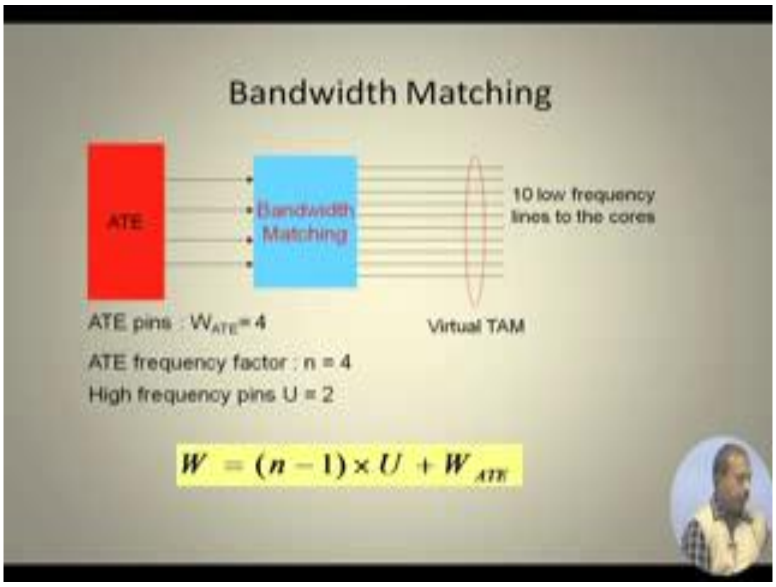
Digital VLSI Testing

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

Lecture - 50
System/Network – On-Chip Test (Contd.)

(Refer Time: 00:18) matching problems so when we have got this fast ATE and this core testing is slower so this ATE can provide data at a higher rate and then we have to have this testing to be done at a lower frequency. So, that is the problem of bandwidth matching.

(Refer Slide Time: 00:30)



Bandwidth Matching

ATE pins : $W_{ATE} = 4$
ATE frequency factor : n = 4
High frequency pins U = 2

$$W = (n-1) \times U + W_{ATE}$$

So, we have the ATE that has got number of channels going out of it, and then some of the channels are high frequency lines; like in this ATE suppose there are two lines which are high frequency lines and there are two lines which are low frequency line. And then we have got this the overall ATE is it has got four pins of W ATE equal to 4; the width of the ATE is equal to 4; and ATE frequency factor n equal to 4. So, frequency factor means compare to the core at what is the high frequency rate for the ATE. So, the ratio of the two frequencies is a 4. So, ATE is working at a 4 times faster than this core for the testing purpose.

And number of high frequency pins equal to 2 U equal to 2 that we have just seen. Now, so W into ATE; so this formula has to be satisfied. So, W into ATE is the width of the

ATE and multiplied by f ATE. So, the number of bits that are transferred through this channel per unit time, so that should be same as W into TAM divided by a into the f TAM. So, number of bits which are actually reaching the core, so they should be matched.

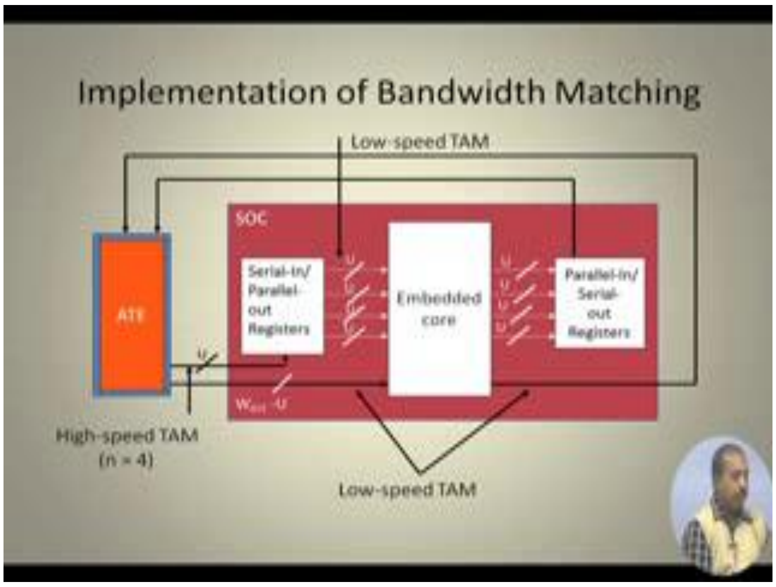
So, there is naturally there is a mismatch here, if you just extend it so this W ATE into f ATE. So, W ATE. So, if you just expand it, so it is become this. High frequency part that is operating at this f frequency of ATE, and W TAM minus U so that is the number of pins which are occur operating at a lower frequency that is operating at the frequency of this lower frequency TAM So, this part does not create any problem, so for this speed matching between this ATE and this core.

However, the first part is going to be extended because this f ATE is n times f TAM because this ATE is n time faster than TAM. So, I can substitute this f ATE by n into f TAM, so this expression becomes something like this. So, W is n minus 1 into U plus W ATE. So, if you solve this problem expression then this will turn out to be this. So, W must be equal to n minus 1 into U plus W ATE. So, that actually gives the expansion in the number of channels that we can get at the ATE the core end or the SOC end.

So, this is the bandwidth matching part which will be doing these things. So, it will be increasing the rate at which this data is been fed to the channels. Now, this virtual TAM, so it has as if I have to 10 low frequency lines to the cores. So, by looking into this formula so n minus 1, so n is equal to 4 and U equal to 2; so 4 into 2 that is 4 minus 1 into 2 that is 6 plus this W ATE is 4. So, that is total 10. That means, on this virtual line I have got as if the 10 low frequency lines to the cores.

So, though there are four channels coming from the ATE, but for the scheduling purpose we see we have got 10 lines to be fed to the SOC. So, that makes it very flexible; like I when I am doing this test scheduling. So, I can assume that whatever technique I am following partitioning or bin packing etcetera so that can take help of this 10 low frequency lines.

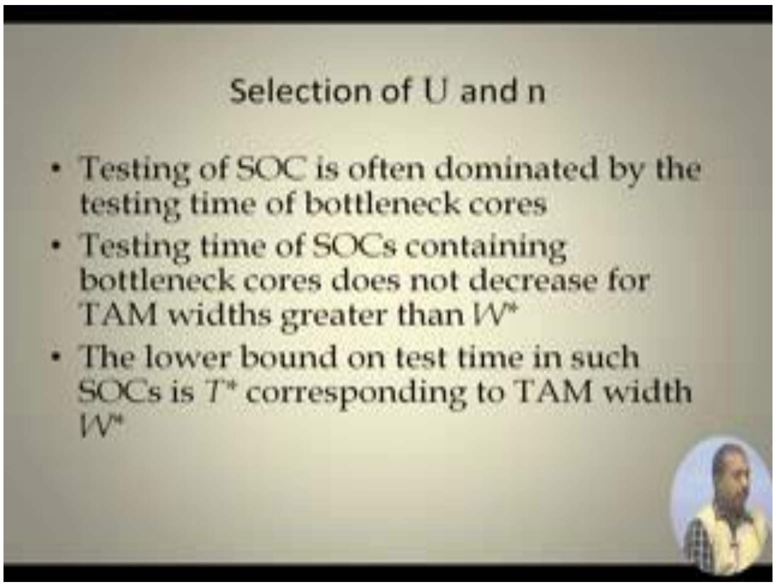
(Refer Slide Time: 04:27)


Implementation of Bandwidth Matching
Low-speed TAM
SOC
ATE
Serial-in/ Parallel-out Registers
U
Embedded core
Parallel-in/ Serial-out Registers
$W_{ATE}-U$
High-speed TAM (n = 4)
Low-speed TAM

So, how this will be done? So this high speed this bandwidth matching part from the ATE we have got this U high speed lines. So, U high speed lines, they are fed to a serial in parallel out register. So, these U lines are coming here. And then this each of these lines are feeding this U number of lines parallely. So, this U bit patterns are going parallely to the core. And then this W ATE minus U so that many lines are coming here. So, this part is generating U lines and this part is W ATE minus U so that is fed here; this is a low speed TAM part. So, that is being fed to the core and then this is actually, this U lines plus W; these U lines will again be converted via this parallel in serial out register that will be converted into serial value and that will be transferred back to the ATE.

So, we have got this low speed ATE and low speed TAM lines here, we have got low speed TAM lines at this point, you have got high speed TAM line here; so which is implementing the bandwidth matching issue. So, the scheduling policy can be benefited by having this thing.

(Refer Slide Time: 05:46)

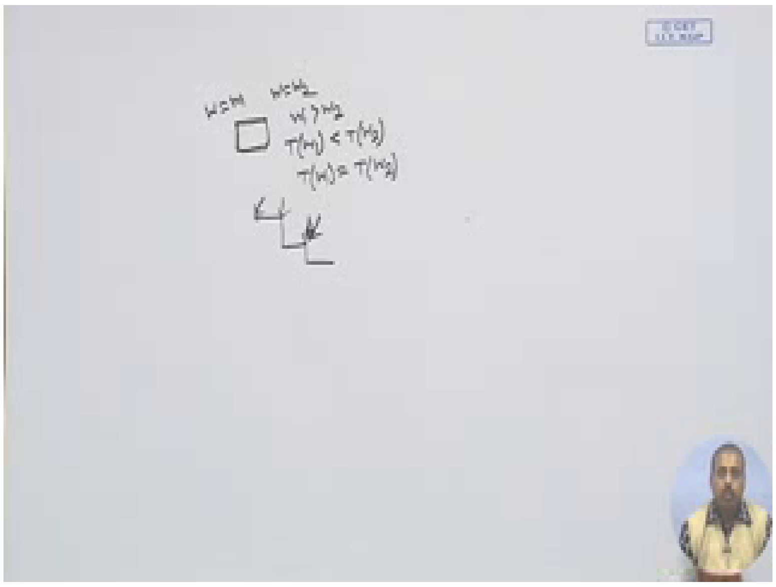


Selection of U and n

- Testing of SOC is often dominated by the testing time of bottleneck cores
- Testing time of SOCs containing bottleneck cores does not decrease for TAM widths greater than W^*
- The lower bound on test time in such SOCs is T^* corresponding to TAM width W^*

So, how can we select this U and n; like how many high speed lines and how it is the frequency fact. So, testing SOC is often dominated by the testing time of bottleneck cores. So, bottleneck cores are normally; what is the situation?

(Refer Slide Time: 06:05)



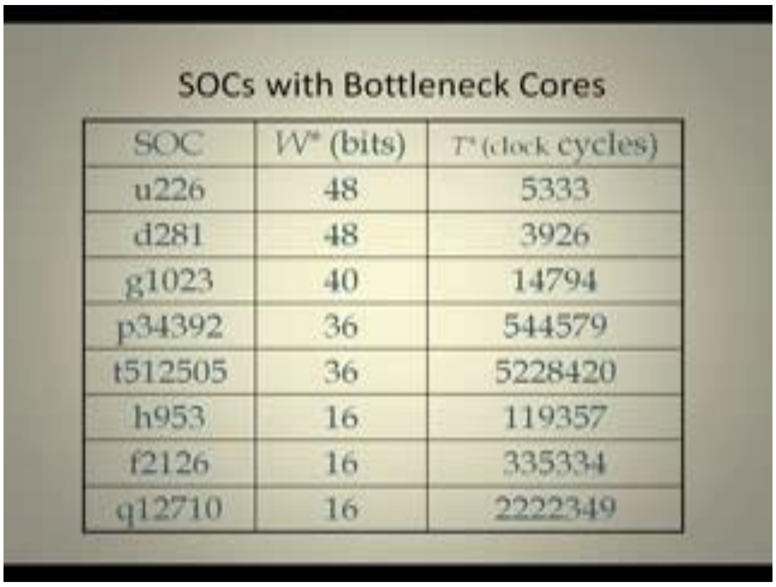
Like if I have a core then if this core is given more TAM width. So, it is given say sometime say width W equal to W 1 and sometimes it is given W equal to W 2. So, if W 1 is more than W 2 then the corresponding test time T of W 1 is naturally expected to be less than T of W 2. Now we have seen that due to this Pareto-optimality this always does not happen. So, in many cases what happens is the T of W 1 remains equal to T of W 2.

But in general we have got this staircase type of behavior, so this Pareto-optima-l points are occurring at each of this points sorry this one then this one, so it occurs at this points. However, it may so happen that for some cores this type of Pareto behavior does not exist. So, whatever be the TAM that we assigned to this core it takes a huge amount of time. So, this may be due to some module, some core, some functional module inside the core that takes a huge amount of time and it is not dependent on the; maybe some of some scan chain has been kept there which is very long. As a result this whatever be the TAM line we assign to it, so it is going to be dominated by the length of the scan chain.

And this assigning mod number of TAM bits to the core does not help in improving its test time. So, that may be the situation and that type of cores, so they are called bottleneck cores because whatever be the amount of test width assigned to that core the test time of it does not decrease. So, total test time is of the SOC starts be getting dominated by the test time of that particular cores. So, this type of cores they are known as bottleneck cores. So, testing time of SOCs containing the bottleneck cores does not decrease for TAM width greater than some value W star. So, if you increase; even if you assign more TAM width to the SOC then also it does not improve because that bottleneck core it will stop the improvement in test time further.

The lower bound on test time in each in such SOCs is T star corresponding to the TAM width W star. So, we cannot reduce the test time beyond this T star value. So, if I got in my scheduling, so if I got some test time results which are close to T star then we should be happy that we will not be able to beat this bottleneck this lower bound W T star. So, we know that perhaps we have reach the optimum solution for this particular SOC.

(Refer Slide Time: 09:05)



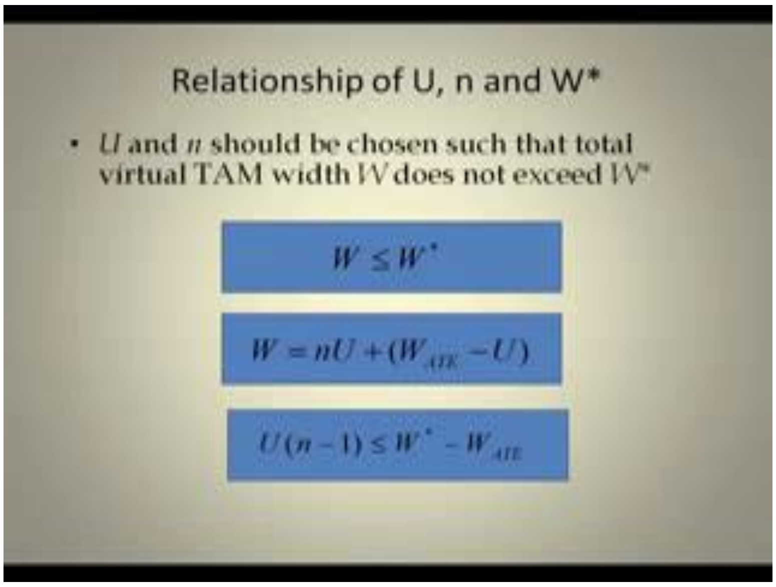
SOCs with Bottleneck Cores

SOC	W^* (bits)	T^* (clock cycles)
u226	48	5333
d281	48	3926
g1023	40	14794
p34392	36	544579
t512505	36	5228420
h953	16	119357
f2126	16	335334
q12710	16	2222349

So, these are the some bottleneck cases like say u226. So, this particular SOC they are all from the ITC 02 benchmarks so that we are talking about. So, u226 it has got a bottleneck at 48 bits. So, even if you increase it test time W width beyond 48 test time cannot be reduced, so it remains at 5333.

Similarly d281, so there also it remains at 3926. So, all this cores, all this SOCs have been mentioned here and for each of them you see that there are some bottleneck core due to which even if you increase the W star this TAM width beyond the this W star bits it will not improve the test time beyond this T star values; clock cycles in terms of ATE.

(Refer Slide Time: 09:55)



Relationship of U, n and W*

- U and n should be chosen such that total virtual TAM width W does not exceed W^*

$$W \leq W^*$$

$$W = nU + (W_{ATE} - U)$$

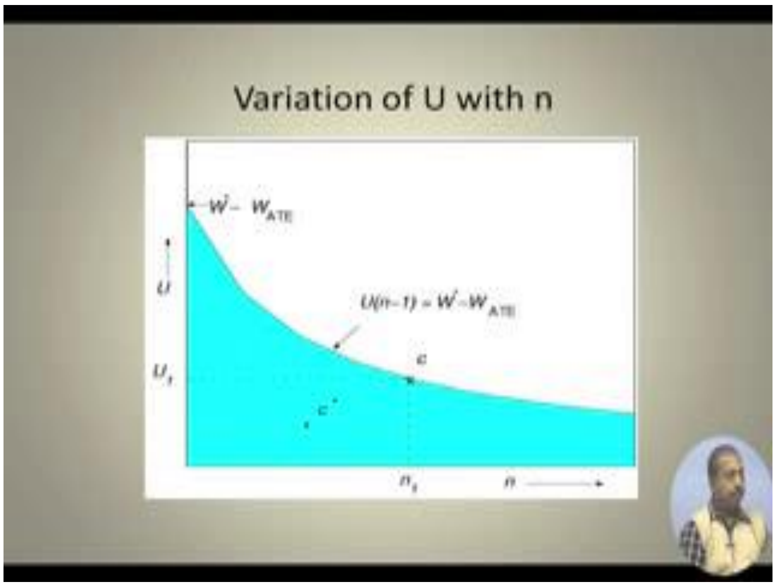
$$U(n-1) \leq W^* - W_{ATE}$$

So, if we try to find a relationship between this U n and W star then this U and n they should be chosen such that the total virtual TAM width W does not exceed W star, because incase in beyond W star is not going to help. So, it is not going to decrease the test time for that core anymore. So, the virtual TAM width W it must be less or equal W star. And again you know the W is given by this formula n into U plus W ATE minus U; where n is the frequency factor, U is the number of high speed channels and this is the; so W ATE minus U that gives the number of low speed channel. So, that is the total TAM width W that we get in a virtual TAM environment.

So, if you substitute it here then we get U into n minus 1 it should be less or equal W star minus W ATE. So, this way you can choose these values of U and n such that this constraint is satisfied. So that gives us a guideline like in a virtual TAM based environment. So, may be U is fixed by the ATE number of high speed channels that it can provide, so that is specified by the ATE.

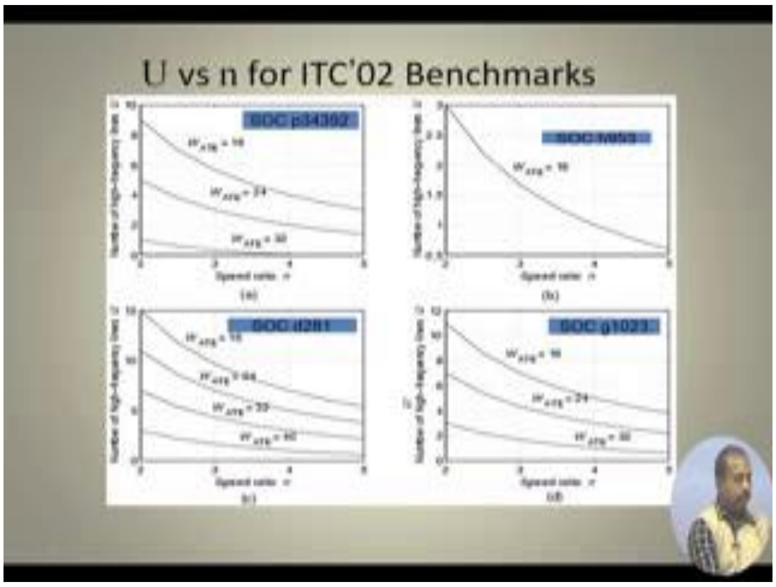
So, we can do a choice of this n. Or if we have got this thing, if we have got that frequency relationship available the speed the frequency factor between the ATE and the chip then we can put that value and try to get what is the value of U. And the U should not; may be ATE is providing a number of high speed channels, but we may not be able to use all of them so that is possible. So, you can go for we can use them as low speed channels and the remaining ones are the high speed channels.

(Refer Slide Time: 11:49)


Variation of U with n

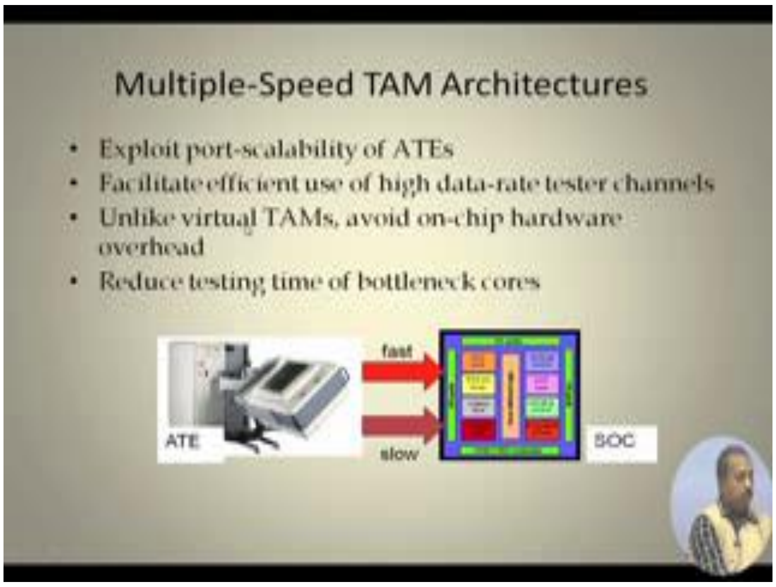
So, this graph actually shows the variation of U with n. So, if you increase the value of n this U value starts decreasing; so this U into n minus 1, so this is actually the plotting of that U into n minus 1 satisfying this relationship. So, if it is at this point n 1, so this is the corresponding U 1. So, this may be one choice and beyond this it is not improving further so we can go for that.

(Refer Slide Time: 12:19)


U vs n for ITC'02 Benchmarks

So, for U versus n for different ITC 02 benchmarks circuits have been chosen accordingly the proper point can be chosen. So, which u, which n and which U should be used that can be chosen by the test engineer by looking into different points, looking into the different graphs so it can be chosen.

(Refer Slide Time: 12:43).



Multiple-Speed TAM Architectures

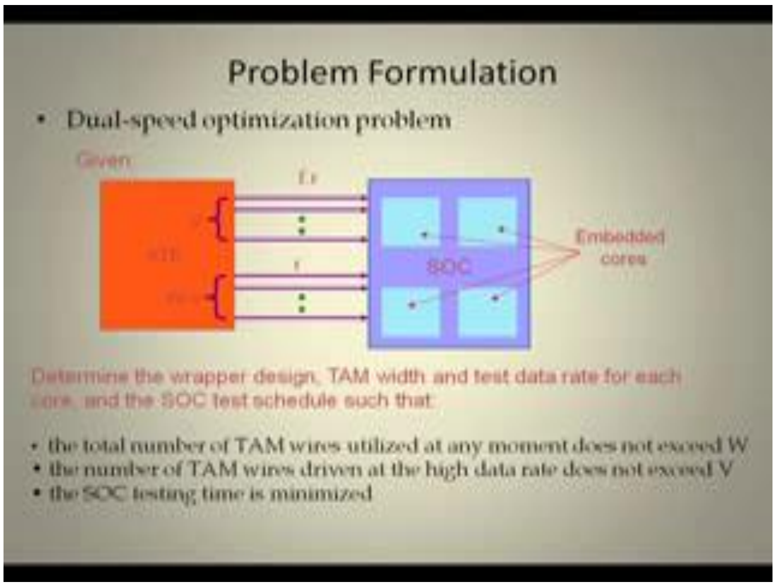
- Exploit port-scalability of ATEs
- Facilitate efficient use of high data-rate tester channels
- Unlike virtual TAMs, avoid on-chip hardware overhead
- Reduce testing time of bottleneck cores

ATE
fast
slow
SOC

Another possibility that we can have in the ATEs is known as multiple speed TAM architectures. So, multiple speed TAM architectures they exploit the port scalability features of ATE. So, port scalability means that it facilitate efficient use of high data rate tester channels, some of the tester channels are fast, some of the tester channels are slow. However, it does not use on chip hardware overhead; like incase of virtual TAM what was required was that there was a requirement to have that bandwidth matching hardware so which will be doing this shifting and all that, serial to parallel converters were necessary on the chip so that we can do the speed matching between the virtual between the ATE and this SOC and using this virtual TAM environment.

So, this will try to reduce the test time for this SOC by using some fast and slow channel. So, test time also gets reduced for the bottleneck cores. So, that is another advantage that we have in this particular policy; multiple speed TAM architecture.

(Refer Slide Time: 14:01).



Problem Formulation

- Dual-speed optimization problem

Given:

f.r
V
ATE
r
W-V
SOC
Embedded cores

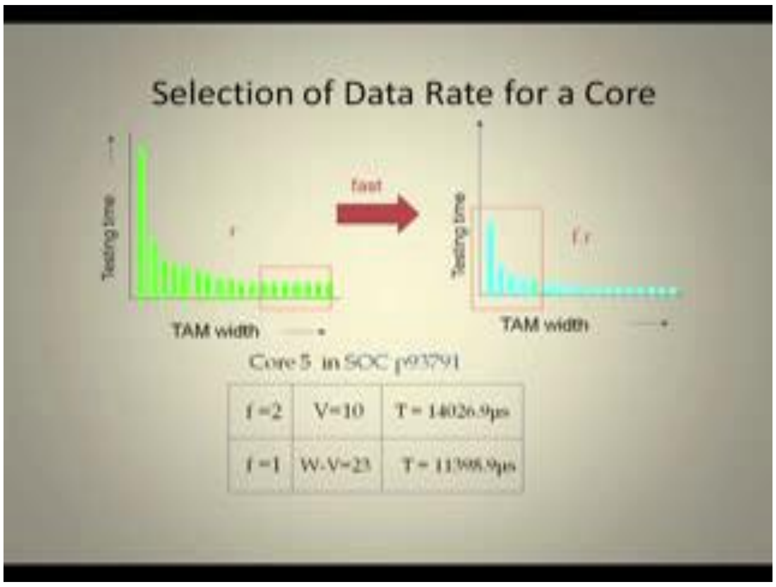
Determine the wrapper design, TAM width and test data rate for each core, and the SOC test schedule such that:

- the total number of TAM wires utilized at any moment does not exceed W
- the number of TAM wires driven at the high data rate does not exceed V
- the SOC testing time is minimized

The basic idea is like this that; suppose we have got a dual speed optimization problem. So, this ATE has got V number of channels which can operate whose data rate is high f into r and this one it has got W minus V low speed channels so that is operating at a lower data rate r; and these are the embedded cores that we have. So, with the problem is to determine the wrapper design for individual cores TAM width and test data rate for each core. So, at which rate it should be tested; where lower at r at rate one or at a rate f. So, that is the rate has to be chosen.

And the SOC test schedule such that total number of TAM wires utilized at any moment does not exceed W, so, that is always there. Then the number of TAM wires driven by the high data rate does not exceed V. So, this high data rate only V and the low data remaining will be low data rate; W minus V will be low data rate and the SOC testing time is minimized. So, these first towards the constraints, total TAM wires should not be more than W and this high speed wires more should not be more than V so these two are two constants and the optimization is to minimize the SOC testing time.

(Refer Slide Time: 15:23)



Selection of Data Rate for a Core

Core 5 in SOC p93791

f =2	V=10	T = 14026.9μs
f =1	W-V=23	T = 11398.9μs

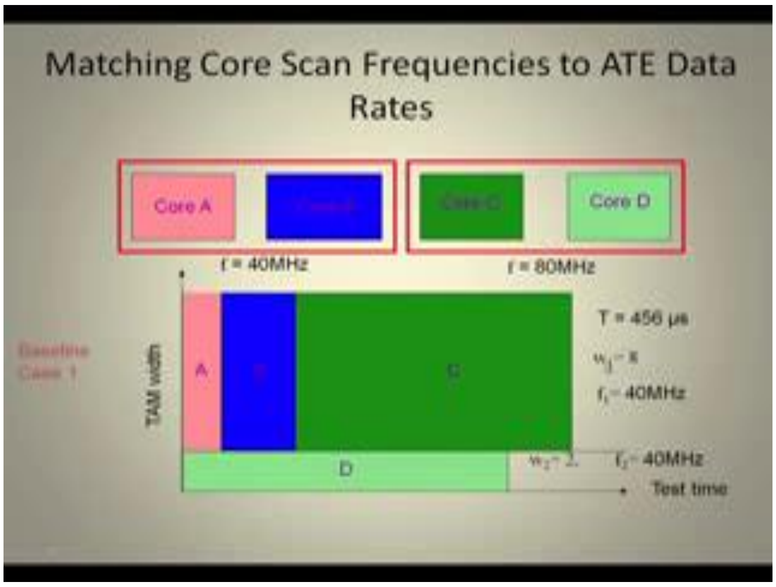
So, how can we do this selection? Suppose we have got this Pareto-optimal distribution like as we are increasing the TAM width at; so this is the tester. So, at r data rate it happens to be like this. So, if the testing is done faster then we see that this test time reduces significantly. So, this f into r; so data testing time will even if I use low lower TAM width so test time is even smaller. So, maybe for normal TAM lines I have to use some this type of TAM width, because beyond that this test time is quite high. So, I do not have any choice, so I have to use this mode number of low speed channels.

But when I have got this high speed channels then I do not need to use this higher number of channels, higher number of higher TAM width. So, low TAM width may be sufficient because the test time that we get is reduced significantly. So, I can use say some values in this range so that it is it is done.

So, core 5 of this particular SOC p93791. So, if you are doing at a slower data rate then this W minus V equal to 23 so that if you put it on a slow channel then suppose it has got 23 lines then the test time becomes 11398.9 microsecond, whereas if you put it on a faster line giving 10 faster lines, so it can test it at 14026.9. So, you see the number of TAM lines requirement has reduced from 23 to 10, so remaining lines are available for doing something else.

And this test time increase may not be very significant, so from 11000 it is increasing to 14000, but in this remaining time we can may be able to schedule some other tests, some other core tests so that this test time becomes better utilize and overall SOC test time reduces. So, we can do some selection like that.

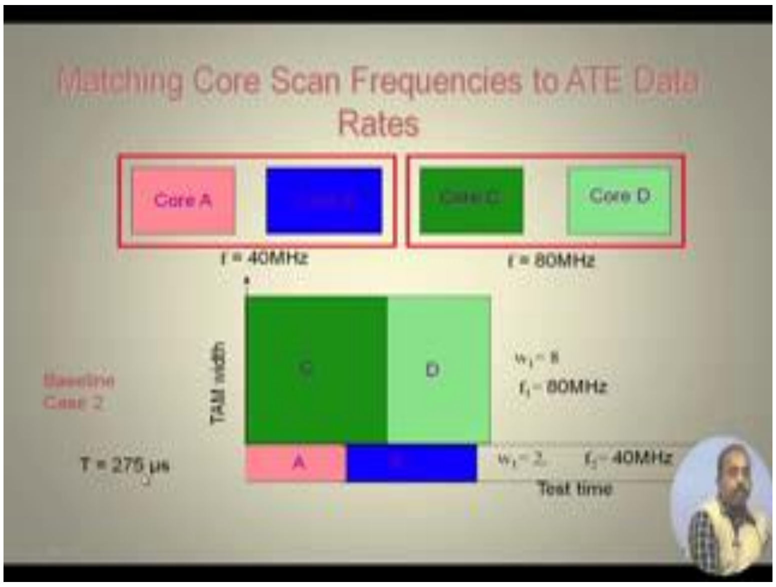
(Refer Slide Time: 17:49)


Matching Core Scan Frequencies to ATE Data Rates

So, may be after how to match this core scan frequencies to ATE data rate. So, you may decide that this core A and core B, so they will be operating at 40 megahertz and core C and core D they will be operating at 80 megahertz. So, what we do with this core D we put it on say on this 40 megahertz line, so this is a W 2 width is 2 and it put it on a 40 megahertz frequency. And this core A they are put on W 1 so that W 1 equal to 8. So, the width that test width data base is test width is 8.

So, this is the base line case that is I have this divided the 10 lines of TAM that we have into two: the first one is 8, the TAM width is 8 and the second TAM width is 2. So, the schedule it generates is that core A is scheduled here, core B is scheduled here, core C is scheduled here and core D is scheduled here. So, overall test time is T equal to 456 microseconds. So, it aims at this point, so that is 456 microsecond.

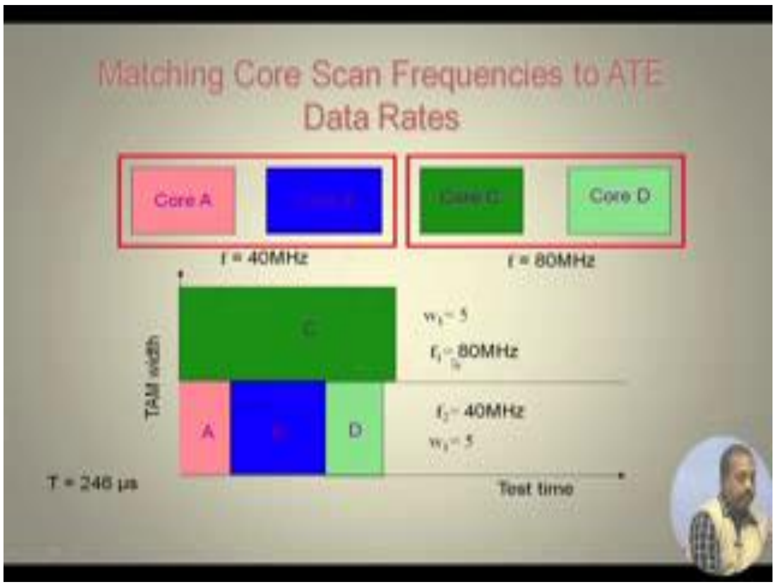
(Refer Slide Time: 19:11)


Matching Core Scan Frequencies to ATE Data Rates

Now in another case if we do the schedule in a slightly different fashion. So, what we do? We put this core C and core D on this 80 megahertz part with the width 8 and this. In the previous example, so both f 1 and f 2 they were 40 megahertz. They were none of them were put on the fast 80, first channel or the first cores.

But here I am putting this one this W 1 equal to 8; so this C and D they are put on faster channels and this A and B they are put on the slower channel. So, that way this C and D they are they were very big rectangles. Previously if you see that they were very big rectangle C and D. So, this putting them on faster channel can reduce the test time. So, that is done here the C and D they are put on faster channels. So, the test time reduces and this overall test time becomes 275 microsecond.

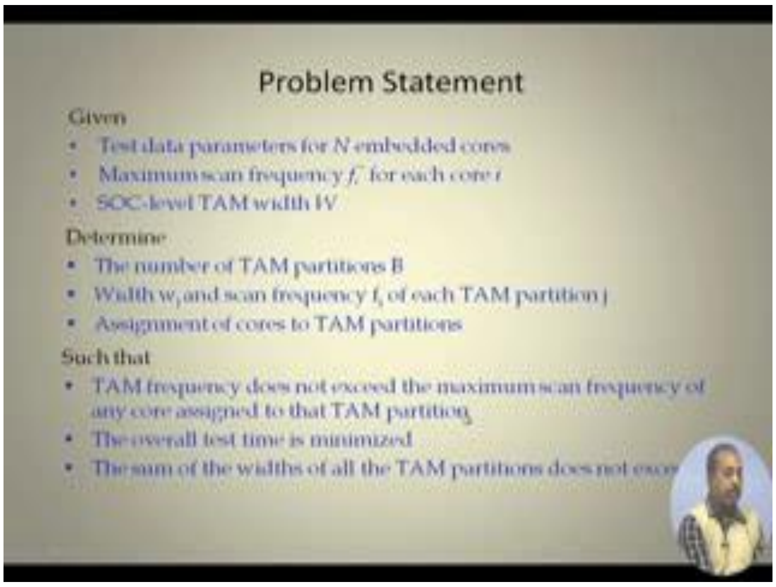
(Refer Slide Time: 20:17)


Matching Core Scan Frequencies to ATE Data Rates

Now, take another case where this W 1, this distribution of TAM is done like this; W 1 is 5 and W 2 is also 5. And then this W 1 operates at this higher frequency 80 megahertz and W 2 operates at lower frequency 40 megahertz. Now this on this 40 megahertz I can; this on this 80 megahertz only core C and D can operate, but it put only core C here and core A B and D all of them are put on this 40 megahertz frequency with W 2 equal to 5.

And then the test time reduces to 246 microseconds. So, in this way if we can match these core frequencies with these TAM frequencies and all that then and do a scheduling properly; so select we can do very good work in terms of this test time reduction.

(Refer Slide Time: 21:14)



Problem Statement

Given

- Test data parameters for N embedded cores
- Maximum scan frequency f_i^* for each core i
- SOC-level TAM width W

Determine

- The number of TAM partitions B
- Width w_j and scan frequency f_j of each TAM partition j
- Assignment of cores to TAM partitions

Such that

- TAM frequency does not exceed the maximum scan frequency of any core assigned to that TAM partition
- The overall test time is minimized
- The sum of the widths of all the TAM partitions does not exce

So, what is the overall problem statement; we are given the test data parameters for n embedded cores, so we have got that testing their test links, their test times and the frequency that which they can be tested and the Pareto-optimal points are definitely required. The maximum scan frequency f i star for each core. So, their frequencies are known at which it can operate, so that we do not do a violation. So, it is not that we put it on a frequency higher than that it can sustain.

And the SOC level TAM width W, then we have to determine the following. The number of TAM partitions how many partitions we are doing. Here interestingly, here it is a partition based approach. So, it is no more a bin packing because of there the flexible TAM is not possible, because I have to the same TAM line it is not possible that sometimes it works as high frequency, sometimes it works at low frequency. So, if the frequencies are same. Of course, within a TAM we can think about this bin packing, and that may possibly be try by first doing this type of assignment. And after that we can we may try to improve that assignment by means of some bin packing, so that may be there. But the basic first level solution it has to do with this partitioned TAM.

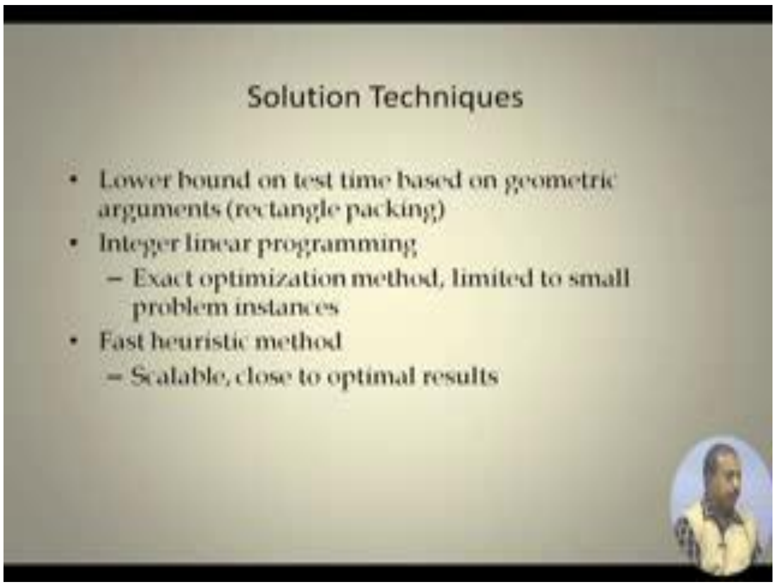
So, we determine the number of TAM partitions B; the width W j and scan frequency f j of each TAM partition j. So, at which what you should the TAM width and what is the frequency of that TAM; and then assignment of cores to TAM partition. So, this is useful if the ATE has got the feature that; second we can tell that these numbers of these lines want to put on this frequency.

So, the ATE has restriction like how many out of the total W number of channels only V number of them can be put on say high frequency, but which V if we have choice. So, then this particular formulation is trying to capture that situation. Otherwise we cannot have a choice of frequency for each TAM partition, but assuming that we have that choice. So, this f j the scan frequency for the partition has to be assigned. And then the assignment of cores to the TAM partitions; so this is the problem statement such that these things are made and the TAM frequency does not exceed the maximum scan frequency of any core assigned to that partition.

So, as we have seen that if a core can operate maximum at 40 megahertz we cannot put it on a 80 megahertz channel, so that is there. And the overall test time is minimized, the test time has to be reduced and the sum of widths of all TAM partitions does not exceed W. So, this is the third requirement, so that is from the physical point of view.

So, second two bullets are fine the last two bullets they are for any problem, but the first part is important that TAM frequency of a TAM it should not exceed the frequencies of the cores that which they are going to be tested.

(Refer Slide Time: 24:39)



Solution Techniques

- Lower bound on test time based on geometric arguments (rectangle packing)
- Integer linear programming
  - Exact optimization method, limited to small problem instances
- Fast heuristic method
  - Scalable, close to optimal results

Solution technique; so there is lower bound on test time based on geometric arguments rectangle packing. So, you can do a rectangle packing and try to see what is the minimum time. We can say that it cannot be reduced beyond that. So, that is the lower bound based approach.

Then there are some approaches that have been reported based on integer linear programming. So, they are actually trying to formulate this mapping problem using this integer linear programming that we have seen in the fixed TAM partitioning case. So, here also that can be extended; so the only thing is that the frequency part will come as a result some additional constraints will get added. But problem is that it is limited to small problem instances. So, it is that that way if the ILP; the basic problem is that it does not work if we have got large number of equations, large number of constraints or large number of variables. So, it cannot be done beyond that.

So, for small sized problems it works, but it gives exact result, so we may be interested in that. And there are some fast heuristic methods that have also been proposed, that gives rise to scalable close to optimal results. So, that they have also be done. Basically the heuristics that have been used for the problems like p a w and p n p a w, so they have also been used for solving the problems under this category. Because what extra thing that we need to do is to do the frequency assignment part. So, I am accepting that rest of the things is same, so they can be used as well for doing this partitioning of TAM and assignment of cores.