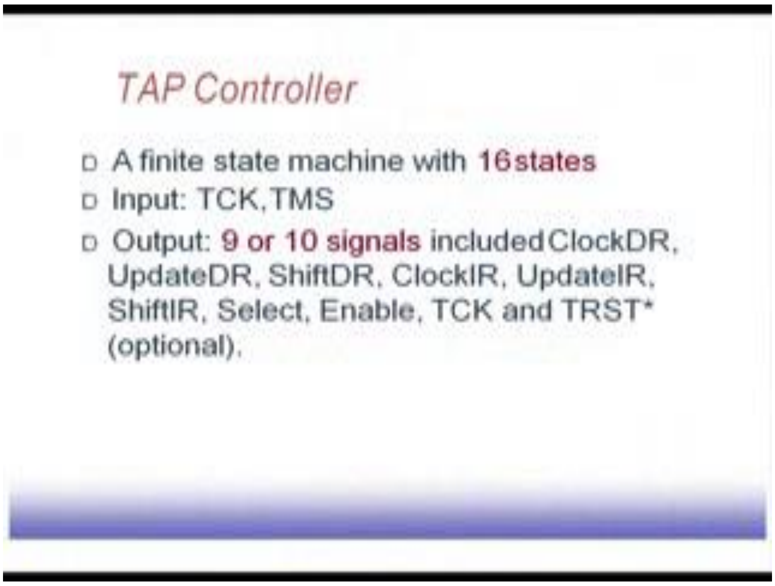
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture- 41
Boundary Scan (Contd.)

(Refer Slide Time: 00:22)

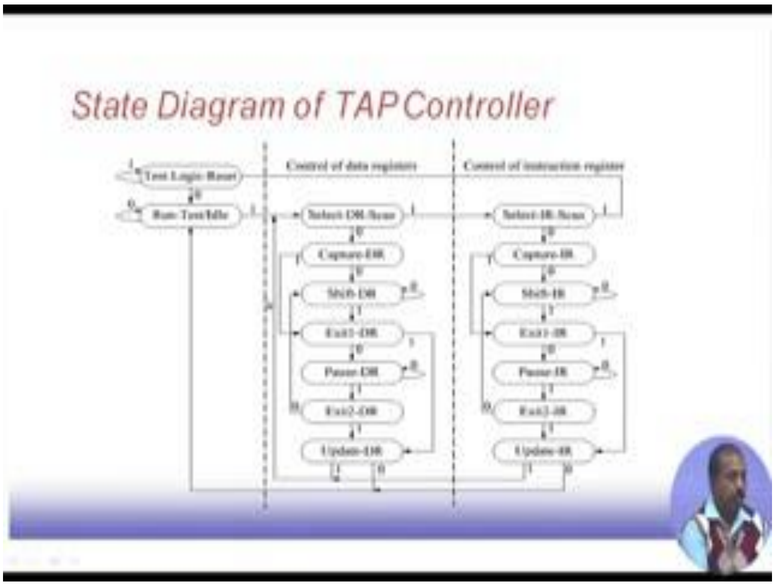


TAP Controller

- A finite state machine with 16states
- Input: TCK,TMS
- Output: 9 or 10 signals includedClockDR, UpdateDR, ShiftDR, ClockIR, UpdateIR, ShiftIR, Select, Enable, TCK and TRST* (optional).

So, next we see, what is a TAP controller? So, TAP controller, it is basically controlling all other modules in the boundary scan. And you see that it has got a number of states, so there are 16 states in this TAP controller. So, it works on this test clock TCK and this TMS; and it outputs 9 or 10 signals included in clock DR, update DR, shift DR, so all this signal values will be updated as it passes through the states.

(Refer Slide Time: 00:50)



State Diagram of TAP Controller

So, this is basically the state transition diagram of this TAP controller, so we will try to understand a part of it. So, this test logic reset, so when this test logic reset is pressed, so this whole thing gets reset and it comes to the state one. It comes to the state test logic reset. Now, you see that whenever it is a whenever this mode this test logic reset whenever this particular pit pin is pressed, so this will be coming to this. Similarly, when this is basically the after that this test mode signal if it is 0, so it will come to this test run test idle cycle and it will remain for in this state as long as this TMS input is 0.

And from this run or test idle, so if that TMS input is 1, so it comes to this particular state, so it select data register scan. They need goes to is if now this transitions are on the TMS line, so a TMS line if it is 0, then it will go to the capture data register; then if it is 0 then it will again going to shift at as long as it is 0, so it is remaining in the shifting mode and then in 1, so it is exiting the data register this exiting this actually the shifting operation is exited. Now there can be a pause, so you can put this TMS value at 0, so as long as this is 0, so it will remain in the pause mode.

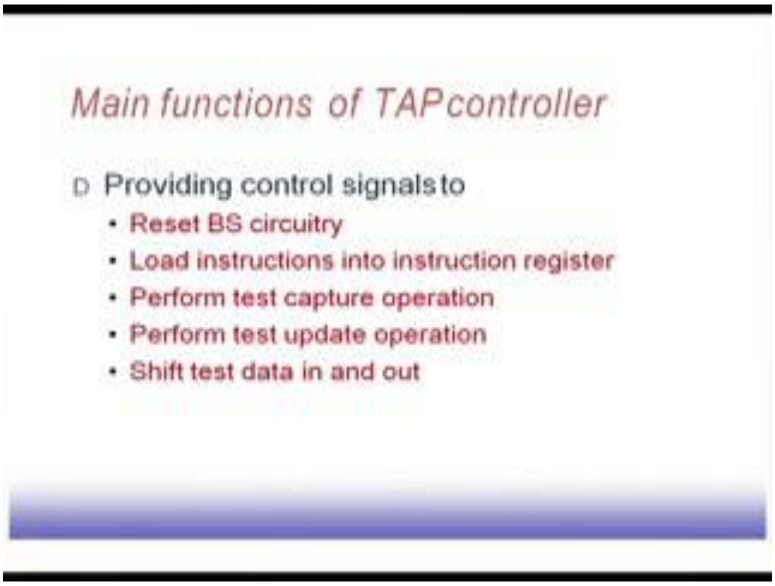
Now, if it is 1, then it comes to another state where exit2-DR. So, this is basically the updation of the input from the circuit. So, for the exit1-pause and this exit2, so there actually taking care they are actually doing the that launching mode or the pattern is getting applied to the circuit and then after that again they if it is 0 then again it will go to the next shift DR, so that the next pattern will be shifted in. So, this way this part is responsible for doing the shifting of test pattern applying the test pattern, etcetera.

Now whenever if you give a one signal TMS equal to 1 that means, you want to update the other register R 2, so this update data register takes place on this. So, if it is 1, it goes to this state again select the data register scan and all that; if it is 0, then it goes to the idle as if this no more testing has to be done, so it has it will remain idle. Again if the TMS is 1, so it will again go into the data register scan mode. Now, in the serial data register scan mode, so if you give this input as 1 then it will go to the instruction register update. Basically in this part it was shifting in the test pattern, getting the responses, updating the boundary scan cells and all that, so that was taking part here, so this is basically the data register updation.

And sometimes we need to update the other part the instruction register update and all that, so that is done by putting 2 TMS high for 2 successive cycles. So, TMS is high for 2 successive cycles from the run test idle mode, it will come to this instruction register mode. So, it is select IR scan, so it will select the instruction register. Then if it is 1, so it will go back to the test logic reset state; if it is 0, it will capture the instruction register. So, it whatever the value is coming on the TDI line, so that will be captured into this instruction register; in this shift IR, so this will be the successive bits will be TDI, so they will be shifted here. Then we have got exit1-IR, so exit pause, so this is actually for applying the instruction that we have got here, so we want to decode it, so that decoding is done, the decoding time is given here then there is update instruction register mode, so where it will be doing the updation.

So, a full description of this will be available in 1149 manual, but these are actually this shows this TAP controller, it goes via a number of states. And by controlling this TMS signal, we can control the transition of this TAP controller, so that it can make the either the test pattern to pass into the boundary scan register cells or it can it can update this internal register for various other things. So, other control related instructions that need to be updated, so that can be taken into the instruction register and other registers.

(Refer Slide Time: 05:41)

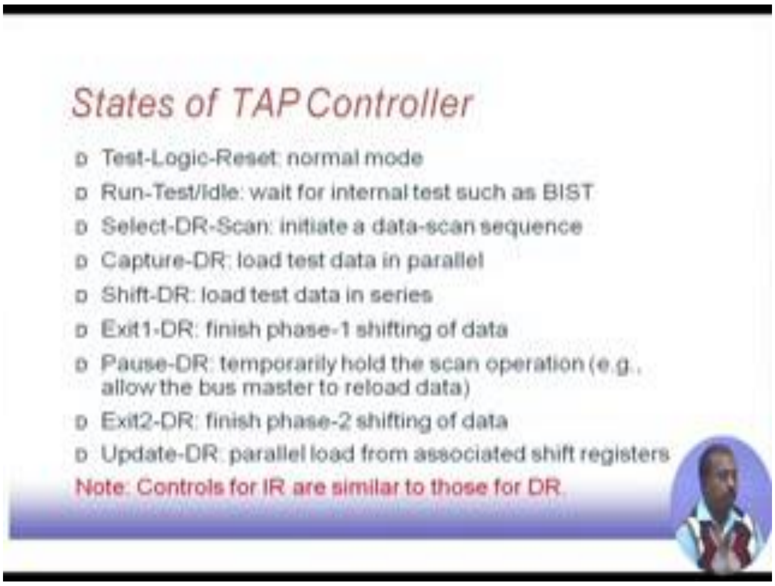


Main functions of TAP controller

- Providing control signals to
  - Reset BS circuitry
  - Load instructions into instruction register
  - Perform test capture operation
  - Perform test update operation
  - Shift test data in and out

So, these are the these are the major functions that are provided by the TAP controller the resetting the boundary scans circuitry, then load instructions into instruction register perform test capture operation, perform test update operation and shift test data in and out, so these are the major functions to be provided by the TAP controller.

(Refer Slide Time: 06:04)



States of TAP Controller

- Test-Logic-Reset: normal mode
- Run-Test/Idle: wait for internal test such as BIST
- Select-DR-Scan: initiate a data-scan sequence
- Capture-DR: load test data in parallel
- Shift-DR: load test data in series
- Exit1-DR: finish phase-1 shifting of data
- Pause-DR: temporarily hold the scan operation (e.g., allow the bus master to reload data)
- Exit2-DR: finish phase-2 shifting of data
- Update-DR: parallel load from associated shift registers

Note: Controls for IR are similar to those for DR.

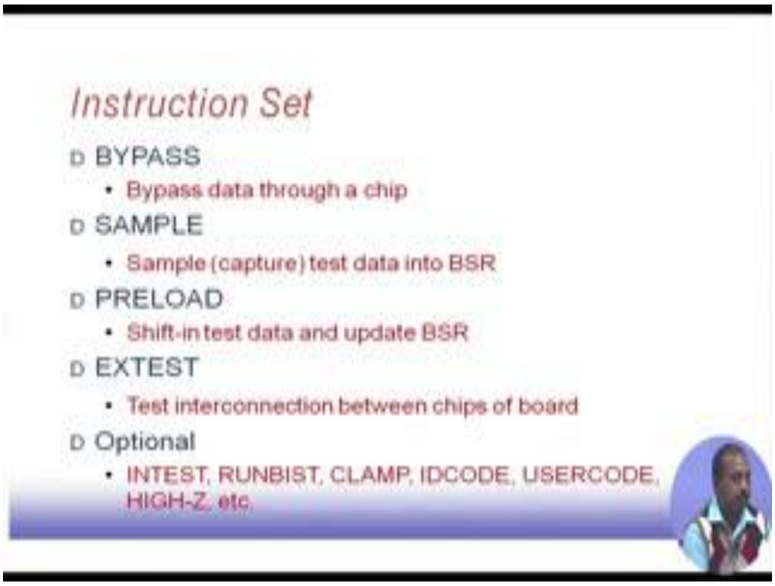
So, these are the various states that we have listed previously. So, test logic reset, so that is the normal mode, so basically the TAP controller will remain in this mode in most of the time, so there is no function of the boundary scan cell register, no testing is going on, so no data needs to be transferred to through the boundary scan registers. Then this run test idle, so wait for internal test such as BIST, so run test idle means the test has been initiated and now when the test will be over the then it should be informed. So, that way it waits for the internal test to be over such as BIST. Then this select DR scan, so it initiate a data scan sequence, so in this DR scan, so you see the run test idle is it is some internal test is going on, then this select DR scan means it some data registers scan operation is going to start, so that is the select DR scan.

Then the capture DR, load test data in parallel; so this capture DR is coming from this is, so capture DR mode, so if this bit TMS bit is 0, so it is coming to capture DR, so this is coming from the from that circuit as the capture input. So, from the inline that we have, so that will be put into the DR registers, so that is into the r 1 register, so that is load test data in parallel. Then the shift DR, so it is the test from the TDI line or the serial input line, so if we are trying to load the data, so then this shift DR state will come load data in series then exit1-DR, so it finish phase one shifting of data, so first phase of shifting data is over. Now, there is some other operation has to be done because the maybe the test pattern will be applied and the response will be captured things like that, so you need to store wait for some time, so that is basically the pause DR operation, so temporarily hold the scan operation.

So, for example, allow the bus master to reload data; bus master will reload data means the pattern will be applied to the circuit and the response will be available on the bus, so that way it will be there. Then this exit2-DR, so it finish phase 2 shifting of data, so here actually the response will be of current pattern will be going out and then that is the phase exit2. And there is update DR, so parallel load from associated shift registers. So, this update DR, so this is basically from the shift register, the parallely the values will be updated.

So, controls for IR are similar to those for DR. So, as I was showing you that, so this entire operation is divided into 2 segments; in one segment we are updating the data register doing some operation the data register; another part we are updating the instruction register and the related operations. So, they are going to be similar the philosophy of operation is similar only in one case the boundary scan register that gets updated, and another case this instruction register and other related places they get updated.

(Refer Slide Time: 09:19)



Instruction Set

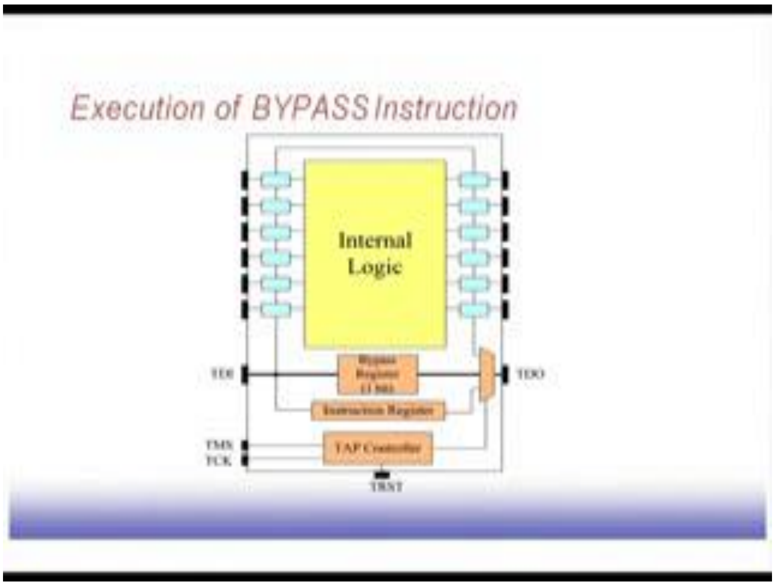
- BYPASS
  - Bypass data through a chip
- SAMPLE
  - Sample (capture) test data into BSR
- PRELOAD
  - Shift-in test data and update BSR
- EXTEST
  - Test interconnection between chips of board
- Optional
  - INTEST, RUNBIST, CLAMP, IDCODE, USERCODE, HIGH-Z, etc

Now, what are the instructions that this step controller can support. First instruction is a bypass operation, so it bypasses data through a chip, so this is a bypass operation. Then there is a sample operation, so that will sample or capture the test data into the boundary scan register. Then there is a preload, so shift in test data and update the boundary scan register. Then there is an EXTEST mode, where the test interconnection between chips of a board, so they will be tested. And there is an optional INTEST, RUNBIST, clamp, ID code, then user code, high-Z, so these are some optional instruction, but these are the 4 basic instructions that are to be supported for a board level testing.

So, bypass this is required because you can bypass a chip, if the next instruction is not meaningful instruction or data is not meaningful for this particular chip, so you can bypass it putting it into bypass mode. So, sample mode, so this is basically to get the, it is basically getting the test data the captured data into the boundary scan register. Then preload, basically shifting the shifting in the test data, so that is there. And EXTEST is basically for testing interconnect between the boards.

So, there is a difference between testing a chip and testing an interconnection. So, testing a chip means there we are willing to apply the particular pattern to the inputs of the chip, but when you are trying to test the interconnects, so output of one chip connects the input of another chip. So as a result the test pattern actually goes to the output of the chip, and it is captured at the input of the second chip which is counter of what we do for chip testing, where the input is applied to the input of the chip and output is collected from the output pin of the chip. So, it is, so that is why it is called an EXTEST mode. So, naturally there has to be something called an INTEST mode, where the pattern is applied to the chip for testing, so that is what it may not be there like this capture mode automatically may take care of that. RUNBIST, so if the chip is BIST fitted then this RUNBIST mode is useful. Then clamp, so it will take the value at particular point, so we will see them.

(Refer Slide Time: 11:45)

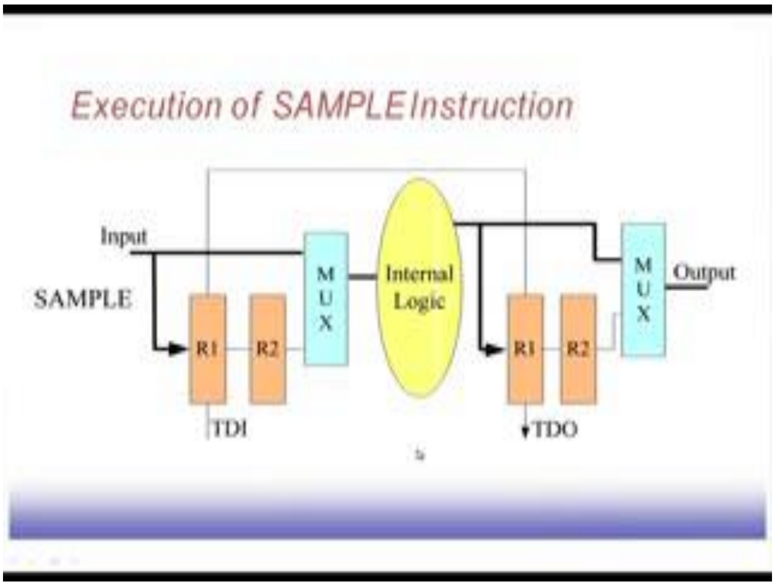


Execution of BYPASS Instruction

So, how this bypass instruction is executed, so in the bypass instruction mode, so the instruction register gets that bypass instruction. And then this TAP controller, we will select this multiplexer in such a fashion that whatever is coming in TDI is going to the

TDO. And this bypass register is a one bit register, so it will introduce a delay of only one unit through the chip. So, instead of having a delay of equal to all this sum of all this boundary scan cells, so it will have a delay of only one bit to get the input transferred from TDI to the TDO line, so that is the bypass instruction.

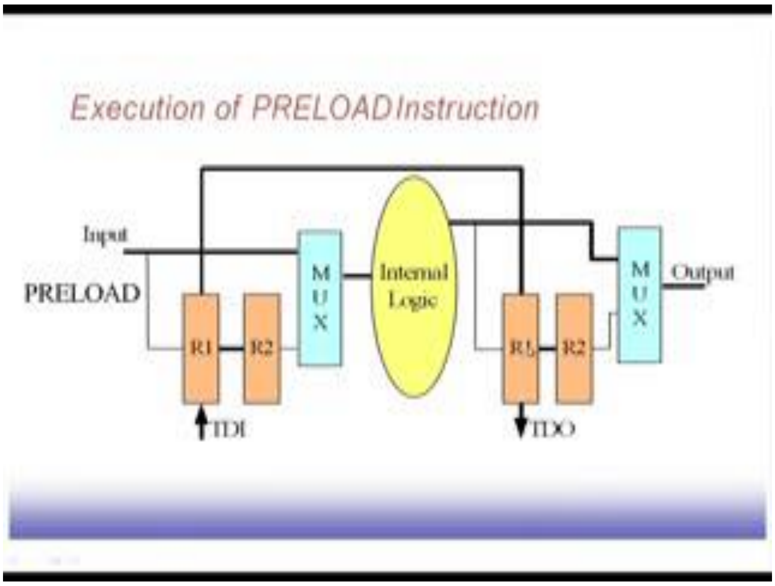
(Refer Slide Time: 12:26)



Execution of SAMPLE Instruction

Then the sample instruction, so sample instruction, so what is happening, first this the input line which is the functional input, so that is going to this multiplexer, and then this is also coming to this R 1 register, now this multiplexer through this multiplexer it is applied to the internal logic. So, R 2 is not connected to this, this is multiplexer is programmed in such a fashion that is input goes to the internal logic, so the result is available at this point. Now, this result is coming to this multiplexer available at the output; at the same time, this result is also available at the R 1 line here; the R 1 register of this cell, so you see that the input has been captured here and this output has been captured at this point, so that is the sample operation. So, by application of a particular input, what happens to the output, so that is captured in the R 1 line, R 1 register.

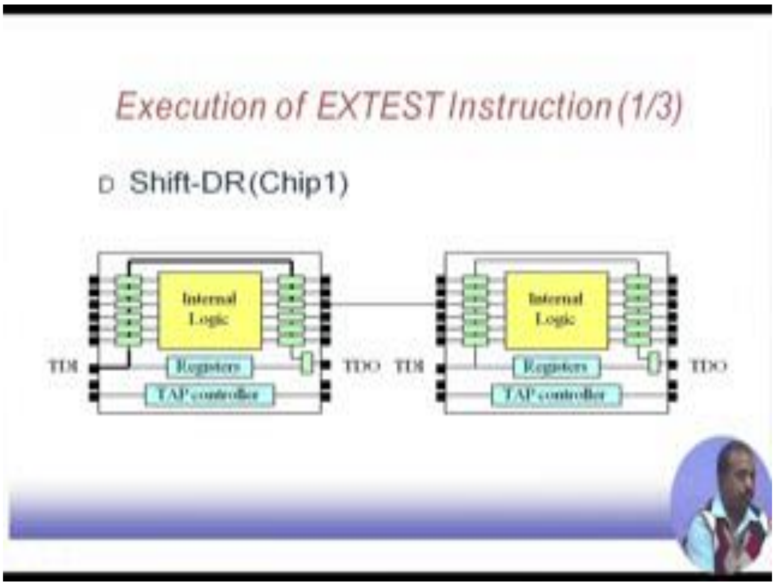
(Refer Slide Time: 13:38)



Execution of PRELOAD Instruction

Then there is a preload instruction, so when a preload instruction whatever is coming through this TDI line, so that is loaded into this register, and that is also loaded into this R 1 register. But the functional input is functional part goes unaltered like this internal input, so it is loaded into through this multiplexer; multiplexer is selecting this input line. So, it operates functionally the internal logic the logic part of the circuit, so it operates the in its normal mode. But this through this TDI line we can preload some pattern onto this boundary scan cells, so this R 1, R 2 register, so you can preload some pattern, so that is the preload instruction.

(Refer Slide Time: 14:30)

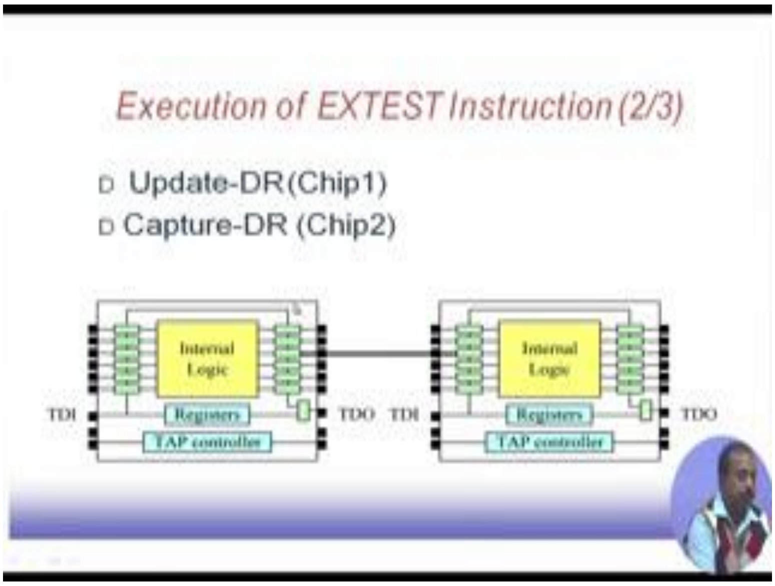


Execution of EXTEST Instruction (1/3)

- Shift-DR(Chip1)

Then this EXTEST instruction, so what is done first, so EXTEST, I have said that it will be testing the interconnect between the chips like we are interested to check say this particular interconnection. Then how can we test this thing, first of all this test data input that we want to apply at this point maybe we want to check whether this line is stuck at 1 or not, so you have to send a 0 onto this line, so that is 0 bit is shifted through this TDI line like this. So, there the chip one it operates in the shift data register mode and chip 2 does not have to do anything, so it is not at working in the test mode, so it is working in the normal mode. So, this bit through a number of shifting, so it comes to this point.

(Refer Slide Time: 15:20)

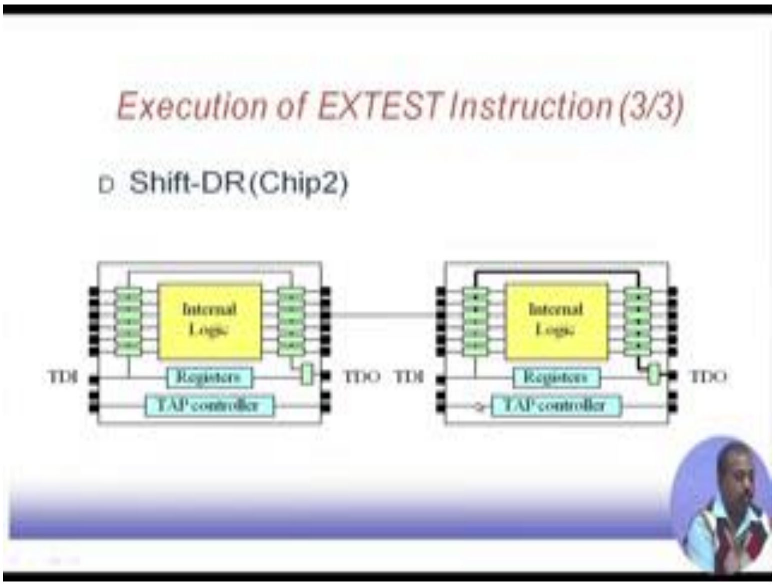


Execution of EXTEST Instruction (2/3)

- Update-DR(Chip1)
- Capture-DR (Chip2)

Then in the second step, this chip one is put into update DR mode. So, if you put into the update DR mode then you know that when this shifting was going on, so it was only affecting the R 1 register, so this R 1 register was only getting affected, so it was not available on the R 2 register. So, at the output sides, so it was not available on the R 2 side, it was only available at R 1 during the shift operation. So, this updated DR operation this value is available at R 2 as a result the value is available onto this particular pin. Now, this pin being, so now, this circuit, so this is put in capture DR mode, so from the input it from this input, so it is getting captured onto this cell. So, this cells R 1 it has got this value now. Now, this has to be transported to this output.

(Refer Slide Time: 16:19)

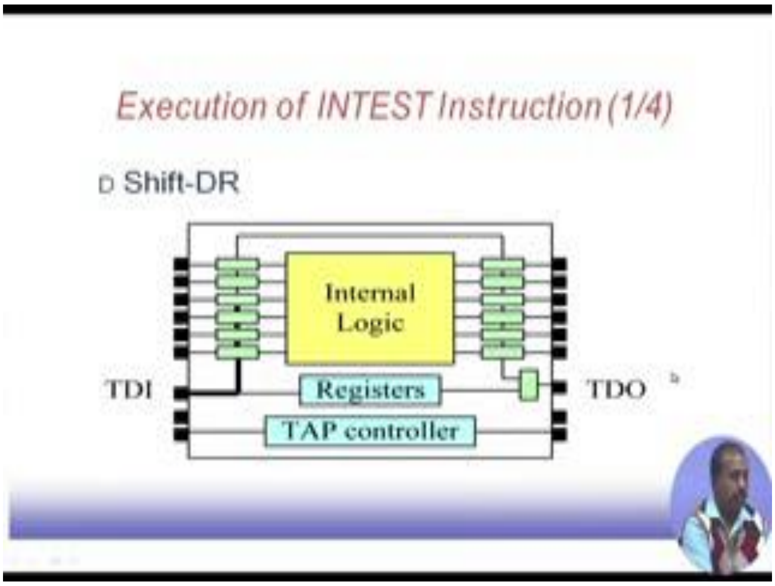


Execution of EXTEST Instruction (3/3)

- Shift-DR(Chip2)

So, how that is done, so that is done in this fashion. Now, this chip 2 is put into shift DR mode, so this R 1, so of through this R 1 line, this R 1 inputs R 1 register values they get shifted in through this registers and they come to this TDO, so all these are done by the R 1 line. So, that way we can get it on to the that serial outline of the boundary scan cell that is coming here and that is finally, connected to the test data output for the entire chip, so that way this is in the 3 steps, so this can be done. This EXTEST execution first the chip is this first chip is put in the shift mode, then the first chip is put in the update mode second chip in the capture mode, and in the third phase the second chip is put into the shift mode to finally, get the shifted value at the output.

(Refer Slide Time: 17:17)

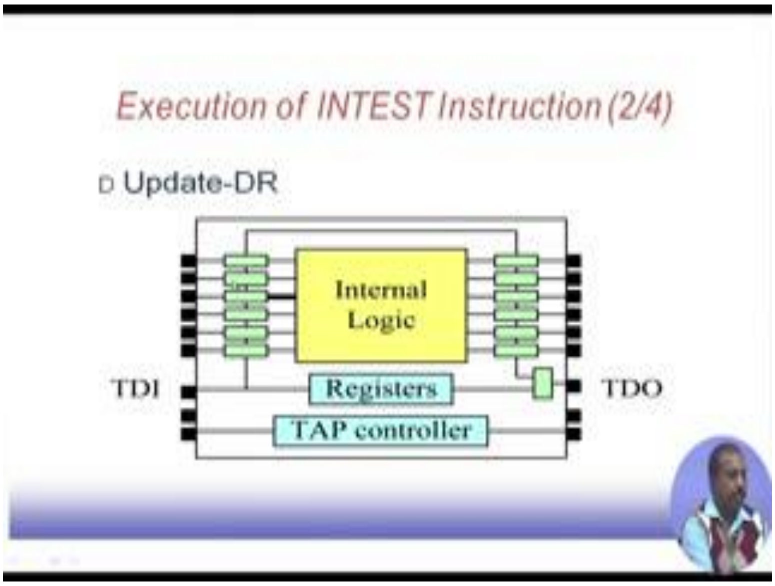


Execution of INTEST Instruction (1/4)

- Shift-DR

Now, the INTEST instruction, so now, here what is happening is the some test pattern that is coming through this TDI line that needs to be applied to the circuit. So, first it is this TAP controller will put this shift it will put it into the shift DR state, so this shift DR mode, so this test data that is coming to the TDI line, so they get shifted into this R 1 registered of this of the boundary scan cells.

(Refer Slide Time: 17:47)

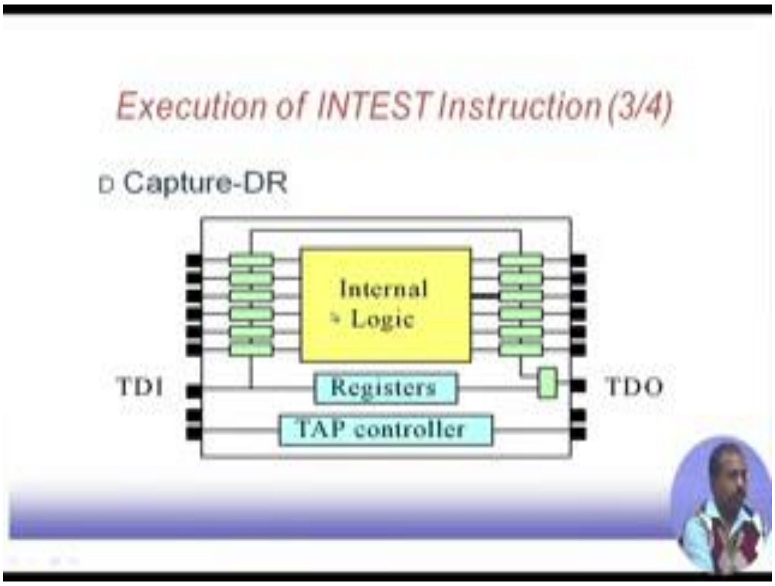


Execution of INTEST Instruction (2/4)

- Update-DR

And after that it goes into the update data register mode. So, it is assumed that this particular operation it requires a certain pattern to be applied only on this line only on this bit I need to put some special value; rest of the bits for the internal logic they are assumed to already have a assumed some values only another bit has to be same. Maybe they are those values were obtained from say by normal operation that, so the normal operation the bits got transferred there and all that only this through sample operation. And only if only one bit we need to transfer that one bit transfer is taking place, so this shift operation that bit comes to the R 1 register of this particular cell R 1 register of say this cell. And after that by update DR, this R 1 value goes to R 2 value and R 2 register; and R 2 register content is available at the out pin, so this outline is now fed to the is fed to the internal logic, so all this value is available here. So, these update DR, so this test pattern can be applied on to this internal logic now.

(Refer Slide Time: 19:00)

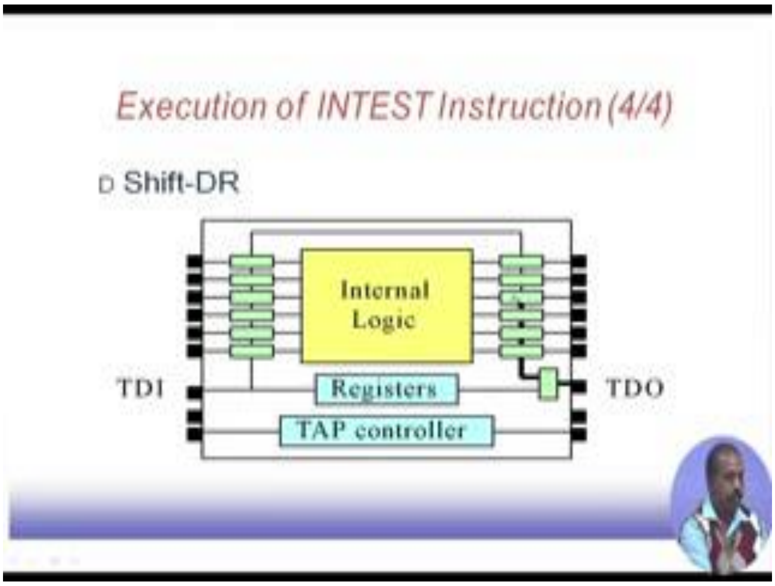


Execution of INTEST Instruction (3/4)

- Capture-DR

After that there has to be captured, so this value whatever has been applied to this internal logic, so that produces some output. Suppose, it has modified several bits, but we are interested to get the value that is captured onto this register alone, so that way again, so it goes in the captured data register mode, so that way this value will be available onto the R 1 register of this cell.

(Refer Slide Time: 19:27)

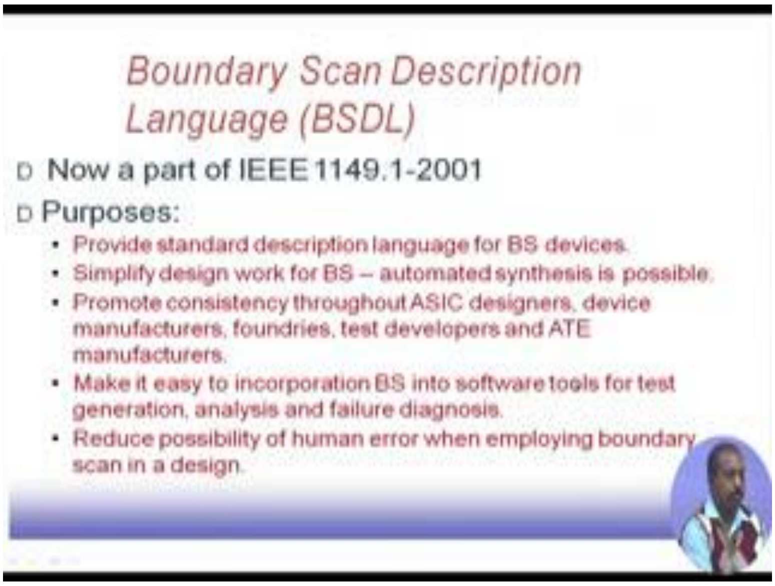


Execution of INTEST Instruction (4/4)

- Shift-DR

And again there will be a shifting. So, shifting will be going of the all this R 1 register, so they are put onto this chain and through this shifting operation it finally goes to the test data output pin of the chip. So, this way I can for testing for INTEST, where the pattern that I apply has to go through this logic, so first the bit is set here by shift operation. Then by an update operation, so it is put on to R 2 here then this is then this circuit operates after that it goes into a captured DR where the response of this logic is captured by the at the output cell R 1 register. And then at the next point the value will be shifted out through a number of cycles into through this TDO line through this R 1 register forming that SI So that real shifting, so it will come to the TDO pin of the chip. So, this way the INTEST instruction is taking place.

(Refer Slide Time: 20:29)

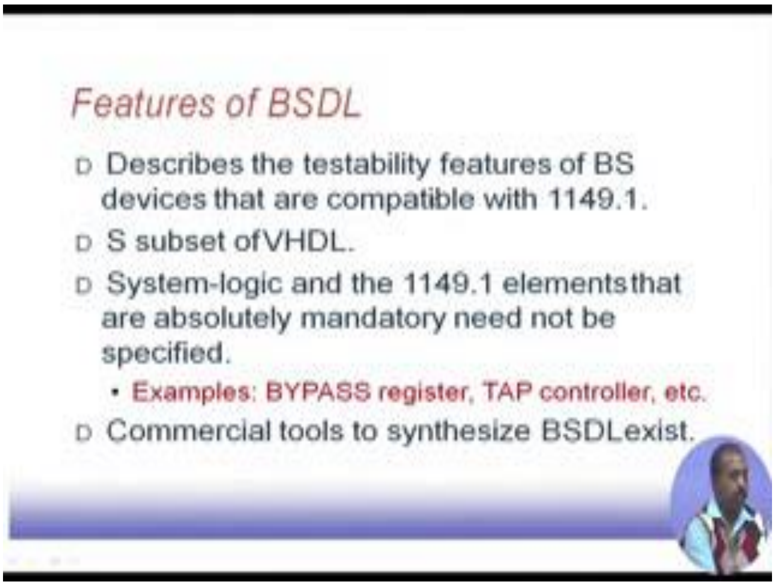


Boundary Scan Description Language (BSDL)

- Now a part of IEEE 1149.1-2001
- Purposes:
  - Provide standard description language for BS devices.
  - Simplify design work for BS – automated synthesis is possible.
  - Promote consistency throughout ASIC designers, device manufacturers, foundries, test developers and ATE manufacturers.
  - Make it easy to incorporation BS into software tools for test generation, analysis and failure diagnosis.
  - Reduce possibility of human error when employing boundary scan in a design.

So, then next we look into this boundary scan description language or BSDL, so this is a part of 1149.1. There are several purposes of it, it provides standard description language for this boundary scan devices, simplifies the design work of boundary scan automated synthesis possible. So, basically there is a language then the these are the boundary scan designers and this test the chip designer who are designing their chips to be compatible with boundary scan, so that becomes easier. So, consistency in the ASIC design phase, manufacturing phase, foundries, and this stage developer, ATE manufacturers all of them they can be unified. It makes it easy to incorporate boundary scan into software tools for test generation, analysis, failure diagnosis etcetera; and it reduces the possibility of human error when employing boundary scan in a design. So, this is a part of 1149.1, it was been introduced 2001 the boundary scan description language.

(Refer Slide Time: 21:36)

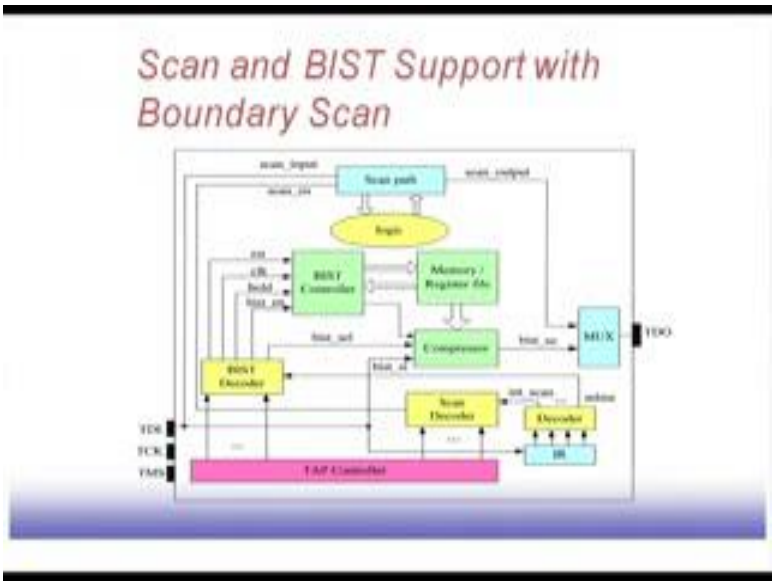


Features of BSDL

- Describes the testability features of BS devices that are compatible with 1149.1.
- S subset ofVHDL.
- System-logic and the 1149.1 elementsthat are absolutely mandatory need not be specified.
  - Examples: BYPASS register, TAP controller, etc.
- Commercial tools to synthesize BSDLexist.

What are the features; it describes the testability features of boundary scan devices that are compatible with 1149.1. So, it does not try to augment that particular standard, but it only provides a description policy for the boundary scan device. So, this is a subset of S subset of VHDL, so subset is the synthesizable subset, VHDL synthesizable subset has been used for describing it. System-logic and 1149.1 elements that are absolutely mandatory did not be specified. So, bypass register, TAP controller, so they are always there, so they are the mandatory part, so they need not be specified only the remaining things are to be specified like whether the TRST mode is there, reset mode is there, how many internal registers are there, so those things are to be specified. So, like the basic things are not to be specified. And there are some commercial tools that have come up for this for synthesizing in the boundary scan description language that we have. So, if you have some description in that language then commercial tools are there which can do the synthesis part.

(Refer Slide Time: 22:53)



Scan and BIST Support with Boundary Scan

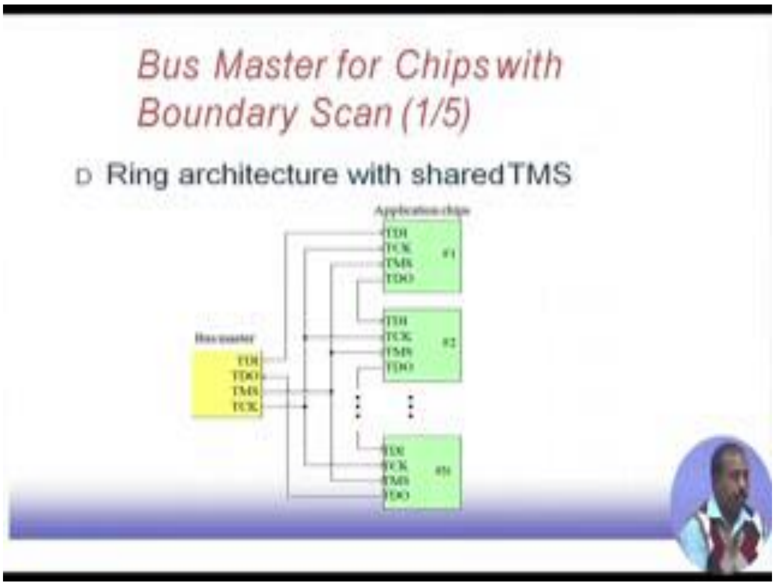
So, this is the overall structure, the BIST support scan and BIST support with boundary scan. Like what can happen, so this is the entire chip, so this is the logic that we have. Now, this logic may be tested via some external input or it may be tested via some BIST. So, what is done is that this TAP controller, so this test access, so if you are trying to send some input serially to the scan chain, so through this TDI line, you can feed the input into the scan chain. And similarly this scan outline, so that can be connected in to the TDO test data output point through this multiplexer. So, this TAP controller can do the that serial shifting through this scan path pattern may be applied response maybe captured and all that, so that may be done.

Second thing is that it can there may be some BIST. So, this TAP controller can be informing the BIST decoder like what to do, so the that way this BIST decoder will in turn till the BIST controller like whether it should do the operation, if there is a reset signal, clock signal, hold signal, so they are given to the BIST controller. Then this memory and register files, so they are normally tested by BIST. So, here the logic may also be tested by BIST, but the logic part of the circuit, but it is normally not done because the reason that logic part is I should say more heterogeneous across different designs, so they are the test patterns will be different.

But when you are talking about memory and register files, so they are more regular in nature. So, possibly we can just take help of some regular test patterns for doing this say memory register file updation so and their testing. So, the BIST controller can be normally used for this part and naturally there has to be some complex service which will complex all this responses and then it will be selecting the BIST signature output, so the serial output, so that can be given out of TDO.

Now, this TAP cont this boundary scan, it will have the instruction register as well, so instruction register it will go to the decoder, so decoder will tell whether it is a BIST instruction, it is an internal scan. So, if it is an internal scan the decoder of this TAP controller, so it will inform the scan decoder, and it will tell how to input the scan it goes it scannable signal to the scan path, so what sort of operations are to be done, so that is told by the scan decoder. So, rest of the things are not shown there may be some control lines needed for controlling the scan path, but that is not shown here. Similarly, we have this thing, this BIST operation. So, if it is a BIST operation then this BIST decoder will be informed and accordingly the BIST decoder will be controlling the BIST session. So, this way we can have this boundary scan which can support this scan and BIST operations in the entire operation.

(Refer Slide Time: 26:08)



Bus Master for Chips with Boundary Scan (1/5)

- Ring architecture with shared TMS

Sometimes we can have bus master for chips with boundary scan. So, we can have this type of structure, so ring architecture with shared test mode signal. So, here what is happening is that, so these are some chips one, 2, 3, these are different chips and you see the test mode signal that is here, so it is common to all the 3 chips. And the TDI line - the test data input line is a going to the first chip; and this TDO of first chip is going to the TDI of second and that way TDO of second will go to the TDI of third that way it goes. So, these actually forming some sort of a ring structure all these chips as if they are put on a ring something like a daisy-chain. So, if the first though the test mode signal is given to all of them this test data input and test data output, so they are actually chain like this.

So, in the test mode signal is common to all the chips, so all of them will be doing the same operation, all of them will be doing the same operation. But this test data input lines they are serially connected, so you can put different values onto this data registers, so that they can be operating on different test data. But the test mode line being same there transitions of the TAP controller will be same, so all of them will be going into the same set of transitions, so this way we can configure these chips onto different modes and this different architectures can be made out of this boundary scan. We continue in the next class.