VLSI / SOC Testing

Lecture 23

1. Where defects are located
   - random defects are controlled by process integrity
   - systematic failures/defects due to poor design, layout need to be identified to improve future yield
2. What causes systematic failure
   - aggressive design/layout styles
   - high-traffic lines too narrow
   - high-traffic lines too close
   - hard-to-fab regions, where density of material changes drastically in a short distance

   Example 1:

3. Inductive failure analysis
   - given layout, identify nets/signals that are susceptible to bridges, opens, etc.
   - statistical assumption: defects caused by particles of varying sizes of known probability of occurrence
     $\longmapsto$ given weighted critical area for bridges, one can compute the likelihood of a bridge for every net.

   Example 2:

4. Material density

   - a virtual grid is placed on the design/layout, where the dimensions of the grid are larger than the feature size
   - density is defined as the fraction of the grid covered
   - density gradient: change in density from one grid to another
   - observation: if density gradient are small for a given grid in all directions (within ϵ), then this grid is on a relatively uniform region
   - conversely, if density gradients change abruptly, it is generally harder to control the fabrication of such regions

   Example 3:

5. Design for yield improvement

   - identify potential places where fabrication may be difficult
   - sweep through the virtual grid plane, one layer at a time, to identify high-risk regions
   - can weigh each grid, giving more weight to nets with heavier traffic load

6. Yield prediction

   - prior to routing, estimate statistically channel density by approximating number of wires within channel
   - estimate density gradient based on approximate routing

7. Diagnosing a failure

   - objective: find the source of defect that caused the failure
     $\longmapsto$ may result in a number of possible sources/locations
   - static approach: based on signatures of potential modeled faults stored in a dictionary
   - dynamic approach: incrementally simulate and analyze the circuit to identify failure site

8. Static, dictionary-based diagnosis

   - idea: record signatures of every fault in a dictionary
   - Don't need to store complete responses for all primary outputs, just record the responses of erroneous outputs
   - using the dictionary, inductively search for the potential defect
   - Issues:
     - dictionary built by fault simulation without fault dropping, thus the cost could be high
     - dictionary may be large for large circuits
     - dictionary built for given modeled faults, which may not fully represent the actual defect, thus the response of actual defect may not match exactly with those obtained by fault simulation

   Example 4:

Example 5:

9. Adaptive diagnosis
   - pick the best vector to apply based on results obtained with each step

   Example 6:

10. Because defect response may not exactly match the simulated faulty response
    - score each fault correspondingly
    - find the fault that most closely resembles the defect's response
11. Distinguishability of faults
    - For a given test set T, the faulty response for two non-equivalent faults may be identical $\longmapsto$ T cannot distinguish these 2 faults
    - Can add more vectors to distinguish all distinguishable fault pairs
12. Diagnostic test generation
    - objective: enrich the test set so that more faults can be distinguished
    - modeled as a constrained ATPG problem on a pair of faults: search for a vector that detects one but not the other
13. Issues in diagnosis
    - Diagnostic accuracy: was the actual defect included in the final list of candidate sites
    - Diagnostic speed: time taken to find the candidate fault sites
    - Diagnostic resolution: how many fault sites did the diagnosis report
      - high resolution:
      - low resolution:
14. Simulation-based diagnosis
    - idea: gradually filter the nets (not faults) that the defect may be linked to, based on circuit structure and simulation results
    - helpful guides: cone intersection, sensitization, back-propagation, etc.
15. Diagnosis by cone intersection
    - step 1: divide the outputs into correct outputs and faulty outputs
    - step 2: compute fanin cones starting from the faulty outputs
    - step 3: compute intersection of the faulty output cones
    - Key: any gate/signal not included in the intersection cannot be solely responsible for the defect

Example 7:

16. Diagnosis by sensitization

    - idea: if gate a is responsible for the failure, then there must exist a sensitizable path from a to some faulty output(s)
    - note: there may result in multiple candidate gates, and there might have existed multiple defects in circuit
    - Algorithm:

```
for each erroneous vector $v$
    logic simulate with $v$
    for each signal $a$ in circuit
        complement the value at $a$
        simulate the fanout cone of $a$ due to this complementation
        if one or more erroneous output value is flipped, then $v$ can
            sensitize a discrepancy from $a$
```

Example 8

17. Diagnosis by back propagation

    - similar to critical path tracing, it finds candidate signals by backtracing from erroneous outputs
    - backtracing is done by traversing through sensitizable paths

    Example 9

18. Diagnosis for sequential circuits

    - complexity an order of magnitude higher
    - dictionary can still be built
    - fault can propagate through a number of time-frames before it reaches a PO

19. Diagnosis for BIST structures

    - only one signature for applying 100,000+ BIST patterns
    - increasing resolution is essential