VLSI / SOC Testing

Lecture 16

1. Testing circuits with clock gating

2. Testing circuits with multiple clocks

3. Testing Bridging Faults

   - Recall that a bridge is formed between a pair of signal lines
   - Types of bridges:
     - AND-bridge:
     - OR-bridge:
     - a-dom:
     - b-dom:
     - etc.

Example 1:

Example 2:

4. Modeling bridging fault as a single stuck-at fault:

5. Feedback bridges
   - forward and back signals can affect one another

   Example 3:

6. Issues on BF testing
   - AND, OR, a-dom, etc. simple models
   - Recall the Byzantine problem:
7. Bridges involving dynamic CMOS gates
   - bridge can cause the discharge time to increase or decrease
   - thus, the testing speed critical, may need slow testing to catch defect!

   Example 4:

8. Testing Delay-related Defects

   - took 20 years to go from 100KHz to 100 MHz; but took only 6 years to go from 100MHz to 1 GHz, and 18 months from 1 to 2 GHz
   - Deep submicron (DSM) effects now make delay harder to test, in which delay can be caused by
     - noise: coupling/cross-talk
     - process variations: change in device parameters
     - thermal induced: heat generated when holes and electrons recombine; instantaneous switching causes local hotspots (many tens of degree hotter than elsewhere). Can't reach this temperature during testing, thus insufficiently stressing the circuit.
     - power-induced: some nodes place a higher demand on power-grid, causing lower VDD locally, thus slowing switches
     - performance-induced: lower V_t (threshold voltage) make chip more susceptible to noise
   - Thus path delays much harder to predict and estimate
   - At-speed test is needed. Structural path-oriented tests may cause yield loss (detection of non-functional delay-path). Such faults won't hurt circuit performance.
     $\longmapsto$ just need to exercise functional paths. (non-functional paths may be helpful for diagnosis)

9. Issues related to delay-testing

   - With DSM effects, testing for only critical paths becomes insufficient. Perhaps to pick the paths that are most likely to cause a delay problem based on statistical methods? But how?
   - Shrinking transistor sizes $\rightarrow$ more transistors need to be interconnected, resulting in longer wires.
     $\longmapsto$ On-chip interconnect becoming the major performance limiter.
   - Longer wires have higher resistance, thus to reduce resistance, wires are grown taller (not wider). But taller wires have more side-wall capacitance, causing more coupling/cross-talk problems.
     $\longmapsto$ Intro of copper wires instead of aluminum - one-time relief only.

- Scan-based design: load state, launch and capture. But this can potentially cause a large current surges.
- Clock skew: clock is distributed across the chip - so every FF may see the clock arriving at a slightly different time. A shorter path to a FF that clock arrives first may cause problems.
  Further, if a delay-defect is on the clock, delivery of clock to (a set) FF may be further skewed.

10. Transition fault model

- Slow-to-rise (STR) and Slow-to-fall (STF) faults
- Aim: capture spot (lump) delay defects on a signal in the combinational path
- a slow-to-rise (fall) signal may cause the transition to arrive late at the FF
- Need: 2 vectors to test for a transition.
  - vector 1: initialization vector
  - vector 2: test vector (launch transition and propagate to an observation point)

Example 5:

11. Testing methods

    - Skewed Load: N-bit vector is loaded by shifting in the first N-1 bits, where N is the scan chain length. The last shift clock is used to launch the transition, followed by a quick capture.
      $\longmapsto$ only one vector is stored for each transition pattern in tester scan memory; the first vector is a shifted version of the stored vector.
    - Broadside testing (also called functional justification): a vector is scanned in and the functional clock pulsed to create the transition and subsequently capture the response.
      $\longmapsto$ only one vector is stored in tester per test; the second vector is derived from the first by pulsing the functional clock.
    - Enhanced-scan: two vectors (V1, V2) are stored in the tester memory. The first scan shift loads V1. It is then applied to the circuit under test to initialize it. Next, V2 is loaded, followed by an apply and a capture of the response. During shifting in of V2 it is assumed that the initialization of V1 is not destroyed.

12. Testing speed

    - High-speed testers extremely costly, testers usually slower than CUT
    - One solution is to add extra logic to circuit so that the speed of circuit in test mode becomes slower and comparable to tester speed
    - Alternative is to apply a vector k consecutive times
      $\longmapsto$ 1 ATE clock = k internal chip clock
      $\longmapsto$ output observed every k internal clocks

Example 7

13. If ATE is fast enough, we may want to apply slow-fast-slow testing in non-scan or partial-scan sequential circuits
    - Apply vectors for initializing starting state at slow speed such that the circuit can be considered delay-fault free
    - Apply the activation time-frame at regular speed
    - Apply fault effect propagation also at slow speed
14. Path-delay fault
    - a more general delay model
    - Aim: capture defects due to process variation (delay for each gate increased by δ), cumulative delay of a combinational path exceeding the specified amount
    - Path starts at a PI/FF and ends at a PO/FF
    - a signal is an on-input of path if it is directly on the path; a signal is an off-input of the path if it is an input to a gate on path but is not an on-input
    - Exponential number of paths possible
      $\longmapsto$ instead of testing all paths, select only the longest $\sim$100 paths from circuit
15. Static sensitization
    - a path is static sensitizable if $\exists$ a vector such that all off-inputs in the path settle at non-controlling values
16. Single-path sensitization test
    - all off-inputs of a test pair must be non-controlling values
    - most stringent
    - very few paths are single-path sensitization testable
17. Non-robust PDF test
    - target path is the only faulty path.
    - condition 1: a transition is produced at path input and propagates to a path destination
    - condition 2: all off-path inputs assume non-controlling states only in vector 2

Example 8

18. Robust PDF test

- guarantees detection of PDF irrespective of delay faults in other paths
- condition 1: all on-path signals have different values for V1 and V2 (real events occurred on the path)
- condition 2: all off-path inputs assume non-controlling states in vector 2
- if on-path event is a transition from non-controlling to controlling value, all off-path inputs for this gate must have a steady non-controlling value in both V1 and V2, in order to ensure that delays in off-paths will not affect propagation of target path
- $\longmapsto$ a robust test is also a non-robust test

Example 9:

19. Path Counting

    - one-pass through the circuit - inexpensive
    - even when the number of paths could be 10^{20} or more

    Example 10:

20. Speed binning

    - if a chip fails at 1GHz, it may work at a lower frequency
    - Key: functionally, it may still be correct

21. More advanced concepts in path-delay faults

    - Validatable non-robust path delay fault
    - Functional (not static) sensitizable paths
    - Primitive faults that involve co-sensitization
    - False path identification

22. Validatable non-robust path delay fault

    - Non-robust test for a path assumes no other path delay faults exists $\longmapsto$ otherwise, this non-robust test may be invalidated
    - Must make sure all transitions to off-inputs come from paths that are not delayed

    Example 11:

23. Functional sensitizable path delay faults

    - is **not** statically sensitizable
    - more than one faulty path exists in circuit, usually are related paths that fanout and reconverge along points in circuit

24. More on functional sensitizable path delay faults

    - if at least one off-input is not late, then this path is not functionally sensitizable, and needs not be considered (cannot impact circuit)

    Example 12:

    Example 13:

25. Primitive Faults

    - robust, non-robust, and validatable non-robust are all primitive faults of cardinality 1.
    - for cardinality > 1, we need:
      - the multiple path is static sensitizable
      - no proper subset of the multiple path is static sensitizable
    - the single paths in such primitive faults are said to be *co-sensitized*

Example 14: