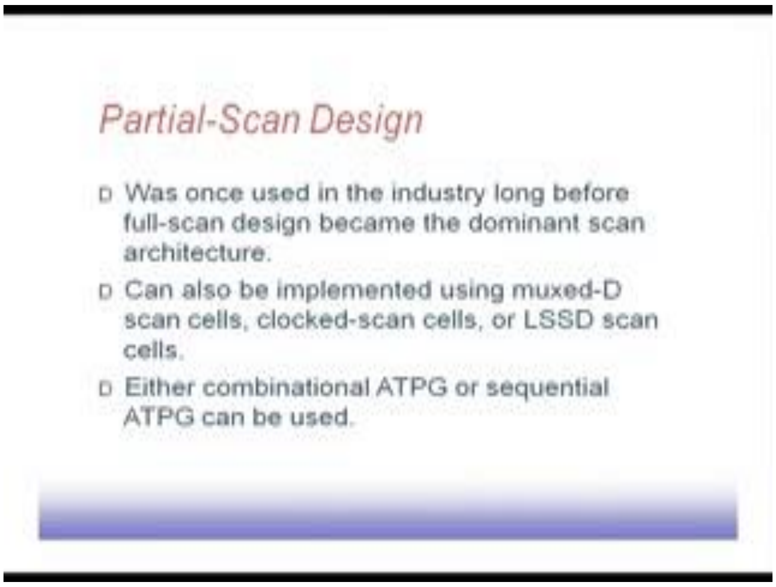
Digital VLSI Testing

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

Lecture - 08
DFT (Contd.)

(Refer Slide Time: 00:21)



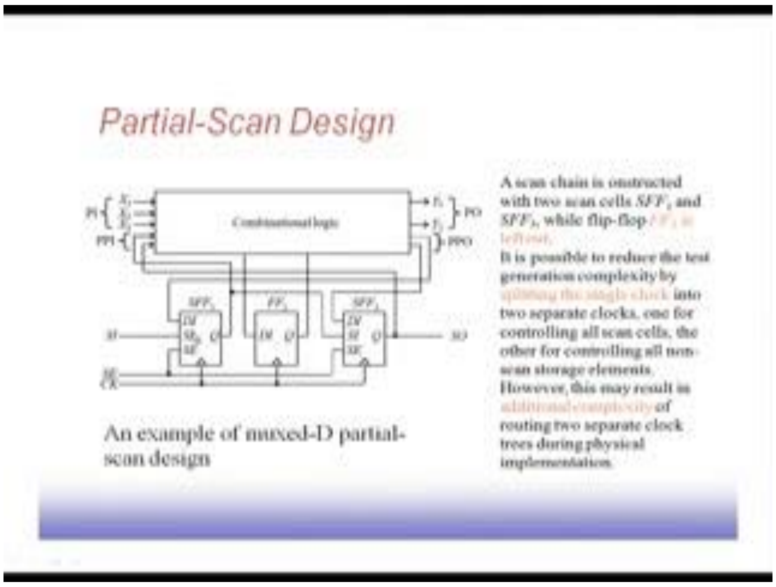
Partial-Scan Design

- Was once used in the industry long before full-scan design became the dominant scan architecture.
- Can also be implemented using muxed-D scan cells, clocked-scan cells, or LSSD scan cells.
- Either combinational ATPG or sequential ATPG can be used.

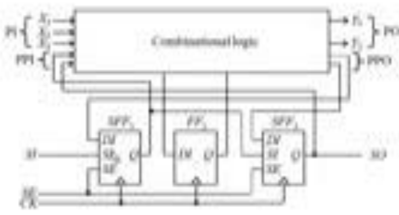
So partial scan design, so this was once used in the industry long before full scan design and it became the dominance can architecture because as I was telling that there are some ad hoc design styles. So, some designer they found that if you put some of the flip flops on to on some access path then it is easy to do the testing. So, as a result some partial scan architecture was staying there. So, that continued and after this full scan came into picture even then, this partial scan in many cases they are used. They can be also be implemented using this Muxed D scan flip flops, clocked scan flip flop or LSSD scan cells. So that does not make any difference.

Now, what happens is that some of the flip flops are converted or some of the latches are converted into their scan counterpart. So, we can use combinational ATPG or sequential ATPG. So, combinational ATPG, if we use then of course, we are losing on the fault coverage because it may not be able to generate tests for the sequential, for the portions which are driven by those flip flops or latches. But, in many cases it is seen that many faults are can be tested by applying random patterns as well. So, the combinational patterns dedicated for testing combinational faults, they may also become sufficient for testing these types of faults. So, that way many of them will get detected. So, or we can use a sequential ATPG which, is more complex for getting this type of test patterns.

(Refer Slide Time: 02:02)



Partial-Scan Design



An example of muxed-D partial-scan design

A scan chain is onstructed with two scan cells SFF_1 and SFF_3, while flip-flop FF_2 is leftout.

It is possible to reduce the test generation complexity by splitting the single clock into two separate clocks, one for controlling all scan cells, the other for controlling all non-scan storage elements.

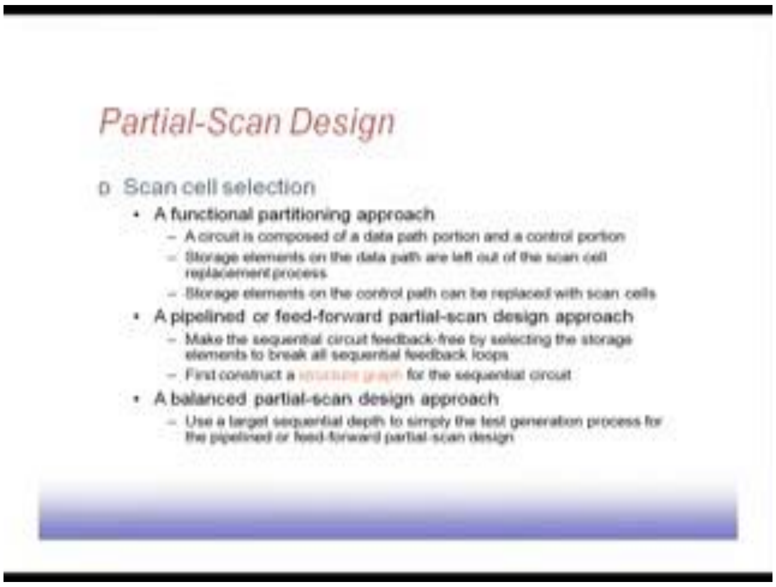
However, this may result in additional complexity of routing two separate clock trees during physical implementation.

So, this is a case where we have got a Muxed D scan, Muxed D based partial scan design. So, here you see that in the design we have got three flip flops; F F 1, 2 and 3. Out of them this first flip flop and third flip flops. So, they are converted into additional they are converted into scan flip flops whereas, flip flop 2 is not converted into a scan flip flop. So, this is some type of partial scan design. So, this flip flop S F F 1 and S F F 3 being on the, being scan flip flop. So, they are actually connected in the chain from SI to Q of this first flip flop to the SI of second third flip flop and then going to the Q to SO. So, that is possible, but flip flop 2 is left out.

So, it is possible to reduce test generation complexity by splitting the single clock into two separate clocks; one for controlling all scans the, other for controlling all non scans storage elements. So, this is actually a suggestion for these ATPG algorithms. So, what it says is that, you divide the whole design into two parts; one part is having where we have got this scan clock driving the circuit and another clock and another case normal clock driving the circuit. So, as a result we get two different portions also, two different portions are triggered by those clocks and as we can have this test pattern generator, generating faults; generating test pattern. In the first case for the scan chain based, scan flip flop based portion. So, there are it is normal combinational ATPG and for other flip flop the normal clock. So, that becomes a sequential ATPG, but the amount of job given to the sequential ATPG is less because only a portion of the circuit will be given to it, as a result it may do a good job.

So; however, there are problems because now we have to, if there are two separate clocks like in this particular case we have not shown two separate clocks. The same clock is given to both the all the flip flops, but if there is a separate clock then this routing of the separate clock also is an issue like I have already said that this clock distribution in the for the chip itself is a problem. How to you distribute the clock? But, if now, if you have got more number of clocks then, the distribution becomes more complex.

(Refer Slide Time: 04:41)



Partial-Scan Design

- Scan cell selection
  - A functional partitioning approach
    - A circuit is composed of a data path portion and a control portion
    - Storage elements on the data path are left out of the scan cell replacement process
    - Storage elements on the control path can be replaced with scan cells
  - A pipelined or feed-forward partial-scan design approach
    - Make the sequential circuit feedback-free by selecting the storage elements to break all sequential feedback loops
    - First construct a structure graph for the sequential circuit
  - A balanced partial-scan design approach
    - Use a target sequential depth to simply the test generation process for the pipelined or feed-forward partial-scan design

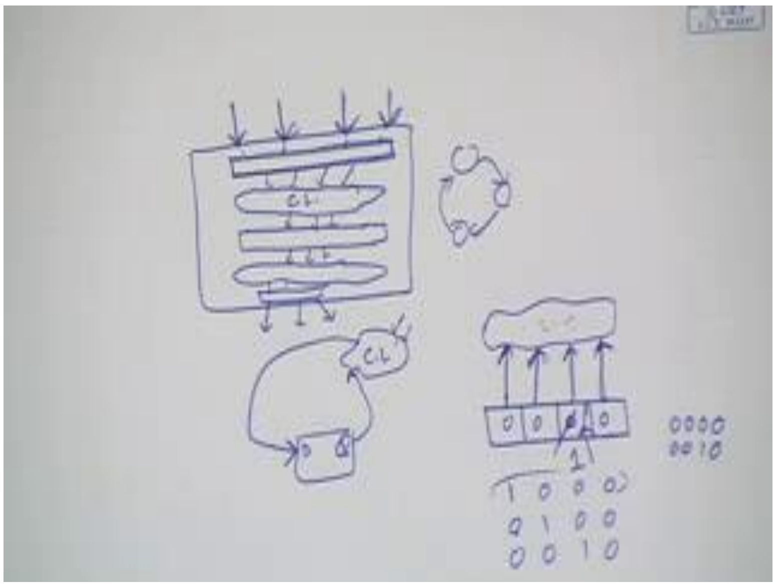
Now, the question is that in my design that maybe number of flip flops or number of latches. Now how do you select those sequential element that will be that must be put on to scan cell or that must be converted into scan cells. So, first approach is to have a functional partitioning approach. So, a circuit is composed of a data path portion and the control portion. So, data path portion is actually doing all the computations like there are registers, multipliers, multiplexers, adders, subtractors. So, like that. So, they are actually forming the data path.

Now, for controlling this data path we need to activate several signals like; first maybe flip flops, first the registered will get the value from the outside world. There in that may be done in the first clock cycle, in the second clock cycle maybe the adder will do the addition taking the values from some registers and then writing the result back onto another register. So, that way those for this input registers I need to enable the, I need to make their enable line high. So, that the values are available and for the destination register I must have the load signal given to it. So, and for the adder module I need to give the add signal.

So, these are done by the control part. So, what happens is that the storage elements on the data path they are left out of the scan cell replacement process because data path, flip flop they normally form registers. So, those registers are not that much necessary to test, but the control path the flip flop they are actually forming the finite state machine. So, that part needs to be tested and this control part will actually excite the data path. So, in some sense I can say that, if I have this control part flip flop setting easy then the test pattern generation may become simpler.

So, that is why the storage elements of the control path may be replaced by scan cells whereas, storage element in a data path they are not put on a scan cells because they are we do not want to put some values because ultimately from the system primary input the values will be coming and in fact, any digital design, if you look into.

(Refer Slide Time: 07:10)



So, it is like this, if this is the system boundary. So, we have got the inputs coming from the outside world and the first stage there will be registers which will be holding these values.

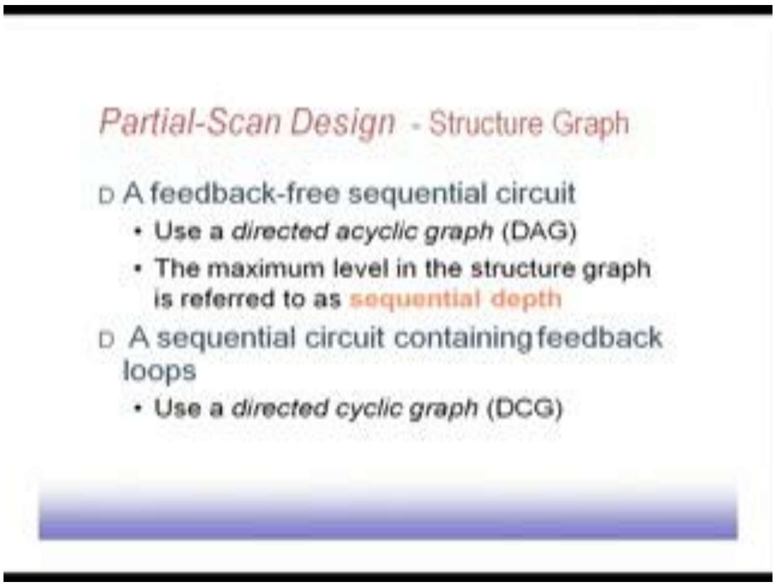
Then after that, after these values have been obtained on to the register then there will be some combinational logic which will do some computation based on these values. And then these values of this outputs of this combinational logic, they will again be stored into some registers and again these values are used by another combinational logic for the next stage of computation ultimately. So, there is another register here and these registered values are available as output. So, that is why these data path registers. So, they are normally getting controlled by the primary inputs coming from the outside world.

On the other hand for controlling all these registers combinational logic etcetera the control path. So, it is implemented in by means of some FSM. So, that at various states it controls the data path elements. So, I may need to test for a particular functionality as a result I need to put this FSM in a particular state for doing that fault excitation. So, for that purpose it is easy, if the corresponding flip flops can be accessed very easily and that will help us in the testing process. So, storage elements on the control path so they are replaced by scan cells storage elements of the data path. So, they may not be replaced, they may not be used for that part.

Second point is that a pipelined or feed forward partial scan design approach. So, it says that sequential circuit is make the sequential circuit feedback free by selecting the storage elements to break all sequential feed feedback loops. Like if we find that there is a flip flop here and its D input, Q output. So, this Q output is feeding some combinational logic and this combinational logic output is again feeding the D. So, in this case it becomes difficult to control this particular flip flop because to set this flip flop to some value I need to set this combinational logic to some value, combinational logic input to some value. And for making the combinational logic to get a proper output for Q, I again need to set this Q value to some value. So, that way it becomes a circular logic. So, that has to be avoided. So, it creates a loop in the reasoning process. So, these has to be avoided.

So, whenever we have got a sequential circuit. So, we have we want to make it feedback free by selecting the storage element to break the feedback loops. So, for that purpose one structure graph is constructed for the sequential circuit, we will see how to do this thing. Third approach is a balanced partial scan design approach. So, we use a target sequential depth to simply the tested, to simplify the test generation process for pipeline or feed forward partial scan design. So, we say the will define this sequential depth essentially after. So, how many flip flops you are seeing in the path? So, if you may tolerate up to certain number of flip flops in a path and after that we want to break that path. So, that is the balanced design style. So, we will see them.

(Refer Slide Time: 10:39)

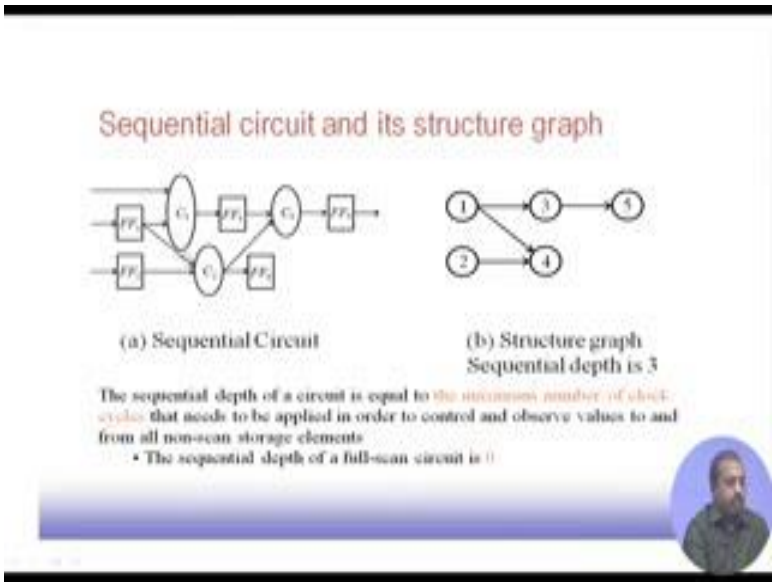


Partial-Scan Design - Structure Graph

- A feedback-free sequential circuit
  - Use a *directed acyclic graph* (DAG)
  - The maximum level in the structure graph is referred to as sequential depth
- A sequential circuit containing feedback loops
  - Use a *directed cyclic graph* (DCG)

So, this structure graph, so if feedback free sequential circuit. So, that is what we are looking for the, for that purpose we use it, first of all we construct a directed acyclic graph. The maximum level in the structure graph is referred to the sequential depth. I think there is an example.

(Refer Slide Time: 10:59)



Sequential circuit and its structure graph

(a) Sequential Circuit

(b) Structure graph
Sequential depth is 3

The sequential depth of a circuit is equal to the minimum number of clock cycles that needs to be applied in order to control and observe values to and from all non-scan storage elements

- The sequential depth of a full-scan circuit is 0

So, here you see that, suppose this is a sequential circuits. So, where we have got this flip flops F F 1, F F 2 feeding this combinational logic C 1. Similarly, this C 2 is feed by F F 1 and F F 2, C 1 is generating output for F F 3. So, this way it is occurring.

Now, if you consider this flip flops and then see like which flip flop can affect which one. So, each node of this structure graph. So, it is corresponding to one flip flop. So, flip flop 1 node 1, flip flop 2 node 2, like this. Now you see there is no dependency between flip flop 1 and 2. So, there is no edge between them, but flip flop 1 may affect flip flop 3 via combinational logic 1. So, there is an edge in the graph from flip flop 1, for node 1 to node 3. Similarly from node 1 to node 4 so there is an edge because through C 2, it can control it from 2 to 4 also is there. Again from flip flop 5 is controlling nobody. So, it is just taken like this.

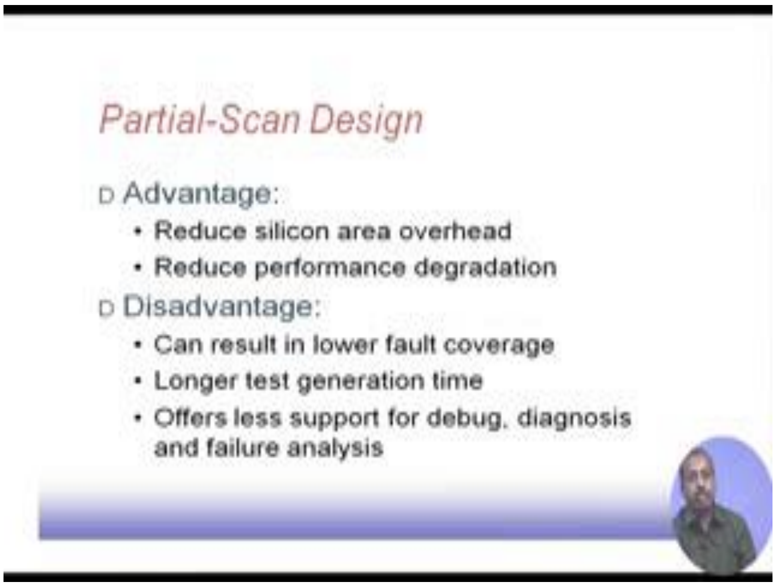
So, if we consider the sequential depth. So, sequential depth is 3, in this case because there are maximum three levels in this graph. So, this is the DFT acyclic graph. So, if you levelized. So, this is level 1, level 2 as a 1 and 2 comes at level 1, 3 and 4 comes at level 2 and 5 comes at level 3. So, maximum depth is 3. So, the maximum level in the structure graph is referred to the sequential depth. A sequential circuit it may contains, so it will happen if the circuit is feedback free, the sequential circuit is feedback free. So, if you look in to this diagram there is nothing from some say from flip flop 3 feeding back to flip flop 1. So, this graph is also a directed acyclic graph. So, there is no cycle in this graph.

So, other possibilities that a sequential circuit containing feedback loops, so there may be feedback loops. So, in that case we will construct the, if there is a feedback loop in the graph then it becomes difficult. So, those a flip flops we need to consider and then we have to break those feedback loop by making them scan flip flops. Like here if there was an edge from say node 5 back to node 1 then we need to convert one of these flip flops to scan flip flops. So, that it is treated as scan input. So, this flip flop 5 in that case will vanished. So, this flip flop 5 is, if there was an edge from flip flop 5 to flip flop 1 and flip flop 5 is converted into a scan flip flop then naturally that edge will vanish.

So, that way this circular paths can be avoided. So, when I have got a full scan design. So, this is an interesting observation. So, is a sequential depth of a full scan circuit is 0 because then no flip flop depends on the other because everything is controlled by the scan input. So, everything is a controllable everything is observable. So, that full scan the circuit the sequential depth is 0.

So, again coming back to the definition, sequential depth is equal to the maximum number of clock cycles that needs to be applied in order to control and observed values to and from all non scan storage elements. So, this is actually if we want to say observe the value of this flip flop then I have to take it to these output then only the because that is a primary output. So, I can see output at this point only. So, somehow I have to excite this C 1, C 3 in such a fashion that this flip flop 1s response propagate through this C 1 flip flop 3 and C 3 and comes to flip flop 5. So, whether that is possible or not that is the different issue, but still because that will depend on other inputs that I have to set for C 1 and C 3, but if it is possible in even in that case it requires three clock cycles for this flip flop 1 value to come at the output of the flip flop 5. So, the sequential depth is 3 in this case.

(Refer Slide Time: 15:06)



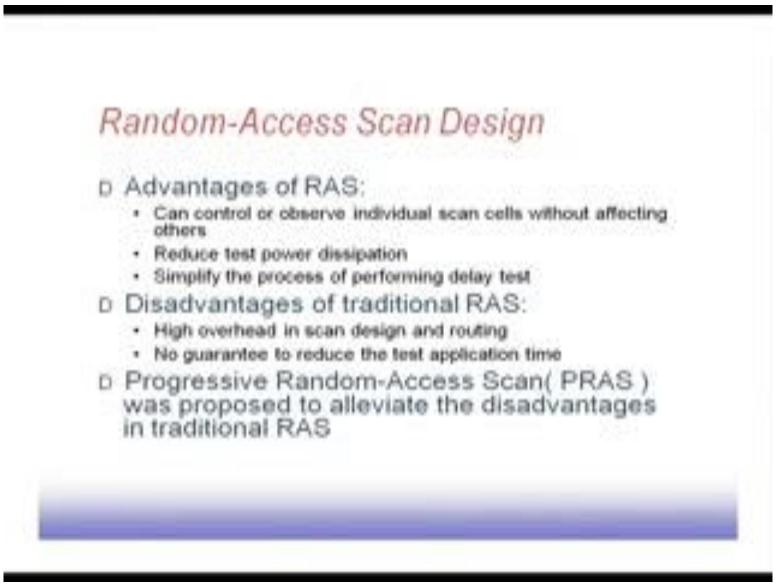
Partial-Scan Design

- Advantage:
  - Reduce silicon area overhead
  - Reduce performance degradation
- Disadvantage:
  - Can result in lower fault coverage
  - Longer test generation time
  - Offers less support for debug, diagnosis and failure analysis

Why partial scan design? So, it has got advantages like the silicon area overhead is less because least number of flip flops are converted into scan flip flop. And performance degradation is also less because for flip flops. So, we are not putting any multiplexer before them. So, as a result, if some flip flop is on the critical path. So, they are at performance is not affected.

Disadvantage can result in lower fault coverage. That is because we are not having control over all the flip flops. So, they are naturally there will be a fault coverage, there may be a fault coverage problem. Test generation time maybe longer because now we many cases we have to use the sequential ATPG. So, when we are using this sequential ATPG. So, take this sequential ATPG algorithm take longer time to run. So, that will take more time and the support for debug diagnosis and failure analysis. So, that is also less. So, it offers less support for debugging diagnosis and failure analysis because the same reason that the entire circuit is not visible. So, the circuit flip flop content many other flip flops will continue to be normal flip flops. So, they are not visible at the output.

(Refer Slide Time: 16:26)



Random-Access Scan Design

- Advantages of RAS:
  - Can control or observe individual scan cells without affecting others
  - Reduce test power dissipation
  - Simplify the process of performing delay test
- Disadvantages of traditional RAS:
  - High overhead in scan design and routing
  - No guarantee to reduce the test application time
- Progressive Random-Access Scan(PRAS) was proposed to alleviate the disadvantages in traditional RAS

Other strategy is random access scan. So, random access scan design. So, we want to access any of the flip flops at any order. Like in a serial scan what happens is, if you want to access a particular flip flop you have to shift in the patterns through the serial input line of previous flip flop, previous scan flip flops and then only it will come there. So, if I say a flip flop is that level K of the chain then, after K pulses, K clock pulses only you will be able to said that flip flop to the desired value.

Similarly, if a flip flop is say M step away from the scan output line, the output flip flop then it will take require M clock pulses to get the output of this get the cell to the output point. So, that sometimes create difficulty because you are unable to access them randomly. So, that requires more time. So, this random access design this tells that i will be able to access any cell in any order. So, you can control or observe individual scan cells without affecting others and reduced test power. So, and this is another issue that reduce test power dissipation. So, what happens is that, if I have a chain. Suppose I have got a four cell scan chain and I have initialized it to say 0, 0, 0, 0, fine they are initially all 0 and there is a combinational logic which is being fed from this scan chain.

Now, what I require is that, for this particular cell I want to make it 1. Now if I want to make it 1. So, I cannot do it directly in a normal scan. So, first I have to make this one and then give shift pulses. Then I have to give another pulse. So, that the pattern becomes 0 1 0 0, another pulse 0 0 1 0. Now when this thing is happening, when these bits are changing like this, it is always feeding this combinational logic. So, hardware is never idle. So, this combinational logic is never sitting idle. So, as soon as this pattern is changing from 0 0 0 0 to 1 0 0 0, it is doing some computation some of the gates are changing their logic value. Though the calculation is meaningless for our purpose, but something will happen.

Similarly, this one from this pattern when it changes to this pattern again some other gates will get excited and some logic transformation will take place. So, this way every time I shift a bit into this scan chain, the scan chain pattern changes and as a result this bits will also changed for the combinational logic input and the combinational logic will do some transition. So, and as an when the make the transition it will consume power and you know that for (Refer Time: 19:18) circuits this power is consumed only when its output changes. The dynamic power is consumed only when its output changes.

So, if the input does not change output does not change. So, if the input is changing then there is very high chance that the output will also change for different gates in the logic. So, if I am doing a serials scan then, at every time, at every shift I am consuming some power because that will go to the combinational logic that input will go to the combinational logic. Whereas, in case of random access scan. So, I am allowed to access a particular cell directly. So, in this diagram, I do not have to do serial shifting like this. I can directly access this cell and modify this value to 1. So, if that is possible. So, previous to the input was 0 0 0 0 from the combinational logic. Now it is 0 0 1 0. So, number of transitions that, the logic series is match less. So, as a result the power consumption will also be less.

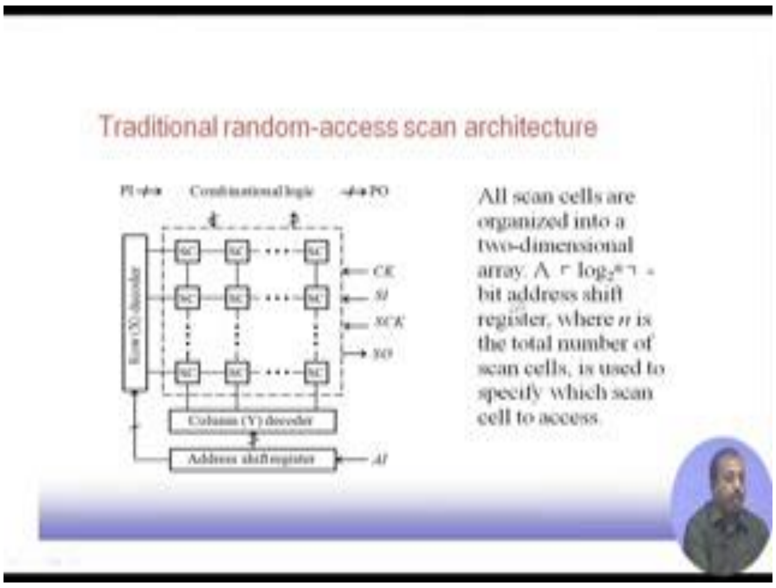
So, this that is why it says that that is reduces the test power dissipation. And simplify the process of performing delay test. So, what happens in case of delay test is that we need to measure the delay. So, first we apply one pattern and. So, that the output settings to some value then, we apply a second pattern to excite a particular gate. So, that that gate makes a transition and that transition gates are transferred to the output and the delay of this gate in making the transition that gates reflected in the delay at the output, at the primary output of the cycle.

So, this is this has to be that. So, two patterns are to be applied; first one pattern for setting the value and the second one is for changing the value. So, this two patterns are necessary. Now in a scan based environment if I am trying to do this; first I apply a pattern then, I have to apply totally new patterns so; that means, I have to again to lot of shifting for applying the second pattern. So, that is cumbersome the other hand, if I have got random access scan then, I can possibly change this bits directly. So that, I can go to the second pattern very easily, it simplifies the process of performing the delay test.

Disadvantage: high overhead in scan design and routing, now, every scan cell has to be addressable. So, as a result I need to have a proper decoding of this is scan address. So, it becomes with cumbersome and routing because now this scan cells are actually put on to the system by the designer. So, test engineer is cannot tell that I want to the scan flip flop at this position or that position. So, that is that is done by the designer, designer has put the flip flops from the design angle and they have put it as close as possible to the required portion.

Now, later on when this test engineer come. So, they have to design this scan cell as scan chain. So, they try to make as good a job as possible, but that may not be sufficient this random access can. So, that may be putting them on to this two dimensional organization may be difficult. Then there is progressive random access scan. So, this traditional random access scan probably issues, they are resolved to some extent in the progressive random access scan.

(Refer Slide Time: 23:00)

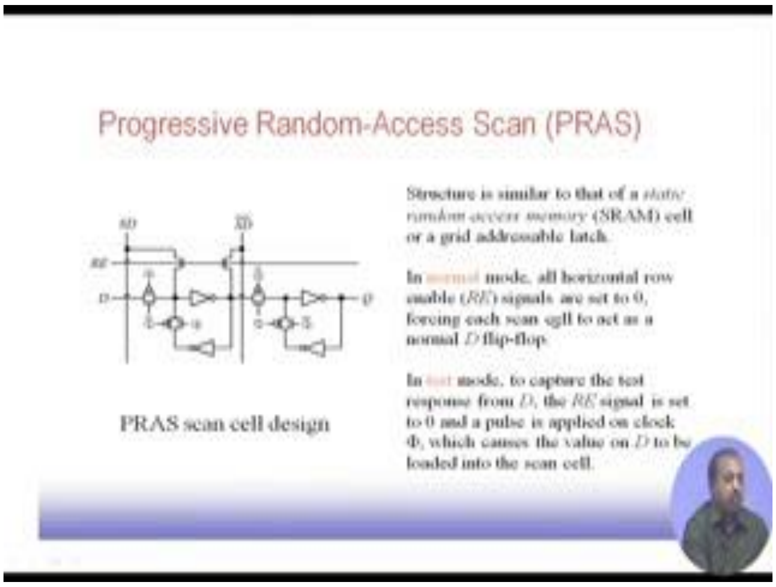


Traditional random-access scan architecture

All scan cells are organized into a two-dimensional array. A ⌈$\log_2 n$⌉ -bit address shift register, where n is the total number of scan cells, is used to specify which scan cell to access.

So, this is the traditional random access scan architecture. So, as I was telling scan cells are organized into a dimensional array. So, if there are n number of scan cells, then a log n bit address will be log n bit address shift register will be necessary. And then out of this log n bits some of the bits will be given to the row decoder, some of the bits will be given to the column decoders. So, this address shift register so, this has got the cell address which we want to change and then this is give a some part of it given to the row decoder, some part of it is given to the column decoder. So, they decode the values and then the appropriate scan cells will get selected by this and the scan input whatever we are looking for. So, that will be loaded into the corresponding scan cell. And similarly we can also if you want to read the content of a particular cell. So, this can outline the content of that cell will be available on the scan outline.

(Refer Slide Time: 24:03)



Progressive Random-Access Scan (PRAS)

Structure is similar to that of a *static random access memory* (SRAM) cell or a grid addressable latch.

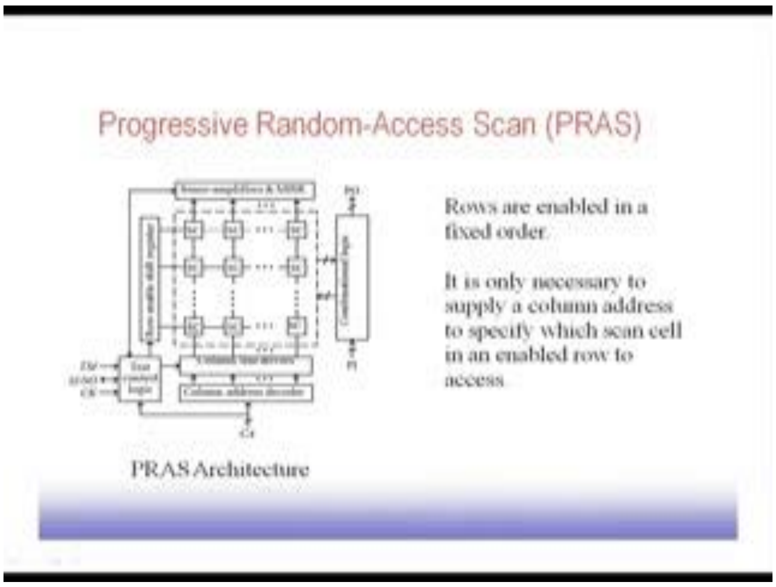
In normal mode, all horizontal row enable (*RE*) signals are set to 0, forcing each scan cell to act as a normal *D* flip-flop.

In test mode, to capture the test response from *D*, the *RE* signal is set to 0 and a pulse is applied on clock Φ, which causes the value on *D* to be loaded into the scan cell.

On the other hand, this progressive random access scan. So, this is similar to say one static ram type of design in normal mode. So, all horizontal row enable signal that RE signals are set to 0. So, all these RE signals are at 0, so that each scan cell will act as a normal flip flop. So, these back to back inverter. So, they are actually that flip flop that will be there and this will force each scan cell to they will if this RE line is 0; that means, these a transistors are off. So, this SD, SD bar both lines are off. So, they will not get selected as a result it will continue as normal D flip flops, but when in the test mode the test response from D, to capture the test response from D. This RE signal is set to 0. So, and a pulse is applied to the clock. So, this RE signal is 0 and this clock pulse is applied as a result from the test pattern or test bit whatever you want to store. So, that will comes through this transmission gate and it will get latched onto this back to back inverters.

In normal mode of operation, you see that this is driven by this clock signals they are complement of each other. So, this phi is phi bar here and this phi is phi bar there. So, the as a result we have got complimented signal. So, when this d is when the clock is high then only when this phi bar is high the clock is low then only this D can come into this thing. So, that way we can put this D value on to this flip flop. So, that is actually the capture operation. And similarly when we are trying to read this design, read the content of a cell then we what you have to do is that, we have to make this RE line high and as we make RE line high the content of this cell will be available in SD and SD bar lines. So, this is typical S ram design. So, this is SD and SD bar. So, we can compute the difference between their voltage levels and there are sense amplifiers and all that a normal S ram organization. So, it is done exactly in the that fashion, we will not going into that detail.

(Refer Slide Time: 26:30)



Progressive Random-Access Scan (PRAS)

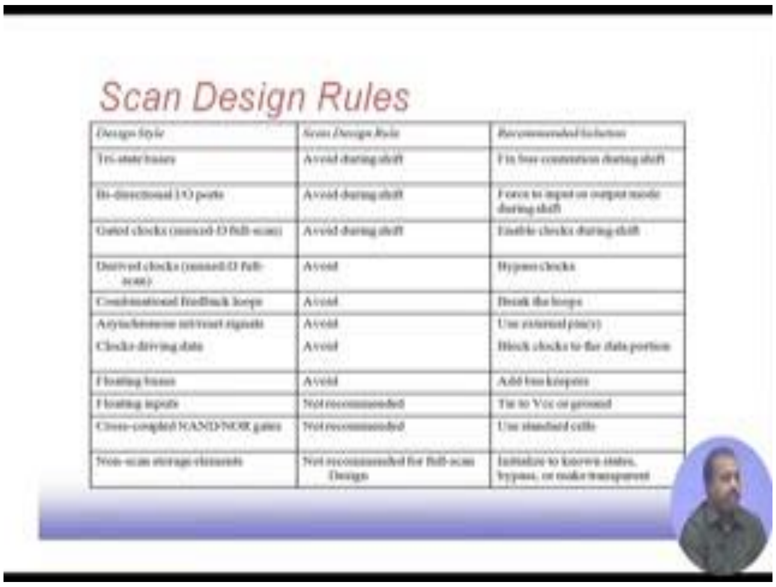
Rows are enabled in a fixed order

It is only necessary to supply a column address to specify which scan cell in an enabled row to access

PRAS Architecture

So, rows are enabled in a. So, the sense amplifiers are there. So, basically this column driver and sense amplifiers are there. So, they will drive individual column and sense amplifier will since the difference between SD and SD bar line and see the difference. So, rows will be enabled in a fixed order. So, that we can read all these scan cell elements and it is only necessary to supply column address to specify which scan cell in enabled row to access. So, this is done in a sequential fashion; this row access is a sequential fashion by the column line address decoder. So, we can select a particular column like which value we want there. So, that way that value will come to the combinational logic and that will be applied. So, this combinational logic has got n inputs. So, all this n inputs will be coming to this through this random access scan architecture.

(Refer Slide Time: 27:26)



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 Clocks driving data	Avoid Avoid	Use external pin(s) 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

There are several scan design rules. So, we will continue with this in the next class.