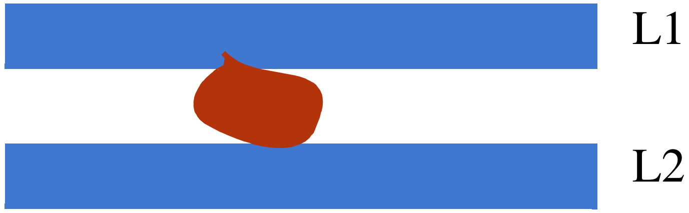
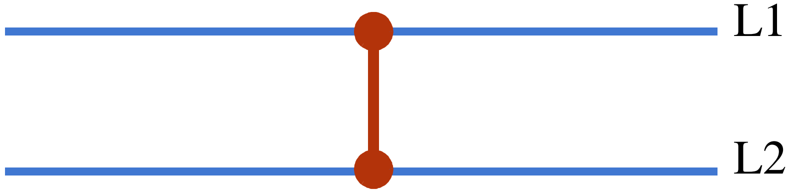
Failures in Integrated Circuits

Failure mode is used in reference to the manifestation of a ***defect*** at the electrical level.

Failure modes are modeled as ***faults*** at logic or behavioral level of abstraction

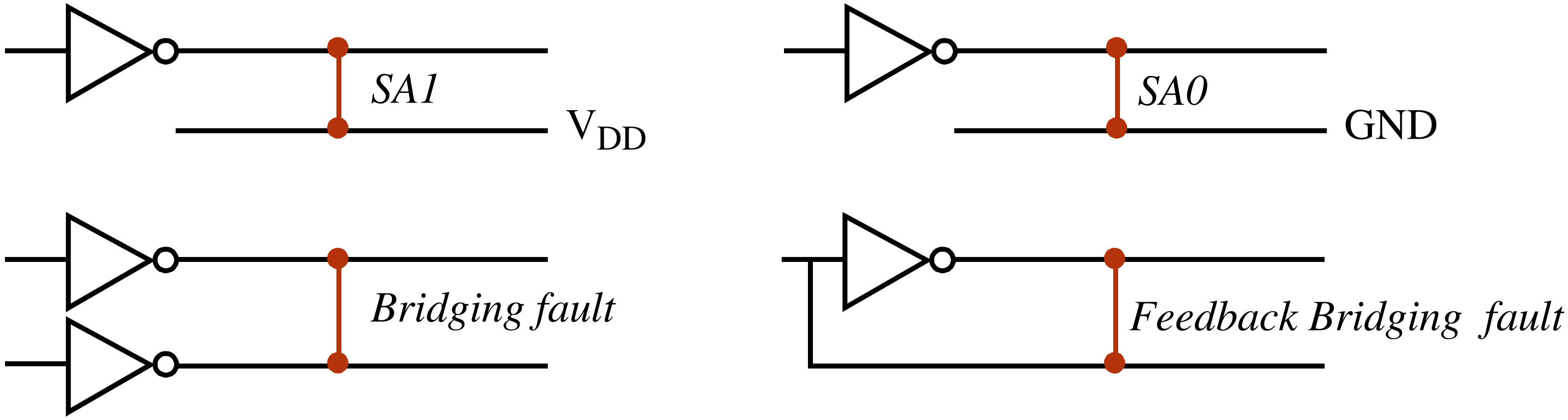


Physical defect



Physical model

At the logic level, failure mode can be interpreted in different ways.



Failure Mechanisms

Defects are due to ***failure mechanisms***.

It is important to derive the principal failure mechanisms of a process.

These mechanisms are tied to variations in the fabrication process:

- Random fluctuations in the actual environment, e.g.,
  Turbulent flow of gases used for diffusion and oxidation.
- Inaccuracies in the control of the furnace.
- Variations in the physical and chemical parameters of the material, e.g.,
  Fluctuation in the density and viscosity of the photoresist.
  Water and gas contaminants.

Extra or Missing Materials

Can be caused by dust particles on the mask, wafer surface or processing chemicals, e.g. photoresist.

Can Result in ***unwanted material*** or ***unwanted etching*** of the material.

Physical Defects

Gate Oxide Shorts

- ❐ Pinhole defects are common thin-ox defects.

  Caused by chemical contamination, nitride cracking at oxidation, crystal defects ...

- ❐ A GOS can be created in post fabrication procedures and operational conditions

  Electric field stress, ESD (electro-static discharge), time dependent electric breakdown ***(TDDB)***.

Electromigration

- ❐ One of the major failure mechanisms in interconnects.
- ❐ Scaling is reducing the ***Mean Time To Failure (MTTF)***.

  Proportional to metal width and thickness, inversely proportional to current density.

- ❐ Can cause voids in metal lines, splinters between same or next level metal lines.

Physical Defects

Shorting Defects

- ▲Shorts can occur between metal lines and V_{DD} or V_{SS}.
- ▲Between two nodes or subnets as ***bridging faults***.
  - Oxide surface conduction.
  - Lateral charge spreading
  - Electromigration.
- ▲Gate-Oxide Shorts (discussed earlier)
- ▲Via punch-through, parasitic transistor leakage and defect pn junctions.

Open Defects

- ▲Defined as opens and breaks by missing conducting material or by extra insulating material.
- ▲Opens are more difficult to detect in CMOS circuits.
- ▲Faults can be ***time-dependent***.
  - Narrow (tunneling) opens.
  - Opens where coupling capacitance interactions and leakage currents effect the state of the node.

Why Fault Models?

- I/O functions tests inadequate for manufacturing (functionality versus interconnect and component testing).
- Real defects too numerous and often not analyzable.
- A fault model identifies targets for testing
- A fault model makes analysis possible
- Effectiveness measurable by experiments.

Defect: An unintended difference between the implemented hardware and its intended function.

Error: A wrong output signal produced by a defective system.

Fault: It is a logic level abstraction of a physical defect.

- Used to describe the change in the logic function of a device caused by the defect.
- Fault abstractions reduce the number of conditions that must be considered in deriving tests.

Fault Model: A collection of faults, all of which are based on the same set of assumptions concerning the nature of defects.

Fault Models

- Assertion Fault
- Behavioral Fault
- Branch Fault
- Bridging
- Bus Fault
- Cross-point Fault
- Defect-oriented Fault (physical level faults, bridging, stuck-open, I_{DDQ}).
- Delay Fault (transition, gate-delay, line-delay, segment-delay, path-delay).
- Functional Fault
- Gate-delay Fault
- Hyperactive Fault
- Initialization Fault
- Instruction Fault

Fault Models

- Intermittent Fault
- Line-delay Fault
- Logical Fault (often Stuck-at)
- Memory Fault (SA0/1, pattern sensitive, cell coupling faults).
- Multiple Fault
- Non-classical Fault (not stuck-at, stuck-open or stuck-on for CMOS).
- Oscillation Fault (or star-faults, bridging faults in combo logic).
- Parametric Fault (changes the values of electrical parameters).
- Path-delay Fault
- Pattern Sensitive Fault
- Permanent Fault
- Physical Fault

Fault Models

- Pin Fault (SA faults on the signal pins of all modules in the circuit).
- PLA Fault
- Potentially Detectable Fault (a subset of the initialization faults).
- I_{DDQ} Fault
- Race Fault
- Redundant Fault
- Segment-delay Fault
- Structural Fault
- Stuck-at Fault
- Stuck-open and Stuck-short Fault
- Transistor Fault (Stuck-open and Stuck-short faults)
- Transition Fault
- Untestable Fault

Logical Faults

Logical faults are used to represent *physical faults*.

Simplifies the fault analysis process and reduces the number of faults.

A logical fault changes (usually simplifies) the logic function of the circuit.

Structural faults modify the interconnection among the components.

Functional faults change the truth table of a circuit or component.

Most of our discussion is based on the *single* fault assumption.

However, multiple faults cannot be ignored because:

- Some physical faults can generate multiple logical faults.
- Testing that does not guarantee 100% fault coverage does not detect some faults and, in certain circuit configurations, these faults can mask faults that are tested.

Fortunately, in most cases, multiple faults are detected by single fault tests.

Single Stuck-at Fault (SSF)

Start with the circuit represented as a ***netlist*** of Boolean gates.

Assumes faults only affect the ***interconnection*** between gates **(*structural*)**.

Short and open defects usually cause the signal net or line to remain at a fixed voltage level.

The corresponding logical fault consists of a signal being ***Stuck-at-0 (SA0)*** or ***Stuck-at-1 (SA1)***.

A circuit with ***n*** lines can have 3^n - 1 ***multiple*** stuck line combinations since each line can be in one of SA0, SA1 and fault-free.

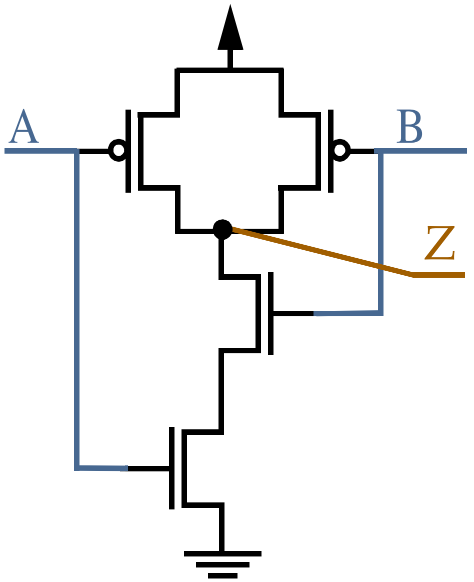
From a practical standpoint, this number is too large.

On the other hand, an ***n***-line circuit can have at most ***2n*** SSF faults.

This number can be further reduced through ***fault collapsing*** discussed soon.

SSF Fault Example

How many SSF faults can occur on an *n-input* NAND gate?



Inputs	Fault-Free	Faulty Response					
AB	Response	A/0	B/0	Z/0	A/1	B/1	Z/1
00	1	1	1	**0**	1	1	1
01	1	1	1	**0**	**0**	1	1
10	1	1	1	**0**	1	**0**	1
11	0	**1**	**1**	0	0	0	**1**

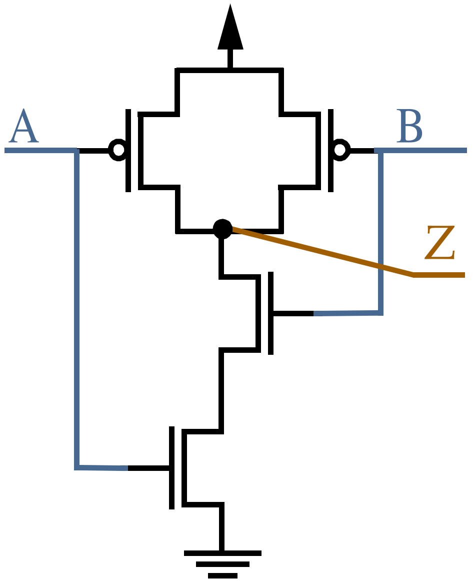
What fault(s) does the pattern *AB = 01* detect?

What is the *minimum* number of tests needed to “detect” all them?

What are the tests?

SSF Fault Example

A ***dominant*** input value is defined as the value that *determines* the state of the output independent of the other values of the inputs.



Inputs	Fault-Free	Faulty Response					
AB	Response	**A/0**	**B/0**	**Z/0**	**A/1**	**B/1**	**Z/1**
00	1	1	1	**0**	1	1	1
01	1	1	1	**0**	**0**	1	1
10	1	1	1	**0**	1	**0**	1
11	0	**1**	**1**	0	0	0	**1**

What is the dominant input value for the NAND?

How many tests do you need to diagnosis the fault?

Can you distinguish between all of the faults?

SSF Fault Definitions

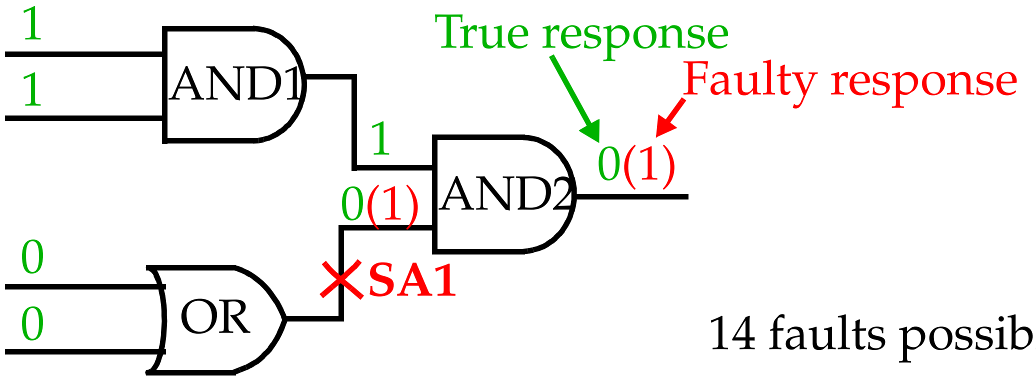
Three properties:

(1) only one line is faulty.

(2) the faulty line is permanently set to either *0* or *1*.

(3) the fault can be at an input or output of a gate.

01, *10*, and *11*
do not provoke
the fault



14 faults possible here.

Fault detection requires:

- A test *t* activates or provokes the fault *f*.
- *t* propagates the error to an observation point (primary output (PO) or scan latch).

A line whose value changes with *f* present is said to be ***sensitized*** to the fault site.

Fault propagation requires that ***off-path*** gate inputs be set to non-dominant values.

Fault Detection

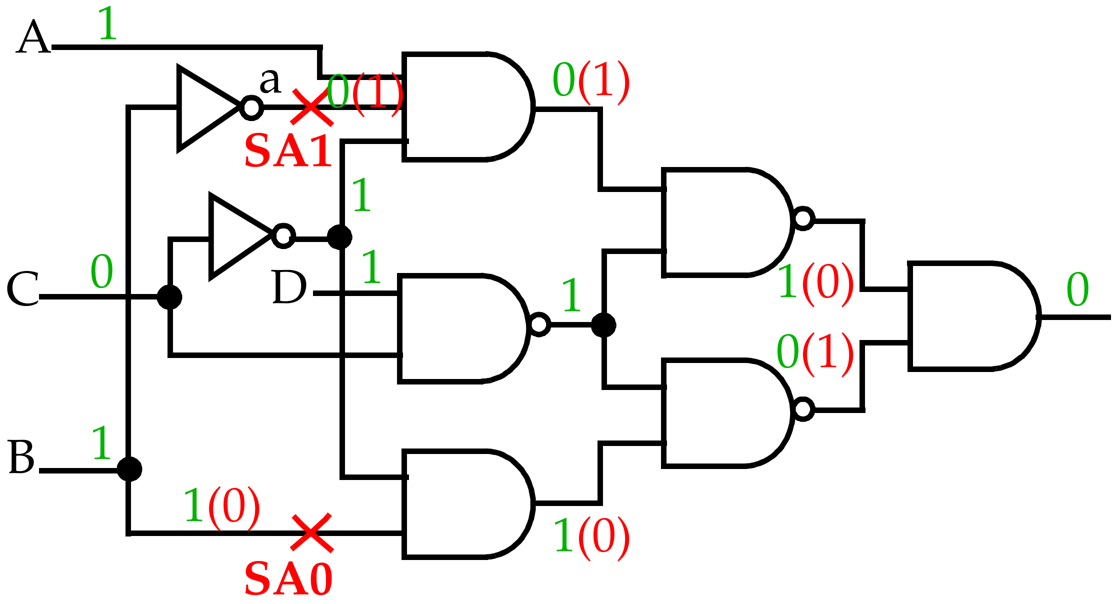
A fault is ***detectable*** if there exists a test ***t*** that defects ***f***.

If ***f*** is undetectable, then no test simultaneously activates ***f*** and creates a sensitized path to a PO.

Undetectable faults may appear to be harmless.

However, a *complete* test set may not be sufficient if one is present.

The fault *b* SA0 is not detectable by the test *t* = (*1101*) if the fault *a* SA1 is present.



Redundancy

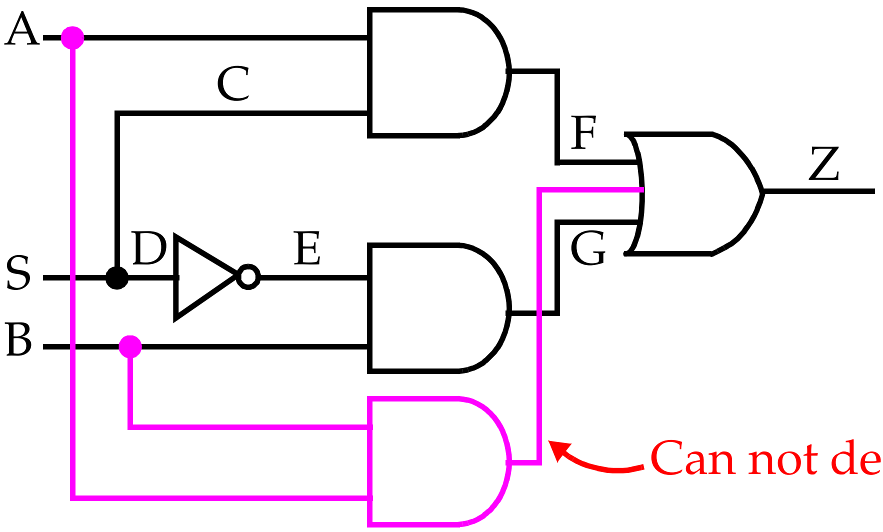
Redundancy: A combinational circuit that contains an undetectable fault is said to be redundant.

Redundant faults cause ATPG algorithms to exhibit worst-case behavior.

Redundancy is not always undesirable.

Triple modular redundancy (TMR) in fault tolerant design.

Redundancy to avoid hazards



Can not detect SA0 -- why?

Describe the transition that causes a static hazard without the magenta AND.

$F(A,B,S) = AS + B\overline{S}$

$F(A,B,S) = AS + B\overline{S} + AB$

redundant product term

Fault Equivalence

Number of fault sites in a Boolean gate circuit = ***# PI + # gates + # (fanout branches)***.

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.

Fault collapsing: All single faults of a logic circuit can be divided into disjoint equivalence subsets, where all faults in a subset are mutually equivalent. A collapsed fault set contains one fault from each equivalence subset.

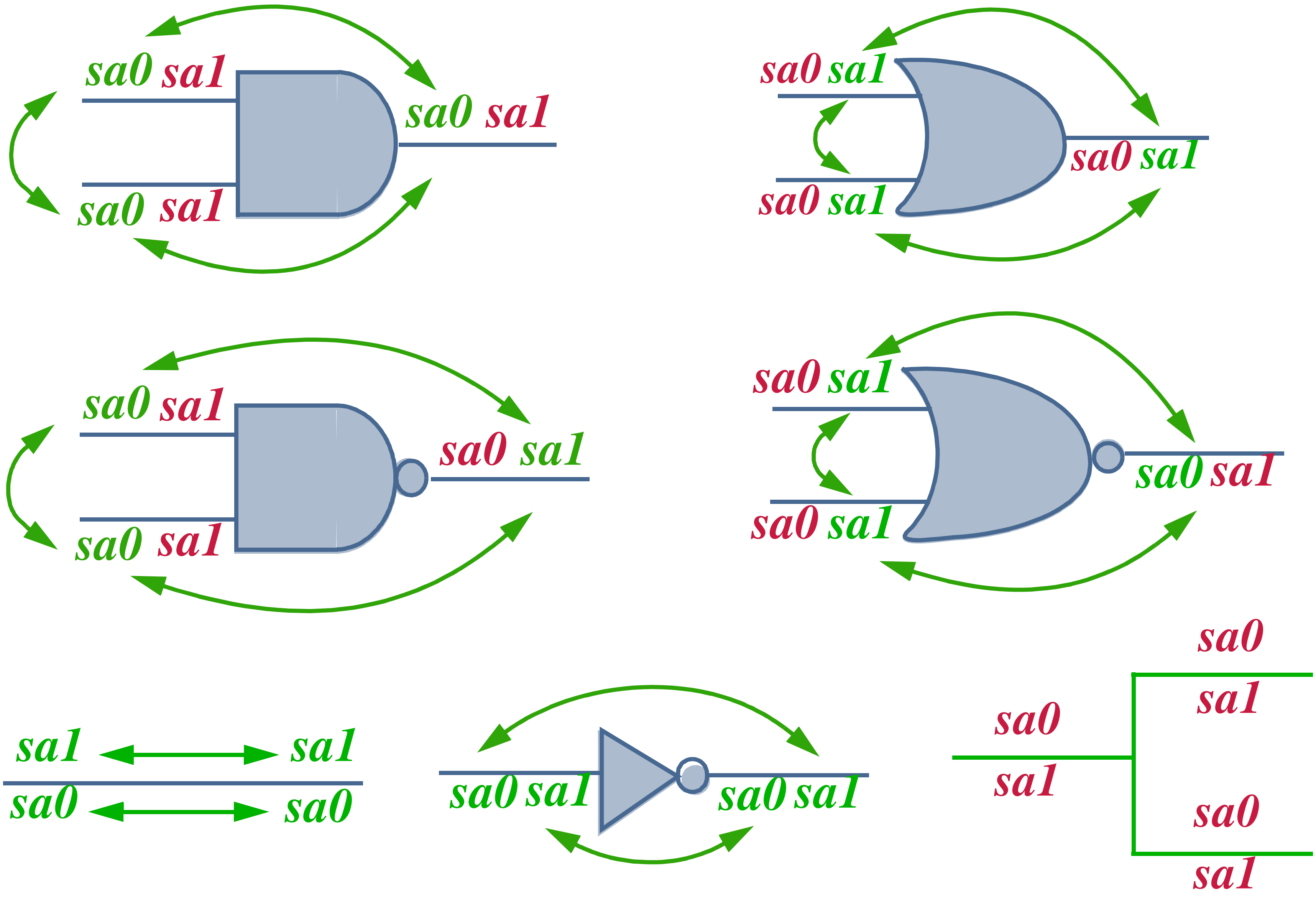
This property is useful for test generation programs.

- Fault equivalence reduces the size of the fault list.

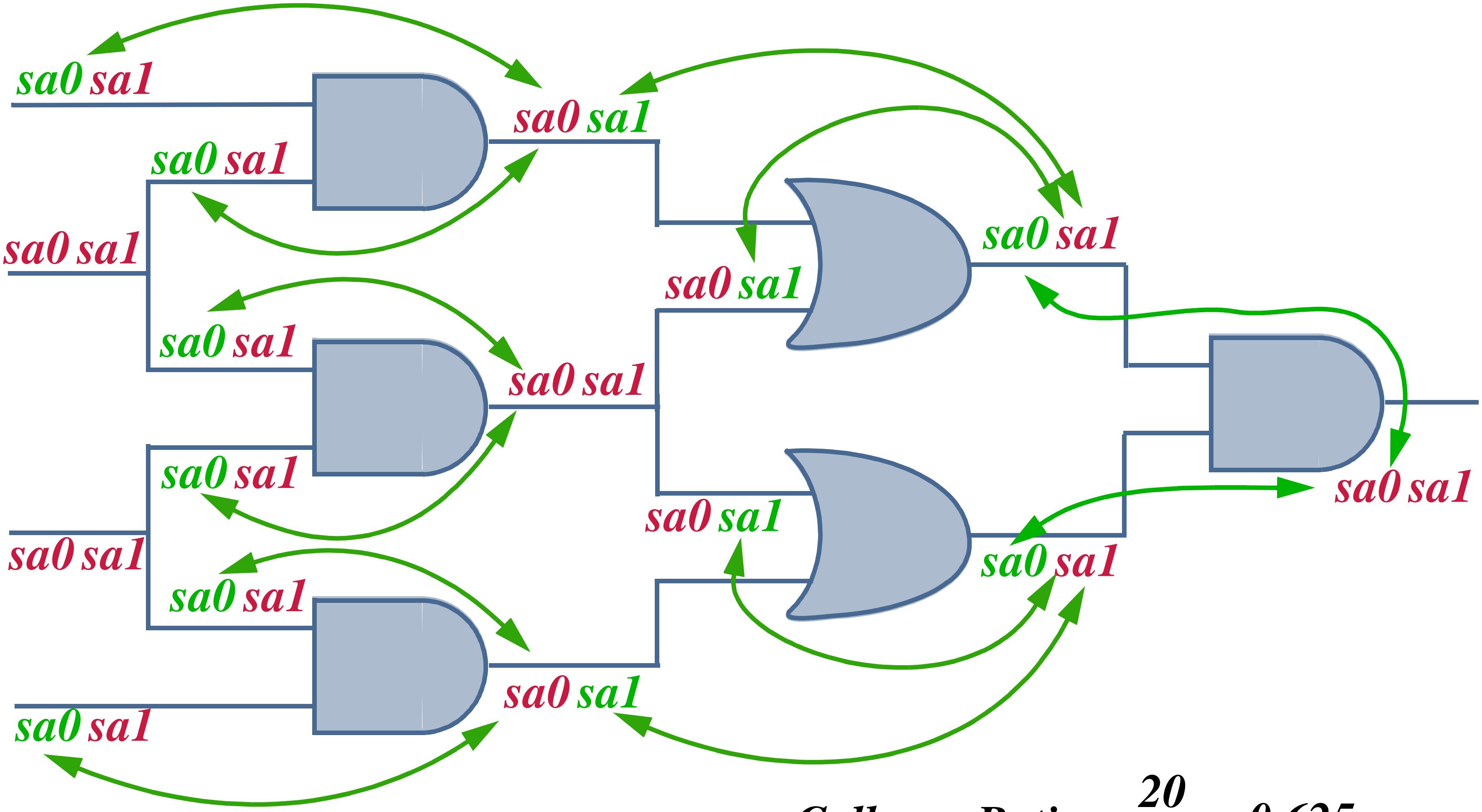
Fault equivalence is important for fault location analysis as well.

- A *complete location test* set can diagnose a fault to within a functional equivalence class.
- However, for large circuits a complete detection or location test are not generally used.

Equivalence Rules



Fault Equivalence Example



$$\textit{Collapse Ratio} = \frac{20}{32} = 0.625$$