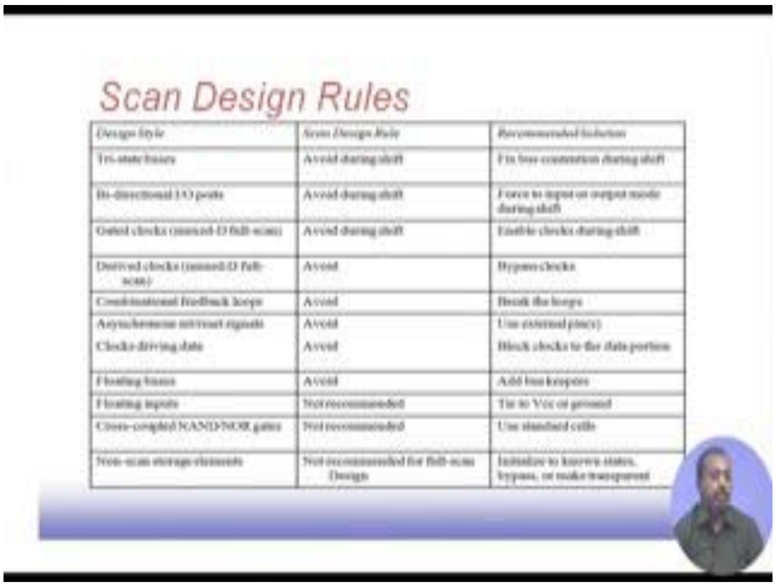
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 09
DFT (Contd.)

(Refer Slide Time: 00:21)



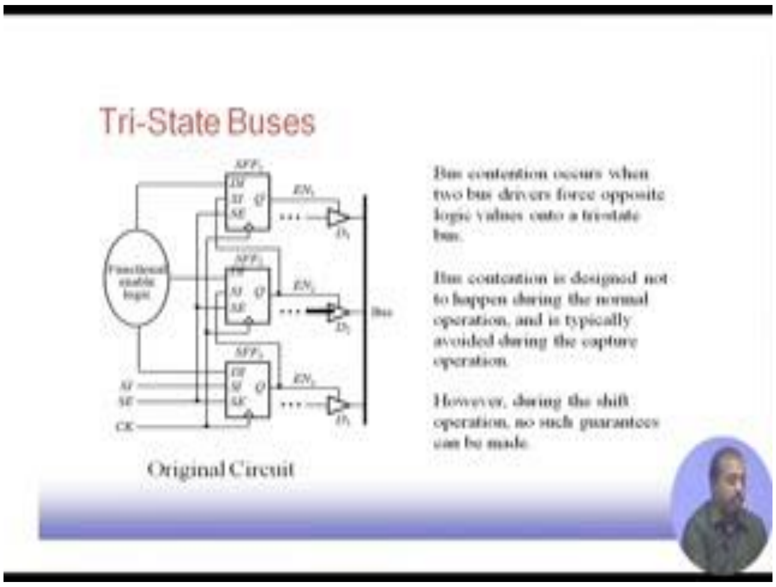
Scan Design Rules

Design Style	*Scan Design Rule*	*Recommended Solution*
Tri-state buses	Avoid during shift	Fix bus contention during shift
Bi-directional I/O ports	Avoid during shift	Force to input or output mode during shift
Gated clocks (muxed-D full-scan)	Avoid during shift	Enable clocks during shift
Derived clocks (muxed-D full-scan)	Avoid	Bypass clocks
Combinational feedback loops	Avoid	Break the loops
Asynchronous set/reset signals	Avoid	Use external pin(s)
Clocks driving data	Avoid	Block clocks to the data portion
Floating buses	Avoid	Add bus keepers
Floating inputs	Not recommended	Tie to Vcc or ground
Cross-coupled NAND/NOR gates	Not recommended	Use standard cells
Non-scan storage elements	Not recommended for full-scan Design	Initialize to known states, bypass, or make transparent

As far as scan design rules are concerned so there are several design styles and for them, for some of them, it creates difficulty while we are putting them were putting the scan on top of that and they are to be corrected or the design has to be modified like the tri-state buses. So, many of our designs; they have tri-state buses. So, scan design rules tells that is tri-state buses be avoided during shift operation so because during shift operation if some bus is tri-stated then the pattern will not shift through that bus. So, we have to fix this bus contention during shift. So, we have to ensure that there is no bus contention occurring during the shift operation.

Similarly bidirectional input output ports; so we have to force the input output mode during shifts, somehow you have to ensure that it they become unidirectional. So, that way there are many issues that will be coming which needs to be corrected in the scan design process and as I have already said that this cad tools are available which will do this thing. So, once you have given submitted design to the tool, so for this scan conversion so it will do that it will rectify all those problems and it will be coming up with a design which is correct.

(Refer Slide Time: 01:42)



Tri-State Buses

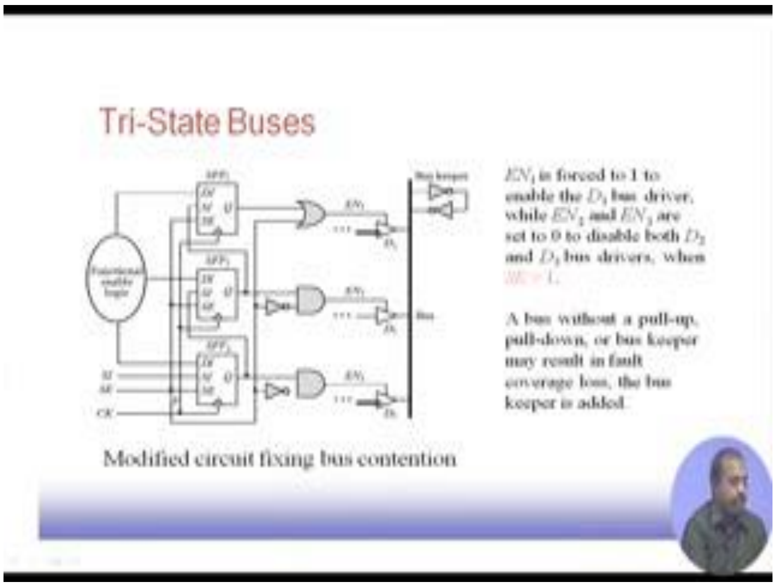
Bus contention occurs when two bus drivers force opposite logic values onto a tristate bus.

Bus contention is designed not to happen during the normal operation, and is typically avoided during the capture operation.

However, during the shift operation, no such guarantees can be made.

First one, the tri-state bus; so, here what was happened is that you see we have got a number of flip flops; number of scan flip flop is there and then they are; so in this there is a bus contention like this these are this D 1, D 2, D 3, etcetera. So, these are actually some tri-state buffers and from some logic this tri-state buffers. So, they are getting the values. So, maybe in the normal functional mode of operation, so something the designer has done so that these tri-state buffers will not be driven at the same time, but for the testing mode of operation. So, we cannot rely on what has been done at the design level.

(Refer Slide Time: 02:36)



Tri-State Buses

EN_1 is forced to 1 to enable the D_1 bus driver, while EN_2 and EN_3 are set to 0 to disable both D_2 and D_3 bus drivers, when $SE = 1$.

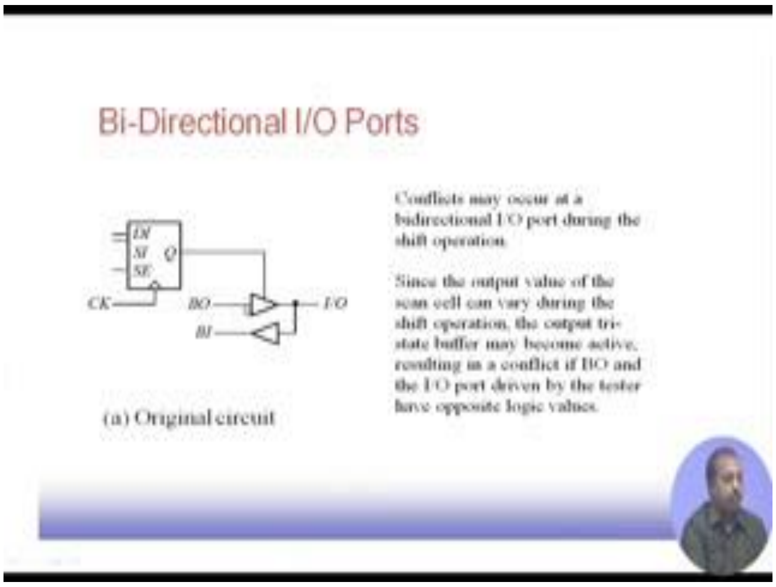
A bus without a pull-up, pull-down, or bus keeper may result in fault coverage loss, the bus keeper is added.

Modified circuit fixing bus contention

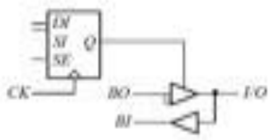
What is done is that for the testing mode, we need to modify. So, how do we modify? So, we modified it like this that in the scan enable mode so this for the scan when this A C line is; scan enable line is high. So, it is going there at the same time, this is scan enable line, it is also coming to this, they are connected to or here and via complemented version here. So, EN 1 is forced to one to enable the D 1 bus driver while EN 2 and EN 3 are set to 0 to disable D 2 and D 3. So, these 2 are disabled and this 1 is enabled. So, it ensures that during the shifting operation only this EN 1 line will be high. So, only this value will be available on the bus.

A bus without a pull up, pull down or bus keeper may result in fault coverage loss because so that is why sometimes in the test mode while going the test mode. So, we keep this bus keeper logic. So, it will be doing this, it will be keeping the voltage level is so this type of fix may be added. So, what we are doing? Out of so many enable lines that are there for this bus so we are in a; we are ensuring that only one of them will be enabled. So, rest are all disabled. So, this is done by means of doing this or and operation of this of this OR AND operation along with this enable lines.

(Refer Slide Time: 04:10)



Bi-Directional I/O Ports



(a) Original circuit

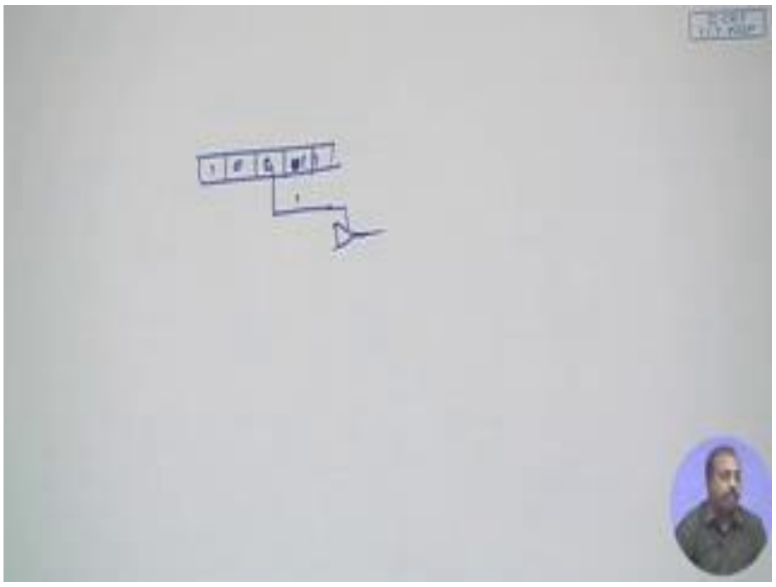
Conflicts may occur at a bidirectional I/O port during the shift operation.

Since the output value of the scan cell can vary during the shift operation, the output tri-state buffer may become active, resulting in a conflict if BO and the I/O port driven by the tester have opposite logic values.

Then there are bidirectional I O ports like say this is a see a structure where we have got this from an input output pin; bidirectional pin there are buffers. So, this buffer is going for putting the value to the buffered input, similarly the buffered output will go through this buffer and it will come to the output.

This is a very standard design when we have got bidirectional I O. So, now, the conflicts may occur at a bidirectional I O port during the shift operation. So, if you are trying to shift then the value that we are maybe we are just willing to put it on to the next flip flop, but that way it may come back again as the input the output value of the scan cell can vary during the shift operation the output tri-state buffer may become active. So, what is happening is that. So, this is a scan flip flop. So, there may be a chain of scan flip flops. So, and this Q output may be actually driving this buffer for the final test pattern it may so happened that this give output will not enable this, but in bit in between it may so happened that some pattern that is shifting while shifting. So, this buffer becomes active.

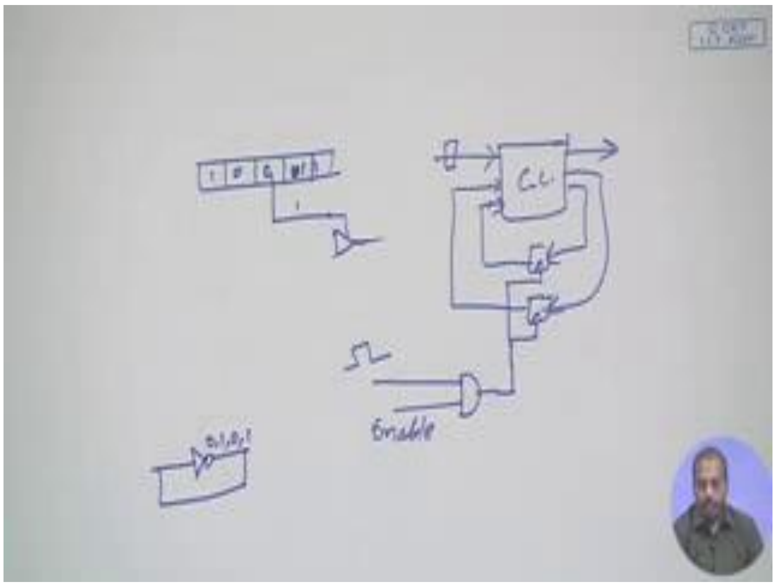
(Refer Slide Time: 05:25)



What we mean is this one that I have got a number of a scan cells. So, we are trying to set this scan chain to some value maybe 1 0 1 0 1. So, these value, but when it is shifting, so each of the cell; they are seeing the values of 1s and 0s maybe this bit is actually controlling that 1 enable line of a tri-state buffer; one enable line of a tri-state buffer. Now ideally this should be enabled, but or it may be this is 0 and said this is 1. So, ideally this should be disabled for the final operation, but while I am shifting the pattern through this chain, so sometimes it is getting the value 1 as a result this buffer is getting enabled.

That may create problem with the content of the circuit. So, output value of the scan cell can vary during the shift operation, the output tri-state buffer may become active resulting in a conflict if this buffered output and the I O port driven by the tester have opposite logic values. So, tester has given sum value here for shifting into the system and this B O line is something else computed by the combinational logic. So, if these 2 values differ then there is a conflict at this point. So, that has to be avoided.

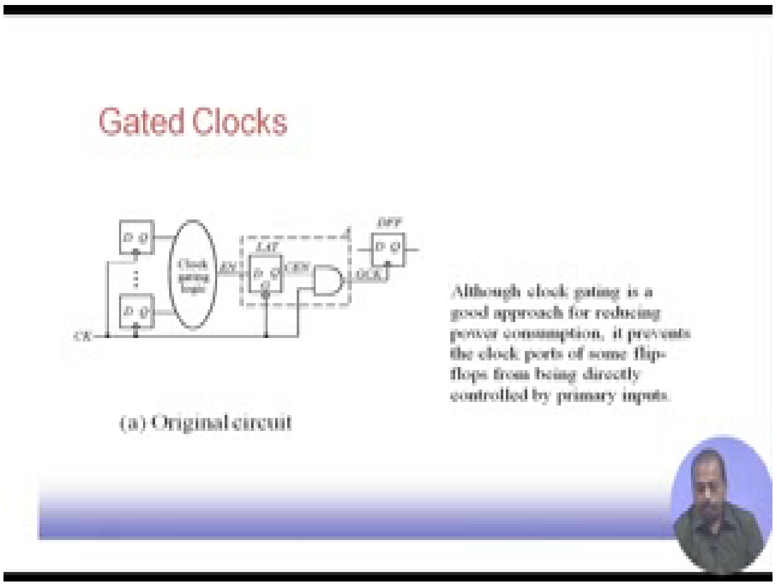
(Refer Slide Time: 06:47)



How to do this thing? So, for doing this, so this buffer has to be disabled. So, how do you do this? When SE line is 1, here I am getting a 0 so that whatever be this Q value so this and gate output is 0 as a result is buffer is disabled. So from this buffered output, so it cannot conflict with this I O bit value so that way it cannot where get affected. So, whatever value that test are puts on this I O lines so that will come through this B I line. So, during the capture operation, the applied test vector will determine whether a bidirectional I O is used as input and output and then it will and controls the tester appropriately.

So, that is the scan shifting time for or scan enable time, but when it is actually the capture cycle then of course, this is 0. So, this is 1. So, it depends on this Q setting and that is actually decided by the combinational logic coming through this D I input this Q is set. So, this Q is 1 then this buffer may be enabled as a result whatever is the B O value will come to the I O and that that in turn will go to the tester that will go for as the response pattern. So, that can happen.

(Refer Slide Time: 07:59)



Gated Clocks

Although clock gating is a good approach for reducing power consumption, it prevents the clock ports of some flip-flops from being directly controlled by primary inputs.

(a) Original circuit

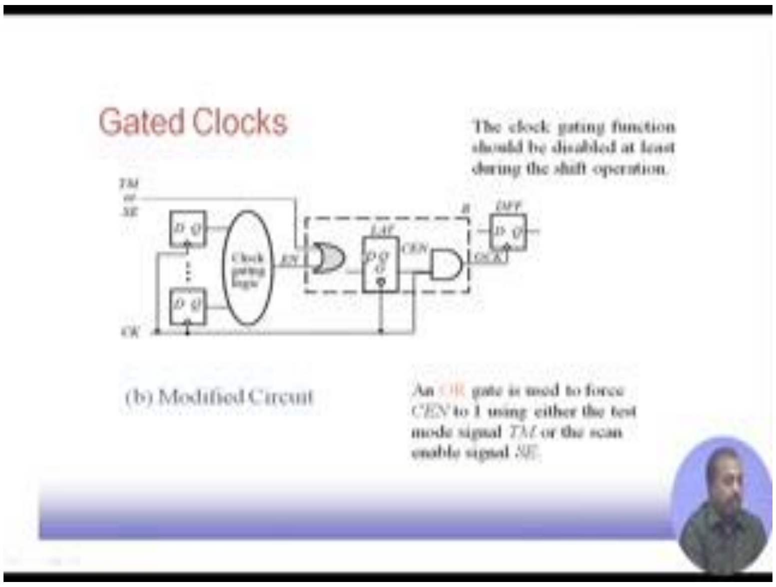
Another very important a design style that we have is the clock gating circuits. So, clock gating is done so that we want some portion of the circuit to become inactive for power consumption point of view. So, this is a good approach for reducing power consumption because if we want that say in a typical sequential circuit design as I have said that there is a combinational logic and there are flip flops; there are flip flops which are feeding this combinational logic, now if we want that if we find that for some time this system should not work.

It should not work means they are its output is not that much that necessary. So, what we can do the clock signal that we have so we can disable this clock signal. We can disable this clock signal so that this flip flops will not change their values and if they do not change the value, this combinational logic will also not get any change in this pseudo primary input and possibly it will not change the pseudo primary output as well. You can also have some sort of latch here to ensure that the primary inputs are also not changing so that this combinational logic becomes totally idle.

Now this is known as clock getting. So, to do this; to do is clock gating, what you have to do is put some sort of a NAND gate. So, here I have got the actual clock signal and I have got the some enable line here. So, this enable line when enable is equal to 1 then the clock will go, if enable is equal to 0 then the clock will not go. So, what happens is that in case; so I have got this outputs driving this clock, getting logic clock, getting logic generate this enable signal and this enable signal is latched here and then this is ended with the clock and this is a finally generates the clock enable.

This clock enable line is generated and that is the gated clock given to the final D flip flop which if wire, we should have connected the clock line. So, this is a popular approach for reducing power consumption at as it prevents the clock ports or 2 of some flip flops from being directly controlled by the primary input.

(Refer Slide Time: 10:22)


Gated Clocks

The clock gating function should be disabled at least during the shift operation.

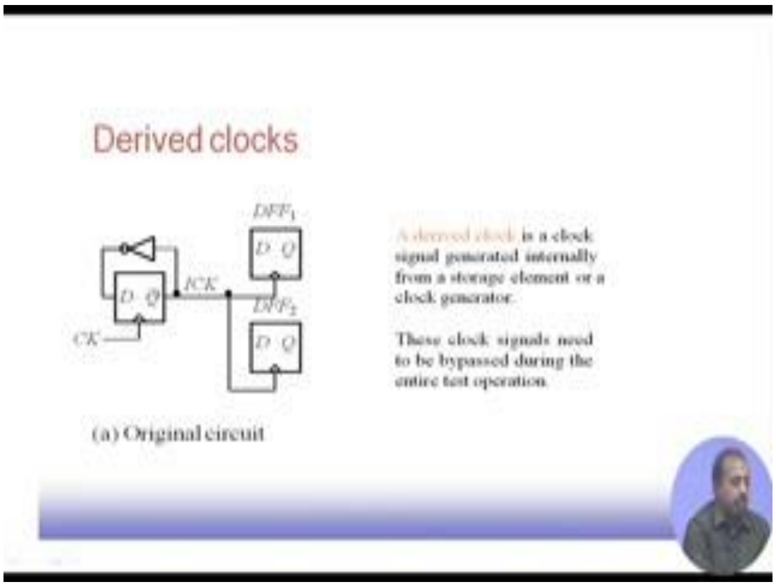
(b) Modified Circuit

An OR gate is used to force *CEN* to 1 using either the test mode signal *TM* or the scan enable signal *SE*.

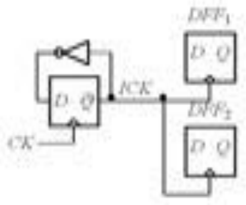
However it has got difficulty because during shift operation, so we there may be problem. So, they are what happens is that the clock gating function should be disabled at least during the shift operation. So, how do we do this thing? So, this test mode or this shift enable mode, so at that time so there or this enable signal. So, in these cases, so this or gate output is connected to this D input so that this latch value will be coming here and then this or gate is used to force this CEN, this clock enable value to 1.

This in TM in test mode or in a shift enable mode. So, this or gate output will be 1, as a result, this CEN output of this flip flop will definitely be 1. So, this and gate will and so that clock enable line is high. So, this flip flop will always get the clock if this TM or SE is high. So, in the test mode whatever this clock gating logic computes so that is getting over written by this TM or SE signals so that way the gated clock problem can be resolved.

(Refer Slide Time: 11:37)



Derived clocks



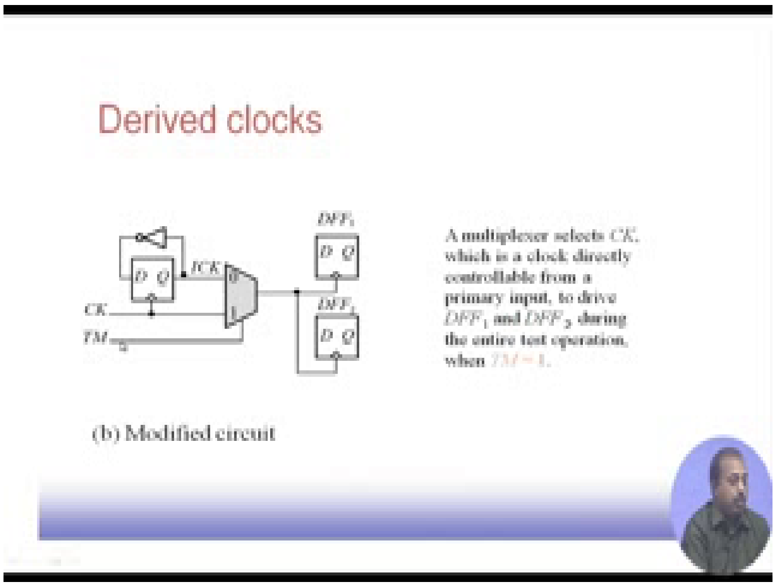
(a) Original circuit

A derived clock is a clock signal generated internally from a storage element or a clock generator.

These clock signals need to be bypassed during the entire test operation.

Sometimes we have got derived clock. So, derived clock means in from 1 clock signal, we derived some other clock signal internally. So, like this, so I have got 1 clock CK and I have put an inverter and connected to D flip flop. So, that this clock will be divided by 2 I think, so that it will be generating; the D is continually whenever D is 1. So, this is this will become 1 then this is 0. So, this is then this will become 0, but that will happen at the half the clock gate that we have here. So, this it is a clock divider type of circuit and then this derived clock is given to the D flip flop. So, this is the derived clock circuit, again in the problem is that now this shifting pattern through the through this all this flip flops when they converted in the scan flip flops the shifting pattern becomes different for different portions.

(Refer Slide Time: 12:37)

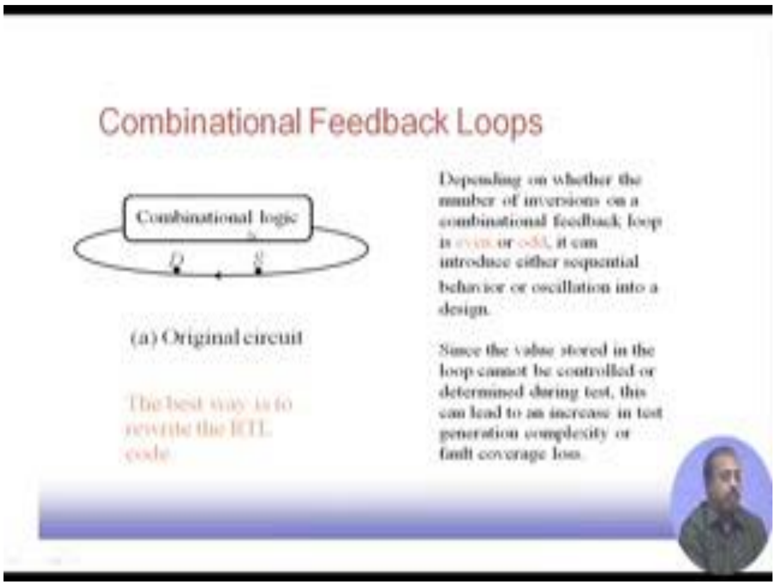

Derived clocks

A multiplexer selects CK, which is a clock directly controllable from a primary input, to drive DFF_1 and DFF_2 during the entire test operation, when $TM = 1$.

(b) Modified circuit

We have to bypass this. So, what is done? So, we put a multiplexer here. So, this for the normal operation of the circuit, this test mode is off as a result whatever is the derived clocks so that will go to the flip flop, but in case of test mode, if test mode is on then this clock value that is coming so that will go directly into this D flip flop. So, this derived clock does not have any meaning here. So, this is required otherwise this flip flop we cannot convert to scan flip flop so because they will be driven by a different clock in that case. So, this derived clock is another obstacle for the scan shifting and that is resolved by introducing this type of modification.

(Refer Slide Time: 13:24)


Combinational Feedback Loops

(a) Original circuit

The best way is to rewrite the RTL code

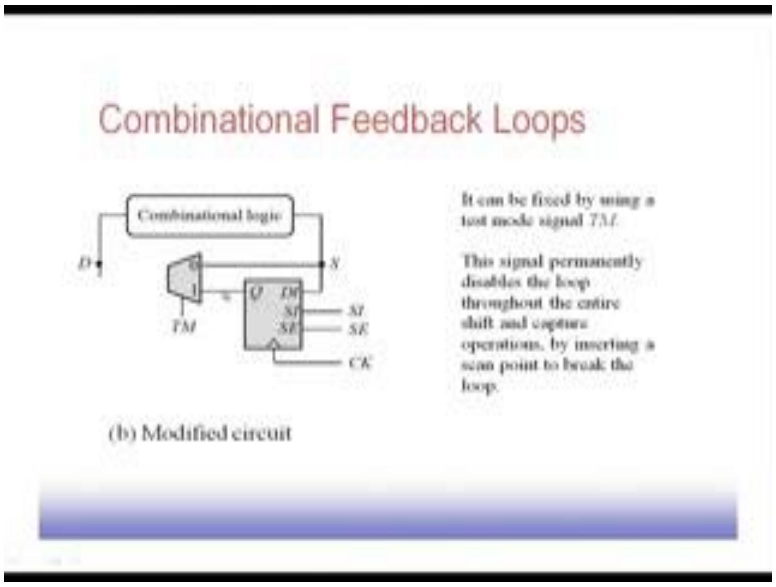
Depending on whether the number of inversions on a combinational feedback loop is even or odd, it can introduce either sequential behavior or oscillation into a design.

Since the value stored in the loop cannot be controlled or determined during test, this can lead to an increase in test generation complexity or fault coverage loss.

Then there may be combinational feedback loop. So here what was happen is that I have got this combinational logic, its output is against feeding the input. So, that way there may be some combinational feedback loop. So, depending on whether the number of inversions on a combinational feedback loop is even or odd, it can introduce either sequential behavior or oscillation into a design. So, you can count the number of inverters. So, naturally if I have got a situation like this, there is a single inverter then if this inverter is initially 0 then it will after 1 clocks I after sometime depending upon the delay of the inverter, it will become 1, again after the delay, it will become 0, after sometime it will become 1. So, that way it becomes an oscillatory behavior and similarly if it is even then it will show that if this will be where this will have some sequential behavior.

That way we have got different undesirable behavior of this combinational logic. So, the best way to this dissolve this combinational feedback loop is to rewrite the code and we have got a better design that does not have this combinational feedback loops because the loop has the value stored in the loop cannot be controlled or determined during test. So, this can lead to an increase in the test generation complexity or fault coverage loss.

(Refer Slide Time: 14:58)



Combinational Feedback Loops

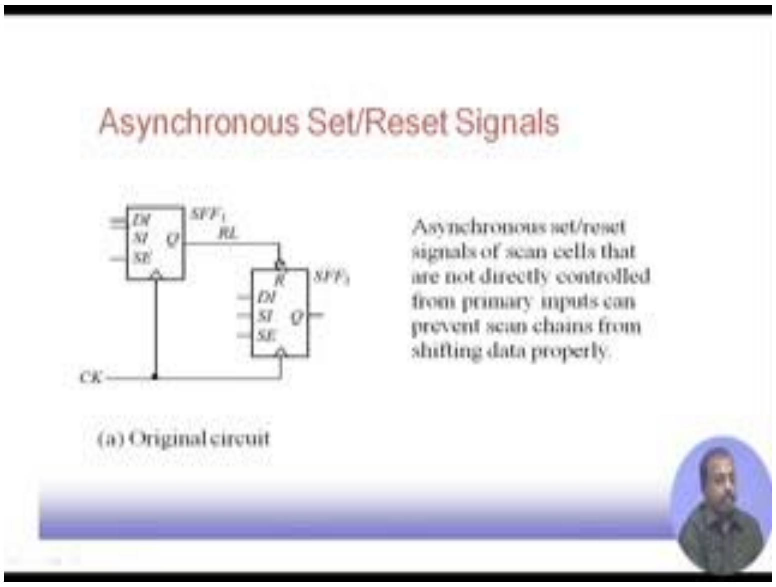
(b) Modified circuit

It can be fixed by using a test mode signal *TM*.

This signal permanently disables the loop throughout the entire shift and capture operations, by inserting a scan point to break the loop.

How to solve this problem? So, for solving this problem what we can do? We can in the test mode this TM signal, so this signal permanent is a. So, this signal when TM is equal to it, it will disable the loop throughout the entire shift operation shift and capture operation by inserting a scan point to break the loop so, this point is actually observable point. So, this is going from when this clock signal; when this TM line is high then this the point becomes visible and this otherwise this D point that we have so that is actually coming to this DI data input line. So, we can we can see the value of this Q point through this.

(Refer Slide Time: 15:51)



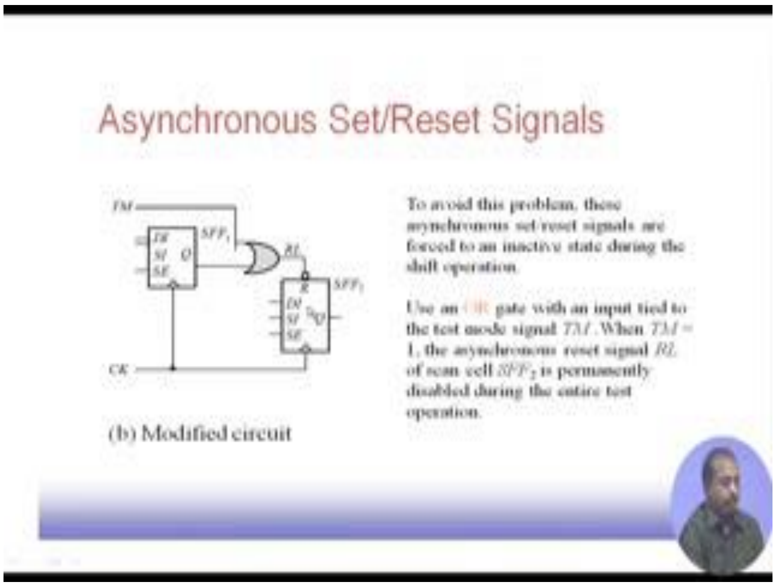
Asynchronous Set/Reset Signals

Asynchronous set/reset signals of scan cells that are not directly controlled from primary inputs can prevent scan chains from shifting data properly

(a) Original circuit

Asynchronous set reset signals like if you have asynchronous set, reset signals then that creates difficulty because this flip flops; they made may become reset arbitrarily. So, in the original circuit was an; what has happened is that whenever this Q value becomes 0. So it resets this particular latch. So, this RL signal is generated. So, this is latch gates disabled this latch gates reset. So, this has to be prevented because they are not control directly from the primary inputs. So, you have to prevent this type of resetting because when the patterns are shifting through this latches it may, so happen that some intermediary point this value becomes equal to 0, as a result this latch gets reset. So, that has to be avoided.

(Refer Slide Time: 16:44)



Asynchronous Set/Reset Signals

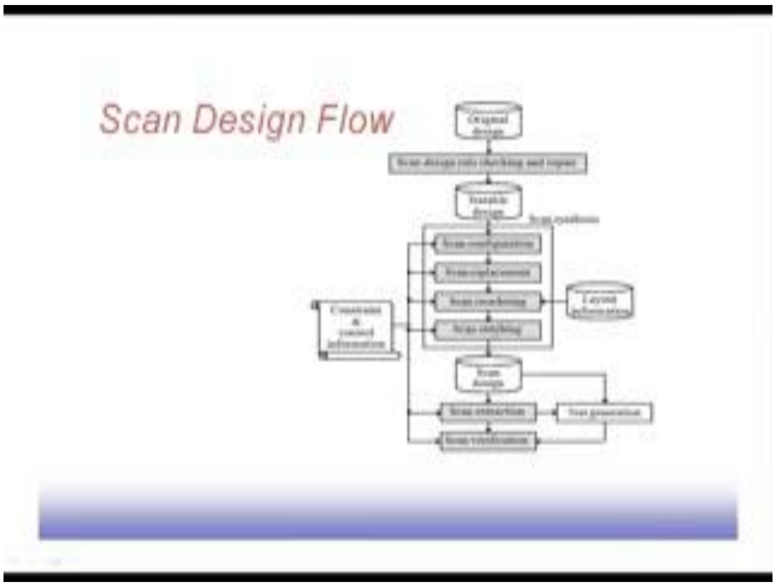
(b) Modified circuit

To avoid this problem, these asynchronous set/reset signals are forced to an inactive state during the shift operation.

Use an OR gate with an input tied to the test mode signal TM. When TM = 1, the asynchronous reset signal RL of scan cell SFF_2 is permanently disabled during the entire test operation.

And for doing this thing, again the way we have to do it is that this in under the test mode this TM line; this R L is put through an OR gates so this Q output and this TM so they will be odd here and then it will go to this thing.

(Refer Slide Time: 17:18)

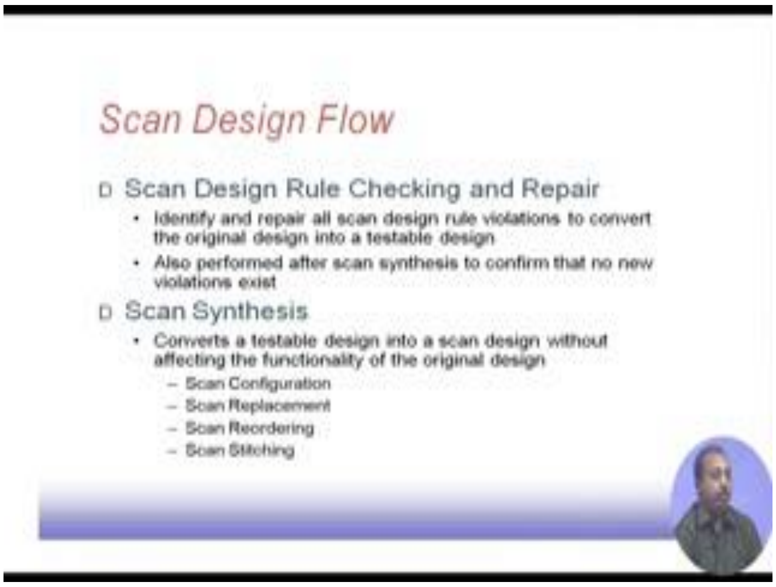


Scan Design Flow

If the TM is high then this RL will always be 1 as a result, this latch will never be reset as long as the test mode is on so that way it can avoid resetting of this latch 2. Now in the scan design flow, how this has to be done like we have seen that there are several scan D circuits, several ways of converting flip flops to scan flip flops, we have seen that there are some set of design rules for that now how they are taken care of in the whole design process. So, this original design so it goes first, it goes to a scan design rules checking and repair. So, this actually corrects all those problems like say that that clock gating or the asynchronous reset. So, all those are taken care of and the design is modified by this rule checking and repair parts so that gives raise this testable design.

Once I have got this testable design, it goes into a face which is known as scan synthesis, in the scan synthesis there are several actual parts like scan configuration replacement reordering and stitching. So, at the reordering stage, we need this layout information and then this also we need this constraint and control information. So, they are added into this one into this scan synthesis process that gives us the scan design and after the scan design has been done, the scan is extracted and the test generation task is also started. So, test generation tests are generated scan chains are extracted and the scan verification takes place and this process continues again. So, that if there is some flow then some problem is occurring then again this scan synthesis part is repeated.

(Refer Slide Time: 18:58)



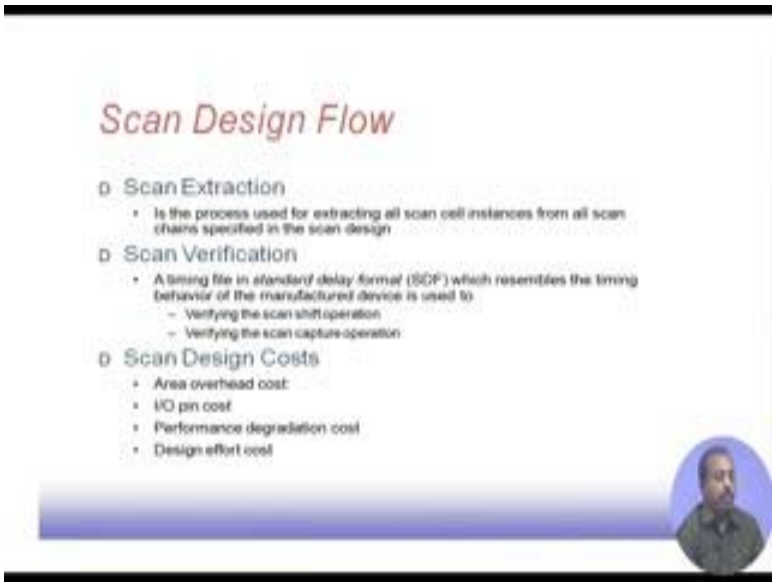
Scan Design Flow

- Scan Design Rule Checking and Repair
  - Identify and repair all scan design rule violations to convert the original design into a testable design
  - Also performed after scan synthesis to confirm that no new violations exist
- Scan Synthesis
  - Converts a testable design into a scan design without affecting the functionality of the original design
    - Scan Configuration
    - Scan Replacement
    - Scan Reordering
    - Scan Stitching

We will see; we will look into this design flow. So, first part of this design flow is the scan design rule checking and repair. So, this is for identifying and repairing all scan design rule violations to convert the original design into a testable design. So, whatever design rules we have seen, the scan design rules we have seen; so they are checked and if there is a violation then they are corrected for making the design test table and it is performed after scan synthesis to confirm that no new violations have occurred. So, after the modification has been done. So, it is it should be the case that no new violation is occurred. So, again the rule checking has to be done in the scan synthesis. So, it will convert the testable design into a scan design without affecting functionality of the original design.

It will be converted it into scan design. So, the flip flops will be converted to scan flip flops, the scan stitching and all that. So, these are the various stages in it is scan configuration replacement reordering and stitching.

(Refer Slide Time: 20:06)



Scan Design Flow

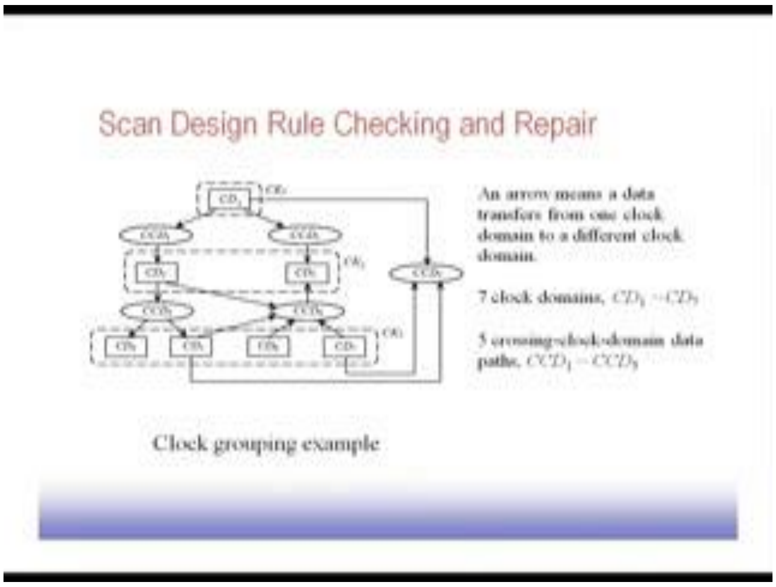
- Scan Extraction
  - Is the process used for extracting all scan cell instances from all scan chains specified in the scan design
- Scan Verification
  - A timing file in *standard delay format* (SDF) which resembles the timing behavior of the manufactured device is used to
    - Verifying the scan shift operation
    - Verifying the scan capture operation
- Scan Design Costs
  - Area overhead cost
  - I/O pin cost
  - Performance degradation cost
  - Design effort cost

Then we have got scan extraction, it is the process used for extracting all scan cell instances from all scan chains specified in the scan design. So, this is this will find out all the scan cells that have all the flip flow that had been converted into scan cells, all the latches that have been converted into scan cells and when which in they are existing then there comes the scan verification. So, this is a timing file in standard delay format which resembles the timing behavior of manufacture device is used to verify the scan shift operation and scan capture operation whether those 2 operation delays are coming in a proper timing or not. So, this standard delay format files. So, they will have the timing information.

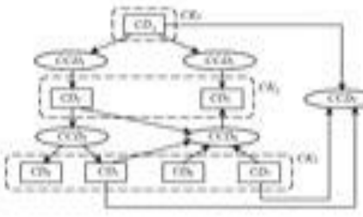
What are the costs of this scan design? Definitely area overhead is one thing like we have converted many of them into scan cells and all that. So, that is there then there is IO pin cost into IO pin cost. So, this is basically the extra pins that are needed like this scan in scan out lines then the clock SC case scan enable test mode all these extra pins that we are talking about. So, they are to be you have to pay for that. So, that their pin cost is there plus this clock distribution of this scan clock distribution is another issue. So, that has to be taken care of then the performance degradation cost. So, performance gets degraded because of the reason that some extra circuitry has been added so that may come into the critical path of the design. So, delay will increase.

Design effort cost that is the extra amount of man hours and the efforts that need to be put. So, that this design testing this scan insertion is done properly and testing can be carried out. So, that is another issue.

(Refer Slide Time: 22:05)



Scan Design Rule Checking and Repair



An arrow means a data transfers from one clock domain to a different clock domain.

7 clock domains, $CD_1 - CD_7$

5 crossing-clock-domain data paths, $CCD_1 - CCD_5$

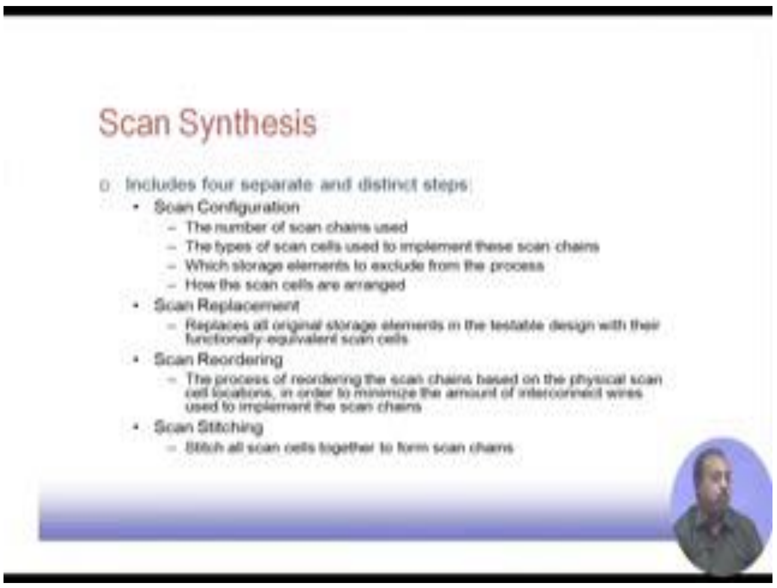
Clock grouping example

What sort of scan design rule checking and repair can be done like you see that it may so happen that in my circuit, there are several clock domains like this CD 1, CD 2, CD 3, CD 4, CD 5. So, they are different clock domain. So, what do we mean is that all modules in the particular region, they are driven by the same clock.

Now it may so happen that from clock domain 1 to clock domain 2, there is a signal which is being passed. So, they are called crossing clock domain data path like this CD 1. So, there is a path from CD 1 to CD 2 in the circuit. So, that is actually crossing the clock domain similarly here CD 1 to CD 3. So, there is a crossing clock domain, but from CD 2 to CD 3, there is no crossing. Now if there is no crossing so what we can do is that now we are trying to assign some clock values to this individual clock domain. So, CD 1; CD 1 and CD 2, they cannot be assigned the same clock because they are from 2 different clock domains and they are coming at there is some age which is called there is some data that will be coming from one clock domain to another clock domain.

We cannot group these 2 clocks together. So, I have to give separate clocks for CD 1 and CD 2. So, that is done, but between CD 2 and CD 3. So, there is no such data transfer. So, we can put it; put them on to the same clock. So, basically when the circuit is designed at that time, we may find that they are different clock domain, but having. So, many different clocks in the in the system may be problematic. So, they routing of all those clocks are difficult. So, we may like to group some of these clock domains into a single flop like CD 2 and CD 3, since they are giving data in the say they are not interacting with each other. So, they can be given in the same clock similarly CD 4, 5, 6 and 7, they can be given a surprising a single clock. So, ultimately this 7 clock domains that we have so they can be clubbed into or they can be merged into 3 such clock groups CK 1, CK 2, and CK 3. So, this design rule, it may check this thing that it may find that if there is any violation or not.

(Refer Slide Time: 24:35)



Scan Synthesis

- Includes four separate and distinct steps
  - Scan Configuration
    - The number of scan chains used
    - The types of scan cells used to implement these scan chains
    - Which storage elements to exclude from the process
    - How the scan cells are arranged
  - Scan Replacement
    - Replaces all original storage elements in the testable design with their functionally equivalent scan cells
  - Scan Reordering
    - The process of reordering the scan chains based on the physical scan cell locations, in order to minimize the amount of interconnect wires used to implement the scan chains
  - Scan Stitching
    - Stitch all scan cells together to form scan chains

Then comes scan synthesis, the first part of this scan synthesis is scan configuration. So, this will be the first input that we have is the number of scan chains used. So, if I have got a single chain then what will happen is that all the flip flops that I have. So, they will be on that chain. So, controlling individual flip flops in that chain becomes difficult. So, that way it is a bit cumbersome the time to load the scan chain will be high. So, that has to be a taken care of. So, it may be that if I am allowed to provide multiple scan in lines multiple scan out lines etcetera then I can possibly go for multiple scan chains where the individual scan chains will be small is smaller in length as a result the loading and unloading will be faster.

Number of scan chains that used so that is one important issue, second thing is that the type of scan cells used to implement this chains, so we have seen that there are several types of scan cells like the marked D then this 2 clock then we have got LSSD. So, like that we have got several types of scan cells. So, which scan cells we are going to use for implementing this scan cells then with storage elements to exclude from the process. So, as we have; say there is a partial scan design. So, which one are not to be excluded which one are not to be count to considered etcetera that is one thing and how the scan cells are arranged. So, what is the order in which this scan cells will be put then the scan replacement it will replace all storage elements in the testable design with their functional equivalent scan cells.

Once we have decided to modify some of the storage elements to scan cells, so this can replacement process will modify all those storage element to the scan versions then we do a scan reordering. So, reordering the scan chains based on the physicals cell locations so if because between 2 successive scan cells, I should have a connection. So, if the 2 cells are located far away then the chain that is that the where that will run will be long and if there are. So, I need to find a proper path for all these scan cells to be put on to a chain so that the length of this interconnects is the minimum. So, for that purpose I may need to reorder the cell numbers. So, reorders the cells that have been identified. So, that I can minimize the amount of interconnect wires and finally, scan stitching. So, had to stitch all those scan cells together to form scan chains. So, the output of one scan cell should go to the input of another scan cell like that. So, those connections are to be established. So, that is actually the scans reaching.

So, we will continue in the next class.