what will be the circuit that is synthesized and this scan verification. So, it will rely on generating a flush test bench to simulate the flush test flush test is basically checking all the scan chains; the scan chain. So, that they have been fabricated correctly or not. So, that is tested by this flush pattern. So, this flush test bench will be generated and to see whether this will be consist correct or not.

That this can be used for both RTL and gate level design because the this is the scan chains are actually test then this then we apply the broadside load test for verifying the scan capture operation. So, broadside load test is applying the test pattern and seeing whether it is getting the value properly or not we will come back to this when you go to the detail testing this BIST testing and all that. So, far scan capture operation of the RTL this can be applied.

(Refer Slide Time: 17:13)



DFT has become one of the vital tool for ensuring product quality scan design is the most widely used DFT technique because that is now more or less accepted that we have this scan chains are to be inserted. So, all these cad tools that we have, they are now supporting this DFT insertion technique new design and test challenges that are coming up for this DFT design for reducing test power consumption then this test data volume and test application time. So, what is happening is that as you the circuit is becoming more complex. So, we have got this test data volume also increasing significantly. So, storing all those test data onto ATE is a challenge.

That way we can try to do something by means of something on a test compression and all that so that this data volume is less and application time also needs to be reduced. So, application time can be reduced by using many other techniques like one is definitely reducing the number of test patterns another is by transferring the test patterns faster from ATE to the device. So, that way and of course, we need to cope with physical failures of the nanometre design era. So, more types of failures are occurring in the nanometre domain nanometre era. So, those for those defects are to be covered by the fault models and accordingly we have to come up with some DFT for those types of faults.

Thank you.