Detectability

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

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

©Wen-Ben Jone No:1-9
Redundancy

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

Reason: there are some unnecessary lines or gates.

- Example:

\[ L \xrightarrow{x} \quad \text{s-a-1 is undetectable} \]

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

L → x

s-a-0 is undetectable

VLSI Testing

©Wen-Ben Jone No:1-11
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

② Two 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 ckt

Def: Two faults $f$ and $g$ are functional equivalent $\iff Z_f(x) = Z_g(x)$ 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
We have $2(n+1)$ signal stuck-at faults, but just need to consider only $n+2$ signal stuck-at faults.

$n$-input gate

$X_1 \ldots X_n, s-a-0 \approx y s-a-0 :. \text{one is enough}(y s-a-0) \bigcirc \text{fault}$

$X_1 : \text{stuck-at-1}$

$X_2 : \quad$  

$X_3 : \quad$  

$X_n : \text{stuck-at-1}$

$Y \quad$  

- 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

$$
\leftrightarrow R_f(q_{fr}, T') = R_g(q_{lg}, T') \text{ for any } T'
$$

Note: If faults prevent initialization

$$
\Rightarrow \text{this relation does not hold.}
$$

**Def:** $F$ and $g$ are functionally equivalent under test sequence

$$
T = \{T_1, T'\} \leftrightarrow R_f(q_{fr}, T') = R_g(q_{lg}, T')
$$

**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 $T_g$.

$$
\therefore \text{If } f \text{ dominates } g \Rightarrow \text{any set detects } g \text{ also detects } f.
$$
Example:

\[ T_g = \{ 10. \} \]
\[ T_f = \{ 10, 01, 00 \} \]
\[ \therefore f \text{ dominates } g. \]

\[ \therefore f \text{ s-a-0 can be removed, since a test set detects } y \text{ s-a-1 definitely detects } z, s-a-0. \]

This is called **dominance fault collapsing**.
Example:

Equivalent fault collapsing

Dominance fault collapsing

\{\begin{align*}
  z \text{ s-a-0} & \text{ dominates } x \text{ s-a-1} \\
  \Rightarrow z \text{ s-a-0} & \text{ is removed} \\
  x, y \text{ s-a-0} & \text{ are equivalent to } z \text{ s-a-1} \\
  \Rightarrow x, y \text{ s-a-0} & \text{ are removed}
\end{align*}\}
Note: We have two faults $f$ and $g$ such that any test detects $g$ also detects $f$, without $f$ dominating $g$.

example:

$$T_g = \{ 1, 0 \} \quad \text{also detects } f$$
$$T_f = \{ x, 0 \}$$

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

Note: $f$ dominant $g$ $\rightarrow$ under $T_g$, $f$ and $g$ are equivalent $\not\rightarrow$ does not necessarily hold!!
The Single Stuck-at Fault Model

- Most widely used model
- Represents many physical faults
Fault list

reduced by
Fault equivalence relation

reduced by
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:

- $c \text{ s-a-1}$ and $d \text{ s-a-1}$ are functionally equivalent
- There is no structural relation between $c$ and $d$,
  So no simple way to determine the equivalence of $c$ and $d \text{ s-a-1}$
- Structure equivalence analysis: local analysis based on structure of the circuit
- Functional equivalence analysis: global analysis (entire function)
Structure Analysis

example:

```
  ⬜️ ×
  ⬜️ ×
  ⬜️ ×
  ⬜️ ×
  ⬜️ ×

  ⬜️ :
  ⬜️ :

  o : stuck-at-1
  o : stuck-at-0

  ➔
  ➔
  ➔
  ➔

  ➔
  ➔
  ➔
  ➔
```
Use relation of purely equivalent relation.
Example:

But, our labeling technique 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 $C$ detects 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 $C$ detects all SSFs in $C$.

**proof:** use contradiction
Example:

- 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 undetectable.
- If we consider only checkpoint faults and we generate a test set that detects all detectable 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.

\[
\text{(: redundant faults)}
\]
The relations between faults on a stem line and on its branch lines

\[
\begin{array}{c}
  & b \\
  a & c & b \ s-a-v \\
    & d
\end{array}
\quad a \ s-a-v \approx c \ s-a-v
\quad d \ s-a-v
\quad \text{multiple fault}
\]

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:

\[ a \oplus b \oplus c = 0 \oplus 1 \oplus 1 \] is the only test pattern for \( c \rightarrow a \rightarrow 0 \).

\[ \therefore T_{c \rightarrow a \rightarrow 0} = \{ 0 \ 1 \ 1 \} \]

Now, if a \( s-a-1 \)

\[ \Rightarrow T_{c \rightarrow a \rightarrow 0} \text{ can't detect } c \rightarrow a \rightarrow 0 \]

\[ \therefore a \rightarrow s-a-1 \text{ masks } c \rightarrow a \rightarrow 0. \]

\[ \therefore \text{No single fault test set can guarantee to detect all multiple faults.} \]
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 \( T_c s\rightarrow_0 \).

So far: we know that

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

Question:

Given a complete single fault test set \( T \), can there exist a multiple fault \( F = \{ f_1, f_2, \ldots, f_k \} \) 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 masking can result in undetectable multiple stack-at faults (Multiple-line redundancy)

Example: Page 120 Breuer.

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