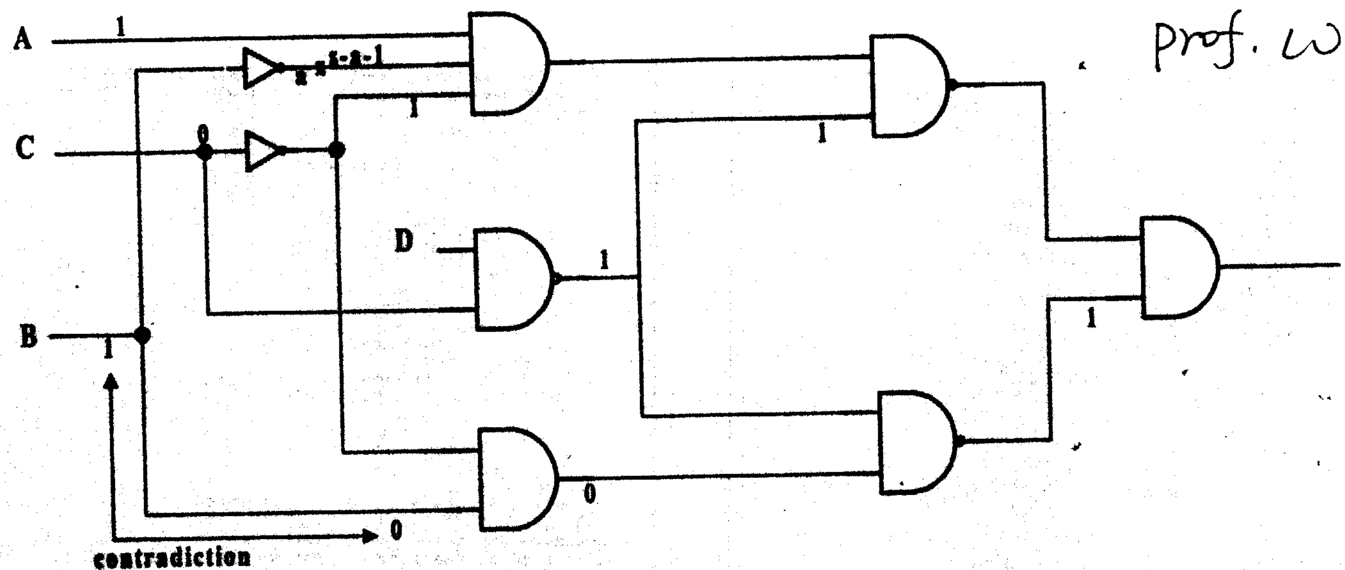
Detectability

A fault f is detectable if there exists a test t which detects f.

ECES 682
VLSI Test & Validation
Prof. Wen-Ben Jone

- Example :
  a s-a-1 in (a) is not detectable! Why?



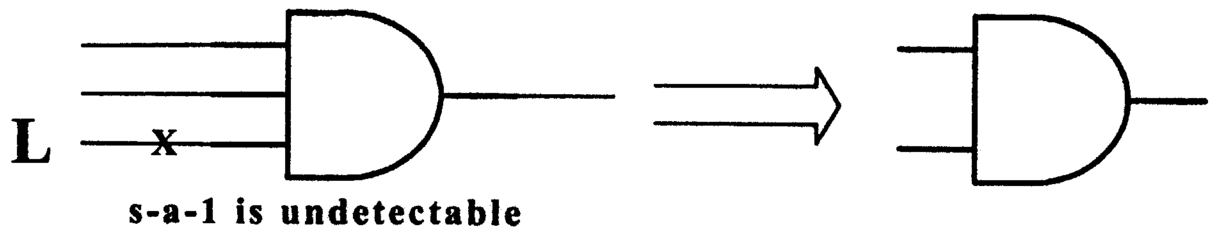
F41

Redundancy

A combinational circuit containing an undetectable stuck-at fault is called redundant.

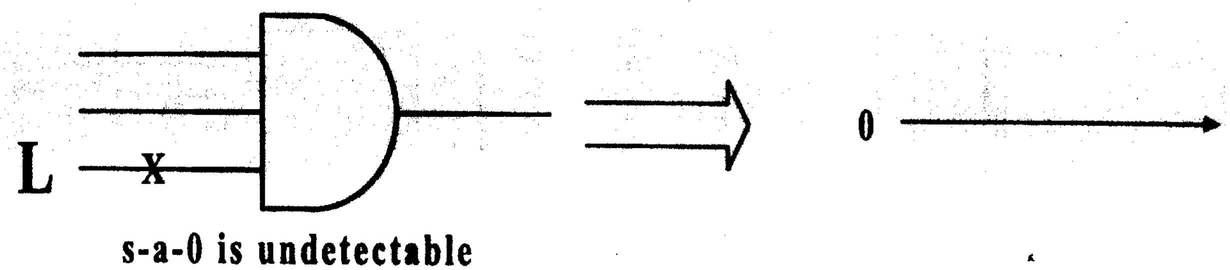
Reason: there are some unnecessary lines or gates.

- Example :



Since line L can be always 1 without changing the function, line L can be removed.

Another Example



Some other properties of Redundancy

① If f is detectable, g is undetectable

⇒f may become undetectable in the presence of g

∴f is called a 2nd generation redundant fault

②Two undetectable single faults f and g may become detectable, if they are present simultaneously.

Important thing

How to recognize redundant fault , s.t. we don't need to try all input combinations for a redundant fault?

☆No difference between an undetectable fault and hard-to-detect fault.

try long cpu time still can't find a test pattern

☆complexity(test generation) = complexity(identify redundant fault) = NP-complete.

☆Although test generation problem is NP-complete, practical test generation algorithms run in polynomial time.

Say n^3 get coverage 99% for very large circuits.

☆NP-complete means no polynomial time algorithm can find test patterns for all circuits.

☆Redundant fault is usually the major factor for the worst case.

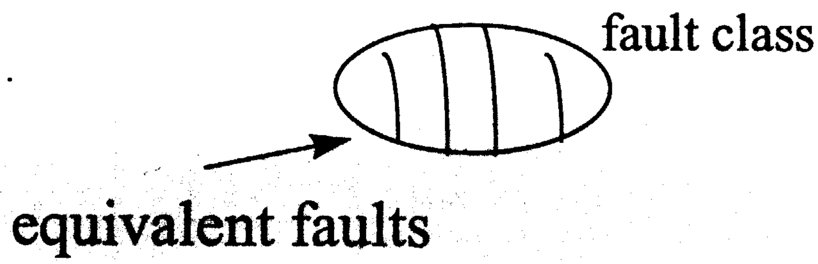
Fault equivalence and fault Location

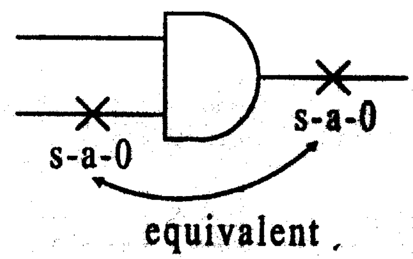
combinational ckts

Def : Two faults f and g are functional equivalent $\leftrightarrow$ $Z_f(x)=Z_g(x)$ for all x

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

Example :

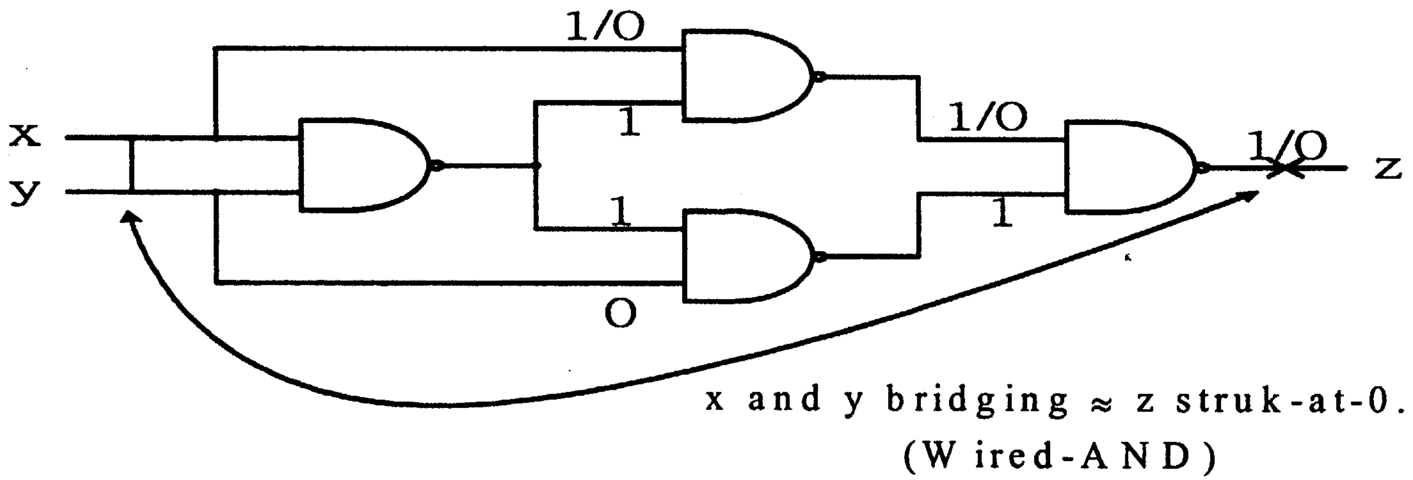




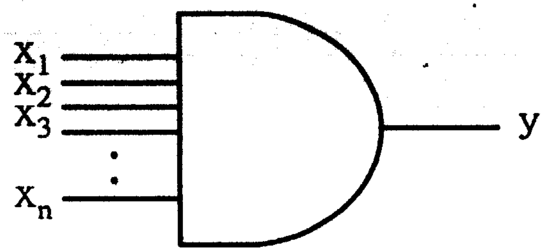
☆just need to detect each class , instead of each fault

Equivalence relation is not restricted in the same fault type

example :



Example : x=1 , y=0 , z=0 , but z should be a 1

- 

n-input gate

We have 2(n+1) signal stuck-at faults , but just need to consider only n+2 signal stuck-at faults.

$X_1 \ldots X_n$, s-a-0 ≈ y s-a-0 ∴ one is enough(y s-a-0) (1) fault

X_1 : stuck-at-1
X_2 : .
X_3 : .
.
X_n : stuck-at-1
} (n+1) faults

Y .

∴ n+2 faults

- The process of reducing faults by equivalence function is called **fault collapsing**
- fault location : discussed latter

Def : Two faults f and g are functionally equivalent

$\leftrightarrow R_f(q_{If}, T') = R_g(q_{Ig}, T')$ for any T'

Note : If faults prevent initialization

$\Rightarrow$ this relation does not hold.

Def : F and g are functionally equivalent under test sequence

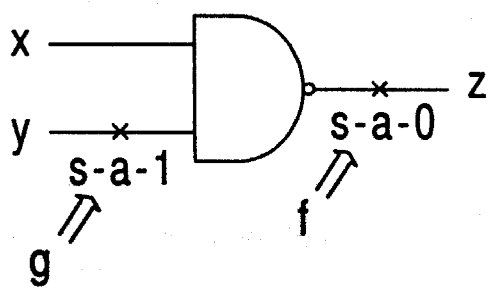
$T = \{T_I, T'\} \leftrightarrow R_f\{q_{If}, T'\} = R_g\{q_{Ig}, T'\}$

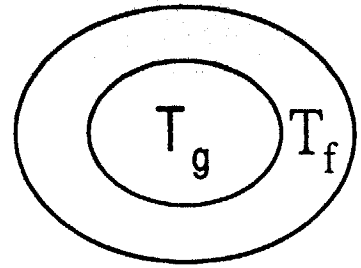
Fault Dominance

Def : Let T_g be the set of tests that detect a fault q. We say that a fault f dominate the fault q $\leftrightarrow$ f and g are functionally equivalent under T_g.

$\therefore$ If f dominates g $\Rightarrow$ any set detects g also detects f .

Example :





Tg = { 10 }

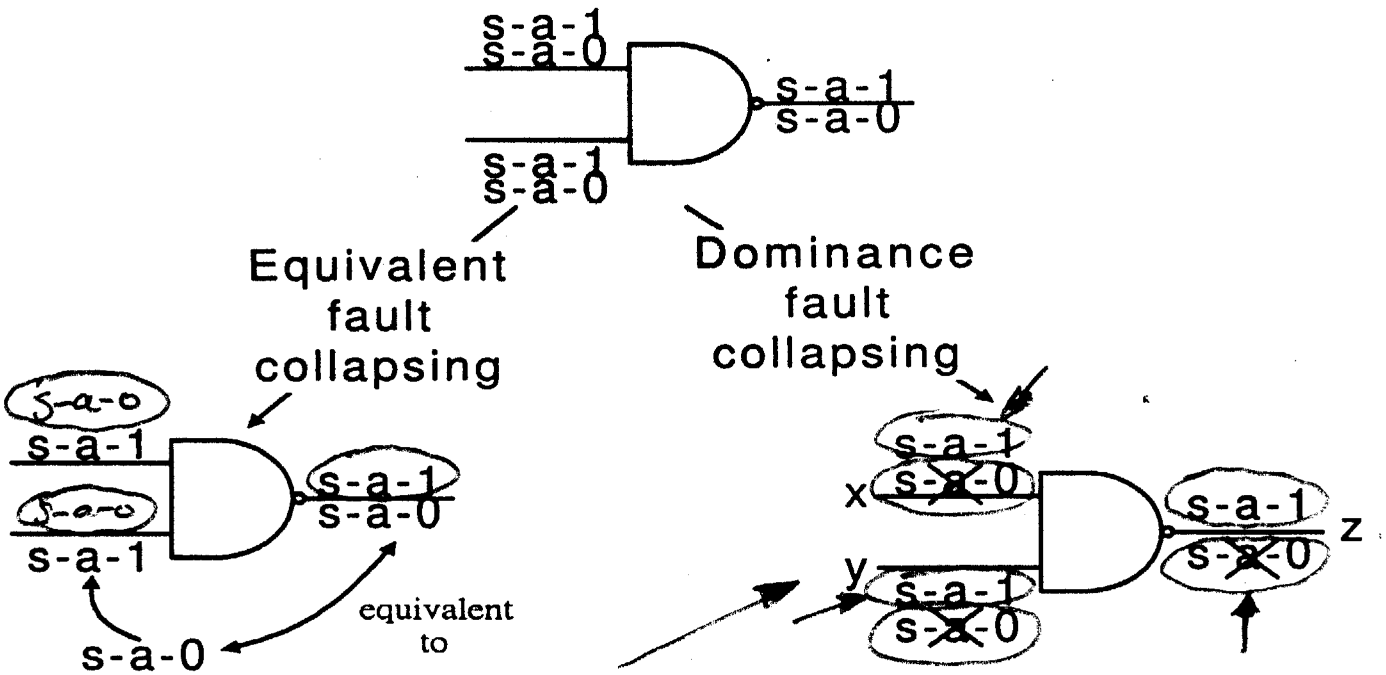
Tf = { 10 , 01 , 00 }

∴ f dominates g .

∴ f s-a-0 can be removed , since a test set detects y s-a-1 definitely detects z , s-a-0 .

This is called **dominance fault collapsing** .

Example :

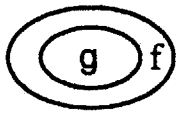


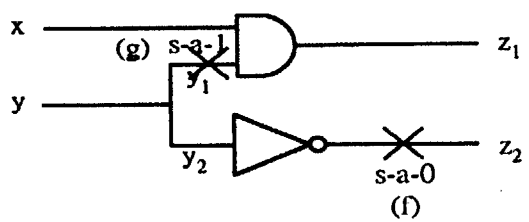
{ z s-a-0 dominates x s-a-1
=> z s-a-0 is removed .

{ x , y s-a-0 are equivalent to z s-a-1
=> x , y s-a-0 are removed .

Note : We have two faults f and g such that any test detects g also detects f , without f dominating g.

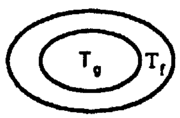
example:





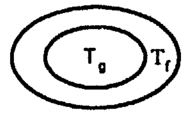
T_g = { 1 0 } also detects f

T_f = { x 0 }



but under xy = 10 , f and g are not functionally equivalent .

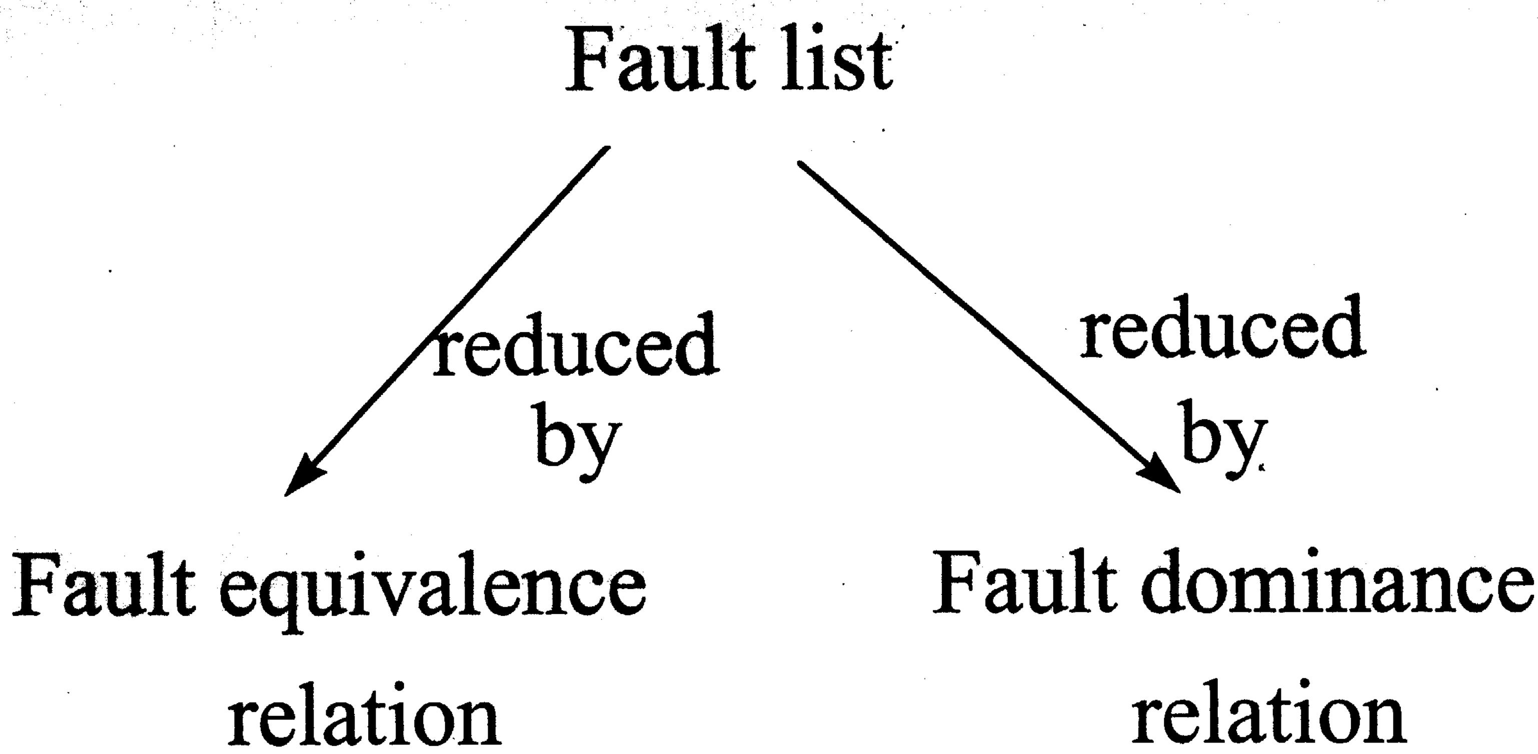
Note : f dominant g -> under T_g f and g are equivalent ⟹



does not necessarily hold !!

The Single Stuck-at Fault Model

- Most widely used model
- Represents many physical faults


Fault list
reduced by
reduced by
Fault equivalence relation
Fault dominance relation

Fault Equivalence relation

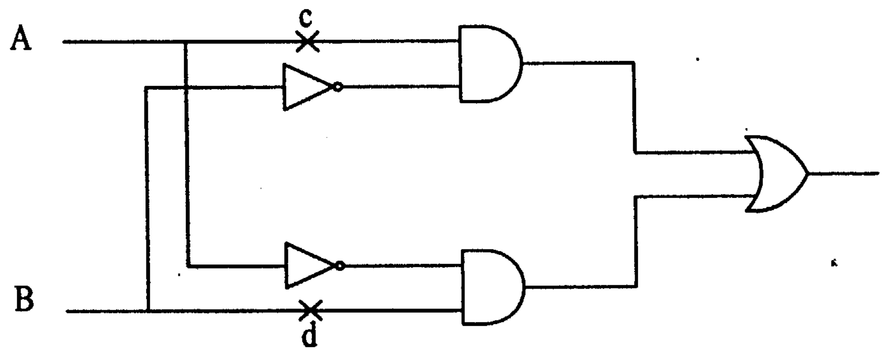
Equivalent fault

Can be discussed from Functional Equivalence Relation

Or from Structure Equivalence Relation

Functional equivalent relation detection is NP complete

Example :



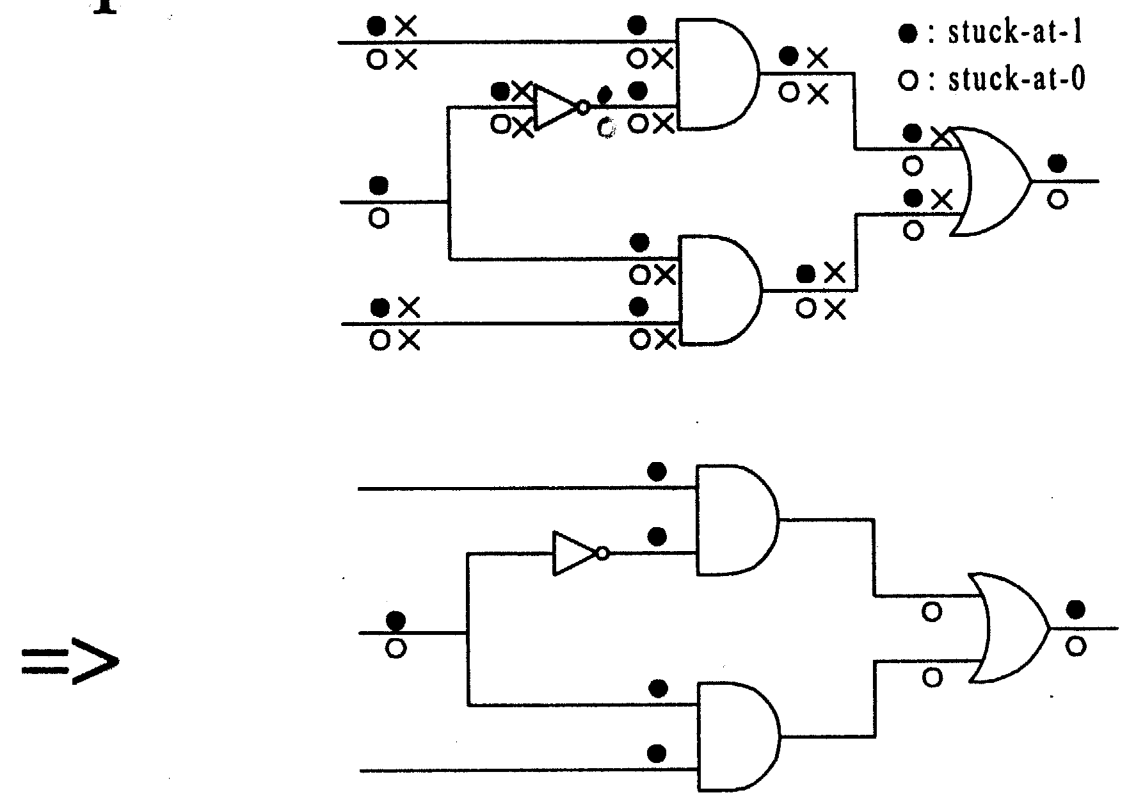
- c s-a-1 and d s-a-1 are functionally equivalent
- There is no structural relation between c and d , So no simple way to determine the equivalence of c and d s-a-1

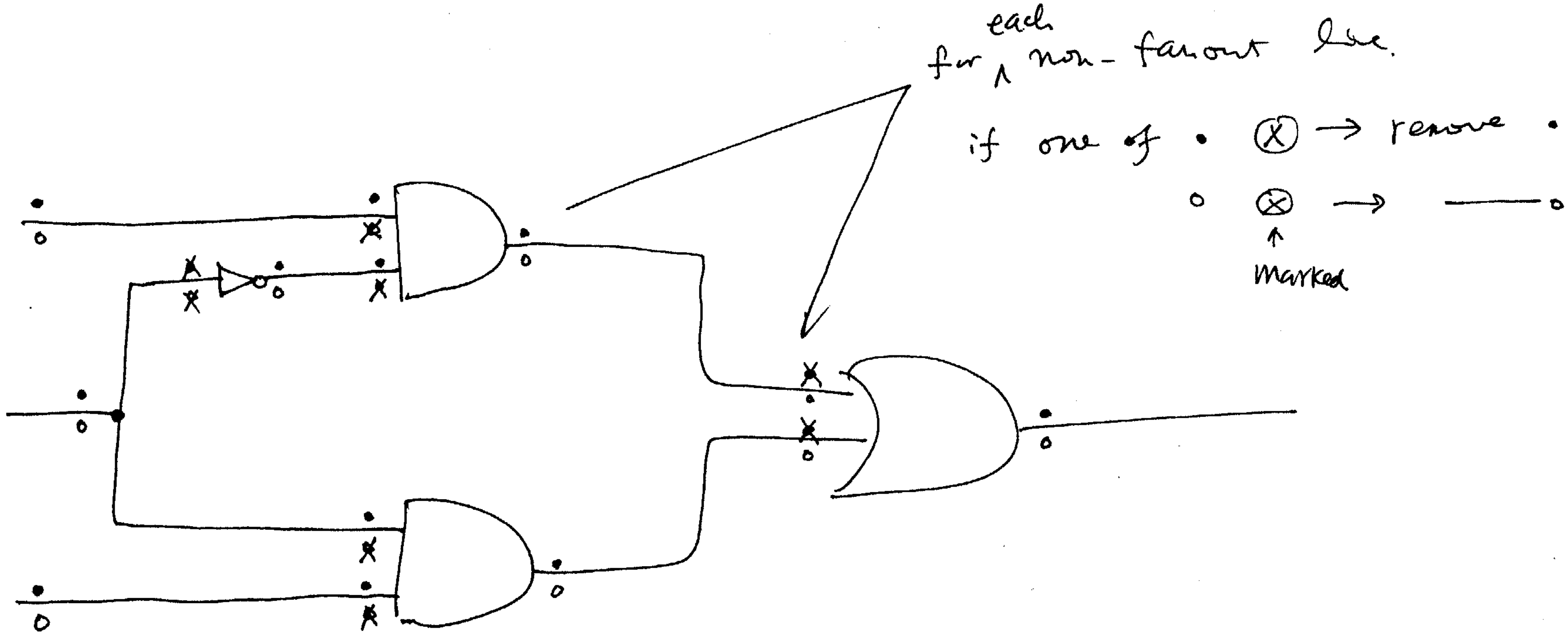
- Structure equivalence analysis: local analysis based on structure of the circuit

- Functional equivalence analysis: global analysis (entire function)

Structure Analysis

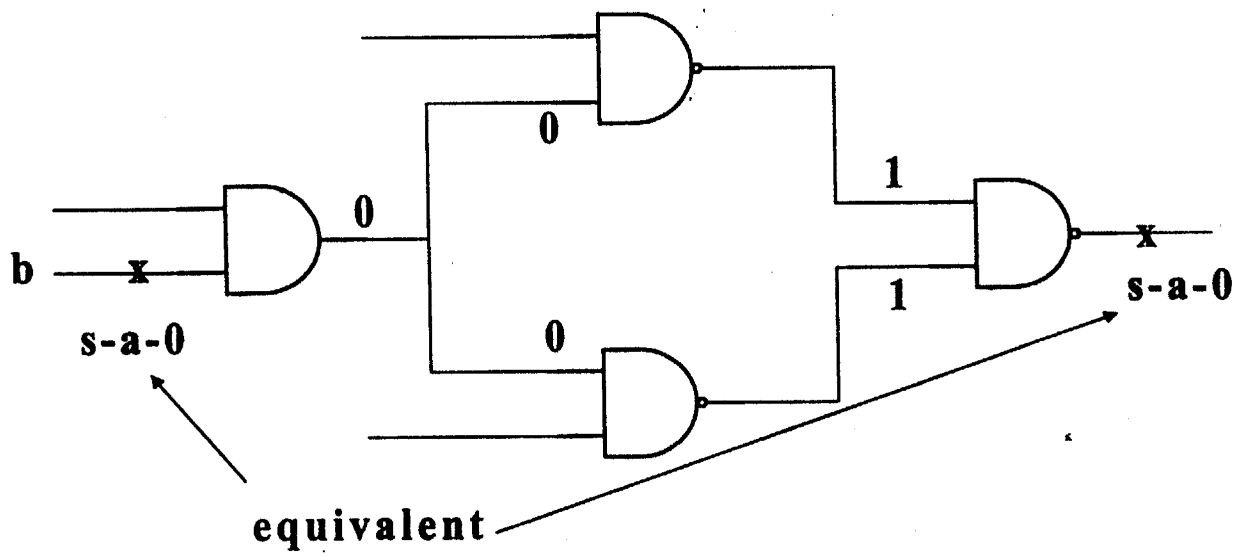
example :



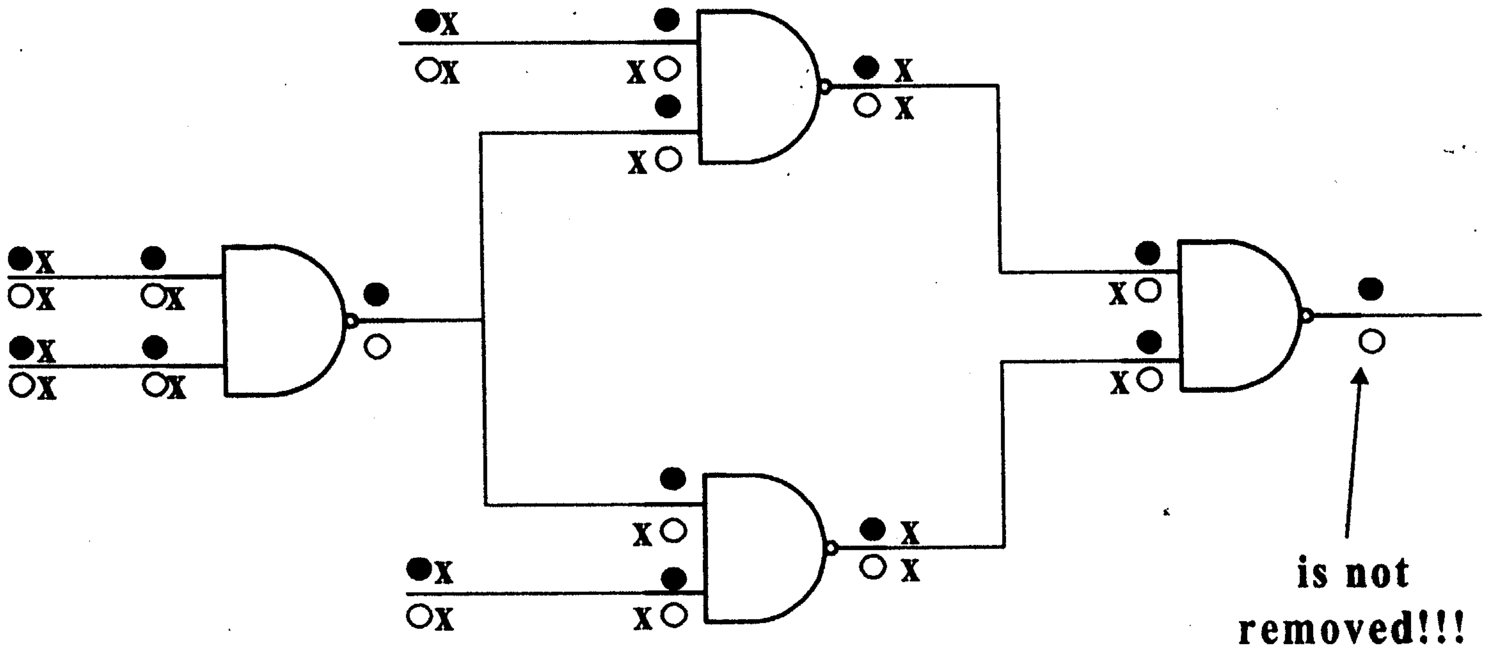


use relation of purely equivalent relation

- Example :



But, our labeling technoque can not find this, since reconvergent fanout is not taken into account.



* The process does not keep track of the fanout reconvergent information

- The obtained structural equivalence class is not maximal

- Try to expand the power of labeling technique?

  NO! The gain achievable does not justify the cost of additional cost.
  (i.e., spend so much effort, might just find very limited # of equivalent faults.)

- The structural fault collapsing reduces the initial fault list by 50% (on the average).

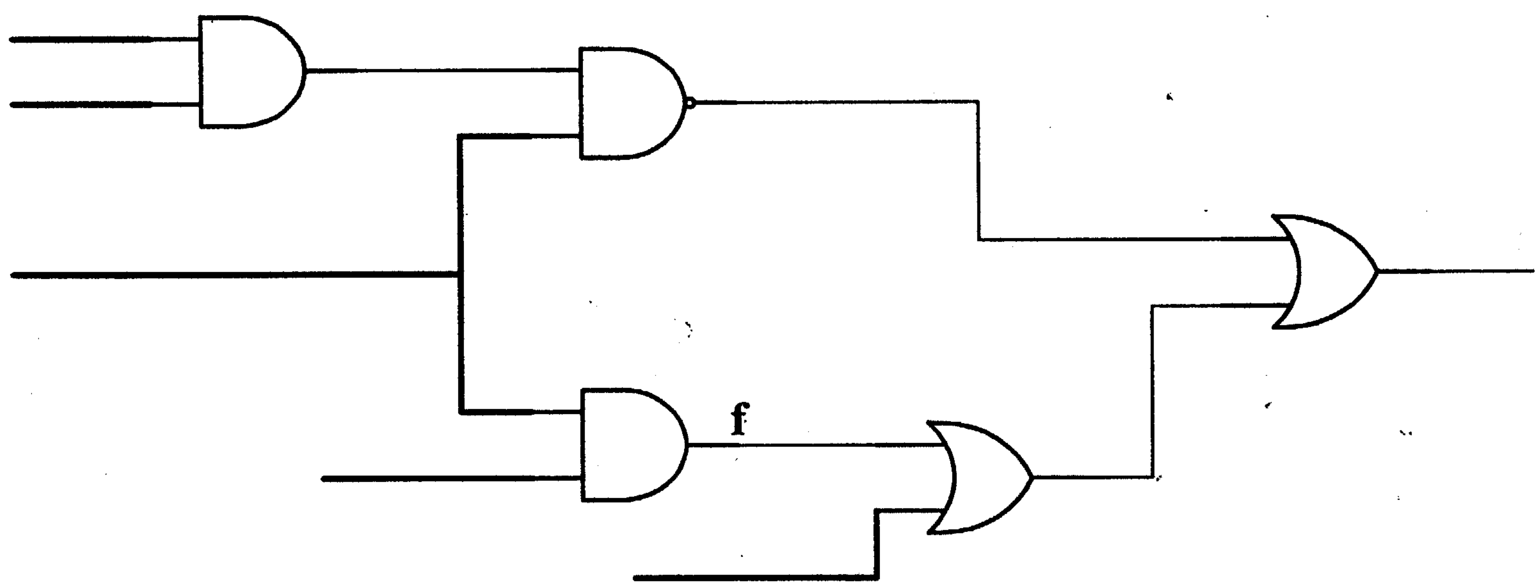
Fault Dominance Relation

Thm: In a fanout-free combinational circuit C, any test set that detects all SSF's on the primary inputs of C dectects all SSFs in C.

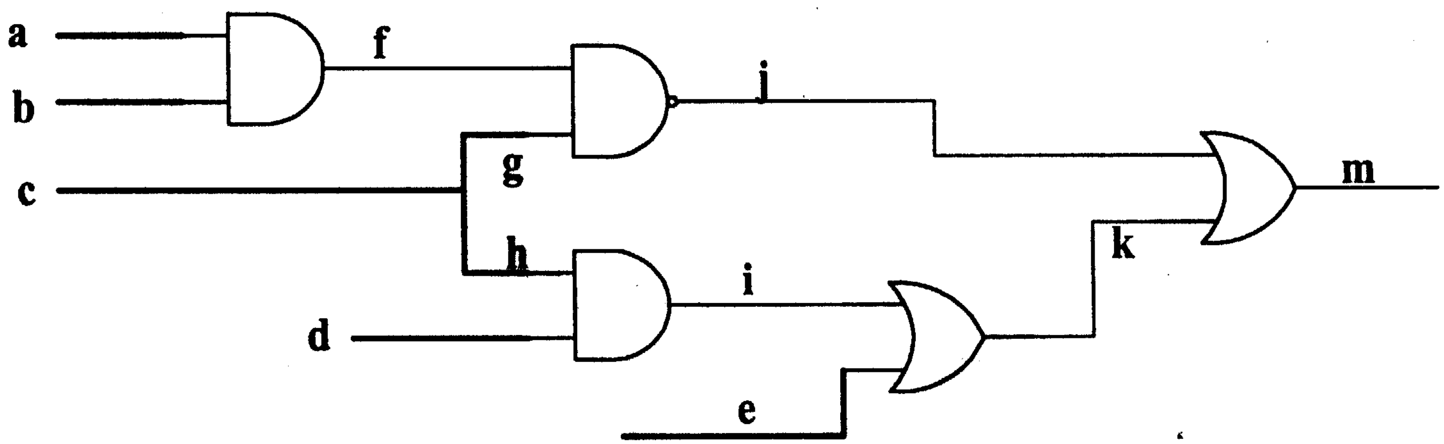
proof: use contradiction.

Thm: In a combinational circuit C, any test set that detects all SSF's on the primary inputs and the fanout branches of C dectects all SSFs in C.

proof : use contradiction



Example :



a	s-a-0 (equivalent)	s-a-1	(dominate)
b	~~s-a-0~~	s-a-1	
c	s-a-0	s-a-1	
d	~~s-a-0~~	s-a-1	
e	s-a-0 (equivalent)	~~s-a-1~~ → i: s-a-1	
g	s-a-0	~~s-a-1~~ → f: s-a-1	(dominate)
h	s-a-0	s-a-1	

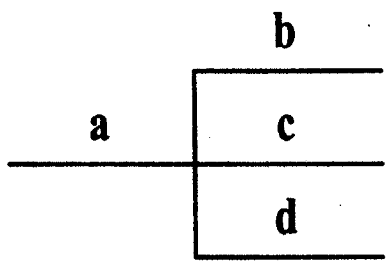
- Total 12 lines, 24 faults
- 7 checking lines, 14 checking faults
- 24 faults have been reduced to 10 faults only

Note :

- Checkpoint fault detection is only generated for irredundant circuits.
- In redundant circuit, some of the checking faults are undectable.
- If we consider only checkpoint faults and we generate a test set that detects all detectable checkpoint faults, this test set is not guaranteed to detect all detectable SSFs of the circuit; in such a case, additional tests may be needed to obtain a complete detection test set.

(∵ redundant faults)

The relations between faults on a stem line and on its branch lines



$$a \; s\text{-}a\text{-}v \approx \left.\begin{matrix} b \; s\text{-}a\text{-}v \\ c \; s\text{-}a\text{-}v \\ d \; s\text{-}a\text{-}v \end{matrix}\right\} \text{ multiple fault}$$

How about a and individual branch?

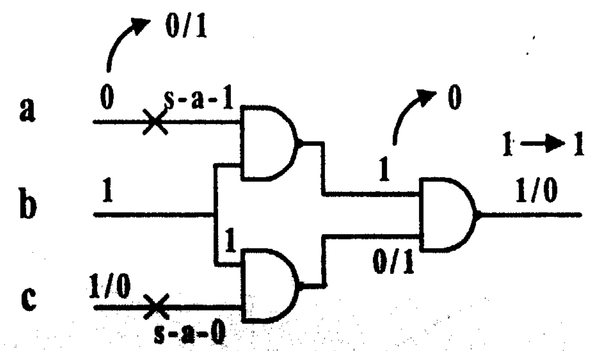
⇒ No equivalence or dominance relations

Multiple Stack-at Fault model

- Masking relations among faults.

Def : Let T_g be the set of all tests that detect a fault g , we say fault f functionally masks the fault g if and only if {f , g} is not detected by any test in T_g .

Example:



a b c = 0 1 1 is the only test pattern for c s-a-0.

$\therefore T_{c\ s-a-0} = \{0\ 1\ 1\}$

Now , if a s-a-1

$\Rightarrow T_{c\ s-a-0}$ can't detect c s-a-0

$\therefore$ a s-a-1 masks c s-a-0.

$\therefore$ No single fault test set can guarantee to detect all multiple faults.

Note: If f masks g $\Rightarrow$ the fault $\{f, g\}$ is not detected by T_g

But $\{f, g\}$ may be detected by other tests.

Example: In the above example , { a *s-a-1* , c *s-a-0*} can be detected by 010 which is not in $T_{c\ s\text{-}a\text{-}0}$.

So far: we know that

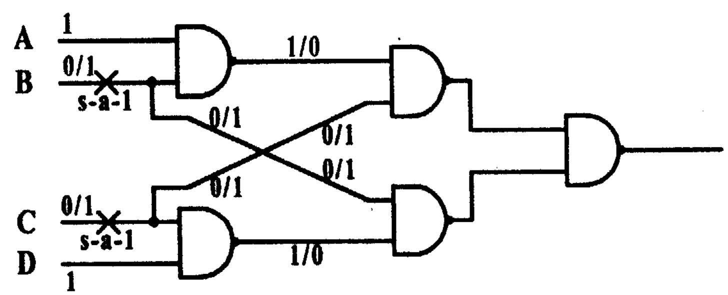
$\exists$ multiple fault $\{f, g\} \Rightarrow \{f, g\}$ is not detected by T_g .

Question:

Given a complete single fault test set T, can there exist a multiple fault $F=\{f_1, f_2, \cdots, f_k\}$ s.t. F is not detected by T ?

Answer: yes.

Circular Masking

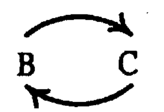


under test pattern A B C D ={ 1 0 0 1}

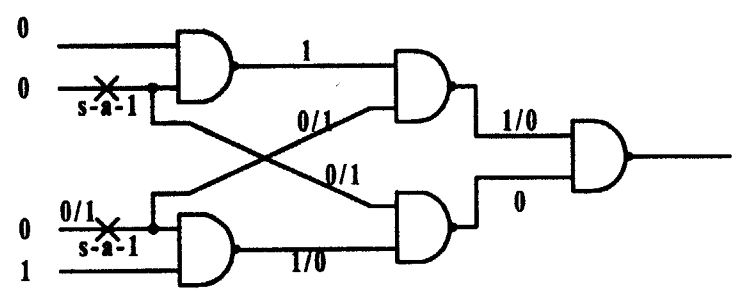
B s-a-1 masks C s-a-1

C s-a-1 masks B s-a-1

B and c are circular masking



circular



Circular masking can result in undetectable multiple stack-at faults (Multiple-line redundancy)

Example: Page 120 Breuer.

Note: circular masking is a necesarry but not sufficient condition for undetection.