Fault Modeling

• Fault Model: a set of assumed faults in a system such that testing for them will test for most faults of a specific class.
• Used for test generation, simulation and quality evaluation.
• Hide complexities of actual defects. Infinitely many defects possible.
• Based on past knowledge of defect modes and modeling experience.

Common Fault Models

• No model: test exhaustively
• Hardware fault models:
  – Gate level: stuck-at 0/1 bridging delay faults
  – Transistor level: stuck on/open bridging
• Functional fault models
• Software fault model
  – Is there one?
Stuck-at 0/1 Model

- Classical model, well developed results/methods
- May not describe many defects in today’s VLSI, still a nice way of structural “probing”.
- Model: any one or more of these may be stuck at 0 or 1: a gate input, a gate output, a primary input.
- Justification: many lower level defects can be shown to have an equivalent effect.

Stuck-at 0/1 Example

Find a test vector for x2 s-a-1:
Input = (x1,x2,x3) = (0,0,1) Output = 0 normally
1 if faulty
Single Fault Assumption

- Assumption: only one fault is present at a time.
- Significantly reduces complexity.
- Good for fault detection: complete single stuck test set will detect almost all multiple faults.
- Not good for fault location.
- How many multiple faults?
  - Assume k lines
  - 3 states per line: normal, s-a-0, s-a-1
  - Total $3^k - 1$ faulty situations (k=1000, total $1.3 \times 10^{477}$)

Bridging (Short) Fault Model

- Common assumption: only nearby lines can be bridged
- Model: Two lines $x$ and $y$ bridged can cause both to take the value
  - $x \ \& \ \ y$ AND-bridging (0-dominance)
  - $x \ \lor \ \ y$ OR-bridging (1-dominance)
  - Depends on technology, transistor dimensions etc.
Bridging Faults

With AND bridging:
\[ Z_1 = x_1 \cdot x_2 \]
\[ Z_2 = (x_1 \cdot x_2) + x_3 \]

Feedback bridging:
- Oscillations: odd inversions, sufficient delay
- Settling at intermediate voltage level

Delay Fault Model

- Excessive path delay or gate delay
- Signals may be sampled before stabilization
MOS Transistors

N-channel

0 = open
1 = closed

P-channel

0 = closed
1 = open

CMOS NOR Gate

A=1
B=0

V_{DD}=H

Output=0

Gnd=L

verify

NAND?
Switch-level Fault Model (1)

- **Model:** A transistor may be
  - stuck-open
  - Stuck-on

  - **PA stuck-open:** output effectively s-a-0
  - **NA stuck-open:** To sensitize output
    -(A,B)=(1,0)
    - Normal output: 0
    - Faulty output: high imp: previous value: sequential behavior!
  - **Needed test-pair**
    - \( T_1 \) \((0,0)\) output= 1 (initialize)
    - \( T_2 \) \((1,0)\) normal
    - 1 if faulty

Switch-level Fault Model (2): **Stuck-ON**

- **Assume PA is stuck-ON**
  - \((A,B)=(1,0) \rightarrow \text{normal out}=0\)
  - **faulty out=?**
  - **Depends on relative resistances (dimensions etc)**
  - **Low resistance between \( V_{DD} \) and Gnd: very high supply current (I_{DDQ})**
Shorts and IDDQ

• Logical value can not be predicted in general.
• Very high supply current ($I_{DDQ}$)
• Generally $I_{DDQ}$ very effective for detecting some defects.