

CSE-308: Digital System Design

Lecture 8

TESTING – Part I

Why Testing?

- To determine the presence of fault(s), not the absence of fault(s), in a given circuit.
 - No amount of testing can guarantee that a circuit (chip, board or system) is *fault-free*.
 - We carry out testing to increase our confidence in proper working of the circuit.
- Verification is an alternative to testing, used to verify the correctness of a design.
 - Simulation-based approach.
 - Formal methods.

Verification


v/s

Testing

- Verifies correctness of design.
- Performed by simulation, h/w emulation, or formal methods.
- Performed once prior to manufacturing.
- Responsible for quality of design.

- Verifies correctness of manufactured h/w.
- Two-part process:
 1. Test generation
 2. Test application
- Test application performed on every manufactured device.
- Responsible for quality of devices.

Levels of Testing

- Testing can be carried out at the level of
 - Chip ✓
 - Board ✓
 - System ✓
- Cost :: Rule of 10
 - It costs 10 times more to test a device as we move to the next higher level in the product manufacturing process.

Level of Testing (contd.)

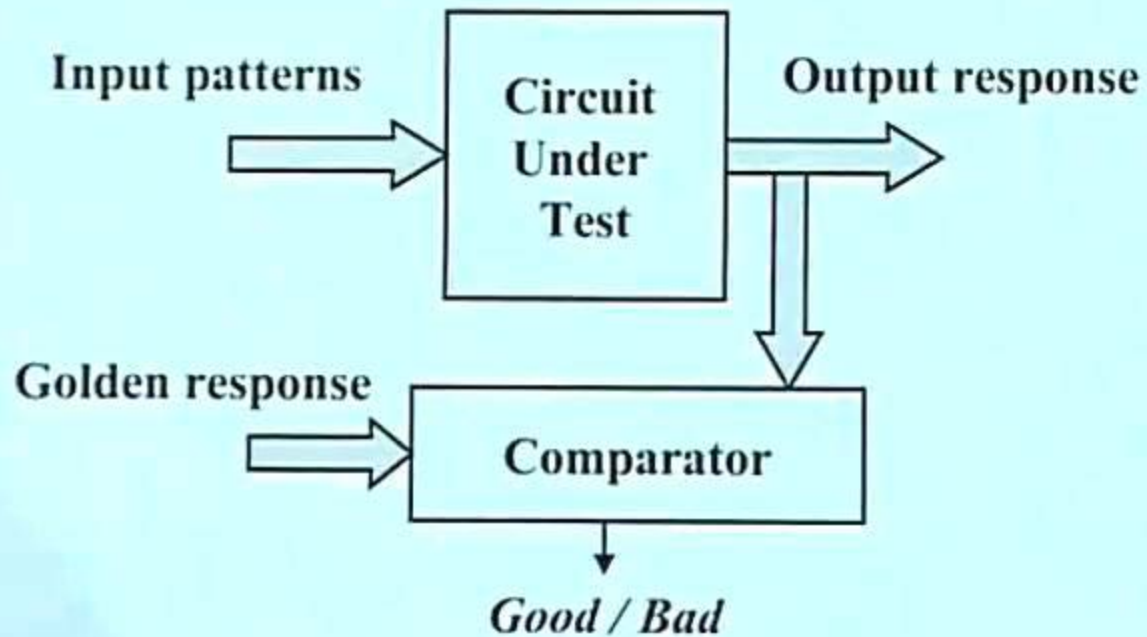
- Other ways to define levels:
 - Important to develop correct *fault models* and *simulation models*.
 - Transistor
 - Gate
 - RTL
 - Functional / Behavioral

Costs of Testing

- Design for Testability (DFT)
 - Chip area overhead ✓
 - Yield reduction ✓
 - Performance overhead ✓
- Software processes of test
 - Test generation ✓
 - Fault simulation ✓
- Manufacturing test
 - Automatic Test Equipment (ATE) cost
 - Test center operational cost

Basic Testing Principle

© CET
I.I.T. KGP



Fault Modeling

Why Fault Models?

- Actual number of physical defects in a circuit / chip are too many.
 - Not possible to consider individually.
- Some logical fault models are considered in practice.
 - Drastically reduces the number of faults to be handled.
 - Covers most of the possible physical failures.

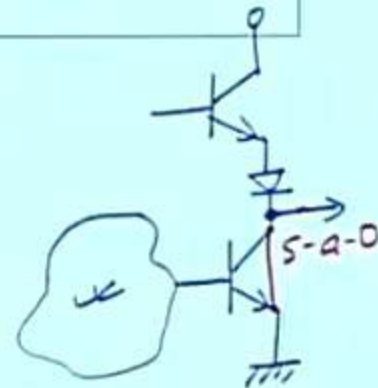
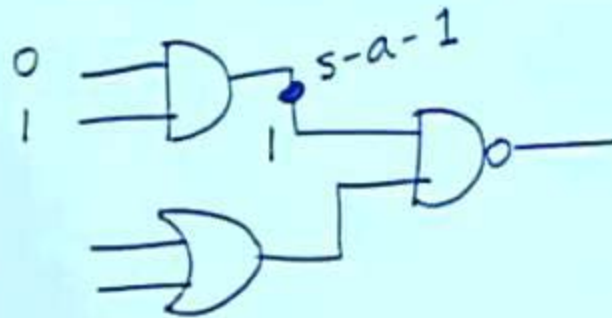
Common Fault Models

- Stuck-at faults ✓
 - single, multiple
- Transistor faults ✓
 - open, short
- Memory faults
 - coupling, pattern-sensitive }
- PLA faults
 - stuck-at, cross-point, bridging }
- Delay faults
 - transition, path }
- Functional faults ✓

Stuck-at Faults

© CET
I.I.T. KGP

- Some line(s) in the circuit are permanently stuck at logic 0 or logic 1.
- Two types:
 - Single stuck-at faults
 - Multiple stuck-at faults



- Fault Equivalence and Fault Dominance
 - Reduces the number of single stuck-at faults to be considered.
- Why single stuck-at faults?
 - Simpler to handle computationally.
 - Reasonably good fault coverage.
 - A test set for detecting single stuck-at faults detects a large percentage of multiple stuck-at faults as well.

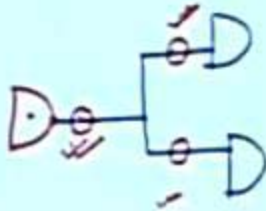
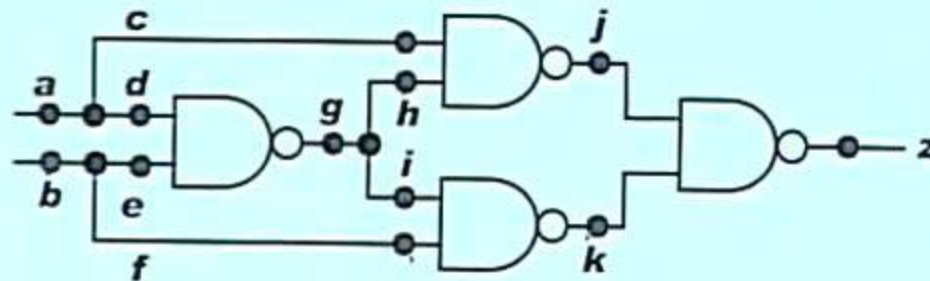
99.5%

Single Stuck-at Fault

- Three properties define a single stuck-at fault.
 - Only one line is faulty at a time.
 - The faulty line is permanently set to 0 or 1.
 - Not of intermittent nature
 - The fault can be at an input or output of a gate / module.
- For a circuit with k lines, the total number of single stuck-at faults possible is $2k$.
- Most widely used fault model in the industry.

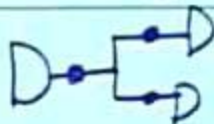


Example: XOR circuit has 12 fault sites
and 24 single stuck-at faults



Fault Equivalence

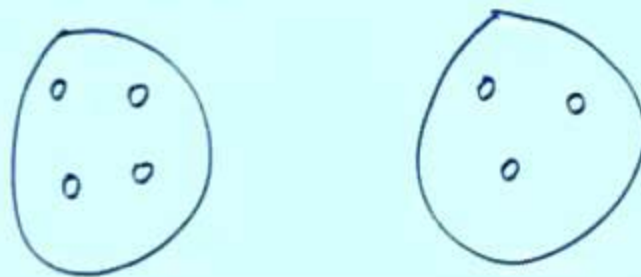
- Number of fault sites in a gate-level circuit
= #PI + #gates + # (fanout branches)
- Fault equivalence:
 - Two faults f_1 and f_2 are *equivalent* if all tests that detect f_1 also detect f_2 .
 - Example:
 - An input line s-a-0 and output line s-a-0 in an AND gate.
- If faults f_1 and f_2 are equivalent then the corresponding faulty functions are identical.



Contd.

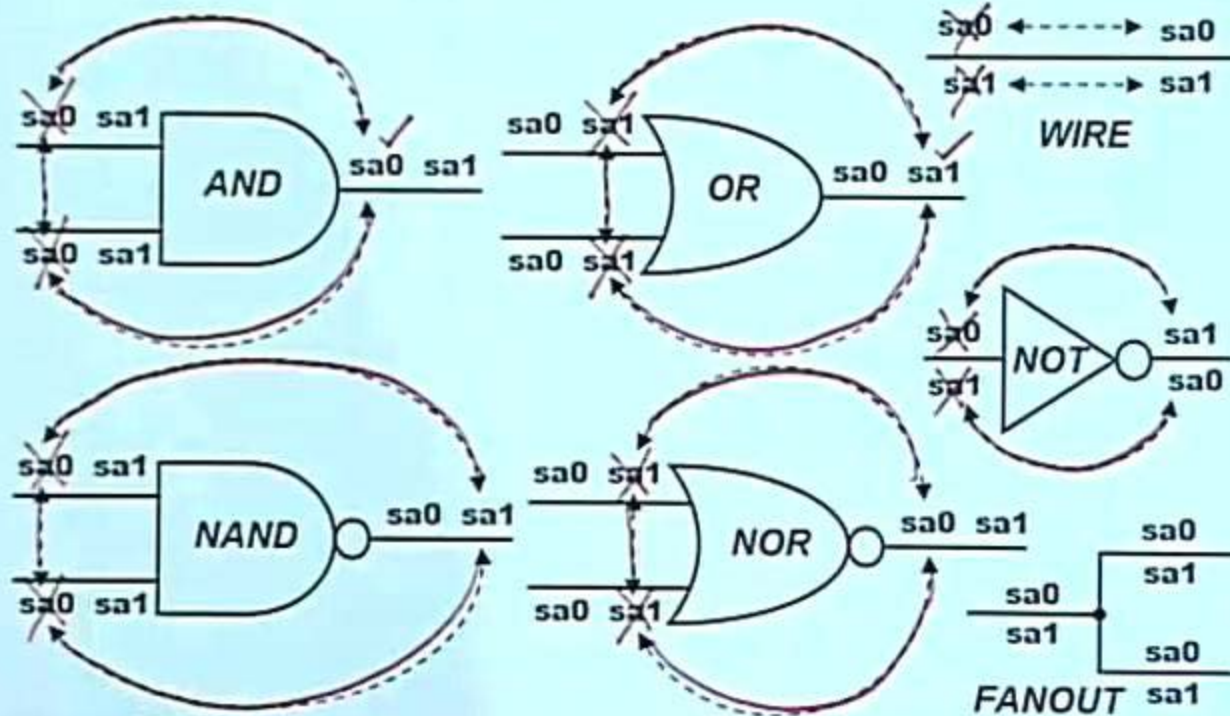
- **Fault collapsing:**

- All single faults of a logic circuit can be divided into disjoint equivalence subsets.
- All faults in a subset are mutually equivalent.
- A collapsed fault set contains one fault from each equivalence subset.



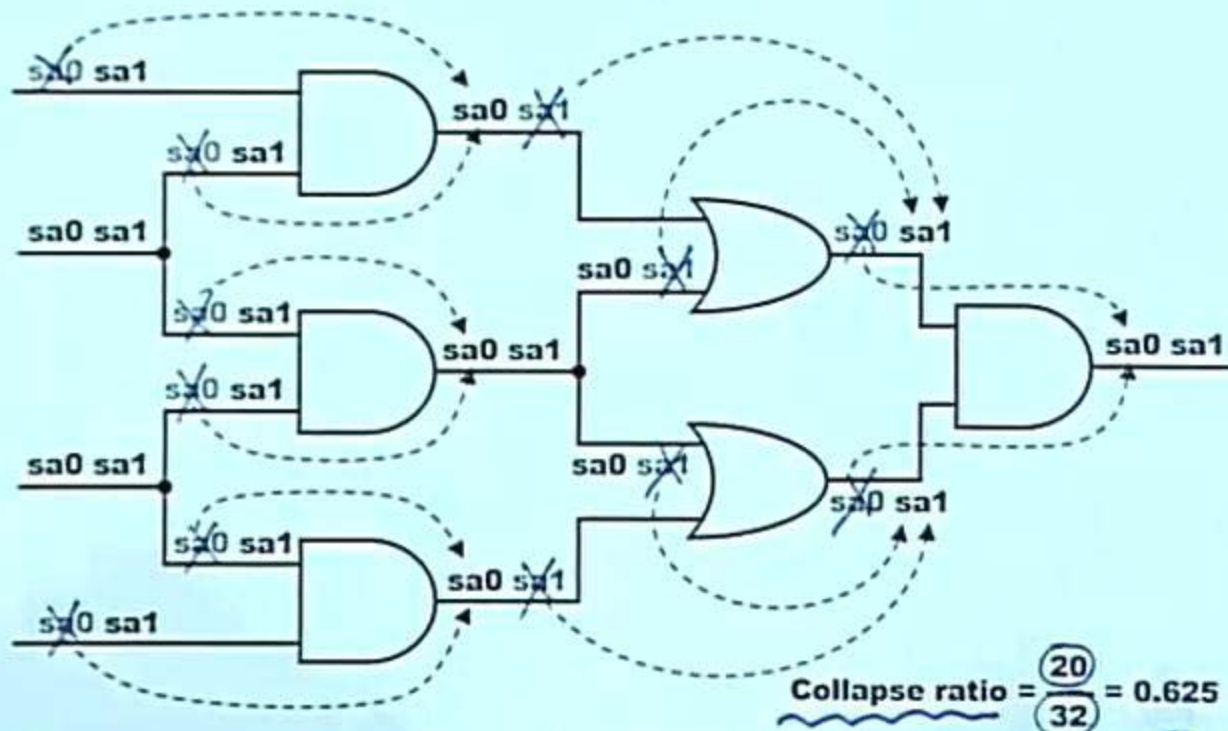
Equivalence Rules

© CET
I.I.T. KGP



Equivalence Example

©CET
I.I.T. KGP

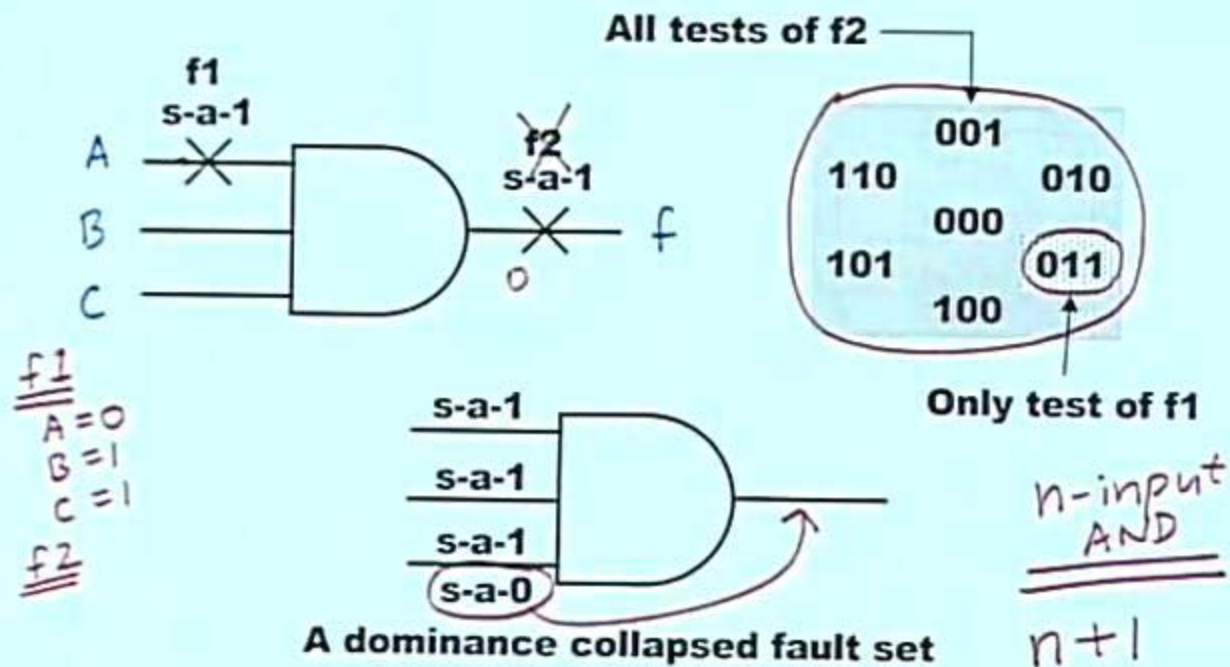


Fault Dominance

- If all tests for some fault f1 detect another fault f2, then f2 is said to *dominate* f1.
- Dominance fault collapsing:
 - If fault f2 dominates f1, then f2 is removed from the fault list.
- In a tree circuit the primary input faults form a dominance collapsed fault set.
- If two faults dominate each other then they are equivalent.

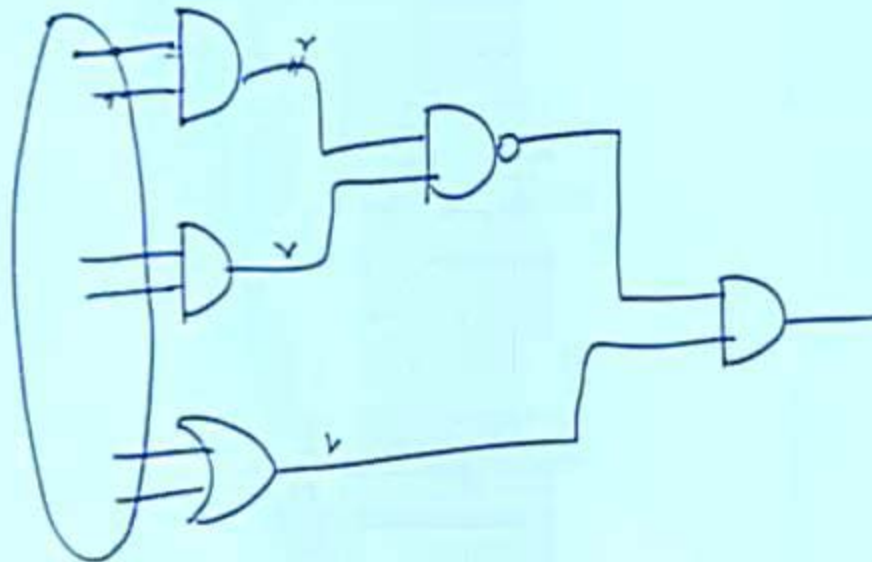
Dominance Example

CCET
I.I.T. KGP



Fault Dominance

- If all tests for some fault f1 detect another fault f2, then f2 is said to *dominate* f1.
- Dominance fault collapsing:
 - If fault f2 dominates f1, then f2 is removed from the fault list.
- In a tree circuit the primary input faults form a dominance collapsed fault set.
- If two faults dominate each other then they are equivalent.

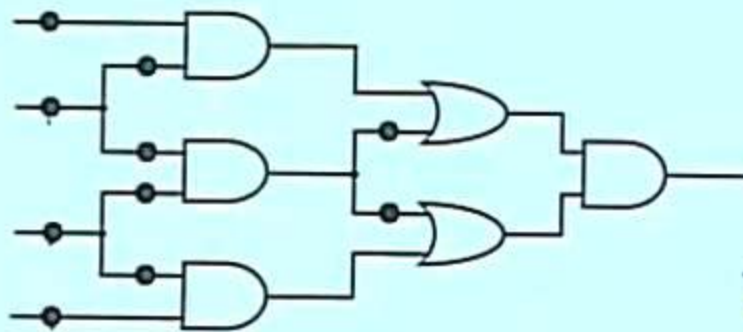


Checkpoints

- Primary inputs and fanout branches of a combinational circuit are called checkpoints.
- Checkpoint theorem:
 - A test set that detects all single (multiple) stuck-at faults on all checkpoints of a combinational circuit, also detects all single (multiple) stuck-at faults in that circuit.

Total fault sites = 16 //

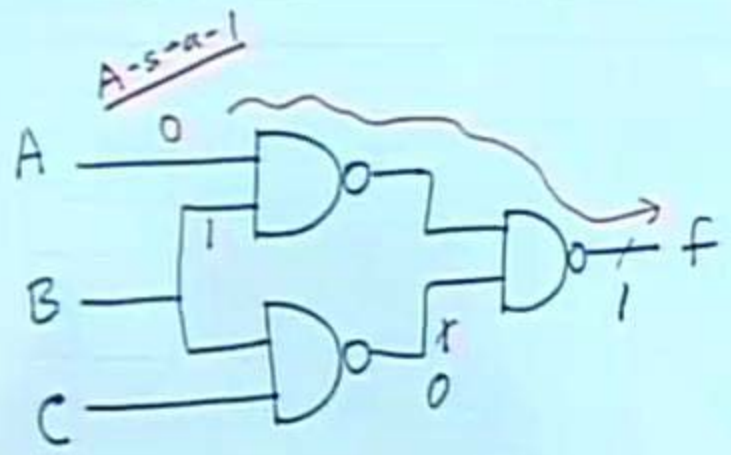
Checkpoints = 10 //



32
↓
20

Multiple Stuck-at Faults

- Any set of lines is permanently stuck-at some combination of (0,1) values.
- The total number of single and multiple stuck-at faults in a circuit with k single fault sites is $3^k - 1$. $\cdot 2^k$
- A single fault test can fail to detect the target fault if another fault is also present.
– called fault masking, which is rare.
- Statistically, single fault tests cover a very large number of multiple faults.



A/1 :
A = 0
B = 1
C = 0

A/1, C/0

\Rightarrow undetectable
fault