Stuck-open and Stuck-on Faults

Transistor level circuit representation is needed to understand how MOS circuits can fail.

Here, we can model the MOS transistor as an ideal switch.

Defects cause the switch to remain permanently open or closed.

Under this model, the I/O behavior of a faulty MOS circuit cannot be exactly represented by the SA fault model.

Stuck-Open Faults (SOP)

Defect creates an unintended high-impedance state on the output node of a gate.

The SOP model adds new fault types to the standard list of *Stuck-At* faults.

Stuck-Open Faults

Consider a 2-input NOR gate:

- *Stuck-At* faults include: *A-SA0*, *B-SA0*, *Out-SA0* and *Out-SA1*.

- *SOP* model adds three *non-classical* faults:

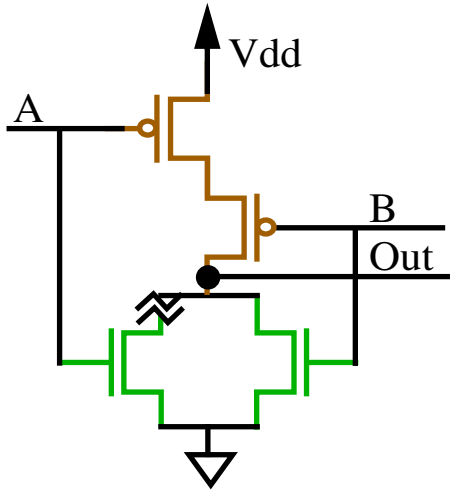
  A-Stuck-Open (*A-SOP*)

  B-Stuck-Open (*B-SOP*)

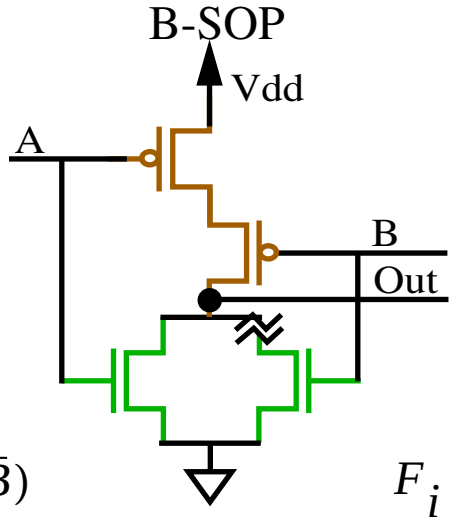
  Vdd-Stuck-Open (V_{DD}-*SOP*)

A-SOP

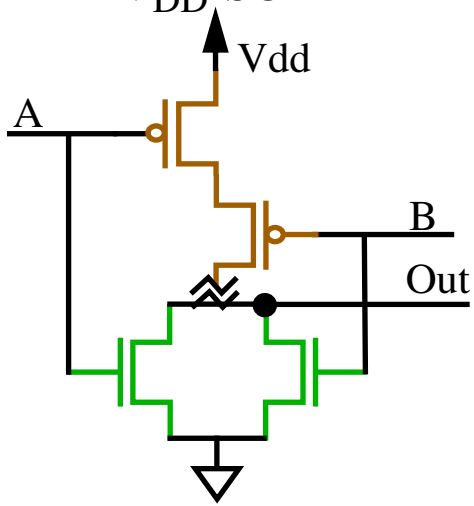
$$F_i = (\bar{A} \bullet \bar{B}) + (F_{i-1} \bullet \bar{A} \bullet B)$$



B-SOP



V_{DD}-SOP



$$F_i = (\bar{A} \bullet \bar{B}) + (F_{i-1} \bullet A \bullet \bar{B})$$

$$F_i = (\bar{A} \bullet \bar{B}) + (F_{i-1} \bullet \bar{A} \bullet \bar{B})$$

Stuck-Open Faults

Exhaustive test $AB = (00, 01, 10, 11)$ does not detect all faults.

For SOP, **vector order** is important.

For example, the sequence $AB = (00, 01, 00, 10)$ is able to detect all faults on the NOR gate, including the SOP faults.

This test sequence tests *each path* between the output and V_{DD} and GND independently.

The tests are really **test sequences**.

First pattern initializes the output node, second pattern checks for the presence of the fault.

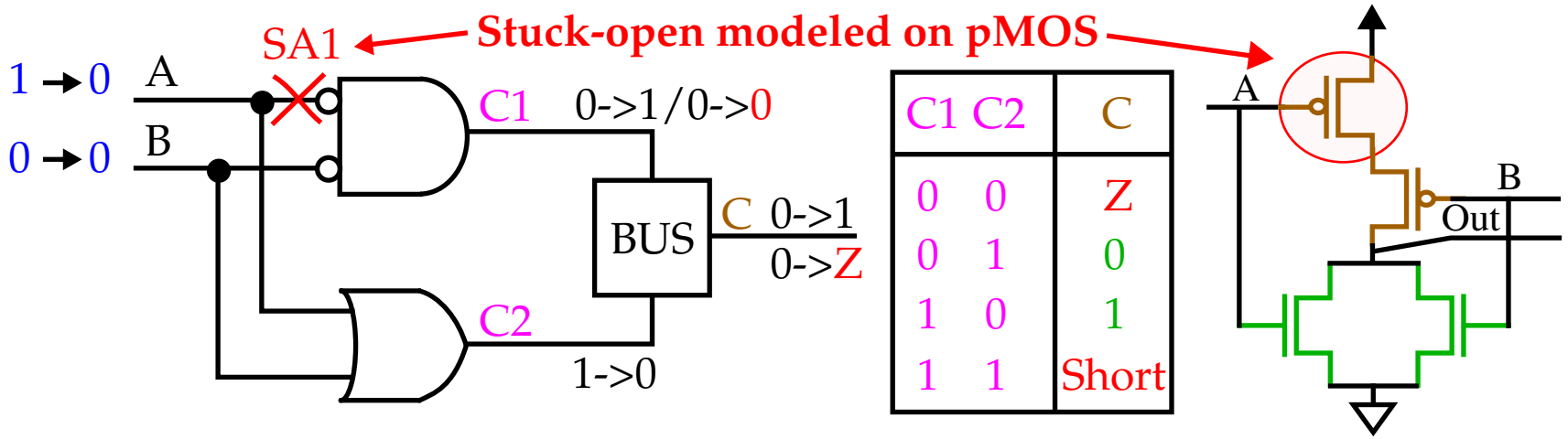
Difficulties:

- Delays to the inputs of the gates may be different creating intermediate states, *invalidating* the initializing test pattern.
- Two-vector sequence TPG time for *SOP* faults is significantly longer than single-vector TPG time for *SSF* faults.

Stuck-Open Faults

Switch-level test generation algorithms can automate the generation of these tests.

A gate level model is also possible:



Series connections of transistors are replaced with AND, parallel connections are replaced with OR, and pMOS inputs are inverted.

Rules: When the 2 inputs to BUS are *different*, output determined by C1.

In fault free circuit, this must be true since complementary logic functions drive BUS.

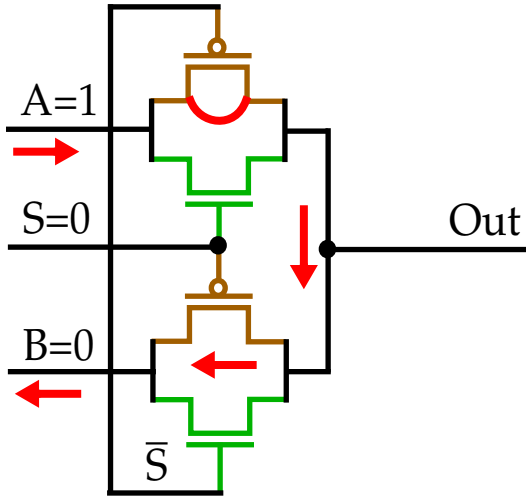
Stuck-short (Stuck-on) Faults

Complement of the *Stuck-Open* fault model is the *Stuck-On* fault model.

Stuck-On faulty gate output is difficult to predict.

A transistor that is permanently stuck-on will, for some input combination(s), compete with its complementary transistors for control of the output.

Sometimes this competition does not result in a catastrophic failure.



Output value depends on:
- Driving transistor resistances.
- Strength of up-stream drivers.

Delay fault is likely if functional behavior is preserved.

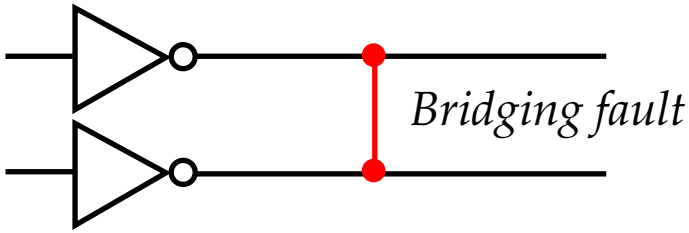
I_{DDQ} test may detect it.

Stuck-short faults modeled as SA0 on pMOS in gate-level equivalent model.

Bridging Faults

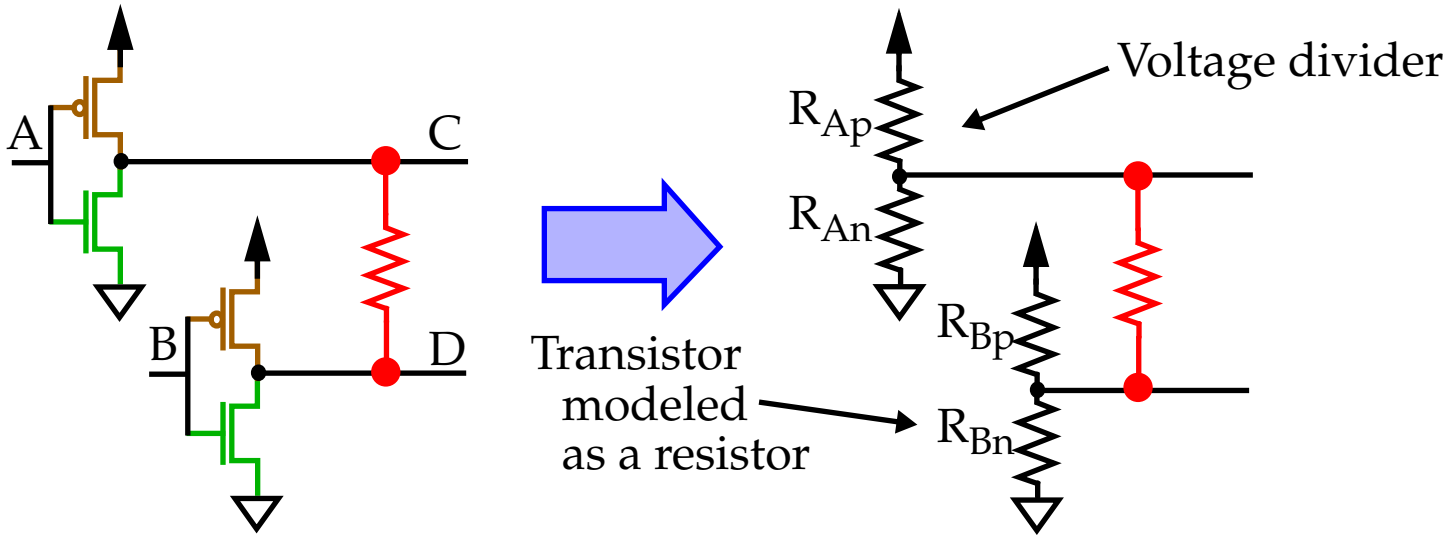
Modeled at the gate or transistor level as a short between 2 (simple) or more of **signal** lines.

Non-feedback versus feedback (memory) versions.



Fault is usually modeled using **wired logic**: AND and OR.

For CMOS, it **depends** on the *type of gates* driving the shorted lines and their *input values*.

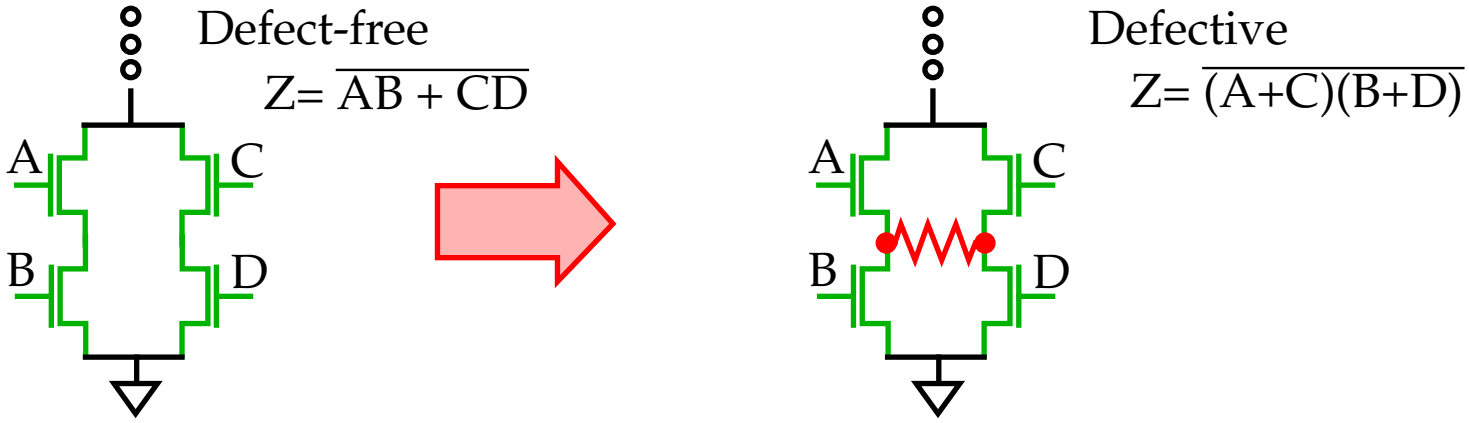


Bridging Faults

The transistor **resistances** determine the appropriate model:

Input values	Resistance relationships	Resulting output value	Wired logic model.
A=B	Any ratio	C = D	AND, OR
A=0, B=1	$R_{Ap} > R_{Bn}$	C = D = 0	AND
	$R_{Ap} < R_{Bn}$	C = D = 1	OR
A=0, B=1	$R_{An} > R_{Bp}$	C = D = 1	OR
	$R_{An} < R_{Bp}$	C = D = 0	AND

Bridging faults that cannot be represented by a **known** fault model.

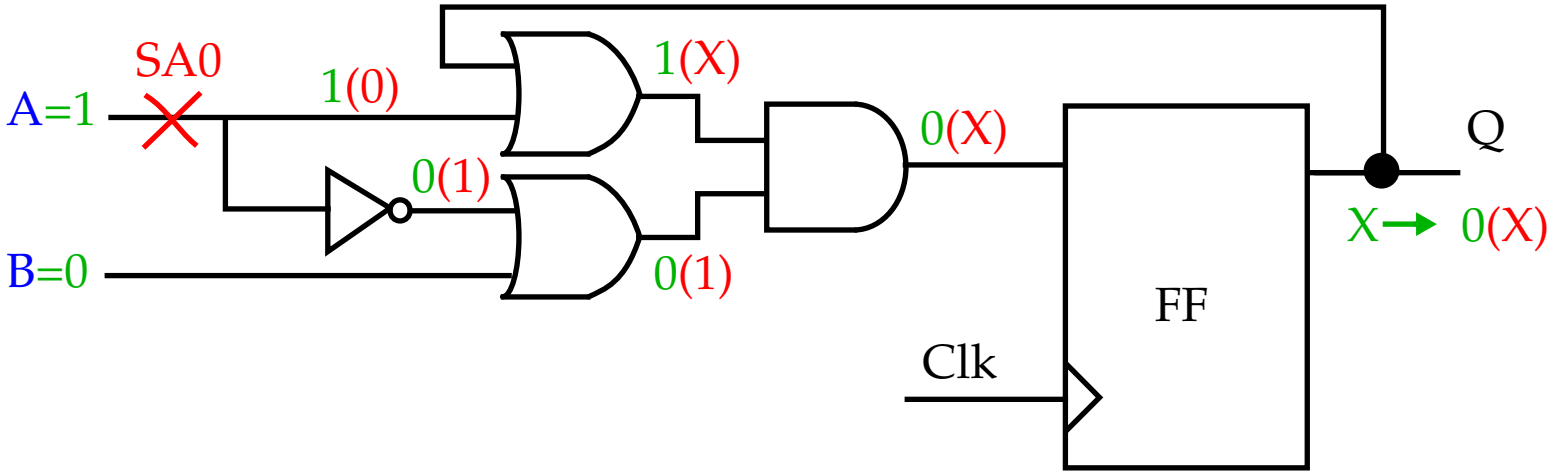


Some convert **combination** circuits to **sequential** (feedback bridging).

Initialization Faults

Circuits with memory elements (FFs) need to be initialized.

Faults that interfere with this process are *initialization faults*.



The initial state of the FF is unknown after power-up.

To initialize *Q* to *0*, set *A = 1* & *B = 0* and apply *Clk*.

With the SA0 fault, the FF remains in the unknown state.

Redundant Faults

A fault that *does not modify* the input-output function of the circuit.

They can be removed from the circuit.

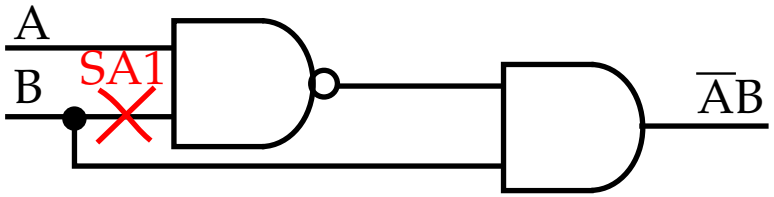
A redundant fault **cannot** be detected using SSF tests.

For combinational circuits, identifying redundant faults is used in circuit optimizations.

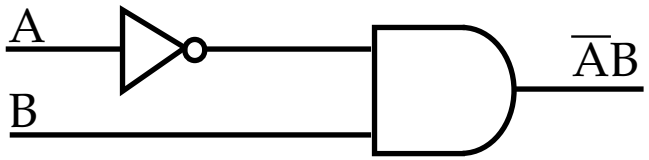
For sequential circuits, it's more difficult to identify and remove redundancy.

In general, faults in sequential circuits for which no test can be found are called *untestable faults*.

Redundant faults are a subset of these.



A redundant SA fault
(reconvergent fanout)



Equivalent circuit
(tree)

Multiple Faults

The simultaneous presence of single faults, usually of the same type.

Usually not considered in practice:

- We indicated earlier that there are 3^n-1 possible multiple stuck-fault (MSF) in a circuit with n SSF sites.
- Tests for SSFs cover a high percentage of MSFs.

Situations in which it is important to consider them:

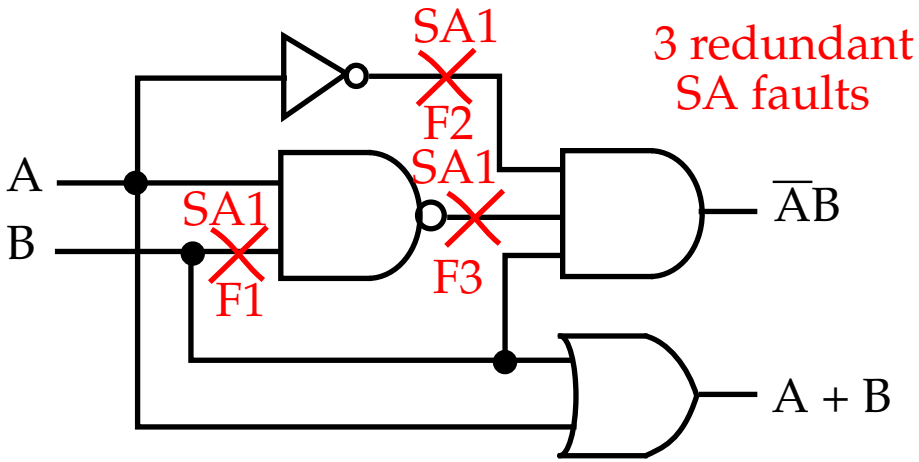
- Diagnostic or fault location procedures don't work with multiple faults.
- A circuit with *redundant SSFs* (SSFs on redundant gates) can malfunction even when it passes the SSF tests.

Solution is to enhance test set to cover multiple faults or remove redundancy.

The latter is usually easier.

Multiple Faults

An example:



MSF	Output at $\overline{A}B$	Test
F1, F2	$\overline{A}B$	Redundant
F1, F3	$\overline{A}B$	Redundant
F2, F3	B	11
F1, F2, F3	B	11

The three faults, SA1-3, are *redundant* because the presence of any one of them has no effect on the logic function.

You can verify that the vectors *00*, *01*, and *10* detect all **other** single SA faults.

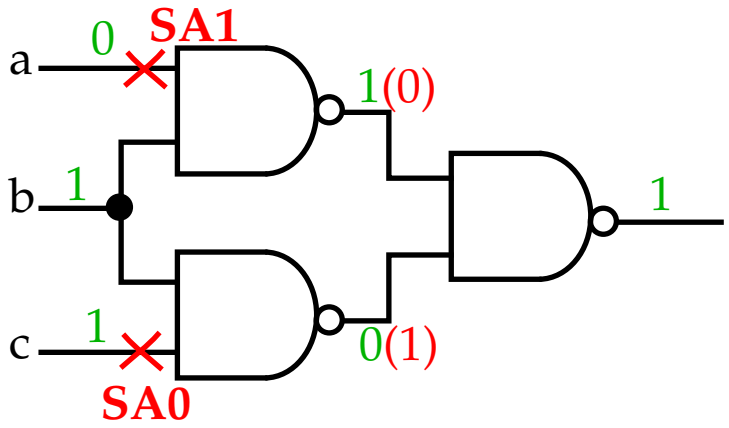
However, in combination they can cause a problem.

For input combination *11*, the fault combinations in the last two rows are detected, but this pattern is not part of the test set.

Multiple Stuck Fault Model

Intuitively, it seems that detecting all SSFs is sufficient to detect the MSFs.

Unfortunately, **functional masking** introduced by MSFs can prevent detection of SSFs.



Only test that detects C SA0 is *abc = 011*.

Presence of *a* SA1 masks it.

Functional masking implies masking under any test set.

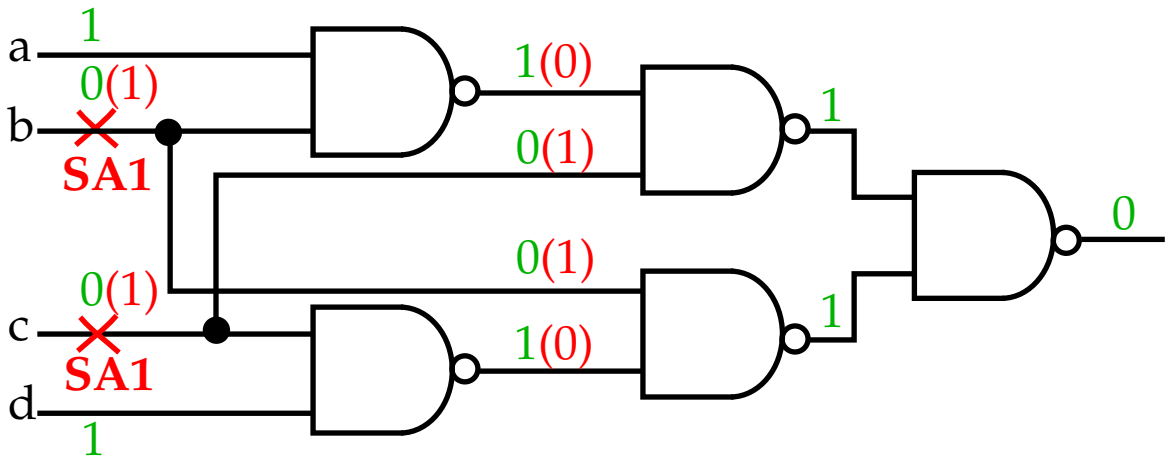
However, it's possible that a multiple fault is masked under a given test and not under another.

The test *abc = 010* detects the MSF {*c* SA0, *a* SA1}.

Multiple Stuck Fault Model

Given a complete test set T for SSFs, can there exist a MSF $F = \{f_1, f_2, ..., f_k\}$ such that F is not detected by T?

Unfortunately, the answer is yes.



The test set $T = \{1111, 0111, 1110, 1001, 1010, 0101\}$ detects all SSFs.

The only test in T that detects both f = b SA1 and g = c SA1 is *1001*.

However, the **circular masking** of f and g under T prevents the MSF $\{f, g\}$ from being detected.

Fortunately, circular masking relations are seldom encountered in practice.

Delay Fault Models

- **Transition Fault (gross-delay faults)**:
  Gate delay increased to point where transition does not reach output before end of clock period, even along the shortest path.
- **Gate-delay Fault**:
  Defect increases input to output delay of a single logic gate.
- **Line-delay Fault**:
  In contrast to transition fault, a test here must propagate the transition through the *longest sensitizable path*.
- **Path-delay Fault**:
  This fault causes the cummulative propagation delay of a path to increase beyond some specified time duration.
- **Segment-delay Faults**:
  A segment (of length L gates) delay fault increases the delay of a segment such that all paths containing the segment have a path-delay fault.
  Segment-delay = Path-delay if L is the maximum combinational depth of the circuit.
  Segment-delay = Transition fault if L is 1.