

Fault Simulation

Virendra Singh

Associate Professor

Computer Architecture and Dependable Systems Lab

Dept. of Electrical Engineering

Indian Institute of Technology Bombay

viren@ee.iitb.ac.in



EE 709: Testing & Verification of VLSI Circuits

Lecture – 8 (Jan 19, 2012)

Fault Model - Summary

- Fault models are **analyzable approximations** of defects and are essential for a test methodology.
- For digital logic **single stuck-at fault model offers best advantage** of tools and experience.
- Many other faults (bridging, stuck-open and multiple stuck-at) are largely covered by stuck-at fault tests.
- Stuck-short and delay faults and technology-dependent faults require special tests.
- Memory and analog circuits need other specialized fault models and tests.

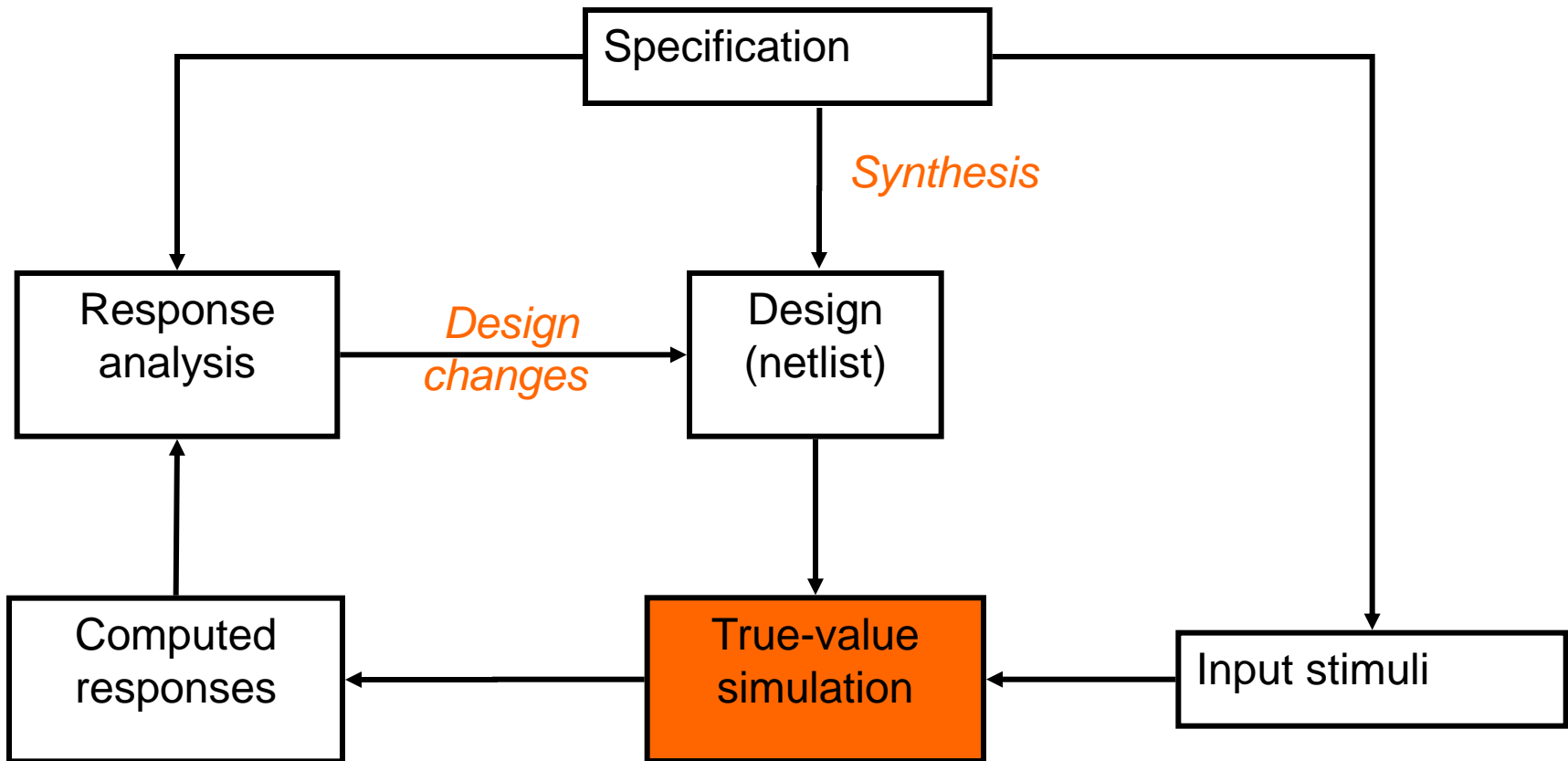
Fault Simulation



Simulation Defined

- Definition: Simulation refers to modeling of a design, its function and performance.
- A software simulator is a computer program; an emulator is a hardware simulator.
- Simulation is used for design verification:
 - Validate assumptions
 - Verify logic
 - Verify performance (timing)
- Types of simulation:
 - Logic or switch level
 - Timing
 - Circuit
 - Fault

Simulation for Verification



Modeling Levels

Modeling level	Circuit description	Signal values	Timing	Application
Function, behavior, RTL	Programming language-like HDL	0, 1	Clock boundary	Architectural and functional verification
Logic	Connectivity of Boolean gates, flip-flops and transistors	0, 1, X and Z	Zero-delay unit-delay, multiple-delay	Logic verification and test
Switch	Transistor size and connectivity, node capacitances	0, 1 and X	Zero-delay	Logic verification
Timing	Transistor technology data, connectivity, node capacitances	Analog voltage	Fine-grain timing	Timing verification
Circuit	Tech. Data, active/passive component connectivity	Analog voltage, current	Continuous time	Digital timing and analog circuit verification

True-Value Simulation Algorithms

❖ Compiled-code simulation

- Applicable to zero-delay combinational logic
- Also used for cycle-accurate synchronous sequential circuits for logic verification
- Efficient for highly active circuits, but inefficient for low-activity circuits
- High-level (e.g., C language) models can be used

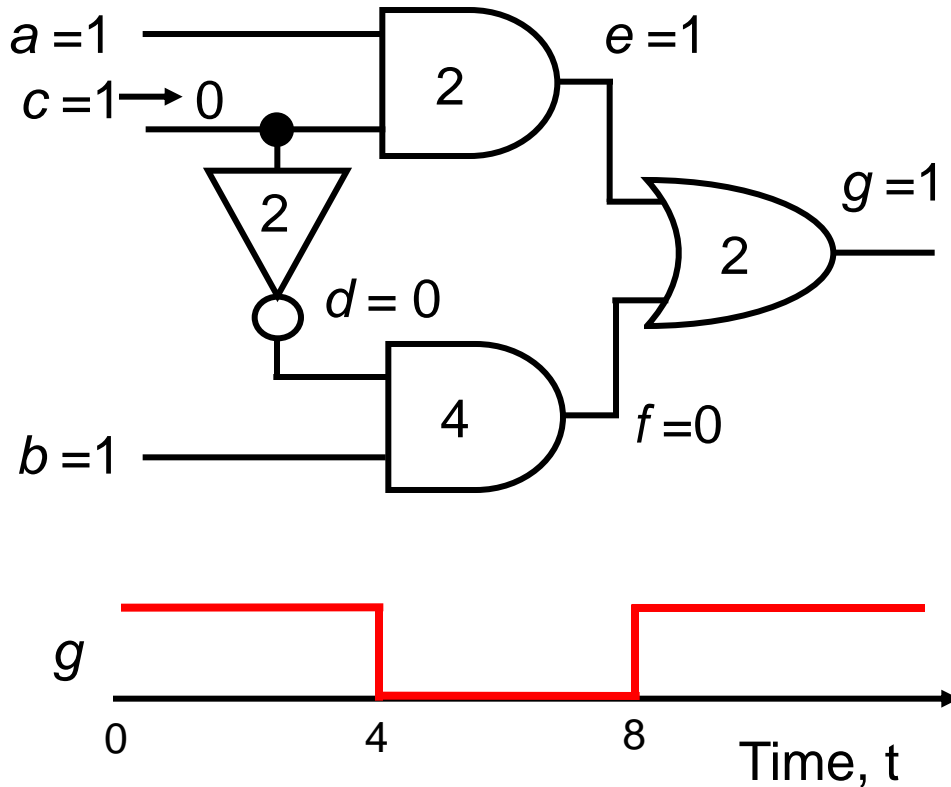
❖ Event-driven simulation

- Only gates or modules with input events are evaluated (*event means a signal change*)
- Delays can be accurately simulated for timing verification
- Efficient for low-activity circuits
- Can be extended for fault simulation

Compiled-Code Algorithm

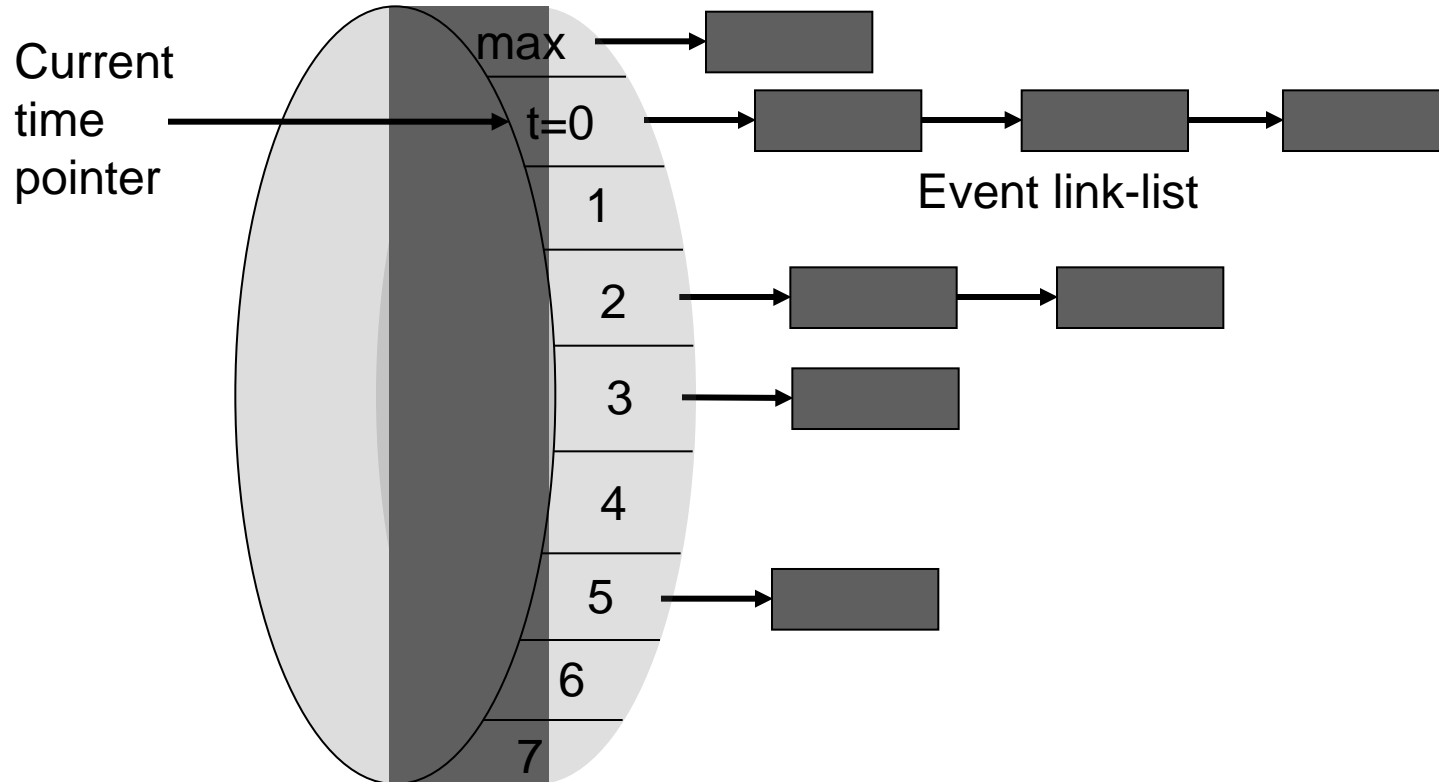
- ❖ Step 1: Levelize combinational logic and encode in a compilable programming language
- ❖ Step 2: Initialize internal state variables (flip-flops)
- ❖ Step 3: For each input vector
 - Set primary input variables
 - Repeat (until steady-state or max. iterations)
 - Execute compiled code
 - Report or save computed variables

Event-Driven Algorithm



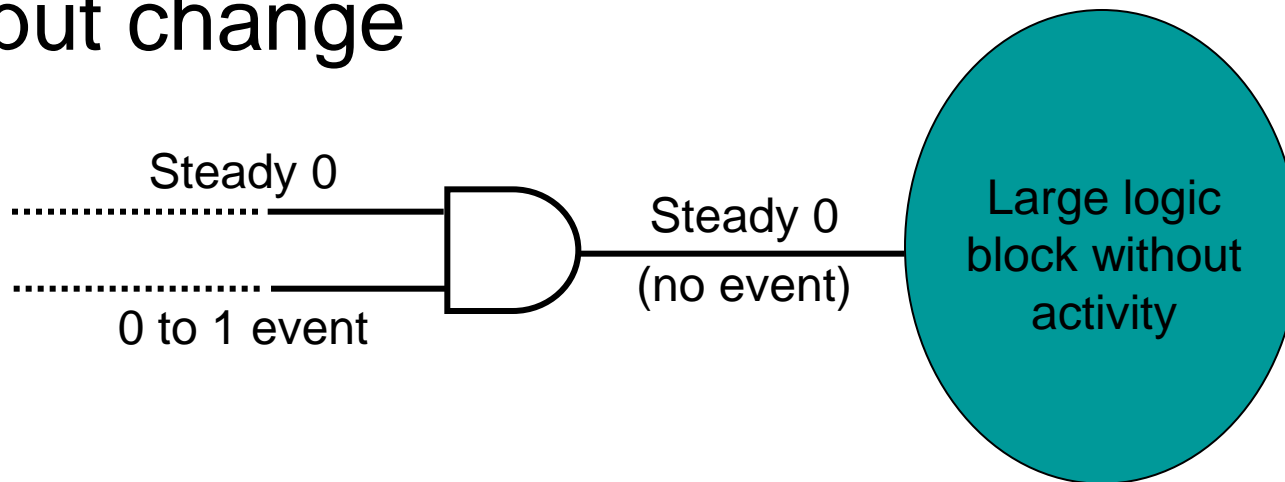
	Scheduled events	Activity list
Time stack	$t = 0$	$c = 0$ d, e
	1	
	2	$d = 1, e = 0$ f, g
	3	
	4	$g = 0$
	5	
	6	$f = 1$ g
	7	
8	$g = 1$	

Time Wheel (Circular Stack)



Efficiency of Event-driven Simulator

- Simulates events (value changes) only
- Speed up over compiled-code can be ten times or more; in large logic circuits about 0.1 to 10% gates become active for an input change



Problem and Motivation

● Fault simulation Problem: Given

- A circuit
- A sequence of test vectors
- A fault model

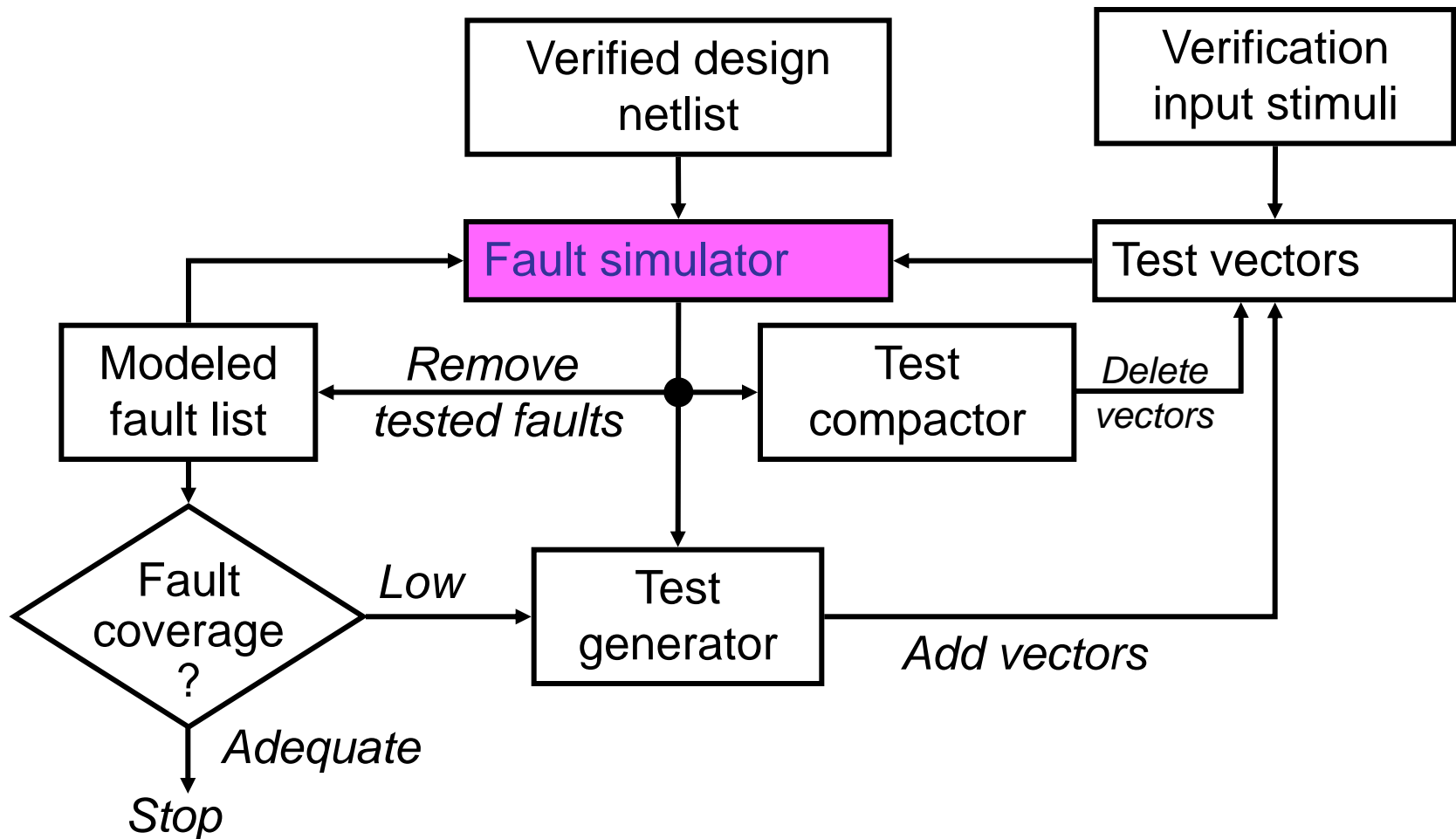
– Determine

- 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

Fault simulator in a VLSI Design Process



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

Fault Simulation Scenario

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

Fault Simulation Algorithms

❖ **Serial**

❖ **Parallel**

❖ **Deductive**

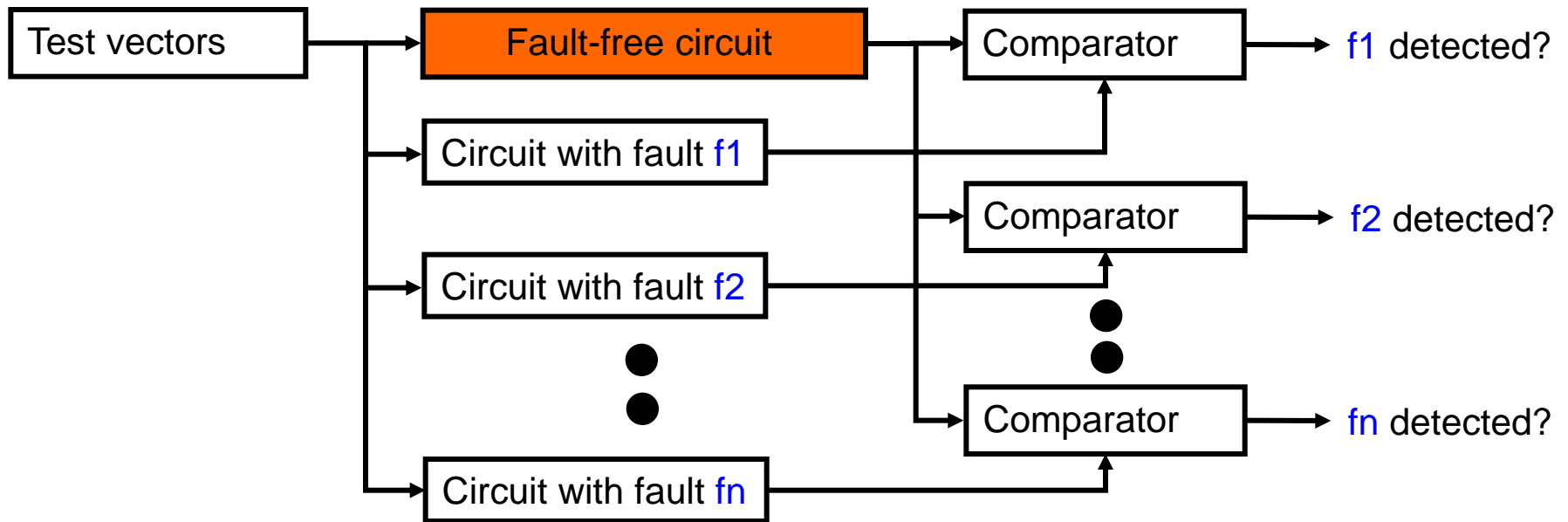
❖ **Concurrent**

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

Serial Algorithm

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



Thank You

