Fault models

Mani Soma

Acknowledgements

- Some materials from various sources
  - Dr. Phil Nigh, IBM
  - “Principles of Testing Electronic Systems” by S. Mourad & Y. Zorian
  - “Essentials of Electronic Testing” by M.L. Bushnell and V.D. Agrawal

Defects, faults and errors

- Physical defects
  - fabrication defects (missing or extra materials)
- Faults
  - behavior due to defect or abstract model of defect
- Errors
  - incorrect operation
  - design errors, fabrication errors, faults
  - physical failures (including wear-out, etc.)

Faults and time duration

- Permanent: failure, fault or error always present and stable.
- Intermittent: fault or error only occasionally present due to unstable hardware or varying hardware states.
- Transient: fault or error resulting from temporary environmental conditions.
- Reliability: failure, fault or error not initially present, but occurs during operation due to a physical change in hardware.

Physical defect mechanisms

Fault types

- Abstract models of physical defects
- Structural faults
- Functional faults
- Timing and delay faults
- Permanent, intermittent, transient faults
Fault assumptions

- Single fault
  - fan-out issues: stem and branch faults
- Structural:
  - Interconnect faults
    - A s-a-0 (A/0) or s-a-1 (A/1)
    - bridging fault (short): AND, OR
    - open fault (break)
  - Component (transistor) faults
    - Stuck-open (or stuck-off), stuck-short (or stuck-on)

Test vector

- A test vector \( t \) is a specification that includes both:
  - input to be applied
  - expected output
  - example: \( t = I/O \) (e.g. 0001/11)
- A test vector detects a fault if the output under fault is different than expected output

Detecting stuck-at faults

<table>
<thead>
<tr>
<th>Input</th>
<th>Fault Pos</th>
<th>Fault Response at X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Example

- fault: OR-bridge between X1 and X2
- test: 011/01 detects this fault (Z1 -> 1)
- what if:
  - AND-bridge? detect at Z2
  - X2/0?
  - X3/0?

Fault detection requirements

- Activation of the faults
  - apply appropriate inputs
- Propagation of the fault effect(s) to output (or observable point)
  - concept of "path sensitization"
  - controlling inputs of a logic block
- \( f \) is detectable if there exists a test \( t \) that detects \( f \)
Fault detection example

- b/0 detectable
- a/1 undetectable
- b/0 undetectable if a/1 exists

Undetectable faults

- Invalidation of single-fault assumption
- Path re-convergence
  - negative re-convergence
  - positive re-convergence
- Design with redundancies
  - unintentional: needs to be optimized
  - intentional: redundancy (e.g. triple-modular redundancy), hazard reduction

Hazard reduction vs. fault

abc = 111
--> 011

OUT = ab + bc + a'c = ab + a'c ==> Y/0 undetectable

Notes on fault detection

- Remove or disable intentional design redundancies while testing
- Fault detection: NP-complete problem
- Polynomial algorithms exist for realistic cases

Single stuck-at fault (SSF) model

- represents many physical faults
- independent of technology
- detecting SSFs also detects many other faults
- reasonable test set size: ~2n faults for n-net circuits
- can be used to model other faults

Notes on SSF detection

- Combinational fanout-free circuits: need to detect only SSFs at PIs.
- Combinational circuits with fanouts: detects SSFs at PIs and fanout branches.
- Core of most test generation tools.
- Can be used for sequential circuits with DFT (e.g. scan).
Bridging faults

- 2 or more lines shorted together
- Common occurrence with open faults
- Deterministic: wired-OR, wired-AND
- Indeterminate: bridge with impedance (CMOS)
- Feedback or non-feedback
  - feedback: rare occurrence
- Determined by layout (NOT schematic)

Detection

- Procedure:
  - set two signals to opposite values
  - propagate two fault effects
  - justify lines for both values
- Feedback fault: potential oscillation, sequential problem
- IDDQ detection

IDD current waveform

- Input transitions
- IDD of fault-free circuit
  - Monitor quiescent IDD (IDDQ) current
  - Some defects generate high IDDQ current while not creating any stuck-at effects

IDD waveform with fault

- Input transitions
- IDD of fault-free circuit
- IDD of faulty circuit
  - High-Current state

Test for bridging faults

- SSF tests detect a large number (80%-90%) of bridging faults
- Dual-path propagation to generate tests
- May need 2 or more tests to detect one bridging fault
- IDDQ test generation and DFT
- Conservative layout rules to reduce faults

Gate oxide shorts

- p-substrate (GND)
- Gate oxide shorts
- Pinhole connection gate & source node
- Vdd
Gate oxide shorts (2)

- Predominant cause of reliability failures and quality problems
- Degrades over time
  - 1 MΩ to 10 KΩ to 10 Ω
- May cause high IDD before affecting performance

Open faults

- Common in ICs and systems (boards, MCMs)
- Modeling problem:
  - high-value resistor may be inappropriate
  - other models: capacitor, controlled source
- Introduces “state” into a circuit
  - CMOS inverter example
- IDDQ test techniques

Resistive metal opens

- longer delay
- detectable with SSF tests

Various open faults

- Polysilicon opens
  - Before fan-out:
    - similar to metal opens
    - detectable with SSF tests
  - After fan-out
    - longer delays, higher IDD
- Contacts, source, drain
  - Longer delays, higher IDD

MOS faults

- Stuck-open (or stuck-off), stuck-short (or stuck-on)
- G, D, S, B open; two-terminal short (6 faults)
- Oxide defects and breakdown
- Effects: sequential operation, changes in timing, etc.
- Possible detection by IDDQ
Stuck-open example

Vector 1: test for $A\ s-a-0$
(Initialization vector)

Two-vector test can be constructed by
ordering two $s/a$ tests

Vector 2 (test for $A\ s-a-1$)

Stuck-open example

Stuck-short example

Test vector for $A\ s-a-0$

I_{DDQ} path in
faulty circuit

Good circuit state

Faulty circuit states

Stuck-short example

Good circuit state

Faulty circuit states

Sequential circuits and faults

- More complex behavior due to feedback
- Test sequence (instead of a single vector) required.
- A test sequence $T$ detects a fault if and only if for every pair of initial states $q$ and $q_f$, the output sequences are different for some sub-vector in $T$

Faults in a latch

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Fault-free $Q$</th>
<th>Faulty Response $Q_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR 01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SR 00</td>
<td>$Q_{n-1}$ 1</td>
<td>$Q_{n-1}$ 1</td>
</tr>
<tr>
<td>SR 10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SR 11</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Sequential circuit example

Sequential fault detection

- State table (next state, out)
- Output sequence with input $T=10111$
- State transition: ADADBC

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Fault-free Output</th>
<th>$T$</th>
<th>$Q_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011</td>
<td>0011</td>
<td>1100</td>
<td>11101</td>
</tr>
<tr>
<td>00010</td>
<td>0010</td>
<td>1100</td>
<td>11101</td>
</tr>
<tr>
<td>00011</td>
<td>0011</td>
<td>1100</td>
<td>11101</td>
</tr>
<tr>
<td>01011</td>
<td>0111</td>
<td>1100</td>
<td>11101</td>
</tr>
</tbody>
</table>
Test sequences
- Initialization sequence
  - to get the circuit to a known state
  - design guideline: Master-Reset to all FFs
  - note: fault detection on the Reset line
- Homing sequence: to get to the state where fault is manifested
- Propagation sequence: to get the fault effect to the output or observable point

Fault equivalence
- Equivalence: same outputs under two faults
- No test can distinguish 2 equivalent faults
- Fault equivalence classes: disjoint and cover entire fault set (Set Theory results)

Fault dominance and collapsing
- Fault dominance
  - f1 dominates f2 if the test set T2 for f2 is a subset of the test set T1 for f1
  - Any test detecting f2 also detects f1
  - Sufficient to generate tests for f2 only
- Fault collapsing to reduce fault list size
  - Use equivalence relations
  - Use dominance relations

Fault collapsing example
- Test set reduction
  - good for test cost reduction
    - a0 ~ b0, d0 ~ h0, g1 ~ f1, e1 ~ i1...
    - 24 faults -> 10 faults
  - bad for fault diagnosis

Multiple Stuck-at fault model
- each line has 3 states: good, s/0, s/1.
- number of faults: \(3^n - 1\).
- assume only k faults at a time:
  - \(i\) faults at a time -> number of faults is \(2^i\)
  - number of possible combinations of \(i\) lines in an \(n\)-line circuit is \(\binom{n}{i}\)
  - \(i\) can vary from 1 to \(k\)
  - number of MSF = \(\sum_{i=1}^{k} \binom{n}{i} 2^i\)

Other fault models
- Functional faults
  - based on functions and specifications
- Timing and delay faults
  - gate delay faults, path delay faults
  - Usually two-pattern tests
  - critical path: static and dynamic
  - very important in high-speed circuits
  - some techniques rely on SSF model
  - see papers in IEEE Trans. CAD, ITC
Other fault models (2)

- Transient faults
  - Power supply, radiation
- Metastability
  - Latch or FF switches only halfway
- Intermittent faults
  - Pattern-sensitivity (e.g. in RAM)
  - Crosstalk, ground bounce, etc.
  - Possible thermal causes
- Design needs to stay within margins

Transition Fault Model

Six transition faults
1. A *slow-to-rise*
2. A *slow-to-fall*
3. B *slow-to-rise*
4. B *slow-to-fall*
5. OUT *slow-to-rise*
6. OUT *slow-to-fall*

Defect-fault correspondence

<table>
<thead>
<tr>
<th>Defect type</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net-to-net short</td>
<td>Logical fault &amp; high IDD</td>
</tr>
<tr>
<td>Net-to-net power supply short</td>
<td>Stuck fault &amp; IDD</td>
</tr>
<tr>
<td>Power supply-power supply short</td>
<td>Gross logical failure &amp; IDD</td>
</tr>
<tr>
<td>Opens causing a global net open</td>
<td>Stuck fault (IDD?)</td>
</tr>
<tr>
<td>Opens causing a single floating</td>
<td>Delay fault &amp; IDD</td>
</tr>
<tr>
<td>Sourse/Drain node</td>
<td>Logical fault or very long delay fault (IDD?)</td>
</tr>
<tr>
<td>Resistive source/drain contact</td>
<td>Delay or logical fault</td>
</tr>
<tr>
<td>Source-to-drain shorts</td>
<td>Delay or logical fault &amp; IDD</td>
</tr>
<tr>
<td>Gate-oxide shorts</td>
<td>Delay or logical fault &amp; IDD</td>
</tr>
<tr>
<td>P-n junction leakage</td>
<td>Delay or logical fault &amp; IDD</td>
</tr>
<tr>
<td>Blocked implant</td>
<td>Delay or logical fault</td>
</tr>
</tbody>
</table>

Fault coverage comparison

Test Coverage

- Hardware measurements (Sematech project)
- IDDq coverage

Defect vs. Coverage vs. Yield

- Defect Level (% DPM)
- Fault Coverage, %
- Yield (%)

Fault coverage comparison

- 100% Coverage
- 50% Coverage

Defect level vs. Fault coverage

- Theoretical
- Experimental
Conclusion

- Fault models and effects for digital systems
- Test vectors and detection methods
- New issues: deep-submicron technologies, high operating frequencies, power supply reduction, increasing noise
- Faults in mixed analog-digital systems