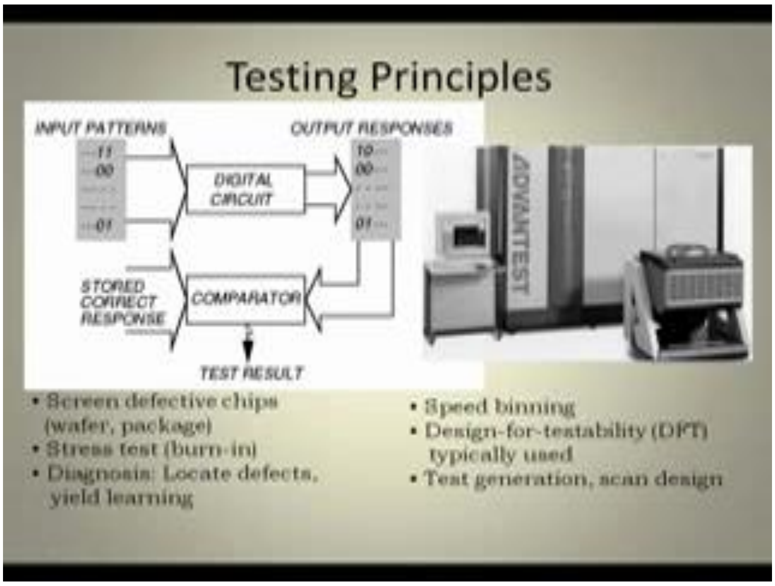
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 46
System/Network – On – Chip Test (Contd.)

(Refer Slide Time: 00:21)



Testing Principles

- Screen defective chips (wafer, package)
- Stress test (burn-in)
- Diagnosis: Locate defects, yield learning
- Speed binning
- Design-for-testability (DFT) typically used
- Test generation, scan design

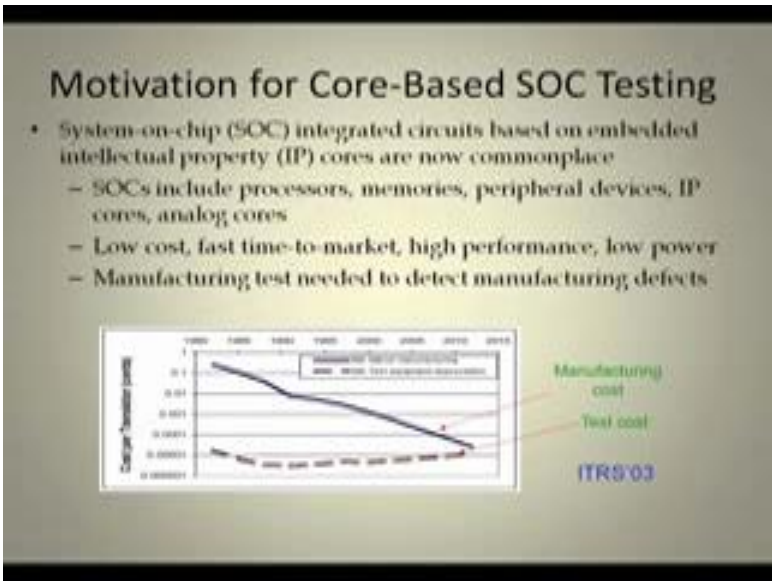
So, testing principles, so this diagram we are bit familiar, but what is happening is that the input patterns they are applied to the digital circuit, so the responses are collected, so they are compared with the stored correct response and get the test result whether it is true or not; but after this has been done, so we have to screen the defective chips at the wafer level or at the package level. So, the defective chips are to be separated then the stress test, so it has to do a, so if the chip is operated for a number of continuous hours then whether the chip starts malfunctioning or not sorry the stress test or burn in face.

Then for the diagnosis purpose, so we need to locate the defect and then we see what is the yield that we are getting. So, yield learning actually what happens is that in some in some production if we find that the chips are failing, they are not passing the tests then we try to figure out like there may be some problem with the manufacturing process, so we try to locate the defects. And if a number of chips show the same defect location that means, there is some problem with the manufacturing process. So, the yield got decreased, so we need to correct those process, so that the yield will increase again.

Then speed binning, so there may be different versions of the chip like some of the chips they may not be able to operate at a very high frequency, they may be able to operate in the kilo hertz range, but not in the giga hertz range. So, even after the this fabrication process everything is similar, so due to some process variation, so it may so happen that we cannot reach some frequency level of operation for some chips. So, that way we may divide the chips into different speed bins and then that you can say that some of them are slow version some of them are fast version like that.

Then design for testability, so that is typically used, so that we can do this testing process in a methodical fashion. And then this test pattern application response gathering analysis they can be done in a very well defined way. Test generation to generate the test pattern and scan design, so these are the various issues that we have in the testing process.

(Refer Slide Time: 02:46)



Motivation for Core-Based SOC Testing

- System-on-chip (SOC) integrated circuits based on embedded intellectual property (IP) cores are now commonplace
  - SOCs include processors, memories, peripheral devices, IP cores, analog cores
  - Low cost, fast time-to-market, high performance, low power
  - Manufacturing test needed to detect manufacturing defects

So, why do we go for this core based SOC testing, why is it so important. So, you see that system on chip integrated circuits based on embedded intellectual property cores are now commonplace, so this is a very now there are many components many chips that are being designed which are designed as SOCs. In fact, the embedded VLSI system has become very popular and then it is actually this SOC based system design.

SOCs include processors, memories, peripheral devices, IP cores, analog cores all that, so any company who is involved in digital IC design analog IC design, so they are also designing the cores. Because as a single chip solution, so that their design has to go inside those systems corresponding to some application, so we need the core based design. So, this SOC becoming more and more popular, so this core vendors or IP core designers they are also becoming more and more important.

So, for example, this arm processor, so arm is never designed as a chip, so it is always designed as a core. So, depending upon your requirement, so we can just take some part of arm processor whereas, we do not take other features of the arm. For example, the floating point portion may not be required for my operation, so I do not take the floating point part in it. I may or may not be interested in the security aspect, so as a result I may or may not include the security part in the processor design. So, the processor itself is getting designed as a core only, so that depending upon the requirement, so I can customize and I can take only that core into my system. So, this SOC design, now include this core based designs.

Low cost, fast time-to-market, high performance, low power. So, low cost because we are not wasting time in getting the chip fabricated for this individual components. And since the vendors, so they are providing a net-list which is they are providing a layout level description, so that is at a much higher-level, so that is they did not have to go into this fabrication process of the chip, so this cost of this core is going to be much less than the cost of the chip.

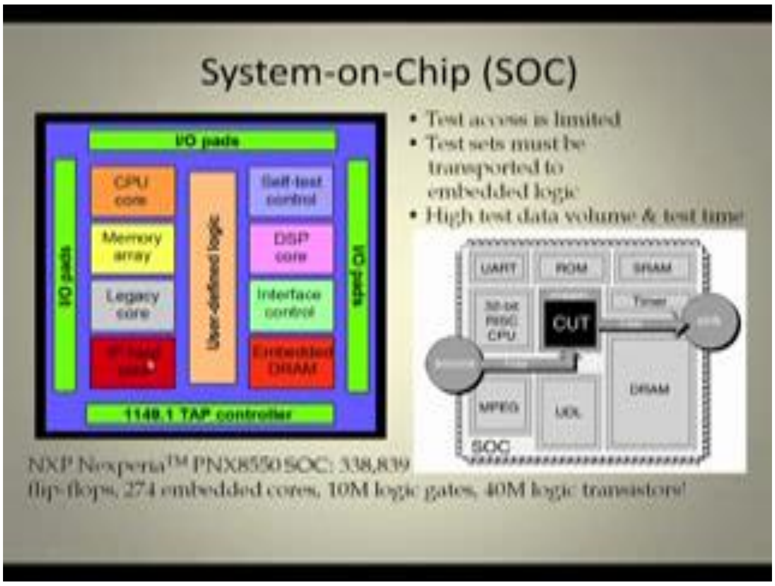
Then fast time-to-market because now as a designer, so if I think that my system will have a number of different components like say CPU, memory, DSP etcetera and for the sake of best performance what I need to do is to take all of them into a single chip. Now if I myself try to design all these components or in a then it will take huge amount of time; and no company that we have, so it is it is possible that we have got expertise in all these domains. So, every company they do specialize in one or two domains, so that way this if I go one for high performance may be some company is doing high performance CPU design, somebody is doing high performance DSP processor design, so that way there may be issues. So, that way I need to take their designs into my system, so that is why we have to go for this core based SOC design.

Low power, so power minimization is again another very important issue. And again this IP cores that are getting generated that are getting developed, so this power may be a part there and this they may be optimized better for the power, so that way I have to take I can go for this cores which are branded as low power versions, so they will consume less power.

Manufacturing test needed to detect manufacturing defects. So, this is very important because now previously as in a board level design this individual chip testing was not that much important, but here all the cores are to be tested. And in fact, this particular road map it shows how this manufacturing cost has in increased. So, over the years, so if you look into this number of transistors cost per transistor in terms of cents, so it has reduced drastically. Why, because the large numbers of transistors are being fabricated on the same die, so cost part transistor it has come down significantly.

On the other hand, the test cost though it initially dropped with the increase in the number of transistors, so the ratio actually made it to reduce in terms of this ATE cost and this cost per transistor, so that way it was reducing. But now it has again started increasing, so from see 2000 onwards, we can see a steady increase in the test cost. And it is very much likely that there will be a there will be an overshoot that test cost will cross the manufacturing cost and that is because these ATEs are becoming costly, so if we want to test more and more chips at high frequency. So, if we want to test that ATE cost are going to be high, so that way the test cost is going to be high. The complexity the amount of test time to be devoted is going to be high, so it is going to have some good amount of time requirement, which is translated into the cost of the system, so test cost is going to overcome the manufacturing cost.

(Refer Slide Time: 08:35)



System-on-Chip (SOC)

- Test access is limited
- Test sets must be transported to embedded logic
- High test data volume & test time

NXP Nexperia™ PNX8550 SOC: 338,839 flip-flops, 274 embedded cores, 10M logic gates, 40M logic transistors!

So, this is a typical system on chip architecture, you see that we have got a number of core CPU core, memory array, legacy core, IP hard core then this self test control, DSP core, interface control, embedded DRAM etcetera. There may be some so we have got some I O pads around it through which this I O pins for the chip are available. For serial testing boundary scan structure, this 1149.1 tap controller is introduced, so that I can put a boundary scan cells around it and that way it can be operated that it can be tested.

Now, if you look into this system you see that the test access is limited, so for accessing this DSP core, so only may be only a few of the pins are available on the I O pads, rest are not. And in the worst case, it may so happen for that some of the cores which are embedded deep inside the system none of the pins are available at the system I O pads. So, then how do we apply the test patterns to them, how do we see what is the response generated by them. So what is this, so test access is limited, so what is required that test sets must be transported to embedded logic. So, as we are discussing in this example that this test pattern must be transported from the test at the source that may be an ATE may be a BIST, whatever it is from there through this TAM it needs to be transported to this circuit under test the core under test and the response is to the sink.

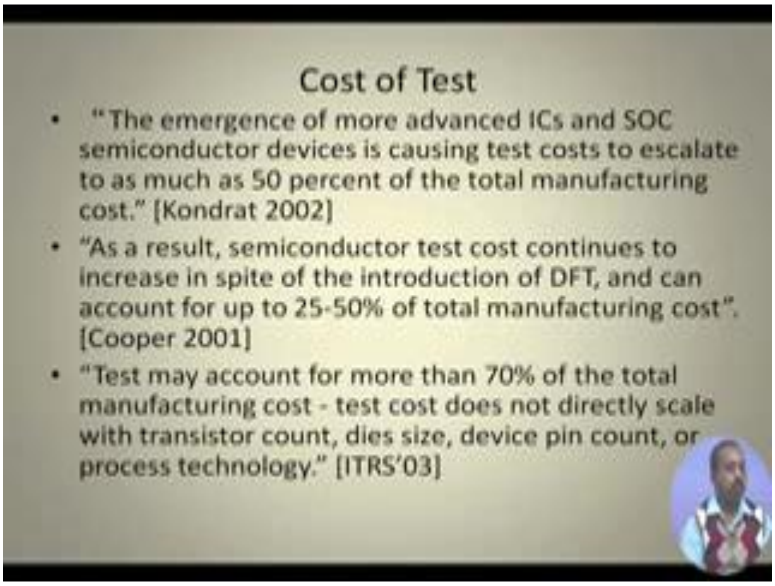
And third important point is high test data volume and test time. So, high test data volume because large number of patterns for every core and there are a large number of cores, so total test data becomes very large. So, can we store, so much of test data in the

ATE memory, so that becomes a fundamental question. If the answer is no, then we have to use some sort of test compression mechanism, test vector compression mechanism. And then as soon as we do that then there is a decoding stage that will be coming, so before applying the coded patterns to the core for testing we have to decode them and get back the original pattern set. So, that way it is going to happen that this test time and test data volume they are going to be an important factor.

So, even if the ATE allows you to store a large volume of data test data in it test large number of test patterns in it, so you need to transport those test patterns from the source to the core under test. So that itself will take a number of clock cycles number of ATE cycles to transfer over the access mechanism, so that way the test time will become high. So, high test data volume has got two problem one is the ATE memory requirement will be high, and the second point is the test time for the chip, so that will also become very high.

So, this is a typical example like NXP Nexperia processor PNX 8550 SOC, so it has got, so many flip flops 274 embedded cores, 10 million logic gates and 40 million logic transistors, so that is a huge amount of space huge amount of complexity that we have. And now we must be able to test such complex systems in a SOC paradigm.

(Refer Slide Time: 12:12)



Cost of Test

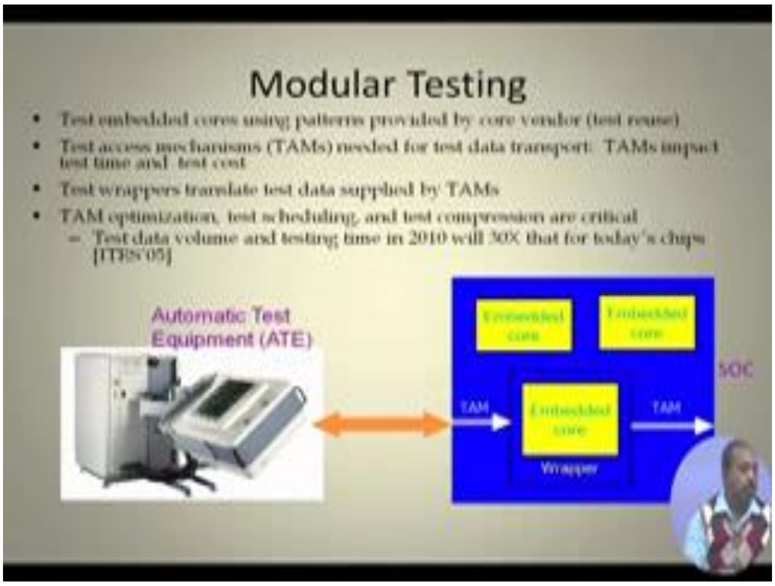
- "The emergence of more advanced ICs and SOC semiconductor devices is causing test costs to escalate to as much as 50 percent of the total manufacturing cost." [Kondrat 2002]
- "As a result, semiconductor test cost continues to increase in spite of the introduction of DFT, and can account for up to 25-50% of total manufacturing cost". [Cooper 2001]
- "Test may account for more than 70% of the total manufacturing cost - test cost does not directly scale with transistor count, dies size, device pin count, or process technology." [ITRS'03]

Cost of test: The emergence of more advanced IC s and SOC semiconductor devices is causing test cost to escalate to as much as 50 percent of the total manufacturing cost. So, this was a statement in 2002, so now it is 2016, so naturally the cost has really become much higher 70 almost 60 to 70 percent of the manufacturing the cost the test cost has gone up. As a result semiconductor test cost continues to increase in spite of introduction of DFT and it can account for up to 25 to 50 percent of total manufacturing cost. So, a test cost becomes a huge part o though we have got DFT scan and all that, so they are going to make this testing process easy, but the sheer complexity or sheer volume of test itself is making the process very costly.

And test may account for more than seventy percent of the total manufacturing cost. So, this is that I have already said. And test cost does not directly scale with transistor count, die size, device pin count or process technology. So, this is another problem. So, what happens is that, so for a VLSI manufacturing process, so as you are increasing the number of chips that you are producing, so the per unit cost comes down. So, we have for VLSI chip design and manufacture, so we have got two cost, one is the non recording engineering cost which is the design part of it, and then per unit we have got some cost. And since we are doing this in a massively parallel way the large number of chips are manufactured simultaneously, so this non recording engineering cost is high, but this unit cost is low, so that way overall cost comes down while we manufacture them in large volume, so that has happened. So, the designers they are happy with producing more and more number of chips with the same design.

However each of these chips that are being produced are to be tested, so you cannot leave one chip untested. So, this increase in the number of chips that are being produced, so also have proportionately increases the test type, so test cost will go up, it does not scale down or rather scales up with the number of chips that we are going to fabricate that we are going to manufacture. So, we can say in some other sense that with the increasing transistor count, so test cost also increases with the number of dies, it is increases, number of device pin count it is increasing; that means we have got more functionality to be tested. And with the process technology, we are putting more and more chips on fabricated simultaneously, so the cost is also increases.

(Refer Slide Time: 15:16)



Modular Testing

- Test embedded cores using patterns provided by core vendor (test reuse)
- Test access mechanisms (TAMs) needed for test data transport: TAMs impact test time and test cost
- Test wrappers translate test data supplied by TAMs
- TAM optimization, test scheduling, and test compression are critical
  - Test data volume and testing time in 2010 will 30X that for today's chips [ITRS'05]

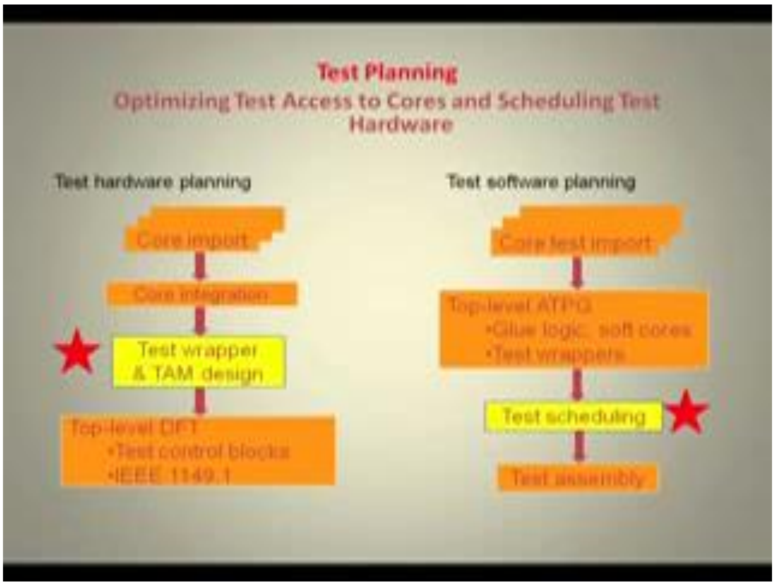
So, what is required is some sort of modular testing. So, test embedded cores using patterns provided by core vendors, so that is definitely there, so we cannot avoid this situation. And test access mechanisms needed for test data transport. So, what is happening is that, so this is the ATE. From ATE, it has to transfer data to the embedded core via some test access mechanism or TAM, and again the responses are to be collected through some TAM and it has to be given back to the ATE and then this ATE will again analyze it. So, test access mechanisms needed to for test data transport, so TAMs will impact test time and test cost.

So, if you can have a wider TAM then you can transfer more test bits parallely as a result test time will be less, but wider TAM means it will be requiring more amount of silicon area. So, as a result and that TAM has to be routed, TAM is not feeding a single core, so it is feeding a number of cores, so that TAM also needs to be routed through the chip. So, as a result this becomes a very important constraint. So, the test cost will increase because I have to provide more amount of the silicon, more amount of footprint on to the silicon for the TAM or holding the TAM.

Then this test wrappers they translate test data supplied by TAMs, so TAM is having the test access mechanism or TAM, so it has got a fixed width say 32 bit width. Now, each of these cores may not have 32 input some of them may have say 100 input, some of them may have say 64 input, some of them may have 90 input like that, so we need to do some sort of adjustment between the TAM lines and the number of inputs and outputs for the cores. So, that is how this test wrappers will take care, so it will translate this test data that is supplied by TAM into a format by in which you can apply the pattern to the individual components in the core.

This TAM optimization test scheduling, test compression these are very critical issues and test data volume and test testing time in 2010 was expected to be 30 x of that of today’s chips that was predicted in ITRS 2005, so that is that is crossed now. So, there is a huge increase in the test data volume and the testing time, so that is the issue, so we have to handle such complex situation.

(Refer Slide Time: 18:01)



Test Planning

Optimizing Test Access to Cores and Scheduling Test Hardware

So, we have to know some sort of test planning. So, we just cannot leave it to the adhoc solution, so we have to do some sort of test planning. So, we need to optimize test access to cores and we also need to do scheduling of test hardware. So, how this is done, so we have to do some test hardware planning, so test hardware planning, so cores are to be taken, so cores are for a particular design, so cores are imported, the cores are integrated, so how the cores will be connected between each other, so that is basically the designer part. Once that has been done each of these cores they need to be test wrapped, so we have to put the we have to design the corresponding test wrappers for each of these cores, and then we have to design the TAM like how these cores will be accessed in a test access mechanism. So, how they are going to be utilized in a test access mechanism, so that is the TAM design has to be done.

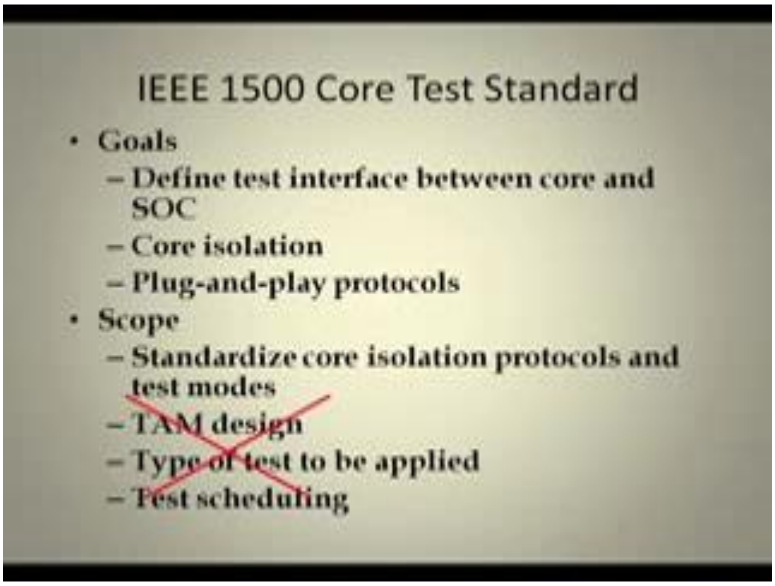
And after that we have got this top level DFT, so that will have this test control blocks IEEE 1149.1, so they are to be integrated. So, this is the thing. So, you see that it starts much early in the manufacturing process, so after the cores are been imported and they are been integrated into the design, so we know what are the cores, so what are their test requirements and all that, and immediately we have to start doing the other operation that is wrapper design, TAM design etcetera. And on the software side, so once the core has being imported side by side we also get the corresponding test data. So, test patterns and the correct responses, so they are also provided by the core vendor, so they have given the core as well as the test part test pattern part.

And then at atop level ATPG can be done, so that we can generate the test patterns for the glue logic. And basically the glue logic are the portions which are extra small logic that are added for making the functionality of the chip complete may be this requires some translation of data from one format to another format or some extra small, small logics to be added inside the design to make the design complete. So that is the glue logic. But for testing this glue logic, we have to apply test pattern, so to know what is the test pattern for testing this glue logic, so we have to take help of some ATPG.

And also sometimes, some cores are available as soft cores. Soft core means the design is provided at the behavior level itself or may be at most, yeah it is provided at the behavior level. So, if it is that then we can generate test patterns at that point, so it is not at a layout level, but at a much higher level which at the level of may be at the level of net-list may be at the level of behavior. So, I can run some ATPG tool to generate the test vectors for them, so this for soft cores also, we can do it. But for hard cores we cannot do anything for hard cores we have to rely on the patterns provided by the core vendor. So, these test wrappers are to be designed.

And then we have to go for the scheduling policy. So, we have to find out like how when are you going to schedule which core for testing, and we have several constant like say this wrapper this TAM should be available to which this core is attached. Then the power limit should not be violated the thermal power should not be violated then there may be some precedence constant, so keeping all these things into mind, so we have to come up with a test schedule. So, that gives rise to a number of alternative works that are been reported in the literature on this test scheduling algorithms. And ultimately the test assembly; test assembly means that I have got this scheduling algorithm, I have got the compression mechanism, I have got the response analysis mechanism, so all these things are to be integrated together for the whole system to operate, so that is the test assembly process. So, these are the two important points, the wrapper design TAM design and test scheduling these are the two important points to be addressed in test planning.

(Refer Slide Time: 22:31)



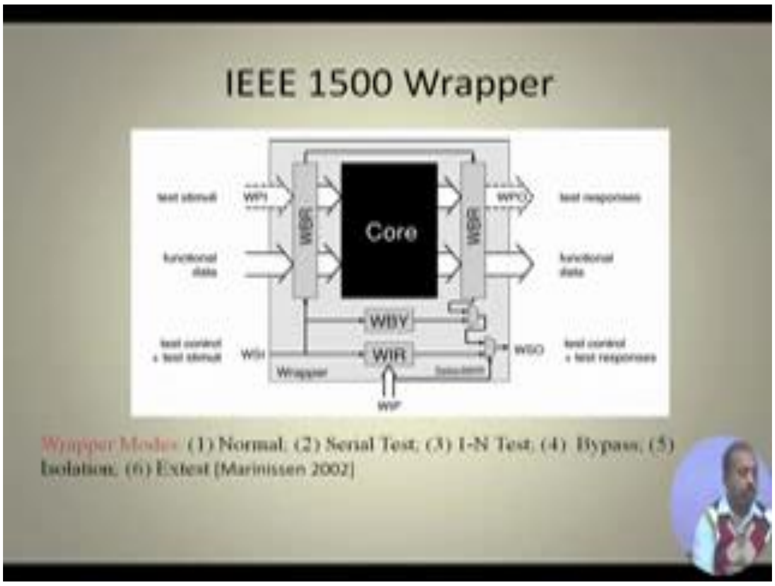
IEEE 1500 Core Test Standard

- Goals
  - Define test interface between core and SOC
  - Core isolation
  - Plug-and-play protocols
- Scope
  - Standardize core isolation protocols and test modes
  - ~~TAM design~~
  - ~~Type of test to be applied~~
  - ~~Test scheduling~~

So, 1500 core test standard, so this is provided by IEEE. It defines test interface between core and SOC, provides core isolation and plug and play protocols, so these are the goals. And the scope is to standardize core isolation protocols and test modes, but so this tells like how can we make a core isolated from others, and apply the test pattern to that core only. So, we have seen previously that rest of the cores put into bypass mode and whenever we are transferring test pattern for a particular core, so we put others in bypass mode, and then that test pattern is applied.

So, either serially or parallely whatever we can think about, but the rest of the thing like the TAM design type of test to be applied test scheduling, so these are not answered in the 1500 standard. Because this is going to be this is TAM part may be the parallel access, so that is an optional part of 1500, so though the instructions and every things are provided, but it does not tell how to design a tam, so that is left to the designer. So, the test engineer may come up with different TAM architecture and accordingly after the TAM architecture has been designed can decide upon the test scheduling problem how to schedule different tests for the application.

(Refer Slide Time: 24:00)



IEEE 1500 Wrapper

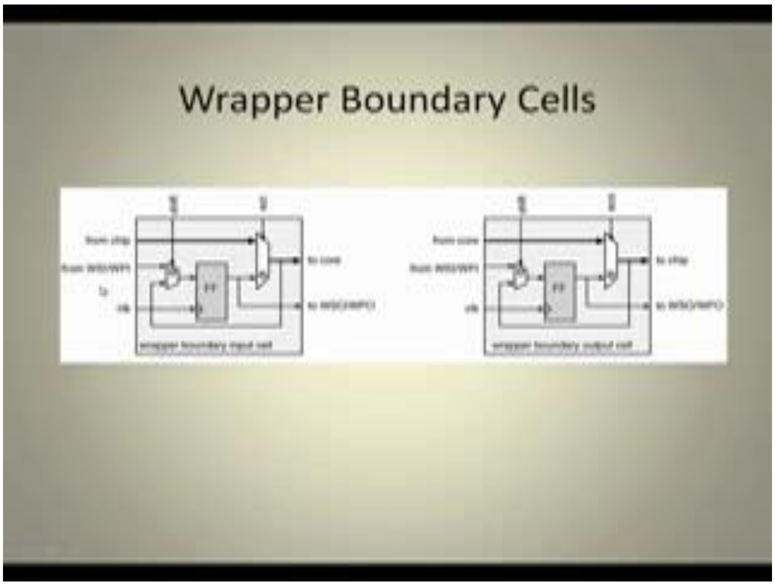
Wrapper Modes: (1) Normal; (2) Serial Test; (3) 1-N Test; (4) Bypass; (5) Isolation; (6) Extest [Marinissen 2002]

So, this is that 1500 wrapper, so we have got the normal mode of operation, serial test, 1 to N test, bypass, isolation and extest, so these are the various modes. So, we have already discussed this thing that this WSI the serial input can be used for giving the serial data some we may like to configure this boundary scan registers, boundary scan sales they are through the serial input, and or we can have some parallel input as well. So, this test stimuli can be applied parallely also this wrapper parallel input and wrapper parallel output, so using that also this WBRs can be configured. And WBR can also get input from the functional data that is in the normal mode of operation, so it can get the functional data and then the functional data can be applied to the core, and they will be applied functional, so that is the normal operation of the system.

In the test mode, we have got serial tests in which through this WSI line, the test patterns can be loaded into this WBCs that we have here. Or we can have 1 to N test, so there is a multicast sort of thing, so one from one point, so I want to test inputs parallely, so all of them will be loaded simultaneously, all those scan chains will be loaded simultaneously. Or it may be a bypass mode, where this through this WBY wrapper bypass register, so it will go to WSO, so this that mode is there.

Then isolation, so we can put into the safe mode basically that safe mode that we had discussed previously, so that is the isolation mode. And this extest, so if you are trying to test the interconnect, so you can put this into extest mode and where this output, so we can load this WBR. And after that whatever interconnects are there this functional data wherever it connects o the next core, so there we can check whether the value have been BIST properly or not, so that way we can have some extest mode also we can test this glue logic part using this extest mode that we have seen previously.

(Refer Slide Time: 26:09)



Wrapper Boundary Cells

So, this is the wrapper boundary cell structure, so quickly looking into it. So, from chip this is the functional input, and we can have this WSI. So, for serial inputs, so we can have this WSI or WPI, so they will be coming. And this shift bit will be given, so if it is 0, then this WSI, WPI they will be coming to this flip-flop and they will be loaded. Otherwise, if you want to load some functional value then from the core, so this value may be taken back and put here. So, we have got this type of boundary input cell and this is the wrapper boundary output cell. So, in the output cell, so this is going to the WSO or WPO line, and this is going towards the chip, so that way we can design the output cell. So, here from the WCI line, so it is controlling whether the input will come from chip or it will come from this serial this wrapper lines WSI or WPI that will be applied to the core. I will continue in the next class.