Lecture Slides for ATPG: Reducing Test Generation Effort, the Check-Point Theorems and Issues with Multiple Stuck-Faults
Fault Collapsing

• We have to derive s/0, s/1 tests for the entire circuit
  • Each net, including fanout stems and branches
• Primary goal is test detection
• Deriving a test for a fault at each net is too much work
• Thankfully, we can exploit the concept of fault equivalence and fault dominance to reduce the test generation effort
• Analyze fault equivalence and dominance “structurally”
• Equivalent faults cannot be distinguished, so test only one of these from the equivalence class
• Reduce the number of tests required to test the whole circuit
Structural Fault Equivalence

- For an AND gate: \(a/0 = b/0 = c/0\) (equivalent faults)
  - Same test vector, and same fault effect
  - Test only one of these
- NAND gate: \(a/0 = b/0 = c/1\)
- OR gate: \(a/1 = b/1 = c/1\)
- XOR gate:
  - \(a/0\) implies output \(c = b\); \(a/1\) implies \(c = b'\)
  - \(b/0\) implies \(c = a\); \(b/1\) implies \(c = a'\)
  - \(c/0\) implies \(c = 0\); \(c/1\) implies \(c = 1\)
  - No fault equivalence at all
Fault Dominance

- Let $T_g$ be the set of all tests that detect fault $g$
- Fault $f$ dominates fault $g$ if $Z_f(t) = Z_g(t)$ $\forall t \in T_g$
- Clearly $T_f \supseteq T_g$
- If the goal is fault detection (and not fault distinguishing/diagnosis), then $T_f$ is not needed, $T_g$ suffices to detect fault $f$
Fault Dominance for NAND/NOR Gates

- Let $T_g$ be the set of all tests that detect fault $g$
- Fault $f$ dominates fault $g$ if $Z_f(t) = Z_g(t) \ \forall t \in T_g$
- Clearly $T_f \supseteq T_g$.
- If the goal is fault detection (and not fault distinguishing/diagnosis), then $T_f$ is not needed, $T_g$ suffices to detect fault $f$
Fault Collapsing on a Fanout-Free Circuit

- If a test set $T$ detects all $s/0$ and $s/1$ faults on the primary inputs (PIs) in a fanout-free circuit with AND/OR/INV gates, the $T$ detects all $s/0$ and $s/1$ faults in the circuit!
- Gate output fault is either equivalent to gate input faults, or it dominates gate input faults.
- Tests for gate output faults need not be derived!
But What about Fanouts? The Check Point Theorem!

- Check points of a circuit = PIs + fanout branches!
  - Fanout stems = gate outputs or PIs
  - If T detects all checkpoint faults, T detects all single-stuck faults in the circuit

\[ \text{But } e/o = a/o \neq b/o \]
\[ v/e \neq b/i \]
\[ v/e \neq b/i \]
\[ i/o = q/o \checkmark \]
\[ i/i > q/i \checkmark \]
\[ \times \checkmark \text{ not necessary. } \]
Multiple-Stuck Faults

- Case 1: Let a SSF be undetectable. This implies redundancy. A multi-stuck-fault (MSF) under redundancy becomes detectable!
Multiple-Stuck Faults

- Case 2: Let a SSF $f$ be detectable, and another SSF $g$ be undetectable. Then the MSF $(f, g)$ becomes undetectable.
- This is called “test invalidation” in the presence of redundancies. That’s why we prefer to do SSF tests, under a frequent testing strategy.
ATPG Fault Coverage

- Fault coverage = \( F_{cov} = \frac{\# \text{ of detectable faults}}{\# \text{ of total faults}} \)
- Fault efficiency of an ATPG tool
  \[ \frac{\text{total detectable faults} - \text{aborted faults}}{\text{total faults}} \]
- In modern ATPG tools, fault coverage is very high 95+%.
- In the early days, ATPG algorithms D-algorithm, PODEM, FAN. Now a days, SAT solver based ATPG is very efficient.
  - Miter model: Fault free (spec), faulty (with a stuck-line) implementation.