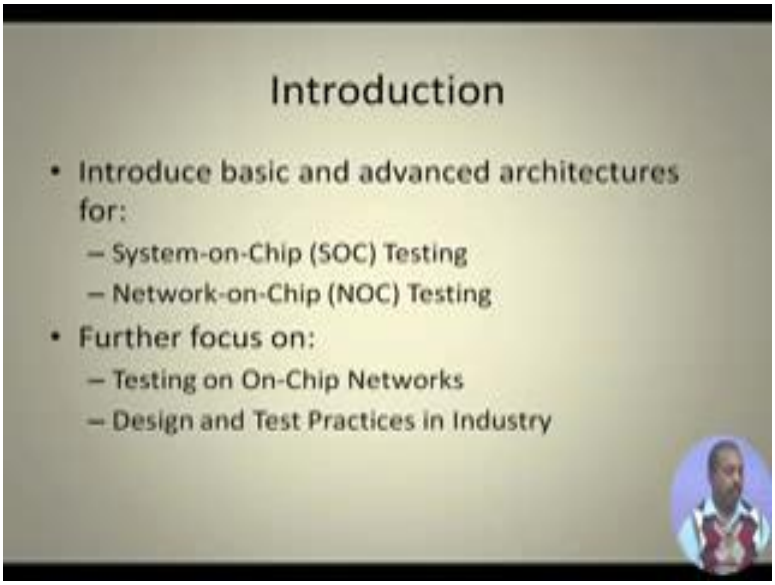
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 45
System/Network - On - Chip Test

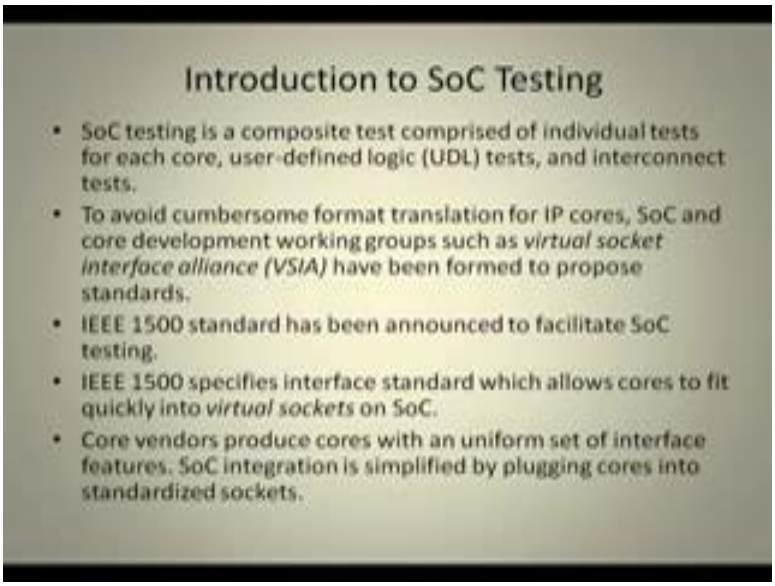
After we have seen this wrapper design IEE 1500 standard and all that so the next state is to look into the system on chip and network on chip architectures. So, system on chip is a design style in which we fabricate the entire system on a particular chip on a single silicon wafer. And in this network on chip what is done is that this communication problem that we get in today's IC, so they are getting addressed by introducing some sort of on chip network to which this individual functional modules can be connected. And naturally testing issues they become more severe in both the cases, so system on chip and network on chip. So, in this part of the lecture, so we will be looking into those issues.

(Refer Slide Time: 01:10)



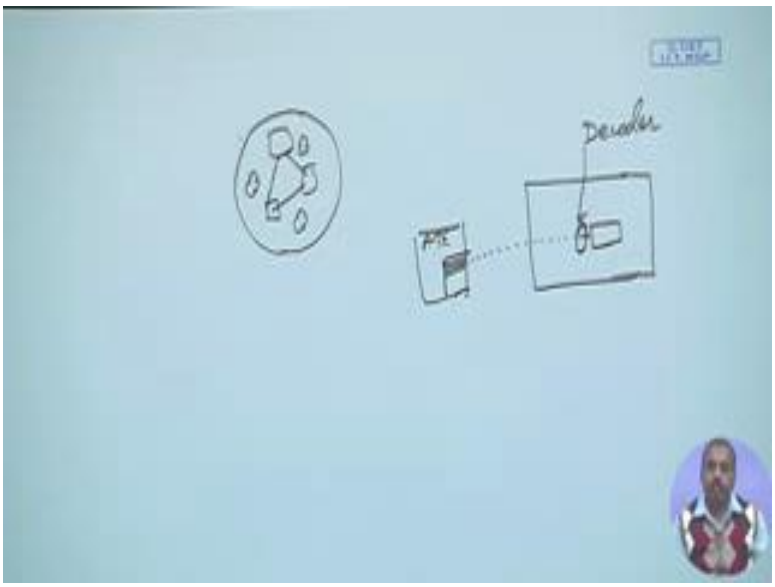
So, what we will do is that we will introduce these basic and advanced architectures for system on chip testing and network on chip testing. And try to focus on this network or this on chip networks testing of on chip network and then some of the design practices followed.

(Refer Slide Time: 01:27)



So, SOC testing is a composite test comprised of individual tests for each core, user-defined logic and interconnect tests.

(Refer Slide Time: 01:41)



So, what is there in an SOC is that in a single chip, we have fabricated all these components. So, the individual cores are there then we have got this interconnects between them and also we have got some extra glue logic may put here and there to make the system complete. So, this testing process, it must test this individual cores, it

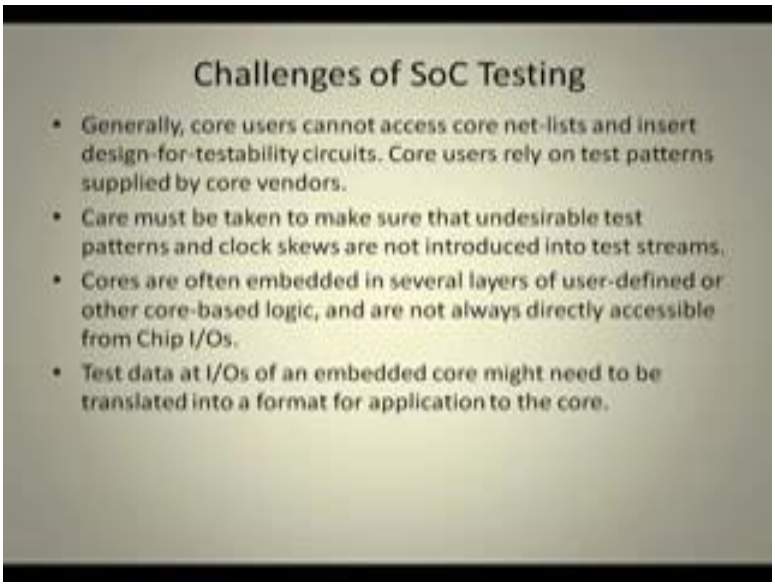
must test this interconnect and also this user defined logic, so all of them need to be tested and the SOC testing.

So, to avoid this cumbersome format translation for IP cores SOC and core development working groups such as virtual socket interface alliance have been formed to propose standards. So, what has what is happening is that as we have already discussed this SOC design, it is based on cores from IP vendors. So, IP vendors when they are describing their description, so they are doing it in some format as a system integrator, so designer they take those designs and try to put them into their design. Now they are naturally if it is coming from two different group of bodies then there is an interfacing standard problem, so there is a virtual socket interface alliance that has been created which is trying to address this issue and they are trying to come up with different standards for this purpose.

We have seen this IEEE 1500 that is that has been announced to facilitate this SOC testing. So, IEEE 1500 specifies interface standards which allows cores to fit quickly into the virtual sockets on SOC. So, if a core is supporting IEEE 1500 then it can be incorporated very easily into an SOC design paradigm and that testing for testing purpose the this serial interface and parallel interface available for 1500 they can be exploited by the test engineer to design the proper test mechanism.

So, core vendors, they provide cores with an uniform set of interface features and system SOC integration is simplified by plugging cores into the standardized sockets. So, we have the designer might have designed these sockets previously because what are the types of how to send different comments what are the interfacing signals and all that, so they are known once the standard has been fixed. So, based on that they can tell like what are the requirements and then this the core vendors simultaneously they come when they do their design in 1500 compatible format, so they also know what are the comments that we need to support and what are the interfacing signals that should be there. So, that way the whole process becomes a bit simple, so this is most of the time it can be a plug and play type of operation.

(Refer Slide Time: 04:44)



So, core users generally cannot access core net-lists and insert design for testability circuits. So, this is the major problem, because for say if we if we want to enhance the controllability and observability of a design, so we have to work at the net-list level. So, we have to know the at least at the logic gate level, what are the components and then if we buy some fault simulation or something like that we may know that this part of the core is difficult to test, so we put an observable point there; or we could we make that part controllable by putting some external input there. So, these may be tried out, but you see what is happening is that in this IP core base design. So, the system integrator does not know the detail of the core, so that is available with the core vendor which is never told to the system integrator because that is the that is an intellectual right is protected there.

So, the core the users of the core that is the system integrator they do not have access to net-list and they cannot insert design for testability circuit into it. So, core users rely on test patterns supplied by the core vendors, so core vendors responsibility becomes two point; one is to provide the layout level description of the system and the second thing is to provide the test vectors or the test patterns that can be used for testing the operation of the core. Now, the problem that we have is that core vendor they do not know in which platform this is going to be put, as a result they try to make it as exhaustive as possible. And these system integrators also do not know the net-list, so they cannot try out some sort of patterns to be more important or less important, so like that.

So, care must be taken to make sure that undesirable test patterns and clock skews are not introduced into the test streams. So, this is very important as we have seen that that the x values are to be stopped unknown sources and the clock skew are to be stopped when we are doing this testing. So, particularly that will affect your this compact, so the compression mechanism, compaction mechanism and all that, so they are to be stopped. So, how whether this has been taken care of by the core vendor or not, because as a designer as an integrator one cannot take care of this because the details are not known, so as from the core vendors point of view, so this has to be done.

Cores are often embedded in several layers of user defined or other core based logic and they are not always directly accessible from chip input outputs. So, this is another problem with core based design because the system input outputs are not directly feeding each and every core in the system. Now, in a normal circuit, we are handling the situation by doing some sort of justification through the logic circuit or the net-list. But in case of core based designs since the cores are coming from different vendors and they are the details are not known, so it is not possible to do a justification sort of thing, so that we can take care of this test pattern flow to a particular core. So they are not always directly accessible.

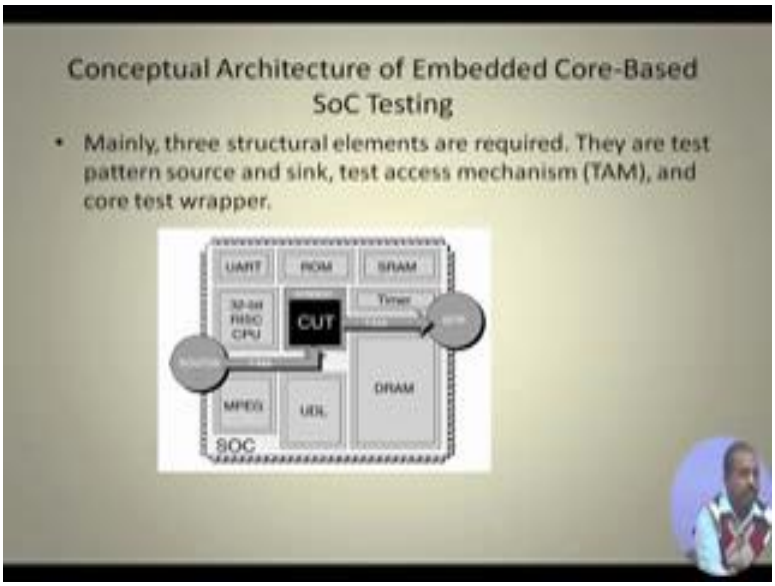
So, test data at I OS of an embedded core might need to be translated into a format for application to the core. So, this is another issue. So, may be sometimes we need to do some format conversions, may be the test pattern has been stored in the ATE in some format and then when it is applied to the circuit applied to the core, so it needs some conversion. It is particularly true, when we have got this test compression into picture. So, this test pattern that is stored in the in the ATE is in some encoded form, and on the encoded form the test data is transported somehow to the core and then at that core we need to do something, so that we get back the original test vector.

So, it is like this, if this is say that system chip that we have and this is the core that we are going to test. So, in the ATE memory, so we have stored the pattern for this particular core may be in this region we have got those patterns, but as we know that they are not stored often they are not stored directly as patterns, but they are stored in some coded form. And now this coded pattern somehow transferred to the input to the core, but this cannot be applied directly because they are, so I need a translation from this coded form to the back to the original form. So, I need some sort of circuitry which

is acting as a decoder which is acting as a decoder to get back the original pattern from these encoded bits that had been transferred.

Now, this decoder becomes a glue logic, so this decoder is part is designed by the system integrator and that is possible because integrator knows like what are the patterns and then what is the encoding policy followed in that particular design by the group, so that way they can design this decoder part. And this decoder part becomes a glue logic for the entire system. So, as a testing process is concerned, this decoder also has to be tested, so it is not that we can take this decoder as correct that may also develop some fault, so that glue logic testing also becomes an important part.

(Refer Slide Time: 10:50)



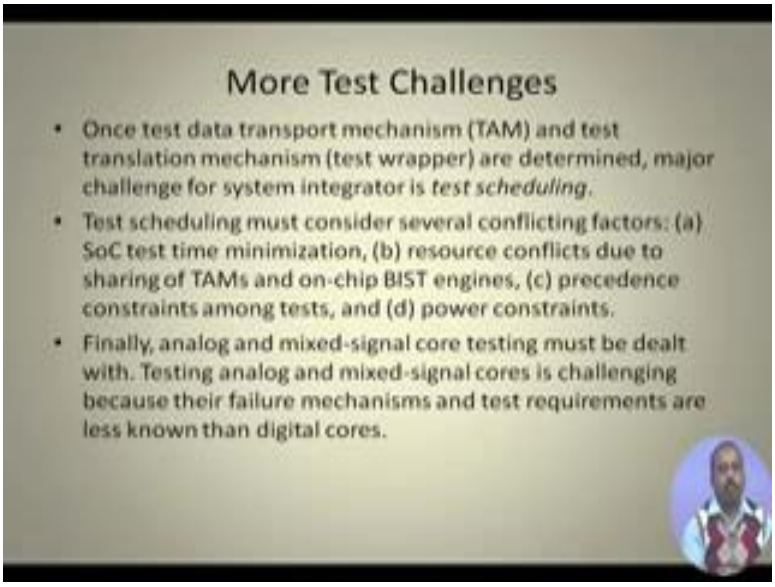
So, this typically shows what is happening in a testing of an embedded core based SOC system. So, we have got mainly three structural elements are required test pattern source and sink test access mechanism and core test wrapper. So, this is a typical SOC that has got a number of components or number of modules like UART, ROM, SRAM, CPU, timer, DRAM, UDL etcetera. Now, you see, suppose this is the circuit that we are going to test, so that is marked as core under test and CUT.

Now, as we know that every core is wrapped into a 1500 wrapper. So, this is the wrapper design that we have, so we will see how this wrapper design can be done, so this wrapper has been designed. And from the source the test patterns they had to be transported to

this core under test, and the responses are to be collected and sent to the sink, so that is the responses they are collected at the sink, so this is what is required.

Now, how this TAM can be designed, so that is an issue, so somebody may say I know the pattern, so I will make the patterns this module such that this CPU, so it will be transferring the patterns through it and it will be applying to this cut. But that that is a very I should say a very high expectation form the operation of this particular core, and without knowing the details of this, so it is very difficult. So, very often what is done is that we make some alternative arrangement may be in terms of some separate test lines through which these patterns are applied from source to this core under test, and from the core under test to sink. So, these are the three things that we need - the wrapper design, the TAM design and the test source and sink, so these are the three parts that are needed for testing individual cores in a SOC.

(Refer Slide Time: 12:59)



So, once test data transport mechanism and test translation mechanism are determined major challenge for system integrator is test scheduling. So, we have decided like what should be our TAM architecture, what should be the translation mechanism, so test wrapper. So, what happens is that may be my core has got a number of scan chains and this may be my core has got say 5 scan chains, and the number of pins that are dedicated for testing this particular core is 3. That means, this five scan chains, so they need to be converted into some form, so that ultimately this 3 inputs are feeding this 5 scan chains.

So, this type of modifications are not needed in the wrapper design face, so apart from that wrapper cells that we have seen previously while discussing on boundary scan part of the cores.

So, apart from ensuring how these parallel bits can be transferred etcetera some more things are to be done because individual cores may have different number of scan chains different number of inputs and outputs, and we may not be able to dedicate, so many pins for feeding all those inputs parallelly. So, we need to convert some of them to serial one, so that is basically a part of the test wrapper design problem. Apart from taking the other part other things like say this wrapper control part, this bypass register, the buffer register, boundary scan register and all that, so apart from those things so this scan chain management becomes an important issue.

And assuming that all these have been done, so we have decided on the test access mechanism, we have decided on this translation mechanism, the test wrapper design. So, main challenge becomes test scheduling, the system integrator has to perform scheduling of test for these different components, so that becomes an important issue. So, test scheduling must consider several conflicting factors like the test minimization SOC test time minimization.

So, as we have already said that since the SOC has got a SOC may have a large number of cores and each core vendor might have provided large number of test patterns, now all of them are to be applied, so that increases the test session length. So, we want to minimize this test time. So, for minimizing this test time one possibility is we want to parallelize number of test, but if we want to parallelize number of test for the number of cores to be tested then I need to feed all those cores with the test patterns. So, that in turn requires large number of pins to be there large number of ATE channels to be there to transfer the test patterns. So, that way this is one conflicting requirement that is the test time has to be minimized.

Whereas, the second important conflict that we have there are resource conflicts. So, if we want to minimize time so resource conflict may not allow us like sharing of TAMs. Like the same test access mechanism may be utilized for controlling number of cores, so they are all of them are getting input from the same time, so I cannot transport test data of two cores two or more cores simultaneously on the same time. So, it has to be

exclusive. Or it may so happen that some of the cores are BISTED, but they do not have the BIST engine with them. So, in my whole SOC, there may be a number of BIST engines who can provide this BIST patterns to the system. It is particularly true for this memories like if system that are designed today they have got large number of memories in them and this memories they have got this type of BISTED designs, so they use some BISTs for getting the test patterns.

Now, if there are a number of memory modules, so putting BISTs for all of them separately may be costly. So, only for a few BISTs are kept and then BISTs are allocated to different memories at different points of time. So, the test scheduler will find out which BIST to be allocated to which core which memory for testing at some point of time, and that way there this BIST engines are to be utilized. So, there may be conflict in BIST engines, so even if I want that a multiple number of memory tests should go parallely due to this restriction on the BIST number of BIST engines, so that is not possible.

Precedence constants among tests, so it may so happen that one module; it generates some inputs that are relevant for testing another core. So, one core generates some output which is relevant for testing another core, so in this type of situation until and unless the first one is tested the next one cannot be tested, so there is there may be some precedent constant among the test and finally, the power constants. So most of this systems that we are designing today VLSI designs, so they are going to be low powered; so low powered means there will be a power limit for the chip then the chip should not consume more than that power.

Now, the designers they have got many facilities because what they can do is that they can turn off some part of the chip, they can segment they can partition the chip into different portions in terms of the number of modules that need to be on simultaneously. So, that way the power budget can be decided. But for the test engineer what is happening is that now in an effort to do this test fast, we may like to turn on different modules simultaneously.

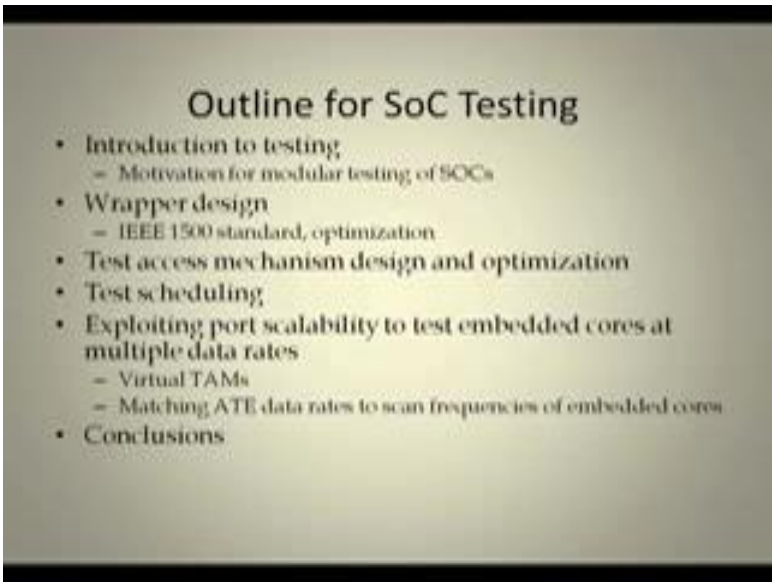
And as soon as we do that, it may so happen that two modules which are normally switched on in an exclusive fashion in a mutually exclusive fashion they get turned on simultaneously. So, if they are consumed lot of power then this power consumption

during testing will go up, so that is, so that these are conflicting like if I want to minimize this SOC testing time may be I will violate some power constraint. If I want to be within the power limit, so I just schedule one core at a time then of the test time requirement will be very large. If I want to test a number of memory cores simultaneously then this BIST engine it becomes a constant, so that and some certain tests I cannot do in an arbitrary order due to this precedence constraint. So, these are the problems that are there in the testing process of this SOC.

And analog and mixed signal core testing it must be dealt with. So, no system which interacts with the environment is purely digital, so there must be some analog part in it and some mixed signal part will be there. So, analog and mixed signal are there are to be taken care of and testing this analog and mixed signal cores is challenging because their failure mechanisms and test requirements are less known than digital ones. So, analog test is not yet very well developed.

So, many of the cases, so they this analog circuit they use similar type of procedure for testing digital circuits, but it is not yet well developed. Normally, some sort of these current testing current measurements are taken to check the correctness and all that, but that is not yet very standard one. So, this analog and mixed signal testing becomes an important issue, but when you are talking about SOC testing, so we cannot ignore this analog mixed signal testing.

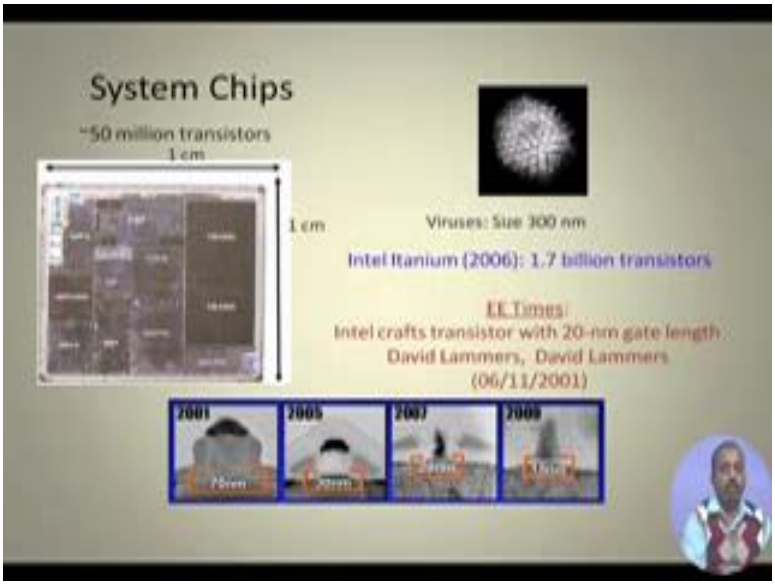
(Refer Slide Time: 21:09)



So, this SOC testing, so we will go like this. So, we will motivation for modular testing of SOC then we will look into this wrapper design IEEE 1500 standard and optimization test access mechanism design, test scheduling, so these are the various problems. So, this wrapper design is an important problem, then apart from this 1500 standard that we have discussed, so we need to do some sort of optimization. So, optimization part we will see for a particular core they may be different wrappers that we can come up with and that may decide this testing time of that core.

Then this test access mechanism design like how do we, so we have seen some test access mechanisms, but what are the alternatives. Then once we have decided on this wrapper and test access mechanism what sort of scheduling policy can we follow. Then there are some cores they have got some special feature like they can have some port scalability, so port can be made to operate at different frequencies, it can support multiple data rates like there is something called virtual TAM. Then the ATE can provide data at higher rate, so that way it can be taken care off, so there are there are many things to be taken care of in the SOC testing.

(Refer Slide Time: 22:27)

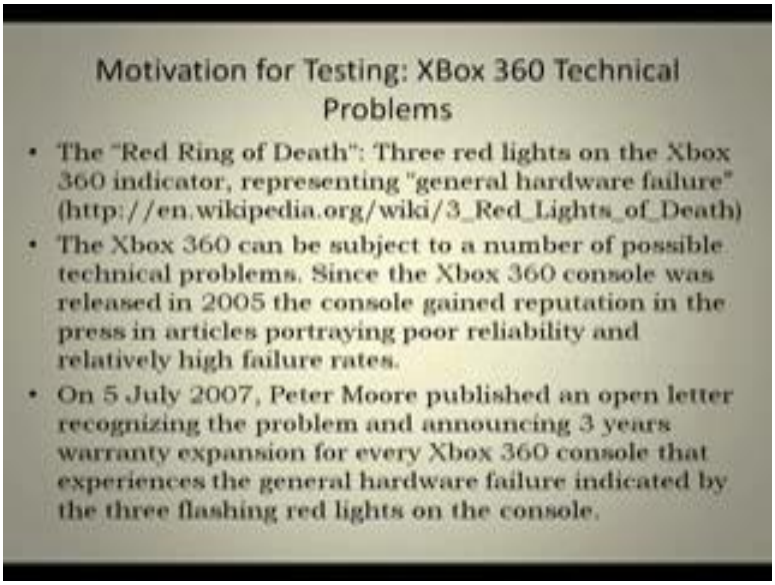


To start with, so these system chips, so they are having about 50 million transistors in a 1-centimeter by 1-centimeter area. So, just for a comparison the viruses the size is 300 nanometer, so even much smaller than this. And this Intel Itanium that came up in 2006 is 1.7 billion transistors, so its size is naturally it has got a very large number of devices,

so the possibility of failure is very large. And this actually shows that how this with the advancement in technology, how these transistors are going. So, this is 2001, where the gate length was about 70 nanometer; then 2005, it came down to 30 nanometer; then 2007, 20 nanometer; and 2009, 15 nanometer, so that way this gate length is reducing. So, that makes it more susceptible to many faults.

So, Intel has craft a transistor with 20 nanometer gate length, so that came up in 2001 actually. So, this way we have got different, so that shows the speed at which this design community they are progressing, so these devices are becoming smaller and smaller and that way we can put many devices into a single chip. But that way the possibility of failure or possibility of defect, so that increases, so possibility of faults also increases.

(Refer Slide Time: 24:10)

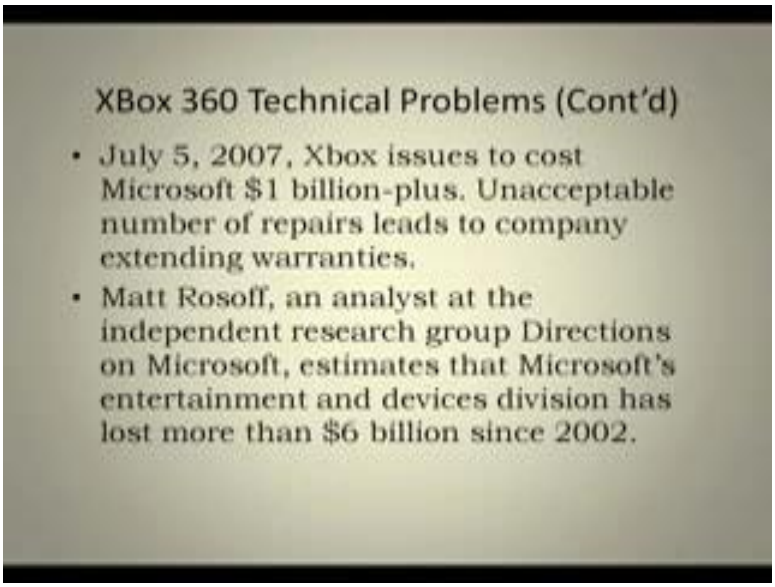


So, why is it so much important to do this type of detailed testing? So, there is an Xbox 360 technical problem, so it says the Red Ring of Death problems. So, there is on X box, there are three light indicator three red lights a X box 360 indicator representing general hardware failure, so this is available as an Wikipedia article. Then this can be subject to a number of possible technical problems, since this console was released 2005, the console gained reputation in the press in articles portraying poor reliability and relatively high failure rates. So, since that actually all cases could not be covered in the design of x box all the failures could not be covered. So, it was kept that if something hardware failure is detected, so it will glow three it will have a red ring three red lights will be there and

then that the system has to be reset, so that is an error. And it started occurring very frequently.

And ultimately, on 5th July 2007, Peter Moore published an open letter recognizing the problem and announcing 3 years warranty expansion for every X box console that experiences the general hardware failure indicated by the three flashing red lights on the console. So, this is actually a great loss because they are extending the warranty because of this hardware failure, so need to extend the warranty, so that becomes a great loss, so that indicates that why is it so important to do testing.

(Refer Slide Time: 25:51)



July 5, 2007, Xbox issues to cost Microsoft 1 billion dollar more than that. So, that is that is an unacceptable number of repair leads to company extending warranties the warranty has to be increased and then that cost, so much of loss. And that loss of an analyst at the independent research group directions on Microsoft estimates that Microsoft's entertainment and device division has lost more than 6 billion dollar since 2002 due to this x box, so that actually tells the system may become complex, but we cannot avoid this testing. So, testing has to be done and it has to be done very exhaustively.