EEDG/CE 6303: Testing and Testable Design

Mehrdad Nourani

Dept. of ECE
Univ. of Texas at Dallas

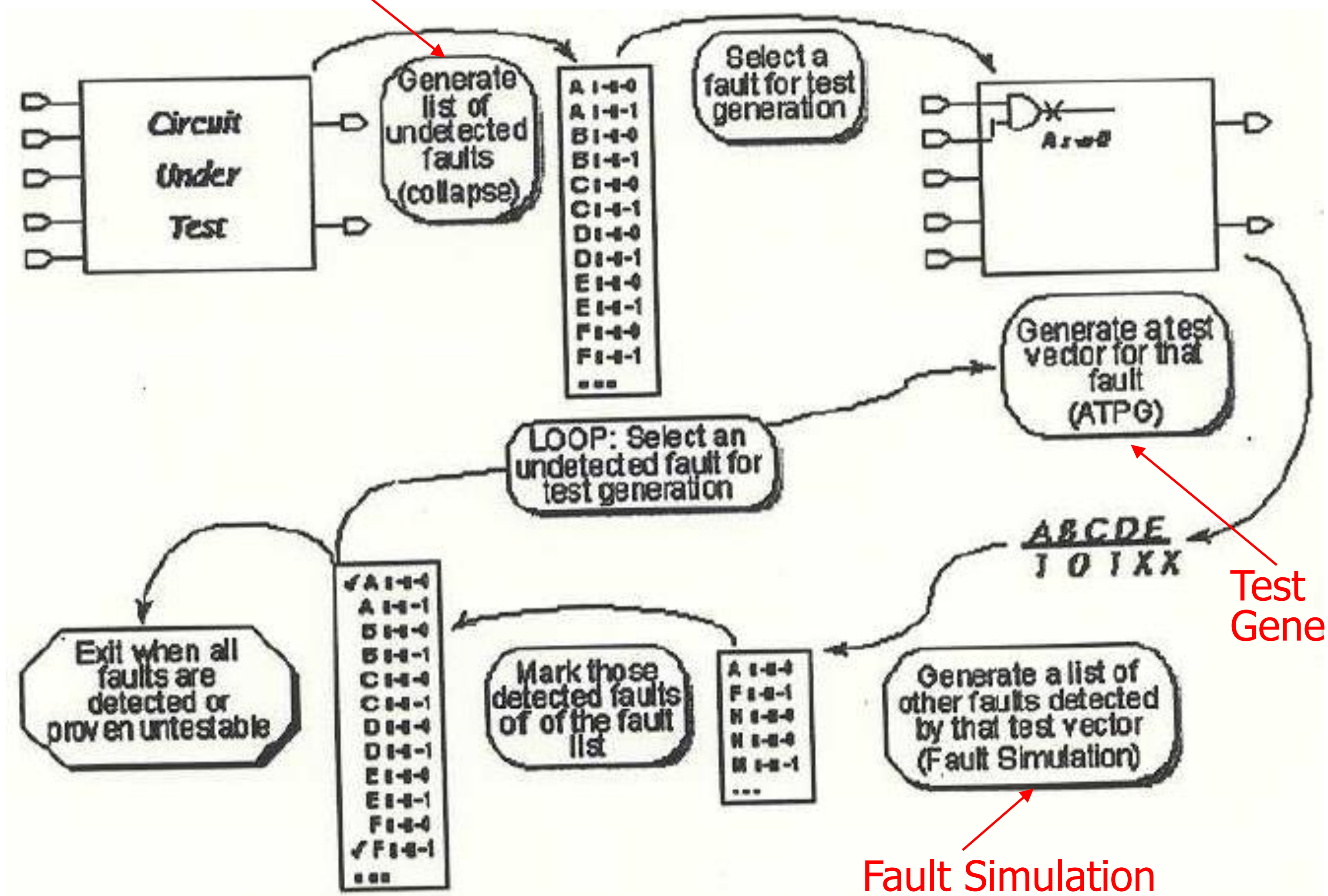
Session 02

Fault Modeling

Fault Analysis System

Methods: 1. Level-by-Level, 2. Checkpoint Theorem

Fault Collapsing



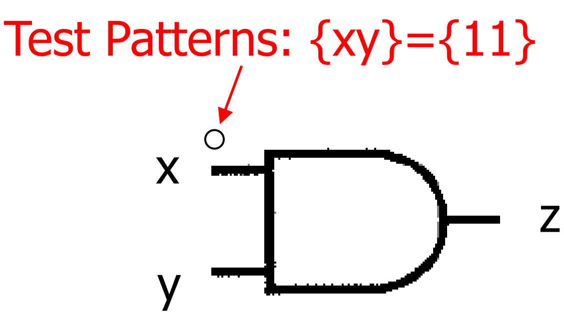
Test Generation

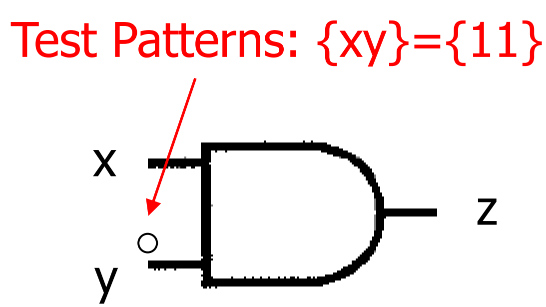
Fault Simulation

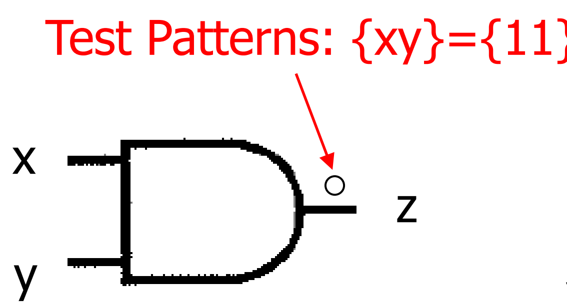
Fault Collapsing

Fault Equivalence

- Two faults f and g are functionally equivalent iff $z_f(t) = z_g(t)$ under any test set T ($t \in T$).
- Test (vector) t distinguishes between two faults f and g if $z_f(t) \neq z_g(t)$ (i.e. $z_f(t) \oplus z_g(t) = 1$ for a single-output function).
- Equivalency and distinguishability are defined for the faults of the same nature (fault-universe).
- Example (consider sa0 faults):

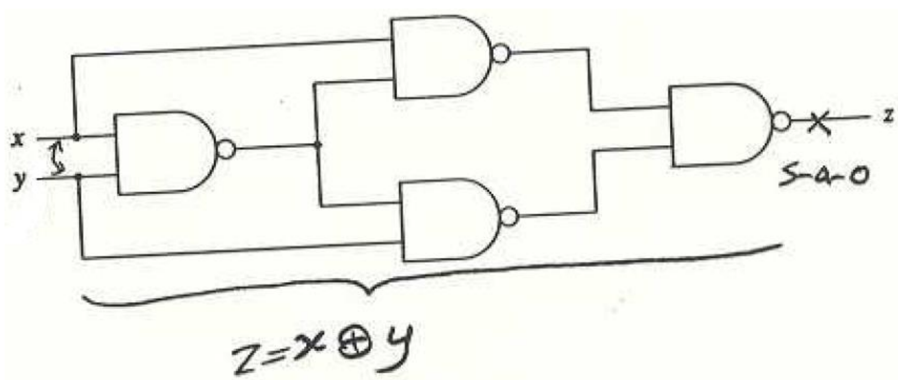






Fault Equivalence (cont'd)

- Equivalency and distinguishability are defined for the faults of the same nature (fault-universe). For example, bridging fault between x and y is equivalent to s-a-0 on Z, but equivalency is not considered.

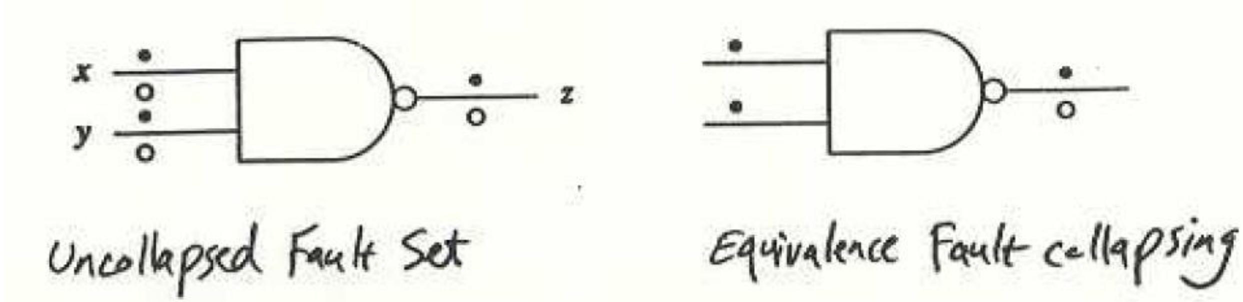


Fault Equivalence (cont'd)

- For a gate with controlling value c and inversion i, all "s-a-c on input" faults and "s-a-(c⊕i)" on outputs are equivalent.

gate	c	i
AND	0	0
NAND	0	1
OR	1	0
NOR	1	1

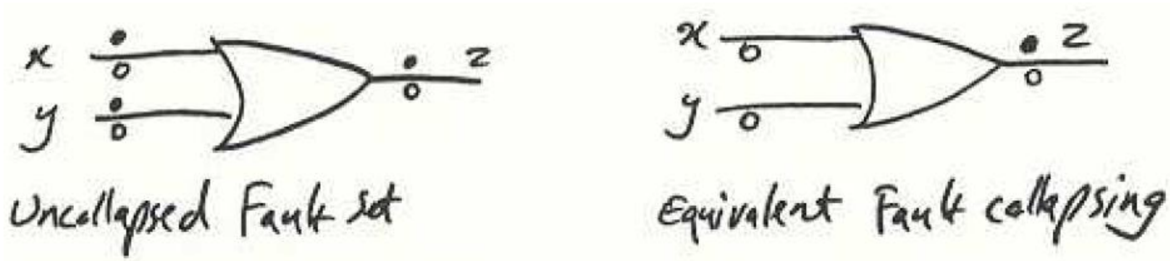
- Example:
  - s-a-1: •
  - s-a-0: ◦



  - For NAND: s-a-0 on x ≡ s-a-0 on y ≡ s-a-1 on z
  - Test pattern xy=11 can detect all of them.

Fault Equivalence (cont'd)

- Example:

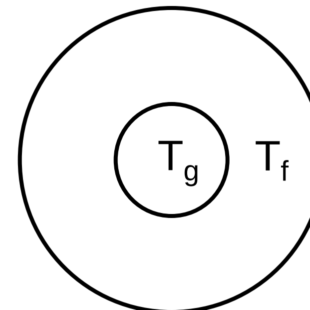


  - For OR: s-a-1 on x ≡ s-a-1 on y ≡ s-a-1 on z
  - Test pattern xy=00 can detect all of them.

- An n-input, one-output gate (n>1) has totally 2(n+1) s-a-x faults. Due to equivalency, we need to consider only (n+2) faults.

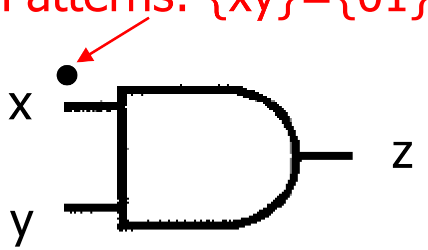
Fault Dominance

- Let T_f and T_g be the set of all tests that detect fault f and g, respectively. These are equivalent statements:
  - Fault f dominates g ($g \subseteq f$)
  - f and g are functionally equivalent under T_g
  - $T_g \subseteq T_f$
  - Any test that detects g, i.e. $z_g(t) \neq z(t)$ will also detect f on the same primary output because $z_f(t) = z_g(t)$

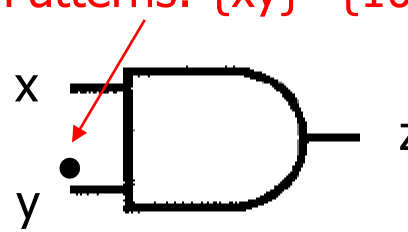


- To reduce efforts in testing we can identify and remove the dominating faults (such as f) from the set of faults. Example (consider sa1 faults):

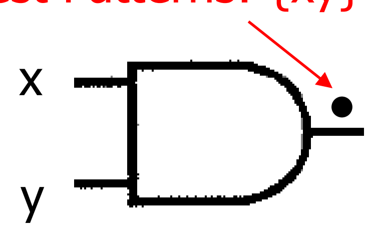
Test Patterns: {xy}={01}



Test Patterns: {xy}={10}

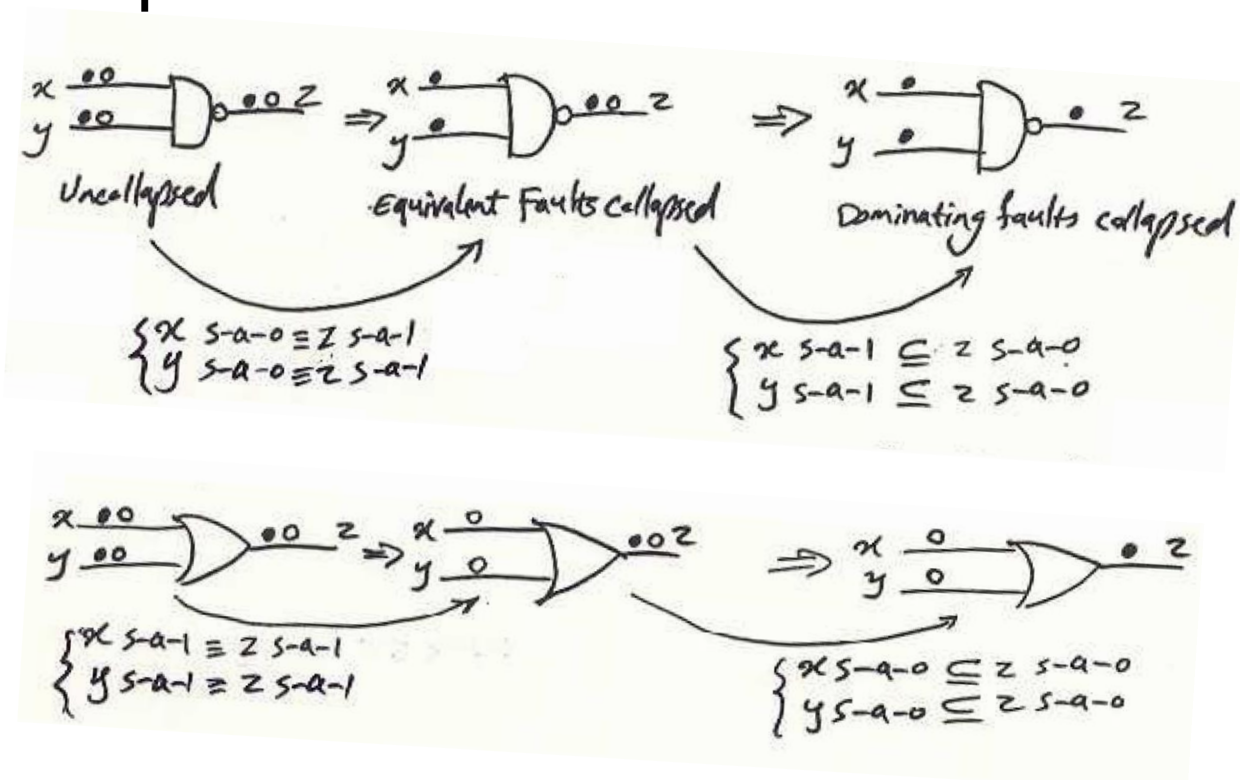


Test Patterns: {xy}={00,01,10}



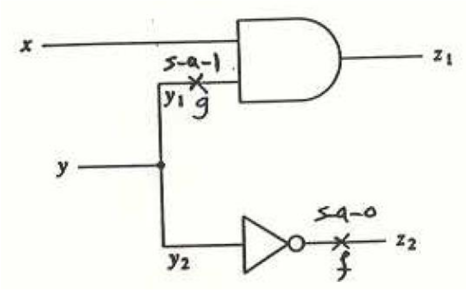
Fault Dominance (cont'd)

- For a gate with controlling value c and inversion i, $s\text{-}a\text{-}(\bar{c}\oplus i)$ on the output dominates $s\text{-}a\text{-}\bar{c}$ on any input.
- Example:



Fault Dominance (cont'd)

- In general, f dominates g => $T_g \subseteq T_f$ but the reverse may not be true.
- Example:



  - f: z_2 s-a-0
  - g: y_1 s-a-1
  - $T_g = \left\{ \begin{array}{c} xy \\ \hline 10 \\ \\ \end{array} \right\}$ & observation point: z_1
  - $T_f = \{00, 10\}$ & observation point: z_2
  - According to the definition, since observation points are different f does not dominate g.
  - In this example for fault detection, it is not necessary to consider f.

Removing Equivalent/Dominant Faults

- We can remove all equivalent faults except one as representative.
- The dominating faults can be removed.
- In general, equivalency does not have transitive property. That is, if $f \equiv g$ and $g \equiv h$, we cannot conclude that $f \equiv h$. Why?
- Rules for gates:
  - AND, OR, NAND, NOR:

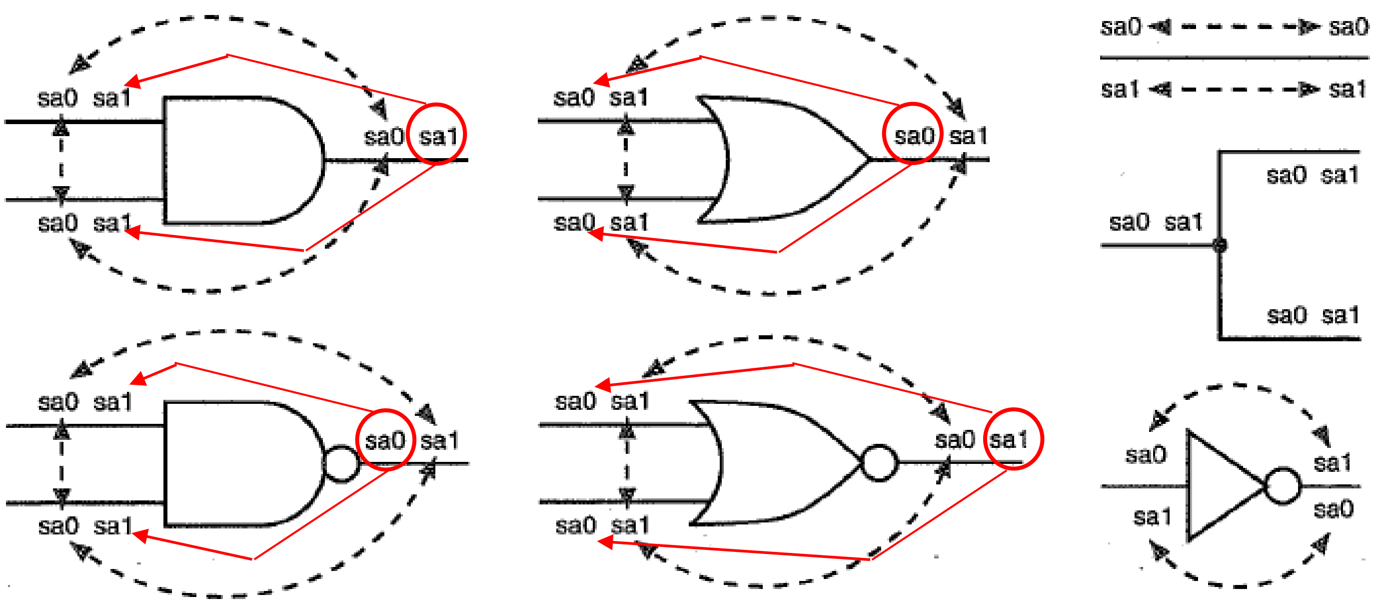
$$\begin{cases} \text{input s-a-}c \equiv \text{output s-a-}(c \oplus i) & \Rightarrow \text{remove all input s-a-}c \\ \text{input s-a-}\bar{c} \subseteq \text{output s-a-}(\bar{c} \oplus i) & \Rightarrow \text{remove all output s-a-}(\bar{c} \oplus i) \end{cases}$$

  - Inverter:

$$\begin{cases} \text{input s-a-0} \equiv \text{output s-a-1} \\ \text{input s-a-1} \equiv \text{output s-a-0} \end{cases}$$

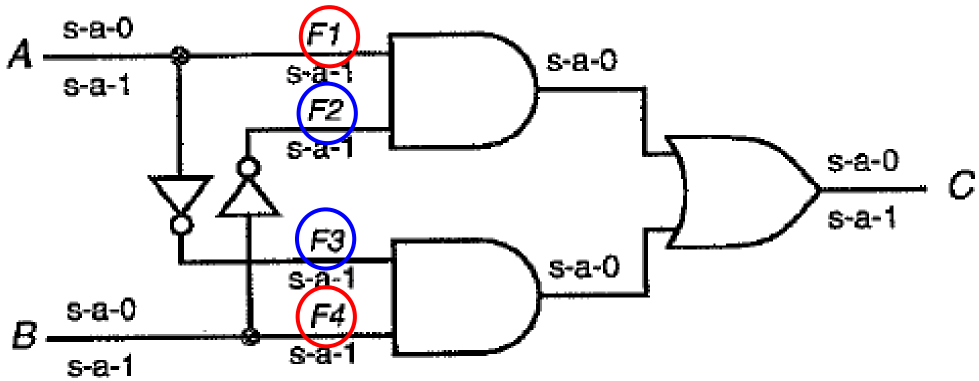
gate	c	i
AND	0	0
NAND	0	1
OR	1	0
NOR	1	1

Summary of Equivalent/Dominant Faults



What about XOR Gate?

- Consider only equivalence:

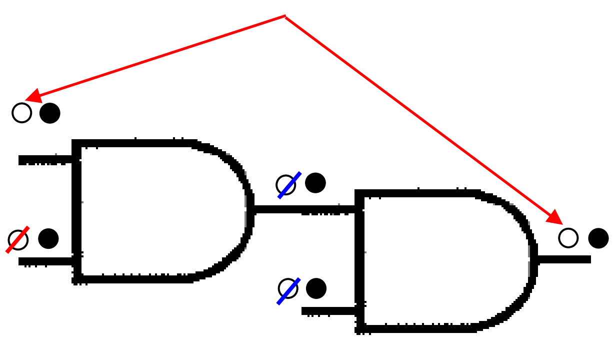


- When XOR is considered as a module, all 6 faults should be included. Why?
  - Hint: Write all test patterns for each and every one of those six faults (one by one).

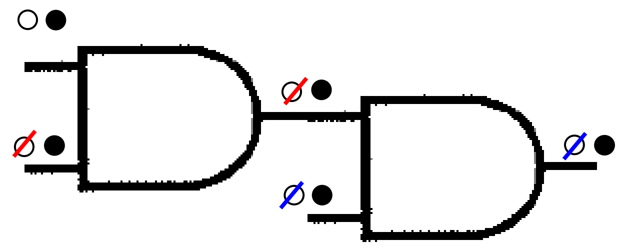
Caution in Fault Collapsing

- Try to work on faults in one direction. For example, start from the output gate, apply both equivalency and dominancy to the gate and and remove as many faults as possible in the output (keep the rest in the input) before going to the next level.
- If you work on both directions when removing fault, some equivalency/dominancy may not be seen.
- Example (consider only fault equivalence):

Fault equivalence
may not be seen



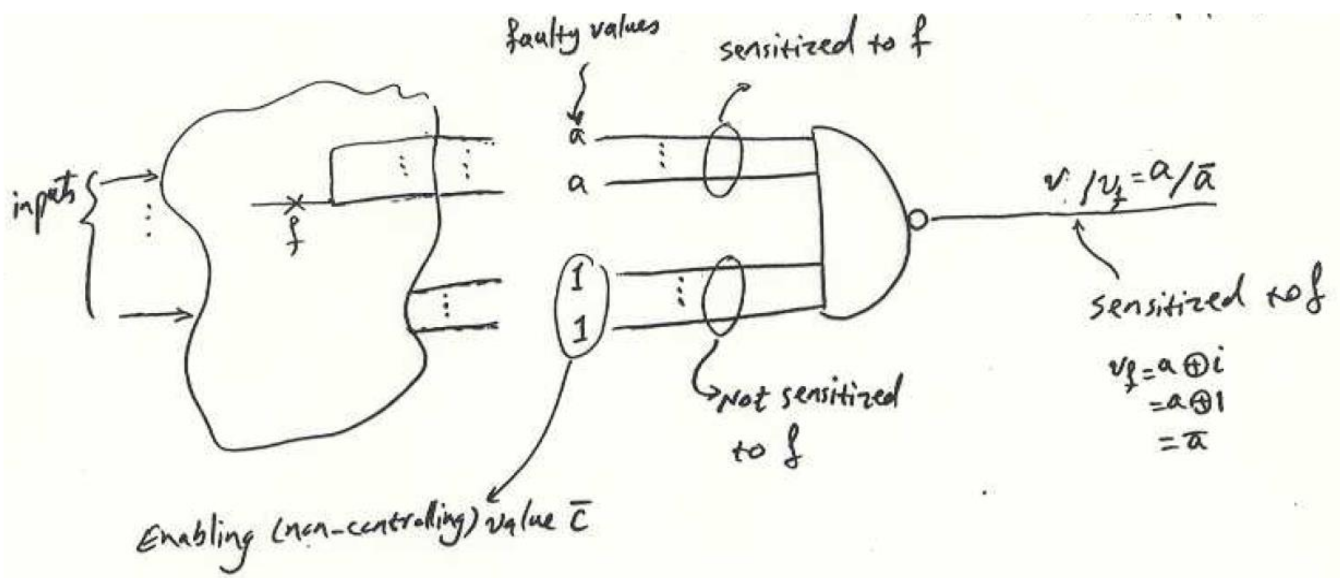
Step 1: /
Step 2: /



Propagating the Fault Effect

- Let gate G be a gate with controlling value c and inversion value i whose output is sensitized to a fault f (by a test t). Then:
  - All inputs of G sensitized to f have the same value (e.g. a∈ {0,1}).
  - All inputs of G not sensitized to f (if any) have value c'.
  - The output of G has value a⊕i.

gate	c	i
AND	0	0
NAND	0	1
OR	1	0
NOR	1	1



Level-Based Analysis

- Procedure **OutputLevelize()** can compute $\eta_{inp}(c_i)$ for all lines

Procedure 3.2: ~~InputLevelize()~~ output

1. *Initialization* : For each circuit line c, η_{inp} = undefined.

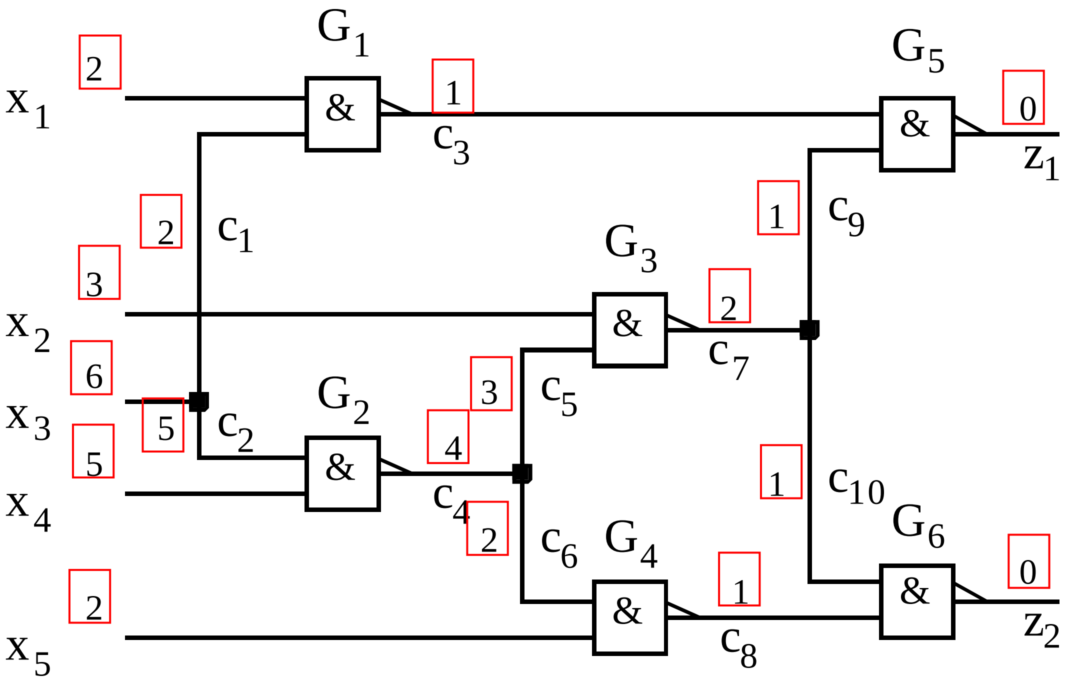
2. For each primary ~~input~~ output x_i, $\eta_{inp} = 0$.

3. While there exists one or more logic elements such that (i) η_{inp} is defined for each of the element's ~~inputs~~ outputs, and (ii) η_{inp} is undefined for any of its ~~outputs~~ inputs, select one such element.

   (a) If the selected element is a ~~gate~~ Fanout with ~~inputs~~ branches c_{i_1}, c_{i_1}, ..., c_{i_α} and ~~output~~ stem c_j, then assign $\eta_{inp}(c_j) = \max[\eta_{inp}(c_1), \eta_{inp}(c_2), \ldots, \eta_{inp}(c_\alpha)] + 1.$

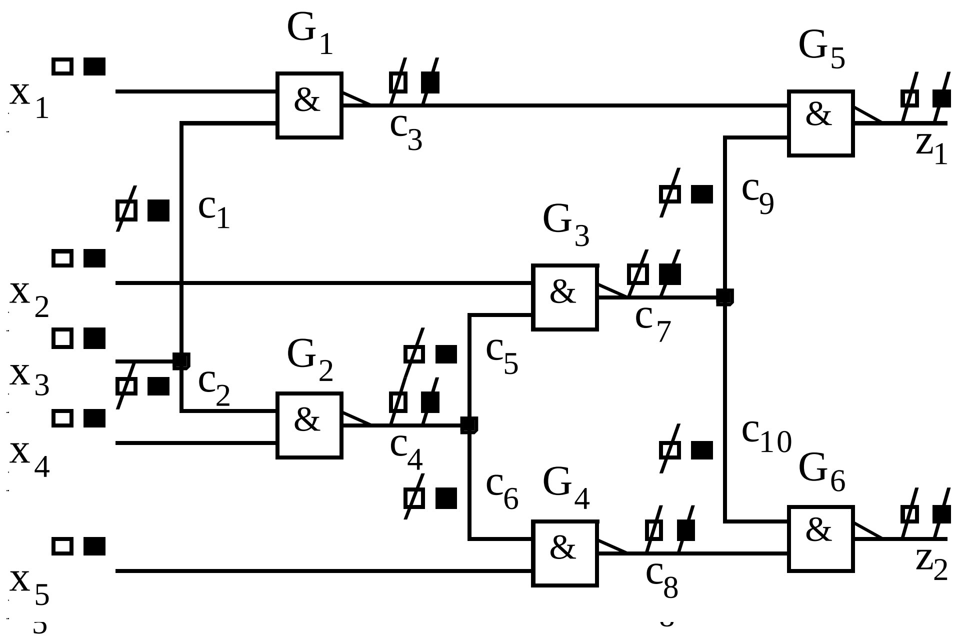
   (b) If the selected element is a ~~fanout~~ gate ~~system~~ with ~~stem~~ output c_i and ~~branches~~ inputs c_{j_1}, c_{j_2}, ..., c_{j_β}, then for each ~~output~~ input c_{j_l}, where $l = 1, 2, \ldots, \beta$, assign $\eta_{inp}(c_{j_l}) = \eta_{inp}(c_i) + 1.$

Fault Collapsing (Dropping) – An Example



- Circuit lines ordered in non-decreasing order of their output level values as $\{z_2, z_1, c_{10}, c_9, c_8, c_3, c_7, c_6, c_1, x_5, x_1, c_5, x_2, c_4, c_2, x_4, x_3\}$

Fault Collapsing (Dropping) – An Example



- Circuit lines ordered in non-decreasing order of their output level values as $\{z_2, z_1, c_{10}, c_9, c_8, c_3, c_7, c_6, c_1, x_5, x_1, c_5, x_2, c_4, c_2, x_4, x_3\}$
- Collapse Ratio=$N_{\text{remaining faults}}/N_{\text{all\_faults}}$ =16/34=0.47
- Note: Faults remain only at primary inputs and fanout branches (Why?)

Fault Collapsing in Benchmark Circuits

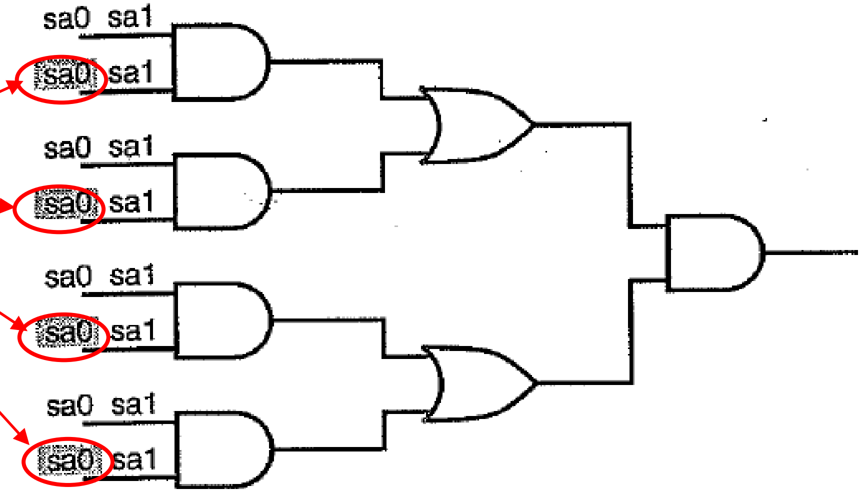
Has a lot of XORs

Circuit name	No. of gates	No. of inputs	No. of outputs	Number of faults		
				All	Collapsed	Collapse ratio
c432	160	36	7	864	524	0.61
c499	202	41	32	998	758	0.76
c880	383	60	26	1,760	968	0.55
c1355	546	41	32	2,710	1,606	0.59
c1908	880	33	25	3,816	2,041	0.54
c2670	1,193	233	140	5,340	2,943	0.55
c3540	1,669	50	22	7,080	3,651	0.52
c5315	2,307	178	123	10,630	5,663	0.53
c7552	3,513	207	108	15,104	8,084	0.54
s27	10	4	1	52	32	0.62
s9234	5,597	19	22	10,572	3,862	0.37
s38584	19,257	12	278	78,854	36,303	0.47

Fault Detection Theorem

- A test set that detects all single stuck-at faults on all primary inputs of a **fanout-free circuit** must detect all single stuck-at faults in that circuit.
- Example:

Further fault dropping after applying the theorem is possible



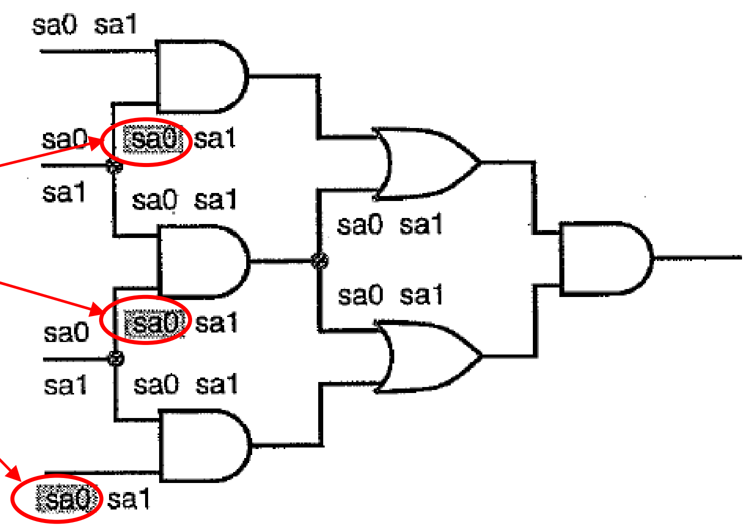
Collapse ratio = 12/30 = 0.40

- Proof: By contradiction.

Checkpoint Theorem

- Definition: Primary inputs and fanout branches of a combinational circuit are called **checkpoints**.
- A test set that detects all single stuck-at faults **at the checkpoints** of a combinational circuit detects all single stuck-at faults in that circuit.
- Example:

Further fault dropping after applying the theorem is possible

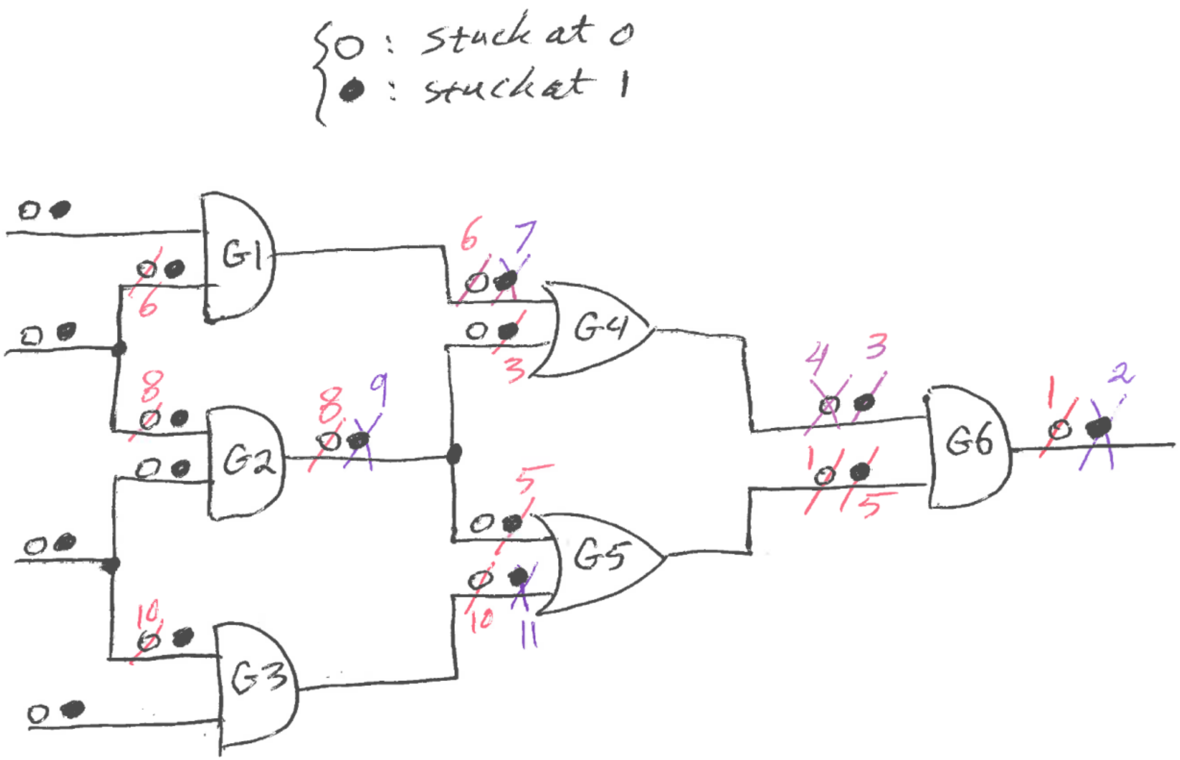


Collapse ratio = 17/32 = 0.53

- Proof: By contradiction.

Method 2: Checkpoint Theorem

- Note: The checkpoint theorem does not claim “optimality”.
- Using fault dropping method may produce better results.
- Example (level by level fault collapsing achieves a smaller fault set):



1: Equivalence for G6
2: Dominance for G6
3: Equivalence for G4
4: Dominance for G4
5: Equivalence for G5
6: Equivalence for G1
7: Dominance for G1
8: Equivalence for G2
9: Dominance for G2
10: Equivalence for G3
11: Dominance for G3

Collapse ratio = 15/32 = 0.47