VLSI Design Verification and Testing

Fault Simulation

Mohammad Tehranipoor
Electrical and Computer Engineering
University of Connecticut

Overview

- Problem and motivation
- Fault simulation algorithms
  - Serial
  - Parallel
  - Deductive
  - Concurrent
  - Other algorithms
- Random Fault Sampling
- Summary

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

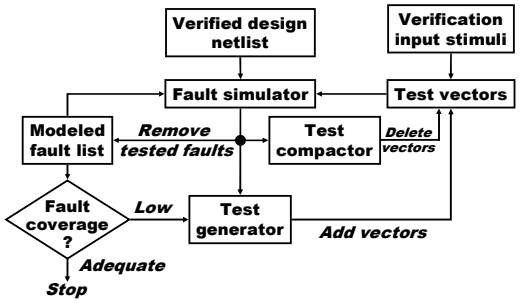
Usages of Fault Simulators

- Test grading – as explained before
- Test Generation
- Fault diagnosis
- Design for test (DFT) – identification of points that may help improve test quality known as "Test Point Insertion"
- Fault-tolerance – identification of damage a fault can cause

Alternatives and Their Limitations

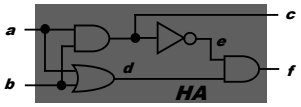
- Prototyping with fault injection capabilities
  - Costly
  - Limited fault injection capability
  - Design changes hard to implement
  - Long lead time
- Hardware emulators
  - Costly
  - Require special hardware

Fault Simulator in a VLSI Design Process

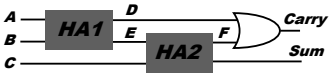


Example

Half-Adder

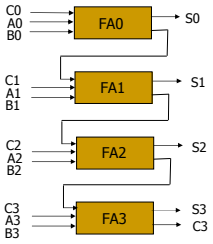


Full-Adder



7

Example



8

Fault Simulation Results

4-bit FA: 36 logic gates, 9 PIs, 5POs, 186 single stuck-at faults.

Vector Number	186 Uncollapsed Faults		114 Collapsed Faults	
	Detected	Coverage	Detected	Coverage
1	61	33%	37	32%
2	113	61%	65	57%
3	125	67%	77	68%
4	143	77%	89	78%
5	162	87%	102	89%
6	186	100%	114	100%
7	186	100%	114	100%
8	186	100%	114	100%

9

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

10

Fault Simulation Scenario (cont.)

- 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

11

Fault Simulation Algorithms

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

12

Serial Algorithm

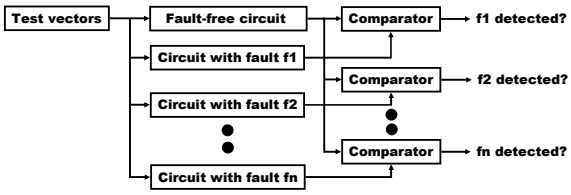
- Algorithm: Simulate fault-free circuit and save responses. Repeat following steps for each fault in the fault list:
  - Modify netlist by injecting one fault
  - Simulate modified netlist, vector by vector, comparing responses with saved responses
  - If response differs, report fault detection and 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

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 compute the value based on the signal value and the value in the additional variable

Serial Algorithm (Cont.)

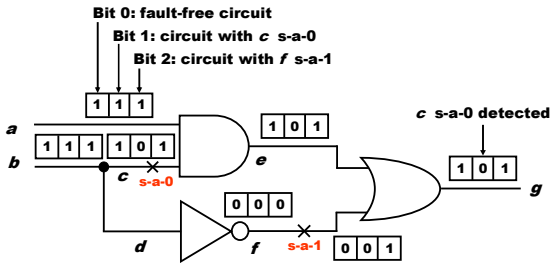
- Disadvantage: Much repeated computation; CPU time prohibitive for VLSI circuits
- Alternative: Simulate many faults together



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 ~ w-1
- Not suitable for circuits with timing-critical and non-Boolean logic

Parallel Fault Sim. Example



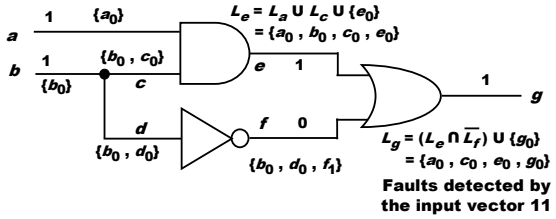
Input ab = 11

Deductive Fault Simulation

- One-pass simulation, one fault-free circuit is simulated.
- 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

Deductive Fault Sim. Example

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

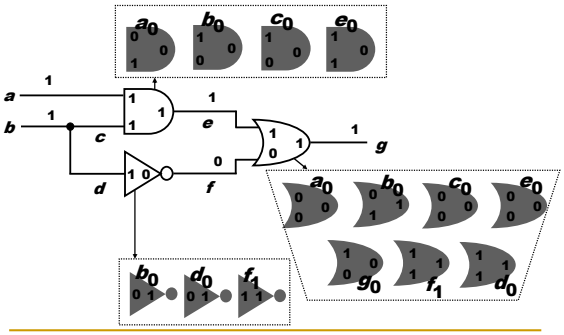


See Table 5.6 in Page 110 (text book) for fault list propagation rules

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.

Conc. Fault Sim. Example



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

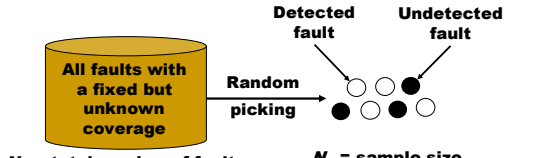
Fault Sampling

- 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.

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

Random Sampling Model



N_p = total number of faults (population size)

C = fault coverage (unknown)

N_s = sample size

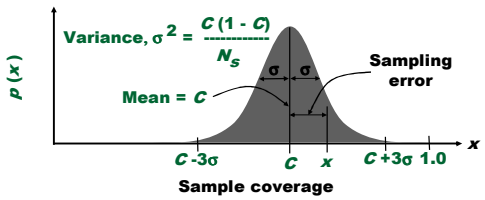
$N_s << N_p$

c = sample coverage (a random variable)

25

Probability Density of Sample Coverage, c

$$p(x) = \text{Prob}(x < c < x + dx) = \frac{1}{\sigma (2\pi)^{1/2}} e^{-\frac{(x-C)^2}{2\sigma^2}}$$



Gaussian Probability density of sample fault coverage c

26

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% ± 3%. CPU time for sample simulation was about 10% of that for all faults.

27