Functional Testing

- Original testing method
- Run chip from reset
- Tester emulates the outside world
- Chip runs functionally with internally generated clock
  - PLL is working (frequency multiplied)
  - Can run much faster than the tester
  - At speed testing
- Low fault coverage (70% good)
- Test generation very difficult
  - designer writes assembly code for test vectors
Quiescent Current Testing

- Quiet CMOS draws little quiescent current
- Iddq testing is sensitive to these defects by simply monitoring the supply when CMOS clock is stopped
- Provides massive observability
- Need to perform test with several patterns
- Iddq testing catches defects that don’t cause logical failures but could be reliability problems
  - studies show better reliability with Iddq testing

General Interview Question
• Fault model represents many common faults
• Requires activating the defect
  – many patterns?
• ATPG tools all support Iddq stuck at fault detection
Opens Fault

- Common defect with shrinking metal geometries
- Easily missed with conventional testing
Gate Oxide Shorts

- Most common defect
- Considered a reliability issue
- Doesn’t necessarily cause a logical fault
Bridging

- Difficult to generate patterns for all possible bridging faults
- Common defect given new shrinking technologies
Iddq misses

- Leaky good devices cause hit to yield
- Low leakage bad devices cause test escapes
- Difficult to set the bar
Deep submicron technology

- In deep sub-micron designs, leakages are exploding and masking the effects of defects.
- This diminishes the effectiveness of Iddq.
- Can compare values for different patterns.
Delta Iddq

- Don’t look at large Iddq, look for large differences in Iddq between vectors on the same device.

<table>
<thead>
<tr>
<th>Vector 1</th>
<th>Vector 2</th>
<th>Vector 3</th>
<th>Vector 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2u</td>
<td>3u</td>
<td>110u</td>
<td>101u</td>
</tr>
<tr>
<td>3u</td>
<td>12u</td>
<td>111u</td>
<td>113u</td>
</tr>
<tr>
<td>2u</td>
<td>2u</td>
<td>110u</td>
<td>102u</td>
</tr>
<tr>
<td>1u</td>
<td>2u</td>
<td>113u</td>
<td>100u</td>
</tr>
</tbody>
</table>
Extreme Vdd Testing

- Can get greater sensitivity for defects running test much lower or higher than normal functional levels
- Very Low Voltage testing (VLV) is normally set to 2.5 times Vt
- High voltage testing (Burn-in levels) is 20% higher than maximum functional Vdd
- CMOS voltages are limited to oxide breakdown
- 10 MV/cm is intrinsic oxide breakdown level
Very Low Voltage Testing
MAX Vdd Testing

- Can stress marginal gate oxide shorts
  - latent to full defects
- Speeds up devices so wire delay is more prominent
- Burn-in
  - run parts for 168 hours at 1.2 MAX Vdd at high temperature
  - induce infant mortality latent defects by accelerating life
  - difficult to do now due to high leakage currents in DSM
Performance Testing with Scan

- Scan mainly used to test for DC faults
- Scan chains and control are not normally timed
- Possible to perform test with double vector to initiate a launch and capture of data in single functional period – need a transition to measure performance

Difficulties:
- creating two vectors from scan chains
- running test signals at speed
- testing all possible data paths
Delay Testing

- **Path Delay Testing**
  - Many paths in a design – too many to test
  - so test critical path based on static timing analysis

- **Transition Delay Testing**
  - extend stuck-at testing by providing initializing vector to each of the existing test vectors
  - only 2 times number of nets faults so high fault coverage
  - doesn’t check distributed problems along critical paths

- Both require two vectors to provide transition
Performance testing with shifting

Circuit with 12 x 2 different paths and 11 different nets
24 potential path delay defects (grows exponentially)
22 potential transition delay defects (grows linearly)
Ignoring the difference between rise and fall paths…
path CZ is the longest path (critical)
The transition should occur at time = 7 and is captured at Z at time = 8 (specified period). Circuit is considered good. Max Frequency is 1/8nS (125MHz).
However, slow processing could add 1 to each gate delay. This results in CZ growing from 7 to 11 and the test functions as intended by identifying the bad part. Part has a min period of 11 which gives an Fmax of 91MHz – slower than specified in the datasheet and is therefore rejected.
Example of Path Delay Test weakness

However, the number of paths explodes with increased circuit complexity and only the critical paths are tested.

If a delay defect on J exists and adds 4 units to all associated paths that travel through net J, path AX becomes critical as it is extended from 5 to 9 time units - longer than the specified period. However this path is not delay tested – thus resulting in a test escape.
Transition testing is more like SA in that only one net is inspected. Easy to implement over many nets.

If delay defect on D adds 4 units of delay, the example test will catch the slow to rise defect as the transition arrives at net Z at time 10 but is captured at 8.
If delay defect on K adds 4 units of delay, the example test will catch the slow to rise defect as the transition arrives at net X at time = 9 but is captured at 8.
Example of Transition Test weakness

However, if D has a delay defect of 1 and K has a delay defect of 2, both previously described transition tests will pass, but the path DX becomes critical due to the accumulated increased delay of the two defects and the chip becomes a test escape.
Double Vector Generation

- **Broadside – Functional Justification**
  - shift in first vector
  - then generate second from functional logic
  - requires ATPG across two functional periods
  - computationally expensive

- **Scan shifting**
  - shift in first vector
  - then generate second by additional single bit shift
  - easy to generate but limited

- **Special flip-flops that hold two test vectors**
Scan in 01 to set up initialization level.
Shown values are statically held prior to test.
Delay testing example scan shifting

Shift scan chain one more time shifting in a 0. Creates a transition on through the critical path.
shift scan chain one more time shifting in a 0. Creates a transition on through the critical path.
Simple 2 Flop shift delay test example

- **clk**
- **scan_enable**
- **scan_in**
- **scan_out**
- **functional input Q1**

Scan in test vector, Launch edge, perform test, scan out results (scan in next vector)

time
Scan in 00 to set up initialization level. Shown values are statically held prior to test.
scan enable is low. need two cycles. First cycle captures justified data and starts transition.
scan enable is low. need two cycles.
First cycle captures justified data and starts transition.
Second cycle captures at speed results 10.
Simple 2 Flop broadside example

- clk
- scan_enable
- scan_in
- scan_out
- functional input Q1

Scan_in test vector
Launch edge
Perform test
Scan out results (scan in next vector)
Three Pattern Tests

- Delay testing requires two patterns
  - initialize
  - launch
- Some gates require preconditioning vector to get true worst case timings
- Internal nodes are pre-charged high or low
- Real problem – no one does the test
  - too difficult to implement
Three Pattern Tests

P I L
A 1 0 1
B 1 0 1
C 0 0 1

Internal nodes pre-charged
N-Test

- Statistically proven that testing for a stuck-at fault multiple times improves reliability
- Not just applying same test vector over and over, but...
- Testing for stuck-at fault with different test vectors
- Create test vectors that that test 99% of nodes N times vs. one time (traditional)
- Adds to number of test vectors – requiring additional tester memory and more ATPG time
- Also requires increases test time

AMD Interview Question