

Overview of Faults - Organization

- Fault Modeling
- Single vs. Multiple Faults
- Common Fault Models
 - ⇒ Gate Level Faults
 - ⇒ Transistor Level Faults
 - ⇒ Bridging Faults
- Fault Detection
- Fault Simulations
- Fault Coverage

Fault Modeling

- A good fault model meets two requirements:
 1. accurately reflects the behavior of a physical defect
 2. is computationally efficient with respect to simulation
- 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
 - ⇒ Delay faults (excessive delay in a gate or path)

Single vs. Multiple Fault Models

- Let N = number of fault sites
 - ⇒ fault sites are gate inputs/outputs, transistors, etc.
- Single vs. multiple fault models
 - ⇒ Single - there can be only one fault at a time
 - ⊙ # Possible Faulty Circuits = $2N$
 - ⊙ less accurate but more efficient to simulate
 - ⇒ Multiple - there any number of faults at a time
 - ⊙ # Possible Faulty Circuits = $3^N - 1$
 - ↳ the -1 circuit is the fault-free circuit
 - ⊙ more accurate but cannot possibly simulate all fault groups
 - ↳ statistical fault sampling is only practical solution
- High single Fault Coverage (FC) ⇒ high multiple FC
 - ⇒ in general, 100% single FC ⇒ > 98% multiple FC

Gate Level Stuck-at Fault Model

- Gate inputs or outputs can be:

⇒ Stuck-at-0 (sa0)

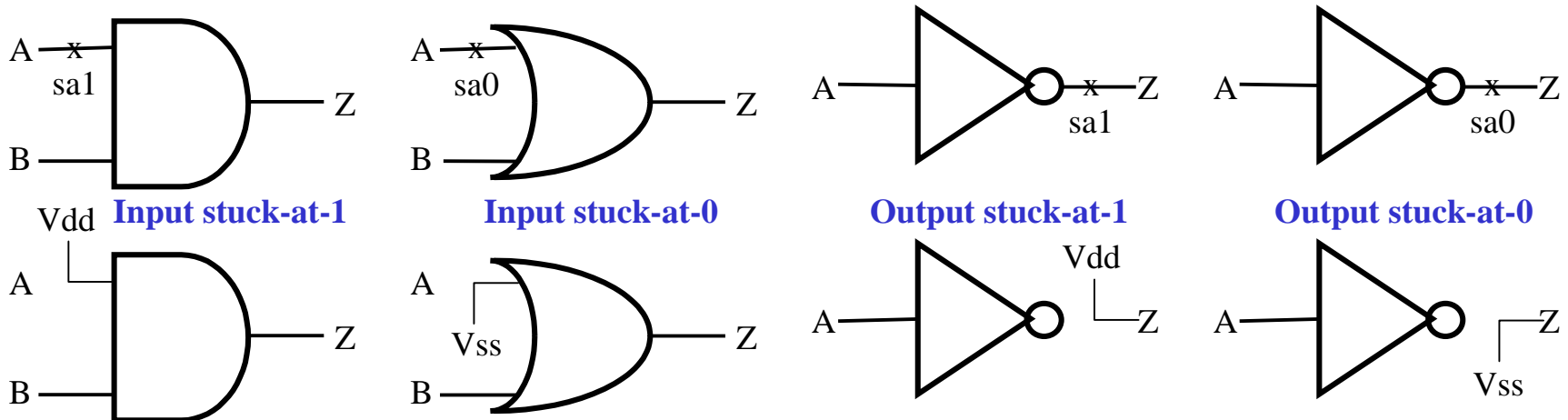
⊙ as if input or output were disconnected and tied low to V_{ss}

⇒ Stuck-at-1 (sa1)

⊙ as if input or output were disconnected and tied high to V_{dd}

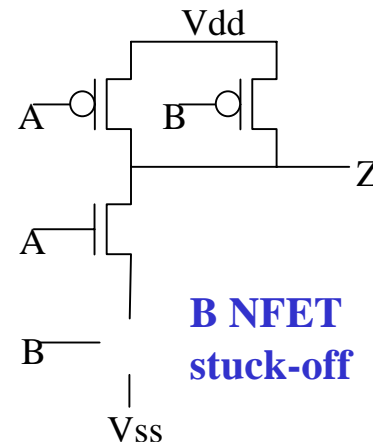
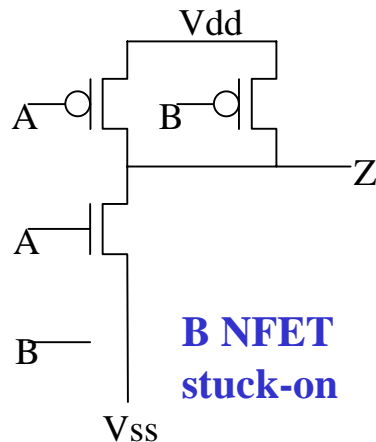
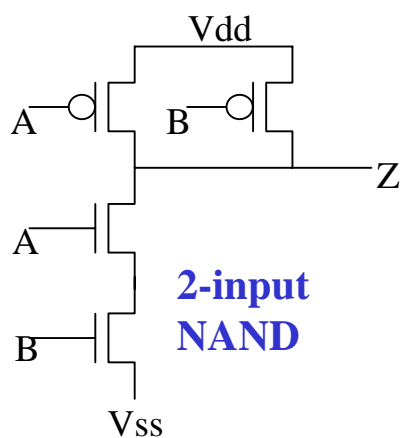
⇒ fault site denoted by 'X' with 'sa0' or 'sa1'

⊙ Note: there is no feedback of fault value from fault site!



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



		B NFET	
AB	Z	s-on	s-off
00	1	1	1
01	1	1	1
10	1	I_{DDQ}	1
11	0	0	mem

Bridging Fault Models

- Two current models for wires shorted together:

⇒ **Wired-AND/Wired-OR** fault model

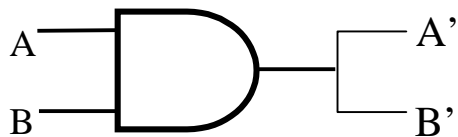
- Shorted wires perform logical AND/OR

⇒ **Dominant** fault model

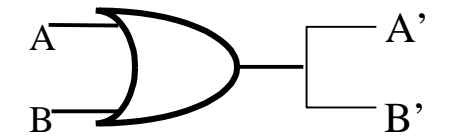
- Stronger driving gate dominates the short

- For N nets, # pair-wise bridging faults = $N^2 - N$

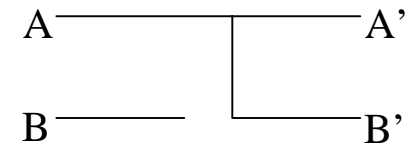
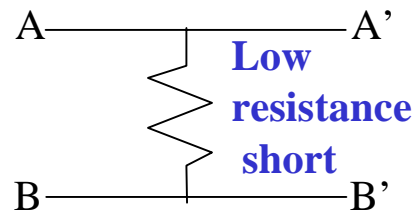
⇒ Either **wired** or **dominant** faults under single fault model



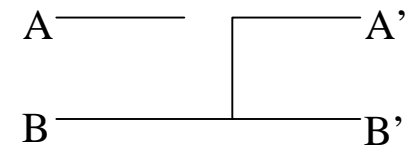
Wired-AND fault model



Wired-OR fault model



A dominates B model



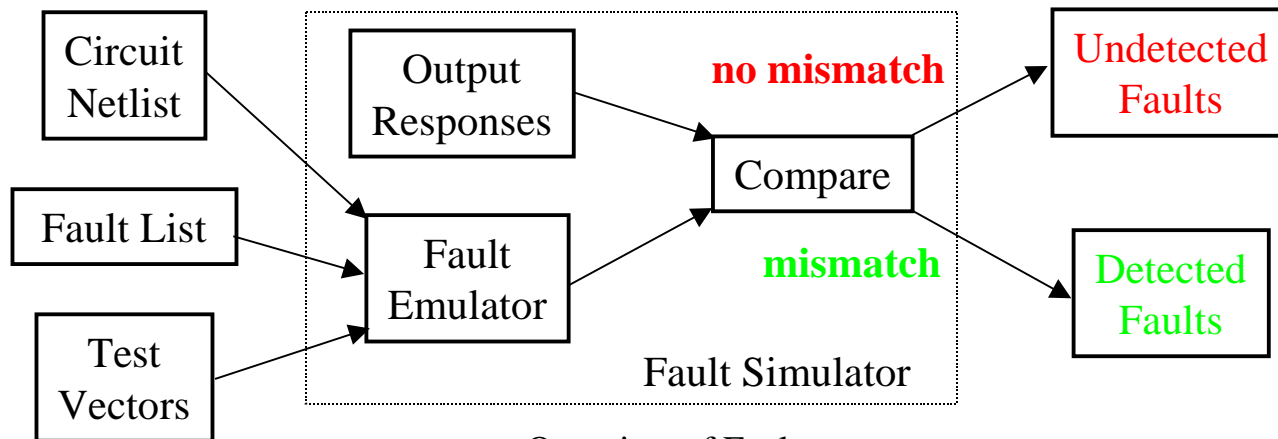
B dominates A model

Fault Detection

- Fault detection requires:
 - ⇒ observation of an error (from the 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
 - ↳ also considered part of *observability*
- Testability of fault \propto **controllability** & **observability** of site
- Circuit testability \propto overall circuit **controllability** & **observability**

Fault Simulation

- Fault simulators:
 - ⇒ emulate faults
 - ⇒ compare output response to known good circuit output response
 - ↳ for a given set of input test patterns
- Fault simulation time is long for large fault lists - speed-up by:
 - ⇒ simulation of a given fault ends on detection
 - ⇒ parallel fault simulation emulates 1 fault/bit (computer word)
 - ⇒ statistical fault sampling (>1000 samples = good estimate)



Fault Coverage

- Given a set of test vectors, each fault in the fault set for the circuit can be:

⇒ $D = \text{detected faults}$

- ⊙ Targeted faults and faults “accidentally” detected

⇒ $U = \text{undetected faults}$

- ⊙ Could not find a vector to detect fault

↳ *but there might be one*

⇒ $T = \text{total faults} = D + U$

- Fault coverage = D / T

⇒ For that specific set of test vectors

⇒ Also referred to as *fault grading*

Testability Tidbits

- In the absence of global presets/resets (my experience):
 - ⇒ testability \propto initializability
 - ⇒ difficult to initialize subcircuits \Rightarrow difficult to test
 - ⇒ therefore, logic simulation gives early indication of difficult to test portions of design
 - ⊙ logic simulation provides cheap testability analysis
- Logic gates are 100% testable & easily testable
 - ⇒ Therefore, testability problems come from design
 - ⊙ the way designers interconnect the gates
 - ⇒ Hence, the need for Design for Testability