EEDG/CE 6303: Testing and Testable Design

Mehrdad Nourani

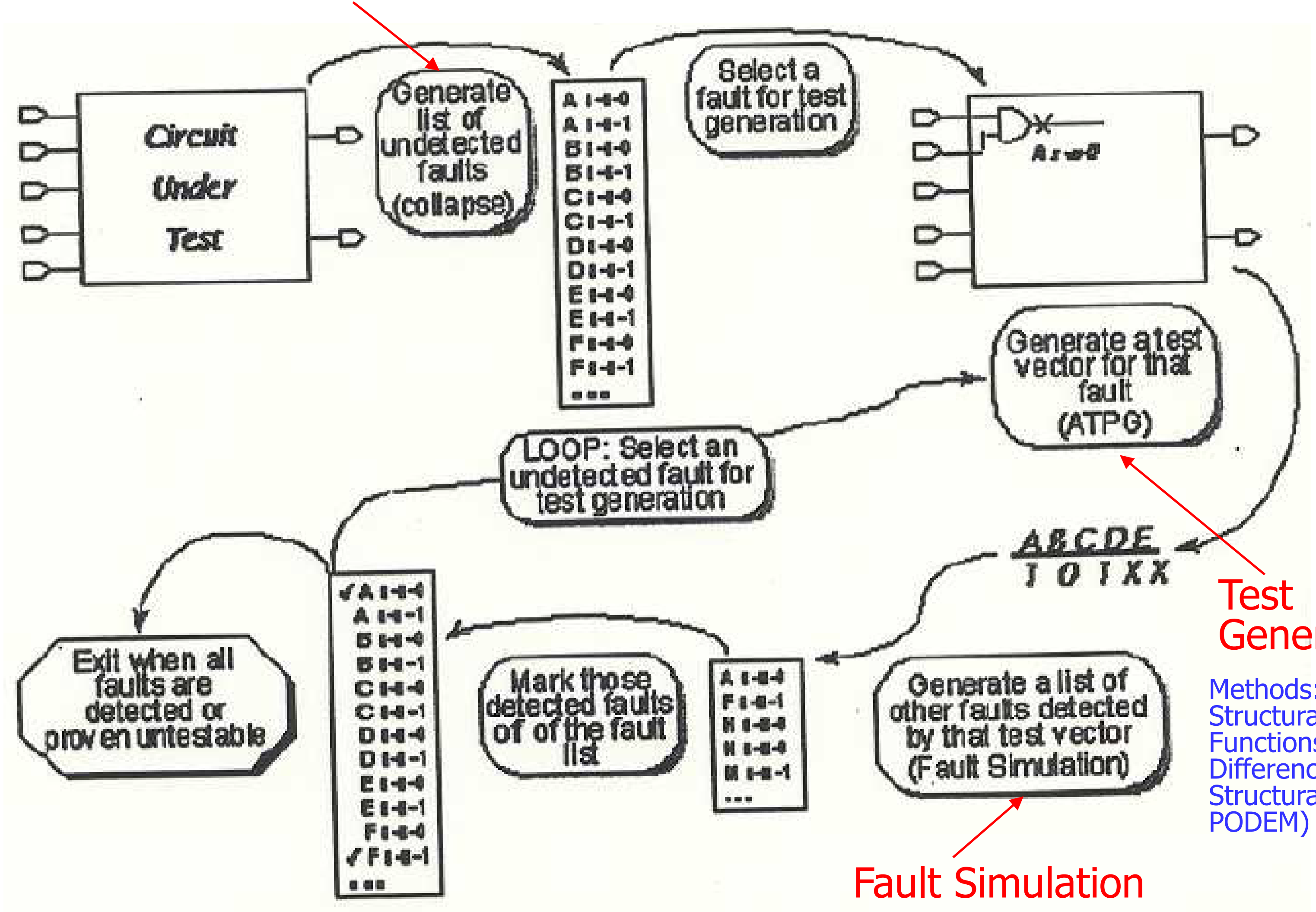
Dept. of ECE
Univ. of Texas at Dallas

Session 03

Test Generation for Combinational Circuits

Fault Analysis System (Review)

Fault Collapsing


Circuit
Under
Test
Generate list of undetected faults (collapse)
A s-a-0
A s-a-1
B s-a-0
B s-a-1
C s-a-0
C s-a-1
D s-a-0
D s-a-1
E s-a-0
E s-a-1
F s-a-0
F s-a-1
...
Select a fault for test generation
A s-a-0
Generate a test vector for that fault (ATPG)
LOOP: Select an undetected fault for test generation
A B C D E
1 0 1 X X
Generate a list of other faults detected by that test vector (Fault Simulation)
A s-a-0
F s-a-1
H s-a-0
H s-a-0
M s-a-1
...
Mark those detected faults of of the fault list
✓A s-a-0
A s-a-1
B s-a-0
B s-a-1
C s-a-0
C s-a-1
D s-a-0
D s-a-1
E s-a-0
E s-a-1
F s-a-0
✓F s-a-1
...
Exit when all faults are detected or proven untestable

Test Generation

Methods: 1. Non-Structural (Direct Functions, Boolean Difference, CNF), 2. Structural (D-Algo., PODEM)

Fault Simulation

Test Generation Techniques

- There are two main test vector generation techniques

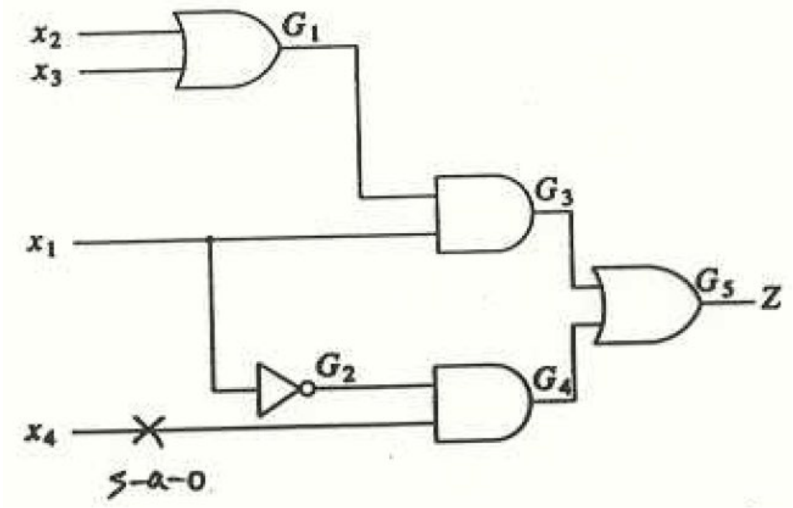
  1. Non-Structural (Analytical)
     - Analyzes the gate-level description of a circuit and implicitly enumerate all possible input combinations to find a test vector for a target fault.
     - Methods: (i) Direct Function, (ii) Boolean Difference, (iii) CNF (product-of-sum form), …

  2. Structural
     - Analyzes the structure of a given circuit to generate a test vector for a given target fault, or declare it untestable.
     - Methods: (i) D-Algorithm, (ii) PODEM, …

Non-Structural Test Generation

Fault Detection in Combinational Circuits

- A test (vector) t detects a fault f if and only if $Z_f(t) \neq Z(t)$ (i.e., at least one of the outputs are different in N and N_f).
- For a single output circuit $Z_f(t) \neq Z(t)$ is equivalent to $Z_f(t) \oplus Z(t) = 1$
- Example: find tests to detect s-a-0 at x_4.



Fault Detection – Formalization I

- Fault-free output: $Z = (x_2 + x_3) \cdot x_1 + \overline{x}_1 \cdot x_4$
- Faulty output: $Z_f = (x_2 + x_3) \cdot x_1$
- $Z \oplus Z_f = \left[(x_2 + x_3) \cdot x_1 + \overline{x}_1 \cdot x_4\right] \oplus \left[(x_2 + x_3) \cdot x_1\right] = \ldots = \overline{x}_1 \cdot x_4$
- Test vectors to detect s-a-0 at x_4 satisfy $\overline{x}_1 x_4 = 1$ that are:

x_1	x_2	x_3	x_4
0	x	x	1

$\equiv$

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

Boolean Difference

- Consider a Boolean function f(x) where x=(x_1,x_2,…,x_i,…,x_n) is the input vector.
- Cofactors of f:

$$\begin{cases} \mathrm{f}_{\mathrm{x}_\mathrm{i}} = \mathrm{f}\big|_{\mathrm{x}_\mathrm{i}=1} = f\left(x_1,\ldots,x_i=1,\ldots,x_1\right) \\ \mathrm{f}_{\overline{\mathrm{x}}_\mathrm{i}} = \mathrm{f}\big|_{\mathrm{x}_\mathrm{i}=0} = f\left(x_1,\ldots,x_i=0,\ldots,x_1\right) \end{cases}$$

- Shannon Expansion Theorem:

$$\begin{cases} \mathrm{f(x)} = \mathrm{x_i} \cdot \mathrm{f}_{\mathrm{x_i}} + \overline{\mathrm{x}}_\mathrm{i} \cdot \mathrm{f}_{\overline{\mathrm{x}}_\mathrm{i}} \\ \mathrm{f(x)} = \left(\mathrm{x_i} + \mathrm{f}_{\overline{\mathrm{x}}_\mathrm{i}}\right) \cdot \left(\overline{\mathrm{x}}_\mathrm{i} + \mathrm{f}_{\mathrm{x_i}}\right) \end{cases}$$

- Boolean Difference:

$$\frac{\mathrm{df}}{\mathrm{dx_i}} = \mathrm{f}_{\mathrm{x_i}} \oplus \mathrm{f}_{\overline{\mathrm{x}}_\mathrm{i}}$$

Fault Detection – Formalization II

- Definition of Boolean difference:

$$\frac{df}{dx_i} = f_{x_i} \oplus f_{\overline{x}_i}$$

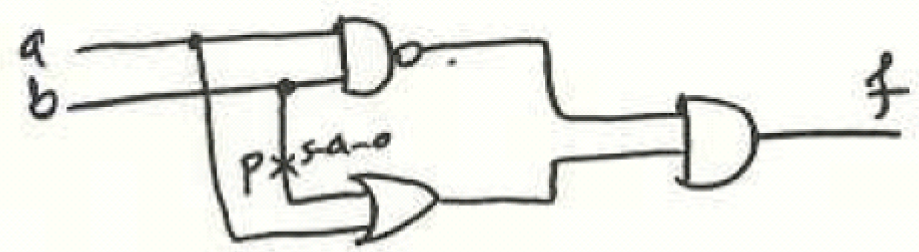
- $\frac{df}{dx_i} = 0$ means s-a-0 or s-a-1 at x_i cannot be observed at the output. In other words, s-a-f at x_i is undetectable.
- If $\frac{df}{dx_i} = 1$ then:
  - $\frac{df}{dx_i} \cdot x_i = 1$ gives all test vectors that can detect s - a - 0 at x_i
  - $\frac{df}{dx_i} \cdot \overline{x}_i = 1$ gives all test vectors that can detect s - a - 1 at x_i

Stimulate the fault

Propagate the fault effect

Boolean Difference – Formalization II (cont'd)

- Example:



$$f = \overline{ab} \cdot (a + p)$$

$$\frac{df}{dp} = f\big|_{p=0} \oplus f\big|_{p=1} = \left[\overline{ab} \cdot a\right] \oplus \left[\overline{ab}\right] = \left[(ab + \bar{a}) \cdot \overline{ab}\right] + (\overline{ab} \cdot a) \cdot ab$$

$$= ab \cdot \overline{ab} + (\bar{a} \cdot \overline{ab}) = \bar{a} \cdot (\bar{a} + \bar{b}) = \bar{a}$$

$$\Rightarrow \frac{df}{dp} \cdot p = \bar{a} \cdot b = 1 \Rightarrow ab = 01$$

- So, ab=01 is the only test vector that can detect s-a-0 at line p.

Generalization – Formalization III

- There are systematic approaches that generalize ATPG (automatic test pattern generation) analytically.

- ATPG using **satisfiability** requires two independent steps
  1. extraction of the CNF (**conjunctive normal form**, i.e. AND of OR-clauses) formula
  2. identification of a satisfying assignment

CNF for Test Generation

- The complete CNF is made of three CNF terms corresponding to

1. The fault-free circuit (CNF_g)
2. Faulty circuit (CNF_f)
3. Condition for fault detection in a single-output circuit (CNF_d)

ATPG Using Satisfiability

- **Step 1**
  - extraction of the CNF (conjunctive normal form – i.e. AND of OR-clauses) formula

0. CNF for the Fault-Free Gates

- Example (a 2-input AND):

$$v(c_j)=v(c_{i1})\cdot v(c_{i2})$$

$$[v(c_{i1})\cdot v(c_{i2})]\oplus v(c_j)=0$$

$$\overline{v(c_{i1})}\cdot v(c_j)+\overline{v(c_{i2})}\cdot v(c_j)+v(c_{i1})\cdot v(c_{i2})\cdot \overline{v(c_j)}=0$$

$$[v(c_{i1})+\overline{v(c_j)}][v(c_{i2})+\overline{v(c_j)}][\overline{v(c_{i1})}+\overline{v(c_{i2})}+v(c_j)]=1$$

c_{i1}	c_{i2}	c_j	CNF (Fault-Free)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

c_{i1}	c_{i2}	c_j	CNF (Faulty)
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	0

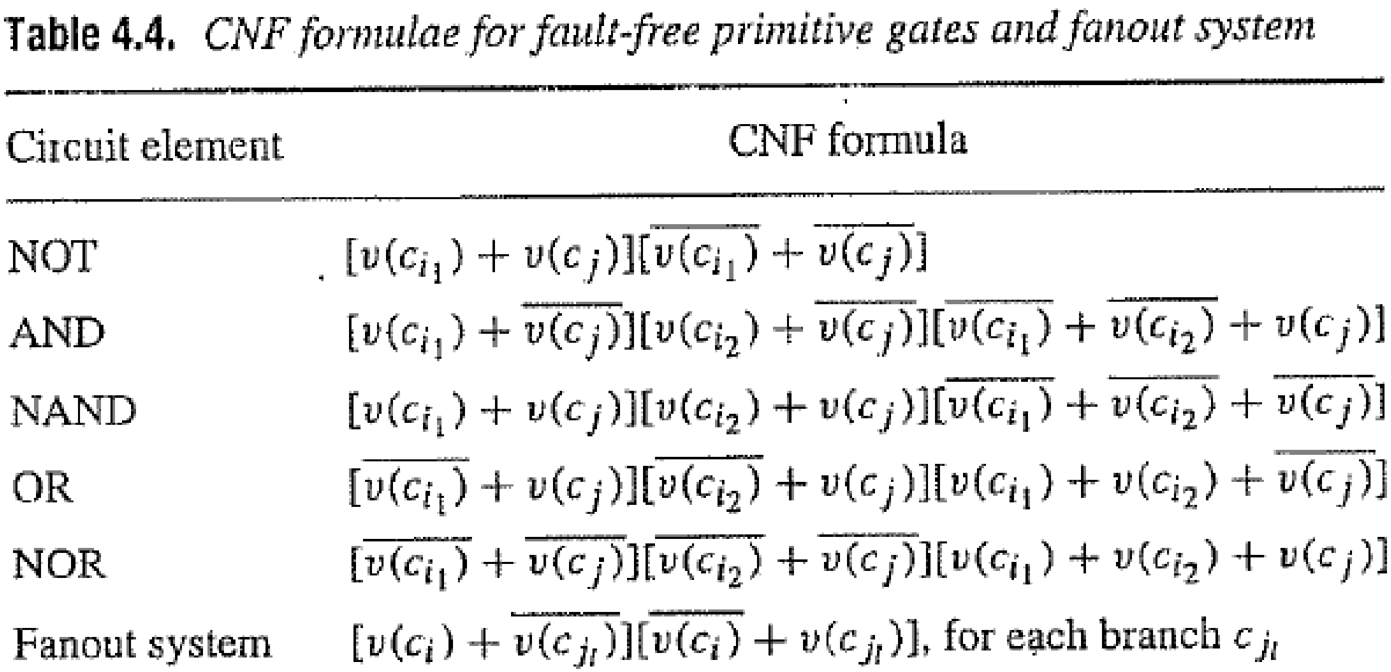
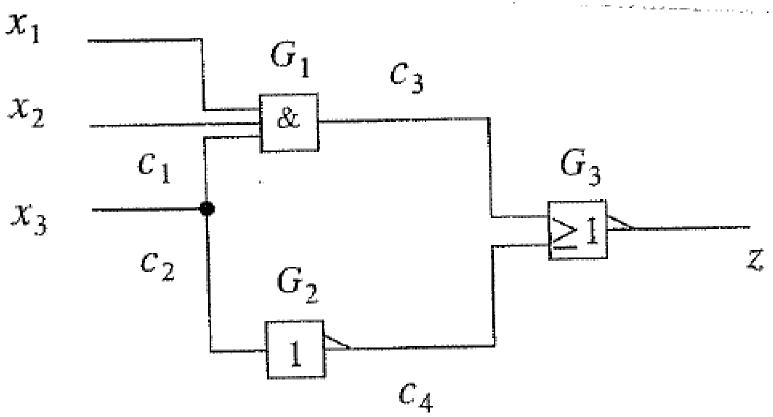


Table 4.4. *CNF formulae for fault-free primitive gates and fanout system*

Circuit element	CNF formula
NOT	$[v(c_{i_1})+v(c_j)][\overline{v(c_{i_1})}+\overline{v(c_j)}]$
AND	$[v(c_{i_1})+\overline{v(c_j)}][v(c_{i_2})+\overline{v(c_j)}][\overline{v(c_{i_1})}+\overline{v(c_{i_2})}+v(c_j)]$
NAND	$[v(c_{i_1})+v(c_j)][v(c_{i_2})+v(c_j)][\overline{v(c_{i_1})}+\overline{v(c_{i_2})}+\overline{v(c_j)}]$
OR	$[\overline{v(c_{i_1})}+v(c_j)][\overline{v(c_{i_2})}+v(c_j)][v(c_{i_1})+v(c_{i_2})+\overline{v(c_j)}]$
NOR	$[\overline{v(c_{i_1})}+\overline{v(c_j)}][\overline{v(c_{i_2})}+\overline{v(c_j)}][v(c_{i_1})+v(c_{i_2})+v(c_j)]$
Fanout system	$[v(c_i)+\overline{v(c_{j_l})}][\overline{v(c_i)}+v(c_{j_l})]$, for each branch c_{j_l}

1. CNF for the Fault-Free Circuit

- Concatenate CNFs for each individual gate and fanout branches



fanout branch c_1

fanout branch c_2

AND G1

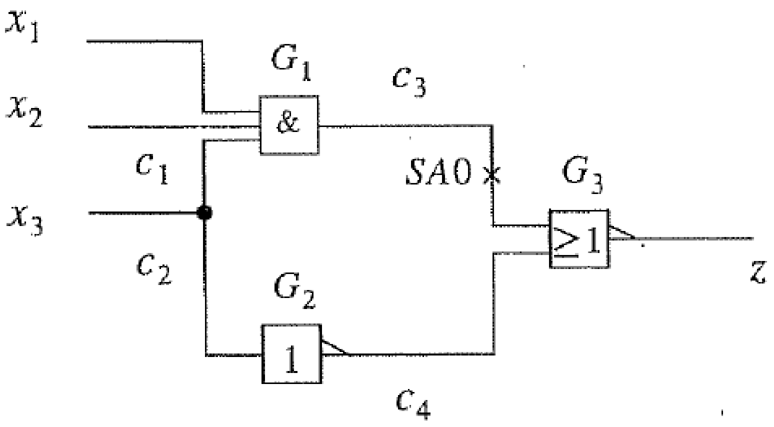
$$
\begin{aligned}
CNF_g = {} & [v(x_3)+\overline{v(c_1)}][\overline{v(x_3)}+v(c_1)] \\
& [v(x_3)+\overline{v(c_2)}][\overline{v(x_3)}+v(c_2)] \\
& [v(c_{x1})+\overline{v(c_3)}][v(c_{x2})+\overline{v(c_3)}][v(c_1)+\overline{v(c_3)}][\overline{v(c_{x1})}+\overline{v(c_{x2})}+\overline{v(c_1)}+v(c_3)] \\
& [v(c_2)+v(c_4)][\overline{v(c_2)}+\overline{v(c_4)}][\overline{v(c_3)}+\overline{v(z)}][\overline{v(c_4)}+\overline{v(z)}][v(c_3)+v(c_4)+v(z)]
\end{aligned}
$$

NOT G2

NOR G3

2. CNF for the Faulty Circuit

- It includes only the clause for the faulty gate.
  - The v' variables are introduced to indicate that lines may have different values in the fault-free and faulty circuits.



$$CNF_f = [\overline{v'(c_3)} + \overline{v'(z)}][\overline{v(c_4)} + \overline{v'(z)}][v'(c_3) + v(c_4) + v'(z)]$$

Faulty NOR G3

3. CNF for Fault Detection Condition

- We define a(c_j) as an active variable for the transitive fanout (e.g. circuit outputs) of the fault site. If a fault effect appears at c_j, then it implies the fault-free and faulty values are complement of each other and can be detected at c_j(e.g. a circuit output). Note that: (p→q)' and p.q' are equivalent.

$$a(c_j) \Rightarrow v(c_j) \neq v'(c_j)$$

$$a(c_j) \Rightarrow v(c_j) \oplus v'(c_j)$$

$$a(c_j) \cdot \overline{[v(c_j) \oplus v'(c_j)]}$$

$$a(c_j) \cdot [v(c_j) \cdot v'(c_j) + \overline{v(c_j)} \cdot \overline{v'(c_j)}]$$

$$[a(c_j) \cdot \overline{v(c_j)} \cdot \overline{v'(c_j)}] + [a(c_j) \cdot v(c_j) \cdot v'(c_j)]$$

- Finally, complementing it again:

$$CNF_a(c_j) = [\overline{a(c_j)} + v(c_j) + v'(c_j)][\overline{a(c_j)} + \overline{v(c_j)} + \overline{v'(c_j)}]$$

3. CNF for Fault Detection Condition (cont.)

- The condition under which the fault effect appears in the output z requires an active clause that says the faulty and fault-free values on z will be different.

$$CNF_d = \overbrace{CNF_a(z)}^{\substack{\text{Necessary Condition} \\ \text{(in output v(z)\#v'(z))}}} \cdot \overbrace{a(z)}^{\substack{\text{Sufficient Condition} \\ \text{(Activate \& propagate to z)}}}$$

$$= [\overline{a(z)} + v(z) + v'(z)][\overline{a(z)} + \overline{v(z)} + \overline{v'(z)}] \cdot a(z)$$

- Overall, any assignment of primary inputs that satisfies this CNF formula is a test vector for the target fault in the single-output circuit.

$$CNF_g \cdot CNF_f \cdot CNF_d$$

3. CNF for Fault Detection Condition (cont.)

- Extension to multi-output circuit is straightforward.
- The only part that needs change is CNF_d

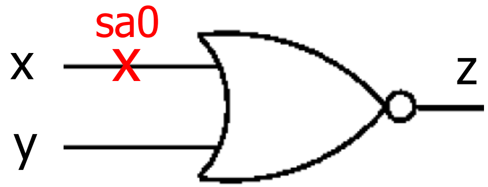
$$CNF_d = CNF_a(z_1) \cdot CNF_a(z_2) \cdots CNF_a(z_m)[a(z_1) + a(z_2) + \cdots + a(z_m)]$$

- Again, any assignment of primary inputs that satisfies this CNF formula is a test vector for the target fault in the multi-output circuit.

$$CNF_g \cdot CNF_f \cdot CNF_d$$

A Small Example of CNF

- Consider a NOR gate (as a small circuit)
  - After collapsing, there are 3 faults: x sa0, y sa0, y sa1.
- For fault x sa0:



$$CNF_g = [\overline{x} + \overline{z}][\overline{y} + \overline{z}][x + y + z]$$

$$CNF_f = [\overline{x'} + \overline{z'}][\overline{y} + \overline{z'}][x' + y + z']$$

$$CNF_d = [\overline{a} + z + z'][\overline{a} + \overline{z} + \overline{z'}][a]$$

- To find the pattern we need to satisfy:

$$CNF_g \cdot CNF_f \cdot CNF_d = 1$$

- Only: a=1, x=1, x′=0, y=0, z=0, z′=1 satisfies this CNF, which means there is only one pattern xy=10.

ATPG Using Satisfiability

- **Step 2**
  - identification of a satisfying assignment
  - Binary Decision Diagram (BDD) is often used