Fault Modeling
Outline

- Single Stuck-At Fault Model
- Other Fault Models
- Redundancy and Untestable Faults
- Fault Equivalence and Fault Dominance
- Method of Boolean Difference

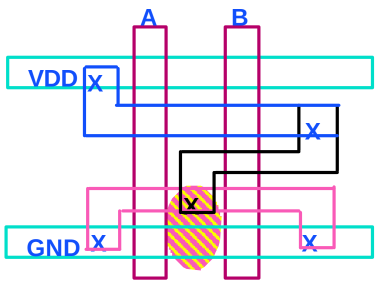
Fault Modeling

Modeling the effects of physical defects on the logic function and timing

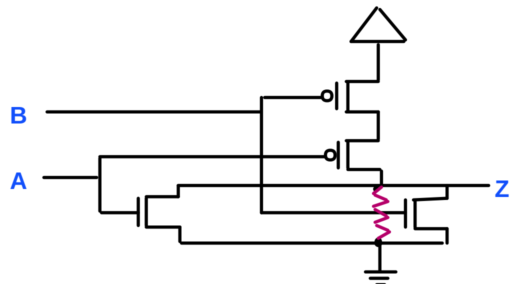
- Physical Defects
  - **Silicon defects**
  - **Photolithographic defects**
  - **Mask contamination**
  - **Process variations**
  - **Defective oxide**
- Electrical Effects
  - **Shorts (0 resistance)**
  - **Opens (∞ resistance)**
  - **Transistor stuck-on, stuck-open**
  - **Resistive shorts and opens**
  - **Change in threshold voltage**
- Logical Effects
  - **Logic s-a-0 or 1**
  - **Slower transition (delay faults)**
  - **AND-bridging, OR-bridging**

Fault Modeling

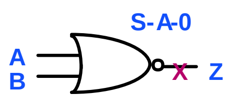
- **Physical Defect**



- **Electrical**



- **Logical**



Causes of Physical Defects

- Manufacture
  - **Mask contamination, dust particles**
  - **Fabrication area contamination**
- Aging Effects
  - **Metal migration**
  - **Oxide degradation due to trapped charge**
- Handling
  - **Electrostatic discharge**

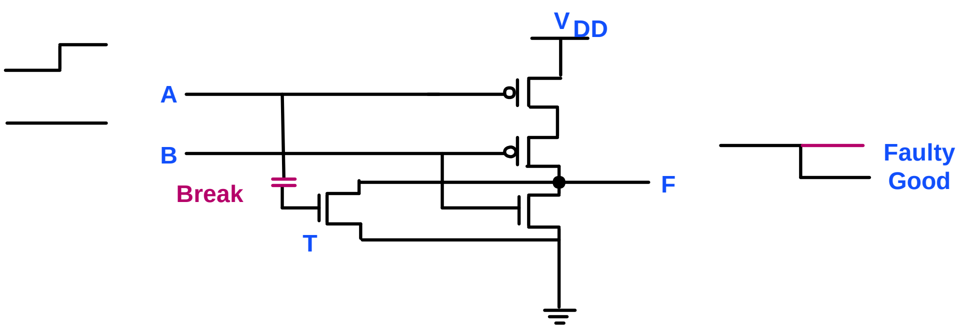
Single Stuck-At Fault Model

- Why Use Single Stuck-At Fault (SSF) Model?
  - **Complexity is greatly reduced. Many different physical defects may be modeled by the same logical stuck-at fault.**
  - **SSF is technology independent**
    - Has been successfully used on TTL, ECL, CMOS, etc.
  - **Single stuck-at tests cover a large percentage of multiple stuck-at faults.**
  - **Single stuck-at tests cover a large percentage of unmodeled physical defects.**

Relationship Between Single and Multiple Faults

- Irredundant two-level circuit
  - **Complete test for SSF also detects all MSF [Kohavi and Kohavi, 1972]**
- Fanout-free circuit
  - **Any complete test for SSF detects all double and triple faults [Hayes, 1971]**
- Internal fanout-free circuit (fanout > 1 on primary inputs only)
  - **Any complete test for SSF detects greater than 98% of MSF with 5 or fewer faults [Agarwal and Fung, 1981]**

CMOS Stuck Open Faults



- The break in the gate input to transistor T causes it to remain open

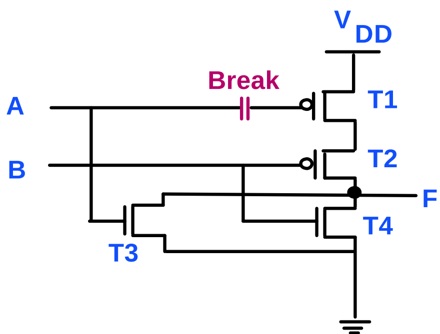
CMOS Stuck Open Faults

A	B	F_{good}	F_{faulty}
0	0	1	1
0	1	0	0
1	0	0	previous F (floating F)
1	1	0	0

Test Sequence for T stuck-open: A,B = 0, 0 then 1, 0

CMOS Stuck On Faults

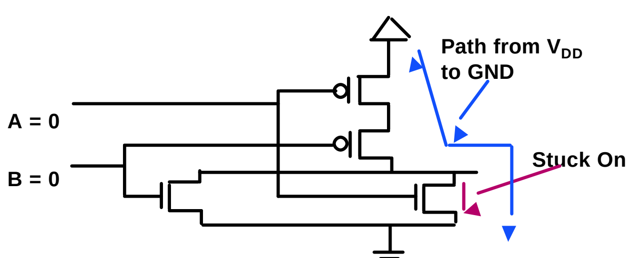
- Transistor T1 is always conducting



- Observations:
  – Correct logic function, but degraded logic levels
  – Higher signal transition times, results in a Delay Fault.
  – When A=1, and B=0, transistors T1, T2, and T3 are all conducting resulting in an IDDQ Fault
  – If the open is resistive, IDDQ fault will not happen, but Delay Fault is very likely

I_{DDQ} Faults

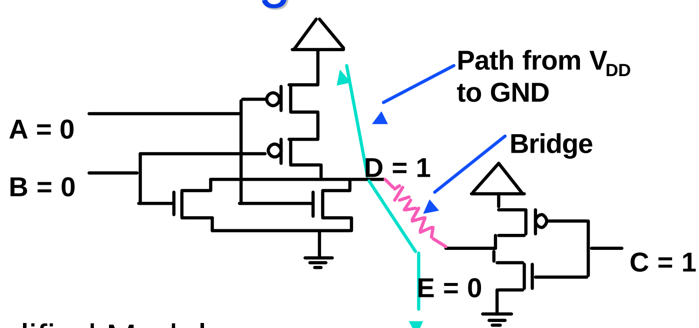
- A path that draws current from Vdd to ground



I_{DDQ} Faults

- Advantages
  - **Covers most bridge faults**
  - **Covers some open faults**
  - **Higher defect coverage than stuck-at tests**
- Disadvantages
  - **Circuit must be designed with low I_{DDQ}**
  - **Test application slow**
  - **Some open faults escape I_{DDQ} tests**
  - **Some timing faults escape I_{DDQ} tests**
  - **Current threshold has to be empirically established**

Bridge Faults



- Simplified Models
  - Wired-AND, Wired-OR
- More Realistic Models:
  - Bridge resistance
  - V_{th} of successor gates

Transition Faults

- Slow-to-rise (0 to 1) transition
- Requires a two-pattern sequence <V1, V2> for a slow-to-rise fault on line k:
  - **V1 sets line k to 0**
  - **V2 tests fault k stuck-at-0**

- Slow-to-fall (1 to 0) transition
- Requires a two-pattern sequence <V1, V2> for a slow-to-fall fault on line k:
  - **V1 sets line k to 0**
  - **V2 tests fault k stuck-at-0**

Delay Faults

- Model defects that affect the circuit timing (resistive shorts and opens)
- *Transition* faults and *Gate Delay* faults
  - **Models slow-to-rise or slow-to-fall transition on logic gate**
- *Path Delay* Faults (robust and non-robust testing):
  - **Models slow-to-rise or slow-to-fall transition on some path(s) from primary input to primary outputs**
    - **Advantage:** covers transition and gate delay faults
    - **Disadvantages:** test application in non-scan sequential circuits cannot be done at-speed. Number of paths may be (exponentially) large

Testing: Structured vs. Unstructured

- **UNSTRUCTURED:** functional vectors, ATPG vectors
  - **Takes time to get high fault coverage, especially for sequential designs.**
  - **Hard as today's ASIC designs use embedded logic blocks, memories etc**
    - "I did not design the whole chip"
    - "With automated synthesis I do not know the details of my logic"
- **STRUCTURED:** DFT, BIST
  - **Adds additional hardware (increases area)**
  - **Trade-off between area and test time**

Structured Testing: Stuck-at Faults

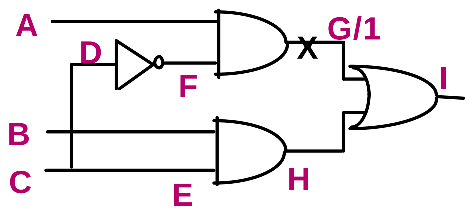
- **Sensitized line for vector t:** a line whose logic value is not correct for vector t in presence of a fault
- **Sensitized path for t:** a path of sensitized lines

- A fault is **detectable (testable)** if ∃ vector t that:
  - **Excites the fault:** produces complemented to correct (faulty) logic value at fault location

    AND

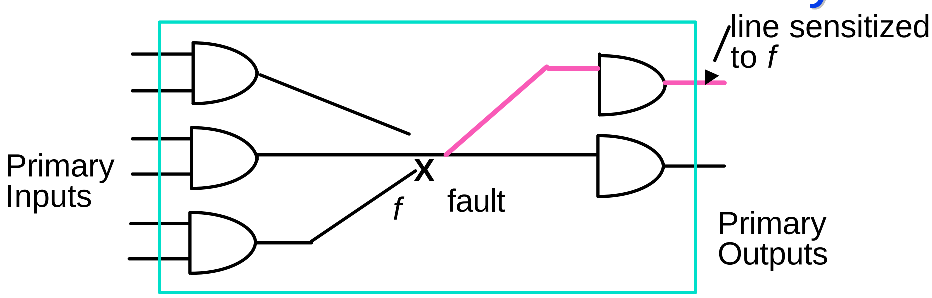
  - **Propagates the faulty effect:** it produces a sensitized path to some primary output from faulty location

Testing of Stuck-at Faults



- <u>**Fault Excitation**</u>: Applying a logic value opposite to the stuck-at value at the fault site.
- <u>**Error Propagation**</u>: Applying appropriate logic values in the circuit to make the error visible at the primary outputs.
- ABC = 00x is a test vector for G/1 (G s-a-1).

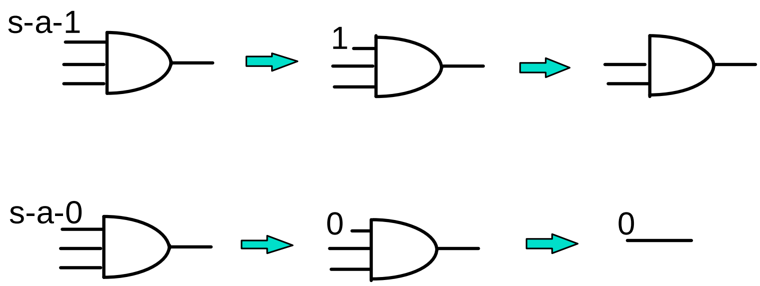
Combinational Redundancy



- Sometimes a fault *f* on line *l* cannot be excited or cannot be propagated or both.
- Then the fault *f* is termed untestable.
- If the fault f is untestable, then the fault f is redundant, i.e., the line l or the associated gate can be removed from the circuit without changing the logic function.

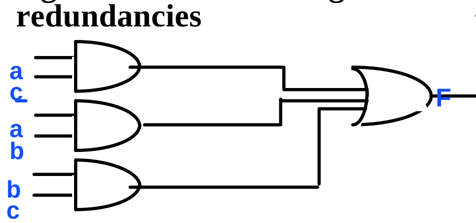
Untestable Faults

- If a fault *f* on line *l* is untestable, then either the line or the gate can be removed without changing the function.



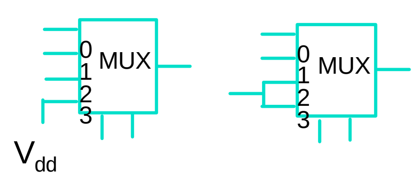
Why Redundancy Exists

- **Unintentional redundancies occur due to poorly optimized designs. This can happen for hand designed or synthesized circuits**
- **Interconnect of synthesized individually non-redundant logic blocks can create global redundancies**

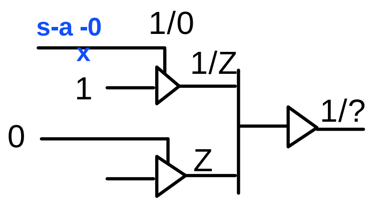


e.g. $F = ac + \bar{a}b + bc$

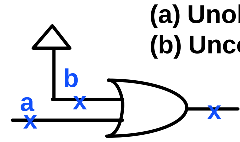
- Intentional Redundancies
  - Duplicated logic for increase in drive, speed (carry logic)
  - Duplicated logic for error detection
  - Additional terms for hazard removal
  - Use of over-designed library cells, e.g., for a 3-to-1 mux, use 4-to-1 mux



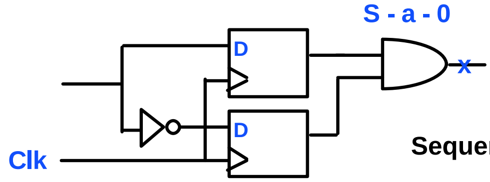
Other Untestable Fault Classes



TriState Untestable



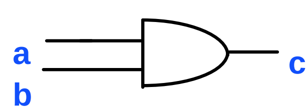
(a) Unobservable
(b) Uncontrollable



Sequentially Redundant

Fault Equivalence

- A fault 'a' is equivalent to fault 'b' in the logic circuit F, if the logic function F(a) realized in the presence of fault 'a' is identical to the logic function F(b) in presence of fault 'b'.
- Fault 'a' s-a-0 is equivalent to faults 'b' s-a-0 and c s-a-0
- Equivalence is useful in reducing the size of a fault list

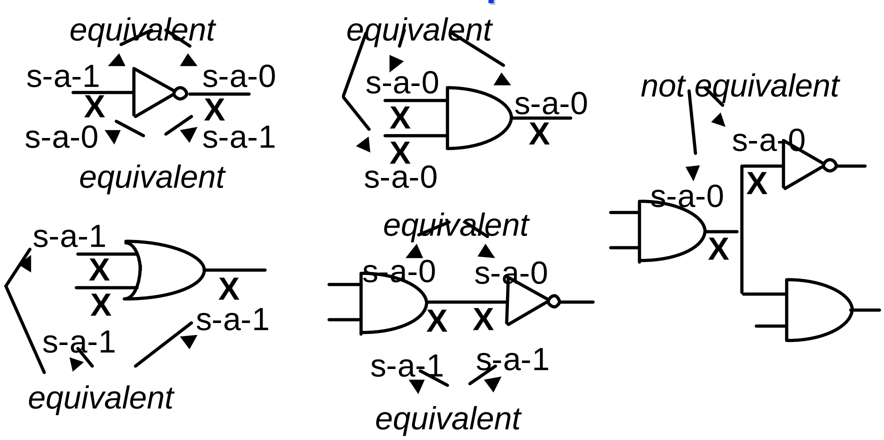


For n-input gates, need only to consider n+2 faults.

Method of Structural Equivalence

- Introduce fault *f* in the network N and reduce the structure to $S(N_f)$.
- Similarly obtain the structure $S(N_g)$.
- If $S(N_f) = S(N_g)$ then clearly function $N_f = N_g$ and therefore *f* and *g* are equivalent faults.
- Structural equivalence implies functional equivalence.

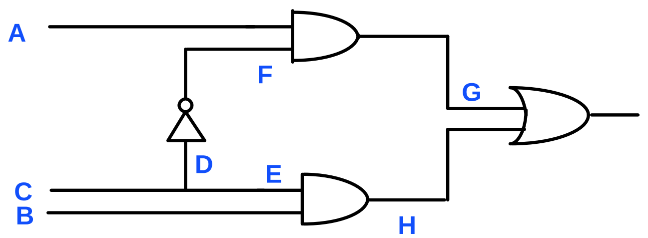
Local Analysis for Structural Equivalence



- Equivalence classes obtained may not be maximal, but they are obtained quickly.

Fault Collapsing

- Fault collapsing is the process of retaining only one fault from each group of equivalent faults
- In the 2-input multiplexer there are 18 single fault sites
- The collapsed list has 10 faults:
  - *{A1, B1, C0, C1, E1, F1, G0, H0, I0, I1}*



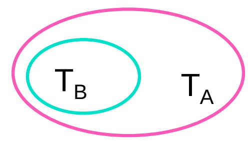
Fault Collapsing

ALGORITHM:

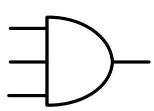
- **Insert all possible s-a-0 and s-a-1 faults on every line in the circuit**
- **Traverse circuit gates and collapse faults**
- **Easy way: depth=1. Can do for higher depth (but it can become exponential)**

Fault Dominance

- Let T_A be the set of all test vectors for fault A and T_B be the set of all test vectors for fault B.
- Then fault A dominates fault B (written $B \subseteq A$) iff $f_A = f_B$ for all vectors in T_B.
- It follows that $T_B \subseteq T_A$
  - **A test for B is a test for A.**
  - **If B is tested, then A is tested.**
  - **A can be removed.**
- If $T_B \subseteq T_A$ then it does not always mean that the boolean functions $f_A = f_B$ for vectors in T_B.



Fault Dominance



{a/0, b/0, c/0, d/0}

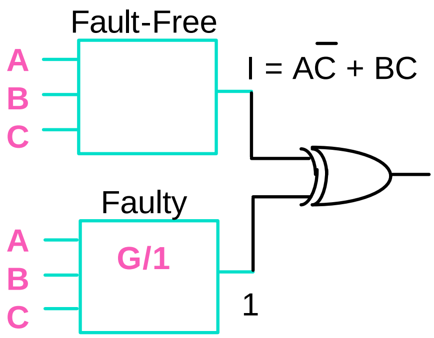
$T_{a/1}$ = {011} $T_{b/1}$ = {101} $T_{c/1}$ = {110}

$T_{d/1}$ = {0xx, x0x, xx0}

d/1 dominates a/1, b/1, and c/1

- No fault equivalence or dominance relationship exists between a stem and its fanout branches.
- Dominance does not hold in sequential circuits.

Boolean Difference



$\overline{(A\overline{C} + BC)}$

$= (\overline{A}+C)(\overline{B}+\overline{C})$

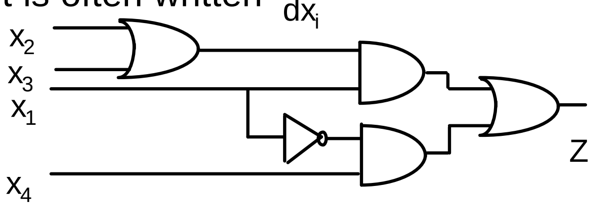
$= \overline{A}\,\overline{B} + \overline{A}C + \overline{B}\,\overline{C}$

All test vectors:

ABC		ABC
00X	➡	000
0X0		001
X01		010
		101

Method of Boolean Difference

- Definition: Given a function $Z(x_1, x_2, \ldots x_n)$, the Boolean Difference of Z with respect to x_i is defined as $Z(x_1, x_2, \ldots x_n)_{x_i = 0} \oplus Z(x_1, x_2, \ldots x_n)_{x_i = 1}$
- It is often written $\frac{DZ}{dx_i}$



$Z = x_1(x_2+x_3) + \overline{x_1}x_4$

$$\frac{DZ}{dx_4} = Z_{x_4 = 0} \oplus Z_{x_4 = 1} = [x_1(x_2+x_3)] \oplus [x_1(x_2+x_3) + \overline{x_1})] = \overline{x_1}$$

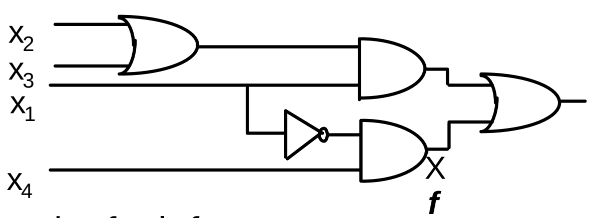
Method of Boolean Difference

- Now consider x_4 stuck-at-0. Any test vector for $x_4/0$ must satisfy $Z_{x_4=0} \neq Z_{x_4=1}$
- In addition, the test must satisfy $x_4 = 1$.
- Therefore, all test vectors satisfy

$$\frac{DZ}{dx_4} x_4 = 1 \Rightarrow \overline{x_1} x_4 = 1$$

$$(x_1, x_2, x_3, x_4) = (0,-,-,1)$$

Internal Nodes



- Consider the fault *f*.
  - **Rewrite Z by cutting the wire at *f* as a 5-variable function**
    - $Z(x_1, x_2, x_3, x_4, f) = x_1(x_2+x_3) + f$
  - $\frac{DZ}{df}$ **must be 1 for f to propagate to the output**
  - **Also express $f = F(x_1, x_2, x_3, x_4) = x_1 x_4$**
  - **If f is stuck-at-0, the excitation requirement is $F(x_1, x_2, x_3, x_4) = 1$**

Internal Nodes

- **If f is stuck-at-1 then $F(x_1, x_2, x_3, x_4) = 0$.**
- **Therefore, all test vectors are expressed by the equation**
  - $\frac{DZ}{df} f = 1$ for f stuck-at-0
  - $\frac{DZ}{df} \overline{f} = 1$ for f stuck-at-1
- $\frac{DZ}{df}$ **$= x_1(x_2+x_3) \oplus 1 = \overline{x}_1 + \overline{x}_2\overline{x}_3$**
- **Tests for f stuck-at-0 are**
  - $\overline{x}_1 x_4 [\overline{x}_1+\overline{x}_2\overline{x}_3)] = \overline{x}_1 x_4$

