VLSI / SOC Testing

Lecture 17

1. Path delay fault simulation
   - Exponential number of paths
     $\longmapsto$ can't simulate per path, needs non-enumerative method
2. Path status graph
   - Vertices: PIs, POs, and internal nodes with fanout > 1

   Example 1:

3. Simulation Procedure
   - Mark edges in PSG as covered or not
   - will need to split/merge nodes if necessary

   Example 2:

4. Problem of false paths
   - Every redundant fault corresponds to one or more untestable path
   - Some false paths do not have a corresponding redundant fault

   Example 3:

5. Identification of false paths
   - Find conflicting value combinations

6. Find unsensitizable segments

   - all paths that contain such segments will be false
   - Define: prime segment: a functional unsensitizable segment, Q, where no proper sub-segment of Q is a functional unsensitizable segment
     $\longmapsto$ prime means smallest segment that is unsensitizable
   - Algorithm:
     - Step 1: find an unsensitizable sub-path, P, from a given PI
       this is done by concatenating edges until sub-path becomes unsensitizable
     - Step 2: going backwards in P, find the shortest unsensitizable segment
     - Step 3: Substitute the last leg of P with another edge and repeat steps 1 and 2
     - Step 4: repeat for all PIs and for all subpaths

Example 4:

7. Testing for signal integrity
   - crosstalk between interconnects
   - transition on aggressor line causing a delay or glitch on another transition on victim line
   - needs: extract capacitive coupling between interconnect signals
   - ATPG: generate 2-pattern test that launches crosstalk behavior

   Example 5:

8. Functional Testing
   - Testing the circuit at a level/view other than netlist of basic gates
   - eg. FSM level, Register Transfer Level (RTL), Behavioral

9. FSM Testing
   - Given the FSM for the circuit, derive a test sequence that tests all states and transitions
   - Step 1: verify all n states exist in FSM
   - Step 2: test all transitions (verify correct transitional relationship)
   - Some terminologies:
     - Transfer sequence: takes the circuit to a desired state
     - Distinguishing sequence: a sequence that produces a unique output sequence for each starting state
       $\longmapsto$ by looking at the output sequence, we can determine the starting state prior to application of dist seq
   - Basic algorithm:

Apply an initialization sequence to a known state
Apply dist sequence to check for correct known state
for each state A in FSM
    compute a sequence from FSM to get to A
    apply dist seq to verify arriving at A
    transfer back to A
    for all transitions going out of A
        apply transition
        verify correct ending state with dist seq
        return to A (and test next transition out of A)

Example 6:

10. Effectiveness of FSM test sequence

    - if a s-a fault f is present and it is not untestable, then f must affect at least one transition in the FSM

11. Test sequences for FSM testing can be very long, if there are many states and transitions

    - Remedy: merge tests of different states together

12. FSM testing can be applied to testing regular structures

    - Key: test each cell in the regular structure exhaustively and simultaneously

    Example 7: (regular structure)

13. Define: a circuit is **C-testable** if it has an ILA structure and has a constant number of test vectors independent of the number of cells in the ILA

    Example 8:

14. The test vectors for regular structure can be obtained from the state diagram for the ILA representation!

    - exercise every transition
    - since the final cell is always observable, there is no need for distinguishing sequences!

    Example 9:

15. Theorem: if the state machine for the ILA is reduced and each state is repeatable, then the ILA is C-testable.

    - a state s_1 in transition $(s_1 \rightarrow s_2)$ is repeatable if $\exists$ a sequence T that takes the machine from s_2 back to s_1

16. All ILA's can be made C-testable with additional PIs

    Example 10

17. Universal test set: a test set that can detect all detectable faults regardless of the underlying implementation of the combinational function
$\longmapsto$ Want this universal test set to be as small as possible

18. A vector v_1 *covers* another vector v_2 if for every logic 1 $\in v_2$, there is a logic 1 $\in v_1$.

19. Unate and binate: in a function z, if a variable x only appears in one polarity, then the function z is said to be unate in variable x, otherwise z is binate in x.

20. In order to compute universal test sets, we view the circuit as a truth table (can be constructed as a PLA), where inverters only appear at the PIs.

21. If the circuit has *no* unate variables (primary inputs), then the universal test set becomes the exhaustive test set!

22. true vectors and false vectors: an input vector that makes the function z evaluate to logic 1 is a *true* vector for z. Conversely for false vectors.

23. Monotone property: in a circuit with only AND and OR gates (no inverters), if $v_1 \supseteq v_2$, then $z(v_1) \supseteq z(v_2)$.

24. In a fault circuit with fault f, if $v_1 \supseteq v_2$, then $z_f(v_1) \supseteq z_f(v_2)$. Why?

25. Given true vectors v_1 and v_2, if $v_1 \supseteq v_2$, and if v_1 detects fault f, then we know $z(v_1) = 1$ and $z_f(v_1) = 0$. Since $z_f(v_1) \supseteq z_f(v_2)$, $z_f(v_2)$ must be 0 as well. Thus, if $z_f(v_1) = 0$, $z_f(v_2)$ must be 0; converse is not true.
$\longmapsto$ THEREFORE, we prefer true vector v_2 over true vector v_1 because v_2 will detect all faults v_1 detects, and possibly more.

26. minimal true vector: a true vector that does not cover any other true vector.

27. Given false vectors v_1 and v_2, if $v_1 \supseteq v_2$, and if v_2 detects fault f, then we know $z(v_2) = 0$ and $z_f(v_2) = 1$. Since $z_f(v_1) \supseteq z_f(v_2)$, $z_f(v_1)$ must be 1 as well. Thus, if $z_f(v_2) = 1$, $z_f(v_1)$ must be 1; converse is not true.
$\longmapsto$ THEREFORE, we prefer false vector v_1 over false vector v_2 because v_1 will detect all faults v_2 detects, and possibly more.

28. maximal false vector: a false vector that is not covered by any other false vector.

29. Universal test set: the union of min true vectors and max false vectors

    Example 11