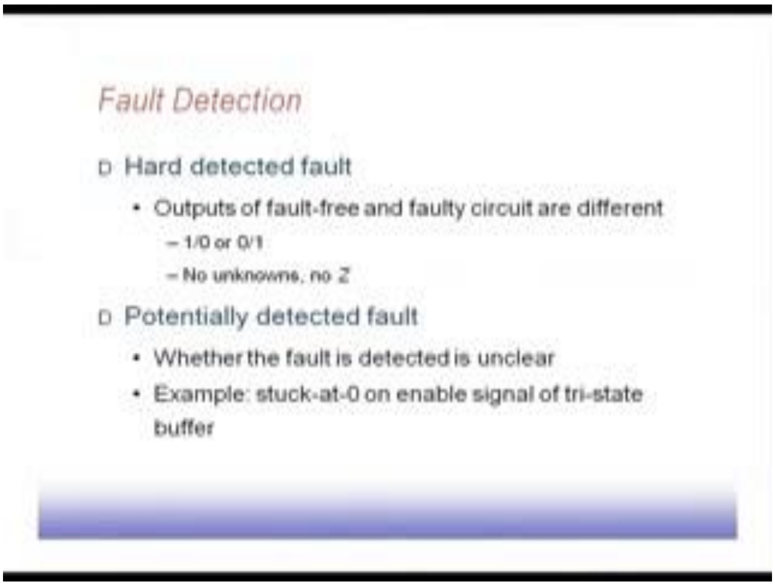
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 16
Logic and Fault Simulation (Contd.)

Fault simulation process. So, there are certain cases where the faults can be easily detected and there are certain faults which are difficult to detect.

(Refer Slide Time: 00:30)



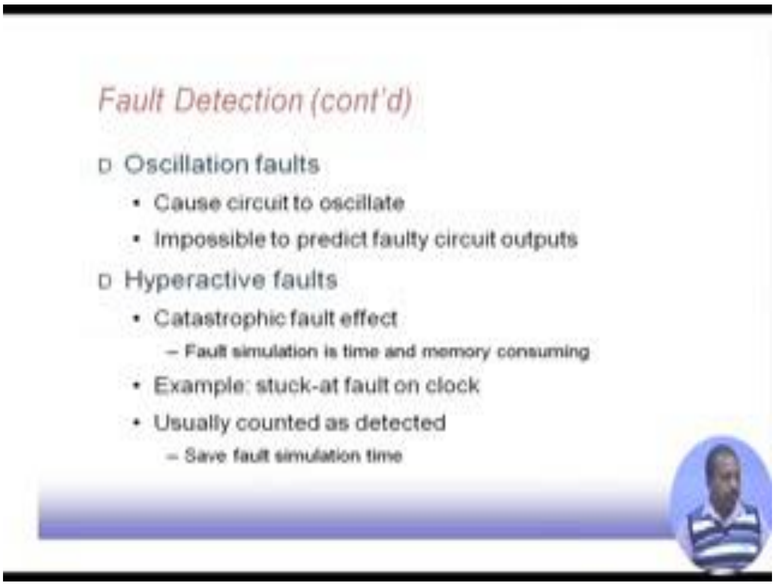
Fault Detection

- Hard detected fault
  - Outputs of fault-free and faulty circuit are different
    - 1/0 or 0/1
    - No unknowns, no Z
- Potentially detected fault
  - Whether the fault is detected is unclear
  - Example: stuck-at-0 on enable signal of tri-state buffer

There are hard detected faults. So, outputs of fault free and faulty circuit are different may be faulty circuit is 0, faulty circuit is 1 or the other wise that is 0; faulty circuit is 0 and faulty circuit is 1. So, that cases the faults are detected and there is no unknown. So, in that process none of the outputs are X and none of the outputs are Z. So, in that situation, so we say that it the one hard fault has been detected, there are potentially detected fault whether the fault is detected is not very clear.

For example, say stuck-at-0 fault at the enable signal of tri-state gate. So, if the output is Z, so you do not know whether it is due to a stuck-at-0 fault or due to some problem at the tri-state gate output itself, so that way it is difficult that we cannot detect this fault. So, whether the tri-state gate out enable signal stuck-at-0 or not. So, say there are certain faults which are oscillatory in natures; they cause the circuit to oscillate.

(Refer Slide Time: 01:31)



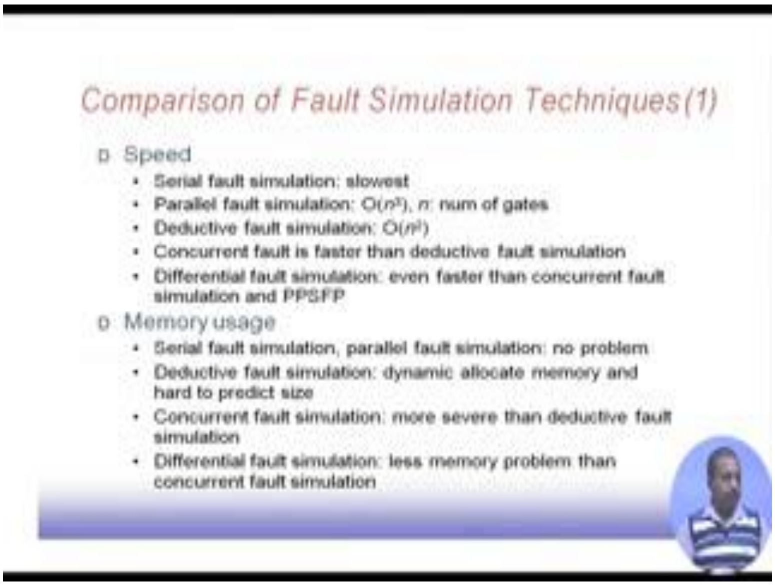
Fault Detection (cont'd)

- Oscillation faults
  - Cause circuit to oscillate
  - Impossible to predict faulty circuit outputs
- Hyperactive faults
  - Catastrophic fault effect
    - Fault simulation is time and memory consuming
  - Example: stuck-at fault on clock
  - Usually counted as detected
    - Save fault simulation time

Output is never stable and in this case, it is in impossible to predict faulty circuit outputs so what is the; because output is continually oscillating, so we cannot say. Certain faults are called hyperactive faults and hyperactive faults say there are catastrophic. So, they call lot of events in the circuit. So, as a result fault simulation becomes very time consuming as well as memory consuming. So, whatever fault simulation techniques we have seen, so if those techniques are applied for a stuck-at fault on a clock line. So, a stuck-at fault of the clock line since clock line goes to almost everywhere in the circuit.

So, it will create a large number of events. So, when it does that, so although events are to be stimulated at the same time we need to store lot of faulty gate information responses and all that. So, a memory department will also be high. So, if that thing occurs, so the fault stimulators they normally take count it has detected because so many so much of event is there. So, it will; it is definitely going to be detected. So, there may be a limitation on the fault simulation on the number of events for a particular fault and if there list if that limit crosses then that fault may be taken as detective. So, it saves the fault simulation time.

(Refer Slide Time: 02:51)



Comparison of Fault Simulation Techniques(1)

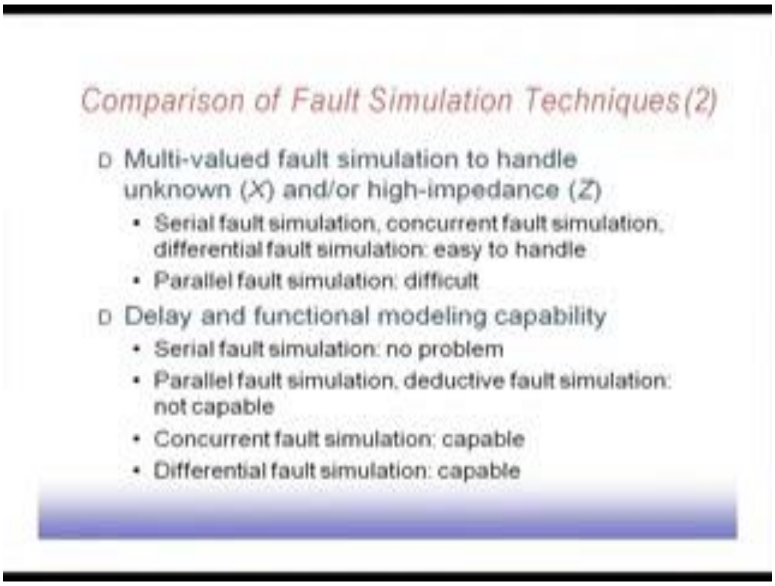
- Speed
  - Serial fault simulation: slowest
  - Parallel fault simulation: $O(n^3)$, n: num of gates
  - Deductive fault simulation: $O(n^2)$
  - Concurrent fault is faster than deductive fault simulation
  - Differential fault simulation: even faster than concurrent fault simulation and PPSFP
- Memory usage
  - Serial fault simulation, parallel fault simulation: no problem
  - Deductive fault simulation: dynamic allocate memory and hard to predict size
  - Concurrent fault simulation: more severe than deductive fault simulation
  - Differential fault simulation: less memory problem than concurrent fault simulation

If we compare across this fault simulation techniques first of all from the speed point of view. So, serial fault simulation is the slowest. So, naturally if we if it is taking each fault separately and each pattern separately and was stimulating them so it is the slowest one.

Parallel fault simulation; it is of the order of n cube where n is the number of gates. So, for deductive fault simulation is big O n square, concurrent fault simulation may; concurrent fault is faster than the deductive fault simulation. So, concurrent fault simulation because so many circuits; so many gates are created and they are done simultaneously so that simulation is faster. Differential fault simulation is even faster than fault simulation; concurrent fault simulation because it stimulates only the differential part of the circuit. From the memory usage of point of view, serial fault simulation; parallel fault simulation, there is no problem because serial fault simulation is taking only one fault and one pattern at a time, parallel fault simulation also usually taking; using the memory word or the word size of the processor.

As a result, there is no extra overrate that is coming. So, memory usage is whether you are storing 1 bit or 1 byte. So, it does not matter, ultimately it is counted it terms of words, so that way, they are same. Deductive fault simulation; so that requires dynamic allocation of memory and naturally it is hard to predict the size because which faults will get added so that is dynamic so that way it is going to have some dynamic allocation. Concurrent fault simulation is more severe than deductive fault simulation because now we have to remember all the faulty gates that are there, so whatever the due to some faults. So, what are the faulty gates outputs so that are to be stored? So, that way it is memory requirement is high. Differential fault simulation requires less memory than this concurrent fault simulation because of the simple reason that it stimulates only the differential part of the circuit.

(Refer Slide Time: 05:01)



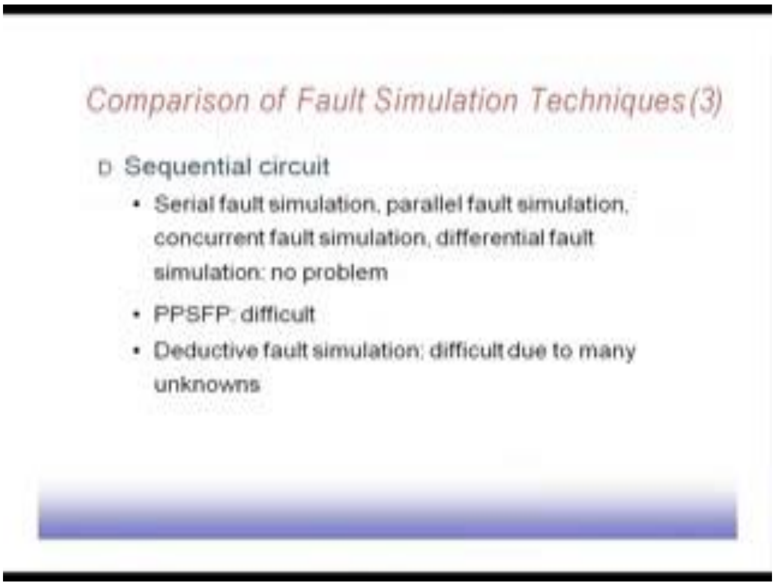
Comparison of Fault Simulation Techniques(2)

- Multi-valued fault simulation to handle unknown (X) and/or high-impedance (Z)
  - Serial fault simulation, concurrent fault simulation, differential fault simulation: easy to handle
  - Parallel fault simulation: difficult
- Delay and functional modeling capability
  - Serial fault simulation: no problem
  - Parallel fault simulation, deductive fault simulation: not capable
  - Concurrent fault simulation: capable
  - Differential fault simulation: capable

Multi valued fault simulation to handle unknown X or high impedance Z, so how can we handle this situation for serial fault simulation, concurrent fault simulation and differential fault simulation? It is easy to do because there we have got well defined logic by which we define the operations on X and Z. So, accordingly we can handle those situations, but parallel fault simulation is difficult because now in the memory word I need to store X or Z. So, that way it becomes a difficult one.

Now, delay and functional modelling capabilities, serial fault simulation there is no problem because it; you can say that after this much time these particular fault; these transition is going to occur so you can stimulate after that time. Parallel fault simulation; deductive fault simulation, so they cannot handle delay and functional modelling faults because particularly the delay faults because now if you are trying to module delay fault and every discrete time interval. So, we need to keep a copy of the circuit. So, that is difficult. A concurrent fault simulation; we can do this delay fault testing and differential fault simulation; this is also able to do this delay fault simulation. Sequential circuit so see a serial fault simulation, parallel fault simulation, concurrent fault simulation and differential fault simulation, there will no problem.

(Refer Slide Time: 06:16)

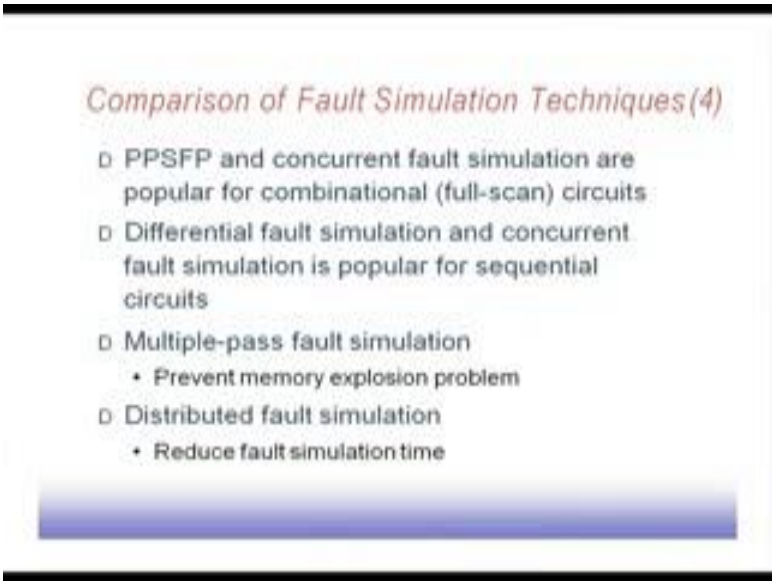


Comparison of Fault Simulation Techniques (3)

- Sequential circuit
  - Serial fault simulation, parallel fault simulation, concurrent fault simulation, differential fault simulation: no problem
  - PPSFP: difficult
  - Deductive fault simulation: difficult due to many unknowns

Parallel pattern single fault propagation, so this particular case; various difficult, deductive fault simulation is difficult because there may be many unknowns. So, sequential circuit will create more number of unknown as there as flip flops. So, due to the increase in the number of unknown, deductive fault simulation is difficult for parallel sequential circuits.

(Refer Slide Time: 06:46)



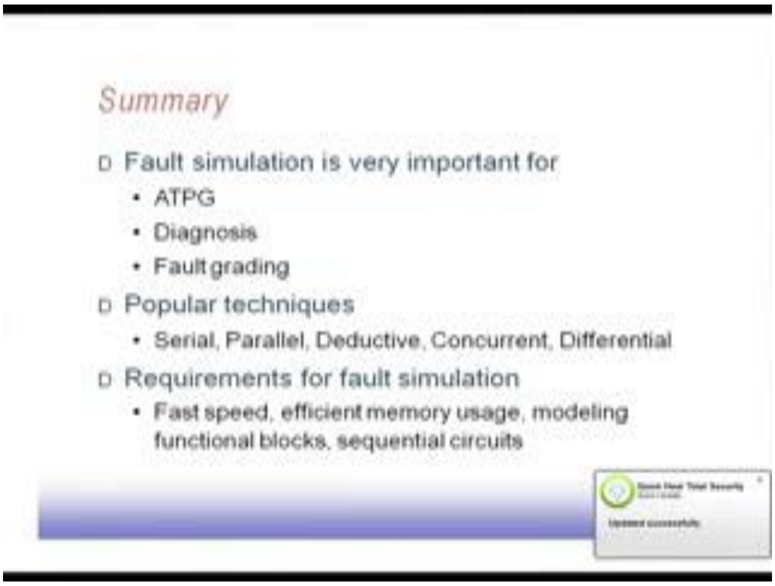
Comparison of Fault Simulation Techniques (4)

- PPSFP and concurrent fault simulation are popular for combinational (full-scan) circuits
- Differential fault simulation and concurrent fault simulation is popular for sequential circuits
- Multiple-pass fault simulation
  - Prevent memory explosion problem
- Distributed fault simulation
  - Reduce fault simulation time

This PPSFP and concurrent fault simulator are popular for combinational circuits, combinational and full scan circuits, full scan means sequential circuit where all flip flops are connected over a scan change. So, that become a full scan circuit, for those cases, these PPSFP and concurrent fault simulations; so they are popular, differential fault simulation and concurrent fault simulation, they are popular for sequential circuit.

We have seen multiple paths fault simulation, so because in 1 pass, it creates all the events and in the next pass, it have the evaluation. So, this prevents memory explosion problems. So, that way it is good. Distributed fault simulation, so it reduces fault simulation time. So, if we can distribute the overall fault simulation job over a number of computers so that there can be distributed over a number of processors so as a result the fault simulation time will get reduced.

(Refer Slide Time: 07:46)



Summary

- Fault simulation is very important for
  - ATPG
  - Diagnosis
  - Fault grading
- Popular techniques
  - Serial, Parallel, Deductive, Concurrent, Differential
- Requirements for fault simulation
  - Fast speed, efficient memory usage, modeling functional blocks, sequential circuits

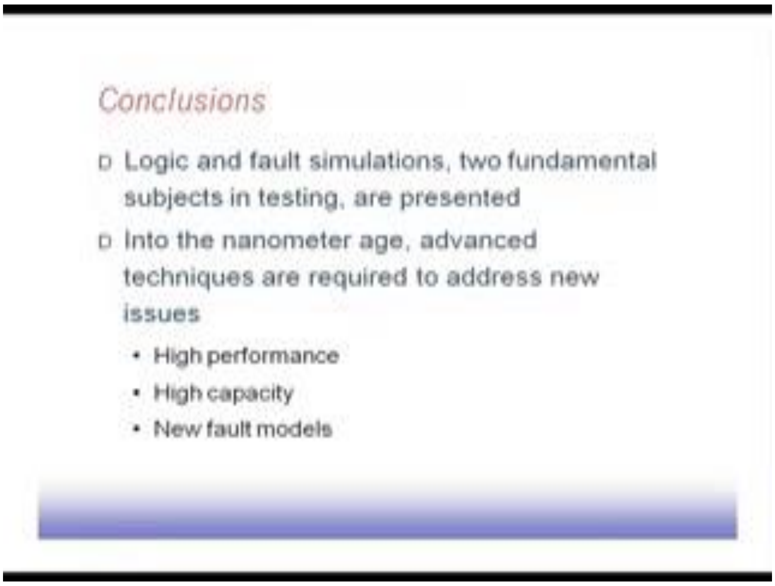
Quick Heal Total Security

Updated successfully.

To summarize fault simulation is a very important from many angles like ATPG. So, if you are trying to develop some alternative test pattern generation algorithm so we need to see about the faults that are already covered by the generated test set and then we tried to augment it by generating further test pattern. So, that way ATPG algorithms, so they very well need this fault simulation. For diagnosis purpose, we need to know what are all the faults detected by a particular pattern. So, we need to identify all those faults so for diagnosis purpose. So, that also there also fault simulation is important. Fault grading like how difficult is a fault to capture. So, that way we can grade that type of faults that are present in the circuit, some faults are hard to detect, some faults are random resistance fault. So, that way there are so there can be several grading.

Popular techniques for fault simulation include parallel, serial, deductive, concurrent and differential simulation. So, what we are looking for in fault simulation is the speed should be high, memory usage should be low, functional blocks; we should be able to model and also we want that sequential circuits also to be treated in fault simulation though we can say that sequential circuits are mostly converted today into scan circuits and then it becomes a combinational testing problem.

(Refer Slide Time: 09:20)



Conclusions

- Logic and fault simulations, two fundamental subjects in testing, are presented
- Into the nanometer age, advanced techniques are required to address new issues
  - High performance
  - High capacity
  - New fault models

That way, these sequential circuit fault simulation is may not be that much required. So coming to the concluding remarks, so logic and fault simulation; there are 2 fundamental subjects in testing that that we have discussed and into the nanometre age, advanced techniques are required to address new issues like high performance, high capacity and naturally the type of faults or type of defects that can occur in the new generation technology. So, that is going to differ from the other; from the previous technology as a result the fault models also need to be updated so that we can take care of those defects. So, new fault models are also necessary.

In the next, we will continue with the test pattern generation techniques.