VLSI / SOC Testing

Lecture 24

1. Diagnostic test generation
   - Given a fault pair, generate a test that can detect one but not the other
   - Define: two faults α and β are distinguishable if $\exists$ a test t such that the output of fault $\alpha \neq$ the output of fault β by test t
   - Indistinguishability can be defined conversely. If two faults are indistinguishable, they are also functionally equivalent
   - To improve diagnostic test generation, it would be nice to determine if 2 faults are distinguishable quickly in advance
2. Functional equivalence of two faults
   - Recall that a dominator gate of gate g is a gate through which all paths from g to any PO must pass
   - A *common dominator gate* for gates g_1 and g_2 is one that both pass
   - Common dominator cone: starting from the common dominator gate and backtrace in the circuit, including g_1 and g_2, together with all gates that are sufficient to completely determine the functions of the common dominator gate

   Example 1:
3. Properties of dominator cones
   - If logic functions at the common dominator gate for faults α and β are identical when expressed in terms of the inputs of the common dominator cone, then faults α and β are functionally equivalent

- Even if the logic functions expressed at the inputs of cone are not identical, α may still be functionally equivalent to β if the inputs at the cone that distinguishes the faults cannot be justified $\longmapsto$ if α and β are different for tests $t_1,\ t_2, ...,\ t_m$ at the cone inputs, and none of $t_1,\ t_2, ...,\ t_m$ is justifiable from the PIs, then α and β are functionally equivalent
- Note that the PIs responsible for propagating the fault-effect from common dominator gate to a PO are not included in the dominator cone, since they are not needed to define the common dominator gate

4. Use of redundancy information

   - if faults α and β produce same fault-effect at the common dominator gate output for a given test t, and fault β is known to be redundant, then test t must not be justifiable at the PIs of the circuit

   Example 2:

5. Distinguishability of faults in sequential circuits
    - a fault α in sequential circuit is present in every time-frame in the ILA model of the circuit
      $\longmapsto$ denote this fault α_k
    - two faults α and β are indistinguishable if α_k and β_k are indistinguishable for any starting state of the ILA $\longmapsto$ if two faults are indistinguishable for $k = 1$, then they are combinationally equivalent

    Example 3:

6. What if the starting state for the ILA is illegal/unreachable?
    - Only need to consider valid states for circuits C_α and C_β
      $\longmapsto$ valid states = set of all reachable states
      $\longmapsto$ valid states for C_α may not be the same for C_β
    - If either circuit is unsynchronizable, we can consider a subset of states
      $\longmapsto$ this subset may contain some unreachable states
    - Define: $\mathrm{RS}(\alpha,\ m)$ = set of states reachable when fault α is present within m cycles. $\mathrm{RS}(\alpha,\ 0)$ = all possible states
      $\longmapsto$ $\mathrm{RS}(\alpha,\ i+1) \subseteq \mathrm{RS}(\alpha,\ i)$

    Example 4:

7. Compaction of Fault Dictionaries

   - Given a circuit with f faults, o POs, and v vectors, a naive construction of the matrix-like fault dictionary would involve $f \times v \times o$ entries
   - Conventional compaction by avoiding storage of all faults or all PO values can result in loss of information
   - Is there a way to compact the dictionary without loss of info?

8. Compaction without loss of info is possible since:

   - the number of distinct fault effects generally less than 2^o
     $\longmapsto$ don't need to store all PO values in each entry, rather, store a pointer to which of the n distinct fault-effect it is
     $\longmapsto$ if $n < o$, then the savings simply by this method would be $\frac{2^o}{2^n}$
   - Further, since a distinct fault effect may be shared by many faults at various test vector positions, they can all point to the same distinct fault effect
     $\longmapsto$ more savings here

   Example 5:

9. Diagnosing Transistor Stuck-open Faults
   - Do we want to build another dictionary (or other methods) for stuck-open faults, or can we use SSF techniques?
   - Want: diagnose stuck-open faults with known stuck-at diagnosis techniques
   - Review: stuck-open fault detected by a 2-vector pair.

   Example 6:

10. Diagnosis approach
    - After identifying the failing chips, first diagnose assuming the failure due to a stuck-at defect
    - Then, based on the diagnostic info on SSF, deduce which stuck-open faults could cause this
    - Need: simply build a table to match behavior

    Example 7:

11. A defect may not match any fault model *exactly*
    - Can we come up with a technique that captures the possible locations of of the defect without any given fault model?
    - Motivation: if a defect is active for test vector t, it must affect at least one signal in its vicinity. And the affected signal must have a propagation path to a PO.

    Example 8:

12. Region-based diagnosis
    - Any defect within the region must propagate a FE to at least one output of region for the detecting vector
    - Number of regions in the order of number of gates: each gate can be the center node for a region
    - Don't enumerate all possible fault-effects at the region outputs, since there can be many
      $\longmapsto$ Simply inject don't-cares (X) at the region outputs to rule out false candidate regions
    - Can perform diagnosis hierarchically, starting from large regions down to small regions

    Example 9:

13. For candidate regions where the defect may reside, focus on gates within these regions
    - May enumerate all fault-effects if number of region outputs few

    Example 10: