Design Verification and Test of Digital VLSI Designs
Prof. Dr. Santosh Biswas
Prof. Dr. Jatindra Kumar Deka
Indian Institute of Technology, Guwahati

Module - 11
Built in Self Test (BIST)
Lecture - 3
Memory Testing - 1

Welcome to the 3rd lecture on module 11 which is on Memory Testing. So, till now I mean we have been discussing about testing ADPG for various side of sequences circuits, combinational circuits, offline. And then we have seen that, if you are going for offline testing that is it is using 80, they can be problems in in CQ testing. So, we have gone to a built in sectors mechanism of base and so forth.

So, the wide range of coverage we have built in now, comprises many of sequential and combinational circuits. So, sequential and combinational circuits if you think, there actually composed of gates, flip flops, wires and all those things, but there is another very important module of VLSI circuits, that is actually the memory. So, in case of memory testing, as we have to observe that there will be quite different, from the other digital testing of sync synchronous, asynchronous or combinational circuits, there is sequential and combinational circuits.

(Refer Slide Time: 01:12)

While because, the idea is that I mean, all the digital circuits sequential combinational whatever we discussed till now, we combination on logic gates and flip flop, flip flop themselves comprised of logic gates and all, but the memory blocks actually what (( )) differently, see the idea is that, what is the memory block like. So, if you have a block of RAM, DRAM, SRAM what ever. So, you have lots of memory cells over here and they comprise the value of 0 and 1 you can write it and you can read it back.

So, it is unlike I mean it is unlike the operation, I mean it is unlike, I mean with the idea is that you have to compress a large number of memory cells in a very small area, it is unlike you D flip flop where you can store your area. So, if you have to make a memory element, memory block of say 1 GB using only flip flop it will be, so large that you cannot make it, fabricate it or self profitable in the market.

So, if you have to make a 1 GB of ram kind of stuff. So, the technology to be used is very different because, you have to have a very, very compressed area and one more thing the memories is very symmetric because, it has memory cells, you can access the cells. And whatever you can do is that, read the cells, you can write that cells that is the only thing you want to do, there is no other logical operations involved in the memory that is the complex of the flip-flop. So, that that also stores that the bit, bit 0 or 1, but actually it also involving lot of what you can call logic operations.

So, the philosophy of using a flip flop and the philosophy of mem using a memory bit different, even if the job is the same. So, in the case of flip flops there are not basically symmetric. So, I mean in case of a memory. So, in memory or idea is that we have a symmetric structure and we have to compress as many as bit as possible and the operation is only read and write. So, we have to compress it and in case of flip flops the philosophy is different here, also you have to store a bit and retain a bit it also involves lot of, what do you call combinational logics formation logic logic operations.

So, we cannot have a symmetric. So, that is the bit different and, so in case of flip flop based memory or in types of flip flop itself. So, you can have spare to have somewhat larger area because, non symmetric and the purpose will be different, but in case of memory as is only read and write operations, very symmetric in nature. So, you have to compress as many bits as possible, using a very small area there is somewhat slight philosophical they are different.

So, that is why this necessitates different, fault models and test techniques for memory blocks because, of the very basically design philosophy and, also the memory block design. So, you might have heard about 10 years ago. So, when we had 1 GB of hard disk size to be a very good thing and RAM size of 500, 12 MB of 128 MB kind of thing, but now, what is the hard disk in TERA byte says and you RAM and ram and your ROM (( )) I mean what you call and your DRAM and the system what you are put in you PC they have gone to 4 GB or 8 GB.

So, that is why the memory technology is increasing very, very fast, but in spite of that in spite of your memory size increase, the size of the memory chips or the area of the component is increased. So, what you are doing is that, we are using very advance technology.

So, in the same dye area or a very minimum increase in dye area, we are trying to put as many as more bit cells possible that is living complex, that is why it is saying, it never take cap capacity for quadruples every 3 years, which leads to decrease in memory price per bit that is being stored. That is what saying, that you cannot your size of the RAM or size of RAM chips are decreased lot, you compared 10 years back and still then the amount of memory that you can take in or the capacity has been increased, that is what the same area, we are trying to put as many bits as possible in a very compressed way of fashion of the memory.

So, that the the cost per bit is decrease that is why, money is of price of the RAM chips are decreasing and the capacitor is increasing, all this happening because, of the extensive increase in fabrication technology. So, that is what we are seeing high storage capacity is obtained by, increase in density which implies decrease in the size of the capacitor which is used to store a bit.

So, the basic idea is that, there is some storage charge to a device which actually stores the memory. So, we are not going to a memory, this is not memory design class. So, you can look at many analogy design or memory design book, so there are different memory structures like EPROM, PROM and RAM, DRAM, SDRAM lot of different architectures are there, but the basic idea is that, there is some kind of capacitive element or some are the element you can take, photo, port you can say for the modern very modern memory.

They some kind of capacitive driver charge to store the device, where you stored the charge if the value is to be stored 1 and otherwise will store a charge of 0 need the value to be stored the logic 0. So, now what is happening. So, now, a days very, very new technology I mean, new material or fabrication technologies are die dielectric materials like barium and all those things all these, these are new I mean what do you call fabrication technology, using different dielectric materials coming up to build up the capacitors, which is charge through devices, which will have to have a greater capacitors and maintain the same space.

That is what is the idea is there, with this new lot of new technologies like this types of dielectric materials and this is actually physical layout stuff, which you can find out from many standard, physical device, physical design books or physical device lectures or physical device stuffs in vein, but we are not going to cover in this lecture on this on this testing up because, this covering mainly on the memory design, but the very basic issue I want to emphasize is that, by using lot of advance fabrication technologies, using newer and newer types of dielectric materials used for doping and making a IC's, we are able to compress much more stuffs or much more memory capacity into small area.

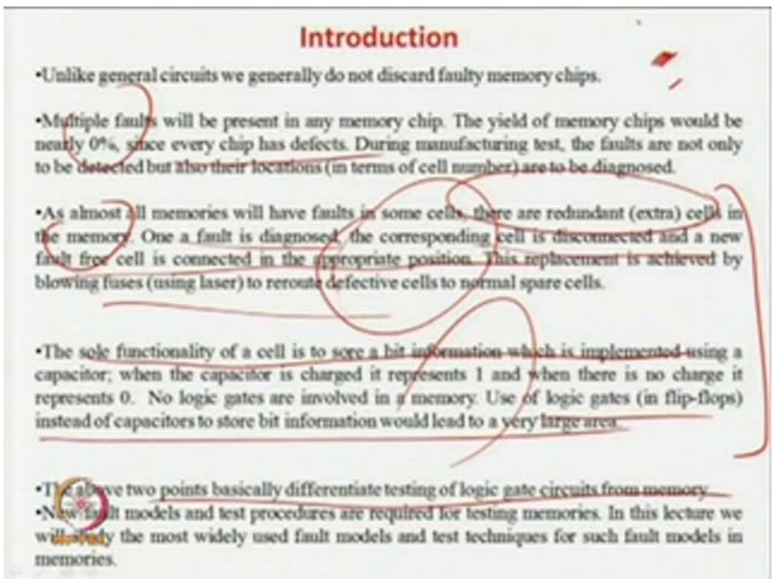
That is by increasing the capacitance at the, by keeping the same amount of capacitance you can assume and in a very less amount of area that is les greater capacitance, we maintained in this same physical space that is in the idea is that, in other words we are having more capacity increasing in the less amount of space. Further memories are now techniques are also be developed. So, that you can access the memory very fast, with fast stage mode access is there extended data output synchronous and also the double data rate.

Also new technologies are coming up in memory design, like fast page mode means you can very quickly access the memory and paging's are faster all those things, in details you can find out many memory design book and also I mean note you call double data rate SDRAMS are all those things are advance in technologies, that is double data rate means in both the clock just you can access data are you can write the data from memory.

So, the idea is the memory technologies advancing far end beliefs and bounds, like a real going for also for the increase in your combinational and sequential circuit design or so

is slowly shrinking in nature, by using this some micron technology, similarly in the memory I mean period time also, we are having higher memory sizes in same area, higher fas faster memory access rate, like higher reliability, double data rate acceptor. So, these are all the future features increasing also in the memory area.

(Refer Slide Time: 07:34)



So, what happens is the, a lot of failure as in the cases of simple sequential circuit or combinational circuits, memory circuit chips are total failures, but there are two or three philosophical difference, which will make memory testing bit different from what we are going from this sequential and the combinational circuit testing, in sequential and combinational circuit testing what we are seen, is that if the circuit is having any defect that is found on testing you throw it off.

And you ship only the good chips which are I means which are detected fault free, but here, if you try to do this with the memory, then you are going to find that around only 1 or 2 percent of the chips are even only lower that will be the eat that you are going to test you memory chips and find out the royal through out all the chips as defects, then you will handle the situation that, your defect that I mean your yield will be 1 percent or less than that, then what is the idea, in case of memory as it is very symmetric. So, we are using such a technology.

So, that we are pack trying to pack as many stuffs in a very small area the area as possible. So, if you make use fabrication in a stuff manner on a stack way manner will

have lot of possibility of fault symmetries. So, what is the solution. So, unlike general circuits, we do not discard any faulty memory chips. So, why is that, to start we need in the end we are going to have one, we will find out that 1 or 2 percent of the chips you can sale in the market, so you cannot tolerate this.

So, one thing is that, you can make actually make memory layout in past manner. So, that if probability of faults are there, less or then what will happen is said the chip area will be more and then it will be normal profitable. So, what would you doing is that, I means, we say in seeing I mean the yield of memory chips will be near about 0 percent, see the average chips almost have a defect.

So, I mean here you have put in different philosophy, so what is the philosophy is that, now that is the found out that as we discussed right now that all memory chips more or less 100 percent chips will have 1 or 2 defects. So, what is the idea is the you cannot throw them out. So, what is the idea, you have to repair them and you have to ship it them. So, what the people do is that, they take the memory IC and while fabricating say 1 GB of memory they will fabricate something extra, will not be exactly 1 GB will be something more than that.

Now, what do you do is that first we have to detect the faults, in in case of combinational sequential circuits we detect the faults finally, this faulty with the 4 digit from the been for chips already been goods and heads with a different philosophy. So, what we do, as you know that all the chips are fault. So, as we fabricate somewhat extra part in the memory, 1 GB want to say fabricate 1.2 or something like that. Now, what happens you find out that there fault detection and you also diagnose these faults.

So, so in all memory we find some cells there are actually redundancy and some extra cells there in the memory because, you have to repair and sell. So, if you saying that I am shipping a 1 GB memory, you cannot say say you are giving 500 MB kind of memory that is not possible. So, you have to I means somehow fabricate more area, then what you are claiming and then you have to discard to repair your chip that somehow you have to repair, when repairing is not possible, somehow you have to do something with the fault part with your IC and then you have to replace with them the redundant cells.

And, so that totally 1 GB your shipping with something like, say you you have to ship some 10 mangoes and 1 or 2 mango may be below 10. So, what do is that you you you

get some 12 or 13 mangoes and then you find out the this 1 or 2 or 3 mangoes are not proper. So, you de market them because, you cannot once the chips are fabricate you cannot means cut out the part and threw it off, use the de market the mangoes and with the I means not the good mangoes and still stay maintain 11 mangoes are good and you can ship them to the customers, same philosophy use are used about here.

So, these are cheaper 1 GB you can fabricate something more, you detect the fault as well as diagnose it. So, in case of combinational and sequential circuits this thing you did not diagnose this fault, but here also we diagnose, which area or which cells of this circuit diagram fault. So, once you find out that there is fault. So, you have to mark those cells and you have to do some there is always multiplexing arrangements is there. So, these some memory repairing, you are not going in details, but the idea to understand, if there is some what, some kind of repairing available that is how do you repair you cannot repair.

So, there are some 10 cells which are damaged. So, there is multiplexing arrangement in which case, you can connect the inputs of which are going into the damaging cells to the extra redundant boot cells you have been there. So, you connect them and, similarly to the output of the failure cells, should be again couple de couple drafts this is cut off cut off using a multiplexing arrangement and, so that the output of the boot cells which are extra redundant in market I mean redundant in your chips can be brought into the normal memory area.

So, while using of the multiplexing, so once in faults diagnose, the corresponding cells is disconnected and new fault free cell is connected appropriate position. So, how we do sometimes do it by multiplexing arrangement and sometimes, we achieve it by blowing up some laser fuses that is there is some u u we you can have some multiplexing arrangements, so you find out that these circuits are not working properly.

So, how you will de market it and as well as connection a a input and output the cells can be blown off, by using these laser cutters or some something kind de multiplexing or multiplexing chips and you can I mean permanently select or deselect the appro for example, this is the input, which is input for input to the faulty faulty faulty what you call faulty memory cell, this is for the good cell. So, what we have to do, this there should be

multiplexer kind of a thing, this is your marks. So, this is your input. So, it is corresponds to 0. So, this corresponds to 0, this corresponds to 1.

So, you should always find out that, the input should never go here because, it is faulty the inputs already go here, so you that if I put 0 in the multiplexer sorry if you put 0 in the marks, select line of the marks. So, data will go to the normal part and it is normal end of now and if we put one in this select line of the multiplexer, data will come to the faulty part. So, you permanently make this multiplexing input as 0. So, that data will always will come to the normal extra memory cell.

So, this can be done by I mean using this permanently fixing the multiplexing selective select lines. So, that your data is going to the normal part of this redundant cells is fabricator, it will go there and the other parts it will not be used or sometimes you can use. So, you can laser technology to cut out, like for example, this is the pattern, this coming to here and both it is coming to here and here. So, demark that this is faulty. So, you blow that faulty using a laser. So, that your data will come to the normal circuits.

Similarly, you can do it for the output but, these are actually I am telling in a very, very crude manner because, there are all memory technologies involved, like memory if they failed detection, diagnosis and repair all those things. So, that is also the scope of the present course, but I am just giving a very brief or a broad, I mean famine kind of an idea. So, if you are interested you can read through any, I mean memory design books some of the reference also you can find out in the course.

So, now, that is the two philosophical difference. So, first philosophical difference is that you have to diagnose the fault, as well as detect and diagnose which is not in the case of sequential and combinational circuits. Secondly, you have to also repair there is redundant part of the circuit. So, you have to also repair this and second thing is that, these are two one philosophical stuff, another philosophical stuff is, the soul functionality cell is to store information and charge and discharge that is to store something and retrieve it back.

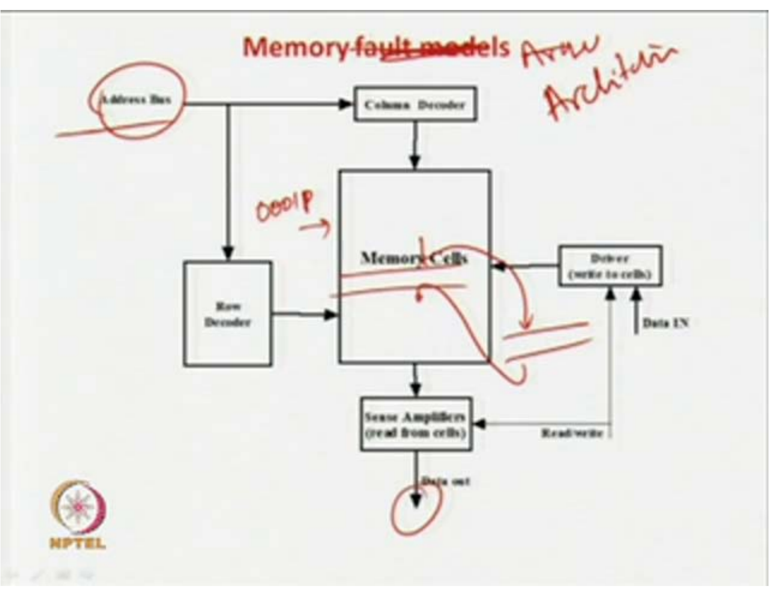
So, use of logic gate I mean. So, you I mean. So, the idea is here idea is that. So, an logic gates are used in for many other kind of logical information. So, this whole structure of a memories, will be different from the structure of logic gates. So, these two will actually lead to the above two points will differentiate testing logic gates from, memory testing.

So, this will actually lead to a different kind of form model and also I mean this types of things, says that you have to go for a repair and you should have an redundant part. So, the repair is possible.

So, because, of this repairing facilities require required because, to solve all the chips are faults. So, you have to repair it otherwise your yield with the less than 1 percent or you mean less than that and your memory architecture is very, very symmetric and, so many things can be stacked up in one place and functionality is nothing, but storing 0 or 1 and retrieving them back. So, these two points make the philosophy of what do you call, the circuit I mean the circuit testing different from memory testing.

So, in this course I mean in this lectures, 2 lectures will see what are the different memory fault models and how we can detect them. So, that is very basic idea we are going to see.

(Refer Slide Time: 15:10)



So, these are memories architectures sorry, this is the actually memory architecture this is memory architecture or memory model, this is basic memory architecture what do you have. So, this is standard memory in your book you can find out. So, memory will have cells. So, these are row decoders. So, give a address. So, address will tell you that say you give the address as 00010. So, it will be the second memory location will be accessed and you have to find out the column decoder.

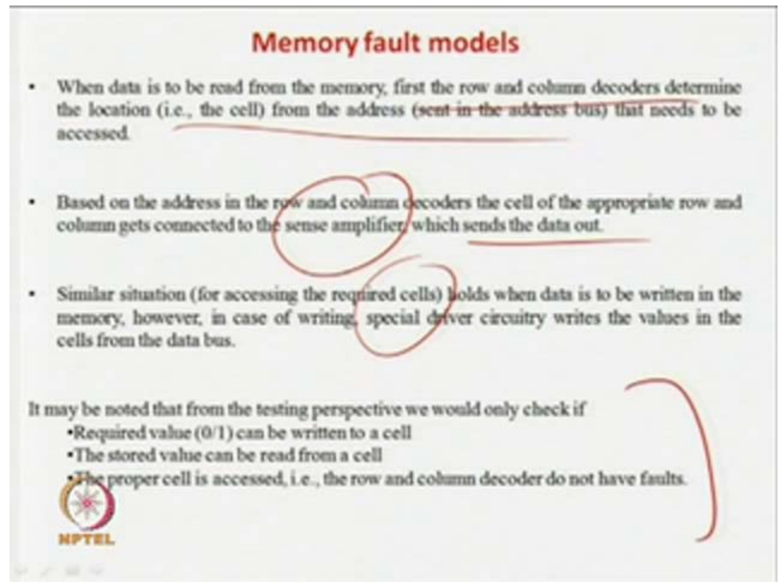
So, this is memory will also bearing some columns. So, which which part of the data will you access then really it can be, if you want to access the whole word. So, this is the whole word (( )) access a part of it. So, the column it will tell which part of the column you have to decode and there are cells amplifiers which will analog circuits which we take this data out from the different part.

So, this similarly, but if you want to write the memory location, then also you have to tell the address bars which is the which area which memory location say for example, 3. So, this is the memory location which I want to write and then the column decoder will tell which part we have to write. So, this part will be selected and your this some driver in the circuit, which will tell what values you have to write, these are very basic memory architecture.

So, the components are this is the memory cell, which is the stack part is quite different from what do you call our normal sequential and combinational circuits and I have already told you, there is some redundancy of air. So, if you are I mean some part of cells get damaged. So, what happens is that, it will actually re orient the connection with the glowing fuses and all whatever, we discussed. So, that this this is the bypass, so what I am saying means bypass means. So, that this part of the memories cell has gone back. So, this input will become over here, this is the redundant sorry.

So, this is the is some. So, this part of this was I mean some kind of a problem. So, this is are some redundant cell. So, this is the rednudant. So, this will not join here, it will come back here and from here it will come back. So, this is bypassed and this mu extra redundant cell comes into picture, these are this repairing them after detecting some faults. So, these are very, very broad outline of memory architecture volts, what is the memory architecture is.

(Refer Slide Time: 17:04)

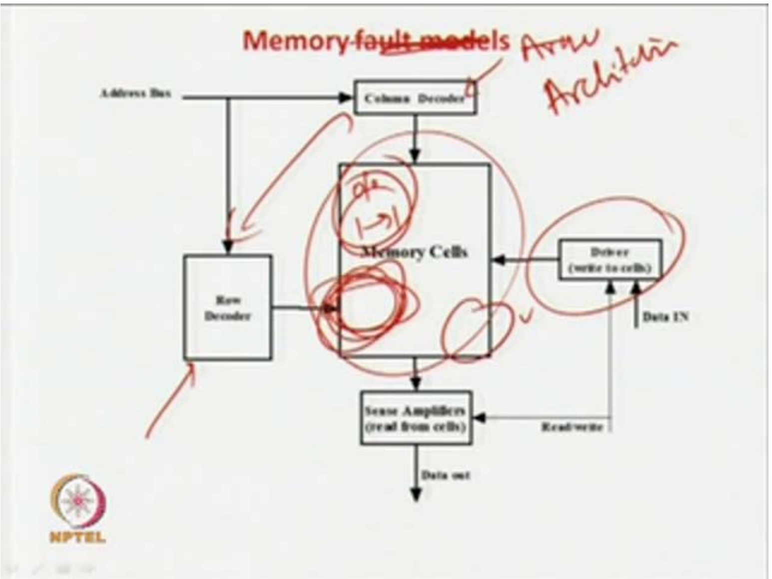


So, I mean they already told you, wherever you read data from the memory. So, from the row and column has to be selected accordingly to the data bus, based on the address and the row and column decoder, which is the appropriate part of the memory of the cell that is get selected and get connected to the same cell repair which same cell of the data. So, when we have to write it. So, you have to do it same thing, you select the row and column and then, we deriver will write the value to this one.

So, what do you require to trace the memory, we do not know other logical function accepting it is just actually read some data, write some data that is it. So, what you do, (( )) take any kind of ending, oaring all those things are not required. So, what is the two steps you want to do. So, repair a value of 01 to be that in a that it should be very fine, this is self should be able to be 0 and 1. So, Secondly, the value should be also be able to rate from a cells either 0, it should be able to write 0, if you are able to 1, it should able to write 1 and this should be possible in all the memory cells.

Secondly, proper excess is there that is, if you want to access this part of memory. So, by giving the proper values in the row and column decoder So, these are decoder are basically combinational circuit decoder.

(Refer Slide Time: 18:08)

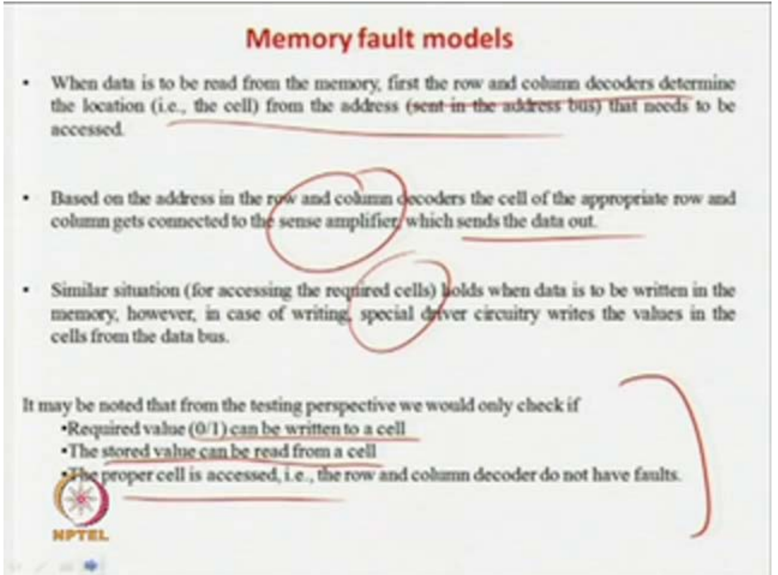


So, there are also basically normal circuits, I means normal logical circuits. So, they just decoders. So, only this part is quite different from what do you call general combinational and sequential circuits. So, the third fault point is, it should be if your are working to a address I means to access this part of the memory. So, it should be able to do it, by giving proper values to the row and column decoder.

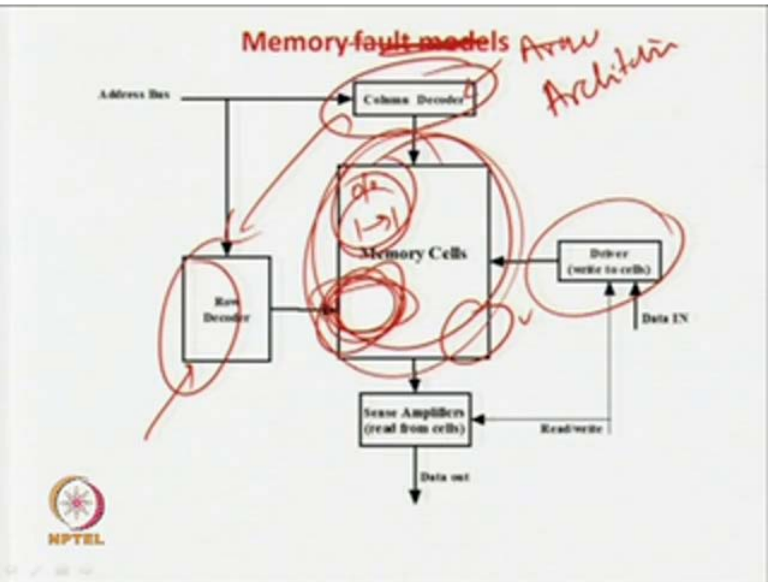
So, you should be able to select this that is the fault decoder for this row and column decoder. So, you should able to see that if I want to write 0 and I should be able to get a 0, if I want if I want to 1 I should able to get 1 from the same memory cell and as well as if I want to access the proper memory cell to the address. So, I should able to access the proper part, if I want to access this and by this some partial some failure in the row and column decoder I am access this for the (( )) that is another type of fault.

So, you should be I mean able to check. So, only these three types of faults you should be able to test in the memory block. So, that’s it. So, based on this, we should be developing the fault model for the memory So, I mean there as I told you, there are three three different types of circuits in the memory.

(Refer Slide Time: 18:49)

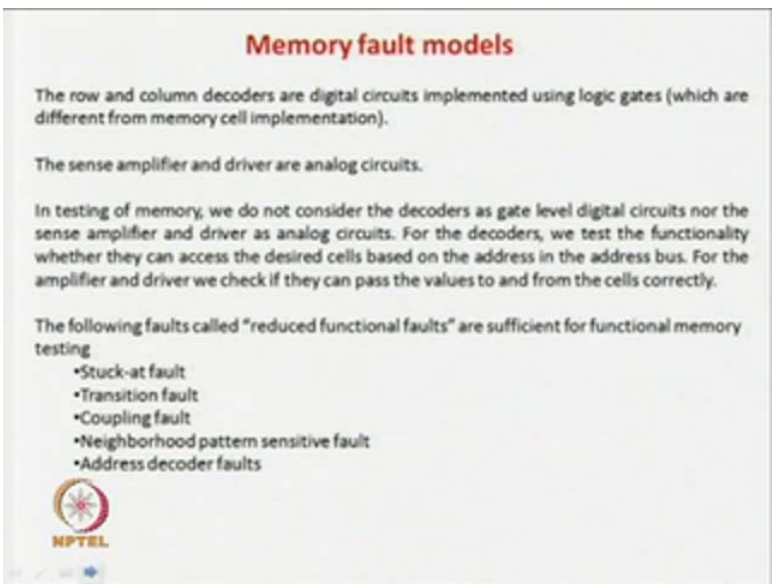


(Refer Slide Time: 19:04)



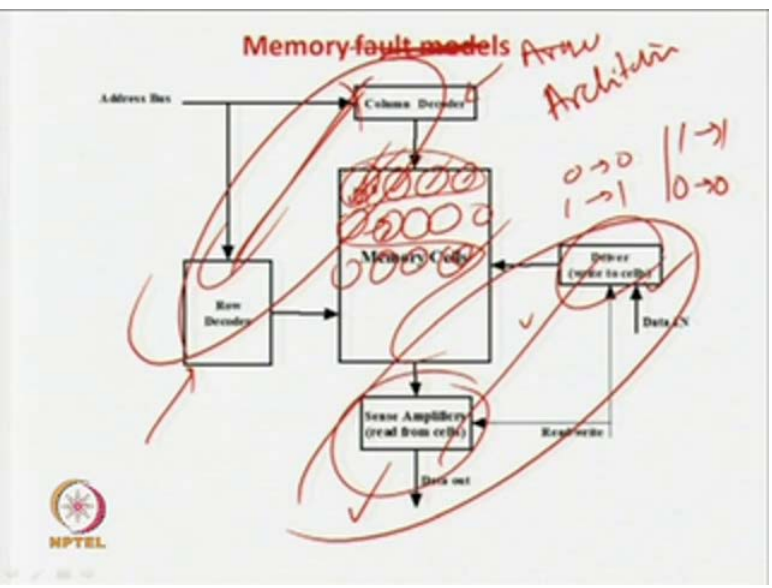
So, this one is the different fault from this which is mainly fault main fault to be tested because, which you are actually this is the compressed capacity kind of memory cells or whatever, either the different from the combination sequential circuits, these are your standard digital decoders, so general circuits and same simplifiers and divers they are and all circuits, there are not digital circuits.

(Refer Slide Time: 19:25)



So, with the three different type of circuits are there and which should be able to tested. So, now, what is the idea.

(Refer Slide Time: 19:31)



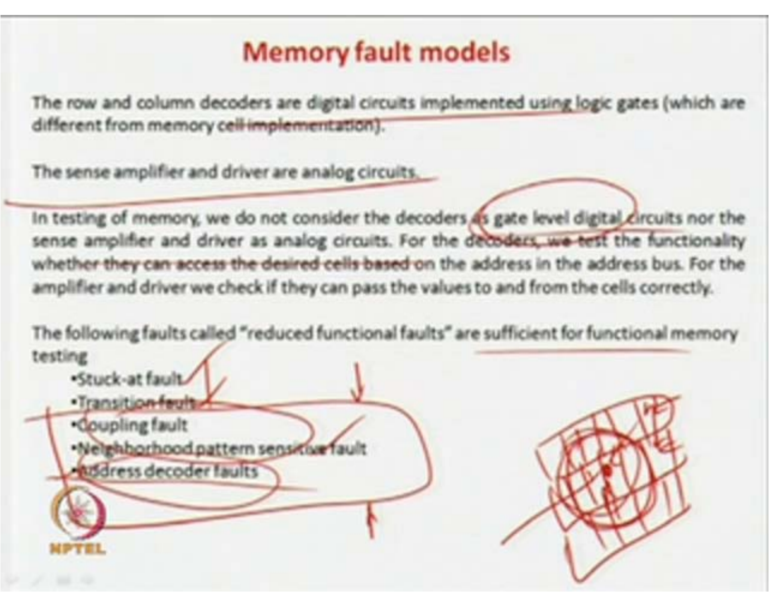
So, idea you see, that you do not general go explicitly for testing out, these these combination circuits and this digital circuits, well what we do is that, we travels each memory cell one by one and we will try to write 0 get 0 will write 1 data 1, again we can also put the reverse, write 1, get 1 write 0 get 0. So, if you would try to do it for all individual cells one by one in sequential manner. So, all the points will be tested.

If you are going in a sequential manner first this then, this then, this then, this that is the idea is that row and the column decoder should be working in a proper fashion because, otherwise, when you make it jumbled up a time, you are access this your are accessing this point so you make data, long data. So, if your using this all this point in a very sequential manner. So, row and column in a what is different in in in another sense be tested or verified.

Similarly, sense amplifiers means you write some data you are accessing some data. So, if your are getting writing some proper data, getting some proper data; that means, sense amplifier is correct, if your getting it proper data and if your able to write properly then also testing will be driver. So, explicitly you need not test all this faults. So, if you sequential write all faults of memory cells or in some predefined fashion, you know RAM writing in this location and to get back to this location.

So, so predefined order which will know, write out the memory and you read them back and when indirectly everything is tested. So, that is what is actually you are going to do.

(Refer Slide Time: 20:46)



So, what are the different fault models. So, the same sampler as I told along circuits and decoders are general digital circuits, but we do not consider digital and the gate level and sense amplifier and logic circuits, do not consider and test the because, analog testing is totally different, we are not going to test them in a explicit manner, even the decoders we

are not going to state using a stuck and fault model or what do you can call bagging fault model you know.

We are actually either I told you, will write the sequential memory blocks or memory cells as manner and retrieving them back and there is a want to see, weather I mean everything is proper or not, as I having that is the case, then we know that all the blocks are indirectly tested. So, what are the different fault models, which is sufficient for memory blocks people have followed, stuck at fault that is each of the that is non of the memory fault stuck to 0 stuck at 1.

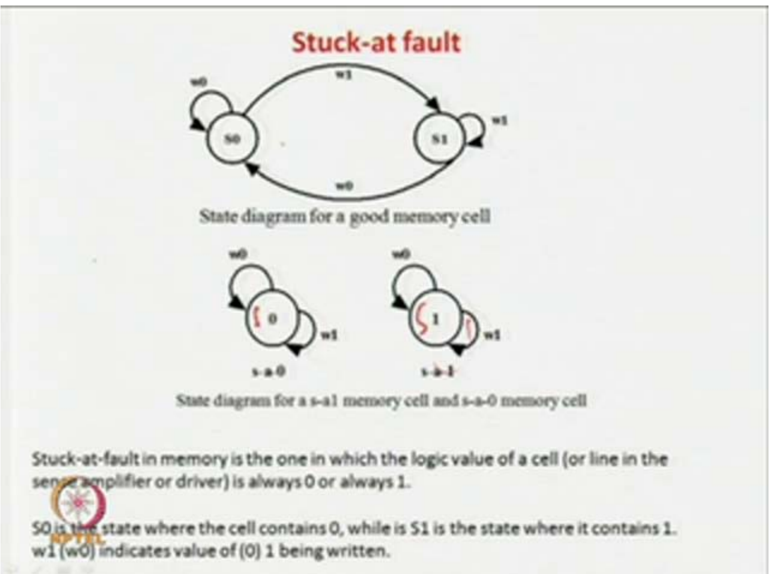
So, it can be we can write 0 or 1, second is transition fault, transition fault means you should have you should have if 1 is 0 is have a written, if you try change it from 0 to 1. So, that transition after cells be possible, coupling faults one very importance that to standard form, like I mean you should able to go from 0 to 1 and 1 to 0. So, these are other very important, these are some these are also available in sequential and combinational circuits. So, although we have discussed only about stuck at models.

But, transition faults, delay faults using faults are also consider in case of a sequential and combinational circuits, but these are some of the are the very important test we generally exist in the memory this is called coupling form because, the memory cells are like this and they are, very much stack with the nearly each other, this is stack very much in the cells of a memory near each other.

So, one may upon is this two cells may get coupled. So, any changes here, may affect here, similarly if you see like this. So, if you have three memory cells this is the memory cells, surrounded by some other cells and they very near each other. So, whatever happening in the peripheral of this may affect this cell. So, this things are called coupling fault, neighbourhood pattern sensation fault a address decoder under accept of faults in will see that it can be automatically covered if you are covering this, so that we will see.

So, this coupling fault and never defaults cells are happening because, the memory is very much stacked and all the cells are very near to each other in a very small area. So, anything happening periphery, may try to affect the central part of the, one of the one cell of the other.

(Refer Slide Time: 23:01)



So, this coupling fault and neighbourhood pattern density faults are mainly are new kinds of faults which is tested memory, they result behind his is is very in interesting because, everything is stacked are near to each other. So, I mean normal of this cell can, remain very much free are, it can move free under default case form compare to it is neighbours. So, that is why this neighbourhood part, neighbourhood pattern, sensitive faults and coupling faults are every it in a very important role in the memory.

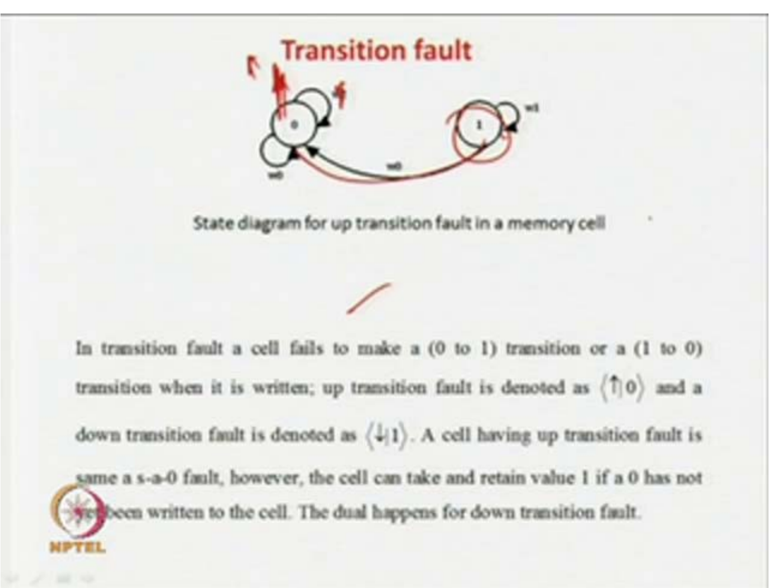
So, slowly see elaborate one by one each of them and we will also see why they are more element in a memory. So, there is first the stuck at faults. So, I mean how will this stuck at fault, I mean in a normal case. So, state 0 replaces that some memories state is having 0 and 1 means that memory cell has retains 1. So, if you write as 0, initially if it is 0 that memory cell if you write 0 w 0. So, you come back to this one, if you want to write 1. So, you write w equals to 1 you go to state s 1.

Similarly, if you from if you state value is 1 if you write a 1. So, you will be in state s 1 and if you write as 0 from you l go to state will be normal state diagram from node cell, but now you will start at 0. So, what is going to happens, if you are only in this state that it is 0 and if you write as 0. So, no problem differently try to write one in the cell, you will be in that same state. So, that is actually from stuck at 0 form here, if I start at from what is going to happen if the only in this state.

So, if you write as 0 you cannot go to, go back to this state only you only in that is on state only and if you write a 1 that is and this is the case. So, I means this is the state diagram for a normal memory cell and this state diagram for stuck at 0 and stuck at fault. So, that is what is been elaborate here. So, what happens? So, stuck at is moving logic value is always 0 or 1 this stuck at fault are sequential or combinational circuit in this case also, stuck at fault in a memory is 1 in with the logic value of a form of 0 and 1.

So, it state 0 it 0 and state 1 plus 1, but if you cannot move from this to this with the stuck at 1 or stuck at 0 for.

(Refer Slide Time: 24:51)



Now, from (( )) transition fault, transition fault is what, transition fault means transition in a cell, if transition fault in a cell phase to make a 0 to 1 or 1 to 0, transition that is if you want to move this will that is 0 to 1 if you want to move in a cell. So, it will be stuck to 0 it will not allow to denote called a rising term with default or you can call up up transition something like that and also may be happening that it want to go from a cell from 1 to 0, but that is not possible because, defaults to remains as 1, so you can call as a down transition fault.

So, these are up and down transition faults are also very much prominent in case of memory cells, where these also exits in a see combinational circuits, sequential circuits I mean if you are not discussed that, but I mean in case that is actually called delayed raise in delayed faults are there, so that you are want to go from 0 to 1 with subtraction of a

same individual, some nano signals particularly it happens, it waits for a long time because, of the delay faults. So, this is the also map to transition faults, in a sequential and combinational circuit.

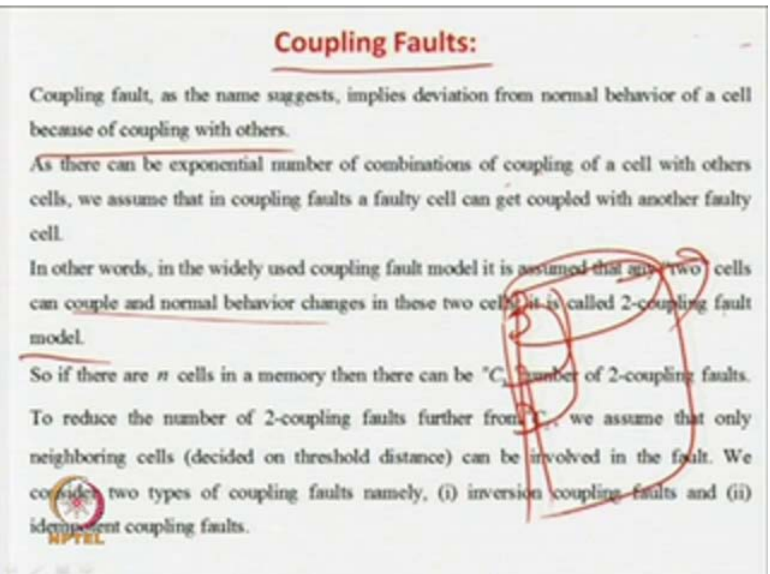
But, if you now look in memory perspective, we just look at this state diagram for up transition for this is not allowed. So, you see, but this is not a stuck at 0 faults. So, these are difference with this one in a stuck at 0 fault and in this case you see it is a 0. So, if you want to write 0. So, of this will be there, if you want to write a sorry if you want to write 1 over a w 1. So, from 0 to 1 you want to go that is you cannot, go that you will remain here, but these not a stuck at fault.

So, there will be a state of 1. So, if some how some how some thing not 0, some how will be 1, in some case mainly fault has occurs. So, 1 to 0 you can come back, from (( )) s 1 that is 1 in this you can come back to this 1, but after we cannot raise the value and if you want to write 1 that you will be in same cell, but if you want to stuck at 0 faults. So, this part would be not have been there in cells. So, these are the difference between in a stuck at 0 fault and in a transition faults.

Transition faults is you cannot go from 0 to 1, but if differently occurrence of the faults it would be have in that this the together one over sum of cells using this start up a main memory, the cells are all filled up with garbage value. So, it would be possible that particular cell has value 1. So, you can confirm on to 0, but you want to go for high. So, you cannot do that sorry this is a 1. So, it will be remaining that. So, this is a up transition first similarly, you can also a down transition for this way.

Now, very important is the coupling faults because, as there are already told that, memory cells are very near to each other. So, I means they too people very near each other, then behaviour of on cellular will affect the other and if one person is he he he is very near to about 10 about the other people. So, you already know that the character of the person who is around 10 other people have been influenced by the other to other people nearing to as.

(Refer Slide Time: 27:03)



Now, same case in memory is very must stack. So, I means it is a cells. So, you have something over. So, all the people around here. So, they are fact of all are these cells will be affecting the this central cell in off where this one. So, that is why coupling fault is there are two cells, one is the coupled with each other and there affecting each other. So, that that kind of fault is called coupling fault.

So, this is a coupling fault as name suggest implies, deviation from normal behaviour of a cell because, of coupling with others because, of getting I mean called getting influenced other cells, so it happen. So, I mean you know that a cell can be, say this is fault cell a fault cell can be coupled with (( )) memory for n plus 1 cells in the memory let assume, then for this particular fault case or the faulty cell. So, it may get coupled with one cell.

So, one means this ends see one any one of them, then it cannot coupled with two cells. So, it will be how much n c 2 dot, dot, dot you can be called in my n by n c n. So, it can get upgraded with any one of the end faults it can be coupled with two cells dot, dot it can go to coupling with all end of other cells. So, the number of the coupling faults can be extremely high its n c 1 plus n c 2 dot dot n c h.

So, it is a very high in number. So, that number of different all possible faults case will be extremely exponentially in number, you cannot go for that, so standard listing taking that is called to coupling fault model it assumes that one fault can be or one cell can be

coupled with at most any other cell and you can try out with the all are the cells like. So, if there is end cells. So, if you can get coupled with any one of them. So, there will be n different types of faults only compared to n c 1 plus n c 2 dot, dot n c n.

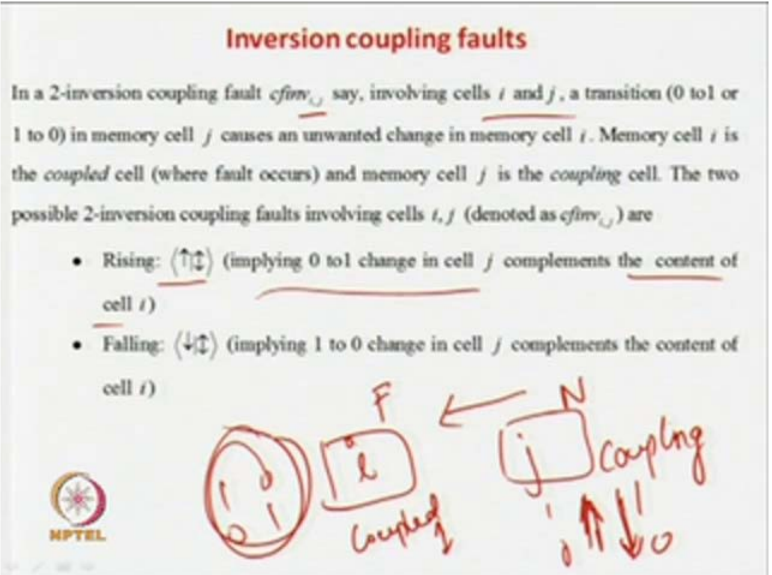
And well other stuck at fault model, we will have find out that (( )) boot coverage as well as boot confidence on you I mean in fault testing. So, this called the coupling, any two cells get coupled as well this one. So, if there end cells in the memory because, there can be c end to number of coupling of faults; obviously, and to reduce that. So, you generally you can, go you can understand that. So, there is a big memory like here. So, this cell cannot be coupled with this cell because, they are. So, far from each other.

So, you can make a window at all this stuffs or you can make a shifting window kind of a thing. So, this window will be shifted and this window shifted, this window shifted. So, shifting window concept, like this one. So, this is one window. So, then it will be again shifted, this window will be again shifted, this window will be again shifted here this remain. So, we all know this shifting window concept.

So; that means, we can consider that two cells which lie within ours certain square area like for example, in this case only cells within this area can get compute and this is a cell here, cannot get above coupling effective because, this cells is higher, but this cell will be ever getting coupling effective. So, this is a moving window kind of a thing. So, you move the window here, you can move also this one only this direction or in the in the this direction.

So, by this one you can, reduce the number of all possible couplings from ends n c 2 to or this one. So, there are different types of (( )). So, the number is fixed. So, it admits n c 2 if considering 2 gate coupling faults and secondly, also we can we can reduce the another down from n c 2 that by looking at the standard of the faults that these two cells are very near each other. So, moving window concepts you can do that.

(Refer Slide Time: 30:28)



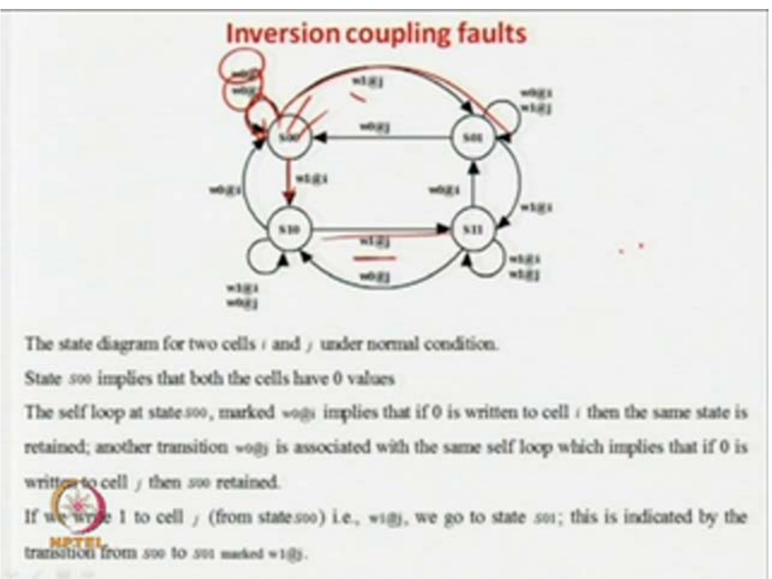
So, now here will be studying two types of coupling, this one inverse coupling faults and independent coupling faults, so that there are two different types we will see and that will make the things more clear. So, first we will see the inverse coupling faults. So, here actually you should see that, these are cell called i and these are cell called j and these two things are getting coupled are with each other or in the problem. So, these cell actually this is cell the coupling cell and this is the coupled cell.

Now, you see the coupled cell actually the faulty cell. So, this diagram after this one this is normal. So, this coupling cell acts the coupled cell and this guy is getting fault, where is the basic idea. So, in a two inv inversion coupling fault will take i and j then two cells i and j are, this that transition form of 0 to 1 or transition from 0 to a in cell j that you go from here or you go from here, that is the cell j with develop of unwanted change something in cell i that is actually the inverse coupling fault.

So, in inverse coupling fault what is happening, so there is a cell called n and there is a cell called i. So, this is the coupling cell, as well as the coupled cell. So, if this 0 to 1 and 1 to 0 of changing happening in the coupled cell j and it is affects the coupled cell, in some undesired way. So, this is faulty cell and this is the fault causing cell. So, what are the type what are the types of inverse coupling the, so 1 is the raising and flay, that mean change from 0 to 1 in cell j that is the 0 to 1 in cell j compliment the content of cell i.

So, this gets compliments. So, if it is a 0 now, it will be 1 and if it is 1 it will be 0. So, any 0 to 1 raising here, will comp complement this cell, will complement this content here, similarly down picture also falling 1 to 0 sorry 1 to 0 here and again call this undesired actually. So, this called the falling and this called the rais raising inverse coupling fault.

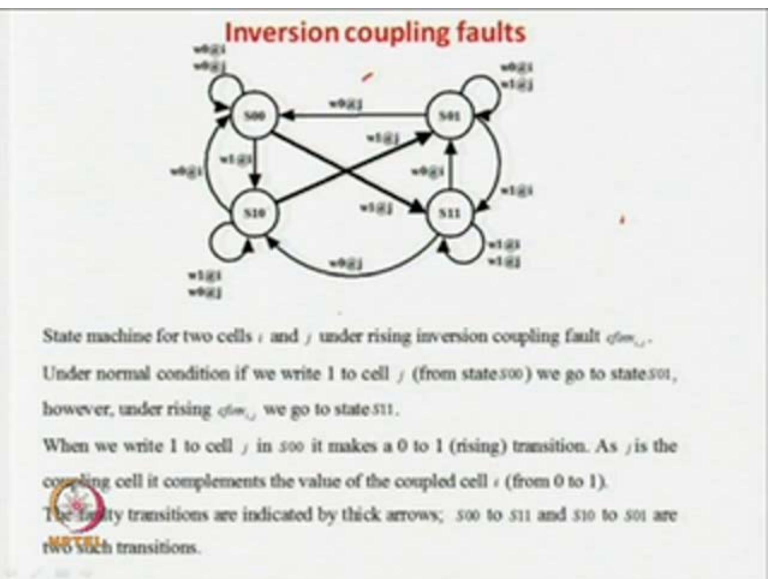
(Refer Slide Time: 32:15)



Now, again let us try to you can represent with, this one, this type of diagram. So, this is normal now for that require we require, we know that we require i and j we require two of a called for representing coupling faults. So, with the stuck at faults, we just require one state s 1 or s 0, but in case of a coupling fault, we know that we require two cells to be used. So, the idea here is that. So, the. So, two cells are i and j. So, this is s 0 s 01 10 and 11 this is making s i and j.

So, first bit corresponding to state i, second bit corresponding to state j. So, if you see that. So, 000110 and 11. So, if I am state 00. So, if you write 0 at i, if you write 0 at j there will be a self load. So, if you want to write 1 and j. So, you will go to state 01 from here, if you want to write 0 at j. So, you will come here right. So, from here, if you want to write 1 in this cell j i. So, it will (( )) in this case it would be 10, this stands for i and this stand for j. So, from here if you want to write 1 at this cell j you write as 1 at j you go similarly, the whole stuff can be explained.

(Refer Slide Time: 33:19)



Now, we will see the inversing coupling faults. So, some examples we will see. So, we are going to see, this one is the inverse coupling faults, this one the state diagram for a inverse coupling fault. So, you see these are that two (( )) I have marked, the correspond to with some kind of a inverse coupling fault, you see what happens, in this case this transition all other cells are normal that the two transition corresponding coupling faults. Let us see, what is they a, in this case you are seeing that write 1 in j.

So, what happens is 00 your are going to write 1 in j. So, you should have 0 1 because, you are writing j to 0 to 1 this is transition, but because, of this inverse coupling faults this is a up coupling fault, what happens these values gets complimented. So, unnecessarily this will become a 1 and will go by that, if you with the normal case it should have the transition from here to here, because, you are writing 1 in j, but because, you are this up. So, also is value of x value of i is also changing form 0 to 1.

So, therefore, we are going not going to a 0, but we are not going to a s 11 similarly, also there another fault over here, like for example, the down coupling fault. So, in state j say you write as your 0 and you want to write 1 over here. So, you are going to make a sorry that initial it was a 0 to 1 initially it was a raising coupling fault and now this is the this the was the fall.

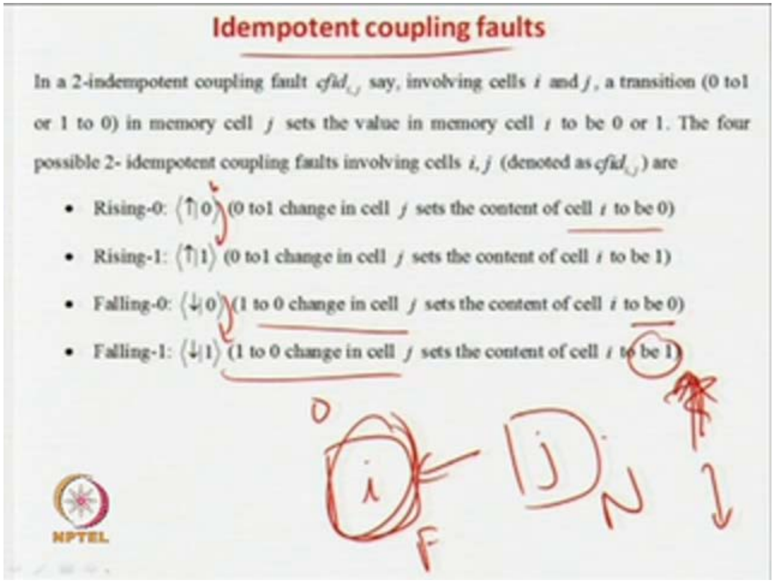
So, with that they are about same actually, we can have within both the ways. So, this is actually where both showing the raising coupling fault only, down coupling faults

coupling fault can also showing in a similar way. So, in this case you are writing same thing as I told you write 0 to 1 in the cell j and unnecessarily this s also gets complemented. Similarly, here you are writing a 0 from cell j to 1 in cell j. So, that 0 to 1 and this content of this guy, in a sorry the content of cell i is also getting changed from 1 to 0.

So, you should have you should have gone here, if you want to write a 1 you and j from 0 to 1 in energy should gonna a because, the content of i not touching by this sorry, but that the by this 1 w 1 and j, but because of this coupling fault, the content of cell j is changing is also very compliment from is also we have this two extra transition, for this up come transition, is similarly we can easily draw for a down coupling fault also you can draw, that is I mean, if you want to change the cell value from what do you call this value of from 1 to 0.

Like for example, if I having a cell from here, you want to this one, you want to change it from 1 to 0 that value. So, it is a falling. So, now, this value of x 1 will get complemented. So, you will instead of this transition, will have a transition from here to here. So, you are changing this 1 to 0 that is you are going form down in j, but along with the the value of x that is 1 will also get complemented. So, this transition will not be there in your transition parameter.

(Refer Slide Time: 36:26)

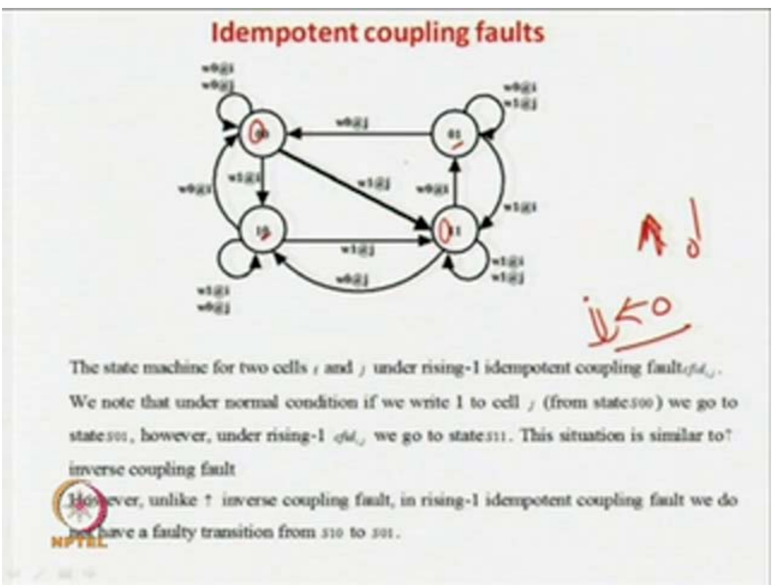


So, this one you can design a module for a down, I mean this fault that is actually called this, falling inversion coupling fault. For this why you can draw the state diagrams, for all this this is about, these are about up. So, you you can also draw it for the down. Now, the another type of idempotent coupling faults that we are going to see. So, what is the different types I mean let us in this case, this was a what you can what do you say is that. So, inversion coupling fault actually is a compliment 0 to 1 or a 1 to 0. So, that was about the inversion coupling fault, other type is the idempotent coupling fault.

So, in important coupling faults are the difference what is the idea. So, in this case again there are two cells i and cell j. So, this will actually couple this is coupling cells, this is the coupled cells, this is normal and this actually calls a fault FN. So, what is that, there are four types, rising if there is raising form here to here, then in this case the value of x will be set to 0, whatever may be the cell, what is change in 0 to 1 and cell changes the content of i to be 0, if the 0 to be 0 fine, if it is 1 it would be 0 with something like up up here, will actually make here stuck at 0.

Similarly, it can happen that raising 1. So, if is the raising at n. So, it can make this x that x i sorry stuck at 0. Similarly, the other way if there is a fall in this n j sorry the falling j falling j, will set the content of i to be stuck at 0 at i, similarly if there is a fall in cell j, the content of i is to be 1 that is stuck at one fault. So, in other case, we had a transition, if the raise here. So, that is that was about the inversion coupling fault.

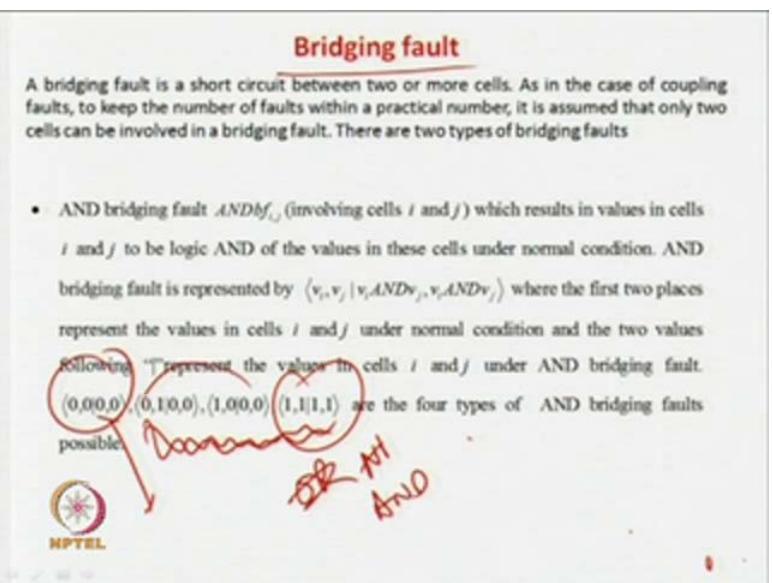
(Refer Slide Time: 38:15)



So, if there are, this is the this raise here there is a fault, if there is raise here there is a fault. So, as and in case of and important slightly different, even if there is raise or a fall. So, raise or a fall. So, the x also i actually the fault, where this fault, the coupled cell either will stuck at 0 or stuck at 1 by default, So, this slightly difference in this. So, you will see this is the idempotent coupling fault example. So, in this case what we have done. So, in this case you are changing from i your changing form 0 to 1 for this case and now it is actually making the stuck at faults. So, in this faults you are having, you are changing j from 0 to 1 and value of x gets changed to 1. So, it is stuck at one kind of other thing.

But, now if you remember the previous incidents that is the reverse coupling, what do you call inversion coupling fault, so there was an another fault over here, so in this same thing was happening were changing the value of i sorry j from 0 to 1. So, the value of x was getting flipped, where sorry i was getting flipped from 1 to 0, but in this case it is not a flipped, in this case the value of 0 i is actually permanently stuck at 0. So, whenever there is change in i come 0 to 1. So, you do not have, this transition in case of a idempotent coupling fault compared to other case of inversion coupling fault.

(Refer Slide Time: 39:12)



Now, third fault model is a bridging fault, this also very well known fault model in in case of what you can call, were circuits also in combinational sequential circuits also you have bridging faults. So, what is a bridging fault, is a similar stuff like what you can call

like this coupling fault and all, in this case what happens, there are two cells say i and j. So, the both the cell will act each other. So, that is the important, in the case of a coupling fault there was a 1 person was dominating and the other person was getting dominated.

So, this j was affecting i kind of a thing, but here both of them will affecting just because, there are getting briefed, the bridging fault there is something called AND bridging and bridging something called a OR bridging, in case of and bridging what is going to happen. So, if there is a value of 0 and here, 1 here, both of them are anded the answer is 0. So, we will get 00 over in both the cells, these are called AND bridging fault. Now, if both the cells like cells i and j, i and j is here.

So, if I say 01 and bridging fault. So, this guy will be getting changed to 0 in this case i is affecting j, but using a AND bridging fault. Now, OR bridging fault, let us say that in case of OR bridging fault 0 and 0 or 1 or if 1. So, in this case this 1 will be converted to 1 to cell will be 1. So, in case of OR bridging fault the value of j is the dominating over i kind of a bridge. So, here both can dominate each other.

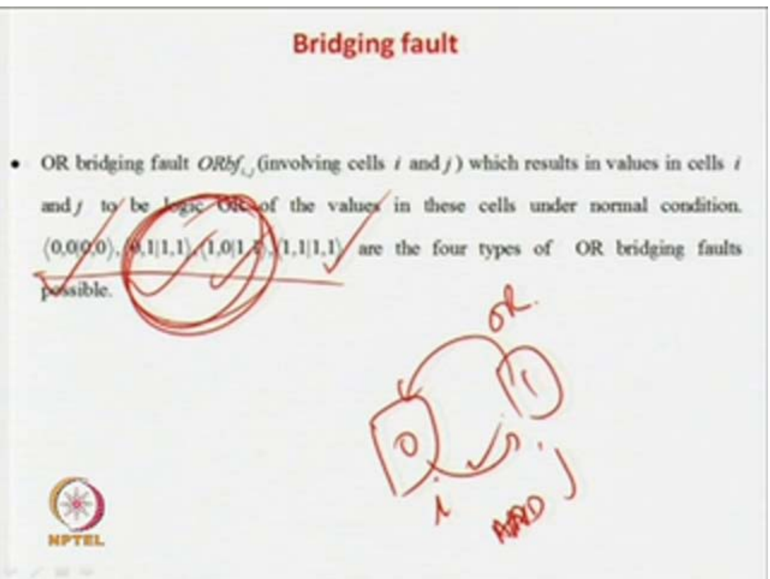
So, there is nothing called one is the dominator and other getting dominates. So, both can dominate sorry both of them can get affect by the other and in case of. So, if the value is 00 the output is 00, in case of AND bridging because, and of 0 and 0 is 0011. So, both of them kind will be 0 kind of stuff because, of what do you can say and of 1 minus 0 is 010 both 011 is both them are leading 1 and 1 is this one.

So, in this case this one and this one, use of the AND bridging fault, they are not getting affect with the AND bridging fault, but there is a list, what you can call these two cases and are affected by AND bridging fault. Like because, 0 is 1 and still 1 is in another cell. So, both will be converted to 00. So, there is a problem, similarly it is 10 or 11 the output it will be under case of and bridging fault also 11 is the output will be 11 kind of a stuff and 00 the output will be 00 also, so no problem there is no changing in case of a fault in case of AND bridging fault.

So, if you taking a OR bridging fault, this is 0 and 0 or 0 is 0. So, this will be 00 0 and 1 it will be 1 and 1 in case of a or 1 0's will be 1 and 1. So, this is the problem and 1 and 1 or is 1. So, you get 1. So, in case of OR bridging fault, this was will be getting this process will be no problem in 0 if a cell as both the cells as 00. So, AND bridging, OR

bridging no problem, but all these three are having problems in case of OR bridging and if you are using AND bridging. So, only this only never problem in case of AND bridging only this will be have a problem in and bridging. So, this about the AND bridging fault.

(Refer Slide Time: 41:59)



Well as I told you are the OR bridging faults like 00 to 00 01 will be 11. So, these are two gates, with are having some kind of a problem in case of OR bridging fault correct. So, and in case that is what. So, in case of AND bridging or OR bridging whatever you take. So, you are going to have a problem mainly with this two blocks, here two things because, is a both and and or whatever you take these two things are absolutely proper.

So, in case of OR bridging fault, so if i and j both of them can affect each other, there is nothing called dominating or being dominated, but they should have different values to be (( )) problem, in case of a for like both of the 00 there is no problem in both and them are 0 or 0 is 0 and 0 and 0 if both of them 11. So, oaring them will get also 11, if both an ending of this is again 11. So, no problem, but this is 0 and, but if it is 0 and if it is 1 then in case of OR bridging fault, this guy will be dominating this one, in case of OR bridging fault and if AND bridging fault then this guy will be dominating this one. So, this the case, so two cells will have different value to be getting affect by a bridging fault.

(Refer Slide Time: 43:05)

Neighborhood pattern sensitive coupling faults

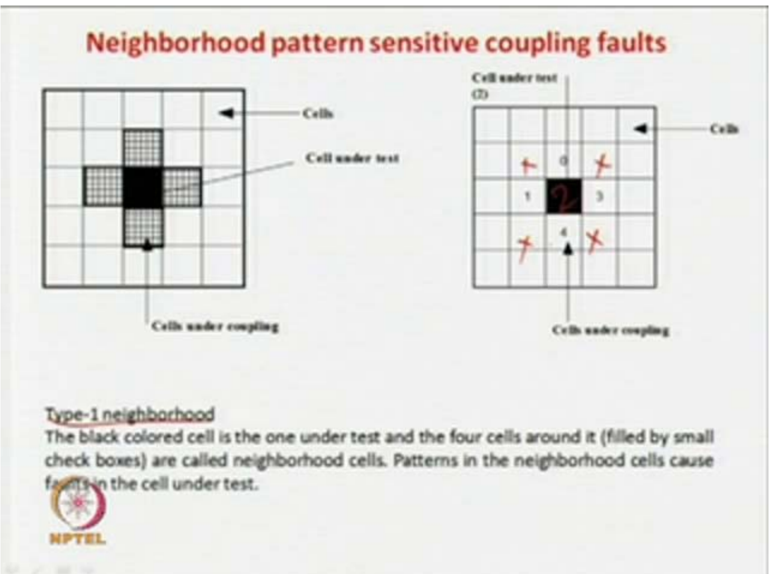
One of the most important and different kind of fault in memory compared logic gate circuits is neighborhood pattern sensitive faults (NPSFs). As memory cells are very close to each other, the cells behave normally except for certain patterns in the neighborhood cells. For example, if a cell i has 0 and all the neighboring cells have 1, then the value of cell i may be pulled up to 1. It is obvious that given a cell there can be infinite number of neighborhood combinations. However for all practical cases there are two types of neighborhoods used in fault modeling for the cell under test.

Next is something called which is very important in case of, what do you can call this memory testing as again emphasizing is called the neighbourhood, pattern sensitive fault, that is very important, that is in case of memory you have this types of cells, kinds of thing and this is the cell, so around that you will have, another last numbers of cell cells. So, all this fail to effective. So, the it is been affected by neighbour.

So, this type of fault problems you can see, if not available or if not they or means are not actually studied or required, in case of sequential and combinational circuits because, they were I there is nothing no symmetric is there, neighbours also affect, but this affect is less, but must less compare to memory cell because, in memory accessing is very much compact, placed were near each other that is actually the problem over here.

That is why, we will say that the different kind of memory fault in some is actually called, memory one of the most and different kind of fault modelling in the vey compact logic gates, such that the neighbour is this one, that this is what I told you. So, once in me very near to each other. So, you should be here, for example, an with different example there then will come into that. So, basically the idea that, this is the case it is the surrounded by things which is very near and compressed in nature. So, they are effective.

(Refer Slide Time: 44:19)



So, that is why the most important for in case of memory and quite different from circuit I mean they are the logic circuit. So, that there two types I mean neighbourhood that are been considered to see what they are type one neighbourhood. So, you mark this cell as 0123 and 4. So, these guys are affecting this, so this is one thing we assume that, diagonal elements are bit farer away from this cell. So, they are not having that kind of cell because, you see, if you generalise the a neighbouring concept, you can say that this is one neighbourhood, you can say this in one another neighbourhood, further an other kind of neighbourhood.

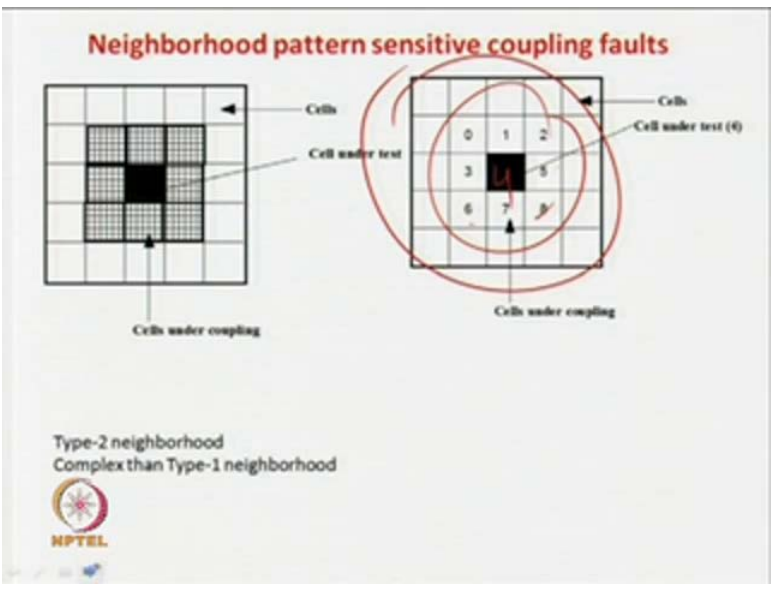
So, they neighbourhood can set also be this way. So, the different tangents of, what you call different types of like for example, (( )) coupling you said that 2 cells can couple, 3 cells can couple, 4 cells can couple, so for, but in case of the neighbourhood to define a neighbourhood is a very, very exponential problem in physical kind of problem. Neighbourhood can be anything, you can also also consider as this guy, this guy, this guy and this guy as well, which are diagnose each other for a very specific case type of circuits.

Also also you can say that, these things all these things can be considered as a neighbourhood, so and also you can say that, these things are all considered as neighbour this one, this one, this one and this one are consider as neighbour here, we are not considering the diagonal elements are not considered. So, the idea is that neighbourhood

can be constantly in a any type of fashion and that is actually going to kill you. So, you cannot have, any any type of neighbourhood defined and is not actually recovered because, of the exponential nature the problem.

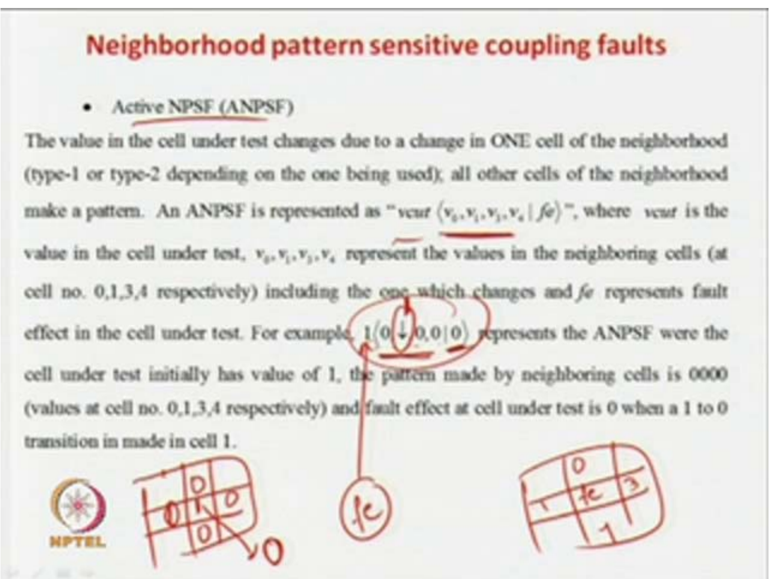
So, what you do is that, we define in a testing and band statically and many other matter they found out that, there are two kind of neighbourhood, which in which is enough to do this testing and which can reasonably, lowering complexity as well as it can be reasonably with an good coverage and confidence, so this one type of type one neighbourhood, in which we have this cell. So, these things are not diagnosed cells are not involved in this and 0 1 and 2 is this cell which is an effect, 3 and 4 are the neighbourhood cells is another type of neighbourhood.

(Refer Slide Time: 46:06)



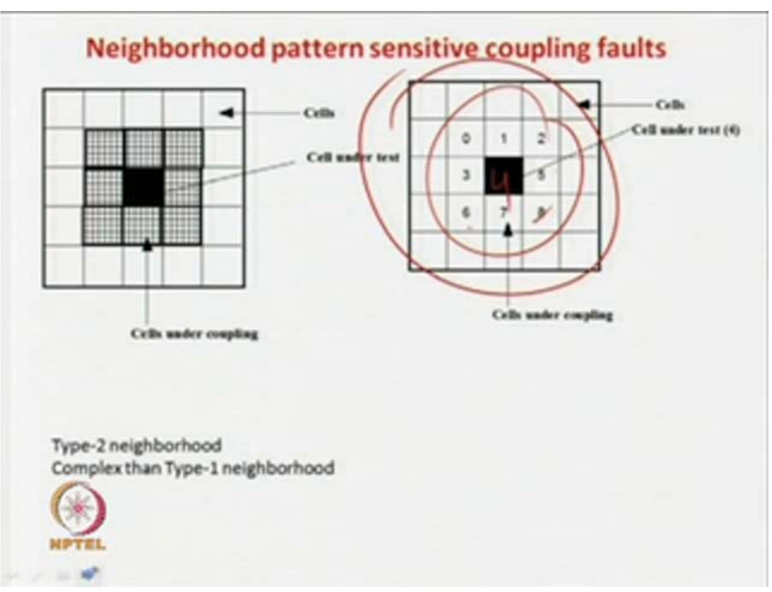
(( )) more complex. So, it is they also consider diagonal elements, can also will like this one, this one and also to be involved in this coupling because, they are all nearby also they they bit more complex, but takes more, I mean more number of cells in concept ration making this fault modelling effects. So, here is 0, this is 123, this one is 4 is the affected 15678. So, this is a more complex coupling fault sorry neighbourhood fault model, but you can also consider the third area fourth in a making it horrible complex and it will grow considering. So, in testing, we consider only these type of this one.

(Refer Slide Time: 46:40)



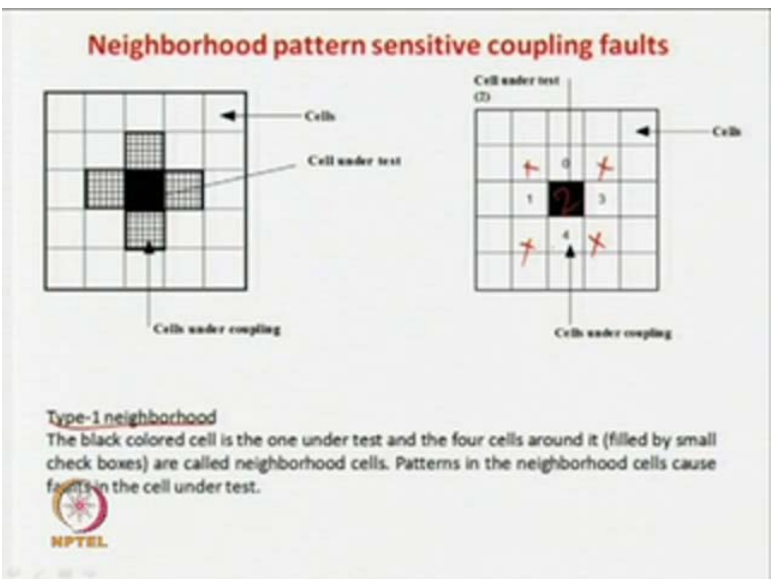
So, type 1 is, but for type 2, I mean we consider bit more complex stuff which will not discuss.

(Refer Slide Time: 46:45)



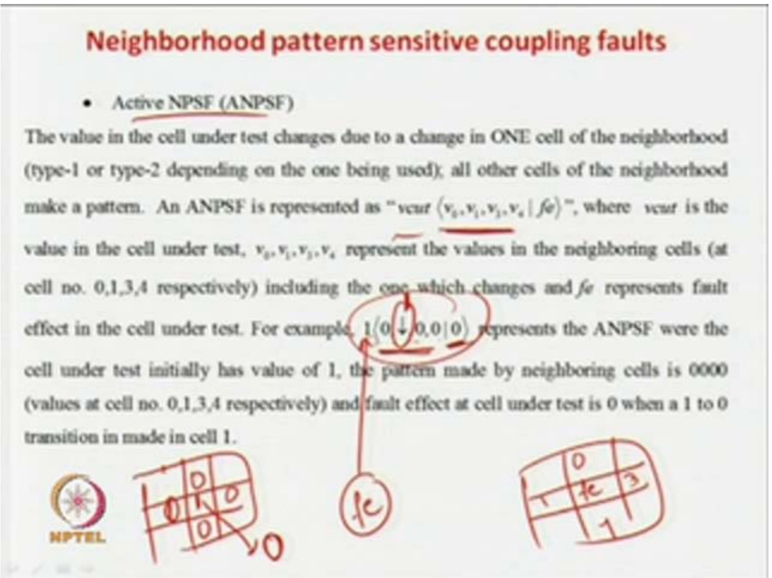
You will think that you have to assume also 8 1 2 3 4 5 6 7 8, 8 different type of I mean 8 different neighbourhood cells have to consider.

(Refer Slide Time: 46:52)



While in case of one you need to consider 0 1 2 3 only sorry 1 2 3 4 only 4 cells which is in the peripheral.

(Refer Slide Time: 46:58)



So, now, we will see the different type of two or three different types of faults models, in this one in case of neighbourhood pattern sensitive coupling faults. So, one is the active one. So, what is a active one. So, active one says that. So, v 0, v 1, v 3, v 4 these are the four cells, which are actually peripheral of this one 0 1 3 4 and this two cell is a actually been affected. So, that is why we write 01234 there all these cells which are very nearby

this one, at these are you can say that, 013 and 4 and this is your affected part, this one right and we can this value of some cut something like this.

So, now this is one type of a fault. So, what is this one active active neural sensitive part, active means there should be some activity in this four cells, where this is this one. So, it is seeing that initially the value of f e equals to 1 for value equal to 1, this value equal to 1, initially.

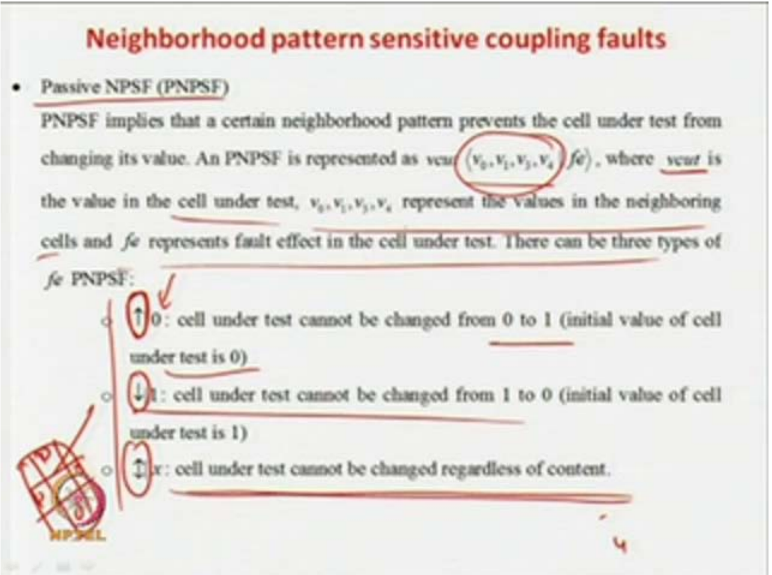
Now, you say that, all the other values was 0000 where the four values, this this guy, this guy, this guy, were values are having sorry sorry sorry this value were having 1, initially it was 0110, initially this configuration was nothing, but initially it was something like this, it was having the value of 0 was having the value 0, 1 was very thing that of 1, this were having the value of 0, this 1 was having value of 0, f e was having the value of 1.

Now, we says that, what is default is a slight this is a drop in this one. So, this one changes to 0, so if you do this it sayes that this cell also get affected and it is converted using now, why is it a kind of thing. So, this things. So, near about everything one of this 0. So, all of them together, push it down to 0, but initially it was not 0 because, this cell was also 1 and that was actually had to be to 1.

So, now this gate to be a 0 for some reason. So, this by being periphery of all other 0's get change to 0. So, it is I mean 0. So, because of this activity from 1 to 0 in this term (( )) this one is example of a active never root faults that is because, activity here or here there also try to pull this guy to 1 and 0, so this is one example of this one.

So, this called active because, active means in some changes in any one of this neighbourhood change from 0 to 1 1 to 0 there actually happening at same (( )).

(Refer Slide Time: 49:11)



This one is the active, this something called passive, is something called passive. Now, what is called a passive, we call this is as a passive because, in this case we will not have many changes in the never roots cells because, of some passive nature only well be 0 or 1 some natures, some pattern nature, we also have felt this internal parts we will see. So, in this case, we say that v 0, v 1, v 3, v 4. So, v 1 this is v 3 v 4. So, here some values, we call this value of this cell under that is true and these are represent the values of the neighbouring cells and f e represents the fault end cell.

So, there three kinds this is the example is better, it says that due to some pattern of over here. So, these are these are your cells. So, this is 1, sorry 01234 this is f e. So, this is a 101234 because, of some value over here, like you can say that small values like 111 and now 1 something like this, you cannot change these values from 0 to 1 or also sorry we can say we can say all these things are 0 let us assume let us assume all the values are 0 over here.

So, now, it says that v cut, this v cut means this little cell that is under test, we want to read from 0 to 1, but all other peripheries are 0, all other 0 assume 1 this is 1, value 0, value 1, value 3, value 4, value 0, 13 4 kind of 0 134 all these are 0. Now, because of this everything is 0 nearby, so now it will not allow the central part this is cell number 2 or f e 2 raise of 0 to 1. So, it says that, we want to raise the values of this cell under test from

0 to 1, initial value of cell will assume, but these not along (()) this is a one kind of a fault.

So, similarly it says that, this may be one kind of all can be like say for example all the these if all these things are now 1111 this is also 1 now, you want to say that, you want to make this one from 1 to 0 that is we want make it fault, but because all of the peripherals are 1, you will not be able to design. So, this another kind of a neighbourhood pattern sensitive for means passive in nature, the all these as 0, this all these are ones. So, passive we are trying to hold the values of the other neighbouring cell, which is being affected and this is a active is a what was the thing.

Same thing was there, but do they used to some kind of activity in this scalar, this scalar, this scalar, this which is to be changing the values. So, there are too slight difference, another thing is that I mean because, of some pattern sensitive defaults, I means because, of some passive faults, what you were doing, you were actually unnecessarily the cell cannot be changed regarding of context.

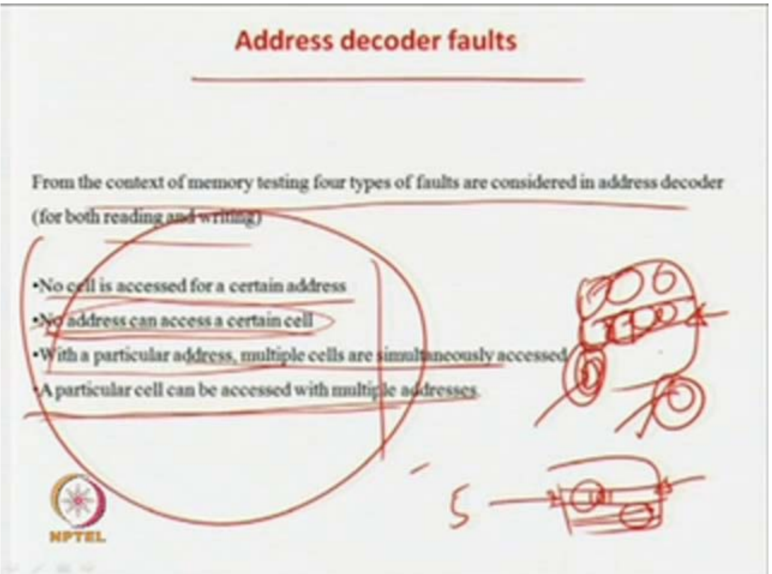
Like in this case, this some, this cell is there some values you put like 0000 or if you put 11 some values were there, but now this cell become fixed for some (( )) were try to change this cell to 0 to 1 1 to 0 that you are not avail to do this, this cell becomes fixed at the values. So, there are three types of neighbourhood pattern, cell passive pattern neighbour pattern part. So, why they are called passive because, in this case some values are there in the cell number 013 and 4 and there are affecting the cell and test which is in the centre.

And why this is called active because, here some patterns are there are some actives either are here, here and here braces and the activities change changing some fault over it, that is why, this is one is called the active in nature and other the passive in nature and the very important in case of memory testing because, memory architecture is something like that, the cells are very much coupled and near each other.

So, what happen, wants fits in other in a neighbourhood fashion this, is like take an example that, if a would person is in is in amongst friendship of several bad friends, the person front try to will try to become bad, in other way similarly like if a bad person is among, all other persons which which we are good I mean, if a person is bad, but he stays with good people and he tends to happens that he also become a good person.

So, there is a neighbourhood are is having a very much affect from the cells. So, that is one another (()) we have seen.

(Refer Slide Time: 52:53)



So, there is last part is that actual address decoder faults, which is called the address decoders are nothing, but there another type of combinational circuit decoder, they do not test there in a normal manner like a combinational circuits because, or idea is that, if you do not want because, it is coupled to a memory. So, it is job is only to access some memory locations and deliver the value by the, what do you can call cells amplifier and the driver cells.

So, that there are actually two of the parts of the where I told you the analog parts, which is the sense amplifier, which sense the values of the circuit or something something which drives the values, to write the memory cell. So, those parts also will not explicitly and similarly, decoders are not else explicitly. So, what will do is that because, the idea of the memory decoder is that, you select some cell you write it and you retrieve it back.

So, this decoder should happen to select the cell property that is what the idea, no cell is accessed for a certain access I means. So, from the context of memory testing, four types of faults you can considering address decoding, with the particular address, multiple cells are simultaneously accessed which should not be possible it should not be the case, particular cell should be accessed with a multiplexed this should not be possible that is

say this is the these are cell, these are cell kind of a thing. So, it should be accessed with only one particular address that is the proper address.

So, what is the idea, with a particular cell address should be, particular cell can be accessed with multiple address that is should not be possible that is say, it is cell numbers 5, you should be able to access only with cell number, address 5 it should not be with 3215. So, with all 3215 which should be able to access this should not be possible.

Secondly, with a particular address multiples should not be simultaneously accessed the straight, you want to say access 5 says this should be access, simultaneously your access in this part and for some of the cellular accessing this part, that should also not be there. And it should not be there something like, no address for a certain cell, like for example, there is a cell, this cannot be accessed any memory any memory address, say this is cell is 7.

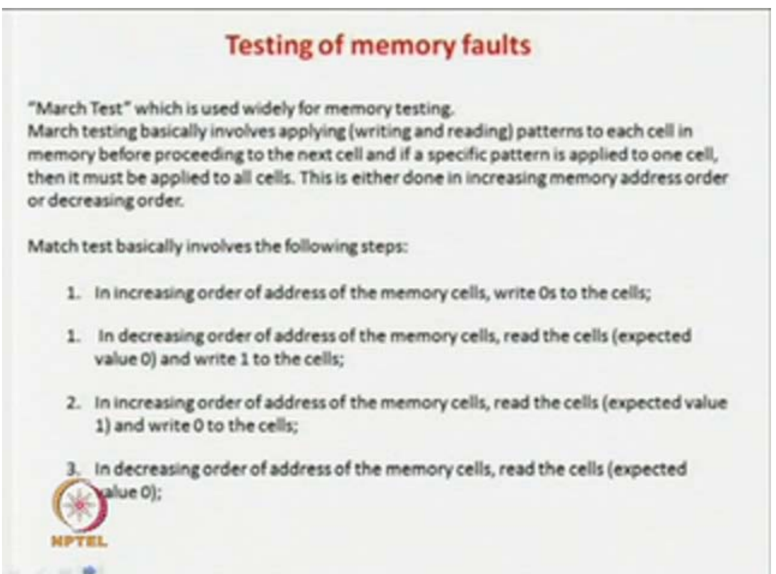
So, by 7 you should be able to access this, you should not be the case that you are trying to put, when you should not able to access is they ever, you give 7 you access some other cell and you cannot access this cell. So, that should (( )) they should not be in case, no cell access for certain address. So, here some of the cases, which are active means address decoder faults, but generally the idea is that, we may not go for testing of this things in this explicitly manner.

Then what we can do, we can access these cells sequential, which are able to all the cells sequential write proper values and read proper values, then you can be very much sure that none of these faults are there because, you should be able because, the idea is that, in one time you can access only one cell because, if you are accessing multiple cells, then whatever you are writing you will get some other kind of output, what you are expecting are output not be there, use some some cell here is been accept and other cell is being accept with physical address let not be possible.

So, I mean also these thing not be possible with the everything is proper, with the single address, multiple cell cannot be accessed, like with the particular access with multiple that is also not be there because, you are asking for something, you are getting something; that means, what all try to do is for all cells; that means, all cells are accessed, all cells are properly accessed by unique addresses and no, I means no addresses are accessing less than ones.

So, if you less access all the cells, mean one by one and check everything is proper, than all this things should be done in a proper way.

(Refer Slide Time: 55:58)



So, now. So, that was about the memory faults. So, in the next lecture what you are going to see, we are going because, you see very simple right, so the idea is that you have to access some memory location, you have to write 0, you have to read 0 and, so forth. So, that is the main idea, so you do not propagate the fault value to the output here, you did not set faults although things are not required that is very simple because, the memory should be able to hold a 0 and or 1 and you should be able to retrieve through required in a via proper addressing mechanism.

So, that is what we are going to do, you are going to sell one by one write a 0, read a 0, write a 1, read a 1, this the thing if you do sequentially for all these cells, then your job is done. So, here you do not require a standard ATPJ algorithm, de algorithm propagating the value of the outputs sense in non frequent, access these cells write a 0, read a 0, write a 1 data 1 and your going to been done. So, this is what called the testing of memory models. So, in the next lecture, we are going to see how we can go a model, this is the very famous testing of marks test. So, we should be seeing in the next lecture.

Thank you.