COE 571 Digital System Testing

Dr. Aiman El-Maleh

Fault Modeling

1. **Fault detection & redundancy**
2. **Fault equivalence and fault location**
3. **Equivalence fault collapsing**
4. **Fault dominance**
5. **Single stuck-at fault model**
6. **Checkpoint theorem**
7. **The multiple stuck-at fault model**

Fault Modeling

- Logical faults represent the effect of physical faults on the behavior of the modeled system.

- Faults can affect:
  - Logic function
  - Operating speed of system (delay faults)

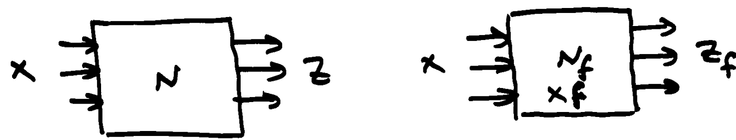
- Advantages of modeling physical faults as logical faults:
  1. Fault analysis becomes logical problem and its complexity is greatly reduced; many physical faults can be modeled by same logical fault
  2. Logical fault models can be technology independent ⇒ testing & diagnosis methods remain valid

- Fault modeling is closely related to the type of modeling used
  - Structural faults: modify the interconnection of components
  - Functional faults: may change truth table of a component or inhibit an RTL operation

- Single-fault assumption:
  At most one logical fault exists in system
- Faults are assumed to be permenant;
  intermittent & transient faults are dealt with
  by on-line testing.
- Often, a multiple fault can be detected by
  tests designed for the single faults that compose
  the multiple one.
- Typical faults affecting interconnections:
  - Short: connecting points not intended to be
    connected
  - Open: breaking a connection
- A short between a signal line and ground
  or power can make the signal having a fixed
  voltage level
  ⇒ Signal is stuck-at a fixed logic value
  $v \in \{0, 1\}$ denoted s-a-v
- A short between two signal lines usually
  creates a new logic function
  ⇒ Bridging fault: AND Bridging faults;
  OR Bridging faults
- In many technologies, the effect of open
  faults appear as a stuck fault.

Fault Detection & Redundancy

- Let $Z(x)$ be the logic function of a combinational circuit N, where x represents an arbitrary input vector
- The presence of a fault f transforms N into a new circuit N_f with function $Z_f(x)$.



- **Definition:** A test vector t detects a fault f iff $Z_f(t) \neq Z(t)$
- For a single-output circuit, a test t detects a fault if $Z(t) \oplus Z_f(t) = 1$
- The set of all tests that detect f is given by the solutions of the equation:

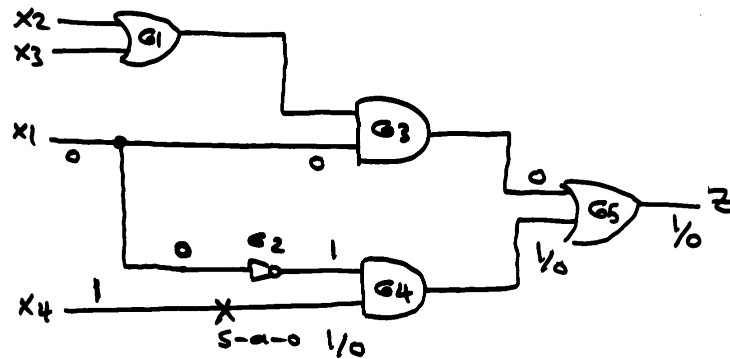
$$Z(x) \oplus Z_f(x) = 1$$

- For a circuit with m outputs, a test t detects a fault f iff:

$$[Z_1(t) \oplus Z_{1f}(t)] + [Z_2(t) \oplus Z_{2f}(t)] + \cdots + [Z_m(t) \oplus Z_{mf}(t)] = 1$$

i.e. there is at least one output different in the fault-free & faulty machines

Example



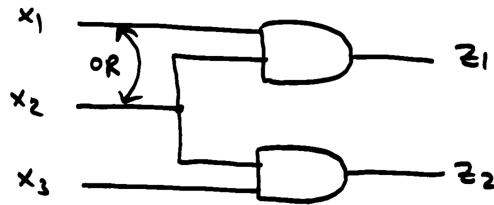
$$z = (x_2 + x_3)x_1 + \bar{x}_1 x_4$$

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

$$z \oplus z_f = 1 \Rightarrow \bar{x}_1 x_4 = 1$$

$$T = \{0xx1\}$$
$$= \{0001, 0011, 0101, 0111\}$$

Example:



let f be the OR bridging fault between x_1 and x_2.

$Z_{1f} = x_1 + x_2$; $Z_1 = x_1 x_2$

$Z_{2f} = (x_1 + x_2) x_3$; $Z_2 = x_2 x_3$

The set of tests that detect this fault are:

$$[(x_1 + x_2) \oplus x_1 x_2] + [(x_1 + x_2) x_3 \oplus x_2 x_3] = 1$$

$$\Rightarrow \bar{x}_1 x_2 + x_1 \bar{x}_2 + x_1 \bar{x}_2 x_3 = 1$$

$$\Rightarrow T = \{01x, 10x, 101\} = \{01x, 10x\}$$

($01x$, $10x$): fault effect observed at z_1

(101): fault effect observed at z_1 & z_2

- A test t that detects a fault f:
  - activates f: generates an error or fault effect by creating different v and v_f values at the fault site
  - propagates error: propagates fault effect from fault site to at least one primary output w
    - ⇒ makes all lines along at least one path between fault site and output have different v and v_f values.

- A line whose value in the test t changes in the presence of the fault f is said to be sensitized to the fault f by the test t.

- A path composed of sensitized lines is called sensitized path.

- A gate whose output is sensitized to a fault f has at least one of its inputs sensitized to f as well.

Lemma

Let G be a gate with inversion i and controlling value c, whose output is sensitized to a fault f by a test t

1. All inputs of G sensitized to f have the same value (say a)
2. All inputs of G not sensitized to f (if any) have value $\bar{c}$
3. The output of G has value $a \oplus i$

Corollary

Let j be a line sensitized to the fault l s-a-v (by a test t);
Let p be the inversion parity of a sensitized path between l and j

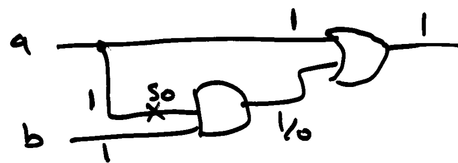
1. The value of j in t is $\bar{v} \oplus p$
2. If there are several sensitized paths between l and j, then all of them have the same inversion parity.

- Fault Detectability:
  A fault is said to be detectable if there exists a test t that detects f; otherwise f is an undetectable fault.

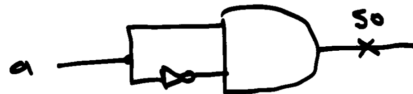
- For an undetectable fault, f, $Z_f(x) = Z(x)$ and no test can simultaneously activate f and create a sensitized path to a primary output.

- Examples of undetectable faults:

  1. A fault that can be excited but not propagated simultaneously



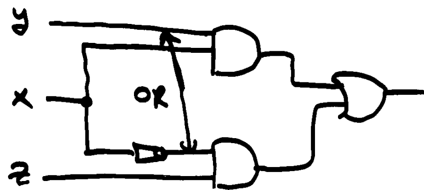
  2. A fault that cannot be excited or activated



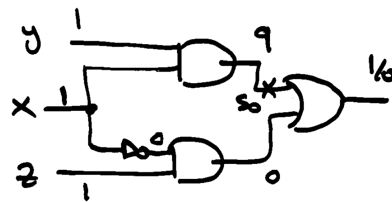
⇒ A complete test set may not be sufficient to detect all detectable faults if there is an undetectable fault in the circuit

- The situation in which the presence of an undetectable fault prevents the detection of another fault by a certain test is not limited to faults from same category.

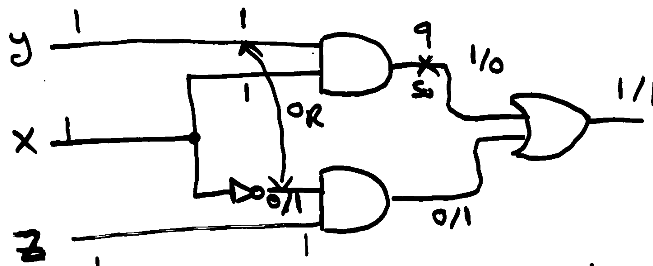
- An undetectable bridging fault can invalidate a complete test set for stuck faults



OR bridging fault between y & $\bar{x}$ is undetectable

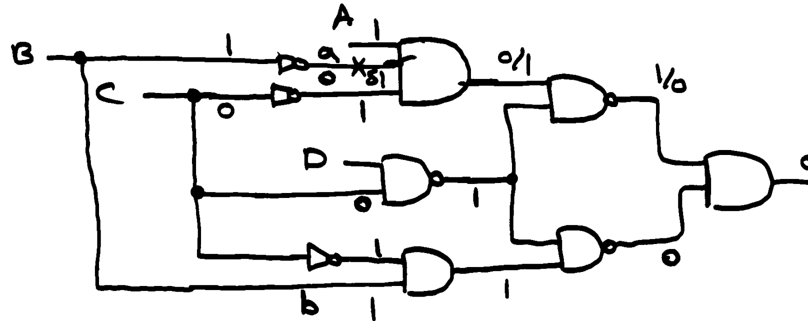


The test (111) detects the fault q s-a-0

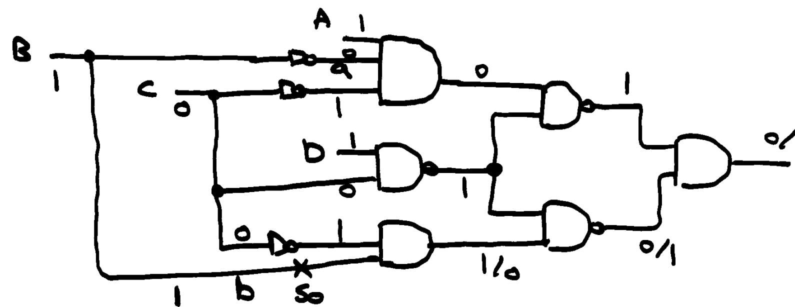


The test (111) does not detect the fault q s-a-0 in presence of the undetectable Bridging fault.

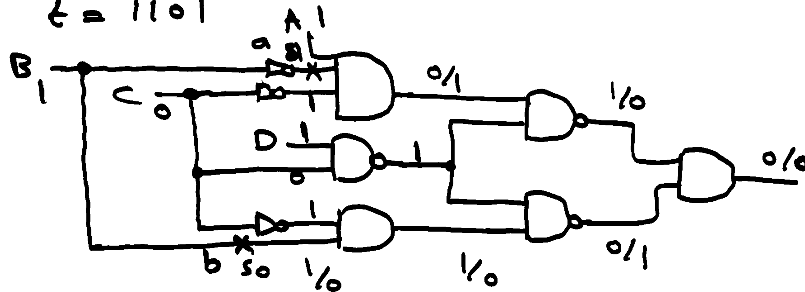
Example



⇒ The fault a s-a-1 is undetectable



⇒ The fault b s-a-0 is detected by $t = 1101$



⇒ The fault b s-a-0 is no longer detected by $t = 1101$ in presence of the undetectable fault a s-a-1.

Redundancy

- A combinational circuit that contains an undetectable stuck fault is said to be <u>redundant</u>
  ⇒ it can be simplified by removing at least one gate or gate input

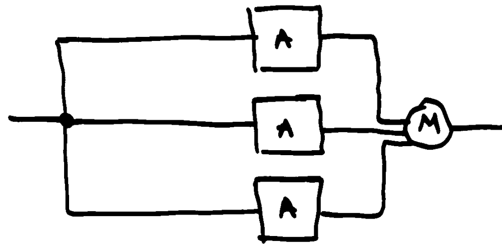
- <u>Simplification rules:</u>

undetectable fault	Simplification rule
AND (NAND) input s-a-1	Remove input
AND (NAND) input s-a-0	Remove gate, replace by 0(1)
OR (NOR) input s-a-0	Remove input
OR (NOR) input s-a-1	Remove gate, replace by 1(0)

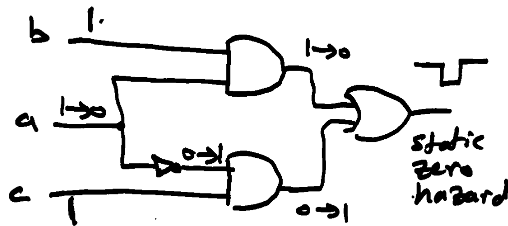
- A combinational circuit in which all stuck faults are detectable is said to be <u>irredundant</u>.

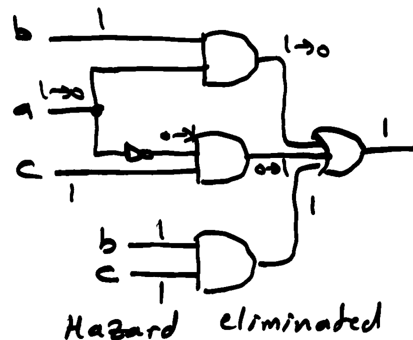
- Redundancy does not necessarily denote an inefficient or undesirable implementation of a function

- Triple Modular Redundancy (TMR)
    - Basic technique used in fault-tolerant design
    - Untestable by offline testing



- Redundancy may be introduced to avoid hazards

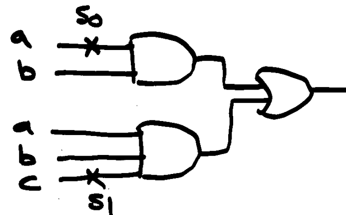




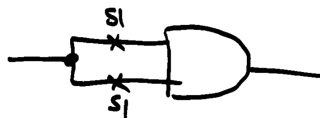
- Problems that can arise in redundant circuits include:

1. The presence of a redundant fault may invalidate a complete test set

2. If f is a detectable fault and g is an undetectable fault, then f may become undetectable in presence of g.



3. Two undetectable single faults f and g may become detectable if simultaneously present.



Fault Equivalence & Fault Location

- **Definition:** Two faults f and g are said to be functionally equivalent iff $Z_f(x) = Z_g(x)$

- A test t is said to distinguish between two faults f and g if $Z_f(t) \neq Z_g(t)$

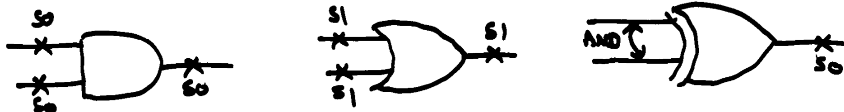
- The functional equivalence relation partitions the set of all possible faults into functional equivalence classes

- It is sufficient to consider only one representative fault from every equivalent class

- For a single-output circuit, the set of all tests that distinguish between faults f and g is given:

$$Z_f(x) \oplus Z_g(x) = 1$$

- Equivalence fault relation is not restricted to the same fault universe

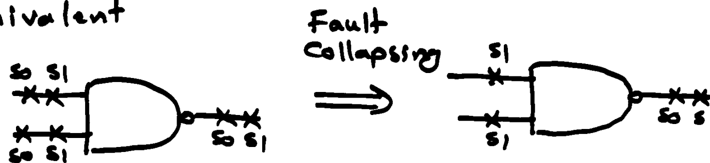


Examples of equivalent faults

Equivalence Fault Collapsing

- An n-input gate has $2(n+1)$ single stuck faults

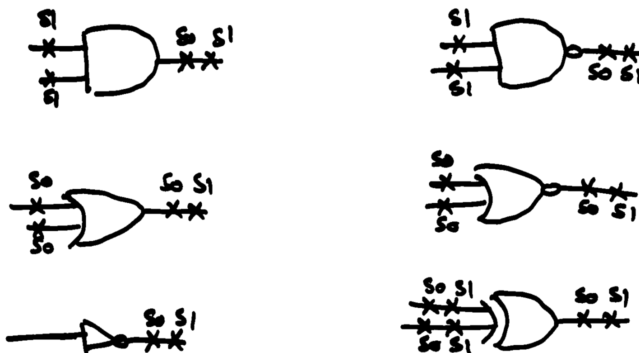
- For a NAND gate, all the input s-a-0 faults and the output s-a-1 are functionally equivalent



- For a gate with controlling value c and inversion i, all the input s-a-c faults and the output s-a-$(c \oplus i)$ are functionally equivalent.

- An n-input primitive gate $(n>1)$, has **$n+2$** single stuck-at fault equivalent classes

- Examples:

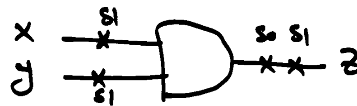


- A <u>complete location test set</u> distinguishes between every pair of distinguishable faults in a circuit

- A complete location test set must include a complete detection test set

- A complete location test set can diagnose a fault to within a functional equivalence class

- <u>Example</u>



x	y	z	z_{x-s0}	z_{y-s1}	z_{s0}	z_{s1}
0	0	0	0	0	0	1
0	1	0	1	0	0	1
1	0	0	0	1	0	1
1	1	1	1	1	0	1

← fault dictionary

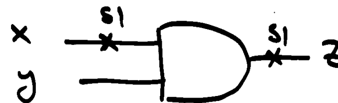
The test set {01, 10, 11} is a complete detection and complete location test set

* The presence of an undetectable fault may invalidate a complete location test set.

- <u>Definition</u>: Two faults f and g are functionally equivalent under a test T iff $Z_f(t) = Z_g(t)$ for every test $t \in T$

- Functional equivalence implies equivalence under any test set, but equivalence under a given test set does not imply functional equivalence

- <u>Example</u>:



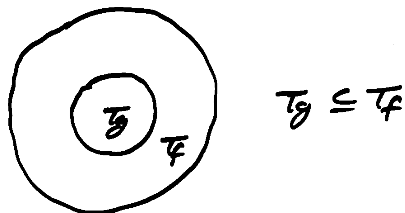
The test $t = \{01\}$ detects the fault x s-a-1 and the fault z s-a-1. The two faults are functionally equivalent under t. However, they are not functionally equivalent.

- The set of tests that detect $x_{s1} = \{01\}$
- The set of tests that detect $z_{s1} = \{00, 01, 10\}$
- Note that $T_{xs1} \subseteq T_{zs1}$

Fault Dominance

- **Definition**: Let T_g be the set of all tests that detect a fault g. We say that a fault f dominates the fault g iff f and g are functionally equivalent under T_g.

- For fault detection, it is unnecessary to consider the dominating fault f, since by deriving a test to detect g, we obtain a test that detects f



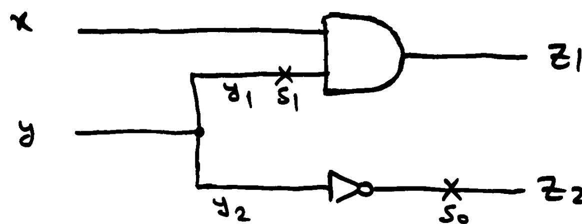
- For a gate with controlling value c and inversion i, the output s-a-($\bar{c} \oplus i$) fault dominates any input s-a-$\bar{c}$ fault.
  - ⇒ The output fault s-a-($\bar{c} \oplus i$) can be removed from faults considered for test generation

- Dominance fault collapsing: reduction of set of faults to be analysed based on dominance relation

- if a fault f dominates g then
  $T_g \subseteq T_f$

- However, $T_g \subseteq T_f$ does not imply that fault f dominates fault g

- <u>Example</u>



Let f be z_2 s-a-0 and g be y_1 s-a-1

$T_g = \{10\}$

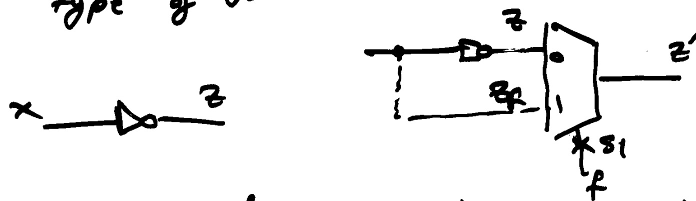
$T_f = \{00, 10\}$ $\quad T_g \subseteq T_f$

However, fault f does not dominate fault g by definition

- Note that for this example it is not necessary to consider fault f for fault detection. However, it is difficult to determine this from analysis of circuit.

Single Stuck-at Fault Model

- It is the 1st used and referred to as the classical or standard fault model
  - represents many different physical faults
  - independent of technology
  - tests that detect SSFs detect many nonclassical faults as well
  - compared to other fault models, the number of SSFs is small
  - SSFs can be used to model other type of faults

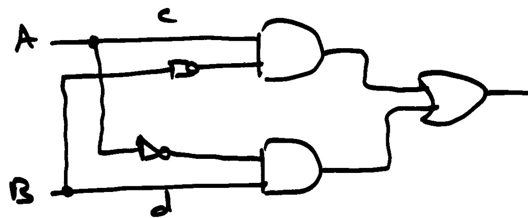


- The number of SSFs in a circuit depends on both the gate count and on the average fanout count

Fault Collapsing

- Determining whether two arbitrary faults are functionally equivalent is an <u>NP-complete</u> problem.

- Example



There is no simple way to know that c s-a-1 and d s-a-1 are functionally equivalent

- Structural equivalence
  - Two faults f and g are structurally equivalent if corresponding simplified circuits $S(N_f)$ and $S(N_g)$ are identical
  - Allows fault collapsing to be done as a simple local analysis process

Structural Fault Collapsing

- start by inserting s-a-1 and s-a-0 faults on every signal source (gate output or primary input) and destination (gate input)

- traverse the circuit and construct structural equivalent classes along the way

- For a signal line with a fanout of 1, faults inserted at its source are structurally equivalent to corresponding faults at destination

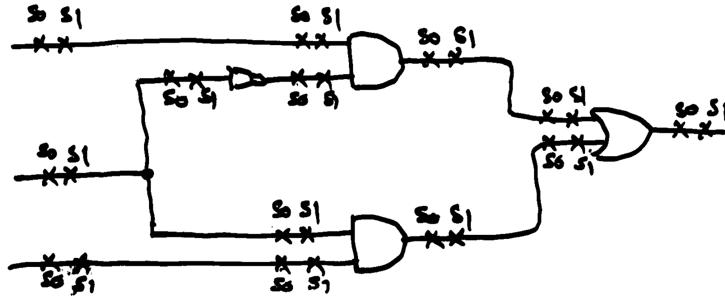
- For a gate with controlling value c and inversion i, any s-a-c input fault is structurally equivalent to s-a-($c \oplus c$) on the output

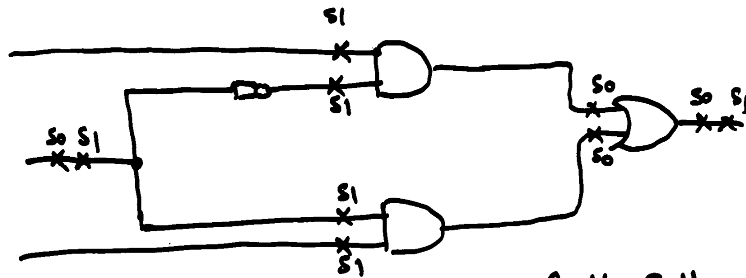
- A s-a-0 (s-a-1) input fault of an inverter is structurally equivalent to output s-a-1 (s-a-0)

- From every equivalence class, retain one fault as representative

- This analysis is performed on fanout-free regions of the circuit

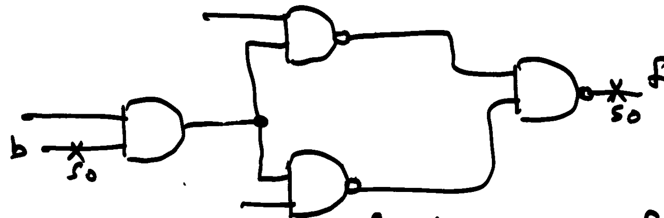
Example





after structural equivalent fault collapsing

- Reconvergent fanout may create structurally equivalent faults in different fanout regions. However, the potential gain from extending the analysis across fanout regions does not justify the cost



structurally equivalent faults across fanout regions

- If we are interested in fault detection only, we can do also dominance fault collapsing across fanout regions

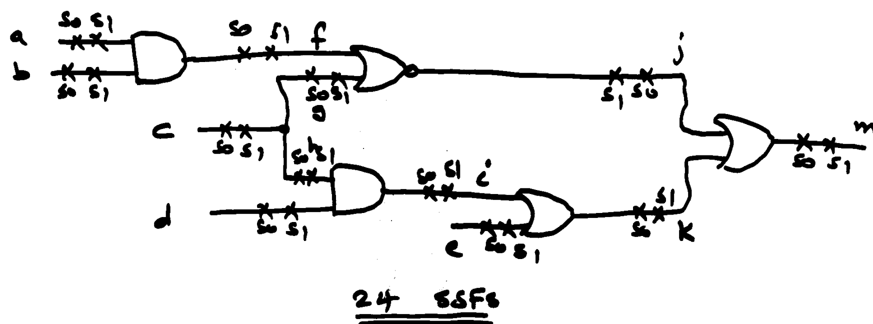
- <u>Theorem</u>: In a fanout-free combinational circuit C, consisting of primitive gates, any test set that detects all SSFs on the primary inputs of C detects all SSFs in C.

- The primary inputs and the fanout branches are called <u>checkpoints</u>

- <u>Checkpoint Theorem:</u>
  In a combinational circuit C, any test set that detects all SSFs on the checkpoins of C detects all SSFs in C.

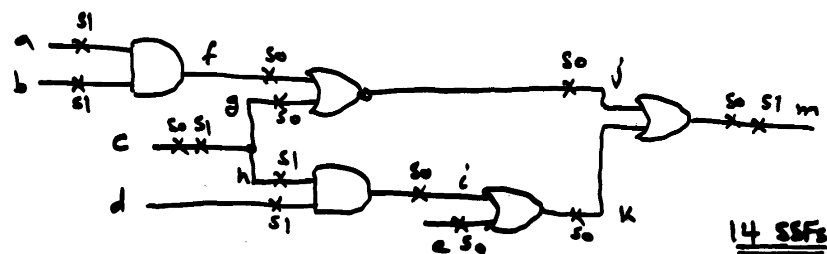
- Note that this theorem does not apply on compound gates like the XOR gate.

Example :



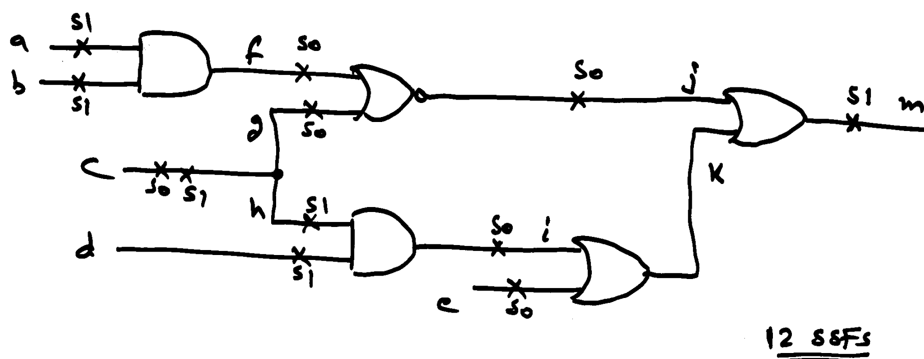
24 SSFs

- After collapsing based on fault equivalence



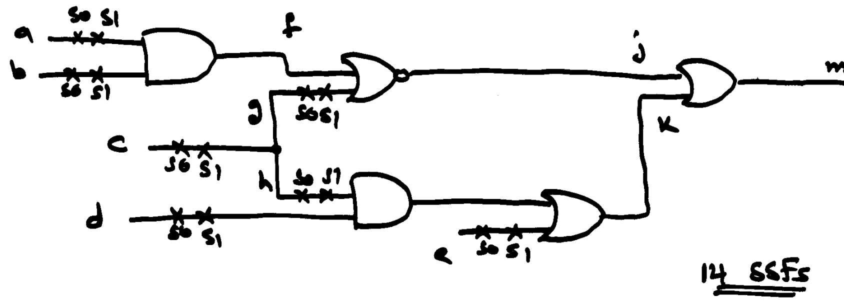
14 SSFs

- After collapsing based on fault equivalence & fault dominance

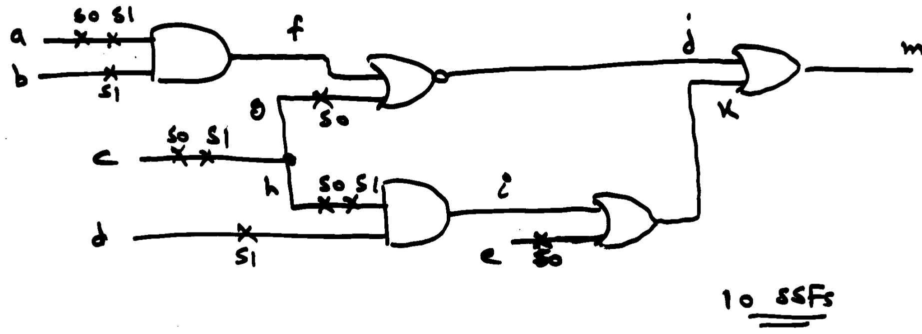


12 SSFs

- Checkpoint faults:



- Checkpoint faults after collapsing based on fault equivalence & dominance



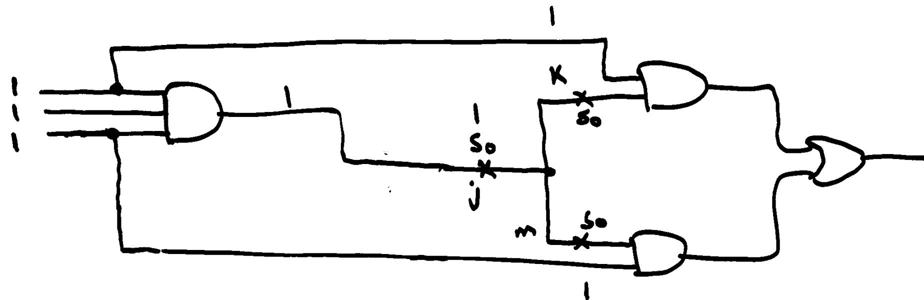
- Generating tests only for the checkpoints is valid if the circuit is irredundant

- In a redundant circuit, some of the checkpoint faults are undetectable

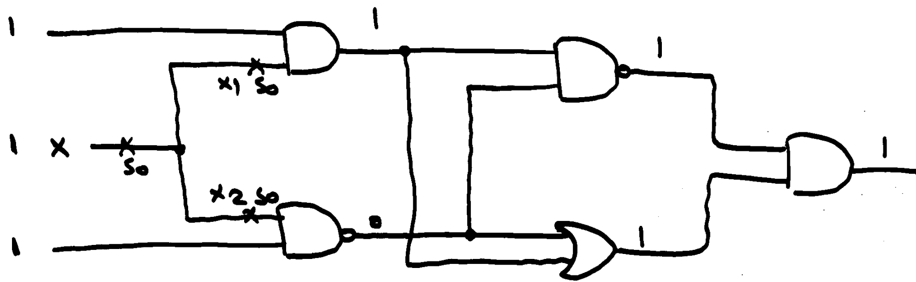
- A test set that detects all detectable checkpoint faults is not guaranteed to detect all detectable checkpoint faults

- A stem s-a-v is equivalent to the multiple fault composed of all its fanout branches s-a-v.

- In general, neither equivalence nor dominance relations exist between a stem s-a-v and an individual fanout branch s-a-v



The stem fault j s-a-0 is detectable while the single faults on its branches k s-a-0 and m s-a-0 are not.



The single faults on fanout branches x_1 s-a-0 and x_2 s-a-0 are detected but the stem fault x s-a-0 is not.

The Multiple Stuck-Fault Model

- Assume that we have a circuit with n lines
  - # of SSFs = $2n$
  - # of MSFs = $3^n - 1$

- If we assume that the multiplicity of a fault, i.e. # of lines simultaneously stuck, is no greater than a constant K
  - # MSFs = $\sum_{i=1}^{K} \binom{n}{i} 2^i$

- **Example**: # of double faults (K=2) in a circuit with n = 1000 ≈ two million

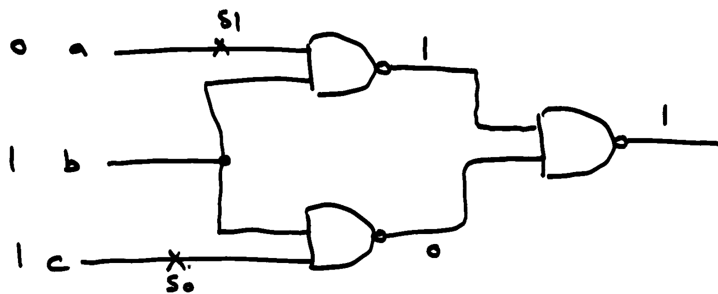
- Number of multiple faults is too large to deal with explicitly

Why MSFs ?!

- Since a multiple fault F is just a set $\{f_1, f_2, \ldots, f_k\}$ of single faults f_i, why isn't F detected by the tests that detect the single faults f_i?
  This is due to masking relations among faults

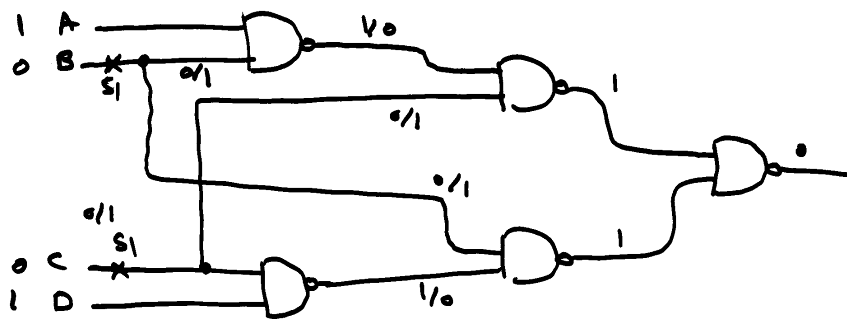
- Definition Let T_g be the set of all tests that detect a fault g. We say that a fault f functionally masks the fault g iff the multiple fault $\{f, g\}$ is not detected by any test in T_g.

- Example:



The fault a s-a-1 masks the fault c s-a-0

- Given a complete test set T for single faults, it is possible to have a multiple fault $F = \{f_1, f_2, \ldots, f_K\}$ such that F is not detected by T.



- let f be B s-a-1 & g be C s-a-1
- T = {1111, 0111, 1110, 1001, 1010, 0101} detects all SSFs in the circuit
- The only test in T that detects f and g is 1001.
- The multiple fault {f,g} is not detected by T because under the test 1001, f masks g and g masks f

  ⟹ circular masking under T

- Circular functional masking may result in an undetectable multiple faulty in a completely SSF testable circuit.

- An important practical question:

What percentage of the MSFs can escape detection by a test set designed to detect SSFs?

- The answer depends on the circuit structure not on the size. The following results have been obtained for combinational circuits:

1. In an irredundant two-level circuit, any complete test set for SSFs also detects all MSFs.

2. In a fanout-free circuit, any complete test set for SSFs detects all double & triple faults & there exists a complete test set for SSFs that detects all MSFs.

3. In an internal fanout-free circuit C (i.e. only PIs may be stems) any complete test set for SSFs detects at least 98% of all MSFs of multiplicity < 6

4. A test set that detects all MSFs on the checkpoints of a circuit detects all multiple faults.