More Details on Faults - Organization

- Fault Modeling
  - Equivalent Faults & Collapsing
  - Gate Level Faults
  - Transistor Level Faults
  - Bridging Faults

- Fault Detection
  - Gate Level Faults
  - Transistor Level Faults
  - Bridging Faults
  - Fault Simulations
    - Undetectable & Potentially Detected
    - Fault Coverage
Fault Modeling

*Recall:*

- A good fault model has 2 requirements:
  1. accurately reflects the behavior of a physical defect
  2. is computationally efficient with respect to simulation
- Single fault models are used for requirement # 2
- The most commonly used current fault models include:
  - Gate level stuck-at faults
    - stuck-at-0 (sa0) & stuck-at-1 (sa1)
  - Transistor level stuck faults
    - stuck-on (stuck-closed) & stuck-off (stuck-open)
  - Bridging faults (shorts between wires)
    - wired-AND & wired-OR
    - dominant (one driving source dominates the other)
    - Note: opens in wires typically covered by stuck-faults
Gate Level Stuck-at Fault Model

- Gate inputs or outputs can be:
  - **Stuck-at-0 (sa0)**
    - as if input or output were disconnected and tied low to Vss
  - **Stuck-at-1 (sa1)**
    - as if input or output were disconnected and tied high to Vdd
  - fault site denoted by ‘X’ with sa0/sa1
    - Note: there is no feedback of fault value from fault site!
Gate Level Equivalent Faults and Fault Collapsing

- Equivalent faults are indistinguishable & can be collapsed
  ⇒ 1 fault in set of equivalent faults represents all in set
  ⇒ Fewer faults to simulation ⇒ faster fault simulations

- Gate level collapsing
  ⇒ **AND (NAND) gates**
    - any input sa0 = output sa0 (sa1)
      ⇒ # Collapsed Faults = $I + 2$ (where $I = \#$ inputs)
  ⇒ **OR (NOR) gates**
    - any input sa1 = output sa1 (sa0)
      ⇒ # Collapsed Faults = $I + 2$ (where $I = \#$ inputs)
  ⇒ **INVERTER**
    - input sa0 (sa1) = output sa1 (sa0)
      ⇒ # Collapsed Faults = 2
Gate Level Equivalent Faults and Fault Collapsing

**AND**

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Bsa0</th>
<th>Bsa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td><strong>0</strong></td>
<td>0</td>
<td><strong>0</strong></td>
<td>0</td>
<td><strong>0</strong></td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td><strong>0</strong></td>
<td><strong>1</strong></td>
<td>0</td>
</tr>
</tbody>
</table>

**INVERTER**

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
</tbody>
</table>

**OR**

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Bsa0</th>
<th>Bsa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td><strong>1</strong></td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td><strong>1</strong></td>
<td><strong>1</strong></td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td><strong>1</strong></td>
<td>0</td>
</tr>
</tbody>
</table>
Structural Equivalent Faults and Fault Collapsing

Cannot collapse faults at fan-out stem since it would violate single stuck-at fault model

Fan-out stem sa1 = K sa1 & J sa1
Fan-out stem sa0 = K sa0 & j sa0

Only 2 collapsed faults for inverter chain (Z sa0 & Z sa1)

# collapsed faults = 12
Collapsed vs. Uncollapsed Gate Level Fault Sets

- For a given circuit (assuming elementary logic gates)
  ⇒ # uncollapsed faults = \(2(G + G_I)\)
    - \(G\) = total # gates
    - \(G_I\) = total # gate inputs
  ⇒ # collapsed faults = \(2(O_P + F_S) + G_I - N_I\)
    - \(O_P\) = # primary outputs
    - \(F_S\) = # fan-out stems
    - \(N_I\) = # inverters
  ⇒ typically #collapsed flts \(\approx \frac{1}{2}\) # uncollapsed flts

- Should faults be collapsed?
  ⇒ YES: for TPG and fault simulation (more efficient)
  ⇒ NO: for computing fault coverage (more accurate)
    - but longer fault simulation times
Gate Level Fault Detection

**AND**

Collapsed fault set

\[ Z_{sa0} = A_{sa0} = B_{sa0} \]

\[ Z_{sa1} = A_{sa1} = B_{sa1} \]

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Bsa0</th>
<th>Bsa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minimum Set of Test Vectors

01
10
11

**INVERTER**

Collapsed fault set

\[ Z_{sa1} = A_{sa1} \]

\[ Z_{sa0} = A_{sa0} \]

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Minimum Set of Test Vectors

01
1

**OR**

Collapsed fault set

\[ Z_{sa1} = A_{sa1} = B_{sa1} \]

\[ Z_{sa0} = A_{sa0} \]

\[ Z_{sa0} = A_{sa1} \]

<table>
<thead>
<tr>
<th>AB</th>
<th>Z</th>
<th>Asa0</th>
<th>Asa1</th>
<th>Bsa0</th>
<th>Bsa1</th>
<th>Zsa0</th>
<th>Zsa1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Minimum Set of Test Vectors

00
01
10
Minimum Set of Test Vectors for Gate Level Faults

- Inverter requires both input logic values (0 and 1)
  \[ \Rightarrow \text{# vectors} = 2 \]

- \(N\)-input AND or NAND gate
  \[ \Rightarrow \text{# vectors} = N + 1 \]
  - all 1s
  - walk 0 through a field of 1s

- \(N\)-input OR or NOR gate
  \[ \Rightarrow \text{# vectors} = N + 1 \]
  - all 0s
  - walk 1 through a field of 0s

- XOR ≠ elementary logic gate (*made from multiple gates*)
  \[ \Rightarrow \text{# faults depends on construction of gate} \]
  - 3 vectors required for pin faults
  - all 4 vectors required for internal faults
Transistor Level Fault Model

- Transistor can be:
  - Stuck-on (a.k.a. stuck-short)
    - can result in excessive $I_{DDQ}$
  - Stuck-off (a.k.a. stuck-open)
    - can result in “memory” node (logic gate $\Rightarrow$ latch)

- Gate level fault accurate for NMOS but not for CMOS
  - gate input stuck-at = 2 transistors stuck-at in CMOS

<table>
<thead>
<tr>
<th>AB</th>
<th>A PFET s-on</th>
<th>A NFET s-on</th>
<th>B PFET s-on</th>
<th>B NFET s-on</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
</tr>
<tr>
<td>01</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
</tr>
<tr>
<td>10</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
<td>1 mem</td>
</tr>
<tr>
<td>11</td>
<td>0 mem</td>
<td>0 mem</td>
<td>0 mem</td>
<td>0 mem</td>
</tr>
</tbody>
</table>
Transistor Level Fault Equivalence & Collapsing

- Stuck-off faults in series transistors are equivalent
- Stuck-on faults in parallel transistors are equivalent
- \# collapsed transistor faults = \(2T - N_{ser} + G_{ser} - N_{par} + G_{par}\)
  \(\triangleright N_{ser} = \text{total \# series transistors}\)
  \(\triangleright G_{ser} = \text{total \# groups of series transistors}\)
  \(\triangleright N_{par} = \text{total \# parallel transistors}\)
  \(\triangleright G_{par} = \text{total \# groups of parallel transistors}\)

- # collapsed faults = 6
  A PFET s-on = B PFET s-on
  A NFET s-off = B NFET s-off
  A PFET s-off
  B PFET s-off
  A NFET s-on
  B NFET s-on
Transistor Level Fault Detection

- Transistor fault detection more difficult than gate level
  - **Stuck-on faults**
    - voltage divider may not produce incorrect logic values
    - monitoring $I_{DDQ}$ is best approach
  - small currents may be lost in leakage of >2M transistors

- **Stuck-off can**
  - will produce wrong logic values
  - need ordered set of 2 vectors to detect

<table>
<thead>
<tr>
<th>A PFET</th>
<th>A NFET</th>
<th>B PFET</th>
<th>B NFET</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>s-on</td>
<td>s-off</td>
<td>s-on</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>mem</td>
<td>$I_{DDQ}$</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>$I_{DDQ}$</td>
<td>0</td>
</tr>
</tbody>
</table>

**Vectors to detect:**
- A PFET stuck-off
  - 11 - to get $Z=0$
  - 01 - to detect $Z=0$
- B PFET stuck-off
  - 11 - to get $Z=0$
  - 10 - to detect $Z=0$
- A/B NFETs s-off
  - 0x or x0 - to get $Z=1$
  - 11 to detect $Z=1$

Min. #vectors = 4

detects s-on w/ $I_{DDQ}$
Bridging Fault Models

- Two current models for wires shorted together:
  - **Wired-AND/Wired-OR** fault model
    - Shorted wires perform logical AND or OR
  - **Dominant** fault model
    - Stronger driving gate dominates the short
- For $N$ nets, \# pair-wise bridging faults = $N^2 - N$ (either model)
  - No fault equivalence $\Rightarrow$ no fault collapsing
Bridging Fault Detection

- Wired-AND/Wired-OR faults
  - 1 vector (01 or 10) with 2 outputs (A’ and B’), or
  - 1 output (A’ or B’) with 2 vectors (01 and 10)
- Dominant faults
  - 1 vector (01 or 10) with 2 outputs (A’ and B’)
    - harder to detect than Wired-AND/OR (less observable)

Detecting all dominant BFs \(\Rightarrow\) detecting all Wired-AND/OR

Wired-AND fault model

Wired-OR fault model

<table>
<thead>
<tr>
<th>AB</th>
<th>A’B’</th>
<th>WAND</th>
<th>WOR</th>
<th>AdomB</th>
<th>BdomA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

A dominates B model

B dominates A model
Fault Detection

*Recall:*

- Fault detection requires:
  - Observation of an error (from fault) at a primary output
    - Observability of the fault site
      - The ease at which we can observe the fault behavior
  - Input stimuli that creates an error as a result of fault
    - Controllability of the fault site
      - The ease at which we can control the fault behavior
  - Controllability of path from fault site to primary output
    - Typically considered part of observability

- Testability $\propto$ controllability & observability

- Any given fault may be:
  - Detectable
  - Undetectable
  - Potentially detectable
Gate Level Fault Detection - Path Sensitization

1. At fault site, assign logic value opposite that of stuck-at fault
2. From fault site, choose a path to a PO assigning non-controlling values to all other inputs to gates in that path
   \[ 1 = \text{non-controlling value for AND/NAND gates} \]
   \[ 0 = \text{non-controlling value for OR/NOR gates} \]
3. For all assigned values, back-trace to PIs selecting input values that will produce the assigned values
4. If there is a conflict, repeat Steps 2 & 3 choosing new paths and/or values in Step 3
   If no path can be found without conflict, the fault *may be* undetectable, otherwise values at PIs form test vector
Undetectable Faults

- No test vector can detect the fault
  ⇒ usually difficult to prove a fault is undetectable
- Undetectable faults due to:
  ⇒ Re-convergent fan-out, and
  ⇒ Redundant logic

Hazard-free multiplexer has undetectable faults due to re-convergent fanout and redundancy
Undetectable Faults (cont.)

- Minimize circuit to remove redundancy & these faults
  \(\Rightarrow\) undetectable fault may not effect circuit operation
    - wasted area
  \(\Rightarrow\) undetectable fault slows down fault simulation
    - all test vectors are simulated (no trip on mismatch)

- Sometimes undetectable faults are unavoidable
  \(\Rightarrow\) hazard-free circuits
    - glitch-free clock multiplexing
  \(\Rightarrow\) initialization circuitry
    - power-up presets
    - global resets
  \(\Rightarrow\) use these cktts sparingly to minimize undetectable faults
Potentially Detected Faults

- Due to undefined logic values ($U,2,X$)
  - an artifact of logic simulation
    - can be a 0 or a 1 but simulator doesn’t know
    - used for un-initialized logic
  - faults preventing initialization produce $U,2,X$
- Potential detect fault if good ckt = 1/0 & faulty ckt = $U,2,X$
  - potential detect faults may be detected by other vectors
  - probability of detect of a fault $\propto$ # potential detects
    - for high data activity, otherwise probability = 0.5
  - in real circuit, fault may/may not be detected
    - depends on power-up value
  - in simulation, PDFs also show up as undetected faults
Potentially Detected Faults (cont.)

Examples of potentially detected fault

- Select input to MUX stuck-at-0
  \[\Rightarrow\] flip-flop cannot be initialized
- Clock stuck-at-0 and stuck-at-1
  \[\Rightarrow\] flip-flop cannot be initialized
- 3 potentially detected faults
  \[\Rightarrow\] high detection probability for
    - high data activity on Sel & Din
    - high clock frequency
  \[\Rightarrow\] otherwise these may not be detected
Fault Simulation

- Fault simulator emulates faults and compares resultant response to known good circuit output responses
- Fault simulation long for large fault lists - speed-up:
  - simulation of a given faults ends on detection
  - parallel flt simulation emulates 1 flt/bit (computer word)
  - statistical fault sampling (>1000 samples = good estimate)
Fault Coverage/Grading

- Given a set of test vectors, each fault in fault set can be:
  - $D = $ detected faults
    - Targeted faults and faults “accidentally” detected
  - $X = $ undetectable faults
    - There are NO vectors that can detect these faults
  - $U = $ undetected faults
    - Could not find vector to detect fault (but there could be one)
  - $P = $ potentially detected faults (PDF)
    - Also included in $U$
  - $T = $ total faults = $D + X + U$

- Fault coverage = $(D+P/2) / T$
  - Detectable FC = $(D+P/2) / (T - X)$
    - Note: assumes PDF detection probability = 0.5
Need to Add

- Dominant-AND/OR BF model
- N-detectability