Fault Modeling

Logical Fault Models

- Logical faults represents the effect of physical faults on the behavior of the modeled system
  - Faults that affects the logic function
  - Delay faults

Logical Fault Models

- What do we gain by modeling physical faults as a logical faults?
- First
  - Fault analysis become logical
  - Complexity is greatly reduces
- Second
  - Some logical faults are technology independent, same faults model is applicable to many technologies.
  - Testing & diagnosis method developed for such faults remain valid despite changes in technology
- Third
  - Test derived for the logical faults can be used for physical faults, whose effect on circuit behavior is not completely known.

Explicit & Implicit Faults

- Explicit fault
  - A fault universe in which every fault is individually identified and hence the faults to be analyzed can be explicitly enumerated.

- Implicit fault
  - A fault universe by collectively identifying the faults of interest – typically by defining their characteristic properties

Structural & Functional Faults

- Structural fault
  - In conjunction with structural model
- Functional fault
  - In conjunction with functional model

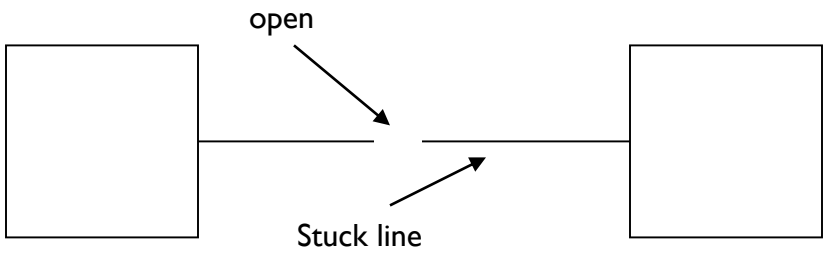
Frequent Testing Strategy

- Single fault assumption
- Many physical faults can affect an area containing several components
- Newly manufactured system prior to testing, multiple faults are likely to exist.
- Multiple fault - If the testing experiments does not detect the every single faults, then the circuit may contain one of the undetected faults at any time, and the occurrence of a second single faults between two testing experiments creates a multiple fault.

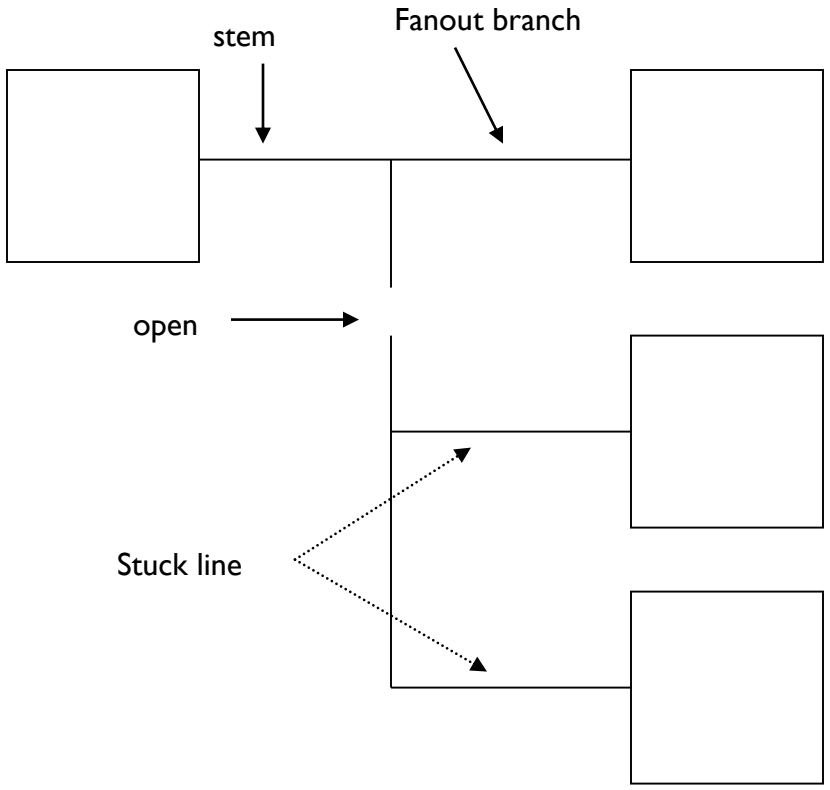
Stuck at Faults

- Faults affecting interconnections are **short & open**
- Short
  - By connecting points not intended to be connected
- Open
  - Breaking between connections
- A short between ground or supply and a signal line can make the signal remain at a fixed logic value v (v ∈ 0,1) and it is denoted by s-a-v
- A short between two signal lines usually creates a new logic function – **Bridging faults**

Stuck faults caused by open



Single stuck faults



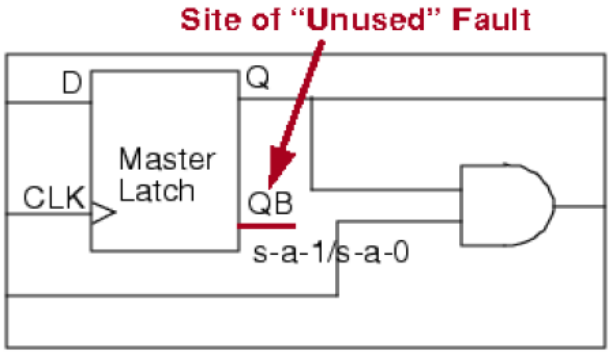
Multiple stuck faults

Fault Detection

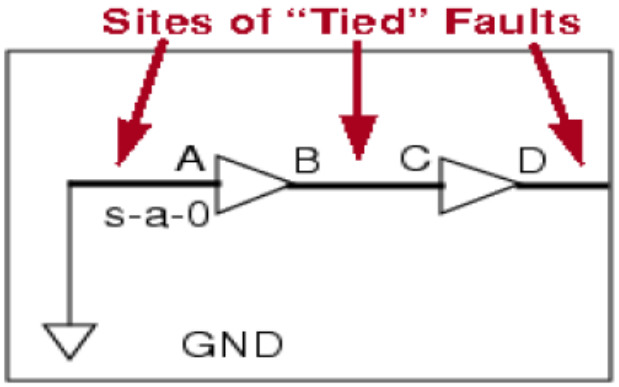
- Faults detection works by comparing the response of a known-good version of the circuit to that of the actual circuit, for a given stimulus set.
- A fault exists if there is any difference in the responses.
- One common fault detection approach is path sensitization.
- The path sensitization method, which is used by the tool to detect stuck-at faults, starts at the fault site and tries to construct a vector to propagate the fault effect to a primary output

Fault Classes

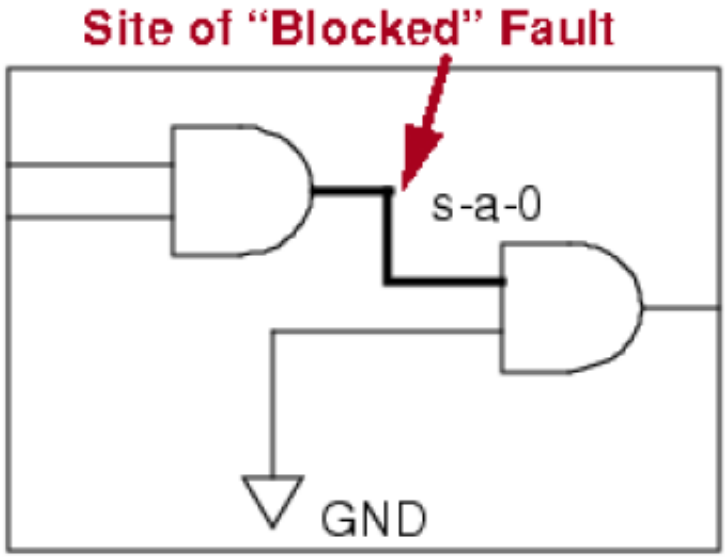
- **Untestable (UT)**
- Untestable (UT) faults are faults for which no pattern can exist to detect them. Untestable faults cannot cause functional failures, so the tools exclude them when calculating test coverage.
- The following are the Untestable Faults.
  - **Un-used (UU)**
    - The unused fault class includes all faults on circuitry unconnected to any circuit observation point and faults on floating primary outputs.



- **Tied (TI)**
- The tied fault class includes faults on gates where the point of the fault is tied to a value identical to the fault stuck value.
- In below fig, node A is connected to GND and there assumed to be a s-a-0 fault at A.



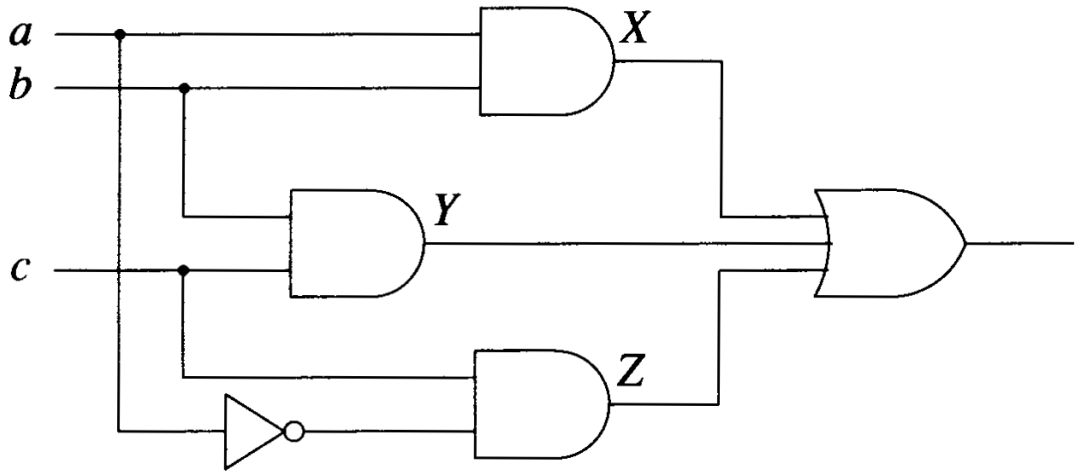
- **Blocked (BL)**
- The blocked fault class includes faults on circuitry for which tied logic blocks all paths to an observable point



Redundant Fault

- The redundant fault class includes faults that test generator considers undetectable.
- After the test pattern generator exhausts all patterns, it performs a special analysis to verify that the fault is undetectable under any conditions.

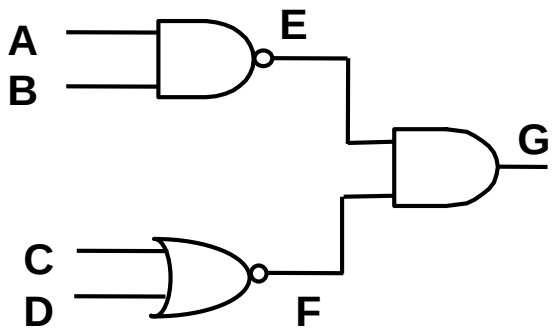
ab+bc+a'c



Undetectable if Y is s-a-0

Fault Modeling

- The effects of physical defects
- Most commonly used fault model: Single stuck-at fault



A s-a-1 B s-a-1 C s-a-1 D s-a-1
A s-a-0 B s-a-0 C s-a-0 D s-a-0

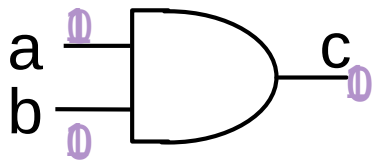
E s-a-1 F s-a-1 G s-a-1
E s-a-0 F s-a-0 G s-a-0

14 faults

Fault Coverage (FC)

$$FC = \frac{\text{\# faults detected}}{\text{\# faults in fault list}}$$

Example:



6 stuck-at faults
(a_0,a_1,b_0,b_1,c_0,c_1)

Test	faults detected	FC
{(0,0)}	c_1	16.67%
{(0,1)}	a_1,c_1	33.33%
{(1,1)}	a_0,b_0,c_0	50.00%
{(0,0),(1,1)}	a_0,b_0,c_0,c_1	66.67%
{(1,0),(0,1),(1,1)}	all	100.00%

Fault Modeling

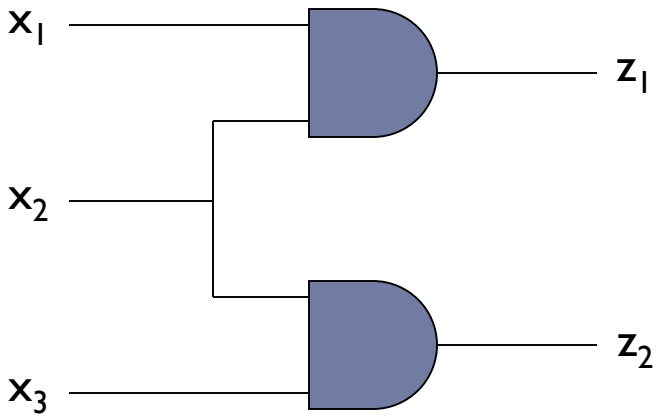
- Single stuck at faults
  - S @ 0
  - S @ 1
- Bridge fault
- Multiple stuck at faults

Single stuck at fault

Fault free expression

Z1 = x1.x2

Z2 = x2.x3



Single stuck at fault - s@0

Fault free expression

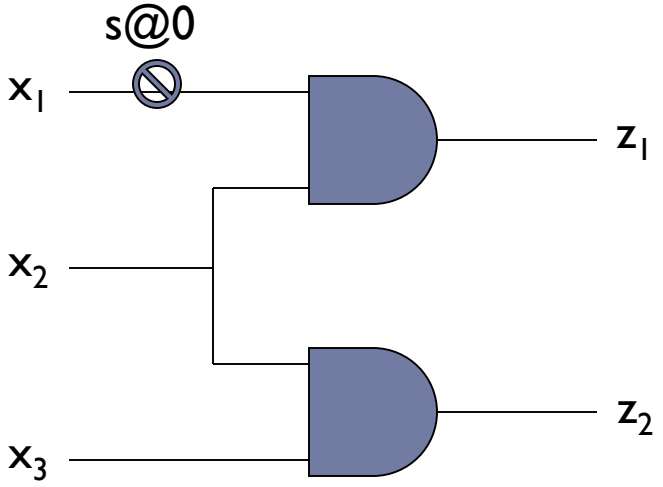
Z1 = x1.x2

Z2 = x2.x3

Faulty expression

Z1 = 0

Z2 = x2.x3



Single stuck at fault - s@1

Fault free expression

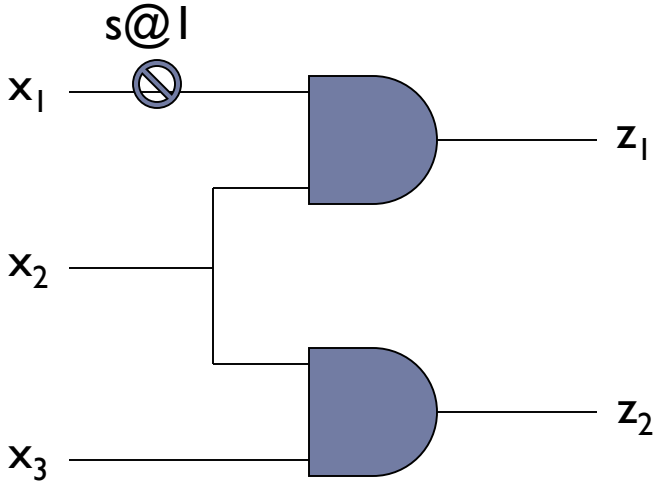
Z1 = x1.x2

Z2 = x2.x3

Faulty expression

Z1 = x2

Z2 = x2.x3

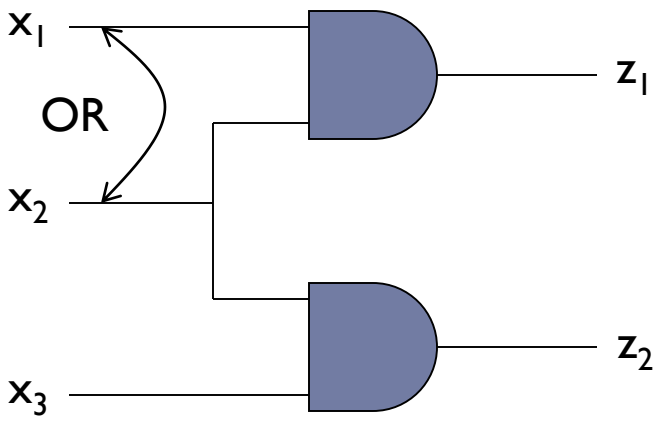


Single stuck at fault - bridge fault

Fault free expression

Z1 = x1.x2

Z2 = x2.x3

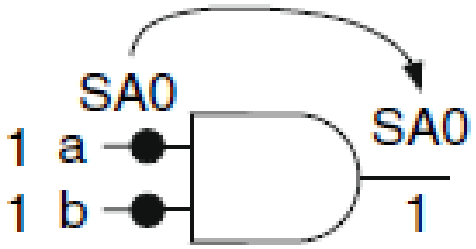


Faulty expression

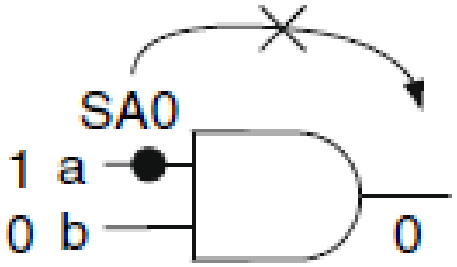
Z1 = (x1+x2).x2 = x2

Z2 = x2.x3

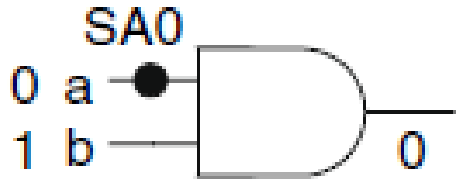
Fault Simulation Terminologies



a:SA0 propagates



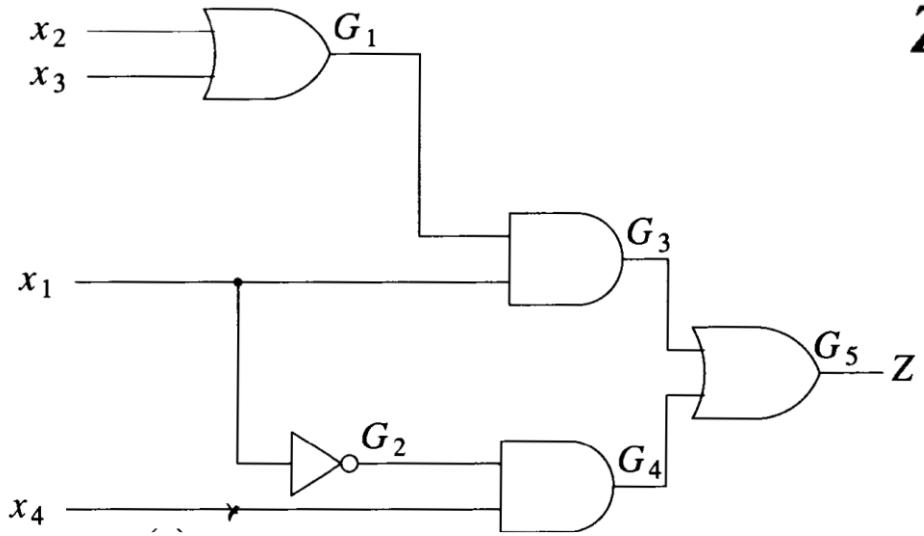
a:SA0 is blocked



a:SA0 is not activated

Fault Detection

- For a single output circuit, a test t that detects a fault f makes Z(t) = 0 and $Z_f(t)$ = 1 or vice-versa.
- Thus set of all tests that detect f is given by the solutions of the equation
  - Z(x) XOR $Z_f(x)$ = 1



$$Z = (x_2+x_3)x_1 + \bar{x}_1 x_4$$

Let f be x_4 s-a-0.

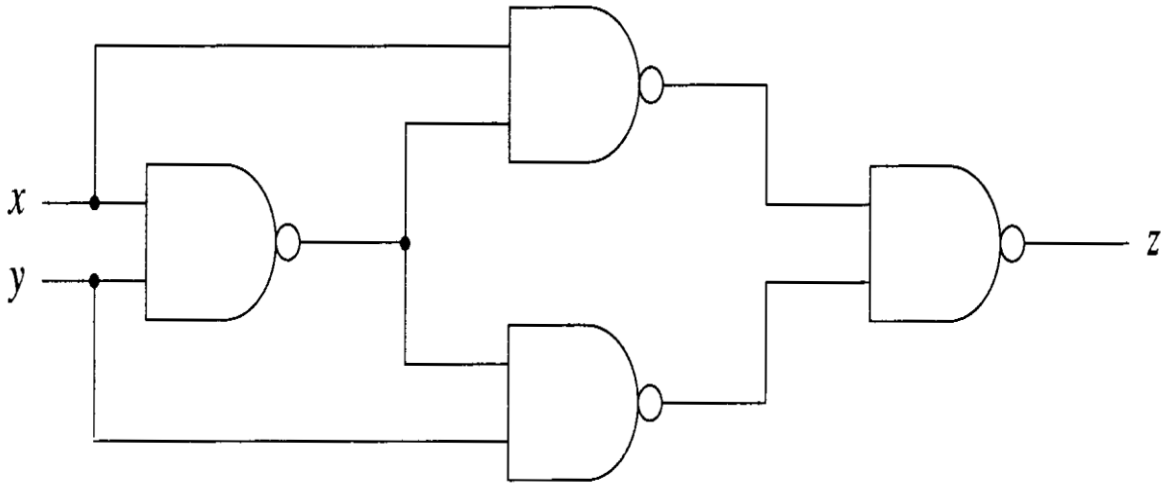
$$Z_f = (x_2+x_3)x_1$$

$$\bar{x}_1 x_4 = 1.$$

Thus any test in which $x_1 = 0$ and $x_4 = 1$ is a test for f. any of the four tests (0001, 0011, 0101, 0111) that detect f.

Fault Equivalence

- **Fault equivalence:** Two faults f1 and f2 are equivalent if all tests that detect f1 also detect f2.
- If faults f1 and f2 are equivalent then the corresponding faulty functions are identical.

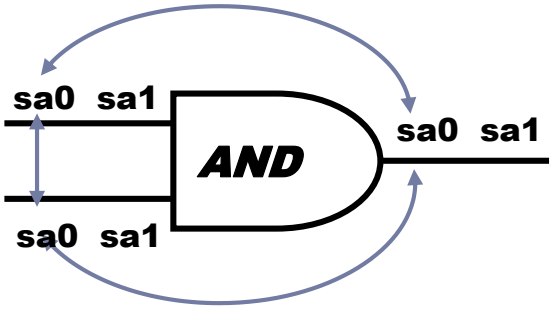


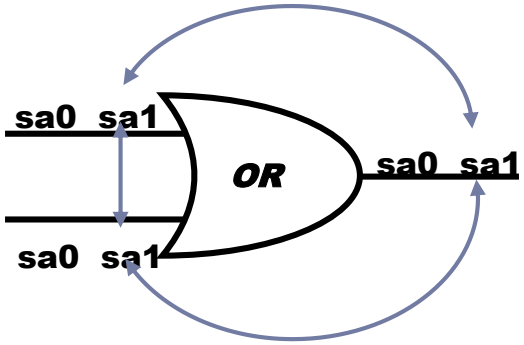
- AND bridging fault of x, y is functionally equivalent to z s-a-0

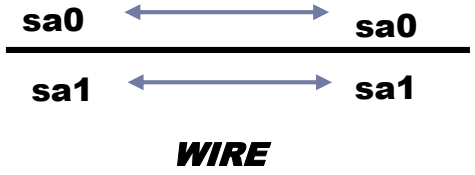
Fault Collapsing

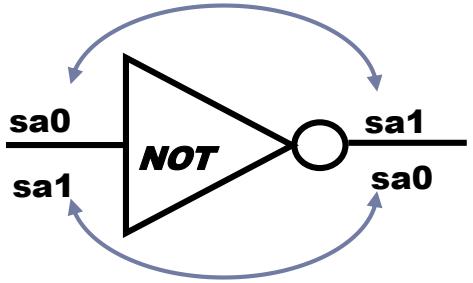
- With any n input gate we have 2(n+1) stuck at faults
  - For a 2 input nand gate all the input s-a-0 is equivalent to output s-a-1.
- This type of reduction of set of faults to be analyzed based on the equivalence relations is called equivalence fault collapsing

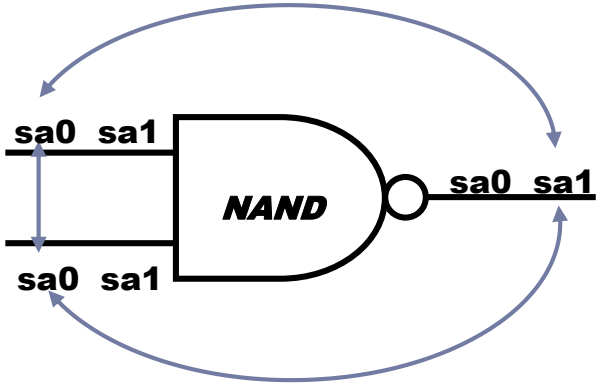
Equivalence Rules

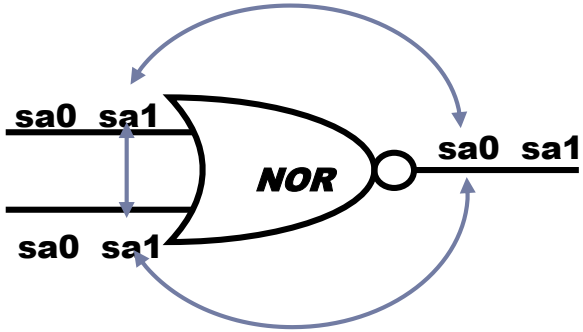


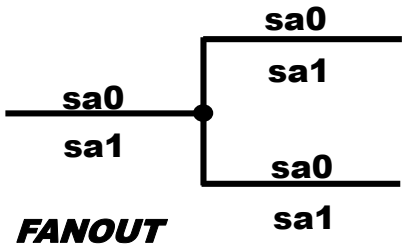






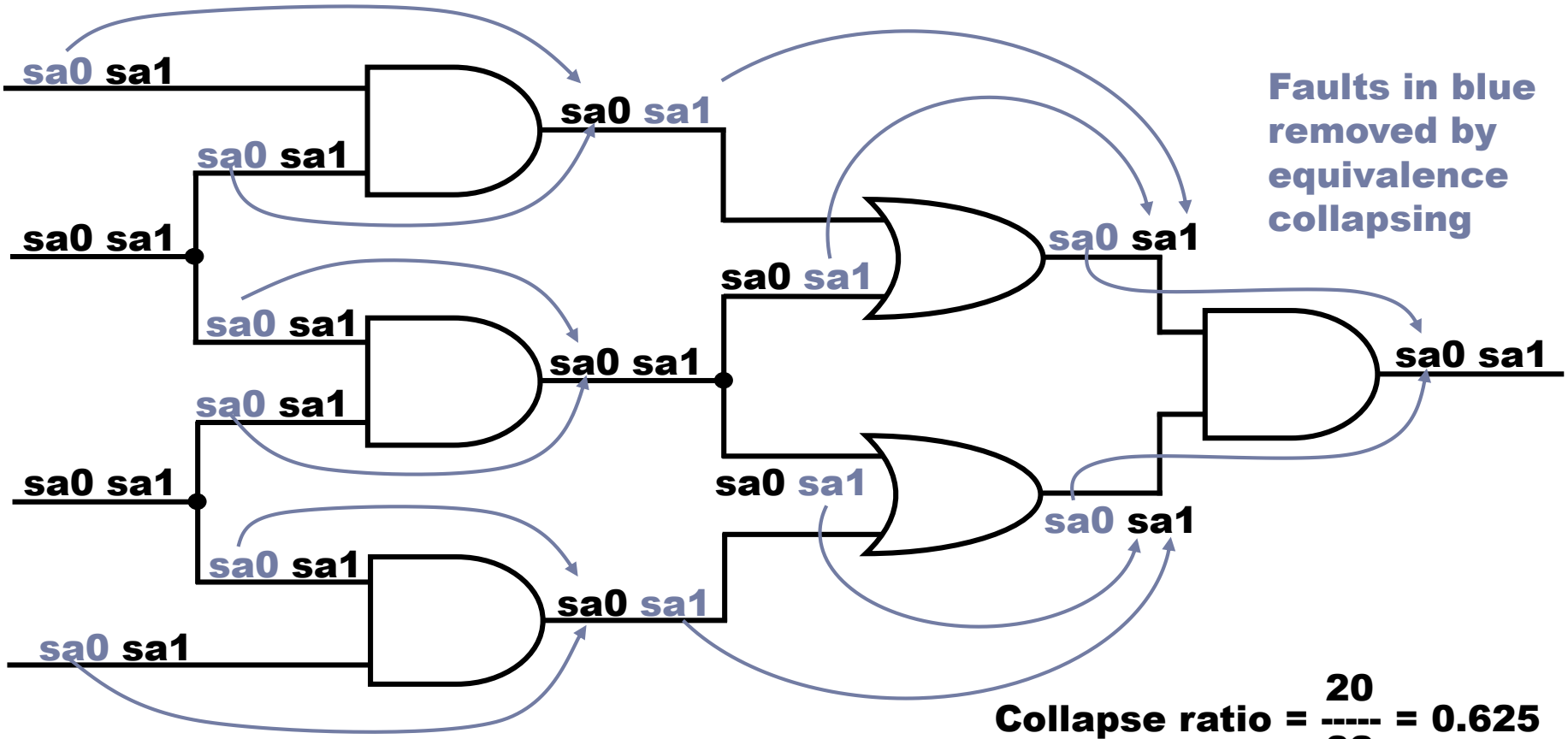






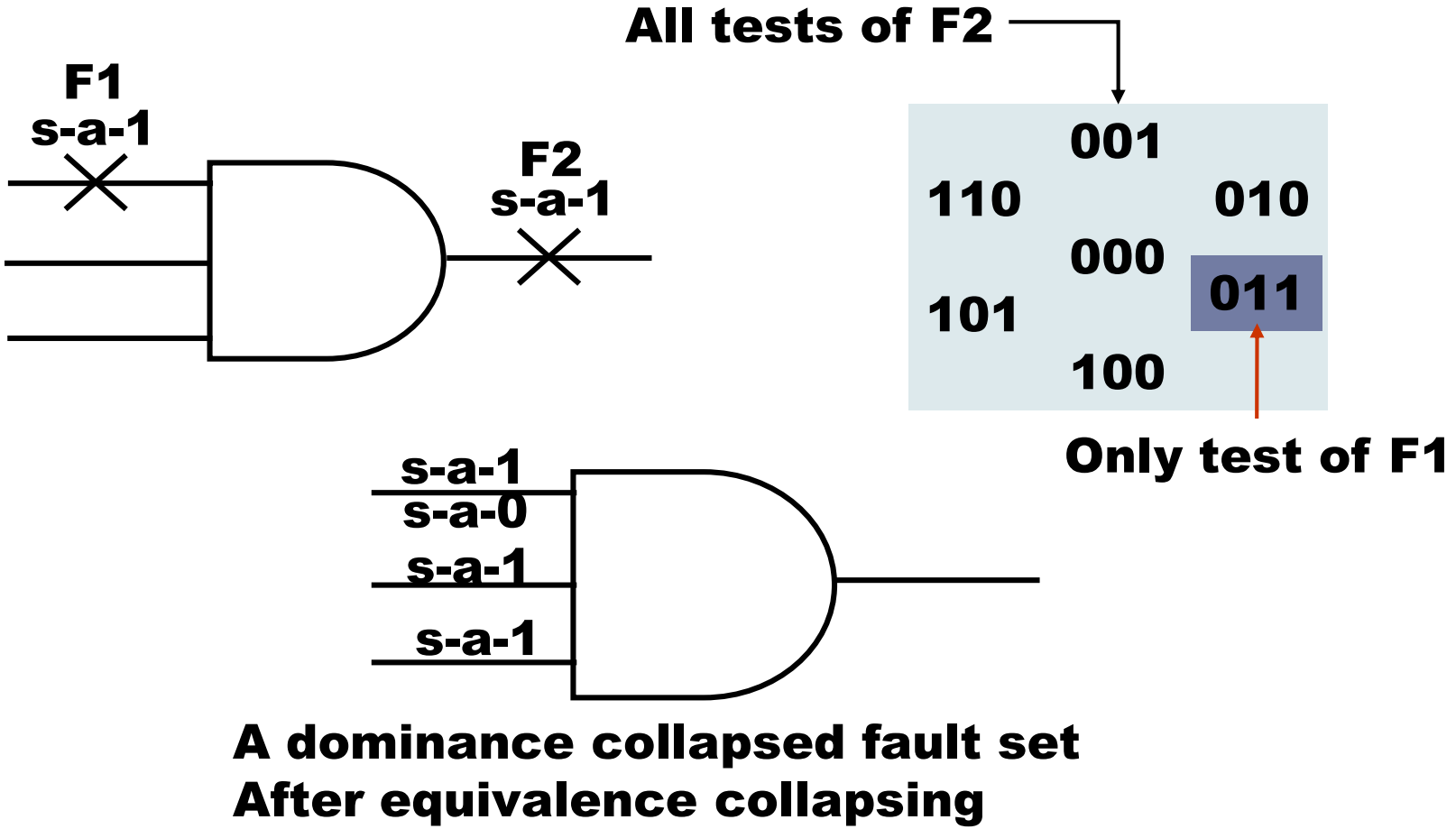
- Need to consider only n+2 stuck at faults.

Equivalence Example



Dominance Example

In a tree circuit (without fanouts) PI faults form a dominance collapsed fault set.



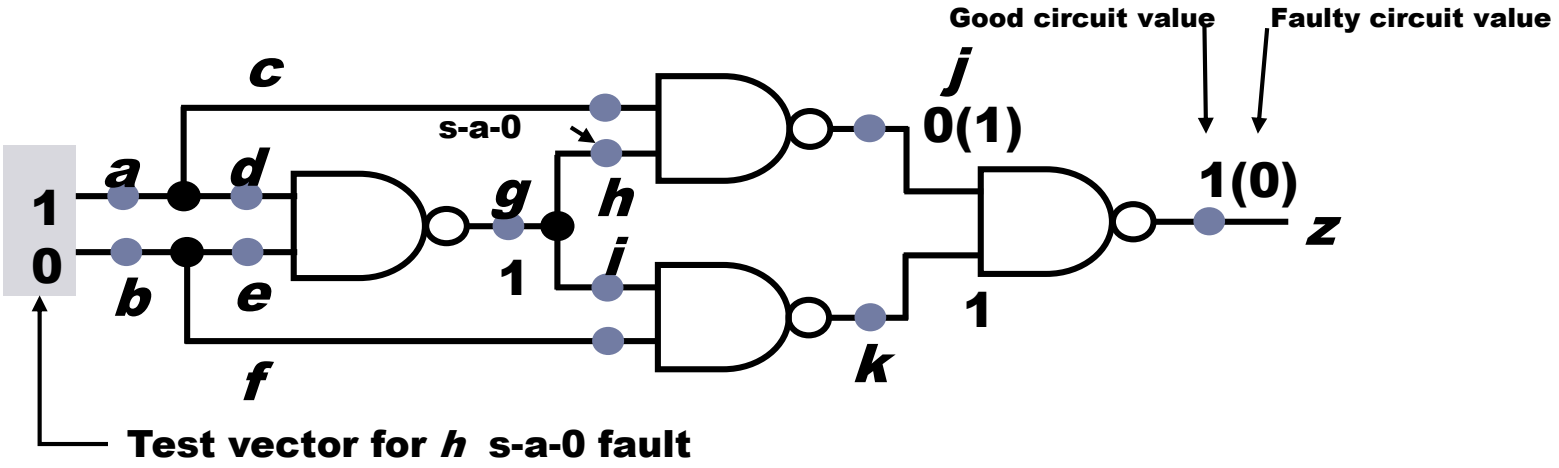
Fault Dominance

- If all tests of fault F2 detects another fault F1, then F1 dominates F2.
- **Dominance fault collapsing:** If fault F2 dominated by F1, then F2 is removed from the fault list.
- When dominance fault collapsing is used, it is sufficient to consider only the input faults of Boolean gates
- If two faults dominate each other then they are equivalent.

Single Stuck-at Fault Model

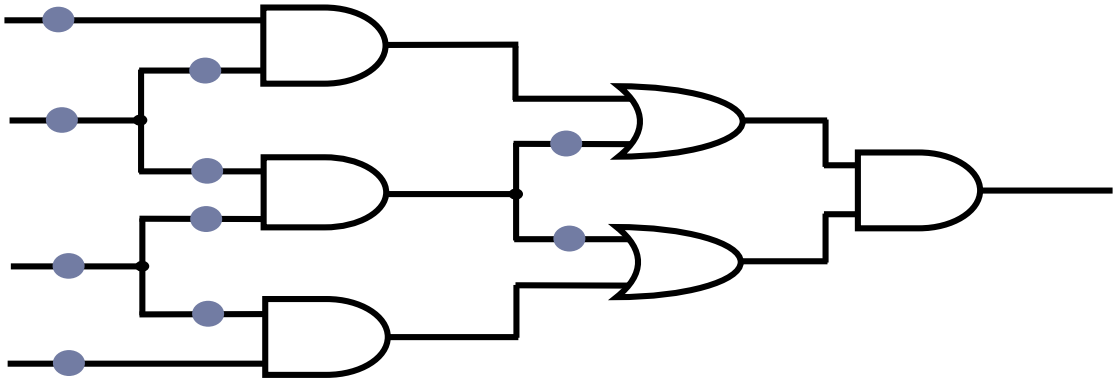
- Three properties define a single stuck-at fault
  - Only one line is faulty
  - The faulty line is permanently set to 0 or 1
  - The fault can be at an input or output of a gate

- Example: XOR circuit has 12 fault sites (●) and 24 single stuck-at faults



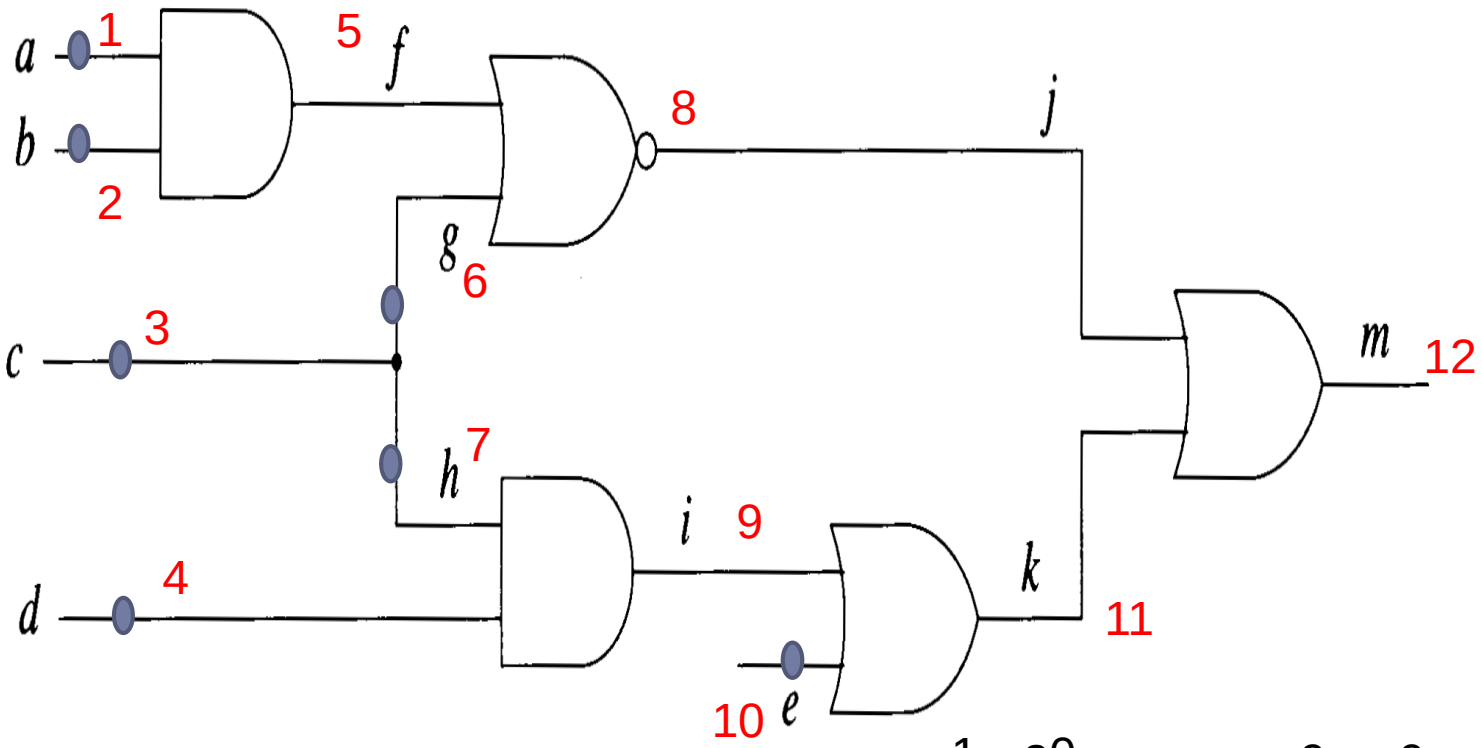
Checkpoints

- Primary inputs and fanout branches of a combinational circuit are called ***checkpoints*.**
- **Checkpoint theorem:** A test set that detects all single (multiple) stuck-at faults on all checkpoints of a combinational circuit, also detects all single (multiple) stuck-at faults in that circuit.



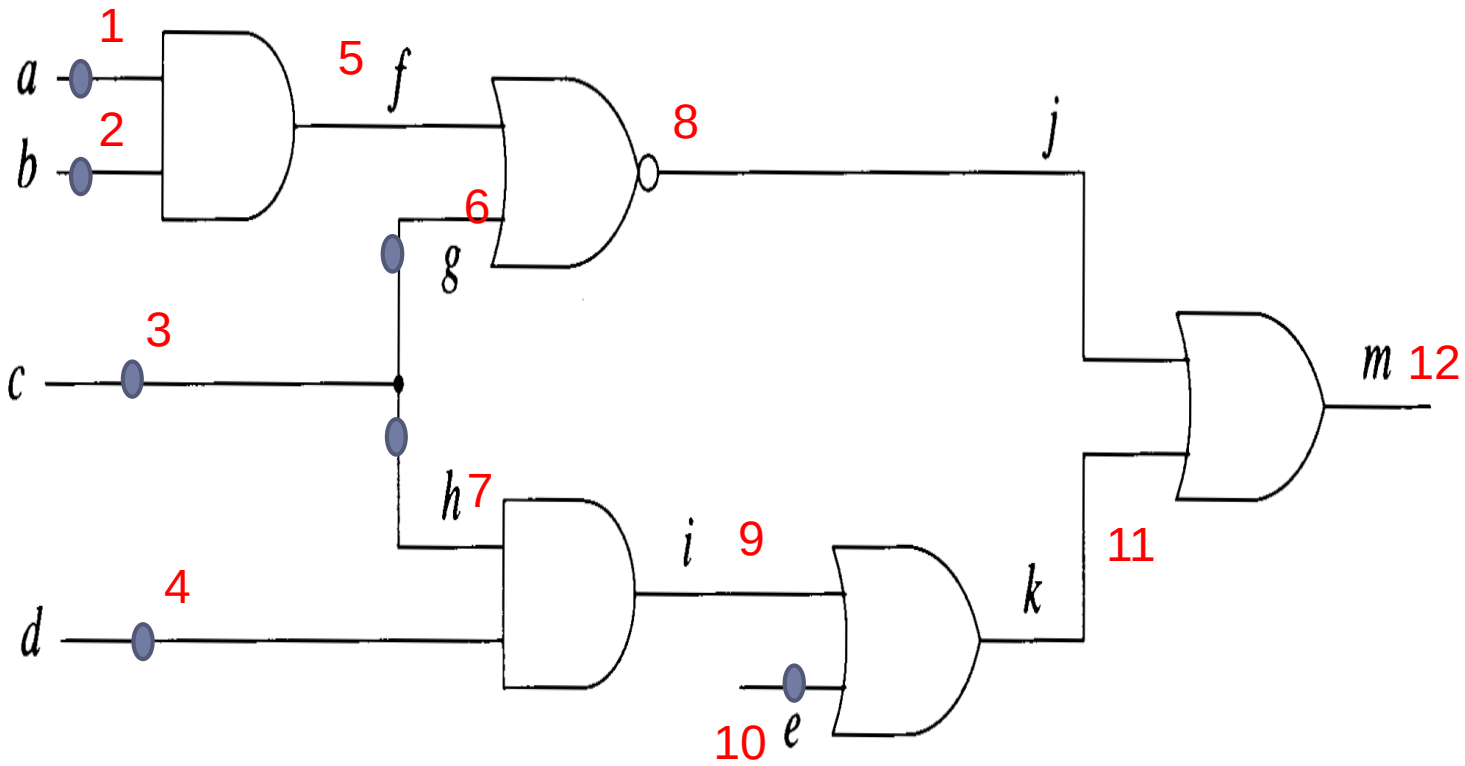
Total fault sites = 16

Checkpoints (●) = 10



- 12 fault sites – 24 stuck at faults
- 7 Checkpoints

1. a0
2. a1
3. b0
4. b1
5. c0
6. c1
7. d0
8. d1
9. e0
10. e1
11. g0
12. g1
13. h0
14. h1



- Input *a* s-a-0 and *b* s-a-0 are equivalent – remove *b* s-a-0
- Input *d* s-a-0 equivalent to *h* s-a-0, remove it
- *g* s-a-1 equivalent to *f* s-a-1, which dominates *a* s-a-1, so remove it
- *e* s-a-1 equivalent to *i* s-a-1, which dominates *h* s-a-1, so remove it
- *Number of faults reduced from 24 to 10*

Assignment

- Current (or IDDQ) and Very Low-level Voltage (VLV) test