

Fault Modeling

- **Some Definitions**
- **Why Modeling Faults**
- **Various Fault Models**
- **Fault Detection**
- **Fault Collapsing**

(Source: NCTU 周景揚 教授)

Some Real Defects in Chips

- **Processing Faults**
 - missing contact windows
 - parasitic transistors
 - oxide breakdown
- **Material Defects**
 - bulk defects (cracks, crystal imperfections)
 - surface impurities (ion migration)
- **Time-Dependent Failures**
 - dielectric breakdown
 - electromigration
- **Packaging Failures**
 - contact degradation
 - seal leaks

Faults, Errors and Failures

- **Fault: A physical defect within a circuit or a system**
 - May or may not cause a system failure
- **Error: Manifestation of a fault that results in incorrect circuit (system) outputs or states**
 - Caused by faults
- **Failure: Deviation of a circuit or system from its specified behavior**
 - Fails to do what it should do
 - Caused by an error
- **Fault ---> Error ---> Failure**

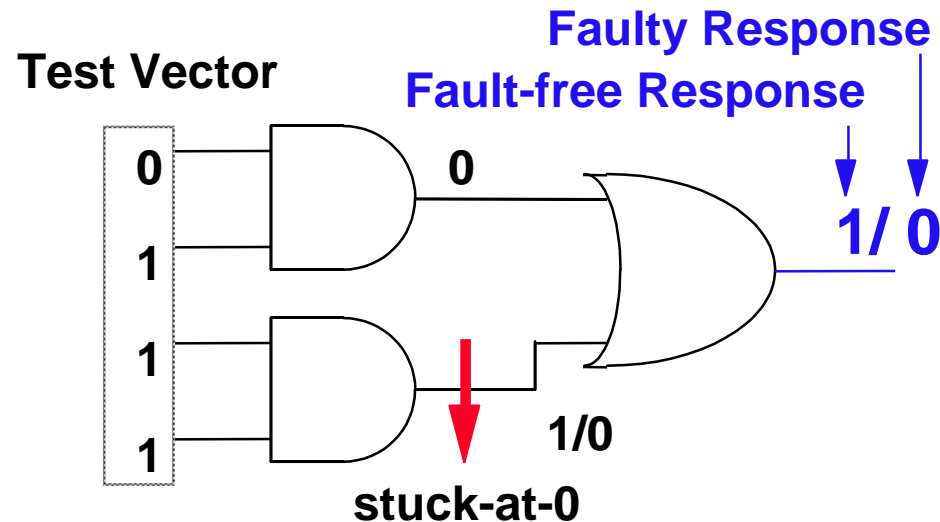
Why Model Faults ?

- **Fault model identifies target faults**
 - Model faults most likely to occur
- **Fault model limits the scope of test generation**
 - Create tests only for the modeled faults
- **Fault model makes effectiveness measurable by experiments**
 - Fault coverage can be computed for specific test patterns to reflect its effectiveness
- **Fault model makes analysis possible**
 - Associate specific defects with specific test patterns

Fault Models

- **Stuck-At Faults**
- **Bridging Faults**
- **Transistor Stuck-On/Open Faults**
- **Functional Faults**
- **Memory Faults**
- **PLA Faults**
- **Delay Faults**
- **State Transition Faults**

Single Stuck-At Faults



- Assumptions:
- Only one line is faulty.
 - Faulty line permanently set to 0 or 1.
 - Fault can be at an input or output of a gate.

Multiple Stuck-At Faults

- **Several stuck-at faults occur at the same time**
 - Important in high density circuits
- **For a circuit with k lines**
 - there are $2k$ single stuck-at faults
 - there are $3^k - 1$ multiple stuck-at faults

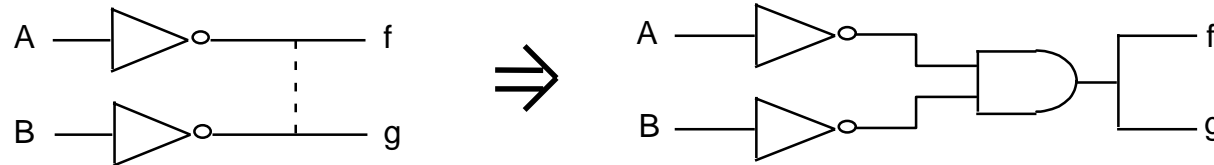
Why Single Stuck-At Fault Model?

- **Complexity is greatly reduced.**
Many different physical defects may be modeled by the same logical single stuck-at fault.
- **Single stuck-at fault is technology independent.**
Can be applied to TTL, ECL, CMOS, etc.
- **Single stuck-at fault is design style independent.**
Gate Arrays, Standard Cell, Custom VLSI
- **Even when single stuck-at fault does not accurately model some physical defects, the tests derived for logic faults are still valid for most defects.**
- **Single stuck-at tests cover a large percentage of multiple stuck-at faults.**

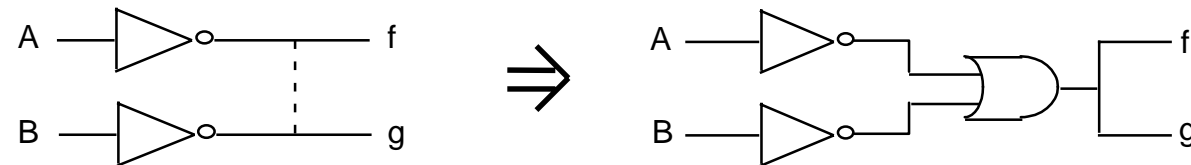
Bridging Faults

- **Two or more normally distinct points (lines) are shorted together**
 - Logic effect depends on technology

- **Wired-AND for TTL**

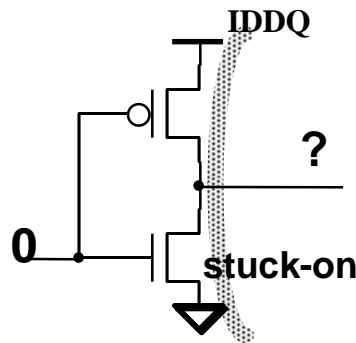


- **Wired-OR for ECL**



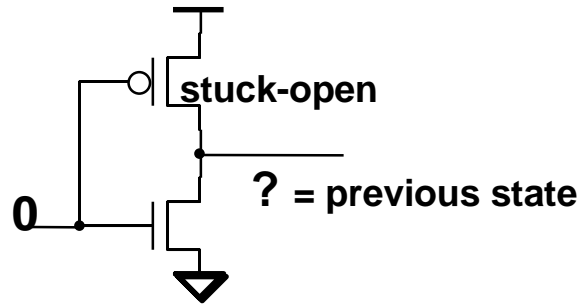
- **CMOS ?**

CMOS Transistor Stuck-ON



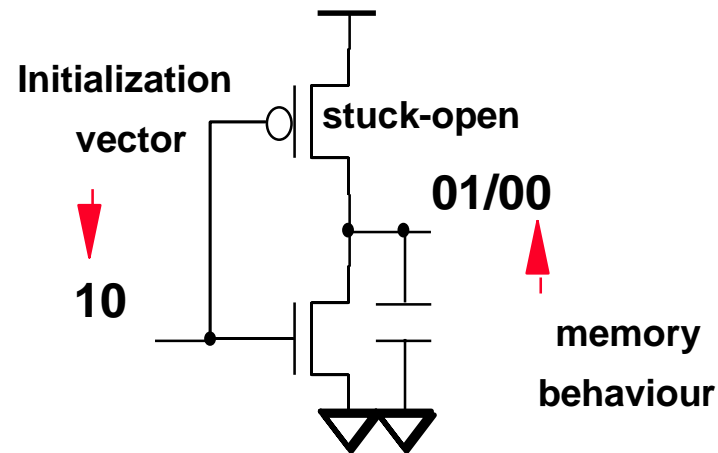
- **Transistor stuck-on may cause ambiguous logic level.**
 - depends on the relative impedances of the pull-up & pull-down networks
- **When input is low, both P and N transistors are conducting causing increased quiescent current, called IDDQ fault.**

CMOS Transistor Stuck-OPEN



- Transistor stuck-open may cause output floating.

CMOS Transistor Stuck-OPEN (Cont.)



- Can turn the circuit into a sequential one
- Stuck-open faults require two-vector tests

Functional Faults

- **Fault effects modeled at a higher level than logic for function modules, such as**

Decoders

Multiplexers

Adders

Counters

RAMs

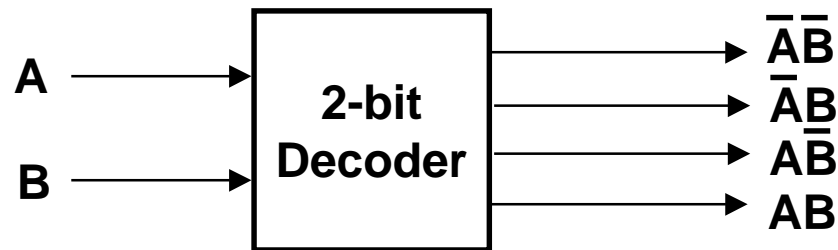
ROMs

Functional Faults of Decoder

$f(L_i/L_j)$: Instead of line L_i , Line L_j is selected

$f(L_i/L_i+L_j)$: In addition to L_i , L_j is selected

$f(L_i/0)$: None of the lines are selected



Memory Faults

- **Parametric Faults**
 - Output Levels
 - Power Consumption
 - Noise Margin
 - Data Retention Time
- **Functional Faults**
 - Stuck Faults in Address Register, Data Register, and Address Decoder
 - Cell Stuck Faults
 - Adjacent Cell Coupling Faults
 - Pattern-Sensitive Faults

Memory Faults (Cont.)

- **Pattern-sensitive faults: the presence of a faulty signal depends on the signal values of the nearby points**
 - Most common in DRAMs

0	0	0
0	d	b
0	a	0

$a=b=0 \rightarrow d=0$

$a=b=1 \rightarrow d=1$

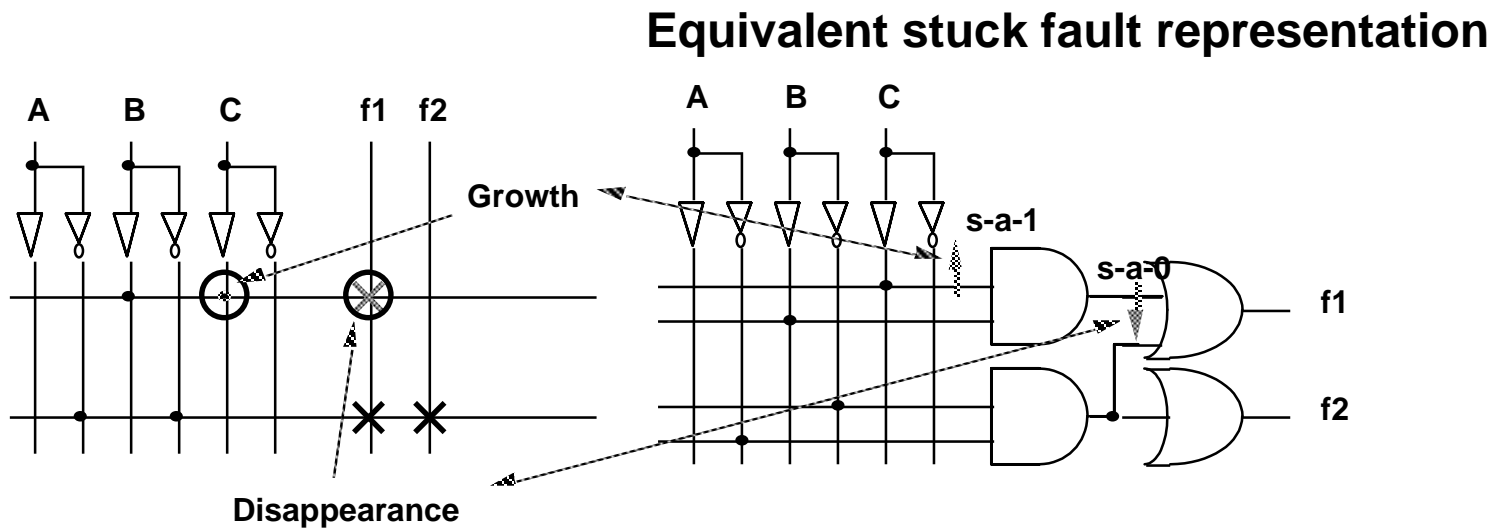
- **Adjacent cell coupling faults**
 - Pattern sensitivity between a pair of cells

PLA Faults

- **Stuck Faults**
- **Crosspoint Faults**
 - **Extra/Missing Transistors**
- **Bridging Faults**
- **Break Faults**

Missing Crosspoint Faults in PLA

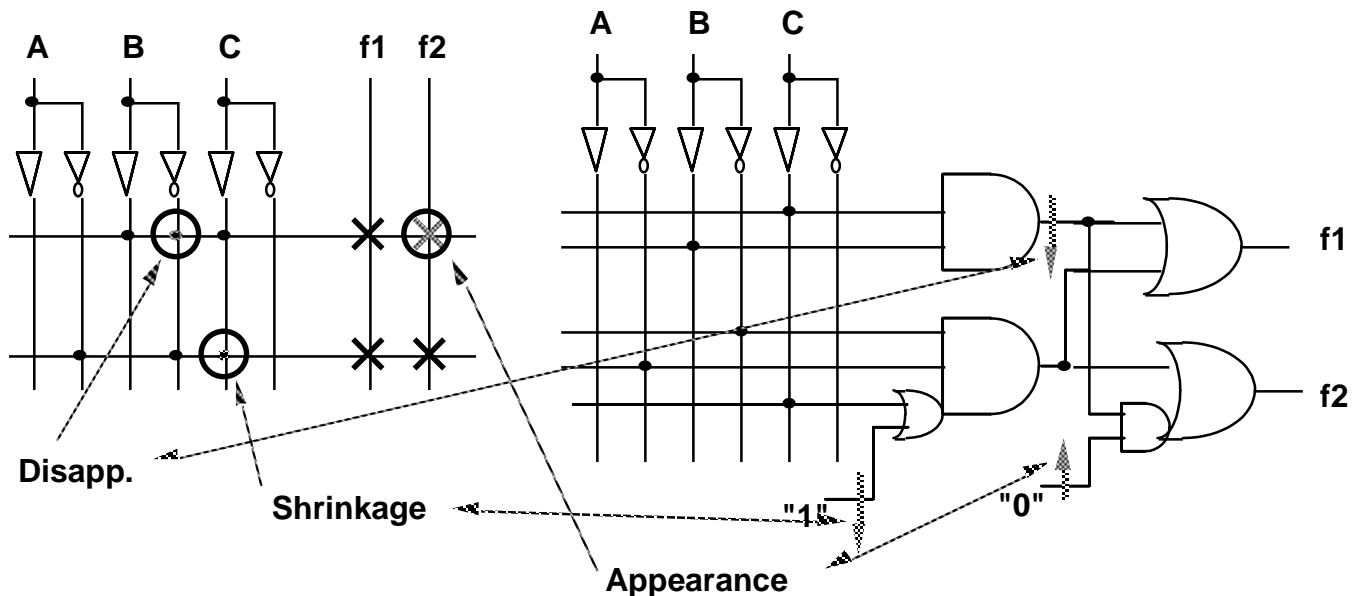
- **Missing crosspoint in AND-array**
 - Growth fault
- **Missing crosspoint in OR-array**
 - Disappearance fault



Extra Crosspoint Faults in PLA

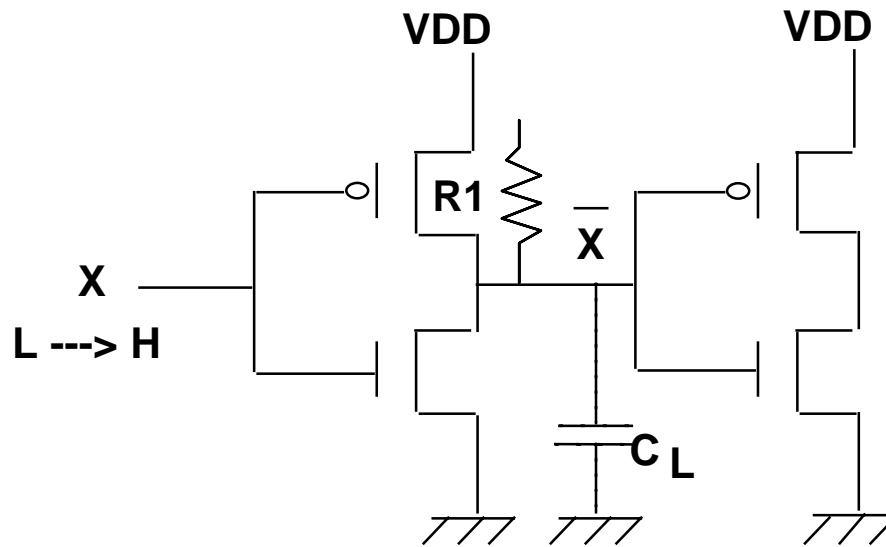
- **Extra crosspoint in AND-array**
 - Shrinkage or disappearance fault
- **Extra crosspoint in OR-array**
 - Appearance fault

Equivalent stuck fault representation

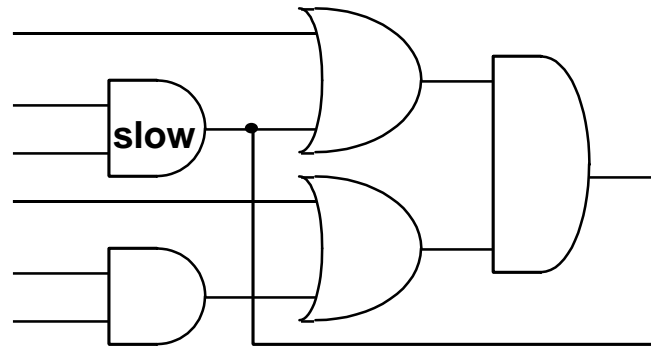


Gate-Delay-Fault

- **Slow to rise, slow to fall**
 - \bar{x} is slow to rise when channel resistance R1 is abnormally high



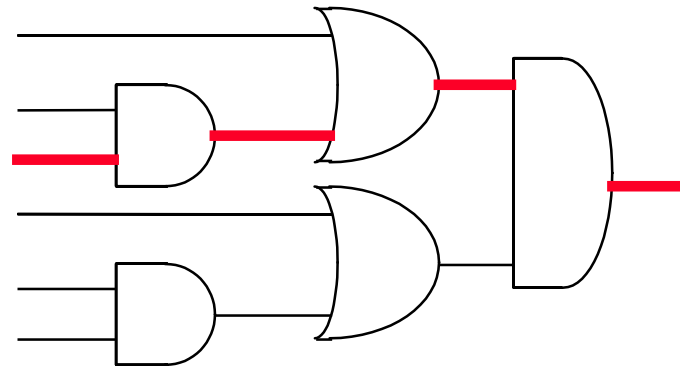
Gate-Delay-Fault



- **Disadvantage:**
Delay faults resulting from the sum of several small incremental delay defects may not be detected.

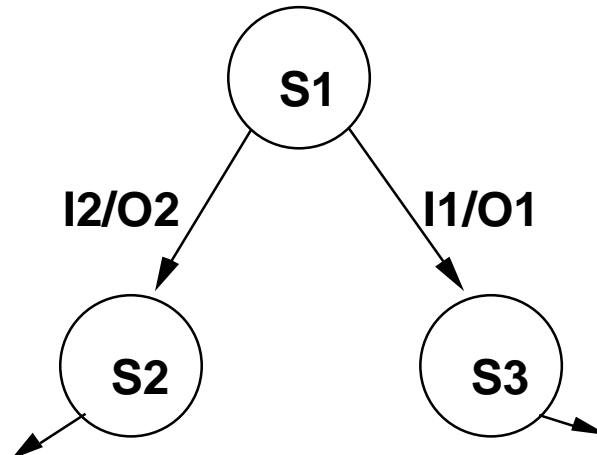
Path-Delay-Fault

- Propagation delay of the path exceeds the clock interval.
- The number of paths grows exponentially with the number of gates.



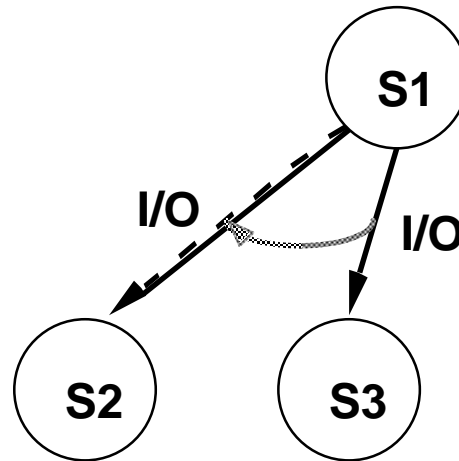
State Transition Graph

- Each state transition is associated with a 4-tuple: (source state, input, output, destination state)



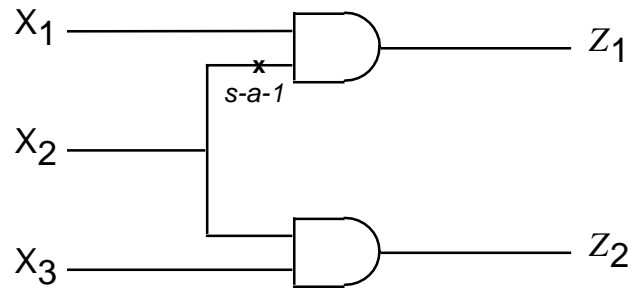
Single State Transition Fault Model

- A fault causes a single state transition to a wrong destination state.



Fault Detection

- A test (vector) t detects a fault f iff $z(t) \oplus z_f(t) = 1$
 - t detects $f \iff z_f(t) \neq z(t)$
- Example



$$Z_1 = X_1 X_2$$

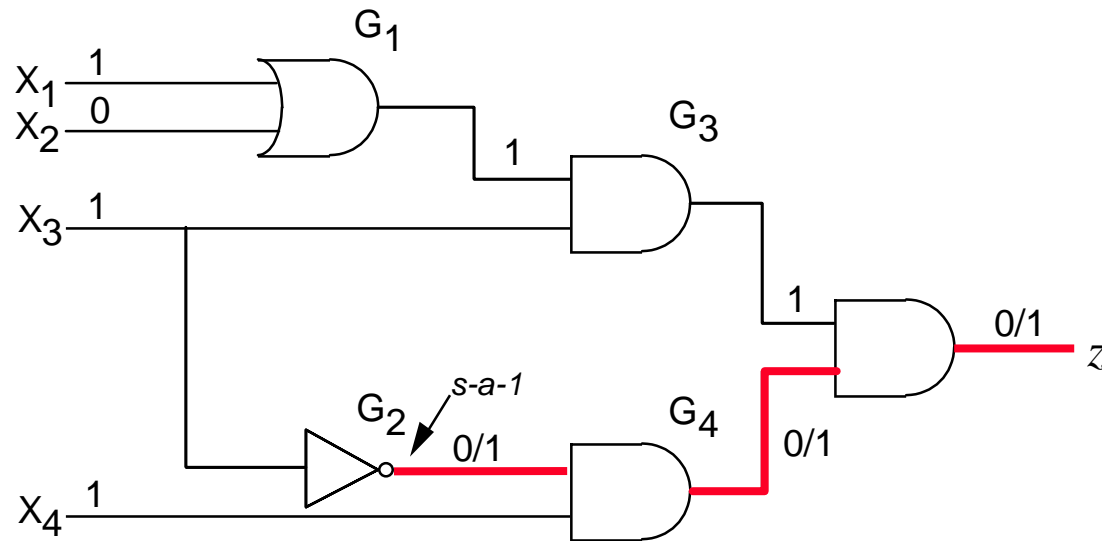
$$Z_2 = X_2 X_3$$

$$Z_{1f} = X_1$$

$$Z_{2f} = X_2 X_3$$

The test 001 detects f because $z_1(001)=0$ while $z_{1f}(001)=1$

Sensitization



$$z(1011)=0$$

$$z_f(1011)=1$$

1011 detects the fault f (G_2 stuck-at 1)

v/v_f : v = signal value in the fault free circuit

v_f = signal value in the faulty circuit

Sensitization

- **A test t that detects a fault f**
 - **Activates f (or generate a fault effect) by creating different v and v_f values at the site of the fault**
 - **Propagates the error to a primary output w by making all the lines along at least one path between the fault site and w have different v and v_f values**
- **A line whose value in the test changes in the presence of the fault f is said to be sensitized to the fault f by the test**
- **A path composed of sensitized lines is called a sensitized path**

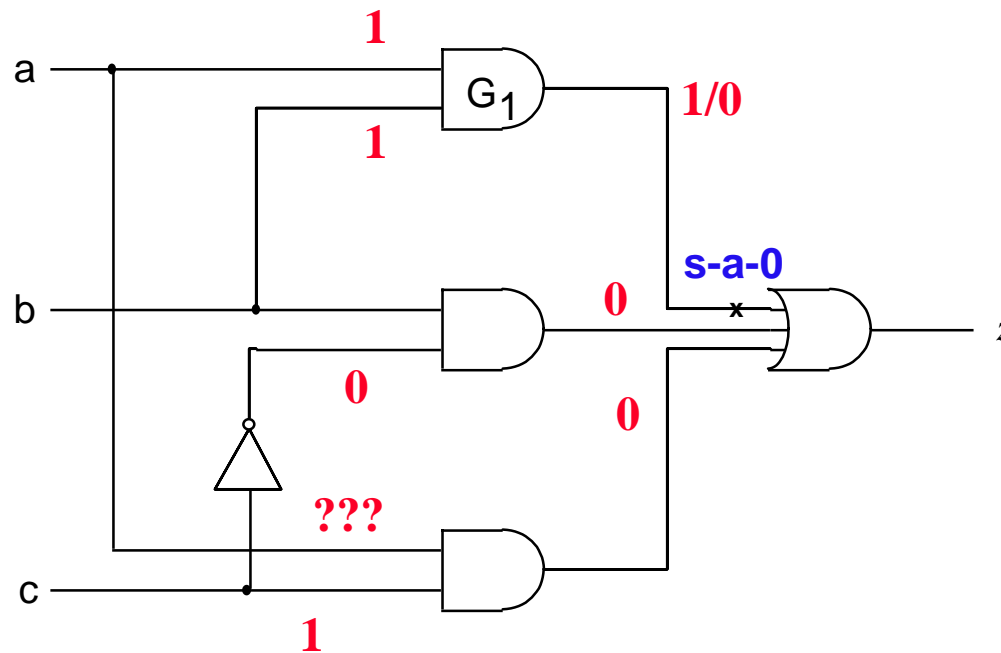
Detectability

- A fault f is said to be detectable if there exists a test t that detects f ; otherwise, f is an undetectable fault
- For an undetectable fault f

$$z_f(x) = z(x)$$

- No test can simultaneously activate f and create a sensitized path to a primary output

Undetectable Fault



- **G₁ output stuck-at-0 fault is undetectable**
 - Undetectable faults do not change the function of the circuit
 - The related circuit can be deleted to simplify the circuit

Test Set

- **Complete detection test set:** A set of tests that detect any detectable faults in a class of faults
- **The quality of a test set is measured by fault coverage**
- **Fault coverage:** Fraction of faults that are detected by a test set
- **The fault coverage can be determined by fault simulation**
 - >95% is typically required for single stuck-at fault model
 - >99.9% in IBM

Fault Equivalence

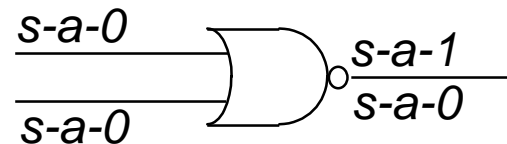
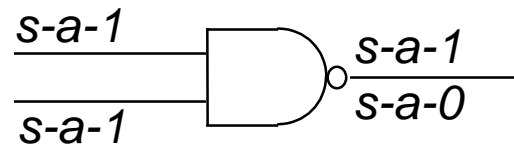
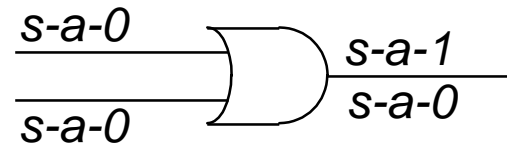
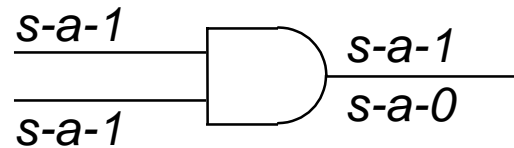
- **A test t distinguishes between faults α and β if $z_\alpha(t) \neq z_\beta(t)$**
- **Two faults, α & β are said to be equivalent in a circuit, iff the function under α is equal to the function under β for any input combination (sequence) of the circuit.**
 - $z_\alpha(t) = z_\beta(t)$ for all t
 - No test can distinguish between α and β
 - Any test which detects one of them detects all of them

Fault Equivalence

- AND gate: all *s-a-0* faults are equivalent
- OR gate: all *s-a-1* faults are equivalent
- NAND gate: all the input *s-a-0* faults and the output *s-a-1* faults are equivalent
- NOR gate: all input *s-a-1* faults and the output *s-a-0* faults are equivalent
- Inverter: input *s-a-1* and output *s-a-0* are equivalent
input *s-a-0* and output *s-a-1* are equivalent

Equivalence Fault Collapsing

- $n+2$ instead of $2n+2$ faults need to be considered for an n -input gate.



Fault Dominance

- A fault β is said to *dominate* another fault α in an irredundant circuit, iff every test (sequence) for α is also a test (sequence) for β .

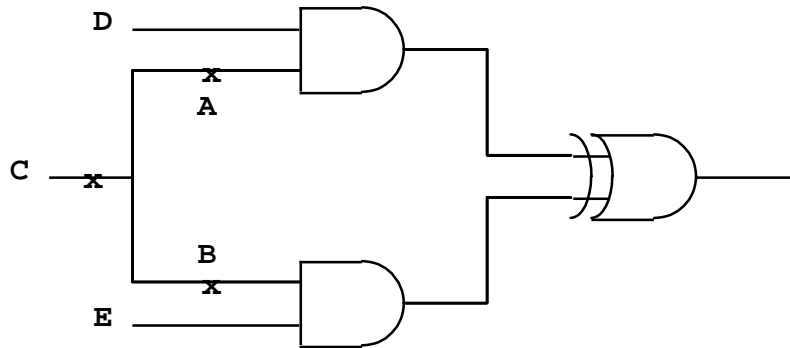
$$T_\alpha * T_\beta$$

- No need to consider fault β for fault detection

Fault Dominance

- **AND gate: Output *s-a-1* dominates any input *s-a-1***
- **NAND gate: Output *s-a-0* dominates any input *s-a-1***
- **OR gate: Output *s-a-0* dominates any input *s-a-0***
- **NOR gate: Output *s-a-1* dominates any input *s-a-0***
- **Dominance fault collapsing: The reduction of the set of faults to be analyzed based on dominance relation**

Fault Dominance



- **Detect A sa1:**

$$z(t) \oplus z_f(t) = (CD \oplus CE) \oplus (D \oplus CE) = D \oplus CD = 1$$

$$\Rightarrow (C = 0, D = 1)$$

- **Detect C sa1:**

$$z(t) \oplus z_f(t) = (CD \oplus CE) \oplus (D \oplus E) = 1$$

$$\Rightarrow (C = 0, D = 1) \text{ or } (C = 0, E = 1)$$

C sa1 --> A sa1

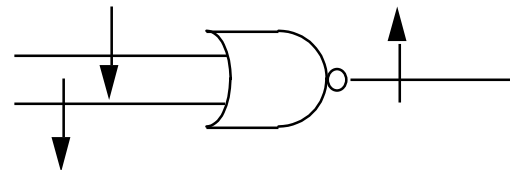
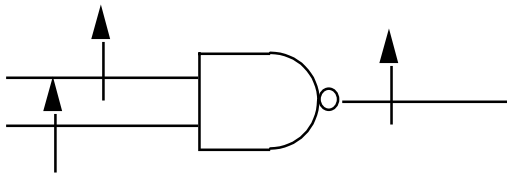
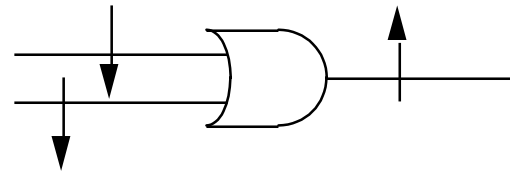
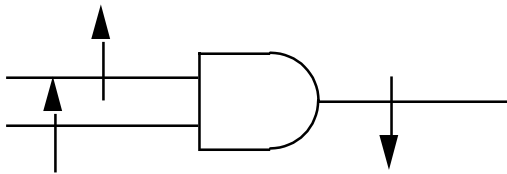
- **Similarly C sa1 --> B sa1**

C sa0 --> A sa0

C sa0 --> B sa0

Fault Collapsing

- For each n -input gate, we only need to consider $n+1$ faults



Prime Fault

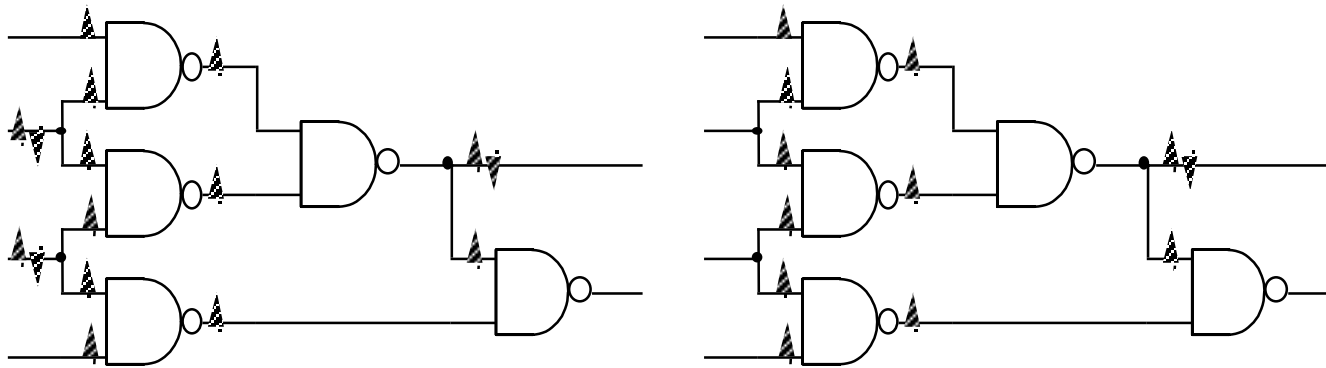
- α is a prime fault if every fault that is dominated by α is also equivalent to α
- **Representative Set of Prime Fault (RSPF)**
 - A set that consists of exactly one prime fault from each equivalence class of prime faults
 - True minimal RSPF is difficult to find

Why Fault Collapsing?

- **Memory & CPU-Time saving**
 - ⇒ **To ease the burden for test generation and fault simulation in testing**

# of total faults	# of equivalent faults	# of prime faults
1	60%	40%

Fault Collapsing for a Combinational Circuit

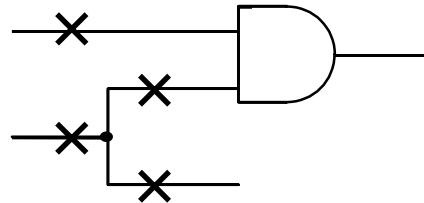


• 30 total faults \Rightarrow 12 prime faults

Checkpoint Theorem

- **Primary-input & Fanout-Branches**

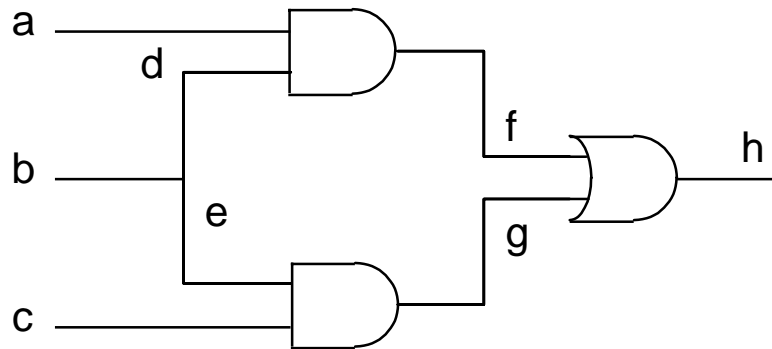
⇒ a *sufficient and necessary* set of checkpoints in irredundant combinational circuits



- In fanout-free combinational circuits, primary inputs are the set of checkpoints
- **Any test set which detects all signal (multiple) stuck faults on check points will detect all signal (multiple) stuck faults**

Fault Collapsing

- The set of checkpoint faults can be further collapsed by using equivalence and dominance relation
- Example



- 10 checkpoint faults
- $a \text{ s-a-}0 \leftrightarrow d \text{ s-a-}0$, $c \text{ s-a-}0 \leftrightarrow e \text{ s-a-}0$
 $b \text{ s-a-}0 \rightarrow d \rightarrow 0$, $b \text{ s-a-}1 \rightarrow d \rightarrow 1$
- 6 tests are enough