

ECE 553: TESTING AND TESTABLE DESIGN OF DIGITAL SYSTEMS

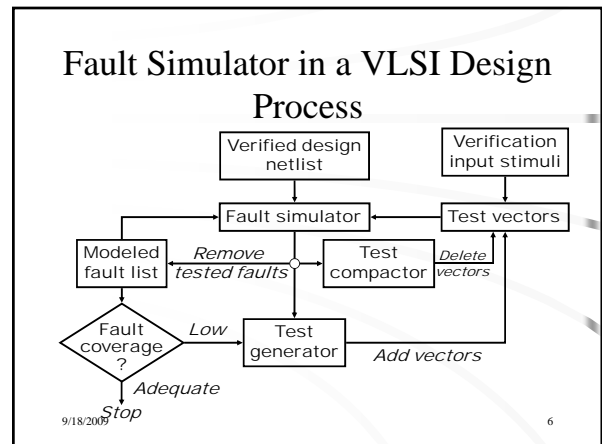
Fault Simulation

- ### Overview
- Problem and motivation
 - Fault simulation algorithms
 - Serial
 - Parallel
 - Deductive
 - Concurrent
 - Other algorithms
 - Random Fault Sampling
 - Summary
- 9/18/2009 2

- ### Problem and Motivation
- Fault simulation Problem: Given
 - A circuit
 - A sequence of test vectors
 - A fault model
 - Determine
 - Fault coverage - fraction (or percentage) of modeled faults detected by test vectors
 - Set of undetected faults
 - Motivation
 - Determine test quality and in turn product quality
 - Find undetected fault targets to improve tests
- 9/18/2009 3

- ### Usages of Fault Simulators
- Test grading – as explained before
 - Test Generation
 - Fault diagnosis
 - Design for test (DFT) – identification of points that may help improve test quality
 - Fault-tolerance – identification of damage a fault can cause
- 9/18/2009 4

- ### Alternatives and Their Limitations
- Prototyping with fault injection capabilities
 - Costly
 - Limited fault injection capability
 - Design changes hard to implement
 - Long lead time
 - Hardware emulators
 - Costly
 - Require special hardware
- 9/18/2009 5



Fault Simulation Scenario

- **Circuit model: mixed-level**
 - Mostly logic with some switch-level for high-impedance (Z) and bidirectional signals
 - High-level models (memory, etc.) with pin faults
- **Signal states: logic**
 - Two (0, 1) or three (0, 1, X) states for purely Boolean logic circuits
 - Four states (0, 1, X, Z) for sequential MOS circuits
- **Timing:**
 - Zero-delay for combinational and synchronous circuits
 - Mostly unit-delay for circuits with feedback

9/18/2009 7

Fault Simulation Scenario (continued)

- **Faults:**
 - Mostly single stuck-at faults
 - Sometimes stuck-open, transition, and path-delay faults; analog circuit fault simulators are not yet in common use
 - Equivalence fault collapsing of single stuck-at faults
 - Fault-dropping -- a fault once detected is dropped from consideration as more vectors are simulated; fault-dropping may be suppressed for diagnosis
 - Fault sampling -- a random sample of faults is simulated when the circuit is large

9/18/2009 8

Fault Simulation Algorithms

- Serial
- Parallel
- Deductive
- Concurrent
- Others
 - Differential
 - Parallel pattern
 - etc.

9/18/2009 9

Serial Algorithm

- **Algorithm:** Simulate fault-free circuit and save responses. Repeat following steps for each fault in the fault list:
 - Modify netlist by injecting one fault
 - Simulate modified netlist, vector by vector, comparing responses with saved responses
 - If response differs, report fault detection and suspend simulation of remaining vectors
- **Advantages:**
 - Easy to implement; needs only a true-value simulator, less memory
 - Most faults, including analog faults, can be simulated

9/18/2009 10

Fault Injection

- Modifying netlist for every run can be expensive
- **Alternative**
 - Check if a net is faulty or fault-free
 - If faulty change its value to the stuck-value
 - Else leave it to the computed value
 - Mux based fault insertion
 - Use additional variables and computed the value based on the signal value and the value in the additional variable

9/18/2009 11

Serial Algorithm (Cont.)

- **Disadvantage:** Much repeated computation; CPU time prohibitive for VLSI circuits
- **Alternative:** Simulate many faults together

```

    graph LR
      TV[Test vectors] --> FFC[Fault-free circuit]
      TV --> CF1[Circuit with fault f1]
      TV --> CF2[Circuit with fault f2]
      TV --> CFn[Circuit with fault fn]
      FFC --> C1[Comparator]
      CF1 --> C1
      C1 --> D1[f1 detected?]
      FFC --> C2[Comparator]
      CF2 --> C2
      C2 --> D2[f2 detected?]
      FFC --> Cn[Comparator]
      CFn --> Cn
      Cn --> Dn[fn detected?]
  
```

9/18/2009 12

Parallel Fault Simulation

- Compiled-code method; best with two-states (0,1)
- Exploits inherent bit-parallelism of logic operations on computer words
- Storage: one word per line for two-state simulation
- Multi-pass simulation: Each pass simulates $w-1$ new faults, where w is the machine word length
- Speed up over serial method $\sim w-1$
- Not suitable for circuits with timing-critical and non-Boolean logic

9/18/2009 13

Parallel Fault Sim. Example

9/18/2009 14

Deductive Fault Simulation

- One-pass simulation
- Each line k contains a list L_k of faults detectable on k
- Following true-value simulation of each vector, fault lists of all gate output lines are updated using set-theoretic rules, signal values, and gate input fault lists
- PO fault lists provide detection data
- Limitations:
 - Set-theoretic rules difficult to derive for non-Boolean gates
 - Gate delays are difficult to use

9/18/2009 15

Deductive Fault Sim. Example

Notation: L_k is fault list for line k
 k_n is s-a-n fault on line k

9/18/2009 16

Concurrent Fault Simulation

- Event-driven simulation of fault-free circuit and only those parts of the faulty circuit that differ in signal states from the fault-free circuit.
- A list per gate containing copies of the gate from all faulty circuits in which this gate differs. List element contains fault ID, gate input and output values and internal states, if any.
- All events of fault-free and all faulty circuits are implicitly simulated.
- Faults can be simulated in any modeling style or detail supported in true-value simulation (offers most flexibility.)
- Faster than other methods, but uses most memory. 17

9/18/2009

Conc. Fault Sim. Example

9/18/2009 18

Other Fault Simulation Algorithms

- Parallel pattern single fault simulation (PPSFP)
 - Simulate many vectors in parallel
 - Inject only one fault – hence one event
 - Simulate the circuit from the fault site
 - Limitation – well suited for combinational circuits only

9/18/2009

19

Fault Sampling

- A randomly selected subset (sample) of faults is simulated.
- Measured coverage in the sample is used to estimate fault coverage in the entire circuit.
- Advantage: Saving in computing resources (CPU time and memory.)
- Disadvantage: Limited data on undetected faults.

9/18/2009

20

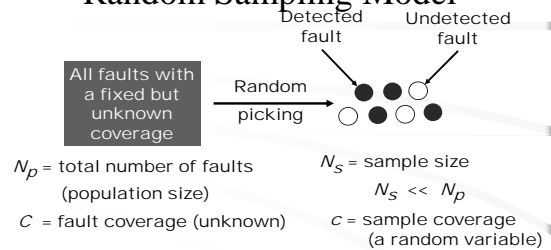
Motivation for Sampling

- Complexity of fault simulation depends on:
 - Number of gates
 - Number of faults
 - Number of vectors
- Complexity of fault simulation with fault sampling depends on:
 - Number of gates
 - Number of vectors

9/18/2009

21

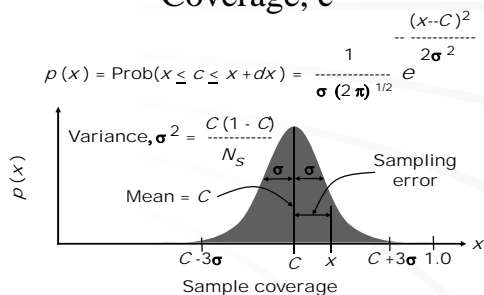
Random Sampling Model



9/18/2009

22

Probability Density of Sample Coverage, c



9/18/2009

23

Sampling Error Bounds

$$|x - C| = 3 \left[\frac{C(1-C)}{N_s} \right]^{1/2}$$

Solving the quadratic equation for C , we get the 3-sigma (99.7% confidence) estimate:

$$C_{3\sigma} = x \pm \frac{4.5}{N_s} [1 + 0.44 N_s x (1-x)]^{1/2}$$

Where N_s is sample size and x is the measured fault coverage in the sample.
 Example: A circuit with 39,096 faults has an actual fault coverage of 87.1%. The measured coverage in a random sample of 1,000 faults is 88.7%. The above formula gives an estimate of $88.7\% \pm 3\%$. CPU time for sample simulation was about 10% of that for all faults.

9/18/2009

24

Summary

- Fault simulator is an essential tool for test development.
- Concurrent fault simulation algorithm offers the best choice.
- For restricted class of circuits (combinational and synchronous sequential with only Boolean primitives), differential algorithm can provide better speed and memory efficiency (Section 5.5.6.)
- For large circuits, the accuracy of random fault sampling only depends on the sample size (1,000 to 2,000 faults) and not on the circuit size. The method has significant advantages in reducing CPU time and memory needs of the simulator.

9/18/2009

25