Lecture 9:
Fault List Reduction
Test Compaction

Instructor: M. Tahoori
Motivation

- When testing for single stuck-at faults it is not necessary to consider each fault explicitly
  - detection of a subset of faults guarantees that all faults will be detected
- *fault collapsing*
  - process of reducing the number of faults that must be considered explicitly during test pattern generation or fault simulation
    - Reduces fault simulation time
    - Reduces test pattern generation time
- Two types
  - Fault equivalence
  - Fault dominance

Fault Equivalence

- Number of fault sites in a Boolean gate circuit
  - \( = \#PI + \#gates + \# \text{ (fanout branches)} \).
- Fault equivalence:
  - Two faults \( f_1 \) and \( f_2 \) are equivalent if and only if any test pattern that detects \( f_1 \) also detects \( f_2 \) and vice-versa.
  - If faults \( f_1 \) and \( f_2 \) are equivalent then the corresponding faulty functions are identical.
- Equivalence Fault collapsing:
  - All single faults of a logic circuit can be divided into disjoint equivalence subsets, where all faults in a subset are mutually equivalent.
  - A collapsed fault set contains one fault from each equivalence subset.
Equivalence Rules

Equivalence Example

Faults in red removed by equivalence collapsing

Collapse ratio = \( \frac{20}{32} = 0.625 \)
Fault Dominance

- One fault $F_1$ is said to dominate a fault $F_2$
  - if and only if any test vector that detects $F_2$ also detects $F_1$
- Dominance fault collapsing:
  - If fault $F_1$ dominates $F_2$, then $F_1$ is removed from the fault list.

Dominance Example

A dominance collapsed fault set
Fault Dominance

- If F1 dominates F2 and F2 dominates F1, then F1 and F2 are equivalent.
- Suppose F1 dominates F2, a set of tests generated without considering F1, but which detects F2, will automatically detect F1.
  - However, it is possible to have a test set that detects F1 but does not detect F2.
  - In particular, it is possible for F1 to be detectable even if F2 is undetectable.
- Fault dominance is a transitive relation:
  - If F1 dominates F2, and F2 dominates F3, then F1 dominates F3.

Fault Dominance

- The two stuck-at faults on the output of an elementary gate dominate the two stuck-at faults on any one of the inputs to the gate.
  - One of the output faults is equivalent to one of the input faults
  - the other output fault dominates the other input fault but is not equivalent to it.
  - An n-input elementary gate has n+1 non-dominating faults in its minimal length dominance-reduced fault list.
- When dominance fault collapsing is used, it is sufficient to consider only the input faults of Boolean gates.
- In a fanout-free circuit, the set of all single-stuck faults on the circuit primary inputs is a dominance-reduced fault list for the circuit.
  - Thus a test that detects all the single-stuck primary input faults, detects all the internal single-stuck faults as well.
Dominance-Reduced Fault List

A reduced dominance-reduced fault list for a fanout-free combinational circuit consists of:

- (a) stuck-at-1 faults on all circuit inputs connected to inputs of AND or NAND gates,
- (b) stuck-at-0 faults on all circuit inputs connected to inputs of OR or NOR gates,
- (c) stuck-at-1 faults on all outputs of OR or NAND gates that have as gate inputs only circuit inputs, and
- (d) stuck-at-0 faults on all outputs of AND or NOR gates that have as gate inputs only circuit inputs.

Classify gates in a fanout-free circuit

- **Input gates**
  - All gate inputs are primary inputs
- **Internal gates**
  - All gate inputs are outputs of other gates
- **Mixed gates**
  - At least one input is a circuit input and at least one input comes from the output of another gate

A dominance-reduced fault list for a fanout-free combinational circuit consists of:

- (1) non-dominating stuck faults on all inputs and outputs of input gates, and
- (2) non-dominating stuck faults on all the inputs of mixed gates that are circuit inputs

For a fanout-free circuit with n primary inputs and h input gates

- there is a list of n+h faults whose detection guarantees detection of all single-stuck faults in the circuit
Example

- Fanout-free circuit with set of dominance-reduced faults
- Stuck-at-0 fault on the output of gate f can be dropped
  - It is equivalent to a stuck-at-0 fault on the output of gate h
    - Which dominates the stuck-at-0 fault on the lower input of gate g

Dominance-Reduced Fault List

- A Dominance-reduced Fault List for a General Combinational Circuit Can be Obtained by:
  - Partitioning the Circuit into Fanout-free Sub-circuits,
  - Forming the dominance-reduced fault list for each fanout-free Sub-circuit
Example

- There are no general dominance relations between fanout stems and branches.
- It is possible to drop the stuck-at-0 fault on lead M.
  - A test that detects the stuck-at-0 fault on lead P.
    - Must detect the stuck-at-0 on M.

Dominance-Reduced Fault List

- Fanout stem stuck faults
  - Dominate fanout branch stuck faults if
    - Fanout branches do not reconverge (multi-output circuit) or
    - Fanout branches do reconverge with same inversion parity.
  - May not dominate fanout branch stuck faults if
    - Fanout branches do reconverge with odd inversion parity.

- The computation required to determine whether a fanout point reconverges
  - Can involve many gates.
  - Therefore may or may not be cost effective.
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.

![Diagram](image)

- Total fault sites = 16
- Checkpoints (●) = 10

Checkpoints

- If any of the checkpoint faults are undetectable
  - it is impossible to derive a test set that detects all of the checkpoint faults
  - It is possible to have a detectable fault that dominates an undetectable checkpoint fault
  - it is necessary to attempt to generate tests for the faults that dominate the undetectable checkpoint
- A test set for a combinational circuit that detects all detectable (reduced) checkpoint faults and all faults that dominate undetectable checkpoint faults detects all detectable faults in the circuit
Example

Test Compaction
Motivation

- Test pattern for a stuck fault contains a lot of don’t care bits
  - The maximum number of specified bits in the test pattern for a single stuck-at fault at a circuit node is less than or equal to the number of inputs in the fan-in cones of the outputs reachable from that node
- On the tester, the don’t care bits in a test pattern must be filled with 1s and 0s
- Definition
  - The technique for reducing the number of test patterns without sacrificing fault coverage is called test pattern compaction.
- Types of compaction
  - Static compaction,
  - Dynamic compaction
  - By fault simulation

Static Compaction

- Primary Input Test Values are Usually Partially Specified
- Combine Tests to Reduce Test Length
- Example
  - \( t_1 = 0 \ 1 \ X, t_2 = 0 \ X \ 1, t_3 = 0 \ X \ 0, t_4 = X \ 0 \ 1, \)
  - Length-3 Test Set
    - \( t_{12} = 0 \ 1 \ 1, t_3 = 0 \ X \ 0, t_4 = X \ 0 \ 1, \)
  - Length-2 Test Set
    - \( t_{13} = 0 \ 1 \ 0, t_{24} = 0 \ 0 \ 1, \)
  - Minimum-Length Test Computationally Expensive to Find
Static Compaction

- Two Vectors with \{0,1,X\} Elements are *Compatible* iff
  - No Position is 0 in One Vector and 1 in the Other Vector
- Example
  - \(t_1 = \text{ABCD} = 10XX\) and \(t_2 = \text{ABCD} = 0XX1\), then \(t_1\) and \(t_2\) are not compatible.
  - \(t_1 = \text{ABCD} = 10XX\) and \(t_2 = X0X1\), then \(t_1\) and \(t_2\) are compatible
- *Intersection* of Two Compatible Vectors has X Elements
  - Only where Both Vectors are X, Else
  - Value is Same as Specified in One of the Vectors

Static Compaction

- Given a set of \(n\) test patterns \(\{t_1, t_2, \ldots, t_n\}\)
  - a static compaction technique partitions the set into disjoint subsets
    - such all test patterns in any subset are compatible with each other
- Static Test Set Compression (Test Set Compaction)
  - Compatible Test Set Vectors Replaced by their Intersection
Dynamic Compaction

- Unlike static compaction, during dynamic compaction several faults are targeted to be detected by a single test pattern.
- Dynamic test set compression (compaction)
  - After each test is generated, select another fault.
  - Try to generate a test for it
    - Don't change input values in test already generated.

Issues

- Which faults must be targeted by a test pattern?
- How long to continue targeting new faults by the same test pattern?

Test Compaction Using Fault Simulation

- Reverse order simulation technique
  - Perform fault simulation using the entire fault list in reverse order of the test patterns generated
    - The test pattern generated last is fault simulated first and so on
  - Reverse order simulation doesn’t necessarily guarantee reduction of the number of test patterns
Test Compaction Using Fault Simulation

Example

- Suppose that fault f1 dominates fault f2, and both stay in the fault list
  - Since ATPG tools typically don’t implement fault dominance
- Suppose that the ATPG tool first generates a test pattern t1 for f1
  - The pattern t1 can be such that it detects f1 but doesn’t detect f2.
- During fault simulation after the generation of test pattern t1, fault f1 and possibly some other faults will be dropped from list.
- However, fault f2 will not be dropped from the fault list.
- Later during ATPG a test pattern t2 is generated which detects f2.
- During reverse order simulation, when t2 is fault simulated both f1 and f2 will be detected.